A DOUBLY-CURVED FINITE ELEMENT ANALYSIS OF THIN ARBITRARY SHELL STRUCTURES

by

John Billings Burchnall

Thesis submitted to the Graduate Faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in
Civil Engineering

APPROVED

S. M. Holzer, Chairman

R. H. Plaut

A. E. Somers, Jr.

March, 1977

Blacksburg, Virginia

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation and thanks to Dr. S. M. Holzer for his continual confidence, encouragement, and support during the development of this thesis. Additional thanks are given to Dr. R. H. Plaut, and especially to Dr. A. E. Somers, Jr., for their advice and encouragement. Appreciation is also expressed for the candid remarks of fellow graduate students, most notably

, , and .

Sincere thanks are given to the Civil Engineering Department for their financial support in the form of a Graduate Teaching Assistantship, and computer funds.

Dr. C. S. Desai, Dr. S. M. Holzer, and Dr. A. E. Somers have the author's sincere gratitude for the loan of some of their personal literatures, without which this research would not have been completed.

Special thanks are affectionately expressed to my wife, , to whom this thesis is dedicated, for her continual understanding and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

The following is a compilation of the important symbols and variables used in the text, along with the equation number where each is first used.

$\{\ \}$	column or row vector
$[\ ]$, $[\ ]^T$	rectangular matrix and its transpose
$(\ ,\ )$	matrix dimensions (# rows, # columns)
$b_i, c_i, i=1,2,3$	constants in the triangular coordinate transformation matrix (3.6)
$\{f\}$	element applied nodal force vector (1.5)
$\{f_p\}$	element nodal force vector due to applied tractive forces (1.5)
$\{f_{\varepsilon_o}\}$	element nodal force vector due to initial strains (1.5)
$[k]$	element stiffness matrix
$[k_F]$	element "flexure" stiffness matrix (3.28)
$[k_K]$	element "curvature" stiffness matrix (3.24)
$[k_{KM}]$	element "curvature-membrane" stiffness matrix (3.39)
$[k_M]$	element "membrane" stiffness matrix (3.33)
$[k_{MK}]$	element "membrane-curvature" stiffness matrix (section 3.2.5.4)
n	number of generalized coordinates per element (1.2)
$\{p\}$	element applied tractive force vector (3.51)

p_1, p_2, p_3	element surface coordinate applied tractive forces (3.51)
$\{q\}$	element generalized coordinate vector (1.2)
s_1, s_2, s_3	element local curvilinear surface coordinates (Section 3.2.1)
u, v, w	element midsurface tangential and normal displacements (1.1)
x, y, z	element local base-plane cartesian coordinates (Section 3.2.1)
z_x, z_y	partial derivatives of the midsurface elevation with respect to x and y (3.1)
z_{xx}, z_{yy}, z_{xy}	second partials of z (3.6)
z_4, z_5, z_6	elevations of element midside nodes 4, 5, 6, respectively (3.5)
A	area of the element base-plane (3.6)
[B]	element compatibility matrix (1.3)
[D]	element constitutive matrix (1.4)
E	Young's modulus of elasticity (3.9a)
$\{F\}$	system applied force vector (3.54)
$\{F^i\}$	element surface coordinate applied nodal force vector (3.46) (3.50)
$\{F_p\}$	element surface coordinate applied nodal force vector due to applied tractive forces (3.53)

$[K]$	system stiffness matrix (3.54)
$[K^i]$	element surface coordinate stiffness matrix (3.46) (3.50)
L_1, L_2, L_3	element local base-plane triangular (area) (natural) coordinates (Section 3.2.1)
N	total number of system generalized coordinates (3.54)
$[N]$	element interpolation (shape) (displacement) function (1.2)
$\{Q\}$	system generalized coordinate vector (3.54)
$\{Q^i\}$	element generalized surface coordinate vector (3.46) (3.50)
S_1, S_2, S_3	global system curvilinear surface coordinates (Section 3.2.1)
$[T]$	coordinate transformation matrix (3.46)
$\{W\}$	element midsurface displacement field vector (1.1)
X, Y, Z	global cartesian coordinates (Section 3.2.1)
$\{\varepsilon\}$	element midsurface strain field vector (1.3)
$\{\varepsilon_0\}$	element midsurface initial strain state vector (1.4)
$\varepsilon_1, \varepsilon_2, \gamma_{12}$	element midsurface strains (3.8)
ν	Poisson's ratio (3.9a)
θ_y, θ_x	nodal tangential rotation generalized coordinates (Section 3.2.6)

$\{\sigma\}$	element midsurface stress field vector (1.4)
$\sigma_1, \sigma_2, \tau_{12}$	element midsurface stresses (3.9a)
ς	midsurface normal strains (3.8)

CHAPTER 1

INTRODUCTION

1.1 Objectives

This thesis topic was originally undertaken in an effort to solely perform a linear elastic, static finite element analysis of thin arbitrary shell structures. In the process of carrying out this study, the three objectives in the following paragraphs developed.

Realizing that others may be in the same position as the author was, the first objective of this thesis is to present a discussion of the various factors involved in selecting a shell finite element. Advantages and disadvantages are discussed, along with the reasoning behind the selection of the shell element utilized in this study.

The second objective of this paper is to present a rather detailed formulation of a 27 degrees of freedom, arbitrary, doubly-curved, nonconforming, triangular shell element. Element formulation utilizes the shallow shell theory of Novozhilov [56]. Curvatures are assumed constant within an element by specifying elevations to vary quadratically with respect to the two base plane coordinates. Both the normal and tangential displacement fields are expressed by the "incomplete" cubic plate bending displacement functions introduced by Bazeley, Cheung, Irons, and Zienkiewicz [7]. These functions are expressed in "natural", "area", or "triangular" coordinates, thus simplifying integration procedures in deriving a semi-explicit

element stiffness matrix. To the author's knowledge, no published text or article has attempted to use these particular interpolation (shape) functions to describe both the normal and tangential displacement fields.

The development and presentation of a WATFIV/FORTRAN computer code to facilitate the implementation of the formulated shell element in a linear elastic, static analysis of thin arbitrary shells is the third objective of this thesis. In order to obtain a more compact program, a few simplifications were made. Besides being capable of analyzing completely arbitrarily shaped shell surfaces, this program can analyze rectangular plate sections, right circular cylindrical shell sections, hyperbolic paraboloid shell sections, and elliptic paraboloid shell sections, with very little input data. Further program limitations, capabilities, and methodology are discussed in a following chapter. Demonstration problems are presented and comparisons are made with solutions in the literature.

Unfortunately, an unidentifiable problem is present in either the implementation of the theory into the computer code, or the code itself. As a result, the displacement values as calculated by the computer are in great error. Attempts to identify and rectify this problem within the time constraints available have failed. Since the theory behind the coding is believed correct, and it incorporates some sophisticated and useful techniques, the complete program listing is contained in the appendix.

1.2 The Finite Element Method

It is not the purpose of this paper to present a detailed discussion of the history, background, principles, and mathematical foundations of the general analysis technique known as the "finite element method". Rather, what follows in this section is a brief introduction and definition of the finite element method as it pertains to structural analysis. For a more comprehensive coverage of the finite element method the reader is urged to examine other publications [11, 20, 27, 32, 40, 53, 63, 79].

The finite element technique is a numerical modeling process by which the behavior of a continuum, with infinite generalized coordinates, can be approximated by the combined actions of a discrete assemblage of subregions each with a finite number of generalized coordinates. More generally, it is a method for piecewise approximating the governing differential equations of a continuous system by a set of algebraic simultaneous equations involving a finite number of variables. In structural applications, this involves dividing up the structural domain into subdivisions, called elements, and selecting a number of elemental points, called nodes, which usually lie along the inter-element boundaries of the elements. The displacements and/or force measures at these nodes are taken as the generalized coordinates. The displacements and/or force measures at any point within the interior of an element are uniquely defined in terms of its nodal generalized coordinates through the use of a set of interpolation functions. Governing behavioral equations for each element are then

generated by the application of an appropriate variational principle. Finally, these elemental equations are assembled together, through the use of inter-element nodal compatibility and equilibrium, to form one set of algebraic equations that approximates the behavior of the entire structure. The simultaneous solution of these governing equations yields the approximate structural response of the continuum structure subject to various actions or loading conditions placed upon it.

The popularity of the finite element method in structural analysis is due to a number of factors. One of the main reasons is its great flexibility in accounting for arbitrary geometries, boundary conditions, loading, and material property variations. In addition, the finite element method can be used to analyze a structural model composed of a variety of structural forms such as plates, solids, shell segments, and ribs or stiffners. The relative simplicity of the method's physical interpretations and mathematical formulations has contributed to its widespread popularity. One of the main incentives for the finite element method, and indeed other numerical approximation techniques, was the development of the modern high capacity, high speed computer. The ease with which finite element methods can be programmed has further aided its development and popularity. Another major advantage of the finite element method over most other approximation methods is the very sparse and banded form of the governing algebraic equation coefficient matrix. This property enables large savings to be made in terms of storage costs and computational efficiencies. One of the more powerful uses

of the finite element method is the analysis of completely arbitrary shell structures. This capability is so powerful because, as Flugge [30 (p. 415)] relates, the analysis of completely arbitrary shells is almost completely out of reach of formal classical closed form solution methods. Razzaque concurs and observes that the eighth-order partial differential equation that follows from a theoretical simplified shallow shell analysis can only be solved for in a very few specialized cases [63 (p.1.1)]. Thus, the finite element method, in conjunction with the computer, represents a realistic tool with which most complex structural engineering problems, particularly shells, can be analyzed.

1.2.1 The Displacement Approach

Early procedures for formulating the elemental governing behavioral equations, such as force-displacement relations, were based on a direct approach similar to those employed in the Direct Stiffness matrix structural analysis method. Since the finite element method is now regarded as having its foundations in the Calculus of Variations and the Ritz method [59, 60], several more versatile variational approaches have evolved for element formulations. For a review of the general concepts and principles of Variational Calculus and energy methods the reader is referred to a text on that subject [48], or to many of the previously mentioned finite element texts [11, 27, 32, 40].

There are basically four variational approaches used in elemental formulations. These have led to the following classifications of the finite element method:

(1) The Displacement (Compatibility) Method

(2) The Equilibrium (Force) Method

(3) The Mixed Methods

(4) The Hybrid Methods

The displacement method involves assumed displacement functions and an element formulation based on the Principle of Stationary Potential Energy. The Principle of Stationary Complementary Energy is used in the assumed stress field equilibrium method. Both displacement and stress fields are assumed in the mixed methods which are based on Reissner's variational principle. Various hybrid finite element methods have developed in which displacement fields are assumed for part of the element, while stress distributions are assumed for the rest of the element. Certain modified energy principles are utilized in the elemental formulation. A more complete description and understanding of the several finite element method classifications is presented in various publications [4, 27, 33, 60, 63].

The displacement method is considered the most general and simplistic approach. Consequently many reliable and accurate formulations have been developed using this method. Some successful elements have been derived utilizing the equilibrium method, although difficulties have been encountered in the definition of

applied load states and in the calculation of displacements [32 (p. 357)]. Strict mathematical upper or lower bound properties are lacking in the use of the mixed or hybrid methods [32 (p. 207)]. Due to the relative simplicity of derivation and generality of application the more popular displacement approach is selected as the finite element method used in this thesis. Further comments in this work concerning the finite element method shall be implicitly restricted to the finite element displacement method.

In the displacement approach, the generalized coordinates are the nodal displacements and some partial derivatives thereof. Because of the physical interpretation of these generalized coordinates, they are often described as nodal degrees of freedom (DOF), or simply nodal displacements.

Matrix algebra is generally used in simplifying the expressions and operations involved in finite element formulations. Thus, a flow chart is presented in Figure 1.1 showing the basic steps involved in the formulation of a displacement type finite element. Following discretization of the continuum, the displacements and strains of an arbitrary point on the interior of an element are defined as functions of the element's generalized coordinates, or displacement parameters. These relationships, along with the theory of elasticity constitutive laws (Hooke's law), are combined with the Principle of Virtual Work or the stationary Potential Energy Principle to derive expressions for the element local stiffness matrix, the

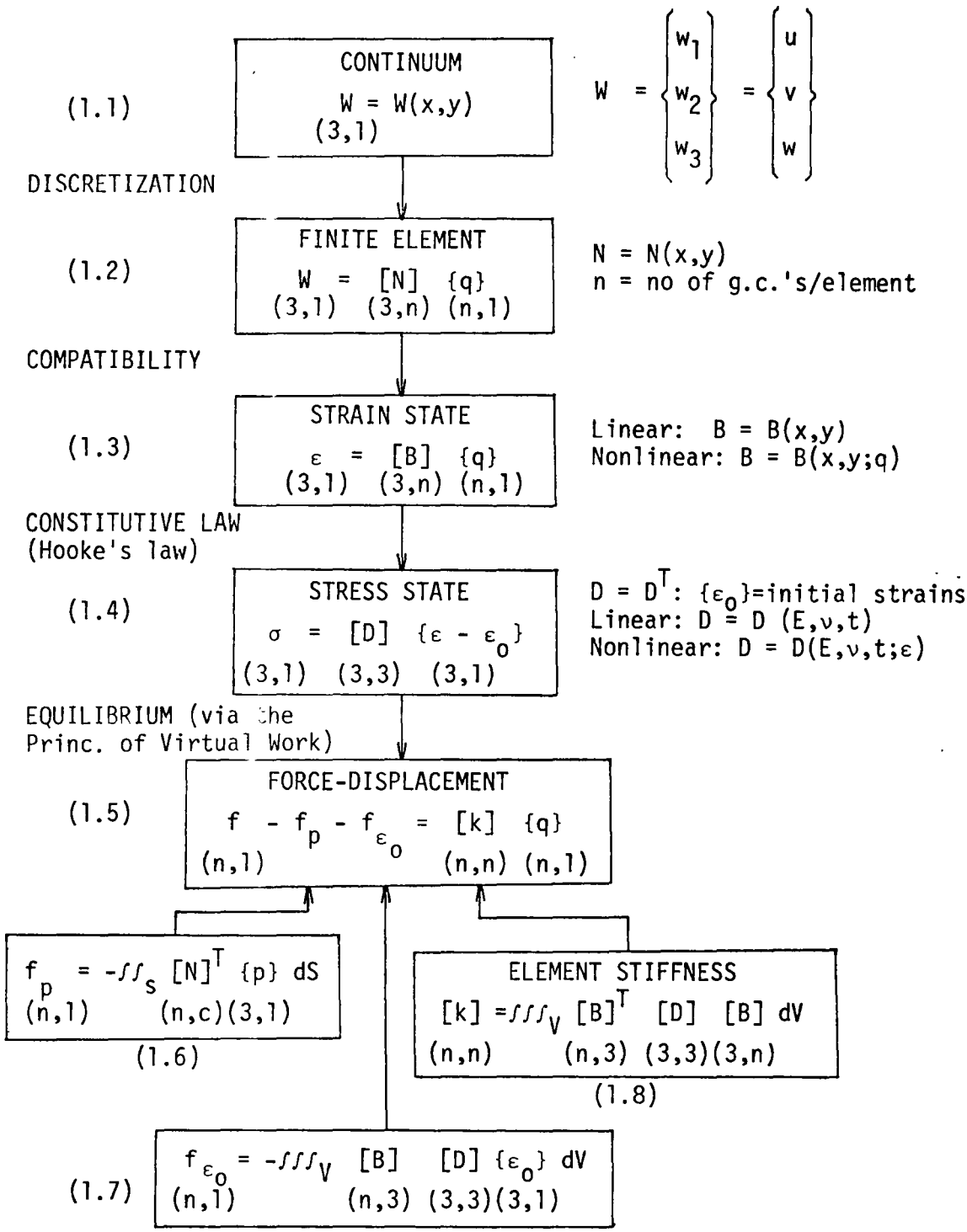


Figure 1.1. Schematic representation of the displacement approach finite element formulation for plate or shell structures [38].

generalized nodal force vector caused by tractive forces, and the generalized nodal force vector caused by initial strains. For a more detailed presentation of the displacement type finite element formulation the reader is referred to other literature [11, 27, 32, 38, 40, 53, 63, 79].

Once these elemental expressions are derived, analysis procedures follow the lines of the well known Direct Stiffness matrix structural analysis method [39, 62, 76]. The primary difference being that element equilibrium conditions are satisfied only in the overall sense in the finite element method, whereas in the direct stiffness matrix method the elemental forces are pointwise equilibrilating [38].

1.2.2 Convergence

The accuracy and performance of any finite element analysis depends on how closely the continuum model is represented by the discrete (finite element) model. Two inherent approximations are generally associated with the finite element displacement method [16, 18]:

(1) Geometric Approximations (i.e. the selection of element shape, curvature, and proportions; and the manner and degree of mesh refinement and boundary matching).

(2) Selection of Elemental Assumed Displacement Fields.

The effects of type (1) approximations, while usually influencing the rate at which the finite element solutions converge to the "exact" continuum solutions, tend to vanish in the limit as the mesh size and element curvatures approach zero. Thus, the convergence of finite element solutions is largely dependent upon the type and form of the assumed displacement fields. These assumed displacement functions generally govern the interelement compatibility and strain state characteristics of the elements.

Another approximation frequently occurring in finite element analyses involves the numerical integration used in calculating stiffness matrix coefficients. The use of lower order numerical integration causes inaccuracies that lead to a softening of the structure [20 (chap. 5)]. However, these inaccuracies also tend to diminish in the limit as element size approaches zero. Thus, they do not prevent convergence, but they may affect the rate of convergence. Nevertheless, care should be exercised in the selection of numerical integration schemes.

It should be realized that the finite element discrete model can, at its best (usually in the limit), only precisely predict the "exact" behavioral response of the continuum model, the latter of which is usually only a close approximation to the actual response of the actual structure that is being analyzed. These last inaccuracies are due to the approximations inherent to the various structural mechanics theories used in simplifying the actual

structure to a continuum mathematical model. Thus, approximations of this type do not affect the convergence of the finite element solutions to the "exact" continuum model solutions. Some of these approximations will be discussed in the next chapter (sections 2.3 and 2.6).

In an attempt to guarantee convergence, several guidelines for the selection of displacement (interpolation, shape) functions have evolved. These guidelines are specializations of, or derived from, the two basic mathematical convergence criteria [22, 35, 38, 40 (p. 79), 59, 72, 79 (p. 35)]:

(1) Completeness - the displacement field must form a complete set and exhibit C^s continuity on the interior of the element, where s = the highest order derivative in the potential or strain energy functional.

(2) Conformity - the displacement field must exhibit C^{s-1} continuity at the inter-element boundaries of the element.

In the particular case of beam, plate, or shallow shell analysis, s = 2. Thus, satisfaction of the completeness criterion means that the assumed displacement functions, and their first and second derivatives, must be continuous and exist (i.e. not identically vanishing) in the limit as the element size diminishes to zero. The completeness criterion also states that the displacement functions must be complete; i.e., in the case of polynomials, they

must include all terms up to, and inclusive of, an arbitrary order in Pascal's polynomial triangle (see Figure 2.1). Fulfillment of the conformity criterion requires that displacements and slopes (first derivatives of displacements) be continuous and compatible at inter-element boundaries. Elements whose shape functions satisfy criterion (1) are called "complete elements", while those satisfying criterion (2) are called either "conforming elements" or "compatible elements".

In order for mathematical convergence to be proved, the process of mesh refinement must be regular [38, 40 (p. 79)]:

(a) Every point of the solution domain must always be within a finite element.

(b) All previous meshes must be contained in all refined meshes.

(c) The form of the interpolation functions must always remain unchanged.

If the mesh refinement is regular and both the completeness and conformity criteria are satisfied, then monotonic convergence (in a stable domain) is mathematically guaranteed [14, 22, 38, 40 (p. 79), 58, 59]. This is because the finite element method now becomes a specialized case of the well known Ritz method. Consequently, the potential energy functional converges monotonically from above to the potential energy value of the corresponding continuum, while the displacement values converge monotonically

from below to their respective continuum "exact" values. Thus, converging finite element displacement solutions always underestimate the "actual" solutions [27 (p. 80), 40 (p. 225)], and the stiffness matrix is always symmetric and positive definite [79 (p. 31)].

Before these rigorous mathematical convergence criteria were fully understood or formulated, various intuitive finite element convergence requirements were proposed. Today, these intuitive conditions are generally regarded as supplemental convergence requirements, which mostly represent qualifications, specific applications, or physical interpretations of the completeness or convergence criteria. Many of these auxiliary convergence requirements, along with some experimental convergence tests for complete but nonconforming elements, are discussed in the following two sections.

1.2.2.1 Completeness Considerations

The following two intuitive convergence criteria are frequently used, which in some cases represent substitutional criteria for the completeness criterion [27 (p. 81), 32 (p. 212), 40 (pp. 81, 224), 43, 63(p. 1.9), 79 (p. 28)]:

(1a) The displacement functions must be of a form capable of allowing elements to experience rigid body motions (translations or rotations), while maintaining zero strain energy.

(1b) The displacement functions must be of a form capable of allowing elements to exhibit constant strain states.

It can be reasoned that condition (1a) is a specialized case of this condition.

Condition (1a) is reasoned necessary since spurious straining due to rigid body motions (the most elementary deformations) would cause the strain energy functional to be in error. But, condition (1a) tends to be, and need only be, satisfied in the limit as element size and curvature go to zero [57 (p. 116)]. Thus, condition (1a) is not a necessary condition for convergence. In practice, the satisfaction of the rigid body modes criterion may require extremely complicated procedures and expressions in elemental formulation. This is particularly true in the case of curved elements, where difficulties arise in the expression of rigid body translations and rotations in terms of curvilinear coordinates [32 (p. 213), 33, 63 (p. 43)]. Thus, many curved element formulations consciously violate condition (1a), with the understanding that convergence will be slowed, but not precluded [32 (p. 213), 33, 57 (p. 116)]. In the case of curved shell elements, Greene, et al. [35] suggest that rigid body modes be at least approximately represented. They state that this can be adequately accomplished by selecting equal and sufficiently high degree (order) complete polynomial expansions for the displacement functions of u, v, and w. Interestingly, it has been shown that the highest order of complete polynomials of the displacement fields also governs the rate of convergence [32 (p. 226), 38]. Cook also arrives at the conclusion that satisfaction of

requirement (1a) is a desirable, but not necessary, condition for convergence [20 (p. 87)]. Cowper, et al., feel that the rigid body mode requirement is over-emphasized [23]. It should be noted that the number of rigid body modes contained in a system of element stiffness equations corresponds to the number of zero eigenvalues of the stiffness coefficient matrix [32 (pp. 46, 213)].

Satisfaction of requirement (1b), at least in the limit, is however a necessary condition for convergence to the continuum solution [20 (p. 87), 42, 57 (p. 116), 79 (pp. 28, 171)]. Violation of this condition may result in convergence to significantly incorrect values, since individual elements would not be capable of assuming the constant strain states that they would normally converge to in the limit [27 (p. 81), 32 (p. 213), 33, 79 (p. 28)]. Satisfaction of this requirement can usually be achieved by assuming sufficiently high degree complete polynomial expansions for the displacement functions.

A third auxiliary condition associated with the completeness criterion is usually mentioned in convergence discussions [20 (p. 90), 27 (p. 83), 32 (pp. 216, 258), 40 (p. 134)].

(1c) Individual elements must demonstrate the same general convergence behavior, irregardless of their geometric orientation in the global analysis domain.

This behavior is called "geometric isotropy" or "geometric invariance," and elements that exhibit this behavior are called "isotropic" or

"invariant". Satisfaction of this requirement is easily achieved if complete, or "balanced" complete polynomial expressions, or incomplete polynomials expressed in "intrinsic" or "natural" local coordinates, are selected for displacement functions [20 (p. 90), 40 (p. 135)]. Thus, condition (1c) is satisfied if the general completeness criterion, (1), is satisfied.

In the case of beams and plates, conditions (1a) and (1b) are identical to criterion (1). However, for curved beams and shells they are not the same [38, 57 (p. 116)]. Therefore, it is important to keep in mind the more fundamental completeness criterion of (1).

1.2.2. Conformity (Compatibility) Considerations

Satisfaction of the inter-element compatibility criterion, (2), implies that adjacent elements must deform without exhibiting openings (gaps), overlaps, or discontinuities (kinks) along inter-element boundaries [27 (p. 81)]. Violation of conformity would result in infinite strains at element interfaces, and would result in undesirable virtual work contributions [40 (p. 81), 79 (pp. 25,29)]. Conformity can be achieved in the case of beam, plate, or shallow shell elements (i.e. C^1 elements), if the following three requirements are satisfied [27 (p. 178)]:

(2a) The same isotropic displacement model should be used for connecting elements.

(2b) Interelement nodal compatibility, including at least displacements and their first derivatives (i.e. slopes), must be enforced.

(2c) For each element, the displacements and slopes along each interface must be uniquely defined by the nodal generalized coordinates (i.e. displacements, slopes, etc.) along that boundary.

Condition (2a) is usually satisfied, but is really not necessary as long as the displacement functions and nodes of adjacent elements are similarly compatible. Condition (2b) is satisfied by the standard assembly procedure. Condition (2c), however, is not easily satisfied.

In the case of plate or shell elements, satisfaction of the conformity requirement may be mathematically difficult and may involve excessive numerical computation [33, 40 (p. 81), 79 (p. 171)]. In a single field flat triangular plate bending element formulation, a complete polynomial of at least fifth order (21 DOF) in the description of the transverse displacement field, and the inclusion of midside nodes (at least initially) is required to completely satisfy inter-element compatibility [32 (p. 347), 33]. As will be discussed later, the use of high order displacement functions and midside nodes results in considerable mathematical difficulties and computational inefficiencies. Conformity is also difficult to achieve in the case of plate or shell problems, due to neglecting transverse shear deformations in the assumption that rotations are equivalent to derivatives of the transverse displacement [14, 38, 57 (p. 143), 59]. Additionally, plate or shell elements do not usually conform in the

cases of box girders, folded plates, roof lines, and branching shells [63 (p. 4.8)]. This is because the in-plane or tangential displacements of some elements must match the bending or transverse displacements of other elements. Accomplishment of this form of inter-element compatibility requires equality of the order of polynomial representation of both bending and membrane displacements [63 (p. 4.8)].

Because of the difficulties or resulting complications in satisfying the conformity requirement, (2), several investigations have consciously violated this criterion in formulations of plate and shell elements [7, 17, 51]. There have been a number of cases where these complete, but incompatible, elements have been used successfully. Not only do some nonconforming elements yield acceptable results, but they usually provide superior results for the same discretization [7, 20 (p. 89), 27 (p. 181), 40 (p. 81), 79 (pp. 188, 205)]. This may well be due to the fact that the incompatible solution no longer strictly follows the variational method energy bounds, and consequently has a greater freedom (flexibility) to assume the best shape [79 (p. 189)]. Cook [20 (p. 89)] observes that the assumption of relatively low order displacement fields results in a "stiffer" structure, while inter-element incompatibilities (i.e. separations, overlapping, kinks) tend to "soften" the structure. Thus, the cancellation of these two effects frequently results in nonconforming elements performing better than their "stiffer" conforming element

counterparts. It should be noted that the convergence of some non-conforming elements is mesh dependent [7, 45, 57 (p. 107), 58, 59, 79 (p. 188)].

The successful convergence of some incompatible elements suggests the conclusion that conformity is not a necessary condition for convergence [40 (p. 176), 79 (pp. 29, 171)]. Indeed, it can be reasoned that the undesirable effects of inter-element incompatibility will diminish in the limit as elements assume more simplified, and thus more easily conforming, shapes. Zienkiewicz [79 (p. 29)] and Cook [20 (p. 89)] concur with this reasoning and suggest that the ability of elements to assume constant strain conditions in the limit is the main convergence criterion. In support of this experimental conclusion, Oliveira mathematically shows that completeness, and not conformity, is the necessary condition for convergence; while both completeness and conformity together represent sufficient conditions for convergence [38, 58, 59].

Consequently, various criteria and numerical tests have been developed in an effort to examine the convergence properties of complete, but nonconforming, formulations.

Oliveira states that a sufficient condition for convergence is the stipulation that displacement components and their derivatives up to order "s" must be capable of assuming any constant value within an element [59]. Furthermore. Oliveira shows that sufficient conditions for convergence, in the case of C^1 elements, are the requirements that the displacement functions must be complete through

the second order and that the third derivatives of the shape functions must be continuous and bounded within the element as its size approaches zero in the limit [45, 53 (p. 25), 59]. Oliveira also shows that uniform regular mesh conditions (i.e. three equally spaced parallel line mesh generators) allow for the boundedness of the third derivatives of the Bazeley, et al. nonconforming plate bending shape functions [45, 53 (p. 25)].

One of the most well known numerical tests for examining the convergence characteristics of complete, but nonconforming, elements is the "patch-test" of Irons and Razzaque [38, 44, 53 (p. 26), 63 (p. 1.11), 79 (p. 188)]. This test involves prescribing displacements (i.e. deflections and slopes) at the boundary nodes of a small patch of elements corresponding to a known, but arbitrary, state of constant strain. If the free nodes in the "interior" of the patch voluntarily assume the displacements resulting in constant strains at every point within the patch, then the element will converge in the limit. Strang proved that passage of this test is a sufficient condition for convergence [70]. It may be reasoned [20 (p. 88), 63 (p. 1.11)] that this test is mainly a numerical demonstration of the previously discussed necessary convergence criterion of constant strain states (1b). Thus, the patch-test is both a necessary and sufficient convergence criteria [63 (p. 1.11)]. Irons uses the patch-test to substantiate Oliveira's previous claim that the convergence of the Bazeley, et al. nonconforming triangular

plate element is guaranteed only for uniform regular meshes [45]. Veubeke uses the patch-test in examining the properties of twelve different triangular elements [74].

Other numerical procedures have been developed for evaluating the convergence properties of nonconforming finite elements. Many of these methods are reviewed in articles by Gallagher [33] and Robinson [65]. Robinson also presents and demonstrates in his article a new "simple and practical" test known as a "single element test".

In the case of nonconforming elements, it should be remembered that monotonic convergence of displacements from below is not guaranteed. This is because a nonconforming formulation is not a specialized condition of the Ritz method. Thus, bounds are not available in the case of incompatible formulations. It is interesting to note that displacement solutions resulting from nonconforming elements usually converge monotonically from above, although occasionally they converge non-monotonically from either side [79, (p. 205)].

As a final note, finite element formulations should be judged not only on how they perform in the limit, but also on the accuracy that they can achieve with a given mesh size [38]. Perhaps the combination of simplicity of formulation and this accuracy consideration are the reasons behind the practicality and popularity of some nonconforming elements [33, 79 (p. 172)].

CHAPTER 2

GENERAL SHELL ELEMENT CONSIDERATIONS

2.1 Introduction

A number of factors must be considered when selecting a particular finite element for use in general shell analysis. A case in point is the conformity discussion of the previous chapter. Compromises between sophistication of element formulation and computational expense are necessary to ensure a practical, yet accurate, shell analysis technique.

A qualitative discussion of the various factors involved in selecting a "shell" finite element is presented in this chapter. A comprehensive survey of existing shell elements is not presented here, although, in some instances, reference is made to a specific previously formulated finite element. It is felt that this presentation approach is more organized, and consequently of more value to practicing engineers and graduate students, than the usual survey-type approach found in other theses and publications. However, since element surveys are also informative, some of the better references of this type are mentioned in the following paragraph.

Gallagher presents an exhaustive survey of plate and shell elements in his article [33]. Attention is also paid to the formulative procedures, accuracies, and pitfalls of various elements. Fairly extensive surveys, discussions, and numerical comparisons of various plate bending elements, along with some formulations, are presented in books by Brebbia and Conner [11], Cook [20], Desai and

Abel [27], Gallagher [32], Huebner [40], and Zienkiewicz [79]. Additionally, Brebbia/Conner and Desai/Abel present helpful tables summarizing the references for, and characteristics of, existing plate bending elements. Razzaque offers a conprehensive survey and discussion of both plate and shell elements [63 (Chap. 1)], while Millward presents a good review of flat triangular bending elements [53]. Perhaps the best coverage of finite element analysis of thin shell structures is found in articles by Clough and Wilson [18] and Clough and Johnson [16]. Excellent discussions and comparisons of many shell elements are presented in these two articles, along with a review of the various approximations and procedures associated with thin shell finite element analysis.

It is hoped that the following discussions may particularly benefit those presently engaged in selecting a finite element formulation to be used in thin shell analysis. The discussions presented herein are as unbiased as possible, although the author's preferences and selections are ultimately and necessarily revealed. These selections are also summarized in the conclusion of the chapter.

2.2 Assumed Displacement and/or Stress Fields

As previously stated in Section 1.2.1, the displacement approach is selected as the method of finite element analysis used in this thesis. This choice is made due to the popularity, familiarity, simplicity, accuracy, and reliability of the displacement approach. Consequently, assumed displacement fields are solely used in describing the element behavioral configurations.

However, several successful bending elements have been formulated using assumed stress fields with or without the conjunction of assumed displacement fields. Recently, Allman has developed a "simple cubic displacement element for plate bending" utilizing hybrid techniques [1]. Two hybrid elements along with an excellent description of the hybrid stress approach are presented in an article by Cook [21]. Mixed-type triangular and rectangular elements are formulated by Poceski [61]. Further references for assumed stress and/or displacement elements are contained in the excellent review by Razzaque [63 (Chap. 1)], or in many of the previously mentioned finite element texts.

2.3 Thin vs. Thick Shell Analysis

Novozhilov suggests that if the ratio of the radius of curvature to thickness of a shell structure is greater than or equal to twenty, then that shell can be classified as "thin" [36, 56]. This ratio for most practical shell structures falls within the range of 50 to 1000 [36, 56]. Thus "thick" shells are the exception rather than the rule.

When thin shell analyses are used, several Fundamental Assumptions of Thin Shell Theory [30, 36, 56, 66] are used in developing a more simplified continuum model. Perhaps the most important assumption of this type is the Kirchhoff-Love Hypothesis which assumes that normals to the middle surface remain straight, unstrained, and normal to the deformed middle surface. Consequently, transverse shear stresses and

deformations may be ignored in thin shell or plate analyses. As a result, the Kirchhoff-Love Hypothesis allows for a two dimensional (surface) representation of a three dimensional shell or plate problem, and thus it simplifies the stress-strain relationships (constitutive law matrix) and the strain-displacement relationships (compatibility matrix) considerably. Because this approximation is reasonable for homogeneous thin plates and shells [11 (p. 199)], most thin shell and plate elements are formulated using this approach.

