DYNAMIC ANALYSIS OF MULTISTORY BUILDINGS
BY COMPONENT MODE SYNTHESIS

RESEARCH REPORT

SETEC-CE-85-008

BY

MORTEZA A. M. TORKAMANI

JUI TIEN HUANG

UNIVERSITY OF PITTSBURGH

Department of Civil Engineering

November 1984

NATIONAL SCIENCE FOUNDATION

Project PFR-8001506

Project CEE-8206909

ACKNOWLEDGMENTS

This research is partially sponsored by the National Science Foundation (NSF) under Grants PFR-8001506 and CEE-8206909. The authors thank NSF for this support. University of Pittsburgh, University Computer Center provided computer time and other facilities for this work. The cooperation and assistance of the Computer Center staff is appreciated.

The authors also thank Mike Bussler, President, and Blaine Myers, engineer, of Algor Interactive Systems, Inc., Pittsburgh, for computer time, access to facilities and assistance needed for the solution of an example by SUPERSAP.

Any opinions, findings, conclusions or recommendations expressed in this publication are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

ABSTRACT

DYNAMIC ANALYSIS OF MULTISTORY BUILDINGS

by COMPONENT MODE SYNTHESIS

Modal and transient analyses of a linearly elastic building subjected to ground accelerations are core and time intensive computations. To save computing time and to solve the problem at a lower core requirement, a unique combination of reduction procedures, with fixed-interface component mode synthesis as the central theme augmented by static condensation and Guyan reduction, is formulated and implemented for the given structure and load case.

The method of fixed-interface component synthesis reduces component matrices by transforming them into a linear space spanned by boundary degrees of freedom and a truncated set of normal mode shapes extracted from components with fixed boundaries. Static condensation reduces the matrices entering component eigensolutions. Guyan reduction, a step employed after synthesis, eliminates degrees of freedom on the boundary. The outcomes are substantially reduced system matrices for eigensolution and transient analyses.

A six-story 3-D frame was solved for natural frequencies and mode shapes. The validity of the procedures and program was established by comparing results to that obtained from SUPERSAP, a general purpose finite element program. The agreement is very good. A twelve-story three-dimensional building with an L-shape floor plan was also analyzed. The results indicate

that the combined procedures are advantageous in terms of convergence, the structural characteristics preserved and the percentage of reduction achieved. The results also confirm the importance of floor flexibility in the example studied. Assuming inadequate diaphragm design, other cases in which the floor flexibility can be a significant factor are: buildings with U, T or H-shape floor plan, buildings having setback or local irregularities, buildings supporting heavy masses on floors. The procedures are suitable for the given structure and load case because of the stiffness characteristics of a building and the predominance of lower component and system modes. The penalties partially offsetting the advantages are the needs to solve component eigensolutions and to perform many transformations.

DESCRIPTORS

Building

Computer Program

Floor flexibility

Modal analysis

Seismic analysis

Structure analysis

Transient analysis

Component mode synthesis

Finite element analysis

Guyan reduction

Model reduction

Static condensation

Substructure

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$\{a\}$	a vector indicating scale factors for ground acceleration
B	boundary or region to be retained
$[C]_i$	damping matrix of i-th component
$[C]$	system damping matrix
d_{rn}	damping ratio of the n-th system mode
$\{F\}_i$	interaction force at common boundaries of the i-th component
I	interior or region to be liminated
$[K]_i$	stiffness matrix of the i-th component
$[K^*]$	reduced stiffness matrix after static condensation or Guyan reduction
$[k]_i$	stiffness matrix of the i-th component in p-coordinates
$[\overline{K}]_i$	stiffness matrix of the i-th component in q-coordinates
$[K]$	system stiffness matrix in q-coordinates
$[M]_i$	mass matrix of the i-th component
$[M^*]$	reduced mass matrix after static condensation or Guyan reduction
$[m]_i$	mass matrix of the i-th component in p-coordinates
$[\overline{M}]_i$	mass matrix of the i-th component in q-coordinates
$[M]$	system mass matrix in q-coordinates
$\{p\}_i$	generalized coordinates of the i-th component
$\{p_B\}_i$	boundary coordinates of the i-th component
$\{p_N\}_i$	normal mode coordinates of the i-th component
$\{{}^{u}p_{eff}(t)\}$	effective seismic load vector in u-coordinates
$\{{}^{q}p_{eff}(t)\}$	effective seismic load vector in q-coordinates
$\{q(t)\}$	system displacement relative to the ground

$\{\dot{q}(t)\}$	system velocity relative to the ground
$\{\ddot{q}(t)\}$	system acceleration relative to the ground
$\{q_B\}$	boundary coordinates of the system
$\{q_N\}$	normal mode coordinates of the system
Q_n	amplitude of the n-th decoupled system mode
$[T]$	transformation matrix between two coordinate systems
$\{T_n^*\}$	n-th mode shape of a component
$[T_{IN}^*]$	modal matrix of a component
$\{T_n\}$	n-th mode shape of the system
$\{u(t)\}$	physical displacement relative to the ground
$\{\dot{u}(t)\}$	velocity in u-coordinates
$\{\ddot{u}(t)\}$	acceleration in u-coordinates
$\{u_B\}$	physical DOF to be retained
$\{u_I\}$	physical DOF to be eliminated
ω_n	frequency of the n-th mode of a component or system
ω_{dn}	damped frequency of the n-th mode of the system

1.0 INTRODUCTION

The main theme of the research effort is the application of fixed-interface component mode synthesis, augmented by static condensation and Guyan reduction, in order to evaluate dynamic characteristics and displacement response of a linearly elastic multistory building subjected to ground accelerations.

To date, application of the fixed-interface component mode method to buildings has been limited to a few highly idealized cases. Efforts are made here to formulate and implement the method as applied to seismic analyses of large buildings, and also to test as well as examine its feasibility, advantages and disadvantages. In formulating the modal synthesis, several simplified transformations are derived to upgrade computing efficiency.

The component mode method is a dynamic substructuring technique within the general domain of the finite element approach. By this method, normal mode shapes are extracted from components (substructures) and then used to obtain a reduced master model defined over physical coordinates (boundary coordinates) and generalized coordinates (component normal mode shapes). The master model is then analyzed at lower time and core requirements than that required of an unreduced full scale model. The 'fixed-interface component mode method' is selected for its compatibility at the boundaries and its clarity in implementation.

To take advantage of the high stiffness in a floor plane, static condensation is applied before modal synthesis and Guyan reduction is applied afterward. The Guyan reduction serves to reduce the boundary DOF, which are wholly retained after synthesis. This unique combination of procedures resulted in a substantial reduction in the model size at both component and system levels when applied to solve a twelve-story 3-D building.

The preservation of the dynamic characteristics of components and part of the saving in core and computing time are achieved by the use of a truncated set of component modes. A small truncated set may be used for the given type of problem because of low input energy imparting onto higher modes and low participation by higher modes. Additional savings are achieved by sharing the same allocated core for sequential computations of many components. The penalties partially offsetting the above advantages are the needs to solve component eigenvalue problems and to perform many additional transformations.

2.0 STATE-OF-THE-ART

2.1 Seismic Analysis

The typical configuration of a building is a three-dimensional (3-D) moment-resisting frame, with or without bracing members and shear walls. The bracing members and shear walls serve to enhance the lateral stiffness. Floor diaphragms serve to couple shear walls and frames together, forcing them to respond as a system. During earthquakes, the ground displacement and rocking motions experienced by a building are approximately equivalent to time-varying horizontal and vertical forces consisting of various frequency components. They are random in both form and magnitude. The response of a building depends on the intensities and time history of ground motions and the dynamic properties of the building-foundation-soil system.

Given a set of earthquake load specifications, the goals of seismic analysis are to ensure design adequacy in terms of requirements (such as allowable stresses and story drifts) and to improve reliability and economy within these requirements. Currently, there are three methods by which earthquake loads are specified: (a) equivalent static force, (b) design response spectra, and (c) time history of ground accelerations. A brief discussion is as follws.

The equivalent static force is primarily an approximation to the first mode effect. An example of equivalent static forces is that specified by the Uniform Building Code[(1)],* which consists of the magnitude and distribution of lateral loads over the height of a building. The required 'minimum total lateral seismic force' is based on factors such as the seismic zone coefficient, occupancy importance factor, horizontal force factor (based on building frame type), seismic coefficient (as function of the period of fundamental mode), local geology and soil condition factor, and total dead load. Somewhat different but similar forms of equivalent static forces are specified by the Applied Technology Council[(2)]. Regardless of the details and scale factors, this method provides an approximation to the first mode dynamic loads , with adjustment to partially account for the second mode effect. One drawback is that all higher modes are neglected. Another drawback is that static analysis alone renders little insight into the dynamic characteristics of the system and hence it is less effective in uncovering undesirable aspects of a design.

A design response spectrum consists of a family of curves, where every point represents the absolute value of the maximum (peak) response of a single-degree-of-freedom (SDOF) system to a given time history of ground accelerations. The maximum responses of a set of SDOF systems having the same damping value are plotted on the same curve, where the abscissa is the natural frequency (or period) of the SDOF system; and the ordinate is the maximum response. The response

*Parenthetical references placed superior to the line of text refer to the bibliography.

may be either displacement, velocity, or acceleration; its values as function of time are calculated from numerical integration. It should be noted that the time at which a peak response occurs is not shown on the curve. It should also be noted that it is an implicit way of specifying the loads; i.e. it shows how a set of SDOF systems react to a given time history of ground accelerations without indicating what the history is. To design for such load specification, modal analysis is first performed to calculate the natural frequencies and mode shapes. The responses of individual modes are then calculated from the curves and the SDOF system parameters, which are damping values and natural frequencies (or periods). A popular method to estimate the maximum of a response quantity, for example, the displacement at a nodal point, is to calculate the square root of the sum of the squares (SRSS) of the modal values of that response quantity. No numerical integration is needed. Such estimate of maximum response is often satifactory, but its accuracy may not be good if the system has closely spaced frequencies.

The time history of ground accelerations explicitly describes the amplitude, the frequency contents and duration of random pulses. Although they are not likely to reoccur, the data do allow for accurate simulation of building vibration in response to one possible sequence of events.

To arrive at an economical design that satisfies requirements, the following items may be considered:

1. Adequate lateral stiffness and good load transfer among different regions so that lateral displacements and resulting stresses are below target limits.

2. Appropriate frequency characteristics of the building for the local geological and soil conditions so that the dynamic loads induced are lower.

3. Appropriate building configuration so that dynamic effects and undesirable vibration modes are minimized.

4. Balanced deflection patterns and sufficient ductility so that much energy can be safely absorbed or dissipated during elastic or inelastic deformation.

Modal analysis and response history analysis using finite element models are the best approaches to provide information needed for evaluating a design from the above viewpoints. But they are core and time intensive computations. There is always a need to save cost. In addition, unreduced full scale models may be too big to run on an available facility.

2.2 Model Reduction

One way to stretch hardware capacity so that the same amount of available core can be used to solve a larger problem and to save computing cost is to reduce the size of a full scale model. This can be accomplished by using reduction techniques discussed below.

2.2.1 Substructuring and Static Condensation

The key idea of static condensation in reducing the stiffness matrix is to eliminate 'unwanted' interior degrees of freedom (DOF) by expressing them in terms of a set of DOF to be retained. The operation is equivalent to partially executed Gaussian eliminations. Static condensation can be applied to reduce a global model. It can also be applied to substructures before they are assembled.

Many computer codes adopt the static condensation technique. For example, ANSYS provides a "super-element' feature permitting the user to apply static substructuring to reduce model size. Another example is the TAB program family, i.e. ETABS, TABS and TABS'77, which was specifically developed for analysis of large buildings. For a three-dimensional building, the program automatically performs static condensation floor by floor, retaining only three DOF per floor, namely, two horizontal translation DOF and one rotational DOF about the vertical axis passing through the mass centroid of the floor.

When the method is applied to dynamic problems, the drawbacks are: (a) the local mode shapes involving eliminated DOF are lost, and (b) the lumping of masses to the retained DOF is done by judgement. In the case of TAB program family, the reduction scheme implies that, in all vibration mode shapes extracted from the reduced model, every floor collectively acts like a rigid body having only three out of six possible rigid body DOF. This is a good approximation for the type of buildings in which floor systems are very stiff and floor plans are convex shapes with low aspect ratios. In reality, many buildings do not fall in this category. Incidentally, another problem with the TAB programs is that they cannot accommodate bracing members that run in a vertical plane across several floors, a design feature that is incorporated in many high-rise buildings.

2.2.2 Guyan Reduction

To facilitate reduction in dynamic problems, Guyan [3] extended static condensation. In his formulation, .the same transformation relating the complete set and the reduced set of coordinates was used to reduce the mass matrix so that the kinetic energy is invariant to coordinate transformation. It is a significant improvement over static condensation in that the mass lumping is based on stiffness relationships rather than judgement. But, again, the local mode shapes involving eliminated DOF are lost.

When local mode shapes reflecting floor flexibility are significant, an appropriate way to economically include them in the system model is the method of component mode synthesis.

2.2.3 Component Mode Synthesis

Since Hurty's [4] first proposal in 1960, the method of component mode synthesis (CMS) has been extensively applied in the aerospace industry. The method was initially slow in spreading, but recently there has been rapid proliferation in application to other fields.

Excellent reviews of the subject were provided by Craig[5], Noor[6], Nelson[7] and Meirovitch[8]. Their reports have served as a guide to this short survey.

The procedures of component mode synthesis are as follows:

1. Form stiffness and mass matrices and solve the eigenvalue problem for all substructures.

2. Perform coordinate transformations to reduce all component matrices. The new set of DOF consists of physical coordinates and a truncated set of normal coordinates.

3. Assemble all respective component matrices to obtain system stiffness and mass matrices.

4. Solve the master model for static or dynamic responses. Provide adjustment at the boundaries if necessary.

The key is the use of a truncated set of component normal modes as generalized coordinates. It is really an extension of the Ritz method. Without truncation, the process would simply be extra exercises. Without the use of normal modes, the convergence will most likely be very poor.

Methods of component mode synthesis differ in the way compatibility at the boundaries (components interfaces) is enforced. The first method is Hurty's 'fixed-interface normal mode' method[4]. His method requires that all boundary DOF are retained and that for the purpose of calculating component normal modes the component boundaries are fixed. The consequences are these: (a) Compatibility at the boundary DOF is not impaired. After component matrices are assembled, it is not necessary to adjust the boundaries to account for component interactions. (b) The reduction is carried out in the interior regions only. The total number of boundary DOF remains the same.

The second approach was proposed by Gladwell[9]. A component with a fixed interface is attached to another component which is free at the same interface. The modes of a substructure are calculated with all other connected substructures assumed to be rigid. This approach is called the 'branch component' method. It is suitable for chain-like structures. The third method, proposed by Goldman and Hou [10], is called the 'free-interface normal mode' method. There are hybrid versions of these three methods by MacNeal and Klosterman[11]. Details of these methods can be found in the literature cited; however, the main focus here is the fixed-interface method.

Applications of the methods to different types of structures are summarized as follows:

(a). Idealized structures: cylindrical shell mounted to a flat plate by Cromer[12], two flat plates joined at a right angle by Jezequel[13] and L-shaped bent cantilever beam by Hurty[4].

(b) Aerospace structures: launch vehicle by McAleese[14], Saturn V by Grimes[15], general aerospace structure by Seaholm[16], space shuttle by Fralich[17] and by Zalesak[18], spacecraft by Case[19], spacecraft by Kuhar[20], missile by Gubser[21], and Viking orbiter by Wada[22].

(c) Mechanical structures: automobile components by Klosterman[11, 23, 24], railroad cars by Bronowicki[25], turbine blades by Srinivasan[26] and by Perlman[27], and rotor bearing by Glasgow[28].

(d) Civil engineering structures: piping system by Singh[29], rod group supported by thick circular plate by Lee[30], soil-structure interaction by Gutierrez[31], building and machine foundation by Warburton[32], two-story plane frame by Hurty[4], multistory shear building by Kukreti[33], two-story plane frame by Gladwell[9].

2.3 Remarks and Objectives

After reviewing the works related to model reduction procedures, the following observations are apparent:

1. Applications of the fixed-interface component mode method to buildings have to date been limited to a few highly idealized cases such as 'shear building'' or very small plane frame. A procedure that works well in a two-dimensional case may encounter difficulties when it is extended to a three-dimensional case. Whereas the component mode synthesis method has been implemented in the MSC/NASTRAN, it was not developed specifically for the case of a building for which justifiable treatments can lead to better computing efficiency.

2. No work has been done to employ all three reduction procedures, allowing each one to complement the others, whenever structure reality permits. As will be discussed in the next chapter, some chracteristics of a building can be utilized to achieve reduction in addition to what can be accomplished by the method of component mode synthesis alone.

3. Many computer codes developed for analyzing buildings are based on the assumption that floor systems are rigid in plane during vibration. It is a good approximation when the floor plan is a convex shape with a low aspect ratio. For buildings with other types of floor plans such as L, H, T and U-shapes, or buildings having setbacks or local irregularities, or buildings supporting heavy equipment, failure to account for floor flexibility in the model when

the diaphragm design is inadequate can lead to detrimental errors.

The objectives of this work therefore are:

1. Formulate and implement the method of fixed-interface component mode synthesis as applied to the case of a building subjected to ground accelerations.

2. Investigate the feasibility, advantages and disadvantages of the method by examining its procedures and by making a case study which will also demonstrate that a medium-sized building can be solved by the program using a limited amount of core.

3. Achieve a large percentage of reduction, so that the average retained DOF per floor is larger, but not much greater than three DOF per floor; and that important dynamic properties are preserved in the reduced system. An average retained DOF per floor of value between 12 to 36 will be satisfying.

3.0 FINITE ELEMENT MODEL, REDUCTION AND SOLUTION

3.1 Model Reduction

Before the finite element model of a building is presented, it is useful to discuss some structural realities that lead to a combination of reduction procedures to be used in this work. First, a building behaves laterally like a vertical cantilever beam. The axial (or in-plane stretching) and bending stiffness of a floor are usually higher than the overall lateral stiffness of a building. The lower local modes of a floor may be of some significance, but the higher local modes would most likely be of little importance to the system. Second, within a floor system, the axial stiffness is higher than the flexural bending stiffness. Several joints in a girder would have nearly equal axial displacements along its axis. Thus along the same girder, one may condense out some axial DOF while retaining a selected number of DOF to preserve the most flexible local modes, which are in-plane and out-of-plane flexural bending modes. This concept is illustrated in Figures 1 and 2, where the numbers of retained boundary DOF are 24 and 9, respectively. The total number of translational DOF per boundary is 42. Since there is little kinetic energy associated with rotational DOF, a well accepted fact underlying the use of translational lump masses, all rotational DOF may be condensed out.

The reduction procedures to be employed are as follows:

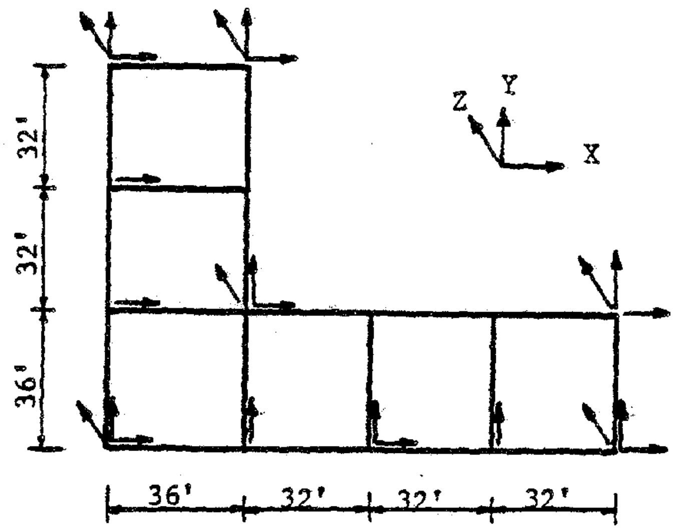


Figure 1 Retained DOF on Boundary Floor-Pattern A

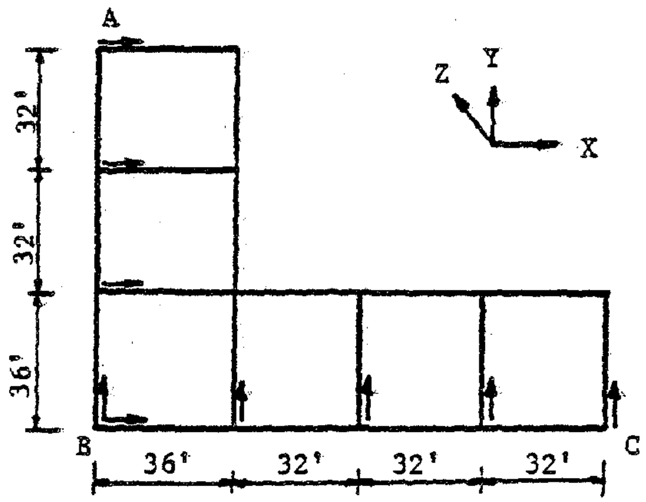


Figure 2 Retained DOF on Boundary Floor-Pattern B

1. For interior nodes in each component, use static condensation to condense out all rotational DOF and some translational DOF that are connected by high stiffness to other retained DOF on the same floor.

2. For each component, which includes several floors, apply the CMS method to reduce the remaining DOF in its interior region. The component normal modes extracted and included are inter-floor local modes.

3. For the system after synthesis, use Guyan reduction to eliminate all rotational DOF and some translational DOF that are connected by high stiffness to other retained DOF on the same floor. This is done at all boundaries.

As stated previously, by the fixed-interface component mode method, only interior DOF are reduced. All boundary DOF must be retained. This works out nicely for small plane frames. For larger building structures, the model size after synthesis is still large. The Guyan reduction used here serves to reduce DOF at the boundaries. The application of both static condensation and Guyan reduction therefore enhances the merit of the component mode method when applied to building structures. The combined procedures are appropriate because of favorable structure realities.

The TAB program family retains only three out of six possible rigid body DOF of a floor. As discussed previously, it is a good approximation when the floor plan is a convex shape with a low aspect ratio. For buildings in which the floor flexibility is a significant factor, failure to account for it can lead to detrimental errors in assessing design adequacy. The reduction procedures employed here provide a good compromise between an unreduced model and oversimplified ones.

During the combined reduction processes, the stiffness and mass matrices are defined over a total of six coordinate systems. They are:

1. Coordinates before static condensation at the component level. With respect to the references, component matrices and vectors are formed.

2. Coordinates after static condensation at the component level. With respect to the references, the reduced component matrices and vectors are defined.

3. Mixed coordinates for components. With respect to the references, further reduced component matrices and vectors are defined. The reduction is the outcome of discarding higher component modes.

4. Mixed cooordinates for the system after synthesis. The component matrices and vectors are transformed and assembled.

5. Mixed coordinates for the system after Guyan reduction. Based on the new references, the reduced system matrices and vector are defined.

6. Normal coordinates of the system after decoupling. System matrices and vector are redefined. A truncated set of system normal modes is then taken.

The combined reduction in model size is substantial, but the resulting increase in programming efforts for transformations and bookkeeping is enormous.

3.2 Equation of Motion

For a component, the unreduced equation of motion subjected to ground acceleration is

$$[{}^{\backprime}M]\{\ddot{u}^{abs}(t)\} + [C]\{\dot{u}(t)\} + [K]\{u(t)\} = \{F\} \qquad (3\text{-}1)$$

where

$[{}^{\backprime}M]$ = component mass matrix

$[\,K]$ = component stiffness matrix

$[\,C]$ = component damping matrix

$\{\ddot{u}^{abs}(t)\}$ = absolute or total accelerations

$\{u(t)\}$ = displacement relative to the ground

$\{F\}$ = interaction forces at the common boundaries

In these terms, a subsript 'i' indicating the component number is implied, although not explicitly printed. These variables are defined over global coordinates (X,Y,Z). A component mass matrix is formed by directly lumping masses to the the boundary DOF and to the interior DOF that are to be retained. A component stiffness matrix is formed by assembling element stiffness matrices in global coordinates. The element stiffness matrices in local and global coordinates are given in APPENDIX A.

The total acceleration may be expressed as

$$\{\ddot{u}^{abs}(t)\} = \{\ddot{u}(t)\} + \{a\}\ddot{u}_g(t) \qquad (3\text{-}2)$$

in which the scalar time series $\ddot{u}_g(t)$ are ground acclerations, and $\{a\}$ is a vector indicating the scale factors. It is constructed as follows: assign value '0' to all rotational DOF and assign values a_x, a_y, and a_z to translational DOF parallel to global axes X, Y and Z, respectively. The horizontal direction of the earthquake is indicated by the vector (a_x, a_y). Eq.(3-1) can now be rewritten as

$$
\begin{aligned}
&[\bar{M}]\{\ddot{u}(t)\} + [C]\{\dot{u}(t)\} + [K]\{u(t)\} \qquad (3\text{-}3)\\
&= \{{}^{1}p_{eff}(t)\} + \{F\}\\
&= [\bar{M}]\{a\} \quad (-\ddot{u}_g(t)) + \{F\}\\
&= \{{}^{1}p_{eff}\}^{0} \ (-\ddot{u}_g(t)) + \{F\}
\end{aligned}
$$

where the superscript to the left of a variable indicates the coordinate system. The seismic load vector is based on an unreduced diagonal mass matrix. The scalar time function is factored out for convenience in programming.

The initial finite element models of the components are subsequently reduced through static condensation and component mode synthesis at the component level, and through Guyan reduction and modal decoupling at the system level before solution for responses. Each of these operations results in a new set of stiffness and mass matrices as well as load vector. After synthesis, the system equation of motion remains the same in form as that of a component shown above; except that at all boundaries the respective sum of component interactions vanishes. They are internal forces of the system, and they must cancel (or be in equilibrium) themselves at every common boundary.

The damping matrix $[C]$ is never formed. Instead, damping ratios are assigned to the uncoupled modes of the synthesized system. This is a matter of choice, because these two methods of assigning damping are directly related.

3.3 Static Condensation and Guyan Reduction

Let the static force-displacement relation be

$$[K]\{u\} = \{F\}$$

or

$$\begin{bmatrix} K_{BB} & K_{BI} \\ K_{IB} & K_{II} \end{bmatrix} \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = \begin{Bmatrix} F_B \\ 0 \end{Bmatrix} \tag{3-4}$$

where

$[K]$ = stiffness matrix

$\{u\}$ = displacement vector

$\{F\}$ = load vector

The subscript 'B' indicates boundary or DOF that are to be retained, while the subcript 'I' denotes interior or DOF that are to be eliminated. There are no seismic loads or inertial forces at the unwanted interior DOF, because no masses are assigned to them. After static condensation, the new static force-displacement relation becomes

$$[K^*_{BB}]\{u_B\} = \{F_B\}$$

where

$$[K^*_{BB}] = ([K_{BB}] - [K_{BI}][K_{II}]^{-1}[K_{IB}]) \tag{3-5}$$

and the solution is,

$$\{u_B\} = [K^*_{BB}]^{-1}\{F_B\} \tag{3-6}$$

Equations (3-6) and (3-5) can be derived by rewriting Eq. (3-4) into two equations, solving the second to get

$$\{u_I\} = -[K_{II}]^{-1}[K_{IB}]\{u_B\} \tag{3-7}$$

and then substituting $\{u_I\}$ back to the first equation.

The static condensation can be readily applied to a dynamic problem when the mass matrix has the form

$$[M] = \begin{bmatrix} M_{BB} & 0 \\ 0 & 0 \end{bmatrix}$$

If the mass matrix is sparse, namely,

$$[M] = \begin{bmatrix} M_{BB} & M_{BI} \\ M_{IB} & M_{II} \end{bmatrix}$$

then a more general procedure known as Guyan reduction is needed. By Guyan reduction [3], the reduced stiffness matrix is calculated in exactly the same way as that indicated by Eq.(3-5). The reduced mass matrix is,

$$\begin{aligned} [M^*_{BB}] = {} & [M_{BB}] - [M_{BI}][K_{II}]^{-1}[K_{IB}] \\ & - ([K_{II}]^{-1}[K_{IB}])' \, ([M_{IB}] - [M_{II}][K_{II}]^{-1}[K_{IB}]) \end{aligned} \tag{3-8}$$

The reduction process to obtain $[K^*_{BB}]$ is equivalent to the transformation

$$[K^*_{BB}] = [T]'[K][T]$$

in which $[T]$ is such that

$$\{u\} = \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = [T]\{u_B\} \tag{3-9}$$

where

$$[T] = \begin{Bmatrix} I \\ T^* \end{Bmatrix}$$

and

$$[T^*] = -[K_{II}]^{-1}[K_{IB}] \tag{3-10}$$

Likewise, the reduction from $[M]$ to $[M^*_{BB}]$ is equivalent to the transformation

$$[M^*_{BB}] = [T]'[M][T]$$

Both Eq.(3-5) and Eq.(3-8) can be deduced from the potential energy

$$V = (0.5)\{u\}'[K]\{u\} = (0.5)\{u_B\}'[K^*]\{u_B\}$$

and the kinetic energy

$$KE = (0.5)\ \{\dot{u}\}'[M]\{\dot{u}\} = (0.5)\{\dot{u}_B\}'[M^*]\{\dot{u}_B\}$$

respectively, the latter expression was proposed by Guyan.

