RICE UNIVERSITY

RESPONSE OF SLIDING ELASTIC SYSTEMS
TO GROUND EXCITATION

by

Michael Christopher Tucker

A THESIS SUBMITTED
IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF

Master of Science

Thesis director's signature:

Houston, Texas
April, 1978
ABSTRACT

Developed in this investigation are a method of analysis and a computer program for a sliding two-degree-of-freedom elastic system excited by ground motion. The system consists of two masses connected by a spring in parallel with a viscous damper. The motion of one of the masses, which slides on its support (or the ground), is damped in addition by a Coulomb (constant-force) damper representing sliding friction. Such a system has significant applications in the analysis of systems such as cranes, pipelines, bridges, and fluid-filled tanks. A study is made of the response of the system to a half-cycle velocity pulse, and the results of the study are summarized in the form of response spectra. Complete documentation of the FORTRAN computer program is included.
ACKNOWLEDGEMENTS

The author wishes to record his sincere gratitude to his research advisor, Dr. E. C. Holt, for his encouragement, his many helpful suggestions, and his generosity with his time, during the course of this investigation. The author considers himself privileged to have studied under the direction of a teacher of his caliber.

Appreciation is expressed to the other members of the author's research committee, Dr. W. J. Austin and Dr. A. S. Veletsos, for their interest in the investigation.

The author also wishes to recognize his parents for their encouragement and sacrifices so that he could complete his studies.
# TABLE OF CONTENTS

**LIST OF FIGURES**  
**LIST OF TABLES**  
**NOTATION**  
1. **OBJECTIVE AND SCOPE**  
2. **METHOD OF ANALYSIS**  
   2.1 System Considered  
   2.2 Ground Motions  
   2.3 Differential Equations of Motion  
      2.3.1 Derivation  
      2.3.2 Computation of Response Coefficients  
   2.4 Solution of Equations  
      2.4.1 Solution Sequence  
      2.4.2 Criterion for Sliding to Begin  
      2.4.3 Criterion for Sliding to Cease  
      2.4.4 Algorithm for Equation Solvers  
      2.4.5 Algorithm for Tangent Point Equation Solvers  
   2.5 System Parameters  
3. **RESULTS OF PROGRAM DEVELOPMENT**  
4. **STUDY OF RESPONSE DUE TO HALF-CYCLE VELOCITY PULSE**  
5. **CONCLUSIONS**  
**REFERENCES**  
**APPENDICES**  
   A. Computer Program Description  
      A.1 Running the Program  
      A.1.1 Accelerogram Description
A.1.2  Time-History Printout Option  34
A.1.3  Preparation of Input  34
A.2  Program Documentation  36
   A.2.1  Organization  36
   A.2.2  Quantities Available for Output  38
   A.2.3  Detailed Subroutine Description  38
B.  Computer Program Listing  91
LIST OF FIGURES

2.1 System Considered  
2.2 Half-Cycle Velocity Accelerogram  
2.3 General Accelerogram  
2.4 General Form of Function f  
2.5 Solution of f=0 (case 1)  
2.6 Solution of f=0 (case 2)  
2.7 Solution of f=0 (case 3)  
2.8 Solution of f=0 (case 4)  
4.1 Relative Response q for Half-Cycle Velocity Pulse (\( \rho = .95 \))  
4.2 Relative Response q for Half-Cycle Velocity Pulse (\( \rho = .7071 \))  
4.3 Relative Response q for Half-Cycle Velocity Pulse (\( \rho = .4 \))  
4.4 Sliding Response x for Half-Cycle Velocity Pulse (\( \rho = .95 \))  
4.5 Sliding Response x for Half-Cycle Velocity Pulse (\( \rho = .7071 \))  
4.6 Sliding Response x for Half-Cycle Velocity Pulse (\( \rho = .4 \))

LIST OF TABLES

2.1 Coefficients for Relative Response q  
2.2 Coefficients for Sliding Response x  
2.3 Coefficients for Absolute Responses u, v
NOTATION

All symbols are defined when first introduced in the text. A summary is provided here for convenience.

\( A_j \) coefficient of \((\sin \omega t)\) term in response quantity
\( a \) initial ground acceleration for current time step
\( B_j \) coefficient of \((\cos \omega t)\) term in response quantity
\( b \) slope of current accelerogram segment
\( C_j \) constant term in response quantity
\( c \) damping constant
\( D_j \) coefficient of linear term in response quantity
\( E_j \) coefficient of quadratic term in response quantity
\( F \) maximum value of friction force
\( FF \) friction factor
\( f_2 \) non-sliding (free-free) undamped natural frequency
\( G_j \) coefficient of cubic term in response quantity
\( k \) spring stiffness
\( M \) total mass of system \( m_1 + m_2 \)
\( m \) \( m_1 m_2 / (m_1 + m_2) \)
\( m_1 \) mass damped by ground sliding
\( m_2 \) mass not damped by ground sliding
\( p_1 \) undamped circular natural frequency of system (sliding case)
\( p_2 \) undamped circular natural frequency of system (non-sliding case)
\( q \) displacement of mass \( m_2 \) relative to mass \( m_1 \)
\( q_m \) maximum absolute value of \( q \)
\( q_0 \) value of \( q \) at \( t=0 \)
\( \dot{q}_0 \) value of \( \dot{q} \) at \( t=0 \)
s friction force (non-sliding case)
t local time for current time step
t_{ref} reference time for accelerogram
u absolute displacement of mass m₁
v absolute displacement of mass m₂
x displacement of mass m₁ relative to ground
x₀ value of x at t=0
ẋ₀ value of ẋ at t=0
x₂ displacement of mass m₂ relative to ground
xₘ maximum absolute value of x
y ground displacement
yₘ maximum ground displacement
ẏₘ maximum ground velocity
ÿₘ maximum ground acceleration
β₁ damping factor (sliding case)
β₂ damping factor (non-sliding case)
ρ frequency ratio ρ₂/ρ₁
ω₁ damped circular natural frequency of system (sliding case)
ω₂ damped circular natural frequency of system (non-sliding case)
CHAPTER 1

OBJECTIVE AND SCOPE

The objective of this investigation is to develop a method of analysis and a computer program for an elastic system which slides on its support and is excited by ground motion. The system consists of two masses connected by a spring in parallel with a viscous damper. The motion of one of the masses, which slides on the ground, is damped in addition by a Coulomb (constant force) damper representing sliding friction. The other mass is damped only by the viscous damper.

Comparatively little investigation has been done on the response of Coulomb-damped systems, as opposed to viscously damped systems. Jacobsen\(^1\) presented an approximate formula for the steady-state response of a Coulomb-damped single-degree-of-freedom system subjected to a sinusoidal excitation, and Den Hartog\(^2\) gave an exact solution to the same problem for various types of damping. Mindlin\(^3\) examined the response of a single-degree-of-freedom system subjected to a sudden shock. Bielak\(^4\) investigated the effects of Coulomb damping in bilinear single-degree-of-freedom systems. Crede and Ruzicka\(^5\) examined Coulomb damping in single-degree-of-freedom shock isolation systems.

This problem has significant applications in several areas, including cranes sliding on runways, pipelines on supports, bridges on abutments, and fluid-filled tanks on slabs. The response of any of these systems to ground excitation is (after a certain threshold is reached) a combination of vibration and sliding. Both responses are of interest -- the vibration from the standpoint of stresses induced in the structure, and the sliding from the standpoint of movement of the structure.
CHAPTER 2

METHOD OF ANALYSIS

2.1 System Considered

The system under consideration (Fig. 2.1) consists of two rigid masses $m_1$ and $m_2$, a weightless elastic spring with stiffness $k$, and a viscous damper with damping coefficient $c$. When mass $m_1$ does not slide on the ground, it is held in place by a friction force $s$ whose maximum absolute value is $F$. When sliding, the motion of mass $m_1$ is damped by a constant friction force $F$. The displacement of mass $m_1$ with respect to ground is denoted by $x$, that of mass $m_2$ with respect to ground by $x_2$, and the relative displacement of mass $m_2$ with respect to mass $m_1$ by $q = x_2 - x$. Let the motion of the ground be denoted by $y$, the absolute
displacement of mass \( m_1 \) by \( u = x + y \), and the absolute displacement of
mass \( m_2 \) by \( v = x + y + q \). Dots over any of the quantities \( x, q, y, u, \) or \( v \) represent differentiation with respect to time.

2.2 Ground Motions

To facilitate solution of the equations, all ground accelerations
are assumed to vary piecewise linearly. Earthquake accelerations may
be closely approximated in this manner. Discontinuities in the ground
acceleration are acceptable to the program, but any accelerogram sup¬
plied must integrate to zero, so that the final ground velocity is
equal to zero.

For purposes of this investigation, a half-cycle velocity pulse
was used (Fig. 2.2). A maximum acceleration of 1 was chosen, since the
ratio of masses to acceleration may be adjusted by varying other para¬
meters.

\[ \text{FIGURE 2.2} \]
2.3 Differential Equations of Motion

2.3.1 Derivation. The differential equations governing the response of the system take one of two forms, depending on whether mass \( m_1 \) is sliding or stationary with respect to ground at a given instant of time.

Case 1: Sliding. If \( m_1 \) slides with respect to ground, the system has two degrees of freedom, and the equations of motion are:

\[
\begin{align*}
 m_1 (\ddot{x} + \ddot{y}) &= k (x_2 - x) + c (\dot{x}_2 - \dot{x}) - F \operatorname{sgn} \dot{x} \quad (2.1) \\
 m_2 (\ddot{x}_2 + \ddot{y}) &= k (x - x_2) + c (\dot{x} - \dot{x}_2) \quad (2.2)
\end{align*}
\]

The sign function \( \operatorname{sgn} u \) is defined to be \( u/|u| \), and is undefined for \( u = 0 \). The sum of equations (2.1) and (2.2) is

\[
m_1 \ddot{x} + m_2 \ddot{x}_2 + (m_1 + m_2) \ddot{y} = - F \operatorname{sgn} \dot{x} \quad (2.3)
\]

Since \( x_2 = q + x \), (2.3) may be written

\[
(m_1 + m_2) \ddot{x} + m_2 \ddot{q} + (m_1 + m_2) \ddot{y} = - F \operatorname{sgn} \dot{x} \quad (2.4)
\]

Subtraction of (2.1) from (2.2) yields

\[
\ddot{x}_2 - \ddot{x} = k \left( \frac{1}{m_1} + \frac{1}{m_2} \right) (x - x_2) + c \left( \frac{1}{m_1} + \frac{1}{m_2} \right) (\dot{x} - \dot{x}_2) + F \operatorname{sgn} \dot{x} \quad (2.5)
\]

Define \( m = m_1 m_2 / (m_1 + m_2) \), \( p_1 = \sqrt{k/m_1} \), \( \beta_1 = \frac{c}{2mp_1} \); then (2.5) can be
rewritten

\[ \ddot{q} + 2\beta_1 p_1 \dot{q} + p_1^2 q = \frac{F}{m_1} \text{sig} \dot{x} \quad (2.6) \]

(2.4) can be rewritten

\[ \ddot{x} = -\frac{m}{m_1} \ddot{q} - \dddot{y} - \frac{m}{m_1 m_2} F \text{sig} \dot{x} \quad (2.7) \]

In a time interval during which \( \dot{x} \) does not change sign, \( q \) may be evaluated from (2.6) independently of \( x \), if \( \text{sig} \dot{x} \) is known. \( x \) may then be evaluated from \( q \) using (2.7).

**Case 2: Non-sliding.** If \( m_1 \) does not slide with respect to ground, the system has one degree of freedom, and the equations of motion are:

\[ m_1 \ddot{y} = kj + cj + s \quad (2.8) \]

\[ m_2 (\dot{x}_2 + \ddot{y}) = -kj - cj \quad (2.9) \]

( \( \dot{x}=\dot{x} = 0, x = \text{constant} \) )

The sum of equations (2.8) and (2.9) is (note \( x_2 = q \))

\[ (m_1 + m_2) \ddot{y} + m_2 \dddot{q} = s \quad (2.10) \]

This equation holds only for \( |s| \leq F \). If (2.10) gives \( |s| > F \), mass \( m_2 \) slides and the sliding equations of motion apply. Equation (2.9) can be rewritten
\[ \ddot{q} + \frac{c}{m_2} \dot{q} + \frac{k}{m_2} q = -\ddot{y} \]  

(2.11)

Define \( p_2 = \sqrt{k/m_2} \), \( \beta_2 = \frac{c}{2m_2 p_2} \), then (2.11) can be expressed as

\[ \ddot{q} + 2\beta_2 p_2 \dot{q} + p_2^2 q = -\ddot{y} \]  

(2.12)

### 2.3.2 Computation of Response Coefficients

The ground acceleration varies piecewise linearly, so for any time segment it may be expressed as

\[ \ddot{y} = a + bt \]  

(2.13)

where \( t \) is the local time since the beginning of the segment, \( a \) is the acceleration at \( t=0 \), and \( b \) is the jerk during the particular segment.

So whether mass \( m_1 \) is sliding with respect to the ground or not sliding, the evaluation of the response quantity \( q \) hinges on the solution of the general equation

\[ \ddot{q} + 2\beta p \dot{q} + p^2 q = -g - ht \]  

(2.14)

If \( h=0 \) and \( g=-\frac{F}{m_1} \cdot \text{sgn} \cdot \dot{x} \), then (2.14) becomes (2.6) for the sliding case.

If \( g=a \) and \( h=b \), then (2.14) becomes (2.12) for the non-sliding case.

The solution of equation (2.14) is of the form

\[ q = e^{-\beta pt} \left( A_1 \sin \omega t + B_1 \cos \omega t \right) + C_1 + D_1 t \]  

(2.15)

where \( \omega = \sqrt{1-\beta^2} \), and \( A_1, B_1, C_1, \) and \( D_1 \) have the values listed in
The derivatives of \( q \) may likewise be expressed

\[
\ddot{q} = e^{-\beta pt}(A_2 \sin \omega t + B_2 \cos \omega t) + C_2 \tag{2.16}
\]
\[
\ddot{q} = e^{-\beta pt}(A_3 \sin \omega t + B_3 \cos \omega t) \tag{2.17}
\]
\[
\ddot{q} = e^{-\beta pt}(A_4 \sin \omega t + B_4 \cos \omega t) \tag{2.18}
\]

where \( A_j, B_j, C_j, \) and \( D_j \) are listed in table (2.1).

For the non-sliding case, \( \ddot{x} = \dot{x} = 0 ; x = \text{constant}. \)

For the sliding case, \( x \) and its derivatives can be evaluated from (2.7) in terms of \( q \) and its derivatives.

\[
x = e^{-\beta_1 p_1 t}(A_5 \sin \omega_1 t + B_5 \cos \omega_1 t) + C_5 + D_5 t + E_5 t^2 + G_5 t^3 \tag{2.19}
\]
\[
\dot{x} = e^{-\beta_1 p_1 t}(A_6 \sin \omega_1 t + B_6 \cos \omega_1 t) + C_6 + D_6 t + E_6 t^2 \tag{2.20}
\]
\[
\ddot{x} = e^{-\beta_1 p_1 t}(A_7 \sin \omega_1 t + B_7 \cos \omega_1 t) + C_7 + D_7 t \tag{2.21}
\]
\[
\dddot{x} = e^{-\beta_1 p_1 t}(A_8 \sin \omega_1 t + B_8 \cos \omega_1 t) + C_8 \tag{2.22}
\]

where \( A_j, B_j, C_j, D_j, E_j, \) and \( G_j \) are listed in table (2.2)

The absolute accelerations \( \ddot{u} \) and \( \ddot{v} \) and their derivatives may then be expressed in terms of \( x, y, \) and \( q \).

\[
\ddot{u} = e^{-\beta pt}(A_9 \sin \omega t + B_9 \cos \omega t) + C_9 + D_9 t \tag{2.23}
\]
\[
\dddot{u} = e^{-\beta pt}(A_{10} \sin \omega t + B_{10} \cos \omega t) + C_{10} \tag{2.24}
\]
\[
\dddot{v} = e^{-\beta pt}(A_{11} \sin \omega t + B_{11} \cos \omega t) + C_{11} + D_{11} t \tag{2.25}
\]
\[
\dddot{v} = e^{-\beta pt}(A_{12} \sin \omega t + B_{12} \cos \omega t) + C_{12} \tag{2.26}
\]

where \( A_j, B_j, C_j, \) and \( D_j \) are listed in table (2.3).
For computational purposes any of the quantities \( q, x, \ddot{u}, \dot{v}, \) and their derivatives may be expressed in the general form

\[
e^{-\beta_i p_i t} \left( A_j \sin \omega_i t + B_j \cos \omega_i t \right) + C_j + D_j t + E_j t^2 + G_j t^3 \quad (2.27)
\]

where \( i \) defines the state of the system, and \( j \) defines the response quantity. All coefficients \( A_j, B_j, C_j, D_j, E_j, \) and \( G_j \) are summarized in tables (2.1), (2.2), and (2.3). Values not given are zero.

2.4 Solution of Equations

2.4.1 Solution Sequence. The equations are solved in a stepwise manner. The time coordinate \( t \) in the equations is a local time such that at the beginning of each time step, \( t = 0 \). The following quantities are known from the previous time step:

\[
\begin{align*}
\dot{x}_0 & \quad q_0 \\
\dot{x}_0 & \quad \dot{q}_0
\end{align*}
\]

The state of the system (sliding or non-sliding) is known, and the current accelerogram segment is defined in terms of three quantities \( a, b, \) and \( T_F \) (Fig. 2.3), which may be read from the accelerogram or recalculated from their previous values. \( a \) and \( b \) were defined in (2.3.2); \( T_F \) is the remaining length of the current accelerogram segment. The coefficients \( A, B, C, D, E, \) and \( G \) necessary for evaluation of the response quantities may then be calculated.

If the system does not change state (sliding to non-sliding or vice versa) on the time interval \((0, T_F)\), then maxima of responses are com-
puted on that interval.

If the system changes state, say at time TB, then maxima of response quantities are computed on the time interval (0, TB), and new values for TF and a are calculated.