However, realizing the slight inaccuracies or limited applications of the above mentioned elements, several investigators have included transverse shear strains in their formulations of both thin and thick shell elements. Approaches range from additional assumed rotational deformation functions, to the use of complete or degenerate isoparametric or super-parametric 3-D brick-type elements. Again, Razzaque presents excellent discussions and references on this topic [63 (Chap. 1)]. Zienkiewicz also offers some discussion, references, and formulations on this subject [79]. Transverse shear deformations, with reference to shell structures, are discussed by Wempler, et al. [77].

Because most shell structures are considered "thin", and since element formulation is significantly simplified, the classical thin shell assumptions and approaches are followed in this thesis. Thus, any subsequent references to shell structures in this work specifically imply "thin" shell structures.

2.4 Triangular vs. Other Shaped Elements

In flat plate analysis, the investigator has a wide field of successful element shapes to select from. Among the most commonly used shapes are triangles, rectangles, rings, and general quadrilaterals. Isoparametric forms of these elements [27, 32, 79] are frequently used, especially in curved boundary problems. Conformity requirements are more easily met in rectangular element formulations, since their sides meet at right angles [35]. However, the approximation of curved boundaries is difficult with these elements. Besides the isoparametric forms, the triangle is the most versatile element for use in curved boundary situations [63 (p. 1.14), 79 (p. 115)]. The simplest plate element formulation is that of a Rectangle (79 (p. 172)], while isoparametric formulations or applications are generally the most complex. Triangular plate element meshes can be easily refined in areas of high stress gradients, without changing much of the surrounding mesh pattern [63 (p. 1.14)].

In thin shell analysis, the choice of element shapes narrows somewhat. Quadrilateral elements are rarely capable of approximating the geometry of a curved shell surface. Rectangular elements can only be used in the case of developable or translational shells. This is because co-planar four point locations are generally not prevalent on the doubly-curved middle surface of arbitrary shell structures. Perhaps the most successful element of this shape is the 15 DOF compatible rectangular curved element developed by Bogner, Fox, and

Schmit (6, 9, 40 (p. 225)]. Various annular, ring, or conical frustra elements have been successfully formulated and used [69, 79 (Chap. 12)], but these are generally limited to the analysis of shells of revolution or restricted to simple axi-symmetric loading conditions. As mentioned in the previous section, some isoparametric and super-parametric [see Ref. 63] elements have been used in doubly-curved shell analysis. However, these formulations are complex and lead to an excessive number of DOF when used in thin shell analysis. Either flat or doubly-curved triangular element assemblages can be easily used in geometrically representing or approximating an arbitrarily curved shell surface.

Because triangular shaped elements are more powerful in analyzing high stress gradients and general shell geometries, they are selected as the type elements used in this thesis. This decision is reached knowing that some triangular element formulations are fairly complex, and conformity requirements are often violated. Further discussions of shell element considerations in this chapter specifically refer to triangular shaped elements.

2.5 Flat vs. Doubly-Curved Triangular Elements

Having selected the basic element shape, the next step is to decide whether flat or doubly-curved triangular element assemblages are to be used in approximating and analyzing the shell structure. To a large extent, both approaches have been successfully, and fairly comparatively, used. Thus, the decision in this instance is not as obvious and clear-cut as those of previous sections.

The beauty and simplicity of the flat element approach is that an element is formed by simply combining a flat plate bending element with a plane stress element. Hence, no new element formulation is required if existing plate and plane stress elements are used. The effects of membrane and bending coupling occur only at the nodes. Thus, the complexities and arbitrariness of various shell theories are avoided. Zero element curvature simplifies the expression of rigid body modes. Because of the simplicity of derivation of flat shell elements, it is possible to use higher order displacement functions in an effort to more closely describe the continuum behavior.

However, there are many disadvantages in using flat elements to represent a curved shell surface. In many cases, a very large number of flat elements are required to adequately approximate the shell geometry. Consequently, an extremely large number of equations may have to be stored and solved in cases of strongly curved shell structures. Additionally, the accuracy gained by the use of higher order displacement functions is wasted due to the extremely fine mesh refinements required. Also, the additional number of DOF and the concurrent increase in the number of simultaneous equations associated with higher order elements may exceed computer storage and excution time capabilities. Moreover, legitimate questions as to whether individual elements can be classified as "thin" may arise in the case of very small mesh sizes.

This faceted approximation of the shell structure leads to other difficulties. Since membrane and bending coupling never occurs within elements, the behavior of the governing differential equations of the continuum shell structure is not approached in the limit [33]. This factor is of minor effect, but does point out the need for a more refined mesh in gaining acceptable results. Also, the initial non-planar assemblage of flat elements makes conformity requirements impossible to satisfy [16, 33, 53]. Consequently, convergence rates are poor when assemblages of flat elements are used to approximate curved shells [53]. Since it is well known that relatively small slope discontinuities of a shell midsurface can lead to large pertubations in the local stress distribution, it is reasoned that this may occur at the juncture of non-planar flat elements [71]. Indeed, these discontinuities of slope between adjacent elements can and do cause spurious or residual bending moments in some cases [33]. However, this difficulty can be circumvented at the cost of increasing solution complexity [33]. Solutions may falsely show that kinking or folding of the shell has occurred in some cases where highly unbalanced loading conditions or irregular meshes are used [53]. A final disadvantage of the faceted shell approach is that element local stiffness matrices must undergo a coordinate transformation before assembly into the global stiffness matrix. This transformation requires extra calculations of transformation matrix coefficients and usually an increase in the size of the element stiffness matrix. This is because an additional in-plane nodal

rotation coordinate is usually included in the global assembly transformation. This additional DOF at each node usually increases solution time by a 1.728 factor, along with increasing the complexity of assembly and size of storage required [18, 63 (p. 4.11)]. In addition, singularities are possible in the global stiffness matrix if all the elements joining at a particular node are coplanar [18, 63 (p. 4.11)]. However, these problems can be avoided by using tangent plane transformations that utilize nodal mean normals [18, 63 (p. 412)]; but the definition of these mean normals introduces further complexities.

The advantages gained by using doubly-curved elements to analyze shell structures are numerous. The continuum is more closely represented by doubly-curved shells, since membrane and bending coupling occur at the element level. Additionally, inter-element compatibility, of at least the primary displacements, is more easily satisfied [71]. Since doubly-curved elements more closely approximate the continuum geometry, the use of larger mesh sizes (and thus fewer elements and DOF) is feasible. Futhermore, it is possible to avoid the intricacies of differential geometry prevalent in general shell theories by using a specialized shallow shell theory [20 (p. 192)]. This is discussed more fully in the following section. More variations of membrane stresses are possible in doubly-curved elements, as compared to comparable flat elements, due to the curvature coupling effect of membrane stresses now depending on normal displacements in addition to in-plane displacements [71]. Additionally, since it is well

known that the rapid variation of membrane stresses near a shell edge is primarily caused by the rapid variation of the normal displacement near that edge, it is reasoned that doubly-curved elements can more accurately represent these edge stresses [71]. Because of the previously stated factors, one may expect that doubly-curved element assemblages would out-perform comparable flat element assemblages. Experimental results show that only slightly better results are obtained with doubly-curved elements in some cases [16], while significantly better results are shown in other cases [28, 10]. One final advantage of doubly-curved shell elements is that a coordinate transformation is not normally required during the assembly process [10]. This is because the local stiffness matrices are related to coordinates defined on the element surfaces, and thus they are usually inherently expressed in terms of a common surface coordinate system (i.e. curvilinear coordinates).

Naturally, there are also many disadvantages in using doubly-curved shell elements. Perhaps the main drawback is that a more substantial amount of development expense is required for the element formulations. While shallow shell theories simplify doubly-curved formulations considerably, complexities in the calculation of element stiffness matrices and curvatures still exist. The definition of element geometry, especially with regard to curvatures, can be extremely difficult, tedious, or data consuming, in the cases of irregularly shaped shells whose geometry cannot be expressed in

mathematical form. This becomes an important factor when coordinate transformation matrices (possibly needed for the calculation of the system applied load vector) are desired. Also, the use of doubly-curved elements does not usually result in an exact representation of the continuum shell surface, although the approximation is normally very close. The inaccuracies result from the common simplifying assumption that each element initially assumes a constant principle curvature state. Although the use of curvilinear coordinates effectively eliminates the necessity of coordinate transformation, it hinders the representation of rigid body modes (as discussed in Chapter 1). Additionally, displacement functions which satisfy rigid body modes may in turn violate compatibility requirements [63 (p. 4.3)]. However, as discussed in Chapter 1, violation of the rigid body modes requirement is permissible, as good results are still obtainable [10, 16, 19, 33]. Gallagher maintains that equality of the order of all in-plane and transverse displacement functions is mandatory if conformity conditions are to be satisfied [33], but Cowder, et al., debate this point [23]. When doubly-curved elements are used, difficulties occur in the selection of an appropriate shell theory [33]. The arbitrariness and simplifying assumptions of these various shell theories is subject to some concern [79 (p. 124)]. A final disadvantage in using doubly-curved shell elements is the possible difficulty in calculating consistent nodal load vectors corresponding to various distributed applied loads [10]. This complication occurs with the use of shallow shell formulations. As

a result, lumped concentrated nodal loads are used in place of consistent nodal load vectors.

With all factors considered, the doubly-curved shell element approach is selected as the method used in this thesis project. This approach is chosen since it more closely represents the continuum problem. Also, formulative and geometric problems can be alleviated by using shallow shell theories, and "automatic" node generation and curvature calculation schemes. Thus, the next section in this chapter is concerned with the selection of an appropriate shell theory.

2.6. Selection of an Appropriate Shell Theory

The finite element discrete model can never yield better results than the continuum model it approximates. Thus, it is important to develop a continuum model that accurately predicts the behavior of the actual structure in question. In the case of shell structures, the accuracy of the continuum model depends on the assumptions or simplifications inherent to the particular shell theory used in formulating the continuum model of the actual structure. The previously mentioned Fundamental Assumptions of Thin Shell Theory (Section 2.3) are examples of the types of approximations inherent to shell theories. With this background in mind, the following discussions are concerned with the selection of an appropriate shell theory to be used in thin shell finite element formulations.

A variety of shell theories are available for use in doubly-curved shell element formulations. Basically, a decision must be reached

between using one of the simpler shallow shell theories of Flugge [30], Marguerre [50], Novozhilov [56], and Reissner [64]; or the more exact, but complex, nonshallow shell theories of Flugge [30], Love [49], and Novozhilov [56]. Some of these individual theories are further discussed by Brebbia and Conner [11 (Chap. 7)], Kraus [47], Wempner [78], and Seide [66]. However, the important issue is whether a shallow or nonshallow shell theory is to be used.

In shallow shell theory the squares of the middle surface slopes of an element with respect to its projected base plane are assumed negligible with respect to unity. This assumption enables surface integrals of various functions in the element mid-surface to be evaluated over the area of the base plane [71]. Also, shape functions defining tangential (tangent to middle surface) and normal (normal to middle surface) displacements are expressed in terms of the base plane coordinates. Consequently, precisely the same shape functions as used in flat elements can be used in corresponding shallow elements; while all integrations are carried out in the same plane as in the case of flat elements [79 (p. 214)]. In order to satisfy the fundamental shallow shell assumption and to simplify element formulation somewhat, the undeformed initial shape of the element midsurface is usually expressed as a quadratic variation of the base plane coordinates. Thus, by taking advantage of the fact that individual element surfaces do not deviate much from a plane, a relatively simple shell element that includes membrane-bending coupling (vault-plate interaction) [30 (p. 415)] can be formulated. The avoidance

of the intricacies and complexities of the differential geometries and tensor notations prevalent in general deep shell theories is perhaps the main advantage of the shallow shell approach [20 (p. 192), 23]. Also, since shallow element formulations are only a little more complicated than corresponding flat element formulations, numerical integration may be avoided in some instances [23]. Moreover, shallow element formulations are not restricted to shallow shell structures. This is because most practical subdivisions required in the analysis of deep shell structures are more than adequately refined (small enough) so as to make each individual element shallow with respect to their particular base plane [33]. Indeed, good numerical results are obtained when assemblages of shallow shell elements are used in analyzing deep shells [20 (p. 193)].

There are several disadvantages generally associated with the shallow shell approach. First, as mentioned in the previous section, the generation of consistent nodal load vectors due to distributed applied loads is very difficult. This is because the application of Equation (1.5) results in a consistent nodal load vector corresponding to distributed loads acting on the element base plane, rather than acting on the curvilinear mid-surface. Also, the resulting nodal loads correspond to base plane coordinates rather than curvilinear coordinates. The use of shallow shell theory limits to some extent the number of possible applications of these elements. Intersecting shell elements in particular may not be analyzed by the use of shallow shell elements [63 (p. 4.2)]. Finally, again as

mentioned in the previous section, assemblages of shallow shell elements usually still only approximate the continuum shell surface geometry. This is because the constant curvature representation of individual elements does not ensure continuity of slopes and curvatures between elements [10].

The advantage in using nonshallow shell theories is that the geometric and structural behavior of the continuum is very nearly duplicated. Compatibility conditions are rarely violated. However, element formlulations are extremely complicated, and high order displacement functions are frequently used. This results in the mandatory usage of high order numerical integration, and elements with large numbers of DOF.

Despite its complexities, nonshallow shell elements have been successfully formulated by Argyris, et al. [2, 3], Cowper, et al. [23], Dawe [25], and Garnet and Crouzet-Pascal [34]. Excellent results have been obtained, especially with the SHEBA family of elements developed by Argyris, et al.

However, the shallow shell approach has been successfully employed by a larger number of investigators. Bonnes, et al. [10], Cowper, et al. [23, 24], Dhatt [28, 29], Strickland and Loden [71], Utku [73], and Vos [75] have all formulated shallow shell elements. The popularity of this approach is most likely due to its relative simplicity and accuracy.

Doubly-curved triangular elements were selected for this thesis research with the understanding that a relatively simple, accurate,

and explicitly integrated element could still be formulated. Thus, the shallow shell approach is rationally selected as the method employed in this research. More specifically, the Novozhilov shallow shell theory [56], identical to the Flugge shallow shell theory [30], is selected for use in the derivation of the doubly-curved triangular shell element developed in this paper. This particular shallow shell theory is selected because the quantities in the strain-displacement compatibility equations relate to curvilinear coordinates in the mid-surface, rather than to coordinates in the base plane, as in the theory of Marguerre [71].

2.7. Natural vs. Cartesian Interpolation Polynomials

Since triangular elements are generally used in aribitrary shell analysis, the possibility of using displacement (interpolation, shape) functions expressed in terms of natural (triangular, area, intrinsic) coordinates, rather than the usual cartesian coordinates, occurs. For a review of triangular coordinates the reader is referred to Appendix A.

The evidence is overwhelming in favor of using area coordinate polynomials in interpolation functions of triangular elements. Because these coordinates are dimensionless, they simplify element derivations [27 (p. 88), 38]. Also, since triangular coordinates are internally defined within the boundaries of a triangle, the global orientation of the element makes no difference in the element behavioral characteristics. Thus, the geometric anisotropy problems

of some earlier cartesian polynomial triangular element formulations are eliminated [33]. Finally, integrals of area coordinate polynomials can be easily evaluated using simple factorial expressions. Thus, explicit forms of all integrals, particularly stiffness matrix coefficients, are obtainable.

However, these advantages are only a factor in fairly simple element derivations, since explicit calculations become exasperatingly complex in the cases of higher order elements. Consequently, the stiffness matrix of any conforming triangular element, whether formulated with natural or cartesian polynomials, has never been explicitly expressed. Since triangular coordinate polynomial expressions have one more variable than similar cartesian expressions, it can be reasoned that some mathematical operations with triangular expressions may be more involved than the same operation involving an equivalent cartesian expression.

The selection of natural or cartesian interpolation polynomials also depends on the characteristics and type of the desired interpolation functions. Thus, this decision is delayed until subsequent considerations are discussed.

2.8 Single-Field vs. Subelement Formulation

One of two general approaches is usually followed in formulating triangular displacement-type finite elements. The first, and more familiar, method involves initially selecting one set of interpolation functions that define all displacements within the entire element

domain. Since this one set of assumed displacement functions covers the total domain of the element, this method is called the "single-field" approach. Alternately, a "subdomain" or "subelement" approach may be used. In this method, the element is separated into a number (usually two or three in the case of triangles) of subregions, each with its own separate set of assumed displacement functions. A reduced single set of interpolation functions governing displacements over the entire element is then formulated by applying compatibility constraints and condensation procedures.

The primary advantage of the subdomain approach is that it can be used to formulate a conforming, yet low order, plate bending element [14, 15, 17]. Also, by proper selection of the local coordinates used for the subelements, isotropic elements can be formulated even though incomplete polynomials may be used for subelement assumed displacement functions. However, subelement formulations are more involved than single-field formulations because of their reduction process calculations. The condensation process is particularly annoying, since the inversion of a 9 x 9 matrix, or larger, is typically required. Despite this handicap, Clough and Tocher [17], Clough and Johnson [15], Clough and Felippa [14], and Shieh, et al. [67] successfully formulate "simple" conforming triangular plate elements using the subelement approach. Furthermore, Bonnes, et al. [10] and Dhatt [34] use this method in developing doubly-curved shallow shell triangular elements.

While single-field formulations of low order triangular plate bending elements usually result in nonconforming elements, the derivations are more straightforward and do not require the previously mentioned matrix inversion. Consequently, Bazeley, et al. [7] and many other investigators use the single field approach in deriving low order triangular plate elements. Also, Dhatt [28] and Strickland and Loden [71] use the single-field approach in deriving "simple" doubly-curved triangular shallow shell elements. Additionally, since compatibility requirements are more easily satisfied in higher order single-field triangular element formulations, practically all investigators use the single-field approach in developing high performance elements.

Since a relatively simple element is desired for use in this thesis research, the selection of either a single-field or subelement formulation becomes relevant. While possibly sacrificing isotropy and/or conformity, the single-field approach is selected as the method employed in the research. This decision is made in order to keep element formulation as simple as possible and as explicitly expressed as possible, while realizing that isotropic element formulations are still possible.

2.9 Comments on Internal and Side Nodes

Before discussing the selection of displacement functions, a few comments on the inclusion of internal and side nodes are in order. In some cases it may be necessary or seem desirable to include

"secondary", especially midside, nodal variables as additional element generalized coordinates or DOF. However, the retention of these extraneous nodes in the final element formulation results in computational efficiency penalities.

These penalities are especially severe in the case of side nodes. Inclusion of side nodal variables results in a significant increase in the bandwidth of the assembled system stiffness matrix. Additionally, side nodes, each of which is common to but two elements, have a disproportionately large effect on the dimensions of the assembled system stiffness matrix as compared to corner nodes [8]. Thus, the accompanying inflation of computer storage requirements coupled with the increased computer execution time imposes disproportionately severe penalities upon computational efficiency [55]. Consequently, the retention of side nodes in final element formulations is highly undesirable, although many elements so formulated exhibit excellent numerical results.

Early investigators used midside nodes to satisfy conformity requirements in "simple" triangular bending element formulations. These undesirable midside nodes are subsequently eliminated by imposing constraints on the midside variable(s). However, this results in an "unduly overstiff structure" [55]. Thus, the development of higher order conforming elements is encouraged, since midside connections are easily eliminated at the element level, while

sacrificing little loss in accuracy [55]. However, such highly refined elements are beyond the scope of this thesis research.

The use of interior nodes also causes similar significant penalities in computational efficiencies, although not quite as severe as in the case of side nodes. Static condensation [37] may be used to eliminate interior nodes from final formulations but this sometimes results in an extremely overly flexible element [17].

Another disadvantage in using side or interior nodes is that matrix handling schemes are frequently annoyingly complicated. These complications result from the typically unequal number of DOF assigned to the side or interior nodes as compared to the corner nodes.

Consequently, the exclusion of these "secondary" nodes in element formulations is strongly recommended [8, 79 (p. 208)].

2.10 Order of Interpolation Polynomials

Selection of the type and order of the polynomial expansions used in formulating interpolation (displacement) functions is perhaps the most important and the most difficult decision to be made in the finite element formulation process. This one decision may have a pronouced effect on: Geometric isotropy properties, satisfaction of the completeness criterion, satisfaction of the conformity criterion, the convergence rate, the complexity of element formulation, the possibility of explicit integrations, the use of secondary nodes

(i.e. interior and side nodes), the use of second derivatives as nodal variables, and the possible limitations in structural application. In short, the general performance and computational efficiency of a finite element largely depends on the form of the assumed displacement functions.

The selections of the orders of the polynomial expressions used in defining the two surface-tangent (tangential) displacements, "u" and "v", and the one surface-normal (normal) displacement, "w", are of primary importance and concern in choosing interpolation functions. Some typical polynomial forms used in formulating element shape functions are listed in Figure 2.1. For reference, the numerical order of complete polynomials is given, along with the number of DOF per element initially required for complete utilization of each particular polynomial order.

Triangular plate bending elements with the normal displacement, "w", expressed in terms of "cubic" or lower order polynomials are generally termed "lower order", "simple", or "less refined" elements. Conversely, if "quartic" or higher order polynomial expressions are used to describe "w', then the triangular elements are usually classified as "higher order", "high precision", or "refined" elements.

However, in the case of triangular shell elements, a different classification scheme is used. The behavior and efficiency of shell elements additionally depends on a suitable balance of the separate polynomial orders used in defining tangential and normal displacement

						ORDER	#DOF
1						Constant (0)	1
x	y					Linear (1st)	3
x^2	xy	y^2				Quadratic (2nd)	6
x^3	x^2y	xy^2	y^3			Cubic (3rd)	10
x^4	x^3y	x^2y^2	xy^3	y^4		Quartic (4th)	15
x^5	x^4y	x^3y^2	x^2y^3	xy^4	y^5	Quintic(5th)	21
x^6	x^5y	s^4y^2	x^3y^3	x^2y^4	...	Sextic (6th)	28
x^7	x^6y	x^5y^2	x^4y^3	x^3y^4	...	Septic (7th)	36

COMPLETE CARTESIAN POLYNOMIALS

$1 = L_1 + L_2 + L_3$	Constant	1
$L_1 \quad L_2 \quad L_3$	Linear	3
$L_1^2 \quad L_2^2 \quad L_3^2 \quad L_1L_2 \quad L_2L_3 \quad L_3L_1$	Quadratic	6
$L_1^3 \quad L_2^3 \quad L_3^3 \quad L_1^2L_2 \quad L_2^2L_3 \quad L_3^2L_1$ $L_1^2L_3 \quad L_2^2L_1 \quad L_3^2L_2 \quad L_1L_2L_3$	Complete Cubic	10
$L_1 \quad L_2 \quad L_3$ $(L_1^2L_2 + \frac{1}{2}L_1L_2L_3) \quad (L_1^2L_3 + \frac{1}{2}L_1L_2L_3)$ $(L_2^2L_3 + \frac{1}{2}L_1L_2L_3) \quad (L_2^2L_1 + \frac{1}{2}L_1L_2L_3)$ $(L_3^2L_1 + \frac{1}{2}L_1L_2L_3) \quad (L_3^2L_2 + \frac{1}{2}L_1L_2L_3)$	Bazeley, et al. "Cubic" [7]	9

AREA COORDINATE POLYNOMIALS

Figure 2.1 Some polynomials used in formulating finite element displacement (shape) functions, (Note: Each order of area coordinate polynomials inherently contains all previous lower order polynomial expressions).

functions. Thus, the following combinations of complete polynomial orders respectively describing tangential ("membrane") and normal ("bending") displacement functions are used in the classification of triangular shell elements [63 (p. 4.6)]:

(1) Linear-Quadratic [5, 26]

(2) Linear-Cubic [15, 71]

(3) Quadratic-Cubic [63]

(4) Cubic-Cubic [10, 28, 29, 34]

(5) Quadratic-Quartic [13]

(6) Cubic-Quintic (incomplete) [23, 24]

(7) Quintic-Quintic [2]

(8) Higher orders ... [2]

Elements in categories (1) - (4) are generally classified as "lower order" elements, while elements of categories (5) - (8) are usually referred to as "higher order" elements.

On a one-to-one basis, the convergence performance of higher order elements is superior to that of lower order elements, since their increased flexibility allows a closer approximation of the continuum deformations. But, higher order element formulations can be extremely complex. Thus, many investigators question whether the use of higher order elements is an advantage or a disadvantage. Razzaque says that the accuracy of solutions, and the continuity of stresses, generally will efficiently improve with the use of higher order displacement functions, provided the number of elements

used to represent the shell structure remains constant [63]. However, he also relates that the corresponding increase in the number of generalized coordinates (DOF) in each element will considerably increase the number of simultaneous equations to be solved for, along with an associated increase in bandwidth. Thus, Razzaque concludes that the extra computer storage and computational effort are important factors in considering the use of higher order elements, since they may seriously limit the total number of elements that can be used. Zienkiewicz offers the general rule that given the same accuracy of representation, the total number of unknowns is reduced with the use of higher order elements [79 (p. 104)]. But, he adds that the reduction in the total number of variables does not necessarily result in economical advantages in data preparation or computational costs, since element formulation costs increase along with the decrease in equation solution time. Cook concludes that the performance of higher order plate elements may be surpassed by simpler plate elements, when judged on the combined basis of accuracy and computational effort [20 (p. 174)].

Some other generalities and comments concerning the selection of orders of polynomial displacement functions should be realized. For instance, the order of the displacement functions used in a shell element should be consistent with the manner in which that element geometrically approximates the continuum. In other words, higher order displacement functions should not be used in flat shell elements, since the resulting increase in accuracy is offset by the poor

geometric approximation arising from using fewer elements [63 (p. 4.4)]. Conversely, very low order displacement functions should not be used in shallow shell element formulations, since the more accurate geometric approximation is wasted by the poor displacement behavior. Recalling that membrane strains generally depend on the first order derivatives of the in-plane displacement functions, while bending strains depend on the second order derivatives of the normal displacement functions, it seems reasonable that shape functions should be consistently selected so that the order of the normal coordinate displacement polynomial is of one order higher than the order of the tangential displacement polynomials. Indeed, this is one of the reasons behind the use of category (1) and (3) elements. However, as previously discussed, equality in the order of all polynomials is necessary to help satisfy conformity requirements in the cases of intersecting shell or plate structures. It should also be pointed out that the satisfaction of conformity requirements in lower order triangular shell elements is generally not possible. In fact, for straight forward single field formulations, complete quintic displacement polynomials are minimumly required in order to totally satisfy conformity requirements [32 (p. 347), 33]. Finally, it should be remembered that geometric anisotropy problems may occur if "unbalanced" incomplete polynomial expressions are used in defining displacement functions.

The use of higher order elements presents some additonal difficulties. Since higher order displacement functions contain

greater numbers of polynomial terms, additional element DOF are required for higher order elements. These additional DOF are included by either adding secondary nodes, or by specifying higher order derivatives as additional DOF at the primary nodes. As discussed in the previous section, midside nodes are very undesirable. Thus, the second approach is typically favored [27 (p. 84)]. However difficulties in physical interpretations arise when derivatives of second order, or above, are used as nodal DOF. The imposition of boundary conditions and the interpretation of nodal forces are particularly confusing [79 (p. 209)]. Also, difficulties occur in applications to discontinuously thick plates or shells [63 (p. 1.26)].

With this general background in mind, some more specific observations concerning particular shell element classifications can be made. Some comments from Razzaque's Ph. D. thesis [63 (p. 4.7)] are included in the following discussion.

Category (1) and (2) elements exhibit very poor convergence characteristics, particularly in problems where membrane action predominates. Both membrane and bending stresses are constant in the case of flat category (1) elements, but conformity requirements are not satisfied, and midside nodes are utilized. Bending stresses are sufficiently approximated in category (2) elements, but membrane stresses are still poorly approximated. Conformity may be accomplished in elements of category (2), but at the expense of using the subdomain approach and/or numerical integration.

Recently, Razzaque devised a category (3) element [63]. This flat shell element exhibits linear variations in both the membrane and bending stresses. Low order numerical integration is used, and convergence is guaranteed for any mesh pattern. Despite the presence of midside nodes, this appears to be the simplest successful flat triangular shell element yet devised [63].

Category (4) elements appear to represent the best compromise between higher and lower order elements. The midside nodes of category (3) elements are avoided at the expense of adding six additional DOF, two at each corner node. Additionally, these elements possess similar difficulties in satisfying the conformity and rigid body modes conditions as in the case of category (2) elements.

Many of the disadvantages of category (5), (6), (7), and (8) elements have already been discussed. Because of their extreme flexibility properites, these elements may exhibit spurious bending and false folding responses. For reasons previously discussed, flat shell elements of these categories should never be used.

2.11 Conclusion

As shown in the preceding sections, the selection of an optimal shell element is not an easy decision to make. Razzaque relates that it is important to consider the best practical balance between simplicity of formulation, simplicity of implementation, versatility of application, reliability, computational effort, and accuracy

[63 (p. 4.5)]. More generally, the consideration of most factors involves seeking the best possible balance between displacement approximations and geometric approximations. Razzaque also states that no one element is presently universally accepted as the best all-around shell element, since no one element is superiorly efficient in analyzing every type of shell structure [63 (p. 4.5)]. Gallagher adds that tradeoffs between element formulative efforts and computer solution costs are not yet clearly established [33].

Taking all factors into account, a 27 DOF "incomplete" cubic-cubic, nonconforming, doubly-curved, triangular, shallow shell element is selected, by the author, for use in the following finite element analysis of thin, arbitrary shell structures. (A category (4) element is selected over a category (3) element because straightforward derivations, explicit integrations, and the exclusion of midside nodes are preferred by the author). The Bazeley, et al. [7] nonconforming, "incomplete" cubic interpolation polynomials are selected for use in prescribing both the tangential, and the normal, displacements; thus resulting in a 27 DOF element. The nine terms of these "cubic" natural coordinate polynomials are given in Figure 2.1. The advantage in using these particular displacement functions is that only three DOF per corner node (per displacement coordinate) are initially needed. Thus, secondary nodes or subelement approaches are not needed as in the case of other cubic formulations. The tenth term in the Bazeley, et al. "cubics" is ingeniously eliminated by initially incorporating it into six of the other nine terms [79 (p. 185)].

This can be done because the tenth term, $L_1 L_2 L_3$, is a purely internal node which exhibits zero values and slopes at all three corners [79 (p. 185)]. Additionally, the use of natural coordinates eliminates anisotropy problems, and simplifies explicit integrations.

A "cubic" triangular plate bending element has been formulated using these polynomials [7]. It exhibits reasonably good accuracies in the prediction of deflections, but sometimes gives unexpected stress patterns [63 (p. 1.26)]. Also, convergence is not guaranteed for all mesh patterns [7]. Conformity is not satisfied, but explicit integrations and semi-explicit stiffness coefficient matrix expressions are possible. Gallagher states that this element gives excellent results compared to the simplicity of formulation [32 (p. 351)], and thus is more desirable from a combined standpoint of accuracy and simplicity of formulation [33]. Because of these reasons, this nonconforming plate triangle is still widely used [63 (p. 126)]. .

These "cubic" polynomials have also been used in describing the normal displacement, "w", in both flat and shallow shell formulations of some category (2) shell elements. However, the convergence rates of these shell elements are poor, particularly in the case of doubly-curved formulations. Clough and Johnson attribute this slow convergence to the crude representation of membrane displacements [16].

Considering the characteristics of the above mentioned elements, it seems reasonable that a fairly accurate, yet fairly simple, doubly-curved triangular shell element could be formulated by using

the Bazeley, et al. "cubic" polynomial expressions in the definition of both the tangential displacements, "u" and "v", as well as the normal displacement, "w". Consequently, the shell element presented in this thesis research is so formulated. To the author's knowledge, no published article or test has utilized this particular element in the analysis of thin shell structures.

CHAPTER 3

A DOUBLY-CURVED FINITE ELEMENT THIN SHELL ANALYSIS

3.1 Introduction

A finite element discrete model for use in linear elastic, static analysis of thin arbitrary shell structures is presented in this chapter. A comprehensive formulation of the previously mentioned 27 DOF, "incomplete" cubic-cubic, nonconforming, doubly-curved, triangular, shallow shell finite element is given. Semi-explicit stiffness matrices are also developed and presented.

3.2 Shell Element Formulation

3.2.1 Element Geometry

The spatial orientation of a typical doubly-curved triangular shallow shell element is given in Figure 3.1, along with a graphical display of some of the coordinate systems used in this analysis. An explanation of these various coordinate systems is necessary before further formulations are discussed.

There are basically five different coordinate systems present in this finite element analysis. They are as follows:

X, Y, Z — Global cartesian coordinates. Only needed if some applied loads are initially expressed in terms of these coordinates.

x, y, z Local (base-plane) cartesian coordinates. For simplicity, the x-axis is assumed to intersect primary nodal points (1) and (2). The y-axis also lies in the base-plane, while the z-axis is normally to the base-plane.

L_1, L_2, L_3 Local (base-plane) natural coordinates used in formulating shape functions, describing the midsurface elevation, and calculating explicit integrals.

s_1, s_2, s_3 Local surface (curvilinear) coordinates. Usually the same as the global surface (curvilinear) coordinates, thus an assembly transformation is not normally required. Curvilinear coordinate lines s_1 and s_2 are determined by base-plane normal projections of axes x and y, respectively. Coordinate lines s_3 are everywhere normal to the element midsurface. Note that coordinate lines s_1 and s_2 will in general not be orthogonal, but this error may be neglected due to the shallow shell assumption [71].

S_1, S_2, S_3 Global (system) surface (curvilinear) coordinates. As previously stated, these coordinates are normally identical to the local surface coordinates.