3.4 Fixed-Interface Component Mode Synthesis

In order to focus attention to the required operations on stiffness and mass matrices, the free vibration case is discussed first, which is then followed by the forced vibration case.

3.4.1 Undamped Free Vibration

Let $\{u\}$ be the nodal displacement vector. After the component stiffness and mass matrices are formed and condensed statically, the component equation of motion under undamped free vibration is

$$[K^*]\{u\} + [{}^{\backprime}M^*]\{\ddot{u}\} = \{0\} \tag{3-11}$$

in which

$$\{u\} = \begin{Bmatrix} u_B \\ u_I \end{Bmatrix}$$

$$[K^*] = \begin{bmatrix} K_{BB} & K_{BI} \\ K_{IB} & K_{II} \end{bmatrix}$$

and

$$[{}^{\backprime}M^*] = \begin{bmatrix} M_{BB} & 0 \\ 0 & M_{II} \end{bmatrix}$$

where the subscripts 'B' and 'I' denote boundary and interior DOF, respectively. The diagonal mass matrix remains the same after static condensation. As stated previously, damping values will be assigned to individual modes of the synthesized system. Hence, without loss of generality, the damping matrix is dropped in this section.

The eigenvalue problem for a component with fixed-boundaries is now solved to obtain eigenvalues w_n and eigenvectors $\{T^*{}_n\}$, where $n=1,2\ldots N_I$, and N_I is the number of interior DOF of the component. The modal matrix is $[T^*{}_{NN}]$, its j-th column being the j-th eigenvector. Henceforth, $[T^*{}_{NN}]$ will be written as $[T^*{}_{IN}]$, where I denotes interior DOF in u-coordinates and N denotes normal (natural) mode coordinates. Its size is $N_I x N_I$.

The coordinate transformation, as Hurty proposed, is

$$\begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = [T^*] \begin{Bmatrix} p_B \\ p_N \end{Bmatrix} \tag{3-12}$$

where

$$[T^*] = \begin{bmatrix} T^*{}_{BB} & 0 \\ T^*{}_{IB} & T^*{}_{IN} \end{bmatrix} \tag{3-13}$$

The reason for such a transformation is apparent from the developments to follow. The submatrices derived by Hurty are:

1. The submatrix $[T^*{}_{BB}]$ relates $\{u_B\}$ to $\{p_B\}$ to maintain compatibility at the boundary.

2. The submatrix $[T^*{}_{IN}]$ is the modal matrix of the component with fixed-boundary.

3. The submatrix $[T^*{}_{IB}]$ is defined by

$$[T^*_{IB}] = -[K_{II}]^{-1}[K_{IB}] \tag{3-14}$$

The derivation will be shown later.

4. The null submatrix is a consequence of the 'fixed-interface', namely, the amplitudes $\{p_N\}$ contribute nothing to $\{u_B\}$.

In this work,

$$[T^*_{BB}] = [`I] \tag{3-15}$$

is selected to simplify further development. This requires an one to one coordinate transformation between $\{u_B\}$ and $\{p_B\}$. There are a total of N_I mutually orthogonal component normal modes. Less than N_I modes will be taken, so both $[T^*]$ and $[T^*_{IN}]$ become rectangular matrices. Note that if no component normal mode is retained, then $[T^*] = (T^*_{BB}{}'\ T^*_{IB}{}')'$, and hence $\{u_I\} = [T^*_{IB}]\{p_B\}$, which is the same transformation for static condensation.

To see that Eq.(3-14) is true, consider Eqs.(3-12) and (3-13) and a dynamic equilibrium relation

$$\begin{bmatrix} K_{BB} & K_{BI} \\ K_{IB} & K_{II} \end{bmatrix} \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = \begin{Bmatrix} F_B \\ F_I \end{Bmatrix}$$

where the force vector on the right hand side includes all dynamic forces. Now let the normal mode displacement $\{p_N\} = \{0\}$. Correspondingly, both of the load vectors $\{P_N\}$ and $\{F_I\}$ vanish. Therefore,

$$\{u_I\} = [T^*_{IB}]\ \{p_B\} = [T^*_{IB}]\ \{u_B\}$$

and

$$[K_{IB}]\{u_B\} + [K_{II}]\{u_I\} = \{0\}$$

Comparing the two expressions, we get Eq.(3-14).

After transformation, the component equation of free vibration becomes

$$[k]\{p\} + [m]\{\ddot{p}\} = \{0\} \tag{3-16}$$

where

$$[k] = [T^*]'[K^*][T^*] \tag{3-17}$$

and

$$[m] = [T^*]'[M^*][T^*] \tag{3-18}$$

The procedures to perform the transformations efficiently are discussed in a later section.

Next the system generalized coordinates $\{q\}$ are defined such that

$$\begin{Bmatrix} p_B \\ p_N \end{Bmatrix} = \begin{bmatrix} T_B & 0 \\ 0 & T_N \end{bmatrix} \begin{Bmatrix} q_B \\ q_N \end{Bmatrix} \tag{3-19}$$

where N denotes 'component normal modes'. Compatibility at the boudary is maintained through transformation from $\{p_B\}$ to $\{q_B\}$ via $[T_B]$. The matrix $[T_N]$ is a Boolean matrix relating each DOF in $\{p_N\}$ to an appropriate location in $\{q_N\}$.

Now let the whole transformation matrix above be $[T]$. Upon completion of transformation, the component equation of free vibration is

$$[\bar{K}]\{q\} + [\bar{M}]\{\ddot{q}\} = \{0\} \tag{3-20}$$

where

$$[\overline{K}] = [T]'[k][T]$$

and

$$[\overline{M}] = [T]'[m][T]$$

In the above expressions, a subscript 'i' indicating the component number is implied. The same procedures can be applied to all components. The stiffness and mass matrices for the region not included in any component can be formed in $\{q\}$ coordinates, or in other coordinates and then transformed. The next step is to assemble the component matrices to obtain system matrices.

Indeed, the fixed interfaces allow for relatively straightforward implementation, Once component matrices are transformed to q-coordinates, they may be assembled to form system matrices by the same procedures as that used in static condensation.

Up to this point the boundaries have never been reduced. If further reduction of model size is needed, Guyan reduction may be applied, because the mass matrix is now sparse. When it is completed, the system equation of motion for free vibration is,

$$[K]\{q\} + [M]\{\ddot{q}\} = \{0\} \tag{3-21}$$

3.4.2 Forced Vibration

For the case of forced vibration under ground accelerations, the 'appropriate forms' of the seismic load vector as described in section (3.2) should be used to replace the null load vectors in the free vibration equations. The procedures to obtain the 'appropriate forms' of the seismic load vector are as follows:

The unreduced seismic load vector of a component is the first term on the right hand side of Eq.(3.3). Each one of the subsequent reduction processes is equivalent to a specific coordinate transformation. Consequently, the loading should be transformed according to the following general equation

$$[T^{12}]' \{{}^{1}p_{eff}(t)\} = \{{}^{2}p_{eff}(t)\} \tag{3-22}$$

where $[T^{12}]'$ denotes the transpose of the transformation matrix from coordinate system 1 to 2, i.e., $[T^{12}]$ is such that $\{{}^{1}x\} = [T^{12}]\{{}^{2}x\}$. Thus, the sequences of computations are:

1. For static condensation at the component level, simply delete the zero terms associated with the unwanted interior DOF. No computation is necessary, because no mass is allocated to any unwanted interior DOF and hence no inertial force is generated there.

2. Parallel to the operations on each component, apply Eq.(3-22) and the applicable rotation matrix in the form described in Eq.(3-13) to transform the component load vector.

3. Assemble the component load vectors to form the system load vector. This step is equivalent to the transformation defined by Eq.(3-19).

4. Corresponding to Guyan reduction at the system level, apply

Eq.(3-22) and the rotation matrix given in Eq.(3-10) to reduce the system load vector.

3.5 Efficient Matrix Operations

By taking advantage of the choice of $[T^*_{BB}] = [\,^\backslash I]$, lump masses, the zero submatrix, and the orthonormal property of $[T^*_{IN}]$, expressions can be derived to efficiently carry out the transformations given in Eqs. (3-17) and (3-18).

Let the outcome of the matrix operations defined by Eq.(3-17) be

$$[k] = \begin{bmatrix} k_{BB} & k_{BN} \\ k_{NB} & k_{NN} \end{bmatrix} \tag{3-23}$$

Using the expressions given in Eqs.(3-13),(3-15) and (3-14) to evaluate Eq.(3-17), we get

$[k_{NN}] = [\,^\backslash w_n^2]$,

if component mode shapes are normalized, and

$[K_{BN}] = [K_{NB}] = [0]$,

as result of cancellations, and

$$[k_{BB}] = [K_{BB}^*] - [K_{BI}^*]\,[K_{II}^*]^{-1}\,[K_{IB}^*] \tag{3-24}$$

where the operation required to get $[k_{BB}]$ is precisely the same as that required in Guyan reduction and static condensation as shown in Eq.(3-5).

Likewise, let the outcome of Eq.(3-18) be

$$[m] = \begin{bmatrix} m_{BB} & m_{BN} \\ m_{NB} & m_{NN} \end{bmatrix} \tag{3-25}$$

Using the expressions given in Eqs. (3-13),(3-15) and (3-14), knowing that the component mass matrix remains diagonal after static condensation as a consequence of our method of assigning the unreduced component mass matrix, we can evaluate Eq.(3-18) to obtain

$$[m_{BB}] = [\grave{}m_{BB}{}^*] + [T_{IB}{}^*]' \, [\grave{}m_{II}{}^*] \, [T_{IB}{}^*]$$

$$[m_{BN}] = [T_{IB}{}^*]' \, [\grave{}m_{II}{}^*] \, [T_{IN}{}^*]$$

$$[m_{NB}] = [m_{BN}]'$$

and

$$[m_{NN}] = [\grave{}I] \tag{3-26}$$

To reiterate, these equations are based on fixed interfaces, a diagonal mass matrix entering CMS, $[T_{BB}{}^*] = [\grave{}I]$ and normalized component eigenvectors. They are substantially more simplified than the submatrices that can be derived otherwise. It should also be noted that $[m_{BB}]$ is essentially the same as the Guyan mass matrix defined in Eq.(3-8), except that the required operation here is much simpler because of the diagonal mass matrix entering CMS.

As stated previously, the operation defined by Eq.(3-5) is equivalent to the partially executed Gaussian elimination. We can see this by considering the following

$$\begin{bmatrix} K_{BB} & K_{BI} \\ K_{IB} & K_{II} \end{bmatrix} \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = \begin{Bmatrix} F_B \\ 0 \end{Bmatrix} \tag{3-27}$$

After partial triangularization, we have

$$\begin{bmatrix} \underline{K}_{BB} & 0 \\ \underline{K}_{IB} & \underline{K}_{II} \end{bmatrix} \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = \begin{Bmatrix} F_B \\ 0 \end{Bmatrix} \tag{3-28}$$

Rewriting the first equation, we get

$$[\underline{K}_{BB}] \{u_B\} = \{F_B\} \tag{3-29}$$

Comparing the expression to Eq.(3-5), we see that indeed the matrix $[\underline{K}_{BB}]$ derived from partial triangularization is the stiffness matrix desired. The matrix inversion and multiplications are therefore bypassed.

Finally, a novel process can be used to calculate the ubiquitous transformation matrix given in Eq.(3-10). Suppose we further reduce Eq.(3-28) to the following form,

$$\begin{bmatrix} \underline{K}_{BB}^* & 0 \\ \underline{K}_{IB}^* & I^* \end{bmatrix} \begin{Bmatrix} u_B \\ u_I \end{Bmatrix} = \begin{Bmatrix} F_B \\ 0 \end{Bmatrix} \tag{3-30}$$

Rewriting the second equation, we get

$$\{u_I\} = -[\underline{K}_{IB}^*] \{u_B\}$$

Comparing this expression to Eq.(3-7), it is evident that

$$[T^*] = -[\underline{K}_{IB}^*] = -[K_{II}]^{-1} [K_{IB}]$$

3.6 System Response to Ground Accelerations

The response of a linear system to time varying ground accelerations can be determined by decoupling the equation into a truncated set of SDOF systems, solving for individual modal responses, and then adding them up. It is a standard procedure. Because of truncation, this approach is much more economical than the direct integration method, by which the dynamic equilibrium relating several whole matrices must be satisfied at all integration steps. Such a requirement is compounded by the need to use very small time increments in order to maintain accuracy and to minimize numerical damping.

3.6.1 Decoupling of System Equation of Motion

After the system matrices are formed and condensed, the system equation of motion in $\{q\}$ coordinates is,

$$[M]\{\ddot{q}(t)\} + [C]\{\dot{q}(t)\} + [K]\{q(t)\} = \{^{q}p_{eff}(t)\} \qquad (3\text{-}31)$$

$$= \{^{q}p_{eff}\}^{0}\,(-\ddot{u}_{g}(t))$$

where

$[M]$ = mass matrix

$[C]$ = damping matrix

$[K]$ = stiffness matrix

$\{^{q}p_{eff}(t)\}$ = effective seismic load

$\{\ddot{u}_{g}(t)\}$ = time series of ground accelerations

The first step is to solve the eigenvalue problem

$$[M]\{q^{\cdot}(t)\} + [K]\{q(t)\} = \{0\}$$

or

$$-w^2[M]\{q(t)\} + [K]\{q(t)\} = \{0\} \tag{3-32}$$

to obtain the natural frequencies w_n and the corresponding eigenvectors or mode shapes $\{T_n\}$, $n=1,2...N_q$. The modal matrix is $[T]$, each of its columns is an eigenvector. The solution is based on an undamped system, because the effect of damping on natural frequencies is nil.

Due to symmetry in $[K]$ and $[M]$ or Betti's Law, the mode shapes obtained from Eq.(3-32) satisfy orthogonal conditions as follows,

$$\{T_m\}'[M]\{T_n\} = 0$$
$$\{T_m\}'[K]\{T_n\} = 0 \tag{3-33}$$

for m not equal to n, and

$$M_n = \{T_n\}'[M]\{T_n\}$$
$$K_n = \{T_n\}'[K]\{T_n\} = w_n{}^2M_n \tag{3-34}$$

for m=n. The damping matrix is assumed to satisfy the orthogonality conditions

$$\{T_m\}'[C]\{T_n\} = 0 \tag{3-35}$$

for m not equal to n, and

$$C_n = \{T_n\}'[C]\{T_n\} = 2d_{rn}w_nM_n \tag{3-36}$$

for m=n. At this juncture, a damping value is assigned to each individual mode in the form of the damping ratio d_{rn}.

The orthogonality conditions permit decoupling as shown below. Premultiplying Eq.(3-31) by $[T]'$, applying a transformation

$$\{q(t)\} = [T]\{Q_n(t)\} \qquad (3\text{-}37)$$

and using the orthogonality conditions, the system equation of motion is reduced to N_q-decoupled SDOF equations of motion in the form

$$M_n \ddot{Q}_n(t) + C_n \dot{Q}_n(t) + K_n Q_n(t) = P_n(t)$$

or

$$\ddot{Q}_n(t) + 2d_{rn} w_n \dot{Q}_n(t) + w_n^2 Q_n(t) = P_n(t)/M_n \qquad (3\text{-}38)$$

where the damping ratio of the n-th mode $d_{rn} = C_n/2M_n w_n$. The loading imparting onto the n-th mode is,

$$P_n(t) = \{T_n\}'\{{}^q p_{eff}(t)\} = \{T_n\}'\{{}^q p_{eff}\}^0(-\ddot{u}_g(t)) \qquad (3\text{-}39)$$
$$= E_n\ (-\ddot{u}_g(t))$$

where E_n is the 'modal earthquake excitation factor,' a term used by Clough and Penzien. It is directly proportional to the scalar product of the n-th mode shape and the spatial distribution of the seismic load vector. It partially accounts for the predominance of lower modes for the given type of problem. The 'modal participation factor,' a term used by Biggs, is equal to E_n/M_n. The two factors are equivalent when mode shapes are orthonormalized.

In Eq.(3-37), the whole modal matrix is used to maintain generality. In application, a truncated transformation matrix may be used to obtain the lowest N' modes that are significant.

3.6.2 Solution of Modal and System Responses

Assuming a system initially at rest, the solution of Eq.(3-38) is

$$Q_n(t) = (1/w_{dn})$$
$$(1/M_n)\left[\int_0^t P_n(x)\, e^{-d_{rn} w_n (t-x)} \sin w_{dn}(t-x)dx\right]$$

or

$$Q_n(t) = (1/w_{dn})\,(E_n/M_n)$$
$$\left[\int_0^t (-\ddot{u}_g(x))\, e^{-d_{rn} w_n (t-x)} \sin w_{dn}(t-x)dx\right] \qquad (3\text{-}40)$$

where the damped natural frequency $w_{dn} = w_n(1-d_r^{\,2})^{1/2}$. Using the step-wise explicit integration method discussed in APPENDIX C to evaluate Eq.(3-40), the N' modal responses are calculated and then added according to Eq.(3-37) to obtain the time history of system response $\{q(t)\}$. Those modes that are higher than N' can be neglected.

4.0 EXAMPLE

The validity of the procedures and program was established by solving a six-story 3-D frame discussed in APPENDIX D. This chapter presents an example that was done to demonstrate that a fairly large 3-D building can be analyzed by the procedures using a limited amount of core. The results are quite interesting; they substantiate the importance of the floor flexibility mentioned in chapter 3, among other things.

Figure 3 shows a perspective view of the twelve story 3-D building. Figure 4 shows a typical floor framing plan. The floor plan and lump mass distribution are applicable to all floors. The dead weight is 940 kips per floor, which is equivalent to 133.2 psf. X-bracing members are used in vertical planes 4-6, 7-8 and 13-14. The bracing layout is similar to that of a floor plan. Table 1 shows the size of structure members. In the table, I_c and I_{sp} are component number and section property number, respectively. Although the design features and dimensions are assumed, they are realistic.

The building was represented by four components numbered sequentially upward. There were three common boundaries and an optional roof boundary. The latter was included to enhance accuracy, the former were needed to maintain compatibility.

For both components No.2 and 3, the DOF number was reduced from 336 (or 4x14x6) to 252 (or 2x84+2x42) when all rotational DOF of

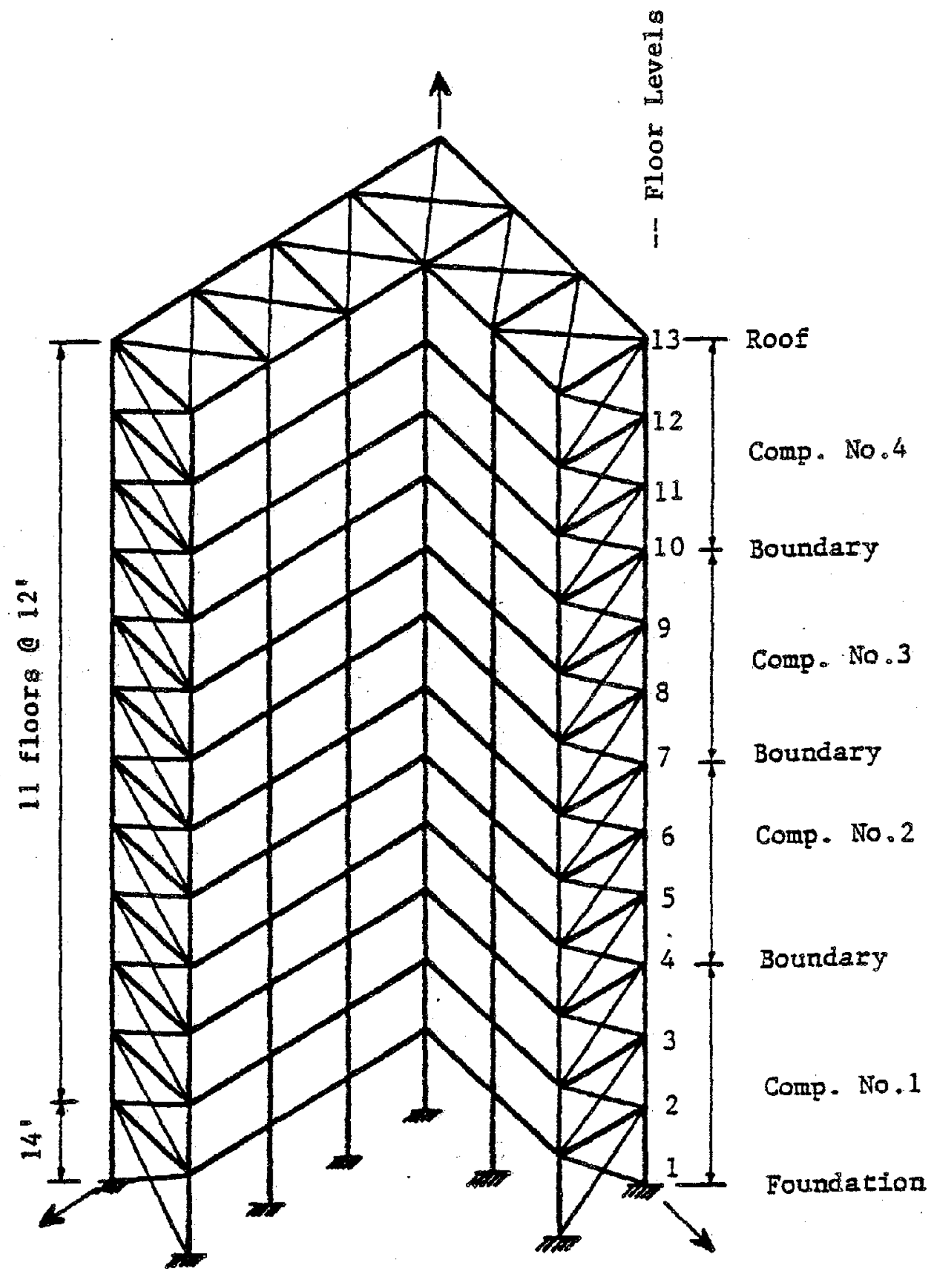
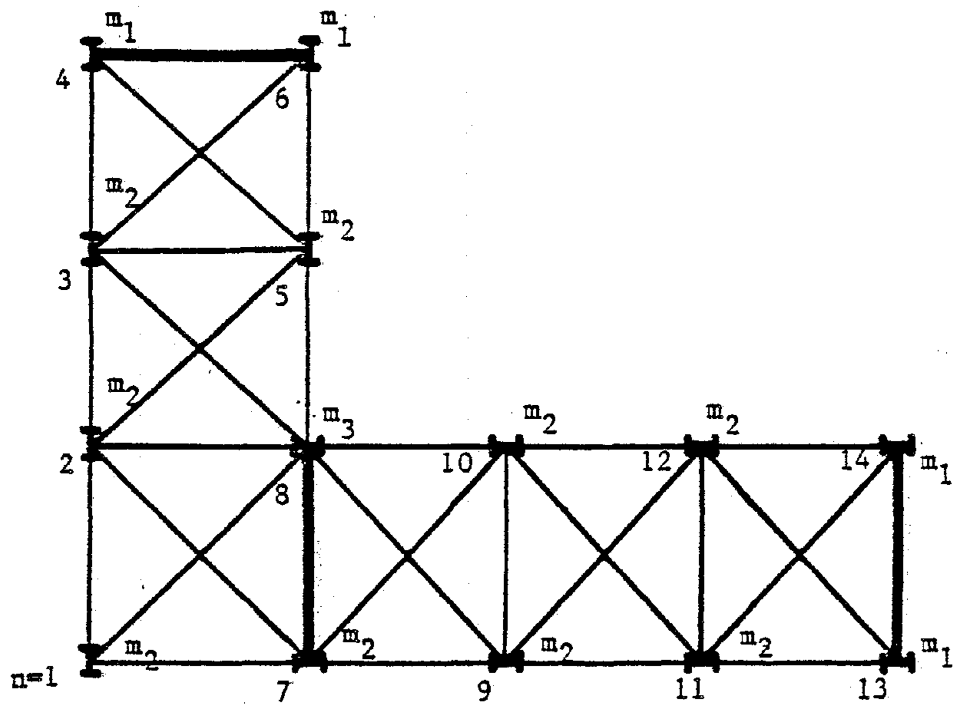


Figure 3 A Perspective View of the Building



Note: 1. The signs I & H indicate positioning of columns.

2. n = node number

3. Lamp masses :

m_1 = 0.101367 kip-Sec^2/ in

m_2 = 0.202733

m_3 = 0.304097

4. Heavy line indicates a vertical plane frame with diagonal bracing.

Figure 4 Typical Floor and Roof Framing Plan

Table 1. Section Properties

I_c	I_{sp}	A	I_y	I_z	J	Remarks
1	1/2	117.	2170./6000.	6000./2170.	272.	column I/H
	3	24.8	2850.	106.	2.79	floor beam
	4	5.5	0	0.	0.	floor bracing
	5	7.5	0	0.	0.	frame bracing
2	6/7	68.5	1150./3010.	3010./1150.	62.6	column I/H
	8	24.7	2370.	94.4	3.72	floor beam
	9	4.865	0.	0.	0.	floor bracing
	10	6.6	0.	0.	0.	frame bracing
3	11/12	46.7	748./1900.	1900./748.	19.5	column I/H
	13	22.4	2100.	82.5	2.70	floor beam
	14	4.22	0.	0.	0.	floor bracing
	15	5.7	0.	0.	0.	frame bracing
4	16/17	42.7	677./1710.	1710./677.	14.2	column I/H
	18	20.	1830.	70.4	1.86	floor beam
	19	3.555	0.	0.		floor bracing
	20	4.805	0.	0.		frame bracing

interior nodes were condensed out. (The program permits further reduction of some translational DOF in the interior.) For the other two components, the DOF number was similarily reduced. On the roof boundary, all rotational DOF were treated as unwanted interior DOF and were condensed out at the component level.

There were two interior floors in each of the four components. The dimension of all matrices entering the component eigensolution was 84 or 2X14X3. The number of normal modes in a component was therefore 84.

To simplify presentation, the following modeling parameters are defined : (a) N_B = the number of retained DOF per boundary , (b) N_R = the number of retained DOF on the roof, (c) N' = the number of retained modes per component, and (d) N_C = the number of components taken. By changing these parameters, the following cases were solved.

(a) N_B=24, N_R=42, N'=12, and N_C=4. After assembling, the system model had 342 DOF : 4x12 component modes, 42 DOF on the roof and 3x14x6 DOF on the common boundaries. The Guyan reduction cut the model size down to 162 DOF, which included 24 retained DOF per common boundary as shown in Figure 1. This is an 84 % reduction from the unreduced model having a total of 1008 DOF. The operations required a main array of 57K plus nominal common areas.

(b) N_B=24, N_R=42, N_C=4, and N'=1,4,6,24,36. Table 2 shows the caculated natural frequencies of the first 13 modes. Evidently, when all the other conditions remain the same, the calculated natural

Table 2. Calculated System Natural Frequencies, CPS

Mode number	N_R=42, N_B=24, N_C=4					
	N'=1	N'=4	N'=6	N'=12	N'=24	N'=36
1	0.3845	0.3841	0.3841	0.3841	0.3841	0.3841
2	0.5126	0.5111	0.5111	0.5111	0.5111	0.5111
3	0.7660	0.7624	0.7623	0.7621	0.7619	0.7619
4	1.1219	1.1117	1.1117	1.1115	1.1115	1.1115
5	1.4789	1.4582	1.4576	1.4573	1.4572	1.4572
6	2.0486	1.9809	1.9809	1.9765	1.9762	1.9762
7	2.1434	2.0975	2.0959	2.0930	2.0914	2.0913
8	2.2070	2.1568	2.1524	2.1452	2.1407	2.1407
9	2.5917	2.5513	2.5446	2.5413	2.5400	2.5400
10	2.9629	2.9116	2.9115	2.8801	2.8787	2.8787
11	3.3481	3.2149	3.2079	3.1894	3.1803	3.1801
12	3.4132	3.3027	3.2837	3.2755	3.2729	3.2726
13	3.6196	3.5504	3.5012	3.4907	3.4872	3.4872

frequencies become lower and lower to approach the 'true values' as more and more component modes are included. These trends are in agreement with the known fact that the calculated natural frequencies are upper bounds. It is also evident that it is not necessary to include many component modes in this case. This is not surprising. We know that a cantilever beam modeled by four elements with consistent mass can produce good results. We may similarly expect Guyan reduction to yield good results if there are four components or four boundaries, and if there are enough retained DOF per boundary. When no component mode is taken, the method of CMS is equivalent to Guyan reduction. Therefore it can only do better when some component modes are included.

