MCR-81-602
NAS8-33956

User
Guide

September 1982

Structural Dynamics Payload Loads Estimates

(NASA-CR-170682) STRUCTUAL DYNAMICS PAYLOAD LOADS ESTIMATES: USER GUIDE Final Report (Martin Marietta Aerospace) 179 p
HC A09/MF A01 CSCL 20K
N83-13496
Unclas
G3/39 01556

MARTIN MARIETTA

MCR-82-602
NAS8-33556

User Guide September 1982

STRUCTURAL DYNAMICS
PAYLOAD LOADS ESTIMATES

R. Engels

Authors:
T. G. Shanahan
R. C. Engels

Harry Harcrow

Approved:
Harry Harcrow
Program Manager

MARTIN MARIETTA CORPORATION
DENVER AEROSPACE
P.O. Box 179
Denver, CO 80201

PREFACE

This User Guide is submitted to the National Aeronautics and Space Administration's George C. Marshall Space Flight Center, Huntsville, Alabama, in response to the contract provisions of deliverable items associated with Structural Dynamics Payload Loads Estimates, Contract Number NAS8-33556.

The study took place during the period from August 1979 to October 1982 under the direction of Mr. W. Holland of MSFC, Huntsville, Alabama.

During this three year period the following documents were produced:

1. Methodology Assessment Report
   August 1980, MCR-80-553

2. Methodology Development Report
   August 1981, MCR-81-602

3. Final Report
   September 1982, MCR-82-601

4. User Guide
   September 1982, MCR-82-602

5. Monthly Progress Reports

The Final Report together with the User Guide are intended to be largely selfcontained. Chapter I of the User Guide deals with an overview of an Integration Scheme to Determine the Response of a Launch Vehicle with Multiple Payloads. Chapter II discusses the software package associated with the Integration Scheme together with several sample problems. We also discuss the short-cut version of the above mentioned integration technique. This User Guide concludes with a list of references and the listings of the subroutines.

The author wishes to thank D. Devers, H. Harcrow and G. Morosow for their constructive comments and for reviewing parts of the manuscript.

TABLE OF CONTENTS

CHAPTER I: An Integration Scheme to Determine the Response of a Launch Vehicle with Multiple Payloads.

1. INTRODUCTION

In Chapter II of Structural Dynamics Payload Loads Estimates, Final Report, June 1982, we discussed in detail a direct numerical integration scheme to determine the response of a launch vehicle with several payloads. The objective of this chapter is to briefly review the derivation of this integration scheme. This will help us to understand the development of the software package, to be discussed in Chapter II.

2. THE EQUATIONS OF MOTION

The objective of this section is the derivation of the equations of motion of the launch vehicle/payload(s) system. Figure 1 shows the free body diagrams of the booster B and the payloads P1 and P2. The booster and the payloads are connected to each other through the interface. We also consider superfluous interface coordinates. Indeed, often the booster interface contains more degrees of freedom than is necessary to attach the payloads. The reason for this is that the booster organization provides a set of interface restrained booster modes which can be used to accommodate many different payload configurations. It would be prohibitive to recalculate a set of cantilevered launch vehicle modes every time the interface with the payloads changes.

From the free body diagrams in Figure 1 we can easily write the uncoupled equations of motion for the booster B and the payloads P1 and P2.

$$\left[\begin{array}{c|c|c} M_B & & \\ \hline & M_{P1} & \\ \hline & & M_{P2} \end{array}\right] \left\{\begin{array}{c} \ddot{X}_B \\ \hline \ddot{X}_{P1} \\ \hline \ddot{X}_{P2} \end{array}\right\} + \left[\begin{array}{c|c|c} K_B & & \\ \hline & K_{P1} & \\ \hline & & K_{P2} \end{array}\right] \left\{\begin{array}{c} X_B \\ \hline X_{P1} \\ \hline X_{P2} \end{array}\right\}$$

$$= \left\{\begin{array}{c} F_B \\ \hline F_{P1} \\ \hline F_{P2} \end{array}\right\} + \left\{\begin{array}{c} R_B \\ \hline R_{P1} \\ \hline R_{P2} \end{array}\right\} \qquad (1)$$

Several well known procedures exist to couple the above three equations[1]. The objective is to eliminate the redundant interface displacements and in the process also the unknown reaction vectors. One such technique uses so-called "cantilevered" booster and payload displacements[4]. To set the stage, let us note that we can partition the vectors $\{X_B\}$, $\{X_{P1}\}$, $\{X_{P2}\}$, $\{F_B\}$ etc. as follows:

$$\{X_B\} = \left\{\begin{array}{c} X_N^B \\ \hline X_{I1}^B \\ \hline X_{I2}^B \\ \hline X_{IS}^B \end{array}\right\} , \qquad X_{P1} = \left\{\begin{array}{c} X_N^{P1} \\ \hline X_{I1}^{P2} \end{array}\right\}$$

$$\{X_{P2}\} = \left\{\begin{array}{c} X_N^{P2} \\ \hline X_{I2}^{P2} \end{array}\right\} \quad , \quad \{F_B\} = \left\{\begin{array}{c} F_N^B \\ \hline 0 \\ \hline 0 \\ \hline 0 \end{array}\right\}$$

$$\{F_{P1}\} = \left\{\begin{array}{c} 0 \\ \hline 0 \end{array}\right\} \quad , \quad \{F_{P2}\} = \left\{\begin{array}{c} 0 \\ \hline 0 \end{array}\right\} \tag{2}$$

$$\{R_B\} = \left\{\begin{array}{c} 0 \\ \hline R_1 \\ \hline R_2 \\ \hline 0 \end{array}\right\} \quad , \quad \{R_{P1}\} = \left\{\begin{array}{c} 0 \\ \hline -R_1 \end{array}\right\}$$

$$\{R_{P2}\} = \left\{\begin{array}{c} 0 \\ \hline -R_2 \end{array}\right\}$$

Similar partitions can be written for the velocities and the accelerations. In the above equations we assumed that the external forces only act at the non-interface booster degrees of freedom. This assumption is only made for convenience. Also it should be noted that this development is not limited to two payloads.

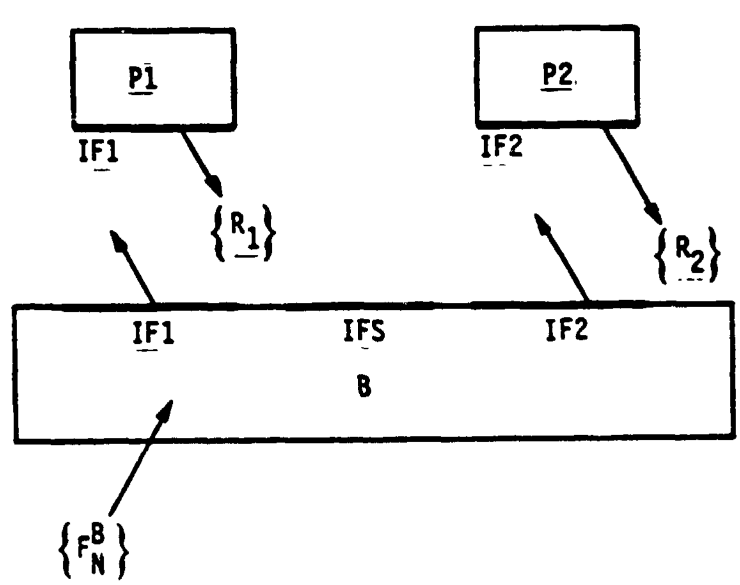


Figure 3 Free-Body Diagrams of Booster B and Payloads P1 and P2

Next, it is easy to show that $\{X_N^B\}$, $\{X_N^{P1}\}$, and $\{X_N^{P2}\}$ can be written as follows:

$$
\begin{aligned}
\{X_N^B\} &= -[K_{NN}^B]^{-1}\,[K_{NI}^B]\,\{X_I^B\} + \{\bar{X}_N^B\} \\
\{X_N^{P1}\} &= -[K_{NN}^{P1}]^{-1}\,[K_{NI}^{P1}]\,\{X_{I1}^{P1}\} + \{\bar{X}_N^{P1}\} \qquad (3)\\
\{X_N^{P1}\} &= -[K_{NN}^{P2}]^{-1}\,[K_{NI}^{P2}]\,\{X_{I1}^{P2}\} + \{\bar{X}_N^{P2}\}
\end{aligned}
$$

