# STRUCTURAL ANALYSIS MODELS FOR BLOCK PALLETS

by

Robert G. Colclough

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

Forest Products

APPROVED:

Dr. T. E. McLain, Chairman

Dr. M. S. White

Dr. J. R. Loferski

November 25, 1987

Blacksburg, Virginia

# STRUCTURAL ANALYSIS MODELS
# FOR BLOCK PALLETS

by

Robert G. Colclough

Dr. T. E. McLain, Chairman

Forest Products

(ABSTRACT)

A large percentage of the total annual lumber production in the U.S.A is consumed by the pallet industry. However, standardized design procedures for these products have only recently been developed. A four-year cooperative pallet research, development and application program was undertaken by the National Wooden Pallet and Container Association, Virginia Polytechnic Institute and State University, and the U.S Forest Service. This research is directed towards developing standardized design procedures for both stringer and block-type pallets. Phase I dealt exclusively with stringer-type pallets while Phase II expands the scope to include block-type pallets. The objective of this work was to develop methods to analyze the effects of loads, supports and geometry on the response of block-style pallets.

The developed analysis procedures are based on matrix structural analysis methods. A quarter symmetric 3-dimensional model is used to simulate pallets racked across the stringerboards (RAS) and a half symmetric 2-dimensional model is used for the racked across deckboards (RAD) and sling support modes. Both models are used in the stack

condition. Deckboard/stringerboard joints are modeled as a single spring in the RAS model and the deck-block joint in both the RAS and RAD models are modeled as a framework of rigidly connected members and five springs (2 rotational and three axial). The procedure has the capability to handle both uniformly distributed and line loads in rack, stack, or sling support modes, and a wide variety of commonly used geometries.

The developed analysis methodology is presented in computerized form and will provide the user a means of communication with pallet manufacturers for defining expected performance.

# ACKNOWLEDGEMENTS

The author would like to acknowledge his major professor Dr. T. E. McLain for his endless encouragement and help throughout the program. The author also wishes to thank his graduate advisory committee members, Dr. M. S. White and Dr. J. R. Loferski for their encouragement and guidance.

Special thanks is also given the author's colleagues, John McLeod and Sandhya Samarasinghe, for their help throughout the program. For their continual assistance throughout the experimental phase of the project, the author would also like to extend his thanks to the laboratory technicians J.W. Akers and Kenny Albert.

The author thanks his family for their moral support throughout his studies.

Finally, the author thanks the Cooperative Pallet Research Program funded by Virginia Tech, and the NWPCA for their financial support throughout the project.

# TABLE OF CONTENTS:

Chapter Page

# LIST OF ILLUSTRATIONS:

Page

# 1.0 INTRODUCTION

## 1.1 Overview

At present, the pallet industry is the second largest consumer of lumber in the U.S.A accounting for approximately 12-18% of all lumber cut in the United States (Mc Leod,1985). Development of engineering and design procedures for these components can result in more effective and efficient use of wood. This can directly benefit consumers and provide them with a more reliable and cost-effective product.

Until recently, there were no standardised pallet design procedures which rationally accounted for the influence of design variables on pallet performance. However, a four - year cooperative pallet research, development and application program, undertaken by the National Wooden Pallet and Container Association, Virginia Polytechnic Institute and State University, and the U.S Forest service, resulted in reliability - based design procedures for wooden stringer -type pallets. These procedures, known as the Pallet Design System (P.D.S), (NWPCA copyright) provide the consumer a better means of communication with the manufacturer by enabling him to specify performance of the finished product.

Stringer - type pallets, account for the bulk of the pallet industry production in the U.S. However, about 20% of all pallets produced are the block - type which are used extensively in the beer, canning and chemical industries. However, this is not the case in Europe where the block design is preferred and is more extensively used.

The major objective of Phase II of the cooperative program is the development of design procedures for block - type pallets similar to those for stringer - type pallets. This thesis details the development of the strength and stiffness design components for block-type pallets.

## 1.2 Study objectives

The principal objective of the study was to develop and verify analysis models of the actions of block - type wood pallets subject to typical unit loads and support conditions.
Necessary sub - objectives are :-

(1) To develop and verify appropriate three-dimensional SPACEPAL models of block pallet behaviour which adequately predict the effects of loading and support conditions on pallet performance.

(2) From the results of objective (1), a) conduct sensitivity investigations to determine important model parameters and level of required input accuracy, and b) develop and verify simplified models for use in a personal computer format.

## 1.3 Why block pallets ?

Wooden pallets play an important role in the fabrication, storage, and transportation of goods in the form of unit loads. Most of the pallets produced in the U.S. are of the stringer-type design. However, block designs are also used, especially for single use, non-returnable pallets. In other parts of the world block pallets are very common. "This difference can be explained by the fact that American pallets are fabricated mostly of dense hardwoods and relatively dense softwoods, which are assembled with slender and often hardened-steel pallet nails. These nails reduce splitting during and after driving"(Stern,1986). For block pallets larger nails and softer woods with less splitting tendency are frequently used. The wider blocks also contribute to reduced splitting. For any pallet performance can be greatly increased with more effective fastening of the pallet members.

There are other advantages and disadvantages to the block pallet design over the stringer-type pallets :

Advantages :

(1) Block pallets are true 4-way entry platforms (i.e they are fully accessible on all four sides).

(2) They do not require strength reducing notches for four-way entry.

(3) Slightly less lumber is used in block pallets compared to stringer-type pallets for same load capacity.

(4) Higher strength values for same pallet dimensions.

Disadvantages :

(1) More cutting, handling and fasteners are needed in block pallets. All contribute to a higher labor cost.

(2) Block pallets are currently fabricated in a two-step operation rather than the one step used for stringer-type assembly.

(3) They are marginally more expensive than the stringer-type pallets.

These arguments for block pallets together with the need to remain competitive in the pallet market, justify the need to develop a standardized design procedure to include block-type pallets.

# 2.0 BLOCK PALLET DESIGN

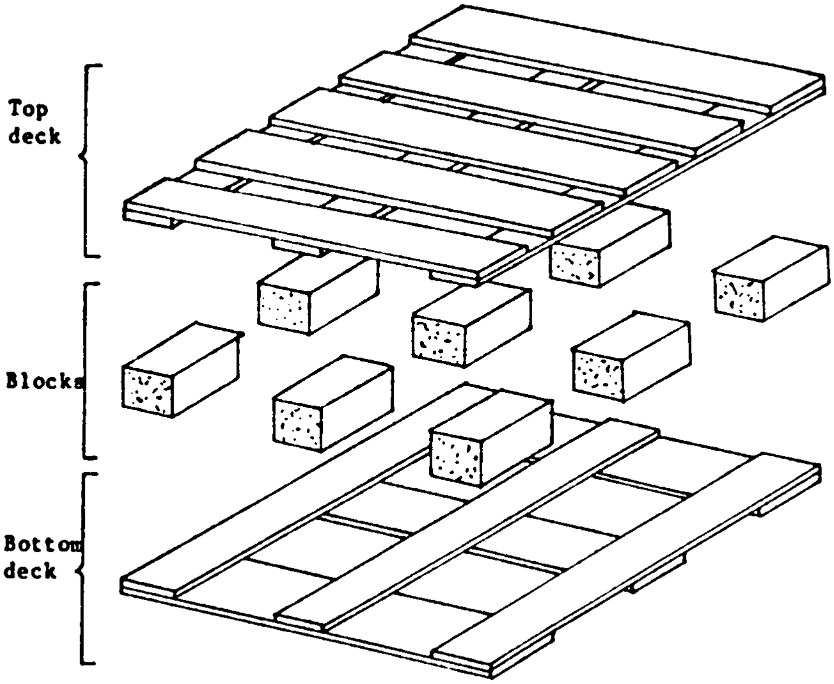
The block pallet may be viewed as a semi - rigidly connected framework of components consisting of stringerboards, deckboards and blocks. These components may be assembled in any number of ways to produce pallets of varying geometries. A block pallet may be thought of as three seperate elements, a top deck, a bottom deck, and connecting blocks. The decks may include stringerboards and/or deckboards. Some designs may only include a top deck.

Given a desired pallet geometry the design process must recognize the various sources of variability in a) the resistance of the pallet to the effects of applied load and b) variation in applied load. The ultimate objective of design is to rationally account for these variations in such a manner as to result in a safe, economical product.

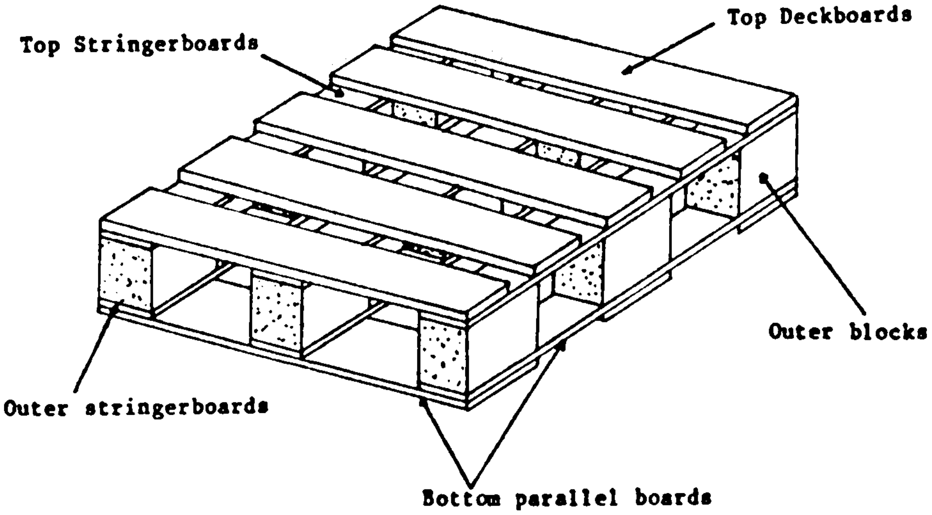
## 2.1 Pallet geometry

### 2.1.1 Definitions

The three sections which make up a block pallet are shown in Figure 2.1. The top deck contains both stringerboards and deckboards of any cross - sectional area. The length of these members is taken as their end to end distance. Pallet length is defined as the length of the top stringerboards. Conversely, the pallet width is defined as the length of the top deckboards.



a) 3 sections in a typical block pallet design



b) Assembled block pallet design

Figure 2.1. Block pallet components

The blocks which make up the center section of the pallet may be rectangular, square or cylindrical in shape. The length of the block is defined as the larger dimension which may be oriented with either the pallet length or width axes. Block pallets typically contain between 4 and 9 blocks which are usually located at the intersections of stringerboards and deckboards.

The bottom deck, if present, may be made up of stringerboards and/or deckboards of varying cross sectional area. The member lengths are defined as for the top deck. Definitions of some bottom deck designs are outlined below:-

a) Unidirectional-type deck: This case consists of all bottom boards aligned in only one direction, either parallel to top stringerboards or parallel to the top deckboards, are present. These boards are defined as bottom deckboards.

b) Perimeter-type deck: Bottom boards parallel to the top stringerboards as well as bottom boards parallel to top deckboards, all in the same plane. The two boards which run the full length or width of the pallet are defined as outer boards. The two or three boards which are perpendicular to and butt into the outer boards are defined as butted boards.

c) Overlap-type deck: Bottomboards parallel to top stringerboards as well as bottom boards parallel to top deckboards, not all in the same plane. The layer of bottom boards in direct contact with blocks are defined as bottom stringerboards and may be either parallel or perpendicular to the top stringerboards. The lower most layer of bottom boards are defined as bottom deckboards and may be either parallel or perpendicular to the top deckboards.

### 2.1.2 Geometry variation

Possible geometry alternatives with stringer-type pallets involve number, size and placement of stringers and deckboards. However, with block pallets there are more possible options. The number of blocks per stringerboard may vary within a pallet. The number of deckboards in a deck (top or bottom) may be greater than the number of stringerboards or vice versa. The bottom deck may or may not include both stringerboards and deckboards. Fastenings, nail types and nail patterns may also vary within a pallet.

### 2.1.3 Study limitations - geometry.

Because of the high number of geometric options some limitations were imposed for this study. These were chosen to maintain user flexibility but bypass potentially difficult analysis schemes. The limitations listed below were selected through discussion with industry advisors. The scope of this research was limited to :-

a. Common block pallet geometries. Pallets with 2 or 3 stringerboards of any rectangular geometry on top and/or bottom decks, and containing 4,6,8 or 9 blocks of any rectangular geometry. A maximum of 3 blocks per stringerboard were considered.

b. Deckboards of any rectangular geometry with a maximum of 13 boards per deck. In the case of cantilevered top stringerboards the maximum for the top deck is 15 deckboards.

c. Deckboard ends may be flush with the edge of the block, single winged or double winged.

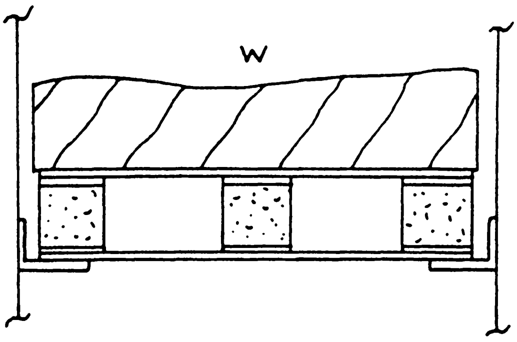
d. Stringerboard ends may be flush with the edge of the block, single cantilevered or double cantilevered (see Figure 7.1 Design 5).

e. Most commonly used fasteners and joint types resulting from typical assembly practices.

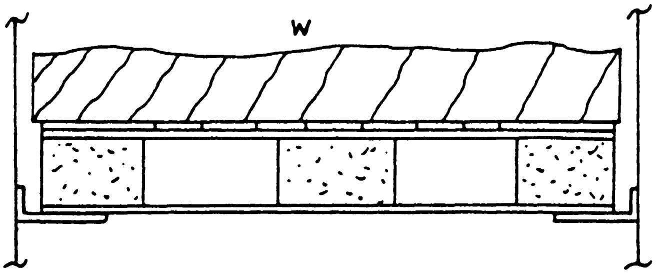
## 2.2 Pallet use conditions

### 2.2.1 Support modes.

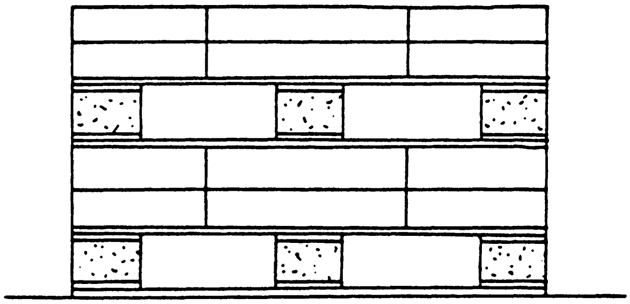
The types of support models considered within the scope of this study were similar to those used for stringer-type pallets (Loferski,1985). These are racked across stringers (RAS), racked across deckboards (RAD), sling supported and stack (see Figure 2.2). The RAD mode causes the top deckboards and bottom parallel boards to be stressed as a composite beam. Similarly, the RAS mode causes the top stringerboards and bottom parallel boards to be stressed as a composite beam. The stack mode causes the top deck of the bottom pallet and the bottom deck of the second pallet in a stack to function independently as continuous beams. The sling support mode causes actions in the top and



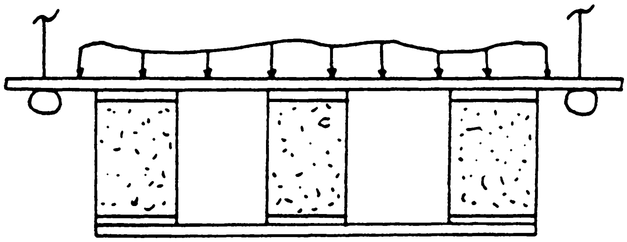
a) Racked across deckboards (RAD) - (end view)



b) Racked across stringerboards (RAS) - (side view)



c) Stack support



d) Sling support - (end view)

Figure 2.2. Block pallet support modes.

bottom decks similar to the RAD mode except that load is transferred through the top deck wing to the support.

### 2.2.2 Load models.

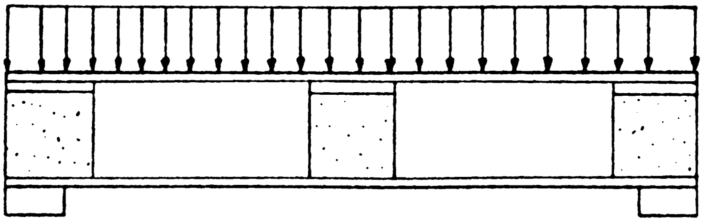
There are five different load types that were considered for use with the analysis procedures: a uniformly distributed load covering the entire top deck, a partial uniform load, and 1, 2, or 3 rigid line loads. Line loads may act parallel to either the top stringerboards or deckboards. These are shown in Figure 2.3. The uniformly distributed load is applied directly to the deckboards and is represented as point loads at the deckboard-stringerboard nodes as shown in Figure 2.4. The line loads are represented as point loads acting on the stringerboards as shown in Figure 2.5. For load cases involving 1 or 3 line loads the applied load at the centerline is modeled as half the applied value because of symmetry.

## 2.3 Design methodology

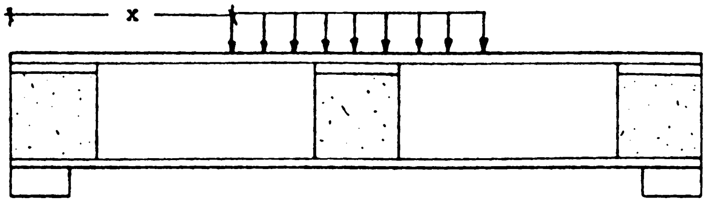
### 2.3.1 Design steps.

This section outlines, in a general sense, the steps that a pallet designer would use to devise a pallet which properly balances functionality and safety. These steps fall into three categories: structure definition, analysis of load effects, and safety/serviceability criteria.

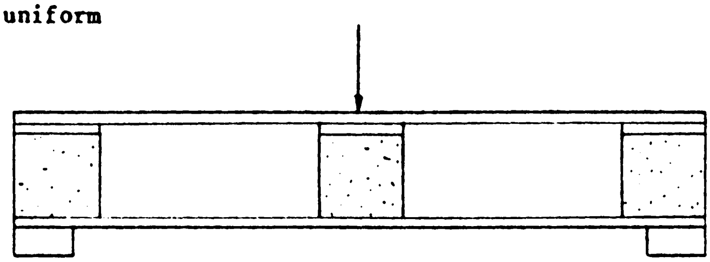
a. Geometry: The global geometry of the pallet is defined in terms of the number, location and size of each element. Typically, a



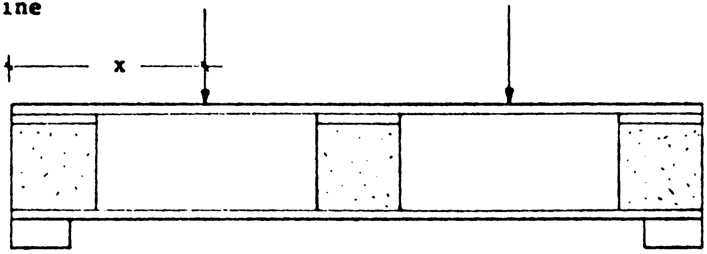
a) Full uniform load



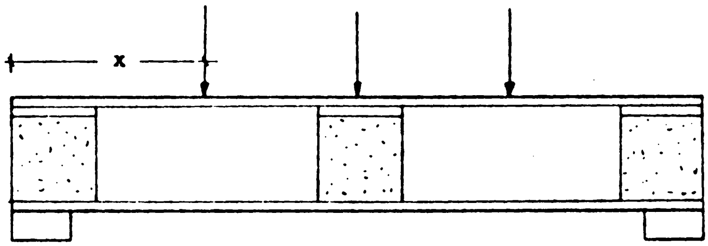
b) Partial uniform Load



c) Single line Load



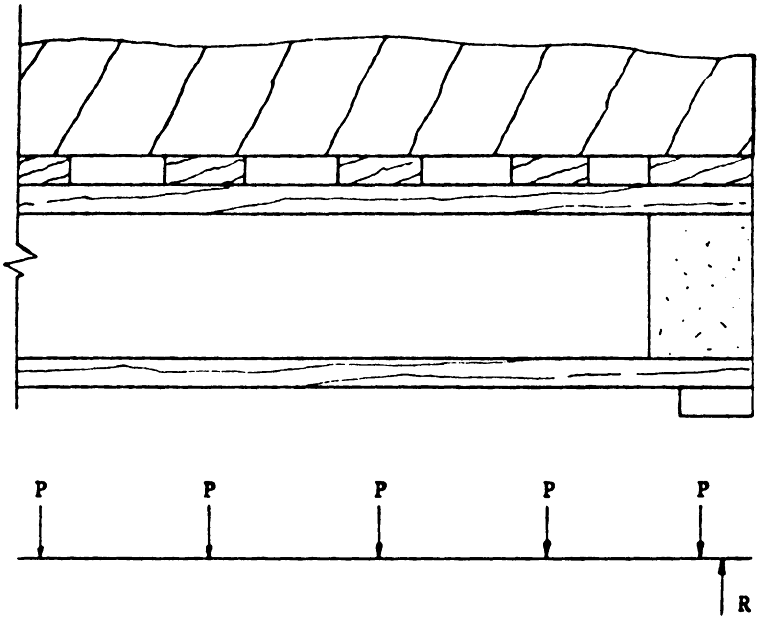
d) Two line loads



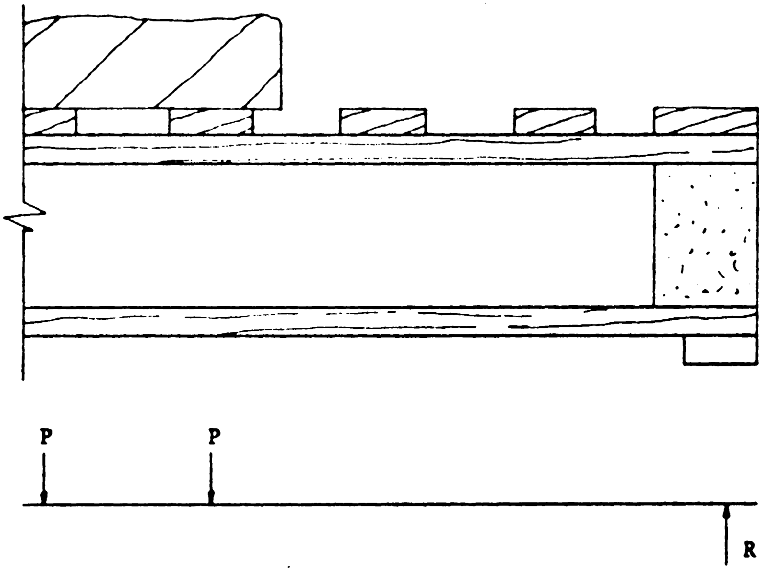
e) Three line loads

where : x = input distance

Figure 2.3. Load types analyzed in block pallet design

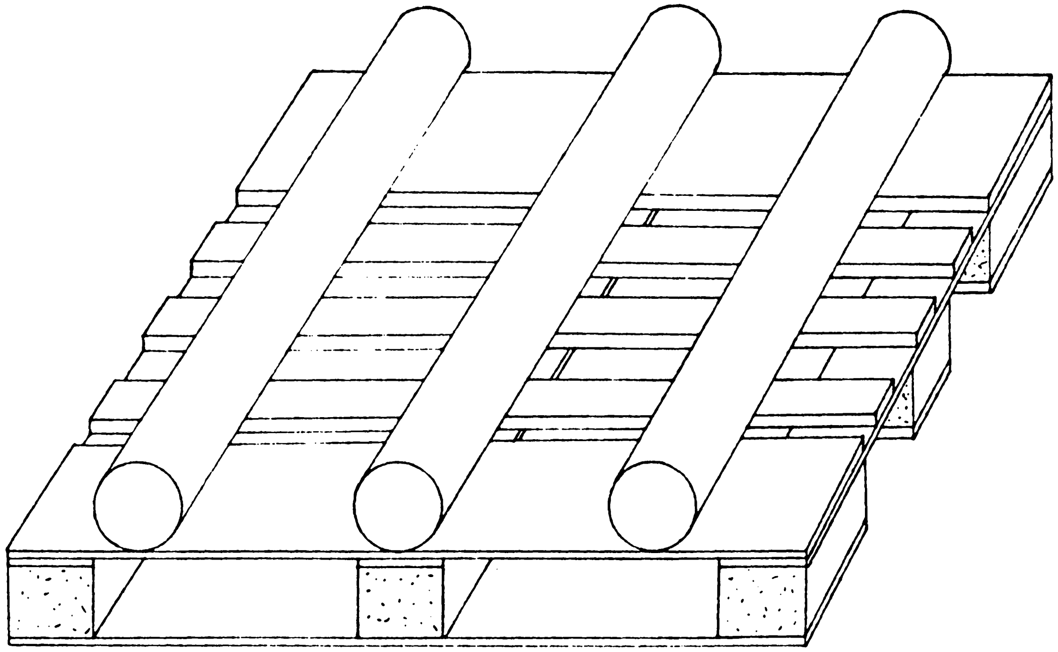


a) Full uniform load

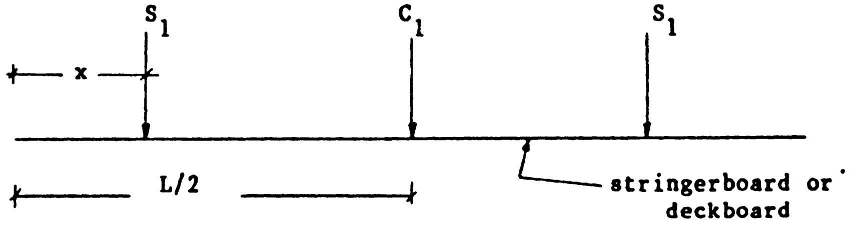


b) Partial uniform load

Figure 2.4. Load analog models for full and partial uniform loads



a) Real pallet



b) 2 - D load analog

where : x = input distance

Figure 2.5. Load analog models for line loads

trial geometry is first chosen and then refined by successive iteration of the design process.

b. Material properties: These are defined for each member and represent the expected properties of the components in the pallet.

c. Support condition: For each support condition the location of the supports and, hence, the spans are established.

d. Loading conditions: For each specified load type the applied load is apportioned to the individual elements to simulate the actions of the actual loading condition.

e. Element stresses: The load effects are calculated using principles of engineering science and structural analysis techniques. The location and magnitude of the most critical load effect is reported back to the user.

f. Compare with material resistance: The maximum element stress is compared to the estimated material resistance with consideration for safety using the reliability-based methodology (Loferski, 1985). Additionally, serviceability in the form of deformation requirements are considered using material stiffness and a user defined serviceability limit.

### 2.3.2 Reliability-based safety format (RBD)

In all cases of design the component resistance must be greater than the load effects by an amount sufficient to maintain an acceptable probability of failure. When comparing the resistance with the load effects using the techniques employed by PDS, an acceptable balance of safety and economy must result. For block-type pallets a mean-value reliability procedure (based on exact formulation for comparing log-normally distributed variables) was used to maintain this balance similar to that employed by Loferski (1987), Loferski & McLain (1987).

The RBD procedure can be used for either a "DESIGN" or "ANALYSIS" option. The DESIGN option allows the user to optimize the structure in terms of minimum member dimensions. Conversely, the ANALYSIS option is used to compare pallet designs on a load capacity basis. Equation 2.3 formally describes this linkage between load effects and resistance. It may be rearranged to solve for either the mean resistance or mean load effects:

$$\beta = \frac{\ln(R/S)\sqrt{(1+V_s^2)\ /\ (1+V_r^2)}}{\sqrt{\ln[(1+V_s^2)(1+V_r^2)]}} \qquad [2.3]$$

where: $\beta$ = safety index

$R$ = mean resistance

$S$ = mean load effects

$V_s$ = coefficient of variance for S

$V_r$ = coefficient of variance for R

A safety index, $\beta$, is used to explicitly describe the level of inherent safety to be built into the structure. This index is set based on "calibration" with current practice. The interested reader is referred to Loferski and McLain (1987) or Loferski (1985) for details. In block-type pallets only an ANALYSIS procedure is provided at this time; that is, only load capacity is output.

## 3.0 BLOCK PALLET ANALYSIS

### 3.1 Matrix structural analysis

To analyze the actions of a block pallet, the stiffness or displacement method of Matrix Structural Analysis was used. The actual structure is represented as an assembly of discrete elements interconnected at joints (nodes). A discrete model is formulated to have a finite number of degrees of freedom.

In the development of simplified block pallet design procedures, two analysis models were used. The first, a space frame, or three-dimensional model, was used to emulate the pallet under applied load in the RAS support mode. The second model used a plane frame, or two-dimensional model to estimate the pallet actions in the RAD mode.

A space frame model is an assembly of interconnected elements in a three dimensional framework. Each joint is initially considered to have six degrees of freedom with three translations in the global 1, 2, and 3 directions and three rotations about the three global axes. For typical loads the actions of each frame element is characterized by axial deformation, flexural deformations about the two principal axes (i.e global 2 and 3 axes), and torsional deformation.

A plane frame model is an assembly of rigidly interconnected elements which lie only in the global 1-2 coordinate plane. Each joint has three possible actions, two translations in the global 1 and 2 directions, and a rotation about the global 3 axis. The axial and flexural deformation element models are sufficient to model plane frame structures.

In the matrix displacement method the joint displacements are selected as the unknowns. Each element is represented by its individual stiffness matrix which relates the element - end displacements to the element - end forces for that element.

$$\{f\} = [k]\ \{d\} \qquad [3.1]$$

where: $\{f\}$= local element forces

$[k]$= local element stiffness matrix

$\{d\}$= local displacements of element.

The discrete model uses conditions of compatibility, equilibrium, and constitutive laws. Constitutive laws, representing material behaviour are included in the element models. The compatibility conditions consist of continuity relationships between element and joint displacements, including joint constraints. By balancing the applied joint forces with element - end forces, equilibrium is satisfied. The system model relates the joint displacements to the joint forces through the system stiffness matrix which includes the element stiffness relations assembled in a rational order.

$$\{Q\} = [K]\ \{q\} \qquad [3.2]$$

where; $\{Q\}$= equivalent joint force vector

$[K]$= system stiffness matrix

$\{q\}$= local displacement vector.

"Once the system model is solved for the joint displacements,any measure of response can be determined" (Holzer,1982).

### 3.2 Joint behavior

Joint behavior is a very important variable in modeling block pallets. Loferski(1985) states that "joints should be flexible to allow stressing without failure but stiff enough to resist bending stresses up to the crushing strength of the wood." Previous research on nailed wood joints has been directed towards two areas, load-slip behavior (shear resistance to lateral load) and load-separation behavior (withdrawal resistance to axial load and rotation). Most studies have evaluated a load-slip curve. However, Mack (1975) found that the slip at the deckboard-stringer joint was a key variable in stringer pallet design. Because of the obvious difference in physical geometry between blocks and stringers in pallet construction, block pallet joints were modeled differently than stringer pallet joints. The main difference lies in the finite width of the block at the joint. The deckboard-block joint models utilize a separation modulus, a rotation modulus and the lateral stiffness in the global 1 direction.

The rotation modulus and the separation modulus (withdrawal stiffness) are "constants describing the degree of fixity of a nailed joint under moment and axial force respectively. The separation modulus is defined as the ratio of the applied withdrawal force to the corresponding separation; the rotation modulus is the ratio of the applied moment to the angular rotation" (Kyokong,1979). Samarasinghe (1987) developed a model to compute rotation modulus based on head pull through, withdrawal stiffness and edge crushing of the block using matrix structural analysis. See Figure 3.1.

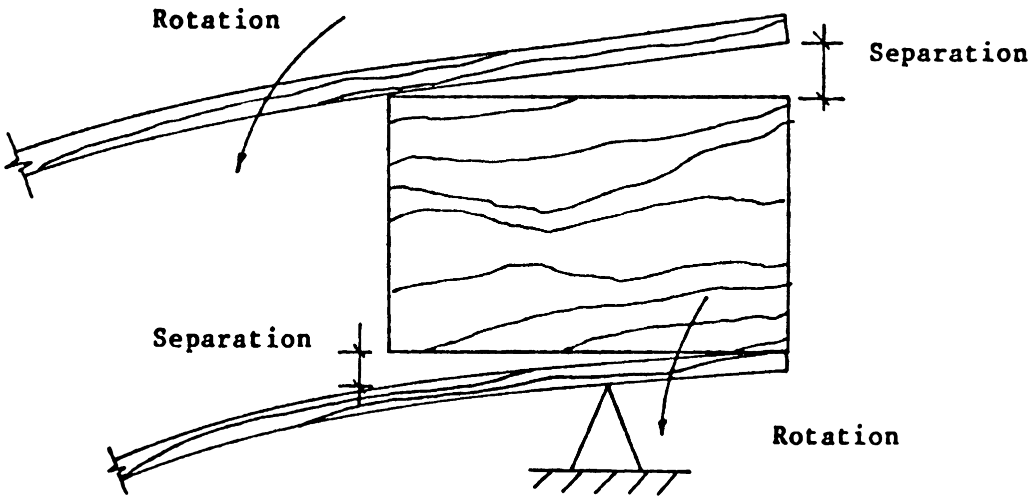


Figure 3.1 Rotation modulus and Separation modulus

The lateral stiffness (slip modulus) is a measure of joint stiffness in lateral loading. It is a ratio of the applied lateral load to the translation in the direction of the load.

### 3.3 Definitions

The symbols, conventions and techniques used in this paper to represent both plane and space frame structures in matrix form are summarized below:

(1) The structures are assumed to lie in the global 1-2 and global 1-2-3 coordinate planes respectively.

(2) The joints are numbered in sequence from 1 to NJ, where NJ is the number of joints.

(3) The joint displacements and the joint loads, denoted $q_k$ and $Q_k$, respectively, are numbered in the sequence of the global coordinate axes from the lowest to the highest numbered joint, $k = 1,2,3....n$, where n is the number of joint displacements.

(4) The elements are numbered in sequence from 1 to NE, where NE is the number of elements.

(5) The local reference frame for each element is defined as follows : The local 1-axis is directed from the lower (a-end)

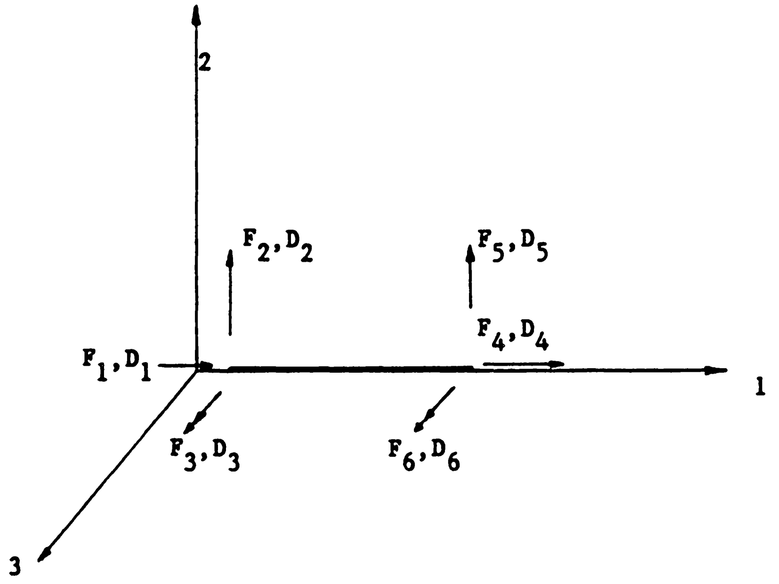
to the higher (b-end) numbered joint to which the element is incident; the local 3-axis is oriented normal to the local 1-2 frame; and the local 2-axis is chosen to make the local frame of reference follow the right hand convention.

(6) The local element displacements and forces, denoted by lowercase letters $d_j$ and $f_j$, are numbered in the sequence of the local coordinate axes from the a-end to the b-end of the element. The subscript j refers to the element number. The global element displacements and forces are denoted by the uppercase letters $D_j$ and $F_j$, and are numbered in the sequence of the global coordinate axes from the a-end to the b-end of the element.

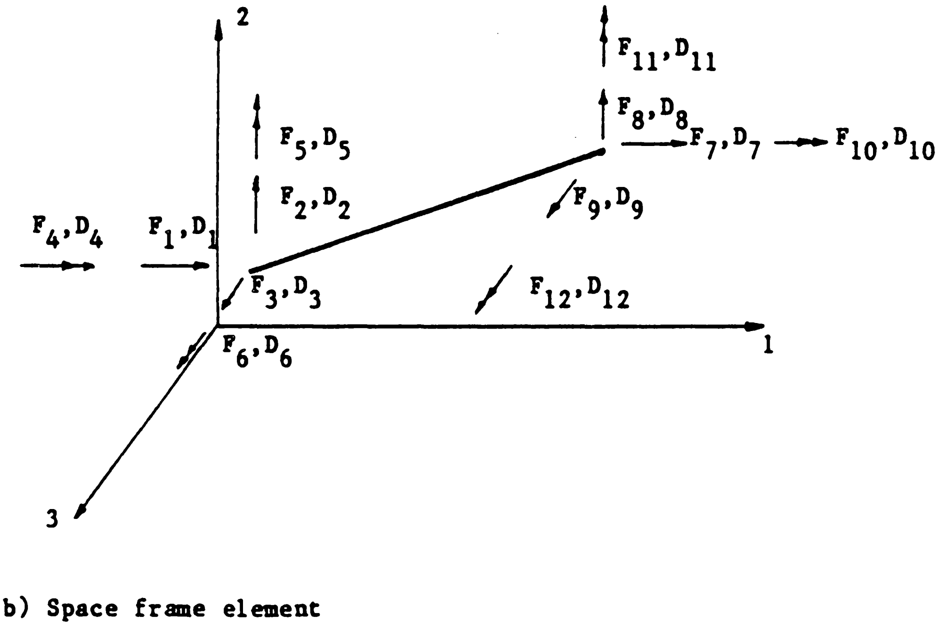
(7) The sign of a displacement or force at a point is determined by the direction of the corresponding coordinate axes. The right-hand rule determines the positive sense of the rotation or moment about the 1, 2, or 3 axes. See Figure 3.2.

(8) The conditions of compatibility and equilibrium are expressed in compact form by the member code and joint code matrices. The member code matrix is defined as :-

$$\text{MCODE}(L,I) = \begin{array}{l} K \text{ if } D = q_k \\ 0 \text{ otherwise} \end{array}$$



a) Plane frame element



b) Space frame element

Figure 3.2. Plane and space frame elements showing possible global displacements and forces

where K is the degree of freedom number associated with element I and D is the Lth global displacement of element I. k has the same value as K. The values of L for space frame elements range from 1 to 12 and from 1 to 6 for plane frame elements.

The JCODE matrix is defined as follows :-

$$\mathrm{JCODE}(L,J) = \begin{cases} K & \text{if } U_{lj} = q_k \\ 0 & \text{otherwise} \end{cases}$$

where $U_{lj}$ is the displacement in the Lth direction of joint j and $q_k$ is the Kth joint displacement. The values of L for space frame elements range from 1 to 6 and from 1 to 3 for plane frame elements.

## 3.4 Solution

Below is a summary of the steps used to obtain stresses from the applied loads using the matrix approach. Specific details for each support condition are found in chapters 4-6.

1. First, a model is constructed to simulate the actions of a pallet under a unit load. All members in the model are given material properties and geometries to simulate those of the elements in the real pallet.

2. All possible joint displacements are then identified in the model

and assembled into the displacement vector, $\{D\}$. When all constraints and supports are applied to the model the sum of the degrees of freedom is the total number of degrees of freedom for the structure.

3. A system stiffness matrix, [K] is computed. This matrix defines the relationship between the applied joint load vector $\{F\}$ and the displacement vector $\{D\}$. It is assembled by adding together all appropriate element stiffness matrices and transferring them to the relevent cells of the [K] matrix.

4. The applied joint force vector, $\{Q\}$ is defined. The assembly process is similar to that for the displacement vector and contains all applied joint forces in a numbered sequence.

5. At this point the matrix components are assembled into the governing relationship :

$$\{Q\} = [K]\,\{D\} \qquad [3.4]$$

where; $\{Q\}$= equivalent joint load vector

$[K]$= system stiffness matrix

$\{D\}$= global displacement vector.

The Cholesky method using computer techniques is used to solve for the joint displacements which define the displaced configuration of the structure.

6. From the displacements obtained in the solution routine, the

element end forces and stresses may be calculated.

## 3.5 Sensitivity analysis and Model selection

The main tool used to develop simplified analysis models was "SPACEPAL". This general purpose structural analysis program is written in Fortran IV and was originally developed for stringer-type pallets (Mulheren,1984). It is based on a three dimensional matrix displacement (stiffness) method and has non linear joint analysis capability. SPACEPAL, as a general purpose program is too cumbersome for direct application to pallet design. However, it has been exhaustively tested and could be used with confidence to develop simplified, efficient models for block pallet design.

To identify simple models which can best simulate the actions of block-type pallets under service loads, many different alternative models were input to SPACEPAL and iteratively modified. The model predictions were compared to experimental results (described in chapter 7). A model was accepted when predicted and experimental results consistently agreed. If possible, symmetry was used to reduce problem size and shear releases were positioned along the lines of symmetry.

In the RAS support mode the full model was reduced to a quarter symmetric 3-dimensional model. This model allows full consideration of load sharing between stringerboards and the influence of deckboard location on pallet strength and stiffness. The RAD model is a half symmetric 2-dimensional model chosen to take advantage of the reduced complexity of member interaction in this mode. Additionally, experimental results of RAD pallets showed that bending about only one axis dominated pallet action. For the stack condition both models are used to calculate the load effects in the critical top and bottom deck members. Details of these models are found in the next three chapters.

Sensitivity studies were carried out on SPACEPAL models of RAS and RAD pallets to determine the important joint parameters and the level of required input accuracy. These were assessed successively by modifying the stiffness in the six local directions for the spring elements employed in the models. In the RAS model springs were used to model both deck-block joints and stringerboard/deckboard joints. For the RAD model only deck-block joints are modeled.

An iterative process was used to examine the sensitivity of pallet actions to spring element stiffness in each of the six local directions. Varying degrees of stiffness were assigned to each direction for all spring elements, one by one, with any corresponding change in center point deflection noted. Based on this sensitivity analysis the following conclusions can be drawn;

<u>RAS model:</u>

- With stringerboard/deckboard joints large changes in the translational stiffness in the three global directions and in rotation about the global 2-axis did not appreciably affect pallet response. For a 100% change in translational or rotational joint stiffness there was a 0 - 2% change in center point deflection.

These stiffnesses were fixed and given high values in the working model.

- Rotational stiffness about the global 1-axis for the modeled stringerboard/deckboard joint showed similar insensitivity and was given a fixed finite value. Rotational stiffness about the global 3-axis was assigned zero stiffness to preclude geometric anomalies.
- The observed experimental joint behavior was best modeled when the axial springs connecting deckboards to blocks were given zero stiffness in all directions except the global 2-direction (withdrawal). The center spring on the top deck of the deck-block joint was given a finite stiffness (separation modulus) based on results of Samarasinghe, 1987. The two other axial springs are initially given zero stiffness in the global 2-direction and then are made very stiff if required by conditions of physical compatibility.
- The rotational stiffness about the global 3-axis of the deck-block joint was found to be a sensitive input parameter (i.e a 100% change in stiffness produced a 10% change in center point deflection). This stiffness (rotation modulus) is assigned a realistic finite value based on the results of Samarasinghe, 1987. Stiffness in all other directions are given high values.

<u>RAD model:</u>

- Similar sensitivity results were found for the RAD model as for the RAS model. As a result, the axial springs in the deck-block joint are given the same stiffness values as in the RAS model. To simulate the experimentally observed joint behavior the stiffness of the top deck rotational spring in the deck-block joint was given zero stiffness in the global 1 and 2 directions. The corresponding bottom deck rotational spring was given a high stiffness in both the global 1 and 2 directions.

Values for rotation modulus and separation modulus are based on results from Samarasinghe, 1987. Stiffness values assigned to spring elements in both models are presented in Figures 4.4 and 5.11.

# 4.0 ANALYSIS OF RACKED ACROSS STRINGERBOARDS SUPPORT CONDITION (RAS)

The objective of this chapter is to outline in detail the methodology used to analyze block pallets for the racked across stringerboards support condition (RAS). The key to the design process is the use of a matrix structural analysis model to identify the most critical member in the structure and greatest deformation and to report this back to the user for subsequent comparison with the component resistance or serviceability criterion under the particular loading condition.

## 4.1 Assumptions and limitations

Because of the wide variety of geometries and designs that accompany block pallet analysis, the scope of the pallet design system must be limited to the most practical situations. Assumptions are made for different load and support conditions to best model the real pallet. Limitations are also placed on load and support conditions, and block pallet geometries to make the problem manageable. These assumptions and limitations for the RAS mode are detailed below:

a. Five loading conditions may be analyzed : uniformly distributed load, partial uniform load, and 1, 2, or 3 symmetrically placed line loads.

b. The line loads are considered to act directly on the stringerboard. Line loads can not be applied to cantilevered top stringerboard members.

c. Partial uniform loads may not extend over the cantilevered top stringerboard members.

d. The supports may be placed only under the bottom deck. The support joint inside the outer blocks must lie between the inside edge of the outer block and the center of the nearest bottom deckboard to the left.

