A LINEAR, THREE-DIMENSIONAL MODEL FOR THE VIBRATIONS OF A SEMI-TRAILER TRUCK

by

CHRISTOPHER CHARLES CHAPMAN

B. S., Kansas State University, 1970

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

Approved by:

Hugh S. Walker

Major Professor

TABLE OF CONTENTS

ILLEGIBLE DOCUMENT

THE FOLLOWING DOCUMENT(S) IS OF POOR LEGIBILITY IN THE ORIGINAL

THIS IS THE BEST COPY AVAILABLE

LIST OF FIGURES

NOMENCLATURE

Coordinates

x_1 vertical motion of center of mass for front tractor axle

x_2 vertical motion of center of mass for tractor

x_3 vertical motion of center of mass for rear tractor axle

x_4 vertical motion of center of mass for trailer

x_5 vertical motion of center of mass for trailer axle

x_6 longitudinal motion of center of mass for tractor

x_7 transverse motion of center of mass for tractor

x_8 longitudinal motion of center of mass for trailer

x_9 transverse motion of center of mass for trailer

z_1 transverse motion of center of mass for front tractor axle

z_3 transverse motion of center of mass for rear tractor axle

z_5 transverse motion of center of mass for trailer axle

θ_1 rolling motion for front tractor axle

θ_2 rolling motion for tractor

θ_3 rolling motion for rear tractor axle

θ_4 rolling motion for trailer

θ_5 rolling motion for trailer axle

ϕ_2 pitching motion for tractor

ϕ_4 pitching motion for trailer

ξ_2 yawing motion for tractor

ξ_4 yawing motion for trailer

Dimensions

- a transverse offset of tractor center of mass from vehicle center line
- b longitudinal distance of tractor center of mass from kingpin
- c vertical distance of tractor center of mass from horizontal plane containing tractor spring attachment points
- d transverse offset of trailer center of mass from vehicle center line
- e longitudinal distance of trailer center of mass from kingpin
- f vertical distance of trailer center of mass from horizontal plane containing trailer spring attachment points
- g longitudinal distance from tractor center of mass to front spring attachment points of tractor
- h longitudinal distance from tractor center of mass to rear spring attachment points of tractor
- p vertical distance from tractor center of mass to kingpin
- q vertical distance from trailer center of mass to kingpin
- r longitudinal distance from trailer center of mass to spring attachment points of trailer
- s transverse distance between spring attachment points of tractor
- t transverse distance between spring attachment points of trailer
- u one-half the distance between axle suspension points
- v assumed rigid height for axle suspension
- w one-half the transverse distance between vehicle tires
- y assumed rigid height for vehicle tires

Inertia Parameters

m_1 mass of tractor front axle assembly

m_2 mass of tractor less suspension

m_3 mass of tractor rear axle assembly

m_4 mass of trailer less suspension

m_5 mass of trailer axle assembly

m_6 mass of tractor

m_7 mass of tractor less suspension

m_8 mass of trailer

m_9 mass of trailer less suspension

I_1 rolling inertia of tractor front axle

I_2 rolling inertia of tractor rigid body

I_3 rolling inertia of tractor rear axle

I_4 rolling inertia of trailer rigid body

I_5 rolling inertia of trailer axle

J_2 pitching inertia of tractor rigid body

J_4 pitching inertia of trailer rigid body

H_2 yawing inertia of tractor rigid body

H_4 yawing inertia of trailer rigid body

P_{62} product of inertia for longitudinal-vertical plane of tractor

P_{67} product of inertia for longitudinal-transverse plane of tractor

P_{27} product of inertia for vertical-transverse plane of tractor

P_{84} product of inertia for longitudinal-vertical plane of trailer

P_{89} product of inertia for longitudinal-transverse plane of trailer

P_{49} product of inertia for vertical-transverse plane of trailer

Springs and Dampers

k_1, c_1 equivalent vertical spring and damping constant for left front suspension of tractor

k_2, c_2 equivalent vertical spring and damping constant for right front suspension of tractor

k_3, c_3 equivalent vertical spring and damping constant for left rear suspension of tractor

k_4, c_4 equivalent vertical spring and damping constant for right rear suspension of trailer

k_5, c_5 equivalent vertical spring and damping constant for left suspension of trailer

k_6, c_6 equivalent vertical spring and damping constant for right suspension of trailer

k_7, c_7 equivalent vertical spring and damping constant for left front tire of tractor

k_8, c_8 equivalent vertical spring and damping constant for right front tire of tractor

k_9, c_9 equivalent vertical spring and damping constant for left rear tire of tractor

k_{10}, c_{10} equivalent vertical spring and damping constant for right rear tire of tractor

k_{11}, c_{11} equivalent vertical spring and damping constant for left rear tire of trailer

k_{12}, c_{12} equivalent vertical spring and damping constant for right rear tire of trailer

k_{13}, c_{13} equivalent transverse spring and damping constant for left front tire of tractor

k_{14}, c_{14} equivalent transverse spring and damping constant for right front tire of tractor

k_{15}, c_{15} equivalent transverse spring and damping constant for left front suspension of tractor

k_{16}, c_{16} equivalent transverse spring and damping constant for right front suspension of tractor

k_{17}, c_{17} equivalent transverse spring and damping constant for left rear tire of tractor

k_{18}, c_{18} equivalent transverse spring and damping constant for right rear tire of tractor

k_{19}, c_{19} equivalent transverse spring and damping constant for left rear suspension of tractor

k_{20}, c_{20} equivalent transverse spring and damping constant for right rear suspension of tractor

k_{21}, c_{21} equivalent transverse spring and damping constant for left tire of trailer

k_{22}, c_{22} equivalent transverse spring and damping constant for right tire of trailer

k_{23}, c_{23} equivalent transverse spring and damping constant for left suspension of trailer

k_{24}, c_{24} equivalent transverse spring and damping constant for right suspension of trailer

Chapter I

INTRODUCTION

Semi-trailer trucks are assuming an ever-increasing role in the transportation of consumer and capital goods. They provide the necessary link between the manufacturer and the consumer. As the advertisement states, "If you got it, a truck brought it." Although the semi-trailer truck has obtained increased importance, the understanding of its vibrational characteristics has lagged. Therefore the development of a three-dimensional model for the vibrations of a semi-trailer truck shall aid in a better understanding of the vehicle's motions.

A number of investigations have analyzed the vibrational characteristics of semi-trailer trucks, but have limited their studies to planar motions (8), (11), (16), (17), (18).* The most comprehensive planar analysis was given by Potts (14), (15). The assumptions necessary for a planar analysis preclude certain physical realities. In a planar analysis the yawing and rolling motions of the vehicle are ignored. No distinction is made between the "bumps" encountered by the left and right side wheel sets. In the model to be presented, all these effects are accounted for.

After a number of assumptions were made to reduce the complexity of a "real" truck, energy methods and Lagrange's equation were used to develop the equations of motion for the vehicle. Twenty-one coordinates were initally chosen to describe the motions of the vehicle's components. Three kinematic

---

*Numbers in parentheses refer to references given in the selected bibliography.

equations of constraint, due to the fifth wheel kingpin, and one ignorable coordinate were discovered. Thus the number of independent degrees of freedom was reduced to seventeen. Two digital computer programs were developed to solve various steady state problems. While these programs were used primarily as a check for the correctness of the mathematical model, they could be used in an extensive steady state analysis.

The three-dimensional model developed in this thesis privides a useful addition to the simulation of semi-trailer truck vibrations.

Chapter II

PHYSICAL DESCRIPTION OF MODEL

Physical Assumptions

The complexity of a real semi-trailer truck was reduced by making a number of simplifying assumptions. The tractor and trailer, less their suspensions, were assumed to be rigid bodies. Tandems and dual tire sets were lumped into an equivalent single axle-tire assembly; thus the vehicle was assumed to have three axles and six tires. Each of the equivalent suspension assemblies was assumed to be rigid in the longitudinal direction, thereby constraining axle movement to planar motion normal to the direction of travel. Each of the six tires was supposed equivalent to linear springs and viscous (linear) dampers possessing different properties in the vertical and transverse directions, as shown in figures 3, 4 and 5. The axle suspensions were also substituted by equivalent linear springs and dampers in both the vertical and transverse directions. These springs and dampers were assumed to be attached to the tractor and trailer directly above the axle.

Rigid bars, normal to the horizontal plane, were fixed to the rigid axles at the spring and tire attachment points (see Fig. 3). The transverse motion of these bar end points was used to describe the interaction between chassis, axle, and tire, due to the rolling motions of the vehicle components.

Oscillations about the equilibrium positions were supposed small, so that rigid body motions could be described by linear functions of the chosen coordinates.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

At the tire-road interface, slippage was not permitted. Nonlinearites such as cargo movement, wheel-hop, and coulomb friction were ignored.

Describing Coordinates

The equilibrium positions of the sprung and unsprung centers of mass were used as the origins of the fixed coordinate systems. Figure 1 shows a general set of rectangular axes used. The longitudinal and transverse motions of the center of mass were described by the X and Z coordinates, respectively, where the X and Z axes lie in a horizontal plane parallel to the road surface, with the positive X axis coincident with the direction of travel. The positive Z axis points outward and left of the vehicle, as defined by the driver. The vertical component of motion was given along the Y axis. The rigid body rotations of roll, pitch, and yaw were given respectively by θ, ϕ and ξ. Positive angular rotations were defined by the right-hand rule.

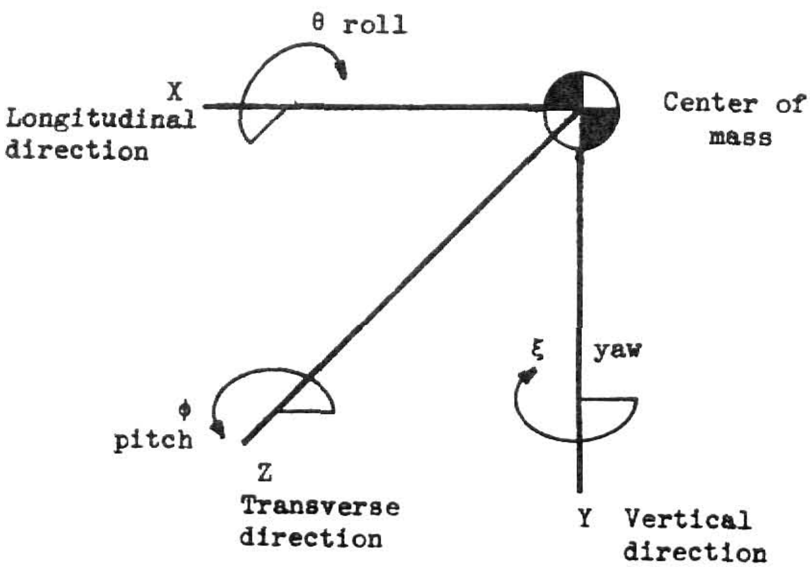

Figure 1: Rectangular axes.

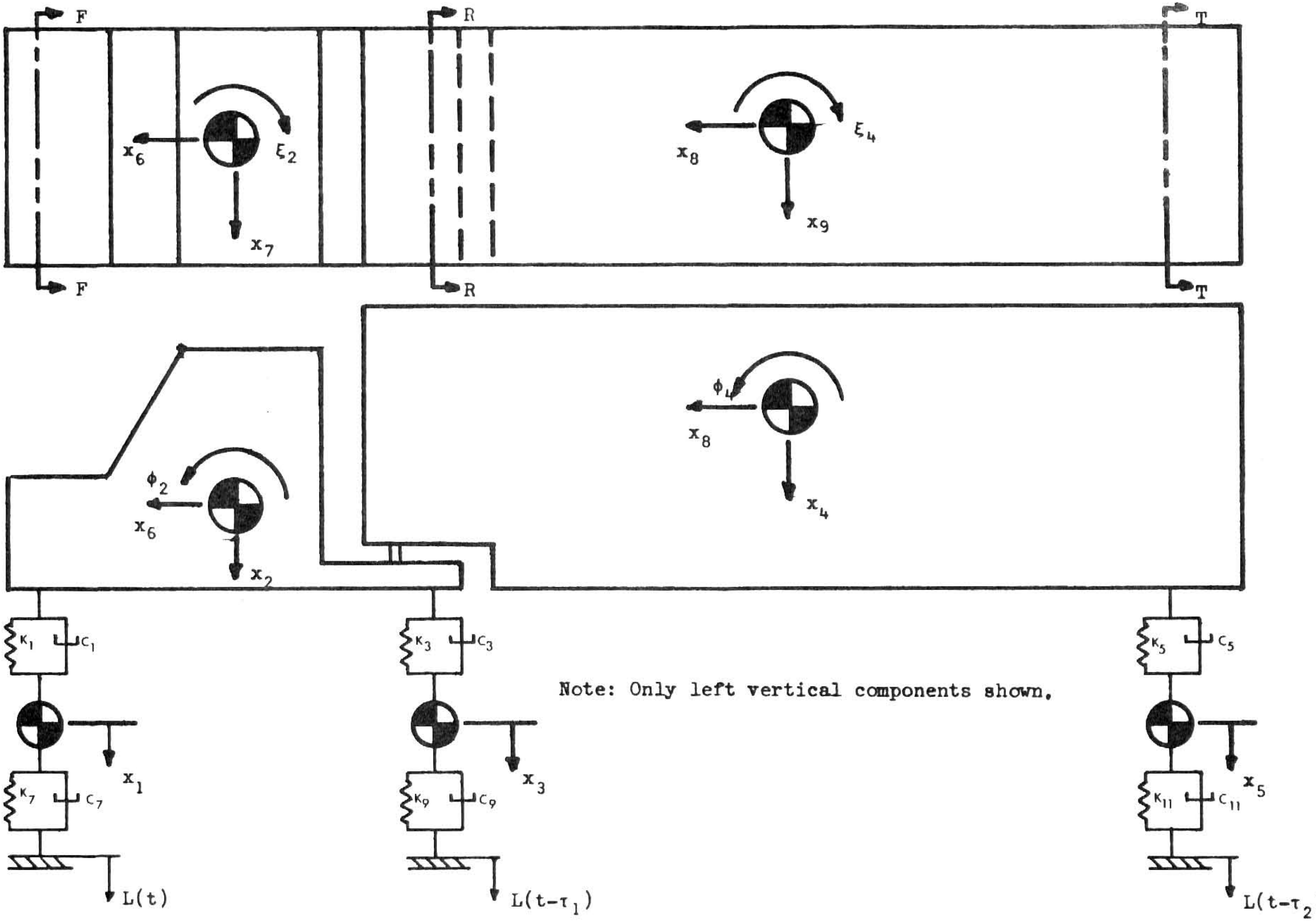

Figure 2: Top and left side view of semi-trailer truck.

Figures 2, 3, 4, and 5 show that twenty-one coordinates were used to describe the motion of the vehicle; six coordinates each for the tractor and the trailer, and three coordinates for each axle assembly.

Section F-F

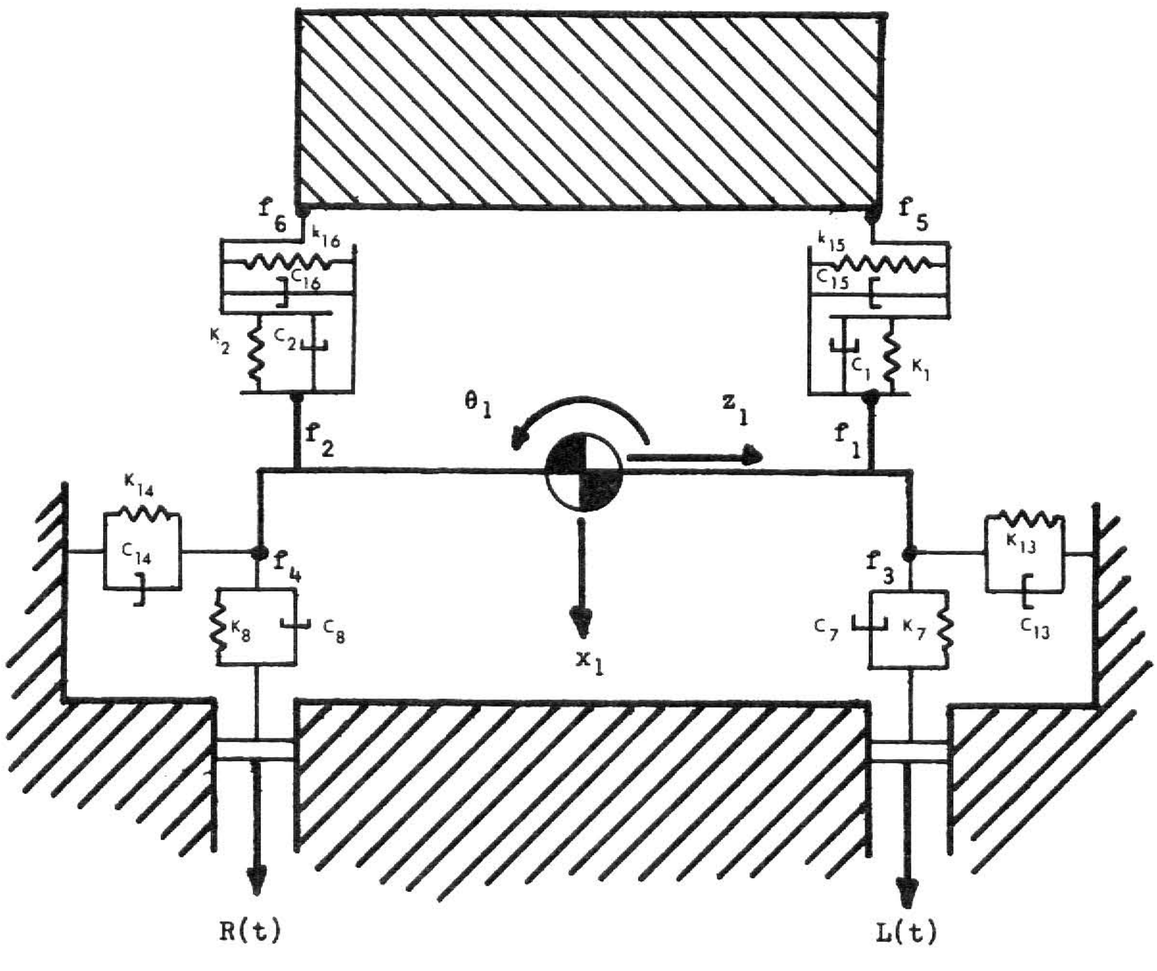

Figure 3: Transverse section of tractor at front axle.

Section R-R

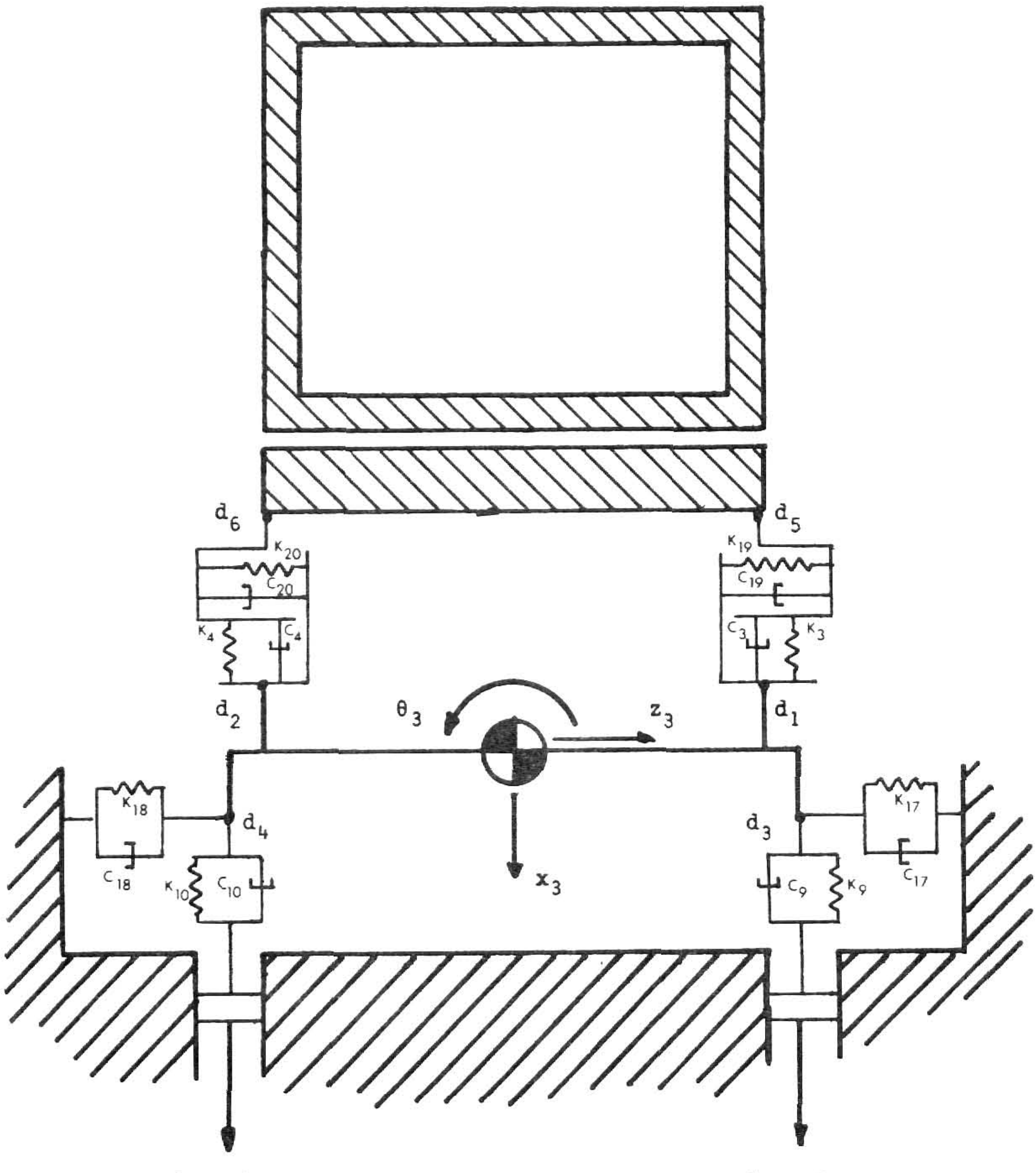

Figure 4: Transverse section of tractor at rear axle.

Section T-T

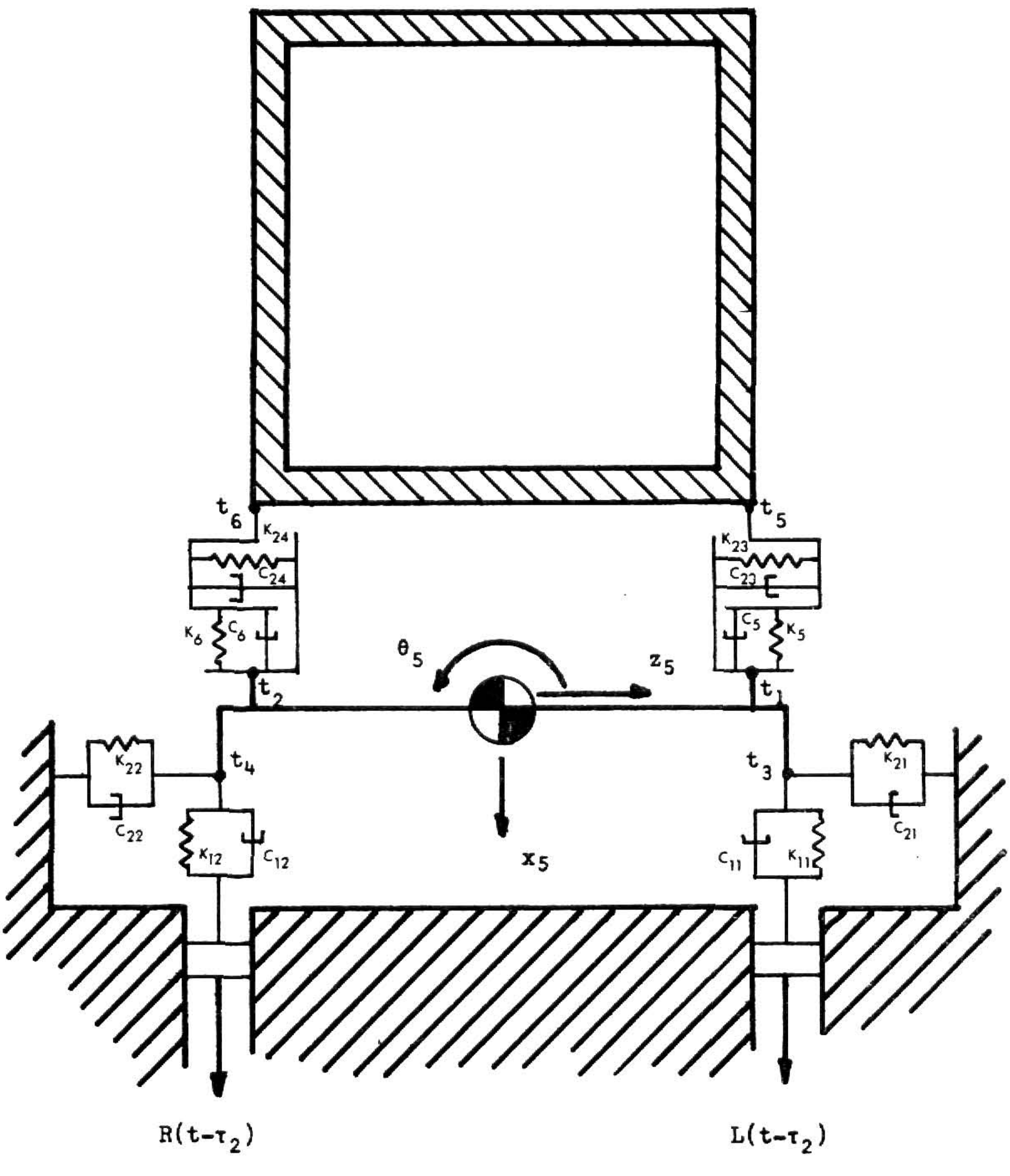

Figure 5: Transverse section at trailer axle.