Using Equations (3) and the fact that

$$
\{X_{I1}^B\} = \{X_{I1}^{P1}\} \quad , \quad \{X_{I2}^B\} = \{X_{I2}^{P2}\} \qquad (4)
$$

we can form the following transformation,

$$
\begin{Bmatrix} X_B \\ \\ X_{P1} \\ \\ X_{P2} \end{Bmatrix} =
\left[\begin{array}{c|c|c|c}
I_B & T_B & 0_1 & 0_2 \\ \hline
 & 0_3 \quad 0_4 & & \\
0_5 & T_{P1} & I_{P1} & 0_6 \\
 & 0_7 \quad 0_8 & & \\ \hline
 & 0_9 \quad 0_{10} & & \\
0_{11} & T_{P2} & 0_{12} & I_{P2} \\
 & 0_{13} \quad 0_{14} & &
\end{array}\right]
\begin{Bmatrix} \bar{X}_N^B \\ \hline X_I^B \\ \hline \bar{X}_N^{P1} \\ \hline \bar{X}_N^{P2} \end{Bmatrix} \qquad (5)
$$

where,

$$
[illegible] = \left[\begin{array}{ccc} \multicolumn{3}{c}{-K_{NN}^{B^{-1}} K_{NI}^B} \\ \hline I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{array}\right] \; (BxIF) \; , \qquad
X_I^B = \begin{Bmatrix} X_{I1}^B \\ X_{I2}^B \\ X_{IS}^B \end{Bmatrix}
\begin{array}{l} (1xIF1) \\ (1xIF2) \\ (1xIFS) \end{array} \qquad (6)
$$

$$T_{P1} = \begin{bmatrix} -K_{NN}^{P1^{-1}} K_{NI}^{P1} \\ \hline I \end{bmatrix} \quad (P1xIF1), \qquad I_{P1} = \begin{bmatrix} I \\ \hline 0 \end{bmatrix} \quad (P1xNP1)$$

$$T_{P2} = \begin{bmatrix} -K_{NN}^{P2^{-1}} K_{NI}^{P2} \\ \hline I \end{bmatrix} \quad (P2xIF2), \qquad I_{P2} = \begin{bmatrix} I \\ \hline 0 \end{bmatrix} \quad (P2xNP2)$$

$$I_B = \begin{bmatrix} I \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (BxNB)$$

<u>NOTE</u>:

IF = IF1 + IF2 + IFS

B = NB + IF

P1 = NP1 + IF1

P2 = NP2 + NP2

also, the zero matrices 0_1, 0_2, 0_3, 0_4, 0_5, 0_6, 0_7, 0_8, 0_9, 0_{10}, 0_{11}, 0_{12}, 0_{13}, and 0_{14} have dimensions BxNP1, BxNP2, NP1xIF2, NP1xIFS, P1xNB, P1xNP2, IF1xIF2, IF1xIFS, NP2xIF1, NP2xIFS, P2xNB, P2xNP1, IF2xIF1 and IF2xIF2 respectively.

Taking into account that the reactions between interfaces are equal but opposite we can substitute transformation (5) into Equation (1) and then pre-multiply by the transpose of the transformation matrix. This will eliminate the redundant interface displacement vectors as well as the reaction vectors. The resulting equations are the coupled discrete equations of motion and can be written as:

$$\left[\begin{array}{c|ccc|c|c}
I_B^T M_B I_B & & I_B^T M_B T_B & & 0 & 0 \\
\hline
 & & & & T_{P1}^T M_{P1} I_{P1} & 0 \\
T_B^T M_B I_B & & M_{II} & & 0 & T_{P2}^T M_{P2} I_{P2} \\
 & & & & 0 & 0 \\
\hline
0 & I_{P1}^T M_{P1} T_{P1} & 0 & 0 & I_{P1}^T M_{P1} I_{P1} & 0 \\
\hline
0 & 0 & I_{P2}^T M_{P2} T_{P2} & 0 & 0 & I_{P2}^T M_{P2} I_{P2}
\end{array}\right]
\left\{\begin{array}{c}
\ddot{\bar{X}}_N^B \\ \hline \ddot{X}_I^B \\ \hline \ddot{\bar{X}}_N^{P1} \\ \hline \ddot{\bar{X}}_N^{P2}
\end{array}\right\} +$$

$$
+\left[\begin{array}{c|c|c|c}
I_B^T K_B I_B & 0 & 0 & 0 \\ \hline
0 & K_{II} & 0 & 0 \\ \hline
0 & 0 & I_{P1}^T M_{P1} I_{P1} & 0 \\ \hline
0 & 0 & 0 & I_{P2}^T M_{P2} I_{P2}
\end{array}\right]
\left\{\begin{array}{c}
X_N^B \\ \hline X_I^B \\ \hline X_N^{P1} \\ \hline X_N^{P2}
\end{array}\right\}
=
\left\{\begin{array}{c}
I_B^T F_B \\ \hline I_B^T F_B \\ \hline 0 \\ \hline 0
\end{array}\right\}
\tag{7}
$$

where,

$$
\begin{aligned}
M_{II} &= T_B^T M_B T_B + \begin{bmatrix} T_{P1}^T M_{P1} T_{P1} & 0 & 0 \\ 0 & T_{P2}^T M_{P2} T_{P2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \\
K_{II} &= T_B^T K_B T_B + \begin{bmatrix} T_{P1}^T K_{P1} T_{P1} & 0 & 0 \\ 0 & T_{P2}^T K_{P2} T_{P2} & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}
\tag{8}
$$

and it can be shown that $[I_B^T K_B T_B]$, $[I_{P1}^T K_{P1} T_{P1}]$ and $[I_{P2}^T K_{P2} T_{P2}]$ always vanish and $[T_B^T K_B T_B]$, $[T_{P1}^T K_{P1} T_{P1}]$ and $[T_{P2}^T K_{P2} T_{P2}]$ vanish when the corresponding interface is determinate

Next, we can introduce the interface restrained modes, i.e.

$$
\begin{aligned}
[\bar{\Phi}_N^B]\,\{\bar{q}_N^B\} &= \{\bar{x}_N^B\} , \\
[\bar{\Phi}_N^{P1}]\,\{\bar{q}_N^{P1}\} &= \{\bar{x}_N^{P1}\} , \\
[\bar{\Phi}_N^{P2}]\,\{\bar{q}_N^{P2}\} &= \{\bar{x}_N^{P2}\}
\end{aligned}
\tag{9}
$$

These modes can be truncated according to a predetermined cut-off frequency, thereby reducing the size of Equation (7). Also, we can solve the interface eigenvalue problem for $[M_{II}]$ and $[K_{II}]$, to get the interface modes $[\Phi_I]$ and frequencies $[\omega_I]$. Substituting Equations (9) into equations (7) and introducing the interface modes we obtain:

$$
\begin{bmatrix} I & B^T & 0 \\ B & I & P \\ 0 & P^T & I \end{bmatrix}
\begin{Bmatrix} \ddot{\bar{q}}_N^B \\ \ddot{q}_I^B \\ \ddot{\bar{q}}_N^P \end{Bmatrix}
+
\begin{bmatrix} 2\bar{\zeta}_B\bar{\omega}_B & & \\ & 2\zeta_I\omega_I & \\ & & 2\bar{\zeta}_P\bar{\omega}_P \end{bmatrix}
\begin{Bmatrix} \dot{\bar{q}}_N^B \\ \dot{q}_I^B \\ \dot{\bar{q}}_N^P \end{Bmatrix}
+
$$

$$
+ \begin{bmatrix} \bar{\omega}_B^2 & & \\ & \omega_I^2 & \\ & & \bar{\omega}_P^2 \end{bmatrix}
\begin{Bmatrix} \bar{q}_N^B \\ q_I^B \\ \bar{q}_N^P \end{Bmatrix}
=
\begin{Bmatrix} \bar{\Phi}_N^{B^T} I_B^T F_B \\ \Phi_I^T T_B^T F_B \\ 0 \end{Bmatrix}
\tag{10}
$$

where

$$
P = \Phi_I^{B^T} \begin{bmatrix} T_{P1}^T M_{P1} I_{P1} \bar{\Phi}_N^{P1} & 0 \\ 0 & T_{P2}^T M_{P2} I_{P2} \bar{\Phi}_N^{P2} \\ 0 & 0 \end{bmatrix} , \quad \left\{ \bar{q}_N^P \right\} = \begin{Bmatrix} \bar{q}_N^{P1} \\ \bar{q}_N^{P2} \end{Bmatrix}
$$

$$
B = \Phi_I^{B^T} T_B^T M_B I_B \bar{\Phi}_N^B
$$

$$
\bar{\omega}_P^2 = \begin{bmatrix} \bar{\omega}_{P1}^2 & \\ & \bar{\omega}_{P2}^2 \end{bmatrix}
$$

Note that we introduced modal damping. Although the booster and the payload modes can be truncated, it is imperative not to truncate any interface definition. This is of special significance when the interface modes are introduced. The motivation for this will be explained later.

3. THE NUMERICAL INTEGRATION SCHEME

A common approach to the solution of Equation (10) is to decouple these equations by solving an undamped eigenvalue problem for the set of system equations. In order to avoid this rather costly eigenvalue problem we shall attempt to directly numerically integrate Equation (10). The special structure of Equation (10) may lead to a very economical solution. Indeed, the only time consuming operations involve the matrices [B] and [P] . However, both those matrices have row dimensions equal to IF which usually is relatively small.

As indicated in reference 3, the Newmark-Chan-Beta numerical scheme is very well suited to solve this problem. Not only can it take complete advantage of the special structure of Equation (10) but it also allows us to track the highest frequency in the applied force instead of the highest frequency in the system without becoming numerically unstable.

