USER'S Manual
for

QUAD4M

A COMPUTER PROGRAM TO EVALUATE THE SEISMIC RESPONSE OF SOIL STRUCTURES USING FINITE ELEMENT PROCEDURES AND INCORPORATING A COMPLIANT BASE

by

MARTIN HUDSON
I. M. IDRISS
MOHSEN BEIKAE

Sponsored by

The National Science Foundation
Washington, D. C.

Center for Geotechnical Modeling
Department of Civil & Environmental Engineering
University of California
Davis, California

May 1994

USER'S Manual
for

QUAD4M

A COMPUTER PROGRAM TO EVALUATE THE SEISMIC RESPONSE OF SOIL STRUCTURES USING FINITE ELEMENT PROCEDURES AND INCORPORATING A COMPLIANT BASE

by

MARTIN HUDSON
I. M. IDRISS
MOHSEN BEIKAE

Sponsored by

The National Science Foundation
Washington, D. C.

Center for Geotechnical Modeling
Department of Civil & Environmental Engineering
University of California
Davis, California

May 1994

Table of Contents

QUAD4M

A COMPUTER PROGRAM TO EVALUATE THE SEISMIC RESPONSE OF SOIL STRUCTURES USING FINITE ELEMENT PROCEDURES AND INCORPORATING A COMPLIANT BASE

Martin Hudson[1], I.M. Idriss[2], and Mohsen Beikae[3]

Abstract

QUAD4M, a dynamic, time domain, equivalent linear two-dimensional computer program, was written as a modification of **QUAD4** to implement a transmitting base, an improved time-stepping algorithm, seismic coefficient calculations, a restart capability, a change in the algorithm by which damping is set, and various computational enhancements to fully bring the program into the environment of the microcomputer. Various sample problems were run to verify the processing of the program, and the results are presented. The results compare well with **SHAKE91**, a one-dimensional closed-form solution program.

Introduction

The finite element method of analysis is a widely used computational procedure for the solution of problems in continuum mechanics, as well as many other fields. The procedure has been found very powerful for modelling the seismic response of soil deposits and earth structures. Programs to solve such response have been written using time domain solutions as well as frequency domain solutions in the past 30 years.

---

[1] Project Engineer, Law/Crandall, Los Angeles, California; formerly graduate student, University of California, Davis

[2] Professor, Department of Civil Engineering and Environmental Engineering, University of California, Davis

[3] Senior Engineer, Woodward-Clyde Consultants, Santa Ana, California

QUAD4 (Idriss, Lysmer, Hwang, and Seed, 1973) was written as a two-dimensional, time domain solution to dynamic soil response. It incorporated for the first time independent damping in each element in the continuum.

QUAD4M incorporates into **QUAD4** a transmitting base so that the half-space beneath a mesh can be modeled and the need to assume a rigid foundation can be eliminated. The shear and compression wave velocities and the unit weight for the material underlying the mesh can be entered, and the response of the mesh on top of that half-space can now be modeled with greater accuracy.

In addition, seismic coefficients have been added in this version of the program. This feature is particularly useful in deformation analyses. The program also has a restart capability. The acceleration, velocity, and displacement are stored for the restart so that the program continues as if no interruption had occured. This feature is useful for changing material properties during the shaking event.

Finally, **QUAD4M** incorporates a new method for the formulation of damping matrices which results in a significant reduction of the damping of higher frequencies commonly associated with the use of a Rayleigh damping formulation.

Evaluation of seismic response using the finite element procedure

The finite element procedure has been used extensively over the past 30 years for estimating the response of soil structures or deposits to static and dynamic loading conditions. It consists of numerically modeling a continuum with a finite number of elements interconnected at their common nodes.

The finite element procedure uses a system of equations represented in matrix form as:

$$[M]\underline{\ddot{u}} + [C]\underline{\dot{u}} + [K]\underline{u} = \underline{R} \qquad (1)$$

where:

$[M]$	mass matrix (in this case using the assumption of a lumped mass formulation);
$[C]$	damping matrix;
$[K]$	stiffness matrix;
$\underline{R}$	load vector, which is given by: $\underline{R} = [M]\underline{\ddot{u}}_g$
$\underline{u}$	relative displacement vector; and dots represent differentiation with respect to time;
$\underline{\ddot{u}}_g$	outcrop acceleration.

As was originally constructed in QUAD4, the damping matrix is formulated using the assemblage of element damping matrices constructed using the Rayleigh formulation (1945):

$$[C]_q = \alpha_q[M]_q + \beta_q[K]_q \qquad (2)$$

for each element q. The use of strain compatible damping at the element level was first introduced in Idriss, Lysmer, Hwang, and Seed (1973). The values of α_q and β_q are chosen as described in the section on damping.

The entire solution is iterated upon the number of times specified by the user, in order to obtain strain compatible damping and modulus values.

To solve equation (1), it is necessary to introduce equations relating $\underline{\ddot{u}}$, $\underline{\dot{u}}$, and $\underline{u}$. The Newmark family of methods (e.g. Hughes, 1987) uses the following equations to fulfill the above requirement:

$$
\begin{aligned}
\underline{\dot{u}_N} &= \underline{\dot{u}_{N-1}} + \Delta t[(1-\gamma)\underline{\ddot{u}_{N-1}} + \gamma\underline{\ddot{u}_N}] \\
\underline{u_N} &= \underline{u_{N-1}} + \Delta t\underline{\dot{u}_{N-1}} + \frac{\Delta t^2}{2}[(1-2\beta)\underline{\ddot{u}_{N-1}} + 2\beta\underline{\ddot{u}_N}]
\end{aligned}
\tag{3}
$$

where N is the current time step (quantities unknown), and N-1 is the previous time step (quantities known). The use of equations (3) with $\gamma = 0.5$ and $\beta = 0.25$ is called the trapezoidal rule and provides a time-stepping algorithm with unconditional stability, quadratic convergence, and no numerical damping of any frequencies (Hughes, 1987).

Using the Trapezoidal rule, the following equations are obtained for solving the displacement, velocity, and acceleration at each time step:

$$\underline{u_{N+1}} = [\bar{K}]^{-1}[\bar{R}]_{N+1} \tag{4a}$$

$$\underline{\ddot{u}_{N+1}} = \frac{4}{\Delta t^2}\left(\underline{u_{N+1}}-\underline{u_N}\right) - \frac{4}{\Delta t}\underline{\dot{u}_N} - \underline{\ddot{u}_N} \tag{4b}$$

$$\underline{\dot{u}_{N+1}} = \underline{\dot{u}_N} + \frac{\Delta t}{2}(\underline{\ddot{u}_N} + \underline{\ddot{u}_{N+1}}) \tag{4c}$$

$$[\bar{K}] = \frac{4}{\Delta t^2}[M] + \frac{2}{\Delta t}[C] + [K] \tag{4d}$$

$$[\bar{R}]_{N+1} = [R]_{N+1} + [M]\underline{A_{N+1}} + [C]\underline{B_{N+1}} + [K]\alpha\underline{u_N} \tag{4e}$$

$$\underline{A_{N+1}} = \frac{4}{\Delta t^2}\left(\underline{u_N} + \Delta t\underline{\dot{u}_N} + \frac{\Delta t^2}{4}\underline{\ddot{u}_N}\right) \tag{4f}$$

$$\underline{B_{N+1}} = \frac{2}{\Delta t}\underline{u_N} + \underline{\dot{u}_N} \tag{4g}$$

Transmitting Boundaries

In order for a two-dimensional finite mesh to represent the response of an infinite field condition, the artificial reflection of seismic waves from side boundaries, as well as from the underlying half-space, should be minimized. Lysmer and Kuhlemeyer (1969) introduced a simple procedure to accomplish this. They suggested the use of dampers as illustrated in figure 1 for the case of a vibrating footing. In the case of a soil mass subjected to earthquake vibrations, the implementation of a compliant base in **QUAD4M** is the same as the Lysmer and Kuhlemeyer scheme.

The implementation of these dampers involves adding damping at each of the nodes that make up the base and sides of the finite model. For the present study, only the base dampers have been implemented. The base dampers are more essential to incorporate than the side dampers because the finite element system under consideration will always be placed over a half-space. The effects of side boundaries can be readily minimized by increasing the extent of the finite element mesh.

To mathematically implement these dampers, the parts of the applicable element matrices have the transmitting boundary damping term added to the diagonal terms. This produces an adjustable force in the x and y direction proportional to the velocity of the specified nodes. The coefficients added on to the diagonal terms are obtained as:

Term for direction perpendicular to boundary: $\rho V_p L$
Term for direction parallel to boundary: $\rho V_s L$

The velocity of the P or S waves is used for the material in the half space below the finite element model, as is the density, ρ. The "tributary width" of the node, L is that length corresponding to half of the distance to the next node on both sides.

When a transmitting base is used, the input motion is a function of the material properties of the half-space below the mesh, and the properties and geometry of the mesh. This is the correct choice for a boundary condition when the input motion represents an outcrop acceleration, recorded at an outcrop of the half-space material. If an infinitely stiff ($V_s \rightarrow \infty$) rock is specified under the underlying stratum, then the input motion will not be affected by the mesh above.

Seismic Coefficient Computation

A seismic coefficient is the ratio of the force induced by an earthquake in a block of the mesh, over the weight of that block.

The forces acting on the block are computed by multiplying the shear and normal stresses acting on an element by the width of that element. Since the surface of the block is specified as going through the nodes (between elements) in **QUAD4M**, the average stress is found between the elements on either side of the interface. The summation of forces acting on a block is computed as a function of time. The seismic coefficient is then computed for each time step.

Restart Capability

A feature has been added in **QUAD4M** whereby at the conclusion of the calculations, the acceleration, velocity, and displacement of every node is saved. This can then be used to restart the program at the time step following the last time step used in the previous run of the program. Before the program is restarted, the soil properties can be changed The program can be stopped and restarted as many times as is desired during the course of an earthquake.

Damping

The damping matrix is formulated using the assemblage of element damping matrices constructed in this manner:

$$[C]_q = \alpha_q [M]_q + \beta_q [K]_q \qquad (5)$$

for each element q. The use of rayleigh damping in this manner results in a frequency dependent damping applied to the problem, with

$$\lambda_q = \frac{1}{2}\left(\frac{\alpha_q}{\omega} + \beta_q \times \omega\right) \qquad (6)$$

The damping in soil is not frequency dependent. Therefore, the choice of α_q and β_q must be made that provides for damping values that have minimum variations over the range of frequencies of interest. In **QUAD4**, the constants were chosen in such a way that the damping was minimized at the fundamental frequency of the entire finite element model, ω_1. The justification for this is that the first mode of vibration has the highest participation factor of all the modes. Using this criterion, the values of α_q and β_q are chosen as follows for each element:

$$\begin{aligned} \alpha_q &= \lambda_q \times \omega_1 \\ \beta_q &= \lambda_q \,/\, \omega_1 \end{aligned} \qquad (7)$$

As is the case with all procedures that utilize a Rayleigh Damping formulation, the higher frequencies are overdamped. Therefore, in **QUAD4M**, a new scheme for setting damping is employed. Instead of using a single frequency (the fundamental frequency of the model), and a slope (0 at the fundamental frequency of the model) to establish the constants in equation 6, two frequencies are used to establish these constants. The choice of these two frequencies has been studied using several different earthquakes and several different one-dimensional deposits. One-dimensional deposits were used because comparison with **SHAKE-91** (Idriss and Sun, 1992) (which uses a constant value of damping for all frequencies) can be made. One frequency is chosen at the fundamental frequency of the model, as in **QUAD4**. The second frequency is established as

$$\omega_2 = n\,\omega_1 \qquad (8)$$

where n is an odd integer. This choice was guided by the response of a shear beam in which the frequencies of higher modes are odd multiples of the frequency of the fundamental mode of the beam. The parameter n is chosen such that:

$$n = \textit{closest odd integer greater than } \omega_i / \omega_1 \qquad (9)$$

where ω_i is the predominant frequency of the input earthquake motion.

To set damping at two frequencies, the values of α_q and β_q are then given by the following expression for each element (Hudson, 1994):

$$\begin{aligned} \alpha_q &= 2\lambda_q \frac{\omega_1 \omega_2}{\omega_1 + \omega_2} \\ \beta_q &= 2\lambda_q \frac{1}{\omega_1 + \omega_2} \end{aligned} \qquad (10)$$

The use of this two-frequency scheme results in under-damping between ω_1 and ω_2, and over-damping outside that range. This scheme allows the model to respond to the predominant frequencies of the input motion without experiencing significant over-damping.

The element damping ratios, λ_q, are chosen based upon the average strain developed in the element. The value of ω_1, the fundamental frequency of the system, is internally calculated by solving the following system of equations:

$$K\underline{\phi}^1 = \omega_1^2 M \underline{\phi}^1 \qquad (11)$$

where the first mode shape is represented by ϕ^1.

The difference between the **QUAD4** and the **QUAD4M** damping schemes is illustrated using a 200 foot soil deposit with an average shear wave velocity, V_s = 1200 ft/sec, and a total unit weight, γ = 120 pcf. The modulus reduction and damping curves for sand were used for this example. The N-S component of the Santa Cruz record of the Loma Prieta earthquake was used as input rock outcrop motion.

The fundamental frequency of this 200-ft soil layer under small strain conditions is given by:

$$f_1 = \frac{V_s}{4H} = \frac{1200}{4\times200} = 1.5\,hz$$

The response spectrum for the input rock outcrop motion is shown in figure 3. The predominant period, T_i, of this motion is about 0.15 sec; the predominant frequency $f_i = 1/T_i \approx 6.7$ hz. Using Equation 8, the ratio $n = 6.7/1.5 = 4.1$, hence a value of $n = 5$ is used in the first iteration. For the last iteration, the strain compatible moduli lead to a fundamental period of 0.78 sec or $f_1 = 1.28$ hz. The ratio $f_i/f_1 = 5.23$; hence $n = 7$ is used in the last iteration and the variation of damping with frequency is shown in figure 2. Figure 4 illustrates the response spectrum obtained at the top of the layer using programs **SHAKE91**, **QUAD4M**, and **QUAD4**.

Implementation of Finite Element Model

The finite element computational procedure described previously was implemented by modifying program **QUAD4** as presented in Idriss, Lysmer, Hwang, and Seed (1973). The revisions consisted of changing the time-stepping algorithm, incorporating transmitting base elements, incorporating the calculation of seismic coefficients for selected potential sliding blocks, adding restart capability, modifying the damping computation, and modifying some computational procedures. All other features of **QUAD4** are still intact in the new program.

The time stepping method was changed from the Wilson-θ method to the trapezoidal rule as described previously.

The transmitting base elements were incorporated as described above, and the user must now specify the P and S wave velocities and the unit weight of the half-space below the base elements.

Seismic coefficients can now be computed. The user specifies elements within the block, and the nodes bounding the block.

The restart feature is utilized by specifying a switch in the input file, and providing a file name to which the restart input file is recorded. The restart input file echoes all the information in the original input file in addition to the displacement, velocity, and acceleration of each node at the end of the last time step. This input file can then be modified to continue the calculations with the previous properties or with a new set of properties.

The new damping scheme is controlled by specifying the predominant period of the input motion, obtained from the response spectrum. The output file records the two frequencies at which the damping is set.

Computational modifications were made to make the program conform to a structured Fortran language, implementing data structures to describe the elements and nodes. In addition, the arrays and matrices have been made allocatable in size, so that the program adjusts its memory usage according to the input problem. When the code is compiled for Microsoft Windows 3.x using the Microsoft Fortran 5.1 compiler, or using the Microsoft Powerstation compilers, the program can accept as large a problem as there is available memory on the microcomputer.

Finally, a few of the subroutines were rewritten to be easier to follow and to increase their computational efficiency.

The new code, called **QUAD4M,** is presented in Appendix A.

Examples and Comparisons with Other Solutions

100-ft Layer of Sand

A 100-ft layer of sand having a total unit weight, γ = 125 pcf, K_0 = 0.5, and K_{2max} = 65 was used, in which K_0 is the coefficient of earth pressure at rest, K_{2max} is a parameter relating maximum shear modulus, G_{max}, and effective confining pressure, σ'_m, by $G_{max}=1000(K_2)_{max}\sqrt{\sigma'_m}$. The effective confining pressure is given by $\sigma'_m=\frac{1+2K_0}{3}\sigma'_v$, in which σ'_v is the effective vertical pressure. Note that the values of σ'_v, σ'_m and G_{max} are in pounds per square foot in these equations. This sand layer was assumed to be underlain by a half space having a shear wave velocity of 3000 ft/sec, a compression wave velocity of 7350 ft/sec and a total unit weight of 135 pcf.

The variation of shear modulus and damping with shear strain used for evaluating the response of this layer are shown in Figure 6 and Table 1. These values were taken from the **SHAKE91** manual. The finite mesh used for the 100′ sand layer is shown in figure 5.

The case of a 100′ dense sand layer with the Santa Cruz record of the Loma Prieta earthquake scaled to 0.3g and 0.6g are shown in figures 7 through 10.

Figure 7 and 8 show the response spectra and time histories, respectively, obtained at the surface of the 100' layer when the Santa Cruz record of the Loma Prieta earthquake scaled to a peak acceleration of 0.3g is applied at the base. The **QUAD4M** and **SHAKE91** results are compared in these figures. Figures 9 and 10 show the response spectra and time histories, respectively, obtained in the same manner, but using the input acceleration time history scaled to 0.6g.

Figures 11 and 12 show the variation in peak horizontal acceleration and maximum shear stress with depth for the case of the 0.3g input and the 0.6g input, respectively. These figures also show the results obtained using both **QUAD4M** and **SHAKE91**.

Table 1:
Dynamic Soil Properties for Sand

Shear Strain (%)	G/Gmax (%)	Damping (%)
.0001	100	.24
.0003	100	.42
.001	99	.8
.003	96	1.4
.01	85	2.8
.03	64	5.1
.1	37	9.8
.3	18	15.5
1	8	21

These comparisons show that the **QUAD4M** and **SHAKE91** provide almost identical results for maximum shear stresses and for peak horizontal accelerations. Differences between the results of the two programs arise because of the use of the Rayleigh formulation for damping in **QUAD4M.**

Example of a Seismic Coefficient Calculation

A second analysis was performed to illustrate the seismic coefficient capabilities of **QUAD4M.** This problem consists of a 50 foot embankment underlain by 50 feet of soil. The embankment is shown in figure 13. The Young's modulus used is 1 x 10^6, the unit weight is 120 pcf, the Poison's ratio is 0.45, and the shear wave velocity is hence 304 fps.

In this example, twelve surfaces are chosen with various depths and extents into the embankment. The surfaces are shown in figure 14. The maximum seismic coefficient is shown for those twelve surfaces in figure 15. It can be seen that the surfaces extending farther into the embankment approximate more closely the semi-infinite solution. This indicates that the seismic coefficients are correctly converging to the free field solution.

Summary

QUAD4 has been updated to include the trapezoidal rule, transmitting boundaries and sliding block seismic coefficients have been added, restart capability has been introduced, the damping formulation has been changed, and various computational changes have been performed to update the program to use on a personal computer.

It is hoped that this new version of **QUAD4** will provide improved means for calculating the response of soil deposits and soil structures during earthquakes using a time-domain method of analysis.

Acknowledgments

The work reported in this report was supported by Grant Number NSF BCS-90-11322 "I" from the National Science Foundation; the contract monitor was Dr. Cliff Astil. This support is gratefully acknowledged. Our thanks are also extended to Professor Karl Romstad of the University of California at Davis, for valuable discussions regarding the use of the Rayleigh damping formulation.

References

Hudson, Martin B (1994). *Behavior of Slopes and Earth Dams During Earthquakes.* Doctoral Thesis, University of California, Davis.

Hughes, Thomas J.R. (1987). The Finite Element Method. London: Prentice Hall.

Idriss, I.M., J. Lysmer, R. Hwang, and H. Bolton Seed (1973). *Quad-4: A Computer Program for Evaluating the Seismic Response of Soil Structures by Variable Damping Finite Element Procedures.* EERC Report 73-16. Berkeley: University of California, Berkeley.

Idriss, I. M. and Joseph I Sun (1992). *"A Computer Program for Conducting Equivalent Linear Seismic Response Analyses of Horizontally Layered Soil Deposits" - Program Modified based on the Original SHAKE program published in December 1972 by Schnabel, Lysmer & Seed.* Center for Geotechnical Modeling, Department of Civil & Environmental Engineering, University of California, Davis, California.

Lysmer, John, and Roger L. Kuhlemeyer (1969). "Finite Dynamic Model for Infinite Media" *Journal of the Engineering Mechanics Division, Proceedings of the American Society of Civil Engineers.* EM4:859-877.

Newmark, N.M. (1965). "Effects of Earthquakes on Dams and Embankments" *Geotechnique*. XV:139-160.

Rayleigh, L. (1945). *Theory of Sound, Vol. I.* New York: Dover Publication.

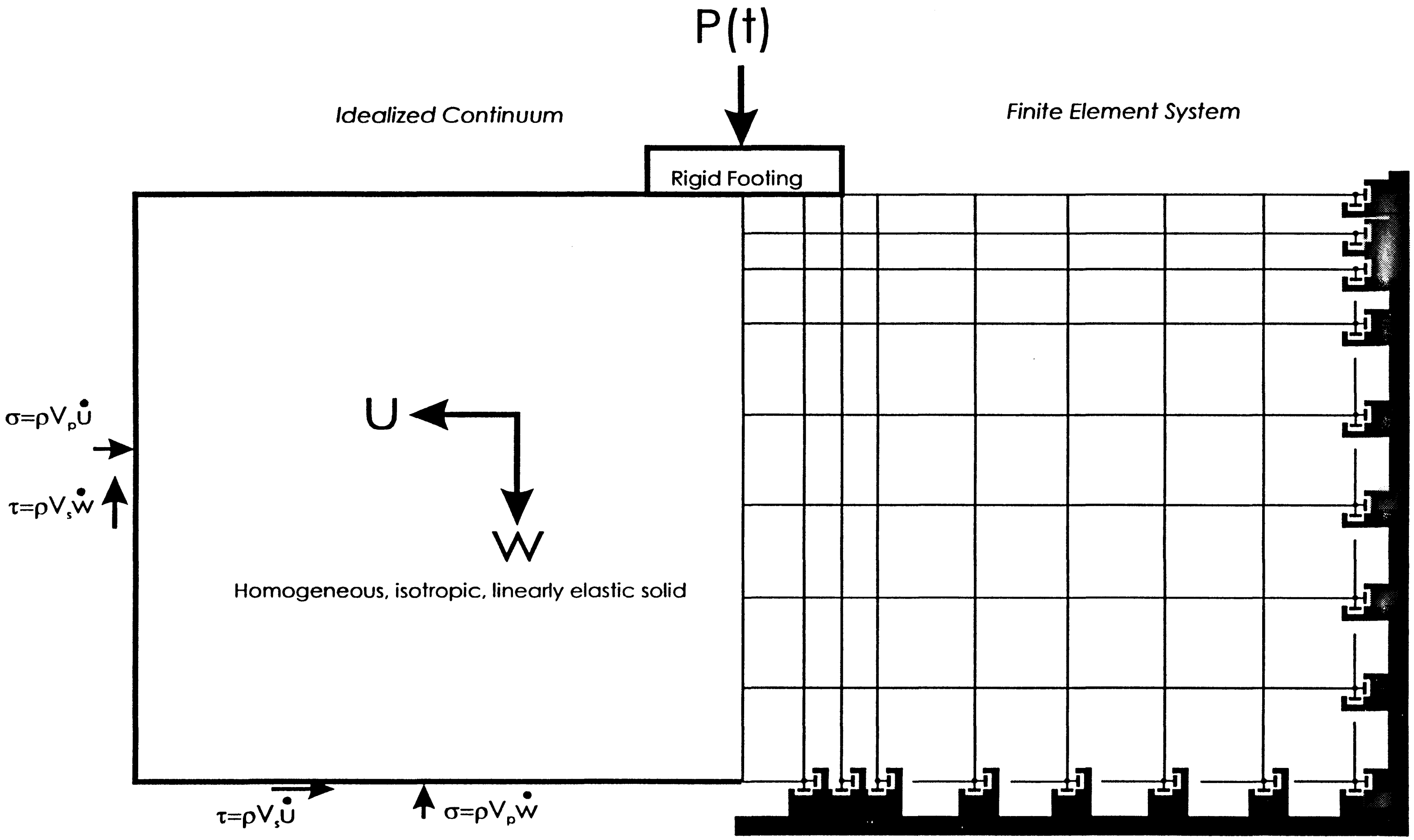


Figure 1
Finite Models for Footing on Half Space
(after Lysmer & Kuhlemeyer, 1969)

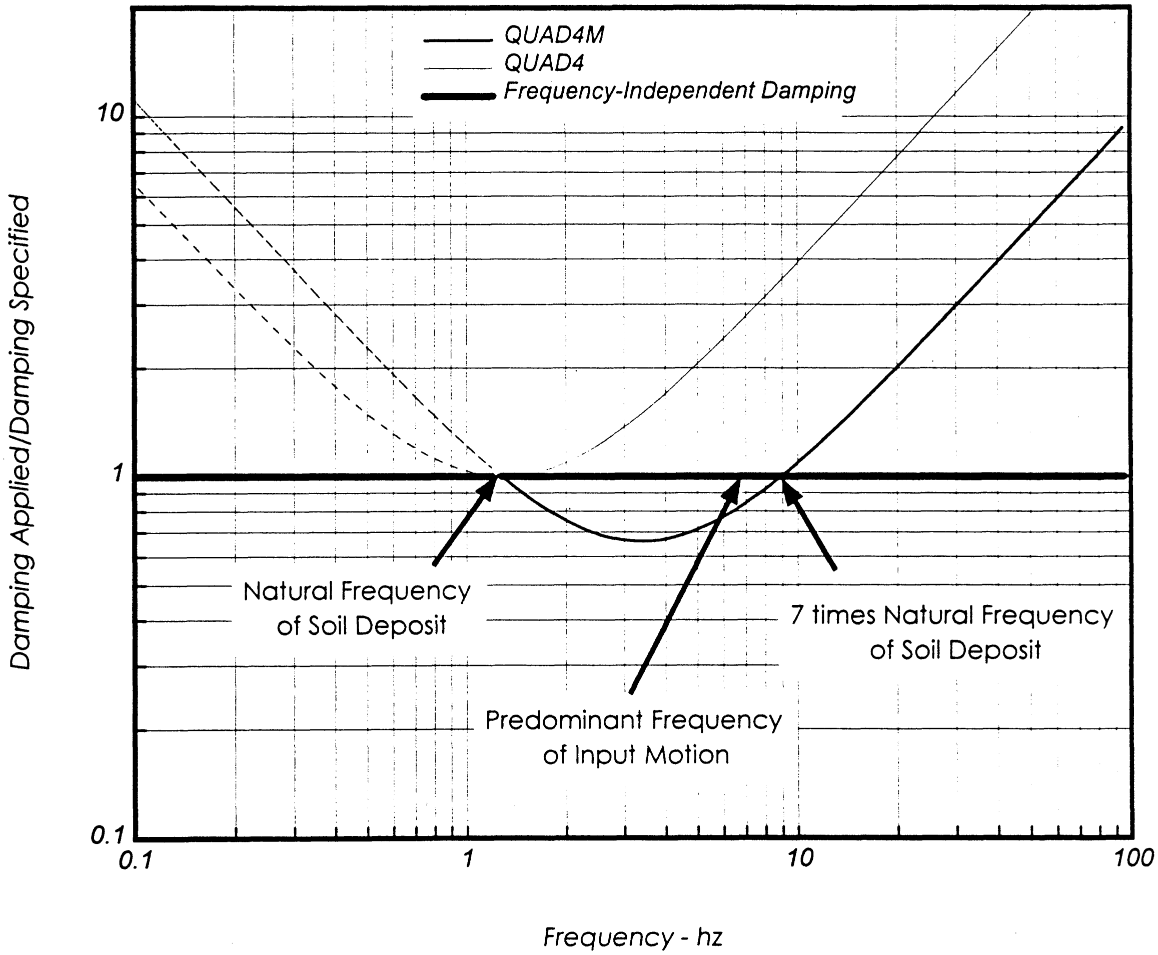


Figure 2
Variation of Damping with Frequency
200 foot sand deposit
Santa Cruz Record of Loma Prieta Earthquake