(c) N_B=9, N_R=9, N_C=4, N'=1,4,6. Figure 2 shows the retention pattern. Table 3 shows the results. This case proves that it is not necessary to retain many DOF per boundary, a viewpoint suggested in chapter 3. Note that if N'=6, the size of the reduced system is 42, which is equivalent to 3.5 DOF per floor.

(d) N_B=9, N_R=9, N'=12,21, and N_C=2. In the previous cases, the use of four boundaries helped. Could the procedures do well if there are only two boundaries? This case demonstrates that they can. It is remarkable that, after so much number crunching, the results are so close to that of the previous cases in which the sequences and domains of formations and reductions were quite different. We can attribute the success to the capability of the method of component mode synthesis to preserve structure properties effectively. From this case, it is

Table 3. Calculated System Natural Frequencies, CPS

Mode number	$N_R=9$, $N_B=9$				
	$N_C=4$			$N_C=2$	
	N'=1	N'=4	N'=6	N'=12	N'=21
1	0.3854	0.3850	0.3850	0.3854	0.3854
2	0.5134	0.5119	0.5119	0.5121	0.5121
3	0.7664	0.7627	0.7626	0.7626	0.7624
4	1.1393	1.1289	1.1289	1.1374	1.1371
5	1.4952	1.4731	1.4725	1.4786	1.4782
6	2.1230	2.0539	2.0537	2.1005	2.0988
7	2.1483	2.1025	2.1010	2.1182	2.1175
8	2.2573	2.1993	2.1937	2.1796	2.1767
9	2.6466	2.6088	2.6008	2.6398	2.6388
10	3.1806	3.1083	3.1083	2.7813	2.7759
11	3.4051	3.2829	3.2721	3.2667	3.2571
12	3.4348	3.3034	3.2899	3.2796	3.2720
13	3.6972	3.6165	3.5573	3.4204	3.4018

judged that the model representing a twenty-story building with the same floor plan can be reduced to a system model consisting of (6 to 12)X4+9X4 = 60 to 84 DOF or 3 to 4.2 DOF per floor, and yields comparable answers.

Finally, the mode shapes obtained deserve some attention. Figures 5 to 8 show the shapes of two adjacent roof edges in the the first 11 modes for the case with N_R=42, N_B=24, N_C=4, and N'=12. In the first three modes, the roof behaves almost like a rigid body, but starting from the fourth mode, it deflects in the forms of in-plane bending in a significant or predominant way. The message is clear: for the assumed case, the importance of the floor flexibility cannot be overemphasized.

In summary, the solution made of a twelve-story 3-D building using the procedures produced high percentage reductions and yet preserved the most important characteristics of the building.

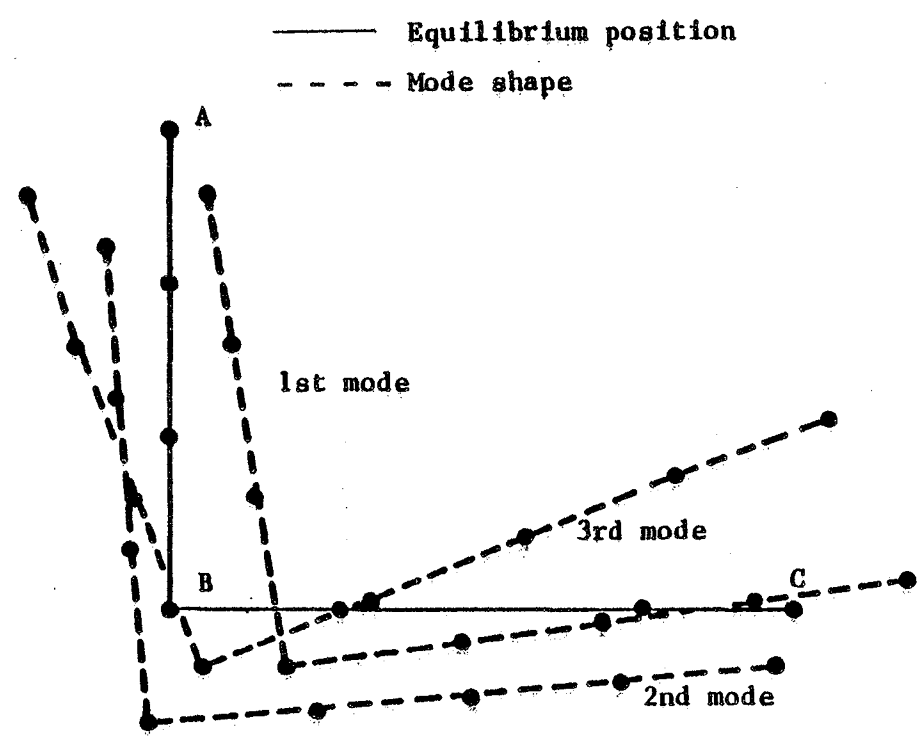


Figure 5 Plan View of Roof Edge Vibration Mode Shapes

— 1st to 3rd modes

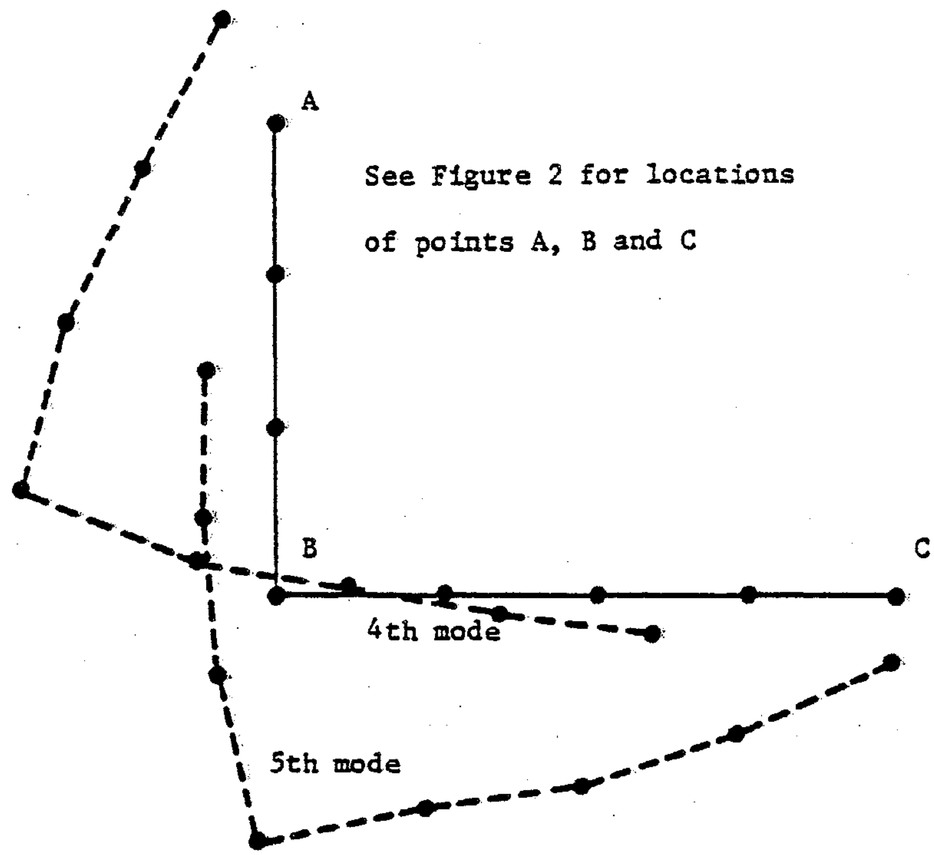


Figure 6 Plan View of Roof Edge Vibration Mode Shapes
— 4th and 5th modes

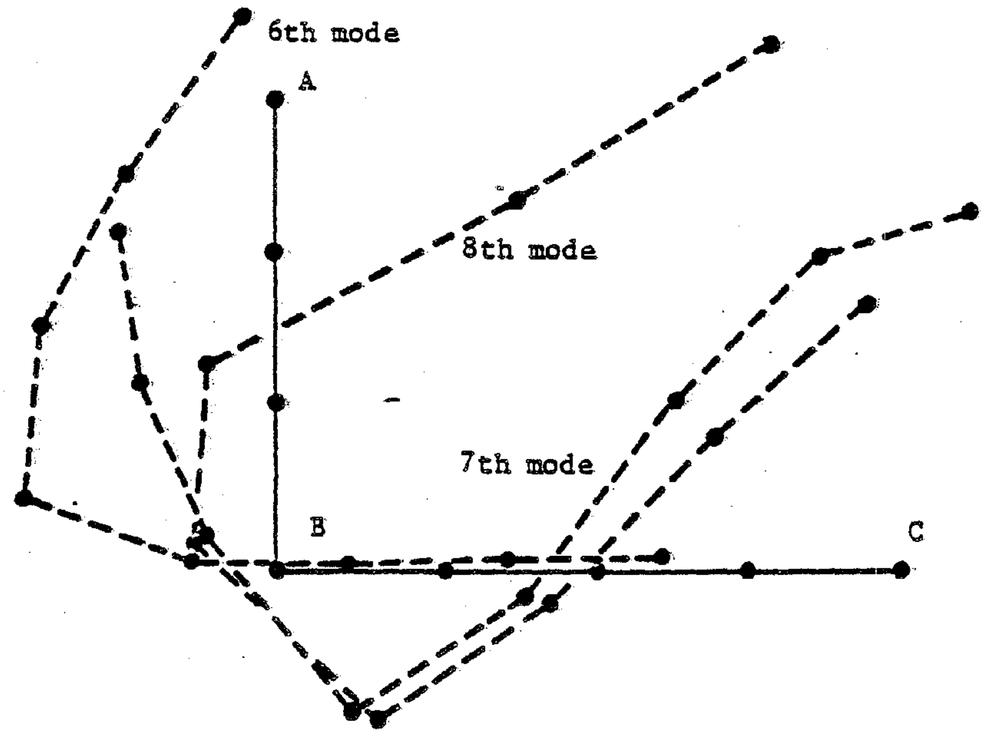


Figure 7 Plan View of Roof Edge Vibration Mode Shapes

— 6th to 8th modes

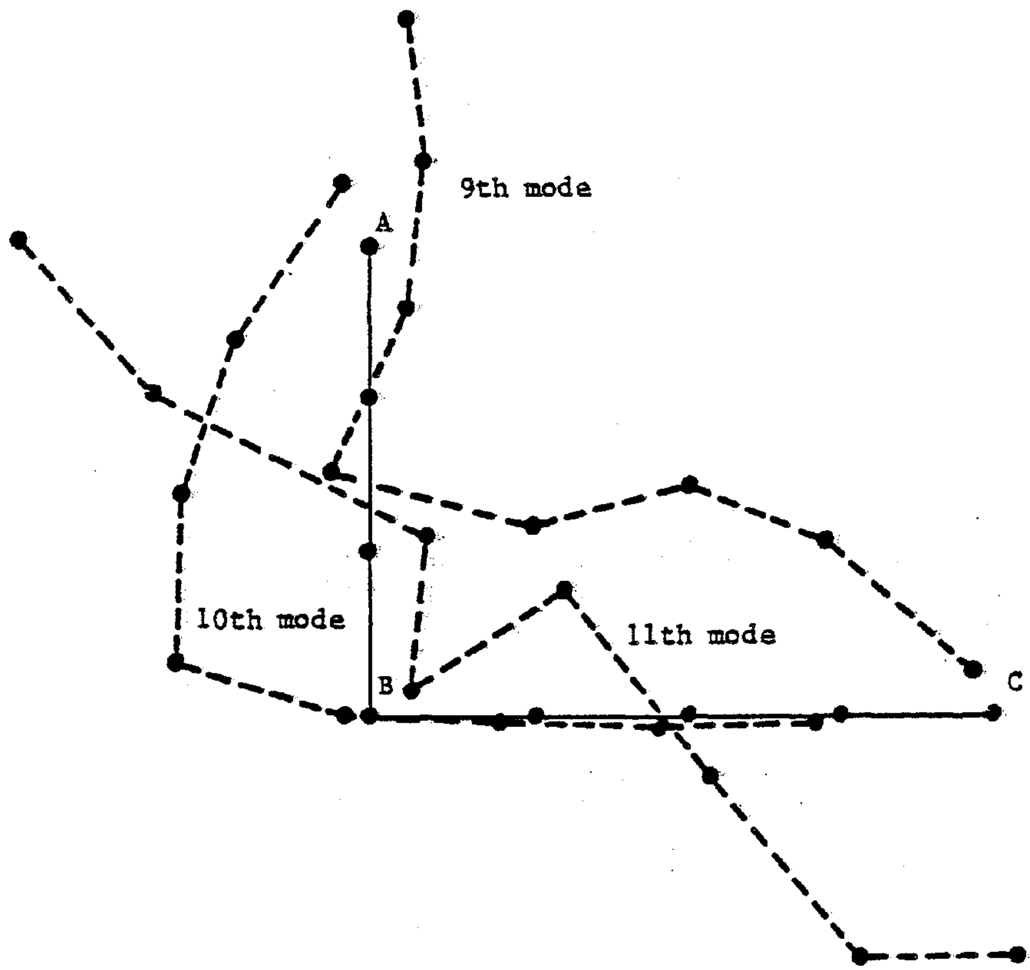


Figure 8 Plan View of Roof Edge Vibration Mode Shapes

— 9th to 11th modes

5.0 CONCLUSION

Modal and transient analyses needed for evaluation of dynamic characteristics and responses of a building to ground accelerations are time and core intensive computations. To save computing time and/or solve the problem at a lower core requirement, reduction techniques such as static condensation, Guyan reduction or component mode synthesis can be applied to reduce a full scale finite element model to a smaller size before these analyses are executed.

Literature review showed that a class of computer codes developed specially for buildings are based on the assumption that floor systems are rigid in plane. It is an oversimplification that can lead to serious errors in some cases. Assuming inadequate diaphragm design, examples are: buildings with an L,T, H or U-shape floor plan, buildings having setback or local irregularites, building/space-frames supporting heavy equipment on floors.

The review also revealed that the application of fixed-interface component mode synthesis to buildings has been limited to a highly idealized 'shear building' model or very small plane frame. As will be discussed later, what works beautifully on small 2-D problems does not necessarily work well on larger 3-D problems. The review also showed that although the method has been incorporated in the MSC/NASTRAN, it was not developed specifically for the given structure and load case for which special treatments can lead to improved computing efficiency.

The main objective of this work originally was to formulate and implement the method as applied to the case of a building subjected to ground accelerations, and to examine its feasibility, advantages and disadvantages. It was hoped that the average number of DOF per floor that must be retained would be somewhat larger, but not much greater, than three DOF per floor; and that the results would be fairly accurate, with all the important dynamic properties preserved in the mode shapes of the reduced system.

That goal has been accomplished, as can be seen from the summaries and conclusions to be presented in the next paragraphs.

5.1 Summary and Conclusion

The fixed-interface component mode method was first applied to determine the natural frequencies and mode shapes of a twenty-story 2-D frame. The accuracy of the results is satisfying. (See APPENDIX E). As attempts were made to analyze a medium sized 3-D building, however, the following difficulties were encountered: (a) The component in itself was big enough to warrant treatment prior to component eigensolution. (b) The system matrices assembled after synthesis were still big, because the method merely reduces interior DOF while retaining all boundary DOF.

A unique combination of reduction procedures, with fixed-interface component mode synthesis as the central theme augmented by static condensation and Guyan reduction, was therefore formulated and implemented for the given structure and load case. Of course, the structure in question must be linearly elastic. The combined procedures were conceived to take advantage of the stiffness characteristics of a building. Although they have been known for some time, no work has been done to date to combine them in order to let them complement one another and become more powerful.

In this work, the applicability and consequences of each method as well as the similarities and differences among them were examined. How they may be justifiably applied in a specified sequence was explained. In essence, static condensation reduces the matrices entering component eigensolution. The method of CMS transforms component matrices to reduced matrices defined over boundary DOF and a truncated set of normal mode shapes extracted from components with fixed boundaries. Guyan reduction eliminates DOF on the boundary after synthesis. In addition, by the choice of the manner in which a few intermediate steps can be treated, several simplified transformations for carrying out modal synthesis were derived to upgrade computing efficiency.

A program package was developed. The matrices former, reducers and solvers as well as the package were validated. The package will direct the computer to read data and form component matrices, accept specifications for retaining interior and boundary DOF that are arbitrarily patterned, and then perform three stage reductions and solve for natural frequencies, mode shapes and displacement responses on a much reduced system model. The program uses dynamic core allocation and out-of-core operations so that until reduced forms are obtained, only one major matrix, whether it be stiffness or mass, component or assembled system, will occupy the CPU at a given time. In developing the package, much attention was given to economizing computing and core use. For example, several subroutines were written to replace the subroutine 'NROOT' in the IBM Scientific Subroutines Package (SSP). Roughly one third of the core need is thus saved.

The combined reduction procedures were applied to carry out dynamic analyses of a twelve-story 3-D building. Several solutions were made of reduced models by changing parameters such as the number of components,the number of retained component modes, and the number of retained DOF per boundary. The results demonstrated the importance of the floor fexibility in modes as low as the fourth for the case studied. The resluts also consistently showed that good convergence was achieved by much-reduced models, a pleasant but not at all surprising finding indeed. A rationale is offered in the next paragraph.

Much credit should be attributed to the Ritz or component mode method and to Guyan reduction. But perhaps the characteristics of the given structure and load case deserve some attention. The given structure and load case can be characterized as follows: (a) The floor systems are stiff compared to the whole building laterally. For a typical floor, only its most flexible local modes need to be represented in the reduced model. (b) The energy contents in the high frequency components of ground accelerations are lower than that in the low frequency components. (c) Due to the zigzagging of higher mode shapes, their participation in the total response of a building is lower than that of the lower modes. Therefore, during the three stage reduction process, we have a choice to (a) retain a relatively small number of interior DOF for component eigensolution, (b) retain a relatively small number of component normal modes for transformation and synthesis, (c) retain a relatively small number of boundary DOF in the synthesized matrices, and (d) retain a relatively small number of decoupled normal modes of the reduced system, and still expect to obtain system results without significant loss in accuracy.

Admittedly, the procedures are subjected to the following penalties: (a) Component eigensolutions are required. (b) Many transformations are needed. But the payoffs are large savings in core achieved by substructuring, and huge savings in computing time to be gained by performing eigensolution and transient analyses on a much smaller model. By comparing the alternatives, it is obvious that the gains far exceed the penalities.

5.2 Suggestions

Suggestions for future works are as follows:

1. The program package as is can be readily applied to structures such as buildings, bridges, space frames, piping and some plant equipment under seismic loads if linearity is satisfied. Minor changes can be made in the program for application to other structures and load cases.

2. Consider multilevel substructuring hierarchy, i.e. substructures within a substructure. This will greatly enhance the capacity of the program.

3. Establish a good criterion for retaining component modes. Hurty suggested that the cut-off frequency of component modes be 50 % higher than the highest frequency of interest. Based on the characteristics of building and ground accelerations, it is suggested that this criterion be relaxed; or alternatively, one may discard a component mode if the absolute value of the product of its participation factor and dynamic factor falls below a certain number, which is a fraction times that of the most significant component mode.

4. Expand the program: Add elements such as a beam with rigid ends, a beam with flexible joints, a plane stress element for shear wall and a solid element for soil strata supporting the foundation.

APPENDICES

APPENDIX A

STIFFNESS MATRIX FOR 3-D PRISMATIC BEAM

APPENDIX A

STIFFNESS MATRIX FOR 3-D PRISMATIC BEAM

This appendix describes a 3-D prismatic beam element which does not require the third node to define the direction of the major principal axis of its section. (The concept was used in STRUDL and ANSYS.) The stiffness matrix defines a force-displacement relationship as follows,

$$[k^*]\ \{u^*\} = \{p^*\}$$

where both $\{u^*\}$ and $\{p^*\}$ consist of 12 components: 3 translational and 3 rotational terms at each one of the two beam ends. The stiffness matrix $[k^*]$ defined in the local coordinate system (x^*,y^*,z^*) is shown in Table A-2. The local x^*-axis extends from one end denoted by node number 'i' to the other end 'j'. It coincides with the centroidal axis of the beam. The local coordinates are parallel to the principal axes of the section.

If the local coordinates (x^*,y^*,z^*) are related to the global coordinates (X,Y,Z) by

$$(x^*,y^*,z^*)' = [T^*]\ (X,Y,Z)',$$

then the nodal displacements in local coordinates $\{u^*\}$ and the nodal displacements in global coordinates $\{u\}$ can be related by

$$\{u^*\} = [T]\{u\}$$

The stiffness matrix in global coordinates is then

$$[k] = [T]'[k^*][T]$$

This transformation can be derived from the potential energy

$$V = (0.5)\{u^*\}'[k^*]\{u^*\} = (0.5)\{u\}'[k]\{u\}$$

The local coordinate system (x^*, y^*, z^*) is shown in Figure A-1. The transformation matrices $[T^*]$ and $[T]$ are given in Table A-1. The definitions of local coordinates for a beam in an arbitrary direction (x^*, y^*, z^*) and for a beam whose axis coincides with any one of the three global axes are shown in Figure A-1.

Table A-1. Coordinate Transformation Matrices

$$
(T) = \begin{bmatrix} T^* & 0 & 0 & 0 \\ 0 & T^* & 0 & 0 \\ 0 & 0 & T^* & 0 \\ 0 & 0 & 0 & T^* \end{bmatrix}
$$

$$
(T^*) = \left[\begin{array}{c|c|c} C_\alpha C_\beta & S_\alpha C_\beta & S_\beta \\ \hline -C_\alpha S_\beta S_\theta - S_\alpha C_\theta & C_\alpha C_\theta - S_\alpha S_\beta S_\theta & S_\theta C_\beta \\ \hline S_\alpha S_\theta - C_\alpha S_\beta C_\theta & -S_\alpha S_\beta C_\theta - C_\alpha S_\theta & C_\theta C_\beta \end{array}\right]
$$

$C_\alpha = \cos\alpha$, $C_\beta = \cos\beta$, $C_\theta = \cos\theta$

$S_\alpha = \sin\alpha$, $S_\beta = \sin\beta$, $S_\theta = \sin\theta$

Table A-2. Stiffness Matrix for a 3-d Uniform 'beam' in local coordinates

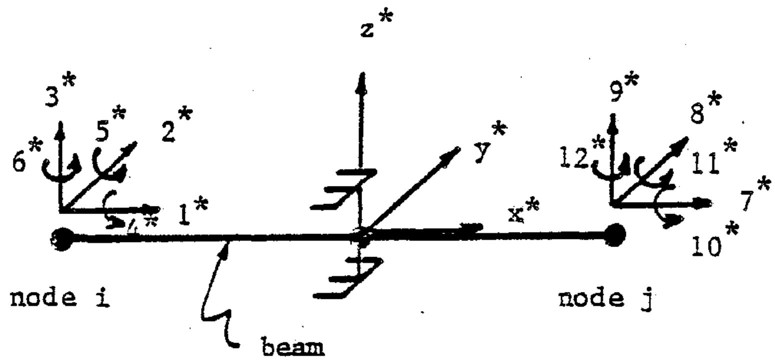
$$
[k^*_e] = \begin{array}{cccccccccccc|c}
1^* & 2^* & 3^* & 4^* & 5^* & 6^* & 7^* & 8^* & 9^* & 10^* & 11^* & 12^* & \\
a_1 & 0 & 0 & 0 & 0 & 0 & -a_1 & 0 & 0 & 0 & 0 & 0 & 1^* \\
0 & C_1 & 0 & 0 & 0 & C_2 & 0 & -C_1 & 0 & 0 & 0 & C_2 & 2^* \\
0 & 0 & d_1 & 0 & -d_2 & 0 & 0 & 0 & -d_1 & 0 & -d_2 & 0 & 3^* \\
0 & 0 & 0 & t_1 & 0 & 0 & 0 & 0 & 0 & -t_1 & 0 & 0 & 4^* \\
0 & 0 & -d_2 & 0 & 2d_3 & 0 & 0 & 0 & d_2 & 0 & d_3 & 0 & 5^* \\
0 & C_2 & 0 & 0 & 0 & 2C_3 & 0 & -C_2 & 0 & 0 & 0 & C_3 & 6^* \\
-a_1 & 0 & 0 & 0 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 & 0 & 7^* \\
0 & -C_1 & 0 & 0 & 0 & -C_2 & 0 & C_1 & 0 & 0 & 0 & -C_2 & 8^* \\
0 & 0 & -d_1 & 0 & d_2 & 0 & 0 & 0 & d_1 & 0 & d_2 & 0 & 9^* \\
0 & 0 & 0 & -t_1 & 0 & 0 & 0 & 0 & 0 & t_1 & 0 & 0 & 10^* \\
0 & 0 & -d_2 & 0 & d_3 & 0 & 0 & 0 & d_2 & 0 & 2d_3 & 0 & 11^* \\
0 & C_2 & 0 & 0 & 0 & C_3 & 0 & -C_2 & 0 & 0 & 0 & 2C_3 & 12^*
\end{array}
$$

$a_1 = (\frac{EA}{\ell})$

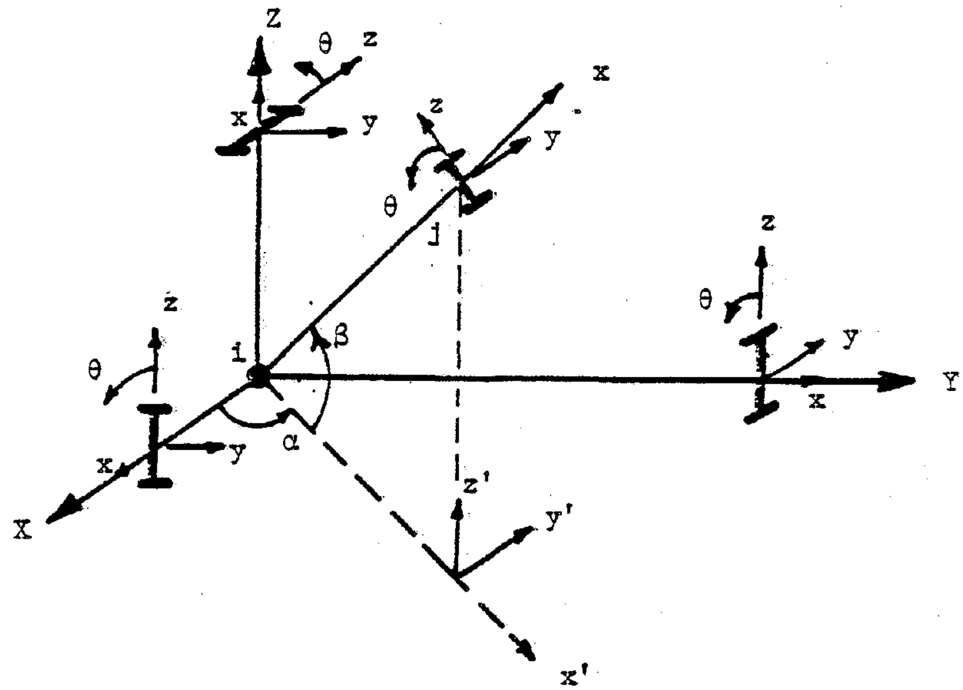
$t_1 = (\frac{GJ}{\ell})$

$C_1 = 12\ (EI_z{}^*/\ell^3)$, $C_2 = 6(EI_z{}^*/\ell^2)$, $C_3 = 2(EI_z{}^*/\ell)$

$d_1 = 12\ (EI_y{}^*/\ell^3)$, $d_2 = 6(EI_y{}^*/\ell^2)$, $d_3 = 2(EI_y{}^*/\ell)$



(a). Local Coordinates (x^*, y^*, z^*) & DOFs @ Beam Ends



(x^*, y^*, z^*) = rotation of (x,y,z) with respect to x by θ
(x, y, z) = rotation of (x',y',z') with respect to y' by β
(x',y',z') = rotation of (X,Y,Z) with respect to Z by α

(b). Global coordinates (X,Y,Z) & Local Coordinates

Figure A-1 Coordinate Systems for 3-D Beam

APPENDIX B

STANDARIZATION

OF GENERAL EIGENVALUE PROBLEM

APPENDIX B

STANDARDIZATION OF GENERAL EIGENVALUE PROBLEM

The general eigenvalue problem

$$[K][U] = [M][U][\,\grave{}\,w^2] \tag{B-1}$$

where both $[K]$ and $[M]$ are symmetric, is to be reduced to a standard form

$$[\underline{K}][\underline{U}] = [\,\grave{}\,I][\underline{U}][\,\grave{}\,w^2]$$

so that the subroutine 'EIGEN' in IBM Scientific Subroutine Package may be applied. The dimension of the matrices and vector is n. The solutions to be sought are eigenvalues $[\,\grave{}\,w^2]$ and eigenvectors $[U]$[34].

