THE UNIVERSITY OF MANITOBA

A NEW RECTANGULAR PLATE-BENDING FINITE ELEMENT FOR DETERMINING THE DYNAMIC CHARACTERISTICS OF COMPLEX, THREE DIMENSIONAL STRUCTURES

by

D. H. MC DONALD

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

DEPARTMENT OF MECHANICAL ENGINEERING

WINNIPEG, MANITOBA

May, 1971

THE UNIVERSITY OF MANITOBA ELIZABETH DAFOE LIBRARY

© D.H. McDonald 1972

ACKNOWLEDGEMENT

The author would like to express his appreciation to Dr. N. Popplewell for his helpful and inspiring supervision, and Mr. R. King for experimental data of the box.

Further, the author is indebted to National Research Council for its financial support and for loan of their computer program.

ABSTRACT

A fully conforming, displacement type finite element of rectangular shape is developed and applied to the solution of rectangular plates in bending. The transverse deflection is represented by a polynomial expression which includes terms up to degree eight. Six deflection parameters, the displacement and its first and second derivatives, are specified at each corner - giving a total of 24 degrees of freedom for the element. This element is shown to be generally more accurate for dynamic problems than previously available elements.

The assumed deflection polynomial is such that the rectangular element conforms with a triangular element. Then a combination of rectangular and triangular elements is capable of describing arbitrarily shaped boundary configurations. The accuracy of various combinations of the two types of elements is tested by analysing the free vibrations of plates with a rectangular and simple non-rectangular geometry.

Subsequently, a combination of these elements is used to determine the dynamic characteristics of a box with a sloping roof. The accuracy of the theoretical natural frequencies and mode shapes is checked by comparing them with experimental values.

CONTENTS

NOTATION

Structural Properties

ρ	mass/area
ℓ	some standard length
g	acceleration due to gravity
E	Young's modulus
ν	Poisson' ratio
t	thickness
I	cross sectional moment of inertia
D	$\frac{Et^3}{12(1-\nu^2)}$ flexural rigidity
w	transverse displacement
$\underline{M}$	overall mass matrix
$\underline{M}_e$	element mass matrix
$\underline{M}_c$	constrained overall mass matrix
$\underline{K}$	overall stiffness matrix
$\underline{K}_e$	element stiffness matrix
$\underline{K}_c$	constrained overall stiffness matrix

General

$\underline{T}$	transformation matrix
$\underline{R}$	rotation matrix
λ	wavelength
X,Y,Z	global co-ordinate system
$\left.\begin{matrix} x,y,z \\ \xi,\eta,\psi \end{matrix}\right\}$	local co-ordinate systems

Subscripts

c	constrained
e	element
x,y	partial derivative with respect to x and y respectively

Symbols

{ }	vector matrix
$\_$	square matrix

CHAPTER 1

Introduction

The prospect of supersonic passenger transport services has resulted in concern with the effects of sonic boom on people and buildings. (A sonic boom is the pressure disturbance observed on the ground when an aircraft flies overhead at a speed greater than that of sound). Information regarding these effects would be invaluable in the prevention of damage to buildings and the reduction of annoyance to people within the buildings. To gain such knowledge from full scale tests is expensive and causes annoyance to the general public. In an attempt to utilize other methods, Popplewell (1) successfully employed two simulation techniques - one experimental and the other mathematical - to determine the response of a flat-roofed structure to a sonic boom. The mathematical approach, however, is much more flexible as structural changes, which might be incorporated at the design stage, can be accomodated easily.

In the interest of extending the mathematical method to more familiar structural geometries, the dynamic characteristics of a box-type structure with a sloping roof are investigated. The theoretical frequencies and mode shapes are evaluated and compared to experimental results.

The experimental model was constructed of Perspex and dynamic characteristics determined by using harmonic excitation.

The mathematical model employed the finite element displacement method. In the application of this method, it is assumed that the structure is composed of a grid of interconnected elements with known mass and stiffness characteristics. Using similar assumptions as reference 1, the free vibrations of a three dimensional structure may be reduced, under certain boundary constraints, to a complex shaped, flat plate undergoing flexual vibrations.

Triangular, unlike rectangular finite elements, can realistically approximate such complicated boundary geometries but they suffer from the disadvantage that their orientation can have a significant effect on accuracy. Cowper et al (10) compare previously available rectangular and triangular plate-bending elements and conclude that their conforming triangular element NRCD is superior. (Conformity in this context means that transverse displacements and slopes are continuous between elements). However, a conforming rectangular element, UM6, is developed in this thesis and shown to be better than NRCD for dynamic problems. UM6 has the additional advantage that it conforms with NRCD so that a combination of these two elements is capable of describing irregularly shaped boundaries, but the combination is only useful if it is more accurate than using triangular elements alone. This is demonstrated to be the case for the particular examples of a rectangular and a cantilevered triangular plate in bending. Then the combination is shown to give satisfactory agreement with experimental data for the free vibrations of a box with a sloping roof.

CHAPTER 2

2.0 Introduction

Over its long period of development, the finite element displacement method has proved to be a powerful technique for the dynamic and static analysis of structures. In this method a structure is represented by an assembly of elements each having an appropriate mass and stiffness. Much recent research has been undertaken to generate element properties more accurately with increasingly sophisticated forms of the assumed displacement distribution within an element. To assess the relative accuracy of different approximations, a rectangular simply-supported plate is often considered,as exact solutions are well known.

A survey of various approximations for triangular and rectangular plate-bending elements was presented by Cowper et al (10). It was suggested that conformity is a desirable property of these elements providing fields of strain are represented accurately. Conformity in this context means that only transverse displacements and slopes are continuous between elements. Bogner et al (16), Butlin and Leckie (18) and later Mason (3) go one step further by imposing total compatibility between rectangular elements. Compatibility implies that the unknown displacements and derivatives are not only continuous at nodal points but also along common edges between elements. Hence, the deflection, its first derivatives and twist are continuous for a 16 degree-of-freedom rectangular element and, additionally, curvatures are continuous for 24 degrees of freedom. Subsequently, Cowper et al (10) developed a different displacement function for a conforming triangular element (NRCD) which proved superior to elements previously available.

However, a conforming rectangular element, UM6, is formulated in this paper and shown to be better than NRCD for dynamic problems. UM6 has the additional advantage that it conforms with NRCD so that a combination of these two elements can be used to describe complex boundary shapes. A study of various ratios of the number of rectangular to triangular elements is presented for the particular examples of a simply-supported rectangular and a cantilevered triangular plate in bending.

2.1 A Conforming Combination of Rectangular and Triangular Elements

The form of the displacement approximation for UM6 is based on the algebraic polynomial used for NRCD. This polynomial has a quintic and cubic variation of deflection and normal slope respectively along any edge in the edgewise co-ordinate. To ensure continuity of displacements and slopes between elements having a common edge (conformity), the coefficients must be determined uniquely by quantities at the terminal nodal points. These requirements are satisfied by using six deflection parameters at each nodal point. For UM6 to conform with NRCD, it must have the same variations along edges and identical parameters at coincident nodal points. These parameters are the transverse deflection and its first and second derivatives $(w, w_x, w_y, w_{xx}, w_{xy}, w_{yy})$ - a total of 18 and 24 degrees of freedom for the triangle and rectangle respectively. Then the six coefficients of the quintic polynomial are fixed uniquely by the deflection, edgewise slope and edgewise curvature at each of two terminal nodal points. Similarly, the cubic is defined uniquely by the normal slope and the twist at each of the terminal points.

Using the local co-ordinate axis shown in Figure 1(a), the displacement approximation satisfying the deflection and normal slope conditions along any edge of the rectangle can be written as

$$\begin{aligned} w(x,y) = {} & a_1 + a_2x + a_3y + a_4x^2 + a_5xy + a_6y^2 + a_7x^3 + \\ & + a_8x^2y + a_9xy^2 + a_{10}y^3 + a_{11}x^4 + a_{12}x^3y + a_{13}x^2y^2 + \\ & + a_{14}xy^3 + a_{15}y^4 + a_{16}x^5 + a_{17}x^3y^2 + a_{18}x^2y^3 + a_{19}y^5 + \\ & + a_{20}x^4 \left(\tfrac{1}{2}\, yb - \tfrac{2}{3}\, y^3/b\right) + a_{21}\left(\tfrac{1}{2}\, xa - \tfrac{2}{3}\, x^3/a\right)y^4 + \\ & + a_{22}x^3y^3 + a_{23}x^5 \left(\tfrac{1}{2}\, yb - \tfrac{2}{3}\, y^3/b\right) + a_{24}\left(\tfrac{1}{2}\, xa - \tfrac{2}{3}\, x^3/a\right)y^5. \qquad (1) \end{aligned}$$

Terms up to order eight are present within the rectangular element but quintic deflection variations exist along edges. Bracketed terms in the above expression are required to satisfy the cubic normal slope conditions.

The 24 generalised displacements for rectangular element UM6 are assembled into a column vector $\{\underset{\sim}{w}_1\}$ whose transpose is

$$\{\underset{\sim}{w}_1\}^T = (w_1, w_{x1}, w_{y1}, w_{xx1}, w_{xy1}, w_{yy1}, w_2 \ldots, w_3 \ldots, w_4 \ldots) \qquad (2)$$

where subscripts 1, 2, 3 and 4 refer to the numbered nodal points in Figure 1(a). Using equation (1), each term in $\{\underset{\sim}{w}_1\}$ can be expressed in terms of the a_i to give

$$\{\underset{\sim}{w}_1\} = \underset{\sim}{T}\, \{\underset{\sim}{A}\} \; . \qquad (3)$$

$\underset{\sim}{T}$ is a 24 x 24 matrix and $\{\underset{\sim}{A}\}$ is a column vector of the twenty four polynomial coefficients $a_1, a_2, a_3, \ldots a_{24}$. The determinant of $\underset{\sim}{T}$ has the value $-3.160494\ (ab)^{30}$; it is only zero when the area of the rectangular element vanishes. Hence, $\underset{\sim}{T}$ is not singular in practical situations so that from equation (3),

$$\{\underline{A}\} = \underline{T}^{-1} \{\underline{w}_1\}. \qquad (4)$$

In deriving an identical relationship to (4) for triangular element NRCD, three unknowns have to be eliminated from the general quintic polynomial assumed for the deformation. Again it is achieved by stipulating that the normal slope has a cubic variation along element boundaries. Thus the term x^4y is omitted to satisfy the slope condition along edge $y = 0$ - resulting in the twenty term polynomial expression shown in Figure 1(b). Two further coefficients have to be eliminated as the triangle has only eighteen generalized displacements corresponding to six displacements at each of its three nodal points. The slope condition along edges P_1P_3 and P_2P_3 in Figure 1(b) produces two equations to determine the extraneous coefficients in terms of the remainder. Then Cowper et al are able to invert a 20 x 20 $\underline{T}$ matrix and reduce the size of the inverse by omitting the last two columns. This results in an inverse matrix which is consistent with the twenty coefficients and eighteen generalized displacements contained in vectors corresponding to $\{\underline{A}\}$ and $\{\underline{w}_1\}$ respectively.

In the following section, mass and stiffness matrices are derived for the rectangular element in an identical manner used by Cowper et al for their element. Differences in detail occur because of the dissimilarity of shape between the triangle and rectangle. Consequently, integrals over element areas require different limits and a more direct approach can be used to impose normal slope conditions on the rectangle's displacement approximation. Closed-form expressions are obtained in both cases which allow the entries of the matrices to be generated automatically within the computer. Then a system's equations of motion are derived in such a way that the total energy is stationary.

2.2 Generation of Element Mass and Stiffness Matrices

The stiffness and mass matrices of elements UM6 and NRCD are obtained by considering the strain and kinetic energy of a uniform isotropic plate in bending. Classical theory gives

$$U_e = \frac{1}{2} D \int_{A_e} \{w_{xx}^2 + w_{yy}^2 + 2\nu w_{xx} w_{yy} + 2(1 - \nu) w_{xy}^2\} \, dA_e \tag{5}$$

as the strain energy of an element. D is the flexural rigidity, ν is Poisson's ratio and A_e is the area of the element. The only difference in the following derivation and that of reference (10) is caused by areas A_e of the triangular and rectangular elements being different. Due to the discontinuity of the variation of y at vertex P_3 in Figure 1(b), the integration for NRCD is taken as the sum of the integrals over triangles P_1QP_3 and QP_2P_3. Typical limits are then $0 \leq x \leq a$ $0 \leq y \leq c(1 - x/a)$ for QP_2P_3. Limits for UM6 in Figure 1(a) on the other hand can be written simply as $-a/2 \leq x \leq a/2$ and $-b/2 \leq y \leq b/2$.

Reference (10) shows that if the approximate deformation over the triangular element is written in the form

$$w(x,y) = \sum_i a_i x^{m_i} y^{n_i} \tag{6}$$

then a typical term $\int_{A_e} w_{xx}^2 \, dA_e$ in the strain energy equation (5) may be written as

$$\int_{A_e} w_{xx}^2 \, dA_e = \sum_{ij}\sum a_i a_j \{m_i m_j (m_i - 1)(m_j - 1) F(m_i + m_j - 4, n_i + n_j)\} \tag{7}$$

where

$$F(m,n) = c^{n+1} \{a^{m+1} - (-b)^{m+1}\} \, m!n!/(m + n + 2)! \, . \tag{8}$$

Following the same procedure, equations (6) and (7) remain valid for UM6 and equation (8) takes the form

$$F(m,n) = \begin{cases} 0 & \text{either } m \text{ or } n \text{ even} \\ a^{m+1}b^{n+1}/(2^{m+n})mn & m \text{ and } n \text{ both odd .} \end{cases} \tag{9}$$

Other terms in equation (5) are evaluated in a similar manner and then combined to give U_e in the quadratic form

$$U_e = \frac{1}{2} D \, \{\underline{A}\}^T \; \underline{k} \; \{\underline{A}\}. \tag{10}$$

A typical component k_{ij} of matrix $\underline{k}$ is given by

$$\begin{aligned} k_{ij} = {} & m_i m_j (m_i - 1)(m_j - 1) \, F(m_i + m_j - 4, n_i + n_j) + \\ & n_i n_j (n_i - 1)(n_j - 1) \, F(m_i + m_j, n_i + n_j - 4) + \\ & \{2(1 - \nu) m_i m_j n_i n_j + m_i n_j (m_i - 1)(n_j - 1) + \\ & \nu m_j n_i (m_j - 1)(n_i - 1)\} \, F(m_i + m_j - 2, n_i + n_j - 2) \end{aligned} \tag{11}$$

for both NRCD and UM6. Combining equations (4) and (10) produces

$$U_e = \frac{1}{2} D \, \{\underline{w}_1\}^T \; \underline{K}_1 \; \{\underline{w}_1\} \tag{12}$$

where

$$\underline{K}_1 = (\underline{T}^{-1})^T \, \underline{k} \, \underline{T}^{-1}. \tag{13}$$

$\underline{K}_1$ has to be transformed now form local (x,y) to global co-ordinates (X,Y). Employing the well known relations

$$\begin{aligned} w_x &= w_X \cos\theta + w_Y \sin\theta \\ w_{xx} &= w_{XX} \cos^2\theta + w_{XY} \, 2 \sin\theta \cos\theta + w_{YY} \sin^2\theta \\ &\text{etc.} \end{aligned} \tag{14}$$

where θ is the angle between the x and X axes, gives

$$\{\underline{w}_1\} = \underline{R}\ \{\underline{w}_e\}. \tag{15}$$

$\underline{R}$ is the rotation matrix containing trigonometric functions of θ. It depends only upon an element's orientation with respect to the global axes. Using equations (12), (13) and (15), U_e is given by

$$U_e = \frac{1}{2} D\ \{\underline{w}_e\}^T\ \underline{K}_e\ \{\underline{w}_e\} \tag{16}$$

where

$$\underline{K}_e = \underline{R}^T\ (\underline{T}^{-1})^T \underline{k} (\underline{T}^{-1})\ \underline{R} \tag{17}$$

is the form of the element stiffness matrix in the global co-ordinate system for both NRCD and UM6.

The mass matrices for both elements are derived in the same way as the stiffness matrices. Consider the kinetic energy of an element

$$T_e = \frac{1}{2}\ \omega^2 \rho t \int_{A_e} w^2\ dA_e \tag{18}$$

where ρ and t are the density and thickness of the plate respectively and ω is the circular frequency for sinusoidally time-dependent vibrations. It can be shown that mass matrix $\underline{M}_e$ corresponding to $\underline{K}_e$ takes the form

$$\underline{M}_e = \underline{R}^T\ (\underline{T}^{-1})^T\ \underline{m}(\underline{T}^{-1})\underline{R}. \tag{19}$$

The components of $\underline{m}$ are determined from

$$m_{ij} = F(m_i + m_j,\ n_i + n_j) \tag{20}$$

for both NRCD and UM6 and m_i and n_i are the exponents in equation (6).

The total energy E_n for the free transverse motion of a uniform plate is obtained by summing the strain and kinetic energies over all elements. Applying compatibility conditions at each of the nodal points gives

$$E_n = \{\underline{w}\}^T[\frac{D}{2}\underline{K} - \frac{\rho t\omega^2}{2}\underline{M}]\{\underline{w}\}. \tag{21}$$

$\underline{K}$ and $\underline{M}$ are the overall stiffness and mass matrices, respectively, for the plate and $\{\underline{w}\}$ is the displacement vector which contains all the possible nodal displacements. Equation (21) is valid for any combination of rectangular and triangular elements providing care is exercised in applying compatibility conditions.

Some of the nodal displacements must be constrained to zero in order to include the plate's edge conditions. Only displacement constraints can be applied directly to equation (21) which leads to some disagreement in published literature (3) whether displacements corresponding to force boundary conditions should be employed. To facilitate the comparison with the bulk of previously published work, force as well as kinematic boundary conditions are used in this thesis for simply-supported and fully fixed edges but not for free edges. Thus, for example, the normal curvature as well as displacement and edgewise slope and curvature are constrained to be zero at nodal points on a simply-supported edge. Details of other edge conditions are well known and need no further comment. Then quadratic energy matrix (21) can be partitioned and reduced in size to produce

$$E_n = \{\underline{w}_c\}^T\ [\frac{D}{2}\underline{K}_c - \frac{\rho t\omega^2}{2}\underline{M}_c]\ \{\underline{w}_c\} \tag{22}$$

where $\{\underline{w}_c\}$ is the vector containing only the non-zero nodal displacements. $\underline{K}_c$ and $\underline{M}_c$ are the constrained stiffness and mass matrices, respectively, for the plate.

Having constrained the matrices, the energy is made stationary by differentiating equation (22) with respect to each of the nodal displacements in turn and equating it to zero. This procedure gives the required equations of motion as

$$[D\,\underline{K}_c - \rho t\omega^2\,\underline{M}_c]\,\{\underline{w}_c\} = \{\underline{0}\}. \tag{23}$$

For non-singular $\underline{K}_c$, the above equation may be reduced to the standard eigenvalue problem

$$[\underline{S}_c - \lambda \underline{I}]\,\{\underline{w}_c\} = \{\underline{0}\} \tag{24}$$

where

$$\underline{S}_c = \underline{K}_c^{-1}\,\underline{M}_c \quad \text{and} \quad \lambda = D/\rho t\omega^2. \tag{25}$$

Using double precision arithmetic throughout, eigenvalue problem (24) is solved using the QR method - details of which are given by Wilkinson (16) . The solutions give the required natural frequencies and the displacements of the constrained system.

Mode shapes of the original system are obtained by reinserting the zero nodal displacement constraints.

2.3 Discussion of Results

The natural frequencies and normal modes of a plate with side lengths in the ratio 40 to 27 were determined using various assemblages of UM6, Cowper et al's triangular element NRCD and Mason's rectangular elements. Figure 2 shows the arrangements for the two support conditions considered; all edges were either simply-supported or fully fixed. The complete plate was idealized with N identically dimensioned elements. Advantage was taken of symmetry about the plate's non-diagonal axes so that only a quarter of the plate need be considered with symmetry constraints imposed on appropriate boundaries. This simplification reduces the eigenvalue problem (24) to about a quarter of the size of the original problem with a corresponding reduction in computer storage requirements.

By considering two arrangements (shown in Figure 2 as 'P' and 'Q') Cowper et al showed that the orientation of NRCD has a significant effect on accuracy for static problems. This point is investigated for dynamic cases using the same arrangements.

