

Copyright

by

Alexander Robert Hanuska

1998

Thermal Characterization of Complex Aerospace Structures

by

Alexander Robert Hanuska, B.S.

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

Master of Science

in

Mechanical Engineering

Dr. Elaine Scott, Chair
Dr. Brian Vick
Dr. Thomas Diller

April 1998

Blacksburg, Virginia

Thermal Characterization of Complex Aerospace Structures

Approved by
Supervising Committee:

Thermal Characterization of Complex Aerospace Structures

Alexander Robert Hanuska, M.S.

Virginia Polytechnic Institute and State University, 1998

Advisor: Dr. Elaine P. Scott

ABSTRACT

Predicting the performance of complex structures exposed to harsh thermal environments is a crucial issue in many of today's aerospace and space designs. To predict the thermal stresses a structure might be exposed to, the thermal properties of the independent materials used in the design of the structure need to be known. Therefore, a noninvasive estimation procedure involving Genetic Algorithms was developed to determine the various thermal properties needed to adequately model the **Outer Wing Subcomponent (OWS)**, a structure located at the trailing edge of the **High Speed Civil Transport's (HSCT)** wing tip.

Due to the nature of the nonlinear least-squares estimation method used in this study, both theoretical and experimental temperature histories were required. Several one-dimensional and two-dimensional finite element models of the **OWS** were developed to compute the transient theoretical temperature histories. The experimental data were obtained from optimized experiments that were run at various surrounding temperature settings to investigate the temperature dependence of the estimated properties. An experimental optimization was performed to provide the most accurate estimates and reduce the confidence intervals.

The simultaneous estimation of eight thermal properties, including the volumetric heat capacities and out-of-plane thermal conductivities of the facesheets, the honeycomb, the skins, and the torque tubes, was successfully completed with the one-dimensional model and the results used to evaluate the remaining in-plane thermal conductivities of the

facesheets, the honeycomb, the skins, and the torque tubes with the two-dimensional model. Although experimental optimization did not eliminate all correlation between the parameters, the minimization procedure based on the Genetic Algorithm performed extremely well, despite the high degree of correlation and low sensitivity of many of the parameters.

Acknowledgments

I would like to thank the following people for their ample support and help they have given me throughout my research and my thesis work.

Dr. Elaine P. Scott, my advisor and mentor, for giving me the opportunity to work on this project and guiding me throughout my graduate studies.

The Thermal Structures Branch at the NASA-Langley Research Center in Hampton for giving me the resources to perform my work, especially Carl Martin who always had an answer to my many EAL questions.

Kamran Daryabeigi who helped me substantially in performing the experiments at the NASA-Langley Research Center. Thank you for helping me see the forest through the trees, for teaching me the basics of heat transfer, making me a better engineer, and above all for keeping my sanity and for all the premarital advice.

Thank you to Wayne who did a tremendous job installing the sensors on the test panel on such short notice and for giving me an extended tour of the nondestructive and nonintrusive test facilities at the NASA-Langley Research Center..

My committee members, Dr. Brian Vick and Dr. Tom Diller who were partially responsible for my interest in Heat and Mass Transfer and for giving insight and suggestions by answering my many questions.

The Heat Transfer Lab, including Christie Staton, Jim Dolan, Greg Walker, and Paul Robinson for answering all the Linux and L^AT_EX questions, for making the lab a more enjoyable (less productive) place, and for sharing my Corvette dreams.

Thank you to my good friend Michael Grondalski who always found the right words to lure me into the gym and whose diplomatic ways were a great influence. Thanks also to Mr. Chris Fannin, my roommate and friend with whom I shared many memorable moments.

Finally I would like to thank my wife Carol who always found a way to balance my

life despite writing her own thesis. I would like to thank my family Viktor, Mireille, and Nathalie for their continuous support and the pep-talks-calls from Hong-Kong and Berlin.

I would also like to thank my newly acquired parents Dick and Rosemary Steichen for the relaxing getaway weekends in Akron and for fulfilling my Corvette dreams.

ALEXANDER ROBERT HANUSKA

Virginia Polytechnic Institute and State University

April 1998

Contents

Abstract	iv
Acknowledgments	vi
List of Tables	xiii
List of Figures	xv
Nomenclature	xx
Chapter 1 Introduction	1
1.1 Overall Goal and Objectives	2
Chapter 2 Literature Review	5
2.1 Techniques to Determine Thermophysical Properties of Materials	6
2.2 The Nonlinear Least Squares Method	7
2.3 Genetic Algorithms as a Minimization Technique	9
2.4 Design of Experiments	11
Chapter 3 Theory and Mathematical Analysis	13
3.1 Heat Transfer Models	14
3.1.1 Steady-State Analytical Model	14
3.1.2 Numerical Models	17
3.1.3 Boundary Conditions	20
3.2 Optimization and Estimation Techniques	24
3.2.1 Sensitivity Coefficients and Correlation	24
3.2.2 Confidence Intervals	25

3.2.3	Design Optimization	26
3.2.4	Estimation Procedure	28
3.3	The Genetic Algorithm	29
3.3.1	A General Description of the Extended Elitist Genetic Algorithm . .	29
Chapter 4	Experimental Procedures	33
4.1	The Outerwing Subcomponent	33
4.1.1	The Upper and Lower Outer Facesheets	34
4.1.2	The Honeycomb	34
4.1.3	The Upper and Lower Skins	38
4.1.4	The Torque Tubes	38
4.1.5	The Edge Channels	38
4.2	The Experimental Configurations	40
4.2.1	The One-Dimensional Experimental Setup	41
4.2.2	The Two-Dimensional Experimental Setup	42
4.3	The Experimental Hardware	48
4.3.1	The Oven	48
4.3.2	The Insulation	51
4.3.3	The Thermotron	51
4.3.4	The Electric Resistance Heaters	51
4.3.5	The Heat Sink	52
4.3.6	The Thermal Grease	57
4.3.7	The Temperature Sensors	57
4.3.8	The Data Acquisition System	59
4.4	The Experimental Procedure	60
4.5	Experimental Data	62
Chapter 5	Implementation of the Finite Element Program EAL	65
5.1	The Finite Element Program EAL	66
5.1.1	An Overview of EAL	66
5.1.2	EAL's Transient Solution Technique	66
5.2	Determination of Optimal Mesh Size and Time Step	67
5.2.1	The Optimal Mesh Sizes of the Finite Element Program	67

5.2.2	The Optimal Time Steps of the Finite Element Program	68
5.3	An Analytical Validation of the Finite Element Package EAL	71
5.3.1	Implementation of Genetic Processors into EAL	72
Chapter 6	Results and Discussion	76
6.1	Steady-State Analysis	76
6.2	Design Optimization	77
6.2.1	Optimal Sensor Locations	77
6.2.2	Optimal Heating Time	79
6.3	Sensitivity Coefficients	83
6.4	Estimation Results	88
6.4.1	One-Dimensional Effective Estimation Results	88
6.4.2	One-Dimensional Detailed Estimation Results	102
6.4.3	Validation of the One-Dimensional Models	109
6.5	Convergence of the Genetic Algorithm	109
Chapter 7	Summary and Conclusions	116
7.1	Summary	116
7.2	Conclusions	118
7.2.1	Finite Element Models	118
7.2.2	Experimental Optimization	119
7.2.3	Implementation of the Genetic Algorithm	119
7.2.4	Thermal Estimates	119
7.3	Overall Conclusion	120
Chapter 8	Recommendations	121
	Bibliography	123
Appendix A	Calculation of the Effective Thermal Conductivity for the In- sulation Boards	127
Appendix B	Greens Function Solution of the One-Dimensional Effective Model	131
B.1	Problem Statement and Differential Element Representation	131

B.2	Conservation of Energy	131
B.3	Boundary and Initial Conditions	133
B.4	Selection of the Green's Function	133
B.5	Selection of the Eigenfunctions and Eigenvalues	134
B.6	Solution of the Green's Function	135
B.7	Solution of the Temperature Distribution in Terms of the Green's Function	136
B.8	Final Temperature Distribution	137
Appendix C EAL Processors		138
C.1	Frequently Used Processors	138
C.2	The Genetic Processors	139
C.2.1	The GENA PROC Processor	143
C.2.2	The GENB PROC Processor	150
C.2.3	The GENC PROC Processor	161
C.2.4	The GEND PROC Processor	172
C.2.5	The GENE PROC Processor	182
Appendix D The Steady-State Analysis Program Composite.f		189
Appendix E EAL One-Dimensional Effective Insulated Model: Forward Program		221
Appendix F EAL One-Dimensional Effective Convective Model: Forward Program		246
Appendix G EAL One-Dimensional Detailed Model: Forward Program		271
Appendix H EAL One-Dimensional Detailed Half-Model: Forward Program		301
Appendix I EAL One-Dimensional Effective Insulated Model: Estimation Program		329
Appendix J EAL One-Dimensional Convective Model: Estimation Program		345

Appendix K EAL One-Dimensional Detailed Model: Estimation Program	365
Appendix L EAL One-Dimensional Detailed Half-Model: Estimation Program	386
Appendix M EAL Two-Dimensional Model: Estimation Program	407
Appendix N EAL Two-Dimensional Model: Forward Program	502
Vita	596

List of Tables

4.1	Properties of Cotronics' 360 Standard Ceramic Boards	51
4.2	Thermal diffusivity of several materials.	55
4.3	Calibration Values for Thermocouple Spool Type K	59
5.1	The Optimal Mesh Sizes for the Finite Element Models	68
5.2	The Optimal Time Steps for the Finite Element Models	71
5.3	Test Parameters for EAL Verification	72
6.1	Approximate Effective Thermal Conductivity Using Steady-State Analysis .	78
6.2	Optimal Sensor Locations for the One-Dimensional and Two-Dimensional Experimental Setups	79
6.3	Optimal Heating Times for the One-Dimensional and Two-Dimensional Ex- perimental Setups	80
6.4	One-Dimensional Effective Thermal Conductivity Estimates with Insulated Top Boundary	90
6.5	One-Dimensional Effective Volumetric Heat Capacity Estimates with Insu- lated Top Boundary	90
6.6	Average One-Dimensional Effective Thermal Conductivity Estimates with Insulated Top Boundary	92
6.7	Average One-Dimensional Effective Volumetric Heat Capacity Estimates with Insulated Top Boundary	92
6.8	One-Dimensional Effective Thermal Conductivity Estimates with Convective Top Boundary	94
6.9	One-Dimensional Effective Volumetric Heat Capacity Estimates with Con- vective Top Boundary	94

6.10	Average One-Dimensional Effective Thermal Conductivity Estimates with Convective Top Boundary	96
6.11	Average One-Dimensional Effective Volumetric Heat Capacity Estimates with Convective Top Boundary	96
6.12	Effect of the Heat Transfer Coefficient of the Convective Boundary on the Estimated Parameters at 375K	97
6.13	Sum-Of-Squares Objective Functions of Estimated Designs for the One-Dimensional Effective Model with Insulated Boundary Condition	100
6.14	Sum-Of-Squares Objective Functions of Estimated Designs for the One-Dimensional Effective Model with Convective Boundary Condition	101
6.15	One-Dimensional Thermal Conductivity Estimates with Confidence Intervals with Convective Top Boundary for Skins, Honeycomb, Torque Tubes, and Facesheets	105
6.16	One-Dimensional Volumetric Heat Capacity Estimates with Confidence Intervals with Convective Top Boundary for Skins, Honeycomb, Torque Tubes, and Facesheets	106
6.17	Sum-Of-Squares Objective Functions of Estimated Designs for the One-Dimensional Detailed Half-Model	110
6.18	One-Dimensional Effective Thermal Conductivity Estimates with Average and Independent Temperature Histories	111
A.1	Effective Thermal Conductivity of the Insulation Boards as a Function of the Heat Transfer Coefficient.	130
C.1	Overview of the Genetic Processors	142

List of Figures

3.1	Representation of the Outer Wing Subcomponent	14
3.2	Resistance Layout for Steady-State Analysis	16
3.3	One-Dimensional Representation of the Outer Wing Subcomponent Modeled as One Material	19
3.4	One-Dimensional Representation of the Outer Wing Subcomponent Modeled in Detail	21
3.5	Two-Dimensional Representation of the Outer Wing Subcomponent	22
3.6	One-Dimensional Boundary Conditions	23
3.7	Two-Dimensional Boundary Conditions	23
3.8	The Extended Elitist Genetic Algorithm Developed by Sandrine Garcia (1996) With Convergence Criteria	30
4.1	The Location of the Outer Wing Subcomponent on the Fuselage	35
4.2	The Position of the Outer Wing Subcomponent Relative to the Wing Tip .	35
4.3	A View of the Cross-Section of the Outer Wing Subcomponent	36
4.4	The Upper and Lower Outer Facesheets of the Outer Wing Subcomponent .	36
4.5	The Upper and Lower Honeycomb Structures of the Outer Wing Subcomponent	37
4.6	The Honeycomb Edge Closeouts of the Outer Wing Subcomponent	37
4.7	The Upper and Lower Skins of the Outer Wing Subcomponent	39
4.8	The Torque Tubes of the Outerwing Substructure	39
4.9	The Edge Channels of the Outer Wing Substructure	40
4.10	The One-Dimensional Setup	42
4.11	The One-Dimensional Top And Bottom Sensor Layout	43
4.12	The One-Dimensional Inside Sensor Layout	44

4.13	The Two-Dimensional Setup	45
4.14	The Two-Dimensional Top And Bottom Sensor Layout	46
4.15	The Two-Dimensional Inside Sensor Layout	47
4.16	The Overall Test Setup of the Experiment	49
4.17	A View of the Oven Frame and Oven Walls	49
4.18	A View of the Custom Built Oven	50
4.19	The Full-Size Resistance Heater on the Top Surface of the Structure	53
4.20	The Half-Size Resistance Heater on the Top Surface of the Structure	53
4.21	Thermal Imaging of a Plain Resistance Heater	54
4.22	Thermal Imaging of a Resistance Heater Bonded to a Copper Sheet	54
4.23	The Resistance Heater Used to Implement the Heat Flux	55
4.24	A View of the Heat Sink Inside the Oven	56
4.25	Application Locations of the Thermal Grease	58
4.26	The Calibration Curve for the Thermocouple Wire	60
4.27	Typical Experimental Temperature Distributions	63
4.28	Typical Offset Experimental Temperature Distribution	64
5.1	The Influence of the Mesh Size on the Temperature at the Optimal Sensor Location for the Effective One-Dimensional Model with a Nominal Mesh Size of 198 Elements and Time Step of 0.005 sec	69
5.2	The Influence of the Mesh Size on the Temperature at the Optimal Sensor Location for the Detailed One-Dimensional Model with a Nominal Mesh Size of 222 Elements and Time Step of 0.005 sec	69
5.3	The Influence of the Time Step on the Temperature at the Optimal Sensor Location for the One-Dimensional Effective Model with a Nominal Mesh Size 198 Elements and Time Step of 0.005 sec	70
5.4	The Influence of the Time Step on the Temperature at the Optimal Sensor Location for the Detailed One-Dimensional Model with a Nominal Mesh Size of 222 Elements and a Time Step of 0.005 sec	70
5.5	A Comparison of the Temperature History Solutions at the Heated Surface of the Outer Wing Subcomponent Obtained with the Finite Element Model and the Green's Function Method	73

5.6	Residuals of the Temperature History Solutions at the Heated Surface of the Outer Wing Subcomponent Obtained with the Finite Element Model and the Green's Function Method	73
5.7	Comparison of the Extended Elitist Genetic Algorithm Developed by Sandrine Garcia (1997) With Convergence Criteria and the EAL Genetic Processors	75
6.1	Effective Thermal Conductivity from Steady-State Analysis	78
6.2	Optimal Sensor Location for the One-Dimensional Experimental Setup . . .	80
6.3	Optimal Sensor Location for the Two-Dimensional Experimental Setup . .	81
6.4	Heating and Cooling Phases of the One-Dimensional Experimental Setup .	82
6.5	Heating and Cooling Phases of the Two-Dimensional Experimental Setup .	82
6.6	Comparison Between the Sensitivity Coefficients of the Thermal Conductivity and Volumetric Heat Capacity for the One-Dimensional Effective Model with and without Convective Top Boundary	85
6.7	One-Dimensional Half-Model Derived from Detailed One-Dimensional Model	86
6.8	Sensitivity Coefficients for the One-Dimensional Detailed Model	87
6.9	Sensitivity Coefficients for the Two-Dimensional Model	87
6.10	Thermal Conductivity Estimates with Confidence Intervals for the One-Dimensional Effective Model with Insulated Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	91
6.11	Volumetric Heat Capacity Estimates with Confidence Intervals for the One-Dimensional Effective Model with Insulated Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	91
6.12	Average Thermal Conductivity Estimates with Confidence Intervals for the One-Dimensional Effective Model with Insulated and Convective Boundary at Surrounding Temperatures of 295K, 375K, and 445K	92
6.13	Average Volumetric Heat Capacity Estimates with Confidence Intervals for the One-Dimensional Effective Model with Insulated and Convective Boundary at Surrounding Temperatures of 295K, 375K, and 445K	93
6.14	Thermal Conductivity Estimates with Confidence Intervals for the One-Dimensional Effective Model with Convective Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	95

6.15	Volumetric Heat Capacity Estimates with Confidence Intervals for the One-Dimensional Effective Model with Convective Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	95
6.16	Experimental and Calculated Temperature Histories for the One-Dimensional Effective Model with Insulated Boundary Condition using the Thermal Property Estimates	99
6.17	Experimental and Calculated Temperature Histories for the One-Dimensional Effective Model with Convective Boundary Condition using the Thermal Property Estimates	99
6.18	Residuals between Experimental and Calculated Temperature Histories for the One-Dimensional Effective Model with Insulated Boundary Condition .	103
6.19	Residuals between Experimental and Calculated Temperature Histories for the One-Dimensional Effective Model with Convective Boundary Condition	103
6.20	Thermal Conductivity Estimates for the One-Dimensional Detailed Model with Convective Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	107
6.21	Volumetric Heat Capacity Estimates for the One-Dimensional Detailed Model with Convective Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	107
6.22	Average Thermal Conductivity Estimates for the One-Dimensional Detailed Model with Convective Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	108
6.23	Average Volumetric Heat Capacity Estimates for the One-Dimensional Detailed Model with Convective Top Boundary at Surrounding Temperatures of 295K, 375K, and 445K	108
6.24	Experimental and Calculated Temperature Histories for the One-Dimensional Detailed Model using the Thermal Property Estimates	110
6.25	Residuals between Experimental and Calculated Temperature Histories for the One-Dimensional Detailed Model	112
6.26	Convergence of the Genetic Algorithm	114
6.27	Convergence of the Average of the 20 Best Objective Functions	115

A.1	Convection Rolls	129
A.2	Resistance Diagram for Insulation and Convection	130
B.1	One-Dimensional Representation of the Outer Wing Subcomponent	132
B.2	Differential Element Representation of the One-Dimensional Effective Model	132
C.1	EAL Genetic Runstream	141
C.2	The GENA PROC	144
C.3	The GENB PROC	152
C.4	The GENC PROC	163
C.5	The GEND PROC	173
C.6	The GENE PROC	183

Nomenclature

A	surface area
C	volumetric heat capacity (W/m^3K)
c_p	specific heat ($kJ/kg K$)
E	applied voltage (V)
G	Green's function
h	convection coefficient (W/m^2K)
I	applied current (A)
\mathbf{K}	stiffness matrix
k	thermal conductivity ($W/m K$)
L	length
n_{best1}	Elitist parameters used in EEGA
n_{best2}	Elitist parameters used in EEGA
n_{best3}	Elitist parameters used in EEGA
n_c	number of chromosomes
n_g	number of generations
n_p	number of parameters to be estimated
n_{pop}	number of initial populations randomly searched in EEGA
n_s	population size
q''	heat flux (W/m^2)
R	thermal resistance
S	objective function (K^2)
s	standard deviation
t	time (sec)

t_0	initial time (<i>sec</i>)
$t_{1-\alpha/2}$	student's t-distribution
T	calculated temperature (<i>K</i>)
T_i	initial temperature
V	variance of distribution
X	sensitivity matrix
x	space coordinate (<i>m</i>)
Y	measured temperature (<i>K</i>)
<u>Superscript</u>	
n	number of observations
T	transpose
<u>Subscripts</u>	
<i>cond</i>	conduction
<i>cont</i>	contact
<i>conv</i>	convection
<i>eff</i>	effective
<i>i</i>	sensor location
<i>kk</i>	diagonal term of a matrix
<i>m</i>	fluid type
<i>MAP</i>	Maximum A Posteriori Estimator
<i>ML</i>	Maximum Likelihood Estimator
<i>N</i>	number of observations
<i>n</i>	number of experiments
<i>OLS</i>	Ordinary Least Squares Estimator
<i>p</i>	number of parameters
<i>rad</i>	radiation
<i>s</i>	surface
<u>Greek</u>	
α	thermal diffusivity (m^2/s)
β	estimated parameter
β_m	eigenvalues
δ	incremental

ϵ	emissivity
η	calculated temperature vector (K)
μ	mean of distribution
Ψ	covariance matrix
σ	Stefan-Boltzmann constant

Chapter 1

Introduction

By the year 2007, the new **H**igh **S**peed **C**ivil **T**ransport (**HSCT**) will be replacing the older generation Concorde, providing the customer with shorter trip times, lower fares, and more cabin space. The **HSCT** project required many different areas of research, heat transfer being one of them.

Due to friction caused by the speed of flight (Mach 2.5), parts of the wings and fuselage of the **HSCT** will be exposed to harsh thermal environments. Thermal stresses are a direct consequence of the steep thermal gradients encountered by the wings and fuselage and can negatively affect the performance of the structures. Therefore, in order to determine the thermal stresses developed in the structures, the thermal properties of the various materials used in its construction must be known.

The **O**uter **W**ing **S**ubcomponent (**OWS**), located at the trailing edge of the **HSCT**'s wing tip, is a perfect example of a structure exposed to such harsh thermal environments and will be used to determine a new noninvasive method of estimating thermal properties of complex aerospace structures.

Complex structures such as the **OWS** are designed with new light and high strength-to-weight ratio materials including carbon/fiber/epoxy composites and honeycomb blankets. Many new materials are developed to fulfill the needs of a continuously advancing industry requiring lighter and stronger materials. Much work has been done to develop methods to estimate the thermal properties of these materials either independently or simultaneously. However, many of these methods are invasive or require material samples. Furthermore, it would be advantageous to be able to model and predict the performance of an assembled

structure and its components without having to dismantel it.

Therefore, the motivation for this research was to noninvasively determine the effective thermal properties of the entire structure as well as the in-plane and out-of-plane independent thermal properties of the materials within the **OWS**. The specific objectives set for this study are described in the following section.

1.1 Overall Goal and Objectives

The development of the noninvasive property estimation technique for complex structures involved several steps and short term objectives. These objectives are to:

1. develop a simple thermal resistance model of the **OWS** to gain initial insight in the physics of the problem;
2. develop two one-dimensional and one two-dimensional finite element model to perform analytical thermal analyses, which include conduction, convection, and radiation;
3. determine the optimal experimental parameters to be used to set up the one-dimensional and two-dimensional experiments to collect data for the parameter estimation procedure;
4. assemble the one-dimensional and two-dimensional experimental setups, conduct the experiments, and record the experimental data;
5. introduce a Genetic Algorithm based estimation procedure into the finite element models; and,
6. estimate the thermal properties in the one-dimensional and two-dimensional heat transfer models.

The first objective was to investigate the contribution of each component of the **OWS** to the overall heat transfer in the structure. The results of this initial analysis were used to determine which components were essential to the one-dimensional model of the **OWS** to properly mimic the heat transfer in the structure. The thermal resistance model also yielded an approximate value for the effective thermal conductivity which helped determine the initial parameter ranges in the Genetic Algorithm during the estimation process.

With the results of the simple thermal resistance model in mind, several finite element heat transfer models were developed for the **OWS**. Due to the complexity of the geometry and the variety of the materials used to assemble the structure, the finite element technique was the most appropriate choice to solve for the analytical temperature distributions of the **OWS**. Two one-dimensional models were created. The first model was geared towards estimating the effective thermal properties of the structure, the second, more detailed, was created to determine the thermal properties of the independent materials. The two-dimensional model was needed to determine the remaining in-plane thermal conductivities of the independent materials. It included all the details of a cross-section of the **OWS**.

The nonlinear regression technique requires both theoretical and experimental data. In order to obtain the most accurate estimates and reduce the width of the confidence intervals, the experiments should be designed to provide the maximum information possible for the estimation of the desired properties. Therefore, experimental optimization was necessary to determine the optimal heating time, the total experimental time and sensor locations to maximize the sensitivity of the recorded temperature with respect to the estimated parameters. The findings of the optimization were then applied to the experimental setup.

Two independent experimental setups were created to collect data for the one-dimensional and two-dimensional models. Each setup, including the oven construction, sensor positioning, and data acquisition system, was established at the Thermal Structures Branch Lab, NASA Langley Research Center in Hampton, VA. The one-dimensional experimental setup consisted of an aluminum heat sink imposing a constant temperature boundary condition on the lower surface of the outer wing subcomponent, a heater covering the entire top surface of the structure to introduce a temperature gradient through the sample, and insulation on the sides and over the heater to minimize the heat lost to the surroundings. The two-dimensional experimental setup was almost identical to the one-dimensional one, with the exception of the heater which only covered half of the top surface of the structure to introduce lateral heat transfer. Variable surrounding temperatures were obtained by placing the sample in an oven and heating the oven to a specified temperature to determine the temperature dependence of the thermal properties.

A parameter estimation technique was developed to estimate the thermal properties of the materials in the **OWS**. The Genetic Algorithm written by Garcia et al. (1998) was incorporated into the finite element models via processors and was used as minimization procedure. The minimization procedure was applied to both the optimization and estimation of parameters.

Finally, the estimation procedure was applied by combining the experimentally recorded data, the finite element models, and the minimization procedure to determine the thermal properties. The thermal properties estimated included the effective thermal conductivity and volumetric heat capacity of the structure and the in-plane and out-of-plane thermal conductivity and volumetric heat capacity of the independent materials, including the facesheets, the skins, the honeycomb blanket, the torque tubes and the air.

Chapter 2

Literature Review

Knowing the thermophysical properties of structures comprised of advanced materials has become a crucial issue in many of today's designs due to the correlation between a material's performance and its thermal properties. This is particularly true when the structures are exposed to extremely dynamic thermal environments such as in aerospace applications, where thermal stresses can be significant. It is therefore important, from the design standpoint, to be able to model the structure's thermal response to determine the resulting thermal stresses under these conditions. Thus the first step is to determine the structure's relevant thermal properties in order to model the thermal response. Note here, that typical structures in aerospace applications can be quite complex, and might include carbon and metallic composites and other advanced materials. Much work has been done to develop various methods to accurately determine a material's thermal properties, including the thermal diffusivity α , the heat capacity c_p , and the thermal conductivity k . However, many of these techniques require samples of each individual material to determine its thermal properties. This requirement is not always a feasible option with already assembled complex structures such as the outer wing subcomponent that involve several materials at once. Therefore, a noninvasive parameter estimation technique was adopted in this study to estimate the thermal properties of the outer wing subcomponent, a complex structure, without having to disassemble it. Among the thermal properties estimated were not only the effective thermal properties, but also the one-dimensional and two-dimensional thermal properties of the individual materials used in the design of the outer wing subcomponent.

This chapter discusses several techniques used to determine thermal properties of

materials, the previous work that has been done in the area of parameter estimation, how to design optimal experiments, and finally describes previous uses of **Genetic Algorithms (GA)** which were introduced in this study as minimization procedure.

2.1 Techniques to Determine Thermophysical Properties of Materials

Various steady-state and non-steady-state techniques have been developed to determine thermal properties in independent materials. Among the most common methods in determining thermal properties is the flash method first described by Parker et al. (1961). By exposing the front surface of the material sample to a short radiant energy pulse and recording the temperature history of the sensor equipped back surface of the sample, Parker et al. (1961) were able to determine the thermal diffusivity α , the heat capacity c_p , and the thermal conductivity k of copper, silver, iron, nickel, aluminum, tin, zinc, and some alloys. The laser-flash method was also adopted by Taylor et al. (1985) to evaluate the thermal diffusivity of fiber-reinforced composites, whereas Brennan et al. (1982) took advantage of the composite method, a combination of the laser-flash method and the Rule-of-Mixtures, to measure the thermal conductivity and diffusivity of fibres of amorphous silicon carbide contained within an aluminosilicate glass-ceramic.

The transient hot wire technique is another approach to measure thermal properties of materials. The method is based on the change of the temperature in time of a thin metallic wire in response to a step change in voltage applied to it. A variation of the hot-wire technique called the periodic hot-wire heating method was adopted by Fukai et al. (1991) to simultaneously measure the thermal conductivity and diffusivity of packed beds of aluminum oxide particles and potassium perchlorate particles. In his study, Fukai et al. (1991) showed that the thermophysical properties of the materials could be determined from the amplitude and phase lag of the temperature response within the sample cylinder.

Both the Searle's bar system and the Lees' disc system determine the thermal properties of materials by heating one end of the samples and measuring the temperature gradient along them. The major difference between the two methods is the sample shape which, as the name infers, is rectangular for the Searle's bar system and round for the Lees' disc system. The Searle's bar system with small modifications and the Lees' disc system were

used by Pilling et al. (1979) to evaluate the in-plane and out-of-plane thermal conductivities of a series of unidirectional and bidirectional specimens of epoxy resin $DX210/BF_3400$ reinforced with Morganite high modulus (HMS) and high strength (HTS) carbon fibres. The disc apparatus was also employed by Harris et al. (1982) to report the thermal conductivities of Shell $DX210/BF_3400$ resin reinforced with Kevlar 49 fibres in directions parallel and perpendicular to their length.

The heat flow sensor technique described by Penn et al. (1986) was used to acquire the values of the thermal conductivity of two composites that consist of fibrous or granular components embedded in a resin matrix. It was reported that the results were similar in accuracy to the ones from the guarded hotplate.

Unfortunately, the previously stated techniques all require material samples that need to be cut or shaped to fit the test apparatus of choice. In the case of complex structures, removing samples of each individual component of the structure represents a major inconvenience and is not always feasible. Therefore, a methodology was adopted in this study to simultaneously estimate the thermal properties (thermal conductivity k and volumetric heat capacity $C = \rho c_p$) of the different materials present in the outer wing subcomponent. Thus, the entire subcomponent was used as the test specimen to estimate the different thermal properties.

2.2 The Nonlinear Least Squares Method

The area of parameter estimation consists of various estimation methods, some of which were described in the previous section. The majority of these methods require both experimental and theoretical data to perform the estimation which is usually provided by an experimental setup and a fairly simple heat transfer model such as some linearized relationship. The nonlinear least squares method introduced to the field of heat transfer by Beck (1964) and Beck (1966) also relies on both transient experimental data, including the amount of heat applied to the structure or material and the temperature histories at different locations on the sample, and theoretical temperature histories that can be obtained from a mathematical model. In this method, an appropriate objective function is formed, depending on the amount of information available on the experimental data and the parameters to be estimated. Beck and Arnold (1977) describe three different estima-

tion schemes, which differ by the amount of information included in the specified objective function. The name of the estimation schemes and the corresponding expressions for their objective functions are listed below:

- **Ordinary Least Squares estimation (OLS)**

$$S_{OLS} = \sum_{i=1}^n [\mathbf{Y}_i - \eta_i]^T [\mathbf{Y}_i - \eta_i]$$

The **OLS** estimator requires no prior knowledge about either the parameters estimated or the variances of the measurement errors.

- **Maximum Likelihood estimation (ML)**

$$S_{ML} = \sum_{i=1}^n [\mathbf{Y}_i - \eta_i]^T \Psi^{-1} [\mathbf{Y}_i - \eta_i]$$

The **ML** estimator requires prior knowledge about the variances of the measurement errors.

- **Maximum A Posteriori estimation (MAP)**

$$S_{MAP} = \sum_{i=1}^n [\mathbf{Y}_i - \eta_i]^T \Psi^{-1} [\mathbf{Y}_i - \eta_i] + [\beta - \mu_\beta]^T V_\beta^{-1} [\beta - \mu_\beta]$$

The **MAP** estimator requires both prior knowledge about the variances of the measurement errors and the estimated parameters.

Once the appropriate objective function has been determined, a minimization technique can be applied to find the optimal design in the parameter space for which the value of the objective function is the smallest.

In many cases, the ordinary least squares estimator **OLS** is the estimator of choice because of the small amount of information required to implement it. Due to the lack of prior information about both the estimated parameters and the variances of the measurement errors, the **OLS** estimator was also used in this study to describe the objective function.

The majority of the methods mentioned in the previous section lend themselves to investigating the thermal properties of one material at a time. The nonlinear least squares methods, on the other hand, can be applied to the simultaneous estimation of multiple properties in complex multidimensional structures with one or more materials.

Multiple property estimation was performed by Pfahl (1966), who solved a seven parameter problem describing the internal ablation mechanism of a polymeric material by applying the nonlinear least squares method. Six thermal properties of a charring carbon-phenolic ablator, an anisotropic material, were estimated by Pfahl and Mitchell (1970)

also using the nonlinear least squares method. The estimates were compared with results obtained from conventional methods to emphasize the validity of the method.

Multidimensional property estimation was performed by Dowding et al. (1996), who estimated the thermal conductivity normal and parallel to the fibers as well as the volumetric heat capacity of an orthotropic carbon-carbon composite using a two-dimensional model. The results of the two-dimensional analysis confirmed the previous experimental findings for the thermal conductivity normal to the fibers and the volumetric heat capacity of the same material performed by Dowding et al. (1995) using a one-dimensional model. The two-dimensional results obtained for the thermal conductivity normal to the fibers coincided well with the results of the one-dimensional model within 6%.

2.3 Genetic Algorithms as a Minimization Technique

As discussed previously, the nonlinear regression method requires both experimental and theoretical data to form the least squares objective function used in this study. However, the minimization procedure used to determine minima of the objective function can be chosen among a variety of different techniques. The Gauss linearization technique was initially used by Beck (1966) to simultaneously estimate the thermal conductivity and specific heat of Nickel. Scott and Beck (1992) also used this method to simultaneously determine the temperature dependent thermal conductivity and volumetric heat capacity of carbon composites.

The Gauss linearization method was improved by Box and Kanemasu (1972) and Bard (1970) to try to circumvent the difficulties of slow or non-convergence in cases of extremely flat objective functions, when two or more parameters were nearly correlated, or when the objective function had more than one minimum. The modified Box-Kanemasu minimization technique was used by Moncman (1994), Hanak (1995), (also in Moncman et al. (1995)), Copenhaver (1996) (also in Copenhaver et al. (1997)), and Guynn (1996) (also in Garcia et al. (1998)) to perform various thermal parameter estimations. However, the limitations of the modified Box-Kanemasu minimization technique became apparent for Copenhaver (1996) and Guynn (1996), when the simultaneous estimation of nearly correlated parameters failed due to non-convergence of the objective function. Copenhaver (1996) implemented a penalty function technique to successfully minimize the objective

function and estimate the thermal capacitance of the facesheets, the core thickness, and the emissivity of a honeycomb structure. Guynn (1996) resorted to Genetic Algorithms to estimate multi-mode heat transfer properties in styrofoam, even with the existing correlation between the thermal conductivity and the radiation parameters, and between the heat capacity of the styrofoam and the heat capacity of the heater.

Genetic algorithms were developed by John Holland in 1975 who was convinced that difficult problems could be solved using simple encodings and reproduction mechanisms such as mutation and cross-overs, that can be found in nature. The laws of evolution were therefore incorporated in Genetic Algorithms that were designed to be used on digital computers. In Genetic Algorithms, each parameter is associated with a gene. A chromosome, on the other hand, is a combination of genes and represents one specific solution or design.

The original Genetic Algorithms relied on binary encoding, however, Furuya and Haftka (1993) used integer encoding to determine the optimal locations of actuators on space structures. Carroll (1996a) and Carroll (1996b) successfully applied Genetic Algorithms to chemical laser modeling and performance optimization of a laser system. Jones et al. (1996) used Genetic Algorithms to determine the optical properties and the particle size distribution function of propellant smoke from angular light-scattering measurements.

Few cases were found where Genetic Algorithms were used in heat transfer applications. Fabbri (1997) applied Genetic Algorithms to thermal design, when he optimized the thermal performance of finned surfaces. The performance of the finned surface was increased by introducing undulations in its longitudinal profile in cases where the heat transfer coefficient was very low. Raudensky et al. (1995) reported using Genetic Algorithms to solve an inverse heat conduction problem. For a simple one-dimensional case, he demonstrated that the Genetic Algorithm was able to give him excellent estimates for the heat transfer coefficient. As mentioned earlier, Guynn (1996) was able to estimate multi-mode heat transfer properties in styrofoam. Garcia et al. (1998) used Genetic Algorithms to simultaneously estimate the in-plane and orthogonal thermal conductivities of Hanak (1995)'s problem, despite the correlation between the thermal properties.

2.4 Design of Experiments

The accuracy of the thermal properties estimated with the nonlinear least squares technique can be greatly improved by designing optimal experiments that provide maximum sensitivity of the measured temperature with respect to the parameters estimated, that is, the data provides the maximum information possible, (Beck and Arnold, 1977). Therefore this results in an increase in the confidence in the estimates. Beck and Arnold (1977) discuss three major distinct optimization criteria which are all based on the product of the sensitivity coefficients and their transpose, namely $\mathbf{X}^T\mathbf{X}$. The D-optimal criterion maximizes the determinant of the $\mathbf{X}^T\mathbf{X}$ matrix, the A-optimal criterion maximizes the trace of the $\mathbf{X}^T\mathbf{X}$ matrix, whereas the E-optimal criterion maximizes the minimum eigenvalue of the $\mathbf{X}^T\mathbf{X}$ matrix. Other optimization criteria such as the C, L, V criteria, and all variants of the D-criterion, are discussed by Scott and Haftka (1995), who also describe how to design experiments to obtain a maximum amount of information, optimization using experiments when analytical optimization is not an option, and finally the validation of optimization through experiments.

The D-optimal criterion was used by Taktak et al. (1993) to optimize the dimensionless heating and total experimental times at the heated boundary and the location of the temperatures sensors for experiments involving one-dimensional heat conduction within cured composite materials and within composite materials undergoing curing. Optimizing for the experimental design parameters used in transient one-dimensional experiments proved to increase the accuracy of the estimates for the thermal conductivity and volumetric heat capacity of carbon-fiber/epoxy matrix composite materials.

The D-optimal criterion, introduced by Beck (1969) and one of the more commonly used criterion, was also used by Moncman et al. (1995) and Hanak (1995) to optimize one-dimensional and two-dimensional experiments for the in-plane and perpendicular sensor locations, the heating area, the heating time, and total experimental time for two different sets of boundary conditions. The experiments involved the simultaneous estimation of thermal conductivity and volumetric heat capacity of anisotropic composite materials. The same parameters were optimized by Garcia and Scott (1998) as test cases for an Extended Elitist Genetic Algorithm. The results proved to be very encouraging since they were in complete agreement with the optimization results determined with conventional methods.

Continuous improvements are made regarding the experimental setups and the time required to prepare experiments. Garnier et al. (1994) proposed solutions to overcome the affect of nonuniform heating patterns of many of the available conventional resistance heaters on the estimation results, and to decrease the overall time to obtain estimates for the thermal conductivity and volumetric heat capacity by using resistance thermometers instead of thermocouples or RTD's.

Chapter 3

Theory and Mathematical Analysis

The discussion in this chapter revolves around the mathematical models used to simultaneously estimate the thermal conductivity k and volumetric heat capacity C of several materials incorporated in the design of the outer wing subcomponent, shown in Fig. 3.1. Due to the asymmetric geometry of the panel and the anisotropic nature of the materials (e.g., woven carbon-fiber composites, honeycomb blanket), the volumetric heat capacity and both the in-plane and through the thickness thermal conductivities were estimated. Genetic Algorithms were implemented in the property estimation process as well as in the optimization of experiments. Genetic algorithms require both calculated and experimental temperature distributions to carry out the minimization procedure discussed later in this chapter. The experimental temperature distributions were obtained through experimental testing and are explained in more detail in Chapter 4. The calculated temperature distributions on the other hand, were acquired through numerical models described in this chapter.

The first section in this chapter outlines the heat transfer models developed to analyze the outer wing subcomponent. Firstly, an initial one-dimensional steady-state analysis was performed to obtain a better understanding of the structure involved, and to provide an approximate value for the effective thermal conductivity k_{eff} through the thickness. Secondly, one-dimensional and two-dimensional numerical models created to obtain the calculated temperature distributions in the outer wing subcomponent are described. Thirdly, the boundary conditions implemented in the numerical models are explained. The second section concentrates on the optimization, estimation, and minimization procedures used,

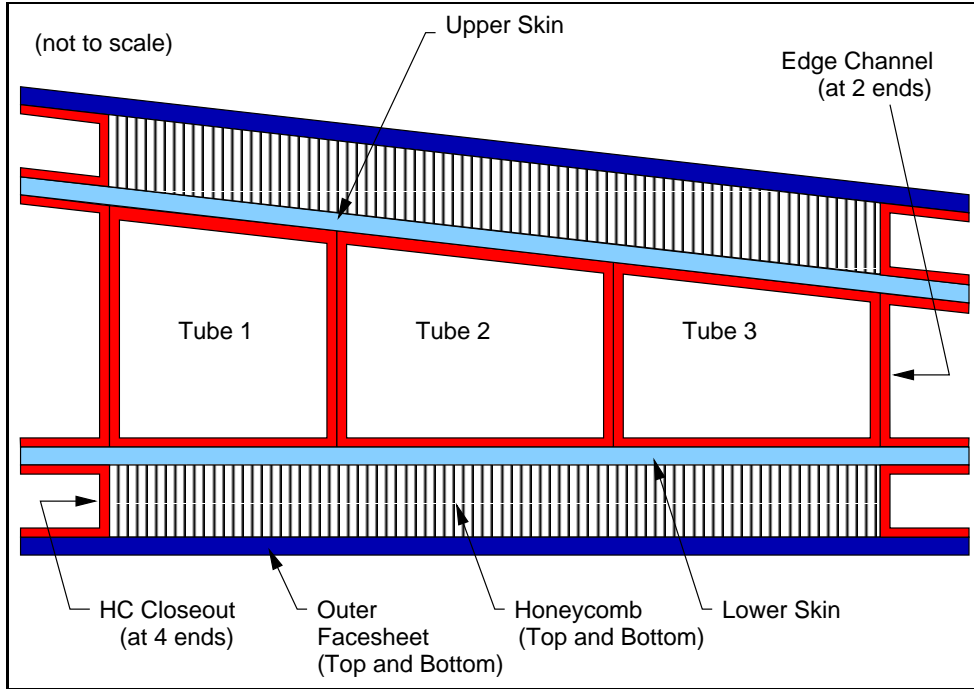


Figure 3.1 Representation of the outer wing subcomponent.

including the sensitivity coefficient analysis, and the confidence intervals. The last section describes the implementation of the Genetic Algorithms as minimization procedure.

3.1 Heat Transfer Models

Analytical solution techniques can be applied to many heat transfer problems, but are limited to relatively simple applications. For problems that are geometrically involved or for which common assumptions are not applicable, numerical methods are advisable to obtain representative temperature distributions. Many problem specific numerical solution techniques are available, although for the problem presented in this study, the finite element method was decided to be the most appropriate. Preceding the more detailed descriptions of the finite element models is the one-dimensional steady-state thermal resistance analysis described below.

3.1.1 Steady-State Analytical Model

A preliminary analytical steady-state analysis was conducted on a one-dimensional cross-section of the outer wing subcomponent, to determine the magnitude of the effective thermal

conductivity k_{eff} that could be expected during the estimation process. Fig. 3.2 shows the thermal circuit that was analyzed to determine effective thermal properties at steady-state conditions. This procedure also helped to reduce the range of the parameter values for the initial population in the genetic algorithm based estimation procedure described later in this chapter. During the steady-state analysis, a constant temperature boundary condition was applied to the bottom of the structure. The top of the structure was exposed to a constant heat flux, whereas the rest of the outer wing subcomponent was assumed insulated. Therefore, only heat conduction through the thickness of the structure was taken into consideration, thus disregarding any lateral heat conduction effects.

The analysis was performed using the principles of thermal resistances. During the setup, different thermal resistances such as conduction, convection, and radiation resistances were encountered. Contact resistances at the interfaces of the materials were also introduced into the analysis, and were treated like thermal convection resistances. Materials such as the upper and lower outer facesheets, the honeycomb blankets, and the upper and lower skins were represented with thermal conduction resistances only. The analysis of the C-shaped channels such as the honeycomb closeouts and the edge channels as well as the torque tube structures all involved conduction in the tube wall and the air, radiation, and convection.

The first thermal resistance encountered when starting at the top of the thermal resistance diagram in Fig. 3.2 represents the conduction through the upper outer facesheet ①. The three following parallel heat transfer paths correspond to the two honeycomb closeouts (② and ④) and the honeycomb blanket ③, and all start with a contact resistance between the outer facesheet and either the honeycomb blanket or the honeycomb closeouts. As mentioned earlier, the honeycomb blanket was modeled as a uniform material with a thermal conduction resistance. The honeycomb closeouts, on the other hand, consist of two distinct thermal paths. The first includes a conduction resistance through the material, followed by conduction, radiation and convection resistances of the air pocket, followed by another conduction resistance through the material. The second path consists of a conduction resistance through the material. The upper skin ⑤ is in contact with the two honeycomb closeouts (② and ④) and the honeycomb blanket ③ on the upper surface, and with the two edge channels (⑥ and ⑩) and three torque tubes (⑦, ⑧, and ⑨) on the bottom surface. As previously discussed, the upper skin ⑤ was modeled as a conduction resistance. The analysis for the edge channels (⑥ and ⑩) is identical to the one established for the honeycomb

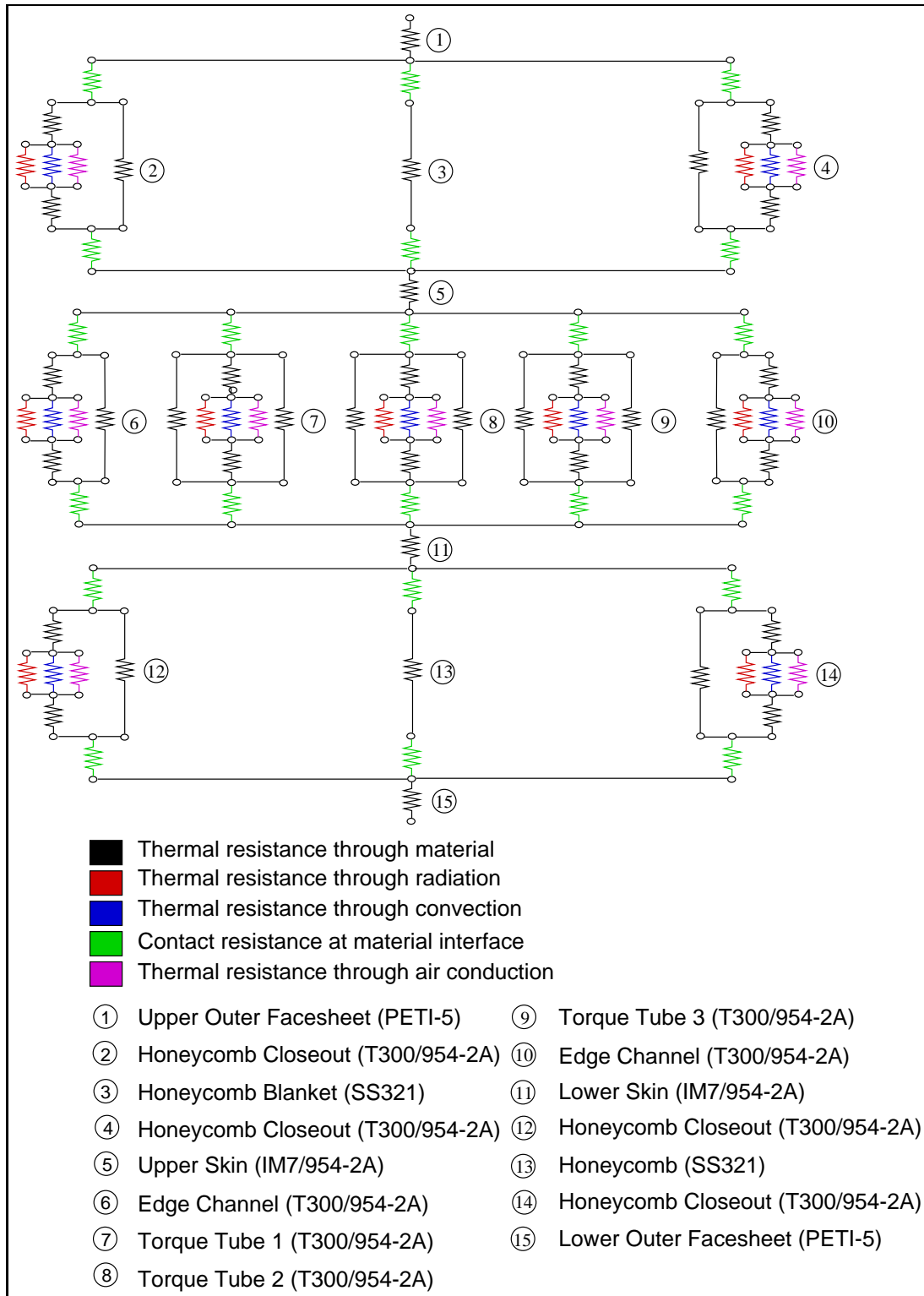


Figure 3.2 Resistance layout for steady-state analysis.

closeouts. The setup for the torque tubes (⑦, ⑧, and ⑨) is also similar, but includes a third path that, unlike the path through the air pocket, only consists of a conduction resistance through the tube wall and is identical to the second. The bottom of the entire structure is a replica of the top with the lower skin ①①, the honeycomb closeouts (①② and ①④), the honeycomb blanket ①③, and the lower outer facesheet ①⑤.

The different thermal resistances were calculated using Eqs. 3.1-3.4.

$$R_{m,cond} = \frac{L}{k_m A_{cond}}, \quad (3.1)$$

$$R_{m,conv} = \frac{1}{h_m A_{conv}}, \quad (3.2)$$

$$R_{m,rad} = \frac{1}{\sigma \epsilon_m A_{rad}}, \quad (3.3)$$

$$R_{m,cont} = \frac{1}{h_m A_{cont}}, \quad (3.4)$$

where the subscript m stands for the material or fluid type. The thermal resistances were then added in parallel or series, depending on their location and connectivity with respect to each other.

The FORTRAN program, `Composite.f`, developed to perform the steady-state analysis takes into account, not only the exact geometry of the individual subcomponents of the panel, but also the temperature dependency of the material properties. The material properties to be estimated in this study were replaced by properties of similar materials, in order to obtain an approximate value for the effective thermal conductivity. The analysis was performed over a temperature window, ranging from 295K to 445K, in 15 degree increments. The FORTRAN program `Composite.f` can be found in Appendix D.

3.1.2 Numerical Models

One-Dimensional Numerical Models

After the initial steady-state analysis was performed, more systematic and detailed numerical models were developed, for the optimization and estimation processes. Following, are

descriptions of the two one-dimensional and the two-dimensional finite element models that were written in the Engineering Analysis Language (EAL). The details of a sample EAL runstream is described in Chapter 5.

Two one-dimensional finite element models were developed and incorporated into the optimization and estimation programs. The first model was created to estimate the effective thermal properties of the outer wing subcomponent, whereas the second one was designed to estimate the individual thermal properties of the materials used in the construction of the outer wing subcomponent. The following subsections give more detailed explanations of the nature of the models and some of the features that were incorporated into the individual models.

Effective One-Dimensional Model The first one-dimensional model of the outer wing subcomponent was developed with the intent to estimate the effective volumetric heat capacity C_{eff} and thermal conductivity k_{eff} through the thickness of the outer wing subcomponent. The different materials used to build the outer wing subcomponent were fused together and modeled as one effective material with effective thermal properties in the in-plane direction.

Several modifications were undertaken when going from the three-dimensional outer wing subcomponent to the effective one-dimensional model illustrated in Fig. 3.3. The first important step in the simplification process was to assume that there were no significant temperature changes through the depth of the structure, therefore eliminating any thermal effects in that specific direction. The validity of the previous assumption can be justified by observing the amount of change of the cross-sectional area of each component of the structure, when progressing in depth. The second important step in the simplification process was to determine a representative height of the effective one-dimensional model. To find the appropriate height of the model, the trapezoidal shape of the cross-section of the outer wing subcomponent was replaced by a rectangular box with, as height, the average of the highest and lowest locations of the trapezoid. Any lateral thermal effects that were eliminated in the effective one-dimensional model, as a result of the previously mentioned simplification step, would be absorbed into the estimates of the effective thermal properties of the material modeled.

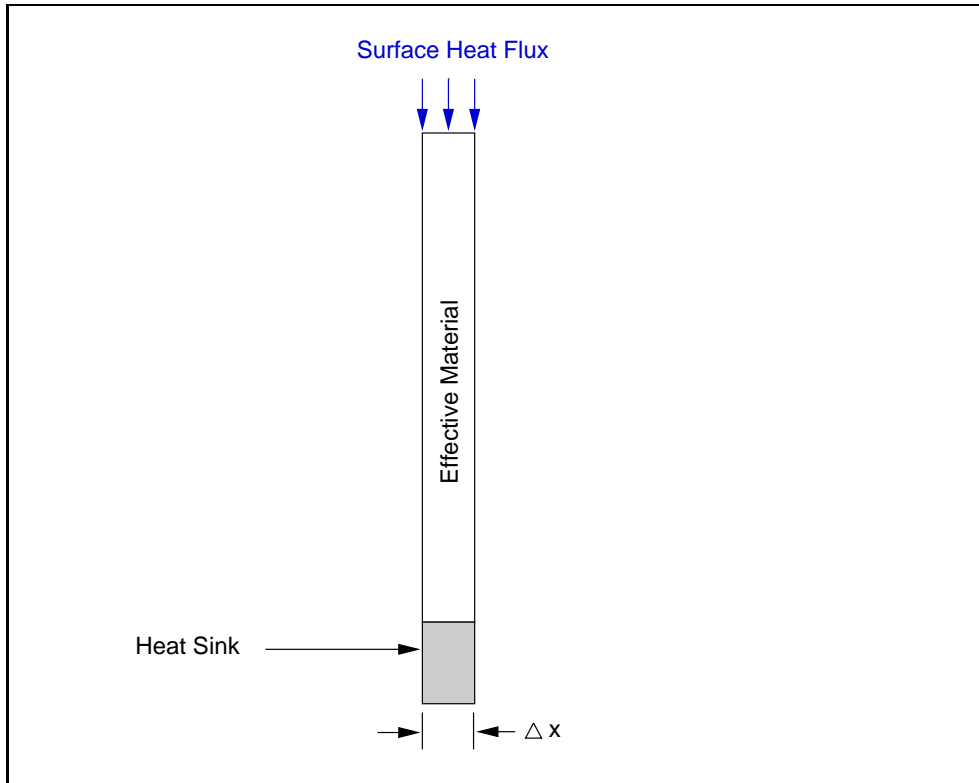


Figure 3.3 One-dimensional representation of the outer wing subcomponent modeled as one material.

Detailed One-Dimensional Model The second one-dimensional model was developed in much more detail, including the interfaces between the materials, making the model as realistic as possible, as illustrated in Fig. 3.4. This model gave us the possibility to estimate several additional through the thickness thermal conductivities of the materials used in the construction of the outer wing subcomponent. The properties estimated included the through the thickness thermal conductivity and volumetric heat capacity of the outer facesheets, the honeycomb blanket, the skins, and the torque tubes.

As for the effective one-dimensional model, the three-dimensional outer wing subcomponent needed to undergo several simplification steps before arriving at the detailed one-dimensional model. In addition to the first two simplification steps described in the previous section, a third important assumption needed to be made concerning the honeycomb closeouts and the edge channels. According to the steady-state analysis described previously in this chapter, the contributions of the various modes of heat transfer of the honeycomb closeouts and the edge channels were negligible compared to the contribution of the torque tubes and the air gap. Therefore, the honeycomb closeouts were disregarded in

the model and their small contribution willingly lumped together with the dominant properties of the honeycomb blanket. The influence of the edge channels, on the other hand, was lumped together with the impact of the air and torque tube ribs which are both very significant contributors to the heat transfer through the middle section of the outer wing subcomponent.

Two-Dimensional Numerical Model

The one-dimensional models provided us with the tools to estimate both the effective thermal conductivity through the thickness k_{eff} and effective volumetric heat capacity C_{eff} . It also allowed us to estimate the individual volumetric heat capacity along with the individual thermal conductivity through the thickness of various other materials used in the construction of the panel, described above. The remaining individual in-plane thermal conductivity of the facesheets, the honeycomb blanket, the skins, the torque tubes, and the thermal conductivity and volumetric heat capacity of the air present in the torque tube cavities, can be estimated using the two-dimensional model described below.

The two-dimensional model created to give us the ability to estimate the remaining unknown in-plane thermal conductivities of the materials used in the construction of the outer wing subcomponent, was designed with all the details and characteristic features of a cross-section of the outer wing subcomponent. Only the first step of the simplification process (i.e. no temperature change in the third dimension) described previously was used during the design of the two-dimensional model. No additional assumptions were needed to successively create the two-dimensional finite element model illustrated in Fig. 3.5.

3.1.3 Boundary Conditions

One important and often crucial issue in the design of finite element models is the choice of the applied boundary conditions. Small changes in the applied boundary conditions can induce significant changes in the obtained temperature distributions and results. The choices of the boundary conditions applied to the one-dimensional and two-dimensional finite element models and the reasoning behind them are explained below.

Since both, the volumetric heat capacity and thermal conductivity were sought, it was necessary to incorporate a heat flux boundary condition to introduce the thermal conductivity independent of the specific heat. Fourier's law was therefore used to specify

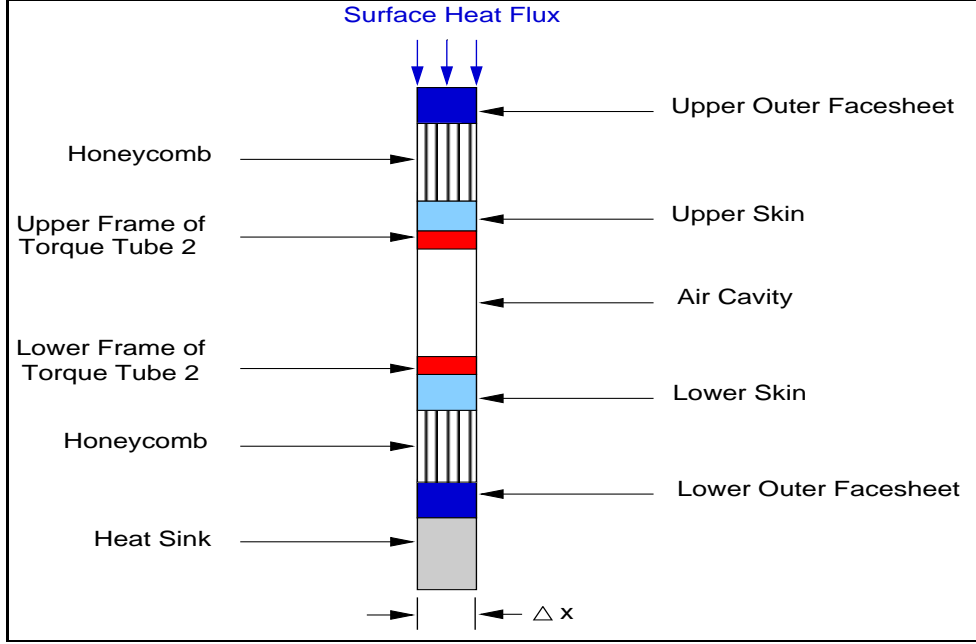


Figure 3.4 One-dimensional representation of outer wing subcomponent modeled in detail.

the boundary condition and introduce the thermal conductivity k , via a heat flux boundary condition on the top surface of the outer wing subcomponent, as shown in Eq. 3.5.

$$q'' = -k \frac{\partial T}{\partial x}. \quad (3.5)$$

The constant heat flux boundary condition was applied to the top surface of the outer wing subcomponent using different flexible silicone heaters described in more detail in Chapter 4. Since the two one-dimensional models require a constant heat flux applied over the entire top surface of the outer wing subcomponent to mimic the one-dimensional effect, the fullsize flexible silicone heater was used to cover the top surface of the outer wing subcomponent in its entirety, as illustrated in Fig. 3.6. In the two-dimensional model, lateral heat transfer needed to be introduced, and therefore the halfsize flexible silicone heater was used to cover exactly half of the outer wing subcomponent's top surface's area, as shown in Fig. 3.7. The amount of heat applied to the outer wing subcomponent was chosen to allow for a maximum temperature rise of about 4 or 5°C to exclude the need to use temperature dependent thermal properties in the one-dimensional and two-dimensional models. The optimal heating times for the one-dimensional and two-dimensional experimental setups were obtained through optimization to maximize the amount of information on the sought

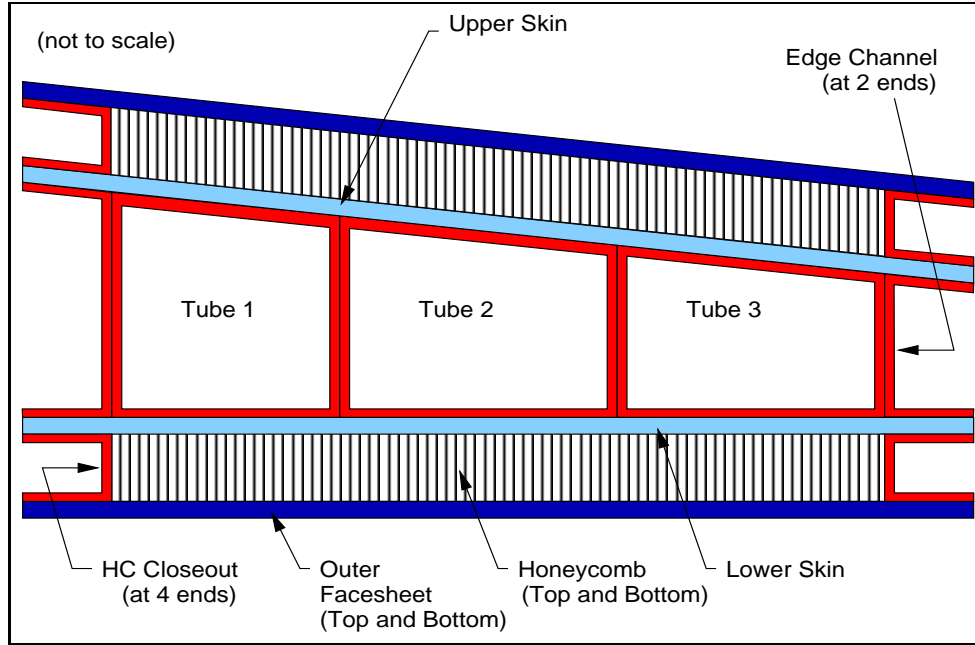


Figure 3.5 Two-dimensional representation of the outer wing subcomponent.

parameters.

The bottom surface of the outer wing subcomponent was exposed to the aluminum heat sink also described in more detail in Chapter 4, which implemented the constant temperature boundary condition for both the one-dimensional and two-dimensional models. Since it is very difficult to obtain a perfectly constant temperature boundary condition, several thermocouples were placed on the bottom surface of the outer wing subcomponent, and the temperature history recorded and read into the finite element model as a known temperature boundary condition. The constant temperature boundary condition was necessary to obtain a temperature gradient across the outer wing subcomponent that would become constant once steady-state was reached, and therefore obtain the maximum amount of information about the thermal conductivity k to be estimated.

The remaining sides of the outer wing subcomponent were insulated to minimize the amount of heat lost to the surroundings, and channel the heat into instead of away from the structure. The insulation also limited the unwanted lateral heat transfer effects within the material that might have affected the results.

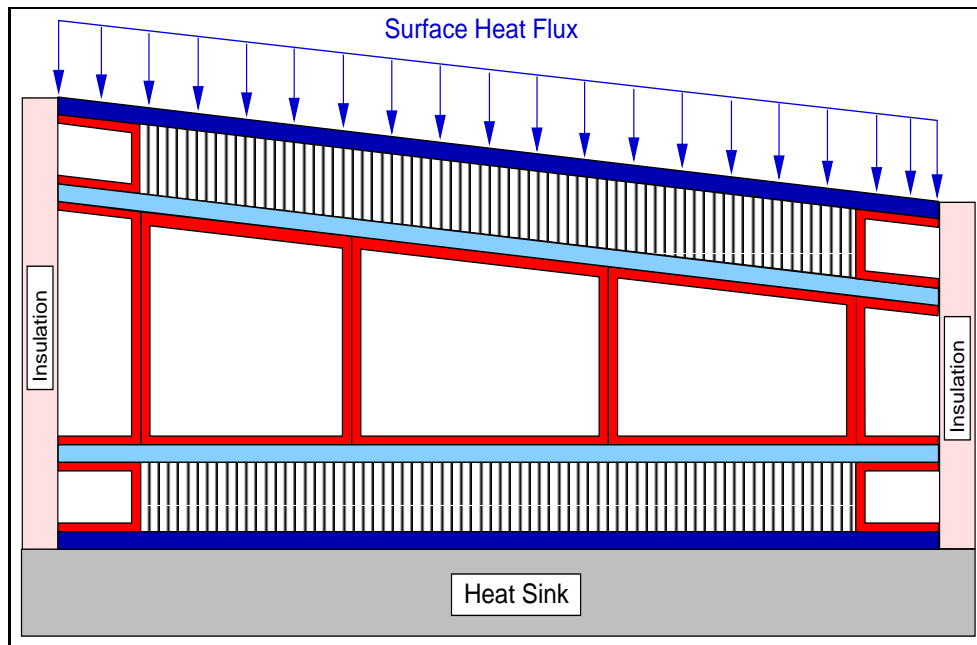


Figure 3.6 One-dimensional boundary conditions.

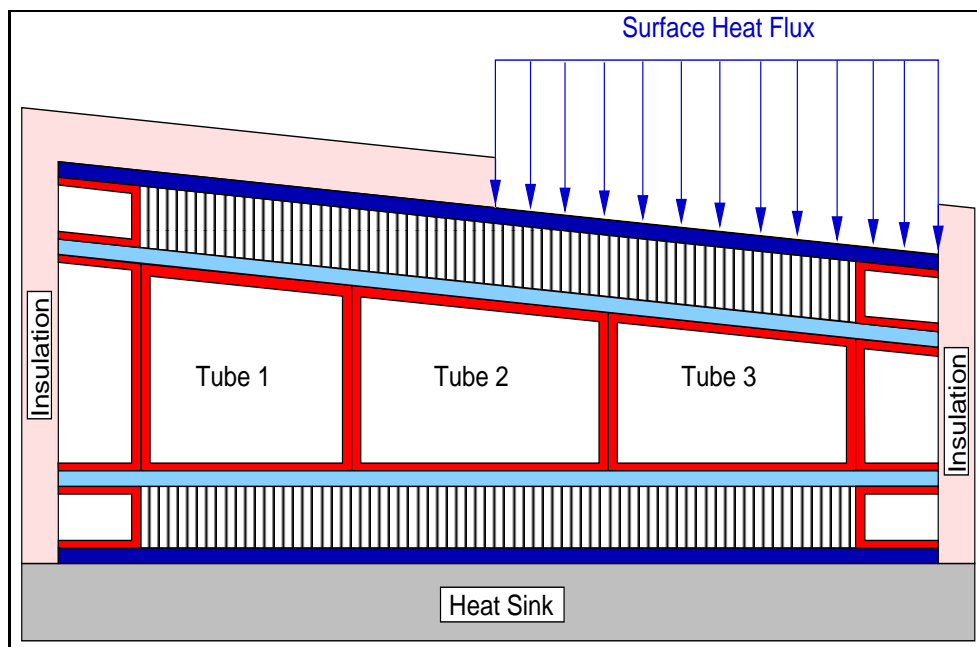


Figure 3.7 Two-dimensional boundary conditions.

3.2 Optimization and Estimation Techniques

Optimization of experiments has become very popular among experimentalists, not only because of its money and time saving abilities, but also because of the quality and usefulness of the data resulting from an optimized experiment. In the parameter estimation process, optimization of experiments is an important step for increasing the accuracy of the estimated properties. The sensitivity coefficients and confidence intervals are both important concepts in the process of estimating parameters, and are described in detail in the following subsections.

3.2.1 Sensitivity Coefficients and Correlation

Sensitivity coefficients are an important issue in parameter estimation, and should be one of the first concerns when dealing with parameter estimation problems. The sensitivity coefficients are defined as the the magnitude of change of the response η due to perturbations in the values of the parameters Beck and Arnold (1977). Eq. 3.6 gives a mathematical interpretation of the concept of sensitivity coefficients.

$$X_i = \frac{\partial \eta}{\partial \beta_i}. \quad (3.6)$$

In this study, the response η represents the temperature distribution of the outer wing subcomponent and β_i is the i^{th} parameter estimated. The objective of the optimization procedure is to determine the optimal point in the design space, for which the sensitivity coefficients are maximized. When only one parameter is to be optimized, the optimization procedure will immediately determine the design point defined by the maximum of the sensitivity history. When two or more parameters are sought, the optimization becomes more complex and compromises between sensitivity coefficients are inevitable. A more detailed explanation of the optimization procedure can be found in the next section.

One important point that needs to be addressed when estimating several parameters simultaneously, is the linear dependency of two or more parameters. If parameters are linearly dependent, they all have the same impact on the response η , when perturbed. As a result, perfectly correlated parameters can not be estimated simultaneously due to the non-uniqueness of the solution. There are many situations, however, when the parameters are

nearly, but not perfectly correlated. In this case three approaches can be used to overcome the correlation obstacle.

- Nearly correlated parameters may set limitations to conventional gradient based estimation methods, depending on the level of correlation. Non-derivative methods such as Genetic Algorithms proved to be very efficient in such situations.
- Simplifying the model by reducing the number of estimated parameters may be a second option.
- The last resource to surmount the correlation, is to change the model. For example, if both the thermal conductivity k and volumetric heat capacity C are sought, a heat flux boundary condition (Eq. 3.7) can be added, introducing the thermal conductivity k independent of the volumetric heat capacity C .

$$-k \frac{\partial T}{\partial x} = q'' \quad (3.7)$$

Eq. 3.6 shows that the sensitivity coefficients are only dependent on the temperature history of the sample, and therefore, indirectly, only dependent on time and space. Thus, for initial surrounding temperatures of 295K, 375K, and 445K, the sensitivity coefficients will have the same exact graphical representation as long as the initial temperature distribution of the sample is uniform and the thermal properties, with the radiation effects fused together with the conduction properties, are assumed constant over the temperature range explored.

3.2.2 Confidence Intervals

Confidence intervals of estimated parameters are used to determine the accuracy of the estimates. For a given experiment, the confidence interval of the estimate can be calculated using Eq. 3.8:

$$\beta_k = b_k \pm \frac{[P_{kk}]^{\frac{1}{2}} t_{1-\alpha/2(N-p)}}{\sqrt{N-1}}, \quad (3.8)$$

where β is the estimated property, b_k is an estimate for β , P_{kk} represents the k^{th} diagonal term of the \mathbf{P} -matrix, p is the number of thermal properties estimated, t is the student t-distribution for $N - p$ degrees of freedom, N is the number of observations, and $\alpha/2$ is

the confidence range. The confidence interval of a particular property is directly related to its sensitivity coefficient through the \mathbf{P} -matrix, since $\mathbf{P} = \mathbf{X}^T \mathbf{X}^{-1}$, where \mathbf{X} is a vector of the sensitivity coefficients defined by Eq. 3.6.

In the case of multiple experiments, the confidence interval of the mean value of the estimates can be calculated using, either Eq. 3.9 or Eq. 3.10:

$$\bar{b}_k \pm \frac{t_{1-\alpha/2(n-p)} \frac{\sum_{i=1}^n (N_i-1)s_i}{(N_1-1)(N_2-2)\dots(N_n-1)}}{\sqrt{n}}, \quad (3.9)$$

$$\bar{b}_k \pm \frac{t_{1-\alpha/2(n-p)} s}{\sqrt{n}}, \quad (3.10)$$

where, \bar{b}_k is the mean value of the estimates, n is the number of experiments, t is the student t-distribution for $n-p$ degrees of freedom, $\alpha/2$ is the confidence range, and s_i is the sample standard deviation of each individual experiment. Eq. 3.9 give more accurate results than Eq. 3.10, since it takes in consideration the sample standard deviation s_i of each individual experiment compared to a constant standard deviation s in Eq. 3.10.

The standard deviation of an experiment can be calculated using the following equation:

$$s = \frac{\sqrt{P_{kk}} t_{1-\alpha/2(n-p)}}{\sqrt{N}}, \quad (3.11)$$

where P_{kk} is the diagonal term of the \mathbf{P} matrix.

3.2.3 Design Optimization

Optimizing an experiment with respect to specific parameters translates into designing an experiment that will give the maximum amount of information about the optimizing parameters. Many optimization schemes use an objective function and an extremization process. An optimized experiment will yield more accurate results with much smaller confidence intervals for the estimated parameters.

Several different extremization criteria can be applied to obtain an objective function that yields the optimal parameter values, and therefore the optimal design. The three most common criteria are the D-optimal, E-optimal, and A-optimal, described below:

- The D-optimal criterion maximizes the determinant of the $\mathbf{X}^T\mathbf{X}$ matrix,
- The E-optimal criterion maximizes the minimum eigenvalue of the $\mathbf{X}^T\mathbf{X}$ matrix,
- The A-optimal criterion maximizes the trace of the $\mathbf{X}^T\mathbf{X}$ matrix.

In this study the D-optimal criterion was used to define the objective function for the optimization of experiments. the objective function is therefore defined as the determinant of the $\mathbf{X}^T\mathbf{X}$ matrix, where \mathbf{X} represents the sensitivity matrix described in Eq. 3.6. The parameters to be optimized are not explicitly visible in the formulation of the objective function, but affect it through the sensitivity coefficient matrix \mathbf{X} . In Beck and Arnold (1977), it is recommended to use the D-optimal criterion because it is equivalent to minimizing the hypervolume of the confidence region. In order to apply this criterion, the following assumptions need to be satisfied:

- Additive errors in \mathbf{Y}
- Zero mean
- Constant variance
- Uncorrelated errors in \mathbf{Y}
- Normally distributed errors in \mathbf{Y}
- Statistically known independent variables
- Errorless independent variable
- Constant β with no prior information

where \mathbf{Y} is the measured temperature vector, and β is the estimate of the property.

In order to obtain the most accurate temperature readings, the optimization scheme was applied to finding the optimal sensor locations and optimal heating times for the one-dimensional effective, one-dimensional detailed, and two-dimensional models. The Genetic Algorithm combined with the finite element package EAL were used to perform the optimization. Notice that in this case, the Genetic Algorithm was set to rank the objective functions in descending order, since the chromosomes with the highest objective functions are seeked. Details of the Genetic Algorithm are described in later in this section.

3.2.4 Estimation Procedure

Similar to the optimization procedure, the estimation process requires an objective function to compare the performance between different designs. In this study, the objective function was chosen as the sum of squares function defined by

$$S = [\mathbf{Y} - \eta(\beta)]^T [\mathbf{Y} - \eta(\beta)], \quad (3.12)$$

for the one-dimensional effective model, where \mathbf{Y} is a vector of the measured temperatures and η is vector of the calculated temperatures as a function of the unknown parameters contained in the vector β . The objective function described in Eq. 3.12 compares the temperatures at only one location on the outer wing subcomponent, where the sensitivity is maximum. For the one-dimensional detailed model, the temperature distributions at three distinct locations needed to be compared to estimate the various thermal properties. Eq. 3.12 was therefore modified by adding the sum of squares at the two additional locations on the outer wing subcomponent to reflect the discrepancies between the calculated and experimental temperature distributions. The objective function for the one-dimensional detailed model was expressed as:

$$S = [\mathbf{Y}_1 - \eta_1(\beta)]^T [\mathbf{Y}_1 - \eta_1(\beta)] + [\mathbf{Y}_2 - \eta_2(\beta)]^T [\mathbf{Y}_2 - \eta_2(\beta)] + [\mathbf{Y}_3 - \eta_3(\beta)]^T [\mathbf{Y}_3 - \eta_3(\beta)], \quad (3.13)$$

where \mathbf{Y}_i are the measured temperature vectors of three thermocouples at three different locations on the outer wing subcomponent, $\eta_i(\beta)$ is the calculated temperature vector with respect to the estimated parameter, and S is the objective function as a scalar.

In the two-dimensional estimation procedure, the temperature distribution was compared at many more locations. The objective function for the two-dimensional model needed to be redefined to accomodate the sensor locations used in the two-dimensional model.

$$S = \sum_{i=1}^n [\mathbf{Y}_i - \eta_i(\beta)]^T [\mathbf{Y}_i - \eta_i(\beta)], \quad (3.14)$$

the subscript i defines the sensor number, and n defines the total number of sensors to be looked at. The minimization procedure compares both, the measured and calculated

temperature distributions of the outer wing subcomponent, squares the difference at each timestep, and adds the squared difference together over the length of the experiment (number of timesteps). Both the calculated and measured temperature vectors need to reflect the temperature histories at the same location on the outer wing subcomponent. The minimization procedure was performed using Genetic Algorithms. A detailed description of the Genetic Algorithm and its introduction into the finite element program can be found in the following section.

3.3 The Genetic Algorithm

Design optimization and parameter estimation problems can be solved using either gradient based techniques such as the Gauss and the modified Box-Kanemasu methods, or non-derivative based methods such as Genetic Algorithms. Both the gradient and non-gradient based methods have their specific areas of application. Derivative based methods are tailored towards optimization and estimation applications with well defined objective functions and mostly uncorrelated parameters. Non-derivative based methods, on the other hand, are better suited for solving optimization and parameter estimation problems involving objective functions with many local maxima and minima where gradient methods could potentially stall, or for estimation problems where the level of correlation of the parameters is either very high or complete.

The first section gives a detailed explanation on the various steps involved in the **Extended Elitist Genetic Algorithm (EEGA)** developed by Sandrine Garcia, whereas the second section refers to the genetic processors that were written to incorporate the Genetic Algorithm based optimization and estimation scheme into the finite element programs.

3.3.1 A General Description of the Extended Elitist Genetic Algorithm

An overview of the Genetic Algorithm based optimization and estimation processes is illustrated in Fig. 3.8. Genetic Algorithms are non-derivative based optimization and estimation methods that are built upon natural selection mechanisms and the Darwinian principle of the “survival of the fittest”, Garcia and Scott (1998). The Genetic Algorithm presented below is called **EEGA**, which is the improved **Basic Elitist Genetic Algorithm (BEGA)** described in Garcia and Scott (1998).

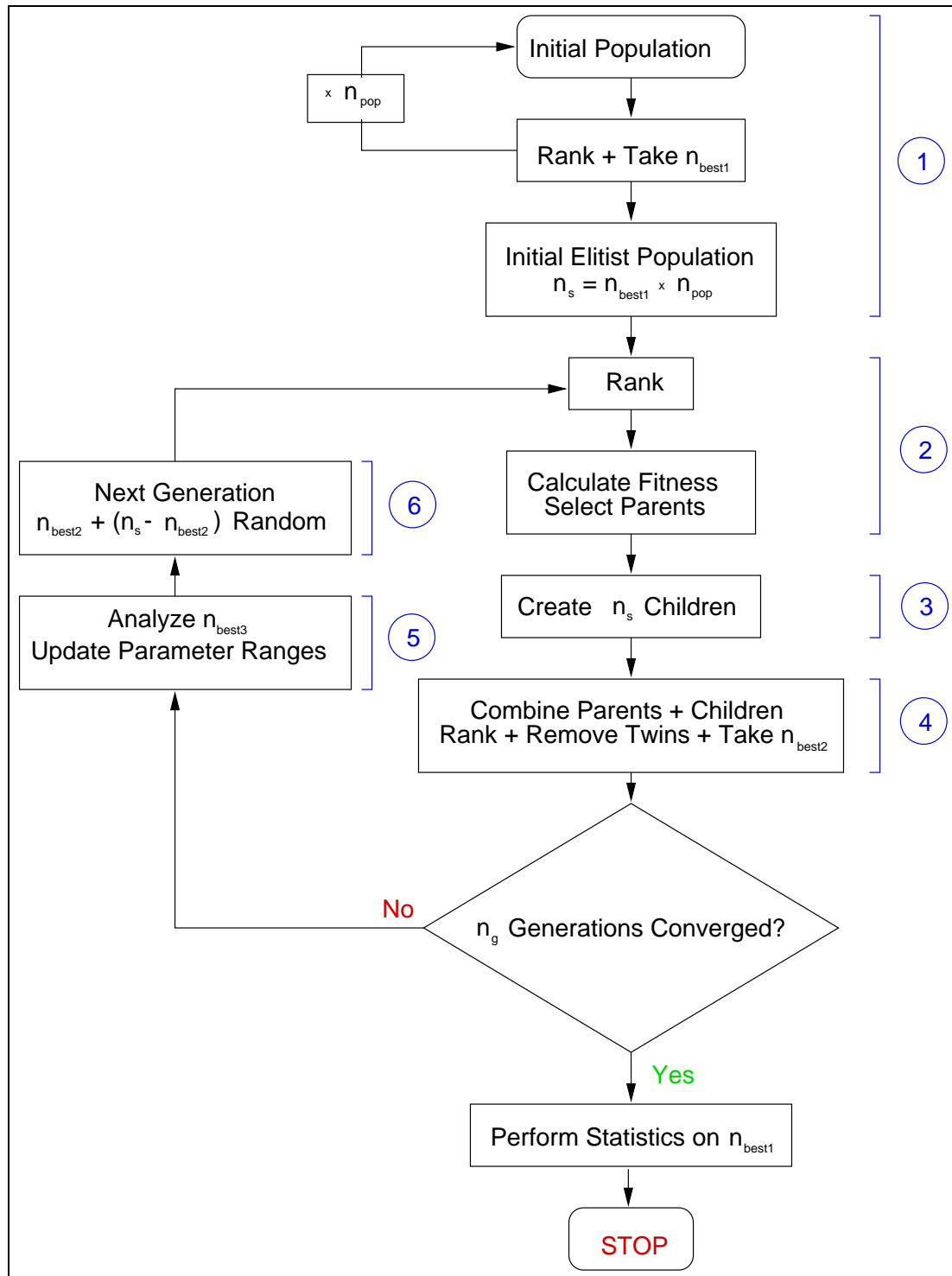


Figure 3.8 The Extended Elitist Genetic Algorithm developed by Sandrine Garcia (1996) with convergence criteria.

The first step performed by the **EEGA** is the generation of the elitist initial parent population (shown by ① in Fig. 3.8). This population is created by forming n_{pop} initial populations and taking the n_{best1} best chromosomes from each initial population and assembling the initial elitist parent population which will then consist of n_s chromosomes ($n_s = n_{pop} \times n_{best1}$). The chromosomes of each population are ranked according to the objective function they yield. Depending on the application, the objective functions are ranked either in descending order for an optimization scheme or in ascending order for an estimation program.

Each initial population's chromosomes contain n_p genes which represent the parameters to be estimated or optimized for. Furthermore, each gene of the initial populations is selected randomly within a certain initial range delimited by a maximum value and a minimum value predetermined by the user. Usually, when the approximate value of the parameter to be estimated or optimized is not known, the initial range is chosen generously to include the value of the parameter in the parameter space. The downside of increasing the initial parameter range, is the increase in computational time due to the additional number of generations required to achieve convergence. It is therefore encouraged to perform an initial parametric study to downsize the initial parameter space. Should the approximate value of the parameter be known, then the parameter space can be adjusted accordingly, cutting down the number of iterations dramatically, thus speeding up the convergence process of the parameters.

Once the elitist initial parent population is formed, the chromosomes are ranked and their fitness calculated ②. The level of fitness of the chromosomes dictates their potential to be chosen as parents. A chromosome with a high fitness level is more likely to be chosen as a parent than a chromosome with a low fitness level. Two parents are chosen simultaneously at random to undergo a single point crossover and mutations to form a child. The amount of mutation can be varied by the user by changing the p_m factor, the probability of mutation, to introduce more or less new chromosomes into the population.

Once the children population of size n_s is created ③, it is combined with the parent population to form a total population of size $2n_s$ ④. The parent-children population is ranked according to the performance of their objective functions before undergoing the twin-removal phase. Eliminating any created twins in the population represents an important step in the convergence process since it reduces the possibility of convergence on local minima. Only

the n_{best2} best chromosomes are kept from this new population, where $n_{best2} \leq n_s$. The initial parameter range is consequently updated ⑤ using the n_{best3} best chromosomes of this newly formed population, where $n_{best3} \leq n_{best2}$.

The $n_s - n_{best2}$ new chromosomes created using the updated parameter space are added to the n_{best2} chromosomes of the twin-removed parent-children population to form the new generation representing the new parent population ⑥.

The original **EEGA** developed by Garcia and Scott (1998) did not contain a convergence criterion to terminate the execution of the estimation or optimization procedure. The genetic algorithm was stopped once the user-defined maximum number of generations was achieved. In order to assure convergence and to limit the amount of generations to be performed, a convergence criterion was therefore added to the original **EEGA**. Once the new parent population is finalized, and their objective functions calculated and ranked, the average of the top 20 objective functions was recorded and compared from one iteration to another. Once the average of the 20 best objective functions of two consecutive iterations were within 1.0%, an additional iteration was run to assure that convergence was reached. If the additional iteration yielded an average of the top 20 objective functions within 1.0% of the previous iteration, the population was said to be converged. If the average of the additional iteration did not lie within 1.0% of the last iteration performed, the entire convergence process was started from the beginning and additional iterations were run.

Chapter 4

Experimental Procedures

Parameter estimation problems, also known as finite dimensional minimization problems, can be solved using the traditional nonlinear least-squares method along with an appropriate minimization procedure, Özişik (1993). Thus, the use of the least-squares method requires both calculated and experimental data. This chapter describes the experimental methods employed to gather experimental data for the estimation of the thermal properties of the outer wing subcomponent. Transient experiments, specified boundary conditions, and optimal experimental designs are essential to the simultaneous estimation of the thermal conductivity k and volumetric heat capacity C .

Due to the resources needed to build the experimental setups, all the experiments were run in the laboratories of the NASA-Langley Research Center in Hampton, VA. The first section describes the outer wing subcomponent in much detail and the second section presents the hardware used to build the experimental setups. The third section focuses on the one-dimensional and two-dimensional configurations. The fourth section explains the experimental procedures, whereas the last section concentrates on experimental data and data manipulation.

4.1 The Outerwing Subcomponent

The outer wing subcomponent was designed and fabricated in 1995 by the MDA-St. Louis High Speed Research Team. The objective of this research was to determine if it were conceivable to use unitized composite construction in the wing of the **H**igh-Speed **C**ivil **T**ransport (**HSCT**). Located at the trailing edge of the wingtip, as shown in Figs. 4.1 and

4.2, the outer wing subcomponent would be lightly loaded and not require frequent in-service access. Therefore, the structure was a perfect candidate for unitized composite construction. Fig. 4.3 shows a cross-section of the test specimen delivered to the NASA-Langley Research Center for experimentation and exploration of thermal properties in complex aerospace structures. The following section describes the different constituents incorporated in the outer wing subcomponent.

4.1.1 The Upper and Lower Outer Facesheets

The upper and lower outer facesheets, representing the outermost surfaces of the outer wing subcomponent, sandwich the remainder of the structure between them. Both, the upper and lower outer facesheets were assembled with 24 plies of IM7/PETI-5 tape prepreg with a nominal cured thickness of 0.132mm per ply. Each ply has a specific ply orientation depending on its ply number and its location within the thickness of the facesheet. Fig. 4.4 gives the overall dimensions of the facesheets. The facesheets are directly bonded onto the honeycomb structures with FM400 adhesive.

4.1.2 The Honeycomb

Honeycomb structures are well known for their high strength-to-weight ratio and are therefore often implemented in aircraft and aerospace applications. The honeycomb structures are located under the upper and lower facesheets. The honeycomb blanket is made out of stainless steel 321 foil with a nominal thickness of 0.0508mm. The honeycomb cells are vertical and measure 25.4mm in height with a diameter of 4.763mm. Fig. 4.5 shows the upper and lower honeycomb structures.

Both the upper and lower honeycomb structures have honeycomb edge closeouts which were introduced to protect the honeycomb blanket. The honeycomb edge closeouts are made of four T300/954-2A 8 harness satin cloth prepreg with a nominal cured thickness of 0.3556mm. Here again the plies have different orientations depending on their ply number and their location within the thickness of the closeout. Fig. 4.6 shows the detailed honeycomb edge closeouts.

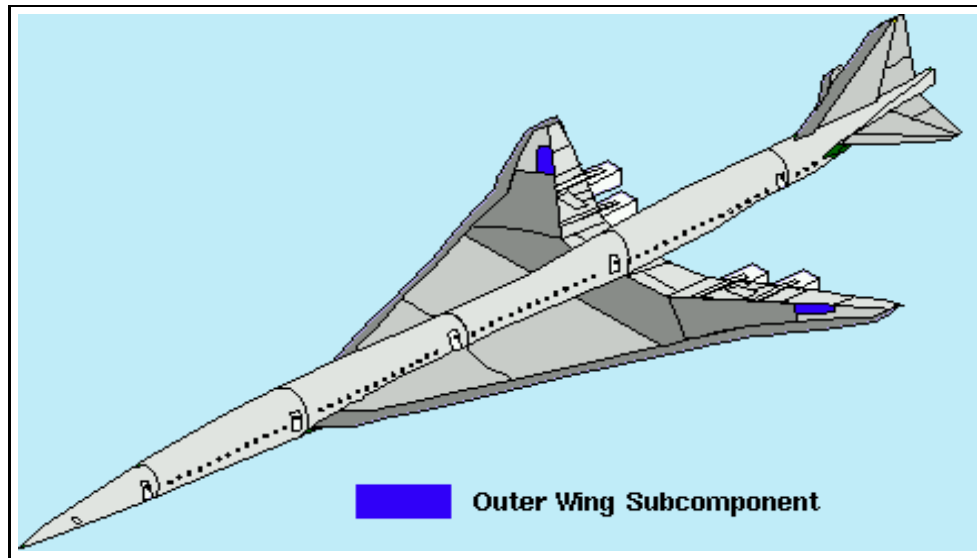


Figure 4.1 The Location of the outer wing subcomponent on the fuselage.

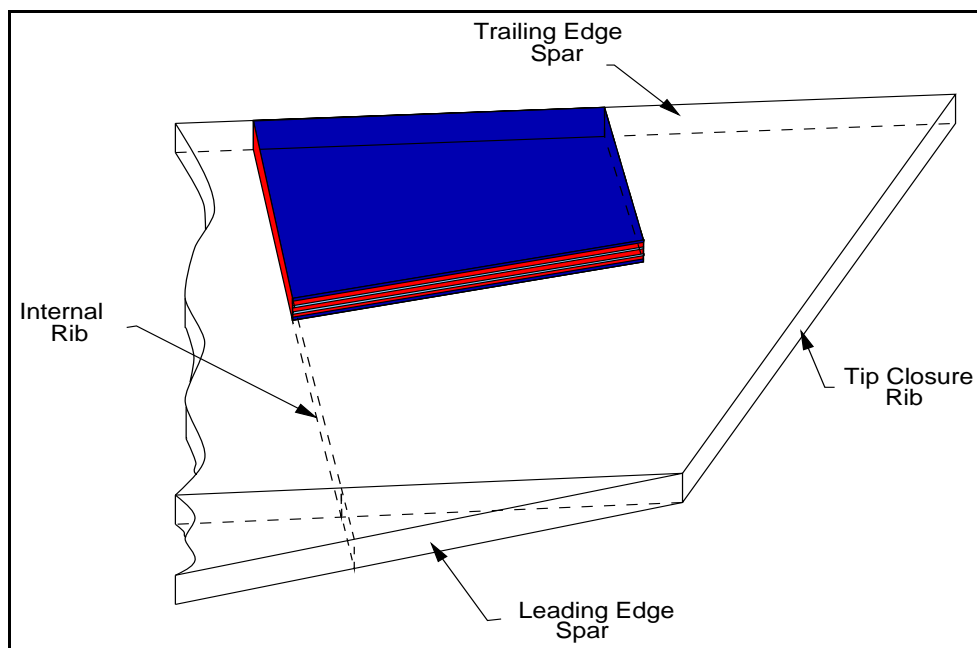


Figure 4.2 The position of the outer wing subcomponent relative to the wing tip.

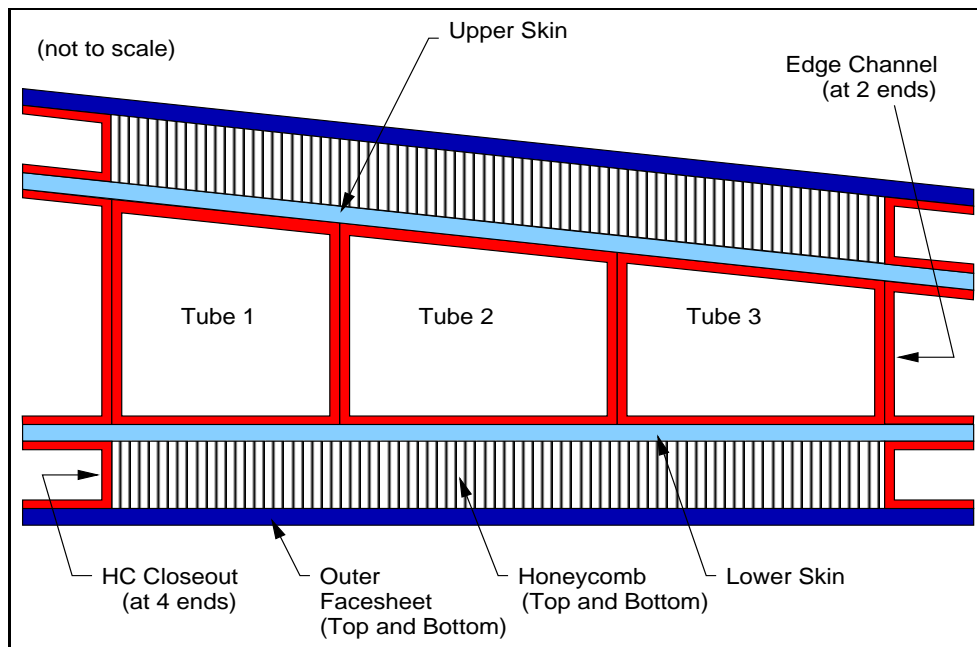


Figure 4.3 A view of the cross-section of the outer wing subcomponent.

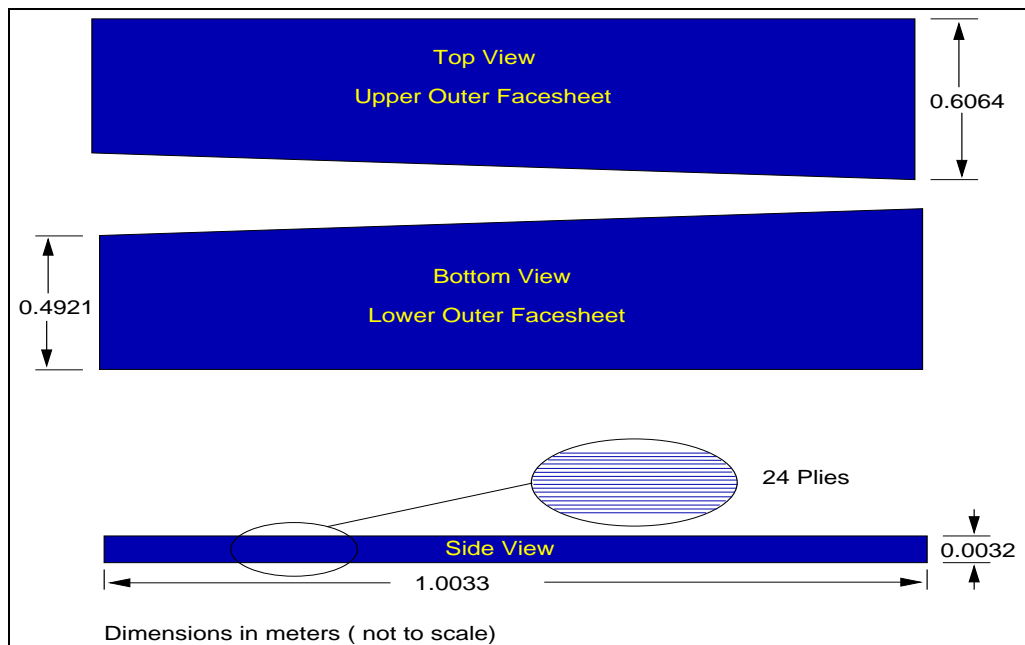


Figure 4.4 The upper and lower outer facesheets of the outer wing subcomponent.

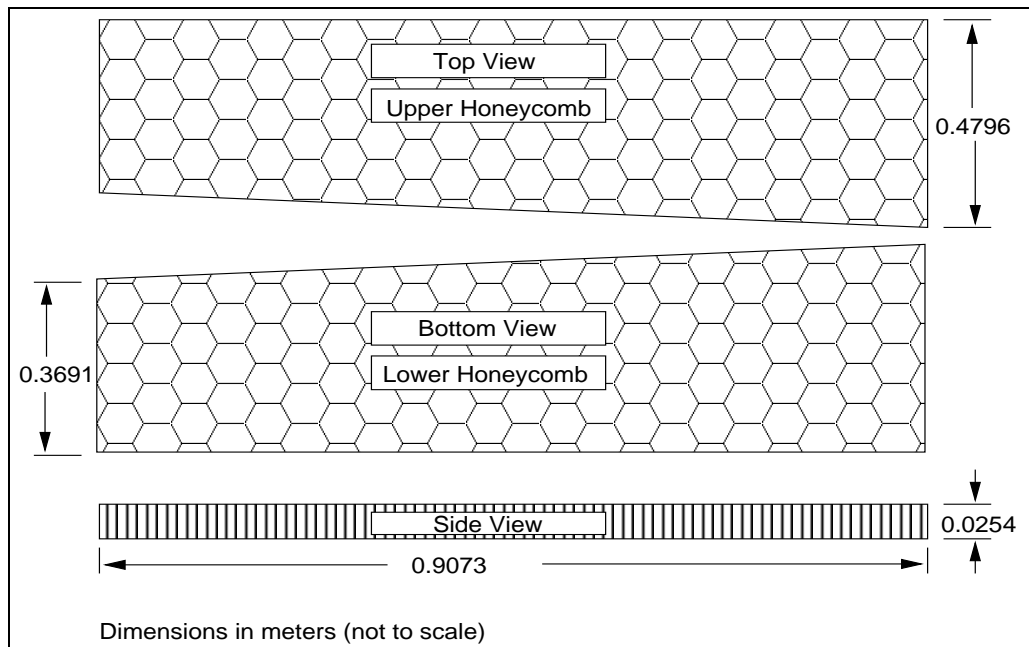


Figure 4.5 The upper and lower honeycomb structures of the outer wing sub-component.

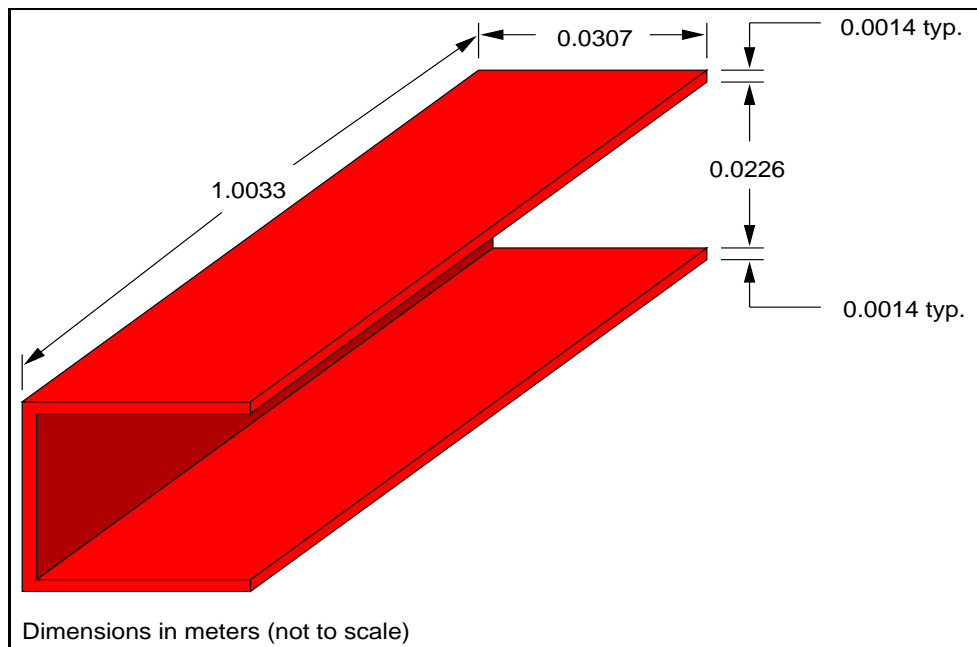


Figure 4.6 The honeycomb edge closeouts of the outer wing subcomponent.

4.1.3 The Upper and Lower Skins

The upper and lower skins are situated between the honeycomb structures and the torque tubes. The skins were fabricated with 28 plies of IM/954-2A tape prepreg with a nominal thickness of 0.2642mm. Each ply has a corresponding orientation and an end-of-ply, indicating that each ply does not cover the entire surface of the skin. Therefore, the thickness of the upper and lower skins gradually decreases when measuring from the wide end to the narrower end of the skins. Fig. 4.7 contains the particulars of the upper and lower skins.

4.1.4 The Torque Tubes

The wing of an airplane is exposed to various thermal and structural stresses that appear at different stages of a normal flight. Structural features such as torque tubes were therefore incorporated into the construction of the wing to resist various torsional forces impinging on the wing. The substructure consists of three main torque tubes which were constructed with 6 plies of T300/954-2A 8 harness satin cloth prepreg with a nominal cured thickness of 0.2642mm. The plies have orientations corresponding to their location within the thickness of the tube wall. As shown in Fig. 4.8, each torque tube's cross-sectional area increases as the torque tube lengthens. The slope of the top surface of the tubes is the same for all three torque tubes. Notice that the shapes of the torque tubes are very similar without being replicas of one another due to the differences in the dimensions.

4.1.5 The Edge Channels

The two edge channels located on either side of the three torque tubes were also made of 6 plies of T300/954-2A 8 harness satin cloth prepreg with a nominal cured thickness of 0.3556mm. The plies have different orientations depending on their ply number and their location within the channel thickness. Due to the slope of the structure, the two edge channels have very different dimensions even though they were both made using the same number of plies and manufacturing techniques. The differences are illustrated in Fig. 4.9.

The section following Section 4.2 contains details on the experimental hardware discussed in the section below, including the heater, the thermocouples, the insulation, the thermal grease, the thermotron, and the heat sink.

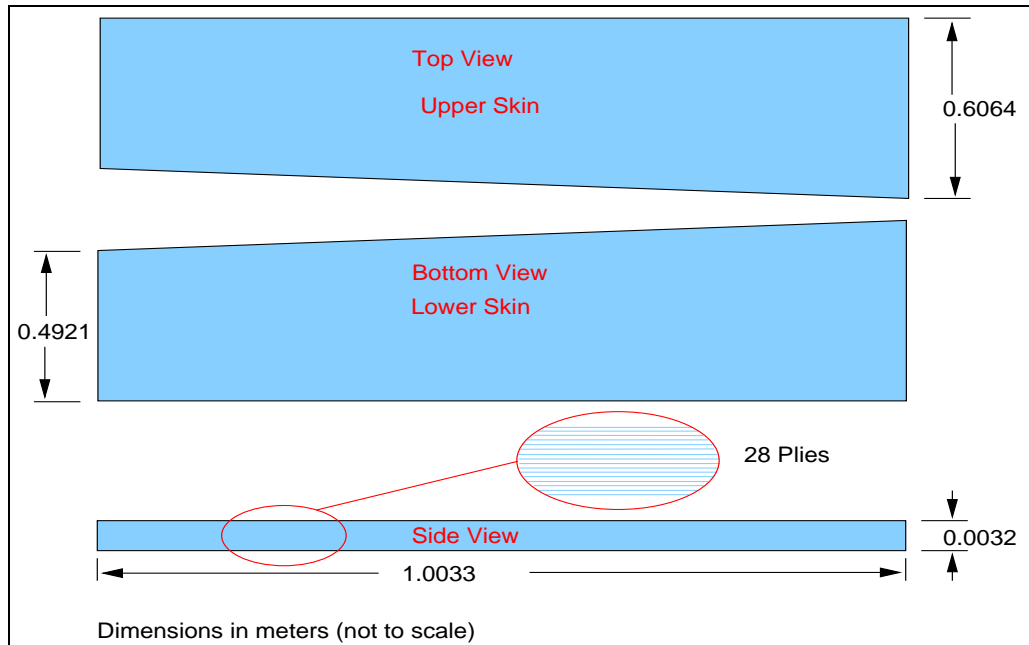


Figure 4.7 The upper and lower skins of the outer wing subcomponent.

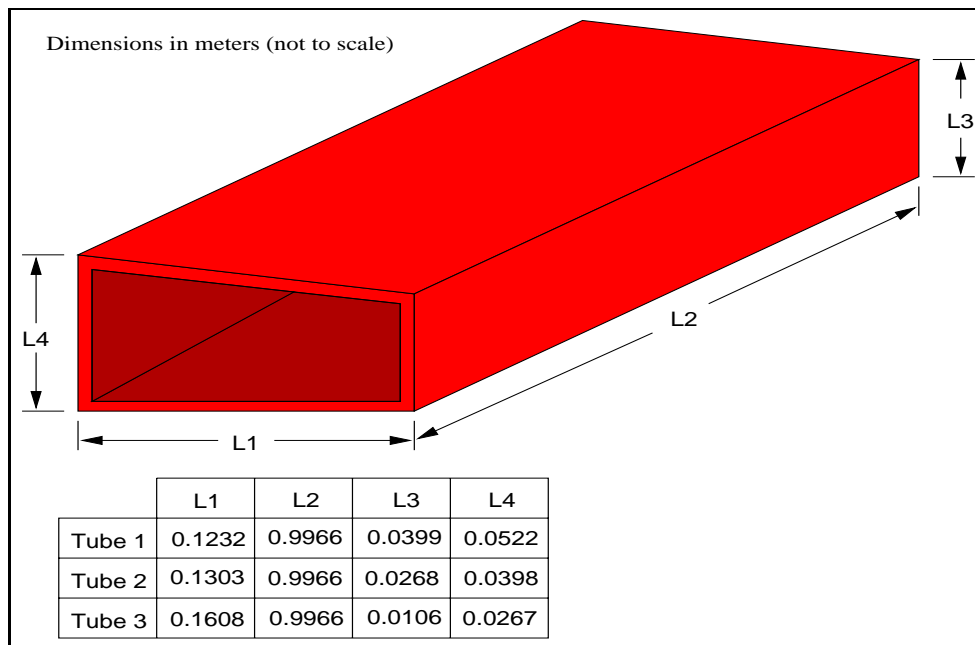


Figure 4.8 The torque tubes of the outerwing substructures.

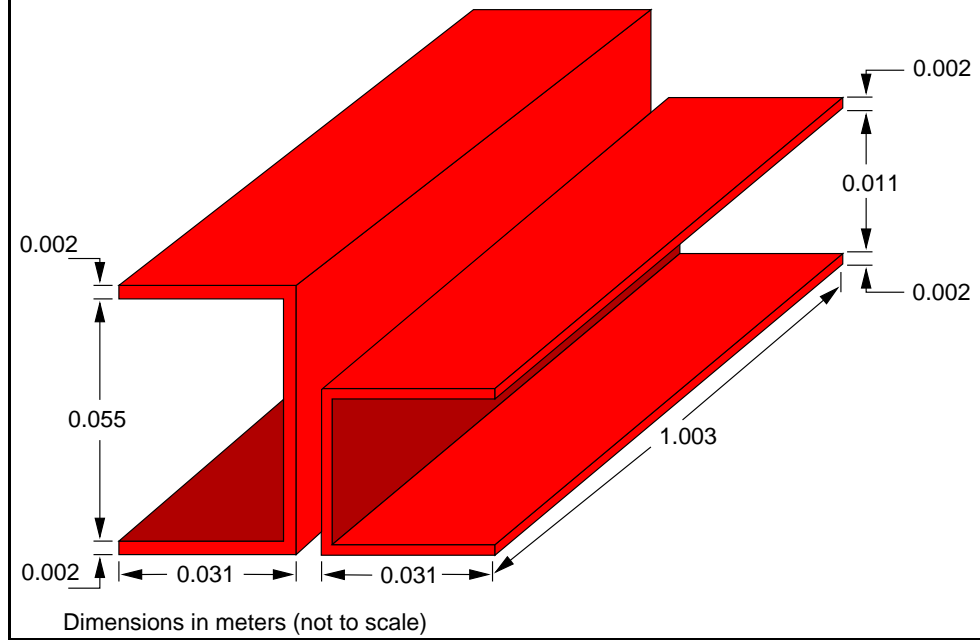


Figure 4.9 The edge channels of the outer wing substructures.

4.2 The Experimental Configurations

This section gives more detail about the two experimental setups that were used to gather data for the estimation procedure. Many of the materials and the hardware mentioned in this section will be described in more detail in the following section.

The volumetric heat capacity C of a material is non directional, whereas the thermal conductivity k is not necessarily an isotropic property. Woven materials such as carbon fiber based materials are perfect examples for possessing anisotropic properties. Therefore, two independent setups were needed to determine both the in-plane and through the thickness thermal conductivities of the materials.

The one-dimensional setup was used to estimate the unknown volumetric heat capacities and thermal conductivities through the thickness. The two-dimensional setup, on the other hand, provided the necessary information to estimate the in-plane thermal conductivities. The details of the two different setups are explained in the following subsections.

4.2.1 The One-Dimensional Experimental Setup

The one-dimensional experimental setup, shown in Fig. 4.10 was designed to transform the three-dimensional outer wing subcomponent into an experiment yielding temperature histories representative of a one-dimensional experiment. To obtain such results, it was important to link the theoretical design criteria to the experimental ones. Therefore, common experimental design practices were used to translate the theoretical boundary conditions and design considerations into experimental ones and are described below.

The applied heat flux boundary on the top surface of the outer wing subcomponent was mimicked by placing the fullsize flexible silicone heater, once bonded to a copper sheet, on the top surface of the wing panel. By covering the entire top surface of the outer wing subcomponent, no lateral heat conduction was introduced experimentally by the heat flux boundary condition.

The constant temperature boundary condition on the bottom surface of the outer wing subcomponent was implemented by the aluminum heat sink, which also provided an ideal surface on which to deposit the wing structure.

The sides of the outer wing subcomponent were carefully insulated by cutting insulation boards to the outer dimensions of the outer wing subcomponent, and placing them around the structure as tightly as possible to reduce any heat loss. The same procedure was used to place insulation boards over the flexible silicone heater on the top surface of the outer wing subcomponent to direct the majority of the heat into the structure instead of away of the structure, into the surroundings. Insulating the sides of the outer wing subcomponent is an important step to eliminate any lateral heat transfer effects that were not due to the asymmetric geometry of the structure.

The thermocouple locations were predetermined by the optimization program that not only gave the optimal heating time, but also the optimal sensor locations, where the sensitivity was maximized. Since the results of the optimization program yielded sensor locations on the heated surface, the thermocouples were placed either directly on the copper sheet carrying the silicone heater or on the top surface of the outer wing subcomponent as shown in Fig. 4.11. Since the estimation program also required the temperature history of the constant temperature surface, thermocouples were adhered on the bottom surface of the outer wing subcomponent to witness the changes of the temperature over time, as shown

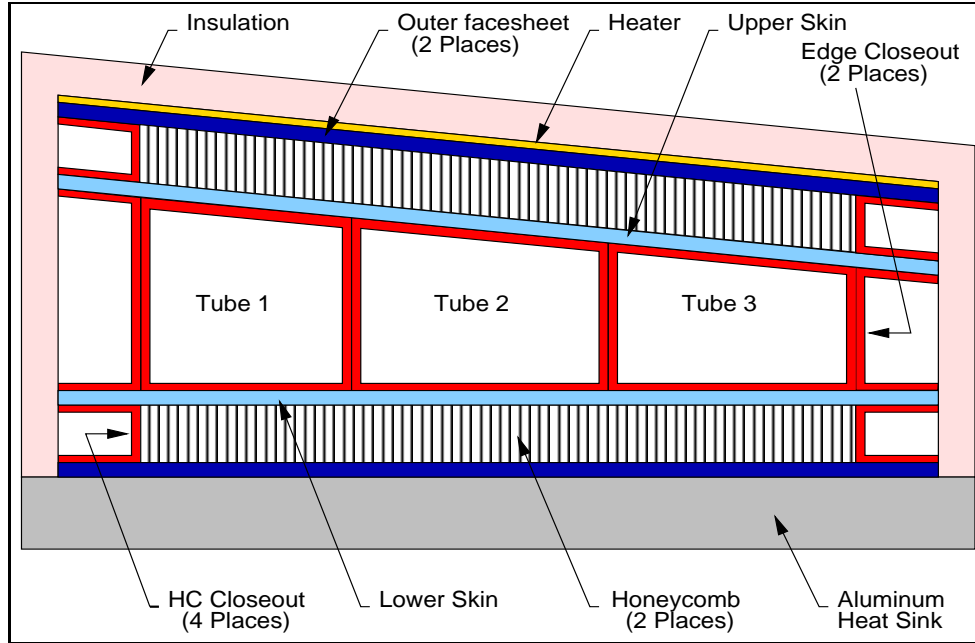


Figure 4.10 The One-dimensional setup.

in Fig. 4.11. The detailed one-dimensional numerical model also relies on temperature measurements taken at the top and bottom of the air gap located in the torque tubes. Thermocouples were bonded to the top and bottom inside surfaces of the torque tubes as shown Fig. 4.12. As mentioned earlier, all the thermocouples were installed within 0.64cm of their specified location.

4.2.2 The Two-Dimensional Experimental Setup

The two-dimensional experimental setup shown in Fig. 4.13 was designed to induce lateral heat transfer effects through the structure. As in the one-dimensional setup, the theoretical boundary conditions were translated into experimental ones. The setup of the two-dimensional experiment is very similar to the one-dimensional one, therefore the discussion in this subsection will revolve around the differences between the two.

The heat flux boundary condition at the top surface of the outer wing subcomponent was altered by removing the full-size heater and replacing it by the half-size heater which was also bonded to a copper sheet. The half-size heater covered only half the top surface of the outer wing subcomponent. This change induced the lateral conduction effects, which

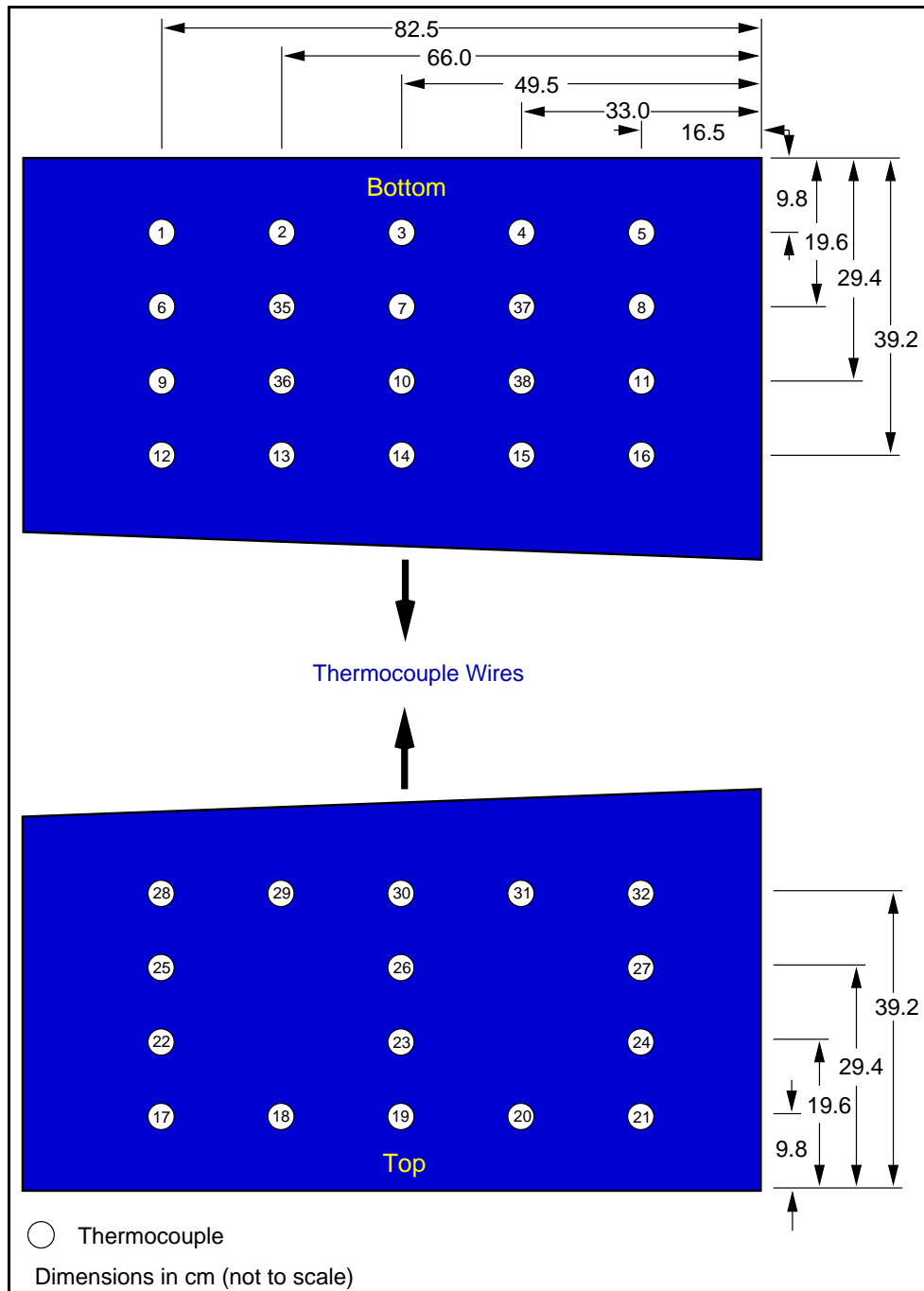


Figure 4.11 The one-dimensional top and bottom sensor layout.

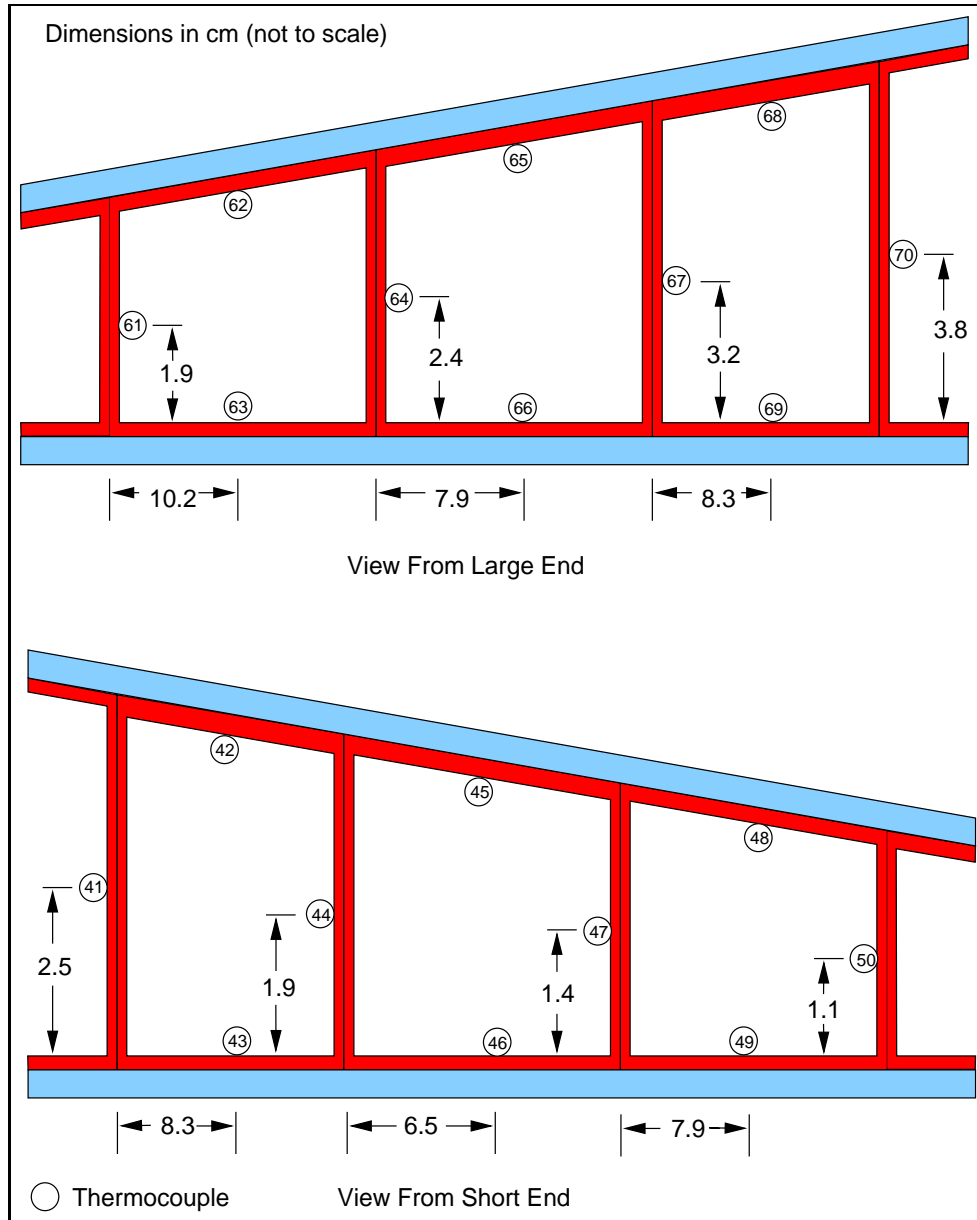


Figure 4.12 The one-dimensional inside sensor layout.

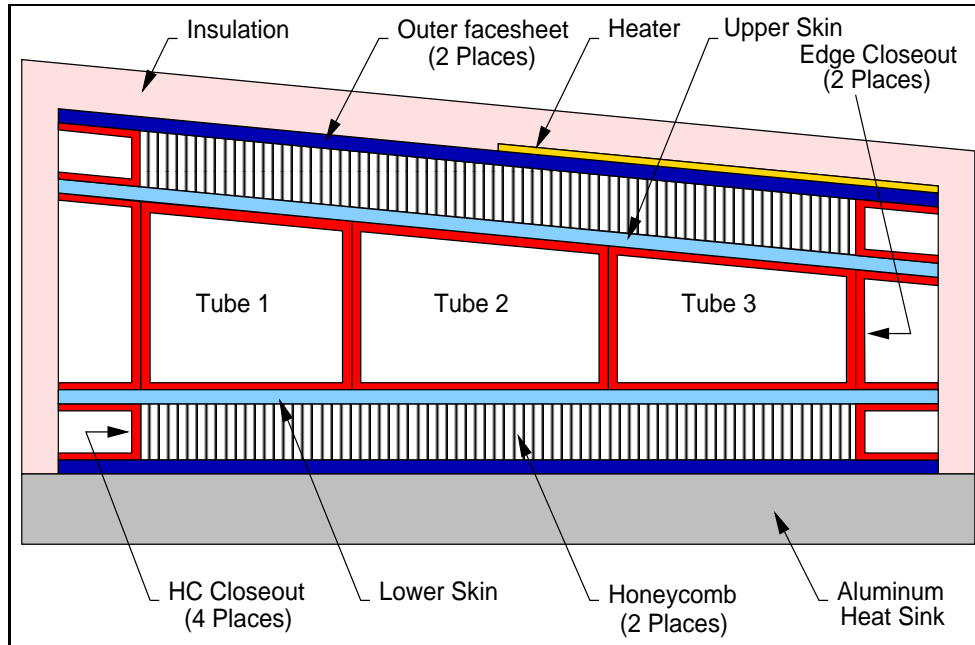


Figure 4.13 The two-dimensional setup.

were sought to estimate the in-plane thermal conductivity of the materials.

The thermocouple locations were also changed from the one-dimensional setup to be in accordance with the results obtained from the two-dimensional optimization program. The optimal sensor location was determined to be under the heater closest to the outer edge of the outer wing subcomponent. Therefore, in order to assure good temperature readings, two thermocouples were bonded right next to each other onto the surface of the outer wing subcomponent at the optimal sensor location, as shown in Fig. 4.14. Other thermocouples were evenly distributed over the top surface of the outer wing subcomponent. Similar to the one-dimensional setup, thermocouples were placed on the bottom surface of the outer wing subcomponent to provide the data to the estimation program for the known temperature boundary condition. Thermocouples were also bonded to the top and bottom inside surfaces of the torque tubes to record the temperature histories needed for the two-dimensional parameter estimation program, as shown in Fig. 4.15.

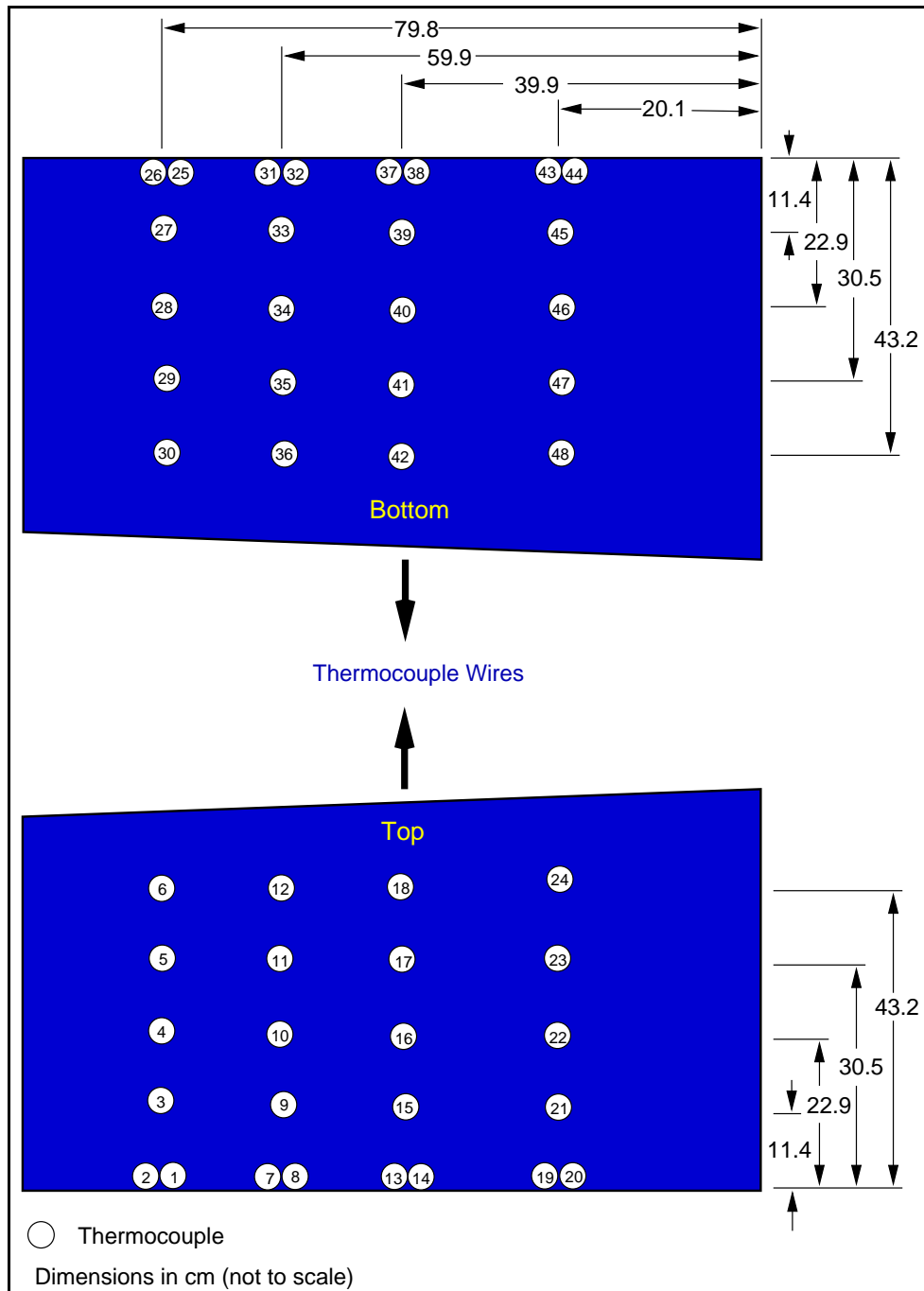


Figure 4.14 The two-dimensional top and bottom sensor layout.

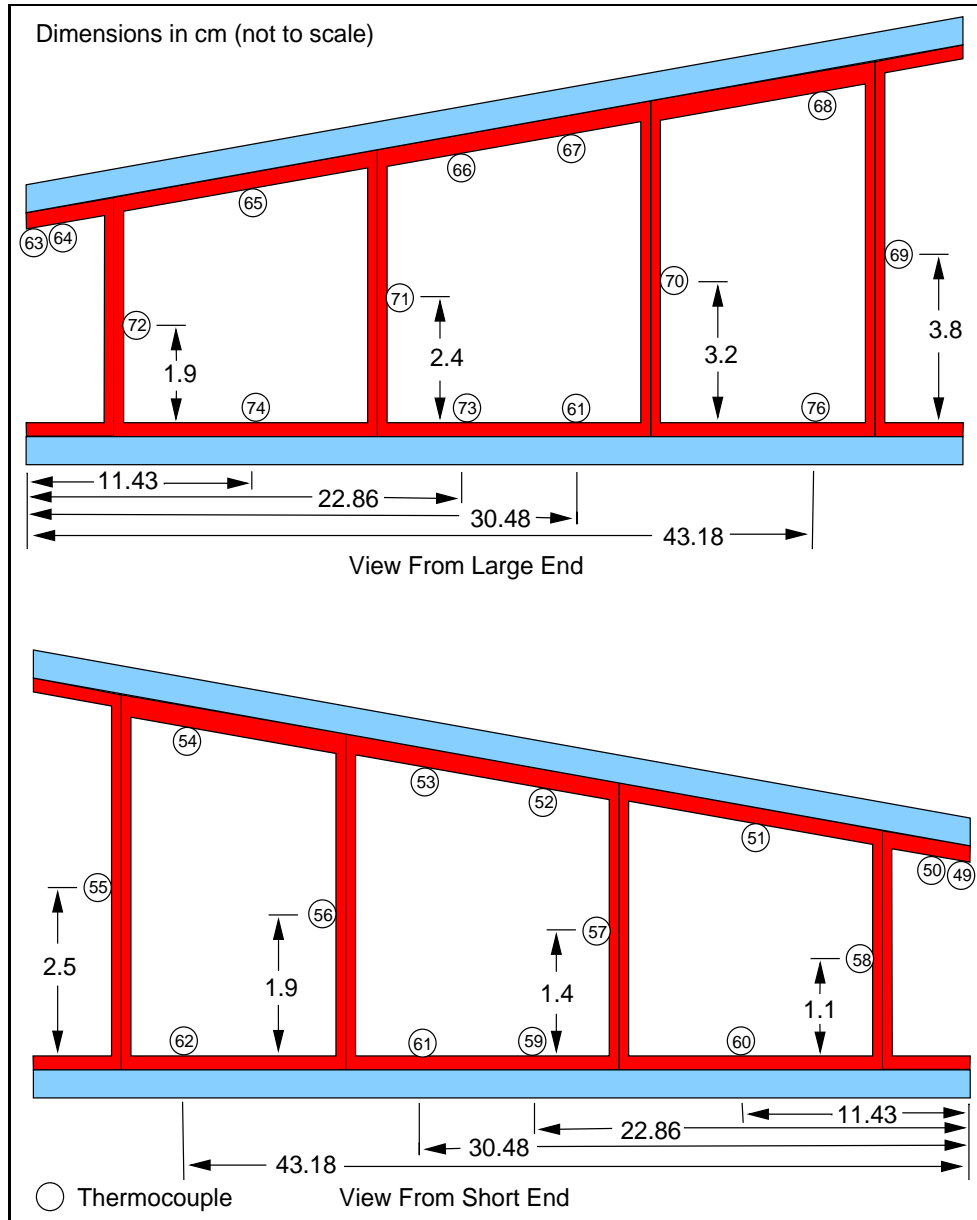


Figure 4.15 The two-dimensional inside sensor layout.

4.3 The Experimental Hardware

The purpose of this section is to describe some of the hardware used to build the test apparatus. It includes details on the various materials used to build the oven, to generate the heat flux, and to implement the insulated and constant temperature boundary conditions. An overview of the overall test setup is given in Fig. 4.16.

4.3.1 The Oven

The oven in which the experiments were conducted was designed and built at NASA Langley because the sizes of commercially available ovens were too small to accommodate the test panel. Consequently, an in-house oven was assembled to fit the sample, while leaving enough room on either side of the structure to allow for adequate air circulation, which will provide uniform and steady surrounding temperatures of 295K, 375K, and 445K to run the elevated temperature tests.

The two major components of the oven assembly were the frame and the walls. The frame was made out of U-channel Unistrut© which was cut to size and screwed together using various kinds of brackets also provided by Unistrut©. The walls that were later attached to the frame, were fabricated in two stages. First, ceramic insulation boards were wrapped in aluminum foil in order to reduce the heat loss due to radiation, especially at elevated temperatures. The wrapped insulation boards were then screwed onto plywood sheets which had previously been cut to the outer dimensions of the oven frame. Fig. 4.17 is a cross-section of the oven and reveals the oven's construction.

A sliding tray, which was to support the experiment, was added to the inside of the oven to facilitate the manipulation of the setup, which included the aluminum heat sink, the outer wing subcomponent, the insulation and the instrumentation. Two rectangular holes were finally cut into one of the wall panels to insert the air inlet and outlet from the Thermotron©, discussed later in this chapter, as shown in Fig. 4.18 below. The inlet hole was cut towards the bottom of the chamber, whereas the outlet was placed towards the top of the oven to create a flow pattern around the structure. Therefore the need for good air circulation to assure temperature uniformity throughout the oven was fulfilled.

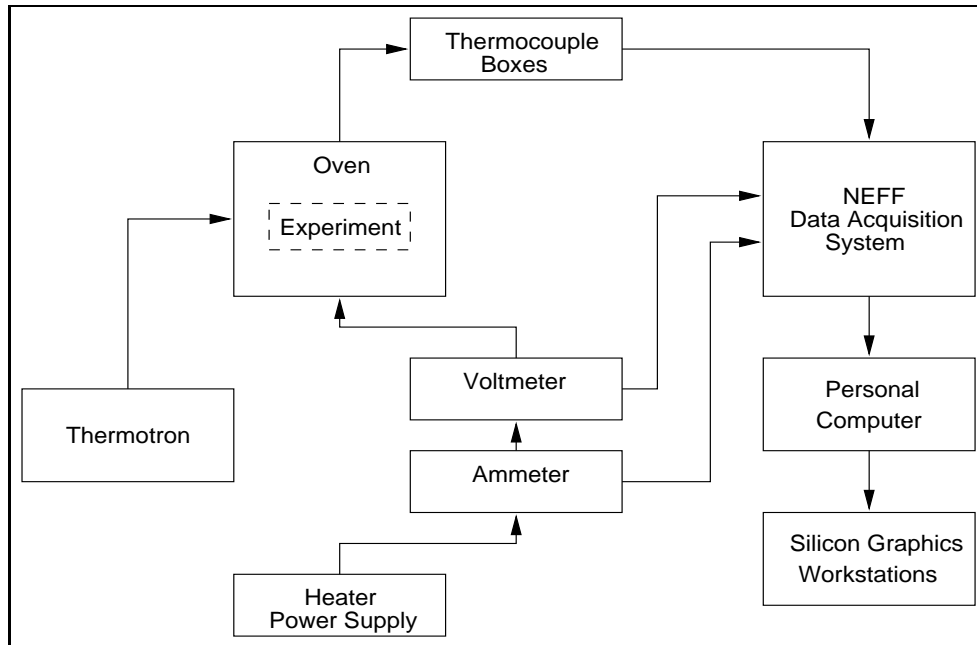


Figure 4.16 The overall test setup of the experiment.

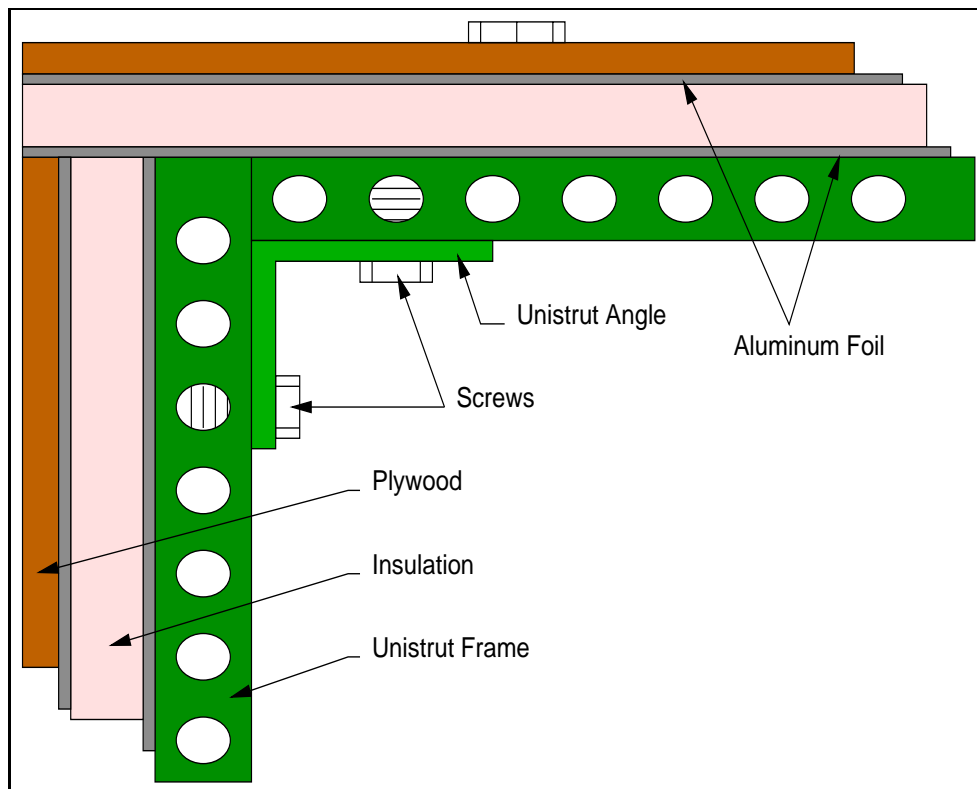


Figure 4.17 A view of the oven frame and oven walls.