New values of \( x_0, \dot{x}_0, q_0, \dot{q}_0 \) are calculated, and the next time step begins at \( t = TF \) or \( TB \), whichever is applicable, and \( t \) is reset to zero.

\[
\begin{align*}
\ddot{y} \\
\text{global time} \\
T=0 \\
t=0 & \quad t=TF
\end{align*}
\]

**FIGURE 2.3**

2.4.2 Criterion for Sliding to Begin. The amount of friction holding mass \( m_1 \) stationary with respect to ground is, from equation (2.10):

\[
s = (m_1 + m_2)\ddot{y} + m_2\ddot{q} \quad |s| \leq F 
\]  

(2.28)

Mass \( m_1 \) begins sliding when \( |s| \) exceeds \( F \). If \( |s| = F \), mass \( m_1 \) re-
mains stationary. Substituting equation (2.17) into (2.28) yields

\[ s = (m_1+m_2)(a+bt) + m_2e^{-\beta_2 P_2 t} (A_3 \sin \omega_2 t + B_3 \cos \omega_2 t) \]  

(2.29)

Rearranging, and setting \( s = \pm F \)

\[ e^{-\beta_2 P_2 t} (A_3 m_2 \sin \omega_2 t + B_3 m_2 \cos \omega_2 t) + (m_1+m_2)a + (m_1+m_2)bt = \pm F \]  

(2.30)

As \( s \) exceeds \( +F \), sliding begins such that \( \text{sig}(x) = -1 \).
As \( s \) exceeds \( -F \), sliding begins such that \( \text{sig}(x) = +1 \).

2.4.3 Criterion for Sliding to Cease. The velocity of mass \( m_1 \) with respect to ground is given by (2.20):

\[ \dot{x} = e^{-\beta_1 P_1 t} (A_6 \sin \omega_1 t + B_6 \cos \omega_1 t) + C_6 + D_6 t + E_6 t^2 \]

As \( \dot{x} \) becomes zero, mass \( m_1 \) becomes stationary with respect to ground.
Therefore, sliding of mass \( m_1 \) ceases when

\[ e^{-\beta_1 P_1 t} (A_6 \sin \omega_1 t + B_6 \cos \omega_1 t) + C_6 + D_6 t + E_6 t^2 = 0 \]  

(2.31)

2.4.4 Algorithm for Equation Solvers. The equations to be solved are all of the form:

\[ f = e^{-\beta pt} (A \sin \omega t + B \cos \omega t) + C +Dt + Et^2 = 0 \]  

(2.32)

It is desired to find the smallest root \( t \) such that \( t > 0 \).
Let $TA$ be the time of the first extremum, if any exist, of the function $f$, obtained by setting the derivative $f'$ equal to zero. If $f$ is monotone, an equation solver which searches the points where $f$ is tangent to its decay envelope must be used (see 2.4.5).

If a sign change exists on $(0,TA)$, Newton's Method may be used to locate the root.

If no sign change exists on $(0,TA)$, let $TB$ be the time of the next extremum of $f$. If $f$ is monotone after the first extremum $TA$, see 2.4.5.

If a sign change exists on $(TA,TB)$, Newton's Method may be used to locate the root.

If no sign change exists on $(TA,TB)$, see 2.4.5.

2.4.5 Algorithm for Tangent Point Equation Solvers. In the event that the function given by equation (2.32) becomes monotone before a root is found, or if the root lies beyond the first two extrema computed, an efficient method is required to search for the root. Since the points where the function is tangent to its decay envelope on a given side are $2\pi/\omega$ apart (Figure 2.4), the function may be evaluated at the first such tangent point, and at intervals of $2\pi/\omega$ thereafter, until a sign change is found. Newton's Method may then be used to locate the root.

Tangent points of $f$ occur when the derivative of its sinusoidally varying component is equal to zero, or

$$A \cos \omega t - B \sin \omega t = 0$$

(2.33)

To eliminate those tangent points which are not on the $x$-axis side
(Fig. 2.4), the condition is imposed that the sign of the derivative of (2.33) be the same as \( \text{sig}(F_0) \), where \( F_0 \) is the value of \( f \) at \( t=0 \), or

\[
\text{sig} (-A \sin \omega t - B \cos \omega t) = \text{sig}(F_0)
\]

If \( B=0 \), a tangent point occurs at \( \omega t_a = \pi/2 \), and criterion (2.34) may be expressed

\[
-\text{sig}(F_0) = \text{sig} A
\]

So if \( F_0 A > 0 \), \( \pi/\omega \) must be added to \( t_a \).

If \( B \neq 0 \), since \( \sin \omega t = \frac{A}{B} \cos \omega t \), equation (2.34) may be rewritten

\[
-\text{sig}(F_0) = \text{sig} \left( \frac{A^2}{B} \cos \omega t + B \cos \omega t \right)
\]
A tangent point (designated $t_a$) may be obtained from (2.33):

$$t_a = \tan^{-1}(A/B)$$

(2.36)

Since $-\pi/2 < t_a < \pi/2$, $\cos \omega t_a > 0$, so that criterion (2.35) may be expressed

$$-\text{sig}(F_0) = \text{sig} B$$

(2.37)

So if $F_0 x B > 0$, $\pi/\omega$ must be added to $t_a$, or if $t_a < 0$, $2\pi/\omega$ must be added to $t_a$.

A partial evaluation $PP$ of the function $f$ is made at time $t_a$:

$$PP = e^{-\beta^{pt_a}} \left( A \sin \omega t_a + B \cos \omega t_a \right)$$

(2.38)

so that

$$f(t_a) = PP + C + Dt_a + Et_a^2$$

(2.39)

Then, to evaluate $f$ at the next tangent point, it is necessary only to multiply $PP$ by $e^{-2\pi\beta p/\omega}$, and to increment $t_a$ in equation (2.39) by $2\pi/\omega$. Thus $f$ may be evaluated at each tangent point on the x-axis side using equation (2.39) until a sign change is found, without repeated evaluations of sines and cosines.

When a sign change in $f$ has been located, let $t_a$ be the last tangent point occurring before the sign change. The first extremum $t_c$ greater than $t_a$ (if such an extremum exists) can be obtained by solving $f' = 0$, using the appropriate equation solver.
If the sign change occurs on \((t_a, t_c)\), then the root may be located using Newton's method with initial trial \(t_a\) (Figure 2.5).

\[
\begin{array}{c}
\text{FIGURE 2.5}
\end{array}
\]

If no sign change occurs on \((t_a, t_c)\) or if \(t_c\) does not exist (i.e., \(f\) is monotonic) then \(f\) is evaluated at tangent point \(t_b = t_a + 2\pi/\omega\).

If the sign change occurs on \((t_a, t_b)\), the root lies on the interval \((t_c, t_b)\) if \(t_c\) exists. In this case Newton's Method is applied with initial trial \(t_b\) (Figure 2.6).

\[
\begin{array}{c}
\text{FIGURE 2.6}
\end{array}
\]
If $t_c$ does not exist, Newton's Method is applied with initial trial $t_a$ or $t_b$, whichever has the smallest absolute value, to insure convergence (Figure 2.7).

If no sign change occurs on $(t_a, t_b)$, the root lies between $t_b$ and the next $t_a$; Newton's Method is applied with initial trial $t_b + \pi/2\omega$ (Figure 2.8).
2.5 System Parameters

Any system to be analyzed may be described by four nondimensional parameters:

1. $f_2 t_{\text{ref}}$, a parameter nondimensionalizing the non-sliding natural frequency with some reference time such as the duration of an accelerogram

2. $\rho$, the ratio of non-sliding to sliding frequencies $p_2/p_1$; may be calculated from the system masses directly: $\rho = \sqrt{m_1/M} < 1$

3. $\beta_2$, the non-sliding damping factor

4. $FF$, a friction factor, the ratio of the friction force $F$ to the product of the total system mass $M$ and the maximum ground acceleration $\ddot{y}_m$: $FF = \frac{F}{My_m}$

The program uses a system normalized such that the total mass $M$ is equal to 1. Inspection of the equations of motion of the system shows that multiplying $M$, $k$, $F$, and $c$ by the same constant has no effect on $q$ and $x$. 
### Table 2.1
Coefficients for Relative Response $q$

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>$\frac{\beta_i p_i}{\omega_i} q_o + \frac{\dot{q}_o}{\omega_i} - \frac{\beta_i g}{p_i \omega_i} + \frac{(1-2\beta_i^2) h}{p_i \omega_i}$</td>
</tr>
<tr>
<td>$B_1$</td>
<td>$q_o + \frac{g}{p_i^2} - \frac{2\beta_i h}{p_i^3}$</td>
</tr>
<tr>
<td>$C_1$</td>
<td>$\frac{g}{p_i^2} + \frac{2\beta_i h}{p_i^3}$</td>
</tr>
<tr>
<td>$D_1$</td>
<td>$-\frac{h}{p_i^2}$</td>
</tr>
<tr>
<td>$A_2$</td>
<td>$-\frac{\beta_i p_i}{\omega_i} \ddot{q}_o - \frac{p_i^2 q_o}{\omega_i} - \frac{g}{\omega_i} + \frac{\beta_i h}{p_i \omega_i}$</td>
</tr>
<tr>
<td>$B_2$</td>
<td>$\ddot{q}_o + \frac{h}{p_i^2}$</td>
</tr>
<tr>
<td>$C_2$</td>
<td>$-\frac{h}{p_i^2}$</td>
</tr>
<tr>
<td>$A_3$</td>
<td>$-\frac{p_i^2(1-2\beta_i^2)}{\omega_i} \ddot{q}_o + \frac{\beta_i p_i^3}{\omega_i} \dot{q}_o + \frac{\beta_i p_i g}{\omega_i} - \frac{h}{\omega_i}$</td>
</tr>
<tr>
<td>$B_3$</td>
<td>$-2\beta_i p_i \ddot{q}_o - p_i^2 q_o - g$</td>
</tr>
<tr>
<td>$A_4$</td>
<td>$\frac{\beta_i p_i^3(3-4\beta_i^2)}{\omega_i} \ddot{q}_o + \frac{p_i^4(1-2\beta_i^2)}{\omega_i} \dot{q}_o + \frac{p_i^2(1-2\beta_i^2)}{\omega_i} g + \frac{\beta_i p_i}{\omega_i} h$</td>
</tr>
<tr>
<td>$B_4$</td>
<td>$-p_i^2(1-4\beta_i^2) \ddot{q}_o + 2\beta_i p_i^3 q_o + 2\beta_i p_i g - h$</td>
</tr>
</tbody>
</table>
TABLE 2.2
COEFFICIENTS FOR SLIDING RESPONSE $x$

<table>
<thead>
<tr>
<th>$A_5 = -\frac{m}{m_1} A_1$</th>
<th>$A_6 = -\frac{m}{m_1} A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_5 = -\frac{m}{m_1} B_1$</td>
<td>$B_6 = -\frac{m}{m_1} B_2$</td>
</tr>
<tr>
<td>$C_5 = \frac{m}{m_1} (q_o - c_1) + x_0$</td>
<td>$C_6 = \frac{m}{m_1} (\dot{q}_o - c_2) + \dot{x}_0$</td>
</tr>
<tr>
<td>$D_5 = \frac{m}{m_1} (\dot{q}_o - D_1) + \ddot{x}_0$</td>
<td>$D_6 = -a - \frac{m}{m_1 m_2} F \text{ sig } \dot{x}$</td>
</tr>
<tr>
<td>$E_5 = -\frac{a}{2} - \frac{m}{2m_1 m_2} F \text{ sig } \dot{x}$</td>
<td>$E_6 = -\frac{b}{2}$</td>
</tr>
<tr>
<td>$G_5 = -\frac{b}{6}$</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$A_7 = -\frac{m}{m_1} A_3$</th>
<th>$A_8 = -\frac{m}{m_1} A_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_7 = -\frac{m}{m_1} B_3$</td>
<td>$B_8 = -\frac{m}{m_1} B_4$</td>
</tr>
<tr>
<td>$C_7 = -a - \frac{m}{m_1 m_2} F \text{ sig } \dot{x}$</td>
<td>$C_8 = -b$</td>
</tr>
<tr>
<td>$D_7 = -b$</td>
<td></td>
</tr>
</tbody>
</table>

(These coefficients are for the sliding case only)
<table>
<thead>
<tr>
<th></th>
<th>sliding case</th>
<th>non-sliding case</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_9$</td>
<td>$-\frac{m}{m_1}A_3$</td>
<td>0</td>
</tr>
<tr>
<td>$B_9$</td>
<td>$-\frac{m}{m_1}B_3$</td>
<td>0</td>
</tr>
<tr>
<td>$C_9$</td>
<td>$-\frac{m}{m_1 m_2}F \text{ sig } \dot{x}$</td>
<td>a</td>
</tr>
<tr>
<td>$D_9$</td>
<td>0</td>
<td>b</td>
</tr>
<tr>
<td>$A_{10}$</td>
<td>$-\frac{m}{m_1}A_4$</td>
<td>0</td>
</tr>
<tr>
<td>$B_{10}$</td>
<td>$-\frac{m}{m_1}B_4$</td>
<td>0</td>
</tr>
<tr>
<td>$C_{10}$</td>
<td>0</td>
<td>b</td>
</tr>
<tr>
<td>$D_{10}$</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$A_{11}$</td>
<td>$\frac{m}{m_2}A_3$</td>
<td>$A_3$</td>
</tr>
<tr>
<td>$B_{11}$</td>
<td>$\frac{m}{m_2}B_3$</td>
<td>$B_3$</td>
</tr>
<tr>
<td>$C_{11}$</td>
<td>$-\frac{m}{m_1 m_2}F \text{ sig } \dot{x}$</td>
<td>a</td>
</tr>
<tr>
<td>$D_{11}$</td>
<td>0</td>
<td>b</td>
</tr>
<tr>
<td>$A_{12}$</td>
<td>$\frac{m}{m_2}A_4$</td>
<td>$A_4$</td>
</tr>
<tr>
<td>$B_{12}$</td>
<td>$\frac{m}{m_2}B_4$</td>
<td>$B_4$</td>
</tr>
<tr>
<td>$C_{12}$</td>
<td>0</td>
<td>b</td>
</tr>
<tr>
<td>$D_{12}$</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
CHAPTER 3
RESULTS OF PROGRAM DEVELOPMENT

The method chosen for solving the differential equations of motion is intended to require a minimum of computation time while retaining much flexibility in input options. The subroutines which search for zeroes or for maxima of a function take advantage of the periodic nature of the response quantities, thereby avoiding repetitious evaluations of trigonometric functions in their searches.

In those subroutines which use Newton's Method, special provision is made for functions such as those which approximate \((1 - \cos x)\). In such cases, roundoff may prevent the convergence of the solution to within the normally required tolerance. Newton's method is therefore terminated after fifteen iterations to avoid infinite looping, since the precision attained under such conditions is adequate. If such a termination should occur during a given case, a diagnostic is printed beside the results of that case.

For a given system and accelerogram, both positive and negative maxima of the response quantities \(q, \dot{q}, \ddot{q}, x, \dot{x}, \ddot{x}, \dddot{u}, \) and \(\dddot{v}\) are calculated, as well as the global time associated with each maximum. Not all of these quantities are printed in the program output, but provision is made in Appendix A for examination of any of these quantities.

The accuracy of the program was verified in two ways. Cases which reduce to a single-degree-of-freedom system were run, and the results were compared with those given by Bielak and found to be in excellent agreement. Cases which do not reduce to a single-degree-of-freedom system were examined by use of a time-history printout option. Poten-
tial, kinetic, and frictional energy of the system were obtained from the masses, stiffnesses, displacements, and velocities. Examination of these quantities produced confidence in the program's capability and accuracy, insuring that the program neither adds energy to nor subtracts energy from the system. The result is a program capable of solving efficiently a large number of cases using a minimum amount of core (128K) for a wide variety of accelerograms. The program is fully documented in Appendix A for ease of use and modification.
CHAPTER 4
STUDY OF RESPONSE DUE TO HALF-CYCLE VELOCITY PULSE

The response of the system due to the half-cycle velocity pulse shown in Figure (2.2) was evaluated at three representative values of \( \rho \), the frequency ratio:

1. \( \rho = .95 \) (\( m_1 \) constitutes 90\% of total mass \( M \))
2. \( \rho = .7071 \) (\( m_1 = m_2 \))
3. \( \rho = .4 \) (\( m_2 \) constitutes 84\% of total mass \( M \))

The case \( \beta = 0 \) was examined to make the effects of the other parameters more pronounced in the frequency region considered.

For each value of \( \rho \), the friction factor \( \text{FF} \) was varied to obtain the entire range of possible responses. One limiting case is that of \( \text{FF} = 0 \), for which the relative response \( q \) is zero, while the sliding displacement \( x \) is identically equal to the ground displacement \( y \). The other limiting case is a value of \( \text{FF} \) for which sliding never occurs, so that \( x \) is zero. This system reduces to a single-degree-of-freedom system, one which agrees with Bielak's results. All other cases fall between these two extremes.

As with any multi-degree-of-freedom system, several alternatives were possible for normalization and presentation of the results. The most logical approach was to compare the responses of the sliding systems to similar non-sliding systems for which the results were known. Therefore, the systems were described in terms of their non-sliding, rather than their sliding parameters (hence \( f_{2\text{ref}} \) and \( \beta_2 \) were used as nondi-
The maximum value $q_m$ of the relative response $q$ was normalized with respect to the maximum ground displacement $y_m$. $q_m$ also determines a relative pseudovelocity $V = p^2 q_m$ and a relative pseudoacceleration $A = p^2 q_m$, which were normalized appropriately with respect to the maximum ground velocity and acceleration $\dot{y}_m$ and $\ddot{y}_m$ (see Figures 4.1-4.3).

Relative Response $q$. The quantities $q_m/y_m$, $V/\dot{y}_m$, and $A/\ddot{y}_m$ were plotted logarithmically against the frequency parameter $f^2 t_{ref}$. Spectra for all three $\rho$ values showed the same general characteristics as the single-degree-of-freedom response spectrum. In the low-frequency region, the spectra tend to constant displacement. For $\rho = .95$, the displacement is about 15% less than that of the single-degree-of-freedom system; for the lower $\rho$ values, the responses are the same as that of the single-degree-of-freedom system. In the medium-frequency region, the peak of the normalized pseudovelocity is reduced by about 50% for all $\rho$ values. In the high-frequency region, the normalized pseudoacceleration tends to about .5 for all values of $\rho$. Overall, $q$ is not extremely sensitive to variation in $\rho$, which is a surprising result.