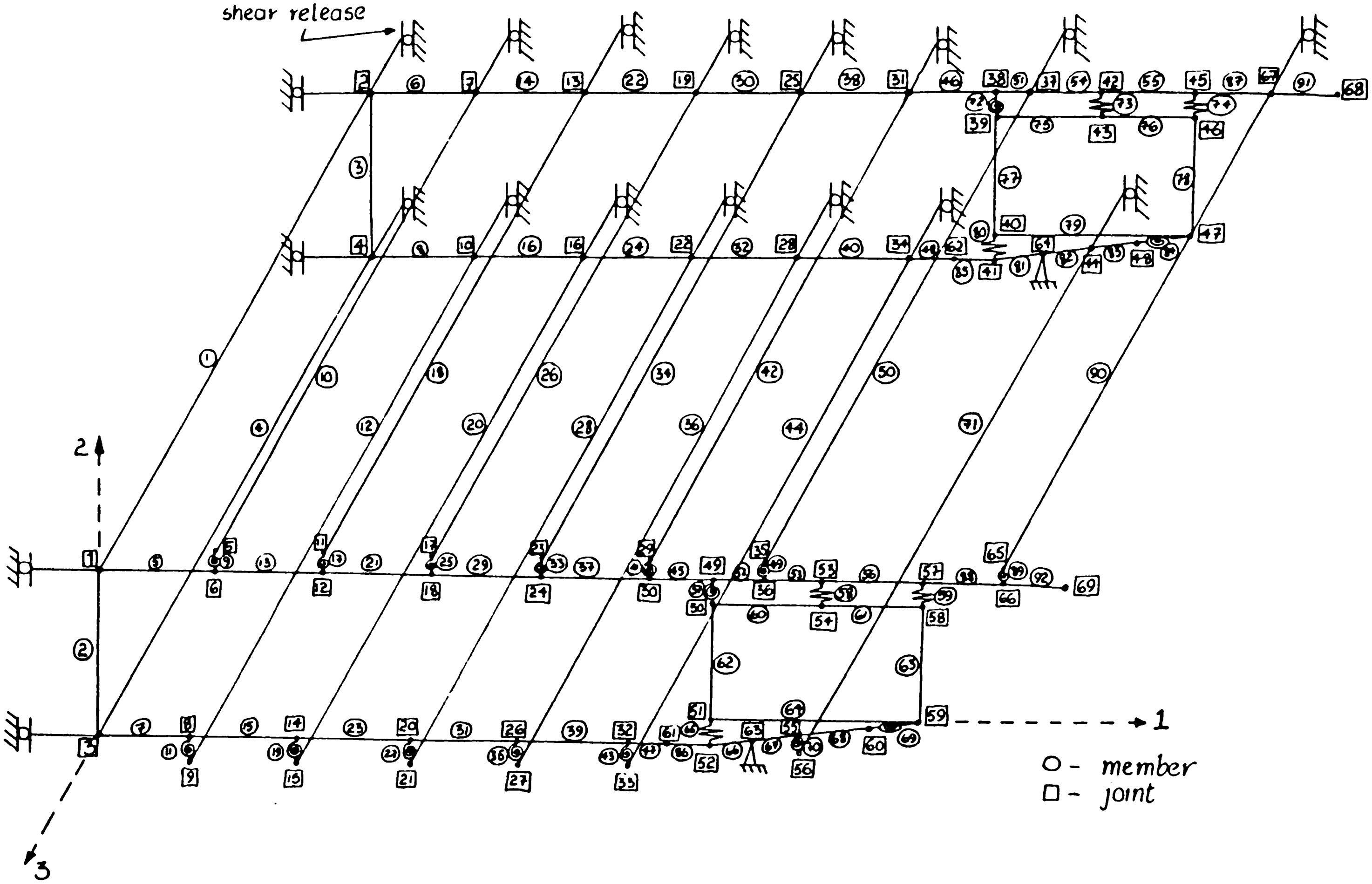
e. With cantilevered bottom stringerboards, the support may not be placed under the cantilever members.

f. Only one deckboard is allowed to be placed over the outer blocks, and in the event of cantilevered top stringerboards, only one deckboard is permitted over this portion.

### 4.2 RAS analog model

The matrix structural analysis analog model used for the RAS support condition is a quarter-symmetric space frame model shown in Figure 4.1. The Fortran code used with this analysis is shown in Appendix A. Each joint has six degrees of freedom - three translations along each of the three global axes and three rotations about the axes. To ensure symmetry, shear releases are placed at member ends along the 1 and 3 centerline axes. These shear releases allow vertical deflection (i.e in the 2 - direction) but are restrained in other directions as follows:-

Figure 4.1. RAS analog model



a. Shear releases along the 3 - axis (i.e joints 1, and 3): These are restrained from translation along the 1 - axis, rotation about the 2 - axis, and rotation about the 3 - axis.

b. Shear releases along the 1 - axis: These prohibit translation along the 3 - axis, rotation about the 2 - axis, and rotation about the 3 - axis.

c. Shear releases at the center point of the pallet are restrained in all directions except translation along the 2 and 3 - axes.

The model can analyze pallets with either 4, 6, 8, or 9 blocks and up to 13 deckboards. However, if the top stringerboards are cantilevered it will actually model up to 15 top deckboards. The program will automatically add two deckboards if the top stringerboards are cantilevered. Therefore the number of top and bottom deckboards is always input as the number between the outer edge of the outside blocks in the RAS span.

The model is set up to have a maximum of 92 members with 291 degrees of freedom for the structure with 15 top deckboards. For geometries having fewer top deckboards, the model is dynamically restructured with the number of elements employed dependent on the number of top deckboards in the pallet. This considerably reduces the number of degrees of freedom for the smaller geometries thereby reducing computation time.

The rationale for the model of the deck-block joint is based on observations of physical tests and is discussed in chapter 5. Members representing stringerboards in the model are oriented with their length in the global 1 - direction. Deckboard members are oriented in the global 3 direction, and block members are in both the global 2 and global 1 directions. A maximum of three stringerboards per deck and a maximum of nine blocks can be accomodated.

## 4.3 Analysis methodology

This section outlines, step by step, the procedure followed to analyze block pallets in the RAS support mode.

### 4.3.1 Structure definition:

From the user input a model is assembled, representative of the real pallet, based on Figure 4.1. All material properties and geometries for each member must be defined in addition to the locations of all constraints and support joints. Joints are numbered as shown in Figure 4.1. Members representing the outer blocks and supports are mobile and can be located according to the pallet geometry. Those elements to the right of the outer block elements are unnecessary and are removed from the model. This minimizes the half band width thereby reducing the size of the system stiffness matrix. The computation time required to solve equation [3.2] depends on the number of top deckboards in the pallet. The following steps required to define the structure are executed by the program:

a. From the input number of top deckboards in the pallet, the number of elements needed in the model can be determined. A dynamically dimensioned array, MN (maximum dimension 13) corresponds to the number of elements up to and including the last top deckboard. This number is then added to 36 (the number of elements in two blocks). A "flag" variable is introduced into the end of the equation to add six more elements if the top stringerboards are cantilevered. A similar procedure with array JN is used to compute the number of joints in the model.

$$NE = MN(NTD) + 36 + (6 * ICANT) \qquad [4.3]$$

$$NJ = JN(NTD) + I + (5 * ICANT)$$

where:

NE = number of elements in the model,

NJ = number of joints in the model,

NTD = number of top deckboards,

ICANT = cantilever flag: 0 = no, 1 = yes,

I = number dependent on the number of top deckboards,

$$I = 29, \text{ if } NTD = 1$$

$$\text{or } I = 27, \text{ if } NTD > 1$$

b. The joint coordinates are established in subroutine 'COORDS'. The models' origin is located at the center of the bottom outer stringerboard. The location of the support joint is half the span and is stored for later use in determining the location of joint constraints. All 2 - coordinates are one of two values, zero or block height. The 3 - coordinates also are either zero or half the width of the pallet. The 1 - coordinates are based on centerline locations of deckboards and leading edge locations of block elements.

c. Member incidences (i.e the joint numbers at the a and b-ends of the member) up to and including the last top deckboard are initially defined in the main program. When the geometry and size of the pallet are established, modifications to the member incidences are made to account for the location of the outside block elements and cantilever stringerboard elements. Further modifications are made in subroutine "MINCS" to account for the position of the support joint and the top deckboard directly over the outer blocks.

d. The next step is to define joint constraints and to determine the number of degrees of freedom for the structure. Each joint in the model initially has six degrees of freedom. Constraints are added to some joints along the centerline axes to account for shear releases. The degrees of freedom for support joints are modified to include constraints in the proper directions. An array, $\{JCODE\}$ is set up to identify all constraints. The $\{JCODE\}$ array has six rows which correspond to the degrees of freedom in the global 1, 2, and 3 directions. The number of columns in $\{JCODE\}$ is equal to the number of joints in the model. Initially all cells in the $\{JCODE\}$ array are set equal to "1" to represent freedom in all directions. The array is then modified to account for the constraints and support joints by entering "0" in any cell corresponding to a constraint.

All joints along the axes of symmetry are structured to act as shear releases except for the block member joints on the global 1 - axis. The shear releases along the 1 - axis permit translation in the global 1 and 2 directions and rotation about the global 3 axis. Those located on the global 3 axis allow translation in the global 2 and 3 directions and rotation about the global 1 - axis. The shear releases at the center point of the pallet are free to translate vertically and along the 3 - axis. All support joints are free only to rotate about the 3 - axis.

The $\{$JCODE$\}$ is further modified by assigning numbers in sequence to each non zero element in the array. The procedure followed goes from the first column to the last column for each element, ending with the degrees of freedom for the structure.

A member code array, $\{$MCODE$\}$ is then set up with twelve rows and one column for each member in the model. From the member incidences the joints at the "a" and "b" ends of each member are identified. The column in $\{$JCODE$\}$ corresponding to the joint at the "a" end of the member is transferred into the first six cells of $\{$MCODE$\}$. Similarly, the second six cells are filled with the $\{$JCODE$\}$ for the joint at the "b" end of the member.

e. From the defined member incidences the lengths of all elements are computed in subroutine "LENGTS". All members are assigned properties and geometries in "PROPS". Subsequently, members that are inherent to the model but not present in the real pallet are assigned zero properties in "DUMMY" . For example, if the real pallet had no bottom deckboards then those elements in the model that normally represent bottom deckboards would be given zero properties. For all real elements in the model the following parameters are calculated or assigned:

. Modulus of elasticity, XMOE

. Modulus of rigidity, XMG = XMOE / 16

. Moments of inertia (EI) about the three principal axes,

$$EI(1,I) = W(I)T(I)^3[0.33-0.209(T(I)/W(I))TANH(1.5705(W(I)/T(I)))]$$

("SPACEPAL", 1984)

$$EI(2,I) = T(I)\ W(I)^3/12$$

$$EI(3,I) = W(I)\ T(I)^3/12$$

. section moduli for the three principal axes,

$$S(1,I) = EI(1,I)\ 2.0/T(I)$$

$$S(2,I) = EI(2,I)\ 2.0/W(I)$$

$$S(3,I) = EI(3,I)\ 2.0/T(I)$$

. cross sectional area,

$$A(I) = W(I) * T(I)$$

where, I = element no.

W = width

T = thickness

These values are stored in arrays (I, W, and T) where the row number corresponds to the member number. See subroutine 'PROPS'.

### 4.3.2 Equivalent joint loads:

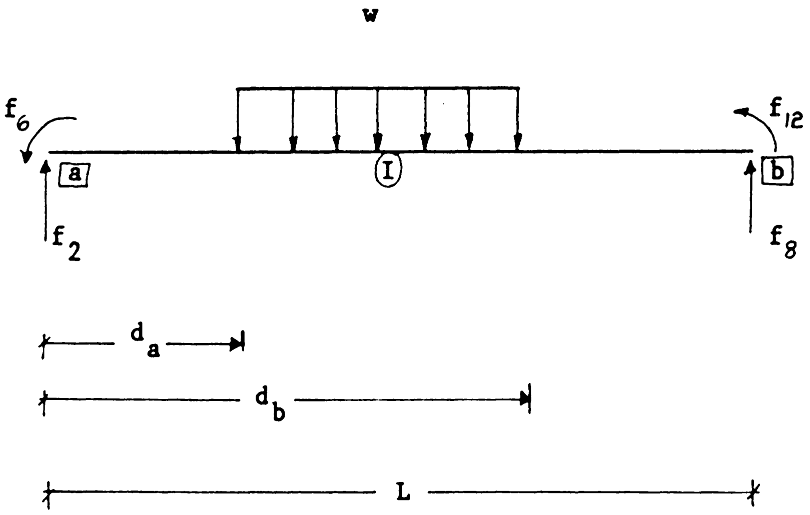
The equivalent joint loads are computed from the applied member loads in subroutines "RASUDL" and "RALINE". Uniform loads are applied directly to the deckboards. Each member load is transferred into equivalent joint forces (i.e Joint forces that cause the same member actions as the applied load). The equations in Figure 4.2 are used to calculate the fixed end forces for the loaded deckboard. The magnitude of the load on individual deckboard elements is determined by dividing the total load on the pallet by the total loaded area and then multiplying by the width of the deckboard.

Line loads are considered to act as point loads on the stringerboards. The equations to calculate the fixed end forces for members subject to line loads are shown in Figure 4.3.

With this information, an equivalent joint force vector, $\{q\}$, is assembled using $\{\text{MCODE}\}$ to identify the degree of freedom associated with each equivalent joint load.

### 4.3.3 Solution and stress computation:

The system stiffness matrix is next assembled in subroutine "RAASSM". The element stiffness matrix is first defined for each member



$$\hat{f}(2,I) = w/2L^2\ [2L^3(d_b-d_a) - 2L(d_b{}^3-d_a{}^3) + (d_b{}^4-d_a{}^4)]$$

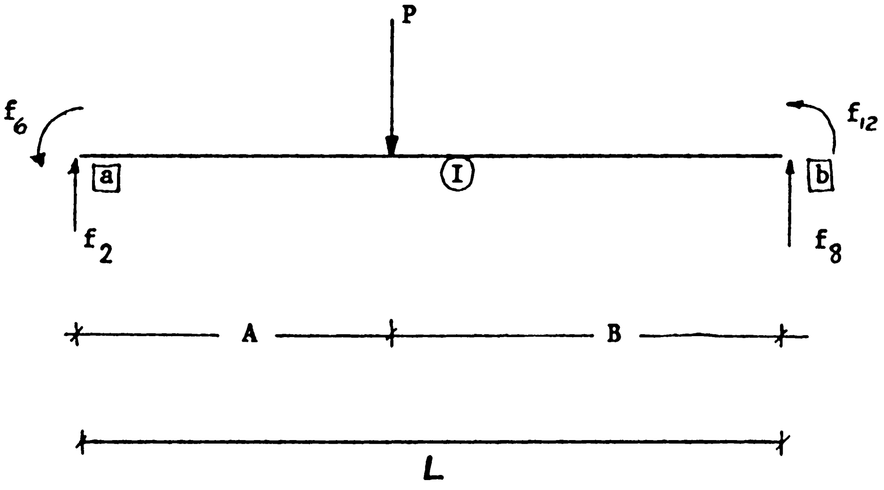
$$\hat{f}(6,I) = w/12L^2\ [6L^2(d_b{}^2-d_a{}^2)-8L(d_b{}^3-d_a{}^3)+3(d_b{}^4-d_a{}^4)]$$

$$\hat{f}(8,I) = w/2L^3\ [2L(d_b{}^3-d_a{}^3) - (d_b{}^4-d_a{}^4)]$$

$$\hat{f}(12,I) = w/12L^2\ [4L(d_b{}^3-d_a{}^3) - 3(d_b{}^4-d_a{}^4)]$$

where: $\hat{f}(N,I)$ = Nth fixed-end force for member I

Figure 4.2. Equations to compute the fixed end forces for deckboards subject to uniform loads.



$\hat{f}(2,I) = PB^2/L^3\ (3A + B)$

$\hat{f}(6,I) = PAB^2/L^2$

$\hat{f}(8,I) = PA^2/L^3\ (A + 3B)$

$\hat{f}(12,I) = PA^2B/L^2$

Figure 4.3. Equations to calculate the fixed end forces for stringerboards subject to line loads.

and then transferred into the proper locations of the system stiffness matrix. The array $\{INDEX\}$ is defined (Figure 4.4) and is used to "cross reference the position and number of each cell in the element stiffness matrix with the computed numerical value of the cell. The system stiffness matrix is assembled by referencing $\{MCODE\}$ to determine the cell coordinate in the system matrix, and the index array, to determine the value to be placed into previously identified cells" (Loferski 1985). The element stiffness matrices for all model elements are defined in Figure 4.5.

The Cholesky method (Holzer, 1983) is used to solve for the unknown displacements in the system equation (subroutine "SOLVE").

$$\{Q\} = [K] \{D\} \qquad [3.2]$$

Given the displacements of each member in the structure, the next step is to determine the corresponding stresses. This is done in subroutine "RASMST". In the RAS mode stresses are calculated for the stringerboard and deckboard elements. The stresses in the stringerboard elements are calculated using the equations in Figure 4.6. For line loads the load is considered to act directly on the stringerboard and the maximum stress may be located anywhere along the length of the member. The equations to compute stresses in stringerboard elements subject to line loads are shown in Figure 4.7. The equations to calculate the stresses in the deckboard elements are shown in Figure 4.8.

$$\text{INDEX} = \begin{bmatrix}
G_1 & G_2 & G_3 & G_{13} & G_{14} & G_{15} & -G_1 & -G_2 & -G_3 & G_{13} & G_{14} & G_{15} \\
 & G_4 & G_5 & G_{16} & G_{17} & G_{18} & -G_2 & -G_4 & -G_5 & G_{16} & G_{17} & G_{18} \\
 & & G_6 & G_{19} & G_{20} & G_{21} & -G_3 & -G_5 & -G_6 & G_{19} & G_{20} & G_{21} \\
 & & & G_7 & G_8 & G_9 & -G_{13} & -G_{16} & -G_{19} & G_{22} & G_{23} & G_{24} \\
 & & & & G_{10} & G_{11} & -G_{14} & -G_{17} & -G_{20} & G_{23} & G_{25} & G_{26} \\
 & & & & & G_{12} & -G_{15} & -G_{18} & -G_{21} & G_{24} & G_{26} & G_{27} \\
 & & & & & & G_1 & G_2 & G_3 & -G_{13} & -G_{14} & -G_{15} \\
 & & & & & & & G_4 & G_5 & -G_{16} & -G_{17} & -G_{18} \\
 & \text{sym} & & & & & & & G_6 & -G_{19} & -G_{20} & -G_{21} \\
 & & & & & & & & & G_7 & G_8 & G_9 \\
 & & & & & & & & & & G_{10} & G_{11} \\
 & & & & & & & & & & & G_{12}
\end{bmatrix}$$

where;

G = Global element stiffness coefficient

| Rotational springs | | | | Axial springs | | | | Stringerboard, block and deckboard elements |
|---|---|---|---|---|---|---|---|---|
| | deckboard | block | | | Block | | | |
| | | | | | Top | | Bottom | |
| | | Top | Bottom | | Middle | Right | | |
| $G_1$ = | $1\times10^7$ | $1\times10^7$ | $1\times10^7$ | $G_1$ = | 0 | 0 | 0 | See Figure 4.5 |
| $G_2$ = | $1\times10^7$ | $1\times10^7$ | $1\times10^7$ | $G_4$ = | Gam4 | Gam6 | Gam5 | |
| $G_6$ = | $1\times10^7$ | $1\times10^7$ | $1\times10^7$ | $G_6$ = | $1\times10^7$ | 0 | 0 | |
| $G_7$ = | $9\times10^4$ | 0 | $1\times10^7$ | $G_7$ = | 0 | 0 | 0 | |
| $G_{10}$ = | $1\times10^7$ | $1\times10^7$ | $1\times10^7$ | $G_{10}$ = | $1\times10^7$ | 0 | 0 | |
| $G_{12}$ = | 0 | Gam3 | Gam3 | $G_{12}$ = | 0 | 0 | 0 | |
| $G_{22}$ = | $-G_7$ | 0 | $-G_7$ | $G_{22}$ = | 0 | 0 | 0 | |
| $G_{25}$ = | $-G_{10}$ | $-G_{10}$ | $-G_{10}$ | $G_{25}$ = | $-G_{10}$ | 0 | 0 | |
| $G_{27}$ = | 0 | −Gam3 | −Gam3 | $G_{27}$ = | 0 | 0 | 0 | |
| All other G values are equal to zero | | | | | | | | |

Gam3 = Rotation modulus
Gam4 = Separation modulus
Gam5 and Gam6 are initially equal to zero

Figure 4.4. Element stiffness matrix INDEX coefficients for rotational and axial springs

$$g_1 = \gamma l_{11}^2 + 12\alpha l_{21}^2 + 12\beta l_{31}^2$$

$$g_2 = \gamma l_{11} l_{12} + 12\alpha l_{21} l_{22} + 12\beta l_{31} l_{32}$$

$$g_3 = \gamma l_{11} l_{13} + 12\alpha l_{21} l_{23} + 12\beta l_{31} l_{33}$$

$$g_4 = \gamma l_{12}^2 + 12\alpha l_{22}^2 + 12\beta l_{32}^2$$

$$g_5 = \gamma l_{12} l_{13} + 12\alpha l_{22} l_{23} + 12\beta l_{32} l_{33}$$

$$g_6 = \gamma l_{13}^2 + 12\alpha l_{23}^2 + 12\beta l_{33}^2$$

$$g_7 = \delta l_{11}^2 + 4L^2\alpha l_{31}^2 + 4L^2\beta l_{21}^2$$

$$g_8 = \delta l_{11} l_{12} + 4L^2\alpha l_{31} l_{32} + 4L^2\beta l_{21} l_{22}$$

$$g_9 = \delta l_{11} l_{13} + 4L^2\alpha l_{31} l_{33} + 4L^2\beta l_{21} l_{23}$$

$$g_{10} = \delta l_{12}^2 + 4L^2\alpha l_{32}^2 + 4L^2\beta l_{22}^2$$

$$g_{11} = \delta l_{12} l_{13} + 4L^2\alpha l_{32} l_{33} + 4L^2\beta l_{22} l_{23}$$

$$g_{12} = \delta l_{13}^2 + 4L^2\alpha l_{33}^2 + 4L^2\beta l_{23}^2$$

$$g_{13} = 6L\alpha l_{21} l_{31} - 6L\beta l_{31} l_{21}$$

$$g_{14} = 6L\alpha l_{21} l_{32} - 6L\beta l_{31} l_{22}$$

$$g_{15} = 6L\alpha l_{21} l_{33} - 6L\beta l_{31} l_{23}$$

$$g_{16} = 6L\alpha l_{22} l_{31} - 6L\beta l_{32} l_{21}$$

$$g_{17} = 6L\alpha l_{22} l_{32} - 6L\beta l_{32} l_{22}$$

$$g_{18} = 6L\alpha l_{22} l_{33} - 6L\beta l_{32} l_{23}$$

$$g_{19} = 6L\alpha l_{23} l_{31} - 6L\beta l_{33} l_{21}$$

$$g_{20} = 6L\alpha l_{23} l_{32} - 6L\beta l_{33} l_{22}$$

$$g_{21} = 6L\alpha l_{23} l_{33} - 6L\beta l_{33} l_{23}$$

$$g_{22} = -\delta l_{11}^2 + 2L^2\alpha l_{31}^2 + 2L^2\beta l_{21}^2$$

$$g_{23} = -\delta l_{11} l_{12} + 2L^2\alpha l_{31} l_{32} + 2L^2\beta l_{21} l_{22}$$

$$g_{24} = -\delta l_{11} l_{13} + 2L^2\alpha l_{31} l_{33} + 2L^2\beta l_{21} l_{23}$$

$$g_{25} = -\delta l_{12}^2 + 2L^2\alpha l_{32}^2 + 2L^2\beta l_{22}^2$$

$$g_{26} = -\delta l_{12} l_{13} + 2L^2\alpha l_{32} l_{33} + 2L^2\beta l_{22} l_{23}$$

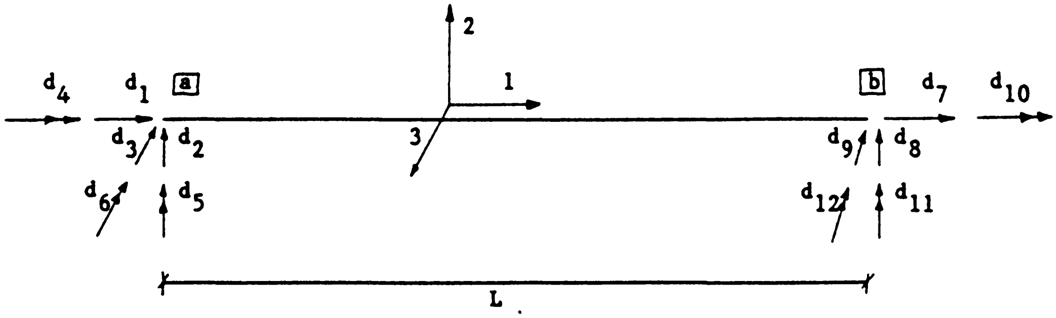
$$g_{27} = -\delta l_{13}^2 + 2L^2\alpha l_{33}^2 + 2L^2\beta l_{23}^2$$

where

$$\alpha = \frac{EI_3}{L^3}, \quad \beta = \frac{EI_2}{L^3}, \quad \gamma = \frac{EA}{L}, \quad \delta = \frac{GJ}{L}$$

**Taken from Holzer, 1985**

**Figure 4.5. Element stiffness matrix INDEX coefficients for stringerboard, deckboard, and block elements.**



$f_1 = \gamma(d_1 - d_7)$

$f_2 = 12\alpha d_2 + 6L\alpha d_6 - 12\alpha d_8 + 6L\alpha d_{12}$

$f_3 = 12\beta d_3 - 6L\beta d_5 - 12\beta d_9 - 6L\beta d_{11}$

$f_5 = -6L\beta d_3 + 4L^2\beta d_5 + 6L\beta d_9 + 2L^2\beta d_{11}$

$f_6 = 6L\alpha d_2 + 4L^2\alpha d_6 - 6L\alpha d_8 + 2L^2\alpha d_{12}$

$f_{11} = -Lf_3 - f_5$

$f_{12} = Lf_2 - f_6$

$$\sigma_a = \left|\frac{f_1}{A}\right| + \left|\frac{f_5}{S_2}\right| + \left|\frac{f_6}{S_3}\right| \quad ; \quad \sigma_b = \left|\frac{f_1}{A}\right| + \left|\frac{f_{11}}{S_2}\right| + \left|\frac{f_{12}}{S_3}\right|$$

where; A = cross sectional area

$S_2$ = section modulus with reference to local 2-axis

$S_3$ = section modulus with reference to local 3-axis

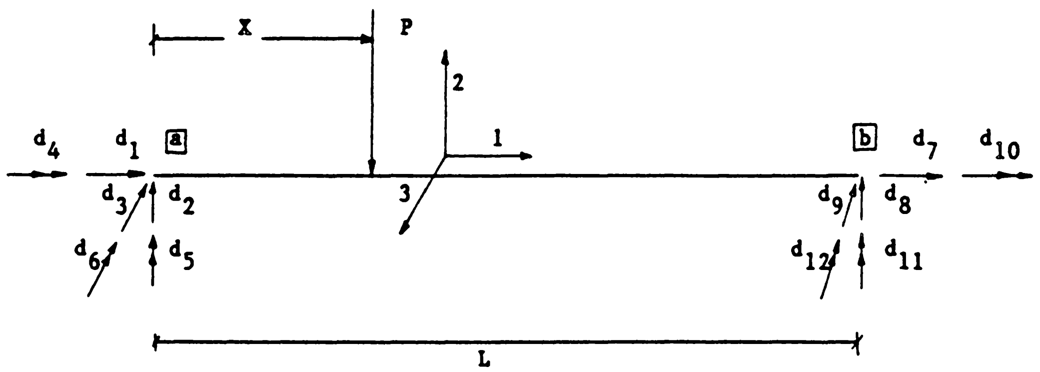
$d_i$ = element degree of freedom i

$\gamma = EA/L$

$\alpha = EI_3/L^3$ ; $I_3$ is the moment of inertia about the local 3-axis

$\beta = EI_2/L^3$ ; $I_2$ is the moment of inertia about the local 2-axis

Figure 4.6. Equations to compute stress from displacements - RAS



$$lf_1 = \gamma[(d_1 - d_7)] + \hat{f}(1,I)$$
$$lf_2 = [12\alpha d_2 + 6L\alpha d_6 - 12\alpha d_8 + 6L\alpha d_{12}] + \hat{f}(2,I)$$
$$lf_3 = [12\beta d_3 - 6L\beta d_5 - 12\beta d_9 - 6L\beta d_{11}] + \hat{f}(3,I)$$
$$lf_5 = [-6L\beta d_3 + 4L^2\beta d_5 + 6L\beta d_9 + 2L^2\beta d_{11}] + \hat{f}(5,I)$$
$$lf_6 = [6L\alpha d_2 + 4L^2\alpha d_6 - 6L\alpha d_8 + 2L^2\alpha d_{12}] + \hat{f}(6,I)$$

$$Mmax = -lf_6 + lf_2X + lf_5$$

$$\sigma = \left|\frac{f_1}{A}\right| + \left|\frac{Mmax}{S_3}\right|$$

where; A = cross sectional area

$S_2$ = section modulus with reference to local 2-axis

$S_3$ = section modulus with reference to local 3-axis

$d_i$ = element degree of freedom i

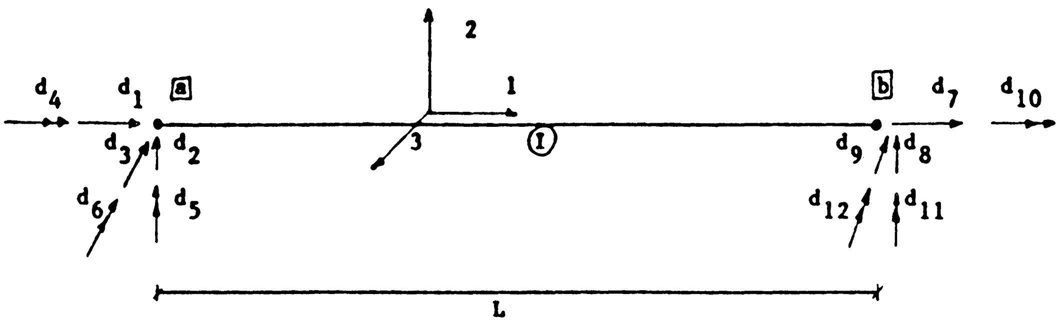
$\gamma$ = EA/L

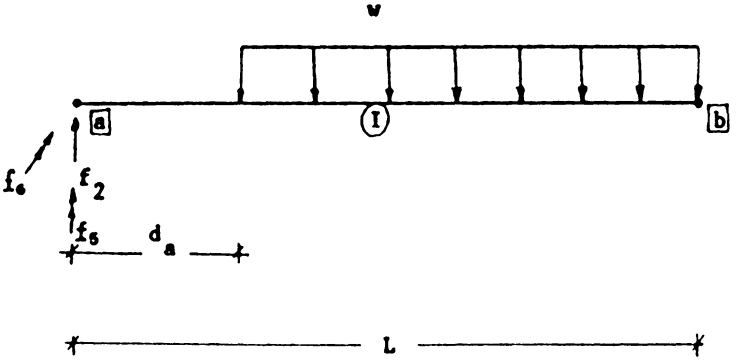
$\alpha = EI_3/L^3$ ; $I_3$ is the moment of inertia about the local 3-axis

$\beta = EI_2/L^3$ ; $I_2$ is the moment of inertia about the local 2-axis

$lf_N = \hat{f}(N,I) + f_n$

Figure 4.7. Equations to calculate stress in stringerboards subject to line loads.





$$Xmax = d_a + lf(2,I)/w$$

$$Mmax = - lf(6,I) + lf(2,I)*(Xmax+d_a/2) + lf(5,I)$$

$$\sigma = |lf(1,I)/A| + |Mmax/S_2| + |Mmax/S_3|$$

where; $\sigma$ = Maximum bending stress

Mmax = Maximum bending moment

Figure 4.8. Equations to compute stresses deckboard elements

# 5.0 ANALYSIS OF RACKED ACROSS DECKBOARDS SUPPORT MODE (RAD):

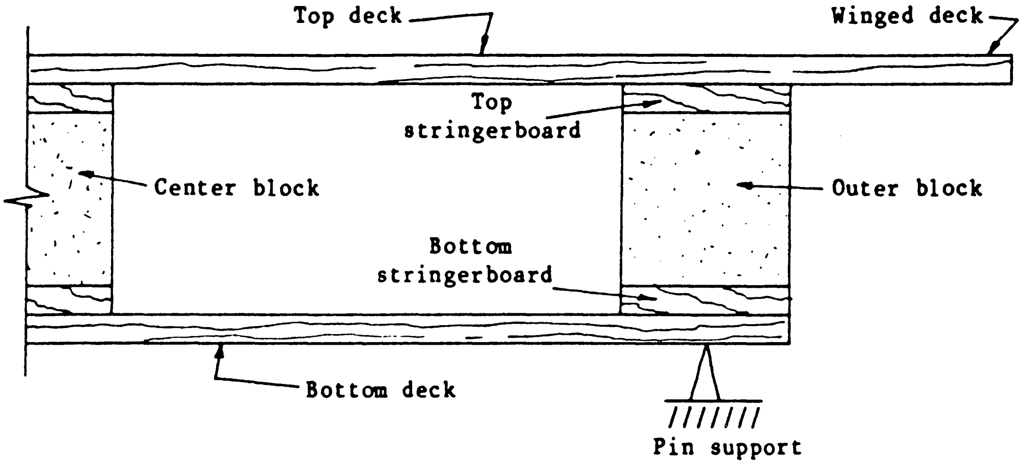
The previous chapter outlines the methodology used to analyze pallet response in the racked across stringers (RAS) support mode. This chapter details similar methodology for designing with the racked across deckboards support condition (RAD).

The applied load is translated into load effects (i.e stress and deflection ) in the pallet which are compared to the resistance of the components and any serviceability criteria. The overall strength of the pallet is governed by the maximum load effect. In the RAD mode this is assumed to be in either the top or bottom deckboard elements.

## 5.1 Assumptions and limitations

The two dimensional model used in the RAD mode is shown in Figure 5.1. The top deck is modeled as a single deckboard with its width equal to the summation of all the top deckboard widths. Similarly, the bottom deck is the summation of all bottom deckboards. Winged top deck members are included in the model and are given zero properties if absent from the pallet. For the sling support condition joint 15 is taken as the location of the support joint. The two possible RAD support joints are 3 and 14. For stability reasons, joint 3 is located near the inside edge of the block. The outside block is modeled as a combination of rigidly interconnected members; the center block is modeled as a single vertical element.

a) cross section of a real pallet:



b) RAD model:

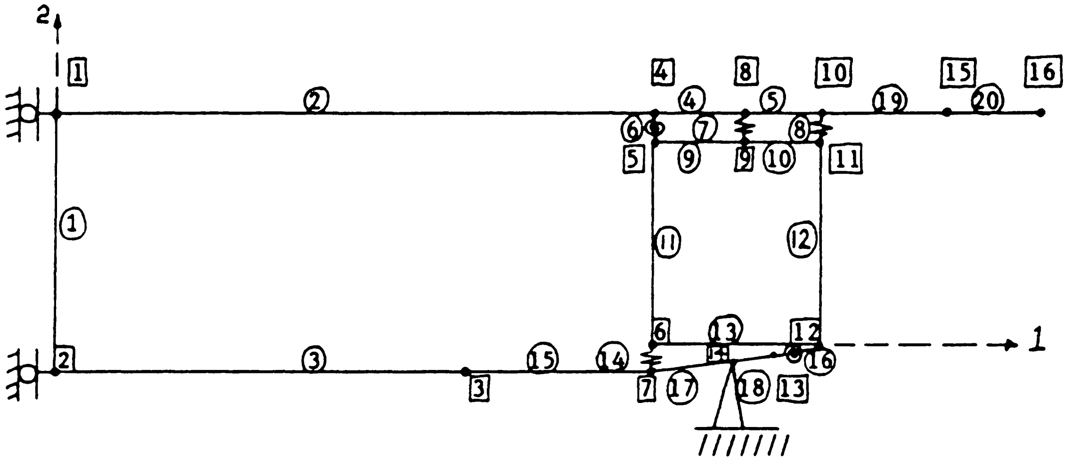


Figure 5.1. Analog model of RAD pallet

With stringer - type pallets, the inner stringers of three-and four-stringer pallets were assumed to cause the top and bottom decks to deflect equally. For block-type pallets, racked across the deckboards, the centerpoint of the middle stringerboard deflects more than do the ends, suggesting two-dimensional bending of the structure. However, from test observation and SPACEPAL results, the difference in deflection is negligible and hence, only bending about the global-3 axis is considered. For pallet geometries with no centerline blocks the maximum deflection will be in the top deck.

## 5.2 RAD Analog model

### 5.2.1 General model

The RAD analysis procedure utilizes a model which simulates the behavior of a full pallet subject to a unit load in the RAD mode. This model was developed after observing the actions of pallets tested in the RAD support condition. The most important feature of the model is the deck-to-block joint and how it mimics the action of the deckboard relative to the block. From observations, the top deck was seen to pivot around the inside edge of the block. However, the bottom deck pivots around some point on or near the block depending on the support location (Fig.5.2). For example, if the support is located under the block to the right of its centerline, the joint tends to open when the pallet bends. If the support is to the left of the centerline a greater degree of fixity is seen. When the support is located at joint 3 (not under the block) the bottom joint becomes stiffer with increased load.

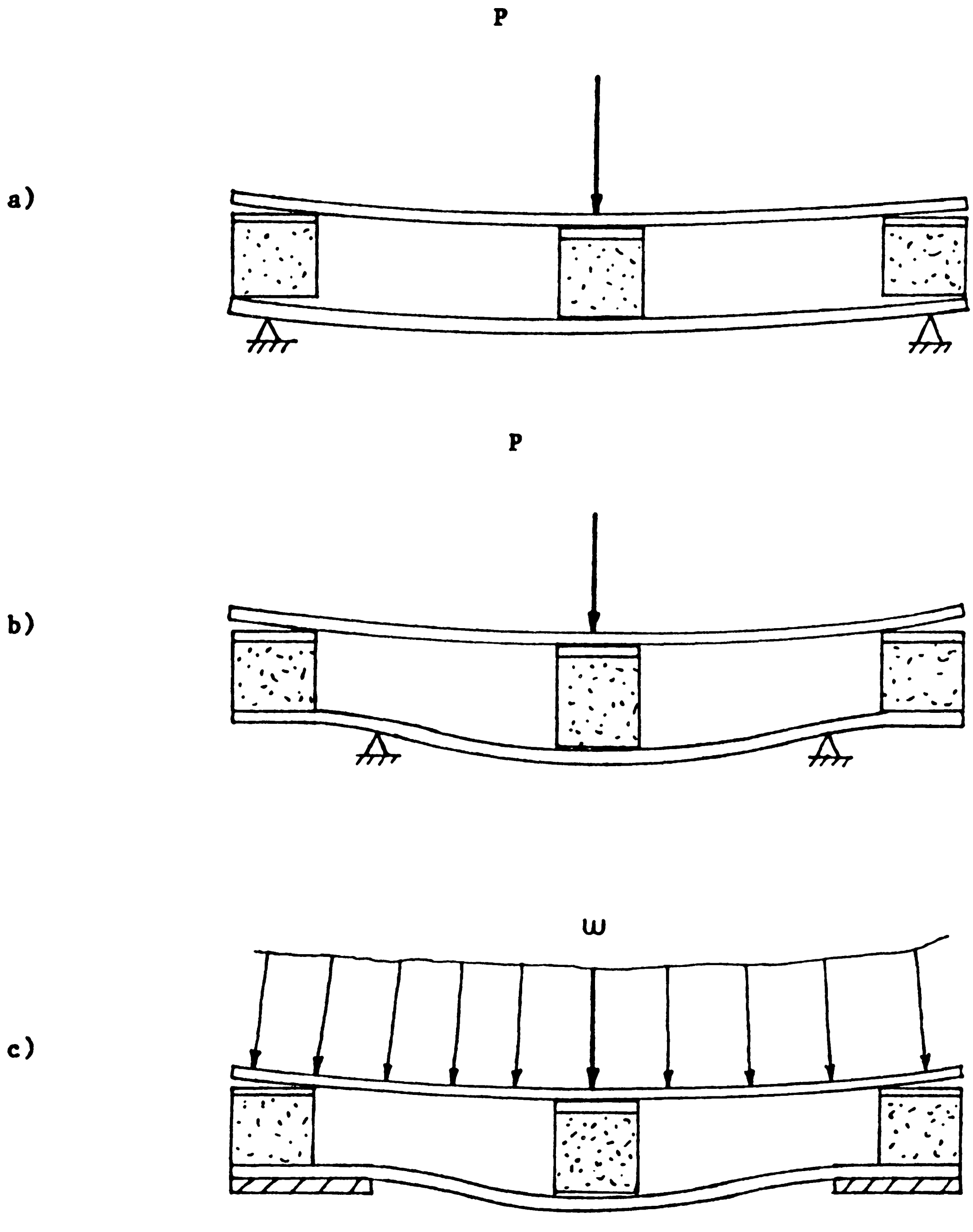
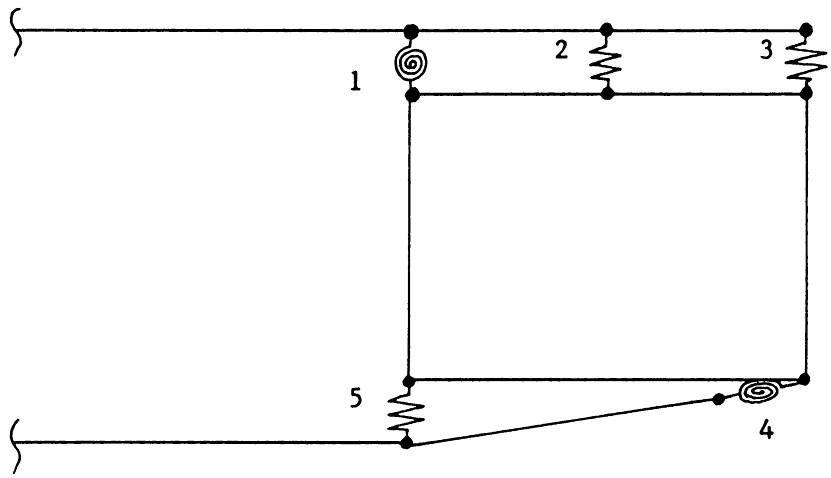


Figure 5.2. Pivot point for RAD support mode.

To accurately predict the response of a pallet to applied loads, the model must account for all these joint actions. The selected model considers the block as a rigidly interconnected framework of elements connected to the top and bottom decks by zero length springs (Figure 5.3). These springs are assigned stiffness values corresponding to lateral, withdrawal, and rotational stiffnesses obtained from test sections. The actions of a block pallet in the RAD mode are simulated with a two dimensional model. This neglects bending about the global-2 axis but results in a model that has fewer degrees of freedom than the three dimensional space frame model that was used with the RAS mode. Hence, the required computer memory and computation time is reduced.

The model is set up as the right half of a symmetric pallet as shown in Figure 5.1. The actions of 4, 6, 8, and 9 block pallets are simulated by assigning the appropriate member properties to the elements of the model. For example, four and six block pallets require that member 1 be assigned zero stiffness.

The model is initially solved with axial spring elements 7, 8, and 14 given zero stiffness values in all directions. The model is then checked for compatibility and their axial stiffness value may be adjusted depending on the support location. For example, if the support is at joint 3 then spring element 14 will have a high axial stiffness to ensure that joint 7 does not pass joint 6. For the case of a winged pallet similar conditions are anticipated for spring elements 7 and 8. These compatibility conditions are insured in the program code.



Rotational springs - 1 and 4

Axial springs - 2 , 3 and 5

Figure 5.3. Top and bottom deck connections to block.

To fully utilize symmetry, the model includes shear releases at the centerline (i.e at joints 1 and 2). These restrain translation in the 1 - direction and rotation about the 3 - axis.