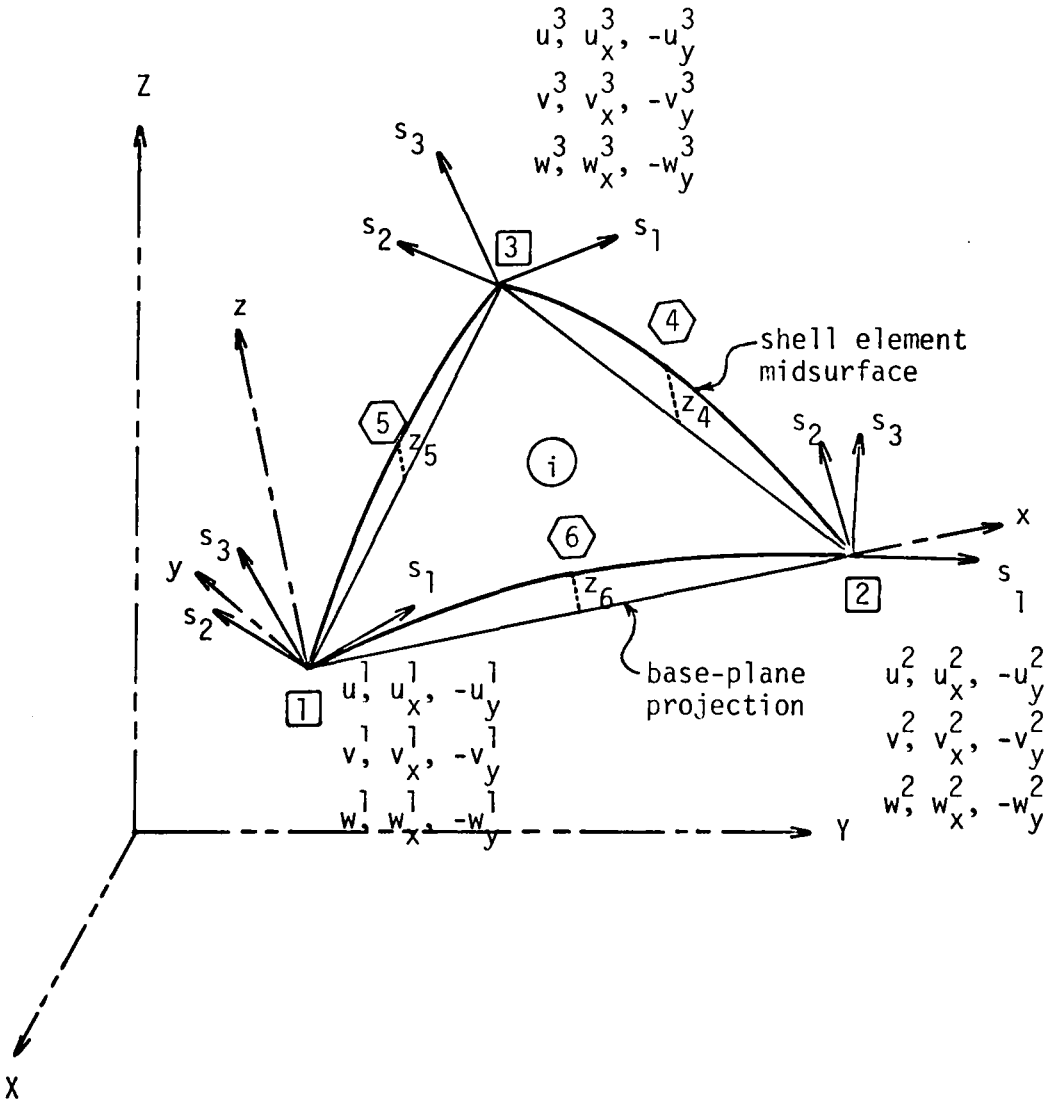


Figure 3.1 Orientation and initial nodal generalized coordinates of a doubly-curved triangular shallow shell element in space.

Even though there are five coordinate systems present in this analysis, only four of them are usually distinct, and only three of those are normally needed in actual analysis. Additionally, only one minor coordinate transformation is usually required during the analysis process. This one transformation involves a simple conversion from the nodal values of the first partial derivatives of the normal displacement "w" with respect to the base-plane x and y coordinates, to the corresponding nodal rotation components of the local surface coordinates. The details of this conversion are discussed in a following section. Other more complex transformations may be necessary if the applied loads are not initially expressed in the local or global surface coordinates. Strickland and Loden describe these various transformations in their article [71].

The midsurface of each element is assumed shallow only with respect to its own base-plane. In mathematical terms this means that

$$z_x^2 << 1, \quad z_y^2 << 1, \quad |z_x z_y| << 1 \tag{3.1}$$

where

$$z_x = \frac{\partial z}{\partial x} \quad , \quad z_y = \frac{\partial z}{\partial y} \tag{3.2}$$

and

$z = z(x,y) = z(L_1,L_2,L_3)$ = the element midsurface elevation with respect to the element base-plane.

These shallow shell assumptions lead to the following specializations of general shell theory curvature relations [10, 29]:

$$\begin{aligned} 1/R_{11} &= -z_{xx} \\ 1/R_{22} &= -z_{yy} \\ 1/R_{12} &= -z_{xy} \end{aligned} \tag{3.3}$$

where R_{11} corresponds to the radius of curvature of the element midsurface with respect to the s_1 local curvilinear coordinate.

In order to satisfy the shallow shell assumptions, and to simplify element formulation, the element midsurface elevation, z, is assumed to vary quadratically with respect to the base-plane coordinates. Mathematically this means that:

$$z(x,y) = B_1 + B_2x + B_3y + B_4xy + B_5x^2 + B_6y^2$$

or, in terms of natural coordinates,

$$z = C_1L_1 + C_2L_2 + C_3L_3 + C_4L_2L_3 + C_5L_1L_2 + C_6L_1L_2 \tag{3.4}$$

Constants C_i can be solved for by substituting the natural coordinates of the three primary corner nodes and the three secondary midside nodes into equation (3.4). Since the corner nodes lie in the base-plane, constants C_1, C_2, and C_3 assume zero values. Thus the equation of the midsurface becomes:

$$z = 4(z_4L_2L_3 + z_5L_1L_3 + z_6L_1L_2) \tag{3.5}$$

where z_4, z_5, and z_6 represent the elevations of their respective element midside nodes above the base-plane, as shown in Figure 2.1.

Using relations (A.9) and Equation (3.5) the following second-order partial derivatives of the midsurface elevation coordinate are obtained:

$$z_{xx} = \frac{2}{A^2}(z_6 b_1 b_2 + z_5 b_1 b_3 + z_4 b_2 b_3)$$

$$z_{yy} = \frac{2}{A^2}(z_6 c_1 c_2 + z_6 c_1 c_3 + z_4 c_2 c_3) \qquad (3.6)$$

$$z_{xy} = \frac{1}{A^2}[z_6(b_1 c_2 + b_2 c_1) + z_5(b_3 c_1 + b_1 c_3) + z_4(b_2 c_3 + b_3 c_2)]$$

Since these values are all constant, substitution into equations (3.3) also yields constant radii of curvature.

In practice the values of z_4, z_5, and z_6 are determined by inputing the global coordinates of the three midside nodes (4), (5), and (6), in addition to those normally inputed for the three primary nodes (1), (2), and (3). By use of appropriate geometrical relations, the quantities in question can then be closely approximated.

The triangular coordinate constants b_i, c_i, and A, can also be calculated through the use of certain trigonometric and geometric relations.

Finally, due to the assumptions of shallow shell theory, surface integrals of various functions taken with respect to the midsurface coordinates may be equated with area integrals of the same functions

taken with respect to the base-plane coordinates [71]. Thus, the simple explicit factorial integrations of natural coordinate polynomials (see Appendix A) may be used in simplifying the calculations of the element stiffness matrix coefficients.

3.2.2 Constitutive and Compatibility Relationships

In general, a shell can be classified as thin if [29]:

$$|\zeta\, z_{xx}|,\ |\zeta\, z_{yy}|,\ |\zeta\, z_{xy}| \ll 1 \tag{3.7}$$

where ζ is the thickness coordinate normal to the midsurface. In the special case of shallow shells, these criteria are more easily satisfied since the second partials of z are small in magnitude, as per relations (3.1).

Since the shell element developed in this thesis is assumed to be both shallow and thin, the following specialized strain-displacement (compatibility) relationships of Novozhilov [56 (p. 97)], or Flugge [30 (p. 419)], are used:

$$\begin{aligned}
\varepsilon_1 &= \frac{\partial u}{\partial x} - w\frac{\partial^2 z}{\partial x^2} - \zeta\frac{\partial^2 w}{\partial x^2} \\
\varepsilon_2 &= \frac{\partial v}{\partial y} - w\frac{\partial^2 z}{\partial y^2} - \zeta\frac{\partial^2 w}{\partial y^2} \\
\varepsilon_{12} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} - 2w\frac{\partial^2 z}{\partial x \partial y} - 2\zeta\frac{\partial^2 w}{\partial x \partial y}
\end{aligned} \tag{3.8}$$

where ε_1, ε_2, and γ_{12} are the strains in the element midsurface, with the subscripts referring to the s_1 and s_2 curvilinear coordinate

subscripts. Note that u, v, and w are midsurface displacements in the s_1, s_2, and s_3 directions, respectively; and not in the x, y, and z directions. This fact makes it possible to avoid the extensive assembly coordinate transformations found in other finite element formulations (i.e. flat shell element approaches).

Due to the fundamental assumptions of thin shell analysis, and more specifically the Kirchhoff-Love Hypothesis, two dimensional stress-strain constitutive relationships (Hooke's law) may be used. These stress-strain relationships are expressed in matrix form as follows [38]:

$$\begin{Bmatrix} \sigma_1 \\ \sigma_2 \\ \tau_{12} \end{Bmatrix} = \frac{E}{1 - \nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix} \begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix} \tag{3.9a}$$

or, simply

$$\underset{(3,1)}{\sigma} = \underset{(3,3)(3,1)}{[D]\ \{\varepsilon\}} \tag{3.9b}$$

Note that equation (3.9) is the same as equation (1.4) in Figure 1.1, if no initial strains are present. For simplicity, these initial strains are assumed absent in this thesis work.

3.2.3 Interpolation (Shape) (Displacement) Functions

Both the tangential displacement fields, u and v, and the normal displacement field, w, are expressed by the Bazeley, et al. [7] nine term, "incomplete" cubic, natural coordinate, polynomials (see Figure 2.1). For example, the normal displacement, w, at any point on the element midsurface may be expressed as follows:

$$\begin{aligned} w = {} & C_1L_1 + C_2L_2 + C_3L_3 + C_4(L_2^2L_1 + \tfrac{1}{2}L_1L_2L_3) \\ & + C_5(L_3^2L_1 + \tfrac{1}{2}L_1L_2L_3) + C_6(L_1^2L_2 + \tfrac{1}{2}L_1L_2L_3) \\ & + C_7(L_3^2L_2 + \tfrac{1}{2}L_1L_2L_3) + C_8(L_1^2L_3 + \tfrac{1}{2}L_1L_2L_3 \\ & + C_9(L_2^2L_3 + \tfrac{1}{2}L_1L_2L_3) \end{aligned} \tag{3.10}$$

where the C_i's are constants. Identical expressions are used to define the u and v tangential displacement fields.

In order to determine the nine constants of each displacement field expression, nine nodal generalized coordinates are defined for each displacement field. Thus, a total of 27 nodal generalized coordinates, 9 for each of the three displacement field, are required. These 27 nodal generalized coordinates, or DOF, are defined as the nodal values of the three displacements u, v, and w, and their first partial derivatives with respect to x and y, at each of the three primary corner nodes (1), (2), and (3) (see Figure 3.1).

The constants C_i are determined in the usual manner by substituting the natural coordinates of these 27 DOF into their respective

first partial derivative displacement field expressions, and then solving for the unknown C_i's. This process amounts to an imposition of boundary conditions and the solution of unknowns [38]. This general procedure is well documented [27 (p. 85), 32 (p. 110), 38, 40 (p. 137), 79 (p. 106)], and thus will not be presented here.

After solving for constants C_i, the displacement fields are explicitly expressed as follows [79 (p. 187)]:

$$\underset{(1,1)}{u} = \underset{(1,9)}{\left\{\begin{matrix} N_A^T \\ N_B^T \\ N_C^T \end{matrix}\right\}^T} \underset{(9,1)}{\left\{\begin{matrix} u_A \\ u_B \\ u_C \end{matrix}\right\}}, \quad \underset{(1,1)}{v} = \underset{(1,9)}{\left\{\begin{matrix} N_A^T \\ N_B^T \\ N_C^T \end{matrix}\right\}^T} \underset{(9,1)}{\left\{\begin{matrix} u_A \\ u_B \\ u_C \end{matrix}\right\}}, \quad \underset{(1,1)}{w} = \underset{(1,9)}{\left\{\begin{matrix} N_A^T \\ N_B^T \\ N_C^T \end{matrix}\right\}^T} \underset{(9,1)}{\left\{\begin{matrix} u_A \\ u_B \\ u_C \end{matrix}\right\}}$$

where

$$\begin{aligned} \underset{(1,3)}{N_A} &= \{N^1 \quad N_x^1 \quad N_y^1\} \\ \underset{(1,3)}{N_B} &= \{N^2 \quad N_x^2 \quad N_y^2\} \\ \underset{(1,3)}{N_C} &= \{N^3 \quad N_x^3 \quad N_y^3\} \end{aligned} \tag{3.12}$$

and the interpolation (shape) function terms are [79 (p. 187)]

$$N^1 = L_1 + L_1^2 L_2 + L_1^2 L_3 - L_2^2 L_1 - L_3^2 L_1$$

$$
\begin{aligned}
N_x^1 &= c_3(L_1^2 L_2 + \tfrac{1}{2} L_1 L_2 L_3) - c_2(L_1^2 L_3 + \tfrac{1}{2} L_1 L_2 L_3) \\
N_y^1 &= b_3(L_1^2 L_2 + \tfrac{1}{2} L_1 L_2 L_3) - b_2(L_1^2 L_3 + \tfrac{1}{2} L_1 L_2 L_3) \\
N^2 &= L_2 + L_2^2 L_3 + L_2^2 L_1 - L_3^2 L_2 - L_1^2 L_2) \\
N_x^2 &= c_1(L_2^2 L_3 + \tfrac{1}{2} L_1 L_2 L_3) - c_3(L_2^2 L_1 + \tfrac{1}{2} L_1 L_2 L_3) \\
N_y^2 &= b_1(L_2^2 L_3 + \tfrac{1}{2} L_1 L_2 L_3) - b_3(L_2^2 L_1 + \tfrac{1}{2} L_1 L_2 L_3) \\
N^3 &= L_3 + L_3^2 L_1 + L_3^2 L_2 - L_1^2 L_3 - L_2^2 L_3 \\
N_x^3 &= c_2(L_3^2 L_1 + \tfrac{1}{2} L_1 L_2 L_3) - c_1(L_3^2 L_2 + \tfrac{1}{2} L_1 L_2 L_3) \\
N_y^3 &= b_2(L_3^2 L_1 + \tfrac{1}{2} L_1 L_2 L_3) - b_1(L_3^2 L_2 + \tfrac{1}{2} L_1 L_2 L_3)
\end{aligned}
\tag{3.13}
$$

and the nodal generalized coordinates are

$$
\underset{(3,1)}{u_A} = \begin{Bmatrix} u^1 \\ u_x^1 \\ -u_y^1 \end{Bmatrix}, \quad \underset{(3,1)}{u_B} = \begin{Bmatrix} u^2 \\ u_x^2 \\ -u_y^2 \end{Bmatrix}, \quad \underset{(3,1)}{u_C} = \begin{Bmatrix} u^3 \\ u_x^3 \\ -u_y^3 \end{Bmatrix}
$$

similarly

$$
\underset{(3,1)}{v_A} = \begin{Bmatrix} v^1 \\ v_x^1 \\ -v_y^1 \end{Bmatrix}, \quad \underset{(3,1)}{v_B} = \begin{Bmatrix} v^2 \\ v_x^2 \\ -v_y^2 \end{Bmatrix}, \quad \underset{(3,1)}{v_C} = \begin{Bmatrix} v^3 \\ v_x^3 \\ -v_y^3 \end{Bmatrix} \tag{3.14}
$$

$$\underset{(3,1)}{w_A} = \begin{Bmatrix} w^1 \\ w_x^1 \\ -w_y^1 \end{Bmatrix}, \quad w_B = \begin{Bmatrix} w^2 \\ w_x^2 \\ -w_y^2 \end{Bmatrix}, \quad w_C = \begin{Bmatrix} w^3 \\ w_x^3 \\ -w_y^3 \end{Bmatrix}$$

The superscripts on the generalized coordinates refer to nodal numbers, while the subscripts x and y refer to partial derivatives with respect to the x and y axes, respectively.

Note that the generalized coordinates which represent partial derivatives with respect to y are actually negative derivatives. These "negative" DOF are arbitrarily used since they lead to a more consistent sign convention in the shape function terms of equations (3.13). Note also that some shape function terms are simply cyclic subscript rotations of other terms, and three pairs of terms differ only by an interchange of "b" or "c". These properties are typical of triangular coordinate expressions, and consequently should be taken advantage of in double-checking derivations.

Equations (3.11) may be combined to define the complete element displacement field discretization, as follows:

$$\underset{(3,1)}{\begin{Bmatrix} u \\ v \\ w \end{Bmatrix}} = \underset{(3,27)}{\begin{bmatrix} N_A & 0 & 0 & N_B & 0 & 0 & N_C & 0 & 0 \\ 0 & N_A & 0 & 0 & N_B & 0 & 0 & N_C & 0 \\ 0 & 0 & N_A & 0 & 0 & N_B & 0 & 0 & N_C \end{bmatrix}} \underset{(27,1)}{\begin{Bmatrix} q_A \\ q_B \\ q_C \end{Bmatrix}} \quad (3.15a)$$

or, simply

$$\underset{(3,1)}{W} = \underset{(3,27)}{[N]}\ \underset{(27,1)}{\{q\}} \tag{3.15b}$$

where

$$\underset{(9,1)}{q_A} = \begin{Bmatrix} u_A \\ v_A \\ w_A \end{Bmatrix}, \quad \underset{(9,1)}{q_B} = \begin{Bmatrix} u_B \\ v_B \\ w_B \end{Bmatrix}, \quad \underset{(9,1)}{q_C} = \begin{Bmatrix} u_C \\ v_C \\ w_C \end{Bmatrix} \tag{3.16}$$

Note that Equation (3.15) is identical to Equation (1.2) of Figure (1.1), where [N] = the element shape function matrix, and {q} = the element nodal generalized coordinate vector.

3.2.4 Compatibility Matrix

Substitution of Equation (3.15) into relations (3.8) yields the following strain-displacement matrix equation that defines the element strain state in terms of the 27 nodal generalized coordinates:

$$\underset{(3,1)}{\begin{Bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \gamma_{12} \end{Bmatrix}} = \underset{(27,1)}{[B_A \quad B_B \quad B_C]} \begin{Bmatrix} q_A \\ q_B \\ q_C \end{Bmatrix} \tag{3.17a}$$

or, simply

$$\underset{(3,1)}{\varepsilon} = \underset{(3,27)}{[B]}\ \underset{(27,1)}{\{q\}} \tag{3.17b}$$

For simplicity, the element compatibility matrix, [B], is taken as the sum of three secondary compatibility matrices, i.e.

$$\underset{(3,27)}{B} = \underset{(3,27)}{B_M} + \underset{(3,27)}{B_K} + \underset{(3,27)}{B_F} \tag{3.18}$$

where

B_M contains the terms involving flat plate membrane actions.

B_K contains the terms involving membrane-flexure coupling actions.

B_F contains the terms involving flat plate flexure (bending) actions.

similarly

$$\begin{aligned} \underset{(3,9)}{B_A} &= \underset{(3,9)}{B_{MA}} + \underset{(3,9)}{B_{KA}} + \underset{(3,9)}{B_{FA}} \\ B_B &= B_{MB} + B_{KB} + B_{FB} \\ B_C &= B_{MC} + B_{KC} + B_{FC} \end{aligned} \tag{3.19}$$

where

$$\underset{(3,9) \text{ or } (3,6)}{B_{MA}} = \begin{bmatrix} \frac{\partial N_A}{\partial x} & 0 & 0 \\ 0 & \frac{\partial N_A}{\partial y} & 0 \\ \frac{\partial N_A}{\partial y} & \frac{\partial N_A}{\partial x} & 0 \end{bmatrix}$$

$$B_{KA}\ (3,9)\text{ or }(3,3) = \begin{bmatrix} 0 & 0 & -N_A \frac{\partial^2 z}{\partial x^2} \\ 0 & 0 & -N_A \frac{\partial^2 z}{\partial y^2} \\ 0 & 0 & -2N_A \frac{\partial^2 z}{\partial x \partial y} \end{bmatrix} \qquad (3.20)$$

$$B_{FA}\ (3,9)\text{ or }(3,3) = \begin{bmatrix} 0 & 0 & -\zeta \frac{\partial^2 N_A}{\partial x^2} \\ 0 & 0 & -\zeta \frac{\partial^2 N_A}{\partial y^2} \\ 0 & 0 & -\zeta \frac{\partial^2 N_A}{\partial x \partial y} \end{bmatrix}$$

and

$$B_{M}\ (3,27)\text{ or }(3,18) = [B_{MA} \quad B_{MB} \quad B_{MC}]$$

$$B_{K}\ (3,27)\text{ or }(3,9) = [B_{KA} \quad B_{KB} \quad B_{KC}] \qquad (3.21)$$

$$B_{F}\ (3,27)\text{ or }(3,9) = [B_{FA} \quad B_{FB} \quad B_{FC}]$$

The other six submatrix terms of equations (3.19) and (3.21) are similarly defined by changing the subscript letter of N to match the second subscript letter of the submatrix in question, in Equations (3.20).

Thus, with the combination of Equations (3.12), (3.13), (3.17), (3.19), (3.20), and (A.9), the compatibility matrix, [B], is completely defined. Note that Equation (3.17) is identical to Equation (1.3) of Figure 1.1.

The notations and submatrices presented in this section greatly aid the stiffness matrix derivations in the next section. Additionally, the many zero terms in Equations (3.20) help to condense (i.e., note the condensed matrix dimensions in Equations (3.20) and (3.21)) some of the following derivations.

3.2.5 Stiffness Matrix

From Equation (1.8) of Figure 1.1, the element stiffness matrix is defined as:

$$\underset{(27,27)}{[k]} = \iiint_V \underset{(27,3)}{[B]^T}\ \underset{(3,3)}{[D]}\ \underset{(3,27)}{[B]}\ dV \tag{3.22}$$

where matrices [B] and [D] are defined in the previous two sections. Substitution of Equation (3.18) into the integrand of Equation (3.22) yields (the matrix brackets are dropped):

$$\begin{aligned} \underset{(27,27)}{B^TDB} &= B_M{}^T DB_M + B_K{}^T DB_K + B_F{}^T DB_F \\ &+ B_K{}^T DB_M + B_M{}^T DB_K + B_M{}^T DB_F \\ &+ B_F{}^T DB_M + B_K{}^T DB_F + B_F{}^T DB_K \end{aligned} \tag{3.23}$$

Thus, the element stiffness (coefficient) matrix terms are determined

by the summation of the terms resulting from the volumetric integrations of the nine right hand side matrix expressions in Equation (3.22). However, this is not quite as formidable a task as it appears to be, since integration of the last four matrix expressions of Equation (3.23) with respect to the thickness coordinate results in null set matrices. Also, the fourth and fifth matrix expressions of Equation (3.23) are transposes of each other. Thus, the volumetric integrals of only the first four matrix expressions of Equation (3.23) need be evaluated in determining the element stiffness matrix. Each of these four expressions will be separately integrated and then assembled to form the complete element stiffness coefficient matrix, [k]. For reference purposes, the stiffness terms resulting from the volumetric integration of the first four matrix expressions in Equation (3.23) are called the membrane terms, the curvature terms, the flexure terms, and the curvature-membrane terms, in that order.

3.2.5.1 Curvature Terms

Due to the many zero terms in the full B_K matrix (see Equations (3.20) and (3.21)), condensed forms of the B_K matrix and the resulting k_K matrix are used in deriving the 81 "curvature" terms of the element stiffness matrix. Thus, from Equations (3.21) - (3.23):

$$\underset{(9,9)}{k_K} = \iiint_V \underset{(9,3)}{B_K^T}\ \underset{(3,3)}{D}\ \underset{(3,9)}{B}\ dV \qquad (3.24)$$

where

$$\underset{(9,9)}{B_K{}^T D B_K} = \begin{bmatrix} B_{KA}{}^T D B_{KA} & B_{KA}{}^T D B_{KB} & (\text{sym.}) \\ (\text{sym.}) & B_{KB}{}^T D B_{KB} & B_{KB}{}^T D B_{KC} \\ B_{KC}{}^T D B_{KA} & (\text{sym.}) & B_{KC}{}^T D B_{KC} \end{bmatrix} \tag{3.25}$$

Only two of the nine partioned matrix expressions in Equation (3.25) need be integrated to fully define all the terms in k_K, since symmetry and rotational subscript properties may be used to "integrate" the other seven expressions of Equation (3.25). The integration of a typical expression in Equation (3.25) yields:

$$\iint_A \int_{-t/2}^{t/2} \underset{(3,3)}{B_{KA}{}^T D B_{KB}} d\zeta dA = \frac{Et}{1-\nu^2} C_K \iint_A \underset{(3,3)}{N_A{}^T N_B} \, dA$$

where

$$C_K = \left(\frac{\partial^2 z}{\partial x^2}\right)^2 + 2\nu\left(\frac{\partial^2 z}{\partial x^2}\right)\left(\frac{\partial^2 z}{\partial y^2}\right) + \left(\frac{\partial^2 z}{\partial y^2}\right)^2 + 2(1-\nu)\left(\frac{\partial^2 z}{\partial x \partial y}\right)^2 \tag{3.26}$$

Note the similarity of subscript letters.

Thus, explicit natural coordinate integrations of expressions similar to the $N_A{}^T N_B$ matrix in Equation (3.26) yields the following explicit form of the "curvature" stiffness matrix:

$$\underset{(9,9)}{k_K} = \frac{AEt}{2520(1-\nu^2)} C_K \begin{bmatrix} k_{KAA} & k_{KAB} & (\text{sym.}) \\ (\text{sym.}) & k_{KBB} & k_{KBC} \\ k_{KCA} & (\text{sym.}) & k_{KCC} \end{bmatrix} \tag{3.27}$$

where

$$k_{KAA}_{(3,3)} = \begin{bmatrix} 484 & 52(c_3-c_2) & 52(b_3-b_2) \\ (sym.) & 7.75(c_3-c_2)^2 + 6c_3c_2 & 7.75(c_3b_3+c_2b_2) -4.75(b_2c_3+b_3c_2) \\ (sym.) & (sym.) & 7.75\,(b_3-b_2)^2 + 6b_2b_3 \end{bmatrix}$$

$$k_{KAB}_{(3,3)} = \begin{bmatrix} 178 & 19c_1-34c_3 & 19b_1-34b_3 \\ 34c_3 -19c_2 & (13c_1c_3-11c_1c_2 +13c_2c_3-25c_3^2)/4 & (13b_1c_3-11b_1c_2 +13c_2b_3-25c_3b_3)/4 \\ 34b_3 -19b_2 & (13c_1b_3-11c_1c_2 +13b_2c_3-25b_3c_3)/4 & (13b_1b_3-11b_1b_2 +13b_2b_3-25b_3^2)/4 \end{bmatrix}$$

The other curvature stiffness submatrices of Equation (3.27) are similarly defined by cyclic rotations of subscript letters and subscript numbers, of the above two submatrices.

3.3.5.2 Flexure Terms

Similarly, condensed forms of the B_F matrix (see Equations (3.20) and (3.21)) and the resulting k_F matrix are used in deriving the 81 "flexure" terms of the element stiffness matrix, i.e.:

$$\underset{(9,9)}{k_F} = \iiint_V \underset{(9,3)}{B_F^T} \; \underset{(3,3)}{D} \; \underset{(3,9)}{B_F} \; dV \tag{3.28}$$

However, the evaluation and explicit integration of this "flexure" stiffness matrix is much more complex than that of the "curvature" stiffness matrix in the preceding section. This is because the "flexure" compatibility matrix, B_F, contains terms that are second partial derivatives of the shape functions.

Thus, an ingenious approach is used to simplify and clarify the derivation of the "flexure" stiffness matrix, k_K. This approach involves expressing the B_F matrix as the product of a constant coefficient matrix, and a triangular coordinate matrix, e.g.:

$$\underset{(3,9)}{B_F} = \frac{-\zeta}{4A^2} \; \underset{(3,9)}{C_F} \; \underset{(9,9)}{B_F'} \tag{3.29}$$

where

$$\underset{(3,9)}{C_F} = \begin{bmatrix} L_1 & L_2 & L_3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L_1 & L_2 & L_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2L_1 & 2L_2 & 2L_3 \end{bmatrix}$$

and

$$\underset{(9,9)}{B_F'} = \underset{(9,3)}{[\, B_{FA}'} \; \underset{(9,3)}{B_{FB}'} \; \underset{(9,3)}{B_{FC}'\,]} \tag{3.30}$$

where

$$B_{FA}'_{(9,3)} = \begin{bmatrix} -2(2b_1^2+b_2^2+b_3^2) & b_2b_3(c_3-c_2)+8Ab_1 & b_2b_3(b_3-b_2) \\ 2b_1(b_1-2b_2) & b_1c_3(2b_1+b_3)-c_2b_1b_3 & b_1b_3(2b_1-b_2+b_3) \\ 2b_1(b_1-2b_3) & b_1b_2c_3-b_1c_2(2b_1+b_2) & b_1b_2(b_3-2b_1-b_2) \\ -2(2c_1^2+c_2^2+c_3^2) & c_2c_3(c_3-c_2) & c_2c_3(b_3-b_2)-8Ac_1 \\ 2c_1(c_1-2c_2) & c_1c_3(2c_1-c_2+c_3) & c_1b_3(2c_1+c_2)-b_2c_1c_3 \\ 2c_1(c_1-2c_3) & c_1c_2(c_3-2c_1-c_2) & c_1c_2b_3-c_1b_1(2c_1+c_2) \\ -2(2c_1b_1+c_2b_2+c_3b_3) & .5b_2c_3(5c_1+2c_3)-.5b_3c_2(5c_1+2c_2) & .5c_2b_3(5b_1+2b_3)-.5c_3b_2(5b_1+2b_2) \\ 2c_1(b_1-b_2)-2c_2b_1 & .5b_1c_3(5c_1+2c_3)+.5b_3c_1(c_3-c_2) & .5c_1b_3(5b_1+2b_3)+.5c_3b_1(b_3-b_2) \\ 2c_1(b_1-b_3)-2c_3b_1 & .5b_2c_1(c_3-c_2)-.5b_1c_2(5c_1+2c_2) & .5c_2b_1(b_3-b_2)-.5c_1b_2(5b_1+2b_2) \end{bmatrix} \quad (3.31)$$

Matrices B_{FB}' and B_{FC}' are obtained from matrix B_{FA}' by a combination of cyclic subscript rotations, and downward cyclic row rotations among rows 1, 2, and 3, exclusively; rows 4, 5, and 5, exclusively; and rows 7, 8, and 9, exclusively.

Substituting Equation (3.29) into (3.28), and integrating yields the following semi-explicit definition of the "flexure" stiffness matrix:

$$k_{F\,(9,9)} = \frac{Et^3}{2304(1-\nu^2)A^3}\, B_F'^{T}\, R_F\, B_F' \tag{3.32}$$

where

$$R_{F\,(9,9)} = \begin{bmatrix} Q_F & \nu Q_F & 0 \\ \nu Q_F & Q_F & 0 \\ 0 & 0 & 2(1-\nu)Q_F \end{bmatrix}$$

and

$$Q_{F\,(3,3)} = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$$

Equation (3.32) also represents the entire element stiffness matrix of the Bazeley, et al. [7] nonconforming flat plate bending element. However, to the author's knowledge, only one publication has presented the semi-explicit derivation in the form of Equation (3.32). This one article is that of Slyper [68], who derives

a similar semi-explicit stiffness matrix for triangular plate elements of variable thickness. Slyper also presents all the terms of the compatibility coefficient matrix B_F' of Equation (3.30). Cheung, et al. [12] also present a semi-explicit derivation for the same plate element with constant thicknesses throughout, but its form is not nearly as transparent or concise as the one presented here, or in Slyper's article.

3.2.5.3 Membrane Terms

Derivation of the "membrane" stiffness matrix initially follows procedures similar to those employed in deriving the "curvature" stiffness matrix (i.e. section 3.2.5.1). Thus, from Equations (3.21) - (3.23):

$$\underset{(18,18)}{k_M} = \iiint_V \underset{(18,3)}{B_M^T} \underset{(3,3)}{D} \underset{(3,18)}{B_M} \, dV \tag{3.33}$$

where

$$\underset{(18,18)}{B_M^T D B_M} = \begin{bmatrix} B_{MA}^T D B_{MA} & B_{MA}^T D B_{MB} & B_{MA}^T D B_{MC} \\ (\text{sym.}) & B_{MB}^T D B_{MB} & B_{MB}^T D B_{MC} \\ (\text{sym.}) & (\text{sym.}) & B_{MC}^T D B_{MC} \end{bmatrix} \tag{3.34}$$

However, the evaluation and explicit integration of the nine matrix expressions in Equation (3.34) is extremely tedious and complex. This is primarily due to the presence of shape function first partial derivatives in the "membrane" compatibility matrix, B_M (see

Equation 3.20). Thus, an approach similar to that employed in the preceeding section is used in deriving the following semi-explicit expressions for the "membrane" stiffness matrix of Equation (3.33).