Sliding Response $x$. The maximum value $x_m$ of the sliding response $x$ was normalized with respect to the maximum ground displacement $y_m$ (see Figures 4.4-4.6) and plotted logarithmically against $f^2 t_{ref}$. The effect of $\rho$ is more pronounced on $x$ than on $q$. For medium and low values of $\rho$, breakloose of mass $m_1$ is observed in a limited range of frequencies only, while for $\rho = .95$, breakloose occurs for a wide range of frequencies, even for higher values of FF. At high values of $\rho$, mass
$m_1$ is dominant, and the frequency of the system is less important because the response is dependent on the magnitude of the acceleration. At low values of $\rho$, mass $m_2$ is dominant, and the spring in the system plays a more important part in preventing mass $m_1$ from sliding at low system frequencies, even for low values of FF.
FIGURE 4.2
HALF-CYCLE VELOCITY PULSE

\[ \frac{p_2 q}{\dot{V}_m} = \frac{V}{V_m} \]

\( p = 0.7071 \)

\( \beta_2 = 0 \)

FF = 1.8 (no sliding)

\[ \frac{p_2^2 q}{\dot{V}_m} = \frac{A}{V_m} \]
FIGURE 4.3
HALF-CYCLE VELOCITY PULSE

\[ \frac{p_2 q}{V_{m}} = \frac{V}{V_m} \]

\[ \rho = .4 \]
\[ \beta_2 = 0 \]

FF = 2.6 (no sliding)

\[ \frac{p_2^2 q}{V_{m}} = \frac{A}{V_m} \]
\[ \frac{x_m}{V_m} \]

**FIGURE 4.4**

HALF-CYCLE VELOCITY PULSE

\[ \rho = 0.95 \]
\[ \beta_2 = 0 \]

\[ FF = 0.2 \]
\[ FF = 0.4 \]
\[ FF = 0.6 \]
\[ FF = 0.8 \]
\[ FF = 1.0 \]

\[ f_2 t_{\text{ref}} \]
FIGURE 4.5
HALF-CYCLE VELOCITY PULSE

\[ \frac{x_m}{Y_m} \]

\( \rho = 0.7071 \)
\( \beta_2 = 0 \)

\( FF = 0.2 \)
\( FF = 0.6 \)
\( FF = 1.0 \)
\( FF = 1.4 \)
Figure 4.6
Half-Cycle Velocity Pulse

\[ \frac{x_m}{y_m} \]

- FF = 0.2
- FF = 0.6
- FF = 2.2
- FF = 1.4

\[ \rho = 0.4 \]
\[ \beta_2 = 0 \]
CHAPTER 5
CONCLUSIONS

Full documentation and input description for the computer program developed in this investigation are provided in Appendix A. The program is listed in Appendix B.

The program may be used as a direct analysis/design aid for specific problems such as cranes, given the natural frequencies, sliding forces, damping factors, and masses of a system. Analysis of response due to a half-cycle velocity pulse (Chapter 4) showed the relative displacement between the masses (which determines the stresses induced in a structure) to be similar to, but reduced in amplitude from, the relative displacement of a single-degree-of-freedom system. This displacement is relatively insensitive to changes in the ratio of non-sliding to free-free natural frequencies, and therefore relatively insensitive to changes in the ratio of the two masses. Thus the method in which the mass of a continuous system such as a crane is lumped into the two masses in the program would not be critical to the evaluation of the internal stresses induced by ground excitation.

The program may also be used as a tool for further investigation of the system considered to observe response trends for other types of accelerograms, such as earthquakes. Investigation of other methods of nondimensionalizing the system parameters could prove worthwhile.
REFERENCES


A.1 Running the Program.

A.1.1 Accelerogram Description. The program requires a piecewise linear accelerogram, which is supplied in two arrays, \( T \) and \( ACC \). The ground acceleration is assumed to begin at time zero and to end at time \( T(N) \). If \( T(1) \) is nonzero, the initial segment of the accelerogram begins at zero acceleration. The value of \( ACC(N) \) need not be zero. Arrays for two typical pulses are shown below:
The program is designed to run large numbers of cases for a given accelerogram; however, a single case is easily run.

A.1.2 Time-History Printout Option. It may be desired to obtain a time-history of the response of a particular system to a given accelerogram. The time-history option prints the relative displacements and velocities of both masses at any desired time increment.

A.1.3 Preparation of Input. Input data must be punched onto cards in the sequence and FORTRAN formats below:

1. Option card (I5, 4F10.0)
   
   M  time-history printout option switch (M=1 for printout)
   EPS tolerance for Newton's Method (suggested value 10^-4)
   TREF reference time for accelerogram
   DT  time increment for time-history printout (immaterial for m ≠ 1)
   TEND time to end time-history printout (immaterial for m ≠ 1)

2. Accelerogram information card (F10.0, 2I5, 3F10.0)
   
   FACT scale factor by which accelerogram is to be multiplied
   N  number of points in accelerogram (maximum = 100)
   ISW accelerogram printout option (ISW=0 for printout)
   YM  maximum ground displacement
   YMD maximum ground velocity
   YMDD maximum ground acceleration
   (program will compute these if left blank.)

3. Accelerogram format card (20A4)
   
   FORM format in which accelerogram will be supplied - must be in parentheses. Example (8F10.0)

4. Accelerogram cards (as needed to supply N accelerogram points in format FORM above)
T(j)  accelerogram global time (abscissa)
ACC(j)  ground acceleration (ordinate) corresponding to T(j)
j = 1 to N, alternating T(j) and ACC(j)

5. Frequency ratio header card ( I5 )
   NR  number of frequency ratio (\( \rho \)) values to be supplied (maximum NR = 20)

6. Frequency ratio cards ( 8F10.0 ), one or more, as required
   RLIST(20) frequency ratio (\( \rho \)) list

7. Damping factor header card ( I5 )
   NB  number of damping factor (\( \beta_2 \)) values to be supplied (maximum NB = 20)

8. Damping factor cards ( 8F10.0 ), one or more, as required
   BLIST(20) damping factor (\( \beta_2 \)) list

9. Friction factor header card ( I5 )
   NF  number of friction factor (FF) values to be supplied (maximum NF = 100)

10. Friction factor cards ( 8F10.0 ), one or more, as required
    FFLIST(100) friction factor (FF) list

11. Frequency parameter header card ( I5 )
    NT  number of frequency parameter (\( f_2 t_{ref} \)) values to be supplied (maximum NT = 100)

12. Frequency parameter cards ( 8F10.0 ), one or more, as required
    FTLIST(100) frequency parameter (\( f_2 t_{ref} \)) list

Cards 2 through 12 may be repeated as needed.
A.2 Program Documentation

A.2.1 Organization. The program is organized into a main program and twenty-seven subroutines, written in FORTRAN IV for the IBM 370/155. A brief description of each subroutine is given below. More detailed descriptions follow.

**MAIN**
Controls stepwise integration of the differential equations of motion. Calls PARAM, CONST, GROUND, COEFQ, COEXUV, TIMEE, HIST, F, ONEMAX, SLPCK, GETMAX, and OUTPUT.

**PARAM**
Reads accelerogram and dimensionless parameters; computes required system coefficients. Calls GRNDRD.

**GRNDRD**
Initially reads accelerogram. Calls AGRAM.

**AGRAM**
Integrates accelerogram to obtain maximum ground displacement, velocity, and acceleration.

**CONST**
Calculates constants required for a given system.

**GROUND**
Furnishes current accelerogram segment.

**COEFQ**
Computes coefficients for displacements q.

**COEXUV**
Computes coefficients for displacements x, u, and v.

**TIMEE**
Equation solver (quadratic function plus sine wave). Calls QUADR, TIMED, F, TANMON, SOLVEE, and TANPTE.

**QUADR**
Solves quadratic equation.

**TIMED**
Equation solver (linear function plus sine wave). Calls TIMEC, F, TANPT, and SOLVET.

**TIMEC**
Equation solver (constant plus sine wave). Calls TIMEB, F, and SOLVET.
TIMEB  Equation solver (sine wave).

F     Function evaluator.

SOLVET Newton's Method routine (linear function plus sine wave).

TANPT Tangent point equation solver (linear function plus sine wave). Calls F and TIMEC.

TANMON Tangent point equation solver (quadratic function plus sine wave - monotonic case). Calls F and SOLVEE.

SOLVEE Newton's Method routine (quadratic function plus sine wave).

TANPTE Tangent point equation solver (quadratic function plus sine wave). Calls F, TIMED, and SOLVEE.

HIST  Prints a time-history of responses.

ONEMAX Revises extrema.

SLPCK Determines whether system slides in a given time interval. Calls QKSLP and SLPFND.

QKSLP Determines whether system on verge of sliding will slide or not.

SLPFND Determines time and direction of sliding. Calls TIMEC, SOLVET, F, and TANPT.

GETMAX Computes maxima of responses. Calls ONEMAX, EMAX, TIMEC, F, and DMAX.

EMAX  Maximizing routine (quadratic function plus sine wave).

DMAX  Maximizing routine (linear function plus sine wave).

OUTPUT Prints desired maximum responses.
A.2.2 Quantities Available for Output. The program calculates a number of quantities which are not printed in the OUTPUT routine. With minor modifications to that routine, any of the following quantities can be examined:

<table>
<thead>
<tr>
<th>RESPONSE QUANTITY</th>
<th>MAXIMUM POSITIVE VALUE</th>
<th>TIME OF MAXIMUM POSITIVE VALUE</th>
<th>MAXIMUM NEGATIVE VALUE</th>
<th>TIME OF MAXIMUM NEGATIVE VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>q</td>
<td>QMP</td>
<td>TQMP</td>
<td>QMN</td>
<td>TQMN</td>
</tr>
<tr>
<td>q</td>
<td>QMPD</td>
<td>TQMPD</td>
<td>QMND</td>
<td>TQMND</td>
</tr>
<tr>
<td>q</td>
<td>QMPDD</td>
<td>TQMPDD</td>
<td>QMNDD</td>
<td>TQMNDD</td>
</tr>
<tr>
<td>x</td>
<td>XMP</td>
<td>TXMP</td>
<td>XMN</td>
<td>TXMN</td>
</tr>
<tr>
<td>x</td>
<td>XMPD</td>
<td>TXMPD</td>
<td>XMND</td>
<td>TXMND</td>
</tr>
<tr>
<td>x</td>
<td>XMPDD</td>
<td>TXMPDD</td>
<td>XMNDD</td>
<td>TXMNDD</td>
</tr>
<tr>
<td>u</td>
<td>UMPDD</td>
<td>TUMPDD</td>
<td>UMNDD</td>
<td>TUMNDD</td>
</tr>
<tr>
<td>v</td>
<td>VMPDD</td>
<td>TVMPDD</td>
<td>VMNDD</td>
<td>TVMNDD</td>
</tr>
</tbody>
</table>

A.2.3 Detailed Subroutine Description. Documentation begins with MAIN; the rest are in alphabetical order. The numbers in parentheses at the right of each step are the labels on the right ends of the FORTRAN cards.

**MAIN**

Variables

A(12) array of coefficients of \((\sin \omega t)\) in response quantities

AIN initial acceleration for current time step

AM mass constant \(m = \frac{m_1 m_2}{m_1 + m_2}\)
AMA mass $m_1$
AMB mass $m_2$
AMC total mass of system $m_1 + m_2$
B(12) array of coefficients of $(\cos \omega t)$ in response quantities
BETA(2) array of damping factors $\beta_1, \beta_2$
BIG $10^{-10}$, a flag
BP(2) array of damping factors $\beta_1 P_1, \beta_2 P_2$
BSL slope of current accelerogram segment
C(12) array of constant terms in response quantities
CO(32) array of constants used to compute coefficients of response quantities
D(12) array of coefficients of linear terms in response quantities
DT time increment for time-history printout
EPS tolerance for Newton's Method
EX(3) array of coefficients of quadratic and cubic terms in response quantities
FORCE value of friction force $s$ at $t = 0$ (non-sliding case)
FRIC value of maximum friction force $F$ (sliding case)
I system state index ($I=1$ for sliding; $I=2$ for non-sliding)
J cut-off counter - stops response computation
M time-history printout indicator
P(2) array of undamped circular natural frequencies $p_1, p_2$
QMN maximum negative value of $q$
QMND maximum negative value of $\dot{q}$
QMNDD maximum negative value of $\ddot{q}$
QMP maximum positive value of $q$
QMPD maximum positive value of $\dot{q}$
QMPDD maximum positive value of $\ddot{q}$
QO  value of $q$ at beginning of current time interval
QOD value of $\dot{q}$ at beginning of current time interval
S  sliding direction indicator (S=-1 for sig $\dot{x} = -1$; S=0 for non-sliding case; S=1 for sig $\dot{x} = 1$)
TB  time of system state change in current time interval
TBG global time of system state change
TEND time to stop time-history printout
TF  time length of current accelerogram segment
THIS global time of next time-history printout
TLIM minimum of (TB,TF)
TQMN time of maximum negative value of $q$
TQMND time of maximum negative value of $\dot{q}$
TQMNDD time of maximum negative value of $\ddot{q}$
TQMP time of maximum positive value of $q$
TQMPD time of maximum positive value of $\dot{q}$
TQMPDD time of maximum positive value of $\ddot{q}$
TREF reference time for given accelerogram
TT  global time corresponding to $t = 0$
TUMNDD time of maximum negative value of $\dddot{u}$
TUMPDD time of maximum positive value of $\dddot{u}$
TVMNDD time of maximum negative value of $\dddot{v}$
TVMPDD time of maximum positive value of $\dddot{v}$
TXMN time of maximum negative value of $x$
TXMND time of maximum negative value of $\dot{x}$
TXMNDD time of maximum negative value of $\ddot{x}$
TXMP time of maximum positive value of $x$
TXMPD time of maximum positive value of $\dot{x}$
The program proceeds in the steps below.

1. Read option card

2. Call PARAM to obtain necessary parameters for a given excitation and system; stop if no data

3. Call CONST to calculate constants used in response coefficients
4. Initialize response quantities, counters, and indices (0070-0490)
5. Call GROUND to obtain next accelerogram segment (0520)
6. Optional print (0530)
7. If sliding (I=1) go to step 9 (0540)
8. Call COEFQ to calculate response coefficients for q; go to step 13 (0550-0560)
9. Error check -- if XOD x S > 0 go to step 12 (0564)
10. Roundoff error occurred -- cease sliding; optional print (0565)
11. Go to step 43 (0566)
12. Call COEFQ to calculate response coefficients for q (0570)
13. Call COEXUV to calculate response coefficients for x, u, and v (0580)
14. If not sliding (I=2) go to step 26 (0590)
15. Use TIMEE to compute time when sliding ceases TB (0600)
16. If TB = BIG , sliding will not cease in current time interval; go to step 18 (0602)
17. Calculate TBG, optional print (0604-0610)
18. Calculate TLIM (0620)
19. Calculate XODD, UODD, VODD ; call ONEMAX to compare with previous maxima (0630-0680)
20. Optional printout of time-history on (0,TLIM) (0690)
21. Call GETMAX to maximize $\ddot{x},\dot{x}$ (0700-0730)
22. Calculate XO,XOD for next time step (0740-0750)
23. Call ONEMAX to maximize x (0770)
24. If TLIM = TB , system ceases sliding in current time interval; set S = 0 (0780)
25. Go to step 34 (0790)
26. Calculate FORCE (0800)
27. If \text{FORCE} > \text{FRIC}, system begins sliding immediately; go to step 41. (0820)

28. If \text{FORCE} < -\text{FRIC}, system begins sliding immediately; go to step 42. (0830)

29. Call \text{SLPCK} to obtain time when sliding begins \text{TB} and associated direction \text{S}. (0840-0850)

30. If \text{TB} = \text{BIG}, sliding will not occur in current time interval, go to step 32. (0862)

31. Calculate \text{TBG}, optional print. (0864-0870)

32. Calculate \text{TLIM}. (0875)

33. Optional printout of time-history on (0,\text{TLIM}). (0880)

34. Call \text{GETMAX} to maximize \text{q}, \dot{\text{q}}, \ddot{\text{q}}, \text{u}, \ddot{\text{v}}. (0890-0980)

35. If \text{TLIM} = \text{BIG}, excitation completed and sliding will not occur again, go to step 48. (0990)

36. Calculate \text{QO}, \text{QOD}, \text{TT} for next time step. (1000-1030)

37. Optional print. (1035)

38. If \text{TLIM} = \text{TF}, end of current accelerogram segment; go to step 5. (1040)

39. If \text{TF} = \text{BIG}, excitation completed, go to step 43. (1060)

40. Calculate \text{TF} and \text{AIN} for next time step. (1070-1080)

41. Set \text{S} = -1, go to step 43. (1100-1110)

42. Set \text{S} = 1. (1120)

43. System state change, change \text{I}. (1130)

44. Set \text{XOD} = 0. (1135)

45. If \text{J} = 2, no further response computation; go to step 48. (1140)

46. If \text{TF} = \text{BIG}, increment \text{J} by 1. (1145)

47. Go to step 7. (1150)

48. Call \text{OUTPUT} to print results for current system. (1160-1165)

49. Go to step 2. (1180)
AGRAM

AGRAM calculates maximum (absolute value) ground displacement, velocity, and acceleration from a given accelerogram which varies piecewise linearly. For a given segment,

\[
\ddot{y} = a + bt \\
\dot{y} = \dot{y}_0 + at + \frac{1}{2}bt^2 \\
y = y_0 + \dot{y}_0 t + \frac{1}{2}at^2 + \frac{1}{6}bt^3
\]

Arguments are:

YM       maximum ground displacement
YMD      maximum ground velocity
YMDD     maximum ground acceleration
ACC(100) array of acceleration ordinates
T(100)   array of global times corresponding to accelerations in ACC
N        length of arrays T, ACC

Other variables are:

BB        current slope of accelerogram
D         scratch variable
DT        time length of current accelerogram segment
I         scratch counter
TA        time of relative extremum of \( y \)
TB        time of relative extremum of \( \dot{y} \)
Y         relative extremum of \( y \)
YD        relative extremum of \( \dot{y} \)
YO        initial displacement in current accelerogram segment
YOD       initial velocity in current accelerogram segment
Procedure:

1. Initialization
   
2. Maximize current $\ddot{y}$ by comparison with previous $\ddot{y}$
   
3. Calculate BB, DT
   
4. Calculate relative maximum $\dot{y}$ by setting $\ddot{y} = 0$, compare with previous maximum $\ddot{y}$
   
5. Calculate relative maximum $y$ by setting $\ddot{y} = 0$, compare with previous maximum $y$
   
6. Calculate $y$ and $\dot{y}$ at end of current accelerogram segment, compare with previous maxima
   
7. Do steps 2-6 for each accelerogram segment
   
8. Return

**COEFQ**

COEFQ calculates coefficients necessary for evaluation of relative responses $q$, $\dot{q}$, $\ddot{q}$, and $\dddot{q}$. Depending on the values of arguments supplied, the coefficients may be calculated for either the sliding or non-sliding case. No procedure is given for this routine.