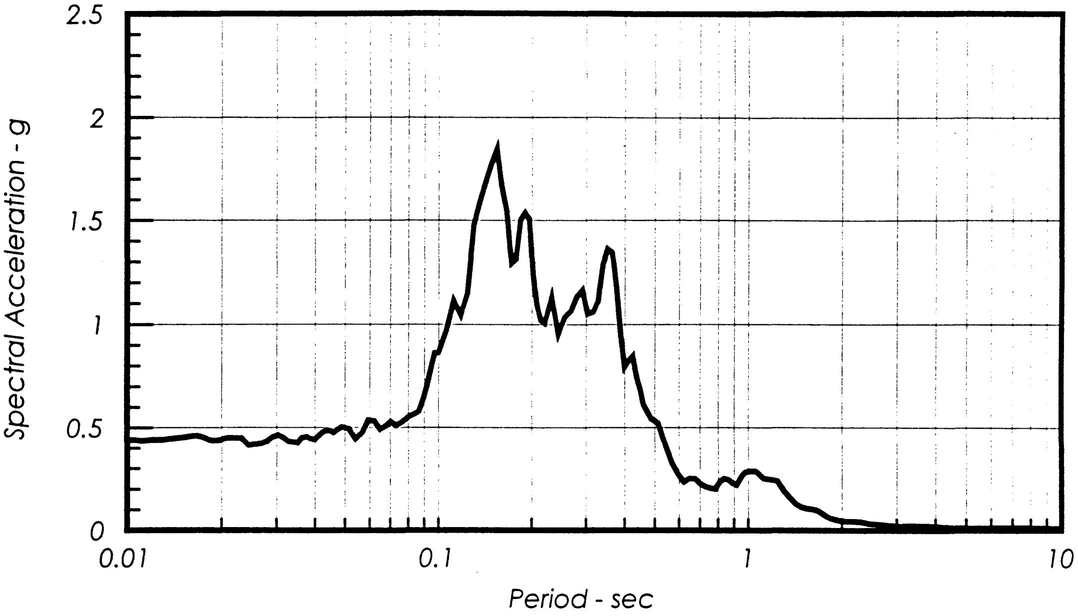


Figure 3
Response Spectrum for Earthquake Ground Motion used as
Input Rock Outcrop Motion in Sample Profile
Santa Cruz Record, 0 degree component,
Loma Prieta Earthquake

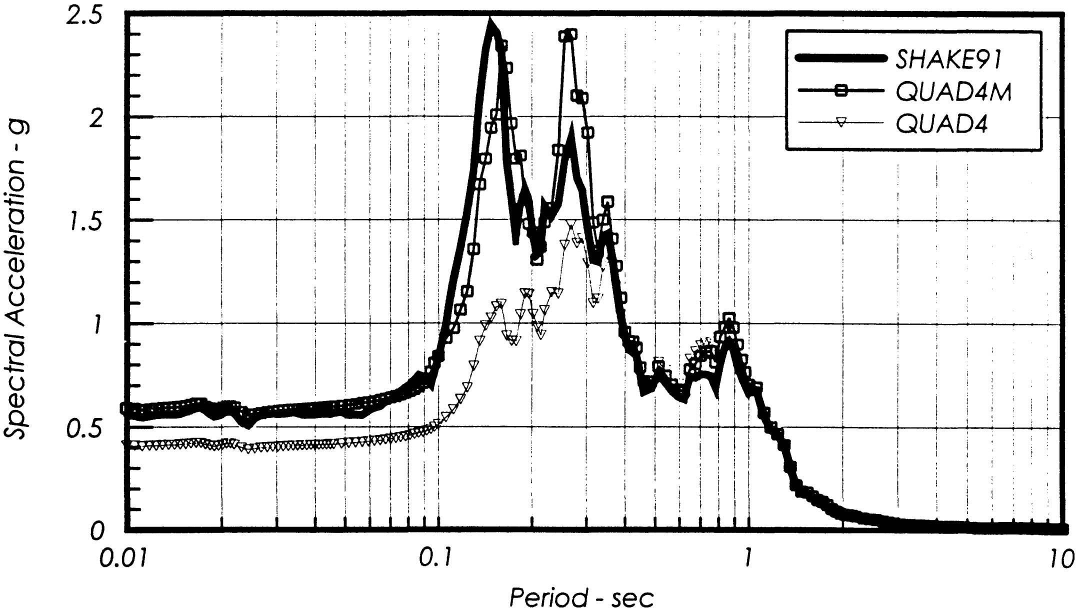


Figure 4
Computed Response
Spectral Ordinates at Ground Surface of Sample Profile
Using Various Damping Schemes

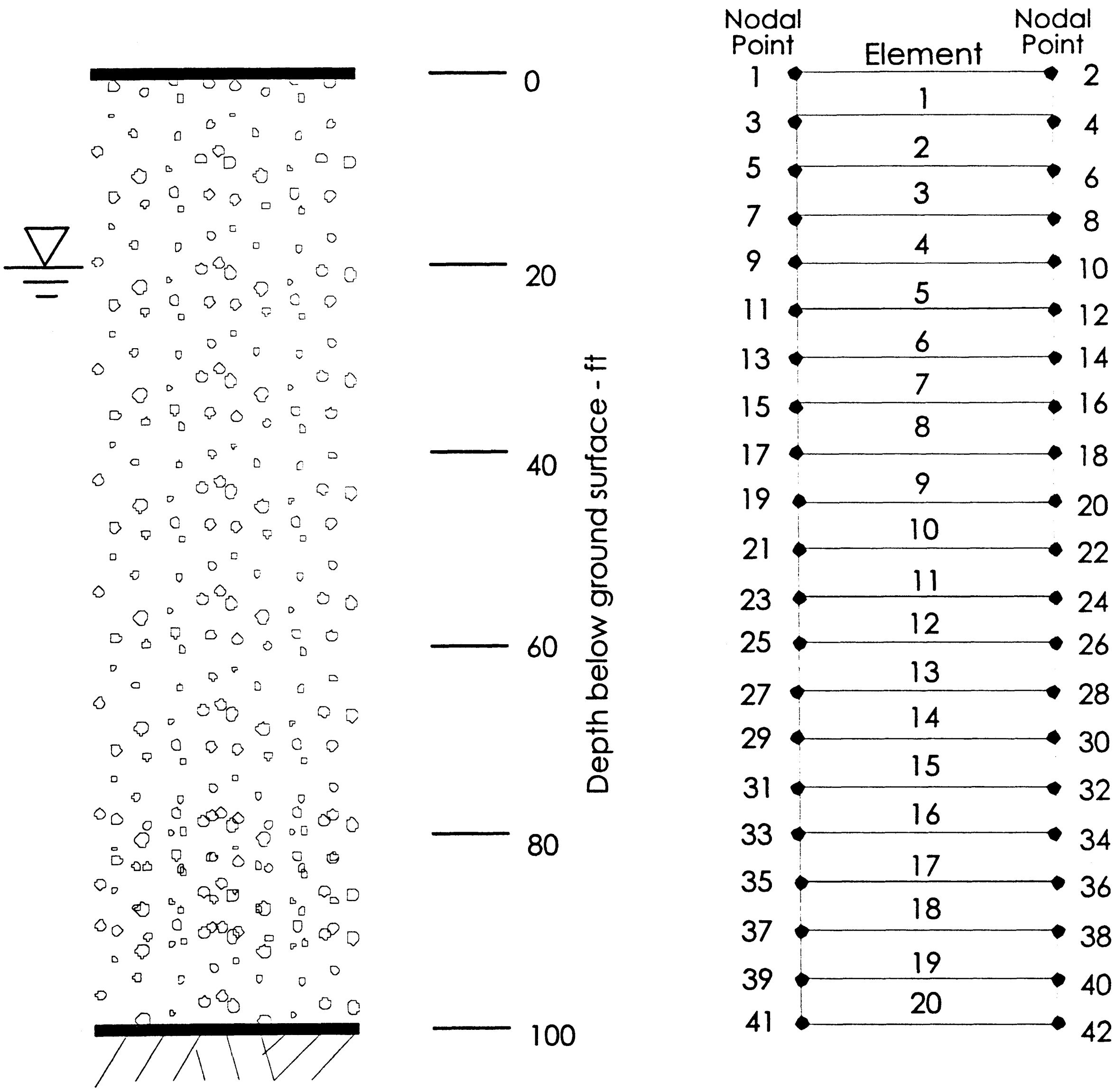


Figure 5
Finite Element Mesh Used for 100-ft Layer of Sand

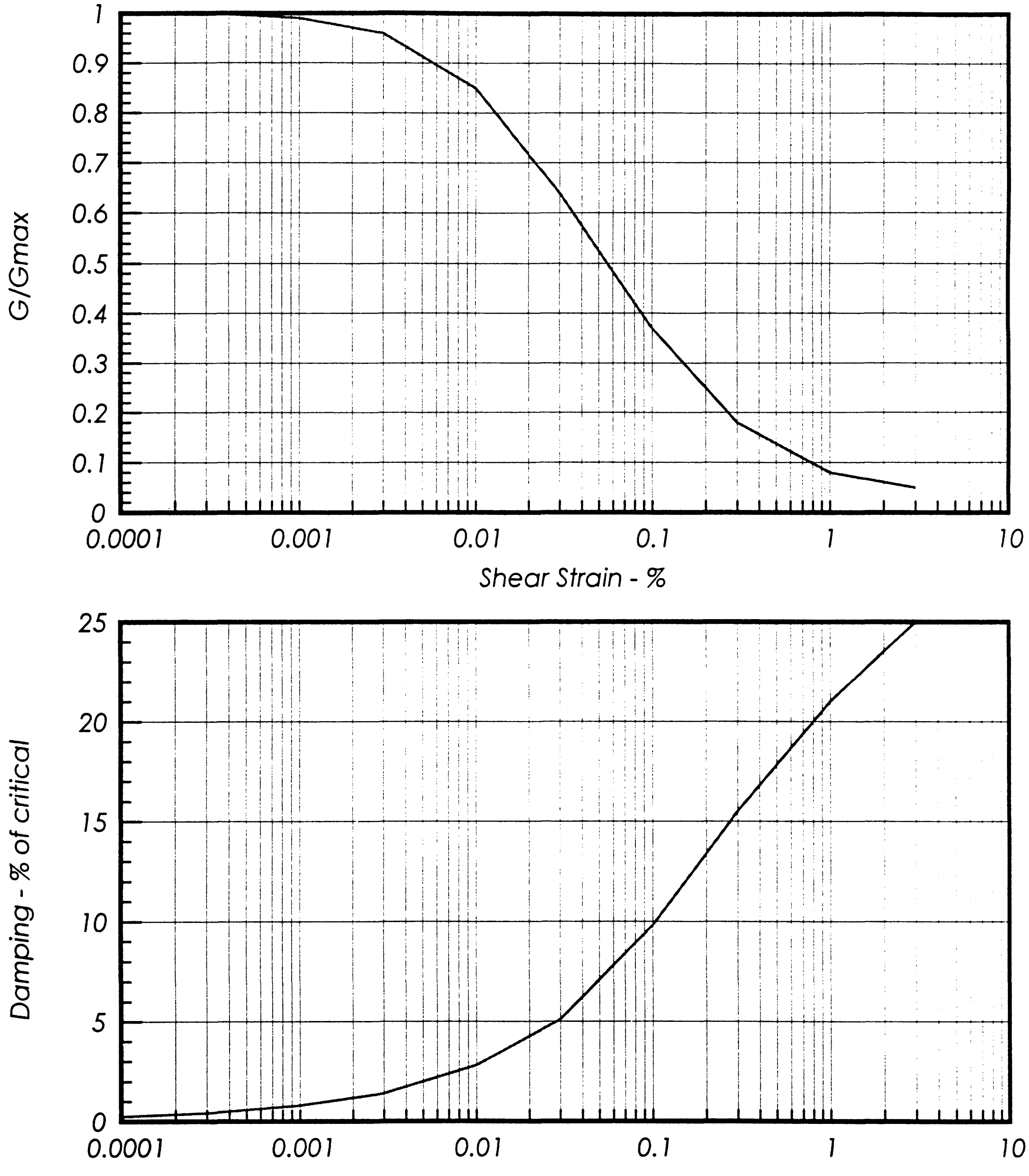


Figure 6
Dynamic Soil Properties for Sand

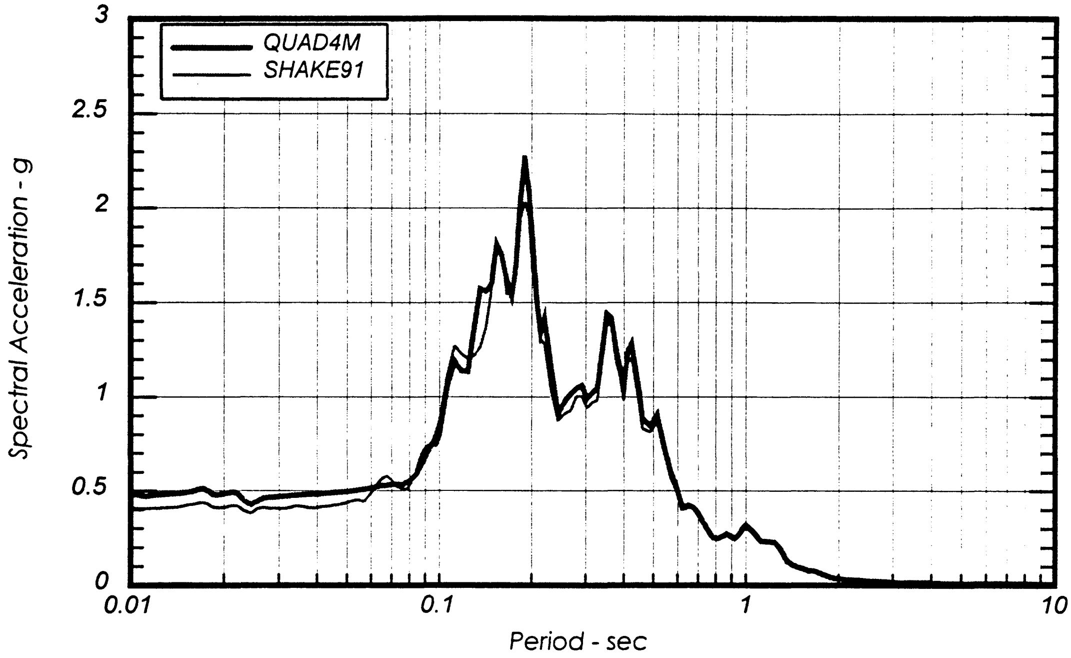


Figure 7
Comparison of Response Spectra at Ground Surface
Using Programs QUAD4M and SHAKE91
for the 100 Foot Dense Sand Layer --
Input Motion: Santa Cruz Record Scaled to 0.3g

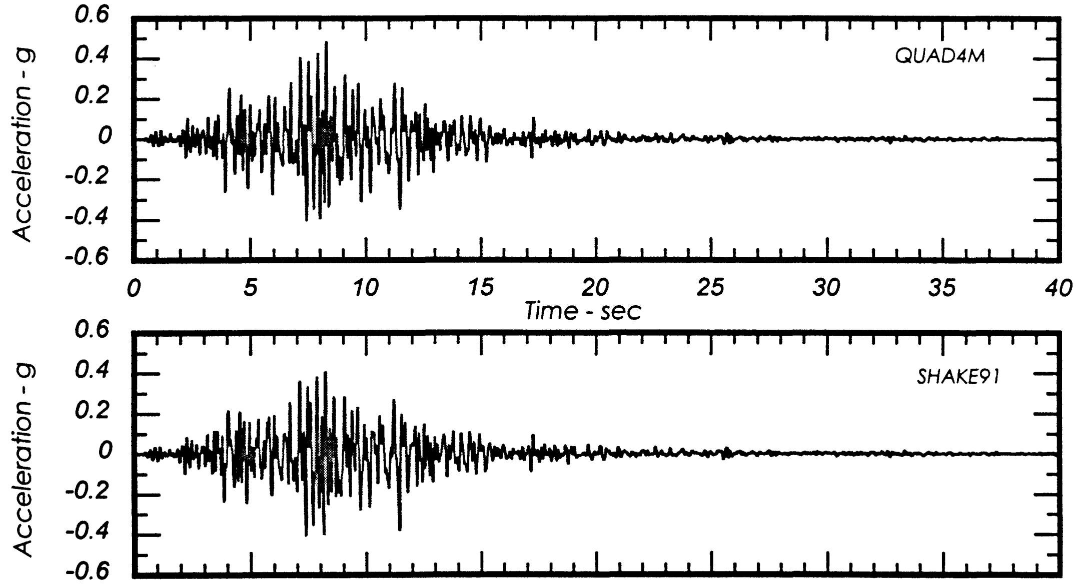


Figure 8
Comparison of Time Histories at Ground Surface
Computed Using QUAD4M and SHAKE91
for the 100 Foot Dense Sand Layer --
Input Motion: Santa Cruz Record Scaled to 0.3g

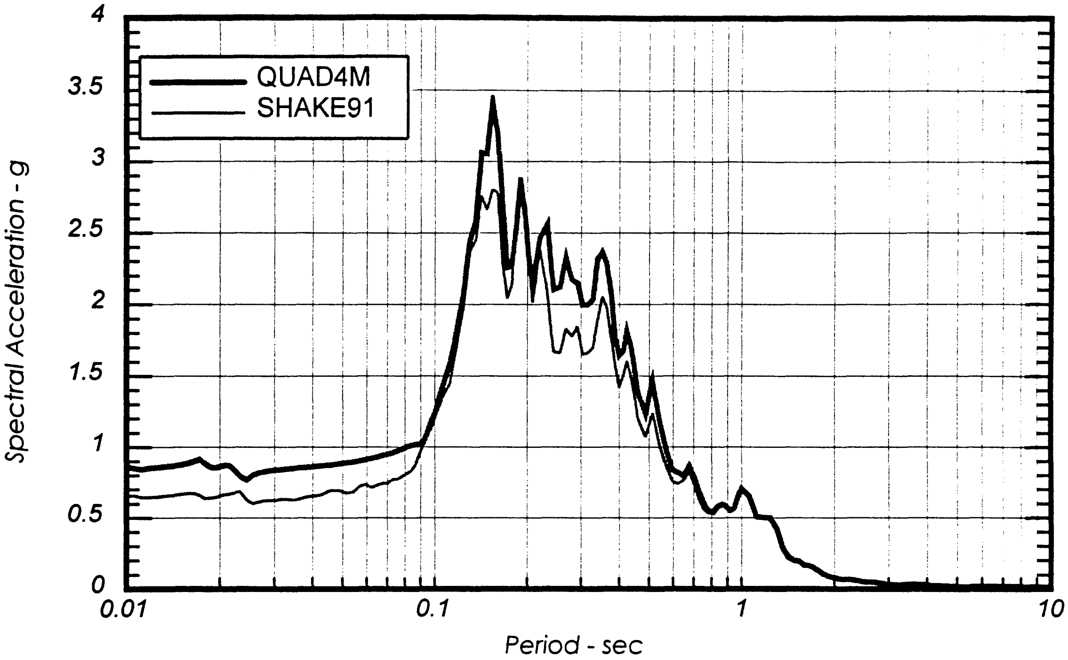


Figure 9
Comparison of Response Spectra at Ground Surface
Using Programs QUAD4M and SHAKE91
for the 100 Foot Dense Sand Layer --
Input Motion: Santa Cruz Record Scaled to 0.6g

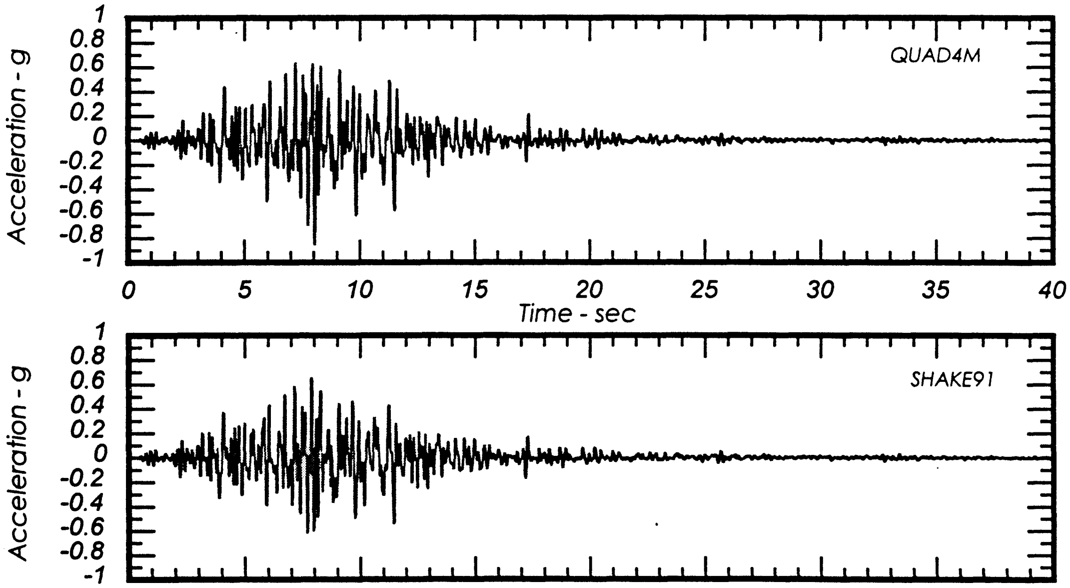


Figure 10
Comparison of Time Histories at Ground Surface
Using Programs QUAD4M and SHAKE91
for the 100 Foot Dense Sand Layer --
Input Motion: Santa Cruz Record Scaled to 0.6g

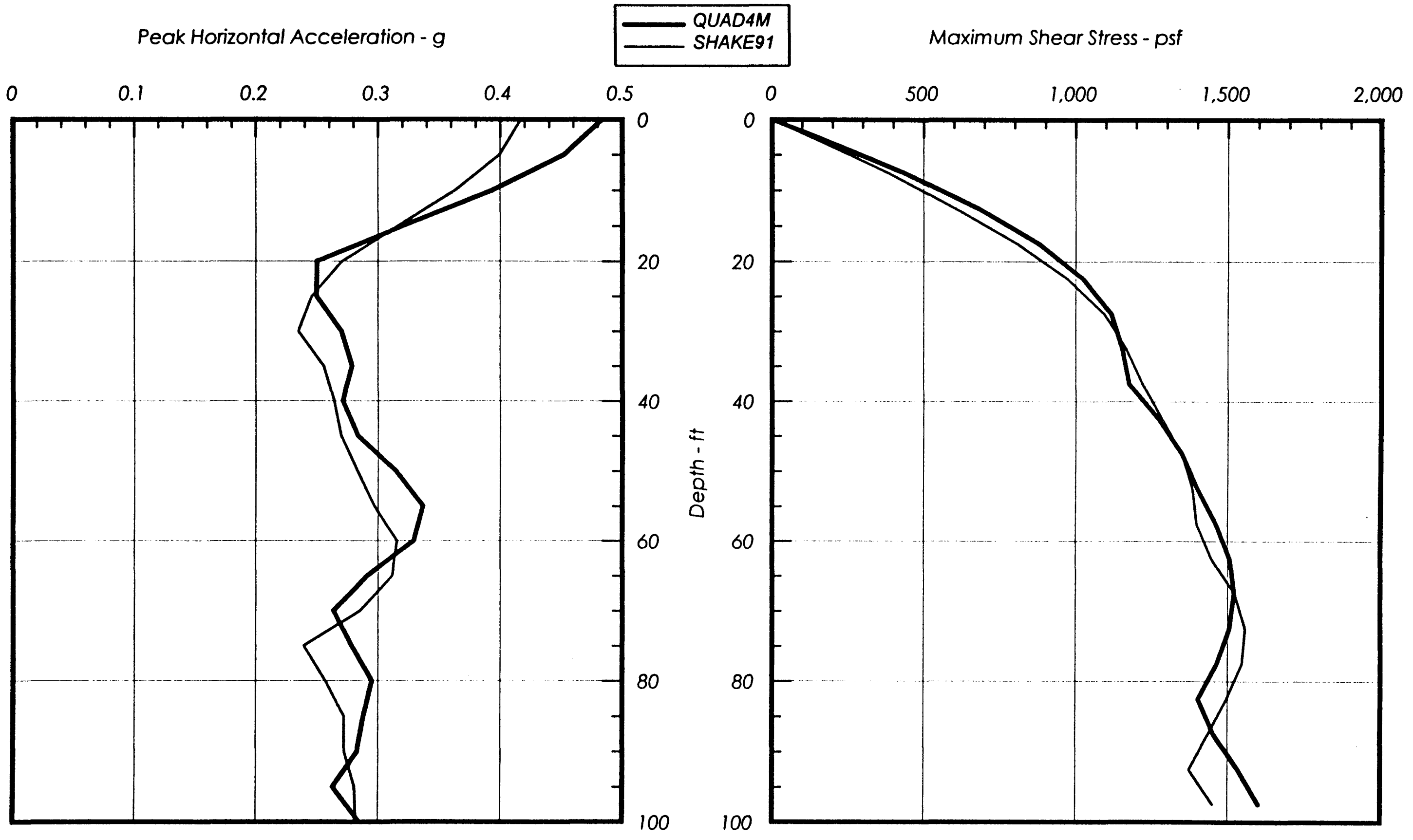


Figure 11
Comparison of Peak Horizontal Accelerations and Maximum Shear Stresses
Computed Using Programs QUAD4M and SHAKE91
for the 100' Dense Sand Layer --
Input Motion: Santa Cruz Record Scaled to 0.3g

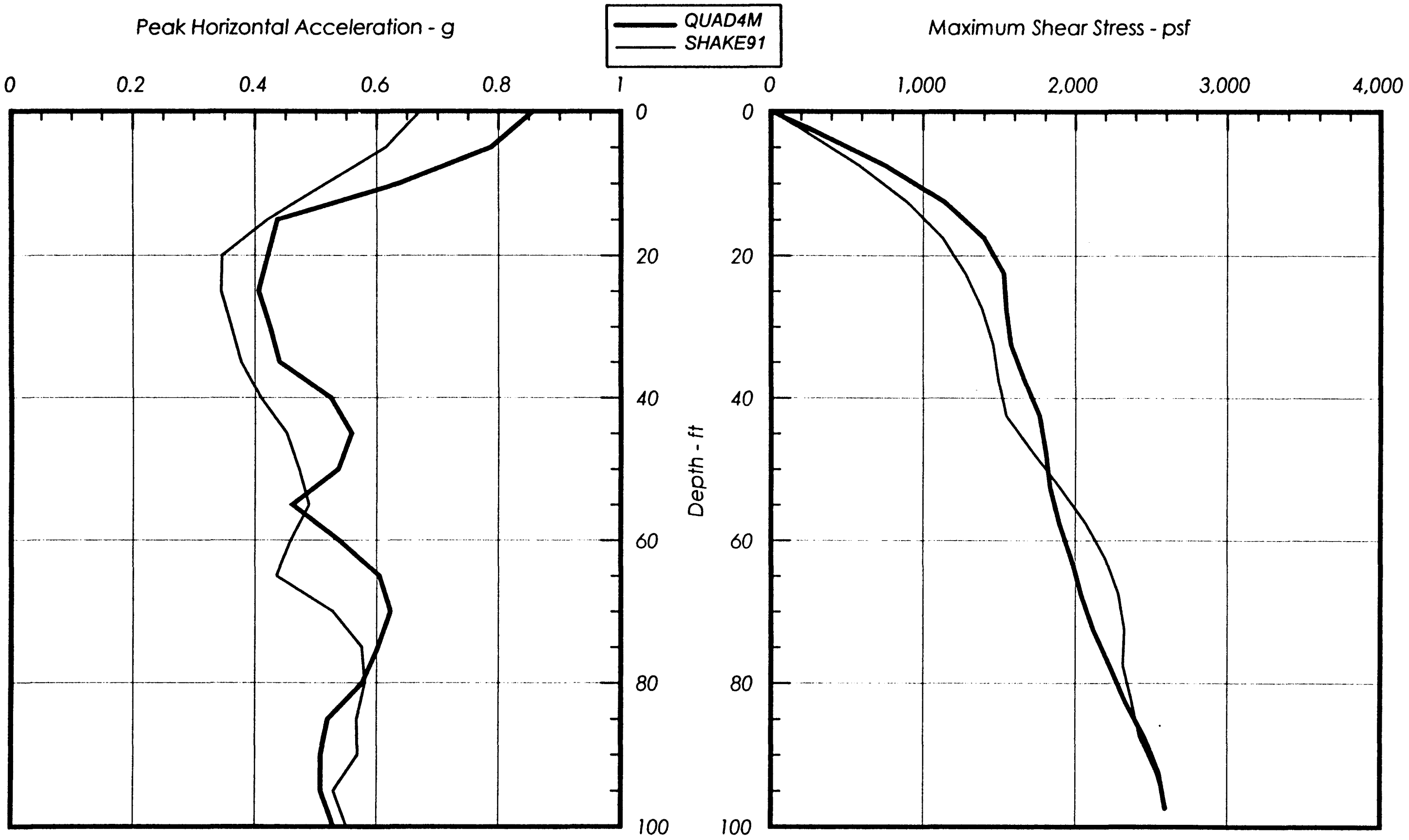


Figure 12
Comparison of Peak Horizontal Accelerations and Maximum Shear Stresses
Computed Using Programs QUAD4M and SHAKE91
for the 100' Dense Sand Layer --
Input Motion: Santa Cruz Record Scaled to 0.6g