Let

$$\underset{(3,6)}{B_{MA}} = \frac{1}{2A} \underset{(3,21)}{C_M} \underset{(21,6)}{B_{MA}'}$$

$$B_{MB} = \frac{1}{2A} C_M B_{MB}' \tag{3.35}$$

$$B_{MC} = \frac{1}{2A} C_M B_{MC}'$$

where

$$\underset{(3,21)}{C_M} = \begin{bmatrix} L_M & 0 & 0 \\ 0 & L_M & 0 \\ 0 & 0 & L_M \end{bmatrix} \tag{3.36}$$

and

$$\underset{(1,7)}{L_M} = \{1 \quad L_1^2 \quad L_2^2 \quad L_3^2 \quad L_1L_2 \quad L_2L_3 \quad L_3L_1\}$$

and, where

$$\underset{(21,6)}{B_{MA}'} = \begin{bmatrix} N_{AX} & 0 \\ 0 & N_{AY} \\ N_{AY} & N_{AX} \end{bmatrix}, \underset{(21,6)}{B_{MB}'} = \begin{bmatrix} N_{BX} & 0 \\ 0 & N_{BY} \\ N_{BY} & N_{BX} \end{bmatrix}, \underset{(21,6)}{B_{MC}'} = \begin{bmatrix} N_{CX} & 0 \\ 0 & N_{CY} \\ N_{CY} & N_{CX} \end{bmatrix} \tag{3.37}$$

where

$$N_{AX}_{(7,3)} = \begin{bmatrix} b_1 & 0 & 0 \\ -b_1 & 2A & 0 \\ -b_1 & 0 & 0 \\ -b_1 & 0 & 0 \\ 2(b_1-b_2) & .5b_3(c_3-c_2) + 2b_1c_3 & .5b_3(b_3-b_2) + 2b_1b_3 \\ 0 & .5b_1(c_3-c_2) & .5b_1(b_3-b_2) \\ 2(b_1-b_3) & .5b_2(c_3-c_2) - 2b_1c_2 & .5b_2(b_3-b_2) - 2b_1b_2 \end{bmatrix}$$

and

$$N_{AY}_{(7,3)} = \begin{bmatrix} c_1 & 0 & 0 \\ -c_1 & 0 & -2A \\ -c_1 & 0 & 0 \\ -c_1 & 0 & 0 \\ 2(c_1-c_2) & .5c_3(c_3-c_2) + 2c_1c_3 & .5c_3(b_3-b_2) + .5c_1b_3 \\ 0 & .5c_1(c_3-c_2) & .5c_1(b_3-b_2) \\ 2(c_1-c_3) & .5c_2(c_3-c_2) - 2c_1c_2 & .5c_2(b_3-b_2) - 2c_1b_2 \end{bmatrix}$$

Matrices N_{BX} and N_{CX} are obtained from matrix N_{AX} by a combination of cyclic subscript rotations, and downward cyclic row rotations among rows 2, 3, and 4, exclusively; and rows 5, 6, and 7, exclusively. The first row does not undergo a row rotation. Matrices N_{BY} and N_{CY} are similarly derived from matrix N_{AY}.

Substituting Equations (3.35) and(3.34) into Equation (3.33), and integrating, yields the following semi-explicit definition of the "membrane" stiffness matrix:

$$k_{M_{(18,18)}} = \frac{Et}{720(1-\nu^2)A}$$

$$\times \begin{bmatrix} B_{MA}'^{T} R_M B_{MA}' & B_{MA}'^{T} R_M B_{MB}' & B_{MA}'^{T} R_M B_{MC}' \\ (\text{sym.}) & B_{MB}'^{T} R_M B_{MB}' & B_{MB}'^{T} R_M B_{MC}' \\ (\text{sym.}) & (\text{sym.}) & B_{MC}'^{T} R_M B_{MC}' \end{bmatrix} \tag{3.38}$$

where

$$R_{M_{(21,21)}} = \begin{bmatrix} Q_M & \nu Q_M & 0 \\ \nu Q_M & Q_M & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} Q_M \end{bmatrix}$$

and

$$
Q_{M}\underset{(7,7)}{} = \begin{bmatrix} 180 & 30 & 30 & 30 & 15 & 15 & 15 \\ 30 & 12 & 2 & 2 & 3 & 1 & 3 \\ 30 & 2 & 12 & 2 & 3 & 3 & 1 \\ 30 & 2 & 2 & 12 & 1 & 3 & 3 \\ 15 & 3 & 3 & 1 & 2 & 1 & 1 \\ 15 & 1 & 3 & 3 & 1 & 2 & 1 \\ 15 & 3 & 1 & 3 & 1 & 1 & 2 \end{bmatrix}
$$

3.2.5.4 Curvature-Membrane Terms

Derivation of the "curvature-membrane" stiffness matrix (or its transpose, the "membrane-curvature" stiffness matrix) follows procedures similar to, but more complex than, those used in the last section.

From Equations (3.21) - (3.23):

$$
k_{KM}\underset{(9,18)}{} = \iiint_{V} B_{K}\underset{(9,3)}{}^{T} \; D\underset{(3,3)}{} \; B_{M}\underset{(3,18)}{} \; dV \tag{3.39}
$$

where

$$
B_{K}{}^{T}DB_{M}\underset{(9,18)}{} = \begin{bmatrix} B_{KA}{}^{T}DB_{MA} & B_{KA}{}^{T}DB_{MB} & B_{KA}{}^{T}DB_{MC} \\ B_{KB}{}^{T}DB_{MA} & B_{KB}{}^{T}DB_{MB} & B_{KB}{}^{T}DB_{MC} \\ B_{KC}{}^{T}DB_{MA} & B_{KC}{}^{T}DB_{MB} & B_{KC}{}^{T}DB_{MC} \end{bmatrix} \tag{3.40}
$$

and

$$
\begin{aligned}
B_{KA} &= -C_{KA}\ B_{KA}' \\
(3,3) &\quad (3,18)(18,3) \\
B_{KB} &= -C_{KB}\ B_{KB}' \\
B_{KC} &= -C_{KC}\ B_{KC}'
\end{aligned}
\tag{3.41}
$$

where

$$
C_{KA} = \begin{bmatrix} L_{KA} & 0 & 0 \\ 0 & L_{KA} & 0 \\ 0 & 0 & 2L_{KA} \end{bmatrix}
\tag{3.42}
$$

$$(3,18)$$

and

$$
L_{KA} = \{L_1 \quad L_1^2 L_2 \quad L_1^2 L_3 \quad L_1 L_2^2 \quad L_1 L_3^2 \quad L_1 L_2 L_3\}
$$

$$(1,6)$$

Matrices C_{KB} and C_{KC} are cyclic subscript rotations of matrix C_{KA}.

Also:

$$
B_{KA}' = \begin{bmatrix} z_{xx} \cdot N_{KA} \\ \hline z_{yy} \cdot N_{KA} \\ \hline z_{xy} \cdot N_{KA} \end{bmatrix}
\tag{3.43}
$$

$$(18,3)$$

where

$$N_{\substack{KA \\ (6,3)}} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & c_3 & b_3 \\ 1 & -c_2 & -b_2 \\ -1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & \frac{(c_3-c_2)}{2} & \frac{(b_3-b_2)}{2} \end{bmatrix}$$

Matrices B_{KB}' and B_{KC}' are cyclic subscript rotations of matrix B_{KA}'.

Substituting Equations (3.35), (3.41), and (3.40) into Equation (3.39), yields the following semi-explicit definition of the "curvature-membrane" stiffness matrix:

$$k_{\substack{KM \\ (9,18)}} = -\frac{Et}{30240(1-\nu^2)}$$

$$x \begin{bmatrix} B_{KA}'^{T} R_{KA} B_{MA}' & B_{KA}'^{T} R_{KA} B_{MB}' & B_{KA}'^{T} R_{KA} B_{MC}' \\ B_{KB}'^{T} R_{KB} B_{MA}' & B_{KB}^{T} R_{KB} B_{MB}' & B_{KB}'^{T} R_{KB} B_{MC}' \\ B_{KC}'^{T} R_{KC} B_{MA}' & B_{KC}'^{T} R_{KC} B_{MB}' & B_{KC}'^{T} R_{KC} B_{MC}' \end{bmatrix} \qquad (3.44)$$

where

$$R_{\substack{KA \\ (18,21)}} = \begin{bmatrix} Q_{KA} & \nu Q_{KA} & 0 \\ \nu Q_{KA} & Q_{KA} & 0 \\ 0 & 0 & (1-\nu)Q_{KA} \end{bmatrix}$$

and

$$Q_{KA}_{(6,7)} = \begin{bmatrix} 420 & 126 & 42 & 42 & 42 & 21 & 42 \\ 42 & 12 & 6 & 2 & 6 & 2 & 3 \\ 42 & 12 & 2 & 6 & 3 & 2 & 6 \\ 42 & 6 & 12 & 2 & 6 & 3 & 2 \\ 42 & 6 & 2 & 12 & 2 & 3 & 6 \\ 21 & 3 & 3 & 3 & 2 & 2 & 2 \end{bmatrix}$$

Matrices R_{KB} and R_{KC} are obtained from matrix R_{KA} by left-to-right cyclic column rotations among columns 2, 3, and 4, exclusively; and columns 5, 6, and 7, exclusively; of matrix Q_{KA}.

All the parts of the element stiffness are now determined, since $k_{MK} = k_{KM}^{T}$.

3.2.5.5 Assembly of Stiffness Terms

The following five stiffness matrix components have been formulated:

$$k_{K}_{(9,9)} \quad , \quad k_{F}_{(9,9)} \quad , \quad k_{M}_{(18,18)} \quad , \quad k_{KM}_{(9,18)} \quad , \quad k_{MK}_{(18,9)}$$

Note that the dimensions of the above matrices are not identical. This is because condensed versions of the compatibility matrices were used to simplify the various formulations. Thus the above five matrices must be properly assembled (i.e. expanded) to form the complete 27 x 27 element stiffness matrix, [k]. This assembly (expansion) is very simple, and thus it is graphically described as follows:

$$k_{(27,27)} = \begin{bmatrix} k_{AA} & k_{AB} & k_{AC} \\ k_{BA} & k_{BB} & k_{BC} \\ k_{CA} & k_{CB} & k_{CC} \end{bmatrix} \tag{3.45}$$

where typical terms are

$$k_{AA\ (9,9)} = \left[\begin{array}{c|c} k_{MAA}\ (6,6) & k_{MKAA}\ (6,3) \\ \hline k_{KMAA}\ (3,6) & k_{KAA}\ (3,3) + k_{FAA}\ (3,3) \end{array}\right]$$

and

$$k_{AB\ (9,9)} = \left[\begin{array}{c|c} k_{MAB}\ (6,6) & k_{MKAB}\ (6,3) \\ \hline k_{KMAB}\ (3,6) & k_{KAB}\ (3,3) + k_{FAB}\ (3,3) \end{array}\right]$$

where

$$k_{MAA} = \frac{Et}{720(1-\nu^2)A} B_{MA}{}^{T} R_M B_{MA}$$

$$k_{MAB} = \frac{Et}{720(1-\nu^2)A} B_{MA}{}^{T} R_M B_{MB}$$

etc...

3.2.6 Transformation

Since an element stiffness matrix in terms of generalized surface coordinates is desired, a minor coordinate transformation should be imposed on the previously formulated element stiffness matrix, [k]. This transformation is necessary to that nodal tangential rotation coordinates appear in place of the normal first partial derivative coordinates [71]. The latter of which are actually base-plane, and not surface, coordinates.

By the use of small deflection geometry, it can be shown that:

$$\theta_y = w_x + uz_{xx}$$

and

$$\theta_x = -w_y - vz_{yy}$$

where θ_y and θ_x are rotations about the y and x axes, respectively. Thus, the following coordinate transformation relation can be formulated:

$$\underset{(27,1)}{Q} = \underset{(27,27)}{T}\ \underset{(27,1)}{q} \qquad (3.46)$$

or

$$q = T^T Q \tag{3.47}$$

where

$Q_{(27,1)}$ = the surface generalized coordinate vector, of an element

$q_{(27,1)}$ = the previous part surface and part base-plane generalized coordinate vector, of an element

$T_{(27,27)}$ = the coordinate transformation matrix, defined by placing $(-z_{xx})$ terms at positions (1,8), (10,17), and (19,26); and (z_{yy}) terms at positions (2,9), (11,18), and (20,27); of an otherwise identity matrix.

The previous element force - displacement relations (see Equation (1.5), Figure 1.1) of

$$f_{(27,1)} = k_{(27,27)} q_{(27,1)} \tag{3.48}$$

can now be transformed into totally surface coordinate relations by substitution of (3.47) into Equation (3.48) and pre-multiplying by [T] to preserve symmetry [38, 39]. Thus:

$$F_{(27,1)} = K_{(27,27)} Q_{(27,1)} \tag{3.49}$$

where

$F = T\,f$ = the element nodal applied load vector with respect to surface coordinates.

$K = TkT^T$ = the element stiffness matrix with respect to surface coordinates.

Q = the same as $\{q\}$, except that (w_x) and $(-w_y)$ coordinates are replaced by θ_y and θ_x coordinates, respectively.

Equation (3.49) represents the force-displacement properties of just one element in the element mesh pattern. Thus, superscripts are used to denote this fact, e.g.:

$$F^i = K^i\ Q^i \tag{3.50}$$

where

i = the element identification number (see Figure 3.1) whose properties are defined by the above equation.

3.2.7 Applied Load Vector

The nodal applied load vector, f^i or F^i, of the preceeding section represents applied concentrated loads applied at the element nodes only. Thus, if applied loads are to be applied at other points on the element surface, a statically equivalent nodal load vector should be determined. This is done by using Equation (1.6) or Figure 1.1, where the element applied tractive load vector is:

$$p = \begin{Bmatrix} p_1 \\ p_2 \\ p_3 \end{Bmatrix} \tag{3.51}$$

where p_1, p_2, and p_3 correspond to applied tractive forces in the s_1, s_2, and s_3 coordinate axis directions, respectively. For uniformly distributed applied tractive forces, i.e. p_1, p_2, and p_3 are constants, substitution of Equation (3.51) into (1.6), and integrating, yields:

$$f_p = -\frac{A}{3}\begin{Bmatrix} f_{pA} \\ f_{pB} \\ f_{pC} \end{Bmatrix} \tag{3.52}$$

$$f_{\substack{pA \\ (9,1)}} = \begin{Bmatrix} p_1 \\ p_1(c_3-c_2)/8 \\ p_1(b_3-b_2)/8 \\ p_2 \\ p_2(c_3-c_2)/8 \\ p_2(b_3-b_2)/8 \\ p_3 \\ p_3(c_3-c_2)/8 \\ p_3(b_3-b_2)/8 \end{Bmatrix}$$

Vectors f_{pB} and f_{pC} are cyclic "c" and "b" subscript rotations of f_{pA}. More complex equivalent nodal force vectors can be formulated for concentrated loads by using the Dirac delta function [38].

To be rigorously correct, Equation (3.52) must be pre-multiplied by the transformation matrix [T], of the previous section, to determine the element consistent nodal applied load vector, $[F_p]$, corresponding to the applied tractive loads. However, even with this slight transformation, there is some doubt as to whether these "consistent" load vectors are really statically equivalent to the actual applied load state. The use of shallow shell derivations is believed to be the cause behind this. The author feels that further research is needed in this area of doubly-curved shell element formulations.

To avoid this problem, most investigators use semi-equivalent lumped (or concentrated) nodal force vectors [10, 29], instead of consistent nodal force vectors, in approximating the applied load state due to uniformly distributed loads. Thus, the following relations are used instead of Equation (3.52). Note that these lumped applied nodal force vectors are inherently expressed in terms of surface coordinates, thus:

$$F_p = f_p = -\frac{A}{3}\begin{Bmatrix} F_{pA} \\ F_{pB} \\ F_{pC} \end{Bmatrix} \tag{3.53}$$

where

$$F_{pA} = F_{pB} = F_{pC} = \begin{Bmatrix} p_1 \\ 0 \\ 0 \\ p_2 \\ 0 \\ 0 \\ p_3 \\ 0 \\ 0 \end{Bmatrix}$$

(9,1)

3.3 Assembly

The procedures used in placing the various components of the individual element stiffness matrices and applied nodal force vectors, into the assembled system force-displacement matrix expression, are well documented [11 (p. 81), 27 (pp. 176, 181), 32 (pp. 50, 190), 38, 39, 40 (pp. 43, 45, 49), 62, 79 (pp. 7, 14)]. These assembly procedures are the same as those used in the Direct Stiffness Method of matrix structural analysis. Basically, the assembly procedure involves imposing conditions of compatibility (displacement constraints) and equilibrium (force constraints) at the interelement connection points (nodes) [38]. Because these assembly procedures are so well known, they are not detailed in this work.

After assembly, the system force-displacement stiffness relations are expressed in matrix form, as follows:

$$\underset{(N,1)}{F} = \underset{(N,N)}{K}\ \underset{(N,1)}{Q} \tag{3.54}$$

where N = the total number of generalized coordinates, or DOF, of the system; which, in the present case, is equal to nine times the number of nodes in the finite element discrete model.

3.4 Strains and Stresses

The simultaneous solution of Equation (3.54) yields the response of the discrete model subject to the applied load state. This response is in the form of numerical values of the system generalized coordinates. After these generalized coordinate values are determined, the corresponding element stress and strain states may also be determined by appropriate substitution of generalized coordinate values into Equations (3.17) and (3.9), or (1.3) and (1.4).

3.5 Conclusions

A finite element discrete model for use in linear elastic, static analysis of thin arbitrary shell structures has been formulated. A combination of explicit and semi-explicit expressions are given for the element stiffness matrix of a 27 DOF, nonconforming, doubly-curved triangular, shallow shell element. The following chapter is concerned with the use of this element in a computer aided, linear elastic, static analysis of thin shell structures.

CHAPTER 4

THE COMPUTER PROGRAM

4.1 Introduction

A WATFIV/FORTRAN computer code, that utilizes the finite element model of the previous chapter in a linear elastic, static analysis of "arbitrary" thin shell structures, is presented in this chapter. (For those unfamiliar with the WATFIV or FORTRAN computer programming language, attention is directed to an excellent text on that subject [54].) A brief description of the computer program and its capabilities is given. The program is complete, but due to the extreme complexity of the coding theory, and the time limitations involved, the de-bugging process is not yet totally complete. Consequently, the numerical results as given by the program in its present state are in great error. Nevertheless, the complete program listing is presented in Appendix D, since it incorporates some sophisticated and useful techniques.

4.2 Program Description and Capabilities

The program algorithm basically follows the procedures outlined in the previous chapter. Additionally, many self-documenting comment cards are present in the program listing to further aid the deciphering of the computer coding logic. Thus, correlations between the previously presented theory and the program logic can be readily ascertained. However, some parts of the program do not directly relate to any of the previously formulated theory.

Consequently, these parts of the program are briefly described in following sections of this chapter.

The variable names used in the program are slightly different than those used in Chapter 3. Thus, a glossary of the program variables and their definitions is given at the beginning of the computer listing. Additionally, in order to facilitate the use of the program, Appendices B and C describe the particulars of the input data and program output, respectively.

In order to present a more readable, reliable, and organized computer code, many subroutines or subprograms are used in the program. All but one of these subroutines are interconnected at the main program level through the use of COMMON statements [54]. The flow chart of Figure 4.1 shows the order of these subroutines as called by the main program.

The programming logic in many of these subroutines is extremely complex. Some of these programming complexities are attributed to the intricacies involved in defining the interconnectional and geometric properties and configurations of the individual elements. Also, as shown in the previous chapter, the calculation of the element stiffness matrices is extremely complex. The many cyclic rotations of array arguments (subscripts), rows, and/or columns, further compounds the programming logic complexities. Finally, the specialized half-bandwidth form of the system stiffness matrix causes additional programming complications in the assembly (ASEMBL) and solution (BNDSLV) subroutines.

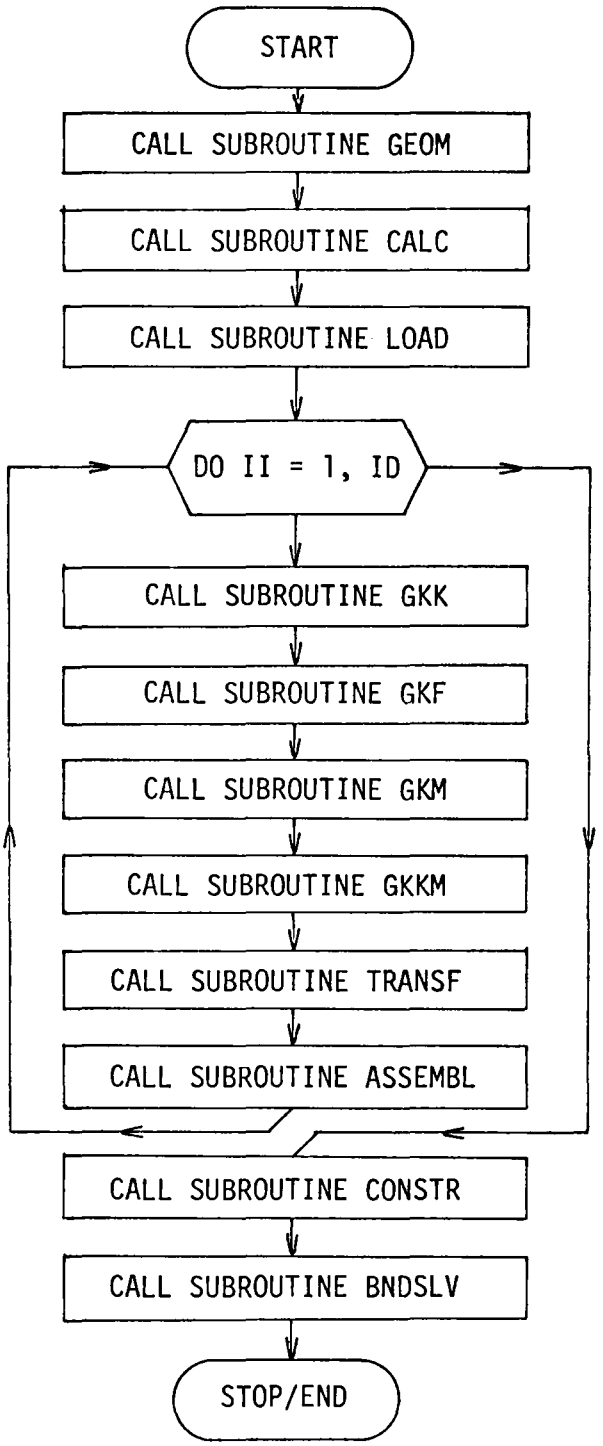


Figure 4.1. Main program flow chart.

In the interests of simplicity, practicality, and economy; several assumptions are made in the program with regard to the shell structure type, geometry, loading, and boundary conditions. These simplifications limit the versatility of the present program in analyzing completely aribtrary thin shell structures. However, these simplifications, in conjunction with some specialized programming, allow large savings to be realized in computer storage and execution time costs. These various simplifications and specialized techniques are also briefly explained in the following sections of this chapter.

It should be noted that all real number variables are expressed in DOUBLE PRECISION [54]. This is necessary to reduce the significance of round-off errors in the solution process. Also, note that, for convenience, the declared units used in the program are inches, pounds, and radians. However, in actuality, any consistent set of units may be freely used, since no unit conversions occur in the program algorithm.

4.2.1 Definition of the Model Geometry

Due to the difficulties frequently encountered in defining the geometry of arbitrary shell surfaces, it was decided to limit the range of the types of thin shell structures that could be most easily analyzed by the computer program. Consequently, the computer program is most effective in analyses of flat plates, right circular cylindrical shells, shallow hyperbolic paraboloid shells, and shallow elliptic paraboloid shells. However,

the amount of input data required to completely define the geometric properties and characteristics of the finite element models used to analyze even these specialized shell structures, is frequently very lengthy and very tedious to prepare.

Consequently, an automated mesh generation scheme is used in the present program which eliminates practically all of the geometric input data. This scheme takes advantage of the mathematical equations that define the middle surfaces of the four specialized shell types as listed in the first paragraph. However, in order to implement this mesh generation scheme, certain additional restrictions are placed on the finite element model geometry. These restrictions involve the shape of the analysis domain, and the configuration of the mesh pattern.

Basically, the analysis domain is limited to a rectangular shape as projected on the global "base-plane". However, in the case of right circular cylindrical shells, the analysis domain is restricted to a "rectangular" cylindrical shell section. Additionally, the mesh pattern in all cases is restricted to three sets of uniformly spaced parallel lines as projected onto the rectangular domain. Moreover, two of these sets of parallel lines must be parallel to the sides of the rectangular domain. Figures 4.2 - 4.5 graphically display these mesh pattern configurations on the four main structure types that can be more easily analyzed by the computer program. These same four shell structures are analyzed by other investigators [10, 16, 18, 24, 28, 29, 71]. One other limitation is imposed on

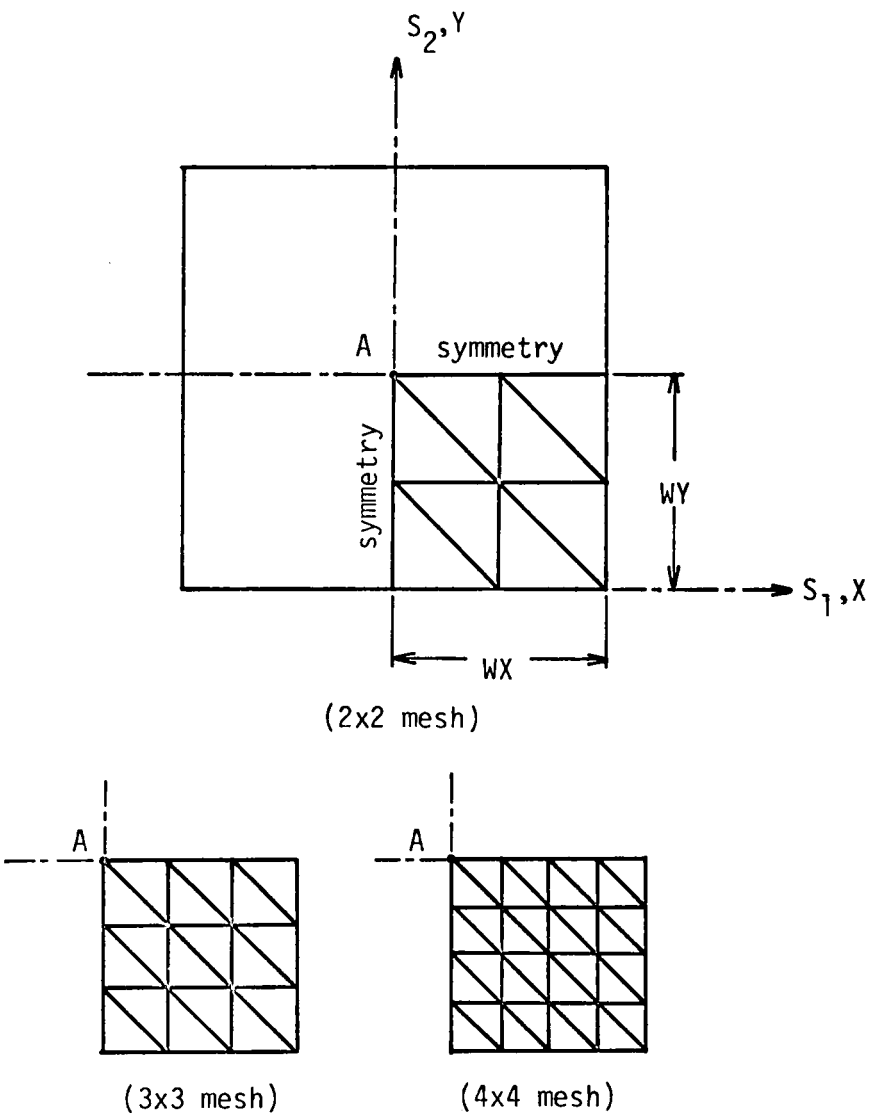
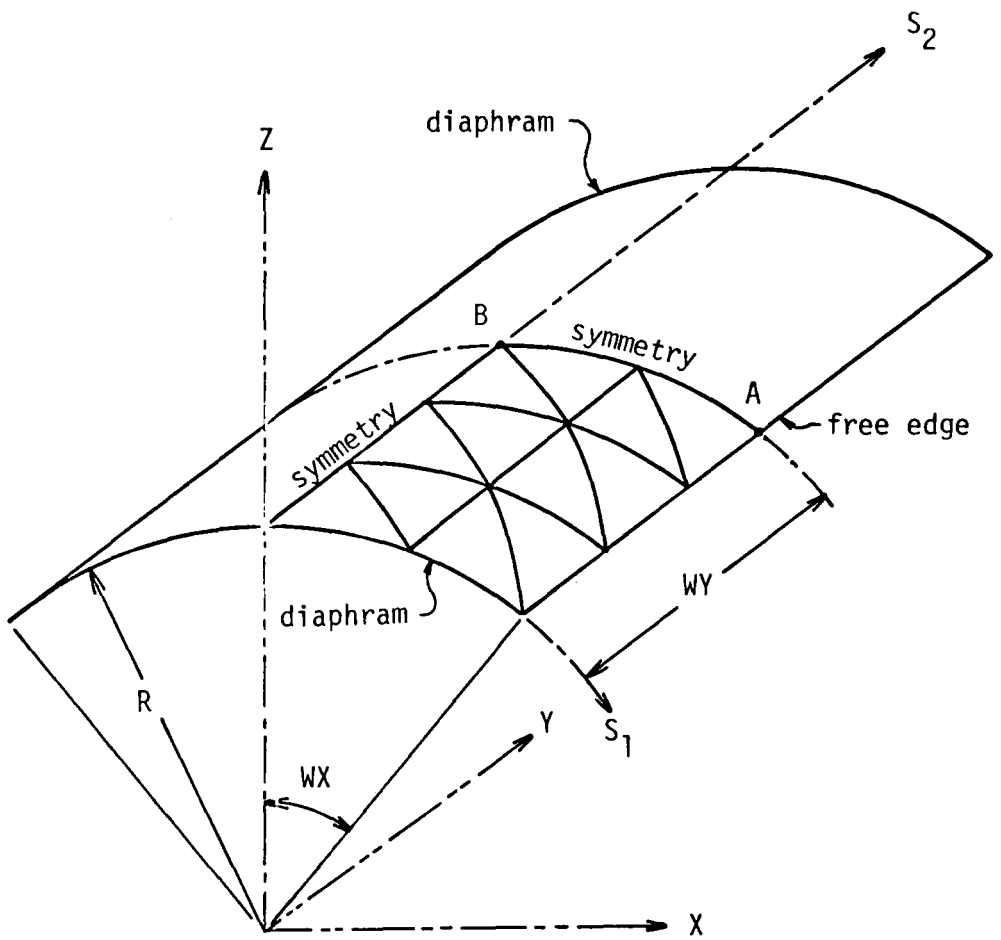
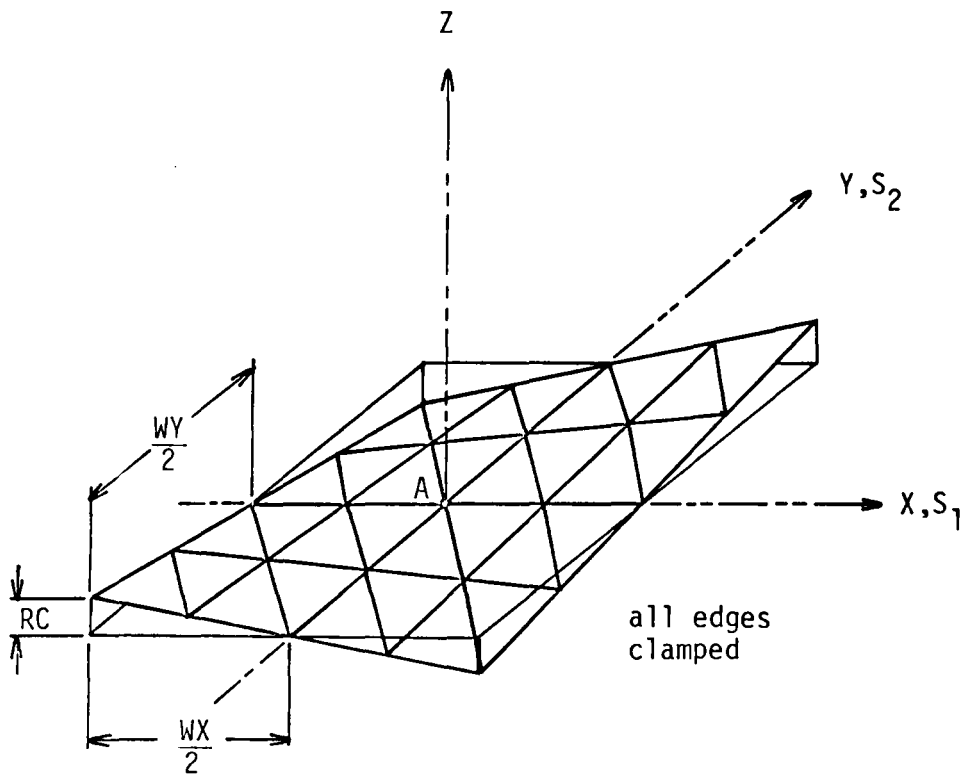


Figure 4.2 Flat plate problem with alternate mesh patterns.



(2x3 mesh)

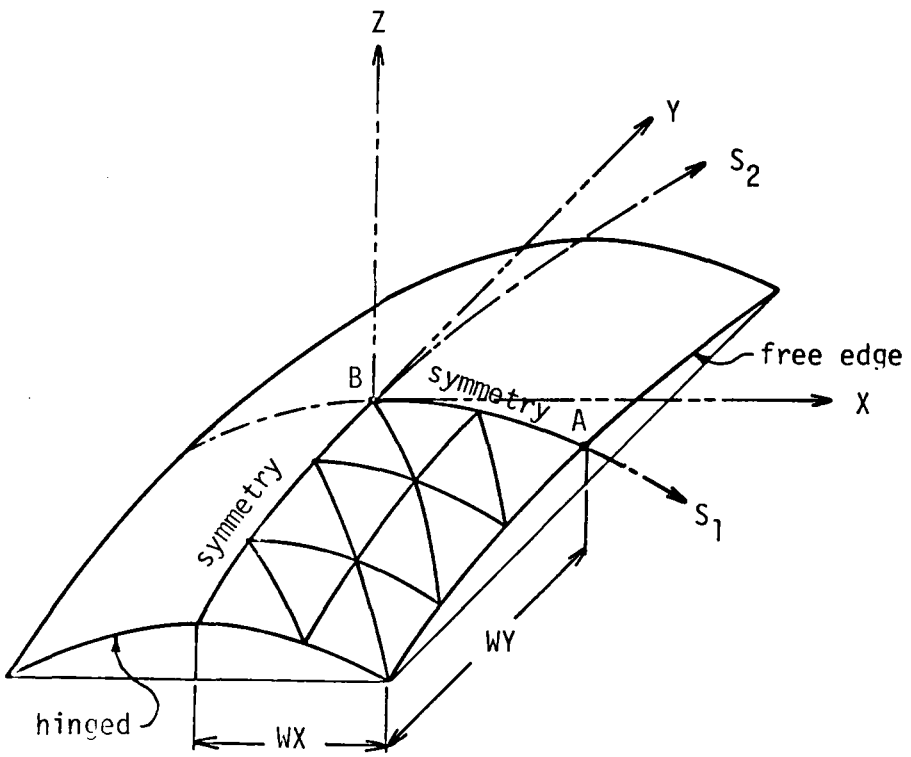
Figure 4.3 Right circular cylindrical shell problem.