Arguments are:

- **A(12)** array of coefficients of $(\sin \omega t)$ terms
- **B(12)** array of coefficients of $(\cos \omega t)$ terms
- **C(12)** array of constant terms
- **D(12)** array of coefficients of linear terms
- **QO** value of $q$ at $t = 0$
- **QOD** value of $\dot{q}$ at $t = 0$
- **AIN** initial ground acceleration (non-sliding case) or frictional acceleration (sliding case)
- **BSL** ground jerk (non-sliding case) or zero (sliding case)
CO(32) array of constants

**COEXUV**

COEXUV calculates coefficients necessary for evaluation of responses \( x, \dot{x}, \ddot{x}, \dddot{x}, \dot{v}, \) and \( \ddot{v} \). Depending on the values of arguments supplied, the coefficients may be calculated for either the sliding or non-sliding case. No procedure is given for this routine.

Arguments are:

- A(12) array of coefficients of \((\sin \omega t)\) terms
- B(12) array of coefficients of \((\cos \omega t)\) terms
- C(12) array of constant terms
- D(12) array of coefficients of linear terms
- EX(3) array of coefficients of quadratic and cubic terms
- CO(32) array of constants
- Q0 value of \( q \) at \( t = 0 \)
- QOD value of \( \dot{q} \) at \( t = 0 \)
- XO value of \( x \) at \( t = 0 \)
- XO value of \( \dot{x} \) at \( t = 0 \)
- FS \( F \times \text{sig} \dot{x} \) (sliding case) or zero (non-sliding case)
- AIN initial ground acceleration for current time step
- BSL ground jerk for current time step

**CONST**

CONST calculates those constants which are used repeatedly for a given system and stores them in an array. No procedure is given for this routine.
Arguments are:

- **C(32)** array of constants
- **B(2)** array of damping factors $\beta_1, \beta_2$
- **P(2)** array of undamped circular natural frequencies $p_1, p_2$
- **W(2)** array of damped circular natural frequencies $\omega_1, \omega_2$
- **AM** mass constant $m$
- **AMA** mass constant $m_1$
- **AMB** mass constant $m_2$

**DMAX**

DMAX obtains the absolute maximum and minimum of a function of the form

$$f = e^{-\beta pt}(A_1 \sin \omega t + B_1 \cos \omega t) + C_1 + D_1 t$$

on an interval $(TA, TF)$ by examining zeros of its first derivative on that interval. Derivatives are of the form

$$f' = e^{-\beta pt}(A_2 \sin \omega t + B_2 \cos \omega t) + C_2$$
$$f'' = e^{-\beta pt}(A_3 \sin \omega t + B_3 \cos \omega t)$$

Arguments are:

- **A(3)** array of coefficients of $(\sin \omega t)$ terms
- **B(3)** array of coefficients of $(\cos \omega t)$ terms
- **C(2)** array of constant terms
- **D(1)** coefficient of linear term
- **BP** damping factor $\beta p$
W damped circular frequency \( \omega \)
TF interval end time
TA time of extremum of \( f \)
EPS tolerance for Newton's Method
FMP maximum positive value of \( f \)
FMN maximum negative value of \( f \)
TFMP time of maximum positive value of \( f \)
TFMN time of maximum negative value of \( f \)
TT global time corresponding to \( t = 0 \)

Other variables are:
E \( e^{-2\pi \beta p/\omega} \)
FA value of \( f \) at \( t = TA \)
FI value of \( f' \) at \( t = TI \)
P \( \pi \)
PP \( e^{-\beta pt}(A_2 \sin \omega t + B_2 \cos \omega t) \) at \( t = TI \)
TI time of extremum of \( f' \)
TP \( 2\pi \)

Procedure:
1. Use TIMEB to obtain TI  
2. Increment TI by \( 2\pi/\omega \) until \( TI > TA \)  
3. Compute PP and E  
4. Increment TI by \( 2\pi/\omega \) and evaluate FI  
5. If \( D_1 < 0 \) go to step 8  
6. If \( FI > 0 \), \( f \) is monotonic on \((TI, TF)\), return  
7. Go to step 9  
8. If \( FI \leq 0 \), \( f \) is monotonic on \((TI, TF)\), return  
9. Use SOLVET at initial time \( TA + 2\pi/\omega \) to find next \( TA \), solving \( f' = 0 \)
10. If $TA > TF$ return (0170)
11. Evaluate FA (0180)
12. If $D_1 < 0$ go to step 16 (0190)
13. If $FA < FMP$ go to step 4 (0200)
14. Revise FMP and TFMP (0210-0220)
15. Go to step 4 (0230)
16. If $FA > FMN$ go to step 4 (0240)
17. Revise FMN and TFMN (0250-0260)
18. Go to step 4 (0270)
EMAX obtains the absolute maximum or minimum of a function of the form

\[ f = e^{-\beta pt}( A_1 \sin \omega t + B_1 \cos \omega t ) + C_1 + D_1 t + E t^2. \]

on an interval (0,TF) by examining zeros of its first derivative on that interval. Derivatives are of the form

\[ f' = e^{-\beta pt}( A_2 \sin \omega t + B_2 \cos \omega t ) + C_2 + D_2 t \]
\[ f'' = e^{-\beta pt}( A_3 \sin \omega t + B_3 \cos \omega t ) + C_3 \]

It is assumed that \( f \) does not change sign on (0,TF).

Arguments are:

A(3) array of coefficients of (sin \( \omega t \)) terms
B(3) array of coefficients of (cos \( \omega t \)) terms
C(3) array of constant terms
D(2) array of coefficients of linear terms
E coefficient of quadratic term of \( f \)
BP damping factor \( \beta p \)
W damped circular natural frequency \( \omega \)
TF interval end time
EPS tolerance for Newton's Method
FMP maximum positive value of \( f \)
FMN maximum negative value of \( f \)
TFMP time of maximum positive value of \( f \)
TFMN time of maximum negative value of \( f \)
TT global time corresponding to \( t = 0 \)

Other variables are:

- **BIG** \( 10^{10} \), a flag
- **EE** \( e^{-2\pi\beta p/\omega} \)
- **FA** value of \( f \) at \( t = TA \)
- **FAB** sum of \( FA \) and \( FB \)
- **FB** value of \( f \) at \( t = TB \)
- **FC** value of \( f' \) at \( t = TC \)
- **FF** value of \( f \) at \( t = TF \)
- **HP** \( \pi/2 \)
- **P** \( \pi \)
- **PP** \( e^{-\beta pt}( A_2 \sin \omega t + B_2 \cos \omega t ) \) at \( t = TC \)
- **TA** time of extremum of \( f \)
- **TB** time of second extremum of \( f \)
- **TC** time of tangent point of \( f' \) to its decay envelope
- **TFIN** time corresponding to vertex of parabola \( C_1 + D_1 t + E t^2 \)
- **TP** \( 2 \pi \)
- **WT** scratch variable

Procedure:

1. Use TIMED to calculate \( TA \)  
   (0040)
2. If \( TA = BIG \), \( f \) is monotonic, go to step 34  
   (0050)
3. Calculate first extremum \( FA \)  
   (0060)
4. Call ONEMAX to revise maxima  
   (0070)
5. Use TIMED to calculate \( TB \)  
   (0080)
6. If \( TB = BIG \), \( f \) is monotonic after the first extremum, go to step 34  
   (0090)
7. Calculate second extremum \( FB \)  
   (0100)
8. Call ONEMAX to revise maxima

9. Calculate FAB; if FAB > 0, only maxima will be examined; go to step 11

10. Only minima will be examined; set TA equal to time of first minimum; go to step 12

11. Set TA equal to time of first maximum TB

12. Obtain first value for tangent point TC (see 2.4.5)

13. Increment TC by $2\pi/\omega$ until TC > TA

14. If FAB x E > 0, go to step 19

15. Decrease TC by $2\pi/\omega$ so that TC is less than TA

16. Calculate TFIN; f will be examined on (TA, TFIN)

17. If TFIN < TA go to step 34, see Fig. A.3

18. Go to step 20

19. Set TFIN = TF, see Fig. A.2

20. Calculate PP and E

21. Increment TC by $2\pi/\omega$, calculate new PP, FC

22. Set TB = TA + $2\pi/\omega$ as estimate for next extremum of f

23. If FC x D > 0, f may have one further extrema, go to step 30

24. Use SOLVET at t = TB to obtain next extremum TA

25. If TA > TF, go to step 34

26. Calculate FA

27. Call ONEMAX to revise extrema

28. If TA ≥ TFIN, go to step 34

29. Go to step 20

30. Use TIMED to find last extremum TA, if any

31. If TA = BIG, go to step 34

32. Calculate FA
33. Call ONEMAX to revise maxima

34. If TF = BIG, return

35. Calculate FF

36. Call ONEMAX to revise maxima

37. Return
FUNCTION F

F evaluates the function

\[ e^{-\beta p t} ( A \sin \omega t + B \cos \omega t ) + C + D t \]

Arguments are:

A
B coefficients above
C
D
BP damping factor \( \beta p \)
W damped circular natural frequency \( \omega \)
T \( t \), above

Other variables are:

WT scratch variable \( \omega t \)

No procedure is given for this routine.

GETMAX

GETMAX computes maximum or minimum values of a function of the form:

\[ f = e^{-\beta p t} ( A_1 \sin \omega t + B_1 \cos \omega t ) + C_1 + D_1 t + E t^2 \]

on an interval \((0,TF)\). Its derivative is of the form

\[ f' = e^{-\beta p t} ( A_2 \sin \omega t + B_2 \cos \omega t ) + C_2 + D_2 t \]
Arguments are:

A(2) array of coefficients of \((\sin \omega t)\) terms
B(2) array of coefficients of \((\cos \omega t)\) terms
C(2) array of constant terms
D(2) array of coefficients of linear terms
E coefficient of quadratic term
BP damping factor \(\beta_p\)
W damped circular frequency
TF duration of current time step
EPS tolerance for Newton's Method
FMP maximum positive value of \(f\)
FMN maximum negative value of \(f\)
TFMP time of maximum positive value of \(f\)
TFMN time of maximum negative value of \(f\)
TT global time corresponding to \(t=0\)

Other variables are:

BIG \(10^{10}\), a flag
FA value of \(f\) at \(t=TA\)
FB value of \(f\) at \(t=TB\)
FF value of \(f\) at \(t=TF\)
FV value of \(f\) at \(t=TV\)
P \(\pi\)
TA time of first extremum of \(f\)
TB time of second extremum of \(f\)
TP \(2\pi\)
TV time corresponding to vertex of parabola \(C_1 + D_1 t + E t^2\)
Procedure:
1. If $E=0$ go to step 10
2. If either $A_1$ or $B_1 \neq 0$ go to step
3. $f$ is of the form $C_1 + D_1 t + Et^2$, calculate $TV$
4. If $TV < 0$ or $TV > TF$ go to step 24
5. Calculate $FV$
6. Call ONEMAX to revise extrema
7. Go to step 24
8. Call EMAX to examine extrema of $f$
9. Return
10. Use TIMEC to compute $TA$
11. If $TA \geq TF$ go to step 24
12. Calculate $FA$
13. Call ONEMAX to revise maxima
14. Use TIMEC to compute $TB$
15. If $TB \geq TF$ go to step 24
16. Calculate $FB$
17. Call ONEMAX to revise maxima
18. If $D_1=0$ return
19. If $D_1 < 0$ go to step 22
20. Only maxima will be examined; if $FB > FA$ set $TA=TB$
21. Go to step 23
22. Only minima will be examined; if $FB < FA$ set $TA=TB$
23. Call DMAX to examine extrema of $f$
24. If $TF=BIG$, return
25. Calculate $FF$
26. Call ONEMAX to revise extrema (0400)
27. Return (0410)

**GROUND or GRNDRD**

GROUND furnishes time duration TF, initial acceleration a, and slope b of the current accelerogram segment.

GRNDRD reads the entire accelerogram and related parameters initially.

Arguments of GROUND are:
- TF: interval end time
- A: initial acceleration for current segment
- B: slope of accelerogram for current segment
- I: index of current accelerogram array element

Arguments of GRNDRD are:
- YM: maximum ground displacement $y_m$
- YMD: maximum ground displacement $\dot{y}_m$
- YMDD: maximum ground displacement $\ddot{y}_m$

Other variables are:
- ACC(100): array of acceleration ordinates
- FACT: accelerogram scale factor
- FORM(20): accelerogram card format
- ISW: optional accelerogram printout switch
- J: scratch index
- N: length of arrays ACC and T
- T(100): array of times corresponding to ACC
Procedure:

GROUND

1. If I ≠ 1 go to step 6 (0030)
2. First accelerogram segment; if T(1) ≠ 0, go to step 4 (0040)
3. Set I = 2, go to step 10 (0050-0060)
4. Set A = 0
   \[ B = \frac{\text{ACC}(1)}{T(1)} \]
   \[ TF = T(1) \]
   \[ I = 2 \]
   (0070-0100)
5. Return (0110)
6. If I > N, go to step 12 (0120)
7. If T(I) ≠ T(I-1) go to step 10 (0130)
8. Discontinuity in accelerogram, set I = I + 1 (0140)
9. If I > N, go to step 12 (0150)
10. Set \[ A = \text{ACC}(I-1) \]
    \[ B = \frac{[\text{ACC}(I) - \text{ACC}(I-1)]}{[T(I) - T(I-1)]} \]
    \[ TF = T(I) - T(I-1) \]
    \[ I = I + 1 \]
    (0160-0190)
11. Return (0200)
12. Last segment of accelerogram; set A = 0
    \[ B = 0 \]
    \[ TF = 10^{10} \]
    \[ I = 1 \]
    (0210-0235)
13. Return (0240)

GRNDRD

14. Read FACT, N, ISW, YM, YMD, YMDD; if no data, go to step 23 to stop program execution (0260)
15. Read FORM (0300)
16. Read T, ACC according to FORM and N (0320)
17. If ISW ≠ 0 go to step 19 (0330)
18. Optional printout of accelerogram (0340-0360)
19. If FACT=1 go to step 21

20. Multiply array ACC by FACT

21. If any of YM, YMD, YMDD = 0 (i.e., are unknown), call AGRAM to compute them

22. Return

23. Stop

HIST

HIST prints a time-history of the system responses q, \dot{q}, x, \dot{x} at time increments DT for detailed examination of the system's response.

Arguments are:

THIS \quad \text{global time of next printout}

TT \quad \text{global time corresponding to } t=0

TLIM \quad \text{duration of a given time step}

DT \quad \text{time increment for printout}

TEND \quad \text{time to stop printout}

A(12) \quad \text{arrays of response coefficients}

B(12)

C(12) \quad \text{arrays of response coefficients}

D(12)

EX(3)

BP(2) \quad \text{array of damping constants } \beta_p

W(2) \quad \text{array of damped circular natural frequencies } \omega

I \quad \text{system state index}

Other variables are

QA \quad \text{value of q at time TA}
QAD value of $\dot{q}$ at time TA

TA local time of next printout

XA value of $x$ at time TA

XAD value of $\dot{x}$ at time TA

Procedure:

1. If THIS $>$ TEND return

2. Calculate TA

3. If TA $>$ TLIM no more printout, return

4. Print heading

5. Calculate and print QA, QAD, XA, XAD

6. Increment THIS, TA

7. If THIS $>$ TEND return

8. If TA $>$ TLIM no more printout, return

9. Go to step 5

ONEMAX

ONEMAX compares the current value of a function with supplied positive and negative maxima, and revises, if necessary, a maximum and its corresponding global time.

Arguments are:

R current value of function

T current local time

AMP previous maximum positive value of function

AMN previous maximum negative value of function

TMP global time corresponding to AMP
TMN = global time corresponding to AMN
TT = global time at t=0

Procedure:

1. If $R \leq AMP$ go to step 4

2. Revise positive maximum and corresponding time:
   set $AMP = R$
   $TMP = T + TT$

3. Return

4. If $R > AMN$ return

5. Revise negative maximum and corresponding time:
   set $AMN = R$
   $TMN = T + TT$

6. Return

OUTPUT

OUTPUT normalizes and writes the results of a particular case.

Arguments are:

- QMP = maximum positive value of $q$
- QMN = maximum negative value of $q$
- XMP = maximum positive value of $x$
- XMN = maximum negative value of $x$
- UMPDD = maximum positive value of $\ddot{u}$
- UMNDD = maximum negative value of $\ddot{u}$
- VMPDD = maximum positive value of $\ddot{v}$
- VMNDD = maximum negative value of $\ddot{v}$
- YM = maximum ground displacement $y_m$
- YMD = maximum ground displacement $\dot{y}_m$
YMDD  maximum ground acceleration $\ddot{y}_m$

P2  non-sliding undamped circular natural frequency $p_2$

TREF  reference time for accelerogram

Other variables are:

F2TREF  dimensionless frequency parameter $f_2^{\text{ref}}$

IDIAG  scratch variable

IFINE  printout -- no convergence diagnostic 'bbbb'

IFLAG  convergence flag -- common to SOLVET, SOLVEE

IPUKE  printout -- convergence diagnostic 'CONV'

PQ  normalized pseudovelocity $p_2 q_m / \ddot{y}_m$

PX  normalized displacement parameter $p_2 x_m / \ddot{y}_m$

QM  $q_m$, maximum absolute value of relative displacement q

QNOR  normalized $q_m = q_m / \ddot{y}_m$

UNORDD  normalized $u = u_m / \ddot{y}_m$

VNORDD  normalized $v = v_m / \ddot{y}_m$

XM  $x_m$, maximum absolute value of sliding displacement x

XNOR  normalized $x_m = x_m / \ddot{y}_m$

Procedure:

1. If IFLAG = 1, convergence diagnostic must be printed, go to step 4

2. Set IDIAG = IFINE for no convergence diagnostic

3. Go to step 5

4. Set IDIAG = IPUKE for convergence diagnostic

5. Calculate parameters and normalized maxima

6. Print maxima and convergence diagnostic, if any

7. Reset IFLAG=0 for next case

8. Return
PARAM

PARAM initially reads an accelerogram and nondimensional parameters for a given set of cases, and subsequently calculates required frequencies, masses, and damping factors for each case.