The first step is to solve a standard eigenvalue problem

$$[M][Z] = [Z][\,\grave{}\,r]$$

to obtain eigenvalues r_j, j=1,2 ..n, and the corresponding modal matrix $[Z]$, which is normalized such that

$$[Z]'[Z] = [\,\grave{}\,I] \quad \text{and hence} \quad [Z]' = [Z]^{-1}$$

By orthonormality, we have

$$[M] = [Z][\,\grave{}\,r][Z]'$$

If $[M]^{0.5}$ and $[M]^{-0.5}$ are defined such that

$$[M]^{0.5}\,[M]^{0.5} = [M]$$
$$[M]^{0.5}\,[M]^{-0.5} = [\,\grave{}\,I]$$

then it can be verified that

$$[M]^{0.5} = [Z][{}^{\backprime}r^{0.5}][Z]'$$

$$[M]^{-0.5} = [Z][{}^{\backprime}r^{-0.5}][Z]'$$

Hence Eq.(B-1) can be rewritten as

$$[M]^{-0.5}[K][M]^{-0.5}[M]^{0.5}[U] =$$

$$[M]^{-0.5}[M]^{0.5}[M]^{0.5}[U][{}^{\backprime}w^2]$$

By introducing

$$[\underline{K}] = [M]^{-0.5}[K][M]^{-0.5}$$

$$[\underline{U}] = [M]^{0.5}[U]$$

Eq.(B-1) is reduced to

$$[\underline{K}][\underline{U}] = [\underline{U}][{}^{\backprime}w^2]$$

which is in standard form with $[\underline{U}]$ and $[{}^{\backprime}w^2]$ as its solutions. The solutions for Eq.(B-1) are $[U]$ and $[{}^{\backprime}w^2]$, where

$$[U] = [M]^{-0.5}[\underline{U}]$$

APPENDIX C

SOLUTION OF LINEAR SDOF SYSTEM RESPONSE

APPENDIX C

SOLUTION OF LINEAR SDOF SYSTEM RESPONSE

This appendix describes a step-wise explicit integration method for solving the response of a linear SDOF system subjected to piece-wise linear loads. The method is accurate because the solutions at the end of each step are based on explicit expressions derived from integration. The routine is highly efficient because, for a linear system, the coefficients in the recurrent formulae need to be calculated only once if the time increment is constant and because the answers at the end of each step are simple algebraic expressions.

Let the equation of motion of a SDOF system be

$$m\ddot{y}(t) + c\dot{y}(t) + ky(t) = p(t) = p_0 f(t)$$

or

$$\ddot{y}(t) + 2d_r w\dot{y}(t) + w^2 y(t) = (p_0/m)\ f(t)$$

The numerical integration is to be carried out step-wise at time increments dt, which may be constant within a range. If the state variables $(y_i, \dot{y}_i)$ at $t=t_i$ are known, and the loading between t_i and t_{i+1} is linear, namely,

$$p(x) = p_i + (p_{i+1}-p_i)\ (x/dt)$$

where $x=t-t_i$ is not larger than dt, then the state variables $(y_{i+1}, \dot{y}_{i+1})$ at $t=t_{i+1}$ can be determined by the following recurrent formulae

$$y_{i+1} = A(p_i) + B(p_{i+1}) + C(y_i) + D(\dot{y}_i)$$

$$\dot{y}_{i+1} = A'(p_i) + B'(p_{i+1}) + C'(y_i) + D'(\dot{y}_i)$$

where

$$A = \{ E_1[(-Z_1-d_r w dt)S_1/w_d + (-2d_r/w-dt)C_1] + 2d_r/w \}/(kdt)$$

$$B = \{ E_1[(Z_1)S_1/w_d + (2d_r/w)C_1] - 2d_r/w + dt \}/(kdt)$$

$$C = E_1 \, [C_1 + (d_r w/w_d)S_1]$$

$$D = (1/w_d)E_1 S_1$$

$$A' = (1/kdt) \, \{E_1[(d_r w + w^2 dt)S_1/w_d + C_1] - 1\}$$

$$B' = (1/kdt) \, \{-E_1[(d_r w/w_d)S_1 + C_1] + 1\}$$

$$C' = -(w^2/w_d)E_1 S_1$$

$$D' = E_1[C_1 - (d_r w/w_d)S_1]$$

in which

$$E_1 = e^{-d_r w dt}$$

$$Z_1 = 2d_r^{\,2} - 1$$

$$C_1 = \cos(w_d dt)$$

$$S_1 = \sin(w_d dt)$$

$$d_r = c/(2m)$$

$$w_d = w \, (1 - d_r^{\,2})^{1/2}$$

Save for minor differences in form, these equations are the same as those in Craig's book(35).

APPENDIX D

DYNAMIC ANALYSES OF A SIX-STORY 3-D FRAME

This example was done to validate the procedures and program by comparing results to that obtained from SUPERSAP, a general purpose finite element program. The verification is in addition to many self-sustained tests which the program has passed.

Figures D-1 and D-2 show the perspective view of the frame and the floor plan, respectively. The frame is distinctively weaker in the Y-direction. The floor plan and mass distribution are the same for all floors. Each floor weighs 48.7 kips, which is equivalent to about 150 psf. The floor systems are braced for in-plane rigidity. The section properties are given in Table D-1.

The structure was divided into three components that are bounded by two common interfaces and a roof boundary. There is one interior floor in each component. Figure D-2 shows the 8 interior DOF that were retained after static condensation. The same retention pattern was used for Guyan reduction of the boundary DOF in a later step. Each component eigensolution resulted in 8 normal modes, all of which were retained.

The assembled system model has 80 DOF : 8X3 normal modes, 8 boundary DOF on the roof and 4X6 DOF on each of the two common boundaries. After Guyan reduction, the size of the system model was cut down to 48 DOF.

The same frame was solved for system natural frequencies and mode shapes using SUPERSAP, a program which does not employ a reduction technique prior to eigensolution. Table D-2 shows a comparison of the two sets of results. In the table, the frequencies calculated by FEM are ranked in ascending order, while that obtained by CMS are not. The intention is to compare modes based on mode types or characteristics. The agreement is excellent for the first three bending modes in the Y-direction, the first three bending modes in the X-direction, the first three torsional modes and the first three rocking modes. The agreement for higher modes is less satisfying.

Table D-3 shows a comparison of the two sets of mode shapes. The quality of agreement is similar to that for the natural frequencies, although the agreemnet for two higher modes is poor. The mode shapes calculated by SUPERSAP indeed confirm that in this case every floor behaves like a rigid body; hence the retention pattern used in the CMS solution is proper.

The displacements in the Y-direction at nodes No.11, No.19 and No.27 in response to the Imperial Valley Earthquake, better known as the El Centro earthquake, of May 18, 1940, scaled to a magnitude of 0.2G, are shown in Table D-5. The accelerations were applied in the Y-direction. Although the responses of individual modes were computed starting from at rest at t=0, they were added to obtain system responses starting from a later time step in order to save computation and printing.

Because of low energy contents in high frequency excitations, and the low participation by higher modes (see Table D-4), the lack of good agreement for higher modes should not be a problem of concern. It is therefore concluded that the accuracy of the results is satisfying.

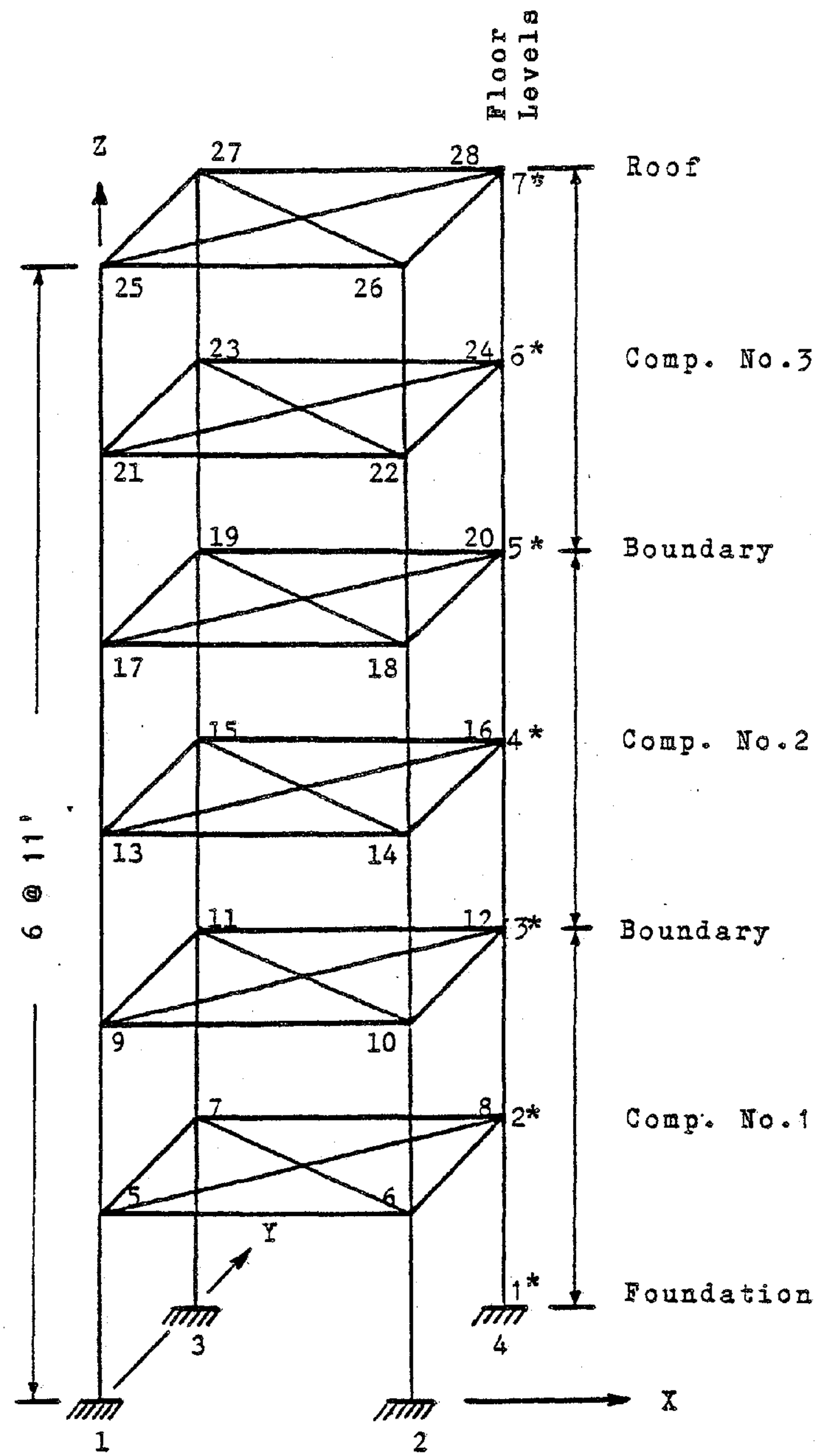
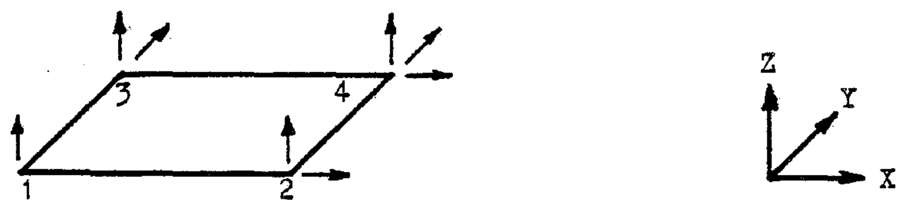
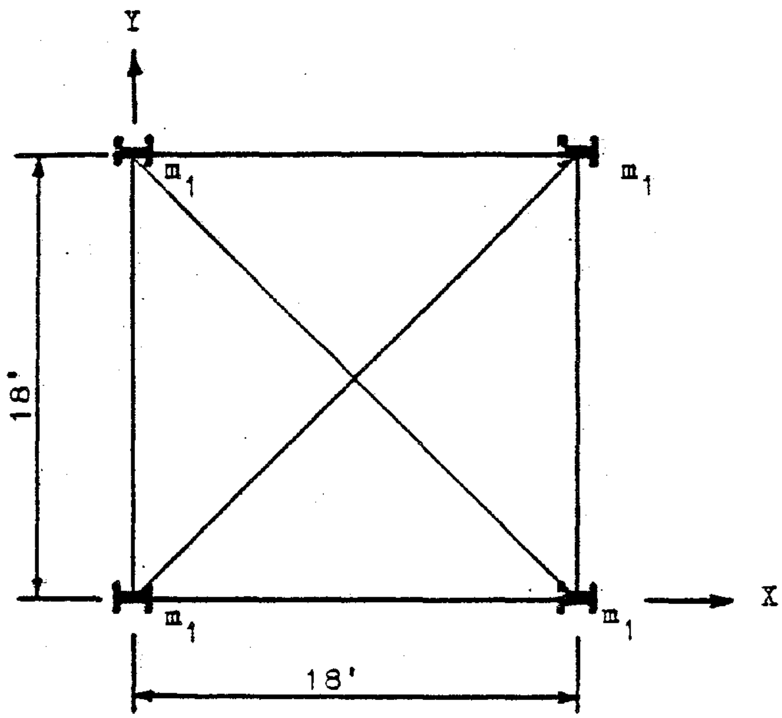


Figure D-1 A Perspective View of the 3-D Frame



(a). Retained DOF on All Floors



m_1 = 0.0315 kip.Sec^2/in.

E = 29500.ksi

(b). Floor and Roof Framing Plan

Figure D-2 Floor Plan and Retention Pattern

Table D-1 Section Properties, Example 1

I_c	I_{sp}	A	I_y	I_z	J	Remarks
1	1	32.60	1240.0	447.00	7.12	column
2	2	26.50	999.0	362.00	4.06	column
3	3	21.80	796.0	134.00	3.88	column
1	4	14.70	800.0	40.10	1.24	floor beam
2,3	5	11.80	518.0	28.90	0.79	floor beam
1,2,3	6	3.83	11.3	3.86	0.15	floor bracing

I_c = component number

I_{sp} = section property number

Table D-2 Comparison of Calculated System Natural Frequencies, CPS

Mode Number	FEM	CMS		Mode Type		
1	1.00305	1.0030		$(\delta_y)_1$		
2	1.1411	1.1410				$(\theta_z)_1$
3	1.2164	1.2164	$(\delta_x)_1$			
4	2.7622	2.7706		$(\delta_y)_2$		
5	3.3812	3.3930				$(\theta_z)_2$
6	3.8421	3.8550	$(\delta_x)_2$			
7	4.9048	4.9324		$(\delta_y)_3$		
8	6.1845	6.1957				$(\theta_z)_3$
9	6.6624	7.0121		δ_y		
10	7.2536	7.2480	$(\delta_x)_3$			
11	9.4737	10.984				θ_z
12	9.5517	10.687		δ_y		
13	11.707	14.439	δ_x			
14	12.747	12.963		δ_y		
15	12.908	14.108				θ_z
16	16.498	16.565				θ_z
17	16.570	17.925	δ_x			
18	17.329	17.329			$(\delta_z)_1$	
19	17.748	17.747			$(\delta_z)_2$	
20	18.052	18.074			$(\delta_z)_3$	

Table D-3 Comparison of Mode Shapes - 1st Mode

Node number	FEM δ_y	CMS δ_y
9	0.4938	-
10	0.4938	-
11	0.4938	0.4941
12	0.4938	0.4939
17	1.173	-
18	1.173	-
19	1.173	1.174
20	1.173	1.173
25	1.774	-
26	1.774	-
27	1.774	1.775
28	1.774	1.774

Table D-3 Comparison of Mode Shapes - 2nd Mode (Continued)

Node number	FEM δ_x	FEM δ_y	CMS δ_x	CMS δ_y
9	0.3513	-0.3516	-	-
10	0.3513	0.3516	0.3515	-
11	-0.3513	-0.3516	-	-0.3516
12	-0.3513	0.3516	-0.3516	0.3519
17	0.8678	-0.8651	-	-
18	0.8678	0.8651	0.8682	-
19	-0.8678	-0.8651	-	-0.8652
20	-0.8678	0.8651	-0.8684	0.8659
25	1.2279	-1.2288	-	-
26	1.2279	1.2288	1.228	-
27	-1.2279	-1.2288	-	-1.229
28	-1.2279	1.2288	-1.229	1.230

Table D-3 Comparison of Mode Shapes - 3rd Mode (Continued)

Node number	FEM δ_x	CMS δ_x
9	0.4832	-
10	0.4832	0.4834
11	0.4832	-
12	0.4832	0.4834
17	1.2351	-
18	1.2351	1.236
19	1.2351	-
20	1.2351	1.236
25	1.7400	-
26	1.7400	1.741
27	1.7400	-
28	1.7400	1.741

Table D-3 Comparison Mode Shapes - 4th Mode (Continued)

Node number	FEM δ_y	CMS δ_y
9	-1.2465	-
10	-1.2465	-
11	-1.2465	-1.254
12	-1.2465	-1.254
17	-1.1089	-
18	-1.1089	-
19	-1.1089	-1.118
20	-1.1089	-1.118
25	1.5626	-
26	1.5626	-
27	1.5626	1.585
28	1.5626	1.585

Table D-3 Comparison of Mode Shapes - 5th Mode (Continued)

Node number	FEM δ_x	FEM δ_y	CMS δ_x	CMS δ_y
9	-0.9288	0.9358	-	-
10	-0.9288	-0.9358	-0.9363	-
11	0.9288	0.9358	-	0.9436
12	0.9288	-0.9358	0.9364	-0.9435
17	-0.6695	0.6906	-	-
18	-0.6695	-0.6906	-0.6760	-
19	0.6695	0.6906	-	0.6974
20	0.6695	-0.6906	0.6761	-0.6974
25	1.0963	-1.1094	-	-
26	1.0963	1.1094	1.116	-
27	-1.0963	-1.1094	-	-1.129
28	-1.0963	1.1094	-1.116	1.129

Table D-3 Comparison of Mode Shapes - 6th Mode (Continued)

Node number	FEM δ_x	CMS δ_x
9	1.3540	-
10	1.3540	1.366
11	1.3540	-
12	1.3540	1.366
17	0.9131	-
18	0.9131	0.9226
19	0.9131	-
20	0.9131	0.9225
25	-1.5328	-
26	-1.5328	-1.560
27	-1.5389	-
28	-1.5389	-1.560

Table D-3 Comparison of Mode Shapes - 7th Mode (Continued)

Node number	FEM δ_y	CMS δ_y
9	1.4430	-
10	1.4430	-
11	1.4430	1.524
12	1.4430	1.524
17	-1.2203	-
18	-1.2203	-
19	-1.2203	-1.257
20	-1.2203	-1.258
25	1.2692	-
26	1.2692	-
27	1.2692	1.210
28	1.2692	1.210

Table D-3 Comparison of Mode Shapes - 8th Mode (Continued)

Node number	FEM δ_x	FEM δ_y	CMS δ_x	CMS δ_y
9	1.0530	-1.0925	-	-
10	1.0530	1.0925	1.097	-
11	-1.0530	-1.0925	-	-1.139
12	-1.0530	1.0925	-1.097	1.139
17	-0.9835	1.0229	-	-
18	-0.9835	-1.0229	-1.005	-
19	0.9835	1.0229	-	1.048
20	0.9835	-1.0229	1.006	-1.048
25	0.8516	-0.9033	-	-
26	0.8516	0.9033	0.8125	-
27	-0.8516	-0.9033	-	-0.8629
28	-0.8516	0.9033	-0.8127	0.8629

Table D-3 Comparison of Mode Shapes - 9th Mode (Continued)

Node number	FEM δ_y	CMS δ_y
9	0.9293	-
10	0.9293	-
11	0.9293	0.7592
12	0.9293	0.7591
17	-1.1421	-
18	-1.1421	-
19	-1.1421	-1.354
20	-1.1421	-1.354
25	-0.8419	-
26	-0.8419	-
27	-0.8419	-0.8963
28	-0.8419	-0.8963

Table D-3 Comparison of Mode Shapes - 10th Mode (Continued)

Node number	FEM δ_x	CMS δ_x
9	1.5263	-
10	1.5263	1.581
11	1.5263	-
12	1.5263	1.581
17	-1.4586	-
18	-1.4586	-1.487
19	-1.4586	-
20	-1.4586	-1.487
25	1.2518	-
26	1.2518	1.205
27	1.2518	-
28	1.2518	1.205

Table D-3 Comparison of Mode Shapes - 11th Mode (Continued)

Node number	FEM δ_x	FEM δ_y	CMS δ_x	CMS δ_y
9	-0.5095	0.5708	-	-
10	-0.5095	-0.5708	-0.3493	-
11	0.5095	0.5708	-	0.4153
12	0.5095	-0.5708	0.3493	-0.4152
17	0.3274	-0.4326	-	-
18	0.3274	0.4326	0.6005	-
19	-0.3274	-0.4326	-	-0.8419
20	-0.3274	0.4326	-0.6005	0.8417
25	0.5749	-0.6878	-	-
26	0.5749	0.6878	0.5958	-
27	-0.5749	-0.6878	-	-0.7852
28	-0.5749	0.6878	-0.5959	0.7851

Table D-3 Comparison of Mode Shapes - 12th Mode (Continued)

Node number	FEM δ_y	CMS δ_y
9	-0.1213	-
10	-0.1213	-
11	-0.1213	-0.1736
12	-0.1213	-0.1736
17	1.4817	-
18	1.4817	-
19	1.4817	1.210
20	1.4817	1.210
25	0.1507	-
26	0.1507	-
27	0.1507	0.1215
28	0.1507	0.1229

Table D-3 Comparison of Mode Shapes - 13th Mode (Continued)

Node number	FEM δ_x	CMS δ_x
9	-0.7240	-
10	-0.7240	-0.4946
11	-0.7240	-
12	-0.7240	-0.4946
17	0.2737	-
18	0.2737	0.8239
19	0.2737	-
20	0.2737	0.8239
25	0.8611	-
26	0.8611	0.9606
27	0.8611	-
28	0.8611	0.9609

Table D-3 Comparison of Mode Shapes - 14th Mode (Continued)

Node number	FEM δ_y	CMS δ_y
9	1.7828	-
10	1.7828	-
11	1.7828	1.786
12	1.7828	1.786
17	0.5834	-
18	0.5834	-
19	0.5834	0.6569
20	0.5834	0.6569
25	0.0227	-
26	0.0227	-
27	0.0227	-0.0289
28	o.0227	-0.0289

Table D-3 Comparison of Mode Shapes - 15th Mode (Continued)

Node	FEM		CMS	
number	δ_x	δ_y	δ_x	δ_y
9	-0.2298	0.2805	-	-
10	-0.2298	-0.2805	-0.1674	-
11	0.2298	0.2805	-	0.2112
12	0.2298	-0.2805	0.1676	-0.2110
17	0.9469	-1.3211	-	-
18	0.9469	1.3211	0.7169	-
19	-0.9469	-1.3211	-	-1.109
20	-0.9469	1.3211	-0.7173	1.109
25	0.0201	-0.3048	-	-
26	0.0201	0.3048	0.0481	-
27	-0.0201	-0.3048	-	-0.0952
28	-0.0201	0.3048	-0.0485	0.0951

Table 3-D Comparison of Mode Shapes - 16th Mode (Continued)

Node number	FEM δ_x	FEM δ_y	CMS δ_x	CMS δ_y
9	0.9322	-1.4336	-	-
10	0.9322	1.4336	0.8968	-
11	-0.9322	-1.4336	-	-1.493
12	-0.9322	1.4336	-0.8968	1.493
17	0.3638	-0.6544	-	-
18	0.3638	0.6544	0.3497	-
19	-0.3638	-0.6544	-	-0.6671
20	-0.3638	0.6544	-0.3498	0.6671
25	0.0278	-0.0715	-	-
26	0.0278	0.0715	-0.0153	-
27	-0.0278	-0.0715	-	0.0317
28	-0.0278	0.0715	0.0153	-0.0317

Table D-3 Comparison of Mode Shapes - 17th Mode (Continued)

Node number	FEM δ_x	CMS δ_x
9	0.5500	-
10	0.5500	0.3948
11	0.5500	-
12	0.5500	0.3948
17	-1.5985	-
18	-1.5985	-1.395
19	-1.5985	-
20	-1.5985	-1.395
25	-0.4420	-
26	-0.4420	-0.2146
27	-0.4420	
28	-0.4420	-0.2146

Table D-3 Comparison of Mode Shapes - 18th Mode (Continued)

Node number	FEM δ_z	CMS δ_z
9	0.6440	0.6440
10	0.6440	0.6440
11	0.6440	0.6440
12	0.6440	0.6440
17	1.2577	1.258
18	1.2577	1.258
19	1.2577	1.258
20	1.2577	1.258
25	1.6208	1.621
26	1.6208	1.621
27	1.6208	1.621
28	1.6208	1.621

Table D-3 Comparison of Mode Shapes - 19th Mode (Continued)

Node number	FEM δ_z	CMS δ_z
9	0.6396	0.6390
10	0.6396	0.6402
11	-0.6396	-0.6402
12	-0.6396	-0.6391
17	1.2549	1.254
18	1.2549	1.256
19	-1.2549	-1.256
20	-1.2549	-1.254
25	1.6242	1.623
26	1.6242	1.623
27	-1.6242	-1.625
28	-1.6242	-1.623

Table D-3 Comparison of Mode Shapes - 20th Mode (Continued)

Node number	FEM δ_z	CMS δ_z
9	-0.6381	-0.6365
10	0.6381	0.6355
11	-0.6381	-0.6356
12	0.6381	0.6366
17	-1.2488	-1.250
18	1.2488	1.248
19	-1.2488	-1.248
20	1.2488	1.250
25	-1.6101	-1.609
26	1.6101	1.607
27	-1.6101	-1.607
28	1.6101	1.610

Table D-4 Modal Earthquake Excitation Factor

Mode Number		Frequency CPS	Modal Earthquake Excitation Factor
1	δ_y	1.0030	58.7828
2		1.1410	- 0.0177
3		1.2164	0.0057
4	δ_y	2.7706	-24.8687
5		3.3930	- 0.0007
6		3.8550	0.0131
7	δ_y	4.9324	-15.3648
8		6.1957	0.0008
9	δ_y	7.0121	- 7.2553
10		7.2480	0.0466
11	δ_y	10.6872	11.0114
12		10.9842	0.0006
13	δ_y	12.9672	5.5799
14		14.1082	0.0000
15		14.4391	0.0223
16		16.5652	0.0001
17		17.3292	0.0000
18		17.7474	0.0377
19		17.9250	0.0402
20		18.0735	- 0.0018

Note: Modes are ranked in ascending order of natural frequencies.