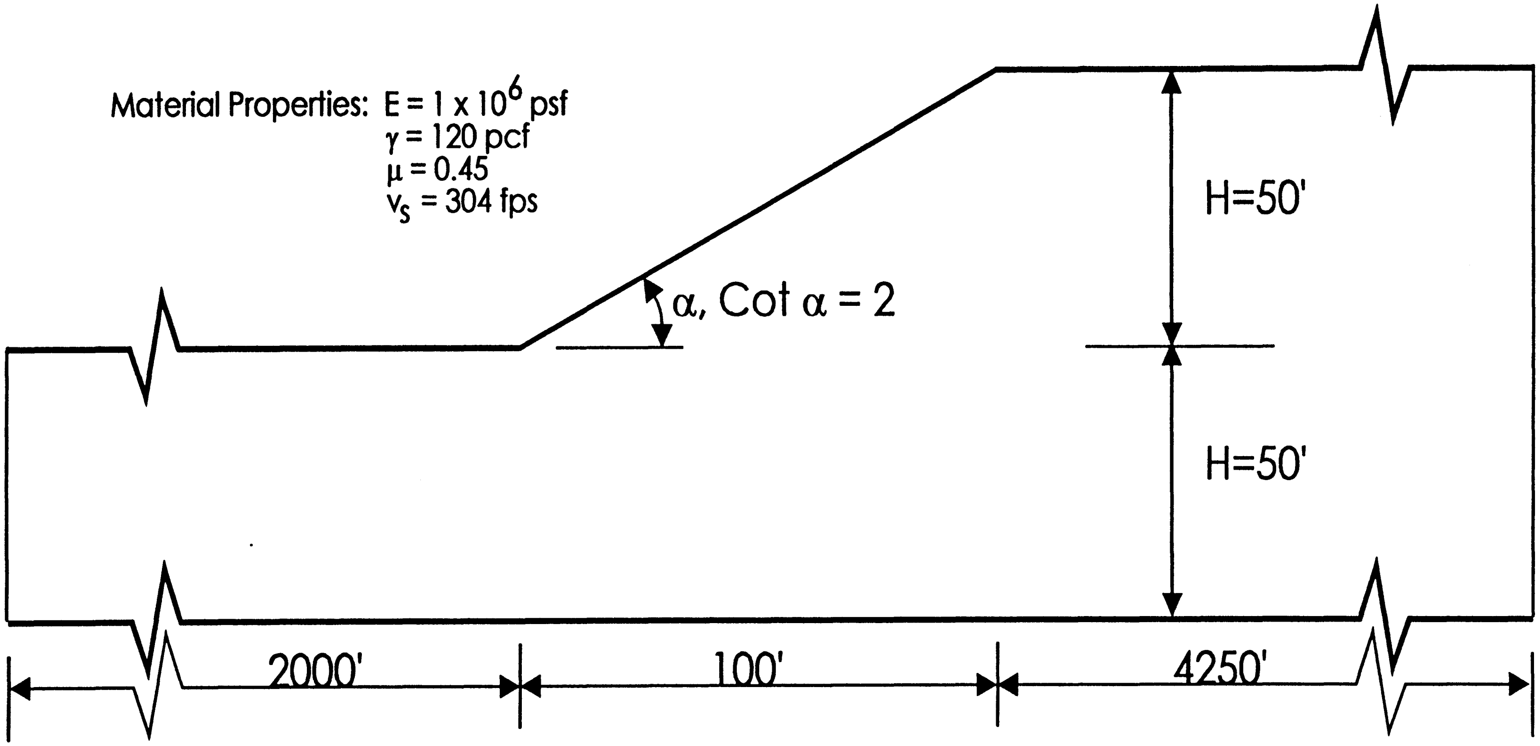
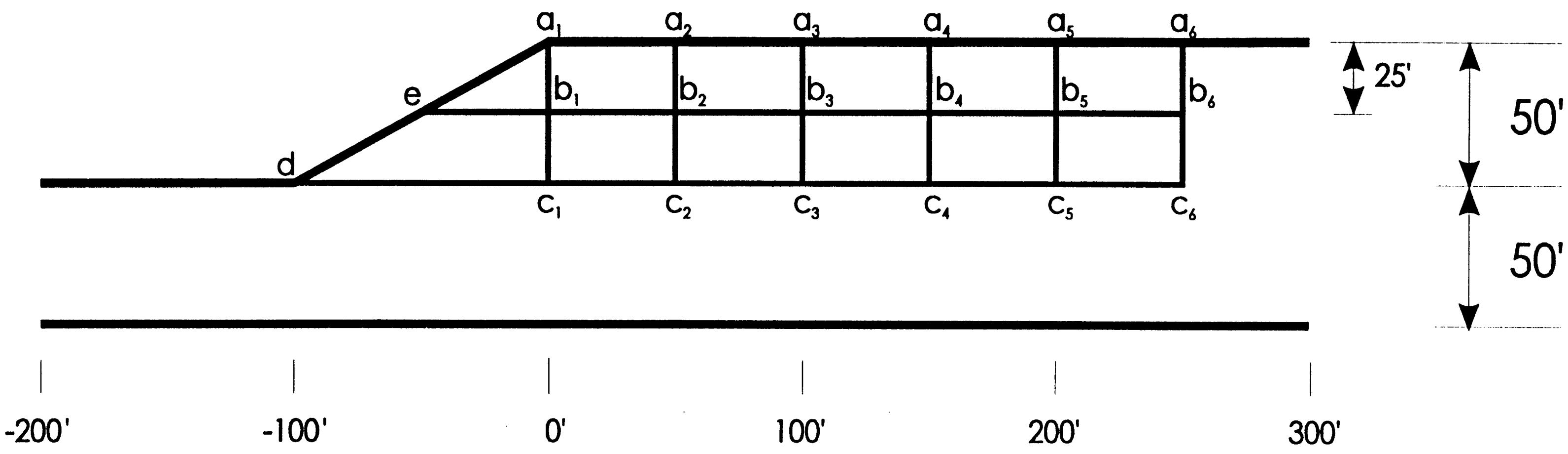


Figure 13

Geometry and Material Properties for Bank of Sample Problem



Half Height	Full Height
Wedge 1 - eb$_1$a$_1$	Wedge 7 - dc$_1$a$_1$
Wedge 2 - eb$_2$a$_2$	Wedge 8 - dc$_2$a$_2$
Wedge 3 - eb$_3$a$_3$	Wedge 9 - dc$_3$a$_3$
Wedge 4 - eb$_4$a$_4$	Wedge 10 - dc$_4$a$_4$
Wedge 5 - eb$_5$a$_5$	Wedge 11 - dc$_5$a$_5$
Wedge 6 - eb$_6$a$_6$	Wedge 12 - dc$_6$a$_6$

Figure 14

Description of Wedges Used in Seismic Coefficient Evaluation

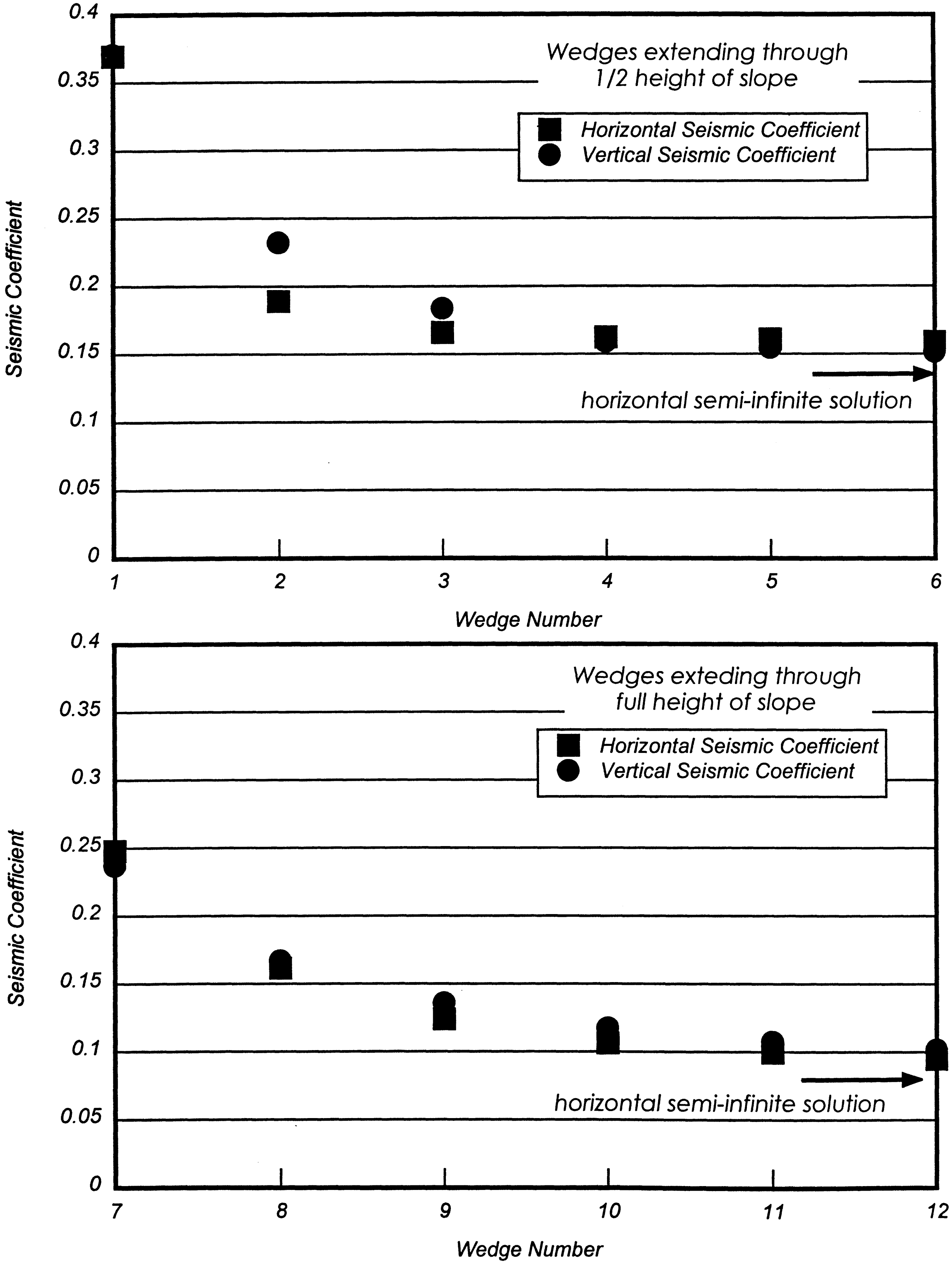


Figure 15
Peak Seismic Coefficients
Example Slope Problem

Appendix A - Program Listing of QUAD4M

Program **QUAD4M** is listed herein. The program has utilized some features of MICROSOFT FORTRAN 5.1 not implemented in the ANSI FORTRAN 77 specifications. These include data structures and dynamic allocation. When using MICROSOFT FORTRAN 5.1, the program can be compiled to run under MICROSOFT WINDOWS 3.1 or DOS, independent of the source listing, but the advantage to using WINDOWS 3.1 is virtually unrestricted access to memory, and in combination with the dynamically allocatable variables, allows large finite element representations to be analyzed on a microcomputer. When using MICROSOFT FORTRAN POWERSTATION 1.0, the program runs many times faster, and has full access to machine memory, but can only be compiled for DOS or Windows NT.

In order to compile under MICROSOFT FORTRAN 5.1, the program was split into 3 modules, **QUAD4M1**, **QUAD4M2**, and **QUAD4M3**. Also, the program must be compiled using the options /AH and /Gt in the compile command. In addition, in order to open enough files at once, a patch may need to be performed on the compiler. This patch and instructions for use are included on the source disk and are available from MICROSOFT.

The program was also compiled to run under MICROSOFT FORTRAN POWERSTATION 1.0, which needs no special considerations for compilation of this program.

The included program was compiled using POWERSTATION and instructions for use are included on the disk.