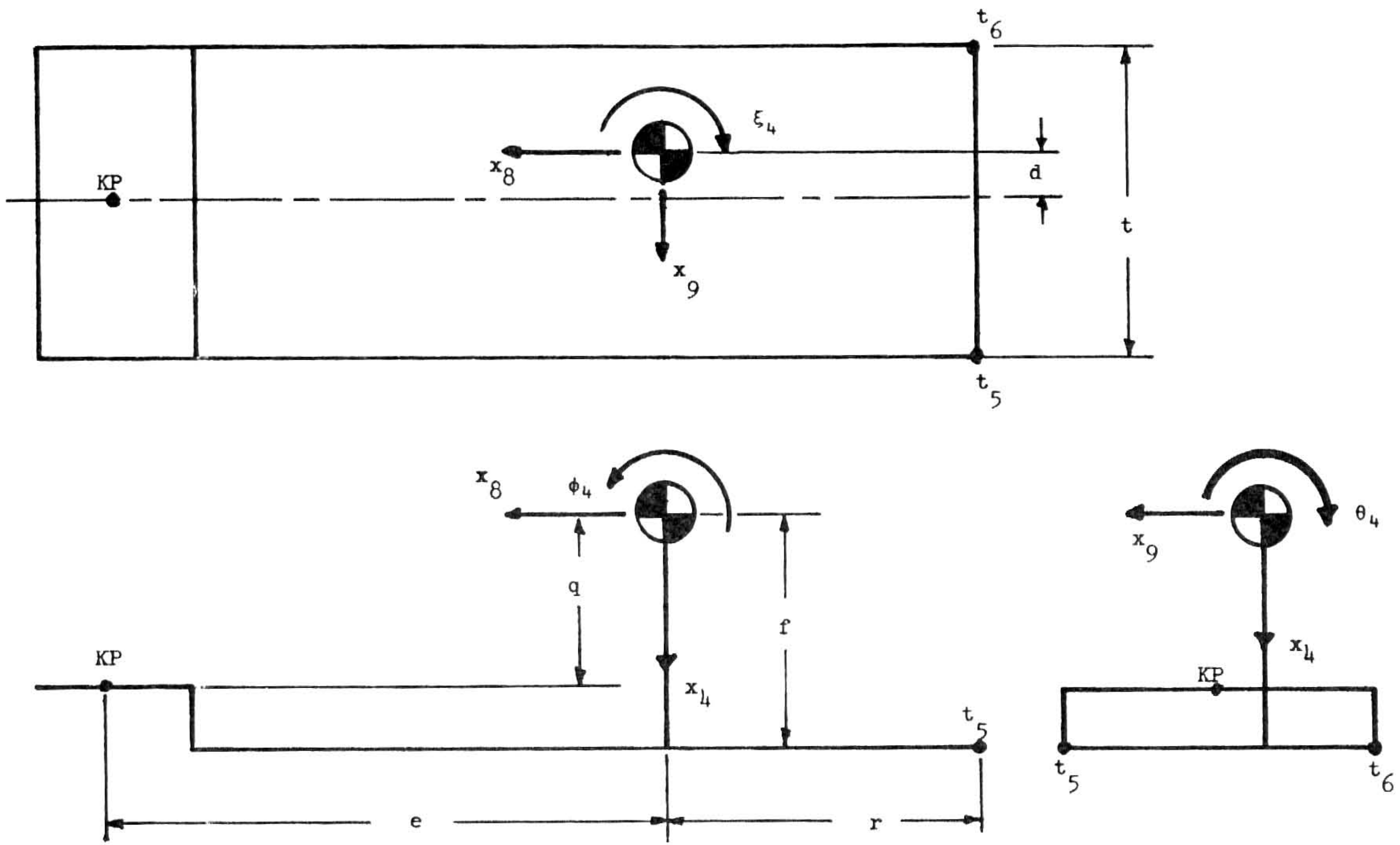

Figure 6: Dimensional schematic of trailer.

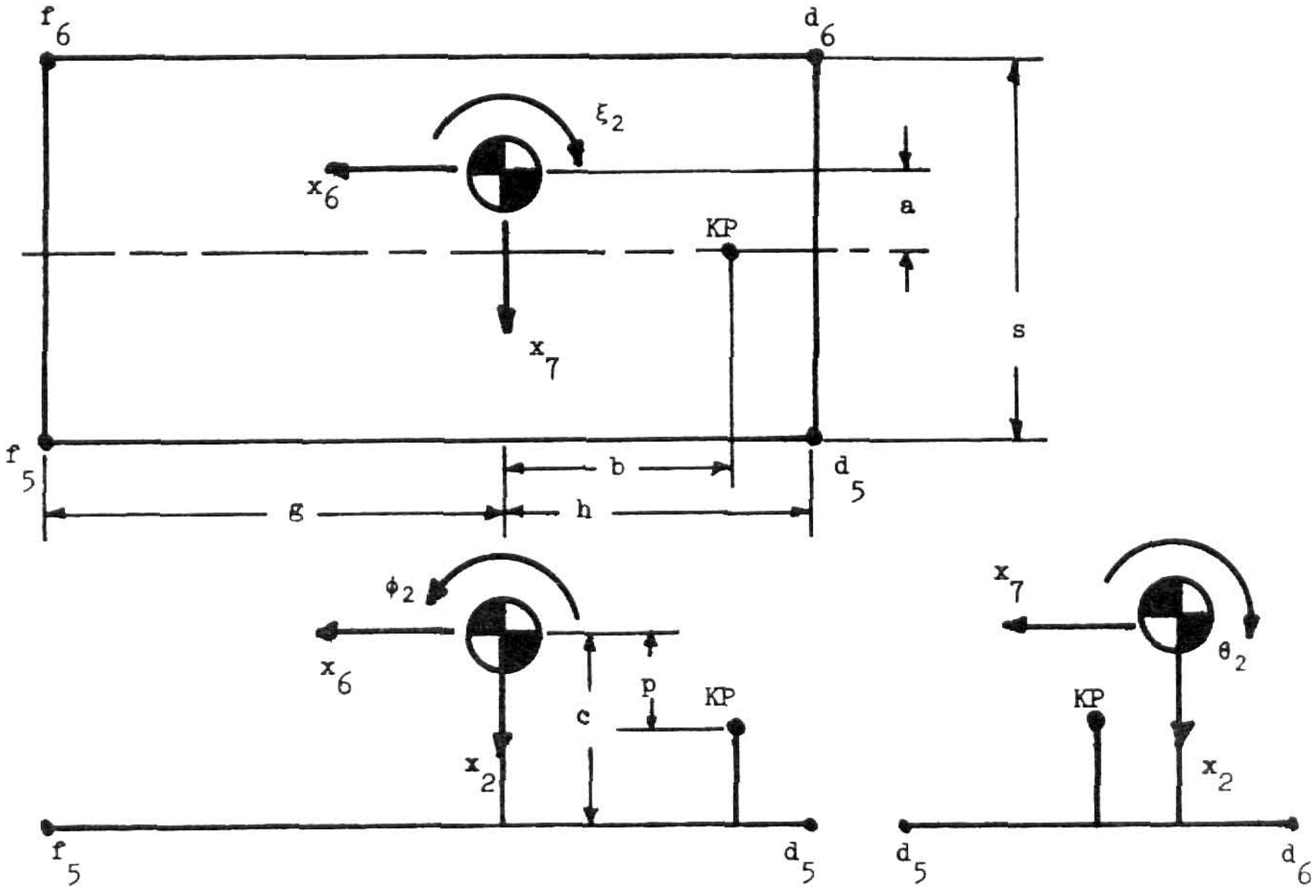

Figure 7: Dimensional schematic of tractor.

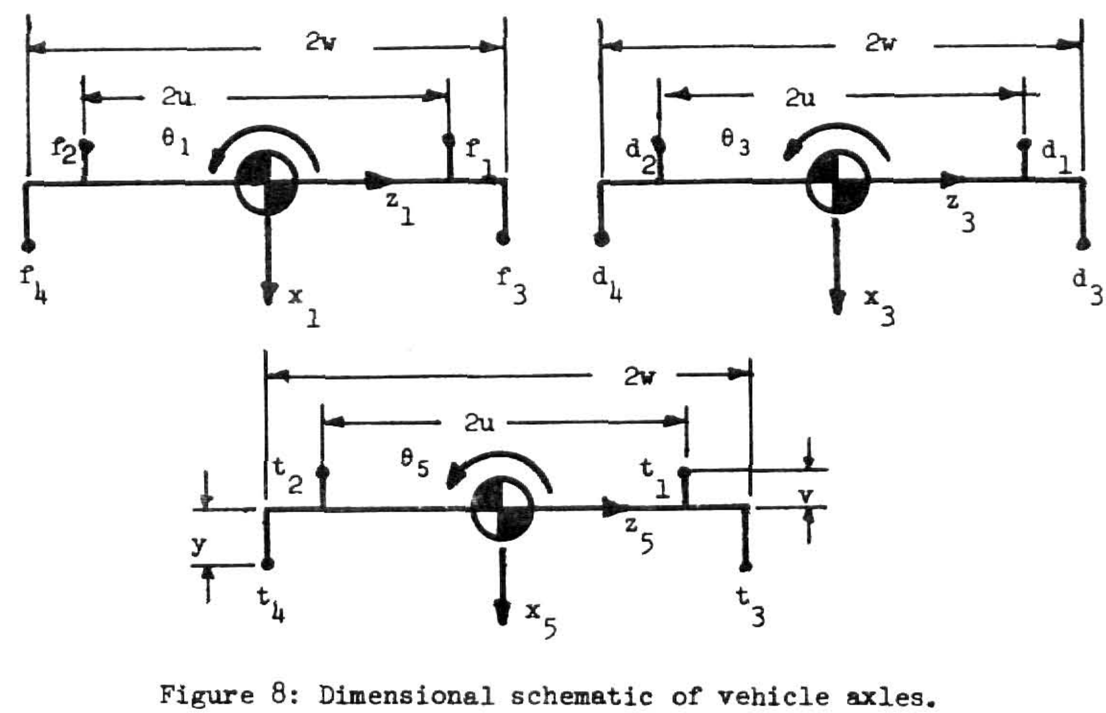

Figure 8: Dimensional schematic of vehicle axles.

Chapter III

MATHEMATICAL DESCRIPTION OF MODEL

Lagrange's Equation

The Lagrangian equations of motion are

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{x}_i}\right) - \frac{\partial T}{\partial x_i} + \frac{\partial D}{\partial \dot{x}_i} + \frac{\partial V}{\partial x_i} = q_i \, , \qquad [1]$$

where

$i = 1, 2, \ldots, n$,

n = number of degrees of freedom of the system,

T = kinetic energy function,

D = dissipation function,

V = potential energy function,

x = generalized coordinate,

q = generalized force.

When linear theory is used to analyze a vibrations problem, certain simplifications in the derivation of the equations of motion result. For linear theory, Langhaar(10) shows that the kinetic, potential, and dissipation functions can be expressed as homogeneous quadratic functions of the form

$$T = \frac{1}{2} \sum_i^n \sum_j^n m_{ij} \dot{x}_i \dot{x}_j ,$$

$$V = \frac{1}{2} \sum_i^n \sum_j^n k_{ij} x_i x_j ,$$

$$D = \frac{1}{2} \sum_i^n \sum_j^n c_{ij} \dot{x}_i \dot{x}_j ,$$

where the m_{ij}'s, k_{ij}'s, and c_{ij}'s are constants. Substituting the above into [1] yeilds an equivalent expression in summation form as

$$\sum_j^n (m_{ij} \ddot{x}_j + c_{ij} \dot{x}_j + k_{ij} x_j) = q_i \qquad (i = 1, 2, . . . ,n),$$

or in matrix form as

$$[m_{ij}]\{\ddot{X}\} + [c_{ij}]\{\dot{X}\} + [k_{ij}]\{X\} = \{Q\},$$

where

$$m_{ij} = \frac{\partial^2 T}{\partial \dot{x}_i \partial \dot{x}_j} ,$$

$$c_{ij} = \frac{\partial^2 D}{\partial \dot{x}_i \partial \dot{x}_j} , \qquad [2]$$

$$k_{ij} = \frac{\partial^2 V}{\partial x_i \partial x_j} .$$

Thus after the energy functions have been determined, the matrix elements are easily obtained by taking partial derivatives of the respective functions.

Energy Functions

Referring to figures 2, 3, 4, and 5, the kinetic energy can be written down immediately as

$$T = \frac{1}{2}\{m_1\dot{x}_1^2 + m_2\dot{x}_2^2 + m_3\dot{x}_3^2 + m_4\dot{x}_4^2 + m_5\dot{x}_5^2 + m_6\dot{x}_6^2 + m_7\dot{x}_7^2 + m_8\dot{x}_8^2 +$$

$$m_9\dot{x}_9^2 + m_1\dot{z}_1^2 + m_3\dot{z}_3^2 + m_5\dot{z}_5^2 + I_1\dot{\theta}_1^2 + I_2\dot{\theta}_2^2 + I_3\dot{\theta}_3^2 + I_4\dot{\theta}_4^2 +$$

$$I_5\dot{\theta}_5^2 + J_2\dot{\phi}_2^2 + J_4\dot{\phi}_4^2 + H_2\dot{\xi}_2^2 + H_4\dot{\xi}_4^2 - 2P_{62}\dot{\theta}_2\dot{\xi}_2 - 2P_{67}\dot{\theta}_2\dot{\phi}_2 -$$

$$2P_{27}\dot{\xi}_2\dot{\phi}_2 - 2P_{84}\dot{\theta}_4\dot{\xi}_4 - 2P_{89}\dot{\theta}_4\dot{\phi}_4 - 2P_{49}\dot{\xi}_4\dot{\phi}_4\}. \qquad [3]$$

The potential energy in terms of the rigid body attachment points (see figures 3, 4, and 5) is given by

$$V = \frac{1}{2}\{k_7[f_{3v}-L(t)]^2 + k_{13}[f_{3h}]^2 + k_8[f_{4v}-R(t)]^2 + k_{14}[f_{4h}]^2 +$$

$$k_{11}[t_{3v}-L(t-\tau_2)]^2 + k_{21}[t_{3h}]^2 + k_{12}[t_{4v}-R(t-\tau_2)]^2 +$$

$$k_{10}[d_{4v}-R(t-\tau_1)]^2 + k_{18}[d_{4h}]^2 + k_3[d_{5v}-d_{1v}]^2 + k_{17}[d_{3h}]^2 +$$

$$k_{19}[d_{5h}-d_{1h}]^2 + k_4[d_{6v}-d_{2v}]^2 + k_{20}[d_{6h}-d_{2h}]^2 + k_{22}[t_{4h}]^2 +$$

$$k_{16}[f_{6h}-f_{2h}]^2 + k_9[d_{3v}-L(t-\tau_1)]^2 + k_1[f_{5v}-f_{1v}]^2 +$$

$$k_{15}[f_{5h}-f_{1h}]^2 + k_5[t_{5v}-t_{1v}]^2 + k_{23}[t_{5h}-t_{1h}]^2 + k_6[t_{6v}-t_{2v}]^2 +$$

$$k_{24}[t_{6h}-t_{2h}]^2 + k_2[f_{6v}-f_{2v}]^2\}, \qquad [4]$$

where the second subscripts, v and h, represent respectively the vertical and transverse displacements of the various attachment points.

Before the potential energy function can be written in terms of the chosen coordinates, the displacement of a general point in a rigid body due to small rotations is required. Consider the general point shown in figure 9 and defined by the column vector

$$\{R\} = \begin{vmatrix} r_x \\ r_y \\ r_z \end{vmatrix} .$$

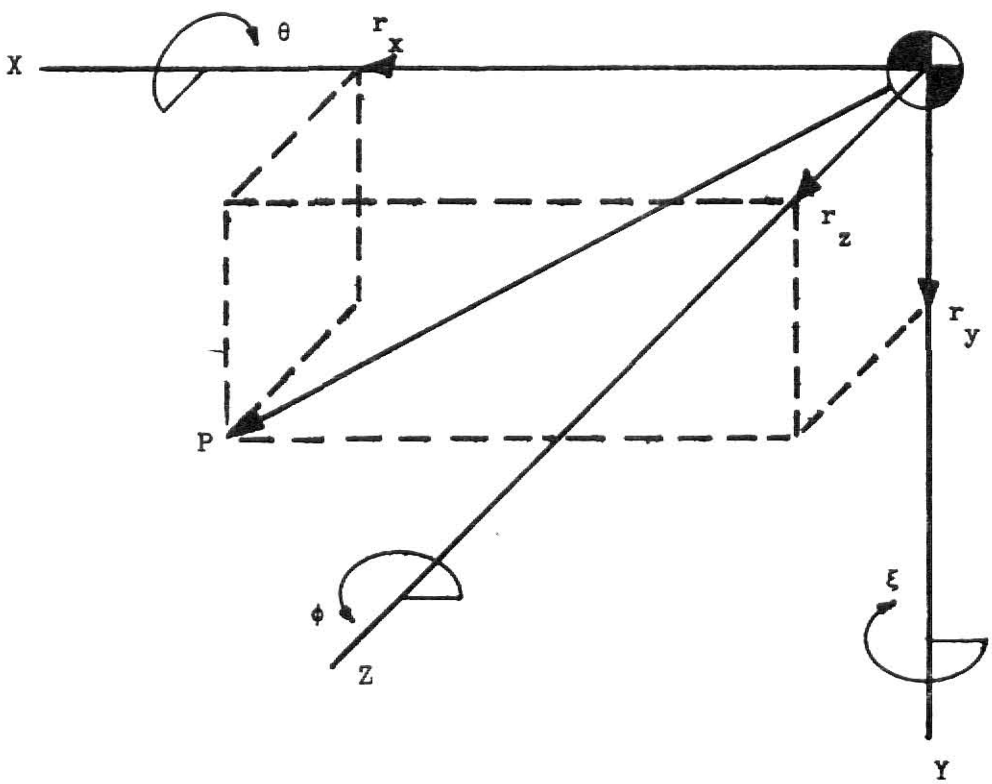

Figure 9: Definition of rigid body point.

The final position of point P after three successive rotations is

$$\{R\} = [\Xi]\,[\Phi]\,[\Theta]\,\{R\},$$

where

$$[\Xi] = \begin{vmatrix} \cos\xi & 0 & \sin\xi \\ 0 & 1 & 0 \\ -\sin\xi & 0 & \cos\xi \end{vmatrix},$$

$$[\Phi] = \begin{vmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{vmatrix},$$

$$[\Theta] = \begin{vmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{vmatrix}.$$

From theory of matrices, it is known that general matrix products are noncommutative; thus the final position of the point P is dependent upon the order in which the rotations are assumed to occur. However, the transformation matrices become commutative if the magnitudes of their elements are sufficiently small. Since in linear theory we stipulate that the coordinates are limited to small oscillations, it may be assumed that $\cos\xi \simeq 1$, $\cos\phi \simeq 1$, $\cos\theta \simeq 1$; and that $\sin\xi \simeq \xi$, $\sin\phi \simeq \phi$, $\sin\theta \simeq \theta$. If in addition it is assumed that the products of the small quantities (ξ, ϕ, θ) are similar to zero, the final position of the point P is not dependent upon the order of rotations.

With the above assumptions, the new position vector is given by

$$\{R'\} = \begin{vmatrix} 1 & -\phi & \xi \\ \phi & 1 & -\theta \\ -\xi & \phi & 1 \end{vmatrix} \{R\}.$$

The displacement of P is

$$\{\bar{R}\} = \{R'\} - \{R\} ,$$

or in scalar form,

$$\bar{r}_x = -r_y\phi + r_z\xi ,$$

$$\bar{r}_y = r_x\phi - r_z\theta , \qquad [5]$$

$$\bar{r}_z = -r_x\xi + r_y\theta .$$

The displacements of the attachment points used in the potential energy function [4] are determined by referring to the dimensional schematics for the vehicle (see figures 6, 7, and 8) and using the equations of [5]. Substituting the appropriate relationships for the point displacements gives the potential energy function in terms of the chosen coordinates as

$$
\begin{aligned}
V = \frac{1}{2}\{ & k_7[x_1-w\theta_1-L(t)]^2 + k_{13}[z_1+y\theta_1]^2 + k_8[x_1+w\theta_1-R(t)]^2 + k_{14}[z_1+y\theta_1]^2 + \\
& k_{12}[x_5+w\theta_5-R(t-\tau_2)]^2 + k_{22}[z_5+y\theta_5]^2 + k_5[x_4-r\phi_4-(\tfrac{t+2d}{2})\theta_4-x_5+u\theta_5]^2 + \\
& k_{20}[x_7+h\xi_2+c\theta_2-z_3+v\theta_3]^2 + k_{11}[x_5-w\theta_5-L(t-\tau_2)]^2 + k_{21}[z_5+y\theta_5]^2 + \\
& k_9[x_3-w\theta_3-L(t-\tau_1)]^2 + k_{17}[z_3+y\theta_3]^2 + k_{10}[x_3+w\theta_3-R(t-\tau_1)]^2 + \\
& k_1[x_2+g\phi_2-(\tfrac{s+2a}{2})\theta_2-x_1+u\theta_1]^2 + k_{15}[x_7-g\xi_2+c\theta_2-z_1+v\theta_1]^2 + \\
& k_2[x_2+g\phi_2+(\tfrac{s-2a}{2})\theta_2-x_1-u\theta_1]^2 + k_{16}[x_7-g\xi_2+c\theta_2-z_1+v\theta_1]^2 + \\
& k_{19}[x_7+h\xi_2+c\theta_2-z_3+v\theta_3]^2 + k_4[x_2-h\phi_2+(\tfrac{s-2a}{2})\theta_2-x_3-u\theta_3]^2 + \\
& k_{23}[x_9+r\xi_4+f\theta_4-z_5+v\theta_5]^2 + k_6[x_4-r\phi_4+(\tfrac{t-2d}{2})\theta_4-x_5-u\theta_5]^2 + \\
& k_{18}[z_3+y\theta_3]^2 + k_3[x_2-h\phi_2-(\tfrac{s+2a}{2})\theta_2-x_3+u\theta_3]^2 + \\
& k_{24}[x_9+r\xi_4+f\theta_4-z_5+v\theta_5]^2 \}. \qquad [6]
\end{aligned}
$$

Since it was assumed that a viscous damper was in parallel with every spring, the dissipation function takes a form analogous to that of the potential energy function. By simply substituting the k's by the c's and the displacements by their time derivatives, the dissipation function could be determined. However before writing the dissipation function explicitly, it should be recognized that the chosen coordinates are not all independent.

After studying figure 2, it is seen that the kingpin is a point common to both the tractor and the trailer; thus the motion of this attachment point can be described in terms of either tractor or trailer coordinates. This common description of the kingpin motion results in three kinematic constraint equations. The motion of the kingpin in terms of the tractor and trailer coordinates is given respectively by the following vector equations.

$$\overline{KP} = \overline{S}_c + \overline{S}_{kp/c} ,$$

$$\overline{KP} = \overline{S}_{c'} + \overline{S}_{kp/c'} .$$

Equating the two expressions gives

$$\overline{S}_c + \overline{S}_{kp/c} = \overline{S}_{c'} + \overline{S}_{kp/c'} .$$

Resolving this vector equation into its rectangular components by using the equations of [5] yeilds the three constraint equations

$$x_8 - q\phi_4 + d\xi_4 = x_6 - p\phi_2 + a\xi_2 ,$$

$$x_4 + e\phi_4 - d\theta_4 = x_2 - b\phi_2 - a\theta_2 ,$$

$$x_9 - e\xi_4 + q\theta_4 = x_7 + b\xi_2 + p\theta_2 .$$

Solving for x_8, x_4, x_9 in the above equations results in expressions for these dependent coordinates, given by

$$x_8 = x_6 - p\phi_2 + a\xi_2 + q\phi_4 - d\xi_4 \ ,$$

$$x_4 = x_2 - b\phi_2 - a\theta_2 - e\phi_4 + d\theta_4 \ , \qquad [7]$$

$$x_9 = x_7 + b\xi_2 + p\theta_2 + e\xi_4 - q\theta_4 \ .$$

Thus the number of coordinates required to describe the motion of the vehicle has been reduced by three.

