ANALYSIS OF LARGE TANKERS USING GRILLAGE AND FINITE ELEMENT METHODS

Edward A. Chazal, Jr.
Jerry Marsh Payne

A dissertation submitted in partial fulfillment of the requirements for Professional Degrees of Naval Architect in The University of Michigan,
1971

Professional Degree Committee:

Professor Finn C. Michelsen, Chairman
Professor Richard B. Couch
Mister Arthur M. Reed

ABSTRACT

Two of the most important contributions to ship structure analysis have been the finite element method and the grillage method. Each has made its own important contribution, yet neither has been able to offer itself as a satisfactory design tool. The grillage method lacks the detailed stress patterns needed by the designer, and the finite element method has shown itself to be impractical when used to analyze a model as large as a ship hull. There is an obvious solution to this problem, for the weaknesses of one method are the strong points of the other. By using each method where it is best suited, the analysis can be performed with the necessary detail and without great expense.

This thesis presents a method developed by Dr. Pin-Yu Chang, ComCode Corporation, which combines the simplicity of the overall analysis using a grillage model and the fine analysis of the finite element technique for critical portions of the structure. Some results using this method are compared here to full scale test data. Computer programs for a longitudinal and a transverse analysis are also included.

ACKNOWLEDGEMENTS

All authors of technical dissertations draw from resources of talent quite independent of their own. This paper is no exception. Many persons have contributed substantially to its completion. It will be impossible to name all the contributors but they have not been forgotten.

The authors want to thank the faculty members of the Department of Naval Architecture and Marine Engineering and the Department of Engineering Mechanics, The University of Michigan. The knowledge and ability of the authors to conduct such a project as this is due largely to the patient efforts of various faculty members.

The authors received guidance and clarification many times from Dr. Pin-Yu Chang, the originator of the method presented in this paper. Particular thanks for their assistance must also go to Professor Raymond A. Yagle and LCDR C. S. Loosemore, USCG who were the sources for several obscure references. Professor Richard B. Couch and Mr. Arthur Reed of the thesis committee provide valuable advice in the practical aspects of the problem and computer programming respectively.

The authors are deeply grateful to Professor Finn C. Michelsen, the chairman of the thesis committee, who has provided us with the guidance and theoretical insight to approach the problem. Professor Michelsen has the ability to couple a very physical approach to problems with the mathematics required to effectively model those problems. He has been counselor, teacher, and the single most important influence in our academic endeavors.

Finally there are those who contributed through their administrative or supportive roles. Professor T. Francis Ogilvie has been our graduate program advisor and his enthusiasm and support were a real foundation for our efforts. The beauty and accuracy of the manuscript testify to the fine ability of the typist, Chris Seidl. To our wives Susan Chazal and Donna Payne who by their patience, understanding, proof reading skills, and advice contributed substantially in this project.

April, 1971
Ann Arbor, Michigan

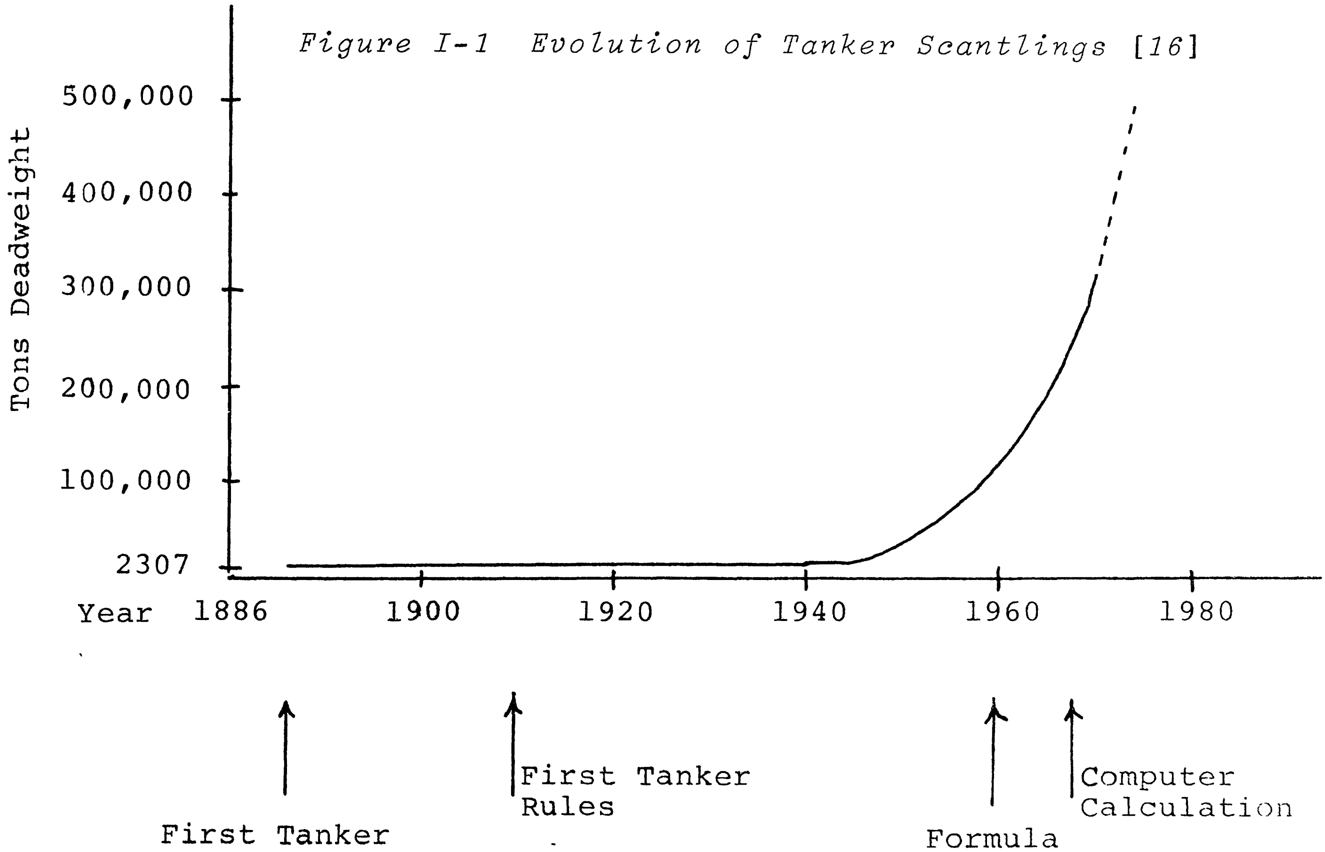
Edward A. Chazal, Jr.
Jerry M. Payne

TABLE OF CONTENTS

INTRODUCTION

The past decade has ushered in several numerical methods for achieving solutions to the ship structure analysis problem. Prior to this time most analyses were conducted on the basis of painstakingly accumulated empirical data. The empirical formulations produced satisfactory results for the first half of the twentieth century. In the late 1950's economic advantages produced a whole new size generation of tankers [16].

Figure I-1 Evolution of Tanker Scantlings [16]



These ships were much larger than any prior vessels. They were so much larger that given hindsight, it can be seen that prior empirical analyses should not have been expected to produce accurate results. This is the case for the early jumbo tankers which exhibit a history of structural failures in the vicinity of the intersections of prime longitudinals and the transverse members. The failures showed quite often the characteristic features of deformation under excessive shear loads. We can see that these shear loads must be treated as an important factor [4].

These failures and the general lack of empirical data suggest a need for a more satisfactory design procedure. The basic nature of the problem suggests difficulty in ever achieving a completely accurate formulation. In order to attack the problem one must be cognizant of the many static and dynamic loadings imposed on the vessel. In addition there must be a satisfactory method for analyzing the structure itself.

There have been several advances in structural analysis that would appear to lend themselves well to the problem solution. Two prominent achievements have been the very elegant theory of grillages [5,11,20] and the versatile finite element method [6,7,10,12,17,22].

The finite element method has been used effectively for a number of years by civil and aeronautical engineers. The grillage theory was developed principally by naval architects. There are advantages and disadvantages to both methods. In order to be applied effectively and efficiently they both need a reasonable amount of skill or understanding of the art.

There are several published examples of applications of both techniques to the ship structure problem. Generally these are all limited to a particular area of interest. It is no accident that the particular areas chosen lend themselves quite well to solutions by the method selected. It seems reasonable to point out that the investigators selected the most efficient tool for solving the problem at hand.

In dealing with the whole ship analysis the complex nature of the problem demands even greater skill [10] (see Figure I-2). For some time the finite element method alone seemed to offer the best route to eventual success. The efforts of Kamel et al, at the University of Arizona, achieved reasonable results via this route with the DAISY program [6,7]. As published, this analysis makes use of a macro mesh to reduce round off error and computer expenses. The macro mesh solutions are then applied to

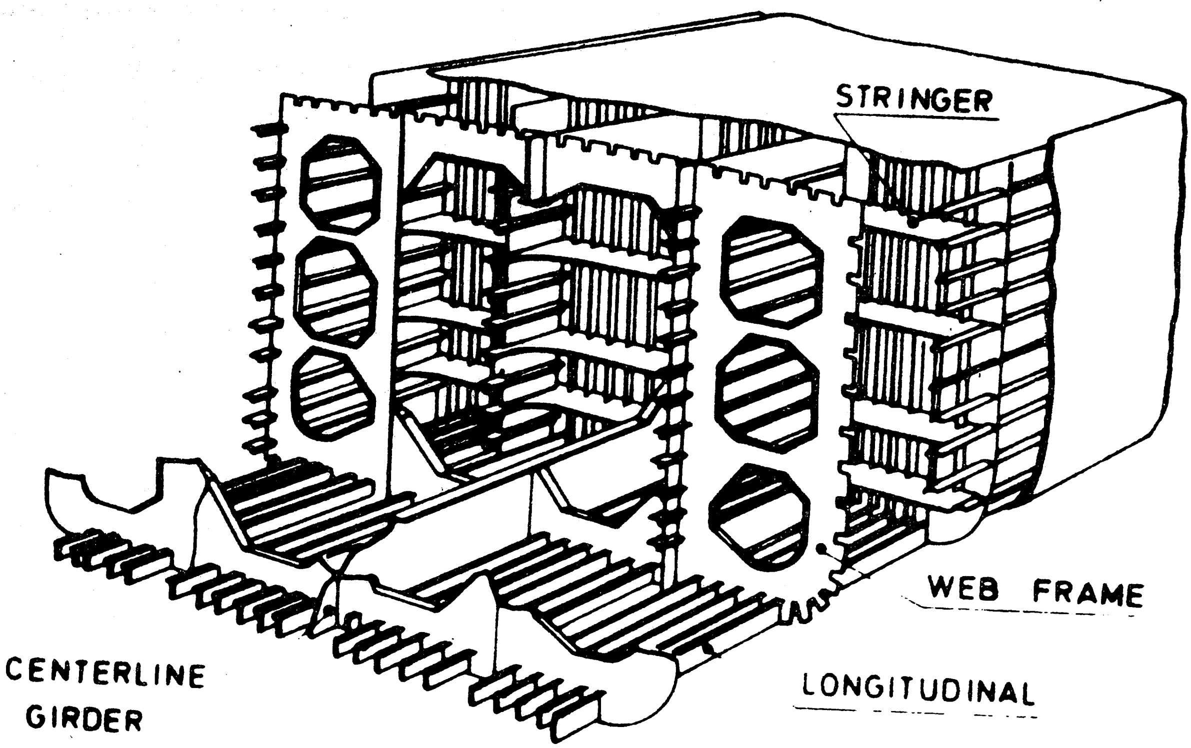


Figure I-2 Structure Detail of Typical Tanker [10]

a micro mesh analysis of the area in question. The actual cost of analysis for a typical ship using DAISY has not been published. Roberts commented on a similar effort in Great Britain that required 24 hours of computer time and several months of data preparation. As a design tool this type of approach is very expensive but it does give results. It is a complete analysis tool which solves the entire ship problem but it forces the finite element method to its limits. As a result, certain compromises are made to reduce computation time but they also reduce accuracy.

At this stage it seems entirely reasonable to seek methods that will allow the finite element technique optimum utilization. Extending the use of this method blindly to a point at which its results are questionable is merely a misuse of a good tool. A structural analyst, to be successful, must use all available information to his advantage. If he allows himself to stop thinking and merely feeds numbers into a computer, he has removed the benefit of his knowledge and judgement from the solution of the problem. St. Denis has suggested that the ship structure problem be solved by a judicious combination of many methods of analysis. The final tool he envisioned would make the best use of each technique and really would be no more than a synthesis of available methods each used to its best advantage [6].

It is against this background that Dr. Pin-Yu Chang formulated his approach to the entire ship structure problem. The total problem has not yet been solved and much work remains to be done. However Dr. Chang's effort clearly leads the way to a complete rational analysis. This analysis makes maximum use of the skill of the naval architect and should be more efficient than the brute force application of one particular method.

The technique uses grillage analysis for the overall ship problem and finite element methods for the local analyses. The employment of the grillage greatly reduces the computation time while the finite element methods allow a detailed analysis of stress patterns. The problem has been divided into two principal parts, the transverse and longitudinal strength analyses. The transverse problem depends upon certain output from the longitudinal problem.

The longitudinal analysis treats the hull as a grillage of simple shear beams. The result is the set of interaction forces between the members which are used in the transverse analysis as shear forces between the primary members of the structure. These shear forces along with the external loads complete the loading pattern which can be used to compute stresses and bending moments in the primary structural members.

The transverse analysis uses the grillage properties of the hull to uncouple the governing differential equations by means of coordinate transformation. This reduces the transverse analysis to a simple two dimensional problem which can be solved using finite elements. This reduces necessary computational time and avoids the use of finite elements in the macro analysis which seems to have questionable accuracy [4].

LONGITUDINAL STRENGTH ANALYSIS

For longitudinal strength consideration, the ship is treated as a grillage consisting of four longitudinal members and several transverse stiffeners (see Figure II-1). The prime longitudinal members are the side shells and two longitudinal bulkheads. The transverse members are bulkheads and web frames which act as stiffeners. Both longitudinal and transverse members include portions of the bottom and deck as flanges. This insures that the total moment of inertia of the model will be the same as that derived in the conventional manner. It is assumed that each member of the grillage behaves as a simple shear beam. This assumption has been verified by Vasta for medium size ships and there is no evidence that would invalidate it for large tankers [19]. The transverse beams are free at both ends and the longitudinals are simply supported. Since the external loads on the grillage are self balanced, the shear forces at the simply supported ends are nearly zero. Then the longitudinals are equivalent to free-free beams.

The external loads acting on the plate are transmitted to the longitudinals and then transferred to the transverses. The load is then distributed as concentrated forces on the transverse members. This loading transfer pattern emphasizes the importance of the shear forces at the intersections of the prime members. The primary deflections of the longitudinal members are computed by distributing the loads uniformly along a longitudinal between transverses.

Consider a particular (α-th) transverse supported by longitudinals and acted upon by a symmetrical loading system. In the following figure, the reactions R_1 and R_2 represent the shear forces on the longitudinals. Since the beam and the loading pattern are symmetrical, it is only necessary to consider half of the beam.

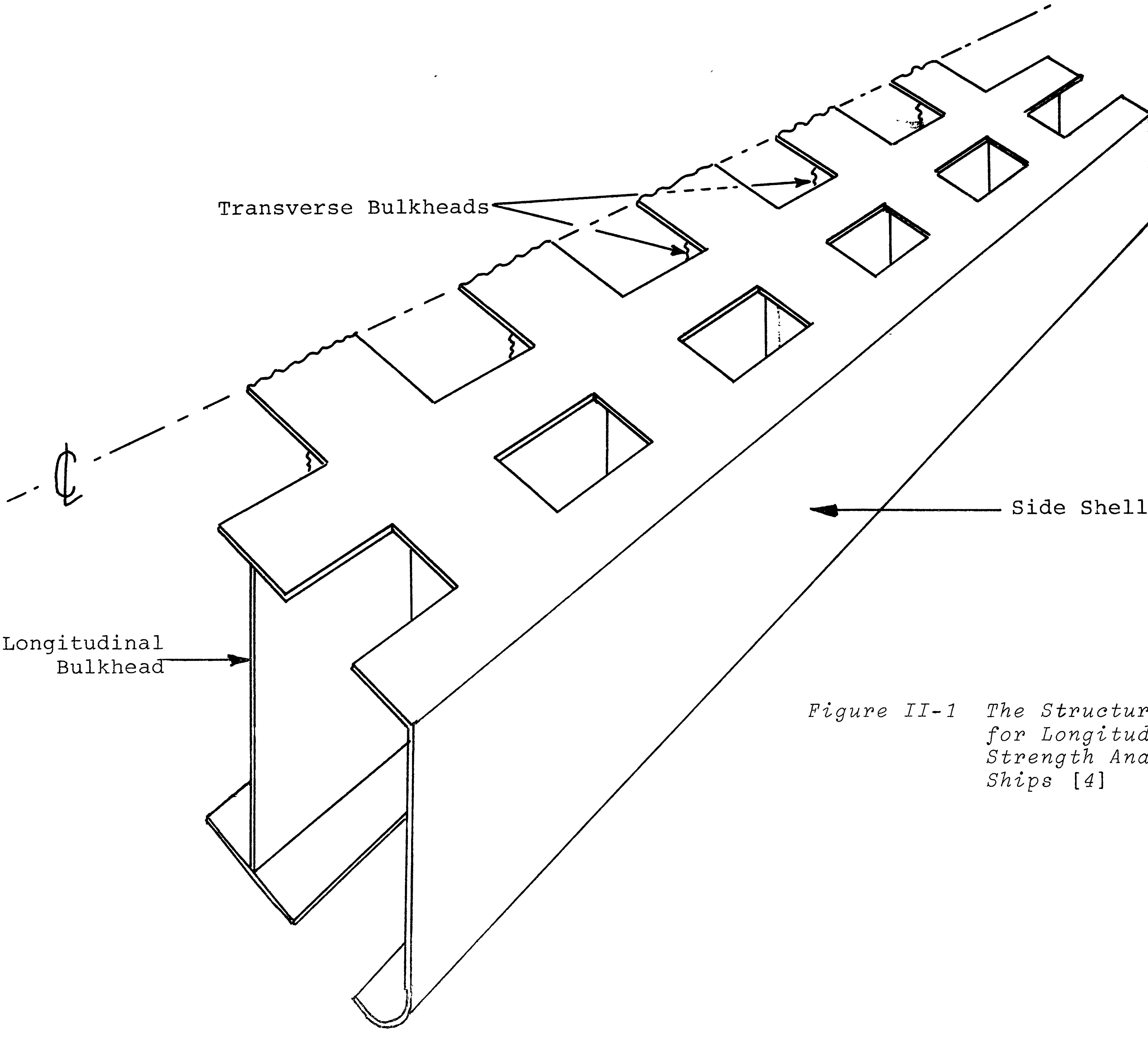


Figure II-1 The Structural Model for Longitudinal Strength Analysis of Ships [4]

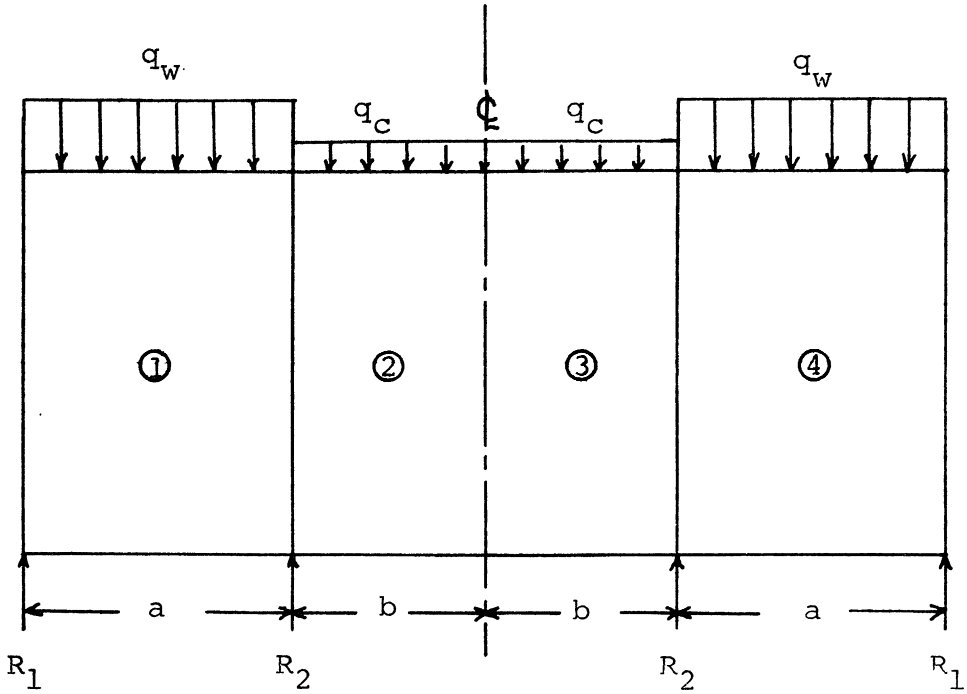


Figure II-2 Transverse Modeled as a Short Deep Beam with Symmetrical Loading [4]

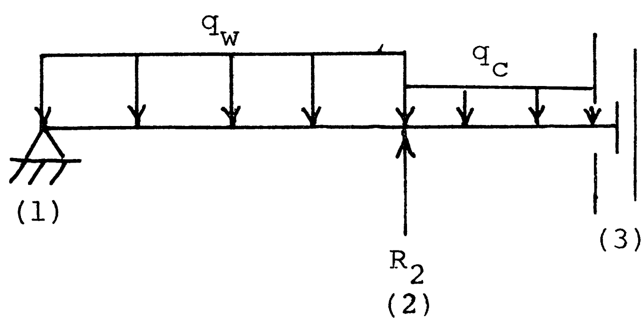


Figure II-3 Model of Transverse Taking Advantage of Symmetry [4]

The load in the wing tank (q_w) and the load in the central tank (q_c) have been here assumed to be uniform. Since it will be necessary only to find the relative displacements along this member, the left end has been simply supported. The relative displacement between points (1) and (2) of the α-th transverse is Δd^α . Then

$$\Delta d^\alpha = W_2^\alpha - k_{22}^\alpha R_2^\alpha \tag{II-1}$$

where W_2^α is the displacement of point (2) due to the external loads on the beam and k_{22}^α is an influence coefficient.

The influence coefficients (k_{ij}^{α}) of the α-th transverse are a set of deflections at x_i due to a unit load at x_j . Let S^n be a vector of state variables at station n such that

$$S^n = \begin{bmatrix} W \\ \theta \\ M \\ V \end{bmatrix}_n \qquad \text{(II-2)}$$

where W is displacement, θ is rotation, M is bending moment and V is shear force. Then by line solution

$$\begin{bmatrix} W \\ \theta \\ M \\ V \\ 1 \end{bmatrix}_2 = \begin{bmatrix} 1 & -a & \frac{-a^2}{2EI_1} & \frac{-a^3}{6EI_1} + \frac{a}{GA_1} & q_w \frac{a^4}{24EI_1} - \frac{a^2}{2GA_1} \\ 0 & 1 & \frac{a}{EI_1} & \frac{a^2}{2EI_1} & \frac{-q_w a^3}{6EI_1} \\ 0 & 0 & 1 & a & \frac{-q_w a^2}{2} \\ 0 & 0 & 0 & 1 & -q_w a \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} W \\ \theta \\ M \\ V \\ 1 \end{bmatrix}_1$$

where I_1 is the moment of inertia of section (1) of the beam, A_1 is the shear area and E , G , are constants of elasticity. This expression may for convenience be written

$$S^2 = L_1 S^1 \qquad \text{(II-4)}$$

Similarly

$$S^3 = L_2 S^2 \qquad \text{(II-5)}$$

Combining the above equations

$$S^3 = L_2 L_1 S^1 \qquad \text{(II-6)}$$

The boundary conditions for the beam being considered are

$$W^1 = M^1 = \theta^3 = V^3 = 0 \qquad \text{(II-7)}$$

Letting $L^n = L_n L_{n-1} \ldots L_1$, (II-6) can be rewritten as

$$S^3 = L^2 S^1 \qquad \text{(II-8)}$$

Let L^n_{ij} be a particular element in the L^n matrix. Using (II-7) in (II-6), the deflections at point two and three are found to be

$$W^2 = L^1_{12}\ \theta^1 + L^1_{14}\ V^1 + L^1_{15} \qquad \text{(II-9)}$$

$$W^3 = L^2_{12}\ \theta^1 + L^2_{14}\ V^1 + L^2_{15} \qquad \text{(II-10)}$$

in terms of the initial parameters θ^1 and V^1 . These parameters are also obtainable from the same equations since

$$L^2_{22}\ \theta^1 + L^2_{24}\ V^1 + L^2_{25} = 0$$

$$L^2_{42}\ \theta^1 + L^2_{44}\ V^1 + L^2_{45} = 0$$

or

$$\theta^1 = \frac{L^2_{25}\ L^2_{44} - L^2_{24}\ L^2_{45}}{L^2_{22}\ L^2_{44} - L^2_{24}\ L^2_{42}} \qquad \text{(II-11)}$$

$$V^1 = \frac{L^2_{22}\ L^2_{45} - L^2_{25}\ L^2_{42}}{L^2_{22}\ L^2_{44} - L^2_{24}\ L^2_{42}} \qquad \text{(II-12)}$$

The displacement W^α_2 for the α-th transverse is the displacement which is needed in equation (II-1).

In order to find influence coefficients by this line solution method, the uniform loads (q_w and q_c) must be set equal to zero in the transfer matrices L^1 and L^2. In addition a point matrix is added at the location of the required unit load. The point matrix is

$$L^p = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

Then $S^i = L^{i-1} S^1$

where $L^{i-1} = L_{i-1} L_{i-2} \cdots L_{j+1} L^p L_j \cdots L_1$ (II-13)

Then the displacement W^i calculated from (II-10) is the influence coefficient

$$k^\alpha_{ij} = L^{i-1}_{12} \theta^1 + \left(L^{i-1}_{14} V + L^{i-1}_{15} \right)_{\alpha j} \tag{II-14}$$

The unknown reaction R^α_2 can now be expressed in terms of these values by rewriting equation (II-1):

$$R^\alpha_2 = \frac{1}{k^\alpha_{22}} W^\alpha_2 - \Delta d^\alpha \tag{II-15}$$

Since q_w and q_c are uniform loads, then R^α_1 can be expressed as

$$R^\alpha_1 = aq^\alpha_c + bq^\alpha_c - R^\alpha_2 \tag{II-16}$$

Now except for Δd^α all values are known and the reactions can be calculated.

Treating the four prime longitudinals as independent simple beams which are simply supported and distributing the external loads on these members, a set of displacements may be obtained from simple beam theory for each member. The longitudinals here are considered not to be supported by the transverse members. The deflection, d_m^α, is that of the m-th longitudinal at the location where it should be supported by the α-th transverse. Due to symmetry it is necessary to calculate only the deflections of one side shell (d_1^α) and a longitudinal bulkhead (d_2^α). Now Δd^α can be computed as

$$\Delta d^\alpha = d_2^\alpha - d_1^\alpha$$

and the reactions R_m^α can be found.

Once these reactions between the grillage members have been found the external loading system is complete. By methods of elasticity, bending moments and shear stresses can be computed at points of interest.

TRANSVERSE STRENGTH ANALYSIS

The method of transverse analysis presented here is similar to other three-dimensional finite element solutions, except that a much finer mesh is used. This fine mesh can be adopted without increasing computer cost because uncoupling techniques used in the overall analysis substantially reduce the amount of required computer time. The basic effect of the uncoupling is that it reduces the three dimensional system mathematically into a set of two dimensional equivalent transverse members. Each of these members can then be analyzed in two dimensions.

The model used for this part of the analysis is a three dimensional body (figure III-1). The longitudinal members (side shell, bulkhead, deck, and bottom) are represented by bars and the transverse members are represented by plates which are reinforced by bars. Due to symmetry about the ship's center plane the model represents only half of the hull. The transverses are restrained from moving in the horizontal direction along their intersection with the center plane. In the analysis of an individual transverse, the section is restrained from moving vertically at the intersection of the longitudinal bulkhead and the bottom (figure III-2). The longitudinals are simple supported at both ends. It is assumed that all longitudinals are similar beams and that all transverses are of proportional stiffness.

External loads are transmitted from the plate to the longitudinals and then transferred to the transverses. The external loads can be expressed as a function of z which is the distance from the forward perpendicular in a direction perpendicular to the planes of the transverses. Then the α-th longitudinal is acted upon by $q_{\alpha}(z)$ and the horizontal and vertical reactions of the i-th transverse, $X_{i\alpha}$, $Y_{i\alpha}$. Let $d_{xi\alpha}$ be the deflection caused by $q_{x\alpha}(z)$ (in the x direction) and let $A_{ij}^{\alpha x}$ be the influence coefficients of the α-th longitudinal between z_i

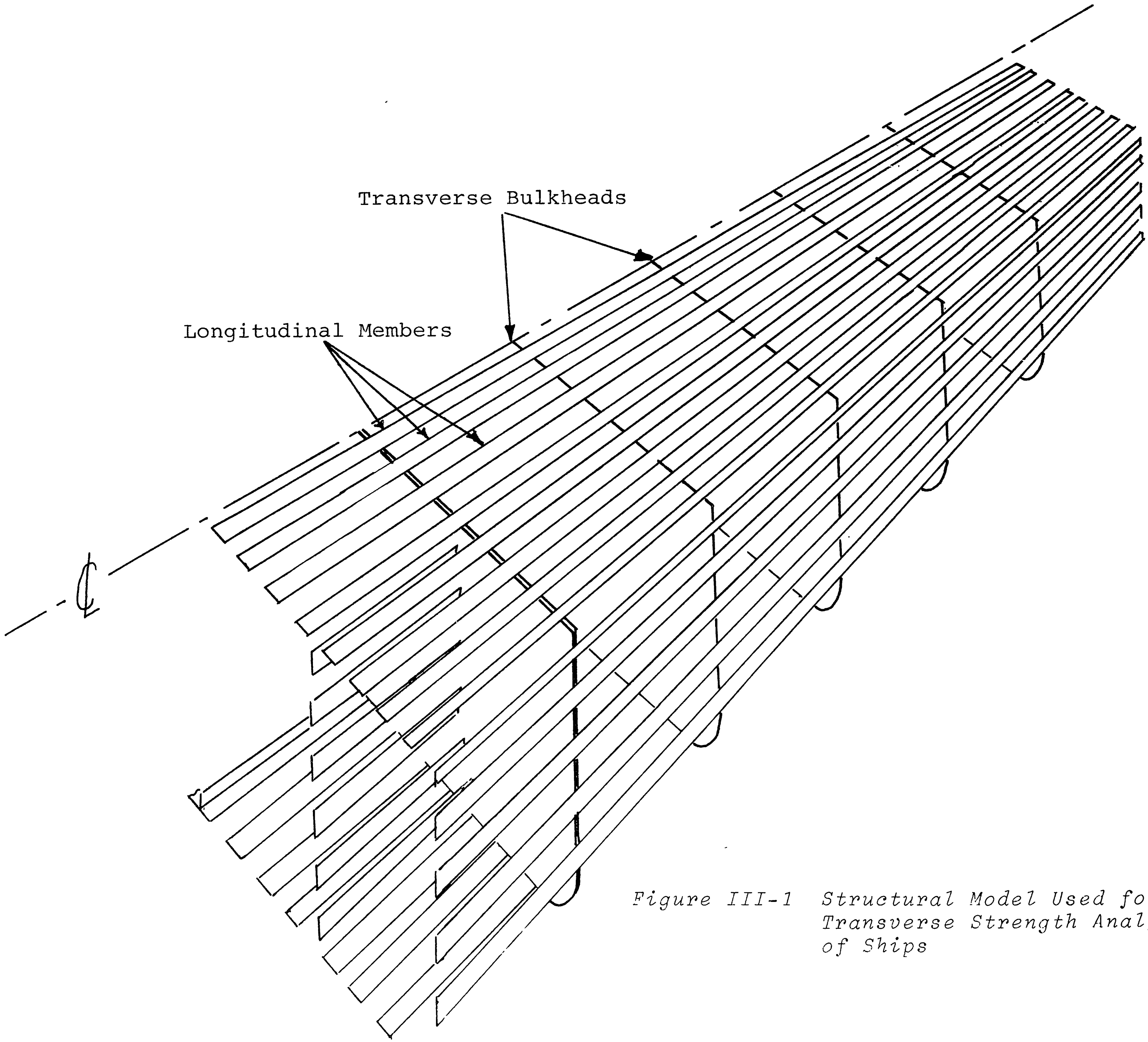


Figure III-1 Structural Model Used for the Transverse Strength Analysis of Ships and z_j in the x direction. Then the deflections of the longitudinals can be expressed as

$$U_{i\alpha} = d_{xi\alpha} - \sum_j A^x_{ij} X_{j\alpha}$$

$$V_{i\alpha} = d_{yi\alpha} - \sum_j A^y_{ij} Y_{j\alpha} \qquad \text{(III-1)}$$

The deflections, $d_{i\alpha}$, and the influence coefficients, A_{ij}, are found by line solution methods as described in the longitudinal analysis section. It should be noted that A_{ij} is the same for each longitudinal since they are of similar stiffness. Let the reactions, $X_{i\alpha}$ and $Y_{i\alpha}$, be represented by a set of equivalent forces, $\overline{X}_{i\alpha}$ and $\overline{Y}_{i\alpha}$, which are available from the longitudinal analysis. Now the displacements at the intersection can be expressed as

$$U_{i\alpha} = \sum_\beta \left[\frac{L_{xx\alpha\beta}}{P_i}\left(X_{i\beta} - \overline{X}_{i\beta}\right) + \frac{L_{xy\alpha\beta}}{P_i}\left(Y_{i\beta} - \overline{Y}_{i\beta}\right)\right] + U_{0i}$$

$$V_{i\alpha} = \sum_\beta \left[\frac{L_{xy\alpha\beta}}{P_i}\left(X_{i\beta} - \overline{X}_{i\beta}\right) + \frac{L_{yy\alpha\beta}}{P_i}\left(Y_{i\beta} - \overline{Y}_{i\beta}\right)\right] + V_{0i} \qquad \text{(III-2)}$$

where U_0, V_0 are rigid body displacements of the transverse, and L_{xx}, L_{xy}, L_{yy} are influence coefficients of a general transverse. Since all transverses have proportional stiffness then the influence coefficient for the i-th transverse, say $L^i_{xx\alpha\beta}$, between the intersection with the α-th and β-th longitudinals can be expressed as $L_{xx\alpha\beta}/P_i$ where P_i is the relative stiffness factor for the i-th transverse. P_i is computed as the ratio of the shear area of a particular transverse to that of the general transverse. The rigid body displacements of the transverses are available from the longitudinal strength analysis.