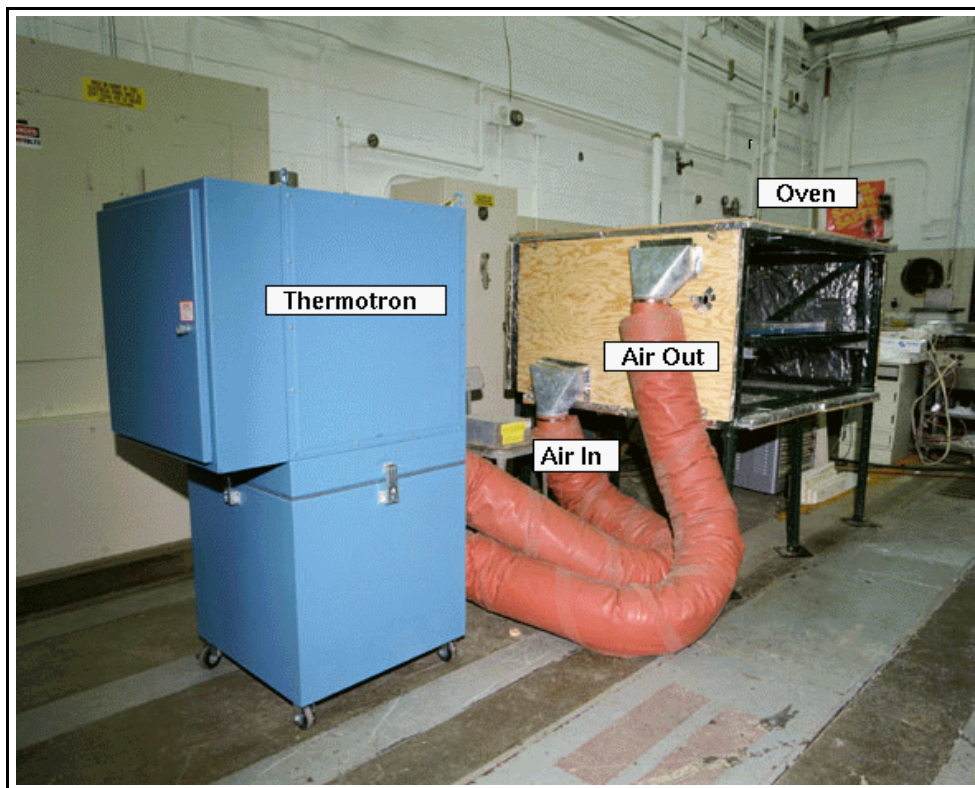


Figure 4.18 A view of the custom built oven

4.3.2 The Insulation

The insulation used to build the oven and to insulate the outer wing subcomponent was developed by Cotronics©. Their 360 Standard Ceramic Boards were recommended for applications exceeding 1250°C and rated with a thermal conductivity k of 0.0375 (W/m²K) at 260°C. Table 4.1 gives an overview of the properties of the 360 Standard Ceramic Boards. The boards are easily cut and shaped to fit various geometries and therefore obtain a good fit around the test sample, thus reducing the heat loss at the edges.

4.3.3 The Thermotron

The surrounding temperature in the oven was achieved by using the Thermotron© model number P-CH-810. The Thermotron© is a portable self-contained unit that supplies air at temperatures ranging from -74°C to 177°C, depending on the application. The micro-processor based controller, model number 4800, allows the user to set the temperature of the outcoming air, whether hot or cold. The heavy-duty blower installed in the Thermotron pushes the air through the insulated flexible duct into the test chamber and recycles it by sucking it back out through the exit duct as shown in Fig. 4.18.

4.3.4 The Electric Resistance Heaters

The temperature gradient across the outer wing subcomponent was obtained by applying a heat flux on the top surface of the structure and keeping the opposite surface at a constant temperature. A heat flux can be created using radiant heater panels, heating blankets, Kapton heaters, silicone rubber heaters and many other devices.

Table 4.1 Properties of Cotronics' 360 standard ceramic boards.

	360 Standard
Density (kg/m ³)	256.3
Specific Heat (J/kg°C)	1046.7
Thermal Conductivity (W/m ² K)	
260.0°C	0.0375
537.8°C	0.0580
815.6°C	0.0790
Thickness (m)	0.0254

The flexible silicone rubber fiberglass insulated heater was best suited to provide the specified heat flux boundary condition for several reasons. Unlike radiant heater panels, for which the surface properties of the structure determine the amount of heat absorbed, the silicone rubber heater transmits all of its generated heat to the specimen via conduction. Another positive characteristic of silicone rubber heaters is their higher maximum operating temperature (235°C), compared to Kapton heaters for which the maximum operating temperature (200°C) is set by the melting point of Kapton.

Consequently, two custom built flexible silicone rubber fiberglass insulated heaters were ordered from Watlow© to fit the exact dimensions of the top surface of the outer wing subcomponent. One of the heaters was to cover the entire surface as shown in Fig. 4.19, whereas the other was designed to cover only half the surface as shown in Fig. 4.20. Both heaters were rated at 1600 Watts.

One of the concerns of using a resistance heater was the uniformity of the heating pattern. Fig. 4.21 shows that the filaments of the heater are clearly visible through the lense of a thermal imaging system (Agema Thermovision©880), thus making the heating pattern irregular due to hot spots. The solution to this problem was to bond the heater onto a 16mm thick copper sheet, as shown in Fig. 4.23. The heating pattern was drastically improved as shown in Fig. 4.22, distributing the heat in a uniform manner over the top surface of the structure. In both Fig. 4.21 and Fig. 4.22, as we observe the color of the heater from top to bottom, we notice a darkening as we get closer to the bottom of the heater. This is due to natural convection, since the heaters were suspended vertically while tested for uniformity. Natural convection would not be an issue during experimental testing, their application being on a horizontal surface. Due to the length of the heating time of the experiments, the additional time delay caused by the capacitance of the thin copper sheet and the silicone rubber heater proved not to affect the results and were not included in the models.

4.3.5 The Heat Sink

A constant temperature boundary can be established in a number of different ways. Many of the experimental solutions to keeping a surface at a constant temperature can get fairly elaborate and expensive, depending on how much heat needs to be evacuated. The heat sink is one of the most common and inexpensive solutions to this requirement as shown in

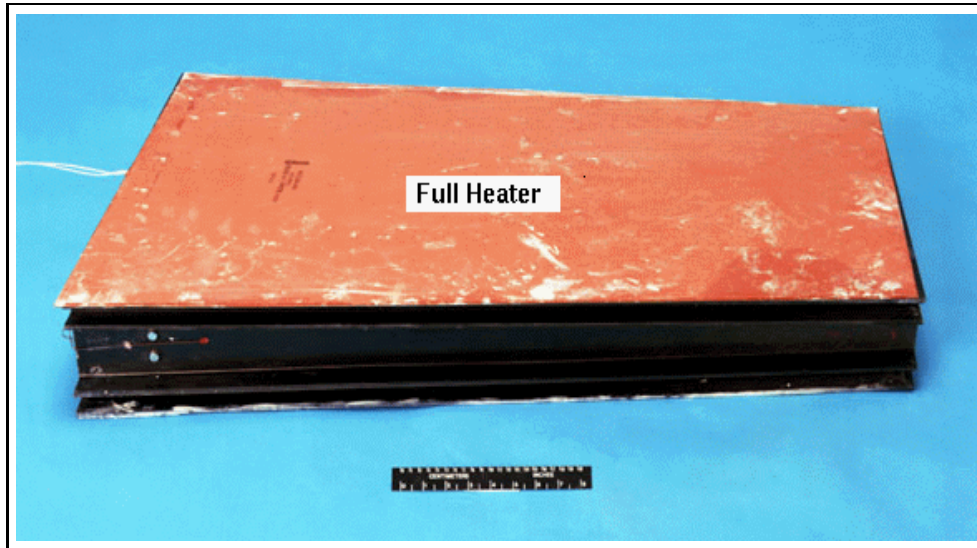


Figure 4.19 The full-size resistance heater on the top surface of the structure.



Figure 4.20 The half-size resistance heater on the top surface of the structure.

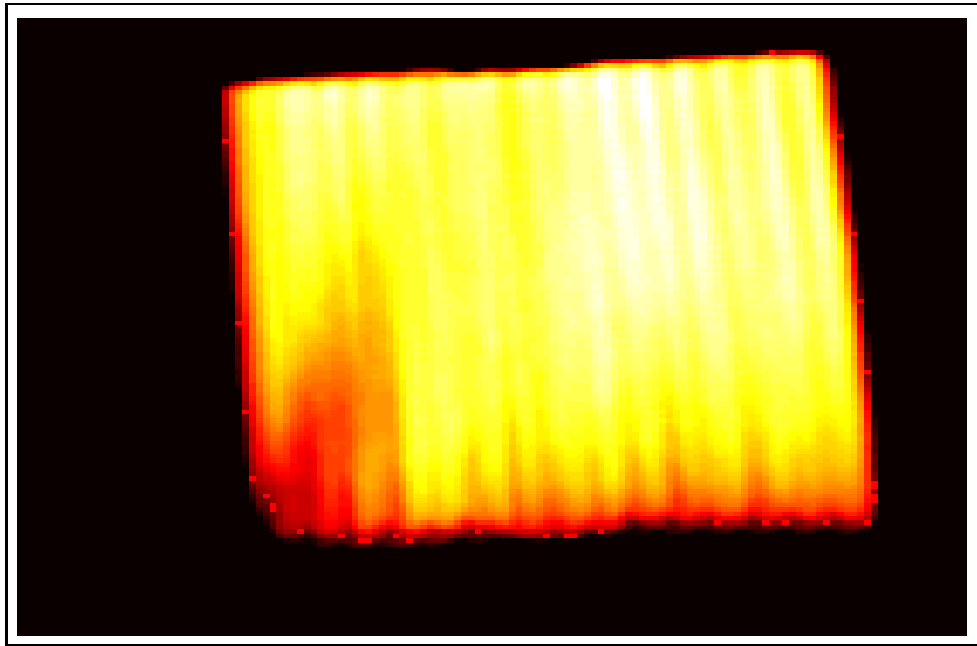


Figure 4.21 Thermal imaging of a plain resistance heater.

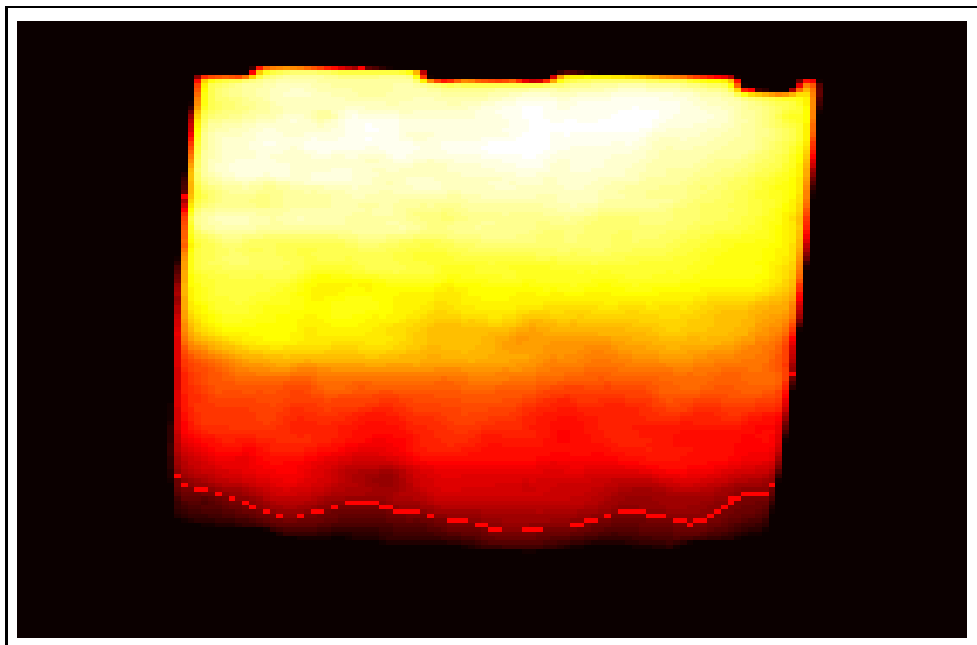


Figure 4.22 Thermal imaging of a resistance heater bonded to a copper sheet.

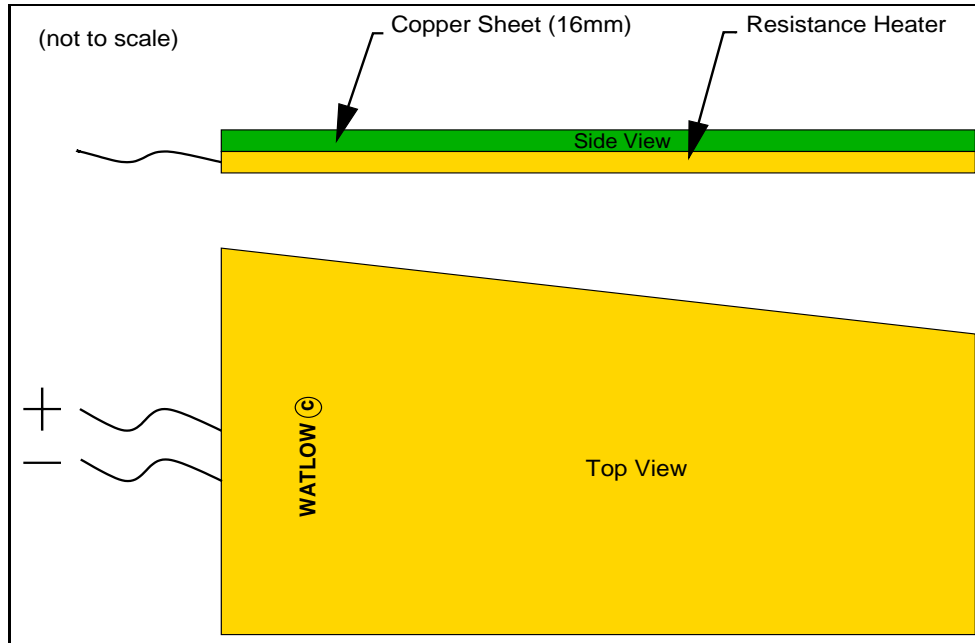


Figure 4.23 The resistance heater used to implement the heat flux.

Fig. 4.24.

The material used to fabricate the heat sink was aluminum because of its high thermal diffusivity and low cost compared to silver, gold, and copper, as shown in Table 4.2. Two piles of three 2.54cm thick aluminum plates were placed onto the tray inside the oven. The dimensions of the plates were chosen arbitrarily, yet once the specimen was set on top of the plates, a 4-5 inch space was left on either side of the structure to place the insulation.

Table 4.2 Thermal diffusivity of several materials.

	Thermal Diffusivity α (m^2/s)
Silver	174.0
Gold	127.0
Copper	117.0
Aluminum	97.1
Silicone	89.2

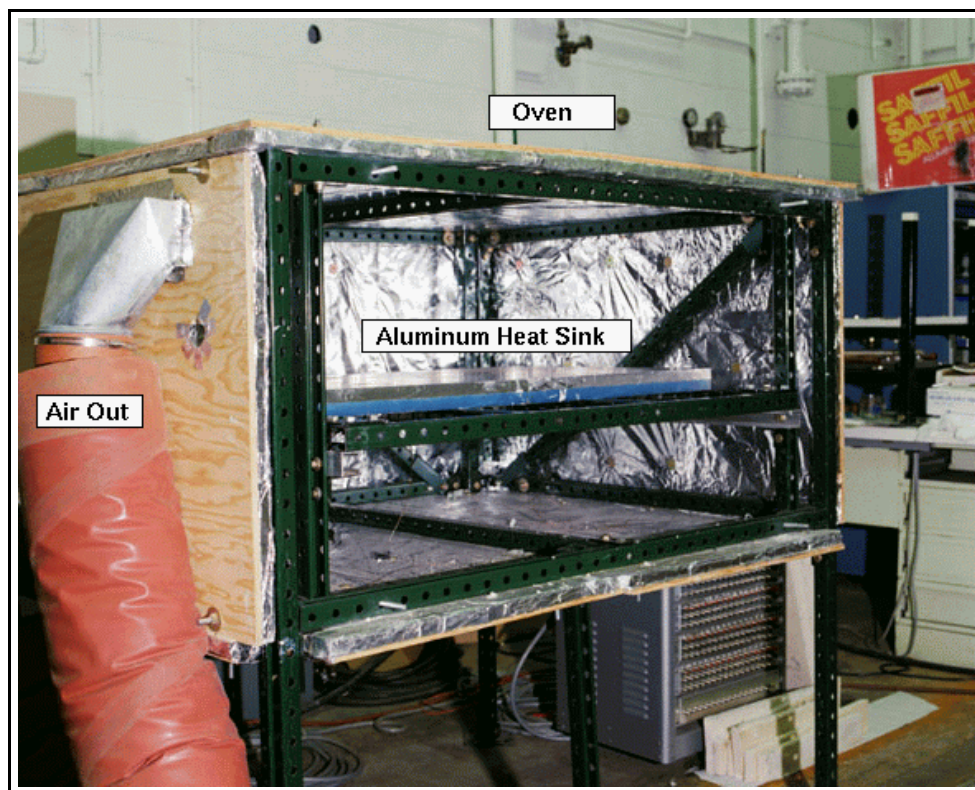


Figure 4.24 A view of the heat sink inside the oven.

4.3.6 The Thermal Grease

Thermal grease was used to minimize the contact resistance between the bottom surface of the structure and the aluminum heat sink, and between the heater and the top surface of the pannel as shown in Fig. 4.25. The thermal grease used was ordered from Dow Corning© under model number DC 340 Heat Sink Compound. This compound came as a greasy-like silicone material filled with heat-conductive metal oxides guaranteeing better thermal conductivity than air and many other materials. On the other hand, the compound was electrically insulating such that the thermocouples would not be affected by any voltages. It remained in a greasylike form even after being exposed to heat or after several weeks of testing, and was easily removed with an all purpose cleaner. The specification on the thermal conductivity k of the compound was 0.4183 (W/mK). As discussed in the experimental procedures, the thermal grease was applied sparingly and any excess removed in order to obtain a thin film and therefore reduce the thermal resistance.

4.3.7 The Temperature Sensors

The temperature was measured at various locations on the outer wing subcomponent, as described in Section 4.2. Temperature can be measured using different types of measuring devices, such as thermocouples and resistance temperature detectors (RTD's). Thermocouples were selected as the temperature measuring device of choice, for several reasons. Firstly, thermocouples measure the temperature at specific locations, which is very important for multi-dimensional analyses. Secondly, they are relatively easy to install, and finally they do not require a bridge circuit to be used.

The thermocouple family consists of many different thermocouple types, each designed for specific applications. An ANSI type K thermocouple was used for the instrumentation of the test specimen because of its wide temperature range (-200°C to 1250°C), and because of its output EMF that is relatively high over the useful temperature range (-5.973mV to 50.633mV). Type K thermocouples have a positive lead (yellow) made of Nickel-Chromium alloy and a negative lead (red) made of Nickel-Alomega alloy. The wire gauge was chosen to be AWG 30 (0.010in) to make them as transparent as possible to the experimental setup, thus reducing the contact resistance between the structure, the heat sink, and the heater.

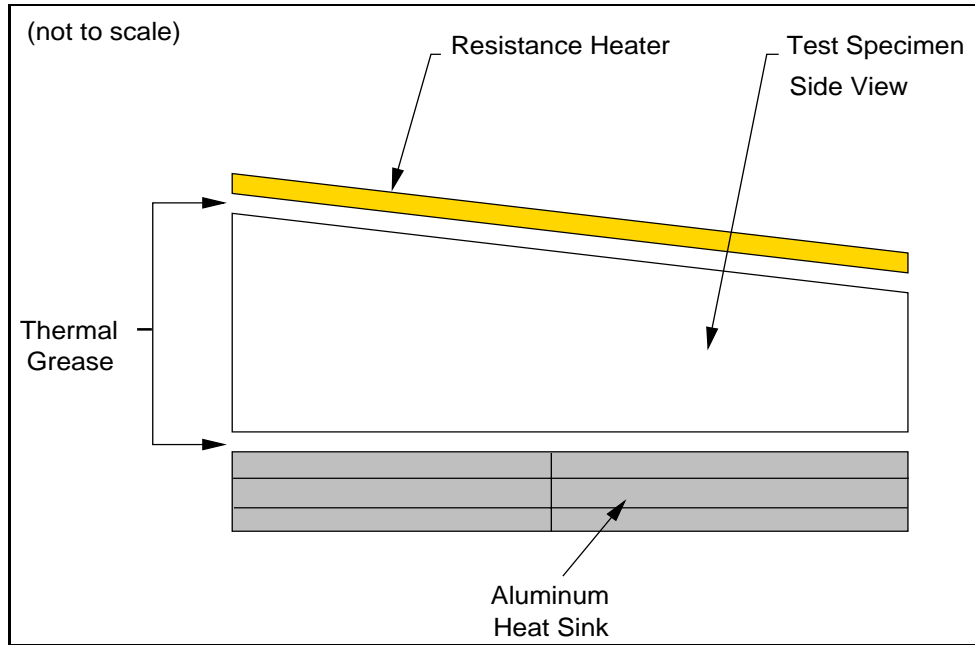


Figure 4.25 Application locations of the thermal grease.

A calibration curve for the spool of thermocouple wire was obtained and the calibration polynomial entered into the data acquisition system to adjust the temperature readings. A typical value for the accuracy of a type K thermocouple is 2.2°C or 0.75% of the measured temperature. Table 4.3 shows the temperature deviations of the thermocouple readings which range from 0.2°C to 0.7°C . The previous accuracy discrepancies affect absolute temperature measurements. However, the estimation procedure is based on relative temperature measurements, temperature differences, and therefore is not directly affected by absolute temperature inaccuracies. A more important issue regarding the temperature measurements, is the experimental temperature variation or noise which can be evaluated for each dataset, because of its direct effect on the confidence intervals of the estimates. The noise present in the temperature measurements was determined to be around 0.3°C . The calibration curve is shown in Fig. 4.26.

A total of 6 thermocouple boxes, at 15 thermocouple entries per box, were connected to the data acquisition system for a maximum of 90 thermocouple readings. The one-dimensional test apparatus contained 54 thermocouples, whereas the two-dimensional setup counted 81, all of which were bonded to the surfaces with RTV. The thermocouple locations were sandblasted for better adhesion of the bonding material. The thermocouple boxes were

in-house products, and were each equipped with their own electronic ice bath (reference junction). The thermocouples were successfully installed within 0.64cm of their specified location.

4.3.8 The Data Acquisition System

The data acquisition system used to record the data was a NEFF 470 Data Acquisition Unit with 6 NEFF 470050/90023130 16-Channel Differential MUX cards all made by NEFF Instrument Corporation©. The software installed on the Gateway Personal Computer was Autonet 5.0 by Imagination Systems, Inc.©. The voltage and current supplied to the heater were read using Keithley© multimeters, and the signal was forwarded to the NEFF data acquisition system.

The flexibility of the Autonet software allowed for data processing during recording. That feature was used to calculate the heat flux delivered by the heater, using Eq. 4.1.

$$q'' = \frac{EI}{A}, \quad (4.1)$$

where q'' is the heat flux applied to the top surface of the structure, E is the voltage applied to the heater, I is the current applied to the heater, and A is the area of the heated surface.

The thermocouple voltages were read into the data acquisition system, where the Autonet software would translate them into temperatures using the calibration polynomial of the thermocouples. Several test experiments were run at various sampling rates. After observing the rate of change of the temperature readings at the heated surface and keeping in mind the length of the experiments, the sampling rate was set to 15 seconds. At the end of each experiment, the datafiles were transferred to a Silicon Graphics Inc.© workstation where the data manipulation and estimation programs were located.

Table 4.3 Calibration values for thermocouple spool type K

Actual Temperature (°C)	Thermocouple Reading (°C)
65.56	65.83
93.33	93.78
121.11	121.72
148.89	149.50
176.67	176.89

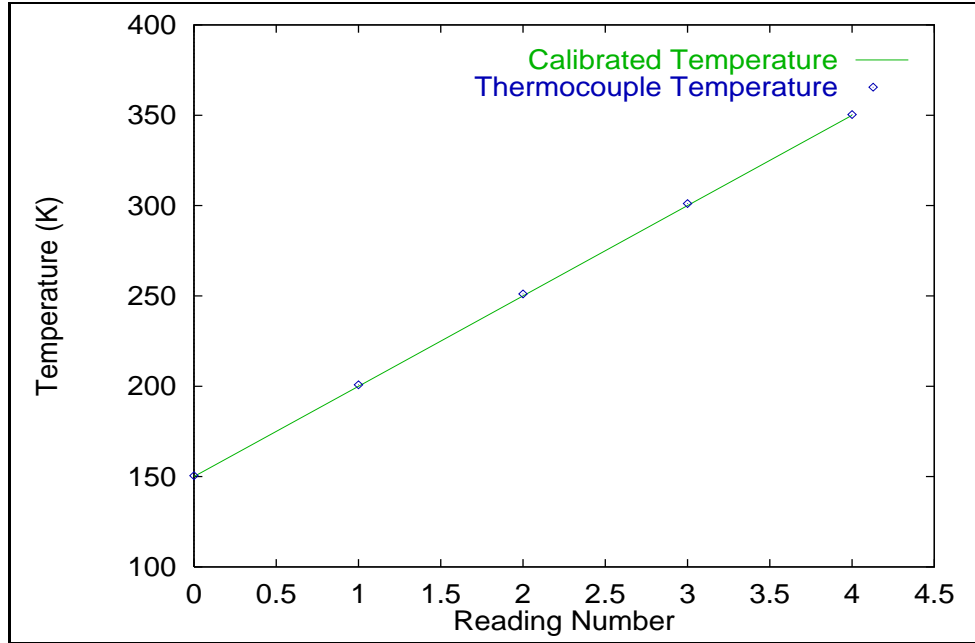


Figure 4.26 The calibration curve for the thermocouple wire.

4.4 The Experimental Procedure

The objective of the experiments was to determine the thermal properties of several materials used in the outer wing subcomponent at different temperatures. The three temperatures at which the experiments were conducted were 295K, 375K, and 445K. Temperatures above 445K were not investigated, since the structure would not be exposed to temperatures exceeding 445K during its normal life of operation.

Six experiments were run at each temperature setting for two reasons. Firstly, it was statistically more meaningful to obtain several datasets of experiments conducted under the same conditions. Secondly, it was important to assure the repeatability of the experiments.

The experimental setup consisted of the same boundary conditions for both the one- and two-dimensional test apparatus. One surface was held at a constant temperature, the opposite surface had an applied specified heat flux, and the remaining sides were insulated. Each experiment was performed using the following procedure:

1. The 6 aluminum heat sink plates were placed onto the sliding tray in the oven. Thermal grease was applied between the aluminum heat sink plates to reduce the contact resistance between the plates.

2. A thin layer of thermal grease was spread onto the top surface of the heat sink and concentrated towards the center of the heat sink. This ensured the evacuation of any remaining air trapped between the sample and the heat sink and reduced the contact resistance once the sample was pushed down onto the aluminum plates.
3. The wingbox was placed on the top surface of the heat sink and pressure was applied from the top, until the grease was squeezed out from underneath. Care was taken to minimize the thickness of the grease layer, without compromising the contact between the parts.
4. The cut insulation was placed around the test specimen, assuring a close fit to the structure.
5. A thin layer of thermal grease was spread onto the top surface of the structure, concentrating it in the middle of the surface.
6. The heater, full-heater or half-heater depending on the setup, was deposited on top of the thermal grease and pressure was applied, until the grease was squeezed out.
7. The remaining insulation was placed on top of the heater.
8. The heater and thermocouple leads were fed through an opening cut into the oven and connected to the power supply and thermocouple boxes respectively.
9. The Thermotron© controller was programmed to heat the oven to a specific temperature.
10. After making sure that the surroundings, the heat sink, the structure, and the insulation were at steady state, the data acquisition system was turned on.
11. A few datapoints were recorded before the heater was turned on, not only to be able to calculate the variance (noise) in the data, but also to calculate the offset between thermocouple channels (reference junctions).
12. Once the optimal heating time was reached, the heater was turned off. The time was kept using the computer clock of the data acquisition system.
13. The data acquisition system was stopped once the sample had cooled down and reached steady state at the initial temperature.

4.5 Experimental Data

Once the temperature data were recorded, the files needed to undergo several modifications before they could be used in the estimation program. This section describes typical datasets obtained with the experimental setups described above, and explains the steps involved in reducing the data.

As stated earlier, the four areas of interest of the outer wing subcomponent were the top outer surface (beneath the heater), the top and bottom inside surfaces of the torque tubes, and the bottom outer surface (between the outer wing subcomponent and the heat sink). Fig. 4.27 represents a plot of typical raw temperature histories obtained from thermocouples located in each area of interest.

The trends described by the temperature distributions recorded by the thermocouples located on the top outer surface and the top and bottom inside surfaces of the torque tubes are very similar. Their temperature distributions are characterized by the exponential increase and decay of the temperature, outlining two very distinct phases. The first phase, the heating phase, delineates the period between the heater turn on and the heater turn off, once the optimal heating time has been reached. The second phase, the cooling phase, starts as the heater is turned off when the temperature of the outer wing subcomponent has almost reached steady-state, and ends once the entire structure has cooled down to its initial temperature. The temperature history described by the thermocouple on the bottom outer surface of the outer wing subcomponent corresponds to the constant temperature boundary and was later used in the estimation program as the constant temperature boundary condition. The steady decline, however, does not comply with the definition of a constant temperature boundary, thus demonstrates an imperfection in the experimental setup, causing the surrounding temperature to decrease over time.

One noticeable feature of the different temperature distributions displayed in Fig. 4.27 is the considerable offset at the beginning and at the end of the experiment. This offset is due to the varying reference temperature from one thermocouple box to another and can be considered as bias error. Therefore, even though the specimen is initially at a uniform temperature, all the thermocouples do not read the exact same value. To account for the generated offset, a small FORTRAN program called `sort.new.f` was developed to bring all the temperature measurements to a common initial steady state value, within

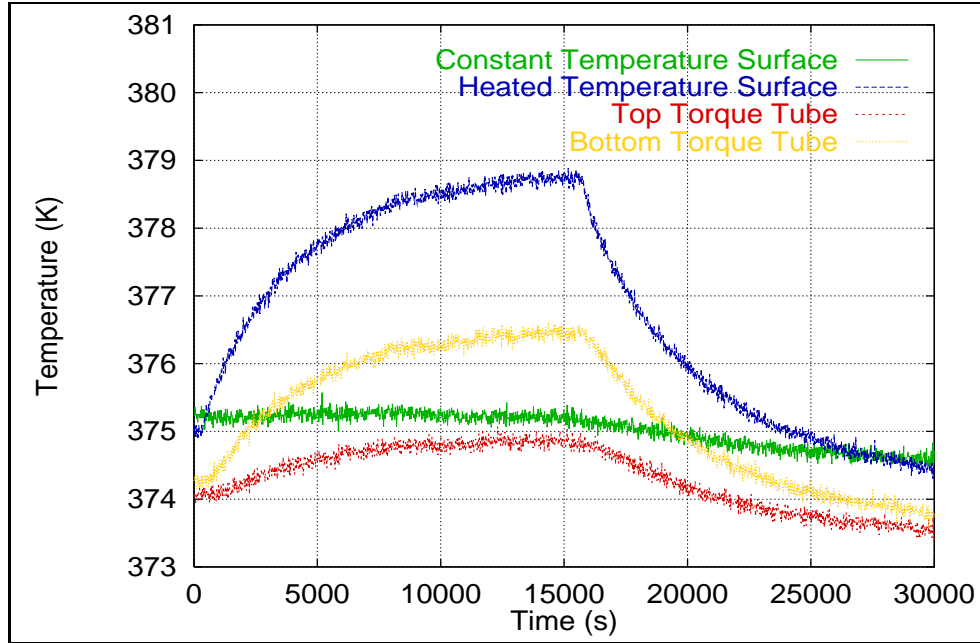


Figure 4.27 Typical experimental temperature distributions.

the accuracy of the thermocouple readings. As mentioned earlier, absolute temperatures and absolute temperature measurement inaccuracies do not affect the estimation procedure which is based on the changes in temperature. Therefore, offsetting the temperature readings will have no impact on the property estimates. Fig. 4.28 shows the same temperature distributions plotted in Fig. 4.27 after elimination of the offset through data manipulation.

After data manipulation, all the temperature distributions start with a common initial temperature and are free of bias error due to the reference junctions. Additional data manipulation was necessary to convert the datafiles into a useable format compatible with the finite element program EAL, described in Chapter 5.

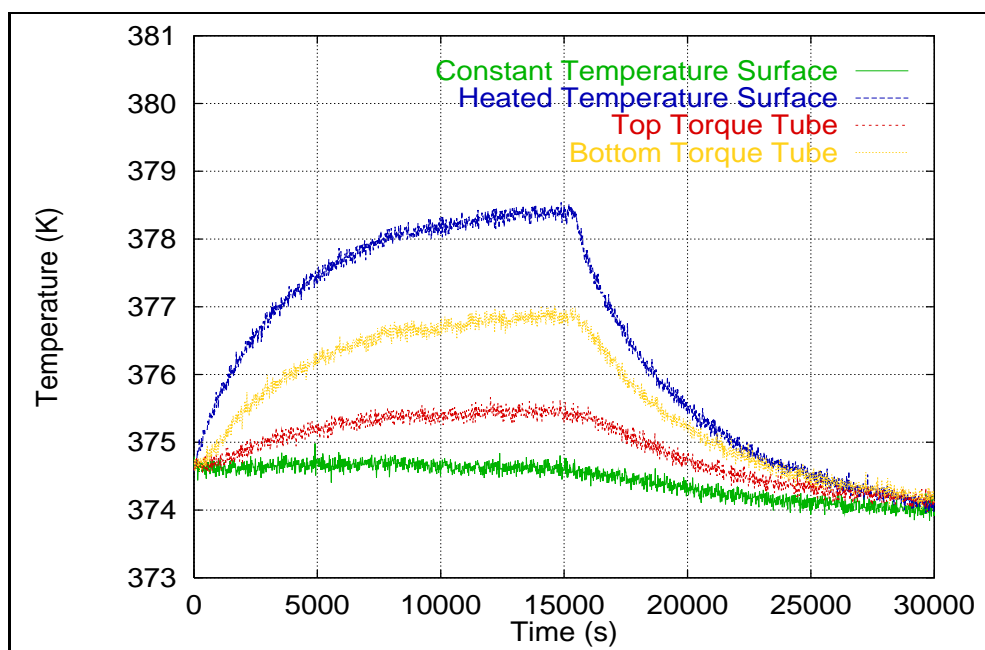


Figure 4.28 Typical offset experimental temperature distribution.

Chapter 5

Implementation of the Finite Element Program EAL

The numerical models used to perform the optimization of experiments and estimation of parameters were developed using the **E**ngineering **A**nalysis **L**anguage (EAL), created by Engineering Information Systems Incorporated©. EAL is a finite element software package that was mainly designed for structural applications, but it also has strong thermal analysis capabilities. The initial steep learning curve is quickly rewarded by the irreplaceable flexibility offered by this software. One important feature of EAL that was extensively used during this project, was the ability to complement the already existing processor library with new, user defined, task specific processors by recompiling the source code. The additional processors were developed to incorporate the Genetic Algorithm into the EAL runstream, and therefore introduce the minimization procedure used during the optimization and estimation processes.

The first section in this chapter gives an overview of the finite element software package EAL. The second section revolves around the determination of the optimal integration time step and mesh size. To increase our confidence in the results of the finite element package used throughout this study, the one-dimensional effective model was solved both analytically and numerically, the results compared and described in the section following the integration time step and mesh size analysis.

5.1 The Finite Element Program EAL

This section is divided into two parts. The first part gives a broad overview of the finite element program EAL, whereas the second describes EAL's transient temperature solver.

5.1.1 An Overview of EAL

EAL is a finite element software package that requires the knowledge of the interpreting programming language, through which the user can issue commands that EAL transforms into actions. EAL is a system of independent processors which communicate with each other through random-access databases consisting of one or more libraries of named datasets, Whetstone (1983). Most of the processors have predetermined reset options that allow the user to fine-tune the available processors to the specific applications. Many of the processors contain built-in self-testing features that caution the user when possible sources of errors arise. A good example of such a self-testing feature is the ASPECT RATIO warning, cautioning the user about the aspect ratio of certain elements in the specified mesh, which may have a negative effect on the solution.

5.1.2 EAL's Transient Solution Technique

The two one-dimensional and the two-dimensional models developed in EAL to calculate the transient temperature histories at different locations of the outer wing subcomponent primarily used one-dimensional linear (K21) and two-dimensional bilinear (K41) conduction elements. The matrix equations of heat transfer were developed using the Galerkin method. The governing matrix equation is defined in Eq. 5.1:

$$(\mathbf{K}_{\text{cond}} + \mathbf{K}_{\text{conv}} + \mathbf{K}_{\text{rad}} + \mathbf{K}_{\text{mass}})\mathbf{T} + \mathbf{C}\dot{\mathbf{T}} = \mathbf{COND} + \mathbf{CONV} + \mathbf{RAD}, \quad (5.1)$$

where \mathbf{K}_{cond} is the conduction matrix, \mathbf{K}_{conv} is the convection matrix, \mathbf{K}_{rad} is the radiation matrix, \mathbf{K}_{mass} is the mass transport matrix, \mathbf{C} is the capacitance matrix, \mathbf{COND} is the conduction load vector, \mathbf{CONV} is the convection load vector, and \mathbf{RAD} is the radiation load vector. However, any convection or radiation effects in the models were integrated into the conduction terms, whereas mass transport was non-existent, thus reducing Eq. 5.1 to Eq. 5.2:

$$\mathbf{K}_{\text{cond}}\mathbf{T} + \mathbf{C}\dot{\mathbf{T}} = \mathbf{COND}. \quad (5.2)$$

During the calculation of the transient temperatures, the stiffness matrix \mathbf{K}_{cond} and the capacitance matrix \mathbf{C} were only calculated once, since for each surrounding temperature, the thermophysical properties and the boundary conditions were considered constant and independent of temperature due to the small temperature rise (5°C-10°C). The finite difference solution in the time domain was applied to finally obtain numerical solutions of the temperature in space and time. The finite difference solution was generated using the central difference method, also known as the Crank-Nicolson method, with $\theta = 0.5$.

5.2 Determination of Optimal Mesh Size and Time Step

The following sections are dedicated to presenting how the optimal meshsize and time step for the one-dimensional and two-dimensional finite element models were determined

5.2.1 The Optimal Mesh Sizes of the Finite Element Program

One of the biggest advantages EAL offers to its users is its flexibility in the area of mesh construction. When using a numerical solution technique such as finite difference or finite elements, it is very important to determine an adequate mesh size for which the solution is as accurate as possible, requiring the least amount of computational time. To investigate which mesh size yields an acceptable solution, the solutions to many different mesh sizes need to be compared. During the mesh size analysis, the time step used to calculate the temperature distributions was extremely small, in order to eliminate its influence on the output.

A technique that is widely used among finite element analysts consists in increasing the mesh density in the areas of steepest gradients and is called the mesh refinement technique. The mesh was enhanced by doubling the number of elements in the appropriate areas until the increase in mesh density had no affect on the sought temperature histories. This method was applied to the one-dimensional and two-dimensional models to determine the optimal mesh size and element distribution.

Figures 5.1-5.2 are plots of the residuals between the temperature distribution corresponding to the optimal mesh size and the temperature distributions of other non-optimal meshes for the one-dimensional effective and one-dimensional detailed model respectively. The plots reveal the highest temperature discrepancies immediately following where the

heater is turned on and turned off, where the temperature gradients are the highest. Only an appropriate mesh has the capability to resolve the steep temperature gradients occurring immediately after the heater is turned on and off. The appropriate mesh sizes that were chosen for the one-dimensional effective, one-dimensional detailed, and two-dimensional model are summarized in Table 5.1.

5.2.2 The Optimal Time Steps of the Finite Element Program

Determining the optimal time step to use in a numerical solution method is as crucial as establishing the optimal mesh size. Both the mesh size and the integral time step can have detrimental effects on the solution if chosen incorrectly. Although the Crank-Nicolson scheme usually yields a stable solution, the solution may still converge on the wrong temperature distributions. Therefore, once the optimal mesh size was determined, only the integral time step was varied to get a better understanding of its influence on the solution. The process of determining the optimal time step is very similar to determining the optimal mesh size. The time step was halved from run to run, until the temperature distribution solution did not change significantly. Fig. 5.3 and Fig. 5.4 are plots of the residuals of the temperature history at the most sensitive location between the optimal time step and non-optimal time step for the one-dimensional effective, one-dimensional detailed, and two-dimensional models, respectively. Fig. 5.3 and Fig. 5.4 reveal that for large time steps, the physics of the problem can not be captured, introducing noticeable error into the solution. Choosing a time step that is much smaller than the optimal one, will not increase the accuracy of the solution significantly, but will drastically increase the computational time. Table 5.2 summarizes the optimal time steps that yield accurate results for the one-dimensional effective, one-dimensional detailed, and two-dimensional models.

Table 5.1 The optimal mesh sizes for the finite element models.

Finite Element Model	Number of Elements
One-dimensional effective	58
One-dimensional detailed	103
Two-dimensional	5152

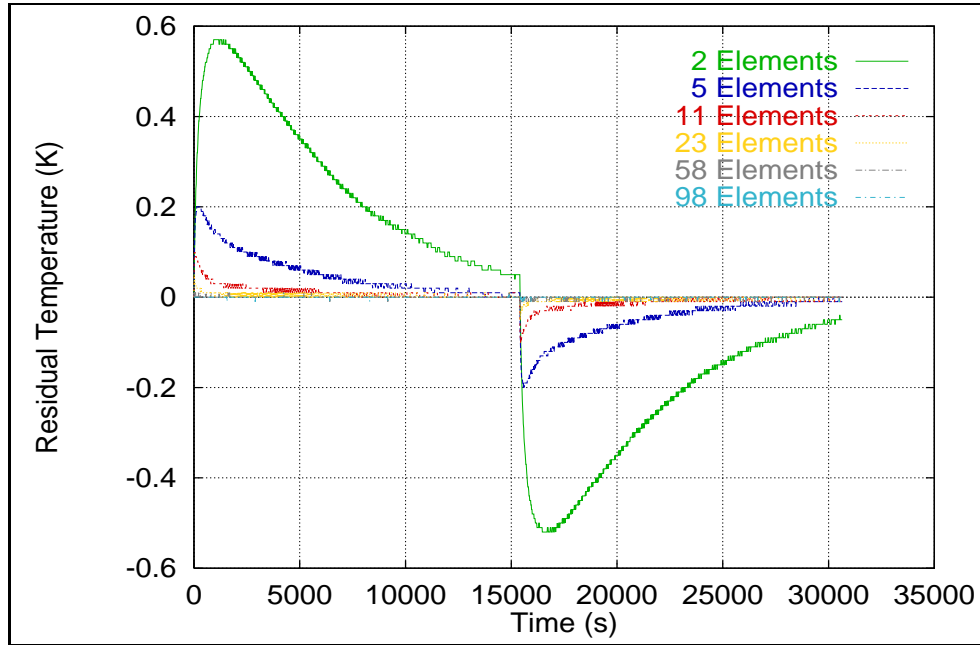


Figure 5.1 The influence of the mesh size on the temperature at the optimal sensor location for the effective one-dimensional model with a nominal mesh size of 198 elements and time step of 0.005 sec.

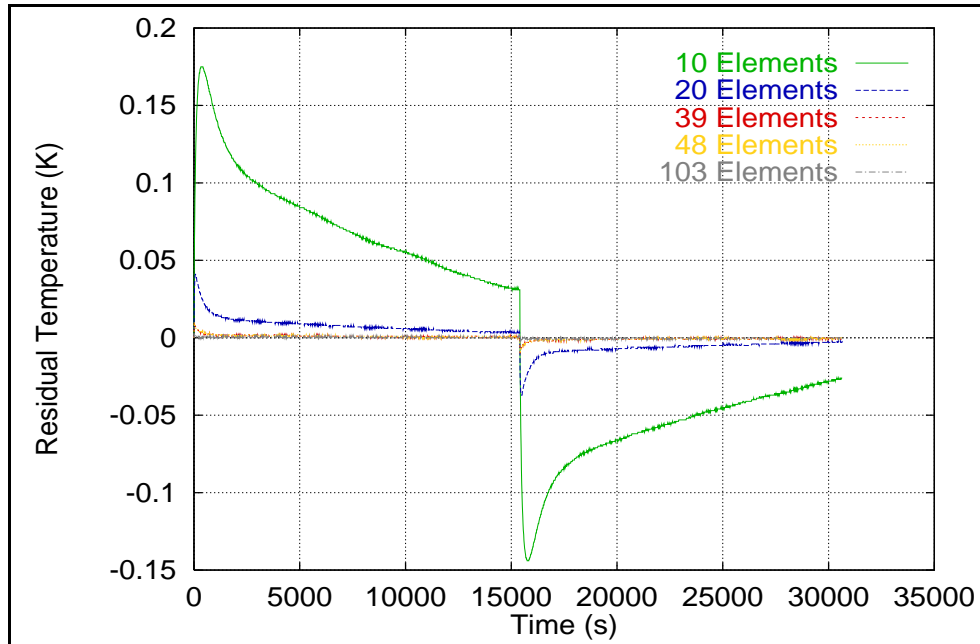


Figure 5.2 The influence of the mesh size on the temperature at the optimal sensor location for the detailed one-dimensional model with a nominal mesh size of 222 elements and a time step of 0.005 sec.

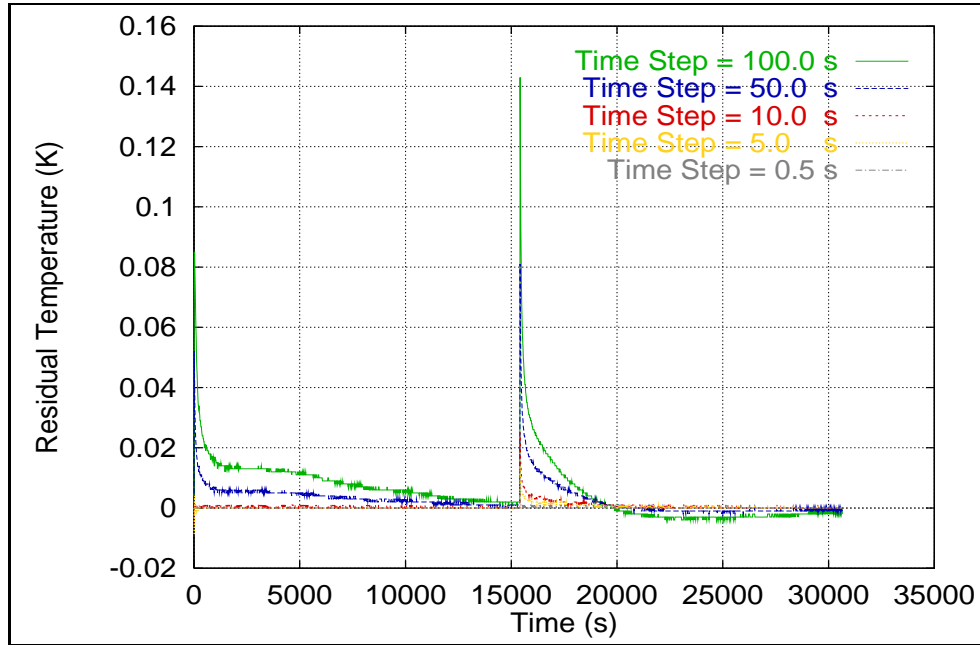


Figure 5.3 The influence of the time step on the temperature at the optimal sensor location for the one-dimensional effective model with a nominal mesh size of 198 elements and a time step of 0.005 sec.

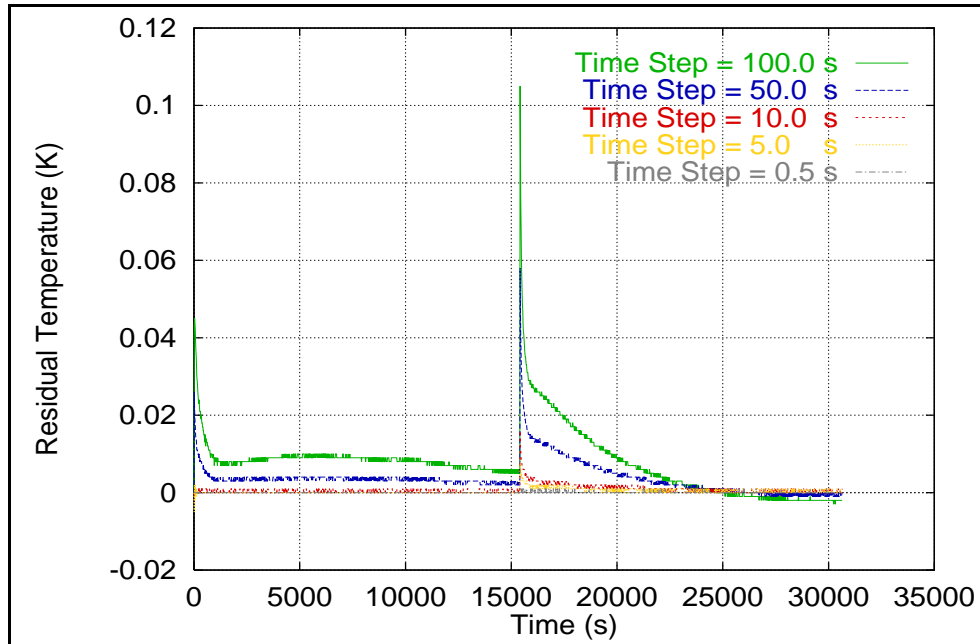


Figure 5.4 The influence of the time step on the temperature at the optimal sensor location for the detailed one-dimensional model with a nominal mesh size of 222 elements and a time step of 0.005 sec.

Table 5.2 The optimal time steps for the finite element models.

Finite Element Model	Integral Time Step (s)
One-dimensional effective	0.5
One-dimensional detailed	0.5
Two-dimensional	0.5

5.3 An Analytical Validation of the Finite Element Package EAL

The finite element package EAL was used throughout this study to generate optimization and estimation results. In order to increase the confidence of the EAL output, the one-dimensional effective model was translated mathematically into the following governing equation, boundary conditions and initial conditions:

$$\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t} \quad , \quad 0 \leq x \leq L, \quad t > 0, \quad (5.3)$$

$$-k \frac{\partial T}{\partial x} = q_s \quad , \quad x = 0, \quad t > 0 \quad (5.3a)$$

$$T = T_i \quad , \quad x = L, \quad t > 0 \quad (5.3b)$$

$$T = T_i \quad , \quad t = 0, \quad 0 \leq x \leq L \quad (5.3c)$$

where k_{eff} is the effective thermal conductivity, x is the location through the thickness of the outer wing subcomponent, L is the total thickness of the structure, α_{eff} is the effective thermal diffusivity, T_i is the initial temperature as well as the temperature at the constant boundary, and q_s is the constant heat flux applied to the top surface. The analytical solution to Eq. 5.3 was determined using the Green's function method, and can be expressed as follows, by Eq. 5.4:

$$T(x, t) = T_i + \frac{q_s}{k} \frac{2}{L} \sum_{m=1}^{\infty} \cos(\beta_m x) \frac{1}{\beta_m^2} (1 - e^{-\alpha \beta_m^2 (t-t_0)}), \quad (5.4)$$

where $\cos(\beta_m x)$ are the eigenfunctions, β_m are the eigenvalues, and t_0 is the time at which the heat flux is applied to the structure. The details of the derivation can be found in

Appendix B. Table 5.3 gives a summary of the test parameters that were used to solve for the transient temperature distribution at the heated surface using the finite element software EAL and the analytical solution.

Fig. 5.5 shows the transient temperature distributions at the heated node on the top surface of the outer wing subcomponent, solved both numerically and analytically. The two plotted solutions of the temperature histories are superposed and follow the expected exponential trend before reaching steady-state. Since the two solution curves are almost identical, it is interesting to view the residuals between the two solutions. Fig. 5.6 is a plot of the difference between the analytical and numerical solutions. The magnitude of the residuals is on the order of 10^{-3} , which is about two orders of magnitude smaller than the accuracy of the sensors, and therefore insignificant. The slight discrepancy in the solutions is due to the amount of terms used in the summation in Eq. 5.4. Varying the number of eigenvalues and eigenfunctions used to calculate the temperature distribution with the Green's function method influences the accuracy of the solution.

The comparison of the results of the two independent solution techniques was important to increase the confidence in the remaining output of the software. A detailed explanation on the exact solution technique used by the transient analyzer of EAL can be read in the EAL SPAR Thermal Analysis Processors Reference Manual Volume 2 which was dedicated to explaining the theory behind the transient thermal analyzer (TRTB-processor). However, a brief overview of the Eal's transient analyzer is given in the following section.

5.3.1 Implementation of Genetic Processors into EAL

Appendix C is dedicated to introduce the reader to the details of some of the EAL processors that were widely used in the optimization and estimation programs. Appendix C also

Table 5.3 Test parameters for EAL verification.

Thermal Conductivity	k_{eff}	0.7	(W/mK)
Thermal Diffusivity	α_{eff}	10^{-6}	(m/s)
Structure Height	L	0.1235	(m)
Heat Flux	q_s	20.0	(W/m ²)
Time of Heat Flux Application	t_0	0.0	(s)
Initial Temperature	T_i	375.0	(K)

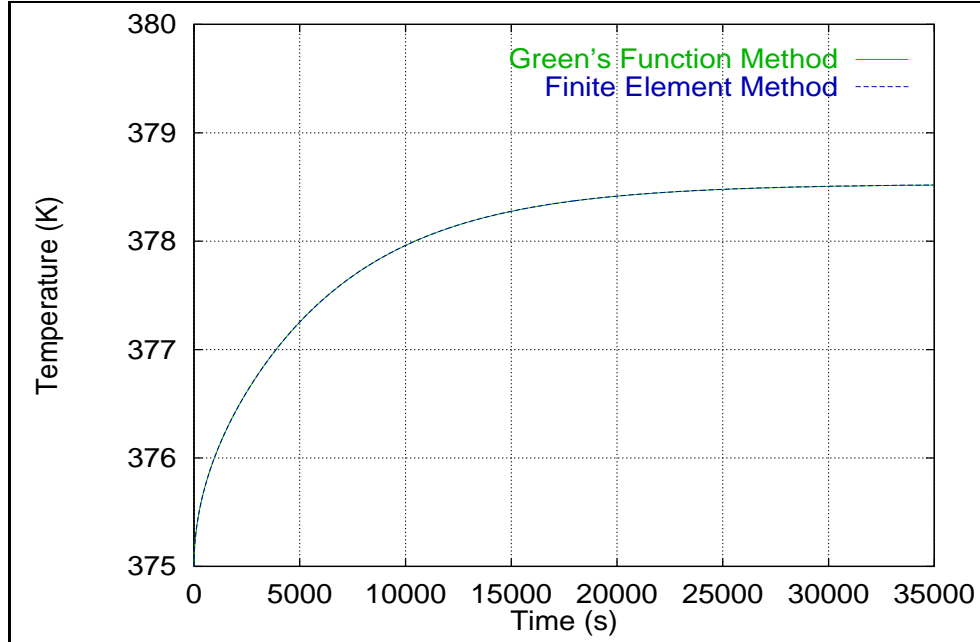


Figure 5.5 A comparison of the temperature history solutions at the heated surface of the outer wing subcomponent obtained with the finite element model and the Green's function.

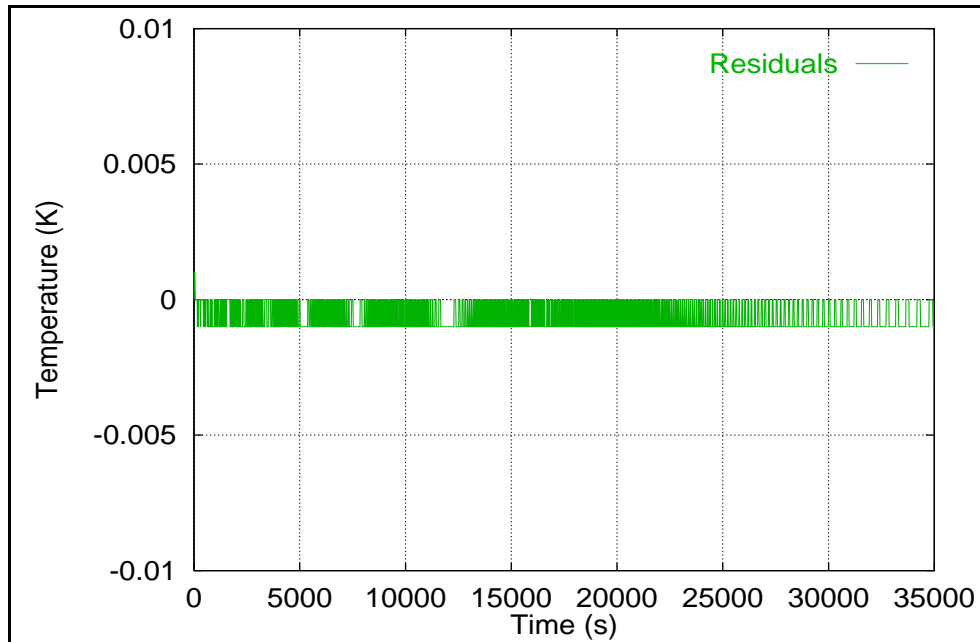


Figure 5.6 Residuals of the temperature history solutions at the heated surface of the outer wing subcomponent obtained with the finite element model and the Green's function.

describes how the additional genetic processors were developed and introduced into the finite element package. Fig. 5.7 shows how the original **EEGA** was translated into the five distinct processors GENA, GENB, GENC, GEND, GENE.

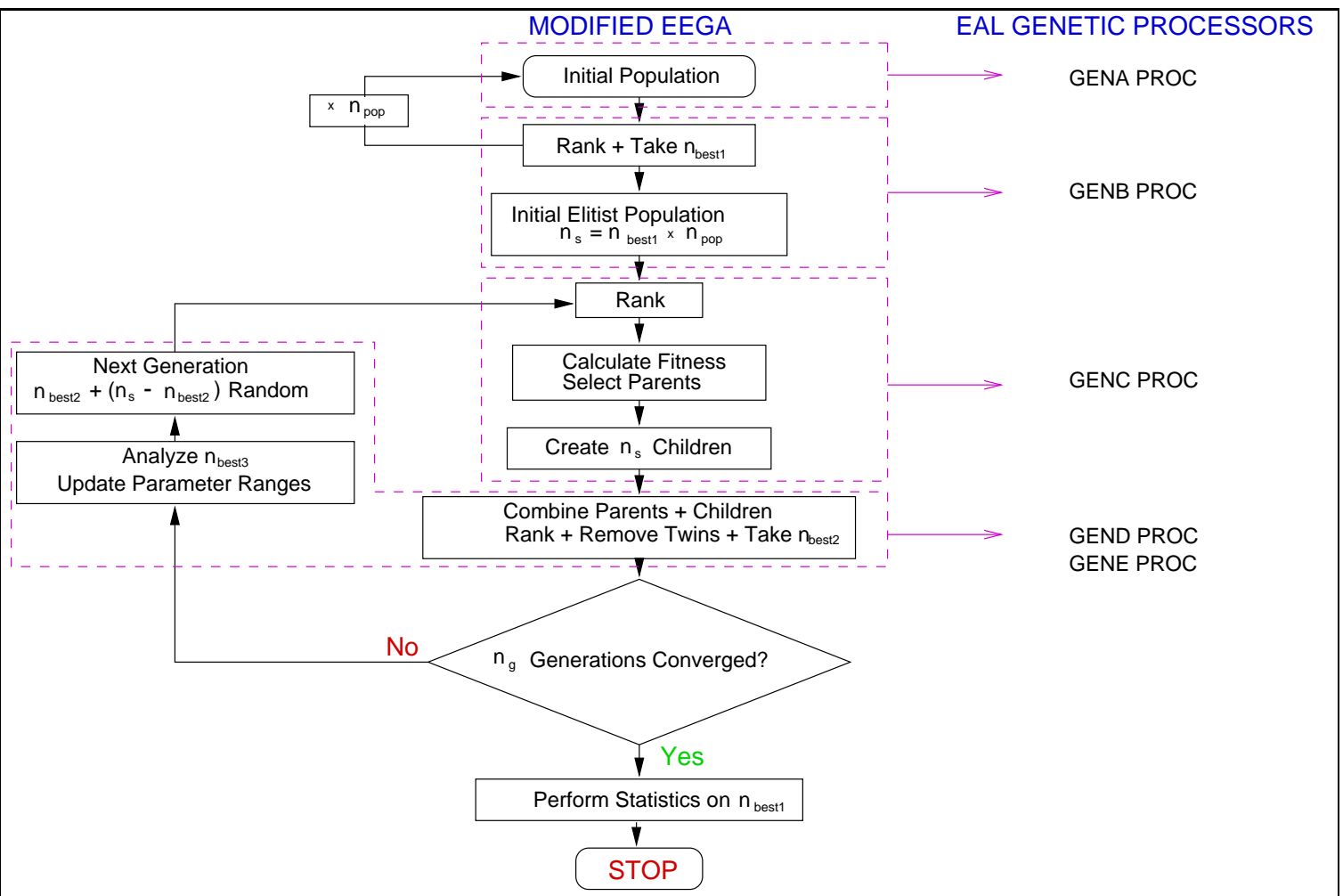


Figure 5.7 Comparison of the Extended Elitist Genetic Algorithm developed by Sandrine Garcia (1997) with convergence criteria and the EAL genetic processors.

Chapter 6

Results and Discussion

This chapter is dedicated to presenting the results of the different analyses performed on the outer wing subcomponent. The first section reveals the findings of the initial steady-state analysis and shows how the results affected the remaining models. In the second section, the findings of the design optimization program for the one-dimensional and two-dimensional models are presented, including the optimal sensor locations and the optimal heating times. The third section gives an overview of the sensitivity coefficients and discusses their importance. The second to last section provides the estimation results for the different thermal properties, the estimate's confidence intervals, and the corresponding residuals, whereas the last section shows how convergence was treated using the Genetic Algorithm.

6.1 Steady-State Analysis

The steady-state analysis was performed using the thermal resistance diagram described in Chapter 3 and illustrated in Fig. 3.2, to obtain a better understanding of the problem at hand and to give an approximate value of the effective thermal conductivity of the outer wing subcomponent. Table 6.1 gives an overview of the approximated values of the effective thermal conductivity of the outer wing subcomponent obtained through the steady-state thermal resistance analysis. Fig. 6.1 illustrates Table 6.1, evoking a near linear temperature dependence of the effective thermal conductivity. The material properties used to generate the approximate values for the effective thermal conductivity were selected among material properties of similar type. Therefore, these results only give an indication of the order of magnitude that should be expected for the values of the effective thermal conductivity for

the more detailed finite element models. Based on these preliminary results, the initial parameter range for the effective thermal conductivity in the genetic algorithm could be set in the neighborhood of the values found. This drastically decreased the computational time of the genetic algorithm, by decreasing the number of generations that needed to be performed to reach convergence.

6.2 Design Optimization

The objective of the optimization process was to design an optimal experiment that would yield optimal data for the estimation process, and therefore produce the most accurate estimates with the smallest confidence intervals, by maximizing the sensitivity coefficients of the parameters to be estimated. The parameters that were optimized included the sensor locations, the heating time and indirectly the total experimental time. The optimization was performed for both, the one-dimensional and two-dimensional experimental setups using the D-optimal criterion. This criterion maximizes the determinant of the $\mathbf{X}^T \mathbf{X}$ matrix, And defines the optimal design as the one with the largest determinant.

Following are two sections describing the findings of the optimal sensor locations and heating times for the one-dimensional and two-dimensional experimental setups.

6.2.1 Optimal Sensor Locations

The optimization for the sensor location for the one-dimensional and two-dimensional experimental setups was achieved by determining the location of the outer wing subcomponent for which the time averaged determinant of the $\mathbf{X}^T \mathbf{X}$ matrix was maximized. A time averaged determinant is defined as the the determinant of the $\mathbf{X}^T \mathbf{X}$ matrix divided by the elapsed experimental time. An EAL runstream with an incorporated genetic algorithm was written to perform this analysis. The initial parameters such as applied power or initial temperature used in the optimization were selected based on results obtained by Hanak (1995) and Moncman (1994). A summary of the optimal sensor locations for the one-dimensional and two-dimensional experimental setups is given in Table 6.2 below. The findings of this optimization correspond with the results published by Moncman (1994), who provided optimal non-dimensional sensor locations for one-dimensional and two-dimensional experiments. Visualization of the one-dimensional and two-dimensional optimal sensor locations is given by

Table 6.1 Approximate effective thermal conductivity using steady-state analysis.

Temperature (K)	Thermal Conductivity (W/mK)
295	0.219
310	0.222
325	0.226
340	0.228
355	0.232
370	0.235
385	0.238
400	0.240
415	0.243
430	0.245
445	0.248

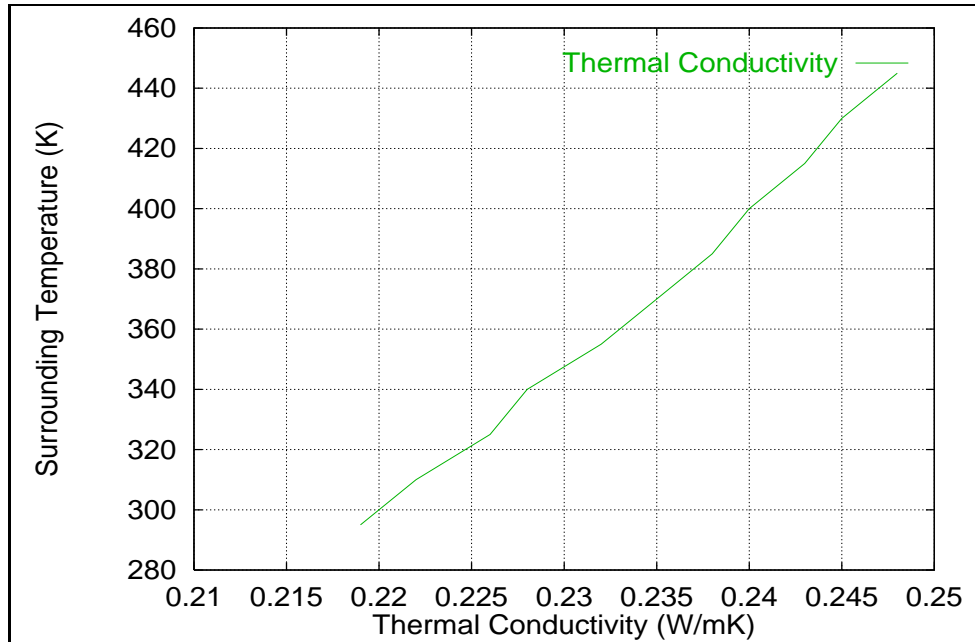


Figure 6.1 Effective thermal conductivity values from steady-state analysis.

Figs. 6.2 and 6.3.

6.2.2 Optimal Heating Time

The second parameter that was optimized for both the one-dimensional and two-dimensional experimental setups was the heating time. Optimizing the heating time can greatly reduce the experimental time, since in this case, the total experimental time was defined as twice the heating time. Moreover, if the heating time is too large and the structure has reached quasi steady-state, although the sensitivity is at its maximum, the additional information obtained for the thermal conductivity is redundant. No, or very little information is gained for the volumetric heat capacity when approaching steady-state due to the transient nature of the thermal property, as will be shown in the sensitivity coefficient analysis.

The optimization process adopted to determine the optimal heating times was the same as the one described above and included the maximization of the time averaged determinant of the $\mathbf{X}^T\mathbf{X}$ matrix. The optimal heating time remained unchanged from one surrounding temperature to another, since the thermal properties were assumed temperature independent. Table 6.3 gives a summary of the results obtained for the optimal heating time.

From Fig. 6.4 and Fig. 6.5 which represent typical recorded temperature histories for the one-dimensional and two-dimensional experimental setups, it becomes apparent that the optimal heating time is reached slightly before the temperature curves reach steady-state. The optimal heating time for the two-dimensional experiment was slightly longer than the optimal heating time for the one-dimensional case.

Table 6.2 Optimal sensor locations for the one-dimensional and two-dimensional experimental setups.

Setup Type	Optimal Sensor Location X (m)	Optimal Sensor Location Y (m)
One-Dimensional	0.0	n/a
Two-Dimensional	0.0	0.4883

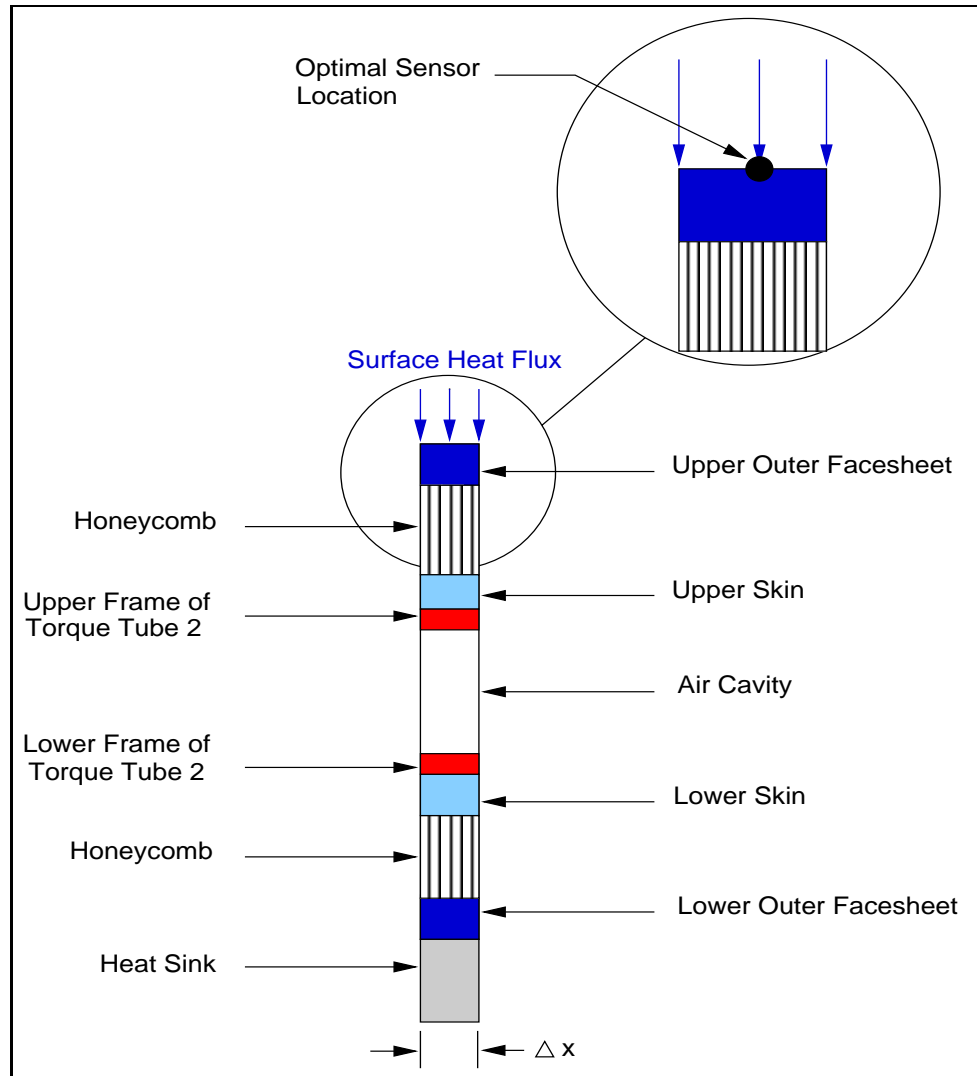


Figure 6.2 Optimal sensor location for the one-dimensional experimental setup.

Table 6.3 Heating Times for the one-dimensional and two-dimensional experimental setups.

Setup Type	Optimal Heating Time (s)
One-Dimensional	15825.0
Two-Dimensional	17068.0

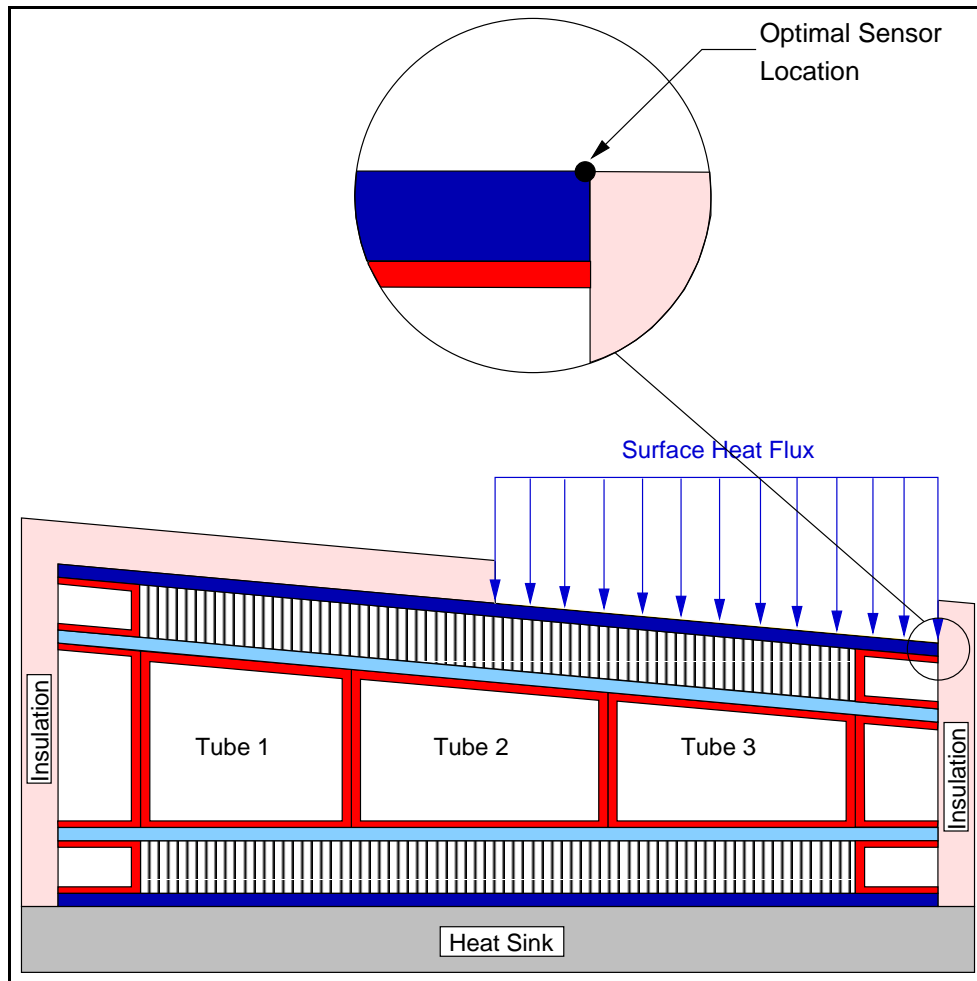


Figure 6.3 Optimal sensor location for the two-dimensional experimental setup.

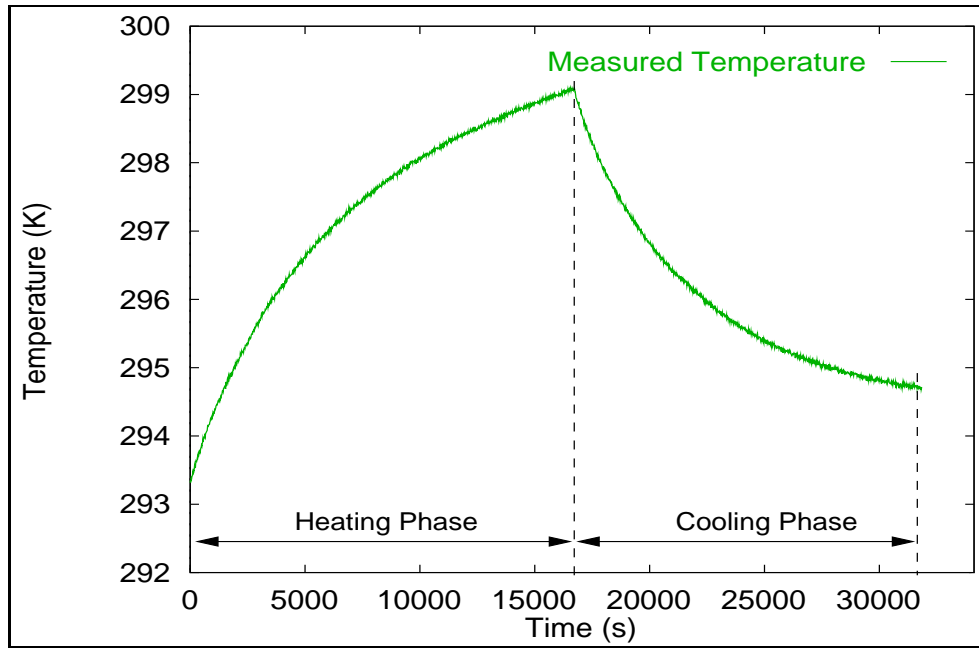


Figure 6.4 Heating and cooling phases of one-dimensional experimental setup.

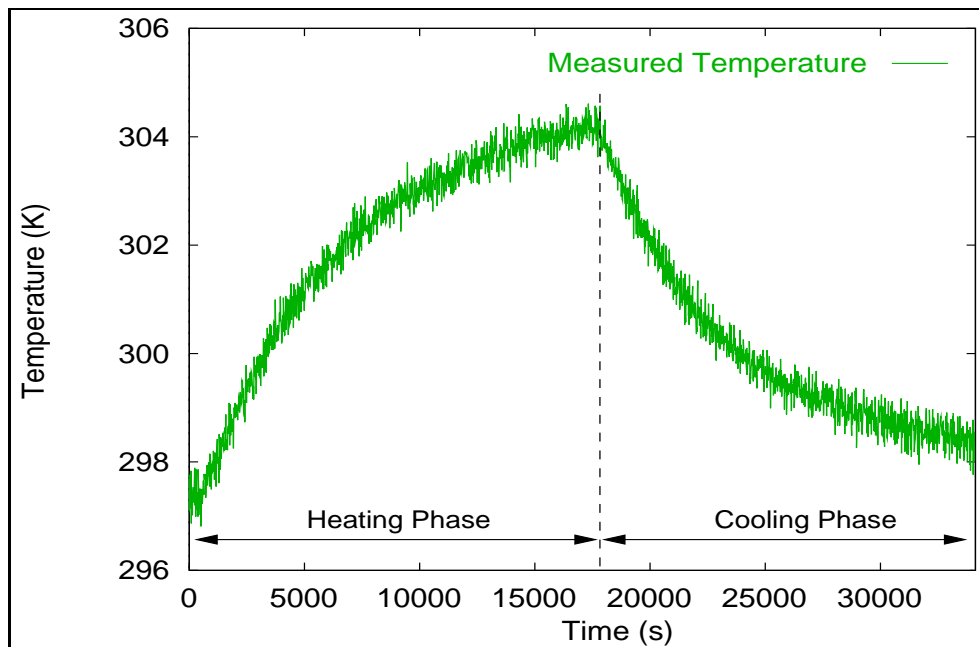


Figure 6.5 Heating and cooling phases of two-dimensional experimental setup.

6.3 Sensitivity Coefficients

Determining the sensitivity coefficients can give much insight in the problem in hand. The larger the sensitivity coefficients, the more accurate the estimates and the smaller the confidence intervals. Although genetic algorithms are more robust in handling correlated parameters, linear dependence between the sensitivity coefficients of the parameters can result in accuracy losses and estimation difficulties. Following are three sections describing the sensitivity coefficients of the thermal properties for the one-dimensional effective model, the one-dimensional detailed model, and the two-dimensional model.

The Sensitivity Coefficients for the One-Dimensional Effective Model

For the one-dimensional effective model, the sensitivity coefficients of the effective thermal conductivity k_{eff} and volumetric heat capacity C_{eff} were analyzed, to determine which of the two properties had a dominant effect on the temperature solution. The shapes of the sensitivity curves displayed in Fig. 6.6 describe two very distinct paths, and therefore do not show any signs of correlation. The sensitivity of the thermal conductivity sees a steady increase until the optimum heating time is reached. Once the heater is turned off, the sensitivity decreases steadily to ultimately reach zero once the structure reaches equilibrium at the surrounding temperature. The maximum sensitivity for the thermal conductivity is reached when the temperature is close to steady-state. The sensitivity of the volumetric heat capacity increases rapidly at early stages of the heating and cooling phases, reaching its greatest magnitude increase during the steep temperature gradient phase before flattening out as the temperature reaches steady-state. Close to steady-state, very little additional information is gained for the volumetric heat capacity, due to the transient nature of the thermal property. To double the amount of information on the parameters, data were recorded throughout the experiment, including the heating and the cooling phases of the experimental runs.

During the estimation process, a boundary condition adjustment was carried out on the one-dimensional effective model, to more closely represent the physics of the problem. The formerly completely insulated top boundary of the outer wing subcomponent was changed by adding a convective boundary condition over the insulation boards on the top surface of the outer wing subcomponent. The convective effects were incorporated into the

conduction properties of the insulation boards as described in Appendix A. The impact of this model change is significant and will be shown in the estimation results. Fig. 6.6 compares the sensitivities of the volumetric heat capacity and thermal conductivity for both the convective and insulated boundary conditions. The magnitude of the sensitivities was very much affected by the different boundary conditions as shown in Fig. 6.6. The magnitude of the sensitivity of the thermal conductivity is significantly higher for the insulated case, whereas the sensitivity of the volumetric heat capacity is barely affected by the boundary change.

The Sensitivity Coefficients for the One-Dimensional Detailed Model

The sensitivity coefficients of the volumetric heat capacity and thermal conductivity of the facesheets, honeycomb, skins, and torque tubes were calculated using a half-model derived from the one-dimensional detailed model, shown in Fig. 6.7. The change in model was necessary after estimation difficulties were encountered due to the highly nonlinear radiation heat transfer occurring in the torque tubes, preventing the estimator to converge on a solution. Since both the upper and lower sections of the outer wing subcomponent are geometrically equivalent, either section could have been used as the half-model. However, since the upper section was closest to the heated surface, the sensitivity was the highest, therefore increasing the accuracy of the estimates.

The sensitivities of the facesheets, honeycomb, skins, and torque tubes are illustrated in Fig. 6.8. The plot reveals great magnitude differences between the different sensitivity coefficients. The sensitivity coefficient of the volumetric heat capacity and thermal conductivity of the honeycomb are the greatest followed by the sensitivity coefficient of the thermal conductivity of the skins. The other sensitivity coefficients are very small in magnitude. The thickness of the different components plays an important role in determining the sensitivity coefficients. Since the skins, facesheets, and torque tubes have very thin wall thicknesses, it is difficult to obtain a temperature gradient across the thickness, therefore reducing the sensitivity of the component. Thus, for the previously mentioned components, the lumped capacitance model is very appropriate to characterize the thermal response of these parts.

Another noticeable trend is the correlation between the different sensitivities. Many of the curve shapes are very similar to each other, introducing a high degree of correlation

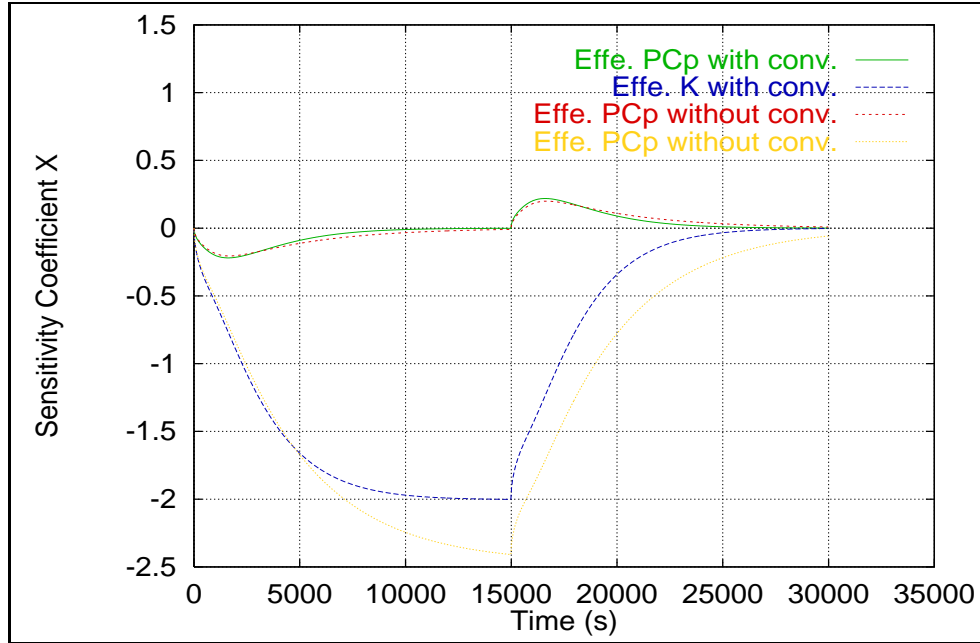


Figure 6.6 Comparison between the sensitivity coefficients of the thermal conductivity and volumetric heat capacity for the one-dimensional effective model with and without convective top boundary.

between the parameters and making the estimation process very difficult due to the non-uniqueness of the solution.

The Sensitivity Coefficients for the Two-Dimensional Model

The sensitivity coefficients for the two-dimensional model only involved the in-plane thermal conductivities of the facesheets, the skins, the honeycomb blanket, and the torque tubes, as well as the volumetric heat capacity and thermal conductivity of the air residing in the cavities of the torque tubes. Fig. 6.9 illustrates the sensitivity coefficients for the thermal properties listed previously. It is apparent that the dominant sensitivities are the thermal conductivity of the honeycomb and the air. The remaining sensitivities are almost non-existent which will have an important impact on the accuracy of the estimates.

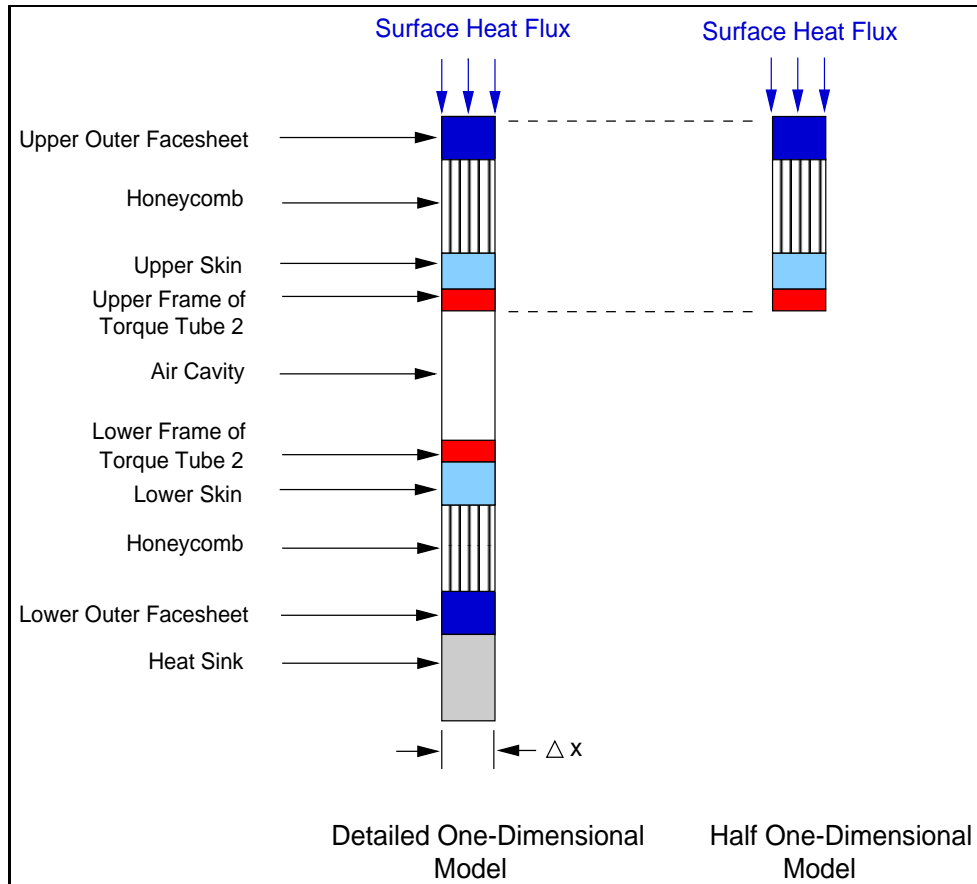


Figure 6.7 One-dimensional half-model derived from one-dimensional detailed model.

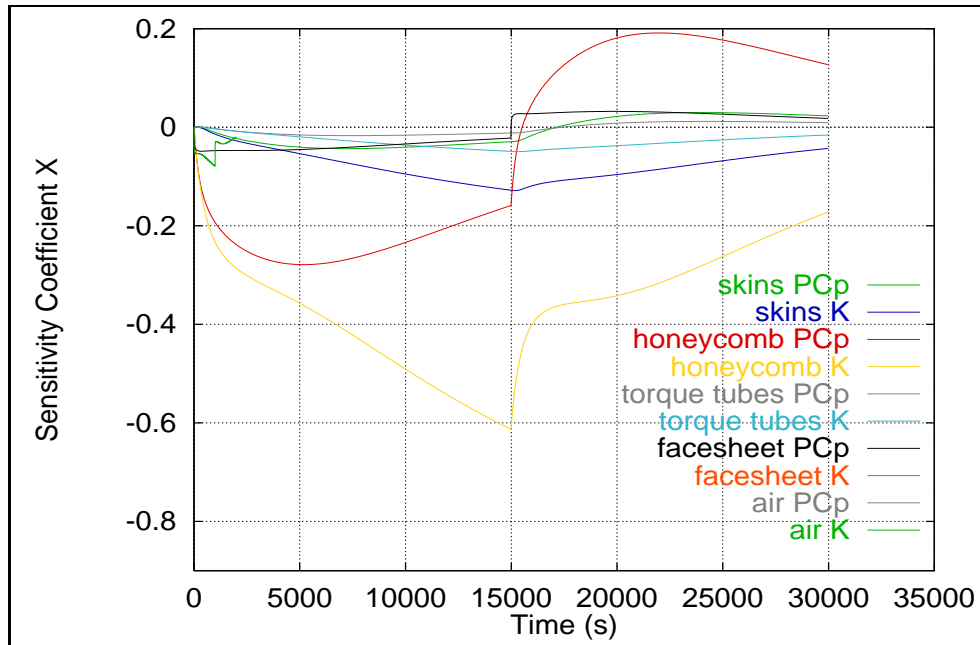


Figure 6.8 Sensitivity coefficients for the one-dimensional detailed model.

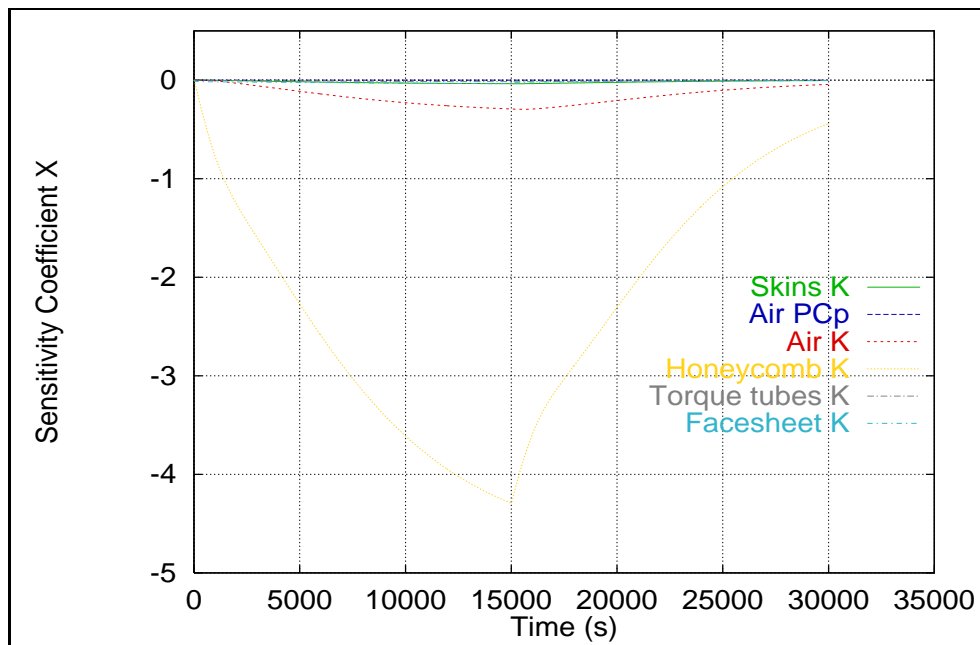


Figure 6.9 Sensitivity coefficients for the two-dimensional model.

6.4 Estimation Results

Several models were investigated to solve for the different thermal properties. Two one-dimensional effective models were developed to determine the effective thermal conductivity and volumetric heat capacity of the outer wing subcomponent and to evaluate both an insulated and convective boundary condition over the insulation boards of the top surface. A third detailed one-dimensional model was developed to estimate the volumetric heat capacity and through the thickness thermal conductivity of the facesheets, honeycomb, skins, torque tubes, and air. However, due to the highly nonlinear radiative properties of the torque tubes, the model was simplified to a half-model, including only the top section of the detailed one-dimensional model. The estimates obtained from the one-dimensional half-model were implemented into a two-dimensional model which in turn estimated the remaining in-plane thermal conductivities of the facesheets, honeycomb, skins, torque tubes, as well as the volumetric heat capacity and thermal conductivity of the air present in the torque tubes.

6.4.1 One-Dimensional Effective Estimation Results

The effective thermal properties were estimated for both the insulated and convective boundary conditions, and the results were then compared by calculating the residuals between the experimental and calculated temperature histories for each model.

Thermal Property Estimates with Confidence Intervals for the Insulated Boundary Condition

The first one-dimensional effective model was developed using a perfectly insulated boundary condition on the insulation boards of the top surface of the outer wing subcomponent. The values of the estimates for the thermal conductivity and volumetric heat capacity for each experiment are tabularized in Table 6.4 and Table 6.5 with the corresponding 95% confidence intervals calculated using Eq. 3.8 described in Chapter 3. The results are illustrated in Fig. 6.10 and Fig. 6.11. The averages of the estimates of the thermal conductivity and volumetric heat capacity at each surrounding temperature setting are given in Table 6.6 and Table 6.7 along with the 95% confidence intervals, which were calculated using equation Eq. 3.9 described in Chapter 3.

The values for the thermal conductivity range from 0.276 to 0.784 W/mK . The size of the interval for the means of the estimates is due to the temperature dependence of the thermal conductivity with respect to the surrounding temperature, although the thermal properties were assumed constant for each surrounding temperature setting, as mentioned earlier. The temperature dependence of the thermal conductivity is clearly visible in Fig. 6.12 and was verified using a student's *t* means test. The mean value of the thermal conductivity increases as the surrounding temperature is raised as shown in Table 6.6. However, the estimates are consistent at each surrounding temperature setting, showing little variation. The confidence intervals for experiments at room temperature are smallest and increase slightly as the surrounding temperature for the experiments is raised, which may indicate that the accuracy is lost as the temperature increases.

The range of the estimates of the volumetric heat capacity in the case of the insulated boundary on the top surface of the outer wing subcomponent varies from 2.424 to $4.719 \times 10^5 J/m^3K$. From Fig. 6.11 and Fig. 6.13, no temperature dependence could be determined for the volumetric heat capacity. A student's *t* means test proved that the confidence intervals for the means for each surrounding temperature setting overlapped and therefore did not show temperature dependence. The confidence intervals increase slightly as the surrounding temperature is raised.

Thermal Property Estimates with Confidence Intervals for the Convective Boundary Condition

The estimates for the effective thermal conductivity and volumetric heat capacity for the one-dimensional effective model assuming a convective boundary condition over the insulation boards of the top surface are summarized in Table 6.8 and Table 6.9 and illustrated in Fig. 6.14 and Fig. 6.15 respectively. Both figures are plots of the mean values of the estimates with the corresponding 95% confidence intervals which were also calculated using Eq. 3.8. The averages of the estimates of the thermal conductivity and volumetric heat capacity for each surrounding temperature setting are represented in Table 6.10 and Table 6.11 respectively along with the 95% confidence intervals evaluated with Eq. 3.9.

The majority of the estimates for the thermal conductivity are contained in an interval ranging from 0.212 to 0.688 W/mK . As for the insulated case, the spread of the thermal conductivity values is due to its temperature dependence with respect to the

Table 6.4 One-dimensional effective thermal conductivity estimates with insulated top boundary.

Temperature (K)	Experiment Number	Thermal Conductivity (W/mK)	95% Confidence Interval (W/mK)
295	1	0.276	± 0.0036
295	2	0.468	± 0.0002
295	3	0.444	± 0.0002
295	4	0.379	± 0.0001
295	5	0.415	± 0.0002
375	1	0.617	± 0.0003
375	2	0.603	± 0.0003
375	3	0.609	± 0.0003
375	4	0.567	± 0.0003
375	5	0.583	± 0.0003
375	6	0.590	± 0.0003
445	1	0.680	± 0.0005
445	2	0.716	± 0.0004
445	3	0.658	± 0.0004
445	4	0.784	± 0.0004
445	5	0.612	± 0.0003
445	6	0.698	± 0.0004

Table 6.5 One-dimensional effective volumetric heat capacity estimates with insulated top boundary.

Temperature (K)	Experiment Number	Volumetric Heat Capacity ($\times 10^5 J/m^3 K$)	95% Confidence Interval ($\times 10^5 J/m^3 K$)
295	4	4.719	± 0.008
295	5	2.424	± 0.003
295	6	3.379	± 0.004
295	7	3.612	± 0.004
295	8	3.358	± 0.004
375	1	3.195	± 0.005
375	2	3.415	± 0.005
375	3	3.220	± 0.005
375	4	3.865	± 0.006
375	5	3.200	± 0.005
375	6	3.463	± 0.006
445	1	4.447	± 0.008
445	2	3.147	± 0.006
445	3	3.998	± 0.007
445	4	3.103	± 0.007
445	5	4.343	± 0.007
445	6	3.436	± 0.006

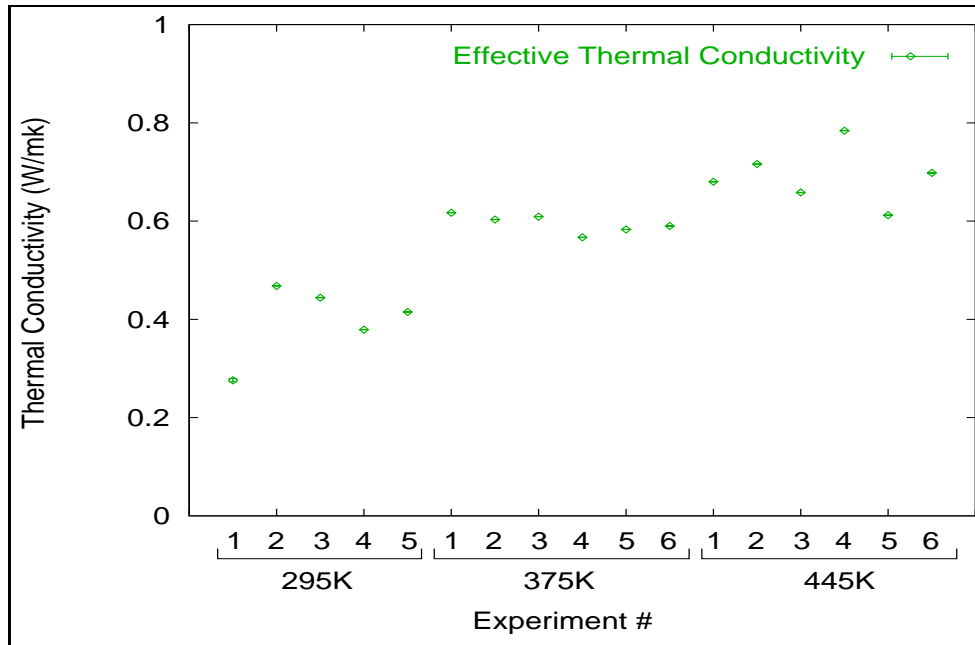


Figure 6.10 Thermal conductivity estimates with confidence intervals for the one-dimensional effective model with insulated top boundary at surrounding temperatures of 295K, 375K, and 445K.

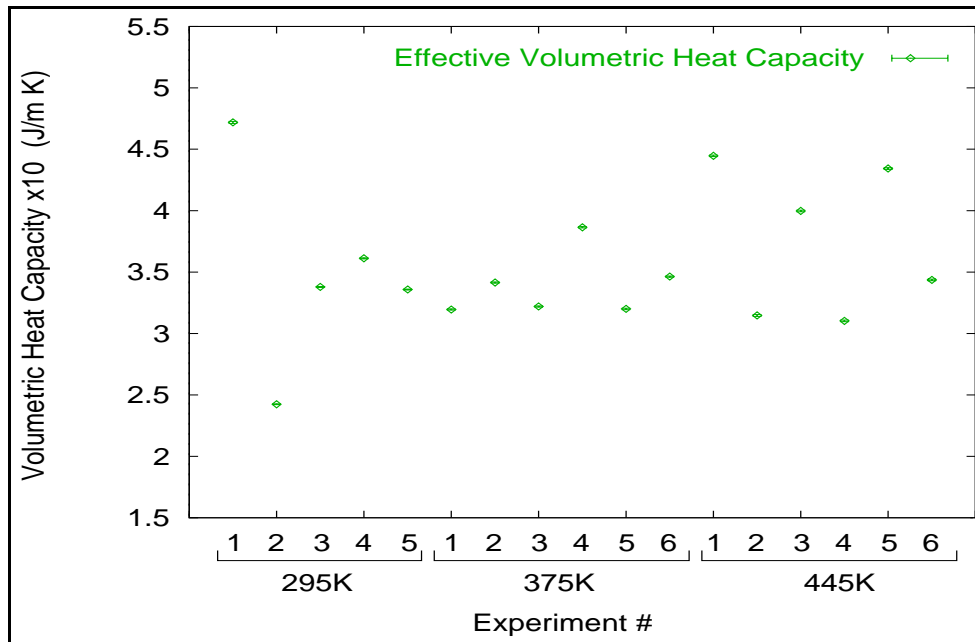


Figure 6.11 Volumetric heat capacity estimates with confidence intervals for the one-dimensional effective model with insulated top boundary at surrounding temperatures of 295K, 375K, and 445K.

Table 6.6 Average one-dimensional effective thermal conductivity estimates with insulated top boundary.

Temperature (K)	Thermal Conductivity (W/mK)	95% Confidence Interval (W/mK)
295	0.396	± 0.052
375	0.595	± 0.007
445	0.691	± 0.011

Table 6.7 Average one-dimensional effective volumetric heat capacity estimates with insulated top boundary.

Temperature (K)	Volumetric Heat Capacity ($\times 10^5 J/m^3 K$)	95% Confidence Interval ($\times 10^5 J/m^3 K$)
295	3.499	± 0.155
375	3.393	± 0.138
445	3.746	± 0.178

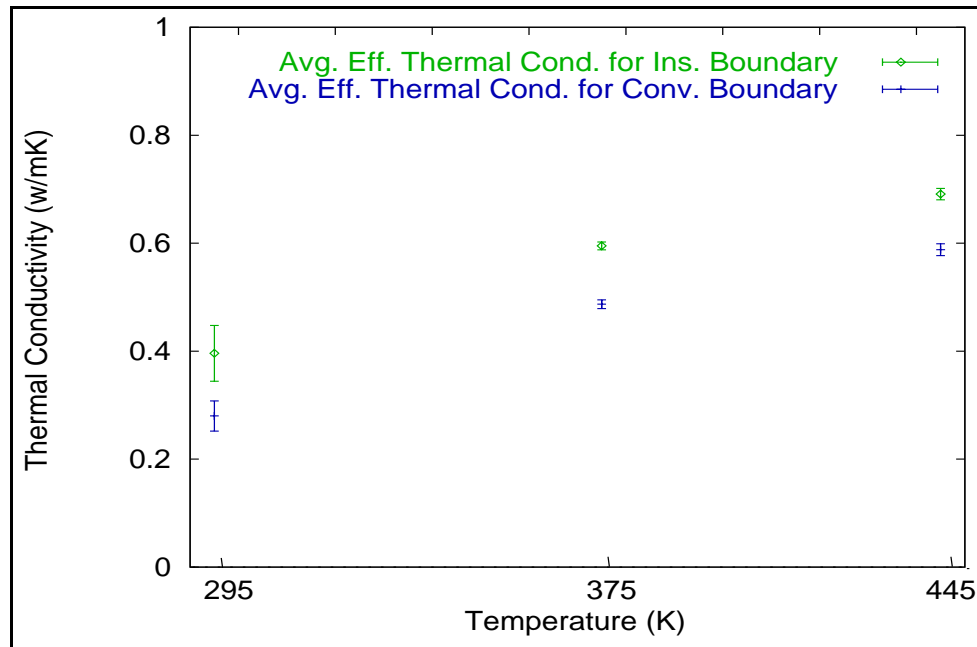


Figure 6.12 Average thermal conductivity estimates with confidence intervals for the one-dimensional effective model with insulated and convective boundary at surrounding temperatures of 295K, 375K, and 445K.

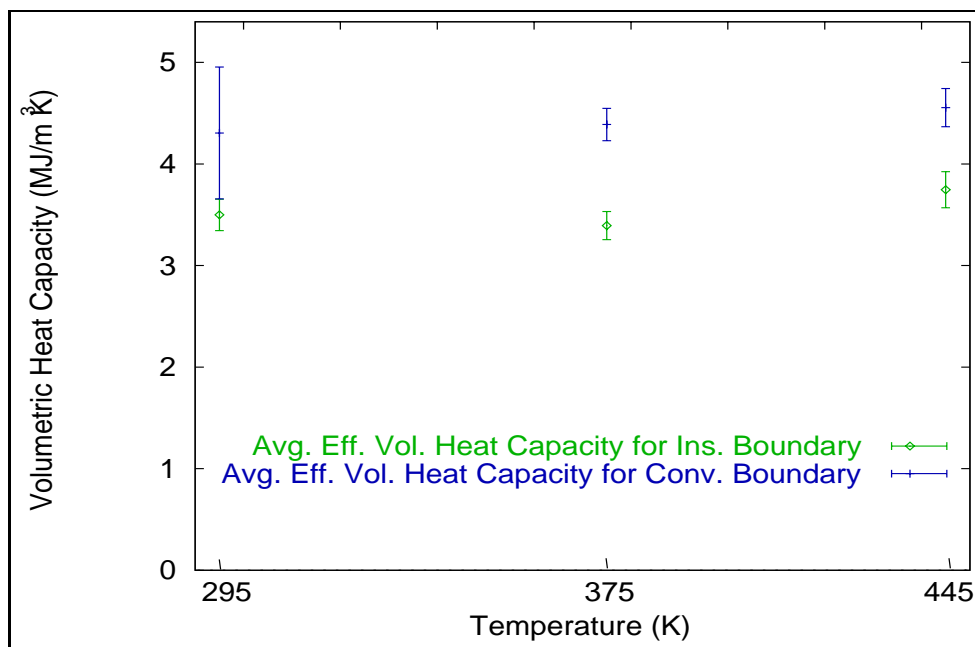


Figure 6.13 Average Volumetric heat capacity estimates with confidence intervals for the one-dimensional effective model with insulated and convective boundary at surrounding temperatures of 295K, 375K, and 445K.

surrounding temperature and not with respect of the 10°C temperature change within each surrounding temperature setting. Fig. 6.14 reveals a slight temperature dependence of the thermal conductivity, whose magnitude increases as the surrounding temperature is raised. The temperature dependence of the thermal conductivity is also supported by a student's *t* means test which was applied to the results of Table 6.8. Experiments at room temperature, 295K, did not yield very consistent estimates, however, at surrounding temperatures of 375K and 445K, the estimates were consistent with each other.

The estimates for the volumetric heat capacity range between 0.4 and 0.5 MJ/m^3K . No temperature dependence is visible from Fig. 6.15 or Fig. 6.13. Here again, statistical analyses showed that the means were not significantly different from one surrounding temperature setting to another, and therefore no temperature dependence can be deducted.

Table 6.8 One-dimensional effective thermal conductivity estimates with convective top boundary.

Temperature (K)	Experiment Number	Thermal Conductivity (W/mK)	95% Confidence Interval (W/mK)
295	4	0.212	± 0.0006
295	5	0.363	± 0.0002
295	6	n/a	n/a
295	7	0.266	± 0.0002
295	8	n/a	n/a
375	1	0.508	± 0.0003
375	2	0.497	± 0.0003
375	3	0.508	± 0.0003
375	4	0.455	± 0.0003
375	5	0.471	± 0.0003
375	6	0.480	± 0.0003
445	1	0.573	± 0.0005
445	2	0.616	± 0.0004
445	3	0.553	± 0.0004
445	4	0.688	± 0.0004
445	5	0.508	± 0.0004
445	6	0.590	± 0.0004

Table 6.9 One-dimensional effective volumetric heat capacity estimates with convective top boundary.

Temperature (K)	Experiment Number	Volumetric Heat Capacity ($\times 10^5 J/m^3 K$)	95% Confidence Interval ($\times 10^5 J/m^3 K$)
295	4	4.988	± 0.015
295	5	3.173	± 0.003
295	6	n/a	n/a
295	7	4.753	± 0.005
295	8	n/a	n/a
375	1	4.195	± 0.005
375	2	4.424	± 0.006
375	3	4.109	± 0.005
375	4	4.966	± 0.006
375	5	4.235	± 0.005
375	6	4.340	± 0.006
445	1	4.963	± 0.008
445	2	4.047	± 0.006
445	3	4.917	± 0.008
445	4	4.057	± 0.007
445	5	4.963	± 0.008
445	6	4.378	± 0.006

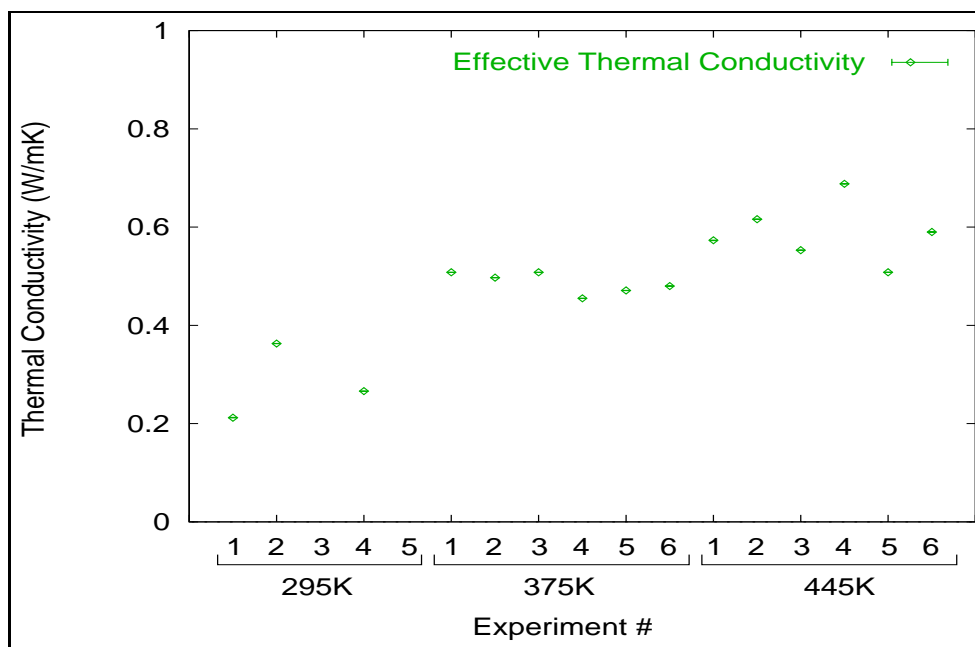


Figure 6.14 Thermal conductivity estimates with confidence intervals for the one-dimensional effective model with convective top boundary at surrounding temperatures of 295K, 375K, and 445K.

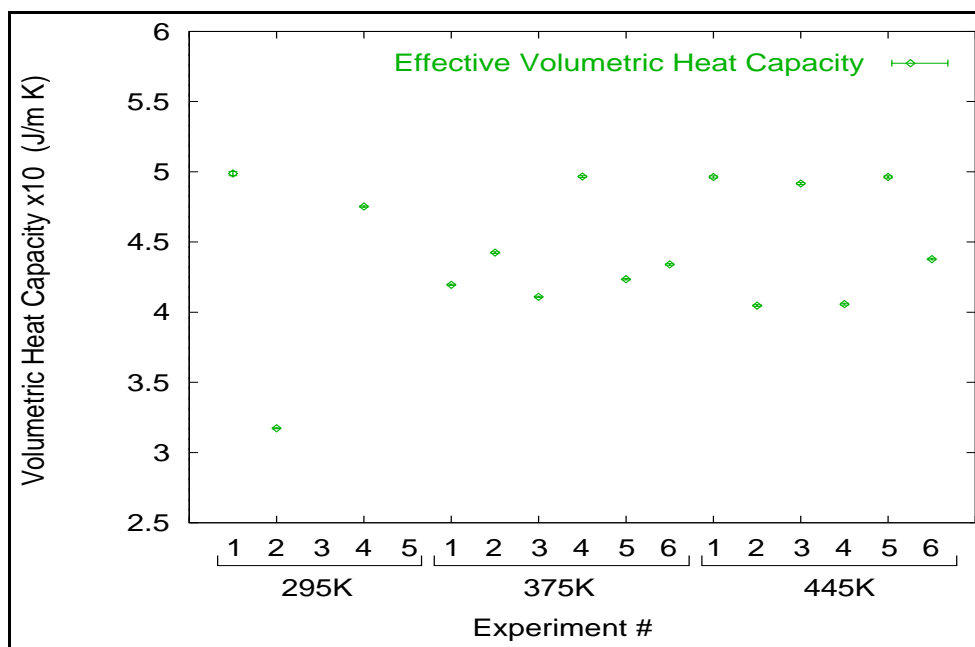


Figure 6.15 Volumetric heat capacity estimates with confidence intervals for the one-dimensional effective model with convective top boundary at surrounding temperatures of 295K, 375K, and 445K.

Table 6.10 Average one-dimensional effective thermal conductivity estimates with convective top boundary.

Temperature (K)	Thermal Conductivity (W/mK)	95% Confidence Interval (W/mK)
295	0.280	± 0.028
375	0.487	± 0.008
445	0.588	± 0.011

Table 6.11 Average one-dimensional effective volumetric heat capacity estimates with convective top boundary.

Temperature (K)	Volumetric Heat Capacity ($\times 10^5 J/m^3 K$)	95% Confidence Interval ($\times 10^5 J/m^3 K$)
295	4.305	± 0.650
375	4.388	± 0.160
445	4.554	± 0.188

Comparison of the Thermal Property Estimates Between Insulated and Convective Boundary Condition

When comparing both the insulated and convective boundary models, there were no major differences in the trends the estimates described. A slight temperature dependence was revealed for the effective thermal conductivity for both boundary conditions as shown in Fig. 6.12, whereas the effective volumetric heat capacity stayed fairly constant over the 150K temperature span as shown in Fig. 6.13. However, the magnitude of the estimates do not agree with each other from one model to the other. The values of the estimated thermal conductivity of the convective boundary model are lower than the ones of the insulated boundary model as one would expect due to the loss of heat to the surroundings. The same argument explains the higher values of the volumetric heat capacity of the convective boundary model over the insulated boundary model.

An important issue needs to be addressed concerning the adequacy of the boundary condition. At room temperature, the thermotron was not used to circulate the air in the oven, and therefore only natural convection over the insulation panels was possible. At elevated surrounding temperatures (375K and 445K) however, the thermotron kept the air circulating around the test panel in the oven to assure constant surrounding temperature conditions, thus creating forced convection over the insulation boards. In order to determine the effect of the heat transfer coefficient used for the convective boundary condition, several

cases with different values of the heat transfer coefficient were investigated. The results are summarized in Table 6.12.

Table 6.12 shows that for a particular dataset, a significant increase in the value of the heat transfer coefficient does not affect the estimates or the amount of heat lost due to convection. The objective function values are consistent and the amount of heat lost to the surroundings increases only slightly as the value of the heat transfer coefficient is raised. This proves that the amount of insulation used to run the experiments was not adequate since about 10 % of the total heat is lost to the surroundings. At steady-state and room temperature, the heat transfer coefficient was calculated to be around $3.4 \text{ W/m}^2\text{K}$, and therefore we anticipate the values for the convection coefficient to increase as the surrounding temperature is raised and forced convection applied. It is therefore appropriate to use a convective boundary condition for the remaining one-dimensional detailed model and the two-dimensional model.

Overall, the estimates for the thermal conductivity agree with the estimates obtained previously from the one-dimensional steady-state thermal resistance analysis, for which the estimates revolved around 0.23 (W/mK) , which increases the confidence in the results.

The confidence intervals corresponding to the insulated boundary model are slightly lower than the ones for the convective boundary model, indicating that the insulated model yields estimates that are more accurate. This was predicted by the higher sensitivity coefficients for the thermal conductivity and volumetric heat capacity of the insulated case as shown in Fig. 6.6.

Table 6.12 Effect of the heat transfer coefficient on the estimated parameters at 375K.

Heat Transfer Coefficient ($\text{W/m}^2\text{K}$)	Estimated Parameters		Objective Function (K^2)	Heat Lost (W/m^2)	Heat Used (W/m^2)
	Thermal Conductivity (W/mK)	Volumetric Heat Capacity ($\times 10^5 \text{ J/m}^3\text{K}$)			
2.37 (natural conv.)	0.508	4.195	58.42	1.67	15.44
10.0 (forced conv.)	0.483	4.196	62.21	1.94	14.66
20.0 (forced conv.)	0.489	4.133	60.54	1.99	14.86
30.0 (forced conv.)	0.486	4.146	60.72	2.01	14.75

Residuals

Once the estimates of the thermal conductivity and volumetric heat capacity were obtained, the values were entered into the one-dimensional effective forward model for the insulated and convective boundary case to calculate the corresponding temperature history at the most sensitive node location and compared to the experimental temperature history at that same location. Figs. 6.16 and 6.17 represent typical calculated and experimental temperature histories for the insulated and convective boundary condition models respectively, using the estimates for the thermal properties. The resulting standard deviations, calculated using Eq. 3.11, for the estimated designs are summarized in Tables 6.13 and 6.14 for both the insulated and convective boundary condition models respectively. The standard deviations for the insulated boundary condition are only slightly lower than the ones for the convective boundary condition.

It is difficult to determine any bias in the models from the foregoing temperature plots and therefore it is necessary to observe the residuals between the experimental and theoretical temperature distributions. Figs. 6.18 and 6.19 show the temperature residuals between the experimental and theoretical temperature histories for both the insulated and convective boundary models. The two residual plots show that the magnitude of the residuals is the highest when the heater is turned on and later when it is turned off. Therefore, a small bias is present at both heater turn on and turn off. However, the maximum magnitude of the residuals does not exceed 0.3 degrees, which represents 8% of the total temperature change. It is important to realize that for each temperature setting, the thermal properties are assumed constant and therefore do not vary in the x-direction. This translates into both one-dimensional effective models applying a linear temperature profile through the thickness of the structure at steady-state which, with the many different materials and components, has an overall highly nonlinear temperature profile. Thus the estimates of the volumetric heat capacity and thermal conductivity perform adequately.

However, since the maximum magnitude of the temperature residuals for the insulated and convective boundary cases are very similar, it is difficult to determine if either boundary condition yields better estimates. In addition, the objective functions listed in Tables 6.13 and 6.14 are very close from one boundary condition to the other. Although the confidence intervals for the insulated case are slightly smaller and the magnitude of

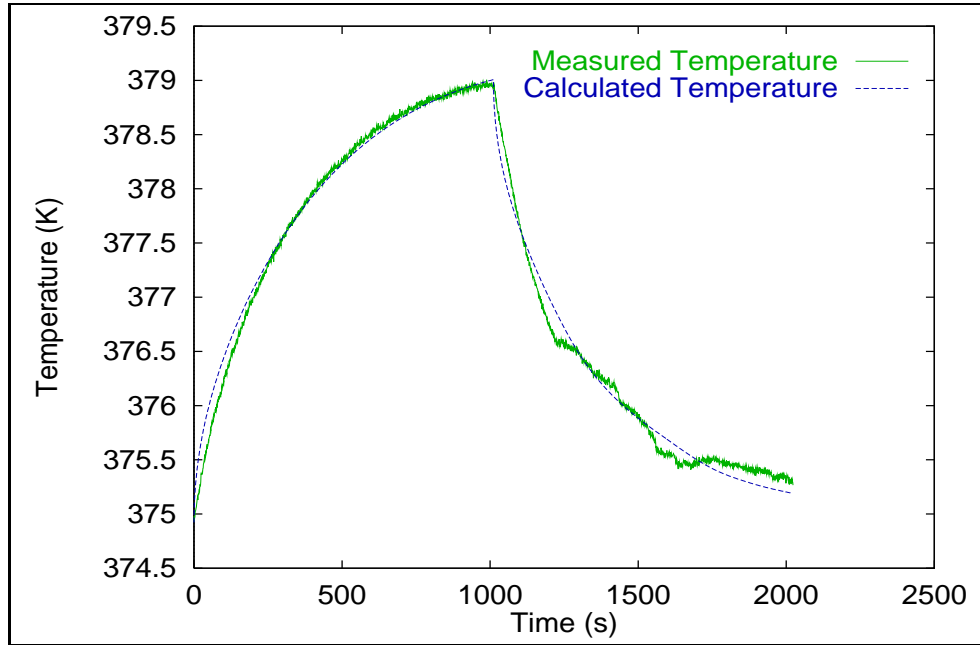


Figure 6.16 Experimental and calculated temperature histories for the one-dimensional effective model with insulated boundary condition using the thermal property estimates.

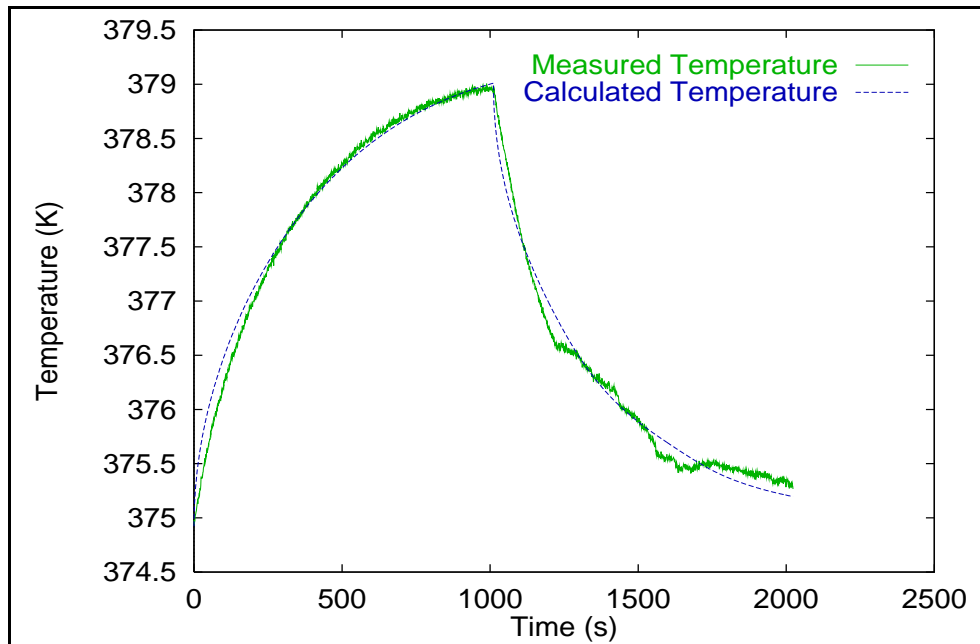


Figure 6.17 Experimental and calculated temperature histories for the one-dimensional effective model with convective boundary condition using the thermal property estimates.

Table 6.13 Sum-of-squares objective functions of estimated designs for the one-dimensional effective model with insulated boundary condition.

Temperature (K)	Experiment #	Objective Function	Number of Observations	Standard Deviation (s)
295	4	48.22	2026	0.154
	5	234.78	2017	0.341
	6	79.82	2056	0.197
	7	62.00	2126	0.171
	8	49.60	2056	0.155
295	average	94.88	n/a	0.215
375	1	55.43	2044	0.165
	2	48.04	2033	0.154
	3	55.69	2068	0.164
	4	28.47	2024	0.119
	5	65.71	2016	0.181
	6	44.08	2052	0.147
375	average	49.57	n/a	0.156
445	1	45.17	2038	0.149
	2	26.78	1817	0.121
	3	28.73	2040	0.119
	4	53.66	1918	0.167
	5	34.18	2101	0.128
	6	28.08	2042	0.117
445	average	36.10	n/a	0.135

Table 6.14 Sum-of-squares objective functions of estimated designs for the one-dimensional effective model with convective boundary condition.

Temperature (K)	Experiment #	Objective Function	Number of Observations	Standard Deviation (s)
295	4	321.53	2026	0.398
	5	273.86	2017	0.369
	6	n/a	2056	n/a
	7	130.08	2126	0.247
	8	n/a	2056	n/a
295	average	241.82	n/a	0.343
375	1	58.42	2044	0.169
	2	52.87	2033	0.161
	3	59.50	2068	0.170
	4	31.15	2024	0.124
	5	74.74	2016	0.193
	6	49.56	2052	0.155
375	average	54.37	n/a	0.163
445	1	59.58	2038	0.171
	2	24.84	1817	0.117
	3	33.08	2040	0.127
	4	52.53	1918	0.166
	5	48.58	2101	0.152
	6	27.25	2042	0.116
445	average	40.96	n/a	0.144

the estimates quite different, a comparison of the objective functions does not reflect an obvious superior boundary condition to model the physics over the insulation boards of the top surface of the outer wing subcomponent. Therefore, the convective boundary condition was assumed to represent the physics of the problem more realistically and was kept to estimate the remaining properties.

6.4.2 One-Dimensional Detailed Estimation Results

Once the thermal properties of the overall structure were defined by estimating the effective thermal conductivity and volumetric heat capacity, it was necessary to evaluate the thermal properties of the individual materials used in the fabrication of the outer wing subcomponent. The detailed one-dimensional half-model was thus used to determine the the volumetric heat capacity and the through the thickness thermal conductivity of the facesheets, the honeycomb, the skins and the torque tubes. The following section describes the estimation results obtained for the thermal properties of the independent materials used in the construction of the outer wing subcomponent.

Thermal Property Estimates with Confidence Intervals

The results for the estimates of the thermal conductivity are summarized in Table 6.16 and illustrated in Fig. 6.20.

Fig. 6.20 shows very inconsistent results for the estimates of the thermal conductivity of the facesheets, the skins and the torque tubes, whereas the estimates for the honeycomb blanket fall within a window ranging between 0.176 and 0.719 W/mK . The same behavior is described by the averages of the estimates for each surrounding temperature, shown in Fig. 6.22. The spread in the estimates can directly be linked to the sensitivity coefficients of the different parts of the structure. As observed earlier, the majority of the sensitivity coefficients for the thermal conductivity are highly correlated, making the estimation process very difficult. In addition, the magnitude of mostly all the sensitivity coefficients for the thermal conductivity are very small except for the honeycomb blanket which is reflected in the estimation results.

Similar estimation trends were obtained for the volumetric heat capacity of the facesheets, the skins, the torque tubes and the honeycomb blanket, as shown in Table 6.15 and Fig. 6.21. The best individual estimates were found for the honeycomb blanket. How-

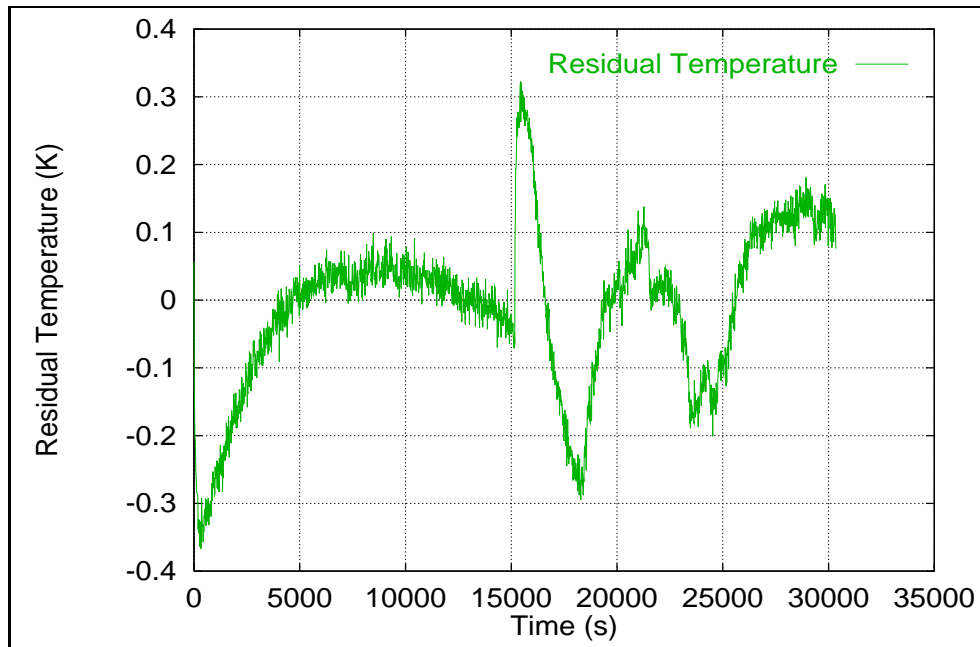


Figure 6.18 Residuals between experimental and calculated temperature histories for the one-dimensional effective model with insulated boundary condition.

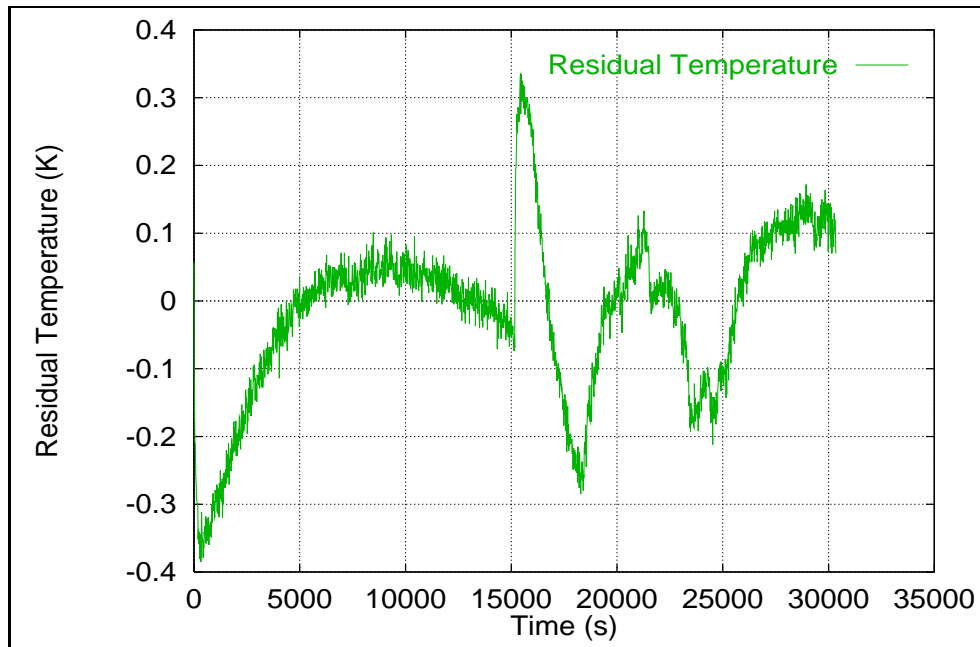


Figure 6.19 Residuals between experimental and calculated temperature histories for the one-dimensional effective model with convective boundary condition.

ever, Fig. 6.23 which represents the averages of the volumetric heat capacities for each material at the various surrounding temperatures, shows that the average estimates for the facesheets, the skins, the torque tubes, and the honeycomb are fairly consistent over the surrounding temperature range of 150K.

A shocking observation was made when the confidence intervals were calculated. The 95% confidence intervals were evaluated using Eq. 3.8 for the independent confidence intervals and Eq. 3.9 for the 95% confidence interval of the average estimates. Since the values seemed extremely high, a step by step approach was taken to isolate the existing problem, if any. It was discovered that the off-diagonal terms of the P matrix, the covariance matrix $((\mathbf{X}^T \mathbf{X})^{-1})$, were very large, sometimes 3 or 4 orders of magnitude larger than the diagonal terms themselves, indicating a high degree of correlation. Therefore, the large confidence intervals are a direct consequence of the small sensitivity coefficients for the different materials and the correlation between them.

Residuals

The sum-of-squares objective functions for the designs obtained during the estimation process of the one-dimensional detailed half-model were extremely low as shown in Table 6.17, yielding small standard deviations, and therefore small residuals. This translates into having a good match between the experimental and calculated temperature histories. Although the estimates for the thermal conductivity and volumetric heat capacity are not very consistent for the facesheets, the skins, and the torque tubes, due to the small sensitivity coefficients, the outer wing subcomponent can successfully be modeled without knowing the exact values of the thermal properties of the materials of these components. However it is important to obtain accurate results for the thermal conductivity and volumetric heat capacity of the honeycomb blanket since the honeycomb has a dominant effect on the thermal response of the outer wing subcomponent as reflected in the magnitude of its sensitivity coefficients. Accurate results for the honeycomb blanket's thermal properties is therefore absolutely necessary to accurately determine the thermal characteristics of the entire structure, whereas the accuracy of the thermal properties of the materials of the remaining components is less crucial to correctly model the outer wing subcomponent.

A typical comparison between the experimental and calculated temperature histories is shown in Fig. 6.24 with its corresponding residuals in Fig. 6.25. The magnitude of the

Table 6.15 One-dimensional thermal conductivity estimates with confidence intervals with convective top boundary for skins, honeycomb, torque tubes, and facesheets.

Temperature (K)	Experiment #	Thermal Conductivity (W/mK)			
		Skins	Honeycomb	Torque Tubes	Facesheets
295	4	2.177±2.7E4	0.176±1.0E2	1.686±3.8E5	3.963±3.8E2
	5	2.137±7.1E4	0.243±0.7E2	1.842±1.7E5	0.980±1.5E1
	6	n/a	n/a	n/a	n/a
	7	3.723±1.6E6	0.225±0.7E2	3.478±3.3E6	1.168±2.2E1
	8	n/a	n/a	n/a	n/a
295	ave	2.679±6.7E7	0.215±5.6E3	2.335±1.4E8	2.034±1.5E4
375	1	3.383±7.1E5	0.346±3.1E2	0.515±3.1E4	0.308±7.7E0
	2	0.183±1.5E2	0.528±1.2E2	0.338±6.4E2	0.353±6.0E1
	3	0.166±4.4E2	0.719±0.7E2	2.765±2.9E5	0.825±5.3E2
	4	0.166±2.7E2	0.422±1.7E2	3.862±4.2E5	2.517±1.1E2
	5	0.183±4.2E2	0.426±1.2E2	3.828±5.1E5	0.477±8.2E0
	6	0.166±3.3E2	0.353±4.9E1	2.541±2.0E5	4.606±5.3E2
375	ave	0.708±7.7E7	0.466±4.4E3	2.308±8.1E6	1.514±8.3E3
445	1	3.290±1.8E5	0.428±3.8E2	1.352±1.0E5	3.370±3.1E2
	2	2.649±2.0E5	0.373±4.5E2	2.432±1.2E5	0.356±3.3E2
	3	0.166±2.2E2	0.428±1.1E2	0.495±4.0E3	4.746±2.9E2
	4	1.362±5.3E4	0.452±2.1E2	0.353±6.6E3	0.176±2.2E0
	5	0.166±3.5E2	0.526±1.3E2	2.437±1.9E5	0.353±5.0E0
	6	3.477±9.1E5	0.453±5.1E2	0.835±1.0E5	0.192±1.4E1
445	ave	1.852±1.0E7	0.443±8.7E3	1.317±2.9E6	1.532±5.6E3

Table 6.16 One-dimensional volumetric heat capacity estimates with confidence intervals with convective top boundary for skins, honeycomb, torque tubes, and facesheets.

Temperature (K)	Experiment #	Volumetric Heat Capacity (MJ/m^3K)			
		Skins	Honeycomb	Torque Tubes	Facesheets
295	4	0.727±1.2E5	1.012±5.8E2	3.260±9.3E4	1.281±4.9E0
	5	1.506±1.0E5	0.344±1.0E2	4.831±1.4E5	2.282±9.6E0
	6	n/a	n/a	n/a	n/a
	7	1.434±5.6E5	0.888±2.6E2	0.915±1.7E6	1.160±5.6E0
	8	n/a	n/a	n/a	n/a
295	ave	1.222±2.5E7	0.748±2.6E4	3.002±7.1E7	1.575±4.9E2
375	1	1.694±1.6E5	0.964±8.6E2	4.546±4.2E5	1.530±3.3E1
	2	1.086±1.8E3	0.659±1.4E2	4.862±2.2E3	1.089±1.8E2
	3	1.141±3.1E3	0.644±6.2E1	1.076±9.5E4	0.520±3.5E2
	4	1.141±1.8E3	0.490±2.0E2	3.632±3.8E5	1.893±1.2E1
	5	1.793±4.2E3	0.675±2.0E2	3.915±4.7E5	1.432±1.7E1
	6	1.141±2.4E3	0.826±1.1E2	1.814±1.3E5	0.862±1.2E1
375	ave	1.333±1.8E6	0.710±1.0E4	3.307±8.1E6	1.221±4.3E3
445	1	0.425±5.3E4	0.128±1.3E2	2.163±1.2E5	4.513±0.9E2
	2	0.937±6.3E4	1.204±1.6E2	2.037±1.4E5	1.447±1.0E3
	3	1.141±2.2E3	0.644±1.6E2	2.981±1.2E4	3.007±1.7E1
	4	1.962±3.7E4	0.544±2.5E2	2.550±1.1E5	4.729±5.0E1
	5	0.288±1.9E3	0.644±1.6E2	2.981±2.4E5	2.795±2.8E1
	6	1.117±1.7E5	0.892±1.0E3	2.163±4.4E5	1.976±1.4E2
445	ave	0.978±2.0E6	0.676±1.2E4	2.480±6.0E6	3.078±2.2E3

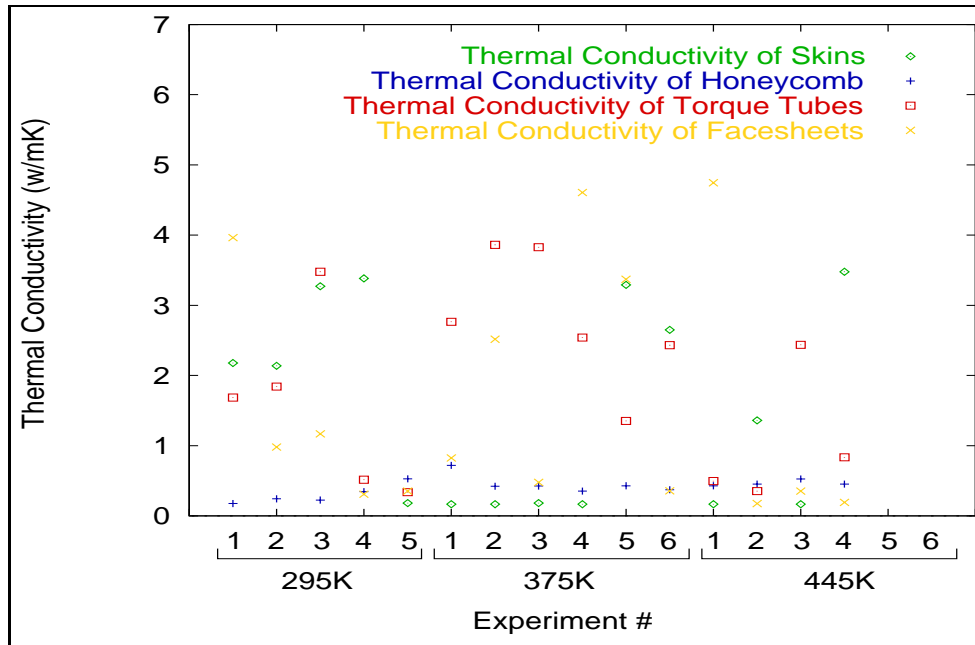


Figure 6.20 Thermal conductivity estimates for the one-dimensional detailed model with convective top boundary at surrounding temperatures of 295K, 375K, and 445K.