(4x4 mesh)

$$Z(X,Y) = XY/RC$$

Figure 4.4 Clamped hyperbolic paraboloid shallow shell problem.



(2x3 mesh)

$$Z = \frac{X^2}{RA} + \frac{Y^2}{RB}$$

Figure 4.5 Elliptic paraboloid shallow shell problem.

the cylindrical shell structure in the present program. This is the limitation that WX does not exceed ninety degrees. This restriction will be eliminated in future versions of the program.

The automated mesh generation scheme is contained in subroutine GEOM. Using a bare minimum of input data, subroutine GEOM generates the identification numbers of all nodes (both primary and secondary) and elements, the global cartesian coordinates of each node point, and the nodal incidence properties of each element. Figure 4.6 shows a typical node and element numbering scheme as generated by subroutine GEOM. In addition to the previously listed items, subroutine GEOM performs other functions. These other functions are revealed in latter discussions.

Subroutine CALC finishes the geometric definition of the model by using the quantities generated in subroutine GEOM to calculate the individual element curvatures, base-plane areas, and some of the quantities used in the triangular coordinate transformation matrix (see Appendix A). In order to calculate these quantities, the computer must know whether the element has Type A or Type B orientation (see Figure 4.7). Consequently, the element orientation types are also defined in subroutine GEOM.

Subroutine CALC also inputs the structural material property constants, namely: Poisson's ratio, Young's modulus of elasticity, and the shell thickness. It is assumed that the material is homogeneous and isotropic. Furthermore, for simplicity, the three

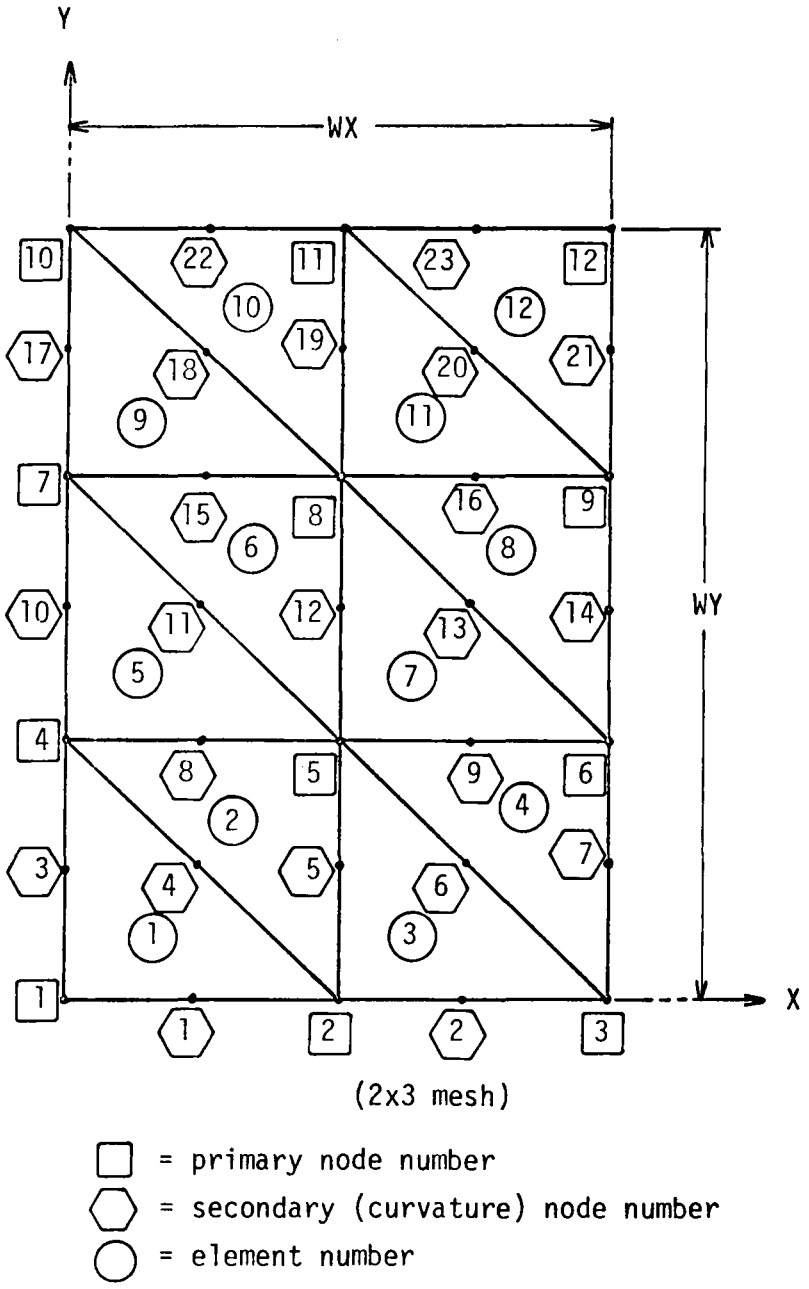
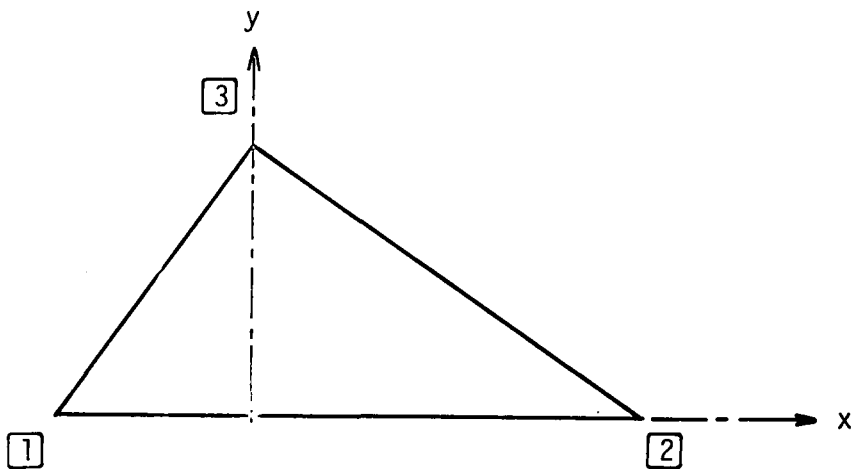
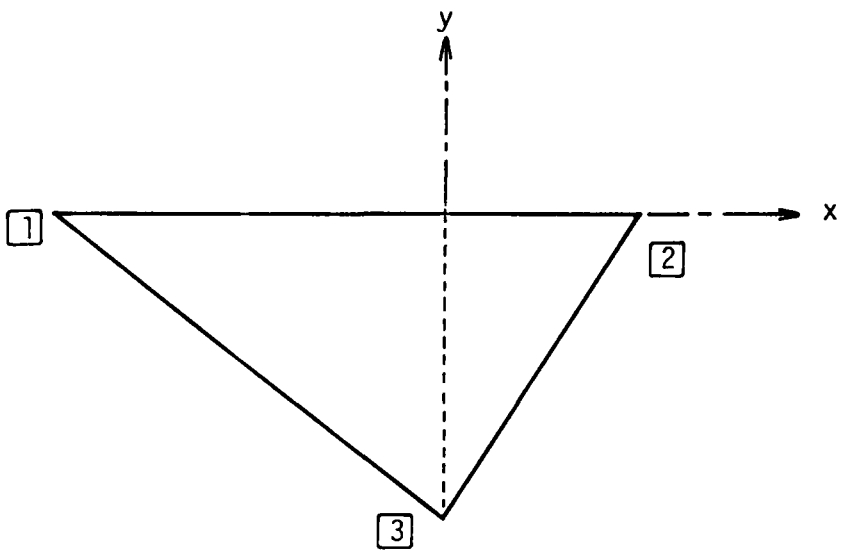


Figure 4.6 A typical node and element numbering scheme as generated by subroutine GEOM.



TYPE A



TYPE B

Figure 4.7 Element orientation types.

previously mentioned quantities are assumed constant in the present computer program. Variations of these quantities between elements will be possible in future versions of the program.

Note that there are actually only two elements with uniquely different geometric and stiffness properties in flat plate or cylindrical shell problems. The computer program takes advantage of these identical element properties in avoiding the execution of many unnecessary duplicate calculations. In order to recognize these identical elements the computer refers to the inputed integer constant "ID" and the fifth terms of the INC(NE,5) integer array. If the shell structure is one of the four previously mentioned types, then the fifth terms of the INC (NE,5) matrix are also automatically generated in subroutine GEOM. If not, then these quantities must be inputed.

4.2.2 Boundary Conditions

Only four integer quantities are inputed to completely define the boundary constraints of the model. These four inputed quantities, IB1, IB2, IB3, and IB4, represent the shell edge boundary conditions corresponding to the X = 0, X = WX, Y = 0, and Y = WY sides of the rectangular analysis domain, respectively. Using these four quantities, subroutine GEOM automatically constrains the appropriate boundary node generalized coordinates, or DOF, to exhibit zero motions. The five different edge boundary conditions that may be used are: free edge, knive edge (diaphram), hinged (simply supported) edge,

symmetrical "edge", and clamped (fixed) edge. Note that the symmetry properties of many types of shell structures may be used to substantially reduce the size of the analysis domain.

4.2.3 Applied Loads

Again in the interests of simplicity, only limited forms of applied loads are used in the present program. The possible loading conditions are restricted to uniformly distributed loads in the directions of the S_1, S_2, and S_3 global curilinear (surface) coordinate axes. However, an additional loading condition corresponding to a self-weight "gravity" load may be used in the special case of right circular cylindrical shell sections. This "gravity" load acts in the negative Z-axis direction (see Figure 4.3).

The applied nodal force vectors corresponding to these applied distributed load states are calculated in subroutine LOAD. The semi-equivalent lumped approach (as discussed in section 3.2.7), rather than the consistent approach, is used to calculate these applied load vectors. It should be pointed out that the use of this lumped approach will not in itself prevent convergence, but may slightly affect the rate of convergence.

4.2.4 Stiffness Matrix Storage

The dimensions of the complete assembled system stiffness matrix are frequently very large in finite element plate or thin shell analysis. This is due to the large numbers of DOF per node, and the two dimensional configuration of the mesh pattern. The number of

computer storage locations (bytes) required to store this complete stiffness matrix is further inflated (in fact, doubled) by the use of DOUBLE PRECISION [54] real number variables. Consequently, the total number of bytes required to completely store the system stiffness matrix may frequently exceed the capacity of the computer. However, by recognizing and taking advantage of certain specialized features of the system stiffness matrix, it is possible to significantly reduce the amount of storage space actually required in the analysis [27 (p. 19), 79 (p. 451)].

The inherent symmetry of the structural system stiffness matrix makes it possible to restrict solution operations to the upper diagonal area (triangle) of the matrix [79 (p. 451)]. Thus, only slightly over one-half of the complete square system stiffness matrix need actually be stored.

Further economies in storage costs may be gained by recognizing the sparsely populated nature of the stiffness matrix. The non-zero terms in finite element stiffness matrices are frequently grouped fairly close to the main diagonal of the matrix. However, this banded nature of the stiffness matrix can only be taken advantage of if the particular soltuion method employed does not require the interchange of rows or columns (i.e. pivoting) [27 (p. 19)]. Fortunately, pivoting is not required due to the positive definite property of the structural stiffness matrix [27 (p. 19), 79 (p. 31)]. Consequently, large savings in computer storage allocation costs may be realized by storing only the upper triangle terms that lie within the banded region of the stiffness matrix.

This is exactly the procedure used in the present program (i.e. in subroutine ASEMBL). Unfortunately, in order to fully realize these savings in storage costs, the arrangement of the stiffness matrix terms must be altered. Figure 4.8 graphically displays this transformation from a banded square stiffness matrix, to a highly compact rectangular stiffness matrix. Note that the width of the rectangular stiffness matrix is equal to the bandwidth plus one, divided by two. This quantity is called the semi-bandwidth or half-bandwidth.

While this storage transformation introduces annoying complexities in the assembly and solution techniques, large savings are gained in the total number of kilobytes (1000 bytes) required to store the stiffness properties of the structural model. Table 4.1 quantitatively demonstrates some of the savings involved.

4.2.5 Solution Process

One of two basic methods is generally employed in solving the large sets of force-displacement simultaneous equations that occur in finite element analyses [27 (p. 19), 53, 79 (p. 451)]. These two basic solution methods are the direct (or elimination) approach, and the iterative (or iteration) approach. Millward presents a brief explanation and discussion of these two methods [53].

Since the elimination techniques are generally more reliable and versatile [27 (p. 19)], this basic solution method is used in the present computer program. More specifically, a modified

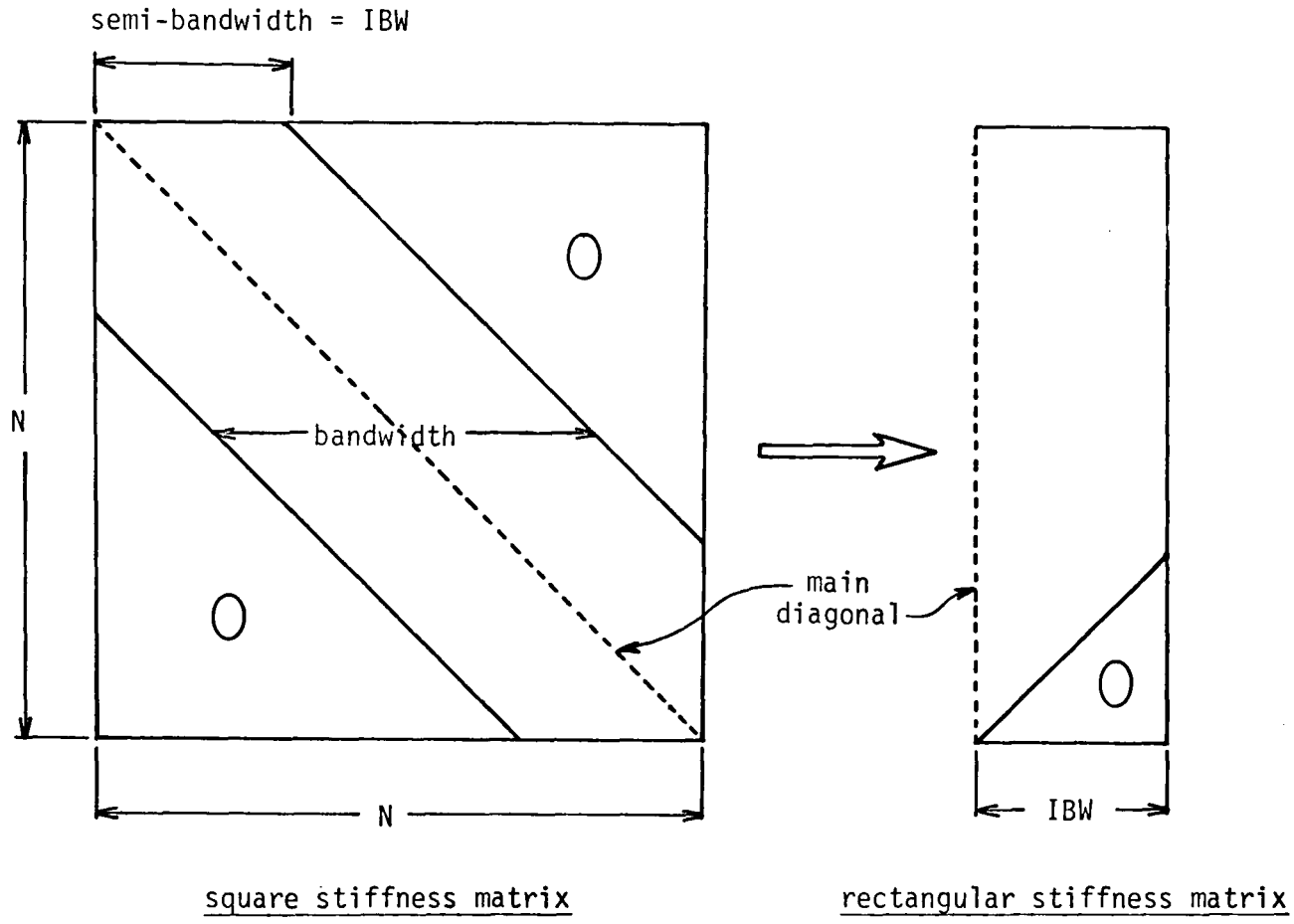


Figure 4.8 Symmetric banded system stiffness matrix transformation. (N = the total number of generalized coordinates of the system).

Table 4.1 Comparison of stiffness matrix storage requirements.

Mesh Size	N (No. of DOF)	IBW (semi-Bandwidth)	No. of Kilobytes Required to Store Complete Square Stiffness Matrix	No. of Kilobytes Required to Store Compact Rect. Stiffness matrix	Savings (%)
2x2	81	36	52.488	23.328	55.6
2x3	108	36	93.312	31.104	66.7
3x3	162	45	209.952	58.320	72.2
4x4	225	54	405.000	97.200	76.0
4x5	270	54	583.200	116.640	80.0
4x6	315	54	793.800	136.080	82.9
5x5	324	63	839.808	163.296	80.6
6x6	441	72	1555.848	254.016	83.7
6x9	630	72	3175.200	362.880	88.6
7x7	576	81	2654.208	373.248	85.9
8x8	729	90	4251.528	524.880	87.7
8x12	1053	90	8870.472	758.160	91.5
9x9	900	99	6480.000	712.800	89.0
10x10	1089	108	9487.368	940.896	90.1

(for use in a banded symmetric stiffness matrix) Gauss elimination solution procedure is used. Descriptions and/or comparisons of the various elimination techniques are found elsewhere [27 (p. 21), 31, 41, 46 (Chap. 7), 52, 53, 76 (pp. 14-16), 79 (p. 451)]. Melosh and Bamford note that no solution method, using rational operations, can take fewer operations than Gauss elimination when a dense coefficient matrix is present [52].

Subroutine BNDSLV uses this modified Gauss elimination technique to solve for the displacements corresponding to the global surface generalized coordinates. In the interests of simplicity, the element strains and stresses are not calculated in the present program.

4.3 Numerical Results

Because of the yet unlocated errors in the computer coding, the displacement values as solved by subroutine BNDSLV are in great error. Since these computed values are not very meaningful, they are not reported here. however, the "correct" results, as found by other investigators using similar methods, are reported in the following sections. These "correct" displacement values are given as references for future program de-bugging procedures. Refer to Figures 4.2 - 4.5 in interpreting the following results.

4.3.1 Square Flat Plate

Results for this problem are given in References [10], [24], and [29]. The vertical (Z-axis) deflection of point A is defined as:

$$W_A = \alpha \frac{12(1-PR^2)\,PZ\,(2WX)^4}{(EL)(TH)^3} \; ; \quad WX = WY$$

where α = +.00406 For simply supported boundary conditions.

and α = +.00126 For clamped boundary conditions.

4.3.2 Right Circular Cylindrical Section

Results for this problem are given in the many articles [10, 16, 18, 24, 28, 29, 71]. The problem parameters are:

WX = 40°

WY = R = 300"

TH = 3.0"

PR = 0.0

EL = 3×10^6 psi

W (self-weight) = 0.625 psi

The reported values of the vertical (Z-axis) deflection at point A range from W_A = -3.45" to W_A = -3.78". The values of the curvilinear displacements at point A are also given in Reference [24]. These are w_A = -4.099", a u_A = +0.8761".

The reported values of the vertical deflection at point B range from W_B = +0.524" to W_B = +0.552".

4.3.3 Shallow Hyperbolic Paraboloid

Results for this problem are given in References [16], [28], and [29]. The problem parameters are:

WX = WY = 12.92"

RC = 1.304"

TH = 0.25"

PR = 0.39

EL = 5 x 10^5 psi

PZ = -1.0 psi

The reported values of the vertical (Z-axis) deflection at point A range from W_A = -.00868" to W_A = -.00895".

4.3.4 Shallow Elliptic Paraboloid

Results for this problem are given in Reference [29]. The problem parameters are:

WX = 650 cm

WY = 856.5 cm

RA = 2312 cm

RB = 3920 cm

TH = 7.0 cm

PR = 0.0

The vertical (Z-axis) deflections at points A and B are defined as:

$$W_{A \text{ or } B} = \frac{12(1-PR^2)PZ(2WX)^4}{(EL)(TH)^3} (\alpha_{A \text{ or } B})$$

where the values of α_A range from α_A = +2700 to α_A = +2981, and the values of α_B range from α_B = +600 to α_B = +641.

CHAPTER 5

SUMMARY AND CONCLUSIONS

In performing this thesis study, the author has reached the overriding conclusion that the finite element analysis of thin arbitrary shell structures by the use of doubly-curved finite elements is exceedingly complex and laborious. These complexities exist despite the many simplifying assumptions that are frequently used in the analysis. Complexities occur in both the theoretical formulation of the finite element model, and in the implementation of this model into a practical and efficient shell analysis computer program. In spite of these difficulties, the author feels that many useful accomplishments are presented in this thesis.

First of all, a fairly extensive discussion on the convergence properties of plate and shell finite element models is presented in Chapter 1. It is hoped that this discussion will aid the student in more fully understanding the various convergence properties and criteria involved in plate or shell finite element analyses.

Perhaps the most valuable and useful part of this thesis is the extensive discussions concerning the numerous considerations involved in selecting a general shell finite element. These discussions of Chapter 2 should be of great benefit to those students wishing to conduct thin shell finite element analyses.

The third accomplishment of this thesis research is the complete formulation and presentation of a 27 DOF, nonconforming,

"incomplete" cubic-cubic, doubly-curved, triangular shallow shell element. To the author's knowledge, no previous publication has attempted to formulate or use this particular shell element. The various operations involved in formulating this element are described in detail in Chapter 3. Additionally, the complete element stiffness matrix is given in practical semi-explicit form. It is hoped that the availability of this semi-explicit stiffness matrix will stimulate further research in the use of this element in the analysis of thin shell structures.

The development of a practical and efficient WATFIV/FORTRAN computer program that utilizes the previously formulated element in a linear elastic, static, finite element analysis of thin arbitrary shell structures is the final objective of this thesis study. Although the program still contains some errors, this goal has basically been accomplished. Several innovative techniques are incorporated in the computer coding. After the de-bugging processes are completed, more versatile versions of the present program can be developed. Although meaningful numerical results are not yet available to properly evaluate the performance of this newly derived shell element, it can be reasoned that the rate of convergence of the solutions obtained by using this element will be at least slightly better than the solutions obtained by using similar linear-cubic doubly-curved elements, or comparable flat shell elements.

As a final note, it should be pointed out that some type of automatic structure generation scheme is practically mandatory in computer programs that analyse shell structures. Otherwise, the vast quantity of input data required to completely define the structural problem parameters may effectively render the program unless in practice. Perhaps further research should be conducted in this area.

REFERENCES

The following abbreviations are used for some commonly cited references:

Abbreviation	Explanation
AIAA J.	Journal of the American Institute of Aeronautics and Astronautics, New York.
CMAME	Computer Methods in Applied Mechanics and Engineering, North-Holland Publishing Company, Amsterdam.
IJCS	An International Journal - Computers & Structures, Pergamon Press, New York.
IJNME	International Journal for Numerical Methods in Engineering, John Wiley & Sons, New York.
IJSS	International Journal of Solids and Structures, Pergamon Press, New York.
ASCE-EM	Journal of the Engineering Mechanics Division - Proceedings of the American Society of Civil Engineers, New York.
First Conf.	Proceedings of the (First) Conference (Oct. 26-28, 1965) on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, Nov. 1966, AD646300.
Second Conf.	Proceedings of the Second Conference (Oct. 15-17, 1968) on Matrix Methods in Structural Mechanics, Wright-Patterson Air Force Base, Ohio, Dec. 1969, AFFDL-TR-68-150.
Third Conf.	Proceedings of the Third Conference (Oct. 19-21, 1971) on Matrix Methods in Structural Mechanics, Weight-Patterson Air Force Base, Ohio, Dec. 1973, AFFDL-TR-71-160.
Symp. FEM	Proceedings of the Symposium (Nov. 13-14, 1969) on Application of Finite Element Methods in Civil Engineering, ASCE - Vanderbilt University, Nashville, Tenn., 1969.

1. Allman, D. J., "A Simple Cubic Displacement Element for Plate Bending," IJNME, Vol. 10, No. 2, 1976, pp. 263-281.

2. Argyris, J. H., Buck, K. E., Scharpf, D. W., Hilber, H. M., and Marczek, G., "Some New Elements for the Matrix Displacement Method," Second Conf., pp. 333-398.

3. Argyris, J. H., and Scharpf, D. W., "Matrix Displacement Analysis of Shells and Plates Including Transverse Shear Strain Effects," CMAME, Vol. 1, No. 1, 1972, pp. 81-139.

4. Atluri, S., and Pian, T. H. H., "Theoretical Formulation of Finite-Element Methods in Linear Elastic Analysis of General Shells," Journal of Structural Mechanics, Marcel Dekker, New York, Vol. 1, No. 1, 1972, pp. 1-41.

5. Backlund, J., and Wennerstrom, H., "Finite Element Analysis of Elasto-Plastic Shells," IJNME, Vol. 8, No. 2, 1974, pp. 415-424.

6. Bandyopadhyay, M., "A Finite Element Analysis of Plate and Shell Structures," M. Sc. Thesis, Virginia Polytechnic Institute and State University, Blacksburg, Va., March 1976.

7. Bazeley, G. P., Cheung, Y. K., Irons, B. M., and Zienkiewicz, O. C., "Triangular Elements in Plate Bending - Conforming and Non-Conforming Solutions," First Conf., pp. 547-576.

8. Bell, K., "A Refined Triangular Plate Bending Finite Element," IJNME, Vol. 1, No. 1, 1969, pp. 101-122.

9. Bogner, F. K., Fox. R. L., and Schmit, J. R., The Generation of Interelement Compatible Stiffness and Mass Matrices by the Use of Interpolation Formulas," First Conf., pp. 397-444.

10. Bonnes, G., Dhatt, G., Giroux, Y. M., and Robichaud, L. P. A., "Curved Triangular Elements for the Analysis of Shells," Second Conf., pp. 617-639.

11. Brebbia, C. A., and Connor, J. J., Fundamentals of Finite Element Techniques, Halsted Press, John Wiley & Sons, New York, 1974.

12. Cheung, Y. P., King, I. P., and Zienkiewicz, O. C., "Slab Bridges with Arbitrary Shape and Support Conditions - A General Method of Analysis Based on Finite Elements," Proceedings of the Institution of Civil Engineers, William Clowes & Sons, London, Vol. 40, 1968, pp. 9-36.

13. Chu, T. C., and Schnobrich, "Finite Element Analysis of Translational Shells," IJCS, Vol. 2, 1972, pp. 197-222.

14. Clough, R. W., and Fellipa, C. A., "A Refined Quadrilateral Element for Analysis of Plate Bending," Second Conf., pp. 399-440.

15. Clough, R. W., and Johnson, C. P., "A Finite Element Approximation for the Analysis of Thin Shells," IJSS, Vol. 4, No. 1, 1968, pp. 43-60.

16. Clough, R. W., and Johnson, C. P., "Finite Element Analysis of Arbitrary Thin Shells," Concrete Thin Shells - ACI Publication SP-28, ACI, Detroit, Mich., 1971, pp. 333-363.

17. Clough, R. W., and Tocher, J. L., "Finite Element Stiffness Matrices for Analysis of Plate Bending," First Conf., pp. 515-545.

18. Clough, R. W., and Wilson, E. L., "Dynamic Finite Element Analysis of Arbitrary Thin Shells," IJCS, Vol. 1, No. 1/2, 1971, pp. 33-56.

19. Conner, J. J. Jr., and Brebbia, C. A., "Stiffness Matrix for Shallow Rectangular Shell Element," ASCE-EM, Vol. 93, No. 5, 1967.

20. Cook, R. D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, New York, 1974.

21. Cook, R. D., "Two Hybrid Elements for Analysis of Thick, Thin and Sandwich Plates," IJNME, Vol. 5, No. 2, 1972, pp. 277-288.

22. Cowper, G. R., "Variational Procedures and Convergence of Finite Element Methods," Symp. FEM.

23. Cowper, G. R., Lindberg, G. M., and Olson, M. D., "Comparison of Two High-Precision Triangular Finite Elements for Arbitrary Deep Shells," Third Conf. pp. 277-304.

24. Cowper, G. W., Lindberg, G. M., and Olson, M. D., "A Shallow Shell Finite Element of Triangular Shape," IJSS, Vol. 6, No. 8, 1970, pp. 113-1156.

25. Dawe, D. J., "High-Order Triangular Finite Element for Shell Analysis," IJSS, Vol. 11, No. 10, 1975, pp. 1097-1110.

26. Dawe, D. J., "Shell Analysis Using a Simple Facet Element," The Journal of Strain Analysis, The Institution of Mechanical Engineers, Westminster, Vol. 7, No. 4, 1972, pp. 266-270.

27. Desai, C. S., and Abel, J. F., Introduction to the Finite Element Method, Van Nostrand Reinhold Co., New York, 1972.

28. Dhatt, G. S., "An Efficient Triangular Shell Element," AIAA J., Vol. 8, No. 11, 1970, pp. 2100-2102.

29. Dhatt, G. S., "Numerical Analysis of Thin Shells by Curved Triangular Elements Based on Discrete Kirchhoff Hypothesis," Symp. FEM, pp. 255-278.

30. Flugge, W., Stresses in Shells, Springer-Verlag, New York, 2nd Edition, 1973.

31. von Fuchs, G., Roy, J. R., and Schrem, E., "Hypermatrix Solution of Large Sets of Symmetric Positive-Definite Linear Equations," CMANE, Vol. 1, No. 2, 1972, pp. 197-216.

32. Gallagher, R. H., Finite Element Analysis: Fundamentals, Prentice-Hall, Englewood Cliffs, New Jersey, 1975.

33. Gallagher, R. H., "Analysis of Plate and Shell Structures," Symp. FEM, pp. 155-206.

34. Garnet, H., and Crouzet-Pascal, J., "Doubly Curved Triangular Finite Elements for Shells of Arbitrary Shape," Grumman Research Dept. Report RE-453, Bethpage, New York, April 1973.

35. Green, B. E., Jones, R. E., McLay, R. W., and Strome, D. R., "Dynamic Analysis of Shells Using Doubly-Curved Finite Elements," Second Conf., pp. 185-212.

36. Holzer, S. M., "Lecture Notes on CE 5202 - Advanced Structural Analysis (Shell Theory)," VPI & SU, Blacksburg, Va., Winter Quarter, 1975-76.

37. Holzer, S. M., "Lecture Notes on CE 6020 - Dynamics of Structures," VPI & SU, Blacksburg, Va., Spring Quarter, 1976.

38. Holzer, S. M., "Lecture Notes on CE 6030 - Finite Element Analysis of Structures," VPI & SU, Blacksburg, Va., Spring Qrt., 1976.

39. Holzer, S. M., "Lecture Notes on CE 4001, 4002 - Matrix Structural Analysis," VPI & SU, Blacksburg, Va., Fall & Winter Qrts., 1974-1975.

40. Huebner, K. H., The Finite Element Method for Engineers, John Wiley & Sons, New York, 1975.

41. Irons, B. M., "A Frontal Solution Program for Finite Element Analysis," IJNME, Vol. 2, 1970, pp. 5-32.

42. Irons, B. M., and Barlow, J., "Comments on Matrices for the Direct Stiffness Method," AIAA J., Vol. 2, No. 2, 1964, p. 403.

43. Irons, B. M., and Draper, K. J., "Inadequacy of Nodal Connections in a Stiffness Solution of Plate Bending," AIAA J., Vol. 3, No. 5, 1965.

44. Irons, B. M., and Razzaque, A., "Experience with the Patch Test for Convergence of Finite Elements," O.N.R. Conference on the Mathematical Foundations of the Finite Element Method, University of Maryland, Baltimore, June 1972.

45. Irons, B. M., Zienkiewicz, O. C., and Oliveira, E. R. A., "Comments on the Paper: Theoretical Foundations of the Finite Element Method," IJSS, Vol. 6, pp. 695-697.

46. Jenkins, W. M., Matrix and Digital Computer Methods in Structural Analysis, McGraw-Hill, 1969.

47. Kraus, H., Thin Elastic Shells, John Wiley & Sons, New York, 1967.

48. Langhaar, H. L., Energy Methods in Applied Mechanics, John Wiley & Sons, New York, 1962.

49. Love, A. E. H., A Treatise on the Mathematical Theory of Elasticity, Dover Publications, New York, 4th Edition, 1927.

50. Marguerre, K., "Zur Theorie der Gekrummenten Platte Grosser Formanderung," Proc. 5th International Congress of Applied Mechanics, Cambridge, Mass., 1938.

51. Melosh, R., "A flat Triangular Shell Element Stiffness Matrix," First Conf., pp. 503-514.

52. Melosh, R. J., and Bamford, R. M., "Efficient Solution of Load-Deflection Equations," Journal of the Structural Division - Proceedings of the ASCE, New York, Vol. 95, ST4, 1969.

53. Millward, D. R., "A Finite Element Analysis of Thin Shell Structures," M. Sc. Thesis, VPI & SU, Blacksburg, Va., March 1976.

54. Moore, J. B., WATFIV: FORTRAN Programming with the WATFIV Compiler, Reston Publishing Co., Reston, Va., 1975.

55. Morley, L. S. D., and Merrifield, B. C., "On the Conforming Cubic Triangular Element for Plate Bending," IJCS, Vol. 2, No. 51[illegible], 1972, pp. 875-892.

56. Novozhilov, V. V., Thin Shell Theory, P. Noordhoff, Groningen, Netherlands, 2nd Edition, 1964.

57. Oden, J. T., Finite Elements of Nonlinear Continua, McGraw-Hill, New York, 1972.

58. Oliveira, E. R. A., "Completeness and Convergence in the Finite Element Method," Second Conf., pp. 1061-1090.

59. Oliveira, E. R. A., "Theoretical Foundations of the Finite Element Method," IJSS, Vol. 4, No. 10, 1968, pp. 929-952.

60. Pian, T. H. H., and Tong, P., "Basis of Finite Element Methods for Solid Continua," IJNME, Vol. 1, No. 1, 1969, pp. 3-28.