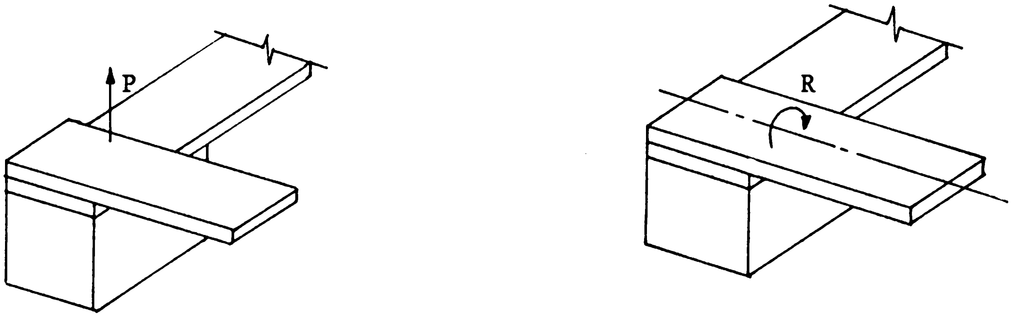
### 5.2.2 Deckboard - block joint characterization

To simulate the semi - rigid action of the deckboard - block joint five springs are used to represent the top and bottom deck connections to the block (Figure 5.3). In a 3 - dimensional system each joint has six possible degrees of freedom: three translational stiffnesses along the three global axes (i.e parallel and perpendicular to the grain of the deckboard and withdrawal stiffness), in-plane and out-of- plane rotational stiffness, and twisting about the global 1 - axis. These are shown in Figure 5.4.

With the two dimensional RAD model the only relevant stiffnesses are translation parallel to the deckboard length, withdrawal stiffness, and out of plane rotational stiffness. From sensitivity studies carried out on full pallets and pallet sections using SPACEPAL, the most important joint parameter was found to be the rotational stiffness. Rotational and withdrawal stiffnesses (rotation and separation moduli) are assigned values representative of test results. Two springs (6 and 16) are assumed free and rigid respectively in the global 1 and 2 directions, and springs 8 and 14 are initially given zero stiffness in withdrawal. Depending on the loading condition, geometry of the pallet, and support

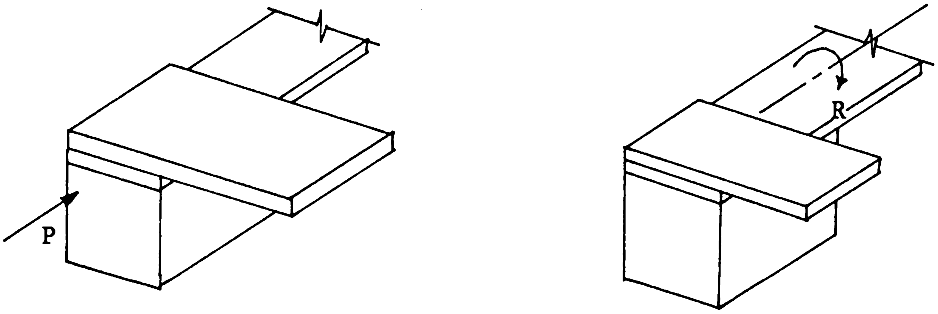
1) Lateral - parallel to deckboard

4) In - plane rotation



2) Withdrawal

5) Twisting



3) Lateral - parallel to the stringerboard

6) Out of plane rotation (Rotation modulus)

Figure 5.4. Six possible stiffness components of deck - block joints.

condition, the axial stiffness of springs 8 and 14 can change to insure geometric compatibility (see Figure 5.11).

Values for rotational stiffness for nailed joints were determined by Samarasinghe (1987) for different species and nail combinations. The rotation modulus is dependent on edge crushing of the block, head pull through and withdrawal stiffness. For both the lateral stiffness and rotation modulus the values derived for a single nailed joint are multiplied by the number of nails on one side of the pallet to determine their effective value.

### 5.2.3 Wing pallet modifications

For winged pallets, members 19 and 20 are used to simulate a wing and are given properties and dimensions similar to the rest of the top deck elements. However for flush pallets these elements are assigned zero stiffness. For the sling support condition the point of support is located at joint 15. It is free to move in the 1 - direction and to rotate about the 3 - axis. All members are assigned real properties corresponding to those of the elements of the pallet.

## 5.3 Analysis methodology

The RAS model in the previous chapter contained a variable number of elements depending on the number of top deckboards in the pallet. The analysis procedure uses an automatic assembly technique to compute the system stiffness matrix. This technique is also used in the RAD model. However, the RAD model has a fixed number of elements representing any pallet geometry. Modifications to the stiffness matrix due to support placement are automatically carried out in the assembly technique. The algorithm of the assembly and solution techniques are outlined in section 3.4.

### 5.3.1 Definition of the structure:

a.) The model is set up as a half symmetric version of the full pallet (either winged or flush). Member incidences are fixed for all geometries in "MAIN" with the numbering scheme shown in Figure 5.1.

b). The $\{JCODE\}$ and $\{MCODE\}$ arrays are defined and the unknown joint displacements are identified. The $\{JCODE\}$ is initially defined in MAIN and later modified to account for constraints in subroutine "CODES" corresponding to the degrees of freedom associated with each joint in the model. The numbering sequence of the degrees of freedom may change depending on the support location as shown in Figures 5.5 - 5.7. From this, $\{MCODE\}$ is defined for each element in subroutine "CODES", matching the unknown end displacements with the 'a' and 'b' ends of the members. The degree of freedom numbers of the joint at the 'a' end of the member are transferred into the first three cells of $\{MCODE\}$ and those for the 'b' end into the second three cells.

c). The material properties for each element are defined in subroutine "PROPS". These are set to represent the members in the actual pallet. The following are calculated or defined for each member in the model:-

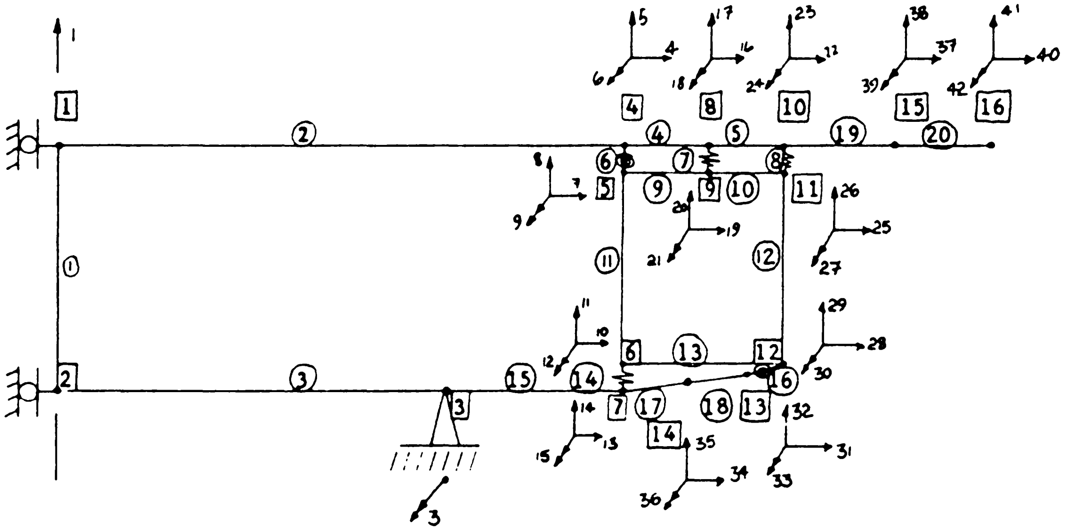


Figure 5.5 Possible joint motions used to develop JCODE for RAD support at joint 3

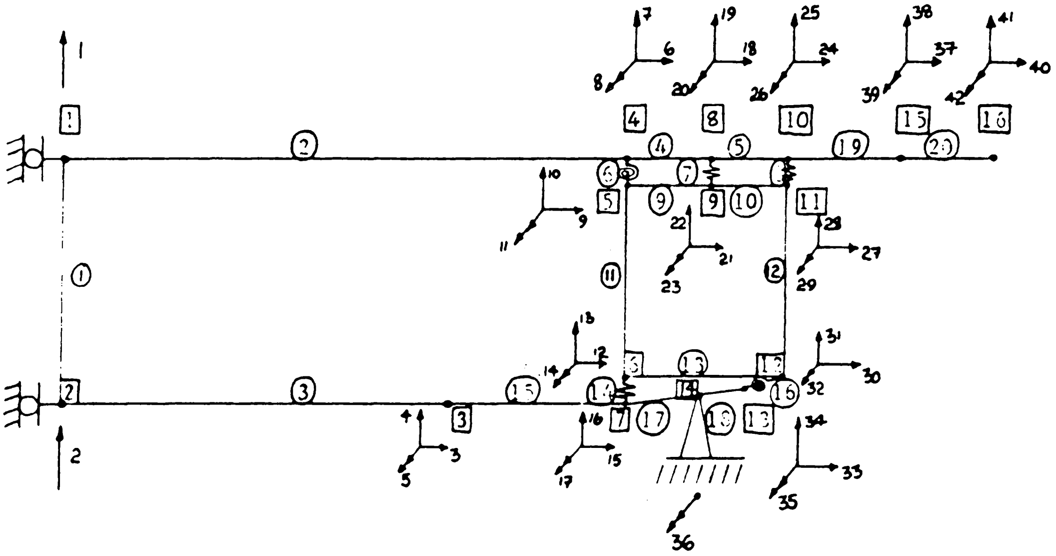


Figure 5.6. Possible joint motions used to develop JCODE for RAD support at joint 14

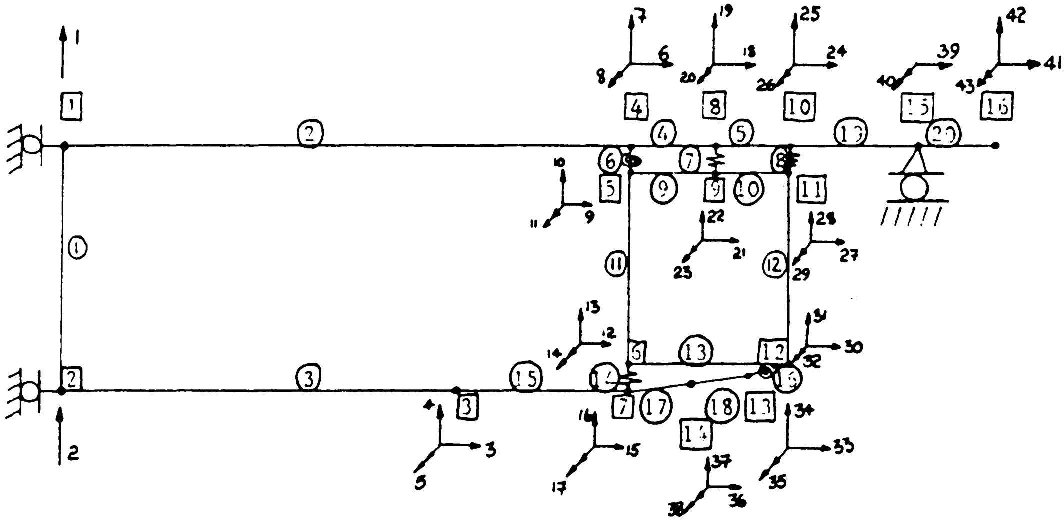


Figure 5.7. Possible joint motions used to develop JCODE for RAD support at joint 15

- Modulus of elasticity, XMOE
- Area, A(I) = W(I) * T(I)
- Moment of inertia about the 3-axis

  $$EI(I) = W(I)\ T(I)^3/12$$

- Section modulus

  $$S(I) = EI(I)\ 2.0/T(I)$$

- Modulus of rigidity

  $$XMG(I) = XMOE(I)\ /\ 16$$

- Length, EL(I)

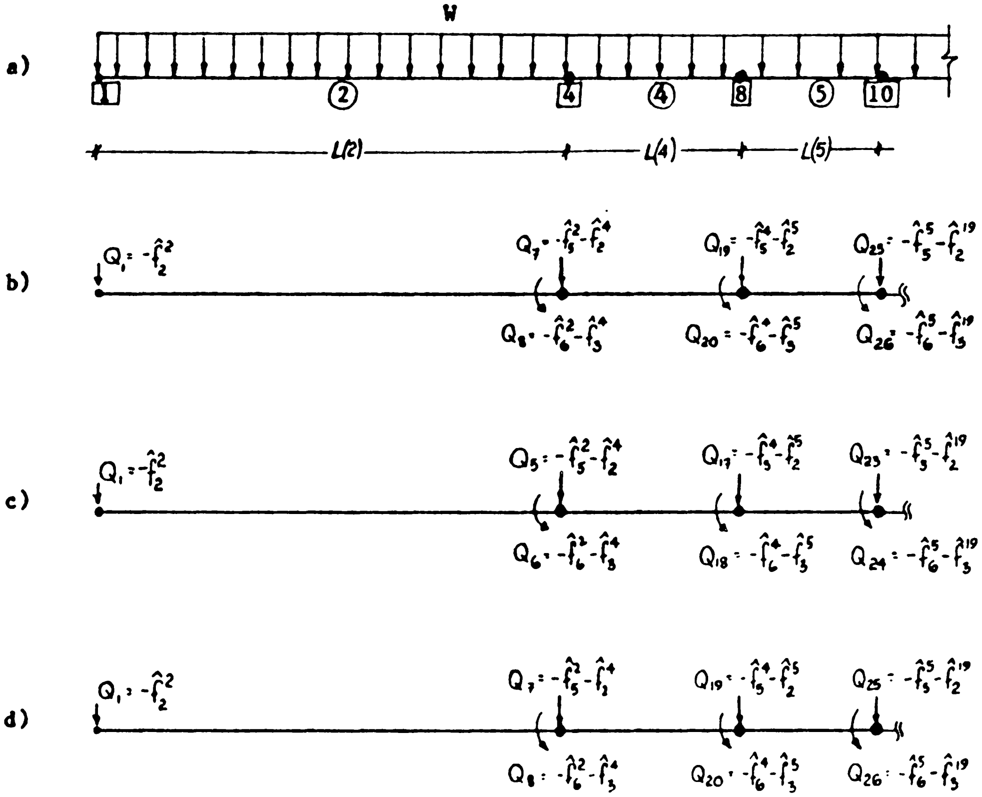
The direction cosines, which transform the elements from their local reference planes to the global reference plane, are also defined for each element in "PROPS".

### 5.3.2 Equivalent joint loads:

For each load type the member loads are transformed into equivalent joint loads and assembled into an array, Q . Two conditions are allowed: Uniform load (full or partial) and line load.

a. Uniform loads - may fully or partially cover the entire top deck. The equations to calculate the equivalent joint loads are defined in subroutine "UDL" and are shown in Figures 5.8, and 5.9.

b. Line loads - 1, 2, or 3 symmetrically placed line loads are simulated by point loads on the top deck members in the model.



For i = 2, 4, 5

$$\hat{f}_2^{\,i} = \hat{f}_5^{\,i} = WL(i)/2 \;;\quad \hat{f}_3^{\,i} = WL(i)^2/12 \;;\quad \hat{f}_6^{\,i} = -\hat{f}_3^{\,i}$$

where : $\hat{f}_n^{\,i}$ = local fixed-end force in the nth direction for member i

Figure 5.8. Load models for full uniform load for all support conditions.
a) Full uniform load over the entire top deck
b) Equivalent joint loads for support at joint 14
c) Equivalent joint loads for support at joint 3
d) Equivalent joint loads for support at joint 15

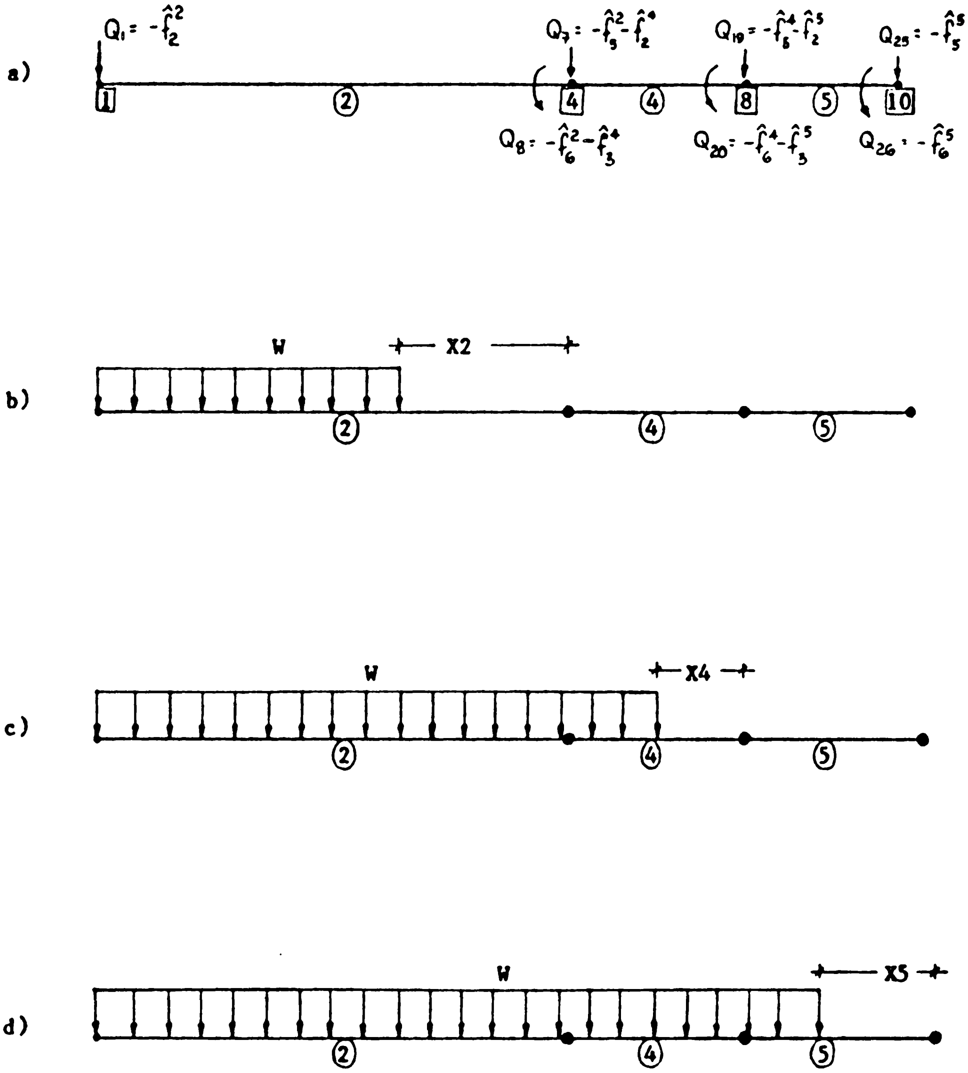


Figure 5.9. Load models for partial uniform load over the top deck
a) fixed end forces corresponding to degree of freedom of structure.
b) Partial uniform load over member 2
c) Partial uniform load over member 4
d) Partial uniform load over member 5

| Member: | Equivalent fixed end forces: |
|---|---|
| 2 | $\hat{f}_5^{\,2} = W/2L^3[2L^3(L - X2) - 2L(L^3 - X2^3) + (L^4 - X2^4)]$<br>$\hat{f}_2^{\,2} = W(L - X2) - f_5^{\,2}$<br>$\hat{f}_3^{\,2} = W/12L^2[4L(L^3 - X2^3) - 3(L^4 - X2^4)]$<br>$\hat{f}_6^{\,2} = -W/12L^2[6L^2(L^2 - X2^2) - 8L(L^3 - X2^3) + 3(L^4 - X2^4)]$ |
| 4 | $\hat{f}_5^{\,4} = W/2L^3[2L^3(L - X4) - 2L(L^3 - X4^3) + (L^4 - X4^4)]$<br>$\hat{f}_2^{\,4} = W(L - X4) - f_5^{\,4}$<br>$\hat{f}_3^{\,4} = W/12L^2[4L(L^3 - X4^3) - 3(L^4 - X4^4)]$<br>$\hat{f}_6^{\,4} = -W/12L^2[6L^2(L^2 - X4^2) - 8L(L^3 - X4^3) + 3(L^4 - X4^4)]$ |
| 5 | $\hat{f}_5^{\,5} = W/2L^3[2L^3(L - X5) - 2L(L^3 - X5^3) + (L^4 - X5^4)]$<br>$\hat{f}_2^{\,5} = W(L - X5) - f_5^{\,5}$<br>$\hat{f}_3^{\,5} = W/12L^2[4L(L^3 - X5^3) - 3(L^4 - X5^4)]$<br>$\hat{f}_6^{\,5} = -W/12L^2[6L^2(L^2 - X5^2) - 8L(L^3 - X5^3) + 3(L^4 - X5^4)]$ |

L = length of member

Figure 5.9a. Equations to compute the fixed end forces for a partial uniform load over the top deck

In all three cases the loads are allowed to act only on member 2. Because of symmetry, the load applied at the centerline (joint 1) is one half the applied load for 1 and 3 line loads. The equations to calculate the equivalent joint load vector used in subroutine "LINE" are shown in Figure 5.10.

### 5.3.3 Stiffness matrix:

The local element stiffness matrices are next defined and then assembled into the system matrix. There are three element types employed in the RAD model requiring three different element stiffness matrices. These types define a:

a. zero length rotational spring,
b. zero length axial spring,
c. stringerboard, deckboard, or block element.

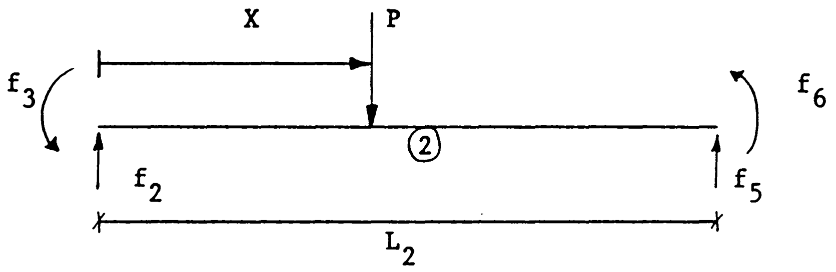
These matrices are shown in Figure 5.11 and are coded in subroutine "MELEMS".

The $\{MCODE\}$ is used to fill the cells of the system stiffness matrix [K]. It identifies the members which influence each cell and assembles it, element by element into the stiffness matrix in subroutine "ASSEMS".

### 5.3.4 Solution and stress computation:

The displacement vector D in the basic matrix equation

$$\{F\} = [K]\{D\} \qquad [5.4]$$



$$\hat{f}(2,2) = P/L_2^{\,3}\ [(L_2 - X)^2(L_2 + 2X)]$$

$$\hat{f}(3,2) = P\ X\ [(L_2 - X)/L_2]^2$$

$$\hat{f}(5,2) = P\ X^2\ (3L_2 - 2X)/L_2^{\,3}$$

$$\hat{f}(6,2) = -P\ (L_2 - X)\ (X/L_2)^2$$

Figure 5.10. Computation of fixed end forces for member 2 subject to line loads

$$\text{INDEX} = \begin{bmatrix} G_1 & G_2 & G_4 & -G_1 & -G_2 & G_4 \\ & G_3 & G_5 & -G_2 & -G_3 & -G_5 \\ & & G_6 & -G_4 & -G_5 & -G_7 \\ & & & G_1 & G_2 & -G_4 \\ & \text{Symmetric} & & & G_3 & -G_5 \\ & & & & & G_6 \end{bmatrix}$$

G = Global element stiffness coefficient

where;

| Stringerboard, block deckboard elements | | Rotational springs | | | Axial springs | | | |
|---|---|---|---|---|---|---|---|---|
| | | | 6 | 16 | | 7 | 8 | 14 |
| $G_1 =$ | $\alpha(\beta C_1^2 + 12C_2^2)$ | $G_1 =$ | 0 | $1\text{x}10^7$ | $G_1 =$ | 0 | 0 | 0 |
| $G_2 =$ | $\alpha C_1 C_2(\beta - 12)$ | $G_2 =$ | 0 | 0 | $G_2 =$ | 0 | 0 | 0 |
| $G_3 =$ | $\alpha(\beta C_2^2 + 12C_1^2)$ | $G_3 =$ | 0 | $1\text{x}10^6$ | $G_3 =$ | Gam4 | Gam6 | Gam5 |
| $G_4 =$ | $-\alpha 6LC_2$ | $G_4 =$ | 0 | 0 | $G_4 =$ | 0 | 0 | 0 |
| $G_5 =$ | $\alpha 6LC_1$ | $G_5 =$ | 0 | 0 | $G_5 =$ | 0 | 0 | 0 |
| $G_6 =$ | $\alpha 4L^2$ | $G_6 =$ | Gam3 | Gam3 | $G_6 =$ | 0 | 0 | 0 |
| $G_7 =$ | $\alpha 2L^2$ | $G_7 =$ | -Gam3 | -Gam3 | $G_7 =$ | 0 | 0 | 0 |

$\alpha = EI_3/L^3$ ; $\beta = AL^2/I_3$

$C_1$, $C_2$ = Direction cosines

Gam3 = Rotational modulus

Gam4 = Separation modulus

Gam5, Gam6 are initially zero

Figure 5.11. Element stiffness matrices for rotational and axial springs, and stringerboards, deckboards and block elements

where $\{F\}$ = global element force vector.

is solved by the Cholesky method in subroutine "SOLVE". From the resulting element end displacements the element stresses are calculated in subroutine "MSTRES". These are computed for the 'a' and 'b' ends of each deckboard member using the formulae in Figure 5.12. The element end stresses are determined for all top and bottom deckboard elements. Bending stresses are calculated in the loaded top deckboard members as follows:-

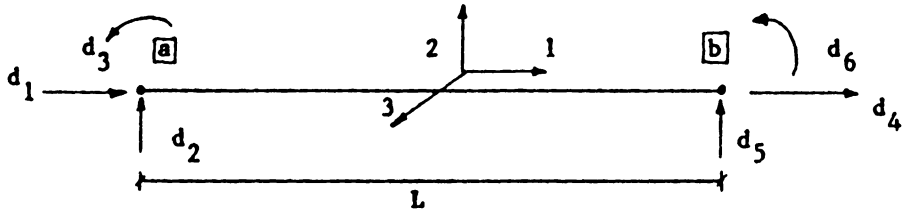
a. Members 2, 4, and 5 - stresses are computed using the equations shown in Figure 5.13. Bending stresses are determined in all three members for a uniform load and only in member 2 for point loads (Figure 5.14).

b. Members 3 and 15 - stresses are computed using equations shown in Figure 5.15 if the support location is at joint 3.

The element with the maximum critical stress, and its location are determined and reported to the user. The vertical deflection is checked for all joints in the model and the maximum value is identified and reported.

### 5.3.5 Winged pallet and sling support condition :

No modifications to the basic model or the procedure used to set up the system stiffness matrix are necessary for a winged pallet. The only loading condition which members 19 and 20 can experience is a



$$f_1 = EA/L\ (d_1 - d_4)$$

$$f_2 = 12EI/L^3\ (d_2 - d_5) + 6EI/L^2\ (d_3 + d_6)$$

$$f_3 = 6EI/L^2\ (d_2 - d_5) + 2EI/L\ (2d_3 + d_6)$$

$$f_6 = f_2 L - f_3$$

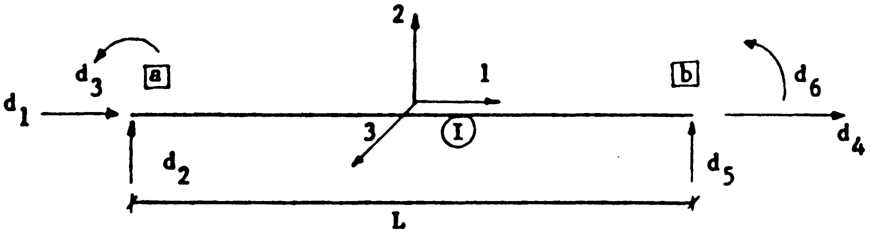
$$\sigma_a = \left|\frac{f_1}{A}\right| + \left|\frac{f_3}{S}\right| \quad ; \quad \sigma_b = \left|\frac{f_1}{A}\right| + \left|\frac{f_6}{S}\right|$$

where ; A = cross sectional area

S = section modulus

$d_i$ = element degree of freedom i

Figure 5.12. General equations to compute stress from displacements - RAD



$$f_1 = EA/L\ (d_1 - d_4)$$

$$f_2 = 12EI/L^3\ (d_2 - d_5) + 6EI/L^2\ (d_3 + d_6)$$

$$f_3 = 6EI/L^2\ (d_2 - d_5) + 2EI/L\ (2d_3 + d_6)$$

$$f_6 = f_2L - f_3$$

$$\sigma_a = \left|\frac{f_1}{A}\right| + \left|\frac{f_3}{S}\right| \quad ; \quad \sigma_b = \left|\frac{f_1}{A}\right| + \left|\frac{f_6}{S}\right|$$

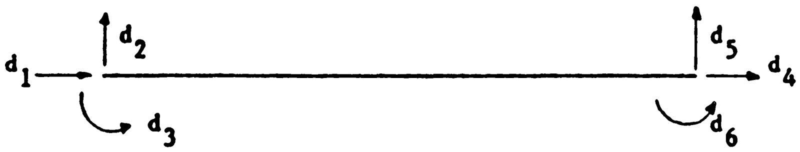
where ; A = cross sectional area

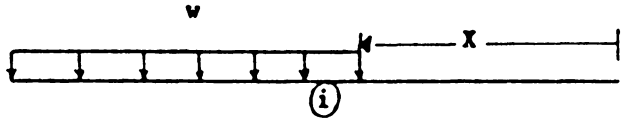
S = section modulus

$d_i$ = element degree of freedom i

I = 2, 4, 5

Figure 5.13. Equations to compute stress in members 2, 4, and 5 from displacements - RAD





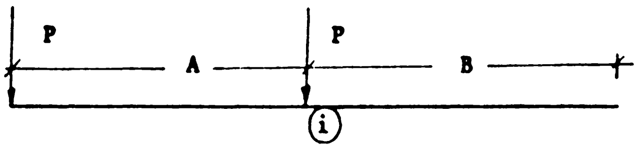
$i = 2, 4, 5$

$X_{max} = X + lf(5,i)/w$

$M_{max} = -lf(6,i) + lf(5,i)[(X_{max} + X)/2]$

$\sigma = |lf(1,i)/A(i)| + |M_{max}/S(i)|$

a) Uniform loads



$i = 2$

$M_{max} = - lf(3,i) + [lf(2,i) * A]$

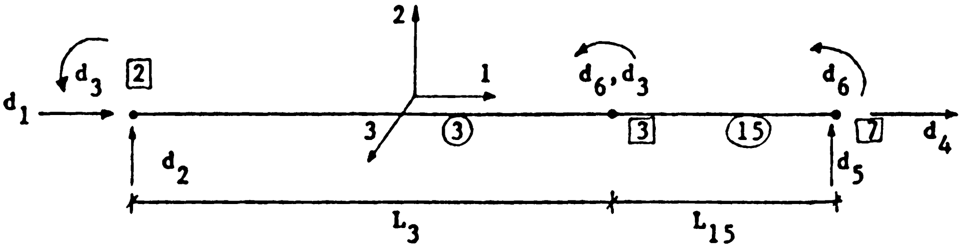
$\sigma = |lf(1,i)/A(i)| + |M_{max}/S(i)|$

b) Line loads

where: $lf(n,i) = \hat{f}_n^{\,i} + f_n$

$\sigma$ = Maximum bending stress

Figure 5.14 Equations to compute stress in top deck members.



a) member 3:

$$f_1 = EA/L_3\,(d_1)$$

$$f_2 = 12EI/L_3^{\,3}\,(d_2) + 6EI/L_3^{\,2}\,(d_3 + d_6)$$

$$f_3 = 6EI/L_3^{\,2}\,(d_2) + 2EI/L_3\,(2d_3 + d_6)$$

$$f_6 = f_2L_3 - f_3$$

b) member 15:

$$f_1 = EA/L_{15}\,(-d_4)$$

$$f_2 = 12EI/L_{15}^{\,3}\,(-d_5) + 6EI/L_{15}^{\,2}\,(d_3 + d_6)$$

$$f_3 = 6EI/L_{15}^{\,2}\,(-d_5) + 2EI/L_{15}\,(2d_3 + d_6)$$

$$f_6 = f_2L_{15} - f_3$$

---

$$\sigma_a = \left|\frac{f_1}{A}\right| + \left|\frac{f_3}{S}\right| \quad ; \quad \sigma_b = \left|\frac{f_1}{A}\right| + \left|\frac{f_6}{S}\right|$$

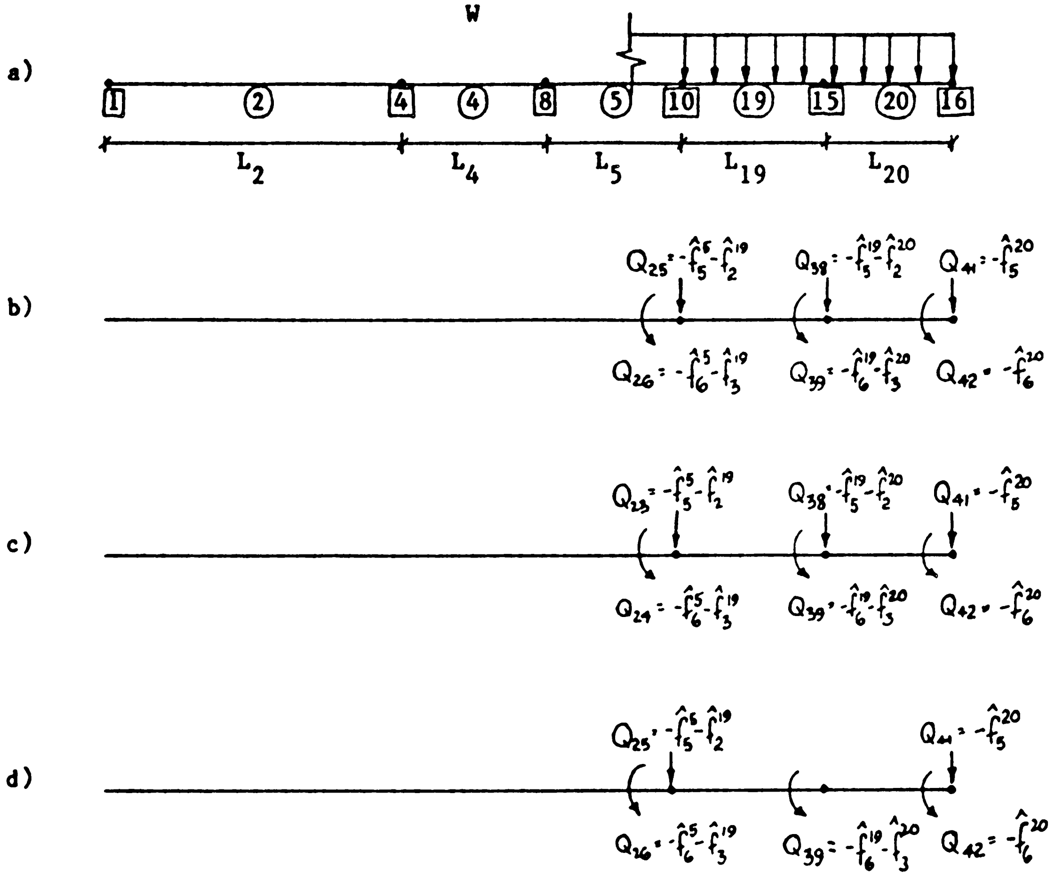
where ; A = cross sectional area of bottom deck

S = section modulus of bottom deck

Figure 5.15. Computation of stress in RAD members 3 and 15, with support at joint 3.

full uniform load over their lengths. The fixed end forces are calculated for these members as shown in Figure 5.16.

Element stresses are calculated for members 19 and 20 with no acting loads using the equations shown in Figures 5.17 & 5.18. When a uniform load is applied to members 19 and 20 the stresses are calculated using the equation in Figure 5.19.



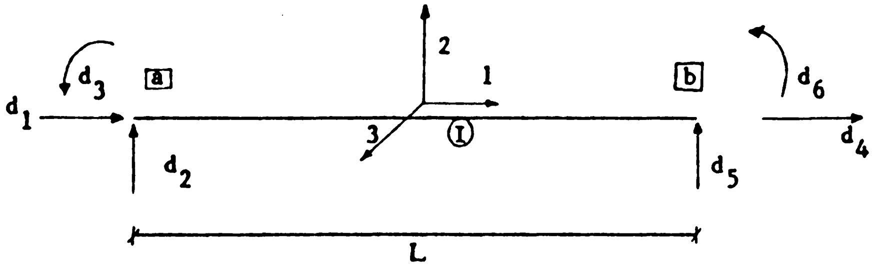
$\hat{f}_2^{19} = \hat{f}_5^{19} = WL(19)/2$

$\hat{f}_3^{19} = WL(19)^2/12$ ; $\hat{f}_6^{19} = -\hat{f}_3^{19}$

$\hat{f}_2^{20} = \hat{f}_5^{20} = WL(20)/2$

$\hat{f}_3^{20} = WL(20)^2/12$ ; $\hat{f}_6^{20} = -\hat{f}_3^{20}$

Figure 5.16. Load models for full uniform load on members 19 and 20 for all support conditions.
a) Full uniform load over the entire top deck
b) Equivalent joint loads for support at joint 14
c) Equivalent joint loads for support at joint 3
d) Equivalent joint loads for support at joint 15



$$f_1 = EA/L\ (d_1 - d_4)$$

$$f_2 = 12EI/L^3\ (d_2 - d_5) + 6EI/L^2\ (d_3 + d_6)$$

$$f_3 = 6EI/L^2\ (d_2 - d_5) + 2EI/L\ (2d_3 + d_6)$$

$$f_6 = f_2L - f_3$$

$$\sigma_a = \left|\frac{f_1}{A}\right| + \left|\frac{f_3}{S}\right| \quad ; \quad \sigma_b = \left|\frac{f_1}{A}\right| + \left|\frac{f_6}{S}\right|$$

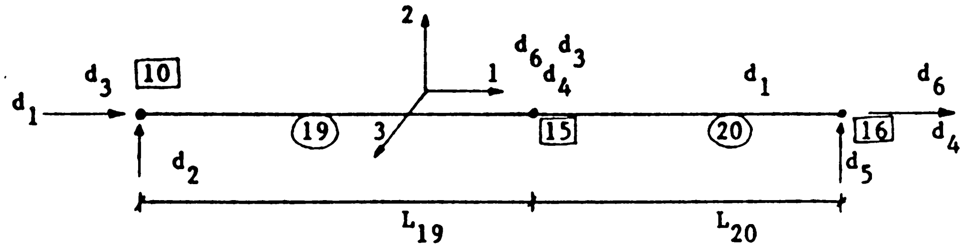
where ; A = cross sectional area

S = section modulus

$d_i$ = element degree of freedom i

I = 19, 20

Figure 5.17. Computation of stress in RAD members, 19, and 20 for support joints 3 and 14.



a) member 19:

$$f_1 = EA/L_{19}\,(d_1 - d_4)$$

$$f_2 = 12EI/L_{19}^3\,(d_2) + 6EI/L_{19}^2\,(d_3 + d_6)$$

$$f_3 = 6EI/L_{19}^2\,(d_2) + 2EI/L_{19}\,(2d_3 + d_6)$$

$$f_6 = f_2 L_{19} - f_3$$

b) member 20:

$$f_1 = EA/L_{20}\,(d_1 - d_4)$$

$$f_2 = 12EI/L_{20}^3\,(-d_5) + 6EI/L_{20}^2\,(d_3 + d_6)$$

$$f_3 = 6EI/L_{20}^2\,(-d_5) + 2EI/L_{20}\,(2d_3 + d_6)$$

$$f_6 = f_2 L_{20} - f_3$$

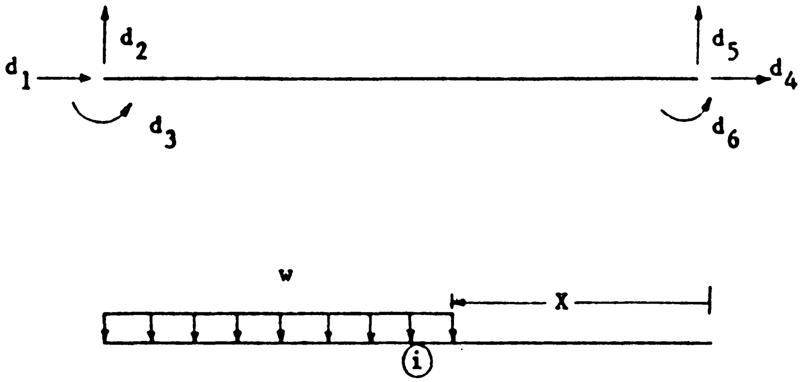
---

$$\sigma_a = \left|\frac{f_1}{A}\right| + \left|\frac{f_3}{S}\right| \quad ; \quad \sigma_b = \left|\frac{f_1}{A}\right| + \left|\frac{f_6}{S}\right|$$

where ; A = cross sectional area of top deck

S = section modulus of top deck

Figure 5.18. Computation of stress in RAD members 19 and 20, for sling support condition, (support at joint 15).



$i = 19,\ 20$

$Xmax = X + lf(5,i)/w$

$Mmax = -lf(6,i) + lf(5,i)[(Xmax + X)/2]$

$\sigma = \left|lf(1,i)/A(i)\right| + \left|Mmax/S(i)\right|$

where: $lf(n,i) = \hat{f}_n^{\,i} + f_n$

Figure 5.19 Equation to calculate the stresses in members 19 and 20 for a uniform load.

# 6.0 EXTENSION TO THE STACK CONDITION

The previous two chapters have outlined the analysis methodology used for block pallets in the RAD and RAS support modes. However, pallets are often stacked on one another or supported on a rigid base. The objective of this chapter is to extend the analysis methods to include the stacked condition.

## 6.1 Basic assumptions

As with the RAS and RAD support modes, the load effects are determined in terms of stress and deflection. For the stack condition the critical members are the top deck members of the bottom pallet and the bottom deck members of the second pallet in a stack. The method of load transfer is assumed to be different than the racked support modes. The blocks act as support columns and transmit some of the load directly to the floor surface. The remainder is distributed over the stringerboards and deckboards, as shown in Figure 6.1. The critical elements are then deck members stressed in bending. Depending on the distance between blocks and the number of boards, either stringerboards or deckboards may be critical.

The two models presented in chapters 4 and 5 are employed in the stacked analysis procedure. The RAS model is used to determine the load effects in the critical top stringerboard members and the parallel bottom deck members. The RAD model calculates the maximum load effects for the top deckboard members and the parallel bottom deck members. In both models

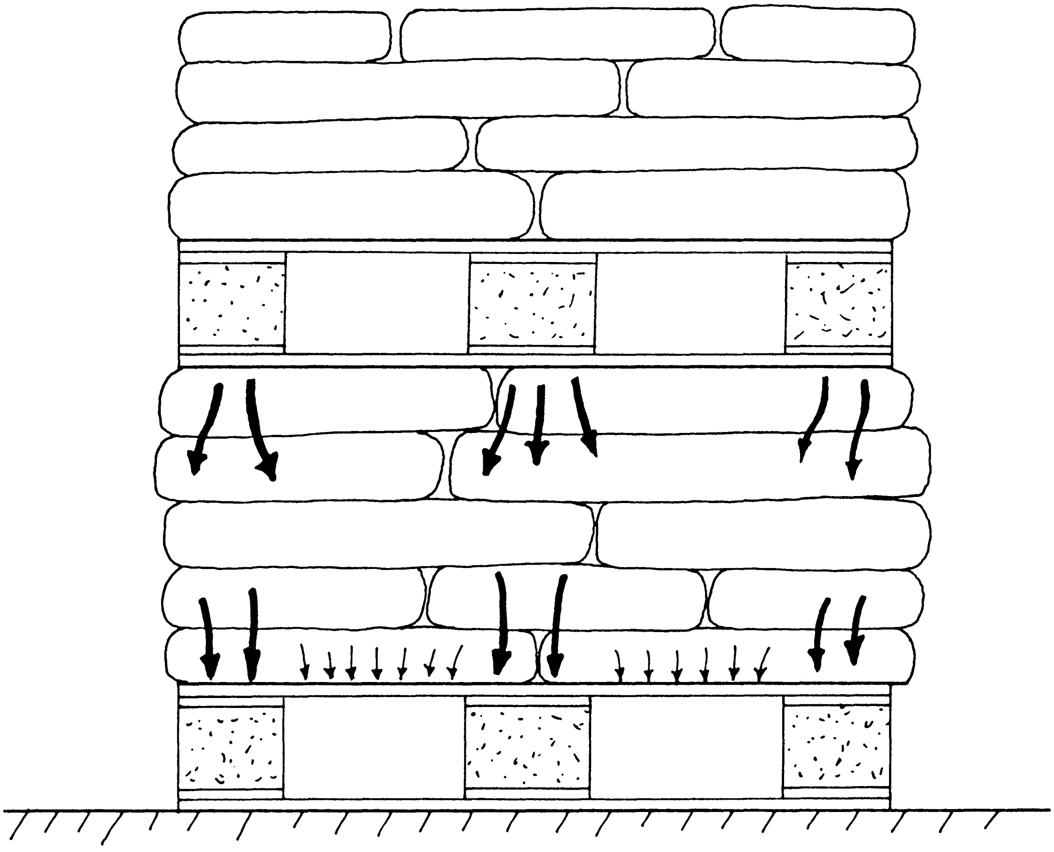


Figure 6.1. Schematic diagram showing the assumed load transfer in stacked mode

the top and bottom decks are considered to act independently of one another (i.e not as a composite structure). All assumptions and limitations inherent in the development of the RAS and RAD models also apply to the stack analysis.