Natural frequencies computed with the different elements are compared initially in Figure 3 for the simply supported-plate, since the exact solution for this boundary condition is well known. The percentage error in this figure is defined as

$$\left(\frac{\text{approximate value - exact value}}{\text{exact value}}\right) \times 100\% \; . \qquad (26)$$

It is seen that UM6 generally gives the most accurate natural frequencies for a given number of elements. Exceptions occur when nine or fewer elements are used to determine the higher frequencies. Although Mason's 24 degree-of-freedom element is most accurate in these cases, UM6 always produces least error in the first two natural frequencies. Figure 3 also shows that the orientation of NRCD can be significant for dynamic problems when only eight elements idealise the plate. As for the static case, Q is superior to the P pattern. There is very little difference between the two arrangements for a larger number of elements but, surprisingly, the P is then consistently better than the Q pattern.

The computed natural frequencies of UM6 converge towards their theoretical values far more quickly with an increasing number of elements than corresponding frequencies calculated with the other elements. Mason's element, although relatively accurate for a small number of elements, has the worst rate of convergence.

A comparison on the basis of number of elements alone is unfair to the triangular elements as two triangles can constitute one rectangular element with both systems having the same degrees of freedom. Hence it may be argued that a better basis for comparison is the total constrained degrees of freedom of the complete structure, as this number gives a measure of computational effort and storage requirements. Here constrained degrees of freedom is the difference between the total degrees of freedom of the structure and the number of boundary constraints. Using constrained degrees of freedom, the only variation from previous observations is that the difference in the characteristic convergence rates produced by UM6 and NRCD is not so marked although UM6's is still the best.

Also given in Figure 3(a)are the limits on the errors of the first ten natural frequencies obtained for a given number of constrained degrees of freedom with Mason's 16 degree-of-freedom rectangular element. It is seen that a larger number of the 16 degree-of-freedom element is superior to the use of the element with 24 generalized displacements - a fact noted by the various originators of these elements. However a similar comparison shows it is preferable to use UM6.

The precise accuracy of a calculated mode shape is difficult to assess and to present. The accuracy of a natural frequency, however, gives some indication of that of the associated mode shape. For simply-supported edge conditions, the mode shapes should be double sine (or cosine) waves. Using at least 82 constrained degrees of freedom, the computed nodal displacements, slopes and second derivatives agree with theoretical values to at least six, three and two significant figures, respectively, for UM6. Corresponding accuracies are reduced generally for the other two elements to four, three and one for NRCD and six, one and one significant figure for Mason's element. Thus, for high precision, it is important to determine accurately high order derivatives (particularly the slopes) as well as the displacements.

Subsequently, free vibrations were calculated for the same plate but with fully fixed edges. Since exact solutions for these edge conditions are not known, computed results were compared with those given by Warburton (9) and with those of Claasen and Thorne (8). These comparisons are made in Figure 4 and Table 4 where the percentage variation is calculated using equation (26), taking respectively Warburton's and Claasen and Thorne's results as 'exact values'. The reference results are themselves approximate and overestimate the values of the natural frequencies - explaining the reason for the negative 'errors'. As for the simply-supported case, UM6 and NRCD produce natural frequencies which are generally lower than those of Mason for the largest number of constrained degrees of freedom considered. All are less than Warburton's values and, apart from the sixth mode, UM6's are slightly less than those of Claasen and Thorne. For fully fixed, unlike simply supported edges, Figure 4 indicates that UM6 has the smallest variation from the reference values of most of the higher as well as the lower frequencies for few constrained degrees of freedom. The percentage variations not shown in these cases are too large to be plotted in Figure 4.

Exact mode shapes are not known for the fully fixed plate. However, the fundamental mode shapes produced by all the elements for this condition have slightly different normalized displacements along center lines parallel to plate edges. This was first noted by Mason who attributed the phenomenon to the fact that a truly separable solution cannot be obtained for a rectangular plate with all edges fully fixed.

Only those boundary shapes which can be represented exactly with either rectangular or triangular elements have been considered in the preceding discussion. There are a great variety of non-rectangular shapes, however, which can be approximated realistically with triangular but not rectangular elements alone. Although rectangular element UM6 is unable, by itself, to describe complex geometries, it is more accurate than triangular element NRCD. Nevertheless, UM6 is conformable with NRCD so that is would seem desirable to use some combination of these elements when describing irregular boundaries.

To obtain a 'feeling' for an optimum combination, the free vibrations of the simply-supported rectangular plate and a simple example of a complex boundary shape - a right triangular plate having one edge fully fixed and the other two edges free - are determined using various combinations of the two elements. Material properties of the triangular plate are

Modulus of elasticity	$E = 30 \times 10^6$ lb/sec^2
Poisson's ratio	$\nu = 0.3$
Density	$\rho = .0007381737$ slug/in^3

Thickness	t = 0.061 in
Span (distance of free vertex from the fully fixed edge)	10 in.
Length of fully fixed edge	10 in.

Table 1 gives experimental values in Hertz of the triangular plate together with the first ten natural frequencies computed with an increasing number of rectangular to triangular elements. Discrepancies between the frequencies calculated with triangular elements alone and those given in reference (10) are due to the gravitational constant g being assigned the more usual value of 386.088 in/sec^2 rather than 387.75 in/sec^2 assumed by Cowper et al. The table shows that as the ratio of the number of rectangular to triangular elements increases, the computed frequencies decrease and tend to experimental values given in reference (21) However, the difference between extreme computed values for a given frequency is marginal for the six lowest modes but becomes progressively more significant for higher frequencies. Further, there is little difference in corresponding mode shapes obtained using the various assemblages. All give good agreement with experimental results.

Then three different combinations of UM6 and NRCD in which the triangular elements are arranged in a "P" sense along the diagonal of the quarter section are used to idealize the 40:27, simply-supported rectangular plate. The associated percentage error of the ten lowest natural frequencies are listed in Table 3 together with the errors produced by the idealizations composed of triangles and rectangles alone. For the three different constrained degrees of freedom, the accuracy of the combination is always bounded by those of the triangles or rectangles alone.

For 18 constrained degrees of freedom, the combination usually gives significantly more accurate frequencies than NRCD. This is not only true for the higher modes but is also the case for the lowest modes. The effect is much less apparent,however, for the lowest modes with idealisations using large degrees of freedom. Hence, a combination of rectangular and triangular elements should be used when few elements or higher natural frequencies are required. It would seem advantageous in these cases to use a minimum number of triangular elements.

2.4 Conclusions

A 24 degree of freedom rectangular plate-bending element has been formulated which proves to be generally more accurate than previous elements for dynamic problems. Hence it is advantageous to use this element alone if a system's shape can be approximated with rectangles. However, there is a great variety of shapes for which rectangular elements are unsuitable. The newly developed element conforms with Cowper et al's triangular element so that a combination of the two elements can realistically describe irregular boundaries. The greatest benefit of using the combination in preference to the triangular element alone occurs when few elements or higher natural frequencies are required. A minimum number of triangular elements would seem advisable in these cases.

CHAPTER 3

3.0 Introduction

The development of supersonic commerical aircraft has led to numerous investigations related to the effects of sonic boom on structures (19),(20). Full scale tests were found to be expensive and caused a great deal of inconvenience to the general public. Therefore, it has been found advantageous to use comparatively cheap simulation techniques which localize the annoyance. In reference (1), Popplewell investigated the dynamic behaviour of a flat-roofed box which was subjected to a simulated sonic boom. As well as full scale tests, two simulation techniques - a small scale Perspex model and a mathematical model - were used. It was concluded that the mathematical model provided valid results and could be used to give useful data pertaining to structural response to sonic booms.

The finite element displacement method was used as a basis of the mathematical model. This technique, unlike the series solution of Dickinson and Warburton (7), was sufficiently flexible to account for discontinuities like windows in the structure.

To simplify the analysis, Popplewell reduced the three dimensional box to a cross shaped flat plate configuration.

The application of a flat plate analysis to more realistic building configurations required the description of non-rectangular boundaries. The development of UM6 allowed a conformal combination of UM6 and NRCD to accurately represent arbitrary structural shapes. As a consequence, the flat plate analysis could be extended to accomodate more familiar configurations like a house with a sloping roof. This chapter deals with the inclusion of a sloping roof on the box-type structure. The reduction of this structure to a flat plate, required triangular elements to describe the non-rectangular boundary caused by the sloping roof. From the previous chapter it was observed that a finite element respresentation composed of UM6 and a minimum number of NRCD could describe most accurately the flat plate configuration.

Due to symmetry, it was possible to consider only one quarter of the structure. The natural frequencies and mode shapes were evaluated using the finite element idealization of Figure 6 and a comparison was made with experimental values obtained for a small scale Perspex model.

The effect of the sloping roof was discussed relative to the results for the flat-roofed structure.

3.1 The Mathematical Model

The structure was assumed to be built from plates, each of which was perfectly elastic, homogeneous and isotropic. The transverse displacement of each plate was considered small compared with the wavelength of flexual vibrations. In-plane displacements were neglected and acoustical coupling between individual faces was assumed to be negligible. The base edges of the box were fixed while remaining edges were allowed to rotate such that joints remained at fixed angles but did not move in translation. Corner deflections, therefore, were disallowed. If the corners also were assumed not to twist, the sloping roofed box could be represented as a flat plate. A finite element idealization of the plate employed the arrangement shown in Figure 6 . By constraining the displacements of common edges to be the same, differences between the three dimensional and two dimensional analysis were removed.

3.2 Physical Properties, Natural Frequencies and Normal Modes of the Model Box.

To provide a means of determining the validity of the mathematical idealization, the natural frequencies and mode shapes of a small plastic model were determined experimentally. (21)

A sloping roofed model with the dimensions shown in Figure 6 was constructed of Perspex and placed in a wooden jig to simulate clamped base edges. Like all plastics, Perspex suffered from the disadvantages that its physical properties vary with temperature, humidity, thickness and frequency (1). Strict control of temperature and humidity was impossible so that their variation was minimized as far as possible by using the same laboratory.

To determine the effect of frequency on the value of dynamic Young's modulus, the natural frequencies of a cantilevered Perspex beam having the same thickness as the model were evaluated experimentally Figure 7. Details of the standard experimental procedure adopted can be found in reference (21). Assuming negligible damping, the natural frequencies of a cantilevered uniform beam with cross sectional area, A, and length, ℓ, are given in Bishop and Johnson (17) as

$$\omega_p^2 = \frac{EI\ g(\lambda_p \ell)^4}{\rho A \ell^4} \tag{27}$$

for the pth mode. Hence

$$E = \frac{4\pi^2 f^2{}_p}{Ig} \left[\frac{1}{(\lambda p \ell)^4}\right] \rho A \ell^4. \tag{28}$$

The beam functions, $\lambda_p \ell$, are tabulated in Bishop and Johnson as

MODE	$(\lambda \ell)^4$
1	12.3624
2	485.519
3	3806.55
4	14617.3

Substituting the beam's natural frequencies into equation (28) gave the dynamic Young's modulus.

However, it was found that the Perspex was extremely sensitive to humidity. Consequently values of Young's modulus were shown in Table 5 for two different relative humidity levels observed during resonance tests of the model beam.

Table 6 shows the theoretical natural frequencies of the sloping roofed structure obtained from the finite element idealization shown in Figure 6 . The theoretical frequencies agreed reasonably with the experimental frequencies. The theoretical results, however, overestimated the experimental values. This could be contributed in some part to the relative himidity during the model tests being 91% while the Young's modulus values used in the theoretical analysis were calculated for a relative humidity of 79%. Table 5 indicates that the effect of decreasing humidity was to increase the natural frequencies (and Young's modulus) of Perspex. The first six theoretical mode shapes of the sloping roofed structure are given in Figure 8 - experimental results were unavailable at the time.

Im comparing the theoretical frequencies of the flat roofed structure (2) to those of the sloping roofed structure, the effect of the sloping roof was to increase the overall stiffness of the structure. This was due mainly to the constraining effect of the sloping roof's apex.

3.3 Conclusions

The finite element analysis of a box-type structure has been extended to more complicated geometries by using a combination of rectangular and triangular elements. Reasonable agreement has been demonstrated between the theoretical and experimental natural frequencies of a box with a sloping roof. This gives increasing confidence in the future use of the theoretical method in determining the response of complicated structures to sonic booms.

It has been found that the main effect of the sloping roof is to increase the structure's overall stiffness - and, hence, its natural frequencies.

CHAPTER 4

Conclusions

This chapter reviews briefly the conclusions presented in previous chapters.

A new rectangular plate bending element, UM6, has been developed and shown to be superior to previously available elements. The element has the additional advantage that it conforms with an accurate triangular element so that a combination of the two elements can realistically approximate irrigularly-shaped boundaries. For greatest accuracy,the optimum combination of the two elements would seem to be the one which uses a minimum number of triangular elements.

An optimum combination of rectangular and triangular elements is applied to the determination of the freely vibrating characteristics of a box-type structure with a sloping roof. The resulting theoretical results compare favourably with experimental values obtained by exciting a Perspex model harmonically. This validates the assumptions used to simplify the mathematical model. It is shown that the effect of the sloping roof is to increase the model's overall stiffness - and, hence, its natural frequencies.

The reasonable agreement between theoretical and experimental models gives added confidence in using the theoretical model to determine the response of complicated structures to sonic booms. This work will be undertaken in the near future.

REFERENCES

1. N. Popplewell — 1969, Ph.D. thesis presented to the University of Southampton.

2. N. Popplewell — 1971. J. Sound. Vib. vol. 14, no. 3, The vibration of a box-type structure 1. Natural Frequencies and Normal Modes.

3. V. Mason — 1968. Ph.D. thesis presented to the University of Southampton.

4. V. Mason — 1968. J. Sound Vib. vol. 7, no. 3, Rectangular finite elements for analysis of plate vibrations.

5. P.A. Macdonald — 1969. Dept. of Elect. Engr. Publ. , NAG-004, Calculating the inverse of a matrix by Gauss-Jordon Method.

6. J.H. Wilkinson — 1965. The Algebraic Eigenvalue Problem Oxford: Clarendon Press.

7. S.M. Dickinson, G.B. Warburton — 1967. J. Mech. Eng. Science, vol. 9, no. 4, Vibration of box-type structures.

8. R.W. Claasen, C.J. Thorne — 1960. NAV WEPS Rpt. 7016, U.S. Naval Ordnance Test Station, Transverse vibrations of thin rectangular isotropic plates.

9. G.B. Warburton — 1954. Proc. Inst. Mech. Eng.,168, The vibration of rectangular plates.

10. G. Cowper, F. Kosko, G. Lindberg, M. Olsen — 1968. NRC NAE Aero. Report LR-514, National Research Council of Canada, A high precision triangular plate-bending element.

11. I. Holland, K. Bell — 1969. Finite element methods in stress analysis, Trondheim: Tapir.

12. G.A. Butlin — 1966. Ph.D. thesis presented to the University of Cambridge.

13. O.C. Zienkiewicz, Y.K Cheung — 1967. The finite element method in structural and continuum mechanics, New York: McGraw-Hill.

14. J.S. Przemieniecki — 1968. Theory of matrix structural analysis, New York: McGraw-Hill.

15. S.P. Timoshenko, S. Woinowsky - Kreiger — 1959. Theory of plates and shells , New York, McGraw-Hill, 2nd edition.

16. F.K. Bogner, R.L. Fox, L.A. Schmitt — 1965. Conference on Matrix Methods in Structural Mechanics, Wright Patterson Airforce Base, Ohio, The generation of inter element compatible stiffness and mass matrices by use of interpolation formulae.

17. R.E.D. Bishop, D.C. Johnson — 1960. The mechanics of vibration, Cambridge: Cambridge University Press.

18. G.A. Butlin, F.A. Leckie — 1966. A study of finite elements applied to plate flexure, vol. 3, Symposium of numerical Methods for Vibration Problems. University of Southampton.

19. M.J. Crocker — 1969. J. Sound Vib. vol. 9, Structural response to sonic booms.

20. H.H. Hubbard — 1968. Physics Today, vol. 21, Sonic Booms:

21. R. King — 1971. Post graduate work, University of Manitoba.

22. P.N. Gustafson, W.F. Stokey, C.F. Zorowski — 1953. J. Aero. Sci. vol. 20 An experimental study of natural vibrations of cantilever triangular plates.

1	$-a/2$	$-b/2$	$a^2/4$	$ab/4$	$b^2/4$	$-a^3/8$	$-ba^2/8$	$-ab^2/8$	$-b^3/8$	$a^4/16$	$ba^3/16$	$a^2b^2/16$	$ab^3/16$	$b^4/16$	$-a^5/32$	$-a^3b^2/32$	$-a^2b^3/32$	$-b^5/32$	$-a^4b^2/96$	$-b^4a^2/96$	$a^3b^3/64$	$a^5b^2/192$	$b^5a^2/192$
0	1	0	$-a$	$-b/2$	0	$3a^2/4$	$ab/2$	$b^2/4$	0	$-a^3/2$	$-3ba^2/8$	$-ab^2/4$	$-b^3/8$	0	$5a^4/16$	$3a^2b^2/16$	$ab^3/8$	0	$a^3b^2/12$	0	$-3a^2b^3/32$	$-5a^4b^2/96$	0
0	0	1	0	$-a/2$	$-b$	0	$a^2/4$	$ab/2$	$3b^2/4$	0	$-a^3/8$	$-ba^2/4$	$-3ab^2/8$	$-b^3/2$	0	$ba^3/8$	$3a^2b^2/16$	$5b^4/16$	0	$a^2b^3/12$	$-3a^3b^2/32$	0	$-5b^4a^2/96$
0	0	0	2	0	0	$-3a$	$-b$	0	0	$3a^2$	$3ab/2$	$b^2/2$	0	0	$-5a^3/2$	$-3ab^2/4$	$-b^3/4$	0	$-a^2b^2/2$	$b^4/8$	$3ab^3/8$	$5a^3b^2/12$	$-b^5/16$
0	0	0	0	1	0	0	$-a$	$-b$	0	0	$3a^2/4$	ab	$3b^2/4$	0	0	$-3ba^2/4$	$-3ab^2/4$	0	0	0	$9a^2b^2/16$	0	0
0	0	0	0	0	2	0	0	$-a$	$-3b$	0	0	$a^2/2$	$3ab/2$	$3b^2$	0	$-a^3/4$	$-3ba^2/4$	$-5b^3/2$	$a^4/8$	$-b^2a^2/2$	$3ba^3/8$	$-a^5/16$	$5b^3a^2/12$
1	$-a/2$	$b/2$	$a^2/4$	$-ab/4$	$b^2/4$	$-a^3/8$	$ba^2/8$	$-ab^2/8$	$b^3/8$	$a^4/16$	$-ba^3/16$	$a^2b^2/16$	$-ab^3/16$	$b^4/16$	$-a^5/32$	$-a^3b^2/32$	$a^2b^3/32$	$b^5/32$	$a^4b^2/96$	$-a^2b^4/96$	$-a^3b^3/64$	$-a^5b^2/192$	$-b^5a^2/192$
0	1	0	$-a$	$b/2$	0	$3a^2/4$	$-ab/2$	$b^2/4$	0	$-a^3/2$	$3ba^2/8$	$-ab^2/4$	$b^3/8$	0	$5a^4/16$	$3a^2b^2/16$	$-ab^3/8$	0	$-a^3b^2/12$	0	$3a^2b^3/32$	$5a^4b^2/96$	0
0	0	1	0	$-a/2$	b	0	$a^2/4$	$-ab/2$	$3b^2/4$	0	$-a^3/8$	$ba^2/4$	$-3ab^2/8$	$b^3/2$	0	$-ba^3/8$	$3a^2b^2/16$	$5b^4/16$	0	$-b^3a^2/12$	$-3a^3b^2/32$	0	$-5b^4a^2/96$
0	0	0	2	0	0	$-3a$	b	0	0	$3a^2$	$-3ab/2$	$b^2/2$	0	0	$-5a^3/2$	$-3ab^2/4$	$b^3/4$	0	$b^2a^2/2$	$b^4/8$	$-3ab^3/8$	$-5a^3b^2/12$	$b^5/16$
0	0	0	0	1	0	0	$-a$	b	0	0	$3a^2/4$	$-ab$	$3b^2/4$	0	0	$3ba^2/4$	$-3ab^2/4$	0	0	0	$9a^2b^2/16$	0	0
0	0	0	0	0	2	0	0	$-a$	$3b$	0	0	$a^2/2$	$-3ab/2$	$3b^2$	0	$-a^3/4$	$3ba^2/4$	$5b^3/2$	$-a^4/8$	$-b^2a^2/2$	$-3ba^3/8$	$a^5/16$	$-5a^2b^3/12$
1	$a/2$	$-b/2$	$a^2/4$	$-ba/4$	$b^2/4$	$a^3/8$	$-ba^2/8$	$ab^2/8$	$-b^3/8$	$a^4/16$	$-ba^3/16$	$a^2b^2/16$	$-ab^3/16$	$b^4/16$	$a^5/32$	$a^3b^2/32$	$-a^2b^3/32$	$-b^5/32$	$-a^4b^2/96$	$b^4a^2/96$	$-a^3b^3/64$	$-a^5b^2/192$	$-a^2b^5/192$
0	1	0	a	$-b/2$	0	$3a^2/4$	$-ab/2$	$b^2/4$	0	$a^3/2$	$-3ba^2/8$	$ab^2/4$	$-b^3/8$	0	$5a^4/16$	$3a^2b^2/16$	$-b^3a/8$	0	$-a^3b^2/12$	0	$-3a^2b^3/32$	$-5a^4b^2/96$	0
0	0	1	0	$a/2$	$-b$	0	$a^2/4$	$-ab/2$	$3b^2/4$	0	$a^3/8$	$-ba^2/4$	$3ab^2/8$	$-b^3/2$	0	$-ba^3/8$	$3a^2b^2/16$	$5b^4/16$	0	$-b^3a^2/12$	$3a^3b^2/32$	0	$5b^4a^2/96$
0	0	0	2	0	0	$3a$	$-b$	0	0	$3a^2$	$-3ab/2$	$b^2/2$	0	0	$5a^3/2$	$3ab^2/4$	$-b^3/4$	0	$-a^2b^2/2$	$-b^4/8$	$-3ab^3/8$	$-5a^3b^2/12$	$b^5/16$
0	0	0	0	1	0	0	a	$-b$	0	0	$3a^2/4$	$-ab$	$3b^2/4$	0	0	$-3ba^2/4$	$3ab^2/4$	0	0	0	$9a^2b^2/16$	0	0
0	0	0	0	0	2	0	0	a	$-3b$	0	0	$a^2/2$	$-3ab/2$	$3b^2$	0	$a^3/4$	$-3ba^2/4$	$-5b^3/2$	$a^4/8$	$b^2a^2/2$	$-3ba^3/8$	$a^5/16$	$-5b^3a^2/12$
1	$a/2$	$b/2$	$a^2/4$	$ab/4$	$b^2/4$	$a^3/8$	$ba^2/8$	$ab^2/8$	$b^3/8$	$a^4/16$	$ba^3/16$	$a^2b^2/16$	$ab^3/16$	$b^4/16$	$a^5/32$	$a^3b^2/32$	$a^2b^3/32$	$b^5/32$	$a^4b^2/96$	$b^4a^2/96$	$a^3b^3/64$	$a^5b^2/192$	$b^5a^2/192$
0	1	0	a	$b/2$	0	$3a^2/4$	$ab/2$	$b^2/4$	0	$a^3/2$	$3ba^2/8$	$ab^2/4$	$b^3/8$	0	$5a^4/16$	$3a^2b^2/16$	$ab^3/8$	0	$a^3b^2/12$	0	$3a^2b^3/32$	$5a^4b^2/96$	0
0	0	1	0	$a/2$	b	0	$a^2/4$	$ab/2$	$3b^2/4$	0	$a^3/8$	$ba^2/4$	$3ab^2/8$	$b^3/2$	0	$ba^3/8$	$3a^2b^2/16$	$5b^4/16$	0	$b^3a^2/12$	$3a^3b^2/32$	0	$5a^2b^4/96$
0	0	0	2	0	0	$3a$	b	0	0	$3a^2$	$3ab/2$	$b^2/2$	0	0	$5a^3/2$	$3ab^2/4$	$b^3/4$	0	$a^2b^2/2$	$-b^4/8$	$3ab^3/8$	$5a^3b^2/12$	$-b^5/16$
0	0	0	0	1	0	0	a	b	0	0	$3a^2/4$	ab	$3b^2/4$	0	0	$3ba^2/4$	$3ab^2/4$	0	0	0	$9a^2b^2/16$	0	0
0	0	0	0	0	2	0	0	a	$3b$	0	0	$a^2/2$	$3ab/2$	$3b^2$	0	$a^3/4$	$3ba^2/4$	$5b^3/2$	$-a^4/8$	$a^2b^2/2$	$3ba^3/8$	$-a^5/16$	$5a^2b^3/12$