61. Poceski, A., "A Mixed Finite Element Method for Bending of Plates," IJNME, Vol. 9, No. 1, 1975, pp. 3-15.

62. Prizemieniecki, J. S., Theory of Matrix Structural Analysis, McGraw-Hill, New York, 1968.

63. Razzaque, A., "Finite Element Analysis of Plates and Shells," Ph.D. Thesis, University of Wales, Swansea, Dec. 1972.

64. Reissner, E., "On Some Problems in Shell Theory," Proc. First Symposium on Naval Structural Mechanics, Pergamon Press, New York, 1960.

65. Robinson, J., "A Single Element Test," CMAME, Vol. 7, No. 2, 1976, pp. 191-200.

66. Seide, P., Small Elastic Deformations of Thin Shells, Noordhoff, Leyden, Netherlands, 1975.

67. Shieh, W. Y. J., Lee, S. L., and Parmelee, R. A., "Analysis of Plate Bending by Triangular Elements," ASCE-EM, Vol. 94, No. 5, 1968, pp. 1089-1107.

68. Slyper, H. A., "Development of Explicit Stiffness and Mass Matrices for a Triangular Plate Element," IJSS, Vol. 5, No. 3, 1969, pp. 241-249.

69. Stoner, R. E., "Ring Finite Elements Used in the Analysis of Shells of Revolution," M. Sc. Thesis, VPI & SU, Blacksburg, Va., Nov. 1975.

70. Strang, G., "Variational Crimes in the Finite Element Method," O.N.R. Conference on the Mathematical Foundations of the Finite Element Method, University of Maryland, Baltimore, June 1972.

71. Strickland, G. E. Jr., and Loden, W. A., "A Doubly Curved Triangular Shell Element," Second Conf., pp. 641-666.

72. Tong, P., and Pian, T. H. H., "The Convergence of Finite Element Method in Solving Linear Elastic Problems," IJSS, Vol. 3, No. 5, 1967, pp. 865-879.

73. Utku, S., "Stiffness Matrices for Thin Triangular Elements of Nonzero Gaussian Curvature," AAIA J., Vol. 5, No. 9, 1967, pp. 1659-1667.

74. de Veubeke, B. F., "Variational Principles and the Patch Test," IJNME, Vol. 8, No. 4, 1974, pp. 783-801.

75. Vos, R., "Generalization of Plate Finite Elements to Shells," ASCE-EM, Vol. 98, No. 2, 1972, pp. 385-400.

76. Wang, Chu-kia, Matrix Methods of Structural Analysis, International Textbook Co., Scranton, Pa., 1966.

77. Wempler, G. A., Oden, J. T., and Kross, D. A., "Finite-Element Analysis of Thin Shells," ASCE-EM, Vol. 94, 1968, pp. 1273-1294.

78. Wempner, G., "Finite Elements, Finite Rotations and Small Strains of Flexible Shells," IJSS, Vol. 5, No. 2, 1969, pp. 117-153.

79. Zienkiewicz, O. C., The Finite Element Method in Engineering Science, McGraw-Hill, London, (2nd Edition), 1971.

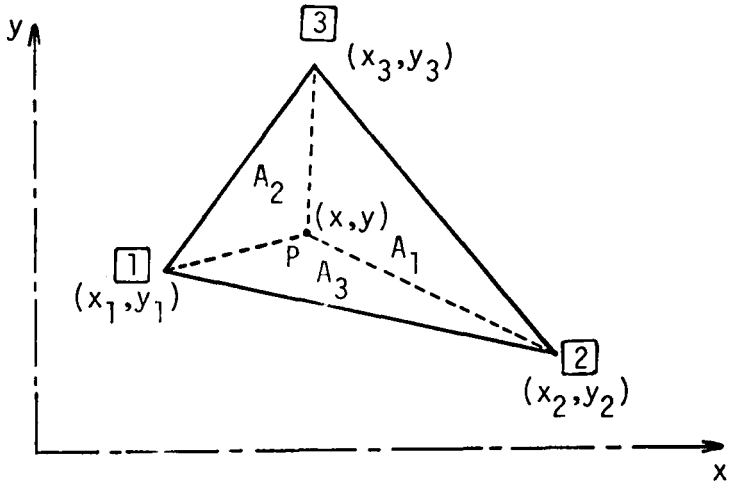
APPENDIX A

TRIANGULAR (AREA) (NATURAL) COORDINATES

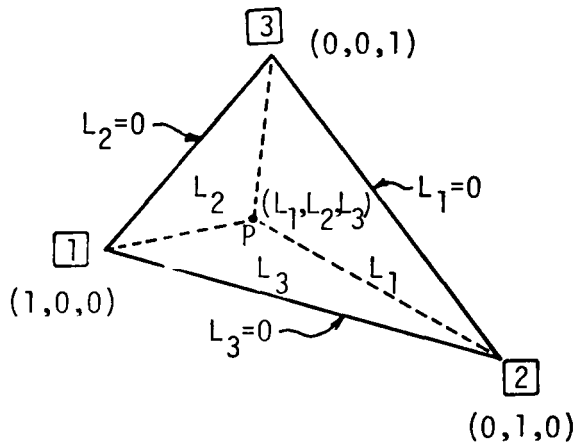
Triangular coordinates are intrinsic local normalized coordinates of triangular domains [38]. The inherent symmetry of the triangular shape is taken advantage of in deriving these dimensionless normalized area coordinates. Each of the three triangular coordinates assumes a unit value at one particular corner (primary) node and zero values at the other two corner (primary) nodes, while exhibiting linear variations between these nodes [40 (p. 139)]. Consequently, the use of area coordinates simplifies triangular element formulations and the construction of isotropic shape functions. Additionally, explicit integrations and numerical integrations (quadratures) are simplified when triangular coordinates are used in triangular element formulations [27 (p. 88), 38, 40 (p. 139), 79 (p. 149)].

Some of the useful relations, properties and transformations involving triangular coordinates and local cartesian coordinates are presented below. For more information on these particular natural coordinates, the reader is referred to other sources [7, 27 (p. 88), 32 (p. 229), 38, 40 (p. 139), 71, 79]. Note that triangular coordinates may only be used for points that lie within or on the boundary of the element triangular domain. The following information pertains to Figure A.1.

$$L_i = A_i/A, \quad \text{where } i = 1, 2, 3 \tag{A.1}$$



LOCAL CARTESIAN SPACE - DOMAIN



NATURAL COORDINATE SPACE - MAPPED IMAGE

Figure A.1 Physical relationship between local cartesian coordinates and natural (triangular) (area) coordinates [38].

and

$$A = A_1 + A_2 + A_3 \tag{A.2}$$

Therefore

$$1 = L_1 + L_2 + L_3$$

also

$$x = x_1L_1 + x_2L_2 + x_3L_3$$

$$y = y_1L_1 + y_2L_2 + y_3L_3$$

or, in matrix form

$$\begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \end{bmatrix} \begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} \tag{A.3}$$

conversely

$$\begin{Bmatrix} L_1 \\ L_2 \\ L_3 \end{Bmatrix} = \frac{1}{2A} \begin{bmatrix} a_1 & b_1 & c_1 \\ a_2 & b_2 & c_2 \\ a_3 & b_3 & c_3 \end{bmatrix} \begin{Bmatrix} 1 \\ x \\ y \end{Bmatrix} \tag{A.4}$$

or

$$L_i = \frac{1}{2A} (a_i + b_ix + c_iy), \qquad i = 1, 2, 3$$

where

$$a_1 = x_2y_3 - x_3y_2$$

$$a_2 = x_3y_1 - x_1y_3$$

$$a_3 = x_1y_2 - x_2y_1$$

$$b_1 = y_2 - y_3$$

$$b_2 = y_3 - y_1 \qquad \text{(A.5)}$$

$$b_3 = y_1 - y_2$$

$$c_1 = x_3 - x_2$$

$$c_2 = x_1 - x_3$$

$$c_3 = x_2 - x_1$$

and

$$2A = b_1c_2 - b_2c_1$$

$$2A = b_2c_3 - b_3c_2 \qquad \text{(A.6)}$$

$$2A = b_3c_1 - b_1c_3$$

also

$$b_1 + b_2 = -b_3$$

$$b_2 + b_3 = -b_1 \qquad \text{(A.7)}$$

$$b_3 + b_1 = -b_2$$

$$
\begin{aligned}
c_1 + c_2 &= -c_3 \\
c_2 + c_3 &= -c_1 \\
c_3 + c_1 &= -c_2
\end{aligned}
\tag{A.7}
$$

and

$$
\begin{aligned}
\frac{\partial L_i}{\partial x} &= \frac{b_i}{2A} \quad , \quad i = 1, 2, 3 \\
\frac{\partial L_i}{\partial y} &= \frac{c_i}{2A} \quad , \quad i = 1, 2, 3
\end{aligned}
\tag{A.8}
$$

Thus, transformation of differentials results in

$$
\begin{aligned}
\frac{\partial(\ )}{\partial x} &= \frac{1}{2A}\left[b_1\frac{\partial(\ )}{\partial L_1} + b_2\frac{\partial(\ )}{\partial L_2} + b_3\frac{\partial(\ )}{\partial L_3}\right] \\
\frac{\partial(\ )}{\partial y} &= \frac{1}{2A}\left[c_1\frac{\partial(\ )}{\partial L_1} + b_2\frac{\partial(\ )}{\partial L_2} + b_3\frac{\partial(\ )}{\partial L_3}\right]
\end{aligned}
\tag{A.9}
$$

and, transformation of integrals yields

$$
\iint_{A_{x,y}} f(x,y)\, dxdy = 2A \iiint_{A_{L_1,L_2,L_3}} f(L_1,L_2,L_3)\, dL_1 dL_2 dL_3
$$

and, conveniently

$$
\iiint_{A_{L_1,L_2,L_3}} L_1^{\alpha} L_2^{\beta} L_3^{\gamma} dL_1 dL_2 dL_3 = \frac{\alpha!\ \beta!\ \gamma!}{(\alpha + \beta + \gamma + 2)!}
$$

consequently

$$
\iint_{A_{x,y}} L_1^{\alpha} L_2^{\beta} L_3^{\gamma}\, dxdy = \frac{(2A)\ \alpha!\ \beta!\ \gamma!}{(\alpha + \beta + \gamma + 2)!} \tag{A.10}
$$

Huebner gives numerical values of these integrals for various values of α, β, and γ [40 (p. 144)].

APPENDIX B

INPUT DATA

A summation and clarification of the various program input quantities is presented in this appendix. Before discussing these input quantities, a few peculiarities of the present computer program are pointed out.

First of all, it should be noted that variable names that begin with the letter I, J, K, M, or N are implicitly defined as INTEGER quantities. Similarly, variable names that begin with any other alphabetical letter are implicitly defined as REAL*8 (DOUBLE PRECISION) quantities. Note that variable names that begin with the letter L are REAL*8, and not integer, quantities.

Secondly, in order to eliminate excessive computer storage costs, the dimensions of various variable arrays (i.e. vectors or matrices) should be adjusted to match those actually required for the particular shell model being analyzed. This annoying complication may be avoided by simply dimensioning the arrays for the largest problem to be analyzed. However, by using this crude approach, savings in computer storage costs for smaller problems are not possible.

The following four labeled COMMON cards are the only cards affected by this dimensioning change. The dimensions that should be adjusted are given in terms of variable names. The values of these variables are also given as follows:

```
      COMMON /C2/X(NN,3),XZ(NZ,3),INZ(NE,3)/
      COMMON /C3/INC(NE,5),ICON(ND)/
      COMMON /C4/B(ID,3),C(ID,3),A(ID),ZX(ID),ZY(ID),ZXY(ID)/
      COMMON /C5/P(ND),GK(ND,IBW)/
```

where

$$ND = 9(MX+1)(MY+1)$$

$$IBW = 9(MX+2)$$

and

$$NE = 2(MX)(MY)$$

$$NN = (MX+1)(MY+1)$$

$$NZ = MX + MY + 3(MX)(MY)$$

However, if ID = 2, then:

$$NE = 2(MX)$$

$$NN = 2(MX+1)$$

$$NZ = 1 + 4(MX)$$

A combination of two different methods are used to input the problem parameters in the present program. Part of the input data is read into the computer by using READ statements that input the quantities contained in the data cards placed at the end of the program. One annoyance of this method is that the data cards are not automatically printed out along with the rest of the program listing and output. Additionally, this method is less

efficient in initializing the values of arrays that are contained in COMMON block statements. Consequently, a BLOCK DATA subprogram [54 (p. 249)] is used in the present program to efficiently input some of the quantities contained in COMMON block statements. Values that are inputed in this manner are printed out along with the program listing. The statements that contain the input data quantities in the BLOCK DATA subroutine are called DATA statements. These DATA statements will be described first.

Only three DATA statement cards are used in the present program. These are as follows:

```
DATA IDS,ID,IB1,IB2,IB3,IB4/xx,xxx,x,x,x,x/
DATA MX,MY/xx,xx/
DATA WX,WY/xx...xx,xx...xx/
```

where the "x"s refer to the inputed values of the previously listed variable quantities on that card, respectively. (Note that WX for cylindrical shells is imputed in degrees and not radians!).

Along with the above three DATA statement cards, two data cards are additionally required. These two data cards, which contain material property and applied load state information, are described as follows:

data card #1 (TH,PR,EL values):

```
1  4          15         26  ← column #
|  |          |          |
°°°xx.xxxx°°°°x.xxxx°°°°°xxxxxxxx.x
```

data card #2 (PX,PY,PZ,W values):

```
 2          11         21         31  ← column #
 |          |          |          |
°xxx.xxxxx°xxx.xxxxx°xxx.xxxxx°xxx.xxxxx
```

where the "°"s correspond to blank spaces, and the "x"s correspond to the inputed quantities.

One additional data card is required in analyses of right circular cylindrical shell, hyperbolic paraboloid shell, and elliptic paraboloid shell, structures. This one card, which is positioned before the other two data cards, contains some additional dimensional data necessary to completely define the geometry of the model. Since the format of this card varies for each of the three structure types, the form of this card is not described here. Rather, the reader is referred to subroutine GEOM for the details of this card.

If a general thin shell structure is to be analyzed, then the INC(NE,5),X(NN,3), and XZ(NZ,3) quantities must also be inputed with the use of data cards. Since the number of these additional required data cards varies, they also are not described here. Refer to subroutine GEOM for the format of these data cards. Note that these cards are also placed before the other two previously discussed data cards.

APPENDIX C

PROGRAM OUTPUT

Besides printing out the structure type, and any dimensional data peculiar to the particular structure type, the following quantities are outputed:

the mesh size (MX x MY)

the total number of elements (NEM)

the total number of primary nodes (NNM)

the total number of secondary nodes (NZM)

the total number of system DOF (ND)

the system stiffness matrix half-bandwidth (IBW)

the analysis domain dimensions (WX x WY)

the analysis domain edge boundary conditions

the values of: X(NN,3),XZ(NZ,3),INC(NE,5),INZ(NZ,3),ICON(ND), TH,PR,EL,B(ID,3),C(ID,3),A(ID),ZX(ID),ZY(ID),ZX(ID),PX,PY,PZ, W,P(ND)

and

the calculated values of the global surface coordinate displacements corresponding to the applied load state.

In addition to these output quantities listed above, many intermediate quantities are presently printed out in order to aid the de-bugging process.

APPENDIX D

THE COMPUTER PROGRAM

```
C*************************************************************************
C  THIS PROGRAM ANALYZES LINEAR ELASTIC THIN SHELL STRUCTURES BY THE
C  DISPLACEMENT TYPE FINITE ELEMENT METHOD UTILIZING NON-CONFORMING
C  DOUBLY CURVED TRIANGULAR SHALLOW SHELL ELEMENTS.  DERIVATION OF THE
C  INTERPOLATION FUNCTIONS IS BASED ON A SINGLE FIELD FORMULATION WITH
C  CUBIC REPRESENTATION OF BOTH THE TANGENTIAL AND TRANSVERSE (NORMAL)
C  DISPLACEMENTS.  EACH ELEMENT HAS 27 DEGREES OF FREEDOM (9 AT EACH VERTEX).
C*************************************************************************
C  GLOSSARY OF MAJOR ARRAYS AND VARIABLES:
C
C  QUANTITIES IN COMMON STATEMENTS:
C    ND = 9*NNM = MAXIMUM NO. OF DEGREES OF FREEDOM IN THE SYSTEM.
C    NEM = 2*MX*MY = MAXIMUM NO. OF TRIANGULAR ELEMENTS (27 D.O.F.).
C    NE = THE NO. OF TRIANGULAR ELEMENTS NEEDED FOR VARIOUS CALCULATIONS.
C    NNM = NX*NY = MAXIMUM NO. OF PRIMARY NODES (9 D.O.F. PER NODE).
C    NN = THE NO. OF PRIMARY NODES NEEDED FOR VARIOUS CALCULATIONS.
C    IBW = 9*(MX+2) = THE HALF-BANDWIDTH OF THE SYSTEM STIFFNESS MATRIX.
C    NZ = THE NO. OF SECONDARY NODES NEEDED TO DEFINE THE ELEMENT CURVATURES.
C    ID = TOTAL NO, OF DISSIMILAR ELEMENTS9
C    WX,WY = RESPECTIVE "WIDTH" DIMENSIONS OF THE RECTANGULAR ANALYSIS
C            DOMAIN IN THE X AND Y DIRECTIONS.
C    MX,MY = THE NO. OF MESH DIVISIONS IN GLOBAL X AND Y DIRECTIONS,
C            RESPECTIVELY.
C    XS,YS = WIDTH OF THE MESH DIVISIONS IN GLOBAL X AND Y DIRECTIONS,
C            RESPECTIVELY.
C    IDS = STRUCTURE TYPE INDICATOR (1 = GENERAL SHELL, 2 = FLAT PLATE,
C          3 = RIGHT CIRCULAR CYLINDRICAL SHELL, 4 = SHALLOW HYPERBOLIC
C          PARABOLOID SHELL, 5 = ELLIPTIC PARABOLOID SHELL).
C    IB1,IB2,IB3,IB4 = BOUNDARY CONDITION INDICATORS OF THE X=0,X=WX,Y=0,Y=WY
C                      SIDES OF THE RECT. ANALYSIS DOMAIN, RESPECTIVELY
C                      (1 = FREE EDGE, 2 = KNIVE(DIAPHRAM) EDGE, 3 = HINGED
C                      (SIMPLY SUPPORTED), 4 = SYMMETRY, 5 = CLAMPED(FIXED)).
```

```
C    TH = UNIFORM SHELL SURFACE THICKNESS.
C    PR = UNIFORM SHELL MATERIAL POISSON'S RATIO.
C    EL = UNIFORM SHELL MATERIAL YOUNG'S MODULUS OF ELASTICITY.
C    X(NN,3) = PRIMARY NODAL COORDINATE MATRIX.
C    XZ(NZ,3) = SECONDARY NODAL COORDINATE MATRIX.
C    INC(NE,5) = ELEMENT PRIMARY NODAL INCIDENCE, ORIENTATION, AND SIMILARITY
C                MATRIX (FIRST THREE TERMS CORRESPOND TO VERTICES 1,2,3,RESP.;
C                FOURTH TERM = ORIENTATION TYPE(0=A,1=B);FIFTH TERM = ELEMENT
C                SIMILARITY INDICATOR(0=NONE)).
C    INZ(NE,3) = ELEMENT SECONDARY NODAL INCIDENCE MATRIX (TERMS CORRESPOND TO
C                SIDES OPPOSITE VERTICES 1,2,3, RESPECTIVELY).
C    ICON(ND) = PRIMARY NODAL DISPLACEMENT STATUS MATRIX (0=FREE,1=CONSTRAINED).
C    B(ID,3),C(ID,3) = TERMS IN THE TRIANGULAR COORDINATE TRANSFORMATION MATRIX.
C    A(ID) = ELEMENT BASE TRIANGLE AREA MATRIX.
C    ZX(ID),ZY(ID),ZXY(ID) = ELEMENT VALUES OF D2Z/(DX)2,D2Z/(DY)2,D2Z/(DX)(DY),
C                            RESPECTIVELY (NEGATIVE INVERSES OF THE RADII
C                            OF CURVATURES).
C    P(ND) = GLOBAL (SURFACE) APPLIED NODAL FORCE VECTOR, OR GLOBAL NODAL
C            DISPLACEMENT SOLUTION VECTOR.
C    GK(ND,IBW) = GLOBAL (SURFACE) (SYSTEM) STIFFNESS MATRIX IN HALF-BANDWIDTH
C                 RECTANGULAR FORM.
C    GL(27,27) = LOCAL ELEMENT STIFFNESS MATRIX.
C    THE REST OF THE COMMON STATEMENT VARIABLES AND ARRAYS ARE USED AS WORKING
C    REGISTERS IN SUBROUTINES GKK, ASMGL, GKF, GKM, AND GKKM.
C
C  QUANTITIES IN SUBROUTINES GEOM AND CALC:
C    NX,NY = (MX+1),(MY+1) = THE NUMBER OF MESH LINES IN THE GLOBAL X AND Y
C                            DIRECTIONS, RESPECTIVELY.
C
C  QUANTITIES IN SUBROUNTINE GEOM:
C    NZM = 3*MX*MY+MX+MY = MAXIMUM NO. OF SECONDARY NODES.
C    NEX = 2*MX = THE NUMBER OF ELS. PER ROW IN THE X DIRECTION.
```

```
C
C QUANTITIES IN SUBROUTINE CALC:
C   L12,L13,L23 = LENGTHS OF ELEMENT SIDES 1-2,1-3,2-3, RESPECTIVELY.
C   ALPHA = ANGLE (IN RADIANS) BETWEEN SIDE 1-3 AND SIDE 1-2.
C   ZAA,ZBB,ZCC = ELEVATIONS OF MIDSIDE SECONDARY NODES 4,5,6, RESPECTIVELY
C                 (CORRESPOND TO Z4,Z5,Z6 OF CHAPTER 3).
C
C QUANTITIES IN SUBROUTINE LOAD:
C   PX,PY,PZ = VALUES OF DISTRIBUTED LOADS IN SURFACE S1,S2,S3 DIRECTIONS,
C                RESPECTIVELY.
C   W = VALUE OF SELF-WEIGHT LOAD (USUALLY POSITIVE).
C   Q(6) = WORKING REGISTER FOR CALCULATING EFFECT OF W ON P(ND).
C
C QUANTITIES IN SUBROUTINE GKF:
C   BFFF(9,9),RCBF(9,9) = WORKING REGISTERS.
C
C QUANTITIES IN SUBROUTINE GKKM:
C   RBMB(18,6),RBMC(18,6),BK(18,3) = WORKING REGISTERS.
C
C QUANTITIES IN SUBROUTINE TRANSF:
C   GT(27,27) = WORKING REGISTER.
C*****************************************************************************
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C2/X(6,3),XZ(9,3),INZ(8,3)
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C5/P(81),GK(81,36)
      COMMON /C6/WX,WY,XS,YS,MX,MY,IDS,IB1,IB2,IB3,IB4
      COMMON /C7/TH,PR,EL
      COMMON /C8/GL(27,27),GW(9,9),II,KRC,KCO,ILO,IUP,JLO,JUP,KIND
      COMMON /C9/BMA(21,6),BMB(21,6),BMC(21,6)
```

```
      COMMON /C10/RB(21,21),RBBM(21,6),QB(7,7)
      CALL GEOM
      CALL CALC
      CALL LOAD
C GK(ND,IBW) PRE-ZEROING:
      DO 5 I=1,ND
      DO 5 J=1,IBW
    5 GK(I,J)=0.0D00
      DO 10 II=1,ID
C GL(27,27) PRE-ZEROING:
      DO 6 I=1,27
      DO 6 J=1,27
    6 GL(I,J)=0.0D00
C SKIP GKK CALCULATION WHEN ANALIZING A PLATE STRUCTURE:
      IF(IDS.EQ.2) GO TO 8
      CALL GKK
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,100) ((GL(I,J),J=1,9),I=1,27)
      WRITE(6,100) ((GL(I,J),J=10,18),I=1,27)
      WRITE(6,100) ((GL(I,J),J=19,27),I=1,27)
    8 CONTINUE
      CALL GKF
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,100) ((GL(I,J),J=1,9),I=1,27)
      WRITE(6,100) ((GL(I,J),J=10,18),I=1,27)
      WRITE(6,100) ((GL(I,J),J=19,27),I=1,27)
      CALL GKM
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,100) ((GL(I,J),J=1,9),I=1,27)
      WRITE(6,100) ((GL(I,J),J=10,18),I=1,27)
      WRITE(6,100) ((GL(I,J),J=19,27),I=1,27)
      CALL GKKM
```

```
C  TEMPORARY DE-BUGGING PRINTOUTS:
       WRITE(6,100) ((GL(I,J),J=1,9),I=1,27)
       WRITE(6,100) ((GL(I,J),J=10,18),I=1,27)
       WRITE(6,100) ((GL(I,J),J=19,27),I=1,27)
       CALL TRANSF
C  TEMPORARY DE-BUGGING PRINTOUTS:
       WRITE(6,100) ((GL(I,J),J=1,9),I=1,27)
       WRITE(6,100) ((GL(I,J),J=10,18),I=1,27)
       WRITE(6,100) ((GL(I,J),J=19,27),I=1,27)
       CALL ASEMBL
    10 CONTINUE
C  TEMPORARY DE-BUGGING PRINTOUTS:
       WRITE(6,100)((GK(I,J),J=1,9),I=1,ND)
       WRITE(6,100)((GK(I,J),J=10,18),I=1,ND)
       WRITE(6,100)((GK(I,J),J=19,27),I=1,ND)
       WRITE(6,100)((GK(I,J),J=28,36),I=1,ND)
       CALL CONSTR
C  TEMPORARY DE-BUGGING PRINTOUTS:
       WRITE(6,100)((GK(I,J),J=1,9),I=1,ND)
       WRITE(6,100)((GK(I,J),J=10,18),I=1,ND)
       WRITE(6,100)((GK(I,J),J=19,27),I=1,ND)
       WRITE(6,100)((GK(I,J),J=28,36),I=1,ND)
       WRITE(6,100)(P(I),I=1,ND)
       CALL BNDSLV
C  TEMPORARY DE-BUGGING PRINTOUTS:
       WRITE(6,100)((GK(I,J),J=1,9),I=1,ND)
       WRITE(6,100)((GK(I,J),J=10,18),I=1,ND)
       WRITE(6,100)((GK(I,J),J=19,27),I=1,ND)
       WRITE(6,100)((GK(I,J),J=28,36),I=1,ND)
       STOP
       END
```

```
      BLOCK DATA
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C6/WX,WY,XS,YS,MX,MY,IDS,IB1,IB2,IB3,IB4
      DATA IDS,ID,IB1,IB2,IB3,IB4/2,2,4,3,3,4/
      DATA MX,MY/ 2, 2/
      DATA WX,WY/10.0D00, 10.0D00/
      END
```

```
      SUBROUTINE GEOM
C=======================================
C SUBROUTINE GEOM READS IN THE GEOMETRICAL DATA AND ESTABLISHES THE
C SHELL STRUCTURE BY DEFINING THE TYPE OF SHELL STRUCTURE, THE GLOBAL
C COORDINATES OF THE NODES, THE ORIENTATION OF THE ELEMENTS, AND THE
C GEOMETRICAL CONSTRAINTS (BOUNDARY CONDITIONS).
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C2/X(6,3),XZ(9,3),INZ(8,3)
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C6/WX,WY,XS,YS,MX,MY,IDS,IB1,IB2,IB3,IB4
      WRITE(6,1)
    1 FORMAT(/1X,'**********'/1X,'UNITS ARE: INCHES, POUNDS, RADIANS (UN
     +LESS OTHERWISE NOTED)'/1X,'**********'//)
      NEX=2*MX
      NX=MX+1
      NY=MY+1
      XS=WX/MX
      YS=WY/MY
      IF(IDS.NE.4) GO TO 2
      XSS=WX/(-2.0D00)
      YSS=WY/(-2.0D00)
      GO TO 4
    2 XSS=0.0D00
      YSS=0.0D00
    4 NEM=2*MX*MY
      NNM=NX*NY
      ND=9*NNM
      NZM=(2*MX+1)*MY+MX*NY
      IBW=9*(MX+2)
```

```
      IF(ID.NE.2) GO TO 6
      NE=2*MX
      NN=2*NX
      NZ=4*MX+1
      GO TO 8
    6 NE=NEM
      NN=NNM
      NZ=NZM
C
C THE FOLLOWING SECTION GENERATES MATRICES INC(NE,4) AND INZ(NE,3):
    8 DO 10 I=1,MY
      K1=(I-1)*NX
      K2=I*NX
      KZ1=(I-1)*(3*MX+1)
      KZ2=KZ1+MX
      KZ3=I*(3*MX+1)
      DO 10 J=2,NEX,2
      K=(I-1)*2*MX+J-1
      M=(I-1)*2*MX+J
      KK=J/2
      M1=K1+KK
      M2=M1+1
      N1=K2+KK
      N2=N1+1
      INC(K,1)=M1
      INC(K,2)=M2
      INC(K,3)=N1
      INC(K,4)=0
      INC(M,1)=N1
      INC(M,2)=N2
      INC(M,3)=M2
      INC(M,4)=1
```

```
      J1=KZ1+KK
      J2=KZ2+J-1
      J3=J2+1
      J4=J3+1
      J5=KZ3+KK
      INZ(K,1)=J3
      INZ(K,2)=J2
      INZ(K,3)=J1
      INZ(M,1)=J4
      INZ(M,2)=J3
      INZ(M,3)=J5
   10 CONTINUE
C THE FOLLOWING SECTION COMPLETES MATRIX INC(NE,5):
      GO TO(400,380,380,390,390),IDS
  380 DO 385 I=2,NEM,2
      K=I-1
      INC(K,5)=1
      INC(I,5)=2
  385 CONTINUE
      GO TO 410
  390 DO 395 I=1,NEM
      INC(I,5)=I
  395 CONTINUE
      GO TO 410
  400 READ(5,405) (INC(I,5),I=1,NEM)
  405 FORMAT((14(2X,I3)))
C
C THE FOLLOWING SECTION GENERATES MATRICES X(NN,2) AND XZ(NZ,2):
  410 IF(ID.NE.2) GO TO 222
      MYA=1
      NYA=2
      GO TO 224
```

```
222 NYA=NY
    MYA=MY
224 DO 230 I=1,NYA
    IA=NX*(I-1)
    IB=(I-1)*(3*MX+1)
    DO 225 J=1,NX
    JJ=IA+J
    X(JJ,1)=(J-1)*XS+XSS
    X(JJ,2)=(I-1)*YS+YSS
225 CONTINUE
    DO 230 J=1,MX
    JA=IA+J
    JB=JA+1
    JC=IB+J
    XZ(JC,1)=(X(JA,1)+X(JB,1))/2.0D00
    XZ(JC,2)=X(JA,2)
230 CONTINUE
    DO 235 I=1,MYA
    IA=(I-1)*NX+1
    IB=I*NX+1
    IZ=(I-1)*(3*MX+1)+MX+1
    XZ(IZ,1)=0.0D00+XSS
    XZ(IZ,2)=(X(IA,2)+X(IB,2))/2.0D00
    DO 235 J=1,MX
    JA=IA+J
    JB=IB+(J-1)
    JC=JB+1
    JAA=(IZ-1)+J*2
    JBB=JAA+1
    XZ(JAA,2)=(X(JA,2)+X(JB,2))/2.0D00
    XZ(JAA,1)=(X(JA,1)+X(JB,1))/2.0D00
    XZ(JBB,1)=(X(JA,1)+X(JC,1))/2.0D00
```

```
      XZ(JBB,2)=(X(JA,2)+X(JC,2))/2.0D00
  235 CONTINUE
C THE FOLLOWING SECTION COMPLETES MATRICES X(NN,3) AND XZ(NZ,3):
      GO TO(325,240,260,285,305),IDS
  240 DO 245 I=1,NN
      X(I,3)=0.0D00
  245 CONTINUE
      DO 250 I=1,NZ
      XZ(I,3)=0.0D00
  250 CONTINUE
      WRITE(6,255)
  255 FORMAT(1X,'FLAT PLATE STRUCTURE')
      GO TO 340
  260 READ(5,265) R
  265 FORMAT(2X,F8.2)
      DO 270 I=1,NN
C CONVERT "X(I,1)" COORDINATES FROM DEGREES TO RADIANS:
      THETA=3.141592654D00*X(I,1)/180.0D00
      X(I,3)=R*DCOS(THETA)
C CONVERT "X(I,1)" COORDINATES FROM RADIANS TO INCHES:
      X(I,1)=R*DSIN(THETA)
  270 CONTINUE
      DO 275 I=1,NZ
C CONVERT "XZ(I,1)" COORDINATES FROM DEGREES TO RADIANS:
      THETA=3.141592654D00*XZ(I,1)/180.0D00
      XZ(I,3)=R*DCOS(THETA)
C CONVERT "XZ(I,1)" COORDINATES FROM RADIANS TO INCHES:
      XZ(I,1)=R*DSIN(THETA)
  275 CONTINUE
      WRITE(6,280) R
  280 FORMAT(1X,'RIGHT CIRCULAR CYLINDRICAL SHELL STRUCTURE'/1X,'RADIUS=
     + ',F8.2)
```

```
      GO TO 340
  285 READ(5,288) RC
  288 FORMAT(2X,F8.5)
      DO 290 I=1,NN
      X(I,3)=X(I,1)*X(I,2)/RC
  290 CONTINUE
      DO 295 I=1,NZ
      XZ(I,3)=XZ(I,1)*XZ(I,2)/RC
  295 CONTINUE
      WRITE(6,300) RC
  300 FORMAT(1X,'HYPERBOLIC PARABOLOID SHELL STRUCTURE'/1X,'RC CONSTANT=
     + ',F8.5)
      GO TO 340
  305 READ(5,308) RX,RY
  308 FORMAT(2(2X,F8.2))
      RA=2.0D00*RX
      RB=2.0D00*RXY
      DO 310 I=1,NN
      X(I,3)=(X(I,1)**2/RA)+(X(I,2)**2/RB)
  310 CONTINUE
      DO 315 I=1,NZ
      XZ(I,3)=(XZ(I,1)**2/RA)+(XZ(I,2)**2/RB)
  315 CONTINUE
      WRITE(6,320) RA,RB
  320 FORMAT(2X,'ELLIPTIC PARABOLOID SHELL STRUCTURE'/1X,'RA= ',F8.2,4X,
     +'RB= ',F8.2)
      GO TO 340
  325 READ(5,330) ((X(I,J),J=1,3),I=1,NN)
      READ(5,330) ((XZ(I,J),J=1,3),I=1,NZ)
  330 FORMAT((6(1X,F11.4)))
      WRITE(6,335)
  335 FORMAT(1X,'GENERAL SHELL PROBLEM')
```

```
C
C THE FOLLOWING SECTION PRINTS OUT MOST OF THE GEOMETRICAL DATA:
  340 WRITE(6,345) MX,MY,NEM,NNM,ND,NZM,IBW,WX,WY
  345 FORMAT(/1X,'MESH SIZE: ',I2,2X,'X',I2/1X,'NO. ELEMENTS= ',I3/1X,'N
     +O. NODES= ',I3/1X,'NO. DEGREES FREEDOM= ',I3/1X,'NO. Z-NODES= ',I3
     +/1X,'HALF-BANDWIDTH= ',I3//1X,'ANALYSIS AREA DIMENSIONS (DEGREES,
     +INCHES):  A= ',F9.4,2X,'B= ',F9.4)
      WRITE(6,360) (I,(X(I,J),J=1,3),I=1,NN)
  360 FORMAT(/1X,'NODE ',6X,'X-AXIS',7X,'Y-AXIS',7X,'Z-AXIS'/((1X,I3,3X,
     +2X,F11.6,2X,F11.6,2X,F11.6)))
      WRITE(6,365) (I,(XZ(I,J),J=1,3),I=1,NZ)
  365 FORMAT(/1X,'Z-NODE ',8X,'X-AXIS',7X,'Y-AXIS',7X,'Z-AXIS'/((2X,I3,6
     +X,2X,F11.6,2X,F11.6,2X,F11.6)))
      WRITE(6,370) (I,(INC(I,K),K=1,5),(INZ(I,J),J=1,3),I=1,NE)
  370 FORMAT(/1X,'ELEMENT',3X,'1-NODE',3X,'2-NODE',3X,'3-NODE',3X,'ORIEN
     +TATION(0=A,1=B)',3X,'SIMILARITY ID NO.',3X,'Z1-NODE',3X,'Z2-NODE',
     +3X,'Z3-NODE'/((3X,I3,6X,I3,6X,I3,6X,I3,13X,I1,20X,I3,13X,I3,7X,I3,
     +7X,I3)))
C
C THE FOLLOWING SECTION GENERATES MATRIX ICON(ND):
      KC=NNM-MX
      WRITE(6,18)
   18 FORMAT(/1X,'BOUNDARY CONDITIONS:')
C THE FOLLOWING INITIALIZES ICON(ND)= 0 :
      DO 15 I=1,ND
      ICON(I)=0
   15 CONTINUE
      GO TO(60,50,40,30,20),IB1
   20 DO 25 I=1,KC,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
```

```
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+8)=1
      ICON(N+9)=1
   25 CONTINUE
      WRITE(6,28)
   28 FORMAT(3X,'X=0: CLAMPED')
      GO TO 65
   30 DO 35 I=1,KC,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+5)=1
      ICON(N+8)=1
   35 CONTINUE
      WRITE(6,38)
   38 FORMAT(3X,'X=0: SYMMETRY')
      GO TO 65
   40 DO 46 I=1,KC,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+9)=1
   46 CONTINUE
      WRITE(6,48)
   48 FORMAT(3X,'X=0: HINGED')
      GO TO 65
   50 DO 55 I=1,KC,NX
```

```
      N=9*(I-1)
      ICON(N+2)=1
      ICON(N+4)=1
      ICON(N+7)=1
      ICON(N+9)=1
55    CONTINUE
      WRITE(6,58)
58    FORMAT(3X,'X=0: DIAPHRAM')
      GO TO 65
60    WRITE(6,62)
62    FORMAT(3X,'X=0: FREE')
65    GO TO(110,100,90,80,70),IB2
70    DO 75 I=NX,NNM,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+8)=1
      ICON(N+9)=1
75    CONTINUE
      WRITE(6,78)
78    FORMAT(3X,'X=A: CLAMPED')
      GO TO 115
80    DO 85 I=NX,NNM,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+5)=1
      ICON(N+8)=1
85    CONTINUE
```

```
      WRITE(6,88)
   88 FORMAT(3X,'X=A: SYMMETRY')
      GO TO 115
   90 DO 96 I=NX,NNM,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+9)=1
   96 CONTINUE
      WRITE(6,98)
   98 FORMAT(3X,'X=A: HINGED')
      GO TO 115
  100 DO 105 I=NX,NNM,NX
      N=9*(I-1)
      ICON(N+2)=1
      ICON(N+4)=1
      ICON(N+7)=1
      ICON(N+9)=1
  105 CONTINUE
      WRITE(6,108)
  108 FORMAT(3X,'X=A: DIAPHRAM')
      GO TO 115
  110 WRITE(6,112)
  112 FORMAT(3X,'X=A: FREE')
  115 GO TO(160,150,140,130,120),IB3
  120 DO 125 I=1,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
```

```
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+8)=1
      ICON(N+9)=1
  125 CONTINUE
      WRITE(6,128)
  128 FORMAT(3X,'Y=0: CLAMPED')
      GO TO 165
  130 DO 135 I=1,NX
      N=9*(I-1)
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+9)=1
  135 CONTINUE
      WRITE(6,138)
  138 FORMAT(3X,'Y=0: SYMMETRY')
      GO TO 165
  140 DO 145 I=1,NX
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+8)=1
  145 CONTINUE
      WRITE(6,148)
  148 FORMAT(3X,'Y=0: HINGED')
      GO TO 165
  150 DO 155 I=1,NX
```

```
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+6)=1
      ICON(N+7)=1
      ICON(N+8)=1
  155 CONTINUE
      WRITE(6,158)
  158 FORMAT(3X,'Y=0: DIAPHRAM')
      GO TO 165
  160 WRITE(6,162)
  162 FORMAT(3X,'Y=0: FREE')
  165 GO TO(210,200,190,180,170),I84
  170 DO 175 I=KC,NNM
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+8)=1
      ICON(N+9)=1
  175 CONTINUE
      WRITE(6,178)
  178 FORMAT(3X,'Y=B: CLAMPED')
      GO TO 214
  180 DO 185 I=KC,NNM
      N=9*(I-1)
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+9)=1
  185 CONTINUE
```

```
      WRITE(6,188)
  188 FORMAT(3X,'Y=B: SYMMETRY')
      GO TO 214
  190 DO 195 I=KC,NNM
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+3)=1
      ICON(N+4)=1
      ICON(N+5)=1
      ICON(N+7)=1
      ICON(N+8)=1
  195 CONTINUE
      WRITE(6,198)
  198 FORMAT(3X,'Y=B: HINGED')
      GO TO 214
  200 DO 205 I=KC,NNM
      N=9*(I-1)
      ICON(N+1)=1
      ICON(N+6)=1
      ICON(N+7)=1
      ICON(N+8)=1
  205 CONTINUE
      WRITE(6,208)
  208 FORMAT(3X,'Y=B: DIAPHRAM')
      GO TO 214
  210 WRITE(6,212)
  212 FORMAT(3X,'Y=B: FREE')
  214 CONTINUE
      WRITE(6,375) (ICON(I),I=1,ND)
  375 FORMAT(/,3X,'W1',4X,'WX1  -WY1   W2    WX2  -WY2   W3    WX3  -WY3
     +'/((4X,I1,5X,I1,5X,I1,5X,I1,5X,I1,5X,I1,5X,I1,5X,I1,5X,I1)))
       RETURN
```