## 6.2 Stack analog models

Modifications to both models redefine the analog structure to model the above actions. The boundary conditions which define the joint constraints were modified to establish a more efficient model to compute critical stresses and deflections. The solution routine used to calculate the load effects remains the same for all cases.

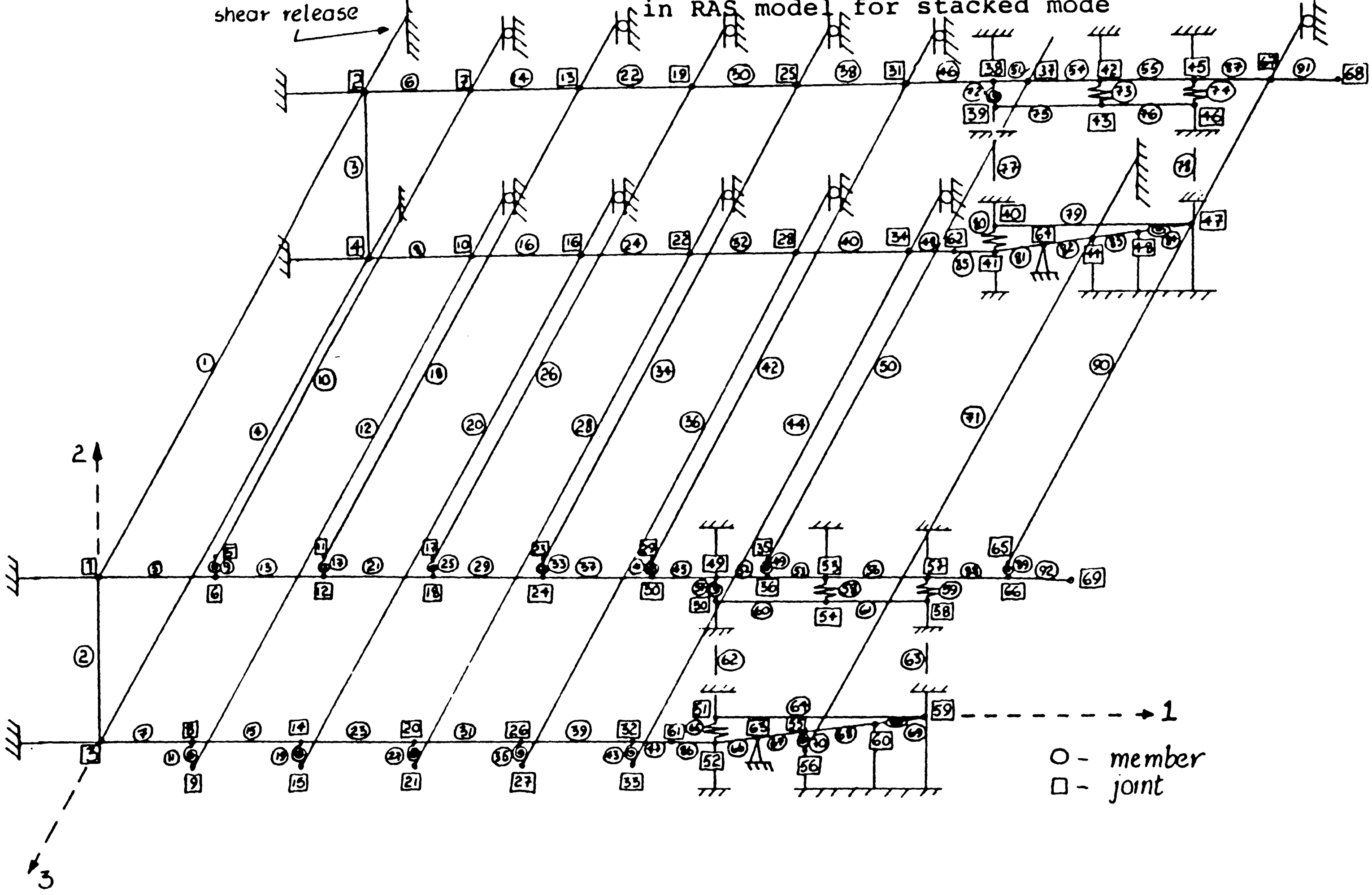
### 6.2.1 Modifications to the RAS analog model:

The model remains a quarter symmetric 3-dimensional structure with no modifications to either the number of elements or the number of joints. Shear releases along the two axes of symmetry allow vertical displacement of the joints located on the center lines. Modifications to the joint constraints for the stack condition are made in subroutine "STACKJ" as follows (see Figure 6.2):

a) Joints 1, 2, 3 and 4 are constrained in all 6 directions if elements 2 and 3 have non zero properties.

b) All joints which connect members in the 2 outer blocks are constrained in all 6 directions if the connecting members have non zero properties.

Figure 6.2. Modifications to joint constraints in RAS model for stacked mode



c) For members that have zero properties the incident joints are free to translate vertically (e.g if there is no center block then joints 2 and 4 are free to translate in the global 2 direction for the first or second pallet in a stack and only joint 2 is free to translate vertically for the bottom pallet).

One analysis is carried out for the STACK-RAS model with loads applied to both the top and bottom decks. The equivalent joint loads on the top deck are calculated as shown in chapter 4. The equivalent joint loads for the bottom deck are calculated in subroutine "STACKL" using the same procedure. The critical deflection is determined in subroutine "STACKD" for the top stringerboard elements and the bottom parallel boards. For pallets containing only 4 blocks the maximum deflection is calculated at joints 1 and 3 or 2 and 4. If there are three blocks per stringerboard the deflection is determined by Lagrangian interpolation between the calculated deflection at points in the span between the center block and the inside edge of the outer block. Only maximum deflection in the span is reported (see Figure 6.3). The method of calculating stress remains unchanged.

### 6.2.2 Modifications to the RAD analog model:

To mimic bending in the stack support condition an additional element and joint were added to the 2-dimensional RAD model as shown in Figure 6.4. Elements 2,3,21 and 15 are of equal length. Joints 3,4,and 17 are left free to rotate and translate. Maximum deflection is calculated at joint 17 when member 1 has non zero

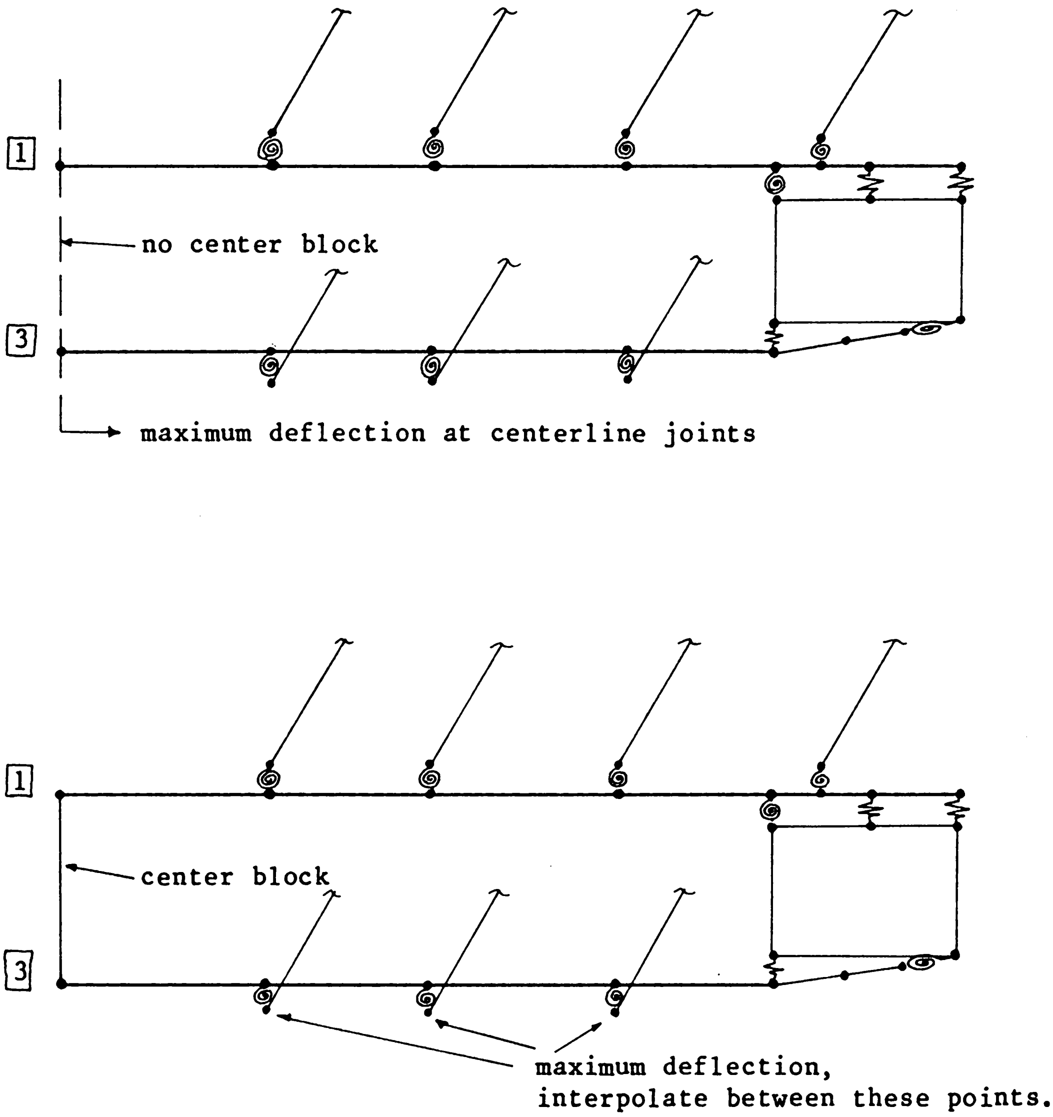


Figure 6.3. Measurement of maximum deflection in stringerboards for stacked condition

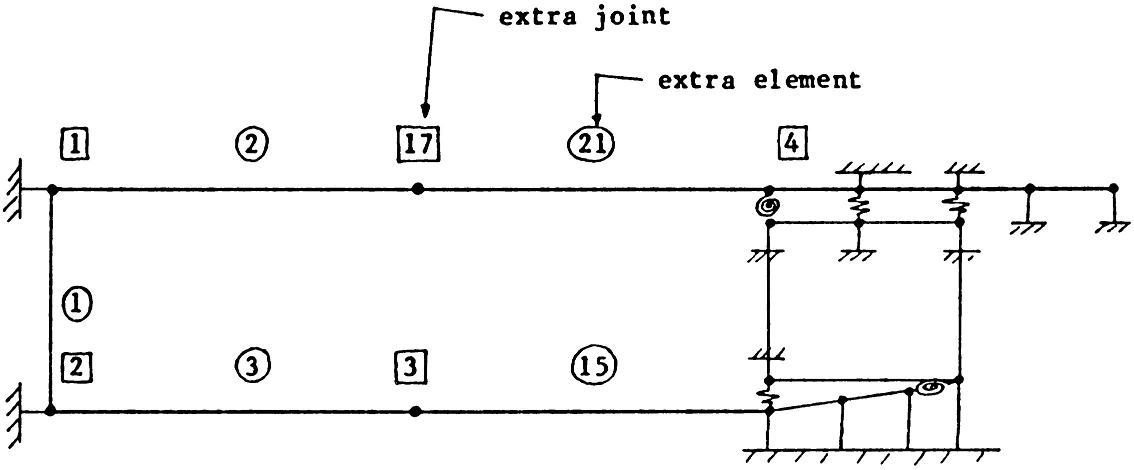


Figure 6.4. Modifications to RAD analog model for stacked condition

stiffness properties. For a 4-block pallet and a 6-block pallet(with three blocks per stringerboard), joints 1 and 2 are allowed to translate vertically and the maximum deflection is calculated at joint 1. A separate analysis is conducted for the top and bottom decks. Properties are assigned to the top members of the model representative of the top deck in the pallet and the load effects are determined. The bottom deck properties of the pallet are then assigned to the top deck members in the model and the analysis procedure is repeated. The method for computing maximum stress is unchanged.

# 7.0 EXPERIMENTAL VERIFICATION

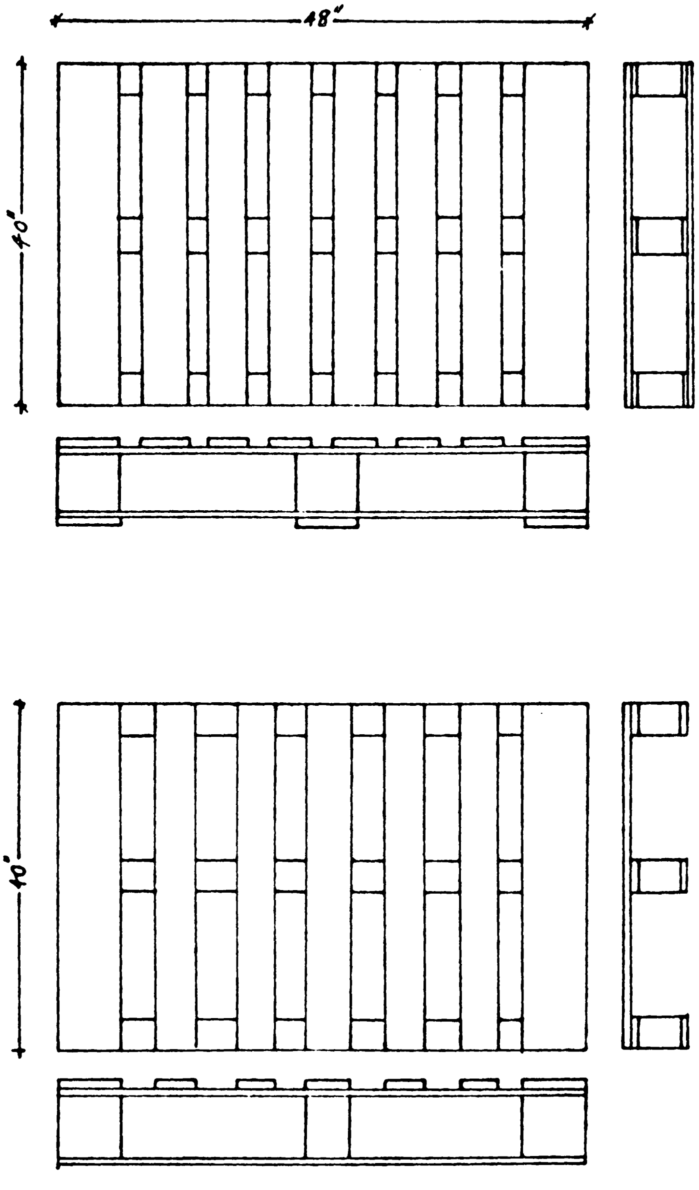
## 7.1 Introduction

The analysis models presented in chapters 4 and 5 required the use of some limitations and assumptions to realistically estimate the results of very complicated actions in a pallet. Some simplifications were made as to the mechanism of load transfer, analog idealizations of load and support conditions and the actions of individual members and joints. The net effect of these simplifications may result in errors in the prediction of load effects. To verify the models and develop confidence in their prediction an experimental program was conducted. The objective of this chapter is to document those tests and make comparisons between test results and predictions.

## 7.2 Experimental design

To verify the RAD and RAS analysis models the test deflections and average stiffnesses of the pallets were compared with the results obtained from the models. The deflection used for the verification procedure was the central deflection. Pallet stiffness was computed as the slope of the linear portion of the total load-center deflection curve for each pallet or section.

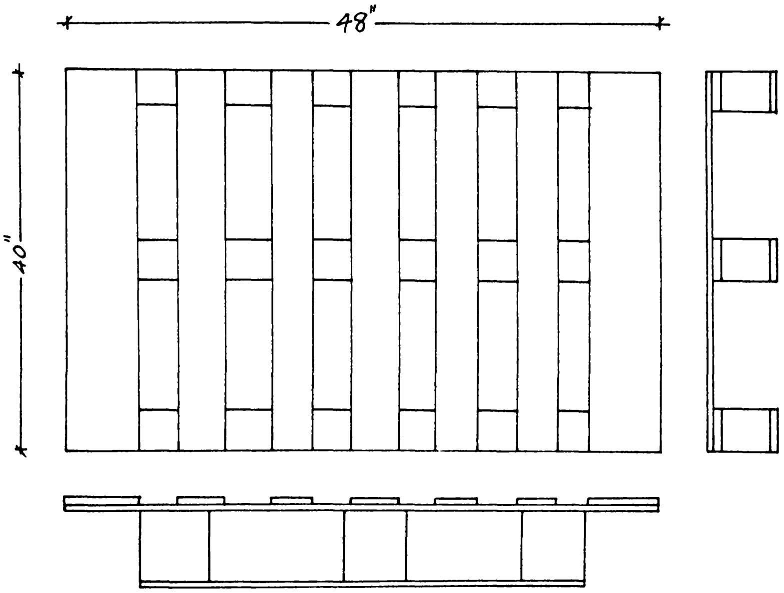
The pallet types selected for testing included five designs which are shown in Figure 7.1 and detailed in Appendix D. The study variables included number of top deck elements, number of bottom deck elements, and dimensions of all members.



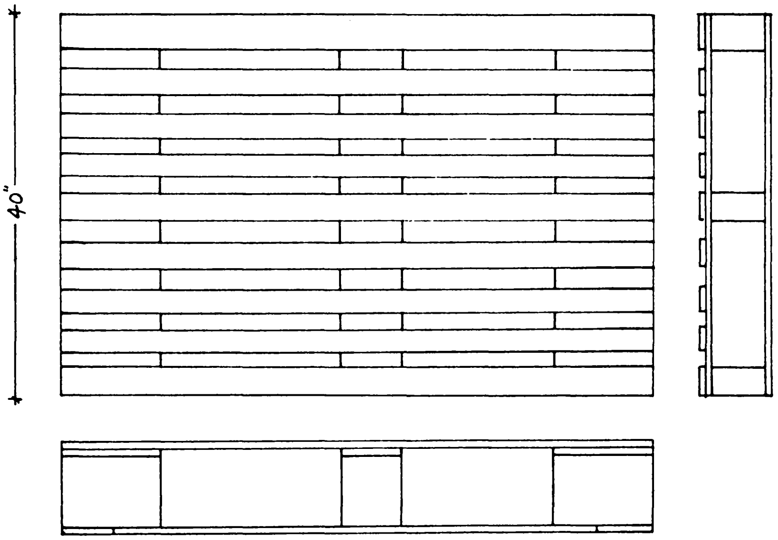
Design 1

Design 2

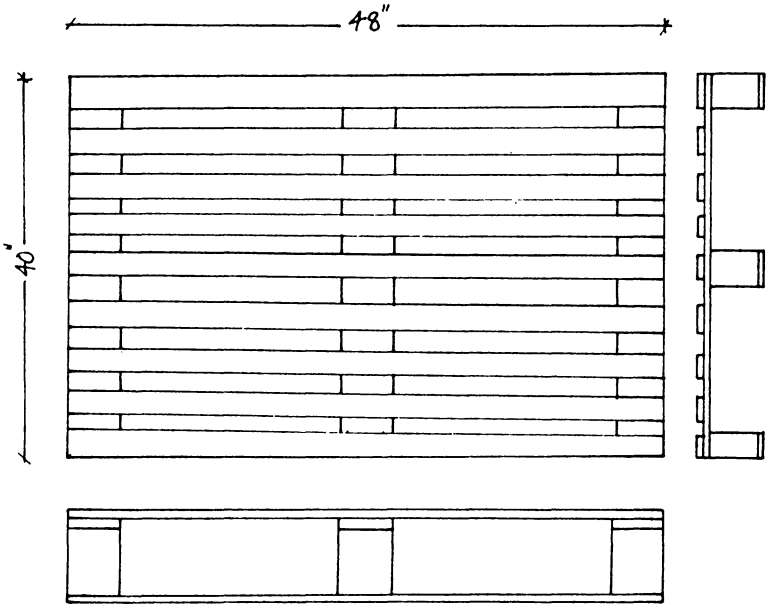
Figure 7.1 Pallet design configurations



Design 3



Design 5



Design 4

These designs were selected to be representative of the most common expected variations found in practice. The blocks in all designs were oriented horizontally (i.e length of block parallel to pallet length) so that the fastener penetrated the side grain. Each deck-block joint was connected with three fasteners, sized according to the thickness of the deckboards. The type, size and length of fasteners used in the experimental pallets are detailed in Table D.3. All pallets were constructed of yellow poplar or red oak material in the green condition (i.e M.C. > 28% ).

In the RAS mode eight pallets of two designs were tested under uniformly distributed loads at each of three different spans - 41, 43, and 45 in. A winged pallet (design 3) was tested RAS at a 33 inch span. Twelve pallets of three designs were tested in the RAD mode also under uniform loading, nine at spans of 39 and 45 in., and three at spans of 41, 43, and 45 in. The spans were varied so that the supports would cause different deckboard-block joint behavior. Six pallets were loaded to failure in RAS after testing at all study spans. All materials were kept green in a refrigerator to retard mold and blue stain. Each pallet was assembled per Figure 7.1 by hand using hammers to insert fasteners.

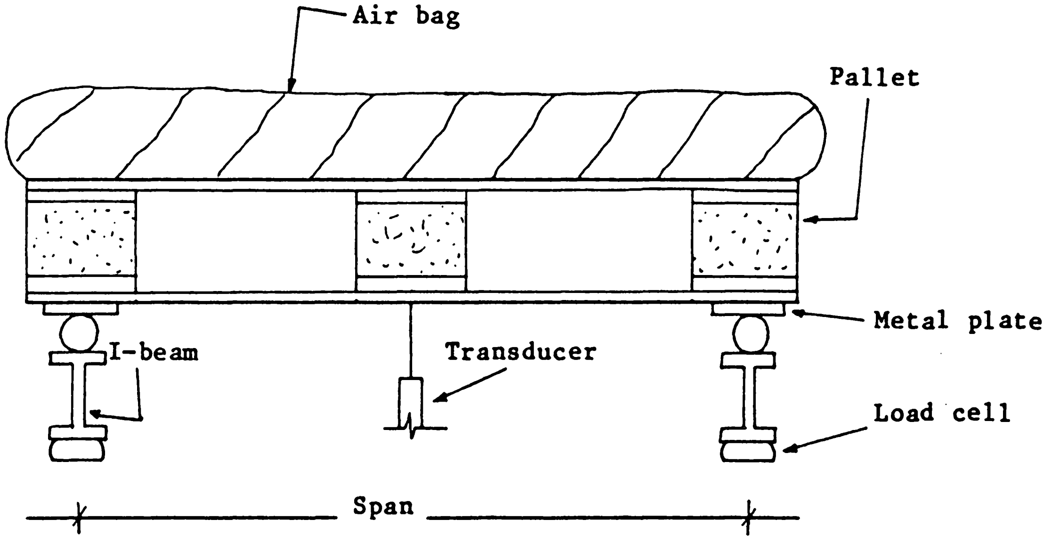
### 7.3 Methods and materials

All stringerboards and deckboards were tested in center point loading to determine the MOE (modulus of elasticity) prior to assembly into a pallet. Three measurements of width and thickness were taken along the length of each piece and the average values recorded. These results are also tabulated in Appendix D.

The test machine used to evaluate pallet response to a uniform loading condition is shown schematically in Figure 7.2 . This machine uses an inflatable dunnage bag to apply a uniform load over the top deck of the pallet. Load is measured by load cells located under the four corners of the pallet. Deflection is measured at three points along the center line of the pallet. Details of the apparatus are in Fagan(1984). During testing the bag was inflated at a metered rate and a HP9000 microcomputer automatically recorded the load and deflection measured by each transducer. The results were later analyzed on a mainframe computer. After testing at the various spans some pallets were loaded to failure and ultimate strength and failure mechanism was recorded.

## 7.4 Results and discussion

The predicted and measured deflections of all pallets tested are presented in Appendix E. The average percentage deviations between predicted and experimental stiffness for all designs are shown in Table 7.1. These results show that, on average, the models overestimated experimental stiffness in both the RAS and RAD modes for uniform loading conditions by 4.6% and 8.2%, respectively. These differences between predicted and actual results may be attributed to a number of reasons; errors in determining actual material properties for the elements, prediction of rotation modulus for stiffness verification, errors during apparatus set up and errors associated with assumptions made in the



Air bag tester - UDL

Figure 7.2. Testing Apparatus

Table 7.1. Summary of the comparison between model predictions of the stiffness of uniformly loaded test pallets and experimental results.

| | | Percent deviation between Predicted and experimental stiffness | |
|---|---|---|---|
| Support mode | Pallet Design | Average | Range |
| RAS | 1 | 5.9 | 2.9 - 10.9 |
| | 2 | 6.7 | 3.2 - 11.6 |
| | 3 | 1.1 | - |
| RAD | 1 | 7.7 | 1.7 - 16.4 |
| | 4 | 7.8 | 1.4 - 23.0 |
| | 5 | 9.1 | 1.9 - 13.0 |

analysis procedure.

Three different support locations were used in the test procedure; a) the support placed at the center of the outer block, b) the support placed to the left of the center line of the outer block, and c) the support placed to the right of the center of the outer block. Each support location had a considerable effect on joint behaviour. Putting the support to the right of the centerline of the block resulted in the largest test span and hence, the largest deflections compared to the two other spans. Here the joint opened in the manner forecast by the models. As the span was shortened the level of fixity of the joint increased and there was less separation of top and bottom deck members from the block. With RAS, the model was slightly more conservative for the longest span. There were no significant differences between the results at the two shorter spans. With RAD no consistent difference in the predictive ability of the model was noted with span.

The thickness of the deckboards and stringerboards used in pallets tested RAD (Table D.2) were varied to confirm the models capability at different levels of stiffness. Both the centerline deflection and the degree of fixity of the deck-block joint increased as the thickness of the boards decreased. The greater deflection may be attributed to the lower flexural rigidity of the deck elements. For the lower thicknesses the top deckboards tended to bend, not around the pivot point, but out from the inside edge of the outer block. The thicker boards, on the other hand, had a higher flexural stiffness and tended to separate more from the block. No difference in the predictive ability of the model was noted with stiffness variation.

Tests of six pallets RAS were successfully carried out to failure. These results are shown in Table 7.2. There is no clear way to assess the ability of the RAS model to predict ultimate strength since component modulus of rupture was not known apriori. However, assuming linear elasticity to failure the predicted bending stresses are well within the range shown for deckboards by McLain et al (1986). The failures were noted to occur at or near the predicted points of maximum stress.

Considering the high variability of the material and its properties, the models adequately predict the response to applied load conditions.

| Pallet ID No. | Span (in.) | Ultimate |  | Predicted failure Bending stress (Linear),psi | Observed failure |
|---|---|---|---|---|---|
|  |  | Load (lbs) | Deflection (in.) |  |  |
| 4 | 45 | 6038 | 2.12 | 8625 | Outer stringer |
| 5 | 43 | 7024 | 2.50 | 7547 | Outer stringer |
| 6 | 41 | 6002 | 2.04 | 6026 | Center stringer |
| 7 | 45 | 7023 | 3.68 | 7557 | Center stringer |
| 8 | 41 | 8032 | 2.72 | 8158 | Outer stringer |
| 9 | 33 | 9203 | 1.83 | 6009 | Center stringer |

Table 7.2. Ultimate test results for uniformly loaded pallets - RAS

# 8.0 SUMMARY AND CONCLUSIONS

Wooden pallet production uses as much as 20% of the total annual U.S. lumber production. However, standardized design procedures for these products have only recently been developed. These procedures are needed to efficiently balance safety and economy in pallet design and to provide a rational means of comparing design alternatives. Research to develop these procedures has been undertaken by the U.S. Forest Service, the National Wood Pallet and Container Association, and Virginia Polytechnic Institute and State University. Phase I of the research was directed towards the stringer-type pallet while the block-type pallet was considered in Phase II. The objective of this thesis is to establish standard methods to analyze block-type pallets for strength and stiffness in response to defined load and support conditions.

The analysis procedures for block pallets were developed in computerized form. The Fortran programs presented in Appendices A, B, and C are used in a version of the Pallet Design System (PDS) geared to block pallet design. Matrix structural analysis models and solution routines were developed to analyze racked across stringers (RAS), racked across deckboards (RAD), and stack support modes. The loading conditions considered include 1) Full uniform load, 2) Partial uniform load, and 3) 1, 2, or 3 line loads. A 3-Dimensional model is employed for the RAS and Stack modes, and a 2-Dimensional model is used for the RAD support condition.

Sensitivity studies were conducted to determine the influence of the magnitude of various joint properties on pallet response. Specifically for pallets RAS and RAD, the deck-block joint was examined for the influence of rotational and withdrawal stiffness on pallet response. Pallet deflection was shown to be insensitive to translational stiffness in the deck-block joint. Pallet response was also not sensitive to the level of translational or rotational stiffness of the deckboard/stringerboard joint in the RAS model. A 100% change in the deckboard/stringerboard joint translational stiffness in the global 1 and 3 directions resulted in a 0 to 2% change in center point deflection. Rotational stiffness about the global 3-axis (i.e deck-block rotation) was found to be the most important joint parameter with a 10% change in pallet center point deflection resulting from a 100% change in rotation modulus.

RAS ANALOG MODEL: The 3-Dimensional model employed in the RAS support mode uses matrix structural analysis methods to calculate stress and deflection in critical members subject to common loading conditions. The model is presented as a quarter symmetric structure of a pallet utilizing shear releases along the lines of symmetry. The number of elements used in the model dynamically changes depending on the number of top deckboards in the pallet. The joints are modeled as zero length spring elements to emulate their semi-rigid nature; all deckboard and stringerboard elements are given material properties representative of those in the full pallet.

RAD ANALOG MODEL: Experimental observations and theoretical studies showed that two-way bending of pallets in the RAD mode was negligible. Hence, a 2-Dimensional model was used to mimic the actions of a block-type pallet in the RAD support mode. The number of elements and joints were fixed at 20 and 16, respectively. The model was reduced to half symmetry employing two shear releases at center line. The top deck elements are assigned material properties equal to the sum of those for the top deck of the full pallet. A similar process is used for the bottom deck elements. Properties of members not used for a specific geometry are given zero stiffness. A unique model of joint behavior was developed to imitate the actions of the very important deck-block joints.

STACK MODELS: The RAS and RAD models are used with some modifications for the stack condition. The critical stringerboard elements are identified using the RAS model; the RAD model is used to determine the critical load effects in the deckboard elements. Only a full uniform load is considered in the stack condition with a minimum of four top deckboards. Modifications are made to the joints by applying constraints in all directions to those joints connecting block elements.

To verify the RAD and RAS models a series of tests were carried out on a total of twenty pallets of five designs. The pallets were loaded by an airbag and deflection noted. For verification the test deflections and average stiffnesses of the pallets were compared with those predicted.

The average percent deviation in stiffness between predicted and experimental for the RAS support mode was 4.6% over a range from 1.1 - 11.6% deviation. For the RAD mode the average percent deviation was 8.2% over a range from 1.7 - 23.0% deviation. Tests of six pallets RAS were successfully carried out to failure. The average predicted failure stress in bending was 7320 psi which is in the range found by McLain et al (1986).

In conclusion, the three models used with the matrix structural analysis techniques adequately predict the response of block pallets subject to the study load and support conditions. These analysis procedures, used in conjunction with Reliability-Based safety checking, ensure that block-type pallets will meet the desired performance requirements with an assured level of safety and economy.

# Bibliography:

Bodig & Jayne, Mechanics of Wood and Wood Composites, (1982). Van Nostrand Reinhold Company, New York, N.Y.

Goehring, C. B., Wallin, W. B., (1981), A Survey of loads, Loading Conditions, and Handling Environment Conditions for Wooden Pallets. Northeastern Forest Experiment Station, Forestry Sciences Laboratory, Princeton, West Virginia.

Holzer, S. M. (1985), Computer Analysis of Structures: Matrix Structural Analysis - Structured Programming. Elsevier Science Publishing Co., Inc. N.Y.

Kyokong, B. (1979), The development of a Model of the Mechanical behavior of Wooden Pallets. Ph D Dissertation, Virginia Polytechnic Institute & State University (VPI & SU), Blacksburg, Va.

Loferski, J. R. (1985), A Reliability based Design Procedure for Wooden Pallets. Ph D Dissertation, Virginia Polytechnic Institute & State University (VPI & SU), Blacksburg, Va.

Mack, J. J. (1985), Contribution of behavior of Deckboard-Stringer joints to Pallet Performance - - a pilot study. Wood Research and Wood

Construction Laboratory Bulletin No.136, VPI & SU, Blacksburg, Va. (May).

Mulheren, K. C. (1982), "SPACEPAL" - - The Development and Verification of a Space Frame Analysis Model for Wood Pallets. Department of Forest Products, VPI & SU, Blacksburg, Va. (March).

Stern, G. E. (1986), Wooden Pallets - Indespensible for Nort American Industries. New Zealand Journal of Timber Construction, Volume 2, No.2. (June).

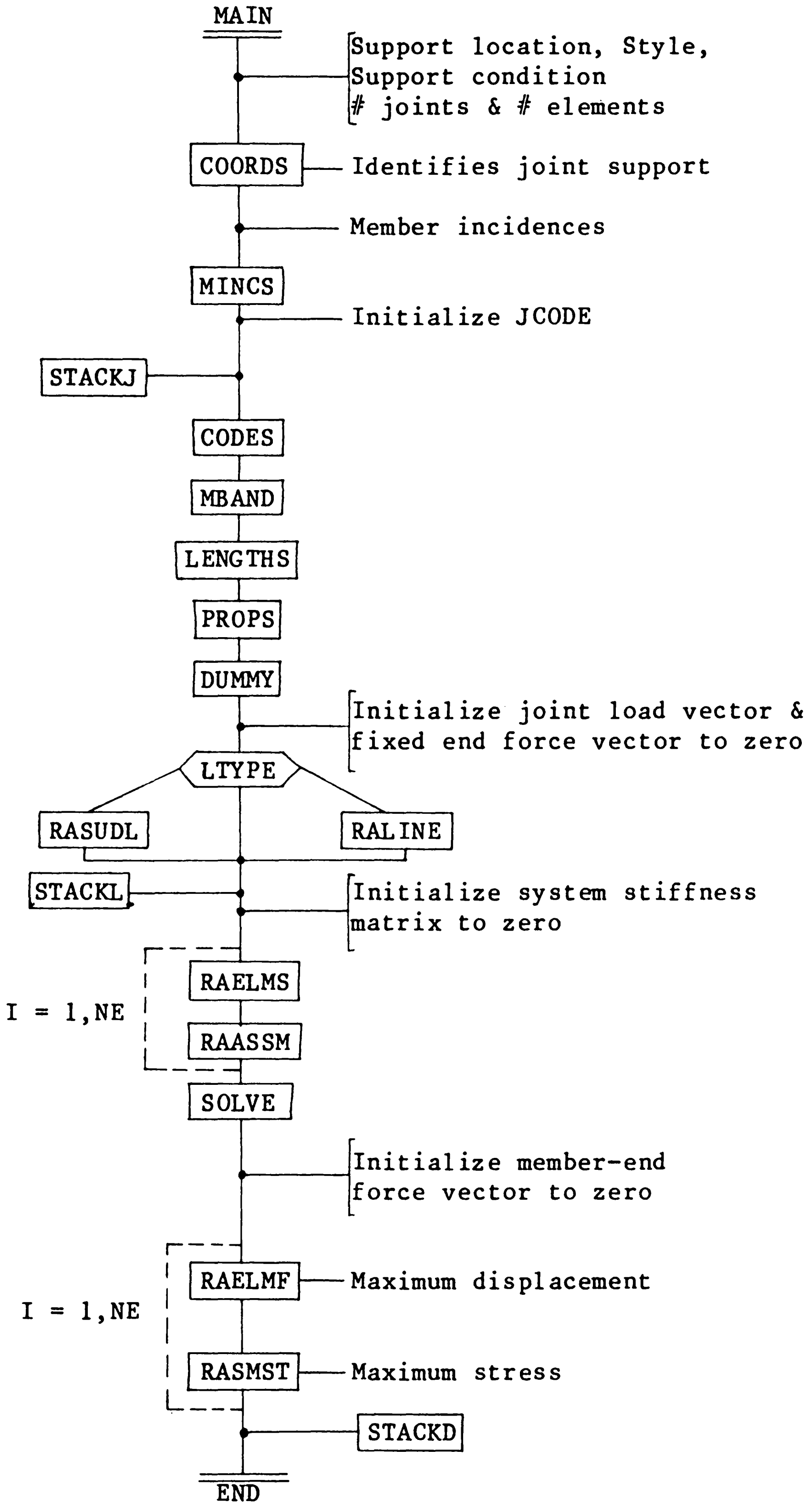
Stern, G. E. (1987), Pallets serving as a Barometer of the U.S. Economy. ASME Codes and Standards, (July).

Timber Construction Manual, (1974), American Institute of Timber Construction, John Wiley and Sons, New York, N.Y.

Wilkinson, T. L. (1985), Rotational Characteristics of Pallet Joints. Forest Service. Research paper FPL 457, (August).

# Appendix A:

Program listing - (RAS)