Acceleration time histories, stress time histories, and seismic coefficient time histories can be requested as described below. The program requires an input data file, earthquake time history files for the horizontal and vertical components, if needed, and a soil property curve file. The format of these files is described in the beginning comment lines of the program.

```
      PROGRAM QUAD4M
C                          QUAD4M
C                      U.C. Davis, 1993
C                   by Martin Byrd Hudson,
C                         I.M. Idriss,
C                      and Mohsen Beikae
C
C                   Modified from QUAD4, 1973,
C        by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
C
C     FINITE ELEMENT ANALYSIS OF PLANE SOIL STRUCTURES SUBJECT TO HORIZONTAL
C     EARTHQUAKE EXCITATION AT THE BASE. THE EQUIVALENT LINEAR METHOD USED
C     INVOLVES DIRECT STEP-BY-STEP INTEGRATION OF EQUATIONS OF MOTION. STRAIN
C     COMPATIBLE RAYLEIGH DAMPING IN ALL ELEMENTS. STRAIN COMPATIBLE SHEAR
C     MODULUS IN ALL ELEMENTS
C
C     ORIGINAL PROGRAM WRITTEN BY I.M. IDRISS - SUMMER 1968 - USING TRIANGULAR
C     ELEMENTS
C
C     MODIFICATIONS
C        1. ADDITION OF QUADRILATERAL ELEMENT BY J. LYSMER AND R. HWANG
C           1971-1972
C        2. INCORPORATION OF DYNAMIC STORAGE BY N. SERFF - 1972
C        3. FACILITY TO PUNCH STRESS HISTORIES ADDED BY N. SERFF - 1973
C        4. ADDITION OF DYNAMIC ALLOCATION.
C           REMOVAL OF INTERPOLATION ABILITIES.
C           REMOVAL OF SUBROUTINE BANSOL.
C           SUBSTITUTION OF BANEIG WITH NEW SUBROUTINE EIGEN.
C           REMOVAL OF USE OF TEMPORARY DISK FILES.
C           BY MARTIN BYRD HUDSON - 1991
C        5. ADDITION OF COMPLIANT BASE
C           BY MARTIN BYRD HUDSON AND MOHSEN BEIKAE - 1991
C        6. CHANGE TIME-STEPPING METHOD FROM WILSON THETA TO
C           TRAPEZOIDAL RULE.
C           USE OF STRUCTURES FOR VARIABLES.
C           ELIMINATION OF MOST LINE NUMBERS
C           BY MARTIN BYRD HUDSON - 1991
C        7. ADDITION OF SEISMIC COEFFICIENTS
C           BY MARTIN BYRD HUDSON AND MOHSEN BEIKAE - 1992
C        8. CHANGE IN INPUT STRUCTURE.
C           ADDITION OF RESTART CAPABILITY
C           BY MARTIN BYRD HUDSON - 1993
C        9. CHANGE DAMPING: DAMPING NOW SET AT TWO PERIODS
C           BY MARTIN BYRD HUDSON - 1993
C
C
C  ** INPUT FILE STRUCTURE:
C       *********************************************************
C       ****** NOTE: COMMENT LINES ARE INTERSPERSED IN THE ******
C       ******       INPUT FILE AS NOTED BELOW            ******
C       *********************************************************
C
C     * LINE 1
C        TITLE       JOB IDENTIFICATION - ANY CHARACTERS
C     * LINE 2 (Comment line)
C     * LINE 3
C        UNITS       SPECIFICATION FOR UNITS USED
C                    "E" FOR ENGLISH
C                    "S" FOR SI
C     * LINE 4 (Comment line)
C     * LINE 5 (5F10.0)
C        DRF         DAMPING REDUCTION FACTOR
C        PRM         FACTOR FOR CONVERTING MAX. STRAIN TO EQUIVALENT UNIFORM
C                    STRAIN (USUALLY 0.55 TO 0.75)
C          THE REMAINDER OF LINE 5 IS NECESSARY ONLY IF A COMPLIANT BASE IS USED
C        ROCKVP      P WAVE VELOCITY (FEET/SEC OR METER/SEC) OF ROCK
C        ROCKVS      S WAVE VELOCITY (FEET/SEC OR METER/SEC) OF ROCK
C        ROCKRHO     AVERAGE ROCK UNIT WEIGHT (PCF OR N/M^3)
C     * LINE 6 (Comment line)
C     * LINE 7 (3I5)
C        NELM        NUMBER OF FINITE ELEMENTS
C        NDPT        TOTAL NUMBER OF NODAL POINTS
C        NSLP        NUMBER OF SURFACES FOR SEISMIC COEFFICIENT ANALYSIS
C     * LINE 8 (Comment line)
C     * LINE 9 - COMPUTATION SWITCHES (9I5)
C        KGMAX       NUMBER OF TIME STEPS IN INPUT EARTHQUAKE RECORD
C        KGEQ        NUMBER OF LAST TIME STEP FOR LAST ITERATION
C        N1EQ        NUMBER OF FIRST TIME STEP FOR LAST ITERATION
C        N2EQ        NUMBER OF FIRST TIME STEP FOR FIRST ITERATIONS
C        N3EQ        NUMBER OF LAST TIME STEP FOR FIRST ITERATIONS
C        NUMB        NUMBER OF ITERATIONS ON SOIL PROPERTIES
C        KV          SWITCH FOR READING VERTICAL COMPONENT OF EQ. RECORD
C                     (2 IF VERT. COMPONENT TO BE READ, 1 IF NO VERT.
C                     COMPONENT TO BE READ)
C        KSAV        SWITCH FOR SAVING STATE OF SYSTEM AT LAST STEP OF
C                     LAST ITERATION FOR RESTART
C                     (0: DON'T SAVE, 1: SAVE)
C     * LINE 10 (Comment line)
C     * LINE 11 - EARTHQUAKE FILE DESCRIPTORS (5F10.0,4I5,F10.0)
C        DTEQ        TIME STEP OF INPUT MOTION (SECONDS)
C        EQMUL(1)    SCALING FACTOR FOR HORZ INPUT MOTION
C        EQMUL(2)    SCALING FACTOR FOR VERT INPUT MOTION
C        UGMAX(1)    SPECIFIED MAXIMUM HORIZONTAL ACCELERATION TO WHICH THE
C                    INPUT MOTION WILL BE SCALED. (G)
C        UGMAX(2)    SPECIFIED MAXIMUM VERTICAL ACCELERATION TO WHICH THE
C                    INPUT MOTION WILL BE SCALED. (G)
C                    NOTE:  EITHER EQMUL OR UGMAX IS USED BY THE PROGRAM TO
C                           OBTAIN THE SCALING FACTOR FOR THE INPUT MOTION.
C                           THEREFORE, FOR EACH COMPONENT, EITHER EQMUL OR
C                           UGMAX MUST BE 0
C        HDRX        NUMBER OF HEADER LINES IN THE HORIZONTAL INPUT TIME
C                    HISTORY
C        HDRY        NUMBER OF HEADER LINES IN THE VERTICAL INPUT TIME HISTORY
C        NPLX        NUMBER OF DATA POINTS PER LINE IN THE HORIZONTAL INPUT
C                    TIME HISTORY.  0 INDICATES 'NOT SPECIFIED'
C        NPLY        NUMBER OF DATA POINTS PER LINE IN THE VERTICAL INPUT
C                    TIME HISTORY.  0 INDICATES 'NOT SPECIFIED'
```

```
C         PRINPUT     PERIOD CORRESPONDING TO MAXIMUM SPECTRAL ACCELERATION OF
C                     THE HORIZONTAL INPUT MOTION (SECONDS).
C      * LINE 12 (Comment line)
C      * LINE 13:  (A)
C         EARTHQH     NAME OF FILE WITH HORIZONTAL INPUT TIME HISTORY
C      * LINE 14:  (A)
C         EQINPFMT(1) EARTHQUAKE FILE INPUT FORMAT.  ENTER '*' IN COLUMN 1 IF
C                     FREE FORMAT IS DESIRED, OR REGULAR FORMAT SPECIFIER IN
C                     PARENTHESES.  USED FOR HORIZONTAL INPUT TIME HISTORY.
C                     NOTE:  WHEN USING FORMAT SPECIFIER,
C                             POINTS PER LINE MUST BE INCLUDED:  e.g. (8F9.6).
C      * LINE 15:  (A) (ONLY USE IF KV=2)
C         EARTHQV     NAME OF FILE WITH VERTICAL INPUT TIME HISTORY
C      * LINE 16:  (A) (ONLY USE IF KV=2)
C         EQINPFMT(2) EARTHQUAKE FILE INPUT FORMAT.  ENTER '*' IN COLUMN 1 IF
C                     FREE FORMAT IS DESIRED, OR REGULAR FORMAT SPECIFIER IN
C                     PARENTHESES.   USED FOR VERTICAL INPUT TIME HISTORY.
C                     SEE NOTE FOR EQINPFMT(1) ABOVE.
C      * LINE 17 (Comment line)
C      * LINE 18: (3I5) (ALL 3 VARIABLES 0 OR 1)
C         SOUT        SWITCH TO READ STRESS OUTPUT FILE DESCRIPTORS: 1=READ
C         AOUT        SWITCH TO READ ACCELERATION OUTPUT FILE DESCRIPTORS: 1=READ
C         KOUT        SWITCH TO READ SEISMIC COEFFICIENT OUTPUT FILE DESCRIPTORS: 1=READ
C-----------------------------------------------------------------------------------
C      * STRESS OUTPUT FILE DESCRIPTORS (ONLY USE LINES 19-22 IF SOUT=1)
C        LINE 19 (Comment line)
C        LINE 20:  (A)
C         SHISTFMT    OUTPUT FORMAT:
C                       EITHER
C                         'COMBINED'
C                         WHICH COMBINES ALL REQUESTED STRESS HISTORIES INTO
C                         ONE FILE SUCH THAT EACH STRESS HISTORY IS OUTPUT AS A
C                         SEPARATE COLUMN WITHIN THAT FILE
C                       OR
C                         'MULTIPLE'
C                         WHICH PRODUCES EACH STRESS HISTORY AS A SEPARATE
C                         FILE WITH FORMAT 8F8.1
C        LINE 21:  (A)
C         SFILEOUT    OUTPUT FILE NAME ROOT (8 CHARACTERS MAX IF COMBINED
C                     OUTPUT, OR 6 CHARACTERS MAX IF MULTIPLE OUTPUT)
C        LINE 22:  (A)
C         SSUFFIX     OUTPUT FILE NAME SUFFIX (3 CHARACTERS MAX)
C-----------------------------------------------------------------------------------
C      * ACCELERATION OUTPUT FILE DESCRIPTORS (ONLY USE LINES 23-26 IF AOUT=1)
C        LINE 23 (Comment line)
C        LINE 24:  (A)
C         AHISTFMT    OUTPUT FORMAT:
C                       EITHER
C                         'COMBINED'
C                         WHICH COMBINES ALL REQUESTED ACCELERATION HISTORIES
C                         INTO ONE FILE SUCH THAT EACH STRESS HISTORY IS
C                         OUTPUT AS A SEPARATE COLUMN WITHIN THAT FILE
C                       OR
C                         'MULTIPLE'
```

```
C                         WHICH PRODUCES EACH ACCEL HISTORY AS A SEPARATE FILE
C                         WITH FORMAT 8F9.6
C        LINE 25:  (A)
C          AFILEOUT    OUTPUT FILE NAME ROOT (8 CHARACTERS MAX IF COMBINED
C                        OUTPUT, OR 6 CHARACTERS MAX IF MULTIPLE OUTPUT)
C        LINE 26:  (A)
C          ASUFFIX     OUTPUT FILE NAME SUFFIX (3 CHARACTERS MAX)
C---------------------------------------------------------------------------------------
C      * SEISMIC COEFFICIENT OUTPUT FILE DESCRIPTORS (ONLY USE LINES 27-30 IF KOUT=1)
C        LINE 27 (Comment line)
C        LINE 28:  (A)
C          KHISTFMT    OUTPUT FORMAT:
C                        EITHER
C                         'COMBINED'
C                         WHICH COMBINES ALL REQUESTED ACCELERATION HISTORIES
C                         INTO ONE FILE SUCH THAT EACH STRESS HISTORY IS OUTPUT
C                         AS A SEPARATE COLUMN WITHIN THAT FILE
C                        OR
C                         'MULTIPLE'
C                         WHICH PRODUCES EACH ACCEL HISTORY AS A SEPARATE FILE
C                         WITH FORMAT 8F9.6
C        LINE 29:  (A)
C          KFILEOUT    OUTPUT FILE NAME ROOT (8 CHARACTERS MAX IF COMBINED
C                        OUTPUT, OR 6 CHARACTERS MAX IF MULTIPLE OUTPUT)
C        LINE 30:  (A)
C          KSUFFIX     OUTPUT FILE NAME SUFFIX (3 CHARACTERS MAX)
C---------------------------------------------------------------------------------------
C      * RESTART FILE NAME DESCRIPTOR (ONLY USE LINES 31-32 IF KSAV=1)
C        LINE 31 (Comment line)
C        LINE 32 - FILE NAME TO USE FOR SAVING FINAL STATE OF SOIL FOR RESTART (A)
C          SAVEFILE    FILE NAME (EXCLUDING PATH)
C---------------------------------------------------------------------------------------
C      * SEISMIC COEFFICIENT LINES (ONLY USE LINES 33-38 IF NSLP>0;REPEAT NSLP TIMES)
C        LINE 33 (Comment line)
C        LINE 34: (2I5)
C          NSEG(I)     NUMBER OF NODES INTERSECTED BY SURFACE I
C          ESEG(I)     NUMBER OF ELEMENTS WITHIN SURFACE I
C        LINE 35 (Comment line)
C        LINE 36:  (15I5)  (must be repeated if NSEG(I)>15)
C          NOSEG(I,J)  NODE J INTERSECTED BY SURFACE I (NSEG(I) NODES)
C        LINE 37 (Comment line)
C        LINE 38:  (15I5)  (must be repeated if ESEG(I)>15)
C          ELSEG(I,J)   ELEMENT J WITHIN SURFACE I (ESEG(I) ELEMENTS)
C---------------------------------------------------------------------------------------
C      * LINE 39 (Comment line)
C      * LINE 40 - ELEMENT LINES (REPEAT LINE 40 NELM TIMES) (6I5,5F10.0,I5)
C          N           ELEMENT NUMBER
C          EL(N).NODE(1)   1ST. NODAL POINT NUMBER
C          EL(N).NODE(2)   2ND. NODAL POINT NUMBER
C          EL(N).NODE(3)   3RD. NODAL POINT NUMBER
C          EL(N).NODE(4)   4TH. NODAL POINT NUMBER
C                      NOTE  1. SPECIFY NODAL POINTS IN COUNTER-CLOCKWISE
C                               DIRECTION
C                            2. EL(N).NODE(3) = EL(N).NODE(4) PRODUCES
```

```
C                      TRIANGULAR ELEMENT
C        EL(N).TYPE  MATERIAL TYPE
C                      CORRESPONDS TO SOILS IN THE SOIL DATA FILE
C                      0 INDICATES MATERIAL PROPERTIES USED FOR ITERATIONS
C                        ARE NOT TO BE CHANGED FROM THOSE SPECIFIED FOR THE
C                        1ST ITERATION
C        EL(N).DENS  TOTAL UNIT WEIGHT (PCF OR N/M^3)
C        EL(N).PO    POISSONS RATIO
C        EL(N).GMX   SHEAR MODULUS AT SMALL STRAINS (KSF OR KN/M^2)
C                    LEAVE BLANK IF TYPE(N) = 6
C        EL(N).G     SHEAR MODULUS FOR 1ST ITERATION (KSF OR KN/M^2)
C        EL(N).XL    INITIAL FRACTION OF CRITICAL DAMPING (DECIMAL)
C        EL(N).LSTR  KEY FOR DETERMINING WHETHER STRESS OUTPUT
C                    IS SIGX, SIGY, TAUXY, OR ANY COMBINATION OF
C                    THESE
C                    1 WILL PRINT HISTORY FOR SIGX
C                    2 WILL PRINT HISTORY FOR SIGY
C                    4 WILL PRINT HISTORY FOR TAUXY
C    * LINE 41 (Comment line)
C    * LINE 42 - NODAL POINT LINES (REPEAT LINE 42 NDPT TIMES) (I5,2F10.0,2I5,6F10.0)
C        M           NODAL POINT NUMBER
C        NO(M).XORD     X-COORDINATE (FEET OR METERS)
C        NO(M).YORD     Y-COORDINATE (FEET OR METERS)
C        NO(M).BC    KEY FOR BOUNDARY CONDITION - SEE BELOW
C                    0 - FREE NODAL POINT
C                    1 - INPUT HORIZONTAL EARTHQUAKE MOTION APPLIED, FREE IN
C                        Y-DIRECTION
C                    2 - INPUT VERTICAL EARTHQUAKE MOTION APPLIED, FREE IN
C                        X-DIRECTION
C                    3 - INPUT HORIZONTAL AND VERTICAL EARTHQUAKE MOTION
C                        APPLIED
C                    4 - TRANSMITTING BASE NODE
C        NO(M).OUT      KEY FOR ACCELERATION OUTPUT - SEE BELOW
C                    0 - NO ACCELERATION HISTORY OUTPUT
C                    1 - X-DIRECTION ACCELERATION OUTPUT
C                    2 - Y-DIRECTION ACCELERATION OUTPUT
C                    3 - X & Y-DIRECTION ACCELERATION OUTPUT
C        NO(M).X2I(1) INITIAL HORIZONTAL ACCELERATION
C        NO(M).X1I(1) INITIAL HORIZONTAL VELOCITY
C        NO(M).XI(1)  INITIAL HORIZONTAL DISPLACEMENT
C        NO(M).X2I(2) INITIAL VERTICAL ACCELERATION
C        NO(M).X1I(2) INITIAL VERTICAL VELOCITY
C        NO(M).XI(2)  INITIAL VERTICAL DISPLACEMENT
C                    NOTE  1. INITIAL CONDITIONS ARE SPECIFIED AT 1 TIME
C                             STEP BEFORE SHAKING BEGINS
C                          2. ALL INITIAL CONDITIONS ARE RELATIVE TO THE INPUT
C                             HORIZONTAL AND VERTICAL EARTHQUAKE MOTION
C
C-----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
C
C
C  ** EARTHQUAKE ACCELERATION FILE STRUCTURE:
C     * LINE 1 - EARTHQUAKE IDENTIFICATION
C        EQNH        IDENTIFICATION FOR HORIZONTAL COMPONENT
C     OR  EQNV        IDENTIFICATION FOR VERTICAL COMPONENT
C     * LINE 2 - OTHER EARTHQUAKE HEADER LINES NOT ECHOED
C             (REPEAT LINE 2 HDRX-1 or HDRY-1 TIMES)
C     * LINE 3 - INPUT ACCELERATIONS (REPEAT LINE 3 SUFFICIENT TIMES FOR DATA)
C        U2G(I,J), I=1,KGMAX, J=1 IF HORIZONTAL ACCELERATION (G),
C                             J=2 IF VERTICAL ACCELERATION (G)
C
C-----------------------------------------------------------------------------
C-----------------------------------------------------------------------------
C
C
C  ** SOIL DATA FILE STRUCTURE
C     * LINE 1 - NUMBER OF SOIL TYPES IN FILE (I5)
C        NUMPROPS
C-----------------------------------------------------------------------------
C     * REPEAT LINES 2-7 NSLP TIMES (ONCE FOR EACH SOIL TYPE (I)):
C       LINE 2 (I5,A75)
C         NUMPOINTS(1,I) NUMBER OF DATA POINTS FOR SHEAR MODULUS
C         IDPROP(1,I) IDENTIFICATION
C       LINE 3 - SHEAR STRAIN COORDINATES (8F10.0)
C         SDATA(1,I,J), J=1,NUMPOINTS(1,I)  (VALUES IN PERCENT)
C       LINE 4 - G/GMAX COORDINATES (8F10.0)
C         GDATA(I,J), J=1,NUMPOINTS(1,I) FRACTION OF GMAX
C       LINE 5 (I5,A75)
C         NUMPOINTS(2,I) NUMBER OF DATA POINTS FOR DAMPING
C         IDPROP(2,I) IDENTIFICATION
C       LINE 6 - SHEAR STRAIN COORDINATES (8F10.0)
C         SDATA(2,I,J), J=1,NUMPOINTS(2,I)  (VALUES IN PERCENT)
C       LINE 7 - DAMPING COORDINATES (8F10.0)
C         DDATA(I,J), J=1,NUMPOINTS(2,I)  (VALUES IN PERCENT)
C-----------------------------------------------------------------------------
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      COMMON/CONST4/SHISTFMT,AHISTFMT,KHISTFMT,SFILEOUT,AFILEOUT,
     &           KFILEOUT,SSUFFIX,ASUFFIX,KSUFFIX,DIROUT,SAVEFILE,
     &           DATAIN
      CHARACTER SHISTFMT*1,AHISTFMT*1,KHISTFMT*1,SFILEOUT*8,AFILEOUT*8,
     &         KFILEOUT*8,SSUFFIX*3,ASUFFIX*3,KSUFFIX*3,DIROUT*72,
     &         SAVEFILE*12,DATAIN*72
      COMMON/CONST5/HDRX,HDRY,EARTHQH,EARTHQV,EQINPFMT(2),ROCKVP,ROCKVS,
     &           ROCKRHO,IHISTSTR,IHISTACC,NPLX,NPLY
```

```
      CHARACTER EQINPFMT*20,EARTHQH*72,EARTHQV*72
      INTEGER SOUT,AOUT,KOUT,HDRX,HDRY,ESEG,ELSEG,ESEGMAX
      CHARACTER*72 FILEIN,FILEOUT,IDPROP*75
      RECORD /ELEMENT/EL [ALLOCATABLE] (:)
      RECORD /NODE/NO [ALLOCATABLE] (:)
      DIMENSION ZMASS [ALLOCATABLE] (:),
     & SMASS [ALLOCATABLE] (:),
     & R [ALLOCATABLE] (:),
     & AT [ALLOCATABLE] (:),
     & BT [ALLOCATABLE] (:),
     & X1 [ALLOCATABLE] (:),
     & X2 [ALLOCATABLE] (:),
     & DSMAX [ALLOCATABLE] (:),
     & TIME1 [ALLOCATABLE] (:),
     & NBC [ALLOCATABLE] (:),
     & U2G [ALLOCATABLE] (:,:),
     & X [ALLOCATABLE] (:),
     & ST [ALLOCATABLE] (:,:),
     & DS [ALLOCATABLE] (:,:),
     & IDPROP [ALLOCATABLE] (:,:),
     & SDATA [ALLOCATABLE] (:,:,:),
     & GDATA [ALLOCATABLE] (:,:),
     & DDATA [ALLOCATABLE] (:,:),
     & NUMPOINTS [ALLOCATABLE] (:,:),
     & NSEG [ALLOCATABLE] (:),
     & NOSEG [ALLOCATABLE] (:,:),
     & ESEG [ALLOCATABLE] (:),
     & ELSEG [ALLOCATABLE] (:,:)
C
C      INPUT DATA
C
      WRITE(*,9000)
      WRITE(*,9001)
      READ(*,'(A)') FILEIN
      OPEN (5,FILE=FILEIN,STATUS='OLD')
      WRITE(*,9002)
      READ(*,'(A)') DATAIN
      OPEN (7,FILE=DATAIN,STATUS='OLD')
      WRITE(*,'(A,A)')' Enter Output directory (with ending\)',
     &              ' (.\ for current dir):'
      READ(*,'(A)') DIROUT
      WRITE(*,'(A)') ' Output File Name (without directory name):'
      READ(*,'(A)') FILEOUT
      OPEN (6,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILEOUT)
      WRITE (6,9000)
      PRINT*, ' Read Header Lines...'
      READ(5,'(A//A)') TITLE,UNITS
      IF (UNITS.EQ.'E') THEN
         GRAV = 32.2
      ELSE IF (UNITS.EQ.'S') THEN
         GRAV = 9.81
      ELSE
         WRITE(*,'(A)') ' INVALID UNITS SPECIFICATION'
      END IF
```

```
      READ(5,'(/5F10.0)') DRF,PRM,ROCKVP,ROCKVS,ROCKRHO
      ROCKRHO=ROCKRHO/GRAV       ! Convert from unit weight to density
      ROCKBETAS=ROCKRHO*ROCKVS
      ROCKBETAP=ROCKRHO*ROCKVP
      READ(5,'(/3I5)') NELM,NDPT,NSLP
      READ(5,'(/9I5)') KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
C
C     READ EARTHQUAKE FILE DESCRIPTORS
C
      PRINT*, ' Read Earthquake File Descriptors... '
      READ(5,'(/5F10.0,4I5,F10.0)') DTEQ,EQMUL(1),EQMUL(2),UGMAX(1),
     &                               UGMAX(2),HDRX,HDRY,NPLX,NPLY,PRINPUT
      READ(5,'(/A)') EARTHQH
      OPEN(8,FILE=EARTHQH,STATUS='OLD')
      READ(5,'(A)') EQINPFMT(1)
      IF (KV.EQ.2) THEN
        READ(5,'(A)') EARTHQV
        OPEN(9,FILE=EARTHQV,STATUS='OLD')
        READ(5,'(A)') EQINPFMT(2)
      END IF
      READ(5,'(/3I5)') SOUT,AOUT,KOUT
      IF (ABS(SOUT).GT.1.OR.ABS(AOUT).GT.1.OR.ABS(KOUT).GT.1.OR.
     &    ABS(KSAV).GT.1) THEN
        PRINT*, ' One of SOUT, AOUT, KOUT, or KSAV is neither 0 nor 1!'
        STOP
      END IF
      IF (ABS(PRINPUT).LT.0.0000001) THEN
        PRINT*, ' PRINPUT MUST BE USED!'
        STOP
      END IF
C
C     READ STRESS OUTPUT FILE DESCRIPTORS
C
      IF (SOUT.EQ.1) THEN
        PRINT*, ' Read Stress Output File Descriptors... '
        READ(5,'(/A1)') SHISTFMT
        IF (SHISTFMT.EQ.'M'.OR.SHISTFMT.EQ.'m') THEN
          READ(5,'(A6/A3)') SFILEOUT,SSUFFIX
        ELSE
          READ(5,'(A8/A3)') SFILEOUT,SSUFFIX
        END IF
      END IF
C
C     READ ACCELERATION OUTPUT FILE DESCRIPTORS
C
      IF (AOUT.EQ.1) THEN
        PRINT*, ' Read Acceleration Output File Descriptors... '
        READ(5,'(/A1)') AHISTFMT
        IF (AHISTFMT.EQ.'M'.OR.AHISTFMT.EQ.'m') THEN
          READ(5,'(A6/A3)') AFILEOUT,ASUFFIX
        ELSE
          READ(5,'(A8/A3)') AFILEOUT,ASUFFIX
        END IF
      END IF
```

```
C
C     READ SEISMIC COEFFICIENT OUTPUT FILE DESCRIPTORS
C
      IF (KOUT.EQ.1.OR.NSLP.GT.0) THEN
        IF (KOUT.EQ.0) THEN
          PRINT*, ' Seismic Coefficient Output File Descriptors',
     &            ' read because NSLP is greater than 0'
        ELSE
          PRINT*,' Read Seismic Coefficient Output File Descriptors.'
        END IF
        READ(5,'(/A1)') KHISTFMT
        IF (KHISTFMT.EQ.'M'.OR.KHISTFMT.EQ.'m') THEN
          READ(5,'(A6/A3)') KFILEOUT,KSUFFIX
        ELSE
          READ(5,'(A8/A3)') KFILEOUT,KSUFFIX
        END IF
      END IF
C
C     READ SYSTEM STATE OUTPUT FILE NAME
C
      IF (KSAV.EQ.1) THEN
        PRINT*, ' Read Output File Name for Restart'
        READ(5,'(/A)') SAVEFILE
        OPEN(4,FILE=DIROUT(:LEN_TRIM(DIROUT))//SAVEFILE)
      END IF

      CALL BANDW(MAXW,MAXEL,IHISTSTR,IHISTACC,NSEGMAX,ESEGMAX)

      CALL SOILSCAN(NUMPROPS,MAXPOINTS)

      REWIND 5
      LINESKIP=14+2*KV+4*SOUT+4*AOUT+4*KOUT+2*KSAV
      DO M=1,LINESKIP
        READ (5,'(/\)')
      END DO
      NF = 2*(NB1-1)       ! Degrees of Freedom for 2 unknowns per node
      ALLOCATE (NO(NDPT),EL(NELM),ZMASS(NDPT),SMASS(NF),R(NF+1),AT(NF),
     &          BT(NF+1),X1(NF),X2(NF),DSMAX(NF),TIME1(NF),X(2*NDPT),
     &          ST(NF,MAXW),DS(NF,MAXW),IDPROP(2,NUMPROPS),
     &          SDATA(2,NUMPROPS,MAXPOINTS),GDATA(NUMPROPS,MAXPOINTS),
     &          DDATA(NUMPROPS,MAXPOINTS),NUMPOINTS(2,NUMPROPS))
      IF (KV.EQ.2) THEN
        ALLOCATE (U2G(0:KGMAX,2))
      ELSE
        ALLOCATE (U2G(0:KGMAX,1))
      END IF
      IF (NBP.GT.0) THEN
        ALLOCATE (NBC(NBP))
      END IF
      IF (NSLP.GT.0) THEN
        ALLOCATE (NOSEG(NSLP,NSEGMAX),NSEG(NSLP),ELSEG(NSLP,ESEGMAX),
     &            ESEG(NSLP))
      END IF
C
      CALL INPT(EL,NO,U2G,IDPROP,SDATA,GDATA,DDATA,NUMPROPS,NUMPOINTS,
     &          NOSEG,NSEG,ESEG,ELSEG)
      IF (KV.EQ.1) THEN
        WRITE(*,92) TITLE,EARTHQH,EQNH
        WRITE(6,92) TITLE,EARTHQH,EQNH
      ELSE
        WRITE(*,93) TITLE,EARTHQH,EQNH,EARTHQV,EQNV
        WRITE(6,93) TITLE,EARTHQH,EQNH,EARTHQV,EQNV
      END IF
      CALL INIT(EL,U2G)
      CALL ECHO(EL,NO,ROCKBETAP,ROCKBETAS,IDPROP,DDATA,GDATA,SDATA,
     &          NUMPOINTS,NUMPROPS,NBC,NSEG,NOSEG,ESEG,ELSEG,MAXW,MAXEL)
      CALL QUD4D(EL,NO,ZMASS,SMASS,R,AT,BT,X1,X2,DSMAX,TIME1,NBC,U2G,X,
     &           ST,DS,MAXW,IHISTSTR,IHISTACC,SDATA,GDATA,DDATA,
     &           NUMPROPS,NUMPOINTS,ROCKBETAS,ROCKBETAP,NSEG,NOSEG,ESEG,
     &           ELSEG)
C
   92 FORMAT(5X,A,/2X,
     &       'HORIZONTAL ACCELERATION INPUT FILE: ',/,5X,
     &       A/,2X,'WITH FIRST LINE: ',/,5X,A/)
   93 FORMAT(5X,A,/2X,
     &       'HORIZONTAL ACCELERATION INPUT FILE: ',/,5X
     &       A/,2X,'WITH FIRST LINE: ',/,5X,A/2X,
     &       'VERTICAL ACCELERATION INPUT FILE: ',/,5X
     &       A/,2X,'WITH FIRST LINE: ',/,5X,A/)
 9000 FORMAT(//
     &10X,'****************************************************'/
     &10X,'****************************************************'/
     &10X,'** QUAD4M   A COMPUTER PROGRAM FOR EVALUATING THE **'/
     &10X,'**          SEISMIC RESPONSE OF SOIL STRUCTURES   **'/
     &10X,'**                 U.C.Davis, 1993                **'/
     &10X,'**              by Martin Byrd Hudson,            **'/
     &10X,'**                 I.M.Idriss,                    **'/
     &10X,'**                 and Mohsen Beikae              **'/
     &10X,'**     MODIFIED FROM QUAD4, 1973                  **'/
     &10X,'**              by I.M.Idriss,                    **'/
     &10X,'**                 J. Lysmer,                     **'/
     &10X,'**                 R. Hwang and                   **'/
     &10X,'**                 H. Bolton Seed                 **'/
     &10X,'****************************************************'/
     &10X,'****************************************************'/)
 9001 FORMAT(1X,'Enter Input File Name:')
 9002 FORMAT(1X,'Enter Soil Data File Name:')
      STOP
      END
C
C *****************************************************************
C
      SUBROUTINE BANDW(MAXW,MAXEL,IHISTSTR,IHISTACC,NSEGMAX,ESEGMAX)
C
C     Finds maximum values of matrices for dimensioning
C
C                         Written for QUAD4M
C                          U.C. Davis, 1993
```

```
C           by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      INTEGER TYPE,BC,OUT,ESEGMAX,ESEG,ELSEG
      DIMENSION BC [ALLOCATABLE] (:),NP [ALLOCATABLE] (:,:),
     &          NOSEG [ALLOCATABLE] (:),ELSEG [ALLOCATABLE] (:)
C
      ALLOCATE (BC(NDPT),NP(NELM,4))
      NSEGMAX=0
      ESEGMAX=0
      NB1=NDPT+1
      J = 0
      IHISTSTR = 0
C
C     Read Seismic Coefficient Surface Information
C
      IF (NSLP.GT.0) THEN
        PRINT*, ' Read Seismic Coefficient Surface Lines... '
        DO M=1,NSLP
          READ (5,'(/\)')
          READ(5,'(2I5)') NSEG,ESEG
          ALLOCATE (NOSEG(NSEG),ELSEG(ESEG))
          READ (5,'(/\)')
          READ(5,'(15I5)') NOSEG
          READ (5,'(/\)')
          READ(5,'(15I5)') ELSEG
          NSEGMAX=MAX(NSEG,NSEGMAX)
          ESEGMAX=MAX(ESEG,ESEGMAX)
          DEALLOCATE (NOSEG,ELSEG)
        END DO
      END IF
      PRINT*, ' Read Element Lines... '
      READ (5,'(/\)')
      DO M=1,NELM
        READ(5,'(6I5,5F10.0,I5)',ERR=99) N,NP(N,1),NP(N,2),NP(N,3),
     &        NP(N,4),TYPE,DENS,PO,GMX,G,XL,LSTR
        IF (LSTR.EQ.1.OR.LSTR.EQ.2.OR.LSTR.EQ.4) THEN
          IHISTSTR=IHISTSTR+1
        ELSE IF (LSTR.EQ.3.OR.LSTR.EQ.5.OR.LSTR.EQ.6) THEN
          IHISTSTR=IHISTSTR+2
        ELSE IF (LSTR.EQ.7) THEN
          IHISTSTR=IHISTSTR+3
```

```
        END IF
      END DO
      NBP=0
      IHISTACC = 0
      PRINT*, ' Read Node Lines... '
      READ (5,'(/\)')
      DO M=1,NDPT
        READ(5,'(I5,2F10.0,2I5,6F13.0)',ERR=99) N,XORD,YORD,BC(N),OUT,
     &                                       X2IH,X1IH,XIH,X2IV,X1IV,XIV
        IF (OUT.EQ.1.OR.OUT.EQ.2) THEN
          IHISTACC=IHISTACC+1
        ELSE IF (OUT.EQ.3) THEN
          IHISTACC=IHISTACC+2
        END IF
      END DO
C
      DO N=NDPT,1,-1
        IF (BC(N).EQ.3.AND.N.EQ.NB1-1) THEN
          NB1=N
        ELSE IF (BC(N).EQ.1.OR.BC(N).EQ.2) THEN
          NBP=NBP+1
        ELSE IF (BC(N).EQ.3) THEN
          NBP=NBP+2
        ELSE IF (BC(N).EQ.4.AND.KV.EQ.1) THEN
          NBP=NBP+1
        ELSE IF (BC(N).EQ.5.AND.KV.EQ.1) THEN
          NBP=NBP+1
        END IF
      END DO
C
      DO N=1,NELM
        DO I = 1,4
          DO L = 1,4
            IF (NP(N,I).GE.NB1.OR.NP(N,L).GE.NB1) CYCLE ! Disregard
                                                        ! fixed base
            K = IABS(NP(N,I)-NP(N,L))
            IF (K.GT.J) THEN
              J = K
              MAXEL=N
            END IF
          END DO
        END DO
      END DO
      MAXW = 2*J + 2     ! Half-Bandwidth based upon 2 unknowns per node
C
      DEALLOCATE (BC,NP)
      RETURN
   99 PRINT*, ' ERROR READING ELEMENT OR NODE NO. ',M
      STOP
      END
C
C ***********************************************************************
C
      SUBROUTINE BOUNDARY(NO,M,N,NBC)
```

```
C
C     Marks nodes according to boundary conditions and echoes boundary
C     conditions
C
C                            Written for QUAD4M
C                             U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                        Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      RECORD /NODE/NO
      DIMENSION NBC(*)
C
      K = NO.BC + 1
      IF (NO.BC.EQ.0) THEN          ! Free Node
        WRITE(6,152) M,NO.XORD,NO.YORD,NO.TRIBLEN
      ELSE IF (NO.BC.EQ.1) THEN     ! Cannot move in x-dir
        WRITE(6,1012) M,NO.XORD,NO.YORD,NO.TRIBLEN
        N=N+1
        NBC(N)=2*M-1
      ELSE IF (NO.BC.EQ.2) THEN     ! Cannot move in y-dir
        WRITE(6,1022) M,NO.XORD,NO.YORD,NO.TRIBLEN
        N=N+1
        NBC(N)=2*M
      ELSE IF (NO.BC.EQ.3) THEN     ! Can't move in x or y
        WRITE(6,1002) M,NO.XORD,NO.YORD,NO.TRIBLEN
        IF (M.GE.NB1) RETURN
        N=N+1
        NBC(N)=2*M-1
        N=N+1
        NBC(N)=2*M
      ELSE IF (NO.BC.EQ.4) THEN     ! Trans. base node
        WRITE (6,1003) M,NO.XORD,NO.YORD,NO.TRIBLEN
        IF (KV.EQ.1) THEN
          N=N+1
          NBC(N)=2*M
        END IF
      ELSE
        WRITE (6,*) ' ILLEGAL NODE BOUNDARY CONDITION SPECIFICATION'
      END IF
      RETURN
  152 FORMAT (I8,3F15.3)
  102 FORMAT (I8,3F15.3,3X,'FIXED')
  103 FORMAT (I8,3F15.3,3X,'COMPLIANT')
  112 FORMAT (I8,3F15.3,3X,'X FIXED')
  122 FORMAT (I8,3F15.3,3X,'Y FIXED')
 1002 FORMAT (I8,3F15.3,3X,'CANNOT MOVE IN BOTH X AND Y DIRECTIONS')
 1003 FORMAT (I8,3F15.3,3X,'COMPLIANT BASE NODE')
 1012 FORMAT (I8,3F15.3,3X,'CANNOT MOVE IN X DIRECTION')
 1022 FORMAT (I8,3F15.3,3X,'CANNOT MOVE IN Y DIRECTION')
      END
C
C ***********************************************************************
C
      SUBROUTINE ECHO(EL,NO,ROCKBETAP,ROCKBETAS,IDPROP,DDATA,GDATA,
     &                SDATA,NUMPOINTS,NUMPROPS,NBC,NSEG,NOSEG,ESEG,
     &                ELSEG,MAXW,MAXEL)
C
C     Echos input data to screen and sets up file names
C
C                            Written for QUAD4M
C                             U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                        Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      COMMON/CONST4/SHISTFMT,AHISTFMT,KHISTFMT,SFILEOUT,AFILEOUT,
     &              KFILEOUT,SSUFFIX,ASUFFIX,KSUFFIX,DIROUT,SAVEFILE,
     &              DATAIN
      CHARACTER SHISTFMT*1,AHISTFMT*1,KHISTFMT*1,SFILEOUT*8,AFILEOUT*8,
     &          KFILEOUT*8,SSUFFIX*3,ASUFFIX*3,KSUFFIX*3,DIROUT*72,
     &          SAVEFILE*12,DATAIN*72
      COMMON/CONST5/HDRX,HDRY,EARTHQH,EARTHQV,EQINPFMT(2),ROCKVP,ROCKVS,
     &              ROCKRHO,IHISTSTR,IHISTACC,NPLX,NPLY
      CHARACTER EQINPFMT*20,EARTHQH*72,EARTHQV*72
      INTEGER HDRX,HDRY,ESEG,ELSEG
      CHARACTER FILENAME*12,IDPROP*75
      DIMENSION IDPROP(2,*),SDATA(2,NUMPROPS,*),GDATA(NUMPROPS,*),
     &          DDATA(NUMPROPS,*),NUMPOINTS(2,*),NBC(*),NSEG(*),
     &          NOSEG(NSLP,*),ESEG(*),ELSEG(NSLP,*)
      RECORD /ELEMENT/EL(*)
      RECORD /NODE/ NO (*)
C
      WRITE(*,82)
```

```
      WRITE(6,82)
C
      WRITE(*,102) NELM,NDPT,NF,MAXW,MAXEL,NBP,NUMB,KGMAX,N1EQ,KGEQ,
     &             N2EQ,N3EQ,DTEQ,PRM,DRF,PRINPUT,EQMUL(1),
     &             UG(1)*EQMUL(1)
      WRITE(6,102) NELM,NDPT,NF,MAXW,MAXEL,NBP,NUMB,KGMAX,N1EQ,KGEQ,
     &             N2EQ,N3EQ,DTEQ,PRM,DRF,PRINPUT,EQMUL(1),
     &             UG(1)*EQMUL(1)
      IF (KV.EQ.2) THEN
        WRITE(*,121) EQMUL(2),UG(2)*EQMUL(2)
        WRITE(6,121) EQMUL(2),UG(2)*EQMUL(2)
      END IF
      IF (IHISTSTR+IHISTACC+NSLP.GT.0) THEN
        WRITE(*,1122) IHISTSTR,IHISTACC,NSLP
        WRITE(*,1131)
        WRITE(6,1122) IHISTSTR,IHISTACC,NSLP
        WRITE(6,1131)
      END IF
C
C     Open stress history output files
C
      IF (IHISTSTR.GT.0) THEN
        IF(SHISTFMT.EQ.'C'.OR.SHISTFMT.EQ.'c') THEN
          FILENAME=SFILEOUT(:LEN_TRIM(SFILEOUT))//'.'//SSUFFIX
          OPEN (10,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
          WRITE (10,'(A/A)') TITLE,
     &       ' Stress Histories '
          WRITE (10,'(A,\)') 'Time-sec'
          DO N=1,NELM
            IF (EL(N).LSTR.EQ.1.OR.EL(N).LSTR.EQ.3.OR.EL(N).LSTR.EQ.5
     &         .OR.EL(N).LSTR.EQ.7) THEN
              WRITE (10,'(I7,A\)') N,'X'
              WRITE(*,1132) 'ELEMENT',N,'SIGMA X',FILENAME
              WRITE(6,1132) 'ELEMENT',N,'SIGMA X',FILENAME
            END IF
            IF (EL(N).LSTR.EQ.2.OR.EL(N).LSTR.EQ.3.OR.EL(N).LSTR.EQ.6
     &         .OR.EL(N).LSTR.EQ.7) THEN
              WRITE (10,'(I7,A\)') N,'Y'
              WRITE(*,1132) 'ELEMENT',N,'SIGMA Y',FILENAME
              WRITE(6,1132) 'ELEMENT',N,'SIGMA Y',FILENAME
            END IF
            IF (EL(N).LSTR.EQ.4.OR.EL(N).LSTR.EQ.5.OR.EL(N).LSTR.EQ.6
     &         .OR.EL(N).LSTR.EQ.7) THEN
              WRITE (10,'(I6,A\)') N,'XY'
              WRITE(*,1132) 'ELEMENT',N,'TAU XY',FILENAME
              WRITE(6,1132) 'ELEMENT',N,'TAU XY',FILENAME
            END IF
          END DO
          WRITE (10,'(/,\)')
        ELSE
          I=0
          DO N=1,NELM
            IF (EL(N).LSTR.EQ.1.OR.EL(N).LSTR.EQ.3.OR.EL(N).LSTR.EQ.5
     &         .OR.EL(N).LSTR.EQ.7) THEN
              I=I+1
              CALL OPENSTRES(I,FILENAME)
              WRITE (I+9,'(A/A/A,F10.3,A)') TITLE,
     &          ' Stress History (ENG: PSF or SI: N/M^2)',
     &          ' Time Step = ',DTEQ,' Sec'
              WRITE (I+9,'(A,I4,A)') ' Elm. ',N,'X'