Arguments are:

BETA(2) array of damping factors $\beta_1, \beta_2$
BP(2) array of damping coefficients $\beta_1 p_1, \beta_2 p_2$
P(2) array of undamped circular natural frequencies $p_1, p_2$
W(2) array of damped circular natural frequencies $\omega_1, \omega_2$
AM mass constant $m = m_1 m_2 / (m_1 + m_2)$
AMA mass $m_1$
AMB mass $m_2$
AMC total mass $M = m_1 + m_2$
FRIC friction force
YMDD maximum ground acceleration $\ddot{y}_m$
YMD maximum ground velocity $\dot{y}_m$
YM maximum ground displacement $y_m$
TREF reference time for a given accelerogram

Other variables are:

BA scratch variable
BB scratch variable
BLIST(20) list of $\beta_2$ values
F2TREF dimensionless frequency parameter $f_2 t_{ref}$
FF dimensionless frequency parameter $FF = F / (M \ddot{y}_m)$
FFLIST(100) list of FF values
FTLIST(100) list of $f_2 t_{ref}$ values
I         scratch index
IB        index for BLIST
IF        index for FFLIST
IR        index for RLIST (routine assigns initial value 0)
IT        index for FTLIST
NB        length of array BLIST
NF        length of array FFLIST
NR        length of array RLIST
NT        length of array FTLIST
RHO       dimensionless frequency parameter $\rho = p_2/p_1$
RLIST(20) list of $\rho$ values

Procedure:
1. If IR$\neq 0$ go to step 5
2. Call GRNDRD to read accelerogram
3. Read lists of parameters
4. Set IR=1
5. Compile next set of dimensionless parameters from lists
6. Heading printout
7. Increment appropriate index or indices
8. If last case reset IR to 0
9. Calculate required parameters and coefficients
10. If system overdamped, go to step 12
11. Return
12. Write diagnostic
13. Go to step 5
QKSLP examines derivatives of

\[ f = e^{-\beta t} (A \sin \omega t + B \cos \omega t) + C + Dt \]

at \( t=0 \) on order to determine whether or not a system on the threshold of sliding will begin to slide (i.e., when \(|s| = F\) at \( t=0 \)).

Arguments are:

- A
- B coefficients above
- D
- BP damping constant \( \beta p \)
- W damped circular natural frequency \( \omega \)
- K indicator for immediate sliding to begin

Other variables are:

- G coefficient of \((\cos \omega t)\) term in first derivative of \( f \)
- H coefficient of \((\sin \omega t)\) term in first derivative of \( f \)
- S coefficient of \((\cos \omega t)\) term in second derivative of \( f \)

Procedure:

1. Set \( K = 1 \) (0020)
2. Calculate \( H \) to examine first derivative (0030)
3. If \( H+D < 0 \) go to step 7
   - \( H+D = 0 \) go to step 4
   - \( H+D > 0 \) go to step 6 (0040)
4. Calculate \( G \) and \( S \) (0050-0060)
5. Examine second derivative; if \( S > 0 \) go to step 6, otherwise return (0070-0080)
6. Set K=0 (immediate sliding)                           (0090)
7. Return                                               (0100)

QUADR

QUADR calculates the smallest real positive solution of a quadratic
equation

\[ Ax^2 + Bx + C = 0 \]

If no such solution exists, a flag of \(10^{10}\) is returned as a root.

Arguments are:

A
B coefficients above
C
T root

Other variables are:

BIG \(10^{10}\), a flag
DIS discriminant
R scratch variable
S square root of discriminant

No procedure is given for this routine.

SLPCK

SLPCK calculates time and direction of sliding by solving

\[ e^{-\beta t} ( A \sin \omega t + B \cos \omega t ) + C + Dt = \pm R \]
on the interval $0 \leq t < T_F$. If sliding does not occur on that interval, a flag time of $10^{10}$ is returned.

Arguments are:

A
B
C
D
BP  damping constant $\beta p$
W  damped circular natural frequency $\omega$
TF  end of time interval
EPS  tolerance for Newton's Method
R  critical force level for sliding
S  $\text{sig } \dot{x}$ at beginning of sliding, or 0 if no sliding occurs
T  time to begin sliding

Other variables are:

K  indicator for immediate sliding (at $t=0$)
SUM  value of function at $t=0$

Procedure:

1. Calculate SUM  

2. If SUM $\neq R$ go to step 7  

3. Call QKSLP to determine whether sliding begins immediately  

4. If $K \neq 0$, go to step 7  

5. Sliding at $T=0$, $S= -1$  

6. Return  

7. If SUM $\neq -R$ go to step 12
8. Call QKSLP to determine whether sliding begins immediately

9. If K ≠ 0, go to step 12

10. Sliding at T=0, S=1

11. Return

12. Call SLPFND to determine whether system slides on the open interval (0,TF)

13. Return

SLPFND

SLPFND calculates time and direction of sliding by solving

\[ f = e^{-\beta t} (A \sin \omega t + B \cos \omega t) + C + Dt = \pm R \]

for the smallest root, if any, on the open interval (0,TF). If sliding does not occur, a flag time of 10^{10} is returned.

Arguments are:

A
B
coefficients above
C
D
BP damping constant \( \beta p \)
W damped circular natural frequency \( \omega \)
TF end of time interval
EPS tolerance for Newton's Method
R critical force for sliding
S \( \dot{x} \) at beginning of sliding, or 0 if no sliding occurs
T time to begin sliding

Other variables are:

BIG $10^{10}$, a flag
FA value of $f$ at $t=TA$
FB value of $f$ at $t=TB$
FO value of $f + R \cdot \text{sig } D$ at $t=0$
G coefficient of $(\sin \omega t)$ in derivative of $f$
H coefficient of $(\cos \omega t)$ in derivative of $f$
TA time of first extremum of $f$
TB time of second extremum of $f$

Procedure:

1. Check for simple cases
2. If $D \neq 0$, go to step 4
3. Set $S=0$, go to step 5
4. Set $S = -\text{sig } D$
5. Calculate G,H
6. Use TIMEC to obtain $TA$
7. If $TA=\text{BIG}$, $f$ is monotonic, go to step 25
8. Calculate $FA$
9. If $FA \leq R$, go to step 11
10. Set $S=-1$, go to step 13
11. If $FA \geq R$, go to step 14
12. Set $S = 1$
13. Root exists on $(0,TA)$; use SOLVET at initial time $TA/2$ to find root $T$, go to step 28
14. If $TA > TF$, no root exists; go to step 29

(0030-0110) (0112) (0114-0116) (0120) (0130-0140) (0150) (0160) (0170) (0180) (0190-0200) (0210) (0220) (0230) (0250)
15. Use TIMEC to obtain TB
16. If TB=BIG, f is monotonic after first extremum; go
to step 25
17. Calculate FB
18. If FB < R, go to step 20
19. Set S = -1, go to step 22
20. If FB ≥ R, go to step 23
21. Set S = 1
22. Root exists on (TA,TB); use SOLVET at initial time
(TA+TB)/2 to obtain root T, go to step 28
23. If D=0, no root exists; go to step 29
24. If TB > TF, go to step 29; no root exists
25. Calculate FO
26. If FO=0, set FO=FA; TANPT requires nonzero FO
27. Call TANPT to obtain root T, if any
28. If T < TF, return
29. Set T=BIG, set S=0, return

FUNCTION SOLVEE

SOLVEE uses Newton's Method to solve

\[ f = e^{-\beta t}( A \sin \omega t + B \cos \omega t ) + C + Dt + Et^2 = 0 \]

given an initial trial for which the Newton's Method convergence criteria are satisfied. The derivative of the function f is of the form

\[ f' = e^{-\beta t}( G \sin \omega t + H \cos \omega t ) + D + 2Et \]
Computation ceases when the change in the root $T$ is less than a tolerance times the last value of $T$. However, this precision may not be attainable if, for example, $B$ is roughly equal to $C$. In such cases, computation ceases after 15 cycles, and an indicator IFLAG is set equal to 1.

Arguments are:

A
B
C coefficients of $f$, above
D
E
BP damping coefficient $\beta_p$
W damped circular natural frequency $\omega$
TO initial trial solution
EPS tolerance for Newton's Method

Other variables are:

CC
DT scratch variables
EE
G coefficients in $f'$, above
H
I scratch counter
IFLAG convergence flag, common to SOLVET, OUTPUT
S
WT scratch variables
T
Procedure:
1. Set \( I = 0 \)
2. Calculate \( G, H \)
3. Set \( T = T_0 \)
4. Increment \( I \) by 1
5. Calculate \( DT = f(T) / f'(T) \)
6. New \( T = old \ T - DT \)
7. If \( I = 15 \) go to step 11
8. If \( |DT| > |T| \times EPS \) go to step 4
9. Set \( SOLVEE = T \)
10. Return
11. Set \( SOLVEE = T \)
12. Set \( IFLAG = 1 \)
13. Return

FUNCTION SOLVET

SOLVET uses Newton's Method to solve

\[
f = e^{-\beta t}(A \sin \omega t + B \cos \omega t) + C + Dt = 0
\]

given an initial trial value for which the Newton's Method convergence criteria are satisfied. The derivative of this function is of the form

\[
f' = e^{-\beta t}(G \sin \omega t + H \cos \omega t) + D
\]

The procedure is the same as that for SOLVEE.
Arguments are:
A
B
coefficients of \( f \), above
C
D
BP
damping coefficient \( \beta_p \)
W
damped circular natural frequency \( \omega \)
TO
initial trial solution
EPS
tolerance for Newton's Method

Other variables are:
CC
DT
scratch variables
E
G
coefficients of \( f' \), above
H
I
scratch counter
IFLAG
convergence flag, common to SOLVEE, OUTPUT
S
WT
scratch variables
T

**TANMON**

**TANMON** solves

\[
f = e^{-\beta_p t}( A \sin \omega t + B \cos \omega t ) + C + Dt + Et^2 = 0
\]
on an interval \((T_0, T_F)\) on which \(f\) is monotonic by searching for sign changes at points where \(f\) is tangent to its decay envelope. (See 2.4.5)

Arguments are:

- \(A\)
- \(B\)
- \(C\) coefficients above
- \(D\)
- \(E\)
- \(BP\) damping coefficient \(\beta_p\)
- \(W\) damped circular natural frequency \(\omega\)
- \(T_0\) beginning of time interval
- \(F_0\) value of \(f\) at \(t=T_0\)
- \(T_F\) interval end time
- \(F_F\) value of \(f\) at \(t=T_F\)
- \(EPS\) tolerance for Newton's Method
- \(T\) solution

Other variables are:

- \(EE\) \(e^{-2\pi\beta_p/\omega}\)
- \(FA\) value of \(f\) at \(t=TA\)
- \(FB\) value of \(f\) at \(t=TB\)
- \(HP\) \(\pi/2\)
- \(P\) \(\pi\)
- \(PP\) \(e^{-\beta_p t}(A \sin \omega t + B \cos \omega t)\) at \(t=TA\)
- \(TA\) tangent point on axis side
- \(TB\) tangent point on opposite side from axis
- \(TP\) \(2\pi\)
WT scratch variable

Procedure:
1. Obtain first tangent point \( TA > T_0 \)  
2. Calculate first \( TB > T_0 \), and \( FB \)  
3. If \( TA < TF \) go to step 12  
4. If \( TB > T_0 \) go to step 6  
5. Use SOLVEE at initial time \((T_0 + TF)/2\) to compute root \( T \), return  
6. If \( TB > TF \) go to step 11  
7. If sign change on \((T_0, TB)\) go to step 9  
8. Use SOLVEE at initial time \((TB + TF)/2\) to compute root \( T \), return  
9. If \( FB = 0 \) go to step 44  
10. Go to step 42  
11. Use SOLVEE at initial time \( TF \) to compute root \( T \), return  
12. Compute PP, EE, FA  
13. If no sign change on \((T_0, TA)\) go to step 20  
14. If \( FA = 0 \) go to step 43  
15. If \( TB > T_0 \) go to step 17  
16. Use SOLVEE at initial time \((TO + TA)/2\) to compute root \( T \), return  
17. If \( FB = 0 \) go to step 44  
18. If sign change on \((TO, TB)\) go to step 42  
19. Go to step 30  
20. Increment \( TA \) by \( 2\pi/\omega \)  
21. Compute new PP, FA  
22. If sign change on \((TO, TA)\) go to step 25
23. If $TA > TF$ stop (protection against infinite loop) \hspace{1cm} (0480)
24. Go to step 20 \hspace{1cm} (0490)
25. If $FA = 0$ go to step 43 \hspace{1cm} (0500)
26. Decrease $TA$ by $2\pi/\omega$ \hspace{1cm} (0510)
27. Calculate new $TB > TA$, and $FB$ \hspace{1cm} (0520-0530)
28. If $FB = 0$, go to step 44 \hspace{1cm} (0540)
29. If sign change on $(T0, TB)$ go to step 40 \hspace{1cm} (0550)
30. Use SOLVEE at initial time $(TB + \pi/2\omega)$ to compute root $T$, return \hspace{1cm} (0560-0570)
31. Calculate new $FA$ \hspace{1cm} (0575)
32. If $|FA| < |FB|$ set $TB = TA$ \hspace{1cm} (0580)
33. Use SOLVEE at initial time $TB$ to compute root $T$, return \hspace{1cm} (0590-0600)
34. Set $T = TA$, return \hspace{1cm} (0610-0620)
35. Set $T = TB$, return \hspace{1cm} (0630-0640)

**TANPT**

TANPT obtains the smallest positive solution, if any, of

$$f = e^{-\beta t} \left( A \sin \omega t + B \cos \omega t \right) + C + Dt = 0$$

on an interval $(T0, TF)$ by searching for sign changes at points where $f$ is tangent to its decay envelope. If no solution exists, a flag time of $10^{10}$ is returned. See (2.4.5).

Arguments are:

$A$ \hspace{1cm} coefficient of $f$, above
B coefficients of \( f \), above

D damping coefficient \( \beta_p \)

BP damped circular natural frequency \( \omega \)

TO beginning of time interval

FO nonzero initial value of \( f \) supplied by calling program to initialize sign change search

TF interval end time

EPS tolerance for Newton's Method

T solution

Other variables are:

BIG \( 10^{10} \), a flag

E \( e^{-2\pi\beta_p/\omega} \)

G coefficient of \( \sin \omega t \) term in first derivative of \( f \)

H coefficient of \( \cos \omega t \) term in first derivative of \( f \)

FA value of \( f \) at \( t=TA \)

FB value of \( f \) at \( t=TB \)

FC value of \( f \) at \( t=TC \)

HP \( \pi/2 \)

P \( \pi \)

PP \( e^{-\beta_p t}( A \sin \omega t + B \cos \omega t ) \) at \( t=TA \)

TA tangent point on axis side

TB tangent point on opposite side from axis

TC time of first extremum of \( f \), if any, >TA

TP \( 2\pi \)

WT scratch variable
Procedure:
1. Obtain first tangent point $TA > T0$ (0030-0110)
2. Calculate $PP, E$ (0115-0120)
3. Calculate $FA$ (0180)
4. If sign change on $(T0, TA)$ go to step 9 (0190)
5. If $TA > TF$ go to step 8 (no solution) (0200)
6. Increment $TA$ by $2\pi/\omega$, calculate new $PP$ (0210-0220)
7. Go to step 3 (0230)
8. Set $T = BIG$, return (0240-0250)
9. Root exists on $(T0, TA)$; if $FA \neq 0$ go to step 11 (0260)
10. Set $T = TA$, return (0270-0280)
11. Decrease $TA$ by $2\pi/\omega$ (0290)
12. If $TA < T0$ go to step 21 (0292)
13. Calculate $G, H$ (0294-0296)
14. Use TIMEC to compute next extremum time $TC$ greater than $TA$, if any (0300)
15. If $TC = BIG$, go to step 26 (0310)
16. Calculate $FC$ (0320)
17. If no sign change on $(T0, TC)$ go to step 19 (0330)
18. Use SOLVET at initial time $TA$ to compute root $T$, return (0340-0350)
19. If $FC \neq 0$ go to step 26 (0360)
20. Set $T = TC$, return (0370-0380)
21. Set $TB = TA + \pi/\omega$ (0382)
22. Calculate $FB$ (0383)
23. If no sign change on $(T0, TB)$ go to step 35 (0384)
24. If $TB < T0$ go to step 35 (0385)
25. Go to step 34 (0388)
26. Set $TB + TA + \pi/\omega$  \(0390\)

27. Calculate $FB$  \(0400\)

28. If no sign change on $(T_0,T_B)$ go to step 35  \(0410\)

29. If $FB \neq 0$ go to step 31  \(0440\)

30. Set $T = TB$, return  \(0450-0460\)

31. If $TC \neq BIG$ go to step 34  \(0464\)

32. Calculate $FA$  \(0470\)

33. If $|FA| < |FB|$ set $TB = TA$  \(0480\)

34. Use SOLVET at initial time $TB$ to compute root $T$, return  \(0490-0500\)

35. Use SOLVET at initial time $(TB + \pi/2\omega)$ to compute root $T$, return  \(0510-0520\)

**TANPTE**

TANPTE obtains the smallest positive solution, if any of

$$f = e^{-\beta t}\left( A \sin \omega t + B \cos \omega t \right) + C + Dt + Et^2 = 0$$

on the open interval $(0,TF)$ by searching for sign changes at points where $f$ is tangent to its decay envelope. If no solution exists, a flag time of $10^{10}$ is returned. See (2.4.5).
G  coefficient of $(\sin \omega t)$ term in first derivative of $f$
H  coefficient of $(\cos \omega t)$ term in first derivative of $f$
BP  damping coefficient $\beta p$
W  damped circular natural frequency $\omega$
TF  interval end time
EPS  tolerance for Newton's Method
FO  nonzero initial value of $f$ supplied by calling program to initialize sign change search
T  solution