By multiplying (III-2) by A_{ij} and using (III-1), (III-2) can be expressed as

$$A_{ij}\ P_j U_{j\alpha} = L_{xx\alpha\beta} r^{\beta} A_{ij}^{\beta x} \left(X_{j\beta} - \overline{X}_{j\beta} \right) + L_{xy\alpha\beta}\ t^{\beta} A_{ij}^{\beta y} \left(Y_{j\beta} - \overline{Y}_{j\beta} \right)$$

$$+ A_{ij}\ P_j U_{0j} \quad \ldots \text{ (sum on j and } \beta) \qquad \text{(III-3)}$$

where r^{β} and t^{β} are scalar factors such that

$$A_{ij} = r^{\beta} A_{ij}^{\beta x} = t^{\beta} A_{ij}^{\beta y} \qquad \text{(III-4)}$$

Let

$$D_{ix\beta} = A_{ij}^{\beta x}\ X_{j\beta}\ , \quad \overline{D}_{xi\beta} = A_{ij}^{\beta x}\ \overline{X}_{j\beta}$$

$$\overline{U}_{i\alpha} = P_i^{1/2} U_{i\alpha}$$

$$A = \left[A_{ij} \right] \qquad \ldots \text{ (matrix)}$$

$$P^{1/2} = \left\{ P^{1/2} \right\} \qquad \ldots \text{ (diagonal matrix)}$$

then

$$P^{1/2} A P^{1/2} \overline{U}_{\alpha} = L_{xx\alpha\beta} r^{\beta} \left[P^{1/2} \left(D_{x\beta} - \overline{D}_{x\beta} \right) - \overline{U}_{\beta} \right]$$

$$+ L_{xy\alpha\beta} t^{\beta} \left[P^{1/2} \left(D_{y\beta} - \overline{D}_{y\beta} \right) - \overline{V}_{\beta} \right] + P^{1/2} A P U_0 \qquad \text{(III-5)}$$

$$\ldots \text{ (sum on } \beta)$$

It can be shown that the matrix $P^{1/2} A P^{1/2}$ is symmetric. Then there exists a unitary matrix B such that

$$B^T P^{1/2} A P^{1/2} B = \left\{ \lambda_i \right\} \qquad \text{(III-6)}$$

where B^T is the transpose of B and λ_i are the eigenvalues of the matrix $P^{1/2}AP^{1/2}$. Multiplying (III-5) by B^T , letting $\bar{U}_{i\alpha} = B_{ij}\bar{\bar{U}}_{j\alpha}$ and using (III-6), equation (III-5) can be rewritten as

$$\left\{\lambda_i\right\}\bar{\bar{U}}_{\alpha} = L_{xx\alpha\beta}r^{\beta}\left[B^TP^{1/2}\left(D_{x\beta} - \bar{D}_{x\beta}\right) - \bar{\bar{U}}_{\beta}\right] \qquad \text{(III-7)}$$

$$+ L_{xy\alpha\beta}t^{\beta}\left[B^TP^{1/2}\left(D_{y\beta} - \bar{D}_{y\beta}\right) - \bar{\bar{V}}_{\beta}\right] + B^TP^{1/2}APU_0$$

... (sum on β)

Let

$$C_{xi} = \sum_j B^T_{ij}P^{1/2}_i \left(D_{xj\alpha} - \bar{D}_{xj\alpha}\right)$$

$$C_{yi} = \sum_j B^T_{ij}P^{1/2}_i \left(D_{yj\alpha} - \bar{D}_{yj\alpha}\right)$$

$$\bar{\bar{U}}_{0i} = B^TP^{1/2}APU_{0i}$$

$$\bar{\bar{V}}_{0i} = B^TP^{1/2}APV_{0i}$$

then

$$\lambda_i\bar{\bar{U}}_i = L_{xx}R\left(C_{xi} - \bar{\bar{U}}_i\right) + L_{xy}T\left(C_{yi} - \bar{\bar{V}}_i\right) + \bar{\bar{U}}_{0i} \qquad \text{(III-9)}$$

for each α , where $\left(L_{xx}RC\right)_{\alpha} = \sum_{\beta}\left[L_{xx\alpha\beta}r^{\beta}C_{\beta}\right]_{\alpha}$

Similarly

$$\lambda_i\bar{\bar{V}}_i = L_{xy}R\left(C_{xi} - \bar{\bar{U}}_i\right) + L_{yy}T\left(C_{yi} - \bar{\bar{V}}_i\right) + \bar{\bar{V}}_{0i} \qquad \text{(III-10)}$$

Equations (III-9) and (III-10) involving $\bar{\bar{U}}_{i\alpha}$, $\bar{\bar{V}}_{i\alpha}$ are uncoupled equations involving only one i . These equations can be further simplified by letting

$$L = \begin{bmatrix} L_{xx} & L_{xy} \\ L_{xy} & L_{yy} \end{bmatrix} \begin{bmatrix} R & 0 \\ 0 & T \end{bmatrix}$$

$$\bar{\bar{W}}_i = \begin{bmatrix} \bar{\bar{U}} \\ \bar{\bar{V}} \end{bmatrix} , \quad C_i = \begin{bmatrix} C_{xi} \\ C_{yi} \end{bmatrix} , \quad \bar{\bar{W}}_{0i} = \begin{bmatrix} \bar{\bar{U}}_{0i} \\ \bar{\bar{V}}_{0i} \end{bmatrix}$$

Now (III-9) and (III-10) can be written as

$$\lambda_i \bar{\bar{W}}_i = L \left(C_i - \bar{\bar{W}}_i \right) + \bar{\bar{W}}_{0i}$$

or

$$\bar{\bar{W}}_i = \left(L + \lambda_i I \right)^{-1} \left(LC_i + \bar{\bar{W}}_{0i} \right) \qquad \text{(III-11)}$$

All values on the right hand side of this equation are known so $\bar{\bar{W}}_i$ can be computed. This is a set of transformed displacements which can be computed with relative ease. By applying the inverse transformation to these displacements, the real displacements can be computed.

$$\begin{bmatrix} U \\ V \end{bmatrix} = \begin{bmatrix} P^{-1/2}B & 0 \\ 0 & P^{-1/2}B \end{bmatrix} \begin{bmatrix} \bar{\bar{U}} \\ \bar{\bar{V}} \end{bmatrix} \qquad \text{(III-12)}$$

With these deflections computed, equation (III-1) can be used to find the reactions, $X_{i\alpha}$ and $Y_{i\alpha}$. This completes the loading pattern on any particular section of the ship so stresses and bending moments can be found at particular points of interest.

Equation (III-10) is equivalent to the equation of the deformation of an elastic body which is supported by springs and is acted upon by a set of concentrated loads. These forces and spring factors are known so the deformations can be computed. This is a more economical computation if the stiffness of the longitudinals is much less than the stiffness of the transverses.

Physically this appears to be a reasonable assumption. If $\lambda_i >> L_{\alpha\beta}$, then

$$\left(L + \lambda_i I\right)^{-1} = \frac{1}{\lambda_i}\left(I - \frac{L}{\lambda_i} + \frac{L}{\lambda_i^2} + \ldots\right) \qquad \text{(III-13)}$$

Using this, equations (III-9) and (III-10) can be rewritten as

$$\begin{bmatrix} \bar{\bar{U}}_i \\ \bar{\bar{V}}_i \end{bmatrix} = \sum_{n=1}^{\infty} \begin{bmatrix} L_{xx}R & L_{xy}T \\ L_{xy}R & L_{yy}T \end{bmatrix} \left(\frac{1}{\lambda_i}\right)^n \begin{bmatrix} C_{xi} - \lambda_i^{-1}\,\bar{\bar{U}}_{0i} \\ C_{yi} - \lambda_i^{-1}\,\bar{\bar{V}}_{0i} \end{bmatrix} + \frac{1}{\lambda_i}\begin{bmatrix} \bar{\bar{U}}_{0i} \\ \bar{\bar{V}}_{0i} \end{bmatrix} \qquad \text{(III-14)}$$

Now if we let

$$\begin{bmatrix} \bar{\bar{U}}_i \\ \bar{\bar{V}}_i \end{bmatrix}^1 = \frac{L}{\lambda_i}\left(C_i - \bar{\bar{W}}_{0i}/\lambda_i\right) \qquad \text{(III-15)}$$

and

$$\begin{bmatrix} \bar{\bar{U}}_i \\ \bar{\bar{V}}_i \end{bmatrix}^n = -\frac{L}{\lambda_i}\begin{bmatrix} \bar{\bar{U}}_i \\ \bar{\bar{V}}_i \end{bmatrix}^{n-1} \qquad \text{(III-16)}$$

Then (III-14) can be written

$$\begin{bmatrix} \bar{\bar{U}}_i \\ \bar{\bar{V}}_i \end{bmatrix} = \sum_{n=1}^{\infty}\begin{bmatrix} \bar{\bar{U}}_i \\ \bar{\bar{V}}_i \end{bmatrix}^n + \frac{1}{\lambda_i}\begin{bmatrix} \bar{\bar{U}}_{0i} \\ \bar{\bar{V}}_{0i} \end{bmatrix} \qquad \text{(III-17)}$$

The series term in (III-17) will converge rapidly, and generally only a few terms will be needed. The accuracy in this approach can be determined by applying the inverse transformation to the preceding equations.

First we can express the x-component of (III-15) as

$$\bar{\bar{U}}_\alpha = L_{xx\alpha\beta}r^\beta(1/\lambda)C_{x\beta} - \bar{\bar{U}}_0/\lambda + L_{xy\alpha\beta}t^\beta(1/\lambda)C_{y\beta} - \bar{\bar{V}}_0/\lambda \qquad \text{(III-18)}$$

But from (III-8)

$$C_{x\beta} = B^T P^{1/2} \left(D_{x\beta} - \bar{D}_{x\beta}\right)$$

and (III-19)

$$C_{y\beta} = B^T P^{1/2} \left(D_{y\beta} - \bar{D}_{y\beta}\right)$$

Now substituting (III-19), (III-8), (III-12) into (III-18) results in

$$U^1_\alpha = L_{xx\alpha\beta} r^\beta P^{-1/2} B \lambda^{-1} \left[B^T P^{1/2} \left(D_{x\beta} - \bar{D}_{x\beta}\right) - \lambda^{-1} B^T P^{1/2} A P U_0\right]$$

(III-20)

$$+ L_{xy\alpha\beta} t^\beta P^{-1/2} B \lambda^{-1} \left[B^T P^{1/2} \left(D_{y\beta} - \bar{D}_{y\beta}\right) - \lambda^{-1} B^T P^{1/2} A P V_0\right]$$

Then using the relations of (III-6)

$$U^1_\alpha = L_{xx\alpha\beta} r^\beta P^{-1} A^{-1} \left(D_{x\beta} - \bar{D}_{x\beta} - U_0\right) + L_{xy\alpha\beta} t^\beta P^{-1} A^{-1} \left(D_{y\beta} - \bar{D}_{y\beta} - V_0\right) \quad \text{(III-21)}$$

Introducing (III-4) and the fact that $L^i_{xx\alpha\beta} = L_{xx\alpha\beta}/P_i$, (III-21) becomes

$$U^1_\alpha = L^i_{xx\alpha\beta} \bar{A}^{\beta x}_{ij} \left(D_{xj\beta} - \bar{D}_{xj\beta} - U_{0j}\right) + L^i_{xy\alpha\beta} \bar{A}^{\beta y}_{ij} \left(D_{yj\beta} - \bar{D}_{yj\beta} - V_{0j}\right) \quad \text{(III-22)}$$

A similar procedure can be used to give the following expression for the y-displacement:

$$V^1_\alpha = L^i_{xy\alpha\beta} \bar{A}^{\beta x}_{ij} \left(D_{xj\beta} - \bar{D}_{xj\beta} - U_{0j}\right) + L^i_{yy\alpha\beta} \bar{A}^{\beta y}_{ij} \left(D_{yj\beta} - \bar{D}_{yj\beta} - V_{0j}\right) \quad \text{(III-23)}$$

where $\bar{A}^{\beta x}_{ij} = \left[A^{\beta x}_{ij}\right]^{-1}$

Also as in (III-16)

$$U^n = - L^i_{xx\alpha\beta} \bar{A}^{\beta x}_{ij} U^{n-1}_j - L^i_{xy\alpha\beta} \bar{A}^{\beta y}_{ij} V^{n-1}_j$$

(III-24)

$$V^n = - L^i_{xy\alpha\beta} \bar{A}^{\beta x}_{ij} U^{n-1}_j - L^i_{yy\alpha\beta} \bar{A}^{\beta y}_{ij} V^{n-1}_j$$

Using the inverse transformation of (III-12), equation (III-14) reduces to

$$\begin{bmatrix} U \\ V \end{bmatrix} = \sum_{n=1}^{\infty} \begin{bmatrix} U \\ V \end{bmatrix}^n + \begin{bmatrix} U_0 \\ V_0 \end{bmatrix} \qquad \text{(III-25)}$$

As long as the external loading can be represented by a set of concentrated forces $\tilde{X}_{i\alpha}$, $\tilde{Y}_{i\alpha}$ then (III-25) can be written as

$$\begin{aligned} U_{i\alpha} &= L^i_{xx\alpha\beta}\tilde{X}_{i\beta} + L^i_{xy\alpha\beta}\tilde{Y}_{i\beta} + \tilde{U}_{i\alpha} + \sum_{n=2}^{\infty} U^n_i + U_{i0} \\ V_{i\alpha} &= L^i_{xy\alpha\beta}\tilde{X}_{i\beta} + L^i_{yy\alpha\beta}\tilde{Y}_{i\beta} + \tilde{V}_{i\alpha} + \sum_{n=2}^{\infty} V^n_i + V_{i0} \end{aligned} \qquad \text{(III-26)}$$

where

$$\begin{aligned} \tilde{U}_{i\alpha} &= - L^i_{xx\alpha\beta}\left(\tilde{X}_{i\beta} + \bar{A}^{\beta x}_{ij}\, U_{0j}\right) - L^i_{xy\alpha\beta}\left(\tilde{Y}_{i\beta} + \bar{A}^{\beta x}_{ij}\, V_{0j}\right) \\ \tilde{V}_{i\alpha} &= - L^i_{xy\alpha\beta}\left(\tilde{X}_{i\beta} + \bar{A}^{\beta x}_{ij}\, U_{0j}\right) - L^i_{yy\alpha\beta}\left(\tilde{Y}_{i\beta} + \bar{A}^{\beta x}_{ij}\, V_{0j}\right) \end{aligned} \qquad \text{(III-27)}$$

In examining (III-26), it can be seen that without the third and fourth terms, it is equivalent to a two dimensional analysis. In ignoring these terms, an error is introduced into the analysis. This error increases as the stiffness of the longitudinals increases relative to that of the transverses. There is in fact a limit at which the series term in (III-26) is divergent. Unless the condition of (III-13) is met, the analysis cannot be continued in this manner. Instead, the direct application of (III-11) is needed to complete the analysis.

DISCUSSION

Dr. Chang's analysis utilizes several well documented techniques of structural analysis. Each is covered in sufficient detail in the body of this paper to provide the reader with an adequate understanding of the problem as a whole. Those who are not familiar with one or more methods and who desire greater detail should study the suggested references.

The fundamental analysis tool is the finite element method. This very versatile technique is undergoing a state of constant change and improvement. One of the first applications of this method in the field of Naval Architecture was presented by J. R. Paulling in 1964. Since that time many investigators have used finite elements to solve difficult structural problems with much success. However, as increasingly larger problems were attempted using this method, the cost became greater and the results less accurate. The method is very appealing on the surface. It treats problems of varying complexity and size with equal respect. It therefore offers the practicing engineer a tool which requires very little thinking. Such a tool can be very dangerous if used without understanding and prudence.

The finite element method depends on four factors for accuracy. They are discretization, element type, number of elements and rounding error, and the accuracy of the boundary conditions. If conforming elements are used, the solution should converge to the exact solution as the element area approaches zero. Unfortunately, as the mesh size is decreased the accuracy limitations on the computer itself may result in an accumulation of round off errors. There always will be a limit where the increase in rounding error is larger than the decrease in discretization error. For a small structure, the actual error obtained in using the finite element method can be made quite small. In a ship, this limit may be reached when the actual error is still quite significant. Then a decrease in mesh size will cause deterioration in accuracy. Even if this limit has not been reached, the additional computation time may be uneconomical. One attempt at skirting this problem has been the use of a large mesh for the whole structure and then taking the results and applying them as boundary conditions of a smaller region analyzed with smaller elements. This procedure can then be repeated as often as necessary until the region of interest is isolated. This process breaks down however, because the known inaccuracy in the original whole structure analysis is applied on the smaller region. Improvements to this procedure are needed whether they be more highly developed elements or more reliable ways in which to subdivide the structure [6,7,10,12,17].

The method proposed by Chang combines the advantages of the finite element method in the analysis of individual transverse members and the advantage of grillage theory in the overall analysis. Chang's basic contribution is the application of coordinate transformation techniques of grillage analysis. This isolates the transverses of interest and treats them with a simple two dimensional finite element analysis. There are two basic grillage techniques available for use. The first, a grillage of infinite element technique, is usually associated with Wah. The second, grillage beam on elastic foundations, was revived by Vedler with the most notable recent developments attributable to Michelsen, Nielsen, and Chang [5,9,11,20]. The grillage techniques require that the analysis take into consideration the discrete nature of a system of stiffeners. It has been shown that as long as certain conditions are met the method produces reliable results in a very efficient manner.

At various points in Dr. Chang's analysis, beam solutions are required. Most of these calculations are performed using the line solution method. The line solution method had its origin over a hundred years ago in Germany. It was then developed in Russia and was reintroduced to the West when computers rendered its full potential useful. In general, the method provides solutions that describe static and dynamic response as well as stability criteria of structural members for various loading, geometry and boundary conditions. It is often called the method of initial parameters (Krilov, Clebsh, McCauley) and is closely related to the Laplace transform approach of Nielsen. It is a simple method which can handle both elementary and advanced structural response problems. The beauty of the approach lies in the simplicity which makes it understandable and useable to anyone with a moderate structural background [13].

Dr. Chang's technique contains no new theory, but rather is a careful application of existing theories. The general rationale is that each technique will be used in the area where it is most accurate and efficient. Unquestionably, a certain amount of skill must be applied in such a composite approach. The final result, however, is a design tool that is simple and relatively inexpensive to use. In applying these various techniques to the whole ship analysis certain justifications must be made. Many of these assumptions are obvious from the nature of the problem, but others may not be so apparent. The following paragraphs will clarify the reasoning behind those assumptions.

The shear forces in the deck and bottom plating are neglected in comparison to the reactions between the prime longitudinals and transverses at their intersections. This assumption is based on the fact that measured results indicate a difference of about an order of magnitude between the stresses at those locations. Roberts reported of tests on a 90,000 dwt tanker which gave the following picture of calculated and measured shear stresses [16].

We can see that the prime longitudinal and transverse intersections have shear forces that are at least on the average an order of magnitude greater than the average at the deck and bottom. Similar results were also found by Vasta [19].

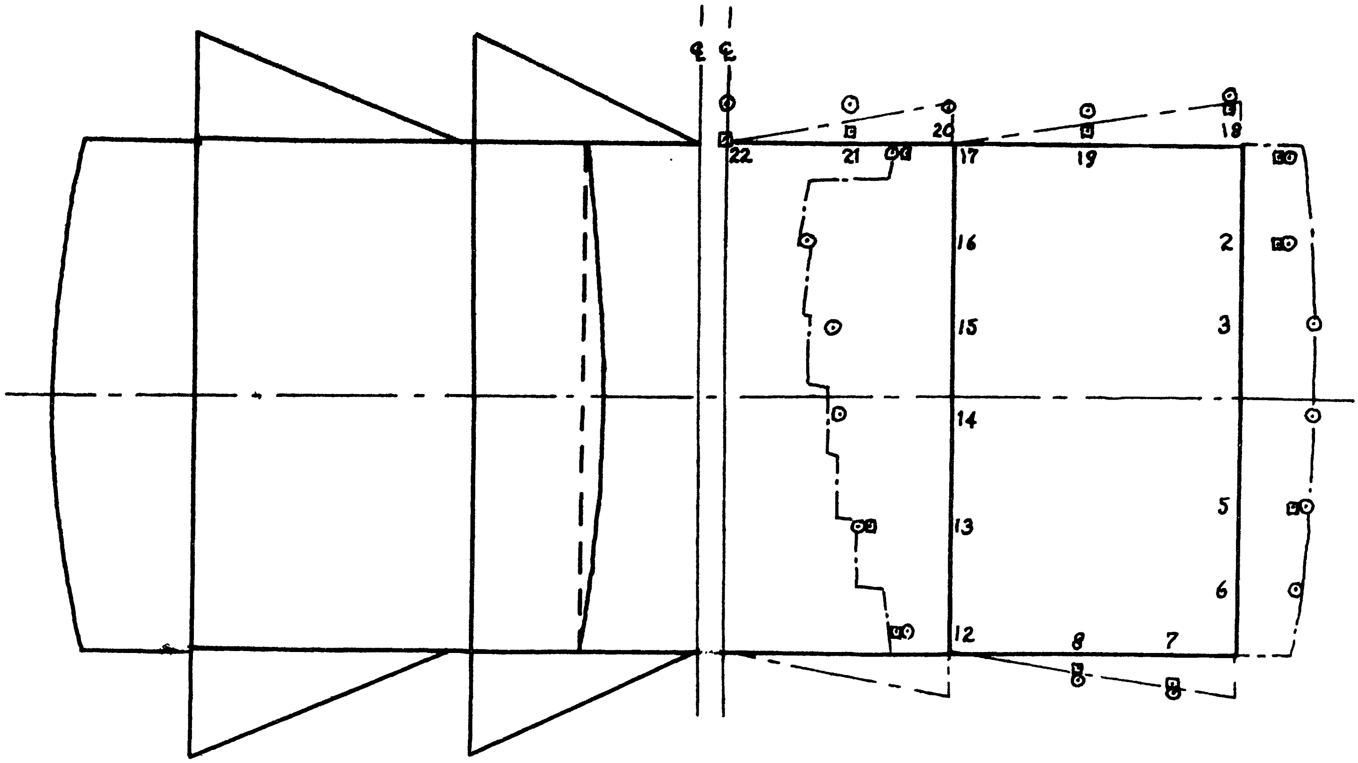


Figure IV-1 Comparison between Measured and Calculated Shear Stresses in a 90,000 dwt Tanker [161

Excluding the deck and bottom shears simplifies the calculations and reduces computational time. Such an assumption is bound to introduce some error in the final results. It appears that this is a point for future improvement. It seems possible that a good approximation can be made for applying these shear forces in the deck and bottom. From Figure IV-1 it can be seen that the deck and bottom shears follow a simple linearly varying stepped function. Rather than neglecting these forces, they could be represented by a simple function and applied as loads in the transverse analysis.

The effect of the torsional rigidity of the longitudinals has been neglected because the typical member has an open cross section. It can be shown from the theory of elasticity that such cross sections have very little torsional stiffness. The longitudinals are not designed nor are they required to exhibit torsional stiffness in tankers. The torsional stiffness that is exhibited by the longitudinals is ignored because the plane finite elements used do not allow for in plane twisting at the nodal points. Since the weight of all arguments indicates that this type of analysis produces satisfactory results, the error introduced is negligible.

The assumption of similarity of transverses or, at the very least, proportional stiffness among the transverses is important because it reduces the computations to a single finite element analysis of a typical transverse. Then, using proportional stiffness, the effective spring constants of all transverses upon each longitudinal can be computed.

For very large ships the relative uniformity of cross section should allow one to conclude that at least the web frames are similar. Since the web frame is usually considered among the most critical of ship members, it is selected as the typical transverse.

The assumption maintains that oil tight and swash bulkheads have proportional stiffness to the web frames. Thus it is said that the influence coefficients of one transverse are directly proportional to those of any other transverse.

One can derive the relative stiffness factor by comparing deflections of two transverses when each is acted on by a unit load applied as shown at α .

The proportional stiffness factor is

$$P_b = \frac{d^b_\alpha}{d^w_\alpha} \qquad \text{(IV-1)}$$

where b , w represent the bulkhead the web frame and d is deflection. The experiments of Roberts [16] show that both types of transverses can be treated as shear beams with little

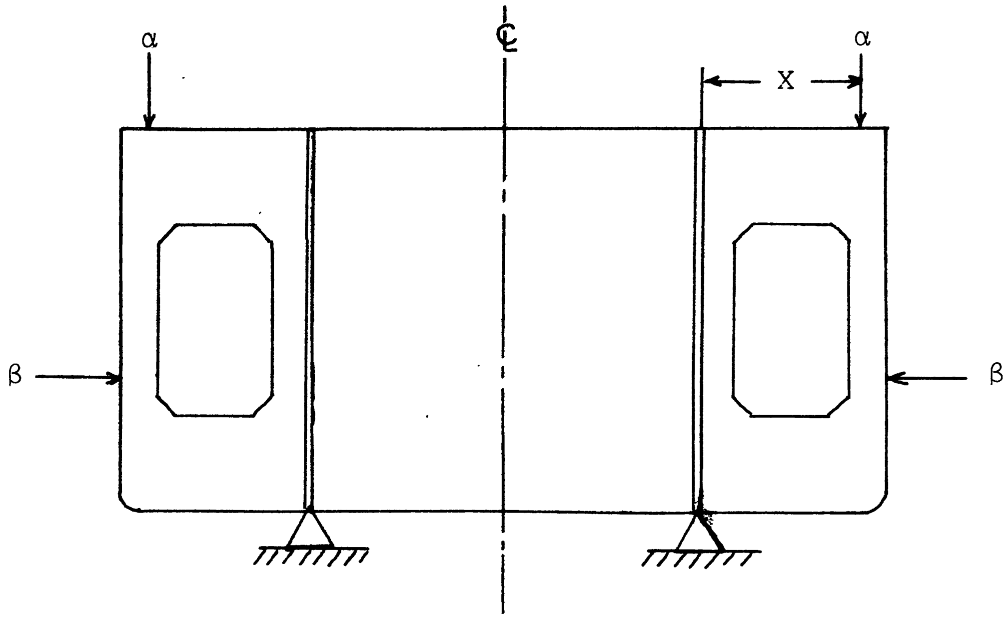


Figure IV-2 Sample Transverse Showing Model for Obtaining Influence Coefficients [4]

bending deflection.

$$d_{\alpha}^{b} = \frac{x}{GA_b} , \qquad d_{\alpha}^{w} = \frac{x}{GA_w} \tag{IV-2}$$

where G , A are the shear modulus and shear area. Simple substitution reduces (IV-1) to

$$P_b = \frac{A_b}{A_w} \tag{IV-3}$$

which is an approximate solution. The question of applying this same factor to other positions on the transverse, as for example at β , should be examined. It might be expected that this is a large source of error. That, however, is not the case. Dr. Chang shows that for a large tanker, an error in stiffness factor of 100% will produce an error of less than 1% in the results. He deliberately increased the stiffness factor by 100% for the oil tight bulkheads. The resultant changes in boundary forces were different by less than 0.5%. The results of this analysis are shown in table IV-1.

Forces in kg's

Longitudinal	P = 5.6	P = 11.2
1	102,000	102,100
2	70,180	70,210
3	70,200	70,140
4	105,300	105,100
5	140,200	140,100
6	105,200	105,300
7	70,070	70,150
8	70,200	70,180
9	70,060	70,090
10	105,100	105,100
12	-80,600	-80,510
13	-53,740	-53,800
14	-107,500	-107,300
15	-161,100	-160,000
16	-107,500	-107,500
17	-53,810	-53,700
18	-53,860	-53,850
19	-53,630	-53,750

P = Stiffness Factor

Table IV-1 The Dominant Boundary Forces on Web Frame 127 Due to Two Different Stiffness Factors [P] for the Oil-Tight Bulkheads. [4]

The reason for this somewhat surprising result comes from simple beam theory. Consider a single longitudinal represented by a simply supported beam. Let that beam be acted upon by some loading function, $q(x)$, and a linear spring with stiffness k attached at the mid-point of the beam.

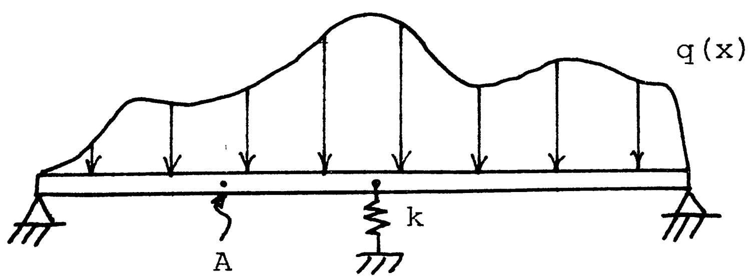


Figure IV-3

The deflection, d_A , can be computed at some arbitrary point, A . If the spring stiffness, k , is allowed to vary, then d_A can be written as a function of k . The results for a simple example are plotted below.

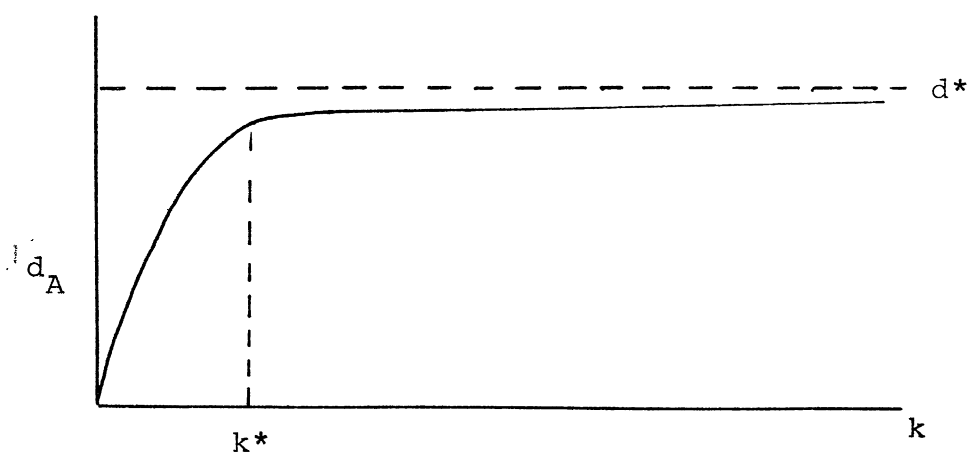


Figure IV-4

This shows that for some value of k, say k^*, the deflection, $d_A(k)$, will approach d^*, the deflection for $k=\infty$. Thus for any k greater than k^*, $d_A(k) \simeq d^*$ (a constant).

It is obvious, then, that for a sufficiently rigid transverse, an error in computing the apparent spring factor of that member will induce a negligible error in the shear force between that transverse and the longitudinal. It is physically apparent that the prime transverses of a ship are very rigid. This has been verified by Chang's results.

Dr. Chang assumes that the load transfer pattern for the transverse analysis is that which is normally used in a grillage analysis. The uniform load is borne by the stiffeners which in turn transfer it as concentrated loads to the girders. In longitudinally framed ships, the longitudinals are the stiffeners in the three dimensional grillage. Then the uniform load on the plates is transfered to the longitudinals which in turn apply it as concentrated loads on the transverses. This assumption has long been recognized in theoretical naval architecture [15,20].

The technique, as presently coded for computer use, does not take into account the twisting of the entire hull due to unsymmetrical loads. This limits the use of the existing program to tankers and vessels of similar cross section which do not exhibit large openings in the deck. From basic elasticity we know that closed section thin wall members have great torsional rigidity in comparison to open section members of about the same configuration.

In tankers the hull forms a closed section and it is assumed that even unsymmetrical loads will produce very little twist. This assumption is not valid for container ships and ore carriers, indeed for any vessel with large deck or side openings that result in an open cross section in the hull. Of course, as long as loads are symmetrical, the existing code can easily handle the analysis for such vessels.

The theory as presented can be utilized in a partial analysis of the hull. Such an analysis offers several advantages: the greatest structural interest is centered on the mid-body; the smaller section of interest reduces input data and computation time; and, finally, the results should be accurate enough for design purposes.

The question of accuracy would seem to involve three significant parameters, which are load distribution, geometry of the structure, and the section being studied. The relationships of these parameters will be examined in order.

The load distribution is important because in a structure composed of a finite number of discrete elements certain conditions occur. The external loads are apportioned among the various elements in such a way that equilibrium and compatibility are satisfied throughout the structure. The stiffer elements or substructures will share more of the load than the weaker members. The actual sharing proportion within the structure or even the terminal forces on each element are difficult to resolve without a complete analysis of the whole structure. However, certain types of loadings may be determined with reasonable accuracy. Dr. Chang has considered two such special loads.

First, consider the ship-like composite box girder with equally spaced and identical transverses.

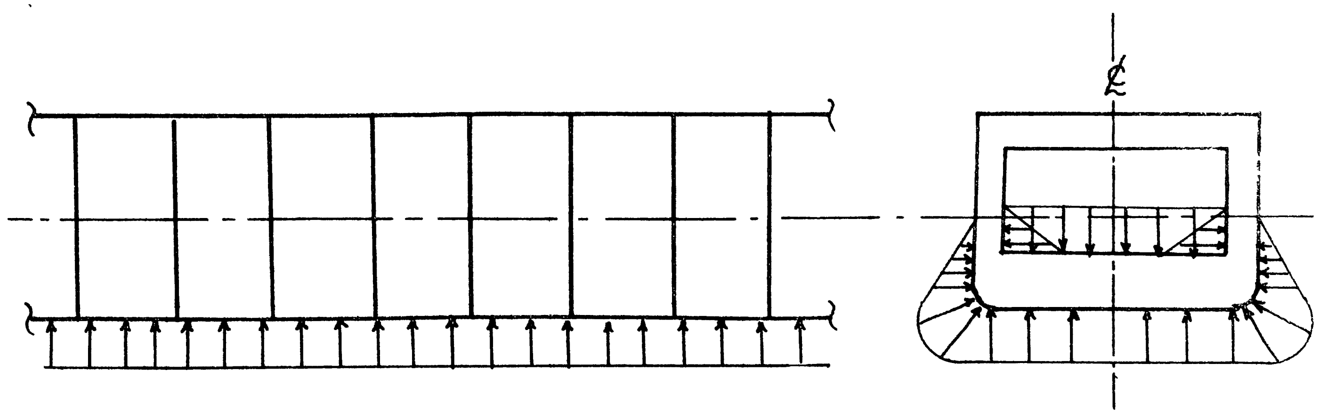


Figure IV-5 Uniform Ship-Like Girder Subjected to Uniform Load

If this structure is subjected to a uniform self-balanced external load, then each transverse will share the same amount of load independently of the longitudinal stiffness. This loading will reduce the problem to the point where the conventional two-dimensional analysis will yield the same results as the three dimensional analysis for the same hull girder.

The second case is for a non-uniform, periodic load. Then the load-sharing will occur only within the several transverses.