END

```
      SUBROUTINE CALC
C=====================================
C SUBROUTINE CALC READS IN SOME, AND CALCULATES OTHER, STRUCTURAL
C CONSTANTS TO BE USED IN SUBSEQUENT SUBROUTINES.
C=====================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C2/X(6,3),XZ(9,3),INZ(8,3)
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C6/WX,WY,XS,YS,MX,MY,IOS,IB1,IB2,IB3,IB4
      COMMON /C7/TH,PR,EL
      READ(5,4) TH,PR,EL
    4 FORMAT(3X,F7.4,4X,F6.4,5X,F10.1)
      WRITE(6,6) TH,PR,EL
    6 FORMAT(/,1X,'THICKNESS',2X,'POISSONS RATIO',2X,'ELASTICITY MODULUS
     +'/2X,F7.4,6X,F6.4,11X,F10.1)
      WRITE(6,8)
    8 FORMAT(/,1X,'ELEMENT ID',3X,'TYPICAL EL.',4X,'B1',7X,'B2',7X,'B3',
     +7X,'C1',7X,'C2',7X,'C3',7X,'AREA',7X,'D2ZXX',7X,'D2ZYY',7X,'D2ZXY'
     +)
      DO 40 K=1,ID
C CHECK TO SEE IF ALL ELEMENTS ARE DIFFERENT:
      IF(INC(1,5).EQ.0) GO TO 15
      DO 10 I=1,NE
C SEARCH FOR ELEMENT WITH IDENTITY NUMBER EQUAL TO K:
      IF(INC(I,5).EQ.K) GO TO 20
   10 CONTINUE
   15 I=K
   20 XA=X(INC(I,1),1)
      YA=X(INC(I,1),2)
      ZA=X(INC(I,1),3)
```

```
      XB=X(INC(I,2),1)
      YB=X(INC(I,2),2)
      ZB=X(INC(I,2),3)
      XC=X(INC(I,3),1)
      YC=X(INC(I,3),2)
      ZC=X(INC(I,3),3)
      XZA=XZ(INZ(I,1),1)
      YZA=XZ(INZ(I,1),2)
      ZZA=XZ(INZ(I,1),3)
      XZB=XZ(INZ(I,2),1)
      YZB=XZ(INZ(I,2),2)
      ZZB=XZ(INZ(I,2),3)
      XZC=XZ(INZ(I,3),1)
      YZC=XZ(INZ(I,3),2)
      ZZC=XZ(INZ(I,3),3)
      L12=DSQRT((XB-XA)**2+(YB-YA)**2+(ZB-ZA)**2)
      L13=DSQRT((XC-XA)**2+(YC-YA)**2+(ZC-ZA)**2)
      L23=DSQRT((XC-XB)**2+(YC-YB)**2+(ZC-ZB)**2)
      ZAA=DSQRT(((XB+XC)/2.0D00-XZA)**2+((YB+YC)/2.0D00-YZA)**2+((ZB+ZC)
     +/2.0D00-ZZA)**2)
      ZBB=DSQRT(((XA+XC)/2.0D00-XZB)**2+((YA+YC)/2.0D00-YZB)**2+((ZA+ZC)
     +/2.0D00-ZZB)**2)
      ZCC=DSQRT(((XA+XB)/2.0D00-XZC)**2+((YA+YB)/2.0D00-YZC)**2+((ZA+ZB)
     +/2.0D00-ZZC)**2)
C CHECK FOR POSITIVE OR NEGATIVE CURVATURE AND CORRECT:
      IF(ZZA.LT.((ZB+ZC)/2.0D00)) ZAA=-1.0D00*ZAA
      IF(ZZB.LT.((ZA+ZC)/2.0D00)) ZBB=-1.0D00*ZBB
      IF(ZZC.LT.((ZA+ZB)/2.0D00)) ZCC=-1.0D00*ZCC
      ALPHA=DARCOS((L12**2+L13**2-L23**2)/(2.0D00*L12*L13))
      A(K)=L12*L13*DSIN(ALPHA)/2.0D00
C CHECK FOR "B" TYPE ELEMENT:
      IF(INC(I,4).EQ.1) ALPHA=-1.0D00*ALPHA
```

```
      XX1=-1.0D00*L13*DCOS(ALPHA)
      XX2=L12+XX1
      YY3=L13*DSIN(ALPHA)
      B(K,1)=-1.0D00*YY3
      B(K,2)=YY3
      B(K,3)=0.0D00
      C(K,1)=-1.0D00*XX2
      C(K,2)=XX1
      C(K,3)=L12
      ZX(K)=2.0D00*(ZCC*B(K,1)*B(K,2)+ZBB*B(K,1)*B(K,3)+ZAA*B(K,2)*B(K,3
     +))/(A(K)**2)
      ZY(K)=2.0D00*(ZCC*C(K,1)*C(K,2)+ZBB*C(K,1)*C(K,3)+ZAA*C(K,2)*C(K,3
     +))/(A(K)**2)
   27 ZXY(K)=(ZCC*(B(K,1)*C(K,2)+B(K,2)*C(K,1))+ZBB*(B(K,1)*C(K,3)+B(K,3
     +)*C(K,1))+ZAA*(B(K,2)*C(K,3)+B(K,3)*C(K,2)))/(A(K)**2)
   28 WRITE(6,30) K,I,(B(K,J),J=1,3),(C(K,J),J=1,3),A(K),ZX(K),ZY(K),ZXY
     +(K)
   30 FORMAT(4X,I3,11X,I3,6X,6(F8.4,1X),F9.4,3(2X,F10.7))
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LOAD
C=======================================
C SUBROUTINE LOAD READS IN THE STRUCTURE LOADING DATA, CALCULATES THE
C FP FORCES (BASED ON LUMPED FORCES RATHER THAN A CONSISTENT FORCE
C MATRIX), AND SUBTRACTS THOSE FP FORCES INTO THE GLOBAL JOINT FORCE MATRIX.
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      DIMENSION Q(6)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C5/P(81),GK(81,36)
      COMMON /C6/WX,WY,XS,YS,MX,MY,IDS,IB1,IB2,IB3,IB4
C P(ND) PRE-ZEROING:
      DO 5 I=1,ND
      P(I)=0.0D00
    5 CONTINUE
      READ(5,10) PX,PY,PZ,W
   10 FORMAT(4(1X,F9.5))
      WRITE(6,15) PX,PY,PZ,W
   15 FORMAT(/1X,'LOADING(LBS/IN2):'/3X,'X-DISTR.= ',F9.5/3X,'Y-DISTR.=
     +',F9.5/3X,'Z-DISTR.= ',F9.5/3X,'SELF-WEIGHT= ',F9.5)
C CHECK FOR NO SELF-WEIGHT LOADING:
      IF(W.EQ.0.0) GO TO 35
C CHECK FOR STRUCTURE TYPE OTHER THAN RIGHT CIRCULAR CYLINDER:
      IF(IDS.NE.3) GO TO 35
C
C THE FOLLOWING SECTION CALCULATES THE CONTRIBUTION OF W TO P(ND):
C CONVERT XS FROM DEGREES TO RADIAN:
      XS=XS*3.141592654D00/180.0D00
      DO 30 I=1,MX
      THETA1=XS*(I-1)
```

```
      THETA2=XS*I
      T=W*A(1)/3.0D00
C THE FOLLOWING VALUES OF Q(6) & II APPLY TO "A" ORIENTATION ELEMENTS:
      II=2*I-1
      Q(1)=T*DSIN(THETA1)
      Q(2)=T*DSIN(THETA2)
      Q(3)=Q(1)
      Q(4)=-1.0D00*T*DCOS(THETA1)
      Q(5)=-1.0D00*T*DCOS(THETA2)
      Q(6)=Q(4)
   20 DO 25 J=1,MY
      M=II+(J-1)*2*MX
      JA=(INC(M,1)-1)*9
      JB=(INC(M,2)-1)*9
      JC=(INC(M,3)-1)*9
      P(JA+1)=Q(1)+P(JA+1)
      P(JB+1)=Q(2)+P(JB+1)
      P(JC+1)=Q(3)+P(JC+1)
      P(JA+7)=Q(4)+P(JA+7)
      P(JB+7)=Q(5)+P(JB+7)
      P(JC+7)=Q(6)+P(JC+7)
   25 CONTINUE
      IF(INC(II,4).NE.0) GO TO 30
C THE FOLLOWING VALUES OF Q(6) & II APPLY TO "B" ORIENTATION ELEMENTS:
      II=2*I
      Q(3)=Q(2)
      Q(6)=Q(5)
      GO TO 20
   30 CONTINUE
C CHECK FOR DISTRIBUTED LOADINGS PX,PY,PZ:
      IF(PX.NE.0.0) GO TO 35
      IF(PY.NE.0.0) GO TO 35
```

```
      IF(PZ.EQ.0.0) GO TO 52
C
C THE FOLLOWING SECTION CALCULATES THE CONTRIBUTIONS OF PX,PY,PZ TO P(ND):
   35 DO 50 I=1,NEM
      JA=(INC(I,1)-1)*9
      JB=(INC(I,2)-1)*9
      JC=(INC(I,3)-1)*9
      IF(IDS.EQ.2) GO TO 40
      IF(IDS.EQ.3) GO TO 40
      T=A(I)/3.0D00
      GO TO 45
   40 T=A(1)/3.0D00
   45 P(JA+1)=T*PX+P(JA+1)
      P(JB+1)=T*PX+P(JB+1)
      P(JC+1)=T*PX+P(JC+1)
      P(JA+4)=T*PY+P(JA+4)
      P(JB+4)=T*PY+P(JB+4)
      P(JC+4)=T*PY+P(JC+4)
      P(JA+7)=T*PZ+P(JA+7)
      P(JB+7)=T*PZ+P(JB+7)
      P(JC+7)=T*PZ+P(JC+7)
   50 CONTINUE
   52 WRITE(6,55)
   55 FORMAT(/1X,'TRANSFORMED NODAL FORCES:'/6X,'W1',12X,'WX1',10X,'-WY1
     +',11X,'W2',12X,'WX2',10X,'-WY2',11X,'W3',12X,'WX3',10X,'-WY3')
      WRITE(6,60) (P(I),I=1,ND)
   60 FORMAT((1X,9(F12.3,2X)))
      RETURN
      END
```

```
      SUBROUTINE GKK
C=======================================
C SUBROUTINE GKK CALCULATES THAT PART OF ELEMENT II'S STIFFNESS MATRIX,
C GL(27,27), WHICH INVOLVES ONLY TRANSVERSE CURVATURE TERMS.
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C7/TH,PR,EL
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      BA=B(II,1)
      BB=B(II,2)
      BC=B(II,3)
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
      JJ=1
      ILO=1
      IUP=3
      JLO=1
      JUP=3
      KRO=6
      KCO=6
      KIND=1
C CONSTANT CALCULATION:
      CON=(2.0D00*A(II)/5.04D03)*(ZX(II)**2+2.0D00*PR*ZX(II)*ZY(II)+ZY(I
     +I)**2+2.0D00*(1.0D00-PR)*ZXY(II)**2)*EL*TH/(1.0D00-PR**2)
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1234) CON
 1234 FORMAT(//1X,F16.4/)
C NATNA CALCULATION AND POSITIONING INTO GL(27,27):
      GW(1,1)=CON*4.84D02
   15 GW(1,2)=CON*5.20D01*(CC-CB)
```

```
      GW(1,3)=CON*5.20D01*(BC-BB)
      GW(2,2)=CON*(7.75D00*(CC-CB)**2+6.0D00*CC*CB)
      GW(3,3)=CON*(7.75D00*(BC-BB)**2+6.0D00*BC*BB)
      GW(2,3)=CON*(7.75D00*(CC*BC+CB*BB)-4.75D00*(BB*CC+BC*CB))
      GW(2,1)=GW(1,2)
      GW(3,1)=GW(1,3)
      GW(3,2)=GW(2,3)
      CALL ASMGL
      GO TO (20,25,30),JJ
C NBTNB CALCULATION AND POSITIONING INTO GL(27,27):
   20 JJ=2
      KRO=15
      KCO=15
      BA=BB
      BB=BC
      BC=B(II,1)
      CA=CB
      CB=CC
      CC=C(II,1)
      GO TO 15
C NCTNC CALCULATION AND POSITIONING INTO GL(27,27):
   25 JJ=3
      KRO=24
      KCO=24
      BA=BB
      BB=BC
      BC=B(II,2)
      CA=CB
      CB=CC
      CC=C(II,2)
      GO TO 15
C NATNB CALCULATION AND POSITIONING INTO GL(27,27):
```

```
   30 JJ=4
      KRO=6
      KCO=15
      BA=B(II,1)
      BB=B(II,2)
      BC=B(II,3)
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
      GW(1,1)=CON*1.78D02
   35 GW(1,2)=CON*(1.9D01*CA-3.4D01*CC)
      GW(1,3)=CON*(1.9D01*BA-3.4D01*BC)
      GW(2,1)=CON*(3.4D01*CC-1.9D01*CB)
      GW(3,1)=CON*(3.4D01*BC-1.9D01*BB)
      GW(2,2)=CON*0.25D00*(1.3D01*CA*CC-1.1D01*CA*CB+1.3D01*CB*CC-2.5D01
     +*CC*CC)
      GW(2,3)=CON*0.25D00*(1.3D01*BA*CC-1.1D01*BA*CB+1.3D01*CB*BC-2.5D01
     +*CC*BC)
      GW(3,3)=CON*0.25D00*(1.3D01*BA*BC-1.1D01*BA*BB+1.3D01*BB*BC-2.5D01
     +*BC*BC)
      GW(3,2)=CON*0.25D00*(1.3D01*CA*BC-1.1D01*CA*BB+1.3D01*BB*CC-2.5D01
     +*BC*CC)
   38 CONTINUE
      CALL ASMGL
      GO TO (30,30,30,40,45,50,55,60,65),JJ
C NBTNA CALCULATION AND POSITICNING INTO GL(27,27):
   40 JJ=5
      KRO=15
      KCO=6
      KIND=2
      GO TO 38
C NBTNC CALCULATION AND POSITICNING INTO GL(27,27):
```

```
   45 JJ=6
      KRO=15
      KCO=24
      KIND=1
      BA=BB
      BB=BC
      BC=B(II,1)
      CA=CB
      CB=CC
      CC=C(II,1)
      GO TO 35
C NCTNB CALCULATION AND POSITIONING INTO GL(27,27):
   50 JJ=7
      KRO=24
      KCO=15
      KIND=2
      GO TO 38
C NATNC CALCULATION AND POSITIONING INTO GL(27,27):
   55 JJ=8
      KRO=6
      KCO=24
      BA=BB
      BB=BC
      BC=B(II,2)
      CA=CB
      CB=CC
      CC=C(II,2)
      GO TO 35
C NCTNA CALCULATION AND POSITIONING INTO GL(27,27):
   60 JJ=9
      KRO=24
      KCO=6
```

```
      KIND=1
      GO TO 38
   65 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ASMGL
C=======================================
C SUBROUTINE ASMGL ASSEMBLES VARIOUS TERMS OF SUROUTINES GKK AND GKF
C INTO THE LOCAL STIFFNESS MATRIX, GL(27,27).
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      IF(KIND.EQ.2) GO TO 15
      DO 10 I=ILO,IUP
      DO 10 J=JLO,JUP
      KA=KRO+I
      KB=KCO+J
   10 GL(KA,KB)=GW(I,J)+GL(KA,KB)
      GO TO 25
   15 DO 20 I=ILO,IUP
      DO 20 J=JLO,JUP
      KA=KRO+I
      KB=KCO+J
   20 GL(KA,KB)=GW(J,I)+GL(KA,KB)
   25 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GKF
C=======================================
C SUBROUTINE GKF CALCULATES THAT PART OF ELEMENT II'S STIFFNESS MATRIX,
C GL(27,27), WHICH INVOLVES ONLY FLEXURE TERMS.
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      DIMENSION BFFF(9,9),RCBF(9,9)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C7/TH,PR,EL
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      COMMON /C10/RC(21,21),RBBM(21,6),QC(7,7)
      CON=EL*TH**3/(2.304D03*A(II)**3*(1.0D00-PR**2))
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1234) CON
 1234 FORMAT(//1X,F16.4/)
C RCBF(9,9) PRE-ZEROING:
      DO 5 I=1,9
      DO 5 J=1,9
    5 RCBF(I,J)=0.0D00
C QC(3,3) CALCULATION:
      QC(1,1)=2.0D00*CON
      QC(2,2)=2.0D00*CON
      QC(3,3)=2.0D00*CON
      QC(1,2)=1.0D00*CON
      QC(1,3)=1.0D00*CON
      QC(2,3)=1.0D00*CON
      QC(2,1)=1.0D00*CON
      QC(3,1)=1.0D00*CON
      QC(3,2)=1.0D00*CON
C RC(9,9) CALCULATION:
      DO 10 I=1,3
      IA=I+3
```

```
      IAA=I+6
      DO 10 J=1,3
      JA=J+3
      JAA=J+6
      RC(I,J)=QC(I,J)
      RC(IA,JA)=QC(I,J)
   10 RC(IAA,JAA)=(1.0D00-PR)*2.0D00*QC(I,J)
      DO 15 I=1,3
      IA=I+3
      DO 15 J=1,3
      JA=J+3
      RC(I,JA)=PR*QC(I,J)
   15 RC(IA,J)=PR*QC(I,J)
      DO 20 I=1,6
      DO 20 J=7,9
      RC(I,J)=0.0D00
   20 RC(J,I)=0.0D00
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,5000) ((RC(I,J),J=1,9),I=1,9)
C
C BFFF(9,9) CALCULATIONS:
C BFFF(C,A) CALCULATIONS:
      JC=0
      JJ=1
      BA=B(II,1)
      BB=B(II,2)
      BC=B(II,3)
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
      KA=7
      KB=8
```

```
      KC=9
   30 BFFF(KA,JC+1)=-2.0D00*(2.0D00*CA*BA+CB*BB+CC*BC)
      BFFF(KB,JC+1)=(2.0D00*CA*(BA-BB)-2.0D00*CB*BA)
      BFFF(KC,JC+1)=(2.0D00*CA*(BA-BC)-2.0D00*CC*BA)
      K=2
   35 BFFF(KA,JC+K)=(BB*(2.0D00*CA*CC+0.5D00*CC**2-0.5D00*CB*CC)+BC*(0.5
     +D00*CB*CC-2.0D00*CA*CB-0.5D00*CB**2))
      BFFF(KB,JC+K)=(BA*(2.0D00*CA*CC+0.5D00*CC**2-0.5D00*CB*CC)+3.5D00*
     +BC*CA*(CC-CB))
      BFFF(KC,JC+K)=(BA*(0.5D00*CB*CC-2.0D00*CA*CB-0.5D00*CB**2)+0.5D00*
     +BB*CA*(CC-CB))
      IF(K.EQ.3) GO TO 40
      K=3
      UA=BA
      UB=BB
      UC=BC
      BA=CA
      BB=CB
      BC=CC
      CA=UA
      CB=UB
      CC=UC
      GO TO 35
   40 GO TO (45,50,55),JJ
C BFFF(C,B) CALCULATIONS:
   45 JJ=2
      CA=C(II,2)
      CB=C(II,3)
      CC=C(II,1)
      BA=B(II,2)
      BB=B(II,3)
      BC=B(II,1)
```

```
      KA=8
      KB=9
      KC=7
      JC=3
      GO TO 30
C BFFF(C,C) CALCULATIONS:
   50 JJ=3
      CA=C(II,3)
      CB=C(II,1)
      CC=C(II,2)
      BA=B(II,3)
      BB=B(II,1)
      BC=B(II,2)
      KA=9
      KB=7
      KC=8
      JC=6
      GO TO 30
C BFFF(A,A) CALCULATIONS:
   55 JJ=1
      BA=B(II,1)
      BB=B(II,2)
      BC=B(II,3)
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
      KA=1
      KB=2
      KC=3
      JC=0
   60 BFFF(KA,JC+1)=-2.0D00*(2.0D00*BA**2+BB**2+BC**2)
      BFFF(KB,JC+1)=2.0D00*BA*(BA-2.0D00*BB)
```

```
      BFFF(KC,JC+1)=2.0D00*BA*(BA-2.0D00*BC)
      BFFF(KA,JC+2)=(BB*BC*(CC-CB)+8.0D00*A(II)*BA)
      BFFF(KB,JC+2)=(BA*CC*(2.0D00*BA+BC)-CB*BA*BC)
      BFFF(KC,JC+2)=(BA*BB*CC-BA*CB*(2.0D00*BA+BB))
      BFFF(KA,JC+3)=BB*BC*(BC-BB)
      BFFF(KB,JC+3)=BA*BC*(2.0D00*BA-BB+BC)
      BFFF(KC,JC+3)=BA*BB*(BC-2.0D00*BA-BB)
      GO TO (65,70,75),JJ
C BFFF(A,B) CALCULATIONS:
   65 JJ=2
      KA=2
      KB=3
      KC=1
      JC=3
      BA=BB
      BB=BC
      BC=B(II,1)
      CA=CB
      CB=CC
      CC=C(II,1)
      GO TO 60
C BFFF(A,C) CALCULATIONS:
   70 JJ=3
      KA=3
      KB=1
      KC=2
      JC=6
      BA=BB
      BB=BC
      BC=B(II,2)
      CA=CB
      CB=CC
```

```
      CC=C(II,2)
      GO TO 60
C BFFF(B,A) CALCULATIONS:
   75 JJ=1
      KA=4
      KB=5
      KC=6
      JC=0
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
      BA=B(II,1)
      BB=B(II,2)
      BC=B(II,3)
   78 BFFF(KA,JC+1)=-2.0D00*(2.0D00*CA**2+CB**2+CC**2)
      BFFF(KB,JC+1)=2.0D00*CA*(CA-2.0D00*CB)
      BFFF(KC,JC+1)=2.0D00*CA*(CA-2.0D00*CC)
      BFFF(KA,JC+2)=CB*CC*(CC-CB)
      BFFF(KB,JC+2)=CA*CC*(2.0D00*CA-CB+CC)
      BFFF(KC,JC+2)=CA*CB*(CC-2.0D00*CA-CB)
      BFFF(KA,JC+3)=CB*CC*(BC-BB)-8.0D00*A(II)*CA
      BFFF(KB,JC+3)=CA*BC*(2.0D00*CA+CC)-BB*CA*CC
      BFFF(KC,JC+3)=CA*CB*BC-CA*BB*(2.0D00*CA+CB)
      GO TO (80,85,90),JJ
C BFFF(B,B) CALCULATIONS:
   80 JJ=2
      KA=5
      KB=6
      KC=4
      JC=3
      CA=CB
      CB=CC
```

```
      CC=C(II,1)
      BA=BB
      BB=BC
      BC=B(II,1)
      GO TO 78
C BFFF(B,C) CALCULATIONS:
   85 JJ=3
      KA=6
      KB=4
      KC=5
      JC=6
      CA=CB
      CB=CC
      CC=C(II,2)
      BA=BB
      BB=BC
      BC=B(II,2)
      GO TO 78
C
C RCBF(9,9) CALCULATIONS:
   90 CONTINUE
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,5000) ((BFFF(I,J),J=1,9),I=1,9)
      DO 92 IRO=1,9
      DO 92 ICO=1,9
      DO 92 I=1,9
   92 RCBF(IRO,ICO)=RCBF(IRO,ICO)+RC(IRO,I)*BFFF(I,ICO)
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,5000) ((RCBF(I,J),J=1,9),I=1,9)
 5000 FORMAT(/(1X,9(F14.2)))
C
C GW(9,9) PRE-ZEROING:
```

```
      DO 94 I=1,9
      DO 94 J=1,9
   94 GW(I,J)=0.0D00
C GW(9,9) CALCULATION:
      DO 96 IRO=1,9
      DO 96 ICO=1,9
      DO 96 I=1,9
   96 GW(IRO,ICO)=GW(IRO,ICO)+BFFF(I,IRO)*RCBF(I,ICO)
C GW(9,9) POSITIONING INTO GL(27,27):
      KIND=1
      DO 100 I=1,3
      ILO=3*(I-1)+1
      IUP=ILO+2
      KRO=7+9*(I-1)-ILO
      DO 100 J=1,3
      JLO=3*(J-1)+1
      JUP=JLO+2
      KCO=7+9*(J-1)-JLO
      CALL ASMGL
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GKM
C=======================================
C SUBROUTINE GKM CALCULATES THAT PART OF ELEMENT II'S STIFFNESS MATRIX,
C GL(27,27), WHICH INVOLVES ONLY MEMBRANE ACTION TERMS.
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C7/TH,PR,EL
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      COMMON /C9/BMA(21,6),BMB(21,6),BMC(21,6)
      COMMON /C10/RB(21,21),RBBM(21,6),QB(7,7)
C BMA(21,3),BMB(21,3),BMC(21,3) CLACULATIONS:
C BMA, BMB, BMC PARTIAL PRE-ZEROING:
      DO 5 I=1,7
      DO 5 J=4,6
      BMA(I,J)=0.0D00
      BMB(I,J)=0.0D00
      BMC(I,J)=0.0D00
    5 CONTINUE
      DO 6 J=8,14
      DO 6 I=1,3
      BMA(J,I)=0.0D00
      BMB(J,I)=0.0D00
      BMC(J,I)=0.0D00
    6 CONTINUE
C CONSTANT CALCULATION:
      CON=EL*TH/(7.2D02*A(II)*(1-PR**2))
      WRITE(6,1234) CON
 1234 FORMAT(//1X,F16.4/)
C NAX PART CALCULATION OF BMA(21,6):
      BA=B(II,1)
      BB=B(II,2)
```

```
      BC=B(II,3)
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
      I=1
      J=1
   10 KA=I+4
      KB=I+5
      KC=I+6
      DO 15 KK=1,4
      K=KK-1
      DO 15 M=1,2
   15 BMA(I+K,J+M)=0.0D00
      DO 20 K=1,3
   20 BMA(I+K,J)=-1.0D00*BA
      BMA(I+1,J+1)=2.0D00*A(II)
      BMA(I,J)=BA
      BMA(KA,J)=2.0D00*(BA-BB)
      BMA(KC,J)=2.0D00*(BA-BC)
      BMA(KB,J)=0.0D00
      BMA(KA,J+1)=(2.0D00*BA*CC+0.5D00*BC*(CC-CB))
      BMA(KA,J+2)=(2.0D00*BA*BC+0.5D00*BC*(BC-BB))
      BMA(KB,J+1)=0.5D00*BA*(CC-CB)
      BMA(KB,J+2)=0.5D00*BA*(BC-BB)
      BMA(KC,J+1)=(-2.0D00*BA*CB+0.5D00*BB*(CC-CB))
      BMA(KC,J+2)=(-2.0D00*BA*BB+0.5D00*BB*(BC-BB))
      IF(I.EQ.15) GO TO 25
      I=15
      J=4
      GO TO 10
C NBX PART CALCULATION OF BMB:
   25 BA=BB
```

```
      BB=BC
      BC=B(II,1)
      CA=CB
      CB=CC
      CC=C(II,1)
      I=1
      J=1
   30 KA=I+5
      KB=I+6
      KC=I+4
      DO 32 KK=1,4
      K=KK-1
      DO 32 M=1,2
   32 BMB(I+K,J+M)=0.0D00
      DO 34 K=1,3
   34 BMB(I+K,J)=-1.0D00*BA
      BMB(I+2,J+1)=2.0D00*A(II)
      BMB(I,J)=BA
      BMB(KA,J)=2.0D00*(BA-BB)
      BMB(KC,J)=2.0D00*(BA-BC)
      BMB(KB,J)=0.0D00
      BMB(KA,J+1)=(2.0D00*BA*CC+0.5D00*BC*(CC-CB))
      BMB(KA,J+2)=(2.0D00*BA*BC+0.5D00*BC*(BC-BB))
      BMB(KB,J+1)=0.5D00*BA*(CC-CB)
      BMB(KB,J+2)=0.5D00*BA*(BC-BB)
      BMB(KC,J+1)=(-2.0D00*BA*CB+0.5D00*BB*(CC-CB))
      BMB(KC,J+2)=(-2.0D00*BA*BB+0.5D00*BB*(BC-BB))
      IF(I.EQ.15) GO TO 35
      I=15
      J=4
      GO TO 30
C NCX PART CALCULATION OF BMC
```

```
35 BA=BB
   BB=BC
   BC=B(II,2)
   CA=CB
   CB=CC
   CC=C(II,2)
   I=1
   J=1
40 KA=I+6
   KB=I+4
   KC=I+5
   DO 42 KK=1,4
   K=KK-1
   DO 42 M=1,2
42 BMC(I+K,J+M)=0.0D00
   DO 44 K=1,3
44 BMC(I+K,J)=-1.0D00*BA
   BMC(I+3,J+1)=2.0D00*A(II)
   BMC(I,J)=BA
   BMC(KA,J)=2.0D00*(BA-BB)
   BMC(KC,J)=2.0D00*(BA-BC)
   BMC(KB,J)=0.0D00
   BMC(KA,J+1)=(2.0D00*BA*CC+0.5D00*BC*(CC-CB))
   BMC(KA,J+2)=(2.0D00*BA*BC+0.5D00*BC*(BC-BB))
   BMC(KB,J+1)=0.5D00*BA*(CC-CB)
   BMC(KB,J+2)=0.5D00*BA*(BC-BB)
   BMC(KC,J+1)=(-2.0D00*BA*CB+0.5D00*BB*(CC-CB))
   BMC(KC,J+2)=(-2.0D00*BA*BB+0.5D00*BB*(BC-BB))
   IF(I.EQ.15) GO TO 45
   I=15
   J=4
   GO TO 40
```

```
C NBY PART CALCULATION OF BMA:
   45 I=8
      J=4
      BA=B(II,1)
      BB=B(II,2)
      BC=B(II,3)
      CA=C(II,1)
      CB=C(II,2)
      CC=C(II,3)
   50 KA=I+4
      KB=I+5
      KC=I+6
      DO 55 KK=1,4
      K=KK-1
      DO 55 M=1,2
   55 BMA(I+K,J+M)=0.0D00
      DO 60 K=1,3
   60 BMA(I+K,J)=-1.0D00*CA
      BMA(I+1,J+2)=2.0D00*A(II)
      BMA(I,J)=CA
      BMA(KA,J)=2.0D00*(CA-CB)
      BMA(KC,J)=2.0D00*(CA-CC)
      BMA(KB,J)=0.0D00
      BMA(KA,J+1)=(2.0D00*CA*CC+0.5D00*CC*(CC-CB))