Other variables are:
AMP  initial amplitude of sine wave $\sqrt{A^2+B^2}$
BIG  $10^{10}$
EE  $e^{-2\pi \beta p/\omega}$
FA  value of $f$ at $t=TA$
FB  value of $f$ at $t=TB$
FC  value of $f$ at $t=TC$
FV  value of $A \sin \omega t + B \cos \omega t + C + Dt + Et^2$ at $t=TLIM$
HP  $\pi/2$
P  $\pi$
PP  $e^{-\beta pt}( A \sin \omega t + B \cos \omega t)$ at $t=TA$
S  sig FO
TA  tangent point on axis side
TB  tangent point on opposite side from axis
TC  time of first extremum of $f > TA$
TLIM  initially, time corresponding to vertex of parabola $C + Dt + Et^2$; subsequently, time beyond which search will not continue
TP  $2\pi$
WT scratch variable

Procedure:
1. Calculate vertex time TLIM
2. If TLIM < 0, go to step 8
3. If F0 x E < 0 go to step 9
4. If TLIM > TF go to step 9
5. Calculate AMP, S, FV
6. If FV x C > 0, no root on (0, TLIM); go to step 39
7. Go to step 10
8. If FO x E > 0, no root on (0, TF); go to step 39
9. Set TLIM = TF
10. Obtain first tangent point TA
11. Calculate PP, EE
12. Calculate FA
13. If sign change on (0, TA), go to step 17
14. If TA ≥ TLIM go to step 19
15. Increment TA by 2π/ω, calculate new PP
16. Go to step 12
17. If FA ≠ 0 go to step 20
18. Set T = TA, return
19. If TLIM = TF, no root, go to step 39
20. Decrease TA by 2π/ω
21. Use TIMED to compute TC
22. If TC = BIG, go to step 28
23. Calculate FC
24. If sign change on (0, TC), go to step 26
25. Use SOLVEE at initial time TA to compute root T, return

26. If FC≠0 go to step 28

27. Set T=TC, return

28. If sign change on (0,TA), go to step 39

29. Set TB = TA + π/ω

30. Calculate FB

31. If FB≠0 go to step 33

32. Set T = TB , return

33. If sign change on (0,TB) go to step 38

34. If TC≠BIG go to step 37

35. Calculate FA

36. If |FA| < |FB| set TB = TA

37. Use SOLVEE at initial time TB to compute root T return

38. Use SOLVEE at initial time TB + π/2ω to compute root T, return

39. Set T=BIG, return

FUNCTION TIMEB

TIMEB furnishes the smallest positive real solution of

\[ f = e^{pt}( A \sin \omega t + B \cos \omega t ) = 0 \]

Arguments are:

A

coefficients of f, above

B
W \quad \text{damped circular natural frequency } \omega

Other variables are:

HP \quad \frac{\pi}{2}

P \quad \pi

WT \quad \text{scratch variable}

Procedure:

1. If $A \neq 0$ go to step 4

2. Set $\text{TIMEB} = \frac{\text{HP}}{W}$

3. Return

4. Set $WT = \arctan(-B/A)$

5. If $WT \leq 0$, set $WT = WT + P$

6. Set $\text{TIMEB} = \frac{WT}{W}$

7. Return

FUNCTION TIMEC

TIMEC furnishes the smallest positive real solution, if any, of

$$f = e^{-\beta t} \left( A \sin \omega t + B \cos \omega t \right) + C = 0$$

such that the root is a lower limit $T_0$. If no such solution is found, a flag time of $10^{10}$ is returned. The derivative of $f$ is of the form

$$f' = e^{-\beta t} \left( G \sin \omega t + H \cos \omega t \right)$$
Arguments are:

A

B coefficients of f, above

C

BP damping coefficient βp

W damped circular natural frequency ω

TO lower limit on solution

EPS tolerance for Newton's Method

Other variables are:

BIG $10^{10}$, a flag

FA value of f at t=TA

FB value of f at t=TB

FO value of f at t=TO

G coefficients of f', above

H

P

T scratch variable

TA time of first extremum of f > TO

TB time of second extremum of f > TO

Procedure:

1. If A=0 and B=0 go to step 24

2. If C≠0 go to step 6

3. Set T = TIMEB (A,B,W)

4. Increment T by $\pi/\omega$ until T > TO

5. Go to step 25

6. Calculate G,H
7. Use TIMEB to calculate $\text{TA}$

8. Increment $\text{TA}$ by $\pi/\omega$ until $\text{TA} > \text{TO}$

9. Calculate $\text{FA}$

10. If $\text{FA} \neq 0$ go to step 13

11. Set $T = \text{TA}$

12. Go to step 25

13. Calculate $\text{FO}$

14. If $\text{FO}$ is very near zero, set $\text{FO} = 0$

15. If no sign change on $(\text{TO}, \text{TA})$ go to step 4

16. Use SOLVET at $(\text{TA}+\text{TO})/2$ to obtain root, go to step 25

17. Calculate $\text{TB}$

18. Calculate $\text{FB}$

19. If $\text{FB} \neq 0$ go to step 22

20. Set $T = \text{TB}$

21. Go to step 25

22. If no sign change of $(\text{TA}, \text{TB})$ go to step 24

23. Use SOLVET at initial time $(\text{TA}+\text{TB})/2$, go to step 25

24. Set $T = \text{BIG}$ (no solution)

25. Set $\text{TIMEC} = T$

26. Return

**FUNCTION TIMED**

TIMED solves

$$f = e^{-\beta pt}( A \sin \omega t + B \cos \omega t ) + C + D t = 0$$
for the smallest root, if any, on the time interval (T0, TF). If no such solution is found, a flag time of \(10^{10}\) is returned.

Arguments are:

A
B
C coefficients in \(f\), above
D
BP damping factor \(\beta_p\)
W damped circular frequency \(\omega\)
TO beginning of time interval
TF interval end time
EPS tolerance for Newton's Method

Other variables are:

BIG \(10^{10}\), a flag
FA value of \(f\) at \(t=TA\)
FB value of \(f\) at \(t=TB\)
FO value of \(f\) at \(t=TO\)
G coefficient of \((\sin \omega t)\) term in first derivative of \(f\)
H coefficient of \((\cos \omega t)\) term in first derivative of \(f\)
HP \(\pi/2\)
P \(\pi\)
T scratch variable
TA time of first extremum of \(f > TO\)
TB time of second extremum of \(f > TO\)
TP \(2\pi\)

Procedure:
1. Check for simple cases (0030-0110)
2. Calculate G,H (0120-0130)
3. Use TIMEC to obtain TA (0140)
4. Calculate FO (0150)
5. If FO is very near zero, set FO = 0 (0160)
6. If TA = BIG, function is monotonic; go to step 20 (0170)
7. Calculate FA (0180)
8. If no sign change of (0,TA), go to step 10 (0190)
9. Use SOLVET at (TA+TO)/2 initial time; go to step 22 (0200-0210)
10. If TA > TF go to step 23 (no solution) (0220)
11. If FA = 0 go to step 12; otherwise go to step 22 (0230-0250)
12. Use TIMEC to obtain TB (0260)
13. If TB ≠ BIG, go to step 15
14. Function is monotonic after first extremum; if
    FA x D > 0, go to step 23; Otherwise go to step 21 (0280-0290)
15. Calculate FB (0300)
16. If no sign change on (TA,TB) go to step 18 (0310)
17. Use SOLVET at initial time (TA+TB)/2 to obtain root,
    go to step 22 (0320-0330)
18. If TB > TF go to step 23 (no solution) (0340)
19. If FB ≠ 0, go to step 20; otherwise go to step 22 (0350-0370)
20. If FO x D ≥ 0, go to step 23 (no solution) (0380)
21. Call TANPT to search for sign change at points where
    f is tangent to its decay envelope (0400)
22. If root < TF, go to step 34 (0400)
23. Root = BIG (0410)
24. Set TIMED = root (0420)
25. Return (0430)
FUNCTION TIMEE
TIMEE solves

\[ f = e^{-\beta t} \left( A \sin \omega t + B \cos \omega t \right) + C + Dt + Et^2 = 0 \]

for the smallest positive real root, if any, on the open interval \((0, TF)\).
If \(TF = 10^{10}\), \(E\) must be zero for the routine to function. If no such
solution is found, a flag time of \(10^{10}\) is returned.

Arguments are:

A  
B  
C  coefficients of \(f\), above  
D  
E  
BP  damping constant \(\beta_p\)  
W  damped circular natural frequency \(\omega\)  
FO  value of \(f\) at \(t=0\)  
TF  interval end time  
EPS  tolerance for Newton's Method  

Other variables are

BIG  \(10^{10}\), a flag  
FA  value of \(f\) at \(t=TA\)  
FB  value of \(f\) at \(t=TB\)  
FF  value of \(f\) at \(t=TF\)  
G  coefficient of \((\sin \omega t)\) term in first derivative of \(f\)  
H  coefficient of \((\cos \omega t)\) term in first derivative of \(f\)
HP        \pi/2
P          \pi
T          scratch variable
TA         time of first extremum of f
TB         time of second extremum of f
TP         2\pi

Procedure:
1. Check for simple cases; solve by calling QUADR or TIMED; go to step 29      (0050-0100)
2. Calculate G,H                                                               (0110-0120)
3. Use TIMED on interval (0,TF+2\pi/\omega) to compute TA                     (0140)
4. If TA\neq BIG go to step 8                                                 (0190)
5. f is monotonic on (0,TF); calculate FF                                    (0200)
6. If no sign change on (0,TF), no solution; go to step 30                  (0210)
7. Call TANMON on interval (0,TF) to obtain root T, go to step 29          (0220-0225)
8. Calculate FA                                                              (0230)
9. If no sign change on (0,TA) go to step 11                                 (0240)
10. f is monotonic on (0,TA); if TA > \pi/\omega go to step 12                (0242)
11. Use SOLVEE at initial time TA/2 to compute root T; go to step 29        (0244-0246)
12. Call TANMON on interval (0,TA) to obtain root T; go to step 29           (0250-0260)
13. If TA > TF go to step 30 (no solution)                                    (0270)
14. If FA\neq 0 go to step 16                                                  (0280)
15. Set T = TA, go to step 31                                                 (0290-0300)
16. Use TIMED on interval (TA,TF+2\pi/\omega) to compute TB                  (0310)
17. If TB\neq BIG go to step 21                                               (0340)
18. f is monotonic on (TB,TF); calculate FF
19. If no sign change on (TA,TF), no solution; go to step 30
20. Call TANMON on interval (TA,TF) to obtain root T, go to step 29
21. Calculate FB
22. If no sign change on (TA,TB) go to step 24
23. Use SOLVEE at initial time (TA+TB)/2 to obtain root T, go to step 29
24. If TB > TF, no solution; go to step 30
25. If FB ≠ 0 go to step 27
26. Set T=TB, go to step 27
27. Set F0=FA so that F0 is nonzero for TANPTE
28. Call TANPTE to search for sign changes at points where f is tangent to its decay envelope to obtain root T
29. If T > TF go to step 31
30. Set T=BIG
31. Set TIMEE = T
32. Return
APPENDIX B

PROGRAM LISTING
DIMENSION BETA(2),P(2),W(2),A(12),B(12)
C(12),D(12),BP(2),CO(32),EX(3)
DATA BIG/1.E10/
READ (5,102) M, EPS, TREF, DT, TEND
10 CALL PARAM (BETA, BP, P, W, A, AM, AMB, ANC, FRIC, YMDD, YMD, YM, TREF)
CALL CONST (CO, BETA, P, W, A, AM, AMB)
THIS=DT
TT=0.
QMP=0.
QMPD=0.
QMND=0.
XMN=0.
XMND=0.
UMN=0.
UMND=0.
VMN=0.
VMND=0.
TQMP=0.
TQMPD=0.
TQMPDD=0.
TXMP=0.
TXMPD=0.
TXMPDD=0.
TUMP=0.
TUMPDD=0.
TVMP=0.
TVMPDD=0.
TQMN=0.
TQMNDD=0.
TXMN=0.
TXMND=0.
UMB=0.
UMB=0.
VMPD=0.
VMPDD=0.
XMPD=0.
XMPDD=0.

25 CALL GROUND (TF,AINTBL)
IF (M.EQ.1) WRITE (6,101) TF, AINTBL
26 IF (1.EQ.1) GO TO 30
CALL COEFQ (A, B, C, D, CO(16), G0,G0D, AINTB)
GO TO 35
30 IF (XOD+S.GE.0.) GO TO 32
IF (M.EQ.1) WRITE (6,103) TT
GO TO 55
32 CALL COEFQ (A, B, C, D, CO, QQ, QQD, FRIC*S/AMA, 0.)
35 CALL COEXUV (A, B, C, D, EX, CO, QQ, QQD, XOD, XO, XCO, FRIC*S, AINTB)
IF (1.EQ.2) GO TO 40
TB=TIMEE + (A(6), B(6), C(6), D(6), EX(1), BP(1), W(1), XDD, TF, EPS)
IF (TB.EQ.BIG) GO TO 37
TB=TB+TT
IF (M.EQ.1) WRITE (6,103) TBG
37  TLIM=AMIN(TB,TF)
   XDDD=B(7)+C(7)
   UDDD=B(9)+C(9)
   VDDD=B(11)+C(11)
   CALL ONEMAX (XDDD,0.,XMPDD,XMMDD,XTMPDD,XTXMND,TT)
   CALL ONEMAX (UDDD,0.,UMPDD,UMNDD,TUMPPD,TUMNDD,TT)
   CALL ONEMAX (VDDD,0.,VMPPD,VMNDD,TVMPDD,TVXMNDD,TT)
IF (M.EQ.1) CALL HIST (THIS,TT,DT,TLIM,TEND,A,B,C,D,EX,EP,TW,I)
   CALL GETMAX (A(7),B(7),C(7),D(7),0.,BP(I),1.,W(I),1.,TLIM,EP,E,
   UXMND,XTXMPD,XTXMND,TT)
   CALL GETMAX (A(6),B(6),C(6),D(6),EX(1),BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMNP,UXMN,TXMP,TXMN,TT)
   CALL GETMAX (A(5),B(5),C(5),D(5),BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UTMPDD,TXMNNDD,TT)
   CALL GETMAX (A(4),B(4),C(4),D(4),BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UTMPDD,TXMNNDD,TT)
   CALL GETMAX (A(3),B(3),C(3),D(3),BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UTMPDD,TXMNNDD,TT)
   CALL GETMAX (A(2),B(2),C(2),D(2),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UTMPDD,TXMNNDD,TT)
   CALL GETMAX (A(1),B(1),C(1),D(1),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UTMPDD,TXMNNDD,TT)
   CALL GETMAX (X0,TLIM,XMP,XMN,XTMP,XTMN,TT)
IF (TLIM.EQ.TB) S=0
   GO TO 45
40  FORCE=AMC+AIND+AMB*B(3)
   IF (FORCE.GT.FRIC) GO TO 48
   IF (FORCE.LT.FRIC) GO TO 50
   CALL SLPCK (A(3)*AMB,B(3)*AMB,AIND,AMC*BSL,BP(2),W(2),TF)
   EPS=FRIC*S(TB)
   IF (TB.EQ.BIG) GO TO 42
   TBG=TB+TT
   IF (M.EQ.1) WRITE (6,104) TBG,S
   GO TO 45
42  TLIM=AMIN(TB,TF)
IF (M.EQ.1) CALL HIST (THIS,TT,DT,TLIM,TEND,A,B,C,D,EX,EP,TW,I)
   CALL GETMAX (A(1),B(1),C(1),D(1),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL GETMAX (A(2),B(2),C(2),D(2),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL GETMAX (A(3),B(3),C(3),D(3),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL GETMAX (A(4),B(4),C(4),D(4),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL GETMAX (A(5),B(5),C(5),D(5),BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL GETMAX (A(6),B(6),C(6),D(6),EX(1),BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL GETMAX (A(7),B(7),C(7),D(7),0.,BP(1),1.,W(1),1.,TLIM,EP,E,
   UXMND,UXMN,UXMPD,UXXMN,TT)
   CALL HIST (THIS,TT,DT,TLIM,TEND,A,B,C,D,EX,EP,TW,I)
   IF (TF.EQ.BIG) GO TO 60
   TBG=TB+TT
   IF (M.EQ.1) WRITE (6,105) TT,QQ,QQD,XO,XOD
   GO TO 37
46  QQ=F(A(1),B(1),C(1),D(1),BP(1),1.,WTI)
   QQD=F(A(2),B(2),C(2),D(2),BP(1),1.,WTI)
   GO TO 45
IF (M.EQ.1) WRITE (6,105) TT,QQ,QQD,XO,XOD
48  S=M+1
   GO TO 55
50  S=1
55  I=J=1
   XOD=0
   IF (J.EQ.2) GO TO 60
   IF (TF.EQ.BIG) J=J+1
   GO TO 26
60  CALL OUTPUT (QMP,QMN,XMP,XMN,UMPDD,UMNDD,VMPPD,VMNDD,YM,YMD)
   GO TO 10
101 FORMAT ('OTF*E15.6', 'A*E15.6', 'B*E15.6')
SUBROUTINE AGRAM(YM,YMD,YMDD,ACC,T,N)
DIMENSION T(11),ACC(1)
YM=0.
YMD=0.
YMDD=ABS(ACC(1))
Y0=0.
Y0D=0.
IF (T(1)=EQ.0.) GO TO 1
BB=ACC(1)/T(1)
Y0D=0.5*BB*T(1)**2
Y0=Y0D/3.*T(1)
YMDD=ABS(Y0D)
YM=ABS(Y0)
1 DO 7 I=2,N
IF (ABS(ACC(I))>YMDD) YMDD=ABS(ACC(I))
IF (T(I)=EQ.T(I-1)) GO TO 7
DT=T(I)-T(I-1)
BB=ACC(I)/DT
IF (BB=EQ.0.) GO TO 3
TT=ACC(I)/BB
IF (TT=LE.0.) GO TO 2
IF (TT=GE.DT) GO TO 2
Y0=(ACC(I)+0.5*BB*TT)*TT*Y0D
IF (ABS(Y0)>YT*YMD) YMD=ABS(Y0D)
7 CONTINUE
RETURN
END
SUBROUTINE COEFQ (A,B,C,D,CO,QD,QOD,AIN,BSL)
DIMENSION A(1),B(1),C(1),D(1),CO(1)
A(1) = CO(7)*QD + CO(10)*QD + CO(9)*AIN + CO(12)*BSL
B(1) = QD + CO(2)*AIN = CO(5)*BSL
C(1) = -CO(2)*AIN + CO(5)*BSL
D(1) = -CO(2)*BSL
A(2) = -CO(10)*QOD = CO(8)*QD = CO(7)*AIN + CO(9)*BSL
B(2) = QOD + CO(2)*BSL
C(2) = -CO(2)*BSL
D(2) = 0.
A(3) = -CO(13)*QOD + CO(11)*QD + CO(10)*AIN = CO(7)*BSL
B(3) = -CO(3)*QOD = CO(1)*QD + AIN
C(3) = 0.
D(3) = 0.
A(4) = CO(15)*QOD + CO(14)*QD + CO(13)*AIN + CO(10)*BSL
B(4) = -CO(6)*QOD + CO(4)*QD + CO(3)*AIN = BS
C(4) = 0.
D(4) = 0.*
RETURN
END