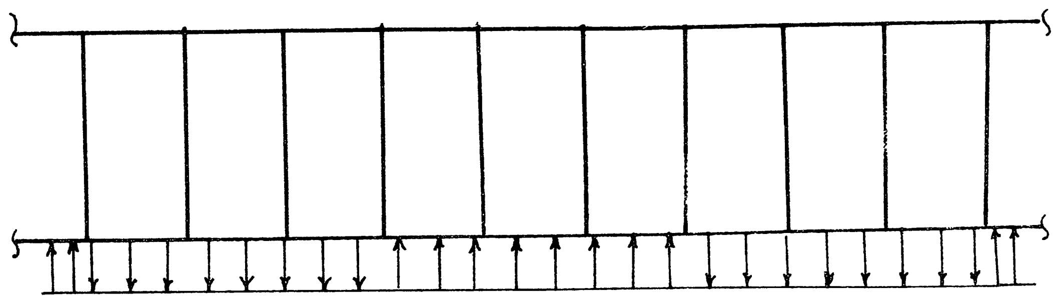


Figure IV-6 Uniform Ship-Like Girder Subjected to Periodic Load

The parameter of structure geometry can be useful if certain conditions are met. The transverses of the box girder must be of proportional stiffness and must be arranged in the same regular pattern. The load distribution must also follow this pattern for a partial analysis to be successful.

Most ships are designed with the proportional stiffness and geometric conditions satisfied. One cannot restrict loads to this extent, so it must be anticipated that loads which depart from the regular pattern must induce error.

In order to assess the magnitude of this error, Dr. Chang returns to the longitudinal strength formulation (Chapter II). The deflection of the hull can be obtained from a grillage analysis. This deflection results in the rigid body motion of the transverses.

$$V_{oi} = A^{Y}_{ij}\ Y_j \tag{IV-4}$$

where A^{Y}_{ij} are the influence coefficients of the prime longitudinals treated as simple beams, and Y_j is the difference in shear force at the j-th transverse. Now, noting that the ship hull has the same length and same fixed conditions as a prime longitudinal, it seems reasonable that the A^{Y}_{ij} can be approximated as a scalar proportion of the hull influence coefficients (a_{ij}).

$$A^{Y}_{ij} = a_{ij}/F \tag{IV-5}$$

Returning to equation III-26 in the transverse analysis, the last term will result in

$$\bar{a}_{ij}{}^{\beta y}\ V_{oj} \simeq \frac{t^{\beta}Y}{F}\ i \tag{IV-6}$$

where $\bar{a}_{ij} = a_{ij}{}^{-1}$. A similar result for the horizontal plane follows:

$$\bar{a}_{ij}{}^{\beta y}\ V_{oj} \simeq \frac{r^{\beta}X}{G}\ i \tag{IV-7}$$

In examining the left hand side of these expressions, it can be seen that neither the length of the portion nor the fixed conditions of the longitudinals is present. It can be concluded that a partial analysis of the ship will result in the same first two terms in equations (III-26) and (III-27). The third term is almost the same as for the global analysis. The only significant difference occurs in the last term. If equation (III-24) is examined, the series represents the coupling effects of the transverse deformations. In a global analysis, all of the transverses are considered and all their coupling effects are accounted for. In a partial analysis, only those transverses within that section are considered. If the load distribution meets the aforementioned criteria, then the coupling will not include all transverses in any case. The case of identical transverses and uniform loads will result in all terms being negligible except the first two. It will reduce to the conventional two dimensional analysis.

This analysis and the computer code based on it have neglected the coupling effects of the sections of the structure not included in the partial analysis. The error induced by this omission will increase with the stiffness of the longitudinals. Dr. Chang believes that this error is probably less significant than conducting a full analysis which will introduce greater round-off error because of the increased degrees of freedom. In the total three dimensional finite element analysis, the error due to discretization by using a coarse mesh can also be significant.

RECOMMENDATIONS

The theory as presented thus far has been incorporated into a set of computer codes originally developed by Dr. Chang and more recently modified by the authors to permit usage on the Michigan Terminal System. There are limitations on the program as presented. The following should clarify these limitations and offer some possible future improvements.

The torsion and horizontal bending of the hull have been neglected. This limits the analysis to vessels with substantial horizontal and torsional stiffness for the hull. Torsional stiffness might be considered adequate for tankers and similar vessels of closed cross section and torsion might be ignored. Horizontal bending is another matter, since the response of the hull to horizontal loading seems to be approximately proportional to the vertical response for the same load. The proportional factor would be the ratio of the respective moments of inertia. In horizontal bending, only unsymmetric loads can introduce an effective horizontal load. Since stability questions arise from very unsymmetric loading, it appears that horizontal bending can be ignored in still water calculations.

The desirability of including both effects arises when considering dynamic loading. It is the dynamic problem that should generate the greatest interest. There have been several recent advances in ship motions and sea loads. The marriage of a good computer analysis in that field with the structural analysis is a natural one and should be ardently pursued. Dr. Chang has considered the horizontal bending and torsion problems and has proposed an extension to the existing program.

The basic assumption is that any deformation in the ship's structure is small enough to allow independent calculation of the stresses due to vertical bending, horizontal bending and twisting. The method of calculating the horizontal bending is similar to the prior treatment of the vertical bending. The change is in the loads applied to obtain the results. In the vertical bending problem, only vertical loads were considered. Now the same calculation is repeated using the horizontal loading condition. The resulting stresses are then added to those obtained in the vertical bending calculations.

In order to analyze the twisting problem the hull is modeled as an open thin wall beam with braces as shown in Figure V-1.

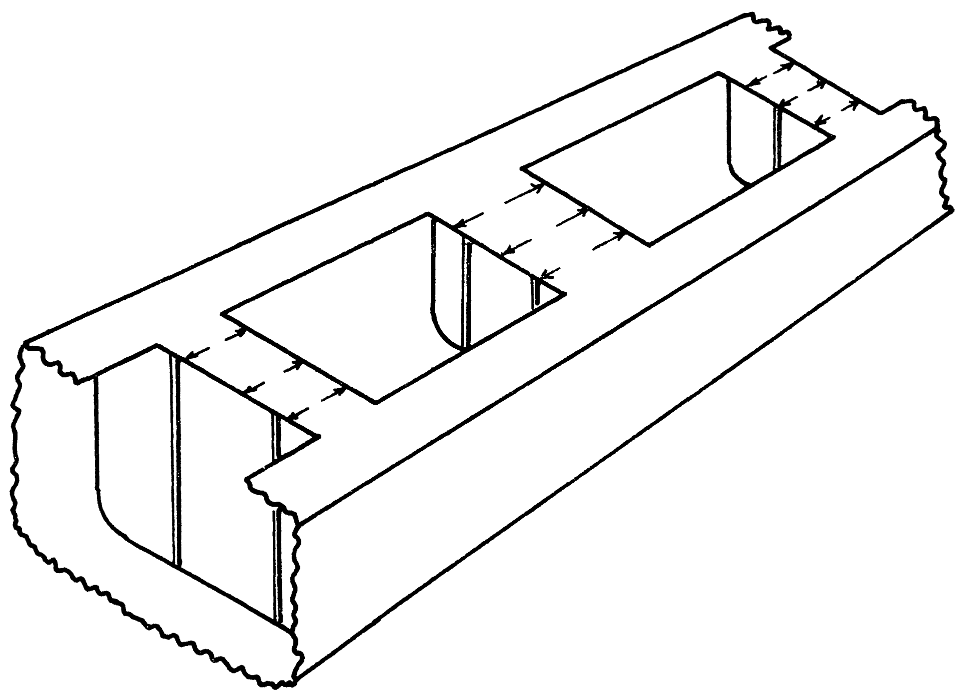


Figure V-1 Ship-Like Model for Torsion Effects Showing the Longitudinal Stresses of Interest at Deck Openings

The assumption is made that the cross section between the braces is constant. The line solution method is then applied. The transfer matrix between the state variables between two stations without loads is

$$\begin{bmatrix} \phi \\ \psi \\ M_B \\ M_T \\ 1 \end{bmatrix}_{i+1} = \begin{bmatrix} L_{11} & L_{12} & L_{13} & L_{14} & 0 \\ L_{21} & L_{22} & L_{23} & L_{24} & 0 \\ L_{31} & L_{32} & L_{33} & L_{34} & 0 \\ L_{41} & L_{42} & L_{43} & L_{44} & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}_i \begin{bmatrix} \phi \\ \psi \\ M_B \\ M_T \\ 1 \end{bmatrix}_i \qquad \text{(V-1)}$$

or $S_{i+1} = L_i S_i$ where

ϕ = the twisting angle

ψ = the derivative of the twisting angle

M_B = the bending moment

M_T = the twisting moment

L_{ij} $(i = 1,4;\ j = 1,4)$ are given in table IV-2.

$\beta^2 = C_\psi / C_\omega$

C_ψ = the torsional rigidity

C_ω = the warping rigidity

Table IV-2 - The Transfer Matrix (L_{ij})

1	$-\dfrac{\sin\beta(x-a_i)}{\beta}$	$-\dfrac{(1-\cos\beta(x-a_i))}{C_\omega\beta^2}$	$\dfrac{-\beta(x-a_i)+\sin\beta(x-a_i)}{C_\omega\beta^3}$
0	$\cos\beta(x-a_i)$	$\dfrac{\sin\beta(x-a_i)}{C_\omega\beta}$	$\dfrac{1-\cos\beta(x-a_i)}{C_\omega\beta^2}$
0	$-\beta C_\omega \sin\beta(x-a_i)$	$\cos\beta(x-a_i)$	$\dfrac{\sin\beta(x-a_i)}{\beta}$
0	0	0	1

The transfer matrix for a concentrated twisting moment, M_T , is

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -M_T \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The global matrix L is formed in the standard way [13] as

$$S_{n+1} = L_n L_{n-1} \cdots L_0 S_0$$

or $$S_{n+1} = LS_0$$

where $$L = L_n L_{n-1} \cdots L_0 \qquad \text{(V-2)}$$

If the effects of the braces are treated as redundant and if Z_j is the total shear force across the middle section of the i-th brace, as shown in Figure V-1, then Z_j can be computed as follows:

$$Z_j = \frac{-d_i}{\alpha_{ij} + \delta_{ij}\beta_j} \qquad \text{(V-3)}$$

where d_i = the deformation at the i-th brace due to external loads

δ_{ij} = Dirac delta function = 1 if $i = j$ or 0 if $i \neq j$

α_{ij} = deformation at the i-th cutout due to unit load at j-th cutout

β_j = deformation at the j-th cutout due to deformation of the j-th brace.

α_{ij} and d_i can be computed using the line solution method by treating the ship hull as a thin wall beam without using braces. B_j can be found using shear beam theory. Thus equation V-3 can be solved directly for Z_j . Now applying the line solution method with Z_j known, the real deformations and stresses on the hull can be determined. Again, due to the basic assumption of superposition, these stresses can be added to those for vertical and horizontal bending. Since it is a known fact that stress concentrations occur at the hatch corners, they should probably be analyzed by the finite element method using the known stress distribution for boundary conditions.

The effect of including the stresses due to horizontal bending and twisting will vary depending on the problem conditions. The principal factors would appear to be the magnitude of unsymmetrical loads, the relative dimensions of the openings to the hull dimensions, and the design of the cross braces.

In any program which is to be used as a design tool, data preparation is an important consideration. The user must expend a certain amount of effort due to the complexity of the problem, but this should be reduced wherever possible to minimize error. The input data preparation required for the use of the existing program is somewhat tedious. Automatic data generation routines could overcome this inconvenience. This is essentially a programming problem. Any changes made here should be considered as part of interfacing this program with a sea loads program. The other inputs to the program are very straight forward and could be made directly from a blueprint. The program graphically depicts much of the structural input so that errors are pictorially obvious. The authors included one small change in the Michigan version which allows the program to be stopped after all input and after the eigenvalues have been determined. This enables the batch user to stop the program and check the data input. At the same time he can check the eigenvalues and determine how many should be used to attain the desired level of accuracy. This can be done at very low cost. When all the input errors are eliminated the code is changed and the program may be run to completion.

The efficiency of a program is determined by the type of machine the investigator is utilizing. The existing program of Dr. Chang is most efficient on computers with limited storage capacity. The running time on the Univac 1108 was about six minutes for the example used. The finite element routine used was developed by Paulling for an IBM 7094. The limited storage capacity of smaller computers necessitates temporary storage of intermediate data on some device in order to insure adequate core storage for computations. The original version is thus entirely satisfactory for limited storage computers.

The features of data removal and replacement within the program are time consuming and unnecessary for large machines such as the IBM 360/67. The authors modified the finite element routine extensively as far as data manipulation is concerned and were successful in cutting CPU time for this section in half for a small test case. This test run was a simple cantilever beam problem which was modelled using 33 nodes or 66 degrees of freedom and 30 quadrilateral elements. The analysis time after modifications was 43 seconds CPU time. In our experience this does not compare favorably with existing finite element method routines developed specifically for the Michigan Terminal System. The authors ran a similar plane problem having 72 nodes or 144 degrees of freedom with 105 triangular elements in 7.1 seconds of CPU time using such a program. It is our strong belief that adapting the analysis around a Michigan Department of Naval Architecture finite element routine would greatly reduce analysis time over all. Since the evidence indicates that the transverse analysis can be handled as a two dimensional problem there is no need to incorporate three dimensional elements. Perhaps the best combination would be two dimensional isoparametric elements and pin ended bars to represent plating and flanges. In order to appreciate the extent to which the finite element analysis dominates the computation time, consider that our experience shows it to be 90% of the total analysis time. Any improvement in efficiency of this section will substantially affect the total analysis.

Time has prevented the authors from initiating more changes themselves. A small test data set and results which can be run at minimal cost has been developed to encourage such a change, however.

The problems of unsymmetric loading in regard to horizontal bending and torsion have been discussed. In the case of vertical bending, the present program can handle unsymmetric loading as the sum of a symmetric and an antisymmetric system. Automatic data generation routines previously recommended should include computer breakdown of the unsymmetric loads into the two systems.

It has been assumed that the shear forces on the deck and bottom at each transverse are negligible in comparison with those in the longitudinal bulkhead and side shell. For programming purposes the shears on the prime longitudinals have been evenly distributed. This assumption is based on the findings of Roberts [16]. Examining Roberts' test example and the full scale tests reported by Vasta [19], both of these assumptions appear to be somewhat questionable. In fact these are the only assumptions of Dr. Chang which the authors believe are not well supported by current understanding of the problem. Dr. Chang himself admitted in commenting on results of his test analysis of a certain large tanker that the second assumption probably produced some error. On examining the full scale results of both Vasta and Roberts the authors believe that both shear conditions can be handled by a simple linear loading function. This function could be determined empirically from the existing full scale test results. Then the reactions between the transverses and the prime longitudinals could be distributed in a more realistic manner. There is essentially nothing wrong with the present approach except that some other linear distributions may result in better answers. This would be an easy step to accomplish and would not alter the analysis time appreciably.

The reduction of most problems to a partial analysis as presented by Dr. Chang and outlined in Chapter IV, seems reasonable for the present time. The authors believe that once again the potential user must be cautioned to weigh the restrictions this method implies. The error caused by this type of analysis will be negligible only if the coupling effects caused by the excluded structure are negligible. The parameters which affect this error have already been discussed. They are load distribution, structure geometry, and the section to be analysed. The geometric parameter should in general be satisfied for large tankers and other bulk carriers over a considerable portion of the mid-body. If one considers the entire ship, the bow and stern coupling effects would appear to be minimal. This would not be the case if analysis were limited to a small section of the parallel mid-body. In this example, the potential error will be introduced by the types of loadings in the excluded section. This difficulty would appear to increase when considering a dynamic analysis. The sea loads even at the bow and stern sections might be significant under certain conditions. Some means must be provided for including the coupling effects of the excluded sections in such a problem.

As a final recommendation for future study, these coupling effects must be examined and a means provided for entering them into the analysis. Dr. Chang offers some very convincing arguments as to why the partial analysis is preferable to the full ship analysis. There must exist a practical point at which his arguments are no longer valid. At this point the round-off and discretization errors may indeed be less than the partial error due to excluded coupling effects. Economic considerations weigh heavily for this partial analysis, but the user must have an idea of the error magnitude acquired by such an economic measure.

The authors favor an approach to the problem that will retain the basic economy of the partial analysis, without excluding any significant coupling effects. Perhaps a quasi-empirical approach for determining these effects, and then applying them as end conditions for the partial analysis would offer the greatest potential. Such an approach would admittedly contain error, but, like the deck shear function previously suggested, it would hopefully reduce the error incured by completely ignoring the effect. It seems reasonable to anticipate that not all of the coupling effects will prove significant. However, only further study can reveal the important ones.

SUMMARY AND CONCLUSION

The scope of this project as stated in our prospectus was two fold. The first objective was to produce a useable Michigan Terminal System version of the subject set of computer programs, and a user's guide was to be written for future utilization of the program. This objective has been accomplished. A direct comparison was made with the results of a full scale problem used by Dr. Chang to support his program. This comparison showed that the Michigan Terminal System version performs the same analysis as the original. The second objective was to make a search of related literature in order to obtain an understanding of the assumptions made and of the restrictions they impose on future uses of the program. The second objective has been accomplished and the remainder of this section is devoted to the authors findings in this area.

In conducting the literature search, it became apparent that most of the meaningful investigations of ships structures have occured within the last decade. In discussing the work of Vasta in 1958, St. Denis commented about the progress of structural analysis. He notes, "... one may well ask what new ideas have been introduced in ship structures during the past 25 years. The list is hardly impressive" [19]. That was a rather sweeping indictment of structures research prior to that time, but since then several important advances have been made. The most significant advance was the introduction of high speed computers. This encouraged the study of complex problems which had previously been avoided. The computer age soon produced results through three very important techniques which Dr. Chang has adopted for his work. These are the finite element, grillage, and line solution methods. During that decade the basic ship structure problem was studied in detail. Most of the basic assumptions used in this paper became popular and gained support within that period. These assumptions are discussed in Chapter IV. Some of them restrict the use of the program to certain classes of ships and loading conditions.

There is one assumption which the authors believed was not fully supported. This assumption neglected the shear forces in the deck and bottom when compared with those in the prime longitudinals. This conclusion is not well supported, but is an assumption of convenience rather than necessity. The authors have suggested a program modification to handle this problem.

On the basis of their study, the authors believe that the concept is sound and workable. The restrictions introduced by certain of the assumptions must be observed. For most of these restrictions an alternate approach has been discussed in Chapter V, but additional computer coding will be needed to implement these extensions.

The potential of this technique is fantastic because the cost of an analysis can be made low enough to permit its use as a design tool. As it is presently coded, the analysis of a loading condition on a typical large tanker required 6 minutes of CPU time on a Univac 1108 and 23 minutes CPU time on an IBM 360/67, a medium speed computer. The principal cost area in the IBM 360/67 run is the finite element analysis of the transverse. This portion requires about 90% of the total time.

In summary, the concepts advanced by Dr. Chang are essentially sound and supported by current evidence. Because of its low cost his program has great potential as a design aid in the structural analysis of several types of ships. Certain changes in programming could be made to adapt the program to various computers. This would increase the efficiency for a given system. The data generation routines could be improved to reduce the preparation time; at present data preparation while not difficult is time consumming. The automatic generation of data will increase the value of the program as a design tool with a small cost penalty.

In order to increase confidence in the method, a limited number of additional studies of problems with known results should be made. This would serve to confirm the basic assumptions which investigators other than Dr. Chang have made concerning the analysis of this type of ship structure. Most of these assumptions were advanced during the past decade. As a result, those years were most fertile for ship structural analysis. If the next decade is to be as successful, the practicing naval architect must be willing to accept and use these new ideas.

BIBLIOGRAPHY

1. Abrahamsen, E., Nordenstrom, and Roren, E., "Design and Reliability of Ship Structures," SNAME Spring Meeting, April, 1970.

2. Abrahamsen, E., "Recent Developments in the Practical Philosophy of Ship Structural Design," SNAME Spring Meeting, July, 1967.

3. Abrahamsen, E., "Structural Analysis of Large Ships," SNAME Annual Meeting, November, 1969.

4. Chang, Pin-Yu, "Report on Analysis of Large Tankers Using Grillage and Finite Element Techniques." (unpublished)

5. Chang, Pin-Yu, "Elastic Analysis of Grillages Including Torsional Effect and Stability," The University of Michigan, 1967.

6. Kamel, H. A., et al., "An Automated Approach to Ship Structure Analysis," SNAME Transactions, Vol. 77, 1969.

7. Kamel, H. A., and Liu, D., "Application of the Finite Element Method to Ship Structures," soon to be published in Journal of Computers and Structures.

8. Hagiwara, Koichi, et al., "Structural Strength of Large Bulk Carriers," Mitsubishi Heavy Industries Technical Review, January, 1970.

9. Huang, Hou-Wen, "Transverse Strength of a Ship Hull Under Lateral and In-Plane Loads," Berkeley, California, 1967.

10. Moe, J., "Finite Element Techniques in Ship Structures Design," Trondheim, Norway, 1969.

11. Nielsen, R. Analysis of Plane and Space Grillages Under Arbitrary Loading by Use of the Laplace Transformation Technique. Copenhagen, 1965.

12. Paulling, J. R., "The Analysis of Complex Ship Structures by the Finite Element Technique," Journal of Ship Research, Vol. 8, No. 3, December, 1964.

13. Pilkey, W. D., Manual for the Response of Structural Members. U. S. Department of Commerce, 1969.

14. Report of Committee Four, "Stress Distribution in Hull Structure," ISSC, Tokyo, 1970.

15. Report of Committee Five, "Stiffened Panels in 3-Dimensional Structures," ISSC, Tokyo, 1970.

16. Roberts, W. J., "Strength of Large Tankers," No. 55, Lloyd's Register of Shipping, London, January, 1970.

17. Roren, Elvald M. Q., "Transverse Strength of Tankers - Finite Element Applications," European Shipbuilding, Vol. 17, Nos. 3, 4, 1968.

18. Schade, H. G., "World-wide Survey of Ship Structure Research," Berkeley, California, 1967.

19. Vasta, John, "Lessons Learned from Full-Scale Ship Structural Tests," SNAME Transactions, Vol. 66, 1958.

20. Vedeler, G., "Grillage Beams in Ships and Similar Structures," Oslo, 1945.

21. Wah, Thein, A Guide for the Analysis of Ship Structures. U. S. Department of Commerce, 1960.

22. Zienkiewicz, O. C., The Finite Element Method in Structural and Continuum Mechanics. London, 1967.

APPENDIX A: RESULTS

The following sketch of a transverse web frame contains full scale results as well as results from the computer programs contained in this paper. Strain gage data from both sides of this plate are plotted. While this is not offered as absolute proof of the method, it does show that good correlation can be made using this method.

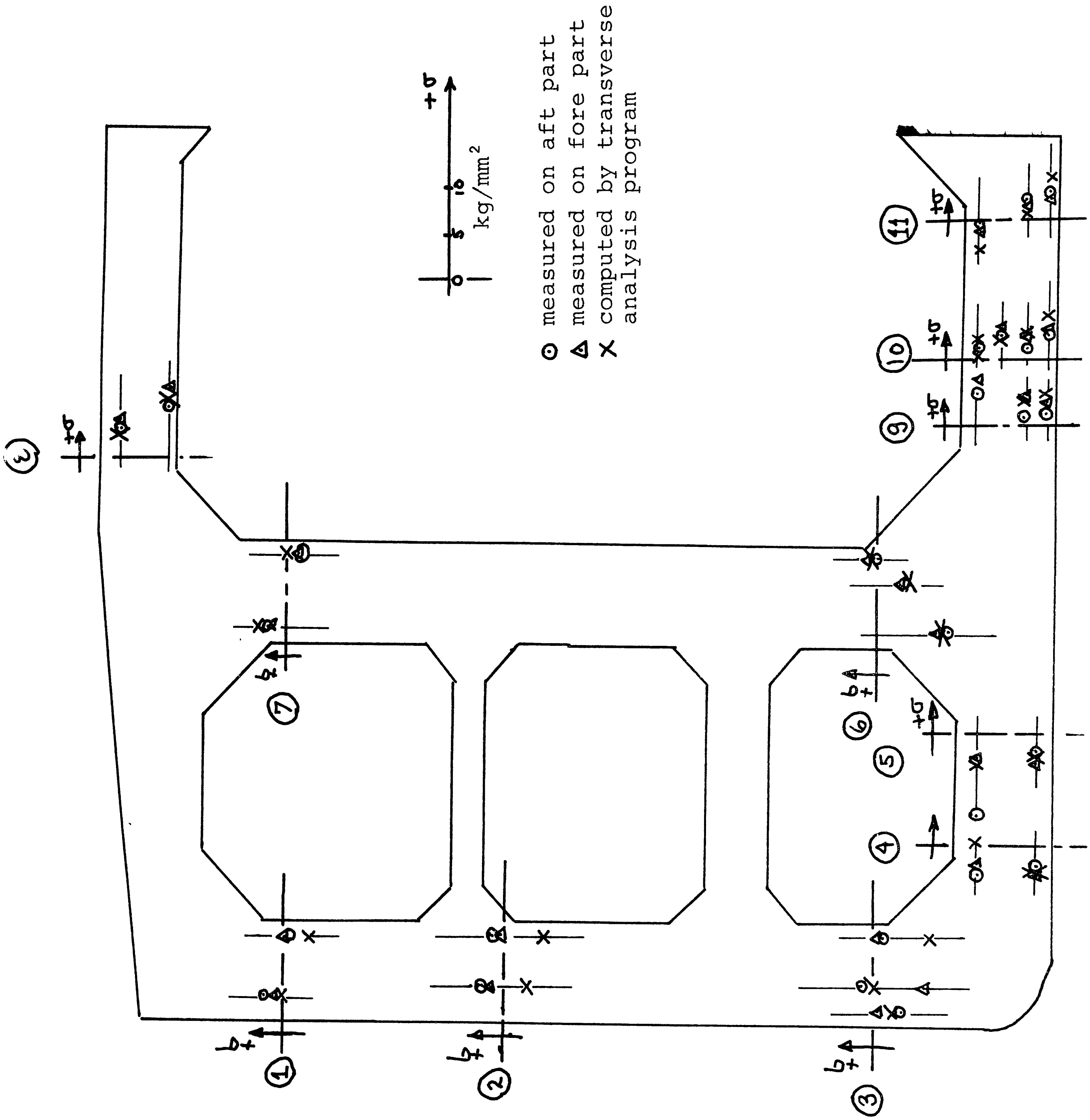


Figure A-1 Normal Stresses on Particular Web Frame of a Large Tanker [4]

APPENDIX B: COMPUTER PROGRAMS

Two computer programs are included here: one for longitudinal strength, one for transverse strength. Included with these programs are lists of input and output data for a test problem to be used with both programs.

/

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                     C
C     LONGITUDINAL STRENGTH ANALYSIS                                  C
C                                                                     C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C     WRITTEN BY PIN YU CHANG
C
C     ADAPTED FOR MTS BY
C           E. A. CHAZAL
C           J. M. PAYNE
C
C     THIS PROGRAM ANALIZES THE SHIP STRUCTURE LONGITUDINALLY MODELED AS
C     A GRILLAGE.  FOR MORE DETAILED INFORMATION, THE USER IS REFERED TO
C     THE USER'S GUIDE FOR THIS PROGRAM AND THE PROFFESSIONAL THESIS BY
C     BY CHAZAL AND PAYNE, APRIL 1971.
C
      CALL LONGAL
      STOP
      END


      SUBROUTINE LONGAL
C
C     PRIMARY LONGITUDINAL STRENGTH ROUTINE
C
        DIMENSION A(3,2),YI(3,2),Q(50,2),AF(50,50),AE(50,50)
        DIMENSION IND(50,3),JD(50),T(50),R(50)
        DIMENSION YB(50),YC(50),D(50)
         DIMENSION DY(50,2),LQ(50)
      DIMENSION DAP (50,50)
      DOUBLE PRECISION DAP
      READ(5,150) NPROB,MT
C     PROBLEM NUMBER , NUMBER OF TRANSVERSES
      READ(5,152) (JD(I),I=1,MT)
        READ (5,151)AI,AJ,XI,XJ,E,GNU,ZLEN
        READ (5,151)Y1,Y2,SM1,SM2,SN1,SN2
C AI,AJ.....WEB AREA OF THE SHELLS AND THE LONGITUDINAL BULKHEADS
C XI,XJ.....MOMENT OF INERTIA OF THE SHELLS AND BULKHEADS
C ZLEN......LENGTH OF THE HOLDS
C Y1,Y2.....WIDTH OF THE WING AND CENTRAL TANKS
C     SM1, SN1.....SECTION MODULI OF SHELL AT DECK AND BOTTOM
C     SM2, SNI.....SECTION MODULI OF BULKHEAD AT DECK AND BOTTOM
        READ (5,151)(A(I,1),I=1,3)
        READ (5,151)(A(I,2),I=1,3)
C A(I.J)....SHEAR AREA OF THE WEB FRAMES,SWASH BULKHEADS AND OIL-
C           TIGHT BULKHEADS
        READ (5,151)(YI(I,1),I=1,3)
        READ (5,151)(YI(I,2),I=1,3)
C YI(I,J)...MOMENT OF INERTIA OF THE WEB FRAMES,SWASH BULKHEADS,
C           AND OIL-TIGHT BULKHEADS
         MY=MT+1
         DO 77 I=1,MY
   77  READ (5,154)(Q(I,J),J=1,2),LQ(I)
C     UNIFORM LOADS OF THE TRANSVERSES
      WRITE(6,100) NPROB
        WRITE (6,101)
        WRITE (6,99) ZLEN,E,GNU,Y1,Y2
```

```
      WRITE (6,102)
       WRITE (6,99) AI,AJ,XI,XJ
      WRITE(6,106)
      WRITE(6,99) SM1,SN1,SM2,SN2
      WRITE(6,227)
      WRITE(6,99) ((A(I,J),I=1,3),J=1,2)
      WRITE (6,104)  ((YI(I,J),I=1,3),J=1,2)
      WRITE (6,105)
        DO 1 I=1,MY
    1  WRITE (6,99) (Q(I,J),J=1,2)
      WRITE(6,155) (JD(I),I=1,MT)
      SP= ZLEN/MY
        DO 3 I=1,MT
      X= I*SP
        DO 2 J=1,2
       DY(I,J)=0.
        DO 2 K=1,MY
        M=K-1
         IF (LQ(K).EQ.0) GO TO 2
      X1= M*SP
      X2= K*SP
        C=X2-X1
        XD=ZLEN-X1/2.-X2/2.
         XW=Q(K,J)
         XW=XW/E
        R1=XW*XD/ZLEN
       ADD=8.*R1*(X*X-ZLEN**2)*X
       ADD=ADD+XW*X*(8.*XD**3-2.*X2*C*C+C**3)/ZLEN
        IF (X.GT.X2)  GO TO 222
       ADD=ADD+XW*X*2.*C*C
        IF (X.LT.X1) GO TO 223
       ADD=ADD-2.*XW*(X-X1)**4/C
        GO TO 223
  222  ADD=ADD-8.*XW*(X-X1/2.-X2/2.)**3+XW*(2.*X2*C*C-C**3)
  223   DY(I,J)=DY(I,J)-ADD/48.
    2  CONTINUE
        DO 39 J=1,MT
       IF (I.GT.J)  GO TO 224
      B= ZLEN-J*SP
       AF(I,J)=X*B/E*(ZLEN**2-B*B-X*X)/ZLEN/6.
        GO TO 39
  224  AF(I,J)=AF(J,I)
   39  CONTINUE
    3 CONTINUE
C   MINSUB IS DOUBLE PRECISION MATRIX INVERSION
      CALL MINSUB (AF,DAP,MT,DE)
      IF (DE.EQ.0.)  GO TO 225
        DO 4 I=1,MT
         DO 4 J=1,2
        Q(I,J)=0.
        DO 4 K=1,MT
    4   Q(I,J)=Q(I,J)+AF(I,K)*DY(K,J)
        WRITE (6,226)
      DO 900 I=1,MT
  900  WRITE (6,99) (Q(I,L),L=1,2)
      WRITE (6,156)
         DO 10 I=1,MT
        X=FLOAT(I)*SP
        DO 10 J=1,MT
        IF (I.GT.J) GO TO 9
```

```
      B= ZLEN-J*SP
      AC=X*B/E*(ZLEN**2-B*B-X*X)/ZLEN/6.
      AD=X*B/ZLEN/E*2.*(1.+GNU)
      AF(I,J)=AC/XI+AD/AI
      AE(I,J)=AC/XJ+AD/AJ
        GO TO 10
    9  AF(I,J)=AF(J,I)
        AE(I,J)=AE(J,I)
   10  CONTINUE
        DO 20 I=1,MT
        IJ=JD(I)
        A1=A(IJ,1)
        A2=A(IJ,2)
        B1=YI(IJ,1)
        B2=YI(IJ,2)
        Q1=Q(I,1)/Y1
        Q2=Q(I,2)/Y2
        CALL DECO(B1,B2,A1,A2,Y1,Y2,Q1,Q2,XK,XD,1)
        D(I)=XD
          WRITE (6,99) D(I),XK
         DO 18 J=1,MT
         AE(I,J)=AE(I,J)+AF(I,J)
        IF (I.NE.J) GO TO 18
        AE(I,J)=AE(I,J)+XK
   18   CONTINUE
       DO 19 J=1,MT
   19  D(I)=D(I)+AF(I,J)*(Q(J,1)+Q(J,2))
   20  CONTINUE
      WRITE (6,157)
            WRITE (6,99) (D(K),K=1,MT)
C   MINSUB IS DOUBLE PRECISION MATRIX INVERSION
      CALL MINSUB (AE,DAP,MT,DE)
      IF (DE.EQ.0.)  GO TO 225
      GO TO 30
  225   WRITE (6,21)
        STOP
   30        WRITE (6,33) (K,K=1,2)
        DO 40 I=1,MT
        R(I)=0.
         DO 35 J=1,MT
   35   R(I)=R(I)+AE(I,J)*D(J)
        T(I)=Q(I,1)+Q(I,2)-R(I)
      WRITE(7,103) T(I),R(I)
   40  WRITE (6,46) I,T(I),R(I)
   46  FORMAT (I15,2E16.5)
        WRITE (6,62) (K,K=1,2)
      WRITE (6,65)
         XM=0.
         XN=0.
        DO 50 I=1,MT
      XN=XN+R(I)*(1.-FLOAT(I)/FLOAT(MY))
      XM=XM+T(I)*(1.-FLOAT(I)/FLOAT(MY))
   50  CONTINUE
        YB(1)=XM*SP
        YC(1)=XN*SP
        DO 60 I=2,MT
        J=I-1
         XM=XM-T(J)
        XN=XN-R(J)
        YB(I)=YB(J)+XM*SP
```