              WRITE(*,1132) 'ELEMENT',N,'SIGMA X',FILENAME
              WRITE(6,1132) 'ELEMENT',N,'SIGMA X',FILENAME
            END IF
            IF (EL(N).LSTR.EQ.2.OR.EL(N).LSTR.EQ.3.OR.EL(N).LSTR.EQ.6
     &         .OR.EL(N).LSTR.EQ.7) THEN
              I=I+1
              CALL OPENSTRES(I,FILENAME)
              WRITE (I+9,'(A/A/A,F10.3,A)') TITLE,
     &          ' Stress History (ENG: PSF or SI: N/M^2)',
     &          ' Time Step = ',DTEQ,' Sec'
              WRITE (I+9,'(A,I4,A)') ' Elm. ',N,'Y'
              WRITE(*,1132) 'ELEMENT',N,'SIGMA Y',FILENAME
              WRITE(6,1132) 'ELEMENT',N,'SIGMA Y',FILENAME
            END IF
            IF (EL(N).LSTR.EQ.4.OR.EL(N).LSTR.EQ.5.OR.EL(N).LSTR.EQ.6
     &         .OR.EL(N).LSTR.EQ.7) THEN
              I=I+1
              CALL OPENSTRES(I,FILENAME)
              WRITE (I+9,'(A/A/A,F10.3,A)') TITLE,
     &          ' Stress History (ENG: PSF or SI: N/M^2)',
     &          ' Time Step = ',DTEQ,' Sec'
              WRITE (I+9,'(A,I4,A)') ' Elm ',N,'XY'
              WRITE(*,1132) 'ELEMENT',N,'TAU XY',FILENAME
              WRITE(6,1132) 'ELEMENT',N,'TAU XY',FILENAME
            END IF
          END DO
        END IF
      END IF
C
C     Open acceleration output files
C
      IF (IHISTACC.GT.0) THEN
        IF(AHISTFMT.EQ.'C'.OR.AHISTFMT.EQ.'c') THEN
          FILENAME=AFILEOUT(:LEN_TRIM(AFILEOUT))//'.'//ASUFFIX
          OPEN (40,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
          WRITE (40,'(A/A)') TITLE,
     &       ' Acceleration Histories (g''s)'
          WRITE (40,'(A,\)') '  Time-sec'
          DO N=1,NF/2
            IF (NO(N).OUT.EQ.1.OR.NO(N).OUT.EQ.3) THEN
              WRITE (40,'(A,I4,A\)') ' Node',N,'X'
              WRITE(*,1132) 'NODE ',N,'X DIR',FILENAME
              WRITE(6,1132) 'NODE ',N,'X DIR',FILENAME
            END IF
            IF (NO(N).OUT.EQ.2.OR.NO(N).OUT.EQ.3) THEN
              WRITE (40,'(A,I4,A\)') ' Node',N,'Y'
              WRITE(*,1132) 'NODE ',N,'Y DIR',FILENAME
              WRITE(6,1132) 'NODE ',N,'Y DIR',FILENAME
```

```
            END IF
          END DO
          WRITE (40,'(/,\)')
        ELSE
          I=0
          DO N=1,NF/2
            IF (NO(N).OUT.EQ.1.OR.NO(N).OUT.EQ.3) THEN
              I=I+1
              FILENAME=AFILEOUT(:LEN_TRIM(AFILEOUT))//'0'//
     &                 CHAR(47+I)//'.'//ASUFFIX
              OPEN(I+39,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
              WRITE (I+39,'(A/A/A,F10.3,A)') TITLE,
     &            '  Acceleration History (g''s)',
     &            ' Time Step = ',DTEQ,' Sec'
              WRITE (I+39,'(A,I4,A)') ' Node ',N,'X'
              WRITE(*,1132) 'NODE ',N,'X DIR',FILENAME
              WRITE(6,1132) 'NODE ',N,'X DIR',FILENAME
            END IF
            IF (NO(N).OUT.EQ.2.OR.NO(N).OUT.EQ.3) THEN
              I=I+1
              FILENAME=AFILEOUT(:LEN_TRIM(AFILEOUT))//'0'//
     &                 CHAR(47+I)//'.'//ASUFFIX
              OPEN(I+39,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
              WRITE (I+39,'(A/A/A,F10.3,A)') TITLE,
     &            '  Acceleration History (g''s)',
     &            ' Time Step = ',DTEQ,' Sec'
              WRITE (I+39,'(A,I4,A)') ' Node ',N,'Y'
              WRITE(*,1132) 'NODE ',N,'Y DIR',FILENAME
              WRITE(6,1132) 'NODE ',N,'Y DIR',FILENAME
            END IF
          END DO
        END IF
      END IF
C
C     Open seismic coefficient output files
C
      IF (NSLP.GT.0) THEN
        IF(KHISTFMT.EQ.'C'.OR.KHISTFMT.EQ.'c') THEN
          FILENAME=KFILEOUT(:LEN_TRIM(KFILEOUT))//'.'//KSUFFIX
          OPEN (50,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
          WRITE (50,'(A/A)') TITLE,
     &         ' Seismic Coefficient Surface Histories'
          IF (KV.EQ.1) THEN
            WRITE (50,'(A,\)') '  Time-sec'
            DO N=1,NSLP
              WRITE (50,'(A8,I2,\)') ' Block ',N
              WRITE(*,1132) 'SURFACE',N,'X DIR',FILENAME
              WRITE(6,1132) 'SURFACE',N,'X DIR',FILENAME
            END DO
          ELSE
            WRITE (50,'(A,\)') '  Time-sec'
            DO N=1,NSLP
              WRITE (50,'(A8,I2,10(1H )\)') ' Block ',N
            END DO
            WRITE (50,'(/,A,\)') '              '
            DO N=1,NSLP
              WRITE (50,'(2A\)') '    X-dir ','    Y-dir '
              WRITE(*,1132) 'SURFACE',N,'X DIR',FILENAME
              WRITE(6,1132) 'SURFACE',N,'X DIR',FILENAME
              WRITE(*,1132) 'SURFACE',N,'Y DIR',FILENAME
              WRITE(6,1132) 'SURFACE',N,'Y DIR',FILENAME
            END DO
          END IF
          WRITE (50,'(/,\)')
        ELSE
          I=0
          IF (KV.EQ.1) THEN
            IF (NSLP.GT.10) THEN
              WRITE (*,'(/A,A)') '   PROGRAM TERMINATED DUE TO USE OF ',
     &         'MORE THAN 10 SEISMIC COEFF SURFACES AND MULTIPLE OUTPUT'
              STOP
            END IF
            DO N=1,NSLP
              I=I+1
              FILENAME=KFILEOUT(:LEN_TRIM(KFILEOUT))//'0'//
     &                 CHAR(47+I)//'.'//KSUFFIX
              OPEN(I+49,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
              WRITE (I+49,'(A/A/A,F10.3,A)') TITLE,
     &            '   Seismic Coefficient Surface History',
     &             ' Time Step = ',DTEQ,' Sec'
              WRITE(I+49,'(A10,I3/)') '  Surface ',N
              WRITE(*,1132) 'SURFACE',N,'X DIR',FILENAME
              WRITE(6,1132) 'SURFACE',N,'X DIR',FILENAME
            END DO
          ELSE
            IF (NSLP.GT.5) THEN
              WRITE (*,'(/A,A,A)') '   PROGRAM TERMINATED DUE TO USE ',
     &         'OF MORE THAN 5 SEISMIC COEFF SURFACES, MULTIPLE ',
     &         'OUTPUT, AND BIDIRECTIONAL SHAKING'
              STOP
            END IF
            DO N=1,NSLP
              DO IADD=0,1
                I=I+1
                FILENAME=KFILEOUT(:LEN_TRIM(KFILEOUT))//'0'//
     &                   CHAR(47+I)//'.'//KSUFFIX
                OPEN(I+49,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
                WRITE (I+49,'(A/A/A,F10.3,A)') TITLE,
     &            '   Seismic Coefficient Surface History',
     &             ' Time Step = ',DTEQ,' Sec'
                WRITE(I+49,'(A10,I3\)') '  Surface ',N
                IF (IADD.EQ.0) THEN
                  WRITE(I+49,'(A13)') ' X-dir '
                  WRITE(*,1132) 'SURFACE',N,'X DIR',FILENAME
                  WRITE(6,1132) 'SURFACE',N,'X DIR',FILENAME
                ELSE
                  WRITE(I+49,'(A13)') ' Y-dir '
                  WRITE(*,1132) 'SURFACE',N,'Y DIR',FILENAME
```

```
            WRITE(6,1132) 'SURFACE',N,'Y DIR',FILENAME
          END IF
        END DO
      END DO
    END IF
  END IF
END IF
IF (KSAV.EQ.1) THEN
  WRITE (*,1133) SAVEFILE
  WRITE (6,1133) SAVEFILE
END IF
WRITE (6,1009) DATAIN
DO I=1, NUMPROPS
  WRITE(6,1004) I,IDPROP(1,I),IDPROP(2,I)
  WRITE(6,1005)
  M=MAX0(NUMPOINTS(1,I),NUMPOINTS(2,I))
  DO J=1,M
    IF (J.LE.NUMPOINTS(1,I).AND.J.LE.NUMPOINTS(2,I)) THEN
      WRITE(6,1006)SDATA(1,I,J),GDATA(I,J),SDATA(2,I,J),DDATA(I,J)
    ELSE IF (J.LE.NUMPOINTS(1,I)) THEN
      WRITE(6,1007)SDATA(1,I,J),GDATA(I,J)
    ELSE
      WRITE(6,1008)SDATA(2,I,J),DDATA(I,J)
    END IF
  END DO
END DO
IF (UNITS.EQ.'E') THEN
  IF (ROCKRHO.NE.0) THEN
    WRITE (6,1010) ROCKRHO*GRAV,'PCF',ROCKBETAS/ROCKRHO,'FT/SEC',
&                  ROCKBETAP/ROCKRHO,'FT/SEC'
  ELSE
    WRITE (6,1010) 0.,'PCF',0.,'FT/SEC',0.,'FT/SEC'
  END IF
ELSE
  IF (ROCKRHO.NE.0) THEN
    WRITE (6,1010) ROCKRHO*GRAV,'N/M^3 ',ROCKBETAS/ROCKRHO,
&                  'M/SEC',ROCKBETAP/ROCKRHO,'M/SEC'
  ELSE
    WRITE (6,1010) 0.,'N/M^3 ',0.,'M/SEC',0.,'M/SEC'
  END IF
END IF
IF (NSLP.GT.0) THEN
  WRITE(6,92)
END IF
DO N=1,NSLP
  WRITE(6,93) N,NSEG(N),ESEG(N)
  WRITE(6,94) (NOSEG(N,K),K=1,NSEG(N))
  WRITE(6,95)
  WRITE(6,94) (ELSEG(N,K),K=1,ESEG(N))
END DO
IF (UNITS.EQ.'E') THEN
  WRITE(6,122)
ELSE
  WRITE(6,123)
```

```
      END IF
      WRITE(6,132) (N,(EL(N).NODE(K),K=1,4),
     &             EL(N).TYPE,EL(N).DENS,EL(N).PO,EL(N).GMX,
     &             EL(N).G/1000.,EL(N).XL,EL(N).AREA,N=1,NELM)
      WRITE(6,142)
C
C     Boundary condition for dynamic computation
C
      N=0
      GINIT=0.
      DO M=1,NDPT
        CALL BOUNDARY(NO(M),M,N,NBC)
        GINIT=MAX(GINIT,ABS(NO(M).X2I(1)),ABS(NO(M).X2I(2)),
     &          ABS(NO(M).X1I(1)),ABS(NO(M).X1I(2)),
     &          ABS(NO(M).XI(1)),ABS(NO(M).XI(2)))
      END DO
      IF(N .NE. NBP) THEN
        WRITE(*,1098)
        WRITE(6,1098)
        STOP
      END IF
C
C     Initial Conditions Echo
C
      IF (GINIT.GT.0) THEN
        WRITE(*,99)
        WRITE(6,1099)
        DO M=1,NDPT
          WRITE(6,1100) M,NO(M).X2I(1),NO(M).X1I(1),NO(M).XI(1),
     &                  NO(M).X2I(2),NO(M).X1I(2),NO(M).XI(2)
        END DO
      END IF
      RETURN
   82 FORMAT(///'1')
   92 FORMAT(///'1'/' SEISMIC COEFFICIENT SURFACE DATA:')
   93 FORMAT(/' BLOCK:',I3,5X,'NSEG:',I4,5X,'ESEG:',I4,/
     &       '    SURFACE PASSES THROUGH NODES:')
   94 FORMAT(15I5)
   95 FORMAT ('    SURFACE INCLUDES ELEMENTS:')
   99 FORMAT (///'1'/' INITIAL CONDITIONS USED AT THE NODES:',/,' NODE',
     &       8X,'X2IH',8X,'X1IH',9X,'XIH',
     &       8X,'X2IV',8X,'X1IV',9X,'XIV'/)
  102 FORMAT(/
     & 20X,'NO. OF ELEMENTS =',I6/
     & 16X,'NO. OF NODAL POINTS =',I6/
     & 17X,'DEGREES OF FREEDOM =',I6/
     & 21X,'HALF-BANDWIDTH =',I6/
     & 16X,'CONTROLLING ELEMENT =',I6/
     & 10X,'NO. OF FIXED BNDRY CONDS. =',I6/
     & 18X,'NO. OF ITERATIONS =',I6/
     & 6X,'TOTAL EQ. POINTS READ (KGMAX) =',I6/
     & 1X,'LAST EQ. PTS. USED (N1EQ TO KGEQ)  =',2I6/
     & 3X,'INT. EQ. PTS USED (N2EQ TO N3EQ) =',2I6/
     & 11X,'TIME INTERVAL OF RECORDS = ',F8.4,'  SECONDS'/
```

```
     &  11X,'STRAIN CONVERSION FACTOR = ',F8.4/
     &  5X,'DAMPING RATIO REDUCTION FACTOR = ',F8.3/
     &  4X,'PREDOMINANT INPUT MOTION PERIOD = ',F8.4,'  SECONDS'/
     &  5X,'EQ. MULT. FACTOR (HORZ. COMP.) = ',F8.4/
     &  2X,'MAXIMUM ACCEL. USED (HORZ. COMP.) = ',F8.4)
  121 FORMAT(5X,'EQ. MULT. FACTOR (VERT. COMP.) = ',F8.4/
     &  2X,'MAXIMUM ACCEL. USED (VERT. COMP.) = ',F8.4)
  122 FORMAT (///'1'/'   ELM',5X,'NODE 1',2X,'NODE 2',2X,'NODE 3',2X,
     &  'NODE 4',2X,'MAT.TYPE',2X,'DENSITY',4X,'POISSON R.',
     &  5X,'GMX',4X,'SH. MODULUS',2X,'DAMP. RATIO',5X,'AREA'/
     &  54X,'(PCF)',T79,'(KSF)',T90,'(KSF)',T115,'(FT^2)'/)
  123 FORMAT (///'1'/'   ELM',5X,'NODE 1',2X,'NODE 2',2X,'NODE 3',2X,
     &  'NODE 4',2X,'MAT.TYPE',2X,'DENSITY',4X,'POISSON R.',
     &  5X,'GMX',4X,'SH. MODULUS',2X,'DAMP. RATIO',5X,'AREA'/
     &  53X,'(N/M^3)',T77,'(KN/M^2)',T89,'(KN/M^2)',T116,'(M^2)'/)
  132 FORMAT (6I8,6F12.3)
  142 FORMAT(///'1'/4X,'NODE',11X,'XORD',11X,'YORD',7X,'TRIBUTARY LEN'/)
 1004 FORMAT(/' **********************'/
     &        '  MATERIAL TYPE NO.',I2,/
     &      ' **********************'/' MODULUS: ',A75/' DAMPING: ',A75)
 1005 FORMAT(  '  ===================     ==================='/
     &          '     STRAIN    G/Gmax       STRAIN    DAMPING'/
     &          '    --------   -------     --------  --------')
 1006 FORMAT(1X,F9.4,4X,F6.3,6X,F9.4,4X,F6.2)
 1007 FORMAT(1X,F9.4,4X,F6.3)
 1008 FORMAT(26X,F9.4,4X,F6.2)
 1009 FORMAT (/'  SOIL DATA TAKEN FROM FILE: ',A)
 1010 FORMAT (/' UNDERLYING STRATUM DATA:'//5X,'UNIT WEIGHT: ',
     &        F10.3,2X,A,/5X,'SHEAR WAVE VELOCITY: ',F10.2,2X,A,
     &        /5X,'COMPRESSION WAVE VELOCITY: ',F10.2,2X,A)
 1098 FORMAT (' THE TOTAL NUMBER OF BOUNDARY CONDITIONS DOES NOT AGRE'
     &        'E WITH THAT GIVEN BEFORE BY NBP')
 1099 FORMAT (///'1'/' INITIAL CONDITIONS USED AT THE NODES:',/,' NODE',
     &        9X,'X2IH',9X,'X1IH',10X,'XIH',
     &        9X,'X2IV',9X,'X1IV',10X,'XIV'/)
 1100 FORMAT (I5,6E13.6)
 1122 FORMAT(/2X,I4,' STRESS HISTORIES REQUESTED.',
     &       /2X,I4,' ACCEL HISTORIES REQUESTED.',
     &       /2X,I4,' SEIS COEFF HISTORIES REQUESTED')
 1131 FORMAT(10X,'OUTPUT FILES ARE AS FOLLOWS:',/)
 1132 FORMAT(A8,I4,'. ',A7,' IN FILE:  ',A12)
 1133 FORMAT (/' FINAL CONDITIONS OF NODES SAVED FOR RESTART',
     &          ' IN FILE:  ',A12)
      END
C
C ***********************************************************************
C
      SUBROUTINE INIT(EL,U2G)
C
C     Initializes Earthquake and Modulus Data
C
C                          Written for QUAD4M
C                          U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                          Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      RECORD /ELEMENT/EL(*)
      DIMENSION U2G(0:KGMAX,*)
C
C     Find earthquake factor if none specified
C
      DO N = 1,KV
        UG(N) = ABS(U2G(1,N))
        DO L = 2,KGMAX
          IF (ABS(U2G(L,N)).GT.UG(N)) UG(N) = ABS(U2G(L,N))
        END DO
      END DO
      IF (EQMUL(1).EQ.0.) THEN
        DO N = 1,KV
          EQMUL(N) = UGMAX(N)/UG(N)
        END DO
      END IF
C
C     Convert accel to ft/sec^2 or m/sec^2, and factor earthquake
C
      DO N = 1,KV
        DO L = 1,KGEQ
          U2G(L,N) = GRAV*EQMUL(N)*U2G(L,N)
        END DO
      END DO
C
C     Initialization of Modulus Values
C
      DO N=1,NELM
        EL(N).G=1000.*EL(N).G ! Conv. from KSF to PSF or KN/M^2 to N/M^2
        EL(N).E=EL(N).G*2.*(1.+EL(N).PO)        ! Modulus from G and nu
        EL(N).EN=EL(N).E
      END DO
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE INPT(EL,NO,U2G,IDPROP,SDATA,GDATA,DDATA,NUMPROPS,
     &                NUMPOINTS,NOSEG,NSEG,ESEG,ELSEG)
C
C     Reads the element data, the nodal point data,
```

```
C     the horizontal component and vertical component (if any) of
C     the earthquake, and the soil data.
C
C                            Written for QUAD4M
C                             U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                        Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      COMMON/CONST4/SHISTFMT,AHISTFMT,KHISTFMT,SFILEOUT,AFILEOUT,
     &              KFILEOUT,SSUFFIX,ASUFFIX,KSUFFIX,DIROUT,SAVEFILE,
     &              DATAIN
      CHARACTER SHISTFMT*1,AHISTFMT*1,KHISTFMT*1,SFILEOUT*8,AFILEOUT*8,
     &          KFILEOUT*8,SSUFFIX*3,ASUFFIX*3,KSUFFIX*3,DIROUT*72,
     &          SAVEFILE*12,DATAIN*72
      COMMON/CONST5/HDRX,HDRY,EARTHQH,EARTHQV,EQINPFMT(2),ROCKVP,ROCKVS,
     &              ROCKRHO,IHISTSTR,IHISTACC,NPLX,NPLY
      CHARACTER EQINPFMT*20,EARTHQH*72,EARTHQV*72
      INTEGER HDRX,HDRY,ESEG,ELSEG
      CHARACTER*75 IDPROP
      RECORD /ELEMENT/EL(*)
      RECORD /NODE/NO(*)
      DIMENSION U2G(0:KGMAX,*),IDPROP(2,*),SDATA(2,NUMPROPS,*),
     &          GDATA(NUMPROPS,*),DDATA(NUMPROPS,*),NUMPOINTS(2,*),
     &          NSEG(*),NOSEG(NSLP,*),ESEG(*),ELSEG(NSLP,*)
C
      IF (KV.EQ.2) THEN
        DO I=0,KGMAX
          DO J=1,2
             U2G(I,J)=0.
          END DO
        END DO
      ELSE
        DO I=0,KGMAX
           U2G(I,1)=0.
        END DO
      END IF
C
C     Read Seismic Coefficient Surface Information
C
```

```
      IF (NSLP.GT.0) THEN
        DO J=1,NSLP
          READ (5,'(/\)')
          READ(5,'(2I5)') NSEG(J),ESEG(J)
          READ (5,'(/\)')
          READ(5,'(15I5)') (NOSEG(J,I),I=1,NSEG(J))
          READ (5,'(/\)')
          READ(5,'(15I5)') (ELSEG(J,I),I=1,ESEG(J))
        END DO
      END IF
C
C     Read Element Information
C
      READ (5,'(/\)')
      DO J=1,NELM
        READ(5,'(6I5,5F10.0,I5)') N,(EL(N).NODE(L),L=1,4),EL(N).TYPE,
     &                            EL(N).DENS,EL(N).PO,EL(N).GMX,EL(N).G,
     &                            EL(N).XL,EL(N).LSTR
c     Reorder nodes if not according to specifications
        DO I=1,2
          DO K=I+1,4
            IF(EL(N).NODE(I).EQ.EL(N).NODE(K)) THEN
              WRITE(6,*) 'Nodes reordered for element: ',N
              IF (I.EQ.1.AND.K.EQ.2) THEN
                EL(N).NODE(1)=EL(N).NODE(3)
                EL(N).NODE(3)=EL(N).NODE(2)
                EL(N).NODE(2)=EL(N).NODE(4)
                EL(N).NODE(4)=EL(N).NODE(3)
              ELSE IF (I.EQ.2.AND.K.EQ.3) THEN
                EL(N).NODE(2)=EL(N).NODE(1)
                EL(N).NODE(1)=EL(N).NODE(4)
                EL(N).NODE(4)=EL(N).NODE(3)
              ELSE IF (I.EQ.1.AND.K.EQ.4) THEN
                EL(N).NODE(1)=EL(N).NODE(2)
                EL(N).NODE(2)=EL(N).NODE(3)
                EL(N).NODE(3)=EL(N).NODE(4)
              ELSE
                PRINT*,'Element ',N,' has improperly repeating nodes'
                STOP
              END IF
            END IF
          END DO
        END DO
      END DO
C
C     Read Node Information
C
      READ (5,'(/\)')
      DO N=1,NDPT
        READ(5,'(I5,2F10.0,2I5,6F13.0)') I,NO(I).XORD,NO(I).YORD,
     &                                   NO(I).BC,NO(I).OUT,
     &                         NO(I).X2I(1),NO(I).X1I(1),NO(I).XI(1),
     &                         NO(I).X2I(2),NO(I).X1I(2),NO(I).XI(2)
      END DO
```

```
C
C     Verify that no seismic coefficient surfaces go through base
C
      IF (NSLP.GT.0) THEN
        DO J=1,NSLP
          DO I=1,NSEG(J)
            IF (NO(NOSEG(J,I)).BC.NE.0) THEN
              WRITE (*,'(A,I4,A,A)') ' NODE ',NOSEG(J,I),' IS NOT A',
     &   ' FREE NODE AND SHOULDN''T BE USED IN A SEISMIC COEFF SURFACE!'
              STOP
            END IF
          END DO
        END DO
      END IF
C
C       Find area of elements
C
      DO J=1,NELM
        X1=NO(EL(J).NODE(1)).XORD
        X2=NO(EL(J).NODE(2)).XORD
        X3=NO(EL(J).NODE(3)).XORD
        X4=NO(EL(J).NODE(4)).XORD
        Y1=NO(EL(J).NODE(1)).YORD
        Y2=NO(EL(J).NODE(2)).YORD
        Y3=NO(EL(J).NODE(3)).YORD
        Y4=NO(EL(J).NODE(4)).YORD
        EL(J).AREA=(Y2-Y4)*(X1+X2-X3-X4)
        EL(J).AREA=EL(J).AREA-(X2-X4)*(Y1+Y2-Y3-Y4)
        EL(J).AREA=0.5*EL(J).AREA
        IF (EL(J).AREA.LE.0.) THEN
          PRINT*,     ' ELEMENT',J,' HAS 0 OR NEGATIVE AREA.  ABORTED'
          WRITE (6,*) ' ELEMENT',J,' HAS 0 OR NEGATIVE AREA.  ABORTED'
          STOP
        END IF
      END DO
C
C     Find Tributary Length of base nodes
C
      DO L=1,NDPT
        NO(L).TRIBLEN=0.
        IF (NO(L).BC.EQ.4) THEN
          DO K=1,NELM
            DO N=1,4
              IF (EL(K).NODE(N).EQ.L) THEN
                IF (N.EQ.1) THEN
                  M=4
                ELSE
                  M=N-1
                END IF
                IF (NO(EL(K).NODE(M)).BC.EQ.4) THEN
                  NO(L).TRIBLEN=NO(L).TRIBLEN+
     &              ABS(NO(EL(K).NODE(M)).XORD-NO(L).XORD)
                END IF
                IF (N.EQ.4) THEN
                  M=1
                ELSE
                  M=N+1
                END IF
                IF (NO(EL(K).NODE(M)).BC.EQ.4) THEN
                  NO(L).TRIBLEN=NO(L).TRIBLEN+
     &              ABS(NO(EL(K).NODE(M)).XORD-NO(L).XORD)
                END IF
                EXIT
              END IF
            END DO
          END DO
          NO(L).TRIBLEN=NO(L).TRIBLEN/2.
        END IF
      END DO
C
C     Read Soil Information
C
      PRINT*, ' Read Soil Properties File... '
      READ (7,'(I5)') NUMPROPS
      DO I=1,NUMPROPS
        READ (7,'(I5,A75)') NUMPOINTS(1,I),IDPROP(1,I)
        READ (7,'(8F10.0)') (SDATA(1,I,J),J=1,NUMPOINTS(1,I))
        READ (7,'(8F10.0)') (GDATA(I,J),J=1,NUMPOINTS(1,I))
        READ (7,'(I5,A75)') NUMPOINTS(2,I),IDPROP(2,I)
        READ (7,'(8F10.0)') (SDATA(2,I,J),J=1,NUMPOINTS(2,I))
        READ (7,'(8F10.0)') (DDATA(I,J),J=1,NUMPOINTS(2,I))
      END DO
C
C     Read Acceleration Input File
C
      PRINT*, ' Read Earthquake File... '
      IF (HDRX.LT.1) THEN
        PRINT*, ' There must be at least one header line in the',
     &          ' acceleration input files.'
        STOP
      END IF
      READ(8,'(A)') EQNH
      DO I=2,HDRX
        READ(8,'(/\)')
      END DO
      IF (NPLX.LT.1) THEN
        IF (EQINPFMT(1)(:1).EQ.'*') THEN
          READ(8,*) (U2G(I,1),I=1,KGMAX)
        ELSE
          READ(8,EQINPFMT(1)) (U2G(I,1),I=1,KGMAX)
        END IF
      ELSE
        NCARDS=(KGMAX-1)/NPLX+1
        IF (EQINPFMT(1)(:1).EQ.'*') THEN
          DO I=1,NCARDS
            IF (I.NE.NCARDS) THEN
              READ(8,*) (U2G(J,1),J=(I-1)*NPLX+1,I*NPLX)
            ELSE
```

```
          READ(8,*) (U2G(J,1),J=(I-1)*NPLX+1,KGMAX)
         END IF
        END DO
       ELSE
        DO I=1,NCARDS
         IF (I.NE.NCARDS) THEN
          READ(8,EQINPFMT(1)) (U2G(J,1),J=(I-1)*NPLX+1,I*NPLX)
         ELSE
          READ(8,EQINPFMT(1)) (U2G(J,1),J=(I-1)*NPLX+1,KGMAX)
         END IF
        END DO
       END IF
      END IF
      IF (KV.EQ.2) THEN
       IF (HDRY.LT.1) THEN
        PRINT*, ' There must be at least one header line in the',
     &      ' acceleration input files.'
        STOP
       END IF
       READ(9,'(A)') EQNV
       DO I=2,HDRY
        READ(9,'(/\)')
       END DO
       IF (NPLY.LT.1) THEN
        IF (EQINPFMT(2)(:1).EQ.'*') THEN
         READ(9,*) (U2G(I,2),I=1,KGMAX)
        ELSE
         READ(9,EQINPFMT(2)) (U2G(I,2),I=1,KGMAX)
        END IF
       ELSE
        NCARDS=(KGMAX-1)/NPLY+1
        IF (EQINPFMT(2)(:1).EQ.'*') THEN
         DO I=1,NCARDS
          IF (I.NE.NCARDS) THEN
           READ(9,*) (U2G(J,2),J=(I-1)*NPLY+1,I*NPLY)
          ELSE
           READ(9,*) (U2G(J,2),J=(I-1)*NPLY+1,KGMAX)
          END IF
         END DO
        ELSE
         DO I=1,NCARDS
          IF (I.NE.NCARDS) THEN
           READ(9,EQINPFMT(2)) (U2G(J,2),J=(I-1)*NPLY+1,I*NPLY)
          ELSE
           READ(9,EQINPFMT(2)) (U2G(J,2),J=(I-1)*NPLY+1,KGMAX)
          END IF
         END DO
        END IF
       END IF
      END IF
      RETURN
      END
C
C ***********************************************************************
```

```
C
      SUBROUTINE OPENSTRES(I,FILENAME)
C
C     Opens Stress History Files
C
C                Written for QUAD4M
C                 U.C. Davis, 1993
C        by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      COMMON/CONST4/SHISTFMT,AHISTFMT,KHISTFMT,SFILEOUT,AFILEOUT,
     &             KFILEOUT,SSUFFIX,ASUFFIX,KSUFFIX,DIROUT,SAVEFILE,
     &             DATAIN
      CHARACTER SHISTFMT*1,AHISTFMT*1,KHISTFMT*1,SFILEOUT*8,AFILEOUT*8,
     &         KFILEOUT*8,SSUFFIX*3,ASUFFIX*3,KSUFFIX*3,DIROUT*72,
     &         SAVEFILE*12,DATAIN*72
      CHARACTER FILENAME*12
C
      IF (I.LT.11) THEN
       FILENAME=SFILEOUT(:LEN_TRIM(SFILEOUT))//'0'//CHAR(47+I)//
     &               '.'//SSUFFIX
      ELSE IF (I.LT.21) THEN
       FILENAME=SFILEOUT(:LEN_TRIM(SFILEOUT))//'1'//CHAR(37+I)//
     &               '.'//SSUFFIX
      ELSE IF (I.LT.31) THEN
       FILENAME=SFILEOUT(:LEN_TRIM(SFILEOUT))//'2'//CHAR(27+I)//
     &               '.'//SSUFFIX
      ELSE
       PRINT*, ' NUMBER OF STRESS HIST REQUESTED TOO LARGE'
       STOP
      END IF
      OPEN(I+9,FILE=DIROUT(:LEN_TRIM(DIROUT))//FILENAME)
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE SOILSCAN(NUMPROPS,MAXPOINTS)
C
C     Finds size of soil properties input file
C
C                Written for QUAD4M
C                 U.C. Davis, 1993
C        by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      MAXPOINTS=0
      PRINT*, ' Read Soil Properties File... '
      READ (7,'(I5)') NUMPROPS
      DO J=1,NUMPROPS
       DO K=1,2
        READ (7,'(I5)') NUMPOINTS
        IF (NUMPOINTS.GT.MAXPOINTS) MAXPOINTS=NUMPOINTS
        READ (7,'(8F10.0)') (DATA,I=1,2*NUMPOINTS)
       END DO
      END DO
```

```
      REWIND 7
      RETURN
      END
```

```
      SUBROUTINE QUD4D(EL,NO,ZMASS,SMASS,R,AT,BT,X1,X2,DSMAX,TIME1,NBC,
     &                 U2G,X,ST,DS,MAXW,IHISTSTR,IHISTACC,SDATA,GDATA,
     &                 DDATA,NUMPROPS,NUMPOINTS,ROCKBETAS,ROCKBETAP,
     &                 NSEG,NOSEG,ESEG,ELSEG)
C
C     Controlling Finite Element Computation Subroutine
C
C                           Written for QUAD4M
C                            U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                         Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      COMMON/CONST4/SHISTFMT,AHISTFMT,KHISTFMT,SFILEOUT,AFILEOUT,
     &             KFILEOUT,SSUFFIX,ASUFFIX,KSUFFIX,DIROUT,SAVEFILE,
     &             DATAIN
      CHARACTER SHISTFMT*1,AHISTFMT*1,KHISTFMT*1,SFILEOUT*8,AFILEOUT*8,
     &          KFILEOUT*8,SSUFFIX*3,ASUFFIX*3,KSUFFIX*3,DIROUT*72,
     &          SAVEFILE*12,DATAIN*72
      INTEGER*2 IHR1,IMIN1,ISEC1,I100TH,IHR2,IMIN2,ISEC2,
     &          IYR1,IMON1,IDAY1,IYR2,IMON2,IDAY2
      INTEGER ESEG(*),ELSEG(NSLP,*)
      RECORD /ELEMENT/EL(*)
      RECORD /NODE/NO(*)
      DIMENSION ZMASS(*),SMASS(*),R(*),AT(*),BT(*),X1(*),X2(*),DSMAX(*),
     &          TIME1(*),NBC(*),U2G(0:KGMAX,*),X(*),ST(NF,*),DS(NF,*),
     &          SDATA(2,NUMPROPS,*),GDATA(NUMPROPS,*),DDATA(NUMPROPS,*),
     &          NUMPOINTS(2,*),NSEG(*),NOSEG(NSLP,*),
     &          ELST [ALLOCATABLE] (:,:,:),
     &          ELMASSORG [ALLOCATABLE] (:),
     &          STMOD [ALLOCATABLE] (:,:),
     &          SEISMAX [ALLOCATABLE] (:,:),
     &          WSLIP  [ALLOCATABLE] (:)
C
      ALLOCATE (ELST(NELM,8,8),ELMASSORG(NELM),STMOD(NF,MAXW))
      IF (NSLP.GT.0) THEN
        ALLOCATE (SEISMAX(2*KV,NSLP),WSLIP(NSLP))
      END IF
      CALL GETTIM(IHR1,IMIN1,ISEC1,I100TH)
      CALL GETDAT(IYR1,IMON1,IDAY1)
```

```
      DO LOOP = 1, NUMB
        IF(LOOP .GE. NUMB) THEN
          N2 = N1EQ
          N3 = KGEQ
        ELSE
          N2 = N2EQ
          N3 = N3EQ
        END IF
        DO N = 1, NF
          TIME1(N)=0.
          DSMAX(N) = 0.
        END DO
        DO N = 1, NELM
          EL(N).TIME2 = 0.
          EL(N).EPSMAX = 0.
          DO L = 1,3
            EL(N).SIG(L) = 0.
            EL(N).SIGMAX(L) = 0.
          END DO
        END DO
        WRITE(*,9205)  LOOP
        WRITE(6,9206)  LOOP
C
C       Set up stiffness and damping matrix and compute eigenvalue
C
        CALL FRMSTF(LOOP,EL,NO,ZMASS,SMASS,NBC,ST,DS,R,MAXW,ELST,
     &              ELMASSORG,ROCKBETAS,ROCKBETAP)
        CALL GETTIM(IHR2,IMIN2,ISEC2,I100TH)
        CALL GETDAT(IYR2,IMON2,IDAY2)
        TIM =60*(60*(24*(365*(IYR2-IYR1)+(MODAY(IMON2)-MODAY(IMON1))+
     &       (IDAY2-IDAY1))+(IHR2-IHR1))+(IMIN2-IMIN1))+(ISEC2-ISEC1)
        WRITE (*,1096) TIM
        WRITE (6,1096) TIM
        IYR1=IYR2
        IMON1=IMON2
        IDAY1=IDAY2
        IHR1=IHR2
        IMIN1=IMIN2
        ISEC1=ISEC2
        DO N=1,NF
          DO L=1,MAXW
            DS(N,L)=DS(N,L)*DRF
          END DO
        END DO
C
C         Form Modified Stiffness Matrix:
C
        DO L = 1, NF
          STMOD(L,1) = ST(L,1) + 2./DTEQ*DS(L,1) +
     &                 4./(DTEQ*DTEQ)*SMASS(L)
        END DO
        DO J = 2, MAXW
          DO L = 1, NF
            STMOD(L,J) = ST(L,J) + 2./DTEQ*DS(L,J)
```

```
          END DO
        END DO
C
        WRITE(*,9305)
        CALL SYMBOL(1,STMOD,R,NF,MAXW) ! Prepares Stiffness matrix
C
C       Initialization
C
        DO N = NF/2+1,2*NDPT
          X(N) = 0.
        END DO
        DO N = 1, NF/2
          X2(2*N-1) = NO(N).X2I(1)
          X2(2*N)   = NO(N).X2I(2)
          X1(2*N-1) = NO(N).X1I(1)
          X1(2*N)   = NO(N).X1I(2)
          X(2*N-1)  = NO(N).XI(1)
          X(2*N)    = NO(N).XI(2)
        END DO
        DO J=1,NSLP
          DO K=1,2*KV
            SEISMAX(K,J) = 0.
          END DO
        END DO
C
C       Start dynamic computation
C
        WRITE(*,9405)
        ILINE=0
        DO KTIME = N2, N3
          IF(ILINE.LT.8) THEN
            ILINE=ILINE+1
          ELSE
            ILINE=1
          END IF
          TIMEF=DTEQ*FLOAT(KTIME)
          WRITE(*,9505) LOOP,KTIME,TIMEF
          UG(1) = U2G(KTIME,1)
          UG(2) = 0.0
          X1BASHORZ=X1BASHORZ + DTEQ/2.*(U2G(KTIME-1,1)+U2G(KTIME,1))
          IF(KV.EQ.2) THEN
            UG(2) = U2G(KTIME,2)
            X1BASVERT=X1BASVERT + DTEQ/2.*(U2G(KTIME-1,2)+U2G(KTIME,2))
          END IF
          DO N = 1, NF
            IF (ABS(X(N)).GE.100.) THEN  ! If disp too large
              WRITE(*,82)
              WRITE(*,7775) KTIME,N,X(N)
              WRITE(6,82)
              WRITE(6,7775) KTIME,N,X(N)
              WRITE(*,8989)
              WRITE(6,8989)
              STOP
            END IF
```

```
            AT(N) = 4.*(X(N)/(DTEQ*DTEQ) + X1(N)/DTEQ + X2(N)/4.)
            BT(N) = X1(N) + X(N)*2./DTEQ
C
C           Modified Force Vector:
C
            R(N) = SMASS(N)*AT(N) + DS(N,1)*BT(N)
C
C           Add on Forces due to relative Acceleration of Points
C
            IF (MOD(N,2).NE.0) THEN
              R(N) = R(N) - SMASS(N)*UG(1)
            ELSE IF (KV.EQ.2) THEN
              R(N) = R(N) - SMASS(N)*UG(2)
            END IF
          END DO
          DO N = 1, NF
            L = N - 1
            DO J = 2, MIN0 (MAXW,NF-N+1)
              K = L + J
              R(N) = R(N) + DS(N,J)*BT(K)
              R(K) = R(K) + DS(N,J)*BT(N)
            END DO
          END DO
C
C         Solve for displacement
C
          CALL SYMBOL(2,STMOD,R,NF,MAXW)
          DO N = 1, NF
            U0=X(N)                          ! Disp at last step
            U10=X1(N)                        ! Velocity at last step
            U20=X2(N)                        ! Accel at last step
            X(N)=R(N)                        ! Present disp
            X2(N)=X(N)*4./(DTEQ*DTEQ) - AT(N)
            X1(N)=U10 + DTEQ/2.*(U20 + X2(N))
          END DO
          DO L = 1,2
            DO N = L, NF, 2
              AJ = ABS(X2(N) + UG(L))
              IF (DSMAX(N) .GE. AJ) CYCLE
              DSMAX(N)=AJ
              TIME1(N)=FLOAT(KTIME)*DTEQ
            END DO
          END DO
          DO N = 1, NELM
            CALL STRESSES(EL(N),NO,KTIME,X)
          END DO
          IF (LOOP.NE.NUMB) CYCLE
C     SAVE STATE OF SYSTEM IN FILE FOR RESTART
          IF (KTIME.EQ.N3.AND.KSAV.EQ.1) THEN
            DO N = 1, NF/2
              NO(N).X2I(1) = X2(2*N-1)
              NO(N).X2I(2) = X2(2*N)
              NO(N).X1I(1) = X1(2*N-1)
              NO(N).X1I(2) = X1(2*N)
```

```
            NO(N).XI(1)  = X(2*N-1)
            NO(N).XI(2)  = X(2*N)
          END DO
          CALL WRITEFILE(NSEG,NOSEG,ESEG,ELSEG,EL,NO)
        END IF
C       Saves requested stress histories
        IF (IHISTSTR.GT.0) THEN
          ISTRES=1
          IF (SHISTFMT.EQ.'C'.OR.SHISTFMT.EQ.'c') THEN
            WRITE(10,'(F8.3,\)') TIMEF
          END IF
          DO N=1,NELM
            IF(EL(N).LSTR.EQ.1.OR.EL(N).LSTR.EQ.3.OR.EL(N).LSTR
     &        .EQ.5.OR.EL(N).LSTR.EQ.7) THEN
              CALL STRESSOUT(SHISTFMT,EL(N).SIG(1),ILINE,ISTRES)
              ISTRES=ISTRES+1
            END IF
            IF(EL(N).LSTR.EQ.2.OR.EL(N).LSTR.EQ.3.OR.EL(N).LSTR
     &        .EQ.6.OR.EL(N).LSTR.EQ.7) THEN
              CALL STRESSOUT(SHISTFMT,EL(N).SIG(2),ILINE,ISTRES)
              ISTRES=ISTRES+1
            END IF
            IF(EL(N).LSTR.EQ.4.OR.EL(N).LSTR.EQ.5.OR.EL(N).LSTR
     &        .EQ.6.OR.EL(N).LSTR.EQ.7) THEN
              CALL STRESSOUT(SHISTFMT,EL(N).SIG(3),ILINE,ISTRES)
              ISTRES=ISTRES+1
            END IF
          END DO
          IF (SHISTFMT.EQ.'C'.OR.SHISTFMT.EQ.'c') THEN
            WRITE(10,'(/,\)')
          END IF
        END IF
C       Saves requested acceleration histories
        IF (IHISTACC.GT.0) THEN
          IACC=1
          IF (AHISTFMT.EQ.'C'.OR.AHISTFMT.EQ.'c') THEN
            WRITE(40,'(F10.3,\)') TIMEF
          END IF
          DO N=1,NF/2
            IF (NO(N).OUT.EQ.1) THEN
              CALL ACCOUT(AHISTFMT,(X2(N*2-1)+UG(1))/GRAV,ILINE,IACC)
            ELSE IF (NO(N).OUT.EQ.2) THEN
              CALL ACCOUT(AHISTFMT,(X2(N*2)+UG(2))/GRAV,ILINE,IACC)
            ELSE IF (NO(N).OUT.EQ.3) THEN
              CALL ACCOUT(AHISTFMT,(X2(N*2-1)+UG(1))/GRAV,ILINE,IACC)
              CALL ACCOUT(AHISTFMT,(X2(N*2)+UG(2))/GRAV,ILINE,IACC)
            END IF
          END DO
          IF (AHISTFMT.EQ.'C'.OR.AHISTFMT.EQ.'c') THEN
            WRITE(40,'(/,\)')
          END IF
        END IF
C       Computes and Saves requested seismic coefficient histories
        IF(NSLP.GT.0) THEN
```

```
            CALL SEISCOEF(EL,NO,KV,NOSEG,NSEG,ELSEG,ESEG,WSLIP,KHISTFMT,
     &                    ILINE,SEISMAX,TIMEF)
          END IF
        END DO
        DO N = 1, NF
          DSMAX(N)=DSMAX(N)/GRAV
        END DO
        DO N = 1, NELM
          EL(N).EPSMAX=100.*EL(N).EPSMAX    ! Convert Strain to Percent
        END DO
C
C  Prints output and Calculates strain compatible properties
C
C       Calculates and prints strain compatible properties
        WRITE(6,612)
        AR = 0.
        Z = 0.
        DO M = 1, NELM
          A = 2.*(1.+EL(M).PO)
          S = PRM*EL(M).EPSMAX    ! Convert Max Strain to Uniform Strain
          EL(M).E=EL(M).E/A       ! Convert E to G
          N = EL(M).TYPE          ! Soil Type
          IF (N.EQ.0) THEN
            D=EL(M).XL
            G=EL(M).E
          ELSE
            CALL CMPRP (N,S,D,G,DDATA,GDATA,SDATA,NUMPOINTS,NUMPROPS)
            D = D/100.             ! Convert Damping from Percent