After eliminating the dependent variables from [6], using the equations of [7], it was observed that V was not a function of x_6 (note this also implies that D is not a function of $\dot{x}_6$). As described by Wells(19), the coordinate x_6 is recognized to be an ignorable coordinate. After the dependent velocities have been eliminated from [3] by the time derivative of the equations of [7], the Lagrangian equation of motion corresponding to the variable x_6 becomes

$$\frac{d}{dt}\left(\frac{\partial T}{\partial x_6}\right) = 0 = m_6\ddot{x}_6 + m_8(\ddot{x}_6-p\ddot{\phi}_2+a\ddot{\xi}_2+q\ddot{\phi}_4-d\ddot{\xi}_4) \ . \qquad [8]$$

Solving [8] for $\ddot{x}_6$ gives

$$\ddot{x}_6 = \frac{m_8}{m_6+m_8}\,(p\ddot{\phi}_2-a\ddot{\xi}_2-q\ddot{\phi}_4+d\ddot{\xi}_4). \qquad [9]$$

Integrating [9] twice with respect to time yeilds

$$x_6 = \frac{m_8}{m_6+m_8}\,(p\phi_2 - a\xi_2 - q\phi_4 + d\xi_4) + C_1 t + C_2 \ . \qquad [10]$$

The constant C_1 is equal to zero because the coordinate x_6 describes oscillations about a fixed point and can not grow with time. Since it was assumed that $x_6(0) = 0$, C_2 must also be equal to zero. Having determined the constants C_1 and C_2, the ignorable coordinate x_6 is discovered to be equal to

$$x_6 = \frac{m_8}{m_6+m_8}\,(p\phi_2 - a\xi_2 - q\phi_4 + d\xi_4) \ . \qquad [11]$$

Replacing the dependent variables by their equivalent functions, the kinetic, potential, and dissipation energy functions assume their final forms as given by [12], [13], and [14] respectively.

$$\begin{aligned} T = \frac{1}{2}\{ & m_1\dot{x}_1^2 + m_2\dot{x}_2^2 + m_3\dot{x}_3^2 + m_4[\dot{x}_2 - b\dot{\phi}_2 - a\dot{\theta}_2 - e\dot{\phi}_4 + d\dot{\theta}_4]^2 + m_5\dot{x}_5^2 + \\ & \frac{m_6 m_8}{m_6+m_8}[p\dot{\phi}_2 - a\dot{\xi}_2 - q\dot{\phi}_4 + d\dot{\xi}_4]^2 + m_7\dot{x}_7^2 + m_9[\dot{x}_7 + b\dot{\xi}_2 + p\dot{\theta}_2 + e\dot{\xi}_4 - q\dot{\theta}_4]^2 + \\ & m_1\dot{z}_1^2 + m_3\dot{z}_3^2 + m_5\dot{z}_5^2 + I_1\dot{\theta}_1^2 + I_2\dot{\theta}_2^2 + I_3\dot{\theta}_3^2 + I_4\dot{\theta}_4^2 + I_5\dot{\theta}_5^2 + J_2\dot{\phi}_2^2 + \\ & J_4\dot{\phi}_4^2 + H_2\dot{\xi}_2^2 + H_4\dot{\xi}_4^2 - 2P_{62}\dot{\theta}_2\dot{\xi}_2 - 2P_{67}\dot{\theta}_2\dot{\phi}_2 - 2P_{27}\dot{\xi}_2\dot{\phi}_2 - \\ & 2P_{84}\dot{\theta}_4\dot{\xi}_4 - 2P_{89}\dot{\theta}_4\dot{\phi}_4 - 2P_{49}\dot{\xi}_4\dot{\phi}_4 \}, \end{aligned} \qquad [12]$$

$$
\begin{aligned}
V = \frac{1}{2}\{&k_9[x_3-w\theta_3-L(t-\tau_1)]^2 + k_{10}[x_3+w\theta_3-R(t-\tau_1)]^2 + k_{11}[x_5-w\theta_5-L(t-\tau_2)]^2 + \\
&k_{12}[x_5+w\theta_5-R(t-\tau_2)]^2 + (k_{13}+k_{14})[z_1+y\theta_1]^2 + (k_{17}+k_{18})[z_3+y\theta_3]^2 + \\
&(k_{15}+k_{16})[x_7-z_1+v\theta_1+c\theta_2-g\xi_2]^2 + (k_{19}+k_{20})[x_7-z_3+c\theta_2+v\theta_3+h\xi_2]^2 + \\
&k_1[x_2-x_1+u\theta_1-\theta_2(\tfrac{s+2a}{2})+g\phi_2]^2 + k_2[x_2-x_1-u\theta_1+\theta_2(\tfrac{s-2a}{2})+g\phi_2]^2 + \\
&k_3[x_2-x_3-\theta_2(\tfrac{s+2a}{2})+u\theta_3-h\phi_2]^2 + k_4[x_2-x_3+\theta_2(\tfrac{s-2a}{2})-u\theta_3-h\phi_2]^2 + \\
&k_5[x_2-x_5-a\theta_2-\tfrac{t\theta_4}{2}+u\theta_5-b\phi_2-(e+r)\phi_4]^2 + k_7[x_1-w\theta_1-L(t)]^2 + \\
&k_6[x_2-x_5-a\theta_2+\tfrac{t\theta_4}{2}-u\theta_5-b\phi_2-(e+r)\phi_4]^2 + k_8[x_1+w\theta_1-R(t)]^2 + \\
&(k_{23}+k_{24})[x_7-z_5+p\theta_2+(f-q)\theta_4+v\theta_4+b\xi_2+(e+r)\xi_4]^2 + \\
&(k_{21}+k_{22})[z_5+y\theta_5]^2\},
\end{aligned}
\qquad [13]
$$

$$
\begin{aligned}
D = \frac{1}{2}\{ & c_9[\dot{x}_3-w\dot{\theta}_3-\dot{L}(t-\tau_1)]^2 + c_{10}[\dot{x}_3+w\dot{\theta}_3-\dot{R}(t-\tau_1)]^2 + c_{11}[\dot{x}_5-w\dot{\theta}_5-\dot{L}(t-\tau_2)]^2 + \\
& c_{12}[\dot{x}_5+w\dot{\theta}_5-\dot{R}(t-\tau_2)]^2 + (c_{13}+c_{14})[\dot{z}_1+y\dot{\theta}_1]^2 + (c_{17}+c_{18})[\dot{z}_3+y\dot{\theta}_3]^2 + \\
& (c_{15}+c_{16})[\dot{x}_7-\dot{z}_1+v\dot{\theta}_1+c\dot{\theta}_2-g\dot{\xi}_2]^2 + (c_{19}+c_{20})[\dot{x}_7-\dot{z}_3+c\dot{\theta}_2+v\dot{\theta}_3+h\dot{\xi}_2]^2 + \\
& c_1[\dot{x}_2-\dot{x}_1+u\dot{\theta}_1-\dot{\theta}_2\frac{(s+2a)}{2}+g\dot{\phi}_2]^2 + c_2[\dot{x}_2-\dot{x}_1-u\dot{\theta}_1+\dot{\theta}_2\frac{(s-2a)}{2}+g\dot{\phi}_2]^2 + \\
& c_3[\dot{x}_2-\dot{x}_3-\dot{\theta}_2\frac{(s+2a)}{2}+u\dot{\theta}_3-h\dot{\phi}_2]^2 + c_4[\dot{x}_2-\dot{x}_3+\dot{\theta}_2\frac{(s-2a)}{2}-u\dot{\theta}_3-h\dot{\phi}_2]^2 + \\
& c_5[\dot{x}_2-\dot{x}_5-a\dot{\theta}_2-\frac{t\dot{\theta}_4}{2}+u\dot{\theta}_5-b\dot{\phi}_2-(e+r)\dot{\phi}_4]^2 + c_7[\dot{x}_1-w\dot{\theta}_1-\dot{L}(t)]^2 + \\
& c_6[\dot{x}_2-\dot{x}_5-a\dot{\theta}_2-\frac{t\dot{\theta}_4}{2}-u\dot{\theta}_5-b\dot{\phi}_2-(e+r)\dot{\phi}_4]^2 + c_8[\dot{x}_1+w\dot{\theta}_1-\dot{R}(t)]^2 + \\
& (c_{23}+c_{24})[\dot{x}_7-\dot{z}_5+p\dot{\theta}_2+(f-q)\dot{\theta}_4+v\dot{\theta}_4+b\dot{\xi}_2+(e+r)\dot{\xi}_4]^2 + \\
& (c_{21}+c_{22})[\dot{z}_5+y\dot{\theta}_5]^2 \}. \qquad [14]
\end{aligned}
$$

Generalized Forces

The generalized forces were determined by considering the virtual work done exclusively by the external forces when every coordinate was given an arbitrary virtual displacement. Since the virtual work done by the external forces is given as

$$\delta W_e = \sum_i^n \frac{\partial W_e}{\partial x_i} \delta x_i = \sum_i^n q_i \delta x_i ,$$

the generalized forces can be determined by the relation

$$q_i = \frac{\partial W_e}{\partial x_i} , \qquad (i = 1, 2, \ldots, n).$$

where

q_i = the generalized force associated with the ith coordinate

W_e = total work done by external forces.

When the external forces are due only to a "bumpy" road, the generalized force vector becomes

$$\{Q\} = \begin{Bmatrix} k_7 L(t) + c_7 \dot{L}(t) + k_8 R(t) + c_8 \dot{R}(t) \\ 0 \\ k_9 L(t-\tau_1) + c_9 \dot{L}(t-\tau_1) + k_{10} R(t-\tau_1) + c_{10} \dot{R}(t-\tau_1) \\ k_{11} L(t-\tau_2) + c_{11} \dot{L}(t-\tau_2) + k_{12} R(t-\tau_2) + c_{12} \dot{R}(t-\tau_2) \\ 0 \\ 0 \\ 0 \\ 0 \\ w[-k_7 L(t) - c_7 \dot{L}(t) + k_8 R(t) + c_8 \dot{R}(t)] \\ 0 \\ w[-k_9 L(t-\tau_1) - c_9 \dot{L}(t-\tau_1) + k_{10} R(t-\tau_1) + c_{10} \dot{R}(t-\tau_1)] \\ 0 \\ w[-k_{11} L(t-\tau_2) - c_{11} \dot{L}(t-\tau_2) + k_{12} R(t-\tau_2) + c_{12} \dot{R}(t-\tau_2)] \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix}$$

After the generalized forces and the energy functions were determined, the seventeen equations of motion were written in the concise matrix form

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = \{Q\} \ , \qquad [15]$$

where the mass, damping, and stiffness matrix elements were determined by applying the relationships of [2] to the energy functions [12], [13], and [14]. The respective non-zero matrix elements are given in subroutine **DEFINE**, found in appendix A.

Chapter IV

THEORY OF SOLUTION FOR MATHEMATICAL MODEL

Reduced Equation

The classical method of solving the equations of [15] in the absence of damping is to find the modal columns which satisfy the homogeneous matrix equation

$$[M]\{\ddot{X}\} + [K]\{X\} = \{0\} \, . \qquad [16]$$

However when viscous damping is included, the modal columns for [16] do not in general uncouple the matrix equation

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = \{0\} \, .$$

Thus the primary objective for using normal coordinates is lost. This difficulty can be overcome by introducing an equivalent matrix equation called the reduced equation.

Employing the generalized velocities as auxiliary variables, equation [15] was written in the equivalent form

$$[A]\{\dot{Y}\} + [B]\{Y\} = \{F\} \, , \qquad [17]$$

where

$$[A] = \begin{vmatrix} [C] & [M] \\ [M] & [0] \end{vmatrix} \, ,$$

$$[B] = \left| \begin{array}{cc} [K] & [0] \\ [0] & -[M] \end{array} \right| ,$$

$$\{F\} = \left| \begin{array}{c} \{Q\} \\ \{0\} \end{array} \right| ,$$

$$\{Y\} = \left| \begin{array}{c} \{X\} \\ \{\dot{X}\} \end{array} \right| .$$

Homogeneous Solution

The homogeneous solution of [17] is determined by premultiplying by $-[A]^{-1}$. Then homogeneous [17] becomes

$$-\{\dot{Y}\} + [D]\{Y\} = 0 \quad , \qquad [18]$$

where

$$[D] - [A]^{-1} [B] = \left| \begin{array}{cc} [0] & [I] \\ -[M]^{-1}[K] & -[M]^{-1}[C] \end{array} \right| .$$

After substituting the assumed solution $\{Y\} = \{\upsilon\} e^{\lambda t}$ into [18], the result is

$$[D-\lambda I]\{\upsilon\} = 0 \; . \qquad [19]$$

Equation [19] possesses a non-trivial solution if and only if

$$\det[D-\lambda I] = 0 \; . \qquad [20]$$

This is recognized to be the characteristic equation of the matrix [D]. The 2n(34) roots of [20] are the eigenvalues of the matrix [D]. In general, the eigenvalues occur in complex conjugate pairs with corresponding conjugate eigenvectors or modal columns.

Orthogonality of Modes

Since each of the 2n eigenvalues with corresponding eigenvectors satisfies [19], the ith eigenvalue and associated eigenvector will also satisfy the equation resulting from premutiplying [19] by -[A],

$$\lambda_i[A]\{\upsilon\}_i + [B]\{\upsilon\}_i = \{0\} \; . \tag{21}$$

Since [A] and [B] are known to be symmetric, the jth eigenvalue and corresponding eigenvector are a solution to the transpose of [21],

$$\lambda_j\{\upsilon\}_j^T[A] + \{\upsilon\}_j^T[B] = \{0\} \; . \tag{22}$$

Premultiplying [21] by $\{\upsilon\}_j^T$ and postmultiplying [22] by $\{\upsilon\}_i$ give respectively

$$\lambda_i\{\upsilon\}_j^T[A]\{\upsilon\}_i + \{\upsilon\}_j^T[B]\{\upsilon\}_i = 0, \tag{23}$$

$$\lambda_j\{\upsilon\}_j^T[A]\{\upsilon\}_i + \{\upsilon\}_j^T[B]\{\upsilon\}_i = 0 \; . \tag{24}$$

Subtracting [24] from [23] yeilds

$$(\lambda_i-\lambda_j)\{\upsilon\}_j^T[A]\{\upsilon\}_i = 0 \; . \tag{25}$$

If the eigenvalues are distinct ($\lambda_i \neq \lambda_j$), we obtain the orthogonality relation

$$\{\upsilon\}_j^T[A]\{\upsilon\}_i = 0 \; . \qquad [26]$$

Particular Solution

The particular solution of [17] can be determined by using the eigenvectors and the orthogonality condition. In order to solve [17], we expand $\{Y\}$ into a modal series of the form

$$\{Y\} = \sum_i^{2n} \{\upsilon\}_i z_i(t) \; . \qquad [27]$$

After substituting [27] into [17], we obtain

$$[A]\sum_i^{2n} \{\upsilon\}_i \dot{z}_i(t) + [B]\sum_i^{2n} \{\upsilon\}_i z_i(t) = \{F\} \; . \qquad [28]$$

Premultiplying [28] by $\{\upsilon\}_j^T$ and recalling the orthogonality condition, it is observed that the only term of the series that survives is for $i = j$; thus equation [28] becomes

$$\{\upsilon\}_j^T[A]\{\upsilon\}_j \dot{z}_j(t) + \{\upsilon\}_j^T[B]\{\upsilon\}_j z_j(t) = \{\upsilon\}_j^T\{F\} \; . \qquad [29]$$

Referring back to equation [23], it is seen that by letting $i = j$,

$$\{\upsilon\}_j^T[B]\{\upsilon\}_j = -\lambda_j\{\upsilon\}_j^T[A]\{\upsilon\}_j \; .$$

After replacing this relation into [29], we obtain

$$\{\upsilon\}_j^T[A]\{\upsilon\}_j\left(\dot{z}_j(t)-\lambda_j z_j(t)\right) = \{\upsilon\}_j^T\{F\} \ . \qquad [30]$$

If we require that $\{\upsilon\}_j^T[A]\{\upsilon\}_j = 1$, equation [30] is reduced to the simple form

$$\dot{z}_j(t) - \lambda_j z_j(t) = f_j, \qquad [31]$$

where

$$f_j = \{\upsilon\}_j^T\{F\} \ .$$

Assuming the initial conditions are zero, the solution of [31] can be written in terms of the convolution integral

$$z_j(t) = \int_0^t f_j(\tau)e^{\lambda_j(t-\tau)}d\tau \ .$$

Recall that

$$\{Y(t)\} = \begin{vmatrix} \{X(t)\} \\ \{\dot{X}(t)\} \end{vmatrix} = [T]\{Z(t)\} = \begin{vmatrix} [\Psi] \\ [\lambda\Psi] \end{vmatrix} \{Z(t)\} \ , \qquad [32]$$

where $[\Psi]$ is the upper n x 2n rectangular matrix of the square modal matrix $[T]$.

From [32] it is seen that the displacement vector is given by

$$\{X(t)\} = [\Psi]\{Z(t)\} \ ,$$

$$\{X(t)\} = \sum_{i}^{2n} \{\psi\}_i \int_0^t e^{\lambda_i(t-\tau)} f_i(\tau)d\tau \ , \qquad [33]$$

Note that since

$$f_i(\tau) = \begin{vmatrix} \{\psi\} \\ \{\lambda\psi\} \end{vmatrix}^T \begin{vmatrix} \{Q(\tau)\} \\ \{0\} \end{vmatrix} ,$$

the expression for $f_i(\tau)$ in [33] may be replaced by $\{\psi\}$ $\{Q(\tau)\}$ to give

$$\{X(t)\} = \sum_{i}^{2n} \{\psi\}_i \int_0^t e^{\lambda_i(t-\tau)} \{\psi\}_i^T\{Q(\tau)\}d\tau \ . \qquad [34]$$

If it is assumed that $\{Q(\tau)\}$ is cosinusoidal (sinusoidal) and of the form

$$\{Q(\tau)\} = \mathrm{Re} \begin{vmatrix} \sum_{k}^{s} \{C\}_k e^{jw_k t} \end{vmatrix} ,$$

where

$\{C\}_k$ = n x 1 column vector of complex constants giving phase information

w_k = one of the S forcing frequencies

j = $\sqrt{-1}$

then [34] becomes

$$X(t) = \sum_{i}^{2n} \{\psi\}_i \int_0^t \{\psi\}_i^T \Big(\sum_k^s \{C\}_k e^{jw_k\tau}\Big) e^{\lambda_i(t-\tau)} d\tau$$

$$= \sum_i^{2n} \{\psi\}_i \{\psi\}_i^T \Big(\sum_k^s \{C\}_k \int_0^t e^{\tau(-\lambda_i+jw_k)+\lambda_i t} d\tau\Big) \ . \qquad [35]$$

The solution of [35] may be readily determined by taking the general rth factor of the sum for {Q} and integrating the definite integral

$$\{C\}_r \int_0^t e^{\tau(-\lambda_i+jw_r)+\lambda_i t} d\tau \ ,$$

$$= \{C\}_r \frac{1}{-\lambda_i+jw_r} e^{\tau(-\lambda_i+jw_r)+\lambda_i t} \Bigg|_0^t \ ,$$

$$= \{C\}_r \frac{(e^{jw_r t} - e^{\lambda_i t})}{-\lambda_i+jw_r} \ . \qquad [36]$$

From this result it is seen that the solution of [35] is

$$\{X(t)\} = \mathrm{Re} \left| \sum_i^{2n} \{\psi\}_i \{\psi\}_i^T \sum_k^s \{C\}_k \frac{(e^{jw_k t} - e^{\lambda_i t})}{-\lambda_i+jw_k} \right| \ . \qquad [37]$$

When only the steady state solution is desired, the second exponential term of [37] is recognized to be the transient response for the assumed zero initial conditions. Therefore, given sufficient time,the contribution of

$e^{\lambda_i t}$ in [37] may be ignored; thus the steady state response for sinusoidal inputs is given by

$$\{X(t)\} = \mathrm{Re}\left| \sum_i^{2n} \{\psi\}_i \{\psi\}_i^T \sum_k^s \{C\}_k \frac{e^{jw_k t}}{-\lambda_i + jw_k} \right| . \qquad [38]$$

Alternate Determination of Particular Solution

The particular solution for the equations of motion

$$[M]\{\ddot{X}\} + [C]\{\dot{X}\} + [K]\{X\} = \{Q\} , \qquad [39]$$

may be solved by the method of undetermined coefficients. When $\{Q\} = \{q_1\}\cos\beta + \{q_2\}\sin\beta$ (where β is a linear function of time), assume that

$$\{X\} = \{E\}\cos\beta + \{F\}\sin\beta . \qquad [40]$$

After substituting the respective time derivatives of [40] into [39] and equating the sine and cosine terms, the following equations result:

$$(-\dot{\beta}^2[M]\{E\} + \dot{\beta}[C]\{F\} + [K]\{E\})\cos\beta = \{q_1\}\cos\beta ,$$

$$(-\dot{\beta}^2[M]\{F\} - \dot{\beta}[C]\{E\} + [K]\{F\})\sin\beta = \{q_2\}\sin\beta .$$

After cancelling $\cos\beta$ and $\sin\beta$ from the above equations, the equations may be written in the equivalent partitioned matrix form as

$$\left|\begin{array}{c|c} -\dot{\beta}^2[M]+[K] & \dot{\beta}[C] \\ \hline -\dot{\beta}[C] & -\dot{\beta}^2[M]+[K] \end{array}\right| \left|\begin{array}{c} \{E\} \\ \hline \{F\} \end{array}\right| = \left|\begin{array}{c} \{q_1\} \\ \hline \{q_2\} \end{array}\right| . \qquad [41]$$

Since the tires on a given side experience the identical sinusoidal bump, the vehicle response for a single input at the respective tires may be determined simultaneously by altering [41] to the form

$$\left|\begin{array}{c|c} -\omega^2[M]+[K] & \omega[C] \\ \hline -\omega[C] & -\omega^2[M]+[K] \end{array}\right| \left|\begin{array}{c|c|c} E_1 & E_2 & E_3 \\ \hline F_1 & F_2 & F_3 \end{array}\right|$$

$$= \left|\begin{array}{c|c|c} q_{11} & q_{12} & q_{13} \\ \hline q_{21} & q_{22} & q_{23} \end{array}\right| , \qquad [42]$$

where ω is the common forcing frequency for a given side of the vehicle.

Assuming the equations of [42] represent forcing of the left side, the solution of the 2n algebraic equations gives the response of the vehicle as

$$\{X\} = \{E_1\} \cos \omega t + \{F_1\} \sin \omega t + \{E_2\} \cos (\omega t-\tau_1) +$$

$$\{F_2\} \sin (\omega t-\tau_1) + \{E_3\} \cos (\omega t-\tau_2) + \{F_3\} \sin (\omega t-\tau_2). \qquad [43]$$

Utilizing the trigonometric identities

$$\sin (y-z) = \sin y \cos z - \cos y \sin z,$$

$$\cos (y-z) = \cos y \cos z + \sin y \sin z,$$

and defining

$$\{R\} = \{E_1\} + \{E_2\} \cos \tau_1 - \{F_2\} \sin \tau_1 + \{E_3\} \cos \tau_2 - \{F_3\} \sin \tau_2,$$

$$\{S\} = \{F_1\} + \{E_2\} \sin \tau_1 + \{F_2\} \cos \tau_1 + \{E_3\} \sin \tau_2 + \{F_3\} \cos \tau_2,$$

it can be shown that [43] is equivalent to

$$\{X\} = \{R\} \cos \omega t + \{S\} \sin \omega t. \tag{44}$$

Following the same procedure as given above, the response due to forcing of the right side can be determined. Adding these results gives the total response of the vehicle as

$$\{X\} = \{R\}_\ell \cos \omega_\ell t + \{S\}_\ell \sin \omega_\ell t + \{R\}_r \cos \omega_r t + \{S\}_r \sin \omega_r t, \tag{45}$$

where the subscripts indicate the respective sides of the vehicle.

Chapter V

DIGITAL COMPUTER SOLUTIONS

Two major digital computer programs were developed to obtain the particular solutions for the semi-trailer truck. The source listings given in appendix A and B are the programs corresponding to the reduced equation method and the undetermined coefficients method discussed in chapter IV. The following paragraphs include a discussion of the output for the two programs along with a brief description of the numerical methods employed.

Reduced Equations Program

The vehicle parameters (springs, dampers, masses and characteristic lengths) are read into the program under namelist format control and then printed out as shown in figure 10. After the mass, damping and stiffness matrix elements are computed in subroutine DEFINE, the undamped natural frequencies and damped eigenvalues are determined, using the method of Francis (5), and printed out as given in figures 11 and 12.

Next the complex conjugate eigenvectors are determined by inverse iteration, as explained by Wilkinson (20), and printed out as shown in figure 13 (note the eigenvectors are normalized as required in equation [31]).

Subroutine RIDE determines the sinusoidal response for the vehicle due to the sinusoidal forcing functions at the six tires, which were calculated in subroutine ROADIN. The sinusoidal road characteristics are explained on the next page, along with the response vectors as displayed in figure 14.

The final page of output, as indicated in figure 15, gives the maximum amplitudes of all the independent coordinates for the given road input.

FOR THIS SEMI-TRAILER TRUCK THE CHARACTERIZING
PARAMETERS ARE AS FOLLOWS.

DAMPERS (KIP-SEC/FT)	SPRINGS (KIP/FT)	DIMENSIONAL LENGTHS (FT)	MASS MOMENTS OF INERTIA (KIP-SEC**2-FT)	MASSES (KIP-SEC**2/FT)
C1= 0.780000E 00	K1= 0.198000E 02	A= 0.0	MI1= 0.488000E 00	M1= 0.488000E-01
C2= 0.780000E 00	K2= 0.198000E 02	B= 0.648400E 01	MI2= 0.238000E 01	M2= 0.258000E 00
C3= 0.240000E 00	K3= 0.150000E 03	CC= 0.250000E 01	MI3= 0.149000E 01	M3= 0.149000E 00
C4= 0.240000E 00	K4= 0.150000E 03	D= 0.0	MI4= 0.512000E 02	M4= 0.171000E 01
C5= 0.350000E 00	K5= 0.198000E 03	E= 0.170000E 02	MI5= 0.127000E 01	M5= 0.127000E 00
C6= 0.350000E 00	K6= 0.198000E 03	F= 0.350000E 01	MJ2= 0.476000E 01	M6= 0.455800E 00
C7= 0.240000E-01	K7= 0.210000E 02	G= 0.376600E 01	MJ4= 0.149900E 03	M7= 0.258000E 00
C8= 0.240000E-01	K8= 0.210000E 02	H= 0.740000E 01	MH2= 0.400000E 01	M8= 0.183700E 01
C9= 0.960000E-01	K9= 0.170000E 03	P= 0.108400E 01	MH4= 0.151200E 03	M9= 0.171000E 01
C10= 0.960000E-01	K10= 0.170000E 03	Q= 0.300000E 01	P67= 0.0	
C11= 0.120000E 00	K11= 0.210000E 03	R= 0.116700E 02	P62= 0.0	
C12= 0.120000E 00	K12= 0.210000E 03	S= 0.350000E 01	P27= 0.0	
C13= 0.480000E-01	K13= 0.150000E 02	T= 0.350000E 01	P84= 0.0	
C14= 0.480000E-01	K14= 0.150000E 02	U= 0.175000E 01	P89= 0.0	
C15= 0.0	K15= 0.120000E 02	V= 0.750000E 00	P49= 0.0	
C16= 0.0	K16= 0.120000E 02	W= 0.295800E 01		
C17= 0.150000E 00	K17= 0.102000E 03	X= 0.0		
C18= 0.150000E 00	K18= 0.102000E 03	Y= 0.168500E 01		
C19= 0.0	K19= 0.187000E 02	Z= 0.0		
C20= 0.0	K20= 0.187000E 02			
C21= 0.200000E 00	K21= 0.126000E 03			
C22= 0.200000E 00	K22= 0.126000E 03			
C23= 0.0	K23= 0.250000E 02			
C24= 0.0	K24= 0.250000E 02			

Figure 10: Characterizing vehicle parameters.