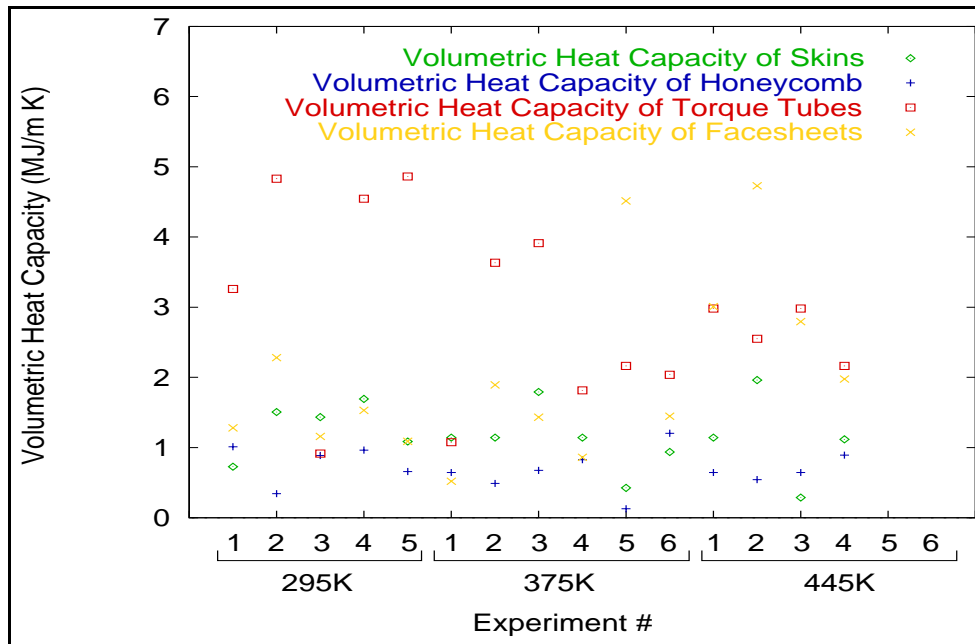


Figure 6.21 Volumetric heat capacity estimates for the one-dimensional detailed model with convective top boundary at surrounding temperatures of 295K, 375K, and 445K.

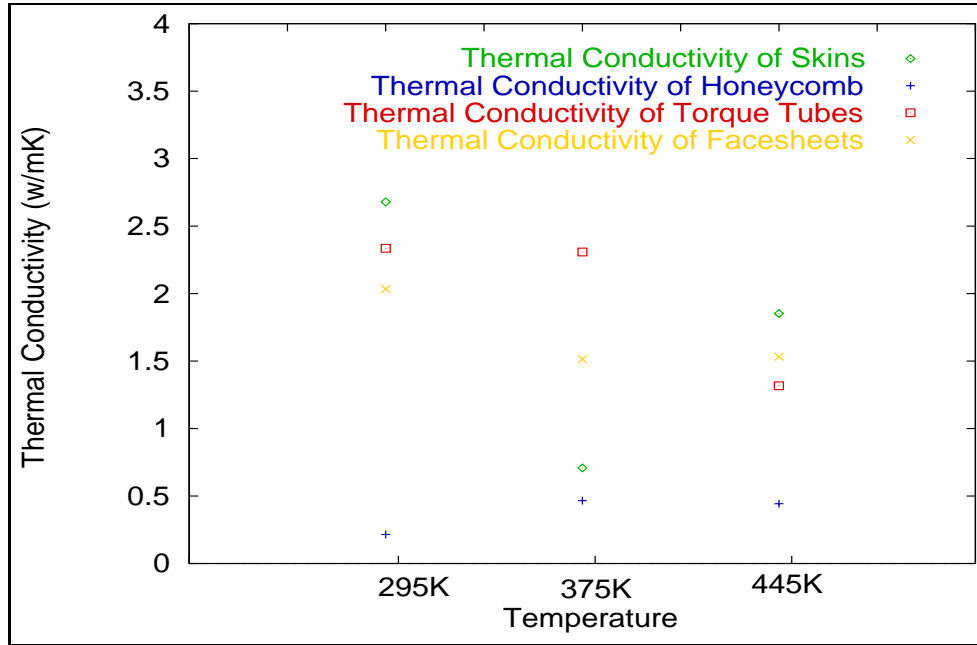


Figure 6.22 Average thermal conductivity estimates for the one-dimensional detailed model with convective top boundary at surrounding temperatures of 295K, 375K, and 445K.

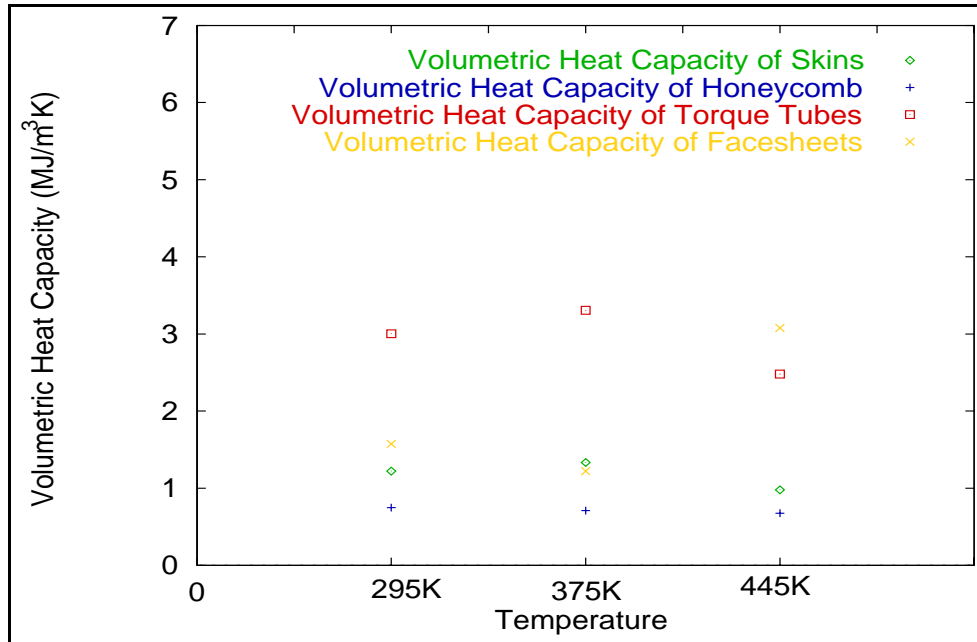


Figure 6.23 Average volumetric heat capacity estimates for the one-dimensional detailed model with convective top boundary at surrounding temperatures of 295K, 375K, and 445K.

residuals are maximum during heater turn on and heater turn off, as for the residuals of the effective models.

6.4.3 Validation of the One-Dimensional Models

One aspect that needs to be addressed when looking at the one-dimensional thermal properties of a structure with such important geometrical variations is the determination of whether the estimates obtained using the average temperature histories of the top thermocouples are consistent with the estimates resulting from the use of independent thermocouple couples. An independent thermocouple couple consists of two thermocouples, one of them located at the heated surface (top surface), the other situated at the corresponding location on the known temperature boundary (bottom surface). From Fig. 4.11, an example of a thermocouple couple would be thermocouple #1 on the bottom and #17 on top. Table 6.18 summarizes the estimates obtained for the one-dimensional effective insulated model, at each surrounding temperature setting, using the average of the temperature histories of the top thermocouples and the bottom thermocouples, and independent temperature histories from the independent thermocouple couples for each a dataset.

As shown in Table 6.18, the average of the estimates obtained using the temperature histories from the thermocouple couples is very close to the estimates obtained using the average temperature history. The standard deviations of the two averages are very close and therefore it was felt that the assumption of using the average temperature history of the top thermocouples as the experimental temperature history in the definition of the objective function in the previous estimates was valid.

6.5 Convergence of the Genetic Algorithm

In many estimation techniques, the convergence criterion is an essential and crucial part of the procedure to successfully determine if the minimum of the objective function has been reached. Finding the minimum of the objective function translates into finding the optimal design and therefore, in our case, the estimates that yield a temperature distribution that most closely matches the experimental data. Many examples involving optimization and estimation using genetic algorithms can be found in the literature for which the termination criterion is specified by a fixed number of generations instead of a convergence criterion.

Table 6.17 Sum-of-squares objective functions of estimated designs for the one-dimensional detailed half-model.

Temperature (K)	Experiment #	Objective Function	Number of Observations	Standard Deviation (s)
295	4	17.06	2026	0.092
	5	16.25	2017	0.090
	6	n/a	2056	n/a
	7	31.98	2126	0.125
	8	n/a	2056	n/a
375	1	12.31	2044	0.078
	2	13.54	2033	0.082
	3	21.48	2068	0.102
	4	10.51	2024	0.072
	5	11.09	2016	0.074
	6	14.68	2052	0.085
445	1	20.65	2038	0.101
	2	10.75	1817	0.077
	3	10.03	2040	0.070
	4	14.41	1918	0.087
	5	37.67	2101	0.134
	6	18.96	2042	0.096

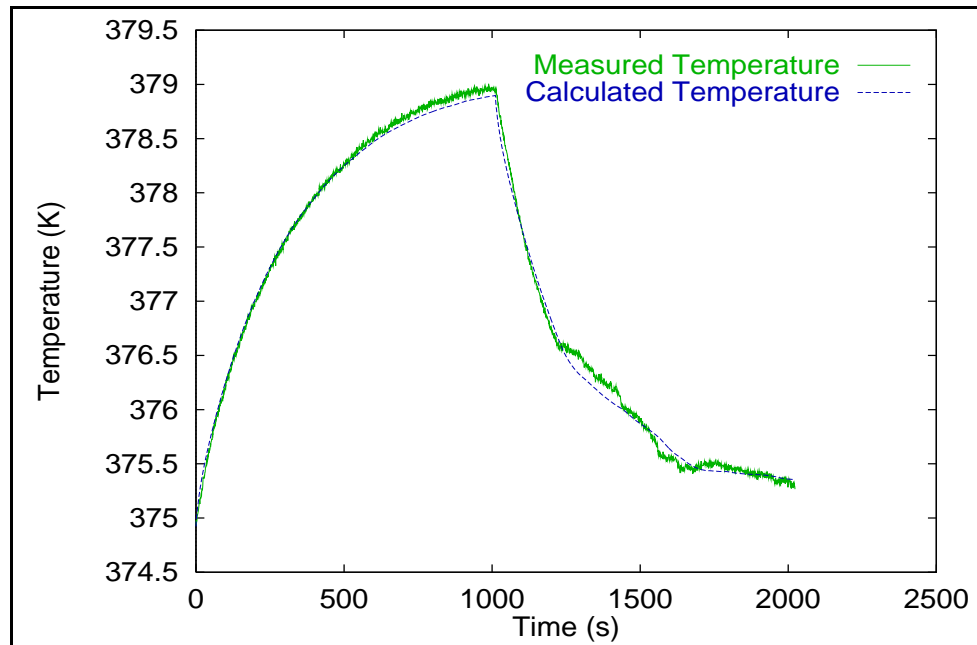


Figure 6.24 Experimental and calculated temperature histories for the one-dimensional detailed model using the thermal property estimates.

Table 6.18 One-dimensional effective thermal conductivity estimates with average and independent temperature histories.

Temperature (K)	Thermal Conductivity (W/mK)	Volumetric Heat Capacity ($\times 10^5 J/m^3 K$)	Objective Function (K^2)	Standard Deviation (K)
295 (8.01.17)	0.418	2.886	41.17	0.142
295 (8.10.26)	0.435	3.234	80.30	0.198
295 (8.12.28)	0.403	3.173	43.41	0.145
295 (8.16.32)	0.407	3.530	68.23	0.182
295 (8.05.21)	0.403	3.354	65.91	0.179
295 (8.00.00)	0.413	3.235	59.80	0.170
295 (average)	0.415	3.358	49.60	0.155
375 (1.01.17)	0.639	2.791	43.25	0.146
375 (1.10.26)	0.609	3.168	70.15	0.185
375 (1.12.28)	0.624	3.003	42.27	0.144
375 (1.16.32)	0.593	3.783	77.81	0.195
375 (1.05.21)	0.590	3.579	88.94	0.209
375 (1.00.00)	0.611	3.265	64.48	0.178
375 (average)	0.617	3.195	55.43	0.165
445 (1.01.17)	0.664	4.198	34.45	0.130
445 (1.10.26)	0.699	4.461	70.66	0.186
445 (1.12.28)	0.646	4.286	44.98	0.149
445 (1.16.32)	0.702	4.703	116.77	0.239
445 (1.05.21)	0.672	4.545	52.93	0.161
445 (1.00.00)	0.676	4.439	63.96	0.177
445 (average)	0.680	4.447	45.17	0.149

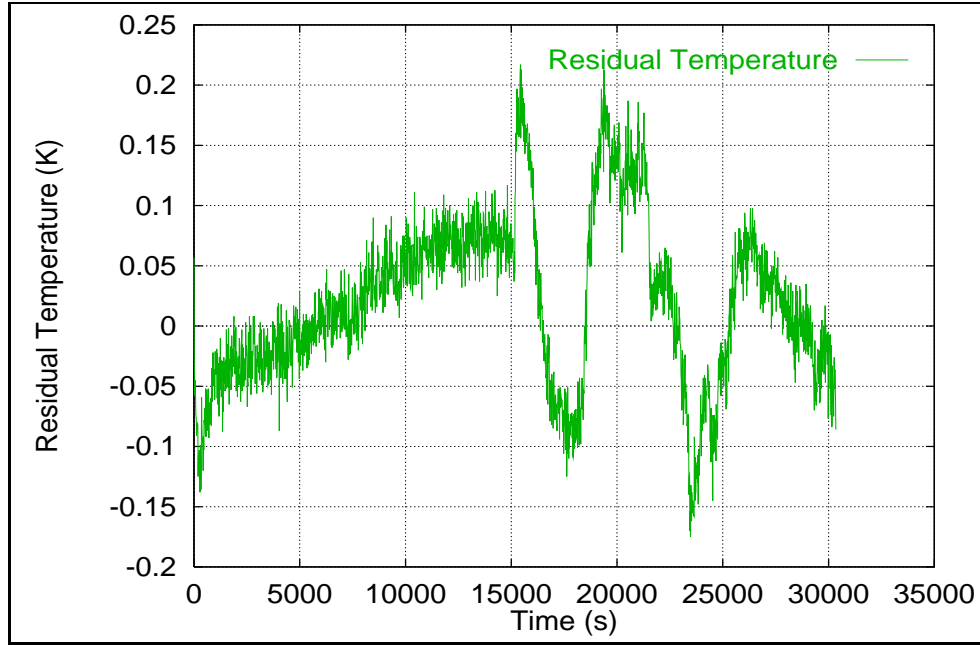


Figure 6.25 Residuals between experimental and calculated temperature histories for the one-dimensional detailed model.

As described in Chapter 5, a convergence criterion was implemented in this study to assure that the minimization procedure had converged. This section shows how the convergence criterion was implemented by the genetic algorithm.

Fig. 6.26 illustrates the convergence of a sample estimation run for the effective thermal conductivity and volumetric heat capacity. Each circle represents a chromosome with specific values for the thermal conductivity and volumetric heat capacity. The colors of the contours describe the magnitude of the various sum-of-squares objective functions corresponding to each individual chromosome, where the lighter colors represent larger objective functions. Fig. 6.26(a) represents an initial population, Fig. 6.26(b) shows the chromosome distribution for the elitist parent population, Fig. 6.26(c) corresponds to the distribution of the chromosomes after the third generation, whereas Fig. 6.26(d) corresponds to the distribution of the chromosomes after the tenth generation. The **Extended Elitist Genetic Algorithm (EEGA)** randomly picks values for the thermal conductivity and volumetric heat capacity from within the initial parameter range which, in this case, was set from 0.1 to 1.2 W/mK for the thermal conductivity and from 0.15 to 0.48 MJ/m^3K for the volumetric heat capacity, to form an initial population of 50 chromosomes (a). From the five

initial populations created, only the 50 chromosomes with the smallest objective functions are kept to create the elitist parent population (b). At that point, the genetic algorithm uses children generations, mutations, and cross-overs to enhance the performance of the population. Fig. 6.26 shows how the objective functions of later generations (c) and (d) converge to yield estimates with the lowest objective functions. After the tenth generation, the convergence has occurred for this specific case.

The convergence criterion implemented in this study takes into account the best 20 objective functions of each generation. Once the average of the 20 best objective functions stays within 1% of its value for three consecutive generations, the solution is said to be converged. Fig. 6.27 is an example for which the convergence of the objective function happened after only 24 generations. The curve in Fig. 6.27 flattens out as the solution reaches convergence. However, if a fixed number of generations had been used, say for example 10 generations, the solution would not have been converged, therefore yielding incorrect estimates for the thermal conductivity and volumetric heat capacity.

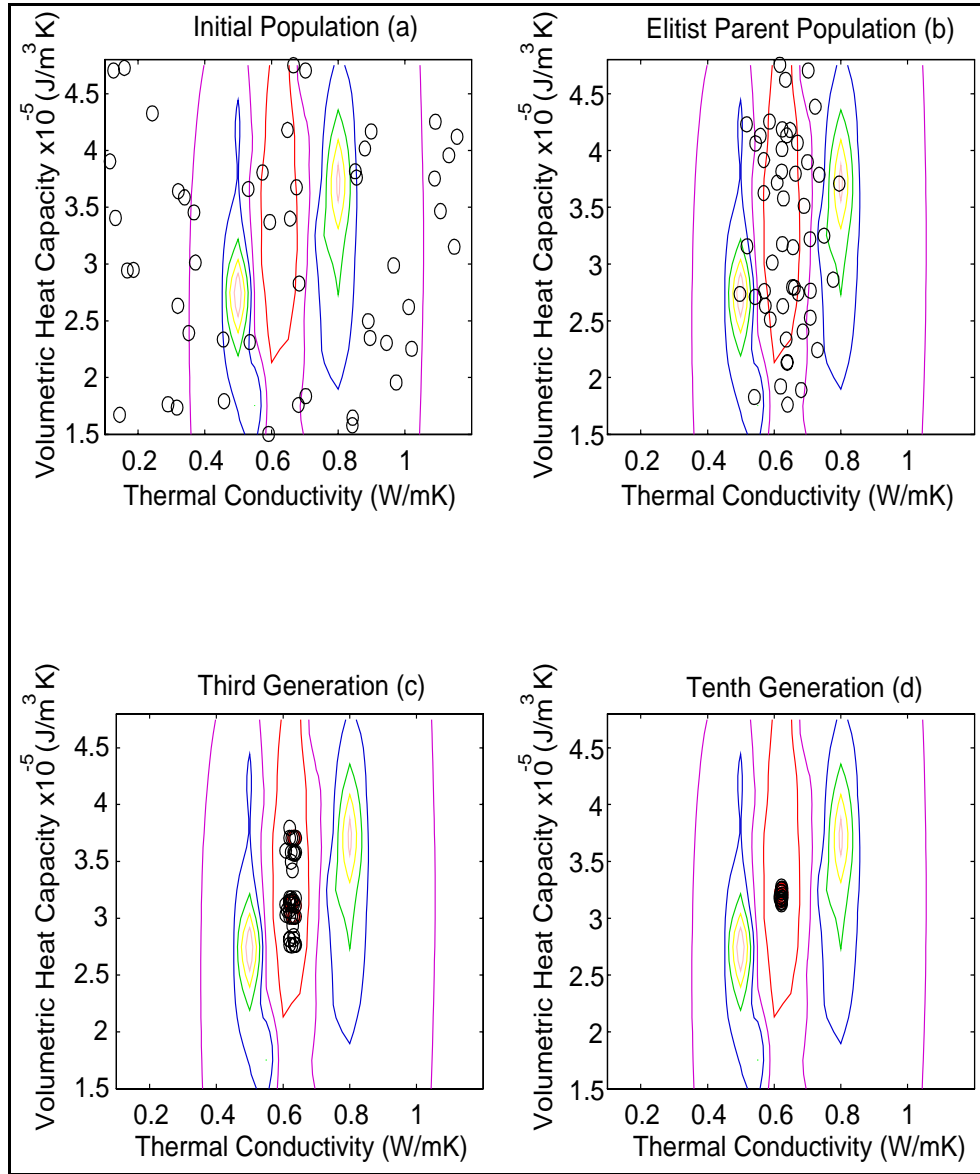


Figure 6.26 Convergence of the genetic algorithm.

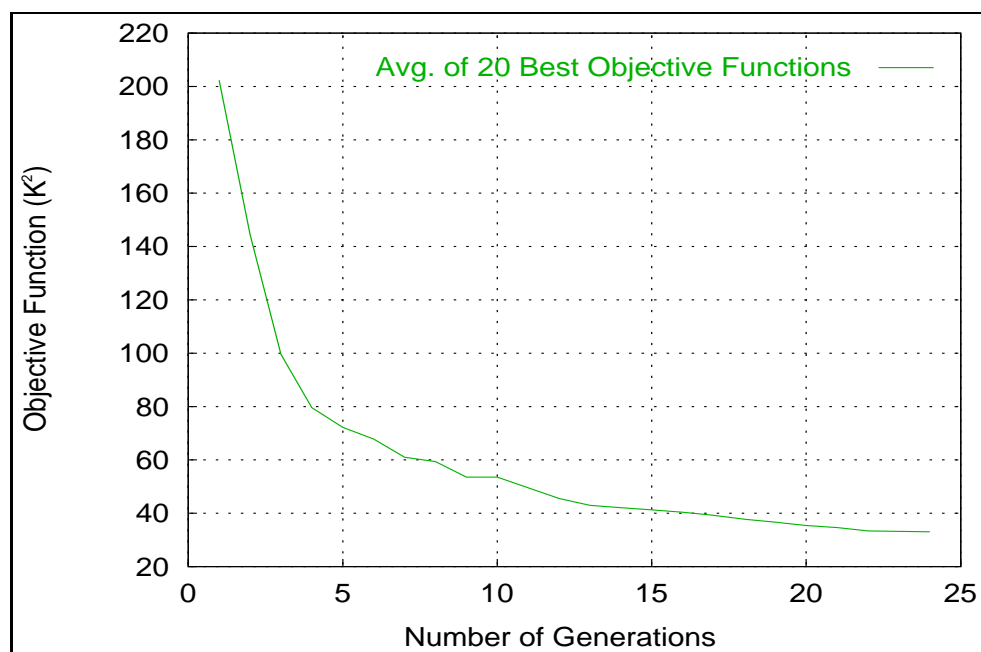


Figure 6.27 Convergence of the average of the 20 best objective functions.

Chapter 7

Summary and Conclusions

This chapter was dedicated to summarizing the procedures used to achieve the objectives set for this project. This chapter is divided in three sections. The first section summarizes the work accomplished throughout this study, the second section shows which and how the objectives were met, whereas the last section describes overall conclusions on the project.

7.1 Summary

The goal of this project was to develop a noninvasive parameter estimation technique to simultaneously evaluate the thermal properties of independent materials in complex structures, using as sample the entire structure. This work is a continuation of the previous studies performed by Moncman (1994), Hanak (1995), Copenhaver (1996), and Guynn (1996) who also developed parameter estimation procedures to estimate thermal properties of composites, honeycomb blankets, and insulative type materials. In this particular study, the test structure used to determine the thermal properties was a part of the trailing edge of the wing tip of the **H**igh **S**peed **C**ivil **T**ransport, called the outer wing subcomponent. The outer wing subcomponent consists of four major components including the outer facesheets, the honeycomb blanket, the skins, and the torque tubes, all of which were made of materials with different thermal properties. Several models were used to determine the thermal properties of the different materials used in the construction of the outer wing subcomponent.

An initial one-dimensional steady-state thermal resistance analysis was performed on a detailed model of the test structure to obtain a better understanding of the ther-

mal characteristics of the components of the outer wing subcomponent. The analysis also provided an initial estimate for the effective thermal conductivity which was later used to bound the original parameter space in the genetic algorithm during the estimation process. The thermal properties used to obtain an approximate value for the effective thermal conductivity of the outer wing subcomponent were taken from materials with similar thermal characteristics.

The estimation and optimization procedures implemented throughout this study consisted of the determination of the objective function and the minimization technique. The objective function was defined using the nonlinear regression technique which requires both experimental and theoretical data. The minimization was performed by a genetic algorithm which was introduced as several processors into the finite element programs to facilitate the computation and decrease the analysis time.

To satisfy the experimental data requirements of the nonlinear regression technique, experimental setups were constructed for both the one-dimensional and two-dimensional models. Appropriate boundary conditions were implemented to mimic one-dimensional and two-dimensional heat transfer. To assure the quality and accuracy of the data, the experiments were optimized for the sensor locations and heating times.

Two one-dimensional effective finite element heat transfer models were developed to assist with the estimation procedure. The first model was created with an insulated boundary condition on the top surface of the outer wing subcomponent, whereas the second model was outfitted with a convective boundary condition over the insulation boards. Both models were used to determine the sensitivity coefficients and estimates of the effective thermal conductivity and volumetric heat capacity of the outer wing subcomponent. The estimates obtained from both models for the effective thermal conductivity, agreed with the results determined with the steady-state thermal resistance analysis and revealed a temperature dependence. The volumetric heat capacity, on the other hand, stayed constant over the 150°C temperature range (295K-445K) investigated. Although the estimates calculated with the insulated boundary were slightly more accurate, the convective boundary condition represented the physics of the problem more accurately and was therefore kept to develop the following one-dimensional detailed model of the outer wing subcomponent.

The one-dimensional detailed model, later modified to the one-dimensional half-model, was created to determine the thermal properties of the independent materials used

in the construction of components of the outer wing subcomponent, including the facesheets, the honeycomb blanket, the skins, and the torque tubes. The sensitivity analysis revealed sensitivity coefficients with extremely small magnitudes and the corresponding sensitivity plots showed heavy correlation between the parameters. The sensitivity coefficients for the thermal conductivity and volumetric heat capacity of the honeycomb blanket were dominant, providing accurate estimates. The estimates for the volumetric heat capacity of the materials were consistent at the three different surrounding temperature settings of 295K, 375K, and 445K. The spread of the thermal conductivity estimates, however, was directly linked to the magnitude of the corresponding sensitivity coefficients and the heavy correlation between the parameters. The standard deviations between the calculated and experimental temperature histories proved that the honeycomb blanket is the component dictating the thermal response of the top and bottom halves of the outer wing subcomponent and that the accuracy of the thermal properties of the honeycomb were most important to accurately model the thermal characteristics of the outer wing subcomponent.

7.2 Conclusions

7.2.1 Finite Element Models

1. The two one-dimensional effective models with the insulated and convective boundary conditions both gave accurate estimates for the effective thermal conductivity and volumetric heat capacity of the outer wing subcomponent. However, since the convective boundary case represented a more realistic model of the true physics of the problem, it was determined to be the best choice.
2. The estimates obtained from the two one-dimensional effective model with the insulated and convective boundary were on the same order of magnitude as those determined from the steady-state thermal resistance analysis.
3. Although the magnitude of the estimates obtained from the two one-dimensional effective models are different, the corresponding temperature histories match the experimental data well, yielding almost identical objective functions.

7.2.2 Experimental Optimization

1. The optimal sensor location was the heated surface for both the one-dimensional and two-dimensional experimental setups
2. The optimal heating time was determined to be longer for the two-dimensional than for the one-dimensional experiment

7.2.3 Implementation of the Genetic Algorithm

1. The computational time was drastically reduced by incorporating the genetic algorithm into the finite element program via processors
2. The convergence criterion assured the convergence of the estimates
3. Although the parameters estimated were highly correlated and the sensitivity coefficients small in magnitude for the one-dimensional half-model, the genetic algorithm was able to converge on acceptable solutions.

7.2.4 Thermal Estimates

1. In addition to being in accordance with the approximate value of the thermal conductivity obtained with the steady-state thermal resistance analysis, the one-dimensional effective thermal properties were accurate but dependent on the boundary condition of the model. The convective boundary condition proved to be more accurate and was therefore used for the one-dimensional detailed and two-dimensional models.
2. The one-dimensional detailed thermal property estimates proved that the dominant component of the top and bottom halves of the outer wing subcomponent is the honeycomb blanket. Accurate thermal behavior can be modeled without knowing the exact thermal properties of the remaining components, due to their small sensitivity.
3. Accurate estimates were gathered for the volumetric heat capacity for all the components, whereas the values for the thermal conductivity were less consistent.

7.3 Overall Conclusion

The noninvasive estimation technique developed in this study was successful in estimating the effective and detailed thermal properties of the materials used in the outer wing subcomponent. This method also determined the simplifications that can be applied to the models without losing accuracy while predicting the thermal response of the structure with specific boundary conditions. The robustness of the genetic algorithm implemented in the estimation process proved to be a great asset and provided estimates for the thermal properties regardless of the correlation or magnitude of the various sensitivity coefficients.

Chapter 8

Recommendations

This research provides a good base for developing further noninvasive simultaneous thermal property estimation procedures geared towards complex structures, for which the conventional methods are not an option. Although, this method proved to deliver good estimates, there are many areas of this research that could be improved to yield better estimates, some of which will be discussed following an overview of the recommendations for future work:

1. Determine the exact boundary condition for the one-dimensional effective model by estimating the value of the heat transfer coefficient h for experiments at room temperature and elevated temperatures.
2. Investigate the effect of the bonding material used to attach the thermocouples on the thermocouple readings.
3. Develop a more efficient Genetic Algorithm that requires less initial populations to reduce the computational time or use a different minimization technique.
4. Use this method to determine any defects in existing structures.

As seen for the estimates of the one-dimensional effective models, the boundary condition applied to the model greatly influenced the magnitude of the estimates. The trends remained unchanged, but due to the increased sensitivity in the insulated model, the estimates were slightly more accurate. However, it is important to match the model as closely as possible to the physics of the problem. Improving or changing the experimental boundary conditions could make the modeling process easier and more accurate. It would therefore be helpful to make the heat transfer coefficient one of the estimated parameters

to get the most appropriate boundary condition possible, unless enough insulation is used to provide a perfectly insulated boundary condition.

The genetic algorithm used throughout this study was very powerful and its robustness allowed to obtain estimates for the thermal properties regardless of the magnitude of the sensitivities and the correlation between them. However, the genetic algorithm is computationally very intensive and therefore, reducing the number of initial populations or generations could reduce the CPU time tremendously. The next step would be to either enhance the genetic algorithm or to replace the existing genetic algorithm with another non-gradient based method.

Finally, a good application of this method would be to implement it to discover irregularities, defects, or nonhomogeneities such as air pockets in composites or delaminated components in structures.

Bibliography

Bard, Y., 1970, "Comparison of Gradient Methods for the Solution of Nonlinear Parameter Estimation Problems," *SIAM Journal of Numerical Analysis*, Vol. 7, pp. 157–186.

Beck, J. V., 1964, "The Optimum Analytical Design of Transient Experiments for Simultaneous Determinations of Thermal Conductivity and Specific Heat," Ph.D. Thesis, Michigan State University.

Beck, J. V., 1966, "Transient Determination of Thermal Properties," *Nuclear Engineering Design*, Vol. 3, pp. 373–381.

Beck, J. V., 1969, "Determination of Optimum, Transient Experiments for Thermal Contact Conductance," *International Journal of Heat and Mass Transfer*, Vol. 12, pp. 621–633.

Beck, J. V. and Arnold, K. J., 1977, *Parameter Estimation in Engineering and Science*, John Wiley and Sons, New York.

Box, G. E. P. and Kanemasu, H., 1972, "Topics in Model Building, Part II, on Nonlinear Least Squares," Tech. Rep. 321, University of Wisconsin, Department of Statistics, Madison, Wisconsin.

Brennan, J. J., Bentsen, L. D., and Hasselman, D. P. H., 1982, "Determination of the Thermal Conductivity and Diffusivity of Thin Fibres by the Composite method," *Journal of Material Science*, Vol. 17, pp. 2337–2342.

Carroll, D. L., 1996a, "Chemical Laser Modeling with Genetic Algorithms," *AIAA Journal*, Vol. 34, No. 2, pp. 338–346.

Carroll, D. L., 1996b, "Genetic Algorithms and Optimizing Chemical Oxygen-Iodine Lasers," *Developments in Theoretical and Applied Mechanics*, Vol. 18.

Copenhaver, D. C., 1996, "Thermal Characterization of Honeycomb Core Sandwich Structures," M.S. Thesis, Virginia Polytechnic Institute and State University, Department of Mechanical Engineering, Blacksburg, VA.

Copenhaver, D. C., Scott, E. P., and Hanuska, A. R., 1997, "Thermal Characterization of Honeycomb Sandwich Structures," in: *32nd Thermophysics Conference*.

Dowding, K. J., Beck, J. V., A. Ulbrich, B. B., and Hayes, J., 1995, "Estimation of Thermal Properties and Surface Heat Flux in Carbon-Carbon Composite," *Journal of Thermophysics and Heat Transfer*, Vol. 9, No. 2, pp. 345–351.

Dowding, K. J., Beck, J. V., and Blackwell, B. F., 1996, "Estimation of Directional-Dependent Thermal Properties in a Carbon-Carbon Composite," *International Journal of Heat and Mass Transfer*, Vol. 39, No. 15, pp. 3157–3164.

Fabbri, G., 1997, "A Genetic Algorithm for Fin Profile Optimization," *International Journal Of Heat and Mass Transfer*, Vol. 40, No. 9, pp. 2165–2172.

Fukai, J., Nogami, H., Miura, T., and Ohtani, S., 1991, "Simultaneous Estimation of Thermophysical Properties by Periodic Hot-Wire Heating Method," *Experimental Thermal and Fluid Science*, Vol. 4, No. 2, pp. 198–204.

Furuya, H. and Haftka, R. T., 1993, "Genetic Algorithms for Placing Actuators on Space Structures," in: *Proceedings of the Fifth International Conference on Genetic Algorithms*, Urbana, IL, pp. 536–542.

Garcia, S., Guynn, J., and Scott, E. P., 1998, "Use of Genetic Algorithms in Thermal Property Estimation: Part II - Simultaneous Estimation of Thermal Properties," *Numerical Heat Transfer*, Vol. 33, No. 2, pp. 149–168.

Garcia, S. and Scott, E. P., 1998, "Use of Genetic Algorithms in Thermal Property Estimation: Part I - Experimental Design Optimization," *Numerical Heat Transfer*, Vol. 33, No. 2, pp. 135–147.

Garnier, B., Delaunay, D., and Beck, J. V., 1994, "Improved Measurement of the Surface Temperature of Composite Materials for the Optimal Estimation of their Thermal Properties," *High Temperatures - High Pressures*, Vol. 26, pp. 15–23.

Guynn, J. H., 1996, "Estimation of Thermal Properties in a Medium with Conduction and Radiation Heat Transfer," M.S. Thesis, Virginia Polytechnic Institute and State University, Department of Mechanical Engineering, Blacksburg, VA.

Hanak, J. P., 1995, "Experimental Verification of Optimal Experimental Designs for the Estimation of Thermal Properties of Composite Materials," M.S. Thesis, Virginia Polytechnic Institute and State University, Department of Mechanical Engineering, Blacksburg, VA.

Harris, J. P., Yates, B., Batchelor, J., and Garrington, J. P., 1982, "The Thermal Conductivity of Kevlar Fibre-Reinforced Composites," *Journal of Material Science*, Vol. 17, pp. 2925–2931.

Jones, M. R., Brewster, M. Q., and Yamada, Y., 1996, "Application of a Genetic Algorithm to the Optical Characterization of Propellant Smoke," *Journal of Thermophysics and Heat Transfer*, Vol. 10, No. 2, pp. 372–377.

Moncman, D. A., 1994, "Optimal Experimental Designs for the Estimation of Thermal Properties of Composite Materials," M.S. Thesis, Virginia Polytechnic Institute and State University, Department of Mechanical Engineering, Blacksburg, VA.

Moncman, D. A., Hanak, J. P., Copenhaver, D. C., and Scott, E. P., 1995, "Optimal Experimental Designs for Estimating Thermal Properties," in: *4th ASME/JSME Thermal Engineering Joint Conference*, Vol. 3, pp. 461–468.

Özişik, M. N., 1993, *Heat Conduction*, 2nd edn., John Wiley & Sons, Inc., 605 Third Avenue, New York, NY 10158-0012.

Parker, W. J., Jenkins, R. J., Butler, C. P., and Abbott, G. L., 1961, "Flash Method of Determining Thermal Diffusivity, Heat Capacity, and Thermal Conductivity," *Journal of Applied Physics*, Vol. 32, No. 9, pp. 1679–1684.

Penn, B. G., III, F. E. L., White, W. T., and Clemons, J. M., 1986, "Measurement of the Thermal Conductivity of Composites Using Heat Flow Sensors," *Journal of Polymer Composites*, Vol. 7, No. 6, pp. 426–430.

- Pfahl, R. C., 1966, "Nonlinear Least Squares: A Method for Simultaneous Thermal Property Determination in Ablating Polymeric Materials," *Journal of Applied Polymer Science*, Vol. 10, pp. 1111–1119.
- Pfahl, R. C. and Mitchell, B. J., 1970, "Simultaneous Measurement of Six Thermal Properties of a Charring Plastic," *Journal of Heat Transfer*, Vol. 13, pp. 275–286.
- Pilling, M. W., Yates, B., Black, M. A., and P. Tattersall, 1979, "The Thermal Conductivity of Carbon Fiber-Reinforced Composites," *Journal of Materials Science*, Vol. 14, pp. 1326–1338.
- Raudensky, M., Woodbury, K., Kral, J., and Brezina, T., 1995, "Genetic Algorithm in Solution of Inverse Heat Conduction Problems," *Journal of Numerical Heat Transfer, Part B*, Vol. 28, pp. 293–306.
- Scott, E. P. and Beck, J. V., 1992, "Estimation of Thermal Properties in Epoxy Matrix/Carbon Fiber Composite Material," *Journal of Composite Materials*, Vol. 26, No. 1, pp. 20–36.
- Scott, E. P. and Haftka, R. T., 1995, "Optimization and Experiments," *American Institute of Aeronautics and Astronautics*, pp. 1–11.
- Taktak, R., Beck, J. V., and Scott, E. P., 1993, "Optimal Experimental Design for Estimating Thermal Properties of Composite Materials," *International Journal of Heat and Mass Transfer*, Vol. 36, No. 12, pp. 2977–2986.
- Taylor, R. E., Jortner, J., and Groot, H., 1985, "Thermal Diffusivity of Fiber-Reinforced Composites Using the Laser Flash Technique," *Carbon*, Vol. 23, No. 2, pp. 215–222.
- Whetstone, D. W., 1983, *Engineering Analysis Language*, Engineering Information Systems, Inc., San Jose, CA.

Appendix A

Calculation of the Effective Thermal Conductivity for the Insulation Boards

The following section describes the steps required to calculate the average heat transfer coefficient \bar{h} due to natural convection over the insulation boards. After determination of the average heat transfer coefficient, an effective thermal conductivity will be sought, combining both the natural convection effects over the insulation boards and the conduction effects through the thickness of the insulation boards. The case presented in this section is for a surrounding experimental temperature of approximately 375K.

The first step is to determine the average temperature at steady-state between the top surface of the insulation boards $T_{insulation}$ and the fluid surrounding the insulation, which in this study is T_{air} . The temperatures of the insulation $T_{insulation}$ and the air T_{air} are given below as:

$$T_{insulation} = 378.55\text{K}, \quad (\text{A.1})$$

$$T_{air} = 374.69\text{K}, \quad (\text{A.2})$$

yielding an average temperature between the top of the insulation and the surrounding air

of:

$$T_{avg} = \frac{T_{air} + T_{insulation}}{2} = 376.62\text{K}. \quad (\text{A.3})$$

The properties of the surrounding fluid (air) can now be determined at the average temperature T_{ave} along with the remaining variables that will be needed to calculate the average heat transfer coefficient \bar{h} and the effective thermal conductivity k_{eff} through the insulation boards. The variables are:

$$g = 9.81 \quad (\text{m/s}^2) \quad (\text{A.4})$$

$$\nu_{air} = 23.67 \times 10^{-6} \quad (\text{m}^2/\text{s}) \quad (\text{A.5})$$

$$\alpha_{air} = 34.10 \times 10^{-6} \quad (\text{m}^2/\text{s}) \quad (\text{A.6})$$

$$k_{air} = 31.90 \times 10^{-3} \quad (\text{W/mK}) \quad (\text{A.7})$$

$$k_{insulation} = 0.0277 \quad (\text{W/mK}) \quad (\text{A.8})$$

$$L_{plate} = 0.4883 \quad (\text{m}) \quad (\text{A.9})$$

$$L_{insulation} = 0.0508 \quad (\text{m}) \quad (\text{A.10})$$

An appropriate form to calculate the Rayleigh Ra_L number is:

$$Ra_L = \frac{g\beta(T_{insulation} - T_{air})L^3}{\alpha\nu} \quad (\text{A.11})$$

$$Ra_L = 1.42 \times 10^7 \quad (\text{A.12})$$

where β , the thermal expansion coefficient, can be determined by taking the inverse of the average temperature T_{avg} for an ideal gas and carries the units of (1/K). There are recommended correlations between the average Nusselt number \overline{Nu}_L and the Rayleigh number Ra_L for the setup described in Fig. A.1 depending on the numerical value of the Rayleigh number Ra_L .

$$\overline{Nu}_L = 0.54Ra_L^{1/4} \quad 10^4 \leq Ra_L \leq 10^7 \quad (\text{A.13})$$

$$\overline{Nu}_L = 0.15Ra_L^{1/3} \quad 10^7 \leq Ra_L \leq 10^{11} \quad (\text{A.14})$$

Since the calculated value of the Rayleigh number is very close to 10^7 , either of the above equations is adequate. The value of the average Nusselt number \overline{Nu}_L was determined using Eq. A.14:

$$Ra_L = 36.32 \quad (\text{A.15})$$

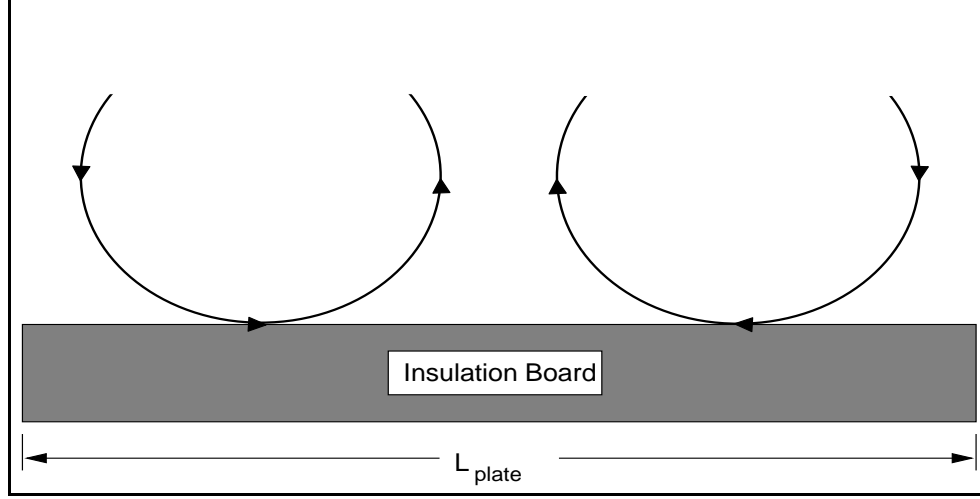


Figure A.1 Convection rolls.

The average heat transfer coefficient \bar{h} can then be determined using the following equation:

$$\bar{h} = \frac{k_{air} \overline{Nu_L}}{L_{plate}} \quad (A.16)$$

$$\bar{h} = 2.37(\text{W/m}^2\text{K}) \quad (A.17)$$

At this point, the effective thermal conductivity k_{eff} that includes the conduction through and the convection over the insulation boards can be determined using a simple resistance network shown in Fig. A.2 that translates into the following expression:

$$k_{eff} = \frac{L_{insulation} \bar{h} k_{insulation}}{L_{insulation} \bar{h} + k_{insulation}} \quad (A.18)$$

Eq. A.18 yields a final value for the effective thermal conductivity K_{eff} of:

$$\boxed{k_{eff} = 0.0225(\text{W/mK})}. \quad (A.19)$$

In the case of forced convection, the heat transfer coefficient h is much higher than for natural convection. Therefore, in order to compare the effect of a higher convection coefficient and the estimates of the thermal conductivity and volumetric heat capacity, the effective thermal conductivity for the insulation boards was calculated. The results are summarized in the Table A.1.

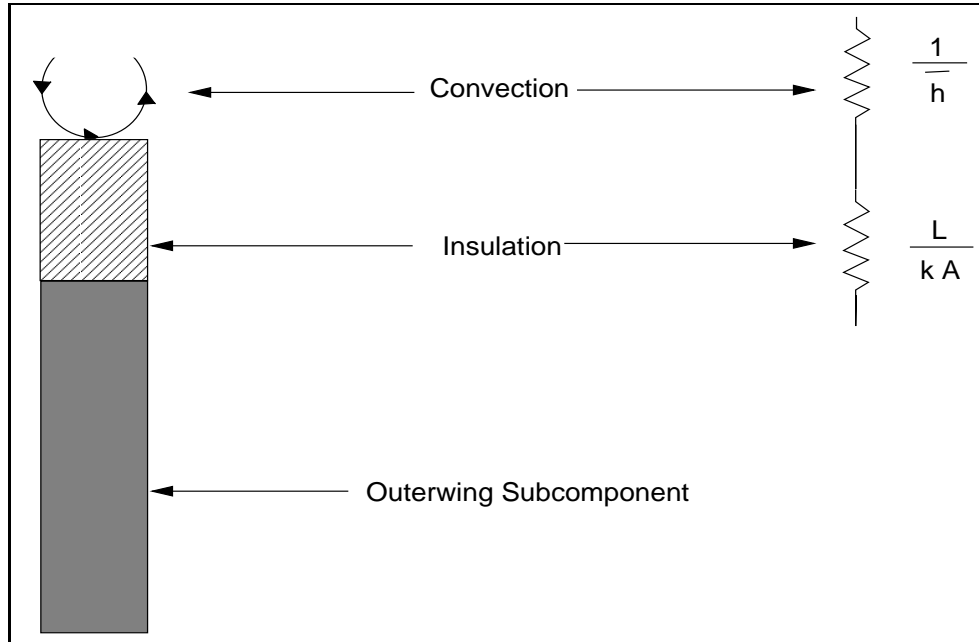


Figure A.2 Resistance diagram for insulation and convection.

Table A.1 Effective thermal conductivity of the insulation boards as a function of the heat transfer coefficient.

Heat Transfer Coefficient (W/m^2K)	Effective Thermal Conductivity (W/mK)
10.0	0.0263
20.0	0.0270
30.0	0.0272

Appendix B

Greens Function Solution of the One-Dimensional Effective Model

This Appendix describes the derivation of the one-dimensional effective model using the Green's function method. The derivation of this particular model was performed to benchmark the finite element package EAL and observe the accuracy of its output.

B.1 Problem Statement and Differential Element Representation

The problem in hand is the one-dimensional effective model of the outer wing subcomponent mentioned throughout this study. The model consists of a constant applied heat flux at the top surface and a constant temperature boundary condition at the bottom surface of the outer wing subcomponent, as shown in Fig. B.1.

The first step in solving any heat transfer problem should be a differential element representation of the posed problem as illustrated in Fig. B.2. Differential element representations usually carry much insight of the problem and facilitate the solution process.

B.2 Conservation of Energy

By applying the conservation of energy to the differential element representation described in the previous section, we can derive the one-dimensional diffusion equation representative

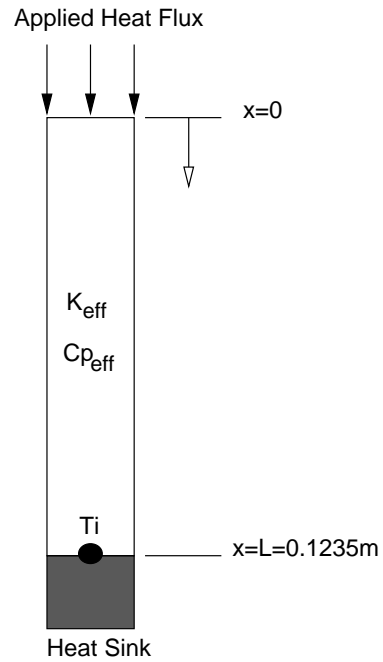


Figure B.1 One-dimensional representation of the outer wing subcomponent.

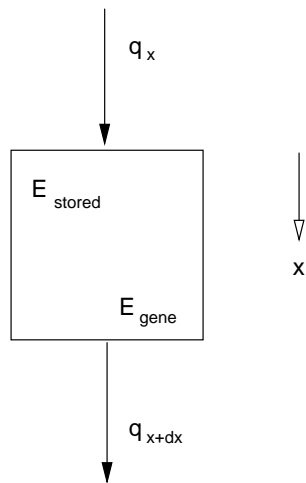


Figure B.2 Differential element representation of the one-dimensional effective model.

of this problem.

$$\boxed{\dot{E}_{STORED} = \dot{E}_{IN} - \dot{E}_{OUT} + \dot{E}_{GENE}}, \quad (\text{B.1})$$

$$\dot{E}_{STORED} = \rho C_{peff} \frac{\partial T}{\partial t} dx, \quad (\text{B.1a})$$

$$\dot{E}_{GENE} = g(x, t) dx, \quad (\text{B.1b})$$

$$\dot{E}_{IN} = \dot{q}_x, \quad (\text{B.1c})$$

$$\dot{E}_{OUT} = \dot{q}_{x+dx}, \quad (\text{B.1d})$$

where \dot{E}_{STORED} is the amount of energy stored, \dot{E}_{GENE} represents the amount of internal heat generation, \dot{E}_{IN} is the amount of energy coming into the system, and \dot{E}_{OUT} is the amount of energy that goes out of the system. For the problem defined above we have no internal heat generation. We can therefore rearrange the above equations and get the following expression by incorporating Fourier's law of heat conduction:

$$\rho C_{peff} \frac{\partial T}{\partial t} dx = k_{eff} \frac{\partial^2 T}{\partial x^2}, \quad (\text{B.2})$$

which can further be simplified to get the typical one-dimensional diffusion equation:

$$\boxed{\frac{\partial^2 T}{\partial x^2} = \frac{1}{\alpha} \frac{\partial T}{\partial t}}. \quad (\text{B.3})$$

B.3 Boundary and Initial Conditions

The boundary and initial conditions that drive the problem are expressed as follows, and are shown in Fig. B.1:

$$-k \frac{\partial T}{\partial x} = q_s \quad @ \ x = 0 \quad (\text{B.3a})$$

$$T = T_i \quad @ \ x = L \quad (\text{B.3b})$$

$$T = T_i \quad @ \ t = 0 \quad (\text{B.3c})$$

B.4 Selection of the Green's Function

The selection of the Green's function is an important step in obtaining the appropriate temperature distribution corresponding to the nature of the problem, the boundary conditions

and initial conditions.

$$\boxed{\frac{\partial^2 G}{\partial x^2} + \frac{1}{\alpha} \delta(t - t_0) = \frac{1}{\alpha} \frac{\partial G}{\partial t}}, \quad (\text{B.4})$$

$$\frac{\partial G}{\partial x} = 0 \quad @ \ x = 0, \quad (\text{B.4a})$$

$$G = 0 \quad @ \ x = L, \quad (\text{B.4b})$$

$$G = 0 \quad @ \ t \leq t_0. \quad (\text{B.4c})$$

The Green's function can be defined as:

$$G(x, t | x_0, t_0) = G(\text{observation point} | \text{source point})$$

B.5 Selection of the Eigenfunctions and Eigenvalues

As in any boundary value problem, the Eigenfunctions and Eigenvalues of the problem in hand need to be determined.

$$\boxed{\frac{\partial^2 \Psi_m}{\partial x^2} + \beta_m^2 \Psi_m = 0}, \quad (\text{B.5})$$

$$\frac{d\Psi_m}{dx} = 0 \quad @ \ x = 0, \quad (\text{B.5a})$$

$$\Psi_m = 0 \quad @ \ x = L. \quad (\text{B.5b})$$

The general solution looks like the following:

$$\psi_m(x) = k_1 \sin(\beta_m x) + k_2 \cos(\beta_m x). \quad (\text{B.6})$$

Applying the boundary conditions described in Eq. B.5a and Eq. B.5b, we can determine the Eigenvalues β_m :

$$\boxed{\beta_m = (2m - 1) \frac{\pi}{2L}}, \quad (\text{B.7})$$

with $k_2 \neq 0$, and $m=1, 2, 3, \dots$. The corresponding Eigenfunctions $\Psi_m(x)$ can be determined with:

$$\boxed{\Psi_m(x) = \cos(\beta_m x)}. \quad (\text{B.8})$$

B.6 Solution of the Green's Function

Now that the Green's function, the Eigenfuctions, and the Eigenvalues have been defined, the Green's function specific to the problem can be solved. Using the orthogonality condition, the Green's function can be written as:

$$G(x, t|x_0, t_0) = \sum_{m=1}^{\infty} C_m(t) \cos(\beta_m x) \quad (\text{B.9})$$

$$\int_{x=0}^L G \cos(\beta_m x) dx = \sum_{m=1}^{\infty} C_m(t) \int_{x=0}^L \cos(\beta_m x) \cos(\beta_j x) dx = \begin{cases} 0 & \text{if } m \neq j \\ N_m = \frac{L}{2} & \text{if } m = j \end{cases} \quad (\text{B.10})$$

where N_m is the normalization integral. The Transform and Inversion used to solve for the Green's function are:

$$\text{Transform: } \quad \overline{G_m(t)} = \int_{x=0}^L G \cos(\beta_m x) dx \quad (\text{B.11})$$

$$\text{Inversion: } G(x, t|x_0, t_0) = \sum_{m=1}^{\infty} \frac{1}{N_m} \overline{G_m(t)} \cos(\beta_m x) \quad (\text{B.12})$$

The first step is to transform Eq. B.4 using the transformation stated in Eq. B.11. Both sides of Eq. B.4 were multiplied by $\cos(\beta_m x)$, the eigenfunctions, and integrated in space.

$$\begin{aligned} \underbrace{\int_{x=0}^L \frac{\partial^2 G}{\partial x^2} \cos(\beta_m x) dx}_{-\beta_m^2 \overline{G_m}} + \underbrace{\frac{1}{\alpha} \int_{x=0}^L \delta(x - x_0) \delta(t - t_0) \cos(\beta_m x) dx}_{\frac{1}{\alpha} \delta(t - t_0) \cos(\beta_m x_0)} \\ = \underbrace{\frac{1}{\alpha} \int_{x=0}^L \frac{\partial G}{\partial t} \cos(\beta_m x) dx}_{\frac{1}{\alpha} \frac{d\overline{G_m}}{dt}}. \end{aligned} \quad (\text{B.13})$$

Rearranging Eq. B.13, we obtain:

$$\frac{d\overline{G_m}}{dt} + \alpha \beta_m^2 \overline{G_m} = \delta(t - t_0) \cos(\beta_m x_0), \quad (\text{B.14})$$

which is an ordinary differential equation that can easily be solved for $\overline{G_m}$ using the initial condition defined earlier, $\overline{G_m} = 0$ @ $t \leq t_0$:

$$\overline{G_m} = e^{-\alpha \beta_m^2 t} \left[\int_{t'=0}^t e^{\alpha \beta_m^2 t'} \delta(t' - t_0) \cos(\beta_m x_0) dt' \right] \quad (\text{B.15})$$

$$\overline{G_m} = e^{-\alpha \beta_m^2 t} \cos(\beta_m x_0) e^{\alpha \beta_m^2 t_0} \quad (\text{B.16})$$

$$\overline{G_m} = \cos(\beta_m x_0) e^{-\alpha \beta_m^2 (t - t_0)} \quad (\text{B.17})$$

Finally, the solution of the Green's function can be obtained using the Inversion formula described in Eq. B.12, yielding the following expression for the Green's function:

$$\boxed{G(x, t|x_0, t_0) = \frac{2}{L} \sum_{m=1}^{\infty} \cos(\beta_m x_0) \cos(\beta_m x) e^{-\alpha \beta_m^2 (t-t_0)}} \quad (\text{B.18})$$

B.7 Solution of the Temperature Distribution in Terms of the Green's Function

The temperature distribution of the one-dimensional effective model can be expressed as a function of the Green's function, which was derived in the previous section. Both sides of Eq. B.3 are multiplied by the Green's function and integrated in time and space.

$$\int_{t_0}^t \int_{x_0=0}^L \left[\underbrace{\frac{\partial^2 T}{\partial x^2}}_I + \underbrace{S(x_0, t_0)}_{II} - \underbrace{\frac{1}{\alpha} \frac{\partial T}{\partial t}}_{III} \right] G dx_0 dt_0 = 0, \quad (\text{B.19})$$

where I , II , and III are defined as:

$$I = \int_{t_0}^t \left[\left. \frac{dT}{dx_0} G \right|_0^L - T \left. \frac{dG}{dx_0} \right|_0^L + \int_0^L T \frac{\partial^2 G}{\partial x^2} dx_0 \right] dt_0 \quad (\text{B.20})$$

$$= \int_{t_0}^t \left[\frac{q_s(t)}{k} G_{x_0=0} - T_s(t) \left. \frac{dG}{dx_0} \right|_{x_0=L} \right] dt_0 + \int_{t_0}^t \int_{x_0=0}^L T \frac{\partial^2 G}{\partial x^2} dx_0 dt_0 \quad (\text{B.21})$$

$$II = \int_{t_0}^t \int_{x_0=0}^L S(x_0, t_0) G dx_0 dt_0 \quad (\text{B.22})$$

$$III = \frac{1}{\alpha} \int_{x_0=0}^L \left[T G|_{t_0=0}^t - \int_{t_0}^t T \frac{dG}{dt_0} dt_0 \right] dx_0 \quad (\text{B.23})$$

$$= \frac{1}{\alpha} \left[- \int_{x_0=0}^L T_i(x_0) G dx_0 - \int_{x_0=0}^L \int_{t_0}^t T \frac{dG}{dt_0} dt_0 dx_0 \right] \quad (\text{B.24})$$

Substituting I , II , III back into Eq. B.19, we obtain the following expression:

$$\begin{aligned} & \underbrace{\int_{t_0}^t \int_{x_0=0}^L T \frac{d^2 G}{dx_0^2} + \frac{1}{\alpha} T \frac{dG}{dt_0} dt_0 dx_0}_{-\frac{1}{\alpha} T(x, t)} \\ & + \int_{t_0}^t \left[\frac{q_s(t)}{k} G_{x_0=0} - T_s(t) \left. \frac{dG}{dx_0} \right|_{x_0=L} \right] dt_0 \\ & + \int_{t_0}^t \int_{x_0=0}^L S(x_0, t_0) G dx_0 dt_0 \\ & + \frac{1}{\alpha} \int_{x_0=0}^L T_i(x_0) G dx_0 = 0 \end{aligned} \quad (\text{B.25})$$

which can be rewritten as:

$$\begin{aligned}
T(x, t) &= \int_{x_0=0}^L T_i(x) G_{t_0=0} dx_0 && \text{Initial conditions} \\
&+ \alpha \int_{t_0}^t \left[\frac{q_s(t)}{k} G_{x_0=0} - T_s(t) \frac{dG}{dx_0} \Big|_{x_0=L} \right] dt_0 && \text{Boundary conditions (B.26)} \\
&+ \alpha \int_{x_0=0}^L \int_{t_0}^t S(x_0, t_0) dt_0 dx_0 && \text{Volumetric sources}
\end{aligned}$$

Eq. B.26 gives good insight to the physics of the problem. Only initial conditions, boundary conditions, or volumetric heat sources can affect the solution, and therefore change the temperature distribution. Since the temperature distribution is written as a function of the Green's function, Eq. B.18 and Eq. B.26 can be used together to get the temperature distribution as a function of space and time only, keeping in mind, that no volumetric heat sources were applied in this problem.

B.8 Final Temperature Distribution

The final temperature distribution for the one-dimensional effective model of the outer wing subcomponent can be expressed by substituting the Green's function solution Eq. B.18 into Eq. B.26 to get:

$$\boxed{T(x, t) = T_i + \frac{q_s}{k} \frac{2}{L} \sum_{m=1}^{\infty} \cos(\beta_m x) \frac{1}{\beta_m^2} (1 - e^{-\alpha \beta_m^2 (t-t_0)})} \quad (\text{B.27})$$

Appendix C

EAL Processors

This Appendix is dedicated to describing the development of the genetic processors, which made it possible to incorporate the **Extended Elitist Genetic Algorithm (EEGA)**, created by Sandrine Garcia, into the finite element package EAL. The first section describes the most frequently used existing EAL processors, whereas the following section concentrates on the development of the genetic processors.

C.1 Frequently Used Processors

This section gives a quick overview and brief explanation of a small fraction of the processors that reside in the EAL processor library that were widely used in the programs. More details can be found about the specific processors in the sections mentioned at the end of each processor description.

- U1: This processor performs executive and runstream utility operations, including the creation, editing, and display of runstream elements and related functions, Whetstone (1983). It is usually the first processor executed in an EAL runstream. (Sections 1-2, 11)
- AUS: This processor performs computational and data entry, editing functions, including many array arithmetic commands, Whetstone (1983). The Arithmetic Utility System processor must be executed before any arithmetic command is used. (Sections 1-2, 15)

- DCU: The Database Utility processor performs database utility functions, including displaying library tables of contents, copying datasets, disabling or enabling datasets, changing dataset names, and many related functions. (Sections 1-2, 12)
- TAB: The TAB processor is executed to create tables of structural properties and joint position coordinates. (Section 3.1)
- ELD: The Element Definition processor uses the information created by the TAB processor and determines the elements. One-, two-, and three-dimensional elements can be created, depending on the geometry and application of the elements. (Section 3.2)
- TGEO: The TGEO processor performs element geometry checks to assure aspect ratios and stops the execution of any further processors if the geometry of the elements would not yield accurate temperature distributions. (Section 4.1)
- TRTB: The TRTB processor is the implicit transient thermal analyzer which calculates the transient temperature distributions. (SPAR Section 6.3)
- EXIT: This processor terminates the runstream and performs the necessary library functions to correctly close the libraries. (Section 1-3)

As mentioned previously, the EAL processors described above represent only a small fraction of the capabilities EAL has to offer. Each processor is task specific and can be adjusted by the use of the various RESET controls, depending on the processor.

C.2 The Genetic Processors

One of the convenient features of the EAL software package, consists in allowing additional user defined and developed processors to be incorporated in the EAL processor library by recompiling and linking of the EAL base code. This feature was extensively used to combine the Genetic Algorithm based optimization and estimation schemes with the transient finite element capabilities of EAL to create an array of one-dimensional and two-dimensional models used throughout this study.

The need to incorporate the Genetic Algorithm into the finite element package was driven by the calculation of the objective functions used in the optimization and estimation

schemes. The calculation of the objective functions required both experimental and calculated temperatures. In the original Genetic Algorithm developed by Sandrine Garcia, the calculated temperatures were obtained using theoretical solution models. In this study, the calculated temperatures were generated using the finite element software package EAL. It is the nature of Genetic Algorithms to be computation intensive, requiring many objective function calculations before reaching convergence. It was therefore a necessity to incorporate the Genetic Algorithm into EAL, disguised as several processors each with a specific task, to eliminate the many file and data transfers that would have been involved otherwise.

The original Genetic Algorithm was therefore broken down into five specific processors that were later added to the EAL processor library. The processors were written in FORTRAN as were many of the processors described in the EAL Reference Manual. Each processor represents a subroutine that needs to be compiled using the program `compile`, and linked to `new333.lib` along with `XQTER.f` to create the new EAL version containing the user defined processors. Appropriate changes need to be made to the file `link` and `XQTER.f` before compiling and linking. It is also important when compiling, linking as well as developing the processors, to determine the exact requirements of the chipset of the machine. All the processors were developed for a Silicon Graphics machine with a IP22 chipset.

The five processors described follow the steps described in Chapter ?? Section 3.3. They have elementary RESET options to fine-tune the processor to specific needs, but do not contain ERROR statements that caution the user when datasets are missing or variables are improperly defined. Fig. C.1 gives an overview on how the processors were used in the EAL runstream. Fig. C.1 also shows the order in which the processors need to be executed to properly work together. Table C.1 summarizes the datasets produced and required by each processor, as well as the available RESET options.

The following subsections each describe the specific processors. The processor programs are also attached at the end of each subsection. The overall capacity of the processors was limited to 10 parameters. However, this number can easily be changed in the FORTRAN source codes.

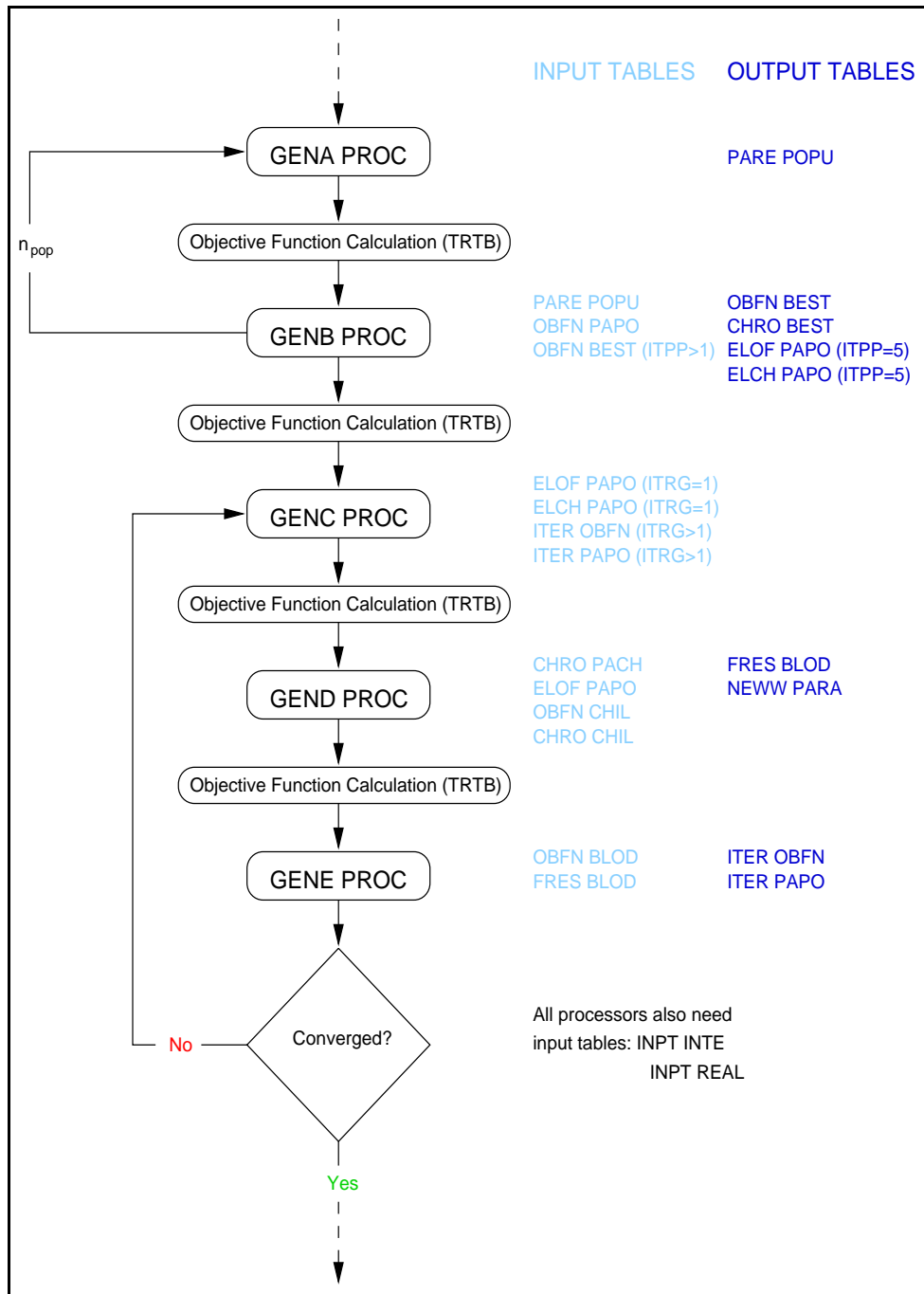


Figure C.1 EAL genetic runstream.

Table C.1 Overview of the genetic processors

Processor Name	Available RESETS	Input Tables	Output Tables
GENA PROC	NUIN NOUT NTIM ITPP	INPT INTE INPT REAL	PARE POPU
GENB PROC	NUIN NOUT ITPP	INPT INTE INPT REAL PARE POPU OBFN PAPO OBFN BEST (ITPP>1)	OBFN BEST CHRO BEST ELOF PAPO (ITPP=5) ELCH PAPO (ITPP=5)
GENC PROC	NUIN NOUT ITPP ITRG	INPT INTE INPT REAL ELOF PAPO (ITRG=1) ELCH PAPO (ITRG=1) ITER OBFN (ITRG>1) ITER PAPO (ITRG>1)	CHRO CHIL CHRO PACH
GEND PROC	NUIN NOUT ITPP ITRG	INPT INTE INPT REAL CHRO PACH ELOF PAPO OBFN CHIL CHRO CHIL	FRES BLOD NEWW PARA
GENE PROC	NUIN NOUT ITPP ITRG	INPT INTE INPT REAL OBFN BLOD FRES BLOD	ITER PAPO ITER OBFN

C.2.1 The GENA PROC Processor

This processor creates an initial population of size n_s . The chromosomes are chosen randomly from an initial range that was determined by the user as the initial parameter space. This processor is executed n_{pop} times to create the initial elitist parent population. This is the first genetic processor that needs to be executed. The RESET options available are:

NUIN: Determines the input library. The processor will go to library NUIN to look for any datasets or tables it needs to complete its task. The default value for this RESET is 1.

NOUT: Determines the output library. Any tables or datasets that are created by the processor will be written to the NOUT library. The default value for this RESET is 1.

NTIM: Determines the amount of printout during the processor execution. The default value is for this RESET 1. A value of 0 will give minimum printout.

ITPP: Determines which parent population the processor is working on. The default value for this RESET is 1.

Fig. C.2 illustrates the steps involved in processor GENA PROC to create the initial population.

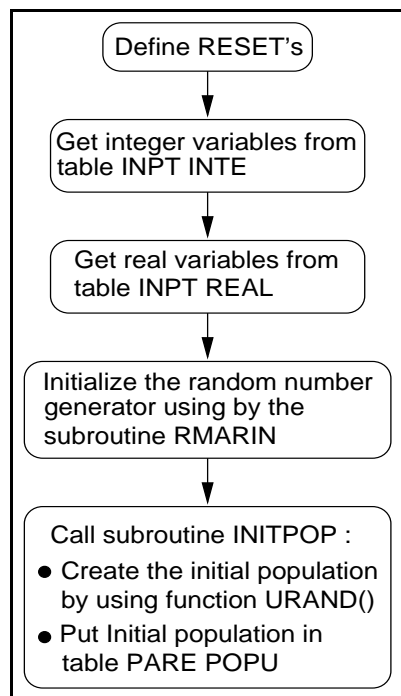


Figure C.2 The GENA PROC.

Fortran Code for the GENA PROC Processor

```

C-----C
C This processor was written by Alexander Hanuska on 10/19/1997 based on the C
C genetic algorithm developed by Sandrine Garcia on 03/16/1996 and is C
C protected under the GPL (GNU General Public License). A copy of GPL can C
C found at the following address: www.gnu.ai.mit.edu. It was designed to C
C be compiled and linked with EAL (Engineering Analysis Language) on a SGI C
C IP22 chipset. C
C-----C

      SUBROUTINE GENETIC1

      PARAMETER (IRSETNUM=4)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1)
      REAL*8 PARMIN,PARMAX,PM

      INTEGER IA(1)
      INTEGER IL,MASK
      INTEGER NC,NPOP,CFSEED1,CFSEED2,SLOPSEED
      INTEGER NBEST1,NBEST2,NBEST3,NG
      INTEGER SEED1,SEED2,NS

      COMMON / CLFLPT /LFLPT
      COMMON / CMINT /LBLCMI
      COMMON / CMFLPT /LBLCMF
      COMMON / RANDOM /SEED1,SEED2
      COMMON / EXPT /NS,NC
      COMMON / CHROMOSP /CP
      COMMON / CHROMOSP2 /CP2
      COMMON / CSTGA /CFSEED1,CFSEED2,SLOPSEED,NG,NPOP,
+ NBEST1,NBEST2,NBEST3,PM
      COMMON / MAXMIN /PARMAX,PARMIN
      COMMON A

      DIMENSION IRESET(IRSETNUM),RRESET(IRSETNUM)
      DIMENSION IL(3,IRSETNUM),RL(IRSETNUM)
      DIMENSION PARMIN(10),PARMAX(10),CP(50,10)

      EQUIVALENCE (A(1),IA(1))

      DATA MASK / 4HMASK /
      DATA IL / 4HNUIN,0,1,
+ 4HNOUT,0,1,
+ 4HNTIM,0,1,
+ 4HITPP,0,1 /

      CALL OPN1(4HGENE)

      IIRSETNUM = IRSETNUM

      CALL XRESET(IRESET,RRESET,IL,RL,IIRSETNUM)
      NUIN = IRESET(1)
      NOUT = IRESET(2)
      NTIM = IRESET(3)
      ITERPOP = IRESET(4)

C The following call statement looks for table INPT INTE in
C library 1, and reads in the values for some of the variables
C used in this genetic algorithm.

      IEA = 2
      INP = 1

      CALL DAL(1,11,IA(INP),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,ITYPE,
+ 4HINPT,4HINTE,MASK,MASK)

```

```

NI      = LB / NJ
NBLK    = NWDS/ LB
NOUTST  = INP+NWDS

DO 10 J=1,NJ
    IPT      = INP+(J-1)*NI
    IOUTPT   = NOUTST+(J-1)*NI
    IA(IOUTPT) = IA(IPT)
10      CONTINUE

NC      = IA(1)
NPOP    = IA(2)
CFSEED1 = IA(3)
CFSEED2 = IA(4)
SLOPSEED = IA(5)
NBEST1  = IA(6)
NBEST2  = IA(7)
NBEST3  = IA(8)
NG      = IA(9)
SEED1   = CFSEED1-SLOPSEED*(ITERPOP-1)
SEED2   = CFSEED2-SLOPSEED*(ITERPOP-1)
NS      = NBEST1*NPOP

IEA = 2
INP = 1

C      The following call statement looks for table INPT REAL in
C      library 1, and reads in the values for some of the variables
C      used in this genetic algorithm

CALL DAL(1,11,A(INP),KORE,IEA,KADNUM,IERR,NWDS,NJ,LB,
+        ITYPE,4HINPT,4HREAL,MASK,MASK)

NI      = LB / NJ
NBLK    = NWDS/ LB
NOUTST  = INP+NWDS

DO 20 J=1,NJ
    IPT      = INP+(J-1)*NI
    IOUTPT   = NOUTST+(J-1)*NI
    A(IOUTPT) = A(IPT)
20      CONTINUE

PARMIN(1) = A(1)
PARMIN(2) = A(2)
PARMIN(3) = A(3)
PARMIN(4) = A(4)
PARMIN(5) = A(5)
PARMIN(6) = A(6)
PARMIN(7) = A(7)
PARMIN(8) = A(8)
PARMIN(9) = A(9)
PARMIN(10) = A(10)
PARMAX(1) = A(11)
PARMAX(2) = A(12)
PARMAX(3) = A(13)
PARMAX(4) = A(14)
PARMAX(5) = A(15)
PARMAX(6) = A(16)
PARMAX(7) = A(17)
PARMAX(8) = A(18)
PARMAX(9) = A(19)
PARMAX(10) = A(20)
PM        = A(21)

IF(NTIM .EQ. 1) THEN

```



```

WRITE(*,*)'-----'
WRITE(*,*)'YOU HAVE ENTERED THE FOLLOWING PARAMETERS:'
WRITE(*,*)'-----'
WRITE(*,*)'NC      =',NC
WRITE(*,*)'NPOP    =',NPOP
WRITE(*,*)'CFSEED1 =',CFSEED1
WRITE(*,*)'CFSEED2 =',CFSEED2
WRITE(*,*)'SLOPSEED =',SLOPSEED
WRITE(*,*)'NBEST1  =',NBEST1
WRITE(*,*)'NBEST2  =',NBEST2
WRITE(*,*)'NBEST3  =',NBEST3
WRITE(*,*)'NG      =',NG
WRITE(*,*)'SEED1   =',SEED1
WRITE(*,*)'SEED2   =',SEED2
WRITE(*,*)'NS      =',NS
WRITE(*,*)
WRITE(*,*)'PARMIN1 =',PARMIN(1)
WRITE(*,*)'PARMIN2 =',PARMIN(2)
WRITE(*,*)'PARMIN3 =',PARMIN(3)
WRITE(*,*)'PARMIN4 =',PARMIN(4)
WRITE(*,*)'PARMIN5 =',PARMIN(5)
WRITE(*,*)'PARMIN6 =',PARMIN(6)
WRITE(*,*)'PARMIN7 =',PARMIN(7)
WRITE(*,*)'PARMIN8 =',PARMIN(8)
WRITE(*,*)'PARMIN9 =',PARMIN(9)
WRITE(*,*)'PARMIN10 =',PARMIN(10)
WRITE(*,*)'PARMAX1 =',PARMAX(1)
WRITE(*,*)'PARMAX2 =',PARMAX(2)
WRITE(*,*)'PARMAX3 =',PARMAX(3)
WRITE(*,*)'PARMAX4 =',PARMAX(4)
WRITE(*,*)'PARMAX5 =',PARMAX(5)
WRITE(*,*)'PARMAX6 =',PARMAX(6)
WRITE(*,*)'PARMAX7 =',PARMAX(7)
WRITE(*,*)'PARMAX8 =',PARMAX(8)
WRITE(*,*)'PARMAX9 =',PARMAX(9)
WRITE(*,*)'PARMAX10 =',PARMAX(10)
WRITE(*,*)'PM      =',PM
WRITE(*,*)'-----'

ENDIF

CALL RMARIN(SEED1,SEED2)
CALL INITPOP
CALL FIN(0,0)

RETURN
END
END

C-----C
C C C
C SUBROUTINE RMARIN(SEED1,SEED2) C
C C
C This is the initialization routine for the random number generator RANMAR() C
C Note: the seed variables can have values between: C
C 0 <= IJ <= 31328 C
C 0 <= KL <= 30081 C
C The random number sequences created by these two seeds are of sufficient C
C length to complete an entire calculation with. For example, if several C
C different groups are working on different parts of the same calculation, C
C each group could be assigned its own IJ seed. This would leave each group C
C with 30000 choices for the second seed. That is to say, this random number C
C generator can create 900 million different subsequences, with each C
C subsequence having a length of approximately  $10^{30}$ . C
C C
C Use IJ = 1802 and KL = 9373 to test the random number generator. The C
C subroutine RANMAR should be used to generate 20000 random numbers. Then C

```

```

C display the nxt six random numbers generated multiplied by 4096*4096. If C
C the random number generator is working properly, the random numbers should C
C be: C
C 6533892.0 14220222.0 7275067.0 6172232.0 8354498.0 10633180.0 C
C-----C

```

```

SUBROUTINE RMARIN(IJ,KL)

IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 U(97),C,CD,CM

INTEGER          I97,J97

COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

LOGICAL TEST

TEST = .FALSE.

IF(IJ .LT. 0 .OR. IJ .GT. 31328 .OR.
+  KL .LT. 0 .OR. KL .GT. 30081) THEN
      WRITE(*,*)'THE FIRST RANDOM NUMBER SEED MUST HAVE'
      WRITE(*,*)'A VALUE BETWEEN 0 AND 31328'
      WRITE(*,*)
      WRITE(*,*)'THE SECOND RANDOM NUMBER SEED MUST HAVE'
      WRITE(*,*)'A VALUE BETWEEN 0 AND 30081'
ENDIF

I = MOD(IJ/177, 177) +2
J = MOD(IJ      , 177) +2
K = MOD(KL/169, 178) +1
L = MOD(KL      , 169)

DO 30 II = 1,97
      S = 0.0
      T = 0.5
      DO 40 JJ = 1,24
            M = MOD(MOD(I*J, 179)*K, 179)
            I = J
            J = K
            K = M
            L = MOD(53*L+1, 169)
            IF(MOD(L*M, 64) .GE. 32) THEN
                  S = S+T
            ENDIF
            T = 0.5*T
40      CONTINUE
            U(II) = S
30      CONTINUE

      C  = 362436.0 / 16777216.0
      CD = 7654321.0 / 16777216.0
      CM = 16777213.0 / 16777216.0
      I97 = 97
      J97 = 33

      TEST = .TRUE.

      RETURN
END

```

```

C-----C
C C
C FUNCTION URAND() C
C C
C Function URAND() is used to initialize th population. This a part C
C of the random number generator proposed by Gearoge Marsaglia in C
C Florida State University report number FSU-SCRI-87-50. C

```

C-----C

```

REAL*8 FUNCTION URAND()

IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 U(97),C,CD,CM

INTEGER I97,J97

COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

LOGICAL TEST

IF(.NOT.TEST) THEN
      WRITE(*,*)'URAND ERROR#1: MUST CALL THE INITIALIZATION ROUTINE'
      WRITE(*,*)'RMARIN BEFORE CALLING URAND.'
ENDIF
UNI = U(I97)-U(J97)
IF(UNI .LT. 0.0) THEN
      UNI = UNI + 1.0
ENDIF
U(I97) = UNI
I97 = I97 -1
IF(I97 .EQ. 0) THEN
      I97 = 97
ENDIF
J97 = J97 - 1
IF(J97 .EQ. 0) THEN
      J97 = 97
ENDIF
C = C - CD
IF(C .LT. 0.0) THEN
      C = C + CM
ENDIF
UNI = UNI - C
IF(UNI .LT. 0.0) THEN
      UNI = UNI + 1.0
ENDIF

URAND = UNI

RETURN
END

```

C-----C

C

C SUBROUTINE INITPOP. C

C

C This subroutine creates the initial population of NC parameters C

C of length NS. The initial population is transferred into table C

C PARE POPU and is located in library 1 in EAL. C

C-----C

```

SUBROUTINE INITPOP

REAL*8      B(1)
REAL*8      URAND,CP(50,10)
REAL*8      PARMAX,PARMIN

INTEGER      IB(1)
INTEGER      SEED1,SEED2,NS,NC

COMMON / RANDOM / SEED1,SEED2
COMMON / EXPT / NS,NC
COMMON / CHROMOSP / CP
COMMON / MAXMIN / PARMAX,PARMIN
COMMON      B

```

```

      DIMENSION PARMAX(10),PARMIN(10),CP(50,10)

      EQUIVALENCE      (B(1),IB(1))

      DO I=1,NS
        DO J=1,NC
          CP(I,J) =(PARMAX(J)-PARMIN(J))*URAND()+PARMIN(J)
        ENDDO
      ENDDO

      K = 0
      DO 50 J = 1,NC
        DO 60 I = 1,NS
          B(I+K) = CP(I,J)
        60 CONTINUE
        K = K + NS
      50 CONTINUE

      IEA      = 2
      INP      = 1
      ITYPE    = -2
      LB       = NS*NC
      NJ       = NS
      NI       = NC
      NWDS     = LB
      NBLK     = 1
      NOUTST   = INP+LB
      COUNT    = NOUTST

      DO 70 I = 1,NS
        DO 80 J = 1,NC
          B(COUNT+(J-1))=B((J-1)*NS+I)
          COUNT1 = COUNT+(J-1)
        80 CONTINUE
        COUNT = COUNT1+1
      70 CONTINUE

      CALL DAL(1,1,B(NOUTST),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,ITYPE,
+          4HPARE,4HPOPU,1,1)

      RETURN
      END

```

C.2.2 The GENB PROC Processor

This processor creates the elitist initial parent population from the first n_{pop} parent populations, by taking the n_{best1} best chromosomes which have previously been ranked according to the performance of their objective functions. Depending on the application, the processor ranks the objective function in descending order for an optimization scheme and in ascending order for an estimation scheme, as do the other genetic processors. The user

needs to adapt the subroutine accordingly to fit either application. The RESET options available are:

NUIN: Determines the input library. The processor will go to library NUIN to look for any datasets or tables it needs to complete its task. The default value for this RESET is 1.

NOUT: Determines the output library. Any tables or datasets that are created by the processor will be written to the NOUT library. The default value for this RESET is 1.

ITRG: Determines the iteration number the processor is working on. The default value for this RESET is 1.

ITPP: Determines which parent population the processor is working on. The default value for this RESET is 1.

Fig. C.3 illustrates the steps involved in processor GENB PROC to create the elitist parent population and rank the chromosomes according to their performance.

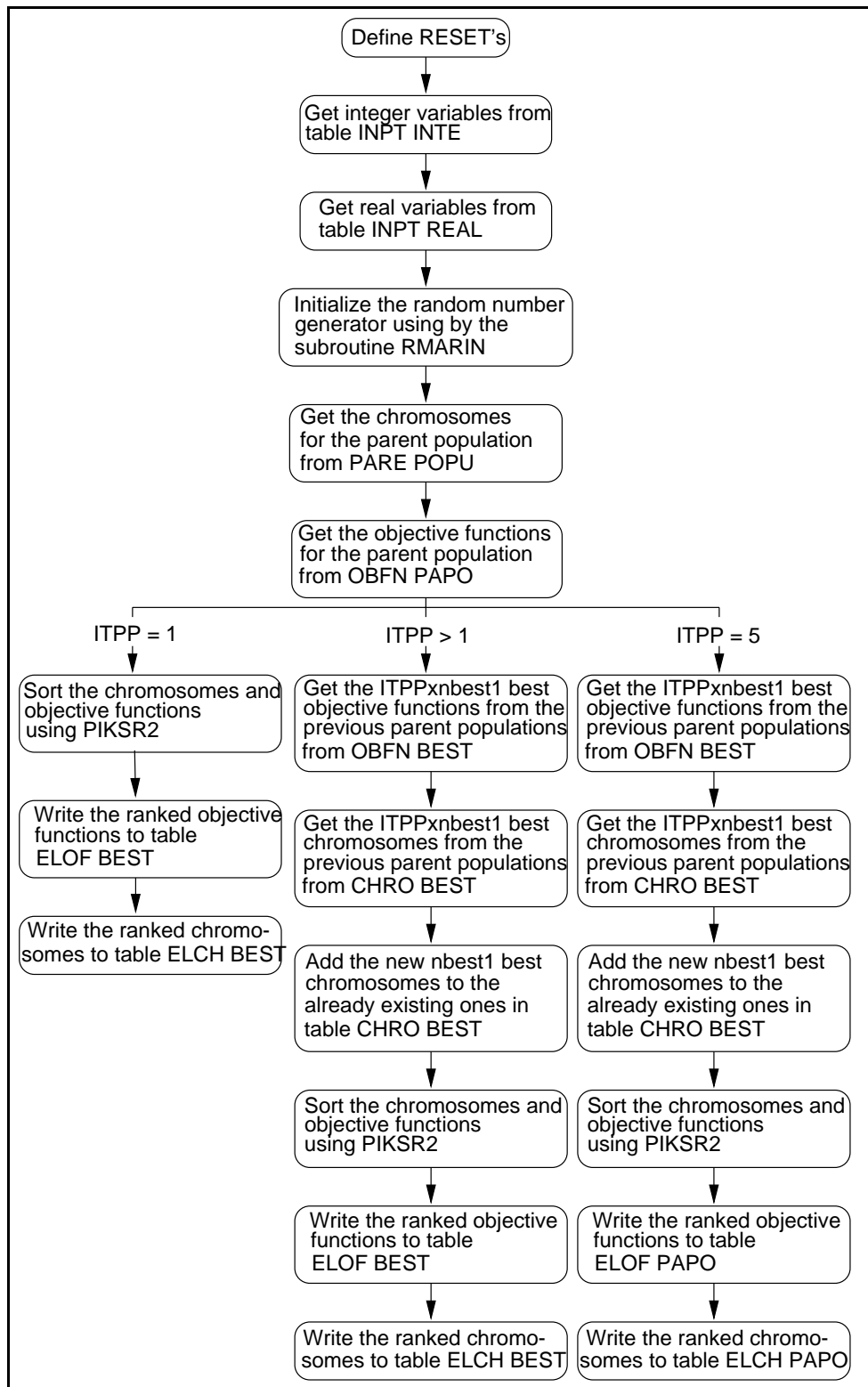


Figure C.3 The GENB PROC.

Fortran Code for the GENB PROC Processor

```

C-----C
C This processor was written by Alexander Hanuska on 10/19/1997 based on the C
C genetic algorithm developed by Sandrine Garcia on 03/16/1996 and is C
C protected under the GPL (GNU General Public License). A copy of GPL can C
C found at the following address: www.gnu.ai.mit.edu. It was designed to C
C be compiled and linked with EAL (Engineering Analysis Language) on a SGI C
C IP22 chipset. C
C-----C

      SUBROUTINE GENETIC2

      PARAMETER (IRSETNUM=3)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1)
      REAL*8 PARMIN,PARMAX,PM
      REAL*8 OF,COPT,OFOPT,OF2,CP2
      REAL*8 P,CHILD

      INTEGER IA(1)
      INTEGER IL,MASK
      INTEGER NC,NPOP,CFSEED1,CFSEED2,SLOPSEED
      INTEGER NBEST1,NBEST2,NBEST3,NG
      INTEGER SEED1,SEED2,NS,PARENT(10)

      COMMON / CLFLPT /LFLPT
      COMMON / CMINT /LBLCFMI
      COMMON / CMFLPT /LBLCFMF
      COMMON / RANDOM /SEED1,SEED2
      COMMON / EXPT /NS,NC
      COMMON / CHROMOSP /CP
      COMMON / CHROMOSP2 /CP2
      COMMON / CSTGA /CFSEED1,CFSEED2,SLOPSEED,NG,NPOP,
+         NBEST1,NBEST2,NBEST3,PM
      COMMON / MAXMIN /PARMAX,PARMIN
      COMMON A

      DIMENSION IRESET(IRSETNUM),RRESET(IRSETNUM)
      DIMENSION IL(3,IRSETNUM),RL(IRSETNUM)
      DIMENSION PARMIN(10),PARMAX(10),CP(50,10)
      DIMENSION OF(50),COPT(50,10),OFOPT(50),OF2(50)
      DIMENSION P(50),CP2(50,10),PARENT(10),CHILD(10)

      EQUIVALENCE (A(1),IA(1))

      DATA MASK / 4HMASK /
      DATA IL / 4HNUIN,0,1,
+         4HNOUT,0,1,
+         4HITPP,0,1 /

      CALL OPN1(4HGENE)

      IIRSETNUM = IRSETNUM
      CALL XRESET(IRESET,RRESET,IL,RL,IIRSETNUM)
      NUIN = IRESET(1)
      NOUT = IRESET(2)
      ITERPOP = IRESET(3)

      IEA = 2
      INP = 1

C The following call statement looks for table INPT INTE in
C library 1, and reads in the values for some of the variables
C used in this genetic algorithm.

      CALL DAL(1,11,IA(INP),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,ITYPE,

```

```

+          4HINPT,4HINTE,MASK,MASK)

NI      = LB / NJ
NBLK    = NWDS/ LB
NOUTST  = INP+NWDS

DO 10 J=1,NJ
    IPT      = INP+(J-1)*NI
    IOUTPT   = NOUTST+(J-1)*NI
    IA(IOUTPT) = IA(IPT)
10  CONTINUE

NC      = IA(1)
NPOP    = IA(2)
CFSEED1 = IA(3)
CFSEED2 = IA(4)
SLOPSEED = IA(5)
NBEST1  = IA(6)
NBEST2  = IA(7)
NBEST3  = IA(8)
NG      = IA(9)
SEED1   = CFSEED1-SLOPSEED*(ITERPOP-1)
SEED2   = CFSEED2-SLOPSEED*(ITERPOP-1)
NS      = NBEST1*NPOP

IEA = 2
INP = 1

C      The following call statement looks for table INPT REAL in
C      library 1, and reads in the values for some of the variables
C      used in this genetic algorithm.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUN,IERR,NWDS,NJ,LB,
+        ITYPE,4HINPT,4HREAL,MASK,MASK)

NI      = LB / NJ
NBLK    = NWDS/ LB
NOUTST  = INP+NWDS

DO 20 J=1,NJ
    IPT      = INP+(J-1)*NI
    IOUTPT   = NOUTST+(J-1)*NI
    A(IOUTPT) = A(IPT)
20  CONTINUE

PARMIN(1) = A(1)
PARMIN(2) = A(2)
PARMIN(3) = A(3)
PARMIN(4) = A(4)
PARMIN(5) = A(5)
PARMIN(6) = A(6)
PARMIN(7) = A(7)
PARMIN(8) = A(8)
PARMIN(9) = A(9)
PARMIN(10) = A(10)
PARMAX(1) = A(11)
PARMAX(2) = A(12)
PARMAX(3) = A(13)
PARMAX(4) = A(14)
PARMAX(5) = A(15)
PARMAX(6) = A(16)
PARMAX(7) = A(17)
PARMAX(8) = A(18)
PARMAX(9) = A(19)
PARMAX(10) = A(20)
PM        = A(21)

CALL RMARIN(SEED1,SEED2)

```



```

        IEA = 2
        INP = 1

C      The following call statement looks for table PARE POPU in
C      library 1, and reads in the values of the chromosomes of
C      the parent population.

        CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HPARE,4HPOPU,MASK,MASK)

        K=1

        DO I=1,NJ,1
          DO J=1,NC
            CP(I,J)=A(K)
            K=K+1
          ENDDO
        ENDDO

        IEA = 2
        INP = 1

C      The following call statement looks for table PARE POPU in
C      library 1, and reads in the values of the objective functions
C      of the parent population.

        CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HOBFN,4HPAPO,MASK,MASK)

        DO I=1,NJ,1
          OF(I)=A(I)
        ENDDO

        CALL PIKSR2(NS,NC,OF,CP)

        WRITE(*,*)'-----'
        WRITE(*,*)'THE PARENT POPULATION AT ITERPOP:',ITERPOP
        WRITE(*,*)'CAN BE DESCRIBED AS FOLLOWS:'
        CALL DESCRIBE(NS,OF)
        WRITE(*,*)'-----'

        IF(ITERPOP .EQ. 1) THEN
          DO I = 1,NBEST1,1
            OFOPT(I) = OF(I)
            A(I)      = OFOPT(I)
          ENDDO

          IEA = 2
          INP = 1
          ITYPE = -2
          NWDS = NBEST1
          NJ = NBEST1
          LB = NBEST1

C      The following call statement creates a new table OBFN BEST and,
C      after having ranked the chromosomes along with their objective
C      function, writes the objective functions to that table in
C      library 1.

          CALL DAL(1,1,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+            ITYPE,4HOBFN,4HBEST,1,1)

          DO I=1,NBEST1
            DO J=1,NC
              COPT(I,J)=CP(I,J)
            ENDDO
          ENDDO

```

```

      K = 0

      DO 100 I=1,NBEST1
        DO 110 J=1,NC
          A(J+K) = CP(I,J)
110        CONTINUE
          K = K + NC
100      CONTINUE

      IEA      = 2
      INP      = 1
      ITYPE    = -2
      LB       = NBEST1*NC
      NJ       = NBEST1
      NI       = NC
      NWDS     = LB
      NBLK     = 1
      NOUTST   = INP + LB
      COUNT    = NOUTST

      DO 120 I = 1,NJ
        DO 125 J = 1,NC
          A(COUNT+(J-1))=A(J+NC*(I-1))
          COUNT1 = COUNT+(J-1)
125        CONTINUE
          COUNT = COUNT1+1
120      CONTINUE

C      The following call statement creates a new table CHRO BEST and
C      after having ranked the chromosomes along with their objective
C      function, writes the NBEST1 chromosomes to that table in
C      library 1.

      CALL DAL(1,1,A(INP),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HCHRO,4HBEST,1,1)
      ENDIF

      IF(ITERPOP .GT. 1) THEN

        IEA = 2
        INP = 1

C      The following call statement looks for table OBFN BEST in
C      library 1, and reads the values into memory.

      CALL DAL(1,11,A(INP),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HOBFN,4HBEST,1,1)

      DO I = 1,NBEST1,1
        OFOPT(I) = OF(I)
        A(I+NJ) = OFOPT(I)
      ENDDO

      NOUTST = INP

      DO J = 1,(NJ+NBEST1),1
        IPT = INP+(J-1)
        IOUTPT = NOUTST+(J-1)
        A(IOUTPT) = A(IPT)
      ENDDO

      NJ = NJ + NBEST1
      LB = LB + NBEST1
      NWDS = NWDS + NBEST1
      ITYPE = -2

C      The following call statement writes to the already existing

```

```

C      table OBFN BEST in library 1.  It adds the NBEST1 additional
C      best to the table from initial populations 2, 3, 4, ...ITERPOP

      CALL DAL(1,1,A(NOUTST),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HOBFN,4HBEST,1,1)

      DO I=1,NBEST1
      DO J=1,NC
      COPT(I,J)=CP(I,J)
      ENDDO
      ENDDO

      IEA =2
      INP =1

C      The following call statement looks for the already existing
C      table CHRO BEST in library 1, and reads the values into
C      memory.

      CALL DAL(1,11,A(INP),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HCHRO,4HBEST,1,1)

      K = 0

      DO 130 I=1,NBEST1
      DO 140 J=1,NC
      A(J+NWDS+K) = CP(I,J)
140    CONTINUE
      K = K + NC
130    CONTINUE

      IYTPE = -2
      LB    = LB + NC*NBEST1
      NJ    = NJ + NBEST1
      NI    = NC
      NWDS  = NWDS + NC*NBEST1
      NBLK  = 1
      NOUTST = INP
      COUNT = NOUTST

      DO 132 I = 1,NJ
      DO 134 J = 1,NC
      A(COUNT+(J-1))=A(J+NC*(I-1))
      COUNT1 = COUNT+(J-1)
134    CONTINUE
      COUNT = COUNT1+1
132    CONTINUE

C      The following call statement writes the NBEST1 additional
C      chromosomes into the already existing table CHRO BEST
C      located in library 1.

      CALL DAL(1,1,A(NOUTST),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HCHRO,4HBEST,1,1)
      ENDIF

      IF(ITERPOP .EQ. 5) THEN

      IEA = 2
      INP = 1

C      The following call statement looks for the already existing
C      table OBFN BEST in library 1 and reads the values into
C      memory.

      CALL DAL(1,11,A(INP),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HOBFN,4HBEST,1,1)

```

```

DO I=1,NJ,1
  OF(I)=A(I)
ENDDO

IEA = 2
INP = 1

C      The following call statement looks for the already existing
C      table CHRO BEST in library 1, and reads the values into memory

      CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HCHRO,4HBEST,1,1)

      K=1

      DO I=1,NJ,1
        DO J=1,NC
          CP(I,J)=A(K)
          K=K+1
        ENDDO
      ENDDO
ENDIF

      CALL PIKSR2(NS,NC,OF,CP)

      DO I=1,NJ
        A(I) = OF(I)
      ENDDO

      IEA      = 2
      INP      = 1
      NJ       = ITERPOP*NBEST1
      NWDS     = ITERPOP*NBEST1
      LB       = ITERPOP*NBEST1
      TYPE     = -2
      NOUTST   = INP

      DO I = 1,NWDS
        IPT      = INP+(I-1)
        IOUTPT   = NOUTST+(I-1)
        A(IOUTPT) = A(IPT)
      ENDDO

C      The following call statement creates a new table ELOF PAPO
C      in library 1, and writes the NBEST1*ITERPOP best objective
C      functions in it.

      CALL DAL(1,1,A(NOUTST),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HELOF,4HPAPO,1,1)

      K = 0

      DO 200 J=1,ITERPOP*NBEST1
        DO 210 I=1,NC
          A(I+K) = CP(J,I)
210      CONTINUE
        K = K + NC
200      CONTINUE

      IEA      = 2
      INP      = 1
      IYTP     = -2
      LB       = NBEST1*ITERPOP*NC
      NJ       = NBEST1*ITERPOP
      NI       = NC
      NWDS     = LB
      NOUTST   = INP
      COUNT    = NOUTST

```

```

DO 202 I = 1,NJ
DO 204 J = 1,NC
A(COUNT+(J-1))=A(J+NC*(I-1))
COUNT1 = COUNT+(J-1)
204 CONTINUE
COUNT = COUNT1+1
202 CONTINUE

C The following call statement creates a new table ELCH PAPO
C in library 1, and writes the NBEST1*ITERPOP best chromosomes
C in it.

CALL DAL(1,1,A(NOUTST),KORE,IEA,KADRRUM,IERR,NWDS,NJ,LB,
+ ITYPE,4HELCH,4HPAPO,1,1)

CALL FIN(0,0)

RETURN
END
END

C-----C
C SUBROUTINE PIKSR2 C
C This subroutine sorts the array of NS into numerical order, by straight C
C insertion, while making the corresponding rearrangements of the array C
C CP(NS,NC). See numerical recipes page 322. C
C-----C

SUBROUTINE PIKSR2(N,M,ARR,BRR)

REAL*8 ARR(50),BRR(50,10),A,B(10),ARR2(50),BRR2(50,10)

INTEGER I,J,K,N,M

DO J=2,N
A=ARR(J)
DO K=1,M
B(K)=BRR(J,K)
ENDDO
DO I=J-1,1,-1
IF(ARR(I) .LE. A) GOTO 90
ARR(I+1)=ARR(I)
DO K=1,M
BRR(I+1,K)=BRR(I,K)
ENDDO
ENDDO
I=0
90 ARR(I+1)=A
DO K=1,M
BRR(I+1,K)=B(K)
ENDDO
ENDDO

RETURN
END

C-----C
C SUBROUTINE RMARIN(SEED1,SEED2) C
C-----C

SUBROUTINE RMARIN(IJ,KL)

IMPLICIT REAL*8 (A-H,O-Z)
REAL*8 U(97),C,CD,CM

INTEGER I97,J97

```

```

COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

LOGICAL TEST

TEST = .FALSE.

IF(IJ .LT. 0 .OR. IJ .GT. 31328 .OR.
+   KL .LT. 0 .OR. KL .GT. 30081) THEN
    WRITE(*,*)'THE FIRST RANDOM NUMBER SEED MUST HAVE'
    WRITE(*,*)'A VALUE BETWEEN 0 AND 31328'
    WRITE(*,*)
    WRITE(*,*)'THE SECOND RANDOM NUMBER SEED MUST HAVE'
    WRITE(*,*)'A VALUE BETWEEN 0 AND 30081'
ENDIF

I = MOD(IJ/177, 177) +2
J = MOD(IJ      , 177) +2
K = MOD(KL/169, 178) +1
L = MOD(KL      , 169)

DO 30 II = 1,97
    S = 0.0
    T = 0.5
    DO 40 JJ = 1,24
        M = MOD(MOD(I*J, 179)*K, 179)
        I = J
        J = K
        K = M
        L = MOD(53*L+1, 169)
        IF(MOD(L*M, 64) .GE. 32) THEN
            S = S+T
        ENDIF
        T = 0.5*T
40    CONTINUE
    U(II) = S
30    CONTINUE

C   = 362436.0 / 16777216.0
CD  = 7654321.0 / 16777216.0
CM  = 16777213.0 / 16777216.0
I97 = 97
J97 = 33

TEST = .TRUE.

RETURN
END

C-----C
C FUNCTION URAND()                                C
C-----C

REAL*8 FUNCTION URAND()

IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 U(97),C,CD,CM

INTEGER I97,J97

COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

LOGICAL TEST

IF(.NOT.TEST) THEN
    WRITE(*,*)'URAND ERROR#1: MUST CALL THE INITIALIZATION ROUTINE'
    WRITE(*,*)'RMARIN BEFORE CALLING URAND.'
ENDIF

```

```

      UNI = U(I97)-U(J97)
      IF(UNI .LT. 0.0) THEN
        UNI = UNI + 1.0
      ENDIF
      U(I97) = UNI
      I97 = I97 -1
      IF(I97 .EQ. 0) THEN
        I97 = 97
      ENDIF
      J97 = J97 - 1
      IF(J97 .EQ. 0) THEN
        J97 = 97
      ENDIF
      C = C - CD
      IF(C .LT. 0.0) THEN
        C = C + CM
      ENDIF
      UNI = UNI - C
      IF(UNI .LT. 0.0) THEN
        UNI = UNI + 1.0
      ENDIF

      URAND = UNI

      RETURN
      END
C-----C
C SUBROUTINE DESCRIBE(NS,OF)                                C
C This subroutine computes OFavg of a generation              C
C-----C

      SUBROUTINE DESCRIBE(N,OF)

      INTEGER N

      REAL*8  OF(50),OFAVG

      OFAVG = 0.DO

      DO I=1,N
        OFAVG = OFAVG + OF(I)
      ENDDO

      OFAVG = OFAVG/N
      WRITE(*,*)'THE AVERAGE OF THE OBJECTIVE FUNCTION IS:'
      WRITE(*,*)'OFAVG=',OFAVG

      RETURN
      END

```

C.2.3 The GENC PROC Processor

This processor creates the children population from the elitist initial parent population or the ITER ATIO population. It performs a test to determine the fitness of each parent, chooses two parents depending on their level of fitness, and implements single point cross-overs and mutations to create a children population of size n_s . Both the parent and children population are combined to form the paren-children population. The RESET options

available are:

NUIN: Determines the input library. The processor will go to library NUIN to look for any datasets or tables it needs to complete its task. The default value for this RESET is 1.

NOUT: Determines the output library. Any tables or datasets that are created by the processor will be written to the NOUT library. The default value for this RESET is 1.

ITRG: Determines the iteration number the processor is working on. The default value for this RESET is 1.

ITPP: Determines which parent population the processor is working on. The default value for this RESET is 1.

Fig. C.4 illustrates the steps involved in processor GENC PROC to create the children and parent-children populations.

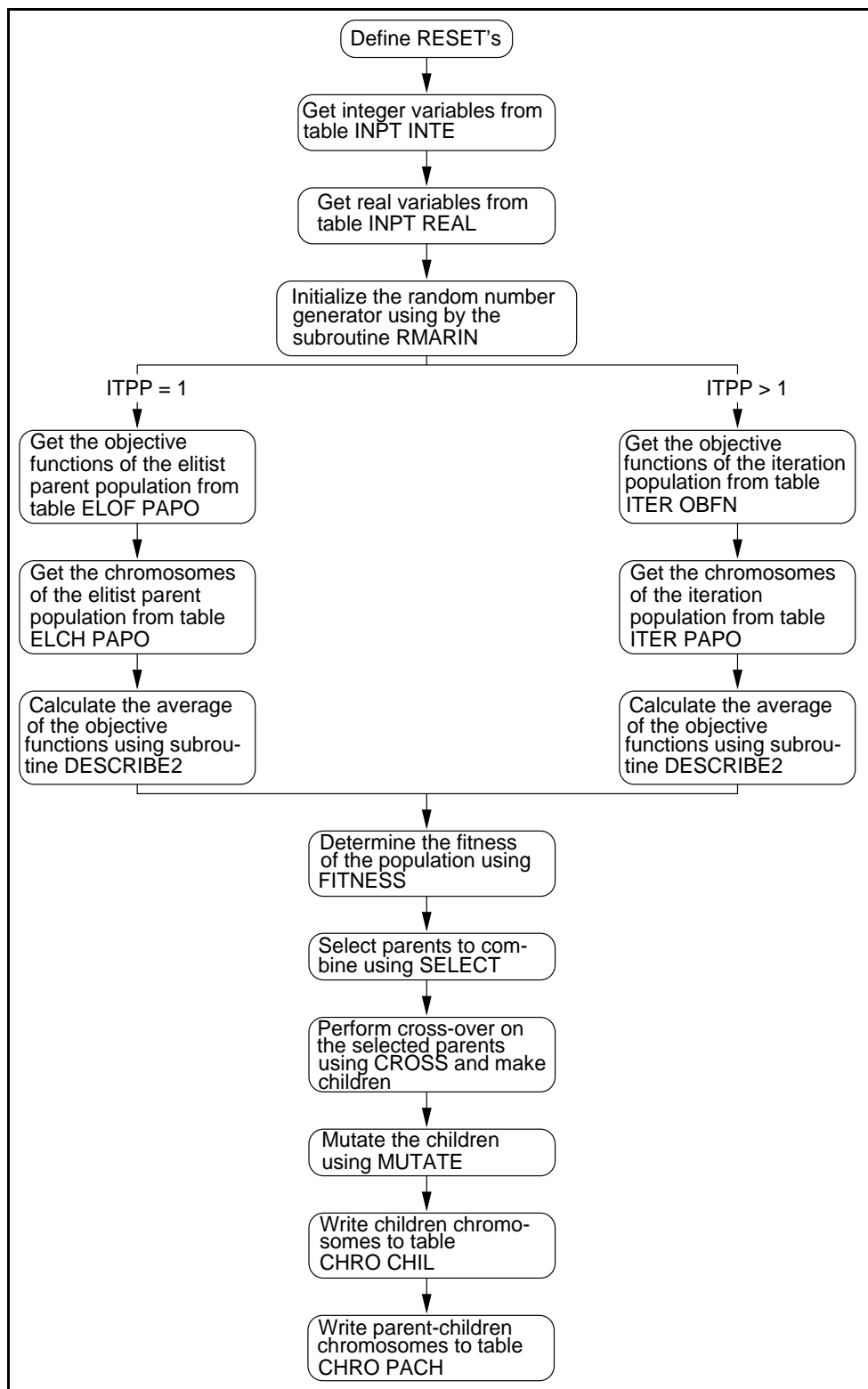


Figure C.4 The GENC PROC.

Fortran Code for the GENC PROC Processor

```

C-----C
C This processor was written by Alexander Hanuska on 10/19/1997 based on the C
C genetic algorithm developed by Sandrine Garcia on 03/16/1996 and is C
C protected under the GPL (GNU General Public License). A copy of GPL can C
C found at the following address: www.gnu.ai.mit.edu. It was designed to C
C be compiled and linked with EAL (Engineering Analysis Language) on a SGI C
C IP22 chipset. C
C-----C

      SUBROUTINE GENETIC3

      PARAMETER (IRSETNUM=4)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1)
      REAL*8 PARMIN,PARMAX,PM
      REAL*8 OF,COPT,OFOPT,OF2,CP2
      REAL*8 P,CHILD

      INTEGER IA(1)
      INTEGER IL,MASK
      INTEGER NC,NPOP,CFSEED1,CFSEED2,SLOPSEED
      INTEGER NBEST1,NBEST2,NBEST3,NG
      INTEGER SEED1,SEED2,NS,PARENT(10),NMUTATE

      COMMON / CLFLPT /LFLPT
      COMMON / CMINT /LBLCFMI
      COMMON / CMFLPT /LBLCFMF
      COMMON / RANDOM /SEED1,SEED2
      COMMON / EXPT /NS,NC
      COMMON / CHROMOSP /CP
      COMMON / CHROMOSP2 /CP2
      COMMON / CSTGA /CFSEED1,CFSEED2,SLOPSEED,NG,NPOP,
+         NBEST1,NBEST2,NBEST3,PM
      COMMON / MAXMIN /PARMAX,PARMIN
      COMMON A

      DIMENSION IRESET(IRSETNUM),RRESET(IRSETNUM)
      DIMENSION IL(3,IRSETNUM),RL(IRSETNUM)
      DIMENSION PARMIN(10),PARMAX(10),CP(50,10)
      DIMENSION OF(50),COPT(50,10),OFOPT(50),OF2(100)
      DIMENSION P(50),CP2(100,10),PARENT(10),CHILD(10)

      EQUIVALENCE (A(1),IA(1))

      DATA MASK / 4HMASK /
      DATA IL / 4HNUIN,0,1,
+         4HNOUT,0,1,
+         4HITPP,0,1,
+         4HITRG,0,1 /

      CALL OPN1(4HGENE)

      IIRSETNUM = IRSETNUM
      CALL XRESET(IRESET,RRESET,IL,RL,IIRSETNUM)
      NUIN = IRESET(1)
      NOUT = IRESET(2)
      ITERPOP = IRESET(3)
      ITERG = IRESET(4)

      IEA = 2
      INP = 1

C The following call statement looks for table INPT INTE in
C library 1, and reads in the values for some of the variables

```

```

C      used in this genetic algorithm.

      CALL DAL(1,11,IA(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,ITYPE,
+         4HINPT,4HINTE,MASK,MASK)

      NI      = LB / NJ
      NBLK    = NWDS/ LB
      NOUTST  = INP+NWDS

      DO 10 J=1,NJ
         IPT      = INP+(J-1)*NI
         IOUTPT   = NOUTST+(J-1)*NI
         IA(IOUTPT) = IA(IPT)
10      CONTINUE

      NC      = IA(1)
      NPOP    = IA(2)
      CFSEED1  = IA(3)
      CFSEED2  = IA(4)
      SLOPSEED = IA(5)
      NBEST1   = IA(6)
      NBEST2   = IA(7)
      NBEST3   = IA(8)
      NG       = IA(9)
      SEED1    = CFSEED1-SLOPSEED*(ITERPOP-1)
      SEED2    = CFSEED2-SLOPSEED*(ITERPOP-1)
      NS       = NBEST1*NPOP

      IEA = 2
      INP = 1

C      The following call statement looks for table INPT REAL in
C      library 1, and reads in the values for some of the variables
C      used in this genetic algorithm.

      CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HINPT,4HREAL,MASK,MASK)

      NI      = LB / NJ
      NBLK    = NWDS/ LB
      NOUTST  = INP+NWDS

      DO 20 J=1,NJ
         IPT      = INP+(J-1)*NI
         IOUTPT   = NOUTST+(J-1)*NI
         A(IOUTPT) = A(IPT)
20      CONTINUE

      PARMIN(1) = A(1)
      PARMIN(2) = A(2)
      PARMIN(3) = A(3)
      PARMIN(4) = A(4)
      PARMIN(5) = A(5)
      PARMIN(6) = A(6)
      PARMIN(7) = A(7)
      PARMIN(8) = A(8)
      PARMIN(9) = A(9)
      PARMIN(10) = A(10)
      PARMAX(1) = A(11)
      PARMAX(2) = A(12)
      PARMAX(3) = A(13)
      PARMAX(4) = A(14)
      PARMAX(5) = A(15)
      PARMAX(6) = A(16)
      PARMAX(7) = A(17)
      PARMAX(8) = A(18)
      PARMAX(9) = A(19)
      PARMAX(10) = A(20)

```

```

PM          = A(21)

CALL RMARIN(SEED1,SEED2)

IF(ITERG .EQ. 1) THEN

    IEA = 2
    INP = 1

C    The following call statement looks for table ELOF PAPO in
C    library 1, and reads in the values of the objective functions
C    of the elitist parent population.

    CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HELOF,4HPAPO,1,1)

    DO I=1,NWDS,1
        OF(I)=A(I)
    ENDDO

    IEA = 2
    INP = 1

C    The following call statement looks for table ELCH PAPO in
C    library 1, and reads in the values of the chromosomes of
C    the elitist parent population.

    CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HELCH,4HPAPO,1,1)

    K=1

    DO I=1,NJ,1
        DO J=1,NC
            CP(I,J)=A(K)
            K=K+1
        ENDDO
    ENDDO

    WRITE(*,*)'-----'
    WRITE(*,*)'THE INITIAL ELITIST PARENT POPULATION CAN BE'
    WRITE(*,*)'DESCRIBED AS FOLLOWS:'
    ITERG = ITERG - 1
    WRITE(*,*)'AT ITERATION=',ITERG
    CALL DESCRIBE2(NS,OF)
    WRITE(*,*)'-----'
    WRITE(*,*)'I AM WORKING ON THE ELITIST PARENT POPULATION'
    ITERG = ITERG + 1
    WRITE(*,*)'AT ITERATION=',ITERG
    NMUTATE = 0

ENDIF

IF(ITERG .GT. 1) THEN

C    The following call statement looks for table ITER OBFN in
C    library 1, and reads in the values of the objective functions
C    of the iteration generation.

    CALL DAL(1,11,A(INP),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HITER,4HOBFN,1,1)

    DO I=1,NWDS,1
        OF(I)=A(I)
    ENDDO

    IEA = 2
    INP = 1

```

```

C      The following call statement looks for table ITER PAPO in
C      library 1, and reads in the values of the chromosomes of
C      the iteration generation.

      CALL DAL(1,11,A(INP),KORE,IEA,KADNRUM,IERR,NWDS,NJ,LB,
+        ITYPE,4HITER,4HPAPO,1,1)

      K=1

      DO I=1,NJ,1
        DO J=1,NC
          CP(I,J)=A(K)
          K=K+1
        ENDDO
      ENDDO

      ITERG = ITERG
      WRITE(*,*)'-----'
      WRITE(*,*)'I AM WORKING ON THE ELITIST PARENT POPULATION'
      WRITE(*,*)'AT ITERATION=',ITERG
      WRITE(*,*)'THE POPULATION CAN BE DESCRIBED AS FOLLOWS:'
      CALL DESCRIBE2(NS,OF)
      WRITE(*,*)'-----'
      NMUTATE = 0
    ENDIF

    DO I=1,NS
      OF2(I)=OF(I)
      DO J=1,NC
        CP2(I,J)=CP(I,J)
      ENDDO
    ENDDO

    CALL FITNESS(NS,P)

    DO 150 I=1,NS
      CALL SELECT(NS,P,PARENT)
      CALL CROSS(PARENT,NC,CHILD)
      CALL MUTATE(NC,PM,NMUTATE,CHILD)

      DO 155 J=1,NC
        CP2(I+NS,J)= CHILD(J)
        CP(I,J) = CHILD(J)
155      CONTINUE
150    CONTINUE

      WRITE(*,*)'NMUTATE=',NMUTATE

      K = 0

      DO 220 I=1,NS
        DO 210 J=1,NC
          A(J+K) = CP(I,J)
210      CONTINUE
          K = K + NC
220    CONTINUE

      IEA = 2
      INP = 1
      ITYPE = -2
      LB = ITERPOP*NBEST1*NC
      NJ = ITERPOP*NBEST1
      NI = NC
      NWDS = LB
      NBLK = 1
      NOUTST = INP + LB
      COUNT = NOUTST

```

```

DO 120 I = 1,NJ
DO 125 J = 1,NC
    A(COUNT+(J-1))=A(J+NC*(I-1))
    COUNT1 = COUNT+(J-1)
125    CONTINUE
    COUNT = COUNT1+1
120    CONTINUE

C      The following call statement creates a new table CHRO CHIL and,
C      after having ranked the chromosomes along with their objective
C      function, writes the objective functions to that table in
C      library 1.

    CALL DAL(1,1,A(NOUTST),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HCHRO,4HCHIL,1,1)

    K = 0
    DO 300 I=1,2*NS
        DO 310 J=1,NC
            A(J+K) = CP2(I,J)
310        CONTINUE
            K = K + NC
300    CONTINUE

    IEA    = 2
    INP    = 1
    ITYPE  = -2
    LB     = 2*ITERPOP*NBEST1*NC
    NJ     = 2*ITERPOP*NBEST1
    NI     = NC
    NWDS   = LB
    NBLK   = 1
    NOUTST = INP + LB
    COUNT  = NOUTST

    DO 304 I = 1,NJ
        DO 302 J = 1,NC
            A(COUNT+(J-1))=A(J+NC*(I-1))
            COUNT1 = COUNT+(J-1)
302        CONTINUE
            COUNT = COUNT1+1
304    CONTINUE

C      The following call statement creates a new table CHRO PACH and,
C      after having ranked the chromosomes along with their objective
C      function, writes the objective functions to that table in
C      library 1. Here it writes both the parent and children population
C      in the table.

    CALL DAL(1,1,A(NOUTST),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HCHRO,4HPACH,1,1)

    CALL FIN(0,0)
    RETURN
    END
    END

C-----C
C SUBROUTINE DESCRIBE(NS,OF)
C This subroutine computes OFavg of a generation
C-----C

SUBROUTINE DESCRIBE2(N,OF)

INTEGER I,N
REAL*8 CP(50,10),OF(50),OFAVG
COMMON / CHROMOSP / CP

```

```

OFAVG=0.DO
DO I = 1,N
    OFAVG=OFAVG+OF(I)
ENDDO

OFAVG=OFAVG/N
WRITE(*,*)'THE AVERAGE OF THE OBJECTIVE FUNCTION IS:',OFAVG

RETURN
END

C-----C
C SUBROUTINE RMARIN(SEED1,SEED2)                                C
C-----C

SUBROUTINE RMARIN(IJ,KL)

IMPLICIT REAL*8 (A-H,O-Z)
    REAL*8 U(97),C,CD,CM

INTEGER          I97,J97

COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

LOGICAL TEST

TEST = .FALSE.

IF(IJ .LT. 0 .OR. IJ .GT. 31328 .OR.
+   KL .LT. 0 .OR. KL .GT. 30081) THEN
    WRITE(*,*)'THE FIRST RANDOM NUMBER SEED MUST HAVE'
    WRITE(*,*)'A VALUE BETWEEN 0 AND 31328'
    WRITE(*,*)
    WRITE(*,*)'THE SECOND RANDOM NUMBER SEED MUST HAVE'
    WRITE(*,*)'A VALUE BETWEEN 0 AND 30081'
ENDIF

I = MOD(IJ/177, 177) +2
J = MOD(IJ      , 177) +2
K = MOD(KL/169, 178) +1
L = MOD(KL      , 169)

DO 30 II = 1,97
    S = 0.0
    T = 0.5
    DO 40 JJ = 1,24
        M = MOD(MOD(I*J, 179)*K, 179)
        I = J
        J = K
        K = M
        L = MOD(53*L+1, 169)
        IF(MOD(L*M, 64) .GE. 32) THEN
            S = S+T
        ENDIF
        T = 0.5*T
40    CONTINUE
        U(II) = S
30    CONTINUE

C   = 362436.0 / 16777216.0
CD  = 7654321.0 / 16777216.0
CM  = 16777213.0 / 16777216.0
I97 = 97
J97 = 33

TEST = .TRUE.

C   WRITE(*,*)C,CD,CM,I97,J97,U(40)

```

```

        RETURN
        END

C-----C
C FUNCTION URAND()C
C-----C

        REAL*8 FUNCTION URAND()

        IMPLICIT REAL*8 (A-H,O-Z)
                REAL*8 U(97),C,CD,CM

        INTEGER I97,J97

        COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

        LOGICAL TEST

        IF(.NOT.TEST) THEN
                WRITE(*,*)'URAND ERROR#1: MUST CALL THE INITIALIZATION ROUTINE'
                WRITE(*,*)'RMARIN BEFORE CALLING URAND.'
        ENDIF
        UNI = U(I97)-U(J97)
        IF(UNI .LT. 0.0) THEN
                UNI = UNI + 1.0
        ENDIF
        U(I97) = UNI
        I97 = I97 -1
        IF(I97 .EQ. 0) THEN
                I97 = 97
        ENDIF
        J97 = J97 - 1
        IF(J97 .EQ. 0) THEN
                J97 = 97
        ENDIF
        C = C - CD
        IF(C .LT. 0.0) THEN
                C = C + CM
        ENDIF
        UNI = UNI - C
        IF(UNI .LT. 0.0) THEN
                UNI = UNI + 1.0
        ENDIF

        URAND = UNI

        RETURN
        END

C-----C
C
C
C SUBROUTINE MUTATEC
C This subroutine mutates the chromosomes within the probability ofC
C mutation PMC
C
C-----C

        SUBROUTINE MUTATE(NC,PM,NMUTATE,ARR)

        INTEGER SEED1,SEED2,NMUTATE
        INTEGER NC

        REAL*8 URAND,ARR(10),PM,RL(10)
        REAL*8 PARMIN(10),PARMAX(10)

        COMMON / RANDOM / SEED1,SEED2

```



```

COMMON / MAXMIN / PARMAX,PARMIN

DO I=1,NC
  RL(I)=URAND()
  IF(RL(I) .LE. PM) THEN
    ARR(I) = (PARMAX(I)-PARMIN(I))*URAND()+PARMIN(I)
    NMUTATE = NMUTATE + 1
  ENDIF
ENDDO

RETURN
END

C-----C
C SUBROUTINE CROSS C
C-----C

SUBROUTINE CROSS(PARENT,NC,CHILD)

INTEGER SEED1,SEED2,PARENT(2)
INTEGER NC

REAL*8 URAND,CP(50,10),CHILD(10)

COMMON / RANDOM / SEED1,SEED2
COMMON / CHROMOSP /CP

K = INT((NC-1)*URAND()+1)
DO J=1,NC
  IF(J .LE. K) THEN
    CHILD(J)=CP(PARENT(1),J)
  ELSE
    CHILD(J)=CP(PARENT(2),J)
  ENDIF
ENDDO

RETURN
END

C-----C
C SUBROUTINE SELECT C
C-----C

SUBROUTINE SELECT(N,P,PARENT)

INTEGER SEED1,SEED2,N,PARENT(2)

REAL*8 URAND,P(50),PLACE

COMMON / RANDOM /SEED1,SEED2

DO 160 I=1,2
  PLACE=URAND()
  DO 170 J=2,N
    IF (PLACE .GT. P(J-1)) GO TO 170
    PARENT(I)=J-1
    GO TO 160
170    CONTINUE
    PARENT(I)=N
160    CONTINUE

RETURN
END

C-----C
C SUBROUTINE FITNESS C
C-----C

```

```

SUBROUTINE FITNESS(N,P)

INTEGER N
REAL*8 P(50),FI,PROBI

P(1) = 2.D0/(N+1)
DO I = 2,N
    FI = N+1-I
    PROBI = 2*FI/(N*(N+1))
    P(I) = P(I-1)+PROBI
ENDDO

RETURN
END

```

C.2.4 The GEND PROC Processor

This processor ranks the children population removes any twins, and redefines the parameter ranges based on the n_{best3} best chromosomes. Thereafter, $n_s - n_{best2}$ new chromosomes chosen within the new parameter bounds are combined with the n_{best2} best chromosomes from the twin-removed parent-children population to obtain a new generation of size n_s .

NUIN: Determines the input library. The processor will go to library NUIN to look for any datasets or tables it needs to complete its task. The default value for this RESET is 1.

NOUT: Determines the output library. Any tables or datasets that are created by the processor will be written to the NOUT library. The default value for this RESET is 1.

ITRG: Determines the iteration number the processor is working on. The default value for this RESET is 1.

ITPP: Determines which parent population the processor is working on. The default value for this RESET is 1.

Fig. C.5 shows the details of processor GEND PROD.

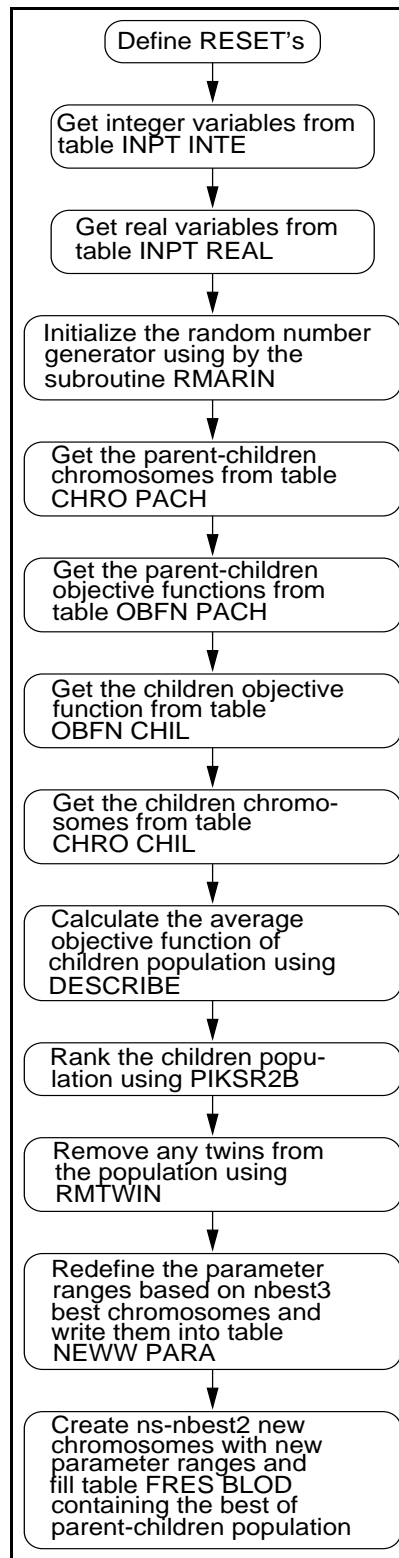


Figure C.5 The GEND PROC.

Fortran Code for the GEND PROC Processor

```

C-----C
C This processor was written by Alexander Hanuska on 10/19/1997 based on the C
C genetic algorithm developed by Sandrine Garcia on 03/16/1996 and is C
C protected under the GPL (GNU General Public License). A copy of GPL can C
C found at the following address: www.gnu.ai.mit.edu. It was designed to C
C be compiled and linked with EAL (Engineering Analysis Language) on a SGI C
C IP22 chipset. C
C-----C

      SUBROUTINE GENETIC4

      PARAMETER (IRSETNUM=4)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1)
      REAL*8 PARMIN,PARMAX,PBREAL,PM
      REAL*8 OF,COPT,OFOPT,OF2,CP2
      REAL*8 P,CHILD

      INTEGER IA(1)
      INTEGER IL,MASK
      INTEGER NC,NPOP,CFSEED1,CFSEED2,SLOPSEED
      INTEGER NBEST1,NBEST2,NBEST3,NG
      INTEGER SEED1,SEED2,NS,PARENT(10)

      COMMON / CLFLPT /LFLPT
      COMMON / CMINT /LBLCFMI
      COMMON / CMFLPT /LBLCFMF
      COMMON / RANDOM /SEED1,SEED2
      COMMON / EXPT /NS,NC
      COMMON / CHROMOSP /CP
      COMMON / CHROMOSP2 /CP2
      COMMON / CSTGA /CFSEED1,CFSEED2,SLOPSEED,NG,NPOP,
+       NBEST1,NBEST2,NBEST3,PM
      COMMON / MAXMIN /PARMAX,PARMIN
      COMMON A

      DIMENSION IRESET(IRSETNUM),RRESET(IRSETNUM)
      DIMENSION IL(3,IRSETNUM),RL(IRSETNUM)
      DIMENSION PARMIN(10),PARMAX(10),CP(50,10)
      DIMENSION OF(50),COPT(50,10),OFOPT(50),OF2(100)
      DIMENSION P(50),CP2(100,10),PARENT(10),CHILD(10)

      EQUIVALENCE (A(1),IA(1))

      DATA MASK / 4HMASK /
      DATA IL / 4HNUIN,0,1,
+       4HNOUT,0,1,
+       4HITPP,0,1,
+       4HITRG,0,0 /

      CALL OPN1(4HGENE)

      IIRSETNUM = IRSETNUM
      CALL XRESET(IRESET,RRESET,IL,RL,IIRSETNUM)
      NUIN = IRESET(1)
      NOUT = IRESET(2)
      ITERPOP = IRESET(3)
      ITERG = IRESET(4)

      IEA = 2
      INP = 1

C The following call statement looks for table INPT INTE in
C library 1, and reads in the values for some of the variables
C used in this genetic algorithm.

```

```

      CALL DAL(1,11,IA(INP),KORE,IEA,KADNRNUM,IERR,NWDS,NJ,LB,ITYPE,
+         4HINPT,4HINTE,MASK,MASK)

      NI      = LB / NJ
      NBLK    = NWDS/ LB
      NOUTST  = INP+NWDS

      DO 10 J=1,NJ
          IPT      = INP+(J-1)*NI
          IOUTPT   = NOUTST+(J-1)*NI
          IA(IOUTPT) = IA(IPT)
10      CONTINUE

      NC      = IA(1)
      NPOP    = IA(2)
      CFSEED1 = IA(3)
      CFSEED2 = IA(4)
      SLOPSEED = IA(5)
      NBEST1  = IA(6)
      NBEST2  = IA(7)
      NBEST3  = IA(8)
      NG      = IA(9)
      SEED1   = CFSEED1-SLOPSEED*(ITERPOP-1)
      SEED2   = CFSEED2-SLOPSEED*(ITERPOP-1)
      NS      = NBEST1*NPOP

      IEA = 2
      INP = 1

C      The following call statement looks for table INPT REAL in
C      library 1, and reads in the values for some of the variables
C      used in this genetic algorithm.

      CALL DAL(1,11,A(INP),KORE,IEA,KADNRNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HINPT,4HREAL,MASK,MASK)

      NI      = LB / NJ
      NBLK    = NWDS/ LB
      NOUTST  = INP+NWDS

      DO 20 J=1,NJ
          IPT      = INP+(J-1)*NI
          IOUTPT   = NOUTST+(J-1)*NI
          A(IOUTPT) = A(IPT)
20      CONTINUE

      PARMIN(1) = A(1)
      PARMIN(2) = A(2)
      PARMIN(3) = A(3)
      PARMIN(4) = A(4)
      PARMIN(5) = A(5)
      PARMIN(6) = A(6)
      PARMIN(7) = A(7)
      PARMIN(8) = A(8)
      PARMIN(9) = A(9)
      PARMIN(10) = A(10)
      PARMAX(1) = A(11)
      PARMAX(2) = A(12)
      PARMAX(3) = A(13)
      PARMAX(4) = A(14)
      PARMAX(5) = A(15)
      PARMAX(6) = A(16)
      PARMAX(7) = A(17)
      PARMAX(8) = A(18)
      PARMAX(9) = A(19)
      PARMAX(10) = A(20)
      PM        = A(21)

```

```

CALL RMARIN(SEED1,SEED2)

IEA = 2
INP = 1

C   The following call statement looks for table CHRO PACH in
C   library1, and reads in the chromosomes of the parent children
C   population.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUNUM,IERR,NWDS,NJ,LB,
+       ITYPE,4HCHRO,4HPACH,1,1)

K=1
DO I=1,NJ
  DO J=1,NC
    CP2(I,J)=A(K)
    K=K+1
  ENDDO
ENDDO

IEA = 2
INP = 1

C   The following call statement looks for table ELOF PAPO in
C   library1, and reads in the objective functions of the elitist
C   parent population.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUNUM,IERR,NWDS,NJ,LB,
+       ITYPE,4HELOF,4HPAPO,1,1)

DO I=1,NJ
  OF2(I)=A(I)
ENDDO

IEA = 2
INP = 1

C   The following call statement looks for table OBFN CHIL in
C   library1, and reads in the objective functions of the children
C   population.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUNUM,IERR,NWDS,NJ,LB,
+       ITYPE,4HOBFN,4HCHIL,1,1)

DO I=1,NWDS,1
  OF(I)=A(I)
ENDDO

IEA = 2
INP = 1

C   The following call statement looks for table CHRO CHIL in
C   library1, and reads in the chromosomes of the children
C   population.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUNUM,IERR,NWDS,NJ,LB,
+       ITYPE,4HCHRO,4HCHIL,1,1)

K=1
DO I=1,NJ,1
  DO J=1,NC
    CP(I,J)=A(K)
    K=K+1
  ENDDO
ENDDO

```

```

DO I=1,NS
  OF2(I+NS)=OF(I)
ENDDO

WRITE(*,*)
WRITE(*,*)'CHILDREN POPULATION:'
CALL DESCRIBE(NS,OF)
CALL PIKSR2B(2*NS,NC,OF2,CP2)
CALL RMTWIN(OF2)

DO I=1,NBEST2
  DO J=1,NC
CP(I,J)=CP2(I,J)
  ENDDO
  OF(I)=OF2(I)
ENDDO

CALL PARRANGE(NBEST3,NC,NS)

DO I =1,NS-NBEST2
  DO J = 1,NC
    CP(I+NBEST2,J)=(PARMAX(J)-PARMIN(J))*URAND()+PARMIN(J)
  ENDDO
ENDDO

k = 0
DO 230 J=1,NS
  DO 240 I=1,NC
    A(I+K) = CP(J,I)
240   CONTINUE
    K=K+NC
230  CONTINUE

IEA      = 2
INP      = 1
ITYPE    = -2
LB       = NC*NS
NJ       = NS
NI       = NC
NWDS     = LB
NBLK     = 1
NOUTST   = INP + LB
COUNT   = NOUTST

DO 232 I = 1,NS
  DO 234 J = 1,NC
    A(COUNT+(J-1))=A(J+NC*(I-1))
    COUNT1 = COUNT+(J-1)
    WRITE(*,*)A(COUNT+(J-1)),A(J+NC*(I-1))
234    CONTINUE
    COUNT = COUNT1 +1
232    CONTINUE

C      The following call statement creates a new table FRES BLOD
C      in library 1, and writes the new chromosomes in it.

      CALL DAL(1,1,A(NOUTST),KORE,IEA,KADARNUM,IERR,NWDS,NJ,LB,
+          ITYPE,4HFRES,4HBLOD,1,1)

CALL FIN(0,0)
RETURN
END
END

```

```

C-----C
C SUBROUTINE PARRANGE                                     C
C-----C

```

```

SUBROUTINE PARRANGE(N,NC,NS)

INTEGER N,NC
REAL*8 CP(50,10)
REAL*8 PARMAX(10),PARMIN(10),RMAX(10),RMIN(10),C(20)

COMMON / CHROMOSP / CP
COMMON / MAXMIN / PARMAX,PARMIN

DO I=1,NC
    RMIN(I)=CP(1,I)
    RMAX(I)=CP(1,I)
ENDDO

DO I=2,N
    DO J=1,NC
        IF(CP(I,J) .LT. RMIN(J)) THEN
            RMIN(J)=CP(I,J)
        ENDIF
        IF(CP(I,J) .GT. RMAX(J)) THEN
            RMAX(J)=CP(I,J)
        ENDIF
    ENDDO
ENDDO

DO I=1,NC
    IF(RMIN(I) .GT. PARMIN(I)) THEN
        PARMIN(I)=RMIN(I)
    ENDIF
    IF(RMAX(I) .LT. PARMAX(I)) THEN
        PARMAX(I)=RMAX(I)
    ENDIF
ENDDO

C(1) = PARMIN(1)
C(2) = PARMIN(2)
C(3) = PARMIN(3)
C(4) = PARMIN(4)
C(5) = PARMIN(5)
C(6) = PARMIN(6)
C(7) = PARMIN(7)
C(8) = PARMIN(8)
C(9) = PARMIN(9)
C(10) = PARMIN(10)
C(11) = PARMAX(1)
C(12) = PARMAX(2)
C(13) = PARMAX(3)
C(14) = PARMAX(4)
C(15) = PARMAX(5)
C(16) = PARMAX(6)
C(17) = PARMAX(7)
C(18) = PARMAX(8)
C(19) = PARMAX(9)
C(20) = PARMAX(10)

IEA = 2
INP = 1
ITYPE = -2
LB = 20
NJ = 20
NI = 1
NWDS = 20
NOUTST = INP

CALL DAL(1,1,C(NOUTST),KORE,IEA,KADRNUM,IERR,NWDS,NJ,LB,
+      ITYPE,4HNEWW,4HPARA,1,1)

```



```

RETURN
END

C-----C
C SUBROUTINE RMTWIN                                     C
C-----C

      SUBROUTINE RMTWIN(OF2)

      INTEGER NS,NSF,NC

      REAL*8  CP2(100,10),OF2(100),OFREF

      COMMON/EXPT/NS,NC
      COMMON/CHROMOSP2/CP2

      NSF=2*NS
      DO J=1,NSF-1
        OFREF=OF2(J)
        DO I=J+1,NSF
3          IF(OF2(I) .EQ. OFREF) THEN
            WRITE(*,*)
            WRITE(*,*) 'TWINS(I,CP2(1,1-NC),OF2(I)):'
            IF(I .EQ. NSF) THEN
              NSF=NSF-1
              IF(J .EQ. NSF)GO TO 4
            ELSE
              DO K=I,NSF-1
                OF2(K)=OF2(K+1)
                DO L=1,NC
                  CP2(K,L)=CP2(K+1,L)
                ENDDO
              ENDDO
              NSF=NSF-1
              GO TO 3
            ENDIF
          ENDIF
        ENDDO
      ENDDO

      WRITE(*,*) 'NUMBER OF TWINS;',2*NS-NSF

4     CONTINUE
      RETURN
      END

C-----C
C SUBROUTINE PIKSR2B                                     C
C-----C

      SUBROUTINE PIKSR2B(N,M,ARR,BRR)

      INTEGER I,J,K,N,M

      REAL*8  ARR(100),BRR(100,10),A,B(10),ARR2(100),BRR2(100,10)

      DO J=2,N
        A=ARR(J)
        DO K=1,M
          B(K)=BRR(J,K)
        ENDDO
        DO I=J-1,1,-1
          IF(ARR(I) .LE. A) GO TO 10
          ARR(I+1)=ARR(I)
          DO K=1,M
            BRR(I+1,K)=BRR(I,K)
          ENDDO
        ENDDO
      ENDDO

```

```

      I=0
10      ARR(I+1)=A
          DO K=1,M
              BRR(I+1,K)=B(K)
          ENDDO
      ENDDO

      RETURN
      END

C-----C
C SUBROUTINE RMARIN(SEED1,SEED2).                C
C-----C

      SUBROUTINE RMARIN(IJ,KL)

      IMPLICIT REAL*8 (A-H,O-Z)
          REAL*8 U(97),C,CD,CM

      INTEGER          I97,J97

      COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

      LOGICAL TEST

      TEST = .FALSE.

      IF(IJ .LT. 0 .OR. IJ .GT. 31328 .OR.
+      KL .LT. 0 .OR. KL .GT. 30081) THEN
          WRITE(*,*)'THE FIRST RANDOM NUMBER SEED MUST HAVE'
          WRITE(*,*)'A VALUE BETWEEN 0 AND 31328'
          WRITE(*,*)
          WRITE(*,*)'THE SECOND RANDOM NUMBER SEED MUST HAVE'
          WRITE(*,*)'A VALUE BETWEEN 0 AND 30081'
      ENDIF

      I = MOD(IJ/177, 177) +2
      J = MOD(IJ      , 177) +2
      K = MOD(KL/169, 178) +1
      L = MOD(KL      , 169)

      DO 30 II = 1,97
          S = 0.0
          T = 0.5
          DO 40 JJ = 1,24
              M = MOD(MOD(I*J, 179)*K, 179)
              I = J
              J = K
              K = M
              L = MOD(53*L+1, 169)
              IF(MOD(L*M, 64) .GE. 32) THEN
                  S = S+T
              ENDIF
              T = 0.5*T
40          CONTINUE
          U(II) = S
30      CONTINUE

      C  = 362436.0 / 16777216.0
      CD = 7654321.0 / 16777216.0
      CM = 16777213.0 / 16777216.0
      I97 = 97
      J97 = 33

      TEST = .TRUE.

      RETURN
      END

```

```

C-----C
C FUNCTION URAND()C
C-----C

      REAL*8 FUNCTION URAND()

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 U(97),C,CD,CM

      INTEGER I97,J97

      COMMON / RASET1 / U,C,CD,CM,I97,J97,TEST

      LOGICAL TEST

      IF(.NOT.TEST) THEN
        WRITE(*,*)'URAND ERROR#1: MUST CALL THE INITIALIZATION ROUTINE'
        WRITE(*,*)'RMARIN BEFORE CALLING URAND.'
      ENDIF
      UNI = U(I97)-U(J97)
      IF(UNI .LT. 0.0) THEN
        UNI = UNI + 1.0
      ENDIF
      U(I97) = UNI
      I97 = I97 -1
      IF(I97 .EQ. 0) THEN
        I97 = 97
      ENDIF
      J97 = J97 - 1
      IF(J97 .EQ. 0) THEN
        J97 = 97
      ENDIF
      C = C - CD
      IF(C .LT. 0.0) THEN
        C = C + CM
      ENDIF
      UNI = UNI - C
      IF(UNI .LT. 0.0) THEN
        UNI = UNI + 1.0
      ENDIF

      URAND = UNI

      RETURN
      END

C-----C
C SUBROUTINE DESCRIBE(NS,OF)C
C-----C

      SUBROUTINE DESCRIBE(N,OF)

      INTEGER N

      REAL*8 OF(50),OFAVG

      OFAVG = 0.D0

      DO I=1,N
        OFAVG = OFAVG + OF(I)
      ENDDO

      OFAVG = OFAVG/N
      WRITE(*,*)'OFAVG=',OFAVG

      RETURN

```

END

C.2.5 The GENE PROC Processor

This processor ranks the chromosomes of the new generation and prepares the iteration generation (ITER ATIO population) to be used by processor GENC PROC. The RESET parameters available are listed below:

NUIN: Determines the input library. The processor will go to library NUIN to look for any datasets or tables it needs to complete its task. The default value for this RESET is 1.