Table 1. Transformation matrix $\underline{T}$

27

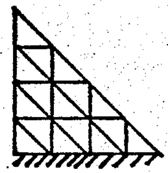
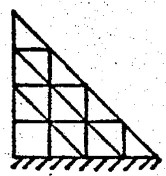
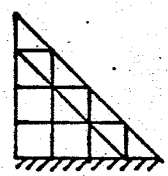
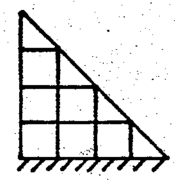
	FREQUENCY (Hz.)				
MODE NUMBER					Experimental Results Reference (23)
1	36.54124	36.53947	36.53897	36.53895	34.5
2	138.9636	138.9567	138.9550	138.9528	136.0
3	193.6010	193.5815	193.5754	193.5699	190.0
4	332.7147	332.7060	332.6953	332.6240	325.0
5	453.2197	453.2150	453.0388	452.9050	441.0
6	589.2431	589.1010	589.0678	588.6882	578.0
7	664.0397	663.8512	663.7170	662.9144	
8	798.0966	797.7241	796.9290	796.3389	
9	948.1436	947.2899	946.1370	944.4312	
10	1092.781	1092.583	1091.741	1088.801	

Table 2. Natural Frequencies of Cantilevered Triangular Plate

MODE	PERCENTAGE ERROR IN NATURAL FREQUENCIES								
	CONSTRAINED DEGREES OF FREEDOM								
	18			82			194		
	Triangular	Combination	Rectangular	Triangular	Combination	Rectangular	Triangular	Combination	Rectangular
1	.051420	.026224	.013694	.001849	.000274	.000137	.000137	.000068	.000000
2	.647567	.561514	.475779	.015254	.008651	.004096	.001307	.000953	.000494
3	2.57661	1.75478	.607637	.053796	.023330	.002483	.005704	.002886	.000224
4	4.33789	3.11280	1.76276	.092325	.063524	.023585	.009692	.006585	.002716
5	2.53034	4.67162	6.94001	.156811	.026241	.013625	.017117	.007823	.001027
6	12.5735	7.96505	4.54769	.302909	.171374	.066020	.033797	.019452	.005377
7	14.3049	12.2900	9.03932	.446281	.311655	.160690	.044374	.028337	.007863
8	21.2642	9.49459	1.46413	.159548	.137561	.017634	.024232	.011004	.001191
9	21.2010	19.7880	18.2945	.740316	.337328	.080119	.086657	.042178	.004697
10	44.0694	18.7799	14.9770	.658372	.525162	.439974	.076978	.046875	.019244

Table 3. Percentage Error Associated with the Ten Lowest Natural Frequencies of the Simply-supported Rectangular Plate

PERCENTAGE VARIATION IN NATURAL FREQUENCIES RELATIVE TO VALUES OF CLAASEN AND THORNE (8)				
MODE	UM6	SKSMP6	NRCD (Q)	NRCD (P)
1	-.17	-.03	-.16	-.17
2	-.43	-.14	-.40	-.41
3	-.05	.08	-.00	-.02
4	-.56	-.25	-.51	-.53
5	-.21	.21	-.06	-.09
6	.07	.36	.17	.15
7	-.97	-.32	-.73	-.80
8	-.03	.08	.07	.07
9	-.45	.32	-.11	-.14
10	-.13	.26	.15	.26

Table 4. Percentage Variation in the Eigenvalues of a 40:27 Clamped Plate for 170 Constrained Degrees of Freedom (Equivalent to 36 Rectangular or 72 Triangular Elements)

RELATIVE HUMIDITY 38%		RELATIVE HUMIDITY 79%	
Frequency (Hz.)	E lb/in^2 $x10^5$	Frequency (Hz.)	E lb/in^2 $x10^5$
12.1	5.10	12.1	5.10
91.9	7.45	87.9	6.81
263.	7.79	260.	7.61
536.	8.44	533.	8.33

$\rho = 0.00492$ $lb./in^2$

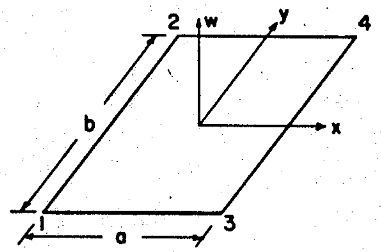
Thickness of beam $t = 0.112$ in.

Table 5. Experimental results for the dynamic Young's modulus of Perspex.

NATURAL FREQUENCIES (Hz)		
Mode	Theoretical	Experimental
1	179	174
2	184	178
3	208	200
4	219	222
5	281	
6	297	
7	303	
8	381	
9	401	
10	407	

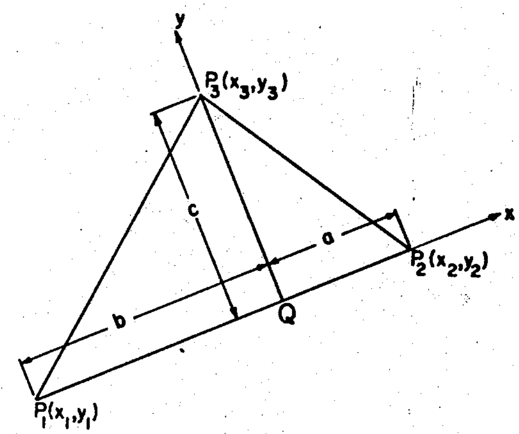
Table 6 . Theoretical and Experimental Natural Frequencies of the Sloping Roofed Perspex Model

(a) UM6



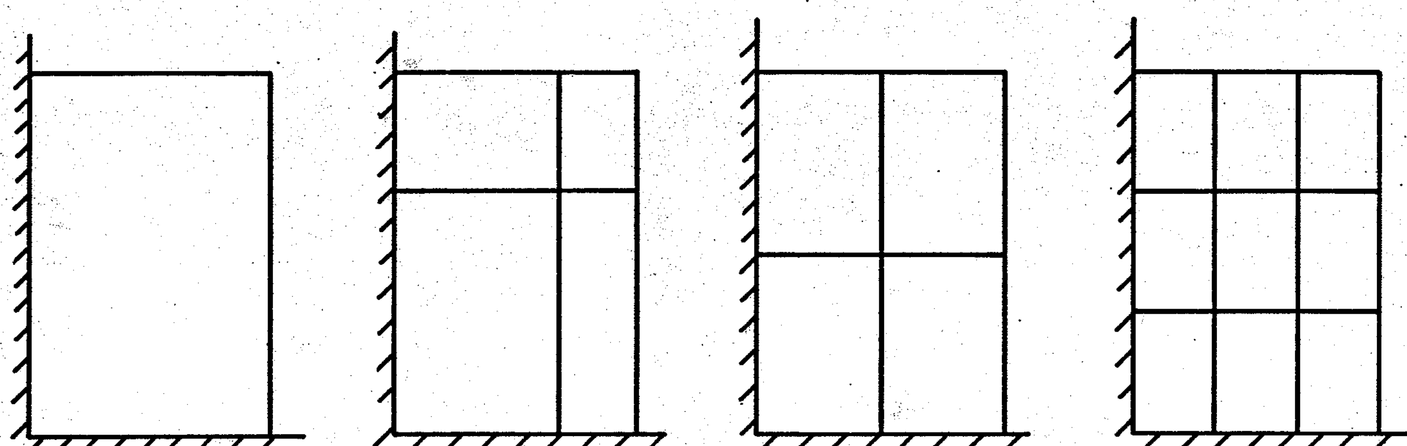
$$w(x,y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 + a_{10} y^3 + a_{11} x^4 + a_{12} x^3 y + a_{13} x^2 y^2 + a_{14} xy^3 + a_{15} y^4 + a_{16} x^5 + a_{17} x^3 y^2 + a_{18} y^3 x^2 + a_{19} y^5 + a_{20}\left(\frac{yb}{2} - \frac{2}{3}\frac{y^3}{b}\right)x^4 + a_{21}\left(\frac{xa}{2} - \frac{2}{3}\frac{x^3}{a}\right)y^4 + a_{22} x^3 y^3 + a_{23}\left(\frac{yb}{2} - \frac{2}{3}\frac{y^3}{b}\right)x^5 + a_{24}\left(\frac{xa}{2} - \frac{2}{3}\frac{x^3}{a}\right)y^5$$

(b) NRCD

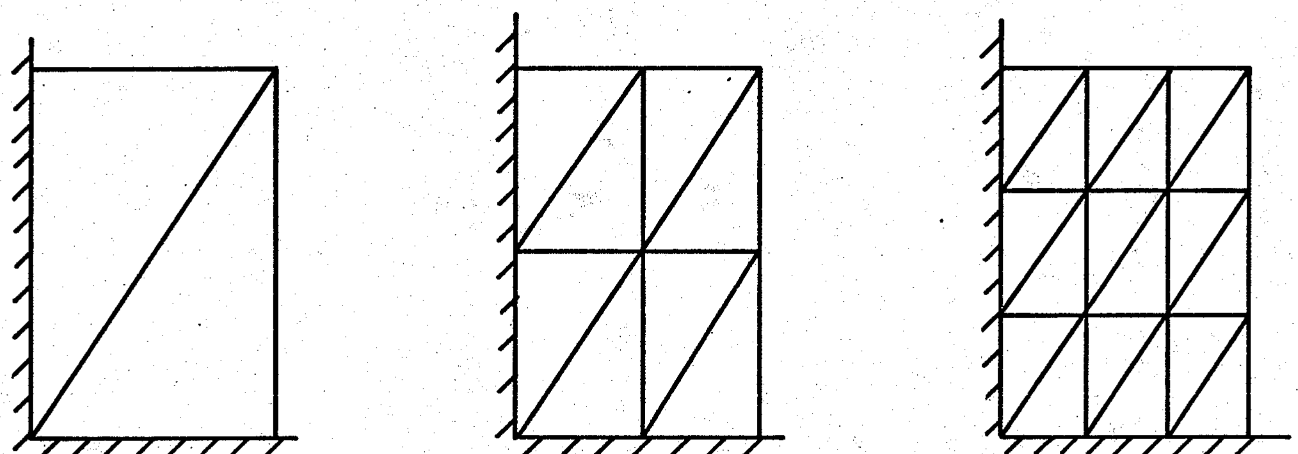


$$w(x,y) = a_1 + a_2 x + a_3 y + a_4 x^2 + a_5 xy + a_6 y^2 + a_7 x^3 + a_8 x^2 y + a_9 xy^2 + a_{10} y^3 + a_{11} x^4 + a_{12} x^3 y + a_{13} x^2 y^2 + a_{14} xy^3 + a_{15} y^4 + a_{16} x^5 + a_{17} x^3 y^2 + a_{18} x^2 y^3 + a_{19} xy^4 + a_{20} y^5$$

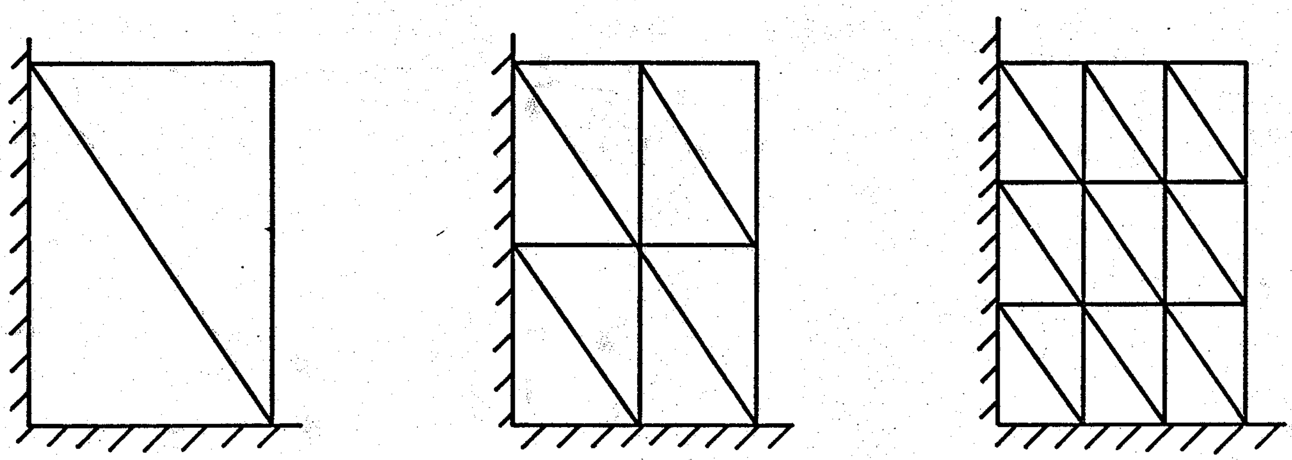
Figure 1. DISPLACEMENT FUNCTIONS FOR THE UM6 AND NRCD ELEMENTS



ARRANGEMENT OF RECTANGULAR ELEMENTS



Q ARRANGEMENT OF NRCD



P ARRANGEMENT OF NRCD

Figure 2. LAYOUT OF FINITE ELEMENTS FOR DYNAMIC ANALYSIS OF THE RECTANGULAR PLATE (NB. only quarter of plate shown)

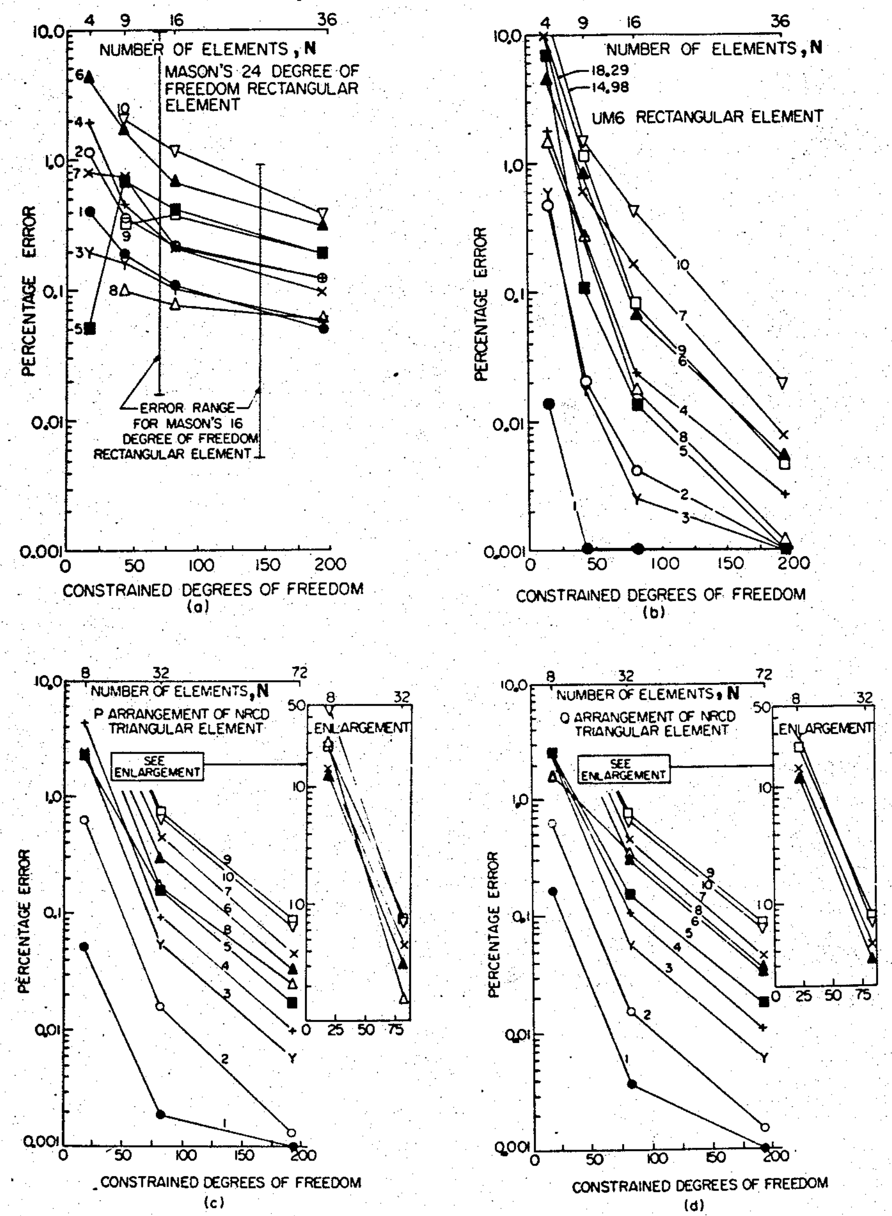


Figure 3. PERCENTAGE ERROR IN THE NATURAL FREQUENCIES OF A 40·27 SIMPLY SUPPORTED PLATE FOR DIFFERENT TRIANGULAR AND RECTANGULAR ELEMENTS.