THE N UNDAMPED CIRCULAR FREQUENCIES ARE

3.3101152642670590 00

6.7264320123992620 00

8.5160046482790790 00

1.0006333740397800 01

1.0894043010320810 01

1.4678983368547670 01

2.0289691539254860 01

2.1134395554377660 01

3.2176894694621080 01

3.7028777073426760 01

3.7809286794358090 01

4.2580381661623450 01

4.4859043734842520 01

5.7976935911773220 01

6.8485182855290340 01

6.9426918030037560 01

8.2025762401394860 01

THE ERROR IN THE EIGENVALUES IS COMPARABLE TO
1.0482861823122200-11

Figure 11: Undamped **natural frequencies.**

THE N DAMPED,COMPLEX CONJUGATE EIGENVALUE PAIRS ARE

REAL PART	IMAGINARY PART
-6.977365367288358D-03	3.310119819904249D 00
-6.977365367288358D-03	-3.310119819904249D 00
-2.433228024426425D-02	6.726678153693640D 00
-2.433228024426425D-02	-6.726678153693640D 00
-2.620388770238901D-02	8.516819042061824D 00
-2.620388770238901D-02	-8.516819042061824D 00
-8.605615222299350D-02	1.001290943392688D 01
-8.605615222299350D-02	-1.001290943392688D 01
-9.924787884981981D-01	1.110775877476442D 01
-9.924787884981981D-01	-1.110775877476442D 01
-1.865423224962665D-01	1.468319147763928D 01
-1.865423224962665D-01	-1.468319147763928D 01
-1.048229249213930D 00	2.044177285985040D 01
-1.048229249213930D 00	-2.044177285985040D 01
-2.475289146000497D-01	2.114228640837730D 01
-2.475289146000497D-01	-2.114228640837730D 01
-2.194299370101939D 00	3.311275532378840D 01
-2.194299370101939D 00	-3.311275532378840D 01
-4.360803660253532D 00	3.594218922000895D 01
-4.360803660253532D 00	-3.594218922000895D 01
-2.064598298907953D 01	3.608972855999870D 01
-2.064598298907953D 01	-3.608972855999870D 01
-7.228919689638735D-01	3.713737221178032D 01
-7.228919689638735D-01	-3.713737221178032D 01
-1.063615812073526D 00	4.485141758916870D 01
-1.063615812073526D 00	-4.485141758916870D 01
-1.769412451678008D 00	5.793561668086176D 01
-1.769412451678008D 00	-5.793561668086176D 01
-2.690821992762824D 00	6.840986622337274D 01
-2.690821992762824D 00	-6.840986622337274D 01
-2.623891127336520D 00	6.936841336596851D 01
-2.623891127336520D 00	-6.936841336596851D 01
-4.064845992317023D 00	8.191993220959171D 01
-4.064845992317023D 00	-8.191993220959171D 01

THE ERROR IN THE EIGENVALUES IS COMPARABLE TO
1.048288804625388D-13

Figure 12: Damped eigenvalues.

THE DAMPED EIGENVALUES AND THEIR ASSOCIATED EIGENVECTORS ARE

-6.977365367288358D-03	3.310119819904249D 00	-6.977365367288358D-03	-3.310119819904249D 00
	1		2
-3.857310191198520D-16	2.889453777045809D-16	-3.857310191198520D-16	-2.889453777045809D-16
-2.635082338230157D-15	3.603405245093768D-16	-2.635082338230157D-15	-3.603405245093768D-16
-1.237190002958478D-15	1.618124432387662D-15	-1.237190002958478D-15	-1.618124432387662D-15
-2.792515701650395D-15	3.690683411937736D-15	-2.792515701650395D-15	-3.690683411937736D-15
-2.742353756977890D-02	2.728199854282866D-02	-2.742353756977890D-02	-2.728199854282866D-02
-6.752793441775530D-03	6.639989073692518D-03	-6.752793441775530D-03	-6.639989073692518D-03
-6.963558947352552D-03	6.969373357444960D-03	-6.963558947352552D-03	-6.969373357444960D-03
-1.987923675324778D-02	1.982098435020087D-02	-1.987923675324778D-02	-1.982098435020087D-02
1.439796670243658D-03	-1.348200883482611D-03	1.439796670243658D-03	1.348200883482611D-03
2.931569626366941D-03	-2.971102975522255D-03	2.931569626366941D-03	2.971102975522255D-03
1.305862643782849D-03	-1.308959528273037D-03	1.305862643782849D-03	1.308959528273037D-03
2.806265752596438D-02	-2.809890927953280D-02	2.806265752596438D-02	2.809890927953280D-02
7.840970224067032D-03	-7.804709946897892D-03	7.840970224067032D-03	7.804709946897892D-03
-1.756311260106459D-17	-4.467990146906618D-16	-1.756311260106459D-17	4.467990146906618D-16
8.758677502703067D-17	-1.859337963008272D-16	8.758677502703067D-17	1.859337963008272D-16
-1.844637369207446D-03	1.844787895679693D-03	-1.844637369207446D-03	-1.844787895679693D-03
-1.283864536494062D-03	1.276314321459505D-03	-1.283864536494062D-03	-1.276314321459505D-03
1.026149675538217D-14	1.420563311001877D-14	1.026149675538217D-14	-1.420563311001877D-14
1.004086481089147D-14	6.759512576679273D-15	1.004086481089147D-14	-6.759512576679273D-15
5.867695762699617D-15	1.137792768387593D-14	5.867695762699617D-15	-1.137792768387593D-14
-9.818707177647042D-16	6.215152260983815D-15	-9.818707177647042D-16	-6.215152260983815D-15
-9.011534006191690D-02	-9.096555171338160D-02	-9.011534006191690D-02	9.096555171338160D-02
-2.193204272967295D-02	-2.239888504112708D-02	-2.193204272967295D-02	2.239888504112708D-02
-2.302087358774764D-02	-2.309884235298370D-02	-2.302087358774764D-02	2.309884235298370D-02
-6.547112844955040D-02	-6.594095383122970D-02	-6.547112844955040D-02	6.594095383122970D-02
4.452660478216533D-03	4.775306384973719D-03	4.452660478216533D-03	-4.775306384973719D-03
9.814252213881111D-03	9.724577194686047D-03	9.814252213881111D-03	-9.724577194686047D-03
4.323701397215145D-03	4.331694908153410D-03	4.323701397215145D-03	-4.331694908153410D-03
9.281495310914840D-02	9.308681523236090D-02	9.281495310914840D-02	-9.308681523236090D-02
2.577981576975443D-02	2.600900725886404D-02	2.577981576975443D-02	-2.600900725886404D-02
1.269433000903279D-14	1.542944655188194D-14	1.269433000903279D-14	-1.542944655188194D-14
1.183010121013263D-14	1.577568512704007D-14	1.183010121013263D-14	-1.577568512704007D-14
-6.093598268102474D-03	-6.118842475507425D-03	-6.093598268102474D-03	6.118842475507425D-03
-4.215795339926313D-03	-4.258650759650020D-03	-4.215795339926313D-03	4.258650759650020D-03

Figure 13: Damped eigenvalues with associated eigenvectors.

THE SPEED (FEET/SECOND) OF THIS VEHICLE IS EQUAL TO 8.8000D 01

THE SINUSOIDAL FORCING FREQUENCIES (RADIANS/SEC) FOR THE LEFT AND FOR THE RIGHT SIDE WHEEL PATHS ARE RESPECTIVELY

WL= 5.00C000000000000D 01 WR= 5.000000000000000D 01

THE AMPLITUDE (FEET) OF THE LEFT AND RIGHT SINUSOIDAL PATHS ARE RESPECTIVELY

8.3500D-02 8.3500D-02

THE WAVELENGTH OF THE LEFT AND RIGHT SINUSOIDAL DISPLACEMENT FUNCTIONS ARE RESPECTIVELY

1.1058406141D 01 1.1058406141D 01

OUTPUT(I) = A(I)*COS(WL*T) + B(I)*SIN(WL*T) + C(I)*COS(WR*T) + D(I)*SIN(WR*T)

A(I)	B(I)	C(I)	D(I)
-1.744000365705763D-02	-4.339015827039611D-03	-1.744000365741596D-02	-4.339015826861528D-03
2.381962320684179D-03	-5.36C6556013424730-03	2.381962320829623D-03	-5.360655601772739D-03
-6.265945881403326D-03	3.062926885515566D-02	-6.265945884112187D-03	3.062926887304442D-02
5.676204177530949D-03	-2.429509768516472D-02	5.676204177995454D-03	-2.429509768736524D-02
1.308015072352829D-04	7.498904524038415D-05	-1.308015070742343D-04	-7.498904502748641D-05
2.381032953616552D-03	-2.13C683702380996D-03	-2.381032955737563D-03	2.130683701562014D-03
5.442037268143397D-03	-5.93C727459877562D-02	-5.442037268517864D-03	5.930727460002120D-02
1.026682145146584D-02	1.066920304009004D-01	-1.026682145055838D-02	-1.066920304077898D-01
1.555864835898250D-03	6.443380996937633D-03	-1.555864835847797D-03	-6.443380996516457D-03
-9.823794764912474D-04	5.031375046708742D-03	9.823794763404316D-04	-5.031375046684189D-03
6.743030194585087D-03	-2.259283226852094D-02	-6.743030194623788D-03	2.259283226855555D-02
2.302108719222561D-05	-1.326618367627412D-04	-2.302108720501149D-05	1.326618368578408D-04
-5.465034784821347D-03	3.76C031774432717D-03	5.465034785412661D-03	-3.760031780045336D-03
1.950767615930420D-04	1.873147234470206D-03	1.950767615707538D-04	1.873147234972687D-03
7.454256192661015D-05	-1.055522145957144D-03	7.454256194256149D-05	-1.055522146052544D-03
1.502103052454132D-04	-2.978774257737958D-04	-1.502103052697328D-04	2.978774258088805D-04
-1.170048031672156D-05	-2.761519904834992D-04	1.170048031141830D-05	2.761519905601393D-04

Figure 14: Vehicle response vectors.

THE MAXIMUM AMPLITUDES ARE

THE RELATIVE PHASE ANGLE IS

0.0

X1	=	3.59D-02
X2	=	1.17D-02
X3	=	6.25D-02
X5	=	4.99D-02
X7	=	2.67D-13
Z1	=	2.27D-12
Z3	=	1.30D-12
Z5	=	6.95D-12
THETA1	=	4.24D-13
THETA2	=	1.53D-13
THETA3	=	5.19D-14
THETA4	=	9.60D-14
THETA5	=	5.64D-12
PHI2	=	3.77D-03
PHI4	=	2.12D-03
XI2	=	4.27D-14
XI4	=	7.68D-14

Figure 15: Maximum amplitudes for the coordinates.

Undetermined Coefficients Program

Except for the absence of the eigenvalues and eigenvectors, the output information and format for this program are identical to those of the previous program. However the numerical algorithums used are vastly different. Although the merits of the two programs will be discussed in the next chapter, it is of interest here to compare the solutions determined by this method, shown in figure 16, to those shown in figure 15. It is seen that the results are in close agreement.

```
THE MAXIMUM AMPLITUDES (FEET,RADIANS) ARE

THE RELATIVE PHASE ANGLE(RAD.) BETWEEN SIDES IS
          0.0

          X1    =  3.590-02
          X2    =  1.170-02
          X3    =  6.250-02
          X5    =  4.990-02
          X7    =  0.0
          Z1    =  0.0
          Z3    =  0.0
          Z5    =  0.0
      THETA1    =  0.0
      THETA2    =  0.0
      THETA3    =  0.0
      THETA4    =  0.0
      THETA5    =  0.0
        PHI2    =  3.770-03
        PHI4    =  2.120-03
         XI2    =  0.0
         XI4    =  0.0
```

Figure 16: Maximum amplitudes for the coordinates using the method of undetermined coefficients.

Chapter VI

RECOMMENDATIONS AND CONCLUSIONS

Recommendations

The scope of this thesis was to model a linear, three-dimensional semi-trailer truck. This object is a necessary first step for a better understanding of the vehicle's vibrations, but does not represent an end in itself. Many areas of analysis need to be initiated for a more complete understanding of the vehicle's vibrational characteristics. Using this model, a number of analyses could be effectively investigated. These are discussed in the following paragraphs.

The steady state solutions presented in this paper were used to verify the correctness of the mathematical model, not for a steady state analysis. However a comprehensive steady state analysis could be completed using this three-dimensional model. Bode plots similar to those presented by Potts (14) could be obtained for this three dimensional vehicle with few additions to the current programs.

The rolling motions experienced on asphalt highways due to phased sinusoidal wave forms could be investigated. The effect of different road frequencies experienced by the left and right sides of the vehicle could also be studied with small modifications to the present programs. Bode plots of the above phenomenon, considered separately or combined, would give a vivid display of the vehicle's response.

The consequences of crosswinds on the vehicle's motion could be explored by adding equivalent sinusoidal forcing functions to the existing programs. Dugoff (2) examined the lateral stability of highway vehicles due to aerodynamic forces, but did not include vehicular vibrations. With the inclusion of vehicular vibrations, a more comprehensive study would result.

For the particular solutions given in chapter V, it was assumed that the road bumps force the vehicle in the vertical direction only. However it is obvious that road bumps also force the tires of the vehicle in the longitudinal and transverse directions. After an appropriate "bump" model is determined, the forces on the vehicle's tires could be readily incorporated into this three dimensional vehicle by adding equivalent forcing functions.

The response of the vehicle to sudden irregularities in a road, such as chuck holes, could be determined with a transient analysis.

Since road profiles represent a random phenomenon, a random vibrational analysis might give the most practical information about the vehicle's motions. Using the set of eigenvalues and eigenvectors determined in the reduced equation program, Lancaster (19) indicates how the response of the system forced randomly may be determined by the simple relation

$$\{X\} = \sum_{i}^{2n} \frac{\{\psi\}_i^T\{Q\}\{\psi\}_i}{jw-\lambda_i}$$

where $\{Q\}$ is the power spectra for the generalized forces.

Conclusions

When deciding between the two programs, the number of frequencies investigated must be considered. The computing time required by the undetermined coefficients program if far less for up to twenty separate solutions. For investigations which exceed thirty solutions, the reduced equation program becomes superior.

Before using the undetermined coefficients program, the great flexibility of the reduced equation program should be considered. A given model is numerically characterized by its set of eigenvalues and eigenvectors determined in the reduced equation program. After they are determined, the set could be stored indefinitely and used repeatedly. Large numbers of particular solutions could be rapidly obtained by matrix multiplications, as opposed to solutions obtained by solving linear equations used in the undetermined coefficients program. As indicated in the recommendations, the reduced equation program can be used in transient and random vibrational analyses.

The three-dimensional model for the vibrations of a semi-trailer truck presented in this thesis is the most comprehensive model known to this author and will add to the understanding of the vibrations of semi-trailer trucks. However before concrete conclusions can be drawn from its use, the accuracy of the model must be determined by comparing it to "real" vehicles.

SELECTED BIBLIOGRAPHY

1. Bellman, Richard. Introduction to Matrix Analysis. New York: McGraw-Hill Book Company, 1970.

2. Dugoff, Howard. "On the Influence of Aerodynamic Forces and Moments on the Lateral Stability of Articulated Highway Vehicles." Presented at First Internl. Conf. on Vehicle Mechanics, Wayne State University, Detroit, Mich., July 16-18, 1968.

3. Duncan, W.J., and Collar, A.R. "Matrices applied to the Motions of Damped Systems." The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science, Vol. 19, No. 125, pp. 197-219, February, 1935.

4. Foss, K. A. "Co-Ordinates Which Uncouple the Equations of Motion of Damped Linear Dynamic Systems." Journal of Applied Mechanics, ASME, pp. 361-364, Paper no. 57-A-86, September, 1958.

5. Francis, J. G. F. "The QR Transformation Unitary Analogue to the LR Transformation, Parts I & II." The Computer Journal, Vol. 4, pp. 265-271, 1961-62.

6. Hammarling, S.J. Latent Roots and Latent Vectors. Toronto: The University of Toronto Press, 1970.

7. Harris, C.M., and Crede, C.E. Shock and Vibrations Handbook, Vol. 3., New York: McGraw-Hill Book Company, Inc., 1961.

8. Janeway, R.N. "A Better Truck Ride for Driver and Cargo." Soc. Auto Enggs. Report no. SP-154, 1958.

9. Lancaster, Peter. Lambda-Matrices and Vibrating Systems. Oxford: Pergamon Press, 1966.

10. Langhaar, Henry L. Energy Methods in Applied Mechanics. New York: John Wiley and Sons, Inc., 1962.

11. LeFevre, W.F. Truck Ride Guide. Detroit: Rockwell-Standard Corporation, 1967.

12. Meirovitch, Leonard. Analytical Methods in Vibrations. New York: The Macmillan Company, 1967.

13. Pipes, Louis A. Matrix Methods for Engineering. Englewood Cliffs: Prentice-Hall, Inc., 1963.

14. Potts, G.R. "Deterministic and Random Linear Theory Compared to Deterministic Nonlinear Theory and Experimental Results of the Vibrations of a Three-Axle Semi-Trailer Truck." A Dissertation, Kansas State University, 1970.

15. Potts, G.R. "Interim Report of a Vibration Analysis of a 3-Axle, Fully Articulated Highway Vehicle." A Master's Report, Kansas State University, 1969.

16. Sharp, R.S., and Goodall, J.R. "A Mathematical Model for the Simulation of Vehicle Motions." Journal of Engineering Mathematics, Vo. 3, No. 3, July 1969.

17. Vaverek, J.C. "Tractor-Trailer Ride on a Chassis Dynamometer." Soc. Auto Enggs. Paper no. 660143. Presented at Internl. Auto. Engg. Congress, Detroit, Mich., January 10-14, 1966.

18. Walther, Wm.D., Gossard, D., and Fensel, P. "Truck Ride---A Mathematical and Empirical Study." Soc. Auto Enggs. Paper no. 690099. Presented at Internatl. Auto. Engg. Congress, Detroit, Mich., January 13-17, 1969.

19. Wells, D.A. Theory and Problems of Lagrangian Dynamics. New York: McGraw-Hill Book Company, 1967.

20. Wilkinson, J.H. The Algebraic Eigenvalue Problem. Oxford: Oxford University Press, 1965.

Appendix A

EXPLANATION OF INPUT DATA FOR REDUCED EQUATION PROGRAM WITH SOURCE LISTING

All input data were read into the program under namelist format control. The values for the springs, dampers, masses, mass moments of inertia, and characteristic lengths were given in namelist data groups SPRING, DAMPER, MASSES, INERTA, and DIMEN respectively. The numerical suffixes of these REAL*4, fortran variables correspond to the subscripted parameters defined in the nomenclature (e.g. K11 = k_{11}). In namelist data group ROAD, the spring and damping constants for the vehicle tires were coded into REAL*8, fortran variables by using roman numerals for subscripts (e.g. KVII = k_7). The remaining REAL*8, input variables are explained below.

AL = amplitude of left side sinusoidal road (feet)

AR = amplitude of right side sinusoidal road (feet)

ZWIDE = (COMPLEX*16) distance from axle center of mass to equivalent tire attachment point (feet)

BT = the lag angle of the right sinusoidal road relative to the left sinusoidal road (radians)

GH1 = longitudinal distance from tractor front axle to tractor rear axle (= g+h, feet)

GBER2 = longitudinal distance from tractor front axle to trailer axle (= g + b + e + r, feet)

SPEED = vehicle speed (feet/second)

FWL = left sinusoidal frequency (radians/second)