```
         SB=YB(I)/SM1
        SD=YB(I)/SN1
        YC(I)=YC(J)+XN*SP
         SC=YC(I)/SM2
        SE=YC(I)/SN2
        WRITE (6,64) I,YB(I),SB,SD,YC(I),SC,SE
   60   CONTINUE
       RETURN
   21   FORMAT (//25H  MATRIX SINGULAR          //)
   33   FORMAT (//32H REACTIONS AT THE INTERSECTIONS   2I10//)
   62   FORMAT(//28H BENDING MOMENT AND STRESSES  I10,I20,//)
 64    FORMAT(I4,6E15.4)
 65    FORMAT(12X,'M SHELL     DECK STRESS  BOTTOM STRESS      M BULKHEAD
     1  DECK STRESS  BOTTOM STRESS',//)
  99   FORMAT((7E15.4))
 100   FORMAT(//,'INPUTS FOR THE LONGITUDINAL STRENGTH: PROBLEM NUMBER',
     +I5,//)
  101   FORMAT (//40H LENGTH E GNU AND THE WIDTH OF THE TANKS  //)
  102   FORMAT(//40H AREAS AND  I  OF THE LONG BHDS & SHELLS  //)
 103   FORMAT(2E15.5)
 104   FORMAT(//37H MOMENT OF INERTIA OF THE TRANSVERSES  //6E15.4)
  105   FORMAT (//34H UNIFORM LOADS OF THE    HOLDS        //)
 106   FORMAT(//57H SECTION MODULI OF SHELL AND BULKHEAD AT DECK & BOTTOM
     +     //)
 150   FORMAT(2I3)
 151   FORMAT(7E10.5)
 152   FORMAT(50I1)
 154   FORMAT(2F10.5,I3)
 155   FORMAT(//,'JD(I)=   ',50I2)
 156   FORMAT (// 75H DEFLECTIONS AND INFLUENCE COEFFICIENTS OF SIMPLY SU
     1PPORTED LONG. BULKHEAD  //)
 157   FORMAT (// 31H DEFLECTIONS OF LONG. BULKHEAD  //)
  226   FORMAT(//35H UNIFORM LOADS OF THE TRANSVERSES    //)
 227   FORMAT(//25H AREAS OF THE TRANSVERSES //)
        END



         SUBROUTINE DECO(XI,YI,A1,A2,A,C,Q1,Q2,XK,XD,M)
C      DETIRMINES DEFLECTIONS OF SIMPLY SUPPORTED LONGITUDINALS
C        THIS IS FOR THE LONGITUDINAL STRESSES OF SHIPS.
         DIMENSION T1(5,5),T2(5,5),T(5,5)
         CALL TM(A1,XI,A,Q1,T1,0.,M)
         CALL TM(A2,YI,C,Q2,T2,0.,M)
         CALL MULT (T,T2,T1,5)
        N=1
    1   QQ=T(2,2)*T(4,4)-T(2,4)*T(4,2)
           U=T(2,4)*T(4,5)/QQ-T(2,5)*T(4,4)/QQ
           V=T(2,5)*T(4,2)/QQ-T(2,2)*T(4,5)/QQ
        X=T1(1,2)*U+T1(1,4)*V+T1(1,5)
         GO TO (2,3),N
    2   XD=X
C      DETIRMINES INFLUENCE COEFF. XK
    6   T1(1,5)=0.
         T1(2,5)=0.
         T1(3,5)=0.
         T1(4,5)=-1.
         T2(1,5)=0.
        T2(2,5)=0.
```

```
      T2(3,5)=0.
      T2(4,5)=0.
      CALL MULT (T,T2,T1,5)
       N=N+1
      GO TO 1
    3 XK=X
        RETURN
      END


      SUBROUTINE MINSUB (AA,A,N,DD)
C  MINSUB IS DOUBLE PRECISION MATRIX INVERSION
      DIMENSION AA(50,50),LL(50),M(50),A(N,N)
      DOUBLE PRECISION A,D
C       THIS LOOP SCALES THE MATRIX TO APPROXIMATELY ONE (1)
      L=0
   10 L=L+1
      AHOW=AA(L,L)
       SCALE=ABS(AHOW)
      IF(SCALE.EQ.0.) GO TO 10
      DO 5 I=1,N
      DO 5 J=1,N
    5   A(I,J)=AA(I,J)/SCALE
      CALL MINV (A,N,D,LL,M)
C       THIS LOOP REMOVES SCALING FACTOR
      DO 6 I=1,N
      DO 6 J=1,N
    6  AA(I,J)= A(I,J)/SCALE
      DD=D
      WRITE (6,30) DD,SCALE
   30 FORMAT(//,'MATRIX INVERSION - DETERMINANT IS',E15.5,/,'SCALING FAC
     1TOR IS',E15.5,//)
      RETURN
      END


      SUBROUTINE MINV(A,N,D,L,M)
C
C     ..................................................................
C
C        SUBROUTINE MINV
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
```

```
C          REMARKS
C             MATRIX A MUST BE A GENERAL MATRIX
C
C          SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C             NONE
C
C          METHOD
C             THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C             IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C             THE MATRIX IS SINGULAR.
C
C       ..................................................................
C
      DIMENSION A(1),L(1),M(1)
C
C          ...............................................................
C
C          IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C          C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C          STATEMENT WHICH FOLLOWS.
C
      DOUBLE PRECISION A,D,BIGA,HOLD
C
C          THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C          APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C          ROUTINE.
C
C          THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C          CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C          10 MUST BE CHANGED TO DABS.
C
C          ...............................................................
C
C          SEARCH FOR LARGEST ELEMENT
C
      D=1.0
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C          INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
```

```
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
```

```
C
C         FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END


      SUBROUTINE MULT(D,B,C,M)
      DIMENSION B(M,M),C(M,M),D(M,M)
      DO 1 I=1,M
       DO 1 J=1,M
      D(I,J)=0.
        DO 11 K=1,M
   11  D(I,J)=D(I,J)+B(I,K)*C(K,J)
    1 CONTINUE
      RETURN
       END


      SUBROUTINE TM(A1.XI,A,Q,T,R,M)
C     SEE (EQ C-1) APP. C. CHANG   LINE SOLUTION FOR BEAM
      DIMENSION T(5,5)
       DO 1 I=1,5
       DO 1 J=1,5
    1  T(I,J)=0.
      T(1,1)=1.
      EI=30000000.*XI
      T(1,2)=-A
      T(1,3)=-A*A/2./EI
      T(1,4)=-A**3/6./EI
      T(1,5)=Q*A**4/24./EI
      E=30000000.
      G=E/2./1.3
```

```
      AG=AI*G
      T(4,4)=1.
      T(5,5)=1.
      T(4,5)=-Q*A-R
      T(2,2)=1.
      T(2,3)=A/EI
      T(2,4)=A*A/2./EI
      T(2,5)=-Q*A**3/6./EI
      T(3,5)=-Q*A*A/2.
      T(3,3)=1.
      T(3,4)=A
      IF (M.EQ.0) GO TO 2
      T(1,4)=T(1,4)+A/AG
      T(1,5)=T(1,5)-Q*A*A/2./AG
2     CONTINUE
      RETURN
      END
```

```
C     THIS REPRESENTS A SIMPLE TEST DATA SET TO BE USED AS INPUT DATA FOR
C    LONGITUDINAL PROGRAM.    CHAZAL-PAYNE APRIL 1971

  2  3
333
  0.50E+04  0.40E+04  0.20E+11  0.20E+11  .205E+07  0.30E+00  1.00E+04
 .1  E+0 4  .1 E+04  .15 E+08  .15 E+08  .17 E+08  .17 E+08
   .66E+03  .17 E+04  .38 E+04
  .706E+03  .22 E+04  .39 E+04
   .5 E+10  .6  E+10  .6  E+10
   .5 E+10  .6  E+10  .6  E+10
   00
            100000.   1
            100000.   1
   00
```

```
$COPY *SOURCE* -F
$RUN -LOAD#+-F MAP

 ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..

 ENTRY = 500608  SIZE = 010E86

NAME      VALUE  T RF    NAME      VALUE  T RF    NAME      VALUE  T RF

GETSPACE 2140AE *        FREESPAC 2143AE *        ERROR#    21C10C *
MTS#     21C128 *        CANREPLY 21E9D6 *        GDINFO    21EA2A *
POINT    21F028 *        SCARDS#  21F7A8 *        SPRINT#   21F7B8 *
SPUNCH#  21F7C8 *        SERCOM#  21F7D8 *        READ#     21F848 *
WRITE#   21F85A *        LCSYMBOL 2204B0 *        MINSUB    5000D8  5000D8
MAIN     500608  500608  MINV     500710  500710  LONGAL    503000  503000
TM       50F000  50F000  DECO     50F398  50F398  MULT      50F890  50F890
REWIND#  50FB80 *50FB80  IBCOM#   510000 *510000  ADCON#    512000 *512000
FCVZO    512154 *        FCVAO    5121FA *        FCVLO     512282 *
FCVIO    5125A8 *        FCVEO    512A9A *        FCVCO     512CAC *
FIOCS#   5130A0 *5130A0

 ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ... ..
```

EXECUTION BEGINS

INPUTS FOR THE LONGITUDINAL STRENGTH: PROBLEM NUMBER 2

LENGTH E GNU AND THE WIDTH OF THE TANKS

0.1000E 05 0.2050E 07 0.3000E 00 0.1000E 00 0.4001E 07

AREAS AND I OF THE LONG BHDS & SHELLS

0.5000E 04 0.4000E 04 0.2000E 11 0.2000E 11

SECTION MODULI OF SHELL AND BULKHEAD AT DECK & BOTTOM

0.1500E 08 0.1700E 08 0.1500E 08 0.1700E 08

AREAS OF THE TRANSVERSES

0.6600E 03 0.1700E 04 0.3800E 04 0.7060E 03 0.2200E 04

MOMENT OF INERTIA OF THE TRANSVERSES

0.5000E 10 0.6000E 10 0.6000E 10 0.5000E 10 0.6000E 10

	M SHELL	DECK STRESS	BOTTOM STRESS	M BULKHEAD	DECK STRESS
2	-0.1611E 09	-0.1074E 02	-0.9479E 01	0.5629E 09	0.3753E 02
3	0.3946E 08	0.2630E 01	0.2321E 01	0.2195E 09	0.1463E 02

STOP 0
EXECUTION TERMINATED

UNIFORM LOADS OF THE HOLDS

0.0	0.0
0.0	0.1000E 06
0.0	0.1000E 06
0.0	0.0

JD(I)= 3 3 3

MATRIX INVERSION - DETERMINANT IS 0.38410E-01
SCALING FACTOR IS 0.57165E 04

UNIFORM LOADS OF THE TRANSVERSES

0.0	0.4643E 05
0.0	0.1143E 06
0.0	0.4643E 05

DEFLECTIONS AND INFLUENCE COEFFICIENTS OF SIMPLY SUPPORTED LONG. BULKHEAD

0.1376E 00	0.2281E-11
0.3388E 00	0.2281E-11
0.1376E 00	0.2281E-11

DEFLECTIONS OF LONG. BULKHEAD

0.2668E 00 0.5312E 00 0.2668E 00

MATRIX INVERSION - DETERMINANT IS 0.37617E 00
SCALING FACTOR IS 0.16418E-05

REACTIONS AT THE INTERSECTIONS 1 2

0.96025E 05	-0.49596E 05	
1	0.96025E 05	-0.49596E 05
-0.16048E 06	0.27477E 06	
2	-0.16048E 06	0.27477E 06
0.96023E 05	-0.49593E 05	
3	0.96023E 05	-0.49593E 05

BENDING MOMENT AND STRESSES 1 2

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C                                                                      C
C     TRANSVERSE STRENGTH ANALYSIS                                     C
C                                                                      C
C                                                                      C
C     WRITTEN BY PIN YU CHANG                                          C
C     ADAPTED FOR MTS BY                                               C
C          E. A. CHAZAL                                                C
C          J. M. PAYNE                                                 C
C                                                                      C
C                                                                      C
C     THIS PROGRAM ANALYZES THE ENTIRE SHIP STRUCTURE MODELED AS A     C
O     3-D GRILLAGE.  THE GOVERNING EQUATIONS ARE UNCOUPLED VIA         C
C     COORDINATE TRANSFOMATION.  PHYSICALLY THIS ISOLATES ANY          C
C     PARTICULAR TRANSVERSE AND PRODUCES THE NECESSARY BOUNDARY        C
C     CONDITIONS TO ANALYZE THAT TRANSVERSE USING FINITE ELEMENTS.     C
C     FOR DETAILED INFORMATION SEE PROFESSIONAL THESIS OF CHAZAL       C
C     AND PAYNE, APRIL 1971.                                           C
C                                                                      C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
        CALL TRANCO(IB)
        STOP
        END



      SUBROUTINE  TRANCO(IB)
C...PRIMARY TRANSVERSE STRENGTH ROUTINE
      DIMENSION JT(5),LO(100),IND(50,3)
      DIMENSION  AP(50,50),BP(50,50),CP(50,50)
      DIMENSION DP(50,50),BL(5,100),BM(5,100)
      DIMENSION DOX(50,100),DOY(50,100),CX(100),CY(100)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION R(2100),DMMMY(7979)
     REAL*8  DAP(50,50),DAF(50,50),DEIG(50)
      COMMON K1,K2,K3,K4,ID,NORO,NN1,
    1       UNITS ,ND     ,NONO   ,N1     ,IPO    ,PIO ,
    2 NUMEO,R,DMMMY
      COMMON/INFLU/AF(50,50),EIG(50),DY(50),NFILE,NF,LNOO(100),
    1 LROW(100),DX(50),MF
      COMMON/SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),NOTR,
    1 ZTR(50),ZLEN,P(50),XI,XA,NSEC
       COMMON/SAFE/BP,CP,DP
     COMMON /MATRL/E,G,GNU,ALPHA,CONVF
     COMMON /WORK/DUMY(1298)
       EQUIVALENCE (DOX(1,1),R(1)),(DOY(1,1),BP(1,1))
     NSCR=9
     NFILE=7
     NCARD=8
C   JPROB IS THE PROBLEM NUMBER SELECTED AT YOUR CONVENIENCE
     READ(6,104) JPROB
     CALL BEGIN(IB,JT)
     WRITE(6,702)
     READ(5,104)NF
     WRITE(6,104)NF
     CALL TRV(NCARD)
     CALL LONGI(NFILE)
      DO 701 I=1,IB
     IF(JT(I)-NOTR) 701,701,700
```

```
701   CONTINUE
         GO TO 777
 700   WRITE (6,707)
          STOP
 777  PF=0.
      DO    110   I=1,NOTR
        PF=PF+ AF(I,I)/FLOAT(NOTR)
 110    P(I)=SQRT(P(I))
      DO    112   I=1,NOTR
      DO 112 J=1,NOTR
      IF (I.GT.J) GO TO 111
       AP(I,J)=AF(I,J)*P(I)*P(J)/PF
            GO TO 112
 111   AP(I,J)=AP(J,I)
112   DP(I,J)=AP(I,J)*PF
      CALL JESSIN (AP,EIG,NOTR,DAP,DAF,DEIG,BP)
      DO 670 J=1,NOTR
      WRITE (6,1964)
      WRITE(6,671) EIG(J)
670   WRITE(6,671)(AP(I,J),I=1,NOTR)
671   FORMAT((7E15.6))
 236    DO 113       I=1,NOTR
 113      EIG(I)=EIG(I)*PF
      CALL TRANS(BP,AP,50,NOTR)
        CALL MULT (CP,DP,AP,50,NOTR)
        CALL MULT(DP,BP,CP,50,NOTR)
         WRITE (6,114)
       WRITE (6,105)(EIG(I),I=1,NOTR)
      WRITE(6,315)
       WRITE (6,105)(DP(I,I),I=1,NOTR)
      NNN=NF
       DO   115 I=1,NNN
             PE=EIG(I)
            PD=(PE-DP(I,I))
             PE=PE*.01
            PD=ABS(PD)
        IF (PD.GT.PE)   GO TO 116
 115   CONTINUE
        GO TO 118
 116   WRITE (6,117)
         GO TO 1000
C     INPUT THE LOADS FROM THE LONGITUDINAL BULKHEADS AND SHELLS
 118          NUM=NOTR
      CALL LOADS(NFILE,NSCR)
      WRITE(6,181)
      READ(5,1001) IDENTO
      WRITE(6,1002) IDENTO
      IF(IDENTO.EQ.0) GO TO 1000
      CALL REWIND (7)
        IF (NOTR.GT.NF)   NUM=NF
      DO 130 I=1,NOLO
      CALL NFILRE(NROW,NODE,LOADC,SFX(I),SFY(I),PHI(I),DX,DY,NOTF)
        ANG=PHI(I)
       DO 129   J=1,NUM
       CX(J)=0.
       CY(J)=0.
        IF (LOADC.EQ.0)   GO TO 149
        DO 127 K=1,NOTR
       CX(J)=CX(J)+AP(K,J)*DX(K)*P(K)
 127   CY(J)=CY(J)+AP(K,J)*DY(K)*P(K)
```

```
      BL(J,I)=SFX(I)*CX(J)*COS(ANG)-SFY(I)*CY(J)*SIN(ANG)
      BM(J,I)=SFX(I)*CX(J)*SIN(ANG)+SFY(I)*CY(J)*COS(ANG)
      GO TO 129
  149 BL(J,I)=0.
      BM(J,I)=0.
  129 CONTINUE
      LNOD(I)=NODE
      LROW(I)=NROW
  130 CONTINUE
      CALL SHIP1(NCARD,MACO)
      DO 2001 L=1,NOLO
      KR=LROW(L)
      LO(L)=LNOD(L)-NONO(KR)
      DO 2001 J=1,KR
 2001 LO(L)=LO(L)+NONO(J)
      NTO=0
      WRITE (6,109) (LO(I),I=1,NOLO)
      DO 201 N=1,NORO
  201 NTO=NTO+NONO(N)*2
      DO 304 NL=1,NUM
      SPRING=1./EIG(NL)
      DO 202 L=1,NTO
  202 R(L)=0.
      DO 203 L=1,NOLO
      J=(LO(L)-1)*2
      J1=J+1
      J2=J+2
      R(J1)=BL(NL,L)/EIG(NL)
  203 R(J2)=BM(NL,L)/EIG(NL)
      CALL SHP5(SPRING,1,MACO)
      DO 205 L=1,NOLO
      J=(LO(L)-1)*2
      J1=J+1
      J2=J+2
      BL(NL,L)=R(J1)
  205 BM(NL,L)=R(J2)
      KJ =0
      CALL REWIND(3)
  304 CONTINUE
      CALL MATINS(AF,50,NOTR,CP,50,0,DE,ID,IND)
      IF (ID.EQ.1) GO TO 60
      WRITE (6,307) ID
      GO TO 1000
   60 CALL REWIND(7)
      DO 405 J=1,NOLO
      CALL NFILRE(NROW,NODE,LOADC,SFX(J),SFY(J),PHI(J),DX,DY,NOTE)
      AN=PHI(J)
      DO 401 I=1,NOTR
      CX(I)=0.
      CY(I)=0.
      DO 402 K=1,NUM
      CX(I)=CX(I)+AP(I,K)*BL(K,J)/P(I)
  402 CY(I)=CY(I)+AP(I,K)*BM(K,J)/P(I)
  401 CONTINUE
      WRITE (6,107) J
      WRITE(6,250)
      WRITE (6,105) (CX(M),M=1,NOTR,2)
      WRITE(6,251)
      WRITE (6,105) (CY(M),M=1,NOTR,2)
      DO 403 I=1,NOTR
```

```
          DOX(I,J)=0.
          DOY(I,J)=0.
          DO 403 K=1,NOTR
          DOX(I,J)=DOX(I,J)+AF(I,K)*(DX(K)-CX(K)*COS(AN)-CY(K)*SIN
     1 (AN))*SFX(J)
 403      DOY(I,J)=DOY(I,J)+AF(I,K)*(DY(K)+CX(K)*SIN(AN)-CY(K)*COS
     1 (AN))*SFY(J)
 405      CONTINUE
          WRITE (6,420)
          DO 404 I=1,NOTR
         WRITE (6,105) (DOX(I,N),N=1,NOLO)
 404     WRITE (6,105) (DOY(I,N),N=1,NOLO)
          WRITE (6,420)
          DO 430 I=1,IB
            L=JT(I)
          DO 431 J=1,NOLO
           AN=PHI(J)
           BL(I,J)=DOX(L,J)*COS(AN)-DOY(L,J)*SIN(AN)
          BM(I,J)=DOX(L,J)*SIN(AN)+DOY(L,J)*COS(AN)
 431      CONTINUE
         WRITE (6,106) L
         WRITE(6,500)
         WRITE (6,105) (BL(I,J),J=1,NOLO)
         WRITE(6,501)
         WRITE (6,105) (BM(I,J),J=1,NOLO)
 430     CONTINUE
          DO 450 I=1,IB
           IF (I.EQ.1)  GO TO 441
         CALL SHIP1(NCARD,MACO)
441       DO 435 M=1,NTO
 435      R(M)=0.
          DO 440 M=1,NOLO
           AN=PHI(M)
          J=(LO(M)-1)*2
          J1=J+1
          J2=J+2
          R(J1)=BL(I,M)
 440      R(J2)=BM(I,M)
        CALL SHP5(0.,0,MACO)
         ID=JPROB
          CALL SHIP4
 450     CONTINUE
        RETURN
1000     STOP
1       FORMAT   (3I5,5F12.5)
 100     FORMAT(//20H TRANSFORMED FORCES    ,2I5)
 101     FORMAT (10F7.2)
104     FORMAT (I5)
105     FORMAT((7E15.5))
 106     FORMAT(//16H TRANSVERSE NO.   I10 //)
107     FORMAT(//,'DEFLECTION OF LONGITUDINAL',I5,'   AT EVERY OTHER TRANSV
     +ERSE')
 109     FORMAT (10I5)
114     FORMAT(/10X,'SCALED EIGENVALUES')
 117 FORMAT (10X,39H EIGENVALUE ERROR CHECK PROGRAM PLEASE//)
 181 FORMAT(//,'ENTER 0 TO STOP PROGRAM HERE.  ENTER 1 TO GO ON')
250     FORMAT(//,4X,'X  DEFLECTIONS',//)
251     FORMAT(//,4X,'Y DEFLECTIONS',//)
 307     FORMAT (//20H MATRIX SINGULAR      I5)
315     FORMAT(//,10X,'DIAGONAL OF MATRIX DP')
```

```
 420  FORMAT(// 33H REAL LOADS UPON THE TRANSVERSES      //)
 500  FORMAT('  X-FORCES  TYPICAL    ')
 501  FORMAT('  Y-FORCES  TYPICAL    ')
 702  FORMAT(/' NUMBER OF EIGENVALUES TO BE USED')
 707  FORMAT (//25H  INPUT ERRORS IN TRANCO   //)
1964  FORMAT(//' ......')
1001  FORMAT(I1)
1002  FORMAT(//,'IDENTO VALUE IS',I5,//)
      END


      SUBROUTINE BEGIN(IB,JT)
C...ROUTINE INPUT BASIC SHIP PARAMETERS
      DIMENSION JT(5)
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      COMMON /SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),
     +NOTR,ZTR(50),ZLEN,P(50),XI,XA,NSEC
      WRITE(6,100)
      WRITE(6,237)
      READ(5,300)CONVF
      WRITE(6,300)CONVF
      WRITE(6,10)
      READ(5,300)ZLEN
      WRITE(6,300)ZLEN
      ZLEN=ZLEN*CONVF
      WRITE(6,12)
      READ(5,301)NOTR
      WRITE(6,301)NOTR
      YSUM=0.
      Y=ZLEN/(NOTR+1)
      DO 25 K=1,NOTR
      ZTR(K)=YSUM+Y
   25 YSUM=YSUM+Y
      WRITE(6,20)
      NSEC=1
      READ(5,302)XI,XA
      WRITE(6,302)XI,XA
      XI=XI*(CONVF**4)
      XA=XA*(CONVF**2)
      WRITE(6,213)
      READ(5,302)E,GNU
      WRITE(6,302)E,GNU
      ALPHA=0.
      G=E/2./(1.+GNU)
      WRITE(6,18)
       DO 9  I=1,NOTR
      READ (5,300) P(I)
    9 WRITE(6,300) P(I)
      WRITE(6,14)
      READ (5,301) IB
      WRITE(6,301) IB
      WRITE(6,16)
      DO 8  I=1,IB
      READ (5,301)  JT(I)
    8 WRITE(6,301)  JT(I)
      RETURN
   10 FORMAT(/,' LENGTH OF LONGITUDINALS')
   12 FORMAT(/,' NO. TRANSVERSES ALONG LENGTH')
```

```
   14 FORMAT(/,' NO. TRANSVERSES TO BE ANALYZED(5)')
   16 FORMAT(/,' LIST TRANSVERSES TO BE ANALYZED BY POSITION FROM',
     +' STERN')
   18 FORMAT(/,' LIST STIFFNESS FACTORS OF ALL TRANSVERSES'
     +' IN ORDER FROM STERN')
   20 FORMAT(/,' STANDARD LONGITUDINAL'/,' MOMENT OF INERTIA,'
     +' SHEAR AREA')
  100 FORMAT(1H1,' TRANSVERSE STRENGTH ANALYSIS OF LONGITUDIN'
     +'ALLY FRAMED SHIPS',/,60('*')//)
  213 FORMAT(/,' YOUNGS MODULUS, POISSONS RATIO')
  237 FORMAT(/' CONVERSION FACTOR TO BE APPLIED TO ALL'
     +,' DIMENSIONAL DATA',/,
     +' INCLUDING COORDINATES, PLATE THICKNESS, BAR AREA',/,
     +' BUT NOT INCLUDING YOUNGS MODULUS')
  300 FORMAT  (F15.5)
  301 FORMAT(I5)
  302 FORMAT  (E15.5,F15.5)
      END


      SUBROUTINE COMSI(TM,SI)
C...ROUTINE TO COMPUTE INITIAL PARAMETERS OF BEAM MEMBER
      DIMENSION TM(5,5),SI(5,1)
      DEL=TM(1,2)*TM(3,4)-TM(3,2)*TM(1,4)
      SI(1,1)=0.
      SI(3,1)=0.
      SI(5,1)=1.
      SI(2,1)=(TM(3,5)*TM(1,4)-TM(1,5)*TM(3,4))/DEL
      SI(4,1)=(TM(3,2)*TM(1,5)-TM(1,2)*TM(3,5))/DEL
      RETURN
      END


      SUBROUTINE DIRCOS
C     DIRECTION COSINE SUBROUTINE FOR PLATE
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      COMMON  K1,K2   , K3      , K4      , ID      , NORO    , NN1
      COMMON  UNITS   , ND      , NONO    , N1      , IPO     , PIO
      COMMON  X       , Y       , Z       , E       , E1      , GNU
      COMMON  GNU1    , MEMNO   , MEMTYP  , IEGNU   , IESF    , IEI
      COMMON  IEJ     , IEK     , IEL     , INI     , JNI     , INJ
      COMMON  JNJ     , INK     , JNK     , INL     , JNL     , P1
      COMMON  P2      , P3      , P4      , P5      , P6      , XJ
      COMMON  YK      , XL      , YL      , DC      , SK      , DI
      COMMON  AI      , AJ      , AK      , AL      , SKAI    , SKAJ
      COMMON  SKAK    , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON  IZ      , NC      , XK
      X1=X(INJ,JNJ)-X(INI,JNI)
      X2=Y(INJ,JNJ)-Y(INI,JNI)
      R1=X(INK,JNK)-X(INI,JNI)
      R2=Y(INK,JNK)-Y(INI,JNI)
      XJ=SQRT(X1*X1+X2*X2)
```

```
      DC(1,1)=X1/XJ
      DC(1,2)=X2/XJ
        DC(2,1)=-DC(1,2)
        DC(2,2)=DC(1,1)
      XK=R1*DC(1,1)+R2*DC(1,2)
      YK=R1*DC(2,1)+R2*DC(2,2)
      RETURN
      END


      SUBROUTINE INFO(I)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      COMMON  K1,K2   , K3      , K4      , ID      , NORO    , NN1
      COMMON  UNITS   , ND      , NONO    , N1      , IPO     , PIO
      COMMON  X       , Y       , Z       , E       , E1      , GNU
      COMMON  GNU1    , MEMNO   , MEMTYP  , IEGNU   , IFSF    , IFI
      COMMON  IFJ     , IFK     , IFL     , INI     , JNI     , INJ
      COMMON  JNJ     , INK     , JNK     , INL     , JNL     , P1
      COMMON  P2      , P3      , P4      , P5      , P6      , XJ
      COMMON  YK      , XL      , YL      , DC      , SK      , DI
      COMMON  AI      , AJ      , AK      , AL      , SKAI    , SKAJ
      COMMON  SKAK    , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON  IZ      , NC      ,DUMMY(7101)
      COMMON/INTO/ IJ(14,500),TA(500)
      MEMNO=IJ(1,I)
      MEMTYP=IJ(2,I)
      IFI=IJ(3,I)
      IFJ=IJ(4,I)
      IFK=IJ(5 ,I)
      IFL=IJ(6 ,I)
      INI=IJ(7 ,I)
      JNI=IJ(8 ,I)
      INJ=IJ(9 ,I)
      JNJ=IJ(10,I)
      INK=IJ(11,I)
      JNK=IJ(12,I)
      INL=IJ(13,I)
      JNL=IJ(14,I)
      P1=TA(I)
      WRITE(6,101) MEMNO,MEMTYP,IFI,IFJ,IFK,IFL,INI,JNI,INJ,JNJ,INK,JNK,
     +INL,JNL,P1
      RETURN
  101 FORMAT(1H ,I4,13I3,F13.5)
      END


      SUBROUTINE INPUT
      DIMENSION UNITS(4),ND(6),NONO(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40)  ,GNU1(4),DC(2,2),NOB(25),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3SKA2(6,6,4),DUMY(7093),XI(6)
```

```
      COMMON K1,MEMTO,NOMAT,MOB,ID,NORO,NN1
      COMMON UNITS,ND,NONO,NOB,IPO,PIO
      COMMON X,Y,Z,E,NOBB,MCOM,NOBO,MCON,GNU
      COMMON GNU1,MEMNO,MEMTYP,FA,IFS,IFI
      COMMON  IFJ     , IFK     , IFL     , INI     , JNI     , INJ
      COMMON  JNJ     , INK     , JNK     , INL     , JNL     , P1
      COMMON  P2      , P3      , P4      , P5      , P6      , XJ
      COMMON  YK      , XL      , YL      , DC      , SK      , DI
      COMMON  AI      , AJ      , AK      , AL      , SKAI    , SKAJ
      COMMON  SKAK    , SKAL    , A1      , A2      , SKA1    , SKA2
      COMMON IZ,NC,DUMY,ITEMP,ALPHA,XI
      WRITE(6,103)
      COMMON/INTO/ IJ(14,500),TA(500)
      READ(8,105,ERR=200) ID,NORO,NN1,NOMAT,ITEMP,(UNITS(I),I=1,4)
      READ(8,100,ERR=200) NOBO,MCON,NOBB,MCOM
      READ(8,100,ERR=200) (NOB(I),I=1,NOBO)
      READ(8,100,ERR=200) MOB
      READ(8,106,ERR=200) FA
      READ(8,106,ERR=200) E,GNU,ALPHA
      READ(8 ,    100,ERR=200)(NONO(I),I=1,NORO)
      DO 1 I=1,NORO
      JOE=NONO(I)
      DO 1 J=1,JOE
    1 READ(8     ,101,ERR=200)X(I,J),Y(I,J)
      IFSF=0
      I=0
    4 I=I+1
      IFS=IFSF
      READ(8,107,ERR=200) IJ(1,I),IJ(2,I),MEMTO,IFSF,IJ(3,I),IJ(4,I),IJ(
     +5,I),IJ(6,I),IJ(7,I),IJ(8,I),IJ(9,I),IJ(10,I),IJ(11,I),IJ(12,I),
     +IJ(13,I),IJ(14,I),TA(I),P2,P3,P4,P5,P6
      IF(IJ(2,I)) 3,3,4
    3 MEMTO=I-1
      DO 2 I=1,NORO
      JOE=NONO(I)
      DO 2 J=1,JOE
    2 WRITE (6,102)I,J,X(I,J),Y(I,J)
      GO TO 223
  200  WRITE (6,221)
         STOP
  223   WRITE (6,104)
      RETURN
  100 FORMAT(25I3)
  101 FORMAT(3F10.2)
  102 FORMAT(1H ,I6,I7, 3F20.5)
  103 FORMAT(18H1NODE  COORDINATES//14H     ROW    NODE,13X,7HX-COORD,
     113X,7HY-COORD//1H )
  104 FORMAT (13H1MEMBER  DATA//
     144H0    M   M   I   I   I   I   I   J   I   J   I   J   I   J.
     211X,2HP1/
     344H     E   E   F   F   F   F   N   N   N   N   N   N   N   N/
     444H     M   M   I   J   K   L   I   I   J   J   K   K   L   L/
     5 8H     N   T/
     6 8H     O   Y/
     7 8H         P)
 105  FORMAT (5I5,4A6)
 106  FORMAT(E10.2,F7.2,E10.2)
 107  FORMAT(I3,7I1,8I2,6E15.7)
  221  FORMAT (//25H INPUT ERROR IN NODEIN      //)
      END
```

```
      SUBROUTINE JESSIN (A,EIG,N,B,C,E,D)
      DIMENSION A(50,50),EIG(50),B(N,N),C(N,N),E(N),D(50,50)
      REAL*8  B,C,E
      DO 1 J=1,N
      DO 1 I=1,N
    1 B(I,J)=A(I,J)
      CALL JESS (N,N,B,N,C,E,&999)
      GO TO 2
  999 WRITE (6,10)
   10 FORMAT ('EIGENVALUES ARE AT BEST WRONG')
      STOP
    2 CONTINUE
      DO 3 J=1,N
      EIG(J)=E(J)
      DO 3 I=1,N
    3 A(I,J)=C(I,J)
      CALL REORD(EIG,50,N,A,D)
      RETURN
      END


      SUBROUTINE K2RE(IRO,ICOL1,ICOL2,BK,ID,A,MACO)
C  THIS ROUTINE RETRIEVES DATA FROM STORAGE MATRICES IN A CERTAIN ORDER
      DIMENSION BK(84,84),A(MACO,MACO)
       COMMON/K23RW/KH(3,49),KHB(4,49),KJ,KJB,IFD,LJ(4)
      INTEGER*2 LEN
      LEN=MACO*MACO*4
      IRO  =KHB(1,KJB)
      ICOL1=KHB(2,KJB)
      ICOL2=KHB(3,KJB)
      IF (ID) 4,5,6
    4 IH=ICOL1
      JH=ICOL2
      GO TO 7
    5 IH=ICOL1
      JH=ICOL1
      GO TO 7
    6 IH=ICOL2
      JH=ICOL1
    7 CONTINUE
      CALL READ(A,LEN,0,M,IFD,&999)
      DO 1 J=1,JH
      DO 1 I=1,IH
    1 BK(I,J)=A(I,J)
      KJB=KJB-1
      IF(KJB.EQ.0) GO TO 8
      LJ(1)=KHB(4,KJB)
      CALL POINT(IFD,LJ,1,&999,&999)
      GO TO 9
    8 DO 2 K=1,3
    2 LJ(K)=0
      CALL POINT(IFD,LJ,7,&999,&999)
    9 CONTINUE
      RETURN
```

```
  999   WRITE(6,70)KJB
        STOP
   70   FORMAT('ERROR IN READING FILE -A AT LINE',I5)
        END


        SUBROUTINE K2WR(IRO,ICOL1,ICOL2,BK,ID,A,MACO)
C    THIS ROUTINE TRANSFERS DATA TO STORAGE MATRICES IN A CERTAIN ORDER
        DIMENSION BK(84,84),A(MACO,MACO)
         COMMON/K23RW/KH(3,49),KHB(4,49),KJ,KJB,IFD,LJ(4)
        INTEGER*2 LEN
        LEN=MACO*MACO*4
        KJB=KJB+1
        CALL NOTE(IFD,LJ)
        KHB(1,KJB)=IRO
        KHB(2,KJB)=ICOL1
        KHB(3,KJB)=ICOL2
        KHB(4,KJB)=LJ(3)
        IF (ID) 4,5,6
    4   IH=ICOL1
        JH=ICOL2
        GO TO 7
    5   IH=ICOL1
        JH=ICOL1
        GO TO 7
    6   IH=ICOL2
        JH=ICOL1
    7   CONTINUE
        DO 1 J=1,JH
        DO 1 I=1,IH
    1   A(I,J)=BK(I,J)
        CALL WRITE(A,LEN,0,M,IFD,&999)
        RETURN
  999   WRITE(6,70)KJB
        STOP
   70   FORMAT('ERROR IN WRITING FILE -A AT LINE',I5)
        END


        SUBROUTINE K3RE(IRO,ICOL1,ICOL2,BK,ID,A,MACO)
C    THIS ROUTINE RETRIEVES DATA FROM STORAGE MATRICES IN A CERTAIN ORDER
        DIMENSION BK(84,84),A(MACO,MACO)
        COMMON/K23RW/KH(3,49),KHB(4,49),KJ,KJB,IFD,LJ(4)
        INTEGER*2 LEN
        LEN=MACO*MACO*4
        KJ =KJ +1
        IRO  =KH (1,KJ )
        ICOL1=KH (2,KJ )
        ICOL2=KH (3,KJ )
        IF (ID) 4,5,6
    4   IH=ICOL1
        JH=ICOL2
        GO TO 7
    5   IH=ICOL1
        JH=ICOL1
        GO TO 7
```

```
 6    IH=ICOL2
      JH=ICOL1
 7    CONTINUE
      CALL READ(A,LEN,0,M,3,&999)
      DO 1 J=1,JH
      DO 1 I=1,IH
1     BK(I,J)=A(I,J)
      RETURN
999   WRITE(6,70)KJ
      STOP
70    FORMAT('ERROR IN READING FILE3 AT LINE',I5)
      END
```

```
      SUBROUTINE K3WR(IRO,ICOL1,ICOL2,BK,ID,A,MACO)
C   THIS ROUTINE TRANSFERS DATA TO STORAGE MATRICES IN A CERTAIN ORDER
      DIMENSION BK(84,84),A(MACO,MACO)
      COMMON/K23RW/KH(3,49),KHB(4,49),KJ,KJB,IFD,LJ(4)
      INTEGER*2 LEN
      LEN=MACO*MACO*4
      KJ =KJ +1
C   MAIN PROGRAM FOR TRANSVERSE STRENGTH
      KH(1,KJ)=IRO
      KH(2,KJ)=ICOL1
      KH(3,KJ)=ICOL2
      IF (ID) 4,5,6
    4 IH=ICOL1
      JH=ICOL2
      GO TO 7
    5 IH=ICOL1
      JH=ICOL1
      GO TO 7
    6 IH=ICOL2
      JH=ICOL1
    7 CONTINUE
      DO 1 J=1,JH
      DO 1 I=1,IH
    1 A(I,J)=BK(I,J)
       CALL WRITE(A,LEN,0,M,3,&999)
      RETURN
  999 WRITE(6,70)KJ
      STOP
   70 FORMAT('ERROR IN WRITING FILE3 AT LINE',I5)
      END



      SUBROUTINE LOADS(NFILE,NSCR)
C...ROUTINE TO INPUT LOADING CONDITION
      INTEGER DEFIN
      COMMON /WORK/XC(42),YC(28),NONO(25),NXC,NYC,
     +DEFIN,NODE(40,25),LROW(100),LNOD(100)
      COMMON /MATRL/E,G,GNU,ALPHA,CONVE
      COMMON /SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),
     +NOTR,ZTR(50),ZLEN,P(50),XI,XA,NSEC
      COMMON /SAFE/LOADC(100),DX(50),DY(50),DBASE(50),
     +ZP(50),PC(50),ZQ(20),Q(20),ZPL(100)
      COMMON /SAFE/TM(5,5),TR(5,5),SI(5,1),SX(5,1),ODATA(8,50)
      COMMON /SAFE/ZI(20),EYE(20)
      COMMON /SAFE/DXL(50),DYL(50)
      COMMON /SAFE/B(50),T(50,2),ST(50,25,2),A(2),R(50,2),DMMY(3659)
      COMMON /INFLU/AF(50,50) ,DMMMY(353)
      ZI(1)=0.
      EYE(1)=XI
      NSEC=1
   50 NREC=0
      NMISC=0
      IERR=0
      WRITE(6,100)
```

```
      WRITE(6,102)
      WRITE(6,103)
      WRITE(6,104)
      WRITE(6,105)
      DO 150 L=1,NOLO
      LOADC(L)=0
  150 CONTINUE
C
      WRITE(6,155)
      READ(5,300)NSETS
      WRITE(6,300)NSETS
      IF(NSETS)9998,9998,156
C
  156 DO 1000  NST=1,NSETS
      IF(NMISC)270,157,165
  157 WRITE(6,160)NST
      WRITE(6,162)
      READ (5,1) NP,NQ,NDIR
      WRITE (6,1) NP,NQ,NDIR
  165 IF(NP)180,180,170
  170 WRITE(6,171)
      K=2*NP
      DO 20 IL=1,K
      READ (5,2) ZPL(IL)
   20  WRITE (6,2) ZPL(IL)
      DO 175  IL=1,NP
      PC(IL)=ZPL(2*IL)
      II=2*(IL-1)+1
      ZP(IL)=ZPL(II)*CONVF
  175 CONTINUE
  180 IF(NQ)270,270,182
  182 WRITE(6,183)
      K=2*NQ
      DO 21 IL=1,K
      READ(5,2) ZPL(IL)
   21  WRITE(6,2)ZPL(IL)
      DO 185  IL=1,NQ
      Q(IL)=ZPL(2*IL)/CONVF
      II=2*(IL-1)+1
      ZQ(IL)=ZPL(II)*CONVF
  185 CONTINUE
      GO TO 270
C......................
C BEGIN READING IN LONGITUDINALS LOADED FOR SET
C......................
  200 WRITE(6,201)
  205 READ(5,3)  SFL,JR01,JR02,ICOL1,ICOL2
      CALL SWTCH(JR01,JR02)
      CALL SWTCH(ICOL1,ICOL2)
      J=JR01
      I=ICOL1
      WRITE(6,3)  SFL,JR01,JR02,ICOL1,ICOL2
C
      IF(JR01)1000,1000,210
  210 IF(JR01-JR02)215,220,215
C--VERTICAL ARRAY OF LONGITUDINALS
  215 LDIR=1
      GO TO 225
C--HORIZONTAL ARRAY OF LONGITUDINALS
  220 LDIR=2
```

```
  225   IF(NODE(I,J)-DEFIN)800,230,800
  230   N=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
C--CHECK IF NODE GIVEN HAS BEEN DEFINED AS A LONGITUDINAL
        DO 250  L=1,NOLO
        IF(J-LROW(L))250,240,250
  240   IF(N-LNOD(L))250,260,250
  250   CONTINUE
        IERR=1
        WRITE(6,252)J,I
        IF(NMISC)1040,800,1000
  260   LOADC(L)=1
        NREC=NREC+1
        DO 265   K=1,NOTR
        GO TO (262,264),NDIR
  262   DX(K)=DBASE(K)*SFL/SFX(L)
        DY(K)=0.
        GO TO 265
  264   DY(K)=DBASE(K)*SFL/SFY(L)
        DX(K)=0.
  265   CONTINUE
                CALL NSCRWR(J,N,DX,DY,NOTR)
        IF(NMISC)1040,800,1000
C----------------             NSCR=9
C----------------
C THIS SECTION COMPUTES DEFLECTIONS FOR BASIC LONGITUDINAL
  270   SFA=0.
        SFL=1.0
        SFS=1.0
        CALL SETOC(NSEC,ZI,EYE,NO,ZO,O,NP,ZP,PC,ZLEN,NOCC,ODATA)
        NCON=0
        CALL TMATT(XB,NB,ZLEN,TM,NOCC,NCON,ODATA,SFS,SFL,SFA,E,G)
        CALL COMSI(TM,SI)
        NCON=0
        DO 500 K=1,NOTR
        CALL TMATT(XB,NB,ZTR(K),TM,NOCC,NCON,ODATA,SFS,SFL,SFA,E,G)
        CALL MMULT(TM,SI,SX,5,5,1)
        DBASE(K)=SX(1,1)
  500   CONTINUE
C-------------
C-------------
        IF(NMISC)230,545,230
  545   IF(NST-1)550,550,200
  550   WRITE(6,551)
        DO 22 K=1,NOTR
   22   WRITE(6,4) K,DBASE(K)
        GO TO 200
  800   GO TO (810,820),LDIR
  810   J=J+1
        IF(J-JRO2)225,225,205
  820   I=I+1
        IF(I-ICOL2)225,225,205
 1000   CONTINUE
        IF(NMISC)1040,1001,1005
 1001   NMISC=1
        WRITE(6,1002)
        WRITE(6,1003)
 1005   READ(5,5)  NP,NO,NDIR,J,I
        WRITE(6,5)  NP,NO,NDIR,J,I
        NSETS=1
        IF(J)1020,1020,156
```

```
1020  NMISC=-1
      NSETS=1
      SFL=1.0
      NP=NOTR
      NO=0
      WRITE(6,1021)
      WRITE(6,1022)
      READ(5,6)  IBHD,MNODE
      WRITE(6,6)  IBHD,MNODE
      WRITE(6,1023)
      A(1)=0.
      A(2)=0.
      DO 1025  J=1,MNODE
      READ(5,7)  B(J),(T(J,NLMEM),NLMEM=1,2)
      B(J)=B(J)*CONVF
      T(J,1)=T(J,1)*CONVF
      T(J,2)=T(J,2)*CONVF
      WRITE(6,7)  B(J),(T(J,NLMEM),NLMEM=1,2)
      A(1)=A(1)+T(J,1)*B(J)
      A(2)=A(2)+T(J,2)*B(J)
1025  CONTINUE
      WRITE(6,1027)
      DO 1036  K=1,NOTR
      READ(5,8)  (R(K,NLMEM),NLMEM=1,2)
      WRITE(6,8)  (R(K,NLMEM),NLMEM=1,2)
      DO 1035  NLMEM=1,2
      S=-R(K,NLMEM)/A(NLMEM)
      DO 1033  J=1,MNODE
1033  ST(K,J,NLMEM)=S*T(J,NLMEM)*B(J)
1035  CONTINUE
1036  CONTINUE
      DO 1038 K=1,NOTR
1038  DX(K)=0.
      J=2
      NLMEM=2
      NDIR=2
      I=IBHD
      SFL=1.
1037  N=NODEI(I,NODE,DEFIN,J,NXC,NONO(J))
      DO 720  L=1,NOLO
      IF(J-LROW(L))720,710,720
 710  IF(N-LNOD(L))720,730,720
 720  CONTINUE
      IERR=1
      WRITE(6,252)J,I
      GO TO 1040
 730  NREC=NREC+1
      DO 735 K=1,NOTR
      DY(K)=0.
      DO 734  M=1,NOTR
      DY(K)=DY(K)+AF(K,M)*ST(M,J,NLMEM)/SFY(L)
 734  CONTINUE
 735  CONTINUE
            CALL NSCRWR(J,N,DX,DY,NOTR)
      GO TO 1040
1040  J=J+1
      IF(J-MNODE)1037,1037,1041
1041  I=1
      MNODE=MNODE-1
      NLMEM=NLMEM-1
```

```
      J=2
      IF(NLMEM)1045,1045,1037
 1045 CONTINUE
      IF(IERR)1048,1048,50
 1048 CALL REWIND (7)
      DO 2000  L=1,NOLO
      DO 1050  K=1,NOTR
      DXL(K)=0.
      DYL(K)=0.
 1050 CONTINUE
      CALL REWIND (9)
      DO 1500  NR=1,NREC
              CALL NSCRRE(J,N,DX,DY,NOTR)
      IF(J-LROW(L))1500,1100,1500
 1100 IF(N-LNOD(L))1500,1200,1500
 1200 DO 1250  K=1,NOTR
      DXL(K)=DXL(K)+DX(K)
      DYL(K)=DYL(K)+DY(K)
 1250 CONTINUE
 1500 CONTINUE
      CALL NFILWR(LROW(L),LNOD(L),LOADC(L),SFX(L),SFY(L),PHI(L),DXL,DYL,
     +NOTR )
 2000 CONTINUE
      CALL REWIND (7)
 9999 RETURN
 9998 STOP
    1 FORMAT (3I5)
    2 FORMAT (E15.5)
    3 FORMAT (F15.5,4I5)
    4 FORMAT (I5,E15.5)
    5 FORMAT (5I5)
    6 FORMAT (2I5)
    7 FORMAT (3F15.5)
    8 FORMAT (2E15.5)
   30 FORMAT (2I5,2E15.5)
   31 FORMAT (3I5,5E12.5)
  100 FORMAT(1H1,' LOADING CONDITION')
  102 FORMAT(/' 1. A LOAD SET IS A SET OF LOADS ACTING IN'
     +' A GIVEN',/,' X OR Y DIRECTION. THE EXTENT OF THESE'
     +' LOADS IS ALONG THE',/,' LENGTH OF A GIVEN LONGITUD'
     +'INAL')
  103 FORMAT(/' 2. ANY LONGITUDINAL MAY BE LOADED WITH ANY'
     +' NUMBER OF LOAD',/,' SETS, WHICH MAY BE ONLY PART'
     +'IALLY APPLIED VIA',/,' A PROPORTIONAL FACTOR')
  104 FORMAT(/' 3. LOCATIONS OF LOADS ARE DISTANCES MEASURED'
     +' FROM THE',/,' STERN')
  105 FORMAT(/,' 4. LOAD SET DIRECTION CODES ARE',/,
     +4X,'=1, X-DIRECTION',/,4X'=2, Y-DIRECTION')
  155 FORMAT(/,' NUMBER OF LOAD SETS')
  160 FORMAT(1H1,'---LOAD SET'I5)
  162 FORMAT(' NO. CONC. LOADS, NO. UNIF. LOADS, DIRECTION '
     +'CODE')
  171 FORMAT(/,' LIST LOCATION, P')
  183 FORMAT(/,' LIST START LOCATION, Q')
  201 FORMAT(/,' LIST LONGITUDINALS SO LOADED',/,
     +' FACTOR, ROW1, ROW2, COL1, COL2')
  252 FORMAT(' **ERROR-NODE ON ROW'I4,', COL'I4,' HAS NOT'
     +' BEEN DEFINED AS A LONGITUDINAL')
  300 FORMAT (I5)
  551 FORMAT(/,' COMPUTED DEFLECTIONS AT TRANSVERSES')
```

```
 740   FORMAT(1H1,' LONGITUDINAL DEFLECTIONS DUE TO LOADS')
 741   FORMAT(/,' ROW',I3,' NODE',I3)
 742   FORMAT(10X,'X-DEFLECTIONS')
 743   FORMAT(10X,'Y-DEFLECTIONS')
1002   FORMAT(/' MISC. LOADINGS')
1003   FORMAT(' NP,NQ, NDIR, ROW, COLUMN')
1021   FORMAT(1H1,' SHEAR LOADS ON TRANSVERSES')
1022   FORMAT(/,' BHD COLUMN NUMBER, NUMBER OF ROWS FOR SHEAR')
1023   FORMAT(/,' WEB LENGTH, SHELL, BHD THICKNESS (CM)')
1027   FORMAT(1H1,' SHELL, BHD SHEARS PER TRANSVERSE STARTING'
      +' FROM STERN')
       END


       SUBROUTINE LONGI(NFILE)
C...ROUTINE INPUTS LONGITUDINAL DATA
       COMMON /MATRL/E,G,GNU,ALPHA,CONVF
       COMMON /SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),
      +NOTR,ZTR(50),ZLEN,P(50),XI,XA,NSEC
       INTEGER DEFIN
       COMMON /WORK/XC(42),YC(28),NONO(25),NXC,NYC,
      +DEFIN,NODE(40,25),LROW(100),LNOD(100)
       COMMON /INFLU/AF(50,50),EIG(50),DMMY(303)
       COMMON /SAFE/NN(25),NCR(51),DUMMY(7424)
       IERR=0
       WRITE(6,276)
       WRITE(6,250)
       WRITE(6,251)
       LOMAX=100
       WRITE(6,278)
       NOLO=0
2000   READ(5,300)XIX,XIY,AX,JROW,ICOL1,ICOL2
       CALL SWTCH(ICOL1,ICOL2)
C--HORIZ.SEQUENCE TERMINATES WITH ZERO ROW NUMBER..............
       WRITE(6,300)XIX,XIY,AX,JROW,ICOL1,ICOL2
       IF(JROW)2020,2020,2005
2005   IF(XIX)2007,2007,2006
2006   IF(XIY)2007,2007,2010
2007   IERR=1
       WRITE(6,2008)
       GO TO 2000
2010   DO 2015   I=ICOL1,ICOL2
       IF(NOLO-LOMAX)2011,2016,2016
2011   NOLO=NOLO+1
       N=NODET(I,NODE,DEFIN,JROW,NXC,NONO(JROW))
       LROW(NOLO)=JROW
       LNOD(NOLO)=N
       CALL NOD(NONO,N,JROW,LNO(NOLO))
       SFX(NOLO)=XIX*(CONVF**4)/XI
       SFY(NOLO)=XIY*(CONVF**4)/XI
       PHI(NOLO)=0.
2015   CONTINUE
       GO TO 2000
2016   WRITE(6,288)LOMAX
       IERR=1
       GO TO 2000
2020   WRITE(6,279)
2030   READ(5,300)XIX,XIY,AX,ICOL,JRO1,JRO2
```

```
      CALL SWTCH(JRO1,JRO2)
C--VERTICAL SEQUENCE TERMINATES WITH ZERO COLUMN NUMBER.........
      WRITE(6,300)XIX,XIY,AX,ICOL,JRO1,JRO2
      IF(ICOL)2050,2050,2035
2035  IF(XIX)2037,2037,2036
2036  IF(XIY)2037,2037,2040
2037  IERR=1
      WRITE(6,2008)
      GO TO 2030
2040  DO 2045   J=JRO1,JRO2
      IF(NOLO-LOMAX)2041,2047,2047
2041  NOLO=NOLO+1
      N=NODE1(ICOL,NODE,DEFIN,J,NXC,NONO(J))
      LROW(NOLO)=J
      LNOD(NOLO)=N
      CALL NOD(NONO,N,J,LNO(NOLO))
      SFX(NOLO)=XIX*(CONVF**4)/XI
      SFY(NOLO)=XIY*(CONVF**4)/XI
      PHI(NOLO)=0.
2045  CONTINUE
      GO TO 2030
2047  WRITE(6,288)LOMAX
      IERR=1
      GO TO 2030
2050  WRITE(6,287)NOLO
      DO 3000   J=1,NYC
3000  NN(J)=J
      WRITE(6,3001)
      WRITE(6,3002)(NN(J),J=1,NYC)
      DO 3050  I=1,NXC
      DO 3045   J=1,NYC
      NCR(J)=9999
      IF(NODE(I,J)-DEFIN)3045,3010,3045
3010  NCR(J)=0
      N=NODE1(I,NODE,DEFIN,J,NXC,NONO(J))
      DO 3040   L=1,NOLO
      IF(LROW(L)-J)3040,3015,3040