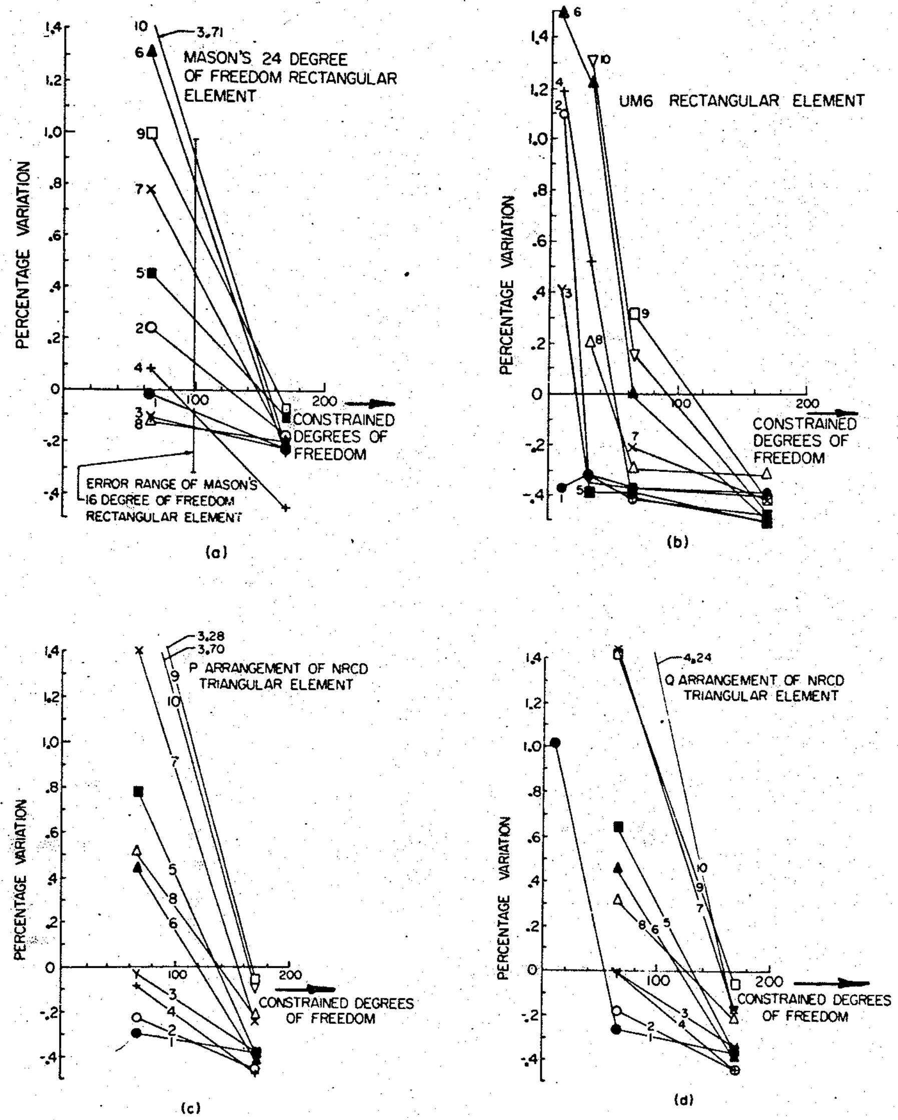
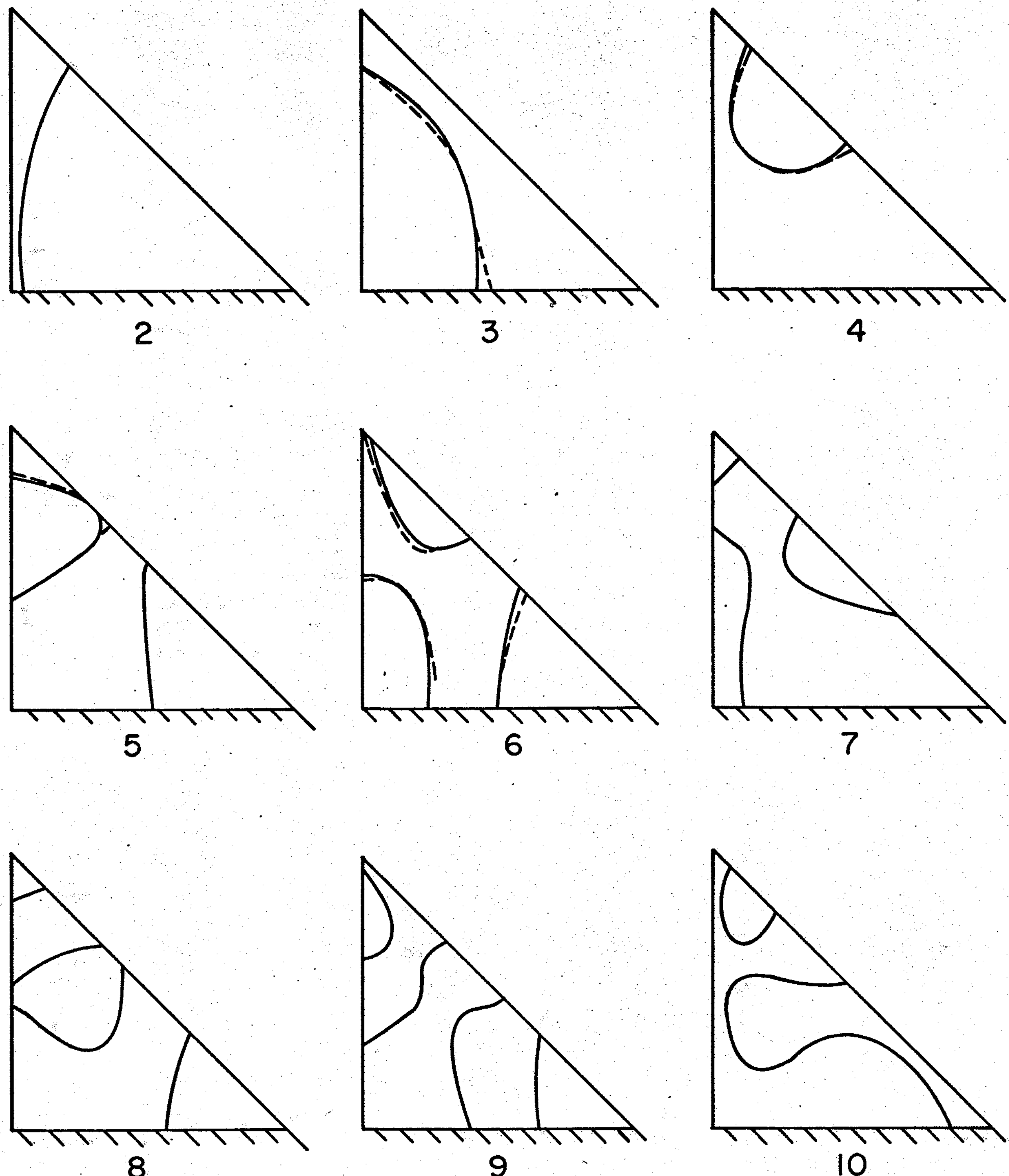


Figure 4. PERCENTAGE VARIATION IN THE NATURAL FREQUENCIES OF A 40:27 CLAMPED PLATE FOR DIFFERENT TRIANGULAR AND RECTANGULAR ELEMENTS



KEY : NUMBERS SHOWN GIVE MODE NUMBERS
------ EXPERIMENTAL NODAL LINES
——— CALCULATED NODAL LINES

Figure 5. EXPERIMENTAL AND CALCULATED NODAL LINES FOR A CANTILEVERED TRIANGULAR PLATE WITH ASPECT RATIO 1

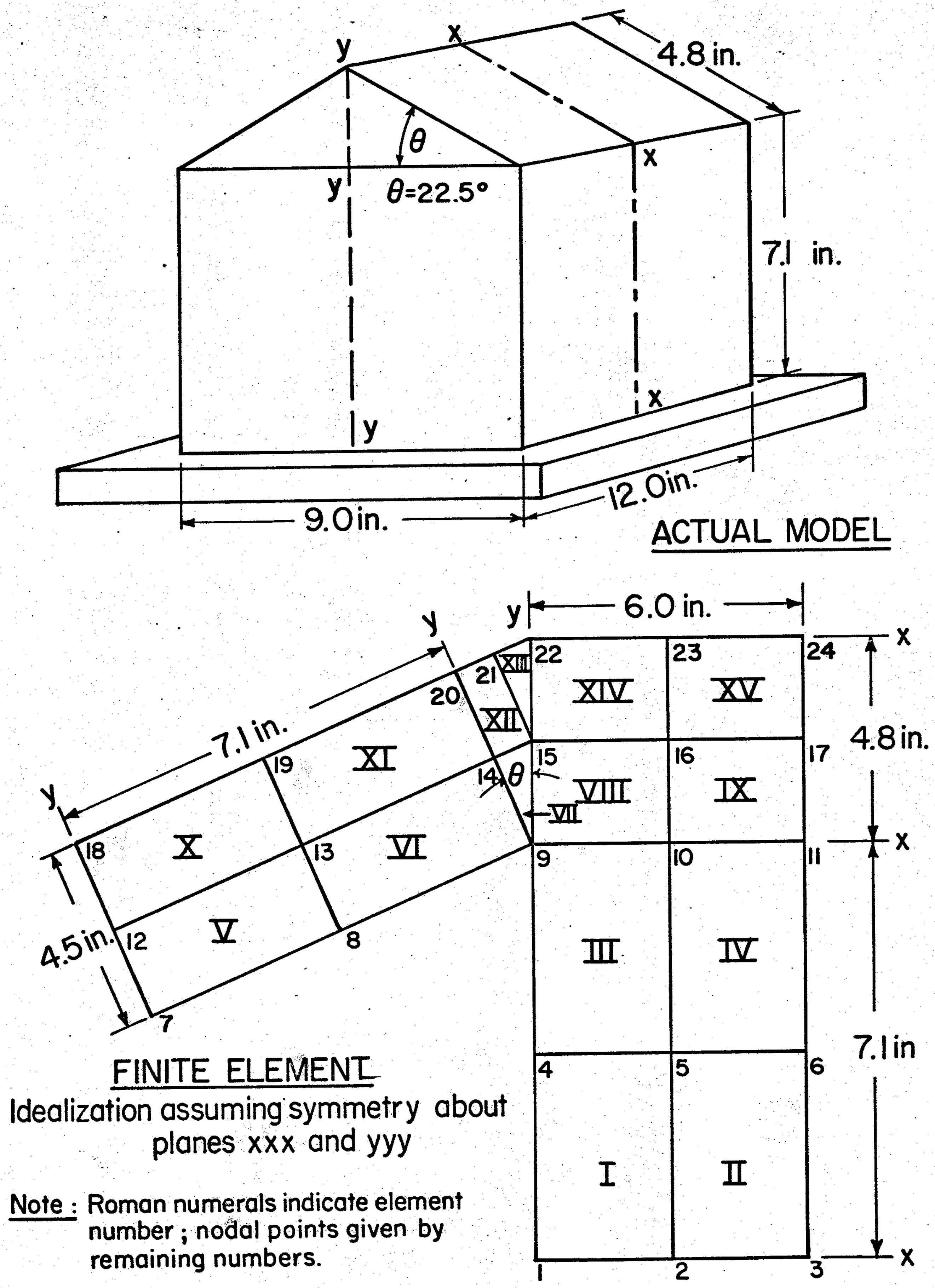
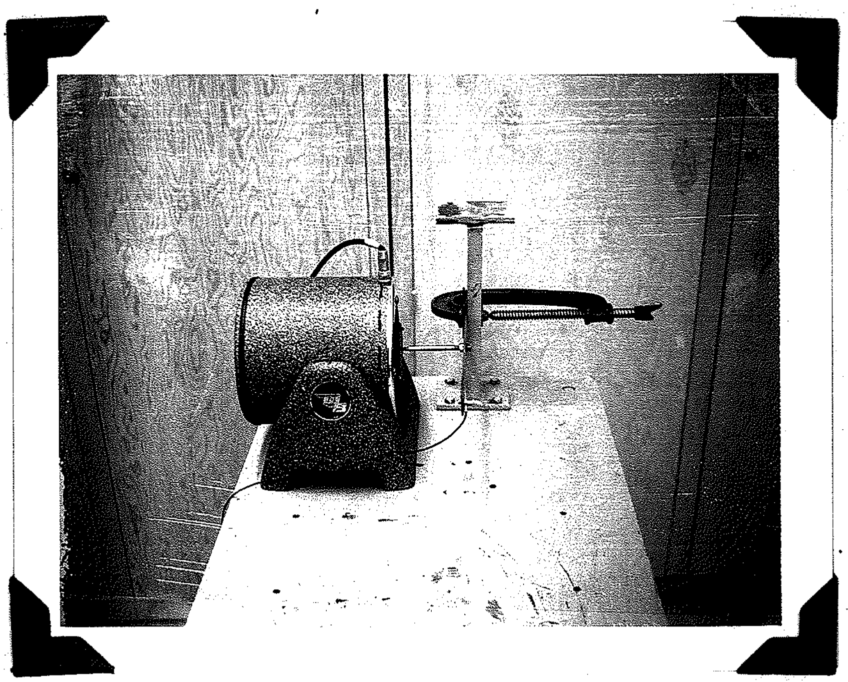


Figure 6 . Showing the Actual Model and a Finite Element Idealization.

(a) Experimental Setup



(b) Method of Attaching the Exciter to the Beam

Figure 7. Experimental Setup for Determining the Bending Stiffness of Perspex

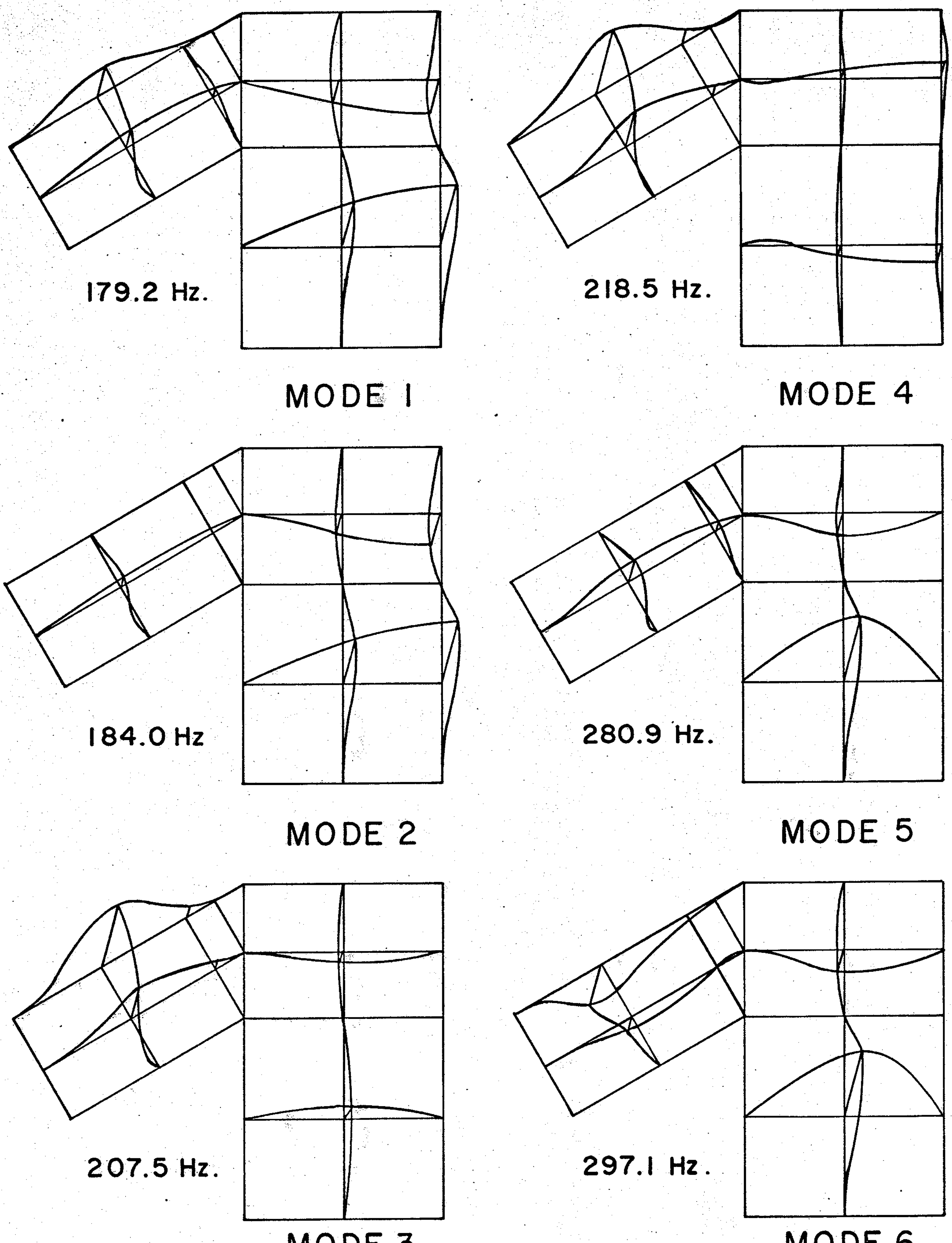


Figure 8 . Showing the first six mode shapes of a box with a 22 1/2° sloping roof.

APPENDIX

THE COMPUTER PROGRAMS

Introduction

FORTRAN subroutines used to calculate the natural frequencies and normal modes of the plate and box with a sloping roof are listed and discussed briefly. All those subroutines not developed by the author are omitted as the necessary specifications have been detailed by the respective authors. The specifications in Section 1 give the use of the subroutines; quantities passed to the subroutines in the argument list;brief details of the computational procedure and relevant particulars such as data required and the output from the subroutine.

Section 2 contains those trigonometric relations which are required in the imposition of boundary conditions along edges not aligned with the global axes.

SECTION 1

Due to the limiting storage facilities of the digital computer,the computer program is divided into three sections. The constrained mass and stiffness matrices are placed on disk in the first two sections. Then the matrices are taken off disk in the third section in order to evaluate the natural frequencies and mode shapes. Figures 9 and 10 illustrate these procedures with the appropriate flow diagrams.

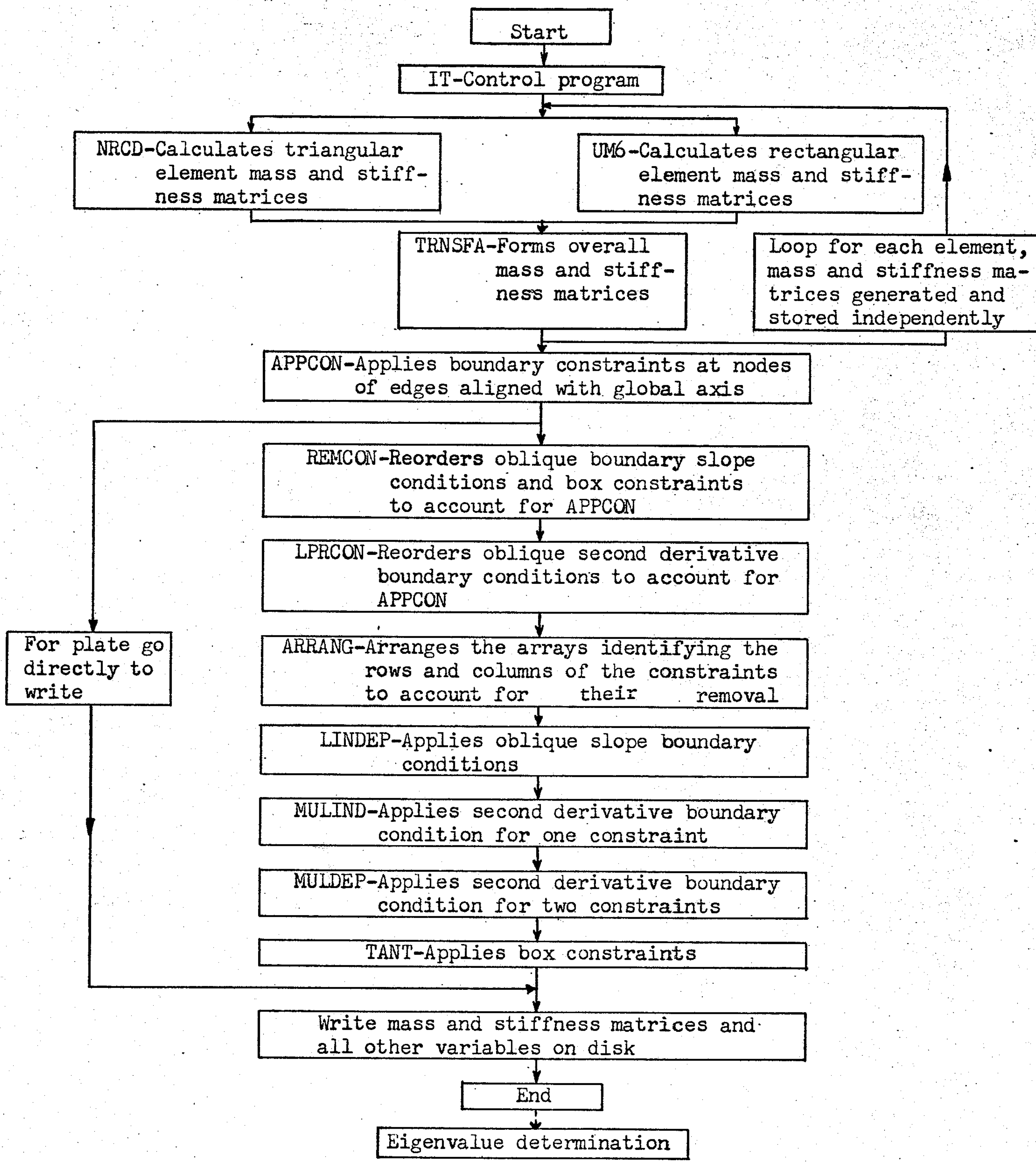
The formulation of the mass and stiffness matrices of the triangular element,NRCD,is identical to that outlined in the program listings supplied by National Research Council (10).