The basic equations can be written as follows:

$$\left[\begin{array}{c|c|c} D_1 & B^T & 0 \\ \hline B & D_2 & P \\ \hline 0 & P^T & D_3 \end{array}\right] \left\{\begin{array}{c} \ddot{\bar{q}}^B_{Ni+1} \\ \hline \ddot{q}^B_{Ii+1} \\ \hline \ddot{\bar{q}}^P_{Ni+1} \end{array}\right\} = \left\{\begin{array}{c} f_{Bi} \\ \hline f_{Ii} \\ \hline f_{Pi} \end{array}\right\} \tag{11}$$

with

$$\begin{aligned} f_{Bi} &= \bar{\phi}_N^{B^T} I_B^T F_{Bi+1} - D_4 \dot{\bar{q}}^B_{Ni} - D_7 \ddot{\bar{q}}^B_{Ni} - \bar{\omega}_B^2 \bar{q}^B_{Ni} \\ f_{Ii} &= \phi_I^{B^T} \cdot T_B^T \cdot F_{Bi+1} - D_5 \dot{q}^B_{Ii} - D_8 \ddot{q}^B_{Ii} \quad \omega_I^2 q^B_{Ii} \\ f_{Pi} &= - D_6 \dot{\bar{q}}^P_{Ni} - D_9 \ddot{\bar{q}}^P_{Ni} - \bar{\omega}_P^2 \bar{q}^P_{Ni} \end{aligned} \tag{12}$$

$$\begin{aligned} D_1 &= I + 2\gamma h \bar{\zeta}_B \bar{\omega}_B + \beta h^2 \bar{\omega}_B^2 \\ D_2 &= I + 2\gamma h \zeta_I \omega_I + \beta h^2 \omega_I^2 \\ D_3 &= I + 2\gamma h \bar{\zeta}_P \bar{\omega}_P + \beta h^2 \bar{\omega}_P^2 \\ D_4 &= 2\bar{\zeta}_B \bar{\omega}_B + h \bar{\omega}_B^2 \\ D_5 &= 2\zeta_I \omega_I + h \omega_I^2 \end{aligned} \tag{13}$$

$$D_6 = 2\bar{\zeta}_P \bar{\omega}_P + h\bar{\omega}_P^2$$

$$D_7 = 2(1-\gamma)h\bar{\zeta}_B\bar{\omega}_B + (0.5-\beta)h^2\bar{\omega}_B^2$$

$$D_8 = 2(1-\gamma)h\,\zeta_I\omega_I - (0.5-\beta)h^2\omega_I^2$$

$$D_9 = 2(1-\gamma)h\bar{\zeta}_P\bar{\omega}_P + (0.5-\beta)h^2\bar{\omega}_P^2$$

Furthermore, if we premultiply Equation (11) by

$\left[-BD_1^{-1} \;\middle|\; I \;\middle|\; -PD_3^{-1}\right]$, we obtain

$$\left\{\ddot{x}^B_{Ii+1}\right\} = A_1\left\{A_2 f_{Bi} + f_{Ii} + A_3 f_{Pi}\right\} \tag{14}$$

with

$$A_1 = \left[D_2 + A_2B^T + A_3P^T\right]^{-1} \tag{15}$$

$$A_2 = -BD_1^{-1}$$

$$A_3 = -PD_3^{-1}$$

also,

$$\bar{q}^B_{Ni+1} = D_1^{-1}f_{Bi} + A_2^T\ddot{x}^B_{Ii+1} \tag{16}$$

$$\bar{q}^P_{Ni+1} = D_3^{-1}f_{Pi} + A_3^T\ddot{x}^B_{Ii+1}$$

Once the acceleration at time i+1 is known, we can find the corresponding velocity and displacement vectors from,

$$\{\dot{q}_{i+1}\} = \{\dot{q}_i\} + (1-\gamma)h\{\ddot{q}_i\} + \gamma h\{\ddot{q}_{i+1}\}$$

$$\{q_{i+1}\} = \{q_i\} + h\{\dot{q}_i\} + (0.5-\beta)h^2\{\ddot{q}_i\} + \beta h^2\{\ddot{q}_{i+1}\} \tag{17}$$

It should be noted that matrix A_2 for example, contains zero partitions of which advantage can be taken. Also, the usual triangular decomposition necessary to solve Equation (11) can be replaced by the inversion of an IF x IF matrix in Equation (15). The final recursive relations then, are given by Equations (12), (14), (16), and (17).

ORIGINAL PAGE IS
OF POOR QUALITY

4. THE LOAD TRANSFORMATION

Hruda and Jones[5], introduced a load transformation which is consistent with modal synthesis techniques. In reference 3 we adapted this technique to the case of the direct integration scheme. When more than one payload is considered and when there are superfluous interface coordinates, the load transformations are derived in a similar manner. The pertinent equations can be presented as follows:

$$\{L_j\} = [k\Psi]_j \; \{x_{Pj}\} \tag{18}$$

which is the load equation for an elementary member or a set of these members. Note that such a member could be part of any payload. For example, if the member belongs to payload P1 then $\{x_{Pj}\} = \{x_{P1}\}$.

Furthermore, if the acceleration approach is used, we can write the load vector as,

$$\{I_j\} = [LT1]_j \begin{Bmatrix} \bar{q}_N^{Pj} \\ --- \\ x_{Ij}^B \end{Bmatrix} + [LT2]_j \; \{x_{Ij}^B\} \tag{19}$$

with

$$\begin{aligned} [LT1]_j &= (-[k\Psi]_j \, [\beta]_j \qquad - [k\Psi]_j \, [\bar{\delta}]_j) \\ [LT2]_j &= [k\Psi]_j \, [\varepsilon]_j \end{aligned} \tag{20}$$

and

$$\begin{aligned} [\beta]_j &= [I_{Pj}] \quad [\bar{\Phi}_N^{Pj}] \quad [\bar{\omega}_P^2]^{-1} \\ [\bar{\delta}]_j &= [I_{Pj}] \quad [E_{Pj}] \quad \times \quad [I_{Pj}^T M_{Pj} T_{Pj}] \\ [\varepsilon]_j &= [T_{Pj}] \quad , \qquad [E_{Pj}] = [I_{Pj}^T K_{Pj} I_{Pj}] \end{aligned} \tag{21}$$

It should be pointed out that it can be shown that if we keep all interface modes that $\{x_{Ij}\}$ in Equation (19) can be used directly and need not be written in terms of accelerations and applied forces. Also, observe that the payload organization can easily save the matrices $[\beta]_j$ $[\delta]_j$ and $[\varepsilon]_j$, so that any member in the payload can now be investigated without recalculating these matrices. In addition, no system eigenvalues and eigenvectors are involved.

5. THE SHORT-CUT VERSION

In Chapter III of the Final Report we developed a short-cut version of the previously derived full-scale numerical integration scheme. In this section we wish to review the basic results in support of the develpment of the software package. The approach is based on estimating the size of the feedback from the payload response into the booster. First, we shall derive the so-called coupled base motion equations for the system. These equations still represent a set of accurate full-scale equations of motion. One possible approach is to completely neglect the feedback of the payload into the booster. This leads to a technique known as the direct base drive or open loop technique as discussed in Chapter I section A3f and B3 of the Final Report.

In many cases this technique leads to acceptable results. Instead of completely neglecting the feedback, we shall subject the magnitude of this feedback to a criterion. The result will be a method which, depending on the structure at hand, will vasilate between a full-up coupled base motion technique and a direct base drive method.

First, note that the equations of motion for the coupled base motion technique are derived in Chapter I A3.f. in the Final Report. Following that derivation we can start by writing down the coupled set of equations of motion as given by Equation (10) of this User Guide :

$$\left[\begin{array}{c|c|c} I & B^T & 0 \\ \hline B & M_{II} & P \\ \hline 0 & P^T & I \end{array}\right] \left\{\begin{array}{c} \ddot{\bar{q}}_N^B \\ \hline \ddot{x}_I \\ \hline \ddot{\bar{q}}_N^P \end{array}\right\} + \left[\begin{array}{c|c|c} 2\bar{\zeta}_B\bar{\omega}_B & & \\ \hline & D_{II} & \\ \hline & & 2\bar{\zeta}_P\bar{\omega}_P \end{array}\right] \left\{\begin{array}{c} \dot{\bar{q}}_N^B \\ \hline \dot{x}_I^B \\ \hline \dot{\bar{q}}_N^P \end{array}\right\} +$$

$$\left[\begin{array}{c|c|c} \bar{\omega}_B^2 & & \\ \hline & K_{II} & \\ \hline & & \bar{\omega}_P^2 \end{array}\right] \left\{\begin{array}{c} \bar{q}_N^B \\ \hline x_I^B \\ \hline \bar{q}_N^P \end{array}\right\} = \left\{\begin{array}{c} \bar{\Phi}_N^{B\,T}\,\bar{T}_E^T F_E \\ \hline \bar{T}_E^T F_B \\ \hline 0 \end{array}\right\}. \tag{22}$$

where,

ORIGINAL PAGE IS
OF POOR QUALITY

$$M_{II} = T_B^T M_B T_B + \begin{bmatrix} T_{P1}^T M_{P1} T_{P1} & 0 & 0 \\ 0 & T_{P2}^T M_{P2} T_{P2} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$K_{II} = T_B^T K_B T_B + \begin{bmatrix} T_{P1}^T K_{P1} T_{P1} & 0 & 0 \\ 0 & T_{P2}^T K_{P2} T_{P2} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$P = \left[\begin{array}{c|c} T_{P1}^T M_{P1} T_{P1} \bar{\Phi}_N^{P1} & 0 \\ 0 & T_{P2}^T M_{P2} T_{P2} \bar{\Phi}_N^{P2} \\ 0 & 0 \end{array}\right], \qquad \{\bar{q}_N^P\} = \left\{\begin{array}{c} \bar{q}_N^{P1} \\ \hline \bar{q}_N^{P2} \end{array}\right\}$$

$$B = T_B^T M_B I_B \bar{\Phi}_N^B, \qquad \bar{\omega}_N^2 = \left[\begin{array}{c|c} \bar{\omega}_{P1}^2 & \\ \hline & \bar{\omega}_{P2}^2 \end{array}\right]$$