NOUT: Determines the output library. Any tables or datasets that are created by the processor will be written to the NOUT library. The default value for this RESET is 1.

ITRG: Determines the iteration number the processor is working on. The default value for this RESET is 1.

ITPP: Determines which parent population the processor is working on. The default value for this RESET is 1.

Fig. C.6 illustrates the steps involved in the last processor that needs to be executed. This processor prepares the datasets for the iteration process.

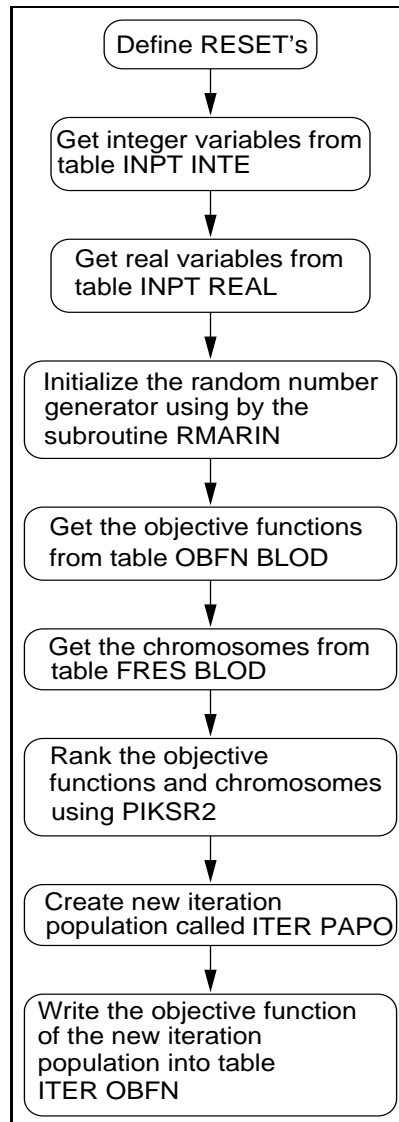


Figure C.6 The GENE PROC.

Fortran Code for the GENE PROC Processor

```

C-----C
C This processor was written by Alexander Hanuska on 10/19/1997 based on the C
C genetic algorithm developed by Sandrine Garcia on 03/16/1996 and is C
C protected under the GPL (GNU General Public License). A copy of GPL can C
C found at the following address: www.gnu.ai.mit.edu. It was designed to C
C be compiled and linked with EAL (Engineering Analysis Language) on a SGI C
C IP22 chipset. C
C-----C

      SUBROUTINE GENETICS5

      PARAMETER (IRSETNUM=4)

      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 A(1)
      REAL*8 PARMIN,PARMAX,PBREAL,PM
      REAL*8 OF,COPT,OFOPT,OF2,CP2
      REAL*8 P,CHILD

      INTEGER IA(1)
      INTEGER IL,MASK
      INTEGER NC,NPOP,CFSEED1,CFSEED2,SLOPSEED
      INTEGER NBEST1,NBEST2,NBEST3,NG
      INTEGER SEED1,SEED2,NS,PARENT(10)

      COMMON / CLFLPT /LFLPT
      COMMON / CMINT /LBLCFMI
      COMMON / CMFLPT /LBLCFMF
      COMMON / RANDOM /SEED1,SEED2
      COMMON / EXPT /NS,NC
      COMMON / CHROMOSP /CP
      COMMON / CHROMOSP2 /CP2
      COMMON / CSTGA /CFSEED1,CFSEED2,SLOPSEED,NG,NPOP,
+       NBEST1,NBEST2,NBEST3,PM
      COMMON / MAXMIN /PARMAX,PARMIN
      COMMON A

      DIMENSION IRESET(IRSETNUM),RRESET(IRSETNUM)
      DIMENSION IL(3,IRSETNUM),RL(IRSETNUM)
      DIMENSION PARMIN(10),PARMAX(10),CP(50,10)
      DIMENSION OF(50),COPT(50,10),OFOPT(50),OF2(100)
      DIMENSION P(50),CP2(100,10),PARENT(10),CHILD(10)

      EQUIVALENCE (A(1),IA(1))

      DATA MASK / 4HMASK /
      DATA IL / 4HNUIN,0,1,
+       4HNOUT,0,1,
+       4HITPP,0,1,
+       4HITRG,0,0 /

      CALL OPN1(4HGENE)

      IIRSETNUM = IRSETNUM
      CALL XRESET(IRESET,RRESET,IL,RL,IIRSETNUM)
      NUIN = IRESET(1)
      NOUT = IRESET(2)
      ITERPOP = IRESET(3)
      ITERG = IRESET(4)

C The following call statement looks for table INPT INTE in
C library 1, and reads in the values for some of the variables
C used in this genetic algorithm.

      IEA = 2
      INP = 1

```

```

      CALL DAL(1,11,IA(INP),KORE,IEA,KADNRNUM,IERR,NWDS,NJ,LB,ITYPE,
+         4HINPT,4HINTE,MASK,MASK)

      NI      = LB / NJ
      NBLK    = NWDS/ LB
      NOUTST  = INP+NWDS

      DO 10 J=1,NJ
          IPT      = INP+(J-1)*NI
          IOUTPT   = NOUTST+(J-1)*NI
          IA(IOUTPT) = IA(IPT)
10      CONTINUE

      NC      = IA(1)
      NPOP    = IA(2)
      CFSEED1  = IA(3)
      CFSEED2  = IA(4)
      SLOPSEED = IA(5)
      NBEST1   = IA(6)
      NBEST2   = IA(7)
      NBEST3   = IA(8)
      NG       = IA(9)
      SEED1    = CFSEED1-SLOPSEED*(ITERPOP-1)
      SEED2    = CFSEED2-SLOPSEED*(ITERPOP-1)
      NS       = NBEST1*NPOP

      IEA = 2
      INP = 1

C      The following call statement looks for table INPT REAL in
C      library 1, and reads in the values for some of the variables
C      used in this genetic algorithm

      CALL DAL(1,11,A(INP),KORE,IEA,KADNRNUM,IERR,NWDS,NJ,LB,
+         ITYPE,4HINPT,4HREAL,MASK,MASK)

      NI      = LB / NJ
      NBLK    = NWDS/ LB
      NOUTST  = INP+NWDS

      DO 20 J=1,NJ
          IPT      = INP+(J-1)*NI
          IOUTPT   = NOUTST+(J-1)*NI
          A(IOUTPT) = A(IPT)
20      CONTINUE

      PARMIN(1) = A(1)
      PARMIN(2) = A(2)
      PARMIN(3) = A(3)
      PARMIN(4) = A(4)
      PARMIN(5) = A(5)
      PARMIN(6) = A(6)
      PARMIN(7) = A(7)
      PARMIN(8) = A(8)
      PARMIN(9) = A(9)
      PARMIN(10) = A(10)
      PARMAX(1) = A(11)
      PARMAX(2) = A(12)
      PARMAX(3) = A(13)
      PARMAX(4) = A(14)
      PARMAX(5) = A(15)
      PARMAX(6) = A(16)
      PARMAX(7) = A(17)
      PARMAX(8) = A(18)
      PARMAX(9) = A(19)
      PARMAX(10) = A(20)
      PM        = A(21)

```

```

CALL RMARIN(SEED1,SEED2)

IF(ITERG .EQ. NG) THEN
  WRITE(*,*)
  WRITE(*,*)'----- THE END -----',
  WRITE(*,*)
  GOTO 3
ENDIF

IEA = 2
INP = 1

C      The next call statement looks for the table OBFN BLOOD in
C      library 1, and reads the values of the objective functions
C      into memory.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUN,IERR,NWDS,NJ,LB,
+       ITYPE,4HOBFN,4HBLOD,1,1)

DO I=1,NWDS,1
  OF(I)=A(I)
ENDDO

IEA = 2
INP = 1

C      The next call statement looks for the table OBFN BLOOD in
C      library 1, and reads the values of the chromosomes into memory.

CALL DAL(1,11,A(INP),KORE,IEA,KADRUN,IERR,NWDS,NJ,LB,
+       ITYPE,4HFRES,4HBLOD,1,1)

K=1
DO I=1,NJ,1
  DO J=1,NC
    CP(I,J)=A(K)
    K=K+1
  ENDDO
ENDDO

CALL PIKSR2(NS,NC,OF,CP)

WRITE(*,*)
WRITE(*,*)'THE NEW PARENT POPULATION:'

CALL DESCRIBE(NS,OF)

K = 0
DO 50 I = 1,NS
  DO 60 J = 1,NC
    A(J+K) = CP(I,J)
60    CONTINUE
    K = K + NC
50  CONTINUE

IEA = 2
INP = 1
ITYPE = -2
LB = NC*NS
NJ = NS
NI = NC
NWDS = LB
NBLK = 1
NOUTST = INP + LB
COUNT = NOUTST

DO 120 I = 1,NJ

```



```

DO 125 J = 1,NC
      A(COUNT+(J-1))=A(J+NC*(I-1))
      COUNT1 = COUNT+(J-1)
125      CONTINUE
      COUNT = COUNT1+1
120      CONTINUE

C      The following call statement creates a new table ITER PAPO
C      in library 1, and writes the chromosomes of the iteration
C      generation in it.

      CALL DAL(1,1,A(NOUTST),KORE,IEA,KADNRNUM,IERR,NWDS,NJ,LB,
+           ITYPE,4HITER,4HPAPO,1,1)

      DO I = 1,NS
        A(I) = OF(I)
      ENDDO

      IEA      = 2
      INP      = 1
      ITYPE    = -2
      LB       = NS
      NJ       = NS
      NI       = 1
      NWDS     = LB
      NBLK     = 1
      NOUTST   = INP + LB

      DO 80 J = 1,NJ
        IPT      = INP+(J-1)*NI
        IOUTPT   = NOUTST+(J-1)*NI
        A(IOUTPT)= A(IPT)
      80 CONTINUE

C      The following call statement creates a new table ITER OBFN
C      in library 1, and writes the objective functions of the iteration
C      generation in it.

      CALL DAL(1,1,A(NOUTST),KORE,IEA,KADNRNUM,IERR,NWDS,NJ,LB,
+           ITYPE,4HITER,4HOBFN,1,1)

3      CALL FIN(0,0)
      RETURN
      END
      END

C-----C
C      SUBROUTINE PIKSR2.                                C
C-----C

      SUBROUTINE PIKSR2(N,M,ARR,BRR)

      REAL*8 ARR(50),BRR(50,10),A,B(10),ARR2(50),BRR2(50,10)

      INTEGER I,J,K,N,M

      DO J=2,N
        A=ARR(J)
        DO K=1,M
          B(K)=BRR(J,K)
        ENDDO
        DO I=J-1,1,-1
          IF(ARR(I) .LE. A) GOTO 90
          ARR(I+1)=ARR(I)
          DO K=1,M
            BRR(I+1,K)=BRR(I,K)
          ENDDO
        ENDDO
      ENDDO

```

```

          I=0
90        ARR(I+1)=A
          DO K=1,M
             BRR(I+1,K)=B(K)
          ENDDO
        ENDDO

        RETURN
      END

C-----C
C      SUBROUTINE DESCRIBE(NS,OF)      C
C-----C

      SUBROUTINE DESCRIBE(N,OF)

      INTEGER N

      REAL*8  OF(50),OFAVG

      OFAVG = 0.DO

      DO I=1,N
         OFAVG = OFAVG + OF(I)
      ENDDO

      OFAVG = OFAVG/N
      WRITE(*,*)'THE AVERAGE OF THE OBJECTIVE FUNCTION IS:'
      WRITE(*,*)'OFAVG=',OFAVG
      RETURN
      END

```

Appendix D

The Steady-State Analysis Program Composite.f

```
PROGRAM COMPOSITE
*****
* This program was written by Alexander Hanuska on 8/25/1996.      *
* This program is used to determine the thermal influence of each  *
* material utilized to produce the outer wing subcomponent of the HSCT. *
* The variables are defined at the beginning of the subroutine they *
* are used in.                                                    *
* In order to use this program, one has to visually cut an area of the *
* wing subcomponent and define the thermal resistances involved in the *
* cross-section of the cut.                                         *
*****
*
      IMPLICIT NONE
      DOUBLE PRECISION QCOND1,DELTA,A,DELTAT,LAMBDA,EPS1,T,
+          SIGMA,D1,QCONV1,QRADI1,FUNC,FUNCT,
+          R11T,T1,R22T,T2,R22AT,T2A,R33,T3,K3,R44,
+          T4,K4,R44A,T4A,R55,T5,K5,R55A,T5A,R66,T6,K6,
+          R66A,T6A,R77,T7,K7,R77A,T7A,R88,T8,K8,R88A,
+          T8A,T9,K9,KA,PRA,VISCA,DIFFA,PI,KSS,K10,
+          R1C,R2C,R3C,R4C,R5C,R6C,R7C,R8C,R9C,R10C,
+          R11C,R12C,R13C,R14C,R15C,R16C,H1C,H2C,H3C,
+          H4C,H5C,H6C,H7C,H8C,H9C,H10C,H11C,H12C,
+          H13C,H14C,H15C,H16C,R1VB,
+          R1VT,R2V,R3V,R4V,R5V,R6V,H1V,H2V,H3V,H4V,H5V,
+          H6V,RAL,GRAV,BETA,RRAD6,RAIR1B,
+          RHCT,R99,RRAD1,RRAD2,RRAD3T,RRAD3B,RRAD4,RRAD5,
+          EPS2,EPS3,EPS4,EPS5,EPS61,RT,Y1T,Y2,Y3,Y4,Y5,
+          Y6,RAIR1T,RAIR2,RAIR3,RAIR4,RAIR5,RAIR6,
+          DONE(11,59),WHC,RHCB,R11B,R22B,R22AB,Y1B
*
      INTEGER FLAG,N,I,FLAGSS,FLAGCLOST,FLAGKA,FLAGK1,FLAGK2,FLAGDIFFA,
+          FLAGPRA,FLAGVISCA,U,V
*
      COMMON/VARIABLE/T,DELTAT
*
      DELTAT = 5.DO
```

```

*
    U = 0
*
    DO T = 295.D0,445.D0,15
*
    U=U+1
*
    DONE(U,1) = T
*
    CALL HONEYT(RHCT)
*
    DONE(U,51)=RHCT
*
    CALL HONEYB(RHCB)
*
    DONE(U,53)=RHCB
*
    CALL CONDUCT(R11T ,R11B ,R22T,R22B,R33 ,R44 ,R55 ,R66,R77,R88,
+          R22AT,R22AB,R44A,R55A,R66A,R77A,R88A,R99)
*
    DONE(U,2) =R11T
    DONE(U,3) =R22T
    DONE(U,4) =R22AT
    DONE(U,5) =R33
    DONE(U,6) =R44
    DONE(U,7) =R44A
    DONE(U,8) =R55
    DONE(U,9) =R55A
    DONE(U,10)=R66
    DONE(U,11)=R66A
    DONE(U,12)=R77
    DONE(U,13)=R77A
    DONE(U,14)=R88
    DONE(U,15)=R88A
    DONE(U,16)=R99
    DONE(U,54)=R11B
    DONE(U,55)=R22B
    DONE(U,56)=R22AB
*
    CALL CONTACT(R1C ,R2C ,R3C ,R4C ,R5C ,R6C ,R7C ,R8C ,R9C ,R10C,
+          R11C,R12C,R13C,R14C,R15C,R16C)
*
    DONE(U,17)=R1C
    DONE(U,18)=R2C
    DONE(U,19)=R3C
    DONE(U,20)=R4C
    DONE(U,21)=R5C
    DONE(U,22)=R6C
    DONE(U,23)=R7C
    DONE(U,24)=R8C
    DONE(U,25)=R9C
    DONE(U,26)=R10C
    DONE(U,27)=R11C
    DONE(U,28)=R12C
    DONE(U,29)=R13C
    DONE(U,30)=R14C
    DONE(U,31)=R15C
    DONE(U,32)=R16C
*

```

```

      CALL CONVECT(R1VT ,R1VB ,R2V ,R3V ,R4V ,R5V ,R6V,RAIR1T,
+             RAIR1B,RAIR2,RAIR3,RAIR4,RAIR5,RAIR6)
*
      DONE(U,33)=R1VT
      DONE(U,34)=R2V
      DONE(U,35)=R3V
      DONE(U,36)=R4V
      DONE(U,37)=R5V
      DONE(U,38)=R6V
      DONE(U,39)=RAIR1T
      DONE(U,40)=RAIR2
      DONE(U,41)=RAIR3
      DONE(U,42)=RAIR4
      DONE(U,43)=RAIR5
      DONE(U,44)=RAIR6
      DONE(U,57)=R1VB
      DONE(U,58)=RAIR1B
*
      CALL RADIAT(RRAD1,RRAD2,RRAD3T,RRAD3B,RRAD4,RRAD5,RRAD6)
*
      DONE(U,45)=RRAD1
      DONE(U,46)=RRAD2
      DONE(U,47)=RRAD3T
      DONE(U,59)=RRAD3B
      DONE(U,48)=RRAD4
      DONE(U,49)=RRAD5
      DONE(U,50)=RRAD6
*
      CALL HCCLOSET(Y1T,R22T,R22AT,RAIR1T,RRAD3T,R1VT)
*
      CALL HCCLOSEB(Y1B,R22B,R22AB,RAIR1B,RRAD3B,R1VB)
*
      CALL TUBE1(Y2,R66A,R66,RRAD4,RAIR4,R2V)
*
      CALL TUBE2(Y3,R77A,R77,RRAD5,RAIR5,R3V)
*
      CALL TUBE3(Y4,R88A,R88,RRAD6,RAIR6,R4V)
*
      CALL CHANNELL(Y5,R44,R44A,RAIR2,RRAD1,R5V)
*
      CALL CHANNELR(Y6,R55,R55A,RAIR3,RRAD2,R6V)
*
      RT = R11T+
+      (1.D0/R1C+1.D0/R2C+1.D0/R2C)**(-1.D0)+
+      (1.D0/RHCT+1.D0/Y1T+1.D0/Y1T)**(-1.D0)+
+      (1.D0/R3C+1.D0/R4C+1.D0/R4C)**(-1.D0)+
+      R33+
+      (1.D0/R5C+1.D0/R6C+1.D0/R7C+1.D0/R8C+1.D0/R9C)**(-1.D0)+
+      (1.D0/Y2+1.D0/Y3+1.D0/Y4+1.D0/Y5+1.D0/Y6)**(-1.D0)+
+      (1.D0/R10C+1.D0/R11C+1.D0/R12C+1.D0/R13C+1.D0/R14C)**
+      (-1.D0)+R99+
+      (1.D0/R3C+1.D0/R4C+1.D0/R4C)**(-1.D0)+
+      (1.D0/RHCB+1.D0/Y1B+1.D0/Y1B)**(-1.D0)+
+      (1.D0/R1C+1.D0/R2C+1.D0/R2C)**(-1.D0)+
+      R11B
*
      DONE(U,52)=RT
*
      ENDDO

```

```

*
      OPEN(40,FILE="composite.out")
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)'TEMP      RHCT      R11T      R22T      R22AT      R33      R44
+      R44A'
      WRITE(40,*)'(K)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)
+      (K/W)'
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)' '
      DO V = 1,11,1
      WRITE(40,30)DONE(V,1),DONE(V,51),DONE(V,2),DONE(V,3),DONE(V,4),
+      DONE(V,5),DONE(V,6),DONE(V,7)
30  FORMAT(F5.1,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,1X,F7.4,
+      1X,F7.4)
      ENDDO
*
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)'TEMP      R55      R55A      R66      R66A      R77      R77A
+      R88'
      WRITE(40,*)'(K)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)
+      (K/W)'
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)' '
      DO V = 1,11,1
      WRITE(40,30)DONE(V,1),DONE(V,8),DONE(V,9),DONE(V,10),DONE(V,11)
+      ,DONE(V,12),DONE(V,13),DONE(V,14)
      ENDDO
*
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)'TEMP      R88A      R99      R1C      R2C      R3C      R4C
+      R5C'
      WRITE(40,*)'(K)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)
+      (K/W)'
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)' '
      DO V = 1,11,1
      WRITE(40,30)DONE(V,1),DONE(V,15),DONE(V,16),DONE(V,17),
+      DONE(V,18),DONE(V,19),DONE(V,20),DONE(V,21)
      ENDDO
*
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)'TEMP      R6C      R7C      R8C      R9C      R10C      R11C
+      R12C'
      WRITE(40,*)'(K)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)      (K/W)
+      (K/W)'
      WRITE(40,*)'-----'
+-----'
      WRITE(40,*)' '
      DO V = 1,11,1
      WRITE(40,30)DONE(V,1),DONE(V,22),DONE(V,23),DONE(V,24),
+      DONE(V,25),DONE(V,26),DONE(V,27),DONE(V,28)
      ENDDO

```

```

*
WRITE(40,*)'-----'
+-----'
WRITE(40,*)'TEMP   R13C   R14C   R15C   R16C   R1VT   R2V
+ R3V'
WRITE(40,*)'(K)    (K/W)  (K/W)  (K/W)   (K/W)  (K/W)   (K/W)
+ (K/W)'
WRITE(40,*)'-----'
+-----'
WRITE(40,*)' '
DO V = 1,11,1
WRITE(40,30)DONE(V,1),DONE(V,29),DONE(V,30),DONE(V,31),
+ DONE(V,32),DONE(V,33),DONE(V,34),DONE(V,35)
ENDDO

*
WRITE(40,*)'-----'
+-----'
WRITE(40,*)'TEMP   R4V    R5V    R6V    RAIR1T  RAIR2   RAIR3
+ RAIR4'
WRITE(40,*)'(K)    (K/W)  (K/W)  (K/W)   (K/W)  (K/W)   (K/W)
+ (K/W)'
WRITE(40,*)'-----'
+-----'
WRITE(40,*)' '
DO V = 1,11,1
WRITE(40,30)DONE(V,1),DONE(V,36),DONE(V,37),DONE(V,38),
+ DONE(V,39),DONE(V,40),DONE(V,41),DONE(V,42)
ENDDO

*
WRITE(40,*)'-----'
+-----'
WRITE(40,*)'TEMP   RAIR5  RAIR6  RRAD1   RRAD2   RRAD3T  RRAD4
+ RRAD5'
WRITE(40,*)'(K)    (K/W)  (K/W)  (K/W)   (K/W)  (K/W)   (K/W)
+ (K/W)'
WRITE(40,*)'-----'
+-----'
WRITE(40,*)' '
DO V = 1,11,1
WRITE(40,30)DONE(V,1),DONE(V,43),DONE(V,44),DONE(V,45),
+ DONE(V,46),DONE(V,47),DONE(V,48),DONE(V,49)
ENDDO

*
WRITE(40,*)'-----'
+-----'
WRITE(40,*)'TEMP   RRAD6   RT    RHCB   R11B   R22B   R22AB
+ R1VB'
WRITE(40,*)'(K)    (K/W)  (K/W)  (K/W)   (K/W)  (K/W)   (K/W)
+ (K/W)'
WRITE(40,*)'-----'
+-----'
WRITE(40,*)' '
DO V = 1,11,1
WRITE(40,30)DONE(V,1),DONE(V,50),DONE(V,52),DONE(V,53),
+ DONE(V,54),DONE(V,55),DONE(V,56),DONE(V,57)
ENDDO

*
WRITE(40,*)'-----'
+-----'

```

```

        WRITE(40,*)'TEMP      RAIR1B      RRAD3B'
        WRITE(40,*)'(K)        (K/W)        (K/W) '
        WRITE(40,*)'-----'
+-----'
        WRITE(40,*)' '
        DO V = 1,11,1
            WRITE(40,30)DONE(V,1),DONE(V,58),DONE(V,59)
*
        ENDDO
*
        STOP
*
        END
*
***** Subroutine Tube1 *****
*
* This subroutine calculates the equivalent thermal resistance for the
* torque tube 1.
*
*****
*
        SUBROUTINE TUBE1(Y2,R66A,R66,RRAD4,RAIR4,R2V)
*
        IMPLICIT NONE
        DOUBLE PRECISION Y2,R66A,R66,RRAD4,RAIR4,R2V
*
        Y2 = (1.DO/R66A+1.DO/R66A+1.DO/((1.DO/RRAD4+1.DO/RAIR4+1.DO/
+          R2V)**(-1.DO)+R66+R66))**(-1.DO)
*
        RETURN
        END
*
***** Subroutine Tube2 *****
*
* This subroutine calculates the equivalent thermal resistance for the
* torque tube 2.
*
*****
*
        SUBROUTINE TUBE2(Y3,R77A,R77,RRAD5,RAIR5,R3V)
*
        IMPLICIT NONE
        DOUBLE PRECISION Y3,R77A,R77,RRAD5,RAIR5,R3V
*
        Y3 = (1.DO/R77A+1.DO/R77A+1.DO/((1.DO/RRAD5+1.DO/RAIR5+1.DO/
+          R3V)**(-1.DO)+R77+R77))**(-1.DO)
*
        RETURN
        END
*
***** Subroutine Tube3 *****
*
* This subroutine calculates the equivalent thermal resistance for the
* torque tube 3.
*
*****
*
        SUBROUTINE TUBE3(Y4,R88A,R88,RRAD6,RAIR6,R4V)
*

```



```

        IMPLICIT NONE
        DOUBLE PRECISION Y4,R88A,R88,RRAD6,RAIR6,R4V
*
        Y4 = (1.DO/R88A+1.DO/R88A+1.DO/((1.DO/RRAD6+1.DO/RAIR6+1.DO/
+           R4V)**(-1.DO)+R88+R88))**(-1.DO)
*
        RETURN
        END
*
***** Subroutine Channell *****
*
* This subroutine calculates the equivalent thermal resistance for the *
* left edge channel. *
*
*****
*
        SUBROUTINE CHANNELL(Y5,R44,R44A,RAIR2,RRAD1,R5V)
*
        IMPLICIT NONE
        DOUBLE PRECISION Y5,R44,R44A,RAIR2,RRAD1,R5V
*
        Y5 = (1.DO/(R44+(1.DO/RAIR2+1.DO/RRAD1+1.DO/R5V)**(-1.DO)+R44)+
+           1.DO/R44A)**(-1.DO)
*
        RETURN
        END
*
***** Subroutine Channelr *****
*
* This subroutine calculates the equivalent thermal resistance for the *
* right edge channel. *
*
*****
*
        SUBROUTINE CHANNELR(Y6,R55,R55A,RAIR3,RRAD2,R6V)
*
        IMPLICIT NONE
        DOUBLE PRECISION Y6,R55,R55A,RAIR3,RRAD2,R6V
*
        Y6 = (1.DO/(R55+(1.DO/RAIR3+1.DO/RRAD2+1.DO/R6V)**(-1.DO)+R55)+
+           1.DO/R55A)**(-1.DO)
*
        RETURN
        END
*
***** Subroutine HC closeoutst *****
*
* this subroutine calculates the equivalent thermal resistance for the *
* honeycomb closeouts closest to the heated surface. *
*
*****
*
        SUBROUTINE HCCLOSET(Y1T,R22T,R22AT,RAIR1T,RRAD3T,R1VT)
*
        IMPLICIT NONE
        DOUBLE PRECISION Y1T,R22T,R22AT,RAIR1T,RRAD3T,R1VT
*
        Y1T = (1.DO/(R22T+(1.DO/RAIR1T+1.DO/RRAD3T+1.DO/R1VT)**(-1.DO)
+           +R22T)+1.DO/R22AT)**(-1.DO)

```

```

*
      RETURN
      END
*
***** Subroutine HC closeoutsb *****
*
* this subroutine calculates the equivalent thermal resistance for the
* honeycomb closeouts furthest from the heated surface.
*
*****
*
      SUBROUTINE HCCLOSEB(Y1B,R22B,R22AB,RAIR1B,RRAD3B,R1VB)
*
      IMPLICIT NONE
      DOUBLE PRECISION Y1B,R22B,R22AB,RAIR1B,RRAD3B,R1VB
*
      Y1B = (1.DO/(R22B+(1.DO/RAIR1B+1.DO/RRAD3B+1.DO/R1VB)**(-1.DO)
+          +R22B)+1.DO/R22AB)**(-1.DO)
*
      RETURN
      END
*
***** Subroutine Honeyt *****
*
* This subroutine was written for the honeycomb structure closest to
* the heated surface. The properties are calculated at a higher
* average temperature.
*
* This subroutine calculates the effective thermal conductivity of any
* honeycomb structure, and the equations were developed by C. W. Stroud
* at the Langley Research Center, Langley Station in Hampton VA. They
* are summarized in a booklet called:
* Experimental Verification of an Analytical Determination of
* Overall Thermal Conductivity of Honeycomb-Core Panels
*
* RHCT : Effective thermal conductivity of the top honeycomb structure
* QCOND1: Heat transfer through the honeycomb through conduction
* KSS : Thermal conductivity of honeycomb material (SS321)
* LHC : Height of the honeycomb structure
* DELTAA: Density of the honeycomb
* A : Density of the honeycomb material
* DELTAT: Maximum temperature difference between the hot and the cold
* surfaces
* QCONV1: Heat transfer through the honeycomb through convection
* KA : Thermal conductivity of gas in honeycomb cell (air)
* LAMBDA: L1/D1, with D1 being the equivalent diameter
* D1 : equivalent diameter of the honeycomb cell
* EPS1 : emissivity of the facesheets of the honeycomb, often chosen
* to be equivalent to the emissivity of the core material of
* honeycomb
* T : Temperature of the top facesheet of the honeycomb structure F
* SIGMA : Stefan-Boltzman constant (W/m2.K4)
* QRADI1: Heat transfer through the honeycomb through radiation
* D : Depth
* WHC : Length over which the total heat flux is applied. The unit
* depth is multiplied by WHC to form the conduction area.
*
* Note: All the thicknesses are in (m)
* All the temperatures are in (K)

```

```

*      All the heat transfers are in (W)                                     *
*      All the thermal conductivities are in (W/m.K)                       *
*                                                                           *
*****
*
      SUBROUTINE HONEYT(RHCT)
*
      IMPLICIT NONE
      DOUBLE PRECISION QCOND1,KSS,DELTA,A,DELTAT,LAMBDA,EPS1,
+          SIGMA,D1,KA,QCONV1,QRADI1,T,LHC,D,PI,WHC,
+          RHCT,KHC
      INTEGER FLAGSS,FLAGKAT
*
      COMMON/VARIABLE/T,DELTAT
*
      T      = T - DELTAT/2.DO
*
      CALL FSS321T(T,KSS,FLAGSS)
      CALL FKAT(T,KA,FLAGKAT)
*
      T      = T + DELTAT/2.DO
*
      PI      = DACOS(-1.DO)
      LHC      = ((0.8610D0+0.8350D0)/2.DO)/39.37D0
      DELTAA = PI*((0.09575D0/39.27D0)**2.DO)
+          -PI*((0.09375D0/39.37D0)**2.DO)
      A      = PI*((0.09575D0/39.27D0)**2.DO)
      D1      = (3.DO/16.DO)/39.37D0
      LAMBDA = LHC/D1
      EPS1    = 0.7D0
      SIGMA   = 5.67D-08
      D       = 1.DO
      WHC     = (24.DO-2.533D0)/39.37D0
*
* Determination of heat transfer by conduction
*
      QCOND1 = (KSS/LHC)*(DELTA/A)*(DELTAT)
*
* Determination of heat transfer by convection
*
      QCONV1 = (KA/LHC)*(1.DO-(DELTA/A))*(DELTAT)
*
* Determination of heat transfer by radiation
*
      QRADI1 = 0.664D0*(LAMBDA+0.3D0)**(-0.69D0)*EPS1**((1.63D0*(
+          LAMBDA+1.DO)**(-.89D0))*SIGMA*((T+DELTAT)**4-(T**4))
*
* Determination of the equivalent thermal conductivity for the
* honeycomb structure
*
      KHC = ((QCOND1+QCONV1+QRADI1)*LHC*D*WHC)/(DELTAT*D*WHC)
*
      RHCT = (DELTAT)/((QCOND1+QCONV1+QRADI1)*D*WHC)
*
      RETURN
      END
*
***** Subroutine Honeyb *****
*

```

```

* This subroutine was written for the honeycomb structure closest to
* the heated surface. The properties are calculated at a higher
* average temperature.
*
* This subroutine calculates the effective thermal conductivity of any
* honeycomb structure, and the equations were developed by C. W. Stroud
* at the Langley Research Center, Langley Station in Hampton VA. They
* are summarized in a booklet called:
* Experimental Verification of an Analytical Determination of
* Overall Thermal Conductivity of Honeycomb-Core Panels
*
* RHCB : Effective thermal conductivity of the top honeycomb structure
* QCOND1: Heat transfer through the honeycomb through conduction
* KSS   : Thermal conductivity of honeycomb material (SS321)
* LHC   : Height of the honeycomb structure
* DELTAA: Density of the honeycomb
* A     : Density of the honeycomb material
* DELTAT: Maximum temperature difference between the hot and the cold
*         surfaces
* QCONV1: Heat transfer through the honeycomb through convection
* KA     : Thermal conductivity of gas in honeycomb cell (air)
* LAMBDA: L1/D1, with D1 being the equivalent diameter
* D1     : equivalent diameter of the honeycomb cell
* EPS1   : emissivity of the facesheets of the honeycomb, often chosen
*         to be equivalent to the emissivity of the core material of
*         honeycomb
* T      : Temperature of the top facesheet of the honeycomb structure F
* SIGMA  : Stefan-Boltzman constant (W/m2.K4)
* QRADI1: Heat transfer through the honeycomb through radiation
* D      : Depth
* WHC    : Length over which the total heat flux is applied. The unit
*         depth is multiplied by WHC to form the conduction area.
*
* Note: All the thicknesses are in (m)
*       All the temperatures are in (K)
*       All the heat transfers are in (W)
*       All the thermal conductivities are in (W/m.K)
*
*****
*
* SUBROUTINE HONEYB(RHCB)
*
* IMPLICIT NONE
* DOUBLE PRECISION QCOND1,KSS,DELTA,A,DELTAT,LAMBDA,EPS1,
* +               SIGMA,D1,KA,QCONV1,QRADI1,T,LHC,D,PI,WHC,
* +               RHCB,KHC
* INTEGER FLAGSS,FLAGKAT
*
* COMMON/VARIABLE/T,DELTAT
*
* T      = T - (2.DO*DELTAT+DELTAT/2.DO)
*
* CALL FSS321T(T,KSS,FLAGSS)
* CALL FKAT(T,KA,FLAGKAT)
*
* T      = T + (2.DO*DELTAT+DELTAT/2.DO)
*
* PI     = DACOS(-1.DO)
* LHC    = ((0.8610D0+0.8350D0)/2.DO)/39.37D0

```

```

      DELTAA = PI*((0.09575D0/39.27D0)**2.D0)
+      -PI*((0.09375D0/39.37D0)**2.D0)
      A      = PI*((0.09575D0/39.27D0)**2.D0)
      D1     = (3.D0/16.D0)/39.37D0
      LAMBDA = LHC/D1
      EPS1   = 0.7D0
      SIGMA  = 5.67D-08
      D      = 1.D0
      WHC    = (24.D0-2.533D0)/39.37D0
*
* Determination of heat transfer by conduction
*
      QCOND1 = (KSS/LHC)*(DELTAA/A)*(DELTAT)
*
* Determination of heat transfer by convection
*
      QCONV1 = (KA/LHC)*(1.D0-(DELTAA/A))*(DELTAT)
*
* Determination of heat transfer by radiation
*
      QRADI1 = 0.664D0*(LAMBDA+0.3D0)**(-0.69D0)*EPS1**(1.63D0*(
+      LAMBDA+1.D0)**(-.89D0))*SIGMA*((T+DELTAT)**4-(T**4))
*
* Determination of the equivalent thermal conductivity for the
* honeycomb structure
*
      KHC = ((QCOND1+QCONV1+QRADI1)*LHC*D*WHC)/(DELTAT*D*WHC)
*
      RHCB = (DELTAT)/((QCOND1+QCONV1+QRADI1)*D*WHC)
*
      RETURN
      END
*
***** Subroutine Interpolation *****
*
* This subroutine will interpolate between temperatures for the
* different material properties, it will extrapolate when the tempera-
* ture falls outside the range present in the array (below or above)
*
* N      : The number of lines forming the array for the material
*          properties
* I      : Line number of the array
* FLAG   : Counter
* T      : Temperature for which the material properties are needed
* FUNC   : Return of the interpolated value of the property of the
*          array
* FUNCT  : Defines the values present in the array
*
* Note: All the temperatures are in (K)
*
*****
*
      SUBROUTINE INTERPOL(N,FUNCT,T,FUNC,FLAG)
*
      IMPLICIT NONE
      DOUBLE PRECISION FUNC,FUNCT(N,2),T
      INTEGER          FLAG,I,N
*
      FLAG = 0

```

```

DO I = 1, N-1
  IF (T .GT. FUNCT(I,1) .AND. T .LE. FUNCT(I+1,1)) THEN
    FUNC = FUNCT(I,2)+(FUNCT(I+1,2)-FUNCT(I,2))*(T-FUNCT(I,1))/
+      (FUNCT(I+1,1)-FUNCT(I,1))
    FLAG = 0
    GOTO 2
  ENDIF
ENDDO
*
2  IF (T .LE. FUNCT(1,1)) THEN
  FUNC = FUNCT(1,2)+(FUNCT(2,2)-FUNCT(1,2))*(T-FUNCT(1,1))/
+    (FUNCT(2,1)-FUNCT(1,1))
  FLAG = 1
  GOTO 3
ENDIF
*
  IF (T .GT. FUNCT(N,1)) THEN
    FUNC = FUNCT(N-1,2)+(FUNCT(N,2)-FUNCT(N-1,2))*(T-FUNCT(N-1,
+    1))/(FUNCT(N,1)-FUNCT(N-1,1))
    FLAG =1
    GOTO 3
  ENDIF
*
3  RETURN
END
*
***** Subroutine Conduction *****
*
* This subroutine will calculate the thermal resistance
* for all the elements that are subjected to heat conduction. Since
* we are using a 1-D model, we will consider a unit conduction area.
* All the thermal conductivities have been calculated such that the
* value corresponds to a given temperature.
*
* R11T : Thermal resistance for the top outer facesheets
* R11B : Thermal resistance for the bottom outer facesheet
* T1 : Thickness of the outer facesheets
* K1 : Thermal conductivity for the material of the outer face-
* sheets
* L1 : Length for the outer facesheet
* R22A : Thermal resistance for the honeycomb edge closeouts (Top
* and Bottom) closest to the heated surface
* R22B : Thermal resistance for the honeycomb edge closeouts (Top
* and Bottom) furthest from the heated surface
* T2 : Thickness of the honeycomb edge closeouts (Top and Bottom)
* K2 : Thermal conductivity for the material of the honeycomb edge
* closeouts
* L2 : Length for the honeycomb edge closeouts (Top and Bottom)
* R22AT : Thermal resistance for the honeycomb edge closeouts (Side
* Supports) closest to the heated surface
* R22AB : Thermal resistance for the honeycomb edge closeouts (Side
* Supports) furthest from the heated surface
* L2A : Length for the honeycomb edge closeouts (side supports)
* T2A : Thickness of the honeycomb edge closeouts (Side Supports)
* R33 : Thermal resistance for the upper skin
* T3 : Thickness of the upper skin
* K3 : Thermal conductivity for the material of the upper skin
* L3 : Length for the upper skin
* R44 : Thermal resistance for the left edge channel (Top and

```

```

*      Bottom)
* T4      : Thickness of the left edge channel (Top and Bottom)
* K4      : Thermal conductivity for the material of the left edge
*          channel
* L4      : Length for the left edge channel (Top and Bottom)
* R44A    : Thermal resistance for the left edge channel (Side Supports)*
* T4A     : Thickness of the left edge channel (Side Supports)
* L4A     : Length for the left edge channel (Side Supports)
* R55     : Thermal resistance for the right edge channel
* T5      : Thickness of the right edge channel
* K5      : Thermal conductivity of the material of the right edge
*          channel
* L5      : Length for the right edge channel (Top and Bottom)
* R55A    : Thermal resistance for the right edge channel (Side
*          Supports)
* T5A     : Thickness of the right edge channel (Side Supports)
* L5A     : Length for the right edge channel (Side Supports)
* R66     : Thermal resistance for the torque tube #1 (Top and Bottom)
* T6      : Thickness of the torque tube #1 (Top and Bottom)
* K6      : Thermal conductivity of the material of the torque tube #1
* L6      : Length for the torque tube #1 (Top and Bottom)
* R66A    : Thermal resistance for the torque tube #1 (Side Supports)
* T6A     : Thickness of the torque tube #1 (Side Supports)
* L6A     : Length for the torque tube #1 (Side Supports)
* R77     : Thermal resistance for the torque tube #2 (Top and Bottom)
* T7      : Thickness of the torque tube #2 (Top and Bottom)
* K7      : Thermal conductivity of the material of the torque tube #2
* L7      : Length for the torque tube #2 (Top and Bottom)
* R77A    : Thermal resistance for the torque tube #2 (Side Supports)
* T7A     : Thickness of the torque tube #2 (Side Supports)
* L7A     : Length for the torque tube #2 (Side Supports)
* R88     : Thermal resistance for the torque tube #3 (Top and Bottom)
* T8      : Thickness of the torque tube #3 (Top and Bottom)
* K8      : Thermal conductivity of the material of the torque tube #3
* L8      : Length for the torque tube #3 (Top and Bottom)
* R88A    : Thermal resistance for the torque tube #3 (Side Supports)
* T8A     : Thickness of the torque tube #3 (Side Supports)
* L8A     : Length for the torque tube #3 (Side Supports)
* R99     : Thermal resistance for the lower skin
* T9      : Thickness of the lower skin
* K9      : Thermal conductivity of the material of the lower skin
* L9      : Length for the lower skin
* D       : Depth
*
* Note: All the thermal conductivity coefficients are in (W/m.K)
*       All the thicknesses are in (m)
*       All the thermal resistances are in (K/W)
*
*****
*
SUBROUTINE CONDUCT(R11T,R11B,R22T,R22B,R33,R44,R55,R66,R77,R88,
+                R22AT,R22AB,R44A,R55A,R66A,R77A,R88A,R99)
*
IMPLICIT NONE
DOUBLE PRECISION R11T,T1,K1,R22T,T2,K2,R22AT,T2A,R33,T3,K3,R44,
+                T4,K4,R44A,T4A,R55,T5,K5,R55A,T5A,R66,T6,K6,
+                R66A,T6A,R77,T7,K7,R77A,T7A,R88,T8,K8,R88A,
+                T8A,T9,K9,D,L1,L2,L2A,L3,L4,L4A,L5,L5A,L6,L6A
+                ,L7,L7A,L8,L8A,L9,T,R99,R11B,R22B,R22AB,

```

```

+          DELTAT
INTEGER FLAGK1,FLAGK2
*
COMMON/VARIABLE/T,DELTAT
*
D      = 1.D0
L1     = 24.000D0/39.37D0
T1     = 0.1680D0/39.37D0
T2     = 2.D0*0.0550D0/39.37D0
L2     = (1.2905D0-0.0665D0)/39.37D0
T2A    = ((2.D0*0.0550D0)+0.8480D0)/39.37D0
L2A    = 0.0665D0/39.37D0
T3     = 0.1610D0/39.37D0
L3     = 24.000D0/39.37D0
T4     = (2.D0*0.0860D0)/39.37D0
L4     = 1.1440D0/39.37D0
T4A    = ((2.D0*0.0860D0)+2.090D0)/39.37D0
L4A    = 0.0735D0/39.37D0
T5     = (2.D0*0.0860D0)/39.37D0
L5     = (1.3150D0-0.0960D0)/39.37D0
T5A    = ((2.D0*0.0860D0)+((0.4160D0+0.3050D0)/2.D0))/39.37D0
L5A    = 0.0960D0/39.37D0
T6     = 2.D0*0.0770D0/39.37D0
L6     = (4.8320D0-(.1590D0/2.D0)-(.1730D0/2.D0))/39.37D0
T6A    = (((2.0130D0+1.5830D0)/2.D0)+(0.0770D0))/39.37D0
L6A    = ((0.1590D0/2.D0)+(0.1730D0/2.D0))/39.37D0
T7     = 0.0770D0/39.37D0
L7     = 5.120D0/39.37D0
T7A    = (((1.5250D0+1.0730D0)/2.D0)+(0.0770D0))/39.37D0
L7A    = ((0.1730D0/2.D0)+(0.1740D0/2.D0))/39.37D0
T8     = 0.0770D0/39.37D0
L8     = (0.6880D0+6.0930D0)/39.37D0
T8A    = ((1.0220D0+0.4320D0)/2.D0+0.0770D0)/39.37D0
L8A    = ((0.1740D0/2.D0)+(0.1820D0/2.D0))/39.37D0
T9     = 0.0610D0/2.D0
L9     = 24.0000D0
*
* Determination of the thermal resistance for the outer facesheet
* closest to the heated surface
*
CALL FACET(T,K1,FLAGK1)
*
R11T = T1/(K1*D*L1)
*
* Determination of the thermal resistance for the outer facesheet
* furthest away from the heated surface
*
T = T - 3.D0*DELTAT
*
CALL FACET(T,K1,FLAGK1)
*
R11B = T1/(K1*D*L1)
*
T = T + 3.D0*DELTAT
*
* Determination of the thermal resistance for the honeycomb edge
* closeouts (-2195)closest to the heated surface (Top and Bottom)
*
T = T - (DELTAT/2.D0)

```



```

*
*      CALL FEDGET(T,K2,FLAGK2)
*
*      R22T = T2/(K2*D*L2)
*
*      T = T + (DELTAT/2.D0)
*
*      Determination of the thermal resistance for the honeycomb edge
*      closeouts (-2195) furthest from the heated surface (Top and
*      Bottom)
*
*      T = T - (2.D0*DELTAT+DELTAT/2.D0)
*
*      CALL FEDGET(T,K2,FLAGK2)
*
*      R22B = T2/(K2*D*L2)
*
*      T = T + (2.D0*DELTAT+DELTAT/2.D0)
*
*      Determination of the thermal resistance for the honeycomb edge
*      closeouts (-2195) closest to the heated surface (Side Supports)
*
*      T = T - (DELTAT/2.D0)
*
*      CALL FEDGET(T,K2,FLAGK2)
*
*      R22AT = T2A/(K2*D*L2A)
*
*      T = T + (DELTAT/2.D0)
*
*      Determination of the thermal resistance for the honeycomb edge
*      closeouts (-2195) furthest from the heated surface (Side Supports)
*
*      T = T - (2.D0*DELTAT+DELTAT/2.D0)
*
*      CALL FEDGET(T,K2,FLAGK2)
*
*      R22B = T2A/(K2*D*L2A)
*
*      T = T + (2.D0*DELTAT+DELTAT/2.D0)
*
*      Determination of the thermal resistance for the upper skin
*      (-2125)
*
*      T = T - DELTAT
*
*      CALL FACET(T,K1,FLAGK1)
*
*      K3 = K1
*
*      R33 = T3/(K3*D*L3)
*
*      T = T + DELTAT
*
*      Determination of the thermal resistance for the left edge channel
*      (-2151) (Top and Bottom)
*
*      T = T - (DELTAT+DELTAT/2.D0)
*

```

```

      CALL FEDGET(T,K2,FLAGK2)
*
      K4 = K2
*
      R44 = T4/(K4*D*L4)
*
* Determination of the thermal resistance for the left edge channel
* (-2151)(Side Supports)
*
      R44A = T4A/(K4*D*L4A)
*
* Determination of the thermal resistance for the right edge channel
* (-2083)(Top and Bottom)
*
      K5 = K2
*
      R55 = T5/(K5*D*L5)
*
* Determination of the thermal resistance for the right edge channel
* (-2083)(Side Supports)
*
      R55A = T5A/(K5*D*L5A)
*
* Determination of the thermal resistance for the torque tube #1
* (-2123)(Top and Bottom)
*
      K6 = K2
*
      R66 = T6/(K6*D*L6)
*
* Determination of the thermal resistance for the torque tube #1
* (-2123)(Side Supports)
*
      R66A = T6A/(K6*D*L6A)
*
* Determination of the thermal resistance for the torque tube #2
* (-2079)(Top and Bottom)
*
      K7 = K2
*
      R77 = T7/(K7*D*L7)
*
* Determination of the thermal resistance for the torque tube #2
* (-2079)(Side Supports)
*
      R77A = T7A/(K7*D*L7A)
*
* Determination of the thermal resistance for the torque tube #3
* (-2081)(Top and Bottom)
*
      K8 = K2
*
      R88 = T8/(K8*D*L8)
*
* Determination of the thermal resistance for the torque tube #3
* (-2081)(Side Supports)
*
      R88A = T8A/(K8*D*L8A)
*

```

```

      T = T + (DELTAT+DELTAT/2.D0)
*
* Determination of the thermal resistance for the lower skin
* (-2085)
*
      T = T - 2.D0*DELTAT
*
      CALL FACET(T,K1,FLAGK1)
*
      R99 = T9/(K9*D*L9)
*
      T = T + 2.D0*DELTAT
*
      RETURN
      END
*
***** Subroutine Contact *****
*
* This subroutine will calculate the thermal resistance for all the
* elements in between which there is a possible contact resistance.
* Here again we will use a unit area for the 1-D model.
* It was made possible to change the value for each Convection
* coefficient in order to analyze different cases. Initially, all the
* convection coefficients are set to be equal.
*
*
* R1C : Contact resistance between the outer facesheet and the
*       honeycomb (2 Places)
* H1C : Convection coefficient for the gap between the outer
*       facesheet and the honeycomb (2 Places)
* B1C : Length for R1C
* R2C : Contact resistance between the outer facesheet and the
*       honeycomb edge closeouts (4 Places)
* H2C : Convection coefficient for the gap between the outer
*       facesheet and the honeycomb edge closeouts (4 Places)
* B2C : Length for R2C
* R3C : Contact resistance between the upper skin and the honeycomb
* H3C : Convection coefficient for the gap between the upper skin
*       and the honeycomb
* B3C : Length for R3C
* R4C : Contact resistance between the upper skin and the honeycomb
*       edge closeouts
* H4C : Convection coefficient for the gap between the upper skin
*       and the honeycomb edge closeouts
* B4C : Length for R4C
* R5C : Contact resistance between the upper skin and the left edge
*       channel
* H5C : Convection coefficient for the gap between the upper skin
*       and the left edge channel
* B5C : Length for R5C
* R6C : Contact resistance between the upper skin and the right edge
*       channel
* H6C : Convection coefficient for the gap between the upper skin
*       and the right edge channel
* B6C : Length for R6C
* R7C : Contact resistance between the upper skin and the torque
*       tube #1
* H7C : Convection coefficient for the gap between the upper skin
*       and the torque tube #1
* B7C : Length for R7C

```



```

+          B6C,B7C,B8C,B9C,B10C,B11C,B12C,B13C,B14C,
+          B15C,B16C
*
D = 1.D0
*
B1C  = (24.000D0-(1.3090D0+1.2720D0)/2.D0)/39.37D0
B2C  = (2.D0*(1.2720D0+1.3090D0)/2.D0)/39.37D0
B3C  = (24.000D0-(1.3090D0+1.2720D0)/2.D0)/39.37D0
B4C  = (2.D0*(1.2720D0+1.3090D0)/2.D0)/39.37D0
B5C  = 1.2160D0/39.37D0
B6C  = 1.3150D0/39.37D0
B7C  = (4.8320D0+(0.1590D0/2.D0)+(0.1730D0/2.D0))/39.370D0
B8C  = (5.120D0+(0.1730D0/2.D0)+(0.1740D0/2.D0))/39.370D0
B9C  = (0.6880D0+6.0930D0+(0.1740D0/2.D0)+(0.1820D0/2.D0))/
+      39.37D0
B10C = 1.2160D0/39.37D0
B11C = 1.3150D0/39.37D0
B12C = (4.8320D0+(0.1590D0/2.D0)+(0.1730D0/2.D0))/39.370D0
B13C = (5.120D0+(0.1730D0/2.D0)+(0.1740D0/2.D0))/39.370D0
B14C = (0.6880D0+6.0930D0+(0.1740D0/2.D0)+(0.1820D0/2.D0))/
+      39.37D0
B15C = (2.D0*(1.2720D0+1.3090D0)/2.D0)/39.37D0
B16C = (2.D0*(1.2720D0+1.3090D0)/2.D0)/39.37D0
*
H1C  = 10.D0
H2C  = H1C
H3C  = H1C
H4C  = H1C
H5C  = H1C
H6C  = H1C
H7C  = H1C
H8C  = H1C
H9C  = H1C
H10C = H1C
H11C = H1C
H12C = H1C
H13C = H1C
H14C = H1C
H15C = H1C
H16C = H1C
*
R1C  = 1.D0/(H1C*D*B1C)
R2C  = 1.D0/(H2C*D*B2C)
R3C  = 1.D0/(H3C*D*B3C)
R4C  = 1.D0/(H4C*D*B4C)
R5C  = 1.D0/(H5C*D*B5C)
R6C  = 1.D0/(H6C*D*B6C)
R7C  = 1.D0/(H7C*D*B7C)
R8C  = 1.D0/(H8C*D*B8C)
R9C  = 1.D0/(H9C*D*B9C)
R10C = 1.D0/(H10C*D*B10C)
R11C = 1.D0/(H11C*D*B11C)
R12C = 1.D0/(H12C*D*B12C)
R13C = 1.D0/(H13C*D*B13C)
R14C = 1.D0/(H14C*D*B14C)
R15C = 1.D0/(H15C*D*B15C)
R16C = 1.D0/(H16C*D*B16C)
*
RETURN

```

END

```
*
***** Subroutine Convection *****
*
* This subroutine will calculate the thermal resistance for all the
* elements that are subjected to convective heat transfer. Here
* again, we will use a unit area for the 1-D model.
*
* R1VT : Thermal resistance for the honeycomb edge closeouts close
*        st to the heated surface
* R1VB : Thermal resistance for the honeycomb edge closeouts
*        furthest from the heated surface
* H1V  : Convection coefficient for the honeycomb edge closeouts
* L1V  : Height of the convection cell of the honeycomb closeouts
* C1C  : Length for R1V
* RAL1 : Raley number for honeycomb edge closeouts
* R2V  : Thermal resistance for cell of torque tube #1
* L2V  : Height of the convection cell of the torque tube #1
* C2V  : Length for R2V
* H2V  : Convection coefficient for the cell of torque tube #1
* RAL2 : Raley number for the cellof torque tube #1
* R3V  : Thermal resistance for cell of torque tube #2
* L3V  : Height of the convection cell for the torque tube #2
* C3V  : Length for R3V
* H3V  : Convection coefficient for the cell of torque tube #2
* RAL3 : Raley number for the cell of torque tube #2
* R4V  : Thermal resistance for cell of torque tube #3
* L4V  : Height of the convection cell for the torque tube #3
* C4V  : Length for R4V
* H4V  : Convection coefficient for the cell of torqur tube #3
* RAL4 : Raley number for the cell of torque tube #3
* R5V  : Thermal resistance for the left edge channel
* L5V  : Height of the convection cell for the left edge channel
* C5V  : Length for R5V
* H5V  : Convection coefficient for the left edge channel
* RAL5 : Raley number for the left edge channel
* R6V  : Thermal resistance for the right edge channel
* L6V  : Height of the convection cell for the right edge channel
* C6V  : Length for R6V
* H6V  : Convection coefficient for the right edge channel
* RAL6 : Raley number for the right edge channel
* KA   : Conduction coefficient for air at given temperature
* T    : Average temperature at which the material properties are
*        calculated at
* D    : Depth
* DELTAT: Maximum temperature differnce between the hot and the cold
*        surfaces
* PRA   : Prandtl number for air at given temperature
* VISCA : Viscosity of air at given temperature
* DIFFA : Diffusivity of air at given temperature
* RAIR1T: Conduction through air for the honeycomb edge closeouts
*        closest to the heated surface
* RAIR1B: Conduction through air for the honeycomb edge closeouts
*        furthest away from the heated surface
* RAIR2 : Conduction through air for the left edge channel
* RAIR3 : Conduction through air for the right edge channel
* RAIR4 : Conduction through air for torque tube #1
* RAIR5 : Conduction through air for torque tube #2
* RAIR6 : Conduction through air for torque tube #3
```

```

*
*****
*
SUBROUTINE CONVECT(R1VT,R1VB,R2V,R3V,R4V,R5V,R6V,RAIR1T,
+               RAIR1B,RAIR2,RAIR3,RAIR4,RAIR5,RAIR6)
*
  IMPLICIT NONE
  DOUBLE PRECISION R1VT,R2V,R3V,R4V,R5V,R6V,H1V,H2V,H3V,H4V,H5V
+               ,H6V,KA,GRAV,BETA,T,DELTAT,DIFFA,VISCA,
+               PRA,D,C1V,C2V,C3V,C4V,C5V,C6V,L1V,L2V,L3V,
+               L4V,L5V,L6V,RAL1,RAL2,RAL3,RAL4,RAL5,RAL6,
+               RAIR1T,RAIR2,RAIR3,RAIR4,RAIR5,RAIR6,R1VB,
+               RAIR1B
  INTEGER FLAGDIFFA,FLAGPRA,FLAGVISCA,FLAGKA
*
  COMMON/VARIABLE/T,DELTAT
*
  GRAV   = 9.807D0
*
  D      = 1.D0
  C1V    = ((1.3090D0+1.2720D0)/2.D0)-(0.0610D0+0.0720D0)/2.D0)
+         /39.37D0
  C2V    = 4.8320D0/39.37D0
  C3V    = 5.1200D0/39.37D0
  C4V    = (0.6880D0+6.0930D0)/39.37D0
  C5V    = (1.2180D0-(0.1480D0/2.D0))/39.37D0
  C6V    = (1.3150D0-0.0960D0)/39.37D0
*
  L1V    = ((0.8350D0+0.8610D0)/2.D0)/39.37D0
  L2V    = ((2.0310D0+1.5830D0)/2.D0)/39.37D0
  L3V    = ((1.5250D0+1.0730D0)/2.D0)/39.37D0
  L4V    = ((1.0220D0+0.4320D0)/2.D0)/39.37D0
  L5V    = 2.0520D0/39.37D0
  L6V    = ((0.4160D0+0.3050D0)/2.D0)/39.37D0
*
  T      = T - DELTAT/2.D0
  BETA   = 1.D0/T
  CALL FDIFFAT(T,DIFFA,FLAGDIFFA)
  CALL FPRAT(T,PRA,FLAGPRA)
  CALL FVISCAT(T,VISCA,FLAGVISCA)
  CALL FKAT(T,KA,FLAGKA)
  T      = T + DELTAT/2.D0
  RAL1   = GRAV*BETA*DELTAT*L1V**3.D0/(DIFFA*VISCA)
  H1V    = KA/L1V*(0.069D0*(RAL1**(1.D0/3.D0))*(PRA**0.074D0))
  R1VT   = 1.D0/(H1V*D*C1V)
  RAIR1T = L1V/(KA*D*C1V)
*
  T      = T - (2.D0*DELTAT+DELTAT/2.D0)
  BETA   = 1.D0/T
  CALL FDIFFAT(T,DIFFA,FLAGDIFFA)
  CALL FPRAT(T,PRA,FLAGPRA)
  CALL FVISCAT(T,VISCA,FLAGVISCA)
  CALL FKAT(T,KA,FLAGKA)
  T      = T + (2.D0*DELTAT+DELTAT/2.D0)
  RAL1   = GRAV*BETA*DELTAT*L1V**3.D0/(DIFFA*VISCA)
  H1V    = KA/L1V*(0.069D0*(RAL1**(1.D0/3.D0))*(PRA**0.074D0))
  R1VB   = 1.D0/(H1V*D*C1V)
  RAIR1B = L1V/(KA*D*C1V)
*

```

```

T      = T - (DELTAT+DELTAT/2.DO)
BETA   = 1.DO/T
CALL FDIFFAT(T,DIFFA,FLAGDIFFA)
CALL FPRAT(T,PRA,FLAGPRA)
CALL FVISCAT(T,VISCA,FLAGVISCA)
CALL FKAT(T,KA,FLAGKA)
T      = T + (DELTAT+DELTAT/2.DO)

*
RAL2 = GRAV*BETA*DELTAT*L2V**3.DO/(DIFFA*VISCA)
RAL3 = GRAV*BETA*DELTAT*L3V**3.DO/(DIFFA*VISCA)
RAL4 = GRAV*BETA*DELTAT*L4V**3.DO/(DIFFA*VISCA)
RAL5 = GRAV*BETA*DELTAT*L5V**3.DO/(DIFFA*VISCA)
RAL6 = GRAV*BETA*DELTAT*L6V**3.DO/(DIFFA*VISCA)

*
H2V = KA/L2V*(0.069D0*(RAL2**(1.DO/3.DO))*(PRA**0.074D0))
H3V = KA/L3V*(0.069D0*(RAL3**(1.DO/3.DO))*(PRA**0.074D0))
H4V = KA/L4V*(0.069D0*(RAL4**(1.DO/3.DO))*(PRA**0.074D0))
H5V = KA/L5V*(0.069D0*(RAL5**(1.DO/3.DO))*(PRA**0.074D0))
H6V = KA/L6V*(0.069D0*(RAL6**(1.DO/3.DO))*(PRA**0.074D0))

*
R2V = 1.DO/(H2V*D*C2V)
R3V = 1.DO/(H3V*D*C3V)
R4V = 1.DO/(H4V*D*C4V)
R5V = 1.DO/(H5V*D*C5V)
R6V = 1.DO/(H6V*D*C6V)

*
RAIR2 = L2V/(KA*D*C2V)
RAIR3 = L3V/(KA*D*C3V)
RAIR4 = L4V/(KA*D*C4V)
RAIR5 = L5V/(KA*D*C5V)
RAIR6 = L6V/(KA*D*C6V)

*
RETURN
END

*
***** Subroutine FKAT *****
*
* This subroutine calculates the conductivity of air at a specific
* temperature. It will call up the subroutine interpolate if
* necessary. The data for the conductivity of air is stored in an
* array. The temperature range is 100 to 800 degrees K.
* The units for the thermal conductivities are (W/m.K).
*
*****
*
SUBROUTINE FKAT(T,KA,FLAG)

*
IMPLICIT NONE
DOUBLE PRECISION KA,KAT(15,2),T
INTEGER FLAG,N

*
N = 15

*
KAT(1,1) = 100.DO
KAT(1,2) = 9.34D-03
KAT(2,1) = 150.DO
KAT(2,2) = 13.8D-03
KAT(3,1) = 200.DO
KAT(3,2) = 18.1D-03

```



```

KAT(4,1) = 250.D0
KAT(4,2) = 22.3D-03
KAT(5,1) = 300.D0
KAT(5,2) = 26.3D-03
KAT(6,1) = 350.D0
KAT(6,2) = 30.0D-03
KAT(7,1) = 400.D0
KAT(7,2) = 33.8D-03
KAT(8,1) = 450.D0
KAT(8,2) = 37.3D-03
KAT(9,1) = 500.D0
KAT(9,2) = 40.7D-03
KAT(10,1)= 550.D0
KAT(10,2)= 43.9D-03
KAT(11,1)= 600.D0
KAT(11,2)= 46.9D-03
KAT(12,1)= 650.D0
KAT(12,2)= 49.7D-03
KAT(13,1)= 700.D0
KAT(13,2)= 52.4D-03
KAT(14,1)= 750.D0
KAT(14,2)= 54.9D-03
KAT(15,1)= 800.D0
KAT(15,2)= 57.3D-03
*
CALL INTERPOL(N,KAT,T,KA,FLAG)
*
RETURN
END
*
***** Subroutine FPRAT *****
*
* This subroutine calculates the Prandtl number for air at a
* specific temperature. It will call up the subroutine interpolate
* if necessary. The data for the Prandtl number is stored in an
* array. The temperature range is 100 to 800 degrees K.
*
*****
*
SUBROUTINE FPRAT(T,PRA,FLAG)
*
IMPLICIT NONE
DOUBLE PRECISION PRA,PRAT(15,2),T
INTEGER FLAG,N
*
N = 15
*
PRAT(1,1) = 100.D0
PRAT(1,2) = 0.786D0
PRAT(2,1) = 150.D0
PRAT(2,2) = 0.758D0
PRAT(3,1) = 200.D0
PRAT(3,2) = 0.737D0
PRAT(4,1) = 250.D0
PRAT(4,2) = 0.720D0
PRAT(5,1) = 300.D0
PRAT(5,2) = 0.707D0
PRAT(6,1) = 350.D0
PRAT(6,2) = 0.700D0

```

```

PRAT(7,1) = 400.D0
PRAT(7,2) = 0.690D0
PRAT(8,1) = 450.D0
PRAT(8,2) = 0.686D0
PRAT(9,1) = 500.D0
PRAT(9,2) = 0.684D0
PRAT(10,1)= 550.D0
PRAT(10,2)= 0.683D0
PRAT(11,1)= 600.D0
PRAT(11,2)= 0.685D0
PRAT(12,1)= 650.D0
PRAT(12,2)= 0.690D0
PRAT(13,1)= 700.D0
PRAT(13,2)= 0.695D0
PRAT(14,1)= 750.D0
PRAT(14,2)= 0.702D0
PRAT(15,1)= 800.D0
PRAT(15,2)= 0.709D0

*
CALL INTERPOL(N,PRAT,T,PRA,FLAG)

*
RETURN
END

*
***** Subroutine FVISCAT *****
*
* This subroutine calculates the viscosity of air at a specific
* temperature. It will call up the subroutine interpolate if
* necessary. The data for the viscosity is stored in an array. The
* temperature range is 100 to 800 degrees K. The units of the
* viscosities are (m2/s).
*
*****
*
SUBROUTINE FVISCAT(T,VISCA,FLAG)

*
IMPLICIT NONE
DOUBLE PRECISION VISCA,VISCAT(15,2),T
INTEGER FLAG,N

*
N = 15

*
VISCAT(1,1) = 100.D0
VISCAT(1,2) = 2.000D-06
VISCAT(2,1) = 150.D0
VISCAT(2,2) = 4.426D-06
VISCAT(3,1) = 200.D0
VISCAT(3,2) = 7.590D-06
VISCAT(4,1) = 250.D0
VISCAT(4,2) = 11.44D-06
VISCAT(5,1) = 300.D0
VISCAT(5,2) = 15.89D-06
VISCAT(6,1) = 350.D0
VISCAT(6,2) = 20.92D-06
VISCAT(7,1) = 400.D0
VISCAT(7,2) = 26.14D-06
VISCAT(8,1) = 450.D0
VISCAT(8,2) = 32.39D-06
VISCAT(9,1) = 500.D0

```

```

VISCAT(9,2) = 38.79D-06
VISCAT(10,1)= 550.D0
VISCAT(10,2)= 45.57D-06
VISCAT(11,1)= 600.D0
VISCAT(11,2)= 52.69D-06
VISCAT(12,1)= 650.D0
VISCAT(12,2)= 60.21D-06
VISCAT(13,1)= 700.D0
VISCAT(13,2)= 68.10D-06
VISCAT(14,1)= 750.D0
VISCAT(14,2)= 76.37D-06
VISCAT(15,1)= 800.D0
VISCAT(15,2)= 84.93D-06
*
CALL INTERPOL(N,VISCAT,T,VISCA,FLAG)
*
RETURN
END
*
***** Subroutine FDIFFAT *****
*
* This subroutine calculates the diffusivity coefficient of air at a *
* specific temperature. It will call up the subroutine interpolate *
* if necessary. The data for the diffusivity coefficient is stored *
* in an array. The temperature range is 100 to 800 degrees k. *
* The units of the diffusivities are (m2/s). *
*
*****
*
SUBROUTINE FDIFFAT(T,DIFFA,FLAG)
*
IMPLICIT NONE
DOUBLE PRECISION DIFFA,DIFFAT(15,2),T
INTEGER FLAG,N
*
N = 15
*
DIFFAT(1,1) = 100.D0
DIFFAT(1,2) = 2.54D-06
DIFFAT(2,1) = 150.D0
DIFFAT(2,2) = 5.84D-06
DIFFAT(3,1) = 200.D0
DIFFAT(3,2) = 10.3D-06
DIFFAT(4,1) = 250.D0
DIFFAT(4,2) = 15.9D-06
DIFFAT(5,1) = 300.D0
DIFFAT(5,2) = 22.5D-06
DIFFAT(6,1) = 350.D0
DIFFAT(6,2) = 29.9D-06
DIFFAT(7,1) = 400.D0
DIFFAT(7,2) = 38.3D-06
DIFFAT(8,1) = 450.D0
DIFFAT(8,2) = 47.2D-06
DIFFAT(9,1) = 500.D0
DIFFAT(9,2) = 56.7D-06
DIFFAT(10,1)= 550.D0
DIFFAT(10,2)= 66.7D-06
DIFFAT(11,1)= 600.D0
DIFFAT(11,2)= 76.9D-06

```

```

        DIFFAT(12,1)= 650.D0
        DIFFAT(12,2)= 87.3D-06
        DIFFAT(13,1)= 700.D0
        DIFFAT(13,2)= 98.0D-06
        DIFFAT(14,1)= 750.D0
        DIFFAT(14,2)= 109.D-06
        DIFFAT(15,1)= 800.D0
        DIFFAT(15,2)= 120.D-06
*
        CALL INTERPOL(N,DIFFAT,T,DIFFA,FLAG)
*
        RETURN
        END
*
***** Subroutine Radiation *****
*
* This subroutine will calculate the thermal resistance for all the
* elements that are subjected to radiative heat transfer. Here
* again, we will use a unit area for the 1-D model. For all the
* channels, we will consider a fourth wall which will be physically
* represented by the insulation, but theoretically we will assume
* the fourth side is made out of the same material as the channel
* itself. The emissivity (EPS**) of the materials used in the
* prototype will have to be either estimated or researched.
*
*
*
* XR*   : The length of the radiating and absorbing surfaces
* YR*   : The width of the radiating and absorbing surfaces
* LR*   : The distance the radiating and the absorbing surfaces are
*         apart (the height of the cell)
* XR*BAR: Defined as a unitless length
* YR*BAR: Defined as a unitless length
* EPS*  : The emissivity coefficient of the material
* F*    : View factor for the specific material
*
*
* Determination of the view factor for:
* - the left edge channel      (-2151) #1
* - the right edge channel     (-2083) #2
* - the honeycomb edge closeouts (-2195) #3
* - the torque tube #1        (-2123) #4
* - the torque tube #2        (-2079) #5
* - the torque tube #3        (-2181) #6
* The equation for the view factor can be found in Incropera and
* Dewitt 1990 on Page 798 Table 13.2.
* That shape factor did not work. We used one out of radiation heat
* transfer by E.M Sparrow and R.D. Cess augmented edition.
*
*****
*
        SUBROUTINE RADIAT(RRAD1,RRAD2,RRAD3T,RRAD3B,RRAD4,RRAD5,RRAD6)
*
        IMPLICIT NONE
        DOUBLE PRECISION SIGMA,PI,EPS1,EPS2,EPS3,EPS4,EPS5,EPS6,
+           XR1,XR2,XR3,XR4,XR5,XR6,YR1,YR2,YR3,YR4,
+           YR5,YR6,LR1,LR2,LR3,LR4,LR5,LR6,XR1BAR,
+           XR2BAR,XR3BAR,XR4BAR,XR5BAR,XR6BAR,YR1BAR,
+           YR2BAR,YR3BAR,YR4BAR,YR5BAR,YR6BAR,F1,F2,F3,
+           F4,F5,F6,RR1,RR2,RR3,RR4,RR5,RR6,QRAD1,QRAD2,
+           QRAD3,QRAD4,QRAD5,QRAD6,RRAD1,RRAD2,RRAD3T,

```

```

+          RRAD3B,RRAD4,RRAD5,RRAD6,DELTAT,T
*
*   COMMON/VARIABLE/T,DELTAT
*
*   SIGMA  = 5.67D-08
*   PI     = DACOS(-1.D0)
*
*   EPS1   = 0.70D0
*   XR1    = (1.2180D0-(0.1480D0/2.D0))/39.37D0
*   YR1    = 1.D0
*   LR1    = ((2.1270D0+2.0520D0)/2.D0)/39.37D0
*   XR1BAR = XR1 / LR1
*   YR1BAR = YR1 / LR1
*
*   EPS2   = EPS1
*   XR2    = (1.3150D0-0.0960D0)/39.37D0
*   YR2    = 1.D0
*   LR2    = ((0.4160D0+0.3050D0)/2.D0)/39.37D0
*   XR2BAR = XR2 / LR2
*   YR2BAR = YR2 / LR2
*
*   EPS3   = EPS1
*   XR3    = (((1.3090D0+1.2720D0)/2.D0)-((0.0610D0+0.0720D0)/2.D0
+       ))/39.37D0
*   YR3    = 1.D0
*   LR3    = ((0.8350D0+0.8610D0)/2.D0)/39.37D0
*   XR3BAR = XR3 / LR3
*   YR3BAR = YR3 / LR3
*
*   EPS4   = EPS1
*   XR4    = 4.8320D0/39.37D0
*   YR4    = 1.D0
*   LR4    = ((2.0130D0+1.5830D0)/2.D0)/39.37D0
*   XR4BAR = XR4 / LR4
*   YR4BAR = YR4 / LR4
*
*   EPS5   = EPS1
*   XR5    = 5.120D0/39.37D0
*   YR5    = 1.D0
*   LR5    = ((1.5250D0+1.0730D0)/2.D0)/39.37D0
*   XR5BAR = XR5 / LR5
*   YR5BAR = YR5 / LR5
*
*   EPS6   = EPS1
*   XR6    = (0.688D0+6.0930D0)/39.37D0
*   YR6    = 1.D0
*   LR6    = ((1.0220D0+0.4320D0)/2.D0)/39.37D0
*   XR6BAR = XR6 / LR6
*   YR6BAR = YR6 / LR6
*
*   F1     = (2.D0/(PI*XR1BAR*YR1BAR))*(DLOG(((1.D0+XR1BAR**2.D0)*(1.D0
+       +YR1BAR**2.D0)/(1.D0+XR1BAR**2.D0+YR1BAR**2.D0))*0.5D0))+
*   +       XR1BAR*((1.D0+YR1BAR**2.D0)**0.5D0)*DATAN(XR1BAR/((1.D0+
*   +       YR1BAR**2.D0)**0.5D0))+YR1BAR*((1.D0+XR1BAR**2.D0)**0.5D0)
*   +       *DATAN(YR1BAR/((1.D0+XR1BAR**2.D0)**0.5D0))-XR1BAR*DATAN(
*   +       XR1BAR)-YR1BAR*DATAN(YR1BAR)
*
*   F2     = (2.D0/(PI*XR2BAR*YR2BAR))*(DLOG(((1.D0+XR2BAR**2.D0)*(1.D0
+       +YR2BAR**2.D0)/(1.D0+XR2BAR**2.D0+YR2BAR**2.D0))*0.5D0))+

```

```

*      +      XR2BAR*((1.D0+YR2BAR**2.D0)**0.5D0)*DATAN(XR2BAR/((1.D0+
*      +      YR2BAR**2.D0)**0.5D0))+YR2BAR*((1.D0+XR2BAR**2.D0)**0.5D0)
*      +      *DATAN(YR2BAR/((1.D0+XR2BAR**2.D0)**0.5D0))-XR2BAR*DATAN(
*      +      XR2BAR)-YR2BAR*DATAN(YR2BAR)
*
*      F3  = (2.D0/(PI*XR3BAR*YR3BAR))*(DLOG(((1.D0+XR3BAR**2.D0)*(1.D0
*      +      +YR3BAR**2.D0)/(1.D0+XR3BAR**2.D0+YR3BAR**2.D0))**0.5D0))+
*      +      XR3BAR*((1.D0+YR3BAR**2.D0)**0.5D0)*DATAN(XR3BAR/((1.D0+
*      +      YR3BAR**2.D0)**0.5D0))+YR3BAR*((1.D0+XR3BAR**2.D0)**0.5D0)
*      +      *DATAN(YR3BAR/((1.D0+XR3BAR**2.D0)**0.5D0))-XR3BAR*DATAN(
*      +      XR3BAR)-YR3BAR*DATAN(YR3BAR)
*
*      F4  = (2.D0/(PI*XR4BAR*YR4BAR))*(DLOG(((1.D0+XR4BAR**2.D0)*(1.D0
*      +      +YR4BAR**2.D0)/(1.D0+XR4BAR**2.D0+YR4BAR**2.D0))**0.5D0))+
*      +      XR4BAR*((1.D0+YR4BAR**2.D0)**0.5D0)*DATAN(XR4BAR/((1.D0+
*      +      YR4BAR**2.D0)**0.5D0))+YR4BAR*((1.D0+XR4BAR**2.D0)**0.5D0)
*      +      *DATAN(YR4BAR/((1.D0+XR4BAR**2.D0)**0.5D0))-XR4BAR*DATAN(
*      +      XR4BAR)-YR4BAR*DATAN(YR4BAR)
*
*      F5  = (2.D0/(PI*XR5BAR*YR5BAR))*(DLOG(((1.D0+XR5BAR**2.D0)*(1.D0
*      +      +YR5BAR**2.D0)/(1.D0+XR5BAR**2.D0+YR5BAR**2.D0))**0.5D0))+
*      +      XR5BAR*((1.D0+YR5BAR**2.D0)**0.5D0)*DATAN(XR5BAR/((1.D0+
*      +      YR5BAR**2.D0)**0.5D0))+YR5BAR*((1.D0+XR5BAR**2.D0)**0.5D0)
*      +      *DATAN(YR5BAR/((1.D0+XR5BAR**2.D0)**0.5D0))-XR5BAR*DATAN(
*      +      XR5BAR)-YR5BAR*DATAN(YR5BAR)
*
*      F6  = (2.D0/(PI*XR6BAR*YR6BAR))*(DLOG(((1.D0+XR6BAR**2.D0)*(1.D0
*      +      +YR6BAR**2.D0)/(1.D0+XR6BAR**2.D0+YR6BAR**2.D0))**0.5D0))+
*      +      XR6BAR*((1.D0+YR6BAR**2.D0)**0.5D0)*DATAN(XR6BAR/((1.D0+
*      +      YR6BAR**2.D0)**0.5D0))+YR6BAR*((1.D0+XR6BAR**2.D0)**0.5D0)
*      +      *DATAN(YR6BAR/((1.D0+XR6BAR**2.D0)**0.5D0))-XR6BAR*DATAN(
*      +      XR6BAR)-YR6BAR*DATAN(YR6BAR)
*
*      F1  = 1.D0/(2.D0*XR1)*((2.D0*(LR1**2.D0+XR1**2.D0)**0.5D0-
*      +      2.D0*LR1))
*
*      F2  = 1.D0/(2.D0*XR2)*((2.D0*(LR2**2.D0+XR2**2.D0)**0.5D0-
*      +      2.D0*LR2))
*
*      F3  = 1.D0/(2.D0*XR3)*((2.D0*(LR3**2.D0+XR3**2.D0)**0.5D0-
*      +      2.D0*LR3))
*
*      F4  = 1.D0/(2.D0*XR4)*((2.D0*(LR4**2.D0+XR4**2.D0)**0.5D0-
*      +      2.D0*LR4))
*
*      F5  = 1.D0/(2.D0*XR5)*((2.D0*(LR5**2.D0+XR5**2.D0)**0.5D0-
*      +      2.D0*LR5))
*
*      F6  = 1.D0/(2.D0*XR6)*((2.D0*(LR6**2.D0+XR6**2.D0)**0.5D0-
*      +      2.D0*LR6))
*
*      Determination of the radiative heat transfer coefficient for:
*      - the left edge channel      (-2151) #1
*      - the right edge channel     (-2083) #2
*      - the honeycomb edge closeouts (-2195) #3
*      - the torque tube #1        (-2123) #4
*      - the torque tube #2        (-2079) #5
*      - the torque tube #3        (-2181) #6
*

```

```

*
      RRAD1 = 1.D0/(SIGMA*EPS1*F1*(2*T-DELTAT)*((T-DELTAT)-
+          (DELTAT))**2.D0
+          + (T-DELTAT)**2.D0)*XR1*YR1)
*
      RRAD2 = 1.D0/(SIGMA*EPS2*F2*(2*T-DELTAT)*((T-DELTAT)-
+          (DELTAT))**2.D0
+          + (T-DELTAT)**2.D0)*XR2*YR2)
*
      RRAD3T = 1.D0/(SIGMA*EPS3*F3*(2*T-DELTAT)*((T-DELTAT)**2.D0
+          + T**2.D0)*XR3*YR3)
*
      RRAD3B = 1.D0/(SIGMA*EPS3*F3*(2*T-DELTAT)*((T-2.D0*DELTAT)
+          - (DELTAT))**2.D0 + (T-2.D0*DELTAT)**2.D0)*XR3*YR3)
*
      RRAD4 = 1.D0/(SIGMA*EPS4*F4*(2*T-DELTAT)*((T-DELTAT)-
+          (DELTAT))**2.D0
+          + (T-DELTAT)**2.D0)*XR4*YR4)
*
      RRAD5 = 1.D0/(SIGMA*EPS5*F5*(2*T-DELTAT)*((T-DELTAT)-
+          (DELTAT))**2.D0
+          + (T-DELTAT)**2.D0)*XR5*YR5)
*
      RRAD6 = 1.D0/(SIGMA*EPS6*F6*(2*T-DELTAT)*((T-DELTAT)-
+          (DELTAT))**2.D0
+          + (T-DELTAT)**2.D0)*XR6*YR6)
*
      RETURN
      END
*
***** Subroutine FSS321T *****
*
* This subroutine calculates the thermal conductivity of Stainless
* Steel 321. It will call up the subroutine interpolate if necessary.
* The data for the thermal conductivity is stored in an array. The
* temperature range is 366.48 to 699.82 degrees K. The units of the
* thermal conductivities are (W/m.K).
*
*****
*
      SUBROUTINE FSS321T(T,KSS,FLAG)
*
      IMPLICIT NONE
      DOUBLE PRECISION KSS,KSST(13,2),T
      INTEGER FLAG,N
*
      N = 13
*
      KSST(1,1) = (200.D0+459.56D0)*5.D0/9.D0
      KSST(1,2) = 9.273D0*(1.D0/0.57782D0)
      KSST(2,1) = (250.D0+459.56D0)*5.D0/9.D0
      KSST(2,2) = 9.636D0*(1.D0/0.57782D0)
      KSST(3,1) = (300.D0+459.56D0)*5.D0/9.D0
      KSST(3,2) = 10.00D0*(1.D0/0.57782D0)
      KSST(4,1) = (350.D0+459.56D0)*5.D0/9.D0
      KSST(4,2) = 10.27D0*(1.D0/0.57782D0)
      KSST(5,1) = (400.D0+459.56D0)*5.D0/9.D0
      KSST(5,2) = 10.46D0*(1.D0/0.57782D0)
      KSST(6,1) = (450.D0+459.56D0)*5.D0/9.D0

```

```

      KSST(6,2) = 10.76D0*(1.D0/0.57782D0)
      KSST(7,1) = (500.D0+459.56D0)*5.D0/9.D0
      KSST(7,2) = 11.05D0*(1.D0/0.57782D0)
      KSST(8,1) = (550.D0+459.56D0)*5.D0/9.D0
      KSST(8,2) = 11.27D0*(1.D0/0.57782D0)
      KSST(9,1) = (600.D0+459.56D0)*5.D0/9.D0
      KSST(9,2) = 11.51D0*(1.D0/0.57782D0)
      KSST(10,1)= (650.D0+459.56D0)*5.D0/9.D0
      KSST(10,2)= 11.73D0*(1.D0/0.57782D0)
      KSST(11,1)= (700.D0+459.56D0)*5.D0/9.D0
      KSST(11,2)= 11.84D0*(1.D0/0.57782D0)
      KSST(12,1)= (750.D0+459.56D0)*5.D0/9.D0
      KSST(12,2)= 12.00D0*(1.D0/0.57782D0)
      KSST(13,1)= (800.D0+459.56D0)*5.D0/9.D0
      KSST(13,2)= 12.25D0*(1.D0/0.57782D0)

*
      CALL INTERPOL(N,KSST,T,KSS,FLAG)
*
      RETURN
      END
*
***** Subroutine FACET *****
*
* This subroutine will calculate the thermal conductivity for the
* outer facesheets. It will call up the subroutine interpolation if
* necessary. The data is stored in an array. The temperature range
* is 219.15 to 433.15 K. The units of the thermal conductivity are
* are (W/m.K). The approximate values were used from IM7/977-3
* a similar material. The measurements were done by HOLOMETRIX.
*
*****
*
      SUBROUTINE FACET(T,K1,FLAG)
*
      IMPLICIT NONE
      DOUBLE PRECISION K1,K1T(3,2),T
      INTEGER FLAG,N
*
      N = 3
*
      K1T(1,1) = 219.15D0
      K1T(1,2) = 0.52D0
      K1T(2,1) = 294.15D0
      K1T(2,2) = 0.62D0
      K1T(3,1) = 433.15D0
      K1T(3,2) = 0.55D0
*
      CALL INTERPOL(N,K1T,T,K1,FLAG)
*
      RETURN
      END
*
***** Subroutine FEDGET *****
*
* This subroutine will calculate the thermal conductivity for the
* honeycomb edge closeouts. It will call up the subroutine
* interpolation if necessary. The data is stored in an array. The
* temperature range is 100 to 400 K. The units of the thermal
* conductivity are (W/m.K). The approximate values were used
*

```



```

* from the datasheet of Graphite Fiber Epoxy.
*
*****
*
      SUBROUTINE FEDGET(T,K2,FLAG)
*
      IMPLICIT NONE
      DOUBLE PRECISION K2,K2T(8,2),T
      INTEGER FLAG,N
*
      N = 8
*
      K2T(1,1) = 100.D0
      K2T(1,2) = 0.0046D0*100.D0
      K2T(2,1) = 150.D0
      K2T(2,2) = 0.0058D0*100.D0
      K2T(3,1) = 200.D0
      K2T(3,2) = 0.0068D0*100.D0
      K2T(4,1) = 250.D0
      K2T(4,2) = 0.0078D0*100.D0
      K2T(5,1) = 273.D0
      K2T(5,2) = 0.0082D0*100.D0
      K2T(6,1) = 300.D0
      K2T(6,2) = 0.0087D0*100.D0
      K2T(7,1) = 350.D0
      K2T(7,2) = 0.0096D0*100.D0
      K2T(8,1) = 400.D0
      K2T(8,2) = 0.0105D0*100.D0
*
      CALL INTERPOL(N,K2T,T,K2,FLAG)
*
      RETURN
      END
*
***** Subroutine FCLOST *****
*
* This subroutine will calculate the thermal conductivity for the
* closure rib. It will call the subroutine interpolate if necessary.
* The data is stored in an array. The temperature range is 255.37 to
* 699.82 K. The units of the thermal conductivities are (W/m.K).
* The approximate values were taken from the datasheet for Ti-6Al-4V.
*
*****
*
      SUBROUTINE FCLOST(T,K10,FLAG)
*
      IMPLICIT NONE
      DOUBLE PRECISION K10,K10T(9,2),T
      INTEGER FLAG,N
*
      N = 9
*
      K10T(1,1) = (000.D0+459.67D0)*5.D0/9.D0
      K10T(1,2) = 4.00D0/0.57782D0
      K10T(2,1) = (100.D0+459.67D0)*5.D0/9.D0
      K10T(2,2) = 4.58D0/0.57782D0
      K10T(3,1) = (200.D0+459.67D0)*5.D0/9.D0
      K10T(3,2) = 5.08D0/0.57782D0
      K10T(4,1) = (300.D0+459.67D0)*5.D0/9.D0

```

```

K10T(4,2) = 5.67D0/0.57782D0
K10T(5,1) = (400.D0+459.67D0)*5.D0/9.D0
K10T(5,2) = 6.17D0/0.57782D0
K10T(6,1) = (500.D0+459.67D0)*5.D0/9.D0
K10T(6,2) = 6.83D0/0.57782D0
K10T(7,1) = (600.D0+459.67D0)*5.D0/9.D0
K10T(7,2) = 7.42D0/0.57782D0
K10T(8,1) = (700.D0+459.67D0)*5.D0/9.D0
K10T(8,2) = 8.00D0/0.57782D0
K10T(9,1) = (800.D0+459.67D0)*5.D0/9.D0
K10T(9,2) = 8.67D0/0.57782D0
*
CALL INTERPOL(N,K10T,T,K10,FLAG)
*
RETURN
END
*
*****

```

Appendix E

EAL One-Dimensional Effective Insulated Model: Forward Program

```
*XQT U1
*ONLINE=0
!MAXCM = SSP(0,13)
*CMPARA(MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of the $
$                               variables used in the program such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $ arithmetic utility system processor
$
!NY1  = 40          $ number of nodes for the effective structure
!NY2  = 10          $ number of nodes for the insulation
$
!EY1  ="NY1"-1      $ number of elements for the effective structure
!EY2  ="NY2"-1      $ number of elements for the insulation
$
!TOTN = "NY1"+"EY2" $ total number of nodes of the model
!NON1  = "NY1"       $ node number of the node closest to the heater
!NONI  = "NY1"+"EY2" $ node number of the node on top of the insulation
!NONJ  = FLOAT("NONI") $ node number of the node on top of the insulation
$                               $ as a real
$
$
*RETURN
*VARI
$
```

```

$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the $
$                               properties of the materials used $
$                               in the model. $
$-----$
$
$(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for effective structure
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
                        J=1: 1.E+5 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
                        J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
                        J=1: 256.29 1046.64 0.0375 0.0375 0.0375
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$                               coordinates of the points used in $
$                               the geometry of the plate. The $
$                               points cited in the subroutine $
$                               describe the following $
$                               substructures: $
$                               1,2 :effective structure $
$                               2A,3 :insulation $
$-----$
$
$(29 PLAT DIME)DIME
$
!X1 = 0. $ x-coordinate for point 1
!X2 = 0. $ x-coordinate for point 2
!X3 = 0. $ x-coordinate for point 3
$
!Y1 = 0. $ y-coordinate for point 1
!Y2 = 0.1235 $ y-coordinate for point 2
!Y2A = 0.1235+(0.0254/"EY2") $ y-coordinate for point 2A
!Y3 = 0.1489 $ y-coordinate for point 3
$
!Z1 = 0. $ z-coordinate for point 1
!Z2 = 0. $ z-coordinate for point 2
!Z3 = 0. $ z-coordinate for point 3
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the coordinates $

```

```

$                                     of the nodes used in the mesh.      $
$-----$
$
$(29 NODE GENE)GENE
$
!INCY = 1          $nodal increment in the y-direction
!T2  = 1 + "NY1"   $starting node number for insulation
$
*XQT TAB           $defines position coordinates of nodes
$
START "TOTN"       $define the total number of nodes
JLOC              $define the joint locations
FORMAT = 1        $specifies rectangular coordinates
$
1,"X1","Y1","Z1","X2","Y2","Z2","NY1","INCY"
*IF("EY2" GT 1):*GOTO 100
"T2","X3","Y3","Z3"
*GOTO 105
*LABEL 100
"T2","X2","Y2A","Z2","X3","Y3","Z3","EY2","INCY"
*LABEL 105
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the      $
$                               elements used in the mesh.        $
$-----$
$
$(29 MAKE ELEM)ELEM
$
*XQT AUS           $ arithmetic utility system processor
$
TABLE(NI=1,NJ=1): K AREA : I = 1  $ section property table for
                        J=1: 1.0    $ K21 conduction elements
$
*XQT ELD           $ element definition processor
RESET NUTED = 1    $ destination library
$
!NOPT = 1
!CLOS = 0
$
!J1 = 1            $ group 1 lower outer facesheet
!J2 = "J1"+"INCY"  $ group 1 lower outer facesheet
!J3 = "J1"+"INCY"*"EY1" $ group 2 heater
!J4 = "J1"+"INCY"*"EY1" $ group 2 heater
!J5 = "J4"         $ group 3 insulation
!J6 = "J5"+"INCY"  $ group 3 insulation
$
K21                $ 1-D conduction elements
$
GROUP = 1          $ group 1 lower outer facesheet
NMAT = 1           $ material 1 conduction properties 1
"J1","J2","NOPT","EY1",1
$
GROUP = 2          $ group 2 heater
NMAT = 2           $ material 2 conduction properties 2
"J3","J4","NOPT",1,1

```

```

$
GROUP = 3                                $ group 3 insulation
NMAT = 3                                $ material 3 conduction properties 3
"J5","J6","NOPT","EY2",1
$
*XQT TGEO                                $ element geometry processor
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the      $
$                               boundary conditions such as:      $
$                               - the constant temperature        $
$                               boundary                          $
$                               - the applied heat flux boundary  $
$                               - the total experimental and      $
$                               heating time                      $
$                               - the heated node boundary        $
$                               temperature                       $
$-----$
$
*(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the
$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=3,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$

```

```

*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT HTMP
DEFINE EE = 5 DELT FLUX
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 2
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=EE,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$

```

```

$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
TABLE(NI=1,NJ=1):5 TEMP NODE
  I=1
      J=1: 1.0
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 2):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=1,NINJ=1
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.           $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrtral time step for TRTB (seconds)
$
!MIN1 = 1.0      $ lower bound for skins (PCp)
!MIN2 = 0.1      $ lower bound for skins (K)
!MIN3 = 10.0     $ lower bound for honeycomb (PCp)
!MIN4 = 10.0     $ lower bound for honeycomb (K)
!MIN5 = 10.0     $ lower bound for torque tubes (PCp)
!MIN6 = 10.0     $ lower bound for torque tubes (K)
!MIN7 = 10.0     $ lower bound for facesheets (PCp)
!MIN8 = 10.0     $ lower bound for facesheets (K)
!MIN9 = 10.0     $ lower bound for air gap (PCp)
!MINX = 10.0     $ lower bound for air gap (K)
!MAX1 = 5.0      $ upper bound for skins (PCp)
!MAX2 = 2.0      $ upper bound for skins (K)
!MAX3 = 10.0     $ upper bound for honeycomb (PCp)

```



```

!MAX4 = 10.0      $ upper bound for honeycomb (K)
!MAX5 = 10.0      $ upper bound for torque tubes (PCp)
!MAX6 = 10.0      $ upper bound for torque tubes (K)
!MAX7 = 10.0      $ upper bound for facesheets (PCp)
!MAX8 = 10.0      $ upper bound for facesheets (K)
!MAX9 = 10.0      $ upper bound for air gap (PCp)
!MAXX = 10.0      $ upper bound for air gap (K)
$
!A1  = 2.54      $ Initial value for skins (PCp)
!A2  = 1.00      $ Initial value for skins (K)
!A3  = 1.00      $ Initial value for honeycomb (PCp)
!A4  = 1.00      $ Initial value for honeycomb (K)
!A5  = 1.00      $ Initial value for torque tubes (PCp)
!A6  = 1.00      $ Initial value for torque tubes (K)
!A7  = 1.00      $ Initial value for facesheets (PCp)
!A8  = 1.00      $ Initial value for facesheets (K)
!A9  = 1.00      $ Initial value for air gap (PCp)
!A10 = 1.00      $ Initial value for air gap (K)
$
!A1I = 2.54      $ Initial value for skins (PCp)
!A2I = 1.00      $ Initial value for skins (K)
!A3I = 1.00      $ Initial value for honeycomb (PCp)
!A4I = 1.00      $ Initial value for honeycomb (K)
!A5I = 1.00      $ Initial value for torque tubes (PCp)
!A6I = 1.00      $ Initial value for torque tubes (K)
!A7I = 1.00      $ Initial value for facesheets (PCp)
!A8I = 1.00      $ Initial value for facesheets (K)
!A9I = 1.00      $ Initial value for air gap (PCp)
!A10I = 1.00     $ Initial value for air gap (K)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by $
$                               changing the Crank Nicholson para- $
$                               meter.                               $
$-----$
$
*(29 TRAN ANAL)ANAL
$
*XQT TRTB      $ implicit transient analyzer
RESET QLIB = 5  $ thermal excitation library
RESET MLIB = 6  $ material and fluids properties library
RESET T1  = "TIMI" $ starting time of analysis
RESET T2  = "TIMF" $ termination time of analysis
RESET DT  = "INTS" $ integration time step size
RESET DEST = 4   $ output library
RESET PRINT = 0  $ online print control parameter
RESET KPRT = 0   $ print control parameter for k matrix
RESET MXNDT = 10000000 $ maximum # of int. time steps during TRTB
RESET BETA  = 0.5 $ Crank Nicholson parameter
RESET KTI   = 1   $ K matrix computation time parameter
KTIME      = 10000. $ times to calculate the conduction load vector
TEMP       = "CSTP" $ initial temperature of structure at T1
TSAVE      = 5 EXPE TIME 1 1 $ storage of nodal temperatures at time

```

```

$                                $ increments of TSAVE
*RETURN
*ANAL
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU                        $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1          $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1          $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                        $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME)    $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1                 $ heated node number - 1
!VAR3 = "TOTN"-1                 $ total node number - 1
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 1):*GOTO 5000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2):*GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3):*GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4):*GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5):*GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6):*GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7):*GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8):*GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9):*GOTO 5080 $ Temperature histories for property 7
$
*LABEL 5000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
$*XQT DCU
$ PRINT 5 TEMP ERAT 1 1
$*XQT AUS
$
*GOTO 6000
$

```

```

*LABEL 5010
TABLE(NI=1,NJ="VAR1"):2 INIT 00 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ Subroutine (29 SENS COEF)COEF: This subroutine calculates $
$                               the sensitivity coefficients $
$                               for up to seven parameters. $
$-----$
$
$(29 SENS COEF)COEF
$
*XQT AUS                      $ Arithmetic Utility System
$
INLIB = 1                      $ Read from library 1
OUTLI = 1                      $ Write to library 1
$
*IF("N2" EQ 0) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 1) :*GOTO 1000    $ No perturbations

```

```

*IF("N2" EQ 2) : *GOTO 1000      $ No perturbations
*IF("N2" EQ 3) : *GOTO 1010      $ Property 1 is perturbed
*IF("N2" EQ 4) : *GOTO 1020      $ Property 2 is perturbed
*IF("N2" EQ 5) : *GOTO 1030      $ Property 3 is perturbed
*IF("N2" EQ 6) : *GOTO 1040      $ Property 4 is perturbed
*IF("N2" EQ 7) : *GOTO 1050      $ Property 5 is perturbed
*IF("N2" EQ 8) : *GOTO 1060      $ Property 6 is perturbed
*IF("N2" EQ 9) : *GOTO 1070      $ Property 7 is perturbed
*IF("N2" EQ 10) : *GOTO 1080     $ Property 8 is perturbed
*IF("N2" EQ 11) : *GOTO 1090     $ Property 9 is perturbed
*IF("N2" EQ 12) : *GOTO 1100     $ Property 10 is perturbed
*GOTO 2000
$
*LABEL 1000
$
!A1  = "A1I"                      $ Set property 1 to initial guess
!A2  = "A2I"                      $ Set property 2 to initial guess
!A3  = "A3I"                      $ Set property 3 to initial guess
!A4  = "A4I"                      $ Set property 4 to initial guess
!A5  = "A5I"                      $ Set property 5 to initial guess
!A6  = "A6I"                      $ Set property 6 to initial guess
!A7  = "A7I"                      $ Set property 7 to initial guess
!A8  = "A8I"                      $ Set property 8 to initial guess
!A9  = "A9I"                      $ Set property 9 to initial guess
!A10 = "A10I"                    $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)              $ Update the thermal properties
*DCALL(29 TRAN ANAL)              $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)              $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1010
$
!A1  = 1.001*"A1I"                $ Perturb property 1
!DA1 = 0.001*"A1I"                $ Amount property 1 is perturbed
!A2  = "A2I"                      $ Set property 2 to initial guess
!A3  = "A3I"                      $ Set property 3 to initial guess
!A4  = "A4I"                      $ Set property 4 to initial guess
!A5  = "A5I"                      $ Set property 5 to initial guess
!A6  = "A6I"                      $ Set property 6 to initial guess
!A7  = "A7I"                      $ Set property 7 to initial guess
!A8  = "A8I"                      $ Set property 8 to initial guess
!A9  = "A9I"                      $ Set property 9 to initial guess
!A10 = "A10I"                    $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)              $ Update the thermal properties
*DCALL(29 TRAN ANAL)              $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)              $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1020
$
!A1  = "A1I"                      $ Set property 1 to initial guess
!A2  = 1.001*"A2I"                $ Perturb property 2
!DA2 = 0.001*"A2I"                $ Amount property 2 is perturbed
!A3  = "A3I"                      $ Set property 3 to initial guess
!A4  = "A4I"                      $ Set property 4 to initial guess
!A5  = "A5I"                      $ Set property 5 to initial guess
!A6  = "A6I"                      $ Set property 6 to initial guess

```

!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1030	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = 1.001*"A3I"	\$ Perturb property 3
!DA3 = 0.001*"A3I"	\$ Amount property 3 is perturbed
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1040	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = 1.001*"A4I"	\$ Perturb property 4
!DA4 = 0.001*"A4I"	\$ Amount property 4 is perturbed
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures)
\$	
*GOTO 2000	
*LABEL 1050	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = 1.001*"A5I"	\$ Perturb property 5
!DA5 = 0.001*"A5I"	\$ Amount property 5 is perturbed
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess

!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1060	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = 1.001*"A6I"	\$ Perturb property 6
!DA6 = 0.001*"A6I"	\$ Amount property 6 is perturbed
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1070	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = 1.001*"A7I"	\$ Perturb property 7
!DA7 = 0.001*"A7I"	\$ Amount property 7 is perturbed
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1080	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = 1.001*"A8I"	\$ Perturb property 8
!DA8 = 0.001*"A8I"	\$ Amount property 8 is perturbed
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess

\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1090	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = 1.001*"A9I"	\$ Perturb property 9
!DA9 = 0.001*"A9I"	\$ Amount property 9 is perturbed
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1100	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = 1.001*"A10I"	\$ Perturb property 10
!DA10 = 0.001*"A10I"	\$ Amount property 10 is perturbed
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 2000	
\$	
*IF("N2" EQ 0) : *GOTO 3000	\$ No sensitivity coefficients are calculated
*IF("N2" EQ 1) : *GOTO 3000	\$ No sensitivity coefficients are calculated
*IF("N2" EQ 2) : *GOTO 3000	\$ No sensitivity coefficients are calculated
*IF("N2" EQ 3) : *GOTO 3010	\$ Sensitivity coefficients for property 1
*IF("N2" EQ 4) : *GOTO 3020	\$ Sensitivity coefficients for property 1,2
*IF("N2" EQ 5) : *GOTO 3030	\$ Sensitivity coefficients for property 1,2,3
*IF("N2" EQ 6) : *GOTO 3040	\$ Sensitivity coefficients for property 1,2,3,4
*IF("N2" EQ 7) : *GOTO 3050	\$ Sensitivity coefficients for property 1,2,3,4,5
*IF("N2" EQ 8) : *GOTO 3060	\$ Sensitivity coefficients for property 1,2,3,4,5,6
*IF("N2" EQ 9) : *GOTO 3070	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7
*IF("N2" EQ 10) : *GOTO 3080	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7,8
*IF("N2" EQ 11) : *GOTO 3090	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9
*IF("N2" EQ 12) : *GOTO 3100	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9,10

```

*GOTO 3000
$
*LABEL 3000
DEFINE T0 = 2 INIT 00 1 1 $ Initial temperature distribution with no perturbations
*GOTO 4000
$
*LABEL 3010
!S1P0 = 1.0/"DA1" $ Positive denominator used to numerically differentiate
!S1NE = -1.0/"DA1" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T1 = 2 PROP I 1 1 $ Temperature distribution for perturbed property 1
DIF1 = SUM("S1P0" T1,"S1NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3020
!S2P0 = 1.0/"DA2" $ Positive denominator used to numerically differentiate
!S2NE = -1.0/"DA2" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T2 = 2 PROP II 1 1 $ Temperature distribution for perturbed property 2
DIF2 = SUM("S2P0" T2,"S2NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3030
!S3P0 = 1.0/"DA3" $ Positive denominator used to numerically differentiate
!S3NE = -1.0/"DA3" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T3 = 2 PROP III 1 1 $ Temperature distribution for perturbed property 3
DIF3 = SUM("S3P0" T3,"S3NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3040
!S4P0 = 1.0/"DA4" $ Positive denominator used to numerically differentiate
!S4NE = -1.0/"DA4" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T4 = 2 PROP IV 1 1 $ Temperature distribution for perturbed property 4
DIF4 = SUM("S4P0" T4,"S4NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3050
!S5P0 = 1.0/"DA5" $ Positive denominator used to numerically differentiate
!S5NE = -1.0/"DA5" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T5 = 2 PROP V 1 1 $ Temperature distribution for perturbed property 5
DIF5 = SUM("S5P0" T5,"S5NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3060
!S6P0 = 1.0/"DA6" $ Positive denominator used to numerically differentiate
!S6NE = -1.0/"DA6" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T6 = 2 PROP VI 1 1 $ Temperature distribution for perturbed property 6
DIF6 = SUM("S6P0" T6,"S6NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3070
!S7P0 = 1.0/"DA7" $ Positive denominator used to numerically differentiate
!S7NE = -1.0/"DA7" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T7 = 2 PROP VII 1 1 $ Temperature distribution for perturbed property 7

```



```

DIF7 = SUM("S7P0" T7,"S7NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3080
!S8P0 = 1.0/"DA8"          $ Positive denominator used to numerically differentiate
!S8NE = -1.0/"DA8"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T8 = 2 PROP VIII 1 1 $ Temperature distribution for perturbed property 8
DIF8 = SUM("S8P0" T8,"S8NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3090
!S9P0 = 1.0/"DA9"          $ Positive denominator used to numerically differentiate
!S9NE = -1.0/"DA9"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T9 = 2 PROP IX 1 1 $ Temperature distribution for perturbed property 9
DIF9 = SUM("S9P0" T9,"S9NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3100
!SXPO = 1.0/"DA10"          $ Positive denominator used to numerically differentiate
!SXNE = -1.0/"DA10"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T10 = 2 PROP X 1 1 $ Temperature distribution for perturbed property 10
DIFX = SUM("SXPO" T10,"SXNE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 4000
!WORD = TOC,NJ(2 INIT 00 1 1) $ Determine the number of lines in INIT 00
$
*IF("NP" EQ 0) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 1) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 2) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 3) :*GOTO 6010 $ Calculate XTX, P11
*IF("NP" EQ 4) :*GOTO 6020 $ Calculate XTX, P11,P22
*IF("NP" EQ 5) :*GOTO 6030 $ Calculate XTX, P11,P22,P33
*IF("NP" EQ 6) :*GOTO 6040 $ Calculate XTX, P11,P22,P33,P44
*IF("NP" EQ 7) :*GOTO 6050 $ Calculate XTX, P11,P22,P33,P44,P55
*IF("NP" EQ 8) :*GOTO 6060 $ Calculate XTX, P11,P22,P33,P44,P55,P66
*IF("NP" EQ 9) :*GOTO 6070 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77
*IF("NP" EQ 10) :*GOTO 6080 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88
*IF("NP" EQ 11) :*GOTO 6090 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99
*IF("NP" EQ 12) :*GOTO 6100 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99,PXX
$
*LABEL 6010
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=2,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
TABLE(NI=1,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=0)
$
DEFINE S = 1 PMAT TOBE 1 1 $ Define table as register (X)
ST = RTRAN(S) $ XT
STS = RPROD(S,ST) $ XTX
STSI = RINV(STS) $ XTXI
!P11 = DS 1,1,1 (1 STSI AUS 1 1) $ numerical value of diagonal 1
*GOTO 5000

```

```

$
*LABEL 6020
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=3,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
TABLE(NI=2,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
$
DEFINE S = 1 PMAT TOBE 1 1
ST = RTRAN(S)
STS = RPROD(S,ST)
STSI = RINV(STS)
!P11 = DS 1,1,1 (1 STSI AUS 1 1)
!P22 = DS 2,2,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6030
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=4,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
TABLE(NI=3,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
$
DEFINE S = 1 PMAT TOBE 1 1
ST = RTRAN(S)
STS = RPROD(S,ST)
STSI = RINV(STS)
!P11 = DS 1,1,1 (1 STSI AUS 1 1)
!P22 = DS 2,2,1 (1 STSI AUS 1 1)
!P33 = DS 3,3,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6040
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=5,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
TABLE(NI=4,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)

```

```

TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6050
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=6,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
TABLE(NI=5,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6060
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=7,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)

```

```

TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
TABLE(NI=6,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6070
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=8,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
TABLE(NI=7,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)

```

```

!P66      = DS 6,6,1 (1 STSI AUS 1 1)
!P77      = DS 7,7,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6080
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=9,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
TABLE(NI=8,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6090
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1

```

```

DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=10,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
TABLE(NI=9,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
!P99    = DS 9,9,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6100
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1
DEFINE TR10 = 1 DIFX AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=11,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=10)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=10)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=10)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=10)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=10)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=10)

```

```

TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=10)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=10)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=10)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=10)
TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=10,DSKIP=10)
TABLE(NI=10,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1 ,1 ,1 (1 STSI AUS 1 1)
!P22    = DS 2 ,2 ,1 (1 STSI AUS 1 1)
!P33    = DS 3 ,3 ,1 (1 STSI AUS 1 1)
!P44    = DS 4 ,4 ,1 (1 STSI AUS 1 1)
!P55    = DS 5 ,5 ,1 (1 STSI AUS 1 1)
!P66    = DS 6 ,6 ,1 (1 STSI AUS 1 1)
!P77    = DS 7 ,7 ,1 (1 STSI AUS 1 1)
!P88    = DS 8 ,8 ,1 (1 STSI AUS 1 1)
!P99    = DS 9 ,9 ,1 (1 STSI AUS 1 1)
!PXX    = DS 10,10,1 (1 STSI AUS 1 1)
$
*GOTO 5000
*LABEL 5000
$
*RETURN
*COEF
$
$-----$
$ Subroutine(29 STAT ISTC)STAT: This subroutine does a statistical $
$                               analysis on the chromosomes, $
$                               including the calculation of $
$                               confidence intervals also called $
$                               CI's. $
$-----$
$
*(29 STAT ISTC)STAT
$
*IF("NP" EQ 0) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 1) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 2) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 3) :*GOTO 6010 $ Confidence intervals for properties 1
*IF("NP" EQ 4) :*GOTO 6020 $ Confidence intervals for properties 1,2
*IF("NP" EQ 5) :*GOTO 6030 $ Confidence intervals for properties 1,2,3
*IF("NP" EQ 6) :*GOTO 6040 $ Confidence intervals for properties 1,2,3,4
*IF("NP" EQ 7) :*GOTO 6050 $ Confidence intervals for properties 1,2,3,4,5
*IF("NP" EQ 8) :*GOTO 6060 $ Confidence intervals for properties 1,2,3,4,5,6
*IF("NP" EQ 9) :*GOTO 6070 $ Confidence intervals for properties 1,2,3,4,5,6,7
*IF("NP" EQ 10):*GOTO 6075 $ Confidence intervals for properties 1,2,3,4,5,6,7,8

```

```

*IF("NP" EQ 11):*GOTO 6080 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9
*IF("NP" EQ 12):*GOTO 6085 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9,10
$
*LABEL 6010
!CI1 = (P11**0.5)*1.960
TABLE(NI=1,NJ=1):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
*GOTO 7000
$
*LABEL 6020
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
TABLE(NI=1,NJ=2):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
*GOTO 7000
$
*LABEL 6030
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
TABLE(NI=1,NJ=3):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
*GOTO 7000
$
*LABEL 6040
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
TABLE(NI=1,NJ=4):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
*GOTO 7000
$
*LABEL 6050
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
TABLE(NI=1,NJ=5):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
*GOTO 7000
$
*LABEL 6060

```



```

!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
TABLE(NI=1,NJ=6):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  *GOTO 7000
$
*LABEL 6070
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
TABLE(NI=1,NJ=7):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  *GOTO 7000
$
*LABEL 6075
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
TABLE(NI=1,NJ=8):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  *GOTO 7000
$
*LABEL 6080
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960

```

```

!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
TABLE(NI=1,NJ=9):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  J=9:"CI9"
*GOTO 7000
$
*LABEL 6085
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
!CI10 = (PXX**0.5)*1.960
TABLE(NI=1,NJ=10):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  J=9:"CI9"
  J=10:"CI10"
*GOTO 7000
$
*LABEL 7000
*XQT DCU
  PRINT 1 CONF INTE 1 1
*XQT AUS
$
*RETURN
*STAT
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$

```

```

*XQT AUS                                $arithmetic utility system processor
$
$the following three comment statemnets can not be put at the end
$of the *TF commands, since it intervenes with the runstream.
$
$open the file containing the data
$read the datafile
$close the datafile
$
*TF OPEN 1'data
*TF READ 1
*TF CLOSE 1
$
!NP = 0
!N2 = 1                                $counter to direct TRAN TEMP
$
*DCALL (29 MAKE VARI)                  $call subroutine MAKE VARI
*DCALL (29 PROB SPEC)                  $call subroutine PROB SPEC
*DCALL (29 MATE PROP)                  $call subroutine MATE PROP
*DCALL (29 PLAT DIME)                  $call subroutine PLAT DIME
*DCALL (29 NODE GENE)                  $call subroutine NODE GENE
*DCALL (29 MAKE ELEM)                  $call subroutine MAKE ELEM
*DCALL (29 BOUN COND)                  $call subroutine BOUN COND
*DCALL (29 TRAN ANAL)                  $call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)                  $call subroutine TEMP TABL
$
!NP = 0
!N2 = 2                                $initial conditions
*DCALL (29 SENS COEF)                  $call subroutine SENS COEF
!NP = 0
!N2 = 3                                $parameter 1
*DCALL (29 SENS COEF)                  $call subroutine SENS COEF
!NP = 4
!N2 = 4                                $parameter 2
*DCALL (29 SENS COEF)                  $call subroutine SENS COEF
*DCALL (29 STAT ISTC)                  $call subroutine STAT ISTC
*XQT EXIT

```

Appendix F

EAL One-Dimensional Effective Convective Model: Forward Program

```
*XQT U1
*ONLINE=0
!MAXCM = SSP(0,13)
*CMPARA(MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of the $
$                               variables used in the program such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $ arithmetic utility system processor
$
!NY1  = 40          $ number of nodes for the effective structure
!NY2  = 10          $ number of nodes for the insulation
$
!EY1  ="NY1"-1      $ number of elements for the effective structure
!EY2  ="NY2"-1      $ number of elements for the insulation
$
!TOTN  = "NY1"+"EY2" $ total number of nodes of the model
!NON1  = "NY1"       $ node number of the node closest to the heater
!NONI  = "NY1"+"EY2" $ node number of the node on top of the insulation
!NONJ  = FLOAT("NONI") $ node number of the node on top of the insulation
$                  $ as a real
$
$
*RETURN
*VARI
$
```

```

$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the $
$                               properties of the materials used $
$                               in the model. $
$-----$
$
$(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for effective structure
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
                        J=1: 1.E+5 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
                        J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
                        J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$                               coordinates of the points used in $
$                               the geometry of the plate. The $
$                               points cited in the subroutine $
$                               describe the following $
$                               substructures: $
$                               1,2 :effective structure $
$                               2A,3 :insulation $
$-----$
$
$(29 PLAT DIME)DIME
$
!X1 = 0. $ x-coordinate for point 1
!X2 = 0. $ x-coordinate for point 2
!X3 = 0. $ x-coordinate for point 3
$
!Y1 = 0. $ y-coordinate for point 1
!Y2 = 0.1235 $ y-coordinate for point 2
!Y2A = 0.1235+(0.0254/"EY2") $ y-coordinate for point 2A
!Y3 = 0.1489 $ y-coordinate for point 3
$
!Z1 = 0. $ z-coordinate for point 1
!Z2 = 0. $ z-coordinate for point 2
!Z3 = 0. $ z-coordinate for point 3
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the coordinates $

```

```

$                                     of the nodes used in the mesh.      $
$-----$
$
$(29 NODE GENE)GENE
$
!INCY = 1          $nodal increment in the y-direction
!T2  = 1 + "NY1"   $starting node number for insulation
$
*XQT TAB           $defines position coordinates of nodes
$
START "TOTN"       $define the total number of nodes
JLOC              $define the joint locations
FORMAT = 1         $specifies rectangular coordinates
$
1,"X1","Y1","Z1","X2","Y2","Z2","NY1","INCY"
*IF("EY2" GT 1):*GOTO 100
"T2","X3","Y3","Z3"
*GOTO 105
*LABEL 100
"T2","X2","Y2A","Z2","X3","Y3","Z3","EY2","INCY"
*LABEL 105
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the      $
$                               elements used in the mesh.        $
$-----$
$
$(29 MAKE ELEM)ELEM
$
*XQT AUS           $ arithmetic utility system processor
$
TABLE(NI=1,NJ=1): K AREA : I = 1  $ section property table for
                        J=1: 1.0    $ K21 conduction elements
$
*XQT ELD           $ element definition processor
RESET NUTED = 1     $ destination library
$
!NOPT = 1
!CLOS = 0
$
!J1 = 1            $ group 1 lower outer facesheet
!J2 = "J1"+"INCY"  $ group 1 lower outer facesheet
!J3 = "J1"+"INCY"*"EY1" $ group 2 heater
!J4 = "J1"+"INCY"*"EY1" $ group 2 heater
!J5 = "J4"          $ group 3 insulation
!J6 = "J5"+"INCY"   $ group 3 insulation
$
K21                $ 1-D conduction elements
$
GROUP = 1           $ group 1 lower outer facesheet
NMAT = 1            $ material 1 conduction properties 1
"J1","J2","NOPT","EY1",1
$
GROUP = 2           $ group 2 heater
NMAT = 2            $ material 2 conduction properties 2
"J3","J4","NOPT",1,1

```

```

$
GROUP = 3                                $ group 3 insulation
NMAT = 3                                $ material 3 conduction properties 3
"J5","J6","NOPT","EY2",1
$
*XQT TGEO                                $ element geometry processor
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the      $
$                               boundary conditions such as:      $
$                               - the constant temperature        $
$                               boundary                          $
$                               - the applied heat flux boundary  $
$                               - the total experimental and      $
$                               heating time                      $
$                               - the heated node boundary        $
$                               temperature                       $
$-----$
$
*(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the
$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=3,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$

```

```

*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT HTMP
DEFINE EE = 5 DELT FLUX
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 2
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=EE,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$

```



```

$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
$
TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
  TRANSFER(SOUR=TB,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
TABLE(NI=1,NJ=2):5 TEMP NODE
  I=1
      J=1: 1.0
      J=2: "NONJ"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 2):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$ that are specific to the goal and $
$ purpose of the program. $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0 $ Starting time of TRTB analysis (seconds)
!INTS = 0.5 $ Integrta time step for TRTB (seconds)
$

```

```

!MIN1 = 1.0          $ lower bound for skins (PCp)
!MIN2 = 0.1          $ lower bound for skins (K)
!MIN3 = 10.0         $ lower bound for honeycomb (PCp)
!MIN4 = 10.0         $ lower bound for honeycomb (K)
!MIN5 = 10.0         $ lower bound for torque tubes (PCp)
!MIN6 = 10.0         $ lower bound for torque tubes (K)
!MIN7 = 10.0         $ lower bound for facesheets (PCp)
!MIN8 = 10.0         $ lower bound for facesheets (K)
!MIN9 = 10.0         $ lower bound for air gap (PCp)
!MINX = 10.0         $ lower bound for air gap (K)
!MAX1 = 5.0          $ upper bound for skins (PCp)
!MAX2 = 2.0          $ upper bound for skins (K)
!MAX3 = 10.0         $ upper bound for honeycomb (PCp)
!MAX4 = 10.0         $ upper bound for honeycomb (K)
!MAX5 = 10.0         $ upper bound for torque tubes (PCp)
!MAX6 = 10.0         $ upper bound for torque tubes (K)
!MAX7 = 10.0         $ upper bound for facesheets (PCp)
!MAX8 = 10.0         $ upper bound for facesheets (K)
!MAX9 = 10.0         $ upper bound for air gap (PCp)
!MAXX = 10.0         $ upper bound for air gap (K)
$
!A1  = 2.54          $ Initial value for skins (PCp)
!A2  = 1.00          $ Initial value for skins (K)
!A3  = 1.00          $ Initial value for honeycomb (PCp)
!A4  = 1.00          $ Initial value for honeycomb (K)
!A5  = 1.00          $ Initial value for torque tubes (PCp)
!A6  = 1.00          $ Initial value for torque tubes (K)
!A7  = 1.00          $ Initial value for facesheets (PCp)
!A8  = 1.00          $ Initial value for facesheets (K)
!A9  = 1.00          $ Initial value for air gap (PCp)
!A10 = 1.00          $ Initial value for air gap (K)
$
!A1I = 2.54          $ Initial value for skins (PCp)
!A2I = 1.00          $ Initial value for skins (K)
!A3I = 1.00          $ Initial value for honeycomb (PCp)
!A4I = 1.00          $ Initial value for honeycomb (K)
!A5I = 1.00          $ Initial value for torque tubes (PCp)
!A6I = 1.00          $ Initial value for torque tubes (K)
!A7I = 1.00          $ Initial value for facesheets (PCp)
!A8I = 1.00          $ Initial value for facesheets (K)
!A9I = 1.00          $ Initial value for air gap (PCp)
!A10I = 1.00         $ Initial value for air gap (K)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by $
$                               changing the Crank Nicholson para- $
$                               meter.                               $
$-----$
$
*(29 TRAN ANAL)ANAL
$
*XQT TRTB          $ implicit transient analyzer
  RESET QLIB = 5    $ thermal excitation library

```

```

RESET MLIB = 6                $ material and fluids properties library
RESET T1   = "TIMI"           $ starting time of analysis
RESET T2   = "TIMF"           $ termination time of analysis
RESET DT   = "INTS"           $ integration time step size
RESET DEST = 4                $ output library
RESET PRINT = 0               $ online print control parameter
RESET KPRT = 0                $ print control parameter for k matrix
RESET MXNDT = 10000000        $ maximum # of int. time steps during TRTB
RESET BETA = 0.5              $ Crank Nicholson parameter
RESET KTI   = 1                $ K matrix computation time parameter
KTIME      = 10000.           $ times to calculate the conduction load vector
TEMP       = "CSTP"           $ initial temperature of structure at T1
TSAVE      = 5 EXPE TIME 1 1   $ storage of nodal temperatures at time
$                               $ increments of TSAVE
$
*RETURN
*ANAL
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU                        $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1         $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1        $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                        $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME)  $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1               $ heated node number - 1
!VAR3 = "TOTN"-1               $ total node number - 1
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 1):*GOTO 5000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2):*GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3):*GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4):*GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5):*GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6):*GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7):*GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8):*GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9):*GOTO 5080 $ Temperature histories for property 7
$
*LABEL 5000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1

```

```

DEFINE C = 2 TRAN TIME 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
  TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
  TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
  TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
$*XQT DCU
$ PRINT 5 TEMP ERAT 1 1
$*XQT AUS
$
*GOTO 6000
$
*LABEL 5010
TABLE(NI=1,NJ="VAR1"):2 INIT OO 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ Subroutine (29 SENS COEF)COEF: This subroutine calculates $

```

```

$                                the sensitivity coefficients $
$                                for up to seven parameters.  $
$-----$
$
$(29 SENS COEF)COEF
$
*XQT AUS                        $ Arithmetic Utility System
$
INLIB = 1                       $ Read from library 1
OUTLI = 1                       $ Write to library 1
$
*IF("N2" EQ 0) :*GOTO 1000      $ No perturbations
*IF("N2" EQ 1) :*GOTO 1000      $ No perturbations
*IF("N2" EQ 2) :*GOTO 1000      $ No perturbations
*IF("N2" EQ 3) :*GOTO 1010      $ Property 1 is perturbed
*IF("N2" EQ 4) :*GOTO 1020      $ Property 2 is perturbed
*IF("N2" EQ 5) :*GOTO 1030      $ Property 3 is perturbed
*IF("N2" EQ 6) :*GOTO 1040      $ Property 4 is perturbed
*IF("N2" EQ 7) :*GOTO 1050      $ Property 5 is perturbed
*IF("N2" EQ 8) :*GOTO 1060      $ Property 6 is perturbed
*IF("N2" EQ 9) :*GOTO 1070      $ Property 7 is perturbed
*IF("N2" EQ 10) :*GOTO 1080     $ Property 8 is perturbed
*IF("N2" EQ 11) :*GOTO 1090     $ Property 9 is perturbed
*IF("N2" EQ 12) :*GOTO 1100     $ Property 10 is perturbed
*GOTO 2000
$
*LABEL 1000
$
!A1 = "A1I"                     $ Set property 1 to initial guess
!A2 = "A2I"                     $ Set property 2 to initial guess
!A3 = "A3I"                     $ Set property 3 to initial guess
!A4 = "A4I"                     $ Set property 4 to initial guess
!A5 = "A5I"                     $ Set property 5 to initial guess
!A6 = "A6I"                     $ Set property 6 to initial guess
!A7 = "A7I"                     $ Set property 7 to initial guess
!A8 = "A8I"                     $ Set property 8 to initial guess
!A9 = "A9I"                     $ Set property 9 to initial guess
!A10 = "A10I"                   $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)            $ Update the thermal properties
*DCALL(29 TRAN ANAL)           $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)           $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1010
$
!A1 = 1.001*"A1I"              $ Perturb property 1
!DA1 = 0.001*"A1I"             $ Amount property 1 is perturbed
!A2 = "A2I"                     $ Set property 2 to initial guess
!A3 = "A3I"                     $ Set property 3 to initial guess
!A4 = "A4I"                     $ Set property 4 to initial guess
!A5 = "A5I"                     $ Set property 5 to initial guess
!A6 = "A6I"                     $ Set property 6 to initial guess
!A7 = "A7I"                     $ Set property 7 to initial guess
!A8 = "A8I"                     $ Set property 8 to initial guess
!A9 = "A9I"                     $ Set property 9 to initial guess
!A10 = "A10I"                   $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)            $ Update the thermal properties

```

*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1020	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = 1.001*"A2I"	\$ Perturb property 2
!DA2 = 0.001*"A2I"	\$ Amount property 2 is perturbed
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1030	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = 1.001*"A3I"	\$ Perturb property 3
!DA3 = 0.001*"A3I"	\$ Amount property 3 is perturbed
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1040	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = 1.001*"A4I"	\$ Perturb property 4
!DA4 = 0.001*"A4I"	\$ Amount property 4 is perturbed
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures

```

$
*GOTO 2000
*LABEL 1050
$
!A1 = "A1I"           $ Set property 1  to initial guess
!A2 = "A2I"           $ Set property 2  to initial guess
!A3 = "A3I"           $ Set property 3  to initial guess
!A4 = "A4I"           $ Set property 4  to initial guess
!A5 = 1.001*"A5I"     $ Perturb property 5
!DA5 = 0.001*"A5I"    $ Amount property 5 is perturbed
!A6 = "A6I"           $ Set property 6  to initial guess
!A7 = "A7I"           $ Set property 7  to initial guess
!A8 = "A8I"           $ Set property 8  to initial guess
!A9 = "A9I"           $ Set property 9  to initial guess
!A10 = "A10I"         $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1060
$
!A1 = "A1I"           $ Set property 1  to initial guess
!A2 = "A2I"           $ Set property 2  to initial guess
!A3 = "A3I"           $ Set property 3  to initial guess
!A4 = "A4I"           $ Set property 4  to initial guess
!A5 = "A5I"           $ Set property 5  to initial guess
!A6 = 1.001*"A6I"     $ Perturb property 6
!DA6 = 0.001*"A6I"    $ Amount property 6 is perturbed
!A7 = "A7I"           $ Set property 7  to initial guess
!A8 = "A8I"           $ Set property 8  to initial guess
!A9 = "A9I"           $ Set property 9  to initial guess
!A10 = "A10I"         $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1070
$
!A1 = "A1I"           $ Set property 1  to initial guess
!A2 = "A2I"           $ Set property 2  to initial guess
!A3 = "A3I"           $ Set property 3  to initial guess
!A4 = "A4I"           $ Set property 4  to initial guess
!A5 = "A5I"           $ Set property 5  to initial guess
!A6 = "A6I"           $ Set property 6  to initial guess
!A7 = 1.001*"A7I"     $ Perturb property 7
!DA7 = 0.001*"A7I"    $ Amount property 7 is perturbed
!A8 = "A8I"           $ Set property 8  to initial guess
!A9 = "A9I"           $ Set property 9  to initial guess
!A10 = "A10I"         $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000

```

```

*LABEL 1080
$
!A1  = "A1I"          $ Set property 1  to initial guess
!A2  = "A2I"          $ Set property 2  to initial guess
!A3  = "A3I"          $ Set property 3  to initial guess
!A4  = "A4I"          $ Set property 4  to initial guess
!A5  = "A5I"          $ Set property 5  to initial guess
!A6  = "A6I"          $ Set property 6  to initial guess
!A7  = "A7I"          $ Set property 7  to initial guess
!A8  = 1.001*"A8I"     $ Perturb property 8
!DA8 = 0.001*"A8I"     $ Amount property 8 is perturbed
!A9  = "A9I"          $ Set property 9  to initial guess
!A10 = "A10I"         $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1090
$
!A1  = "A1I"          $ Set property 1  to initial guess
!A2  = "A2I"          $ Set property 2  to initial guess
!A3  = "A3I"          $ Set property 3  to initial guess
!A4  = "A4I"          $ Set property 4  to initial guess
!A5  = "A5I"          $ Set property 5  to initial guess
!A6  = "A6I"          $ Set property 6  to initial guess
!A7  = "A7I"          $ Set property 7  to initial guess
!A8  = "A8I"          $ Set property 8  to initial guess
!A9  = 1.001*"A9I"     $ Perturb property 9
!DA9 = 0.001*"A9I"     $ Amount property 9 is perturbed
!A10 = "A10I"         $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1100
$
!A1  = "A1I"          $ Set property 1  to initial guess
!A2  = "A2I"          $ Set property 2  to initial guess
!A3  = "A3I"          $ Set property 3  to initial guess
!A4  = "A4I"          $ Set property 4  to initial guess
!A5  = "A5I"          $ Set property 5  to initial guess
!A6  = "A6I"          $ Set property 6  to initial guess
!A7  = "A7I"          $ Set property 7  to initial guess
!A8  = "A8I"          $ Set property 8  to initial guess
!A9  = "A9I"          $ Set property 9  to initial guess
!A10 = 1.001*"A10I"    $ Perturb property 10
!DA10 = 0.001*"A10I"  $ Amount property 10 is perturbed
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 2000
$

```



```

*IF("N2" EQ 0) : *GOTO 3000      $ No sensitivity coefficients are calculated
*IF("N2" EQ 1) : *GOTO 3000      $ No sensitivity coefficients are calculated
*IF("N2" EQ 2) : *GOTO 3000      $ No sensitivity coefficients are calculated
*IF("N2" EQ 3) : *GOTO 3010      $ Sensitivity coefficients for property 1
*IF("N2" EQ 4) : *GOTO 3020      $ Sensitivity coefficients for property 1,2
*IF("N2" EQ 5) : *GOTO 3030      $ Sensitivity coefficients for property 1,2,3
*IF("N2" EQ 6) : *GOTO 3040      $ Sensitivity coefficients for property 1,2,3,4
*IF("N2" EQ 7) : *GOTO 3050      $ Sensitivity coefficients for property 1,2,3,4,5
*IF("N2" EQ 8) : *GOTO 3060      $ Sensitivity coefficients for property 1,2,3,4,5,6
*IF("N2" EQ 9) : *GOTO 3070      $ Sensitivity coefficients for property 1,2,3,4,5,6,7
*IF("N2" EQ 10) : *GOTO 3080     $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8
*IF("N2" EQ 11) : *GOTO 3090     $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9
*IF("N2" EQ 12) : *GOTO 3100     $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9,10
*GOTO 3000
$
*LABEL 3000
DEFINE TO = 2 INIT OO 1 1 $ Initial temperature distribution with no perturbations
*GOTO 4000
$
*LABEL 3010
!S1PO = 1.0/"DA1"          $ Positive denominator used to numerically differentiate
!S1NE = -1.0/"DA1"         $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT OO 1 1 $ Temperature distribution for no perturbations
DEFINE T1 = 2 PROP I 1 1 $ Temperature distribution for perturbed property 1
DIF1 = SUM("S1PO" T1,"S1NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3020
!S2PO = 1.0/"DA2"          $ Positive denominator used to numerically differentiate
!S2NE = -1.0/"DA2"         $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT OO 1 1 $ Temperature distribution for no perturbations
DEFINE T2 = 2 PROP II 1 1 $ Temperature distribution for perturbed property 2
DIF2 = SUM("S2PO" T2,"S2NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3030
!S3PO = 1.0/"DA3"          $ Positive denominator used to numerically differentiate
!S3NE = -1.0/"DA3"         $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT OO 1 1 $ Temperature distribution for no perturbations
DEFINE T3 = 2 PROP III 1 1 $ Temperature distribution for perturbed property 3
DIF3 = SUM("S3PO" T3,"S3NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3040
!S4PO = 1.0/"DA4"          $ Positive denominator used to numerically differentiate
!S4NE = -1.0/"DA4"         $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT OO 1 1 $ Temperature distribution for no perturbations
DEFINE T4 = 2 PROP IV 1 1 $ Temperature distribution for perturbed property 4
DIF4 = SUM("S4PO" T4,"S4NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3050
!S5PO = 1.0/"DA5"          $ Positive denominator used to numerically differentiate
!S5NE = -1.0/"DA5"         $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT OO 1 1 $ Temperature distribution for no perturbations
DEFINE T5 = 2 PROP V 1 1 $ Temperature distribution for perturbed property 5
DIF5 = SUM("S5PO" T5,"S5NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$