The calculation of the mass and stiffness matrices for the rectangular element,UM6,is given in Chapter 2.

The order in which matrix elements are stored in element arrays must be specified. To do this it is only necessary to specify the order in the element diplacement vectors. The elements in the vectors are grouped so that the nodal displacements for any one node are together. For NRCD and UM6,these displacements are stored in the order $w,w_x,w_y,w_{xx},w_{xy},w_{yy}$. In the finite element idealization,nodal points are numbered by the programmer so that the order in a vector is self apparent. The convention adopted for the numbering system of UM6 is shown in Figure 1(a) while NRCD only requires a clockwise sequence. Displacements are stored for node 1 followed by those of node 2 and node 3 and,in the case of UM6,node 4. The element matrices are transferred to overall matrices.

Subroutines developed by Mason(3) and Popplewell(1) (APPCON and LINDEP) are used to apply boundary constraints of sides aligned with the global axes. They also form the basis of subroutines used for the remaining constraints. Boundary constraints on oblique edges require the coding of corresponding trigonometric relations (see Section 2).

A QR two step eigenvalue subroutine deals adequately with matrices of order 96. The reinsertion of constraints into the modal vector employs the subroutines of Mason(3) and Popplewell (1)(UNAPP and UNLIN) as well as subroutines based on these routines.



(Subroutine names given in capitals.)

Figure 9. Flow Diagram of Computer Program Used for Generation of Overall Mass and Stiffness Matrices

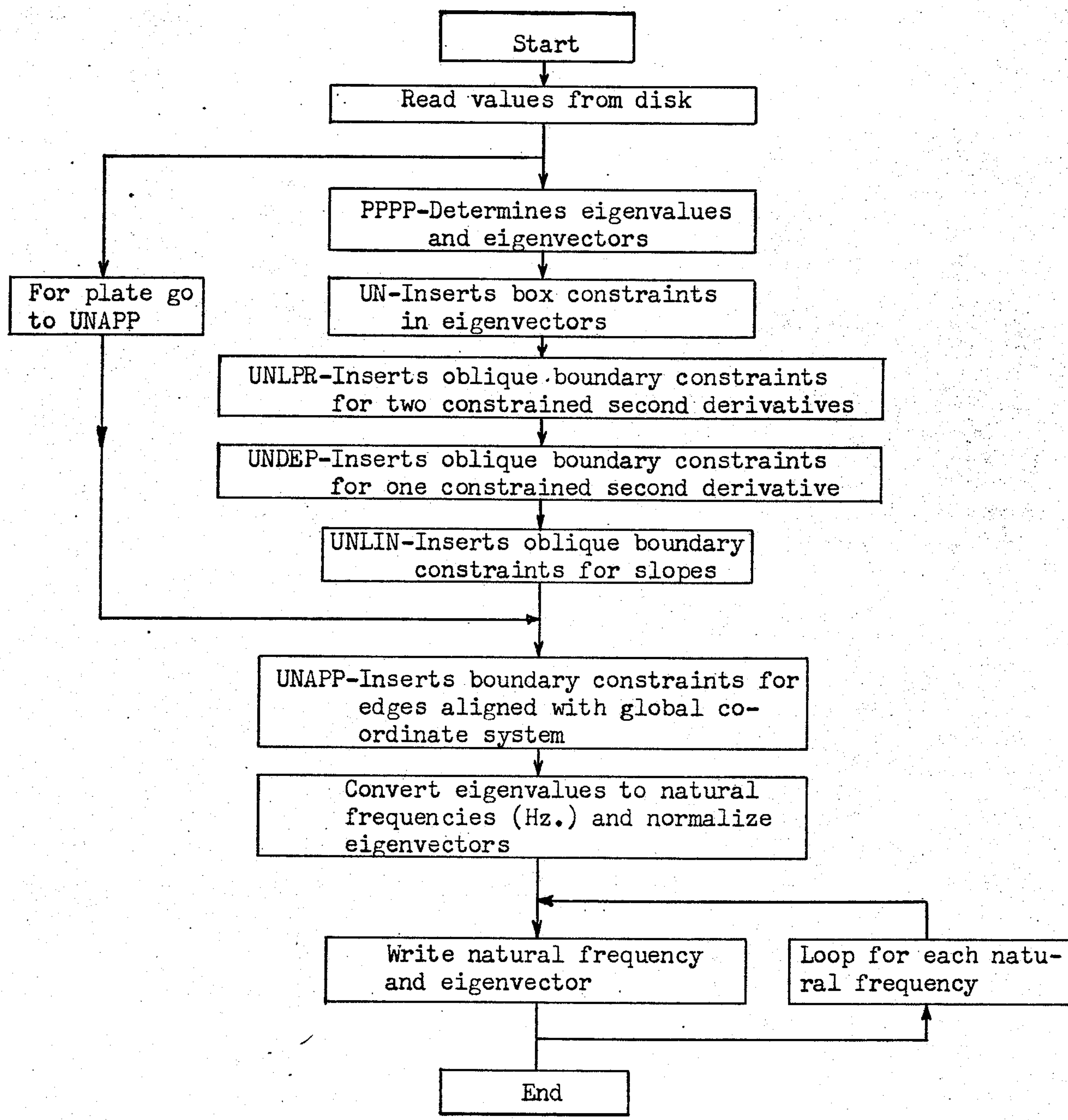


Figure 10. Flow Diagram of Computer Program Used for the Determination of Natural Frequencies and Mode Shapes

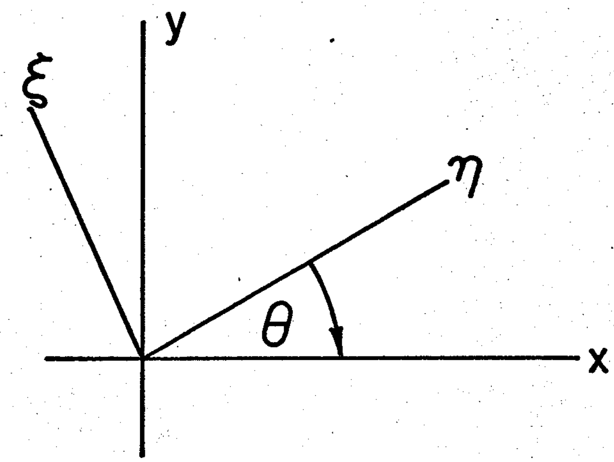


Figure 11. OBLIQUE BOUNDARY

SUBROUTINE UM6

1. Purpose

Calculates the element mass and stiffness matrices of rectangular element UM6.

2. Argument list

SUBROUTINE UM6(X,Y,LTL,GNU,R,T,SI,CO,NE)

X and Y are the x and y dimensions respectively of element UM6. LTL is the element number in the grid idealization; GNU is Poisson's ratio; R and T are the element stiffness and mass matrices,respectively,of size (24,24). SI and CO are the values of sine and cosine for the angle between the global and local x axes respectively. NE is the actual number of elements in the grid idealization.

3. Method

The subroutine calculates the element mass and stiffness matrices of UM6 using formulae (11) and (20).

4. Other routines used

QUT forms the transformation matrix in Table 1.
MINV calculates the inverse of the transformation matrix.

5. Printing

The values of the local x and y dimensions of an element are printed.

6. Miscellaneous

This subroutine is used in a loop with subroutine TRNSFA.

SUBROUTINE TRNSFA

1. Purpose

This subroutine transfers the mass and stiffness matrices of a single element to the overall , unconstrained mass and stiffness matrices for the complete system.

2. Argument list

SUBROUTINE TRNSFA(BKO,SM,SK,NGE,IE,KE,KSIZE,N,NSUB,NCORN, NTCOR,KROP1)

BKO is the overall stiffness or mass matrix(depending on which is being calculated). It is dimensioned as (KSIZE,KSIZE) where KSIZE is the total unconstrained degrees of freedom of the grid idealization. SM and SK are the element mass and stiffness matrices,respectively,of size (24,24). IE is the number of the element within the grid idealization which is to be transferred to the overall matrix. KE is the number of elements in the grid idealization; N always equals 24; NSUB is the number of degrees of freedom per corner; NCORN is the number of corners of UM6(4); and NTCOR is the number of corners of NRCD(3). KROP1 is used to determine whether element IE is rectangular or triangular (if KROP1 is less than IE,then IE is rectangular). NGE is the array giving the global nodal point numbers of each corner of of an element..

3. Method

Element matrices are transferred to the overall matrices. Array NGE is used to give the position within the overall matrices to which element matrices are transferred.

4. Miscellaneous

This subroutine is used normally in a loop with subroutines UM6 and NRCD. The element mass or stiffness matrix of an element is calculated in each loop and transferred to the overall matrix. BKO must be cleared before entering the loop.

SUBROUTINE QUT

1. Purpose

This subroutine calculates the elements of the transformation matrix T shown in Table 1.

2. Argument list

SUBROUTINE QUT(Q,A,B)

Q is the transformation matrix with dimension (24,24). A and B are the x and y dimensions of element UM6 respectively.

3. Miscellaneous

QUT is used with subroutine UM6.

SUBROUTINE BOCOND

1. Purpose

Boundary conditions along edges which are not parallel to the axes of the global co-ordinate system leads to dependent nodal displacements and derivatives. This subroutine establishes the constants within the relationships and identifies the associated rows and columns in the overall mass and stiffness matrices.

2. Argument list

SUBROUTINE BOCOND(N,TH,CHAN,IPOT,DD,ICOT,TAIL,ICOK, NC,LUT,MUT,KUT,MEAT,NUL,NP)

N is the number of nodal points which are constrained in the oblique orientation. TH is the angle of the oblique edge orientation with respect to the global x axis. CHAN,DD, and TAIL are the three sets of multiplication constants (see Section 2 for more details). IPOT,ICOT, and ICOK identify the rows and columns to which the relationships of CHAN,DD, and TAIL are applied. NC is the number of constraints applied to the global co-ordinate system. MUT,LUT, and KUT are the number of times in which relations CHAN,DD, and TAIL,respectively are applied. MEAT and NUL give the required information regarding the type of oblique boundary condition to be applied (see Section 2). NP is the number of nodal points within the grid idealization.

3. Method

From Section 2, relationships are established which correspond to the boundary conditions imposed at nodal points along oblique edges. BOCOND reviews each nodal point with respect to the code of NUL and MEAT and calculates the associated constants at the necessary nodal points. If no oblique boundary constraints are to be applied, NUL and MEAT are set to zero.

4. Printing

The values of LUT,KUT, and MUT are printed.

5. Miscellaneous

The values of LUT,KUT, and MUT are determined within BOCOND and are used in subroutines LINDEP,MULIND,MULDEP,UNLPR, UNDEP, and UNLIN.

6. Restrictions

This subroutine cannot be used when TH= $0^\circ,90^\circ,180^\circ$,---. In these cases the required constraints can be applied within APPCON by making the necessary additions within MCN.

SUBROUTINE UNDEP

1. Purpose

Reinserts previously imposed boundary condition of one oblique second derivative into eigenvector VAB.

2. Argument list

SUBROUTINE UNDEP(CLAP,IPOS,VAB,NP,NV,MRED,KSIZE,NKIMES, DD,KS)

CLAP is a working array of dimension KSIZE. IPOS with dimension (NP,3) contains values which denote the positions of the boundary constraints to be inserted in VAB. VAB contains only non-zero nodal displacements in each column on entry. At this stage there are MRED elements in each column. On exit from the subroutine boundary constraints have been reinserted and there are (MRED+NKIMES) elements per eigenvector. There are NV such vectors. DD are the constant terms in the oblique trigonometric relationships which are evaluated in subroutine BOCOND. Array DD has the dimensions (NP,2).

3. Method

Boundary constraints are reinserted in eigenvector VAB in the IPOS(M,3)th location. The value placed in this location equals the trigonometric constant DD(M,1) times the element in position IPOS(M,1) plus constant DD(M,2) times the element in position IPOS(M,2). The procedure is repeated for M=1,2,---,NKIMES.

4. Miscellaneous

The order of subroutines UNLPR,UNLIN,UN, and UNDEP within the complete program depends upon the order of the subroutines MULDEP,MULIND,LINDEP, and TANT.

SUBROUTINE UNLPR

1. Purpose

Reinserts the previosly imposed boundary conditions of two oblique second derivatives into the eigenvector VAB.

2. Argument list

SUBROUTINE UNLPR(CLAP,IPOS,VAB,NP,NV,MRED,KSIZE,NKIMES, TAIL,KS)

CLAP is a working array of order KSIZE. IPOS with dimension (NP,3) contains values which denote the positions of the boundary constraints to be inserted in VAB.

VAB contains only non-zero nodal displacements in each column on entry. At this stage there are MRED numbers in each column. On exit from the subroutine,the boundary constraints have been reinserted and there are (MRED+2*NKIMES) elements per eigenvector. There are NV such vectors. TAIL are the constant terms in the oblique trigonometric relationships which are evaluated in BOCOND. Array TAIL has dimensions (NP,2).

3. Method

The eigenvectors are expanded and the two boundary constraints are reinserted in IPOS(M,2) and IPOS(M,3) positions of eigenvector VAB. These elements respectively equal 1/TAIL(M,1) and 1/TAIL(M,2) times the element in position IPOS(M,1). The procedure is repeated M=1,2,---,NKIMES.

4. Miscellaneous

The order of subroutines UNLPR,UN,UNLIN, and UNDEP within the complete program depends upon the order of subroutines MULDEP,MULIND,LINDEP, and TANT.

SUBROUTINE UN

1. Purpose

Reinserts the previosly imposed box constraints into eigenvector VAB.

2. Argument list

SUBROUTINE UN(VAB,IOT,MSIZE,KS,KUTIME,NV,MRED,RTH, CLAP,NTIMES)

IOT of dimension (KUTIME,3) contains values which denote the positions of the box constraints to be inserted in VAB. VAB contains in each column the non-zero nodal displacements of the box on entry. At this stage there are MRED numbers in each column. On exit from the subroutine,the box constraints have been reinserted and there are (MRED+2*NTIMES) elements per eigenvector. There are NV such vectors. KUTIME is the number of nodal points where oblique boundary constraints are to be applied. RTH is the angle which the common edge makes with the global x axis. CLAP is a working array of dimension MSIZE. NTIMES denotes the number of nodal points to which box constraints must be applied. MSIZE is the unconstrained size of BKO(the overall mass or stiffness matrix).

3. Method

The box constraints are reinserted using appropriate trigonometric relations into NV eigenvectors at positions prescribed by IOT. For each insertion,the eigenvector is expanded.

4. Miscellaneous

The order of subroutines UNLPR,UN,UNLIN, and UNDEP within the complete program depends upon the order of the subroutines MULDEP,MULIND,LINDEP, and TANT.

SUBROUTINE IT

1. Purpose

This is the controlling subroutine for the calculation of a system's free response. Rectangular and triangular elements with six unknowns per corner are used.

2. Argument list

SUBROUTINE IT(X,Y,NUL,MEAT,NGE,IPOS,IOS,MCN,X1,X2, X3,Y1,Y2,Y3,IOT,BKO,C,ICOK,IPOT,DD,TAIL,CHAN,ICOKE,ICOT,ICOTE, ISIZE,IU,IC,IZ,IN,IH,NUM)

BKO is a square array with dimensions (ISIZE,ISIZE) and is used to store the mass or stiffness matrix. ISIZE is equal to the system's unconstrained degrees of freedom.

NGE contains the global nodal points associated with the corner of each element. X and Y are arrays containing the x and y dimensions of the rectangular elements. They are dimensioned as IZ where IZ equals the number of elements in the grid. X1, X2,X3,Y1,Y2, and Y3 are arrays which contain the x and y co-ordinates of the three corners of the triangular element in terms of its local co-ordinate system. They are dimensioned as IZ.

MCN is the array of constrainted global displacements. These constraints exclude box and oblique boundary constraints. IC is the dimension size of MCN. ICOK,ICOT, and IPOT are the three different oblique boundary constraints. IPOT has dimensions (IU,2) and ICOK and ICOT have dimensions (IU,3),where IU denotes the total number of global nodal points requiring oblique boundary constraints. CHAN,dimensioned as (IU),DD and TAIL(both dimensioned as (IU,2)) are the constants associated with the three oblique boundary constraints(see Section 2). IOT,dimensioned as (IU,3), contains the box constraints. MEAT and NUL are arrays of size (IN) and denote the type of oblique boundary condition to be applied at each node.

IN is equal to the number of global nodal points. NUM,dimensioned as (IN),contains the global numbers of those rectangular elements which have local co-ordinate axes not aligned with the global axes.

3. Method

Outlined in Chapters 2 and 3.

4. Other routines used

BOCOND,NRCD,UM6,TRNSFA,APPCON,REMCON,LPRCON,ARRANG, LINDEP,MULIND,MULDEP, and TANT

REMCON,LPRCON,BOCOND,LINDEP,MULIND,MULDEP,TANT are not necessary for the rectangular plate analysis.

5. Output

Writes the constrained mass and stiffness matrices and other important variables on to disk.

SUBROUTINE ARRANG

1. Purpose

To impose the boundary and box constraints specified in Section 2,four subroutines are used(LINDEP,MULDEP,MULIND, and TANT);one for each of the four types of constraints.ARRANG renumbers the appropriate rows and columns of the overall mass and stiffness matrices to account for the order in which the four types of constraints are applied.

2. Argument list

SUBROUTINE ARRANG(IPOT,ICOT,ICOK,IOS,LUT,KUT,MUT,NTIMES, NP,IPOS,ICOTE,ICOKE,NZZ)

IPOT,ICOT,ICOK, and IOS are arrays which identify those rows and columns within the overall matrices which are to be renumbered. MUT,LUT,KUT, and NTIMES are the respective number of times in which each of the four types of constraints are applied. LUT,MUT, and KUT are evaluated in subroutine BOCOND while NTIMES has to be read into the control program. NP is the number of global nodal points in the grid idealization. IPOS,ICOTE, and ICOKE are arrays used for working space. NZZ equals the number of nodal points with oblique boundary constraints.

3. Method

IPOT(1,2) is compared with the values in ICOT,ICOK, and IOS and,if it is less than any of these values,the values contained in ICOT, ICOK, and IOS are reduced by one. Then IPOT(1,2) is compared with all other elements of array IPOT. All elements of higher value are reduced by one. This is repeated for all values of IPOT(M,2) M=1,2,---,MUT. Following the same procedure, array ICOT is compared with arrays ICOK and IOS. Finally, ICOK is compared to IOS using the same procedure outlined above.

4. Printing

IPOT, ICOK, ICOT, and IOS are printed at the beginning and end of the subroutine.

SUBROUTINE LPRCON

1. Purpose

Due to the application of non-oblique boundary conditions contained in MCN, the elements of IOST have to be renumbered. (IOST contains the position of the rows and columns in the overall matrices of the complete system which are involved in the second derivative oblique boundary constraints.)

2. Argument list

SUBROUTINE LPRCON(MCN,IOST,KSIZE,NSIZE,KC,NC,NTIMES)

MCN is the array containing those boundary constraints which can be applied immediately in the global co-ordinate system. It is dimensioned as KC. NC is equal to KC. NTIMES constitutes the number of global nodal points to which the particular type of second derivative boundary condition is to be applied. IOST defines the rows and columns involved in the constraint. KSIZE and NSIZE equal the total number of degrees of freedom of the system. The integers stored in MCN are in increasing numerical value.

3. Method

Each element of MCN is compared with the elements of array IOST. Any element of IOST which has a value greater than the element in MCN is reduced by one. This process is repeated for each element of MCN.

4. Restrictions

No original element of IOST can be contained in MCN.