FWR = right sinusoidal frequency (radians/second)

```
      IMPLICIT REAL*8 (A-H,O-Y),COMPLEX *16(Z)
      COMPLEX*16 DCMPLX,CDSQRT,DCONJG
      COMMON/COMX/ZVEC(35,34)
      COMMON/DYNAM/D(34,34),VALU(34,2)
      COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
      COMMON/A1/ DL(17,17),AMA(17,17)
      COMMON/ZRIDE/ZRIDL(17),ZRIDR(17)
      COMMON/ZROAD/ZCL(6),ZCR(6)
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      COMMON/RIPPLE/WAVEL,WAVER,FAST,AMPL,AMPR,FAZE
      CALL DEFINE
      DO 2 I=1,N
      DO 2 J=1,N
    2 AMA(J,I)=AM(J,I)
      CALL MINV(AM,N,NN)
      CALL GMPRD(AM,AK,DL,N,N,N,NN,NN,NN)
      DO 1 I=1,N
      LLR=I+N
      DO 1 J=1,N
    1 D(LLR,J)=-DL(I,J)*1.D-2
      CALL FRANCI(DL,VALU,N,ANORM,N)
      CALL ARANGE(VALU,N,NMO,NPO,1)
      ERR=1.110223D-16*ANORM
      WRITE(NWRITE,7) ERR
C
C
      CALL GMPRD(AM,AC,DL,N,N,N,NN,NN,NN)
      DO 11 I=1,N
      LRR=I+N
      DO 11 J=1,N
      LRC=J+N
   11 D(LRR,LRC)=-DL(I,J)*1.D-2
      DO 22 I=1,N
      IPNC=I+N
   22 D(I,IPNC)=1.D-2
C
C     DETERMINE DAMPED EIGENVALUES AND EIGENVECTORS
C
      CALL FRANCI(D,VALU,NT,ANORM,NT)
      DO 300 I=1,NT
      VALU(I,1)=VALU(I,1)*1.D2
  300 VALU(I,2)=VALU(I,2)*1.D2
      CALL ARANGE( VALU,NT,NTMO,NTPO,3)
      ERR=1.110223D-16*ANORM
      WRITE(NWRITE,7) ERR
      DO 43 J=1,NT,2
      DO 33 I=1,NT
   33 ZVEC(I,J)=(1.DO,1.DO)
      ZVEC(35,J)=DCMPLX(VALU(J,1),VALU(J,2))
   43 ZVEC(35,J+1)=DCONJG(ZVEC(35,J))
      CALL VECITR
      CALL BIORTH
      WRITE(NWRITE,155)
  155 FORMAT('1',35X,'THE DAMPED EIGENVALUES AND THEIR ASSOCIATED'
     1,1X,'EIGENVECTORS ARE')
      DO 133 J=1,NT,2
      WRITE(NWRITE,144) ZVEC(NT+1,J),ZVEC(NT+1,J+1)
```

```
 144 FORMAT('0',10X,1P2D25.15,10X,1P2D25.15)
     WRITE(NWRITE,146)
     JPO=J+1
     WRITE(NWRITE,1122) J,JPO
1122 FORMAT('0',38X,I2,58X,I2//)
     DO 134 M=1,NT
 134 WRITE(NWRITE,145) ZVEC(M,J),ZVEC(M,J+1)
 145 FORMAT(' ',10X,1P2D25.15,10X,1P2D25.15)
 133 WRITE(NWRITE,147)
 147 FORMAT('1')
 146 FORMAT(////)
   7 FORMAT('-',42X,'THE ERROR IN THE EIGENVALUES IS COMPARABLE TO'
    1/' ',54X,1PD24.15)
     NOREAD=1
     CALL ROADIN(ZWL,ZWR,WL,WR,NOREAD)
     CALL RIDE(ZWR,ZWL,NT,N)
     WRITE(NWRITE,179)
 179 FORMAT('1')
     WRITE(NWRITE,210) FAST
     WRITE(NWRITE,190)
     WRITE(NWRITE,178) WL,WR
     WRITE(NWRITE,3000) AMPL,AMPR
3000 FORMAT('-',25X,'THE AMPLITUDE (FEET) OF THE LEFT AND RIGHT'
    1,1X,'SINUSOIDAL PATHS ARE RESPECTIVELY'/'0',47X,1P2D15.4)
 190 FORMAT('-',10X,'THE SINUSOIDAL FORCING FREQUENCIES (RADIANS/SEC)'
    1,1X,'FOR THE LEFT AND FOR THE RIGHT SIDE WHEEL PATHS ARE'
    2,1X,'RESPECTIVELY')
 210 FORMAT(' ',30X,'THE SPEED (FEET/SECOND) OF THIS VEHICLE'
    1,1X,'IS EQUAL TO',1P1D12.4)
 178 FORMAT('0',35X,'WL=',1P1D22.15,10X,'WR=',1P1D22.15)
     WRITE(NWRITE,4) WAVEL,WAVER
   4 FORMAT('-',25X,'THE WAVELENGTH OF THE LEFT AND RIGHT SINUSOIDAL'
    1,1X,'DISPLACEMENT FUNCTIONS ARE RESPECTIVELY'/'0',40X,1P2D25.10)
     WRITE(NWRITE,177)
 177 FORMAT('0',25X,'OUTPUT(I) = A(I)*COS(WL*T) + B(I)*SIN(WL*T) + '
    1,1X,'C(I)*COS(WR*T) + C(I)*SIN(WR*T)')
 180 FORMAT('0',1P4D30.15)
     WRITE(NWRITE,181)
 181 FORMAT('-',17X,'A(I)',26X,'B(I)',26X,'C(I)',26X'D(I)')
     DO 149 I=1,N
 149 WRITE(NWRITE,180) ZRIDL(I),ZRIDR(I)
     DIFFER=DABS(WL-WR)
     IF(DIFFER.LE.1.D-7)  CALL WEQULZ
     STOP
     END
```

```
      SUBROUTINE DEFINE
      REAL*8 M,C,K
      REAL *4 M1,M2,M3,M4,M5,M6,M7,M8,M9
      REAL *4 MI1,MI2,MI3,MI4,MI5,MJ2,MJ4,MH2,MH4,P67,P62,P27,P84,P89,
     UP49
      REAL *4 K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16
      REAL *4 K17,K18,K19,K20,K21,K22,K23,K24
      COMMON/INOUT/NREAC,NWRITE,NPUNCH
      COMMON/MATMCK/M(17,17),C(17,17),K(17,17)
      NAMELIST/MASSES/M1,M2,M3,M4,M5,M6,M7,M8,M9
      NAMELIST/INERTA/MI1,MI2,MI3,MI4,MI5,MJ2,MJ4,MH2,MH4,
     1P67,P62,P27,P84,P89,P49
      NAMELIST/DAMPER/C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15
     1,C16,C17,C18,C19,C20,C21,C22,C23,C24
      NAMELIST/SPRING/K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15
     1,K16,K17,K18,K19,K20,K21,K22,K23,K24
      NAMELIST/DIMEN/A,B,CC,D,E,F,G,H,P,Q,R,S,T,U,V,W,X,Y,Z
      READ(NREAD,MASSES)
      READ(NREAD,INERTA)
      READ(NREAD,DAMPER)
      READ(NREAD,SPRING)
      READ(NREAD,DIMEN)
   30 FORMAT('1',T43,'FCR THIS SEMI-TRAILER TRUCK THE CHARACTERIZING'
     1/' ',T53,'PARAMETERS ARE AS FOLLOWS.'////)
   40 FORMAT(' ',T14,'DAMPERS',T40,'SPRINGS',T60,'DIMENSIONAL LENGTHS'
     1,T87,'MASS MOMENTS OF INERTIA',T120,'MASSES')
   60 FORMAT(' ',T12,'(KIP-SEC/FT)',T40,'(KIP/FT)',T68,'(FT)',T91,
     1'(KIP-SEC**2-FT)',T115,'(KIP-SEC**2/FT)')
    1 FORMAT('0',T11,'C1=',E13.6,T37,'K1=',E13.6,T64,'A=',E13.6,T90,
     1'MI1=',E13.6,T116,'M1=',E13.6)
    2 FORMAT('0',T11,'C2=',E13.6,T37,'K2=',E13.6,T64,'B=',E13.6,T90,
     2'MI2=',E13.6,T116,'M2=',E13.6)
    3 FORMAT('0',T11,'C3=',E13.6,T37,'K3=',E13.6,T63,'CC=',E13.6,T90,
     3'MI3=',E13.6,T116,'M3=',E13.6)
    4 FORMAT('0',T11,'C4=',E13.6,T37,'K4=',E13.6,T64,'D=',E13.6,T90,
     4'MI4=',E13.6,T116,'M4=',E13.6)
    5 FORMAT('0',T11,'C5=',E13.6,T37,'K5=',E13.6,T64,'E=',E13.6,T90,
     5'MI5=',E13.6,T116,'M5=',E13.6)
    6 FORMAT('0',T11,'C6=',E13.6,T37,'K6=',E13.6,T64,'F=',E13.6,T90,
     6'MJ2=',E13.6,T116,'M6=',E13.6)
    7 FORMAT('0',T11,'C7=',E13.6,T37,'K7=',E13.6,T64,'G=',E13.6,T90,
     7'MJ4=',E13.6,T116,'M7=',E13.6)
    8 FORMAT('0',T11,'C8=',E13.6,T37,'K8=',E13.6,T64,'H=',E13.6,T90,
     8'MH2=',E13.6,T116,'M8=',E13.6)
    9 FORMAT('0',T11,'C9=',E13.6,T37,'K9=',E13.6,T64,'P=',E13.6,T90,
     9'MH4=',E13.6,T116,'M9=',E13.6)
   10 FORMAT('0',T10,'C10=',E13.6,T36,'K10=',E13.6,T64,'Q=',E13.6,T90,
     2'P67=',E13.6)
   11 FORMAT('0',T10,'C11=',E13.6,T36,'K11=',E13.6,T64,'R=',E13.6,T90,
     2'P62=',E13.6)
   12 FORMAT('0',T10,'C12=',E13.6,T36,'K12=',E13.6,T64,'S=',E13.6,T90,
     2'P27=',E13.6)
   13 FORMAT('0',T10,'C13=',E13.6,T36,'K13=',E13.6,T64,'T=',E13.6,T90,
     2'P84=',E13.6)
   14 FORMAT('0',T10,'C14=',E13.6,T36,'K14=',E13.6,T64,'U=',E13.6,T90,
     2'P89=',E13.6)
   15 FORMAT('0',T10,'C15=',E13.6,T36,'K15=',E13.6,T64,'V=',E13.6,T90,
     2'P49=',E13.6)
   16 FORMAT('0',T10,'C16=',E13.6,T36,'K16=',E13.6,T64,'W=',E13.6)
   17 FORMAT('0',T10,'C17=',E13.6,T36,'K17=',E13.6,T64,'X=',E13.6)
```

```
   18 FORMAT('0',T10,'C18=',E13.6,T36,'K18=',E13.6,T64,'Y=',E13.6)
   19 FORMAT('0',T10,'C19=',E13.6,T36,'K19=',E13.6,T64,'Z=',E13.6)
   20 FORMAT('0',T10,'C20=',E13.6,T36,'K20=',E13.6)
   21 FORMAT('0',T10,'C21=',E13.6,T36,'K21=',E13.6)
   22 FORMAT('0',T10,'C22=',E13.6,T36,'K22=',E13.6)
   23 FORMAT('0',T10,'C23=',E13.6,T36,'K23=',E13.6)
   24 FORMAT('0',T10,'C24=',E13.6,T36,'K24=',E13.6)
      WRITE(NWRITE,30)
      WRITE(NWRITE,40)
      WRITE(NWRITE,60)
      WRITE(NWRITE, 1) C1 ,K1 ,A ,MI1,M1
      WRITE(NWRITE, 2) C2 ,K2 ,B ,MI2,M2
      WRITE(NWRITE, 3) C3 ,K3 ,CC,MI3,M3
      WRITE(NWRITE, 4) C4 ,K4 ,D ,MI4,M4
      WRITE(NWRITE, 5) C5 ,K5 ,E ,MI5,M5
      WRITE(NWRITE, 6) C6 ,K6 ,F ,MJ2,M6
      WRITE(NWRITE, 7) C7 ,K7 ,G ,MJ4,M7
      WRITE(NWRITE, 8) C8 ,K8 ,H ,MH2,M8
      WRITE(NWRITE, 9) C9 ,K9 ,P ,MH4,M9
      WRITE(NWRITE,10) C10,K10,Q,P67
      WRITE(NWRITE,11) C11,K11,R,P62
      WRITE(NWRITE,12) C12,K12,S,P27
      WRITE(NWRITE,13) C13,K13,T,P84
      WRITE(NWRITE,14) C14,K14,U,P89
      WRITE(NWRITE,15) C15,K15,V,P49
      WRITE(NWRITE,16) C16,K16,W
      WRITE(NWRITE,17) C17,K17,X
      WRITE(NWRITE,18) C18,K18,Y
      WRITE(NWRITE,19) C19,K19,Z
      WRITE(NWRITE,20) C20,K20
      WRITE(NWRITE,21) C21,K21
      WRITE(NWRITE,22) C22,K22
      WRITE(NWRITE,23) C23,K23
      WRITE(NWRITE,24) C24,K24
C
C*****************************************************************************
C
      MR=M6*M8/(M6+M8)
      SPA=S*.5+A
      SMA=S*.5-A
      EPR=E+R
      TT=T*.5
      FMQ=F-Q
      M(1,1)=M1
      M(2,2)=M2+M4
      M(2,10)=-M4*A
      M(2,12)=M4*D
      M(2,14)=-M4*B
      M(2,15)=-M4*E
      M(3,3)=M3
      M(4,4)=M5
      M(5,5)=M7+M9
      M(5,10)=M9*P
      M(5,12)=-M9*Q
      M(5,16)=M9*B
      M(5,17)=M9*E
      M(6,6)=M1
      M(7,7)=M3
      M(8,8)=M5
      M(9,9)=MI1
```

```
      M(10,10)=M4*A*A+M9*P*P+MI2
      M(10,12)=-A*D*M4-P*Q*M9
      M(10,14)= M4*A*B-P67
      M(10,15)= M4*A*E
      M(10,16)=M9*P*B-P62
      M(10,17)=M9*P*E
      M(11,11)=MI3
      M(12,12)=M4*D*D+M9*Q*Q+MI4
      M(12,14)=-M4*D*B
      M(12,15)=-M4*D*E-P89
      M(12,16)=-M9*Q*B
      M(12,17)=-M9*Q*E-P84
      M(13,13)=MI5
      M(14,14)=M4*B*B+MR*P*P+MJ2
      M(14,15)=M4*B*E-MR*P*Q
      M(14,16)=-P*A*MR-P27
      M(14,17)=MR*P*D
      M(15,15)=M4*E*E+MR*C*Q+MJ4
      M(15,16)= MR*Q*A
      M(15,17)=-MR*Q*D-P49
      M(16,16)=MR*A*A+M9*B*B+MH2
      M(16,17)=-A*D*MR+B*E*M9
      M(17,17)=MR*D*D+M9*E*E+MH4
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      C(1,1)=C1+C2+C7+C8
      C(1,2)=-C1-C2
      C(1,9)=  W*(C8-C7)+U*(C2-C1)
      C(1,10)= SPA*C1-SMA*C2
      C(1,14)=-G*(C1+C2)
      C(2,2)=C1+C2+C3+C4+C5+C6
      C(2,3)=-C3-C4
      C(2,4)=-C5-C6
      C(2,9)=U*(C1-C2)
      C(2,10)=-SPA*(C1+C3)+SMA*(C2+C4)-A*(C5+C6)
      C(2,11)=U*(C3-C4)
      C(2,12)=TT*(C6-C5)
      C(2,13)=U*(C5-C6)
      C(2,14)=C1*G+C2*G-C3*H-C4*H-C5*B-C6*B
      C(2,15)=-EPR*(C5+C6)
      C(3,3)=C3+C4+C9+C10
      C(3,10)=SPA*C3-SMA*C4
      C(3,11)=W*(C10-C9)+U*(C4-C3)
      C(3,14)=C3*H+C4*H
      C(4,4)=C5+C6+C11+C12
      C(4,10)=A*(C5+C6)
      C(4,12)=TT*(C5-C6)
      C(4,13)=W*(C12-C11)+U*(C6-C5)
      C(4,14)=C5*B+C6*B
      C(4,15)=EPR*(C5+C6)
      C(5,5)=C15+C16+C19+C20+C23+C24
      C(5,6)=-C15-C16
      C(5,7)=-C19-C20
      C(5,8)=-C23-C24
      C(5,9)=V*(C15+C16)
      C(5,10)=CC*(C15+C16+C19+C20)+C23*P+C24*P
      C(5,11)=V*(C19+C20)
      C(5,12)=(C23+C24)*FMQ
      C(5,13)=V*(C23+C24)
```

```
      C(5,16)=-C15*G-C16*G+C19*H+C20*H+C23*B+C24*B
      C(5,17)=(C23+C24)*EPR
      C(6,6)=C13+C14+C15+C16
      C(6,9)=Y*(C13+C14)-V*(C15+C16)
      C(6,10)=-C15*CC-C16*CC
      C(6,16)=C15*G+C16*G
      C(7,7)=C17+C18+C1S+C20
      C(7,10)=-C19*CC-C20*CC
      C(7,11)=Y*(C17+C18)-V*(C19+C20)
      C(7,16)=-C19*H-C2C*H
      C(8,8)=C21+C22+C23+C24
      C(8,10)=-C23*P-C24*P
      C(8,12)=-FMQ*(C23+C24)
      C(8,13)=Y*(C21+C22)-V*(C23+C24)
      C(8,16)=-B*(C23+C24)
      C(8,17)=-(C23+C24)*EPR
      C(9,9)=W*W*(C7+C8)+Y*Y*(C13+C14)+U*U*(C1+C2)+V*V*(C15+C16)
      C(9,10)=-U*(SPA*C1+SMA*C2)+V*CC*(C15+C16)
      C(9,14)=U*G*(C1-C2)
      C(9,16)=-V*G*(C15+C16)
      C(10,10)=SPA*SPA*(C1+C3)+SMA*SMA*(C2+C4)+CC*CC*(C15+C16+C19+C20)+
     1A*A*(C5+C6)+P*P*(C23+C24)
      C(10,11)=-U*(SPA*C3+SMA*C4)+CC*V*(C19+C20)
      C(10,12)=A*TT*(C5-C6)+P*FMQ*(C23+C24)
      C(10,13)=A*U*(C6-C5)+P* V*(C23+C24)
      C(10,14)=-G*(SPA*C1-SMA*C2)+H*(SPA*C3-SMA*C4)+A*B*(C5+C6)
      C(10,15)=A* EPR  *(C5+C6)
      C(10,16)=-CC*G*(C15+C16)+CC*H*(C19+C20)+P*B*(C23+C24)
      C(10,17)=P*EPR  *(C23+C24)
      C(11,11)=W*W*(C9+C10)+Y*Y*(C17+C18)+U*U*(C3+C4)+V*V*(C19+C20)
      C(11,14)=U*H*(C4-C3)
      C(11,16)=V*H*(C19+C20)
      C(12,12)=TT*TT*(C5+C6)+FMQ*FMQ*(C23+C24)
      C(12,13)=-TT*U*(C5+C6)+FMQ*V*(C23+C24)
      C(12,14)=TT*B*(C5-C6)
      C(12,15)=TT*EPR*(C5-C6)
      C(12,16)=B*FMQ*(C23+C24)
      C(12,17)=EPR*FMQ*(C23+C24)
      C(13,13)=W*W*(C11+C12)+Y*Y*(C21+C22)+U*U*(C5+C6)+V*V*(C23+C24)
      C(13,14)=U*B*(C6-C5)
      C(13,15)=U*EPR*(C6-C5)
      C(13,16)=V*B*(C23+C24)
      C(13,17)=V*EPR*(C23+C24)
      C(14,14)=G*G*(C1+C2)+H*H*(C3+C4)+B*B*(C5+C6)
      C(14,15)=B*EPR*(C5+C6)
      C(15,15)=(C5+C6)*EPR*EPR
      C(16,16)=G*G*(C15+C16)+H*H*(C19+C20)+B*B*(C23+C24)
      C(16,17)=B*EPR*(C23+C24)
      C(17,17)=(C23+C24)*EPR*EPR
C
C*KKKKKKKKKK KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
C
      K(1,1)=K1+K2+K7+KE
      K(1,2)=-K1-K2
      K(1,9)=W*(KB-K7)+L*(K2-K1)
      K(1,10)=SPA*K1-SMA*K2
      K(1,14)=-G*(K1+K2)
      K(2,2)=K1+K2+K3+K4+K5+K6
      K(2,3)=-K3-K4
      K(2,4)=-K5-K6
```

```
 K(2,9)=U*(K1-K2)
 K(2,10)=-SPA*(K1+K3)+SMA*(K2+K4)-A*(K5+K6)
 K(2,11)=U*(K3-K4)
 K(2,12)=TT*(K6-K5)
 K(2,13)=U*(K5-K6)
 K(2,14)=K1*G+K2*G-K3*H-K4*H-K5*B-K6*B
 K(2,15)=-EPR*(K5+K6)
 K(3,3)=K3+K4+K9+K10
 K(3,10)=SPA*K3-SMA*K4
 K(3,11)=W*(K10-K9)+U*(K4-K3)
 K(3,14)=K3*H+K4*H
 K(4,4)=K5+K6+K11+K12
 K(4,10)=A*(K5+K6)
 K(4,12)=TT*(K5-K6)
 K(4,13)=W*(K12-K11)+U*(K6-K5)
 K(4,14)=K5*B+K6*B
 K(4,15)=EPR*(K5+K6)
 K(5,5)=K15+K16+K19+K20+K23+K24
 K(5,6)=-K15-K16
 K(5,7)=-K19-K20
 K(5,8)=-K23-K24
 K(5,9)=V*(K15+K16)
 K(5,10)=CC*(K15+K16+K19+K20)+K23*P+K24*P
 K(5,11)=V*(K19+K20)
 K(5,12)=(K23+K24)*FMQ
 K(5,13)=V*(K23+K24)
 K(5,16)=-K15*G-K16*G+K19*H+K20*H+K23*B+K24*B
 K(5,17)=(K23+K24)*EPR
 K(6,6)=K13+K14+K15+K16
 K(6,9)=Y*(K13+K14)-V*(K15+K16)
 K(6,10)=-K15*CC-K16*CC
 K(6,16)=K15*G+K16*G
 K(7,7)=K17+K18+K19+K20
 K(7,10)=-K19*CC-K20*CC
 K(7,11)=Y*(K17+K18)-V*(K19+K20)
 K(7,16)=-K19*H-K20*H
 K(8,8)=K21+K22+K23+K24
 K(8,10)=-K23*P-K24*P
 K(8,12)=-(K23+K24)*FMQ
 K(8,13)=Y*(K21+K22)-V*(K23+K24)
 K(8,16)=-B*(K23+K24)
 K(8,17)=-(K23+K24)*EPR
 K(9,9)=W*W*(K7+K8)+Y*Y*(K13+K14)+U*U*(K1+K2)+V*V*(K15+K16)
 K(9,10)=-U*(SPA*K1+SMA*K2)+V*CC*(K15+K16)
 K(9,14)=U*G*(K1-K2)
 K(9,16)=-V*G*(K15+K16)
 K(10,10)=SPA*SPA*(K1+K3)+SMA*SMA*(K2+K4)+CC*CC*(K15+K16+K19+K20)
1+A*A*(K5+K6)+P*P*(K23+K24)
 K(10,11)=-U*(SPA*K3+SMA*K4)+CC*V*(K19+K20)
 K(10,12)=A*TT*(K5-K6)+P*FMQ*(K23+K24)
 K(10,13)=A*U*(K6-K5)+P*V*(K23+K24)
 K(10,14)=-G*(SPA*K1-SMA*K2)+H*(SPA*K3-SMA*K4)+A*B*(K5+K6)
 K(10,15)= A*EPH*(K5+K6)
 K(10,16)=-CC*G*(K15+K16)+CC*H*(K19+K20)+P*B*(K23+K24)
 K(10,17)=P*EPR   *(K23+K24)
 K(11,11)=W*W*(K9+K10)+ Y*Y*(K17+K18)+U*U*(K3+K4)+V*V*(K19+K20)
 K(11,14)=U*H*(K4-K3)
 K(11,16)=V*H*(K19+K20)
 K(12,12)=TT*TT*(K5+K6)+FMQ*FMQ*(K23+K24)
 K(12,13)=-TT*U*(K5+K6)+FMQ*V*(K23+K24)
```

```
      K(12,14)=TT*B*(K5-K6)
      K(12,15)=TT*EPR*(K5-K6)
      K(12,16)=B*FMQ   *(K23+K24)
      K(12,17)=EPR*FMQ*(K23+K24)
      K(13,13)=W*W*(K11+K12)+Y*Y*(K21+K22)+U*U*(K5+K6)+V*V*(K23+K24)
      K(13,14)=U*B*(K6-K5)
      K(13,15)=U*EPR   *(K6-K5)
      K(13,16)=V*B*(K23+K24)
      K(13,17)=V*EPR*(K23+K24)
      K(14,14)=G*G*(K1+K2)+H*H*(K3+K4)+B*B*(K5+K6)
      K(14,15)=B* EPR* (K5+K6)
      K(15,15)=(K5+K6)*EPR*EPR
      K(16,16)=G*G*(K15+K16)+H*H*(K19+K20)+B*B*(K23+K24)
      K(16,17)=B*EPR   *(K23+K24)
      K(17,17)=(K23+K24)*EPR*EPR
      DO 70 I=1,16
      N=I+1
      DO 80 J=N,17
      H(J,I)=H(I,J)
       C(J,I)=C(I,J)
   80 K(J,I)=K(I,J)
   70 CONTINUE
      RETURN
      END
```

```
C
C     ..............................................................
C
C        SUBROUTINE MINV
C
C        PURPOSE
C           INVERT A MATRIX
C
C        USAGE
C           CALL MINV(A,N,D,L,M)
C
C        DESCRIPTION OF PARAMETERS
C           A - INPUT MATRIX, CESTROYED IN COMPUTATION AND REPLACED BY
C               RESULTANT INVERSE.
C           N - ORDER OF MATRIX A
C           D - RESULTANT DETERMINANT
C           L - WORK VECTOR OF LENGTH N
C           M - WORK VECTOR OF LENGTH N
C
C        REMARKS
C           MATRIX A MUST BE A GENERAL MATRIX
C
C        SUBROUTINES ANC FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE STANDARC GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
C           IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
C           THE MATRIX IS SINGULAR.
C
C     ..............................................................
C
      SUBROUTINE MINV(A,N,NN)
      REAL *8 A,D,BIGA,HOLD,DABS
C     DINENSION A(NN),L(N),M(N)
      DIMENSION L(34),M(34)
      DIMENSION A(NN)
C
C        ...........................................................
C
C        IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
C        STATEMENT WHICH FOLLOWS.
C
C     DOUBLE PRECISION A,D,BIGA,HOLD
C
C        THE C MUST ALSC BE REMOVED FROM DOUBLE PRECISION STATEMENTS
C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
C        ROUTINE.
C
C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
C        10 MUST BE CHANGED TO DABS.
C
C        ...........................................................
C
C        SEARCH FOR LARCEST ELEMENT
C
      D=1.0
      NK=-N
```