3015  IF(LNOD(L)-N)3040,3020,3040
3020  NCR(J)=LNO(L)
      GO TO 3045
3040  CONTINUE
3045  CONTINUE
      WRITE(6,3047)(NCR(J),J=1,NYC)
3050  CONTINUE
      IF(IERR)2060,2060,2055
2055  STOP
2060  N=NOTR+1
      EI=E*XI
        Y=ZLEN/FLOAT(N)
        DO 2  I=1,NOTR
        DO 2  J=1,NOTR
        IF  (I.GT.J)   GO TO 1
        A=Y*FLOAT(J)
        X=Y*FLOAT(I)
       B=ZLEN-A
      GK=0.
      IF(XA.NE.0.)GK=X*B/XA/G/ZLEN
       AF(I,J)=B*X/6./EI*(ZLEN*ZLEN-B*B-X*X)/ZLEN+GK
        GO TO 2
    1  AF(I,J)=AF(J,I)
```

```
    2 CONTINUE
      RETURN
  250 FORMAT(/' IX = MOMENT OF INERTIA OF LONGIT-L BENDING IN'
     +' X-DIRECTION OF TRANSVERSE')
  251 FORMAT(' IY = MOMENT OF INERTIA OF LONGIT-L BENDING IN'
     +' Y-DIRECTION OF TRANSVERSE')
  276 FORMAT(1H1,//' DEFINITIONS OF LONGITUDINALS')
  278 FORMAT(/,' LIST BY HORIZONTAL SEQUENCE'/
     +'   IX,IY,A, ROW, COL1, COL2')
 2008 FORMAT(' **ERROR-MOMENT OF INERTIA FOR ABOVE LONGITL'
     +' NOT DEFINED')
  279 FORMAT(/,' LIST BY VERTICAL SEQUENCE'/
     +'   IX,IY,A, COLUMN, ROW1,ROW2')
  287 FORMAT(/,' THERE ARE A TOTAL OF',I6,' LONGITUDINALS')
  288 FORMAT(' **MAX. LONGITUDINALS ='I8)
  300 FORMAT(2E15.6,F15.6,3I5)
 3001 FORMAT(1H1,' LONGITUDINAL NUMBERING SYSTEM')
 3002 FORMAT(//,1X,'C',/,1X,'O',5X,'ROW',/,1X,'L',25I3,/)
 3047 FORMAT(/,2X,25I3)
      END


      FUNCTION MAXCOL(NN1,NONO,NORO)
C  ORDERS COLUMN INTEGER VECTOR IN DECSENDING ORDER AND  MULTIPLY MAX VALUE
      DIMENSION NONO(NORO),H(25)
      DO 1 I=1,NORO
    1 H(I)=NONO(I)
      N=NORO-1
   10 DO 20 I=1,N
      IF(H(I)-H(I+1))30,20,20
   20 CONTINUE
       GO TO 50
   30  NH=H(I)
      H(I)=H(I+1)
      H(I+1)=NH
      GO TO 10
   50 MAXCOL=H(1)*NN1
      RETURN
      END


       SUBROUTINE MATINS(AA,JJ,N,CP,JK,M,DD,ID,INDEX)
      REAL*8 A(84,84)
C THIS SUBROUTINE WAS NEEDED TO MAKE PROGRAM COMPATIBLE TO AN IBM SSP MATRIX
C  AA IS MATRIX TO BE INVERTED   CP,INDEX ARE NOT NECESSARY IN SUBROUTINE
      DIMENSION AA(JJ,JJ),CP(JJ,JK),INDEX(JJ,JK)
      CALL MINSUB (AA,A,N,DD,JJ)
      IF(DD) 1,2,1
    2 ID=2
      GO TO 3
    1 ID=1
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MEM1
C     TRIANGULAR PLATE SUBMATRIX SUBROUTINE
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084),IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6),
     1 XI(6)
      COMMON  K1,K2   , K3     , K4     , ID     , NORU   , NN1
      COMMON  UNITS   , ND     , NONO   , N1     , IPO    , PIO
      COMMON  X       , Y      , Z      , E      , E1     , GNU
      COMMON  GNU1    , MEMNO  , MEMTYP , IEGNU  , IESF   , IFI
      COMMON  IFJ     , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ     , INK    , JNK    , INL    , JNL    , P1
      COMMON  P2      , P3     , P4     , P5     , P6     , XJ
      COMMON  YK      , XL     , YL     , DC     , SK     , DI
      COMMON  AI      , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK    , SKAL   , A1     , A2     , SKA1   , SKA2
      COMMON  IZ      , NC     , XK     , NOMEM  , ICOUNT , BK
      COMMON  IM      , JM     , NA1    , NA2    , ZAI    , ZAJ
      COMMON  ZAK     , ZAL    , ITEMP  , ALPHA  , XI
      CALL DIRCOS
      CA=(1.0-GNU)/2.0
      CB=E*P1/(2.0*(1.0-GNU*GNU)*XJ*YK)
      SK(1,1)=CB*(YK*YK+CA*XK*XK)
      SK(1,2)=-CB*CA*XJ*XK
      SK(1,3)=CB*GNU*XJ*YK
      SK(2,1)=SK(1,2)
      SK(2,2)=CB*CA*XJ*XJ
      SK(2,3)=0.0
      SK(3,1)=SK(1,3)
      SK(3,2)=0.0
      SK(3,3)=CB*XJ*XJ
      CA=YK/XJ
      CB=XK/XJ
      CC=CB-1.0
      DO 9 I = 1,6
      DO 9 J =1,6
       AI(I,J)=0.
      AJ(I,J) = 0.0
    9 AK(I,J) = 0.0
      DO 1 I=1,3
      AI(1,I)=-DC(1,I)
      AJ(1,I)=DC(1,I)
      AK(1,I)=0.0
      AI(2,I)=-DC(1,I)-CA*DC(2,I)
      AJ(2,I)=CA*DC(2,I)
      AK(2,I)=DC(1,I)
      AI(3,I)=CC*DC(2,I)
      AJ(3,I)=-CB*DC(2,I)
    1 AK(3,I)=DC(2,I)
      IF(NN1-3) 4,5,5
    4 NC=NN1
      GO TO 6
    5 NC=3
    6 CALL MULTRD(AI,INI,JNI,SKAI)
      CALL MULTRD(AJ,INJ,JNJ,SKAJ)
```

```
      CALL MULTRD(AK,INK,JNK,SKAK)
      IF(IFSF) 3,3,2
    2 CA=E/(1.0-GNU*GNU)
      CB=E/((1.0+GNU)*(2.0*YK))
      DI(1,1)=CA/XJ
      DI(1,2)=0.0
      DI(1,3)=CA*GNU/YK
      DI(2,1)=CA*GNU/XJ
      DI(2,2)=0.0
      DI(2,3)=CA/YK
      DI(3,1)=-CB*XK/XJ
      DI(3,2)=CB
      DI(3,3)=0.0
      DO 7 I=1,3
      DO 7 J=1,3
    7 SKAL(I,J)=SK(I,J)
    3 IF(ITEMP) 8,8,10
   10 XI(1)=XJ*ALPHA
      XI(2)=XK*ALPHA
      XI(3)=YK*ALPHA
      CALL TEMPCO(         NC,IZ,SKAI,XI,ZAI)
      CALL TEMPCO(         NC,IZ,SKAJ,XI,ZAJ)
      CALL TEMPCO(         NC,IZ,SKAK,XI,ZAK)
    8 RETURN
      END


      SUBROUTINE MEM2
C     QUADRILATERAL PLATE SUBMATRIX SUBROUTINE
      DIMENSION F(6,6),INDEX(6,3)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084),IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6),
     1 XI(6)
      COMMON  K1,K2   , K3      , K4     , IO     , NORO   , PNI
      COMMON  UNITS   , ND      , NONO   , N1     , IPO    , PIO
      COMMON  X       , Y       , Z      , F      , E1     , GNU
      COMMON  GNU1    , MEMNO   , MEMTYP , IFGNU  , IFSF   , IFI
      COMMON  IFJ     , IFK     , IFL    , INI    , JNI    , INJ
      COMMON  JNJ     , INK     , JNK    , INL    , JNL    , P1
      COMMON  P2      , P3      , P4     , P5     , P6     , XJ
      COMMON  YK      , XL      , YL     , DC     , SK     , DI
      COMMON  AI      , AJ      , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK    , SKAL    , A1     , A2     , SKA1   , SKA2
      COMMON  IZ      , NC      , XK     , NOMEM  , ICOUNT , BK
      COMMON  IM      , JM      , NA1    , NA2    , ZAI    , ZAJ
      COMMON  ZAK     , ZAL     , ITEMP  , ALPHA  , XI
      CALL DIRCOS
      XXL=X(INL,JNL)-X(INI,JNI)
      YYL=Y(INL,JNL)-Y(INI,JNI)
      XL=XXL*DC(1,1)+YYL*DC(1,2)
      YL=XXL*DC(2,1)+YYL*DC(2,2)
      F(1,1)=XJ
      F(1,2)=0.0
      F(1,3)=-GNU*XJ
```

```
      F(1,4)=.5*XJ*F(1,3)
      F(1,5)=0.0
      F(2,1)=XK
      F(2,2)=XK*YK-.5*XJ*YK
      F(2,3)=-GNU*XK
      F(2,4)=-.5*(GNU*XK*XK+YK*YK)
      F(2,5)=2.0*(1.0+GNU)*YK
      F(3,1)=-GNU*YK
      F(3,2)=.5*(XK*XJ-XK*XK-GNU*YK*YK)
      F(3,3)=YK
      F(3,4)=XK*YK
      F(3,5)=0.0
      F(4,1)=XL
      F(4,2)=XL*YL-.5*XJ*YL
      F(4,3)=-GNU*XL
      F(4,4)=-.5*(GNU*XL*XL+YL*YL)
      F(4,5)=2.0*(1.0+GNU)*YL
      F(5,1)=-GNU*YL
      F(5,2)=.5*(XL*XJ-XL*XL-GNU*YL*YL)
      F(5,3)=YL
      F(5,4)=XL*YL
      F(5,5)=0.0
      CALL MATINS(F,6,5,DI,6,0,DD,M,INDEX)
      IF(M-1) 13,13,12
   12 WRITE (6,100)MEMNO
  100 FORMAT(31H SOMETHING WRONG WITH MEMBER    ,I5,12H  TOUGH LUCK)
      STOP
   13 BA=XL
      HA=YL
      BB=XL-XJ
      HB=YL
      BC=XL-XK
      HC=YL-YK
      BD=XK
      HD=YL-YK
      BE=XK
      HE=YK
      XBA=.5*BA
      YBA=.5*HA
      XBB=XL-BB/3.0
      YBB=HB/3.0
      XBC=XK+BC/3.0
      YBC=YL-HC/3.0
      XBD=.5*XK
      YBD=YK+.5*HD
      XBE=BE/3.0
      YBE=.666666666*HE
      AA=BA*HA
      AB=.5*BB*HB
      AC=.5*BC*HC
      AD=BD*HD
      AE=.5*BE*HE
      A=AA-AB-AC-AD-AE
      XM=AA*YBA-AB*YBB-AC*YBC-AD*YBD-AE*YBE
      YM=AA*XBA-AB*XBB-AC*XBC-AD*XBD-AE*XBE
      XIA=AA*HA*HA/3.0
      XIB=AB*(HB*HB/18.0+YBB*YBB)
      XIC=AC*(HC*HC/18.0+YBC*YBC)
      XID=AD*(HD*HD/12.0+YBD*YBD)
      XIE=AE*(HE*HE/18.0+YBE*YBE)
```

```
      XO=XIA-XIB-XIC-XID-XIE
      YIA=AA*BA*BA/3.0
      YIB=AB*(BB*BB/18.0+XBB*XBB)
      YIC=AC*(BC*BC/18.0+XBC*XBC)
      YID=AD*(BD*BD/12.0+XBD*XBD)
      YIE=AE*(BE*BE/18.0+XBE*XBE)
      YI=YIA-YIB-YIC-YID-YIE
      AJ(1,1)=AA*XBA*YBA
      BJ=AB*(AB/18.0+XBB*YBB)
      CJ=AC*(AC/18.0+XBC*YBC)
      DJ=AD*XBD*YBD
      EJ=AE*(AE/18.0+XBE*YBE)
      XYJ=AJ(1,1)-BJ-CJ-DJ-EJ
      CA=E*P1
      CB=-CA*GNU
      DI(1,1)=CA*A
      DI(1,2)=CA*XM
      DI(1,3)=CB*A
      DI(1,4)=CB*YM
      DI(1,5)=0.0
      DI(2,2)=CA*XO
      DI(2,3)=CB*XM
      DI(2,4)=CB*XYJ
      DI(2,5)=0.0
      DI(3,3)=DI(1,1)
      DI(3,4)=CA*YM
      DI(3,5)=0.0
      DI(4,4)=CA*YI
      DI(4,5)=0.0
      DI(5,5)=CA*2.0*(1.0+GNU)*A
      DO 3 I=2,5
      JOE=I-1
      DO 3 J=1,JOE
    3 DI(I,J)=DI(J,I)
      DO 9 I=1,5
      DO 9 J=1,5
      AI(I,J)=0.0
      DO 9 K=1,5
    9 AI(I,J)=AI(I,J)+DI(I,K)*F(K,J)
      DO 10 I=1,5
      DO 10 J=1,5
      SK(I,J)=0.0
      DO 10 K=1,5
   10 SK(I,J)=SK(I,J)+F(K,I)*AI(K,J)
      CA=YK/XJ
      CB=XK/XJ-1.0
      CC=YL/XJ
      CD=XL/XJ-1.0
      CE=-XK/XJ
      CF=-XL/XJ
      IF(NN1-3) 14,15,15
   14 NC=NN1
      GO TO 16
   15 NC=3
   16 DO 17 I = 1,6
      DO 17 J = 1,6
      AI(I,J) = 0.0
      AJ(I,J) = 0.0
      AK(I,J) = 0.0
   17 AL(I,J) = 0.0
```

```
      DO 5 J=1,NC
      AI(1,J)=-DC(1,J)
      AI(2,J)=-DC(1,J)-CA*DC(2,J)
      AI(3,J)=CB*DC(2,J)
      AI(4,J)=-DC(1,J)-CC*DC(2,J)
      AI(5,J)=CD*DC(2,J)
      AJ(1,J)=DC(1,J)
      AJ(2,J)=CA*DC(2,J)
      AJ(3,J)=CE*DC(2,J)
      AJ(4,J)=CC*DC(2,J)
      AJ(5,J)=CF*DC(2,J)
      AK(2,J)=DC(1,J)
      AK(3,J)=DC(2,J)
      AL(4,J)=DC(1,J)
    5 AL(5,J)=DC(2,J)
      CALL MULTRD(AI,INI,JNI,SKAI)
      CALL MULTRD(AJ,INJ,JNJ,SKAJ)
      CALL MULTRD(AK,INK,JNK,SKAK)
      CALL MULTRD(AL,INL,JNL,SKAL)
      IF(ITEMP) 19,19,20
   20 XI(1)=XJ*ALPHA
      XI(2)=XK*ALPHA
      XI(3)=YK*ALPHA
      XI(4)=XL*ALPHA
      XI(5)=YL*ALPHA
      CALL TEMPCO(         NC,IZ,SKAI,XI,ZAI)
      CALL TEMPCO(         NC,IZ,SKAJ,XI,ZAJ)
      CALL TEMPCO(         NC,IZ,SKAK,XI,ZAK)
      CALL TEMPCO(         NC,IZ,SKAL,XI,ZAL)
   19 CONTINUE
      IF(IFSF) 8,8,7
    7 DO 6 I=1,5
      DO 6 J=1,5
      SKAL(I,J)=SK(I,J)
    6 DI(I,J)=E*F(I,J)
    8 RETURN
      END


      SUBROUTINE MEM5
C     PIN-ENDED BAR SUBMATRIX SUBROUTINE  P1=BAR CROSS SECTION AREA
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40),E1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084),IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6),
     1 XI(6)
      COMMON  K1,K2   , K3      , K4     , ID     , NORO   , NN1
      COMMON  UNITS   , ND      , NONO   , N1     , IPO    , PIO
      COMMON  X       , Y       , Z      , E      , E1     , GNU
      COMMON  GNU1    , MEMNO   , MEMTYP , IEGNU  , IFSF   , IFI
      COMMON  IFJ     , IFK     , IFL    , INI    , JNI    , INJ
      COMMON  JNJ     , INK     , JNK    , INL    , JNL    , P1
      COMMON  P2      , P3      , P4     , P5     , P6     , XJ
      COMMON  YK      , XL      , YL     , DC     , SK     , DI
      COMMON  AI      , AJ      , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK    , SKAL    , A1     , A2     , SKA1   , SKA2
```

```
      COMMON  IZ      , NC      , XK      , NOMEM   , ICOUNT , BK
      COMMON  IM      , JM      , NA1     , NA2     , ZAI    , ZAJ
      COMMON  ZAK     , ZAL     , ITEMP   , ALPHA   , XI
       X1=X(INJ,JNJ)-X(INI,JNI)
       X2=Y(INJ,JNJ)-Y(INI,JNI)
       XJ=SQRT(X1*X1+X2*X2)
        DC(1,1)=X1/XJ
        DC(1,2)=X2/XJ
       DC(2,1)=-DC(1,2)
       DC(2,2)=DC(1,1)
    3 SKK=P1*E /XJ
      SKAL(1,1)=SKK
      II=NN1*(JNI-1)
      IJ=NN1*(JNJ-1)
      IF(NN1-3) 4,5,5
    4 NC=NN1
      GO TO 6
    5 NC=3
    6 DO 12 I=1,6
      DO 12 J=1,NN1
      AI (I,J)=0.0
      AJ (I,J)=0.0
      SKAI(I,J)=0.0
   12 SKAJ(I,J)=0.0
      DO 9 JJ=1,NC
      A1(1,JJ,1)=-DC(1,JJ)
      SKA1(1,JJ,1)=-DC(1,JJ)*SKK
      AI(1,JJ)=-DC(1,JJ)
      SKAI(1,JJ)=SKA1(1,JJ,1)
      NA1=1
      IM(1)=II
      IF(INJ-INI) 7,7,8
    7 A1(1,JJ,2)=DC(1,JJ)
      SKA1(1,JJ,2)=SKK*DC(1,JJ)
      NA1=2
      IM(2)=IJ
      AJ(1,JJ)=DC(1,JJ)
      SKAJ(1,JJ)=SKA1(1,JJ,2)
      GO TO 9
    8 A2(1,JJ,1)=DC(1,JJ)
      SKA2(1,JJ,1)=SKK*DC(1,JJ)
      NA2=1
      JM(1)=IJ
      AJ(1,JJ)=DC(1,JJ)
      SKAJ(1,JJ)=SKA2(1,JJ,1)
    9 CONTINUE
      IF(IFSF) 11,11,10
   10 DI(1,1)=P1
   11 IF(ITEMP) 13,13,15
   15 XI(1)=XJ*ALPHA
      CALL TEMPCO(      NC,IZ,SKAI,XI,ZAI)
      CALL TEMPCO(      NC,IZ,SKAJ,XI,ZAJ)
   13 RETURN
      END


      SUBROUTINE MEMB1
C     BRANCH DISPLACEMENTS AND STRESSES FOR TRIANG PLATE  SHIP 4
```

```
      DIMENSION STRESS(6)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION V(2100),UU(6),OO(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2   , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS   , ND     , NONO   , N1     , IPO    , PIO
      COMMON  NUMFO   , V      , KK     , KKK    , III    , IZ
      COMMON  UU      , OO     , MEMNO  , MEMTYP , INI    , JNI
      COMMON  INJ     , JNJ    , INK    , JNK    , INL    , JNL
      COMMON  IFSF    , IFI    , IFJ    , IFK    , IFL    , AI
      COMMON  AJ      , AK     , AL     , SKAI   , SKAJ   , SKAK
      COMMON  SKAL    , DI     , VTEE
      DO 1 I=1,3
    1 UU(I)=0.0
      CALL SR14(KK,3,JNI,AI,UU,III,KKK,INI,INI)
      CALL SR14(KK,3,JNJ,AJ,UU,III,KKK,INI,INJ)
      CALL SR14(KK,3,JNK,AK,UU,III,KKK,INI,INK)
      IF(IPO(1)) 11,11,9
    9 IT=III
      DO 10 I=1,3
   10 UU(I)=UU(I)-VTEE(I,IT)
   11 CONTINUE
      IF(IFSF-2) 7,2,2
    2 CALL SR4A(DI,3,3,UU,STRESS)
      TA=(STRESS(1)+STRESS(2))/2.0
      TB=(STRESS(1)-STRESS(2))/2.0
      TC=SQRT(TB*TB+STRESS(3)*STRESS(3))
      PA=TA+TC
      PB=TA-TC
      ANGLE=28.6478*ATAN(STRESS(3)/TB)
      IF(TB) 3,6,6
    3 IF(STRESS(3)) 4,5,5
    4 ANGLE=ANGLE-90.0
      GO TO 6
    5 ANGLE=ANGLE+90.0
    6 WRITE (6,100)III,INI,MEMNO,(STRESS(I),I=1,3),PA,PB,ANGLE
      IF(IFSF-3) 7,8,8
    7 CALL SR4A(SKAL,3,3,UU,OO)
    8 RETURN
  100 FORMAT(1H0,I9,I10,16H     TRIANG PLATE,I6,4X,5E14.6,F10.4,4H DEG)
      END


      SUBROUTINE MEMB2
C     BRANCH DEFORMATIONS AND STRESSES FOR QUAD PLATE SHIP 4
      DIMENSION STRESS(6)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION V(2100),UU(6),OO(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2   , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS   , ND     , NONO   , N1     , IPO    , PIO
      COMMON  NUMFO   , V      , KK     , KKK    , III    , IZ
      COMMON  UU      , OO     , MEMNO  , MEMTYP , INI    , JNI
      COMMON  INJ     , JNJ    , INK    , JNK    , INL    , JNL
      COMMON  IFSF    , IFI    , IFJ    , IFK    , IFL    , AI
      COMMON  AJ      , AK     , AL     , SKAI   , SKAJ   , SKAK
```

```
110   S=1./GA
120   T(1,2)=-X
      T(1,3)=-X*X/2./EI
      T(2,3)=+X/EI
      T(1,4)=-X*X*X/6./EI + S*X
      T(2,4)=+X*X/2./EI
      T(3,4)=+X
      IF(INFLU)150,150,999
150   Q=PROP(3)*SFL
      RQ=PROP(4)
      T(1,5)=-X*X*(Q*(-X*X/24./EI+S/2.)+RQ*X*(-X*X/120./EI+S/6.))
      T(2,5)=-X*X*X*(Q+RQ*X/4.)/6./EI
      T(3,5)=-X*X*(Q+RQ*X/3.)/2.
      T(4,5)=-X*(Q+RQ*X/2.)
      GO TO (999,200),IDF
200   CALL IDENT(TC,5)
      CF=PROP(5)*SFL
      TC(4,5)=-CF
      CALL MMULT(T,TC,TR,5,5,5)
      CALL EQUAL(T,TR,5,5)
999   RETURN
      END
```

```
      SUBROUTINE READIN(A,B,IZ,NC,K)
      DIMENSION A(6,6),B(6,6,4)
      DO 1 I=1,IZ
      DO 1 J=1,NC
    1 B(I,J,K)=A(I,J)
      RETURN
      END
```

```
      SUBROUTINE  REORD     (ELAM,N,NTV,BP,BL)
C...ROUTINE ARRANGES EIGENVALUES IN DESCENDING ORDER WITH
C...THE CORRESPONDING RE-ARRANGING OF THE EIGENVECTORS
       DIMENSION ELAM(N),BP(N,N),BL(N,N)
       NTV1=NTV-1
   10   DO 20 I=1,NTV1
       IF (ELAM(I)-ELAM(I+1)) 30,20,20
   20  CONTINUE
      GO TO 50
   30   SAVE=ELAM(I)
      ELAM(I)=ELAM(I+1)
       ELAM(I+1)=SAVE
      DO 40 J=1,NTV
       BL(J,I)=BP(J,I)
      BP(J,I)=BP(J,I+1)
      BP(J,I+1)=BL(J,I)
   40  CONTINUE
       GO TO 10
   50  WRITE (6,100)
      RETURN
  100  FORMAT (// 30H  EIGENVALUES AND EIGENVECTORS   //)
      END
```

```
      COMMON  SKAL     , DI      , VTEE
      DO 1  I=1,5
    1 UU(I)=0.0
      CALL SR14(KK,5,JNI,AI,UU,III,KKK,INI,INI)
      CALL SR14(KK,5,JNJ,AJ,UU,III,KKK,INI,INJ)
      CALL SR14(KK,5,JNK,AK,UU,III,KKK,INI,INK)
      CALL SR14(KK,5,JNL,AL,UU,III,KKK,INI,INL)
      IF(IPO(1)) 11,11,19
 19   IT=III
      DO 10  I=1,5
 10   UU(I)=UU(I)-VTEE(I,IT)
 11   CONTINUE
      IF(IFSF-2) 3,2,2
 2    CALL SR4A (DI,5,5,UU,STRESS)
       WRITE(6,100)III,INI,MEMNO,(STRESS(I),I=1,5)
      IF(IFSF-3)3,4,4
 3    CALL SR4A(SKAL,5,5,UU,QQ)
 4    RETURN
 100  FORMAT(1H0,I9,I10,14H     QUAD PLATE,I8,4X,5E14.6)
      END
```

```
       SUBROUTINE MEMB5
C    BRANCH FORCES AND STRESS FOR PIN ENDED BAR
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION V(2100),UU(6),QQ(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2   , K3      , K4      , ID      , NORO    , NN1
      COMMON  UNITS   , ND      , NONO    , N1      , IPO     , PIO
      COMMON  NUMFO   , V       , KK      , KKK     , III     , IZ
      COMMON  UU      , QQ      , MEMNO   , MEMTYP  , INI     , JNI
      COMMON  INJ     , JNJ     , INK     , JNK     , INL     , JNL
      COMMON  IFSF    , IFI     , IFJ     , IFK     , IFL     , AI
      COMMON  AJ      , AK      , AL      , SKAI    , SKAJ    , SKAK
      COMMON  SKAL    , DI      , VTEE
       DO 1  I=1,IZ
    1 UU(I)=0.0
      CALL SR14(KK,IZ,JNI,AI,UU,III,KKK,INI,INI)
      CALL SR14(KK,IZ,JNJ,AJ,UU,III,KKK,INI,INJ)
      IF(IPO(1)) 11,11,19
 19   IT=III
      UU(1)=UU(1)-VTEE(1,IT)
 11   CONTINUE
       QQ(1)=SKAL(1,1)*UU(1)
      IF(IFSF-2) 3,2,2
  2   STRESS=QQ(1)/DI(1,1)
      WRITE(6,100)III,INI,MEMNO,STRESS
 3    RETURN
 100  FORMAT(1H0,I9,I10,7H     BAR,I15,4X,E14.6)
      END
```

```
      SUBROUTINE MULTRD(AA,IN,JN,SKA)
C     PREMULTIPLIES AA BY SK THEN READS AA INTO A1 OR A2 AND
C     SKA INTO SKA1 OR SKA2
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION X(25,40),Y(25,40),F1(4),GNU1(4),DC(2,2),
     1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
     2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
     3 SKA2(6,6,4)
      DIMENSION BK(084,084)
      DIMENSION IM(4),JM(4)
      DIMENSION AA(6,6),SKA(6,6)
      COMMON  K1,K2  , K3     , K4     , ID     , NORO   , NN1
      COMMON  UNITS  , ND     , NONO   , N1     , IPO    , PIO
      COMMON  X      , Y      , Z      , E      , F1     , GNU
      COMMON  GNU1   , MEMNO  , MEMTYP , IFGNU  , IFSF   , IFI
      COMMON  IFJ    , IFK    , IFL    , INI    , JNI    , INJ
      COMMON  JNJ    , INK    , JNK    , INL    , JNL    , P1
      COMMON  P2     , P3     , P4     , P5     , P6     , XJ
      COMMON  YK     , XL     , YL     , DC     , SK     , DI
      COMMON  AI     , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK   , SKAL   , A1     , A2     , SKA1   , SKA2
      COMMON  IZ     , NC     , XK     , NOMEM  , ICOUNT , BK
      COMMON  IM     , JM     , NA1    , NA2
      DO 1 I=1,IZ
      DO 1 J=1,NC
      SKA(I,J)=0.0
      DO 1 K=1,IZ
    1 SKA(I,J)=SKA(I,J)+SK(I,K)*AA(K,J)
      JI=NN1*(JN-1)
      IF(IN-INI) 2,2,3
    2 NA1=NA1+1
      IM(NA1)=JI
      CALL READIN(AA,A1,IZ,NC,NA1)
      CALL READIN(SKA,SKA1,IZ,NC,NA1)
      GO TO 4
    3 NA2=NA2+1
      JM(NA2)=JI
      CALL READIN(AA,A2,IZ,NC,NA2)
      CALL READIN(SKA,SKA2,IZ,NC,NA2)
    4 RETURN
      END
```

```
      SUBROUTINE NOD(NONO,NODE,NROW,L)
C...ROUTINE COMPUTES THE LONGITUDINAL NUMBER FOR GIVEN
C...ROW AND NODE
      DIMENSION NONO(25)
      L=NODE-NONO(NROW)
      DO 1 I=1,NROW
    1 L=L+NONO(I)
      RETURN
      END
```

```
      FUNCTION NODET(I,NODE,DEFIN,J,NXC,NONO)
C...ROUTINE COMPUTES NODE NUMBER FOR GIVEN ROW AND COLUMN
      INTEGER DEFIN
      DIMENSION NODE(40,25)
      IF(NONO-NXC)20,10,10
   10 NODET=I
      RETURN
   20 NODET=0
      DO 50 II=1,I
      IF(NODE(II,J)-DEFIN)50,30,50
   30 NODET=NODET+1
   50 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE MINSUB (AA,A,N,DD,JJ)
      DIMENSION AA(JJ,JJ),LL(84),M(84),A(N,N)
      DOUBLE PRECISION A,D
C       THIS LOOP SCALES THE MATRIX TO APPROXIMATELY ONE (1)

      L=0
   10 L=L+1
      AHOW=AA(L,L)
       SCALE=ABS(AHOW)
      IF(SCALE.EQ.0.) GO TO 10
      DO 5 I=1,N
      DO 5 J=1,N
    5   A(I,J)=AA(I,J) /SCALE
      CALL MINV (A,N,D,LL,M)
C       THIS LOOP REMOVES SCALING FACTOR
      DO 6 I=1,N
      DO 6 J=1,N
    6  AA(I,J)= A(I,J) /SCALE
      DD=D
      RETURN
      END
```

```
      SUBROUTINE MINV(A,N,D,L,M)
C
C        ..................................................................
C
C           SUBROUTINE MINV
C
C           PURPOSE
C              INVERT A MATRIX
C
C           USAGE
C              CALL MINV(A,N,D,L,M)
C
C           DESCRIPTION OF PARAMETERS
C              A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
C                  RESULTANT INVERSE.
C              N - ORDER OF MATRIX A
C              D - RESULTANT DETERMINANT
C              L - WORK VECTOR OF LENGTH N
C              M - WORK VECTOR OF LENGTH N
C
C           REMARKS
C              MATRIX A MUST BE A GENERAL MATRIX
C
C           SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C              NONE
C
C           METHOD
C              THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C              IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C              THE MATRIX IS SINGULAR.
C
C        ..................................................................
C
      DIMENSION A(1),L(1),M(1)
C
C           ...............................................................
C
C           IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C           C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C           STATEMENT WHICH FOLLOWS.
C
      DOUBLE PRECISION A,D,BIGA,HOLD
C
C           THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C           APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C           ROUTINE.
C
C           THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C           CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C           10 MUST BE CHANGED TO DABS.
C
C           ...............................................................
C
C           SEARCH FOR LARGEST ELEMENT
C
      D=1.0
```

```
      NK=-N
      DO 80 K=1,N
      NK=NK+N
      L(K)=K
      M(K)=K
      KK=NK+K
      BIGA=A(KK)
      DO 20 J=K,N
      IZ=N*(J-1)
      DO 20 I=K,N
      IJ=IZ+I
   10 IF(DABS(BIGA)-DABS(A(IJ))) 15,20,20
   15 BIGA=A(IJ)
      L(K)=I
      M(K)=J
   20 CONTINUE
C
C        INTERCHANGE ROWS
C
      J=L(K)
      IF(J-K) 35,35,25
   25 KI=K-N
      DO 30 I=1,N
      KI=KI+N
      HOLD=-A(KI)
      JI=KI-K+J
      A(KI)=A(JI)
   30 A(JI) =HOLD
C
C        INTERCHANGE COLUMNS
C
   35 I=M(K)
      IF(I-K) 45,45,38
   38 JP=N*(I-1)
      DO 40 J=1,N
      JK=NK+J
      JI=JP+J
      HOLD=-A(JK)
      A(JK)=A(JI)
   40 A(JI) =HOLD
C
C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
C        CONTAINED IN BIGA)
C
   45 IF(BIGA) 48,46,48
   46 D=0.0
      RETURN
   48 DO 55 I=1,N
      IF(I-K) 50,55,50
   50 IK=NK+I
      A(IK)=A(IK)/(-BIGA)
   55 CONTINUE
C
C        REDUCE MATRIX
C
      DO 65 I=1,N
      IK=NK+I
      HOLD=A(IK)
      IJ=I-N
      DO 65 J=1,N
```

```
      IJ=IJ+N
      IF(I-K) 60,65,60
   60 IF(J-K) 62,65,62
   62 KJ=IJ-I+K
      A(IJ)=HOLD*A(KJ)+A(IJ)
   65 CONTINUE
C
C        DIVIDE ROW BY PIVOT
C
      KJ=K-N
      DO 75 J=1,N
      KJ=KJ+N
      IF(J-K) 70,75,70
   70 A(KJ)=A(KJ)/BIGA
   75 CONTINUE
C
C        PRODUCT OF PIVOTS
C
      D=D*BIGA
C
C        REPLACE PIVOT BY RECIPROCAL
C
      A(KK)=1.0/BIGA
   80 CONTINUE
C
C        FINAL ROW AND COLUMN INTERCHANGE
C
      K=N
  100 K=(K-1)
      IF(K) 150,150,105
  105 I=L(K)
      IF(I-K) 120,120,108
  108 JQ=N*(K-1)
      JR=N*(I-1)
      DO 110 J=1,N
      JK=JQ+J
      HOLD=A(JK)
      JI=JR+J
      A(JK)=-A(JI)
  110 A(JI) =HOLD
  120 J=M(K)
      IF(J-K) 100,100,125
  125 KI=K-N
      DO 130 I=1,N
      KI=KI+N
      HOLD=A(KI)
      JI=KI-K+J
      A(KI)=-A(JI)
  130 A(JI) =HOLD
      GO TO 100
  150 RETURN
      END


      SUBROUTINE NSCRWR(J,N,DX,DY,NOTR)
       DIMENSION DX(NOTR),DY(NOTR)
      WRITE (9,10) J,N
      WRITE(9,11) (DX(I),I=1,NOTR)
```

```
      WRITE(9,11) (DY(I),I=1,NOTR)
      RETURN
 10   FORMAT (2I5)
 11   FORMAT ((15E16.8))
      END


      SUBROUTINE NSCRRE(J,N,DX,DY,NOTR)
       DIMENSION DX(NOTR),DY(NOTR)
       READ (9,10) J,N
       READ(9,11) (DX(I),I=1,NOTR)
       READ(9,11) (DY(I),I=1,NOTR)
      RETURN
 10   FORMAT(2I5)
 11   FORMAT ((15E16.8))
      END


      SUBROUTINE NFILRE(NA,NB,NC,SFX,SFY,PHI,DXL,DYL,NOTR)
      DIMENSION DXL(NOTR),DYL(NOTR)
       READ(7,14)NA,NB,NC,SFX,SFY,PHI
       READ(7,15)(DXL(I),I=1,NOTR)
       READ(7,15)(DYL(I),I=1,NOTR)
      RETURN
 14   FORMAT(3I10,3E16.8)
 15   FORMAT ((15E16.8))
      END


      SUBROUTINE NFILWR(NA,NB,NC,SFX,SFY,PHI,DXL,DYL,NOTR)
      DIMENSION DXL(NOTR),DYL(NOTR)
      WRITE(7,14)NA,NB,NC,SFX,SFY,PHI
      WRITE(7,15)(DXL(I),I=1,NOTR)
      WRITE(7,15)(DYL(I),I=1,NOTR)
      RETURN
 14   FORMAT(3I10,3E16.8)
 15   FORMAT ((15E16.8))
      END


      SUBROUTINE OCCM(T,PROP,IDF,X,INFLU,SFS,SFL,SFA,E,G)
C--OCCURRENCE MATRIX DEVELOPMENT
      DIMENSION T(5,5),PROP(5),TR(5,5),TC(5,5)
      CALL IDENT(T,5)
      EI=E*PROP(1)*SFS
      GA=G*PROP(2)*SFA
      S=0.
      IF(GA)120,120,110
```

```
      SUBROUTINE SETOC(NSEC,ZI,EYE,NQ,ZQ,Q,NP,ZP,P,ZLEN,NOCC,ODATA)
C--ROUTINE TO SET UP OCCURRENCE DATA VECTORS
      DIMENSION ZI(20),EYE(20),ZQ(20),Q(20),ZP(50),P(50),ODATA(8,50),
     +ZOCC(70)
C
C--DETERMINE LOCATIONS FOR ALL OCCURRENCE CHANGES
      ZOCC(1)=0.
      ZOCC(2)=ZLEN
      NOCC=2
C--FIRST ARRANGE X-SECTION CHANGES IN ASCENDING ORDER LEFT TO RIGHT
      ZI(NSEC+1)=ZLEN
   10 DO 20 N=1,NSEC
      IF(ZI(N)-ZI(N+1))20,20,30
   20 CONTINUE
      GO TO 40
   30 SAVE=ZI(N)
      ZI(N)=ZI(N+1)
      ZI(N+1)=SAVE
      SAVE=EYE(N)
      EYE(N)=EYE(N+1)
      EYE(N+1)=SAVE
      GO TO 10
   40 DO 50 N=1,NSEC
      NOCC=NOCC+1
   50 ZOCC(NOCC)=ZI(N)
C
      IF(NQ)110,110,60
   60 ZQ(NQ+1)=ZLEN
C--ARRANGE UNIFORM LOADS IN ASCENDING ORDER LEFT TO RIGHT
   70 DO 80 N=1,NQ
      IF(ZQ(N)-ZQ(N+1))80,80,90
   80 CONTINUE
      GO TO 95
   90 SAVE=ZQ(N)
      ZQ(N)=ZQ(N+1)
      ZQ(N+1)=SAVE
      SAVE=Q(N)
      Q(N)=Q(N+1)
      Q(N+1)=SAVE
      GO TO 70
   95 DO 100 N=1,NQ
      NOCC=NOCC+1
  100 ZOCC(NOCC)=ZQ(N)
  110 IF(NP)200,200,120
  120 ZP(NP+1)=ZLEN
C--ARRANGE CONCENTRATED LOADS IN ASCENDING ORDER LEFT TO RIGHT
  130 DO 140  N=1,NP
      IF(ZP(N)-ZP(N+1))140,140,150
  140 CONTINUE
      GO TO 160
  150 SAVE=ZP(N)
      ZP(N)=ZP(N+1)
      ZP(N+1)=SAVE
      SAVE=P(N)
      P(N)=P(N+1)
      P(N+1)=SAVE
      GO TO 130
```

```
 160   DO 170 N=1,NP
       NOCC=NOCC+1
 170   ZOCC(NOCC)=ZP(N)
C--ARRANGE OCCURRENCE LOCATIONS IN ASCENDING ORDER LEFT TO RIGHT
 200   CALL SORT(ZOCC,NOCC)
C
C--INSERT OCCURRENCE DATA
       NOCC=NOCC-1
       DO 500 J=1,NOCC
       ODATA(6,J)=ZOCC(J)
       ODATA(7,J)=ZOCC(J+1)
       ODATA(8,J)=1.
C-- ODATA(8,J) INDICATES IF A CONCENTRATED CONDITION OCCURRS AT THE LEFT AND
C  AN OCCURRENCE FIELD.  IF SET TO 1, NONE EXISTS.  IF SET TO 2, ONE DOES.
       DO 210 I=1,5
 210   ODATA(I,J)=0.
       DO 230 N=1,NSEC
       IF(ZI(N)-ZOCC(J))220,240,250
 220   IF(ZI(N+1)-ZOCC(J))230,230,240
 230   CONTINUE
 240   ODATA(1,J)=EYE(N)
 250   IF(NO)300,300,260
 260   DO 280 N=1,NO
       IF(ZO(N)-ZOCC(J))270,290,300
 270   IF(ZO(N+1)-ZOCC(J))280,280,290
 280   CONTINUE
 290   ODATA(3,J)=O(N)
 300   IF(NP)400,400,310
 310   DO 320 N=1,NP
       IF(ZP(N)-ZOCC(J))320,315,400
 315   ODATA(5,J)=P(N)
       ODATA(8,J)=2.
 320   CONTINUE
 400   CONTINUE
 500   CONTINUE
       RETURN
       END



       SUBROUTINE SHIP1(NCARD,MACO)
C...ROUTINE DEVELOPS FINITE ELEMENT STIFFNESS MATRICES
C...FOR THE TRANSVERSE MEMBER
C      FORMATION OF STIFFNESS MATRICES
       DIMENSION BKK(84,84),BKH(84,84),BK(84,84),NOB(25)
       DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
       DIMENSION X(25,40),Y(25,40),GNU1(4),DC(2,2),
      1SK(6,6), DI(6,6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),SKAI(6,6),
      2SKAJ(6,6),SKAK(6,6),SKAL(6,6),A1(6,6,4),A2(6,6,4),SKA1(6,6,4),
      3 SKA2(6,6,4)
       DIMENSION IM(4),JM(4),ZAI(6),ZAJ(6),ZAK(6),ZAL(6) ,XI(6)
        DIMENSION AOX(84,84)
       COMMON K1,MEMTO,NOMAT,MOB,ID,NORO,NN1
       COMMON  UNITS  , ND      , NONO    , N1      , IPO     , PIO
       COMMON X,Y,Z,E,NOBB,MCOM,NOBO,MCON,GNU
       COMMON GNU1,MEMNO,MEMTYP,FA,IFSF,IFI
       COMMON  IFJ     , IFK     , IFL     , INI     , JNI     , INJ
       COMMON  JNJ     , INK     , JNK     , INL     , JNL     , P1
       COMMON  P2      , P3      , P4      , P5      , P6      , XJ
```

```
      COMMON  YK     , XL     , YL     , DC     , SK     , DI
      COMMON  AI     , AJ     , AK     , AL     , SKAI   , SKAJ
      COMMON  SKAK   , SKAL   , A1     , A2     , SKA1   , SKA2
      COMMON  IZ     , NC     , XK     , NOMEM  , ICOUNT , BK
      COMMON  IM     , JM     , NA1    , NA2    , ZAI    , ZAJ
      COMMON  ZAK    , ZAL    , ITEMP  , ALPHA  , XI
      COMMON/K23RW/KH(3,49),KHB(4,49),KJ,KJB,IFD,LJ(4)
C      IPO(1) = TEMPERATURE FLAG, ITEMP
C      PIO(1) IS USED TO PASS THE TIME OF DAY
      CALL INPUT
         IF (NORO.GT.25)  GO TO 331
          IF (NOMAT.GT.4) GO TO 331
       K1=1
      IJK=0
      DO 99 I=1,NOBO
   99 NOB(I)=N1(I)
         IF (NOBO.GT.NORO) GO TO 331
          IF (NOBB.GT.1)  GO TO 331
          IF (MCON.GT.1) GO TO 331
C       NOBO=NUMBER OF ROWS WITH BOUNDARY CONDITIONS
C     NOB(I) IMPLIES BOUNDARY CONDITION AT THE END OF NOB(I)TH ROW
C       NOBB=1,FIXED IN X-DIRECTION,NOBB=0,FIXED IN Y DIRECTION
C      MCON=NUMBER OF BOUNDARY CONDITION AT 1ST ROW
C      MOB(I) IMPLIES BOUNDARY CONDITION AT MOB(I) TH NODE
C      MCOM SAME AS NOBB BUT FOR BOUNDARY CONDITIONS OF 1ST ROW
      KJ =0
       CALL REWIND(3)
      CALL REWIND (4)
      IPO(1)=ITEMP
        ND(1)=3
       ND(2)=5
       ND(5)=1
        GO TO 333
  331  WRITE (6,332)
          STOP
  333 MACO=MAXCOL(NN1,NONO,NORO)
      NOMEM=0
      ICOUNT=1
      N1(1)=0
      ICOL1=NN1*NONO(1)
      ICOL2=NN1*NONO(2)
      DO 9  I=1,ICOL1
      DO 9  J=1,ICOL1
    9 BK(I,J)=0.0
   11 IJK=IJK+1
      CALL INFO(IJK)
      IF (INI-ICOUNT) 12,12,13
   12 IF(MEMTYP) 13,13,26
   26 IZ=ND(MEMTYP)
      IF(NOMEM.GT.0) GO TO 14
      DO 24 I=1,ICOL1
      DO 24 J=1,ICOL2
   24 BKK(I,J)=0.
      DO 25  I=1,ICOL2
      DO 25  J=1,ICOL2
   25 BKH(I,J)=0.
   14  CONTINUE
      DO 20 K=1,4
      IM(K)=0
      JM(K)=0
```

```
      DO 20 I=1,IZ
      DO 20 J=1,NN1
      A1(I,J,K)=0.0
      A2(I,J,K)=0.0
      SKA1(I,J,K)=0.0
   20 SKA2(I,J,K)=0.0
      NA1=0
      NA2=0
      GO TO (1,2,5,5,5),MEMTYP
    1 CALL MEM1
      GO TO 10
    2 CALL MEM2
      GO TO 10
    5 CALL MEM5
C       IFSF = 1        FORCE ONLY
C       IFSF= 2         FORCE AND STRESS
C       IFSF = 3        STRESS ONLY
C       FOR THERMAL STRESSE PROBLEM IFSF MUST BE GREATER