SUBROUTINE MULIND

1. Purpose

This subroutine imposes oblique second derivative boundary conditions on the system's overall matrices for those global nodal points which have one second derivative constraint.

2. Argument list

SUBROUTINE MULIND(BKO,NSIZE,KSIZE,DD,ICOT,NTIMES,NKLIME)

BKO is the mass or stiffness matrix with dimension (NSIZE, NSIZE). KSIZE equals the actual size of matrix BKO on entry to MULDEP. DD,with dimension (NKLIME,2),is the array containing the constants of the trigonometric relationships evolving from the constraint of one second derivative at a global nodal point on a oblique edge (see Section 2).ICOT identifies the rows and columns of matrix BKO to which the trigonometric relations apply. NP is equal to the number of global nodal points involved in oblique boundary constraints.

3. Method

The relations in Section 2 are applied to matrix BKO in order to eliminate dependent nodal displacements. Rows and columns of BKO,specified by ICOT(NKLIME,3), are multiplied by DD(NKLIME,1) and DD(NKLIME,2) and added to rows and columns denoted by ICOT(NKLIME,1) and ICOT(NKLIME,2) respectively. This is repeated NTIMES. The ICOT(NKLIME,3)th row and column are removed from BKO and the matrix size is reduced by one.

4. Miscellaneous

The order in which MULIND,MULDEP,LINDEP, and TANT are applied is dictated by subroutine ARRANG.

SUBROUTINE MULDEP

1. Purpose

The subroutine imposes oblique second derivative boundary conditions on the system's overall matrices for those global nodal points which have two second derivative constraints.

2. Argument list

SUBROUTINE MULDEP(BKO,NSIZE,KSIZE,TAIL,ICOK,NP,NKIMES)

BKO is the mass or stiffness matrix with dimension (NSIZE, NSIZE). KSIZE equals the actual size of matrix BKO on entry to MULDEP. TAIL,with dimension (NKLIME,2), is the array containing the constants of the trigonometric relationships evolving from the constraint of two second derivatives at a global nodal point on an oblique edge (see Section 2).ICOK identifies the rows and columns of matrix BKO to which the trigonometric relations apply. NP is equal to the number of global nodal points involved in oblique boundary constraints.

3. Method

The relations in Section 2 are applied to matrix BKO in order to eliminate dependent nodal displacements. Rows and columns of BKO are specified in ICOK(NP,3) and ICOK(NP,2). They are multiplied by TAIL(NP,2) and TAIL(NP,1) respectively and added to the row and column of ICOK(NP,1). This procedure is repeated NKIMES. The rows and columns of ICOK(NP,2) and ICOK(NP,3) are removed from BKO and the matrix size is reduced by two.

4. Miscellaneous

The order in which MULIND,MULDEP,LINDEP, and TANT are applied is dictated by subroutine ARRANG.

SUBROUTINE TANT

1. Purpose

This subroutine imposes box constraints on the system's overall mass and stiffness matrices.

2. Argument list

SUBROUTINE TANT(BKO;KUTIME,MSIZE,IOT,NTIMES,RTH,MRED)

BKO is the mass or stiffness matrix with dimension (MSIZE, MSIZE). RTH is the angle between the x axis of the global co-ordinate system and the edge where box constraints are to be applied. IOT identifies those rows and columns of the matrix BKO to which box constraints are to be applied. NTIMES denotes the number of box constraints to be applied. KUTIME is equal to the number of global nodal points involved in the oblique boundary constraints. MRED is the size of matrix BKO after box constraints have been applied.

3. Method

The normal slopes of common oblique edges are equal. The resulting dependent rows and columns in matrix BKO are eliminated by adding algebraically rows and columns corresponding to the normal slope of one oblique edge to those corresponding to its common edge. This procedure is repeated for all global nodal points on the remaining common edges.The normal slope of the oblique edge is obtained by using the trigonometric relation involving slopes in the global co-ordinate system(equations (14) in Chapter 2). The edgewise slope must be zero for both common edges in order that the previous assumptions are valid. Therefore,once the rows and columns of BKO corresponding to the normal slopes of the oblique common edge are added to those corresponding to the normal slope of the other common edge, it remains to set the rows and columns corresponding to the oblique slopes(normal and edgewise) to zero. This is done simply by constraining both slopes in the global co-ordinate system to zero. From equation (14) of Chapter 2,the edgewise and normal slopes are effectively constrained to be zero.

Thus,rows and columns within BKO corresponding to the global slopes of points on the oblique common edge are eliminated and the matrix size reduced by two.

4. Miscellaneous

The order in which MULIND,MULDEP,LINDEP, and TANT are applied is dictated by subroutine ARRANG.