MAIN
Support location, Style,
Support condition
# joints & # elements
COORDS
Identifies joint support
Member incidences
MINCS
Initialize JCODE
STACKJ
CODES
MBAND
LENGTHS
PROPS
DUMMY
Initialize joint load vector &
fixed end force vector to zero
LTYPE
RASUDL
RALINE
STACKL
Initialize system stiffness
matrix to zero
I = 1,NE
RAELMS
RAASSM
SOLVE
Initialize member-end
force vector to zero
I = 1,NE
RAELMF
Maximum displacement
RASMST
Maximum stress
STACKD
END

```
*****************************************************************
*  RAS ANALYSIS......                                           *
*****************************************************************
C
C THIS VERSION OF RAS CODING ALLOWS FOR ONLY ONE VALUE FOR MOE TO BE
C ASSIGNED TO BOTH DECKBOARDS AND STRINGERBOARDS ON TOP AND BOTTOM
C DECKS.
C
   IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
   DOUBLE PRECISION BLKLOC(2,3),TDLOC(15),TDW(15),BLKW(2),XLC(100),
  $        YLC(100),ZLC(100),EL(100),TSBW(2),BSBW(2),
  $        BDW(15),XMOE(100),W(100),T(100),XMG(100),EI(3,100),
  $        S(3,100),XMJ(100),Q(600),F(12,100),SS(600,600),G(27),
  $        A(100),P(6,100),D(12),DL(12),FL(12,100),FLO(12),
  $        TSBE,BSBE,BLKL(2),BDLOC(15),DISPL(6,100),
  $        TDBE,BDBE,ST(100),STA(100),DSTB(100),
  $        STB(100),SSMAX,XPOS,WL(100),DST(100),DSMAX,DSTA(100),
  $        RASPAN,XPLENG,XPWIDT,BLKT,TDT,BDT,TSBT,BSBT,CL,SL,XTD,
  $        XTSB,SUPLOC,CANTL,BAREA,BMOE,XLOAD,P1,P2,DEFL(100),
  $        GAM3,GAM4,GAM5,GAM6,YMAX
C
   INTEGER JCODE(6,150),MINC(2,92),MCODE(12,100),JN(13),MN(13),
  $      IFLAG(100),ISL(100)
C
   PARAMETER (MX = 94, MXNEQ = 6 * (MX - 1), BMOE = 10.0D+06,
  $        BAREA = 500.0D+0)
   DATA NTD,NBD,NTSB,NBSB,NBLK/5,0,3,3,9/
   DATA RASPAN,XPLENG,XPWIDT/33.0D+0,48.0D+0,40.0D+0/
   DATA BLKT,NBLKPS,BLKW,BLKL/3.625D0,3,2*3.125D+0,2*5.0D+0/
   DATA BLKLOC(1,1),BLKLOC(1,3)/5.0D+0,38.0D+0/
   DATA TDT,BDT,TSBT,BSBT/4*0.80D+0/
   DATA TDBE,BDBE/1.20D06,1.20D06/
   DATA TSBE,BSBE/1.00D06,0.91D06/
   DATA TSBW,BSBW/3.136D00,3.149D00,3.151D00,3.155D00/
   DATA LTYPE,XLOAD,CL,SL,XTD,XTSB/1,9.2030D03,00.D+0,000.0D+0,0.0D+0
  $      ,00.0D+0/
   DATA (TDLOC(J),J=1,7)/0.0D+0,1.0D00,2.0D0,21.76D00,29.013D00,
  $     39.00D00,43.47D00/
C   DATA (BDLOC(J),J=1,3)/0.0D0,21.5D0,43.0D0/
   DATA ISTYLE/2/
   DATA ISTACK/0/
   DATA (TDW(J),J=1,7)/4.539D00,4.529D00,4.533D00,4.522D00,4.506D00,
  $              4.510D00,4.526/
C   DATA (BDW(J),J=1,3)/3*5.0D+0/
   DATA GAM3/48.0D+04/
   DATA GAM4,GAM5,GAM6/10.000D+06,10.000D-06,10.000D+06/
C
   DATA JN/2,7,7,13,13,19,19,25,25,31,31,37,37/
   DATA MN/1,10,10,18,18,26,26,34,34,42,42,50,50/
   IF (ISTACK.EQ.1) THEN
    SUPLOC =  BLKLOC(1,NBLKPS) - 0.5*(XPLENG) + 1.0D00
    WRITE(6,*) 'SUPLOC =',SUPLOC
   ELSE
    SUPLOC = RASPAN/2.0D0
    WRITE(6,*) 'SUPLOC =',SUPLOC
   END IF
   IF (ISTYLE.EQ.2.OR.ISTYLE.EQ.3) THEN
    ICANT = 1
    CANTL = BLKLOC(1,1)
   ELSE
    ICANT = 0
    CANTL = 0.0D00
   END IF
   WRITE(6,*) 'ICANT =',ICANT,' CANTL =',CANTL
C
C DETERMINING THE SUPPORT CONDITION:
C
   IF (RASPAN/2.LT.(BLKLOC(1,NBLKPS) - 0.5*(XPLENG))) THEN
    ISUPC = 4
   ELSE IF (RASPAN/2.GT.(BLKLOC(1,NBLKPS)-.5*XPLENG+.5*BLKL(1)))THEN
    ISUPC = 3
   ELSE IF (RASPAN/2.EQ.(BLKLOC(1,NBLKPS)-.5*XPLENG+.5*BLKL(1)))THEN
    ISUPC = 2
   ELSE
    ISUPC = 1
   END IF
   WRITE(6,*) 'ISUPC =',ISUPC
C
C DETERMINING THE NO. OF ELEMENTS AND JOINTS:
C
   IF (JN(NTD).EQ.2) THEN
    NJ = JN(NTD) + 29 + (5*ICANT)
   ELSE
    NJ = JN(NTD) + 27 + (5*ICANT)
   END IF
   IF (MN(NTD).EQ.1) THEN
```

```
      NE = MN(NTD) + 7 + 36 + (6*ICANT)
     ELSE
      NE = MN(NTD) + 36 + (6*ICANT)
     END IF
     WRITE(6,*) 'NE =',NE, ' NJ =',NJ
C
     CALL COORDS(NTD,TDLOC,XPLENG,TDW,BLKLOC,BLKL,ICANT,CANTL,JN,
    $ NBLKPS,ISUPC,NJ,SUPLOC,BLKT,BLKW,XPWIDT,XLC,MTTD,JNTSUP,YLC,ZLC)
     WRITE(6,*) 'JNTSUP  =',JNTSUP
     WRITE(6,*) 'MTTD =',MTTD
C
C   READ IN THE MEMBER INCIDENCES:
C
C  THESE ARE THE MEMBER INCIDENCES OF ALL MEMBERS TO THE LEFT OF THE
C  OUTER TWO BLOCKS, INCLUDING THE TOP D.B OVER THE BLOCK ELEMENTS.
C
     DATA MINC/1,2,1,3,2,4,3,4,1,6,2,7,3,8,4,10,5,6,5,7,8,9,9,10,
    $          6,12,7,13,8,14,10,16,11,12,11,13,14,15,15,16,12,18,13,
    $          19,14,20,16,22,17,18,17,19,20,21,21,22,18,24,19,25,
    $          20,26,22,28,23,24,23,25,26,27,27,28,24,30,25,31,26,32,
    $          28,34,29,30,29,31,32,33,33,34,30,49,31,38,32,61,34,62,
    $          35,36,35,37,37,38,36,49,36,53,37,42,42,45,53,57,49,50,
    $          53,54,57,58,50,54,54,58,50,51,58,59,51,59,51,52,52,63,
    $          63,55,55,60,59,60,55,56,56,44,38,39,42,43,45,46,39,43,
    $          43,46,39,40,46,47,40,47,40,41,41,64,44,64,44,48,47,48,
    $          41,62,52,61,45,67,57,66,65,66,65,67,67,68,66,69/
C
     CALL MINCS(ICANT,JN,MN,NJ,NE,MINC,NTD,ISUPC,XLC)
C
C   INITIALISE JCODE:
C
     DO 20 J = 1,(JN(NTD) + 27)
      DO 10 L = 1,6
       JCODE(L,J) = 1
  10  CONTINUE
  20 CONTINUE
C
     JCODE(1,1) = 0
     JCODE(5,1) = 0
     JCODE(6,1) = 0
     JCODE(1,2) = 0
     JCODE(4,2) = 0
     JCODE(5,2) = 0
     JCODE(6,2) = 0
     JCODE(1,3) = 0
     JCODE(5,3) = 0
     JCODE(6,3) = 0
     JCODE(1,4) = 0
     JCODE(4,4) = 0
     JCODE(5,4) = 0
     JCODE(6,4) = 0
     DO 30 J = 7,JN(NTD),3
      JCODE(3,J) = 0
      JCODE(4,J) = 0
      JCODE(5,J) = 0
  30 CONTINUE
     JCODE(3,(JN(NTD)+7)) = 0
     JCODE(4,(JN(NTD)+7)) = 0
     JCODE(5,(JN(NTD)+7)) = 0
     IF (ICANT.EQ.1) THEN
      DO 50 J = NJ,NJ-4,-1
       DO 40 L = 1,6
        JCODE(L,J) = 1
  40   CONTINUE
  50  CONTINUE
     END IF
C
     IF (ICANT.EQ.1) THEN
      JCODE(3,NJ-2) = 0
      JCODE(4,NJ-2) = 0
      JCODE(5,NJ-2) = 0
     END IF
C
     IF (NTSB.EQ.2) THEN
      JCODE(1,JNTSUP-1) = 0
      JCODE(2,JNTSUP-1) = 0
      JCODE(3,JNTSUP-1) = 0
      JCODE(4,JNTSUP-1) = 0
      JCODE(5,JNTSUP-1) = 0
     ELSE
```

```
      JCODE(1,JNTSUP) = 0
      JCODE(2,JNTSUP) = 0
      JCODE(3,JNTSUP) = 0
      JCODE(4,JNTSUP) = 0
      JCODE(5,JNTSUP) = 0
      JCODE(1,JNTSUP-1) = 0
      JCODE(2,JNTSUP-1) = 0
      JCODE(3,JNTSUP-1) = 0
      JCODE(4,JNTSUP-1) = 0
      JCODE(5,JNTSUP-1) = 0
     END IF
C
C  IF ISTACK IS '1' (I.E STACK SUPPORT MODE) THEN STACKJ IS CALLED
C
     DO 55 I = 1,NE
      ISL(I) = 0
   55 CONTINUE
     IF (ISTACK.EQ.1) THEN
      CALL STACKJ(JN,NTD,JCODE,NBLK,NBLKPS)
     END IF
C
     CALL CODES(JCODE,MINC,NE,NJ,MCODE,NEQ)
     MBD = MBAND(MCODE,NE)
     CALL LENGTS(MINC,ZLC,YLC,XLC,NTD,ICANT,NE,IFLAG,EL,MN)
     CALL PROPS(NTD,TSBE,TSBW,TSBT,BSBE,BSBW,BSBT,TDBE,TDW,MN,TDT,BDBE,
    $       BDW,BDT,BMOE,XMOE,W,T,XMG,EI,S,A,XMI,FL,ICANT,MITD,BAREA,
    $  NE,BLKT,IFLAG,BLKL,BLKW,BDLOC,XLC,MINC,NBD,NTSB,NBSB,XPLENG,ISL)
     CALL DUMMY(NBLK,NTSB,NTD,ICANT,NE,XMOE,NBSB,MN,NBD,BDLOC,EL
    $          ,MINC,XPLENG,XLC)
C
C  CALCULATION OF MODULUS OF RIGIDITY FOR ALL REAL ELEMENTS:
C
     DO 60 I = 1,NE
      IF (IFLAG(I).NE.5) THEN
       XMG(I) = XMOE(I)/16.0D0
      END IF
   60 CONTINUE
     WRITE(6,600) MEMBER','LENGTH','MOE','AREA'
  600 FORMAT(/,2X,A6,3X,A6,6X,A3,10X,A4/)
     DO 61 I = 1,NE
      WRITE(6,610) I,EL(I),XMOE(I),A(I)
  610  FORMAT(4X,I2,4X,F7.3,2X,F12.2,2X,F8.3)
   61 CONTINUE
C
C   COMPUTE THE JOINT LOAD VECTOR:
C
     DO 70 K = 1,NEQ
      Q(K) = 0.0D+0
   70 CONTINUE
     DO 90 I = 1,NE
      DO 80 L = 1,12
       F(L,I) = 0.0D+0
   80  CONTINUE
   90 CONTINUE
     DO 100 I = 1,NE
      IF (IFLAG(I).EQ.2) THEN
       WL(I) = 0.0D00
      END IF
  100 CONTINUE
     IF (LTYPE.EQ.1.OR.LTYPE.EQ.2) THEN
      CALL RASUDL(EL,NJ,NTD,XTSB,ICANT,CANTL,XPLENG,F,Q,NE,NEQ,NTSB,
    $    MN,TDLOC,XLOAD,XTD,XPWIDT,MITD,W,TDW,BLKW,BLKLOC,BLKL,WL,
    $    ISTACK,NBLK,NBLKPS)
     ELSE
      CALL RALINE(LTYPE,CL,SL,XPLENG,XTSB,NTD,MITD,TDLOC,TDW,EL,P1
    $           ,P2,Q,F)
     END IF
C
C  FOR STACK SUPPORT CONDITION - CALL SUBROUTINE 'STACKL'
C
     IF (ISTACK.EQ.1) THEN
      CALL STACKL(ISL,W,EL,XLOAD,WL,MN,F,Q,XTD,XPWIDT,BLKW,NE,
    $          NBLK,NBLKPS,NTD)
     END IF
     WRITE(6,*) 'QHAT VECTOR...'
     DO 105 I = 1,NEQ
      WRITE(6,*) I,Q(I)
  105 CONTINUE
C
```

```
C   INITIALISE THE SYSTEM STIFFNESS MATRIX TO ZERO:
C
      DO 120 J = 1,NEQ
        DO 110 I = 1,MBD+1
          SS(I,J) = 0.0D+0
  110   CONTINUE
  120 CONTINUE
      DO 130 N = 1,NE
        CALL RAELMS(N,NTD,GAM4,GAM6,G,IFLAG,ISUPC,XMOE,EL,
     $            A,XMG,XMJ,EI,MN,XLC,JN,GAM3,GAM5,ICANT,NE)
        CALL RAASSM(N,MBD,G,SS,MCODE,MXNEQ)
  130 CONTINUE
      CALL SOLVE(SS,Q,NEQ,MBD,MXNEQ)
C
C   INITIALISE THE MEMBER-END FORCE VECTOR TO ZERO:
C
      DO 150 I = 1,NE
        DO 140 L = 1,12
          FL(L,I) = 0.0D+00
  140   CONTINUE
  150 CONTINUE
C
      YMAX = 0.0D0
      SSMAX = 0.0D0
      DSMAX = 0.0D0
      WRITE(6,250) 'STRESSES:'
  250 FORMAT(/,15X,A9,/)
      WRITE(6,260) 'MEMBER #','STRESS A-END','STRESS B-END'
  260 FORMAT(/,9X,A6,5X,A12,12X,A12,/)
      DO 160 I = 1,NE
        IF (EL(I).NE.0) THEN
          CALL RAELMF(MCODE,Q,IFLAG,ISUPC,XMOE,I,EI,EL,A,XMG,XMJ,D,DL
     $              ,FLO,F,FL,MN,NTD,YMAX,MINC,NE,DEFL,DISPL,JT)
          CALL RASMST(IFLAG,A,S,LTYPE,XPLENG,XTSB,MTTD,NTD,TDLOC,TDW,
     $        MN,FL,ST,STA,STB,P2,EL,FLO,I,SSMAX,DSMAX,NE,XTD,WL,
     $        DST,DSTA,DSTB)
        END IF
  160 CONTINUE
C
      WRITE(6,300) 'JOINT DISPLACEMENTS:'
  300 FORMAT(//15X,A21/)
      WRITE(6,310) 'JT #','1-DIR','2-DIR','3-DIR','4-DIR','5-DIR',
     $         '6-DIR'
  310 FORMAT(A4,4X,A5,5(7X,A5)/)
      DO 200 J = 1,NJ
        WRITE(6,320) J,DISPL(1,J),DISPL(2,J),DISPL(3,J),DISPL(4,J),
     $           DISPL(5,J),DISPL(6,J)
  320   FORMAT(1X,I2,2X,F10.8,5(2X,F10.8))
  200 CONTINUE
C
        WRITE(6,*)
        WRITE(6,*) 'MAX.VERTICAL DEFL. =',YMAX,' AT JT.#',JT
C
      IF (ISTACK.EQ.1) THEN
        CALL STACKD(JN,NTD,XLC,DEFL,XMOE)
      END IF
C
      STOP
      END
C
C
C
C
****************************************************************
*                   COORDS                          *
****************************************************************
      SUBROUTINE COORDS(NTD,TDLOC,XPLENG,TDW,BLKLOC,BLKL,ICANT,CANTL,JN,
     $ NBLKPS,ISUPC,NJ,SUPLOC,BLKT,BLKW,XPWIDT,XLC,MTTD,JNTSUP,YLC,ZLC)
      DOUBLE PRECISION TDW(*),XLC(*),YLC(*),ZLC(*),TDLOC(15),BLKL(2),
     $       BLKLOC(2,3),BLKW(2),XPLENG,CANTL,SUPLOC,BLKT,XPWIDT
      INTEGER JN(13)
C
C  X - COORDINATES OF ALL POINTS:
C
      DO 10 I = 1,4
        XLC(I) = 0.0D+0
   10 CONTINUE
      IF (MOD(NTD,2).EQ.0) THEN
        MTTD = NTD/2 + 1
      ELSE
```

```
      MTTD = INT((NTD/2) + 1.5)
    END IF
    M = 4
    DO 30 J = (MTTD-1),NTD
      DO 20 I = 1,6
        IF (MOD(NTD,2).EQ.0) THEN
          IF (ICANT.EQ.1) THEN
            K = J + 2
          ELSE
            K = J + 1
          END IF
        ELSE
          IF (ICANT.EQ.1) THEN
            K = J + 3
          ELSE
            K = J + 2
          END IF
        END IF
        XLC(M + I) = TDLOC(K) - (XPLENG/2) + (.5*TDW(K))
        IF ((M + I).GE.JN(NTD)) THEN
          GO TO 35
        END IF
 20   CONTINUE
      M = M + 6
 30 CONTINUE
 35 M = JN(NTD)
    DO 60 X = 1,2
      DO 50 J = 0,2
        DO 40 I = 1,4
          XLC(M + I) = BLKLOC(1,NBLKPS) + (J*.5*BLKL(1)) - (0.5*XPLENG)
          IF (J.EQ.1.AND.I.EQ.3.AND.X.EQ.1) THEN
            M = M + 3
            GO TO 50
          END IF
 40     CONTINUE
        M = M + 4
 50   CONTINUE
      M = JN(NTD) + 11
 60 CONTINUE
C
C  NOTE:- 4 SUPPORT CONDITIONS(ISUPC): 1 -SUPPORT UNDER BLOCK TO LEFT OF
C        D.B SPRING. 2 - DIRECTLY UNDER D.B SPRING. 3 - TO RIGHT OF
C        D.B SPRING. 4 - INSIDE THE OUTER BLOCK(IE. UNDER ST.BOARD)
C
    IF (ICANT.EQ.1) THEN
      XLC(NJ) = BLKLOC(1,NBLKPS) + BLKL(1) + CANTL - (.5*XPLENG)
      XLC(NJ-1) = XLC(NJ)
      XLC(NJ-2) = TDLOC(NTD + 2) - XPLENG/2 + .5*TDW(NTD + 2)
      XLC(NJ-3) = XLC(NJ-2)
      XLC(NJ-4) = XLC(NJ-2)
      XLC(JN(NTD) + 34) = BLKLOC(1,NBLKPS) - 0.5D0*XPLENG - 1.0D0
      XLC(JN(NTD) + 35) = BLKLOC(1,NBLKPS) - 0.5D0*XPLENG - 1.0D0
      IF (ISUPC.EQ.4) THEN
        XLC(NJ-7) = SUPLOC
        XLC(NJ-8) = SUPLOC
        JNTSUP = (NJ-7)
        XLC(NJ-5) = BLKLOC(1,NBLKPS) + 1.0D0 - (.5D0*XPLENG)
        XLC(NJ-6) = BLKLOC(1,NBLKPS) + 1.0D0 - (.5D0*XPLENG)
      ELSE
        XLC(NJ-5) = SUPLOC
        XLC(NJ-6) = SUPLOC
        JNTSUP = (NJ-5)
        XLC(NJ-7) = BLKLOC(1,NBLKPS) - 1.0D0 - (.5D0*XPLENG)
        XLC(NJ-8) = BLKLOC(1,NBLKPS) - 1.0D0 - (.5D0*XPLENG)
      END IF
    ELSE IF (ISUPC.EQ.4) THEN
      XLC(NJ) = BLKLOC(1,NBLKPS) + 1.0D0 - (.5D0*XPLENG)
      XLC(NJ-1) = BLKLOC(1,NBLKPS) + 1.0D0 - (.5D0*XPLENG)
      XLC(NJ-2) = SUPLOC
      XLC(NJ-3) = SUPLOC
      JNTSUP = (NJ-2)
    ELSE
      XLC(NJ) = SUPLOC
      XLC(NJ-1) = SUPLOC
      JNTSUP = NJ
      XLC(NJ-2) = BLKLOC(1,NBLKPS) - 1.0D0 - (.5D0*XPLENG)
      XLC(NJ-3) = BLKLOC(1,NBLKPS) - 1.0D0 - (.5D0*XPLENG)
    END IF
C
```

```
C  Y - COORDINATES OF ALL POINTS:
C
      DO 70 I = 1,4
        IF (I.GT.2) THEN
          YLC(I) = 0.0D+0
        ELSE
          YLC(I) = BLKT
        END IF
  70  CONTINUE
      M = 4
  75  DO 80 I = 1,6
        IF (I.LE.3) THEN
          YLC(M+I) = BLKT
          IF ((M+I).GE.JN(NTD)) THEN
            GO TO 85
          END IF
        ELSE
          YLC(M+I) = 0.0D+0
        END IF
  80  CONTINUE
      M = M + 6
      GO TO 75
  85  CONTINUE
      M = JN(NTD)
      DO 110 N = 1,2
        DO 100 J = 1,3
          DO 90 I = 1,4
            IF (I.LE.2) THEN
              YLC(M+I) = BLKT
            ELSE IF (I.EQ.3.AND.J.EQ.2.AND.N.EQ.1) THEN
              YLC(M+I) = 0.0D+0
              M = M + 3
              GO TO 100
            ELSE
              YLC(M+I) = 0.0D+0
            END IF
  90      CONTINUE
          M = M + 4
 100    CONTINUE
        M = JN(NTD) + 11
 110  CONTINUE
      IF (ICANT.EQ.1) THEN
        YLC(NJ) = BLKT
        YLC(NJ-1) = BLKT
        YLC(NJ-2) = BLKT
        YLC(NJ-3) = BLKT
        YLC(NJ-4) = BLKT
        YLC(NJ-5) = 0.0D+0
        YLC(NJ-6) = 0.0D+0
        YLC(NJ-7) = 0.0D+0
        YLC(NJ-8) = 0.0D+0
      ELSE
        YLC(NJ) = 0.0D+0
        YLC(NJ-1) = 0.0D+0
        YLC(NJ-2) = 0.0D+0
        YLC(NJ-3) = 0.0D+0
      END IF
C
C  Z - COORDINATES OF ALL POINTS:
C
      ZLC(1) = 0.0D+0
      ZLC(2) = .5D0*XPWIDT - .5D0*BLKW(1)
      ZLC(3) = 0.0D+0
      ZLC(4) = .5D0*XPWIDT - .5D0*BLKW(1)
      M = 4
      DO 130 I = 1,2*(MTTD-1)
        DO 120 J = 1,3
          IF (J.EQ.3) THEN
            ZLC(M+J) = .5D0*XPWIDT - .5D0*BLKW(1)
            IF ((M+J).EQ.JN(NTD)) THEN
              GO TO 135
            END IF
          ELSE
            ZLC(M+J) = 0.0D+0
          END IF
 120    CONTINUE
        M = M + 3
 130  CONTINUE
 135  M = JN(NTD)
      DO 150 J = 1,2
```

```
      DO 140 I = 1,12
        IF (J.EQ.1) THEN
          ZLC(M + I) = .5D0*XPWIDT - .5D0*BLKW(1)
          IF (I.EQ.11) THEN
            GO TO 145
          END IF
        ELSE
          ZLC(M + I) = 0.0D + 0
        END IF
 140  CONTINUE
 145  M = M + 11
 150 CONTINUE
C  IF THERE ARE ICANTILEVERED STRINGERBOARDS:
     IF (ICANT.EQ.1) THEN
       ZLC(NJ) = 0.0D + 0
       ZLC(NJ-1) = .5D0*XPWIDT - .5D0*BLKW(1)
       ZLC(NJ-2) = .5D0*XPWIDT - .5D0*BLKW(1)
       ZLC(NJ-3) = 0.0D + 0
       ZLC(NJ-4) = 0.0D + 0
       ZLC(NJ-5) = .5D0*XPWIDT - .5D0*BLKW(1)
       ZLC(NJ-6) = 0.0D + 0
       ZLC(NJ-7) = .5D0*XPWIDT - .5D0*BLKW(1)
       ZLC(NJ-8) = 0.0D + 0
     ELSE
       ZLC(NJ) = .5D0*XPWIDT - .5D0*BLKW(1)
       ZLC(NJ-1) = 0.0D + 0
       ZLC(NJ-2) = .5D0*XPWIDT - .5D0*BLKW(1)
       ZLC(NJ-3) = 0.0D + 0
     END IF
     RETURN
     END
*****************************************************************
*                          MINCS                                *
*****************************************************************
     SUBROUTINE MINCS(ICANT,JN,MN,NJ,NE,MINC,NTD,ISUPC,XLC)
     INTEGER MINC(2,*),JN(13),MN(13)
     DOUBLE PRECISION XLC(*)
C
C   THIS SUBROUTINE ASSIGNS THE MEMBER INCIDENCES TO THE OUTER BLOCK
C   ELEMENTS. THE OUTER BLOCK ELEMENTS ARE NUMBERED STARTING FROM THE
C   VALUE MN(NTD), WHERE NTD = # OF TOP D.B'S. THE JOINTS FOR THESE
C   ELEMENTS ARE NUMBERED STARTING FROM THE VALUE JN(NTD).
C
     M = MN(NTD)
     J = JN(NTD)
C
C
     MINC(2,M-2) = J + 25
     MINC(2,M-3) = J + 24
     MINC(2,M-4) = J + 1
     MINC(2,M-5) = J + 12
     MINC(1,M + 1) = J
     MINC(2,M + 1) = J + 1
     MINC(1,M + 2) = J - 1
     MINC(2,M + 2) = J + 12
     MINC(1,M + 3) = J - 1
     MINC(2,M + 3) = J + 16
     MINC(1,M + 4) = J
     MINC(2,M + 4) = J + 5
     MINC(1,M + 5) = J + 5
     MINC(2,M + 5) = J + 8
     MINC(1,M + 6) = J + 16
     MINC(2,M + 6) = J + 20
     MINC(1,M + 7) = J + 12
     MINC(2,M + 7) = J + 13
     MINC(1,M + 8) = J + 16
     MINC(2,M + 8) = J + 17
     MINC(1,M + 9) = J + 20
     MINC(2,M + 9) = J + 21
     MINC(1,M + 10) = J + 13
     MINC(2,M + 10) = J + 17
     MINC(1,M + 11) = J + 17
     MINC(2,M + 11) = J + 21
     MINC(1,M + 12) = J + 13
     MINC(2,M + 12) = J + 14
     MINC(1,M + 13) = J + 21
     MINC(2,M + 13) = J + 22
     MINC(1,M + 14) = J + 14
     MINC(2,M + 14) = J + 22
     MINC(1,M + 15) = J + 14
```

```
      MINC(2,M + 15) = J + 15
      MINC(1,M + 16) = J + 15
      MINC(2,M + 16) = J + 26
      MINC(1,M + 17) = J + 18
      MINC(2,M + 17) = J + 26
      MINC(1,M + 18) = J + 18
      MINC(2,M + 18) = J + 23
      MINC(1,M + 19) = J + 22
      MINC(2,M + 19) = J + 23
      MINC(1,M + 20) = J + 18
      MINC(2,M + 20) = J + 19
      MINC(1,M + 21) = J + 7
      MINC(2,M + 21) = J + 19
      MINC(1,M + 22) = J + 1
      MINC(2,M + 22) = J + 2
      MINC(1,M + 23) = J + 5
      MINC(2,M + 23) = J + 6
      MINC(1,M + 24) = J + 8
      MINC(2,M + 24) = J + 9
      MINC(1,M + 25) = J + 2
      MINC(2,M + 25) = J + 6
      MINC(1,M + 26) = J + 6
      MINC(2,M + 26) = J + 9
      MINC(1,M + 27) = J + 2
      MINC(2,M + 27) = J + 3
      MINC(1,M + 28) = J + 9
      MINC(2,M + 28) = J + 10
      MINC(1,M + 29) = J + 3
      MINC(2,M + 29) = J + 10
      MINC(1,M + 30) = J + 3
      MINC(2,M + 30) = J + 4
      MINC(1,M + 31) = J + 4
      MINC(2,M + 31) = J + 27
      MINC(1,M + 32) = J + 7
      MINC(2,M + 32) = J + 27
      MINC(1,M + 33) = J + 7
      MINC(2,M + 33) = J + 11
      MINC(1,M + 34) = J + 10
      MINC(2,M + 34) = J + 11
      MINC(1,M + 35) = J + 4
      MINC(2,M + 35) = J + 25
      MINC(1,M + 36) = J + 15
      MINC(2,M + 36) = J + 24
C
C IF THE CENTERLINE OF THE TOP DECKBOARD OVER THE OUTER BLOCK IS TO
C THE RIGHT OF THE CENTERLINE OF THE BLOCK:
C


      IF (XLC(JN(NTD)).GT.XLC(JN(NTD) + 5)) THEN
        MINC(1,M + 1) = J + 1
        MINC(2,M + 1) = J + 5
        MINC(1,M + 2) = J + 12
        MINC(2,M + 2) = J + 16
        MINC(1,M + 3) = J - 1
        MINC(2,M + 3) = J + 16
        MINC(1,M + 4) = J
        MINC(2,M + 4) = J + 5
        MINC(1,M + 5) = J
        MINC(2,M + 5) = J + 8
        MINC(1,M + 6) = J - 1
        MINC(2,M + 6) = J + 20
      END IF
C
C
C IF THE SUPPORT CONDITION IS #3:
C
      IF (ISUPC.EQ.3) THEN
        MINC(1,M + 16) = J + 15
        MINC(2,M + 16) = J + 18
        MINC(1,M + 18) = J + 23
        MINC(2,M + 18) = J + 26
        MINC(1,M + 31) = J + 4
        MINC(2,M + 31) = J + 7
        MINC(1,M + 33) = J + 11
        MINC(2,M + 33) = J + 27
      END IF
C
C FOR THE CASE OF CANTILEVERED TOP STRINGER BOARDS:
C
```

```
      IF (ICANT.EQ.1) THEN
        N = NE
        J = NJ
        MINC(1,N) = J - 3
        MINC(2,N) = J
        MINC(1,N-1) = J - 2
        MINC(2,N-1) = J - 1
        MINC(1,N-2) = J - 4
        MINC(2,N-2) = J - 2
        MINC(1,N-3) = J - 4
        MINC(2,N-3) = J - 3
        MINC(1,N-4) = J - 12
        MINC(2,N-4) = J - 3
        MINC(1,N-5) = J - 24
        MINC(2,N-5) = J - 2
      END IF
C
      RETURN
      END
****************************************************************
*                     STACKJ                                   *
****************************************************************
C THIS SUBROUTINE MODIFIES THE JCODE FOR THE STRUCTURE TO ACCOUNT FOR
C A STACKED CONDITION
C
      SUBROUTINE STACKJ(JN,NTD,JCODE,NBLK,NBLKPS)
      INTEGER JCODE(6,*),JN(*)
C
      DO 20 J = (JN(NTD) + 13),(JN(NTD) + 23)
        DO 10 L = 1,6
          JCODE(L,J) = 0
   10   CONTINUE
   20 CONTINUE
      DO 30 L = 1,6
        JCODE(L,JN(NTD) + 26) = 0
   30 CONTINUE
      IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
        DO 50 J = (JN(NTD) + 2),(JN(NTD) + 11)
          DO 40 L = 1,6
            JCODE(L,1) = 0
            JCODE(L,3) = 0
            IF (NBLK.EQ.9) THEN
              JCODE(L,2) = 0
              JCODE(L,4) = 0
            END IF
            JCODE(L,JN(NTD) + 27) = 0
            JCODE(L,J) = 0
   40     CONTINUE
   50   CONTINUE
      ELSE IF (NBLK.EQ.6) THEN
        IF (NBLKPS.EQ.2) THEN
          DO 80 J = (JN(NTD) + 2),(JN(NTD) + 11)
            DO 70 L = 1,6
              JCODE(L,JN(NTD) + 27) = 0
              JCODE(L,J) = 0
   70       CONTINUE
   80     CONTINUE
        END IF
      END IF
      RETURN
      END
****************************************************************
*                     CODES                                    *
****************************************************************
      SUBROUTINE CODES(JCODE,MINC,NE,NJ,MCODE,NEQ)
      INTEGER JCODE(6,*), MINC(2,*), MCODE(12,*)
C
      NEQ = 0
      DO 20 J = 1,NJ
        DO 10 L = 1,6
          IF (JCODE(L,J).NE.0) THEN
            NEQ = NEQ + 1
            JCODE(L,J) = NEQ
          END IF
   10   CONTINUE
   20 CONTINUE
      DO 40 I = 1,NE
        J = MINC(1,I)
        K = MINC(2,I)
        DO 30 L = 1,6
```

```
        MCODE(L,I) = JCODE(L,J)
        MCODE(L+6,I) = JCODE(L,K)
  30   CONTINUE
  40 CONTINUE
     RETURN
     END
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
•                       MBAND                              •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
     FUNCTION MBAND(MCODE,NE)
     INTEGER MCODE(12,*)
C
     MBAND = 0
     DO 10 I = 1,NE
       L = 1
   5   IF (MCODE(L,I).EQ.0.AND.L.LT.12) THEN
         L = L + 1
         GO TO 5
       END IF
       IS = MCODE(L,I)
       L = 12
   7   IF (MCODE(L,I).EQ.0.AND.L.GT.1) THEN
         L = L - 1
         GO TO 7
       END IF
       IL = MCODE(L,I)
       IDIF = IL - IS
       IF (IDIF.GT.MBAND) THEN
         MBAND = IDIF
       END IF
  10 CONTINUE
     RETURN
     END
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
•                       LENGTS                             •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
     SUBROUTINE LENGTS(MINC,ZLC,YLC,XLC,NTD,ICANT,NE,IFLAG,EL,MN)
     DOUBLE PRECISION ZLC(*),YLC(*),XLC(*),EL(*)
     INTEGER MINC(2,*),MN(13),IFLAG(*)
C
C  MEMBER LENGTHS:
C
C FIRSTLY, ALL MEMBERS UP TO THE EDGE OF OUTSIDE BLOCKS (INCLUDES THE
C LAST TOP D.B AND SPRING.
C
C  A FLAG IS ASSIGNED TO EACH ELEMENT IN THE MODEL: 1 - ST.BOARD; 2 -
C  DECKBOARD; 3 - VERTICAL BLOCK ELEMENT; 4 - HORIZ. BLOCK ELEMENT;
C  5 - SPRING.
C
     EL(1) = ZLC(MINC(2,1)) - ZLC(MINC(1,1))
     IFLAG(1) = 2
     EL(4) = EL(1)
     IFLAG(4) = 2
     EL(2) = YLC(MINC(2,2)) - YLC(MINC(1,2))
     IFLAG(2) = 3
     EL(3) = EL(2)
     IFLAG(3) = 3
     DO 190 K = 5,(MN(NTD)-5),8
       DO 170 I = 0,3
         EL(I + K) = XLC(MINC(2,(I + K))) - XLC(MINC(1,(I + K)))
         IFLAG(I + K) = 1
 170   CONTINUE
       M = K + 2
       DO 180 L = 2,4,2
         EL(M + L) = YLC(MINC(2,(M + L))) - YLC(MINC(1,(M + L)))
         IFLAG(M + L) = 5
         EL(M + L + 1) = ZLC(MINC(2,(M + L + 1))) - ZLC(MINC(1,(M + L + 1)))
         IFLAG(M + L + 1) = 2
         IF ((M + L + 1).EQ.MN(NTD)) THEN
           GO TO 195
         END IF
 180   CONTINUE
 190 CONTINUE
 195 K = MN(NTD)
     DO 200 I = 1,6
       EL(K + I) = XLC(MINC(2,(K + I))) - XLC(MINC(1,(K + I)))
       IFLAG(K + I) = 1
 200 CONTINUE
     K = MN(NTD) + 6
     DO 220 J = 1,2
```

```
      DO 210 I = 1,13
        IF (I.EQ.4.OR.I.EQ.5.OR.I.EQ.8.OR.I.GT.9) THEN
          EL(K + I) = XLC(MINC(2,(K + I))) - XLC(MINC(1,(K + I)))
          IFLAG(K + I) = 4
          IF(I.EQ.10.OR.I.EQ.11.OR.I.EQ.12)THEN
            IFLAG(K + I) = 1
          END IF
          IFLAG(K + 13) = 5
          EL(K + 13) = 0.0D + 0
        ELSE
          EL(K + I) = YLC(MINC(2,(K + I))) - YLC(MINC(1,(K + I)))
          IFLAG(K + I) = 5
          IF(I.EQ.6.OR.I.EQ.7)THEN
            IFLAG(K + I) = 3
          END IF
        END IF
 210   CONTINUE
      IF (J.EQ.1) THEN
        M = K + 13
        EL(M + 1) = YLC(MINC(2,(M + 1))) - YLC(MINC(1,(M + 1)))
        IFLAG(M + 1) = 5
        EL(M + 2) = ZLC(MINC(2,(M + 2))) - ZLC(MINC(1,(M + 2)))
        IFLAG(M + 2) = 2
        K = M + 2
      ELSE
        K = K + 13
      END IF
 220 CONTINUE
     DO 230 I = 1,2
      EL(K + I) = XLC(MINC(2,(K + I))) - XLC(MINC(1,(K + I)))
      IFLAG(K + I) = 1
 230 CONTINUE
C
     IF (ICANT.EQ.1) THEN
      DO 240 I = NE-5,NE
        IF (I.EQ.NE-3.OR.I.EQ.NE-2) THEN
          EL(NE-3) = 0.0D0
          IFLAG(NE-3) = 5
          EL(NE-2) = ZLC(MINC(2,NE-2)) - ZLC(MINC(1,NE-2))
          IFLAG(NE-2) = 2
        ELSE
          EL(I) = XLC(MINC(2,I)) - XLC(MINC(1,I))
          IFLAG(I) = 1
        END IF
 240   CONTINUE
     END IF
     DO 250 I = 1,NE
      EL(I) = ABS(EL(I))
      IF (IFLAG(I).NE.5.AND.EL(I).LT.0.1D0) THEN
        EL(I) = 0.1D0
      END IF
 250 CONTINUE
     RETURN
     END
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
•                     PROPS                               •
•••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
     SUBROUTINE PROPS(NTD,TSBE,TSBW,TSBT,BSBE,BSBW,BSBT,TDBE,TDW,MN,
    $     TDT,BDBE,BDW,BDT,BMOE,XMOE,W,T,XMG,EI,S,A,XMJ,EL,ICANT,MITD,
    $     BAREA,NE,BLKT,IFLAG,BLKL,BLKW,BDLOC,XLC,MINC,NBD,NTSB,
    $     NBSB,XPLENG,ISL)
     DOUBLE PRECISION TSBE,TSBW(2),BSBE,BSBW(2),TDW(*),BDW(*),
    $         W(*),T(*),EI(3,*),S(3,*),A(*),EL(*),BLKL(2),BLKW(2),
    $         TDBE,BDBE,XMOE(*),XMG(*),XMJ(*),TSBT,BSBT,TDT,BDT,
    $         BMOE,BAREA,BLKT,BDLOC(*),XLC(*),XPLENG
     INTEGER MN(13),IFLAG(*),MINC(2,*),ISL(*)
C
C  XMOE, WIDTH, THICKNESS AND MOMENT OF INERTIA FOR ALL ELEMENTS:
C
C  FIRSTLY, ALL STRINGER BOARD ELEMENTS:
C
     IF (NTSB.EQ.2) THEN
      TSBW(2) = 1.0D0
     END IF
     IF (NBSB.EQ.2) THEN
      BSBW(2) = 1.0D0
     END IF
     IF (NBSB.EQ.0) THEN
      BSBW(1) = 1.0D0
      BSBW(2) = 1.0D0
```

```
        BSBT = 0.1D0
      END IF

      DO 20 K = 5,(MN(NTD) - 5),8
        DO 10 I = 0,3
          IF (I.EQ.0) THEN
            XMOE(K + I) = TSBE
            W(K + I) = TSBW(1)
            T(K + I) = TSBT
          ELSE IF (I.EQ.1) THEN
            XMOE(K + I) = TSBE
            W(K + I) = TSBW(2)/2.0D00
            T(K + I) = TSBT
          ELSE IF (I.EQ.2) THEN
            XMOE(K + I) = BSBE
            W(K + I) = BSBW(1)
            T(K + I) = BSBT
          ELSE
            XMOE(K + I) = BSBE
            W(K + I) = BSBW(2)/2.0D00
            T(K + I) = BSBT
          END IF
 10     CONTINUE
 20   CONTINUE
      K = MN(NTD)
      DO 30 I = 1,6
        IF (I.EQ.1.OR.I.EQ.4.OR.I.EQ.5) THEN
          XMOE(K + I) = TSBE
          W(K + I) = TSBW(2)/2.0D00
          T(K + I) = TSBT
        ELSE
          XMOE(K + I) = TSBE
          W(K + I) = TSBW(1)
          T(K + I) = TSBT
        END IF
 30   CONTINUE
      K = MN(NTD) + 15
      DO 40 I = 1,3
        XMOE(K + I) = BSBE
        W(K + I) = BSBW(1)
        T(K + I) = BSBT
 40   CONTINUE
      K = MN(NTD) + 30
      DO 50 I = 1,6
        IF (I.EQ.4) THEN
          GO TO 50
        ELSE IF (I.EQ.6) THEN
          XMOE(K + I) = BSBE
          W(K + I) = BSBW(1)
          T(K + I) = BSBT
        ELSE
          XMOE(K + I) = BSBE
          W(K + I) = BSBW(2)/2.0D00
          T(K + I) = BSBT
        END IF
 50   CONTINUE
      IF (ICANT.EQ.1) THEN
        DO 60 I = NE,NE-5,-1
          IF (I.EQ.(NE-3)) THEN
            GO TO 60
          END IF
          IF (I.EQ.NE.OR.I.EQ.(NE-4)) THEN
            XMOE(I) = TSBE
            W(I) = TSBW(1)
          ELSE
            XMOE(I) = TSBE
            W(I) = TSBW(2)/2.0D00
          END IF
          T(I) = TSBT
 60     CONTINUE
      END IF
C
C ALL DECKBOARD ELEMENTS:
C A FLAG IS ASSIGNED TO ALL REAL BOTTOM
C DECKBOARDS SO AS TO IDENTIFY THEM LATER IN SUBROUTINE 'STACKL' AND
C APPLY MEMBER LOADS TO THESE DECKBOARDS.
C
        IF (NBD.EQ.0) THEN
          BDBE = 1.0D0
          BDT = .1D0
```

```
        XMOE(4) = 1.0D00
        W(4) = 1.0D00
        T(4) = 0.1D00
      END IF
      IF (MOD(NTD,2).EQ.0) THEN
        W(1) = 1.0D00
        XMOE(1) = 1.0D00
        T(1) = 0.1D00
        J = 0
      ELSE
        W(1) = TDW(MTTD)/2.0D00
        XMOE(1) = TDBE
        T(1) = TDT
        J = 1
      END IF
      DO 100 I = 10,MN(NTD),8
        IF (I.LT.MN(NTD)) THEN
          XMOE(I) = TDBE
          W(I) = TDW(MTTD+J)
          T(I) = TDT
          XMOE(I+2) = BDBE
          IF (NBD.EQ.0) THEN
            W(I+2) = 1.0D0
            T(I+2) = 0.1D0
          END IF
        ELSE IF (I.EQ.MN(NTD)) THEN
          XMOE(I) = TDBE
          W(I) = TDW(NTD)
          T(I) = TDT
        END IF
        J = J + 1
  100 CONTINUE
      IF (NBD.NE.0) THEN
        K = MN(NTD) + 21
        ISL(K) = 1
        XMOE(K) = BDBE
        W(K) = BDW(NBD)
        T(K) = BDT
C
        IF (MOD(NBD,2).EQ.0) THEN
          XMOE(4) = 1.0D0
          W(4) = 1.0D0
          T(4) = 0.1D0
          L = (NBD/2) + 1
        ELSE
          L = INT((NBD/2) + 2.5)
          ISL(4) = 1
          XMOE(4) = BDBE
          W(4) = BDW(L-1)/2.0D00
          T(4) = BDT
        END IF
        IF (L.EQ.NBD) THEN
          DO 110 K = 4,MN(NTD)-6,8
            IF (K.NE.4) THEN
              XMOE(K) = 1.0D0
              W(K) = 1.0D0
              T(K) = 0.1D0
            END IF
  110     CONTINUE
        ELSE
          N = 10
          DO 130 I = L,NBD-1
            X = (BDLOC(I) - 0.5*XPLENG)
            DO 120 J = N,MN(NTD),8
              IF (X.GT.XLC(MINC(1,J)) + 0.5*EL(J+3)) THEN
                XMOE(J+2) = 1.0D0
                W(J+2) = 1.0D0
                T(J+2) = 0.1D0
              ELSE
                ISL(J+2) = 1
                XMOE(J+2) = BDBE
                W(J+2) = BDW(I)
                T(J+2) = BDT
                N = J + 8
                IF (I.EQ.NBD-1) THEN
                  DO 115 M = N,MN(NTD)-8,8
                    XMOE(M+2) = 1.0D0
                    W(M+2) = 1.0D0
                    T(M+2) = 0.1D0
  115             CONTINUE
```

```
           END IF
           GO TO 130
         END IF
 120     CONTINUE
 130   CONTINUE
     END IF
   ELSE
     K = MN(NTD) + 21
     XMOE(K) = 1.0D0
     W(K) = 1.0D0
     T(K) = 0.1D0
   END IF
C
   IF (ICANT.EQ.1) THEN
     I = NE-2
     XMOE(I) = TDBE
     W(I) = TDW(NTD + 1)
     T(I) = TDT
   END IF
C
C ALL BLOCK ELEMENTS:
C
   XMOE(2) = BMOE
   W(2) = BLKW(1)
   T(2) = BLKL(2)
   A(2) = BAREA
   XMOE(3) = BMOE
   W(3) = BLKW(2)
   T(3) = BLKL(2)
   A(3) = BAREA
   K = MN(NTD) + 10
   DO 250 J = 1,2
     DO 240 I = 0,4
       XMOE(K + I) = BMOE
       IF(J.EQ.1) THEN
         W(K + I) = BLKW(1)
         T(K + I) = BLKL(1)
       ELSE
         W(K + I) = BLKW(2)
         T(K + I) = BLKL(1)
       END IF
       A(K + I) = BAREA
 240   CONTINUE
     K = K + 15
 250 CONTINUE
C
C FOR ALL REAL ELEMENTS, THE MOMENTS OF INERTIA ABOUT THE 3 AXES, AND
C THE AREA FOR ALL STRINGER BOARDS AND DECKBOARDS:
C
   DO 260 I = 1,NE
     IF (IFLAG(I).EQ.3.OR.IFLAG(I).EQ.4) THEN
C       EI(1,I) = W(I)*T(I)**3*(0.3333D0-0.20914*(T(I)/W(I))
C  $             *DTANH(1.5705D0*W(I)/T(I)))
      EI(1,I) = (22.36D0)*(22.36D0)**3*(0.3333D0-0.20914*(1.0D0)
  $             *DTANH(1.5705D0*1.0D0))
C       EI(2,I) = BLKT*BLKL(2)**3/12.0D0
      EI(2,I) = 22.36D0*(22.36D0)**3/12.0D0
C       EI(3,I) = BLKW(1)*BLKT**3/12.0D0
      EI(3,I) = 22.36D0*(22.36D0)**3/12.0D0
      S(1,I) = EI(1,I) * 2.0D0/BLKT
      S(2,I) = EI(2,I) * 2.0D0/BLKL(2)
      S(3,I) = EI(3,I) * 2.0D0/BLKT
      XMJ(I) = (BLKW(1)*BLKT/12.0D0)*(BLKW(1)**2 + BLKT**2)
     ELSE IF (IFLAG(I).EQ.1.OR.IFLAG(I).EQ.2) THEN
      A(I) = W(I) * T(I)
      EI(1,I) = W(I)*T(I)**3*(0.3333D0-0.20914*(T(I)/W(I))
  $             *DTANH(1.5705D0*W(I)/T(I)))
      EI(2,I) = T(I)*W(I)**3/12.0D0
      EI(3,I) = W(I)*T(I)**3/12.0D0
      S(2,I) = EI(2,I)*2.0D0/W(I)
      S(3,I) = EI(3,I)*2.0D0/T(I)
      S(1,I) = EI(1,I)*2.0D0/T(I)
     ENDIF
 260 CONTINUE
   RETURN
   END
*********************************************************************
*                    DUMMY                                 *
*********************************************************************
   SUBROUTINE DUMMY(NBLK,NTSB,NTD,ICANT,NE,XMOE,NBSB,MN,NBD,BDLOC,EL
```

```
     $            ,MINC,XPLENG,XLC)
      INTEGER MN(13),MINC(2,*)
      DOUBLE PRECISION XMOE(*),BDLOC(*),XPLENG,XLC(*),EL(*)
C
C THIS SUBROUTINE ASSIGNS ZERO MATERIAL PROPERTIES TO ALL ELEMENTS
C IN THE MODEL NOT IN THE REAL PALLET, (I.E ALL BOTTOM DECKBOARDS,
C BOTTOM SRINGERBOARDS, AND BLOCKS NOT REQUIRED:
C
      IF (NBLK.EQ.4.OR.NBLK.EQ.6) THEN
        IF (NBLK.EQ.4) THEN
          XMOE(2) = 1.0D0
        END IF
        IF (NTSB.EQ.2) THEN
          XMOE(3) = 1.0D0
          K = MN(NTD) + 25
          DO 10 I = 0,4
            XMOE(K + I) = 1.0D0
 10       CONTINUE
          DO 20 I = 6,MN(NTD)-2
            XMOE(I) = 1.0D0
            XMOE(I + 2) = 1.0D0
 20       CONTINUE
          K = MN(NTD)
          XMOE(K + 1) = 1.0D0
          XMOE(K + 4) = 1.0D0
          XMOE(K + 5) = 1.0D0
          K = MN(NTD) + 31
          XMOE(K) = 1.0D0
          XMOE(K + 1) = 1.0D0
          XMOE(K + 2) = 1.0D0
          XMOE(K + 4) = 1.0D0
          IF (ICANT.EQ.1) THEN
            XMOE(NE-1) = 1.0D0
            XMOE(NE-5) = 1.0D0
          END IF
        ELSE
          XMOE(2) = 1.0D0
          XMOE(3) = 1.0D0
        END IF
      END IF
      IF (NBSB.EQ.2) THEN
        DO 25 L = 8,MN(NTD)-2,8
          XMOE(L) = 1.0D0
 25     CONTINUE
        K = MN(NTD) + 35
        XMOE(K) = 1.0D0
      END IF
      IF (NBLK.EQ.8) THEN
        XMOE(3) = 1.0D0
      END IF
      IF (MOD(NTD,2).EQ.0) THEN
        XMOE(1) = 1.0D0
      END IF
      IF (NBD.EQ.0) THEN
        DO 30 L = 4,MN(NTD)-6,8
          XMOE(L) = 1.0D0
 30     CONTINUE
      ELSE
        IF (MOD(NBD,2).EQ.0) THEN
          XMOE(4) = 1.0D0
          L = (NBD/2) + 1
        ELSE
          L = INT((NBD/2) + 1.5)
        END IF
        N = 10
        DO 35 I = L,NBD-1
          X = (BDLOC(I) - 0.5*XPLENG)
          DO 33 J = N,MN(NTD),8
            IF (X.GT.XLC(MINC(1,J)) + 0.5*EL(J + 3)) THEN
              XMOE(J + 2) = 1.0D0
            ELSE
              N = J + 8
              GO TO 35
            END IF
 33       CONTINUE
 35     CONTINUE
      END IF
      IF (NBSB.EQ.0) THEN
        DO 40 L = 7,MN(NTD)-3,8
          XMOE(L) = 1.0D0
```

```
       XMOE(L + 1) = 1.0D0
  40  CONTINUE
      L = MN(NTD) + 35
      XMOE(L) = 1.0D0
      XMOE(L + 1) = 1.0D0
     END IF
     RETURN
     END
*****************************************************************
*                      RASUDL                              *
*****************************************************************
     SUBROUTINE RASUDL(EL,NJ,NTD,XTSB,ICANT,CANTL,XPLENG,F,Q,NE,NEQ,
    $  NTSB,MN,TDLOC,XLOAD,XTD,XPWIDT,MTTD,W,TDW,BLKW,BLKLOC,BLKL,WL,
    $  ISTACK,NBLK,NBLKPS)
     DOUBLE PRECISION F(12,*),Q(*),EL(*),TDLOC(*),TDW(15),W(*),BLKW(2),
    $ CANTL,XPLENG,XPWIDT,XTSB,XLOAD,XTD,XUDL,BLKL(2),BLKLOC(2,3),WL(*)
    $ ,TOTALA
     INTEGER MN(13)
C
C  THIS SUBROUTINE CALCULATES THE EQUIVALENT JOINT LOAD VECTOR FOR THE
C  CASE OF UNIFORMLY DISTRIBUTED LOADING AND PARTIAL UNIFORM LOADS
C  (LTYPE = 1, 2):
C  NOTE ALL LOADS OF THIS TYPE ARE CONSIDERED TO BE APPLIED DIRECTLY TO
C  THE DECKBOARDS.
C
     XLOAD = XLOAD/4.0D00
     XLUDL = XPLENG - (2.0D0*XTSB)
     WRITE(6,*) 'XLUDL = ',XLUDL
     IF (MOD(NTD,2).EQ.0) THEN
       L = MTTD
       TOTALA = 0.0D+0
     ELSE
       L = MTTD + 1
       TOTALA = W(1)*EL(1)
     END IF
     DO 5 I = 10,MN(NTD),8
       IF ((XLUDL/2.0D0).GT.(TDLOC(L)-.5D0*XPLENG)) THEN
         TOTALA = TOTALA + (W(I)*EL(I))
         L = L + 1
       END IF
   5 CONTINUE
     IF ((XLUDL/2.0D0).GT.((XPLENG/2.0D0)-CANTL)) THEN
       TOTALA = TOTALA + (W(NE-2)*EL(NE-2))
     END IF
     DA = XTD
     DB = (0.5D0*XPWIDT) - DA - (0.5D0*BLKW(1))
     WRITE(6,*) 'DA = ',DA,' DB = ',DB
     WRITE(6,*) 'TOTAL AREA = ',TOTALA
     Y1 = DB - DA
     Y2 = DB**2 - DA**2
     Y3 = DB**3 - DA**3
     Y4 = DB**4 - DA**4
     IF (MOD(NTD,2).NE.0) THEN
       WL(1) = (XLOAD*W(1))/TOTALA
       WRITE(6,*) 'LOAD = ',WL(1)
       F(2,1)= ( + WL(1)/(2.0D0*EL(1)**3))*(2.0D0*EL(1)**3*Y1-2.0D0*EL(1)
    $          *Y3 + Y4)
       F(6,1) = ( + WL(1)/(12.0D0*EL(1)**2))*(6.0D0*EL(1)**2*Y2
    $          -8.0D0*EL(1)*Y3 + 3.0D0*Y4)
       F(8,1) =( + WL(1)/(2.0D0*EL(1)**3))*(2.0D0*EL(1)*Y3 - Y4)
       F(12,1) = (-WL(1)/(12.0D0*EL(1)**2))*(4.0D0*EL(1)*Y3 - 3.0D0*Y4)
       IF (ISTACK.EQ.1) THEN
         IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
           IF (NBLK.EQ.8) THEN
             Q(1) = -F(8,1)
           END IF
         ELSE IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
           Q(1) = -F(8,1)
         ELSE
           Q(1) = -F(2,1)
           Q(3) = -F(6,1)
           Q(4) = -F(8,1)
         END IF
       ELSE
           Q(1) = -F(2,1)
           Q(3) = -F(6,1)
           Q(4) = -F(8,1)
       END IF
     END IF
     XLUDL = XPLENG - (2.0D0*XTSB)
```

```
      IF (ISTACK.EQ.1) THEN
        IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
          IF (NBLK.EQ.9) THEN
            K = 0
          ELSE
            K = 4
          END IF
        ELSE IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
          K = 4
        ELSE
          K = 10
        END IF
      ELSE
        K = 10
      END IF
      IF (MOD(NTD,2).EQ.0) THEN
        L = MTTD
      ELSE
        L = MTTD + 1
      END IF
      DO 10 I = 10,MN(NTD),8
        IF ((XLUDL/2.0D0).GT.(TDLOC(L)-.5D0*XPLENG)) THEN
          WL(I) = (XLOAD*W(I))/TOTALA
          WRITE(6,*) 'LOAD = ',WL(I)
          F(2,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)**3*Y1-2.0D0
     $         *EL(I)*Y3 + Y4)
          F(6,I) = ( + WL(I)/(12.0D0*EL(I)**2))*(6.0D0*EL(I)**2*Y2
     $         -8.0D0*EL(I)*Y3 + 3.0D0*Y4)
          F(8,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)*Y3-Y4)
          F(12,I) = (-WL(I)/(12.0D0*EL(I)**2))*(4.0D0*EL(I)*Y3-3.0D0*Y4)
          Q(K + 2) = -F(2,I)
          Q(K + 4) = -F(6,I)
          Q(K + 14) = -F(8,I)
          K = K + 30
        ELSE
          GO TO 20
        END IF
        L = L + 1
   10 CONTINUE
   20 IF (ICANT.EQ.1) THEN
        IF ((XLUDL/2.0D0).GT.((XPLENG/2.0D0)-CANTL)) THEN
          WL(NE-2) = (XLOAD*W(NE-2))/TOTALA
          WRITE(6,*) 'CANTILEVER WL = ',WL(NE-2)
          IF ((XLUDL/2.0D0).GT.((XPLENG/2.0D0)-CANTL)) THEN
            F(2,NE-2) =( + WL(NE-2)/(2.0D0*EL(NE-2)**3))*(2.0D0*EL(NE-2)
     $           **3*Y1 - 2.0D0*EL(NE-2)*Y3 + Y4)
            F(6,NE-2) = ( + WL(NE-2)/(12.0D0*EL(NE-2)**2))*(6.0D0*EL(NE-2)
     $           **2*Y2 - 8.0D0*EL(NE-2)*Y3 + 3.0D0*Y4)
            F(8,NE-2) = ( + WL(NE-2)/(2.0D0*EL(NE-2)**3))*(2.0D0*EL(NE-2)
     $            *Y3 - Y4)
            F(12,NE-2) = (-WL(NE-2)/(12.0D0*EL(NE-2)**2))*(4.0D0
     $            *EL(NE-2)*Y3 - 3.0D0*Y4)
            Q(NEQ-25) = -F(2,NE-2)
            Q(NEQ-23) = -F(6,NE-2)
            Q(NEQ-13) = -F(8,NE-2)
          END IF
        END IF
      END IF
      RETURN
       END
**************************************************************
*                      RALINE                        *
**************************************************************
      SUBROUTINE RALINE(LTYPE,CL,SL,XPLENG,XTSB,NTD,MTTD,TDLOC,TDW,EL,
     $          P1,P2,Q,F)
      DOUBLE PRECISION TDLOC(15),TDW(15),EL(*),Q(*),F(12,*),CL,SL,XPLENG
     *          ,P1,P2,XTSB
C
C  THIS SUBROUTINE CALCULATES THE EQUIVALENT JOINTFORCE VECTOR FOR
C  1, 2, AND 3 LINE LOADS (I.E LTYPE = 3, 4, 5). LOADS ARE CONSIDERED
C  TO BE APPLIED DIRECTLY TO THE STRINGERBOARDS AS POINT LOADS:
C  FOR THE CASE OF 2, AND 3 LINE LOADS (I.E LTYPE = 4, AND 5) THE SPAN
C  LOAD MAY NOT BE APPLIED TO THE RIGHT OF THE TOP DECKBOARD OVER THE
C  OUTER BLOCK.
C
      IF (LTYPE.EQ.3) THEN
        P1 = 0
        P2 = CL/2.0D0
      ELSE IF (LTYPE.EQ.4) THEN
        P1 = SL/3
```

```
     P2 = 0
   ELSE IF (LTYPE.EQ.5) THEN
     P1 = SL/3
     P2 = CL/2.0D0
   END IF
C
   IF (LTYPE.EQ.4.OR.LTYPE.EQ.5) THEN
     X = (0.5D0*XPLENG) - XTSB
     K = 13
     L = 12
     IF (MOD(NTD,2).EQ.0) THEN
      J = MTTD
     ELSE
      J = MTTD + 1
     END IF
     DO 20 I = J,NTD-1
      Y = (TDLOC(I) + (0.5D0*TDW(I))) - (0.5D0*XPLENG)
      Z = (TDLOC(I + 1) + (0.5D0*TDW(I + 1))) - (0.5D0*XPLENG)
      IF(X.GE.Y.AND.X.LE.Z) THEN
        A = X - Y
        B = EL(K) - A
        DO 10 M = 0,1
         IF (M.EQ.1) THEN
           P1 = P1/2
         END IF
         F(2,K + M) = (P1*B**2/EL(K + M)**3)*(3.0D0*A + B)
         F(6,K + M) = P1*A*B**2/EL(K + M)**2
         F(8,K + M) = (P1*A**2/EL(K + M)**3)*(A + 3.0D0*B)
         F(12,K + M) = -P1*A**2*B/EL(K + M)**2
         IF(M.EQ.0) THEN
           Q(L) = -F(2,K)
           Q(L + 4) = -F(6,K)
           Q(L + 30) = -F(8,K)
           Q(L + 34) = -F(12,K)
         ELSE
           Q(L + 12) = -F(2,K + 1)
           Q(L + 13) = -F(6,K + 1)
           Q(L + 42) = -F(8,K + 1)
           Q(L + 43) = -F(12,K + 1)
         END IF
 10      CONTINUE
        GO TO 30
       END IF
       K = K + 8
       L = L + 30
 20  CONTINUE
   END IF
 30 IF (LTYPE.EQ.3.OR.LTYPE.EQ.5) THEN
     Q(1) = -P2/3
     Q(4) = -P2/6
   END IF
   RETURN
   END
***************************************************************
*                    STACKL                          *
***************************************************************
C
C  THIS SUBROUINE CALCULATES THE FIXED END FORCES FOR THE BOTTOM
C  DECKBOARDS (WHERE PRESENT) IN THE STACK CONDITION. THESE DECKBOARDS
C  ARE IDENTIFIED BY THE FLAG - 'ISL'.
C
   SUBROUTINE STACKL(ISL,W,EL,XLOAD,WL,MN,F,Q,XTD,XPWIDT,BLKW,NE,
  $             NBLK,NBLKPS,NTD)
   DOUBLE PRECISION EL(*),XLOAD,WL(*),F(12,*),Q(*),XTD,XPWIDT,DA,DB
  $            ,BLKW(2),W(*)
   INTEGER ISL(*),MN(*)
C
   DA = XTD
   DB = (0.5D0*XPWIDT) - DA - (0.5D0*BLKW(1))
   Y1 = DB - DA
   Y2 = DB**2 - DA**2
   Y3 = DB**3 - DA**3
   Y4 = DB**4 - DA**4
   AREAB = 0.0D00
C
   DO 10 I = 1,NE
     IF (ISL(I).EQ.1) THEN
       AREAB = AREAB + (W(I)*EL(I))
     END IF
 10 CONTINUE
```

```
   IF (ISL(4).EQ.1) THEN
    WL(4) = (XLOAD*W(4))/AREAB
    WRITE(6,*) 'LOAD = ',WL(4)
    F(2,4) = ( + WL(4)/(2.0D0*EL(4)**3))*(2.0D0*EL(4)**3*Y1-2.0D0*EL(4)
   $        *Y3 + Y4)
    F(6,4) = ( + WL(4)/(12.0D0*EL(4)**2))*(6.0D0*EL(4)**2*Y2
   $         -8.0D0*EL(4)*Y3 + 3.0D0*Y4)
    F(8,4) =( + WL(4)/(2.0D0*EL(4)**3))*(2.0D0*EL(4)*Y3 - Y4)
    F(12,4) = (-WL(4)/(12.0D0*EL(4)**2))*(4.0D0*EL(4)*Y3 - 3.0D0*Y4)
    IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
     IF (NBLK.EQ.8) THEN
      Q(3) = -F(8,4)
     END IF
    ELSE IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
     Q(3) = -F(8,4)
    ELSE
     Q(6) = -F(2,4)
     Q(8) = -F(6,4)
     Q(9) = -F(8,4)
    END IF
   END IF
   IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
    IF (NBLK.EQ.9) THEN
     K = 15
    ELSE
     K = 19
    END IF
   ELSE IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
    K = 19
   ELSE
    K = 25
   END IF
   DO 20 I = 4,MN(NTD)-6,8
    IF (I.NE.4) THEN
     IF (ISL(I).EQ.1) THEN
      WL(I) = (XLOAD*W(I))/AREAB
      WRITE(6,*) 'LOAD = ',WL(I)
      F(2,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)**3*Y1-2.0D0
   $           *EL(I)*Y3 + Y4)
      F(6,I) = ( + WL(I)/(12.0D0*EL(I)**2))*(6.0D0*EL(I)**2*Y2
   $           -8.0D0*EL(I)*Y3 + 3.0D0*Y4)
      F(8,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)*Y3-Y4)
      F(12,I) = (-WL(I)/(12.0D0*EL(I)**2))*(4.0D0*EL(I)
   $           *Y3 - 3.0D0*Y4)
      Q(K + 8) = -F(2,I)
      Q(K + 10) = -F(6,I)
      Q(K + 14) = -F(8,I)
     END IF
     K = K + 30
    END IF
20 CONTINUE
   I = MN(NTD) + 21
   IF (ISL(I).EQ.1) THEN
    WL(I) = (XLOAD*W(I))/AREAB
    WRITE(6,*) 'LOAD = ',WL(I)
    F(2,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)**3*Y1-2.0D0
   $          *EL(I)*Y3 + Y4)
    F(6,I) = ( + WL(I)/(12.0D0*EL(I)**2))*(6.0D0*EL(I)**2*Y2
   $          -8.0D0*EL(I)*Y3 + 3.0D0*Y4)
    F(8,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)*Y3-Y4)
    F(12,I) = (-WL(I)/(12.0D0*EL(I)**2))*(4.0D0*EL(I)
   $          *Y3 - 3.0D0*Y4)
    IF (NBLK.EQ.4) THEN
     Q(213) = -F(8,I)
    END IF
    IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
     Q(213) = -F(8,I)
    END IF
   END IF
   RETURN
   END
*****************************************************************
*                  RAELMS                             *
*****************************************************************
   SUBROUTINE RAELMS(N,NTD,GAM4,GAM6,G,IFLAG,ISUPC,XMOE,
   $           EL,A,XMG,XMJ,EI,MN,XLC,JN,GAM3,GAM5,ICANT,NE)
   DOUBLE PRECISION G(27),EI(3,*),EL(*),A(*),XLC(*),XMOE (*),XMG(*)
   $      ,XMJ(*),GAM3,GAM4,GAM5,GAM6,XL(3,3)
   INTEGER IFLAG(*),MN(13),JN(13)
C
```

```
C  THIS SUBROUTINE CALCULATES THE ELEMENT STIFFNESS COEFFICIENTS FOR
C  ALL ELEMENTS, AND THEN THE DIRECTION COSINES FOR THE REAL ELEMENTS:
C
      DO 20 I = 1,3
        DO 10 J = 1,3
          XL(I,J) = 0.0D0
   10   CONTINUE
   20 CONTINUE
      IF (EL(N).EQ.0) THEN
        DO 30 J = 1,27
          G(J) = 0.0D + 00
   30   CONTINUE
        IF (N.LT.MN(NTD).OR.N.EQ.MN(NTD) + 19.OR.N.EQ.MN(NTD) + 20.OR.N.
     $     EQ.MN(NTD) + 34) THEN
          IF (N.EQ.MN(NTD) + 34.OR.N.EQ.MN(NTD) + 19) THEN
            G(1) = 10.0D + 06
            G(4) = 10.0D + 06
            G(6) = 10.0D + 06
            G(7) = 10.0D + 06
            G(10) = 10.0D + 06
            G(12) = GAM3
            G(22) = -10.0D + 06
            G(25) = -10.0D + 06
            G(27) = -GAM3
          ELSE IF(N.LT.MN(NTD)) THEN
            G(1) = 10.0D + 06
            G(4) = 10.0D + 06
            G(6) = 10.0D + 06
            G(7) = 90.0D + 03
            G(10) = 10.0D + 06
            G(22) = -90.0D + 03
            G(25) = -10.0D + 06
          ELSE
            G(6) = 10.0D + 06
            G(10) = 10.0D + 06
            G(25) = -10.0D + 06
          END IF
        ELSE IF (N.EQ.(MN(NTD) + 7).OR.N.EQ.(MN(NTD) + 22)) THEN
          G(1) = 10.0D + 06
          G(4) = 10.0D + 06
          G(6) = 10.0D + 06
          G(10) = 10.0D + 06
          G(12) = GAM3
          G(25) = -10.0D + 06
          G(27) = -GAM3
        ELSE IF (N.EQ.(MN(NTD) + 8).OR.N.EQ.(MN(NTD) + 23)) THEN
          G(4) = GAM4
          G(6) = 10.0D + 06
          G(10) = 10.0D + 06
          G(25) = -10.0D + 06
        ELSE IF (N.EQ.(MN(NTD) + 9).OR.N.EQ.(MN(NTD) + 24)) THEN
          G(4) = GAM6
        ELSE IF (N.EQ.(MN(NTD) + 15).OR.N.EQ.(MN(NTD) + 30)) THEN
          G(4) = GAM5
        END IF
        IF (ICANT.EQ.1) THEN
          IF (N.EQ.(NE-3)) THEN
            G(1) = 10.0D + 06
            G(4) = 10.0D + 06
            G(6) = 10.0D + 06
            G(7) = 90.0D + 03
            G(10) = 10.0D + 06
            G(22) = -90.0D + 03
            G(25) = -10.0D + 06
          END IF
        END IF
      ELSE
        IF (IFLAG(N).EQ.1.OR.IFLAG(N).EQ.4) THEN
          XL(1,1) = 1
          XL(2,2) = 1
          XL(3,3) = 1
        ELSE IF (IFLAG(N).EQ.2) THEN
          XL(1,3) = -1
          XL(2,2) = 1
          XL(3,1) = 1
        ELSE IF (IFLAG(N).EQ.3) THEN
          XL(1,2) = -1
          XL(2,1) = 1
          XL(3,3) = 1
        END IF
```

```
      IF (N.EQ.(MN(NTD) + 35).OR.N.EQ.(MN(NTD) + 36)) THEN
          XL(1,1) = -1
          XL(2,2) = -1
          XL(3,3) = 1
      END IF
      IF(N.EQ.(MN(NTD) + 17).OR.N.EQ.(MN(NTD) + 32)) THEN
        IF (ISUPC.EQ.1.OR.ISUPC.EQ.2) THEN
          XL(1,1) = -1
          XL(2,2) = -1
          XL(3,3) = 1
        END IF
      END IF
      IF (N.EQ.(MN(NTD) + 18).OR.N.EQ.(MN(NTD) + 33)) THEN
        IF (ISUPC.EQ.3) THEN
          XL(1,1) = -1
          XL(2,2) = -1
          XL(3,3) = 1
        END IF
      END IF
C
C  FOR THE CASE WHERE THE TOP D.B OVER BLOCK IS TO THE RIGHT OF TOP
C  D.B SPRING - THE DIRECTION COSINES CHANGE FOR THE FOLLOWING ELEMENTS.
C
      IF (N.EQ.(MN(NTD) + 1).OR.N.EQ.(MN(NTD) + 2).OR.N.EQ.(MN(NTD) + 3).
     $    OR.N.EQ.(MN(NTD) + 4)) THEN
        IF (XLC(JN(NTD)).GT.XLC(JN(NTD) + 5)) THEN
          IF (N.EQ.(MN(NTD) + 3).OR.N.EQ.(MN(NTD) + 4)) THEN
            XL(1,1) = -1
            XL(2,2) = -1
            XL(3,3) = 1
          END IF
        ELSE
          IF (N.EQ.(MN(NTD) + 1).OR.N.EQ.(MN(NTD) + 2)) THEN
            XL(1,1) = -1
            XL(2,2) = -1
            XL(3,3) = 1
          END IF
        END IF
      END IF
C
      AL = (XMOE(N) * EI(3,N)) EL(N)**3
      B = (XMOE(N) * EI(2,N)) EL(N)**3
      GA = (XMOE(N) * A(N)) EL(N)
      DE = (XMG(N) * EI(1,N)),EL(N)
C
      G(1)= GA*XL(1,1)**2 + 12*AL*XL(2,1)**2 + 12*B*XL(3,1)**2
      G(2)= GA*XL(1,1)*XL(1,2) + 12.0D0*AL*XL(2,1)*XL(2,2)
     $      + 12.0D0*B*XL(3,1)*XL(3,2)
      G(3)= GA*XL(1,1)*XL(1,3) - 12.0D0*AL*XL(2,1)*XL(2,3)
     $      + 12.0D0*B*XL(3,1)*XL(3,3)
      G(4)= GA*XL(1,2)**2 + 12.0D0*AL*XL(2,2)**2 + 12.0D0*B*XL(3,2)**2
      G(5)= GA*XL(1,2)*XL(1,3) + 12.0D0*AL*XL(2,2)*XL(2,3)
     $      + 12.0D0*B*XL(3,2)*XL(3,3)
      G(6)= GA*XL(1,3)**2 + 12.0D0*AL*XL(2,3)**2 + 12.0D0*B*XL(3,3)**2
      G(7)= DE*XL(1,1)**2 + 4.0D0*EL(N)**2*AL*XL(3,1)**2 + 4.0D0
     $      *EL(N)**2*B*XL(2,1)**2
      G(8)= DE*XL(1,1)*XL(1,2) + 4.0D0*EL(N)**2*AL*XL(3,1)*XL(3,2)
     $      + 4.0D0*EL(N)**2*B*XL(2,1)*XL(2,2)
      G(9)= DE*XL(1,1)*XL(1,3) + 4.0D0*EL(N)**2*AL*XL(3,1)*XL(3,3)
     $      + 4.0D0*EL(N)**2*B*XL(2,1)*XL(2,3)
      G(10)= DE*XL(1,2)**2 + 4.0D0*EL(N)**2*AL*XL(3,2)**2
     $      + 4.0D0*EL(N)**2*B*XL(2,2)**2
      G(11)= DE*XL(1,2)*XL(1,3) + 4.0D0*EL(N)**2*AL*XL(3,2)*XL(3,3) +
     $      4.0D0*EL(N)**2*B*XL(2,2)*XL(2,3)
      G(12)= DE*XL(1,3)**2 - 4.0D0*EL(N)**2*AL*XL(3,3)**2
     $      + 4.0D0*EL(N)**2*B*XL(2,3)**2
      G(13)= 6.0D0*EL(N)*AL*XL(2,1)*XL(3,1)-6.0D0*EL(N)*B
     $      *XL(3,1)*XL(2,1)
      G(14)= 6.0D0*EL(N)*AL*XL(2,1)*XL(3,2)-6.0D0*EL(N)*B
     $      *XL(3,1)*XL(2,2)
      G(15)= 6.0D0*EL(N)*AL*XL(2,1)*XL(3,3)-6.0D0*EL(N)*B
     $      *XL(3,1)*XL(2,3)
      G(16)= 6.0D0*EL(N)*AL*XL(2,2)*XL(3,1)-6.0D0*EL(N)*B
     $      *XL(3,2)*XL(2,1)
      G(17)= 6.0D0*EL(N)*AL*XL(2,2)*XL(3,2)-6.0D0*EL(N)*B
     $      *XL(3,2)*XL(2,2)
      G(18)= 6.0D0*EL(N)*AL*XL(2,2)*XL(3,3)-6.0D0*EL(N)*B
     $      *XL(3,2)*XL(2,3)
      G(19)= 6.0D0*EL(N)*AL*XL(2,3)*XL(3,1)-6.0D0*EL(N)*B
     $      *XL(3,3)*XL(2,1)
```

```
      G(20) =  6.0D0*EL(N)*AL*XL(2,3)*XL(3,2)-6.0D0*EL(N)*B
     $       *XL(3,3)*XL(2,2)
      G(21) =  6.0D0*EL(N)*AL*XL(2,3)*XL(3,3)-6.0D0*EL(N)*B
     $       *XL(3,3)*XL(2,3)
      G(22) =  -DE*XL(1,1)**2 + 2.0D0*EL(N)**2*AL*XL(3,1)**2
     $        +  2.0D0*EL(N)**2*B*XL(2,1)**2
      G(23) =  -DE*XL(1,1)*XL(1,2) + 2.0D0*EL(N)**2*AL*XL(3,1)*XL(3,2) +
     $       2.0D0*EL(N)**2*B*XL(2,1)*XL(2,2)
      G(24) =  -DE*XL(1,1)*XL(1,3)  +  2.0D0*EL(N)**2*AL*XL(3,1)*XL(3,3) +
     $       2.0D0*EL(N)**2*B*XL(2,1)*XL(2,3)
      G(25) =  -DE*XL(1,2)**2 + 2.0D0*EL(N)**2*AL*XL(3,2)**2
     $        +  2.0D0*EL(N)**2*B*XL(2,2)**2
      G(26) =  -DE*XL(1,2)*XL(1,3)  +  2.0D0*EL(N)**2*AL*XL(3,2)*XL(3,3) +
     $       2.0D0*EL(N)**2*B*XL(2,2)*XL(2.3)
      G(27) =  -DE*XL(1,3)**2 + 2.0D0*EL(N)**2*AL*XL(3,3)**2
     $        +  2.0D0*EL(N)**2*B*XL(2,3)**2
C
     END IF
     RETURN
     END
****************************************************************
*                      RAASSM                          *
****************************************************************
     SUBROUTINE RAASSM(N,MBD,G,SS,MCODE,MXNEQ)
     DOUBLE PRECISION G(27),SS(600.600)
     INTEGER INDEX(12,12),MCODE(12,*)
C
C  THIS SUBROUTINE ASSEMBLES THE ELEMENT STIFFNESS COEFFICIENTS INTO
C  THE SYSTEM STIFFNESS MATRIX:
C
     DATA INDEX/1,2,3,13,14,15,-1,-2,-3,13,14,15,2,4,5,16,17,
    $          18,-2,-4,-5,16,17,18,3,5,6,19,20,21,-3,-5,-6,19,20,
    $          21,13,16,19,7,8,9,-13,-16,-19,22,23,24,14,17,20,8,
    $          10,11,-14,-17,-20,23,25,26,15,18,21,9,11,12,-15,
    $          -18,-21,24,26,27,-1,-2,-3,-13,-14,-15,1,2,3,-13,
    $          -14,-15,-2,-4,-5,-16,-17,-18,2,4,5,-16,-17,-18,-3,
    $          -5,-6,-19,-20,-21,3,5,6,-19,-20,-21,13,16,19,22,
    $          23,24,-13,-16,-19,7,8,9,14,17,20,23,25,26,-14,-17,
    $          -20,8,10,11,15,18,21,24,26,27,-15,-18,-21,9,11,12/
C
     DO 20 JE  =  1,12
      J  =  MCODE(JE,N)
       IF (J.NE.0) THEN
        DO 10 IE  =  1,JE
         I  =  MCODE(IE,N)
         IF (I.NE.0) THEN
          K  =  I - J  +  MBD  +  1
          L  =  INDEX(IE,JE)
          IF (L.GT.0) THEN
           SS(K,J)  =  SS(K,J)  +  G(L)
          ELSE
           SS(K,J)  =  SS(K,J)  -  G(-L)
          END IF
         END IF
  10    CONTINUE
       END IF
  20 CONTINUE
     RETURN
     END
****************************************************************
*                      SOLVE                           *
****************************************************************
C
     SUBROUTINE SOLVE(SS,Q,NEQ,MBD,MXNEQ)
C
C        THIS SUBROUTINE CALLS SPBFA AND SPBSL FOR THE FIRST XLOAD
C     CONDITION, LC  =  1. FOR SUBSEQUENT XLOAD CONDITIONS, LC GREATER
C     THAN 1, IT CALLS SPBSL.
C
     DOUBLE PRECISION SS(600,600),Q(*)
C     INTEGER SS(MXNEQ,MXNEQ),Q(*)
C     IF (LC.EQ.1)THEN
      CALL SPBFA(SS,MXNEQ,NEQ,MBD,INFO)
      IF (INFO.NE.0)THEN
        PRINT *,'SINGULARITY...'
        STOP
      END IF
C     END IF
     CALL SPBSL(SS,MXNEQ,NEQ,MBD,Q)
     RETURN
```

```
      END
C     THIS IS THE LINPACK SOLUTION ALGORITHM WITH COMMENTS.
C-----------------------------------------------------------------
      SUBROUTINE SPBFA(ABD,LDA,N,M,INFO)
      INTEGER LDA,N,M,INFO
      DOUBLE PRECISION ABD(600,*)
C     REAL ABD(LDA,*)
C
C     SPBFA FACTORS A REAL SYMMETRIC POSITIVE DEFINITE
C     MATRIX STORED IN BAND FORM.
C
C     SPBFA IS USUALLY CALLED BY SPBCO, BUT IT CAN BE CALLED
C     DIRECTLY WITH A SAVING IN TIME IF  RCOND  IS NOT NEEDED.
C
C     ON ENTRY
C
C        ABD     REAL(LDA, N)
C                THE MATRIX TO BE FACTORED.  THE COLUMNS OF THE UPPER
C                TRIANGLE ARE STORED IN THE COLUMNS OF ABD AND THE
C                DIAGONALS OF THE UPPER TRIANGLE ARE STORED IN THE
C                ROWS OF ABD .  SEE THE COMMENTS BELOW FOR DETAILS.
C
C        LDA     INTEGER
C                THE LEADING INTEGER OF THE ARRAY  ABD .
C                LDA MUST BE .GE. M + 1 .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
C
C        M       INTEGER
C                THE NUMBER OF DIAGONALS ABOVE THE MAIN DIAGONAL.
C                0 .LE. M .LT. N .
C
C     ON RETURN
C
C        ABD     AN UPPER TRIANGULAR MATRIX  R , STORED IN BAND
C                FORM, SO THAT  A = TRANS(R)*R .
C
C        INFO    INTEGER
C                = 0  FOR NORMAL RETURN.
C                = K  IF THE LEADING MINOR OF ORDER  K  IS NOT
C                     POSITIVE DEFINITE.
C
C     BAND STORAGE
C
C           IF  A  IS A SYMMETRIC POSITIVE DEFINITE BAND MATRIX,
C           THE FOLLOWING PROGRAM SEGMENT WILL SET UP THE INPUT.
C
C                   M = (BAND WIDTH ABOVE DIAGONAL)
C                   DO 20 J = 1, N
C                      II = MAX0(1, J-M)
C                      DO 10 I = II, J
C                         K = I-J+M+1
C                         ABD(K,J) = A(I,J)
C                10    CONTINUE
C                20 CONTINUE
C
C     LINPACK.  THIS VERSION DATED 08/14/78 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LAB.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS SDOT
C     FORTRAN MAX0,SQRT
C
C     INTERNAL VARIABLES
C
      DOUBLE PRECISION SDOT,T
      DOUBLE PRECISION S
      INTEGER IK,J,JK,K,MU
C     BEGIN BLOCK WITH ...EXITS TO 40
C
C
         DO 30 J = 1, N
            INFO = J
            S = 0.0D0
            IK = M + 1
            JK = MAX0(J-M,1)
            MU = MAX0(M+2-J,1)
            IF (M .LT. MU) GO TO 20
```

```
            DO 10 K = MU, M
               T = ABD(K,J) - SDOT(K-MU,ABD(IK,JK),1,ABD(MU,J),1)
               T = T/ABD(M + 1,JK)
               ABD(K,J) = T
               S = S + T*T
               IK = IK - 1
               JK = JK + 1
   10       CONTINUE
   20       CONTINUE
            S = ABD(M + 1,J) - S
C     ......EXIT
            IF (S .LE. 0.0D0) GO TO 40
            ABD(M + 1,J) = SQRT(S)
   30    CONTINUE
         INFO = 0
   40 CONTINUE
      RETURN
      END
C-----------------------------------------------------
      SUBROUTINE SPBSL(ABD,LDA,N,M,B)
      INTEGER LDA,N,M
      DOUBLE PRECISION ABD(600,*),B(*)
C     REAL ABD(LDA,*),B(*)
C
C     SPBSL SOLVES THE REAL SYMMETRIC POSITIVE DEFINITE
C     BAND SYSTEM  A*X = B
C     USING THE FACTORS COMPUTED BY SPBCO OR SPBFA.
C
C     ON ENTRY
C
C        ABD     REAL(LDA, N)
C                THE OUTPUT FROM SPBCO OR SPBFA.
C
C        LDA     INTEGER
C                THE LEADING INTEGER OF THE ARRAY  ABD .
C
C        N       INTEGER
C                THE ORDER OF THE MATRIX  A .
C
C        M       INTEGER
C                THE NUMBER OF DIAGONALS ABOVE THE MAIN DIAGONAL.
C
C        B       REAL(N)
C                THE RIGHT HAND SIDE VECTOR.
C
C     ON RETURN
C
C        B       THE SOLUTION VECTOR  X .
C
C     ERROR CONDITION
C
C        A DIVISION BY ZERO WILL OCCUR IF THE INPUT FACTOR CONTAINS
C        A ZERO ON THE DIAGONAL.  TECHNICALLY THIS INDICATES
C        SINGULARITY BUT IT IS USUALLY CAUSED BY IMPROPER SUBROUTINE
C        ARGUMENTS.  IT WILL NOT OCCUR IF THE SUBROUTINES ARE CALLED
C        CORRECTLY AND  INFO .EQ. 0 .
C
C     TO COMPUTE  INVERSE(A) * C  WHERE  C  IS A MATRIX
C     WITH  P  COLUMNS
C           CALL SPBCO(ABD,LDA,N,RCOND,Z,INFO)
C           IF (RCOND IS TOO SMALL .OR. INFO .NE. 0) GO TO ...
C           DO 10 J = 1, P
C              CALL SPBSL(ABD,LDA,N,C(1,J))
C        10 CONTINUE
C
C     LINPACK.  THIS VERSION DATED 08/14/78 .
C     CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LAB.
C
C     SUBROUTINES AND FUNCTIONS
C
C     BLAS SAXPY,SDOT
C     FORTRAN MIN0
C
C     INTERNAL VARIABLES
C
      DOUBLE PRECISION SDOT,T
      INTEGER K,KB,LA,LB,LM
C
C     SOLVE TRANS(R)*Y = B
C
```

```
      DO 10 K = 1, N
         LM = MIN0(K-1,M)
         LA = M + 1 - LM
         LB = K - LM
         T = SDOT(LM,ABD(LA,K),1,B(LB),1)
         B(K) = (B(K) - T)/ABD(M + 1,K)
   10 CONTINUE
C
C     SOLVE R*X = Y
C
      DO 20 KB = 1, N
         K = N + 1 - KB
         LM = MIN0(K-1,M)
         LA = M + 1 - LM
         LB = K - LM
         B(K) = B(K)/ABD(M + 1,K)
         T = -B(K)
         CALL SAXPY(LM,T,ABD(LA,K),1,B(LB),1)
   20 CONTINUE
      RETURN
      END
C-------------------------------------------------------
      SUBROUTINE SAXPY(N,SA,SX,INCX,SY,INCY)
C
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      DOUBLE PRECISION SX(*),SY(*),SA
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
      IF(N.LE.0)RETURN
      IF (SA .EQ. 0.0D+0) RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        SY(IY) = SY(IY) + SA*SX(IX)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,4)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        SY(I) = SY(I) + SA*SX(I)
   30 CONTINUE
      IF( N .LT. 4 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,4
        SY(I) = SY(I) + SA*SX(I)
        SY(I + 1) = SY(I + 1) + SA*SX(I + 1)
        SY(I + 2) = SY(I + 2) + SA*SX(I + 2)
        SY(I + 3) = SY(I + 3) + SA*SX(I + 3)
   50 CONTINUE
      RETURN
      END
C-------------------------------------------------------
      DOUBLE PRECISION FUNCTION SDOT(N,SX,INCX,SY,INCY)
C
C     FORMS THE DOT PRODUCT OF TWO VECTORS.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      DOUBLE PRECISION SX(*),SY(*),STEMP
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
      SDOT = 0.0D+0
```

```
      STEMP = 0.0D+0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C       CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C         NOT EQUAL TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        STEMP = STEMP + SX(IX)*SY(IY)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      SDOT = STEMP
      RETURN
C
C       CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C       CLEAN-UP LOOP
C
   20 M = MOD(N,5)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        STEMP = STEMP + SX(I)*SY(I)
   30 CONTINUE
      IF( N .LT. 5 ) GO TO 60
   40 MP1 = M + 1
      DO 50 I = MP1,N,5
        STEMP = STEMP + SX(I)*SY(I) + SX(I + 1)*SY(I + 1) +
     *   SX(I + 2)*SY(I + 2) + SX(I + 3)*SY(I + 3) + SX(I + 4)*SY(I + 4)
   50 CONTINUE
   60 SDOT = STEMP
      RETURN
      END
****************************************************************
*                      RAELMF                                  *
****************************************************************
      SUBROUTINE RAELMF(MCODE,Q,IFLAG,ISUPC,XMOE,I,EI,EL,A,XMG,XMJ,D,DL
     $             ,FLO,F,FL,MN,NTD,YMAX,MINC,NE,DEFL,DISPL,JT)
      DOUBLE PRECISION Q(*),EI(3,*),EL(*),D(*),DL(*),A(*),DEFL(*),
     $         FL(12,*),FLO(12),F(12,*),XMOE(*),XMG(*),XMJ(*),Y,YMAX
     $         ,DISPL(6,*)
      INTEGER IFLAG(*),MCODE(12,*),MN(*),MINC(2,*)
C
C  THIS SUBROUTINE TRANSFORMS THE GLOBAL DISPLACEMENTS INTO LOCAL
C  ELEMENT-END DISPLACEMNTS, AND THEN CALCULATES THE LOCAL ELEMENT
C  FORCES FOR EACH MEMBER:
C
      DO 10 L = 1,12
        K = MCODE(L,I)
        IF (K.NE.0) THEN
          D(L) = Q(K)
        ELSE
          D(L) = 0.0D+0
        END IF
        IF (L.EQ.2) THEN
          DEFL(MINC(1,I)) = ABS(D(2))
        ELSE IF (L.EQ.8) THEN
          DEFL(MINC(2,I)) = ABS(D(8))
        END IF
   10 CONTINUE
C
      Y = ABS(D(2))
      IF (Y.GT.YMAX) THEN
        YMAX = Y
        MEM = I
        JT = MINC(1,I)
      END IF
C
      DISPL(1,MINC(1,I)) = D(1)
      DISPL(2,MINC(1,I)) = D(2)
      DISPL(3,MINC(1,I)) = D(3)
      DISPL(4,MINC(1,I)) = D(4)
      DISPL(5,MINC(1,I)) = D(5)
      DISPL(6,MINC(1,I)) = D(6)
      DISPL(1,MINC(2,I)) = D(7)
```

```
      DISPL(2,MINC(2,I)) = D(8)
      DISPL(3,MINC(2,I)) = D(9)
      DISPL(4,MINC(2,I)) = D(10)
      DISPL(5,MINC(2,I)) = D(11)
      DISPL(6,MINC(2,I)) = D(12)
C
C      WRITE(6,200) I,(D(L),L = 1,12)
C 200 FORMAT (I2,6(1X,F11.8)/3X,6(1X,F11.8))
      IF (IFLAG(I).EQ.1.OR.IFLAG(I).EQ.4) THEN
        IF (I.EQ.MN(NTD) + 35.OR.I.EQ.MN(NTD) + 36) THEN
          DO 30 J = 0,9,3
            DL(J + 1) = -D(J + 1)
            DL(J + 2) = -D(J + 2)
            DL(J + 3) = D(J + 3)
   30     CONTINUE
        ELSE IF(I.EQ.MN(NTD) + 17.OR.I.EQ.MN(NTD) + 32.AND.ISUPC.EQ.1)THEN
          DO 40 J = 0,9,3
            DL(J + 1) = -D(J + 1)
            DL(J + 2) = -D(J + 2)
            DL(J + 3) = D(J + 3)
   40     CONTINUE
        ELSE IF(I.EQ.MN(NTD) + 17.OR.I.EQ.MN(NTD) + 32.AND.ISUPC.EQ.2)THEN
          DO 45 J = 0,9,3
            DL(J + 1) = -D(J + 1)
            DL(J + 2) = -D(J + 2)
            DL(J + 3) = D(J + 3)
   45     CONTINUE
        ELSE IF (I.EQ.MN(NTD) + 18.OR.I.EQ.MN(NTD) + 33.AND.ISUPC.EQ.3)THEN
          DO 50 J = 0,9,3
            DL(J + 1) = -D(J + 1)
            DL(J + 2) = -D(J + 2)
            DL(J + 3) = D(J + 3)
   50     CONTINUE
        ELSE
          DO 60 J = 1,12
            DL(J) = D(J)
   60     CONTINUE
        END IF
      ELSE IF (IFLAG(I).EQ.2) THEN
        DO 70 J = 0,9,3
          DL(J + 1) = -D(J + 3)
          DL(J + 2) = D(J + 2)
          DL(J + 3) = D(J + 1)
   70   CONTINUE
      ELSE IF (IFLAG(I).EQ.3) THEN
        DO 80 J = 0,9,3
          DL(J + 1) = -D(J + 2)
          DL(J + 2) = D(J + 1)
          DL(J + 3) = D(J + 3)
   80   CONTINUE
      END IF
C
      AL = (XMOE(I) * EI(3,I))/(EL(I)**3)
      B = (XMOE(I) * EI(2,I))/(EL(I)**3)
      GA = XMOE(I) * A(I)/EL(I)
C      DE = XMG(I) * XMJ(I)/EL(I)
      DE = XMG(I) * EI(1,I)/EL(I)
C
C  LOCAL FORCES AT A-END OF MEMBER FROM THE LOCAL ELEMENT MODEL (F = KD):
C
      FLO(1) = GA*DL(1) - GA*DL(7)
      FLO(2) = 12.0D0*AL*DL(2) + 6.0D0*EL(I)*AL*DL(6)-12.0D0*AL*DL(8) +
     $       6.0D0*EL(I)*AL*DL(12)
      FLO(3) = 12.0D0*B*DL(3) - 6.0D0*EL(I)*B*DL(5) -12.0D0*B*DL(9) -
     $       6.0D0*EL(I)*B*DL(11)
      FLO(4) = DE*DL(4) - DE*DL(10)
      FLO(5) = -6.0D0*EL(I)*B*DL(3)  +  4.0D0*EL(I)**2*B*DL(5)
     $         + 6.0D0*EL(I)*B*DL(9) +  2.0D0*EL(I)**2*B*DL(11)
      FLO(6) = 6.0D0*EL(I)*AL*DL(2)  +  4.0D0*EL(I)**2*AL*DL(6)
     $      - 6.0D0*EL(I)*AL*DL(8)  +  2.0D0*EL(I)**2*AL*DL(12)
C
C  LOCAL FORCES AT B-END OF MEMBER FROM EQUILIBRIUM:
C
      FLO(7) = -FLO(1)
      FLO(8) = -FLO(2)
      FLO(9) = -FLO(3)
      FLO(10) = -FLO(4)
      FLO(11) = -EL(I)*FLO(3) - FLO(5)
      FLO(12) = EL(I)*FLO(2) - FLO(6)
C
```