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

```
      IF(I-K) 60,65,60                                                  MINV1180
   60 IF(J-K) 62,65,62                                                  MINV1190
   62 KJ=IJ-I+K                                                         MINV1200
      A(IJ)=HOLD*A(KJ)+A(IJ)                                            MINV1210
   65 CONTINUE                                                          MINV1220
C                                                                       MINV1230
C        DIVIDE ROW BY PIVOT                                            MINV1240
C                                                                       MINV1250
      KJ=K-N                                                            MINV1260
      DO 75 J=1,N                                                       MINV1270
      KJ=KJ+N                                                           MINV1280
      IF(J-K) 70,75,70                                                  MINV1290
   70 A(KJ)=A(KJ)/BIGA                                                  MINV1300
   75 CONTINUE                                                          MINV1310
C                                                                       MINV1320
C        PRODUCT OF PIVCTS                                              MINV1330
C                                                                       MINV1340
      D=D*BIGA                                                          MINV1350
C                                                                       MINV1360
C        REPLACE PIVOT BY RECIPROCAL                                    MINV1370
C                                                                       MINV1380
      A(KK)=1.0/BIGA                                                    MINV1390
   80 CONTINUE                                                          MINV1400
C                                                                       MINV1410
C        FINAL ROW AND COLUMN INTERCHANGE                               MINV1420
C                                                                       MINV1430
      K=N                                                               MINV1440
  100 K=(K-1)                                                           MINV1450
      IF(K) 150,150,105                                                 MINV1460
  105 I=L(K)                                                            MINV1470
      IF(I-K) 120,120,108                                               MINV1480
  108 JQ=N*(K-1)                                                        MINV1490
      JR=N*(I-1)                                                        MINV1500
      DO 110 J=1,N                                                      MINV1510
      JK=JQ+J                                                           MINV1520
      HOLD=A(JK)                                                        MINV1530
      JI=JR+J                                                           MINV1540
      A(JK)=-A(JI)                                                      MINV1550
  110 A(JI) =HOLD                                                       MINV1560
  120 J=M(K)                                                            MINV1570
      IF(J-K) 100,100,125                                               MINV1580
  125 KI=K-N                                                            MINV1590
      DO 130 I=1,N                                                      MINV1600
      KI=KI+N                                                           MINV1610
      HOLD=A(KI)                                                        MINV1620
      JI=KI-K+J                                                         MINV1630
      A(KI)=-A(JI)                                                      MINV1640
  130 A(JI) =HOLD                                                       MINV1650
      GO TO 100                                                         MINV1660
  150 RETURN                                                            MINV1670
      END
```

```
C
C     ..............................................................
C
C        SUBROUTINE GMPRD
C
C        PURPOSE
C           MULTIPLY THC GENERAL MATRICES TO FORM A RESULTANT GENERAL
C           MATRIX
C
C        USAGE
C           CALL GMPRD(A,B,R,N,M,L)
C
C        DESCRIPTION OF PARAMETERS
C           A - NAME OF FIRST INPUT MATRIX
C           B - NAME OF SECOND INPUT MATRIX
C           R - NAME OF OUTPUT MATRIX
C           N - NUMBER CF ROWS IN A
C           M - NUMBER CF COLUMNS IN A AND ROWS IN B
C           L - NUMBER CF COLUMNS IN B
C
C        REMARKS
C           ALL MATRICES MUST BE STORED AS GENERAL MATRICES
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX A
C           MATRIX R CANNOT BE IN THE SAME LOCATION AS MATRIX B
C           NUMBER OF CCLUMNS OF MATRIX A MUST BE EQUAL TO NUMBER OF ROW
C           OF MATRIX B
C
C        SUBROUTINES ANC FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           THE M BY L MATRIX B IS PREMULTIPLIED BY THE N BY M MATRIX A
C           AND THE RESULT IS STORED IN THE N BY L MATRIX R.
C
C     ..............................................................
C
      SUBROUTINE GMPRD(A,B,R,N,M,L,NM,ML,NL)
      REAL *8 A,B,R
      DIMENSION A(NM),B(ML),R(NL)
C
      IR=0
      IK=-M
      DO 10 K=1,L
      IK=IK+M
      DO 10 J=1,N
      IR=IR+1
      JI=J-N
      IB=IK
      R(IR)=0
      DO 10 I=1,M
      JI=JI+N
      IB=IB+1
   10 R(IR)=R(IR)+A(JI)*B(IB)
      RETURN
      END
```

```
      SUBROUTINE FRANCI (A,VALU,NSUB,ANORM,NMAX)
C
C     EIGENVALUES OF REAL MATRICES
C
C**** REMOVE OR MODIFY NEXT  FOUR STATEMENTS IN SINGLE PRECISION VERSION
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 DEL
      COMMON /QR/ ITER(200), DUMMY(100)
      DATA EPS/Z3380000C00000000/
      DIMENSION A(NMAX,NSUB), VALU(NMAX,2)
C
      N=NSUB
C
C     REDUCE MATRIX TO UPPER HESSENBERG FORM
C
      CALL SUBDIA(A,N,NMAX)
C
C     COMPUTE MATRIX NORM
C
      L=1
   10 ANORM2=0.0
      DO 40 I=1,N
      I1=I-1+1/I
      DO 40 J=I1,N
      ANORM2=ANORM2+A(I,J)**2
   40 CONTINUE
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      ANORM2=DSQRT(ANORM2)
      IF (L.EQ.2) GO TO 190
      L=2
      ANORM=       ANORM2
C
      DEL=ANORM2*EPS
C
C     BEGINNING OF LOOP FOR ITERATIVE DETERMINATION OF EIGENVALUES
C     (ARRAY ITER HAS EFFECTIVELY BEEN CLEARED TO ZERO BY SUBDIA)
C
   50 K=NSUB+1-N
C
C     FIND ROOTS OF LOWER 2X2 MINOR
C
      IF(N.EQ.0) GO TO 250                                              CC ADDED
   60 ANN=A(N,N)
      IF (N-1) 250, 220, 70
C     CHANGE DABS TO ABS FOR SINGLE PRECISION
   70 IF (DABS(A(N,N-1)).LE.DEL) GO TO 220
      DIF=(A(N-1,N-1)-ANN)*0.5
      DISCSQ=DIF**2+A(N-1,N)*A(N,N-1)
      IF (DISCSQ.LE.0.0) GO TO 100
C**** CHANGE FUNCTION NAMES IN NEXT       STATEMENT  IN SINGLE PRECISION
      DISC=DSIGN(DSQRT(DISCSQ),DIF)
      E=DISC+DIF+ANN
      G=E-DISC-DISC
      F=0.0
      GO TO 110
  100 E=DIF+ANN
      G=E
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      F=DSQRT(-DISCSQ)
  110 IF (N.EQ.2) GO TO 230
```

```
C     CHANGE DABS TO ABS FOR SINGLE PRECISION
      IF (DABS(A(N-1,N-2)).LE.DEL) GO TO 230
C
C     CHOOSE SHIFT TOWARD ACCELERATING CONVERGENCE
C
      SIG=E+G
      RHO=E*G+F**2
      GO TO 200
C
C     SHIFT UNSUCCESSFUL-- CHOOSE DIFFERENTLY TO BREAK UP CLUSTERS
C
  190 SIG=0.125*      ANORM2
C
C     PERFORM QR ITERATION
C
  200 CALL QR2(A,N,DEL,SIG,RHO,NMAX)
C
      ITER(K)=ITER(K)+1
      IF ( MOD (ITER(K),50).NE.0) GO TO 60
      GO TO 10
C
C     SINGLE EIGENVALUE CONVERGED
C
  220 VALU(K,1)=ANN
      VALU(K,2)=0.0
      N=N-1
      GO TO 50
C
C     PAIR OF EIGENVALUES CONVERGED
C
  230 VALU(K,1)=E
      VALU(K,2)=F
      VALU(K+1,1)=G
      VALU(K+1,2)=-F
      N=N-2
      GO TO 50
C
  250 RETURN
      END
```

```
      SUBROUTINE SUBDIA(A,N,NMAX)
C
C     HOUSEHOLDER REDUCTION OF REAL MATRIX TO UPPER HESSENBERG FORM
C
C**** REMOVE OR MODIFY NEXT       STATEMENT  IN SINGLE PRECISION VERSION
      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION A(NMAX,N), WVEC(100), PVEC(100), QVEC(100)
      COMMON /QR/ WVEC, PVEC
      EQUIVALENCE (PVEC,QVEC)
C
      DO 200 I=1,N
C
C     REDUCE COLUMN OF MATRIX
C
      WVEC(I)=0.0
      I1=I+1
      I2=I1+1
      IF (I2.GT.N) GO TC 200
      SUM=0.0
      DO 70 J=I2,N
   70 SUM=SUM+A(J,I)**2
      IF (SUM.EQ.0.0) GC TO 200
   75 J=I1
      TEMP=A(J,I)
C**** CHANGE FUNCTION NAMES IN NEXT  FOUR STATEMENTS IN SINGLE PRECISION
      SUM=DSQRT(SUM+ TEMP  **2)
      A(J,I )=-DSIGN(SUM, TEMP  )
      WVEC(J )=DSQRT(      1.0+DABS( TEMP  )/SUM)
      DIV=DSIGN(     WVEC(J )*SUM, TEMP  )
      DO 85 J=I2,N
   85 WVEC(J)=A(J,I)/DIV
      SCALAR=0.0
      DO 95 J=I1,N
      PVEC(J)=0.0
      DO 90 K=I1,N
   90 PVEC(J)=PVEC(J)+A(K,J)*WVEC(K)
      SCALAR=SCALAR+PVEC(J)*WVEC(J)
   95 CONTINUE
      SCALAR=SCALAR/2.0
      DO 120 J=I1,N
      QVEC(J)=PVEC(J)-SCALAR*WVEC(J)
      DO 120 K=I1,N
      A(K,J)=A(K,J)-WVEC(K)*QVEC(J)
  120 CONTINUE
      DO 180 K=1,N
      QVEC(K)=-SCALAR*WVEC(K)
      DO 170 J=I1,N
  170 QVEC(K)=QVEC(K)+A(K,J)*WVEC(J)
      DO 180 J=I1,N
      A(K,J)=A(K,J)-QVEC(K)*WVEC(J)
  180 CONTINUE
C
  200 CONTINUE
C
      RETURN
      END
```

```
      SUBROUTINE QR2(A,N,DEL,SIG,RHO,NMAX)
C
C     DOUBLE QR ITERATION
C
C**** REMOVE OR MODIFY NEXT THREE STATEMENTS IN SINGLE PRECISION VERSION
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*4 DEL
      REAL*8 KAP
      DIMENSION A(NMAX,N), GAM(3), PSI(2)
           INTEGER P,Q
C
      N1=N-1
      N2=N1-1
C
C     FIND P
C
      DO 50 I1=1,N2
      I=N1-I1
      IF (I1.EQ.1) GO TO 40
      A10=A(I+1,I)
C     CHANGE DABS TO ABS FOR SINGLE PRECISION
      IF (DABS(A10).LE.DEL) GO TO 60
      A12=A(I+1,I+2)
      A21=A(I+2,I+1)
      A11=A(I+1,I+1)
      DIF=A11-SIG
      DENOM=DIF*A11+A12*A21+RHO
      IF (DENOM.EQ.0.0) GO TO 40
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
      A32=DABS(A(I+3,I+2))
      A22=A(I+2,I+2)
C**** CHANGE FUNCTION NAME  IN NEXT       STATEMENT  IN SINGLE PRECISION
C     CHANGE DABS TO ABS FOR SINGLE PRECISION
      IF (DABS(A10*A21*(DABS(A22+DIF)+A32)/DENOM).LE.DEL) GO TO 60
   40 P=I
   50 CONTINUE
C
C     FIND Q
C
   60 DO 100 I1=1,P
      I=P+1-I1
C     CHANGE DABS TO ABS FOR SINGLE PRECISION
      IF (DABS(A(I+1,I)).LE.DEL) GO TO 110
      Q=I
  100 CONTINUE
C
C     ITERATE
C
  110 DO 300 I=P,N1
      IF (I.NE.P) GO TO 130
      DIF=A(I,I)-SIG
      GAM(1)=DIF*A(I,I)+A(I,I+1)*A(I+1,I)+RHO
      GAM(2)=A(I+1,I)*(DIF+A(I+1,I+1))
      GAM(3)=A(I+1,I)*A(I+2,I+1)
      A(I+2,I)=0.0
      GO TO 150
  130 GAM(1)=A(I,I-1)
      GAM(2)=A(I+1,I-1)
C                                                                     CC ADDED
C     THE STATEMENTS WITH CC ADDED WERE ADDED SO THAT THIS PROGRAM WOULD C ADDED
```

```
C      NOT TERMINATE WHILE USING THE WATFIV COMPLIER    3/19/1971          CC ADDED
C                                                                         CC ADDED
       IF(I.EQ.N1) GO TO 140                                              CC ADDED
       GAM(3)=A(I+2,I-1)
C      IF (I.EQ.N1) GAM(3)=0.0                                            CC ADDED
       GO TO 150                                                          CC ADDED
   140 GAM(3)=0.00                                                        CC ADDED
C**** CHANGE FUNCTION NAMES IN NEXT       STATEMENT   IN SINGLE PRECISION
   150 KAP=DSIGN(DSQRT(GAM(1)**2+GAM(2)**2+GAM(3)**2),GAM(1))
       GK=GAM(1)+KAP
       IF (GK.NE.0.0) GO TO 170
       ALF=-2.0
       GK=ALF
       GO TO 180
   170 ALF=-GK/KAP
   180 PSI(1)=GAM(2)/GK
       PSI(2)=GAM(3)/GK
       IF (I.EQ.Q) GO TO 220
       IF (I.EQ.P) KAP=A(I,I-1)
       A(I,I-1)=-KAP
C
C      ROW OPERATION
C
   220 DO 250 J=I,N
       IF(I.EQ.N1) GO TO 230                                              CC ADDED
       ETA=ALF*(A(I,J)+PSI(1)*A(I+1,J)+PSI(2)*A(I+2,J))
       GO TO 221                                                          CC ADDED
   230 ETA=ALF*(A(I,J)+PSI(1)*A(I+1,J))                                   CC ADDED
   221 A(I,J)=ETA+A(I,J)                                                  CC ADDED
       A(I+1,J)=A(I+1,J)+PSI(1)*ETA
       IF (I.NE.N1) A(I+2,J)=A(I+2,J)+PSI(2)*ETA
   250 CONTINUE
C
C      COLUMN OPERATION
C
       L=I+2
       IF (L.GT.N) L=N
       DO 280 J=Q,L
       IF(I.EQ.N1) GO TO 260                                              CC ADDED
       ETA=ALF*(A(J,I)+PSI(1)*A(J,I+1)+PSI(2)*A(J,I+2))
       GO TO 281                                                          CC ADDED
   260 ETA=ALF*(A(J,I)+PSI(1)*A(J,I+1))                                   CC ADDED
   281 A(J,I)=ETA+A(J,I)                                                  CC ADDED
       A(J,I+1)=A(J,I+1)+PSI(1)*ETA
       IF (I.NE.N1) A(J,I+2)=A(J,I+2)+PSI(2)*ETA
   280 CONTINUE
       IF (I.GE.N2) GO TO 300
       ETA=ALF*PSI(2)*A(I+3,I+2)
       A(I+3,I)=ETA
       A(I+3,I+1)=A(I+3,I)*PSI(1)
       A(I+3,I+2)=A(I+3,I+2)+PSI(2)*ETA
   300 CONTINUE
C
       RETURN
       END
```

```
      SUBROUTINE ARANGE(VALU,NT,NTMO,NTPO,NUMBER)
      IMPLICIT REAL*8(A-H,O-Y),COMPLEX*16(Z)
      COMMON/INOUT/NREAC,NWRITE,NPUNCH
      DIMENSION VALU(NT,2),VALRE(34),VALIM(34)
C
C     IF THE EIGENVALUES ARE KNOWN TO BE REAL SET NUMBER.EQ.1.
C
C
C     IF THE EIGENVALUES ARE KNOWN TO BE COMPLEX SET NUMBER.NE.1.
C
      IF(NUMBER.EQ.1) GC TO 100
C
C     COMPLEX EIGENVALUES ARE SORTED INTO ASSCENDING ORDER IN THE FOLLOWING
C     STATEMENTS ENDING ON STATEMENT 99.
C
      DO 1 I=1,NT
      VALRE(I)=VALU(I,1)
    1 VALIM(I)=VALU(I,2)
      BIG=0.D0
      DO 99 I=1,NTMO
      JQUIT=NTPO-I
      JQUTMO=JQUIT-1
      DO 98 J=1,JQUTMO
      IF(DABS(VALIM(J)).LE.DABS(BIG)) GO TO 98
      JPO=J+1
      IF(DABS(VALIM(JPO)).GE.CABS(VALIM(J))) GO TO 98
      BIG=VALIM(J)
      REALPT=VALRE(J)
      DO 97 K=J,JQUTMO
      KPO=K+1
      VALIM(K)=VALIM(KPC)
   97 VALRE(K)=VALRE(KPC)
      VALIM(JQUIT)=BIG
      VALRE(JQUIT)=REALPT
   98 CONTINUE
      BIG=VALIM(JQUTMO)
   99 CONTINUE
C
C     THE DAMPED,COMPLEX EIGENVALUES HAVE BEEN ORDERED FROM SMALLEST TO LARGEST
C     IN ABSOLUTE VALUE WITH RESPECT TO THEIR IMAGINARY PARTS.(WHICH ARE THE
C     CIRCULAR FREQUENCIES)
C
C     WRITE OUT THESE ORDERED EIGENVALUES,PLACE THEM IN VALU(I,2) AND RETURN
C     TO MAIN.
C
      WRITE(NWRITE,7)
    7 FORMAT('1',45X,'THE N DAMPED,COMPLEX CONJUGATE EIGENVALUE PAIRS
     1ARE'/'-',54X,'REAL PART',16X,'IMAGINARY PART')
      WRITE(NWRITE,8) (VALRE(I),VALIM(I),I=1,NT)
    8 FORMAT('0',40X,1P2D28.15/' ',40X,1P2D28.15)
      DO 2 I=1,NT
      VALU(I,1)=VALRE(I)
    2 VALU(I,2)=VALIM(I)
      RETURN
C
C     REAL EIGENVALUES ARE SORTED IN ASSCENDING ORDER IN THE FOLLOWING STATEMENT
C     ENDING ON 78.
C
  100 CONTINUE
      DO 75 I=1,NT
```

```
   75 VALRE(I)=VALU(I,1)
      BIG=0.D0
      DO 78 I=1,NTMO
      JQUIT=NTPO-I
      JQUTMO=JQUIT-1
      DO 77 J=1,JQUTMO
      IF(VALRE(J).LE.BIG) GO TO 77
      JPO=J+1
      IF(VALRE(JPO).GE.VALRE(J)) GO TO 77
      BIG=VALRE(J)
      DO 76 K=J,JQUTMO
      KPO=K+1
   76 VALRE(K)=VALRE(KPO)
      VALRE(JQUIT)=BIG
   77 CONTINUE
      BIG=VALRE(JQUTMO)
   78 CONTINUE
C
C     THE UNDAMPED,REAL EIGENVALUES HAVE BEEN ORDERED. PLACE THESE ORDERED
C     EIGENVALUES IN VALU(I,1). DETERMINE AND WRITE OUT THE UNDAMPED CIRCULAR
C     FREQUENCIES.
C
      DO 79 I=1,NT
   79 VALU(I,1)=VALRE(I)
      DO 81 I=1,NT
   81 VALRE(I)=DSQRT(DABS(VALRE(I)))
      WRITE(NWRITE,83)
   83 FORMAT('1',45X,'THE N UNDAMPED CIRCULAR FREQUENCIES ARE'/)
      WRITE(NWRITE,85) (VALRE(I),I=1,NT)
   85 FORMAT('-',48X,1P1D28.15)
      RETURN
      END
```

```
      SUBROUTINE VECITR
      IMPLICIT REAL*8(A-H,O-Y),COMPLEX*16(Z)
      COMPLEX*16 DCMPLX,DCONJG
      COMMON/COMX/ZVEC(35,34)
      COMMON/DYNAM/D(34,34),VALU(34,2)
      COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
      COMMON/AI/DL(17,17),AMA(17,17)
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DIMENSION ZHOLD(34),ZRIGHT(17)
      DIMENSION OVERZR(34)
      EQUIVALENCE(OVERZR,ZRIGHT)
      DIMENSION RIGHT(34)
      DO 1 L=1,NT,2
      ZLAMB=ZVEC(35,L)
      ROOTR=VALU(L,1)
      ROOTI=VALU(L,2)
      RRMII=ROOTR*ROOTR-ROOTI*ROOTI
      TRI=2.D0*ROOTR*ROOTI
      DO 11 I=1,N
      IPN=I+N
      DO 11 J=1,N
      JPN=J+N
      R=RRMII*AMA(I,J)+ROOTR*AC(I,J)+AK(I,J)
      AI=TRI*AMA(I,J)+ROOTI*AC(I,J)
      D(I,J)=R
      D(IPN,JPN)=R
      D(I,JPN)=-AI
   11 D(IPN,J)=AI
      DO 22 I=1,NT
   22 ZHOLD(I)=ZVEC(I,L)
      DO 33 I=1,N
      ZKX=(0.D0,0.D0)
      ZMY=ZKX
      DO 44 J=1,N
      ZKX=ZKX+DCMPLX(AK(I,J),0.D0)*ZHOLD(J)
   44 ZMY=ZMY+DCMPLX(AMA(I,J),0.D0)*ZHOLD(J+N)
   33 ZRIGHT(I)=ZKX-ZLAMB*ZMY
      DO 55 I=1,N
      IR=2*I-1
      II=2*I
      RIGHT(I)=OVERZR(IR)
   55 RIGHT(I+N)=OVERZR(II)
      CALL DGELG(RIGHT,D,NT,1,1.E-12,IER)
      DO 66 I=1,N
      IPN=I+N
   66 ZHOLD(IPN)=DCMPLX(RIGHT(I),RIGHT(IPN))
      DO 77 I=1,N
   77 ZHOLD(I)=(ZHOLD(I+N)-ZHOLD(I))/ZLAMB
      DO 88 I=1,NT
      ZVEC(I,L)=ZHOLD(I)
   88 ZVEC(I,L+1)=DCONJG(ZHOLD(I))
    1 CONTINUE
      RETURN
      END
```

```
C
C     ..................................................................
C
C        SUBROUTINE DGELG (THIS IS ONE OF THE SUBROUTINES IN THE IBM
C        SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE).
C
C        PURPOSE
C           TO SOLVE A GENERAL SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS.
C
C        USAGE
C           CALL DGELG(R,A,M,N,EPS,IER)
C
C        DESCRIPTION OF PARAMETERS
C           R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX
C                    (DESTROYED). ON RETURN R CONTAINS THE SOLUTIONS
C                    OF THE EQUATIONS.
C           A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX
C                    (DESTROYED).
C           M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.
C           N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.
C           EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS
C                    RELATIVE TOLERANCE FOR TEST ON LOSS OF
C                    SIGNIFICANCE.
C           IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS
C                    IER=0  - NO ERROR,
C                    IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR
C                             PIVOT ELEMENT AT ANY ELIMINATION STEP
C                             EQUAL TO 0,
C                    IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI-
C                             CANCE INDICATED AT ELIMINATION STEP K+1,
C                             WHERE PIVOT ELEMENT WAS LESS THAN OR
C                             EQUAL TO THE INTERNAL TOLERANCE EPS TIMES
C                             ABSOLUTELY GREATEST ELEMENT OF MATRIX A.
C
C        REMARKS
C           INPUT MATRICES R AND A ARE ASSUMED TO BE STORED COLUMNWISE
C           IN M*N RESP. M*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN
C           SOLUTION MATRIX R IS STORED COLUMNWISE TOO.
C           THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS
C           GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS
C           ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN -
C           INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL
C           SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE
C           INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS
C           GIVEN IN CASE M=1.
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH
C           COMPLETE PIVOTING.
C
C     ..................................................................
C
      SUBROUTINE DGELG(R,A,M,N,EPS,IER)
C
C
      DIMENSION A(1),R(1)
      DOUBLE PRECISION R,A,PIV,TB,TOL,PIVI
```

```
      IF(M)23,23,1                                                      DGELG060
C                                                                       DGELG061
C        SEARCH FOR GREATEST ELEMENT IN MATRIX A                        DGELG062
    1 IER=0                                                             DGELG063
      PIV=0.D0                                                          DGELG064
      MM=M*M                                                            DGELG065
      NM=N*M                                                            DGELG066
      DO 3 L=1,MM                                                       DGELG067
      TB=DABS(A(L))                                                     DGELG068
      IF(TB-PIV)3,3,2                                                   DGELG069
    2 PIV=TB                                                            DGELG070
      I=L                                                               DGELG071
    3 CONTINUE                                                          DGELG072
      TOL=EPS*PIV                                                       DGELG073
C     A(I) IS PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF A(I).   DGELG074
C                                                                       DGELG075
C                                                                       DGELG076
C     START ELIMINATION LOOP                                            DGELG077
      LST=1                                                             DGELG078
      DO 17 K=1,M                                                       DGELG079
C                                                                       DGELG080
C     TEST ON SINGULARITY                                               DGELG081
      IF(PIV)23,23,4                                                    DGELG082
    4 IF(IER)7,5,7                                                      DGELG083
    5 IF(PIV-TOL)6,6,7                                                  DGELG084
    6 IER=K-1                                                           DGELG085
    7 PIVI=1.D0/A(I)                                                    DGELG086
      J=(I-1)/M                                                         DGELG087
      I=I-J*M-K                                                         DGELG088
      J=J+1-K                                                           DGELG089
C     I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT               DGELG090
C                                                                       DGELG091
C     PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R      DGELG092
      DO 8 L=K,NM,M                                                     DGELG093
      LL=L+I                                                            DGELG094
      TB=PIVI*R(LL)                                                     DGELG095
      R(LL)=R(L)                                                        DGELG096
    8 R(L)=TB                                                           DGELG097
C                                                                       DGELG098
C     IS ELIMINATION TERMINATED                                         DGELG099
      IF(K-M)9,18,18                                                    DGELG100
C                                                                       DGELG101
C     COLUMN INTERCHANGE IN MATRIX A                                    DGELG102
    9 LEND=LST+M-K                                                      DGELG103
      IF(J)12,12,10                                                     DGELG104
   10 II=J*M                                                            DGELG105
      DO 11 L=LST,LEND                                                  DGELG106
      TB=A(L)                                                           DGELG107
      LL=L+II                                                           DGELG108
      A(L)=A(LL)                                                        DGELG109
   11 A(LL)=TB                                                          DGELG110
C                                                                       DGELG111
C     ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A               DGELG112
   12 DO 13 L=LST,MM,M                                                  DGELG113
      LL=L+I                                                            DGELG114
      TB=PIVI*A(LL)                                                     DGELG115
      A(LL)=A(L)                                                        DGELG116
   13 A(L)=TB                                                           DGELG117
C                                                                       DGELG118
C     SAVE COLUMN INTERCHANGE INFORMATION                               DGELG119
```

```
      A(LST)=J                                                          DGELG120
C                                                                       DGELG121
C     ELEMENT REDUCTION AND NEXT PIVOT SEARCH                           DGELG122
      PIV=0.D0                                                          DGELG123
      LST=LST+1                                                         DGELG124
      J=0                                                               DGELG125
      DO 16 II=LST,LEND                                                 DGELG126
      PIVI=-A(II)                                                       DGELG127
      IST=II+M                                                          DGELG128
      J=J+1                                                             DGELG129
      DO 15 L=IST,MM,M                                                  DGELG130
      LL=L-J                                                            DGELG131
      A(L)=A(L)+PIVI*A(LL)                                              DGELG132
      TB=DABS(A(L))                                                     DGELG133
      IF(TB-PIV)15,15,14                                                DGELG134
   14 PIV=TB                                                            DGELG135
      I=L                                                               DGELG136
   15 CONTINUE                                                          DGELG137
      DO 16 L=K,NM,M                                                    DGELG138
      LL=L+J                                                            DGELG139
   16 R(LL)=R(LL)+PIVI*R(L)                                             DGELG140
   17 LST=LST+M                                                         DGELG141
C     END OF ELIMINATION LOOP                                           DGELG142
C                                                                       DGELG143
C                                                                       DGELG144
C     BACK SUBSTITUTION AND BACK INTERCHANGE                            DGELG145
   18 IF(M-1)23,22,19                                                   DGELG146
   19 IST=MM+M                                                          DGELG147
      LST=M+1                                                           DGELG148
      DO 21 I=2,M                                                       DGELG149
      II=LST-I                                                          DGELG150
      IST=IST-LST                                                       DGELG151
      L=IST-M                                                           DGELG152
      L=A(L)+.5D0                                                       DGELG153
      DO 21 J=II,NM,M                                                   DGELG154
      TB=R(J)                                                           DGELG155
      LL=J                                                              DGELG156
      DO 20 K=IST,MM,M                                                  DGELG157
      LL=LL+1                                                           DGELG158
   20 TB=TB-A(K)*R(LL)                                                  DGELG159
      K=J+L                                                             DGELG160
      R(J)=R(K)                                                         DGELG161
   21 R(K)=TB                                                           DGELG162
   22 RETURN                                                            DGELG163
C                                                                       DGELG164
C                                                                       DGELG165
C     ERROR RETURN                                                      DGELG166
   23 IER=-1                                                            DGELG167
      RETURN                                                            DGELG168
      END                                                               DGELG169
```

```
      BLOCK DATA
      IMPLICIT REAL*8 (A-H,O-Y),COMPLEX *16(Z)
      COMMON/COMX/ZVEC(35,34)
      COMMON/DYNAM/D(34,34),VALU(34,2)
      COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
      COMMON/A1/ DL(17,17),AMA(17,17)
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DATA D,VALU/1224*0.D0/
      DATA DL,AMA/578*0.D0/
      DATA NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT/16,17,18,33,34,35,289,1156/
      DATA NREAD,NWRITE,NPUNCH/5,6,7/
      DATA AM,AC,AK/867*0.D0/
      END
      SUBROUTINE BIORTH
      IMPLICIT REAL*8 (A-H,O-Y),COMPLEX *16(Z)
      COMPLEX*16 CDSQRT,DCMPLX,DCONJG
      COMMON/COMX/ZVEC(35,34)
      COMMON/DYNAM/D(34,34),VALU(34,2)
      COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
      COMMON/A1/ DL(17,17),AMA(17,17)
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DIMENSION ZHOLD(17)
      DO 1 K=1,NT,2
      ROOTR=VALU(K,1)
      ROOTI=VALU(K,2)
      DO 11 I=1,N
   11 ZHOLD(I)=ZVEC(I,K)
      ZSUMT=(0.D0,0.D0)
      DO 22 J=1,N
      ZSUMO=(0.D0,0.D0)
      DO 33 I=1,N
   33 ZSUMO=ZSUMO+
     1DCMPLX(2.D0*ROOTR*AMA(I,J)+AC(I,J),2.D0*ROOTI*AMA(I,J))*ZHOLD(I)
   22 ZSUMT=ZSUMT+ZSUMO*ZHOLD(J)
      ZSQ=CDSQRT((1.D0,0.D0)/ZSUMT)
      ZSQCON=DCONJG(ZSQ)
      DO 44 I=1,NT
      ZVEC(I,K+1)=ZVEC(I,K+1)*ZSQCON
   44 ZVEC(I,K)=ZVEC(I,K)*ZSQ
    1 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE ROADIN(ZWL,ZWR,WL,WR,NOREAD)
      IMPLICIT REAL*8(A-H,K,O-Y),COMPLEX*16(Z)
      COMPLEX*16 DCMPLX
      REAL *8 DSIN,DCOS
      NAMELIST/ROAD/ CVII,CVIII,CIX,CX,CXI,CXII,KVII,KVIII,KIX,KX,KXI,
     1KXII,AL,AR,ZWIDE,BT,GH1,GBER2,SPEED,FWL,FWR
      DIMENSION ZTIME(5)
      COMMON/ZROAD/ZCL(6),ZCR(6)
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      COMMON/RIPPLE/WAVEL,WAVER,FAST,AMPL,AMPR,FAZE
      IF(NOREAD.NE.1) GO TO 10
      READ(NREAD,ROAD)
      WL=FWL
      WR=FWR
      FAST=SPEED
      AMPL=AL
      AMPR=AR
      FAZE=BT
   10 CONTINUE
      T1=WL*GH1/SPEED
      T2=WL*GBER2/SPEED
      ZTIME(1)=DCMPLX(DCOS(T1),-DSIN(T1))
      ZTIME(2)=DCMPLX(DCOS(T2),-DSIN(T2))
      ZTIME(3)=DCMPLX(DCOS(BT),-DSIN(BT))
      T1=WR*T1/WL
      T2=WR*T2/WL
      ZTIME(4)=DCMPLX(DCOS(T1+BT),-DSIN(T1+BT))
      ZTIME(5)=DCMPLX(DCOS(T2+BT),-DSIN(T2+BT))
      ZCL(1)=DCMPLX(CVII*WL*AL,-KVII*AL)
      ZCL(2)=DCMPLX(CIX*WL*AL,-KIX*AL)*ZTIME(1)
      ZCL(3)=DCMPLX(CXI*WL*AL,-KXI*AL)*ZTIME(2)
      ZCL(4)=-ZWIDE*ZCL(1)
      ZCL(5)=-ZWIDE*ZCL(2)
      ZCL(6)=-ZWIDE*ZCL(3)
      ZCR(1)=DCMPLX(CVIII*WR*AR,-KVIII*AR)*ZTIME(3)
      ZCR(2)=DCMPLX(CX*WR*AR,-KX*AR)*ZTIME(4)
      ZCR(3)=DCMPLX(CXII*WR*AR,-KXII*AR)*ZTIME(5)
      ZCR(4)=ZWIDE*ZCR(1)
      ZCR(5)=ZWIDE*ZCR(2)
      ZCR(6)=ZWIDE*ZCR(3)
      ZWL=DCMPLX(0.D0,WL)
      ZWR=DCMPLX(0.D0,WR)
      WAVEL=6.283185307179586*SPEED/WL
      WAVER=6.283185307179586*SPEED/WR
      RETURN
      END
```

```
      SUBROUTINE RIDE(ZWR,ZWL,NT,N)
      IMPLICIT REAL*8(A-H,O-Y),COMPLEX*16(Z)
      COMPLEX*16 DCONJG
      COMMON/COMX/ZVEC(35,34)
      COMMON/ZRIDE/ZRIDL(17),ZRIDR(17)
      COMMON/ZROAD/ZCL(6),ZCR(6)
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DIMENSION ZSCALL(34),ZSCALR(34)
      DIMENSION ZHOLD(6)
      DO 1 K=1,NT
      ZLAMB=ZVEC(NT+1,K)
      ZHOLD(1)=ZVEC(1,K)
      ZHOLD(2)=ZVEC(3,K)
      ZHOLD(3)=ZVEC(4,K)
      ZHOLD(4)=ZVEC(9,K)
      ZHOLD(5)=ZVEC(11,K)
      ZHOLD(6)=ZVEC(13,K)
      ZSUM=(0.D0,0.D0)
      DO 11 I=1,6
   11 ZSUM=ZSUM+ZHOLD(I)*ZCL(I)
      ZSCALL(K)=ZSUM/(ZWL-ZLAMB)
      ZSUM=(0.D0,0.D0)
      DO 22 I=1,6
   22 ZSUM=ZSUM+ZHOLD(I)*ZCR(I)
    1 ZSCALR(K)=ZSUM/(ZWR-ZLAMB)
   55 FORMAT('1',34(' ',1P4D30.15/))
      DO 33 I=1,N
      ZRIDL(I)=(0.D0,0.D0)
   33 ZRIDR(I)=(0.D0,0.D0)
      DO 44 K=1,NT
      DO 44 I=1,N
      ZRIDL(I)=ZRIDL(I)+ZVEC(I,K)*ZSCALL(K)
   44 ZRIDR(I)=ZRIDR(I)+ZVEC(I,K)*ZSCALR(K)
      DO 66 I=1,N
      ZRIDL(I)=DCONJG(ZRIDL(I))
   66 ZRIDR(I)=DCONJG(ZRIDR(I))
      RETURN
      END
```

```
      SUBROUTINE WEQULZ
      IMPLICIT REAL *8(A-H,O-Y),COMPLEX *16(Z)
      COMPLEX*16 CDABS
      COMMON/ZRIDE/ZRIDL(17),ZRIDR(17)
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/RIPPLE/WAVEL,WAVER,FAST,AMPL,AMPR,FAZE
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DIMENSION AMPMAX(17)
      DO 1000 I=1,N
 1000 AMPMAX(I)=CDABS(ZRIDL(I)+ZRIDR(I))
      WRITE(NWRITE,111)
      WRITE(NWRITE,222) FAZE
      WRITE(NWRITE, 1) AMPMAX( 1)
      WRITE(NWRITE, 2) AMPMAX( 2)
      WRITE(NWRITE, 3) AMPMAX( 3)
      WRITE(NWRITE, 4) AMPMAX( 4)
      WRITE(NWRITE, 5) AMPMAX( 5)
      WRITE(NWRITE, 6) AMPMAX( 6)
      WRITE(NWRITE, 7) AMPMAX( 7)
      WRITE(NWRITE, 8) AMPMAX( 8)
      WRITE(NWRITE, 9) AMPMAX( 9)
      WRITE(NWRITE,10) AMPMAX(10)
      WRITE(NWRITE,11) AMPMAX(11)
      WRITE(NWRITE,12) AMPMAX(12)
      WRITE(NWRITE,13) AMPMAX(13)
      WRITE(NWRITE,14) AMPMAX(14)
      WRITE(NWRITE,15) AMPMAX(15)
      WRITE(NWRITE,16) AMPMAX(16)
      WRITE(NWRITE,17) AMPMAX(17)
    1 FORMAT('0',56X,'    X1   = ',1P1D9.2)
    2 FORMAT('0',56X,'    X2   = ',1P1D9.2)
    3 FORMAT('0',56X,'    X3   = ',1P1D9.2)
    4 FORMAT('0',56X,'    X5   = ',1P1D9.2)
    5 FORMAT('0',56X,'    X7   = ',1P1D9.2)
    6 FORMAT('0',56X,'    Z1   = ',1P1D9.2)
    7 FORMAT('0',56X,'    Z3   = ',1P1D9.2)
    8 FORMAT('0',56X,'    Z5   = ',1P1D9.2)
    9 FORMAT('0',56X,'THETA1 = ',1P1D9.2)
   10 FORMAT('0',56X,'THETA2 = ',1P1D9.2)
   11 FORMAT('0',56X,'THETA3 = ',1P1D9.2)
   12 FORMAT('0',56X,'THETA4 = ',1P1D9.2)
   13 FORMAT('0',56X,'THETA5 = ',1P1D9.2)
   14 FORMAT('0',56X,'  PHI2   = ',1P1D9.2)
   15 FORMAT('0',56X,'  PHI4   = ',1P1D9.2)
   16 FORMAT('0',56X,'   XI2   = ',1P1D9.2)
   17 FORMAT('0',56X,'   XI4   = ',1P1D9.2)
  111 FORMAT('1',52X,'THE MAXIMUM AMPLITUDES ARE'//)
  222 FORMAT('-',35X,'THE RELATIVE PHASE ANGLE IS'/'0',60X,1P1D10.3///)
      RETURN
      END
```