C       THAN ZERO FOR ALL MEMBERS
   10 IF (IFSF) 22,22,23
   23 WRITE(4,3)MEMNO,MEMTYP,INI,JNI,INJ,JNJ,INK,JNK,INL,JNL,IFSF,IFI,IF
     1J,IFK,IFL,NC,AI,AJ,AK,AL,SKAI,SKAJ,SKAK,SKAL,DI
      N1(INI)=N1(INI)+1
   22 DO 30 I=1,NA1
       IMM=IM(I)
       DO 30 J=1,NA1
       JMM=IM(J)
 30   CALL TRAMPS(A1,I,IZ,NC,SKA1,J,IMM,JMM,BK)
      IF(NA2)16,16,31
   31 DO 35 I=1,NA1
      IMM=IM(I)
      DO 35 J=1,NA2
      JMM=JM(J)
 35   CALL TRAMPS(A1,I,IZ,NC,SKA2,J,IMM,JMM,BKK)
      DO 36 I=1,NA2
      IMM=JM(I)
      DO 36 J=1,NA2
      JMM=JM(J)
 36   CALL TRAMPS(A2,I,IZ,NC,SKA2,J,IMM,JMM,BKH)
   16 CONTINUE
      NOMEM=NOMEM+1
      GO TO 11
   13 IRO=INI-1
       DO 49 IP=1,NOBO
         IF (IRO.NE.NOB(IP)) GO TO 49
        IK=NONO(IRO)*2-NOBB
         BK(IK,IK)=BK(IK,IK)*10000000.*FA
   49   CONTINUE
         IF (IRO.NE.1) GO TO 233
      JK=MOB*2-MCOM
        BK(JK,JK)=BK(JK,JK)*10000000.*FA
 233  CALL K3WR(IRO,ICOL1,ICOL2,BK,0 ,AOX,MACO)
      IF(INI-NORO.GT.0) GO TO 999
      CALL K3WR(IRO,ICOL1,ICOL2,BKK,-1,AOX,MACO)
      DO 15 I=1,ICOL2
      DO 15 J=1,ICOL2
 15   BK(I,J)=BKH(I,J)
      ICOL1=NN1*NONO(INI)
        IF (INI.EQ.NORO) GO TO 70
      ICOL2=NN1*NONO(INI+1)
```

```
           GO TO 71
   70    ICOL2=ICOL1
   71   ICOUNT=INI
      N1(INI)=0
      NOMEM=0
      IF(JNI)11,11,12
  999 KJ =0
      CALL REWIND (3)
      CALL REWIND (4)
      WRITE (6,102)ID
      WRITE(6,100)ID,NORO,NN1,NOMAT,ITEMP,(UNITS(I),I=1,4)
      WRITE(6,110)NOBO,MCON,NOBB,MCOM
      WRITE(6,110)(NOB(I),I=1,NOBO)
      WRITE(6,110) MOB,IFSF
      WRITE(6,103)FA
      WRITE(6,103) E,GNU,ALPHA
      WRITE(6,110) (NONO(L),L=1,NORO)
      RETURN
    3 FORMAT(16I5,/,(15E16.8))
    4 FORMAT(12I5,/,(15E16.8))
  100 FORMAT(5I5,4A6)
  102 FORMAT (24H1DATA FOR PROBLEM NUMBER,I6)
  103 FORMAT(E10.2,F7.2,E10.2)
  110  FORMAT(20I3)
  332  FORMAT (//25H INPUT ERRORS IN SHIP1      //)
  400  FORMAT (4E15.4)
      END


      SUBROUTINE SHP5(SPRING,MOR,MACH)
C     MATRIX  TRIANGULARIZATION
C     INPUT OF FORCE DATA AND BACK SUBSTITUTION FOR FINAL
C     SOLUTION OF EQUATIONS
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10),
     1 INDEX(084,3)
       DIMENSION BK(84,84),BK2(84,84),BTEMP(84)
       DIMENSION AOX(84,84),DUMB(755)
      DIMENSION R(2100),VTEMP(84)
      COMMON/K23RW/KH(3,49),KHB(4,49),KJ,KJB,IFD,LJ(4)
      COMMON  K1,K2   , K3      , K4      , ID      , NORO    , NN1
      COMMON  UNITS   , ND      , NONO    , N1      , IPO     , PIO
      COMMON NUMFO,R,BK2,BTEMP,VTEMP,DUMB
        COMMON/INFLU/AF(50,50),FIG(50),DY(50),MFILE,MF,LMOD(100)
     1 ,LROW(100),DX(50),MF
      COMMON/SAFE/ BK,DMY(444)
      COMMON/SHIP/NOLO,LNO(100),SFX(100),SFY(100),PHI(100),DUMY(100)
      EXTERNAL GETFD
      INTEGER*4 ADROF
      DATA FN/'-A '/
      CALL RCALL(GETFD,2,0,ADROF(FN),1,IFD)
      KKK=0
      KJ =0
      KJB=0
      CALL REWIND (3)
      CALL K3RE(IRO,ICOL1,ICOL2,BK,0 ,AOX,MACH)
      DO 35 II=1,NORO
        IF (MOR.EQ.0)  GO TO 79
        DO 74 IK=1,NOLO
```

```
          AN=PHI(IK)
         IF (II.NE.LROW(IK))   GO TO 74
          J1=LNOD(IK)*2-1
          J2=J1+1
         A=COS(AN)*SFX(IK)-SIN(AN)*SFY(IK)
          B=SIN(AN)*SFX(IK)+COS(AN)*SFY(IK)
         BK(J1,J1)=BK(J1,J1)+A*SPRING
         BK(J2,J2)=BK(J2,J2)+B*SPRING
   74    CONTINUE
   79   CONTINUE
        CALL MATINS(BK,84,ICOL1,BK2,84,00,DD,M,INDEX)
        GO TO (36,38),M
   38   WRITE (6,111)IRO
        WRITE (6,121) II,DD
        STOP
   36   CALL K2WR(IRO,ICOL1,ICOL2,BK,0 ,AOX,MACO)
        IF(II-NORO) 40,39,39
   40   CALL K3RE (IRO,ICOL1,ICOL2,BK2,-1,AOX,MACO)
        CALL K2WR (IRO,ICOL1,ICOL2,BK2,-1,AOX,MACO)
        DO 44 J=1,ICOL1
        DO 43 K=1,ICOL2
        BTEMP(K)=0.0
        DO 43 I=1,ICOL1
   43   BTEMP(K)=BTEMP(K)+BK(I,J)*BK2(I,K)
        DO 44 I=1,ICOL2
   44   BK(I,J)=BTEMP(I)
        DO 42 I=1,ICOL2
        IIK=KKK+ICOL1+I
        DO 42 J=1,ICOL1
        IKK=KKK+J
   42   R(IIK)=R(IIK)-BK(I,J)*R(IKK)
        KKK=KKK+ICOL1
        DO 50 K=1,ICOL2
        DO 51 I=1,ICOL2
        BTEMP(I)=0.0
        DO 51 J=1,ICOL1
   51   BTEMP(I)=BTEMP(I)+BK(I,J)*BK2(J,K)
        DO 50 I=1,ICOL2
   50   BK2(I,K)=BTEMP(I)
        CALL K3RE(IRO,ICOL1,ICOL2,BK,0,AOX,MACO)
        DO 35 I=1,ICOL1
        DO 35 J=1,ICOL1
   35   BK(I,J)=BK(I,J)-BK2(I,J)
   39   LJ(1)=KHB(4,KJP)
        CALL POINT(IFO,LJ,1)
        KJ =0
        CALL REWIND (3)
         WRITE (6,110) K1,K2,K3,K4
        DO 47 II=1,NORO
        CALL K2RE(IRO,ICOL1,ICOL2,BK,0 ,AOX,MACO)
        DO 45 I=1,ICOL1
        VTEMP(I)=0.
        DO 45 J=1,ICOL1
        IKK=KKK+J
   45   VTEMP(I)=VTEMP(I)+BK(I,J)*R(IKK)
        DO 46 I=1,ICOL1
        IKK=KKK+I
   46   R(IKK)=VTEMP(I)
        IF (II-NORO)    49,54,54
   49   CALL K2RE(IRO,ICOL1,ICOL2,BK,-1,AOX,MACO)
```

```
      KKK=KKK-ICOL1
      DO 47 I=1,ICOL1
      IKK=KKK+I
      DO 47 J=1,ICOL2
   47 R(IKK)=R(IKK)-BK(I,J)*VTEMP(J)
 54   KJB=0
        WRITE (6,400) (R(I),I=1,IKK)
      RETURN
    3 FORMAT (I5)
  110  FORMAT(10I10)
  111 FORMAT(1HO,17HSINGULAR IN ROW   ,I3,14H     TOUGH LUCK)
  121 FORMAT (9H ROW NO =, I10,10H  DETERM =, F20.5)
  400  FORMAT (4E15.4)
      END



      SUBROUTINE SHIP4
      DIMENSION  JACK(25,40),FORCE(25,40,2)
      DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
      DIMENSION V(2100),UU(6),OO(6),AI(6,6),AJ(6,6),AK(6,6),AL(6,6),
     1SKAI(6,6),SKAJ(6,6),SKAK(6,6),SKAL(6,6),DI(6,6)
      DIMENSION VTEE(6,3)
      COMMON  K1,K2  , K3      , K4      , ID      , NORO    , NN1
      COMMON  UNITS  , ND      , NONO    , N1      , IPO     , PIO
      COMMON  NUMEO  , V       , KK      , KKK     , III     , IZ
      COMMON  UU     , OO      , MEMNO   , MEMTYP  , INI     , JNI
      COMMON  INJ    , JNJ     , INK     , JNK     , INL     , JNL
      COMMON  IFSF   , IFI     , IFJ     , IFK     , IFL     , AI
      COMMON  AJ     , AK      , AL      , SKAI    , SKAJ    , SKAK
      COMMON  SKAL   , DI      , VTEE           ,DUMB(7606)
      COMMON/SAFE/JACK,FORCE,DMY(4500)
      WRITE (6,110)ID
      III=1
      WRITE (6,100)UNITS(1),UNITS(2),UNITS(3),UNITS(4)
      CALL REWIND (4)
      KKK=NN1*NONO(1)
      KK=0
      DO 2 I=1,NORO
      JOE=NONO(I)
      DO 2 J=1,JOE
      JACK(I,J)=0
      DO 2 K=1,NN1
    2 FORCE(I,J,K)=0.0
      DO 59 II=1,NORO
      JOE=N1(II)
      IF (JOE) 23,23,22
 22   CONTINUE
      DO 58 JJ=1,JOE
       READ(4,6)MEMNO,MEMTYP,INI,JNI,INJ,JNJ,INK,JNK,INL,JNL,IFSF,IFI,IF
     1J,IFK,IFL,NC,AI,AJ,AK,AL,SKAI,SKAJ,SKAK,SKAL,DI
      IZ=ND(MEMTYP)
      GO TO (4,5,8,8,8),MEMTYP
    4 CALL MEMB1
      GO TO 20
    5 CALL MEMB2
      GO TO 20
    8 CALL MEMB5
   20 IF (IFSF-2) 10,10,58
```

```
 10 IF (IFI) 12,12,11
 11 CALL SR15(AI,OO,INI,JNI,IZ,NN1)
 12 IF(IFJ) 14,14,13
 13 CALL SR15(AJ,OO,INJ,JNJ,IZ,NN1)
 14 IF(IFK) 16,16,15
 15 CALL SR15(AK,OO,INK,JNK,IZ,NN1)
 16 IF(IFL) 58,58,17
 17 CALL SR15(AL,OO,INL,JNL,IZ,NN1)
 58 CONTINUE
 23 CONTINUE
    KK=KKK
 59 KKK=KKK+NN1*NONO(II+1)
    WRITE(6,101) UNITS(1),UNITS(2)
    DO 24 I=1,NORO
    JOE=NONO(I)
    DO 24 J=1,JOE
    IF(JACK(I,J)) 24,24,25
 25 WRITE (6,102)III,I,J,(FORCE(I,J,K),K=1,NN1)
 24 CONTINUE
 26 IK=1
    IKK=NN1
    WRITE (6,103)UNITS(3),UNITS(4)
    DO 27 I=1,NORO
    IN=NONO(I)
    DO 27 II=1,IN
    WRITE (6,104)III,I,II,(V(IV),IV=IK,IKK)
    IK=IK+NN1
 27 IKK=IKK+NN1
    RETURN
104 FORMAT(1H ,I8,2I9,3X,6E14.6)
103 FORMAT(23H1NODE DISPLACEMENTS IN ,2A6,//,
   127H LOAD SYSTEM    ROW      NODE,10X,7HX-DISP ,7X,7HY-DISP  )
102 FORMAT(1H ,3I9,3X,6E14.6)
101 FORMAT(21H1CUT NODE FORCES  IN  ,2A6,//,
   128H LOAD SYSTEM    ROW      NODE,9X,7HX-FORCE,7X,7HY-FORCE)
 6  FORMAT(16I5,/,(15E16.8))
100 FORMAT(20H1MEMBER STRESSES IN ,2A6,14H  PER SQUARE  ,2A6//
   1132H LOAD SYSTEM    ROW   MEMBER TYPE AND NUMBER    X-STRESS
   2Y-STRESS   SHEAR STRESS  1ST PRINC STR 2ND PRINC STR  ANGLE  1ST P
   3RINC/44X,14H(TRIANG PLATE),58X,16HSTRESS TO X-AXIS//48X,
   48HX-STRESS,6X,7HX-GRAD ,7X,8HY-STRESS,6X,7HY-GRAD ,5X,
   512HSHEAR STRESS/46X,12H(QUAD PLATE)//48X,8HX-STRESS/50X,5H(BAR)//)
110 FORMAT(27H1RESULTS FOR PROBLEM NUMBER,I8)
    END
```

```
      SUBROUTINE SORT(X,N)
C...ROUTINE ARRANGES X-ARRAY IN ASCENDING ORDER AND THROWS
C...OUT DUPLICATE VALUES
      DIMENSION X(N)
   5  NM1=N-1
      IF(NM1)99,99,10
 10   DO 15 I=1,NM1
      IF(X(I)-X(I+1))15,20,30
  15  CONTINUE
  99  RETURN
  20  X(I)=X(N)
```

```
      N=N-1
      GO TO 5
   30 SAVE=X(I)
      X(I)=X(I+1)
      X(I+1)=SAVE
      GO TO 10
      END
```

```
      SUBROUTINE SR14(KK,N,JN,AIJK,OU,III,KKK,IIN,IJK)
C     SR14  BRANCH DISPLACEMENTS OR FORCES SHIP 4
      DIMENSION V(2100),AIJK(6,6),OU(6),W(6)
       DIMENSION UNITS(4),ND(6),NONO(25),N1(25),IPO(10),PIO(10)
       COMMON K1,K2,K3,K4,IO,NORO,NN1,UNITS,ND,NONO,N1,IPO,PIO
       COMMON NUMFO,V
      IF (IJK-IIN) 3,3,4
    3 IO=KK+NN1*(JN-1)
      GO TO 5
    4 IO=KKK+NN1*(JN-1)
    5 DO 1 I=1,NN1
      II=IO+I
    1 W(I)=V(II)
      DO 2 I=1,N
      DO 2 J=1,NN1
    2 OU(I)=OU(I)+AIJK(I,J)*W(J)
      RETURN
      END
```

```
      SUBROUTINE SR15(AIJK,OO,II,JI,N,NN1)
C     SR15 NODE FORCES SHIP 4
      DIMENSION AIJK(6,6),OO(6),FORCE(25,40,2),JACK(25,40)
      COMMON/SAFE/JACK,FORCE,DMY(4500)
      DO 1 J=1,NN1
      DO 1 I=1,N
    1 FORCE(II,JI,J)=FORCE(II,JI,J)-AIJK(I,J)*OO(I)
      JACK(II,JI)=1
      RETURN
      END
```

```
      SUBROUTINE SR4A(A,M,N,B,C)
C     SR4A MATRIX MULTIPLICATION SHIP 4
      DIMENSION A(6,6),B(6),C(6)
      DO 1 I=1,M
      C(I)=0.0
      DO 1 J=1,N
    1 C(I)=C(I)+A(I,J)*B(J)
      RETURN
      END
```

```
      SUBROUTINE SWTCH(I1,I2)
      IF(I1-I2)99,99,10
   10 I=I1
      I1=I2
      I2=I
   99 RETURN
      END


       SUBROUTINE TEMPCO(       NC,IZ,SKA,XI,ZA)
      DIMENSION SKA(6,6),XI(6),ZA(6)
      DO 2 I=1,NC
      ZA(I)=0.0
      DO 2 J=1,IZ
    2 ZA(I)=ZA(I)+SKA(J,I)*XI(J)
      RETURN
      END


      SUBROUTINE TMATT(XB,NB.X,TM,NOCC,NCON,ODATA.SES,SEL,SEA,E,G)
C...ROUTINE TO COMPUTE OCCURRENCE MATRIX FROM LOCATION
C...ZER TO X
      DIMENSION TM(5,5),TO(5,5),TR(5,5),PROP(5),ODATA(8,50)
      INFLU=0
      IF(NCON)110,110,100
  100 IF(X-XB)110,120,120
  110 CALL IDENT(TM,5)
      NB=1
      XB=0.
      NCON=1
  120 NBB=NB
      DO 300 N=NBB,NOCC
      X1=ODATA(6,N)
      X2=ODATA(7,N)
      IDF=ODATA(8,N)
      DO 130 I=1,5
  130 PROP(I)=ODATA(I,N)
C--CHECK IF POINT MATRIX HAS BEEN USED IN 1ST OCCURENCE
      IF(N-NBB)140,140,160
  140 IF(XB-X1)160,160,150
  150 IDF=1
  160 NB=N
      IF(X-X2)180,170,190
  170 NB=N+1
  180 NSTOP=1
      Y=X-XB
      XB=X
      GO TO 295
  190 NSTOP=0
      Y=X2-XB
      XB=X2
      GO TO 295
  295 CALL OCCM(TO,PROP,IDF,Y,INFLU,SES,SEL,SEA,E,G)
      CALL MMULT(TO,TM,TR,5,5,5)
      CALL EQUAL(TM,TR,5,5)
      IF(NSTOP)300,300,400
```

```
 300  CONTINUE
 400  RETURN
      END


      SUBROUTINE TRAMPS(A,KA,IZ,NC,SKA,KB,IMM,JMM,BK)
        DIMENSION A(6,6,4),SKA(6,6,4),BK(84,84)
      DO 1 I=1,NC
      DO 1 J=1,NC
      DO 1 K=1,IZ
      IBK=IMM+I
      JBK=JMM+J
    1 BK(IBK,JBK)=BK(IBK,JBK)+A(K,I,KA)*SKA(K,J,KB)
      RETURN
      END


      SUBROUTINE TRV(NCARD)
C...ROUTINE INPUTS DEFINITION OF TRANSVERSE AND GENERATES
C...ALL FINITE ELEMENT DATA
      INTEGER DEFIN,BLANK
      COMMON /WORK/XC(42),YC(28),NONO(25),NXC,NYC,
     +DEFIN,NODE(40,25),LROW(100),LNOD(100)
      COMMON /SAFE/UNITS(4),IT1(50),IT2(50),
     +JT1(50),JT2(50),THK(50),AX(100),IB1(100),IB2(100),
     +JB1(100),JB2(100),IO(100),JO(100),NCP(51),NOB(25)
      COMMON /SAFE/IERR,XLBHD,DECL,DESH,
     +I,J,JROW1,JROW2,ICOL1,ICOL2,BLANK,ID,NORO,
     +NOMAT,NN1,ITEMP,NORB,MCOM,NORO,MCON,NM(25),
     +X,Y,Z,SLOPE,NTA,K,NPAR,IFSF,NOUT,NOMAX,JRO1,JRO2,
     +ICOL,JROW,NXCM1,IEGNU,MEMNO,JP1,IP1,NI1,NI2,NI3,NI4,
     +T,AX1,AX2,AX3,AX4,AX5,J1,J2,JJ,II,IFI,IFJ,IFK,IFL,
     +NUL,P,MEMTYP,MEMTO,N,DMMY(6383)
      COMMON /MATRL/E,G,GNU,ALPHA,CONVF
      KODE=0
      BLANK=0
      DEFIN=1
      IERR=0
      WRITE(6,200)
      WRITE(6,201)
      READ(5,300)XLBHD,DECL,DESH
      WRITE(6,300)XLBHD,DECL,DESH
C**
C**SET UP BASIC GRID COORDINATES
      WRITE(6,202)
      READ(5,1) NXC,NYC
      WRITE(6,1) NXC,NYC
      WRITE(6,203)
      DO 50 I=1,NXC
   50 READ(5,2) XC(I)
      NXC=NXC+1
      XC(NXC)=XLBHD
      CALL SORT(XC,NXC)
      WRITE(6,208)(I,XC(I),I=1,NXC)
      WRITE(6,204)
      DO 51 J=1,NYC
```

```
  51  READ(5,2) YC(J)
      YC(NYC+1)=DECL
      YC(NYC+2)=DESH
      NYC=NYC+2
      CALL SORT(YC,NYC)
      WRITE(6,208)(J,YC(J),J=1,NYC)
      DO 100 I=1,NXC
      DO 100 J=1,NYC
 100  NODE(I,J)=DEFIN
C**
C**DEFINE VOID AREAS WITHIN TRANSVERSE
      WRITE(6,205)
 110  READ(5,3)  JROW1,JROW2,ICOL1,ICOL2
      CALL SWTCH(JROW1,JROW2)
      CALL SWTCH(ICOL1,ICOL2)
      WRITE(6,3)  JROW1,JROW2,ICOL1,ICOL2
C--ZERO JROW1 WILL STOP INPUT OF VOID DEFINITION DATA
      IF(JROW1)150,150,130
 130  DO 135  J=JROW1,JROW2
      DO 135  I=ICOL1,ICOL2
 135  NODE(I,J)=BLANK
      GO TO 110
C--PLOT TRANSVERSE PROFILE
 150  DO 155 J=1,NYC
 155  NN(J)=J
      WRITE(6,207)(NN(J),J=1,NYC)
      DO 140 I=1,NXC
      WRITE(6,206)I,(NODE(I,J),J=1,NYC)
 140  CONTINUE
C**DEFINE NUMBER OF NODES PER ROW
      DO 410  J=1,NYC
      NONO(J)=0
      DO 405  I=1,NXC
      IF(NODE(I,J)-DEFIN)405,404,405
 404  NONO(J)=NONO(J)+1
 405  CONTINUE
 410  CONTINUE
C**
C**BEGIN WRITING ON NCARD FILE
      ID=7777
      NORO=NYC
      NOMAT=1
      NN1=2
      ITEMP=0
      WRITE(6,210)
      READ(5,301)UNITS(1),UNITS(2)
      WRITE(6,301)UNITS(1),UNITS(2)
      WRITE(6,211)
      READ(5,301)UNITS(3),UNITS(4)
      WRITE(6,301)UNITS(3),UNITS(4)
      CALL REWIND (8)
C--NCARD     8
       WRITE (8,212)  ID,NORO,NN1,NOMAT,ITEMP,(UNITS(I),I=1,4)
C--DEFINITION OF BOUNDARY CONDITIONS
      NOBO=0
      DO 166  J=1,NYC
      IF(NODE(NXC,J)-DEFIN)166,167,166
 167  NOBO=NOBO+1
      NOB(NOBO)=J
 166  CONTINUE
```

```
      MCON=1
      DO 168   I=1,NXC
      IF(XC(I)-XLBHD)168,169,169
 168  CONTINUE
 169  MOB=NODET(I,NODE,DEFIN,1,NXC,NONO(1))
      MOBC=I
      WRITE(6,272)
      WRITE(6,273)
      READ(5,1) NOBB,MCOM
      WRITE(6,1) NOBB,MCOM
C--NCARD
      WRITE(8,274)     NOBO,MCON,NOBB,MCOM
      WRITE (8,274)    (NOB(I),I=1,NOBO)
      WRITE(8,274) MOB
      WRITE(6,289)
      READ(5,2) FA
      WRITE(6,2) FA
      WRITE(8,214) FA
      WRITE(6,311)NOBO
      WRITE(6,312)(NOB(I),I=1,NOBO)
      WRITE(6,314)MOBC,MOB
      WRITE(6,313)E,GNU,ALPHA
      WRITE(8,214)     E,GNU,ALPHA
C--NCARD
      WRITE(8,215)     (NONO(J),J=1,NORO)
      WRITE(6,282)
      WRITE(6,207)(NN(J),J=1,NYC)
      DO 406   I=1,NXC
      DO 407   J=1,NYC
      NCR(J)=9999
      IF(NODE(I,J)-DEFIN)407,408,407
 408  NCR(J)=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
 407  CONTINUE
      WRITE(6,251)I,(NCR(J),J=1,NYC)
 406  CONTINUE
C**
C**DEFINE COORDINATES FOR NODES
      Z=0.
      SLOPE=(DECL-DESH)/XLBHD
      DO 430   J=1,NORO
      Y=YC(J)*CONVF
      DO 420   I=1,NXC
      IF(NODE(I,J)-DEFIN)420,411,420
 411  X=XC(I)
      IF(J-NORO)415,412,412
 412  IF(X-XLBHD)413,415,415
 413  Y=DESH+SLOPE*X
      IF(Y-YC(J-1))414,414,416
 414  Y=Y+0.01*(YC(J)-YC(J-1))
 416  Y=Y*CONVF
 415  X=X*CONVF
C--NCARD
      WRITE(8,216)     X,Y,Z
 420  CONTINUE
 430  CONTINUE
C**
C**DEFINE AREAS OF THE TRANSVERSE FOR DIFFERENT PLATE THICKNESSES
      WRITE(6,220)
      WRITE(6,221)
      READ(5,4) NTA
```

```
      WRITE(6,4) NTA
      WRITE(6,260)
      WRITE(6,222)
      DO 450 K=1,NTA
      READ(5,5)  THK(K),JT1(K),JT2(K),IT1(K),IT2(K)
      CALL SWTCH(JT1(K),JT2(K))
      CALL SWTCH(IT1(K),IT2(K))
      WRITE(6,261)THK(K),JT1(K),JT2(K),IT1(K),IT2(K),K
  450 CONTINUE
      WRITE(6,220)
      WRITE(6,207)(NN(J),J=1,NYC)
      DO 456  I=1,NXC
      DO 458  J=1,NYC
      NCR(J)=9999
      IF(NODE(I,J)-DEFIN)458,457,458
  457 DO 455  K=1,NTA
      IF(JT1(K)-J)451,451,455
  451 IF(JT2(K )-J)455,452,452
  452 IF(IT1(K)-I)453,453,455
  453 IF(IT2(K)-I)455,454,454
  454 NCR(J)=K
  455 CONTINUE
  458 CONTINUE
  456 WRITE(6,251)I,(NCR(J),J=1,NYC)
C**
C**DEFINE BAR ELEMENTS
      WRITE(6,223)
      WRITE(6,224)
      READ(5,4) NBAR
      WRITE(6,4) NBAR
      IF(NBAR)484,484,459
  459 WRITE(6,225)
      DO 470 K=1,NBAR
      READ(5,5)  AX(K),JB1(K),JB2(K),IB1(K),IB2(K)
      IF(IB1(K)-IB2(K))461,460,461
  460 CALL SWTCH(JB1(K),JB2(K))
      GO TO 469
  461 IF(JB1(K)-JB2(K))463,462,463
  462 CALL SWTCH(IB1(K),IB2(K))
      GO TO 469
  463 IF(IABS(JB2(K)-JB1(K))-IABS(IB2(K)-IB1(K)))464,465,464
  464 IERR=1
      WRITE(6,262)
      GO TO 469
  465 IF(JB1(K)-JB2(K))469,469,466
  466 CALL SWTCH(JB1(K),JB2(K))
      ITEMP=IB1(K)
      IB1(K)=IB2(K)
      IB2(K)=ITEMP
  469 CONTINUE
      WRITE(6,261)AX(K),JB1(K),JB2(K),IB1(K),IB2(K),K
  470 CONTINUE
C**
C**DEFINE OUTPUT REQUIREMENTS
  484 WRITE(6,226)
      WRITE(6,228)
      READ(5,4) IFSF
      WRITE(6,4) IFSF
      NOMAX=100
      NOUT=0
```

```
      WRITE(6,227)
  471 READ(5,6)   JROW,ICOL1,ICOL2
      CALL SWTCH(ICOL1,ICOL2)
      WRITE(6,6)   JROW,ICOL1,ICOL2
      IF(JROW)477,477,472
  472 DO 475   I=ICOL1,ICOL2
      IF(NODE(I,JROW)-DEFIN)475,473,475
  473 IF(NOUT-NOMAX)474,476,476
  474 NOUT=NOUT+1
      IO(NOUT)=I
      JO(NOUT)=JROW
  475 CONTINUE
      GO TO 471
  476 WRITE(6,229)NOMAX
      IERR=1
      GO TO 471
  477 WRITE(6,230)
  478 READ(5,6)   ICOL,JRO1,JRO2
      CALL SWTCH(JRO1,JRO2)
      WRITE(6,6)   ICOL,JRO1,JRO2
      IF(ICOL)485,485,479
  479 DO 482   J=JRO1,JRO2
      IF(NODE(ICOL,J)-DEFIN)482,480,482
  480 IF(NOUT-NOMAX)481,483,483
  481 NOUT=NOUT+1
      IO(NOUT)=ICOL
      JO(NOUT)=J
  482 CONTINUE
      GO TO 478
  483 WRITE(6,229)NOMAX
      IERR=1
      GO TO 478
  485 CONTINUE
C**
C**BEGIN SETTING UP ELEMENTS BY ROW
      NXCM1=NXC-1
      IEGNU=0
      MEMTO=0
      DO 1000   J=1,NORO
      MEMNO=0
      JP1=J+1
      DO 950   I=1,NXCM1
      IP1=I+1
      NI1=NODET(I,NODE,DEFIN,J,NXC,NONO(J))
      NI2=NODET(IP1,NODE,DEFIN,J,NXC,NONO(J))
      NI3=NODET(I,NODE,DEFIN,JP1,NXC,NONO(JP1))
      NI4=NODET(IP1,NODE,DEFIN,JP1,NXC,NONO(JP1))
      IF(J-NORO)605,675,675
C**
C**TEST FOR THICKNESS AREA
  605 DO 650   K=1,NTA
      IF(JT1(K)-J)610,610,650
  610 IF(JT2(K)-J)650,650,615
  615 IF(IT1(K)-I)620,620,650
  620 IF(IT2(K)-I)650,650,625
  625 T=THK(K)*CONVF
      GO TO 675
  650 CONTINUE
      T=0.
      IF(NODE(I,J)-DEFIN)651,675,651
```

```
      SUBROUTINE SWTCH(I1,I2)
      IF(I1-I2)99,99,10
   10 I=I1
      I1=I2
      I2=I
   99 RETURN
      END



       SUBROUTINE TEMPCO(        NC,IZ,SKA,XI,ZA)
      DIMENSION SKA(6,6),XI(6),ZA(6)
      DO 2 I=1,NC
      ZA(I)=0.0
      DO 2 J=1,IZ
    2 ZA(I)=ZA(I)+SKA(J,I)*XI(J)
      RETURN
      END



      SUBROUTINE TMATT(XB,NB,X,TM,NOCC,NCON,ODATA,SFS,SFL,SFA,E,G)
C...ROUTINE TO COMPUTE OCCURRENCE MATRIX FROM LOCATION
C...ZER TO X
      DIMENSION TM(5,5),TO(5,5),TR(5,5),PROP(5),ODATA(8,50)
      INFLU=0
      IF(NCON)110,110,100
  100 IF(X-XB)110,120,120
  110 CALL IDENT(TM,5)
      NB=1
      XB=0.
      NCON=1
  120 NBB=NB
      DO 300 N=NBB,NOCC
      X1=ODATA(6,N)
      X2=ODATA(7,N)
      IDF=ODATA(8,N)
      DO 130 I=1,5
  130 PROP(I)=ODATA(I,N)
C--CHECK IF POINT MATRIX HAS BEEN USED IN 1ST OCCURENCE
      IF(N-NBB)140,140,160
  140 IF(XB-X1)160,160,150
  150 IDF=1
  160 NB=N
      IF(X-X2)180,170,190
  170 NB=N+1
  180 NSTOP=1
      Y=X-XB
      XB=X
      GO TO 295
  190 NSTOP=0
      Y=X2-XB
      XB=X2
      GO TO 295
  295 CALL OCCM(TO,PROP,IDF,Y,INFLU,SFS,SFL,SFA,E,G)
      CALL MMULT(TO,TM,TR,5,5,5)
      CALL EQUAL(TM,TR,5,5)
      IF(NSTOP)300,300,400
```

```
 651   IF(NODE(IP1,JP1)-DEFIN)675,652,675
 652   IF(NODE(I,JP1)-DEFIN)675,653,675
 653   IF(NODE(IP1,J)-DEFIN)675,655,675
 655   WRITE(6,235)I,J
       IERR=1
C**
C**TEST FOR BAR ELEMENTS
 675   AX1=0.
       AX2=0.
       AX3=0.
       AX4=0.
       AX5=0.
       IF(NBAR)770,770,676
 676   DO 750  K=1,NBAR
C
C--HORIZONTAL BARS.....JB1=JB2
       IF(JB1(K)-JB2(K))688,680,688
C----BOTTOM BAR
 680   IF(JB1(K)-J)750,682,750
 682   IF(IB1(K)-I)684,684,750
 684   IF(IB2(K)-IP1)750,686,686
 686   AX2=AX(K)*CONVF*CONVF
       GO TO 750
 688   IF(J-NORO)696,750,750
C
C--VERTICAL BARS.....IB1=IB2
 696   IF(IB1(K)-IB2(K))714,698,714
C----LEFT HAND BAR
 698   IF(IB1(K)-I)706,700,706
 700   IF(JB1(K)-J)702,702,750
 702   IF(JB2(K)-JP1)750,704,704
 704   AX1=AX(K)*CONVF*CONVF
       GO TO 750
C----RIGHT HAND BAR
 706   IF(IB1(K)-IP1)750,708,750
 708   IF(JB1(K)-J)710,710,750
 710   IF(JB2(K)-JP1)750,712,712
 712   AX5=AX(K)*CONVF*CONVF
       GO TO 750
C
C--DIAGONAL BARS
 714   J1=JB1(K)
       J2=JB2(K)
       IF(IB1(K)-IB2(K))716,716,732
C----BAR BOTTOM LEFT TO TOP RIGHT
 716   IF(IB1(K)-I)718,718,750
 718   IF(IB2(K)-IP1)750,720,720
 720   II=IB1(K)-1
       DO 730  JJ=J1,J2
       II=II+1
       IF(JJ-J)730,722,750
 722   IF(II-I)750,724,750
 724   AX3=AX(K)*CONVF*CONVF
       GO TO 750
 730   CONTINUE
       GO TO 750
C----BAR TOP LEFT TO BOTTOM RIGHT
 732   IF(IB1(K)-IP1)750,734,734
 734   IF(IB2(K)-I)736,736,750
 736   II=IB1(K)+1
```

```
      DO 740  JJ=J1,J2
      II=II-1
      IF(JJ-J)750,738,740
 738  IF(II-IP1)750,739,750
 739  AX4=AX(K)*CONVF*CONVF
      GO TO 750
 740  CONTINUE
      GO TO 750
 750  CONTINUE
C
C--CHECK FOR LAST COLUMN
      IF(IP1-NXC)755,770,770
C--DO NOT INCLUDE RIGHT VERTICAL BAR UNLESS LAST COLUMN
 755  AX5=0.
C
C--CHECK NODES FOR OUTPUT SELECTION
 770  IFI=0
      IFJ=0
      IFK=0
      IFL=0
      NUL=0
      P=0.
      DO 800 K=1,NOUT
      IF(JO(K)-J)800,772,780
 772  IF(IO(K)-I)800,774,776
 774  IFI=1
      GO TO 800
 776  IF(IO(K)-IP1)800,778,800
 778  IFJ=1
      GO TO 800
 780  IF(JO(K)-JP1)800,782.800
 782  IF(IO(K)-I)800,784,786
 784  IFK=1
      GO TO 800
 786  IF(IO(K)-IP1)800,788,800
 788  IFL=1
 800  CONTINUE
      IF(J-NORD)805,900,900
C
C--CHECK NODE DEFINITIONS FOR VOIDS
 805  IF(NODE(I,J)-DEFIN)830,810,830
 810  IF(NODE(IP1,J)-DEFIN)840,812,840
 812  IF(NODE(I,JP1)-DEFIN)850,814,850
 814  IF(NODE(IP1,JP1)-DEFIN)860,816,860
C
C--QUADRILATERAL PLATE ELEMENT
 816  CONTINUE
      IF (NCARD.EQ.0) GO TO 10
      KODE=1
      GO TO 836
  10  MEMNO=MEMNO+1
      MEMTYP=2
      WRITE(8,250)    MEMNO,MEMTYP,IEGNU,IESF.IFI,IFJ,IFK,IFL,
     +J,NI1,J,NI2,JP1,NI3,JP1,NI4,T,P,P,P,P,P
      GO TO 900
C-- J,IP1 VOID - NO PLATE POSSIBLE
 822  AX2=0.
      AX4=0.
      AX5=0.
      GO TO 900
```

```
C-- JP1,I VOID - NO PLATE POSSIBLE
 824  AX1=0.
      AX4=0.
      GO TO 900
C-- JP1,IP1 VOID - NO PLATE POSSIBLE
 826  AX3=0.
      AX5=0.
      GO TO 900
C..(I,J) VOID
 830  AX1=0.
      AX2=0.
      AX3=0.
      IF(NODE(IP1,J)-DEFIN)822,832,822
 832  IF(NODE(IP1,JP1)-DEFIN)826,834,826
 834  IF(NODE(I,JP1)-DEFIN)824,836,824
C--TRI-PLATE UPPER RIGHT
 836  MEMNO=MEMNO+1
      MEMTYP=1
      WRITE(8,250)     MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFK,IFL,NUL,
     +J,NI2,JP1,NI3,JP1,NI4,NUL,NUL,T,P,P,P,P,P
      IF (KODE.EQ.1) GO TO 860
      GO TO 900
C..(I,J) DEF., (IP1,J) VOID
 840  AX2=0.
      AX4=0.
      AX5=0.
      IF(NODE(IP1,JP1)-DEFIN)826,842,826
 842  IF(NODE(I,JP1)-DEFIN)824,844,824
C--TRI-PLATE UPPER LEFT
 844  MEMNO=MEMNO+1
      MEMTYP=1
      WRITE(8,250)     MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFK,IFL,NUL,
     +J,NI1,JP1,NI3,JP1,NI4,NUL,NUL,T,P,P,P,P,P
      GO TO 900
C..(I,J),(IP1,J) DEF., (I,JP1) VOID
 850  AX1=0.
      AX4=0.
      IF(NODE(IP1,JP1)-DEFIN)826,852,826
C--TRI-PLATE LOWER RIGHT
 852  MEMNO=MEMNO+1
      MEMTYP=1
      WRITE(8,250)     MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFL,NUL,
     +J,NI1,J,NI2,JP1,NI4,NUL,NUL,T,P,P,P,P,P
      GO TO 900
C--TRI-PLATE LOWER LEFT
 860  MEMNO=MEMNO+1
      AX3=0.
      AX5=0.
      MEMTYP=1
      WRITE(8,250)     MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,IFK,NUL,
     +J,NI1,J,NI2,JP1,NI3,NUL,NUL,T,P,P,P,P,P
      KODE=0
      GO TO 900
C--BAR ELEMENTS
 900  MEMTYP=5
      IF(AX1)904,904,902
 902  MEMNO=MEMNO+1
      WRITE(8,250)     MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFK,NUL,NUL,
     +J,NI1,JP1,NI3,NUL,NUL,NUL,NUL,AX1,P,P,P,P,P
 904  IF(AX2)908,908,906
```

```
  906 MEMNO=MEMNO+1
      WRITE(8,250)    MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFJ,NUL,NUL,
     +J,NI1,J,NI2,NUL,NUL,NUL,NUL,AX2,P,P,P,P,P
  908 IF(AX3)912,912,910
  910 MEMNO=MEMNO+1
      WRITE(8,250)    MEMNO,MEMTYP,IEGNU,IFSF,IFI,IFL,NUL,NUL,
     +J,NI1,JP1,NI4,NUL,NUL,NUL,NUL,AX3,P,P,P,P,P
  912 IF(AX4)916,916,914
  914 MEMNO=MEMNO+1
      WRITE(8,250)    MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFK,NUL,NUL,
     +J,NI2,JP1,NI3,NUL,NUL,NUL,NUL,AX4,P,P,P,P,P
  916 IF(AX5)950,950,918
  918 MEMNO=MEMNO+1
      WRITE(8,250)    MEMNO,MEMTYP,IEGNU,IFSF,IFJ,IFL,NUL,NUL,
     +J,NI2,JP1,NI4,NUL,NUL,NUL,NUL,AX5,P,P,P,P,P
  950 CONTINUE
      MEMTO=MEMTO+MEMNO
      IF(MEMNO)975,975,1000
  975 WRITE(8,250)    NUL,NUL,NUL,NUL,NUL,NUL,NUL,NUL,
     +J,NUL,NUL,NUL,NUL,NUL,NUL,NUL,P,P,P,P,P,P
 1000 CONTINUE
      J=NORO+1
      WRITE(8,250)    NUL,NUL,NUL,NUL,NUL,NUL,NUL,NUL,
     +J,NUL,NUL,NUL,NUL,NUL,NUL,NUL,P,P,P,P,P,P
      CALL REWIND (8)
      WRITE(6,281)MEMTO
      IF(IERR)9999,9999,9998
 9998 STOP
 9999 RETURN
    1 FORMAT (2I5)
    2 FORMAT (F15.5)
    3 FORMAT(4I5)
    4 FORMAT (I5)
    5 FORMAT(F15.5,4I5)
    6 FORMAT(3I5)
  200 FORMAT(1H1,'TRANSVERSE FINITE ELEMENT DEFINITION')
  201 FORMAT(/' X-LBHD,DEPTH CL, DEPTH SHELL')
  202 FORMAT(/,' NO. X-COORDINATES(40), NO. Y-COORDINATES(25)')
  203 FORMAT(/,' LIST X-COORDINATES'/' COL',4X,'X')
  204 FORMAT(/,' LIST Y-COORDINATES'/' ROW',4X,'Y')
  205 FORMAT(1H1,/,' DEFINITION OF VOIDS'/
     +' LIST ROW1,ROW2,COL1,COL2')
  206 FORMAT(/,I3,1X,25(I1,2X))
  207 FORMAT(//,1X,'C',/,1X,'O',5X,'ROW',/,1X,'L',25I3/)
  208 FORMAT(I4,F12.3)
  210 FORMAT(1H1,//' UNITS OF FORCE')
  211 FORMAT(' UNITS OF LENGTH')
  212 FORMAT(5I5,4A6)
  214 FORMAT(E10.2,F7.2,E10.2)
  215 FORMAT(25I3)
  216 FORMAT(3F10.2)
  220 FORMAT(1H1,/,' PLATE THICKNESS DEFINITION')
  221 FORMAT(' NO. AREAS OF COMMON THICKNESS (50)')
  222 FORMAT(' LIST T,ROW1,ROW2,COL1,COL2')
  223 FORMAT(1H1,/,' BAR ELEMENT DEFINITION')
  224 FORMAT(' NO. BAR ELEMENTS (100)')
  225 FORMAT(' LIST AX,ROW1,ROW2,COL1,COL2')
  226 FORMAT(1H1,/,' OUTPUT SPECIFICATIONS')
  227 FORMAT(/,' SELECTED NODES BY HORIZONTAL SEQUENCE',/,
     +' ROW, COL1, COL2')
```

```
228  FORMAT(' ENTER (1)NODE FORCE ONLY'/7X'(2)FORCE AND STRESS'
    +/7X'(3)STRESS ONLY')
229  FORMAT(' **MAX.NODES FOR OUTPUT SET TO'I8)
230  FORMAT(/,' SELECTED NODES BY VERTICAL SEQUENCE',/,
    +' COLUMN, ROW1, ROW2')
235  FORMAT(' **ERROR-THICKNESS NOT DEFINED FOR ROW',I3,', COL'
    +,I3)
250  FORMAT(I3,7I1,8I2,6E15.7)
251  FORMAT(/,I3,I2,24I3)
260  FORMAT(' NOTE-THICKNESS AREAS FOR BOTTOM ELEMENTS SHOULD'
    +,/,' BE ENTERED FIRST')
261  FORMAT(1X,F12.3,4I6,3X,'(',I2,')')
262  FORMAT(' **ERROR-FOLLOWING BAR ELEMENT INTERSECTS BETWEEN'
    +' NODES')
272  FORMAT(/' BOUNDARY CONDITIONS'/,'   RESTRICTED X-DEFLECTION =1'/,
    +'   RESTRICTED Y-DEFLECTION =0')
273  FORMAT(' RESTRICTION C.L. SUPPORTS, BOTTOM SUPPORTS')
274  FORMAT(20I3)
281  FORMAT(//' YOU HAVE JUST GENERATED',I8,' ELEMENTS')
282  FORMAT(1H1,' ROW NODE NUMBERING SYSTEM'/)
289  FORMAT(/,' BOUNDARY CONDITION WEIGHTING FACTOR')
300  FORMAT(3F15.5)
301  FORMAT(2A6)
311  FORMAT(/,' THERE ARE',I3,' C.L. SUPPORTS')
312  FORMAT(/,' C.L. SUPPORTS ARE DEFINED FOR ROWS'/,
    +25I3)
313  FORMAT(/' E =',F15.1,/,' GNU =',F10.3,/,' ALPHA =',
    +F10.6)
314  FORMAT(/,' SUPPORT AT BOTTOM ON COLUMN'I3,'(NODE'I2,
    +')')
     END
```

```
CZZZZI    FR5  IDENTS,IDENTS,IDENT
      SUBROUTINE IDENT(T,N)
      DIMENSION T(N,N)
      DO 5 I=1,N
      DO 4 J=1,N
    4 T(I,J)=0.
    5 T(I,I)=1.
      RETURN
      END
```

```
CZZZZI    FR5  EQUALS,EQUALS,EQUAL
      SUBROUTINE EQUAL(T,TR,NR,NC)
      DIMENSION T(NR,NC),TR(NR,NC)
      DO 5 I=1,NR
      DO 5 J=1,NC
    5 T(I,J)=TR(I,J)
      RETURN
      END
```

```
CZZZZI    FR5  MMULTS,MMULTS,MMULT
      SUBROUTINE MMULT(A,B,C,N1,N2,N3)
      DIMENSION A(N1,N2),B(N2,N3),C(N1,N3)
      DO 1 I=1,N1
      DO 1 J=1,N3
      C(I,J)=0.
      DO 1 K=1,N2
    1 C(I,J)=C(I,J)+A(I,K)*B(K,J)
      RETURN
      END
```

```
CZZZZIE  FR5 MULT,MULT,MULT
       SUBROUTINE  MULT (A,B,C,L,M)
       DIMENSION  A(L,L),B(L,L),C(L,L)
       DO  2  I=1,M
          DO 2 J=1,M
           A(I,J)=0.0
       DO  1  K=1,M
     1 A(I,J)=A(I,J)+B(I,K)*C(K,J)
     2 CONTINUE
      RETURN
      END
```

```
CZZZZIE  FR5 TRANS,TRANS,TRANS
         SUBROUTINE TRANS(A,B,M,N)
        DIMENSION A(M,M),B(M,M)
        DO 1 I=1,N
        DO 1 J=1,N
     1 A(I,J)=B(J,I)
        RETURN
        END
```

```
C    TRANSVERSE PROGRAM.  CHAZAL-PAYNE  APRIL 1971
C     THIS REPRESENTS A SIMPLE TEST DATA SET TO BE USED AS INPUT DATA FOR TH

1
  0.1
10000.
    3
    2.      E+09    .0
     2.05   E+06     .3
  1.
  1.
  1.
    1
    2
    1
  1000.                2000.               2000.
    5     5
 0.0
  500.
  1000.
  1500.
  2000.
 0.0
  500.
  1000.
  1500.
  2000.
   00
    KG
    CM
    1
  1.0
    1
     16.0              1      5      1      5
    4
  10000.               5      5      1      5
  10000.               1      1      1      5
  10000.               1      5      1      1
  10000.               1      5      3      3
    2
    2      1      5
   00
    4      1      5
   00
     3.0    E+09     2.0    E+09     75000.         5      2      2
     3.0    E+09     2.0    E+09     75000.         5      4      5
     3.0    E+09     2.0    E+09     75000.         1      2      2
     3.0    E+09     2.0    E+09     75000.         1      4      5
        0.  E+00        0.  E+00     0.
     3.0    E+09      2.0   E+09     75000.         3      2      4
     3.0    E+09      2.0   E+09     75000.         1      2      4
     0.     E+00       0.   E+00     0.
    1
    0      3      2
        0.  E+00
        0.  E+00
```

```
 0.25      E+04
  -0.4     E-01
  7.5      E+03
      0.   E+00
   750.             1     1     4     4
   250.             1     1     5     5
 00
 00
  3     4
 00
  840.            24.5              19.
  840.            24.5              19.
  840.            24.5              19.

C    THIS DATA IS AVAILABLE AS PUNCHED OUTPUT FROM LONGITUDINAL PROGRAM

 0.96020 E+05    -0.4959 E+05
 -0.16048E+06     0.27476E+06
 0.96020 E+05    -0.4959 E+05
1
```