SUBROUTINE COEXUV (A,B,C,D,EX,CO,QD,QOD,XO,XOD,FS,AIN,BSL)
DIMENSION A(1),B(1),C(1),D(1),CO(1),EX(1)
IF (FS.EQ.0.) GO TO 2
RA = CO(31)
RB = CO(32)
RC = 1.*RA
A(5) = -RA*A(1)
B(5) = -RA*B(1)
C(5) = RA*(QD=CO(11)) + XO
D(5) = RA*(QD=CO(11)) + XOD
A(6) = -RA*A(2)
B(6) = -RA*B(2)
C(6) = RA*(QOD=CO(2)) + XOD
D(6) = AIN = RB*FS
A(7) = -RA*A(3)
B(7) = -RA*B(3)
C(7) = AIN = RB*FS
D(7) = -BSL
A(8) = -RA*A(4)
B(8) = -RA*B(4)
C(8) = -BSL
D(8) = 0.*
A(9) = A(7)
B(9) = B(7)
C(9) = RB*FS
D(9) = 0.*
A(10) = A(8)
B(10) = B(8)
C(10) = 0.*
D(10) = 0.*
A(11) = RC*A(3)
B(11) = RC*B(3)
C(11) = -RB*FS
D(11) = 0
A(12) = RC*A(4)
B(12) = RC*B(4)
C(12) = 0
D(12) = 0
EX(1) = +BSL*5
EX(2) = -(A1N+R8*FS)*5
EX(3) = EX(1)/3
RETURN
2 DO 3 I=5*10
A(I) = 0
B(I) = 0
3 CONTINUE
C(5) = XO
D(5) = 0
C(6) = 0
D(6) = 0
C(7) = 0
D(7) = 0
C(8) = 0
D(8) = 0
C(9) = AIN
D(9) = BSL
C(10) = BSL
D(10) = 0
A(11) = A(3)
B(11) = B(3)
C(11) = AIN
D(11) = BSL
A(12) = A(4)
B(12) = B(4)
C(12) = BSL
D(12) = 0
EX(1)=0
EX(2)=0
EX(3)=0
RETURN
END
SUBROUTINE CONST (C,B,P,W,AM,AMA,AMB)
DIMENSION C(1),B(1),P(1),W(1)
J=0
DO 1 I=1,2
C(J+1) = P(I)**2
C(J+2) = 1./C(J+1)
C(J+3) = 2.*B(I)*P(I)
C(J+4) = C(J+3)*C(J+1)
C(J+5) = C(J+3)/C(J+1)**2
C(J+6) = C(J+1)*(1.+4.*B(I)**2)
C(J+7) = 1./W(I)
C(J+8) = C(J+1)/W(I)
C(J+9) = B(I)/(P(I)*W(I))
C(J+10) = B(I)*W(I)/W(I)
C(J+11) = C(J+10)*C(J+1)
C(J+12) = (1.+2.*B(I)**2)*C(J+2)/W(I)
C(J+13) = C(J+12)*C(J+1)**2
C(J+14) = C(J+13)*C(J+1)
C(J+15) = C(J+11)*(3.+4.*B(I)**2)
1 J=15
C(31) = AM/AMA
C(32) = AM/AMA/AMB
RETURN
END

SUBROUTINE DMAX (A,B,C,D,BP,W,TF,TA,EPS,FMP,FMN,TFMP,TFMN,TT)
DIMENSION A(1),B(1),C(1),D(1)
DATA P,TP/3.141593,6.283185/
TI=TIMEB (A(3),B(3),W)
35 IF (TI>TA) GO TO 40
TI=TI+P/W
GO TO 35
40 PP=F (A(2),B(2),0..0..BP.,W,TT)
E=EXP(-TP*BP/W)
45 PP=PP*E
FI=PP+C(2)
IF (D(I)>LT.0.) GO TO 50
IF (FI<GE.0.) RETURN
GO TO 55
50 IF (FI*LE.0.) RETURN
55 TA=SOLVET (A(2),B(2),C(2),0..BP.,W,TA+TP/W,EPS)
IF (TA*GE.0.) RETURN
FA=F (A(1),B(1),C(1),D(1),BP,W,TA)
IF (D(I)>LT.0.) GO TO 60
IF (FA*LE.FMP) GO TO 45
FMP=FA
TFMP=TA+TT
GO TO 45
60 IF (FA*GE.FMN) GO TO 45
FMN=FA
TFMN=TA+TT
GO TO 45
END
SUBROUTINE EMAX (A,B,C,D,E,BP*,TP*,EPS,FMP*,FMN*,TFMP*,TFMN*,TT) EMAX0010
DIMENSION A(1),B(1),C(1),D(1) EMAX0020
DATA P,TP,HP,BIG/3.1415936,283185,1,570796,1,E10/ EMAX0030
CTC=CTC+CB(C2)*D2) EMAX0040
IF (TA*EQ*BIG) GO TO 90 EMAX0050
FA=F(A(1),B(1),C(1),D(1),BP*,WA*,TA*) EMAX0060
CALL ONEMAX (FA,TA*,FMP*,FMN*,TFMP*,TFMN*,TT) EMAX0070
TB=TB+TIMED (A(2),B(2),C(2),D(2),BP*,WA*,TA*,TF*,EPS) EMAX0080
IF (TB=EQ*BIG) GO TO 90 EMAX0090
FB=F(A(1),B(1),C(2),D(2),BP*,WA*,TB*) EMAX1000
CALL ONEMAX (FB,TA*,FMP*,FMN*,TFMP*,TFMN*,TT) EMAX1010
FAB=FA+FB EMAX1020
IF (FAB=EQ*BIG) GO TO 40 EMAX1030
IF (FB*LT*TA) TA=TB EMAX1040
GO TO 55 EMAX1050
40 IF (FAB*GT*TA) TA=TB EMAX1060
55 IF (B(2)*NE*0) GO TO 60
WT=AT
IF (A(2)*D(2)*GT*0) WT=AT+P EMAX1070
GO TO 65 EMAX1080
GO TO 66 EMAX1090
60 WT=ATAN(A(2)/B(2)) EMAX2010
IF (B(2)*D(2)*GT*0) WT=AT+P EMAX2020
IF (WT=LT*0) WT=AT+P EMAX2030
65 TC=WT/W EMAX2040
66 IF (TC*GT*TA) TC=TC+TP/W EMAX2050
GO TO 66 EMAX2060
67 IF (FB*GT*TA) TA=TB EMAX2070
TC=TC+TP/W EMAX2080
GO TO 66 EMAX2090
TC=TC+TP/W EMAX2090
TFIN=TA/E121/E EMAX3000
IF (TFIN=LT*TA) GO TO 90 EMAX3010
GO TO 69 EMAX3020
68 TFIN=TF EMAX3030
69 PP=FA(A(1),B(2),C1,0,0,0,0,TC) EMAX3040
EE=EXP(=2*TP*BP/W) EMAX3050
70 PP=PP*EE EMAX3060
TC=TC+TP/W EMAX3070
FC=FC+C(2)*D2*TC EMAX3080
TB=TA+TP/W EMAX3090
IF (FC*D2*GT*0) GO TO 80 EMAX4000
T=TA=TIMED (A(2),B(2),C(2),D(2),BP*,WA*,TB*,EPS) EMAX4010
IF (TA*EQ*TF) GO TO 90 EMAX4020
FA=F(A(1),B(1),C(1),D1,0,0,TA*) EMAX4030
CALL ONEMAX (FA,TA*,FMP*,FMN*,TFMP*,TFMN*,TT) EMAX4040
IF (TA*EQ*TF) GO TO 90 EMAX4050
GO TO 70 EMAX4060
80 T=TIMED (A(2),B(2),C(2),D(2),BP*,WA*,TB*,TFIN*,EPS) EMAX4070
IF (TA*EQ*BIG) GO TO 90 EMAX4080
FA=F(A(1),B(1),C(1),D1,0,0,TA*) EMAX4090
85 CALL ONEMAX (FA,TA*,FMP*,FMN*,TFMP*,TFMN*,TT) EMAX5000
90 IF (TF*EQ*BIG) RETURN EMAX5010
FF=F(A(1),B(1),C(1),D1,0,0,TF) EMAX5020
CALL ONEMAX (FF,TF,FMP*,FMN*,TFMP*,TFMN*,TT) EMAX5030
RETURN EMAX5040
END EMAX5040
FUNCTION F (A,B,C,D,BP,WT)
WT=WT
F = EXP( -BP*T ) * ( A*SIN( WT ) + B*COS( WT ) ) + C + D*T
RETURN
END

SUBROUTINE GET MAX ( A,B,C,D,E,WT,TF,TA,FMP,FMN,TFMP,TFMN,TT )
DATA TP,BIG/3.141593,6.283185,1.E10/
IF ( (E.EQ.0.) ) GO TO 2
IF ( (A(1).NE.0.) .OR. (B(1).NE.0.) ) GO TO 4
TV=.5*D(1)/E
IF ( (TV.LE.0.) .OR. (TV.GE.TF) ) GO TO 65
FA=F ( A(1),B(1),C(1),D(1),BP*WT*TA )
CALL ONEMAX ( FA,TA,FMP,FMN,TFMP,TFMN,TT )
GO TO 65
IF ( (D(1).EQ.0.) ) RETURN
4
CALL EMAX ( A,B,C,D,BP,WT,TF,TA,FMP,FMN,TFMP,TFMN,TT )
GO TO 65
2
TA=TIMEC ( A(2),B(2),C(2),D(2),BP,WT,TA )
IF ( (TA.GE.TF) ) GO TO 65
FB=F ( A(1),B(1),C(1),D(1),BP,TA )
CALL ONEMAX ( FB,TA,FMP,FMN,TFMP,TFMN,TT )
IF ( (TA.GE.TF) ) RETURN
IF ( (D(1).LT.0.) ) GO TO 25
FB=FB*TF
CALL ONEMAX ( FB,TF,FMP,FMN,TFMP,TFMN,TT )
GO TO 65
25
IF ( (FB.LE.0.) ) TA=TA
GO TO 30
30
CALL DMAX ( A,B,C,D,BP,WT,TF,TA,FMP,FMN,TFMP,TFMN,TT )
GO TO 65
65
IF ( (TF.EQ.BIG) ) RETURN
70
FF=FB+TF
CALL ONEMAX ( FF,TF,FMP,FMN,TFMP,TFMN,TT )
RETURN
END
SUBROUTINE GROUND (TF,A,B)
DIMENSION T(100), ACC(100), FGRM(20)
DATA I/1/
IF (I .NE. 1) GO TO 2
IF (T(1) .NE. 0.) GO TO 1
I=2
GO TO 3
1 A=0.
B=ACC(1)/T(1)
TF=T(1)
I=2
RETURN
2 IF (I*GT*N) GO TO 4
IF (T(I) .NE. T(I=I)) GO TO 3
I=I+1
IF (I .GT. N) GO TO 4
3 A=ACC(I=I)
B=(ACC(I) .EQ. ACC(I=I))/(T(I-1) .EQ. T(I=I))
TF=T(I-1)
I=I+1
RETURN
4 A=0.
B=0.
TF=I*E10
I=1
RETURN
ENTRY GRNDRO (YM, YMD, YMDD)
READ (5,10,END=99) FACT, N, ISW, YM, YMD, YMDD
10 FORMAT (F10.0,215,3F10.0)
READ (5,30) FORM
30 FORMAT (20A4)
READ (5, FORM) (T(J), ACC(J), J=1,N)
WRITE (6,40) FACT
40 FORMAT (*ACCELEROMETER FACTOR=*E15.8)
WRITE (6, FORM) (T(J), ACC(J), J=1,N)
IF (FACT .EQ. 1) GO TO 7
DO 6 J=1,N
6 ACC(J) = ACC(J)*FACT
7 IF (YM*YMD*YMDD .EQ. 0.) CALL AGRAM (YM, YMD, YMDD, ACC, T, N)
RETURN
99 STOP
END
SUBROUTINE HIST (THIS, TT, DT, TLIM, TEND, A, B, C, D, EX, BP, W, I)
DIMENSION A(1), B(1), C(1), D(1), EX(1), BP(1), W(1)
IF (THIS > TEND) RETURN
TA = THIS + TT
IF (TA > TLIM) RETURN
WRITE (6, 200)
200 FORMAT ('Q', 'TIME', 'QD', 'QD', 'QD'
!A=F (A(1) * B(1) * C(1) * D(1) * BP(I) * W(I) * TA)
!QAD=F (A(2) * B(2) * C(2) * D(2) * BP(I) * W(I) * TA)
!IF (I EQ 2) GO TO 2
!XA=F(A(5) * B(5) * C(5) * D(5) * BP(I) * W(I) * TA) + (EX(3) * TA + EX(2)) * TA + TA
!XAD=F(A(6) * B(6) * C(6) * D(6) * BP(I) * W(I) * TA) + EX(1) * TA + TA
!GO TO 3
2 XA=C(5)
XAD=0
3 WRITE (6, 100) THIS, QA, QAD, XA, XAD
100 FORMAT (' ', '5E15.6')
!THIS=THIS+DT
!TA=TA+DT
IF (THIS > TEND) RETURN
IF (TA > TLIM) RETURN
GO TO 1
END

SUBROUTINE OMAX (R, T, AMP, AMN, TMP, TMN, TT)
IF (R LE AMP) GO TO 2
AMP=R
TMP=TT
RETURN
2 IF (R GE AMN) RETURN
AMN=R
TMN=TT
RETURN
END
SUBROUTINE OUTPUT (QMP, QMN, XMP, XMN, UMPDD, UMNDD, VMPDD, VMNDD, YM, YMD)

& YMDDD* P2, TREF)

COMMON IFLAG

DATA IFINE, IPUK, E* CONV*/

IF (IFLAG = EQ, 1) GO TO 2

IDIAG = IINE

GO TO 3

2 IDIAG = IPKE

3 CONTINUE

F2TREF = P2* TREF/6.* 283185

QM = AMAX1 (QMP** 1 QMN)

XM = AMAX1 (XMP - XMN)

QNOR = QM/YM

XNOR = XM/YM

UNORDD = AMAX1 (UMPDD - UMNDD)/YMDD

VNORDD = AMAX1 (VMPDD - VMNDD)/YMDD

PQ = P2* QM/YMD

PX = P2* XM/YMD

WRITE (6*101, F2* TREF * QNOR * PQ * XNOR * PX)

10 FORMAT (*, /*F7.4, 6E13.4, 30X, A4)

IFLAG = 0

RETURN

END

SUBROUTINE PARAM (BETA, BP, P, A, AM, AMB, AMC, FRIC, YMDD, YMD, YM)

DIMENSION BETA (1), BP (1), P (1), W (1), FTLIST (100), FFLIST (100)

& BLIST (20), RLIST (20)

DATA IR, IB, IF, IT/0, 1, 1, 1/

IF (IR * NE * 0) GO TO 10

CALL GRND (YM, YMD, YMDD)

READ (5, 100) NR

READ (5, 101) (RLIST(I), I=1, NR)

READ (5, 100) NB

READ (5, 101) (BLIST(I), I=1, NB)

READ (5, 100) NF

READ (5, 101) (FFLIST(I), I=1, NF)

READ (5, 100) NT

READ (5, 101) (FFLIST(I), I=1, NT)

100 FORMAT (15)

101 FORMAT (8F10.0)

IR = 1

10 F2TREF = FFLIST (IT)

FF = FFLIST (IF)

BETA (2) = BLIST (IB)

RHO = RLIST (IR)

IF (IT = EQ, 1) WRITE (6*102) RHO, BETA (2), FF

102 FORMAT (/10X, 4RHO*, 6F10.4, 4* BETA (2) =*, F10.4, 4* FF =*, F10.4, 4*


IF (IT = EQ, NT) GO TO 20

IT = IT + 1

GO TO 50

20 IT = 1

IF (IF = EQ, NF) GO TO 30

IF = IF + 1

GO TO 50

30 IF = 1

IF (IB = EQ, NB) GO TO 40

P0.0010

P0.0020

P0.0022

P0.0023

P0.0024

P0.0025

P0.0026

P0.0027

P0.0028

P0.0030

P0.0032

P0.0034

P0.0035

P0.0036

P0.0038

P0.0040

P0.0042

P0.0044

P0.0046

P0.0048

P0.0050

P0.0052

P0.0054

P0.0056

P0.0058

P0.0060

P0.0062

P0.0064

P0.0066

P0.0067

P0.0068

P0.0069

P0.0070

P0.0072

P0.0074

P0.0076

P0.0078

P0.0080

P0.0082

P0.0084
SUBROUTINE QKSLP (A,B,D,BP,W,K)
K=1
H=BP*B+W*A
IF (H+D) 40,20,30
G=BP*A+W*B
S=BP*H+W*G
IF (S+GE0.) GO TO 30
RETURN
30 K=0
40 RETURN
END
SUBROUTINE QUADR (A*B*C*T)
DATA BIG/I.EIO/
IF (A*NE.0) GO TO 20
IF (B*EQ.0) GO TO 35
T=C/B
IF (T*EQ.0) GO TO 35
RETURN
20 R=R*S*B/A
DIS=R*R-C/A
IF (DIS*LE.0) GO TO 30
S=SQRT(DIS)
T=R+S
IF (T*GT.0) RETURN
IF (T*LE.0) GO TO 35
RETURN
30 IF (DIS*NE.0) GO TO 35
IF (R*LE.0) GO TO 35
T=R
RETURN
35 T=BIG
RETURN
END