Table D-5 Displacement Responses

t=0.	-0.004	-0.032	-0.030	-0.026	-0.028
	-0.035	-0.042	-0.037	-0.032	-0.025
	-0.025	-0.038	-0.052	-0.057	-0.047
	-0.042	-0.032	-0.024	-0.012	-0.019
	-0.038	-0.056	-0.057	-0.019	0.009
	0.041	-0.014	-0.037	-0.042	-0.059
	-0.076	-0.095	-0.090	-0.050	-0.058
	-0.048	-0.048	-0.020	0.007	0.044
	0.069	0.074	0.098	0.135	0.144
	0.123	0.105	0.079	0.069	0.099
t=1.	0.121	0.155	0.187	0.214	0.191
	0.175	0.117	0.117	0.018	-0.151
	-0.230	-0.176	-0.142	-0.073	-0.017
	0.039	0.090	0.146	0.208	0.291
	0.357	0.447	0.424	0.338	0.274
	0.261	0.271	0.246	0.264	0.291
	0.354	0.096	-0.432	-0.605	-0.582
	-0.595	-0.531	-0.505	-0.513	-0.513
	-0.528	-0.477	-0.394	-0.318	-0.229
	-0.126	-0.005	0.105	0.230	0.341
t=2.	0.468	0.574	0.706	0.799	0.888
	0.936	1.000	0.826	0.680	-0.351
	-0.694	-0.480	-0.546	-0.320	-0.220
	-0.051	0.033	0.156	0.262	0.347
	0.514	0.169	-0.770	-0.453	-0.506
	-0.296	-0.169	0.069	-0.196	-0.579
	-0.480	-0.493	-0.433	-0.360	-0.293
	-0.220	-0.153	-0.079	-0.013	0.055
	-0.028	-0.127	-0.245	-0.278	-0.210
	-0.175	-0.098	-0.032	0.054	0.123
t=3.	0.197	-0.028	-0.109	-0.012	0.003
	0.101	0.165	0.258	0.331	0.399
	0.064	0.071	0.200	0.202	0.386
	0.396	0.597	-0.272	-0.383	-0.203
	-0.160	0.021	0.198	-0.312	-0.435
	-0.313	-0.340	-0.223	-0.164	-0.063
	-0.037	-0.197	-0.095	-0.099	-0.032
	0.005	0.088	0.143	0.178	0.065
	-0.009	-0.072	0.023	0.062	0.166
	0.242	0.353	0.433	0.508	0.123

Table D-5 Displacement Responses (Continued)

i	t_i	$\ddot{u}_g(t)$	$(\delta_y)_{11}$	$(\delta_y)_{19}$	$(\delta_y)_{27}$
51	1.00	0.121	0.096	0.187	0.233
52	1.02	0.155	0.101	0.213	0.289
53	1.04	0.187	0.108	0.240	0.346
54	1.06	0.214	0.118	0.268	0.404
55	1.08	0.191	0.132	0.299	0.463
56	1.10	0.175	0.149	0.332	0.519
57	1.12	0.117	0.166	0.368	0.572
58	1.14	0.117	0.182	0.405	0.617
59	1.16	0.018	0.195	0.441	0.655
60	1.18	-0.151	0.204	0.469	0.682
61	1.20	-0.230	0.205	0.484	0.695
62	1.22	-0.176	0.195	0.483	0.692
63	1.24	-0.142	0.178	0.466	0.676
64	1.26	-0.073	0.159	0.434	0.648
65	1.28	-0.017	0.141	0.388	0.611
66	1.30	0.039	0.125	0.333	0.563
67	1.32	0.090	0.111	0.277	0.505
68	1.34	0.146	0.098	0.226	0.437
69	1.36	0.208	0.086	0.182	0.359
70	1.38	0.291	0.076	0.146	0.276
71	1.40	0.357	0.070	0.122	0.192
72	1.42	0.447	0.071	0.111	0.114
73	1.44	0.424	0.080	0.112	0.051
74	1.46	0.338	0.096	0.119	0.007
75	1.48	0.274	0.114	0.128	-0.017
76	1.50	0.261	0.128	0.139	-0.022
77	1.52	0.271	0.137	0.150	-0.006
78	1.54	0.246	0.138	0.161	0.027
79	1.56	0.264	0.134	0.169	0.076
80	1.58	0.291	0.125	0.177	0.140
81	1.60	0.354	0.113	0.192	0.217
82	1.62	0.096	0.103	0.214	0.304
83	1.64	-0.432	0.091	0.234	0.389
84	1.66	-0.605	0.066	0.239	0.454
85	1.68	-0.582	0.026	0.224	0.490

Table D-5 Displacement Responses (Continued)

i	t_i	$\ddot{u}_g(t)$	$(\delta_y)_{11}$	$(\delta_y)_{19}$	$(\delta_y)_{27}$
86	1.70	-0.595	-0.020	0.192	0.492
87	1.72	-0.531	-0.063	0.139	0.459
88	1.74	-0.505	-0.097	0.063	0.392
89	1.76	-0.513	-0.125	-0.028	0.293
90	1.78	-0.513	-0.147	-0.121	0.162
91	1.80	-0.528	-0.166	-0.210	-0.002
92	1.82	-0.477	-0.187	-0.302	-0.198
93	1.84	-0.394	-0.212	-0.400	-0.424
94	1.86	-0.318	-0.239	-0.502	-0.668
95	1.88	-0.229	-0.265	-0.600	-0.918
96	1.90	-0.126	-0.292	-0.695	-1.156
97	1.92	-0.005	-0.320	-0.790	-1.373
98	1.94	0.105	-0.342	-0.884	-1.552
99	1.96	0.230	-0.358	-0.968	-1.689
100	1.98	0.341	-0.368	-1.036	-1.782
101	2.00	0.468	-0.375	-1.086	-1.828
102	2.02	0.574	-0.375	-1.111	-1.827
103	2.04	0.706	-0.367	-1.101	-1.776
104	2.06	0.799	-0.354	-1.043	-1.671
105	2.08	0.888	-0.328	-0.933	-1.511
106	2.10	0.936	-0.281	-0.780	-1.295
107	2.12	1.000	-0.205	-0.592	-1.025
108	2.14	0.826	-0.103	-0.373	-0.705
109	2.16	0.680	0.018	-0.123	-0.342
110	2.18	-0.351	0.144	0.147	0.050
111	2.20	-0.694	0.244	0.406	0.442
112	2.22	-0.480	0.302	0.639	0.816
113	2.24	-0.546	0.329	0.842	1.167
114	2.26	-0.320	0.347	1.008	1.488
115	2.28	-0.220	0.371	1.128	1.778
116	2.30	-0.051	0.402	1.207	2.035
117	2.32	0.033	0.444	1.269	2.256
118	2.34	0.156	0.498	1.335	2.433
119	2.36	0.262	0.558	1.410	2.558
120	2.38	0.347	0.614	1.487	2.622

Table D-5 Displacement Responses (Continued)

i	t_i	$\ddot{u}_g(t)$	$(\delta_y)_{11}$	$(\delta_y)_{19}$	$(\delta_y)_{27}$
121	2.40	0.514	0.671	1.566	2.624
122	2.42	0.169	0.730	1.650	2.573
123	2.44	-0.770	0.770	1.721	2.474
124	2.46	-0.453	0.767	1.747	2.326
125	2.48	-0.506	0.728	1.715	2.151
126	2.50	-0.296	0.671	1.618	1.961
127	2.52	-0.169	0.607	1.454	1.763
128	2.54	0.069	0.534	1.227	1.558
129	2.56	-0.196	0.454	0.955	1.343
130	2.58	-0.579	0.353	0.661	1.102
131	2.60	-0.480	0.209	0.365	0.820
132	2.62	-0.493	0.020	0.079	0.490
133	2.64	-0.433	-0.189	-0.199	0.112
134	2.66	-0.360	-0.385	-0.478	-0.308
135	2.68	-0.293	-0.548	-0.766	-0.756
136	2.70	-0.220	-0.671	-1.056	-1.214
137	2.72	-0.153	-0.742	-1.333	-1.665
138	2.74	-0.079	-0.765	-1.586	-2.095
139	2.76	-0.013	-0.765	-1.806	-2.501
140	2.78	0.055	-0.761	-1.986	-2.883
141	2.80	-0.028	-0.761	-2.115	-3.237
142	2.82	-0.127	-0.773	-2.200	-3.559
143	2.84	-0.245	-0.812	-2.259	-3.831
144	2.86	-0.278	-0.876	-2.304	-4.033
145	2.88	-0.210	-0.946	-2.336	-4.142
146	2.90	-0.175	-1.005	-2.361	-4.144
147	2.92	-0.098	-1.045	-2.391	-4.041
148	2.94	-0.032	-1.058	-2.427	-3.845
149	2.96	0.054	-1.038	-2.447	-3.581
150	2.98	0.123	-0.995	-2.420	-3.271
151	3.00	0.197	-0.950	-2.321	-2.935
152	3.02	-0.028	-0.905	-2.138	-2.587
153	3.04	-0.109	-0.852	-1.876	-2.239
154	3.06	-0.012	-0.782	-1.553	-1.892
155	3.08	0.003	-0.684	-1.192	-1.540

Table D-5 Displacement Responses (Continued)

i	t_i	$\ddot{u}_g(t)$	$(\delta_y)_{11}$	$(\delta_y)_{19}$	$(\delta_y)_{27}$
156	3.10	0.101	-0.535	-0.821	-1.173
157	3.12	0.165	-0.322	-0.460	-0.777
158	3.14	0.258	-0.065	-0.111	-0.346
159	3.16	0.331	0.200	0.237	0.121
160	3.18	0.399	0.452	0.602	0.613
161	3.20	0.064	0.671	0.978	1.114
162	3.22	0.071	0.828	1.343	1.600
163	3.24	0.200	0.914	1.682	2.063
164	3.26	0.202	0.954	1.993	2.507
165	3.28	0.386	0.976	2.266	2.942
166	3.30	0.396	0.993	2.484	3.381
167	3.32	0.597	1.020	2.644	3.820
168	3.34	-0.272	1.072	2.759	4.242
169	3.36	-0.383	1.129	2.829	4.598
170	3.38	-0.203	1.162	2.853	4.847
171	3.40	-0.160	1.164	2.841	4.967
172	3.42	0.021	1.147	2.812	4.949
173	3.44	0.198	1.116	2.783	4.806
174	3.46	-0.312	1.064	2.751	4.560
175	3.48	-0.435	0.995	2.690	4.224
176	3.50	-0.313	0.933	2.572	3.814
177	3.52	-0.340	0.887	2.377	3.348
178	3.54	-0.223	0.843	2.094	2.836
179	3.56	-0.164	0.791	1.737	2.291
180	3.58	-0.063	0.721	1.334	1.721
181	3.60	-0.037	0.605	0.914	1.136
182	3.62	-0.197	0.418	0.494	0.540
183	3.64	-0.095	0.170	0.082	-0.068
184	3.66	-0.099	-0.106	-0.329	-0.681
185	3.68	-0.032	-0.384	-0.757	-1.288
186	3.70	0.005	-0.644	-1.206	-1.869
187	3.72	0.088	-0.859	-1.656	-2.407
188	3.74	0.143	-1.015	-2.078	-2.887
189	3.76	0.178	-1.126	-2.454	-3.309
190	3.78	0.065	-1.207	-2.769	-3.686

APPENDEX E

MODAL ANALYSES OF A TWENTY-STORY PLANE FRAME

9.5a

APPENDIX E

MODAL ANALYSES OF A TWENTY-STORY PLANE FRAME

This appendix records the data and results of modal analyses made of a twenty-story plane frame. The plane frame is shown in Figure E-1; the section properties are given in Table E-1. The two sets of system natural frequencies calculated by CMS and FEM programs, respectively, are compared in Table E-2. The agreement is good for lower modes.

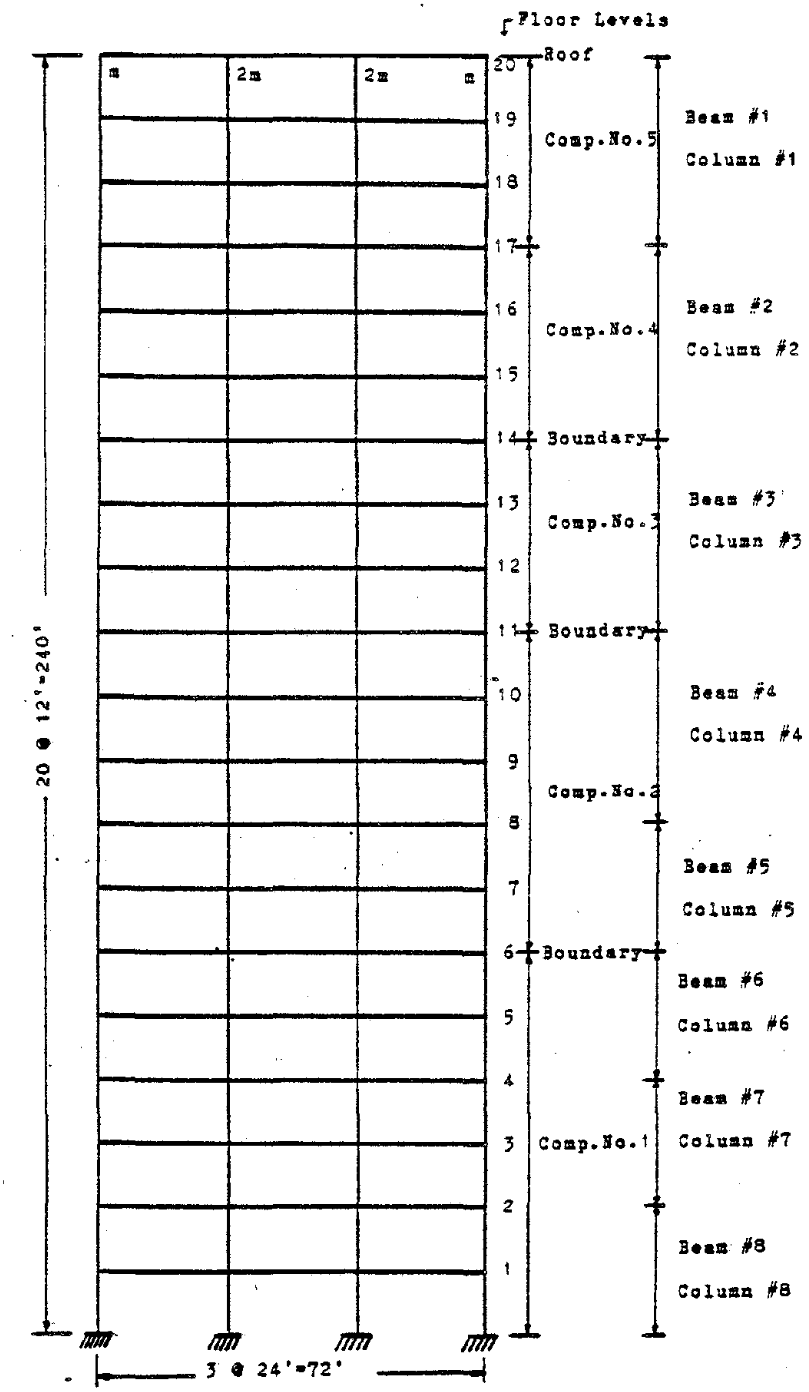


Figure E-1 Front View of A 2-D Frame

Table E-1 Section Properties

	No.	A	A_w	I_x
Beam No.	1	8.83	3.21	290.
	2	10.0	3.45	340.
	3	10.6	4.13	447.
	4	11.8	4.28	517.
	5	13.3	4.83	584.
	6	14.7	5.24	657.
	7	14.7	5.72	802.
	8	16.2	6.16	891.
Column No.	1	9.12	1.77	37.
	2	17.7	3.28	116.
	3	25.0	4.75	235.
	4	31.2	5.89	301.
	5	37.3	6.89	528.
	6	41.8	7.68	660.
	7	46.5	8.25	745.
	8	51.7	9.27	838.

E = 29000. ksi

Table E-2 Calculated Natural Frequencies, CPS

Mode Number	Finite Element Method (ETABS79)	CMS
1	0.202	0.206
2	0.489	0.501
3	0.801	0.822
4	1.135	1.161
5	1.490	1.528
6	1.760	1.807
7	2.148	2.209
8	2.407	2.484
9	2.711	2.796
10	3.155	3.252
11	3.685	3.803
12	4.166	3.972
13	4.614	4.297
14	5.267	4.315
15	5.927	4.392
16	6.651	4.785
17	7.297	4.861
18	8.483	5.016
19	9.889	5.463
20	11.295	5.780

APPENDIX F

COMPUTER PROGRAM "SUPERDYNE"