TRANSVERSE STRENGTH ANALYSIS OF LONGITUDINALLY FRAMED SHIPS

****

CONVERSION FACTOR TO BE APPLIED TO ALL DIMENSIONAL DATA
INCLUDING COORDINATES, PLATE THICKNESS, BAR AREA
BUT NOT INCLUDING YOUNGS MODULUS
0.10000

LENGTH OF LONGITUDINALS
10000.00000

NO. TRANSVERSES ALONG LENGTH
3

STANDARD LONGITUDINAL
MOMENT OF INERTIA, SHEAR AREA
0.20000E 10 0.0

YOUNGS MODULUS, POISSONS RATIO
0.20500E 07 0.30000

LIST STIFFENESS FACTORS OF ALL TRANSVERSES IN ORDER FROM STERN
1.00000
1.00000
1.00000

NO. TRANSVERSES TO BE ANALYZED(5)
1

LIST TRANSVERSES TO BE ANALYZED BY POSITION FROM STERN
2

NUMBER OF EIGENVALUES TO BE USED
1

TRANSVERSE FINITE ELEMENT DEFINITION

X-LBHD,DEPTH CL, DEPTH SHELL
1000.00000 2000.00000 2000.00000

NO. X-COORDINATES(40), NO. Y-COORDINATES(25)
5 5

LIST X-COORDINATES

COL	X
1	0.0
2	500.000
3	1000.000
4	1500.000
5	2000.000

LIST Y-COORDINATES

ROW	Y
1	0.0
2	500.000
3	1000.000
4	1500.000
5	2000.000

```
DEFINITION OF VOIDS
LIST ROW1,ROW2,COL1,COL2
   0    0    0    0
```

COL	ROW 1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1

UNITS OF FORCE
KG
UNITS OF LENGTH
CM

BOUNDARY CONDITIONS
RESTRICTED X-DEFLECTION =1
RESTRICTED Y-DEFLECTION =0
RESTRICTION C.L. SUPPORTS, BOTTOM SUPPORTS
1 0

BOUNDARY CONDITION WEIGHTING FACTOR
1.00000

THERE ARE 5 C.L. SUPPORTS

C.L. SUPPORTS ARE DEFINED FOR ROWS
1 2 3 4 5

SUPPORT AT BOTTOM ON COLUMN 3(NODE 3)

E = 2050000.0
GNU = 0.300
ALPHA = 0.0

ROW NODE NUMBERING SYSTEM

COL	ROW 1	2	3	4	5
1	1	1	1	1	1
2	2	2	2	2	2
3	3	3	3	3	3
4	4	4	4	4	4
5	5	5	5	5	5

```
PLATE THICKNESS DEFINITION
NO. AREAS OF COMMON THICKNESS (50)
   1
NOTE-THICKNESS AREAS FOR BOTTOM ELEMENTS SHOULD
BE ENTERED FIRST
LIST T,ROW1,ROW2,COL1,COL2
      16.000      1      5      1      5   ( 1)
```

PLATE THICKNESS DEFINITION

COL	ROW 1	2	3	4	5
1	1	1	1	1	1
2	1	1	1	1	1
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1

```
BAR ELEMENT DEFINITION
NO. BAR ELEMENTS (100)
    4
LIST AX,ROW1,ROW2,COL1,COL2
    10000.000       5       5       1       5     (  1)
    10000.000       1       1       1       5     (  2)
    10000.000       1       5       1       1     (  3)
    10000.000       1       5       3       3     (  4)
```

```
OUTPUT SPECIFICATIONS
ENTER (1)NODE FORCE ONLY
      (2)FORCE AND STRESS
      (3)STRESS ONLY
    2

SELECTED NODES BY HORIZONTAL SEQUENCE
ROW, COL1, COL2
    2     1     5
    0     0     0

SELECTED NODES BY VERTICAL SEQUENCE
COLUMN, ROW1, ROW2
    4     1     5
    0     0     0
  12020011 1 1 1 2 2 1 2 2   0.1599999E 01
  25020100 1 1 2 1 0 0 0 0   0.9999991E 02
  35020000 1 1 1 2 0 0 0 0   0.9999991E 02
  42020011 1 2 1 3 2 2 2 3   0.1599999E 01
  55020000 1 2 1 3 0 0 0 0   0.9999991E 02
  62020111 1 3 1 4 2 3 2 4   0.1599999E 01
  75020100 1 3 2 3 0 0 0 0   0.9999991E 02
  85020100 1 3 1 4 0 0 0 0   0.9999991E 02
  92021011 1 4 1 5 2 4 2 5   0.1599999E 01
 105021000 1 4 1 5 0 0 0 0   0.9999991E 02
  12021100 2 1 2 2 3 1 3 2   0.1599999E 01
  25021000 2 1 3 1 0 0 0 0   0.9999991E 02
  32021100 2 2 2 3 3 2 3 3   0.1599999E 01
  42021101 2 3 2 4 3 3 3 4   0.1599999E 01
  55021000 2 3 3 3 0 0 0 0   0.9999991E 02
  62021110 2 4 2 5 3 4 3 5   0.1599999E 01
  12020000 3 1 2 2 4 1 4 2   0.1599999E 01
  25020000 3 1 4 1 0 0 0 0   0.9999991E 02
  32020000 3 2 3 3 4 2 4 3   0.1599999E 01
  42020101 3 3 3 4 4 3 4 4   0.1599999E 01
  55020000 3 3 4 3 0 0 0 0   0.9999991E 02
  62021010 3 4 3 5 4 4 4 5   0.1599999E 01
  12020000 4 1 4 2 5 1 5 2   0.1599999E 01
  25020000 4 1 5 1 0 0 0 0   0.9999991E 02
  32020000 4 2 4 3 5 2 5 3   0.1599999E 01
  42020101 4 3 4 4 5 3 5 4   0.1599999E 01
  55020000 4 3 5 3 0 0 0 0   0.9999991E 02
  62021010 4 4 4 5 5 4 5 5   0.1599999E 01
  15020000 5 1 5 2 0 0 0 0   0.9999991E 02
  25020000 5 2 5 3 0 0 0 0   0.9999991E 02
  35020100 5 3 5 4 0 0 0 0   0.9999991E 02
  45021000 5 4 5 5 0 0 0 0   0.9999991E 02
  00000000 6 0 0 0 0 0 0 0   0.0                  0.0

YOU HAVE JUST GENERATED       32 ELEMENTS
```

DEFINITIONS OF LONGITUDINALS

IX = MOMENT OF INERTIA OF LONGIT-L BENDING IN X-DIRECTION OF TRANSVERSE
IY = MOMENT OF INERTIA OF LONGIT-L BENDING IN Y-DIRECTION OF TRANSVERSE

LIST BY HORIZONTAL SEQUENCE
IX,IY,A, ROW, COL1, COL2

0.300000E 10	0.200000E 10	75000.000000	5	2	2
0.300000E 10	0.200000E 10	75000.000000	5	4	5
0.300000E 10	0.200000E 10	75000.000000	1	2	2
0.300000E 10	0.200000E 10	75000.000000	1	4	5
0.0	0.0	0.0	0	0	0

LIST BY VERTICAL SEQUENCE
IX,IY,A, COLUMN, ROW1,ROW2

0.300000E 10	0.200000E 10	75000.000000	3	2	4
0.300000E 10	0.200000E 10	75000.000000	1	2	4
0.0	0.0	0.0	0	0	0

THERE ARE A TOTAL OF 12 LONGITUDINALS

LONGITUDINAL NUMBERING SYSTEM

C
O
L

ROW 1	2	3	4	5
0	6	11	16	0
2	0	0	0	22
0	8	13	18	0
4	0	0	0	24
5	0	0	0	25

EIGENVALUES AND EIGENVECTORS

......
0.278438E 01
0.500000E 00   0.707107E 00   0.500000E 00

......
0.176470E 00
-0.707108E 00   0.101763E-05   0.707106E 00

......
0.391455E-01
0.499999E 00   -0.707107E 00   0.500001E 00

SCALED EIGENVALUES
0.10022E-03   0.63516E-05   0.14089E-05

DIAGONAL OF MATRIX DP
0.10022E-03   0.63516E-05   0.14090E-05

LOADING CONDITION

1. A LOAD SET IS A SET OF LOADS ACTING IN A GIVEN X OR Y DIRECTION. THE EXTENT OF THESE LOADS IS ALONG THE LENGTH OF A GIVEN LONGITUDINAL

2. ANY LONGITUDINAL MAY BE LOADED WITH ANY NUMBER OF LOAD SETS, WHICH MAY BE ONLY PARTIALLY APPLIED VIA A PROPORTIONAL FACTOR

3. LOCATIONS OF LOADS ARE DISTANCES MEASURED FROM THE STERN

4. LOAD SET DIRECTION CODES ARE
   =1, X-DIRECTION
   =2, Y-DIRECTION

NUMBER OF LOAD SETS
   1

```
---LOAD SET     1
NO. CONC. LOADS, NO. UNIF. LOADS, DIRECTION CODE
    0    3    2

LIST START LOCATION, Q
    0.0
    0.0
    0.25000E 04
   -0.40000E-01
    0.75000E 04
    0.0

COMPUTED DEFLECTIONS AT TRANSVERSES
    1    -0.63516E-02
    2    -0.90510E-02
    3    -0.63516E-02

LIST LONGITUDINALS SO LOADED
FACTOR, ROW1, ROW2, COL1, COL2
      750.00000     1     1     4     4
      250.00000     1     1     5     5
        0.0         0     0     0     0

MISC. LOADINGS
NP,NQ, NDIR, ROW, COLUMN
    0     0     0     0     0
```

SHEAR LOADS ON TRANSVERSES

BHD COLUMN NUMBER, NUMBER OF ROWS FOR SHEAR
3 4

WEB LENGTH, SHELL, BHD THICKNESS (CM)

0.0	0.0	0.0
84.00000	2.45000	1.90000
84.00000	2.45000	1.90000
84.00000	2.45000	1.90000

```
SHELL, BHD SHEARS PER TRANSVERSE STARTING FROM STERN
   0.96020E 05    -0.49590E 05
  -0.16048E 06     0.27476E 06
   0.96020E 05    -0.49590E 05

ENTER 0 TO STOP PROGRAM HERE.  ENTER 1 TO GO ON

IDENTQ VALUE IS     1
```

NODE COORDINATES

ROW	NODE	X-COORD	Y-COORD
1	1	0.0	0.0
1	2	50.00000	0.0
1	3	100.00000	0.0
1	4	150.00000	0.0
1	5	200.00000	0.0
2	1	0.0	50.00000
2	2	50.00000	50.00000
2	3	100.00000	50.00000
2	4	150.00000	50.00000
2	5	200.00000	50.00000
3	1	0.0	100.00000
3	2	50.00000	100.00000
3	3	100.00000	100.00000
3	4	150.00000	100.00000
3	5	200.00000	100.00000
4	1	0.0	150.00000
4	2	50.00000	150.00000
4	3	100.00000	150.00000
4	4	150.00000	150.00000
4	5	200.00000	150.00000
5	1	0.0	200.00000
5	2	50.00000	200.00000
5	3	100.00000	200.00000
5	4	150.00000	200.00000
5	5	200.00000	200.00000

MEMBER DATA

MEMNO	MEMTYP	IFI	IFJ	IFK	IFL	INI	JNI	INJ	JNJ	INK	JNK	INL	JNL	P1
1	2	0	0	1	1	1	1	1	2	2	1	2	2	1.60000
2	5	0	1	0	0	1	1	2	1	0	0	0	0	99.99991
3	5	0	0	0	0	1	1	1	2	0	0	0	0	99.99991
4	2	0	0	1	1	1	2	1	3	2	2	2	3	1.60000
5	5	0	0	0	0	1	2	1	3	0	0	0	0	99.99991
6	2	0	1	1	1	1	3	1	4	2	3	2	4	1.60000
7	5	0	1	0	0	1	3	2	3	0	0	0	0	99.99991
8	5	0	1	0	0	1	3	1	4	0	0	0	0	99.99991
9	2	1	0	1	1	1	4	1	5	2	4	2	5	1.60000
10	5	1	0	0	0	1	4	1	5	0	0	0	0	99.99991
1	2	1	1	0	0	2	1	2	2	3	1	3	2	1.60000
2	5	1	0	0	0	2	1	3	1	0	0	0	0	99.99991
3	2	1	1	0	0	2	2	2	3	3	2	3	3	1.60000
4	2	1	1	0	1	2	3	2	4	3	3	3	4	1.60000
5	5	1	0	0	0	2	3	3	3	0	0	0	0	99.99991
6	2	1	1	1	0	2	4	2	5	3	4	3	5	1.60000
1	2	0	0	0	0	3	1	3	2	4	1	4	2	1.60000
2	5	0	0	0	0	3	1	4	1	0	0	0	0	99.99991
3	2	0	0	0	0	3	2	3	3	4	2	4	3	1.60000
4	2	0	1	0	1	3	3	3	4	4	3	4	4	1.60000
5	5	0	0	0	0	3	3	4	3	0	0	0	0	99.99991
6	2	1	0	1	0	3	4	3	5	4	4	4	5	1.60000
1	2	0	0	0	0	4	1	4	2	5	1	5	2	1.60000
2	5	0	0	0	0	4	1	5	1	0	0	0	0	99.99991
3	2	0	0	0	0	4	2	4	3	5	2	5	3	1.60000
4	2	0	1	0	1	4	3	4	4	5	3	5	4	1.60000
5	5	0	0	0	0	4	3	5	3	0	0	0	0	99.99991
6	2	1	0	1	0	4	4	4	5	5	4	5	5	1.60000
1	5	0	0	0	0	5	1	5	2	0	0	0	0	99.99991
2	5	0	0	0	0	5	2	5	3	0	0	0	0	99.99991
3	5	0	1	0	0	5	3	5	4	0	0	0	0	99.99991
4	5	1	0	0	0	5	4	5	5	0	0	0	0	99.99991
0	0	0	0	0	0	6	0	0	0	0	0	0	0	0.0

```
DATA FOR PROBLEM NUMBER   7777
7777    5    2    1    0    KG              CM
 5  1  1  0
 1  2  3  4  5
 3  2
 0.10E 01
 0.20E 07    0.30   0.0
 5  5  5  5  5
  22    24    25    2    4    5    8   13   18    6
  11    16
         1          32          1          3
   -0.5147E-02   -0.1190E-01   -0.5442E-02   -0.1382E-01
   -0.7541E-02   -0.1788E-08   -0.6310E-02   -0.8341E-01
   -0.7227E-09   -0.8032E-01
```

DEFLECTION OF LONGITUDINAL 1 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.41908E-02   0.41908E-02

Y DEFLECTIONS

-0.95904E-02   -0.95904E-02

DEFLECTION OF LONGITUDINAL 2 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.22039E-02   0.22039E-02

Y DEFLECTIONS

-0.17856E-01   -0.17856E-01

DEFLECTION OF LONGITUDINAL 3 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.24190E-09   0.24190E-09

Y DEFLECTIONS

-0.19843E-01   -0.19843E-01

DEFLECTION OF LONGITUDINAL 4 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

-0.27212E-02 -0.27212E-02

Y DEFLECTIONS

-0.69410E-02 -0.69410E-02

DEFLECTION OF LONGITUDINAL 5 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

-0.31550E-02 -0.31550E-02

Y DEFLECTIONS

-0.41703E-01 -0.41703E-01

DEFLECTION OF LONGITUDINAL 6 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

-0.36133E-09 -0.36133E-09

Y DEFLECTIONS

-0.40161E-01 -0.40161E-01

DEFLECTION OF LONGITUDINAL 7 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.26321E-02 0.26321E-02

Y DEFLECTIONS

-0.85547E-02 -0.85547E-02

DEFLECTION OF LONGITUDINAL 8 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.28638E-02 0.28638E-02

Y DEFLECTIONS

-0.11813E-01 -0.11813E-01

DEFLECTION OF LONGITUDINAL 9 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.28850E-02 0.28850E-02

Y DEFLECTIONS

-0.12972E-01 -0.12972E-01

DEFLECTION OF LONGITUDINAL 10 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

-0.10056E-02 -0.10056E-02

Y DEFLECTIONS

-0.61349E-02 -0.61349E-02

DEFLECTION OF LONGITUDINAL 11 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.69279E-03 0.69279E-03

Y DEFLECTIONS

-0.63144E-02 -0.63144E-02

DEFLECTION OF LONGITUDINAL 12 AT EVERY OTHER TRANSVERSE

X DEFLECTIONS

0.26294E-02 0.26294E-02

Y DEFLECTIONS

-0.66346E-02 -0.66346E-02

REAL LOADS UPON THE TRANSVERSES

-0.62724E 02	-0.32987E 02	-0.36205E-05	0.40729E 02	0.47222E 02
-0.42864E 02	-0.43181E 02	0.15051E 02	-0.10369E 02	-0.39355E 02
0.95695E 02	0.17817E 03	0.19800E 03	0.69258E 02	-0.34404E 05
0.16649E 05	0.16661E 05	-0.31946E 05	-0.31944E 05	0.66201E 02
-0.88707E 02	-0.46651E 02	-0.51203E-05	0.57601E 02	0.66782E 02
-0.60619E 02	-0.61068E 02	0.21285E 02	-0.14665E 02	-0.55657E 02
0.13533E 03	0.25197E 03	0.28002E 03	0.97948E 02	-0.85128E 05
-0.91421E 05	-0.91405E 05	0.53580E 05	0.53583E 05	0.93626E 02
-0.62724E 02	-0.32986E 02	-0.36205E-05	0.40729E 02	0.47222E 02
-0.42865E 02	-0.43181E 02	0.15051E 02	-0.10369E 02	-0.39355E 02
0.95695E 02	0.17817E 03	0.19800E 03	0.69258E 02	-0.34402E 05
0.16649E 05	0.16660E 05	-0.31946E 05	-0.31944E 05	0.66201E 02

REAL LOADS UPON THE TRANSVERSES

TRANSVERSE NO. 2

X-FORCES TYPICAL

-0.88707E 02	-0.46651E 02	-0.51203E-05	0.57601E 02	0.66782E 02
-0.60619E 02	-0.61068E 02	0.21285E 02	-0.14665E 02	-0.55657E 02

Y-FORCES TYPICAL

0.13533E 03	0.25197E 03	0.28002E 03	0.97948E 02	-0.85128E 05
-0.91421E 05	-0.91405E 05	0.53580E 05	0.53583E 05	0.93626E 02

1	32 1	3	
-0.3369E-01	0.2487E-01	-0.3135E-01	-0.9420E-02
-0.2649E-01	-0.4045E-08	-0.1613E-01	-0.1131E 00
-0.1639E-08	-0.1126E 00		

MEMBER STRESSES IN KG PER SQUARE CM

ROW	MEMBER TYPE AND NUMBER		X-STRESS (TRIANG PLATE) / X-STRESS (QUAD PLATE) / X-STRESS (BAR)	Y-STRESS / X-GRAD	SHEAR STRESS / Y-STRESS	1ST PRINC STR / Y-GRAD	2ND PRINC STR / SHEAR STRESS	ANG STR
1	QUAD PLATE	1	0.123152E 03	-0.222229E 01	0.107625E 03	-0.358900E 00	-0.320006E 03	
1	BAR	2	0.873468E 02					
1	BAR	3	0.935565E 02					
1	QUAD PLATE	4	-0.396903E 02	0.768242E 01	0.115113E 03	-0.358310E 02	0.639529E 02	
1	BAR	5	0.194508E 03					
1	QUAD PLATE	6	0.354897E 03	-0.986928E 00	-0.162308E 04	0.570058E 02	-0.863481E 03	
1	BAR	7	-0.172215E 04					
1	BAR	8	0.414278E 03					
1	QUAD PLATE	9	0.898370E 03	-0.165591E 02	0.127345E 04	-0.171835E 02	0.541458E 00	
1	BAR	10	0.645210E 03					
2	QUAD PLATE	1	-0.832408E 02	0.305032E 01	-0.781944E 02	-0.562990E 01	-0.337212E 03	
2	BAR	2	-0.760996E 02					
2	QUAD PLATE	3	0.431056E 03	-0.790029E 01	-0.287530E 03	-0.817517E 01	-0.417941E 03	
2	QUAD PLATE	4	0.292595E 03	-0.557664E 01	-0.721149E 03	0.192011E 02	-0.384406E 03	
2	BAR	5	-0.766353E 03					
2	QUAD PLATE	6	-0.630351E 02	-0.102272E 01	0.167119E 03	0.927667E 01	-0.224874E 03	

3	BAR	2	-0.291269E 03				
3	QUAD PLATE	3	0.114820E 03	-0.513956E 01	-0.202429E 03	-0.107545E 01	-0.535979E 03
3	QUAD PLATE	4	0.505080E 02	-0.538672E 01	-0.277349E 03	0.701484E 01	-0.132087E 03
3	BAR	5	-0.252101E 03				
3	QUAD PLATE	6	-0.197285E 03	-0.118032E 01	0.306030E 02	0.365557E 01	-0.990100E 02
4	QUAD PLATE	1	-0.143525E 02	-0.206385E 01	-0.103865E 03	0.422200E 00	-0.307030E 03
4	BAR	2	-0.840802E 02				
4	QUAD PLATE	3	-0.787172E 02	-0.507384E 01	-0.124639E 03	0.427182E 01	-0.453643E 03
4	QUAD PLATE	4	-0.181982E 03	-0.564692E 01	0.536743E 02	-0.131335E 01	-0.275325E 02
4	BAR	5	0.150621E 03				
4	QUAD PLATE	6	-0.288258E 03	-0.363629E 01	-0.287964E 02	0.177049E 01	-0.232211E 02
5	BAR	1	-0.895526E 02				
5	BAR	2	-0.327056E 03				
5	BAR	3	-0.470581E 03				
5	BAR	4	-0.474713E 03				

CUT NODE FORCES IN KG

LOAD SYSTEM	ROW	NODE	X-FORCE	Y-FORCE
1	1	4	-0.666875E 02	0.851277E 05
1	2	1	-0.213008E 02	-0.535302E 05
1	2	2	-0.312500E-01	-0.546875E-01
1	2	3	0.555977E 02	0.914663E 05
1	2	4	-0.156250E-01	-0.199219E 00
1	2	5	-0.196695E 05	-0.132813E 00
1	3	4	-0.781250E-01	-0.503906E 00
1	4	4	-0.156250E-01	-0.238281E 00
1	5	4	0.467031E 02	-0.251861E 03

NODE DISPLACEMENTS IN CM

LOAD SYSTEM	ROW	NODE	X-DISP	Y-DISP
1	1	1	-0.336888E-01	0.248744E-01
1	1	2	-0.313499E-01	-0.942021E-02
1	1	3	-0.264872E-01	-0.404521E-08
1	1	4	-0.161303E-01	-0.113071E 00
1	1	5	-0.163923E-08	-0.112645E 00
1	2	1	-0.185814E-01	0.270581E-01
1	2	2	-0.190204E-01	-0.768516E-02
1	2	3	-0.455466E-02	-0.430537E-01
1	2	4	0.456863E-02	-0.848670E-01
1	2	5	-0.676641E-09	-0.105921E 00
1	3	1	-0.414483E-02	0.251556E-01
1	3	2	-0.770880E-03	-0.166250E-01
1	3	3	0.381948E-02	-0.622125E-01
1	3	4	0.584703E-02	-0.800244E-01
1	3	5	0.605363E-09	-0.894825E-01
1	4	1	0.135214E-01	0.173739E-01
1	4	2	0.138624E-01	-0.215832E-01
1	4	3	0.120283E-01	-0.685151E-01
1	4	4	0.732242E-02	-0.775585E-01
1	4	5	0.909923E-09	-0.824471E-01
1	5	1	0.340475E-01	0.157719E-01
1	5	2	0.318087E-01	-0.231575E-01
1	5	3	0.236323E-01	-0.647495E-01
1	5	4	0.118678E-01	-0.754347E-01
1	5	5	0.118784E-08	-0.781102E-01

STOP 0
EXECUTION TERMINATED