```

```

*LABEL 3060
!S6P0 = 1.0/"DA6"          $ Positive denominator used to numerically differentiate
!S6NE = -1.0/"DA6"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00    1 1 $ Temperature distribution for no perturbations
DEFINE T6 = 2 PROP VI    1 1 $ Temperature distribution for perturbed property 6
DIF6 = SUM("S6P0" T6,"S6NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3070
!S7P0 = 1.0/"DA7"          $ Positive denominator used to numerically differentiate
!S7NE = -1.0/"DA7"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00    1 1 $ Temperature distribution for no perturbations
DEFINE T7 = 2 PROP VII   1 1 $ Temperature distribution for perturbed property 7
DIF7 = SUM("S7P0" T7,"S7NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3080
!S8P0 = 1.0/"DA8"          $ Positive denominator used to numerically differentiate
!S8NE = -1.0/"DA8"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00    1 1 $ Temperature distribution for no perturbations
DEFINE T8 = 2 PROP VIII  1 1 $ Temperature distribution for perturbed property 8
DIF8 = SUM("S8P0" T8,"S8NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3090
!S9P0 = 1.0/"DA9"          $ Positive denominator used to numerically differentiate
!S9NE = -1.0/"DA9"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00    1 1 $ Temperature distribution for no perturbations
DEFINE T9 = 2 PROP IX    1 1 $ Temperature distribution for perturbed property 9
DIF9 = SUM("S9P0" T9,"S9NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3100
!SXPO = 1.0/"DA10"         $ Positive denominator used to numerically differentiate
!SXNE = -1.0/"DA10"        $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00    1 1 $ Temperature distribution for no perturbations
DEFINE T10 = 2 PROP X     1 1 $ Temperature distribution for perturbed property 10
DIFX = SUM("SXPO" T10,"SXNE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 4000
!WORD = TOC,NJ(2 INIT 00 1 1) $ Determine the number of lines in INIT 00
$
*IF("NP" EQ 0) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 1) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 2) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 3) :*GOTO 6010 $ Calculate XTX, P11
*IF("NP" EQ 4) :*GOTO 6020 $ Calculate XTX, P11,P22
*IF("NP" EQ 5) :*GOTO 6030 $ Calculate XTX, P11,P22,P33
*IF("NP" EQ 6) :*GOTO 6040 $ Calculate XTX, P11,P22,P33,P44
*IF("NP" EQ 7) :*GOTO 6050 $ Calculate XTX, P11,P22,P33,P44,P55
*IF("NP" EQ 8) :*GOTO 6060 $ Calculate XTX, P11,P22,P33,P44,P55,P66
*IF("NP" EQ 9) :*GOTO 6070 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77
*IF("NP" EQ 10):*GOTO 6080 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88
*IF("NP" EQ 11):*GOTO 6090 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99
*IF("NP" EQ 12):*GOTO 6100 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99,PXX
$
*LABEL 6010
DEFINE TR1 = 1 DIF1 AUS 1 1

```

```

DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=2,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
TABLE(NI=1,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=0)
$
DEFINE S = 1 PMAT TOBE 1 1          $ Define table as register (X)
ST      = RTRAN(S)                  $ XT
STS     = RPROD(S,ST)               $ XTX
STSI    = RINV(STS)                 $ XTXI
!P11    = DS 1,1,1 (1 STSI AUS 1 1) $ numerical value of diagonal 1
*GOTO 5000
$
*LABEL 6020
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=3,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
TABLE(NI=2,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6030
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=4,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
TABLE(NI=3,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6040

```

```

DEFINE TR1  = 1 DIF1 AUS  1 1
DEFINE TR2  = 1 DIF2 AUS  1 1
DEFINE TR3  = 1 DIF3 AUS  1 1
DEFINE TR4  = 1 DIF4 AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=5,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
TABLE(NI=4,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
$
DEFINE S = 1 PMAT TOBE 1 1
ST       = RTRAN(S)
STS      = RPROD(S,ST)
STSI     = RINV(STS)
!P11     = DS 1,1,1 (1 STSI AUS 1 1)
!P22     = DS 2,2,1 (1 STSI AUS 1 1)
!P33     = DS 3,3,1 (1 STSI AUS 1 1)
!P44     = DS 4,4,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6050
DEFINE TR1  = 1 DIF1 AUS  1 1
DEFINE TR2  = 1 DIF2 AUS  1 1
DEFINE TR3  = 1 DIF3 AUS  1 1
DEFINE TR4  = 1 DIF4 AUS  1 1
DEFINE TR5  = 1 DIF5 AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=6,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
TABLE(NI=5,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
$
DEFINE S = 1 PMAT TOBE 1 1
ST       = RTRAN(S)
STS      = RPROD(S,ST)
STSI     = RINV(STS)
!P11     = DS 1,1,1 (1 STSI AUS 1 1)
!P22     = DS 2,2,1 (1 STSI AUS 1 1)
!P33     = DS 3,3,1 (1 STSI AUS 1 1)
!P44     = DS 4,4,1 (1 STSI AUS 1 1)
!P55     = DS 5,5,1 (1 STSI AUS 1 1)
*GOTO 5000
$

```

```

*LABEL 6060
DEFINE TR1  = 1 DIF1 AUS  1 1
DEFINE TR2  = 1 DIF2 AUS  1 1
DEFINE TR3  = 1 DIF3 AUS  1 1
DEFINE TR4  = 1 DIF4 AUS  1 1
DEFINE TR5  = 1 DIF5 AUS  1 1
DEFINE TR6  = 1 DIF6 AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=7,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
TABLE(NI=6,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
$
DEFINE S = 1 PMAT TOBE 1 1
ST       = RTRAN(S)
STS      = RPROD(S,ST)
STSI     = RINV(STS)
!P11     = DS 1,1,1 (1 STSI AUS 1 1)
!P22     = DS 2,2,1 (1 STSI AUS 1 1)
!P33     = DS 3,3,1 (1 STSI AUS 1 1)
!P44     = DS 4,4,1 (1 STSI AUS 1 1)
!P55     = DS 5,5,1 (1 STSI AUS 1 1)
!P66     = DS 6,6,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6070
DEFINE TR1  = 1 DIF1 AUS  1 1
DEFINE TR2  = 1 DIF2 AUS  1 1
DEFINE TR3  = 1 DIF3 AUS  1 1
DEFINE TR4  = 1 DIF4 AUS  1 1
DEFINE TR5  = 1 DIF5 AUS  1 1
DEFINE TR6  = 1 DIF6 AUS  1 1
DEFINE TR7  = 1 DIF7 AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=8,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
TABLE(NI=7,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)

```

```

TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6080
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=9,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
TABLE(NI=8,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)

```

```

!P88      = DS 8,8,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6090
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=10,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
TABLE(NI=9,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
!P99    = DS 9,9,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6100
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1

```

```

DEFINE TR7  = 1 DIF7 AUS  1 1
DEFINE TR8  = 1 DIF8 AUS  1 1
DEFINE TR9  = 1 DIF9 AUS  1 1
DEFINE TR10 = 1 DIFX AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=11,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=10)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=10)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=10)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=10)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=10)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=10)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=10)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=10)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=10)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=10)
  TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=10,DSKIP=10)
TABLE(NI=10,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
  TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1 ,1 ,1 (1 STSI AUS 1 1)
!P22    = DS 2 ,2 ,1 (1 STSI AUS 1 1)
!P33    = DS 3 ,3 ,1 (1 STSI AUS 1 1)
!P44    = DS 4 ,4 ,1 (1 STSI AUS 1 1)
!P55    = DS 5 ,5 ,1 (1 STSI AUS 1 1)
!P66    = DS 6 ,6 ,1 (1 STSI AUS 1 1)
!P77    = DS 7 ,7 ,1 (1 STSI AUS 1 1)
!P88    = DS 8 ,8 ,1 (1 STSI AUS 1 1)
!P99    = DS 9 ,9 ,1 (1 STSI AUS 1 1)
!PXX    = DS 10,10,1 (1 STSI AUS 1 1)
$
*GOTO 5000
*LABEL 5000
$
*RETURN
*COEF
$
$-----$
$ Subroutine(29 STAT ISTC)STAT: This subroutine does a statistical $
$                               analysis on the chromosomes, $
$                               including the calculation of $
$                               confidence intervals also called $
$                               CI's. $
$-----$
$

```



```

*(29 STAT ISTD)STAT
$
*IF("NP" EQ 0) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 1) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 2) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 3) :*GOTO 6010 $ Confidence intervals for properties 1
*IF("NP" EQ 4) :*GOTO 6020 $ Confidence intervals for properties 1,2
*IF("NP" EQ 5) :*GOTO 6030 $ Confidence intervals for properties 1,2,3
*IF("NP" EQ 6) :*GOTO 6040 $ Confidence intervals for properties 1,2,3,4
*IF("NP" EQ 7) :*GOTO 6050 $ Confidence intervals for properties 1,2,3,4,5
*IF("NP" EQ 8) :*GOTO 6060 $ Confidence intervals for properties 1,2,3,4,5,6
*IF("NP" EQ 9) :*GOTO 6070 $ Confidence intervals for properties 1,2,3,4,5,6,7
*IF("NP" EQ 10):*GOTO 6075 $ Confidence intervals for properties 1,2,3,4,5,6,7,8
*IF("NP" EQ 11):*GOTO 6080 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9
*IF("NP" EQ 12):*GOTO 6085 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9,10
$
*LABEL 6010
!CI1 = (P11**0.5)*1.960
TABLE(NI=1,NJ=1):1 CONF INTE 1 1
I=1
J=1:"CI1"
*GOTO 7000
$
*LABEL 6020
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
TABLE(NI=1,NJ=2):1 CONF INTE 1 1
I=1
J=1:"CI1"
J=2:"CI2"
*GOTO 7000
$
*LABEL 6030
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
TABLE(NI=1,NJ=3):1 CONF INTE 1 1
I=1
J=1:"CI1"
J=2:"CI2"
J=3:"CI3"
*GOTO 7000
$
*LABEL 6040
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
TABLE(NI=1,NJ=4):1 CONF INTE 1 1
I=1
J=1:"CI1"
J=2:"CI2"
J=3:"CI3"
J=4:"CI4"
*GOTO 7000
$
*LABEL 6050
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960

```

```

!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
TABLE(NI=1,NJ=5):1 CONF INTE 1 1
  I=1
    J=1:"CI1"
    J=2:"CI2"
    J=3:"CI3"
    J=4:"CI4"
    J=5:"CI5"
  *GOTO 7000
$
*LABEL 6060
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
TABLE(NI=1,NJ=6):1 CONF INTE 1 1
  I=1
    J=1:"CI1"
    J=2:"CI2"
    J=3:"CI3"
    J=4:"CI4"
    J=5:"CI5"
    J=6:"CI6"
  *GOTO 7000
$
*LABEL 6070
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
TABLE(NI=1,NJ=7):1 CONF INTE 1 1
  I=1
    J=1:"CI1"
    J=2:"CI2"
    J=3:"CI3"
    J=4:"CI4"
    J=5:"CI5"
    J=6:"CI6"
    J=7:"CI7"
  *GOTO 7000
$
*LABEL 6075
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
TABLE(NI=1,NJ=8):1 CONF INTE 1 1
  I=1

```

```

J=1:"CI1"
J=2:"CI2"
J=3:"CI3"
J=4:"CI4"
J=5:"CI5"
J=6:"CI6"
J=7:"CI7"
J=8:"CI8"
*GOTO 7000
$
*LABEL 6080
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
TABLE(NI=1,NJ=9):1 CONF INTE 1 1
I=1
J=1:"CI1"
J=2:"CI2"
J=3:"CI3"
J=4:"CI4"
J=5:"CI5"
J=6:"CI6"
J=7:"CI7"
J=8:"CI8"
J=9:"CI9"
*GOTO 7000
$
*LABEL 6085
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
!CI10 = (PXX**0.5)*1.960
TABLE(NI=1,NJ=10):1 CONF INTE 1 1
I=1
J=1 : "CI1"
J=2 : "CI2"
J=3 : "CI3"
J=4 : "CI4"
J=5 : "CI5"
J=6 : "CI6"
J=7 : "CI7"
J=8 : "CI8"
J=9 : "CI9"
J=10:"CI10"
*GOTO 7000
$
*LABEL 7000

```

```

*XQT DCU
  PRINT 1 CONF INTE 1 1
*XQT AUS
$
*RETURN
*STAT
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS                      $arithmetic utility system processor
$
$the following three comment statemnets can not be put at the end
$of the *TF commands, since it intervenes with the runstream.
$
$open the file containing the data
$read the datafile
$close the datafile
$
*TF OPEN  1'data
*TF READ  1
*TF CLOSE 1
$
!NP = 0
!N2 = 1                      $counter to direct TRAN TEMP
$
*DCALL (29 MAKE VARI)        $call subroutine MAKE VARI
*DCALL (29 PROB SPEC)        $call subroutine PROB SPEC
*DCALL (29 MATE PROP)        $call subroutine MATE PROP
*DCALL (29 PLAT DIME)        $call subroutine PLAT DIME
*DCALL (29 NODE GENE)        $call subroutine NODE GENE
*DCALL (29 MAKE ELEM)        $call subroutine MAKE ELEM
*DCALL (29 BOUN COND)        $call subroutine BOUN COND
*DCALL (29 TRAN ANAL)        $call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)        $call subroutine TEMP TABL
$
!NP = 0
!N2 = 2                      $initial conditions
*DCALL (29 SENS COEF)        $call subroutine SENS COEF
!NP = 0
!N2 = 3                      $parameter 1
*DCALL (29 SENS COEF)        $call subroutine SENS COEF
!NP = 4
!N2 = 4                      $parameter 2
*DCALL (29 SENS COEF)        $call subroutine SENS COEF
*DCALL (29 STAT ISTC)        $call subroutine STAT ISTC
*XQT EXIT

```

Appendix G

EAL One-Dimensional Detailed Model: Forward Program

```
*XQT U1
!MAXCM = SSP(0,13)
*CMPARA (MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of $
$                               the variables used in the program $
$                               such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each $
$                               section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS          $ arithmetic utility system processor
$
!NY1 = 8           $ number of nodes for the lower outer facesheet
!NY2 = 36          $ number of nodes for the lower honeycomb structure
!NY3 = 12          $ number of nodes for the lower inner facesheet
!NY4 = 8           $ number of nodes for the lower torque tube
!NY5 = 68          $ number of nodes for the air cavity
!NY6 = 8           $ number of nodes for the upper torque tube
!NY7 = 12          $ number of nodes for the upper inner facesheet
!NY8 = 36          $ number of nodes for the upper honeycomb structure
!NY9 = 8           $ number of nodes for the upper outer facesheet
!NY10 = 36         $ number of nodes for the insulation
$
!EY1 ="NY1" -1     $ number of elements for the lower outer facesheet
!EY2 ="NY2" -1     $ number of elements for the lower honeycomb structure
!EY3 ="NY3" -1     $ number of elements for the lower inner facesheet
!EY4 ="NY4" -1     $ number of elements for the lower torque tube
!EY5 ="NY5" -1     $ number of elements for the air cavity
!EY6 ="NY6" -1     $ number of elements for the upper torque tube
!EY7 ="NY7" -1     $ number of elements for the upper inner facesheet
```

```

!EY8 ="NY8" -1      $ number of elements for the upper honeycomb structure
!EY9 ="NY9" -1      $ number of elements for the upper outer facesheet
!EY10 ="NY10"-1     $ number of elements for the insulation
$
!TOTN = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"+"EY10"
$                  $ total number of nodes in the model
!NON1 = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"
$                  $ node number at the heater (high sensitivity)
!NON2 = "NY1"+"EY2"+"EY3"+"EY4"
$                  $ node number at the inside of the lower torque tube
!NON3 = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"
$                  $ node number at the inside of the upper torque tube
!NONJ = FLOAT("TOTN")
$                  $ node number on top of the insulation
$
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the      $
$                               properties of the materials used    $
$                               in the model.                        $
$-----$
$
*(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for IM7/954-2A
$ upper and lower skins
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
                      J=1: 1.E+6 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
                      J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
                      J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
$ conduction properties for air
$
TABLE(NI=9,NJ=1):6 COND PROP 4:I=2 3 4
                      J=1: 1.E+6 "A9" "A10"
$
$                      J=1: 0.9950 1009.0 0.7025
$
$TABLE(NI=9,NJ=14):6 COND PROP 4:I=1 2 3 4 5 6
$      J= 1: 250.0 1.3947 1006. 0.1758 0.1758 0.1758
$      J= 2: 300.0 1.1614 1007. 0.2916 0.2916 0.2916
$      J= 3: 350.0 0.9950 1009. 0.4513 0.4513 0.4513
$      J= 4: 400.0 0.8711 1014. 0.6627 0.6627 0.6627
$      J= 5: 450.0 0.7740 1021. 0.9327 0.9327 0.9327
$      J= 6: 500.0 0.6964 1030. 1.2690 1.2690 1.2690
$      J= 7: 550.0 0.6329 1040. 1.6788 1.6788 1.6788
$      J= 8: 600.0 0.5804 1051. 2.1694 2.1694 2.1694

```

```

$          J= 9: 650.0 0.5356 1063. 2.7483 2.7483 2.7483
$          J=10: 700.0 0.4975 1075. 3.4229 3.4229 3.4229
$          J=11: 750.0 0.4643 1087. 4.2005 4.2005 4.2005
$          J=12: 800.0 0.4354 1099. 5.0885 5.0885 5.0885
$          J=13: 850.0 0.4097 1110. 6.0943 6.0943 6.0943
$          J=14: 900.0 0.3868 1121. 7.2256 7.2256 7.2256
$
$ conduction properties for SS321 honeycomb (actually SS347)
$
TABLE(NI=9,NJ=1):6 COND PROP 5: I = 2 3 4
          J=1: 1.E+6 "A3" "A4"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tubes, edge channels
$
TABLE(NI=9,NJ=1):6 COND PROP 6: I= 2 3 4
          J=1: 1.E+6 "A5" "A6"
$
$ conduction properties for PETI-5 of the upper and lower outer
$ facesheets.
$
TABLE(NI=9,NJ=1):6 COND PROP 7: I= 2 3 4
          J=1: 1.E+6 "A7" "A8"
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$                               coordinates of the points used in $
$                               the geometry of the plate. The $
$                               points cited in the following $
$                               subroutine correspond to these $
$                               substructures: $
$                               1,2   :lower outer facesheet $
$                               2A,3  :lower honeycomb $
$                               3A,4   :lower inner facesheet $
$                               4A,5   :lower torque tube $
$                               5A,6   :air cavity $
$                               6A,7   :upper torque tube $
$                               7A,8   :upper inner facesheet $
$                               8A,9   :upper honeycomb $
$                               9A,10  :upper outer facesheet $
$                               10A,11 :insulation $
$-----$
$
*(29 PLAT DIME)DIME
$
!X1 = 0.          $ x-coordinate for point 1
!X2 = 0.          $ x-coordinate for point 2
!X3 = 0.          $ x-coordinate for point 3
!X4 = 0.          $ x-coordinate for point 4
!X5 = 0.          $ x-coordinate for point 5
!X6 = 0.          $ x-coordinate for point 6
!X7 = 0.          $ x-coordinate for point 7
!X8 = 0.          $ x-coordinate for point 8
!X9 = 0.          $ x-coordinate for point 9
!X10 = 0.         $ x-coordinate for point 10
!X11 = 0.         $ x-coordinate for point 11

```

```

$
!Y1 = 0.          $ y-coordinate for point 1
!Y2 = 0.0032      $ y-coordinate for point 2
!Y2A = 0.0032+(0.0254/"EY2") $ y-coordinate for point 2A
!Y3 = 0.0286      $ y-coordinate for point 3
!Y3A = 0.0286+(0.0061/"EY3") $ y-coordinate for point 3A
!Y4 = 0.0347      $ y-coordinate for point 4
!Y4A = 0.0347+(0.0021/"EY4") $ y-coordinate for point 4A
!Y5 = 0.0368      $ y-coordinate for point 5
!Y5A = 0.0368+(0.0510/"EY5") $ y-coordinate for point 5A
!Y6 = 0.0878      $ y-coordinate for point 6
!Y6A = 0.0878+(0.0021/"EY6") $ y-coordinate for point 6A
!Y7 = 0.0899      $ y-coordinate for point 7
!Y7A = 0.0899+(0.0050/"EY7") $ y-coordinate for point 7A
!Y8 = 0.0949      $ y-coordinate for point 8
!Y8A = 0.0949+(0.0254/"EY8") $ y-coordinate for point 8A
!Y9 = 0.1203      $ y-coordinate for point 9
!Y9A = 0.1203+(0.0032/"EY9") $ y-coordinate for point 9A
!Y10 = 0.1235     $ y-coordinate for point 10
!Y10A = 0.1235+(0.0254/"EY10") $ y-coordinate for point 10A
!Y11 = 0.1489     $ y-coordinate for point 11
$
!Z1 = 0.          $ z-coordinate for point 1
!Z2 = 0.          $ z-coordinate for point 2
!Z3 = 0.          $ z-coordinate for point 3
!Z4 = 0.          $ z-coordinate for point 4
!Z5 = 0.          $ z-coordinate for point 5
!Z6 = 0.          $ z-coordinate for point 6
!Z7 = 0.          $ z-coordinate for point 7
!Z8 = 0.          $ z-coordinate for point 8
!Z9 = 0.          $ z-coordinate for point 9
!Z10 = 0.         $ z-coordinate for point 10
!Z11 = 0.         $ z-coordinate for point 11
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the $
$                               coordinates of the nodes used in $
$                               the mesh. $
$-----$
$
*(29 NODE GENE)GENE
$
!INCY = 1          $ nodal increment in the y-direction
!T2 = 1 + "NY1"    $ starting node number for lower honeycomb
!T3 = "T2"+"EY2"   $ starting node number for lower inner facesheet
!T4 = "T3"+"EY3"   $ starting node number for lower torque tube edge
!T5 = "T4"+"EY4"   $ starting node number for cavity
!T6 = "T5"+"EY5"   $ starting node number for upper torque tube edge
!T7 = "T6"+"EY6"   $ starting node number for upper inner facesheet
!T8 = "T7"+"EY7"   $ starting node number for upper honeycomb
!T9 = "T8"+"EY8"   $ starting node number for upper outer facesheet
!T10 = "T9"+"EY9"  $ starting node number for insulation
$
*XQT TAB          $ table generator, structural properties
$
START "TOTN"      $ memory allocation for all the nodes

```



```

JLOC          $ initiation of the creation of the dataset containing
$             $ the joint position coordinates
FORMAT = 1    $ specifies rectangular coordinates
$
1,"X1","Y1","Z1","X2","Y2","Z2","NY1","INCY"
*IF("EY2" GT 1):*GOTO 100
"T2","X3","Y3","Z3"
*GOTO 105
$
*LABEL 100
"T2","X2","Y2A","Z2","X3","Y3","Z3","EY2","INCY"
*LABEL 105
*IF("EY3" GT 1):*GOTO 110
"T3","X4","Y4","Z4"
*GOTO 115
$
*LABEL 110
"T3","X3","Y3A","Z3","X4","Y4","Z4","EY3","INCY"
*LABEL 115
*IF("EY4" GT 1):*GOTO 120
"T4","X5","Y5","Z5"
*GOTO 125
$
*LABEL 120
"T4","X4","Y4A","Z4","X5","Y5","Z5","EY4","INCY"
*LABEL 125
*IF("EY5" GT 1):*GOTO 130
"T5","X6","Y6","Z6"
*GOTO 135
$
*LABEL 130
"T5","X5","Y5A","Z5","X6","Y6","Z6","EY5","INCY"
*LABEL 135
*IF("EY6" GT 1):*GOTO 140
"T6","X7","Y7","Z7"
*GOTO 145
$
*LABEL 140
"T6","X6","Y6A","Z6","X7","Y7","Z7","EY6","INCY"
*LABEL 145
*IF("EY7" GT 1):*GOTO 150
"T7","X8","Y8","Z8"
*GOTO 155
$
*LABEL 150
"T7","X7","Y7A","Z7","X8","Y8","Z8","EY7","INCY"
*LABEL 155
*IF("EY8" GT 1):*GOTO 160
"T8","X9","Y9","Z9"
*GOTO 165
$
*LABEL 160
"T8","X8","Y8A","Z8","X9","Y9","Z9","EY8","INCY"
*LABEL 165
*IF("EY9" GT 1):*GOTO 170
"T9","X10","Y10","Z10"
*GOTO 175
$
*LABEL 170

```

```

"T9","X9","Y9A","Z9","X10","Y10","Z10","EY9","INCY"
*LABEL 175
*IF("EY10" GT 1):*GOTO 180
"T10","X11","Y11","Z11"
*GOTO 185
$
*LABEL 180
"T10","X10","Y10A","Z10","X11","Y11","Z11","EY10","INCY"
*LABEL 185
$
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the $
$                               elements used in the mesh. $
$-----$
$
*(29 MAKE ELEM)ELEM
$
*XQT AUS                                $ arithmetic utility System processor
$
TABLE(NI=1,NJ=1): K THICK : I = 1 $ section property table for
                    J=1: 1.0      $ K41 conduction elements
$
TABLE(NI=1,NJ=1): K AREA  : I = 1 $ section property table for
                    J=1: 1.0      $ K21 conduction elements
$
*XQT ELD                                $ element definition processor
RESET NUTED = 1                        $ destination library
$
!NOPT = 1                              $ increment for the element number
!CLOS = 0
$
!J1 = 1                                $ lower outer facesheet group#1
!J2 = "J1"+"INCY"                      $ lower outer facesheet group#1
!J3 = "J1"+"INCY"*"EY1"                $ lower honeycomb group#2
!J4 = "J3"+"INCY"                      $ lower honeycomb group#2
!J5 = "J3"+"INCY"*"EY2"                $ lower inner facesheet group#3
!J6 = "J5"+"INCY"                      $ lower inner facesheet group#3
!J7 = "J5"+"INCY"*"EY3"                $ lower torque tube group#4
!J8 = "J7"+"INCY"                      $ lower torque tube group#4
!J9 = "J7"+"INCY"*"EY4"                $ air cavity group#5
!J10= "J9"+"INCY"                      $ air cavity group#5
!J11= "J9"+"INCY"*"EY5"                $ upper torque tube group#6
!J12= "J11"+"INCY"                     $ upper torque tube group#6
!J13= "J11"+"INCY"*"EY6"                $ upper inner facesheet group#7
!J14= "J13"+"INCY"                     $ upper inner facesheet group#7
!J15= "J13"+"INCY"*"EY7"                $ upper honeycomb group#8
!J16= "J15"+"INCY"                     $ upper honeycomb group#8
!J17= "J15"+"INCY"*"EY8"                $ upper outer facesheet group#9
!J18= "J17"+"INCY"                     $ upper outer facesheet group#9
!J19= "J17"+"INCY"*"EY9"                $ heater group#10
!J20= "J19"                            $ heater group#10
!J21= "J19"                            $ insulation group#11
!J22= "J19"+"INCY"                     $ insulation group#11
$
K21                                    $ 1-D conduction elements
$

```

```

GROUP = 1
NMAT = 7
"J1","J2","NOPT","EY1",1
$
GROUP = 2
NMAT = 5
"J3","J4","NOPT","EY2",1
$
GROUP = 3
NMAT = 1
"J5","J6","NOPT","EY3",1
$
GROUP = 4
NMAT = 6
"J7","J8","NOPT","EY4",1
$
GROUP = 5
NMAT = 4
"J9","J10","NOPT","EY5",1
$
GROUP = 6
NMAT = 6
"J11","J12","NOPT","EY6",1
$
GROUP = 7
NMAT = 1
"J13","J14","NOPT","EY7",1
$
GROUP = 8
NMAT = 5
"J15","J16","NOPT","EY8",1
$
GROUP = 9
NMAT = 7
"J17","J18","NOPT","EY9",1
$
GROUP = 10
NMAT = 2
"J19","J20","NOPT",1,1
$
GROUP = 11
NMAT = 3
"J21","J22","NOPT","EY10",1
$
*XQT TGEO
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the $
$ boundary conditions such as: $
$ - the constant temperature $
$ boundary $
$ - the applied heat flux boundary $
$ - the total experimental and $
$ heating time $
$ - the heated node boundary $
$ temperature $

```

```

$-----$
$
$(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the
$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TOPT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BOTT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=5,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTT 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$

```

```

$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!TOPT = DS 1,"CNTR",1 (5 EXPE TOPT 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
TABLE(NI=1,NJ=1):5 DELT TOPT:I=1:J=1:"TOPT"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT HTMP
DEFINE EE = 5 DELT FLUX
DEFINE FF = 5 DELT TOPT
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 10
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=EE,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)

```

```

$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM3
  TRANSFER(SOUR=F,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=FF,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=F,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
DEFINE TC = 5 APPL TEM3
$
$ One of the following table commands needs to be commented out
$ depending on the model used.
$
TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
  TRANSFER(SOUR=TB,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
TABLE(NI=1,NJ=2):5 TEMP NODE
  I=1
      J=1: 1.0
      J=2: "NONJ"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 10):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.           $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrta time step for TRTB (seconds)
$

```

```

!MIN1 = 1.0      $ lower bound for skins (PCp)
!MIN2 = 0.1      $ lower bound for skins (K)
!MIN3 = 10.0     $ lower bound for honeycomb (PCp)
!MIN4 = 10.0     $ lower bound for honeycomb (K)
!MIN5 = 10.0     $ lower bound for torque tubes (PCp)
!MIN6 = 10.0     $ lower bound for torque tubes (K)
!MIN7 = 10.0     $ lower bound for facesheets (PCp)
!MIN8 = 10.0     $ lower bound for facesheets (K)
!MIN9 = 10.0     $ lower bound for air gap (PCp)
!MINX = 10.0     $ lower bound for air gap (K)
!MAX1 = 5.0      $ upper bound for skins (PCp)
!MAX2 = 2.0      $ upper bound for skins (K)
!MAX3 = 10.0     $ upper bound for honeycomb (PCp)
!MAX4 = 10.0     $ upper bound for honeycomb (K)
!MAX5 = 10.0     $ upper bound for torque tubes (PCp)
!MAX6 = 10.0     $ upper bound for torque tubes (K)
!MAX7 = 10.0     $ upper bound for facesheets (PCp)
!MAX8 = 10.0     $ upper bound for facesheets (K)
!MAX9 = 10.0     $ upper bound for air gap (PCp)
!MAXX = 10.0     $ upper bound for air gap (K)
$
!A1  = 2.54      $ Initial value for skins (PCp)
!A2  = 1.00      $ Initial value for skins (K)
!A3  = 1.00      $ Initial value for honeycomb (PCp)
!A4  = 1.00      $ Initial value for honeycomb (K)
!A5  = 1.00      $ Initial value for torque tubes (PCp)
!A6  = 1.00      $ Initial value for torque tubes (K)
!A7  = 1.00      $ Initial value for facesheets (PCp)
!A8  = 1.00      $ Initial value for facesheets (K)
!A9  = 1.00      $ Initial value for air gap (PCp)
!A10 = 1.00      $ Initial value for air gap (K)
$
!A1I = 2.54      $ Initial value for skins (PCp)
!A2I = 1.00      $ Initial value for skins (K)
!A3I = 1.00      $ Initial value for honeycomb (PCp)
!A4I = 1.00      $ Initial value for honeycomb (K)
!A5I = 1.00      $ Initial value for torque tubes (PCp)
!A6I = 1.00      $ Initial value for torque tubes (K)
!A7I = 1.00      $ Initial value for facesheets (PCp)
!A8I = 1.00      $ Initial value for facesheets (K)
!A9I = 1.00      $ Initial value for air gap (PCp)
!A10I = 1.00     $ Initial value for air gap (K)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by $
$                               changing the Crank Nicholson para- $
$                               meter.                               $
$-----$
$
*(29 TRAN ANAL)ANAL
$
*XQT TRTB      $ implicit transient analyzer
  RESET QLIB = 5    $ thermal excitation library

```

```

RESET MLIB = 6                $ material and fluids properties library
RESET T1 = "TIMI"             $ starting time of analysis
RESET T2 = "TIMF"             $ termination time of analysis
RESET DT = "INTS"             $ integration time step size
RESET DEST = 4                $ output library
RESET PRINT = 0               $ online print control parameter
RESET KPRT = 0                $ print control parameter for k matrix
RESET MXNDT = 10000000        $ maximum # of int. time steps during TRTB
RESET BETA = 0.5              $ Crank Nicholson parameter
RESET KTI = 1                 $ K matrix computation time parameter
KTIME = 10000.                $ times to calculate the conduction load vector
TEMP = "CSTP"                 $ initial temperature of structure at T1
TSAVE = 5 EXPE TIME 1 1       $ storage of nodal temperatures at time
$                               $ increments of TSAVE
$
*RETURN
*ANAL
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU                       $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1         $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1        $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                       $ arithmetic utility processor
ERASE 4
$
!VAR1 = TOC,NWDS(2 TRAN TIME)  $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1               $ heated node number - 1
!VAR3 = "TOTN"-1               $ total node number - 1
!VAR4 = "NON2"-1               $ node at bottom of the torque tube
!VAR5 = "NON3"-1               $ node at top of the torque tube
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 1) :*GOTO 5000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2) :*GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3) :*GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4) :*GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5) :*GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6) :*GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7) :*GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8) :*GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9) :*GOTO 5080 $ Temperature histories for property 7
*IF("N2" EQ 10) :*GOTO 5082 $ Temperature histories for property 8
*IF("N2" EQ 11) :*GOTO 5084 $ Temperature histories for property 9
*IF("N2" EQ 12) :*GOTO 5086 $ Temperature histories for property 10
$

```



```

*LABEL 5000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
TABLE(NI=1,NJ="VAR1"):2 TOPT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR5",SSKIP="VAR3")
$
TABLE(NI=1,NJ="VAR1"):2 BOTT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR4",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
DEFINE D = 2 TOPT NODE 1 1
DEFINE E = 2 BOTT NODE 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTT 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
TABLE(NI=3,NJ="NDS"):5 TEMP TOPT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=F,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=D,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
TABLE(NI=3,NJ="NDS"):5 TEMP BOTT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=G,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=E,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
TABLE(NI=7,NJ="NDS"):8 ALL TEMP 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=6)
    TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=6)
    TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=6)
    TRAN(SOUR=F,ILIM=1,JLIM="NDS",DBASE=3,DSKIP=6)
    TRAN(SOUR=D,ILIM=1,JLIM="NDS",DBASE=4,DSKIP=6)
    TRAN(SOUR=G,ILIM=1,JLIM="NDS",DBASE=5,DSKIP=6)
    TRAN(SOUR=E,ILIM=1,JLIM="NDS",DBASE=6,DSKIP=6)
$
*GOTO 6000
$
*LABEL 5010
TABLE(NI=1,NJ="VAR1"):2 INIT 00 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$

```

```

*LABEL 5040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5082
TABLE(NI=1,NJ="VAR1"):2 PROP VIII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5084
TABLE(NI=1,NJ="VAR1"):2 PROP IX 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5086
TABLE(NI=1,NJ="VAR1"):2 PROP X 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ Subroutine (29 SENS COEF)COEF: This subroutine calculates $
$ the sensitivity coefficients $
$ for up to seven parameters. $
$-----$
$
$(29 SENS COEF)COEF
$
*XQT AUS $ Arithmetic Utility System
$
INLIB = 1 $ Read from library 1
OUTLI = 1 $ Write to library 1
$
*IF("N2" EQ 0) :*GOTO 1000 $ No perturbations
*IF("N2" EQ 1) :*GOTO 1000 $ No perturbations

```

```

*IF("N2" EQ 2) : *GOTO 1000      $ No perturbations
*IF("N2" EQ 3) : *GOTO 1010      $ Property 1 is perturbed
*IF("N2" EQ 4) : *GOTO 1020      $ Property 2 is perturbed
*IF("N2" EQ 5) : *GOTO 1030      $ Property 3 is perturbed
*IF("N2" EQ 6) : *GOTO 1040      $ Property 4 is perturbed
*IF("N2" EQ 7) : *GOTO 1050      $ Property 5 is perturbed
*IF("N2" EQ 8) : *GOTO 1060      $ Property 6 is perturbed
*IF("N2" EQ 9) : *GOTO 1070      $ Property 7 is perturbed
*IF("N2" EQ 10) : *GOTO 1080     $ Property 8 is perturbed
*IF("N2" EQ 11) : *GOTO 1090     $ Property 9 is perturbed
*IF("N2" EQ 12) : *GOTO 1100     $ Property 10 is perturbed
*GOTO 2000
$
*LABEL 1000
$
!A1  = "A1I"                    $ Set property 1 to initial guess
!A2  = "A2I"                    $ Set property 2 to initial guess
!A3  = "A3I"                    $ Set property 3 to initial guess
!A4  = "A4I"                    $ Set property 4 to initial guess
!A5  = "A5I"                    $ Set property 5 to initial guess
!A6  = "A6I"                    $ Set property 6 to initial guess
!A7  = "A7I"                    $ Set property 7 to initial guess
!A8  = "A8I"                    $ Set property 8 to initial guess
!A9  = "A9I"                    $ Set property 9 to initial guess
!A10 = "A10I"                   $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)            $ Update the thermal properties
*DCALL(29 TRAN ANAL)            $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)            $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1010
$
!A1  = 1.001*"A1I"              $ Perturb property 1
!DA1 = 0.001*"A1I"              $ Amount property 1 is perturbed
!A2  = "A2I"                    $ Set property 2 to initial guess
!A3  = "A3I"                    $ Set property 3 to initial guess
!A4  = "A4I"                    $ Set property 4 to initial guess
!A5  = "A5I"                    $ Set property 5 to initial guess
!A6  = "A6I"                    $ Set property 6 to initial guess
!A7  = "A7I"                    $ Set property 7 to initial guess
!A8  = "A8I"                    $ Set property 8 to initial guess
!A9  = "A9I"                    $ Set property 9 to initial guess
!A10 = "A10I"                   $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)            $ Update the thermal properties
*DCALL(29 TRAN ANAL)            $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)            $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1020
$
!A1  = "A1I"                    $ Set property 1 to initial guess
!A2  = 1.001*"A2I"              $ Perturb property 2
!DA2 = 0.001*"A2I"              $ Amount property 2 is perturbed
!A3  = "A3I"                    $ Set property 3 to initial guess
!A4  = "A4I"                    $ Set property 4 to initial guess
!A5  = "A5I"                    $ Set property 5 to initial guess
!A6  = "A6I"                    $ Set property 6 to initial guess

```

!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1030	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = 1.001*"A3I"	\$ Perturb property 3
!DA3 = 0.001*"A3I"	\$ Amount property 3 is perturbed
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1040	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = 1.001*"A4I"	\$ Perturb property 4
!DA4 = 0.001*"A4I"	\$ Amount property 4 is perturbed
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures)
\$	
*GOTO 2000	
*LABEL 1050	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = 1.001*"A5I"	\$ Perturb property 5
!DA5 = 0.001*"A5I"	\$ Amount property 5 is perturbed
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess

!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1060	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = 1.001*"A6I"	\$ Perturb property 6
!DA6 = 0.001*"A6I"	\$ Amount property 6 is perturbed
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1070	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = 1.001*"A7I"	\$ Perturb property 7
!DA7 = 0.001*"A7I"	\$ Amount property 7 is perturbed
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1080	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = 1.001*"A8I"	\$ Perturb property 8
!DA8 = 0.001*"A8I"	\$ Amount property 8 is perturbed
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess

\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1090	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = 1.001*"A9I"	\$ Perturb property 9
!DA9 = 0.001*"A9I"	\$ Amount property 9 is perturbed
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1100	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = 1.001*"A10I"	\$ Perturb property 10
!DA10 = 0.001*"A10I"	\$ Amount property 10 is perturbed
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 2000	
\$	
*IF("N2" EQ 0) : *GOTO 3000	\$ No sensitivity coefficients are calculated
*IF("N2" EQ 1) : *GOTO 3000	\$ No sensitivity coefficients are calculated
*IF("N2" EQ 2) : *GOTO 3000	\$ No sensitivity coefficients are calculated
*IF("N2" EQ 3) : *GOTO 3010	\$ Sensitivity coefficients for property 1
*IF("N2" EQ 4) : *GOTO 3020	\$ Sensitivity coefficients for property 1,2
*IF("N2" EQ 5) : *GOTO 3030	\$ Sensitivity coefficients for property 1,2,3
*IF("N2" EQ 6) : *GOTO 3040	\$ Sensitivity coefficients for property 1,2,3,4
*IF("N2" EQ 7) : *GOTO 3050	\$ Sensitivity coefficients for property 1,2,3,4,5
*IF("N2" EQ 8) : *GOTO 3060	\$ Sensitivity coefficients for property 1,2,3,4,5,6
*IF("N2" EQ 9) : *GOTO 3070	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7
*IF("N2" EQ 10) : *GOTO 3080	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7,8
*IF("N2" EQ 11) : *GOTO 3090	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9
*IF("N2" EQ 12) : *GOTO 3100	\$ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9,10

```

*GOTO 3000
$
*LABEL 3000
DEFINE T0 = 2 INIT 00 1 1 $ Initial temperature distribution with no perturbations
*GOTO 4000
$
*LABEL 3010
!S1P0 = 1.0/"DA1" $ Positive denominator used to numerically differentiate
!S1NE = -1.0/"DA1" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T1 = 2 PROP I 1 1 $ Temperature distribution for perturbed property 1
DIF1 = SUM("S1P0" T1,"S1NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3020
!S2P0 = 1.0/"DA2" $ Positive denominator used to numerically differentiate
!S2NE = -1.0/"DA2" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T2 = 2 PROP II 1 1 $ Temperature distribution for perturbed property 2
DIF2 = SUM("S2P0" T2,"S2NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3030
!S3P0 = 1.0/"DA3" $ Positive denominator used to numerically differentiate
!S3NE = -1.0/"DA3" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T3 = 2 PROP III 1 1 $ Temperature distribution for perturbed property 3
DIF3 = SUM("S3P0" T3,"S3NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3040
!S4P0 = 1.0/"DA4" $ Positive denominator used to numerically differentiate
!S4NE = -1.0/"DA4" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T4 = 2 PROP IV 1 1 $ Temperature distribution for perturbed property 4
DIF4 = SUM("S4P0" T4,"S4NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3050
!S5P0 = 1.0/"DA5" $ Positive denominator used to numerically differentiate
!S5NE = -1.0/"DA5" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T5 = 2 PROP V 1 1 $ Temperature distribution for perturbed property 5
DIF5 = SUM("S5P0" T5,"S5NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3060
!S6P0 = 1.0/"DA6" $ Positive denominator used to numerically differentiate
!S6NE = -1.0/"DA6" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T6 = 2 PROP VI 1 1 $ Temperature distribution for perturbed property 6
DIF6 = SUM("S6P0" T6,"S6NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3070
!S7P0 = 1.0/"DA7" $ Positive denominator used to numerically differentiate
!S7NE = -1.0/"DA7" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T7 = 2 PROP VII 1 1 $ Temperature distribution for perturbed property 7

```

```

DIF7 = SUM("S7P0" T7,"S7NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3080
!S8P0 = 1.0/"DA8"          $ Positive denominator used to numerically differentiate
!S8NE = -1.0/"DA8"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T8 = 2 PROP VIII 1 1 $ Temperature distribution for perturbed property 8
DIF8 = SUM("S8P0" T8,"S8NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3090
!S9P0 = 1.0/"DA9"          $ Positive denominator used to numerically differentiate
!S9NE = -1.0/"DA9"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T9 = 2 PROP IX 1 1 $ Temperature distribution for perturbed property 9
DIF9 = SUM("S9P0" T9,"S9NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3100
!SXPO = 1.0/"DA10"          $ Positive denominator used to numerically differentiate
!SXNE = -1.0/"DA10"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T10 = 2 PROP X 1 1 $ Temperature distribution for perturbed property 10
DIFX = SUM("SXPO" T10,"SXNE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 4000
!WORD = TOC,NJ(2 INIT 00 1 1) $ Determine the number of lines in INIT 00
$
*IF("NP" EQ 0) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 1) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 2) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 3) :*GOTO 6010 $ Calculate XTX, P11
*IF("NP" EQ 4) :*GOTO 6020 $ Calculate XTX, P11,P22
*IF("NP" EQ 5) :*GOTO 6030 $ Calculate XTX, P11,P22,P33
*IF("NP" EQ 6) :*GOTO 6040 $ Calculate XTX, P11,P22,P33,P44
*IF("NP" EQ 7) :*GOTO 6050 $ Calculate XTX, P11,P22,P33,P44,P55
*IF("NP" EQ 8) :*GOTO 6060 $ Calculate XTX, P11,P22,P33,P44,P55,P66
*IF("NP" EQ 9) :*GOTO 6070 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77
*IF("NP" EQ 10) :*GOTO 6080 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88
*IF("NP" EQ 11) :*GOTO 6090 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99
*IF("NP" EQ 12) :*GOTO 6100 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99,PXX
$
*LABEL 6010
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=2,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
TABLE(NI=1,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=0)
$
DEFINE S = 1 PMAT TOBE 1 1 $ Define table as register (X)
ST = RTRAN(S) $ XT
STS = RPROD(S,ST) $ XTX
STSI = RINV(STS) $ XTXI
!P11 = DS 1,1,1 (1 STSI AUS 1 1) $ numerical value of diagonal 1
*GOTO 5000

```



```

$
*LABEL 6020
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=3,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
TABLE(NI=2,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
$
DEFINE S = 1 PMAT TOBE 1 1
ST = RTRAN(S)
STS = RPROD(S,ST)
STSI = RINV(STS)
!P11 = DS 1,1,1 (1 STSI AUS 1 1)
!P22 = DS 2,2,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6030
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=4,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
TABLE(NI=3,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
$
DEFINE S = 1 PMAT TOBE 1 1
ST = RTRAN(S)
STS = RPROD(S,ST)
STSI = RINV(STS)
!P11 = DS 1,1,1 (1 STSI AUS 1 1)
!P22 = DS 2,2,1 (1 STSI AUS 1 1)
!P33 = DS 3,3,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6040
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=5,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
TABLE(NI=4,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)

```

```

TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6050
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=6,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
TABLE(NI=5,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6060
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=7,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)

```

```

TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
TABLE(NI=6,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6070
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=8,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
TABLE(NI=7,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)

```

```

!P66      = DS 6,6,1 (1 STSI AUS 1 1)
!P77      = DS 7,7,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6080
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=9,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
TABLE(NI=8,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6090
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1

```

```

DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=10,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
TABLE(NI=9,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
!P99    = DS 9,9,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6100
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1
DEFINE TR10 = 1 DIFX AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=11,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=10)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=10)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=10)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=10)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=10)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=10)

```

```

TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=10)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=10)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=10)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=10)
TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=10,DSKIP=10)
TABLE(NI=10,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1 ,1 ,1 (1 STSI AUS 1 1)
!P22    = DS 2 ,2 ,1 (1 STSI AUS 1 1)
!P33    = DS 3 ,3 ,1 (1 STSI AUS 1 1)
!P44    = DS 4 ,4 ,1 (1 STSI AUS 1 1)
!P55    = DS 5 ,5 ,1 (1 STSI AUS 1 1)
!P66    = DS 6 ,6 ,1 (1 STSI AUS 1 1)
!P77    = DS 7 ,7 ,1 (1 STSI AUS 1 1)
!P88    = DS 8 ,8 ,1 (1 STSI AUS 1 1)
!P99    = DS 9 ,9 ,1 (1 STSI AUS 1 1)
!PXX    = DS 10,10,1 (1 STSI AUS 1 1)
$
*GOTO 5000
*LABEL 5000
$
*RETURN
*COEF
$
$-----$
$ Subroutine(29 STAT ISTC)STAT: This subroutine does a statistical $
$                               analysis on the chromosomes, $
$                               including the calculation of $
$                               confidence intervals also called $
$                               CI's. $
$-----$
$
*(29 STAT ISTC)STAT
$
*IF("NP" EQ 0) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 1) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 2) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 3) :*GOTO 6010 $ Confidence intervals for properties 1
*IF("NP" EQ 4) :*GOTO 6020 $ Confidence intervals for properties 1,2
*IF("NP" EQ 5) :*GOTO 6030 $ Confidence intervals for properties 1,2,3
*IF("NP" EQ 6) :*GOTO 6040 $ Confidence intervals for properties 1,2,3,4
*IF("NP" EQ 7) :*GOTO 6050 $ Confidence intervals for properties 1,2,3,4,5
*IF("NP" EQ 8) :*GOTO 6060 $ Confidence intervals for properties 1,2,3,4,5,6
*IF("NP" EQ 9) :*GOTO 6070 $ Confidence intervals for properties 1,2,3,4,5,6,7
*IF("NP" EQ 10):*GOTO 6075 $ Confidence intervals for properties 1,2,3,4,5,6,7,8

```

```

*IF("NP" EQ 11):*GOTO 6080 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9
*IF("NP" EQ 12):*GOTO 6085 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9,10
$
*LABEL 6010
!CI1 = (P11**0.5)*1.960
TABLE(NI=1,NJ=1):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
*GOTO 7000
$
*LABEL 6020
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
TABLE(NI=1,NJ=2):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
*GOTO 7000
$
*LABEL 6030
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
TABLE(NI=1,NJ=3):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
*GOTO 7000
$
*LABEL 6040
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
TABLE(NI=1,NJ=4):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
*GOTO 7000
$
*LABEL 6050
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
TABLE(NI=1,NJ=5):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
*GOTO 7000
$
*LABEL 6060

```

```

!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
TABLE(NI=1,NJ=6):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  *GOTO 7000
$
*LABEL 6070
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
TABLE(NI=1,NJ=7):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  *GOTO 7000
$
*LABEL 6075
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
TABLE(NI=1,NJ=8):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  *GOTO 7000
$
*LABEL 6080
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960

```



```

!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
TABLE(NI=1,NJ=9):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  J=9:"CI9"
*GOTO 7000
$
*LABEL 6085
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
!CI10 = (PXX**0.5)*1.960
TABLE(NI=1,NJ=10):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  J=9:"CI9"
  J=10:"CI10"
*GOTO 7000
$
*LABEL 7000
*XQT DCU
  PRINT 1 CONF INTE 1 1
*XQT AUS
$
*RETURN
*STAT
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$

```

```

*XQT AUS                                $arithmetic utility system processor
$
$the following three comment statemnets can not be put at the end
$of the *TF commands, since it intervenes with the runstream.
$
$open the file containing the data
$read the datafile
$close the datafile
$
*TF OPEN 1'data
*TF READ 1
*TF CLOSE 1
$
!NP = 0
!N2 = 1                                $ counter to direct TRAN TEMP
$
*DCALL (29 MAKE VARI)                   $ call subroutine MAKE VARI
*DCALL (29 PROB SPEC)                   $ call subroutine PROB SPEC
*DCALL (29 MATE PROP)                   $ call subroutine MATE PROP
*DCALL (29 PLAT DIME)                   $ call subroutine PLAT DIME
*DCALL (29 NODE GENE)                   $ call subroutine NODE GENE
*DCALL (29 MAKE ELEM)                   $ call subroutine MAKE ELEM
*DCALL (29 BOUN COND)                   $ call subroutine BOUN COND
*DCALL (29 TRAN ANAL)                   $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)                   $ call subroutine TEMP TABL
$
!NP = 0
!N2 = 2                                $ initial conditions
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 0
!N2 = 3                                $ parameter 1
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 0
!N2 = 4                                $ parameter 2
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 0
!N2 = 5                                $ parameter 3
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 0
!N2 = 6                                $ parameter 4
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 0
!N2 = 7                                $ parameter 5
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 0
!N2 = 8                                $ parameter 6
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
!NP = 9
!N2 = 9                                $ parameter 7
*DCALL (29 SENS COEF)                   $ call subroutine SENS COEF
*DCALL (29 STAT ISTC)                   $ call subroutine STAT ISTC
*XQT EXIT

```

Appendix H

EAL One-Dimensional Detailed Half-Model: Forward Program

```
*XQT U1
*ONLINE=0
!MAXCM = SSP(0,13)
*CMPARA(MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of $
$                               the variables used in the program $
$                               such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each $
$                               section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $ arithmetic utility system processor
$
!NY6  = 8           $ number of nodes for the upper torque tube
!NY7  = 12          $ number of nodes for the upper inner facesheet
!NY8  = 36          $ number of nodes for the upper honeycomb structure
!NY9  = 8           $ number of nodes for the upper outer facesheet
!NY10 = 36          $ number of nodes for the insulation
$
!EY6  ="NY6"-1      $ number of elements for the upper torque tube
!EY7  ="NY7"-1      $ number of elements for the upper inner facesheet
!EY8  ="NY8"-1      $ number of elements for the upper honeycomb structure
!EY9  ="NY9"-1      $ number of elements for the upper outer facesheet
!EY10 ="NY10"-1     $ number of elements for the insulation
$
!TOTN = "NY6"+"EY7"+"EY8"+"EY9"+"EY10"
$                  $ total number of nodes in the model
!NON1 = "NY6"+"EY7"+"EY8"+"EY9"
$                  $ node number at the heater (high sensitivity)
!NONK = FLOAT("TOTN")
```

```

$                               $ node number on top of the insulation
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the $
$                               properties of the materials used $
$                               in the model. $
$-----$
$
*(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for IM7/954-2A
$ upper and lower skins
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
                        J=1: 1.E+6 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
                        J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
                        J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
$ conduction properties for SS321 honeycomb (actually SS347)
$
TABLE(NI=9,NJ=1):6 COND PROP 5: I = 2 3 4
                        J=1: 1.E+6 "A3" "A4"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tubes, edge channels
$
TABLE(NI=9,NJ=1):6 COND PROP 6: I= 2 3 4
                        J=1: 1.E+6 "A5" "A6"
$
$ conduction properties for PETI-5 of the upper and lower outer
$ facesheets.
$
TABLE(NI=9,NJ=1):6 COND PROP 7: I= 2 3 4
                        J=1: 1.E+6 "A7" "A8"
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$                               coordinates of the points used in $
$                               the geometry of the plate. The $
$                               points cited in the following $
$                               subroutine correspond to these $
$                               substructures: $
$                               6,7 :upper torque tube $

```

```

$              7A,8   :upper inner facesheet      $
$              8A,9   :upper honeycomb            $
$              9A,10  :upper outer facesheet      $
$              10A,11 :insulation                  $
$-----$
$
*(29 PLAT DIME)DIME
$
!X6  = 0.          $ x-coordinate for point 6
!X7  = 0.          $ x-coordinate for point 7
!X8  = 0.          $ x-coordinate for point 8
!X9  = 0.          $ x-coordinate for point 9
!X10 = 0.          $ x-coordinate for point 10
!X11 = 0.          $ x-coordinate for point 11
$
!Y6  = 0.0878      $ y-coordinate for point 6
!Y7  = 0.0899      $ y-coordinate for point 7
!Y7A = 0.0899+(0.0050/"EY7") $ y-coordinate for point 7A
!Y8  = 0.0949      $ y-coordinate for point 8
!Y8A = 0.0949+(0.0254/"EY8") $ y-coordinate for point 8A
!Y9  = 0.1203      $ y-coordinate for point 9
!Y9A = 0.1203+(0.0032/"EY9") $ y-coordinate for point 9A
!Y10 = 0.1235      $ y-coordinate for point 10
!Y10A= 0.1235+(0.0254/"EY10") $ y-coordinate for point 10A
!Y11 = 0.1489      $ y-coordinate for point 11
$
!Z6  = 0.          $ z-coordinate for point 6
!Z7  = 0.          $ z-coordinate for point 7
!Z8  = 0.          $ z-coordinate for point 8
!Z9  = 0.          $ z-coordinate for point 9
!Z10 = 0.          $ z-coordinate for point 10
!Z11 = 0.          $ z-coordinate for point 11
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the      $
$                               coordinates of the nodes used in  $
$                               the mesh.                          $
$-----$
$
*(29 NODE GENE)GENE
$
!INCY = 1          $ nodal increment in the y-direction
!T6   = 1          $ starting node number for upper torque tube edge
!T7   = "T6"+"NY6" $ starting node number for upper inner facesheet
!T8   = "T7"+"EY7" $ starting node number for upper honeycomb
!T9   = "T8"+"EY8" $ starting node number for upper outer facesheet
!T10  = "T9"+"EY9" $ starting node number for insulation
$
*XQT TAB          $ table generator, structural properties
$
START "TOTN"      $ memory allocation for all the nodes
JLOC              $ initiation of the creation of the dataset containing
$                $ the joint position coordinates
FORMAT = 1        $ specifies rectangular coordinates
$
"T6","X6","Y6","Z6","X7","Y7","Z7","NY6","INCY"

```

```

*LABEL 145
*IF("EY7" GT 1):*GOTO 150
"T7","X8","Y8","Z8"
*GOTO 155
$
*LABEL 150
"T7","X7","Y7A","Z7","X8","Y8","Z8","EY7","INCY"
*LABEL 155
*IF("EY8" GT 1):*GOTO 160
"T8","X9","Y9","Z9"
*GOTO 165
$
*LABEL 160
"T8","X8","Y8A","Z8","X9","Y9","Z9","EY8","INCY"
*LABEL 165
*IF("EY9" GT 1):*GOTO 170
"T9","X10","Y10","Z10"
*GOTO 175
$
*LABEL 170
"T9","X9","Y9A","Z9","X10","Y10","Z10","EY9","INCY"
*LABEL 175
*IF("EY10" GT 1):*GOTO 180
"T10","X11","Y11","Z11"
*GOTO 185
$
*LABEL 180
"T10","X10","Y10A","Z10","X11","Y11","Z11","EY10","INCY"
*LABEL 185
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the $
$ elements used in the mesh. $
$-----$
$
*(29 MAKE ELEM)ELEM
$
*XQT AUS $ arithmetic utility System processor
$
TABLE(NI=1,NJ=1): K THICK : I = 1 $ section property table for
J=1: 1.0 $ K41 conduction elements
$
TABLE(NI=1,NJ=1): K AREA : I = 1 $ section property table for
J=1: 1.0 $ K21 conduction elements
$
*XQT ELD $ element definition processor
RESET NUTED = 1 $ destination library
$
!NOPT = 1 $ increment for the element number
!CLOS = 0
$
!J11= 1 $ upper torque tube group#1
!J12= "J11"+"INCY" $ upper torque tube group#1
!J13= "J11"+"INCY"*"EY6") $ upper inner facesheet group#2
!J14= "J13"+"INCY" $ upper inner facesheet group#2
!J15= "J13"+"INCY"*"EY7") $ upper honeycomb group#3

```

```

!J16= "J15"+"INCY"          $ upper honeycomb      group#3
!J17= "J15"+"INCY"*"EY8")    $ upper outer facesheet group#4
!J18= "J17"+"INCY"          $ upper outer facesheet group#4
!J19= "J17"+"INCY"*"EY9")    $ heater            group#5
!J20= "J19"                  $ heater            group#5
!J21= "J19"                  $ insulation         group#6
!J22= "J19"+"INCY"          $ insulation         group#6
$
K21                            $ 1-D conduction elements
$
GROUP = 1                      $ group 1
NMAT = 6                       $ material 6 conduction properties 6
"J11","J12","NOPT","EY6",1
$
GROUP = 2                      $ group 2
NMAT = 1                       $ material 7 conduction properties 7
"J13","J14","NOPT","EY7",1
$
GROUP = 3                      $ group 3
NMAT = 5                       $ material 8 conduction properties 8
"J15","J16","NOPT","EY8",1
$
GROUP = 4                      $ group 4
NMAT = 7                       $ material 9 conduction properties 9
"J17","J18","NOPT","EY9",1
$
GROUP = 5                      $ group 5
NMAT = 2                       $ material 10 conduction properties 10
"J19","J20","NOPT",1,1
$
GROUP = 6                      $ group 6
NMAT = 3                       $ material 11 conduction properties 11
"J21","J22","NOPT","EY10",1
$
*XQT TGEO                      $ element geometry processor
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the $
$                               boundary conditions such as: $
$                               - the constant temperature $
$                               boundary $
$                               - the applied heat flux boundary $
$                               - the total experimental and $
$                               heating time $
$                               - the heated node boundary $
$                               temperature $
$-----$
$
*(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the

```

```

$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TOPT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BOTT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=5,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTT 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)

```



```

!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!TOPT = DS 1,"CNTR",1 (5 EXPE TOPT 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
TABLE(NI=1,NJ=1):5 DELT TOPT:I=1:J=1:"TOPT"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT HTMP
DEFINE EE = 5 DELT FLUX
DEFINE FF = 5 DELT TOPT
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 5
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=EE,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM3
TRANSFER(SOUR=F,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=FF,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=F,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)

```

```

$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
DEFINE TC = 5 APPL TEM3
$
TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=TC,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
  TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
TABLE(NI=1,NJ=2):5 TEMP NODE
  I=1
      J=1:  1.0
      J=2:  "NONK"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21  5):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.           $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrtral time step for TRTB (seconds)
$
!MIN1 = 1.0      $ lower bound for skins (PCp)
!MIN2 = 0.1      $ lower bound for skins (K)
!MIN3 = 10.0     $ lower bound for honeycomb (PCp)
!MIN4 = 10.0     $ lower bound for honeycomb (K)
!MIN5 = 10.0     $ lower bound for torque tubes (PCp)
!MIN6 = 10.0     $ lower bound for torque tubes (K)
!MIN7 = 10.0     $ lower bound for facesheets (PCp)
!MIN8 = 10.0     $ lower bound for facesheets (K)
!MIN9 = 10.0     $ lower bound for air gap (PCp)
!MINX = 10.0     $ lower bound for air gap (K)
!MAX1 = 5.0      $ upper bound for skins (PCp)
!MAX2 = 2.0      $ upper bound for skins (K)
!MAX3 = 10.0     $ upper bound for honeycomb (PCp)

```

```

!MAX4 = 10.0      $ upper bound for honeycomb (K)
!MAX5 = 10.0      $ upper bound for torque tubes (PCp)
!MAX6 = 10.0      $ upper bound for torque tubes (K)
!MAX7 = 10.0      $ upper bound for facesheets (PCp)
!MAX8 = 10.0      $ upper bound for facesheets (K)
!MAX9 = 10.0      $ upper bound for air gap (PCp)
!MAXX = 10.0      $ upper bound for air gap (K)
$
!A1  = 2.54      $ Initial value for skins (PCp)
!A2  = 1.00      $ Initial value for skins (K)
!A3  = 1.00      $ Initial value for honeycomb (PCp)
!A4  = 1.00      $ Initial value for honeycomb (K)
!A5  = 1.00      $ Initial value for torque tubes (PCp)
!A6  = 1.00      $ Initial value for torque tubes (K)
!A7  = 1.00      $ Initial value for facesheets (PCp)
!A8  = 1.00      $ Initial value for facesheets (K)
!A9  = 1.00      $ Initial value for air gap (PCp)
!A10 = 1.00      $ Initial value for air gap (K)
$
!A1I = 2.54      $ Initial value for skins (PCp)
!A2I = 1.00      $ Initial value for skins (K)
!A3I = 1.00      $ Initial value for honeycomb (PCp)
!A4I = 1.00      $ Initial value for honeycomb (K)
!A5I = 1.00      $ Initial value for torque tubes (PCp)
!A6I = 1.00      $ Initial value for torque tubes (K)
!A7I = 1.00      $ Initial value for facesheets (PCp)
!A8I = 1.00      $ Initial value for facesheets (K)
!A9I = 1.00      $ Initial value for air gap (PCp)
!A10I = 1.00     $ Initial value for air gap (K)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by $
$                               changing the Crank Nicholson para- $
$                               meter.                               $
$-----$
$
*(29 TRAN ANAL)ANAL
$
*XQT TRTB      $ implicit transient analyzer
RESET QLIB = 5  $ thermal excitation library
RESET MLIB = 6  $ material and fluids properties library
RESET T1  = "TIMI" $ starting time of analysis
RESET T2  = "TIMF" $ termination time of analysis
RESET DT  = "INTS" $ integration time step size
RESET DEST = 4   $ output library
RESET PRINT = 0  $ online print control parameter
RESET KPRT = 0   $ print control parameter for k matrix
RESET MXNDT = 10000000 $ maximum # of int. time steps during TRTB
RESET BETA = 0.5  $ Crank Nicholson parameter
RESET KTI  = 1    $ K matrix computation time parameter
KTIME     = 10000. $ times to calculate the conduction load vector
TEMP      = "CSTP" $ initial temperature of structure at T1
TSAVE     = 5 EXPE TIME 1 1 $ storage of nodal temperatures at time

```

```

$                                $ increments of TSAVE
*RETURN
*ANAL
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU                        $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1          $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1          $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                        $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME)    $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1                 $ heated node number - 1
!VAR3 = "TOTN"-1                 $ total node number - 1
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 0) :*GOTO 9000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 1) :*GOTO 9000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2) :*GOTO 7010 $ Temperature histories for initial conditions
*IF("N2" EQ 3) :*GOTO 7020 $ Temperature histories for property 1
*IF("N2" EQ 4) :*GOTO 7030 $ Temperature histories for property 2
*IF("N2" EQ 5) :*GOTO 7040 $ Temperature histories for property 3
*IF("N2" EQ 6) :*GOTO 7050 $ Temperature histories for property 4
*IF("N2" EQ 7) :*GOTO 7060 $ Temperature histories for property 5
*IF("N2" EQ 8) :*GOTO 7070 $ Temperature histories for property 6
*IF("N2" EQ 9) :*GOTO 7080 $ Temperature histories for property 7
*IF("N2" EQ 10):*GOTO 7090 $ Temperature histories for property 8
*IF("N2" EQ 11):*GOTO 7100 $ Temperature histories for property 9
*IF("N2" EQ 12):*GOTO 7110 $ Temperature histories for property 10
$
*LABEL 9000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
*GOTO 8000
$

```

```

*LABEL 7010
TABLE(NI=1,NJ="VAR1"):2 INIT 00 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7090
TABLE(NI=1,NJ="VAR1"):2 PROP VIII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7100
TABLE(NI=1,NJ="VAR1"):2 PROP IX 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7110
TABLE(NI=1,NJ="VAR1"):2 PROP X 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 8000
*RETURN
*TEMP
$

```

```

$-----$
$ Subroutine (29 SENS COEF)COEF: This subroutine calculates $
$                               the sensitivity coefficients $
$                               for up to seven parameters. $
$-----$
$
*(29 SENS COEF)COEF
$
*XQT AUS                      $ Arithmetic Utility System
$
INLIB = 1                      $ Read from library 1
OUTLI = 1                      $ Write to library 1
$
*IF("N2" EQ 0) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 1) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 2) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 3) :*GOTO 1010    $ Property 1 is perturbed
*IF("N2" EQ 4) :*GOTO 1020    $ Property 2 is perturbed
*IF("N2" EQ 5) :*GOTO 1030    $ Property 3 is perturbed
*IF("N2" EQ 6) :*GOTO 1040    $ Property 4 is perturbed
*IF("N2" EQ 7) :*GOTO 1050    $ Property 5 is perturbed
*IF("N2" EQ 8) :*GOTO 1060    $ Property 6 is perturbed
*IF("N2" EQ 9) :*GOTO 1070    $ Property 7 is perturbed
*IF("N2" EQ 10):*GOTO 1080    $ Property 8 is perturbed
*IF("N2" EQ 11):*GOTO 1090    $ Property 9 is perturbed
*IF("N2" EQ 12):*GOTO 1100    $ Property 10 is perturbed
*GOTO 2000
$
*LABEL 1000
$
!A1  = "A1I"                  $ Set property 1 to initial guess
!A2  = "A2I"                  $ Set property 2 to initial guess
!A3  = "A3I"                  $ Set property 3 to initial guess
!A4  = "A4I"                  $ Set property 4 to initial guess
!A5  = "A5I"                  $ Set property 5 to initial guess
!A6  = "A6I"                  $ Set property 6 to initial guess
!A7  = "A7I"                  $ Set property 7 to initial guess
!A8  = "A8I"                  $ Set property 8 to initial guess
!A9  = "A9I"                  $ Set property 9 to initial guess
!A10 = "A10I"                 $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)          $ Update the thermal properties
*DCALL(29 TRAN ANAL)          $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)          $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1010
$
!A1  = 1.001*"A1I"            $ Perturb property 1
!DA1 = 0.001*"A1I"            $ Amount property 1 is perturbed
!A2  = "A2I"                  $ Set property 2 to initial guess
!A3  = "A3I"                  $ Set property 3 to initial guess
!A4  = "A4I"                  $ Set property 4 to initial guess
!A5  = "A5I"                  $ Set property 5 to initial guess
!A6  = "A6I"                  $ Set property 6 to initial guess
!A7  = "A7I"                  $ Set property 7 to initial guess
!A8  = "A8I"                  $ Set property 8 to initial guess
!A9  = "A9I"                  $ Set property 9 to initial guess
!A10 = "A10I"                 $ Set property 10 to initial guess

```

\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1020	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = 1.001*"A2I"	\$ Perturb property 2
!DA2 = 0.001*"A2I"	\$ Amount property 2 is perturbed
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1030	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = 1.001*"A3I"	\$ Perturb property 3
!DA3 = 0.001*"A3I"	\$ Amount property 3 is perturbed
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1040	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = 1.001*"A4I"	\$ Perturb property 4
!DA4 = 0.001*"A4I"	\$ Amount property 4 is perturbed
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties

*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures)
\$	
*GOTO 2000	
*LABEL 1050	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = 1.001*"A5I"	\$ Perturb property 5
!DA5 = 0.001*"A5I"	\$ Amount property 5 is perturbed
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1060	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = 1.001*"A6I"	\$ Perturb property 6
!DA6 = 0.001*"A6I"	\$ Amount property 6 is perturbed
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1070	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = 1.001*"A7I"	\$ Perturb property 7
!DA7 = 0.001*"A7I"	\$ Amount property 7 is perturbed
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures


```

$
*GOTO 2000
*LABEL 1080
$
!A1 = "A1I"           $ Set property 1 to initial guess
!A2 = "A2I"           $ Set property 2 to initial guess
!A3 = "A3I"           $ Set property 3 to initial guess
!A4 = "A4I"           $ Set property 4 to initial guess
!A5 = "A5I"           $ Set property 5 to initial guess
!A6 = "A6I"           $ Set property 6 to initial guess
!A7 = "A7I"           $ Set property 7 to initial guess
!A8 = 1.001*"A8I"      $ Perturb property 8
!DA8 = 0.001*"A8I"     $ Amount property 8 is perturbed
!A9 = "A9I"           $ Set property 9 to initial guess
!A10 = "A10I"          $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1090
$
!A1 = "A1I"           $ Set property 1 to initial guess
!A2 = "A2I"           $ Set property 2 to initial guess
!A3 = "A3I"           $ Set property 3 to initial guess
!A4 = "A4I"           $ Set property 4 to initial guess
!A5 = "A5I"           $ Set property 5 to initial guess
!A6 = "A6I"           $ Set property 6 to initial guess
!A7 = "A7I"           $ Set property 7 to initial guess
!A8 = "A8I"           $ Set property 8 to initial guess
!A9 = 1.001*"A9I"      $ Perturb property 9
!DA9 = 0.001*"A9I"     $ Amount property 9 is perturbed
!A10 = "A10I"          $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1100
$
!A1 = "A1I"           $ Set property 1 to initial guess
!A2 = "A2I"           $ Set property 2 to initial guess
!A3 = "A3I"           $ Set property 3 to initial guess
!A4 = "A4I"           $ Set property 4 to initial guess
!A5 = "A5I"           $ Set property 5 to initial guess
!A6 = "A6I"           $ Set property 6 to initial guess
!A7 = "A7I"           $ Set property 7 to initial guess
!A8 = "A8I"           $ Set property 8 to initial guess
!A9 = "A9I"           $ Set property 9 to initial guess
!A10 = 1.001*"A10I"    $ Perturb property 10
!DA10 = 0.001*"A10I"   $ Amount property 10 is perturbed
$
*DCALL(29 MATE PROP)   $ Update the thermal properties
*DCALL(29 TRAN ANAL)   $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)   $ Tabularize specific nodal temperatures
$
*GOTO 2000

```

```

*LABEL 2000
$
*IF("N2" EQ 0) :*GOTO 3000    $ No sensitivity coefficients are calculated
*IF("N2" EQ 1) :*GOTO 3000    $ No sensitivity coefficients are calculated
*IF("N2" EQ 2) :*GOTO 3000    $ No sensitivity coefficients are calculated
*IF("N2" EQ 3) :*GOTO 3010    $ Sensitivity coefficients for property 1
*IF("N2" EQ 4) :*GOTO 3020    $ Sensitivity coefficients for property 1,2
*IF("N2" EQ 5) :*GOTO 3030    $ Sensitivity coefficients for property 1,2,3
*IF("N2" EQ 6) :*GOTO 3040    $ Sensitivity coefficients for property 1,2,3,4
*IF("N2" EQ 7) :*GOTO 3050    $ Sensitivity coefficients for property 1,2,3,4,5
*IF("N2" EQ 8) :*GOTO 3060    $ Sensitivity coefficients for property 1,2,3,4,5,6
*IF("N2" EQ 9) :*GOTO 3070    $ Sensitivity coefficients for property 1,2,3,4,5,6,7
*IF("N2" EQ 10):*GOTO 3080    $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8
*IF("N2" EQ 11):*GOTO 3090    $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9
*IF("N2" EQ 12):*GOTO 3100    $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9,10
*GOTO 3000
$
*LABEL 3000
DEFINE TO = 2 INIT 00 1 1    $ Initial temperature distribution with no perturbations
*GOTO 4000
$
*LABEL 3010
!S1PO = 1.0/"DA1"            $ Positive denominator used to numerically differentiate
!S1NE = -1.0/"DA1"           $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT 00 1 1    $ Temperature distribution for no perturbations
DEFINE T1 = 2 PROP I 1 1    $ Temperature distribution for perturbed property 1
DIF1 = SUM("S1PO" T1,"S1NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3020
!S2PO = 1.0/"DA2"            $ Positive denominator used to numerically differentiate
!S2NE = -1.0/"DA2"           $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT 00 1 1    $ Temperature distribution for no perturbations
DEFINE T2 = 2 PROP II 1 1    $ Temperature distribution for perturbed property 2
DIF2 = SUM("S2PO" T2,"S2NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3030
!S3PO = 1.0/"DA3"            $ Positive denominator used to numerically differentiate
!S3NE = -1.0/"DA3"           $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT 00 1 1    $ Temperature distribution for no perturbations
DEFINE T3 = 2 PROP III 1 1    $ Temperature distribution for perturbed property 3
DIF3 = SUM("S3PO" T3,"S3NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3040
!S4PO = 1.0/"DA4"            $ Positive denominator used to numerically differentiate
!S4NE = -1.0/"DA4"           $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT 00 1 1    $ Temperature distribution for no perturbations
DEFINE T4 = 2 PROP IV 1 1    $ Temperature distribution for perturbed property 4
DIF4 = SUM("S4PO" T4,"S4NE" TO) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3050
!S5PO = 1.0/"DA5"            $ Positive denominator used to numerically differentiate
!S5NE = -1.0/"DA5"           $ Negative denominator used to numerically differentiate
DEFINE TO = 2 INIT 00 1 1    $ Temperature distribution for no perturbations
DEFINE T5 = 2 PROP V 1 1    $ Temperature distribution for perturbed property 5
DIF5 = SUM("S5PO" T5,"S5NE" TO) $Numerical differentiation of numerator

```

```

*GOTO 4000
$
*LABEL 3060
!S6PO = 1.0/"DA6"          $ Positive denominator used to numerically differentiate
!S6NE = -1.0/"DA6"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T6 = 2 PROP VI 1 1 $ Temperature distribution for perturbed property 6
DIF6 = SUM("S6PO" T6,"S6NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3070
!S7PO = 1.0/"DA7"          $ Positive denominator used to numerically differentiate
!S7NE = -1.0/"DA7"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T7 = 2 PROP VII 1 1 $ Temperature distribution for perturbed property 7
DIF7 = SUM("S7PO" T7,"S7NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3080
!S8PO = 1.0/"DA8"          $ Positive denominator used to numerically differentiate
!S8NE = -1.0/"DA8"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T8 = 2 PROP VIII 1 1 $ Temperature distribution for perturbed property 8
DIF8 = SUM("S8PO" T8,"S8NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3090
!S9PO = 1.0/"DA9"          $ Positive denominator used to numerically differentiate
!S9NE = -1.0/"DA9"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T9 = 2 PROP IX 1 1 $ Temperature distribution for perturbed property 9
DIF9 = SUM("S9PO" T9,"S9NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3100
!SXPO = 1.0/"DA10"          $ Positive denominator used to numerically differentiate
!SXNE = -1.0/"DA10"         $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T10 = 2 PROP X 1 1 $ Temperature distribution for perturbed property 10
DIFX = SUM("SXPO" T10,"SXNE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 4000
!WORD = TOC,NJ(2 INIT 00 1 1) $ Determine the number of lines in INIT 00
$
*IF("NP" EQ 0) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 1) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 2) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 3) :*GOTO 6010 $ Calculate XTX, P11
*IF("NP" EQ 4) :*GOTO 6020 $ Calculate XTX, P11,P22
*IF("NP" EQ 5) :*GOTO 6030 $ Calculate XTX, P11,P22,P33
*IF("NP" EQ 6) :*GOTO 6040 $ Calculate XTX, P11,P22,P33,P44
*IF("NP" EQ 7) :*GOTO 6050 $ Calculate XTX, P11,P22,P33,P44,P55
*IF("NP" EQ 8) :*GOTO 6060 $ Calculate XTX, P11,P22,P33,P44,P55,P66
*IF("NP" EQ 9) :*GOTO 6070 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77
*IF("NP" EQ 10) :*GOTO 6080 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88
*IF("NP" EQ 11) :*GOTO 6090 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99
*IF("NP" EQ 12) :*GOTO 6100 $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99,PXX
$

```

```

*LABEL 6010
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=2,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
TABLE(NI=1,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=0)
$
DEFINE S = 1 PMAT TOBE 1 1          $ Define table as register (X)
ST      = RTRAN(S)                  $ XT
STS     = RPROD(S,ST)               $ XTX
STSI    = RINV(STS)                 $ XTXI
!P11    = DS 1,1,1 (1 STSI AUS 1 1) $ numerical value of diagonal 1
*GOTO 5000
$
*LABEL 6020
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=3,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
TABLE(NI=2,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6030
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=4,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
TABLE(NI=3,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
*GOTO 5000

```

```

$
*LABEL 6040
DEFINE TR1  = 1 DIF1 AUS  1 1
DEFINE TR2  = 1 DIF2 AUS  1 1
DEFINE TR3  = 1 DIF3 AUS  1 1
DEFINE TR4  = 1 DIF4 AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=5,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
TABLE(NI=4,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
$
DEFINE S = 1 PMAT TOBE 1 1
ST       = RTRAN(S)
STS      = RPROD(S,ST)
STSI     = RINV(STS)
!P11     = DS 1,1,1 (1 STSI AUS 1 1)
!P22     = DS 2,2,1 (1 STSI AUS 1 1)
!P33     = DS 3,3,1 (1 STSI AUS 1 1)
!P44     = DS 4,4,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6050
DEFINE TR1  = 1 DIF1 AUS  1 1
DEFINE TR2  = 1 DIF2 AUS  1 1
DEFINE TR3  = 1 DIF3 AUS  1 1
DEFINE TR4  = 1 DIF4 AUS  1 1
DEFINE TR5  = 1 DIF5 AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=6,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
TABLE(NI=5,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
$
DEFINE S = 1 PMAT TOBE 1 1
ST       = RTRAN(S)
STS      = RPROD(S,ST)
STSI     = RINV(STS)
!P11     = DS 1,1,1 (1 STSI AUS 1 1)
!P22     = DS 2,2,1 (1 STSI AUS 1 1)
!P33     = DS 3,3,1 (1 STSI AUS 1 1)
!P44     = DS 4,4,1 (1 STSI AUS 1 1)
!P55     = DS 5,5,1 (1 STSI AUS 1 1)

```

```

*GOTO 5000
$
*LABEL 6060
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=7,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
TABLE(NI=6,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
$
DEFINE S = 1 PMAT TOBE 1 1
ST = RTRAN(S)
STS = RPROD(S,ST)
STSI = RINV(STS)
!P11 = DS 1,1,1 (1 STSI AUS 1 1)
!P22 = DS 2,2,1 (1 STSI AUS 1 1)
!P33 = DS 3,3,1 (1 STSI AUS 1 1)
!P44 = DS 4,4,1 (1 STSI AUS 1 1)
!P55 = DS 5,5,1 (1 STSI AUS 1 1)
!P66 = DS 6,6,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6070
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=8,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
TABLE(NI=7,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)

```

```

TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6080
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=9,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
TABLE(NI=8,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)

```

```

!P66      = DS 6,6,1 (1 STSI AUS 1 1)
!P77      = DS 7,7,1 (1 STSI AUS 1 1)
!P88      = DS 8,8,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6090
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=10,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
TABLE(NI=9,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
!P99    = DS 9,9,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6100
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1

```



```

DEFINE TR5  = 1 DIF5 AUS  1 1
DEFINE TR6  = 1 DIF6 AUS  1 1
DEFINE TR7  = 1 DIF7 AUS  1 1
DEFINE TR8  = 1 DIF8 AUS  1 1
DEFINE TR9  = 1 DIF9 AUS  1 1
DEFINE TR10 = 1 DIFX AUS  1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=11,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=10)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=10)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=10)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=10)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=10)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=10)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=10)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=10)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=10)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=10)
  TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=10,DSKIP=10)
TABLE(NI=10,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
  TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
  TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
  TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
  TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
  TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
  TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
  TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1 ,1 ,1 (1 STSI AUS 1 1)
!P22    = DS 2 ,2 ,1 (1 STSI AUS 1 1)
!P33    = DS 3 ,3 ,1 (1 STSI AUS 1 1)
!P44    = DS 4 ,4 ,1 (1 STSI AUS 1 1)
!P55    = DS 5 ,5 ,1 (1 STSI AUS 1 1)
!P66    = DS 6 ,6 ,1 (1 STSI AUS 1 1)
!P77    = DS 7 ,7 ,1 (1 STSI AUS 1 1)
!P88    = DS 8 ,8 ,1 (1 STSI AUS 1 1)
!P99    = DS 9 ,9 ,1 (1 STSI AUS 1 1)
!PXX    = DS 10,10,1 (1 STSI AUS 1 1)
$
*GOTO 5000
*LABEL 5000
$
*RETURN
*COEF
$
$-----$
$ Subroutine(29 STAT ISTC)STAT: This subroutine does a statistical $
$                               analysis on the chromosomes, $
$                               including the calculation of $
$                               confidence intervals also called $
$                               CI's. $

```

```

$-----$
$
*(29 STAT ISTD)STAT
$
*IF("NP" EQ 0) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 1) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 2) :*GOTO 7000 $ No confidence intervals to calculate
*IF("NP" EQ 3) :*GOTO 6010 $ Confidence intervals for properties 1
*IF("NP" EQ 4) :*GOTO 6020 $ Confidence intervals for properties 1,2
*IF("NP" EQ 5) :*GOTO 6030 $ Confidence intervals for properties 1,2,3
*IF("NP" EQ 6) :*GOTO 6040 $ Confidence intervals for properties 1,2,3,4
*IF("NP" EQ 7) :*GOTO 6050 $ Confidence intervals for properties 1,2,3,4,5
*IF("NP" EQ 8) :*GOTO 6060 $ Confidence intervals for properties 1,2,3,4,5,6
*IF("NP" EQ 9) :*GOTO 6070 $ Confidence intervals for properties 1,2,3,4,5,6,7
*IF("NP" EQ 10) :*GOTO 6075 $ Confidence intervals for properties 1,2,3,4,5,6,7,8
*IF("NP" EQ 11) :*GOTO 6080 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9
*IF("NP" EQ 12) :*GOTO 6085 $ Confidence intervals for properties 1,2,3,4,5,6,7,8,9,10
$
*LABEL 6010
!CI1 = (P11**0.5)*1.960
TABLE(NI=1,NJ=1):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
*GOTO 7000
$
*LABEL 6020
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
TABLE(NI=1,NJ=2):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
*GOTO 7000
$
*LABEL 6030
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
TABLE(NI=1,NJ=3):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
*GOTO 7000
$
*LABEL 6040
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
TABLE(NI=1,NJ=4):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
*GOTO 7000
$
*LABEL 6050

```

```

!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
TABLE(NI=1,NJ=5):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
*GOTO 7000
$
*LABEL 6060
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
TABLE(NI=1,NJ=6):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
*GOTO 7000
$
*LABEL 6070
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
TABLE(NI=1,NJ=7):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
*GOTO 7000
$
*LABEL 6075
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960

```

```

TABLE(NI=1,NJ=8):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  *GOTO 7000
$
*LABEL 6080
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
TABLE(NI=1,NJ=9):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  J=9:"CI9"
  *GOTO 7000
$
*LABEL 6085
!CI1 = (P11**0.5)*1.960
!CI2 = (P22**0.5)*1.960
!CI3 = (P33**0.5)*1.960
!CI4 = (P44**0.5)*1.960
!CI5 = (P55**0.5)*1.960
!CI6 = (P66**0.5)*1.960
!CI7 = (P77**0.5)*1.960
!CI8 = (P88**0.5)*1.960
!CI9 = (P99**0.5)*1.960
!CI10 = (PXX**0.5)*1.960
TABLE(NI=1,NJ=10):1 CONF INTE 1 1
  I=1
  J=1:"CI1"
  J=2:"CI2"
  J=3:"CI3"
  J=4:"CI4"
  J=5:"CI5"
  J=6:"CI6"
  J=7:"CI7"
  J=8:"CI8"
  J=9:"CI9"
  J=10:"CI10"
  *GOTO 7000

```

```

$
*LABEL 7000
*XQT DCU
  PRINT 1 CONF INTE 1 1
*XQT AUS
$
*RETURN
*STAT
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS                      $arithmetic utility system processor
$
$the following three comment statemnets can not be put at the end
$of the *TF commands, since it intervenes with the runstream.
$
$open the file containing the data
$read the datafile
$close the datafile
$
*TF OPEN  1'data
*TF READ  1
*TF CLOSE 1
$
!NP = 0
!N2 = 1                      $ counter to direct TRAN TEMP
$
*DCALL (29 MAKE VARI)        $ call subroutine MAKE VARI
*DCALL (29 PROB SPEC)        $ call subroutine PROB SPEC
*DCALL (29 MATE PROP)        $ call subroutine MATE PROP
*DCALL (29 PLAT DIME)        $ call subroutine PLAT DIME
*DCALL (29 NODE GENE)        $ call subroutine NODE GENE
*DCALL (29 MAKE ELEM)        $ call subroutine MAKE ELEM
*DCALL (29 BOUN COND)        $ call subroutine BOUN COND
*DCALL (29 TRAN ANAL)        $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)        $ call subroutine TEMP TABL
$
!NP = 0
!N2 = 2                      $ initial conditions
*DCALL (29 SENS COEF)        $ call subroutine SENS COEF
!NP = 0
!N2 = 3                      $ parameter 1
*DCALL (29 SENS COEF)        $ call subroutine SENS COEF
!NP = 0
!N2 = 4                      $ parameter 2
*DCALL (29 SENS COEF)        $ call subroutine SENS COEF
!NP = 0
!N2 = 5                      $ parameter 3
*DCALL (29 SENS COEF)        $ call subroutine SENS COEF
!NP = 0
!N2 = 6                      $ parameter 4
*DCALL (29 SENS COEF)        $ call subroutine SENS COEF
!NP = 0
!N2 = 7                      $ parameter 5
*DCALL (29 SENS COEF)        $ call subroutine SENS COEF

```

!NP = 0	
!N2 = 8	\$ parameter 6
*DCALL (29 SENS COEF)	\$ call subroutine SENS COEF
!NP = 0	
!N2 = 9	\$ parameter 7
*DCALL (29 SENS COEF)	\$ call subroutine SENS COEF
!NP = 10	
!N2 = 10	\$ parameter 8
*DCALL (29 SENS COEF)	\$ call subroutine SENS COEF
*DCALL (29 STAT ISTC)	\$ call subroutine STAT ISTC
*XQT EXIT	

Appendix I

EAL One-Dimensional Effective Insulated Model: Estimation Program

```
*XQT U1
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of the $
$                               variables used in the program such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $arithmetic utility system processor
$
!NY1 = 40           $number of nodes for the effective structure
!NY2 = 10           $number of nodes for the insulation
$
!EY1 = "NY1"-1      $number of elements for the effective structure
!EY2 = "NY2"-1      $number of elements for the insulation
$
!TOTN = "NY1"+"EY2" $total number of nodes of the model
!NON1 = "NY1"        $node number of the node closest to the heater
!NONI = "NY1"+"EY2"  $node number of the node on top of the insulation
!NONJ = FLOAT("NONI") $node number of the node on top of the insulation
$                   $as a real
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 GEAL VARI)GEAL: This subroutine defines some of $
$                               the variables used in the genetic $
$                               algorithm processors. $
```

```

$-----$
$
$(29 GEAL VARI) GEAL
*XQT AUS
$
TABLE(NI=1,NJ=9,TYPE=0) : INPT INTE
I=1
  J=1:    2    $ NC
  J=2:    5    $ NPOP
  J=3: 15287    $ CFSEED1
  J=4:  9428    $ CFSEED2
  J=5:    15    $ SLOPSEED
  J=6:    10    $ NBEST1
  J=7:    40    $ NBEST2
  J=8:    30    $ NBEST3
  J=9:    5     $ ITMX
$
$ NC          $ number of estimated parameters
$ NPOP        $ number of initial populations to investigate
$ CFSEED1     $ parameter used by the random number generator
$ CFSEED2     $ parameter used by the random number generator
$ SLOPSEED    $ parameter used by the random number generator
$ NBEST1      $ number of chromosomes kept from the NPOP initial
$             $ populations: elitist parameter
$ NBEST2      $ number of chromosomes kept from the combined parent
$             $ and children population:elitist parameter =<ns
$ NBEST3      $ number of chromosomes kept from the combined parent
$             $ and children population used to update the parameter
$             $ ranges:elitist parameter =<NBEST2
$ ITMX        $ maximum number of generations to perform
$
TABLE(NI=1,NJ=7,TYPE=-2) : INPT REAL
I=1
  J=1 : 5.
  J=2 : 1.
  J=3 : "CMIN"    $ PARMIN1 CP
  J=4 : "KMIN"    $ PARMIN2 K
  J=5 : "CMAX"    $ PARMAX1 CP
  J=6 : "KMAX"    $ PARMAX2 K
  J=7 : 0.05      $ PM
$
$ PARMIN1      $ lower limit of parameter 1
$ PARMIN2      $ lower limit of parameter 2
$ PARMAX1      $ upper limit of parameter 1
$ PARMAX2      $ upper limit of parameter 2
$ PARMAX7      $ upper limit of parameter 7
$ PM           $ probability of mutation
$
*RETURN
*GEAL
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the  $
$                               properties of the materials used  $
$                               in the model.                      $
$-----$
$
$(29 MATE PROP)PROP
$

```



```

*XQT AUS
$
$ conduction properties for effective structure
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
      J=1: 1.E+5 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
      J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
      J=1: 256.29 1046.64 0.0277 0.0277 0.0277
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$                               coordinates of the points used in $
$                               the geometry of the plate. The $
$                               points cited in the subroutine $
$                               describe the following $
$                               substructures: $
$                               1,2 :effective structure $
$                               2A,3 :insulation $
$-----$
$
$(29 PLAT DIME)DIME
$
!X1 = 0.          $ x-coordinate for point 1
!X2 = 0.          $ x-coordinate for point 2
!X3 = 0.          $ x-coordinate for point 3
$
!Y1 = 0.          $ y-coordinate for point 1
!Y2 = 0.1235      $ y-coordinate for point 2
!Y2A = 0.1235+(0.0254/"EY2") $ y-coordinate for point 2A
!Y3 = 0.1489      $ y-coordinate for point 3
$
!Z1 = 0.          $ z-coordinate for point 1
!Z2 = 0.          $ z-coordinate for point 2
!Z3 = 0.          $ z-coordinate for point 3
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the coordinates $
$                               of the nodes used in the mesh. $
$-----$
$
$(29 NODE GENE)GENE
$
!INCY = 1          $nodal increment in the y-direction
!T2 = 1 + "NY1"    $starting node number for insulation
$

```

```

*XQT TAB          $defines position coordinates of nodes
$
START "TOTN"      $define the total number of nodes
JLOC             $define the joint locations
FORMAT = 1       $specifies rectangular coordinates
$
1,"X1","Y1","Z1","X2","Y2","Z2","NY1","INCY"
*IF("EY2" GT 1):*GOTO 100
"T2","X3","Y3","Z3"
*GOTO 105
*LABEL 100
"T2","X2","Y2A","Z2","X3","Y3","Z3","EY2","INCY"
*LABEL 105
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the      $
$                               elements used in the mesh.        $
$-----$
$
*(29 MAKE ELEM)ELEM
$
*XQT AUS          $ arithmetic utility system processor
$
TABLE(NI=1,NJ=1): K AREA : I = 1 $ section property table for
                   J=1: 1.0      $ K21 conduction elements
$
*XQT ELD          $ element definition processor
RESET NUTED = 1   $ destination library
$
!NOPT = 1
!CLOS = 0
$
!J1 = 1           $ group 1 lower outer facesheet
!J2 = "J1"+"INCY" $ group 1 lower outer facesheet
!J3 = "J1"+"INCY"*"EY1" $ group 2 heater
!J4 = "J1"+"INCY"*"EY1" $ group 2 heater
!J5 = "J4"        $ group 3 insulation
!J6 = "J5"+"INCY" $ group 3 insulation
$
K21               $ 1-D conduction elements
$
GROUP = 1         $ group 1 lower outer facesheet
NMAT = 1          $ material 1 conduction properties 1
"J1","J2","NOPT","EY1",1
$
GROUP = 2         $ group 2 heater
NMAT = 2          $ material 2 conduction properties 2
"J3","J4","NOPT",1,1
$
GROUP = 3         $ group 3 insulation
NMAT = 3          $ material 3 conduction properties 3
"J5","J6","NOPT","EY2",1
$
*XQT TGEO        $ element geometry processor
$
*RETURN

```

```

*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the      $
$                               boundary conditions such as:        $
$                               - the constant temperature          $
$                               boundary                            $
$                               - the applied heat flux boundary    $
$                               - the total experimental and        $
$                               heating time                        $
$                               - the heated node boundary          $
$                               temperature                         $
$-----$
$
$(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, and
$ heat flux applied to the heated surface are extracted from the data file
$ INPT DATA and put in 4 different tables in library 5.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=3,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=3,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1

```

```

!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following 4 tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
$
DEFINE F = 5 DELT TIME
DEFINE G = 5 DELT CTMP
DEFINE H = 5 DELT HTMP
DEFINE I = 5 DELT FLUX
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 2
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=I,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=F,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=G,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$

```

```

TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=G,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
$
TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
  TRANSFER(SOUR=TB,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=F,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node number to which the constant
$ temperature is applied
$
TABLE(NI=1,NJ=2):5 TEMP NODE
  I=1
      J=1: 1.0
      J=2: "NONJ"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets
$
TOCC (5 SOUR K21 2):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 GENA PROC)GENA: This subroutine is the first $
$                               genetic processor that needs to be $
$                               executed. Its purpose is to read $
$                               the genetic variables defined in $
$                               subroutine GEAL-VARI and to create $
$                               the first parent population $
$                               containing the first 50 $
$                               chromosomes. More information $
$                               about the necessary input and $
$                               output tables as well as the RESET $
$                               is given in the README file in the $
$                               genetic directory. $
$-----$
$
$(29 GENA PROC)GENA
$
*XQT GENA
  RESET NTIM=1      $print reset for the input data in output file
  RESET NUIN=1      $reset for source library
  RESET NOUT=1      $reset for destination library

```

```

    RESET ITPP="ITPP"      $reset for looping process
$
*RETURN
*GENA
$
$-----$
$ Subroutine(29 GENB PROC)GENB: This is the subroutine is the second$
$                               genetic processor that needs to be $
$                               executed. It creates the elitist $
$                               parent population once the GENA $
$                               processor has been executed several $
$                               times.                               $
$-----$
$
$(29 GENB PROC)GENB
$
*XQT GENB
    RESET ITPP="ITPP"      $reset for looping process
    RESET NUIN=1           $reset for source library
    RESET NOUT=1           $reset for destination library
$
*RETURN
*GENB
$
$-----$
$ Subroutine(29 GENC PROC)GENC: This is the third processor that $
$                               needs to be executed. The objective $
$                               of this processor is to create a $
$                               children population using mutation, $
$                               cross-over and fitness criteria $
$                               applied to the elitist parent $
$                               population.                         $
$-----$
$
$(29 GENC PROC)GENC
$
*XQT GENC
    RESET ITPP="ITPP"      $reset for looping process
    RESET ITRG="ITERG"     $reset for generation iteration process
    RESET NUIN= 1           $reset for source library
    RESET NOUT= 1           $reset for destination library
$
*RETURN
*GENC
$
$-----$
$ Subroutine(29 GEND PROC)GEND: This is the fourth genetic processor$
$                               that needs to be executed. The goal $
$                               of this processor is to check for $
$                               twins and remove them, while $
$                               creating new candidate chromosomes. $
$-----$
$
$(29 GEND PROC)GEND
$
*XQT GEND
    RESET ITPP="ITPP"      $reset for looping process
    RESET ITRG="ITERG"     $reset for generation iteration process
    RESET NUIN= 1           $reset for source library

```

```

    RESET NOUT= 1          $reset for destination library
$
*RETURN
*GEND
$
$-----$
$ Subroutine(29 GENE PROC)GENF: This subroutine is genetic processor$
$                               four. Its intent is to create and $
$                               sort either the iteration population$
$                               or the final population depending on$
$                               whether the final number of $
$                               iterations has been achieved or not.$
$-----$
$
*(29 GENE PROC)GENF
$
*XQT GENE
    RESET ITPP="ITPP"      $reset for looping process
    RESET ITRG="ITERG"     $reset for generation iteration process
    RESET NUIN= 1          $reset for source library
    RESET NOUT= 1          $reset for destination library
$
*RETURN
*GENF
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem using $
$                               the TRTB processor. The TRTB processor $
$                               is an implicit transient analyzer which $
$                               can be modified by changing the Crank $
$                               Nicholson parameter. $
$-----$
$
*(29 TRAN ANAL)ANAL
$
*XQT TRTB                                $ implicit transient analyzer
    RESET QLIB = 5                      $ thermal excitation library
    RESET MLIB = 6                      $ material and fluids properties library
    RESET T1   = "TIMI"                 $ starting time of analysis
    RESET T2   = "TIMF"                 $ termination time of analysis
    RESET DT   = "INTS"                 $ integration time step size
    RESET DEST = 4                      $ output library
    RESET PRINT = 0                     $ online print control parameter
    RESET KPRT = 0                      $ print control parameter for k matrix
    RESET MXNDT = 10000000              $ maximum # of int. time steps during TRTB
    RESET BETA = 0.5                    $ Crank Nicholson parameter
    TEMP       = "CSTP"                 $ initial temperature of structure at T1
    TSAVE      = 5 EXPE TIME 1 1        $ storage of nodal temperatures at time
$                                       $ increments of TSAVE
*RETURN
*ANAL
$
$-----$
$ Subroutine (29 SUMO SQUA)SQUA: This subroutine performs a calculation$
$                               of the sum of the squares for the $
$                               different chromosomes and puts them $
$                               into the objective function table. $
$-----$
$

```

```

*(29 SUMO SQUA)SQUA
$
$the following if statements are used to define how far the runstream
$has gotten with respect to the overall genetic process
$
*IF("SSQU" EQ 1):*GOTO 8000          $chromosomes created by GENA
*IF("SSQU" EQ 2):*GOTO 8100          $chromosomes created by GEND
*IF("SSQU" EQ 3):*GOTO 8200          $chromosomes created by GENE
$
*LABEL 8000
*XQT AUS                             $arithmetic utility system processor
$
!LINE = 0                            $initializing counter
$
*XQT DCU                             $database utility function processor
!NWD = TOC,NJ(1 PARE POPU 1)         $determine number of rows in PARE POPU
*XQT AUS                             $arithmetic utility system processor
$
$the following two tables need to be initialized before the beginning of
$the loop, in order to use the update command TABLE,U, the tables are
$both filled with zeros. the first table will contain the value of the
$objective function, whereas the second represents the identity matrix
$
TABLE(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
J=1,"NDS": 1.0
$
*LABEL 105
$
!LINE = LINE + 1                     $increment of the counter
$
!A1 = DS,1,"LINE",1 (1 PARE POPU 1) $update the value of the property
!A2 = DS,2,"LINE",1 (1 PARE POPU 1) $update the value of the property
$
*DCALL (29 MATE PROP)                $call subroutine MATE PROP
*DCALL (29 TRAN ANAL)                $call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)                $call subroutine TEMP TABL
ERASE 6
$
INLIB = 3                            $change source library
OUTLI = 3                            $change destination library
$
DEFINE TO = 2 HEAT NODE 1 1          $define table as variable for arith. oper.
DEFINE TM = 5 EXPE HTMP 1 1          $define table as variable for arith. oper.
DEFINE ON = 7 ONES MATX 1 1          $define table as variable for arith. oper.
$
SU   = SUM(TM,-1. TO)                $(Y-ETA)
SSMX = XTY(SU,SU)                    $(Y-ETA)T(Y-ETA)
!SSVA = DS 1,1,1 (3 SSMX AUS 1 1)    $scalar of (Y-ETA)T(Y-ETA)
$
$the TABLE,U command below updates the table created before the loop and
$inserts the objective function that corresponds to the chromosomes
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
J="LINE":"SSVA"
$
*IF("LINE" LT "NWD"): *GOTO 105
$
*GOTO 8300

```



```

$
$-----$
$
*LABEL 8100
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 CHRO CHIL 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
J=1,"NDS": 1.0
$
*LABEL 106
$
!LINE = LINE + 1
$
!A1 = DS,1,"LINE",1 (1 CHRO CHIL 1)
!A2 = DS,2,"LINE",1 (1 CHRO CHIL 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1
DEFINE TM = 5 EXPE HTMP 1 1
DEFINE ON = 7 ONES MATX 1 1
$
SU = SUM(TM,-1. TO)
SSMX = XTY(SU,SU)
!SSVA = DS 1,1,1 (3 SSMX AUS 1 1)
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
J="LINE":"SSVA"
$
*IF("LINE" LT "NWD"):GOTO 106
$
*GOTO 8300
$
$-----$
$
*LABEL 8200
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 FRES BLOD 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1

```

```

J=1,"NDS": 1.0
$
*LABEL 107
$
!LINE = LINE + 1
$
!A1 = DS,1,"LINE",1 (1 FRES BLOD 1)
!A2 = DS,2,"LINE",1 (1 FRES BLOD 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1
DEFINE TM = 5 EXPE HTMP 1 1
DEFINE ON = 7 ONES MATX 1 1
$
SU      = SUM(TM,-1. TO)
SSMX    = XTY(SU,SU)
!SSVA = DS 1,1,1 (3 SSMX AUS 1 1)
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
J="LINE":"SSVA"
$
*IF("LINE" LT "NWD"):GOTO 107
$
*LABEL 8300
$
*RETURN
*SQUA
$
$-----$
$ Subroutine (29 ITER ATIO)ATIO: This subroutine updates the thermal      $
$                               properties of the parameters and          $
$                               executes the genetic processors n-times    $
$                               for n-generations.                         $
$-----$
$
*(29 ITER ATIO)ATIO
$
!ITERG = 0                                $initializing the counter for the loop
$
*LABEL 400
$
!ITERG = ITERG + 1                        $counter for the loop
!SSQU = 2                                $flag for the SUMO SQUA subroutine
$
*DCALL (29 GENC PROC)                     $call subroutine GENC PROC
*DCALL (29 SUMO SQUA))                   $call subroutine SUMO SQUA
ERASE 2                                  $erase content of library 2
ERASE 3                                  $erase content of library 3
ERASE 4                                  $erase content of library 4
ERASE 6                                  $erase content of library 6
!SSQU = 3                                $flag for the SUMO SQUA subroutine
*DCALL (29 GEND PROC)                     $call subroutine GEND PROC

```

```

*DCALL (29 SUMO SQUA)          $call subroutine SUMO SQUA
$
!CMIN = DS,1,1,1 (1 NEWW PARA) $reset the lower boundary for Cp
!KMIN = DS,1,2,1 (1 NEWW PARA) $reset the lower boundary for K
!CMAX = DS,1,3,1 (1 NEWW PARA) $reset the upper boundary for Cp
!KMAX = DS,1,4,1 (1 NEWW PARA) $reset the upper boundary for K
$
*DCALL (29 GEAL VARI)          $call the subroutine GEAL VARI
  ERASE 2                      $erase content of library 2
  ERASE 3                      $erase content of library 3
  ERASE 4                      $erase content of library 4
  ERASE 6                      $erase content of library 6
*DCALL (29 GENE PROC)          $call subroutine GENE PROC
$
*IF("ITERG" EQ "ITMX"): *GOTO 410
*GOTO 400
*LABEL 410
$
*RETURN
*ATIO
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.             $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0 $ Starting time of TRTB analysis (seconds)
!INTS = 0.5 $ Integrtral time step for TRTB (seconds)
$
!A1 = 3.194540 $ Forward model estimated value for (PCp)
!A2 = 0.617267 $ Forward model estimated value for (k)
$
!A1I = "A1" $ Original estimate for the heat capacity (PCp)
!A2I = "A2" $ Original estimate for the thermal conductivity (K)
$
!CMIN = 1.50 $ lower bound for genetic algorithm parameter (PCp)
!KMIN = 0.10 $ lower bound for genetic algorithm parameter (K)
!CMAX = 4.80 $ upper bound for genetic algorithm parameter (PCp)
!KMAX = 1.20 $ upper bound for genetic algorithm parameter (K)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                     $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1 $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1 $ copy the dataset from lib 4 to lib 2

```

```

$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                                $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME)           $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1                         $ heated node number - 1
!VAR3 = "TOTN"-1                         $ total node number - 1
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 1):*GOTO 5000
*IF("N2" EQ 2):*GOTO 5010
*IF("N2" EQ 3):*GOTO 5020
*IF("N2" EQ 4):*GOTO 5030
*IF("N2" EQ 5):*GOTO 5040
*IF("N2" EQ 6):*GOTO 5050
*IF("N2" EQ 7):*GOTO 5060
*IF("N2" EQ 8):*GOTO 5070
*IF("N2" EQ 9):*GOTO 5080
$
*LABEL 5000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
$
TABLE(NI=3,NJ="NDS"):9 TEMP ERAT 1 1
      TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
      TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
      TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
*XQT DCU
$ PRINT 5 TEMP ERAT 1 1
*XQT AUS
$
*GOTO 6000
$
*LABEL 5010
TABLE(NI=1,NJ="VAR1"):2 INIT OO 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1

```

```

        TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
        TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
        TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
        TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
        TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ This is the main program. All the subroutines must have been defined $
$ before they are called by the main program, therefore the main program $
$ needs to be placed at the end. $
$-----$
$
*XQT AUS
$
*TF OPEN 1'data
*TF READ 1
*TF CLOSE 1
$
!NP = 0
!N2 = 1 $counter to direct TRAN TEMP
$
*DCALL (29 MAKE VARI)
*DCALL (29 PROB SPEC)
*DCALL (29 GEAL VARI)
*DCALL (29 PLAT DIME)
*DCALL (29 NODE GENE)
*DCALL (29 MAKE ELEM)
*DCALL (29 BOUN COND)
!ITPP = 1
!SSQU = 1
*DCALL (29 GENA PROC)
*DCALL (29 SUMO SQUA)
*DCALL (29 GENB PROC)
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 2

```

```

*DCALL (29 GENA PROC)
*DCALL (29 SUMO SQUA)
*DCALL (29 GENB PROC)
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 3
*DCALL (29 GENA PROC)
*DCALL (29 SUMO SQUA)
*DCALL (29 GENB PROC)
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 4
*DCALL (29 GENA PROC)
*DCALL (29 SUMO SQUA)
*DCALL (29 GENB PROC)
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 5
*DCALL (29 GENA PROC)
*DCALL (29 SUMO SQUA)
*DCALL (29 GENB PROC)
ERASE 2
ERASE 3
ERASE 4
ERASE 6
$
!ITMX = 10
*DCALL (29 ITER ATIO)
$
*XQT DCU
  PRINT 1 ITER PAPO
  PRINT 1 ITER OBFN
*XQT EXIT

```

Appendix J

EAL One-Dimensional Convective Model: Estimation Program

```
*XQT U1
*ONLINE=0
!MAXCM = SSP(0,13)
*CMPARA(MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of the $
$                               variables used in the program such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $ arithmetic utility system processor
$
!NY1  = 40          $ number of nodes for the effective structure
!NY2  = 10          $ number of nodes for the insulation
$
!EY1  ="NY1"-1      $ number of elements for the effective structure
!EY2  ="NY2"-1      $ number of elements for the insulation
$
!TOTN  = "NY1"+"EY2" $ total number of nodes of the model
!NON1  = "NY1"       $ node number of the node closest to the heater
!NONI  = "NY1"+"EY2" $ node number of the node on top of the insulation
!NONJ  = FLOAT("NONI") $ node number of the node on top of the insulation
$                               $ as a real
$
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 GEAL VARI)GEAL: This subroutine defines some of $
$                               the variables used in the genetic $
$                               algorithm processors. $
```

```

$-----$
$
*(29 GEAL VARI) GEAL
*XQT AUS
$
TABLE(NI=1,NJ=9,TYPE=0) : INPT INTE
I=1
  J=1:   10      $ NC      : number of estimated parameters
  J=2:    5      $ NPOP    : number of initial populations to investigate
  J=3: 15287     $ CFSEED1 : parameter used by random number generator
  J=4:  9428     $ CFSEED2 : parameter used by random number generator
  J=5:   15      $ SLOPSEED : parameter used by random number generator
  J=6:   10      $ NBEST1   : # chro. kept from NPOP initial population
  J=7:   40      $ NBEST2   : # chro. kept from parent children population
  J=8:   30      $ NBEST3   : # chro. kept from parent children population
  J=9:   10      $ ITMX     : maximum number of generations to perform
$
TABLE(NI=1,NJ=21,TYPE=-2) : INPT REAL
I=1
  J=1 : "MIN1"    $ PARMIN1 : CP lower limit for skins
  J=2 : "MIN2"    $ PARMIN2 : K lower limit for skins
  J=3 : "MIN3"    $ PARMIN3 : CP lower limit for honeycomb
  J=4 : "MIN4"    $ PARMIN4 : K lower limit for honeycomb
  J=5 : "MIN5"    $ PARMIN5 : CP lower limit for torque tubes
  J=6 : "MIN6"    $ PARMIN6 : K lower limit for torque tubes
  J=7 : "MIN7"    $ PARMIN7 : CP lower limit for outer facesheets
  J=8 : "MIN8"    $ PARMIN8 : K lower limit for outer facesheets
  J=9 : "MIN9"    $ PARMIN9 : CP lower limit for air gap
  J=10: "MINX"    $ PARMINX : K lower limit for air gap
  J=11: "MAX1"    $ PARMAX1 : CP upper limit for skins
  J=12: "MAX2"    $ PARMAX2 : K upper limit for skins
  J=13: "MAX3"    $ PARMAX3 : CP upper limit for honeycomb
  J=14: "MAX4"    $ PARMAX4 : K upper limit for honeycomb
  J=15: "MAX5"    $ PARMAX5 : CP upper limit for torque tubes
  J=16: "MAX6"    $ PARMAX6 : K upper limit for torque tubes
  J=17: "MAX7"    $ PARMAX7 : CP upper limit for outer facesheets
  J=18: "MAX8"    $ PARMAX8 : K upper limit for outer facesheets
  J=19: "MAX9"    $ PARMAX9 : CP upper limit for air gap
  J=20: "MAXX"    $ PARMAX10 : K upper limit for air gap
  J=21: 0.05      $ PM      : probability of mutation
$
*RETURN
*GEAL
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the      $
$                               properties of the materials used    $
$                               in the model.                       $
$-----$
$
*(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for effective structure
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
      J=1: 1.E+5 "A1" "A2"
$

```



```

$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
      J=1: 0. 0. 0. 0.

$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
      J=1: 256.29 1046.64 0.02254 0.02254 0.02254

$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$                               coordinates of the points used in $
$                               the geometry of the plate. The $
$                               points cited in the subroutine $
$                               describe the following $
$                               substructures: $
$                               1,2 :effective structure $
$                               2A,3 :insulation $
$-----$
$
*(29 PLAT DIME)DIME
$
!X1 = 0.          $ x-coordinate for point 1
!X2 = 0.          $ x-coordinate for point 2
!X3 = 0.          $ x-coordinate for point 3
$
!Y1 = 0.          $ y-coordinate for point 1
!Y2 = 0.1235      $ y-coordinate for point 2
!Y2A = 0.1235+(0.0254/"EY2") $ y-coordinate for point 2A
!Y3 = 0.1489      $ y-coordinate for point 3
$
!Z1 = 0.          $ z-coordinate for point 1
!Z2 = 0.          $ z-coordinate for point 2
!Z3 = 0.          $ z-coordinate for point 3
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the coordinates $
$                               of the nodes used in the mesh. $
$-----$
$
*(29 NODE GENE)GENE
$
!INCY = 1          $nodal increment in the y-direction
!T2 = 1 + "NY1"    $starting node number for insulation
$
*XQT TAB          $defines position coordinates of nodes
$
START "TOTN"      $define the total number of nodes
JLOC              $define the joint locations
FORMAT = 1        $specifies rectangular coordinates
$
1,"X1","Y1","Z1","X2","Y2","Z2","NY1","INCY"

```

```

*IF("EY2" GT 1):*GOTO 100
"T2","X3","Y3","Z3"
*GOTO 105
*LABEL 100
"T2","X2","Y2A","Z2","X3","Y3","Z3","EY2","INCY"
*LABEL 105
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the $
$ elements used in the mesh. $
$-----$
$
*(29 MAKE ELEM)ELEM
$
*XQT AUS $ arithmetic utility system processor
$
TABLE(NI=1,NJ=1): K AREA : I = 1 $ section property table for
J=1: 1.0 $ K21 conduction elements
$
*XQT ELD $ element definition processor
RESET NUTED = 1 $ destination library
$
!NOPT = 1
!CLOS = 0
$
!J1 = 1 $ group 1 lower outer facesheet
!J2 = "J1"+"INCY" $ group 1 lower outer facesheet
!J3 = "J1"+"INCY"*"EY1" $ group 2 heater
!J4 = "J1"+"INCY"*"EY1" $ group 2 heater
!J5 = "J4" $ group 3 insulation
!J6 = "J5"+"INCY" $ group 3 insulation
$
K21 $ 1-D conduction elements
$
GROUP = 1 $ group 1 lower outer facesheet
NMAT = 1 $ material 1 conduction properties 1
"J1","J2","NOPT","EY1",1
$
GROUP = 2 $ group 2 heater
NMAT = 2 $ material 2 conduction properties 2
"J3","J4","NOPT",1,1
$
GROUP = 3 $ group 3 insulation
NMAT = 3 $ material 3 conduction properties 3
"J5","J6","NOPT","EY2",1
$
*XQT TGEO $ element geometry processor
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the $
$ boundary conditions such as: $
$ - the constant temperature $
$ boundary $

```

```

$                                - the applied heat flux boundary  $
$                                - the total experimental and      $
$                                heating time                      $
$                                - the heated node boundary        $
$                                temperature                       $
$-----$
$
$(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the
$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TOPT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BOTB 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=5,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTB 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$

```

```

!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!TOPT = DS 1,"CNTR",1 (5 EXPE TOPT 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
TABLE(NI=1,NJ=1):5 DELT TOPT:I=1:J=1:"TOPT"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT HTMP
DEFINE EE = 5 DELT FLUX
DEFINE FF = 5 DELT TOPT
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 2
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=EE,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation

```

```

$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM3
  TRANSFER(SOUR=F,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=FF,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=F,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
DEFINE TC = 5 APPL TEM3
$
$TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
  TRANSFER(SOUR=TB,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
TABLE(NI=1,NJ=2):5 TEMP NODE
  I=1
      J=1: 1.0
      J=2: "NONJ"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 2):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 GENA PROC)GENA: This subroutine is the first $
$                               genetic processor that needs to be $
$                               executed. Its purpose is to read $
$                               the genetic variables defined in $
$                               subroutine GEAL-VARI and to create $
$                               the first parent population $
$                               containing the first 50 $
$                               chromosomes. More information $

```

```

$                                about the necessary input and $
$                                output tables as well as the RESET $
$                                is given in the README file in the $
$                                genetic directory. $
$-----$
$
$(29 GENA PROC)GENA
$
*XQT GENA
  RESET NTIM=1      $print reset for the input data in output file
  RESET NUIN=1      $reset for source library
  RESET NOUT=1      $reset for destination library
  RESET ITPP="ITPP" $reset for looping process
$
*RETURN
*GENA
$
$-----$
$ Subroutine(29 GENB PROC)GENB: This is the subroutine is the second$
$                                genetic processor that needs to be $
$                                executed. It creates the elitist $
$                                parent population once the GENA $
$                                processor has been executed several $
$                                times. $
$-----$
$
$(29 GENB PROC)GENB
$
*XQT GENB
  RESET ITPP="ITPP" $reset for looping process
  RESET NUIN=1      $reset for source library
  RESET NOUT=1      $reset for destination library
$
*RETURN
*GENB
$
$-----$
$ Subroutine(29 GENC PROC)GENC: This is the third processor that $
$                                needs to be executed. The objective $
$                                of this processor is to create a $
$                                children population using mutation, $
$                                cross-over and fitness criteria $
$                                applied to the elitist parent $
$                                population. $
$-----$
$
$(29 GENC PROC)GENC
$
*XQT GENC
  RESET ITPP="ITPP" $reset for looping process
  RESET ITRG="ITER" $reset for generation iteration process
  RESET NUIN= 1     $reset for source library
  RESET NOUT= 1     $reset for destination library
$
*RETURN
*GENC
$
$-----$
$ Subroutine(29 GEND PROC)GEND: This is the fourth genetic processor$

```

```

$                                that needs to be executed. The goal $
$                                of this processor is to check for $
$                                twins and remove them, while $
$                                creating new candidate chromosomes. $
$-----$
$
$(29 GEND PROC)GEND
$
*XQT GEND
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITER"  $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GEND
$
$-----$
$ Subroutine(29 GENE PROC)GENF: This subroutine is genetic processor$
$                                four. Its intent is to create and $
$                                sort either the iteration population$
$                                or the final population depending on$
$                                whether the final number of $
$                                iterations has been achieved or not.$
$-----$
$
$(29 GENE PROC)GENF
$
*XQT GENE
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITER"  $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GENF
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                                using the TRTB processor. The TRTB $
$                                processor is an implicit transient $
$                                analyzer that can be modified by $
$                                changing the Crank Nicholson para- $
$                                meter. $
$-----$
$
$(29 TRAN ANAL)ANAL
$
*XQT TRTB
  RESET QLIB = 5      $ implicit transient analyzer
  RESET MLIB = 6      $ thermal excitation library
  RESET T1 = "TIMI"   $ material and fluids properties library
  RESET T2 = "TIMF"   $ starting time of analysis
  RESET DT = "INTS"   $ termination time of analysis
  RESET DEST = 4      $ integration time step size
  RESET PRINT = 0     $ output library
  RESET KPRT = 0      $ online print control parameter
  RESET MXNDT = 1000000 $ print control parameter for k matrix
  RESET BETA = 0.5    $ maximum # of int. time steps during TRTB
                    $ Crank Nicholson parameter

```

```

RESET KTI    = 1                $ K matrix computation time parameter
KTIME       = 10000.            $ times to calculate the conduction load vector
TEMP        = "CSTP"            $ initial temperature of structure at T1
TSAVE       = 5 EXPE TIME 1 1   $ storage of nodal temperatures at time
$                               $ increments of TSAVE

*RETURN
*ANAL
$
$-----$
$ Subroutine (29 SUMO SQUA)SQUA: This subroutine performs a      $
$                               calculation of the sum of the    $
$                               sum of squares for the different  $
$                               chromosomes and puts them into the $
$                               objective function table.         $
$-----$
$
$(29 SUMO SQUA)SQUA
$
$ the following if statements are used to define how far the runstream
$ has gotten with respect to the overall genetic process
$
*IF("SSQU" EQ 1):*GOTO 8000      $ chromosomes created by GENA
*IF("SSQU" EQ 2):*GOTO 8100      $ chromosomes created by GEND
*IF("SSQU" EQ 3):*GOTO 8200      $ chromosomes created by GENE
$
*LABEL 8000
*XQT AUS                         $ arithmetic utility system processor
$
!LINE = 0                        $ initializing counter
$
*XQT DCU                         $ database utility function processor
!NWD = TOC,NJ(1 PARE POPU 1)     $ determine number of rows in PARE POPU
*XQT AUS                         $ arithmetic utility system processor
$
$ the following two tables need to be initialized before the beginning of
$ the loop, in order to use the update command TABLE,U, the tables are
$ both filled with zeros. the first table will contain the value of the
$ objective function, whereas the second represents the identity matrix
$
TABLE(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
J=1,"NDS": 1.0
$
*LABEL 105
$
!LINE = LINE + 1                 $ increment of the counter
$
!A1 = DS,1 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A2 = DS,2 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A3 = DS,3 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A4 = DS,4 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A5 = DS,5 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A6 = DS,6 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A7 = DS,7 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A8 = DS,8 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A9 = DS,9 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A10 = DS,10,"LINE",1 (1 PARE POPU 1) $ update the value of the property
$
*DCALL (29 MATE PROP)            $ call subroutine MATE PROP

```



```

*DCALL (29 TRAN ANAL)          $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)          $ call subroutine TEMP TABL
ERASE 6
$
INLIB = 3                      $ change source library
OUTLI = 3                      $ change destination library
$
DEFINE TO = 2 HEAT NODE 1 1    $ define table as variable for arith. oper.
$DEFINE TP = 2 TOPT NODE 1 1   $ define table as variable for arith. oper.
$DEFINE TQ = 2 BOTI NODE 1 1   $ define table as variable for arith. oper.
DEFINE TL = 5 EXPE HTMP 1 1    $ define table as variable for arith. oper.
$DEFINE TM = 5 EXPE TOPT 1 1   $ define table as variable for arith. oper.
$DEFINE TN = 5 EXPE BOTI 1 1   $ define table as variable for arith. oper.
DEFINE ON = 7 ONES MATX 1 1    $ define table as variable for arith. oper.
$
SU1  = SUM(TL,-1. TO)          $ (Y1-ETA1)
$SU2  = SUM(TM,-1. TP)         $ (Y2-ETA2)
$SU3  = SUM(TN,-1. TQ)         $ (Y3-ETA3)
SS1X = XTY(SU1,SU1)            $ (Y1-ETA1)T(Y1-ETA1)
$SS2X = XTY(SU2,SU2)           $ (Y2-ETA2)T(Y2-ETA2)
$SS3X = XTY(SU3,SU3)           $ (Y3-ETA3)T(Y3-ETA3)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1) $ scalar of (Y1-ETA1)T(Y1-ETA1)
$!SS2A = DS 1,1,1 (3 SS2X AUS 1 1) $ scalar of (Y2-ETA2)T(Y2-ETA2)
$!SS3A = DS 1,1,1 (3 SS3X AUS 1 1) $ scalar of (Y3-ETA3)T(Y3-ETA3)
$!SSTA = "SS1A"+"SS2A"+"SS3A"    $ total scalar value of the objective function
!SSTA = "SS1A"
$
$ the TABLE,U command below updates the table created before the loop and
$ inserts the objective function that corresponds to the chromosomes
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
  J="LINE": "SSTA"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 105
$
*GOTO 8300
$
$-----$
$
*LABEL 8100
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 CHRO CHIL 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 106
$
!LINE = LINE + 1
$
!A1 = DS,1 , "LINE",1 (1 CHRO CHIL 1)

```

```

!A2 = DS,2,"LINE",1 (1 CHRO CHIL 1)
!A3 = DS,3,"LINE",1 (1 CHRO CHIL 1)
!A4 = DS,4,"LINE",1 (1 CHRO CHIL 1)
!A5 = DS,5,"LINE",1 (1 CHRO CHIL 1)
!A6 = DS,6,"LINE",1 (1 CHRO CHIL 1)
!A7 = DS,7,"LINE",1 (1 CHRO CHIL 1)
!A8 = DS,8,"LINE",1 (1 CHRO CHIL 1)
!A9 = DS,9,"LINE",1 (1 CHRO CHIL 1)
!A10 = DS,10,"LINE",1 (1 CHRO CHIL 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1      $ define table as variable for arith. oper.
$DEFINE TP = 2 TOPT NODE 1 1      $ define table as variable for arith. oper.
$DEFINE TQ = 2 BOTI NODE 1 1      $ define table as variable for arith. oper.
DEFINE TL = 5 EXPE HTMP 1 1      $ define table as variable for arith. oper.
$DEFINE TM = 5 EXPE TOPT 1 1      $ define table as variable for arith. oper.
$DEFINE TN = 5 EXPE BOTI 1 1      $ define table as variable for arith. oper.
DEFINE ON = 7 ONES MATX 1 1      $ define table as variable for arith. oper.
$
SU1 = SUM(TL,-1. TO)             $ (Y1-ETA1)
$SU2 = SUM(TM,-1. TP)             $ (Y2-ETA2)
$SU3 = SUM(TN,-1. TQ)             $ (Y3-ETA3)
SS1X = XTY(SU1,SU1)              $ (Y1-ETA1)T(Y1-ETA1)
$SS2X = XTY(SU2,SU2)              $ (Y2-ETA2)T(Y2-ETA2)
$SS3X = XTY(SU3,SU3)              $ (Y3-ETA3)T(Y3-ETA3)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1) $ scalar of (Y1-ETA1)T(Y1-ETA1)
$!SS2A = DS 1,1,1 (3 SS2X AUS 1 1) $ scalar of (Y2-ETA2)T(Y2-ETA2)
$!SS3A = DS 1,1,1 (3 SS3X AUS 1 1) $ scalar of (Y3-ETA3)T(Y3-ETA3)
$!SSTA = "SS1A"+"SS2A"+"SS3A"    $ total scalar value of the objective function
!SSTA = "SS1A"
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
J="LINE": "SSTA"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 106
$
*GOTO 8300
$
$-----$
$
*LABEL 8200
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 FRES BLOD 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN BLOD 1 1

```

```

TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 107
$
!LINE = LINE + 1
$
!A1 = DS,1 ,"LINE",1 (1 FRES BLOD 1)
!A2 = DS,2 ,"LINE",1 (1 FRES BLOD 1)
!A3 = DS,3 ,"LINE",1 (1 FRES BLOD 1)
!A4 = DS,4 ,"LINE",1 (1 FRES BLOD 1)
!A5 = DS,5 ,"LINE",1 (1 FRES BLOD 1)
!A6 = DS,6 ,"LINE",1 (1 FRES BLOD 1)
!A7 = DS,7 ,"LINE",1 (1 FRES BLOD 1)
!A8 = DS,8 ,"LINE",1 (1 FRES BLOD 1)
!A9 = DS,9 ,"LINE",1 (1 FRES BLOD 1)
!A10 = DS,10,"LINE",1 (1 FRES BLOD 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1
$DEFINE TP = 2 TOPT NODE 1 1
$DEFINE TQ = 2 BOTT NODE 1 1
DEFINE TL = 5 EXPE HTMP 1 1
$DEFINE TM = 5 EXPE TOPT 1 1
$DEFINE TN = 5 EXPE BOTT 1 1
DEFINE ON = 7 ONES MATX 1 1
$
SU1 = SUM(TL,-1. TO)
$SU2 = SUM(TM,-1. TP)
$SU3 = SUM(TN,-1. TQ)
SS1X = XTY(SU1,SU1)
$SS2X = XTY(SU2,SU2)
$SS3X = XTY(SU3,SU3)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1)
$!SS2A = DS 1,1,1 (3 SS2X AUS 1 1)
$!SS3A = DS 1,1,1 (3 SS3X AUS 1 1)
$!SSTA = "SS1A"+"SS2A"+"SS3A"
!SSTA = "SS1A"
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
  J="LINE": "SSTA"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 107
$
*LABEL 8300
$
*RETURN
*SQUA
$
$-----$

```

```

$ Subroutine (29 ITER ATIO)ATIO: This subroutine updates the thermal$
$                               properties of the parameters and $
$                               executes the genetic processors $
$                               n-times for n-generations. $
$-----$
$
$(29 ITER ATIO)ATIO
$
!ITER = 0                      $ initializing the counter for the loop
$
TABLE(NI=1,NJ=50):8 CONV CRIT 1 1 $ initializing the table containing the
I=:J=1,50:0.0                  $ average of the first 20 best objective
$                               $ functions per iteration
$
*LABEL 400
$
!ITER = ITER + 1                $ counter for the loop
$
!SSQU = 2                       $ flag for the SUMO SQUA subroutine
$
*DCALL (29 GENC PROC)           $ call subroutine GENC PROC
*DCALL (29 SUMO SQUA))          $ call subroutine SUMO SQUA
  ERASE 2                       $ erase content of library 2
  ERASE 3                       $ erase content of library 3
  ERASE 4                       $ erase content of library 4
  ERASE 6                       $ erase content of library 6
$
!SSQU = 3                       $ flag for the SUMO SQUA subroutine
$
*DCALL (29 GEND PROC)           $ call subroutine GEND PROC
*DCALL (29 SUMO SQUA)           $ call subroutine SUMO SQUA
$
!MIN1 = DS,1,1,1 (1 NEWW PARA)  $ reset the lower boundary for Cp
!MIN2 = DS,1,2,1 (1 NEWW PARA)  $ reset the lower boundary for K
!MIN3 = DS,1,3,1 (1 NEWW PARA)  $ reset the lower boundary for Cp
!MIN4 = DS,1,4,1 (1 NEWW PARA)  $ reset the lower boundary for K
!MIN5 = DS,1,5,1 (1 NEWW PARA)  $ reset the lower boundary for Cp
!MIN6 = DS,1,6,1 (1 NEWW PARA)  $ reset the lower boundary for K
!MIN7 = DS,1,7,1 (1 NEWW PARA)  $ reset the lower boundary for Cp
!MIN8 = DS,1,8,1 (1 NEWW PARA)  $ reset the lower boundary for K
!MIN9 = DS,1,9,1 (1 NEWW PARA)  $ reset the lower boundary for Cp
!MINX = DS,1,10,1 (1 NEWW PARA) $ reset the lower boundary for K
!MAX1 = DS,1,11,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX2 = DS,1,12,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX3 = DS,1,13,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX4 = DS,1,14,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX5 = DS,1,15,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX6 = DS,1,16,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX7 = DS,1,17,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX8 = DS,1,18,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX9 = DS,1,19,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAXX = DS,1,20,1 (1 NEWW PARA) $ reset the upper boundary for K
$
*DCALL (29 GEAL VARI)           $call the subroutine GEAL VARI
  ERASE 2                       $erase content of library 2
  ERASE 3                       $erase content of library 3
  ERASE 4                       $erase content of library 4
  ERASE 6                       $erase content of library 6
*DCALL (29 GENE PROC)           $call subroutine GENE PROC
$

```

```

*XQT AUS
$
!OB1 = DS 1,1 ,1 (1 ITER OBFN 1 1)
!OB2 = DS 1,2 ,1 (1 ITER OBFN 1 1)
!OB3 = DS 1,3 ,1 (1 ITER OBFN 1 1)
!OB4 = DS 1,4 ,1 (1 ITER OBFN 1 1)
!OB5 = DS 1,5 ,1 (1 ITER OBFN 1 1)
!OB6 = DS 1,6 ,1 (1 ITER OBFN 1 1)
!OB7 = DS 1,7 ,1 (1 ITER OBFN 1 1)
!OB8 = DS 1,8 ,1 (1 ITER OBFN 1 1)
!OB9 = DS 1,9 ,1 (1 ITER OBFN 1 1)
!OB10 = DS 1,10 ,1 (1 ITER OBFN 1 1)
!OB11 = DS 1,11 ,1 (1 ITER OBFN 1 1)
!OB12 = DS 1,12 ,1 (1 ITER OBFN 1 1)
!OB13 = DS 1,13 ,1 (1 ITER OBFN 1 1)
!OB14 = DS 1,14 ,1 (1 ITER OBFN 1 1)
!OB15 = DS 1,15 ,1 (1 ITER OBFN 1 1)
!OB16 = DS 1,16 ,1 (1 ITER OBFN 1 1)
!OB17 = DS 1,17 ,1 (1 ITER OBFN 1 1)
!OB18 = DS 1,18 ,1 (1 ITER OBFN 1 1)
!OB19 = DS 1,19 ,1 (1 ITER OBFN 1 1)
!OB20 = DS 1,20 ,1 (1 ITER OBFN 1 1)
$
!OB1
!OB2
!OB3
!OB4
!OB5
!OB6
!OB7
!OB8
!OB9
!OB10
!OB11
!OB12
!OB13
!OB14
!OB15
!OB16
!OB17
!OB18
!OB19
!OB20
$
!OBF1 = ("OB1" +"OB2" +"OB3" +"OB4" +"OB5" )
!OBF2 = ("OB6" +"OB7" +"OB8" +"OB9" +"OB10")
!OBF3 = ("OB11"+"OB12"+"OB13"+"OB14"+"OB15")
!OBF4 = ("OB16"+"OB17"+"OB18"+"OB19"+"OB20")
$
!OBF = (( "OBF1"+"OBF2"+"OBF3"+"OBF4" )/20)
$
TABLE,U(NI=1,NJ=100):8 CONV CRIT 1 1
  I=1
  J="ITER": "OBF"
$
*XQT DCU
PRINT 8 CONV CRIT 1 1
*XQT AUS
$

```

```

*IF("ITER" GE 5):*GOTO 500
*GOTO 510
*LABEL 500
$
!NUM1 = "ITER" - 2
!NUM2 = "ITER" - 1
!NUM3 = "ITER"
!CON1 = DS 1,"NUM1",1 (8 CONV CRIT 1 1)
!CON2 = DS 1,"NUM2",1 (8 CONV CRIT 1 1)
!CON3 = DS 1,"NUM3",1 (8 CONV CRIT 1 1)
$
!CON4 = ("CON1"- "CON2")/"CON1"
!CON5 = ("CON2"- "CON3")/"CON2"
$
*XQT DCU
PRINT 1 ITER PAPO
PRINT 1 ITER OBFN
*XQT AUS
$
*IF("CON4" LE 0.01):*GOTO 520
*GOTO 510
$
*LABEL 520
*IF("CON5" LE 0.01):*GOTO 410
*GOTO 510
$
*LABEL 510
*IF("ITER" EQ "ITMX"):*GOTO 410
*GOTO 400
*LABEL 410
$
*RETURN
*ATIO
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.           $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrta time step for TRTB (seconds)
$
!MIN1 = 1.0      $ lower bound for skins (PCp)
!MIN2 = 0.1      $ lower bound for skins (K)
!MIN3 = 10.0     $ lower bound for honeycomb (PCp)
!MIN4 = 10.0     $ lower bound for honeycomb (K)
!MIN5 = 10.0     $ lower bound for torque tubes (PCp)
!MIN6 = 10.0     $ lower bound for torque tubes (K)
!MIN7 = 10.0     $ lower bound for facesheets (PCp)
!MIN8 = 10.0     $ lower bound for facesheets (K)
!MIN9 = 10.0     $ lower bound for air gap (PCp)
!MINX = 10.0     $ lower bound for air gap (K)
!MAX1 = 5.0      $ upper bound for skins (PCp)
!MAX2 = 2.0      $ upper bound for skins (K)
!MAX3 = 10.0     $ upper bound for honeycomb (PCp)
!MAX4 = 10.0     $ upper bound for honeycomb (K)

```

```

!MAX5 = 10.0      $ upper bound for torque tubes (PCp)
!MAX6 = 10.0      $ upper bound for torque tubes (K)
!MAX7 = 10.0      $ upper bound for facesheets (PCp)
!MAX8 = 10.0      $ upper bound for facesheets (K)
!MAX9 = 10.0      $ upper bound for air gap (PCp)
!MAXX = 10.0      $ upper bound for air gap (K)
$
!A1  = 2.54      $ Initial value for skins (PCp)
!A2  = 1.00      $ Initial value for skins (K)
!A3  = 1.00      $ Initial value for honeycomb (PCp)
!A4  = 1.00      $ Initial value for honeycomb (K)
!A5  = 1.00      $ Initial value for torque tubes (PCp)
!A6  = 1.00      $ Initial value for torque tubes (K)
!A7  = 1.00      $ Initial value for facesheets (PCp)
!A8  = 1.00      $ Initial value for facesheets (K)
!A9  = 1.00      $ Initial value for air gap (PCp)
!A10 = 1.00      $ Initial value for air gap (K)
$
!A1I = 2.54      $ Initial value for skins (PCp)
!A2I = 1.00      $ Initial value for skins (K)
!A3I = 1.00      $ Initial value for honeycomb (PCp)
!A4I = 1.00      $ Initial value for honeycomb (K)
!A5I = 1.00      $ Initial value for torque tubes (PCp)
!A6I = 1.00      $ Initial value for torque tubes (K)
!A7I = 1.00      $ Initial value for facesheets (PCp)
!A8I = 1.00      $ Initial value for facesheets (K)
!A9I = 1.00      $ Initial value for air gap (PCp)
!A10I = 1.00     $ Initial value for air gap (K)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU                $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1    $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1    $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME) $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1          $ heated node number - 1
!VAR3 = "TOTN"-1          $ total node number - 1
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 1):*GOTO 5000 $ Temperature histories for top,bot,heat nodes

```

```

*IF("N2" EQ 2):*GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3):*GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4):*GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5):*GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6):*GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7):*GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8):*GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9):*GOTO 5080 $ Temperature histories for property 7
$
*LABEL 5000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
$*XQT DCU
$ PRINT 5 TEMP ERAT 1 1
$*XQT AUS
$
*GOTO 6000
$
*LABEL 5010
TABLE(NI=1,NJ="VAR1"):2 INIT OO 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5070

```



```

TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
  TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS
$
*TF OPEN 1'data
*TF READ 1
*TF CLOSE 1
$
!N2 = 1
$
*DCALL (29 MAKE VARI) $ Call subroutine MAKE VARI
*DCALL (29 PROB SPEC) $ Call subroutine PROB SPEC
*DCALL (29 GEAL VARI) $ Call subroutine GEAL VARI
*DCALL (29 PLAT DIME) $ Call subroutine PLAT DIME
*DCALL (29 NODE GENE) $ Call subroutine NODE GENE
*DCALL (29 MAKE ELEM) $ Call subroutine MAKE ELEM
*DCALL (29 BOUN COND) $ Call subroutine BOUN COND
!ITPP = 1 $ Reset parameter for initial population 1
!SSQU = 1
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 2
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 3
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6

```

```

!ITPP = 4
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 5
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
$
!ITMX = 50           $ Maximum number of iteration
*DCALL (29 ITER ATIO) $ Call subroutine ITER ATIO
$
*XQT DCU
  PRINT 1 ITER PAPO
  PRINT 1 ITER OBFN
  PRINT 8 CONV CRIT
*XQT EXIT