APPENDIX F

COMPUTER PROGRAM "SUPERDYNE"

```
C SFEALL.FOR      83-02-24        J. T. HUANG
C                                 DEPT. OF CIVIL ENGR
C                                 UNIV. OF PITTSBURGH
C
C MAIN.FOR  83-02-7               (OK, 83-01-01)  JTH
C
C *  MAIN.FOR,  *SMPROP.FOR, * SCMID.FOR, *CRDMAS.FOR
C *COMSTF.FOR,  *  GUY4.FOR, *  CMS3.FOR, *   ELM.FOR
C * EPCCC.FOR,  *  SYSK.FOR, *  SYSM.FOR, *  SYSP.FOR
C * EPAAA.FOR,  *DISPL1.FOR, *FRESP2.FOR, * EPOST.FOR
C                                                        *=OK
C
C
      COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1             ICBS(168,4),ICNS(36,4),NSYS(4)
      COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1             NDCP(4),IFB(4),IFT(4),NCXS(7)
      COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
C
      COMMON/EGV1/IA2(600)
      COMMON/SPROP/SP(20,5)
      COMMON/MPROP/MP(4,3)
      COMMON/COORD/X(182),Y(182),Z(182)
      COMMON/CESTM/SK(144)
      COMMON/IDO/IDO(600)
      COMMON/GARB/C(600)
C
      DIMENSION AA(57000)
C
C  CORE =38K+AA
C-----------------------------------------------------------------
C DEVICES FOR MY JOB
C CDR   1
C LPT   6            RUN RESULTS IF IP=6
C DSK   2,3, 8-63    DATA FILES
C DSK   4            RUN RESULTS IF IP=4
C-----------------------------------------------------------------
      OPEN (UNIT=4,FILE='RUN.DAT',ACCESS='SEQOUT')
C
      READ  ( 5,2000) ICHK,IPX,IPM,IP,ITRAN
      READ  ( 5,2000) NDOF,NCOMP,NCNODX
      READ  ( 5,2000) NMOD,NO,NO2
      READ  ( 5,2010) SK(1),SK(2),SK(3)
C
```

```
      WRITE (IP,2000) ICHK,IPX,IPM,IP,ITRAN
      WRITE (IP,2000) NDOF,NCOMP,NCNODX
      WRITE (IP,2000) NMOD,NO,NO2
      WRITE (IP,2010) SK(1),SK(2),SK(3)
      IF (IP.NE.4) IP=6
      KX=1
      NCPO=NCOMP
      NCDMX=NDOF*NCNODX
      IDO(1)=0
      DO 10 J=2,600
   10 IDO(J)=IDO(J-1)+J-1
      IF (ICHK.EQ.1) WRITE (IP,2000) (J,IDO(J),J=1,5)
C
      CALL SMPROP (IP)
      N4=NCDMX
      N1=NCOMP*NCDMX+1
      CALL SCMIDX (IP, IPX,N4,NCPO,AA(1),AA(N1))
      N3=NCXS(4)
      N2=N1+NCOMP*NCDMX
      CALL CRDMAS (IP, IPM,N4,NCPO,N3,AA(1),AA(N1),AA(N2))
C
      N5=IDO(N4)+N4
      WRITE (IP,2050) N4,N5
      CALL GSTIF6 (IP,ICHK,N5,AA)
      WRITE (IP,2020) N5
C
      N1=NSYS(4)
      N2=IDO(N1)+N1
      N4=NSYS(2)
      WRITE (IP,2020) N2
      WRITE (IP,2040) N4,N1,N2
      IF (ICHK.EQ.1) GO TO 20
      CALL SYSP   (AA,N1,NCPO,IP)
      CALL SYSK   (AA,N2,NCPO,IP)
      ISOL=3
      ICOMP=0
      ISCON=0
      CALL STCOND (AA,N5,N4,N1,N2,ISCON,ISOL,ICOMP,IP)
      CALL SYSM   (AA,N2,NCPO,IP)
      CALL GUYRED (AA,N4,N1,N2,IP)
C
   20 N5=IDO(N4)+N4
      N2=1+N5
      N3=N2+N5
      N44=N4*N4
      N6=N3+N44
      N7=N6+N4
      NCPO=N7-N3
      N30=N3-1
```

```
      WRITE (IP,2020) N7
      WRITE (6,2030) N5 ,N2,N3
      WRITE (6,2030) N5 ,KX,N2
      WRITE (6,2030) N44,N3,N6
      WRITE (6,2030) N4 ,N6,N7
      IF (ICHK.EQ.1) GO TO 30
      CALL GEVPS2 (AA(N2),AA(1),AA(N3),AA(N6),N4,N5,NMOD,ICOMP,IP)
C
   30 N2=1+N44
      N3=N2+N4
      N40=N3+N4
      N5=N40+NO
      N6=N5+NO
      N7=N6+NO
      N8=N7+NO2
      N88=NO2*NMOD
      N9=N8+N88
      N10=N9+NMOD+12
      WRITE (IP,2020) N10
      WRITE (6,2030) N44 ,KX ,N2
      WRITE (6,2030) N4  ,N2 ,N3
      WRITE (6,2030) N4  ,N3 ,N40
      WRITE (6,2030) NO  ,N40,N5
      WRITE (6,2030) NO  ,N5 ,N6
      WRITE (6,2030) NO  ,N6 ,N7
      WRITE (6,2030) NO2 ,N7 ,N8
      WRITE (6,2030) N88 ,N8 ,N9
      WRITE (6,2030) NMOD,N9 ,N10
      IF (ICHK.EQ.1 .OR. ITRAN.EQ.0) GO TO 50
      DO 40 I=1,NCPO
   40 AA(I)=AA(N30+I)
      CALL DISPL1 (AA( 1),AA(N2),AA(N3),AA(N40),AA(N5),AA(N6),
     1             AA(N7),AA(N8),AA(N9),  N4,NO,I12,NO2,
     2             DT,NMOD,FMAX,EMIN,IP,ICHK)
C
   50    CONTINUE
         STOP
 2000    FORMAT ( 6I5)
 2010    FORMAT (6F10.5)
 2020    FORMAT (/,80X,'*** NEED AA(',I6,') OR LARGER ***',/)
 2030 FORMAT (83X,6I7)
 2040 FORMAT (80X,'### SYS  ',3I7)
 2050 FORMAT (80X,'### COMP ',3I7)
         END
C
         SUBROUTINE PMATE(A,M1,N1,M2,N2)
         DIMENSION A(M2,N2)
      WRITE (6,2020)
      KO=(N2-N1)/12+1
```

```
      DO 20 K=1,KO
      I1=(K-1)*12+N1
      I2=I1+11
      IF (I2.GT.N2) I2=N2
      WRITE (6,2000) (J,J=I1,I2)
      DO 10 I=M1,M2
   10 WRITE (6,2010) I,(A(I,J),J=I1,I2)
   20 CONTINUE
      WRITE (6,2020)
         RETURN
 2000 FORMAT (//8X,12I10//)
 2010 FORMAT (5X,I3,12E10.4)
 2020 FORMAT (/)
         END
C
C SMPROP.FOR  83-01-03  ( OK,82-12-07,82-10-21)  82-09-14 . JTH
C
         SUBROUTINE SMPROP (IP)
         COMMON/SPROP/SP(20,5)
         COMMON/MPROP/P(4,3)
C
C OK                I=SECTION TYPE NO.
C OK                J=(1,2,3,4,5)  SP(I,J)=(A,IY,IZ,XJ,TH)
C OK                J=(1,2)        SP(I,J)=(A,IZZ)
C
        WRITE (IP,2010)
        READ  ( 5,2000) NA
        WRITE (IP,2000) NA
        CALL ZERO (SP(1,1),SP(NA,5))
      DO 10 I=1,NA
        READ  ( 5,2000) ISP,  (SP(ISP,J),J=1,5)
   10   WRITE (IP,2000) ISP,  (SP(ISP,J),J=1,5)
C
C OK            I=MAT TYPE NO.
C OK            J=(1,2,3),  P(I,J)=(E,EG,PR)
C
        WRITE (IP,2020)
        READ  ( 5,2000) NA
        WRITE (IP,2000) NA
        CALL ZERO (P(1,1),P(NA,3))
      DO 20 I=1,NA
        READ  ( 5,2000) IM,  (P(IM,J),J=1,3)
   20   WRITE (IP,2000) IM,  (P(IM,J),J=1,3)
C
         RETURN
 2000 FORMAT (I5,5F15.7)
 2010 FORMAT (/,/,5X,'SEC-PROP ...',/)
 2020 FORMAT (/,5X,'MAT-PROP ...',/)
         END
```

```
C SCMID.FOR  83-01-27 (OK,11-30,82-09-22) JTH 82-09-10
C
      SUBROUTINE SCMIDX (IP,ICHK,NCDMXO,NCOMPO,ICD2,IXDSB)
      COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1             ICBS(168,4),ICNS(36,4),NSYS(4)
      COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1             NDCP(4),IFB(4),IFT(4),NCXS(7)
      COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
      COMMON/GARB/IDOF(6),ICT(8),ICP2(8),AAO(560),NDXX(9),NFXX(9)
      DIMENSION ICD2(NCDMXO,NCOMPO), IXDSB(NCDMXO,NCOMPO)
C
C       FLOOR & SYS  DATA
C
     WRITE (IP,2000)
     READ  ( 5,2040) NFL,NDICPF
     WRITE (IP,2040) NFL,NDICPF
C
   DO 10 I=1,NFL
     READ  ( 5,2040) IFL,NDL(IFL,1),NDL(IFL,2),NDL(IFL,3)
10   WRITE (IP,2040) IFL,NDL(IFL,1),NDL(IFL,2),NDL(IFL,3)
   DO 20 I=1,NFL
   NDL(I,4)=NDL(I,2)-NDL(I,1)+1
20 NDL(I,2)=NDL(I,4)
   NAO=NDL(1,2)
   NDL(1,2)=1
   DO 30 I=1,NFL-1
   NBO=NDL(I+1,2)
   NDL(I+1,2)=NDL(I,2)+NAO
30 NAO=NBO
   NDD=3
   NDT=3
   IF (NDOF.EQ.6) GO TO 40
   NDD=2
   NDT=1
40 DO 50 I=1,NCOMP
   DO 50 J=1,3
   DO 50 K=1,2
50 NCB(I,J,K)=0
   NCDMX=0
C
C          READ & GEN COMP INDICIES
C          COMP NO. =(1,NCOMP)=(LOWEST,HIGHEST)
C
   NCNDF=0
   NCBDF=0
   DO 360 ICO=1,NCOMP
     READ  (5,2040) ICOMP,IFLB,IFLT,NCRMO(ICOMP)
     WRITE (IP,2200) ICOMP
```

```
        WRITE (IP,2040) ICOMP,IFLB,IFLT,NCRMO(ICOMP)
      IF ( NCRMO(ICOMP).GT.NCNDF) NCNDF=NCRMO(ICOMP)
      IFB(ICOMP)=IFLB
      IFT(ICOMP)=IFLT
      NPT=0
      DO 60 I=IFLB,IFLT
   60 NPT=NPT+NDL(I,4)
      MDOF=NDOF*NPT
      NDCP(ICOMP)=NDL(IFLB,2)-1
      DO 90 I=1,NPT
      NO=(I-1)*NDOF
      DO 70 J=1,NDD
   70 IC(NO+J,ICOMP)=-5
      DO 80 J=1,NDT
   80 IC(NO+NDD+J,ICOMP)=-6
   90 CONTINUE
C
C         READ RETAINED DOF;  IDOF(I)=4
C
        WRITE (IP,2110)
  100   READ  ( 5,2050) NDO,  IFL,  IG1,IC1,  (IDOF(I),I=1,NDD)
        WRITE (IP,2050) NDO,  IFL,  IG1,IC1,  (IDOF(I),I=1,NDD)
        IF (NDO.EQ.0) GO TO 130
      ICO=NDL(IFL,2)-NDL(IFL,1)-NDCP(ICOMP)
      DO 120 I=1,IG1
      NA=(I-1)*IC1
      NDA=NDO+NA
      NDLC=NDA+ICO
      NO=(NDLC-1)*NDOF
      DO 110 K=1,NDD
        IF (IDOF(K).NE.4) GO TO 110
      IC(NO+K,ICOMP)=-IDOF(K)
  110 CONTINUE
  120 CONTINUE
      GO TO 100
C
C         READ FIXED DOF;  IDOF(I)=9
C
  130   WRITE (IP,2120)
  140   READ  ( 5,2060) NDO, IFL, IG1,IC1, (IDOF(I),I=1,NDOF)
        WRITE (IP,2060) NDO, IFL, IG1,IC1, (IDOF(I),I=1,NDOF)
        IF (NDO.EQ.0) GO TO 170
      ICO=NDL(IFL,2)-NDL(IFL,1)-NDCP(ICOMP)
      DO 160 I=1,IG1
      NA=(I-1)*IC1
      NDA=NDO+NA
      NDLC=NDA+ICO
      NO=(NDLC-1)*NDOF
      DO 150 K=1,NDOF
```

```
      IF (IDOF(K).NE.9) GO TO 150
      IC(NO+K,ICOMP)=-IDOF(K)
  150 CONTINUE
  160 CONTINUE
        GO TO 140
C
C          READ  BOUND PTS
C          IBOT=(1,2)=(BOT,TOP)
C
  170    WRITE (IP,2130)
  180    READ  ( 5,2070) NDO, IFL,IG1,IC1,  IBOT,IROOF
         WRITE (IP,2070) NDO, IFL,IG1,IC1,  IBOT,IROOF
         IF (NDO.EQ.0) GO TO 220
      IO3=-3
      IO2=-2
      IF (IROOF.EQ.1) IO3=-6
      IF (IROOF.EQ.1) IO2=-5
      ICO=NDL(IFL,2)-NDL(IFL,1)-NDCP(ICOMP)
      DO 210 I=1,IG1
      NA=(I-1)*IC1
      NDA=NDO+NA
      NDLC=NDA+ICO
      NO=(NDLC-1)*NDOF
      DO 190 K=1,NDT
      IF (IROOF.EQ.1) GO TO 190
      NCB(ICOMP,3,IBOT)=NCB(ICOMP,3,IBOT)+1
  190 IC(NO+NDD+K,ICOMP)=IO3
      DO 200 K=1,NDD
      IF (IROOF.EQ.1) GO  TO 200
      NCB(ICOMP,2,IBOT)=NCB(ICOMP,2,IBOT)+1
  200 IC(NO+K,ICOMP)=IO2
  210 CONTINUE
      GO TO 180
C
C          READ RETAINED BOUNDARY DOF; IDOF(I)=1
C
  220 WRITE (IP,2140)
  230  READ  ( 5,2050) NDO, IFL, IG1,IC1, (IDOF(I),I=1,NDD), IBOT,IROOF
       WRITE (IP,2050) NDO, IFL, IG1,IC1, (IDOF(I),I=1,NDD), IBOT,IROOF
       IF (NDO.EQ.9993) GO TO 260
      ICO=NDL(IFL,2)-NDL(IFL,1)-NDCP(ICOMP)
      DO 250 I=1,IG1
      NA=(I-1)*IC1
      NDA=NDO+NA
      NDLC=NDA+ICO
      NO=(NDLC-1)*NDOF
      DO 240 K=1,NDD
      ICCC=IDOF(K)
      IF(ICCC.NE.1) GO TO 240
```

```
      NCB(ICOMP,ICCC,IBOT)=NCB(ICOMP,ICCC,IBOT)+1
      IC(NO+K,ICOMP)=-ICCC
      IF (IROOF.EQ.1) GO TO 240
      NCB(ICOMP,2,IBOT)=NCB(ICOMP,2,IBOT)-1
  240 CONTINUE
  250 CONTINUE
      GO TO 230
C
C        ASSIGN DOF SEQ
C
  260 ICT(1)=-1
      ICT(2)=-2
      ICT(3)=-3
      ICT(4)=-4
      ICT(5)=-5
      ICT(6)=-6
      ICT(7)=-9
      NNN=0
      DO 270 J=1,3
      DO 270 K=1,2
  270 NNN=NNN+NCB(ICOMP,J,K)
      IF (NNN.GT.NCBDF) NCBDF=NNN
C
C        NO. OF DOF FOR EACH DOF TYPE
C
      KOO=0
      DO 280 J=1,7
      JOO=ICT(J)
      NDC1(J,ICOMP)=0
      DO 280 I=1,MDOF
      IO1=IC(I,ICOMP)
      IF (IO1.NE.JOO) GO TO 280
      NDC1(J,ICOMP)=NDC1(J,ICOMP)+1
      KOO=KOO+1
      ICD2(I,ICOMP)=-IC(I,ICOMP)
      IC(I,ICOMP)=KOO
  280 CONTINUE
      IF (ICHK.NE.1) GO TO 330
      WRITE (IP,2100)
      DO 320 I=IFLB,IFLT
      NEA=NDL(I,1)
      NEB=NDL(I,4)+NEA-1
      ICO=NDL(I,2)-NDL(I,1)-NDCP(ICOMP)
      DO 320 J=NEA,NEB
      NDLC=J+ICO
      NO=(NDLC-1)*NDOF
      DO 290 K=1,NDOF
  290 ICP2(K)=NO+K
      IF (I.NE.IFLB .AND. I.NE.IFLT) GO TO 310
```

```
      JXY=J*10
      DO 300 K=1,NDOF
      KCX=ICP2(K)
      KCY=IC(KCX,ICOMP)
  300 IXDSB(KCY,ICOMP)=JXY+K
  310 KA=ICP2(1)
      KB=ICP2(NDOF)
      WRITE (IP,2090) I,J,(ICP2(K),K=1,NDOF),(ICD2(K,ICOMP),
     1  K=KA,KB),(IC(K,ICOMP),K=KA,KB)
  320 CONTINUE
C
C         STATISTICS FOR  EACH DOF TYPE
C
  330 NDC2(1,ICOMP)=1
      NDC3(1,ICOMP)=NDC1(1,ICOMP)
      DO 340 I=2,7
      NDC3(I,ICOMP)=NDC3(I-1,ICOMP)+NDC1(I,ICOMP)
      NDC2(I,ICOMP)=NDC3(I,ICOMP)-NDC1(I,ICOMP)+1
      IF (NDC2(I,ICOMP).GT.NDC3(I,ICOMP)) NDC2(I,ICOMP)=NDC3(I,ICOMP)
  340 CONTINUE
        NNN=NDC3(7,ICOMP)
        IF (NNN.GT.NCDMX) NCDMX=NNN
        IF (NNN.EQ.MDOF ) GO TO 350
        WRITE (IP,2030) NNN,MDOF
        STOP
  350 CONTINUE
  360 CONTINUE
C
C            STATISTICS FOR COMP INDICES
C
      WRITE (IP,2210)
      WRITE (IP,2220) NCDMX,NCBDF,NCNDF
      DO 380 I=1,7
      NCXO(I)=0
      NCXS(I)=0
      DO 370 J=1,NCOMP
      IF (NCXO(I).LT.NDC1(I,J)) NCXO(I)=NDC1(I,J)
      IF (NCXS(I).LT.NDC3(I,J)) NCXS(I)=NDC3(I,J)
  370 CONTINUE
  380 CONTINUE
C
C          SYS INDICES
C
      KOO=0
      DO 410 IOO=1,NCOMP
      DO 390 K=1,NCBDF
  390 ICBS(K,IOO)=0
      DO 400 K=1,NCNDF
  400 ICNS(K,IOO)=0
```

```
      NCCC=NCRMO(IOO)
      ICP2(IOO)=0
      DO 410 J=1,NCCC
      KOO=KOO+1
  410 ICNS(J,IOO)=KOO
        KOOO=KOO
C
      DO 430 J=1,3
      NSYS(J)=KOO
      DO 430 IOO=1,NCOMP
      NNN=NCB(IOO,J,2)
      ICP2(IOO)=ICP2(IOO)+NCB(IOO,J,1)
      IF (NNN.EQ.0) GO TO 430
      DO 420 K=1,NNN
      ICP2(IOO)=ICP2(IOO)+1
      NCCC=ICP2(IOO)
      KOO=KOO+1
      ICBS(NCCC,IOO)=KOO
  420 CONTINUE
  430 CONTINUE
        NSYS(4)=KOO
        DO 440 I=1,NCOMP
  440   ICP2(I)=0
        KOO=KOOO
      DO 470 J=1,3
      DO 470 IOO=1,NCOMP
      NNN=NCB(IOO,J,1)
      IF (NNN.EQ.0) GO TO 460
      DO 450 K=1,NNN
      KOO=KOO+1
      ICP2(IOO)=ICP2(IOO)+1
      NCCC=ICP2(IOO)
      ICBS(NCCC,IOO)=KOO
  450 CONTINUE
      IF (IOO.NE.NCOMP) GO TO 460
      KOO=KOO+NCB(IOO,J,2)
  460 ICP2(IOO)=ICP2(IOO)+NCB(IOO,J,2)
  470 CONTINUE
C
      IF(ICHK.NE.1) GO TO 530
      IF (NCNDF.EQ.0) GO TO 490
      WRITE (IP,2160)
      DO 480 I=1,NCNDF
  480 WRITE (IP,2180) I,(IOO,ICNS(I,IOO),IOO=1,NCOMP)
  490 CONTINUE
      IF (NCBDF.EQ.0) GO TO 530
      WRITE (IP,2170)
      DO 520 I=1,NCBDF
      DO 510 ICOMP=1,NCOMP
```

```
      NBBB=NDC3(3,ICOMP)
      IF (I.GT.NBBB) GO TO 500
      KKK=IXDSB(I,ICOMP)
      NDXX(ICOMP)=KKK/10
      NFXX(ICOMP)=KKK-KKK/10*10
      GO TO 510
  500 NDXX(ICOMP)=0
      NFXX(ICOMP)=0
  510 CONTINUE
  520 WRITE (IP,2190) I,(IO,ICBS(I,IO),NDXX(IO),NFXX(IO),IO=1,NCOMP)
  530   WRITE (IP,2010)
        RETURN
 2000 FORMAT (1H1,5X,'ETR SCMID',//)
 2010 FORMAT (    5X,'END SCMID',//)
 2020 FORMAT (//)
 2030 FORMAT (50X,'DOF COUNT MAY BE WRONG',2I3,/)
 2040 FORMAT (10I5)
 2050 FORMAT (4I5,4X,3I1,3X,2I5)
 2060 FORMAT (4I5,4X,6I1,2I5)
 2070 FORMAT (4I5,4X,6X,2I5)
 2080 FORMAT (//,20X,'ICOMP=',5I5)
 2090 FORMAT (2X,I3,1X,I4,1X,1X,6('(',3I4,')'))
 2100 FORMAT (//,' I-FL, EXT-NODE NO.,  DOF-SEQ
     1 BY NODE, DOF-TYPE BY NODE, DOF-SEQ FOR COMP ',/)
 2110 FORMAT (//,2X,'RETAINED INTERIOR DOF ...',/)
 2120 FORMAT (//,2X,'FIXED     DOF ...',/)
 2130 FORMAT (//,2X,'BOUNDARY DOF ...',/)
 2140 FORMAT (//,2X,'RETAINED BOUNDARY DOF ...',/)
 2150 FORMAT (//,2X,'COMP. DOF SEQ  ...',/)
 2160 FORMAT (//,2X,'COMP NORMAL MODE DOF VS. SYS DOF',/)
 2170 FORMAT (/,2X,'M=(ICOMP,J)   ... THE M-TH VAR OF ICOMP-TH COMP
     1  IS ASSEMBLED TO THE J-TH VAR OF THE SYSTEM',/)
 2180 FORMAT (2X,I4,2X,12('(',I2,',',I4,')',1X))
 2190 FORMAT (2X,I4,2X,6( '(',I2,',',I4,';',I4,'-',I1,')',1X  ))
 2200 FORMAT (//,80X,'COMP. NO.',I3,' ...',/)
 2210 FORMAT (//,80X,'SYS DATA ...',/)
 2220 FORMAT (5X,'MAX NO. OF DOF IN ANY COMP.    ... ',
     1 /,5X,
     2 '  ALL-DOF=',I4,'  B-DOF= ',I4,'  N-DOF= ',I4,/)
        END
C
C DFLOC1.FOR    OK,82-05-26 JTH
C
        SUBROUTINE DFLOC1 (NDA,IDIR,IDF2,ICOMP)
        COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1                NDCP(4),IFB(4),IFT(4),NCXS(7)
        COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
      IFL=NDA/NDICPF+1
      NDS=NDA-NDL(IFL,1)+NDL(IFL,2)-NDCP(ICOMP)
```

```
      IDFLC=(NDS-1)*NDOF+IDIR
      IDF2=IC(IDFLC,ICOMP)
         RETURN
         END
         SUBROUTINE NDLOC1 (NDA,NDS,ICOMP)
         COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1                  NDCP(4),IFB(4),IFT(4),NCXS(7)
         COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
      IFL=NDA/NDICPF+1
      NDS=NDA-NDL(IFL,1)+NDL(IFL,2)-NDCP(ICOMP)
         RETURN
         END
         SUBROUTINE NDLOS (NDA,NDS)
         COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1                  NDCP(4),IFB(4),IFT(4),NCXS(7)
         COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
      IFL=NDA/NDICPF+1
      NDS=NDA-NDL(IFL,1)+NDL(IFL,2)
         RETURN
         END
         SUBROUTINE DFLOC2 (NDS,IDIR,IDF2,ICOMP)
         COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1                  NDCP(4),IFB(4),IFT(4),NCXS(7)
         COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
      IDFLC=(NDS-1)*NDOF+IDIR
      IDF2=IC(IDFLC,ICOMP)
         RETURN
         END
C
C CRDMAS.FOR   83-02-6 ( OK, 82-11-30,10-21) JTH 82-09-14
C
         SUBROUTINE CRDMAS (IP,ICHK,NCDMXO,NCOMPO,NC4X,ICD2,XMS,PEO)
         DOUBLE PRECISION DMI,DMB,PE1
         COMMON/COORD/X(182),Y(182),Z(182)
         COMMON/CESTM/AO(144)
         COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1                  ICBS(168,4),ICNS(36,4),NSYS(4)
         COMMON/FEIDC/NDL(14,4),IC(336,4),NCB(4,3,2),
     1                  NDCP(4),IFB(4),IFT(4),NCXS(7)
         COMMON/FEIDX/NDOF,NDD,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
         COMMON/GARB/XO(16),YO(16),ZO(28),SMX(3,4),XM3(3),ICP2(3),
     1                ICP1(4),XM(200),XM4(4),P13(3),AAAO(311)
         DIMENSION ICD2(NCDMXO,NCOMPO),XMS(NCDMXO,NCOMPO),
     1               PEO(NC4X,NCOMPO)
C
C          READ & GEN COORD
C
         WRITE (IP,2180)
        READ  ( 5,2070) NX,NY,NZ
```

```
      WRITE (IP,2070) NX,NY,NZ
    DO 10 I=1,NX
      READ  ( 5,2080) J,XO(J)
 10   WRITE (IP,2080) J,XO(J)
    DO 20 I=1,NY
      READ  ( 5,2080) J,YO(J)
 20   WRITE (IP,2080) J,YO(J)
    DO 30 I=1,NZ
      READ  ( 5,2080) J,ZO(J)
 30   WRITE (IP,2080) J,ZO(J)
      WRITE (IP,2120)
 40   READ  ( 5,2070) NDO,  IGZ,  IDZ,  IXO,IYO,IZO
      WRITE (IP,2070) NDO,  IGZ,  IDZ,  IXO,IYO,IZO
      IF (NDO .EQ. 9994) GO TO 60
    IO=IZO-1
    DO 50 K=1,IGZ
    ND=NDO+(K-1)*IDZ
    CALL NDLOS (ND,NDS)
    IO=IO+1
    X(NDS)=XO(IXO)
    Y(NDS)=YO(IYO)
    Z(NDS)=ZO(IO)
    IF (ICHK.NE.1) GO TO 50
    WRITE (IP,2090) ND,IXO,IYO,IO,X(NDS),Y(NDS),Z(NDS)
 50 CONTINUE
    GO TO 40
C
C          READ & GEN MASS
C
 60 WRITE (IP,2170)
    IW1=0
    IW2=0
    IW3=0
    DO 70 J=1,NCOMP
    SMX(1,J)=0.
    SMX(2,J)=0.
    SMX(3,J)=0.
    NMX=NCXS(4)
    CALL ZERO (PEO(1,J),PEO(NMX,J))
    CALL ZERO (XMS(1,J),XMS(NMX,J))
 70 CONTINUE
 80   READ  ( 5,2070) NXM
      WRITE (IP,2070) NXM
    DO 90 I=1,NXM
      READ  ( 5,2080) J,XM(J)
 90   WRITE (IP,2080) J,XM(J)
       WRITE (IP,2120)
C      READ FOR ALL DOF, ALL COMP TILL TERMINATION
100   READ  ( 5,2070) ICOMP,NDO,IDIR,IM,  IG, ID
```

```
      WRITE (IP,2070) ICOMP,NDO,IDIR,IM,  IG, ID
      IF (ICOMP.EQ.9995) GO TO 120
      TM=XM(IM)
      DO 110 K=1,IG
      ND=NDO+(K-1)*ID
        CALL DFLOC1 (ND,IDIR,IDF,ICOMP)
        NMX=NDC3(4,ICOMP)
        XMS(IDF,ICOMP)=XMS(IDF,ICOMP)+TM
        SMX(IDIR,ICOMP)=SMX(IDIR,ICOMP)+TM
      IF (IDF.LE.NMX) GO TO 110
      IW1=IW1+1
      WRITE (6,2060) ICOMP,NDO,IDIR,IM,K,ND,IDF,NMX
  110 CONTINUE
      GO TO 100
C
C          MASS AND SEISMIC LOAD VECTORS
C
  120 DO 140 J=1,NCOMP
      NB=NDC3(7,J)
      DO 140 I=1,NB
      KO=ICD2(I,J)
      IF (KO.GT.4) GO TO 140
      JO=I-I/NDOF*NDOF
      IF (JO.GT.NDD .OR. JO.EQ.0) GO TO 140
      IO=IC(I,J)
      IF (IO.LE.NDC3(4,J)) GO TO 130
      WRITE (IP,2050) J,I,KO,JO,IO
  130 PEO(IO,J)=XMS(IO,J)*AO(JO)
  140 CONTINUE
C
C          CHECK MASS
C
      IF (ICHK.NE.1) GO TO 190
      WRITE (IP,2110)
      DO 160 J=1,NCOMP
       WRITE(IP,2130) J, (IO,SMX(IO,J),IO=1,NDD)
      K=3
      NA=NDC2(K,J)
      NB=NDC3(K,J)
      IF (NA.EQ.O) GO TO 160
      DO 150 KO=NA,NB
      IF (XMS(KO,J).EQ.O.) GO TO 150
      IW3=IW3+1
      WRITE (6,2060) J,KO,NA,NB
  150 CONTINUE
  160 CONTINUE
      WRITE (IP,2070) IW1,IW3
C
C         MAS  TABLE
```

```
C
      WRITE (IP,2140)
      DO 180 J=1,NCOMP
      DO 180 K=1,4
      IF (K.EQ.3) GO TO 180
      WRITE (IP,2120)
      NA=NDC2(K,J)
      NB=NDC3(K,J)
      IF (NDC1(2,J).EQ.0.) GO TO 180
      DO 170 I=NA,NB
  170 WRITE (IP,2150) J,K,I,  XMS(I,J),PEO(I,J)
  180 CONTINUE
      IF (IW1.LT.1  .AND. IW3.LT.1) GO TO 190
      STOP
C
C         WRITE ON DISK
C
  190 ISOL=2
      DO 200 ICOMP=1,NCOMP
      ENCODE (10,2000,DMI) ISOL,ICOMP
      ENCODE (10,2010,DMB) ISOL,ICOMP
      ENCODE (10,2020,PE1) ISOL,ICOMP
C
         OPEN  (UNIT=2,FILE=DMI,ACCESS='SEQOUT')
      NA=NDC2(4,ICOMP)
      NB=NDC3(4,ICOMP)
      WRITE (2,2160) (XMS(K,ICOMP),K=NA,NB)
         CLOSE (UNIT=2,FILE=DMI)
C
         OPEN  (UNIT=2,FILE=PE1,ACCESS='SEQOUT')
      WRITE (2,2160) (PEO(K,ICOMP),K=1,NB)
         CLOSE (UNIT=2,FILE=PE1)
C
         OPEN  (UNIT=2,FILE=DMB,ACCESS='SEQOUT')
      NA=NDC2(1,ICOMP)
      NB=NDC3(2,ICOMP)
      WRITE (2,2160) (XMS(K,ICOMP),K=NA,NB)
         CLOSE (UNIT=2,FILE=DMB)
  200 CONTINUE
C
      WRITE (IP,2190)
         RETURN
 2000 FORMAT ('DMI',I1,I1,'.DAT',1X)
 2010 FORMAT ('DMB',I1,I1,'.DAT',1X)
 2020 FORMAT ('PE1',I1,I1,'.DAT',1X)
 2030 FORMAT (5X,5I5,F15.7)
 2040 FORMAT (30X,4I5)
 2050 FORMAT (70X,4I5)
 2060 FORMAT (50X,'BAD ... ',8I5)
```

```
 2070 FORMAT (8I5)
 2080 FORMAT (I5,F15.7)
 2090 FORMAT (5X,4I5,3F15.4)
 2100 FORMAT (2X,5I5,2F11.7)
 2110 FORMAT (/,5X,'TOTAL MASS FOR EACH D-DIR OF EACH COMP',/)
 2120 FORMAT (/)
 2130 FORMAT (20X,I3,5X,3(I3,F12.7))
 2140 FORMAT (//,5X,'MASS & FORCE TABLE',/)
 2150 FORMAT (2X,3I5,2X,F10.6,2X,F12.3)
 2160 FORMAT (5X,5E15.8)
 2170 FORMAT (//,5X,' MASS ...',//)
 2180 FORMAT (//,5X,'ETR CRDMAS',/,5X,'COORD.  ...',/)
 2190 FORMAT (//,5X,'END CRDMAS',/)
      END
C COMSTF.FOR 83-02-06 ( OK,82-12-23) JTH 82-05-12
C
         SUBROUTINE GSTIF6 (IP,ICHK,NSDMX,S)
         COMMON/IDO/IDO(600)
         COMMON/CESTM/SKEG(144)
         COMMON/COORD/X(182),Y(182),Z(182)
         COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1                ICBS(168,4),ICNS(36,4),NSYS(4)
         COMMON/FEIDX/NDOF,NDDO,NDT,NFL,NPT,NDICPF,NCDMX,NCXO(7),NCOMP
C
         DIMENSION NDD(2),NDC(2),IE(12)
         DIMENSION S(NSDMX)
C
        WRITE (IP,2060)
        NE=NDOF*2
        IELMS=0
      NBX=0
      DO 10 I=1,NCOMP
      IF (NBX.LT.NDC3(3,I)) NBX=NDC3(3,I)
   10 CONTINUE
      NIX=NCXO(4)
      NX=NBX+NIX
      NCOMPO=NCOMP
   20   READ  ( 5,2000) ICOMP,IPSE,IPSK1,ISCON,N400
        IF (ICOMP.EQ.8888) GO TO 130
      IF (ICHK.EQ.0) ISCON=0
        WRITE (IP,2020) ICOMP,IPSE,IPSK1,ISCON,N400
        IELMC=0
        NSDMXO=NDC3(7,ICOMP)
        NSDMXS=NSDMXO+IDO(NSDMXO)
      IF (IPSE.EQ.0 .AND. IPSK1.EQ.0 .AND. ISCON.EQ.0 .AND.
     1   ICHK.EQ.1) GO TO 30
        CALL ZERO (S(1),S(NSDMXS))
C
C          READ ELMS & CAL STF FOR EACH COMP
```

```
C
   30   READ  ( 5,2000) NI,NJ,  IMP,ISP,IG,ID,IPS
        WRITE (IP,2000) NI,NJ,  IMP,ISP,IG,ID,IPS
        IF (NI.EQ.9996) GO TO 100
C
      ICHKG=IPS*ICHK
      ICHKS=ICHKG*IPSE
      DO 90 K=1,IG
      IF (K.GT.1) ICHKG=0
      IF (K.GT.1) ICHKS=0
      IELMC=IELMC+1
      IELMS=IELMS+1
      KK=(K-1)*ID
      NDD(1)=KK+NI
      NDD(2)=KK+NJ
      CALL NDLOS (NDD(1),NAS)
      CALL NDLOS (NDD(2),NBS)
      IF (ICHK.EQ.1) WRITE (IP,2040) NDD(1),NDD(2),NAS,NBS,IELMC,IELMS
      IF (ICHK.NE.1 .OR. ICHKS.EQ.1 .OR. ISCON .EQ.1) GO TO 40
      GO TO 90
   40 DX=X(NBS)-X(NAS)
      DY=Y(NBS)-Y(NAS)
      IF (NDOF.NE.6) GO TO 50
      DZ=Z(NBS)-Z(NAS)
      CALL BMXYZ (ICHKS,IMP,ISP,DX,DY,DZ,IP,IPS)
      GO TO 60
   50 CALL KE2D6 (ICHKS,IMP,ISP,DX,DY,IP,IPS)
   60 IOO=0
      DO 70 I=1,2
      CALL NDLOC1 (NDD(I),NDC(I),ICOMP)
      DO 70 J=1,NDOF
      IOO=IOO+1
      JOO=J
      CALL DFLOC2 (NDC(I),JOO,IDF2,ICOMP)
      IE(IOO)=IDF2
   70   CONTINUE
      DO 80 J=1,NE
      JO=IE(J)
      KO=IDO(JO)
      IJO=(J-1)*NE
      DO 80 I=1,NE
      IO=IE(I)
      IF (IO.GT. JO) GO TO 80
      IJ=IJO+I
      KOO=KO+IO
      S(KOO)=S(KOO)+SKEG(IJ)
   80   CONTINUE
   90   CONTINUE
        GO TO 30
```

```
C
  100 N2=NDC3(6,ICOMP)
      N1=NDC3(4,ICOMP)
      N2S=IDO(N2)+N2
      N4=NDC3(3,ICOMP)
      ISOL=1
      WRITE (IP,2050) IELMC,N1,N2,N2S
      IF (IPSK1.NE.1) GO TO 110
      WRITE (IP,2030)   (S(IO), IO=1,N2S)
      IF (ISCON.NE.1) GO TO 110
      N4=N400
      GO TO 120
  110 IF (ICHK.EQ.1) GO TO 20
  120    CALL STCOND (S,N4,N1,N2,N2S,ISCON,ISOL,ICOMP,IP)
      N4=NCRMO(ICOMP)
      N1=NDC1(4,ICOMP)
      N2=IDO(N1)+N1
      N3=N2+1
      N6=N3+N1*N1
         CALL STDEGC (S(1),S(N3),S(N6),N1,N2,N4,ICOMP,IP)
         GO TO 20
C
  130 IF (ICHK.EQ.1) GO TO 140
      N1=NBX+1
      N2=N1+NIX*NBX
      N3=N2+NIX*NBX
      N4=N3+NBX
      N5=N4+NX
         CALL CMSM1  (S(1),S(N1),S(N2),S(N3),S(N4),S(N5),
     1               NX,NBX,NIX, NCOMPO,IP)
C
  140 WRITE (IP,2070)
         RETURN
 2000 FORMAT (8I5)
 2010 FORMAT (20X,'***  ',3I5,5X,2I5,5X,I5,3X,E10.4)
 2020 FORMAT (/,50X,'COMP. NO.',5I4,/)
 2030 FORMAT (3X,12E10.4)
 2040 FORMAT (50X,'*',3(2I5,5X))
 2050 FORMAT (50X,'**   ',8I7)
 2060 FORMAT (//,5X,'ETR GSTIF6',//)
 2070 FORMAT (//,5X,'END GSTIF6',//)
 2080 FORMAT (/)
         END
C GUY4.FOR 83-02-07 (OK,82-08-10) JTH    (82-03-12)
C
         SUBROUTINE STCOND (S,NB2,NB,N,NS,ISCON,ISOL,ICOMP,IP)
         DOUBLE PRECISION KBB,PIB,KII,PE3
         COMMON/IDO/IDO(600)
         COMMON/GARB/C(600)
```

```
      DIMENSION S(NS)
C
      WRITE (IP,2070)
   10 ENCODE (10,2010,PIB) ISOL,ICOMP
      WRITE (IP,2050) ICOMP,ISOL,NB2,NB,N,NS
C
      M=N
   20 J1=IDO(M)
      DO 30 J=1,M
   30 C(J)=S(J1+J)/S(J1+M)
      DO 40 I=1,M-1
      DO 40 J=I,M
      KO=IDO(J)+I
   40 S(KO)=S(KO)-C(J)*S(J1+I)
C
      DO 50 J=1,M
   50 S(J1+J)=C(J)
      M=M-1
      IF (M-NB) 60,60,20
   60 M=NB+1
   70 KO=IDO(M)
      DO 80 I=1,NB
   80 C(I)=S(KO+I)
      DO 90 JO=M+1,N
      KO=IDO(JO)
      CC=S(KO+M)
      DO 90 I=1,NB
   90 S(KO+I)=S(KO+I)-C(I)*CC
      M=M+1
      IF (M-N) 70,100,100
  100 CONTINUE
C
      IF (ISCON.NE.1) GO TO 120
      WRITE (IP,2060)
      DO 110 J=1,NB
      IO=IDO(J)
  110 WRITE (IP,2040) (S(IO+KO),KO=1,J)
C
  120 IF (ISOL.EQ.1 .AND. ISCON.EQ.0) GO TO 140
      ENCODE (10,2000,KBB) ISOL,ICOMP
      OPEN  (UNIT=3,FILE=KBB,ACCESS='SEQOUT')
      DO 130 J=1,NB
      IO=IDO(J)
  130 WRITE (3,2040) (S(IO+KO),KO=1,J)
      CLOSE (UNIT=3,FILE=KBB)
  140 CONTINUE
      IF (ISOL.NE.3) GO TO 150
      ENCODE(10,2030,PE3) ISOL,ICOMP
      OPEN  (UNIT=3,FILE=PE3,ACCESS='SEQIN')
```

```
         READ  (3,2040) (S(I),I=1,N)
         CLOSE (UNIT=3,FILE=PE3)
  150 CONTINUE
C
         OPEN (UNIT=3,FILE=PIB,ACCESS='SEQOUT')
      JO=IDO(NB+1)
      NI=N-NB
      DO 180 I=1,NB
      IK=JO+I
      DO 160 J=1,NI
      C(J)=-S(IK)
  160 IK=IK+(NB+J)
      IF (ISOL.NE.3) GO TO 180
      CC=0.
      DO 170 J=1,NI
  170 CC=CC+C(J)*S(NB+J)
      S(I)=S(I)+CC
  180 WRITE (3,2040) (C(KO), KO=1,NI)
         CLOSE (UNIT=3,FILE=PIB)
      IF (ISOL.NE.3) GO TO 190
         OPEN  (UNIT=3,FILE=PE3,ACCESS='SEQOUT')
         WRITE (3,2040) (S(I),I=1,NB)
         CLOSE (UNIT=3,FILE=PE3)
  190 CONTINUE
C
      IF (ISOL.NE.1) GO TO 210
      ISOL=2
      N=NB
      NB=NB2
      NS=IDO(N)+N
         ENCODE (10,2020,KII) ISOL,ICOMP
         OPEN (UNIT=3,FILE=KII,ACCESS='SEQOUT')
      NBO=NB+1
      DO 200 J=NBO,N
      KO=IDO(J)
  200 WRITE (3,2040) (S(KO+IO),IO=NBO,J)
         CLOSE (UNIT=3,FILE=KII)
         GO TO 10
  210 WRITE (IP,2080)
         RETURN
 2000   FORMAT ('KBB',I1,I1,'.DAT',1X)
 2010   FORMAT ('PIB',I1,I1,'.DAT',1X)
 2020   FORMAT ('KII',I1,I1,'.DAT',1X)
 2030   FORMAT ('PE3',I1,I1,'.DAT',1X)
 2040 FORMAT (5X,5E15.8)
 2050 FORMAT (/,50X,'!!!  ',6I7,/)
 2060 FORMAT (/)
 2070 FORMAT (//,1X,'ETR STCOND',/)
 2080 FORMAT ( /,1X,'END STCOND',//)
```

```
      END
C
C                                  82-8-12
C
      SUBROUTINE GUYRED (Y,NB,N,NS,IP)
      COMMON/IDO/IDO(600)
      COMMON/GARB/C1(600)
      DIMENSION Y(NS)
C
      OPEN (UNIT=3,FILE='PIB30.DAT',ACCESS='SEQIN')
   WRITE (IP,2010)
   NI=N-NB
   JO=IDO(NB+1)
   DO 50 I=1,NB
   READ  (3,2000) (C1(K),K=1,NI)
   DO 50 J=1,NB
   IK=JO+J
   CC=0.
   DO 10 K=1,NI
   CC=CC+C1(K)*Y(IK)
10 IK=IK+(NB+K)
   KO=IDO(J)+I
   IF (J-I) 40,30,20
20 Y(KO)=Y(KO)+CC
   GO TO 50
30 Y(KO)=Y(KO)+CC*2.
   GO TO 50
40 KO=IDO(I)+J
   Y(KO)=Y(KO)+CC
50 CONTINUE
      CLOSE (UNIT=3,FILE='PIB30.DAT')
C
      OPEN  (UNIT=3,FILE='PIB30.DAT',ACCESS='SEQIN')
   IOO=IDO(NB+1)
   DO 100 J=1,NB
   IO=IOO+J
   READ  (3,2000) (C1(K),K=1,NI)
   DO 90 I=1,NI
   Y(IO)=0.
   IK=IDO(NB+I)+NB
   DO 60 KO=1,I
   IKO=IK+KO