5. Restrictions

The subroutine cannot be used if RTH=0°,90°,180°,---. In such cases constraints can be applied as in reference(1).

```
      SUBROUTINE UM6(X,Y,LTL,GNU,R,T,SI,CO,NE)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION YY(28,28),Z(28,28),M(28),N(28),MUL(4,2)
      DIMENSION Q(24,24),R(24,24),T(24,24),S(24,24),FAC(30)
      DIMENSION DINK(4)
      DIMENSION X(NE),Y(NE)
      DATA M/0,1,0,2,1,0,3,2,1,0,4,3,2,1,0,5,3,2,0,4,4,1,3,3,5,5,1,3/
      DATA N/0,0,1,0,1,2,0,1,2,3,0,1,2,3,4,0,2,3,5,1,3,4,4,3,1,3,5,5/
      MUL(1,1)=20
      MUL(1,2)=21
      MUL(2,1)=21
      MUL(2,2)=22
      MUL(3,1)=23
      MUL(3,2)=24
      MUL(4,1)=24
      MUL(4,2)=25
  901 FORMAT(5X,2D15.4)
      WRITE(6,902) GNU,PNEUO,YMODO,THICKO,YMOD,THICK,RHO,PNEU,SI,CO
  902 FORMAT(10F12.4)
      A=X(LTL)
      B=Y(LTL)
      WRITE(6,901) A,B
      DINK(1)=B
      DINK(2)=A
      DINK(3)=B
      DINK(4)=A
      DO 799 I=1,28
      DO 799 J=1,28
      Z(I,J)=0.0D0
  799 YY(I,J)=0.0D0
      DO 60 I=1,24
      DO 60 J=1,24
      T(I,J)=0.0D0
      Q(I,J)=0.0D0
      R(I,J)=0.0D0
   60 S(I,J)=0.0D0
      CALL QUT(Q,A,B)
      R(1,1)=1.0D0
      R(2,2)=CO
      R(3,3)=CO
      R(3,2)=-SI
      R(2,3)=SI
      R(4,4)=CO**2
      R(5,4)=-SI*CO
      R(6,4)=SI**2
      R(4,5)=2.0D0*SI*CO
      R(5,5)=CO**2-SI**2
      R(6,5)=-2.0D0*SI*CO
      R(4,6)=SI**2
      R(5,6)=SI*CO
      R(6,6)=CO**2
      CALL MINV(Q,24,24,DET)
      WRITE(6,1180)
      DO 61 I=1,24
      DO 61 J=1,6
      SUM1=0.0D0
      SUM2=0.0D0
      SUM3=0.0D0
```

```
      SUM4=0.0D0
      DO 62 K=1,6
      SUM1=SUM1+Q(I,K)*R(K,J)
      SUM2=SUM2+Q(I,K+6)*R(K,J)
      SUM3=SUM3+Q(I,K+12)*R(K,J)
   62 SUM4=SUM4+Q(I,K+18)*R(K,J)
      S(I,J)=SUM1
      S(I,J+6)=SUM2
      S(I,J+12)=SUM3
   61 S(I,J+18)=SUM4
      WRITE(6,1180)
      DO 50 I=1,24
      DO 50 J=1,24
      Q(I,J)=0.0D0
   50 R(I,J)=0.0D0
      WRITE(6,1180)
      DO 1004 I=5,15,2
      DO 1004 J=5,15,2
      Q(I,J)=((A**(I-4))*(B**(J-4)))/((2.0D0**((I-5)+(J-5)))*
     1((J-4)*(I-4)))
 1004 CONTINUE
      WRITE(6,1180)
      DO 51 I=1,28
      DO 51 J=I,28
      MI=M(I)
      MJ=M(J)
      NI=N(I)
      NJ=N(J)
      YY(I,J)=Q(MI+MJ+5,NI+NJ+5)
      YY(J,I)=YY(I,J)
      Z(I,J)=MI*MJ*(MI-1)*(MJ-1)*Q(MI+MJ+1,NI+NJ+5)
     1+ NI*NJ*(NI-1)*(NJ-1)*Q(MI+MJ+5,NI+NJ+1)
     2+(GNU*(MI*NJ*(MI-1)*(NJ-1)+MJ*NI*(MJ-1)*(NI-1))
     3+2.0D0*(1.0D0-GNU)*MI*MJ*NI*NJ)*Q(MI+MJ+3,NI+NJ+3)
      Z(J,I)=Z(I,J)
   51 CONTINUE
 1170 FORMAT(8D15.4)
      DO 185 I=1,4
      JOL1=MUL(I,1)
      JOL2=MUL(I,2)
      DO 80 J=1,28
      Z(JOL1,J)=Z(JOL1,J)*DINK(I)/2.0D0-Z(JOL2,J)*0.66666666/DINK(I)
   80 YY(JOL1,J)=YY(JOL1,J)*DINK(I)/2.0D0-YY(JOL2,J)*0.66666666/DINK(I)
 1180 FORMAT('  BKO ')
      DO 181 K=JOL2,28
      IF(K.EQ.28) GO TO 181
      DO 181 LL=1,28
      Z(K,LL)=Z(K+1,LL)
  181 YY(K,LL)=YY(K+1,LL)
      DO 82 L=1,28
      Z(L,JOL1)=Z(L,JOL1)*DINK(I)/2.0D0-Z(L,JOL2)*0.66666666/DINK(I)
   82 YY(L,JOL1)=YY(L,JOL1)*DINK(I)/2.0D0-YY(L,JOL2)*0.66666666/DINK(I)
      DO 183 II=JOL2,28
      IF(II.EQ.28) GO TO 183
      DO 183 JJ=1,28
      YY(JJ,II)=YY(JJ,II+1)
  183 Z(JJ,II)=Z(JJ,II+1)
  185 CONTINUE
```

```
      DO 187 I=1,24
      DO 187 J=1,24
      T(I,J)=YY(I,J)
  187 R(I,J)=Z(I,J)
  811 FORMAT(8D15.4)
      DO 72 I=1,24
      DO 72 J=1,24
      SUM2=0.0D0
      DO 73 K=1,24
   73 SUM2=SUM2+S(K,I)*R(K,J)
   72 Q(I,J)=SUM2
      DO 74 I=1,24
      DO 74 J=1,24
      SUM3=0.0D0
      DO 75 K=1,24
   75 SUM3=SUM3+Q(I,K)*S(K,J)
      R(I,J)=SUM3
   74 R(J,I)=SUM3
      DO 76 I=1,24
      DO 76 J=1,24
      SUM2=0.0D0
      DO 77 K=1,24
   77 SUM2=SUM2+S(K,I)*T(K,J)
   76 Q(I,J)=SUM2
      DO 78 I=1,24
      DO 78 J=1,24
      SUM3=0.0D0
      DO 79 K=1,24
   79 SUM3=SUM3+Q(I,K)*S(K,J)
      T(I,J)=SUM3
   78 T(J,I)=SUM3
      RETURN
      END
```

```
      SUBROUTINE TRNSFA(BKO,SM,SK,NGE,IE,KE,KSIZE,
     *N,NSUB,NCORN,NTCCR,KROP1       )
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION BKO(KSIZE,KSIZE),SM(N,N),
     *   SK(N,N),NGE(KE,NCORN)
      IF(IE.LT.KROP1) GO TO 2
      DO 3 I1=1,NTCOR
      DO 3 I2=1,NTCOR
      N3=NSUB*(NGE(IE,I1)-1)
      N4=NSUB*(NGE(IE,I2)-1)
      DO 3 L1=1,NSUB
      DO 3 L2=1,NSUB
      LA=L1+N3
      LB=L2+N4
      IA=L1+(I1-1)*NSUB
      IB=L2+(I2-1)*NSUB
    3 BKO(LA,LB)=BKO(LA,LB)+SK(IA,IB)
      GO TO 1
    2 CONTINUE
      DO 4 I1=1,NCORN
      DO 4 I2=1,NCORN
      N3=NSUB*(NGE(IE,I1)-1)
      N4=NSUB*(NGE(IE,I2)-1)
      DO 4 L1=1,NSUB
      DO 4 L2=1,NSUB
      LA=L1+N3
      LB=L2+N4
      IA=L1+(I1-1)*NSUB
      IB=L2+(I2-1)*NSUB
    4 BKO(LA,LB)=BKO(LA,LB)+SK(IA,IB)
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUT(Q,A,B)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION Q(24,24)
      Q(1,1)=1.0D0
      Q(1,2)=-A/2.0D0
      Q(1,3)=-B/2.0D0
      Q(1,4)=(A**2)/4.0D0
      Q(1,5)=A*B/4.0D0
      Q(1,6)=(B**2)/4.0D0
      Q(1,7)=-(A**3)/8.0D0
      Q(1,8)=-(B*A**2)/8.0D0
      Q(1,9)=-(A*B**2)/8.0D0
      Q(1,10)=-(B**3)/8.0D0
      Q(1,11)=(A**4)/1.6D01
      Q(1,12)=(B*A**3)/1.6D01
      Q(1,13)=((A**2)*(B**2))/1.6D01
      Q(1,14)=(A*B**3)/1.6D01
      Q(1,15)=(B**4)/1.6D01
      Q(1,16)=-(A**5)/32.0D0
      Q(1,17)=-((A**3)*(B**2))/3.2D01
      Q(1,18)=-((A**2)*(B**3))/3.2D01
      Q(1,19)=-(B**5)/3.2D01
      Q(1,20)=-((A**4)*(B**2))/9.6D01
      Q(1,21)=-((B**4)*(A**2))/9.6D01
      Q(1,22)=((A**3)*(B**3))/6.4D01
      Q(1,23)=((A**5)*(B**2))/1.92D02
      Q(1,24)=((B**5)*(A**2))/1.92D02
      Q(2,2)=1.0D0
      Q(2,4)=-A
      Q(2,5)=-B/2.0D0
      Q(2,7)=0.75D0*A**2
      Q(2,8)=A*B/2.0D0
      Q(2,9)=(B**2)/4.0D0
      Q(2,11)=-(A**3)/2.0D0
      Q(2,12)=-3.0D0*(B*A**2)/8.0D0
      Q(2,13)=-(A*B**2)/4.0D0
      Q(2,14)=-(B**3)/8.0D0
      Q(2,16)=5.0D0*(A**4)/1.6D01
      Q(2,17)=3.0D0*((A**2)*(B**2))/1.6D01
      Q(2,18)=(A*B**3)/8.0D0
      Q(2,20)=((A**3)*(B**2))/1.2D01
      Q(2,22)=-3.0D0*((A**2)*(B**3))/3.2D01
      Q(2,23)=-5.0D0*((A**4)*(B**2))/9.6D01
      Q(3,3)=1.0D0
      Q(3,5)=-A/2.0D0
      Q(3,6)=-B
      Q(3,8)=(A**2)/4.0D0
      Q(3,9)=A*B/2.0D0
      Q(3,10)=3.0D0*(B**2)/4.0D0
      Q(3,12)=-(A**3)/8.0D0
      Q(3,13)=-(B*A**2)/4.0D0
      Q(3,14)=-(3.0D0*(A*B**2))/8.0D0
      Q(3,15)=-(B**3)/2.0D0
      Q(3,17)=(B*A**3)/8.0D0
      Q(3,18)=3.0D0*((A**2)*(B**2))/1.6D01
      Q(3,19)=5.0D0*(B**4)/1.6D01
      Q(3,21)=((A**2)*(B**3))/1.2D01
      Q(3,22)=-3.0D0*((A**3)*(B**2))/3.2D01
```

```
Q(3,24)=-5.0D0*((B**4)*(A**2))/9.6D01
Q(4,4)=2.0D0
Q(4,7)=-3.0D0*A
Q(4,8)=-B
Q(4,11)=3.0D0*A**2
Q(4,12)=1.5D0*A*B
Q(4,13)=0.5D0*B**2
Q(4,16)=-2.5D0*A**3
Q(4,17)=-0.75D0*A*B**2
Q(4,18)=-0.25D0*B**3
Q(4,20)=-0.5D0*(A**2)*(B**2)
Q(4,21)=(B**4)/8.0D0
Q(4,22)=3.0D0*(A*B**3)/8.0D0
Q(4,23)=5.0D0*((A**3)*(B**2))/1.2D01
Q(4,24)=-(B**5)/1.6D01
Q(5,5)=1.0D0
Q(5,8)=-A
Q(5,9)=-B
Q(5,12)=0.75D0*A**2
Q(5,13)=A*B
Q(5,14)=0.75D0*B**2
Q(5,17)=-0.75D0*B*A**2
Q(5,18)=-0.75D0*A*B**2
Q(5,22)=9.0D0*((A**2)*(B**2))/1.6D01
Q(6,6)=2.0D0
Q(6,9)=-A
Q(6,10)=-3.0D0*B
Q(6,13)=0.5D0*A**2
Q(6,14)=1.5D0*A*B
Q(6,15)=3.0D0*B**2
Q(6,17)=-0.25D0*A**3
Q(6,18)=-0.75D0*B*A**2
Q(6,19)=-2.5D0*B**3
Q(6,20)=(A**4)/8.0D0
Q(6,21)=-0.5D0*(B**2)*(A**2)
Q(6,22)=3.0D0*(B*A**3)/8.0D0
Q(6,23)=-(A**5)/1.6D01
Q(6,24)=5.0D0*((B**3)*(A**2))/1.2D01
Q(7,1)=1.0D0
Q(7,2)=-A/2.0D0
Q(7,3)=B/2.0D0
Q(7,4)=(A**2)/4.0D0
Q(7,5)=-(A*B)/4.0D0
Q(7,6)=(B**2)/4.0D0
Q(7,7)=-(A**3)/8.0D0
Q(7,8)=(B*A**2)/8.0D0
Q(7,9)=-(A*B**2)/8.0D0
Q(7,10)=(B**3)/8.0D0
Q(7,11)=(A**4)/1.6D01
Q(7,12)=-(B*A**3)/1.6D01
Q(7,13)=((A**2)*(B**2))/1.6D01
Q(7,14)=-(A*B**3)/1.6D01
Q(7,15)=(B**4)/1.6D01
Q(7,16)=-(A**5)/3.2D01
Q(7,17)=-((A**3)*(B**2))/3.2D01
Q(7,18)=((A**2)*(B**3))/3.2D01
Q(7,19)=(B**5)/3.2D01
Q(7,20)=((A**4)*(B**2))/9.6D01
```

```
      Q(7,21)=-((A**2)*(B**4))/9.6D01
      Q(7,22)=-((A**3)*(B**3))/6.4D01
      Q(7,23)=-((A**5)*(B**2))/1.92D02
      Q(7,24)=-((B**5)*(A**2))/1.92D02
      Q(8,2)=1.0D0
      Q(8,4)=-A
      Q(8,5)=B/2.0D0
      Q(8,7)=0.75D0*A**2
      Q(8,8)=-0.5D0*A*B
      Q(8,9)=0.25D0*B**2
      Q(8,11)=-0.5D0*A**3
      Q(8,12)=3.0D0*(B*A**2)/8.0D0
      Q(8,13)=-0.25D0*A*B**2
      Q(8,14)=(B**3)/8.0D0
      Q(8,16)=5.0D0*(A**4)/16.0D0
      Q(8,17)=3.0D0*((A**2)*(B**2))/1.6D01
      Q(8,18)=-(A*B**3)/8.0D0
      Q(8,20)=-((A**3)*(B**2))/1.2D01
      Q(8,22)=3.0D0*((A**2)*(B**3))/3.2D01
      Q(8,23)=5.0D0*((A**4)*(B**2))/9.6D01
      Q(9,3)=1.0D0
      Q(9,5)=-A/2.0D0
      Q(9,6)=B
      Q(9,8)=0.25D0*A**2
      Q(9,9)=-0.5D0*A*B
      Q(9,10)=0.75D0*B**2
      Q(9,12)=-(A**3)/8.0D0
      Q(9,13)=0.25D0*B*A**2
      Q(9,14)=-3.0D0*(A*B**2)/8.0D0
      Q(9,15)=0.5D0*B**3
      Q(9,17)=-(B*A**3)/8.0D0
      Q(9,18)=3.0D0*((A**2)*(B**2))/1.6D01
      Q(9,19)=5.0D0*(B**4)/1.6D01
      Q(9,21)=-((B**3)*(A**2))/1.2D01
      Q(9,22)=-3.0D0*((A**3)*(B**2))/3.2D01
      Q(9,24)=-5.0D0*((B**4)*(A**2))/9.6D01
      Q(10,4)=2.0D0
      Q(10,7)=-3.0D0*A
      Q(10,8)=B
      Q(10,11)=3.0D0*A**2
      Q(10,12)=-3.0D0*(A*B)/2.0D0
      Q(10,13)=0.5D0*B**2
      Q(10,16)=-2.5D0*A**3
      Q(10,17)=-0.75D0*A*B**2
      Q(10,18)=0.25D0*B**3
      Q(10,20)=0.5D0*(B**2)*(A**2)
      Q(10,21)=(B**4)/8.0D0
      Q(10,22)=-3.0D0*(A*B**3)/8.0D0
      Q(10,23)=-5.0D0*((A**3)*(B**2))/1.2D01
      Q(10,24)=(B**5)/1.6D01
      Q(11,5)=1.0D0
      Q(11,8)=-A
      Q(11,9)=B
      Q(11,12)=0.75D0*A**2
      Q(11,13)=-A*B
      Q(11,14)=0.75D0*B**2
      Q(11,17)=0.75D0*B*A**2
      Q(11,18)=-0.75D0*A*B**2
```

```
Q(11,22)=9.0D0*((A**2)*(B**2))/1.6D01
Q(12,6)=2.0D0
Q(12,9)=-A
Q(12,10)=3.0D0*B
Q(12,13)=0.5D0*A**2
Q(12,14)=-1.5D0*A*B
Q(12,15)=3.0D0*B**2
Q(12,17)=-0.25D0*A**3
Q(12,18)=0.75D0*B*A**2
Q(12,19)=2.5D0*B**3
Q(12,20)=-(A**4)/8.0D0
Q(12,21)=-0.5D0*(B**2)*(A**2)
Q(12,22)=-3.0D0*(B*A**3)/8.0D0
Q(12,23)=(A**5)/1.6D01
Q(12,24)=-5.0D0*((A**2)*(B**3))/1.2D01
Q(13,1)=1.0D0
Q(13,2)=0.5D0*A
Q(13,3)=-0.5D0*B
Q(13,4)=0.25D0*A**2
Q(13,5)=-0.25D0*B*A
Q(13,6)=0.25D0*B**2
Q(13,7)=(A**3)/8.0D0
Q(13,8)=-(B*A**2)/8.0D0
Q(13,9)=(A*B**2)/8.0D0
Q(13,10)=-(B**3)/8.0D0
Q(13,11)=(A**4)/1.6D01
Q(13,12)=-(B*A**3)/1.6D01
Q(13,13)=((A**2)*(B**2))/1.6D01
Q(13,14)=-(A*B**3)/1.6DC1
Q(13,15)=(B**4)/1.6D01
Q(13,16)=(A**5)/3.2D01
Q(13,17)=((A**3)*(B**2))/3.2D01
Q(13,18)=-((A**2)*(B**3))/3.2D01
Q(13,19)=-((B**5))/3.2D01
Q(13,20)=-((A**4)*(B**2))/9.6D01
Q(13,21)=((B**4)*(A**2))/9.6D01
Q(13,22)=-((A**3)*(B**3))/6.4D01
Q(13,23)=-((A**5)*(B**2))/1.92D02
Q(13,24)=-((A**2)*(B**5))/1.92D02
Q(14,2)=1.0D0
Q(14,4)=A
Q(14,5)=-0.5D0*B
Q(14,7)=0.75D0*A**2
Q(14,8)=-0.5D0*A*B
Q(14,9)=0.25D0*B**2
Q(14,11)=0.5D0*A**3
Q(14,12)=-3.0D0*(B*A**2)/8.0D0
Q(14,13)=0.25D0*A*B**2
Q(14,14)=-(B**3)/8.0D0
Q(14,16)=5.0D0*(A**4)/1.6D01
Q(14,17)=3.0D0*((A**2)*(B**2))/1.6D01
Q(14,18)=-(B**3)*A/8.0D0
Q(14,20)=-((A**3)*(B**2))/1.2D01
Q(14,22)=-3.0D0*((A**2)*(B**3))/3.2D01
Q(14,23)=-5.0D0*((A**4)*(B**2))/9.6D01
Q(15,3)=1.0D0
Q(15,5)=0.5D0*A
Q(15,6)=-B
```

```
Q(15,8)=0.25D0*A**2
Q(15,9)=-0.5D0*A*B
Q(15,10)=0.75D0*B**2
Q(15,12)=(A**3)/8.0D0
Q(15,13)=-0.25D0*B*A**2
Q(15,14)=3.0D0*(A*B**2)/8.0D0
Q(15,15)=-0.5D0*B**3
Q(15,17)=-(B*A**3)/8.0D0
Q(15,18)=3.0D0*((A**2)*(B**2))/1.6D01
Q(15,19)=5.0D0*(B**4)/1.6D01
Q(15,21)=-((B**3)*(A**2))/1.2D01
Q(15,22)=3.0D0*((A**3)*(B**2))/3.2D01
Q(15,24)=5.0D0*((B**4)*(A**2))/9.6D01
Q(16,4)=2.0D0
Q(16,7)=3.0D0*A
Q(16,8)=-B
Q(16,11)=3.0D0*A**2
Q(16,12)=-1.5D0*A*B
Q(16,13)=0.5D0*B**2
Q(16,16)=2.5D0*A**3
Q(16,17)=0.75D0*A*B**2
Q(16,18)=-0.25D0*B**3
Q(16,20)=-0.5D0*(A**2)*(B**2)
Q(16,21)=-(B**4)/8.0D0
Q(16,22)=-3.0D0*(A*B**3)/8.0D0
Q(16,23)=-5.0D0*((A**3)*(B**2))/1.2D01
Q(16,24)=(B**5)/1.6D01
Q(17,5)=1.0D0
Q(17,8)=A
Q(17,9)=-B
Q(17,12)=0.75D0*A**2
Q(17,13)=-A*B
Q(17,14)=0.75D0*B**2
Q(17,17)=-0.75D0*B*A**2
Q(17,18)=0.75D0*A*B**2
Q(17,22)=9.0D0*((A**2)*(B**2))/1.6D01
Q(18,6)=2.0D0
Q(18,9)=A
Q(18,10)=-3.0D0*B
Q(18,13)=0.5D0*A**2
Q(18,14)=-1.5D0*A*B
Q(18,15)=3.0D0*B**2
Q(18,17)=0.25D0*A**3
 Q(18,18)=-0.75D0*B*A**2
Q(18,19)=-2.5D0*B**3
Q(18,20)=(A**4)/8.0D0
Q(18,21)=((B**2)*(A**2))/2.0D0
Q(18,22)=-3.0D0*(B*A**3)/8.0D0
Q(18,23)=(A**5)/1.6D01
Q(18,24)=-5.0D0*((B**3)*(A**2))/1.2D01
Q(19,1)=1.0D0
Q(19,2)=A/2.0D0
Q(19,3)=B/2.0D0
Q(19,4)=0.25D0*A**2
Q(19,5)=0.25D0*A*B
Q(19,6)=0.25D0*B**2
Q(19,7)=(A**3)/8.0D0
Q(19,8)=(B*A**2)/8.0D0
```

```
Q(19,9)=(A*B**2)/8.0D0
Q(19,10)=(B**3)/8.0D0
Q(19,11)=(A**4)/1.6D01
Q(19,12)=(B*A**3)/1.6D01
Q(19,13)=((A**2)*(B**2))/1.6D01
Q(19,14)=(A*B**3)/1.6D01
Q(19,15)=(B**4)/1.6D01
Q(19,16)=(A**5)/3.2D01
Q(19,17)= ((A**3)*(B**2))/3.2D01
Q(19,18)=((A**2)*(B**3))/3.2D01
Q(19,19)=(B**5)/3.2D01
Q(19,20)=((A**4)*(B**2))/9.6D01
Q(19,21)=((B**4)*(A**2))/9.6D01
Q(19,22)=((A**3)*(B**3))/6.4D01
Q(19,23)=((A**5)*(B**2))/1.92D02
Q(19,24)=((B**5)*(A**2))/1.92D02
Q(20,2)=1.0D0
Q(20,4)=A
Q(20,5)=B/2.0D0
Q(20,7)=0.75D0*A**2
Q(20,8)=0.5D0*A*B
Q(20,9)=0.25D0*B**2
Q(20,11)=0.5D0*A**3
Q(20,12)=3.0D0*(B*A**2)/8.0D0
Q(20,13)=(A*B**2)/4.0D0
Q(20,14)=(B**3)/8.0D0
Q(20,16)=5.0D0*(A**4)/1.6D01
    Q(20,17)=3.0D0*((A**2)*(B**2))/1.6D01
Q(20,18)=(A*B**3)/8.0D0
Q(20,20)=((A**3)*(B**2))/1.2D01
Q(20,22)=3.0D0*((A**2)*(B**3))/3.2D01
Q(20,23)=5.0D0*((A**4)*(B**2))/9.6D01
Q(21,3)=1.0D0
Q(21,5)=A/2.0D0
Q(21,6)=B
Q(21,8)=0.25D0*A**2
Q(21,9)=0.5D0*A*B
Q(21,10)=0.75D0*B**2
Q(21,12)=(A**3)/8.0D0
Q(21,13)=0.25D0*B*A**2
Q(21,14)=3.0D0*(A*B**2)/8.0D0
Q(21,15)=0.5D0*B**3
Q(21,17)=(B*A**3)/8.0D0
Q(21,18)=3.0D0*((A**2)*(B**2))/1.6D01
Q(21,19)=5.0D0*(B**4)/1.6D01
Q(21,21)=((B**3)*(A**2))/1.2D01
Q(21,22)=3.0D0*((A**3)*(B**2))/3.2D01
Q(21,24)=5.0D0*((A**2)*(B**4))/9.6D01
Q(22,4)=2.0D0
Q(22,7)=3.0D0*A
Q(22,8)=B
Q(22,11)=3.0D0*A**2
Q(22,12)=1.5D0*A*B
Q(22,13)=0.5D0*B**2
Q(22,16)=2.5D0*A**3
Q(22,17)=0.75D0*A*B**2
Q(22,18)=0.25D0*B**3
Q(22,20)=0.5D0*(A**2)*(B**2)
```

```
Q(22,21)=-(B**4)/8.0D0
Q(22,22)=3.0D0*(A*B**3)/8.0D0
Q(22,23)=5.0D0*((A**3)*(B**2))/1.2D01
Q(22,24)=-(B**5)/1.6D01
Q(23,5)=1.0D0
Q(23,8)=A
Q(23,9)=B
Q(23,12)=0.75D0*A**2
Q(23,13)=A*B
Q(23,14)=0.75D0*B**2
Q(23,17)=0.75D0*B*A**2
Q(23,18)=0.75D0*A*B**2
Q(23,22)=9.0D0*((A**2)*(B**2))/1.6D01
Q(24,6)=2.0D0
Q(24,9)=A
  Q(24,10)=3.0D0*B
Q(24,13)=0.5D0*A**2
Q(24,14)=1.5D0*A*B
Q(24,15)=3.0D0*B**2
Q(24,17)=0.25D0*A**3
Q(24,18)=0.75D0*B*A**2
Q(24,19)=2.5D0*B**3
Q(24,20)=-(A**4)/8.0D0
Q(24,21)=0.5D0*((A**2)*(B**2))
Q(24,22)=3.0D0*B*(A**3)/8.0D0
Q(24,23)=-(A**5)/1.6D01
Q(24,24)=5.0D0*((A**2)*(B**3))/1.2D01
RETURN
END
```

```
      SUBROUTINE BOCOND(N,TH,CHAN,IPOT,DD,ICOT,TAIL,ICOK,NC,
     *LUT,MUT,KUT,MEAT,NUL,NP)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DOUBLE PRECISION DSIN,DCOS
      DIMENSION CHAN(N),IPOT(N,2),DD(N,2),
     *TAIL(N,2),ICOT(N,3),ICOK(N,3),MEAT(NP),NUL(NP)
      L=1
   60 FORMAT(3I10)
      DO 31 M=1,N
      CHAN(M)=1.0
      DO 33 J=1,2
      DD(M,J)=1.0D0
      TAIL(M,J)=1.0D0
   33 IPOT(M,J)=0
      DO 34 JJ=1,3
      ICOT(M,JJ)=0
   34 ICOK(M,JJ)=0
   31 CONTINUE
      MUT=0
      KUT=0
      LUT=0
      DO 1 I=1,NP
      MMM=I-1
      IF(NUL(I).EQ.0) GO TO 1
      IF(NUL(I).EQ.1) GO TO 18
      DIK=-DCOS(TH)/DSIN(TH)
      GO TO 17
   18 IF(NUL(I).GT.2) GO TO 19
      DIK=(DSIN(TH)/DCOS(TH))
      GO TO 17
   19 CONTINUE
      GO TO 1
   17 MUT=MUT+1
      CHAN(L)=DIK*CHAN(L)
      IPOT(L,1)=MMM*6+2
      IPOT(L,2)=MMM*6+3
      WRITE(6,60) IPOT(L,1),IPOT(L,2)
      L=L+1
    1 CONTINUE
      KXK=1
      KXKX=1
      DO 3 I=1,NP
      IF(I.EQ.18) TH=-TH
      MMM=I-1
      IF(MEAT(I).EQ.0) GO TO 3
      IF(MEAT(I).GT.3) GO TO 51
      LUT=LUT+1
      ICOT(KXK,1)=MMM*6+4
      ICOT(KXK,2)=MMM*6+5
      ICOT(KXK,3)=MMM*6+6
      WRITE(6,60) ICOT(KXK,1),ICOT(KXK,2),ICOT(KXK,3)
      IF(MEAT(I).GT.1) GO TO 21
      DD(KXK,1)=-DCOS(TH)*DCOS(TH)/(DSIN(TH)*DSIN(TH))*DD(KXK,1)
      DD(KXK,2)=2.0D0*DCOS(TH)/DSIN(TH)*DD(KXK,2)
      KXK=KXK+1
      GO TO 3
   21 IF(MEAT(I).GT.2) GO TO 22
      DD(KXK,1)=1.0D0*DD(KXK,1)
```

```
      DD(KXK,2)=-((DSIN(TH)*DSIN(TH)-DCOS(TH)*DCOS(TH))/DSIN(TH)*DCOS(
     *TH))*DD(KXK,2)
      KXK=KXK+1
      GO TO 3
   22 IF(MEAT(I).GT.3) GO TO 51
      DD(KXK,1)=-(DSIN(TH)*DSIN(TH)/(DCOS(TH)*DCOS(TH)))*DD(KXK,1)
      DD(KXK,2)=(-2.0D0*DSIN(TH)/DCOS(TH))*DD(KXK,2)
      KXK=KXK+1
      GO TO 3
   51 ICOK(KXKX,1)=MMM*6+4
      ICOK(KXKX,2)=MMM*6+5
      ICOK(KXKX,3)=MMM*6+6
      WRITE(6,60) ICOK(KXKX,1),ICOK(KXKX,2),ICOK(KXKX,3)
      KUT=KUT+1
   23 IF(MEAT(I).GT.4) GO TO 24
      TAIL(KXKX,1)=DSIN(TH)/DCOS(TH)*TAIL(KXKX,1)
      TAIL(KXKX,2)=DSIN(TH)*DSIN(TH)/(DCOS(TH)*DCOS(TH))*TAIL(KXKX,2)
      KXKX=KXKX+1
      GO TO 3
   24 IF(MEAT(I).GT.5) GO TO 25
      TAIL(KXKX,1)=2.0D0*DSIN(TH)*DCOS(TH)/(DCOS(TH)*DCOS(TH)-DSIN(TH)
     **DSIN(TH))*TAIL(KXKX,1)
      TAIL(KXKX,2)=-1.0D0*TAIL(KXKX,2)
      KXKX=KXKX+1
      GO TO 3
   25 TAIL(KXKX,1)=-DCOS(TH)/DSIN(TH)*TAIL(KXKX,1)
      TAIL(KXKX,2)=DCOS(TH)*DCOS(TH)/(DSIN(TH)*DSIN(TH))*TAIL(KXKX,2)
      KXKX=KXKX+1
      GO TO 3
    2 CONTINUE
    3 CONTINUE
      WRITE(6,60) LUT
      WRITE(6,60) KUT
      WRITE(6,60) MUT
      RETURN
      END
```

```
      SUBROUTINE UNDEP(CLAP,IPOS,VAB,NP,NV,MRED,KSIZE,NKIMES,DD,KS)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION CLAP(KSIZE),VAB(KSIZE,KS),DD(NP,2),IPOS(NP,3)
      M=NKIMES
      DO 1 I=1,NV
      DO 11 N=1,MRED
   11 CLAP(N)=VAB(N,I)
      DO 23 LLL=1,NKIMES
      DO 2 JJ=1,MRED
      IF(M.GT.NP) GO TO 2
   10 IF(IPOS(M,3).GT.JJ) GO TO 2
       J=JJ+1
      DO 3 L=JJ,MRED
    3 VAB(L+1,I)=CLAP(L)
      VAB(JJ,I)=0.0D0
      IF(DD(M,2).EQ.0.0D0) GO TO 7
      VAB(JJ,I)=DD(M,2)*VAB(JJ-1,I)
    7 IF(DD(M,1).EQ.0.0D0) GO TO 9
      VAB(JJ,I)=DD(M,1)*VAB(JJ-2,I)+VAB(JJ,I)
    9 MRED=MRED+1
      DO 8 LL=1,MRED
    8 CLAP(LL)=VAB(LL,I)
      M=M-1
      GO TO 23
    2 CONTINUE
   23 CONTINUE
       M=NKIMES
      MRED=MRED-NKIMES
    1 CONTINUE
      MRED=MRED+NKIMES
      RETURN
      END
```

```
      SUBROUTINE UNLPR(CLAP,IPOS,VAB,NP,NV,MRED,KSIZE,NKIMES,TAIL,KS)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION CLAP(KSIZE),VAB(KSIZE,KS),TAIL(NP,2),IPOS(NP,3)
      M=NKIMES
      DO 1 I=1,NV
      DO 11 N=1,MRED
   11 CLAP(N)=VAB(N,I)
      DO 23 LLL=1,NKIMES
      DO 2 JJ=1,MRED
      IF(M.GT.NP) GO TO 2
   10 IF(IPOS(M,1).GT.JJ) GO TO 2
      J=JJ+1
      DO 3 L=J,MRED
    3 VAB(L+2,I)=CLAP(L)
      VAB(J+1,I)=0.0
      VAB(J,I)=0.0
      IF(TAIL(M,2).EQ.0.0D0) GO TO 7
      VAB(J+1,I)=(1.0D0/TAIL(M,2))*VAB(JJ,I)
    7 IF(TAIL(M,1).EQ.0.0D0) GO TO 9
      VAB(J,I)=(1.0D0/TAIL(M,1))*VAB(JJ,I)
    9 MRED=MRED+2
      DO 8 LL=1,MRED
    8 CLAP(LL)=VAB(LL,I)
      M=M-1
      GO TO 23
    2 CONTINUE
   23 CONTINUE
      M=NKIMES
      MRED=MRED-NKIMES*2
    1 CONTINUE
      MRED=MRED+NKIMES*2
      RETURN
      END
```

```
      SUBROUTINE UN(VAB,IOT,MSIZE,KS,KUTIME,NV,MRED,RTH,CLAP,NTIMES)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DOUBLE PRECISION DSIN,DCOS
      DIMENSION VAB(MSIZE,KS),IOT(KUTIME,3),CLAP(MSIZE)
      DO 1 I=1,NV
      DO 81 J=1,MRED
   81 CLAP(J)=VAB(J,I)
      DO 2 K=1,NTIMES
      LT=IOT(K,2)
      DO 3 KK=LT,MRED
    3 VAB((KK+2),I)=CLAP(KK)
      VAB(IOT(K,2),I)=-VAB(IOT(K,1),I)/DSIN(RTH)
      VAB(IOT(K,3),I)=VAB(IOT(K,1),I)/DCOS(RTH)
      MRED=MRED+2
      DO 17 M=1,MRED
   17 CLAP(M)=VAB(M,I)
    2 CONTINUE
      MRED=MRED-2*NTIMES
    1 CONTINUE
      MRED=MRED+2*NTIMES
      RETURN
      END
```

```
      SUBROUTINE IT(X,Y,NUL,MEAT,NGE,IPOS,IOS,MCN,X1,X2,X3,Y1,Y2,Y3,
     *IOT,BKO,C,      ICOK,IPOT,DD,TAIL,CHAN,ICOKE,ICOT,ICOTE,ISIZE,
     *IU,IC,IZ,IN,IH,NUM)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DOUBLE PRECISION DABS,DSQRT,DSIN,DCOS
      DIMENSION X(IZ),Y(IZ),NUL(IN),MEAT(IN),NGE(IZ,4),IOT(IU,3),
     *IPOS(IU,2),IOS(IU,2),MCN(IC),X1(IZ),X2(IZ),X3(IZ),Y1(IZ),Y2(IZ
     *Y3(IZ),ICOT(IU,3),ICOTE(IU,3),SM(24,24),SK(24,24),ICOKE(IU,2)
      DIMENSION BKO(ISIZE,ISIZE),C(ISIZE,ISIZE),ICOK(IU,2),IPOT(IU,2
     *DD(IU,2),TAIL(IU,2),CHAN(IU),NUM(IZ)
  700 FORMAT(I5)
  711 FORMAT(I5,F10.0)
      READ(5,754) NTIMES,NV
      READ(5,754) (IOT(I,1),I=1,NTIMES)
       READ(5,754) ((IOS(I,J),J=1,2),I=1,NTIMES)
       READ(5,754) (NUM(I),I=1,IH)
  754 FORMAT(16I5)
      READ(5,3) KROP1,KROP2,KUTIME
       READ(5,2) THR,THT,TH
 1001 FORMAT(2F15.0)
      READ(5,29) PNEUO,RHOO,YMODO,THICKO
      READ(5,29) PNEU,RHO,YMOD,THICK
   29 FORMAT(4F20.0)
      READ(5,1) NC,(MCN(I),I=1,NC)
      READ(5,3) NE,NP,KS
      READ(5,78) (NUL(I),I=1,NP)
      READ(5,78) (MEAT(I),I=1,NP)
      WRITE(6,78) (NUL(I),I=1,NP)
      WRITE(6,78) (MEAT(I),I=1,NP)
   78 FORMAT(16I5)
      KC=NC
    1 FORMAT(I5/(8I5))
      KE=NE
      KP=NP
   88 FORMAT(2I5)
      READ(5,2) (X1(I),I=1,NE)
      READ(5,2) (X2(I),I=1,NE)
      READ(5,2) (X3(I),I=1,NE)
      READ(5,2) (Y1(I),I=1,NE)
      READ(5,2) (Y2(I),I=1,NE)
      READ(5,2) (Y3(I),I=1,NE)
    2 FORMAT(8F10.0)
    3 FORMAT(3I5)
      MSIZE=NP*6
      READ(5,11) KSIZE,((NGE(I,J),J=1,4),I=1,NE)
  606 FORMAT('  TESRING')
   11 FORMAT(I5/(4I5))
      DO 4 I=1,KSIZE
      DO 4 J=1,KSIZE
    4 BKO(I,J)=0.0D0
      DO 593 I=1,NE
      X(I)=0.0D0
  593 Y(I)=0.0D0
      KE=NE
      NV=10
      TN=32.0D0
      FM=1.0D0
      FK=1.0D0
```

```
      DO 123 I=1,NE
      READ(5,6) X(I),Y(I)
  123 CONTINUE
    6 FORMAT(2F10.0)
      CALL BOCOND(KUTIME,TH,CHAN,IPOT,DD,ICOT,TAIL,ICOK,NC,LUT,MUT,KUT,
     *MEAT,NUL,NP)
      LU=1
      DO 5 L=1,NE
      TZ=DCOS(THT)
      TY=DSIN(THT)
      TA=DSIN(THR)
      TB=DCOS(THR)
      IF(L.EQ.NUM(LU)) GO TO 350
      TA=0.0D0
      TB=1.0D0
  350 CONTINUE
      IF(L.NE.NUM(LU)) GO TO 351
      LU=LU+1
  351 CONTINUE
      IF(L.EQ.11) KROP1=17
      IF(L.EQ.18) KROP1=32
      IF(L.LT.KROP1) GO TO 743
      CALL NRCD(X1(L),X2(L),X3(L),Y1(L),Y2(L),Y3(L),PNEU,SK,SM,TZ,TY)
      GO TO 744
  743 CALL UM6(X,Y,L,PNEU,SK,SM,TA,TB,NE)
  744 CALL TRNSFA(BKO,SM,SK,NGE,L,KE,KSIZE,24,6,4,3,KROP1        )
    5 CONTINUE
      CALL APPCON(BKO,MCN,MSIZE,NC,MRED,KSIZE)
      CALL REMCON(MCN,IOS,KSIZE,MSIZE,KC,NC,KUTIME,CHAN)
      CALL REMCON(MCN,IPOT,KSIZE,MSIZE,KC,NC,KUTIME,CHAN)
      CALL LPRCON(MCN,ICOK,KSIZE,MSIZE,KC,NC,KUTIME)
      CALL LPRCON(MCN,ICOT,KSIZE,MSIZE,KC,NC,KUTIME)
      CALL ARRANG(IPOT,ICOT,ICOK,IOS,LUT,KUT,MUT,NTIMES,NP,IPOS,ICOTE,
     *ICOKE,KUTIME)
      DO 717 KIL=1,NTIMES
      IOT(KIL,2)=IOS(KIL,1)
  717 IOT(KIL,3)=IOS(KIL,2)
      CALL LINDEP(BKO,MSIZE,MRED,CHAN,IPOT,MUT,KUTIME)
      CALL MULIND(BKO,MSIZE,MRED,DD,ICOT,LUT,KUTIME)
      CALL MULDEP(BKO,MSIZE,MRED,TAIL,ICOK,KUTIME,KUT)
      CALL TANT(BKO,KUTIME,MSIZE,IOT,NTIMES,RTH,MRED)
      WRITE(6,753) MRED
  753 FORMAT(I15)
      WRITE(8) (MCN(I),I=1,KC)
      DO 778 I=1,MRED
  778 WRITE(8) (BKO(I,J),J=1,MRED)
      RETURN
      END
```

```
      SUBROUTINE ARRANG(IPOT,ICOT,ICCK,IOS,LUT,KUT,MUT,NTIMES,NP,IPOS,
     *ICOTE,ICOKE,NZZ)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
       DIMENSION IPOT(NZZ,2),ICOT(NZZ,3),ICCK(NZZ,3),IPOS(NZZ,2),
     *IOS(NZZ,2),ICOKE(NZZ,3),ICOTE(NZZ,3)
      WRITE(6,203) ((IPOT(I,J),J=1,2),I=1,MUT)
      WRITE(6,203) ((ICOK(I,J),J=1,3),I=1,KUT)
      WRITE(6,203) ((ICOT(I,J),J=1,3),I=1,LUT)
      WRITE(6,203) ((IOS(I,J),J=1,2),I=1,NTIMES)
      IF(KUT.EQ.0) GO TO 101
      DO 1 I=1,KUT
      DO 1 J=1,3
    1 ICOKE(I,J)=ICOK(I,J)
  101 IF(LUT.EQ.0) GO TO 102
      DO 3 I=1,LUT
      DO 3 J=1,3
    3 ICOTE(I,J)=ICOT(I,J)
  102 IF(NTIMES.EQ.0) GO TO 103
      DO 4 I=1,NTIMES
      DO 4 J=1,2
    4 IPOS(I,J)=IOS(I,J)
  103 IF(MUT.EQ.0) GO TO 104
      DO 8 K=1,MUT
      L=IPOT(K,1)
      IF(LUT.EQ.0) GO TO 110
      DO 52 I=1,LUT
      DO 2 J=1,3
      IF(L.GT.ICOT(I,J)) GO TO 2
      DO 21 MM=J,3
   21 ICOTE(I,MM)=ICOTE(I,MM)-1
      GO TO 52
    2 CONTINUE
   52 CONTINUE
  110 IF(KUT.EQ.0) GO TO 111
      DO 54 I=1,KUT
      DO 5 J=1,3
      IF(L.GT.ICOK(I,J)) GO TO 5
      DO 20 MM=J,3
   20 ICOKE(I,MM)=ICOKE(I,MM)-1
      GO TO 54
    5 CONTINUE
   54 CONTINUE
  111 IF(NTIMES.EQ.0) GO TO 115
      DO 56 I=1,NTIMES
      DO 6 J=1,2
      IF(L.GT.IOS(I,J)) GO TO 6
      DO 22 MM=J,2
   22 IPOS(I,MM)=IPOS(I,MM)-1
      GO TO 56
    6 CONTINUE
   56 CONTINUE
  115 CONTINUE
      IF(K.EQ.1) GO TO 8
      KK=K-1
      DO 9 LT=1,2
    9 IPOT(K,LT)=IPOT(K,LT)-KK
    8 CONTINUE
  104 CONTINUE
```

```
      WRITE(6,203) ((IPOT(I,J),J=1,2),I=1,MUT)
      WRITE(6,203) ((ICOK(I,J),J=1,3),I=1,KUT)
      WRITE(6,203) ((ICOT(I,J),J=1,3),I=1,LUT)
      IF(LUT.EQ.0) GO TO 105
      DO 10 K=1,LUT
      L=ICOT(K,3)
      IF(KUT.EQ.0) GO TO 113
      DO 120 I=1,KUT
      DO 11 J=1,3
      IF(L.GT.ICOK(I,J)) GO TO 11
      DO 13 LL=J,3
   13 ICOKE(I,LL)=ICOKE(I,LL)-1
      GO TO 120
   11 CONTINUE
  120 CONTINUE
  113 IF(NTIMES.EQ.0) GO TO 114
   12 DO 17 I=1,NTIMES
      DO 16 J=1,2
      IF(L.GT.IOS(I,J)) GO TO 16
      DO 32 MM=J,2
   32 IPOS(I,MM)=IPOS(I,MM)-1
      GO TO 17
   16 CONTINUE
   17 CONTINUE
  114 CONTINUE
      IF(K.EQ.1) GO TO 10
      KKK=K-1
      DO 18 LT=1,3
   18 ICOTE(K,LT)=ICOTE(K,LT)-KKK
   10 CONTINUE
  105 CONTINUE
      LFUD=0
      IF(KUT.EQ.0) GO TO 106
      DO 47 K=1,KUT
      L=ICOK(K,2)
      IF(NTIMES.EQ.0) GO TO 116
      DO 44 I=1,NTIMES
      DO 40 J=1,2
      IF(L.GT.IOS(I,J)) GO TO 40
      DO 43 MM=J,2
   43 IPOS(I,MM)=IPOS(I,MM)-2
      GO TO 44
   40 CONTINUE
   44 CONTINUE
  116 CONTINUE
      IF(K.EQ.1) GO TO 47
      DO 49 LT=1,3
      ICOKE(K,LT)=ICOKE(K,LT)-K-LFUD
   49 CONTINUE
      LFUD=LFUD+1
   47 CONTINUE
  106 CONTINUE
      WRITE(6,203) ((IPOT(I,J),J=1,2),I=1,MUT)
      WRITE(6,203) ((ICOK(I,J),J=1,3),I=1,KUT)
      WRITE(6,203) ((ICOT(I,J),J=1,3),I=1,LUT)
      IF(NTIMES.EQ.0) GO TO 107
      DO 61 I=1,NTIMES
      DO 61 J=1,2
```

```
 61 IOS(I,J)=IPOS(I,J)
    DO 90 JJJ=1,NTIMES
    LOR=IOS(JJJ,1)
    JO=JJJ+1
    IF(JO.GT.NTIMES) GO TO 90
    DO 91 KKK=JO,NTIMES
    IOS(JO,1)=IOS(JO,1)-2
    IOS(JO,2)=IOS(JO,2)-2
 91 CONTINUE
 90 CONTINUE
107 IF(KUT.EQ.0) GO TO 108
    DO 25 I=1,KUT
    DO 25 J=1,3
 25 ICOK(I,J)=ICOKE(I,J)
108 IF(LUT.EQ.0) GO TO 109
    DO 19 I=1,LUT
    DO 19 J=1,3
 19 ICOT(I,J)=ICOTE(I,J)
109 CONTINUE
    WRITE(6,203) ((IPOT(I,J),J=1,2),I=1,MUT)
    WRITE(6,203) ((ICOK(I,J),J=1,3),I=1,KUT)
    WRITE(6,203) ((ICOT(I,J),J=1,3),I=1,LUT)
    WRITE(6,203) ((IOS(I,J),J=1,2),I=1,NTIMES)
203 FORMAT('    ICOK',8I5)
    RETURN
    END
```

```fortran
      SUBROUTINE LPRCON(MCN,IOST,KSIZE,NSIZE,KC,NC,NTIMES)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION IOST(NTIMES,3)
      DIMENSION MCN(KC)
      J=1
      K=1
      L=1
      M=1
      N=1
      DO 21 NUM=1,NSIZE
      IF(L.GT.KC) GO TO 7
      IF(MCN(L).EQ.NUM) GO TO 2
    7 IF(K.GT.NTIMES) GO TO 11
      IF(IOST(K,3).NE.NUM) GO TO 11
      IOST(K,3)=J
      K=K+1
      GO TO 24
   11 IF(M.GT.NTIMES)GO TO 12
    5 IF(IOST(M,2).NE.NUM) GO TO 12
      IOST(M,2)=J
      M=M+1
      GO TO 24
   12 IF(N.GT.NTIMES) GO TO 21
    6 IF(IOST(N,1).NE.NUM) GO TO 24
      IOST(N,1)=J
      N=N+1
      GO TO 24
    2 L=L+1
      GO TO 21
   24 J=J+1
   21 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULIND(BKO,NSIZE,KSIZE,DD,ICOT,NTIMES,NKLIME)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION BKO(NSIZE,NSIZE),DD(NKLIME,2),
     *    ICOT(NKLIME,3),RLEC(2)
      DO 2 M=1,NTIMES
      IF(NTIMES.EQ.0) GO TO 2
      RLEC(1)= DD(M,1)
      RLEC(2)= DD(M,2)
      NR=ICOT(M,1)
      NS=ICOT(M,2)
      NT=ICOT(M,3)
      DO 50 I=1,KSIZE
      BKO(NR,I)=BKO(NR,I)+RLEC(1)*BKO(NT,I)
      BKO(NS,I)=BKO(NS,I)+RLEC(2)*BKO(NT,I)
      IF( NS-KSIZE ) 5,50,50
    5 LQ=KSIZE-1
      DO 1 J=NT,LQ
    1 BKO(J,I)=BKO(J+1,I)
   50 CONTINUE
      KQ=KSIZE-1
      DO 51 I=1,KSIZE
      BKO(I,NR)=BKO(I,NR)+RLEC(1)*BKO(I,NT)
      BKO(I,NS)=BKO(I,NS)+RLEC(2)*BKO(I,NT)
      IF(NS-KSIZE) 6,51,51
    6 DO 4 J=NT,KQ
    4 BKO(I,J)=BKO(I,J+1)
   51 CONTINUE
      KSIZE=KSIZE-1
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MULDEP(BKO,NSIZE,KSIZE,TAIL,ICOK,NP,NKIMES)
       IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DIMENSION BKO(NSIZE,NSIZE),TAIL(NP,2),ICOK(NP,3)
     *,RFAC(2)
      DO 2 M=1,NKIMES
      IF(NKIMES.EQ.0) GO TO 2
      RFAC(1)=TAIL(M,1)
      RFAC(2)=TAIL(M,2)
      NR=ICOK(M,1)
      NS=ICOK(M,2)
      NT=ICOK(M,3)
      KIL=KSIZE-2
      DO 50 I=1,KSIZE
      BKO(NR,I)=BKO(NR,I)+RFAC(1)*BKO(NS,I)
      BKO(NR,I)=BKO(NR,I)+RFAC(2)*BKO(NT,I)
      IF((KSIZE-NS).LT.0) GO TO 50
      DO 1 J=NS,KIL
    1 BKO(J,I)=BKO(J+2,I)
   50 CONTINUE
      DO 51 I=1,KSIZE
      BKO(I,NR)=BKO(I,NR)+RFAC(1)*BKO(I,NS)
      BKO(I,NR)=BKO(I,NR)+RFAC(2)*BKO(I,NT)
      IF(NS-KSIZE) 9,51,51
    9 DO 12 II=NS,KIL
      BKO(I,II)=BKO(I,II+2)
   12 CONTINUE
   51 CONTINUE
      KSIZE=KSIZE-2
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE TANT(BKO,KUTIME,MSIZE,IOT,NTIMES,RTH,MRED)
      IMPLICIT REAL*8(A-H,O-Z),INTEGER(I-N)
      DOUBLE PRECISICN DSIN,DCOS
      DIMENSION BKO(MSIZE,MSIZE),IOT(KUTIME,3)
      DO 1 I=1,NTIMES
      DO 2 K=1,MRED
    2 BKO(K,IOT(I,1))=-DSIN(RTH)*BKO(K,IOT(I,2))+DCOS(RTH)*BKO(K,IOT
     *(I,3))+BKO(K,IOT(I,1))
      LOT=MRED-2
      LL=IOT(I,2)
      DO 3 J=1,MRED
      DO 3 L=LL,LOT
    3 BKO(J,L)=BKO(J,L+2)
      DO 4 K=1,MRED
    4 BKO(IOT(I,1),K)=-DSIN(RTH)*BKO(IOT(I,2),K)+DCOS(RTH)*BKO(IOT(I,3
     *),K)+BKO(IOT(I,1),K)
      DO 5 J=1,MRED
      DO 5 L=LL,LOT
    5 BKO(L,J)=BKO(L+2,J)
      MRED=MRED-2
    1 CONTINUE
      RETURN
      END
```