Appendix B

EXPLANATION OF INPUT DATA FOR UNDETERMINED COEFFICIENTS PROGRAM WITH SOURCE LISTING

The required input for this program is identical to that for the program explained in appendix A except for the namelist data group FORCE. Data group FORCE contains the same variables as data group ROAD, except ZWIDE becomes the REAL*8 variable, WIDE, and possesses the additional variables defined below. If a Bode plot is desired for a given vehicle, these variables may be employed.

NUMB = the number of discrete steps desired (integer)

RUNGL = the step size for the left side (radians/second)

RUNGR = the step size for the right side (radians/second)

```
      IMPLICIT REAL*8(A-H,O-Y)
      COMMON/DYNAM/D(34,34),VALU(34,2)
      COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/A1/DL(17,17),AMA(17,17)
      COMMON/INOUT/NREAC,NWRITE,NPUNCH
      COMMON/R1/TIME(10)
      COMMON/MISC/WL,WR,FAST,STEPL,STEPR,AMPL,AMPR,FAZE,NSTEPS
      DIMENSION     CI(17),CII(17),CIII(17),SI(17),SII(17),SIII(17)
      COMMON/DOUT/XCL(17),XSL(17),XCR(17),XSR(17)
      COMMON/CUSSIN/SAC(34,3)
      EQUIVALENCE(SAC(1,1),CI),(SAC(1,2),CII),(SAC(1,3),CIII)
      EQUIVALENCE(SAC(18,1),SI),(SAC(18,2),SII),(SAC(18,3),SIII)
      CALL DEFINE
      DO 2 I=1,N
      DO 2 J=1,N
    2 AMA(I,J)=AM(I,J)
      NOREAD=0
      ERO=0.D0
  200 NSIDE=1
      DO 3 J=1,3
      DO 3 I=1,NT
    3 SAC(I,J)=0.D0
      CALL FORCIN(NOREAC,NSIDE)
      DO 100 I=1,N
      IPN=I+N
      DO 100 J=1,N
      JPN=J+N
      WMK=-WL*WL*AMA(I,J)+AK(I,J)
      WC=WL*AC(I,J)
      D(I,J)=WMK
      D(IPN,JPN)=WMK
      D(I,JPN)=WC
  100 D(IPN,J)=-WC
      CALL DGELG(SAC,D,NT,3,1.E-14,IER)
      IF(IER.NE.0) WRITE(NWRITE,333)
  333 FORMAT('-',50X,I1C)
      DO 110 I=1,N
      XCL(I)=ERO
      XSL(I)=ERO
      XCR(I)=ERO
  110 XSR(I)=ERO
      DO 120 I=1,N
      XCL(I)=CI(I)+CII(I)*TIME(2)-SII(I)*TIME(1)+CIII(I)*TIME(4)-SIII(I)
     1*TIME(3)
  120 XSL(I)=SI(I)+CII(I)*TIME(1)+SII(I)*TIME(2)+CIII(I)*TIME(3)+SIII(I)
     1*TIME(4)
      DO 130 J=1,3
      DO 130 I=1,NT
  130 SAC(I,J)=ERO
      NOREAD=1+NOREAD
      NSIDE=2
      CALL FORCIN(NOREAC,NSIDE)
      DO 140 I=1,N
      IPN=I+N
      DO 140 J=1,N
      JPN=J+N
      WMK=-WR*WR*AMA(I,J)+AK(I,J)
      WC=WR*AC(I,J)
      D(I,J)=WMK
```

```
      D(IPN,JPN)=WMK
      D(I,JPN)=WC
  140 D(IPN,J)=-WC
      CALL DGELG(SAC,D,NT,3,1.E-14,IER)
      IF(IER.NE.0) WRITE(NWRITE,333)
      DO 160 I=1,N
      XCR(I)=CI(I)*TIME(6)-SI(I)*TIME(5)+CII(I)*TIME(8)-SII(I)*TIME(7)
     1+CIII(I)*TIME(10)-SIII(I)*TIME(9)
  160 XSR(I)=CI(I)*TIME(5)+SI(I)*TIME(6)+CII(I)*TIME(7)+SII(I)*TIME(8)
     1+CIII(I)*TIME(9)+SIII(I)*TIME(10)
      WRITE(NWRITE,179)
  179 FORMAT('1')
      WRITE(NWRITE,210) FAST
      WRITE(NWRITE,190)
      WRITE(NWRITE,178) WL,WR
      WRITE(NWRITE,3000) AMPL,AMPR
 3000 FORMAT('-',25X,'THE AMPLITUDE (FEET) OF THE LEFT AND RIGHT'
     1,1X,'SINUSOIDAL PATHS ARE RESPECTIVELY'/'0',47X,1P2D15.4)
  190 FORMAT('-',10X,'THE SINUSOIDAL FORCING FREQUENCIES (RADIANS/SEC)'
     1,1X,'FOR THE LEFT AND FOR THE RIGHT SIDE WHEEL PATHS ARE'
     2,1X,'RESPECTIVELY')
  210 FORMAT(' ',30X,'THE SPEED (FEET/SECOND) OF THIS VEHICLE'
     1,1X,'IS EQUAL TO',1P1D12.4)
  178 FORMAT('0',35X,'WL=',1P1D22.15,10X,'WR=',1P1D22.15)
      WAVEL=6.283185307179586*FAST /WL
      WAVER=6.283185307179586*FAST /WR
      WRITE(NWRITE,4) WAVEL,WAVER
    4 FORMAT('-',25X,'THE WAVELENGTH OF THE LEFT AND RIGHT SINUSOIDAL'
     1,1X,'DISPLACEMENT FUNCTIONS ARE RESPECTIVELY'/'0',40X,1P2D25.10)
      WRITE(NWRITE,177)
  177 FORMAT('0',25X,'OUTPUT(I) = A(I)*COS(WL*T) + B(I)*SIN(WL*T) + '
     1,1X,'C(I)*COS(WR*T) + D(I)*SIN(WR*T)')
      WRITE(NWRITE,181)
  181 FORMAT('-',17X,'A(I)',26X,'B(I)',26X,'C(I)',26X'D(I)')
      DO 170 I=1,N
  170 WRITE(NWRITE,180) XCL(I),XSL(I),XCR(I),XSR(I)
  180 FORMAT('0',1P4D30.15)
      DIFFER=DABS(WL-WR)
      IF(DIFFER.LT.1.D-7) CALL WEQUAL
      WL=WL+STEPL
      WR=WR+STEPR
      IF(NSTEPS.GE.NOREAD) GO TO 200
      STOP
      END
```

```
      SUBROUTINE DEFINE
      REAL*8 M,C,K
      REAL *4 M1,M2,M3,M4,M5,M6,M7,M8,M9
      REAL *4 MI1,MI2,MI3,MI4,MI5,MJ2,MJ4,MH2,MH4,P67,P62,P27,P84,P89,
     UP49
      REAL *4 K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15,K16
      REAL *4 K17,K18,K19,K20,K21,K22,K23,K24
      COMMON/INOUT/NREAC,NWRITE,NPUNCH
      COMMON/MATMCK/M(17,17),C(17,17),K(17,17)
      NAMELIST/MASSES/M1,M2,M3,M4,M5,M6,M7,M8,M9
      NAMELIST/INERTA/MI1,MI2,MI3,MI4,MI5,MJ2,MJ4,MH2,MH4,
     1P67,P62,P27,P84,P89,P49
      NAMELIST/DAMPER/C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13,C14,C15
     1,C16,C17,C18,C19,C20,C21,C22,C23,C24
      NAMELIST/SPRING/K1,K2,K3,K4,K5,K6,K7,K8,K9,K10,K11,K12,K13,K14,K15
     1,K16,K17,K18,K19,K20,K21,K22,K23,K24
      NAMELIST/DIMEN/A,B,CC,D,E,F,G,H,P,Q,R,S,T,U,V,W,X,Y,Z
      READ(NREAD,MASSES)
      READ(NREAD,INERTA)
      READ(NREAD,DAMPER)
      READ(NREAD,SPRING)
      READ(NREAD,DIMEN)
   30 FORMAT('1',T43,'FOR THIS SEMI-TRAILER TRUCK THE CHARACTERIZING'
     1/' ',T53,'PARAMETERS ARE AS FOLLOWS.'////)
   40 FORMAT(' ',T14,'DAMPERS',T40,'SPRINGS',T60,'DIMENSIONAL LENGTHS'
     1,T87,'MASS MOMENTS OF INERTIA',T120,'MASSES')
   60 FORMAT(' ',T12,'(KIP-SEC/FT)',T40,'(KIP/FT)',T68,'(FT)',T91,
     1'(KIP-SEC**2-FT)',T115,'(KIP-SEC**2/FT)')
    1 FORMAT('0',T11,'C1=',E13.6,T37,'K1=',E13.6,T64,'A=',E13.6,T90,
     1'MI1=',E13.6,T116,'M1=',E13.6)
    2 FORMAT('0',T11,'C2=',E13.6,T37,'K2=',E13.6,T64,'B=',E13.6,T90,
     2'MI2=',E13.6,T116,'M2=',E13.6)
    3 FORMAT('0',T11,'C3=',E13.6,T37,'K3=',E13.6,T63,'CC=',E13.6,T90,
     3'MI3=',E13.6,T116,'M3=',E13.6)
    4 FORMAT('0',T11,'C4=',E13.6,T37,'K4=',E13.6,T64,'D=',E13.6,T90,
     4'MI4=',E13.6,T116,'M4=',E13.6)
    5 FORMAT('0',T11,'C5=',E13.6,T37,'K5=',E13.6,T64,'E=',E13.6,T90,
     5'MI5=',E13.6,T116,'M5=',E13.6)
    6 FORMAT('0',T11,'C6=',E13.6,T37,'K6=',E13.6,T64,'F=',E13.6,T90,
     6'MJ2=',E13.6,T116,'M6=',E13.6)
    7 FORMAT('0',T11,'C7=',E13.6,T37,'K7=',E13.6,T64,'G=',E13.6,T90,
     7'MJ4=',E13.6,T116,'M7=',E13.6)
    8 FORMAT('0',T11,'C8=',E13.6,T37,'K8=',E13.6,T64,'H=',E13.6,T90,
     8'MH2=',E13.6,T116,'M8=',E13.6)
    9 FORMAT('0',T11,'C9=',E13.6,T37,'K9=',E13.6,T64,'P=',E13.6,T90,
     9'MH4=',E13.6,T116,'M9=',E13.6)
   10 FORMAT('0',T10,'C10=',E13.6,T36,'K10=',E13.6,T64,'Q=',E13.6,T90,
     2'P67=',E13.6)
   11 FORMAT('0',T10,'C11=',E13.6,T36,'K11=',E13.6,T64,'R=',E13.6,T90,
     2'P62=',E13.6)
   12 FORMAT('0',T10,'C12=',E13.6,T36,'K12=',E13.6,T64,'S=',E13.6,T90,
     2'P27=',E13.6)
   13 FORMAT('0',T10,'C13=',E13.6,T36,'K13=',E13.6,T64,'T=',E13.6,T90,
     2'P84=',E13.6)
   14 FORMAT('0',T10,'C14=',E13.6,T36,'K14=',E13.6,T64,'U=',E13.6,T90,
     2'P89=',E13.6)
   15 FORMAT('0',T10,'C15=',E13.6,T36,'K15=',E13.6,T64,'V=',E13.6,T90,
     2'P49=',E13.6)
   16 FORMAT('0',T10,'C16=',E13.6,T36,'K16=',E13.6,T64,'W=',E13.6)
   17 FORMAT('0',T10,'C17=',E13.6,T36,'K17=',E13.6,T64,'X=',E13.6)
```

```
18 FORMAT('0',T10,'C18=',E13.6,T36,'K18=',E13.6,T64,'Y=',E13.6)
19 FORMAT('0',T10,'C19=',E13.6,T36,'K19=',E13.6,T64,'Z=',E13.6)
20 FORMAT('0',T10,'C20=',E13.6,T36,'K20=',E13.6)
21 FORMAT('0',T10,'C21=',E13.6,T36,'K21=',E13.6)
22 FORMAT('0',T10,'C22=',E13.6,T36,'K22=',E13.6)
23 FORMAT('0',T10,'C23=',E13.6,T36,'K23=',E13.6)
24 FORMAT('0',T10,'C24=',E13.6,T36,'K24=',E13.6)
   WRITE(NWRITE,30)
   WRITE(NWRITE,40)
   WRITE(NWRITE,60)
   WRITE(NWRITE, 1) C1 ,K1 ,A ,MI1,M1
   WRITE(NWRITE, 2) C2 ,K2 ,B ,MI2,M2
   WRITE(NWRITE, 3) C3 ,K3 ,CC,MI3,M3
   WRITE(NWRITE, 4) C4 ,K4 ,D ,MI4,M4
   WRITE(NWRITE, 5) C5 ,K5 ,E ,MI5,M5
   WRITE(NWRITE, 6) C6 ,K6 ,F ,MJ2,M6
   WRITE(NWRITE, 7) C7 ,K7 ,G ,MJ4,M7
   WRITE(NWRITE, 8) C8 ,K8 ,H ,MH2,M8
   WRITE(NWRITE, 9) C9 ,K9 ,P ,MH4,M9
   WRITE(NWRITE,10) C10,K10,Q,P67
   WRITE(NWRITE,11) C11,K11,R,P62
   WRITE(NWRITE,12) C12,K12,S,P27
   WRITE(NWRITE,13) C13,K13,T,P84
   WRITE(NWRITE,14) C14,K14,U,P89
   WRITE(NWRITE,15) C15,K15,V,P49
   WRITE(NWRITE,16) C16,K16,W
   WRITE(NWRITE,17) C17,K17,X
   WRITE(NWRITE,18) C18,K18,Y
   WRITE(NWRITE,19) C19,K19,Z
   WRITE(NWRITE,20) C20,K20
   WRITE(NWRITE,21) C21,K21
   WRITE(NWRITE,22) C22,K22
   WRITE(NWRITE,23) C23,K23
   WRITE(NWRITE,24) C24,K24
C
C*********************************************************************************
C
   MR=M6*M8/(M6+M8)
   SPA=S*.5+A
   SMA=S*.5-A
   EPR=E+R
   TT=T*.5
   FMQ=F-Q
   M(1,1)=M1
   M(2,2)=M2+M4
   M(2,10)=-M4*A
   M(2,12)=M4*D
   M(2,14)=-M4*B
   M(2,15)=-M4*E
   M(3,3)=M3
   M(4,4)=M5
   M(5,5)=M7+M9
   M(5,10)=M9*P
   M(5,12)=-M9*Q
   M(5,16)=M9*B
   M(5,17)=M9*E
   M(6,6)=M1
   M(7,7)=M3
   M(8,8)=M5
   M(9,9)=MI1
```

```
      M(10,10)=M4*A*A+M9*P*P+MI2
      M(10,12)=-A*D*M4-F*Q*M9
      M(10,14)= M4*A*B-P67
      M(10,15)= M4*A*E
      M(10,16)=M9*P*B-P62
      M(10,17)=M9*P*E
      M(11,11)=MI3
      M(12,12)=M4*D*D+M9*Q*Q+MI4
      M(12,14)=-M4*D*B
      M(12,15)=-M4*D*E-P89
      M(12,16)=-M9*Q*B
      M(12,17)=-M9*Q*E-P84
      M(13,13)=MI5
      M(14,14)=M4*B*B+MR*P*P+MJ2
      M(14,15)=M4*B*E-MR*P*Q
      M(14,16)=-P*A*MR-P27
      M(14,17)=MR*P*D
      M(15,15)=M4*E*E+MR*Q*Q+MJ4
      M(15,16)= MR*Q*A
      M(15,17)=-MR*Q*D-P49
      M(16,16)=MR*A*A+M9*B*B+MH2
      M(16,17)=-A*D*MR+B*E*M9
      M(17,17)=MR*D*D+M9*E*E+MH4
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
      C(1,1)=C1+C2+C7+C8
      C(1,2)=-C1-C2
      C(1,9)=  W*(C8-C7)+U*(C2-C1)
      C(1,10)= SPA*C1-SMA*C2
      C(1,14)=-G*(C1+C2)
      C(2,2)=C1+C2+C3+C4+C5+C6
      C(2,3)=-C3-C4
      C(2,4)=-C5-C6
      C(2,9)=U*(C1-C2)
      C(2,10)=-SPA*(C1+C3)+SMA*(C2+C4)-A*(C5+C6)
      C(2,11)=U*(C3-C4)
      C(2,12)=TT*(C6-C5)
      C(2,13)=U*(C5-C6)
      C(2,14)=C1*G+C2*G-C3*H-C4*H-C5*B-C6*B
      C(2,15)=-EPR*(C5+C6)
      C(3,3)=C3+C4+C9+C10
      C(3,10)=SPA*C3-SMA*C4
      C(3,11)=W*(C10-C9)+U*(C4-C3)
      C(3,14)=C3*H+C4*H
      C(4,4)=C5+C6+C11+C12
      C(4,10)=A*(C5+C6)
      C(4,12)=TT*(C5-C6)
      C(4,13)=W*(C12-C11)+U*(C6-C5)
      C(4,14)=C5*B+C6*B
      C(4,15)=EPR*(C5+C6)
      C(5,5)=C15+C16+C19+C20+C23+C24
      C(5,6)=-C15-C16
      C(5,7)=-C19-C20
      C(5,8)=-C23-C24
      C(5,9)=V*(C15+C16)
      C(5,10)=CC*(C15+C16+C19+C20)+C23*P+C24*P
      C(5,11)=V*(C19+C20)
      C(5,12)=(C23+C24)*FMQ
      C(5,13)=V*(C23+C24)
```

```
      C(5,16)=-C15*G-C16*G+C19*H+C20*H+C23*B+C24*B
      C(5,17)=(C23+C24)*EPR
      C(6,6)=C13+C14+C15+C16
      C(6,9)=Y*(C13+C14)-V*(C15+C16)
      C(6,10)=-C15*CC-C16*CC
      C(6,16)=C15*G+C16*G
      C(7,7)=C17+C18+C19+C20
      C(7,10)=-C19*CC-C20*CC
      C(7,11)=Y*(C17+C18)-V*(C19+C20)
      C(7,16)=-C19*H-C20*H
      C(8,8)=C21+C22+C23+C24
      C(8,10)=-C23*P-C24*P
      C(8,12)=-FMQ*(C23+C24)
      C(8,13)=Y*(C21+C22)-V*(C23+C24)
      C(8,16)=-B*(C23+C24)
      C(8,17)=-(C23+C24)*EPR
      C(9,9)=W*W*(C7+C8)+Y*Y*(C13+C14)+U*U*(C1+C2)+V*V*(C15+C16)
      C(9,10)=-U*(SPA*C1+SMA*C2)+V*CC*(C15+C16)
      C(9,14)=U*G*(C1-C2)
      C(9,16)=-V*G*(C15+C16)
      C(10,10)=SPA*SPA*(C1+C3)+SMA*SMA*(C2+C4)+CC*CC*(C15+C16+C19+C20)+
     1A*A*(C5+C6)+P*P*(C23+C24)
      C(10,11)=-U*(SPA*C3+SMA*C4)+CC*V*(C19+C20)
      C(10,12)=A*TT*(C5-C6)+P*FMQ*(C23+C24)
      C(10,13)=A*U*(C6-C5)+P* V*(C23+C24)
      C(10,14)=-G*(SPA*C1-SMA*C2)+H*(SPA*C3-SMA*C4)+A*B*(C5+C6)
      C(10,15)=A* EPR  *(C5+C6)
      C(10,16)=-CC*G*(C15+C16)+CC*H*(C19+C20)+P*B*(C23+C24)
      C(10,17)=P*EPR  *(C23+C24)
      C(11,11)=W*W*(C9+C10)+Y*Y*(C17+C18)+U*U*(C3+C4)+V*V*(C19+C20)
      C(11,14)=U*H*(C4-C3)
      C(11,16)=V*H*(C19+C20)
      C(12,12)=TT*TT*(C5+C6)+FMQ*FMQ*(C23+C24)
      C(12,13)=-TT*U*(C5+C6)+FMQ*V*(C23+C24)
      C(12,14)=TT*B*(C5-C6)
      C(12,15)=TT*EPR*(C5-C6)
      C(12,16)=B*FMQ*(C23+C24)
      C(12,17)=EPR*FMQ*(C23+C24)
      C(13,13)=W*W*(C11+C12)+Y*Y*(C21+C22)+U*U*(C5+C6)+V*V*(C23+C24)
      C(13,14)=U*B*(C6-C5)
      C(13,15)=U*EPR*(C6-C5)
      C(13,16)=V*B*(C23+C24)
      C(13,17)=V*EPR*(C23+C24)
      C(14,14)=G*G*(C1+C2)+H*H*(C3+C4)+B*B*(C5+C6)
      C(14,15)=B*EPR*(C5+C6)
      C(15,15)=(C5+C6)*EPR*EPR
      C(16,16)=G*G*(C15+C16)+H*H*(C19+C20)+B*B*(C23+C24)
      C(16,17)=B*EPR*(C23+C24)
      C(17,17)=(C23+C24)*EPR*EPR
C
CKKKKKKKKKKK KKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKKK
C
      K(1,1)=K1+K2+K7+K8
      K(1,2)=-K1-K2
      K(1,9)=W*(K8-K7)+U*(K2-K1)
      K(1,10)=SPA*K1-SMA*K2
      K(1,14)=-G*(K1+K2)
      K(2,2)=K1+K2+K3+K4+K5+K6
      K(2,3)=-K3-K4
      K(2,4)=-K5-K6
```

```
K(2,9)=U*(K1-K2)
K(2,10)=-SPA*(K1+K3)+SMA*(K2+K4)-A*(K5+K6)
K(2,11)=U*(K3-K4)
K(2,12)=TT*(K6-K5)
K(2,13)=U*(K5-K6)
K(2,14)=K1*G+K2*G-K3*H-K4*H-K5*B-K6*B
K(2,15)=-EPR*(K5+K6)
K(3,3)=K3+K4+K9+K10
K(3,10)=SPA*K3-SMA*K4
K(3,11)=W*(K10-K9)+U*(K4-K3)
K(3,14)=K3*H+K4*H
K(4,4)=K5+K6+K11+K12
K(4,10)=A*(K5+K6)
K(4,12)=TT*(K5-K6)
K(4,13)=W*(K12-K11)+U*(K6-K5)
K(4,14)=K5*B+K6*B
K(4,15)=EPR*(K5+K6)
K(5,5)=K15+K16+K19+K20+K23+K24
K(5,6)=-K15-K16
K(5,7)=-K19-K20
K(5,8)=-K23-K24
K(5,9)=V*(K15+K16)
K(5,10)=CC*(K15+K16+K19+K20)+K23*P+K24*P
K(5,11)=V*(K19+K20)
K(5,12)=(K23+K24)*FMQ
K(5,13)=V*(K23+K24)
K(5,16)=-K15*G-K16*G+K19*H+K20*H+K23*B+K24*B
K(5,17)=(K23+K24)*EPR
K(6,6)=K13+K14+K15+K16
K(6,9)=Y*(K13+K14)-V*(K15+K16)
K(6,10)=-K15*CC-K16*CC
K(6,16)=K15*G+K16*G
K(7,7)=K17+K18+K19+K20
K(7,10)=-K19*CC-K20*CC
K(7,11)=Y*(K17+K18)-V*(K19+K20)
K(7,16)=-K19*H-K20*H
K(8,8)=K21+K22+K23+K24
K(8,10)=-K23*P-K24*P
K(8,12)=-(K23+K24)*FMQ
K(8,13)=Y*(K21+K22)-V*(K23+K24)
K(8,16)=-B*(K23+K24)
K(8,17)=-(K23+K24)*EPR
K(9,9)=W*W*(K7+K8)+Y*Y*(K13+K14)+U*U*(K1+K2)+V*V*(K15+K16)
K(9,10)=-U*(SPA*K1+SMA*K2)+V*CC*(K15+K16)
K(9,14)=U*G*(K1-K2)
K(9,16)=-V*G*(K15+K16)
K(10,10)=SPA*SPA*(K1+K3)+SMA*SMA*(K2+K4)+CC*CC*(K15+K16+K19+K20)
1+A*A*(K5+K6)+P*P*(K23+K24)
K(10,11)=-U*(SPA*K3+SMA*K4)+CC*V*(K19+K20)
K(10,12)=A*TT*(K5-K6)+P*FMQ*(K23+K24)
K(10,13)=A*U*(K6-K5)+P*V*(K23+K24)
K(10,14)=-G*(SPA*K1-SMA*K2)+H*(SPA*K3-SMA*K4)+A*B*(K5+K6)
K(10,15)= A*EPR*(K5+K6)
K(10,16)=-CC*G*(K15+K16)+CC*H*(K19+K20)+P*B*(K23+K24)
K(10,17)=P*EPR  *(K23+K24)
K(11,11)=W*W*(K9+K10)+ Y*Y*(K17+K18)+U*U*(K3+K4)+V*V*(K19+K20)
K(11,14)=U*H*(K4-K3)
K(11,16)=V*H*(K19+K20)
K(12,12)=TT*TT*(K5+K6)+FMQ*FMQ*(K23+K24)
K(12,13)=-TT*U*(K5+K6)+FMQ*V*(K23+K24)
```

```
      K(12,14)=TT*B*(K5-K6)
      K(12,15)=TT*EPR*(K5-K6)
      K(12,16)=B*FMQ  *(K23+K24)
      K(12,17)=EPR*FMQ*(K23+K24)
      K(13,13)=W*W*(K11+K12)+Y*Y*(K21+K22)+U*U*(K5+K6)+V*V*(K23+K24)
      K(13,14)=U*B*(K6-K5)
      K(13,15)=U*EPR  *(K6-K5)
      K(13,16)=V*B*(K23+K24)
      K(13,17)=V*EPR*(K23+K24)
      K(14,14)=G*G*(K1+K2)+H*H*(K3+K4)+B*B*(K5+K6)
      K(14,15)=B* EPR* (K5+K6)
      K(15,15)=(K5+K6)*EPR*EPR
      K(16,16)=G*G*(K15+K16)+H*H*(K19+K20)+B*B*(K23+K24)
      K(16,17)=B*EPR  *(K23+K24)
      K(17,17)=(K23+K24)*EPR*EPR
      DO 70 I=1,16
      N=I+1
      DO 80 J=N,17
      M(J,I)=M(I,J)