Following the philosophy of a base motion technique we first solve the following set of equations,

$$\left[\begin{array}{c|c} I & B^T \\ \hline B & T_B^T M_B T_B \end{array}\right] \left\{\begin{array}{c} \ddot{\bar{q}}_{NO}^B \\ \hline \ddot{x}_{IO}^B \end{array}\right\} + \left[\begin{array}{c|c} 2\bar{\zeta}_B \bar{\omega}_B & 0 \\ \hline 0 & 0 \end{array}\right] \left\{\begin{array}{c} \dot{\bar{q}}_{NO}^B \\ \hline \dot{x}_{IO}^B \end{array}\right\}$$

$$+ \left[\begin{array}{c|c} \bar{\omega}_B^2 & 0 \\ \hline 0 & T_B^T K_B T_B \end{array}\right] \left\{\begin{array}{c} \bar{q}_{NO}^B \\ \hline x_{IO}^B \end{array}\right\} = \left\{\begin{array}{c} \bar{\Phi}_N^{B^T} I_B^T F_B \\ \hline T_B^T F_B \end{array}\right\}$$

This set of equations represent the equations of motion for the booster without payload(s). Note that one could also load the booster with a standard payload which would represent a kind of average payload.

The solution of Equation (24) is the same for all payload configurations and does not change as long as the forcing function and the booster model do not change. A modified Newmark-Chan-Beta numerical integration scheme can be used to obtain the solution of (24):

$$\{\dot{q}\}_{0i+1} = \{\dot{q}\}_{0i} + (1-\gamma)h\{\ddot{q}\}_{0i} + \gamma h\{\ddot{q}\}_{0i+1}$$
$$\{q\}_{0i+1} = \{q\}_{0i} + h\{\dot{q}\}_{0i} + (0.5-\beta)h^2\{\ddot{q}\}_{0i} + h^2\beta\{\ddot{q}\}_{0i+1} \tag{25}$$

$$\left[\begin{array}{c|c} D_1 & B^T \\ \hline B & D_2 \end{array}\right]\left\{\begin{array}{c} \ddot{\bar{q}}^B_{N0i+1} \\ \hline \ddot{x}^B_{I0i+1} \end{array}\right\} = \left\{\begin{array}{c} f_{Bi} \\ \hline f_{Ii} \end{array}\right\} \tag{26}$$

where

$$f_{Bi} = \bar{\Phi}^{B^T}_N I^T_B F_{Bi+1} - D_3\dot{\bar{q}}^B_{N0i} - D_5\ddot{\bar{q}}^B_{N0i} - \bar{\omega}^2_B\,\bar{q}^B_{N0i}$$

$$f_{Ii} = T^T_B F_{Bi+1} - D_4\dot{x}^B_{I0i} - D_6\ddot{x}^B_{N0i} - T^T_B K_B T_B x^B_{N0i} \tag{27}$$

Also,

$$D_1 = I + 2\gamma h\bar{\zeta}_B\,\bar{\omega}_B + \beta h^2\bar{\omega}^2_B$$

$$D_1 = T^T_B M_B T_B + T^T_B K_B T_B h^2\beta \tag{28}$$

$$D_3 = 2\bar{\zeta}_B\bar{\omega}_B + \bar{\omega}^2_B h^2$$

ORIGINAL PAGE IS
OF POOR QUALITY

$$D_4 = T_B^T K_B T_B h^2$$

$$D_5 = 2\bar{\zeta}_B \bar{\omega}_B \, (1-\gamma) h + (0.5-\beta) h^2 \, \bar{\omega}_B^2 \qquad (28)$$

$$D_6 = T_B^T K_B T_B (0.5-\beta) h^2$$

Premultiplying (26) by $[\ -BD_1^{-1} \mid I]$ yields

$$\ddot{x}^B_{IOi+1} = A_1 \, [\ f_{Ii} + A_2 f_{Bi}\ ] \qquad (29)$$

with

$$A_2 = -B\ D_1^{-1}$$

$$A_1 = [\ D_2 + A_2\ B^T\ ]^{-1}$$

also,

$$\ddot{\bar{q}}^B_{NOi+1} = D_1^{-1}\ f_{Bi} + A_2^T\ \ddot{x}^B_{NOi+1} \qquad (30)$$

The equations to use then are Equations (25, 27, 29, 30). The quantities to save are $\{x_{IO}\}$, $\{\dot{x}_{IO}\}$ and $\{\ddot{x}_{IO}\}$.

Returning to Equation (22) we can write:

$$\left\{\bar{q}_N^B\right\} = \left\{\bar{q}_{NO}^B\right\} + \left\{\bar{q}_{NR}^B\right\}$$

$$\left\{x_I^B\right\} = \left\{x_{IO}^B\right\} + \left\{x_{IR}^B\right\} \tag{31}$$

The residual quantities $\left\{\bar{q}_{NR}^B\right\}$ and $\left\{x_{IR}^B\right\}$ are clearly due to the presence of the payload(s), i.e. they are the feedback of the payload(s) back into the booster. Let us write the following vector equation,

$$\begin{Bmatrix} \bar{q}_N^B \\ x_I^B \\ \bar{q}_N^P \end{Bmatrix} = \begin{Bmatrix} \bar{q}_{NO}^B \\ x_{IO}^B \\ 0 \end{Bmatrix} + \begin{Bmatrix} \bar{q}_{NR}^B \\ x_{IR}^B \\ \bar{q}_N^P \end{Bmatrix} \tag{32}$$

Substituting Equation (31) into Equation (22) and taking into account Equations (23 -24) leads to the coupled base motion equations,

$$\left[\begin{array}{c|c|c} I & B^T & 0 \\ \hline B & M_{II} & P \\ \hline 0 & P^T & I \end{array}\right] \left\{\begin{array}{c} \ddot{\bar{q}}_{NR}^B \\ \hline \ddot{x}_{IR}^B \\ \hline \ddot{\bar{q}}_N^P \end{array}\right\} + \left[\begin{array}{c|c|c} 2\bar{\zeta}_B\bar{\omega}_B & 0 & 0 \\ \hline 0 & D_{II} & 0 \\ \hline 0 & 0 & 2\bar{\zeta}_P\bar{\omega}_P \end{array}\right] \left\{\begin{array}{c} \dot{\bar{q}}_{NR}^B \\ \hline \dot{x}_{IR}^B \\ \hline \dot{\bar{q}}_N^P \end{array}\right\}$$

$$+ \left[\begin{array}{c|c|c} \bar{\omega}_B^2 & 0 & 0 \\ \hline 0 & K_{II} & 0 \\ \hline 0 & 0 & \bar{\omega}_P^2 \end{array}\right] \left\{\begin{array}{c} \bar{q}_{NR}^B \\ \hline x_{IR}^B \\ \hline \bar{q}_N^P \end{array}\right\} = \left\{\begin{array}{c} 0 \\ \hline -MP_{II}\ddot{x}_{IO}^B - D_{II}\dot{x}_{IO}^B - KP_{II}x_{IO}^B \\ \hline -P^T\ddot{x}_{IO}^B \end{array}\right\} \tag{33}$$

where

$$[MP_{II}] = \begin{bmatrix} T_{P1}^T M_{P1} T_{P1} & 0 & 0 \\ 0 & T_{P2}^T M_{P2} T_{P2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[KP_{II}] = \begin{bmatrix} T_{P1}^T K_{P1} T_{P1} & 0 & 0 \\ 0 & T_{P2}^T K_{P2} T_{P2} & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (34)$$

This set of Equations (33) could be solved resulting in a full-scale accurate solution. However, physically it is possible that the feedback vector $\{\ddot{x}_{IR}^{\beta}\}$ is small (note that we need not have $\{\dot{x}_{IR}^{\beta}\}$ and $\{x_{IR}^{\beta}\}$ small), i.e. $\{\ddot{x}_{IR}^{\beta}\} \simeq 0$ for all times t. Then from the third partition of Equation (33) we obtain a decoupled equation for $\{\bar{q}_N^P\}$,

$$\{\ddot{\bar{q}}_N^P\} + [2\bar{\zeta}_P \bar{\omega}_B]\{\dot{\bar{q}}_N^P\} + [\bar{\omega}_P^2]\{\bar{q}_N^P\} = -\{P^T x_{IO}^B\} \qquad (35)$$

ORIGINAL PAGE IS
OF POOR QUALITY

which can be easily solved, because $\{\ddot{x}_{IO}^{\beta}\}$ is known. This approach is called the direct base drive technique and has been used successfully.