            G=G*EL(M).GMX*1000     ! G/Gmax * Gmax, converted from kilo
          END IF
          DE=100.*(EL(M).E-G)/G
          DZ = 100.*(EL(M).XL-D)/D
          AR = AR + EL(M).AREA
          Z = Z + D*EL(M).AREA
          WRITE(6,622)  M,EL(M).E,G,DE,EL(M).XL,D,DZ
          EL(M).E=EL(M).E*A
          EL(M).EN=G*A          ! Convert G to E
          EL(M).XL = D
        END DO
        IF (LOOP.EQ.NUMB) THEN
C         Prints Nodal Peak Acc.
          WRITE(6,552)
          DO N = 1, NF/2
            WRITE(6,562) N,NO(N).XORD,NO(N).YORD,DSMAX(2*N-1),
     &                   TIME1(2*N-1),DSMAX(2*N),TIME1(2*N)
          END DO
C         Prints Element Peak Stresses
          WRITE(6,592)
          DO N = 1, NELM
            WRITE(6,602) N,(EL(N).SIGMAX(L),L=1,3),EL(N).EPSMAX,
     &                   EL(N).TIME2
          END DO
C         Prints Seismic Coefficient Max and Min
          IF (NSLP.GT.0) THEN
```

```
         WRITE(6,625)
         IF (KV.EQ.2) THEN
           WRITE(6,626)
         END IF
         WRITE(6,'(/18(1H )\)')
         WRITE(6,627)
         IF (KV.EQ.2) THEN
           WRITE(6,628)
         END IF
         WRITE(6,'(//\)')
        END IF
        DO M = 1, NSLP
         WRITE(6,623)  M,WSLIP(M),SEISMAX(1,M),SEISMAX(2,M)
         IF (KV.EQ.2) THEN
           WRITE(6,624)  SEISMAX(3,M),SEISMAX(4,M)
         END IF
         WRITE(6,'(/\)')
        END DO
       END IF
       AJ = Z/AR
       WRITE(*,652) LOOP,AJ
       WRITE(6,652) LOOP,AJ
       CALL GETTIM(IHR2,IMIN2,ISEC2,I100TH)
       CALL GETDAT(IYR2,IMON2,IDAY2)
       TIM =60*(60*(24*(365*(IYR2-IYR1)+(MODAY(IMON2)-MODAY(IMON1))+
     &          (IDAY2-IDAY1))+(IHR2-IHR1))+(IMIN2-IMIN1))+(ISEC2-ISEC1)
       WRITE (*,1097) N3-N2+1,TIM
       WRITE (6,1097) N3-N2+1,TIM
       IHR1=IHR2
       IMIN1=IMIN2
       ISEC1=ISEC2
       IMON1=IMON2
       IYR1=IYR2
       IDAY1=IDAY2
      END DO
      N=1
      WRITE(*,7676)
      WRITE(6,7676)
      RETURN
C
    2 FORMAT (12A6,I2)
   82 FORMAT(///'1')
  552 FORMAT('1'/,
     &  5X,'PEAK NODAL ACCELERATION VALUES (g''s)',//,10X,'NODE',5X,
     &  'XORD',5X,'YORD',9X,'X-ACC',7X,'AT TIME',11X,'Y-ACC',
     &  7X,'AT TIME',/)
  562 FORMAT(I14,2F9.1,4F14.4)
  592 FORMAT(///'1'/5X,'PEAK ELEMENTS STRESSES (ENG: PSF or SI: N/M^2)'
     & ' AND STRAINS',//,10X,'ELM',11X,'SIG-X',10X,'SIG-Y',9X,'SIG-XY',
     & 9X,'EPS-MAX',8X,'AT TIME'//)
  602 FORMAT (I14,3F15.1,F15.3,F15.3)
  612 FORMAT(//5X,'MODULI (ENG: KSF or SI: KN/M^2) AND DAMPING',/
     &  2X,'ELM',6X,'G-USED',7X,'G-NEW',5X,'DIF-G',3X,
     &  'DAMP-USED',4X,'DAMP-NEW',2X,'DIF-DAMP',//)
  622 FORMAT(I5,-3PF12.1,F12.1,0PF10.1,F10.5,F12.5,F10.1)
  623 FORMAT(I9,5X,3F12.4,\)
  624 FORMAT(2F12.4\)
  625 FORMAT(///'1  MAX & MIN SEISMIC COEFFICIENTS'/'    SURFACE   ',
     &          'WEIGHT(LB or N)   X-DIRECTION'\)
  626 FORMAT(                              '                    Y-DIRECTION'\)
  627 FORMAT(12X,'  NEGATIVE   POSITIVE   '\)
  628 FORMAT('  NEGATIVE   POSITIVE   '\)
  652 FORMAT(//5X,'ITERATION CYCLE NO.',I3,2X,'AVE OVERALL DAMP =',
     & F6.3)
 1096 FORMAT (///' TIME REQUIRED FOR FORMATION AND TRIANGULIZATION O'
     &         'F MATRICES = ',F11.0,'  SEC')
 1097 FORMAT (///' TIME REQUIRED FOR ',I5,'  STEPS = ',F11.0,
     &         ' SEC')
 7676 FORMAT(//
     &5X,'*****************'/
     &5X,'** END OF JOB **'/
     &5X,'*****************')
 7775 FORMAT(' PROGRAM BLEW UP, KTIME =  ',I5,'  N =  ',I5,'  X =  ',
     & F10.3)
 8989 FORMAT(//
     &5X,'*********************'/
     &5X,'** JOB IS ABORTED **'/
     &5X,'*********************')
  152 FORMAT(/)
 9205 FORMAT(/,'1',//5X,'ITERATION NO. ',I3,//15X,
     &        'SET UP STIFFNESS MATRIX AND COMPUTE EIGENVALUE')
 9206 FORMAT(/,'1',//5X,'ITERATION NO. ',I3,/)
 9305 FORMAT(15X,'TRIANGULARIZE EFFECTIVE STIFFNESS MATRIX')
 9405 FORMAT(15X,'START DYNAMIC COMPUTATION'//)
 9505 FORMAT('+',T14,'** IT =',I2,' STEP NO.='I5,' AT TIME =',F10.4)
      END
```

```
      SUBROUTINE ACCOUT(HISTFMT,ACC,ILINE,IACC)
C
C     Outputs acceleration history to proper files
C
C                          Written for QUAD4M
C                           U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      CHARACTER*1 HISTFMT
C
      IF(HISTFMT.EQ.'C'.OR.HISTFMT.EQ.'c') THEN
        WRITE(IACC+39,'(F10.6\)') ACC
      ELSE
        IF(ILINE.LT.8) THEN
          WRITE(IACC+39,'(F10.6,\)') ACC
        ELSE
          WRITE(IACC+39,'(F10.6)') ACC
        END IF
        IACC=IACC+1
      END IF
      RETURN
      END
C
C ************************************************************************
C
      SUBROUTINE CMPRP (N,S,D,G,DDATA,GDATA,SDATA,NUMPOINTS,NUMPROPS)
C
C     Calculates Strain Compatible Modulus and Damping
C
C                          Written for QUAD4M
C                           U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                        Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      DIMENSION SDATA(2,NUMPROPS,*),GDATA(NUMPROPS,*),DDATA(NUMPROPS,*),
     &          NUMPOINTS(2,*),J(2),FRACTION(2)
C
      S  = ABS(S)
      DO K=1,2
        DO L=1,NUMPOINTS(K,N)
          J(K)=L
          IF(S .LE. SDATA(K,N,L)) EXIT
        END DO
      END DO
      DO K=1,2
        IF(S .GE. SDATA(K,N,1)) THEN
          FRACTION(K) = LOG10(S/SDATA(K,N,J(K)-1))/
     &                  LOG10(SDATA(K,N,J(K))/SDATA(K,N,J(K)-1))
        END IF
      END DO
      IF(S .LT. SDATA(2,N,1)) THEN
        D = DDATA(N,1)
```

```
      ELSE
        D = DDATA(N,J(2)-1) +
     &      FRACTION(2)*(DDATA(N,J(2))-DDATA(N,J(2)-1))
      END IF
      IF(S .LT. SDATA(1,N,1)) THEN
        G = GDATA(N,1)
      ELSE
        G = GDATA(N,J(1)-1) +
     &       FRACTION(1)*(GDATA(N,J(1))-GDATA(N,J(1)-1))
      END IF
      RETURN
      END
C
C ************************************************************************
C
      SUBROUTINE EIGEN(LAMBDA,SMASS,A,R,NF,MAXW)
C
C     Eigenvalue solution by inverse iteration
C     for smallest eigenvalue of inv(A)*SMASS
C
C                          Written for QUAD4M
C                           U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      REAL LAMBDA,LAMBDAOLD,MAXRI
      DIMENSION A(NF,*),SMASS(*),R(*)
C
C     Decompose A
C
      CALL SYMBOL(1,A,R,NF,MAXW)
C
C     Make guess for eigenvector
C
      DO L=1,NF
        R(L)=1.0
      END DO
C
C     Iterate
C
      DO I=1,5000
C
C       Compute R<=M*R
C
        DO L=1,NF
          R(L)=R(L)*SMASS(L)
        END DO
C
C       Compute R=INV(A)*(M*R)
C
        CALL SYMBOL(2,A,R,NF,MAXW)
C
C       Find Maximum component of R
C
        MAXRI=R(1)
```

```
        DO L=2,NF
          MAXRI=MAX(R(L),MAXRI)
        END DO
C
C       Find Lambda and check for quitting
C
        LAMBDA=1.0/MAXRI
        IF (I.GT.1) THEN
          IF(ABS(LAMBDA-LAMBDAOLD)/ABS(LAMBDAOLD) .LE. 1.0E-6) RETURN
        END IF
C
C       Normalize R
C
        DO L=1,NF
          R(L)=R(L)/MAXRI
        END DO
C
C       Save Lambda
C
        LAMBDAOLD=LAMBDA
      END DO
      WRITE (*,'(A)') ' FREQUENCY NOT FOUND TO INTERNALLY SPECIFIED',
     &                ' ACCURACY.  PROGRAM ABORTED'
      WRITE (6,'(A)') ' FREQUENCY NOT FOUND TO INTERNALLY SPECIFIED',
     &                ' ACCURACY.  PROGRAM ABORTED'
      STOP
      END
C
C **********************************************************************
C
      SUBROUTINE FRMSTF(NLP,EL,NO,ZMASS,SMASS,NBC,ST,DS,R,MAXW,ELST,
     &                  ELMASSORG,ROCKBETAS,ROCKBETAP)
C
C     Forms Stiffness Matrix
C
C                         Written for QUAD4M
C                          U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                      Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
```

```
      CHARACTER*72 EQNH,EQNV
      RECORD /ELEMENT/EL(*)
      RECORD /NODE/NO(*)
      INTEGER NLP,W1MULT
      DIMENSION ZMASS(*),SMASS(*),NBC(*),ST(NF,*),DS(NF,*),
     &          ELST(NELM,8,8),ELMASSORG(NELM),R(*),A(3,8),C(3,3),
     &          S(8,8),D(8,8)
C
C     Initialize
C
      DO M=1,8
        DO L = 1, 3
          A(L,M) = 0.      ! Initialize A
        END DO
      END DO
      DO M = 1, 3
        DO L = 1, 3
          C(L,M) = 0.      ! Initialize C
        END DO
      END DO
      DO N = 1, NF
        DO M=1,MAXW
          DS(N,M) = 0.     ! Initialize DS  (Damping Matrix)
          ST(N,M) = 0.     ! Initialize ST  (Global Stiffness Matrix)
        END DO
      END DO
C
C     On first iteration on soil properties, create element stiffness
C     matrices
C
      IF(NLP .EQ. 1) THEN
        DO L = 1, NDPT
          ZMASS(L) = 0.
        END DO
C
        DO N=1,NELM
          DO I=1,8
            DO J=1,8
              ELST(N,I,J)=0.
            END DO
          END DO
          ELMASSORG(N)=0.
          EL(N).E=ABS(EL(N).E)
          RO=EL(N).DENS/GRAV ! Unit weight converted to density
C
C         Get element stiffness matrix:
C
          CALL QUAD(EL(N).NODE(1),EL(N).NODE(2),EL(N).NODE(3),
     &              EL(N).NODE(4),S,C,EL(N).E,EL(N).PO,NO)
C
C         Lumped Mass formulation:
C
          ELMASS=0.25*RO*EL(N).AREA ! Contrb of mass to node by elem
C
```

```
C       Sum Mass around each node
C
        DO I=1,4
          ZMASS(EL(N).NODE(I))=ZMASS(EL(N).NODE(I))+ELMASS
        END DO
C
C       Preserve original element stiffness matrices and masses
C
        DO I=1,8
          DO J=1,8
            ELST(N,I,J)=S(I,J)
          END DO
        END DO
        ELMASSORG(N)=ELMASS
       END DO
C
C      Store same mass in both x and y directions for each node
C
       DO M = 1, NF/2
        SMASS(2*M-1) = ZMASS(M)
        SMASS(2*M)   = ZMASS(M)
       END DO
C
C      Preserve nodal masses
C
      END IF
C
C     Assemble global stiffness, mass, and damping matrices
C
      DO N = 1, NELM
C
C      Restore element stiffness matrices and masses
C
       AJ = ELMASSORG(N)
C
C      Correct element stiffness by the new modulus
C
       AK = EL(N).EN/EL(N).E
       DO L = 1, 8
        DO M = 1, 8
         ELST(N,L,M) = AK*ELST(N,L,M)
        END DO
       END DO
       EL(N).E=EL(N).EN
C
C      Assemble global stiffness matrix
C
       DO L = 1, 4
        IF (EL(N).NODE(L).GE.NB1) CYCLE  ! Skip if on rigid base
        I=2*EL(N).NODE(L)-1       ! Global Row position
        II=2*L-1                  ! Element Row position
        DO M = 1, 4
         IF (EL(N).NODE(M).GE.NB1) CYCLE  ! Skip if on rigid base
         IF (EL(N).NODE(L).LE.EL(N).NODE(M)) THEN
          J=2*(EL(N).NODE(M)-EL(N).NODE(L))+1 ! Global Col posn
          JJ=2*M-1              ! Element Column position
                ST(I,J)    =ST(I,J)    +ELST(N,II,JJ)
                ST(I,J+1)  =ST(I,J+1)  +ELST(N,II,JJ+1)
          IF (J.NE.1) ST(I+1,J-1)=ST(I+1,J-1)+ELST(N,II+1,JJ)
                ST(I+1,J)  =ST(I+1,J)  +ELST(N,II+1,JJ+1)
         END IF
        END DO
       END DO
      END DO
C
C     Preserve stiffness without b.c.'s temporarily in DS
C
      DO L = 1, NF
       DO N = 1, MAXW
        DS(L,N) = ST(L,N)
       END DO
      END DO
C
C     Essential boundary conditions for mass and stiffness
C
      DO N = 1, NF/2
       IF (NO(N).BC.EQ.1.OR.NO(N).BC.EQ.3.OR.NO(N).BC.EQ.4) THEN
        ST(2*N-1,1) = 1.
        SMASS(2*N-1) = 0.
        DO J = 2, MAXW
         ST(2*N-1,J) = 0.
         L = 2*N-J
         IF(L .GT. 0) THEN
          ST(L,J) = 0.
         END IF
        END DO
       END IF
       IF (NO(N).BC.EQ.2.OR.NO(N).BC.EQ.3.OR.NO(N).BC.EQ.4) THEN
        ST(2*N,1) = 1.
        SMASS(2*N) = 0.
        DO J = 2, MAXW
         ST(2*N,J) = 0.
         L = 2*N-J+1
         IF(L .GT. 0) THEN
          ST(L,J) = 0.
         END IF
        END DO
       END IF
      END DO
C
C     Find first frequency of matrix pencil
C
      CALL EIGEN(EV,SMASS,ST,R,NF,MAXW)
C
      W1 = SQRT(EV)           ! 1st mode frequency (circular)
      PR1 = 4.*ASIN(1.)/W1        ! Period
c     Find next higher odd integer to PR1/PRINPUT
      IF (PRINPUT.EQ.0) THEN
```

```
          W1MULT = 1
        ELSE
          W1MULT = INT((PR1/PRINPUT-1)/2+.99999)*2+1
        END IF
        W2 = W1*W1MULT                ! Higher frequency
        PR2 = 4.*ASIN(1.)/W2
        WRITE(*,302) W1,PR1,W1MULT,W2,PR2
        WRITE(6,302) W1,PR1,W1MULT,W2,PR2
C
C       Restore stiffness and mass without b.c.'s; initialize damping
C
        DO M = 1, NF/2
          SMASS(2*M-1) = ZMASS(M)
          SMASS(2*M)   = ZMASS(M)
        END DO
        DO L = 1, NF
          DO N = 1, MAXW
            ST(L,N) = DS(L,N)
            DS(L,N) = 0.
          END DO
        END DO
C
C       Set up damping matrix
C
        DO N = 1, NELM
C
C         element mass
C
          AJ = ELMASSORG(N)
C
C         Form element damping matrix:
C         Rayleigh Damping = alpha*mass + beta*stiffness
C
C            To minimize damping at W1:
          if (w1mult.eq.1) then
            A1 = EL(N).XL*W1
            B1 = EL(N).XL/W1
          else
C            To set damping at W1 & W2:
            A1 = 2.*EL(N).XL*W1*W2/(W1+W2)
            B1 = 2.*EL(N).XL/(W1+W2)
          end if
          BJ = A1*AJ
          DO L = 1, 8
            DO M = 1, 8
              D(L,M) = B1*ELST(N,L,M)
            END DO
          END DO
          DO L =1, 8
            D(L,L) = D(L,L) + BJ
          END DO
C
C         Assemble global damping matrix
C
```

```
          DO L = 1, 4
            IF (EL(N).NODE(L).GE.NB1) CYCLE  ! Skip if on rigid base
            I=2*EL(N).NODE(L)-1  ! global row position
            II=2*L-1           ! element row position
            DO M =1, 4
              IF (EL(N).NODE(M).GE.NB1) CYCLE  ! Skip if on rigid base
              IF(EL(N).NODE(L).LE.EL(N).NODE(M)) THEN
                J=2*(EL(N).NODE(M)-EL(N).NODE(L))+1 ! global col posn
                JJ=2*M-1                        ! element column position
                            DS(I,J)    =DS(I,J)    +D(II,JJ)
                            DS(I,J+1)  =DS(I,J+1)  +D(II,JJ+1)
                IF (J.NE.1) DS(I+1,J-1)=DS(I+1,J-1)+D(II+1,JJ)
                            DS(I+1,J)  =DS(I+1,J)  +D(II+1,JJ+1)
              END IF
            END DO
          END DO
        END DO
C
C       Add contribution of compliant boundaries to damping matrix
C
        DO L=1,NF/2
          IF (NO(L).BC.EQ.4) THEN
            DS(2*L-1,1)=DS(2*L-1,1) + ROCKBETAS*NO(L).TRIBLEN
            IF (KV.EQ.2) DS(2*L,1)=DS(2*L,1) + ROCKBETAP*NO(L).TRIBLEN
          END IF
        END DO
C
C       Essential boundary conditions for damping, mass, and stiffness
C
        IF(NBP .GT. 0) THEN
          DO N = 1, NBP
            I = NBC(N)
            ST(I,1) = 1.
            SMASS(I) = 0.
            DS(I,1) = 0.
            DO J = 2, MAXW
              DS(I,J) = 0.
              ST(I,J) = 0.
              L = I-J+1
              IF(L .GT. 0) THEN
                ST(L,J) = 0.
                DS(L,J) = 0.
              END IF
            END DO
          END DO
        END IF
        RETURN
  302   FORMAT(//,
     &        ' DAMPING SET AT THE FOLLOWING TWO FREQUENCIES:',/,
     &        ' THE FIRST NATURAL FREQUENCY: CIRC FREQ=',F12.3,
     &        '; PERIOD=',F10.3,' SEC',/,I4,
     &            ' TIMES THE NATURAL FREQ.: CIRC FREQ=',F12.3,
     &        '; PERIOD=',F10.3,' SEC',/)
        END
```

```
C
C **********************************************************************
C
      SUBROUTINE GLST(X1,X2,X3,X4,Y1,Y2,Y3,Y4,S,C)
C
C     Numerical Integration of Element Matrix
C
C                        Written for QUAD4M
C                         U.C. Davis, 1993
C        by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                   Modified from QUAD4, 1973,
C        by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      DIMENSION ST(3,8),S(8,8),C(3,3)
      DIMENSION VH(6),VS(6),VT(6),AP(2,4),AB(3,8),AS(8,8)
      DATA ST/24*0./
C
      VH(1)=0.5555555555555555556   ! Weighting factors for quad pnts
      VH(2)=0.8888888888888888889
      VH(3)=VH(1)
      VS(1)=0.774596669241483       ! Xi quadrature coordinates
      VS(2)=0.
      VS(3)=-VS(1)
      VT(1)=VS(1)                   ! Eta quadrature coordinates
      VT(2)=0.
      VT(3)=VS(3)
C
C     Numerically integrate using 9 point quadrature
C
      DO I=1,3
        DO J=1,3
          AP(1,1)=-1.+VT(J)       ! dN1/dXi*4
          AP(2,1)=-1.+VS(I)       ! dN1/dEta*4
          AP(1,2)=1.-VT(J)        ! dN2/dXi*4
          AP(2,2)=-1.-VS(I)       ! dN2/dEta*4
          AP(1,3)=1.+VT(J)        ! dN3/dXi*4
          AP(2,3)=1.+VS(I)        ! dN3/dEta*4
          AP(1,4)=-1.-VT(J)       ! dN4/dXi*4
          AP(2,4)=1.-VS(I)        ! dN4/dEta*4
C
C         dy/dEta*4:
C
          AJ1=AP(2,1)*Y1+AP(2,2)*Y2+AP(2,3)*Y3+AP(2,4)*Y4
C
C         -dy/dXi*4:
C
          AJ2=-AP(1,1)*Y1-AP(1,2)*Y2-AP(1,3)*Y3-AP(1,4)*Y4
C
C         -dx/dEta*4:
C
          AJ3=-AP(2,1)*X1-AP(2,2)*X2-AP(2,3)*X3-AP(2,4)*X4
C
C         dx/dXi*4:
C
          AJ4=AP(1,1)*X1+AP(1,2)*X2+AP(1,3)*X3+AP(1,4)*X4
C
C         determinant of Jacobian * 16
C
          COMM=AJ1*AJ4-AJ2*AJ3
C
C         Prepare matrix [AB]
C
          DO K=1,4
            K1=2*K-1
            K2=2*K
            AB(1,K1)=AJ1*AP(1,K)+AJ2*AP(2,K)    ! det(J)*dN/dx*16
            AB(3,K2)=AB(1,K1)                   ! det(J)*dN/dx*16
            AB(1,K2)=0.
            AB(2,K2)=AJ3*AP(1,K)+AJ4*AP(2,K)    ! det(J)*dN/dy*16
            AB(3,K1)=AB(2,K2)                   ! det(J)*dN/dy*16
            AB(2,K1)=0.
          END DO
C
C         [ST] = [C]*[AB] = det(J)*[C]*[B]*16
C
          DO K=1,8
            DO L=1,3
              ST(L,K)=0.
              DO M=1,3
                ST(L,K)=ST(L,K)+C(L,M)*AB(M,K)
              END DO
            END DO
          END DO
C
C         [AS] =[AB]T*[ST]= det(J)^2*[B]T*[C]*[B]*16^2
C
          DO K=1,8
            DO L=1,8
              AS(K,L)=0.
              DO M=1,3
                AS(K,L)=AS(K,L)+AB(M,K)*ST(M,L)
              END DO
            END DO
          END DO
          DO K=1,8
            DO L=1,8
              S(K,L)=S(K,L)+(VH(I)*VH(J)*AS(K,L))/(16.*COMM)
            END DO
          END DO
        END DO
      END DO
      RETURN
      END
C
C **********************************************************************
C
      FUNCTION MODAY(IMON)
```

```
C
C     FINDS NUMBER OF DAYS PER MONTH
C
      INTEGER*2 IMON
      DIMENSION MONTHS(12)
      DATA MONTHS /31,59,90,120,151,181,212,243,273,304,334,365/
C
      MODAY=MONTHS(IMON)
      RETURN
      END
C
C **********************************************************************
C
      SUBROUTINE QUAD(I,J,K,L,S,C,E,PO,NO)
C
C     Computes Element Stiffness Matrix
C
C                          Written for QUAD4M
C                           U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                       Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      RECORD /NODE/NO(*)
      DIMENSION S(8,8),C(3,3)
C
      F = E/(1.+PO)/(1.-2.*PO)
      C(1,1)=F*(1.-PO)
      C(1,2)=F*PO
      C(1,3)=0.0
      C(2,1)=C(1,2)
      C(2,2)=C(1,1)
      C(2,3)=0.0
      C(3,1)=0.0
      C(3,2)=0.0
      C(3,3)=0.5*F*(1.-2.*PO)
      DO II=1,8
        DO JJ=1,8
          S(II,JJ) = 0.0
        END DO
```

```
      END DO
      CALL GLST(NO(I).XORD,NO(J).XORD,NO(K).XORD,NO(L).XORD,
     &          NO(I).YORD,NO(J).YORD,NO(K).YORD,NO(L).YORD,S,C)
      RETURN
      END
C
C **********************************************************************
C
      SUBROUTINE STRESSES(EL,NO,KTIME,X)
C
C     Computes Stresses in Element based upon strains
C
C                          Written for QUAD4M
C                           U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                       Modified from QUAD4, 1973,
C            by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      RECORD /ELEMENT/EL
      RECORD /NODE/ NO(*)
      DIMENSION X(*)
C
      I=EL.NODE(1)
      J=EL.NODE(2)
      K=EL.NODE(3)
      L=EL.NODE(4)
      AJ1= (-NO(I).YORD-NO(J).YORD+NO(K).YORD+NO(L).YORD)
      AJ2=-(-NO(I).YORD+NO(J).YORD+NO(K).YORD-NO(L).YORD)
      AJ3=-(-NO(I).XORD-NO(J).XORD+NO(K).XORD+NO(L).XORD)
      AJ4= (-NO(I).XORD+NO(J).XORD+NO(K).XORD-NO(L).XORD)
      AJ=AJ1*AJ4-AJ2*AJ3
      II=2*I
      JJ=2*J
      KK=2*K
      LL=2*L
      T1=-X(II-1)+X(JJ-1)+X(KK-1)-X(LL-1)
      T2=-X(II-1)-X(JJ-1)+X(KK-1)+X(LL-1)
      T3=-X(II)+X(JJ)+X(KK)-X(LL)
      T4=-X(II)-X(JJ)+X(KK)+X(LL)
      STX=(AJ1*T1+AJ2*T2)/AJ
```

```
      STY=(AJ3*T3+AJ4*T4)/AJ
      GAM=(AJ3*T1+AJ4*T2+AJ1*T3+AJ2*T4)/AJ
      Z=EL.PO
      Y = EL.E/((1.+Z)*(1.-2.*Z))
      EL.SIG(1) = Y*((1.-Z)*STX+Z*STY)
      EL.SIG(2) = Y*(Z*STX+(1.-Z)*STY)
      EL.SIG(3) = 0.5*Y*(1.-2.*Z)*GAM
      EPS=SQRT(GAM*GAM+(STX-STY)*(STX-STY))
      EL.EPSMAX = MAX(ABS(EPS),EL.EPSMAX)
C
      DO L=1,3
        IF (EL.SIGMAX(L).LT.ABS(EL.SIG(L))) THEN
          EL.SIGMAX(L) = ABS(EL.SIG(L))
          IF (L.EQ.3) EL.TIME2=FLOAT(KTIME)*DTEQ
        END IF
      END DO
      RETURN
      END
C
C ************************************************************************
C

      SUBROUTINE STRESSOUT(SHISTFMT,STRESS,ILINE,ISTRES)
C
C     Writes stress value to stress output file
C
C                           Written for QUAD4M
C                            U.C. Davis, 1993
C             by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      CHARACTER*1 SHISTFMT
C
      IF(SHISTFMT.EQ.'C'.OR.SHISTFMT.EQ.'c') THEN
        WRITE(10,'(F8.1,\)') STRESS
      ELSE
        IF(ILINE.LT.8) THEN
          WRITE(ISTRES+9,'(F8.1,\)') STRESS
        ELSE
          WRITE(ISTRES+9,'(F8.1)') STRESS
        END IF
      END IF
      RETURN
      END
C
C ************************************************************************
C
      SUBROUTINE SYMBOL(KKK,A,B,NN,MM)
C
C     SOLVE SET OF ALGEBRAIC EQUATIONS
C     KKK = 1: decomposes A
C     KKK = 2: reduces and backsubstitutes B
C
C                           Written for QUAD4M
C                            U.C. Davis, 1993
C             by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
C                           Modified from QUAD4, 1973,
C             by I.M.Idriss, J. Lysmer, R. Hwang and H. Bolton Seed
C
      DIMENSION A(NN,*),B(*)
C
      IF (KKK.EQ.1) THEN
C
C       TRIANGULARIZE A : A=L*D*Lt
C
        DO N=1,NN                        ! Row Position
          DO L=2,MM                      ! Column Position
            C=A(N,L)/A(N,1)
            I=N+L-1
            IF(NN-I.GE.0) THEN
              J=0
              DO K=L,MM
                J=J+1
                A(I,J)=A(I,J)-C*A(N,K)
              END DO
            END IF
            A(N,L)=C
          END DO
        END DO
        RETURN
C
C     Reduces B
C
      ELSE
        DO N=1,NN
          DO L=2,MM
            I=N+L-1
            IF(NN-I.LT.0) EXIT
            B(I)=B(I)-A(N,L)*B(N)
          END DO
          B(N)=B(N)/A(N,1)
        END DO
C
C       Backsubstitutes B
C
        N=NN-1
        DO WHILE (N.NE.0)
          DO K=2,MM
            L=N+K-1
            IF (NN.GE.L) B(N)=B(N)-A(N,K)*B(L)
          END DO
          N=N-1
        END DO
        RETURN
      END IF
      END
C
C ************************************************************************
```

```
C
      SUBROUTINE SEISCOEF(EL,NO,KV,NOSEG,NSEG,ELSEG,ESEG,WSLIP,KHISTFMT,
     &                    ILINE,SEISMAX,TIMEF)
C
C     COMPUTE AVERAGED SEISMIC COEFFICIENTS
C
C     WSLIP    = WEIGHT OF BLOCK DEFINED BY SEISMIC COEFF SURFACE
C     NSEG     = NUMBER OF NODES THROUGH WHICH SEISMIC COEFF SURFACE PASSES
C     NOSEG    = NODE NUMBER THROUGH WHICH SEISMIC COEFF SURFACE PASSES
C
C                          Written for QUAD4M
C                           U.C. Davis, 1993
C            by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      RECORD /ELEMENT/EL(*)
      RECORD /NODE/NO(*)
      CHARACTER KHISTFMT*1
      INTEGER ESEG,ELSEG
      DIMENSION WSLIP(*),NSEG(*),NOSEG(NSLP,*),SEISMAX(2*KV,*),
     &          ESEG(*),ELSEG(NSLP,*)
C
      N=11
C
C         Compute Weight of Slip Surfaces:
C
      DO I=1,NSLP
        WSLIP(I)=0.
        DO J=1,ESEG(I)
          WSLIP(I)=WSLIP(I)+EL(ELSEG(I,J)).AREA*EL(ELSEG(I,J)).DENS
        END DO
      END DO
      IF (KHISTFMT.EQ.'C'.OR.KHISTFMT.EQ.'c') THEN
        WRITE(N+39,'(F10.3,\)') TIMEF
      END IF
      DO J=1,NSLP
        FORCEH=0.0
        FORCEV=0.0
        NSS=NSEG(J)
        DO I=1,NSS-1
          NELCOUNT=0
          NEL1=0
          NEL2=0
          DO K=1,NELM
            DO L=1,4
              IF (EL(K).NODE(L).EQ.NOSEG(J,I)) THEN
                DO M=1,4
                  IF (EL(K).NODE(M).EQ.NOSEG(J,I+1)) THEN
                    IF (NELCOUNT.EQ.0) THEN
                      NEL1=K
                      NELCOUNT=1
                    ELSE IF (NELCOUNT.EQ.1) THEN
                      NEL2=K
                      NELCOUNT=2
                    ELSE
                      WRITE (*,720) I,I+1,NEL1,NEL2,K
 720  FORMAT (' Too many elements next to nodes',I4,' & ',I4,
     &        ': Elements '/,3I4)
                      STOP
                    END IF
                  EXIT
                  END IF
                END DO
                EXIT
              END IF
            END DO
          END DO
          HEIGHT=ABS(NO(NOSEG(J,I)).YORD-NO(NOSEG(J,I+1)).YORD)
          WIDTH=ABS(NO(NOSEG(J,I)).XORD-NO(NOSEG(J,I+1)).XORD)
          XM=(NO(NOSEG(J,I)).XORD+NO(NOSEG(J,I+1)).XORD)/2.
          YM=(NO(NOSEG(J,I)).YORD+NO(NOSEG(J,I+1)).YORD)/2.
          IF(EL(NEL1).NODE(3).EQ.EL(NEL1).NODE(4)) THEN
            XC1=(NO(EL(NEL1).NODE(1)).XORD+NO(EL(NEL1).NODE(2)).XORD+
     &           NO(EL(NEL1).NODE(3)).XORD)/3.
            YC1=(NO(EL(NEL1).NODE(1)).YORD+NO(EL(NEL1).NODE(2)).YORD+
     &           NO(EL(NEL1).NODE(3)).YORD)/3.
          ELSE
            XC1=(NO(EL(NEL1).NODE(1)).XORD+NO(EL(NEL1).NODE(2)).XORD+
     &           NO(EL(NEL1).NODE(3)).XORD+NO(EL(NEL1).NODE(4)).XORD)
     &           /4.
            YC1=(NO(EL(NEL1).NODE(1)).YORD+NO(EL(NEL1).NODE(2)).YORD+
     &           NO(EL(NEL1).NODE(3)).YORD+NO(EL(NEL1).NODE(4)).YORD)
     &           /4.
          END IF
          IF(EL(NEL2).NODE(3).EQ.EL(NEL2).NODE(4)) THEN
            XC2=(NO(EL(NEL2).NODE(1)).XORD+NO(EL(NEL2).NODE(2)).XORD+
     &           NO(EL(NEL2).NODE(3)).XORD)/3.
            YC2=(NO(EL(NEL2).NODE(1)).YORD+NO(EL(NEL2).NODE(2)).YORD+
     &           NO(EL(NEL2).NODE(3)).YORD)/3.
          ELSE
            XC2=(NO(EL(NEL2).NODE(1)).XORD+NO(EL(NEL2).NODE(2)).XORD+
     &           NO(EL(NEL2).NODE(3)).XORD+NO(EL(NEL2).NODE(4)).XORD)
     &           /4.
            YC2=(NO(EL(NEL2).NODE(1)).YORD+NO(EL(NEL2).NODE(2)).YORD+
     &           NO(EL(NEL2).NODE(3)).YORD+NO(EL(NEL2).NODE(4)).YORD)
     &           /4.
          END IF
          FACTNUM=SQRT((XM-XC1)*(XM-XC1)+(YM-YC1)*(YM-YC1))
          FACTDENOM=SQRT((XC2-XM)*(XC2-XM)+(YC2-YM)*(YC2-YM))+FACTNUM
```

```
          FACTOR=FACTNUM/FACTDENOM
          SIGX=(EL(NEL1).SIG(1)-EL(NEL2).SIG(1))*FACTOR+EL(NEL2).SIG(1)
          SIGY=(EL(NEL1).SIG(2)-EL(NEL2).SIG(2))*FACTOR+EL(NEL2).SIG(2)
          TAUXY=(EL(NEL1).SIG(3)-EL(NEL2).SIG(3))*FACTOR+EL(NEL2).SIG(3)
          FORCEH=FORCEH+SIGX*HEIGHT+TAUXY*WIDTH
          IF(KV .EQ. 2) THEN
            FORCEV=FORCEV+SIGY*WIDTH+TAUXY*HEIGHT
          END IF
        END DO
        ACCAVE=FORCEH/WSLIP(J)
        IF (ACCAVE.LT.SEISMAX(1,J)) SEISMAX(1,J)=ACCAVE
        IF (ACCAVE.GT.SEISMAX(2,J)) SEISMAX(2,J)=ACCAVE
        CALL ACCOUT(KHISTFMT,ACCAVE,ILINE,N)
        IF(KV .EQ. 2) THEN
          AVVAVE=FORCEV/WSLIP(J)
          IF (AVVAVE.LT.SEISMAX(3,J)) SEISMAX(3,J)=AVVAVE
          IF (AVVAVE.GT.SEISMAX(4,J)) SEISMAX(4,J)=AVVAVE
          CALL ACCOUT(KHISTFMT,AVVAVE,ILINE,N)
        ENDIF
      END DO
      IF (KHISTFMT.EQ.'C'.OR.KHISTFMT.EQ.'c') THEN
        WRITE(N+39,'(/\)')
      END IF
      RETURN
      END
C
C ***********************************************************************
C
      SUBROUTINE WRITEFILE(NSEG,NOSEG,ESEG,ELSEG,EL,NO)
C
C     Produces file for restart
C
C                         Written for QUAD4M
C                          U.C. Davis, 1993
C           by Martin Byrd Hudson, I.M. Idriss, Mohsen Beikae
C
      STRUCTURE /ELEMENT/
        REAL GMX,G,E,EN,AREA,XL,TIME2,SIG(3),SIGMAX(3),EPSMAX,PO,DENS
        INTEGER NODE(4),TYPE,LSTR
      END STRUCTURE
      STRUCTURE /NODE/
        REAL XORD,YORD,TRIBLEN,BETAS,BETAP,X2I(2),X1I(2),XI(2)
        INTEGER BC,OUT
      END STRUCTURE
      COMMON/CONST1/TITLE,DRF,NDPT,NB1,NBP,NELM,UNITS,GRAV,NSLP,NF
      CHARACTER TITLE*72,UNITS*1
      COMMON/CONST2/KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV,KSAV
      COMMON/CONST3/DTEQ,EQMUL(2),PRM,UGMAX(2),EQNH,EQNV,UG(2),PRINPUT
      CHARACTER*72 EQNH,EQNV
      COMMON/CONST4/SHISTFMT,AHISTFMT,KHISTFMT,SFILEOUT,AFILEOUT,
     &              KFILEOUT,SSUFFIX,ASUFFIX,KSUFFIX,DIROUT,SAVEFILE,
     &               DATAIN
      CHARACTER SHISTFMT*1,AHISTFMT*1,KHISTFMT*1,SFILEOUT*8,AFILEOUT*8,
     &         KFILEOUT*8,SSUFFIX*3,ASUFFIX*3,KSUFFIX*3,DIROUT*72,
     &          SAVEFILE*12,DATAIN*72
      COMMON/CONST5/HDRX,HDRY,EARTHQH,EARTHQV,EQINPFMT(2),ROCKVP,ROCKVS,
     &              ROCKRHO,IHISTSTR,IHISTACC,NPLX,NPLY
      CHARACTER EQINPFMT*20,EARTHQH*72,EARTHQV*72
      INTEGER SOUT,AOUT,KOUT,HDRX,HDRY,ESEG,ELSEG
      RECORD /ELEMENT/EL(*)
      RECORD /NODE/NO(*)
      DIMENSION NSEG(*),NOSEG(NSLP,*),ESEG(*),ELSEG(NSLP,*)
C
      WRITE (4,'(A)') TITLE
      WRITE (4,'(A,31X,A,12X,A)') 'UNITS (E for English, S for SI):',
     &                            '*** (A1)','***'
      WRITE (4,'(A1)') UNITS
      WRITE (4,'(7X,A,7X,A,13X,A,8X,A)') 'DRF',
     &         'PRM    ROCKVP    ROCKVS   ROCKRHO','*** (5F10.0)','***'
      WRITE (4,'(5G10.5E1)') DRF,PRM,ROCKVP,ROCKVS,ROCKRHO
      WRITE (4,'(A,48X,A,11X,A)') ' NELM NDPT NSLP','*** (3I5)','***'
      WRITE (4,'(3I5)') NELM,NDPT,NSLP
      WRITE (4,'(A,23X,A,11X,A)')
     &      'KGMAX KGEQ N1EQ N2EQ N3EQ NUMB   KV KSAV','*** (8I5)','***'
      WRITE (4,'(7I5)') KGMAX,KGEQ,N1EQ,N2EQ,N3EQ,NUMB,KV
      WRITE (4,'(A,A,A)') '      DTEQ    EQMUL1    EQMUL2    UGMAX1',
     &                    '    UGMAX2 HDRX HDRY NPLX NPLY   PRINPUT',
     &                    ' *** (5F10.0,4I5,F10.0) ***'
      WRITE (4,'(5G10.5E1,4I5,G10.5E1)') DTEQ,EQMUL(1),EQMUL(2),
     &                     UGMAX(1),UGMAX(2),HDRX,HDRY,NPLX,NPLY,PRINPUT
      WRITE (4,'(A,A,A,13X,A)') 'EARTHQUAKE INPUT FILE NAME(S) &',
     &                          ' FORMAT(S) (* for FREE FORMAT)',
     &                          '  *** (A)','***'
      WRITE (4,'(A)') EARTHQH
      WRITE (4,'(A)') EQINPFMT(1)
      IF (KV.EQ.2) THEN
        WRITE (4,'(A)') EARTHQV
        WRITE (4,'(A)') EQINPFMT(2)
      END IF
      WRITE (4,'(A,48X,A,11X,A)') ' SOUT AOUT KOUT','*** (3I5)','***'
      SOUT=0
      AOUT=0
      KOUT=0
      IF (IHISTSTR.GT.0) SOUT=1
      IF (IHISTACC.GT.0) AOUT=1
      IF (NSLP.GT.0) KOUT=1
      WRITE (4,'(3I5)') SOUT,AOUT,KOUT
      IF (IHISTSTR.GT.0) THEN
        WRITE (4,'(A,A)') 'STRESS OUTPUT FORMAT (M or C), FILE PREFIX,',
     &                    ' AND SUFFIX:       *** (A)            ***'
        WRITE (4,'(A1)') SHISTFMT
        WRITE (4,'(A)') SFILEOUT
        WRITE (4,'(A)') SSUFFIX
      END IF
      IF (IHISTACC.GT.0) THEN
        WRITE (4,'(A,A,13X,A)') 'ACCELERATION OUTPUT FORMAT (M or C),',
     &           ' FILE PREFIX, AND SUFFIX:  *** (A)','***'
        WRITE (4,'(A1)') AHISTFMT
```

```
      WRITE (4,'(A)') AFILEOUT
      WRITE (4,'(A)') ASUFFIX
    END IF
    IF (NSLP.GT.0) THEN
      WRITE (4,'(A,A,13X,A)') 'SEISMIC COEFF OUTPUT FORMAT (M or C),',
   &          ' FILE PREFIX, AND SUFFIX: *** (A)','***'
      WRITE (4,'(A1)') KHISTFMT
      WRITE (4,'(A)') KFILEOUT
      WRITE (4,'(A)') KSUFFIX
      DO I=1,NSLP
        WRITE (4,'(A,53X,A)') ' NSEG ESEG','*** (2I5)           ***'
        WRITE (4,'(2I5)') NSEG(I),ESEG(I)
        WRITE (4,'(A,58X,A)') 'NOSEG','*** (15I5)          ***'
        WRITE (4,'(15I5)') (NOSEG(I,J),J=1,NSEG(I))
        WRITE (4,'(A,58X,A)') 'ELSEG','*** (15I5)          ***'
        WRITE (4,'(15I5)') (ELSEG(I,J),J=1,ESEG(I))
      END DO
    END IF
    WRITE (4,'(A,A,A)') '    N  NP1  NP2  NP3  NP4 TYPE',
   & '      DENS        PO       GMX         G        XL LSTR',
   & '  *** (6I5,5F10.0,I5) ***'
    DO N=1,NELM
      WRITE (4,'(6I5,5G10.5E1,I5)') N,(EL(N).NODE(I),I=1,4),
   &          EL(N).TYPE,EL(N).DENS,EL(N).PO,EL(N).GMX,
   &          EL(N).E/(2000.*(1.+EL(N).PO)),EL(N).XL,EL(N).LSTR
    END DO
    WRITE (4,1)
    DO N=1,NDPT
      WRITE (4,'(I5,2G10.4E1,2I5,6E13.6)') N,NO(N).XORD,
   &          NO(N).YORD,NO(N).BC,NO(N).OUT,NO(N).X2I(1),
   &          NO(N).X1I(1),NO(N).XI(1),NO(N).X2I(2),NO(N).X1I(2),
   &          NO(N).XI(2)
    END DO
    RETURN
  1 FORMAT ('    N      XORD      YORD   BC',
   & '  OUT',9X,'X2IH',9X,'X1IH',10X,'XIH',9X,'X2IV',9X,'X1IV',10X,
   & 'XIV   *** (I5,2F10.0,2I5,6E13.0) ***')
    END
```