SECTION 2

In general, the boundaries of a system are not parallel to the global co-ordinate axes. In such cases it is convenient to use local (ξ,η) axes and to transform subsequently to the (x,y) co-ordinate system shown in Figure 11. The trigonometric relationships (together with the necessary computer coding) required for the transformation of various boundary constraints from one co-ordinate to the other are given in this section.

The computer coding dictates that some value be assigned to NUL and MEAT for each global nodal point. NUL and MEAT are set zero if no oblique slope or second derivative boundary constraint is to be applied. Otherwise, NUL and MEAT are given values dictated by the following table.

BOUNDARY CONSTRAINT	TRIGONOMETRIC RELATION	REQUIRED CONSTANT	CODE
1. Slopes			
$w_\xi=0$	$w_x=-w_y\dfrac{\cos\theta}{\sin\theta}$	CHAN(I)= $-\dfrac{\cos\theta}{\sin\theta}$	NUL=2
$w_\eta=0$	$w_x=w_y\dfrac{\sin\theta}{\cos\theta}$	CHAN(I)= $\dfrac{\sin\theta}{\cos\theta}$	NUL=1
2. Second Derivatives			
$w_{\xi\xi}=0$	$w_{yy}=-w_{xy}2\dfrac{\sin\theta}{\cos\theta}-w_{xx}\tan^2\theta$	DD(I,1)=$-\tan^2\theta$ DD(I,2)=$-2\dfrac{\sin\theta}{\cos\theta}$	MEAT=3
$w_{\xi\eta}=0$	$w_{yy}=-\left(\dfrac{\sin^2\theta-\cos^2\theta}{\sin\theta\cos\theta}\right)w_{xy}+w_{xx}$	DD(I,1)=1.0 DD(I,2)= $-\dfrac{\sin^2\theta-\cos^2\theta}{\sin\theta\cos\theta}$	MEAT=2
$w_{\eta\eta}=0$	$w_{yy}=-\dfrac{1.0}{\tan^2\theta}w_{xx}+2\dfrac{\cos\theta}{\sin\theta}w_{xy}$	DD(I,1)=$-\dfrac{1.0}{\tan^2\theta}$ DD(I,2)=$2\dfrac{\cos\theta}{\sin\theta}$	MEAT=1
$w_{\xi\xi}=0$ $w_{\xi\eta}=0$	$w_{xx}=-\dfrac{\cos\theta}{\sin\theta}w_{xy}$ $w_{xx}=\dfrac{\cos^2\theta}{\sin^2\theta}w_{yy}$	TAIL(I,1)= $-\dfrac{\cos\theta}{\sin\theta}$ TAIL(I,2)= $\dfrac{\cos^2\theta}{\sin^2\theta}$	MEAT=6
$w_{\eta\eta}=0$ $w_{\xi\xi}=0$	$w_{xx}=-w_{yy}$ $w_{xx}=2\dfrac{\sin\theta\cos\theta}{\cos^2\theta-\sin^2\theta}w_{xy}$	TAIL(I,1)= $2\dfrac{\sin\theta\cos\theta}{\cos^2\theta-\sin^2\theta}$ TAIL(I,2)= 1.0	MEAT=5
$w_{\eta\eta}=0$ $w_{\xi\eta}=0$	$w_{xx}=\dfrac{\sin\theta}{\cos\theta}w_{xy}$ $w_{xx}=\dfrac{\sin^2\theta}{\cos^2\theta}w_{yy}$	TAIL(I,1)= $\dfrac{\sin\theta}{\cos\theta}$ TAIL(I,2)= $\dfrac{\sin^2\theta}{\cos^2\theta}$	MEAT=4

Table 7.