```

Appendix K

EAL One-Dimensional Detailed Model: Estimation Program

```
*XQT U1
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of $
$                               the variables used in the program $
$                               such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each $
$                               section $
$-----$
$
*(29 MAKE VARI)VARI
$
*XQT AUS           $arithmetic utility system processor
$
!NY1 = 5           $number of nodes for the lower outer facesheet
!NY2 = 15          $number of nodes for the lower honeycomb structure
!NY3 = 5           $number of nodes for the lower inner facesheet
!NY4 = 5           $number of nodes for the lower torque tube
!NY5 = 30          $number of nodes for the air cavity
!NY6 = 5           $number of nodes for the upper torque tube
!NY7 = 5           $number of nodes for the upper inner facesheet
!NY8 = 15          $number of nodes for the upper honeycomb structure
!NY9 = 5           $number of nodes for the upper outer facesheet
!NY10 = 10         $number of nodes for the insulation
$
!EY1 ="NY1"-1      $number of elements for the lower outer facesheet
!EY2 ="NY2"-1      $number of elements for the lower honeycomb structure
!EY3 ="NY3"-1      $number of elements for the lower inner facesheet
!EY4 ="NY4"-1      $number of elements for the lower torque tube
!EY5 ="NY5"-1      $number of elements for the air cavity
!EY6 ="NY6"-1      $number of elements for the upper torque tube
!EY7 ="NY7"-1      $number of elements for the upper inner facesheet
!EY8 ="NY8"-1      $number of elements for the upper honeycomb structure
!EY9 ="NY9"-1      $number of elements for the upper outer facesheet
!EY10 ="NY10"-1    $number of elements for the insulation
```

```

$
!TOTN = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"+"EY10"
$
$total number of nodes in the model
!NON1 = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"
$
$node number at the heater (high sensitivity)
!NON2 = "NY1"+"EY2"+"EY3"+"EY4"
$
$node number at the inside of the lower torque tube
!NON3 = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"
$
$node number at the inside of the upper torque tube
!NONJ = FLOAT("TOTN")$node number on top of the insulation
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 GEAL VARI)GEAL: This subroutine defines some of $
$ the variables used in the genetic $
$ algorithm processors. $
$-----$
$
*(29 GEAL VARI) GEAL
*XQT AUS
$
TABLE(NI=1,NJ=9,TYPE=0) : INPT INTE
I=1
J=1: 7 $ NC : number of estimated parameters
J=2: 5 $ NPOP : number of initial populations to investigate
J=3: 15287 $ CFSEED1 : parameter used by random number generator
J=4: 9428 $ CFSEED2 : parameter used by random number generator
J=5: 15 $ SLOPSEED : parameter used by random number generator
J=6: 10 $ NBEST1 : # chro. kept from NPOP initial population
J=7: 40 $ NBEST2 : # chro. kept from parent children population
J=8: 30 $ NBEST3 : # chro. kept from parent children population
J=9: 10 $ ITMX : maximum number of generations to perform
$
TABLE(NI=1,NJ=15,TYPE=-2) : INPT REAL
I=1
J=1 : "MIN1" $ PARMIN1 : CP lower limit for skins
J=2 : "MIN2" $ PARMIN2 : K lower limit for skins
J=3 : "MIN3" $ PARMIN3 : CP lower limit for honeycomb
J=4 : "MIN4" $ PARMIN4 : K lower limit for honeycomb
J=5 : "MIN5" $ PARMIN5 : CP lower limit for torque tubes
J=6 : "MIN6" $ PARMIN6 : K lower limit for torque tubes
J=7 : "MIN7" $ PARMIN7 : CP lower limit for outer facesheets
J=8 : "MAX1" $ PARMAX1 : CP upper limit for skins
J=9 : "MAX2" $ PARMAX2 : K upper limit for skins
J=10: "MAX3" $ PARMAX3 : CP upper limit for honeycomb
J=11: "MAX4" $ PARMAX4 : K upper limit for honeycomb
J=12: "MAX5" $ PARMAX5 : CP upper limit for torque tubes
J=13: "MAX6" $ PARMAX6 : K upper limit for torque tubes
J=14: "MAX7" $ PARMAX7 : CP upper limit for outer facesheets
J=15: 0.05 $ PM : probability of mutation
$
*RETURN
*GEAL
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the $
$ properties of the materials used $

```

```

$                                     in the model.                                     $
$-----$-----
$
$(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for IM7/954-2A
$ upper and lower skins
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
                      J=1: 1.E+6 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
                      J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
                      J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
$ conduction properties for air
$
TABLE(NI=9,NJ=1):6 COND PROP 4:I=2 3 4
                      J=1: 0.9950 1009.0 0.7025
$
$TABLE(NI=9,NJ=14):6 COND PROP 4:I=1 2 3 4 5 6
$      J= 1: 250.0 1.3947 1006. 0.1758 0.1758 0.1758
$      J= 2: 300.0 1.1614 1007. 0.2916 0.2916 0.2916
$      J= 3: 350.0 0.9950 1009. 0.4513 0.4513 0.4513
$      J= 4: 400.0 0.8711 1014. 0.6627 0.6627 0.6627
$      J= 5: 450.0 0.7740 1021. 0.9327 0.9327 0.9327
$      J= 6: 500.0 0.6964 1030. 1.2690 1.2690 1.2690
$      J= 7: 550.0 0.6329 1040. 1.6788 1.6788 1.6788
$      J= 8: 600.0 0.5804 1051. 2.1694 2.1694 2.1694
$      J= 9: 650.0 0.5356 1063. 2.7483 2.7483 2.7483
$      J=10: 700.0 0.4975 1075. 3.4229 3.4229 3.4229
$      J=11: 750.0 0.4643 1087. 4.2005 4.2005 4.2005
$      J=12: 800.0 0.4354 1099. 5.0885 5.0885 5.0885
$      J=13: 850.0 0.4097 1110. 6.0943 6.0943 6.0943
$      J=14: 900.0 0.3868 1121. 7.2256 7.2256 7.2256
$
$ conduction properties for SS321 honeycomb (actually SS347)
$
TABLE(NI=9,NJ=1):6 COND PROP 5: I = 2 3 4
                      J=1: 1.E+6 "A3" "A4"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tubes, edge channels
$
TABLE(NI=9,NJ=1):6 COND PROP 6: I= 2 3 4
                      J=1: 1.E+6 "A5" "A6"
$
$ conduction properties for PETI-5 of the upper and lower outer
$ facesheets.
$
TABLE(NI=9,NJ=1):6 COND PROP 7: I= 2 3 4

```

J=1: 1.E+6 "A7" 7.025

```
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$ coordinates of the points used in $
$ the geometry of the plate. The $
$ points cited in the following $
$ subroutine correspond to these $
$ substructures: $
$ 1,2 :lower outer facesheet $
$ 2A,3 :lower honeycomb $
$ 3A,4 :lower inner facesheet $
$ 4A,5 :lower torque tube $
$ 5A,6 :air cavity $
$ 6A,7 :upper torque tube $
$ 7A,8 :upper inner facesheet $
$ 8A,9 :upper honeycomb $
$ 9A,10 :upper outer facesheet $
$ 10A,11 :insulation $
$-----$
$
*(29 PLAT DIME)DIME
$
!X1 = 0. $ x-coordinate for point 1
!X2 = 0. $ x-coordinate for point 2
!X3 = 0. $ x-coordinate for point 3
!X4 = 0. $ x-coordinate for point 4
!X5 = 0. $ x-coordinate for point 5
!X6 = 0. $ x-coordinate for point 6
!X7 = 0. $ x-coordinate for point 7
!X8 = 0. $ x-coordinate for point 8
!X9 = 0. $ x-coordinate for point 9
!X10 = 0. $ x-coordinate for point 10
!X11 = 0. $ x-coordinate for point 11
$
!Y1 = 0. $ y-coordinate for point 1
!Y2 = 0.0032 $ y-coordinate for point 2
!Y2A = 0.0032+(0.0254/"EY2") $ y-coordinate for point 2A
!Y3 = 0.0286 $ y-coordinate for point 3
!Y3A = 0.0286+(0.0061/"EY3") $ y-coordinate for point 3A
!Y4 = 0.0347 $ y-coordinate for point 4
!Y4A = 0.0347+(0.0021/"EY4") $ y-coordinate for point 4A
!Y5 = 0.0368 $ y-coordinate for point 5
!Y5A = 0.0368+(0.0510/"EY5") $ y-coordinate for point 5A
!Y6 = 0.0878 $ y-coordinate for point 6
!Y6A = 0.0878+(0.0021/"EY6") $ y-coordinate for point 6A
!Y7 = 0.0899 $ y-coordinate for point 7
!Y7A = 0.0899+(0.0050/"EY7") $ y-coordinate for point 7A
!Y8 = 0.0949 $ y-coordinate for point 8
!Y8A = 0.0949+(0.0254/"EY8") $ y-coordinate for point 8A
!Y9 = 0.1203 $ y-coordinate for point 9
!Y9A = 0.1203+(0.0032/"EY9") $ y-coordinate for point 9A
!Y10 = 0.1235 $ y-coordinate for point 10
!Y10A = 0.1235+(0.0254/"EY10") $ y-coordinate for point 10A
!Y11 = 0.1489 $ y-coordinate for point 11
$
```

```

!Z1 = 0.          $ z-coordinate for point 1
!Z2 = 0.          $ z-coordinate for point 2
!Z3 = 0.          $ z-coordinate for point 3
!Z4 = 0.          $ z-coordinate for point 4
!Z5 = 0.          $ z-coordinate for point 5
!Z6 = 0.          $ z-coordinate for point 6
!Z7 = 0.          $ z-coordinate for point 7
!Z8 = 0.          $ z-coordinate for point 8
!Z9 = 0.          $ z-coordinate for point 9
!Z10 = 0.         $ z-coordinate for point 10
!Z11 = 0.         $ z-coordinate for point 11
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the $
$                               coordinates of the nodes used in $
$                               the mesh. $
$-----$
$
*(29 NODE GENE)GENE
$
!INCY = 1          $ nodal increment in the y-direction
!T2 = 1 + "NY1"    $ starting node number for lower honeycomb
!T3 = "T2"+"EY2"   $ starting node number for lower inner facesheet
!T4 = "T3"+"EY3"   $ starting node number for lower torque tube edge
!T5 = "T4"+"EY4"   $ starting node number for cavity
!T6 = "T5"+"EY5"   $ starting node number for upper torque tube edge
!T7 = "T6"+"EY6"   $ starting node number for upper inner facesheet
!T8 = "T7"+"EY7"   $ starting node number for upper honeycomb
!T9 = "T8"+"EY8"   $ starting node number for upper outer facesheet
!T10 = "T9"+"EY9"  $ starting node number for insulation
$
*XQT TAB          $ table generator, structural properties
$
START "TOTN"      $ memory allocation for all the nodes
JLOC              $ initiation of the creation of the dataset containing
$                $ the joint position coordinates
FORMAT = 1        $ specifies rectangular coordinates
$
1,"X1","Y1","Z1","X2","Y2","Z2","NY1","INCY"
*IF("EY2" GT 1):*GOTO 100
"T2","X3","Y3","Z3"
*GOTO 105
$
*LABEL 100
"T2","X2","Y2A","Z2","X3","Y3","Z3","EY2","INCY"
*LABEL 105
*IF("EY3" GT 1):*GOTO 110
"T3","X4","Y4","Z4"
*GOTO 115
$
*LABEL 110
"T3","X3","Y3A","Z3","X4","Y4","Z4","EY3","INCY"
*LABEL 115
*IF("EY4" GT 1):*GOTO 120
"T4","X5","Y5","Z5"
*GOTO 125

```

```

$
*LABEL 120
"T4","X4","Y4A","Z4","X5","Y5","Z5","EY4","INCY"
*LABEL 125
*IF("EY5" GT 1):*GOTO 130
"T5","X6","Y6","Z6"
*GOTO 135
$
*LABEL 130
"T5","X5","Y5A","Z5","X6","Y6","Z6","EY5","INCY"
*LABEL 135
*IF("EY6" GT 1):*GOTO 140
"T6","X7","Y7","Z7"
*GOTO 145
$
*LABEL 140
"T6","X6","Y6A","Z6","X7","Y7","Z7","EY6","INCY"
*LABEL 145
*IF("EY7" GT 1):*GOTO 150
"T7","X8","Y8","Z8"
*GOTO 155
$
*LABEL 150
"T7","X7","Y7A","Z7","X8","Y8","Z8","EY7","INCY"
*LABEL 155
*IF("EY8" GT 1):*GOTO 160
"T8","X9","Y9","Z9"
*GOTO 165
$
*LABEL 160
"T8","X8","Y8A","Z8","X9","Y9","Z9","EY8","INCY"
*LABEL 165
*IF("EY9" GT 1):*GOTO 170
"T9","X10","Y10","Z10"
*GOTO 175
$
*LABEL 170
"T9","X9","Y9A","Z9","X10","Y10","Z10","EY9","INCY"
*LABEL 175
*IF("EY10" GT 1):*GOTO 180
"T10","X11","Y11","Z11"
*GOTO 185
$
*LABEL 180
"T10","X10","Y10A","Z10","X11","Y11","Z11","EY10","INCY"
*LABEL 185
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the $
$ elements used in the mesh. $
$-----$
$
*(29 MAKE ELEM)ELEM
$
*XQT AUS $ arithmetic utility System processor
$

```



```

TABLE(NI=1,NJ=1): K THICK : I = 1 $ section property table for
                        J=1: 1.0 $ K41 conduction elements
$
TABLE(NI=1,NJ=1): K AREA : I = 1 $ section property table for
                        J=1: 1.0 $ K21 conduction elements
$
*XQT ELD $ element definition processor
RESET NUTED = 1 $ destination library
$
!NOPT = 1 $ increment for the element number
!CLOS = 0
$
!J1 = 1 $ lower outer facesheet group#1
!J2 = "J1"+"INCY" $ lower outer facesheet group#1
!J3 = "J1"+"INCY"+"EY1" $ lower honeycomb group#2
!J4 = "J3"+"INCY" $ lower honeycomb group#2
!J5 = "J3"+"INCY"+"EY2" $ lower inner facesheet group#3
!J6 = "J5"+"INCY" $ lower inner facesheet group#3
!J7 = "J5"+"INCY"+"EY3" $ lower torque tube group#4
!J8 = "J7"+"INCY" $ lower torque tube group#4
!J9 = "J7"+"INCY"+"EY4" $ air cavity group#5
!J10= "J9"+"INCY" $ air cavity group#5
!J11= "J9"+"INCY"+"EY5" $ upper torque tube group#6
!J12= "J11"+"INCY" $ upper torque tube group#6
!J13= "J11"+"INCY"+"EY6" $ upper inner facesheet group#7
!J14= "J13"+"INCY" $ upper inner facesheet group#7
!J15= "J13"+"INCY"+"EY7" $ upper honeycomb group#8
!J16= "J15"+"INCY" $ upper honeycomb group#8
!J17= "J15"+"INCY"+"EY8" $ upper outer facesheet group#9
!J18= "J17"+"INCY" $ upper outer facesheet group#9
!J19= "J17"+"INCY"+"EY9" $ heater group#10
!J20= "J19" $ heater group#10
!J21= "J19" $ insulation group#11
!J22= "J19"+"INCY" $ insulation group#11
$
K21 $ 1-D conduction elements
$
GROUP = 1 $ group 1
NMAT = 7 $ material 1 conduction properties 1
"J1","J2","NOPT","EY1",1
$
GROUP = 2 $ group 2
NMAT = 5 $ material 2 conduction properties 2
"J3","J4","NOPT","EY2",1
$
GROUP = 3 $ group 3
NMAT = 1 $ material 3 conduction properties 3
"J5","J6","NOPT","EY3",1
$
GROUP = 4 $ group 4
NMAT = 6 $ material 4 conduction properties 4
"J7","J8","NOPT","EY4",1
$
GROUP = 5 $ group 5
NMAT = 4 $ material 5 conduction properties 5
"J9","J10","NOPT","EY5",1
$
GROUP = 6 $ group 6
NMAT = 6 $ material 6 conduction properties 6

```

```

"J11","J12","NOPT","EY6",1
$
GROUP = 7                                $ group 7
NMAT = 1                                $ material 7 conduction properties 7
"J13","J14","NOPT","EY7",1
$
GROUP = 8                                $ group 8
NMAT = 5                                $ material 8 conduction properties 8
"J15","J16","NOPT","EY8",1
$
GROUP = 9                                $ group 9
NMAT = 7                                $ material 9 conduction properties 9
"J17","J18","NOPT","EY9",1
$
GROUP = 10                               $ group 10
NMAT = 2                                $ material 10 conduction properties 10
"J19","J20","NOPT",1,1
$
GROUP = 11                               $ group 11
NMAT = 3                                $ material 11 conduction properties 11
"J21","J22","NOPT","EY10",1
$
*XQT TGeo                                $ element geometry processor
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the $
$                                boundary conditions such as: $
$                                - the constant temperature $
$                                boundary $
$                                - the applied heat flux boundary $
$                                - the total experimental and $
$                                heating time $
$                                - the heated node boundary $
$                                temperature $
$-----$
$
*(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the
$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=5,OPER=XSUM)
$

```

```

TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TOPT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BOTT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=5,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTT 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following 4 tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"

```

```

TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
$
DEFINE H = 5 DELT TIME
DEFINE I = 5 DELT CTMP
DEFINE J = 5 DELT HTMP
DEFINE K = 5 DELT FLUX
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 10
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=K,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=H,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=G,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=G,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
$
TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
TRANSFER(SOUR=TB,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=H,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied
$
TABLE(NI=1,NJ=2):5 TEMP NODE
I=1

```

```

J=1: 1.0
J=2: "NONJ"

$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets
$
TOCC (5 SOUR K21 10):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 GENA PROC)GENA: This subroutine is the first $
$ genetic processor that needs to be $
$ executed. Its purpose is to read $
$ the genetic variables defined in $
$ subroutine GEAL-VARI and to create $
$ the first parent population $
$ containing the first 50 $
$ chromosomes. More information $
$ about the necessary input and $
$ output tables as well as the RESET $
$ is given in the README file in the $
$ genetic directory. $
$-----$
$
$
*(29 GENA PROC)GENA
$
*XQT GENA
RESET NTIM=1 $print reset for the input data in output file
RESET NUIN=1 $reset for source library
RESET NOUT=1 $reset for destination library
RESET ITPP="ITPP" $reset for looping process
$
*RETURN
*GENA
$
$-----$
$ Subroutine(29 GENB PROC)GENB: This is the subroutine is the second$
$ genetic processor that needs to be $
$ executed. It creates the elitist $
$ parent population once the GENA $
$ processor has been executed several $
$ times. $
$-----$
$
$
*(29 GENB PROC)GENB
$
*XQT GENB
RESET ITPP="ITPP" $reset for looping process
RESET NUIN=1 $reset for source library
RESET NOUT=1 $reset for destination library
$
*RETURN
*GENB
$
$-----$

```

```

$ Subroutine(29 GENC PROC)GENC: This is the third processor that $
$                               needs to be executed. The objective $
$                               of this processor is to create a $
$                               children population using mutation, $
$                               cross-over and fitness criteria $
$                               applied to the elitist parent $
$                               population. $
$-----$
$
*(29 GENC PROC)GENC
$
*XQT GENC
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITERG" $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GENC
$
$-----$
$ Subroutine(29 GEND PROC)GEND: This is the fourth genetic processor$
$                               that needs to be executed. The goal $
$                               of this processor is to check for $
$                               twins and remove them, while $
$                               creating new candidate chromosomes. $
$-----$
$
*(29 GEND PROC)GEND
$
*XQT GEND
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITERG" $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GEND
$
$-----$
$ Subroutine(29 GENE PROC)GENF: This subroutine is genetic processor$
$                               four. Its intent is to create and $
$                               sort either the iteration population$
$                               or the final population depending on$
$                               whether the final number of $
$                               iterations has been achieved or not.$
$-----$
$
*(29 GENE PROC)GENF
$
*XQT GENE
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITERG" $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GENF
$

```

```

$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by $
$                               changing the Crank Nicholson para- $
$                               meter. $
$-----$
$
$(29 TRAN ANAL)ANAL
$
*XQT TRTB                $ implicit transient analyzer
  RESET QLIB = 5          $ thermal excitation library
  RESET MLIB = 6          $ material and fluids properties library
  RESET T1  = "TIMI"      $ starting time of analysis
  RESET T2  = "TIMF"      $ termination time of analysis
  RESET DT  = "INTS"      $ integration time step size
  RESET DEST = 4          $ output library
  RESET PRINT = 0         $ online print control parameter
  RESET KPRT = 0          $ print control parameter for k matrix
  RESET MXNDT = 10000000  $ maximum # of int. time steps during TRTB
  RESET BETA = 0.5        $ Crank Nicholson parameter
  RESET KTI  = 1          $ K matrix computation time parameter
  KTIME      = 10000.     $ times to calculate the conduction load vector
  TEMP       = "CSTP"     $ initial temperature of structure at T1
  TSAVE      = 5 EXPE TIME 1 1 $ storage of nodal temperatures at time
$                               $ increments of TSAVE
$
*RETURN
*ANAL
$
$-----$
$ Subroutine (29 SUMO SQUA)SQUA: This subroutine performs a $
$                               calculation of the sum of the $
$                               sum of squares for the different $
$                               chromosomes and puts them into the $
$                               objective function table. $
$-----$
$
$(29 SUMO SQUA)SQUA
$
$ the following if statements are used to define how far the runstream
$ has gotten with respect to the overall genetic process
$
*IF("SSQU" EQ 1):*GOTO 8000      $ chromosomes created by GENA
*IF("SSQU" EQ 2):*GOTO 8100      $ chromosomes created by GEND
*IF("SSQU" EQ 3):*GOTO 8200      $ chromosomes created by GENE
$
*LABEL 8000
*XQT AUS                $ arithmetic utility system processor
$
!LINE = 0                $ initializing counter
$
*XQT DCU                $ database utility function processor
!NWD = TOC,NJ(1 PARE POPU 1)    $ determine number of rows in PARE POPU
*XQT AUS                $ arithmetic utility system processor
$
$ the following two tables need to be initialized before the beginning of
$ the loop, in order to use the update command TABLE,U, the tables are
$ both filled with zeros. the first table will contain the value of the

```

```

$ objective function, whereas the second represents the identity matrix
$
TABLE(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 105
$
!LINE = LINE + 1                $ increment of the counter
$
!A1 = DS,1,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A2 = DS,2,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A3 = DS,3,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A4 = DS,4,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A5 = DS,5,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A6 = DS,6,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A7 = DS,7,"LINE",1 (1 PARE POPU 1) $ update the value of the property
$
*DCALL (29 MATE PROP)            $ call subroutine MATE PROP
*DCALL (29 TRAN ANAL)            $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)            $ call subroutine TEMP TABL
ERASE 6
$
INLIB = 3                        $ change source library
OUTLI = 3                        $ change destination library
$
DEFINE TO = 2 HEAT NODE 1 1      $ define table as variable for arith. oper.
DEFINE TP = 2 TOPT NODE 1 1      $ define table as variable for arith. oper.
DEFINE TQ = 2 BOTT NODE 1 1      $ define table as variable for arith. oper.
DEFINE TL = 5 EXPE HTMP 1 1      $ define table as variable for arith. oper.
DEFINE TM = 5 EXPE TOPT 1 1      $ define table as variable for arith. oper.
DEFINE TN = 5 EXPE BOTT 1 1      $ define table as variable for arith. oper.
DEFINE ON = 7 ONES MATX 1 1      $ define table as variable for arith. oper.
$
SU1  = SUM(TL,-1. TO)            $ (Y1-ETA1)
SU2  = SUM(TM,-1. TP)            $ (Y2-ETA2)
SU3  = SUM(TN,-1. TQ)            $ (Y3-ETA3)
SS1X = XTY(SU1,SU1)              $ (Y1-ETA1)T(Y1-ETA1)
SS2X = XTY(SU2,SU2)              $ (Y2-ETA2)T(Y2-ETA2)
SS3X = XTY(SU3,SU3)              $ (Y3-ETA3)T(Y3-ETA3)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1) $ scalar of (Y1-ETA1)T(Y1-ETA1)
!SS2A = DS 1,1,1 (3 SS2X AUS 1 1) $ scalar of (Y2-ETA2)T(Y2-ETA2)
!SS3A = DS 1,1,1 (3 SS3X AUS 1 1) $ scalar of (Y3-ETA3)T(Y3-ETA3)
!SSTA = "SS1A"+"SS2A"+"SS3A"     $ total scalar value of the objective function
$
$ the TABLE,U command below updates the table created before the loop and
$ inserts the objective function that corresponds to the chromosomes
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
  J="LINE": "SSTA"
$
*IF("LINE" LT "NWD"): *GOTO 105
$
*GOTO 8300
$
$-----$
$
*LABEL 8100
*XQT AUS

```



```

$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 CHRO CHIL 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
J=1,"NDS": 1.0
$
*LABEL 106
$
!LINE = LINE + 1
$
!A1 = DS,1,"LINE",1 (1 CHRO CHIL 1)
!A2 = DS,2,"LINE",1 (1 CHRO CHIL 1)
!A3 = DS,3,"LINE",1 (1 CHRO CHIL 1)
!A4 = DS,4,"LINE",1 (1 CHRO CHIL 1)
!A5 = DS,5,"LINE",1 (1 CHRO CHIL 1)
!A6 = DS,6,"LINE",1 (1 CHRO CHIL 1)
!A7 = DS,7,"LINE",1 (1 CHRO CHIL 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1
DEFINE TP = 2 TOPT NODE 1 1
DEFINE TQ = 2 BOTT NODE 1 1
DEFINE TL = 5 EXPE HTMP 1 1
DEFINE TM = 5 EXPE TOPT 1 1
DEFINE TN = 5 EXPE BOTT 1 1
DEFINE ON = 7 ONES MATX 1 1
$
SU1 = SUM(TL,-1. TO)
SU2 = SUM(TM,-1. TP)
SU3 = SUM(TN,-1. TQ)
SS1X = XTY(SU1,SU1)
SS2X = XTY(SU2,SU2)
SS3X = XTY(SU3,SU3)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1)
!SS2A = DS 1,1,1 (3 SS2X AUS 1 1)
!SS3A = DS 1,1,1 (3 SS3X AUS 1 1)
!SSTA = "SS1A"+"SS2A"+"SS3A"
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
J="LINE":"SSTA"
$
*IF("LINE" LT "NWD"):GOTO 106
$
*GOTO 8300
$
$-----$
$

```

```

*LABEL 8200
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 FRES BLOD 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 107
$
!LINE = LINE + 1
$
!A1 = DS,1,"LINE",1 (1 FRES BLOD 1)
!A2 = DS,2,"LINE",1 (1 FRES BLOD 1)
!A3 = DS,3,"LINE",1 (1 FRES BLOD 1)
!A4 = DS,4,"LINE",1 (1 FRES BLOD 1)
!A5 = DS,5,"LINE",1 (1 FRES BLOD 1)
!A6 = DS,6,"LINE",1 (1 FRES BLOD 1)
!A7 = DS,7,"LINE",1 (1 FRES BLOD 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1
DEFINE TP = 2 TOPT NODE 1 1
DEFINE TQ = 2 BOTT NODE 1 1
DEFINE TL = 5 EXPE HTMP 1 1
DEFINE TM = 5 EXPE TOPT 1 1
DEFINE TN = 5 EXPE BOTT 1 1
DEFINE ON = 7 ONES MATX 1 1
$
SU1  = SUM(TL,-1. TO)
SU2  = SUM(TM,-1. TP)
SU3  = SUM(TN,-1. TQ)
SS1X = XTY(SU1,SU1)
SS2X = XTY(SU2,SU2)
SS3X = XTY(SU3,SU3)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1)
!SS2A = DS 1,1,1 (3 SS2X AUS 1 1)
!SS3A = DS 1,1,1 (3 SS3X AUS 1 1)
!SSTA = "SS1A"+"SS2A"+"SS3A"
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
  J="LINE": "SSTA"
$
*IF("LINE" LT "NWD"): *GOTO 107
$
*LABEL 8300
$

```

```

*RETURN
*SQUA
$
$-----$
$ Subroutine (29 ITER ATIO)ATIO: This subroutine updates the thermal$
$                               properties of the parameters and $
$                               executes the genetic processors $
$                               n-times for n-generations. $
$-----$
$
*(29 ITER ATIO)ATIO
$
!ITERG = 0                                $initializing the counter for the loop
$
*LABEL 400
$
!ITERG = ITERG + 1                        $counter for the loop
!SSQU = 2                                $flag for the SUMO SQUA subroutine
$
*DCALL (29 GENC PROC)                    $call subroutine GENC PROC
*DCALL (29 SUMO SQUA)                    $call subroutine SUMO SQUA
  ERASE 2                                $erase content of library 2
  ERASE 3                                $erase content of library 3
  ERASE 4                                $erase content of library 4
  ERASE 6                                $erase content of library 6
!SSQU = 3                                $flag for the SUMO SQUA subroutine
*DCALL (29 GEND PROC)                    $call subroutine GEND PROC
*DCALL (29 SUMO SQUA)                    $call subroutine SUMO SQUA
$
!MIN1 = DS,1,1,1 (1 NEWW PARA)           $reset the lower boundary for Cp
!MIN2 = DS,1,2,1 (1 NEWW PARA)           $reset the lower boundary for K
!MIN3 = DS,1,3,1 (1 NEWW PARA)           $reset the lower boundary
!MIN4 = DS,1,4,1 (1 NEWW PARA)           $reset the lower boundary
!MIN5 = DS,1,5,1 (1 NEWW PARA)           $reset the lower boundary
!MIN6 = DS,1,6,1 (1 NEWW PARA)           $reset the lower boundary
!MIN7 = DS,1,7,1 (1 NEWW PARA)           $reset the lower boundary
!MAX1 = DS,1,8,1 (1 NEWW PARA)           $reset the upper boundary for Cp
!MAX2 = DS,1,9,1 (1 NEWW PARA)           $reset the upper boundary for K
!MAX3 = DS,1,10,1 (1 NEWW PARA)          $reset the upper boundary
!MAX4 = DS,1,11,1 (1 NEWW PARA)          $reset the upper boundary
!MAX5 = DS,1,12,1 (1 NEWW PARA)          $reset the upper boundary
!MAX6 = DS,1,13,1 (1 NEWW PARA)          $reset the upper boundary
!MAX7 = DS,1,14,1 (1 NEWW PARA)          $reset the upper boundary
$
*DCALL (29 GEAL VARI)                    $call the subroutine GEAL VARI
  ERASE 2                                $erase content of library 2
  ERASE 3                                $erase content of library 3
  ERASE 4                                $erase content of library 4
  ERASE 6                                $erase content of library 6
*DCALL (29 GENE PROC)                    $call subroutine GENE PROC
$
*IF("ITERG" EQ "ITMX"): *GOTO 410
*GOTO 400
*LABEL 410
$
*RETURN
*ATIO
$
$-----$

```

```

$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.             $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrta time step for TRTB (seconds)
$
!MIN1 = 0.05     $ lower bound for skins (PCp)
!MIN2 = 0.05     $ lower bound for skins (K)
!MIN3 = 0.05     $ lower bound for honeycomb (PCp)
!MIN4 = 0.05     $ lower bound for honeycomb (K)
!MIN5 = 0.05     $ lower bound for torque tubes (PCp)
!MIN6 = 0.05     $ lower bound for torque tubes (K)
!MIN7 = 0.05     $ lower bound for facesheets (PCp)
!MAX1 = 10.0     $ upper bound for skins (PCp)
!MAX2 = 10.0     $ upper bound for skins (K)
!MAX3 = 10.0     $ upper bound for honeycomb (PCp)
!MAX4 = 10.0     $ upper bound for honeycomb (K)
!MAX5 = 10.0     $ upper bound for torque tubes (PCp)
!MAX6 = 10.0     $ upper bound for torque tubes (K)
!MAX7 = 10.0     $ upper bound for facesheets (PCp)
$
!A1  = 2.54      $ Initial value for skins (PCp)
!A2  = 6.35      $ Initial value for skins (K)
!A3  = 1.72      $ Initial value for honeycomb (PCp)
!A4  = 0.15      $ Initial value for honeycomb (K)
!A5  = 1.26      $ Initial value for torque tubes (PCp)
!A6  = 5.66      $ Initial value for torque tubes (K)
!A7  = 9.58      $ Initial value for facesheets (PCp)
$
!A1I = 2.54      $ Initial value for skins (PCp)
!A2I = 6.35      $ Initial value for skins (K)
!A3I = 1.72      $ Initial value for honeycomb (PCp)
!A4I = 0.15      $ Initial value for honeycomb (K)
!A5I = 1.26      $ Initial value for torque tubes (PCp)
!A6I = 5.66      $ Initial value for torque tubes (K)
!A7I = 9.57      $ Initial value for facesheets (PCp)
$
*RETURN
*SPEC
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                   $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU          $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1      $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1     $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)

```

```

$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                                $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME)           $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1                         $ heated node number - 1
!VAR3 = "TOTN"-1                         $ total node number - 1
!VAR4 = "NON2"-1                         $ node at top of the torque tube
!VAR5 = "NON3"-1                         $ node at the bottom of the torque tube
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 1):*GOTO 5000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2):*GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3):*GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4):*GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5):*GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6):*GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7):*GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8):*GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9):*GOTO 5080 $ Temperature histories for property 7
$
*LABEL 5000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
TABLE(NI=1,NJ="VAR1"):2 TOPT NODE 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR4",SSKIP="VAR3")
$
TABLE(NI=1,NJ="VAR1"):2 BOTT NODE 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR5",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
DEFINE D = 2 TOPT NODE 1 1
DEFINE E = 2 BOTT NODE 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTT 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
      TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
      TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
      TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
TABLE(NI=3,NJ="NDS"):5 TEMP TOPT 1 1
      TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
      TRAN(SOUR=F,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
      TRAN(SOUR=D,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
TABLE(NI=3,NJ="NDS"):5 TEMP BOTT 1 1
      TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
      TRAN(SOUR=G,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
      TRAN(SOUR=E,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
*GOTO 6000
$
*LABEL 5010

```

```

TABLE(NI=1,NJ="VAR1"):2 INIT 00 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
      TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS
$
*TF OPEN 1'data
*TF READ 1
*TF CLOSE 1
$
!N2 = 1
$
*DCALL (29 MAKE VARI) $ Call subroutine MAKE VARI
*DCALL (29 PROB SPEC) $ Call subroutine PROB SPEC

```

```

*DCALL (29 GEAL VARI) $ Call subroutine GEAL VARI
*DCALL (29 PLAT DIME) $ Call subroutine PLAT DIME
*DCALL (29 NODE GENE) $ Call subroutine NODE GENE
*DCALL (29 MAKE ELEM) $ Call subroutine MAKE ELEM
*DCALL (29 BOUN COND) $ Call subroutine BOUN COND
!ITPP = 1          $ Reset parameter for initial population 1
!SSQU = 1
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 2
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 3
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 4
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 5
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
$
!ITMX = 10          $ Maximum number of iteration
*DCALL (29 ITER ATIO) $ Call subroutine ITER ATIO
$
*XQT DCU
  PRINT 1 ITER PAPO
  PRINT 1 ITER OBFN
*XQT EXIT

```

Appendix L

EAL One-Dimensional Detailed Half-Model: Estimation Program

```
*XQT U1
*ONLINE=0
!MAXCM = SSP(0,13)
*CMPARA(MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of $
$                               the variables used in the program $
$                               such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each $
$                               section $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $arithmetic utility system processor
$
!NY6 = 8           $ number of nodes for the upper torque tube
!NY7 = 12          $ number of nodes for the upper inner facesheet
!NY8 = 36          $ number of nodes for the upper honeycomb structure
!NY9 = 8           $ number of nodes for the upper outer facesheet
!NY10 = 36         $ number of nodes for the insulation
$
!EY6 ="NY6"-1      $ number of elements for the upper torque tube
!EY7 ="NY7"-1      $ number of elements for the upper inner facesheet
!EY8 ="NY8"-1      $ number of elements for the upper honeycomb structure
!EY9 ="NY9"-1      $ number of elements for the upper outer facesheet
!EY10 ="NY10"-1    $ number of elements for the insulation
$
!TOTN = "NY6"+"EY7"+"EY8"+"EY9"+"EY10"
$                 $ total number of nodes in the model
!NON1 = "NY6"+"EY7"+"EY8"+"EY9"
$                 $ node number at the heater (high sensitivity)
!NONK = FLOAT("TOTN")
```



```

$                               $ node number on top of the insulation
$
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 GEAL VARI)GEAL: This subroutine defines some of  $
$                               the variables used in the genetic  $
$                               algorithm processors.                $
$-----$
$
*(29 GEAL VARI) GEAL
*XQT AUS
$
TABLE(NI=1,NJ=9,TYPE=0) : INPT INTE
I=1
  J=1:      7      $ NC      : number of estimated parameters
  J=2:      5      $ NPOP     : number of initial populations to investigate
  J=3: 15287      $ CFSEED1  : parameter used by random number generator
  J=4:  9428      $ CFSEED2  : parameter used by random number generator
  J=5:     15      $ SLOPSEED : parameter used by random number generator
  J=6:     10      $ NBEST1   : # chro. kept from NPOP initial population
  J=7:     40      $ NBEST2   : # chro. kept from parent children population
  J=8:     30      $ NBEST3   : # chro. kept from parent children population
  J=9:     10      $ ITMX     : maximum number of generations to perform
$
TABLE(NI=1,NJ=21,TYPE=-2) : INPT REAL
I=1
  J=1 : "MIN1"      $ PARMIN1  : CP lower limit for skins
  J=2 : "MIN2"      $ PARMIN2  : K lower limit for skins
  J=3 : "MIN3"      $ PARMIN3  : CP lower limit for honeycomb
  J=4 : "MIN4"      $ PARMIN4  : K lower limit for honeycomb
  J=5 : "MIN5"      $ PARMIN5  : CP lower limit for torque tubes
  J=6 : "MIN6"      $ PARMIN6  : K lower limit for torque tubes
  J=7 : "MIN7"      $ PARMIN7  : CP lower limit for outer facesheets
  J=8 : "MIN8"      $ PARMIN8  : K lower limit for outer facesheets
  J=9 : "MIN9"      $ PARMIN9  : CP lower limit for air gap
  J=10: "MINX"      $ PARMINX  : K lower limit for air gap
  J=11: "MAX1"      $ PARMAX1  : CP upper limit for skins
  J=12: "MAX2"      $ PARMAX2  : K upper limit for skins
  J=13: "MAX3"      $ PARMAX3  : CP upper limit for honeycomb
  J=14: "MAX4"      $ PARMAX4  : K upper limit for honeycomb
  J=15: "MAX5"      $ PARMAX5  : CP upper limit for torque tubes
  J=16: "MAX6"      $ PARMAX6  : K upper limit for torque tubes
  J=17: "MAX7"      $ PARMAX7  : CP upper limit for outer facesheets
  J=18: "MAX8"      $ PARMAX8  : K upper limit for outer facesheets
  J=19: "MAX9"      $ PARMAX9  : CP upper limit for air gap
  J=20: "MAXX"      $ PARMAX10 : K upper limit for air gap
  J=21: 0.05       $ PM       : probability of mutation
$
*RETURN
*GEAL
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the  $
$                               properties of the materials used  $
$                               in the model.                    $
$-----$
$

```

```

*(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for IM7/954-2A
$ upper and lower skins
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4
      J=1: 1.E+6 "A1" "A2"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
      J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
      J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
$ conduction properties for air
$
TABLE(NI=9,NJ=1):6 COND PROP 4:I=2 3 4
      J=1: 0.9950 1009.0 0.7025
$
$TABLE(NI=9,NJ=14):6 COND PROP 4:I=1 2 3 4 5 6
$      J= 1: 250.0 1.3947 1006. 0.1758 0.1758 0.1758
$      J= 2: 300.0 1.1614 1007. 0.2916 0.2916 0.2916
$      J= 3: 350.0 0.9950 1009. 0.4513 0.4513 0.4513
$      J= 4: 400.0 0.8711 1014. 0.6627 0.6627 0.6627
$      J= 5: 450.0 0.7740 1021. 0.9327 0.9327 0.9327
$      J= 6: 500.0 0.6964 1030. 1.2690 1.2690 1.2690
$      J= 7: 550.0 0.6329 1040. 1.6788 1.6788 1.6788
$      J= 8: 600.0 0.5804 1051. 2.1694 2.1694 2.1694
$      J= 9: 650.0 0.5356 1063. 2.7483 2.7483 2.7483
$      J=10: 700.0 0.4975 1075. 3.4229 3.4229 3.4229
$      J=11: 750.0 0.4643 1087. 4.2005 4.2005 4.2005
$      J=12: 800.0 0.4354 1099. 5.0885 5.0885 5.0885
$      J=13: 850.0 0.4097 1110. 6.0943 6.0943 6.0943
$      J=14: 900.0 0.3868 1121. 7.2256 7.2256 7.2256
$
$ conduction properties for SS321 honeycomb (actually SS347)
$
TABLE(NI=9,NJ=1):6 COND PROP 5: I = 2 3 4
      J=1: 1.E+6 "A3" "A4"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tubes, edge channels
$
TABLE(NI=9,NJ=1):6 COND PROP 6: I= 2 3 4
      J=1: 1.E+6 "A5" "A6"
$
$ conduction properties for PETI-5 of the upper and lower outer
$ facesheets.
$
TABLE(NI=9,NJ=1):6 COND PROP 7: I= 2 3 4
      J=1: 1.E+6 "A7" "A8"
$
*RETURN

```

```

*PROP
$
$-----$
$ Subroutine (29 PLAT DIME)DIME: This subroutine defines the $
$ coordinates of the points used in $
$ the geometry of the plate. The $
$ points cited in the following $
$ subroutine correspond to these $
$ substructures: $
$ 6,7 :upper torque tube $
$ 7A,8 :upper inner facesheet $
$ 8A,9 :upper honeycomb $
$ 9A,10 :upper outer facesheet $
$ 10A,11 :insulation $
$-----$
$
$
*(29 PLAT DIME)DIME
$
!X6 = 0. $ x-coordinate for point 6
!X7 = 0. $ x-coordinate for point 7
!X8 = 0. $ x-coordinate for point 8
!X9 = 0. $ x-coordinate for point 9
!X10 = 0. $ x-coordinate for point 10
!X11 = 0. $ x-coordinate for point 11
$
!Y6 = 0.0878 $ y-coordinate for point 6
!Y7 = 0.0899 $ y-coordinate for point 7
!Y7A = 0.0899+(0.0050/"EY7") $ y-coordinate for point 7A
!Y8 = 0.0949 $ y-coordinate for point 8
!Y8A = 0.0949+(0.0254/"EY8") $ y-coordinate for point 8A
!Y9 = 0.1203 $ y-coordinate for point 9
!Y9A = 0.1203+(0.0032/"EY9") $ y-coordinate for point 9A
!Y10 = 0.1235 $ y-coordinate for point 10
!Y10A = 0.1235+(0.0254/"EY10") $ y-coordinate for point 10A
!Y11 = 0.1489 $ y-coordinate for point 11
$
!Z6 = 0. $ z-coordinate for point 6
!Z7 = 0. $ z-coordinate for point 7
!Z8 = 0. $ z-coordinate for point 8
!Z9 = 0. $ z-coordinate for point 9
!Z10 = 0. $ z-coordinate for point 10
!Z11 = 0. $ z-coordinate for point 11
$
*RETURN
*DIME
$
$-----$
$ Subroutine (29 NODE GENE)GENE: This subroutine defines the $
$ coordinates of the nodes used in $
$ the mesh. $
$-----$
$
$(29 NODE GENE)GENE
$
!INCY = 1 $ nodal increment in the y-direction
!T6 = 1 $ starting node number for upper torque tube edge
!T7 = "T6"+"NY6" $ starting node number for upper inner facesheet
!T8 = "T7"+"EY7" $ starting node number for upper honeycomb
!T9 = "T8"+"EY8" $ starting node number for upper outer facesheet

```

```

!T10 = "T9"+"EY9"  $ starting node number for insulation
$
*XQT TAB           $ table generator, structural properties
$
START "TOTN"       $ memory allocation for all the nodes
JLOC               $ initiation of the creation of the dataset containing
$                 $ the joint position coordinates
FORMAT = 1         $ specifies rectangular coordinates
$
"T6","X6","Y6","Z6","X7","Y7","Z7","NY6","INCY"
*LABEL 145
*IF("EY7" GT 1):*GOTO 150
"T7","X8","Y8","Z8"
*GOTO 155
$
*LABEL 150
"T7","X7","Y7A","Z7","X8","Y8","Z8","EY7","INCY"
*LABEL 155
*IF("EY8" GT 1):*GOTO 160
"T8","X9","Y9","Z9"
*GOTO 165
$
*LABEL 160
"T8","X8","Y8A","Z8","X9","Y9","Z9","EY8","INCY"
*LABEL 165
*IF("EY9" GT 1):*GOTO 170
"T9","X10","Y10","Z10"
*GOTO 175
$
*LABEL 170
"T9","X9","Y9A","Z9","X10","Y10","Z10","EY9","INCY"
*LABEL 175
*IF("EY10" GT 1):*GOTO 180
"T10","X11","Y11","Z11"
*GOTO 185
$
*LABEL 180
"T10","X10","Y10A","Z10","X11","Y11","Z11","EY10","INCY"
*LABEL 185
$
*RETURN
*GENE
$
$-----$
$ Subroutine (29 MAKE ELEM)ELEM: This subroutine defines the      $
$                               elements used in the mesh.         $
$-----$
$
*(29 MAKE ELEM)ELEM
$
*XQT AUS           $ arithmetic utility System processor
$
TABLE(NI=1,NJ=1):  K THICK : I = 1  $ section property table for
                   J=1: 1.0         $ K41 conduction elements
$
TABLE(NI=1,NJ=1):  K AREA  : I = 1  $ section property table for
                   J=1: 1.0         $ K21 conduction elements
$
*XQT ELD           $ element definition processor

```

```

RESET NUTED = 1                                $ destination library
$
!NOPT = 1                                       $ increment for the element number
!CLOS = 0
$
!J11= 1                                       $ upper torque tube      group#1
!J12= "J11"+"INCY"                           $ upper torque tube      group#1
!J13= "J11"+"("INCY"*"EY6")"                 $ upper inner facesheet group#2
!J14= "J13"+"INCY"                           $ upper inner facesheet group#2
!J15= "J13"+"("INCY"*"EY7")"                 $ upper honeycomb        group#3
!J16= "J15"+"INCY"                           $ upper honeycomb        group#3
!J17= "J15"+"("INCY"*"EY8")"                 $ upper outer facesheet group#4
!J18= "J17"+"INCY"                           $ upper outer facesheet group#4
!J19= "J17"+"("INCY"*"EY9")"                 $ heater                  group#5
!J20= "J19"                                   $ heater                  group#5
!J21= "J19"                                   $ insulation              group#6
!J22= "J19"+"INCY"                           $ insulation              group#6
$
K21                                           $ 1-D conduction elements
$
GROUP = 1                                    $ group 1
NMAT = 6                                     $ material 6 conduction properties 6
"J11","J12","NOPT","EY6",1
$
GROUP = 2                                    $ group 2
NMAT = 1                                     $ material 7 conduction properties 7
"J13","J14","NOPT","EY7",1
$
GROUP = 3                                    $ group 3
NMAT = 5                                     $ material 8 conduction properties 8
"J15","J16","NOPT","EY8",1
$
GROUP = 4                                    $ group 4
NMAT = 7                                     $ material 9 conduction properties 9
"J17","J18","NOPT","EY9",1
$
GROUP = 5                                    $ group 5
NMAT = 2                                     $ material 10 conduction properties 10
"J19","J20","NOPT",1,1
$
GROUP = 6                                    $ group 6
NMAT = 3                                     $ material 11 conduction properties 11
"J21","J22","NOPT","EY10",1
$
*XQT TGEO                                    $ element geometry processor
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the $
$ boundary conditions such as: $
$ - the constant temperature $
$ boundary $
$ - the applied heat flux boundary $
$ - the total experimental and $
$ heating time $
$ - the heated node boundary $
$ temperature $

```

```

$-----$
$
$(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DATA
$
$ in the following 4 statements, the experimental time, temperature at the
$ constant temperature boundary, temperature at the heated surface, the
$ applied heat flux, the temperature at the top of the torque tube, and
$ the temperature at the bottom of the torque tube.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TOPT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=5,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BOTT 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=5,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE HTMP 1 1
DEFINE E = 5 EXPE FLUX 1 1
DEFINE F = 5 EXPE TOPT 1 1
DEFINE G = 5 EXPE BOTT 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$

```

```

$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!HTMP = DS 1,"CNTR",1 (5 EXPE HTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!TOPT = DS 1,"CNTR",1 (5 EXPE TOPT 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT HTMP:I=1:J=1:"HTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
TABLE(NI=1,NJ=1):5 DELT TOPT:I=1:J=1:"TOPT"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT HTMP
DEFINE EE = 5 DELT FLUX
DEFINE FF = 5 DELT TOPT
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 SOUR K21 5
TRANSFER(SOUR=E,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=EE,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=E,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM1
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)

```

```

$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM2
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 APPL TEM3
  TRANSFER(SOUR=F,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=FF,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=F,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
DEFINE TA = 5 APPL TEM1
DEFINE TB = 5 APPL TEM2
DEFINE TC = 5 APPL TEM3
$
TABLE(NI=2,NJ="NDS1"):5 APPL TEMP
  TRANSFER(SOUR=TC,ILIM=1,JLIM="NDS1",DBASE=0,DSKIP=1)
  TRANSFER(SOUR=TA,ILIM=1,JLIM="NDS1",DBASE=1,DSKIP=1)
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
TABLE(NI=1,NJ=2):5 TEMP NODE
  I=1
      J=1: 1.0
      J=2: "NONK"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 5):NJ=1,NINJ=1
TOCC (5 APPL TEMP) :NJ=2,NINJ=2
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 GENA PROC)GENA: This subroutine is the first $
$ genetic processor that needs to be $
$ executed. Its purpose is to read $
$ the genetic variables defined in $
$ subroutine GEAL-VARI and to create $
$ the first parent population $
$ containing the first 50 $
$ chromosomes. More information $
$ about the necessary input and $
$ output tables as well as the RESET $
$ is given in the README file in the $
$ genetic directory. $
$-----$

```



```

$
*(29 GENA PROC)GENA
$
*XQT GENA
  RESET NTIM=1          $print reset for the input data in output file
  RESET NUIN=1          $reset for source library
  RESET NOUT=1          $reset for destination library
  RESET ITPP="ITPP"     $reset for looping process
$
*RETURN
*GENA
$
$-----$
$ Subroutine(29 GENB PROC)GENB: This is the subroutine is the second$
$                               genetic processor that needs to be $
$                               executed. It creates the elitist $
$                               parent population once the GENA $
$                               processor has been executed several $
$                               times.                               $
$-----$
$
*(29 GENB PROC)GENB
$
*XQT GENB
  RESET ITPP="ITPP"     $reset for looping process
  RESET NUIN=1          $reset for source library
  RESET NOUT=1          $reset for destination library
$
*RETURN
*GENB
$
$-----$
$ Subroutine(29 GENC PROC)GENC: This is the third processor that $
$                               needs to be executed. The objective $
$                               of this processor is to create a $
$                               children population using mutation, $
$                               cross-over and fitness criteria $
$                               applied to the elitist parent $
$                               population.                         $
$-----$
$
*(29 GENC PROC)GENC
$
*XQT GENC
  RESET ITPP="ITPP"     $reset for looping process
  RESET ITRG="ITERG"    $reset for generation iteration process
  RESET NUIN= 1         $reset for source library
  RESET NOUT= 1         $reset for destination library
$
*RETURN
*GENC
$
$-----$
$ Subroutine(29 GEND PROC)GEND: This is the fourth genetic processor$
$                               that needs to be executed. The goal $
$                               of this processor is to check for $
$                               twins and remove them, while $
$                               creating new candidate chromosomes. $
$-----$

```

```

$
*(29 GEND PROC)GEND
$
*XQT GEND
  RESET ITPP="ITPP"   $reset for looping process
  RESET ITRG="ITERG"  $reset for generation iteration process
  RESET NUIN= 1       $reset for source library
  RESET NOUT= 1       $reset for destination library
$
*RETURN
*GEND
$
$-----$
$ Subroutine(29 GENE PROC)GENF: This subroutine is genetic processor$
$                               four. Its intent is to create and $
$                               sort either the iteration population$
$                               or the final population depending on$
$                               whether the final number of      $
$                               iterations has been achieved or not.$
$-----$
$
*(29 GENE PROC)GENF
$
*XQT GENE
  RESET ITPP="ITPP"   $reset for looping process
  RESET ITRG="ITERG"  $reset for generation iteration process
  RESET NUIN= 1       $reset for source library
  RESET NOUT= 1       $reset for destination library
$
*RETURN
*GENF
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by $
$                               changing the Crank Nicholson para- $
$                               meter.                             $
$-----$
$
*(29 TRAN ANAL)ANAL
$
*XQT TRTB
  RESET QLIB = 5          $ implicit transient analyzer
  RESET MLIB = 6          $ thermal excitation library
  RESET T1   = "TIMI"     $ material and fluids properties library
  RESET T2   = "TIMF"     $ starting time of analysis
  RESET DT   = "INTS"     $ termination time of analysis
  RESET DEST = 4          $ integration time step size
  RESET PRINT = 0         $ output library
  RESET KPRT = 0          $ online print control parameter
  RESET MXNDT = 10000000  $ print control parameter for k matrix
  RESET BETA = 0.5        $ maximum # of int. time steps during TRTB
  RESET KTI  = 1          $ Crank Nicholson parameter
  KTIME      = 10000.     $ K matrix computation time parameter
  TEMP       = "CSTP"     $ times to calculate the conduction load vector
  TSAVE      = 5 EXPE TIME 1 1 $ initial temperature of structure at T1
  $                               $ storage of nodal temperatures at time
  $                               $ increments of TSAVE

```

```

*RETURN
*ANAL
$
$-----$
$ Subroutine (29 SUMO SQUA)SQUA: This subroutine performs a      $
$                               calculation of the sum of the      $
$                               sum of squares for the different    $
$                               chromosomes and puts them into the  $
$                               objective function table.           $
$-----$
$
*(29 SUMO SQUA)SQUA
$
$ the following if statements are used to define how far the runstream
$ has gotten with respect to the overall genetic process
$
*IF("SSQU" EQ 1):*GOTO 8000      $ chromosomes created by GENA
*IF("SSQU" EQ 2):*GOTO 8100      $ chromosomes created by GEND
*IF("SSQU" EQ 3):*GOTO 8200      $ chromosomes created by GENE
$
*LABEL 8000
*XQT AUS                        $ arithmetic utility system processor
$
!LINE = 0                        $ initializing counter
$
*XQT DCU                        $ database utility function processor
!NWD = TOC,NJ(1 PARE POPU 1)    $ determine number of rows in PARE POPU
*XQT AUS                        $ arithmetic utility system processor
$
$ the following two tables need to be initialized before the beginning of
$ the loop, in order to use the update command TABLE,U, the tables are
$ both filled with zeros. the first table will contain the value of the
$ objective function, whereas the second represents the identity matrix
$
TABLE(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 105
$
!LINE = LINE + 1                $ increment of the counter
$
!A1 = DS,1 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A2 = DS,2 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A3 = DS,3 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A4 = DS,4 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A5 = DS,5 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A6 = DS,6 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A7 = DS,7 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A8 = DS,8 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A9 = DS,9 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A10 = DS,10,"LINE",1 (1 PARE POPU 1) $ update the value of the property
$
*DCALL (29 MATE PROP)           $ call subroutine MATE PROP
*DCALL (29 TRAN ANAL)           $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)           $ call subroutine TEMP TABL
ERASE 6
$
INLIB = 3                       $ change source library

```

```

OUTLI = 3                                $ change destination library
$
DEFINE TO = 2 HEAT NODE 1 1              $ define table as variable for arith. oper.
DEFINE TL = 5 EXPE HTMP 1 1              $ define table as variable for arith. oper.
DEFINE ON = 7 ONES MATX 1 1              $ define table as variable for arith. oper.
$
SU1   = SUM(TL,-1. TO)                   $ (Y1-ETA1)
SS1X  = XTY(SU1,SU1)                     $ (Y1-ETA1)T(Y1-ETA1)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1)        $ scalar of (Y1-ETA1)T(Y1-ETA1)
!SSTA = "SS1A"                           $ total scalar value of the objective function
$
$ the TABLE,U command below updates the table created before the loop and
$ inserts the objective function that corresponds to the chromosomes
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
  J="LINE": "SSTA"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 105
$
*GOTO 8300
$
$-----$
$
*LABEL 8100
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 CHRO CHIL 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 106
$
!LINE = LINE + 1
$
!A1 = DS,1 ,"LINE",1 (1 CHRO CHIL 1)
!A2 = DS,2 ,"LINE",1 (1 CHRO CHIL 1)
!A3 = DS,3 ,"LINE",1 (1 CHRO CHIL 1)
!A4 = DS,4 ,"LINE",1 (1 CHRO CHIL 1)
!A5 = DS,5 ,"LINE",1 (1 CHRO CHIL 1)
!A6 = DS,6 ,"LINE",1 (1 CHRO CHIL 1)
!A7 = DS,7 ,"LINE",1 (1 CHRO CHIL 1)
!A8 = DS,8 ,"LINE",1 (1 CHRO CHIL 1)
!A9 = DS,9 ,"LINE",1 (1 CHRO CHIL 1)
!A10 = DS,10,"LINE",1 (1 CHRO CHIL 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3

```

```

OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1      $ define table as variable for arith. oper.
DEFINE TL = 5 EXPE HTMP 1 1      $ define table as variable for arith. oper.
DEFINE ON = 7 ONES MATX 1 1      $ define table as variable for arith. oper.
$
SU1  = SUM(TL,-1. TO)            $ (Y1-ETA1)
SS1X = XTY(SU1,SU1)              $ (Y1-ETA1)T(Y1-ETA1)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1) $ scalar of (Y1-ETA1)T(Y1-ETA1)
!SSTA = "SS1A"                   $ total scalar value of the objective function
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
  J="LINE": "SSTA"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 106
$
*GOTO 8300
$
$-----$
$
*LABEL 8200
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 FRES BLOD 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 107
$
!LINE = LINE + 1
$
!A1 = DS,1 ,"LINE",1 (1 FRES BLOD 1)
!A2 = DS,2 ,"LINE",1 (1 FRES BLOD 1)
!A3 = DS,3 ,"LINE",1 (1 FRES BLOD 1)
!A4 = DS,4 ,"LINE",1 (1 FRES BLOD 1)
!A5 = DS,5 ,"LINE",1 (1 FRES BLOD 1)
!A6 = DS,6 ,"LINE",1 (1 FRES BLOD 1)
!A7 = DS,7 ,"LINE",1 (1 FRES BLOD 1)
!A8 = DS,8 ,"LINE",1 (1 FRES BLOD 1)
!A9 = DS,9 ,"LINE",1 (1 FRES BLOD 1)
!A10 = DS,10,"LINE",1 (1 FRES BLOD 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE TO = 2 HEAT NODE 1 1

```

```

DEFINE TL = 5 EXPE HTMP 1 1
DEFINE ON = 7 ONES MATX 1 1
$
SU1 = SUM(TL,-1. TO)
SS1X = XTY(SU1,SU1)
!SS1A = DS 1,1,1 (3 SS1X AUS 1 1)
!SSTA = "SS1A"
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
J="LINE": "SSTA"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 107
$
*LABEL 8300
$
*RETURN
*SQUA
$
$-----$
$ Subroutine (29 ITER ATIO)ATIO: This subroutine updates the thermal$
$ properties of the parameters and $
$ executes the genetic processors $
$ n-times for n-generations. $
$-----$
$
*(29 ITER ATIO)ATIO
$
!ITERG = 0 $ initializing the counter for the loop
$
*LABEL 400
$
!ITERG = ITERG + 1 $ counter for the loop
$
*IF("ITERG" EQ 1): *GOTO 410
*GOTO 420
*LABEL 410
TABLE(NI=1,NJ=100):8 CONV CRIT
I=1:J=1,100:1.0
*LABEL 420
$
!SSQU = 2 $ flag for the SUMO SQUA subroutine
$
*DCALL (29 GENC PROC) $ call subroutine GENC PROC
*DCALL (29 SUMO SQUA) $ call subroutine SUMO SQUA
ERASE 2 $ erase content of library 2
ERASE 3 $ erase content of library 3
ERASE 4 $ erase content of library 4
ERASE 6 $ erase content of library 6
!SSQU = 3 $ flag for the SUMO SQUA subroutine
*DCALL (29 GEND PROC) $ call subroutine GEND PROC
*DCALL (29 SUMO SQUA) $ call subroutine SUMO SQUA
$
!MIN1 = DS,1,1,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN2 = DS,1,2,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN3 = DS,1,3,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN4 = DS,1,4,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN5 = DS,1,5,1 (1 NEWW PARA) $ reset the lower boundary for Cp

```

```

!MIN6 = DS,1,6,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN7 = DS,1,7,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN8 = DS,1,8,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN9 = DS,1,9,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MINX = DS,1,10,1 (1 NEWW PARA) $ reset the lower boundary for K
!MAX1 = DS,1,11,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX2 = DS,1,12,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX3 = DS,1,13,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX4 = DS,1,14,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX5 = DS,1,15,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX6 = DS,1,16,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX7 = DS,1,17,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX8 = DS,1,18,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX9 = DS,1,19,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAXX = DS,1,20,1 (1 NEWW PARA) $ reset the upper boundary for K
$
*DCALL (29 GEAL VARI) $call the subroutine GEAL VARI
  ERASE 2 $erase content of library 2
  ERASE 3 $erase content of library 3
  ERASE 4 $erase content of library 4
  ERASE 6 $erase content of library 6
*DCALL (29 GENE PROC) $call subroutine GENE PROC
$
!OBF1 = DS 1,1,1 (1 OBFN BLOD 1 1)
!OBF2 = DS 1,2,1 (1 OBFN BLOD 1 1)
!OBF3 = DS 1,3,1 (1 OBFN BLOD 1 1)
!OBF4 = DS 1,4,1 (1 OBFN BLOD 1 1)
!OBF5 = DS 1,5,1 (1 OBFN BLOD 1 1)
$
!OBF = (((!OBF1"+!OBF2"+!OBF3"+!OBF4"+!OBF5")/5)
$
TABLE,U(NI=1,NJ=100):8 CONV CRIT 1 1
  I=1:J="ITERG":!OBF"
$
*IF("ITERG" GE 2):*GOTO 500
*GOTO 510
*LABEL 500
!NUM1 = "ITERG" - 1
!NUM2 = "ITERG"
!CON1 = DS 1,"NUM1",1 (8 CONV CRIT 1 1)
!CON2 = DS 1,"NUM2",1 (8 CONV CRIT 1 1)
$
!CONV = ("CON2"-!CON1")/!CON2"
$
*IF("CONV" LE 0.05):*GOTO 520
*GOTO 510
*LABEL 520
*XQT DCU
PRINT 1 ITER PAPO
PRINT 1 ITER OBFN
*XQT EXIT
$
*LABEL 510
$
*IF("ITERG" EQ "ITMX"):GOTO 410
$
*GOTO 400
*LABEL 410
$

```

```

*RETURN
*ATIO
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.           $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrta time step for TRTB (seconds)
$
!MIN1 = 0.05     $ lower bound for skins (PCp)
!MIN2 = 0.05     $ lower bound for skins (K)
!MIN3 = 0.05     $ lower bound for honeycomb (PCp)
!MIN4 = 0.05     $ lower bound for honeycomb (K)
!MIN5 = 0.05     $ lower bound for torque tubes (PCp)
!MIN6 = 0.05     $ lower bound for torque tubes (K)
!MIN7 = 0.05     $ lower bound for facesheets (PCp)
!MIN8 = 0.05     $ lower bound for facesheets (K)
!MIN9 = 0.05     $ lower bound for air gap (PCp)
!MINX = 0.05     $ lower bound for air gap (K)
!MAX1 = 10.0     $ upper bound for skins (PCp)
!MAX2 = 10.0     $ upper bound for skins (K)
!MAX3 = 10.0     $ upper bound for honeycomb (PCp)
!MAX4 = 10.0     $ upper bound for honeycomb (K)
!MAX5 = 10.0     $ upper bound for torque tubes (PCp)
!MAX6 = 10.0     $ upper bound for torque tubes (K)
!MAX7 = 10.0     $ upper bound for facesheets (PCp)
!MAX8 = 10.0     $ upper bound for facesheets (K)
!MAX9 = 10.0     $ upper bound for air gap (PCp)
!MAXX = 10.0     $ upper bound for air gap (K)
$
!A1  = 2.461010  $ Initial value for skins (PCp)
!A2  = 1.801890  $ Initial value for skins (K)
!A3  = 0.337297  $ Initial value for honeycomb (PCp)
!A4  = 0.461608  $ Initial value for honeycomb (K)
!A5  = 1.935430  $ Initial value for torque tubes (PCp)
!A6  = 1.841770  $ Initial value for torque tubes (K)
!A7  = 0.596219  $ Initial value for facesheets (PCp)
!A8  = 2.545400  $ Initial value for facesheets (K)
!A9  = 1.00      $ Initial value for air gap (PCp)
!A10 = 1.00      $ Initial value for air gap (K)
$
!A1I = 2.54      $ Initial value for skins (PCp)
!A2I = 6.35      $ Initial value for skins (K)
!A3I = 1.72      $ Initial value for honeycomb (PCp)
!A4I = 0.15      $ Initial value for honeycomb (K)
!A5I = 1.26      $ Initial value for torque tubes (PCp)
!A6I = 5.66      $ Initial value for torque tubes (K)
!A7I = 9.57      $ Initial value for facesheets (PCp)
!A8I = 1.00      $ Initial value for facesheets (K)
!A9I = 1.00      $ Initial value for air gap (PCp)
!A10I = 1.00     $ Initial value for air gap (K)
$
*RETURN
*SPEC

```



```

$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT DCU                      $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1        $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1        $ copy the dataset from lib 4 to lib 2
$
!NWRD = TOC,NWDS(2 TRAN TEMP 1 1)
$
TOCC 2 TRAN TEMP 1 1:NJ="NWRD",NINJ="NWRD"
$
*XQT AUS                      $ arithmetic utility processor
$
!VAR1 = TOC,NWDS(2 TRAN TIME)  $ number of words in dataset TRAN TIME
!VAR2 = "NON1"-1              $ heated node number - 1
!VAR3 = "TOTN"-1              $ total node number - 1
$
DEFINE X = 2 TRAN TEMP 1 1
$
*IF("N2" EQ 0) :*GOTO 7000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 1) :*GOTO 7000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2) :*GOTO 7010 $ Temperature histories for initial conditions
*IF("N2" EQ 3) :*GOTO 7020 $ Temperature histories for property 1
*IF("N2" EQ 4) :*GOTO 7030 $ Temperature histories for property 2
*IF("N2" EQ 5) :*GOTO 7040 $ Temperature histories for property 3
*IF("N2" EQ 6) :*GOTO 7050 $ Temperature histories for property 4
*IF("N2" EQ 7) :*GOTO 7060 $ Temperature histories for property 5
*IF("N2" EQ 8) :*GOTO 7070 $ Temperature histories for property 6
*IF("N2" EQ 9) :*GOTO 7080 $ Temperature histories for property 7
*IF("N2" EQ 10):*GOTO 7090 $ Temperature histories for property 8
*IF("N2" EQ 11):*GOTO 7100 $ Temperature histories for property 9
*IF("N2" EQ 12):*GOTO 7110 $ Temperature histories for property 10
$
*LABEL 7000
TABLE(NI=1,NJ="VAR1"):2 HEAT NODE 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
$
DEFINE A = 5 EXPE HTMP 1 1
DEFINE B = 2 HEAT NODE 1 1
DEFINE C = 2 TRAN TIME 1 1
$
TABLE(NI=3,NJ="NDS"):5 TEMP ERAT 1 1
    TRAN(SOUR=C,ILIM=1,JLIM="NDS",DBASE=0,DSKIP=2)
    TRAN(SOUR=A,ILIM=1,JLIM="NDS",DBASE=1,DSKIP=2)
    TRAN(SOUR=B,ILIM=1,JLIM="NDS",DBASE=2,DSKIP=2)
$
*GOTO 8000
$
*LABEL 7010
TABLE(NI=1,NJ="VAR1"):2 INIT 00 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")

```

```

*GOTO 8000
$
*LABEL 7020
TABLE(NI=1,NJ="VAR1"):2 PROP I 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7030
TABLE(NI=1,NJ="VAR1"):2 PROP II 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7040
TABLE(NI=1,NJ="VAR1"):2 PROP III 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7050
TABLE(NI=1,NJ="VAR1"):2 PROP IV 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7060
TABLE(NI=1,NJ="VAR1"):2 PROP V 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7070
TABLE(NI=1,NJ="VAR1"):2 PROP VI 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7080
TABLE(NI=1,NJ="VAR1"):2 PROP VII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7090
TABLE(NI=1,NJ="VAR1"):2 PROP VIII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7100
TABLE(NI=1,NJ="VAR1"):2 PROP IX 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 7110
TABLE(NI=1,NJ="VAR1"):2 PROP X 1 1
    TRAN(SOUR=X,ILIM=1,JLIM="VAR1",SBASE="VAR2",SSKIP="VAR3")
*GOTO 8000
$
*LABEL 8000
*RETURN
*TEMP
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $

```

```

$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS
$
*TF OPEN 1'data
*TF READ 1
*TF CLOSE 1
$
!N2 = 1
$
*DCALL (29 MAKE VARI) $ Call subroutine MAKE VARI
*DCALL (29 PROB SPEC) $ Call subroutine PROB SPEC
*DCALL (29 GEAL VARI) $ Call subroutine GEAL VARI
*DCALL (29 PLAT DIME) $ Call subroutine PLAT DIME
*DCALL (29 NODE GENE) $ Call subroutine NODE GENE
*DCALL (29 MAKE ELEM) $ Call subroutine MAKE ELEM
*DCALL (29 BOUN COND) $ Call subroutine BOUN COND
!ITPP = 1          $ Reset parameter for initial population 1
!SSQU = 1
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 2
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 3
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 4
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 5
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6

```

```
$  
!ITMX = 10           $ Maximum number of iteration  
*DCALL (29 ITER ATIO) $ Call subroutine ITER ATIO  
$  
*XQT DCU  
  PRINT 1 ITER PAPO  
  PRINT 1 ITER OBFN  
*XQT EXIT
```

Appendix M

EAL Two-Dimensional Model: Estimation Program

```
*XQT U1
!MAXCM = SSP(0,13)
*CMPARA (MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of $
$                               the variables used in the program $
$                               such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each $
$                               section $
$                               Note: NX2 can vary from 2 to ... $
$                               NX1 can vary from 3 to ... $
$                               NX3 can vary from 3 to ... $
$                               NX5 can vary from 3 to ... $
$                               NX7 can vary from 3 to ... $
$                               NX5 should be odd $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $arithmetic utility system processor
$
!NX1  = 8           $ number of nodes for left honeycomb edge closeout
!NX2  = 2           $ number of nodes for rib 1
!NX3  = 30          $ number of nodes for torque tube 1
!NX4  = "NX2"       $ number of nodes for rib 2
!NX5  = 33          $ number of nodes for torque tube 2
!NX6  = "NX2"       $ number of nodes for rib 3
!NX7  = 38          $ number of nodes for torque tube 3
!NX8  = "NX2"       $ number of nodes for rib 4
!NX9  = 8           $ number of nodes for right honeycomb edge closeout
$
!EX1  = "NX1"-1     $ number of elements for left honeycomb edge closeout
!EX2  = "NX2"-1     $ number of elements for rib 1
```

```

!EX3 = "NX3"-1      $ number of elements for torque tube 1
!EX4 = "NX4"-1      $ number of elements for rib 2
!EX5 = "NX5"-1      $ number of elements for torque tube 2
!EX6 = "NX6"-1      $ number of elements for rib 3
!EX7 = "NX7"-1      $ number of elements for torque tube 3
!EX8 = "NX8"-1      $ number of elements for rib 4
!EX9 = "NX9"-1      $ number of elements for right honeycomb edge closeout
$
!NY1 = 2            $ number of nodes for the lower outer facesheet
!NY2 = 9            $ number of nodes for the lower honeycomb structure
!NY3 = 3            $ number of nodes for the lower inner facesheet
!NY4 = 2            $ number of nodes for the lower torque tube
!NY5 = 17           $ number of nodes for the air cavity
!NY6 = 2            $ number of nodes for the upper torque tube
!NY7 = 3            $ number of nodes for the upper inner facesheet
!NY8 = 9            $ number of nodes for the upper honeycomb structure
!NY9 = 2            $ number of nodes for the upper outer facesheet
!NY10 = 9           $ number of nodes for the insulation
$
!EY1 = "NY1"-1      $ number of elements for the lower outer facesheet
!EY2 = "NY2"-1      $ number of elements for the lower honeycomb structure
!EY3 = "NY3"-1      $ number of elements for the lower inner facesheet
!EY4 = "NY4"-1      $ number of elements for the lower torque tube
!EY5 = "NY5"-1      $ number of elements for the air cavity
!EY6 = "NY6"-1      $ number of elements for the upper torque tube
!EY7 = "NY7"-1      $ number of elements for the upper inner facesheet
!EY8 = "NY8"-1      $ number of elements for the upper honeycomb structure
!EY9 = "NY9"-1      $ number of elements for the upper outer facesheet
!EY10 = "NY10"-1    $ number of elements for the insulation
$
!XJ = 1             $ nodal increment in the x-direction
$
!YJ = "NX1"+"EX2"+"EX3"+"EX4"+"EX5"+"EX6"+"EX7"+"EX8"+"EX9"
$                  $ nodal increment in the y-direction
!YJE = "EX1"+"EX2"+"EX3"+"EX4"+"EX5"+"EX6"+"EX7"+"EX8"+"EX9"
$                  $ nodal increment in the y-direction -1
!TOTN = "YJ"*( "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"+"EY10")
$                  $total number of nodes in the model
!NON1 = "YJ"*( "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9")
$                  $node number at the heater
!NON2 = "NY1"+"EY2"+"EY3"+"EY4"
$                  $node number at the inside of the lower torque tube
!NON3 = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"
$                  $node number at the inside of the upper torque tube
!NONJ = FLOAT("TOTN")$node number on top of the insulation
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 GEAL VARI)GEAL: This subroutine defines some of $
$                               the variables used in the genetic $
$                               algorithm processors. $
$-----$
$
*(29 GEAL VARI) GEAL
*XQT AUS
$
TABLE(NI=1,NJ=9,TYPE=0) : INPT INTE

```

```

I=1
J=1: 10 $ NC : number of estimated parameters
J=2: 5 $ NPOP : number of initial populations to investigate
J=3: 15287 $ CFSEED1 : parameter used by random number generator
J=4: 9428 $ CFSEED2 : parameter used by random number generator
J=5: 15 $ SLOPSEED : parameter used by random number generator
J=6: 10 $ NBEST1 : # chro. kept from NPOP initial population
J=7: 40 $ NBEST2 : # chro. kept from parent children population
J=8: 30 $ NBEST3 : # chro. kept from parent children population
J=9: 10 $ ITMX : maximum number of generations to perform
$
TABLE(NI=1,NJ=21,TYPE=-2) : INPT REAL
I=1
J=1 : "MIN1" $ PARMIN1 : CP lower limit for skins
J=2 : "MIN2" $ PARMIN2 : K lower limit for skins
J=3 : "MIN3" $ PARMIN3 : CP lower limit for honeycomb
J=4 : "MIN4" $ PARMIN4 : K lower limit for honeycomb
J=5 : "MIN5" $ PARMIN5 : CP lower limit for torque tubes
J=6 : "MIN6" $ PARMIN6 : K lower limit for torque tubes
J=7 : "MIN7" $ PARMIN7 : CP lower limit for outer facesheets
J=8 : "MIN8" $ PARMIN8 : K lower limit for outer facesheets
J=9 : "MIN9" $ PARMIN9 : CP lower limit for air gap
J=10: "MINX" $ PARMINX : K lower limit for air gap
J=11: "MAX1" $ PARMAX1 : CP upper limit for skins
J=12: "MAX2" $ PARMAX2 : K upper limit for skins
J=13: "MAX3" $ PARMAX3 : CP upper limit for honeycomb
J=14: "MAX4" $ PARMAX4 : K upper limit for honeycomb
J=15: "MAX5" $ PARMAX5 : CP upper limit for torque tubes
J=16: "MAX6" $ PARMAX6 : K upper limit for torque tubes
J=17: "MAX7" $ PARMAX7 : CP upper limit for outer facesheets
J=18: "MAX8" $ PARMAX8 : K upper limit for outer facesheets
J=19: "MAX9" $ PARMAX9 : CP upper limit for air gap
J=20: "MAXX" $ PARMAX10 : K upper limit for air gap
J=21: 0.05 $ PM : probability of mutation
$
*RETURN
*GEAL
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the $
$ properties of the materials used $
$ in the model. $
$-----$
$
$(29 MATE PROP)PROP
$
*XQT AUS
$
$ conduction properties for IM7/954-2A
$ upper and lower skins
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4 5
J=1: 1.E+6 1.1688213 1.5596224 "A1"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
J=1: 0. 0. 0. 0.
$

```

```

$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
      J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
$ conduction properties for air
$
TABLE(NI=9,NJ=1):6 COND PROP 4:I=2 3 4 5
      J=1: 1.E+3 "A2" "A3" "A3"
$
      J=1: 0.9950 1009.0 0.7025
$
$
$TABLE(NI=9,NJ=14):6 COND PROP 4:I=1 2 3 4 5 6
$      J= 1: 250.0 1.3947 1006. 0.1758 0.1758 0.1758
$      J= 2: 300.0 1.1614 1007. 0.2916 0.2916 0.2916
$      J= 3: 350.0 0.9950 1009. 0.4513 0.4513 0.4513
$      J= 4: 400.0 0.8711 1014. 0.6627 0.6627 0.6627
$      J= 5: 450.0 0.7740 1021. 0.9327 0.9327 0.9327
$      J= 6: 500.0 0.6964 1030. 1.2690 1.2690 1.2690
$      J= 7: 550.0 0.6329 1040. 1.6788 1.6788 1.6788
$      J= 8: 600.0 0.5804 1051. 2.1694 2.1694 2.1694
$      J= 9: 650.0 0.5356 1063. 2.7483 2.7483 2.7483
$      J=10: 700.0 0.4975 1075. 3.4229 3.4229 3.4229
$      J=11: 750.0 0.4643 1087. 4.2005 4.2005 4.2005
$      J=12: 800.0 0.4354 1099. 5.0885 5.0885 5.0885
$      J=13: 850.0 0.4097 1110. 6.0943 6.0943 6.0943
$      J=14: 900.0 0.3868 1121. 7.2256 7.2256 7.2256
$
$ conduction properties for SS321 honeycomb (actually SS347)
$
TABLE(NI=9,NJ=1):6 COND PROP 5: I = 2 3 4 5
      J=1: 1.E+6 0.7037244 0.4064743 "A4"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tube tops, edge channel tops
$
TABLE(NI=9,NJ=1):6 COND PROP 6: I= 2 3 4 5
      J=1: 1.E+6 2.9149498 1.9172322 "A5"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tube ribs, edge channel ribs
$
TABLE(NI=9,NJ=1):6 COND PROP 8: I= 2 3 4 5
      J=1: 1.E+6 2.9149498 "A5" 1.9172322
$
$ conduction properties for PETI-5 of the upper and lower outer
$ facesheets.
$
TABLE(NI=9,NJ=1):6 COND PROP 7: I= 2 3 4 5
      J=1: 1.E+6 2.0342782 1.6259311 "A6"
$
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 LOWE FACE)LFAC: This subroutine defines the points $
$                               of the lower outer facesheet to $
$                               make the nodes. $
$-----$
$

```



```

*(29 LOWE FACE)LFAC
$
!X1  = 0.
!X2  = 0.0285
!X3  = 0.0328
!X4  = 0.1560
!X5  = 0.1602
!X6  = 0.2905
!X7  = 0.2947
!X8  = 0.4555
!X9  = 0.4597
!X10 = 0.4883
!X11 = 0.
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
$
!X1A = 0.      +(0.0285/"EX1")
!X2A = 0.0285+(0.0043/"EX2")
!X3A = 0.0328+(0.1232/"EX3")
!X4A = 0.1560+(0.0042/"EX4")
!X5A = 0.1602+(0.1345/"EX5")
!X6A = 0.2905+(0.0042/"EX6")
!X7A = 0.2947+(0.1608/"EX7")
!X8A = 0.4555+(0.0042/"EX8")
!X9A = 0.4597+(0.0286/"EX9")
!X11A= 0.      +(0.0285/"EX1")
!X12A= 0.0285+(0.0043/"EX2")
!X13A= 0.0328+(0.1232/"EX3")
!X14A= 0.1560+(0.0042/"EX4")
!X15A= 0.1602+(0.1345/"EX5")
!X16A= 0.2905+(0.0042/"EX6")
!X17A= 0.2947+(0.1608/"EX7")
!X18A= 0.4555+(0.0042/"EX8")
!X19A= 0.4597+(0.0286/"EX9")
$
!Y1  = 0.
!Y2  = 0.
!Y3  = 0.
!Y4  = 0.
!Y5  = 0.
!Y6  = 0.
!Y7  = 0.
!Y8  = 0.
!Y9  = 0.
!Y10 = 0.
!Y11 = 0.0032
!Y12 = 0.0032
!Y13 = 0.0032
!Y14 = 0.0032
!Y15 = 0.0032
!Y16 = 0.0032
!Y17 = 0.0032

```

```

!Y18 = 0.0032
!Y19 = 0.0032
!Y20 = 0.0032
$
!Z1 = 0.
!Z2 = 0.
!Z3 = 0.
!Z4 = 0.
!Z5 = 0.
!Z6 = 0.
!Z7 = 0.
!Z8 = 0.
!Z9 = 0.
!Z10 = 0.
!Z11 = 0.
!Z12 = 0.
!Z13 = 0.
!Z14 = 0.
!Z15 = 0.
!Z16 = 0.
!Z17 = 0.
!Z18 = 0.
!Z19 = 0.
!Z20 = 0.
$
*RETURN
*LFAC
$
$-----$
$ Subroutine (29 LOWE HONY)LHON: This subroutine defines the points $
$                               of the lower honeycomb structure $
$                               to make the nodes. $
$-----$
$
*(29 LOWE HONY)LHON
$
!X11 = 0.
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
!X21 = 0.
!X22 = 0.0285
!X23 = 0.0328
!X24 = 0.1560
!X25 = 0.1602
!X26 = 0.2905
!X27 = 0.2947
!X28 = 0.4555
!X29 = 0.4597
!X30 = 0.4883
$
!X11A= 0.      +(0.0285/"EX1")
!X12A= 0.0285+(0.0043/"EX2")

```

```

!X13A= 0.0328+(0.1232/"EX3")
!X14A= 0.1560+(0.0042/"EX4")
!X15A= 0.1602+(0.1345/"EX5")
!X16A= 0.2905+(0.0042/"EX6")
!X17A= 0.2947+(0.1608/"EX7")
!X18A= 0.4555+(0.0042/"EX8")
!X19A= 0.4597+(0.0286/"EX9")
!X21A= 0.      +(0.0285/"EX1")
!X22A= 0.0285+(0.0043/"EX2")
!X23A= 0.0328+(0.1232/"EX3")
!X24A= 0.1560+(0.0042/"EX4")
!X25A= 0.1602+(0.1345/"EX5")
!X26A= 0.2905+(0.0042/"EX6")
!X27A= 0.2947+(0.1608/"EX7")
!X28A= 0.4555+(0.0042/"EX8")
!X29A= 0.4597+(0.0286/"EX9")
$
!Y11 = 0.0032
!Y12 = 0.0032
!Y13 = 0.0032
!Y14 = 0.0032
!Y15 = 0.0032
!Y16 = 0.0032
!Y17 = 0.0032
!Y18 = 0.0032
!Y19 = 0.0032
!Y20 = 0.0032
!Y21 = 0.0286
!Y22 = 0.0286
!Y23 = 0.0286
!Y24 = 0.0286
!Y25 = 0.0286
!Y26 = 0.0286
!Y27 = 0.0286
!Y28 = 0.0286
!Y29 = 0.0286
!Y30 = 0.0286
$
!Y11B= 0.0032+(0.0254/"EY2")
!Y12B= 0.0032+(0.0254/"EY2")
!Y13B= 0.0032+(0.0254/"EY2")
!Y14B= 0.0032+(0.0254/"EY2")
!Y15B= 0.0032+(0.0254/"EY2")
!Y16B= 0.0032+(0.0254/"EY2")
!Y17B= 0.0032+(0.0254/"EY2")
!Y18B= 0.0032+(0.0254/"EY2")
!Y19B= 0.0032+(0.0254/"EY2")
!Y20B= 0.0032+(0.0254/"EY2")
$
!Z11 = 0.
!Z12 = 0.
!Z13 = 0.
!Z14 = 0.
!Z15 = 0.
!Z16 = 0.
!Z17 = 0.
!Z18 = 0.
!Z19 = 0.
!Z20 = 0.

```

```

!Z21 = 0.
!Z22 = 0.
!Z23 = 0.
!Z24 = 0.
!Z25 = 0.
!Z26 = 0.
!Z27 = 0.
!Z28 = 0.
!Z29 = 0.
!Z30 = 0.
$
*RETURN
*LHON
$
$-----$
$ Subroutine (29 LOWE SKIN)LSKI: This subroutine defines the points $
$                               of the lower skin to make the $
$                               nodes. $
$-----$
$
$
*(29 LOWE SKIN)LSKI
$
!X21 = 0.
!X22 = 0.0285
!X23 = 0.0328
!X24 = 0.1560
!X25 = 0.1602
!X26 = 0.2905
!X27 = 0.2947
!X28 = 0.4555
!X29 = 0.4597
!X30 = 0.4883
!X31 = 0.
!X32 = 0.0285
!X33 = 0.0328
!X34 = 0.1560
!X35 = 0.1602
!X36 = 0.2905
!X37 = 0.2947
!X38 = 0.4555
!X39 = 0.4597
!X40 = 0.4883
$
!X21A= 0.      +(0.0285/"EX1")
!X22A= 0.0285+(0.0043/"EX2")
!X23A= 0.0328+(0.1232/"EX3")
!X24A= 0.1560+(0.0042/"EX4")
!X25A= 0.1602+(0.1345/"EX5")
!X26A= 0.2905+(0.0042/"EX6")
!X27A= 0.2947+(0.1608/"EX7")
!X28A= 0.4555+(0.0042/"EX8")
!X29A= 0.4597+(0.0286/"EX9")
!X31A= 0.      +(0.0285/"EX1")
!X32A= 0.0285+(0.0043/"EX2")
!X33A= 0.0328+(0.1232/"EX3")
!X34A= 0.1560+(0.0042/"EX4")
!X35A= 0.1602+(0.1345/"EX5")
!X36A= 0.2905+(0.0042/"EX6")
!X37A= 0.2947+(0.1608/"EX7")

```

```

!X38A= 0.4555+(0.0042/"EX8")
!X39A= 0.4597+(0.0286/"EX9")
$
!Y21 = 0.0286
!Y22 = 0.0286
!Y23 = 0.0286
!Y24 = 0.0286
!Y25 = 0.0286
!Y26 = 0.0286
!Y27 = 0.0286
!Y28 = 0.0286
!Y29 = 0.0286
!Y30 = 0.0286
!Y31 = 0.0326
!Y32 = 0.0326
!Y33 = 0.0326
!Y34 = 0.0326
!Y35 = 0.0326
!Y36 = 0.0326
!Y37 = 0.0326
!Y38 = 0.0326
!Y39 = 0.0326
!Y40 = 0.0326
$
!Y21B= 0.0286+(0.0040/"EY3")
!Y22B= 0.0286+(0.0040/"EY3")
!Y23B= 0.0286+(0.0040/"EY3")
!Y24B= 0.0286+(0.0040/"EY3")
!Y25B= 0.0286+(0.0040/"EY3")
!Y26B= 0.0286+(0.0040/"EY3")
!Y27B= 0.0286+(0.0040/"EY3")
!Y28B= 0.0286+(0.0040/"EY3")
!Y29B= 0.0286+(0.0040/"EY3")
!Y30B= 0.0286+(0.0040/"EY3")
$
!Z21 = 0.
!Z22 = 0.
!Z23 = 0.
!Z24 = 0.
!Z25 = 0.
!Z26 = 0.
!Z27 = 0.
!Z28 = 0.
!Z29 = 0.
!Z30 = 0.
!Z31 = 0.
!Z32 = 0.
!Z33 = 0.
!Z34 = 0.
!Z35 = 0.
!Z36 = 0.
!Z37 = 0.
!Z38 = 0.
!Z39 = 0.
!Z40 = 0.
$
*RETURN
*LSKI
$

```

```

$-----$
$ Subroutine (29 LOWE TORQ)LTOR: This subroutine defines the points $
$                               of the lower torque tubes.          $
$-----$
$
*(29 LOWE TORQ)LTOR
$
!X1  = 0.
!X2  = 0.0285
!X3  = 0.0328
!X4  = 0.1560
!X5  = 0.1602
!X6  = 0.2905
!X7  = 0.2947
!X8  = 0.4555
!X9  = 0.4597
!X10 = 0.4883
!X11 = 0.
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
$
!X1A = 0.      +(0.0285/"EX1")
!X2A = 0.0285+(0.0043/"EX2")
!X3A = 0.0328+(0.1232/"EX3")
!X4A = 0.1560+(0.0042/"EX4")
!X5A = 0.1602+(0.1345/"EX5")
!X6A = 0.2905+(0.0042/"EX6")
!X7A = 0.2947+(0.1608/"EX7")
!X8A = 0.4555+(0.0042/"EX8")
!X9A = 0.4597+(0.0286/"EX9")
!X11A= 0.      +(0.0285/"EX1")
!X12A= 0.0285+(0.0043/"EX2")
!X13A= 0.0328+(0.1232/"EX3")
!X14A= 0.1560+(0.0042/"EX4")
!X15A= 0.1602+(0.1345/"EX5")
!X16A= 0.2905+(0.0042/"EX6")
!X17A= 0.2947+(0.1608/"EX7")
!X18A= 0.4555+(0.0042/"EX8")
!X19A= 0.4597+(0.0286/"EX9")
$
!Y1  = 0.0326
!Y2  = 0.0326
!Y3  = 0.0326
!Y4  = 0.0326
!Y5  = 0.0326
!Y6  = 0.0326
!Y7  = 0.0326
!Y8  = 0.0326
!Y9  = 0.0326
!Y10 = 0.0326
!Y11 = 0.0347
!Y12 = 0.0347

```

```

!Y13 = 0.0347
!Y14 = 0.0347
!Y15 = 0.0347
!Y16 = 0.0347
!Y17 = 0.0347
!Y18 = 0.0347
!Y19 = 0.0347
!Y20 = 0.0347
$
!Y1B = 0.0326+(0.0021/"EY4")
!Y2B = 0.0326+(0.0021/"EY4")
!Y3B = 0.0326+(0.0021/"EY4")
!Y4B = 0.0326+(0.0021/"EY4")
!Y5B = 0.0326+(0.0021/"EY4")
!Y6B = 0.0326+(0.0021/"EY4")
!Y7B = 0.0326+(0.0021/"EY4")
!Y8B = 0.0326+(0.0021/"EY4")
!Y9B = 0.0326+(0.0021/"EY4")
!Y10B= 0.0326+(0.0021/"EY4")
$
!Z1  = 0.
!Z2  = 0.
!Z3  = 0.
!Z4  = 0.
!Z5  = 0.
!Z6  = 0.
!Z7  = 0.
!Z8  = 0.
!Z9  = 0.
!Z10 = 0.
!Z11 = 0.
!Z12 = 0.
!Z13 = 0.
!Z14 = 0.
!Z15 = 0.
!Z16 = 0.
!Z17 = 0.
!Z18 = 0.
!Z19 = 0.
!Z20 = 0.
$
*RETURN
*LTOR
$
$-----$
$ Subroutine (29 AIRG SECT)AIRG: This subroutine defines the points $
$                               of the airgap existing between the $
$                               torque tubes, the edge channels... $
$-----$
$
$(29 AIRG SECT)AIRG
$
!X31 = 0.
!X32 = 0.0285
!X33 = 0.0328
!X34 = 0.1560
!X35 = 0.1602
!X36 = 0.2905
!X37 = 0.2947

```

```

!X38 = 0.4555
!X39 = 0.4597
!X40 = 0.4883
!X41 = 0.
!X42 = 0.0285
!X43 = 0.0328
!X44 = 0.1560
!X45 = 0.1602
!X46 = 0.2905
!X47 = 0.2947
!X48 = 0.4555
!X49 = 0.4597
!X50 = 0.4883
$
!X31A= 0.      +(0.0285/"EX1")
!X32A= 0.0285+(0.0043/"EX2")
!X33A= 0.0328+(0.1232/"EX3")
!X34A= 0.1560+(0.0042/"EX4")
!X35A= 0.1602+(0.1345/"EX5")
!X36A= 0.2905+(0.0042/"EX6")
!X37A= 0.2947+(0.1608/"EX7")
!X38A= 0.4555+(0.0042/"EX8")
!X39A= 0.4597+(0.0286/"EX9")
!X41A= 0.      +(0.0285/"EX1")
!X42A= 0.0285+(0.0043/"EX2")
!X43A= 0.0328+(0.1232/"EX3")
!X44A= 0.1560+(0.0042/"EX4")
!X45A= 0.1602+(0.1345/"EX5")
!X46A= 0.2905+(0.0042/"EX6")
!X47A= 0.2947+(0.1608/"EX7")
!X48A= 0.4555+(0.0042/"EX8")
!X49A= 0.4597+(0.0286/"EX9")
$
!Y31 = 0.0347
!Y32 = 0.0347
!Y33 = 0.0347
!Y34 = 0.0347
!Y35 = 0.0347
!Y36 = 0.0347
!Y37 = 0.0347
!Y38 = 0.0347
!Y39 = 0.0347
!Y40 = 0.0347
!Y41 = 0.0899
!Y42 = 0.0871
!Y43 = 0.0867
!Y44 = 0.0747
!Y45 = 0.0743
!Y46 = 0.0616
!Y47 = 0.0612
!Y48 = 0.0456
!Y49 = 0.0452
!Y50 = 0.0424
$
!Y42A= (0.0899-(0.0973*"X42A"))
!Y43A= (0.0899-(0.0973*"X43A"))
!Y44A= (0.0899-(0.0973*"X44A"))
!Y45A= (0.0899-(0.0973*"X45A"))
!Y46A= (0.0899-(0.0973*"X46A"))

```



```

!Y47A= (0.0899-(0.0973*"X47A"))
!Y48A= (0.0899-(0.0973*"X48A"))
!Y49A= (0.0899-(0.0973*"X49A"))
$
!Y31B= 0.0347+(0.0552/"EY5")
!Y32B= 0.0347+(0.0524/"EY5")
!Y33B= 0.0347+(0.0520/"EY5")
!Y34B= 0.0347+(0.0400/"EY5")
!Y35B= 0.0347+(0.0396/"EY5")
!Y36B= 0.0347+(0.0269/"EY5")
!Y37B= 0.0347+(0.0265/"EY5")
!Y38B= 0.0347+(0.0109/"EY5")
!Y39B= 0.0347+(0.0105/"EY5")
!Y40B= 0.0347+(0.0077/"EY5")
$
!Y32C= 0.0347+(((0.0899-(0.0973*"X32A"))-0.0347)/"EY5")
!Y33C= 0.0347+(((0.0899-(0.0973*"X33A"))-0.0347)/"EY5")
!Y34C= 0.0347+(((0.0899-(0.0973*"X34A"))-0.0347)/"EY5")
!Y35C= 0.0347+(((0.0899-(0.0973*"X35A"))-0.0347)/"EY5")
!Y36C= 0.0347+(((0.0899-(0.0973*"X36A"))-0.0347)/"EY5")
!Y37C= 0.0347+(((0.0899-(0.0973*"X37A"))-0.0347)/"EY5")
!Y38C= 0.0347+(((0.0899-(0.0973*"X38A"))-0.0347)/"EY5")
!Y39C= 0.0347+(((0.0899-(0.0973*"X39A"))-0.0347)/"EY5")
$
!Z31 = 0.
!Z32 = 0.
!Z33 = 0.
!Z34 = 0.
!Z35 = 0.
!Z36 = 0.
!Z37 = 0.
!Z38 = 0.
!Z39 = 0.
!Z40 = 0.
!Z41 = 0.
!Z42 = 0.
!Z43 = 0.
!Z44 = 0.
!Z45 = 0.
!Z46 = 0.
!Z47 = 0.
!Z48 = 0.
!Z49 = 0.
!Z50 = 0.
$
*RETURN
*AIRG
$
$-----$
$ Subroutine (29 UPPE TORQ)UTOR: This subroutine defines the points $
$                               of the upper torque tube to make $
$                               the nodes.                          $
$-----$
$
*(29 UPPE TORQ)UTOR
$
!X41 = 0.
!X42 = 0.0285
!X43 = 0.0328

```

```

!X44 = 0.1560
!X45 = 0.1602
!X46 = 0.2905
!X47 = 0.2947
!X48 = 0.4555
!X49 = 0.4597
!X50 = 0.4883
!X51 = 0.
!X52 = 0.0285
!X53 = 0.0328
!X54 = 0.1560
!X55 = 0.1602
!X56 = 0.2905
!X57 = 0.2947
!X58 = 0.4555
!X59 = 0.4597
!X60 = 0.4883
$
!X41A= 0.      +(0.0285/"EX1")
!X42A= 0.0285+(0.0043/"EX2")
!X43A= 0.0328+(0.1232/"EX3")
!X44A= 0.1560+(0.0042/"EX4")
!X45A= 0.1602+(0.1345/"EX5")
!X46A= 0.2905+(0.0042/"EX6")
!X47A= 0.2947+(0.1608/"EX7")
!X48A= 0.4555+(0.0042/"EX8")
!X49A= 0.4597+(0.0286/"EX9")
!X51A= 0.      +(0.0285/"EX1")
!X52A= 0.0285+(0.0043/"EX2")
!X53A= 0.0328+(0.1232/"EX3")
!X54A= 0.1560+(0.0042/"EX4")
!X55A= 0.1602+(0.1345/"EX5")
!X56A= 0.2905+(0.0042/"EX6")
!X57A= 0.2947+(0.1608/"EX7")
!X58A= 0.4555+(0.0042/"EX8")
!X59A= 0.4597+(0.0286/"EX9")
$
!Y41 = 0.0899
!Y42 = 0.0871
!Y43 = 0.0867
!Y44 = 0.0747
!Y45 = 0.0743
!Y46 = 0.0616
!Y47 = 0.0612
!Y48 = 0.0456
!Y49 = 0.0452
!Y50 = 0.0424
!Y51 = 0.0920
!Y52 = 0.0892
!Y53 = 0.0888
!Y54 = 0.0768
!Y55 = 0.0764
!Y56 = 0.0637
!Y57 = 0.0633
!Y58 = 0.0477
!Y59 = 0.0473
!Y60 = 0.0445
$
!Y52A= (0.0920-(0.0973*"X52A"))

```

```

!Y53A= (0.0920-(0.0973*"X53A"))
!Y54A= (0.0920-(0.0973*"X54A"))
!Y55A= (0.0920-(0.0973*"X55A"))
!Y56A= (0.0920-(0.0973*"X56A"))
!Y57A= (0.0920-(0.0973*"X57A"))
!Y58A= (0.0920-(0.0973*"X58A"))
!Y59A= (0.0920-(0.0973*"X59A"))
$
!Y41B= 0.0899+(0.0021/"EY6")
!Y42B= 0.0871+(0.0021/"EY6")
!Y43B= 0.0867+(0.0021/"EY6")
!Y44B= 0.0747+(0.0021/"EY6")
!Y45B= 0.0743+(0.0021/"EY6")
!Y46B= 0.0616+(0.0021/"EY6")
!Y47B= 0.0612+(0.0021/"EY6")
!Y48B= 0.0456+(0.0021/"EY6")
!Y49B= 0.0452+(0.0021/"EY6")
!Y50B= 0.0424+(0.0021/"EY6")
$
!Y42C= (0.0899-(0.0973*"X42A"))+(0.0021/"EY6")
!Y43C= (0.0899-(0.0973*"X43A"))+(0.0021/"EY6")
!Y44C= (0.0899-(0.0973*"X44A"))+(0.0021/"EY6")
!Y45C= (0.0899-(0.0973*"X45A"))+(0.0021/"EY6")
!Y46C= (0.0899-(0.0973*"X46A"))+(0.0021/"EY6")
!Y47C= (0.0899-(0.0973*"X47A"))+(0.0021/"EY6")
!Y48C= (0.0899-(0.0973*"X48A"))+(0.0021/"EY6")
!Y49C= (0.0899-(0.0973*"X49A"))+(0.0021/"EY6")
$
!Z41 = 0.
!Z42 = 0.
!Z43 = 0.
!Z44 = 0.
!Z45 = 0.
!Z46 = 0.
!Z47 = 0.
!Z48 = 0.
!Z49 = 0.
!Z50 = 0.
!Z51 = 0.
!Z52 = 0.
!Z53 = 0.
!Z54 = 0.
!Z55 = 0.
!Z56 = 0.
!Z57 = 0.
!Z58 = 0.
!Z59 = 0.
!Z60 = 0.
$
*RETURN
*UTOR
$
$-----$
$ Subroutine (29 UPPE SKIN)USKI: This subroutine defines the points $
$                               of the upper skin to make           $
$                               the nodes.                          $
$-----$
$
*(29 UPPE SKIN)USKI

```

```

$
!X41 = 0.
!X42 = 0.0285
!X43 = 0.0328
!X44 = 0.1560
!X45 = 0.1602
!X46 = 0.2905
!X47 = 0.2947
!X48 = 0.4555
!X49 = 0.4597
!X50 = 0.4883
!X51 = 0.
!X52 = 0.0285
!X53 = 0.0328
!X54 = 0.1560
!X55 = 0.1602
!X56 = 0.2905
!X57 = 0.2947
!X58 = 0.4555
!X59 = 0.4597
!X60 = 0.4883
$
!X41A= 0.      +(0.0285/"EX1")
!X42A= 0.0285+(0.0043/"EX2")
!X43A= 0.0328+(0.1232/"EX3")
!X44A= 0.1560+(0.0042/"EX4")
!X45A= 0.1602+(0.1345/"EX5")
!X46A= 0.2905+(0.0042/"EX6")
!X47A= 0.2947+(0.1608/"EX7")
!X48A= 0.4555+(0.0042/"EX8")
!X49A= 0.4597+(0.0286/"EX9")
!X51A= 0.      +(0.0285/"EX1")
!X52A= 0.0285+(0.0043/"EX2")
!X53A= 0.0328+(0.1232/"EX3")
!X54A= 0.1560+(0.0042/"EX4")
!X55A= 0.1602+(0.1345/"EX5")
!X56A= 0.2905+(0.0042/"EX6")
!X57A= 0.2947+(0.1608/"EX7")
!X58A= 0.4555+(0.0042/"EX8")
!X59A= 0.4597+(0.0286/"EX9")
$
!Y41 = 0.0920
!Y42 = 0.0892
!Y43 = 0.0888
!Y44 = 0.0768
!Y45 = 0.0764
!Y46 = 0.0637
!Y47 = 0.0633
!Y48 = 0.0477
!Y49 = 0.0473
!Y50 = 0.0445
!Y51 = 0.0949
!Y52 = 0.0921
!Y53 = 0.0917
!Y54 = 0.0797
!Y55 = 0.0793
!Y56 = 0.0666
!Y57 = 0.0662
!Y58 = 0.0506

```

```

!Y59 = 0.0502
!Y60 = 0.0474
$
!Y52A= (0.0949-(0.0973*"X52A"))
!Y53A= (0.0949-(0.0973*"X53A"))
!Y54A= (0.0949-(0.0973*"X54A"))
!Y55A= (0.0949-(0.0973*"X55A"))
!Y56A= (0.0949-(0.0973*"X56A"))
!Y57A= (0.0949-(0.0973*"X57A"))
!Y58A= (0.0949-(0.0973*"X58A"))
!Y59A= (0.0949-(0.0973*"X59A"))
$
!Y41B= 0.0920+(0.0029/"EY7")
!Y42B= 0.0892+(0.0029/"EY7")
!Y43B= 0.0888+(0.0029/"EY7")
!Y44B= 0.0768+(0.0029/"EY7")
!Y45B= 0.0764+(0.0029/"EY7")
!Y46B= 0.0637+(0.0029/"EY7")
!Y47B= 0.0633+(0.0029/"EY7")
!Y48B= 0.0477+(0.0029/"EY7")
!Y49B= 0.0473+(0.0029/"EY7")
!Y50B= 0.0445+(0.0029/"EY7")
$
!Y42C= (0.0920-(0.0973*"X42A"))+(0.0029/"EY7")
!Y43C= (0.0920-(0.0973*"X43A"))+(0.0029/"EY7")
!Y44C= (0.0920-(0.0973*"X44A"))+(0.0029/"EY7")
!Y45C= (0.0920-(0.0973*"X45A"))+(0.0029/"EY7")
!Y46C= (0.0920-(0.0973*"X46A"))+(0.0029/"EY7")
!Y47C= (0.0920-(0.0973*"X47A"))+(0.0029/"EY7")
!Y48C= (0.0920-(0.0973*"X48A"))+(0.0029/"EY7")
!Y49C= (0.0920-(0.0973*"X49A"))+(0.0029/"EY7")
$
!Z41 = 0.
!Z42 = 0.
!Z43 = 0.
!Z44 = 0.
!Z45 = 0.
!Z46 = 0.
!Z47 = 0.
!Z48 = 0.
!Z49 = 0.
!Z50 = 0.
!Z51 = 0.
!Z52 = 0.
!Z53 = 0.
!Z54 = 0.
!Z55 = 0.
!Z56 = 0.
!Z57 = 0.
!Z58 = 0.
!Z59 = 0.
!Z60 = 0.
$
*RETURN
*USKI
$
$-----$
$ Subroutine (29 UPPE HONY)UHON: This subroutine defines the points $
$                                     of the upper honeycomb structure $

```

```

$                                     to make the nodes.                                     $
$-----$
$
$(29 UPPE HONY)UHON
$
!X51 = 0.
!X52 = 0.0285
!X53 = 0.0328
!X54 = 0.1560
!X55 = 0.1602
!X56 = 0.2905
!X57 = 0.2947
!X58 = 0.4555
!X59 = 0.4597
!X60 = 0.4883
!X61 = 0.
!X62 = 0.0285
!X63 = 0.0328
!X64 = 0.1560
!X65 = 0.1602
!X66 = 0.2905
!X67 = 0.2947
!X68 = 0.4555
!X69 = 0.4597
!X70 = 0.4883
$
!X51A= 0.      +(0.0285/"EX1")
!X52A= 0.0285+(0.0043/"EX2")
!X53A= 0.0328+(0.1232/"EX3")
!X54A= 0.1560+(0.0042/"EX4")
!X55A= 0.1602+(0.1345/"EX5")
!X56A= 0.2905+(0.0042/"EX6")
!X57A= 0.2947+(0.1608/"EX7")
!X58A= 0.4555+(0.0042/"EX8")
!X59A= 0.4597+(0.0286/"EX9")
!X61A= 0.      +(0.0285/"EX1")
!X62A= 0.0285+(0.0043/"EX2")
!X63A= 0.0328+(0.1232/"EX3")
!X64A= 0.1560+(0.0042/"EX4")
!X65A= 0.1602+(0.1345/"EX5")
!X66A= 0.2905+(0.0042/"EX6")
!X67A= 0.2947+(0.1608/"EX7")
!X68A= 0.4555+(0.0042/"EX8")
!X69A= 0.4597+(0.0286/"EX9")
$
!Y51 = 0.0949
!Y52 = 0.0921
!Y53 = 0.0917
!Y54 = 0.0797
!Y55 = 0.0793
!Y56 = 0.0666
!Y57 = 0.0662
!Y58 = 0.0506
!Y59 = 0.0502
!Y60 = 0.0474
!Y61 = 0.1203
!Y62 = 0.1175
!Y63 = 0.1171
!Y64 = 0.1051

```

```

!Y65 = 0.1047
!Y66 = 0.0920
!Y67 = 0.0916
!Y68 = 0.0760
!Y69 = 0.0756
!Y70 = 0.0728
$
!Y62A= (0.1203-(0.0973*"X62A"))
!Y63A= (0.1203-(0.0973*"X63A"))
!Y64A= (0.1203-(0.0973*"X64A"))
!Y65A= (0.1203-(0.0973*"X65A"))
!Y66A= (0.1203-(0.0973*"X66A"))
!Y67A= (0.1203-(0.0973*"X67A"))
!Y68A= (0.1203-(0.0973*"X68A"))
!Y69A= (0.1203-(0.0973*"X69A"))
$
!Y51B= 0.0949+(0.0254/"EY8")
!Y52B= 0.0921+(0.0254/"EY8")
!Y53B= 0.0917+(0.0254/"EY8")
!Y54B= 0.0797+(0.0254/"EY8")
!Y55B= 0.0793+(0.0254/"EY8")
!Y56B= 0.0666+(0.0254/"EY8")
!Y57B= 0.0662+(0.0254/"EY8")
!Y58B= 0.0506+(0.0254/"EY8")
!Y59B= 0.0502+(0.0254/"EY8")
!Y60B= 0.0474+(0.0254/"EY8")
$
!Y52C= (0.0949-(0.0973*"X52A"))+(0.0254/"EY8")
!Y53C= (0.0949-(0.0973*"X53A"))+(0.0254/"EY8")
!Y54C= (0.0949-(0.0973*"X54A"))+(0.0254/"EY8")
!Y55C= (0.0949-(0.0973*"X55A"))+(0.0254/"EY8")
!Y56C= (0.0949-(0.0973*"X56A"))+(0.0254/"EY8")
!Y57C= (0.0949-(0.0973*"X57A"))+(0.0254/"EY8")
!Y58C= (0.0949-(0.0973*"X58A"))+(0.0254/"EY8")
!Y59C= (0.0949-(0.0973*"X59A"))+(0.0254/"EY8")
$
!Z51 = 0.
!Z52 = 0.
!Z53 = 0.
!Z54 = 0.
!Z55 = 0.
!Z56 = 0.
!Z57 = 0.
!Z58 = 0.
!Z59 = 0.
!Z60 = 0.
!Z61 = 0.
!Z62 = 0.
!Z63 = 0.
!Z64 = 0.
!Z65 = 0.
!Z66 = 0.
!Z67 = 0.
!Z68 = 0.
!Z69 = 0.
!Z70 = 0.
$
*RETURN
*UHON

```

```

$
$-----$
$ Subroutine (29 UPPE FACE)UFAC: This subroutine defines the points $
$ of the upper outer facesheet to $
$ make the nodes. $
$-----$
$
$(29 UPPE FACE)UFAC
$
!X61 = 0.
!X62 = 0.0285
!X63 = 0.0328
!X64 = 0.1560
!X65 = 0.1602
!X66 = 0.2905
!X67 = 0.2947
!X68 = 0.4555
!X69 = 0.4597
!X70 = 0.4883
!X71 = 0.
!X72 = 0.0285
!X73 = 0.0328
!X74 = 0.1560
!X75 = 0.1602
!X76 = 0.2905
!X77 = 0.2947
!X78 = 0.4555
!X79 = 0.4597
!X80 = 0.4883
$
!X61A= 0.      +(0.0285/"EX1")
!X62A= 0.0285+(0.0043/"EX2")
!X63A= 0.0328+(0.1232/"EX3")
!X64A= 0.1560+(0.0042/"EX4")
!X65A= 0.1602+(0.1345/"EX5")
!X66A= 0.2905+(0.0042/"EX6")
!X67A= 0.2947+(0.1608/"EX7")
!X68A= 0.4555+(0.0042/"EX8")
!X69A= 0.4597+(0.0286/"EX9")
!X71A= 0.      +(0.0285/"EX1")
!X72A= 0.0285+(0.0043/"EX2")
!X73A= 0.0328+(0.1232/"EX3")
!X74A= 0.1560+(0.0042/"EX4")
!X75A= 0.1602+(0.1345/"EX5")
!X76A= 0.2905+(0.0042/"EX6")
!X77A= 0.2947+(0.1608/"EX7")
!X78A= 0.4555+(0.0042/"EX8")
!X79A= 0.4597+(0.0286/"EX9")
$
!Y61 = 0.1203
!Y62 = 0.1175
!Y63 = 0.1171
!Y64 = 0.1051
!Y65 = 0.1047
!Y66 = 0.0920
!Y67 = 0.0916
!Y68 = 0.0760
!Y69 = 0.0756
!Y70 = 0.0728

```



```

!Y71 = 0.1235
!Y72 = 0.1207
!Y73 = 0.1203
!Y74 = 0.1083
!Y75 = 0.1079
!Y76 = 0.0952
!Y77 = 0.0948
!Y78 = 0.0792
!Y79 = 0.0788
!Y80 = 0.0760
$
!Y72A= (0.1235-(0.0973*"X72A"))
!Y73A= (0.1235-(0.0973*"X73A"))
!Y74A= (0.1235-(0.0973*"X74A"))
!Y75A= (0.1235-(0.0973*"X75A"))
!Y76A= (0.1235-(0.0973*"X76A"))
!Y77A= (0.1235-(0.0973*"X77A"))
!Y78A= (0.1235-(0.0973*"X78A"))
!Y79A= (0.1235-(0.0973*"X79A"))
$
!Y61B= 0.1203+(0.0032/"EY9")
!Y62B= 0.1175+(0.0032/"EY9")
!Y63B= 0.1171+(0.0032/"EY9")
!Y64B= 0.1051+(0.0032/"EY9")
!Y65B= 0.1047+(0.0032/"EY9")
!Y66B= 0.0920+(0.0032/"EY9")
!Y67B= 0.0916+(0.0032/"EY9")
!Y68B= 0.0760+(0.0032/"EY9")
!Y69B= 0.0756+(0.0032/"EY9")
!Y70B= 0.0728+(0.0032/"EY9")
$
!Y62C= (0.1203-(0.0973*"X62A"))+(0.0032/"EY9")
!Y63C= (0.1203-(0.0973*"X63A"))+(0.0032/"EY9")
!Y64C= (0.1203-(0.0973*"X64A"))+(0.0032/"EY9")
!Y65C= (0.1203-(0.0973*"X65A"))+(0.0032/"EY9")
!Y66C= (0.1203-(0.0973*"X66A"))+(0.0032/"EY9")
!Y67C= (0.1203-(0.0973*"X67A"))+(0.0032/"EY9")
!Y68C= (0.1203-(0.0973*"X68A"))+(0.0032/"EY9")
!Y69C= (0.1203-(0.0973*"X69A"))+(0.0032/"EY9")
$
!Z61 = 0.
!Z62 = 0.
!Z63 = 0.
!Z64 = 0.
!Z65 = 0.
!Z66 = 0.
!Z67 = 0.
!Z68 = 0.
!Z69 = 0.
!Z70 = 0.
!Z71 = 0.
!Z72 = 0.
!Z73 = 0.
!Z74 = 0.
!Z75 = 0.
!Z76 = 0.
!Z77 = 0.
!Z78 = 0.
!Z79 = 0.

```

```

!Z80 = 0.
$
*RETURN
*UFAC
$
$-----$
$ Subroutine (29 UPPE INSU) INSU: This subroutine defines the points$
$                               of the insulation located on the $
$                               top surface of the upper outer $
$                               facesheet. $
$-----$
$
*(29 UPPE INSU)INSU
$
!X1  = 0.0000
!X2  = 0.0285
!X3  = 0.0328
!X4  = 0.1560
!X5  = 0.1602
!X6  = 0.2905
!X7  = 0.2947
!X8  = 0.4555
!X9  = 0.4597
!X10 = 0.4883
!X11 = 0.0000
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
$
!X1A = 0.0000+(0.0285/"EX1")
!X2A = 0.0285+(0.0043/"EX2")
!X3A = 0.0328+(0.1232/"EX3")
!X4A = 0.1560+(0.0042/"EX4")
!X5A = 0.1602+(0.1345/"EX5")
!X6A = 0.2905+(0.0042/"EX6")
!X7A = 0.2947+(0.1608/"EX7")
!X8A = 0.4555+(0.0042/"EX8")
!X9A = 0.4597+(0.0286/"EX9")
!X11A = 0.0000+(0.0285/"EX1")
!X12A = 0.0285+(0.0043/"EX2")
!X13A = 0.0328+(0.1232/"EX3")
!X14A = 0.1560+(0.0042/"EX4")
!X15A = 0.1602+(0.1345/"EX5")
!X16A = 0.2905+(0.0042/"EX6")
!X17A = 0.2947+(0.1608/"EX7")
!X18A = 0.4555+(0.0042/"EX8")
!X19A = 0.4597+(0.0286/"EX9")
$
!Y1  = 0.1235
!Y2  = 0.1207
!Y3  = 0.1203
!Y4  = 0.1083
!Y5  = 0.1079

```

```

!Y6   = 0.0952
!Y7   = 0.0948
!Y8   = 0.0792
!Y9   = 0.0788
!Y10  = 0.0760
!Y11  = 0.1489
!Y12  = 0.1461
!Y13  = 0.1457
!Y14  = 0.1337
!Y15  = 0.1333
!Y16  = 0.1206
!Y17  = 0.1202
!Y18  = 0.1046
!Y19  = 0.1042
!Y20  = 0.1014
$
!Y12A = (0.1489-(0.0973*"X12A"))
!Y13A = (0.1489-(0.0973*"X13A"))
!Y14A = (0.1489-(0.0973*"X14A"))
!Y15A = (0.1489-(0.0973*"X15A"))
!Y16A = (0.1489-(0.0973*"X16A"))
!Y17A = (0.1489-(0.0973*"X17A"))
!Y18A = (0.1489-(0.0973*"X18A"))
!Y19A = (0.1489-(0.0973*"X19A"))
$
!Y1B  = 0.1235+(0.0254/"EY10")
!Y2B  = 0.1207+(0.0254/"EY10")
!Y3B  = 0.1203+(0.0254/"EY10")
!Y4B  = 0.1083+(0.0254/"EY10")
!Y5B  = 0.1079+(0.0254/"EY10")
!Y6B  = 0.0952+(0.0254/"EY10")
!Y7B  = 0.0948+(0.0254/"EY10")
!Y8B  = 0.0792+(0.0254/"EY10")
!Y9B  = 0.0788+(0.0254/"EY10")
!Y10B = 0.0760+(0.0254/"EY10")
$
!Y2C  = (0.1235-(0.0973*"X2A"))+(0.0254/"EY10")
!Y3C  = (0.1235-(0.0973*"X3A"))+(0.0254/"EY10")
!Y4C  = (0.1235-(0.0973*"X4A"))+(0.0254/"EY10")
!Y5C  = (0.1235-(0.0973*"X5A"))+(0.0254/"EY10")
!Y6C  = (0.1235-(0.0973*"X6A"))+(0.0254/"EY10")
!Y7C  = (0.1235-(0.0973*"X7A"))+(0.0254/"EY10")
!Y8C  = (0.1235-(0.0973*"X8A"))+(0.0254/"EY10")
!Y9C  = (0.1235-(0.0973*"X9A"))+(0.0254/"EY10")
$
!Z1   = 0.0000
!Z2   = 0.0000
!Z3   = 0.0000
!Z4   = 0.0000
!Z5   = 0.0000
!Z6   = 0.0000
!Z7   = 0.0000
!Z8   = 0.0000
!Z9   = 0.0000
!Z10  = 0.0000
!Z11  = 0.0000
!Z12  = 0.0000
!Z13  = 0.0000
!Z14  = 0.0000

```

```

!Z15 = 0.0000
!Z16 = 0.0000
!Z17 = 0.0000
!Z18 = 0.0000
!Z19 = 0.0000
!Z20 = 0.0000
$
*RETURN
*INSU
$
$-----$
$ Subroutine (29 NODE GENE)NODE: This subroutine creates the nodes $
$                               for the entire structure.           $
$-----$
$
*(29 NODE GENE)NODE
$
*XQT TAB
$
START "TOTN"
JLOC
FORMAT = 1
$
*DCALL(29 LOWE FACE)
*DCALL(29 LFAC NODE)
*DCALL(29 LFAC FREE)
$
*DCALL(29 LOWE HONY)
*DCALL(29 LHON NODE)
*DCALL(29 LHON FREE)
$
*DCALL(29 LOWE SKIN)
*DCALL(29 LSKI NODE)
*DCALL(29 LSKI FREE)
$
*DCALL(29 LOWE TORQ)
*DCALL(29 LOTO NODE)
*DCALL(29 LTOR FREE)
$
*DCALL(29 AIRG SECT)
*DCALL(29 AIRG NODE)
*DCALL(29 AIRG FREE)
$
*DCALL(29 UPPE TORQ)
*DCALL(29 UPTO NODE)
*DCALL(29 TORU FREE)
$
*DCALL(29 UPPE SKIN)
*DCALL(29 USKI NODE)
*DCALL(29 USKI FREE)
$
*DCALL(29 UPPE HONY)
*DCALL(29 UHON NODE)
*DCALL(29 UHON FREE)
$
*DCALL(29 UPPE FACE)
*DCALL(29 UFAC NODE)
*DCALL(29 UFAC FREE)
$

```

```

*DCALL(29 UPPE INSU)
*DCALL(29 INSU NODE)
*DCALL(29 UINS FREE)
$
*RETURN
*NODE
$
$-----$
$ Subroutine (29 LFAC NODE)NOD1: This subroutine creates the nodes $
$                               for the lower outer facesheet. $
$-----$
$
*(29 LFAC NODE)NOD1
$
!P1 = 1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NX2" GT 2): *GOTO 10
$
"P1","X1","Y1","Z1","X2","Y2","Z2","NX1","XJ","NY1"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X3","Y3","Z3","X13","Y13","Z13","NY1","YJ"
$
"P3","X3A","Y3","Z3","X4","Y4","Z4","EX3","XJ","NY1"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X5","Y5","Z5","X15","Y15","Z15","NY1","YJ"
$
"P5","X5A","Y5","Z5","X6","Y6","Z6","EX5","XJ","NY1"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X7","Y7","Z7","X17","Y17","Z17","NY1","YJ"
$
"P7","X7A","Y7","Z7","X8","Y8","Z8","EX7","XJ","NY1"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X9","Y9","Z9","X19","Y19","Z19","NY1","YJ"
$
"P9","X9A","Y9","Z9","X10","Y10","Z10","EX9","XJ","NY1"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*GOTO 20
$
*LABEL 10
$
"P1","X1","Y1","Z1","X2","Y2","Z2","NX1","XJ","NY1"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X2A","Y2","Z2","X3","Y3","Z3","EX2","XJ","NY1"
"YJ","X12A","Y12","Z12","X13","Y13","Z13"
$

```

```

"P3","X3A","Y3","Z3","X4","Y4","Z4","EX3","XJ","NY1"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X4A","Y4","Z4","X5","Y5","Z5","EX4","XJ","NY1"
"YJ","X14A","Y14","Z14","X15","Y15","Z15"
$
"P5","X5A","Y5","Z5","X6","Y6","Z6","EX5","XJ","NY1"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X6A","Y6","Z6","X7","Y7","Z7","EX6","XJ","NY1"
"YJ","X16A","Y16","Z16","X17","Y17","Z17"
$
"P7","X7A","Y7","Z7","X8","Y8","Z8","EX7","XJ","NY1"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X8A","Y8","Z8","X9","Y9","Z9","EX8","XJ","NY1"
"YJ","X18A","Y18","Z18","X19","Y19","Z19"
$
"P9","X9A","Y9","Z9","X10","Y10","Z10","EX9","XJ","NY1"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*LABEL 20
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD1
$
$-----$
$ Subroutine (29 LHON NODE)NOD2: This subroutine creates the nodes $
$                               for the lower honeycomb structure. $
$-----$
$
*(29 LHON NODE)NOD2
$
!P1 = ("YJ"*"NY1")+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY2" GT 2): *GOTO 50
*IF("NX2" GT 2): *GOTO 30
$
"P1","X21","Y21","Z21","X22","Y22","Z22","NX1","XJ"
$
"P2","X23","Y23","Z23"

```

```

$
"P3","X23A","Y23","Z23","X24","Y24","Z24","EX3","XJ"
$
"P4","X25","Y25","Z25"
$
"P5","X25A","Y25","Z25","X26","Y26","Z26","EX5","XJ"
$
"P6","X27","Y27","Z27"
$
"P7","X27A","Y27","Z27","X28","Y28","Z28","EX7","XJ"
$
"P8","X29","Y29","Z29"
$
"P9","X29A","Y29","Z29","X30","Y30","Z30","EX9","XJ"
$
*GOTO 40
$
*LABEL 30
$
"P1","X21","Y21","Z21","X22","Y22","Z22","NX1","XJ"
$
"P2","X13","Y13B","Z13","X23","Y23","Z13","EX2","XJ"
$
"P3","X23A","Y23","Z23","X24","Y24","Z24","EX3","XJ"
$
"P4","X15","Y15B","Z15","X25","Y25","Z25","EX4","XJ"
$
"P5","X25A","Y25","Z25","X26","Y26","Z26","EX5","XJ"
$
"P6","X17","Y17B","Z17","X27","Y27","Z27","EX6","XJ"
$
"P7","X27A","Y27","Z27","X28","Y28","Z28","EX7","XJ"
$
"P8","X19","Y19B","Z19","X29","Y29","Z29","EX8","XJ"
$
"P9","X29A","Y29","Z29","X30","Y30","Z30","EX9","XJ"
$
*GOTO 40
$
*LABEL 50
$
*IF("NX2" GT 2): *GOTO 60
$
"P1","X11","Y11B","Z11","X12","Y12B","Z12","NX1","XJ","EY2"
"YJ","X21","Y21","Z21","X22","Y22","Z22"
$
"P2","X13","Y13B","Z13","X23","Y23","Z13","EY2","YJ"
$
"P3","X13A","Y13B","Z13","X14","Y14B","Z14","EX3","XJ","EY2"
"YJ","X23A","Y23","Z23","X24","Y24","Z24"
$
"P4","X15","Y15B","Z15","X25","Y25","Z25","EY2","YJ"
$
"P5","X15A","Y15B","Z15","X16","Y16B","Z16","EX5","XJ","EY2"
"YJ","X25A","Y25","Z25","X26","Y26","Z26"
$
"P6","X17","Y17B","Z17","X27","Y27","Z27","EY2","YJ"
$
"P7","X17A","Y17B","Z17","X18","Y18B","Z18","EX7","XJ","EY2"

```

```

"YJ","X27A","Y27","Z27","X28","Y28","Z28"
$
"P8","X19","Y19B","Z19","X29","Y29","Z29","EY2","YJ"
$
"P9","X19A","Y19B","Z19","X20","Y20B","Z20","EX9","XJ","EY2"
"YJ","X29A","Y29","Z29","X30","Y30","Z30"
$
*GOTO 40
$
*LABEL 60
$
"P1","X11","Y11B","Z11","X12","Y12B","Z12","NX1","XJ","EY2"
"YJ","X21","Y21","Z21","X22","Y22","Z22"
$
"P2","X12A","Y12B","Z12","X13","Y13B","Z13","EX2","XJ","EY2"
"YJ","X22A","Y22","Z22","X23","Y23","Z23"
$
"P3","X13A","Y13B","Z13","X14","Y14B","Z14","EX3","XJ","EY2"
"YJ","X23A","Y23","Z23","X24","Y24","Z24"
$
"P4","X14A","Y14B","Z14","X15","Y15B","Z15","EX4","XJ","EY2"
"YJ","X24A","Y24","Z24","X25","Y25","Z25"
$
"P5","X15A","Y15B","Z15","X16","Y16B","Z16","EX5","XJ","EY2"
"YJ","X25A","Y25","Z25","X26","Y26","Z26"
$
"P6","X16A","Y16B","Z16","X17","Y17B","Z17","EX6","XJ","EY2"
"YJ","X26A","Y26","Z26","X27","Y27","Z27"
$
"P7","X17A","Y17B","Z17","X18","Y18B","Z18","EX7","XJ","EY2"
"YJ","X27A","Y27","Z27","X28","Y28","Z28"
$
"P8","X18A","Y18B","Z18","X19","Y19B","Z19","EX8","XJ","EY2"
"YJ","X28A","Y28","Z28","X29","Y29","Z29"
$
"P9","X19A","Y19B","Z19","X20","Y20B","Z20","EX9","XJ","EY2"
"YJ","X29A","Y29","Z29","X30","Y30","Z30"
$
*LABEL 40
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD2
$
$-----$
$ Subroutine (29 LSKI NODE)NOD3: This subroutine creates the nodes $
$                               for the lower skin. $
$-----$
$
*(29 LSKI NODE)NOD3

```



```

$
!P1 = ("YJ"*("NY1"+"EY2"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY3" GT 2): *GOTO 90
*IF("NX2" GT 2): *GOTO 70
$
"P1", "X31", "Y31", "Z31", "X32", "Y32", "Z32", "NX1", "XJ"
$
"P2", "X33", "Y33", "Z33"
$
"P3", "X33A", "Y33", "Z33", "X34", "Y34", "Z34", "EX3", "XJ"
$
"P4", "X35", "Y35", "Z35"
$
"P5", "X35A", "Y35", "Z35", "X36", "Y36", "Z36", "EX5", "XJ"
$
"P6", "X37", "Y37", "Z37"
$
"P7", "X37A", "Y37", "Z37", "X38", "Y38", "Z38", "EX7", "XJ"
$
"P8", "X39", "Y39", "Z39"
$
"P9", "X39A", "Y39", "Z39", "X40", "Y40", "Z40", "EX9", "XJ"
$
*GOTO 80
$
*LABEL 70
$
"P1", "X31", "Y31", "Z31", "X32", "Y32", "Z32", "NX1", "XJ"
$
"P2", "X23", "Y23B", "Z23", "X33", "Y33", "Z33", "EX2", "XJ"
$
"P3", "X33A", "Y33", "Z33", "X34", "Y34", "Z34", "EX3", "XJ"
$
"P4", "X25", "Y25B", "Z25", "X35", "Y35", "Z35", "EX4", "XJ"
$
"P5", "X35A", "Y35", "Z35", "X36", "Y36", "Z36", "EX5", "XJ"
$
"P6", "X27", "Y27B", "Z27", "X37", "Y37", "Z37", "EX6", "XJ"
$
"P7", "X37A", "Y37", "Z37", "X38", "Y38", "Z38", "EX7", "XJ"
$
"P8", "X29", "Y29B", "Z29", "X39", "Y39", "Z39", "EX8", "XJ"
$
"P9", "X39A", "Y39", "Z39", "X40", "Y40", "Z40", "EX9", "XJ"
$
*GOTO 80
$
*LABEL 90
$
*IF("NX2" GT 2): *GOTO 100

```

```

$
"P1","X21","Y21B","Z21","X22","Y22B","Z22","NX1","XJ","EY3"
"YJ","X31","Y31","Z31","X32","Y32","Z32"
$
"P2","X23","Y23B","Z23","X33","Y33","Z33","EY3","YJ"
$
"P3","X23A","Y23B","Z23","X24","Y24B","Z24","EX3","XJ","EY3"
"YJ","X33A","Y33","Z33","X34","Y34","Z34"
$
"P4","X25","Y25B","Z25","X35","Y35","Z35","EY3","YJ"
$
"P5","X25A","Y25B","Z25","X26","Y26B","Z26","EX5","XJ","EY3"
"YJ","X35A","Y35","Z35","X36","Y36","Z36"
$
"P6","X27","Y27B","Z27","X37","Y37","Z37","EY3","YJ"
$
"P7","X27A","Y27B","Z27","X28","Y28B","Z28","EX7","XJ","EY3"
"YJ","X37A","Y37","Z37","X38","Y38","Z38"
$
"P8","X29","Y29B","Z29","X39","Y39","Z39","EY3","YJ"
$
"P9","X29A","Y29B","Z29","X30","Y30B","Z30","EX9","XJ","EY3"
"YJ","X39A","Y39","Z39","X40","Y40","Z40"
$
*GOTO 80
$
*LABEL 100
$
"P1","X21","Y21B","Z21","X22","Y22B","Z22","NX1","XJ","EY3"
"YJ","X31","Y31","Z31","X32","Y32","Z32"
$
"P2","X22A","Y22B","Z22","X23","Y23B","Z23","EX2","XJ","EY3"
"YJ","X32A","Y32","Z32","X33","Y33","Z33"
$
"P3","X23A","Y23B","Z23","X24","Y24B","Z24","EX3","XJ","EY3"
"YJ","X33A","Y33","Z33","X34","Y34","Z34"
$
"P4","X24A","Y24B","Z24","X25","Y25B","Z25","EX4","XJ","EY3"
"YJ","X34A","Y34","Z34","X35","Y35","Z35"
$
"P5","X25A","Y25B","Z25","X26","Y26B","Z26","EX5","XJ","EY3"
"YJ","X35A","Y35","Z35","X36","Y36","Z36"
$
"P6","X26A","Y26B","Z26","X27","Y27B","Z27","EX6","XJ","EY3"
"YJ","X36A","Y36","Z36","X37","Y37","Z37"
$
"P7","X27A","Y27B","Z27","X28","Y28B","Z28","EX7","XJ","EY3"
"YJ","X37A","Y37","Z37","X38","Y38","Z38"
$
"P8","X28A","Y28B","Z28","X29","Y29B","Z29","EX8","XJ","EY3"
"YJ","X38A","Y38","Z38","X39","Y39","Z39"
$
"P9","X29A","Y29B","Z29","X30","Y30B","Z30","EX9","XJ","EY3"
"YJ","X39A","Y39","Z39","X40","Y40","Z40"
$
*LABEL 80
$
!P1 = FREE ()
!P2 = FREE ()

```

```

!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD3
$
$-----$
$ Subroutine (29 LOTO NODE)LOTO: This subroutine creates the nodes $
$                               for the lower skin. $
$-----$
$
*(29 LOTO NODE)LOTO
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY4" GT 2): *GOTO 90
*IF("NX2" GT 2): *GOTO 70
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X13","Y13","Z13"
$
"P3","X13A","Y13","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X15","Y15","Z15"
$
"P5","X15A","Y15","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X17","Y17","Z17"
$
"P7","X17A","Y17","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X19","Y19","Z19"
$
"P9","X19A","Y19","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 80
$
*LABEL 70
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EX2","XJ"
$
"P3","X13A","Y13","Z13","X14","Y14","Z14","EX3","XJ"
$

```

```

"P4","X5","Y5B","Z5","X15","Y15","Z15","EX4","XJ"
$
"P5","X15A","Y15","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EX6","XJ"
$
"P7","X17A","Y17","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EX8","XJ"
$
"P9","X19A","Y19","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 80
$
*LABEL 90
$
*IF("NX2" GT 2): *GOTO 100
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY4"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EY4","YJ"
$
"P3","X3A","Y3B","Z3","X4","Y4B","Z4","EX3","XJ","EY4"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EY4","YJ"
$
"P5","X5A","Y5B","Z5","X6","Y6B","Z6","EX5","XJ","EY4"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EY4","YJ"
$
"P7","X7A","Y7B","Z7","X8","Y8B","Z8","EX7","XJ","EY4"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EY4","YJ"
$
"P9","X9A","Y9B","Z9","X10","Y10B","Z10","EX9","XJ","EY4"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*GOTO 80
$
*LABEL 100
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY4"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X2A","Y2B","Z2","X3","Y3B","Z3","EX2","XJ","EY4"
"YJ","X12A","Y12","Z12","X13","Y13","Z13"
$
"P3","X3A","Y3B","Z3","X4","Y4B","Z4","EX3","XJ","EY4"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X4A","Y4B","Z4","X5","Y5B","Z5","EX4","XJ","EY4"
"YJ","X14A","Y14","Z14","X15","Y15","Z15"
$
"P5","X5A","Y5B","Z5","X6","Y6B","Z6","EX5","XJ","EY4"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"

```

```

$
"P6","X6A","Y6B","Z6","X7","Y7B","Z7","EX6","XJ","EY4"
"YJ","X16A","Y16","Z16","X17","Y17","Z17"
$
"P7","X7A","Y7B","Z7","X8","Y8B","Z8","EX7","XJ","EY4"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X8A","Y8B","Z8","X9","Y9B","Z9","EX8","XJ","EY4"
"YJ","X18A","Y18","Z18","X19","Y19","Z19"
$
"P9","X9A","Y9B","Z9","X10","Y10B","Z10","EX9","XJ","EY4"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*LABEL 80
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*LOTO
$
$-----$
$ Subroutine (29 AIRG NODE)NOD4: This subroutine creates the nodes $
$                               for the mid section of the $
$                               structure. $
$-----$
$
*(29 AIRG NODE)NOD4
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY5" GT 2): *GOTO 130
*IF("NX2" GT 2): *GOTO 110
$
"P1","X41","Y41","Z41","X42","Y42","Z42","NX1","XJ"
$
"P2","X43","Y43","Z43"
$
"P3","X43A","Y43A","Z43","X44","Y44","Z44","EX3","XJ"
$
"P4","X45","Y45","Z45"
$
"P5","X45A","Y45A","Z45","X46","Y46","Z46","EX5","XJ"
$

```

```

"P6","X47","Y47","Z47"
$
"P7","X47A","Y47A","Z47","X48","Y48","Z48","EX7","XJ"
$
"P8","X49","Y49","Z49"
$
"P9","X49A","Y49A","Z49","X50","Y50","Z50","EX9","XJ"
$
$
*GOTO 120
$
*LABEL 110
$
"P1","X41","Y41","Z41","X42","Y42","Z42","NX1","XJ"
$
"P2","X33","Y33B","Z33","X43","Y43","Z43","EX2","XJ"
$
"P3","X43A","Y43A","Z43","X44","Y44","Z44","EX3","XJ"
$
"P4","X35","Y35B","Z35","X45","Y45","Z45","EX4","XJ"
$
"P5","X45A","Y45A","Z45","X46","Y46","Z46","EX5","XJ"
$
"P6","X37","Y37B","Z37","X47","Y47","Z47","EX6","XJ"
$
"P7","X47A","Y47A","Z47","X48","Y48","Z48","EX7","XJ"
$
"P8","X39","Y39B","Z39","X49","Y49","Z49","EX8","XJ"
$
"P9","X49A","Y49A","Z49","X50","Y50","Z50","EX9","XJ"
$
$
*GOTO 120
$
*LABEL 130
$
*IF("NX2" GT 2): *GOTO 140
$
"P1","X31","Y31B","Z31","X32","Y32B","Z32","NX1","XJ","EY5"
"YJ","X41","Y41","Z41","X42","Y42","Z42"
$
"P2","X33","Y33B","Z33","X43","Y43","Z43","EY5","YJ"
$
"P3","X33A","Y33C","Z33","X34","Y34B","Z34","EX3","XJ","EY5"
"YJ","X43A","Y43A","Z43","X44","Y44","Z44"
$
"P4","X35","Y35B","Z35","X45","Y45","Z45","EY5","YJ"
$
"P5","X35A","Y35C","Z35","X36","Y36B","Z36","EX5","XJ","EY5"
"YJ","X45A","Y45A","Z45","X46","Y46","Z46"
$
"P6","X37","Y37B","Z37","X47","Y47","Z47","EY5","YJ"
$
"P7","X37A","Y37C","Z37","X38","Y38B","Z38","EX7","XJ","EY5"
"YJ","X47A","Y47A","Z47","X48","Y48","Z48"
$
"P8","X39","Y39B","Z39","X49","Y49","Z49","EY5","YJ"
$
"P9","X39A","Y39C","Z39","X40","Y40B","Z40","EX9","XJ","EY5"
"YJ","X49A","Y49A","Z49","X50","Y50","Z50"
$

```

```

*GOTO 120
$
*LABEL 140
$
"P1","X31","Y31B","Z31","X32","Y32B","Z32","NX1","XJ","EY5"
"YJ","X41","Y41","Z41","X42","Y42","Z42"
$
"P2","X32A","Y32C","Z32","X33","Y33B","Z33","EX2","XJ","EY5"
"YJ","X42A","Y42","Z42","X43","Y43","Z43"
$
"P3","X33A","Y33C","Z33","X34","Y34B","Z34","EX3","XJ","EY5"
"YJ","X43A","Y43","Z43","X44","Y44","Z44"
$
"P4","X34A","Y34C","Z34","X35","Y35B","Z35","EX4","XJ","EY5"
"YJ","X44A","Y44","Z44","X45","Y45","Z45"
$
"P5","X35A","Y35C","Z35","X36","Y36B","Z36","EX5","XJ","EY5"
"YJ","X45A","Y45","Z45","X46","Y46","Z46"
$
"P6","X36A","Y36C","Z36","X37","Y37B","Z37","EX6","XJ","EY5"
"YJ","X46A","Y46","Z46","X47","Y47","Z47"
$
"P7","X37A","Y37C","Z37","X38","Y38B","Z38","EX7","XJ","EY5"
"YJ","X47A","Y47","Z47","X48","Y48","Z48"
$
"P8","X38A","Y38C","Z38","X39","Y39B","Z39","EX8","XJ","EY5"
"YJ","X48A","Y48","Z48","X49","Y49","Z49"
$
"P9","X39A","Y39C","Z39","X40","Y40B","Z40","EX9","XJ","EY5"
"YJ","X49A","Y49","Z49","X50","Y50","Z50"
$
*LABEL 120
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD4
$
$-----$
$ Subroutine (29 UPTO NODE)UPTO: This subroutine creates the nodes $
$                               for the upper skin. $
$-----$
$
*(29 UPTO NODE)UPTO
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"