Appendix B - Sample Input

FILE: EXAMPLE.Q4I

```
Sliding Block Example Problem
UNITS (E for English, S for SI):                              *** (A1)            ***
E
       DRF       PRM    ROCKVP    ROCKVS   ROCKRHO            *** (5F10.0)        ***
         1      0.65
 NELM NDPT NSLP                                               *** (3I5)           ***
  330  388    4
KGMAX KGEQ N1EQ N2EQ N3EQ NUMB   KV KSAV                      *** (8I5)           ***
 2000 2000    1    1 2000    3    2    1
      DTEQ    EQMUL1    EQMUL2    UGMAX1    UGMAX2 HDRX HDRY NPLX NPLY   PRINPUT *** (5F10.0,4I5,F10.0) ***
      0.02         1         1                         3    3    8    8     0.153
EARTHQUAKE INPUT FILE NAME(S) & FORMAT(S) (* for FREE FORMAT) *** (A)             ***
SC_0.ACC
*
SC_V.ACC
*
 SOUT AOUT KOUT                                               *** (3I5)           ***
    1    1    1
STRESS OUTPUT FORMAT (M or C), FILE PREFIX, AND SUFFIX:       *** (A)             ***
COMBINED
EXAMPLE
Q4S
ACCELERATION OUTPUT FORMAT (M or C), FILE PREFIX, AND SUFFIX: *** (A)             ***
COMBINED
EXAMPLE
Q4A
SEISMIC COEFF OUTPUT FORMAT (M or C), FILE PREFIX, AND SUFFIX: *** (A)            ***
COMBINED
EXAMPLE
Q4K
SYSTEM STATE OUTPUT FILE:                                     *** (A)
EXAMPLE.Q4R
 NSEG ESEG                                                    *** (2I5)          ***
    5    3
NOSEG                                                         *** (15I5)         ***
  122  130  139  138  137
ELSEG                                                         *** (15I5)         ***
  100  108  107
 NSEG ESEG                                                    *** (2I5)          ***
    8    9
NOSEG                                                         *** (15I5)         ***
  122  130  139  148  157  166  165  164
ELSEG                                                         *** (15I5)         ***
  100  108  107  116  115  124  123  132  131
 NSEG ESEG                                                    *** (2I5)          ***
   11   18
NOSEG                                                         *** (15I5)         ***
  111  117  124  132  141  150  159  158  157  156  155
ELSEG                                                         *** (15I5)         ***
   89   94   95  100  101  102  107  108  109  110  115  116  117  118  123
  124  125  126
 NSEG ESEG                                                    *** (2I5)          ***
   14   30
NOSEG                                                         *** (15I5)         ***
  111  117  124  132  141  150  159  168  177  186  185  184  183  182
ELSEG                                                         *** (15I5)         ***
   89   94   95  100  101  102  107  108  109  110  115  116  117  118  123
  124  125  126  131  132  133  134  139  140  141  142  147  148  149  150
    N  NP1  NP2  NP3  NP4 TYPE      DENS        PO       GMX         G        XL LSTR  *** (6I5,5F10.0,I5) ***
    1    1    2    7    6    1       120      0.45       345      249.    .08198
    2    2    3    8    7    1       120      0.45       345      208.    .11187
    3    3    4    9    8    1       120      0.45       345      207.    .11235
    4    4    5   10    9    1       120      0.45       345      186.    .12847
    5    6    7   12   11    1       120      0.45       345      220.    .10279
    6    7    8   13   12    1       120      0.45       345      157.    .15053
```