The problem however is that the magnitude of $\{\ddot{x}_{IR}^{\beta}\}$ is not known in advance and may not always be small. In that case, Equation (33) should be solved retaining the coupling terms. Again, a Newmark-Chan-Beta technique can be used as follows,

$$\{\dot{q}\}_{i+1} = \{\dot{q}\}_i + (1-\gamma)h\{\ddot{q}\}_i + \gamma h\{\ddot{q}\}_{i+1} \qquad (36)$$

$$\{q\}_{i+1} = \{q\}_i + h\{\dot{q}\}_i + (0.5-\beta)h^2\{\ddot{q}\}_i + h^2\beta\{\ddot{q}\}_{i+1}$$

and

$$f_{Bi} = -D_4\,\dot{\bar{q}}_{NRi}^{B} - D_7\,\ddot{\bar{q}}_{NRi}^{B} - \bar{\omega}_B^2\,\bar{q}_{NRi}^{B}$$

$$f_{Ii} = -MP_{II}\,\ddot{x}_{IOi+1}^{B} - D_{II}\,\dot{x}_{IOi+1}^{B} - KP_{II}\,x_{IOi+1}^{B}$$

$$- D_5\,\dot{x}_{IRi}^{B} - D_8\,\ddot{x}_{IRi}^{B} - K_{II}\,x_{IRi}^{B} \qquad (37)$$

$$f_{Pi} = -P^T\,\ddot{x}_{IOi+1}^{B} - D_6\,\dot{\bar{q}}_{Ni}^{P} - D_9\,\ddot{\bar{q}}_{Ni}^{P} - \bar{\omega}_P^2\,\bar{q}_{Ni}^{P}$$

Also,

$$\ddot{x}_{IRi+1}^{B} = A_1(A_2 f_{Bi} + f_{Ii} + A_3 f_{Pi}) \qquad (38)$$

$$\ddot{\bar{q}}_{NRi+1}^{B} = D_1^{-1} f_{Bi} + A_2^T\,\ddot{x}_{IRi+1}^{B}$$

$$\ddot{\bar{q}}_{Ni+1}^{P} = D_3^{-1} f_{Pi} + A_3^T\,\ddot{x}_{IRi+1}^{B} \qquad (39)$$

where

ORIGINAL PAGE IS
OF POOR QUALITY

$$A_1 = [\, D_2 + A_2 B^T + A_3 P^T \,]^{-1}$$

$$A_2 = - B\, D_1^{-1} \quad , \quad A_3 = - P\, D_3^{-1} \tag{40}$$

and

$$D_1 = I + 2\gamma h \bar{\zeta}_B \bar{\omega}_B + \beta h^2 \bar{\omega}_B^2$$

$$D_2 = M_{II} + \gamma h D_{II} + \beta h^2 K_{II}$$

$$D_3 = I + 2\gamma h \bar{\zeta}_P \bar{\omega}_P + \beta h^2 \bar{\omega}_P^2$$

$$D_4 = 2\bar{\zeta}_B \bar{\omega}_B + h\bar{\omega}_B^2 \quad , \quad D_5 = D_{II} + hK_{II}$$

$$D_6 = 2\bar{\zeta}_P \bar{\omega}_P + h\bar{\omega}_P^2 \tag{41}$$

$$D_7 = 2(1-\gamma) h \bar{\zeta}_B \bar{\omega}_B + (0.5-\beta) h^2 \bar{\omega}_B^2$$

$$D_8 = (1-\gamma) h D_{II} + (0.5-\beta) h^2 K_{II}$$

$$D_9 = 2(1-\gamma) h \bar{\zeta}_P \bar{\omega}_P + (0.5-\beta) h^2 \bar{\omega}_P^2$$

At this point, it is possible to introduce a criterion which checks the magnitude of say $\{\ddot{x}^{\beta}_{IRi}\}$. If this magnitude is smaller than a certain preset ε then the quantities $A_2^T\{\ddot{x}^{B}_{IRi+1}\}$ and $A_3^T\{\ddot{x}^{B}_{IRi+1}\}$ in Equations (39) are not calculated. A possible criterion could be of the following form :

$$\|\ddot{x}^B_{IRi}\| \leq \varepsilon \|\ddot{x}^B_{IOi}\| \tag{42}$$

where ε is a preset percentage (e.g. 0.01).

This criterion could partially avoid the premultiplications by A_2^T and A_3^T in Equation (39). There are two more cost generating premultiplications by A_2 and A_3 in Equation (38). These could possibly also be avoided when the feedback acceleration is small. This would mean a direct base drive at that particular time step. Both these approaches were implemented and will be discussed in the next chapter.

The main problem with this kind of approaches is to find an answer to the question: What constitutes a small feedback? This question is still not answered even with an equation like Equation (42). Even though encouraging results were obtained, it is recognized that additional research and development is necessary.

CHAPTER II: The Software - Description and User Guide

1. INTRODUCTION

In this chapter we shall discuss the software package associated with a complete booster/payload response and loads analysis. An attempt will be made to clearly link the theory of Chapter I with the specific program and subroutine descriptions. This will give us the opportunity to touch upon some of the constraints and difficulties invariably associated with the development of a practical payload integration software package. Some factors to consider are: computer core usage; convergence; available data; the separation of booster, payload and integration organizations; work schedules; engineering time; ease of program usage; computer cost and related efficiency of algorithms; reuse of existing information; required accuracy versus cost; handling of potentially large models; etc.

In Section 2 of this chapter we shall present a general description of the organization and components of the software package. In particular, we shall explain the purpose and contents of the components and how they relate to each other. Then, in Section 3 a more detailed description of each of the components will be presented. This will include flow diagrams, description of input and output, how the programs are composed and how they must be used.

Section 4 describes two examples of booster/payload(s) systems. The first is a sample problem which was used to check out the subroutines. This will give us an opportunity to show how the six programs must be used. The other example is the more realistic case of the STS-ST-OMS-Kit Structure. We will also discuss the results of the short-cut approach.

Section 5 represents a selected reference list. Finally, Section 6 contains listings and samples of input and output for the several programs and subroutines.

2. GENERAL DESCRIPTION OF SOFTWARE PACKAGE

In this section we shall discuss in general terms the composition of the payload integration package and relate the several components to the theory in Chapter I.

Due to computer storage limitations, all programs in the software package were written using FORMA partition-logic subroutines. The partition-logic subroutines eliminate large storage requirements by using a random access (direct access) mass storage device which only brings a section of a large matrix into storage at one time. However, this process involves much bookkeeping and reading and writing of intermediate answers to the random access file. These processes make a partition-logic subroutine much slower and costlier than a dense-logic subroutine that does the same function. Thus, the use of partition-logic subroutines involves a trade-off between storage requirements and computation time. In order to make the payload integration programs as efficient as possible with regard to these two parameters it was decided to combine the use of dense and partition-logic. Thus, for our purposes diagonal matrices and column vectors are stored as dense-logic column vectors and all other matrices are stored in partition-logic. Also, to take advantage of this storage method several hybrid dense-logic/partition-logic subroutines were developed. The functions of these subroutines are described in their respective subroutine listings in Section 6 of this chapter.

The software package consists of six major programs and the supporting subroutines. The six programs are: BOOSTER, PAYLOAD, INTFACE, FORCE, RESPONS, LOADS.

These programs are designed to be as independent as possible. Thus, the program BOOSTER contains parameters that only involve the booster/launch vehicle and requires no prior knowledge of the payload configuration. Likewise, program PAYLOAD only requires knowledge of payload parameters. This way the booster and payload organizations can work completely independent of each other. Program INTFACE does the actual model coupling between booster and payloads. However, these results are independent of time and thus, program INTFACE need only be run once for each payload/booster configuration. The remaining programs are time dependent and must be run separately for each load case and test configuration.

The BOOSTER program essentially calculates all the quantities related to the booster only, i.e.,

$$T_B^T M_B I_B \Phi_N^B, \quad T_B, \quad T_B^T M_B T_B, \quad \text{and} \quad T_B^T K_B T_B$$

These quantities are needed in the solution of Equation (10). A flow diagram of the main program BOOSTER can be seen in Figure 2 Subroutine ZBOOST performs the actual computations of the above quantities as we shall discuss in the next section.