      BMA(KA,J+2)=(2.0D00*CA*BC+0.5D00*CC*(BC-BB))
      BMA(KB,J+1)=0.5D00*CA*(CC-CB)
      BMA(KB,J+2)=0.5D00*CA*(BC-BB)
      BMA(KC,J+1)=(-2.0D00*CA*CB+0.5D00*CB*(CC-CB))
      BMA(KC,J+2)=(-2.0D00*CA*BB+0.5D00*CB*(BC-BB))
      IF(I.EQ.15) GO TO 65
      I=15
      J=1
```

```
      GO TO 50
C NBY PART OF CALCULATION OF BMB:
   65 I=8
      J=4
      BA=BB
      BB=BC
      BC=B(II,1)
      CA=CB
      CB=CC
      CC=C(II,1)
   70 KA=I+5
      KB=I+6
      KC=I+4
      DO 72 KK=1,4
      K=KK-1
      DO 72 M=1,2
   72 BMB(I+K,J+M)=0.0D00
      DO 76 K=1,3
   76 BMB(I+K,J)=-1.0D00*CA
      BMB(I+2,J+2)=2.0D00*A(II)
      BMB(I,J)=CA
      BMB(KA,J)=2.0D00*(CA-CB)
      BMB(KC,J)=2.0D00*(CA-CC)
      BMB(KB,J)=0.0D00
      BMB(KA,J+1)=(2.0D00*CA*CC+0.5D00*CC*(CC-CB))
      BMB(KA,J+2)=(2.0D00*CA*BC+0.5D00*CC*(BC-BB))
      BMB(KB,J+1)=0.5D00*CA*(CC-CB)
      BMB(KB,J+2)=0.5D00*CA*(BC-BB)
      BMB(KC,J+1)=(-2.0D00*CA*CB+0.5D00*CB*(CC-CB))
      BMB(KC,J+2)=(-2.0D00*CA*BB+0.5D00*CB*(BC-BB))
      IF(I.EQ.15) GO TO 80
      I=15
```

```
      J=1
      GO TO 70
C NBY PART CALCULATION OF BMC:
   80 I=8
      J=4
      BA=BB
      BB=BC
      BC=B(II,2)
      CA=CB
      CB=CC
      CC=C(II,2)
   85 KA=I+6
      KB=I+4
      KC=I+5
      DO 88 KK=1,4
      K=KK-1
      DO 88 M=1,2
   88 BMC(I+K,J+M)=0.0D00
      DO 92 K=1,3
   92 BMC(I+K,J)=-1.0D00*CA
      BMC(I+3,J+2)=2.0D00*A(II)
      BMC(I,J)=CA
      BMC(KA,J)=2.0D00*(CA-CB)
      BMC(KC,J)=2.0D00*(CA-CC)
      BMC(KB,J)=0.0D00
      BMC(KA,J+1)=(2.0D00*CA*CC+0.5D00*CC*(CC-CB))
      BMC(KA,J+2)=(2.0D00*CA*BC+0.5D00*CC*(BC-BB))
      BMC(KB,J+1)=0.5D00*CA*(CC-CB)
      BMC(KB,J+2)=0.5D00*CA*(BC-BB)
      BMC(KC,J+1)=(-2.0D00*CA*CB+0.5D00*CB*(CC-CB))
      BMC(KC,J+2)=(-2.0D00*CA*BB+0.5D00*CB*(BC-BB))
      IF(I.EQ.15) GO TO 95
```

```
      I=15
      J=1
      GO TO 85
C
C QB(7,7) CALCULATIONS:
   95 CONTINUE
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,3000) ((BMA(I,J),J=1,6),I=1,21)
      WRITE(6,3000) ((BMB(I,J),J=1,6),I=1,21)
      WRITE(6,3000) ((BMC(I,J),J=1,6),I=1,21)
 3000 FORMAT(/(1X,6(4X,F16.4)))
      DO 100 I=2,4
      QB(1,I)=3.0D01*CON
  100 QB(I,1)=3.0D01*CON
      DO 105 I=5,7
      QB(1,I)=1.5D01*CON
  105 QB(I,1)=1.5D01*CON
      QB(1,1)=1.8D02*CON
      DO 115 I=2,4
      DO 110 J=2,4
  110 QB(I,J)=2.0D00*CON
  115 QB(I,I)=1.2D01*CON
      DO 122 I=5,7
      DO 120 J=5,7
  120 QB(I,J)=1.0D00*CON
  122 QB(I,I)=2.0D00
      DO 130 I=2,4
      DO 130 J=5,7
      QB(I,J)=3.0D00*CON
  130 QB(J,I)=3.0D00*CON
      QB(2,6)=1.0D00*CON
      QB(6,2)=1.0D00*CON
```

```
      QB(3,7)=1.0D00*CON
      QB(7,3)=1.0D00*CON
      QB(4,5)=1.0D00*CON
      QB(5,4)=1.0D00*CON
C
C RB(21,21) CALCULATIONS:
      DO 135 I=1,7
      IA=I+7
      IAA=I+14
      DO 135 J=1,7
      JA=J+7
      JAA=J+14
      RB(I,J)=QB(I,J)
      RB(IA,JA)=QB(I,J)
  135 RB(IAA,JAA)=QB(I,J)*(1.0D00-PR)/2.0D00
      DO 140 I=1,7
      IA=I+7
      DO 140 J=1,7
      JA=J+7
      RB(I,JA)=PR*QB(I,J)
  140 RB(IA,J)=PR*QB(I,J)
      DO 145 I=1,14
      DO 145 J=15,21
      RB(I,J)=0.0D00
  145 RB(J,I)=0.0D00
C  TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,4000) ((RB(I,J),J=1,7),I=1,21)
      WRITE(6,4000) ((RB(I,J),J=8,14),I=1,21)
      WRITE(6,4000) ((RB(I,J),J=15,21),I=1,21)
 4000 FORMAT(/(1X,7(F16.4)))
C
C RBBM(21,6) PRE-ZEROING:
```

```
      DO 148 I=1,21
      DO 148 J=1,6
  148 RBBM(I,J)=0.0D00
C RB-BMC=RBBM(21,6) CALCULATIONS:
      DO 150 IRO=1,21
      DO 150 ICO=1,6
      DO 150 I=1,21
  150 RBBM(IRO,ICO)=RBBM(IRO,ICO)+RB(IRO,I)*BMC(I,ICO)
C BMC-RB-BMC CALCULATIONS AND POSITIONING INTO GL(27,27):
      DO 155 IRO=1,6
      IRA=IRO+18
      DO 155 ICO=1,6
      ICA=ICO+18
      DO 155 I=1,21
  155 GL(IRA,ICA)=GL(IRA,ICA)+BMC(I,IRO)*RBBM(I,ICO)
C BMB-RB-BMC AND BMC-RB-BMB CALCULATIONS AND POSITIONING INTO GL(27,27):
      DO 160 IRO=1,6
      IRA=IRO+9
      DO 160 ICO=1,6
      ICA=ICO+18
      DO 160 I=1,21
      GL(IRA,ICA)=GL(IRA,ICA)+BMB(I,IRO)*RBBM(I,ICO)
  160 GL(ICA,IRA)=GL(IRA,ICA)
C BMA-RB-BMC AND BMC-RB-BMA CALCULATIONS AND POSITIONING INTO GL(27,27):
      DO 165 IRO=1,6
      DO 165 ICO=1,6
      ICA=ICO+18
      DO 165 I=1,21
      GL(IRO,ICA)=GL(IRO,ICA)+BMA(I,IRO)*RBBM(I,ICO)
  165 GL(ICA,IRO)=GL(IRO,ICA)
C
C RBBM(21,6) PRE-ZEROING:
```

```
      DO 170 I=1,21
      DO 170 J=1,6
  170 RBBM(I,J)=0.0D00
C RB-BMB=RBBM(21,6) CALCULATIONS:
      DO 175 IRO=1,21
      DO 175 ICO=1,6
      DO 175 I=1,21
  175 RBBM(IRO,ICO)=RBBM(IRO,ICO)+RB(IRO,I)*BMB(I,ICO)
C BMB-RB-BMB CALCULATIONS AND POSITIONING INTO GL(27,27):
      DO 180 IRO=1,6
      IRA=IRO+9
      DO 180 ICO=1,6
      ICA=ICO+9
      DO 180 I=1,21
  180 GL(IRA,ICA)=GL(IRA,ICA)+BMB(I,IRO)*RBBM(I,ICO)
C BMA-RB-BMB AND BMB-RB-BMA CALCULATIONS AND POSITIONING INTO GL(27,27):
      DO 185 IRO=1,6
      DO 185 ICO=1,6
      ICA=ICO+9
      DO 185 I=1,21
      GL(IRO,ICA)=GL(IRO,ICA)+BMA(I,IRO)*RBBM(I,ICO)
  185 GL(ICA,IRO)=GL(IRO,ICA)
C
C RBBM(21,6 PRE-ZEROING:
      DO 190 I=1,21
      DO 190 J=1,6
  190 RBBM(I,J)=0.0D00
C RB-BMA CALCULATIONS:
      DO 195 IRO=1,21
      DO 195 ICO=1,6
      DO 195 I=1,21
  195 RBBM(IRO,ICO)=RBBM(IRO,ICO)+RB(IRO,I)*BMA(I,ICO)
```

```
C BMA-RB-BMA CALCULATIONS AND POSITIONING INTO GL(27,27):
      DO 200 IRO=1,6
      DO 200 ICO=1,6
      DO 200 I=1,21
  200 GL(IRO,ICO)=GL(IRO,ICO)+BMA(I,IRO)*RBBM(I,ICO)
      RETURN
      END
```

```
      SUBROUTINE GKKM
C=====================================
C SUBROUTINE GKKM CALCULATES THAT PART OF ELEMENT II'S STIFFNESS MATRIX
C WHICH INVOLVES TRANSVERSE CURVATURE AND MEMBRANE INTERACTION TERMS.
C=====================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      DIMENSION RBMB(18,6),RBMC(18,6),BK(18,3)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C7/TH,PR,EL
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      COMMON /C9/BMA(21,6),BMB(21,6),BMC(21,6)
      COMMON /C10/RB(21,21),RBBM(21,6),QB(7,7)
C RBBM(18,6), RBMB(18,6) AND RBMC(18,6) PRE-ZEROING:
      DO 5 I=1,18
      DO 5 J=1,6
      RBMB(I,J)=0.0D00
      RBMC(I,J)=0.0D00
    5 RBBM(I,J)=0.0D00
C CONSTANT CALCULATION:
      CON=EL*TH/(-2.52D03*(1.0D00-PR**2))
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1234) CON
 1234 FORMAT(//1X,F16.4/)
C
C PARTIAL BK(18,3) CALCULATIONS:
      DO 10 I=1,13,6
      BK(I+5,1)=0.0D00
      DO 10 J=2,3
      BK(I,J)=0.0D00
      DO 10 K=3,4
   10 BK(I+K,J)=0.0D00
      DO 15 I=1,3
```

```
      BK(I,1)=CON*ZX(II)
      BK(I+6,1)=CON*ZY(II)
   15 BK(I+12,1)=CON*ZXY(II)
      DO 20 I=4,5
      BK(I,1)=-1.0D00*CON*ZX(II)
      BK(I+6,1)=-1.0D00*CON*ZY(II)
   20 BK(I+12,1)=-1.0D00*CON*ZXY(II)
C
C QAA=QB(6,7) CALCULATION:
      QB(1,1)=4.2D02
      QB(1,2)=1.26DC2
      QB(1,6)=2.1D01
      QB(6,1)=2.1D01
      QB(1,7)=4.2D01
      QB(2,1)=4.2D01
      DO 40 I=3,5
      QB(1,I)=4.2D01
   40 QB(I,1)=4.2D01
      DO 45 I=2,4
   45 QB(6,I)=3.0D00
      DO 50 I=5,7
   50 QB(6,I)=2.0D00
      QB(2,2)=1.2D01
      QB(3,2)=1.2D01
      QB(4,3)=1.2D01
      QB(5,4)=1.2D01
      QB(2,7)=3.0D00
      QB(3,5)=3.0D00
      QB(4,6)=3.0D00
      QB(5,6)=3.0D00
      QB(5,3)=2.0D00
      QB(4,7)=2.0D00
```

```
      QB(3,6)=2.0D00
      QB(2,6)=2.0D00
      QB(2,4)=2.0D00
      DO 55 I=3,5
      QB(I,I)=2.0D00
   55 QB(I-1,I)=6.0D00
      QB(2,5)=6.0D00
      QB(3,7)=6.0D00
      QB(5,7)=6.0D00
      QB(5,2)=6.0D00
      QB(4,2)=6.0D00
C
C BKA-RBA-BMX CALCULATIONS AND POSITIONING INTO GL(27,27):
      JJ=1
      RA=6
      INDA=2
      INDB=3
C BKX=BK(18,3) CALCULATIONS:
   60 CB=-1.0D00*C(II,INDA)
      CC=C(II,INDB)
      J=2
   61 BK(2,J)=CON*CC*ZX(II)
      BK(8,J)=CON*CC*ZY(II)
      BK(14,J)=CON*CC*ZXY(II)
      BK(3,J)=CON*CB*ZX(II)
      BK(9,J)=CON*CB*ZY(II)
      BK(15,J)=CON*CB*ZXY(II)
      BK(6,J)=CON*ZX(II)*(CC+CB)/2.0D00
      BK(12,J)=CON*ZY(II)*(CC+CB)/2.0D00
      BK(18,J)=CON*ZXY(II)*(CC+CB)/2.0D00
      IF(J.EQ.3) GO TO 62
      J=3
```

```
      CB=-1.0D00*B(II,INDA)
      CC=B(II,INDB)
      GO TO 61
   62 CONTINUE
C  TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1100)((BK(I,J),J=1,3),I=1,18)
 1100 FORMAT(/(1X,3(4X,F16.4)))
C  RBX=RB(18,21) CALCULATION:
      DO 65 I=1,6
      IA=I+6
      IAA=I+12
      DO 65 J=1,7
      JA=J+7
      JAA=J+14
      RB(I,J)=QB(I,J)
      RB(IA,JA)=QB(I,J)
   65 RB(IAA,JAA)=(1.0D00-PR)*QB(I,J)
      DO 70 I=1,6
      IA=I+6
      DO 70 J=1,7
      JA=J+7
      RB(I,JA)=PR*QB(I,J)
   70 RB(IA,J)=PR*QB(I,J)
      DO 75 I=1,12
      DO 75 J=15,21
   75 RB(I,J)=0.0D00
      DO 80 I=13,18
      DO 80 J=1,14
   80 RB(I,J)=0.0D00
C  TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,5000) ((RB(I,J),J=1,9),I=1,21)
      WRITE(6,5000) ((RB(I,J),J=10,18),I=1,21)
```

```
 5000 FORMAT(/(1X,9(F14.2)))
C RBX-BMA=RBBM(18,6) CALCULATIONS:
      DO 90 IRO=1,18
      DO 90 ICO=1,6
      DO 90 I=1,21
   90 RBBM(IRO,ICO)=RBBM(IRO,ICO)+RB(IRO,I)*BMA(I,ICO)
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1000)((RBBM(I,J),J=1,6),I=1,18)
C RBX-BMB=RBMB(18,6) CALCULATIONS:
      DO 95 IRO=1,18
      DO 95 ICO=1,6
      DO 95 I=1,21
   95 RBMB(IRO,ICO)=RBMB(IRO,ICO)+RB(IRO,I)*BMB(I,ICO)
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1000)((RBMB(I,J),J=1,6),I=1,18)
C RBX-BMC=RBMC(18,6) CALCULATIONS:
      DO 100 IRO=1,18
      DO 100 ICO=1,6
      DO 100 I=1,21
  100 RBMC(IRO,ICO)=RBMC(IRO,ICO)+RB(IRO,I)*BMC(I,ICO)
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,1000)((RBMC(I,J),J=1,6),I=1,18)
 1000 FORMAT(/(1X,6(4X,F16.4)))
      DO 120 IRO=1,3
      IRA=IRO+RA
      DO 120 ICO=1,6
      DO 118 I=1,18
  118 GL(IRA,ICO)=GL(IRA,ICO)+BK(I,IRO)*RBBM(I,ICO)
  120 GL(ICO,IRA)=GL(IRA,ICO)
      CA=9
      DO 125 IRO=1,3
      IRA=IRO+RA
```

```
      DO 125 ICO=1,6
      ICA=ICO+CA
      DO 122 I=1,18
  122 GL(IRA,ICA)=GL(IRA,ICA)+BK(I,IRO)*RBMB(I,ICO)
  125 GL(ICA,IRA)=GL(IRA,ICA)
      CA=18
      DO 130 IRO=1,3
      IRA=IRO+RA
      DO 130 ICO=1,6
      ICA=ICO+CA
      DO 128 I=1,18
  128 GL(IRA,ICA)=GL(IRA,ICA)+BK(I,IRO)*RBMC(I,ICO)
  130 GL(ICA,IRA)=GL(IRA,ICA)
      GO TO (135,150,165),JJ
C
C BKB-RBB-BMX CALCULATIONS AND POSITIONING INTO GL(27,27):
  135 JJ=2
      RA=15
      INDA=3
      INDB=1
C RBBM(18,6), RBMB(18,6) AND RBMC(18,6) PRE-ZEROING:
      DO 138 I=1,18
      DO 138 J=1,6
      RBMB(I,J)=0.0D00
      RBMC(I,J)=0.0D00
  138 RBBM(I,J)=0.0D00
C QBB=QB(6,7) CALCULATION:
      DO 145 I=1,5
      QB(I,4)=QB(I,3)
      QB(I,3)=QB(I,2)
      QB(I,7)=QB(I,6)
  145 QB(I,6)=QB(I,5)
```

```
      QB(1,2)=4.2D01
      QB(2,2)=2.0D00
      QB(3,2)=6.0D00
      QB(4,2)=2.0D00
      QB(5,2)=1.2D01
      QB(2,5)=3.0D00
      QB(3,5)=6.0D00
      QB(4,5)=2.0D00
      QB(5,5)=6.0D00
      GO TO 60
C
C BKC-RBC-BMX CALCULATIONS AND POSITIONING INTO GL(27,27):
  150 JJ=3
      RA=24
      INDA=1
      INDB=2
C RBBM(18,6), RBMB(18,6) AND RBMC(18,6) PRE-ZEROING:
      DO 152 I=1,18
      DO 152 J=1,6
      RBMB(I,J)=0.0D00
      RBMC(I,J)=0.0D00
  152 RBBM(I,J)=0.0D00
C QBC=QB(6,7) CALCULATION:
      DO 160 I=1,5
      QB(I,4)=QB(I,3)
      QB(I,3)=QB(I,2)
      QB(I,7)=QB(I,6)
  160 QB(I,6)=QB(I,5)
      QB(2,2)=6.0D00
      QB(3,2)=2.0D00
      QB(4,2)=1.2D01
      QB(5,2)=2.0D00
```

```
      QB(1,5)=2.1D01
      QB(2,5)=2.0D00
      QB(3,5)=2.0D00
      QB(4,5)=3.0D00
      QB(5,5)=3.0D00
      GO TO 60
  165 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TRANSF
C=======================================
C SUBROUTINE TRANSF PREFORMS THE MATRIX MANIPULATIONS (FINAL TRANSFORMATION)
C NESCESSARY TO EXPRESS ELEMENT II'S STIFFNESS MATRIX, GL(27,27), IN TERMS OF
C GLOBAL NODAL ROTATIONS INSTEAD OF THE PREVIOUS LOCAL NODAL DERIVATIVES.
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      DIMENSION GT(27,27)
      COMMON /C4/B(2,3),C(2,3),A(2),ZX(2),ZY(2),ZXY(2)
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      ZXX=ZX(II)
      ZYY=ZY(II)
C
C GL-T=GT(27,27) CALCULATIONS:
C INITIALIZE GT(27,27)=GL(27,27)
      DO 5 I=1,27
      DO 5 J=1,27
    5 GT(I,J)=GL(I,J)
      DO 10 I=1,19,9
      IA=I+7
      DO 10 J=1,27
   10 GT(J,I)=GT(J,I)-ZXX*GL(J,IA)
      DO 15 I=2,20,9
      IA=I+7
      DO 15 J=1,27
   15 GT(J,I)=GT(J,I)+ZYY*GL(J,IA)
C
C TT-GL-T=GL(27,27) CALCULATIONS:
C INITIALIZE GL(27,27)=GT(27,27):
      DO 20 I=1,27
      DO 20 J=1,27
   20 GL(I,J)=GT(I,J)
```

```
      DO 25 I=1,19,9
      IA=I+7
      DO 25 J=1,27
   25 GL(I,J)=GL(I,J)-ZXX*GT(IA,J)
      DO 30 I=2,20,9
      IA=I+7
      DO 30 J=1,27
   30 GL(I,J)=GL(I,J)+ZYY*GT(IA,J)
      RETURN
      END
```

```
      SUBROUTINE ASEMBL
C=====================================
C SUBROUTINE ASEMBL PLACES THE STIFFNESS MATRIX OF ELEMENT II, AND ALL
C ELEMENTS SIMILAR TO ELEMENT II, INTO THE GLOBAL (BANDED, RECTANGULAR)
C STIFFNESS MATRIX, GK(ND,IBW).
C=====================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C5/P(81),GK(81,36)
      COMMON /C8/GL(27,27),GW(9,9),II,KRO,KCO,ILO,IUP,JLO,JUP,KIND
      DO 60 NN=1,NEM
      IF(INC(NN,5).NE.II) GO TO 60
      KA=9*(INC(NN,1)-1)
      KB=9*(INC(NN,2)-1)
      KC=9*(INC(NN,3)-1)
C GL-AA(9,9) POSITIONING INTO GK(ND,IBW):
      DO 10 I=1,9
      IRO=KA+I
      DO 10 J=I,9
      ICO=(KA+J)-IRO+1
   10 GK(IRO,ICO)=GK(IRO,ICO)+GL(I,J)
C GL-BB(9,9) POSITIONING INTO GK(ND,IBW):
      DO 15 I=1,9
      IA=I+9
      IRO=KB+I
      DO 15 J=I,9
      JA=J+9
      ICO=(KB+J)-IRO+1
   15 GK(IRO,ICO)=GK(IRO,ICO)+GL(IA,JA)
C GL-CC(9,9) POSITIONING INTO GK(ND,IBW):
      DO 20 I=1,9
```

```
      IA=I+18
      IRO=KC+I
      DO 20 J=I,9
      JA=J+18
      ICO=(KC+J)-IRO+1
   20 GK(IRO,ICO)=GK(IRO,ICO)+GL(IA,JA)
C GL-AB(9,9) POSITIONING INTO GK(ND,IBW):
      DO 25 I=1,9
      IRO=KA+I
      DO 25 J=1,9
      JA=J+9
      ICO=(KB+J)-IRO+1
   25 GK(IRO,ICO)=GK(IRO,ICO)+GL(I,JA)
      IF(INC(NN,4).NE.0) GO TO 42
C GL-AC(9,9) POSITIONING INTO GK(ND,IBW):
      DO 30 I=1,9
      IRO=KA+I
      DO 30 J=1,9
      JA=J+18
      ICO=(KC+J)-IRO+1
   30 GK(IRO,ICO)=GK(IRO,ICO)+GL(I,JA)
C GL-BC(9,9) POSITIONING INTO GK(ND,IBW):
      DO 40 I=1,9
      IA=I+9
      IRO=KB+I
      DO 40 J=1,9
      JA=J+18
      ICO=(KC+J)-IRO+1
   40 GK(IRO,ICO)=GK(IRO,ICO)+GL(IA,JA)
      GO TO 60
   42 CONTINUE
C GL-CA(9,9) POSITIONING INTO GK(ND,IBW)
```

```
      DO 45 I=1,9
      IA=I+18
      IRO=KC+I
      DO 45 J=1,9
      ICO=(KA+J)-IRO+1
   45 GK(IRO,ICO)=GK(IRO,ICO)+GL(IA,J)
C GL-CB(9,9) POSITIONING INTO GK(ND,IBW)
      DO 50 I=1,9
      IRO=KC+I
      IA=I+18
      DO 50 J=1,9
      JA=J+9
      ICO=(KB+J)-IRO+1
   50 GK(IRO,ICO)=GK(IRO,ICO)+GL(IA,JA)
   60 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CONSTR
C=======================================
C SUBROUTINE CONSTR FORMS THE CONSTRAINED SYSTEM STIFFNESS MATRIX, GK(ND,IBW),
C AND THE CONSTRAINED APPLIED FORCE VECTOR, P(ND), BY PLACING "ONES" ALONG THE
C MAIN DIAGONAL OF GK(ND,IBW) AND "ZEROES" ELSEWHERE ON THE ROWS AND COLUMNS
C OF GK(ND,IBW), AND THE ROWS OF P(ND), THAT CORRESPOND TO CONSTRAINED NODAL
C MOTIONS.
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C5/P(81),GK(81,36)
      DO 20 I=1,ND
      IF(ICON(I).EQ.0) GO TO 20
C "ZERO" ROW:
      DO 10 J=2,IBW
   10 GK(I,J)=0.0D00
      P(I)=0.0D00
C "ZERO" COLUMN:
      ILIM=I
C CHECK TO DETERMINE IF "I" COLUMNS MAY BE ELIMINATED:
      IF(I.GT.IBW) ILIM=IBW
      DO 15 J=1,ILIM
      IROW=I-J+1
   15 GK(IROW,J)=0.0D00
C "ONE" IN DIAGONAL:
      GK(I,1)=1.0D00
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BNDSLV
C=======================================
C SUBROUTINE BNDSLV FINDS THE GLOBAL NODAL DISPLACEMENTS BY SOLVING P=KQ
C THAT Q(ND) RETURNS IN P(ND)).
C USING A MODIFIED (BANDED-SYMMETRIC) GAUSS ELIMINATION TECHNIQUE (NOTE
C=======================================
      IMPLICIT REAL*8 (A-H,L,O-Z)
      COMMON /C1/NE,NN,ND,NZ,NEM,NNM,ID,IBW
      COMMON /C3/INC(8,5),ICON(81)
      COMMON /C5/P(81),GK(81,36)
      IBWM=IBW-1
      NDM=ND-1
      IPVTCK=1.0D-01
      IFTRCK=1.0D-14
C
C TRIANGULARIZE GK(ND,IBW) AND MODIFY P(ND):
      DO 50 I=1,NDM
C CHECK FOR CONSTRAINED ROW AND COLUMN (I.E. ENTIRE COLUMN ALLREADY ELIMINATED):
      IF(ICON(I).EQ.1) GO TO 50
      PIVOT=GK(I,1)
C CHECK FOR NEARLY ZERO PIVOT:
      IF(DABS(PIVOT).GT.IPVTCK) GO TO 20
      WRITE(6,10) I
   10 FORMAT(//1X,'*****TILT *****PIVOT NUMBER ',I3,' NEARLY ZERO*****')
      GO TO 200
   20 CONTINUE
C BEGIN ROW TRANSFORMATION (COLUMN ELIMINATION):
      JLIM=IBWM
C CHECK TO DETERMINE IF IBWM ROWS (COLUMNS) MAY BE OPERATED ON:
      ITEST=ND-I
      IF(ITEST.LT.IBWM) JLIM=ITEST
      DO 40 J=1,JLIM
```

```
      IVAL=JLIM-J+2
      FACTOR=GK(I,IVAL)
C CHECK FOR EXTREMELY NEARLY ZERO ROW (COLUMN) "FACTOR" ELEMENT:
      IF(DABS(FACTOR).LT.IFTRCK) GO TO 40
      CON=FACTOR/PIVOT
      IROW=I+IVAL-1
      P(IROW)=P(IROW)-CON*P(I)
      DO 30 K=1,J
      ICOL=IVAL+K-1
   30 GK(IROW,K)=GK(IROW,K)-CON*GK(I,ICOL)
   40 CONTINUE
   50 CONTINUE
C TEMPORARY DE-BUGGING PRINTOUTS:
      WRITE(6,100)(P(I),I=1,ND)
  100 FORMAT(/(1X,9(F14.2)))
C
C BACK-SUBSTITUTE AND SOLVE FOR GLOBAL DISPLACEMENTS, Q(ND):
C CALCULATE Q("ND"):
      P(ND)=P(ND)/GK(ND,1)
C CALCULATE REST OF Q(ND):
      DO 80 I=1,NDM
      IROW=ND-I
      SUM=0.0000
      JLIM=IBWM
C CHECK TO DETERMINE IF LESS THAN IBWM TERMS NEED BE SUMMED:
      IF(I.LT.IBWM) JLIM=I
C SUM THE PREVIOUSLY FOUND DISPLACEMENTS MULTIPLIED BY THEIR FACTORS:
      DO 60 J=1,JLIM
      IROS=J+IROW
      ICOS=J+1
   60 SUM=SUM+GK(IROW,ICOS)*P(IROS)
C CALCULATE Q(IROW):
```

```
 80 P(IROW)=(P(IROW)-SUM)/GK(IROW,1)
    WRITE(6,90)
 90 FORMAT(/1X,'GLOBAL NODAL DISPLACEMENTS:'/6X,'W1',12X,'WX1',10X,'-W
   +Y1',11X,'W2',12X,'WX2',10X,'-WY2',11X,'W3',12X,'WX3',10X,'-WY3')
    WRITE(6,95) (P(I),I=1,ND)
 95 FORMAT((1X,9(F14.10)))
200 CONTINUE
    RETURN
    END
```

The vita has been removed from the scanned document

A DOUBLY-CURVED FINITE ELEMENT ANALYSIS OF THIN ARBITRARY SHELL STRUCTURES

by

John Billings Burchnall

(ABSTRACT)

This paper is concerned with the linear elastic static analysis of thin arbitrary shell structures by the use of the displacement approach finite element method. The various factors involved in selecting a shell finite element are discussed.

A comprehensive formulation of a 27 degree of freedom, arbitrary, doubly-curved, nonconforming, triangular, shallow shell element is presented. Both the normal and tangential displacement fields are expressed by "incomplete" cubic, natural coordinate, polynomial interpolation functions.

A WATFIV/FORTRAN computer code utilizing this element in a linear elastic static analysis of thin shell structures is formulated and presented. Demonstration problems are presented and comparisons are made with solutions in the literature.