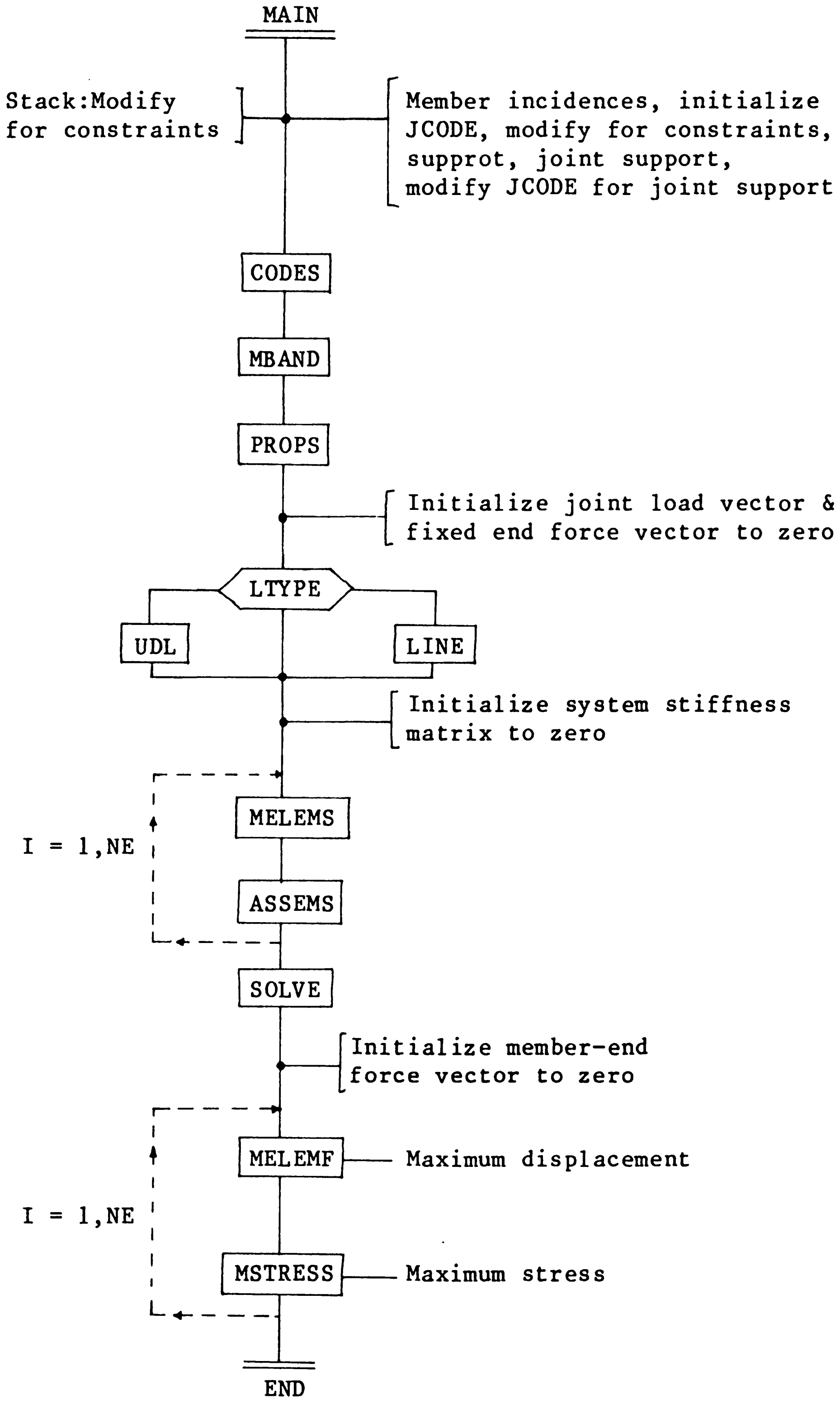
```
     DO 90 J = 1,6
       FL(J,I) = F(J,I) + FLO(J)
  90 CONTINUE
     FL(7,I) = F(7,I) - FLO(1)
     FL(8,I) = F(8,I) - FLO(2)
     FL(9,I) = F(9,I) - FLO(3)
     FL(10,I) = F(10,I) - FLO(4)
     FL(11,I) = F(11,I) - (FLO(3)*EL(I)) + FLO(5)
     FL(12,I) = F(12,I) - (FLO(2)*EL(I)) + FLO(6)
C
     RETURN
     END
*****************************************************************
*                        RASMST                          *
*****************************************************************
     SUBROUTINE RASMST(IFLAG,A,S,LTYPE,XPLENG,XTSB,MTTD,NTD,TDLOC,TDW,
    $               MN,FL,ST,STA,STB,P2,EL,FLO,I,SSMAX,DSMAX,
    $               NE,XTD,WL,DST,DSTA,DSTB)
     DOUBLE PRECISION FL(12,*),ST(*),STA(*),STB(*),EL(*),XTSB,
    $ SSMAX,X,XLOC,XPOS,TDLOC(15),TDW(15),A(*),S(3,*),FLO(12),XPLENG,P2
    $  ,DSMAX,BLOCAT,XTD,WL(*),DST(*),XMAX,DSTA(*),DSTB(*),XBMAX
     INTEGER IFLAG(*),MN(*)
C
C  THIS SUBROUTINE CALCULATES THE STRESSES IN EACH STRINGERBOARD ELEMENT
C  AND IDENTIFIES THE MAXIMUM VALUE AND WHICH MEMBER IT OCCURS IN:
C
     IF (IFLAG(I).EQ.1) THEN
       STA(I) = (ABS(FLO(1)/A(I))) + (ABS(FLO(5)/S(2,I))) + (ABS(FLO(6)
    $          /S(3,I)))
       STB(I) = (ABS(FLO(1)/A(I))) + (ABS(FLO(11)/S(2,I))) +
    $          (ABS(FLO(12)/S(3,I)))
       WRITE(6,*) I,STA(I),STB(I)
       IF (I.LT.MN(NTD)-2.OR.I.GT.MN(NTD) + 38) THEN
         IF (I.EQ.MN(NTD)-4.OR.I.EQ.MN(NTD)-5) GOTO 50
         IF (STA(I).GT.STB(I)) THEN
           X = STA(I)
           XLOC = 0.0D0
         ELSE
           X = STB(I)
           XLOC = EL(I)
         END IF
         IF (X.GT.SSMAX) THEN
           SSMAX = X
           MEM = I
           XPOS = XLOC
         END IF
       END IF
       IF (LTYPE.EQ.3.OR.LTYPE.EQ.4.OR.LTYPE.EQ.5) THEN
         IF (LTYPE.EQ.4.OR.LTYPE.EQ.5) THEN
           X = (0.5D0*XPLENG) - XTSB
             K = 13
           DO 20 J = MTTD,NTD
             Y = (TDLOC(J) + (0.5D0*TDW(J))) - (0.5D0*XPLENG)
             Z = (TDLOC(J + 1) + (0.5D0*TDW(J + 1))) - (0.5D0*XPLENG)
             IF(X.GE.Y.AND.X.LE.Z) THEN
               DO 10 M = 0,1
                 YY = X - Y
                 XBMAX = -FL(6,K + M) + FL(2,K + M)*YY
                 ST(K + M) =(ABS(FL(1,K + M)/A(K + M))) + (ABS(XBMAX/S(3,K + M)))
                 IF (ST(K + M).GT.SSMAX) THEN
                   SSMAX = ST(K + M)
                   MEM = K + M
                   XPOS = YY
                 END IF
 10            CONTINUE
             ELSE
               GO TO 30
             END IF
             K = K + 8
 20        CONTINUE
         END IF
 30      IF(LTYPE.EQ.3.OR.LTYPE.EQ.5) THEN
           IF (MOD(NTD,2).EQ.0) THEN
             XBMAX = 10.0D0
             ST(6) = (ABS(FL(1,6)/A(6))) + (ABS(XBMAX/S(3,6)))
           ELSE
             XBMAX = -FL(6,6)
             ST(6) = (ABS(FL(1,6)/A(6))) + (ABS(XBMAX/S(3,6)))
           END IF
           IF (ST(6).GT.SSMAX) THEN
```

```
          SSMAX = ST(6)
          MEM = 6
          XPOS = 0.0D00
        END IF
      END IF
    END IF
  ELSE
    IF (IFLAG(I).EQ.2) THEN
      DSTA(I) = (ABS(FLO(1)/A(I)))+(ABS(FLO(5)/S(2,I)))+(ABS(FLO(6)
$             /S(3,I)))
      DSTB(I) = (ABS(FLO(1)/A(I))) + (ABS(FLO(11)/S(2,I))) +
$             (ABS(FLO(12)/S(3,I)))
      IF (DSTA(I).GT.DSTB(I)) THEN
        DST(I) = DSTA(I)
        XMAX = 0.0D00
      ELSE
        DST(I) = DSTB(I)
        XMAX = EL(I)
      END IF
      IF (WL(I).NE.0.0D00) THEN
        XMAX = XTD + (ABS(FL(2,I)/WL(I)))
        XBMAX = -FL(6,I) + FL(2,I)*((XMAX + XTD)/2.0D00) + FL(5,I)
        DST(I) = ABS(FL(1,I)/A(I)) + ABS(XBMAX/S(2,I)) +
$               ABS(XBMAX/S(3,I))
      END IF
      IF (DST(I).GT.DSMAX) THEN
        DSMAX = DST(I)
        MEMB = I
        BLOCAT = XMAX
      END IF
    END IF
  END IF
  IF (I.EQ.NE) THEN
    WRITE(6,*)
    WRITE(6,*) 'MAXIMUM BENDING STRESS IN STRINGERBOARDS =',SSMAX
    WRITE(6,*) 'IN MEMBER ',MEM
    WRITE(6,*) 'AT (IN)',XPOS
C
    WRITE(6,*)
    WRITE(6,*) 'MAXIMUM BENDING STRESS IN DECKBOARDS =',DSMAX
    WRITE(6,*) 'IN MEMBER ',MEMB
    WRITE(6,*) 'AT (IN)',BLOCAT
  END IF
50 RETURN
  END
*****************************************************************
*                   STACKD                                      *
*****************************************************************
C  THIS SUBROUTINE CALCULATES THE MAXIMUM DEFLECTION IN THE TOP
C  STRINGERBOARDS FOR THE STACKED MODE
C
    SUBROUTINE STACKD(JN,NTD,XLC,DEFL,XMOE)
    DOUBLE PRECISION XLC(*),DEFL(*),X(20),Y(20),Y1MAX,XTARG,XLENG,
   $             X1(20),Y1(20),SUM,XMOE(*)
    INTEGER JN(*)
C
    IF (NTD.GT.3) THEN
      IF (XMOE(3).GT.1.0D00) THEN
        DO 30 II = 1,2
          IF (II.EQ.1) THEN
            L = 7
            M = 6
          ELSE
            L = 10
            M = 3
          END IF
          NMAX = 2
          X(1) = 0.0D00
          Y(1) = 0.0D00
          J = 1
          DO 10 I = L,JN(NTD)-M,6
            J = J + 1
            NMAX = NMAX + 1
            X(J) = XLC(I)
            Y(J) = DEFL(I)
 10       CONTINUE
          Y(NMAX) = 0.0D00
          IF (II.EQ.1) THEN
            X(NMAX) = XLC(JN(NTD)+1)
            XLENG = XLC(JN(NTD)+1)
```

```
      ELSE
       X(NMAX) = XLC(JN(NTD)+4)
       XLENG = XLC(JN(NTD)+4)
      END IF
      YIMAX = 0.0D00
      DO 20 K = 1,10
       XTARG = (K*XLENG)/10.0D00
       XI(K) = XTARG
       SUM = 0.0D00
       CALL LAGRAN(NMAX,X,SUM,Y,XTARG)
       YI(K) = SUM
       IF (YI(K).GT.YIMAX) THEN
        YIMAX = YI(K)
       END IF
 20    CONTINUE
      IF (II.EQ.1) THEN
       WRITE(6,*) 'MAX.DEFLECTION IN TOP STRINGERBOARD IS =',
  $              YIMAX
      ELSE
       WRITE(6,*) 'MAX.DEFLECTION IN BOTTOM STRINGERBOARD IS ='
  $             ,YIMAX
      END IF
 30    CONTINUE
    ELSE
     WRITE(6,*) 'MAX.DEFLECTION IN TOP STRINGERBOARD IS =',
  $              DEFL(2)
     WRITE(6,*) 'MAX.DEFLECTION IN BOTTOM STRINGERBOARD IS ='
  $             ,DEFL(4)
    END IF
   ELSE
    WRITE(6,*) 'TOO FEW DECKBOARDS FOR STACK ANALYSIS - MINIMUM
  $  NUMBER ALLOWED IS 3.'
   END IF
   RETURN
   END
C         ******************************
C                LAGRAN
C         ******************************
C
   SUBROUTINE LAGRAN(NMAX,X,SUM,Y,XTARG)
   DOUBLE PRECISION X(20),Y(20),SUM,XTARG,XHAT
   DO 20 I = 1,NMAX
    XHAT = 1.0D00
    DO 10 J = 1,NMAX
     IF (I.EQ.J) GO TO 10
     XHAT = XHAT*(XTARG - X(J))/(X(I) - X(J))
 10  CONTINUE
    SUM = SUM + (Y(I)*XHAT)
 20 CONTINUE
   RETURN
   END
```