ORIGINAL PAGE IS
OF POOR QUALITY

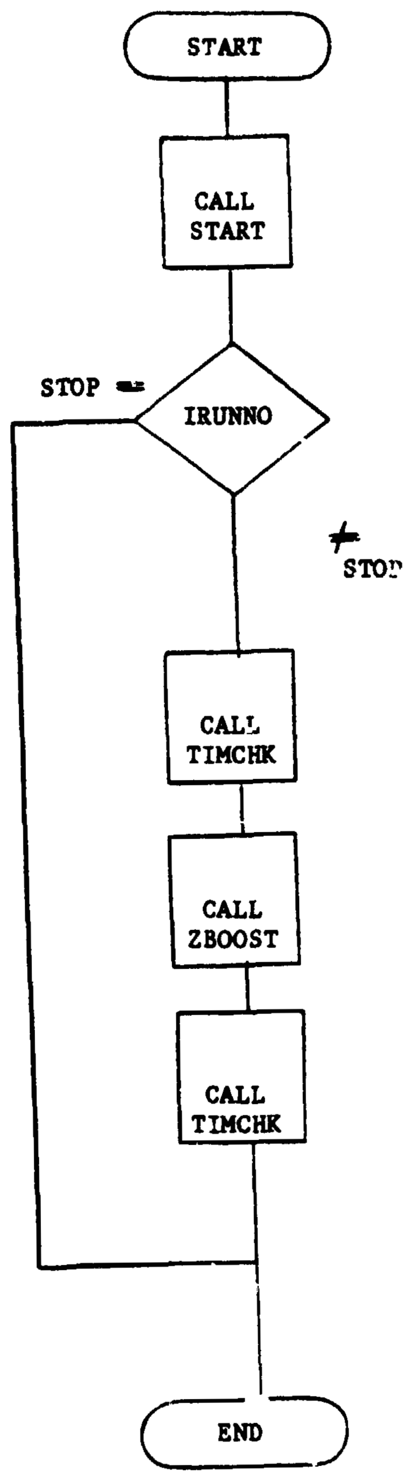


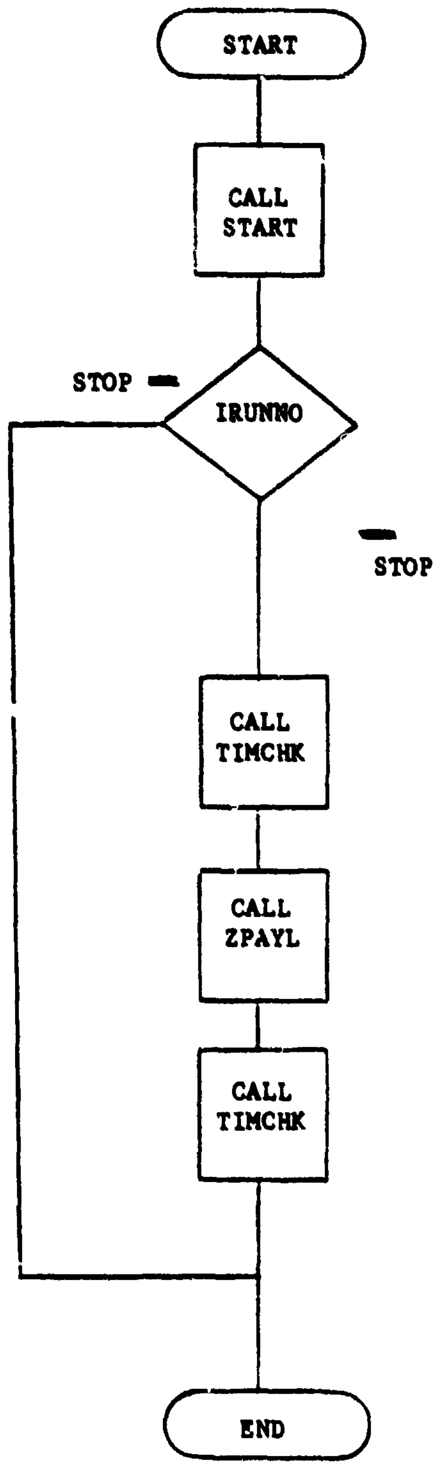
FIGURE 2: FLOW DIAGRAM FOR PROGRAM BOOSTER

Similarly, program PAYLOAD computes all payload quantities needed for the solution of Equation (10). In addition it calculates also quantities needed for the loads calculations in Equations (19-21), i.e.,

$$T_{Pj}^{T} M_{Pj} I_{Pj} \bar{\Phi}_{N}^{Pj}, \quad T_{Pj}, \quad T_{Pj}^{T} M_{Pj} T_{Pj}$$

$$T_{Pj}^{T} K_{Pj} T_{Pj} \quad , \quad E_{Pj} \quad , \quad [\beta]_{j}, \text{ and } [\delta]_{j}$$

Figure 3 represents a flow diagram of program PAYLOAD. Again, the actual calculation of the above quantities is done in subroutine ZPAYL to be discussed in next section. Note that subroutine payload is run separately for each payload.



ORIGINAL PAGE IS OF POOR QUALITY

FIGURE 3: FLOW DIAGRAM FOR PROGRAM PAYLOAD

Program INTFACE essentially combines booster and payload quantities to produce the necessary interface quantities and to synthesize the total payload quantities. Referring to Equation (8) in Chapter I, the interface mass and stiffness matrices, M_{II} and K_{II}, are formed and the interface eigenvalue problem for these quantities are solved resulting in the interface modes, Φ_I^B, and frequencies, ω_I^B. Program INTFACE also collects similar quantities for the several payloads into an overall payload quantity. Referring to Equation (10) these quantities are P, and $\bar{\omega}_P^2$. The actual calculations are done in subroutine ZINTF. Program INTFACE requires input from both the booster organization and all payload organizations. This is where the different organizations will have to interact with each other and therefore a great amount of coordination is often required. A flow diagram of program INTFACE is presented in Figure 4. Note that programs BOOSTER, PAYLOAD and INTFACE are one-time programs in the sense that the output of these programs is not dependent on time. Therefore, they represent a one-time cost in the course of a load cycle. The purpose of these three programs together with program FORCE is to produce the necessary INPUT quantities for programs RESPONSE and LOADS.

Program FORCE generates the necessary force time histories $\{F_B\}$ on the righthand side of Equation (7). It uses a linear interpolation scheme to obtain the correct force amplitudes at the right integration times used in program RESPONS. Note that the form of the force term in Equation (7) involves only the force vectors and booster properties, therefore, it need only be run once for any given booster and force/time history. A flow chart of program FORCE can be seen in Figure 5. Also, the actual calculations are performed by subroutine ZFORCE.

ORIGINAL PAGE IS
OF POOR QUALITY

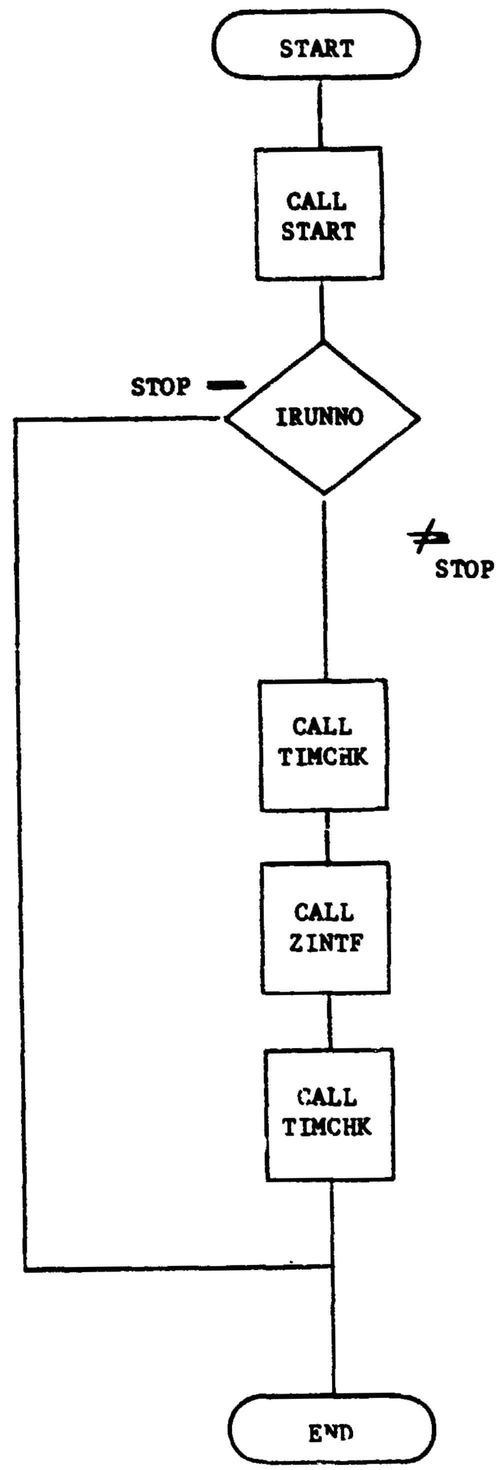


FIGURE 4: FLOW DIAGRAM FOR PROGRAM INTFACE

ORIGINAL PAGE IS
OF POOR QUALITY

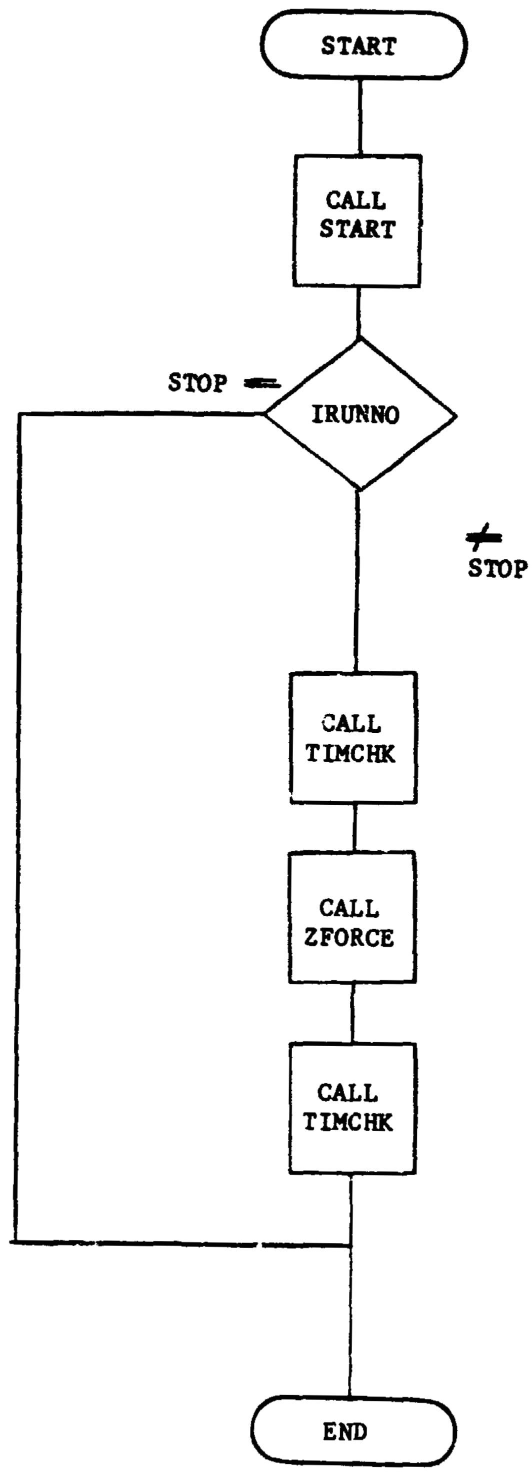


FIGURE 5: FLOW DIAGRAM FOR PROGRAM FORCE

Program RESPONS generates the response of the coupled booster/payload(s) system using the direct integration technique described by Equations (11-17) in Chapter I. A flow diagram of this program is presented in Figure 6. Subroutine ZRESP computes the actual transient response history.