SUBROUTINE SLPCK (A*B*C*D*BP*W*TF*EPS*R*S*T)
SUM=B+C
IF (SUM*NE.R) GO TO 5
CALL QKSLP (A*B*D*BP*W*K)
IF (K*NE.0) GO TO 5
S=1.
T=0.
RETURN
5 IF (SUM*NE.R) GO TO 10
CALL QKSLP (-A*-B*«D.BP*W*KL
IF (K*NE.0) GO TO 10
S=1.
T=0.
RETURN
10 CALL SLPFND (A*B*C*D*BP*W*TF*EPS*R*S*T)
RETURN
END
SUBROUTINE SLPFND (A,B,C,D,BP,W,TF,EPS, R,S,T)

DATA BIG/1.10/ 

IF (A*NE.0.0 OR B*NE.0.0) GO TO 20 
IF (D*GT.0.) GO TO 10 
IF (D*LT.0.) GO TO 12 
GO TO 63 
10 S=-1. 
GO TO 13 
12 S=1. 
13 T=-(R*S-C)/D 
GO TO 62 
20 IF (D*NE.0.) GO TO 25 
S=0. 
GO TO 27 
25 S=SIGN(1.0*D) 
27 G=BP*A+W*A 
H=RP*B+W*A 
TA=TIMEC (G+H*D,BP,W,0.0,EPS) 
IF (TA*EQ.0.0) GO TO 45 
FA=F (A,B,C,D,BP,W,TA) 
IF (FA.LE.R) GO TO 32 
S=1. 
GO TO 33 
32 IF (FA.GE.R) GO TO 34 
33 T=SOLVET (A+B+C*R*S+D*BP,W,TA,S,5,EPS) 
GO TO 62 
34 IF (TA.GT.TF) GO TO 63 
TB=TIMEC (G+H*D,BP,W,TA+EPS) 
IF (TB*EQ.0.0) GO TO 45 
FB=F (A+B+C*D,BP,W,TA) 
IF (FB.LE.R) GO TO 36 
S=1. 
GO TO 37 
36 IF (FB.GE.R) GO TO 40 
S=1. 
37 T=SOLVET (A+B+C+R*S+D*BP,W,(TA+TB)*S,5,EPS) 
GO TO 62 
40 IF (D*EQ.0.) GO TO 63 
41 IF (TB.GT.TF) GO TO 63 
45 FD=BP*C+R*S 
IF (FD*EQ.0.) FO=FA 
CALL TANPT (A+B+C*R*S+D*BP,W,0.0,FD,TF,EPS,T) 
62 IF (TA.LT.TF) RETURN 
63 T=TA*BIG 
64 S=0. 
RETURN 
END
FUNCTION SOLVEE (A*B*C*D+E*BP*W*T0*EPS)
COMMON IFLAG
I=0
G=-BP*A=W*B
H=-BP*B+W*A
T=T0
1 I=I+1
EE=EXP(-BP*T)
WT=W*T
S=SIN(W*T)
CC=COS(W*T)
DT=((A*S+B*CC)*EE+C+(D+T)*T)/((G*S+H*CC)*EE+D+2*S*E*T)
T=T-DT
IF (I.EQ.15) GO TO 2
IF (ABS(DT).GT.ABS(T)*EPS) GO TO 1
SOLVEE=T
RETURN
2 SOLVEE=T
IFLAG=1
RETURN
END

FUNCTION SOLVET (A*B*C*D*BP*W*T0*EPS)
COMMON IFLAG
I=0
G=-BP*A=W*B
H=-BP*B+W*A
T=T0
1 I=I+1
E=EXP(-BP*T)
WT=W*T
S=SIN(W*T)
CC=COS(W*T)
DT=((A*S+B*CC)*E+C+D*T)/((G*S+H*CC)*EE+D+2*S*E*T)
T=T-DT
IF (I.EQ.15) GO TO 4
IF (ABS(DT).GT.ABS(T)*EPS) GO TO 1
SOLVET=T
RETURN
4 SOLVET=T
IFLAG=1
RETURN
END
SUBROUTINE TANMON (A, B, C, D, E, B*, W, TO, FO, TF, FF, EPS, T)

DATA P, TP, HP/3.1415926, 203185.1, 570796/

IF (B*NE.0.) GO TO 10
WT=HP
IF (A*FO*GT.0.) WT=WT+P
GO TO 20
10 WT=ATAN (A/B)
IF (B*NE.0.) GO TO 10
20 TA=WT/W
23 IF (TA*GT.0.) GO TO 25
TA=TA+TP/W
GO TO 23
25 TB=TA*P/W
FB=F (A, B, C, D, B*, W, TB)+E*TB*TB
IF (TA*LT.TF) GO TO 50
IF (TB*GT.TD) GO TO 30
T=SOLVEE (A, B, C, D, E, B*, W, (TO+TF)*5, EPS)
RETURN
30 IF (TB*GT.TF) GO TO 40
IF (FO*FB*LE.0.) GO TO 35
T=SOLVEE (A, B, C, D, E, B*, W, (TB+TF)*5, EPS)
RETURN
35 IF (FB*EQ.0.) GO TO 120
GO TO 105
40 T=SOLVEE (A, B, C, D, E, B*, W, TF, EPS)
RETURN
50 PP=F (A, B, C, D, B*, W, TA)
EE=EXP (-BP*TP/W)
FA=PP+C+(D+E*TA)*TA
IF (FA*FO*GT.0.) GO TO 80
IF (FA*EQ.0.) GO TO 110
IF (TB*GT.TD) GO TO 60
T=SOLVEE (A, B, C, D, E, B*, W, (TO+TA)*5, EPS)
RETURN
60 IF (FB*EQ.0.) GO TO 120
GO TO 95
80 TA=TA+TP/W
PP=PP*EE
FA=PP+C+(D+E*TA)*TA
IF (FA*FO*GT.0.) GO TO 90
IF (TA*LT.TF) STOP
GO TO 80
90 IF (FA*EQ.0.) GO TO 110
TA=TA*P/W
TB=TA+P/W
FB=F (A, B, C, D, B*, W, TB)+E*TE*TB
IF (FB*EQ.0.) GO TO 120
IF (FB*FO*LT.0.) GO TO 100
95 T=SOLVEE (A, B, C, D, E, B*, W, TB+HTP/W, EPS)
RETURN
100 FA=F (A, B, C, D, B*, W, TA)+E*TA*TA
IF (ABS(FA)=ABS(FB)) TB=TA
105 T=SOLVEE (A, B, C, D, E, B*, W, TB, EPS)
RETURN
110 T=TA
RETURN
120 T=TB
RETURN
END
SUBROUTINE TANPT (A,B,C,D,BP,W,T,B,T0,FC,TF,EPS,T)
DATA P,TP,HP,BIG/3.141593,6.263185,1.570796,1.E10/
IF (B*NE.0.) GO TO 130
WT = HP
IF (A*LT.0.) WT=WT+P
GO TO 140
130 WT = ATAN(A/B)
IF (B*LT.0.) WT=*T*P
140 TA=WT/W
GO TO 142
142 IF (TA*GT.T0) GO TO 145
TA=TA+P/W
GO TO 142
145 PP = F (A,B,0.,0.,BP,W,TA)
E=EXP(WTP/WT)
150 IF (FA*NE.0.) GO TO 160
IF (TA*GE.TF) GO TO 156
PP=PP*E
TA=TA+P/W
GO TO 150
156 T=BIG
RETURN
160 IF (FA*NE.0.) GO TO 170
T=TA
RETURN
170 TA=TA+P/W
IF (TA*LT.T0) GO TO 182
G = =BP*A = W*B
H = =BP*B + W*A
TC = TIMEC (G,H,D,BP,W,TA,EPS)
IF (TC*EQ.BIG) GO TO 185
FC = F (A,B,C,D,BP,W,TC)
IF (FC*EQ.BIG) GO TO 180
T = SOLVET (A,B,C,D,BP,W,TA,EPS)
RETURN
180 IF (FC*NE.0.) GO TO 185
T=TC
RETURN
182 TB=TA+P/W
FB=F (A,B,C,D,BP,W,TB)
IF (FB*LT.T0) GO TO 210
GO TO 200
185 TB=TA+P/W
FB=F (A,B,C,D,BP,W,TB)
IF (FB*LT.T0) GO TO 210
GO TO 200
190 IF (FB*NE.0.) GO TO 195
T=TB
RETURN
195 IF (TC*NE.BIG) GO TO 200
FA = F (A,B,C,D,BP,W,TA)
IF (ABS(FA)*LT.*ABS(FB)) TB=TA
RETURN
RETURN
END
SUBROUTINE TANPTfc (A*B*C*D*E*G*H*BP*W*TF*EPS*FO*F)
DATA P*Tp,HP,B1G/3*143*18*3*70796*1*10/
TLIM=-.5*D/E
IF (TLIM<.5) GO TO 10
IF (FO<.5) GO TO 15
IF (TLIM<.5) GO TO 15
AMP=SIGN(A*A+B+B)
S=SIGN(I*I+FU)
FV=(D+*TLIM)*TLIM*AMP*S
IF (FV<C*GT.0) GO TO 100
GO TO 20
10 IF (FO*E*GT.0) GO TO 100
15 TLIM=TF
20 IF (E*NE.0) GO TO 30
W=HP
IF (FO*A*GT.C) WT=WT+P
GO TO 40
30 W=ATAN(A/B)
IF (FO*E*GT.0) WT=WT+P
IF (WH*E*TO.0) WT=WT+P
40 TA=WT/W
PP=EXP(A*E+O.*O.+E*W+H*TA)
EE=EXP(-E*W*P/W)
50 FA=PP*C*TA*(D*E*TA)
IF (FO*FA*NE.0) GO TO 60
IF (TA*E*GE.*TLIM) GO TO 83
TA=TA+TP/W
PP=PP*EE
GO TO 50
80 IF (FA*NE.0) GO TO 85
T=TA
RETURN
83 IF (TLIM<.5) TF GO TO 100
85 TA=TA+TP/W
TC=IMED (G*H*U*E*O.*W*E*W*T*/A*TA+P/W*W*EPS)
IF (TC*EQ.RIG) GO TO 92
FC=F (A*R*C*D*BP*W*TC)+E*TC*TC
IF (FC*EQ.FE.0) GO TO 90
T=SOLVEE (A*A+E*O.*A*B*E*BP*W*TA*EPS)
RETURN
90 IF (FC*NE.0) GO TO 92
T=TC
RETURN
92 IF (FC*FO*GT.0) GO TO 100
TB=TA+P/W
FB=EXP(A*E+C*D*BP*W*TB)+E*TB*TB
IF (FB*NE.E.0) GO TO 94
T=TB
RETURN
94 IF (FB*FO*GT.0) GO TO 97
IF (TC*NE.E.BIG) GO TO 95
FA=F (A*E+C*D*BP*W*TA)+E*TA*TA
IF (ABS(FA)*LT*ABS(FB)) TE=TA
95 T=SOLVEE (A*A+C*D*BP*W*TB*EPS)
RETURN
97 T=SOLVEE (A*A+C*D*BP*W*TB+H*P/W*EPS)
RETURN
100 T=BG
RETURN
END
FUNCTION TIMEB (A*B*W)
DATA P*HP/3.141593.1*570796/
IF (A*NE.0.) GO TO 1
TIMEB = HP/W
RETURN
1 WT = ATAN(-B/A)
IF (WT.LE.0.) WT = WT + P
TIMEB = WT/W
RETURN
END

FUNCTION TIMEC (A*B*C*BP*W*TO*EPS)
DATA P*BIG/3.141593.1*E10/
IF (A*EQ.0.0.*AND*B*EQ.0.0.) GO TO 6
IF (C*NE.0.) GO TO 7
T = TIMEB (A*B*W)
10 IF (T*GT.0.) GO TO 8
T = T + P/W
GO TO 10
7 G = -BP*A = W*B
H = -BP*B + W*A
TA = TIMEB (G+H*W)
1 IF (TA*GT.0.) GO TO 2
TA = TA + P/W
GO TO 1
2 FA = F (A*B*C*0.*BP*T*TA)
IF (FA*NE.0.) GO TO 3
T = TA
GO TO 8
3 FO = F (A*B*C*0.*BP*W*TO)
IF (ABS(FO)LE.SQR(A*A+B*B)*EXP(-BP*TO)*EPS) FO = G.
IF (FO*FA*GE.0.) GO TO 4
T = SOLVET (A*B*C*0.*BP*W*(TA+TO)*0.5*EPS)
GO TO 8
4 TB = TA + P/W
FB = F (A*B*C*0.*BP*W*TB)
IF (FB*NE.0.) GO TO 5
T = TB
GO TO 8
5 IF (FB*FA*GT.0.) GO TO 6
T = SOLVET (A*B*C*0.*BP*W*(TA+TB)*0.5*EPS)
GO TO 8
6 T = BIG
8 TIMEC = T
RETURN
END
FUNCTION TIMED (A,B,C,D,BP,W,TO,TF, EPS)  
DATA P,T,P,HP,BIG/3*,141593,6,283185,1,570796,1,E10/  
IF (A*NE.0*.OR.B*NE.0*) GO TO 20  
 T = MC/D  
IF (T*LE*TO) GO TO 210  
GO TO 220  
20 IF (D*NE.0*) GO TO 30  
T = TIMEC (A,B,C,BP,W,TO, EPS)  
GO TO 200  
30 G = =BP*A = W*B  
 H = =BP*B + W*A  
TA = TIMEC (G,H,D,BP,W,TO, EPS)  
FO=F (A,B,C,D,BP,W,TO)  
IF (ABS(FO)*LE.SQRT(A*A+B*B)*EPS) FO=0.  
IF (TA*NE.BIG) GO TO 40  
IF (FO*D*GE.0) GO TO 210  
CALL TANPT (A,B,C,D,BP,W,TO, EPS)  
GO TO 200  
40 FA = F (A,B,C,D,BP,W,TA)  
IF (FA*FO*GE.0) GO TO 50  
T=SOLVET (A,B,C,D,BP,W,(TA+TO)*0.5, EPS)  
GO TO 200  
50 IF (TA*GT*TF) GO TO 210  
IF (FA*NE.0) GO TO 70  
T = TA  
GO TO 200  
70 TB = TIMEC (G,H,D,BP,W,TA, EPS)  
IF (TB*NE.BIG) GO TO 80  
IF (FA*GT*0) GO TO 210  
CALL TANPT (A,B,C,D,BP,W,TA,TF, EPS,T)  
GO TO 200  
80 FB=F (A,B,C,D,BP,W,TA)  
IF (FB*FA*GE.0) GO TO 90  
T = SOLVET (A,B,C,D,BP,W,(TA+TB)*0.5, EPS)  
GO TO 200  
90 IF (TB*GT*TF) GO TO 210  
IF (FB*NE.0) GO TO 110  
T = TB  
GO TO 200  
110 IF (FO*D*GE.0) GO TO 210  
115 CALL TANPT (A,B,C,D,BP,W,TO, EPS,T)  
200 IF (T*LT*TF) GO TO 220  
210 T=BIG  
220 TIMED=T  
RETURN  
END
FUNCTION TIMEE (A, B, C, D, E, BP, W, FO, TF, EPS)
DATA P, TP, HP, BIG/3.141593, 6.283185, 1.570796, 1.E10/
IF (A*NE.0* OR B*NE.0) GO TO 10
CALL QUADR (E, D, C, T)
GO TO 220
10 IF (E*NE.0) GO TO 20
T=TIMED (A, B, C, D, BP, W, 0, TF, EPS)
GO TO 240
20 G=BP*A+W*B
H=BP*B+W*A
TA=TIMED (G*H, D, 2*E, BP, W, 0, TF, TP/W, EPS)
IF (TA*NE.BIG) GO TO 30
FF=F (A, B, C, D, BP, W, TF)+E*TF*TF
IF (FD*FF*GE.0) GO TO 230
CALL TANMON (A, B, C, D, E, BP, W, 0, FO, TF, FF, EPS, T)
GO TO 220
30 FA=F (A, B, C, D, BP, W, TA)+E*TA*TA
IF (FD*FA*GE.0) GO TO 50
IF (TA*GE.TP/W) GO TO 40
T=SOLVEE (A, B, C, D, E, BP, W, TA, 5, EPS)
GO TO 220
40 CALL TANMON (A, B, C, D, E, BP, W, 0, FO, TA, FA, EPS, T)
GO TO 220
50 IF (TA*GE.TF) GO TO 230
IF (FA*NE.0) GO TO 55
T=TA
GO TO 240
55 TB=TIMED (G*H, D, 2*E, BP, W, TA, TF, TP/W, EPS)
IF (TB*NE.BIG) GO TO 60
FF=F (A, B, C, D, BP, W, TF)+E*TF*TF
IF (FD*FF*GE.0) GO TO 230
CALL TANMON (A, B, C, D, E, BP, W, TA, FA, TF, FF, EPS, T)
GO TO 220
60 FB=F (A, B, C, D, BP, W, TB)+E*TB*TB
IF (FB*FA*GE.0) GO TO 80
T=SOLVEE (A, B, C, D, E, BP, W, (TA+TB), 5, EPS)
GO TO 220
80 IF (TB*GE.TF) GO TO 230
IF (FB*NE.0) GO TO 85
T=TB
GO TO 240
85 FO=FA
CALL TANPTE (A, B, C, D, E, G, H, BP, W, TF, EPS, FO, T)
220 IF (T*LT.TF) GO TO 240
230 T=BIG
240 TIMEE=T
RETURN
END