# Appendix B:

## Program listing - (RAD)


MAIN
Stack:Modify for constraints
Member incidences, initialize JCODE, modify for constraints, supprot, joint support, modify JCODE for joint support
CODES
MBAND
PROPS
Initialize joint load vector & fixed end force vector to zero
LTYPE
UDL
LINE
Initialize system stiffness matrix to zero
I = 1,NE
MELEMS
ASSEMS
SOLVE
Initialize member-end force vector to zero
I = 1,NE
MELEMF
Maximum displacement
MSTRESS
Maximum stress
END

```
****************************************************************
*      RAD ANALYSIS......                                      *
****************************************************************
     IMPLICIT DOUBLE PRECISION (A-H,O-Z)
C
     DOUBLE PRECISION TDW(10,2),BDW(10,2),BLKW(2),BLKLOC(2,3),Q(60),
    $         EL(30),W(30),T(30),EI(30),A(30),S(30),
    $         F(6,30),G(7),SS(100,100),FL(6,30),P(3,20),FLO(12),C2(30)
    $         ,DL(6),ALPHA(30),BETA(30),ST(30),STA(30),STB(30),C1(30),
    $  DISPL(3,20),XMOE(30),XMG(30),XMMAX,XLOAD,BMOE,PLENGT,PWIDTH,TDBT
    $        ,BDBT,BLKT,RADSPA,TDO,CL,SL,XTD,GAM3,GAM4,GAM5,
    $     GAM6,X2,TDMOE,BDMOE,TDBE,BDBE,P1,P2,WINGL,TDBW,YMAX,YS,STMAX
     DIMENSION MCODE(6,30),MINC(2,21),JCODE(3,30)
     PARAMETER (MX  = 25,MXNEQ  =  3 * (MX-1),BMOE  = 10.0D + 06)
C
     DATA NTD,NBD,NTSB,NTDW,NBDW/1,1,3,1,1/
     DATA NBLK,NBLKPS/9,3/
     DATA PLENGT,PWIDTH/5.60D + 0,32.0D + 0/
C     DATA (TDW(L,1),L  =  1,9)/9*1.0D0/
C     DATA (TDW(L.,2),L  =  1,9)/9*3.5025/
C     DATA (BDW(L,1),L  =  1,3)/3*1.0D0/
C     DATA (BDW(L,2),L  =  1,3)/3*3.531/
     DATA TDW(1,1),TDW(1,2)/1.0D00,5.604D00/
     DATA BDW(1,1),BDW(1,2)/1.0D00,4.763D00/
     DATA TDBT,BDBT/.6250D + 0,.62500D + 0/
     DATA TDBE,BDBE/1.674540D + 06,1.672577D + 06/
     DATA BLKT,BLKW(1),BLKLOC(2,3)/3.125D + 0,3.125D + 0,28.875D + 0/
     DATA RADSPA,ISTYLE,TDO/28.875D + 0,1,0.0D + 0/
     DATA LTYPE,XLOAD,CL,SL/3,0.0000D + 00,50.0D + 01,0.0D + 00/
     DATA XTD/0.0D + 00/
     DATA GAM3/60.0D + 03/
     DATA GAM4,GAM5,GAM6/3.77266D + 05,0.00000D + 00,1.0D + 0/
     DATA ISTAC/0/
C
C  CALCULATE THE WING LENGTH IF TOP D.B IS WINGED :-
C
     IF (ISTYLE.EQ.2.OR.ISTYLE.EQ.3) THEN
       IWING  =  1
       WINGL  =  TDO
       IF (ISTAC.EQ.1) THEN
         RADSPA  = PWIDTH - TDO - (0.5*BLKW(1))
       END IF
     ELSE
       IWING  =  0
       IF (ISTAC.EQ.1) THEN
         RADSPA  = PWIDTH - TDO - (0.5*BLKW(1))
       END IF
     END IF
C
C  READING IN MEMBER INCIDENCES
C
     DATA MINC/1,2,1,4,2,3,4,8,8,10,4,5,8,9,10,11,5,9,9,11,5,
    $         6,11,12,6,12,6,7,3,7,12,13,7,14,13,14,10,15,15,16,17,4/
C
C  INITIALISE JCODE TO ZERO - THEN ADD IN CONSTRAINTS :-
C  3 DOF : 1 - TRANS IN 1 DIR, 2 - TRANS IN 2 DIR, 3 - ROTATION ABOUT 3
C  NOTE : 1  =  FREE, AND 0  =  CONSTRAINT
C
     DO 2 J  =  1,16
       DO 1 L  =  1,3
         JCODE(L,J)  =  1
    1  CONTINUE
    2 CONTINUE
C
     IF (ISTAC.EQ.1) THEN
       NJ  =  17
       NE  =  21
       MINC(2,2)  =  17
       MINC(1,21)  =  17
       MINC(2,21)  =  4
       JNTSUP  =  14
       ISUPC  =  2
       DO 4 I  =  1,17
         JCODE(1,I)  =  0
         JCODE(2,I)  =  0
         JCODE(3,I)  =  0
    4  CONTINUE
       JCODE(1,17)  =  1
       JCODE(2,17)  =  1
       JCODE(3,17)  =  1
       JCODE(1,3)  =  1
       JCODE(2,3)  =  1
       JCODE(3,3)  =  1
       JCODE(1,4)  =  1
       JCODE(2,4)  =  1
       JCODE(3,4)  =  1
```

```
      IF (NBLK.EQ.4) THEN
C        JCODE(1,1) = 1
        JCODE(2,1) = 1
C        JCODE(1,2) = 1
        JCODE(2,2) = 1
      END IF
      IF (NBLK.EQ.6) THEN
        IF (NBLKPS.EQ.3) THEN
C          JCODE(1,1) = 1
          JCODE(2,1) = 1
C          JCODE(1,2) = 1
          JCODE(2,2) = 1
        END IF
      END IF
     ELSE
      NJ = 16
      NE = 20
      JCODE(1,1) = 0
      JCODE(3,1) = 0
      JCODE(1,2) = 0
      JCODE(3,2) = 0
C
C  DETERMINE THE SUPPORT CONDITION AND CORRESPONDING SUPPORT JOINT:-
C    ISUPC = 1 (INSIDE THE OUTSIDE BLOCK) ; ISUPC = 2 (UNDER OUTSIDE
C    BLOCK) ; ISUPC = 3 (UNDER WINGED MEMBER).
C
      IF (.5*RADSPA.LT.(BLKLOC(2,NTSB)-.5*PWIDTH)) THEN
        ISUPC = 1
        JNTSUP = 3
      ELSE IF(.5*RADSPA.GT.(BLKLOC(2,NTSB)-(.5*PWIDTH)).AND.(.5*RADSPA
    $       ).LT.(BLKLOC(2,NTSB) + BLKW(1)-(.5*PWIDTH))) THEN
        ISUPC = 2
        JNTSUP = 14
      ELSE
        ISUPC = 3
        JNTSUP = 15
      END IF
      WRITE(6,200) IWING,ISUPC,JNTSUP
  200   FORMAT(1X,'IWING = ',I2,2X,'ISUPC = ',I2,2X,'JNTSUP = ',I2)
      WRITE(6,*) 'WING =',WINGL
C
C  MODIFYING THE JCODE FOR THE SUPPORT JOINT :
C
      IF (ISUPC.EQ.3) THEN
        JCODE(2,JNTSUP) = 0
      ELSE
        JCODE(1,JNTSUP) = 0
        JCODE(2,JNTSUP) = 0
      END IF
     END IF
     CALL CODES(JCODE,MINC,NE,NJ,MCODE,NEQ)
     MBD = MBAND(MCODE,NE)
     CALL PROPS(NTD,TDBE,NBD,BDBE,TDW,BLKT,PLENGT,BMOE,BLKLOC,
    $  TDBT,ISUPC,RADSPA,BDW,BDBT,BLKW,IWING,WINGL,NBLK,NBLKPS,EL,
    $  W,T,EI,A,S,XMOE,XMG,C1,C2,PWIDTH,NTSB,NTDW,NBDW,TDBW,NE,ISTAC)
     WRITE(6,400) 'MEMBER','MOE','LENGTH','AREA'
 400  FORMAT(/3X,A6,9X,A3,10X,A6,7X,A4/)
     DO 5 I = 1,NE
      WRITE(6,410) I,XMOE(I),EL(I),A(I)
   5 CONTINUE
 410  FORMAT(5X,I2,2X,F17.2,2X,F8.2,2X,3X,F8.2)
C
C      INITIALISE THE JOINT LOAD VECTOR AND THE FIXED END
C      FORCE VECTOR TO ZERO:
C
     DO 10 K = 1,NEQ
      Q(K) = 0.0D+0
  10 CONTINUE
     DO 30 I = 1,NE
      DO 20 L = 1,6
       F(L,I) = 0.0D+0
  20   CONTINUE
  30 CONTINUE
     IF (LTYPE.EQ.1.OR.LTYPE.EQ.2) THEN
      CALL UDL(XTD,XLOAD,ISUPC,IWING,EL,LTYPE,F,X2,Q,TDBW,ISTAC,NEQ,
    $         NBLK,NBLKPS)
     ELSE
      CALL LINE(CL,SL,LTYPE,EL,XTD,PWIDTH,F,Q,ISUPC,P1,P2)
     END IF
     WRITE(6,450) 'JOINT LOAD VECTOR:'
```

```
      DO 37 I = 1,NEQ
        WRITE(6,460) I,Q(I)
   37 CONTINUE
  450 FORMAT(/8X,A18/)
  460 FORMAT(8X,I2,5X,F10.4)
C
C      INITIALISE SYSTEM STIFFNESS MATRIX AND ELEMENT LOCAL FORCE
C      VECTOR TO ZERO :
C
      DO 50 J = 1,MXNEQ
        DO 40 I = 1,MXNEQ
          SS(I,J) = 0.0D+0
   40   CONTINUE
   50 CONTINUE
      DO 56 I = 1,30
        DO 55 L = 1,6
          FL(L,I) = 0.0D+0
   55   CONTINUE
   56 CONTINUE
      DO 60 N = 1,NE
        CALL MELEMS(GAM3,GAM4,GAM5,GAM6,XMOE,EI,A,EL,C1,C2,N,G
     $              ,W,T,NBLK,NBLKPS)
        CALL ASSEMS(MCODE,N,MBD,SS,G,MXNEQ)
   60 CONTINUE
      CALL SOLVE(SS,Q,NEQ,MBD,MXNEQ)
C
C
C     FOR ALL NON SPRING ELEMENTS CALCULATE THE FORCES AND STRESSES IN
C     EACH OF THE MEMBERS :
C
      YMAX = 0.0D00
      STMAX = 0.0D00
      WRITE(6,490) 'MEMBER #','STRESS A-END','STRESS B-END'
  490  FORMAT(//9X,A6,5X,A12,12X,A12/)
      DO 90 I = 1,NE
        IF (I.EQ.6.OR.I.EQ.7.OR.I.EQ.8.OR.I.EQ.14.OR.I.EQ.16)THEN
          GO TO 90
        ELSE
          CALL MELEMF(MCODE,Q,C1,C2,XMOE,I,EL,A,F,FL,DL,EI,ALPHA,BETA,
     $                FLO,YMAX,NE,MINC,DISPL,MEM,JT)
          CALL MSTRES(X2,I,LTYPE,XLOAD,FL,A,S,ISUPC,DL,ALPHA,BETA,ST,
     $      STA,STB,FLO,EL,XMOE,EI,CL,SL,P1,P2,PWIDTH,XTD,YS,STMAX,NE)
        END IF
   90 CONTINUE
      WRITE(6,520) 'JOINT DISPLACEMENTS:'
      WRITE(6,510) 'JOINT #','TRANSLATION 1-DIR','TRANSLATION 2-DIR',
     $           'ROTATION ABOUT 3-AXIS'
      DO 110 N = 1,NJ
        WRITE(6,500) N,DISPL(1,N),DISPL(2,N),DISPL(3,N)
  500   FORMAT(1X,I2,7X,F12.9,11X,F12.9,9X,F12.9)
  110 CONTINUE
  510 FORMAT(/A5,3X,A18,3X,A18,3X,A22/)
  520 FORMAT(//17X,A21)
      WRITE(6,*)
      WRITE(6,*) 'MAX. VERTICAL DEFL. =',YMAX,' AT JT.#',JT
      STOP
      END
C
C
C
*************************************************************
*                   CODES                                   *
*************************************************************
C
C     ASSIGN INTEGERS IN SEQUENCE TO ALL NON ZERO ELEMENTS OF JCODE:
C
      SUBROUTINE CODES(JCODE,MINC,NE,NJ,MCODE,NEQ)
      DIMENSION JCODE(3,*),MINC(2,*),MCODE(6,*)
      NEQ = 0
      DO 20 J = 1,NJ
        DO 10 L = 1,3
         IF (JCODE(L,J).NE.0) THEN
           NEQ = NEQ + 1
           JCODE(L,J) = NEQ
         END IF
   10   CONTINUE
   20 CONTINUE
      DO 40 I = 1,NE
        J = MINC(1,I)
        K = MINC(2,I)
```

```
      DO 30 L = 1,3
        MCODE(L,I) = JCODE(L,J)
        MCODE(L+3,I) = JCODE(L,K)
   30   CONTINUE
   40 CONTINUE
      RETURN
      END
*************************************************************
*                      MBAND                            *
*************************************************************
C
C  COMPUTE THE HALF BAND WIDTH:
C
      FUNCTION MBAND(MCODE,NE)
      DIMENSION MCODE(6,*)
      MBAND = 0
      DO 30 I = 1,NE
        L = 1
   10   IF (MCODE(L,I).EQ.0.AND.L.LT.6) THEN
          L = L + 1
          GO TO 10
        END IF
        IS = MCODE(L,I)
        L = 6
   20   IF (MCODE(L,I).EQ.0.AND.L.GT.1) THEN
          L = L - 1
          GO TO 20
        END IF
        IL = MCODE(L,I)
        IDIF = IL - IS
        IF (IDIF.GT.MBAND) THEN
          MBAND = IDIF
        END IF
   30 CONTINUE
      RETURN
      END
*************************************************************
*                     PROPS                            *
*************************************************************
C
C   THIS ROUTINE CALCULATES ALL MEMBER PROPERTIES FOR EACH OF THE
C   ELEMENTS - MOE, LENGTH, WIDTH, THICKNESS, AREA, SECTION MODULUS,
C   AND MOD. OF RIGIDITY :
C
      SUBROUTINE PROPS(NTD,TDBE,NBD,BDBE,TDW,BLKT,PLENGT,BMOE,BLKLOC
     $      ,TDBT,ISUPC,RADSPA,BDW,BDBT,BLKW,IWING,WINGL,NBLK,NBLKPS
     $      ,EL,W,T,EI,A,S,XMOE,XMG,C1,C2,PWIDTH,NTSB,NTDW,NBDW,TDBW,NE
     $      ,ISTAC)
      DOUBLE PRECISION EL(*),T(*),EI(*),A(*),S(*),BLKW(2),XMOE(*),XMG(*)
     $          ,W(*),BDW(10,2),TDW(10,2),BLKLOC(2,NTSB),C1(*),C2(*),TDBW
     $          ,BDBE,BLKT,PLENGT,BMOE,TDBT,RADSPA,BDBT,WINGL,TDBE,PWIDTH
C
C   LENGTH, WIDTH, DEPTH, MOMENT OF INERTIA, XMOE, & G OF MEMBERS
C
   10 TDMOE = (NTD) * (TDBE)
      BDMOE = (NBD) * (BDBE)
      TDBW = 0.0D+0
      DO 20 I = 1,NTDW
        TDBW = TDBW + TDW(I,1)*TDW(I,2)
   20 CONTINUE
      BDBW = 0.0D+0
      DO 30 I = 1,NBDW
        BDBW = BDBW + BDW(I,1)*BDW(I,2)
   30 CONTINUE
      EL(1) = BLKT
      W(1) = PLENGT
      T(1) = BLKT
      XMOE(1) = BMOE
      EL(2) = BLKLOC(2,NTSB) - .5*(PWIDTH)
      W(2) = TDBW
      T(2) = TDBT
      XMOE(2) = TDMOE
C  3 SUPPORT CONDITIONS: 1- INSIDE BLOCK, 2 - UNDER BLOCK, 3 - UNDER
C  IWINGED MEMBER.
      IF (ISUPC.EQ.1) THEN
        EL(3) = .5*(RADSPA)
        EL(15) = BLKLOC(2,NTSB) - EL(3) -.5*(PWIDTH)
      ELSE
        EL(3) = BLKLOC(2,NTSB) - 1.0 - .5*(PWIDTH)
        EL(15) = 1.0
```

```
      END IF
      W(3) = BDBW
      T(3) = BDBT
      XMOE(3) = BDMOE
      W(15) = W(3)
      T(15) = T(3)
      XMOE(15) = BDMOE
      L = 4
      DO 40 I = 0,8
        IF (I.EQ.7.OR.I.EQ.8) THEN
          EL(L + I) = BLKT
          W(L + I) = W(1)
          T(L + I) = T(1)
          XMOE(L + I) = BMOE
        ELSE IF (I.EQ.2.OR.I.EQ.3.OR.I.EQ.4) THEN
          EL(L + I) = 0
        ELSE
          EL(L + I) = .5*(BLKW(1))
          IF (I.EQ.0.OR.I.EQ.1) THEN
            W(L + I) = W(2)
            T(L + I) = T(2)
            XMOE(L + I) = XMOE(2)
          ELSE
            W(L + I) = W(1)
            T(L + I) = T(1)
            XMOE(L + I) = BMOE
          END IF
        END IF
   40 CONTINUE
      EL(13) = BLKW(1)
      W(13) = W(1)
      T(13) = T(1)
      XMOE(13) = 10.0D + 06
      EL(14) = 0.0D + 0
      EL(16) = 0.0D + 0
      IF (ISUPC.EQ.2) THEN
        EL(17) = (.5D0*RADSPA) - (BLKLOC(2,NTSB) - (.5D0*PWIDTH))
        EL(18) = BLKW(1) - EL(17)
      ELSE
        EL(18) = .5*(BLKW(1))
        EL(17) = .5*(BLKW(1))
      END IF
      IF (EL(17).LT.1) THEN
        EL(17) = 1.0D + 0
      END IF
      IF (EL(18).LT.1) THEN
        EL(18) = 1.0D + 0
      END IF
      W(17) = W(3)
      T(17) = T(3)
      XMOE(17) = XMOE(3)
      W(18) = W(3)
      T(18) = T(3)
      XMOE(18) = XMOE(3)
      IF (IWING.EQ.1) THEN
        IF (ISUPC.EQ.3) THEN
          EL(20) = (0.5*PWIDTH) - (.5*RADSPA)
          EL(19) = WINGL - EL(20)
        ELSE
          EL(19) = .5*(WINGL)
          EL(20) = EL(19)
        END IF
        IF (EL(19).LT.1) THEN
          EL(19) = 1.0
        END IF
        IF (EL(20).LT.1) THEN
          EL(20) = 1.0
        END IF
        XMOE(19) = TDMOE
        XMOE(20) = TDMOE
      ELSE
        EL(19) = 1.0
        EL(20) = 1.0
        XMOE(19) = 1.0D0
        XMOE(20) = 1.0D0
      END IF
      W(19) = W(2)
      T(19) = T(2)
      W(20) = W(2)
```

```
      T(20) = T(2)
C
      IF (NBLK.EQ.4) THEN
        XMOE(1) = 1.0D00
      END IF
C
      IF (NBLK.EQ.6) THEN
        IF (NBLKPS.EQ.3) THEN
          XMOE(1) = 1.0D00
        END IF
      END IF
C
C  FOR STACK SUPPORT CONDITION - AN EXTRA ELEMENT IS ADDED TO THE TOP
C  DECK (#21), AND ELEMENTS 2, 3, 15, AND 21 ARE ALL GIVEN SAME LENGTH.
C  DEFLECTIONS ARE CHECKED AT JOINTS 3 AND 17 FOR THIS SUPPORT MODE.
C
      IF (ISTAC.EQ.1) THEN
        EL(2) = 0.5*EL(2)
        EL(3) = EL(2)
        EL(15) = EL(2)
        EL(21) = EL(2)
        XMOE(21) = TDMOE
        W(21) = TDBW
        T(21) = TDBT
      END IF
C
      DO 50 I = 1,NE
        IF (EI (I).NE.0.0D00) THEN
          A(I) = W(I) * T(I)
          EI(I) = (W(I)*T(I)**3)/12.0D00
          S(I) = (EI(I) * 2.0D00)/T(I)
          XMG(I) = XMOE(I)/16.0D00
        END IF
   50 CONTINUE
C
C   COMPUTING THE DIRECTION COSINES FOR THE ELEMENTS
C
      C1(1) = 0.0D+0
      C2(1) = -1.0D+0
      DO 60 I = 2,5
        C1(I) = 1.0D+0
        C2(I) = 0.0D+0
   60 CONTINUE
      C1(6) = 0.0D+0
      C2(6) = 0.0D+0
      C1(7) = 0.0D+0
      C2(7) = 0.0D+0
      C1(8) = 0.0D+0
      C2(8) = 0.0D+0
      C1(9) = 1.0D+0
      C2(9) = 0.0D+0
      C1(10) = 1.0D+0
      C2(10) = 0.0D+0
      C1(11) = 0.0D+0
      C2(11) = -1.0D+0
      C1(12) = 0.0D+0
      C2(12) = -1.0D+0
      C1(13) = 1.0D+0
      C2(13) = 0.0D+0
      C1(14) = 0.0D+0
      C2(14) = 0.0D+0
      DO 70 I = 15,20
        IF (I.EQ.16) THEN
          C1(I) = 0.0D+0
          C2(I) = 0.0D+0
        ELSE IF (I.EQ.18) THEN
          C1(I) = -1.0D+0
          C2(I) = 0.0D+0
        ELSE
          C1(I) = 1.0D+0
          C2(I) = 0.0D+0
        END IF
   70 CONTINUE
      C1(21) = -1.0D00
      C2(21) = 0.0D00
      RETURN
      END
**************************************************************
*          SUBROUTINE UDL                                    *
**************************************************************
```

```
C
C  THIS ROUTINE CALCULATES THE FIXED END FORCES FOR EACH OF THE
C  LOADED MEMBERS FOR THE UNIFORMLY DISTRIBUTED LOAD CASE (BOTH
C  FULL AND PARTIAL UDL),AND THEN CALCULATES THE EQUIVALENT JOINT
C  FORCE VECTOR :
C
      SUBROUTINE UDL(XTD,XLOAD,ISUPC,IWING,EL,LTYPE,F,X2,Q,TDBW,ISTAC,
     $           NEQ,NBLK,NBLKPS)
      DOUBLE PRECISION F(6,*),EL(*),Q(*),XLOAD,XTD,X2,TDBW
C
C FIRSTLY, FOR THE FULL UDL:
C
      XLOAD = XLOAD / TDBW
      IF (LTYPE.EQ.1) THEN
        X2 = 0.0D+0
        F(2,2) = XLOAD*EL(2)/2.0D0
        F(5,2) = F(2,2)
        F(3,2) = XLOAD*(EL(2)**2)/12.0D0
        F(6,2) = -F(3,2)
        F(2,4) = XLOAD*EL(4)/2.0D0
        F(5,4) = F(2,4)
        F(3,4) = XLOAD*(EL(4)**2)/12.0D0
        F(6,4) = -F(3,4)
        F(2,5) = XLOAD*EL(5)/2.0D0
        F(5,5) = F(2,5)
        F(3,5) = XLOAD*(EL(5)**2)/12.0D0
        F(6,5) = -F(3,5)
C        WRITE(6,*) '2',F(2,2),F(3,2),F(5,2),F(6,2)
C        WRITE(6,*) '4',F(2,4),F(3,4),F(5,4),F(6,4)
C        WRITE(6,*) '5',F(2,5),F(3,5),F(5,5),F(6,5)
        IF (ISUPC.EQ.1) THEN
          Q(1) = -F(2,2)
          Q(5) = -F(5,2) - F(2,4)
          Q(6) = -F(6,2) - F(3,4)
          Q(17) = -F(5,4) - F(2,5)
          Q(18) = -F(6,4) - F(3,5)
          Q(23) = -F(5,5)
          Q(24) = -F(6,5)
        ELSE
          Q(1) = -F(2,2)
          Q(7) = -F(5,2) - F(2,4)
          Q(8) = -F(6,2) - F(3,4)
          Q(19) = -F(5,4) - F(2,5)
          Q(20) = -F(6,4) - F(3,5)
          Q(25) = -F(5,5)
          Q(26) = -F(6,5)
        END IF
        IF (IWING.EQ.1) THEN
          IF (ISUPC.EQ.3) THEN
            F(5,19) = XLOAD*EL(19)/2.0D0
            F(3,19) = XLOAD*(EL(19)**2)/12.0D0
            F(2,19) = F(5,19)
            F(6,19) = -F(3,19)
            F(2,20) = XLOAD*EL(20)/2.0D0
            F(5,20) = F(2,20)
            F(3,20) = XLOAD*(EL(20)**2)/12.0D0
            F(6,20) = -F(3,20)
C            WRITE(6,*) '19',F(2,19),F(3,19),F(5,19),F(6,19)
C            WRITE(6,*) '20',F(2,20),F(3,20),F(5,20),F(6,20)
            Q(25) = -F(5,5) - F(2,19)
            Q(26) = -F(6,5) - F(3,19)
            Q(40) = -F(6,19) - F(3,20)
            Q(42) = -F(5,20)
            Q(43) = -F(6,20)
          ELSE
            F(2,19) = XLOAD*EL(19)/2.0D0
            F(5,19) = F(2,19)
            F(3,19) = XLOAD*(EL(19)**2)/12.0D0
            F(6,19) = -F(3,19)
            F(2,20) = XLOAD*EL(20)/2.0D0
            F(5,20) = F(2,20)
            F(3,20) = XLOAD*(EL(20)**2)/12.0D0
            F(6,20) = -F(3,20)
C            WRITE(6,*) '19',F(2,19),F(3,19),F(5,19),F(6,19)
C            WRITE(6,*) '20',F(2,20),F(3,20),F(5,20),F(6,20)
            IF (ISUPC.EQ.1) THEN
              Q(23) = -F(5,5) - F(2,19)
              Q(24) = -F(6,5) - F(3,19)
              Q(38) = -F(5,19) - F(2,20)
              Q(39) = -F(6,19) - F(3,20)
```

```
       Q(41) = -F(5,20)
       Q(42) = -F(6,20)
      ELSE
       Q(25) = -F(5,5) - F(2,19)
       Q(26) = -F(6,5) - F(3,19)
       Q(38) = -F(5,19) - F(2,20)
       Q(39) = -F(6,19) - F(3,20)
       Q(41) = -F(5,20)
       Q(42) = -F(6,20)
      END IF
     END IF
    END IF
   ELSE
C
C SECONDLY THE PARTIAL UDL:
C
    IF (XTD.LT.EL(5)) THEN
     X5 = XTD
     X2 = 0.0D+0
     X4 = 0.0D+0
    ELSE IF (XTD.LT.(EL(4) + EL(5))) THEN
     X4 = XTD - EL(5)
     X2 = 0.0D+0
     X5 = EL(5)
    ELSE
     X2 = XTD - EL(4) - EL(5)
     X4 = EL(4)
     X5 = EL(5)
    END IF
C
    F(5,2) = ( + XLOAD/(2.0D0*EL(2)**3))*(2.0D0*EL(2)**3*(EL(2)-X2)
  $ -2.0D0*EL(2)*(EL(2)**3-X2**3) + (EL(2)**4 - X2**4))
    F(2,2) =  + XLOAD*(EL(2) - X2) - F(5,2)
    F(3,2) =( + XLOAD/(12.0D0*EL(2)**2))*(4.0D0*EL(2)*(EL(2)**3-X2**3)
  $ -3*(EL(2)**4-X2**4))
    F(6,2) = (-XLOAD/(12.0D0*EL(2)**2))*(6.0D0*EL(2)**2*(EL(2)**2
  $ -X2**2)-8.0D0*EL(2)*(EL(2)**3 -X2**3) + 3.0D0*(EL(2)**4 - X2**4))
    F(5,4) = ( + XLOAD/(2.0D0*EL(4)**3))*(2.0D0*EL(4)**3*(EL(4)-X4)
  $ -2.0D0*EL(4)*(EL(4)**3-X4**3) + (EL(4)**4 - X4**4))
    F(2,4) =  + XLOAD*(EL(4) - X4) - F(5,4)
    F(3,4) = (XLOAD/(12.0D0*EL(4)**2))*(4.0D0*EL(4)*(EL(4)**3-X4**3)
  $         -3.0D0*(EL(4)**4-X4**4))
    F(6,4) =(-XLOAD/(12.0D0*EL(4)**2))*(6.0D0*EL(4)**2*(EL(4)**2
  $ -X4**2) - 8.0D0*EL(4)*(EL(4)**3-X4**3) + 3.0D0*(EL(4)**4-X4**4))
    F(5,5) = ( + XLOAD/(2.0D0*EL(5)**3))*(2.0D0*EL(5)**3*(EL(5)-X5)
  $         -2.0D0*EL(5)*(EL(5)**3-X5**3) + (EL(5)**4 - X5**4))
    F(2,5) = XLOAD*(EL(5) - X5) - F(5,5)
    F(3,5) = (XLOAD/(12.0D0*EL(5)**2))*(4.0D0*EL(5)*(EL(5)**3-X5**3)
  $         -3.0D0*(EL(5)**4 - X5**4))
    F(6,5) = (-XLOAD/(12.0D0*EL(5)**2))*(6.0D0*EL(5)**2*(EL(5)**2
  $   -X5**2)-8.0D0*EL(5)*(EL(5)**3-X5**3) + 3.0D0*(EL(5)**4-X5**4))
C
    IF (XTD.LT.EL(5)) THEN
     IF (ISUPC.EQ.1) THEN
      Q(1) = -F(2,2)
      Q(5) = -F(5,2) - F(2,4)
      Q(6) = -F(6,2) - F(3,4)
      Q(17) = -F(5,4) - F(2,5)
      Q(18) = -F(6,4) - F(3,5)
      Q(23) = -F(5,5)
      Q(24) = -F(6,5)
     ELSE
      Q(1) = -F(2,2)
      Q(7) = -F(5,2) - F(2,4)
      Q(8) = -F(6,2) - F(3,4)
      Q(19) = -F(5,4) - F(2,5)
      Q(20) = -F(6,4) - F(3,5)
      Q(25) = -F(5,5)
      Q(26) = -F(6,5)
     END IF
    ELSE IF (XTD.LT.(EL(4) + EL(5))) THEN
     IF (ISUPC.EQ.1) THEN
      Q(1) = -F(2,2)
      Q(5) = -F(5,2) - F(2,4)
      Q(6) = -F(6,2) - F(3,4)
      Q(17) = -F(5,4)
      Q(18) = -F(6,4)
     ELSE
      Q(1) = -F(2,2)
      Q(7) = -F(5,2) - F(2,4)
```

```
          Q(8)  =  -F(6,2) - F(3,4)
          Q(19)  =  -F(5,4)
          Q(20)  =  -F(6,4)
        END IF
      ELSE
        IF (ISUPC.EQ.1) THEN
          Q(1)  =  -F(2,2)
          Q(5)  =  -F(5,2)
          Q(6)  =  -F(6,2)
        ELSE
          Q(1)  =  -F(2,2)
          Q(7)  =  -F(5,2)
          Q(8)  =  -F(6,2)
        END IF
      END IF
    END IF
C
C  FOR STACK SUPPORT CONDITION:
C
    IF (ISTAC.EQ.1) THEN
      F(2,21)  =  XLOAD*EL(21)/2.0D0
      F(5,21)=  F(2,21)
      F(3,21)  =  XLOAD*(EL(21)**2)/12.0D0
      F(6,21)  =  -F(3,21)
      DO 20 K  =  1,NEQ
        Q(K)  =  0.0D00
  20  CONTINUE
      IF (NBLK.EQ.4) THEN
        Q(1)  =  -F(2,2)
        Q(10)  =  -F(5,2) - F(2,21)
        Q(11)  =  -F(6,2) - F(3,21)
        GOTO 30
      END IF
      IF (NBLK.EQ.6) THEN
        IF (NBLKPS.EQ.3) THEN
          Q(1)  =  -F(2,2)
          Q(10)  =  -F(5,2) - F(2,21)
          Q(11)  =  -F(6,2) - F(3,21)
        END IF
        GOTO 30
      END IF
      Q(8)  =  -F(5,2) - F(2,21)
      Q(9)  =  -F(6,2) - F(3,21)
  30  CONTINUE
    END IF
C
C
C       WRITE(6,*) '2',F(2,2),F(3,2),F(5,2),F(6,2)
C       WRITE(6,*) '4',F(2,4),F(3,4),F(5,4),F(6,4)
C       WRITE(6,*) '5',F(2,5),F(3,5),F(5,5),F(6,5)
C       WRITE(6,*) '21',F(2,21),F(3,21),F(5,21),F(6,21)
    RETURN
    END
****************************************************************
*                SUBROUTINE LINE                               *
****************************************************************
    SUBROUTINE LINE(CL,SL,LTYPE,EL,XTD,PWIDTH,F,Q,ISUPC,P1,P2)
    DOUBLE PRECISION F(6,*),EL(*),Q(*),CL,SL,XTD,PWIDTH,P1,P2
C
C  THIS ROUTINE CALCULATES THE FIXED END FORCES FOR THE LINE LOAD CASE.
C  MAX. # OF LINE XLOADS IS 3 - IN ALL CASES THEY ARE CONSIDERED TO BE
C  SYMMETRICALLY PLACED ABOUT THE CENTER LINE. THE SECOND LOAD CAN ONLY
C  BE PLACED ALONG THE SPAN OF MEMBER # 2.
C  THE EQUIVALENT JOINT FORCE VECTOR IS ALSO CALCULATED.
C
    X  =  (.5D+0*PWIDTH) - XTD
    IF (LTYPE.EQ.3) THEN
      P1  = 0.0D+0
      P2  =  CL/2
      Q(1)  =  -P2
    ELSE IF (LTYPE.EQ.4) THEN
      P1  =  SL
      P2  =  0.0D+0
    ELSE
      P1  =  SL
      P2  =  CL/2
    END IF
C
    F(2,2)  =  (P1/EL(2)**3)*(EL(2)-X)**2*(EL(2)+(2.0D+0*X))
    F(3,2)  =  P1*X*((EL(2)-X)/EL(2))**2
```

```
      F(5,2) = P1*(X**2)*(3.0D + 0*EL(2)-2.0D + 0*X)/(EL(2)**3)
      F(6,2) = -P1*(EL(2)-X)*(X/EL(2))**2
C
      IF (LTYPE.EQ.4.OR.LTYPE.EQ.5) THEN
        IF (ISUPC.EQ.1) THEN
          Q(1) = -F(2,2)
          Q(5) = -F(5,2)
          Q(6) = -F(6,2)
        ELSE
          Q(1) = -F(2,2)
          Q(7) = -F(5,2)
          Q(8) = -F(6,2)
        END IF
        IF (LTYPE.EQ.5) THEN
          IF (ISUPC.EQ.1) THEN
            Q(1) = -F(2,2) - P2
          ELSE
            Q(1) = -F(2,2) - P2
          END IF
        END IF
      END IF
C        WRITE(6,*) '2',F(2,2),F(3,2),F(5,2),F(6,2)
C        WRITE(6,*) '4',F(2,4),F(3,4),F(5,4),F(6,4)
C        WRITE(6,*) '5',F(2,5),F(3,5),F(5,5),F(6,5)
      RETURN
      END
**************************************************************
*                        MELEMS                              *
**************************************************************
      SUBROUTINE MELEMS(GAM3,GAM4,GAM5,GAM6,XMOE,EI,A,EL,C1,
     $                  C2,N,G,W,T,NBLK,NBLKPS)
      DOUBLE PRECISION G(7),EL(*),EI(*),A(*),C1(*),C2(*),W(*),T(*)
     $                 ,XMOE(*),GAM3,GAM4,GAM5,GAM6
C
C  FOR ELEMENT N, COMPUTE THE GLOBAL COEFFICIENTS G(1) - G(7):
C
      DO 10 J = 1,7
        G(J) = 0.0D + 0
   10 CONTINUE
C
      IF (N.EQ.16) THEN
        G(1) = 10.0D + 06
        G(3) = 10.0D + 06
        G(6) = GAM3
        G(7) = -GAM3
      ELSE IF (N.EQ.6.OR.N.EQ.7) THEN
        IF (N.EQ.6) THEN
          G(1) = 0.00D00
          G(3) = 0.00D00
          G(6) = GAM3
          G(7) = -GAM3
        ELSE
          G(3) = GAM4
        END IF
      ELSE IF (N.EQ.8.OR.N.EQ.14) THEN
        IF(N.EQ.14) THEN
          G(1) = 10.0D + 06
          G(3) = GAM5
        ELSE
          G(3) = GAM6
        END IF
      ELSE
        ALPHA = (XMOE(N) * EI(N))/EL(N)**3
        B = (A(N) * EL(N)**2)/EI(N)
        G(1) = ALPHA*((B*C1(N)**2) + (12.0D0*C2(N)**2))
        G(2) = ALPHA*C1(N)*C2(N)*(B-12.0D0)
        G(3) = ALPHA*((B*C2(N)**2) + (12.0D0*C1(N)**2))
        G(4) = -ALPHA*6.0D0*EL(N)*C2(N)
        G(5) = ALPHA*6.0D0*EL(N)*C1(N)
        G(6) = ALPHA*4.0D0*EL(N)**2
        G(7) = ALPHA*2.0D0*EL(N)**2
      END IF
C
C
      RETURN
      END
**************************************************************
*                        ASSEMS                              *
**************************************************************
      SUBROUTINE ASSEMS(MCODE,N,MBD,SS,G,MXNEQ)
```

```
      INTEGER MCODE(6,*),INDEX(6,6)
C      DIMENSION SS(MXNEQ,MXNEQ),G(7)
      DOUBLE PRECISION SS(100,100),G(7)
C
C  INITIALISE INDEX: ASSIGN STIFFNESS COEFFICIENTS,G(L), OF ELEMENT N
C  INTO THE SYSTEM STIFFNESS BAND MATRIX, SS, BY INDEX AND MCODE.
C
      DATA INDEX/1,2,4,-1,-2,4.2,3,5,-2,-3.5,4,5,6,-4,-5,7,-1,-2
     $          ,-4,1,2,-4,-2,-3,-5,2,3,-5,4,5,7,-4,-5,6/
      DO 20 JE  =  1,6
        J  =  MCODE(JE,N)
        IF (J.NE.0) THEN
          DO 10 IE  =  1,JE
            I  =  MCODE(IE,N)
            IF (I.NE.0) THEN
              K  =  I-J  +  MBD  +  1
              L  =  INDEX(IE,JE)
              IF (L.GT.0) THEN
                SS(K,J)  =  SS(K,J)  +  G(L)
              ELSE
                SS(K,J)  =  SS(K,J) - G(-L)
              END IF
            END IF
 10     CONTINUE
        END IF
 20   CONTINUE
      RETURN
      END
*****************************************************************
*                        SOLVE                                  *
*****************************************************************
C
      SUBROUTINE SOLVE(SS,Q,NEQ,MBD,MXNEQ)
C
C           THIS SUBROUTINE CALLS SPBFA AND SPBSL FOR THE FIRST XLOAD
C      CONDITION, LC  =  1. FOR SUBSEQUENT XLOAD CONDITIONS, LC GREATER
C      THAN 1, IT CALLS SPBSL.
C
      DOUBLE PRECISION SS(100,100),Q(*)
C      DIMENSION SS(MXNEQ,MXNEQ),Q(*)
C      IF (LC.EQ.1)THEN
        CALL SPBFA(SS,MXNEQ,NEQ,MBD,INFO)
        IF (INFO.NE.0)THEN
          PRINT *,'SINGULARITY...'
          STOP
        END IF
C      END IF
      CALL SPBSL(SS,MXNEQ,NEQ,MBD,Q)
      RETURN
      END
C     THIS IS THE LINPACK SOLUTION ALGORITHM WITH COMMENTS.
C----------------------------------------------------------------
      SUBROUTINE SPBFA(ABD,LDA,N,M,INFO)
      INTEGER LDA,N,M,INFO
      DOUBLE PRECISION ABD(100,*)
C      REAL ABD(LDA,*)
C
C      SPBFA FACTORS A REAL SYMMETRIC POSITIVE DEFINITE
C      MATRIX STORED IN BAND FORM.
C
C      SPBFA IS USUALLY CALLED BY SPBCO, BUT IT CAN BE CALLED
C      DIRECTLY WITH A SAVING IN TIME IF  RCOND  IS NOT NEEDED.
C
C      ON ENTRY
C
C         ABD     REAL(LDA, N)
C                 THE MATRIX TO BE FACTORED.  THE COLUMNS OF THE UPPER
C                 TRIANGLE ARE STORED IN THE COLUMNS OF ABD AND THE
C                 DIAGONALS OF THE UPPER TRIANGLE ARE STORED IN THE
C                 ROWS OF ABD .  SEE THE COMMENTS BELOW FOR DETAILS.
C
C         LDA     INTEGER
C                 THE LEADING DIMENSION OF THE ARRAY  ABD .
C                 LDA MUST BE .GE. M  +  1 .
C
C         N       INTEGER
C                 THE ORDER OF THE MATRIX  A .
C
C         M       INTEGER
C                 THE NUMBER OF DIAGONALS ABOVE THE MAIN DIAGONAL.
```

```
C          0 .LE. M .LT. N .
C
C   ON RETURN
C
C     ABD    AN UPPER TRIANGULAR MATRIX  R , STORED IN BAND
C            FORM, SO THAT  A = TRANS(R)*R .
C
C     INFO   INTEGER
C            = 0  FOR NORMAL RETURN.
C            = K  IF THE LEADING MINOR OF ORDER  K  IS NOT
C               POSITIVE DEFINITE.
C
C   BAND STORAGE
C
C       IF  A  IS A SYMMETRIC POSITIVE DEFINITE BAND MATRIX,
C       THE FOLLOIWING PROGRAM SEGMENT WILL SET UP THE INPUT.
C
C              M  = (BAND WIDTH ABOVE DIAGONAL)
C              DO 20 J = 1, N
C                I1  = MAX0(1, J-M)
C                DO 10 I = I1, J
C                  K  = I-J + M + 1
C                  ABD(K,J) = A(I,J)
C           10   CONTINUE
C           20 CONTINUE
C
C   LINPACK.  THIS VERSION DATED 08/14/78 .
C   CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LAB.
C
C   SUBROUTINES AND FUNCTIONS
C
C   BLAS SDOT
C   FORTRAN MAX0,SQRT
C
C   INTERNAL VARIABLES
C
      DOUBLE PRECISION SDOT,T
      DOUBLE PRECISION S
      INTEGER IK,J,JK,K,MU
C     BEGIN BLOCK WITH ...EXITS TO 40
C
C
         DO 30 J = 1, N
            INFO = J
            S = 0.0D+0
            IK = M + 1
            JK = MAX0(J-M,1)
            MU = MAX0(M + 2-J,1)
            IF (M .LT. MU) GO TO 20
            DO 10 K = MU, M
               T = ABD(K,J) - SDOT(K-MU,ABD(IK,JK),1,ABD(MU,J),1)
               T = T/ABD(M + 1,JK)
               ABD(K,J) = T
               S = S + T*T
               IK = IK - 1
               JK = JK + 1
   10       CONTINUE
   20       CONTINUE
            S = ABD(M + 1,J) - S
C     ......EXIT
            IF (S .LE. 0.0D+0) GO TO 40
            ABD(M + 1,J) = SQRT(S)
   30    CONTINUE
         INFO = 0
   40 CONTINUE
      RETURN
      END
C-----------------------------------------------------
      SUBROUTINE SPBSL(ABD,LDA,N,M,B)
      INTEGER LDA,N,M
      DOUBLE PRECISION ABD(100,*),B(*)
C     REAL ABD(LDA,*),B(*)
C
C   SPBSL SOLVES THE REAL SYMMETRIC POSITIVE DEFINITE
C   BAND SYSTEM  A*X = B
C   USING THE FACTORS COMPUTED BY SPBCO OR SPBFA.
C
C   ON ENTRY
C
C     ABD    REAL(LDA, N)
```

```
C            THE OUTPUT FROM SPBCO OR SPBFA.
C
C      LDA    INTEGER
C            THE LEADING DIMENSION OF THE ARRAY  ABD .
C
C      N      INTEGER
C            THE ORDER OF THE MATRIX  A .
C
C      M      INTEGER
C            THE NUMBER OF DIAGONALS ABOVE THE MAIN DIAGONAL.
C
C      B      REAL(N)
C            THE RIGHT HAND SIDE VECTOR.
C
C    ON RETURN
C
C      B      THE SOLUTION VECTOR  X .
C
C    ERROR CONDITION
C
C      A DIVISION BY ZERO WILL OCCUR IF THE INPUT FACTOR CONTAINS
C      A ZERO ON THE DIAGONAL.  TECHNICALLY THIS INDICATES
C      SINGULARITY BUT IT IS USUALLY CAUSED BY IMPROPER SUBROUTINE
C      ARGUMENTS.  IT WILL NOT OCCUR IF THE SUBROUTINES ARE CALLED
C      CORRECTLY AND  INFO .EQ. 0 .
C
C    TO COMPUTE  INVERSE(A) * C  WHERE  C  IS A MATRIX
C    WITH  P  COLUMNS
C          CALL SPBCO(ABD,LDA,N,RCOND,Z,INFO)
C          IF (RCOND IS TOO SMALL .OR. INFO .NE. 0) GO TO ...
C          DO 10 J = 1, P
C            CALL SPBSL(ABD,LDA,N,C(1,J))
C       10 CONTINUE
C
C    LINPACK.  THIS VERSION DATED 08/14/78 .
C    CLEVE MOLER, UNIVERSITY OF NEW MEXICO, ARGONNE NATIONAL LAB.
C
C    SUBROUTINES AND FUNCTIONS
C
C    BLAS SAXPY,SDOT
C    FORTRAN MIN0
C
C    INTERNAL VARIABLES
C
      DOUBLE PRECISION SDOT,T
      INTEGER K,KB,LA,LB,LM
C
C     SOLVE TRANS(R)*Y = B
C
      DO 10 K = 1, N
         LM = MIN0(K-1,M)
         LA = M + 1 - LM
         LB = K - LM
         T = SDOT(LM,ABD(LA,K),1,B(LB),1)
         B(K) = (B(K) - T)/ABD(M + 1,K)
   10 CONTINUE
C
C     SOLVE R*X = Y
C
      DO 20 KB = 1, N
         K = N + 1 - KB
         LM = MIN0(K-1,M)
         LA = M + 1 - LM
         LB = K - LM
         B(K) = B(K)/ABD(M + 1,K)
         T = -B(K)
         CALL SAXPY(LM,T,ABD(LA,K),1,B(LB),1)
   20 CONTINUE
      RETURN
      END
C-----------------------------------------------------------
      SUBROUTINE SAXPY(N,SA,SX,INCX,SY,INCY)
C
C     CONSTANT TIMES A VECTOR PLUS A VECTOR.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      DOUBLE PRECISION SX(*),SY(*),SA
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
```

```
      IF(N.LE.0)RETURN
      IF (SA .EQ. 0.0D+0) RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        SY(IY) = SY(IY) + SA*SX(IX)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,4)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        SY(I) = SY(I) + SA*SX(I)
   30 CONTINUE
      IF( N .LT. 4 ) RETURN
   40 MP1 = M + 1
      DO 50 I = MP1,N,4
        SY(I) = SY(I) + SA*SX(I)
        SY(I + 1) = SY(I + 1) + SA*SX(I + 1)
        SY(I + 2) = SY(I + 2) + SA*SX(I + 2)
        SY(I + 3) = SY(I + 3) + SA*SX(I + 3)
   50 CONTINUE
      RETURN
      END
C-----------------------------------------------------------
      DOUBLE PRECISION FUNCTION SDOT(N,SX,INCX,SY,INCY)
C
C     FORMS THE DOT PRODUCT OF TWO VECTORS.
C     USES UNROLLED LOOPS FOR INCREMENTS EQUAL TO ONE.
C     JACK DONGARRA, LINPACK, 3/11/78.
C
      DOUBLE PRECISION SX(*),SY(*),STEMP
      INTEGER I,INCX,INCY,IX,IY,M,MP1,N
C
      SDOT = 0.0D+0
      STEMP = 0.0D+0
      IF(N.LE.0)RETURN
      IF(INCX.EQ.1.AND.INCY.EQ.1)GO TO 20
C
C        CODE FOR UNEQUAL INCREMENTS OR EQUAL INCREMENTS
C          NOT EQUAL TO 1
C
      IX = 1
      IY = 1
      IF(INCX.LT.0)IX = (-N+1)*INCX + 1
      IF(INCY.LT.0)IY = (-N+1)*INCY + 1
      DO 10 I = 1,N
        STEMP = STEMP + SX(IX)*SY(IY)
        IX = IX + INCX
        IY = IY + INCY
   10 CONTINUE
      SDOT = STEMP
      RETURN
C
C        CODE FOR BOTH INCREMENTS EQUAL TO 1
C
C
C        CLEAN-UP LOOP
C
   20 M = MOD(N,5)
      IF( M .EQ. 0 ) GO TO 40
      DO 30 I = 1,M
        STEMP = STEMP + SX(I)*SY(I)
   30 CONTINUE
      IF( N .LT. 5 ) GO TO 60
   40 MP1 = M + 1
```

```
      DO 50 I = MP1,N,5
        STEMP = STEMP + SX(I)*SY(I) + SX(I + 1)*SY(I + 1) +
     *  SX(I + 2)*SY(I + 2) + SX(I + 3)*SY(I + 3) + SX(I + 4)*SY(I + 4)
   50 CONTINUE
   60 SDOT = STEMP
      RETURN
      END
************************************************************
*                       MELEMF                             *
************************************************************
      SUBROUTINE MELEMF(MCODE,Q,C1,C2,XMOE,I,EL,A,F,FL,DL,EI,ALPHA,BETA,
     $                 FLO,YMAX,NE,MINC,DISPL,MEM,JT)
      INTEGER MCODE(6,*), MINC(2,*)
      DOUBLE PRECISION Q(*),EI(*),EL(*),DL(6),C1(*),C2(*),XMOE(*),D(6),
     $  DISPL(3,*),A(*),FL(6,*),F(6,*),ALPHA(*),BETA(*),FLO(6),Y,YMAX
C
C  COMPUTES THE LOCAL FORCES OF ELEMENT I, LF(6,I);DETERMINES THE GLOBAL
C  ELEMENT DISPLACEMENTS, D(6), FROM THE JOINT DISPLACEMENT VECTOR , Q,
C  VIA THE MCODE; COMPUTES THE LOCAL FORCES AT THE A-END OF ELEMENT I
C  AND THEN THE B-END BY EQUILIBRIUM.
C
      DO 10 L = 1,6
        K = MCODE(L,I)
        IF (K.NE.0) THEN
          D(L) = Q(K)
        ELSE
          D(L) = 0.0D+0
        END IF
   10 CONTINUE
      DISPL(1,MINC(1,I)) = D(1)
      DISPL(2,MINC(1,I)) = D(2)
      DISPL(3,MINC(1,I)) = D(3)
      DISPL(1,MINC(2,I)) = D(4)
      DISPL(2,MINC(2,I)) = D(5)
      DISPL(3,MINC(2,I)) = D(6)
C
      Y = ABS(D(2))
      IF (Y.GT.YMAX) THEN
        YMAX = Y
        MEM = I
        JT = MINC(1,I)
      END IF
C
C        WRITE(6,*) I,'A-END',D(1),D(2),D(3)
C        WRITE(6,*) I,'B-END',D(4),D(5),D(6)
C
      DL(1) =  C1(I)*D(1) + C2(I)*D(2)
      DL(2) = -C2(I)*D(1) + C1(I)*D(2)
      DL(3) = D(3)
      DL(4) =  C1(I)*D(4) + C2(I)*D(5)
      DL(5) = -C2(I)*D(4) + C1(I)*D(5)
      DL(6) = D(6)
C
      ALPHA(I) = (XMOE(I) * EI(I)) EL(I)**3
      BETA(I) = (A(I) * EL(I)**2) EI(I)
C
      FLO(1) = ALPHA(I)*BETA(I)*(DL(1) - DL(4))
      FLO(2) = ALPHA(I)*(12.0D0*(DL(2)-DL(5)) + 6.0D0*EL(I)*(DL(3) + DL(6)))
      FLO(3) =  ALPHA(I)*(6.0D0*EL(I)*(DL(2)-DL(5)) + 2.0D0*EL(I)**2*
     $       (2.0D0*DL(3) + DL(6)))
      FLO(4) = -FLO(1)
      FLO(5) = -FLO(2)
      FLO(6) = (FLO(2)*EL(I)) - FLO(3)
C
      FL(1,I) = F(1,I) + FLO(1)
      FL(2,I) = F(2,I) + FLO(2)
      FL(3,I) = F(3,I) + FLO(3)
      FL(4,I) = F(4,I) - FLO(1)
      FL(5,I) = F(5,I) - FLO(2)
      FL(6,I) = F(6,I) + (FLO(2)*EL(I))-FLO(3)
C      WRITE(6,*) FL(1,I)
C      WRITE(6,*) FL(2,I)
C      WRITE(6,*) FL(3,I)
C      WRITE(6,*) FL(4,I)
C      WRITE(6,*) FL(5,I)
C      WRITE(6,*) FL(6,I)
C
      RETURN
      END
************************************************************
```

```
*                  MSTRES                              *
*****************************************************************
C
C  CALCULATES THE MEMBER STRESSES FOR ALL ELEMENTS OTHER THAN SPRING
C  AND BLOCK MEMBERS. ALSO, CHECKS BENDING STRESS IN MEMBER 2.
C
      SUBROUTINE MSTRES(X2,I,LTYPE,XLOAD,FL,A,S,ISUPC,DL,ALPHA,BETA,ST,
     $                STA,STB,FLO,EL,XMOE,EI,CL,SL,P1,P2,PWIDTH,XTD,
     $                YS,STMAX,NE)
      DOUBLE PRECISION FL(6,*),FLO(6),A(*),S(*),DL(*),ALPHA(*),BETA(*)
     $          ,ST(*),STA(*),STB(*),EL(*),EI(*),XMOE(*),XLOAD,X2,CL,SL
     $          ,P1,P2,PWIDTH,XTD,XLOC,STMAX,XMEMB,YS,XMAX,XBMAX
C
C  MEMBER 2:- BENDING STRESS :
C
      IF (I.EQ.2) THEN
        IF (LTYPE.EQ.1.OR.LTYPE.EQ.2) THEN
          XMAX  =  X2  +  (FL(5,2)/XLOAD)
          XBMAX  =  -FL(6,2)  +  FL(5,2)*((XMAX + X2)/2.0D0)
          ST(2)  =  (ABS(FL(1,2)/A(2)))  +  (ABS(XBMAX/S(2)))
        ELSE IF (LTYPE.EQ.3) THEN
          XMAX  =  0.0D00
          XBMAX  =   P2*EL(2)/2.0D0
          ST(2)  =  (ABS(FL(1,2)/A(2)))  +  (ABS(XBMAX/S(2)))
        ELSE
          XMAX  =  (0.5*PWIDTH) - XTD
          XBMAX  =  -FL(3,2)  +  (FL(2,2)*XMAX)
          ST(2)  =  (ABS(FL(1,2)/A(2)))  +  (ABS(XBMAX/S(2)))
        END IF
C        WRITE(6,*) 'BENDING STRESS IN 2 = ',ST(2)
        IF (ST(2).GT.STMAX) THEN
          STMAX  =  ST(2)
          XLOC  =  XMAX
          XMEMB  =  2
        END IF
      END IF
C
      IF (I.EQ.2.OR.I.EQ.3.OR.I.EQ.4.OR.I.EQ.5.OR.I.EQ.15.OR.I.EQ.17
     $   .OR.I.EQ.18.OR.I.EQ.19.OR.I.EQ.20) THEN
C
C        MEMBER 3:- WHEN SUPPORT IS UNDER THE BOTTOM D.B
C        (I.E INSIDE BLOCK)
C
        IF (I.EQ.3.AND.ISUPC.EQ.1) THEN
          FLO(1)  =  ALPHA(3)*BETA(3)*(-DL(4))
          FLO(2)  =  (12.0D0*ALPHA(3)*DL(2)) + (6.0D0*ALPHA(3)*EL(3)*DL(6))
          FLO(3)  =  (6.0D0*ALPHA(3)*EL(3)*DL(2)) + ((XMOE(3)*4.0D0*EI(3))
     $            /EL(3))*DL(6)
          FLO(6)  =  -FLO(3)
        END IF
C
        STA(I)  =  (ABS(FLO(1)/A(I)))  +  (ABS(FLO(3)/S(I)))
        STB(I)  =  (ABS(FLO(1)/A(I)))  +  (ABS(FLO(6)/S(I)))
        WRITE(6,*) I,STA(I),STB(I)
        IF (STA(I).GT.STMAX) THEN
          STMAX  =  STA(I)
          XMAX  =  0.0D00
          XMEMB  =  I
        END IF
        IF (STB(I).GT.STMAX) THEN
          STMAX  =  STB(I)
          XMAX  =  EL(I)
          XMEMB  =  I
        END IF
      END IF
      IF (I.EQ.NE) THEN
        WRITE(6,*)
        WRITE(6,*) 'MAX. BENDING STRESS  =  ',STMAX
        WRITE(6,*) 'IN MEMBER #',XMEMB
        WRITE(6,*) 'AT (IN)',XLOC
      END IF
      RETURN
      END
```