```

```

!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY6" GT 2): *GOTO 170
*IF("NX2" GT 2): *GOTO 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X53", "Y53", "Z53"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X55", "Y55", "Z55"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X57", "Y57", "Z57"
$
"P7", "X57A", "Y57A", "Z57", "X58", "Y58", "Z58", "EX7", "XJ"
$
"P8", "X59", "Y59", "Z59"
$
"P9", "X59A", "Y59A", "Z59", "X60", "Y60", "Z60", "EX9", "XJ"
$
*GOTO 160
$
*LABEL 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X43", "Y43B", "Z43", "X53", "Y53", "Z53", "EX2", "XJ"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X45", "Y45B", "Z45", "X55", "Y55", "Z55", "EX4", "XJ"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X47", "Y47B", "Z47", "X57", "Y57", "Z57", "EX6", "XJ"
$
"P7", "X57A", "Y57A", "Z57", "X58", "Y58", "Z58", "EX7", "XJ"
$
"P8", "X49", "Y49B", "Z49", "X59", "Y59", "Z59", "EX8", "XJ"
$
"P9", "X59A", "Y59A", "Z59", "X60", "Y60", "Z60", "EX9", "XJ"
$
*GOTO 160
$
$
*LABEL 170
$
*IF("NX2" GT 2): *GOTO 180
$
"P1", "X41", "Y41B", "Z41", "X42", "Y42B", "Z42", "NX1", "XJ", "EY6"
"YJ", "X51", "Y51", "Z51", "X52", "Y52", "Z52"
$
"P2", "X43", "Y43B", "Z43", "X53", "Y53", "Z53", "EY6", "YJ"
$

```



```

"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY6"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X45","Y45B","Z45","X55","Y55","Z55","EY6","YJ"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY6"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X47","Y47B","Z47","X57","Y57","Z57","EY6","YJ"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY6"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X49","Y49B","Z49","X59","Y59","Z59","EY6","YJ"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY6"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*GOTO 160
$
*LABEL 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY6"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X42A","Y42C","Z42","X43","Y43B","Z43","EX2","XJ","EY6"
"YJ","X52A","Y52A","Z52","X53","Y53","Z53"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY6"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X44A","Y44C","Z44","X45","Y45B","Z45","EX4","XJ","EY6"
"YJ","X54A","Y54A","Z54","X55","Y55","Z55"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY6"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X46A","Y46C","Z46","X47","Y47B","Z47","EX6","XJ","EY6"
"YJ","X56A","Y56A","Z56","X57","Y57","Z57"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY6"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X48A","Y48C","Z48","X49","Y49B","Z49","EX8","XJ","EY6"
"YJ","X58A","Y58A","Z58","X59","Y59","Z59"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY6"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*LABEL 160
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()

```

```

!P9 = FREE ()
$
*RETURN
*UPTO
$
$-----$
$ Subroutine (29 USKI NODE)NOD5: This subroutine creates the nodes $
$                               for the upper skin. $
$-----$
$
*(29 USKI NODE)NOD5
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY7" GT 2): *GOTO 170
*IF("NX2" GT 2): *GOTO 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X53", "Y53", "Z53"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X55", "Y55", "Z55"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X57", "Y57", "Z57"
$
"P7", "X57A", "Y57A", "Z57", "X58", "Y58", "Z58", "EX7", "XJ"
$
"P8", "X59", "Y59", "Z59"
$
"P9", "X59A", "Y59A", "Z59", "X60", "Y60", "Z60", "EX9", "XJ"
$
*GOTO 160
$
*LABEL 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X43", "Y43B", "Z43", "X53", "Y53", "Z53", "EX2", "XJ"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X45", "Y45B", "Z45", "X55", "Y55", "Z55", "EX4", "XJ"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X47", "Y47B", "Z47", "X57", "Y57", "Z57", "EX6", "XJ"
$

```

```

"P7","X57A","Y57A","Z57","X58","Y58","Z58","EX7","XJ"
$
"P8","X49","Y49B","Z49","X59","Y59","Z59","EX8","XJ"
$
"P9","X59A","Y59A","Z59","X60","Y60","Z60","EX9","XJ"
$
*GOTO 160
$
$
*LABEL 170
$
*IF("NX2" GT 2): *GOTO 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY7"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X43","Y43B","Z43","X53","Y53","Z53","EY7","YJ"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY7"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X45","Y45B","Z45","X55","Y55","Z55","EY7","YJ"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY7"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X47","Y47B","Z47","X57","Y57","Z57","EY7","YJ"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY7"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X49","Y49B","Z49","X59","Y59","Z59","EY7","YJ"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY7"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*GOTO 160
$
*LABEL 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY7"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X42A","Y42C","Z42","X43","Y43B","Z43","EX2","XJ","EY7"
"YJ","X52A","Y52A","Z52","X53","Y53","Z53"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY7"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X44A","Y44C","Z44","X45","Y45B","Z45","EX4","XJ","EY7"
"YJ","X54A","Y54A","Z54","X55","Y55","Z55"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY7"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X46A","Y46C","Z46","X47","Y47B","Z47","EX6","XJ","EY7"
"YJ","X56A","Y56A","Z56","X57","Y57","Z57"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY7"

```

```

"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X48A","Y48C","Z48","X49","Y49B","Z49","EX8","XJ","EY7"
"YJ","X58A","Y58A","Z58","X59","Y59","Z59"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY7"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*LABEL 160
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD5
$
$-----$
$ Subroutine (29 UHON NODE)NOD6: This subroutine creates the nodes $
$                               for the upper honeycomb structure. $
$-----$
$
*(29 UHON NODE)NOD6
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY8" GT 2): *GOTO 210
*IF("NX2" GT 2): *GOTO 190
$
"P1","X61","Y61","Z61","X62","Y62","Z62","NX1","XJ"
$
"P2","X63","Y63","Z63"
$
"P3","X63A","Y63A","Z63","X64","Y64","Z64","EX3","XJ"
$
"P4","X65","Y65","Z65"
$
"P5","X65A","Y65A","Z65","X66","Y66","Z66","EX5","XJ"
$
"P6","X67","Y67","Z67"
$
"P7","X67A","Y67A","Z67","X68","Y68","Z68","EX7","XJ"
$
"P8","X69","Y69","Z69"
$

```

```

"P9","X69A","Y69A","Z69","X70","Y70","Z70","EX9","XJ"
$
*GOTO 200
$
*LABEL 190
$
"P1","X61","Y61","Z61","X62","Y62","Z62","NX1","XJ"
$
"P2","X53","Y53B","Z53","X63","Y63","Z63","EX2","XJ"
$
"P3","X63A","Y63A","Z63","X64","Y64","Z64","EX3","XJ"
$
"P4","X55","Y55B","Z55","X65","Y65","Z65","EX4","XJ"
$
"P5","X65A","Y65A","Z65","X66","Y66","Z66","EX5","XJ"
$
"P6","X57","Y57B","Z57","X67","Y67","Z67","EX6","XJ"
$
"P7","X67A","Y67A","Z67","X68","Y68","Z68","EX7","XJ"
$
"P8","X59","Y59B","Z59","X69","Y69","Z69","EX8","XJ"
$
"P9","X69A","Y69A","Z69","X70","Y70","Z70","EX9","XJ"
$
*GOTO 200
$
*LABEL 210
$
*IF("NX2" GT 2): *GOTO 220
$
"P1","X51","Y51B","Z51","X52","Y52B","Z52","NX1","XJ","EY8"
"YJ","X61","Y61","Z61","X62","Y62","Z62"
$
"P2","X53","Y53B","Z53","X63","Y63","Z63","EY8","YJ"
$
"P3","X53A","Y53C","Z53","X54","Y54B","Z54","EX3","XJ","EY8"
"YJ","X63A","Y63A","Z63","X64","Y64","Z64"
$
"P4","X55","Y55B","Z55","X65","Y65","Z65","EY8","YJ"
$
"P5","X55A","Y55C","Z55","X56","Y56B","Z56","EX5","XJ","EY8"
"YJ","X65A","Y65A","Z65","X66","Y66","Z66"
$
"P6","X57","Y57B","Z57","X67","Y67","Z67","EY8","YJ"
$
"P7","X57A","Y57C","Z57","X58","Y58B","Z58","EX7","XJ","EY8"
"YJ","X67A","Y67A","Z67","X68","Y68","Z68"
$
"P8","X59","Y59B","Z59","X69","Y69","Z69","EY8","YJ"
$
"P9","X59A","Y59C","Z59","X60","Y60B","Z60","EX9","XJ","EY8"
"YJ","X69A","Y69A","Z69","X70","Y70","Z70"
$
*GOTO 200
$
*LABEL 220
$
"P1","X51","Y51B","Z51","X52","Y52B","Z52","NX1","XJ","EY8"
"YJ","X61","Y61","Z61","X62","Y62","Z62"

```

```

$
"P2","X52A","Y52C","Z52","X53","Y53B","Z53","EX2","XJ","EY8"
"YJ","X62A","Y62A","Z62","X63","Y63","Z63"
$
"P3","X53A","Y53C","Z53","X54","Y54B","Z54","EX3","XJ","EY8"
"YJ","X63A","Y63A","Z63","X64","Y64","Z64"
$
"P4","X54A","Y54C","Z54","X55","Y55B","Z55","EX4","XJ","EY8"
"YJ","X64A","Y64A","Z64","X65","Y65","Z65"
$
"P5","X55A","Y55C","Z55","X56","Y56B","Z56","EX5","XJ","EY8"
"YJ","X65A","Y65A","Z65","X66","Y66","Z66"
$
"P6","X56A","Y56C","Z56","X57","Y57B","Z57","EX6","XJ","EY8"
"YJ","X66A","Y66A","Z66","X67","Y67","Z67"
$
"P7","X57A","Y57C","Z57","X58","Y58B","Z58","EX7","XJ","EY8"
"YJ","X67A","Y67A","Z67","X68","Y68","Z68"
$
"P8","X58A","Y58C","Z58","X59","Y59B","Z59","EX8","XJ","EY8"
"YJ","X68A","Y68A","Z68","X69","Y69","Z69"
$
"P9","X59A","Y59C","Z59","X60","Y60B","Z60","EX9","XJ","EY8"
"YJ","X69A","Y69A","Z69","X70","Y70","Z70"
$
*LABEL 200
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD6
$
$-----$
$ Subroutine (29 UFAC NODE)NOD7: This subroutine creates the nodes $
$ for the upper outer facesheet. $
$-----$
$
*(29 UFAC NODE)NOD7
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY9" GT 2): *GOTO 250
*IF("NX2" GT 2): *GOTO 230

```

```

$
"P1","X71","Y71","Z71","X72","Y72","Z72","NX1","XJ"
$
"P2","X73","Y73","Z73"
$
"P3","X73A","Y73A","Z73","X74","Y74","Z74","EX3","XJ"
$
"P4","X75","Y75","Z75"
$
"P5","X75A","Y75A","Z75","X76","Y76","Z76","EX5","XJ"
$
"P6","X77","Y77","Z77"
$
"P7","X77A","Y77A","Z77","X78","Y78","Z78","EX7","XJ"
$
"P8","X79","Y79","Z79"
$
"P9","X79A","Y79A","Z79","X80","Y80","Z80","EX9","XJ"
$
*GOTO 240
$
*LABEL 230
$
"P1","X71","Y71","Z71","X72","Y72","Z72","NX1","XJ"
$
"P2","X63","Y63B","Z63","X73","Y73","Z73","EX2","XJ"
$
"P3","X73A","Y73A","Z73","X74","Y74","Z74","EX3","XJ"
$
"P4","X65","Y65B","Z65","X75","Y75","Z75","EX4","XJ"
$
"P5","X75A","Y75A","Z75","X76","Y76","Z76","EX5","XJ"
$
"P6","X67","Y67B","Z67","X77","Y77","Z77","EX6","XJ"
$
"P7","X77A","Y77A","Z77","X78","Y78","Z78","EX7","XJ"
$
"P8","X69","Y69B","Z69","X79","Y79","Z79","EX8","XJ"
$
"P9","X79A","Y79A","Z79","X80","Y80","Z80","EX9","XJ"
$
*GOTO 240
$
*LABEL 250
$
*IF("NX2" GT 2): *GOTO 260
$
"P1","X61","Y61B","Z61","X62","Y62B","Z62","NX1","XJ","EY9"
"YJ","X71","Y71","Z71","X72","Y72","Z72"
$
"P2","X63","Y63B","Z63","X73","Y73","Z73","EY9","YJ"
$
"P3","X63A","Y63C","Z63","X64","Y64B","Z64","EX3","XJ","EY9"
"YJ","X73A","Y73A","Z73","X74","Y74","Z74"
$
"P4","X65","Y65B","Z65","X75","Y75","Z75","EY9","YJ"
$
"P5","X65A","Y65C","Z65","X66","Y66B","Z66","EX5","XJ","EY9"
"YJ","X75A","Y75A","Z75","X76","Y76","Z76"

```

```

$
"P6","X67","Y67B","Z67","X77","Y77","Z77","EY9","YJ"
$
"P7","X67A","Y67C","Z67","X68","Y68B","Z68","EX7","XJ","EY9"
"YJ","X77A","Y77A","Z77","X78","Y78","Z78"
$
"P8","X69","Y69B","Z69","X79","Y79","Z79","EY9","YJ"
$
"P9","X69A","Y69C","Z69","X70","Y70B","Z70","EX9","XJ","EY9"
"YJ","X79A","Y79A","Z79","X80","Y80","Z80"
$
*GOTO 240
$
*LABEL 260
$
"P1","X61","Y61B","Z61","X62","Y62B","Z62","NX1","XJ","EY9"
"YJ","X71","Y71","Z71","X72","Y72","Z72"
$
"P2","X62A","Y62C","Z62","X63","Y63B","Z63","EX2","XJ","EY9"
"YJ","X72A","Y72A","Z72","X73","Y73","Z73"
$
"P3","X63A","Y63C","Z63","X64","Y64B","Z64","EX3","XJ","EY9"
"YJ","X73A","Y73A","Z73","X74","Y74","Z74"
$
"P4","X64A","Y64C","Z64","X65","Y65B","Z65","EX4","XJ","EY9"
"YJ","X74A","Y74A","Z74","X75","Y75","Z75"
$
"P5","X65A","Y65C","Z65","X66","Y66B","Z66","EX5","XJ","EY9"
"YJ","X75A","Y75A","Z75","X76","Y76","Z76"
$
"P6","X66A","Y66C","Z66","X67","Y67B","Z67","EX6","XJ","EY9"
"YJ","X76A","Y76A","Z76","X77","Y77","Z77"
$
"P7","X67A","Y67C","Z67","X68","Y68B","Z68","EX7","XJ","EY9"
"YJ","X77A","Y77A","Z77","X78","Y78","Z78"
$
"P8","X68A","Y68C","Z68","X69","Y69B","Z69","EX8","XJ","EY9"
"YJ","X78A","Y78A","Z78","X79","Y79","Z79"
$
"P9","X69A","Y69C","Z69","X70","Y70B","Z70","EX9","XJ","EY9"
"YJ","X79A","Y79A","Z79","X80","Y80","Z80"
$
*LABEL 240
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD7
$
$-----$
$ Subroutine (29 INSU NODE)NOD8: This subroutine creates the nodes $

```



```

$                                     for the upper outer facesheet.  $
$-----$
$
$(29 INSU NODE)NOD8
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY10" GT 2): *GOTO 250
*IF("NX2" GT 2): *GOTO 230
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X13","Y13","Z13"
$
"P3","X13A","Y13A","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X15","Y15","Z15"
$
"P5","X15A","Y15A","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X17","Y17","Z17"
$
"P7","X17A","Y17A","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X19","Y19","Z19"
$
"P9","X19A","Y19A","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 240
$
*LABEL 230
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EX2","XJ"
$
"P3","X13A","Y13A","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EX4","XJ"
$
"P5","X15A","Y15A","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EX6","XJ"
$
"P7","X17A","Y17A","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EX8","XJ"
$
"P9","X19A","Y19A","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 240

```

```

$
*LABEL 250
$
*IF("NX2" GT 2): *GOTO 260
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY10"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EY10","YJ"
$
"P3","X3A","Y3C","Z3","X4","Y4B","Z4","EX3","XJ","EY10"
"YJ","X13A","Y13A","Z13","X14","Y14","Z14"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EY10","YJ"
$
"P5","X5A","Y5C","Z5","X6","Y6B","Z6","EX5","XJ","EY10"
"YJ","X15A","Y15A","Z15","X16","Y16","Z16"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EY10","YJ"
$
"P7","X7A","Y7C","Z7","X8","Y8B","Z8","EX7","XJ","EY10"
"YJ","X17A","Y17A","Z17","X18","Y18","Z18"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EY10","YJ"
$
"P9","X9A","Y9C","Z9","X10","Y10B","Z10","EX9","XJ","EY10"
"YJ","X19A","Y19A","Z19","X20","Y20","Z20"
$
*GOTO 240
$
*LABEL 260
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY10"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X2A","Y2C","Z2","X3","Y3B","Z3","EX2","XJ","EY10"
"YJ","X12A","Y12A","Z12","X13","Y13","Z13"
$
"P3","X3A","Y3C","Z3","X4","Y4B","Z4","EX3","XJ","EY10"
"YJ","X13A","Y13A","Z13","X14","Y14","Z14"
$
"P4","X4A","Y4C","Z4","X5","Y5B","Z5","EX4","XJ","EY10"
"YJ","X14A","Y14A","Z14","X15","Y15","Z15"
$
"P5","X5A","Y5C","Z5","X6","Y6B","Z6","EX5","XJ","EY10"
"YJ","X15A","Y15A","Z15","X16","Y16","Z16"
$
"P6","X6A","Y6C","Z6","X7","Y7B","Z7","EX6","XJ","EY10"
"YJ","X16A","Y16A","Z16","X17","Y17","Z17"
$
"P7","X7A","Y7C","Z7","X8","Y8B","Z8","EX7","XJ","EY10"
"YJ","X17A","Y17A","Z17","X18","Y18","Z18"
$
"P8","X8A","Y8C","Z8","X9","Y9B","Z9","EX8","XJ","EY10"
"YJ","X18A","Y18A","Z18","X19","Y19","Z19"
$
"P9","X9A","Y9C","Z9","X10","Y10B","Z10","EX9","XJ","EY10"
"YJ","X19A","Y19A","Z19","X20","Y20","Z20"
$

```

```

*LABEL 240
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD8
$
$-----$
$ Subroutine (29 ELEM GENE)ELEM: This subroutine creates the $
$                      elements for the mesh. $
$-----$
$
*(29 ELEM GENE)ELEM
$
*XQT AUS
$
TABLE(NI=1,NJ=1): K AREA : I = 1
                      J= 1: 1.0
$
TABLE(NI=1,NJ=1): K THIC : I = 1
                      J= 1: 1.0
$
*XQT ELD
RESET NUTED = 1
$
!NOPT= 1
$
!S1 = 1
!S2 = "S1"+"XJ"
!S3 = "S2"+"YJ"
!S4 = "S1"+"YJ"
$
!S5 = "S1" +("YJ"*"EY1")
!S6 = "S2" +("YJ"*"EY1")
!S7 = "S3" +("YJ"*"EY1")
!S8 = "S4" +("YJ"*"EY1")
$
!S9 = "S5" +("YJ"*"EY2")
!S10 = "S6" +("YJ"*"EY2")
!S11 = "S7" +("YJ"*"EY2")
!S12 = "S8" +("YJ"*"EY2")
$
!S13 = "S9" +("YJ"*"EY3")
!S14 = "S10"+"YJ"*"EY3")
!S15 = "S11"+"YJ"*"EY3")
!S16 = "S12"+"YJ"*"EY3")
$
!S17 = "S13"+"YJ"*"EY4")
!S18 = "S14"+"YJ"*"EY4")
!S19 = "S15"+"YJ"*"EY4")
!S20 = "S16"+"YJ"*"EY4")

```

```

$
!S21 = "S17"+"XJ"*( "EX1"+"EX2" )
!S22 = "S21"+"XJ"
!S23 = "S22"+"YJ"
!S24 = "S21"+"YJ"
$
!S25 = "S21"+"XJ"*( "EX3"+"EX4" )
!S26 = "S25"+"XJ"
!S27 = "S26"+"YJ"
!S28 = "S25"+"YJ"
$
!S29 = "S25"+"XJ"*( "EX5"+"EX6" )
!S30 = "S29"+"XJ"
!S31 = "S30"+"YJ"
!S32 = "S29"+"YJ"
$
!S33 = "S29"+"XJ"*( "EX7"+"EX8" )
!S34 = "S33"+"XJ"
!S35 = "S34"+"YJ"
!S36 = "S33"+"YJ"
$
!S37 = "S17"+"XJ"*( "EX1" )
!S38 = "S37"+"XJ"
!S39 = "S38"+"YJ"
!S40 = "S37"+"YJ"
$
!S41 = "S21"+"XJ"*( "EX3" )
!S42 = "S41"+"XJ"
!S43 = "S42"+"YJ"
!S44 = "S41"+"YJ"
$
!S45 = "S25"+"XJ"*( "EX5" )
!S46 = "S45"+"XJ"
!S47 = "S46"+"YJ"
!S48 = "S45"+"YJ"
$
!S49 = "S29"+"XJ"*( "EX7" )
!S50 = "S49"+"XJ"
!S51 = "S50"+"YJ"
!S52 = "S49"+"YJ"
$
!S53 = "S17"+"YJ"*( "EY5" )
!S54 = "S53"+"XJ"
!S55 = "S54"+"YJ"
!S56 = "S53"+"YJ"
$
!S57 = "S53"+"YJ"*( "EY6" )
!S58 = "S57"+"XJ"
!S59 = "S58"+"YJ"
!S60 = "S57"+"YJ"
$
!S61 = "S57"+"YJ"*( "EY7" )
!S62 = "S61"+"XJ"
!S63 = "S62"+"YJ"
!S64 = "S61"+"YJ"
$
!S65 = "S61"+"YJ"*( "EY8" )
!S66 = "S65"+"XJ"
!S67 = "S66"+"YJ"

```

```

!S68 = "S65"+"YJ"
$
!S69 = "S65"+"YJ"*"EY9"
!S70 = "S69"+"XJ"
!S71 = "S70"+"YJ"
!S72 = "S69"+"YJ"
$
!R69H= "S69"+"EX1"+"EX2"+"EX3"+"EX4"+"EX5"/2.0)
!R70H= "R69H"+"XJ"
!YJRH = ("EX5"/2.0)+"EX6"+"EX7"+"EX8"+"EX9"
!S69H= IFIX("R69H")
!S70H= IFIX("R70H")
!YJEH= IFIX("YJRH")
$
K21
$
$-----Element generator for the heater-----$
$
GROUP= 1
NMAT = 2
$
"S69H","S70H","NOPT","YJEH"
$
K41
$
$-----Element generator for the lower outer facesheet-----$
$
GROUP= 1
NMAT = 7
$
"S1","S2","S3","S4","NOPT","YJE","EY1"
$
$-----Element generator for the lower honeycomb-----$
$
GROUP= 2
NMAT = 5
$
"S5","S6","S7","S8","NOPT","YJE","EY2"
$
$-----Element generator for the lower skin-----$
$
GROUP= 3
NMAT = 1
$
"S9","S10","S11","S12","NOPT","YJE","EY3"
$
$-----Element generator for the lower torque tube-----$
$
GROUP= 4
NMAT = 6
$
"S13","S14","S15","S16","NOPT","YJE","EY4"
$
$-----Element generator for the airgaps-----$
$
GROUP= 5
NMAT = 4
$
"S17","S18","S19","S20","NOPT","EX1","EY5"

```

```

"S21","S22","S23","S24","NOPT","EX3","EY5"
"S25","S26","S27","S28","NOPT","EX5","EY5"
"S29","S30","S31","S32","NOPT","EX7","EY5"
"S33","S34","S35","S36","NOPT","EX9","EY5"
$
$-----Element generator for the torque tube ribs-----$
$
GROUP= 6
NMAT = 8
$
"S37","S38","S39","S40","NOPT","EX2","EY5"
"S41","S42","S43","S44","NOPT","EX4","EY5"
"S45","S46","S47","S48","NOPT","EX6","EY5"
"S49","S50","S51","S52","NOPT","EX8","EY5"
$
$-----Element generator for the upper torque tube-----$
$
GROUP= 7
NMAT = 6
$
"S53","S54","S55","S56","NOPT","YJE","EY6"
$
$-----Element generator for the upper skin-----$
$
GROUP= 8
NMAT = 1
$
"S57","S58","S59","S60","NOPT","YJE","EY7"
$
$-----Element generator for the upper honeycomb-----$
$
GROUP= 9
NMAT = 5
$
"S61","S62","S63","S64","NOPT","YJE","EY8"
$
$-----Element generator for the upper outer facesheet-----$
$
GROUP= 10
NMAT = 7
$
"S65","S66","S67","S68","NOPT","YJE","EY9"
$
$-----Element generator for the insulation-----$
$
GROUP= 11
NMAT = 3
$
"S69","S70","S71","S72","NOPT","YJE","EY10"
$
*XQT TGEO
RESET ASPECT = 50.0
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 ELEM FREE)FRE8: This subroutine deletes all the $
$ starting node numbers used during $

```

```

$                                     the element generation process.  $
$-----$
$
$(29 ELEM FREE)FRE8
$
!S1  = FREE ()
!S2  = FREE ()
!S3  = FREE ()
!S4  = FREE ()
$
!S5  = FREE ()
!S6  = FREE ()
!S7  = FREE ()
!S8  = FREE ()
$
!S9  = FREE ()
!S10 = FREE ()
!S11 = FREE ()
!S12 = FREE ()
$
!S13 = FREE ()
!S14 = FREE ()
!S15 = FREE ()
!S16 = FREE ()
$
!S17 = FREE ()
!S18 = FREE ()
!S19 = FREE ()
!S20 = FREE ()
$
!S21 = FREE ()
!S22 = FREE ()
!S23 = FREE ()
!S24 = FREE ()
$
!S25 = FREE ()
!S26 = FREE ()
!S27 = FREE ()
!S28 = FREE ()
$
!S29 = FREE ()
!S30 = FREE ()
!S31 = FREE ()
!S32 = FREE ()
$
!S33 = FREE ()
!S34 = FREE ()
!S35 = FREE ()
!S36 = FREE ()
$
!S37 = FREE ()
!S38 = FREE ()
!S39 = FREE ()
!S40 = FREE ()
$
!S41 = FREE ()
!S42 = FREE ()
!S43 = FREE ()
!S44 = FREE ()

```

```

$
!S45 = FREE ()
!S46 = FREE ()
!S47 = FREE ()
!S48 = FREE ()
$
!S49 = FREE ()
!S50 = FREE ()
!S51 = FREE ()
!S52 = FREE ()
$
!S53 = FREE ()
!S54 = FREE ()
!S55 = FREE ()
!S56 = FREE ()
$
!S57 = FREE ()
!S58 = FREE ()
!S59 = FREE ()
!S60 = FREE ()
$
!S61 = FREE ()
!S62 = FREE ()
!S63 = FREE ()
!S64 = FREE ()
!S65 = FREE ()
!S66 = FREE ()
!S67 = FREE ()
!S68 = FREE ()
!S69 = FREE ()
!S70 = FREE ()
!S71 = FREE ()
!S72 = FREE ()
!S69H= FREE ()
!S70H= FREE ()
!R69H= FREE ()
!R70H= FREE ()
$
*RETURN
*FRE8
$
$-----$
$ Subroutine (29 LFAC FREE)FRE1:  This subroutine deletes the  $
$                               coordinates for the lower outer  $
$                               facsheet.                        $
$-----$
$
*(29 LFAC FREE)FRE1
$
!X1  = FREE ()
!X2  = FREE ()
!X3  = FREE ()
!X4  = FREE ()
!X5  = FREE ()
!X6  = FREE ()
!X7  = FREE ()
!X8  = FREE ()
!X9  = FREE ()
!X10 = FREE ()

```



```

!X11 = FREE ()
!X12 = FREE ()
!X13 = FREE ()
!X14 = FREE ()
!X15 = FREE ()
!X16 = FREE ()
!X17 = FREE ()
!X18 = FREE ()
!X19 = FREE ()
!X20 = FREE ()
$
!X1A = FREE ()
!X2A = FREE ()
!X3A = FREE ()
!X4A = FREE ()
!X5A = FREE ()
!X6A = FREE ()
!X7A = FREE ()
!X8A = FREE ()
!X9A = FREE ()
!X11A= FREE ()
!X12A= FREE ()
!X13A= FREE ()
!X14A= FREE ()
!X15A= FREE ()
!X16A= FREE ()
!X17A= FREE ()
!X18A= FREE ()
!X19A= FREE ()
$
!Y1  = FREE ()
!Y2  = FREE ()
!Y3  = FREE ()
!Y4  = FREE ()
!Y5  = FREE ()
!Y6  = FREE ()
!Y7  = FREE ()
!Y8  = FREE ()
!Y9  = FREE ()
!Y10 = FREE ()
!Y11 = FREE ()
!Y12 = FREE ()
!Y13 = FREE ()
!Y14 = FREE ()
!Y15 = FREE ()
!Y16 = FREE ()
!Y17 = FREE ()
!Y18 = FREE ()
!Y19 = FREE ()
!Y20 = FREE ()
$
!Z1  = FREE ()
!Z2  = FREE ()
!Z3  = FREE ()
!Z4  = FREE ()
!Z5  = FREE ()
!Z6  = FREE ()
!Z7  = FREE ()
!Z8  = FREE ()

```

```

!Z9  = FREE ()
!Z10 = FREE ()
!Z11 = FREE ()
!Z12 = FREE ()
!Z13 = FREE ()
!Z14 = FREE ()
!Z15 = FREE ()
!Z16 = FREE ()
!Z17 = FREE ()
!Z18 = FREE ()
!Z19 = FREE ()
!Z20 = FREE ()
$
*RETURN
*FRE1
$
$-----$
$ Subroutine (29 LHON FREE)FRE2: This subroutine deletes the      $
$                               coordinates for the lower honey-   $
$                               comb structure.                     $
$-----$
$
*(29 LHON FREE)FRE2
$
!X11 = FREE ()
!X12 = FREE ()
!X13 = FREE ()
!X14 = FREE ()
!X15 = FREE ()
!X16 = FREE ()
!X17 = FREE ()
!X18 = FREE ()
!X19 = FREE ()
!X20 = FREE ()
!X21 = FREE ()
!X22 = FREE ()
!X23 = FREE ()
!X24 = FREE ()
!X25 = FREE ()
!X26 = FREE ()
!X27 = FREE ()
!X28 = FREE ()
!X29 = FREE ()
!X30 = FREE ()
$
!X11A= FREE ()
!X12A= FREE ()
!X13A= FREE ()
!X14A= FREE ()
!X15A= FREE ()
!X16A= FREE ()
!X17A= FREE ()
!X18A= FREE ()
!X19A= FREE ()
!X21A= FREE ()
!X22A= FREE ()
!X23A= FREE ()
!X24A= FREE ()
!X25A= FREE ()

```

```

!X26A= FREE ()
!X27A= FREE ()
!X28A= FREE ()
!X29A= FREE ()
$
!Y11 = FREE ()
!Y12 = FREE ()
!Y13 = FREE ()
!Y14 = FREE ()
!Y15 = FREE ()
!Y16 = FREE ()
!Y17 = FREE ()
!Y18 = FREE ()
!Y19 = FREE ()
!Y20 = FREE ()
!Y21 = FREE ()
!Y22 = FREE ()
!Y23 = FREE ()
!Y24 = FREE ()
!Y25 = FREE ()
!Y26 = FREE ()
!Y27 = FREE ()
!Y28 = FREE ()
!Y29 = FREE ()
!Y30 = FREE ()
$
!Y11B= FREE ()
!Y12B= FREE ()
!Y13B= FREE ()
!Y14B= FREE ()
!Y15B= FREE ()
!Y16B= FREE ()
!Y17B= FREE ()
!Y18B= FREE ()
!Y19B= FREE ()
!Y20B= FREE ()
$
!Z11 = FREE ()
!Z12 = FREE ()
!Z13 = FREE ()
!Z14 = FREE ()
!Z15 = FREE ()
!Z16 = FREE ()
!Z17 = FREE ()
!Z18 = FREE ()
!Z19 = FREE ()
!Z20 = FREE ()
!Z21 = FREE ()
!Z22 = FREE ()
!Z23 = FREE ()
!Z24 = FREE ()
!Z25 = FREE ()
!Z26 = FREE ()
!Z27 = FREE ()
!Z28 = FREE ()
!Z29 = FREE ()
!Z30 = FREE ()
$
*RETURN

```

```

*FRE2
$
$-----$
$ Subroutine (29 LSKI FREE)FRE3: This subroutine deletes the $
$ coordinates for the lower skin. $
$-----$
$
*(29 LSKI FREE)FRE3
$
!X21 = FREE ()
!X22 = FREE ()
!X23 = FREE ()
!X24 = FREE ()
!X25 = FREE ()
!X26 = FREE ()
!X27 = FREE ()
!X28 = FREE ()
!X29 = FREE ()
!X30 = FREE ()
!X31 = FREE ()
!X32 = FREE ()
!X33 = FREE ()
!X34 = FREE ()
!X35 = FREE ()
!X36 = FREE ()
!X37 = FREE ()
!X38 = FREE ()
!X39 = FREE ()
!X40 = FREE ()
$
!X21A= FREE ()
!X22A= FREE ()
!X23A= FREE ()
!X24A= FREE ()
!X25A= FREE ()
!X26A= FREE ()
!X27A= FREE ()
!X28A= FREE ()
!X29A= FREE ()
!X31A= FREE ()
!X32A= FREE ()
!X33A= FREE ()
!X34A= FREE ()
!X35A= FREE ()
!X36A= FREE ()
!X37A= FREE ()
!X38A= FREE ()
!X39A= FREE ()
$
!Y21 = FREE ()
!Y22 = FREE ()
!Y23 = FREE ()
!Y24 = FREE ()
!Y25 = FREE ()
!Y26 = FREE ()
!Y27 = FREE ()
!Y28 = FREE ()
!Y29 = FREE ()
!Y30 = FREE ()

```

```

!Y31 = FREE ()
!Y32 = FREE ()
!Y33 = FREE ()
!Y34 = FREE ()
!Y35 = FREE ()
!Y36 = FREE ()
!Y37 = FREE ()
!Y38 = FREE ()
!Y39 = FREE ()
!Y40 = FREE ()
$
!Y21B= FREE ()
!Y22B= FREE ()
!Y23B= FREE ()
!Y24B= FREE ()
!Y25B= FREE ()
!Y26B= FREE ()
!Y27B= FREE ()
!Y28B= FREE ()
!Y29B= FREE ()
!Y30B= FREE ()
$
!Z21 = FREE ()
!Z22 = FREE ()
!Z23 = FREE ()
!Z24 = FREE ()
!Z25 = FREE ()
!Z26 = FREE ()
!Z27 = FREE ()
!Z28 = FREE ()
!Z29 = FREE ()
!Z30 = FREE ()
!Z31 = FREE ()
!Z32 = FREE ()
!Z33 = FREE ()
!Z34 = FREE ()
!Z35 = FREE ()
!Z36 = FREE ()
!Z37 = FREE ()
!Z38 = FREE ()
!Z39 = FREE ()
!Z40 = FREE ()
$
*RETURN
*FRE3
$
$-----$
$ Subroutine (29 LTOR FREE)TORL: This subroutine deletes the $
$                               coordinates for the air section of $
$                               the model. $
$-----$
$
*(29 LTOR FREE)TORL
$
!X1  = FREE()
!X2  = FREE()
!X3  = FREE()
!X4  = FREE()
!X5  = FREE()

```

```

!X6  = FREE()
!X7  = FREE()
!X8  = FREE()
!X9  = FREE()
!X10 = FREE()
!X11 = FREE()
!X12 = FREE()
!X13 = FREE()
!X14 = FREE()
!X15 = FREE()
!X16 = FREE()
!X17 = FREE()
!X18 = FREE()
!X19 = FREE()
!X20 = FREE()
$
!X1A = FREE()
!X2A = FREE()
!X3A = FREE()
!X4A = FREE()
!X5A = FREE()
!X6A = FREE()
!X7A = FREE()
!X8A = FREE()
!X9A = FREE()
!X11A= FREE()
!X12A= FREE()
!X13A= FREE()
!X14A= FREE()
!X15A= FREE()
!X16A= FREE()
!X17A= FREE()
!X18A= FREE()
!X39A= FREE()
$
!Y1  = FREE()
!Y2  = FREE()
!Y3  = FREE()
!Y4  = FREE()
!Y5  = FREE()
!Y6  = FREE()
!Y7  = FREE()
!Y8  = FREE()
!Y9  = FREE()
!Y10 = FREE()
!Y11 = FREE()
!Y12 = FREE()
!Y13 = FREE()
!Y14 = FREE()
!Y15 = FREE()
!Y16 = FREE()
!Y17 = FREE()
!Y18 = FREE()
!Y19 = FREE()
!Y20 = FREE()
$
!Y1B = FREE()
!Y2B = FREE()
!Y3B = FREE()

```

```

!Y4B = FREE()
!Y5B = FREE()
!Y6B = FREE()
!Y7B = FREE()
!Y8B = FREE()
!Y9B = FREE()
!Y10B= FREE()
$
!Z1  = FREE()
!Z2  = FREE()
!Z3  = FREE()
!Z4  = FREE()
!Z5  = FREE()
!Z6  = FREE()
!Z7  = FREE()
!Z8  = FREE()
!Z9  = FREE()
!Z10 = FREE()
!Z11 = FREE()
!Z12 = FREE()
!Z13 = FREE()
!Z14 = FREE()
!Z15 = FREE()
!Z16 = FREE()
!Z17 = FREE()
!Z18 = FREE()
!Z19 = FREE()
!Z20 = FREE()
$
*RETURN
*TORL
$
$-----$
$ Subroutine (29 AIRG FREE)FRE4: This subroutine deletes the      $
$                               coordinates for the mid section of $
$                               the model.                          $
$-----$
$
$(29 AIRG FREE)FRE4
$
!X31 = FREE ()
!X32 = FREE ()
!X33 = FREE ()
!X34 = FREE ()
!X35 = FREE ()
!X36 = FREE ()
!X37 = FREE ()
!X38 = FREE ()
!X39 = FREE ()
!X40 = FREE ()
!X41 = FREE ()
!X42 = FREE ()
!X43 = FREE ()
!X44 = FREE ()
!X45 = FREE ()
!X46 = FREE ()
!X47 = FREE ()
!X48 = FREE ()
!X49 = FREE ()

```

```

!X50 = FREE ()
$
!X31A= FREE ()
!X32A= FREE ()
!X33A= FREE ()
!X34A= FREE ()
!X35A= FREE ()
!X36A= FREE ()
!X37A= FREE ()
!X38A= FREE ()
!X39A= FREE ()
!X41A= FREE ()
!X42A= FREE ()
!X43A= FREE ()
!X44A= FREE ()
!X45A= FREE ()
!X46A= FREE ()
!X47A= FREE ()
!X48A= FREE ()
!X49A= FREE ()
$
!Y31 = FREE ()
!Y32 = FREE ()
!Y33 = FREE ()
!Y34 = FREE ()
!Y35 = FREE ()
!Y36 = FREE ()
!Y37 = FREE ()
!Y38 = FREE ()
!Y39 = FREE ()
!Y40 = FREE ()
!Y41 = FREE ()
!Y42 = FREE ()
!Y43 = FREE ()
!Y44 = FREE ()
!Y45 = FREE ()
!Y46 = FREE ()
!Y47 = FREE ()
!Y48 = FREE ()
!Y49 = FREE ()
!Y50 = FREE ()
$
!Y42A= FREE ()
!Y43A= FREE ()
!Y44A= FREE ()
!Y45A= FREE ()
!Y46A= FREE ()
!Y47A= FREE ()
!Y48A= FREE ()
!Y49A= FREE ()
$
!Y31B= FREE ()
!Y32B= FREE ()
!Y33B= FREE ()
!Y34B= FREE ()
!Y35B= FREE ()
!Y36B= FREE ()
!Y37B= FREE ()
!Y38B= FREE ()

```



```

!Y39B= FREE ()
!Y40B= FREE ()
$
!Y32C= FREE ()
!Y33C= FREE ()
!Y34C= FREE ()
!Y35C= FREE ()
!Y36C= FREE ()
!Y37C= FREE ()
!Y38C= FREE ()
!Y39C= FREE ()
$
!Z31 = FREE ()
!Z32 = FREE ()
!Z33 = FREE ()
!Z34 = FREE ()
!Z35 = FREE ()
!Z36 = FREE ()
!Z37 = FREE ()
!Z38 = FREE ()
!Z39 = FREE ()
!Z40 = FREE ()
!Z41 = FREE ()
!Z42 = FREE ()
!Z43 = FREE ()
!Z44 = FREE ()
!Z45 = FREE ()
!Z46 = FREE ()
!Z47 = FREE ()
!Z48 = FREE ()
!Z49 = FREE ()
!Z50 = FREE ()
$
*RETURN
*FRE4
$
$-----$
$ Subroutine (29 TORU FREE)TORU: This subroutine deletes the $
$                               coordinates for the upper torque $
$                               tube. $
$-----$
$
*(29 TORU FREE)TORU
$
!X41 = FREE ()
!X42 = FREE ()
!X43 = FREE ()
!X44 = FREE ()
!X45 = FREE ()
!X46 = FREE ()
!X47 = FREE ()
!X48 = FREE ()
!X49 = FREE ()
!X50 = FREE ()
!X51 = FREE ()
!X52 = FREE ()
!X53 = FREE ()
!X54 = FREE ()
!X55 = FREE ()

```

```

!X56 = FREE ()
!X57 = FREE ()
!X58 = FREE ()
!X59 = FREE ()
!X60 = FREE ()
$
!X41A= FREE ()
!X42A= FREE ()
!X43A= FREE ()
!X44A= FREE ()
!X45A= FREE ()
!X46A= FREE ()
!X47A= FREE ()
!X48A= FREE ()
!X49A= FREE ()
!X51A= FREE ()
!X52A= FREE ()
!X53A= FREE ()
!X54A= FREE ()
!X55A= FREE ()
!X56A= FREE ()
!X57A= FREE ()
!X58A= FREE ()
!X59A= FREE ()
$
!Y41 = FREE ()
!Y42 = FREE ()
!Y43 = FREE ()
!Y44 = FREE ()
!Y45 = FREE ()
!Y46 = FREE ()
!Y47 = FREE ()
!Y48 = FREE ()
!Y49 = FREE ()
!Y50 = FREE ()
!Y51 = FREE ()
!Y52 = FREE ()
!Y53 = FREE ()
!Y54 = FREE ()
!Y55 = FREE ()
!Y56 = FREE ()
!Y57 = FREE ()
!Y58 = FREE ()
!Y59 = FREE ()
!Y60 = FREE ()
$
!Y52A= FREE ()
!Y53A= FREE ()
!Y54A= FREE ()
!Y55A= FREE ()
!Y56A= FREE ()
!Y57A= FREE ()
!Y58A= FREE ()
!Y59A= FREE ()
$
!Y41B= FREE ()
!Y42B= FREE ()
!Y43B= FREE ()
!Y44B= FREE ()

```

```

!Y45B= FREE ()
!Y46B= FREE ()
!Y47B= FREE ()
!Y48B= FREE ()
!Y49B= FREE ()
!Y50B= FREE ()
$
!Y42C= FREE ()
!Y43C= FREE ()
!Y44C= FREE ()
!Y45C= FREE ()
!Y46C= FREE ()
!Y47C= FREE ()
!Y48C= FREE ()
!Y49C= FREE ()
$
!Z41 = FREE ()
!Z42 = FREE ()
!Z43 = FREE ()
!Z44 = FREE ()
!Z45 = FREE ()
!Z46 = FREE ()
!Z47 = FREE ()
!Z48 = FREE ()
!Z49 = FREE ()
!Z50 = FREE ()
!Z51 = FREE ()
!Z52 = FREE ()
!Z53 = FREE ()
!Z54 = FREE ()
!Z55 = FREE ()
!Z56 = FREE ()
!Z57 = FREE ()
!Z58 = FREE ()
!Z59 = FREE ()
!Z60 = FREE ()
$
*RETURN
*TORU
$
$-----$
$ Subroutine (29 USKI FREE)FRE5: This subroutine deletes the $
$                               coordinates for the upper skin. $
$-----$
$
*(29 USKI FREE)FRE5
$
!X41 = FREE ()
!X42 = FREE ()
!X43 = FREE ()
!X44 = FREE ()
!X45 = FREE ()
!X46 = FREE ()
!X47 = FREE ()
!X48 = FREE ()
!X49 = FREE ()
!X50 = FREE ()
!X51 = FREE ()
!X52 = FREE ()

```

```

!X53 = FREE ()
!X54 = FREE ()
!X55 = FREE ()
!X56 = FREE ()
!X57 = FREE ()
!X58 = FREE ()
!X59 = FREE ()
!X60 = FREE ()
$
!X41A= FREE ()
!X42A= FREE ()
!X43A= FREE ()
!X44A= FREE ()
!X45A= FREE ()
!X46A= FREE ()
!X47A= FREE ()
!X48A= FREE ()
!X49A= FREE ()
!X51A= FREE ()
!X52A= FREE ()
!X53A= FREE ()
!X54A= FREE ()
!X55A= FREE ()
!X56A= FREE ()
!X57A= FREE ()
!X58A= FREE ()
!X59A= FREE ()
$
!Y41 = FREE ()
!Y42 = FREE ()
!Y43 = FREE ()
!Y44 = FREE ()
!Y45 = FREE ()
!Y46 = FREE ()
!Y47 = FREE ()
!Y48 = FREE ()
!Y49 = FREE ()
!Y50 = FREE ()
!Y51 = FREE ()
!Y52 = FREE ()
!Y53 = FREE ()
!Y54 = FREE ()
!Y55 = FREE ()
!Y56 = FREE ()
!Y57 = FREE ()
!Y58 = FREE ()
!Y59 = FREE ()
!Y60 = FREE ()
$
!Y52A= FREE ()
!Y53A= FREE ()
!Y54A= FREE ()
!Y55A= FREE ()
!Y56A= FREE ()
!Y57A= FREE ()
!Y58A= FREE ()
!Y59A= FREE ()
$
!Y41B= FREE ()

```

```

!Y42B= FREE ()
!Y43B= FREE ()
!Y44B= FREE ()
!Y45B= FREE ()
!Y46B= FREE ()
!Y47B= FREE ()
!Y48B= FREE ()
!Y49B= FREE ()
!Y50B= FREE ()
$
!Y42C= FREE ()
!Y43C= FREE ()
!Y44C= FREE ()
!Y45C= FREE ()
!Y46C= FREE ()
!Y47C= FREE ()
!Y48C= FREE ()
!Y49C= FREE ()
$
!Z41 = FREE ()
!Z42 = FREE ()
!Z43 = FREE ()
!Z44 = FREE ()
!Z45 = FREE ()
!Z46 = FREE ()
!Z47 = FREE ()
!Z48 = FREE ()
!Z49 = FREE ()
!Z50 = FREE ()
!Z51 = FREE ()
!Z52 = FREE ()
!Z53 = FREE ()
!Z54 = FREE ()
!Z55 = FREE ()
!Z56 = FREE ()
!Z57 = FREE ()
!Z58 = FREE ()
!Z59 = FREE ()
!Z60 = FREE ()
$
*RETURN
*FRE5
$
$-----$
$ Subroutine (29 UHON FREE)FRE6: This subroutine deletes the $
$                               coordinates for the upper honey- $
$                               comb structure. $
$-----$
$
$(29 UHON FREE)FRE6
$
!X51 = FREE ()
!X52 = FREE ()
!X53 = FREE ()
!X54 = FREE ()
!X55 = FREE ()
!X56 = FREE ()
!X57 = FREE ()
!X58 = FREE ()

```

```

!X59 = FREE ()
!X60 = FREE ()
!X61 = FREE ()
!X62 = FREE ()
!X63 = FREE ()
!X64 = FREE ()
!X65 = FREE ()
!X66 = FREE ()
!X67 = FREE ()
!X68 = FREE ()
!X69 = FREE ()
!X70 = FREE ()
$
!X51A= FREE ()
!X52A= FREE ()
!X53A= FREE ()
!X54A= FREE ()
!X55A= FREE ()
!X56A= FREE ()
!X57A= FREE ()
!X58A= FREE ()
!X59A= FREE ()
!X61A= FREE ()
!X62A= FREE ()
!X63A= FREE ()
!X64A= FREE ()
!X65A= FREE ()
!X66A= FREE ()
!X67A= FREE ()
!X68A= FREE ()
!X69A= FREE ()
$
!Y51 = FREE ()
!Y52 = FREE ()
!Y53 = FREE ()
!Y54 = FREE ()
!Y55 = FREE ()
!Y56 = FREE ()
!Y57 = FREE ()
!Y58 = FREE ()
!Y59 = FREE ()
!Y60 = FREE ()
!Y61 = FREE ()
!Y62 = FREE ()
!Y63 = FREE ()
!Y64 = FREE ()
!Y65 = FREE ()
!Y66 = FREE ()
!Y67 = FREE ()
!Y68 = FREE ()
!Y69 = FREE ()
!Y70 = FREE ()
$
!Y62A= FREE ()
!Y63A= FREE ()
!Y64A= FREE ()
!Y65A= FREE ()
!Y66A= FREE ()
!Y67A= FREE ()

```

```

!Y68A= FREE ()
!Y69A= FREE ()
$
!Y51B= FREE ()
!Y52B= FREE ()
!Y53B= FREE ()
!Y54B= FREE ()
!Y55B= FREE ()
!Y56B= FREE ()
!Y57B= FREE ()
!Y58B= FREE ()
!Y59B= FREE ()
!Y60B= FREE ()
$
!Y52C= FREE ()
!Y53C= FREE ()
!Y54C= FREE ()
!Y55C= FREE ()
!Y56C= FREE ()
!Y57C= FREE ()
!Y58C= FREE ()
!Y59C= FREE ()
$
!Z51 = FREE ()
!Z52 = FREE ()
!Z53 = FREE ()
!Z54 = FREE ()
!Z55 = FREE ()
!Z56 = FREE ()
!Z57 = FREE ()
!Z58 = FREE ()
!Z59 = FREE ()
!Z60 = FREE ()
!Z61 = FREE ()
!Z62 = FREE ()
!Z63 = FREE ()
!Z64 = FREE ()
!Z65 = FREE ()
!Z66 = FREE ()
!Z67 = FREE ()
!Z68 = FREE ()
!Z69 = FREE ()
!Z70 = FREE ()
$
*RETURN
*FRE6
$
$-----$
$ Subroutine (29 UFAC FREE)FRE7: This subroutine deletes the $
$                               coordinates for the upper outer $
$                               facesheet. $
$-----$
$
*(29 UFAC FREE)FRE7
$
!X61 = FREE ()
!X62 = FREE ()
!X63 = FREE ()
!X64 = FREE ()

```

```

!X65 = FREE ()
!X66 = FREE ()
!X67 = FREE ()
!X68 = FREE ()
!X69 = FREE ()
!X70 = FREE ()
!X71 = FREE ()
!X72 = FREE ()
!X73 = FREE ()
!X74 = FREE ()
!X75 = FREE ()
!X76 = FREE ()
!X77 = FREE ()
!X78 = FREE ()
!X79 = FREE ()
!X80 = FREE ()
$
!X61A= FREE ()
!X62A= FREE ()
!X63A= FREE ()
!X64A= FREE ()
!X65A= FREE ()
!X66A= FREE ()
!X67A= FREE ()
!X68A= FREE ()
!X69A= FREE ()
!X71A= FREE ()
!X72A= FREE ()
!X73A= FREE ()
!X74A= FREE ()
!X75A= FREE ()
!X76A= FREE ()
!X77A= FREE ()
!X78A= FREE ()
!X79A= FREE ()
$
!Y61 = FREE ()
!Y62 = FREE ()
!Y63 = FREE ()
!Y64 = FREE ()
!Y65 = FREE ()
!Y66 = FREE ()
!Y67 = FREE ()
!Y68 = FREE ()
!Y69 = FREE ()
!Y70 = FREE ()
!Y71 = FREE ()
!Y72 = FREE ()
!Y73 = FREE ()
!Y74 = FREE ()
!Y75 = FREE ()
!Y76 = FREE ()
!Y77 = FREE ()
!Y78 = FREE ()
!Y79 = FREE ()
!Y80 = FREE ()
$
!Y72A= FREE ()
!Y73A= FREE ()

```



```

!Y74A= FREE ()
!Y75A= FREE ()
!Y76A= FREE ()
!Y77A= FREE ()
!Y78A= FREE ()
!Y79A= FREE ()
$
!Y61B= FREE ()
!Y62B= FREE ()
!Y63B= FREE ()
!Y64B= FREE ()
!Y65B= FREE ()
!Y66B= FREE ()
!Y67B= FREE ()
!Y68B= FREE ()
!Y69B= FREE ()
!Y70B= FREE ()
$
!Y62C= FREE ()
!Y63C= FREE ()
!Y64C= FREE ()
!Y65C= FREE ()
!Y66C= FREE ()
!Y67C= FREE ()
!Y68C= FREE ()
!Y69C= FREE ()
$
!Z61 = FREE ()
!Z62 = FREE ()
!Z63 = FREE ()
!Z64 = FREE ()
!Z65 = FREE ()
!Z66 = FREE ()
!Z67 = FREE ()
!Z68 = FREE ()
!Z69 = FREE ()
!Z70 = FREE ()
!Z71 = FREE ()
!Z72 = FREE ()
!Z73 = FREE ()
!Z74 = FREE ()
!Z75 = FREE ()
!Z76 = FREE ()
!Z77 = FREE ()
!Z78 = FREE ()
!Z79 = FREE ()
!Z80 = FREE ()
$
*RETURN
*FRE7
$
$-----$
$ Subroutine (29 UINS FREE) UINS: This subroutine deletes the coor- $
$                               dinates of the insulation           $
$-----$
$
$(29 UINS FREE)UINS
$
!X1  = FREE ()

```

```

!X2  = FREE ()
!X3  = FREE ()
!X4  = FREE ()
!X5  = FREE ()
!X6  = FREE ()
!X7  = FREE ()
!X8  = FREE ()
!X9  = FREE ()
!X10 = FREE ()
!X11 = FREE ()
!X12 = FREE ()
!X13 = FREE ()
!X14 = FREE ()
!X15 = FREE ()
!X16 = FREE ()
!X17 = FREE ()
!X18 = FREE ()
!X19 = FREE ()
!X20 = FREE ()
$
!X1A = FREE ()
!X2A = FREE ()
!X3A = FREE ()
!X4A = FREE ()
!X5A = FREE ()
!X6A = FREE ()
!X7A = FREE ()
!X8A = FREE ()
!X9A = FREE ()
!X11A = FREE ()
!X12A = FREE ()
!X13A = FREE ()
!X14A = FREE ()
!X15A = FREE ()
!X16A = FREE ()
!X17A = FREE ()
!X18A = FREE ()
!X19A = FREE ()
$
!Y1  = FREE ()
!Y2  = FREE ()
!Y3  = FREE ()
!Y4  = FREE ()
!Y5  = FREE ()
!Y6  = FREE ()
!Y7  = FREE ()
!Y8  = FREE ()
!Y9  = FREE ()
!Y10 = FREE ()
!Y11 = FREE ()
!Y12 = FREE ()
!Y13 = FREE ()
!Y14 = FREE ()
!Y15 = FREE ()
!Y16 = FREE ()
!Y17 = FREE ()
!Y18 = FREE ()
!Y19 = FREE ()
!Y20 = FREE ()

```

```

$
!Y12A = FREE ()
!Y13A = FREE ()
!Y14A = FREE ()
!Y15A = FREE ()
!Y16A = FREE ()
!Y17A = FREE ()
!Y18A = FREE ()
!Y19A = FREE ()
$
!Y1B  = FREE ()
!Y2B  = FREE ()
!Y3B  = FREE ()
!Y4B  = FREE ()
!Y5B  = FREE ()
!Y6B  = FREE ()
!Y7B  = FREE ()
!Y8B  = FREE ()
!Y9B  = FREE ()
!Y10B = FREE ()
$
!Y2C  = FREE ()
!Y3C  = FREE ()
!Y4C  = FREE ()
!Y5C  = FREE ()
!Y6C  = FREE ()
!Y7C  = FREE ()
!Y8C  = FREE ()
!Y9C  = FREE ()
$
!Z1   = FREE ()
!Z2   = FREE ()
!Z3   = FREE ()
!Z4   = FREE ()
!Z5   = FREE ()
!Z6   = FREE ()
!Z7   = FREE ()
!Z8   = FREE ()
!Z9   = FREE ()
!Z10  = FREE ()
!Z11  = FREE ()
!Z12  = FREE ()
!Z13  = FREE ()
!Z14  = FREE ()
!Z15  = FREE ()
!Z16  = FREE ()
!Z17  = FREE ()
!Z18  = FREE ()
!Z19  = FREE ()
!Z20  = FREE ()
$
*RETURN
*UINS
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the  $
$                               boundary conditions such as:    $
$                               - the constant temperature     $
$                               boundary                         $

```

```

$                                - the applied heat flux boundary $
$                                - the total experimental and $
$                                heating time $
$                                - the heated node boundary $
$                                temperature $
$-----$
$
$(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DAT1
DEFINE E = 1 INPT DAT2
$
$ in the following statements, the experimental time, temperature at the
$ constant temperature boundary, the applied heat flux, and the temperatures
$ at various locations are read from the pertinent datafiles.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT68 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT67 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT66 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT65 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=6,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT64 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=7,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT63 1 1
TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=8,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB76 1 1
TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB75 1 1
TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB73 1 1
TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB74 1 1
TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH24 1 1
TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=8,OPER=XSUM)

```

```

$
TABLE(NI=1,NJ="NDS"):5 EXPE HH23 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH22 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=6,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH21 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=7,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH20 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=8,SSKIP=8,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE FLUX 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS",1 (5 EXPE TIME 1 1)
!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"

```

```

TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT FLUX
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 APPL LOOP
  TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 SOUR LOOP
  TRANSFER(SOUR=D,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=DD,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=D,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
!YJYJ = "YJ"+"YJ"
!TL   = "NDS1"*"YJEH"
!TN   = "NDS1"*"YJYJ"
!L1   = 0
!L2   = 0
!L3   = 0
!L4   = 0
!L5   = 0
!L6   = 0
$
TABLE(NI=1,NJ="TL"):5 SOUR K21 1
TABLE(NI=1,NJ="TN"):5 APPL TEMP
$
*LABEL 720
!L1   = "L2"+1
!L2   = "L1"+"YJEH"-1
!L3   = "L3"+1
!FLU  = DS 1,"L3",1 (5 SOUR LOOP)
$
TABLE,U(NI=1,NJ="TL"):5 SOUR K21 1
  J="L1","L2": "FLU"
$
*IF("L3" LT "NDS1"): *GOTO 720
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
*LABEL 730
!L4   = "L5"+1

```

```

!L5      = "L4"+"YJYJ"-1
!L6      = "L6"+1
!APL     = DS 1,"L6",1 (5 APPL LOOP)
$
TABLE,U(NI=1,NJ="TN"):5 APPL TEMP
  J="L4","L5":"APL"
$
*IF("L6" LT "NDS1"): *GOTO 730
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
  TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
  TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
  TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
!ZJ      = "YJ" + 1
!NOIN    = "TOTN"-"YJ"+1
$
TABLE(NI=1,NJ="YJYJ",TYPE=0):5 TEMP NODE
  DDATA 1
  RJ=1,"YJ":1
  DDATA 1
  RJ="ZJ","YJYJ":"NOIN"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 1):NJ="YJEH",NINJ="YJEH"
TOCC (5 APPL TEMP) :NJ="YJYJ",NINJ="YJYJ"
$
!ACNT = FREE()
!BCNT = FREE()
!CCNT = FREE()
!NOIN = FREE()
!L1   = FREE()
!L2   = FREE()
!L3   = FREE()
!L4   = FREE()
!L5   = FREE()
!L6   = FREE()
!ZJ   = FREE()
!TL   = FREE()
!TN   = FREE()
!YJYJ = FREE()
!FLU  = FREE()
!APL  = FREE()
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 GENA PROC)GENA: This subroutine is the first      $
$                               genetic processor that needs to be $

```

```

$                                     executed. Its purpose is to read $
$                                     the genetic variables defined in $
$                                     subroutine GEAL-VARI and to create $
$                                     the first parent population $
$                                     containing the first 50 $
$                                     chromosomes. More information $
$                                     about the necessary input and $
$                                     output tables as well as the RESET $
$                                     is given in the README file in the $
$                                     genetic directory. $
$-----$
$
$(29 GENA PROC)GENA
$
*XQT GENA
  RESET NTIM=1          $print reset for the input data in output file
  RESET NUIN=1          $reset for source library
  RESET NOUT=1          $reset for destination library
  RESET ITPP="ITPP"     $reset for looping process
$
*RETURN
*GENA
$
$-----$
$ Subroutine(29 GENB PROC)GENB: This is the subroutine is the second$
$                               genetic processor that needs to be $
$                               executed. It creates the elitist $
$                               parent population once the GENA $
$                               processor has been executed several $
$                               times. $
$-----$
$
$(29 GENB PROC)GENB
$
*XQT GENB
  RESET ITPP="ITPP"     $reset for looping process
  RESET NUIN=1          $reset for source library
  RESET NOUT=1          $reset for destination library
$
*RETURN
*GENB
$
$-----$
$ Subroutine(29 GENC PROC)GENC: This is the third processor that $
$                               needs to be executed. The objective $
$                               of this processor is to create a $
$                               children population using mutation, $
$                               cross-over and fitness criteria $
$                               applied to the elitist parent $
$                               population. $
$-----$
$
$(29 GENC PROC)GENC
$
*XQT GENC
  RESET ITPP="ITPP"     $reset for looping process
  RESET ITRG="ITER"     $reset for generation iteration process
  RESET NUIN= 1         $reset for source library
  RESET NOUT= 1         $reset for destination library

```



```

$
*RETURN
*GENC
$
$-----$
$ Subroutine(29 GEND PROC)GEND: This is the fourth genetic processor$
$                               that needs to be executed. The goal $
$                               of this processor is to check for   $
$                               twins and remove them, while       $
$                               creating new candidate chromosomes. $
$-----$
$
$(29 GEND PROC)GEND
$
*XQT GEND
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITER"  $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GEND
$
$-----$
$ Subroutine(29 GENE PROC)GENF: This subroutine is genetic processor$
$                               four. Its intent is to create and $
$                               sort either the iteration population$
$                               or the final population depending on$
$                               whether the final number of       $
$                               iterations has been achieved or not.$
$-----$
$
$(29 GENE PROC)GENF
$
*XQT GENE
  RESET ITPP="ITPP"  $reset for looping process
  RESET ITRG="ITER"  $reset for generation iteration process
  RESET NUIN= 1      $reset for source library
  RESET NOUT= 1      $reset for destination library
$
*RETURN
*GENF
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by   $
$                               changing the Crank Nicholson para- $
$                               meter.                             $
$-----$
$
$(29 TRAN ANAL)ANAL
$
*XQT TRTB
  RESET QLIB = 5      $ implicit transient analyzer
  RESET MLIB = 6      $ thermal excitation library
  RESET T1   = "TIMI" $ material and fluids properties library
  RESET T2   = "TIMF" $ starting time of analysis
                      $ termination time of analysis

```

```

RESET DT      = "INTS"          $ integration time step size
RESET DEST    = 4                $ output library
RESET PRINT   = 0                $ online print control parameter
RESET KPRT    = 0                $ print control parameter for k matrix
RESET MXNDT   = 10000000        $ maximum # of int. time steps during TRTB
RESET BETA    = 0.5              $ Crank Nicholson parameter
RESET KTI     = 1                $ K matrix computation time parameter
KTIME        = 10000.            $ times to calculate the conduction load vector
TEMP         = "CSTP"           $ initial temperature of structure at T1
TSAVE        = 5 EXPE TIME 1 1  $ storage of nodal temperatures at time
$                               $ increments of TSAVE
*RETURN
*ANAL
$
$-----$
$ Subroutine (29 SUMO SQUA)SQUA: This subroutine performs a      $
$                               calculation of the sum of the    $
$                               sum of squares for the different  $
$                               chromosomes and puts them into the $
$                               objective function table.        $
$-----$
$
*(29 SUMO SQUA)SQUA
$
$ the following if statements are used to define how far the runstream
$ has gotten with respect to the overall genetic process
$
*IF("SSQU" EQ 1):*GOTO 8000      $ chromosomes created by GENA
*IF("SSQU" EQ 2):*GOTO 8100      $ chromosomes created by GEND
*IF("SSQU" EQ 3):*GOTO 8200      $ chromosomes created by GENE
$
*LABEL 8000
*XQT AUS                         $ arithmetic utility system processor
$
!LINE = 0                        $ initializing counter
$
*XQT DCU                         $ database utility function processor
!NWD = TOC,NJ(1 PARE POPU 1)     $ determine number of rows in PARE POPU
*XQT AUS                         $ arithmetic utility system processor
$
$ the following two tables need to be initialized before the beginning of
$ the loop, in order to use the update command TABLE,U, the tables are
$ both filled with zeros. the first table will contain the value of the
$ objective function, whereas the second represents the identity matrix
$
TABLE(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
J=1,"NDS": 1.0
$
*LABEL 105
$
!LINE = LINE + 1                 $ increment of the counter
$
!A1 = DS,1 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A2 = DS,2 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A3 = DS,3 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A4 = DS,4 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A5 = DS,5 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A6 = DS,6 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property

```

```

!A7 = DS,7 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A8 = DS,8 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A9 = DS,9 ,"LINE",1 (1 PARE POPU 1) $ update the value of the property
!A10 = DS,10,"LINE",1 (1 PARE POPU 1) $ update the value of the property
$
*DCALL (29 MATE PROP)                $ call subroutine MATE PROP
*DCALL (29 TRAN ANAL)                $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)                $ call subroutine TEMP TABL
ERASE 6
$
INLIB = 3                            $ change source library
OUTLI = 3                            $ change destination library
$
DEFINE T68T = 2 TT68 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T67T = 2 TT67 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T66T = 2 TT66 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T65T = 2 TT65 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T64T = 2 TT64 NODE 1 1        $ define table as variable for arith. oper.
$
DEFINE T76T = 2 BB76 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T75T = 2 BB75 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T74T = 2 BB74 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T73T = 2 BB73 NODE 1 1        $ define table as variable for arith. oper.
$
DEFINE T24T = 2 HH24 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T23T = 2 HH23 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T22T = 2 HH22 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T21T = 2 HH21 NODE 1 1        $ define table as variable for arith. oper.
DEFINE T20T = 2 HH20 NODE 1 1        $ define table as variable for arith. oper.
$
DEFINE T68E = 5 EXPE TT68 1 1        $ define table as variable for arith. oper.
DEFINE T67E = 5 EXPE TT67 1 1        $ define table as variable for arith. oper.
DEFINE T66E = 5 EXPE TT66 1 1        $ define table as variable for arith. oper.
DEFINE T65E = 5 EXPE TT65 1 1        $ define table as variable for arith. oper.
DEFINE T64E = 5 EXPE TT64 1 1        $ define table as variable for arith. oper.
$
DEFINE T76E = 5 EXPE BB76 1 1        $ define table as variable for arith. oper.
DEFINE T75E = 5 EXPE BB75 1 1        $ define table as variable for arith. oper.
DEFINE T74E = 5 EXPE BB74 1 1        $ define table as variable for arith. oper.
DEFINE T73E = 5 EXPE BB73 1 1        $ define table as variable for arith. oper.
$
DEFINE T24E = 5 EXPE HH24 1 1        $ define table as variable for arith. oper.
DEFINE T23E = 5 EXPE HH23 1 1        $ define table as variable for arith. oper.
DEFINE T22E = 5 EXPE HH22 1 1        $ define table as variable for arith. oper.
DEFINE T21E = 5 EXPE HH21 1 1        $ define table as variable for arith. oper.
DEFINE T20E = 5 EXPE HH20 1 1        $ define table as variable for arith. oper.
$
DEFINE ON  = 7 ONES MATX 1 1        $ define table as variable for arith. oper.
$
SU1  = SUM(T68E,-1. T68T)            $ (Y1 -ETA1 )
SU2  = SUM(T67E,-1. T67T)            $ (Y2 -ETA2 )
SU3  = SUM(T66E,-1. T66T)            $ (Y3 -ETA3 )
SU4  = SUM(T65E,-1. T65T)            $ (Y4 -ETA4 )
SU5  = SUM(T64E,-1. T64T)            $ (Y5 -ETA5 )
SU6  = SUM(T76E,-1. T76T)            $ (Y6 -ETA6 )
SU7  = SUM(T75E,-1. T75T)            $ (Y7 -ETA7 )
SU8  = SUM(T74E,-1. T74T)            $ (Y8 -ETA8 )
SU9  = SUM(T73E,-1. T73T)            $ (Y9 -ETA9 )
SU10 = SUM(T24E,-1. T24T)            $ (Y10-ETA10)

```

```

SU11 = SUM(T23E,-1. T23T)      $ (Y11-ETA11)
SU12 = SUM(T22E,-1. T22T)      $ (Y12-ETA12)
SU13 = SUM(T21E,-1. T21T)      $ (Y13-ETA13)
SU14 = SUM(T20E,-1. T20T)      $ (Y14-ETA14)
$
SS1  = XTY(SU1,SU1)             $ (Y1-ETA1)T(Y1-ETA1)
SS2  = XTY(SU2,SU2)             $ (Y2-ETA2)T(Y2-ETA2)
SS3  = XTY(SU3,SU3)             $ (Y3-ETA3)T(Y3-ETA3)
SS4  = XTY(SU4,SU4)             $ (Y4-ETA4)T(Y4-ETA4)
SS5  = XTY(SU5,SU5)             $ (Y5-ETA5)T(Y5-ETA5)
SS6  = XTY(SU6,SU6)             $ (Y6-ETA6)T(Y6-ETA6)
SS7  = XTY(SU7,SU7)             $ (Y7-ETA7)T(Y7-ETA7)
SS8  = XTY(SU8,SU8)             $ (Y8-ETA8)T(Y8-ETA8)
SS9  = XTY(SU9,SU9)             $ (Y9-ETA9)T(Y9-ETA9)
SS10 = XTY(SU10,SU10)           $ (Y10-ETA10)T(Y10-ETA10)
SS11 = XTY(SU11,SU11)           $ (Y11-ETA11)T(Y11-ETA11)
SS12 = XTY(SU12,SU12)           $ (Y12-ETA12)T(Y12-ETA12)
SS13 = XTY(SU13,SU13)           $ (Y13-ETA13)T(Y13-ETA13)
SS14 = XTY(SU14,SU14)           $ (Y14-ETA14)T(Y14-ETA14)
$
!SV1 = DS 1,1,1 (3 SS1  AUS 1 1) $ scalar of (Y1-ETA1)T(Y1-ETA1)
!SV2 = DS 1,1,1 (3 SS2  AUS 1 1) $ scalar of (Y2-ETA2)T(Y2-ETA2)
!SV3 = DS 1,1,1 (3 SS3  AUS 1 1) $ scalar of (Y3-ETA3)T(Y3-ETA3)
!SV4 = DS 1,1,1 (3 SS4  AUS 1 1) $ scalar of (Y4-ETA4)T(Y4-ETA4)
!SV5 = DS 1,1,1 (3 SS5  AUS 1 1) $ scalar of (Y5-ETA5)T(Y5-ETA5)
!SV6 = DS 1,1,1 (3 SS6  AUS 1 1) $ scalar of (Y6-ETA6)T(Y6-ETA6)
!SV7 = DS 1,1,1 (3 SS7  AUS 1 1) $ scalar of (Y7-ETA7)T(Y7-ETA7)
!SV8 = DS 1,1,1 (3 SS8  AUS 1 1) $ scalar of (Y8-ETA8)T(Y8-ETA8)
!SV9 = DS 1,1,1 (3 SS9  AUS 1 1) $ scalar of (Y9-ETA9)T(Y9-ETA9)
!SV10 = DS 1,1,1 (3 SS10 AUS 1 1) $ scalar of (Y10-ETA10)T(Y10-ETA10)
!SV11 = DS 1,1,1 (3 SS11 AUS 1 1) $ scalar of (Y11-ETA11)T(Y11-ETA11)
!SV12 = DS 1,1,1 (3 SS12 AUS 1 1) $ scalar of (Y12-ETA12)T(Y12-ETA12)
!SV13 = DS 1,1,1 (3 SS13 AUS 1 1) $ scalar of (Y13-ETA13)T(Y13-ETA13)
!SV14 = DS 1,1,1 (3 SS14 AUS 1 1) $ scalar of (Y14-ETA14)T(Y14-ETA14)
$
!ST1 = "SV1" +"SV2" +"SV3" +"SV4" +"SV5"
!ST2 = "SV6" +"SV7" +"SV8" +"SV9"
!ST3 = "SV10"+"SV11"+"SV12"+"SV13"+"SV14"
$
!STT = "ST1" +"ST2" +"ST3"
$
$ the TABLE,U command below updates the table created before the loop and
$ inserts the objective function that corresponds to the chromosomes
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN PAPO 1 1
  J="LINE":"STT"
$
ERASE 3
$
*IF("LINE" LT "NWD"):GOTO 105
$
*GOTO 8300
$
$-----$
$
*LABEL 8100
*XQT AUS
$
!LINE = 0

```

```

$
*XQT DCU
!NWD = TOC,NJ(1 CHRO CHIL 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 106
$
!LINE = LINE + 1
$
!A1 = DS,1 ,"LINE",1 (1 CHRO CHIL 1)
!A2 = DS,2 ,"LINE",1 (1 CHRO CHIL 1)
!A3 = DS,3 ,"LINE",1 (1 CHRO CHIL 1)
!A4 = DS,4 ,"LINE",1 (1 CHRO CHIL 1)
!A5 = DS,5 ,"LINE",1 (1 CHRO CHIL 1)
!A6 = DS,6 ,"LINE",1 (1 CHRO CHIL 1)
!A7 = DS,7 ,"LINE",1 (1 CHRO CHIL 1)
!A8 = DS,8 ,"LINE",1 (1 CHRO CHIL 1)
!A9 = DS,9 ,"LINE",1 (1 CHRO CHIL 1)
!A10 = DS,10,"LINE",1 (1 CHRO CHIL 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE T68T = 2 TT68 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T67T = 2 TT67 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T66T = 2 TT66 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T65T = 2 TT65 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T64T = 2 TT64 NODE 1 1      $ define table as variable for arith. oper.
$
DEFINE T76T = 2 BB76 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T75T = 2 BB75 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T74T = 2 BB74 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T73T = 2 BB73 NODE 1 1      $ define table as variable for arith. oper.
$
DEFINE T24T = 2 HH24 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T23T = 2 HH23 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T22T = 2 HH22 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T21T = 2 HH21 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T20T = 2 HH20 NODE 1 1      $ define table as variable for arith. oper.
$
DEFINE T68E = 5 EXPE TT68 1 1      $ define table as variable for arith. oper.
DEFINE T67E = 5 EXPE TT67 1 1      $ define table as variable for arith. oper.
DEFINE T66E = 5 EXPE TT66 1 1      $ define table as variable for arith. oper.
DEFINE T65E = 5 EXPE TT65 1 1      $ define table as variable for arith. oper.
DEFINE T64E = 5 EXPE TT64 1 1      $ define table as variable for arith. oper.
$
DEFINE T76E = 5 EXPE BB76 1 1      $ define table as variable for arith. oper.
DEFINE T75E = 5 EXPE BB75 1 1      $ define table as variable for arith. oper.
DEFINE T74E = 5 EXPE BB74 1 1      $ define table as variable for arith. oper.
DEFINE T73E = 5 EXPE BB73 1 1      $ define table as variable for arith. oper.

```

```

$
DEFINE T24E = 5 EXPE HH24 1 1      $ define table as variable for arith. oper.
DEFINE T23E = 5 EXPE HH23 1 1      $ define table as variable for arith. oper.
DEFINE T22E = 5 EXPE HH22 1 1      $ define table as variable for arith. oper.
DEFINE T21E = 5 EXPE HH21 1 1      $ define table as variable for arith. oper.
DEFINE T20E = 5 EXPE HH20 1 1      $ define table as variable for arith. oper.
$
DEFINE ON  = 7 ONES MATX 1 1      $ define table as variable for arith. oper.
$
SU1  = SUM(T68E,-1. T68T)          $ (Y1 -ETA1 )
SU2  = SUM(T67E,-1. T67T)          $ (Y2 -ETA2 )
SU3  = SUM(T66E,-1. T66T)          $ (Y3 -ETA3 )
SU4  = SUM(T65E,-1. T65T)          $ (Y4 -ETA4 )
SU5  = SUM(T64E,-1. T64T)          $ (Y5 -ETA5 )
SU6  = SUM(T76E,-1. T76T)          $ (Y6 -ETA6 )
SU7  = SUM(T75E,-1. T75T)          $ (Y7 -ETA7 )
SU8  = SUM(T74E,-1. T74T)          $ (Y8 -ETA8 )
SU9  = SUM(T73E,-1. T73T)          $ (Y9 -ETA9 )
SU10 = SUM(T24E,-1. T24T)          $ (Y10-ETA10)
SU11 = SUM(T23E,-1. T23T)          $ (Y11-ETA11)
SU12 = SUM(T22E,-1. T22T)          $ (Y12-ETA12)
SU13 = SUM(T21E,-1. T21T)          $ (Y13-ETA13)
SU14 = SUM(T20E,-1. T20T)          $ (Y14-ETA14)
$
SS1  = XTY(SU1,SU1)                $ (Y1-ETA1)T(Y1-ETA1)
SS2  = XTY(SU2,SU2)                $ (Y2-ETA2)T(Y2-ETA2)
SS3  = XTY(SU3,SU3)                $ (Y3-ETA3)T(Y3-ETA3)
SS4  = XTY(SU4,SU4)                $ (Y4-ETA4)T(Y4-ETA4)
SS5  = XTY(SU5,SU5)                $ (Y5-ETA5)T(Y5-ETA5)
SS6  = XTY(SU6,SU6)                $ (Y6-ETA6)T(Y6-ETA6)
SS7  = XTY(SU7,SU7)                $ (Y7-ETA7)T(Y7-ETA7)
SS8  = XTY(SU8,SU8)                $ (Y8-ETA8)T(Y8-ETA8)
SS9  = XTY(SU9,SU9)                $ (Y9-ETA9)T(Y9-ETA9)
SS10 = XTY(SU10,SU10)              $ (Y10-ETA10)T(Y10-ETA10)
SS11 = XTY(SU11,SU11)              $ (Y11-ETA11)T(Y11-ETA11)
SS12 = XTY(SU12,SU12)              $ (Y12-ETA12)T(Y12-ETA12)
SS13 = XTY(SU13,SU13)              $ (Y13-ETA13)T(Y13-ETA13)
SS14 = XTY(SU14,SU14)              $ (Y14-ETA14)T(Y14-ETA14)
$
!SV1 = DS 1,1,1 (3 SS1  AUS 1 1)   $ scalar of (Y1-ETA1)T(Y1-ETA1)
!SV2 = DS 1,1,1 (3 SS2  AUS 1 1)   $ scalar of (Y2-ETA2)T(Y2-ETA2)
!SV3 = DS 1,1,1 (3 SS3  AUS 1 1)   $ scalar of (Y3-ETA3)T(Y3-ETA3)
!SV4 = DS 1,1,1 (3 SS4  AUS 1 1)   $ scalar of (Y4-ETA4)T(Y4-ETA4)
!SV5 = DS 1,1,1 (3 SS5  AUS 1 1)   $ scalar of (Y5-ETA5)T(Y5-ETA5)
!SV6 = DS 1,1,1 (3 SS6  AUS 1 1)   $ scalar of (Y6-ETA6)T(Y6-ETA6)
!SV7 = DS 1,1,1 (3 SS7  AUS 1 1)   $ scalar of (Y7-ETA7)T(Y7-ETA7)
!SV8 = DS 1,1,1 (3 SS8  AUS 1 1)   $ scalar of (Y8-ETA8)T(Y8-ETA8)
!SV9 = DS 1,1,1 (3 SS9  AUS 1 1)   $ scalar of (Y9-ETA9)T(Y9-ETA9)
!SV10 = DS 1,1,1 (3 SS10 AUS 1 1)   $ scalar of (Y10-ETA10)T(Y10-ETA10)
!SV11 = DS 1,1,1 (3 SS11 AUS 1 1)   $ scalar of (Y11-ETA11)T(Y11-ETA11)
!SV12 = DS 1,1,1 (3 SS12 AUS 1 1)   $ scalar of (Y12-ETA12)T(Y12-ETA12)
!SV13 = DS 1,1,1 (3 SS13 AUS 1 1)   $ scalar of (Y13-ETA13)T(Y13-ETA13)
!SV14 = DS 1,1,1 (3 SS14 AUS 1 1)   $ scalar of (Y14-ETA14)T(Y14-ETA14)
$
!ST1 = "SV1" +"SV2" +"SV3" +"SV4" +"SV5"
!ST2 = "SV6" +"SV7" +"SV8" +"SV9"
!ST3 = "SV10"+"SV11"+"SV12"+"SV13"+"SV14"
$
!STT = "ST1" +"ST2" +"ST3"

```

```

$
TABLE,U(NI=1,NJ="NWD"):1 OBFN CHIL 1 1
  J="LINE": "STT"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 106
$
*GOTO 8300
$
$-----$
$
*LABEL 8200
*XQT AUS
$
!LINE = 0
$
*XQT DCU
!NWD = TOC,NJ(1 FRES BLOD 1)
*XQT AUS
$
TABLE(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
TABLE(NI=1,NJ="NDS"):7 ONES MATX 1 1
  J=1,"NDS": 1.0
$
*LABEL 107
$
!LINE = LINE + 1
$
!A1 = DS,1 ,"LINE",1 (1 FRES BLOD 1)
!A2 = DS,2 ,"LINE",1 (1 FRES BLOD 1)
!A3 = DS,3 ,"LINE",1 (1 FRES BLOD 1)
!A4 = DS,4 ,"LINE",1 (1 FRES BLOD 1)
!A5 = DS,5 ,"LINE",1 (1 FRES BLOD 1)
!A6 = DS,6 ,"LINE",1 (1 FRES BLOD 1)
!A7 = DS,7 ,"LINE",1 (1 FRES BLOD 1)
!A8 = DS,8 ,"LINE",1 (1 FRES BLOD 1)
!A9 = DS,9 ,"LINE",1 (1 FRES BLOD 1)
!A10 = DS,10,"LINE",1 (1 FRES BLOD 1)
$
*DCALL (29 MATE PROP)
*DCALL (29 TRAN ANAL)
*DCALL (29 TEMP TABL)
ERASE 6
$
INLIB = 3
OUTLI = 3
$
DEFINE T68T = 2 TT68 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T67T = 2 TT67 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T66T = 2 TT66 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T65T = 2 TT65 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T64T = 2 TT64 NODE 1 1      $ define table as variable for arith. oper.
$
DEFINE T76T = 2 BB76 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T75T = 2 BB75 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T74T = 2 BB74 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T73T = 2 BB73 NODE 1 1      $ define table as variable for arith. oper.
$

```

```

DEFINE T24T = 2 HH24 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T23T = 2 HH23 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T22T = 2 HH22 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T21T = 2 HH21 NODE 1 1      $ define table as variable for arith. oper.
DEFINE T20T = 2 HH20 NODE 1 1      $ define table as variable for arith. oper.
$
DEFINE T68E = 5 EXPE TT68 1 1      $ define table as variable for arith. oper.
DEFINE T67E = 5 EXPE TT67 1 1      $ define table as variable for arith. oper.
DEFINE T66E = 5 EXPE TT66 1 1      $ define table as variable for arith. oper.
DEFINE T65E = 5 EXPE TT65 1 1      $ define table as variable for arith. oper.
DEFINE T64E = 5 EXPE TT64 1 1      $ define table as variable for arith. oper.
$
DEFINE T76E = 5 EXPE BB76 1 1      $ define table as variable for arith. oper.
DEFINE T75E = 5 EXPE BB75 1 1      $ define table as variable for arith. oper.
DEFINE T74E = 5 EXPE BB74 1 1      $ define table as variable for arith. oper.
DEFINE T73E = 5 EXPE BB73 1 1      $ define table as variable for arith. oper.
$
DEFINE T24E = 5 EXPE HH24 1 1      $ define table as variable for arith. oper.
DEFINE T23E = 5 EXPE HH23 1 1      $ define table as variable for arith. oper.
DEFINE T22E = 5 EXPE HH22 1 1      $ define table as variable for arith. oper.
DEFINE T21E = 5 EXPE HH21 1 1      $ define table as variable for arith. oper.
DEFINE T20E = 5 EXPE HH20 1 1      $ define table as variable for arith. oper.
$
DEFINE ON  = 7 ONES MATX 1 1      $ define table as variable for arith. oper.
$
SU1  = SUM(T68E,-1. T68T)          $ (Y1 -ETA1 )
SU2  = SUM(T67E,-1. T67T)          $ (Y2 -ETA2 )
SU3  = SUM(T66E,-1. T66T)          $ (Y3 -ETA3 )
SU4  = SUM(T65E,-1. T65T)          $ (Y4 -ETA4 )
SU5  = SUM(T64E,-1. T64T)          $ (Y5 -ETA5 )
SU6  = SUM(T76E,-1. T76T)          $ (Y6 -ETA6 )
SU7  = SUM(T75E,-1. T75T)          $ (Y7 -ETA7 )
SU8  = SUM(T74E,-1. T74T)          $ (Y8 -ETA8 )
SU9  = SUM(T73E,-1. T73T)          $ (Y9 -ETA9 )
SU10 = SUM(T24E,-1. T24T)          $ (Y10-ETA10)
SU11 = SUM(T23E,-1. T23T)          $ (Y11-ETA11)
SU12 = SUM(T22E,-1. T22T)          $ (Y12-ETA12)
SU13 = SUM(T21E,-1. T21T)          $ (Y13-ETA13)
SU14 = SUM(T20E,-1. T20T)          $ (Y14-ETA14)
$
SS1  = XTY(SU1,SU1)                $ (Y1-ETA1)T(Y1-ETA1)
SS2  = XTY(SU2,SU2)                $ (Y2-ETA2)T(Y2-ETA2)
SS3  = XTY(SU3,SU3)                $ (Y3-ETA3)T(Y3-ETA3)
SS4  = XTY(SU4,SU4)                $ (Y4-ETA4)T(Y4-ETA4)
SS5  = XTY(SU5,SU5)                $ (Y5-ETA5)T(Y5-ETA5)
SS6  = XTY(SU6,SU6)                $ (Y6-ETA6)T(Y6-ETA6)
SS7  = XTY(SU7,SU7)                $ (Y7-ETA7)T(Y7-ETA7)
SS8  = XTY(SU8,SU8)                $ (Y8-ETA8)T(Y8-ETA8)
SS9  = XTY(SU9,SU9)                $ (Y9-ETA9)T(Y9-ETA9)
SS10 = XTY(SU10,SU10)              $ (Y10-ETA10)T(Y10-ETA10)
SS11 = XTY(SU11,SU11)              $ (Y11-ETA11)T(Y11-ETA11)
SS12 = XTY(SU12,SU12)              $ (Y12-ETA12)T(Y12-ETA12)
SS13 = XTY(SU13,SU13)              $ (Y13-ETA13)T(Y13-ETA13)
SS14 = XTY(SU14,SU14)              $ (Y14-ETA14)T(Y14-ETA14)
$
!SV1 = DS 1,1,1 (3 SS1  AUS 1 1)  $ scalar of (Y1-ETA1)T(Y1-ETA1)
!SV2 = DS 1,1,1 (3 SS2  AUS 1 1)  $ scalar of (Y2-ETA2)T(Y2-ETA2)
!SV3 = DS 1,1,1 (3 SS3  AUS 1 1)  $ scalar of (Y3-ETA3)T(Y3-ETA3)
!SV4 = DS 1,1,1 (3 SS4  AUS 1 1)  $ scalar of (Y4-ETA4)T(Y4-ETA4)

```



```

!SV5 = DS 1,1,1 (3 SS5 AUS 1 1) $ scalar of (Y5-ETA5)T(Y5-ETA5)
!SV6 = DS 1,1,1 (3 SS6 AUS 1 1) $ scalar of (Y6-ETA6)T(Y6-ETA6)
!SV7 = DS 1,1,1 (3 SS7 AUS 1 1) $ scalar of (Y7-ETA7)T(Y7-ETA7)
!SV8 = DS 1,1,1 (3 SS8 AUS 1 1) $ scalar of (Y8-ETA8)T(Y8-ETA8)
!SV9 = DS 1,1,1 (3 SS9 AUS 1 1) $ scalar of (Y9-ETA9)T(Y9-ETA9)
!SV10 = DS 1,1,1 (3 SS10 AUS 1 1) $ scalar of (Y10-ETA10)T(Y10-ETA10)
!SV11 = DS 1,1,1 (3 SS11 AUS 1 1) $ scalar of (Y11-ETA11)T(Y11-ETA11)
!SV12 = DS 1,1,1 (3 SS12 AUS 1 1) $ scalar of (Y12-ETA12)T(Y12-ETA12)
!SV13 = DS 1,1,1 (3 SS13 AUS 1 1) $ scalar of (Y13-ETA13)T(Y13-ETA13)
!SV14 = DS 1,1,1 (3 SS14 AUS 1 1) $ scalar of (Y14-ETA14)T(Y14-ETA14)
$
!ST1 = "SV1" + "SV2" + "SV3" + "SV4" + "SV5"
!ST2 = "SV6" + "SV7" + "SV8" + "SV9"
!ST3 = "SV10" + "SV11" + "SV12" + "SV13" + "SV14"
$
!STT = "ST1" + "ST2" + "ST3"
$
TABLE,U(NI=1,NJ="NWD"):1 OBFN BLOD 1 1
J="LINE": "STT"
$
ERASE 3
$
*IF("LINE" LT "NWD"): *GOTO 107
$
*LABEL 8300
$
*RETURN
*SQUA
$
$-----$
$ Subroutine (29 ITER ATIO)ATIO: This subroutine updates the thermal$
$                               properties of the parameters and $
$                               executes the genetic processors $
$                               n-times for n-generations. $
$-----$
$
*(29 ITER ATIO)ATIO
$
*XQT AUS
$
!ITER = 0 $ initializing the counter for the loop
$
TABLE(NI=1,NJ=50):8 CONV CRIT 1 1 $ initializing the table containing the
I=1:J=1,50:0.0 $ average of the first 20 best objective
$ $ functions per iteration
*LABEL 400
$
!ITER = ITER + 1 $ counter for the loop
$
!SSQU = 2 $ flag for the SUMO SQUA subroutine
$
*DCALL (29 GENC PROC) $ call subroutine GENC PROC
*DCALL (29 SUMO SQUA)) $ call subroutine SUMO SQUA
ERASE 2 $ erase content of library 2
ERASE 3 $ erase content of library 3
ERASE 4 $ erase content of library 4
ERASE 6 $ erase content of library 6
$
!SSQU = 3 $ flag for the SUMO SQUA subroutine

```

```

$
*DCALL (29 GEND PROC)          $ call subroutine GEND PROC
*DCALL (29 SUMO SQUA)          $ call subroutine SUMO SQUA
$
!MIN1 = DS,1,1,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN2 = DS,1,2,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN3 = DS,1,3,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN4 = DS,1,4,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN5 = DS,1,5,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN6 = DS,1,6,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN7 = DS,1,7,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MIN8 = DS,1,8,1 (1 NEWW PARA) $ reset the lower boundary for K
!MIN9 = DS,1,9,1 (1 NEWW PARA) $ reset the lower boundary for Cp
!MINX = DS,1,10,1 (1 NEWW PARA) $ reset the lower boundary for K
!MAX1 = DS,1,11,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX2 = DS,1,12,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX3 = DS,1,13,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX4 = DS,1,14,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX5 = DS,1,15,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX6 = DS,1,16,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX7 = DS,1,17,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAX8 = DS,1,18,1 (1 NEWW PARA) $ reset the upper boundary for K
!MAX9 = DS,1,19,1 (1 NEWW PARA) $ reset the upper boundary for Cp
!MAXX = DS,1,20,1 (1 NEWW PARA) $ reset the upper boundary for K
$
*DCALL (29 GEAL VARI)          $call the subroutine GEAL VARI
  ERASE 2                      $erase content of library 2
  ERASE 3                      $erase content of library 3
  ERASE 4                      $erase content of library 4
  ERASE 6                      $erase content of library 6
*DCALL (29 GENE PROC)          $call subroutine GENE PROC
$
*XQT AUS
$
!OB1 = DS 1,1 ,1 (1 ITER OBFN 1 1)
!OB2 = DS 1,2 ,1 (1 ITER OBFN 1 1)
!OB3 = DS 1,3 ,1 (1 ITER OBFN 1 1)
!OB4 = DS 1,4 ,1 (1 ITER OBFN 1 1)
!OB5 = DS 1,5 ,1 (1 ITER OBFN 1 1)
!OB6 = DS 1,6 ,1 (1 ITER OBFN 1 1)
!OB7 = DS 1,7 ,1 (1 ITER OBFN 1 1)
!OB8 = DS 1,8 ,1 (1 ITER OBFN 1 1)
!OB9 = DS 1,9 ,1 (1 ITER OBFN 1 1)
!OB10 = DS 1,10 ,1 (1 ITER OBFN 1 1)
!OB11 = DS 1,11 ,1 (1 ITER OBFN 1 1)
!OB12 = DS 1,12 ,1 (1 ITER OBFN 1 1)
!OB13 = DS 1,13 ,1 (1 ITER OBFN 1 1)
!OB14 = DS 1,14 ,1 (1 ITER OBFN 1 1)
!OB15 = DS 1,15 ,1 (1 ITER OBFN 1 1)
!OB16 = DS 1,16 ,1 (1 ITER OBFN 1 1)
!OB17 = DS 1,17 ,1 (1 ITER OBFN 1 1)
!OB18 = DS 1,18 ,1 (1 ITER OBFN 1 1)
!OB19 = DS 1,19 ,1 (1 ITER OBFN 1 1)
!OB20 = DS 1,20 ,1 (1 ITER OBFN 1 1)
$
!OB1
!OB2
!OB3
!OB4

```

```

!OB5
!OB6
!OB7
!OB8
!OB9
!OB10
!OB11
!OB12
!OB13
!OB14
!OB15
!OB16
!OB17
!OB18
!OB19
!OB20
$
!OBF1 = ("OB1" + "OB2" + "OB3" + "OB4" + "OB5" )
!OBF2 = ("OB6" + "OB7" + "OB8" + "OB9" + "OB10")
!OBF3 = ("OB11" + "OB12" + "OB13" + "OB14" + "OB15")
!OBF4 = ("OB16" + "OB17" + "OB18" + "OB19" + "OB20")
$
!OBF = (("OBF1" + "OBF2" + "OBF3" + "OBF4")/20)
$
TABLE,U(NI=1,NJ=50):8 CONV CRIT 1 1
  I=1
  J="ITER": "OBF"
$
*XQT DCU
PRINT 8 CONV CRIT 1 1
*XQT AUS
$
*IF("ITER" GE 5):*GOTO 500
*GOTO 510
*LABEL 500
$
!NUM1 = "ITER" - 2
!NUM2 = "ITER" - 1
!NUM3 = "ITER"
!CON1 = DS 1,"NUM1",1 (8 CONV CRIT 1 1)
!CON2 = DS 1,"NUM2",1 (8 CONV CRIT 1 1)
!CON3 = DS 1,"NUM3",1 (8 CONV CRIT 1 1)
$
!CON4 = ("CON1"- "CON2")/"CON1"
!CON5 = ("CON2"- "CON3")/"CON2"
$
*XQT DCU
PRINT 1 ITER PAPO
PRINT 1 ITER OBFN
*XQT AUS
$
*IF("CON4" LE 0.01):*GOTO 520
*GOTO 510
$
*LABEL 520
*IF("CON5" LE 0.01):*GOTO 410
*GOTO 510
$
*LABEL 510

```

```

*IF("ITER" EQ "ITMX"):GOTO 410
*GOTO 400
*LABEL 410
$
*RETURN
*AT10
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.             $
$-----$
$
$(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrta time step for TRTB (seconds)
$
!MIN1 = 0.1      $ lower bound for skins (K)
!MIN2 = 0.2      $ lower bound for air gap (PCp)
!MIN3 = 0.05     $ lower bound for air gap (K)
!MIN4 = 0.05     $ lower bound for honeycomb (K)
!MIN5 = 0.1      $ lower bound for torque tubes (K)
!MIN6 = 0.1      $ lower bound for facesheets (K)
!MIN7 = 10.0     $
!MIN8 = 10.0     $
!MIN9 = 10.0     $
!MINX = 10.0     $
!MAX1 = 10.0     $ upper bound for skins (K)
!MAX2 = 3.0      $ upper bound for air gap (PCp)
!MAX3 = 3.0      $ upper bound for air gap (K)
!MAX4 = 4.0      $ upper bound for honeycomb (K)
!MAX5 = 10.0     $ upper bound for torque tubes (K)
!MAX6 = 10.0     $ upper bound for facesheets (K)
!MAX7 = 10.0     $
!MAX8 = 10.0     $
!MAX9 = 10.0     $
!MAXX = 10.0     $
$
!A1 = 1.6        $ initial value for skins (K)
!A2 = 1.0        $ initial value for air gap (PCp)
!A3 = 0.7        $ initial value for air gap (K)
!A4 = 0.4        $ initial value for honeycomb (K)
!A5 = 1.9        $ initial value for torque tubes (K)
!A6 = 1.6        $ initial value for facesheets (K)
!A7 = 1.00       $
!A8 = 1.00       $
!A9 = 1.00       $
!A10 = 1.00      $
$
!A1I = 1.6       $ initial value for skins (K)
!A2I = 1.0       $ initial value for air gap (PCp)
!A3I = 0.7       $ initial value for air gap (K)
!A4I = 0.4       $ initial value for honeycomb (K)
!A5I = 1.9       $ initial value for torque tubes (K)
!A6I = 1.6       $ initial value for facesheets (K)
!A7I = 1.00      $
!A8I = 1.00      $
!A9I = 1.00      $

```

```

!A10I = 1.00      $
$
*RETURN
*SPEC
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                      $
$-----$
$
*(29 TEMP TABL)TEMP
$
*XQT AUS
ERASE 2
*XQT DCU          $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1      $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1     $ copy the dataset from lib 4 to lib 2
$
*XQT AUS          $ arithmetic utility processor
ERASE 4
$
!VA1 = TOC,NWDS(2 TRAN TIME)  $ number of words in dataset TRAN TIME
!VA2 = "BB1"-1
!VA3 = "BB2"-1
!VA4 = "BB3"-1
!VA5 = "BB4"-1
!VA6 = "BB5"-1
!VA7 = "TT1"-1
!VA8 = "TT2"-1
!VA9 = "TT3"-1
!VA10 = "TT4"-1
!VA11 = "TT5"-1
!VA12 = "HH1"-1
!VA13 = "HH2"-1
!VA14 = "HH3"-1
!VA15 = "HH4"-1
!VA16 = "HH5"-1
$
DEFINE X = 2 TRAN TEMP 1 1
DEFINE Y = 2 TRAN TIME 1 1
DEFINE Z = 5 EXPE HH20 1 1
$
*IF("N2" EQ 1) :*GOTO 5000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2) :*GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3) :*GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4) :*GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5) :*GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6) :*GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7) :*GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8) :*GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9) :*GOTO 5080 $ Temperature histories for property 7
*IF("N2" EQ 10):*GOTO 5082 $ Temperature histories for property 8
*IF("N2" EQ 11):*GOTO 5084 $ Temperature histories for property 9
*IF("N2" EQ 12):*GOTO 5086 $ Temperature histories for property 10
$
*LABEL 5000

```

```

$
TABLE(NI=3,NJ="VA1"):9 TEMP TABL
  TRAN(SOUR=Y,ILIM=1,JLIM="VA1",DBASE=0,DSKIP=2)
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA11",DBASE=1,DSKIP=2)
  TRAN(SOUR=Z,ILIM=1,JLIM="VA1",DBASE=2,DSKIP=2)
$
TABLE(NI=1,NJ="VA1"):2 TT68 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA11")
TABLE(NI=1,NJ="VA1"):2 TT67 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA10")
TABLE(NI=1,NJ="VA1"):2 TT66 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA9")
TABLE(NI=1,NJ="VA1"):2 TT65 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA8")
TABLE(NI=1,NJ="VA1"):2 TT64 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA7")
$
TABLE(NI=1,NJ="VA1"):2 BB76 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA5")
TABLE(NI=1,NJ="VA1"):2 BB75 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA4")
TABLE(NI=1,NJ="VA1"):2 BB73 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA3")
TABLE(NI=1,NJ="VA1"):2 BB74 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA2")
$
TABLE(NI=1,NJ="VA1"):2 HH24 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA16")
TABLE(NI=1,NJ="VA1"):2 HH23 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA15")
TABLE(NI=1,NJ="VA1"):2 HH22 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA14")
TABLE(NI=1,NJ="VA1"):2 HH21 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA13")
TABLE(NI=1,NJ="VA1"):2 HH20 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
$
$
*GOTO 6000
$
*LABEL 5010
TABLE(NI=1,NJ="VA1"):2 INIT 00 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VA1"):2 PROP I 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VA1"):2 PROP II 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VA1"):2 PROP III 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$

```

```

*LABEL 5050
TABLE(NI=1,NJ="VA1"):2 PROP IV 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VA1"):2 PROP V 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VA1"):2 PROP VI 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VA1"):2 PROP VII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5082
TABLE(NI=1,NJ="VA1"):2 PROP VIII 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5084
TABLE(NI=1,NJ="VA1"):2 PROP IX 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5086
TABLE(NI=1,NJ="VA1"):2 PROP X 1 1
    TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ Subroutine (29 SENS FIND): this subroutine determines which nodes corres $
$ pond best to the sensor locations on the test $
$ panel. As the mesh is refined, the node number $
$ can change without affecting the rest of the $
$ program. $
$-----$
$
$(29 SENS FIND)FIND
$
*XQT AUS
$
!ROW0 = 0
!ROW1 = "YJ"*("NY1"+"EY2"+"EY3")+1
!ROW2 = "ROW1"+"YJ"
$
*LABEL 10
$
!RED = 0
$

```

```

!ROW0 = "ROW0" + 1
$
*IF("ROW0" EQ "ROW1"):GOTO 30
$
*GOTO 10
$
*LABEL 30
$
!XCOR = DS 1,"ROW0",1 (1 JLOC BTAB)
$
!XDF1 = "XCOR" - 0.0565
!XDF2 = "XCOR" - 0.1835
!XDF3 = "XCOR" - 0.2597
!XDF4 = "XCOR" - 0.3740
$
*IF("XDF4" GT 0.0):GOTO 40
*IF("RED" EQ 1) :GOTO 100
*IF("XDF3" GT 0.0):GOTO 50
*IF("RED" EQ 2) :GOTO 100
*IF("XDF2" GT 0.0):GOTO 60
*IF("RED" EQ 3) :GOTO 100
*IF("XDF1" GT 0.0):GOTO 70
*GOTO 100
$
*LABEL 40
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF4")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 42
*GOTO 44
*LABEL 42
!N44 = "ROW7"
*GOTO 80
*LABEL 44
!N44 = "ROW0"
*GOTO 80
$
*LABEL 50
!RED = 1
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF3")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 52
*GOTO 54
*LABEL 52
!N33 = "ROW7"
*GOTO 100
*LABEL 54
!N33 = "ROW0"
*GOTO 100
$
*LABEL 60
!RED = 2
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF2")
!ABS2 = ABS("XPRES")

```



```

*IF("ABS1" GT "ABS2"):GOTO 62
*GOTO 64
*LABEL 62
!N22 = "ROW7"
*GOTO 100
*LABEL 64
!N22 = "ROW0"
*GOTO 100
$
*LABEL 70
!RED = 3
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF1")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 72
*GOTO 74
*LABEL 72
!N11 = "ROW7"
*GOTO 100
*LABEL 74
!N11 = "ROW0"
*GOTO 100
$
*LABEL 80
!BB1 = "ROW2"-1
!BB2 = "N44"
!BB3 = "N33"
!BB4 = "N22"
!BB5 = "N11"
!TT1 = "BB1"+"YJ"*"EY4"
!TT2 = "BB2"+"YJ"*"EY4"
!TT3 = "BB3"+"YJ"*"EY4"
!TT4 = "BB4"+"YJ"*"EY4"
!TT5 = "BB5"+"YJ"*"EY4"
!HH1 = "TT1"+"YJ"*("EY5"+"EY6"+"EY7"+"EY8")
!HH2 = "TT2"+"YJ"*("EY5"+"EY6"+"EY7"+"EY8")
!HH3 = "TT3"+"YJ"*("EY5"+"EY6"+"EY7"+"EY8")
!HH4 = "TT4"+"YJ"*("EY5"+"EY6"+"EY7"+"EY8")
!HH5 = "TT5"+"YJ"*("EY5"+"EY6"+"EY7"+"EY8")
$
*GOTO 200
$
*LABEL 100
!ROW0 = "ROW0" + 1
$
*IF("ROW0" EQ "ROW2"):GOTO 10
*GOTO 30
$
*LABEL 200
$
!RED = FREE()
!ABS1 = FREE()
!ABS2 = FREE()
!XPRES = FREE()
!XCOR = FREE()
!XDF1 = FREE()
!XDF2 = FREE()
!XDF3 = FREE()

```

```

!XDF4 = FREE()
!ROW0 = FREE()
!ROW1 = FREE()
!ROW2 = FREE()
!ROW7 = FREE()
!N11 = FREE()
!N22 = FREE()
!N33 = FREE()
!N44 = FREE()
$
*RETURN
*FIND
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS
$
*TF OPEN 1'data1
*TF READ 1
*TF CLOSE 1
*TF OPEN 1'data2
*TF READ 1
*TF CLOSE 1
$
!N2 = 1
$
*DCALL (29 MAKE VARI) $ Call subroutine MAKE VARI
*DCALL (29 PROB SPEC) $ Call subroutine PROB SPEC
*DCALL (29 GEAL VARI) $ Call subroutine GEAL VARI
*DCALL (29 NODE GENE) $ Call subroutine NODE GENE
*DCALL (29 ELEM GENE) $ Call subroutine MAKE ELEM
*DCALL (29 ELEM FREE) $ Call subroutine ELEM FREE
*DCALL (29 SENS FIND) $ Call subroutine SENS FIND
*DCALL (29 BOUN COND) $ Call subroutine BOUN COND
!ITPP = 1 $ Reset parameter for initial population 1
!SSQU = 1
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 2
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 3
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC

```

```

ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 4
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
!ITPP = 5
*DCALL (29 GENA PROC) $ Call subroutine GENA PROC
*DCALL (29 SUMO SQUA) $ Call subroutine SUMO SQUA
*DCALL (29 GENB PROC) $ Call subroutine GENB PROC
ERASE 2
ERASE 3
ERASE 4
ERASE 6
$
*XQT EXIT
!ITMX = 10           $ Maximum number of iteration
*DCALL (29 ITER ATIO) $ Call subroutine ITER ATIO
$
*XQT DCU
  PRINT 1 ITER PAPO
  PRINT 1 ITER OBFN
*XQT EXIT

```

Appendix N

EAL Two-Dimensional Model: Forward Program

```
*XQT U1
!MAXCM = SSP(0,13)
*CMPARA (MAXCM="MAXCM")
*CM="MAXCM"
$
$-----$
$ Subroutine (29 MAKE VARI)VARI: This subroutine defines some of $
$                               the variables used in the program $
$                               such as: $
$                               - Number of nodes for each section $
$                               - Number of elements for each $
$                               section $
$                               Note: NX2 can vary from 2 to ... $
$                               NX1 can vary from 3 to ... $
$                               NX3 can vary from 3 to ... $
$                               NX5 can vary from 3 to ... $
$                               NX7 can vary from 3 to ... $
$                               NX5 should be odd $
$-----$
$
$(29 MAKE VARI)VARI
$
*XQT AUS           $arithmetic utility system processor
$
!NX1 = 8           $ number of nodes for left honeycomb edge closeout
!NX2 = 2           $ number of nodes for rib 1
!NX3 = 30          $ number of nodes for torque tube 1
!NX4 = "NX2"       $ number of nodes for rib 2
!NX5 = 33          $ number of nodes for torque tube 2
!NX6 = "NX2"       $ number of nodes for rib 3
!NX7 = 38          $ number of nodes for torque tube 3
!NX8 = "NX2"       $ number of nodes for rib 4
!NX9 = 8           $ number of nodes for right honeycomb edge closeout
$
!EX1 = "NX1"-1     $ number of elements for left honeycomb edge closeout
!EX2 = "NX2"-1     $ number of elements for rib 1
```

```

!EX3 = "NX3"-1      $ number of elements for torque tube 1
!EX4 = "NX4"-1      $ number of elements for rib 2
!EX5 = "NX5"-1      $ number of elements for torque tube 2
!EX6 = "NX6"-1      $ number of elements for rib 3
!EX7 = "NX7"-1      $ number of elements for torque tube 3
!EX8 = "NX8"-1      $ number of elements for rib 4
!EX9 = "NX9"-1      $ number of elements for right honeycomb edge closeout
$
!NY1 = 2            $ number of nodes for the lower outer facesheet
!NY2 = 9            $ number of nodes for the lower honeycomb structure
!NY3 = 3            $ number of nodes for the lower inner facesheet
!NY4 = 2            $ number of nodes for the lower torque tube
!NY5 = 17           $ number of nodes for the air cavity
!NY6 = 2            $ number of nodes for the upper torque tube
!NY7 = 3            $ number of nodes for the upper inner facesheet
!NY8 = 9            $ number of nodes for the upper honeycomb structure
!NY9 = 2            $ number of nodes for the upper outer facesheet
!NY10 = 9           $ number of nodes for the insulation
$
!EY1 = "NY1"-1      $ number of elements for the lower outer facesheet
!EY2 = "NY2"-1      $ number of elements for the lower honeycomb structure
!EY3 = "NY3"-1      $ number of elements for the lower inner facesheet
!EY4 = "NY4"-1      $ number of elements for the lower torque tube
!EY5 = "NY5"-1      $ number of elements for the air cavity
!EY6 = "NY6"-1      $ number of elements for the upper torque tube
!EY7 = "NY7"-1      $ number of elements for the upper inner facesheet
!EY8 = "NY8"-1      $ number of elements for the upper honeycomb structure
!EY9 = "NY9"-1      $ number of elements for the upper outer facesheet
!EY10 = "NY10"-1    $ number of elements for the insulation
$
!XJ = 1             $ nodal increment in the x-direction
$
!YJ = "NX1"+"EX2"+"EX3"+"EX4"+"EX5"+"EX6"+"EX7"+"EX8"+"EX9"
$                  $ nodal increment in the y-direction
!YJE = "EX1"+"EX2"+"EX3"+"EX4"+"EX5"+"EX6"+"EX7"+"EX8"+"EX9"
$                  $ nodal increment in the y-direction -1
!TOTN = "YJ"*( "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"+"EY10")
$                  $total number of nodes in the model
!NON1 = "YJ"*( "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9")
$                  $node number at the heater
!NON2 = "NY1"+"EY2"+"EY3"+"EY4"
$                  $node number at the inside of the lower torque tube
!NON3 = "NY1"+"EY2"+"EY3"+"EY4"+"EY5"
$                  $node number at the inside of the upper torque tube
!NONJ = FLOAT("TOTN")$node number on top of the insulation
$
*RETURN
*VARI
$
$-----$
$ Subroutine (29 MATE PROP)PROP: This subroutine contains the $
$                               properties of the materials used $
$                               in the model. $
$-----$
$
$(29 MATE PROP)PROP
$
*XQT AUS
$

```

```

$ conduction properties for IM7/954-2A
$ upper and lower skins
$
TABLE(NI=9,NJ=1):6 COND PROP 1: I = 2 3 4 5
      J=1: 1.E+6 1.1688213 1.5596224 "A1"
$
$ conduction properties for heater
$
TABLE(NI=9,NJ=1):6 COND PROP 2: I = 3 4 5 6
      J=1: 0. 0. 0. 0.
$
$ conduction properties for insulation
$
TABLE(NI=9,NJ=1):6 COND PROP 3: I = 2 3 4 5 6
      J=1: 256.29 1046.64 0.02254 0.02254 0.02254
$
$ conduction properties for air
$
TABLE(NI=9,NJ=1):6 COND PROP 4:I=2 3 4 5
      J=1: 1.E+3 "A2" "A3" "A3"
$      J=1: 0.9950 1009.0 0.7025
$
$TABLE(NI=9,NJ=14):6 COND PROP 4:I=1 2 3 4 5 6
$      J= 1: 250.0 1.3947 1006. 0.1758 0.1758 0.1758
$      J= 2: 300.0 1.1614 1007. 0.2916 0.2916 0.2916
$      J= 3: 350.0 0.9950 1009. 0.4513 0.4513 0.4513
$      J= 4: 400.0 0.8711 1014. 0.6627 0.6627 0.6627
$      J= 5: 450.0 0.7740 1021. 0.9327 0.9327 0.9327
$      J= 6: 500.0 0.6964 1030. 1.2690 1.2690 1.2690
$      J= 7: 550.0 0.6329 1040. 1.6788 1.6788 1.6788
$      J= 8: 600.0 0.5804 1051. 2.1694 2.1694 2.1694
$      J= 9: 650.0 0.5356 1063. 2.7483 2.7483 2.7483
$      J=10: 700.0 0.4975 1075. 3.4229 3.4229 3.4229
$      J=11: 750.0 0.4643 1087. 4.2005 4.2005 4.2005
$      J=12: 800.0 0.4354 1099. 5.0885 5.0885 5.0885
$      J=13: 850.0 0.4097 1110. 6.0943 6.0943 6.0943
$      J=14: 900.0 0.3868 1121. 7.2256 7.2256 7.2256
$
$ conduction properties for SS321 honeycomb (actually SS347)
$
TABLE(NI=9,NJ=1):6 COND PROP 5: I = 2 3 4 5
      J=1: 1.E+6 0.7037244 0.4064743 "A4"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tube tops, edge channel tops
$
TABLE(NI=9,NJ=1):6 COND PROP 6: I= 2 3 4 5
      J=1: 1.E+6 2.9149498 1.9172322 "A5"
$
$ conduction properties for T300/954-2A 8 harness satin cloth
$ honeycomb edge closeouts, torque tube ribs, edge channel ribs
$
TABLE(NI=9,NJ=1):6 COND PROP 8: I= 2 3 4 5
      J=1: 1.E+6 2.9149498 "A5" 1.9172322
$
$ conduction properties for PETI-5 of the upper and lower outer
$ facesheets.
$
TABLE(NI=9,NJ=1):6 COND PROP 7: I= 2 3 4 5

```

J=1: 1.E+6 2.0342782 1.6259311 "A6"

```
$
*RETURN
*PROP
$
$-----$
$ Subroutine (29 LOWE FACE)LFAC: This subroutine defines the points $
$                               of the lower outer facesheet to $
$                               make the nodes. $
$-----$
$
*(29 LOWE FACE)LFAC
$
!X1 = 0.
!X2 = 0.0285
!X3 = 0.0328
!X4 = 0.1560
!X5 = 0.1602
!X6 = 0.2905
!X7 = 0.2947
!X8 = 0.4555
!X9 = 0.4597
!X10 = 0.4883
!X11 = 0.
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
$
!X1A = 0.      +(0.0285/"EX1")
!X2A = 0.0285+(0.0043/"EX2")
!X3A = 0.0328+(0.1232/"EX3")
!X4A = 0.1560+(0.0042/"EX4")
!X5A = 0.1602+(0.1345/"EX5")
!X6A = 0.2905+(0.0042/"EX6")
!X7A = 0.2947+(0.1608/"EX7")
!X8A = 0.4555+(0.0042/"EX8")
!X9A = 0.4597+(0.0286/"EX9")
!X11A= 0.      +(0.0285/"EX1")
!X12A= 0.0285+(0.0043/"EX2")
!X13A= 0.0328+(0.1232/"EX3")
!X14A= 0.1560+(0.0042/"EX4")
!X15A= 0.1602+(0.1345/"EX5")
!X16A= 0.2905+(0.0042/"EX6")
!X17A= 0.2947+(0.1608/"EX7")
!X18A= 0.4555+(0.0042/"EX8")
!X19A= 0.4597+(0.0286/"EX9")
$
!Y1 = 0.
!Y2 = 0.
!Y3 = 0.
!Y4 = 0.
!Y5 = 0.
!Y6 = 0.
```

```

!Y7 = 0.
!Y8 = 0.
!Y9 = 0.
!Y10 = 0.
!Y11 = 0.0032
!Y12 = 0.0032
!Y13 = 0.0032
!Y14 = 0.0032
!Y15 = 0.0032
!Y16 = 0.0032
!Y17 = 0.0032
!Y18 = 0.0032
!Y19 = 0.0032
!Y20 = 0.0032
$
!Z1 = 0.
!Z2 = 0.
!Z3 = 0.
!Z4 = 0.
!Z5 = 0.
!Z6 = 0.
!Z7 = 0.
!Z8 = 0.
!Z9 = 0.
!Z10 = 0.
!Z11 = 0.
!Z12 = 0.
!Z13 = 0.
!Z14 = 0.
!Z15 = 0.
!Z16 = 0.
!Z17 = 0.
!Z18 = 0.
!Z19 = 0.
!Z20 = 0.
$
*RETURN
*LFAC
$
$-----$
$ Subroutine (29 LOWE HONY)LHON: This subroutine defines the points $
$                               of the lower honeycomb structure $
$                               to make the nodes. $
$-----$
$
*(29 LOWE HONY)LHON
$
!X11 = 0.
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
!X21 = 0.
!X22 = 0.0285

```



```

!X23 = 0.0328
!X24 = 0.1560
!X25 = 0.1602
!X26 = 0.2905
!X27 = 0.2947
!X28 = 0.4555
!X29 = 0.4597
!X30 = 0.4883
$
!X11A= 0.      +(0.0285/"EX1")
!X12A= 0.0285+(0.0043/"EX2")
!X13A= 0.0328+(0.1232/"EX3")
!X14A= 0.1560+(0.0042/"EX4")
!X15A= 0.1602+(0.1345/"EX5")
!X16A= 0.2905+(0.0042/"EX6")
!X17A= 0.2947+(0.1608/"EX7")
!X18A= 0.4555+(0.0042/"EX8")
!X19A= 0.4597+(0.0286/"EX9")
!X21A= 0.      +(0.0285/"EX1")
!X22A= 0.0285+(0.0043/"EX2")
!X23A= 0.0328+(0.1232/"EX3")
!X24A= 0.1560+(0.0042/"EX4")
!X25A= 0.1602+(0.1345/"EX5")
!X26A= 0.2905+(0.0042/"EX6")
!X27A= 0.2947+(0.1608/"EX7")
!X28A= 0.4555+(0.0042/"EX8")
!X29A= 0.4597+(0.0286/"EX9")
$
!Y11 = 0.0032
!Y12 = 0.0032
!Y13 = 0.0032
!Y14 = 0.0032
!Y15 = 0.0032
!Y16 = 0.0032
!Y17 = 0.0032
!Y18 = 0.0032
!Y19 = 0.0032
!Y20 = 0.0032
!Y21 = 0.0286
!Y22 = 0.0286
!Y23 = 0.0286
!Y24 = 0.0286
!Y25 = 0.0286
!Y26 = 0.0286
!Y27 = 0.0286
!Y28 = 0.0286
!Y29 = 0.0286
!Y30 = 0.0286
$
!Y11B= 0.0032+(0.0254/"EY2")
!Y12B= 0.0032+(0.0254/"EY2")
!Y13B= 0.0032+(0.0254/"EY2")
!Y14B= 0.0032+(0.0254/"EY2")
!Y15B= 0.0032+(0.0254/"EY2")
!Y16B= 0.0032+(0.0254/"EY2")
!Y17B= 0.0032+(0.0254/"EY2")
!Y18B= 0.0032+(0.0254/"EY2")
!Y19B= 0.0032+(0.0254/"EY2")
!Y20B= 0.0032+(0.0254/"EY2")

```

```

$
!Z11 = 0.
!Z12 = 0.
!Z13 = 0.
!Z14 = 0.
!Z15 = 0.
!Z16 = 0.
!Z17 = 0.
!Z18 = 0.
!Z19 = 0.
!Z20 = 0.
!Z21 = 0.
!Z22 = 0.
!Z23 = 0.
!Z24 = 0.
!Z25 = 0.
!Z26 = 0.
!Z27 = 0.
!Z28 = 0.
!Z29 = 0.
!Z30 = 0.
$
*RETURN
*LHON
$
$-----$
$ Subroutine (29 LOWE SKIN)LSKI: This subroutine defines the points $
$                               of the lower skin to make the $
$                               nodes. $
$-----$
$
$
*(29 LOWE SKIN)LSKI
$
!X21 = 0.
!X22 = 0.0285
!X23 = 0.0328
!X24 = 0.1560
!X25 = 0.1602
!X26 = 0.2905
!X27 = 0.2947
!X28 = 0.4555
!X29 = 0.4597
!X30 = 0.4883
!X31 = 0.
!X32 = 0.0285
!X33 = 0.0328
!X34 = 0.1560
!X35 = 0.1602
!X36 = 0.2905
!X37 = 0.2947
!X38 = 0.4555
!X39 = 0.4597
!X40 = 0.4883
$
!X21A= 0.      +(0.0285/"EX1")
!X22A= 0.0285+(0.0043/"EX2")
!X23A= 0.0328+(0.1232/"EX3")
!X24A= 0.1560+(0.0042/"EX4")
!X25A= 0.1602+(0.1345/"EX5")

```

```

!X26A= 0.2905+(0.0042/"EX6")
!X27A= 0.2947+(0.1608/"EX7")
!X28A= 0.4555+(0.0042/"EX8")
!X29A= 0.4597+(0.0286/"EX9")
!X31A= 0.      +(0.0285/"EX1")
!X32A= 0.0285+(0.0043/"EX2")
!X33A= 0.0328+(0.1232/"EX3")
!X34A= 0.1560+(0.0042/"EX4")
!X35A= 0.1602+(0.1345/"EX5")
!X36A= 0.2905+(0.0042/"EX6")
!X37A= 0.2947+(0.1608/"EX7")
!X38A= 0.4555+(0.0042/"EX8")
!X39A= 0.4597+(0.0286/"EX9")
$
!Y21 = 0.0286
!Y22 = 0.0286
!Y23 = 0.0286
!Y24 = 0.0286
!Y25 = 0.0286
!Y26 = 0.0286
!Y27 = 0.0286
!Y28 = 0.0286
!Y29 = 0.0286
!Y30 = 0.0286
!Y31 = 0.0326
!Y32 = 0.0326
!Y33 = 0.0326
!Y34 = 0.0326
!Y35 = 0.0326
!Y36 = 0.0326
!Y37 = 0.0326
!Y38 = 0.0326
!Y39 = 0.0326
!Y40 = 0.0326
$
!Y21B= 0.0286+(0.0040/"EY3")
!Y22B= 0.0286+(0.0040/"EY3")
!Y23B= 0.0286+(0.0040/"EY3")
!Y24B= 0.0286+(0.0040/"EY3")
!Y25B= 0.0286+(0.0040/"EY3")
!Y26B= 0.0286+(0.0040/"EY3")
!Y27B= 0.0286+(0.0040/"EY3")
!Y28B= 0.0286+(0.0040/"EY3")
!Y29B= 0.0286+(0.0040/"EY3")
!Y30B= 0.0286+(0.0040/"EY3")
$
!Z21 = 0.
!Z22 = 0.
!Z23 = 0.
!Z24 = 0.
!Z25 = 0.
!Z26 = 0.
!Z27 = 0.
!Z28 = 0.
!Z29 = 0.
!Z30 = 0.
!Z31 = 0.
!Z32 = 0.
!Z33 = 0.

```

```

!Z34 = 0.
!Z35 = 0.
!Z36 = 0.
!Z37 = 0.
!Z38 = 0.
!Z39 = 0.
!Z40 = 0.
$
*RETURN
*LSKI
$
$-----$
$ Subroutine (29 LOWE TORQ)LTOR: This subroutine defines the points $
$                               of the lower torque tubes.          $
$-----$
$
$(29 LOWE TORQ)LTOR
$
!X1  = 0.
!X2  = 0.0285
!X3  = 0.0328
!X4  = 0.1560
!X5  = 0.1602
!X6  = 0.2905
!X7  = 0.2947
!X8  = 0.4555
!X9  = 0.4597
!X10 = 0.4883
!X11 = 0.
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
$
!X1A = 0.      +(0.0285/"EX1")
!X2A = 0.0285+(0.0043/"EX2")
!X3A = 0.0328+(0.1232/"EX3")
!X4A = 0.1560+(0.0042/"EX4")
!X5A = 0.1602+(0.1345/"EX5")
!X6A = 0.2905+(0.0042/"EX6")
!X7A = 0.2947+(0.1608/"EX7")
!X8A = 0.4555+(0.0042/"EX8")
!X9A = 0.4597+(0.0286/"EX9")
!X11A= 0.      +(0.0285/"EX1")
!X12A= 0.0285+(0.0043/"EX2")
!X13A= 0.0328+(0.1232/"EX3")
!X14A= 0.1560+(0.0042/"EX4")
!X15A= 0.1602+(0.1345/"EX5")
!X16A= 0.2905+(0.0042/"EX6")
!X17A= 0.2947+(0.1608/"EX7")
!X18A= 0.4555+(0.0042/"EX8")
!X19A= 0.4597+(0.0286/"EX9")
$
!Y1  = 0.0326

```

```

!Y2 = 0.0326
!Y3 = 0.0326
!Y4 = 0.0326
!Y5 = 0.0326
!Y6 = 0.0326
!Y7 = 0.0326
!Y8 = 0.0326
!Y9 = 0.0326
!Y10 = 0.0326
!Y11 = 0.0347
!Y12 = 0.0347
!Y13 = 0.0347
!Y14 = 0.0347
!Y15 = 0.0347
!Y16 = 0.0347
!Y17 = 0.0347
!Y18 = 0.0347
!Y19 = 0.0347
!Y20 = 0.0347
$
!Y1B = 0.0326+(0.0021/"EY4")
!Y2B = 0.0326+(0.0021/"EY4")
!Y3B = 0.0326+(0.0021/"EY4")
!Y4B = 0.0326+(0.0021/"EY4")
!Y5B = 0.0326+(0.0021/"EY4")
!Y6B = 0.0326+(0.0021/"EY4")
!Y7B = 0.0326+(0.0021/"EY4")
!Y8B = 0.0326+(0.0021/"EY4")
!Y9B = 0.0326+(0.0021/"EY4")
!Y10B= 0.0326+(0.0021/"EY4")
$
!Z1 = 0.
!Z2 = 0.
!Z3 = 0.
!Z4 = 0.
!Z5 = 0.
!Z6 = 0.
!Z7 = 0.
!Z8 = 0.
!Z9 = 0.
!Z10 = 0.
!Z11 = 0.
!Z12 = 0.
!Z13 = 0.
!Z14 = 0.
!Z15 = 0.
!Z16 = 0.
!Z17 = 0.
!Z18 = 0.
!Z19 = 0.
!Z20 = 0.
$
*RETURN
*LTOR
$
$-----$
$ Subroutine (29 AIRG SECT)AIRG: This subroutine defines the points $
$ of the airgap existing between the $
$ torque tubes, the edge channels... $

```

```

$-----$
$
*(29 AIRG SECT)AIRG
$
!X31 = 0.
!X32 = 0.0285
!X33 = 0.0328
!X34 = 0.1560
!X35 = 0.1602
!X36 = 0.2905
!X37 = 0.2947
!X38 = 0.4555
!X39 = 0.4597
!X40 = 0.4883
!X41 = 0.
!X42 = 0.0285
!X43 = 0.0328
!X44 = 0.1560
!X45 = 0.1602
!X46 = 0.2905
!X47 = 0.2947
!X48 = 0.4555
!X49 = 0.4597
!X50 = 0.4883
$
!X31A= 0.      +(0.0285/"EX1")
!X32A= 0.0285+(0.0043/"EX2")
!X33A= 0.0328+(0.1232/"EX3")
!X34A= 0.1560+(0.0042/"EX4")
!X35A= 0.1602+(0.1345/"EX5")
!X36A= 0.2905+(0.0042/"EX6")
!X37A= 0.2947+(0.1608/"EX7")
!X38A= 0.4555+(0.0042/"EX8")
!X39A= 0.4597+(0.0286/"EX9")
!X41A= 0.      +(0.0285/"EX1")
!X42A= 0.0285+(0.0043/"EX2")
!X43A= 0.0328+(0.1232/"EX3")
!X44A= 0.1560+(0.0042/"EX4")
!X45A= 0.1602+(0.1345/"EX5")
!X46A= 0.2905+(0.0042/"EX6")
!X47A= 0.2947+(0.1608/"EX7")
!X48A= 0.4555+(0.0042/"EX8")
!X49A= 0.4597+(0.0286/"EX9")
$
!Y31 = 0.0347
!Y32 = 0.0347
!Y33 = 0.0347
!Y34 = 0.0347
!Y35 = 0.0347
!Y36 = 0.0347
!Y37 = 0.0347
!Y38 = 0.0347
!Y39 = 0.0347
!Y40 = 0.0347
!Y41 = 0.0899
!Y42 = 0.0871
!Y43 = 0.0867
!Y44 = 0.0747
!Y45 = 0.0743

```

```

!Y46 = 0.0616
!Y47 = 0.0612
!Y48 = 0.0456
!Y49 = 0.0452
!Y50 = 0.0424
$
!Y42A= (0.0899-(0.0973*"X42A"))
!Y43A= (0.0899-(0.0973*"X43A"))
!Y44A= (0.0899-(0.0973*"X44A"))
!Y45A= (0.0899-(0.0973*"X45A"))
!Y46A= (0.0899-(0.0973*"X46A"))
!Y47A= (0.0899-(0.0973*"X47A"))
!Y48A= (0.0899-(0.0973*"X48A"))
!Y49A= (0.0899-(0.0973*"X49A"))
$
!Y31B= 0.0347+(0.0552/"EY5")
!Y32B= 0.0347+(0.0524/"EY5")
!Y33B= 0.0347+(0.0520/"EY5")
!Y34B= 0.0347+(0.0400/"EY5")
!Y35B= 0.0347+(0.0396/"EY5")
!Y36B= 0.0347+(0.0269/"EY5")
!Y37B= 0.0347+(0.0265/"EY5")
!Y38B= 0.0347+(0.0109/"EY5")
!Y39B= 0.0347+(0.0105/"EY5")
!Y40B= 0.0347+(0.0077/"EY5")
$
!Y32C= 0.0347+(((0.0899-(0.0973*"X32A"))-0.0347)/"EY5")
!Y33C= 0.0347+(((0.0899-(0.0973*"X33A"))-0.0347)/"EY5")
!Y34C= 0.0347+(((0.0899-(0.0973*"X34A"))-0.0347)/"EY5")
!Y35C= 0.0347+(((0.0899-(0.0973*"X35A"))-0.0347)/"EY5")
!Y36C= 0.0347+(((0.0899-(0.0973*"X36A"))-0.0347)/"EY5")
!Y37C= 0.0347+(((0.0899-(0.0973*"X37A"))-0.0347)/"EY5")
!Y38C= 0.0347+(((0.0899-(0.0973*"X38A"))-0.0347)/"EY5")
!Y39C= 0.0347+(((0.0899-(0.0973*"X39A"))-0.0347)/"EY5")
$
!Z31 = 0.
!Z32 = 0.
!Z33 = 0.
!Z34 = 0.
!Z35 = 0.
!Z36 = 0.
!Z37 = 0.
!Z38 = 0.
!Z39 = 0.
!Z40 = 0.
!Z41 = 0.
!Z42 = 0.
!Z43 = 0.
!Z44 = 0.
!Z45 = 0.
!Z46 = 0.
!Z47 = 0.
!Z48 = 0.
!Z49 = 0.
!Z50 = 0.
$
*RETURN
*AIRG
$

```

```

$-----$
$ Subroutine (29 UPPE TORQ)UTOR: This subroutine defines the points $
$                               of the upper torque tube to make $
$                               the nodes.                         $
$-----$
$
*(29 UPPE TORQ)UTOR
$
!X41 = 0.
!X42 = 0.0285
!X43 = 0.0328
!X44 = 0.1560
!X45 = 0.1602
!X46 = 0.2905
!X47 = 0.2947
!X48 = 0.4555
!X49 = 0.4597
!X50 = 0.4883
!X51 = 0.
!X52 = 0.0285
!X53 = 0.0328
!X54 = 0.1560
!X55 = 0.1602
!X56 = 0.2905
!X57 = 0.2947
!X58 = 0.4555
!X59 = 0.4597
!X60 = 0.4883
$
!X41A= 0.      +(0.0285/"EX1")
!X42A= 0.0285+(0.0043/"EX2")
!X43A= 0.0328+(0.1232/"EX3")
!X44A= 0.1560+(0.0042/"EX4")
!X45A= 0.1602+(0.1345/"EX5")
!X46A= 0.2905+(0.0042/"EX6")
!X47A= 0.2947+(0.1608/"EX7")
!X48A= 0.4555+(0.0042/"EX8")
!X49A= 0.4597+(0.0286/"EX9")
!X51A= 0.      +(0.0285/"EX1")
!X52A= 0.0285+(0.0043/"EX2")
!X53A= 0.0328+(0.1232/"EX3")
!X54A= 0.1560+(0.0042/"EX4")
!X55A= 0.1602+(0.1345/"EX5")
!X56A= 0.2905+(0.0042/"EX6")
!X57A= 0.2947+(0.1608/"EX7")
!X58A= 0.4555+(0.0042/"EX8")
!X59A= 0.4597+(0.0286/"EX9")
$
!Y41 = 0.0899
!Y42 = 0.0871
!Y43 = 0.0867
!Y44 = 0.0747
!Y45 = 0.0743
!Y46 = 0.0616
!Y47 = 0.0612
!Y48 = 0.0456
!Y49 = 0.0452
!Y50 = 0.0424
!Y51 = 0.0920

```



```

!Y52 = 0.0892
!Y53 = 0.0888
!Y54 = 0.0768
!Y55 = 0.0764
!Y56 = 0.0637
!Y57 = 0.0633
!Y58 = 0.0477
!Y59 = 0.0473
!Y60 = 0.0445
$
!Y52A= (0.0920-(0.0973*"X52A"))
!Y53A= (0.0920-(0.0973*"X53A"))
!Y54A= (0.0920-(0.0973*"X54A"))
!Y55A= (0.0920-(0.0973*"X55A"))
!Y56A= (0.0920-(0.0973*"X56A"))
!Y57A= (0.0920-(0.0973*"X57A"))
!Y58A= (0.0920-(0.0973*"X58A"))
!Y59A= (0.0920-(0.0973*"X59A"))
$
!Y41B= 0.0899+(0.0021/"EY6")
!Y42B= 0.0871+(0.0021/"EY6")
!Y43B= 0.0867+(0.0021/"EY6")
!Y44B= 0.0747+(0.0021/"EY6")
!Y45B= 0.0743+(0.0021/"EY6")
!Y46B= 0.0616+(0.0021/"EY6")
!Y47B= 0.0612+(0.0021/"EY6")
!Y48B= 0.0456+(0.0021/"EY6")
!Y49B= 0.0452+(0.0021/"EY6")
!Y50B= 0.0424+(0.0021/"EY6")
$
!Y42C= (0.0899-(0.0973*"X42A"))+(0.0021/"EY6")
!Y43C= (0.0899-(0.0973*"X43A"))+(0.0021/"EY6")
!Y44C= (0.0899-(0.0973*"X44A"))+(0.0021/"EY6")
!Y45C= (0.0899-(0.0973*"X45A"))+(0.0021/"EY6")
!Y46C= (0.0899-(0.0973*"X46A"))+(0.0021/"EY6")
!Y47C= (0.0899-(0.0973*"X47A"))+(0.0021/"EY6")
!Y48C= (0.0899-(0.0973*"X48A"))+(0.0021/"EY6")
!Y49C= (0.0899-(0.0973*"X49A"))+(0.0021/"EY6")
$
!Z41 = 0.
!Z42 = 0.
!Z43 = 0.
!Z44 = 0.
!Z45 = 0.
!Z46 = 0.
!Z47 = 0.
!Z48 = 0.
!Z49 = 0.
!Z50 = 0.
!Z51 = 0.
!Z52 = 0.
!Z53 = 0.
!Z54 = 0.
!Z55 = 0.
!Z56 = 0.
!Z57 = 0.
!Z58 = 0.
!Z59 = 0.
!Z60 = 0.