FULL INPUT NOT SHOWN

```
  314  360  361  370  369    1       120      0.45       345      171.    .13981
  315  362  363  372  371    1       120      0.45       345      271.    .06606
  316  363  364  373  372    1       120      0.45       345      207.    .11287
  317  364  365  374  373    1       120      0.45       345      197.    .12035
  318  365  366  375  374    1       120      0.45       345      196.    .12108
  319  366  367  376  375    1       120      0.45       345      176.    .13583
  320  367  368  377  376    1       120      0.45       345      177.    .13538
```

```
  321  368  369  378  377    1        120       0.45        345       171.      .13995
  322  369  370  379  378    1        120       0.45        345       169.      .14123
  323  371  372  381  380    1        120       0.45        345       266.      .06966
  324  372  373  382  381    1        120       0.45        345       243.      .08622
  325  373  374  383  382    1        120       0.45        345       241.      .08720
  326  374  375  384  383    1        120       0.45        345       243.      .08638
  327  375  376  385  384    1        120       0.45        345       228.      .09700
  328  376  377  386  385    1        120       0.45        345       225.      .09885
  329  377  378  387  386    1        120       0.45        345       214.      .10768
  330  378  379  388  387    1        120       0.45        345       210.      .11063
    N       XORD       YORD   BC  OUT       X2IH      X1IH       XIH      X2IV      X1IV       XIV  *** (I5,2F10.0,2I5,6F10.0) ***
    1      -2100         50    3
    2      -2100       37.5    3
    3      -2100         25    3
    4      -2100       12.5    3
    5      -2100          0    3
    6      -1860         50
    7      -1860       37.5
    8      -1860         25
```

FULL INPUT NOT SHOWN

```
  367       3400       37.5
  368       3400         25
  369       3400       12.5
  370       3400          0    3
  371       3770        100
  372       3770       87.5
  373       3770         75
  374       3770       62.5
  375       3770         50
  376       3770       37.5
  377       3770         25
  378       3770       12.5
  379       3770          0    3
  380       4250        100    3
  381       4250       87.5    3
  382       4250         75    3
  383       4250       62.5    3
  384       4250         50    3
  385       4250       37.5    3
  386       4250         25    3
  387       4250       12.5    3
  388       4250          0    3
```

FILE: NEWSOIL.DAT

```
 4
10  #1 MODULUS FOR CLAY (Idriss, 1990; Seed & Sun 1989) upper range
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   1.0       1.0       1.0     0.981     0.941     0.847     0.656     0.438
  0.238     0.144
10     DAMPING FOR CLAY (Idriss, 1990)
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   0.24      0.42       0.8       1.4       2.8       5.1       9.8      15.5
    21.       25.
10  #2 MODULUS FOR SAND (Idriss, 1990; Seed & Idriss 1970) - upper Range
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   1.0       1.0      0.99      0.96      0.85      0.64      0.37      0.18
   0.08     0.050
10     DAMPING FOR SAND (Idriss, 1990) - (about LRng from SI 1970)
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   0.24      0.42       0.8       1.4       2.8       5.1       9.8      15.5
    21.       25.
10  #3 MODULUS FOR CLAY (Idriss, 1990; Seed & Sun 1989) upper range
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   1.0       1.0       1.0     0.981     0.941     0.847     0.656     0.438
  0.238     0.144
10    damping for sand & refuse (Seed & Idriss 1970) - avg-
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   1.00      1.50       2.5       3.8       6.1       8.6      12.4      16.9
    21.       26.
10  #4 MODULUS FOR SAND (Idriss, 1990; Seed & Idriss 1970) - upper Range
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   1.0       1.0      0.99      0.96      0.85      0.64      0.37      0.18
   0.08      .050
10    damping for sand & refuse (Seed & Idriss 1970) - avg-
  .0001     .0003      .001      .003       .01       .03       0.1       0.3
    1.        3.
   1.00      1.50       2.5       3.8       6.1       8.6      12.4      16.9
    21.       26.
```

FILE: SC_0.ACC (partial listing)

```
"Loma P. Eqk","Santa Cruz","H2_0","init. vel:","1.369 c/s","disp: -1.066 cm"
"Total No. of Points :",2000,"@ DT =",.02
"Peak Acceleration (g) =",.4413005,"@ Time (sec) :",7.54
 -0.002839  0.003273  0.004088  0.002397 -0.004423 -0.003957  0.002609  0.003254
 -0.002257 -0.000217  0.005400  0.000527 -0.005296 -0.005056 -0.000565  0.001540
  0.003347  0.010519  0.011245  0.000712 -0.002901 -0.006159 -0.011535 -0.015913
 -0.008552 -0.008890 -0.000954  0.011002  0.021674  0.022078  0.016705  0.003633
 -0.019671 -0.025850 -0.029108 -0.027187 -0.011781  0.020643  0.027084  0.026296
```

FILE: SC_V.ACC (partial listing)

```
"Loma P. Eqk","Santa Cruz","Vert","init. vel:",".051 c/s","disp: -0.143 cm"
"Total No. of Points :",2000,"@ DT =",.02
"Peak Acceleration (g) =",.3307056,"@ Time (sec) :",7.42
 -0.000521 -0.001011  0.000503  0.002590 -0.008624 -0.002452  0.032474  0.034005
 -0.002525 -0.028430 -0.015743  0.007411  0.024565 -0.012625 -0.033764 -0.004639
  0.007550  0.011532  0.006985  0.012103  0.012117  0.011107 -0.000131  0.002244
  0.004372 -0.005036 -0.013182 -0.010066 -0.000011 -0.003768 -0.005271 -0.000947
  0.005143 -0.003727 -0.009742 -0.009777  0.015026  0.014614 -0.001033 -0.019850
```

Appendix C - Sample Output

FILE: EXAMPLE.Q4O

```
          *******************************************************
          *******************************************************
          ** QUAD4M   A COMPUTER PROGRAM FOR EVALUATING THE **
          **          SEISMIC RESPONSE OF SOIL STRUCTURES   **
          **                 U.C.Davis. 1993                **
          **              by Martin Byrd Hudson.            **
          **                 I.M.Idriss.                    **
          **                 and Mohsen Beikae              **
          **    MODIFIED FROM QUAD4. 1973                   **
          **              by I.M.Idriss.                    **
          **                 J. Lysmer.                     **
          **                 R. Hwang and                   **
          **                 H. Bolton Seed                 **
          *******************************************************
          *******************************************************

   Sliding Block Example Problem
 HORIZONTAL ACCELERATION INPUT FILE:
   SC_0.ACC
 WITH FIRST LINE:
   "Loma P. Eqk"."Santa Cruz"."H2_0"."init. vel:"."1.369 c/s"."disp: -1.066
 VERTICAL ACCELERATION INPUT FILE:
   SC_V.ACC
 WITH FIRST LINE:
   "Loma P. Eqk"."Santa Cruz"."Vert"."init. vel:".".051 c/s"."disp: -0.143



1

                    NO. OF ELEMENTS =   330
                 NO. OF NODAL POINTS =   388
                  DEGREES OF FREEDOM =   756
                      HALF-BANDWIDTH =    22
                 CONTROLLING ELEMENT =   108
          NO. OF FIXED BNDRY CONDS. =   112
                   NO. OF ITERATIONS =     3
       TOTAL EQ. POINTS READ (KGMAX) =  2000
 LAST EQ. PTS. USED (N1EQ TO KGEQ)  =     1  2000
   INT. EQ. PTS USED (N2EQ TO N3EQ) =     1  2000
           TIME INTERVAL OF RECORDS =   .0200  SECONDS
           STRAIN CONVERSION FACTOR =   .6500
      DAMPING RATIO REDUCTION FACTOR =  1.000
     PREDOMINANT INPUT MOTION PERIOD =   .1530  SECONDS
       EQ. MULT. FACTOR (HORZ. COMP.) = 1.0000
   MAXIMUM ACCEL. USED (HORZ. COMP.) =   .4413
       EQ. MULT. FACTOR (VERT. COMP.) = 1.0000
   MAXIMUM ACCEL. USED (VERT. COMP.) =   .3307

     4 STRESS HISTORIES REQUESTED.
     3 ACCEL HISTORIES REQUESTED.
     4 SEIS COEFF HISTORIES REQUESTED
         OUTPUT FILES ARE AS FOLLOWS:

 ELEMENT 110. SIGMA X IN FILE:  EXAMPLE.Q4S
 ELEMENT 110. SIGMA Y IN FILE:  EXAMPLE.Q4S
 ELEMENT 110.  TAU XY IN FILE:  EXAMPLE.Q4S
 ELEMENT 148.  TAU XY IN FILE:  EXAMPLE.Q4S
   NODE  137.   X DIR IN FILE:  EXAMPLE.Q4A
   NODE  137.   Y DIR IN FILE:  EXAMPLE.Q4A
```

```
  NODE  191.   X DIR IN FILE:  EXAMPLE.Q4A
SURFACE   1.   X DIR IN FILE:  EXAMPLE.Q4K
SURFACE   1.   Y DIR IN FILE:  EXAMPLE.Q4K
SURFACE   2.   X DIR IN FILE:  EXAMPLE.Q4K
SURFACE   2.   Y DIR IN FILE:  EXAMPLE.Q4K
SURFACE   3.   X DIR IN FILE:  EXAMPLE.Q4K
SURFACE   3.   Y DIR IN FILE:  EXAMPLE.Q4K
SURFACE   4.   X DIR IN FILE:  EXAMPLE.Q4K
SURFACE   4.   Y DIR IN FILE:  EXAMPLE.Q4K

FINAL CONDITIONS OF NODES SAVED FOR RESTART IN FILE:  EXAMPLE.Q4R

 SOIL DATA TAKEN FROM FILE: newsoil.dat

**********************
 MATERIAL TYPE NO. 1
**********************
MODULUS:   #1 MODULUS FOR CLAY (Idriss. 1990: Seed & Sun 1989) upper range
DAMPING:      DAMPING FOR CLAY (Idriss. 1990)
 ==================    =================
   STRAIN    G/Gmax       STRAIN    DAMPING
  --------   -------     --------  --------
    .0001     1.000         .0001       .24
    .0003     1.000         .0003       .42
    .0010     1.000         .0010       .80
    .0030      .981         .0030      1.40
    .0100      .941         .0100      2.80
    .0300      .847         .0300      5.10
  [illegible]  .656         .1000      9.80
    .3000      .438         .3000     15.50
   1.0000      .238        1.0000     21.00
   3.0000      .144        3.0000     25.00

**********************
 MATERIAL TYPE NO. 2
**********************
MODULUS:   #2 MODULUS FOR SAND (Idriss. 1990; Seed & Idriss 1970) - upper Range
DAMPING:      DAMPING FOR SAND (Idriss. 1990) - (about LRng from SI 1970)
 ==================    =================
   STRAIN    G/Gmax       STRAIN    DAMPING
  --------   -------     --------  --------
    .0001     1.000         .0001       .24
    .0003     1.000         .0003       .42
    .0010      .990         .0010       .80
    .0030      .960         .0030      1.40
    .0100      .850         .0100      2.80
    .0300      .640         .0300      5.10
    .1000      .370         .1000      9.80
    .3000      .180         .3000     15.50
   1.0000      .080        1.0000     21.00
   3.0000      .050        3.0000     25.00

**********************
 MATERIAL TYPE NO. 3
**********************
MODULUS:   #3 MODULUS FOR CLAY (Idriss. 1990: Seed & Sun 1989) upper range
DAMPING:     damping for sand & refuse (Seed & Idriss 1970) - avg-
 ==================    =================
   STRAIN    G/Gmax       STRAIN    DAMPING
  --------   -------     --------  --------
    .0001     1.000         .0001      1.00
    .0003     1.000         .0003      1.50
    .0010     1.000         .0010      2.50
    .0030      .981         .0030      3.80
    .0100      .941         .0100      6.10
```

```
   .0300    .847       .0300     8.60
   .1000    .656       .1000    12.40
   .3000    .438       .3000    16.90
  1.0000    .238      1.0000    21.00
  3.0000    .144      3.0000    26.00

**********************
 MATERIAL TYPE NO. 4
**********************
MODULUS:   #4 MODULUS FOR SAND (Idriss. 1990: Seed & Idriss 1970) - upper Range
DAMPING:     damping for sand & refuse (Seed & Idriss 1970) - avg-
 =================     ================
   STRAIN   G/Gmax      STRAIN   DAMPING
  --------  -------    --------  --------
    .0001    1.000       .0001     1.00
    .0003    1.000       .0003     1.50
    .0010     .990       .0010     2.50
    .0030     .960       .0030     3.80
    .0100     .850       .0100     6.10
    .0300     .640       .0300     8.60
    .1000     .370       .1000    12.40
    .3000     .180       .3000    16.90
   1.0000     .080      1.0000    21.00
   3.0000     .050      3.0000    26.00

UNDERLYING STRATUM DATA:

    UNIT WEIGHT:        .000  PCF
    SHEAR WAVE VELOCITY:        .00  FT/SEC
    COMPRESSION WAVE VELOCITY:        .00  FT/SEC



1
 SEISMIC COEFFICIENT SURFACE DATA:

 BLOCK:  1     NSEG:   5     ESEG:   3
    SURFACE PASSES THROUGH NODES:
  122  130  139  138  137
    SURFACE INCLUDES ELEMENTS:
  100  108  107

 BLOCK:  2     NSEG:   8     ESEG:   9
    SURFACE PASSES THROUGH NODES:
  122  130  139  148  157  166  165  164
    SURFACE.INCLUDES ELEMENTS:
  100  108  107  116  115  124  123  132  131

 BLOCK:  3     NSEG:  11     ESEG:  18
    SURFACE PASSES THROUGH NODES:
  111  117  124  132  141  150  159  158  157  156  155
    SURFACE INCLUDES ELEMENTS:
   89   94   95  100  101  102  107  108  109  110  115  116  117  118  123
  124  125  126

 BLOCK:  4     NSEG:  14     ESEG:  30
    SURFACE PASSES THROUGH NODES:
  111  117  124  132  141  150  159  168  177  186  185  184  183  182
    SURFACE INCLUDES ELEMENTS:
   89   94   95  100  101  102  107  108  109  110  115  116  117  118  123
  124  125  126  131  132  133  134  139  140  141  142  147  148  149  150



1
  ELM     NODE 1  NODE 2  NODE 3  NODE 4  MAT.TYPE  DENSITY    POISSON R.    GMX     SH. MODULUS  DAMP. RATIO     AREA
                                                     (PCF)                  (KSF)      (KSF)                     (FT^2)

     1       1       2       7       6        1     120.000        .450    345.000    249.000        .082    3000.000
     2       2       3       8       7        1     120.000        .450    345.000    208.000        .112    3000.000
     3       3       4       9       8        1     120.000        .450    345.000    207.000        .112    3000.000
     4       4       5      10       9        1     120.000        .450    345.000    186.000        .128    3000.000
     5       6       7      12      11        1     120.000        .450    345.000    220.000        .103    2312.500
     6       7       8      13      12        1     120.000        .450    345.000    157.000        .151    2312.500
     7       8       9      14      13        1     120.000        .450    345.000    150.000        .156    2312.500
     8       9      10      15      14        1     120.000        .450    345.000    121.000        .179    2312.500
     9      11      12      17      16        1     120.000        .450    345.000    232.000        .094    2100.000
    10      12      13      18      17        1     120.000        .450    345.000    147.000        .159    2100.000
    11      13      14      19      18        1     120.000        .450    345.000    129.000        .173    2100.000
    12      14      15      20      19        1     120.000        .450    345.000    108.000        .189    2100.000
    13      16      17      22      21        1     120.000        .450    345.000    232.000        .094    1912.500
```

```
    14      17      18      23      22        1     120.000        .450    345.000    144.000        .161    1912.500
```

FULL OUTPUT NOT SHOWN

```
   316     363     364     373     372        1     120.000        .450    345.000    207.000        .113    4625.000
   317     364     365     374     373        1     120.000        .450    345.000    197.000        .120    4625.000
   318     365     366     375     374        1     120.000        .450    345.000    196.000        .121    4625.000
   319     366     367     376     375        1     120.000        .450    345.000    176.000        .136    4625.000
   320     367     368     377     376        1     120.000        .450    345.000    177.000        .135    4625.000
   321     368     369     378     377        1     120.000        .450    345.000    171.000        .140    4625.000
   322     369     370     379     378        1     120.000        .450    345.000    169.000        .141    4625.000
   323     371     372     381     380        1     120.000        .450    345.000    266.000        .070    6000.000
   324     372     373     382     381        1     120.000        .450    345.000    243.000        .086    6000.000
   325     373     374     383     382        1     120.000        .450    345.000    241.000        .087    6000.000
   326     374     375     384     383        1     120.000        .450    345.000    243.000        .086    6000.000
   327     375     376     385     384        1     120.000        .450    345.000    228.000        .097    6000.000
   328     376     377     386     385        1     120.000        .450    345.000    225.000        .099    6000.000
   329     377     378     387     386        1     120.000        .450    345.000    214.000        .108    6000.000
   330     378     379     388     387        1     120.000        .450    345.000    210.000        .111    6000.000


1
   NODE         XORD          YORD      TRIBUTARY LEN

     1     -2100.000        50.000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
     2     -2100.000        37.500          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
     3     -2100.000        25.000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
     4     -2100.000        12.500          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
     5     -2100.000          .000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
     6     -1860.000        50.000          .000
     7     -1860.000        37.500          .000
     8     -1860.000        25.000          .000
     9     -1860.000        12.500          .000
    10     -1860.000          .000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
    11     -1675.000        50.000          .000
    12     -1675.000        37.500          .000
```

FULL OUTPUT NOT SHOWN

```
   375      3770.000        50.000          .000
   376      3770.000        37.500          .000
   377      3770.000        25.000          .000
   378      3770.000        12.500          .000
   379      3770.000          .000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   380      4250.000       100.000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   381      4250.000        87.500          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   382      4250.000        75.000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   383      4250.000        62.500          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   384      4250.000        50.000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   385      4250.000        37.500          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   386      4250.000        25.000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   387      4250.000        12.500          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
   388      4250.000          .000          .000   CANNOT MOVE IN BOTH X AND Y DIRECTIONS
1

    ITERATION NO.   1



 DAMPING SET AT THE FOLLOWING TWO FREQUENCIES:
 THE FIRST NATURAL FREQUENCY: CIRC FREQ=      3.431: PERIOD=    1.831 SEC
  13 TIMES THE NATURAL FREQ.: CIRC FREQ=     44.601: PERIOD=     .141 SEC



 TIME REQUIRED FOR FORMATION AND TRIANGULIZATION OF MATRICES =        10.  SEC


    MODULI (ENG: KSF or SI: KN/M^2) AND DAMPING
  ELM      G-USED       G-NEW     DIF-G  DAMP-USED    DAMP-NEW  DIF-DAMP


    1       249.0       247.5       .6    .08198      .08289     -1.1
    2       208.0       206.5       .7    .11187      .11302     -1.0
    3       207.0       203.9      1.5    .11235      .11502     -2.3
```

```
   4      186.0      175.4      6.0   .12847     .13659    -5.9
   5      220.0      218.3       .8   .10279     .10409    -1.2
```

FULL OUTPUT NOT PRINTED FOR FIRST ITERATIONS

```
    ITERATION CYCLE NO.  2  AVE OVERALL DAMP =  .124


 TIME REQUIRED FOR  2000  STEPS =        184. SEC

1

    ITERATION NO.   3


 DAMPING SET AT THE FOLLOWING TWO FREQUENCIES:
 THE FIRST NATURAL FREQUENCY: CIRC FREQ=      3.428: PERIOD=     1.833 SEC
  13 TIMES THE NATURAL FREQ.: CIRC FREQ=     44.563: PERIOD=      .141 SEC



 TIME REQUIRED FOR FORMATION AND TRIANGULIZATION OF MATRICES =          8.  SEC


    MODULI (ENG: KSF or SI: KN/M^2) AND DAMPING
  ELM       G-USED       G-NEW     DIF-G  DAMP-USED    DAMP-NEW  DIF-DAMP


   1      245.3      244.5       .3   .08446     .08504     -.7
   2      210.0      210.0       .0   .11039     .11039      .0
   3      204.5      204.5       .0   .11455     .11450      .0
   4      172.9      172.1       .4   .13852     .13908     -.4
   5      218.3      218.2       .0   .10409     .10416     -.1
```

FULL OUTPUT NOT SHOWN

```
 314      171.0      170.9       .1   .13991     .13999     -.1
 315      269.5      269.7      -.1   .06718     .06707      .2
 316      208.7      209.0      -.2   .11139     .11115      .2
 317      192.8      192.5       .2   .12337     .12360     -.2
 318      193.0      193.1       .0   .12325     .12320      .0
 319      174.1      174.0       .0   .13760     .13763      .0
 320      175.5      175.4       .0   .13653     .13656      .0
 321      168.2      168.2       .0   .14204     .14207      .0
 322      169.8      169.8       .0   .14083     .14082      .0
 323      272.7      272.8      -.1   .06494     .06482      .2
 324      245.8      246.1      -.1   .08408     .08389      .2
 325      242.1      242.1       .0   .08672     .08677      .0
 326      240.5      240.7      -.1   .08785     .08775      .1
 327      225.4      225.5       .0   .09866     .09861      .1
 328      222.5      222.5       .0   .10089     .10088      .0
 329      210.0      209.9       .0   .11036     .11044     -.1
 330      213.4      213.8      -.2   .10779     .10746      .3
1
    PEAK NODAL ACCELERATION VALUES (g's)

       NODE     XORD     YORD         X-ACC       AT TIME         Y-ACC       AT TIME

          1  -2100.0     50.0         .4413        7.5600         .3307        7.4400
          2  -2100.0     37.5         .4413        7.5600         .3307        7.4400
          3  -2100.0     25.0         .4413        7.5600         .3307        7.4400
          4  -2100.0     12.5         .4413        7.5600         .3307        7.4400
          5  -2100.0       .0         .4413        7.5600         .3307        7.4400
          6  -1860.0     50.0         .3745        8.1600         .3579        7.8200
          7  -1860.0     37.5         .2700        8.9200         .2926        7.8200
          8  -1860.0     25.0         .3380        8.0000         .2274        7.7800
          9  -1860.0     12.5         .4416        7.8000         .2152        7.7600
         10  -1860.0       .0         .4413        7.5600         .3307        7.4400
         11  -1675.0     50.0         .3938       10.7800         .4155        7.8200
         12  -1675.0     37.5         .3359       10.7800         .3519        7.8200
         13  -1675.0     25.0         .3027       10.1000         .2584        7.7800
         14  -1675.0     12.5         .4075        7.8000         .2173        7.7600
         15  -1675.0       .0         .4413        7.5600         .3307        7.4400
         16  -1507.0     50.0         .3884       10.8000         .3766        7.8200
         17  -1507.0     37.5         .3418       10.8000         .3313        7.8200
```

FULL OUTPUT NOT SHOWN

```
        370   3400.0       .0         .4413        7.5600         .3307        7.4400
        371   3770.0    100.0         .2185       10.4600         .1930        7.9600
        372   3770.0     87.5         .1665       10.4600         .1573       10.9800
        373   3770.0     75.0         .1426        8.2400         .1084        8.6400
        374   3770.0     62.5         .2122        8.2200         .1320        9.1400
        375   3770.0     50.0         .1509        7.6600         .1563        6.3800
        376   3770.0     37.5         .1917        8.0600         .1998        7.8600
        377   3770.0     25.0         .2763        7.8600         .2412        7.8400
        378   3770.0     12.5         .3100        7.8000         .2994        7.8200



1
    PEAK ELEMENTS STRESSES (ENG: PSF or SI: N/M^2) AND STRAINS

        ELM          SIG-X         SIG-Y         SIG-XY         EPS-MAX        AT TIME


          1         1149.1         784.4          184.3            .110          8.160
          2          792.9         596.3          399.4            .195          8.180
          3          653.8         525.5          427.3            .211          8.960
          4          416.4         539.6          582.2            .340         10.660
          5          976.5         566.5          267.6            .173          8.160
          6          795.1         760.6          678.0            .458         10.780
          7          852.4        1087.3          798.2            .571         10.760
          8          858.3        1059.4          974.6            .831         10.680
```

FULL OUTPUT NOT SHOWN

```
        315          551.6         478.6          187.3            .070          8.380
        316          660.7         651.5          413.5            .198         10.460
        317          743.1         759.7          482.7            .252          9.180
        318          742.0         765.8          482.0            .250          9.880
        319          630.9         665.0          572.5            .330         10.640
        320          647.7         755.8          567.4            .323         10.020
        321          663.4         830.0          602.3            .360         10.520
        322          693.9         856.9          596.3            .351          7.820
        323          942.6         625.9           93.7            .066          8.400
        324          814.8         603.2          227.2            .107         10.480
        325          763.7         597.2          265.4            .115          9.200
        326          727.8         631.8          268.1            .118          9.880
        327          776.8         720.7          339.5            .156         10.640
        328          767.2         756.7          357.9            .163         10.580
        329          680.1         725.3          410.6            .196         10.520
        330          544.1         629.9          393.4            .185         10.460



1  MAX & MIN SEISMIC COEFFICIENTS
    SURFACE   WEIGHT(LB or N)   X-DIRECTION              Y-DIRECTION
                                NEGATIVE   POSITIVE      NEGATIVE   POSITIVE

        1       75000.0000      -.3273       .3694       -.2858       .3708
        2      525000.0000      -.1460       .1624       -.1531       .1592
        3      900000.0000      -.1120       .1242       -.0960       .1359
        4     1800000.0000      -.0951       .0949       -.0971       .1019

     ITERATION CYCLE NO.  3  AVE OVERALL DAMP =  .124


  TIME REQUIRED FOR  2000  STEPS =        252. SEC


     ****************
     ** END OF JOB **
     ****************
```