Finally, program LOADS accepts the system response from program RESPONSE and generates the payload loads and other additional quantities if necessary (stresses, strains, maximum and minimum loads, etc.). The LOADS program uses the "acceleration" approach as opposed to the so called "displacement" approach. A flow diagram of program LOADS is shown in Figure 7. ZLOADS is the subroutine which produces the desired quantities.

The six programs as described above form a complete booster/payload(s) integration software package which allows for a reasonable amount of flexibility. An attempt was made to generate a reasonably general and flexible software package which is still relatively simple to use. Again, it should be noted that all programs are written in partition-logic and make use of both the Partition and Dense Logic Subroutine Libraries as developed by Martin Marietta Aerospace. Some of these subroutines were adapted to this effort and will be listed as such. Also, a few new basic subroutines were developed and will be discussed later.

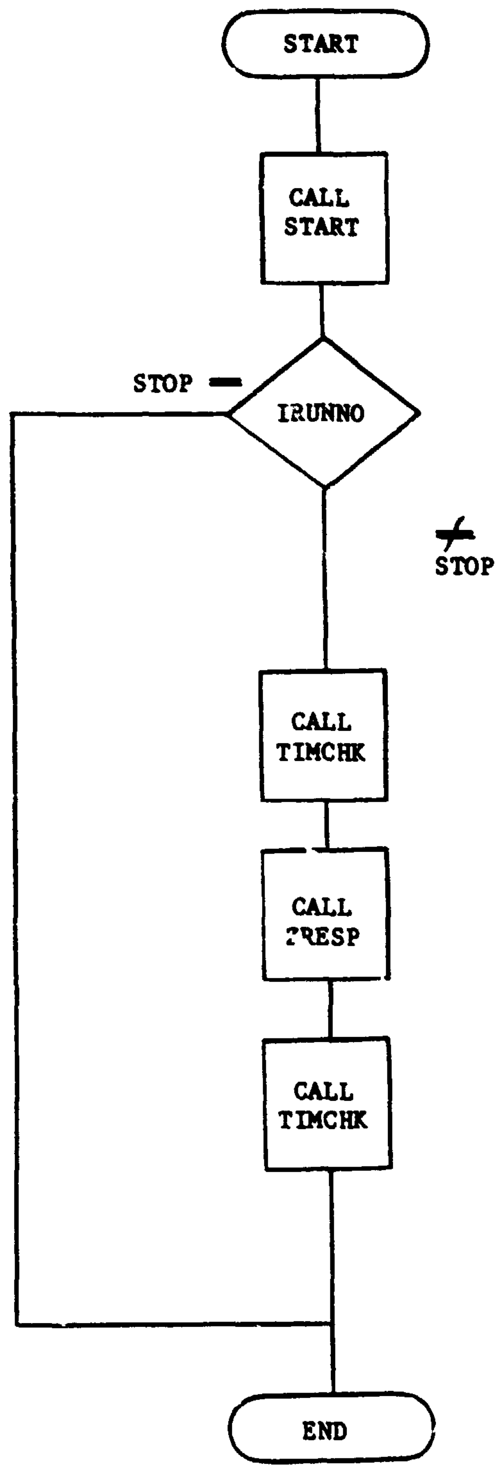


FIGURE 4: FLOW DIAGRAM FOR PROGRAM RESPONS

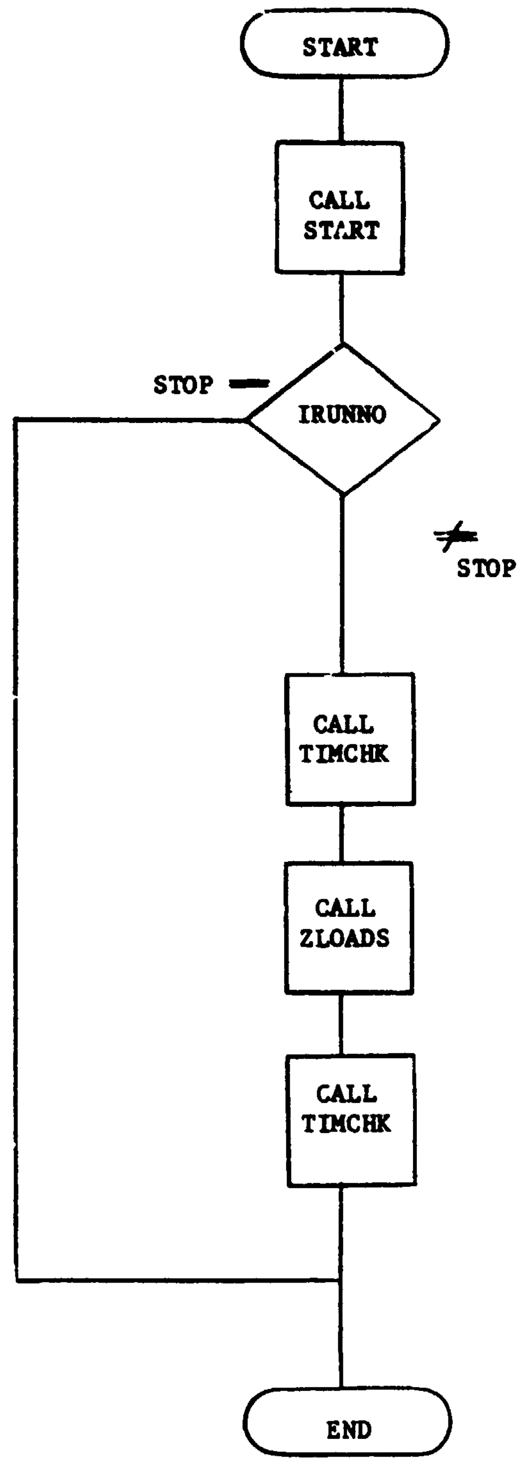


FIGURE 5: FLOW DIAGRAM FOR PROGRAM LOADS

3. DETAILED DESCRIPTION OF SOFTWARE PACKAGE.

In this section we shall present the six programs which together with the FORMA subroutine library constitute a complete booster/payload integration software package. In particular, we shall discuss the contents and usage of each of the components and point out the relationships with the theory as described in Chapter I.

a. Program BOOSTER

As mentioned in section 2, program BOOSTER generates all booster data necessary to run program INTFACE and eventually program RESPONS. Program BOOSTER starts out by calling subroutine START which performs a number of monitoring functions. For example, subroutine START reads the first three input cards of the input deck; interrogates the computer for the time of day, central processor time; defines a few common blocks; stops the program when IRUNNO = STOP, etc. Note that every subroutine in the FORMA library contains a number of comment cards explaining in detail the purpose and functions of the subroutine. Therefore, for a detailed description of subroutine START and any subsequent FORMA subroutine used in this text we refer the reader to the FORMA subroutine manuals (Volume 3A-LISTINGS, DENSE FORMA SUBROUTINES). Next, program BOOSTER calls subroutine TIMCHK (see Volume 3A) which determines the elapsed CP and PP time between defined points in a program. Finally, program BOOSTER calls the new subroutine ZBOOST which performs the actual computations necessary to generate all booster data. Figure 8 represents a flow chart of subroutine ZBOOST.

Subroutine ZBOOST was written in part ion-logic and allows for a reasonable degree of flexibility. The final objective is to produce the following quantities which are necessary for later calculations:

PHINBR = the expanded truncated set of cantilevered booster modes where rows corresponding to zero applied forces are deleted.

FREQBT = truncated cantilevered booster frequencies ($= \bar{f}_B$)

TBR = the constraint modal matrix with interface dofs. included and where columns corresponding to zero applied forces are deleted. ($= T_B$)

BM2 = booster mass matrix reduced to the interface

$$(= T_B^T M_B T_B)$$

BK2 = booster stiffness matrix reduced to the interface

$$(= T_B^T K_B T_B)$$

BM1 = booster coupling mass matrix between interface and non-interface dofs.

$$(= T_B^T M_B I_B \bar{\Phi}_N^B)$$

The first step of subroutine ZBOOST as shown in fig. 8 is to read a set of input parameters describing what information is available to the subroutine. These input parameters are as follows:

IF = Number of interface dofs.

NB = Number of truncated booster modes and frequencies that are to be retained.

ND = Number of discrete booster dofs, (i.e. the size of the discrete booster mass matrix, M_B).

NF = Number of non-zero force components in F_B.