60 Y(IO)=Y(IO)+Y(IKO)*C1(KO)
   IF (I.EQ.NI) GO TO 80
   KO=IK+I
   DO 70 K=I+1,NI
   KO=KO+K-1+NB
70 Y(IO)=Y(IO)+Y(KO)*C1(K)
80 CONTINUE
```

```
   90 IO=IO+(NB+I)
  100 CONTINUE
         CLOSE (UNIT=3,FILE='PIB30.DAT')
C
         OPEN  (UNIT=3,FILE='PIB30.DAT',ACCESS='SEQIN')
      JO=IDO(NB+1)
      DO 120 I=1,NB
        READ  (3,2000) (C1(K),K=1,NI)
      DO 120 J=I,NB
      IK=JO+J
      IO=IDO(J)+I
      CC=0.
      DO 110 KO=1,NI
      CC=CC+Y(IK)*C1(KO)
  110 IK=IK+(NB+KO)
  120 Y(IO)=Y(IO)+CC
         CLOSE (UNIT=3,FILE='PIB30.DAT')
C
         OPEN  (UNIT=3,FILE='MBB30.DAT',ACCESS='SEQOUT')
      DO 130 J=1,NB
      KO=IDO(J)
  130 WRITE (3,2000) (Y(KO+IO),IO=1,J)
         CLOSE (UNIT=3,FILE='MBB30.DAT')
         WRITE (IP,2020)
         RETURN
 2000    FORMAT (5X,5E15.8)
 2010    FORMAT (//,1X,'ETR GUYAN',/)
 2020    FORMAT ( /,1X,'END GUYAN',//)
         END
C
C
C CMS3.FOR  (OK,83-01-31,01-05) 83-01-11  82-08-15 JTH
C
         SUBROUTINE CMSM1 (YBB,PH,YBI,YBN,XM,PE,  NX,NBX,NIX,NCOMPO,IP)
         DOUBLE PRECISION PIB,PIN,DMB,DMI,MBB,MBN,PE1,PE2
         COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1               ICBS(168,4),ICNS(36,4),NSYS(4)
         COMMON/GARB/C1(600)
         DIMENSION YBB(NBX),PH(NIX,NBX),YBI(NIX,NBX),XM(NX),YBN(NBX),
     1              PE(NX)
C
      WRITE (IP,2090)
      ISOL=2
      DO 130 ICOMP=1,NCOMPO
      NB=NDC3(3,ICOMP)
      NI=NDC1(4,ICOMP)
       N=NDC3(4,ICOMP)
      NNP=NCRMO(ICOMP)
      N2=NB+NNP
```

```
      ENCODE (10,2000,PIB) ISOL,ICOMP
      ENCODE (10,2010,PIN) ISOL,ICOMP
      ENCODE (10,2020,DMB) ISOL,ICOMP
      ENCODE (10,2030,MBB) ISOL,ICOMP
      ENCODE (10,2040,MBN) ISOL,ICOMP
      ENCODE (10,2050,DMI) ISOL,ICOMP
      ENCODE (10,2060,PE1) ISOL,ICOMP
      ENCODE (10,2070,PE2) ISOL,ICOMP
C
C        GET COMP MASS IN P-COORD
C
         OPEN (UNIT=3,FILE=DMB,ACCESS='SEQIN')
      CALL ZERO (XM(1),XM(N))
      NA=NDC2(1,ICOMP)
      NC=NDC3(2,ICOMP)
      READ (3,2080) (C1(K),K=1,NC)
      DO 10 I=1,NC
      IO=NA+I-1
   10 XM(IO)=XM(IO)+C1(I)
         CLOSE (UNIT=3,FILE=DMB)
C
         OPEN  (UNIT=3,FILE=DMI,ACCESS='SEQIN')
      NA=NDC2(4,ICOMP)
      NC=NDC1(4,ICOMP)
      READ (3,2080) (C1(K),K=1,NC)
      DO 20 I=1,NC
      IO=NA+I-1
   20 XM(IO)=XM(IO)+C1(I)
         CLOSE (UNIT=3,FILE=DMI)
C
         OPEN  (UNIT=3,FILE=PE1,ACCESS='SEQIN')
         READ (3,2080) (PE(I),I=1,N)
         CLOSE (UNIT=3,FILE=PE1)
C
         OPEN (UNIT=3,FILE=PIB,ACCESS='SEQIN')
      DO 40 J=1,NB
      READ  (3,2080) (PH(K,J), K=1,NI)
      CC=0.
      DO 30 I=1,NI
      CC=CC+PH(I,J)*PE(NB+I)
   30 YBI(I,J)=PH(I,J)*XM(NB+I)
   40 PE(J)=PE(J)+CC
         CLOSE (UNIT=3,FILE=PIB)
C
         OPEN (UNIT=2,FILE=MBB,ACCESS='SEQOUT')
      DO 70 J=1,NB
      CALL ZERO (YBB(1),YBB(J))
      YBB(J)=XM(J)
      DO 60 I=1,J
```

```
      CC=0.
      DO 50 K=1,NI
   50 CC=CC+YBI(K,I)*PH(K,J)
   60 YBB(I)=YBB(I)+CC
   70 WRITE (2,2080) (YBB(JO),JO=1,J)
         CLOSE (UNIT=2,FILE=MBB)
C
         OPEN (UNIT=3,FILE=PIN,ACCESS='SEQIN')
         OPEN (UNIT=2,FILE=MBN,ACCESS='SEQOUT')
      DO 110 J=1,NNP
      READ  (3,2080) (PH(K,J), K=1,NI)
      DO 90 I=1,NB
      CC=0.
      DO 80 K=1,NI
   80 CC=CC+YBI(K,I)*PH(K,J)
   90 YBN(I)=CC
      CC=0.
      DO 100 K=1,NI
  100 CC=CC+PE(NB+K)*PH(K,J)
      XM(J)=CC
  110 WRITE (2,2080) (YBN(IO),IO=1,NB)
         CLOSE (UNIT=3,FILE=PIN)
         CLOSE (UNIT=2,FILE=MBN)
      DO 120 J=1,NNP
  120 PE(NB+J)=XM(J)
         OPEN  (UNIT=3,FILE=PE2,ACCESS='SEQOUT')
         WRITE (3,2080) (PE(I),I=1,N2)
         CLOSE (UNIT=3,FILE=PE2)
  130 CONTINUE
C
      WRITE (IP,2100)
         RETURN
 2000 FORMAT ('PIB',I1,I1,'.DAT',1X)
 2010 FORMAT ('PIN',I1,I1,'.DAT',1X)
 2020 FORMAT ('DMB',I1,I1,'.DAT',1X)
 2030 FORMAT ('MBB',I1,I1,'.DAT',1X)
 2040 FORMAT ('MBN',I1,I1,'.DAT',1X)
 2050 FORMAT ('DMI',I1,I1,'.DAT',1X)
 2060 FORMAT ('PE1',I1,I1,'.DAT',1X)
 2070 FORMAT ('PE2',I1,I1,'.DAT',1X)
 2080 FORMAT (5X,5E15.8)
 2090 FORMAT (//,1X,'ETR CMSM1',/)
 2100 FORMAT ( /,1X,'END CMSM1',//)
         END
C
C
C SYSM.FOR  83-02-07     (OK,83-01-05)  82-08-19   JTH
C
         SUBROUTINE SYSM (SM,NS,NCOMP,IP)
```

```
      DOUBLE PRECISION MBB,MBN
      COMMON/GARB/C1(600)
      COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1              ICBS(168,4),ICNS(36,4),NSYS(4)
      DIMENSION SM(NS)
C
C      GET SYS MASS IN Q-COORD
C
      WRITE (IP,2040)
      JO=NSYS(1)
      CALL ZERO (SM(1),SM(NS))
      DO 10 I=1,JO
      IO=(I*I+I)/2
   10 SM(IO)=1.
      ISOL=2
C
      DO 50 ICOMP=1,NCOMP
      NB=NDC3(3,ICOMP)
      NNP=NCRMO(ICOMP)
      ENCODE (10,2000,MBB) ISOL,ICOMP
      ENCODE (10,2010,MBN) ISOL,ICOMP
      WRITE (IP,2030) ICOMP,NB,NNP
C
         OPEN (UNIT=3,FILE=MBB,ACCESS='SEQIN')
      DO 30 J=1,NB
      JS=ICBS(J,ICOMP)
      READ (3,2020) (C1(JO),JO=1,J)
      JO=JS-1
      KO=(JO*JO+JO)/2
      DO 30 I=1,J
      IS=ICBS(I,ICOMP)
      KS=KO+IS
      IF (IS.GT.JS) GO TO 20
      SM(KS)=SM(KS)+C1(I)
      GO TO 30
   20 WRITE (IP,2030) J,I,JS,IS,KS,ICOMP
         STOP
   30 CONTINUE
         CLOSE (UNIT=3,FILE=MBB)
C
         OPEN (UNIT=3,FILE=MBN,ACCESS='SEQIN')
      DO 40 J=1,NNP
      READ (3,2020) (C1(I),I=1,NB)
      IS=ICNS(J,ICOMP)
      DO 40 I=1,NB
      JS=ICBS(I,ICOMP)
      JO=JS-1
      KS=(JO*JO+JO)/2+IS
   40 SM(KS)=SM(KS)+C1(I)
```

```
         CLOSE (UNIT=3,FILE=MBN)
   50 CONTINUE
C
         RETURN
 2000 FORMAT ('MBB',I1,I1,'.DAT',1X)
 2010 FORMAT ('MBN',I1,I1,'.DAT',1X)
 2020 FORMAT (5X,5E15.8)
 2030 FORMAT (5X,7I5)
 2040 FORMAT (//,1X,'ETR SYSM',/)
         END
C
C
C SYSK.FOR  83-02-07  (OK,83-01-05)    82-08-19   JTH
C
         SUBROUTINE SYSK (SK,NS,NCOMP,IP)
         DOUBLE PRECISION KBB,EVA
         COMMON/GARB/C1(600)
         COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1               ICBS(168,4),ICNS(36,4),NSYS(4)
         DIMENSION SK(NS)
C
C         GET SYS STIFF IN Q-COORD
C
      WRITE (IP,2040)
      CALL ZERO (SK(1),SK(NS))
      INO=0
      ISOL=2
C
      DO 40 ICOMP=1,NCOMP
      NB=NDC3(3,ICOMP)
      NNP=NCRMO(ICOMP)
      ENCODE (10,2000,KBB) ISOL,ICOMP
      ENCODE (10,2010,EVA) ISOL,ICOMP
      WRITE (IP,2030) ICOMP,NB,NNP
C
         OPEN (UNIT=3,FILE=EVA,ACCESS='SEQIN')
      READ (3,2020) (C1(I),I=1,NNP)
      DO 10 I=1,NNP
      INO=INO+1
      IO=(INO*INO+INO)/2
   10 SK(IO)=C1(I)
         CLOSE (UNIT=3,FILE=EVA)
C
         OPEN (UNIT=3,FILE=KBB,ACCESS='SEQIN')
      DO 30 J=1,NB
      JS=ICBS(J,ICOMP)
      READ (3,2020) (C1(JO),JO=1,J)
      JO=JS-1
      KO=(JO*JO+JO)/2
```

```
      DO 30 I=1,J
      IS=ICBS(I,ICOMP)
      KS=KO+IS
      IF (IS.GT.JS) GO TO 20
      SK(KS)=SK(KS)+C1(I)
      GO TO 30
   20 WRITE (IP,2030) J,I,JS,IS,KS,ICOMP
          STOP
   30 CONTINUE
         CLOSE (UNIT=3,FILE=KBB)
   40 CONTINUE
C
         RETURN
 2000 FORMAT ('KBB',I1,I1,'.DAT',1X)
 2010 FORMAT ('EVA',I1,I1,'.DAT',1X)
 2020 FORMAT (5X,5E15.8)
 2030 FORMAT (5X,7I5)
 2040 FORMAT (//,1X,'ETR SYSK',/)
         END
C
C
C SYSP.FOR  83-02-07      82-11-14   J.T.HUANG
C
         SUBROUTINE SYSP (SP,N,NCOMP,IP)
         DOUBLE PRECISION PE2
         COMMON/GARB/PE(600)
         COMMON/FEIDS/NCRMO(4),NDC1(7,4),NDC2(7,4),NDC3(7,4),
     1              ICBS(168,4),ICNS(36,4),NSYS(4)
         DIMENSION SP(N)
C
C         GET SYS LOAD IN Q-COORD
C
      WRITE (IP,2070)
      CALL ZERO (SP(1),SP(N))
      ISOL=2
      DO 30 ICOMP=1,NCOMP
      WRITE (IP,2060)
      NB=NDC3(3,ICOMP)
      NNP=NCRMO(ICOMP)
      N2=NB+NNP
      WRITE (IP,2010) ICOMP,NB,NNP,N2
      ENCODE (10,2000,PE2) ISOL,ICOMP
         OPEN  (UNIT=3,FILE=PE2,ACCESS='SEQIN')
         READ  (3,2040) (PE(I),I=1,N2)
         CLOSE (UNIT=3,FILE=PE2)
      WRITE (IP,2020)
      DO 10 J=1,NB
      JS=ICBS(J,ICOMP)
      SP(JS)=SP(JS)+PE(J)
```

```
   10 WRITE (IP,2050) J,JS,SP(JS)
      WRITE (IP,2030)
      DO 20 J=1,NNP
      JS=ICNS(J,ICOMP)
      SP(JS)=SP(JS)+PE(NB+J)
   20 WRITE (IP,2050) J,JS,SP(JS)
   30 CONTINUE
C
         OPEN  (UNIT=3,FILE='PE330.DAT',ACCESS='SEQOUT')
         WRITE (3,2040) (SP(I),I=1,N)
         CLOSE (UNIT=3,FILE='PE330.DAT')
C
         RETURN
 2000 FORMAT ('PE2',I1,I1,'.DAT',1X)
 2010 FORMAT (25X,I5,3X,3I6,/)
 2020 FORMAT (25X,'BOUNDARY DOF')
 2030 FORMAT (25X,'NORMAL   DOF')
 2040    FORMAT (5X,5E15.8)
 2050    FORMAT (2X,2I5,F12.6)
 2060    FORMAT (/)
 2070    FORMAT (//,1X,'ETR SYSP',/)
         END
C
C
C EPAAA.FOR 83-01-27  (OK,12-19) ( OK,82-06-14) 82-06-12                JTH
C
         SUBROUTINE GEVPS2 (XK,XM,EC,EA,N,NS,NMOD,ICOMP,IP)
         COMMON/IDO/IDO(600)
         DIMENSION XK(NS),XM(NS),EC(N,N),EA(N)
C
         OPEN (UNIT=3,FILE='KBB30.DAT',ACCESS='SEQIN')
         WRITE (IP,2000)
         ISOL=3
      N22=N*N
      DO 10 J=1,N
      KO=IDO(J)
   10 READ (3,2020) (XK(KO+I),I=1,J)
         CLOSE (UNIT=3,FILE='KBB30.DAT')
         CALL EIGEN (XM,EC,N,NS,N22)
      DO 20 I=1,N
      IO=IDO(I)+I
   20 EA(I)=XM(IO)
         CALL NZGMZ (XM,EC,EA,N,NS,IP)
         CALL KBMKM (EC,XK,XM,N,NS,IP)
         CALL EIGEN (XK,EC,N,NS,N22)
      DO 30 I=1,N
      IO=IDO(I)+I
   30 EA(I)=XK(IO)
      N1=N+1
```

```
         CALL EPOST1 (EA,XK(1),XK(N1),N,ISOL,ICOMP,IP)
         CALL SYMFL  (XM,EC,N,NS,IP)
         CALL EPOST2 (EC,N,NMOD,ISOL,ICOMP,IP)
      WRITE (IP,2010)
         RETURN
 2000 FORMAT (//,5X,'*** ETR GEVPS2 ***',/)
 2010 FORMAT (/,5X, '*** END GEVPS2 ***',//)
 2020 FORMAT (5X,5E15.8)
         END
C
C              82-06-12 JTH
C
         SUBROUTINE STDEGC (K,EC,EA,N,NS,NMOD,ICOMP,IP)
         DOUBLE PRECISION PIN,EVA,KII,DMI
         REAL K,M2
         COMMON/IDO/IDO(600)
         COMMON/EGV1/IA2(600)
         COMMON/GARB/M2(600)
         DIMENSION EC(N,N),K(NS),EA(N)
C
         ISOL=2
         ENCODE (10,2000,KII) ISOL,ICOMP
         ENCODE (10,2010,DMI) ISOL,ICOMP
C
      WRITE (IP,2030)
         OPEN  (UNIT=3,FILE=DMI,ACCESS='SEQIN')
         READ (3,2020) (M2(I),I=1,N)
         CLOSE (UNIT=3,FILE=DMI)
      N22=N*N
      DO 10 I=1,N
   10 M2(I)=1./SQRT(M2(I))
         OPEN  (UNIT=3,FILE=KII,ACCESS='SEQIN')
      DO 20 J=1,N
      KO=IDO(J)
      READ (3,2020) (K(KO+IO),IO=1,J)
      DO 20 I=1,J
   20 K(KO+I)=K(KO+I)*M2(I)*M2(J)
         CLOSE (UNIT=3,FILE=KII)
         CALL EIGEN (K,EC,N,NS,N22)
      DO 30 I=1,N
      IO=IDO(I)+I
   30 EA(I)=K(IO)
      N1=N+1
         CALL EPOST1 (EA,K(1),K(N1),N,ISOL,ICOMP,IP)
      DO 40 J=1,N
      JO=IA2(J)
      DO 40 I=1,N
   40 EC(I,JO)=EC(I,JO)*M2(I)
         CALL EPOST2 (EC,N,NMOD,ISOL,ICOMP,IP)
```

```
        WRITE (IP,2040)
        RETURN
 2000 FORMAT ('KII',I1,I1,'.DAT',1X)
 2010 FORMAT ('DMI',I1,I1,'.DAT',1X)
 2020 FORMAT (5X,5E15.8)
 2030 FORMAT (//,40X,'ETR STDEGC',/)
 2040 FORMAT (/, 40X,'END STDEGC',//)
        END
C
C DISPL1.FOR  OK,83-01-27 JTH   10-18                  ( OK,   82-12-27)
C
        SUBROUTINE DISPL1 (EC,EA,P, Y,F,T, Q,QN,EN, N,NO,I1,N2,DT,
     1                 NMD,FX,EM,IP,ICHK)
        DOUBLE PRECISION ACR
        COMMON/EGV1/IA2(600)
        COMMON/GARB/AA(500),ZO(100)
        DIMENSION EC(N,N),EA(N),P(N),T(NO),F(NO),Y(NO),Q(N2),
     1               EN(NMD),QN(N2,NMD)
C
      WRITE (IP,2100)
        OPEN  (UNIT=2,FILE='TRS30.DAT',ACCESS='SEQIN')
        READ  (2,2050) FX,EM
        WRITE (6,2050) FX,EM
      IF (FX.EQ.0.) FX=40.
      IF (EM.EQ.0.) EM=0.001
      WX=FX*2.*3.14159265
      XM=1.
      R1=1.
      R2=1.
      R3=1.
      R4=1.
        OPEN  (UNIT=3,FILE='DAM.DAT',ACCESS='SEQIN')
        READ  (3,2050) (ZO(I),I=1,NMD)
        CLOSE (UNIT=3,FILE='DAM.DAT')
        OPEN  (UNIT=3,FILE='PE330.DAT',ACCESS='SEQIN')
        READ  (3,2030) (P(I),I=1,N)
        CLOSE (UNIT=3,FILE='PE330.DAT')
C
      READ  (2,2070) NTS
      WRITE (6,2070) NTS
      DO 100 ITS=1,NTS
      WRITE (IP,2000) ITS
      READ  ( 2,2010) NO,I1,N2,DT
      WRITE (IP,2010) NO,I1,N2,DT
      WRITE (IP,2060)
      IO=I1-1
      NOO=IO+N2
      IF ( NO.GT.NOO) NO=NOO
      IF (NOO.GT.NO ) N2=NO-IO
```

```
      T(1)=IO*DT
      DO 10 I=2,N2
   10 T(I)=T(I-1)+DT
C
      ENCODE (10,2020,ACR) ITS
         OPEN  (UNIT=3,FILE=ACR,ACCESS='SEQIN')
         READ  (3,2030) (F( JO),JO=1,NO)
         WRITE (6,2040) (F(JO), JO=1,NO)
         CLOSE (UNIT=3,FILE=ACR)
C
      IF (ITS.GT.1) GO TO 50
      DO 30 K=1,NMD
      JO=IA2(K)
      W=SQRT(EA(JO))
         IF (W.GT.WX) GO TO 40
      EY=0.
      DO 20 I=1,N
      EY=EY+P(I)*EC(I,JO)
   20 CONTINUE
      IF (K.EQ.1) E1=ABS(EY)
      EYO=ABS(EY)
      IF (EYO.GT.E1) E1=EYO
      R4=R3
      R3=R2
      R2=R1
      R1=EY/E1
      R1=ABS(R1)
         WRITE (IP,2080) K,W,EY,E1,R1
         IF (R1.LT.EM .AND. R2.LT.EM .AND. R3.LT.EM .AND.
     1       R4.LT.EM ) GO TO 40
         EN(K)=EY
   30    CONTINUE
   40    NMD=K-1
         WRITE (IP,2080) NMD
C
   50 DO 70 K=1,NMD
         EY=EN(K)
         Z=ZO(K)
         JO=IA2(K)
         W=SQRT(EA(JO))
         CPS=W/(2.*3.1415926)
         CALL SDFEXP (EY,F,Y, XM,Z,W, DT,NO)
      R1=EY/E1
      R1=ABS(R1)
      IF (K.GT.20 .OR. CPS.GT.15. .OR. R1.LT.0.05) GO TO 60
      WRITE (6,2060)
      WRITE (6,2040) (Y(I), I=1,NO)
      WRITE (6,2060)
   60 DO 70 IT=1,N2
```

```
   70 QN(IT,K)=Y(IT+IO)
C
      WRITE (IP,2060)
      DO 90 IT=1,N2
      WRITE (IP,2080) IT,T(IT)
      DO 80 IR=1,N
      Q(IR)=0.
      DO 80 IM=1,NMD
      JO=IA2(IM)
      Q(IR)=Q(IR)+EC(IR,JO)*QN(IT,IM)
   80 CONTINUE
      WRITE(IP,2040) (Q(IR),IR=1,N)
   90 CONTINUE
  100 CONTINUE
         CLOSE (UNIT=2,FILE='TRS30.DAT')
C
      WRITE (IP,2110)
         RETURN
 2000 FORMAT (////,80X,I2,'-TH TIME HISTORY',//)
 2010 FORMAT (3I5,5X,F10.3)
 2020 FORMAT ('ACR',I1,'.DAT',2X)
 2030 FORMAT (5X,5E15.8)
 2040 FORMAT (15X,10E11.5)
 2050 FORMAT (8F10.5)
 2060 FORMAT (/)
 2070 FORMAT (8I5)
 2080 FORMAT (1X,I4,F9.4,4F9.3)
 2090 FORMAT (40X,3I4,3F11.4)
 2100 FORMAT (/,5X,'ETR DISPL',/)
 2110 FORMAT (/,5X,'END DISPL',/)
         END
C FRESP2.FOR   83-01-27 (OK,82-12-20) (OK, 82-02-11)    JTH   82-11-28
C
         SUBROUTINE SDFEXP (PO,F,Y,  XM,Z,W,  DT,NP)
         COMMON/GARB/WD,ZW,WW,XK,TZW,W1,ZWW,W2D,AA,TK,WDT,CWDT,SWDT,EO,
     1               BO,AO,A,B,C,D,A1,B1,C1,D1,YO,Y1,NP1, AAAO(573)
         DIMENSION Y(NP),F(NP)
C
      WRITE (6,2000) PO,XM,Z,W,DT,NP
      Y1=0.*PO*F(1)*DT
      WD=W*SQRT(1.-Z*Z)
      ZW=Z*W
      WW=W*W
      TZW=2.*Z/W
      W1=1./WD
      ZWW=ZW*W1
      W2D=WW*W1
      AA=(2.*Z*Z-1.)*W1
      TK=1./(DT*XM*WW)
```

```
      WDT=WD*DT
      CWDT=COS(WDT)
      SWDT=SIN(WDT)
      EO=EXP(-ZW*DT)
C
      BO=AA*SWDT+TZW*CWDT
      AO=-BO-ZW*DT*W1*SWDT-DT*CWDT
      A=(EO*AO+TZW)*TK
      B=(EO*BO-TZW+DT)*TK
      C=EO*(CWDT+ZWW*SWDT)
      D=W1*EO*SWDT
      BO=ZWW*SWDT+CWDT
      AO=BO+W2D*DT*SWDT
      A1=( EO*AO-1.)*TK
      B1=(-EO*BO+1.)*TK
      C1=-W2D*EO*SWDT
      D1=EO*(CWDT-ZWW*SWDT)
         Y(1)=0.
         DO 10 I=1,NP-1
         Y(I+1)= PO*(A *F(I)+B *F(I+1))  +C *Y(I)+D *Y1
   10    Y1=     PO*(A1*F(I)+B1*F(I+1))  +C1*Y(I)+D1*Y1
         RETURN
 2000    FORMAT(  20X,' *** ETR SDFEXP ... ',2F12.6,3F7.3,I4,/)
         END
C
C
C EPOST.FOR  83-01-02 ( OK,82-12-19  82-10-21) JTH  81-03-24
C
         SUBROUTINE EPOST1 (EA,EA2,IA,N,ISOL,ICOMP,IP)
         DOUBLE PRECISION EVA
         COMMON/EGV1/IA2(600)
         DIMENSION EA(N),EA2(N),IA(N)
C
C
      CALL ZERO (IA(1),IA(N))
      DO 20 I=1,N
      X=1.E+12
      DO 10 J=1,N
      IF (IA(J).NE.0) GO TO 10
      IF (X.LT.EA(J)) GO TO 10
      X=EA(J)
      IM=J
   10 CONTINUE
      IA(IM)=I
   20 CONTINUE
      DO 30 J=1,N
      I=IA(J)
   30 IA2(I)=J
C
```

```
      C=2.*3.14159265
      WRITE (IP,2020)
      DO 40 I=1,N
      JO=IA2(I)
      EA2(I)=EA(JO)
      W=SQRT (EA(JO))
      CPS=W/C
      T=1./CPS
   40 WRITE (IP,2010) I,JO,EA(JO),W,CPS,T
      IF (ISOL.NE.2) GO TO 50
         ENCODE (10,2030,EVA) ISOL,ICOMP
         OPEN  (UNIT=3,FILE=EVA,ACCESS='SEQOUT')
         WRITE (3,2000) (EA2(I),I=1,N)
         CLOSE (UNIT=3,FILE=EVA)
   50 CONTINUE
         RETURN
 2000 FORMAT (5X,5E15.8)
 2010 FORMAT (2(2X,I4),2(2X,E15.8),2F12.4)
 2020 FORMAT (/,5X,' I-TH LOWEST, JO-LOC,  W2  W  CPS  T ... ',/)
 2030 FORMAT ('EVA',I1,I1,'.DAT',1X)
         END
C
         SUBROUTINE EPOST2 (EC,N,NMOD,ISOL,ICOMP,IP)
         DOUBLE PRECISION PIN
         COMMON/EGV1/IA2(600)
         DIMENSION EC(N,N)
C
         ENCODE (10,2050,PIN) ISOL,ICOMP
         OPEN  (UNIT=3,FILE=PIN,ACCESS='SEQOUT')
      WRITE (IP,2010)
      NMOD1=NMOD*3/2+3
      NMOD2=N*2/3+1
      IF (NMOD1.GT.NMOD2) NMOD1=NMOD2
      IF (N.EQ.NMOD) NMOD1=N
      PMOD=40
      IF (PMOD.GT.NMOD1) PMOD=NMOD1
      DO 20 I=1,PMOD
         WRITE (IP,2020) I
      JO=IA2(I)
      KO=N/10+1
      DO 10 K=1,KO
      K1=(K-1)*10+1
      K2=K1+9
         IF (K2.GT.N) K2=N
   10    WRITE (IP,2040) K1,K2,  (EC(IDOF,JO), IDOF=K1,K2)
         IF (ISOL.NE.2) GO TO 20
         WRITE ( 3,2000) (EC(KO,JO),KO=1,N)
   20    CONTINUE
         CLOSE (UNIT=3,FILE=PIN)
```

```
        WRITE (IP,2030)
C
        RETURN
 2000 FORMAT (5X,5E15.8)
 2010 FORMAT (////, 10X,' MODE SHAPE ...  (I)=I-TH LOWEST  ',/)
 2020 FORMAT (/,2X,'NO.',I3,' LOWEST MODE',/)
 2030 FORMAT (/,10X, 'END OF MODE SHAPES',/)
 2040 FORMAT (2X,I4,1X,'TO',I4,2X,10E11.4)
 2050 FORMAT ('PIN',I1,I1,'.DAT',1X)
        END
C
C EPCCC.FOR     83-01-27   OK,82-12-19,11-10 80-9-5  JTHUANG
C 80-09-05 FROM IBM SSP
C REVISED
C
         SUBROUTINE EIGEN (A,R,N,NS,N22)
         DIMENSION A(NS),R(N22)
      MV=0
      S2=SQRT(2.)
   10 RG=1.0E-6
      IF (MV-1) 20,50,20
   20 IQ=-N
      DO 40 J=1,N
      IQ=IQ+N
      DO 40 I=1,N
      IJ=IQ+I
      R(IJ)=0.
      IF (I-J) 40,30,40
   30 R(IJ)=1.
   40 CONTINUE
C       INITIAL AND FINAL NORMS (AM & AX)
   50 AM=0.
      DO 70 I=1,N
      DO 70 J=I,N
      IF (I-J) 60,70,60
   60 IA=I+(J*J-J)/2
      AM=AM+A(IA)*A(IA)
   70 CONTINUE
C
      IF (AM) 360,360,80
   80 AM=S2*SQRT(AM)
      AX=AM*RG/FLOAT(N)
C       INITIALIZE INDICATORS & GET THRESHOLD THR
      IND=0
      THR=AM
   90 THR=THR/FLOAT(N)
  100 L=1
  110 M=L+1
C       SIN & COS
```

```
  120 MQ=(M*M-M)/2
      LQ=(L*L-L)/2
      LM=L+MQ
  130 IF (ABS(A(LM))-THR)290,140,140
  140 IND=1
      LL=L+LQ
      MM=M+MQ
      X=0.5*(A(LL)-A(MM))
  150 Y=-A(LM)/SQRT(A(LM)*A(LM)+X*X)
      IF (X) 160,170,170
  160 Y=-Y
  170 SX=Y/SQRT(2.*(1.+(SQRT(1.-Y*Y))))
      SX2=SX*SX
  180 CX=SQRT(1.-SX2)
      CX2=CX*CX
      SCS=SX*CX
C        ROTATE L &M COLUMNS
      ILQ=N*(L-1)
      IMQ=N*(M-1)
      DO 280 I=1,N
      IQ=(I*I-I)/2
      IF (I-L) 190,260,190
  190 IF (I-M) 200,260,210
  200 IM=I+MQ
      GO TO 220
  210 IM=M+IQ
C
  220 IF (I-L) 230,240,240
  230 IL=I+LQ
      GO TO 250
  240 IL=L+IQ
  250 X=A(IL)*CX-A(IM)*SX
      A(IM)=A(IL)*SX+A(IM)*CX
      A(IL)=X
  260 IF (MV-1) 270,280,270
  270 ILR=ILQ+I
      IMR=IMQ+I
      X=R(ILR)*CX-R(IMR)*SX
      R(IMR)=R(ILR)*SX+R(IMR)*CX
      R(ILR)=X
  280 CONTINUE
      X=2.*A(LM)*SCS
      Y=A(LL)*CX2+A(MM)*SX2-X
      X=A(LL)*SX2+A(MM)*CX2+X
      A(LM)=(A(LL)-A(MM))*SCS+A(LM)*(CX2-SX2)
      A(LL)=Y
      A(MM)=X
C         TEST FOR COMPLETION
C         TEST FOR M=LAST COL.
```

```
  290 IF (M-N) 300,310,300
  300 M=M+1
      GO TO 120
C        TEST FOR L=SECOND FROM LAST COL
  310 IF (L-(N-1)) 320,330,320
  320 L=L+1
      GO TO 110
  330 IF (IND-1) 350,340,350
  340 IND=0
      GO TO 100
C        COMPARE THRESHOLD WITH FINAL NORM
  350 IF (THR-AX) 360,360,90
C        SORT EIGVA & EIGVEC
  360 IQ=-N
      DO 400 I=1,N
      IQ=IQ+N
      LL=I+(I*I-I)/2
      JQ=N*(I-2)
      DO 400 J=I,N
      JQ=JQ+N
      MM=J+(J*J-J)/2
      IF (A(LL)-A(MM)) 370,400,400
  370 X=A(LL)
      A(LL)=A(MM)
      A(MM)=X
      IF (MV-1) 380,400,380
  380 DO 390 K=1,N
      ILR=IQ+K
      IMR=JQ+K
      X=R(ILR)
      R(ILR)=R(IMR)
  390 R(IMR)=X
  400 CONTINUE
         RETURN
         END
C
C 82-06-08 JTH    82-11-10
C
         SUBROUTINE SYMFL  (A,B,N,NS,IP)
         DIMENSION A(NS),B(N,N)
         COMMON/GARB/XO(600)
      WRITE (IP,2000)
      DO 40 J=1,N
      DO 10 K=1,N
   10 XO(K)=B(K,J)
      IS=0
      DO 40 I=1,N
      B(I,J)=0.
      DO 20 KO=1,I
```

```
      ISO=IS+KO
   20 B(I,J)=B(I,J)+A(ISO)*XO(KO)
      IF (I.EQ.N)  GO TO 40
      KO=IS+I
      I1=I+1
      DO 30 K=I1,N
      KO=KO+K-1
   30 B(I,J)=B(I,J)+A(KO)*XO(K)
      IS=IS+I
   40 CONTINUE
         RETURN
 2000 FORMAT (/,60X,'*** ETR SYMFL ***',/)
         END
C
         SUBROUTINE NZGMZ (C,A,G, N,NS,IP)
         DIMENSION C(NS),G(N),A(N,N)
         COMMON/GARB/XO(600)
      WRITE (IP,2000)
      INM=0
      IF (INM.EQ.0) GO TO 40
      DO 30 I=1,N
      SM=0.
      DO 10 K=1,N
   10 SM=SM+A(K,I)*A(K,I)
      DO 20 K=1,N
   20 A(K,I)=A(K,I)/SM
   30 CONTINUE
   40 DO 50 I=1,N
   50 G(I)=1./SQRT(G(I))
      IO=0
      DO 70 J=1,N
      DO 60 K=1,N
   60 XO(K)=G(K)*A(J,K)
      DO 70 I=1,J
      IO=IO+1
      C(IO)=0.
      DO 70 K=1,N
      C(IO)=C(IO)+A(I,K)*XO(K)
   70 CONTINUE
         RETURN
 2000 FORMAT (/,60X, '*** ETR NZGMZ ***',/)
         END
C
C 82-06-09 JTH   82-11-10
C
         SUBROUTINE KBMKM (PH,XK,XM,  N,NS,IP)
         COMMON/GARB/XO(600)
         DIMENSION PH(N,N),XK(NS),XM(NS)
      WRITE (IP,2000)
```

```
      IS=0
      DO 70 J=1,N
      DO 10 KO=1,J
      ISO=IS+KO
   10 XO(KO)=XM(ISO)
      IF (J.EQ.N) GO TO 30
      KO=IS+J
      DO 20 K=J+1,N
      KO=KO+K-1
   20 XO(K)=XM(KO)
   30 IK=0
      DO 60 L=1,N
      PH(L,J)=0.
      DO 40 KO=1,L
      IKO=IK+KO
   40 PH(L,J)=PH(L,J)+XK(IKO)*XO(KO)
      IF (L.EQ.N) GO TO 60
      KO=IK+L
      DO 50 K=L+1,N
      KO=KO+K-1
   50 PH(L,J)=PH(L,J)+XK(KO)*XO(K)
   60 IK=IK+L
   70 IS=IS+J
      IO=0
      DO 110 J=1,N
      IK=0
      DO 100 I=1,J
      IO=IO+1
      XK(IO)=0.
      DO 80 KO=1,I
      IKO=IK+KO
   80 XK(IO)=XK(IO)+XM(IKO)*PH(KO,J)
      IF (I.EQ.N) GO TO 100
      KO=IK+I
      DO 90 K=I+1,N
      KO=KO+K-1
   90 XK(IO)=XK(IO)+XM(KO)*PH(K,J)
  100 IK=IK+I
  110 CONTINUE
          RETURN
 2000 FORMAT (/,60X,'*** ETR KBMKM ***',/)
          END
C
C ELM.FOR 82-12-05  (OK, 12-04)   82-09-14,20 OK,   JTH     82-05-12
C
          SUBROUTINE KE2D6 (ICHK,IMP,ISP,DX,DY,IP,IPS)
          COMMON/SPROP/SP(20,5)
          COMMON/MPROP/XP(4,3)
          COMMON/CESTM/XG(6,6),XOO(108)
```

```
      COMMON/GARB/XE(6,6),ST(3,3),T(3,3),C1,C2,C3,C4,
     1        DXDY,C,S,E,XL,A,XI,AAAO(535)
C
      IF (ICHK.EQ.1 .AND. IPS.EQ.1) WRITE (IP,2000) DX,DY
      XL=SQRT(DX*DX+DY*DY)
      IF (XL.EQ.O.) STOP
      E=XP(IMP,1)
      A=SP(ISP,1)
      XI=SP(ISP,2)
      C1=E*A/XL
      C2=E*XI/(XL*XL)
      C3=12.*C2/XL
      C4=4.*C2*XL
      C2=C2*6.
C
      C=DX/XL
      S=DY/XL
      T(1,1)= C
      T(1,2)= S
      T(1,3)= O.
      T(2,1)=-S
      T(2,2)= C
      T(2,3)= O.
      T(3,1)= O.
      T(3,2)= O.
      T(3,3)= 1.
      XE(1,2)=O.
      XE(1,3)=O.
      XE(1,5)=O.
      XE(1,6)=O.
      XE(2,4)=O.
      XE(3,4)=O.
      XE(4,5)=O.
      XE(4,6)=O.
      XE(1,1)=C1
      XE(2,2)=C3
      XE(3,3)=C4
      XE(4,4)=C1
      XE(5,5)=C3
      XE(6,6)=C4
      XE(1,4)=-C1
      XE(2,3)= C2
      XE(2,5)=-C3
      XE(2,6)= C2
      XE(3,5)=-C2
      XE(3,6)= C4*0.5
      XE(5,6)=-C2
      DO 10 J=1,5
      DO 10 I=J+1,6
```

```
   10 XE(I,J)=XE(J,I)
      IF (ICHK.NE.1 .OR. IPS.NE.1) GO TO 20
      CALL PMATE ( T,1,1,3,3)
      CALL PMATE (XE,1,1,6,6)
   20 DO 80 M=1,2
      DO 80 N=1,M
      K=(M-1)*3
      L=(N-1)*3
      DO 30 K1=1,3
      DO 30 K2=1,3
      I=K+K1
      ST(K1,K2)=0.
      DO 30 K3=1,3
      J=L+K3
   30 ST(K1,K2)=ST(K1,K2)+XE(I,J)*T(K3,K2)
      DO 70 K1=1,3
      DO 70 K2=1,3
      I=K+K1
      J=L+K2
      IF (J-I) 40,40,70
   40 XG(I,J)=0.
      DO 50 K3=1,3
   50 XG(I,J)=XG(I,J)+T(K3,K1)*ST(K3,K2)
      IF (J-I) 60,70,70
   60 XG(J,I)=XG(I,J)
   70 CONTINUE
   80 CONTINUE
C
      IF (ICHK.NE.1 .OR.IPS.NE.1) GO TO 90
      CALL PMATE (XG,1,1,6,6)
   90    CONTINUE
         RETURN
 2000 FORMAT (20X,'ETR KE2D6',4X,2F12.3)
         END
C
         SUBROUTINE BMXYZ (ICHK,IMP,ISP,DX,DY,DZ,IP,IPS)
         COMMON/MPROP/XP(4,3)
         COMMON/SPROP/SP(20,5)
         COMMON/CESTM/XG(12,12)
         COMMON/GARB/S(12,12),ST(3,3),T(3,3),  E,EG,PR,A,XIYY,XIZZ,XJJ,
     1            TH,S1,DXY,DL,CT,  CB,C1,CA,SA,SB,  CTH,STH,
     2           Y0,Y1,Y2,Y3,Y4,  Z0,Z1,Z2,Z3,Z4,AAA0(409)
C
      IF (ICHK.EQ.1 .AND. IPS.EQ.1 ) WRITE (IP,2000) DX,DY,DZ
       E=XP(IMP,1)
      EG=XP(IMP,2)
      PR=XP(IMP,3)
       A=SP(ISP,1)
      XIYY=SP(ISP,2)
```

```
      XIZZ=SP(ISP,3)
      XJJ=SP(ISP,4)
      TH=SP(ISP,5)*3.14159265/180.
      IF (EG.EQ.0.) EG=0.5*E/(1.+PR)
      IF (XJJ.EQ.0.) XJJ=XIYY+XIZZ
      XP(IMP,2)=EG
      SP(ISP,4)=XJJ
      S1=DX*DX+DY*DY
      DXY=SQRT(S1)
      DL=S1+DZ*DZ
      DL=SQRT(DL)
         IF (DL.NE.0.) GO TO 10
         WRITE (IP,2010)
         STOP
   10 CA=E*A/DL
      CT=EG*XJJ/DL
      CB=E/(DL*DL)
      YO=XIYY*CB
      Y1=12.*YO/DL
      Y2= 6.*YO
      Y3= 2.*YO*DL
      Y4= 2.*Y3
      ZO=XIZZ*CB
      Z1=12.*ZO/DL
      Z2= 6.*ZO
      Z3= 2.*ZO*DL
      Z4= 2.*Z3
      IF (ICHK.NE.1 .OR. IPS.NE.1) GO TO 20
      WRITE (IP,2020) (SP(ISP,I),I=1,5)
      WRITE (IP,2020) (XP(IMP,I),I=1,3)
      WRITE (IP,2020) DX,DY,DZ,DXY,DL
      WRITE (IP,2020) YO,Y1,Y2,Y3,Y4
      WRITE (IP,2020) ZO,Z1,Z2,Z3,Z4
      WRITE (IP,2020) CA,CT
   20 DO 30 J=1,11
      DO 30 I=J+1,12
   30 S(I,J)=0.
      S(1,1)= CA
      S(7,1)=-CA
      S(7,7)= CA
      S(4,4)= CT
      S(10,4)=-CT
      S(10,10)= CT
      S( 2, 2)= Z1
      S( 8, 2)=-Z1
      S( 8, 8)= Z1
      S( 6, 2)= Z2
      S(12, 2)= Z2
      S( 8, 6)=-Z2
```

```
      S(12, 8)=-Z2
      S( 6, 6)=  Z4
      S(12,12)=  Z4
      S(12, 6)=  Z3
      S( 3, 3)= Y1
      S( 9, 3)=-Y1
      S( 9, 9)= Y1
      S(11, 9)= Y2
      S( 9, 5)= Y2
      S( 5, 3)=-Y2
      S(11, 3)=-Y2
      S( 5, 5)=  Y4
      S(11,11)=  Y4
      S(11, 5)=  Y3
      DO 40 J=2,12
      DO 40 I=1,J-1
   40 S(I,J)=S(J,I)
      CALL ZERO (T(1,1),T(3,3))
      IF (TH.NE.O.) GO TO 80
      IF (DY.EQ.O. .AND. DZ.EQ.O.) GO TO 50
      IF (DX.EQ.O. .AND. DZ.EQ.O.) GO TO 60
      IF (DX.EQ.O. .AND. DY.EQ.O.) GO TO 70
      GO TO 80
   50 C1=DX/DL
      T(1,1)=C1
      T(2,2)=C1
      T(3,3)= 1.
      GO TO 100
   60 C1=DY/DL
      T(1,2)= C1
      T(2,1)=-C1
      T(3,3)=  1.
      GO TO 100
   70 C1=DZ/DL
      T(1,3)= C1
      T(2,2)=  1.
      T(3,1)=-C1
      GO TO 100
   80 SB= DZ/DL
      CTH=COS(TH)
      STH=SIN(TH)
      IF (DXY.EQ.O.) GO TO 90
      CB=DXY/DL
      CA=DX/DXY
      SA=DY/DXY
      T(1,1)=CA*CB
      T(1,2)=SA*CB
      T(1,3)=SB
      T(2,1)=-SA*CTH-CA*SB*STH
```

```
      T(2,2)= CA*CTH-SA*SB*STH
      T(2,3)=+STH*CB
      T(3,1)= SA*STH-CA*SB*CTH
      T(3,2)=-CA*STH-SA*SB*CTH
      T(3,3)= CB*CTH
      GO TO 100
   90 T(1,3)= SB
      T(2,1)=-STH*SB
      T(2,2)= CTH
      T(3,1)=-CTH*SB
      T(3,2)=-STH
  100 DO 160 M=1,4
      DO 160 N=1,M
      K=(M-1)*3
      L=(N-1)*3
      DO 110 K1=1,3
      DO 110 K2=1,3
      I=K+K1
      ST(K1,K2)=0.
      DO 110 K3=1,3
      J= L+K3
  110 ST(K1,K2)=ST(K1,K2)+S(I,J)*T(K3,K2)
      DO 150 K1=1,3
      DO 150 K2=1,3
      I=K+K1
      J=L+K2
      IF (J-I) 120,120,150
  120 XG(I,J)=0.
      DO 130 K3=1,3
  130 XG(I,J)=XG(I,J)+T(K3,K1)*ST(K3,K2)
      IF (J-I) 140,150,150
  140 XG(J,I)=XG(I,J)
  150 CONTINUE
  160 CONTINUE
      IF (ICHK.NE.1 .OR. IPS.NE.1) GO TO 170
      WRITE (IP,2030)
      CALL PMATE (S,1,1,12,12)
      CALL PMATE (T, 1,1,3,3)
      CALL PMATE (XG,1,1,12,12)
  170 CONTINUE
         RETURN
 2000 FORMAT(/,3X, 'ETR BMXYZ',4X,3F12.3/)
 2010 FORMAT (5X,'DL=0.')
 2020 FORMAT (12F14.5)
 2030 FORMAT (/,5X,'S,T AND XG ...')
         END
```