       C(J,I)=C(I,J)
   80 K(J,I)=K(I,J)
   70 CONTINUE
      RETURN
      END
```

```
BLOCK DATA
IMPLICIT REAL*8(A-H,O-Y)
COMMON/DYNAM/D(34,34),VALU(34,2)
COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
COMMON/A1/DL(17,17),AMA(17,17)
COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
COMMON/INOUT/NREAD,NWRITE,NPUNCH
DATA D,VALU/1224*0.D0/
DATA NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT/16,17,18,33,34,35,289,1156/
DATA NREAD,NWRITE,NPUNCH/5,6,7/
DATA AM,AC,AK/867*0.D0/
END
```

```
      SUBROUTINE FORCIN(NOREAD,NSIDE)
      IMPLICIT REAL*8(A-H,K,O-Y)
      REAL*8 DCOS,DSIN
      NAMELIST/FORCE/CVII,CVIII,CIX,CX,CXI,CXII,KVII,KVIII,KIX,KX,KXI,
     1KXII,AL,AR,WIDE,BT,GH1,GBER2,SPEED,FWL,FWR,RUNGL,RUNGR,NUMB
      COMMON/R1/TIME(10)
      COMMON/COSSIN/SAC(34,3)
      COMMON/MISC/WL,WR,FAST,STEPL,STEPR,AMPL,AMPR,FAZE,NSTEPS
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DIMENSION     CI(17),CII(17),CIII(17),SI(17),SII(17),SIII(17)
      EQUIVALENCE(SAC(1,1),CI),(SAC(1,2),CII),(SAC(1,3),CIII)
      EQUIVALENCE(SAC(18,1),SI),(SAC(18,2),SII),(SAC(18,3),SIII)
      IF(NOREAD.NE.0) GC TO 10
      READ(NREAD,FORCE)
      WL=FWL
      WR=FWR
      FAST=SPEED
      AMPL=AL
      AMPR=AR
      STEPL=RUNGL
      STEPR=RUNGR
      FAZE=BT
      NSTEPS=NUMB
   10 CONTINUE
      IF(NSIDE.EQ.2) GO TO 20
      T1=WL*GH1/FAST
      T2=WL*GBER2/FAST
      TIME(1)=DSIN(T1)
      TIME(2)=DCOS(T1)
      TIME(3)=DSIN(T2)
      TIME(4)=DCOS(T2)
      T1=WR*GH1/FAST
      T2=WR*GBER2/FAST
      TIME(5)=DSIN(BT)
      TIME(6)=DCOS(BT)
      TIME(7)=DSIN(BT+T1)
      TIME(8)=DCOS(BT+T1)
      TIME(9)=DSIN(BT+T2)
      TIME(10)=DCOS(BT+T2)
      AW=AL*WL
      CI(1)=AW*CVII
      CI(9)=-WIDE*CI(1)
      SI(1)=AL*KVII
      SI(9)=-WIDE*SI(1)
      CII(3)=AW*CIX
      CII(11)=-WIDE*CII(3)
      SII(3)=KIX*AL
      SII(11)=-WIDE*SII(3)
      CIII(4)=CXI*AW
      CIII(13)=-WIDE*CIII(4)
      SIII(4)=AL*KXI
      SIII(13)=-WIDE*SIII(4)
      RETURN
   20 AW=AR*WR
      CI(1)=AW*CVIII
      CI(9)=WIDE*CI(1)
      SI(1)=AR*KVIII
      SI(9)=WIDE*SI(1)
      CII(3)=AW*CX
      CII(11)=WIDE*CII(3)
```

```
SII(3)=AR*KX
SII(11)=WIDE*SII(3)
CIII(4)=AW*CXII
CIII(13)=WIDE*CIII(4)
SIII(4)=AR*KXII
SIII(13)=WIDE*SIII(4)
RETURN
END
```

```
C
C     ..............................................................DGELG002
C                                                                   DGELG003
C        SUBROUTINE DGELG (THIS IS ONE OF THE SUBROUTINES IN THE IBM
C        SYSTEM/360 SCIENTIFIC SUBROUTINE PACKAGE).
C                                                                   DGELG005
C        PURPOSE                                                    DGELG006
C           TO SOLVE A GENERAL SYSTEM OF SIMULTANEOUS LINEAR EQUATIONS. DGELG007
C                                                                   DGELG008
C        USAGE                                                      DGELG009
C           CALL DGELG(R,A,M,N,EPS,IER)                             DGELG010
C                                                                   DGELG011
C        DESCRIPTION OF PARAMETERS                                  DGELG012
C           R      - DOUBLE PRECISION M BY N RIGHT HAND SIDE MATRIX  DGELG013
C                    (DESTROYED). ON RETURN R CONTAINS THE SOLUTIONS DGELG014
C                    OF THE EQUATIONS.                              DGELG015
C           A      - DOUBLE PRECISION M BY M COEFFICIENT MATRIX     DGELG016
C                    (DESTROYED).                                   DGELG017
C           M      - THE NUMBER OF EQUATIONS IN THE SYSTEM.         DGELG018
C           N      - THE NUMBER OF RIGHT HAND SIDE VECTORS.         DGELG019
C           EPS    - SINGLE PRECISION INPUT CONSTANT WHICH IS USED AS DGELG020
C                    RELATIVE TOLERANCE FOR TEST ON LOSS OF         DGELG021
C                    SIGNIFICANCE.                                  DGELG022
C           IER    - RESULTING ERROR PARAMETER CODED AS FOLLOWS     DGELG023
C                    IER=0  - NO ERROR,                             DGELG024
C                    IER=-1 - NO RESULT BECAUSE OF M LESS THAN 1 OR DGELG025
C                             PIVOT ELEMENT AT ANY ELIMINATION STEP DGELG026
C                             EQUAL TO 0,                           DGELG027
C                    IER=K  - WARNING DUE TO POSSIBLE LOSS OF SIGNIFI- DGELG028
C                             CANCE INDICATED AT ELIMINATION STEP K+1, DGELG029
C                             WHERE PIVOT ELEMENT WAS LESS THAN OR  DGELG030
C                             EQUAL TO THE INTERNAL TOLERANCE EPS TIMES DGELG031
C                             ABSOLUTELY GREATEST ELEMENT OF MATRIX A. DGELG032
C                                                                   DGELG033
C        REMARKS                                                    DGELG034
C           INPUT MATRICES R AND A ARE ASSUMED TO BE STORED COLUMNWISE DGELG035
C           IN M*N RESP. M*M SUCCESSIVE STORAGE LOCATIONS. ON RETURN DGELG036
C           SOLUTION MATRIX R IS STORED COLUMNWISE TOO.             DGELG037
C           THE PROCEDURE GIVES RESULTS IF THE NUMBER OF EQUATIONS M IS DGELG038
C           GREATER THAN 0 AND PIVOT ELEMENTS AT ALL ELIMINATION STEPS DGELG039
C           ARE DIFFERENT FROM 0. HOWEVER WARNING IER=K - IF GIVEN - DGELG040
C           INDICATES POSSIBLE LOSS OF SIGNIFICANCE. IN CASE OF A WELL DGELG041
C           SCALED MATRIX A AND APPROPRIATE TOLERANCE EPS, IER=K MAY BE DGELG042
C           INTERPRETED THAT MATRIX A HAS THE RANK K. NO WARNING IS DGELG043
C           GIVEN IN CASE M=1.                                      DGELG044
C                                                                   DGELG045
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED              DGELG046
C           NONE                                                    DGELG047
C                                                                   DGELG048
C        METHOD                                                     DGELG049
C           SOLUTION IS DONE BY MEANS OF GAUSS-ELIMINATION WITH     DGELG050
C           COMPLETE PIVOTING.                                      DGELG051
C                                                                   DGELG052
C     ..............................................................DGELG053
C                                                                   DGELG054
      SUBROUTINE DGELG(R,A,M,N,EPS,IER)                             DGELG055
C                                                                   DGELG056
C                                                                   DGELG057
      DIMENSION A(1),R(1)                                           DGELG058
      DOUBLE PRECISION R,A,PIV,TB,TOL,PIVI                          DGELG059
```

```
      IF(M)23,23,1
C
C     SEARCH FOR GREATEST ELEMENT IN MATRIX A
    1 IER=0
      PIV=0.D0
      MM=M*M
      NM=N*M
      DO 3 L=1,MM
      TB=DABS(A(L))
      IF(TB-PIV)3,3,2
    2 PIV=TB
      I=L
    3 CONTINUE
      TOL=EPS*PIV
C     A(I) IS PIVOT ELEMENT. PIV CONTAINS THE ABSOLUTE VALUE OF A(I).
C
C
C     START ELIMINATION LOOP
      LST=1
      DO 17 K=1,M
C
C     TEST ON SINGULARITY
      IF(PIV)23,23,4
    4 IF(IER)7,5,7
    5 IF(PIV-TOL)6,6,7
    6 IER=K-1
    7 PIVI=1.D0/A(I)
      J=(I-1)/M
      I=I-J*M-K
      J=J+1-K
C     I+K IS ROW-INDEX, J+K COLUMN-INDEX OF PIVOT ELEMENT
C
C     PIVOT ROW REDUCTION AND ROW INTERCHANGE IN RIGHT HAND SIDE R
      DO 8 L=K,NM,M
      LL=L+I
      TB=PIVI*R(LL)
      R(LL)=R(L)
    8 R(L)=TB
C
C     IS ELIMINATION TERMINATED
      IF(K-M)9,18,18
C
C     COLUMN INTERCHANGE IN MATRIX A
    9 LEND=LST+M-K
      IF(J)12,12,10
   10 II=J*M
      DO 11 L=LST,LEND
      TB=A(L)
      LL=L+II
      A(L)=A(LL)
   11 A(LL)=TB
C
C     ROW INTERCHANGE AND PIVOT ROW REDUCTION IN MATRIX A
   12 DO 13 L=LST,MM,M
      LL=L+I
      TB=PIVI*A(LL)
      A(LL)=A(L)
   13 A(L)=TB
C
C     SAVE COLUMN INTERCHANGE INFORMATION
```

```
      A(LST)=J
C
C     ELEMENT REDUCTION AND NEXT PIVOT SEARCH
      PIV=0.D0
      LST=LST+1
      J=0
      DO 16 II=LST,LEND
      PIVI=-A(II)
      IST=II+M
      J=J+1
      DO 15 L=IST,MM,M
      LL=L-J
      A(L)=A(L)+PIVI*A(LL)
      TB=DABS(A(L))
      IF(TB-PIV)15,15,14
   14 PIV=TB
      I=L
   15 CONTINUE
      DO 16 L=K,NM,M
      LL=L+J
   16 R(LL)=R(LL)+PIVI*R(L)
   17 LST=LST+M
C     END OF ELIMINATION LOOP
C
C
C     BACK SUBSTITUTION AND BACK INTERCHANGE
   18 IF(M-1)23,22,19
   19 IST=MM+M
      LST=M+1
      DO 21 I=2,M
      II=LST-I
      IST=IST-LST
      L=IST-M
      L=A(L)+.5D0
      DO 21 J=II,NM,M
      TB=R(J)
      LL=J
      DO 20 K=IST,MM,M
      LL=LL+1
   20 TB=TB-A(K)*R(LL)
      K=J+L
      R(J)=R(K)
   21 R(K)=TB
   22 RETURN
C
C
C     ERROR RETURN
   23 IER=-1
      RETURN
      END
```

```
      SUBROUTINE WEQUAL
      IMPLICIT REAL*8(A-H,O-Y)
      REAL*8 DSQRT
      COMMON/DOUT/XCL(17),XSL(17),XCR(17),XSR(17)
      COMMON/MISC/WL,WR,FAST,STEPL,STEPR,AMPL,AMPR,FAZE,NSTEPS
      COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
      COMMON/INOUT/NREAD,NWRITE,NPUNCH
      DIMENSION AMPMAX(17)
      DO 21 I=1,N
      XSIN=XSL(I)+ XSR(I)
      XCOS=XCL(I)+XCR(I)
   21 AMPMAX(I)=DSQRT(XSIN*XSIN+XCOS*XCOS)
      WRITE(NWRITE,111)
      WRITE(NWRITE,222) FAZE
      WRITE(NWRITE, 1) AMPMAX( 1)
      WRITE(NWRITE, 2) AMPMAX( 2)
      WRITE(NWRITE, 3) AMPMAX( 3)
      WRITE(NWRITE, 4) AMPMAX( 4)
      WRITE(NWRITE, 5) AMPMAX( 5)
      WRITE(NWRITE, 6) AMPMAX( 6)
      WRITE(NWRITE, 7) AMPMAX( 7)
      WRITE(NWRITE, 8) AMPMAX( 8)
      WRITE(NWRITE, 9) AMPMAX( 9)
      WRITE(NWRITE,10) AMPMAX(10)
      WRITE(NWRITE,11) AMPMAX(11)
      WRITE(NWRITE,12) AMPMAX(12)
      WRITE(NWRITE,13) AMPMAX(13)
      WRITE(NWRITE,14) AMPMAX(14)
      WRITE(NWRITE,15) AMPMAX(15)
      WRITE(NWRITE,16) AMPMAX(16)
      WRITE(NWRITE,17) AMPMAX(17)
    1 FORMAT('0',56X,'    X1   = ',1P1D9.2)
    2 FORMAT('0',56X,'    X2   = ',1P1D9.2)
    3 FORMAT('0',56X,'    X3   = ',1P1D9.2)
    4 FORMAT('0',56X,'    X5   = ',1P1D9.2)
    5 FORMAT('0',56X,'    X7   = ',1P1D9.2)
    6 FORMAT('0',56X,'    Z1   = ',1P1D9.2)
    7 FORMAT('0',56X,'    Z3   = ',1P1D9.2)
    8 FORMAT('0',56X,'    Z5   = ',1P1D9.2)
    9 FORMAT('0',56X,'THETA1 = ',1P1D9.2)
   10 FORMAT('0',56X,'THETA2 = ',1P1D9.2)
   11 FORMAT('0',56X,'THETA3 = ',1P1D9.2)
   12 FORMAT('0',56X,'THETA4 = ',1P1D9.2)
   13 FORMAT('0',56X,'THETA5 = ',1P1D9.2)
   14 FORMAT('0',56X,'  PHI2   = ',1P1D9.2)
   15 FORMAT('0',56X,'  PHI4   = ',1P1D9.2)
   16 FORMAT('0',56X,'   XI2   = ',1P1D9.2)
   17 FORMAT('0',56X,'   XI4   = ',1P1D9.2)
  111 FORMAT('1',46X,'THE MAXIMUN AMPLITUDES (FEET,RADIANS) ARE'//)
  222 FORMAT('-',42X,'THE RELATIVE PHASE ANGLE(RAD.) BETWEEN SIDES IS'
     1/'0',60X,1P1D10.3///)
      RETURN
      END
```

```
BLOCK DATA
IMPLICIT REAL*8(A-H,O-Y)
COMMON/DYNAM/D(34,34),VALU(34,2)
COMMON/MATMCK/AM(17,17),AC(17,17),AK(17,17)
COMMON/A1/DL(17,17),AMA(17,17)
COMMON/ORDER/NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT
COMMON/INOUT/NREAD,NWRITE,NPUNCH
DATA D,VALU/1224*0.D0/
DATA NMO,N,NPO,NTMO,NT,NTPO,NN,NTNT/16,17,18,33,34,35,289,1156/
DATA NREAD,NWRITE,NPUNCH/5,6,7/
DATA AM,AC,AK/867*0.D0/
END
```

ACKNOWLEDGEMENTS

The author wishes to express his appreciation to Dr. Hugh S. Walker, Associate Professor of Mechanical Engineering at Kansas State University, for his help in the preparation of this thesis. The author is also indebted to Dr. Gerald R. Potts for his aid and encouragement, and to Mr. Charles Wilson for his valuable suggestions in the development of the computer programs. Special thanks is due the author's wife, Jo Anne, for her patience throughout, and especially for her help in typing the manuscript of this thesis.

VITA

Christopher Charles Chapman

Candidate for the Degree of

Master of Science

Thesis: A LINEAR, THREE-DIMENSIONAL MODEL FOR THE VIBRATIONS OF A SEMI-TRAILER TRUCK

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Clay Center, Kansas, May 22, 1947, the son of Leo R. and Irene E. Chapman

Education: Attended grade school in Broughton, Kansas; graduated from Clay County Community High School in 1965; received the Bachelor of Science degree, Cum Laude, from Kansas State University, with a major in Mechanical Engineering, in January, 1970; completed the requirements for the Master of Science degree in August, 1971.

Professional Experience: Worked the Summer of 1969 as Design Engineer at Hutchinson Manufacturing Co., Clay Center, Kansas; worked as a Graduate Research Assistant in the Mechanical Engineering Department at Kansas State University; belongs to the American Society of Mechanical Engineers (associate member).

Honorary Awards: Elected to Sigma Tau and Pi Tau Sigma at Kansas State University, 1967; elected to Scabbard and Blade at Kansas State University, 1967; received the title of Distinguished Military Student from the R.O.T.C.; received the Fischer Governor Scholarship in 1968.

A LINEAR, THREE-DIMENSIONAL MODEL FOR THE VIBRATIONS OF A SEMI-TRAILER TRUCK

by

CHRISTOPHER CHARLES CHAPMAN

B. S., Kansas State University, 1970

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Mechanical Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1971

Name: Christopher Charles Chapman Date of Degree: August, 1971

Institution: Kansas State University Location: Manhattan, Kansas

Title of Study: A LINEAR, THREE DIMENSIONAL MODEL FOR THE VIBRATIONS OF A SEMI-TRAILER TRUCK

Pages in Study: 110 Candidate for Degree of Master of Science

Major Field: Mechanical Engineering

Scope and Method of Study: After a number of simplifying assumptions had been made, a mathematical model was developed for simulation of a three dimensional semi-trailer truck. The linear, differential equations of motion were determined using the energy method and Lagrange's Equations.

Two methods of solution were described. The first was based on the reduced equation, which determines the transient solution and the steady state solution. The second was based on the undetermined coefficients method, which determines the steady state solution only.

Two digital computer programs, based on the two methods, were developed for determining the steady state responses of the vehicle due to sinusoidal inputs. The utility of the two programs was compared.

Findings and Conclusions: It was concluded that the three dimensional model provided a more comprehensive addition to simulation of the vibrations of a semi-trailer truck. However it is recommended that the accuracy of the model be determined by comparing it to "real" vehicles. This could be a subject for further study.

MAJOR PROFESSOR'S APPROVAL Hugh S. Walker