# Appendix C:

## Subroutines for Stack condition

```
*****************************************************************
*                     STACKJ                        *
*****************************************************************
C  THIS SUBROUTINE MODIFIES THE JCODE FOR THE STRUCTURE TO ACCOUNT FOR
C  A STACKED CONDITION
C
      SUBROUTINE STACKJ(JN,NTD,JCODE,NBLK,NBLKPS)
      INTEGER JCODE(6,*),JN(*)
C
      DO 20 J = (JN(NTD) + 13),(JN(NTD) + 23)
        DO 10 L = 1,6
          JCODE(L,J) = 0
   10   CONTINUE
   20 CONTINUE
      DO 30 L = 1,6
        JCODE(L,JN(NTD) + 26) = 0
   30 CONTINUE
      IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
        DO 50 J = (JN(NTD) + 2),(JN(NTD) + 11)
          DO 40 L = 1,6
            JCODE(L,1) = 0
            JCODE(L,3) = 0
            IF (NBLK.EQ.9) THEN
              JCODE(L,2) = 0
              JCODE(L,4) = 0
            END IF
            JCODE(L,JN(NTD) + 27) = 0
            JCODE(L,J) = 0
   40     CONTINUE
   50   CONTINUE
      ELSE IF (NBLK.EQ.6) THEN
        IF (NBLKPS.EQ.2) THEN
          DO 80 J = (JN(NTD) + 2),(JN(NTD) + 11)
            DO 70 L = 1,6
              JCODE(L,JN(NTD) + 27) = 0
              JCODE(L,J) = 0
   70       CONTINUE
   80     CONTINUE
        END IF
      END IF
      RETURN
      END
*****************************************************************
*                    STACKL                         *
*****************************************************************
C
C  THIS SUBROUINE CALCULATES THE FIXED END FORCES FOR THE BOTTOM
C  DECKBOARDS (WHERE PRESENT) IN THE STACK CONDITION. THESE DECKBOARDS
C  ARE IDENTIFIED BY THE FLAG - 'ISL'.
C
      SUBROUTINE STACKL(ISL,W,EL,XLOAD,WL,MN,F,Q,XTD,XPWIDT,BLKW,NE,
     $             NBLK,NBLKPS,NTD)
      DOUBLE PRECISION EL(*),XLOAD,WL(*),F(12,*),Q(*),XTD,XPWIDT,DA,DB
     $             ,BLKW(2),W(*)
      INTEGER ISL(*),MN(*)
C
      DA = XTD
      DB = (0.5D0*XPWIDT) - DA - (0.5D0*BLKW(1))
      WRITE(6,*) 'DA = ',DA,' DB = ',DB
      Y1 = DB - DA
      Y2 = DB**2 - DA**2
      Y3 = DB**3 - DA**3
      Y4 = DB**4 - DA**4
      AREAB = 0.0D00
C
      DO 10 I = 1,NE
        IF (ISL(I).EQ.1) THEN
          AREAB = AREAB + (W(I)*EL(I))
        END IF
   10 CONTINUE
      IF (ISL(4).EQ.1) THEN
        WL(4) = (XLOAD*W(4))/AREAB
        WRITE(6,*) 'LOAD = ',WL(4)
        F(2,4) = ( + WL(4)/(2.0D0*EL(4)**3))*(2.0D0*EL(4)**3*Y1-2.0D0*EL(4)
     $          *Y3 + Y4)
        F(6,4) = ( + WL(4)/(12.0D0*EL(4)**2))*(6.0D0*EL(4)**2*Y2
     $           -8.0D0*EL(4)*Y3 + 3.0D0*Y4)
        F(8,4) =( + WL(4)/(2.0D0*EL(4)**3))*(2.0D0*EL(4)*Y3 - Y4)
        F(12,4) = (-WL(4)/(12.0D0*EL(4)**2))*(4.0D0*EL(4)*Y3 - 3.0D0*Y4)
        WRITE(6,*) '4',(F(N,4),N = 1,12)
        IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
          IF (NBLK.EQ.8) THEN
            Q(3) = -F(8,4)
          END IF
        ELSE IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
          Q(3) = -F(8,4)
        ELSE
```

```
        Q(6) = -F(2,4)
        Q(8) = -F(6,4)
        Q(9) = -F(8,4)
      END IF
    END IF
    IF (NBLK.EQ.8.OR.NBLK.EQ.9) THEN
      IF (NBLK.EQ.9) THEN
        K = 15
      ELSE
        K = 19
      END IF
     ELSE IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
      K = 19
    ELSE
      K = 25
    END IF
    DO 20 I = 4,MN(NTD)-6,8
      IF (I.NE.4) THEN
        IF (ISL(I).EQ.1) THEN
          WL(I) = (XLOAD*W(I))/AREAB
          WRITE(6,*) 'LOAD =',WL(I)
          F(2,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)**3*Y1-2.0D0
  $               *EL(I)*Y3 + Y4)
          F(6,I) = ( + WL(I)/(12.0D0*EL(I)**2))*(6.0D0*EL(I)**2*Y2
  $               -8.0D0*EL(I)*Y3 + 3.0D0*Y4)
          F(8,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)*Y3-Y4)
          F(12,I) = (-WL(I)/(12.0D0*EL(I)**2))*(4.0D0*EL(I)
  $               *Y3 - 3.0D0*Y4)
          WRITE(6,*) I,(F(N,I),N = 1,12)
          Q(K + 8) = -F(2,I)
          Q(K + 10) = -F(6,I)
          Q(K + 14) = -F(8,I)
        END IF
        K = K + 30
      END IF
 20 CONTINUE
    I = MN(NTD) + 21
    IF (ISL(I).EQ.1) THEN
      WL(I) = (XLOAD*W(I))/AREAB
      WRITE(6,*) 'LOAD =',WL(I)
      F(2,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)**3*Y1-2.0D0
  $           *EL(I)*Y3 + Y4)
      F(6,I) = ( + WL(I)/(12.0D0*EL(I)**2))*(6.0D0*EL(I)**2*Y2
  $           -8.0D0*EL(I)*Y3 + 3.0D0*Y4)
      F(8,I) = ( + WL(I)/(2.0D0*EL(I)**3))*(2.0D0*EL(I)*Y3-Y4)
      F(12,I) = (-WL(I)/(12.0D0*EL(I)**2))*(4.0D0*EL(I)
  $           *Y3 - 3.0D0*Y4)
      WRITE(6,*) I,(F(N,I),N = 1,12)
      IF (NBLK.EQ.4) THEN
        Q(213) = -F(8,I)
      END IF
      IF (NBLK.EQ.6.AND.NBLKPS.EQ.3) THEN
        Q(213) = -F(8,I)
      END IF
    END IF
    RETURN
    END
*************************************************************
*                    STACKD                                 *
*************************************************************
C   THIS SUBROUTINE CALCULATES THE MAXIMUM DEFLECTION IN THE TOP
C   STRINGERBOARDS FOR THE STACKED MODE
C
    SUBROUTINE STACKD(JN,NTD,XLC,DEFL,XMOE)
    DOUBLE PRECISION XLC(*),DEFL(*),X(20),Y(20),Y1MAX,XTARG,XLENG,
  $              X1(20),Y1(20),SUM,XMOE(*)
    INTEGER JN(*)
C
    IF (NTD.GT.3) THEN
      IF (XMOE(3).GT.1.0D00) THEN
        DO 30 II = 1,2
          IF (II.EQ.1) THEN
            L = 7
            M = 6
          ELSE
            L = 10
            M = 3
          END IF
          NMAX = 2
          X(1) = 0.0D00
```

```
         Y(1) = 0.0D00
         J = 1
         DO 10 I = L,JN(NTD)-M,6
          J = J + 1
          NMAX = NMAX + 1
          X(J) = XLC(I)
          Y(J) = DEFL(I)
 10      CONTINUE
         Y(NMAX) = 0.0D00
         IF (II.EQ.1) THEN
          X(NMAX) = XLC(JN(NTD)+1)
          XLENG = XLC(JN(NTD)+1)
         ELSE
          X(NMAX) = XLC(JN(NTD)+4)
          XLENG = XLC(JN(NTD)+4)
         END IF
         Y1MAX = 0.0D00
         DO 20 K = 1,10
          XTARG = (K*XLENG)/10.0D00
          X1(K) = XTARG
          SUM = 0.0D00
          CALL LAGRAN(NMAX,X,SUM,Y,XTARG)
          Y1(K) = SUM
          IF (Y1(K).GT.Y1MAX) THEN
           Y1MAX = Y1(K)
          END IF
 20      CONTINUE
         IF (II.EQ.1) THEN
          WRITE(6,*) 'MAX.DEFLECTION IN TOP STRINGERBOARD IS =',
     $               Y1MAX
         ELSE
          WRITE(6,*) 'MAX.DEFLECTION IN BOTTOM STRINGERBOARD IS ='
     $               ,Y1MAX
         END IF
 30     CONTINUE
       ELSE
        WRITE(6,*) 'MAX.DEFLECTION IN TOP STRINGERBOARD IS =',
     $             DEFL(2)
        WRITE(6,*) 'MAX.DEFLECTION IN BOTTOM STRINGERBOARD IS ='
     $             ,DEFL(4)
       END IF
      ELSE
       WRITE(6,*) 'TOO FEW DECKBOARDS FOR STACK ANALYSIS - MINIMUM
     $ NUMBER ALLOWED IS 3.'
      END IF
      RETURN
      END
C          ******************************
C                   LAGRAN
C          ******************************
C
      SUBROUTINE LAGRAN(NMAX,X,SUM,Y,XTARG)
      DOUBLE PRECISION X(20),Y(20),SUM,XTARG,XHAT
      DO 20 I = 1,NMAX
       XHAT = 1.0D00
       DO 10 J = 1,NMAX
        IF (I.EQ.J) GO TO 10
        XHAT = XHAT*(XTARG - X(J))/(X(I) - X(J))
 10    CONTINUE
       SUM = SUM + (Y(I)*XHAT)
 20   CONTINUE
      RETURN
      END
```

# Appendix D:

## Test pallet geometries

## and component properties

Table D.1. Characteristics of test pallets constructed for uniform loading, RAS. For geometry see Figure 7.1.

| Design | Pallet I.D No. | Species | No.of Deckboards | | No.of Stringerboards | | Stringerboards | | Deckboards | | Blocks | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Top | Bottom | Top | Bottom | Width (in) | Thickness (in) | Width (in) | Thickness (in) | Width (in) | Height (in) | Length (in) |
| 1 | 1 | Y.P | 8 | 3 | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.625 | 5.0 |
| | 2 | Y.P | 8 | 3 | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.625 | 5.0 |
| | 3 | Y.P | 8 | 3 | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.625 | 5.0 |
| 2 | 4 | Red Oak | 7 | - | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.01 | 5.0 |
| | 5 | Red Oak | 7 | - | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.01 | 5.0 |
| | 6 | Red Oak | 7 | - | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.01 | 5.0 |
| | 7 | Red Oak | 7 | - | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.01 | 5.0 |
| | 8 | Red Oak | 7 | - | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.01 | 5.0 |
| 3 | 9 | Red Oak | 7 | - | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.01 | 5.0 |

All designs have nine blocks.
All lumber in green condition.
Y.P = Yellow Poplar

Table D.2. Characteristics of test pallets constructed for uniform loading, RAD. For geometry see Figure 7.1.

| Design | Pallet I.D No. | Species | No.of Deckboards | | No.of Stringerboards | | Stringerboards | | Deckboards | | Blocks | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Top | Bottom | Top | Bottom | Width (in) | Thickness (in) | Width (in) | Thickness (in) | Width (in) | Height (in) | Length (in) |
| 1 | 1 | Y.P | 8 | 3 | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.625 | 5.0 |
| | 2 | Y.P | 8 | 3 | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.625 | 5.0 |
| | 3 | Y.P | 8 | 3 | 3 | 3 | 3.625 | 0.75 | 4.5 | 0.75 | 3.625 | 3.625 | 5.0 |
| 4 | 1a | Y.P | 9 | 3 | 3 | - | 5.5 | 0.75 | 3.5 | 0.75 | 5.5 | 3.5 | 3.5 |
| | 1b | Y.P | 9 | 3 | 3 | - | 5.5 | 0.44 | 3.5 | 0.44 | 5.5 | 3.5 | 3.5 |
| | 2a | Y.P | 9 | 3 | 3 | - | 5.5 | 0.75 | 3.5 | 0.75 | 5.5 | 3.5 | 3.5 |
| | 3a | Y.P | 9 | 3 | 3 | - | 5.5 | 0.75 | 3.5 | 0.75 | 5.5 | 3.5 | 3.5 |
| | 3b | Y.P | 9 | 3 | 3 | - | 5.5 | 0.44 | 3.5 | 0.44 | 5.5 | 3.5 | 3.5 |
| | 4a | Y.P | 9 | 3 | 3 | - | 5.5 | 0.75 | 3.5 | 0.75 | 5.5 | 3.5 | 3.5 |
| | 4b | Y.P | 9 | 3 | 3 | - | 5.5 | 0.44 | 3.5 | 0.44 | 5.5 | 3.5 | 3.5 |
| 5 | 1c | Y.P | 9 | 3 | 3 | 2 | 5.5 | 0.44 | 3.5 | 0.44 | 5.5 | 3.5 | 3.5 |
| | 3c | Y.P | 9 | 3 | 3 | 2 | 5.5 | 0.44 | 3.5 | 0.44 | 5.5 | 3.5 | 3.5 |
| | 4c | Y.P | 9 | 3 | 3 | 2 | 5.5 | 0.44 | 3.5 | 0.44 | 5.5 | 3.5 | 3.5 |

All designs have nine blocks.
All lumber in green condition.
Y.P = Yellow Poplar

Table D.3. Characteristics of Fasteners used in Experimental pallets

| | Deck-Block Joint | | Deckboard/ Stringerboard Joint |
|---|---|---|---|
| | Deckboard Thickness(in) | | All Thicknesses |
| | 0.375 0.4375 | 0.625 0.75 | |
| VPI Fastener No. | 2496B | 2625 | 2626 |
| type | Helical | Helical | Helical |
| Length (in) | 2.09 | 3.03 | 1.97 |
| Thread Diameter (in) | 0.137 | 0.145 | |
| Thread Length (in) | 1.38 | 2.07 | 1.31 |
| Mibant Angle (Degrees) | 23 | 16 | 16 |
| Wire Diameter (in) | 0.112 | 0.12 | 0.121 |
| Head Diameter (in) | 0.282 | 0.284 | 0.284 |

# Appendix E:

# Test results for center deflection and stiffness

Table E.1. Comparison between predicted and experimental deformations and stiffness of uniformly loaded pallets, RAS. Deckboard and stringerboard thicknesses are equal and 0.75 inches.

| Design | Pallet ID No. | Span (in) | Load (lbs) | Center deflection (in) | | | Stiffness (psi) | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Experiment | Model | % Difference | Experiment | Model | % Difference |
| 1 | 1 | 41 | 1151 | 0.197 | 0.203 | -3.0 | 5840 | 5669 | 2.9 |
| | | 43 | 1015 | 0.180 | 0.175 | 2.5 | 5650 | 5783 | -2.3 |
| | | 45 | 1062 | 0.215 | 0.242 | -12.5 | 4930 | 4388 | 10.9 |
| | 2 | 41 | 956 | 0.156 | 0.155 | 0.66 | 6435 | 6167 | 4.2 |
| | | 43 | 1046 | 0.181 | 0.174 | 4.0 | 5689 | 6021 | -5.8 |
| | | 45 | 1031 | 0.196 | 0.214 | 9.2 | 5132 | 4817 | 6.1 |
| | 3 | 41 | 1459 | 0.231 | 0.222 | 3.9 | 6311 | 6572 | -4.1 |
| | | 43 | 1062 | 0.188 | 0.175 | 6.9 | 5641 | 6065 | -7.5 |
| | | 45 | 1400 | 0.290 | 0.271 | 6.7 | 4735 | 5173 | -9.2 |
| 2 | 4 | 41 | 1469 | 0.261 | 0.27 | -3.4 | 5830 | 5440 | 6.6 |
| | | 43 | 1013 | 0.223 | 0.211 | 5.4 | 4534 | 4800 | -5.9 |
| | | 45 | 1322 | 0.331 | 0.375 | -13.3 | 3989 | 3525 | 11.6 |
| | 5 | 41 | 1038 | 0.181 | 0.188 | -3.9 | 5730 | 5521 | 3.6 |
| | | 43 | 1044 | 0.210 | 0.202 | 3.8 | 4917 | 5168 | -5.1 |
| | | 45 | 1026 | 0.199 | 0.212 | -6.5 | 5157 | 4839 | 6.1 |
| | 6 | 41 | 856 | 0.178 | 0.172 | 3.4 | 4819 | 4976 | -3.2 |
| | | 43 | 1020 | 0.210 | 0.218 | -3.8 | 4957 | 4678 | 5.6 |
| | | 45 | 1003 | 0.213 | 0.228 | -6.81 | 4809 | 4408 | 8.3 |
| | 7 | 41 | 1063 | 0.201 | 0.198 | 1.3 | 4988 | 5357 | -7.3 |
| | | 43 | 1026 | 0.220 | 0.204 | 7.3 | 4660 | 5029 | -7.9 |
| | | 45 | 1030 | 0.207 | 0.216 | -4.4 | 4957 | 4768 | 3.8 |
| | 8 | 41 | 1002 | 0.177 | 0.185 | -4.4 | 5861 | 5422 | 7.4 |
| | | 43 | 753 | 0.139 | 0.148 | -6.5 | 5517 | 5087 | 7.8 |
| | | 45 | 1192 | 0.240 | 0.247 | -5.03 | 5396 | 4825 | 10.6 |
| 3 | 9 | 33 | 1049 | 0.077 | 0.081 | -5.2 | 12803 | 12950 | -1.1 |

Table E.2. Comparison between predicted and experimental deformations and stiffness of uniformly loaded pallets - RAD

| Design | Pallet ID No. | Span (in) | Load (lbs) | Center deflection (in) | | | Stiffness (psi) | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | Experiment | Model | % Difference | Experiment | Model | % Difference |
| 1 | 1 | 34.5 | 1733 | 0.174 | 0.197 | -12.7 | 9920 | 8796 | 11.3 |
| | | 36.25 | 1130 | 0.115 | 0.124 | -8.1 | 9690 | 9112 | 5.9 |
| | | 38.0 | 1085 | 0.134 | 0.151 | -12.2 | 8069 | 7185 | 10.9 |
| | 2 | 34.5 | 1143 | 0.112 | 0.119 | -6.0 | 10102 | 9605 | 4.9 |
| | | 36.25 | 990 | 0.102 | 0.098 | 4.0 | 9782 | 10102 | -3.3 |
| | | 38.0 | 1348 | 0.169 | 0.156 | 8.2 | 7873 | 8641 | -9.7 |
| | 3 | 34.5 | 1521 | 0.163 | 0.172 | -5.2 | 9318 | 8843 | 5.1 |
| | | 36.25 | 1039 | 0.111 | 0.133 | -19.8 | 9349 | 7812 | 16.4 |
| | | 38.0 | 1038 | 0.136 | 0.138 | -1.7 | 7394 | 7521 | -1.7 |
| 4 | 1a | 39 | 2128 | 0.246 | 0.232 | 5.7 | 8640 | 9172 | -6.2 |
| | | 45 | 3157 | 0.422 | 0.428 | -0.5 | 7163 | 7374 | -2.9 |
| | 2a | 39 | 1983 | 0.233 | 0.222 | 4.9 | 8496 | 8940 | -5.2 |
| | | 45 | 2405 | 0.352 | 0.368 | -4.4 | 6821 | 6544 | 4.1 |
| | 3a | 39 | 2331 | 0.275 | 0.258 | 6.2 | 8467 | 9034 | -6.7 |
| | | 45 | 2852 | 0.388 | 0.430 | -10.9 | 7351 | 6632 | 9.7 |
| | 1b | 39 | 2037 | 0.816 | 0.861 | -5.5 | 2468 | 2365 | 4.2 |
| | | 45 | 2001 | 0.934 | 1.01 | -7.7 | 2024 | 2263 | -11.8 |
| | 3b | 39 | 2097 | 0.964 | 0.928 | 3.8 | 2517 | 2260 | 10.2 |
| | | 45 | 2098 | 1.226 | 1.128 | 7.9 | 1711 | 1859 | -8.5 |
| | 4b | 39 | 1583 | 0.854 | 0.69 | 19.0 | 1865 | 2294 | -23.0 |
| | | 45 | 2008 | 1.32 | 1.347 | -2.0 | 1512 | 1490 | 1.4 |
| 5 | 1c | 39 | 2113 | 1.0 | 0.92 | 8.0 | 2031 | 2296 | -13.0 |
| | | 45 | 1928 | 1.383 | 1.28 | 7.5 | 1349 | 1506 | -11.6 |
| | 3c | 39 | 1933 | 0.95 | 0.86 | 9.4 | 2019 | 2247 | -11.3 |
| | | 45 | 1826 | 1.433 | 1.41 | 1.6 | 1270 | 1295 | -1.9 |
| | 4c | 39 | 2224 | 1.034 | 1.0 | 3.3 | 2145 | 2224 | -3.7 |
| | | 45 | 1795 | 1.436 | 1.33 | 7.3 | 1195 | 1349 | -12.9 |