BIBLIOGRAPHY

1. International Conference of Building Officials, Uniform Building Code, Whittier, California, 1979.

2. Applied Technology Council, Structural Engineers Association of California, Tentative Provisions for the Development of Seismic Regulations for Buildings, June 1978.

3. Guyan, R. J., ''Reduction of Stiffness and Mass Matricies,'' AIAA J., No. 3(2), Feb 1965, pp. 380.

4. Hurty, W. C., ''Vibration of Structural Systems by Component Mode Synthesis,'' Journal of the Engineering Mechanics Division, ASCE, No. , Aug 1960, pp. 51-69.

5. Craig, R.R., Jr., ''Methods of Component Mode Synthesis,'' The Shock and Vibration Digest , November 1977, pp. 3-10, The Shock and Vibration Information Center, Naval Research Laboratory, Washington, D. C.

6. Noor, A. K., Kamel, H. A. and Fulton, R. E., ''Substructuring Techniques - Status and Projections,'' Computers and Structures, Vol. 8, 1978, pp. 1313-1319.

7. Nelson, F. C., ''A Review of Substructuring Analysis of Vibrating Systems,'' The Shock and Vibration Digest, May 1980, .

8. Meirovitch, L., Computational Methods in Structural Dynamics, Noordhoff, Rockville, Maryland, 1980.

9. Gladwell, G. M. L., ''Branch Mode Analysis of Vibrating Systems,'' J. Sound Vibration, No. 1, Jan 1964, pp. 41-59.

10. Goldman, R. L., ''Vibration Analysis by Dynamic Partitioning,'' AIAA J., Vol. 7, No. 6, June 1969, pp. 1152-1154.

11. Klosterman, A. L., On the Experimental Determination and Use of Modal Representations of Dynamic Characteristics, PhD dissertation, Dept. Mech. Engr., University of Cincinnati, 1971.

12. Cromer, J.C. and Lalanne, M., ''Dynamic Behavior of Complex Structures, Using Part Experiment, Part Theory,'' Shock Vibration Bulletin, U.S. Naval Research Lab., No. 46-5, Aug 1976, .

13. Jezequel, L., ''A Method of damping Synthesis from Substructure Tests,'' ASME Paper, No. 79-DET-11, 1979, .

14. McAleese, J. D., ''Method for Determining Normal Modes and

Frequencies of a Launch Vechicle Utilizing its Component Normal Modes,'' NASA, No. TN D-4550, 1968, .

15. Grimes, P. J., ''Advancements in Structural Dynamic Technology Resulting from Saturn V Programs,'' NASA, Vol. II, No. CR-1540, 1970, .

16. Seaholm, N. A. and Sodergren, J.H., ''Substructuring Techniques and Their Impact on Dynamic Test Requirements,'' Symposium Substructure Testing and Synthesis, 1972, , NASA Marshall Space Flight Center

17. Fralich, R. W., Green, C. E. and Rheinfurth, M. H., ''Dynamic Analysis for Shuttle Design Verification,'' NASA Space Shuttle Technology Conference, No. NASA TM X-2570, 1972, .

18. Zalesak, J., ''Modal Coupling Procedures Adapted to NASTRAN Analysis of the 1:8-Scale Shuttle Structural Dynamics Model,'' NASA Tech. Report, Vol. I-II, No. NASA CR-132666 and CR-132667, 1975, , Grumman Aerospace Corp., Bethpage, N. Y.

19. Case, W. R., ''Dynamic Substructure Analysis of the International Ultraviolet Explorer (IUE) Spacecraft,'' NASTRAN: Users' Experiences, No. NASA TM X-3278, 2 1975, pp. 221-248.

20. Kuhar, E. J., Jr., ''Vibration analysis of the BSE Spacecraft Using Modal Synthesis and the Dynamic Transformation,'' Shock Vibration Bulletin, U.S. Navy Research Lab., Vol. 46, No. 5, 1976, pp. 231-238.

21. Gubser, J. L. and Zara, J. A., ''Component Mode Analysis of the Harpoon Missile, a Comparison of Analytical and Test Results,'' ASME Paper, No. 75-WA-Aero-6, 1975, .

22. Wada, B. K. and Garba, J. A., ''Dynamic Analysis and Test Results of Viking Orbitor,'' ASME Paper, No. 75-WA-Aero-7, 1975, .

23. Klosterman, A. L., ''A Combined Experimental and Analytical Procedure for Improving Automotive System Dynamics,'' SAE Paper, No. 720093, 1972, .

24. Klosterman, A. L. and McClelland, W. A., ''Combined Experimental and Analytical Techniques for Dynamic System analysis,'' 1973 Tokyo Seminar Finete Element Analysis, Japan, 1973, .

25. Bronowicki, A. and Hasselman, T. K., ''DYNALIST II: A Computer Program for Stability and Dynamic Response Analysis of Rail Vechicle Systems,'' DOT Rept. FRA-OR&D-75-22.IV, July 1976, .

26. Srinivasan, A. V., Leonberger, S. R., and Brown, K. W., ''Dynamic Analysis of an Assembly of Shrouded Blades Using Component Modes,'' J. Mechanical Design, Trans. ASME, No. 100, 2 1978, pp. 520-527.

27. Perman, A. B. and Schaeffer, M. T., ''The Application of Component Mode Synthesis to Covered Group of Blades,'' ASME Paper 79-DET-92, 1979, .

28. Glasgow, D. A. and Nelson, H. P., ''Stability Analysis of Rotor-Bearing Systems Using Component Mode Synthesis,'' ASME Paper, No. 79-DET-92, 1979, .

29. Singh, A. K., ''Dynamic Analysis Using Modal Synthesis,'' ASCE J. of Power Division, April 1978, pp. 131-140.

30. Lee, T. H., ''Axisymmetric Seismic Response of a Thick Circular Plate Supporting Many Rods by Modal Synthesis,'' International Journal of Earthquake Engineering and Structural Dynamics, No. 7, 1979, pp. 235-251.

31. Gutierrez, J. A. and Chopra, A. K., ''A Substructure method for Earthquake Analysis of Structures including Structure-Soil Interactions,'' International Journal of Earthquake Engineering and Structural Dynamics, No. 6, 1978, pp. 51-69.

32. Warburton, G. B. and Clement, J. F., ''Some Recent Developments in the Dynamics of Machine Foundations,'' Dynamics of Machine Foundations, Buzdugan, G. and Rades, M. , eds., Polytechnic of Bucharest, 1973, 44th Euromech Colloquium

33. Kukreti, A. R. and Feng, C. C., ''Dynamic Substructuring for Alternating Subsystems,'' Journal of the Engineering Mechanics Division, American Society of Civil Engineers, Vol. 104, No. EM5, October 1978, pp. 1113-1129.

34. Meirovitch, L., Analytical Methods in Vibrations, The MacMillan Company, New York, 1967.

35. Craig, Jr., R. R., Structural Dynamics, John Wiley & Sons, New York, 1981.