```

```

$
*RETURN
*UTOR
$
$-----$
$ Subroutine (29 UPPE SKIN)USKI: This subroutine defines the points $
$                               of the upper skin to make           $
$                               the nodes.                          $
$-----$
$
*(29 UPPE SKIN)USKI
$
!X41 = 0.
!X42 = 0.0285
!X43 = 0.0328
!X44 = 0.1560
!X45 = 0.1602
!X46 = 0.2905
!X47 = 0.2947
!X48 = 0.4555
!X49 = 0.4597
!X50 = 0.4883
!X51 = 0.
!X52 = 0.0285
!X53 = 0.0328
!X54 = 0.1560
!X55 = 0.1602
!X56 = 0.2905
!X57 = 0.2947
!X58 = 0.4555
!X59 = 0.4597
!X60 = 0.4883
$
!X41A= 0.      +(0.0285/"EX1")
!X42A= 0.0285+(0.0043/"EX2")
!X43A= 0.0328+(0.1232/"EX3")
!X44A= 0.1560+(0.0042/"EX4")
!X45A= 0.1602+(0.1345/"EX5")
!X46A= 0.2905+(0.0042/"EX6")
!X47A= 0.2947+(0.1608/"EX7")
!X48A= 0.4555+(0.0042/"EX8")
!X49A= 0.4597+(0.0286/"EX9")
!X51A= 0.      +(0.0285/"EX1")
!X52A= 0.0285+(0.0043/"EX2")
!X53A= 0.0328+(0.1232/"EX3")
!X54A= 0.1560+(0.0042/"EX4")
!X55A= 0.1602+(0.1345/"EX5")
!X56A= 0.2905+(0.0042/"EX6")
!X57A= 0.2947+(0.1608/"EX7")
!X58A= 0.4555+(0.0042/"EX8")
!X59A= 0.4597+(0.0286/"EX9")
$
!Y41 = 0.0920
!Y42 = 0.0892
!Y43 = 0.0888
!Y44 = 0.0768
!Y45 = 0.0764
!Y46 = 0.0637
!Y47 = 0.0633

```

```

!Y48 = 0.0477
!Y49 = 0.0473
!Y50 = 0.0445
!Y51 = 0.0949
!Y52 = 0.0921
!Y53 = 0.0917
!Y54 = 0.0797
!Y55 = 0.0793
!Y56 = 0.0666
!Y57 = 0.0662
!Y58 = 0.0506
!Y59 = 0.0502
!Y60 = 0.0474
$
!Y52A= (0.0949-(0.0973*"X52A"))
!Y53A= (0.0949-(0.0973*"X53A"))
!Y54A= (0.0949-(0.0973*"X54A"))
!Y55A= (0.0949-(0.0973*"X55A"))
!Y56A= (0.0949-(0.0973*"X56A"))
!Y57A= (0.0949-(0.0973*"X57A"))
!Y58A= (0.0949-(0.0973*"X58A"))
!Y59A= (0.0949-(0.0973*"X59A"))
$
!Y41B= 0.0920+(0.0029/"EY7")
!Y42B= 0.0892+(0.0029/"EY7")
!Y43B= 0.0888+(0.0029/"EY7")
!Y44B= 0.0768+(0.0029/"EY7")
!Y45B= 0.0764+(0.0029/"EY7")
!Y46B= 0.0637+(0.0029/"EY7")
!Y47B= 0.0633+(0.0029/"EY7")
!Y48B= 0.0477+(0.0029/"EY7")
!Y49B= 0.0473+(0.0029/"EY7")
!Y50B= 0.0445+(0.0029/"EY7")
$
!Y42C= (0.0920-(0.0973*"X42A"))+(0.0029/"EY7")
!Y43C= (0.0920-(0.0973*"X43A"))+(0.0029/"EY7")
!Y44C= (0.0920-(0.0973*"X44A"))+(0.0029/"EY7")
!Y45C= (0.0920-(0.0973*"X45A"))+(0.0029/"EY7")
!Y46C= (0.0920-(0.0973*"X46A"))+(0.0029/"EY7")
!Y47C= (0.0920-(0.0973*"X47A"))+(0.0029/"EY7")
!Y48C= (0.0920-(0.0973*"X48A"))+(0.0029/"EY7")
!Y49C= (0.0920-(0.0973*"X49A"))+(0.0029/"EY7")
$
!Z41 = 0.
!Z42 = 0.
!Z43 = 0.
!Z44 = 0.
!Z45 = 0.
!Z46 = 0.
!Z47 = 0.
!Z48 = 0.
!Z49 = 0.
!Z50 = 0.
!Z51 = 0.
!Z52 = 0.
!Z53 = 0.
!Z54 = 0.
!Z55 = 0.
!Z56 = 0.

```

```

!Z57 = 0.
!Z58 = 0.
!Z59 = 0.
!Z60 = 0.
$
*RETURN
*USKI
$
$-----$
$ Subroutine (29 UPPE HONY)UHON: This subroutine defines the points $
$                               of the upper honeycomb structure $
$                               to make the nodes. $
$-----$
$
$(29 UPPE HONY)UHON
$
!X51 = 0.
!X52 = 0.0285
!X53 = 0.0328
!X54 = 0.1560
!X55 = 0.1602
!X56 = 0.2905
!X57 = 0.2947
!X58 = 0.4555
!X59 = 0.4597
!X60 = 0.4883
!X61 = 0.
!X62 = 0.0285
!X63 = 0.0328
!X64 = 0.1560
!X65 = 0.1602
!X66 = 0.2905
!X67 = 0.2947
!X68 = 0.4555
!X69 = 0.4597
!X70 = 0.4883
$
!X51A= 0.      +(0.0285/"EX1")
!X52A= 0.0285+(0.0043/"EX2")
!X53A= 0.0328+(0.1232/"EX3")
!X54A= 0.1560+(0.0042/"EX4")
!X55A= 0.1602+(0.1345/"EX5")
!X56A= 0.2905+(0.0042/"EX6")
!X57A= 0.2947+(0.1608/"EX7")
!X58A= 0.4555+(0.0042/"EX8")
!X59A= 0.4597+(0.0286/"EX9")
!X61A= 0.      +(0.0285/"EX1")
!X62A= 0.0285+(0.0043/"EX2")
!X63A= 0.0328+(0.1232/"EX3")
!X64A= 0.1560+(0.0042/"EX4")
!X65A= 0.1602+(0.1345/"EX5")
!X66A= 0.2905+(0.0042/"EX6")
!X67A= 0.2947+(0.1608/"EX7")
!X68A= 0.4555+(0.0042/"EX8")
!X69A= 0.4597+(0.0286/"EX9")
$
!Y51 = 0.0949
!Y52 = 0.0921
!Y53 = 0.0917

```

```

!Y54 = 0.0797
!Y55 = 0.0793
!Y56 = 0.0666
!Y57 = 0.0662
!Y58 = 0.0506
!Y59 = 0.0502
!Y60 = 0.0474
!Y61 = 0.1203
!Y62 = 0.1175
!Y63 = 0.1171
!Y64 = 0.1051
!Y65 = 0.1047
!Y66 = 0.0920
!Y67 = 0.0916
!Y68 = 0.0760
!Y69 = 0.0756
!Y70 = 0.0728
$
!Y62A= (0.1203-(0.0973*"X62A"))
!Y63A= (0.1203-(0.0973*"X63A"))
!Y64A= (0.1203-(0.0973*"X64A"))
!Y65A= (0.1203-(0.0973*"X65A"))
!Y66A= (0.1203-(0.0973*"X66A"))
!Y67A= (0.1203-(0.0973*"X67A"))
!Y68A= (0.1203-(0.0973*"X68A"))
!Y69A= (0.1203-(0.0973*"X69A"))
$
!Y51B= 0.0949+(0.0254/"EY8")
!Y52B= 0.0921+(0.0254/"EY8")
!Y53B= 0.0917+(0.0254/"EY8")
!Y54B= 0.0797+(0.0254/"EY8")
!Y55B= 0.0793+(0.0254/"EY8")
!Y56B= 0.0666+(0.0254/"EY8")
!Y57B= 0.0662+(0.0254/"EY8")
!Y58B= 0.0506+(0.0254/"EY8")
!Y59B= 0.0502+(0.0254/"EY8")
!Y60B= 0.0474+(0.0254/"EY8")
$
!Y52C= (0.0949-(0.0973*"X52A"))+(0.0254/"EY8")
!Y53C= (0.0949-(0.0973*"X53A"))+(0.0254/"EY8")
!Y54C= (0.0949-(0.0973*"X54A"))+(0.0254/"EY8")
!Y55C= (0.0949-(0.0973*"X55A"))+(0.0254/"EY8")
!Y56C= (0.0949-(0.0973*"X56A"))+(0.0254/"EY8")
!Y57C= (0.0949-(0.0973*"X57A"))+(0.0254/"EY8")
!Y58C= (0.0949-(0.0973*"X58A"))+(0.0254/"EY8")
!Y59C= (0.0949-(0.0973*"X59A"))+(0.0254/"EY8")
$
!Z51 = 0.
!Z52 = 0.
!Z53 = 0.
!Z54 = 0.
!Z55 = 0.
!Z56 = 0.
!Z57 = 0.
!Z58 = 0.
!Z59 = 0.
!Z60 = 0.
!Z61 = 0.
!Z62 = 0.

```

```

!Z63 = 0.
!Z64 = 0.
!Z65 = 0.
!Z66 = 0.
!Z67 = 0.
!Z68 = 0.
!Z69 = 0.
!Z70 = 0.
$
*RETURN
*UHON
$
$-----$
$ Subroutine (29 UPPE FACE)UFAC: This subroutine defines the points $
$                               of the upper outer facesheet to $
$                               make the nodes. $
$-----$
$
$(29 UPPE FACE)UFAC
$
!X61 = 0.
!X62 = 0.0285
!X63 = 0.0328
!X64 = 0.1560
!X65 = 0.1602
!X66 = 0.2905
!X67 = 0.2947
!X68 = 0.4555
!X69 = 0.4597
!X70 = 0.4883
!X71 = 0.
!X72 = 0.0285
!X73 = 0.0328
!X74 = 0.1560
!X75 = 0.1602
!X76 = 0.2905
!X77 = 0.2947
!X78 = 0.4555
!X79 = 0.4597
!X80 = 0.4883
$
!X61A= 0.      +(0.0285/"EX1")
!X62A= 0.0285+(0.0043/"EX2")
!X63A= 0.0328+(0.1232/"EX3")
!X64A= 0.1560+(0.0042/"EX4")
!X65A= 0.1602+(0.1345/"EX5")
!X66A= 0.2905+(0.0042/"EX6")
!X67A= 0.2947+(0.1608/"EX7")
!X68A= 0.4555+(0.0042/"EX8")
!X69A= 0.4597+(0.0286/"EX9")
!X71A= 0.      +(0.0285/"EX1")
!X72A= 0.0285+(0.0043/"EX2")
!X73A= 0.0328+(0.1232/"EX3")
!X74A= 0.1560+(0.0042/"EX4")
!X75A= 0.1602+(0.1345/"EX5")
!X76A= 0.2905+(0.0042/"EX6")
!X77A= 0.2947+(0.1608/"EX7")
!X78A= 0.4555+(0.0042/"EX8")
!X79A= 0.4597+(0.0286/"EX9")

```

```

$
!Y61 = 0.1203
!Y62 = 0.1175
!Y63 = 0.1171
!Y64 = 0.1051
!Y65 = 0.1047
!Y66 = 0.0920
!Y67 = 0.0916
!Y68 = 0.0760
!Y69 = 0.0756
!Y70 = 0.0728
!Y71 = 0.1235
!Y72 = 0.1207
!Y73 = 0.1203
!Y74 = 0.1083
!Y75 = 0.1079
!Y76 = 0.0952
!Y77 = 0.0948
!Y78 = 0.0792
!Y79 = 0.0788
!Y80 = 0.0760
$
!Y72A= (0.1235-(0.0973*"X72A"))
!Y73A= (0.1235-(0.0973*"X73A"))
!Y74A= (0.1235-(0.0973*"X74A"))
!Y75A= (0.1235-(0.0973*"X75A"))
!Y76A= (0.1235-(0.0973*"X76A"))
!Y77A= (0.1235-(0.0973*"X77A"))
!Y78A= (0.1235-(0.0973*"X78A"))
!Y79A= (0.1235-(0.0973*"X79A"))
$
!Y61B= 0.1203+(0.0032/"EY9")
!Y62B= 0.1175+(0.0032/"EY9")
!Y63B= 0.1171+(0.0032/"EY9")
!Y64B= 0.1051+(0.0032/"EY9")
!Y65B= 0.1047+(0.0032/"EY9")
!Y66B= 0.0920+(0.0032/"EY9")
!Y67B= 0.0916+(0.0032/"EY9")
!Y68B= 0.0760+(0.0032/"EY9")
!Y69B= 0.0756+(0.0032/"EY9")
!Y70B= 0.0728+(0.0032/"EY9")
$
!Y62C= (0.1203-(0.0973*"X62A"))+(0.0032/"EY9")
!Y63C= (0.1203-(0.0973*"X63A"))+(0.0032/"EY9")
!Y64C= (0.1203-(0.0973*"X64A"))+(0.0032/"EY9")
!Y65C= (0.1203-(0.0973*"X65A"))+(0.0032/"EY9")
!Y66C= (0.1203-(0.0973*"X66A"))+(0.0032/"EY9")
!Y67C= (0.1203-(0.0973*"X67A"))+(0.0032/"EY9")
!Y68C= (0.1203-(0.0973*"X68A"))+(0.0032/"EY9")
!Y69C= (0.1203-(0.0973*"X69A"))+(0.0032/"EY9")
$
!Z61 = 0.
!Z62 = 0.
!Z63 = 0.
!Z64 = 0.
!Z65 = 0.
!Z66 = 0.
!Z67 = 0.
!Z68 = 0.

```

```

!Z69 = 0.
!Z70 = 0.
!Z71 = 0.
!Z72 = 0.
!Z73 = 0.
!Z74 = 0.
!Z75 = 0.
!Z76 = 0.
!Z77 = 0.
!Z78 = 0.
!Z79 = 0.
!Z80 = 0.
$
*RETURN
*UFAC
$
$-----$
$ Subroutine (29 UPPE INSU) INSU: This subroutine defines the points$
$                               of the insulation located on the $
$                               top surface of the upper outer $
$                               facesheet. $
$-----$
$
*(29 UPPE INSU) INSU
$
!X1  = 0.0000
!X2  = 0.0285
!X3  = 0.0328
!X4  = 0.1560
!X5  = 0.1602
!X6  = 0.2905
!X7  = 0.2947
!X8  = 0.4555
!X9  = 0.4597
!X10 = 0.4883
!X11 = 0.0000
!X12 = 0.0285
!X13 = 0.0328
!X14 = 0.1560
!X15 = 0.1602
!X16 = 0.2905
!X17 = 0.2947
!X18 = 0.4555
!X19 = 0.4597
!X20 = 0.4883
$
!X1A = 0.0000+(0.0285/"EX1")
!X2A = 0.0285+(0.0043/"EX2")
!X3A = 0.0328+(0.1232/"EX3")
!X4A = 0.1560+(0.0042/"EX4")
!X5A = 0.1602+(0.1345/"EX5")
!X6A = 0.2905+(0.0042/"EX6")
!X7A = 0.2947+(0.1608/"EX7")
!X8A = 0.4555+(0.0042/"EX8")
!X9A = 0.4597+(0.0286/"EX9")
!X11A = 0.0000+(0.0285/"EX1")
!X12A = 0.0285+(0.0043/"EX2")
!X13A = 0.0328+(0.1232/"EX3")
!X14A = 0.1560+(0.0042/"EX4")

```



```

!X15A = 0.1602+(0.1345/"EX5")
!X16A = 0.2905+(0.0042/"EX6")
!X17A = 0.2947+(0.1608/"EX7")
!X18A = 0.4555+(0.0042/"EX8")
!X19A = 0.4597+(0.0286/"EX9")
$
!Y1  = 0.1235
!Y2  = 0.1207
!Y3  = 0.1203
!Y4  = 0.1083
!Y5  = 0.1079
!Y6  = 0.0952
!Y7  = 0.0948
!Y8  = 0.0792
!Y9  = 0.0788
!Y10 = 0.0760
!Y11 = 0.1489
!Y12 = 0.1461
!Y13 = 0.1457
!Y14 = 0.1337
!Y15 = 0.1333
!Y16 = 0.1206
!Y17 = 0.1202
!Y18 = 0.1046
!Y19 = 0.1042
!Y20 = 0.1014
$
!Y12A = (0.1489-(0.0973*"X12A"))
!Y13A = (0.1489-(0.0973*"X13A"))
!Y14A = (0.1489-(0.0973*"X14A"))
!Y15A = (0.1489-(0.0973*"X15A"))
!Y16A = (0.1489-(0.0973*"X16A"))
!Y17A = (0.1489-(0.0973*"X17A"))
!Y18A = (0.1489-(0.0973*"X18A"))
!Y19A = (0.1489-(0.0973*"X19A"))
$
!Y1B  = 0.1235+(0.0254/"EY10")
!Y2B  = 0.1207+(0.0254/"EY10")
!Y3B  = 0.1203+(0.0254/"EY10")
!Y4B  = 0.1083+(0.0254/"EY10")
!Y5B  = 0.1079+(0.0254/"EY10")
!Y6B  = 0.0952+(0.0254/"EY10")
!Y7B  = 0.0948+(0.0254/"EY10")
!Y8B  = 0.0792+(0.0254/"EY10")
!Y9B  = 0.0788+(0.0254/"EY10")
!Y10B = 0.0760+(0.0254/"EY10")
$
!Y2C  = (0.1235-(0.0973*"X2A"))+(0.0254/"EY10")
!Y3C  = (0.1235-(0.0973*"X3A"))+(0.0254/"EY10")
!Y4C  = (0.1235-(0.0973*"X4A"))+(0.0254/"EY10")
!Y5C  = (0.1235-(0.0973*"X5A"))+(0.0254/"EY10")
!Y6C  = (0.1235-(0.0973*"X6A"))+(0.0254/"EY10")
!Y7C  = (0.1235-(0.0973*"X7A"))+(0.0254/"EY10")
!Y8C  = (0.1235-(0.0973*"X8A"))+(0.0254/"EY10")
!Y9C  = (0.1235-(0.0973*"X9A"))+(0.0254/"EY10")
$
!Z1  = 0.0000
!Z2  = 0.0000
!Z3  = 0.0000

```

```

!Z4   = 0.0000
!Z5   = 0.0000
!Z6   = 0.0000
!Z7   = 0.0000
!Z8   = 0.0000
!Z9   = 0.0000
!Z10  = 0.0000
!Z11  = 0.0000
!Z12  = 0.0000
!Z13  = 0.0000
!Z14  = 0.0000
!Z15  = 0.0000
!Z16  = 0.0000
!Z17  = 0.0000
!Z18  = 0.0000
!Z19  = 0.0000
!Z20  = 0.0000
$
*RETURN
*INSU
$
$-----$
$ Subroutine (29 NODE GENE)NODE: This subroutine creates the nodes $
$                               for the entire structure.           $
$-----$
$
$(29 NODE GENE)NODE
$
*XQT TAB
$
START "TOTN"
JLOC
FORMAT  = 1
$
*DCALL(29 LOWE FACE)
*DCALL(29 LFAC NODE)
*DCALL(29 LFAC FREE)
$
*DCALL(29 LOWE HONY)
*DCALL(29 LHON NODE)
*DCALL(29 LHON FREE)
$
*DCALL(29 LOWE SKIN)
*DCALL(29 LSKI NODE)
*DCALL(29 LSKI FREE)
$
*DCALL(29 LOWE TORQ)
*DCALL(29 LOTO NODE)
*DCALL(29 LTOR FREE)
$
*DCALL(29 AIRG SECT)
*DCALL(29 AIRG NODE)
*DCALL(29 AIRG FREE)
$
*DCALL(29 UPPE TORQ)
*DCALL(29 UPTO NODE)
*DCALL(29 TORU FREE)
$
*DCALL(29 UPPE SKIN)

```

```

*DCALL(29 USKI NODE)
*DCALL(29 USKI FREE)
$
*DCALL(29 UPPE HONY)
*DCALL(29 UHON NODE)
*DCALL(29 UHON FREE)
$
*DCALL(29 UPPE FACE)
*DCALL(29 UFAC NODE)
*DCALL(29 UFAC FREE)
$
*DCALL(29 UPPE INSU)
*DCALL(29 INSU NODE)
*DCALL(29 UINS FREE)
$
*RETURN
*NODE
$
$-----$
$ Subroutine (29 LFAC NODE)NOD1: This subroutine creates the nodes $
$                               for the lower outer facesheet. $
$-----$
$
*(29 LFAC NODE)NOD1
$
!P1 = 1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NX2" GT 2): *GOTO 10
$
"P1","X1","Y1","Z1","X2","Y2","Z2","NX1","XJ","NY1"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X3","Y3","Z3","X13","Y13","Z13","NY1","YJ"
$
"P3","X3A","Y3","Z3","X4","Y4","Z4","EX3","XJ","NY1"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X5","Y5","Z5","X15","Y15","Z15","NY1","YJ"
$
"P5","X5A","Y5","Z5","X6","Y6","Z6","EX5","XJ","NY1"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X7","Y7","Z7","X17","Y17","Z17","NY1","YJ"
$
"P7","X7A","Y7","Z7","X8","Y8","Z8","EX7","XJ","NY1"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X9","Y9","Z9","X19","Y19","Z19","NY1","YJ"
$
"P9","X9A","Y9","Z9","X10","Y10","Z10","EX9","XJ","NY1"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"

```

```

$
*GOTO 20
$
*LABEL 10
$
"P1","X1","Y1","Z1","X2","Y2","Z2","NX1","XJ","NY1"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X2A","Y2","Z2","X3","Y3","Z3","EX2","XJ","NY1"
"YJ","X12A","Y12","Z12","X13","Y13","Z13"
$
"P3","X3A","Y3","Z3","X4","Y4","Z4","EX3","XJ","NY1"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X4A","Y4","Z4","X5","Y5","Z5","EX4","XJ","NY1"
"YJ","X14A","Y14","Z14","X15","Y15","Z15"
$
"P5","X5A","Y5","Z5","X6","Y6","Z6","EX5","XJ","NY1"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X6A","Y6","Z6","X7","Y7","Z7","EX6","XJ","NY1"
"YJ","X16A","Y16","Z16","X17","Y17","Z17"
$
"P7","X7A","Y7","Z7","X8","Y8","Z8","EX7","XJ","NY1"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X8A","Y8","Z8","X9","Y9","Z9","EX8","XJ","NY1"
"YJ","X18A","Y18","Z18","X19","Y19","Z19"
$
"P9","X9A","Y9","Z9","X10","Y10","Z10","EX9","XJ","NY1"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*LABEL 20
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD1
$
$-----$
$ Subroutine (29 LHON NODE)NOD2: This subroutine creates the nodes $
$                               for the lower honeycomb structure. $
$-----$
$
*(29 LHON NODE)NOD2
$
!P1 = ("YJ"*"NY1")+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"

```

```

!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY2" GT 2): *GOTO 50
*IF("NX2" GT 2): *GOTO 30
$
"P1", "X21", "Y21", "Z21", "X22", "Y22", "Z22", "NX1", "XJ"
$
"P2", "X23", "Y23", "Z23"
$
"P3", "X23A", "Y23", "Z23", "X24", "Y24", "Z24", "EX3", "XJ"
$
"P4", "X25", "Y25", "Z25"
$
"P5", "X25A", "Y25", "Z25", "X26", "Y26", "Z26", "EX5", "XJ"
$
"P6", "X27", "Y27", "Z27"
$
"P7", "X27A", "Y27", "Z27", "X28", "Y28", "Z28", "EX7", "XJ"
$
"P8", "X29", "Y29", "Z29"
$
"P9", "X29A", "Y29", "Z29", "X30", "Y30", "Z30", "EX9", "XJ"
$
*GOTO 40
$
*LABEL 30
$
"P1", "X21", "Y21", "Z21", "X22", "Y22", "Z22", "NX1", "XJ"
$
"P2", "X13", "Y13B", "Z13", "X23", "Y23", "Z13", "EX2", "XJ"
$
"P3", "X23A", "Y23", "Z23", "X24", "Y24", "Z24", "EX3", "XJ"
$
"P4", "X15", "Y15B", "Z15", "X25", "Y25", "Z25", "EX4", "XJ"
$
"P5", "X25A", "Y25", "Z25", "X26", "Y26", "Z26", "EX5", "XJ"
$
"P6", "X17", "Y17B", "Z17", "X27", "Y27", "Z27", "EX6", "XJ"
$
"P7", "X27A", "Y27", "Z27", "X28", "Y28", "Z28", "EX7", "XJ"
$
"P8", "X19", "Y19B", "Z19", "X29", "Y29", "Z29", "EX8", "XJ"
$
"P9", "X29A", "Y29", "Z29", "X30", "Y30", "Z30", "EX9", "XJ"
$
*GOTO 40
$
*LABEL 50
$
*IF("NX2" GT 2): *GOTO 60
$
"P1", "X11", "Y11B", "Z11", "X12", "Y12B", "Z12", "NX1", "XJ", "EY2"
"YJ", "X21", "Y21", "Z21", "X22", "Y22", "Z22"
$
"P2", "X13", "Y13B", "Z13", "X23", "Y23", "Z13", "EY2", "YJ"
$

```

```

"P3","X13A","Y13B","Z13","X14","Y14B","Z14","EX3","XJ","EY2"
"YJ","X23A","Y23","Z23","X24","Y24","Z24"
$
"P4","X15","Y15B","Z15","X25","Y25","Z25","EY2","YJ"
$
"P5","X15A","Y15B","Z15","X16","Y16B","Z16","EX5","XJ","EY2"
"YJ","X25A","Y25","Z25","X26","Y26","Z26"
$
"P6","X17","Y17B","Z17","X27","Y27","Z27","EY2","YJ"
$
"P7","X17A","Y17B","Z17","X18","Y18B","Z18","EX7","XJ","EY2"
"YJ","X27A","Y27","Z27","X28","Y28","Z28"
$
"P8","X19","Y19B","Z19","X29","Y29","Z29","EY2","YJ"
$
"P9","X19A","Y19B","Z19","X20","Y20B","Z20","EX9","XJ","EY2"
"YJ","X29A","Y29","Z29","X30","Y30","Z30"
$
*GOTO 40
$
*LABEL 60
$
"P1","X11","Y11B","Z11","X12","Y12B","Z12","NX1","XJ","EY2"
"YJ","X21","Y21","Z21","X22","Y22","Z22"
$
"P2","X12A","Y12B","Z12","X13","Y13B","Z13","EX2","XJ","EY2"
"YJ","X22A","Y22","Z22","X23","Y23","Z23"
$
"P3","X13A","Y13B","Z13","X14","Y14B","Z14","EX3","XJ","EY2"
"YJ","X23A","Y23","Z23","X24","Y24","Z24"
$
"P4","X14A","Y14B","Z14","X15","Y15B","Z15","EX4","XJ","EY2"
"YJ","X24A","Y24","Z24","X25","Y25","Z25"
$
"P5","X15A","Y15B","Z15","X16","Y16B","Z16","EX5","XJ","EY2"
"YJ","X25A","Y25","Z25","X26","Y26","Z26"
$
"P6","X16A","Y16B","Z16","X17","Y17B","Z17","EX6","XJ","EY2"
"YJ","X26A","Y26","Z26","X27","Y27","Z27"
$
"P7","X17A","Y17B","Z17","X18","Y18B","Z18","EX7","XJ","EY2"
"YJ","X27A","Y27","Z27","X28","Y28","Z28"
$
"P8","X18A","Y18B","Z18","X19","Y19B","Z19","EX8","XJ","EY2"
"YJ","X28A","Y28","Z28","X29","Y29","Z29"
$
"P9","X19A","Y19B","Z19","X20","Y20B","Z20","EX9","XJ","EY2"
"YJ","X29A","Y29","Z29","X30","Y30","Z30"
$
*LABEL 40
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()

```

```

!P9 = FREE ()
$
*RETURN
*NOD2
$
$-----$
$ Subroutine (29 LSKI NODE)NOD3: This subroutine creates the nodes $
$                               for the lower skin. $
$-----$
$
*(29 LSKI NODE)NOD3
$
!P1 = ("YJ"*("NY1"+"EY2"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY3" GT 2): *GOTO 90
*IF("NX2" GT 2): *GOTO 70
$
"P1", "X31", "Y31", "Z31", "X32", "Y32", "Z32", "NX1", "XJ"
$
"P2", "X33", "Y33", "Z33"
$
"P3", "X33A", "Y33", "Z33", "X34", "Y34", "Z34", "EX3", "XJ"
$
"P4", "X35", "Y35", "Z35"
$
"P5", "X35A", "Y35", "Z35", "X36", "Y36", "Z36", "EX5", "XJ"
$
"P6", "X37", "Y37", "Z37"
$
"P7", "X37A", "Y37", "Z37", "X38", "Y38", "Z38", "EX7", "XJ"
$
"P8", "X39", "Y39", "Z39"
$
"P9", "X39A", "Y39", "Z39", "X40", "Y40", "Z40", "EX9", "XJ"
$
*GOTO 80
$
*LABEL 70
$
"P1", "X31", "Y31", "Z31", "X32", "Y32", "Z32", "NX1", "XJ"
$
"P2", "X23", "Y23B", "Z23", "X33", "Y33", "Z33", "EX2", "XJ"
$
"P3", "X33A", "Y33", "Z33", "X34", "Y34", "Z34", "EX3", "XJ"
$
"P4", "X25", "Y25B", "Z25", "X35", "Y35", "Z35", "EX4", "XJ"
$
"P5", "X35A", "Y35", "Z35", "X36", "Y36", "Z36", "EX5", "XJ"
$
"P6", "X27", "Y27B", "Z27", "X37", "Y37", "Z37", "EX6", "XJ"
$

```

```

"P7","X37A","Y37","Z37","X38","Y38","Z38","EX7","XJ"
$
"P8","X29","Y29B","Z29","X39","Y39","Z39","EX8","XJ"
$
"P9","X39A","Y39","Z39","X40","Y40","Z40","EX9","XJ"
$
*GOTO 80
$
*LABEL 90
$
*IF("NX2" GT 2): *GOTO 100
$
"P1","X21","Y21B","Z21","X22","Y22B","Z22","NX1","XJ","EY3"
"YJ","X31","Y31","Z31","X32","Y32","Z32"
$
"P2","X23","Y23B","Z23","X33","Y33","Z33","EY3","YJ"
$
"P3","X23A","Y23B","Z23","X24","Y24B","Z24","EX3","XJ","EY3"
"YJ","X33A","Y33","Z33","X34","Y34","Z34"
$
"P4","X25","Y25B","Z25","X35","Y35","Z35","EY3","YJ"
$
"P5","X25A","Y25B","Z25","X26","Y26B","Z26","EX5","XJ","EY3"
"YJ","X35A","Y35","Z35","X36","Y36","Z36"
$
"P6","X27","Y27B","Z27","X37","Y37","Z37","EY3","YJ"
$
"P7","X27A","Y27B","Z27","X28","Y28B","Z28","EX7","XJ","EY3"
"YJ","X37A","Y37","Z37","X38","Y38","Z38"
$
"P8","X29","Y29B","Z29","X39","Y39","Z39","EY3","YJ"
$
"P9","X29A","Y29B","Z29","X30","Y30B","Z30","EX9","XJ","EY3"
"YJ","X39A","Y39","Z39","X40","Y40","Z40"
$
*GOTO 80
$
*LABEL 100
$
"P1","X21","Y21B","Z21","X22","Y22B","Z22","NX1","XJ","EY3"
"YJ","X31","Y31","Z31","X32","Y32","Z32"
$
"P2","X22A","Y22B","Z22","X23","Y23B","Z23","EX2","XJ","EY3"
"YJ","X32A","Y32","Z32","X33","Y33","Z33"
$
"P3","X23A","Y23B","Z23","X24","Y24B","Z24","EX3","XJ","EY3"
"YJ","X33A","Y33","Z33","X34","Y34","Z34"
$
"P4","X24A","Y24B","Z24","X25","Y25B","Z25","EX4","XJ","EY3"
"YJ","X34A","Y34","Z34","X35","Y35","Z35"
$
"P5","X25A","Y25B","Z25","X26","Y26B","Z26","EX5","XJ","EY3"
"YJ","X35A","Y35","Z35","X36","Y36","Z36"
$
"P6","X26A","Y26B","Z26","X27","Y27B","Z27","EX6","XJ","EY3"
"YJ","X36A","Y36","Z36","X37","Y37","Z37"
$
"P7","X27A","Y27B","Z27","X28","Y28B","Z28","EX7","XJ","EY3"
"YJ","X37A","Y37","Z37","X38","Y38","Z38"

```



```

$
"P8","X28A","Y28B","Z28","X29","Y29B","Z29","EX8","XJ","EY3"
"YJ","X38A","Y38","Z38","X39","Y39","Z39"
$
"P9","X29A","Y29B","Z29","X30","Y30B","Z30","EX9","XJ","EY3"
"YJ","X39A","Y39","Z39","X40","Y40","Z40"
$
*LABEL 80
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD3
$
$-----$
$ Subroutine (29 LOTO NODE)LOTO: This subroutine creates the nodes $
$                               for the lower skin. $
$-----$
$
*(29 LOTO NODE)LOTO
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY4" GT 2): *GOTO 90
*IF("NX2" GT 2): *GOTO 70
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X13","Y13","Z13"
$
"P3","X13A","Y13","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X15","Y15","Z15"
$
"P5","X15A","Y15","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X17","Y17","Z17"
$
"P7","X17A","Y17","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X19","Y19","Z19"
$
"P9","X19A","Y19","Z19","X20","Y20","Z20","EX9","XJ"

```

```

$
*GOTO 80
$
*LABEL 70
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EX2","XJ"
$
"P3","X13A","Y13","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EX4","XJ"
$
"P5","X15A","Y15","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EX6","XJ"
$
"P7","X17A","Y17","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EX8","XJ"
$
"P9","X19A","Y19","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 80
$
*LABEL 90
$
*IF("NX2" GT 2): *GOTO 100
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY4"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EY4","YJ"
$
"P3","X3A","Y3B","Z3","X4","Y4B","Z4","EX3","XJ","EY4"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EY4","YJ"
$
"P5","X5A","Y5B","Z5","X6","Y6B","Z6","EX5","XJ","EY4"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EY4","YJ"
$
"P7","X7A","Y7B","Z7","X8","Y8B","Z8","EX7","XJ","EY4"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EY4","YJ"
$
"P9","X9A","Y9B","Z9","X10","Y10B","Z10","EX9","XJ","EY4"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*GOTO 80
$
*LABEL 100
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY4"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$

```

```

"P2","X2A","Y2B","Z2","X3","Y3B","Z3","EX2","XJ","EY4"
"YJ","X12A","Y12","Z12","X13","Y13","Z13"
$
"P3","X3A","Y3B","Z3","X4","Y4B","Z4","EX3","XJ","EY4"
"YJ","X13A","Y13","Z13","X14","Y14","Z14"
$
"P4","X4A","Y4B","Z4","X5","Y5B","Z5","EX4","XJ","EY4"
"YJ","X14A","Y14","Z14","X15","Y15","Z15"
$
"P5","X5A","Y5B","Z5","X6","Y6B","Z6","EX5","XJ","EY4"
"YJ","X15A","Y15","Z15","X16","Y16","Z16"
$
"P6","X6A","Y6B","Z6","X7","Y7B","Z7","EX6","XJ","EY4"
"YJ","X16A","Y16","Z16","X17","Y17","Z17"
$
"P7","X7A","Y7B","Z7","X8","Y8B","Z8","EX7","XJ","EY4"
"YJ","X17A","Y17","Z17","X18","Y18","Z18"
$
"P8","X8A","Y8B","Z8","X9","Y9B","Z9","EX8","XJ","EY4"
"YJ","X18A","Y18","Z18","X19","Y19","Z19"
$
"P9","X9A","Y9B","Z9","X10","Y10B","Z10","EX9","XJ","EY4"
"YJ","X19A","Y19","Z19","X20","Y20","Z20"
$
*LABEL 80
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*LOTO
$
$-----$
$ Subroutine (29 AIRG NODE)NOD4: This subroutine creates the nodes $
$                               for the mid section of the $
$                               structure. $
$-----$
$
*(29 AIRG NODE)NOD4
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY5" GT 2): *GOTO 130
*IF("NX2" GT 2): *GOTO 110

```

```

$
"P1","X41","Y41","Z41","X42","Y42","Z42","NX1","XJ"
$
"P2","X43","Y43","Z43"
$
"P3","X43A","Y43A","Z43","X44","Y44","Z44","EX3","XJ"
$
"P4","X45","Y45","Z45"
$
"P5","X45A","Y45A","Z45","X46","Y46","Z46","EX5","XJ"
$
"P6","X47","Y47","Z47"
$
"P7","X47A","Y47A","Z47","X48","Y48","Z48","EX7","XJ"
$
"P8","X49","Y49","Z49"
$
"P9","X49A","Y49A","Z49","X50","Y50","Z50","EX9","XJ"
$
*GOTO 120
$
*LABEL 110
$
"P1","X41","Y41","Z41","X42","Y42","Z42","NX1","XJ"
$
"P2","X33","Y33B","Z33","X43","Y43","Z43","EX2","XJ"
$
"P3","X43A","Y43A","Z43","X44","Y44","Z44","EX3","XJ"
$
"P4","X35","Y35B","Z35","X45","Y45","Z45","EX4","XJ"
$
"P5","X45A","Y45A","Z45","X46","Y46","Z46","EX5","XJ"
$
"P6","X37","Y37B","Z37","X47","Y47","Z47","EX6","XJ"
$
"P7","X47A","Y47A","Z47","X48","Y48","Z48","EX7","XJ"
$
"P8","X39","Y39B","Z39","X49","Y49","Z49","EX8","XJ"
$
"P9","X49A","Y49A","Z49","X50","Y50","Z50","EX9","XJ"
$
*GOTO 120
$
*LABEL 130
$
*IF("NX2" GT 2): *GOTO 140
$
"P1","X31","Y31B","Z31","X32","Y32B","Z32","NX1","XJ","EY5"
"YJ","X41","Y41","Z41","X42","Y42","Z42"
$
"P2","X33","Y33B","Z33","X43","Y43","Z43","EY5","YJ"
$
"P3","X33A","Y33C","Z33","X34","Y34B","Z34","EX3","XJ","EY5"
"YJ","X43A","Y43A","Z43","X44","Y44","Z44"
$
"P4","X35","Y35B","Z35","X45","Y45","Z45","EY5","YJ"
$
"P5","X35A","Y35C","Z35","X36","Y36B","Z36","EX5","XJ","EY5"
"YJ","X45A","Y45A","Z45","X46","Y46","Z46"

```

```

$
"P6","X37","Y37B","Z37","X47","Y47","Z47","EY5","YJ"
$
"P7","X37A","Y37C","Z37","X38","Y38B","Z38","EX7","XJ","EY5"
"YJ","X47A","Y47A","Z47","X48","Y48","Z48"
$
"P8","X39","Y39B","Z39","X49","Y49","Z49","EY5","YJ"
$
"P9","X39A","Y39C","Z39","X40","Y40B","Z40","EX9","XJ","EY5"
"YJ","X49A","Y49A","Z49","X50","Y50","Z50"
$
*GOTO 120
$
*LABEL 140
$
"P1","X31","Y31B","Z31","X32","Y32B","Z32","NX1","XJ","EY5"
"YJ","X41","Y41","Z41","X42","Y42","Z42"
$
"P2","X32A","Y32C","Z32","X33","Y33B","Z33","EX2","XJ","EY5"
"YJ","X42A","Y42","Z42","X43","Y43","Z43"
$
"P3","X33A","Y33C","Z33","X34","Y34B","Z34","EX3","XJ","EY5"
"YJ","X43A","Y43","Z43","X44","Y44","Z44"
$
"P4","X34A","Y34C","Z34","X35","Y35B","Z35","EX4","XJ","EY5"
"YJ","X44A","Y44","Z44","X45","Y45","Z45"
$
"P5","X35A","Y35C","Z35","X36","Y36B","Z36","EX5","XJ","EY5"
"YJ","X45A","Y45","Z45","X46","Y46","Z46"
$
"P6","X36A","Y36C","Z36","X37","Y37B","Z37","EX6","XJ","EY5"
"YJ","X46A","Y46","Z46","X47","Y47","Z47"
$
"P7","X37A","Y37C","Z37","X38","Y38B","Z38","EX7","XJ","EY5"
"YJ","X47A","Y47","Z47","X48","Y48","Z48"
$
"P8","X38A","Y38C","Z38","X39","Y39B","Z39","EX8","XJ","EY5"
"YJ","X48A","Y48","Z48","X49","Y49","Z49"
$
"P9","X39A","Y39C","Z39","X40","Y40B","Z40","EX9","XJ","EY5"
"YJ","X49A","Y49","Z49","X50","Y50","Z50"
$
*LABEL 120
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD4
$
$-----$
$ Subroutine (29 UPTO NODE)UPTO: This subroutine creates the nodes $

```

```

$                                     for the upper skin.                                     $
$-----$
$
$(29 UPTO NODE)UPTO
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY6" GT 2): *GOTO 170
*IF("NX2" GT 2): *GOTO 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X53", "Y53", "Z53"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X55", "Y55", "Z55"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X57", "Y57", "Z57"
$
"P7", "X57A", "Y57A", "Z57", "X58", "Y58", "Z58", "EX7", "XJ"
$
"P8", "X59", "Y59", "Z59"
$
"P9", "X59A", "Y59A", "Z59", "X60", "Y60", "Z60", "EX9", "XJ"
$
*GOTO 160
$
*LABEL 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X43", "Y43B", "Z43", "X53", "Y53", "Z53", "EX2", "XJ"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X45", "Y45B", "Z45", "X55", "Y55", "Z55", "EX4", "XJ"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X47", "Y47B", "Z47", "X57", "Y57", "Z57", "EX6", "XJ"
$
"P7", "X57A", "Y57A", "Z57", "X58", "Y58", "Z58", "EX7", "XJ"
$
"P8", "X49", "Y49B", "Z49", "X59", "Y59", "Z59", "EX8", "XJ"
$
"P9", "X59A", "Y59A", "Z59", "X60", "Y60", "Z60", "EX9", "XJ"
$
*GOTO 160

```

```

$
$
*LABEL 170
$
*IF("NX2" GT 2): *GOTO 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY6"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X43","Y43B","Z43","X53","Y53","Z53","EY6","YJ"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY6"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X45","Y45B","Z45","X55","Y55","Z55","EY6","YJ"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY6"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X47","Y47B","Z47","X57","Y57","Z57","EY6","YJ"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY6"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X49","Y49B","Z49","X59","Y59","Z59","EY6","YJ"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY6"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*GOTO 160
$
*LABEL 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY6"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X42A","Y42C","Z42","X43","Y43B","Z43","EX2","XJ","EY6"
"YJ","X52A","Y52A","Z52","X53","Y53","Z53"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY6"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X44A","Y44C","Z44","X45","Y45B","Z45","EX4","XJ","EY6"
"YJ","X54A","Y54A","Z54","X55","Y55","Z55"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY6"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X46A","Y46C","Z46","X47","Y47B","Z47","EX6","XJ","EY6"
"YJ","X56A","Y56A","Z56","X57","Y57","Z57"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY6"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X48A","Y48C","Z48","X49","Y49B","Z49","EX8","XJ","EY6"
"YJ","X58A","Y58A","Z58","X59","Y59","Z59"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY6"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"

```

```

$
*LABEL 160
$
!P1 = FREE ( )
!P2 = FREE ( )
!P3 = FREE ( )
!P4 = FREE ( )
!P5 = FREE ( )
!P6 = FREE ( )
!P7 = FREE ( )
!P8 = FREE ( )
!P9 = FREE ( )
$
*RETURN
*UPTO
$
$-----$
$ Subroutine (29 USKI NODE)NOD5: This subroutine creates the nodes $
$                               for the upper skin. $
$-----$
$
*(29 USKI NODE)NOD5
$
!P1 = ("YJ"*( "NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY7" GT 2): *GOTO 170
*IF("NX2" GT 2): *GOTO 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"
$
"P2", "X53", "Y53", "Z53"
$
"P3", "X53A", "Y53A", "Z53", "X54", "Y54", "Z54", "EX3", "XJ"
$
"P4", "X55", "Y55", "Z55"
$
"P5", "X55A", "Y55A", "Z55", "X56", "Y56", "Z56", "EX5", "XJ"
$
"P6", "X57", "Y57", "Z57"
$
"P7", "X57A", "Y57A", "Z57", "X58", "Y58", "Z58", "EX7", "XJ"
$
"P8", "X59", "Y59", "Z59"
$
"P9", "X59A", "Y59A", "Z59", "X60", "Y60", "Z60", "EX9", "XJ"
$
*GOTO 160
$
*LABEL 150
$
"P1", "X51", "Y51", "Z51", "X52", "Y52", "Z52", "NX1", "XJ"

```



```

$
"P2","X43","Y43B","Z43","X53","Y53","Z53","EX2","XJ"
$
"P3","X53A","Y53A","Z53","X54","Y54","Z54","EX3","XJ"
$
"P4","X45","Y45B","Z45","X55","Y55","Z55","EX4","XJ"
$
"P5","X55A","Y55A","Z55","X56","Y56","Z56","EX5","XJ"
$
"P6","X47","Y47B","Z47","X57","Y57","Z57","EX6","XJ"
$
"P7","X57A","Y57A","Z57","X58","Y58","Z58","EX7","XJ"
$
"P8","X49","Y49B","Z49","X59","Y59","Z59","EX8","XJ"
$
"P9","X59A","Y59A","Z59","X60","Y60","Z60","EX9","XJ"
$
*GOTO 160
$
$
$
*LABEL 170
$
*IF("NX2" GT 2): *GOTO 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY7"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X43","Y43B","Z43","X53","Y53","Z53","EY7","YJ"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY7"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"
$
"P4","X45","Y45B","Z45","X55","Y55","Z55","EY7","YJ"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY7"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X47","Y47B","Z47","X57","Y57","Z57","EY7","YJ"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY7"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X49","Y49B","Z49","X59","Y59","Z59","EY7","YJ"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY7"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*GOTO 160
$
$
*LABEL 180
$
"P1","X41","Y41B","Z41","X42","Y42B","Z42","NX1","XJ","EY7"
"YJ","X51","Y51","Z51","X52","Y52","Z52"
$
"P2","X42A","Y42C","Z42","X43","Y43B","Z43","EX2","XJ","EY7"
"YJ","X52A","Y52A","Z52","X53","Y53","Z53"
$
"P3","X43A","Y43C","Z43","X44","Y44B","Z44","EX3","XJ","EY7"
"YJ","X53A","Y53A","Z53","X54","Y54","Z54"

```

```

$
"P4","X44A","Y44C","Z44","X45","Y45B","Z45","EX4","XJ","EY7"
"YJ","X54A","Y54A","Z54","X55","Y55","Z55"
$
"P5","X45A","Y45C","Z45","X46","Y46B","Z46","EX5","XJ","EY7"
"YJ","X55A","Y55A","Z55","X56","Y56","Z56"
$
"P6","X46A","Y46C","Z46","X47","Y47B","Z47","EX6","XJ","EY7"
"YJ","X56A","Y56A","Z56","X57","Y57","Z57"
$
"P7","X47A","Y47C","Z47","X48","Y48B","Z48","EX7","XJ","EY7"
"YJ","X57A","Y57A","Z57","X58","Y58","Z58"
$
"P8","X48A","Y48C","Z48","X49","Y49B","Z49","EX8","XJ","EY7"
"YJ","X58A","Y58A","Z58","X59","Y59","Z59"
$
"P9","X49A","Y49C","Z49","X50","Y50B","Z50","EX9","XJ","EY7"
"YJ","X59A","Y59A","Z59","X60","Y60","Z60"
$
*LABEL 160
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD5
$
$-----$
$ Subroutine (29 UHON NODE)NOD6: This subroutine creates the nodes $
$                               for the upper honeycomb structure. $
$-----$
$
*(29 UHON NODE)NOD6
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY8" GT 2): *GOTO 210
*IF("NX2" GT 2): *GOTO 190
$
"P1","X61","Y61","Z61","X62","Y62","Z62","NX1","XJ"
$
"P2","X63","Y63","Z63"
$
"P3","X63A","Y63A","Z63","X64","Y64","Z64","EX3","XJ"

```

```

$
"P4","X65","Y65","Z65"
$
"P5","X65A","Y65A","Z65","X66","Y66","Z66","EX5","XJ"
$
"P6","X67","Y67","Z67"
$
"P7","X67A","Y67A","Z67","X68","Y68","Z68","EX7","XJ"
$
"P8","X69","Y69","Z69"
$
"P9","X69A","Y69A","Z69","X70","Y70","Z70","EX9","XJ"
$
*GOTO 200
$
*LABEL 190
$
"P1","X61","Y61","Z61","X62","Y62","Z62","NX1","XJ"
$
"P2","X53","Y53B","Z53","X63","Y63","Z63","EX2","XJ"
$
"P3","X63A","Y63A","Z63","X64","Y64","Z64","EX3","XJ"
$
"P4","X55","Y55B","Z55","X65","Y65","Z65","EX4","XJ"
$
"P5","X65A","Y65A","Z65","X66","Y66","Z66","EX5","XJ"
$
"P6","X57","Y57B","Z57","X67","Y67","Z67","EX6","XJ"
$
"P7","X67A","Y67A","Z67","X68","Y68","Z68","EX7","XJ"
$
"P8","X59","Y59B","Z59","X69","Y69","Z69","EX8","XJ"
$
"P9","X69A","Y69A","Z69","X70","Y70","Z70","EX9","XJ"
$
*GOTO 200
$
*LABEL 210
$
*IF("NX2" GT 2): *GOTO 220
$
"P1","X51","Y51B","Z51","X52","Y52B","Z52","NX1","XJ","EY8"
"YJ","X61","Y61","Z61","X62","Y62","Z62"
$
"P2","X53","Y53B","Z53","X63","Y63","Z63","EY8","YJ"
$
"P3","X53A","Y53C","Z53","X54","Y54B","Z54","EX3","XJ","EY8"
"YJ","X63A","Y63A","Z63","X64","Y64","Z64"
$
"P4","X55","Y55B","Z55","X65","Y65","Z65","EY8","YJ"
$
"P5","X55A","Y55C","Z55","X56","Y56B","Z56","EX5","XJ","EY8"
"YJ","X65A","Y65A","Z65","X66","Y66","Z66"
$
"P6","X57","Y57B","Z57","X67","Y67","Z67","EY8","YJ"
$
"P7","X57A","Y57C","Z57","X58","Y58B","Z58","EX7","XJ","EY8"
"YJ","X67A","Y67A","Z67","X68","Y68","Z68"
$

```

```

"P8","X59","Y59B","Z59","X69","Y69","Z69","EY8","YJ"
$
"P9","X59A","Y59C","Z59","X60","Y60B","Z60","EX9","XJ","EY8"
"YJ","X69A","Y69A","Z69","X70","Y70","Z70"
$
*GOTO 200
$
*LABEL 220
$
"P1","X51","Y51B","Z51","X52","Y52B","Z52","NX1","XJ","EY8"
"YJ","X61","Y61","Z61","X62","Y62","Z62"
$
"P2","X52A","Y52C","Z52","X53","Y53B","Z53","EX2","XJ","EY8"
"YJ","X62A","Y62A","Z62","X63","Y63","Z63"
$
"P3","X53A","Y53C","Z53","X54","Y54B","Z54","EX3","XJ","EY8"
"YJ","X63A","Y63A","Z63","X64","Y64","Z64"
$
"P4","X54A","Y54C","Z54","X55","Y55B","Z55","EX4","XJ","EY8"
"YJ","X64A","Y64A","Z64","X65","Y65","Z65"
$
"P5","X55A","Y55C","Z55","X56","Y56B","Z56","EX5","XJ","EY8"
"YJ","X65A","Y65A","Z65","X66","Y66","Z66"
$
"P6","X56A","Y56C","Z56","X57","Y57B","Z57","EX6","XJ","EY8"
"YJ","X66A","Y66A","Z66","X67","Y67","Z67"
$
"P7","X57A","Y57C","Z57","X58","Y58B","Z58","EX7","XJ","EY8"
"YJ","X67A","Y67A","Z67","X68","Y68","Z68"
$
"P8","X58A","Y58C","Z58","X59","Y59B","Z59","EX8","XJ","EY8"
"YJ","X68A","Y68A","Z68","X69","Y69","Z69"
$
"P9","X59A","Y59C","Z59","X60","Y60B","Z60","EX9","XJ","EY8"
"YJ","X69A","Y69A","Z69","X70","Y70","Z70"
$
*LABEL 200
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD6
$
$-----$
$ Subroutine (29 UFAC NODE)NOD7: This subroutine creates the nodes $
$                               for the upper outer facesheet. $
$-----$
$
*(29 UFAC NODE)NOD7
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"))+1

```

```

!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY9" GT 2): *GOTO 250
*IF("NX2" GT 2): *GOTO 230
$
"P1","X71","Y71","Z71","X72","Y72","Z72","NX1","XJ"
$
"P2","X73","Y73","Z73"
$
"P3","X73A","Y73A","Z73","X74","Y74","Z74","EX3","XJ"
$
"P4","X75","Y75","Z75"
$
"P5","X75A","Y75A","Z75","X76","Y76","Z76","EX5","XJ"
$
"P6","X77","Y77","Z77"
$
"P7","X77A","Y77A","Z77","X78","Y78","Z78","EX7","XJ"
$
"P8","X79","Y79","Z79"
$
"P9","X79A","Y79A","Z79","X80","Y80","Z80","EX9","XJ"
$
*GOTO 240
$
*LABEL 230
$
"P1","X71","Y71","Z71","X72","Y72","Z72","NX1","XJ"
$
"P2","X63","Y63B","Z63","X73","Y73","Z73","EX2","XJ"
$
"P3","X73A","Y73A","Z73","X74","Y74","Z74","EX3","XJ"
$
"P4","X65","Y65B","Z65","X75","Y75","Z75","EX4","XJ"
$
"P5","X75A","Y75A","Z75","X76","Y76","Z76","EX5","XJ"
$
"P6","X67","Y67B","Z67","X77","Y77","Z77","EX6","XJ"
$
"P7","X77A","Y77A","Z77","X78","Y78","Z78","EX7","XJ"
$
"P8","X69","Y69B","Z69","X79","Y79","Z79","EX8","XJ"
$
"P9","X79A","Y79A","Z79","X80","Y80","Z80","EX9","XJ"
$
*GOTO 240
$
*LABEL 250
$
*IF("NX2" GT 2): *GOTO 260
$
"P1","X61","Y61B","Z61","X62","Y62B","Z62","NX1","XJ","EY9"

```

```

"YJ", "X71", "Y71", "Z71", "X72", "Y72", "Z72"
$
"P2", "X63", "Y63B", "Z63", "X73", "Y73", "Z73", "EY9", "YJ"
$
"P3", "X63A", "Y63C", "Z63", "X64", "Y64B", "Z64", "EX3", "XJ", "EY9"
"YJ", "X73A", "Y73A", "Z73", "X74", "Y74", "Z74"
$
"P4", "X65", "Y65B", "Z65", "X75", "Y75", "Z75", "EY9", "YJ"
$
"P5", "X65A", "Y65C", "Z65", "X66", "Y66B", "Z66", "EX5", "XJ", "EY9"
"YJ", "X75A", "Y75A", "Z75", "X76", "Y76", "Z76"
$
"P6", "X67", "Y67B", "Z67", "X77", "Y77", "Z77", "EY9", "YJ"
$
"P7", "X67A", "Y67C", "Z67", "X68", "Y68B", "Z68", "EX7", "XJ", "EY9"
"YJ", "X77A", "Y77A", "Z77", "X78", "Y78", "Z78"
$
"P8", "X69", "Y69B", "Z69", "X79", "Y79", "Z79", "EY9", "YJ"
$
"P9", "X69A", "Y69C", "Z69", "X70", "Y70B", "Z70", "EX9", "XJ", "EY9"
"YJ", "X79A", "Y79A", "Z79", "X80", "Y80", "Z80"
$
*GOTO 240
$
*LABEL 260
$
"P1", "X61", "Y61B", "Z61", "X62", "Y62B", "Z62", "NX1", "XJ", "EY9"
"YJ", "X71", "Y71", "Z71", "X72", "Y72", "Z72"
$
"P2", "X62A", "Y62C", "Z62", "X63", "Y63B", "Z63", "EX2", "XJ", "EY9"
"YJ", "X72A", "Y72A", "Z72", "X73", "Y73", "Z73"
$
"P3", "X63A", "Y63C", "Z63", "X64", "Y64B", "Z64", "EX3", "XJ", "EY9"
"YJ", "X73A", "Y73A", "Z73", "X74", "Y74", "Z74"
$
"P4", "X64A", "Y64C", "Z64", "X65", "Y65B", "Z65", "EX4", "XJ", "EY9"
"YJ", "X74A", "Y74A", "Z74", "X75", "Y75", "Z75"
$
"P5", "X65A", "Y65C", "Z65", "X66", "Y66B", "Z66", "EX5", "XJ", "EY9"
"YJ", "X75A", "Y75A", "Z75", "X76", "Y76", "Z76"
$
"P6", "X66A", "Y66C", "Z66", "X67", "Y67B", "Z67", "EX6", "XJ", "EY9"
"YJ", "X76A", "Y76A", "Z76", "X77", "Y77", "Z77"
$
"P7", "X67A", "Y67C", "Z67", "X68", "Y68B", "Z68", "EX7", "XJ", "EY9"
"YJ", "X77A", "Y77A", "Z77", "X78", "Y78", "Z78"
$
"P8", "X68A", "Y68C", "Z68", "X69", "Y69B", "Z69", "EX8", "XJ", "EY9"
"YJ", "X78A", "Y78A", "Z78", "X79", "Y79", "Z79"
$
"P9", "X69A", "Y69C", "Z69", "X70", "Y70B", "Z70", "EX9", "XJ", "EY9"
"YJ", "X79A", "Y79A", "Z79", "X80", "Y80", "Z80"
$
*LABEL 240
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()

```

```

!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD7
$
$-----$
$ Subroutine (29 INSU NODE)NOD8: This subroutine creates the nodes $
$                               for the upper outer facesheet. $
$-----$
$
*(29 INSU NODE)NOD8
$
!P1 = ("YJ"*("NY1"+"EY2"+"EY3"+"EY4"+"EY5"+"EY6"+"EY7"+"EY8"+"EY9"))+1
!P2 = "P1"+"NX1"
!P3 = "P2"+"EX2"
!P4 = "P3"+"EX3"
!P5 = "P4"+"EX4"
!P6 = "P5"+"EX5"
!P7 = "P6"+"EX6"
!P8 = "P7"+"EX7"
!P9 = "P8"+"EX8"
$
*IF("NY10" GT 2): *GOTO 250
*IF("NX2" GT 2): *GOTO 230
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X13","Y13","Z13"
$
"P3","X13A","Y13A","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X15","Y15","Z15"
$
"P5","X15A","Y15A","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X17","Y17","Z17"
$
"P7","X17A","Y17A","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X19","Y19","Z19"
$
"P9","X19A","Y19A","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 240
$
*LABEL 230
$
"P1","X11","Y11","Z11","X12","Y12","Z12","NX1","XJ"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EX2","XJ"
$
"P3","X13A","Y13A","Z13","X14","Y14","Z14","EX3","XJ"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EX4","XJ"
$

```

```

"P5","X15A","Y15A","Z15","X16","Y16","Z16","EX5","XJ"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EX6","XJ"
$
"P7","X17A","Y17A","Z17","X18","Y18","Z18","EX7","XJ"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EX8","XJ"
$
"P9","X19A","Y19A","Z19","X20","Y20","Z20","EX9","XJ"
$
*GOTO 240
$
*LABEL 250
$
*IF("NX2" GT 2): *GOTO 260
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY10"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X3","Y3B","Z3","X13","Y13","Z13","EY10","YJ"
$
"P3","X3A","Y3C","Z3","X4","Y4B","Z4","EX3","XJ","EY10"
"YJ","X13A","Y13A","Z13","X14","Y14","Z14"
$
"P4","X5","Y5B","Z5","X15","Y15","Z15","EY10","YJ"
$
"P5","X5A","Y5C","Z5","X6","Y6B","Z6","EX5","XJ","EY10"
"YJ","X15A","Y15A","Z15","X16","Y16","Z16"
$
"P6","X7","Y7B","Z7","X17","Y17","Z17","EY10","YJ"
$
"P7","X7A","Y7C","Z7","X8","Y8B","Z8","EX7","XJ","EY10"
"YJ","X17A","Y17A","Z17","X18","Y18","Z18"
$
"P8","X9","Y9B","Z9","X19","Y19","Z19","EY10","YJ"
$
"P9","X9A","Y9C","Z9","X10","Y10B","Z10","EX9","XJ","EY10"
"YJ","X19A","Y19A","Z19","X20","Y20","Z20"
$
*GOTO 240
$
*LABEL 260
$
"P1","X1","Y1B","Z1","X2","Y2B","Z2","NX1","XJ","EY10"
"YJ","X11","Y11","Z11","X12","Y12","Z12"
$
"P2","X2A","Y2C","Z2","X3","Y3B","Z3","EX2","XJ","EY10"
"YJ","X12A","Y12A","Z12","X13","Y13","Z13"
$
"P3","X3A","Y3C","Z3","X4","Y4B","Z4","EX3","XJ","EY10"
"YJ","X13A","Y13A","Z13","X14","Y14","Z14"
$
"P4","X4A","Y4C","Z4","X5","Y5B","Z5","EX4","XJ","EY10"
"YJ","X14A","Y14A","Z14","X15","Y15","Z15"
$
"P5","X5A","Y5C","Z5","X6","Y6B","Z6","EX5","XJ","EY10"
"YJ","X15A","Y15A","Z15","X16","Y16","Z16"
$
"P6","X6A","Y6C","Z6","X7","Y7B","Z7","EX6","XJ","EY10"

```



```

"YJ","X16A","Y16A","Z16","X17","Y17","Z17"
$
"P7","X7A","Y7C","Z7","X8","Y8B","Z8","EX7","XJ","EY10"
"YJ","X17A","Y17A","Z17","X18","Y18","Z18"
$
"P8","X8A","Y8C","Z8","X9","Y9B","Z9","EX8","XJ","EY10"
"YJ","X18A","Y18A","Z18","X19","Y19","Z19"
$
"P9","X9A","Y9C","Z9","X10","Y10B","Z10","EX9","XJ","EY10"
"YJ","X19A","Y19A","Z19","X20","Y20","Z20"
$
*LABEL 240
$
!P1 = FREE ()
!P2 = FREE ()
!P3 = FREE ()
!P4 = FREE ()
!P5 = FREE ()
!P6 = FREE ()
!P7 = FREE ()
!P8 = FREE ()
!P9 = FREE ()
$
*RETURN
*NOD8
$
$-----$
$ Subroutine (29 LFAC FREE)FRE1:  This subroutine deletes the  $
$                                coordinates for the lower outer  $
$                                facsheet.                        $
$-----$
$
$
*(29 LFAC FREE)FRE1
$
!X1  = FREE ()
!X2  = FREE ()
!X3  = FREE ()
!X4  = FREE ()
!X5  = FREE ()
!X6  = FREE ()
!X7  = FREE ()
!X8  = FREE ()
!X9  = FREE ()
!X10 = FREE ()
!X11 = FREE ()
!X12 = FREE ()
!X13 = FREE ()
!X14 = FREE ()
!X15 = FREE ()
!X16 = FREE ()
!X17 = FREE ()
!X18 = FREE ()
!X19 = FREE ()
!X20 = FREE ()
$
!X1A = FREE ()
!X2A = FREE ()
!X3A = FREE ()
!X4A = FREE ()

```

```

!X5A = FREE ()
!X6A = FREE ()
!X7A = FREE ()
!X8A = FREE ()
!X9A = FREE ()
!X11A= FREE ()
!X12A= FREE ()
!X13A= FREE ()
!X14A= FREE ()
!X15A= FREE ()
!X16A= FREE ()
!X17A= FREE ()
!X18A= FREE ()
!X19A= FREE ()
$
!Y1  = FREE ()
!Y2  = FREE ()
!Y3  = FREE ()
!Y4  = FREE ()
!Y5  = FREE ()
!Y6  = FREE ()
!Y7  = FREE ()
!Y8  = FREE ()
!Y9  = FREE ()
!Y10 = FREE ()
!Y11 = FREE ()
!Y12 = FREE ()
!Y13 = FREE ()
!Y14 = FREE ()
!Y15 = FREE ()
!Y16 = FREE ()
!Y17 = FREE ()
!Y18 = FREE ()
!Y19 = FREE ()
!Y20 = FREE ()
$
!Z1  = FREE ()
!Z2  = FREE ()
!Z3  = FREE ()
!Z4  = FREE ()
!Z5  = FREE ()
!Z6  = FREE ()
!Z7  = FREE ()
!Z8  = FREE ()
!Z9  = FREE ()
!Z10 = FREE ()
!Z11 = FREE ()
!Z12 = FREE ()
!Z13 = FREE ()
!Z14 = FREE ()
!Z15 = FREE ()
!Z16 = FREE ()
!Z17 = FREE ()
!Z18 = FREE ()
!Z19 = FREE ()
!Z20 = FREE ()
$
*RETURN
*FRE1

```

```

$
$-----$
$ Subroutine (29 LHON FREE)FRE2: This subroutine deletes the $
$                               coordinates for the lower honey- $
$                               comb structure. $
$-----$
$
$(29 LHON FREE)FRE2
$
!X11 = FREE ()
!X12 = FREE ()
!X13 = FREE ()
!X14 = FREE ()
!X15 = FREE ()
!X16 = FREE ()
!X17 = FREE ()
!X18 = FREE ()
!X19 = FREE ()
!X20 = FREE ()
!X21 = FREE ()
!X22 = FREE ()
!X23 = FREE ()
!X24 = FREE ()
!X25 = FREE ()
!X26 = FREE ()
!X27 = FREE ()
!X28 = FREE ()
!X29 = FREE ()
!X30 = FREE ()
$
!X11A= FREE ()
!X12A= FREE ()
!X13A= FREE ()
!X14A= FREE ()
!X15A= FREE ()
!X16A= FREE ()
!X17A= FREE ()
!X18A= FREE ()
!X19A= FREE ()
!X21A= FREE ()
!X22A= FREE ()
!X23A= FREE ()
!X24A= FREE ()
!X25A= FREE ()
!X26A= FREE ()
!X27A= FREE ()
!X28A= FREE ()
!X29A= FREE ()
$
!Y11 = FREE ()
!Y12 = FREE ()
!Y13 = FREE ()
!Y14 = FREE ()
!Y15 = FREE ()
!Y16 = FREE ()
!Y17 = FREE ()
!Y18 = FREE ()
!Y19 = FREE ()
!Y20 = FREE ()

```

```

!Y21 = FREE ()
!Y22 = FREE ()
!Y23 = FREE ()
!Y24 = FREE ()
!Y25 = FREE ()
!Y26 = FREE ()
!Y27 = FREE ()
!Y28 = FREE ()
!Y29 = FREE ()
!Y30 = FREE ()
$
!Y11B= FREE ()
!Y12B= FREE ()
!Y13B= FREE ()
!Y14B= FREE ()
!Y15B= FREE ()
!Y16B= FREE ()
!Y17B= FREE ()
!Y18B= FREE ()
!Y19B= FREE ()
!Y20B= FREE ()
$
!Z11 = FREE ()
!Z12 = FREE ()
!Z13 = FREE ()
!Z14 = FREE ()
!Z15 = FREE ()
!Z16 = FREE ()
!Z17 = FREE ()
!Z18 = FREE ()
!Z19 = FREE ()
!Z20 = FREE ()
!Z21 = FREE ()
!Z22 = FREE ()
!Z23 = FREE ()
!Z24 = FREE ()
!Z25 = FREE ()
!Z26 = FREE ()
!Z27 = FREE ()
!Z28 = FREE ()
!Z29 = FREE ()
!Z30 = FREE ()
$
*RETURN
*FRE2
$
$-----$
$ Subroutine (29 LSKI FREE)FRE3: This subroutine deletes the $
$                                     coordinates for the lower skin. $
$-----$
$
$(29 LSKI FREE)FRE3
$
!X21 = FREE ()
!X22 = FREE ()
!X23 = FREE ()
!X24 = FREE ()
!X25 = FREE ()
!X26 = FREE ()

```

```

!X27 = FREE ()
!X28 = FREE ()
!X29 = FREE ()
!X30 = FREE ()
!X31 = FREE ()
!X32 = FREE ()
!X33 = FREE ()
!X34 = FREE ()
!X35 = FREE ()
!X36 = FREE ()
!X37 = FREE ()
!X38 = FREE ()
!X39 = FREE ()
!X40 = FREE ()
$
!X21A= FREE ()
!X22A= FREE ()
!X23A= FREE ()
!X24A= FREE ()
!X25A= FREE ()
!X26A= FREE ()
!X27A= FREE ()
!X28A= FREE ()
!X29A= FREE ()
!X31A= FREE ()
!X32A= FREE ()
!X33A= FREE ()
!X34A= FREE ()
!X35A= FREE ()
!X36A= FREE ()
!X37A= FREE ()
!X38A= FREE ()
!X39A= FREE ()
$
!Y21 = FREE ()
!Y22 = FREE ()
!Y23 = FREE ()
!Y24 = FREE ()
!Y25 = FREE ()
!Y26 = FREE ()
!Y27 = FREE ()
!Y28 = FREE ()
!Y29 = FREE ()
!Y30 = FREE ()
!Y31 = FREE ()
!Y32 = FREE ()
!Y33 = FREE ()
!Y34 = FREE ()
!Y35 = FREE ()
!Y36 = FREE ()
!Y37 = FREE ()
!Y38 = FREE ()
!Y39 = FREE ()
!Y40 = FREE ()
$
!Y21B= FREE ()
!Y22B= FREE ()
!Y23B= FREE ()
!Y24B= FREE ()

```

```

!Y25B= FREE ()
!Y26B= FREE ()
!Y27B= FREE ()
!Y28B= FREE ()
!Y29B= FREE ()
!Y30B= FREE ()
$
!Z21 = FREE ()
!Z22 = FREE ()
!Z23 = FREE ()
!Z24 = FREE ()
!Z25 = FREE ()
!Z26 = FREE ()
!Z27 = FREE ()
!Z28 = FREE ()
!Z29 = FREE ()
!Z30 = FREE ()
!Z31 = FREE ()
!Z32 = FREE ()
!Z33 = FREE ()
!Z34 = FREE ()
!Z35 = FREE ()
!Z36 = FREE ()
!Z37 = FREE ()
!Z38 = FREE ()
!Z39 = FREE ()
!Z40 = FREE ()
$
*RETURN
*FRE3
$
$-----$
$ Subroutine (29 LTOR FREE)TORL: This subroutine deletes the $
$                               coordinates for the air section of $
$                               the model. $
$-----$
$
*(29 LTOR FREE)TORL
$
!X1  = FREE()
!X2  = FREE()
!X3  = FREE()
!X4  = FREE()
!X5  = FREE()
!X6  = FREE()
!X7  = FREE()
!X8  = FREE()
!X9  = FREE()
!X10 = FREE()
!X11 = FREE()
!X12 = FREE()
!X13 = FREE()
!X14 = FREE()
!X15 = FREE()
!X16 = FREE()
!X17 = FREE()
!X18 = FREE()
!X19 = FREE()
!X20 = FREE()

```

```

$
!X1A = FREE()
!X2A = FREE()
!X3A = FREE()
!X4A = FREE()
!X5A = FREE()
!X6A = FREE()
!X7A = FREE()
!X8A = FREE()
!X9A = FREE()
!X11A= FREE()
!X12A= FREE()
!X13A= FREE()
!X14A= FREE()
!X15A= FREE()
!X16A= FREE()
!X17A= FREE()
!X18A= FREE()
!X39A= FREE()
$
!Y1  = FREE()
!Y2  = FREE()
!Y3  = FREE()
!Y4  = FREE()
!Y5  = FREE()
!Y6  = FREE()
!Y7  = FREE()
!Y8  = FREE()
!Y9  = FREE()
!Y10 = FREE()
!Y11 = FREE()
!Y12 = FREE()
!Y13 = FREE()
!Y14 = FREE()
!Y15 = FREE()
!Y16 = FREE()
!Y17 = FREE()
!Y18 = FREE()
!Y19 = FREE()
!Y20 = FREE()
$
!Y1B = FREE()
!Y2B = FREE()
!Y3B = FREE()
!Y4B = FREE()
!Y5B = FREE()
!Y6B = FREE()
!Y7B = FREE()
!Y8B = FREE()
!Y9B = FREE()
!Y10B= FREE()
$
!Z1  = FREE()
!Z2  = FREE()
!Z3  = FREE()
!Z4  = FREE()
!Z5  = FREE()
!Z6  = FREE()
!Z7  = FREE()

```

```

!Z8  = FREE()
!Z9  = FREE()
!Z10 = FREE()
!Z11 = FREE()
!Z12 = FREE()
!Z13 = FREE()
!Z14 = FREE()
!Z15 = FREE()
!Z16 = FREE()
!Z17 = FREE()
!Z18 = FREE()
!Z19 = FREE()
!Z20 = FREE()
$
*RETURN
*TORL
$
$-----$
$ Subroutine (29 AIRG FREE)FRE4: This subroutine deletes the      $
$                               coordinates for the mid section of $
$                               the model.                          $
$-----$
$
*(29 AIRG FREE)FRE4
$
!X31 = FREE ()
!X32 = FREE ()
!X33 = FREE ()
!X34 = FREE ()
!X35 = FREE ()
!X36 = FREE ()
!X37 = FREE ()
!X38 = FREE ()
!X39 = FREE ()
!X40 = FREE ()
!X41 = FREE ()
!X42 = FREE ()
!X43 = FREE ()
!X44 = FREE ()
!X45 = FREE ()
!X46 = FREE ()
!X47 = FREE ()
!X48 = FREE ()
!X49 = FREE ()
!X50 = FREE ()
$
!X31A= FREE ()
!X32A= FREE ()
!X33A= FREE ()
!X34A= FREE ()
!X35A= FREE ()
!X36A= FREE ()
!X37A= FREE ()
!X38A= FREE ()
!X39A= FREE ()
!X41A= FREE ()
!X42A= FREE ()
!X43A= FREE ()
!X44A= FREE ()

```



```

!X45A= FREE ()
!X46A= FREE ()
!X47A= FREE ()
!X48A= FREE ()
!X49A= FREE ()
$
!Y31 = FREE ()
!Y32 = FREE ()
!Y33 = FREE ()
!Y34 = FREE ()
!Y35 = FREE ()
!Y36 = FREE ()
!Y37 = FREE ()
!Y38 = FREE ()
!Y39 = FREE ()
!Y40 = FREE ()
!Y41 = FREE ()
!Y42 = FREE ()
!Y43 = FREE ()
!Y44 = FREE ()
!Y45 = FREE ()
!Y46 = FREE ()
!Y47 = FREE ()
!Y48 = FREE ()
!Y49 = FREE ()
!Y50 = FREE ()
$
!Y42A= FREE ()
!Y43A= FREE ()
!Y44A= FREE ()
!Y45A= FREE ()
!Y46A= FREE ()
!Y47A= FREE ()
!Y48A= FREE ()
!Y49A= FREE ()
$
!Y31B= FREE ()
!Y32B= FREE ()
!Y33B= FREE ()
!Y34B= FREE ()
!Y35B= FREE ()
!Y36B= FREE ()
!Y37B= FREE ()
!Y38B= FREE ()
!Y39B= FREE ()
!Y40B= FREE ()
$
!Y32C= FREE ()
!Y33C= FREE ()
!Y34C= FREE ()
!Y35C= FREE ()
!Y36C= FREE ()
!Y37C= FREE ()
!Y38C= FREE ()
!Y39C= FREE ()
$
!Z31 = FREE ()
!Z32 = FREE ()
!Z33 = FREE ()

```

```

!Z34 = FREE ()
!Z35 = FREE ()
!Z36 = FREE ()
!Z37 = FREE ()
!Z38 = FREE ()
!Z39 = FREE ()
!Z40 = FREE ()
!Z41 = FREE ()
!Z42 = FREE ()
!Z43 = FREE ()
!Z44 = FREE ()
!Z45 = FREE ()
!Z46 = FREE ()
!Z47 = FREE ()
!Z48 = FREE ()
!Z49 = FREE ()
!Z50 = FREE ()
$
*RETURN
*FRE4
$
$-----$
$ Subroutine (29 TORU FREE)TORU: This subroutine deletes the $
$                               coordinates for the upper torque $
$                               tube. $
$-----$
$
$(29 TORU FREE)TORU
$
!X41 = FREE ()
!X42 = FREE ()
!X43 = FREE ()
!X44 = FREE ()
!X45 = FREE ()
!X46 = FREE ()
!X47 = FREE ()
!X48 = FREE ()
!X49 = FREE ()
!X50 = FREE ()
!X51 = FREE ()
!X52 = FREE ()
!X53 = FREE ()
!X54 = FREE ()
!X55 = FREE ()
!X56 = FREE ()
!X57 = FREE ()
!X58 = FREE ()
!X59 = FREE ()
!X60 = FREE ()
$
!X41A= FREE ()
!X42A= FREE ()
!X43A= FREE ()
!X44A= FREE ()
!X45A= FREE ()
!X46A= FREE ()
!X47A= FREE ()
!X48A= FREE ()
!X49A= FREE ()

```

```

!X51A= FREE ()
!X52A= FREE ()
!X53A= FREE ()
!X54A= FREE ()
!X55A= FREE ()
!X56A= FREE ()
!X57A= FREE ()
!X58A= FREE ()
!X59A= FREE ()
$
!Y41 = FREE ()
!Y42 = FREE ()
!Y43 = FREE ()
!Y44 = FREE ()
!Y45 = FREE ()
!Y46 = FREE ()
!Y47 = FREE ()
!Y48 = FREE ()
!Y49 = FREE ()
!Y50 = FREE ()
!Y51 = FREE ()
!Y52 = FREE ()
!Y53 = FREE ()
!Y54 = FREE ()
!Y55 = FREE ()
!Y56 = FREE ()
!Y57 = FREE ()
!Y58 = FREE ()
!Y59 = FREE ()
!Y60 = FREE ()
$
!Y52A= FREE ()
!Y53A= FREE ()
!Y54A= FREE ()
!Y55A= FREE ()
!Y56A= FREE ()
!Y57A= FREE ()
!Y58A= FREE ()
!Y59A= FREE ()
$
!Y41B= FREE ()
!Y42B= FREE ()
!Y43B= FREE ()
!Y44B= FREE ()
!Y45B= FREE ()
!Y46B= FREE ()
!Y47B= FREE ()
!Y48B= FREE ()
!Y49B= FREE ()
!Y50B= FREE ()
$
!Y42C= FREE ()
!Y43C= FREE ()
!Y44C= FREE ()
!Y45C= FREE ()
!Y46C= FREE ()
!Y47C= FREE ()
!Y48C= FREE ()
!Y49C= FREE ()

```

```

$
!Z41 = FREE ()
!Z42 = FREE ()
!Z43 = FREE ()
!Z44 = FREE ()
!Z45 = FREE ()
!Z46 = FREE ()
!Z47 = FREE ()
!Z48 = FREE ()
!Z49 = FREE ()
!Z50 = FREE ()
!Z51 = FREE ()
!Z52 = FREE ()
!Z53 = FREE ()
!Z54 = FREE ()
!Z55 = FREE ()
!Z56 = FREE ()
!Z57 = FREE ()
!Z58 = FREE ()
!Z59 = FREE ()
!Z60 = FREE ()
$
*RETURN
*TORU
$
$-----$
$ Subroutine (29 USKI FREE)FRE5: This subroutine deletes the      $
$                               coordinates for the upper skin.    $
$-----$
$
*(29 USKI FREE)FRE5
$
!X41 = FREE ()
!X42 = FREE ()
!X43 = FREE ()
!X44 = FREE ()
!X45 = FREE ()
!X46 = FREE ()
!X47 = FREE ()
!X48 = FREE ()
!X49 = FREE ()
!X50 = FREE ()
!X51 = FREE ()
!X52 = FREE ()
!X53 = FREE ()
!X54 = FREE ()
!X55 = FREE ()
!X56 = FREE ()
!X57 = FREE ()
!X58 = FREE ()
!X59 = FREE ()
!X60 = FREE ()
$
!X41A= FREE ()
!X42A= FREE ()
!X43A= FREE ()
!X44A= FREE ()
!X45A= FREE ()
!X46A= FREE ()

```

```

!X47A= FREE ()
!X48A= FREE ()
!X49A= FREE ()
!X51A= FREE ()
!X52A= FREE ()
!X53A= FREE ()
!X54A= FREE ()
!X55A= FREE ()
!X56A= FREE ()
!X57A= FREE ()
!X58A= FREE ()
!X59A= FREE ()
$
!Y41 = FREE ()
!Y42 = FREE ()
!Y43 = FREE ()
!Y44 = FREE ()
!Y45 = FREE ()
!Y46 = FREE ()
!Y47 = FREE ()
!Y48 = FREE ()
!Y49 = FREE ()
!Y50 = FREE ()
!Y51 = FREE ()
!Y52 = FREE ()
!Y53 = FREE ()
!Y54 = FREE ()
!Y55 = FREE ()
!Y56 = FREE ()
!Y57 = FREE ()
!Y58 = FREE ()
!Y59 = FREE ()
!Y60 = FREE ()
$
!Y52A= FREE ()
!Y53A= FREE ()
!Y54A= FREE ()
!Y55A= FREE ()
!Y56A= FREE ()
!Y57A= FREE ()
!Y58A= FREE ()
!Y59A= FREE ()
$
!Y41B= FREE ()
!Y42B= FREE ()
!Y43B= FREE ()
!Y44B= FREE ()
!Y45B= FREE ()
!Y46B= FREE ()
!Y47B= FREE ()
!Y48B= FREE ()
!Y49B= FREE ()
!Y50B= FREE ()
$
!Y42C= FREE ()
!Y43C= FREE ()
!Y44C= FREE ()
!Y45C= FREE ()
!Y46C= FREE ()

```

```

!Y47C= FREE ()
!Y48C= FREE ()
!Y49C= FREE ()
$
!Z41 = FREE ()
!Z42 = FREE ()
!Z43 = FREE ()
!Z44 = FREE ()
!Z45 = FREE ()
!Z46 = FREE ()
!Z47 = FREE ()
!Z48 = FREE ()
!Z49 = FREE ()
!Z50 = FREE ()
!Z51 = FREE ()
!Z52 = FREE ()
!Z53 = FREE ()
!Z54 = FREE ()
!Z55 = FREE ()
!Z56 = FREE ()
!Z57 = FREE ()
!Z58 = FREE ()
!Z59 = FREE ()
!Z60 = FREE ()
$
*RETURN
*FRE5
$
$-----$
$ Subroutine (29 UHON FREE)FRE6: This subroutine deletes the $
$                               coordinates for the upper honey- $
$                               comb structure.                    $
$-----$
$
*(29 UHON FREE)FRE6
$
!X51 = FREE ()
!X52 = FREE ()
!X53 = FREE ()
!X54 = FREE ()
!X55 = FREE ()
!X56 = FREE ()
!X57 = FREE ()
!X58 = FREE ()
!X59 = FREE ()
!X60 = FREE ()
!X61 = FREE ()
!X62 = FREE ()
!X63 = FREE ()
!X64 = FREE ()
!X65 = FREE ()
!X66 = FREE ()
!X67 = FREE ()
!X68 = FREE ()
!X69 = FREE ()
!X70 = FREE ()
$
!X51A= FREE ()
!X52A= FREE ()

```

```

!X53A= FREE ()
!X54A= FREE ()
!X55A= FREE ()
!X56A= FREE ()
!X57A= FREE ()
!X58A= FREE ()
!X59A= FREE ()
!X61A= FREE ()
!X62A= FREE ()
!X63A= FREE ()
!X64A= FREE ()
!X65A= FREE ()
!X66A= FREE ()
!X67A= FREE ()
!X68A= FREE ()
!X69A= FREE ()
$
!Y51 = FREE ()
!Y52 = FREE ()
!Y53 = FREE ()
!Y54 = FREE ()
!Y55 = FREE ()
!Y56 = FREE ()
!Y57 = FREE ()
!Y58 = FREE ()
!Y59 = FREE ()
!Y60 = FREE ()
!Y61 = FREE ()
!Y62 = FREE ()
!Y63 = FREE ()
!Y64 = FREE ()
!Y65 = FREE ()
!Y66 = FREE ()
!Y67 = FREE ()
!Y68 = FREE ()
!Y69 = FREE ()
!Y70 = FREE ()
$
!Y62A= FREE ()
!Y63A= FREE ()
!Y64A= FREE ()
!Y65A= FREE ()
!Y66A= FREE ()
!Y67A= FREE ()
!Y68A= FREE ()
!Y69A= FREE ()
$
!Y51B= FREE ()
!Y52B= FREE ()
!Y53B= FREE ()
!Y54B= FREE ()
!Y55B= FREE ()
!Y56B= FREE ()
!Y57B= FREE ()
!Y58B= FREE ()
!Y59B= FREE ()
!Y60B= FREE ()
$
!Y52C= FREE ()

```

```

!Y53C= FREE ()
!Y54C= FREE ()
!Y55C= FREE ()
!Y56C= FREE ()
!Y57C= FREE ()
!Y58C= FREE ()
!Y59C= FREE ()
$
!Z51 = FREE ()
!Z52 = FREE ()
!Z53 = FREE ()
!Z54 = FREE ()
!Z55 = FREE ()
!Z56 = FREE ()
!Z57 = FREE ()
!Z58 = FREE ()
!Z59 = FREE ()
!Z60 = FREE ()
!Z61 = FREE ()
!Z62 = FREE ()
!Z63 = FREE ()
!Z64 = FREE ()
!Z65 = FREE ()
!Z66 = FREE ()
!Z67 = FREE ()
!Z68 = FREE ()
!Z69 = FREE ()
!Z70 = FREE ()
$
*RETURN
*FRE6
$
$-----$
$ Subroutine (29 UFAC FREE)FRE7: This subroutine deletes the $
$                               coordinates for the upper outer $
$                               facesheet. $
$-----$
$
$(29 UFAC FREE)FRE7
$
!X61 = FREE ()
!X62 = FREE ()
!X63 = FREE ()
!X64 = FREE ()
!X65 = FREE ()
!X66 = FREE ()
!X67 = FREE ()
!X68 = FREE ()
!X69 = FREE ()
!X70 = FREE ()
!X71 = FREE ()
!X72 = FREE ()
!X73 = FREE ()
!X74 = FREE ()
!X75 = FREE ()
!X76 = FREE ()
!X77 = FREE ()
!X78 = FREE ()
!X79 = FREE ()

```



```

!X80 = FREE ()
$
!X61A= FREE ()
!X62A= FREE ()
!X63A= FREE ()
!X64A= FREE ()
!X65A= FREE ()
!X66A= FREE ()
!X67A= FREE ()
!X68A= FREE ()
!X69A= FREE ()
!X71A= FREE ()
!X72A= FREE ()
!X73A= FREE ()
!X74A= FREE ()
!X75A= FREE ()
!X76A= FREE ()
!X77A= FREE ()
!X78A= FREE ()
!X79A= FREE ()
$
!Y61 = FREE ()
!Y62 = FREE ()
!Y63 = FREE ()
!Y64 = FREE ()
!Y65 = FREE ()
!Y66 = FREE ()
!Y67 = FREE ()
!Y68 = FREE ()
!Y69 = FREE ()
!Y70 = FREE ()
!Y71 = FREE ()
!Y72 = FREE ()
!Y73 = FREE ()
!Y74 = FREE ()
!Y75 = FREE ()
!Y76 = FREE ()
!Y77 = FREE ()
!Y78 = FREE ()
!Y79 = FREE ()
!Y80 = FREE ()
$
!Y72A= FREE ()
!Y73A= FREE ()
!Y74A= FREE ()
!Y75A= FREE ()
!Y76A= FREE ()
!Y77A= FREE ()
!Y78A= FREE ()
!Y79A= FREE ()
$
!Y61B= FREE ()
!Y62B= FREE ()
!Y63B= FREE ()
!Y64B= FREE ()
!Y65B= FREE ()
!Y66B= FREE ()
!Y67B= FREE ()
!Y68B= FREE ()

```

```

!Y69B= FREE ()
!Y70B= FREE ()
$
!Y62C= FREE ()
!Y63C= FREE ()
!Y64C= FREE ()
!Y65C= FREE ()
!Y66C= FREE ()
!Y67C= FREE ()
!Y68C= FREE ()
!Y69C= FREE ()
$
!Z61 = FREE ()
!Z62 = FREE ()
!Z63 = FREE ()
!Z64 = FREE ()
!Z65 = FREE ()
!Z66 = FREE ()
!Z67 = FREE ()
!Z68 = FREE ()
!Z69 = FREE ()
!Z70 = FREE ()
!Z71 = FREE ()
!Z72 = FREE ()
!Z73 = FREE ()
!Z74 = FREE ()
!Z75 = FREE ()
!Z76 = FREE ()
!Z77 = FREE ()
!Z78 = FREE ()
!Z79 = FREE ()
!Z80 = FREE ()
$
*RETURN
*FRE7
$
$-----$
$ Subroutine (29 UINS FREE) UINS: This subroutine deletes the coor- $
$                               dinates of the insulation          $
$-----$
$
*(29 UINS FREE)UINS
$
!X1  = FREE ()
!X2  = FREE ()
!X3  = FREE ()
!X4  = FREE ()
!X5  = FREE ()
!X6  = FREE ()
!X7  = FREE ()
!X8  = FREE ()
!X9  = FREE ()
!X10 = FREE ()
!X11 = FREE ()
!X12 = FREE ()
!X13 = FREE ()
!X14 = FREE ()
!X15 = FREE ()
!X16 = FREE ()

```

```

!X17 = FREE ()
!X18 = FREE ()
!X19 = FREE ()
!X20 = FREE ()
$
!X1A = FREE ()
!X2A = FREE ()
!X3A = FREE ()
!X4A = FREE ()
!X5A = FREE ()
!X6A = FREE ()
!X7A = FREE ()
!X8A = FREE ()
!X9A = FREE ()
!X11A = FREE ()
!X12A = FREE ()
!X13A = FREE ()
!X14A = FREE ()
!X15A = FREE ()
!X16A = FREE ()
!X17A = FREE ()
!X18A = FREE ()
!X19A = FREE ()
$
!Y1 = FREE ()
!Y2 = FREE ()
!Y3 = FREE ()
!Y4 = FREE ()
!Y5 = FREE ()
!Y6 = FREE ()
!Y7 = FREE ()
!Y8 = FREE ()
!Y9 = FREE ()
!Y10 = FREE ()
!Y11 = FREE ()
!Y12 = FREE ()
!Y13 = FREE ()
!Y14 = FREE ()
!Y15 = FREE ()
!Y16 = FREE ()
!Y17 = FREE ()
!Y18 = FREE ()
!Y19 = FREE ()
!Y20 = FREE ()
$
!Y12A = FREE ()
!Y13A = FREE ()
!Y14A = FREE ()
!Y15A = FREE ()
!Y16A = FREE ()
!Y17A = FREE ()
!Y18A = FREE ()
!Y19A = FREE ()
$
!Y1B = FREE ()
!Y2B = FREE ()
!Y3B = FREE ()
!Y4B = FREE ()
!Y5B = FREE ()

```

```

!Y6B = FREE ()
!Y7B = FREE ()
!Y8B = FREE ()
!Y9B = FREE ()
!Y10B = FREE ()
$
!Y2C = FREE ()
!Y3C = FREE ()
!Y4C = FREE ()
!Y5C = FREE ()
!Y6C = FREE ()
!Y7C = FREE ()
!Y8C = FREE ()
!Y9C = FREE ()
$
!Z1 = FREE ()
!Z2 = FREE ()
!Z3 = FREE ()
!Z4 = FREE ()
!Z5 = FREE ()
!Z6 = FREE ()
!Z7 = FREE ()
!Z8 = FREE ()
!Z9 = FREE ()
!Z10 = FREE ()
!Z11 = FREE ()
!Z12 = FREE ()
!Z13 = FREE ()
!Z14 = FREE ()
!Z15 = FREE ()
!Z16 = FREE ()
!Z17 = FREE ()
!Z18 = FREE ()
!Z19 = FREE ()
!Z20 = FREE ()
$
*RETURN
*UINS
$
$-----$
$ Subroutine (29 ELEM GENE)ELEM: This subroutine creates the $
$                      elements for the mesh. $
$-----$
$
*(29 ELEM GENE)ELEM
$
*XQT AUS
$
TABLE(NI=1,NJ=1): K AREA : I = 1
                      J= 1: 1.0
$
TABLE(NI=1,NJ=1): K THIC : I = 1
                      J= 1: 1.0
$
*XQT ELD
RESET NUTED = 1
$
!NOPT= 1
$

```

```

!S1 = 1
!S2 = "S1"+"XJ"
!S3 = "S2"+"YJ"
!S4 = "S1"+"YJ"
$
!S5 = "S1" +("YJ"*"EY1")
!S6 = "S2" +("YJ"*"EY1")
!S7 = "S3" +("YJ"*"EY1")
!S8 = "S4" +("YJ"*"EY1")
$
!S9 = "S5" +("YJ"*"EY2")
!S10 = "S6" +("YJ"*"EY2")
!S11 = "S7" +("YJ"*"EY2")
!S12 = "S8" +("YJ"*"EY2")
$
!S13 = "S9" +("YJ"*"EY3")
!S14 = "S10"+("YJ"*"EY3")
!S15 = "S11"+("YJ"*"EY3")
!S16 = "S12"+("YJ"*"EY3")
$
!S17 = "S13"+("YJ"*"EY4")
!S18 = "S14"+("YJ"*"EY4")
!S19 = "S15"+("YJ"*"EY4")
!S20 = "S16"+("YJ"*"EY4")
$
!S21 = "S17"+("XJ"*("EX1"+"EX2"))
!S22 = "S21"+"XJ"
!S23 = "S22"+"YJ"
!S24 = "S21"+"YJ"
$
!S25 = "S21"+("XJ"*("EX3"+"EX4"))
!S26 = "S25"+"XJ"
!S27 = "S26"+"YJ"
!S28 = "S25"+"YJ"
$
!S29 = "S25"+("XJ"*("EX5"+"EX6"))
!S30 = "S29"+"XJ"
!S31 = "S30"+"YJ"
!S32 = "S29"+"YJ"
$
!S33 = "S29"+("XJ"*("EX7"+"EX8"))
!S34 = "S33"+"XJ"
!S35 = "S34"+"YJ"
!S36 = "S33"+"YJ"
$
!S37 = "S17"+("XJ"*"EX1")
!S38 = "S37"+"XJ"
!S39 = "S38"+"YJ"
!S40 = "S37"+"YJ"
$
!S41 = "S21"+("XJ"*"EX3")
!S42 = "S41"+"XJ"
!S43 = "S42"+"YJ"
!S44 = "S41"+"YJ"
$
!S45 = "S25"+("XJ"*"EX5")
!S46 = "S45"+"XJ"
!S47 = "S46"+"YJ"
!S48 = "S45"+"YJ"

```

```

$
!S49 = "S29"+"XJ"*"EX7"
!S50 = "S49"+"XJ"
!S51 = "S50"+"YJ"
!S52 = "S49"+"YJ"
$
!S53 = "S17"+"YJ"*"EY5"
!S54 = "S53"+"XJ"
!S55 = "S54"+"YJ"
!S56 = "S53"+"YJ"
$
!S57 = "S53"+"YJ"*"EY6"
!S58 = "S57"+"XJ"
!S59 = "S58"+"YJ"
!S60 = "S57"+"YJ"
$
!S61 = "S57"+"YJ"*"EY7"
!S62 = "S61"+"XJ"
!S63 = "S62"+"YJ"
!S64 = "S61"+"YJ"
$
!S65 = "S61"+"YJ"*"EY8"
!S66 = "S65"+"XJ"
!S67 = "S66"+"YJ"
!S68 = "S65"+"YJ"
$
!S69 = "S65"+"YJ"*"EY9"
!S70 = "S69"+"XJ"
!S71 = "S70"+"YJ"
!S72 = "S69"+"YJ"
$
!R69H= "S69"+"EX1"+"EX2"+"EX3"+"EX4"+"EX5"/2.0)
!R70H= "R69H"+"XJ"
!YJRH = ("EX5"/2.0)+"EX6"+"EX7"+"EX8"+"EX9"
!S69H= IFIX("R69H")
!S70H= IFIX("R70H")
!YJEH= IFIX("YJRH")
$
K21
$
$-----Element generator for the heater-----$
$
GROUP= 1
NMAT = 2
$
"S69H","S70H","NOPT","YJEH"
$
K41
$
$-----Element generator for the lower outer facesheet-----$
$
GROUP= 1
NMAT = 7
$
"S1","S2","S3","S4","NOPT","YJE","EY1"
$
$-----Element generator for the lower honeycomb-----$
$
GROUP= 2

```

```

NMAT = 5
$
"S5","S6","S7","S8","NOPT","YJE","EY2"
$
$-----Element generator for the lower skin-----$
$
GROUP= 3
NMAT = 1
$
"S9","S10","S11","S12","NOPT","YJE","EY3"
$
$-----Element generator for the lower torque tube-----$
$
GROUP= 4
NMAT = 6
$
"S13","S14","S15","S16","NOPT","YJE","EY4"
$
$-----Element generator for the airgaps-----$
$
GROUP= 5
NMAT = 4
$
"S17","S18","S19","S20","NOPT","EX1","EY5"
"S21","S22","S23","S24","NOPT","EX3","EY5"
"S25","S26","S27","S28","NOPT","EX5","EY5"
"S29","S30","S31","S32","NOPT","EX7","EY5"
"S33","S34","S35","S36","NOPT","EX9","EY5"
$
$-----Element generator for the torque tube ribs-----$
$
GROUP= 6
NMAT = 8
$
"S37","S38","S39","S40","NOPT","EX2","EY5"
"S41","S42","S43","S44","NOPT","EX4","EY5"
"S45","S46","S47","S48","NOPT","EX6","EY5"
"S49","S50","S51","S52","NOPT","EX8","EY5"
$
$-----Element generator for the upper torque tube-----$
$
GROUP= 7
NMAT = 6
$
"S53","S54","S55","S56","NOPT","YJE","EY6"
$
$-----Element generator for the upper skin-----$
$
GROUP= 8
NMAT = 1
$
"S57","S58","S59","S60","NOPT","YJE","EY7"
$
$-----Element generator for the upper honeycomb-----$
$
GROUP= 9
NMAT = 5
$
"S61","S62","S63","S64","NOPT","YJE","EY8"

```

```

$
$-----Element generator for the upper outer facesheet-----$
$
GROUP= 10
NMAT = 7
$
"S65","S66","S67","S68","NOPT","YJE","EY9"
$
$-----Element generator for the insulation-----$
$
GROUP= 11
NMAT = 3
$
"S69","S70","S71","S72","NOPT","YJE","EY10"
$
*XQT TGEO
RESET ASPECT = 50.0
$
*RETURN
*ELEM
$
$-----$
$ Subroutine (29 ELEM FREE)FRE8: This subroutine deletes all the $
$ starting node numbers used during $
$ the element generation process. $
$-----$
$
*(29 ELEM FREE)FRE8
$
!S1 = FREE ()
!S2 = FREE ()
!S3 = FREE ()
!S4 = FREE ()
$
!S5 = FREE ()
!S6 = FREE ()
!S7 = FREE ()
!S8 = FREE ()
$
!S9 = FREE ()
!S10 = FREE ()
!S11 = FREE ()
!S12 = FREE ()
$
!S13 = FREE ()
!S14 = FREE ()
!S15 = FREE ()
!S16 = FREE ()
$
!S17 = FREE ()
!S18 = FREE ()
!S19 = FREE ()
!S20 = FREE ()
$
!S21 = FREE ()
!S22 = FREE ()
!S23 = FREE ()
!S24 = FREE ()
$

```


!S25 = FREE ()
!S26 = FREE ()
!S27 = FREE ()
!S28 = FREE ()
\$
!S29 = FREE ()
!S30 = FREE ()
!S31 = FREE ()
!S32 = FREE ()
\$
!S33 = FREE ()
!S34 = FREE ()
!S35 = FREE ()
!S36 = FREE ()
\$
!S37 = FREE ()
!S38 = FREE ()
!S39 = FREE ()
!S40 = FREE ()
\$
!S41 = FREE ()
!S42 = FREE ()
!S43 = FREE ()
!S44 = FREE ()
\$
!S45 = FREE ()
!S46 = FREE ()
!S47 = FREE ()
!S48 = FREE ()
\$
!S49 = FREE ()
!S50 = FREE ()
!S51 = FREE ()
!S52 = FREE ()
\$
!S53 = FREE ()
!S54 = FREE ()
!S55 = FREE ()
!S56 = FREE ()
\$
!S57 = FREE ()
!S58 = FREE ()
!S59 = FREE ()
!S60 = FREE ()
\$
!S61 = FREE ()
!S62 = FREE ()
!S63 = FREE ()
!S64 = FREE ()
!S65 = FREE ()
!S66 = FREE ()
!S67 = FREE ()
!S68 = FREE ()
!S69 = FREE ()
!S70 = FREE ()
!S71 = FREE ()
!S72 = FREE ()
!S69H= FREE ()
!S70H= FREE ()

```

!R69H= FREE ( )
!R70H= FREE ( )
$
*RETURN
*FRE8
$
$-----$
$ Subroutine (29 BOUN COND)COND: This subroutine defines the $
$                               boundary conditions such as: $
$                               - the constant temperature $
$                               boundary $
$                               - the applied heat flux boundary $
$                               - the total experimental and $
$                               heating time $
$                               - the heated node boundary $
$                               temperature $
$-----$
$
*(29 BOUN COND)COND
$
*XQT AUS
$
DEFINE A = 1 INPT DAT1
DEFINE E = 1 INPT DAT2
$
$ in the following statements, the experimental time, temperature at the
$ constant temperature boundary, the applied heat flux, and the temperatures
$ at various locations are read from the pertinent datafiles.
$
TABLE(NI=1,NJ="NDS"):5 EXPE TIME 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE CTMP 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE FLUX 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT68 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT67 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT66 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT65 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=6,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT64 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=7,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE TT63 1 1
  TRANSFER(SOUR=A,ILIM=1,JLIM="NDS",SBASE=8,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB76 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=0,SSKIP=8,OPER=XSUM)
$

```

```

TABLE(NI=1,NJ="NDS"):5 EXPE BB75 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=1,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB73 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=2,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE BB74 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=3,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH24 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=4,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH23 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=5,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH22 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=6,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH21 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=7,SSKIP=8,OPER=XSUM)
$
TABLE(NI=1,NJ="NDS"):5 EXPE HH20 1 1
  TRANSFER(SOUR=E,ILIM=1,JLIM="NDS",SBASE=8,SSKIP=8,OPER=XSUM)
$
DEFINE B = 5 EXPE TIME 1 1
DEFINE C = 5 EXPE CTMP 1 1
DEFINE D = 5 EXPE FLUX 1 1
$
$ the rest of this subroutine is dedicated to insert an additional
$ timestep at the heater turn off point to overcome the discontinuity
$
!CNTR = 0
$
$ the loop reads the data and determines where the heater
$ is turned off, that is when the value of the heat flux
$ is recorded at less than 1.0 W/m2K.
$
*LABEL 1000
!CNTR = "CNTR" + 1
!HEAT = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
*IF("HEAT" LT 1.0):*GOTO 1100
*GOTO 1000
*LABEL 1100
$
!ACNT = "CNTR" - 1
!BCNT = "NDS" - "ACNT"
!CCNT = "ACNT" + 1
!NDS1 = "NDS" + 1
$
$ the following registers are the values for the applied heat flux,
$ the constant temperature boundary, the temperature at the heated
$ surface, as well as the heating time that will be added to the
$ tables to overcome the discontinuity due to heater turn off.
$ the values are taken at the timestep preceding heater turn off.
$
!FLUX = DS 1,"CNTR",1 (5 EXPE FLUX 1 1)
!CNTR = "CNTR" - 1
!TIMH = DS 1,"CNTR",1 (5 EXPE TIME 1 1)
!TIMF = DS 1,"NDS" ,1 (5 EXPE TIME 1 1)

```

```

!CTMP = DS 1,"CNTR",1 (5 EXPE CTMP 1 1)
!CSTP = DS 1, 1 ,1 (5 EXPE CTMP 1 1)
!DTIM = "TIMH" + 1.E-8
$
$ the following tables are created to hold the values for the
$ applied heat flux, the constant temperature boundary, the
$ temperature at the heated surface, the temperature at the
$ top of the torque tube, and the heating time, and
$ will be used in the TRANSFER command later on
$
TABLE(NI=1,NJ=1):5 DELT TIME:I=1:J=1:"DTIM"
TABLE(NI=1,NJ=1):5 DELT CTMP:I=1:J=1:"CTMP"
TABLE(NI=1,NJ=1):5 DELT FLUX:I=1:J=1:"FLUX"
$
DEFINE BB = 5 DELT TIME
DEFINE CC = 5 DELT CTMP
DEFINE DD = 5 DELT FLUX
$
$ the following table contains the information for the applied
$ heat flux.
$
TABLE(NI=1,NJ="NDS1"):5 APPL LOOP
TRANSFER(SOUR=C,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=CC,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=C,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
TABLE(NI=1,NJ="NDS1"):5 SOUR LOOP
TRANSFER(SOUR=D,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=DD,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=D,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
!YJYJ = "YJ"+"YJ"
!TL = "NDS1"*"YJEH"
!TN = "NDS1"*"YJYJ"
!L1 = 0
!L2 = 0
!L3 = 0
!L4 = 0
!L5 = 0
!L6 = 0
$
TABLE(NI=1,NJ="TL"):5 SOUR K21 1
TABLE(NI=1,NJ="TN"):5 APPL TEMP
$
*LABEL 720
!L1 = "L2"+1
!L2 = "L1"+"YJEH"-1
!L3 = "L3"+1
!FLU = DS 1,"L3",1 (5 SOUR LOOP)
$
TABLE,U(NI=1,NJ="TL"):5 SOUR K21 1
J="L1","L2":"FLU"
$
*IF("L3" LT "NDS1"):GOTO 720
$
$ the following table contains the times at which the heat flux
$ is applied to the structure
$
TABLE(NI=1,NJ="NDS1"):5 SOUR TIME

```

```

TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the information for the constant
$ temperature boundary, this simulates a heat sink on the bottom
$ surface and a convective boundary condition on the top of the
$ insulation
$
*LABEL 730
!L4   = "L5"+1
!L5   = "L4"+"YJYJ"-1
!L6   = "L6"+1
!APL  = DS 1,"L6",1 (5 APPL LOOP)
$
TABLE,U(NI=1,NJ="TN"):5 APPL TEMP
      J="L4","L5":"APL"
$
*IF("L6" LT "NDS1"):GOTO 730
$
$ the following table contains the times at which the constant
$ temperature is applied to the bottom surface
$
TABLE(NI=1,NJ="NDS1"):5 TEMP TIME
      TRANSFER(SOUR=B,ILIM=1,JLIM="ACNT",OPER=XSUM)
      TRANSFER(SOUR=BB,ILIM=1,JLIM=1,DBASE="ACNT",OPER=XSUM)
      TRANSFER(SOUR=B,ILIM=1,JLIM="BCNT",SBASE="ACNT",DBASE="CCNT",OPER=XSUM)
$
$ the following table contains the node numbers to which the constant
$ temperature is applied.
$
!ZJ   = "YJ" + 1
!NOIN = "TOTN"- "YJ"+1
$
TABLE(NI=1,NJ="YJYJ",TYPE=0):5 TEMP NODE
      DDATA 1
      RJ=1,"YJ":1
      DDATA 1
      RJ="ZJ","YJYJ":"NOIN"
$
$ a table of contents change is necessary to have NDS1 blocks instead
$ of NDS1 rows for the following two datasets since the number of blocks
$ needs to match the number of rows of corresponding TIME datasets.
$
TOCC (5 SOUR K21 1):NJ="YJEH",NINJ="YJEH"
TOCC (5 APPL TEMP) :NJ="YJYJ",NINJ="YJYJ"
$
!ACNT = FREE()
!BCNT = FREE()
!CCNT = FREE()
!NOIN = FREE()
!L1   = FREE()
!L2   = FREE()
!L3   = FREE()
!L4   = FREE()
!L5   = FREE()
!L6   = FREE()
!ZJ   = FREE()
!TL   = FREE()

```

```

!TN   = FREE()
!YJYJ = FREE()
!FLU  = FREE()
!APL  = FREE()
$
*RETURN
*COND
$
$-----$
$ Subroutine (29 PROB SPEC)SPEC: This subroutine defines variables $
$                               that are specific to the goal and $
$                               purpose of the program.             $
$-----$
$
*(29 PROB SPEC)SPEC
$
!TIMI = 0.0      $ Starting time of TRTB analysis (seconds)
!INTS = 0.5      $ Integrta time step for TRTB (seconds)
$
!MIN1 = 0.1      $ lower bound for skins (K)
!MIN2 = 0.2      $ lower bound for air gap (PCp)
!MIN3 = 0.05     $ lower bound for air gap (K)
!MIN4 = 0.05     $ lower bound for honeycomb (K)
!MIN5 = 0.1      $ lower bound for torque tubes (K)
!MIN6 = 0.1      $ lower bound for facesheets (K)
!MIN7 = 10.0     $
!MIN8 = 10.0     $
!MIN9 = 10.0     $
!MINX = 10.0     $
!MAX1 = 10.0     $ upper bound for skins (K)
!MAX2 = 3.0      $ upper bound for air gap (PCp)
!MAX3 = 3.0      $ upper bound for air gap (K)
!MAX4 = 4.0      $ upper bound for honeycomb (K)
!MAX5 = 10.0     $ upper bound for torque tubes (K)
!MAX6 = 10.0     $ upper bound for facesheets (K)
!MAX7 = 10.0     $
!MAX8 = 10.0     $
!MAX9 = 10.0     $
!MAXX = 10.0     $
$
!A1   = 1.6      $ initial value for skins (K)
!A2   = 1.0      $ initial value for air gap (PCp)
!A3   = 0.7      $ initial value for air gap (K)
!A4   = 0.4      $ initial value for honeycomb (K)
!A5   = 1.9      $ initial value for torque tubes (K)
!A6   = 1.6      $ initial value for facesheets (K)
!A7   = 1.00     $
!A8   = 1.00     $
!A9   = 1.00     $
!A10  = 1.00     $
$
!A1I  = 1.6      $ initial value for skins (K)
!A2I  = 1.0      $ initial value for air gap (PCp)
!A3I  = 0.7      $ initial value for air gap (K)
!A4I  = 0.4      $ initial value for honeycomb (K)
!A5I  = 1.9      $ initial value for torque tubes (K)
!A6I  = 1.6      $ initial value for facesheets (K)
!A7I  = 1.00     $
!A8I  = 1.00     $

```

```

!A9I   = 1.00      $
!A10I  = 1.00      $
$
*RETURN
*SPEC
$
$-----$
$ Subroutine (29 TRAN ANAL)ANAL: transient analysis of the problem $
$                               using the TRTB processor. The TRTB $
$                               processor is an implicit transient $
$                               analyzer that can be modified by   $
$                               changing the Crank Nicholson para- $
$                               meter.                             $
$-----$
$
$(29 TRAN ANAL)ANAL
$
*XQT TRTB                                $ implicit transient analyzer
  RESET QLIB = 5                         $ thermal excitation library
  RESET MLIB = 6                         $ material and fluids properties library
  RESET T1   = "TIMI"                    $ starting time of analysis
  RESET T2   = "TIMF"                    $ termination time of analysis
  RESET DT   = "INTS"                    $ integration time step size
  RESET DEST = 4                         $ output library
  RESET PRINT = 0                        $ online print control parameter
  RESET KPRT = 0                         $ print control parameter for k matrix
  RESET MXNDT = 10000000                 $ maximum # of int. time steps during TRTB
  RESET BETA = 0.5                       $ Crank Nicholson parameter
  RESET KTI  = 1                         $ K matrix computation time parameter
  KTIME      = 10000.                    $ times to calculate the conduction load vector
  TEMP       = "CSTP"                    $ initial temperature of structure at T1
  TSAVE      = 5 EXPE TIME 1 1           $ storage of nodal temperatures at time
$                                         $ increments of TSAVE
*RETURN
*ANAL
$
$-----$
$ Subroutine(29 TEMP TABL)TEMP: This subroutine puts the temperature$
$                               histories in a different table for $
$                               each different design, ie. for each $
$                               chromosome pair.                     $
$-----$
$
$(29 TEMP TABL)TEMP
$
*XQT AUS
ERASE 2
*XQT DCU                                $ database utility function processor
$
COPY 4 2 TRAN TEMP 1 1                  $ copy the dataset from lib 4 to lib 2
COPY 4 2 TRAN TIME 1 1                  $ copy the dataset from lib 4 to lib 2
$
*XQT AUS                                $ arithmetic utility processor
ERASE 4
$
!VA1 = TOC,NWDS(2 TRAN TIME)            $ number of words in dataset TRAN TIME
!VA2 = "BB1"-1
!VA3 = "BB2"-1
!VA4 = "BB3"-1

```

```

!VA5  = "BB4"-1
!VA6  = "BB5"-1
!VA7  = "TT1"-1
!VA8  = "TT2"-1
!VA9  = "TT3"-1
!VA10 = "TT4"-1
!VA11 = "TT5"-1
!VA12 = "HH1"-1
!VA13 = "HH2"-1
!VA14 = "HH3"-1
!VA15 = "HH4"-1
!VA16 = "HH5"-1
$
DEFINE X = 2 TRAN TEMP 1 1
DEFINE Y = 2 TRAN TIME 1 1
DEFINE Z = 5 EXPE HH20 1 1
$
*IF("N2" EQ 1) : *GOTO 5000 $ Temperature histories for top,bot,heat nodes
*IF("N2" EQ 2) : *GOTO 5010 $ Temperature histories for initial conditions
*IF("N2" EQ 3) : *GOTO 5020 $ Temperature histories for property 1
*IF("N2" EQ 4) : *GOTO 5030 $ Temperature histories for property 2
*IF("N2" EQ 5) : *GOTO 5040 $ Temperature histories for property 3
*IF("N2" EQ 6) : *GOTO 5050 $ Temperature histories for property 4
*IF("N2" EQ 7) : *GOTO 5060 $ Temperature histories for property 5
*IF("N2" EQ 8) : *GOTO 5070 $ Temperature histories for property 6
*IF("N2" EQ 9) : *GOTO 5080 $ Temperature histories for property 7
*IF("N2" EQ 10) : *GOTO 5082 $ Temperature histories for property 8
*IF("N2" EQ 11) : *GOTO 5084 $ Temperature histories for property 9
*IF("N2" EQ 12) : *GOTO 5086 $ Temperature histories for property 10
$
*LABEL 5000
$
TABLE(NI=3,NJ="VA1"):9 TEMP TABL
  TRAN(SOUR=Y,ILIM=1,JLIM="VA1",DBASE=0,DSKIP=2)
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA11",DBASE=1,DSKIP=2)
  TRAN(SOUR=Z,ILIM=1,JLIM="VA1",DBASE=2,DSKIP=2)
$
TABLE(NI=1,NJ="VA1"):2 TT68 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA11")
TABLE(NI=1,NJ="VA1"):2 TT67 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA10")
TABLE(NI=1,NJ="VA1"):2 TT66 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA9")
TABLE(NI=1,NJ="VA1"):2 TT65 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA8")
TABLE(NI=1,NJ="VA1"):2 TT64 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA7")
$
TABLE(NI=1,NJ="VA1"):2 BB76 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA5")
TABLE(NI=1,NJ="VA1"):2 BB75 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA4")
TABLE(NI=1,NJ="VA1"):2 BB73 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA3")
TABLE(NI=1,NJ="VA1"):2 BB74 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA2")
$
TABLE(NI=1,NJ="VA1"):2 HH24 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA16")

```



```

TABLE(NI=1,NJ="VA1"):2 HH23 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA15")
TABLE(NI=1,NJ="VA1"):2 HH22 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA14")
TABLE(NI=1,NJ="VA1"):2 HH21 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA13")
TABLE(NI=1,NJ="VA1"):2 HH20 NODE
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
$
*GOTO 6000
$
*LABEL 5010
TABLE(NI=1,NJ="VA1"):2 INIT 00 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5020
TABLE(NI=1,NJ="VA1"):2 PROP I 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5030
TABLE(NI=1,NJ="VA1"):2 PROP II 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5040
TABLE(NI=1,NJ="VA1"):2 PROP III 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5050
TABLE(NI=1,NJ="VA1"):2 PROP IV 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5060
TABLE(NI=1,NJ="VA1"):2 PROP V 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5070
TABLE(NI=1,NJ="VA1"):2 PROP VI 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5080
TABLE(NI=1,NJ="VA1"):2 PROP VII 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5082
TABLE(NI=1,NJ="VA1"):2 PROP VIII 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 5084
TABLE(NI=1,NJ="VA1"):2 PROP IX 1 1
  TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")

```

```

*GOTO 6000
$
*LABEL 5086
TABLE(NI=1,NJ="VA1"):2 PROP X      1 1
      TRAN(SOUR=X,ILIM=1,JLIM=1,SBASE="VA12")
*GOTO 6000
$
*LABEL 6000
*RETURN
*TEMP
$
$-----$
$ Subroutine (29 SENS COEF)COEF: This subroutine calculates $
$                               the sensitivity coefficients $
$                               for up to seven parameters. $
$-----$
$
*(29 SENS COEF)COEF
$
*XQT AUS                      $ Arithmetic Utility System
$
INLIB = 1                      $ Read from library 1
OUTLI = 1                      $ Write to library 1
$
*IF("N2" EQ 0) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 1) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 2) :*GOTO 1000    $ No perturbations
*IF("N2" EQ 3) :*GOTO 1010    $ Property 1  is perturbed
*IF("N2" EQ 4) :*GOTO 1020    $ Property 2  is perturbed
*IF("N2" EQ 5) :*GOTO 1030    $ Property 3  is perturbed
*IF("N2" EQ 6) :*GOTO 1040    $ Property 4  is perturbed
*IF("N2" EQ 7) :*GOTO 1050    $ Property 5  is perturbed
*IF("N2" EQ 8) :*GOTO 1060    $ Property 6  is perturbed
*IF("N2" EQ 9) :*GOTO 1070    $ Property 7  is perturbed
*IF("N2" EQ 10):*GOTO 1080    $ Property 8  is perturbed
*IF("N2" EQ 11):*GOTO 1090    $ Property 9  is perturbed
*IF("N2" EQ 12):*GOTO 1100    $ Property 10 is perturbed
*GOTO 2000
$
*LABEL 1000
$
!A1  = "A1I"                  $ Set property 1  to initial guess
!A2  = "A2I"                  $ Set property 2  to initial guess
!A3  = "A3I"                  $ Set property 3  to initial guess
!A4  = "A4I"                  $ Set property 4  to initial guess
!A5  = "A5I"                  $ Set property 5  to initial guess
!A6  = "A6I"                  $ Set property 6  to initial guess
!A7  = "A7I"                  $ Set property 7  to initial guess
!A8  = "A8I"                  $ Set property 8  to initial guess
!A9  = "A9I"                  $ Set property 9  to initial guess
!A10 = "A10I"                 $ Set property 10 to initial guess
$
*DCALL(29 MATE PROP)          $ Update the thermal properties
*DCALL(29 TRAN ANAL)          $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)          $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 1010
$

```

!A1 = 1.001*"A1I"	\$ Perturb property 1
!DA1 = 0.001*"A1I"	\$ Amount property 1 is perturbed
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1020	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = 1.001*"A2I"	\$ Perturb property 2
!DA2 = 0.001*"A2I"	\$ Amount property 2 is perturbed
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1030	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = 1.001*"A3I"	\$ Perturb property 3
!DA3 = 0.001*"A3I"	\$ Amount property 3 is perturbed
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1040	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess

!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = 1.001*"A4I"	\$ Perturb property 4
!DA4 = 0.001*"A4I"	\$ Amount property 4 is perturbed
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures)
\$	
*GOTO 2000	
*LABEL 1050	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = 1.001*"A5I"	\$ Perturb property 5
!DA5 = 0.001*"A5I"	\$ Amount property 5 is perturbed
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1060	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = 1.001*"A6I"	\$ Perturb property 6
!DA6 = 0.001*"A6I"	\$ Amount property 6 is perturbed
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1070	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess

!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = 1.001*"A7I"	\$ Perturb property 7
!DA7 = 0.001*"A7I"	\$ Amount property 7 is perturbed
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1080	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = 1.001*"A8I"	\$ Perturb property 8
!DA8 = 0.001*"A8I"	\$ Amount property 8 is perturbed
!A9 = "A9I"	\$ Set property 9 to initial guess
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1090	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess
!A7 = "A7I"	\$ Set property 7 to initial guess
!A8 = "A8I"	\$ Set property 8 to initial guess
!A9 = 1.001*"A9I"	\$ Perturb property 9
!DA9 = 0.001*"A9I"	\$ Amount property 9 is perturbed
!A10 = "A10I"	\$ Set property 10 to initial guess
\$	
*DCALL(29 MATE PROP)	\$ Update the thermal properties
*DCALL(29 TRAN ANAL)	\$ Calculate the temperature distributions
*DCALL(29 TEMP TABL)	\$ Tabularize specific nodal temperatures
\$	
*GOTO 2000	
*LABEL 1100	
\$	
!A1 = "A1I"	\$ Set property 1 to initial guess
!A2 = "A2I"	\$ Set property 2 to initial guess
!A3 = "A3I"	\$ Set property 3 to initial guess
!A4 = "A4I"	\$ Set property 4 to initial guess
!A5 = "A5I"	\$ Set property 5 to initial guess
!A6 = "A6I"	\$ Set property 6 to initial guess

```

!A7  = "A7I"           $ Set property 7  to initial guess
!A8  = "A8I"           $ Set property 8  to initial guess
!A9  = "A9I"           $ Set property 9  to initial guess
!A10 = 1.001*"A10I"    $ Perturb property 10
!DA10 = 0.001*"A10I"   $ Amount property 10 is perturbed
$
*DCALL(29 MATE PROP)    $ Update the thermal properties
*DCALL(29 TRAN ANAL)    $ Calculate the temperature distributions
*DCALL(29 TEMP TABL)    $ Tabularize specific nodal temperatures
$
*GOTO 2000
*LABEL 2000
$
*IF("N2" EQ 0) :*GOTO 3000 $ No sensitivity coefficients are calculated
*IF("N2" EQ 1) :*GOTO 3000 $ No sensitivity coefficients are calculated
*IF("N2" EQ 2) :*GOTO 3000 $ No sensitivity coefficients are calculated
*IF("N2" EQ 3) :*GOTO 3010 $ Sensitivity coefficients for property 1
*IF("N2" EQ 4) :*GOTO 3020 $ Sensitivity coefficients for property 1,2
*IF("N2" EQ 5) :*GOTO 3030 $ Sensitivity coefficients for property 1,2,3
*IF("N2" EQ 6) :*GOTO 3040 $ Sensitivity coefficients for property 1,2,3,4
*IF("N2" EQ 7) :*GOTO 3050 $ Sensitivity coefficients for property 1,2,3,4,5
*IF("N2" EQ 8) :*GOTO 3060 $ Sensitivity coefficients for property 1,2,3,4,5,6
*IF("N2" EQ 9) :*GOTO 3070 $ Sensitivity coefficients for property 1,2,3,4,5,6,7
*IF("N2" EQ 10):*GOTO 3080 $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8
*IF("N2" EQ 11):*GOTO 3090 $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9
*IF("N2" EQ 12):*GOTO 3100 $ Sensitivity coefficients for property 1,2,3,4,5,6,7,8,9,10
*GOTO 3000
$
*LABEL 3000
DEFINE T0 = 2 INIT 00 1 1 $ Initial temperature distribution with no perturbations
*GOTO 4000
$
*LABEL 3010
!S1P0 = 1.0/"DA1"        $ Positive denominator used to numerically differentiate
!S1NE = -1.0/"DA1"       $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T1 = 2 PROP I 1 1  $ Temperature distribution for perturbed property 1
DIF1 = SUM("S1P0" T1,"S1NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3020
!S2P0 = 1.0/"DA2"        $ Positive denominator used to numerically differentiate
!S2NE = -1.0/"DA2"       $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T2 = 2 PROP II 1 1 $ Temperature distribution for perturbed property 2
DIF2 = SUM("S2P0" T2,"S2NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3030
!S3P0 = 1.0/"DA3"        $ Positive denominator used to numerically differentiate
!S3NE = -1.0/"DA3"       $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T3 = 2 PROP III 1 1 $ Temperature distribution for perturbed property 3
DIF3 = SUM("S3P0" T3,"S3NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3040
!S4P0 = 1.0/"DA4"        $ Positive denominator used to numerically differentiate
!S4NE = -1.0/"DA4"       $ Negative denominator used to numerically differentiate

```

```

DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T4 = 2 PROP IV 1 1 $ Temperature distribution for perturbed property 4
DIF4 = SUM("S4P0" T4,"S4NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3050
!S5P0 = 1.0/"DA5" $ Positive denominator used to numerically differentiate
!S5NE = -1.0/"DA5" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T5 = 2 PROP V 1 1 $ Temperature distribution for perturbed property 5
DIF5 = SUM("S5P0" T5,"S5NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3060
!S6P0 = 1.0/"DA6" $ Positive denominator used to numerically differentiate
!S6NE = -1.0/"DA6" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T6 = 2 PROP VI 1 1 $ Temperature distribution for perturbed property 6
DIF6 = SUM("S6P0" T6,"S6NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3070
!S7P0 = 1.0/"DA7" $ Positive denominator used to numerically differentiate
!S7NE = -1.0/"DA7" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T7 = 2 PROP VII 1 1 $ Temperature distribution for perturbed property 7
DIF7 = SUM("S7P0" T7,"S7NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3080
!S8P0 = 1.0/"DA8" $ Positive denominator used to numerically differentiate
!S8NE = -1.0/"DA8" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T8 = 2 PROP VIII 1 1 $ Temperature distribution for perturbed property 8
DIF8 = SUM("S8P0" T8,"S8NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3090
!S9P0 = 1.0/"DA9" $ Positive denominator used to numerically differentiate
!S9NE = -1.0/"DA9" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T9 = 2 PROP IX 1 1 $ Temperature distribution for perturbed property 9
DIF9 = SUM("S9P0" T9,"S9NE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 3100
!SXPO = 1.0/"DA10" $ Positive denominator used to numerically differentiate
!SXNE = -1.0/"DA10" $ Negative denominator used to numerically differentiate
DEFINE T0 = 2 INIT 00 1 1 $ Temperature distribution for no perturbations
DEFINE T10 = 2 PROP X 1 1 $ Temperature distribution for perturbed property 10
DIFX = SUM("SXPO" T10,"SXNE" T0) $Numerical differentiation of numerator
*GOTO 4000
$
*LABEL 4000
!WORD = TOC,NJ(2 INIT 00 1 1) $ Determine the number of lines in INIT 00
$
*IF("NP" EQ 0) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 1) :*GOTO 5000 $ Exit this subroutine
*IF("NP" EQ 2) :*GOTO 5000 $ Exit this subroutine

```

```

*IF("NP" EQ 3) : *GOTO 6010      $ Calculate XTX, P11
*IF("NP" EQ 4) : *GOTO 6020      $ Calculate XTX, P11,P22
*IF("NP" EQ 5) : *GOTO 6030      $ Calculate XTX, P11,P22,P33
*IF("NP" EQ 6) : *GOTO 6040      $ Calculate XTX, P11,P22,P33,P44
*IF("NP" EQ 7) : *GOTO 6050      $ Calculate XTX, P11,P22,P33,P44,P55
*IF("NP" EQ 8) : *GOTO 6060      $ Calculate XTX, P11,P22,P33,P44,P55,P66
*IF("NP" EQ 9) : *GOTO 6070      $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77
*IF("NP" EQ 10) : *GOTO 6080     $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88
*IF("NP" EQ 11) : *GOTO 6090     $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99
*IF("NP" EQ 12) : *GOTO 6100     $ Calculate XTX, P11,P22,P33,P44,P55,P66,P77,P88,P99,PXX
$
*LABEL 6010
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=2,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
TABLE(NI=1,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=0)
$
DEFINE S = 1 PMAT TOBE 1 1      $ Define table as register (X)
ST      = RTRAN(S)              $ XT
STS     = RPROD(S,ST)           $ XTX
STSI    = RINV(STS)             $ XTXI
!P11    = DS 1,1,1 (1 STSI AUS 1 1) $ numerical value of diagonal 1
*GOTO 5000
$
*LABEL 6020
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=3,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
TABLE(NI=2,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=1)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=1)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6030
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=4,NJ="WORD"):1 SENS COEF 1 1
  TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
  TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
  TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
TABLE(NI=3,NJ="WORD"):1 PMAT TOBE 1 1
  TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=2)

```



```

TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=2)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=2)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6040
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=5,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
TABLE(NI=4,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=3)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=3)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=3)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=3)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6050
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=6,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
TABLE(NI=5,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=4)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=4)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=4)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=4)

```

```

TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=4)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6060
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=7,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
TABLE(NI=6,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=5)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=5)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=5)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=5)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=5)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=5)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6070
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
TABLE(NI=8,NJ="WORD"):1 SENS COEF 1 1

```

```

TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
TABLE(NI=7,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=6)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=6)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=6)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=6)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=6)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=6)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=6)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6080
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=9,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
TABLE(NI=8,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=7)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=7)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=7)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=7)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=7)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=7)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=7)

```

```

TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=7)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)
!P66    = DS 6,6,1 (1 STSI AUS 1 1)
!P77    = DS 7,7,1 (1 STSI AUS 1 1)
!P88    = DS 8,8,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6090
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=10,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
TABLE(NI=9,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=8)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=8)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=8)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=8)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=8)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=8)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=8)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=8)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=8)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1,1,1 (1 STSI AUS 1 1)
!P22    = DS 2,2,1 (1 STSI AUS 1 1)
!P33    = DS 3,3,1 (1 STSI AUS 1 1)
!P44    = DS 4,4,1 (1 STSI AUS 1 1)
!P55    = DS 5,5,1 (1 STSI AUS 1 1)

```

```

!P66      = DS 6,6,1 (1 STSI AUS 1 1)
!P77      = DS 7,7,1 (1 STSI AUS 1 1)
!P88      = DS 8,8,1 (1 STSI AUS 1 1)
!P99      = DS 9,9,1 (1 STSI AUS 1 1)
*GOTO 5000
$
*LABEL 6100
DEFINE TR1 = 1 DIF1 AUS 1 1
DEFINE TR2 = 1 DIF2 AUS 1 1
DEFINE TR3 = 1 DIF3 AUS 1 1
DEFINE TR4 = 1 DIF4 AUS 1 1
DEFINE TR5 = 1 DIF5 AUS 1 1
DEFINE TR6 = 1 DIF6 AUS 1 1
DEFINE TR7 = 1 DIF7 AUS 1 1
DEFINE TR8 = 1 DIF8 AUS 1 1
DEFINE TR9 = 1 DIF9 AUS 1 1
DEFINE TR10 = 1 DIFX AUS 1 1
DEFINE TR11 = 2 TRAN TIME 1 1
$
TABLE(NI=11,NJ="WORD"):1 SENS COEF 1 1
TRAN(SOUR=TR11,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=10)
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=10)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=10)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=10)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=10)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=10)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=10)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=10)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=10)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=10)
TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=10,DSKIP=10)
TABLE(NI=10,NJ="WORD"):1 PMAT TOBE 1 1
TRAN(SOUR=TR1,ILIM=1,JLIM="WORD",DBASE=0,DSKIP=9)
TRAN(SOUR=TR2,ILIM=1,JLIM="WORD",DBASE=1,DSKIP=9)
TRAN(SOUR=TR3,ILIM=1,JLIM="WORD",DBASE=2,DSKIP=9)
TRAN(SOUR=TR4,ILIM=1,JLIM="WORD",DBASE=3,DSKIP=9)
TRAN(SOUR=TR5,ILIM=1,JLIM="WORD",DBASE=4,DSKIP=9)
TRAN(SOUR=TR6,ILIM=1,JLIM="WORD",DBASE=5,DSKIP=9)
TRAN(SOUR=TR7,ILIM=1,JLIM="WORD",DBASE=6,DSKIP=9)
TRAN(SOUR=TR8,ILIM=1,JLIM="WORD",DBASE=7,DSKIP=9)
TRAN(SOUR=TR9,ILIM=1,JLIM="WORD",DBASE=8,DSKIP=9)
TRAN(SOUR=TR10,ILIM=1,JLIM="WORD",DBASE=9,DSKIP=9)
$
DEFINE S = 1 PMAT TOBE 1 1
ST      = RTRAN(S)
STS     = RPROD(S,ST)
STSI    = RINV(STS)
!P11    = DS 1 ,1 ,1 (1 STSI AUS 1 1)
!P22    = DS 2 ,2 ,1 (1 STSI AUS 1 1)
!P33    = DS 3 ,3 ,1 (1 STSI AUS 1 1)
!P44    = DS 4 ,4 ,1 (1 STSI AUS 1 1)
!P55    = DS 5 ,5 ,1 (1 STSI AUS 1 1)
!P66    = DS 6 ,6 ,1 (1 STSI AUS 1 1)
!P77    = DS 7 ,7 ,1 (1 STSI AUS 1 1)
!P88    = DS 8 ,8 ,1 (1 STSI AUS 1 1)
!P99    = DS 9 ,9 ,1 (1 STSI AUS 1 1)
!PXX    = DS 10,10,1 (1 STSI AUS 1 1)
$
*GOTO 5000

```

```

*LABEL 5000
$
*RETURN
*COEF
$
$-----$
$ Subroutine (29 SENS FIND): this subroutine determines which nodes corres $
$                                pond best to the sensor locations on the test $
$                                panel. As the mesh is refined, the node number $
$                                can change without affecting the rest of the $
$                                program. $
$-----$
$
*(29 SENS FIND)FIND
$
*XQT AUS
$
!ROW0 = 0
!ROW1 = "YJ"*("NY1"+"EY2"+"EY3")+1
!ROW2 = "ROW1"+"YJ"
$
*LABEL 10
$
!RED = 0
$
!ROW0 = "ROW0" + 1
$
*IF("ROW0" EQ "ROW1"):*GOTO 30
$
*GOTO 10
$
*LABEL 30
$
!XCOR = DS 1,"ROW0",1 (1 JLOC BTAB)
$
!XDF1 = "XCOR" - 0.0565
!XDF2 = "XCOR" - 0.1835
!XDF3 = "XCOR" - 0.2597
!XDF4 = "XCOR" - 0.3740
$
*IF("XDF4" GT 0.0):*GOTO 40
*IF("RED" EQ 1) :*GOTO 100
*IF("XDF3" GT 0.0):*GOTO 50
*IF("RED" EQ 2) :*GOTO 100
*IF("XDF2" GT 0.0):*GOTO 60
*IF("RED" EQ 3) :*GOTO 100
*IF("XDF1" GT 0.0):*GOTO 70
*GOTO 100
$
*LABEL 40
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF4")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 42
*GOTO 44
*LABEL 42
!N44 = "ROW7"
*GOTO 80

```

```

*LABEL 44
!N44  = "ROW0"
*GOTO 80
$
*LABEL 50
!RED  = 1
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF3")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 52
*GOTO 54
*LABEL 52
!N33  = "ROW7"
*GOTO 100
*LABEL 54
!N33  = "ROW0"
*GOTO 100
$
*LABEL 60
!RED  = 2
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF2")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 62
*GOTO 64
*LABEL 62
!N22  = "ROW7"
*GOTO 100
*LABEL 64
!N22  = "ROW0"
*GOTO 100
$
*LABEL 70
!RED  = 3
!ROW7 = "ROW0" - 1
!XPRES = DS 1,"ROW7",1 (1 JLOC BTAB)
!ABS1 = ABS("XDF1")
!ABS2 = ABS("XPRES")
*IF("ABS1" GT "ABS2"):GOTO 72
*GOTO 74
*LABEL 72
!N11  = "ROW7"
*GOTO 100
*LABEL 74
!N11  = "ROW0"
*GOTO 100
$
*LABEL 80
!BB1 = "ROW2"-1
!BB2 = "N44"
!BB3 = "N33"
!BB4 = "N22"
!BB5 = "N11"
!TT1 = "BB1"+"(YJ"*"EY4")
!TT2 = "BB2"+"(YJ"*"EY4")
!TT3 = "BB3"+"(YJ"*"EY4")
!TT4 = "BB4"+"(YJ"*"EY4")

```

```

!TT5 = "BB5"+"(YJ"*"EY4")
!HH1 = "TT1"+"(YJ"*("EY5"+"EY6"+"EY7"+"EY8"))
!HH2 = "TT2"+"(YJ"*("EY5"+"EY6"+"EY7"+"EY8"))
!HH3 = "TT3"+"(YJ"*("EY5"+"EY6"+"EY7"+"EY8"))
!HH4 = "TT4"+"(YJ"*("EY5"+"EY6"+"EY7"+"EY8"))
!HH5 = "TT5"+"(YJ"*("EY5"+"EY6"+"EY7"+"EY8"))
$
*GOTO 200
$
*LABEL 100
!ROW0 = "ROW0" + 1
$
*IF("ROW0" EQ "ROW2"): *GOTO 10
*GOTO 30
$
*LABEL 200
$
!RED = FREE()
!ABS1 = FREE()
!ABS2 = FREE()
!XPRES = FREE()
!XCOR = FREE()
!XDF1 = FREE()
!XDF2 = FREE()
!XDF3 = FREE()
!XDF4 = FREE()
!ROW0 = FREE()
!ROW1 = FREE()
!ROW2 = FREE()
!ROW7 = FREE()
!N11 = FREE()
!N22 = FREE()
!N33 = FREE()
!N44 = FREE()
$
*RETURN
*FIND
$
$-----$
$ This is the main program. All the subroutines must have been $
$ defined before they are called by the main program, therefore the $
$ main program needs to be placed at the end. $
$-----$
$
*XQT AUS $arithmetic utility system processor
$
$the following three comment statemnets can not be put at the end
$of the *TF commands, since it intervenes with the runstream.
$
$open the file containing the data
$read the datafile
$close the datafile
$
*TF OPEN 1'data1
*TF READ 1
*TF CLOSE 1
*TF OPEN 1'data2
*TF READ 1
*TF CLOSE 1

```



```

$
!NP = 0
!N2 = 1
$ counter to direct TRAN TEMP
$
*DCALL (29 MAKE VARI)      $ call subroutine MAKE VARI
*DCALL (29 PROB SPEC)     $ call subroutine PROB SPEC
*DCALL (29 MATE PROP)     $ call subroutine MATE PROP
*DCALL (29 NODE GENE)     $ call subroutine NODE GENE
*DCALL (29 ELEM GENE)     $ call subroutine MAKE ELEM
*DCALL (29 ELEM FREE)     $ call subroutine ELEM FREE
*DCALL (29 SENS FIND)     $ call subroutine SENS FIND
*DCALL (29 BOUN COND)     $ call subroutine BOUN COND
*DCALL (29 TRAN ANAL)     $ call subroutine TRAN ANAL
*DCALL (29 TEMP TABL)     $ call subroutine TEMP TABL
$
!NP = 0
!N2 = 2
$ initial conditions
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
!NP = 0
!N2 = 3
$ parameter 1
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
!NP = 0
!N2 = 4
$ parameter 2
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
!NP = 0
!N2 = 5
$ parameter 3
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
!NP = 0
!N2 = 6
$ parameter 4
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
!NP = 0
!N2 = 7
$ parameter 5
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
!NP = 8
!N2 = 8
$ parameter 6
*DCALL (29 SENS COEF)     $ call subroutine SENS COEF
$
!BB1
!BB2
!BB3
!BB4
!BB5
!HH1
!HH2
!HH3
!HH4
!HH5
!TT1
!TT2
!TT3
!TT4
!TT5
$
*XQT DCU
PRINT 1 SENS COEF
*XQT EXIT

```

Vita

The author, Alexander R. Hanuska was born on January 7th, 1974 in Bamberg (Germany), where he was raised by his parents Viktor and Mireille Hanuska. After six years spent in Bavaria, he moved to Karlsruhe (Germany) where he was admitted in the first grade of the French section of the European School of Karlsruhe in 1980. In 1990 he moved to South Carolina, and finished his junior and senior year in the French Michelin Highschool of Greenville. His education continued at Virginia Polytechnic Institute and State University in Blacksburg, Virginia, where he received his Bachelor of Science degree in Mechanical Engineering in May of 1996. He gained job experience during his two co-op assignments at BWNT (Framatome) in Lynchburg, Virginia. Following his graduation, Alexander enrolled in the graduate program at Virginia Tech, under the direction of Dr. Elaine P. Scott in the department of Mechanical Engineering. On August 9th, 1997, he married Carol A. Steichen who is also pursuing her graduate studies in the Mechanical Engineering department at Virginia Tech.

Permanent Address: 911 University City Boulevard Apartment A-15
Blacksburg, VA 24060

This thesis was typeset with $\text{\LaTeX 2}_{\epsilon}$ ¹ by the author.

¹ $\text{\LaTeX 2}_{\epsilon}$ is an extension of \LaTeX . \LaTeX is a collection of macros for \TeX . \TeX is a trademark of the American Mathematical Society. The macros used in formatting this thesis were written by Greg Walker, Department of Mechanical Engineering, Virginia Tech.