NWFILE = Logical file number to which the output is written (for example NWFILE = 11).

NWRKFL = Logical file number of a partition-logic work file (for example NWRKFL = 1).

NWRITE = Flag parameter. If NWRITE = 0, the results are not printed on paper. If NWRITE = 1, the results are printed on paper.

NEXP = Flag parameter. If NEXP = 0, the cantilevered booster modes are not expanded to include interface dofs. If NEXP = 1, the modes are expanded. (i.e. $I_B \bar{\Phi}_N^B$ is available)

NTB = Flag parameter. If NTB = 0, the booster interface reduction transformation T_B is not available, and must be calculated. If NTB = 1, T_B is available.

NBMK12 = Flag parameter. If NBMK12 = 0 the quantities

$$BM1 = T_B^T M_B I_B \bar{\Phi}_N^B$$

$$BM2 = T_B^T M_B T_B$$

$$\text{and, } BK2 = T_B^T K_B T_B$$

are not available and if NBMK12 = 1, all these quantities are available.

NDET = Flag parameter. NDET = 0 then the interface is indeterminate, and if NDET = 1, the interface is determinate (i.e. BK2 = 0).

The next function of subroutine ZBOOST is to generate the expanded truncated cantilevered booster modes, PHINBR. The user must input the cantilevered modes and the flag, NEXP tells whether or not the cantilevered modes have been expanded to include the interface degrees of freedom (dofs). If the modes are not expanded, an integer vector, IFACE, is defined which locates the interface dofs. in the booster, which is then used to execute the expansion. Many times more modes are calculated than are necessary for the accurate determination of the response. Therefore, the user must input the flag, NB, informing the program how many modes should be retained. The purpose of this feature is to truncate higher frequencies above the so-called cut-off frequency, thereby, reducing the size of the equations to be solved without loss of much accuracy in the response and loads calculations. Subroutine ZBOOST contains a mechanism to truncate both the modes and frequencies according to the desired cut-off frequency. Next, in order to reduce the size of later

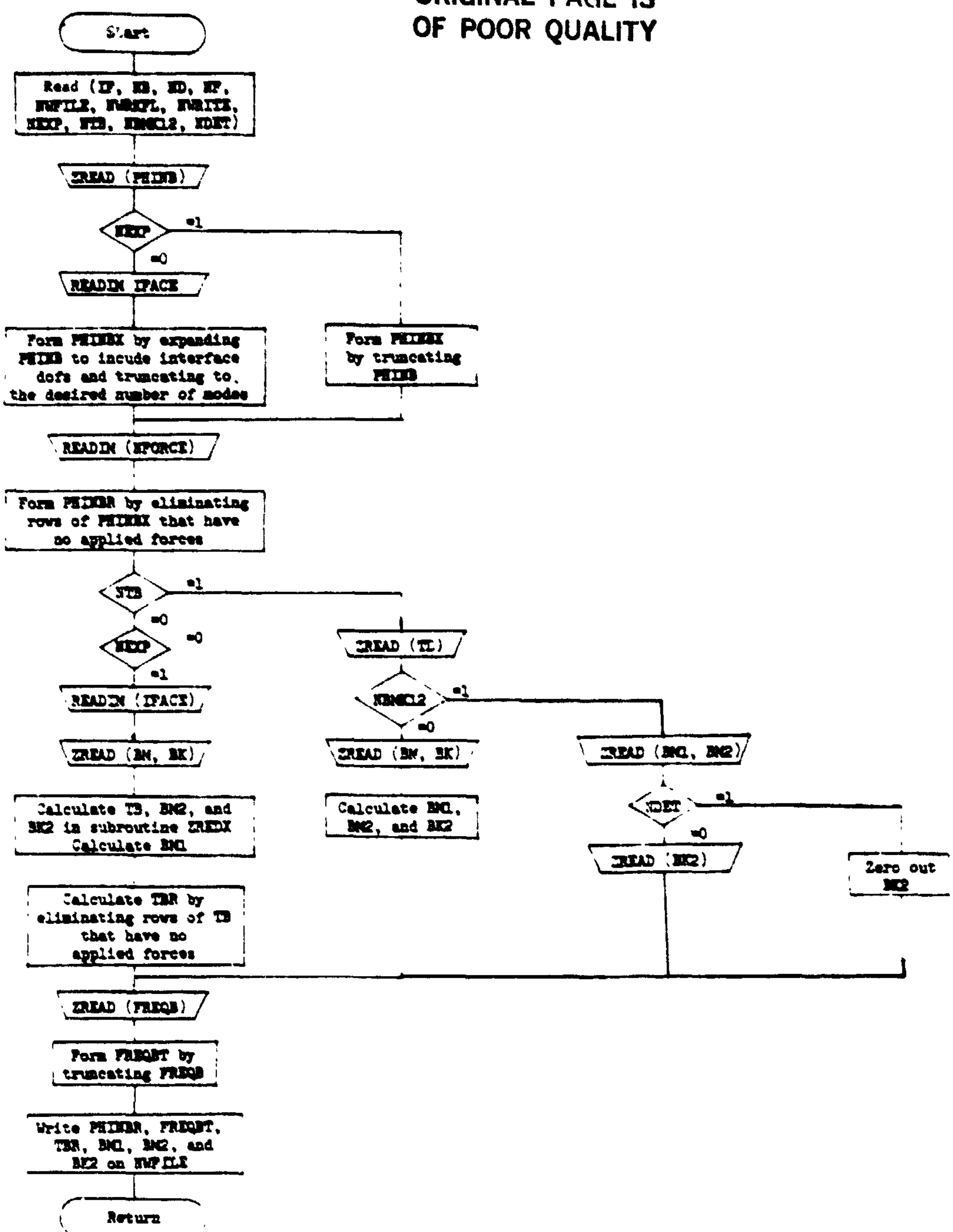


Figure 8. Flow chart of subroutine ZBOOST

multiplications in the forcing terms of Eq. (10), rows of PHINB that correspond to non-forced degrees of freedom are eliminated. In order to accomplish this function, an integer vector, NFORCE, describing the locations of forced degrees of freedom is input, and a reduction is performed resulting in the final quantity PHINBR.

Subroutine ZBOOST then computes TBR, BM1, BM2 and BK2 as needed. The flag, NTB, tells whether or not T_B is given. If T_B is not given (NTB = 0) then it is assumed that BM1, BM2 and BK2 are also not given, and we must calculate all four of these quantities. To calculate these quantities we need the free mass and stiffness of the booster and the locations of the interface dof locations. These quantities are BM, BK and the IFACE vector which may have been read in when forming PHINBR. Subroutine ZBOOST calculates T_B, then perf... the correct triple products to form BM1, BM2 and BK2.

If T_B is given (NTB = 1), then subroutine ZBOOST checks the flag NBMK12. When NBMK12 = 1, the quantities BM1, BM2 and BK2 are assumed to be given and are therefore read into the program. If NBMK12 = 0, then BM and BK are input and these quantities are formed by their respective triple products.

Once again we can note from Eq. (10) that we can eliminate many needless multiplications in the force term if we delete the rows of T_B which correspond to dofs with no forces applied. Therefore, we use the vector NFORCE to form TBR which contains only the forced rows of T_B.

Finally, FREQB, a vector of the cantilevered booster frequencies is input and truncated to the NB frequencies below the cut-off frequency. The new vector is FREQBT. Subroutine ZBOOST is then ended by writing the quantities, PHINBR, FREQBT, TBR, BM1, BM2 and BK2 out on the tape, NWFILE, and on request (i.e. NWRITE = 1) writes them on paper.

It should be pointed out that subroutine ZBOOST has maximum capabilities of 200 interface dofs and 700 booster dofs. These limits may be changed by increasing the corresponding values in the DATA and DIMENSION statements of the subroutine.

It could be argued that the vector $2\bar{\zeta}_{i}\bar{\omega}_{i}$ could also be calculated in subroutine ZBOOST because this quantity depends on the booster only. However, this is done in program RESPONS because $\bar{\zeta}_B$ could be subject to trial and error. This way we avoid a rerun of program BOOSTER for each case. Similarly, we calculate the forcing function terms $\bar{\Phi}_N^{BT} I_B^T F_B$ and $T_B^T F_B$ in a separate program because there could be several different force profiles.

Subroutine ZBOOST does not have any subroutine arguments. Therefore, all information given to the program is given in the data deck. The specific structure and format is given in the documentation to program BOOSTER and must be followed exactly as given.

b. Program PAYLOAD

Program PAYLOAD generates all payload data necessary to run program INTFACE, program RESPONS and program LOADS. Program PAYLOAD is largely identical to program BOOSTER except for the calculation of quantities related to the load transformations. Subroutine ZPAYL generates the following DATA:

FREQPT = Truncated cantilevered payload frequency vector. ($= \bar{f}_P$)

TP = The payload constraint modal matrix ($= T_P$)

PK2 = The payload stiffness matrix reduced to the interface

$$(= T_P^T K_P T_P)$$

PM2 = The payload mass matrix reduced to the interface

$$(= T_P^T M_P T_P)$$

PM1 = The payload coupling mass matrix between interface and non-interface degrees of freedom

$$(= T_P^T M_P \quad I_P \bar{\Phi}_N^P)$$

PL1 = A load transformation $(= I_P E_P I_P^T M_P I_P \bar{\Phi}_N^P)$

PL2 = A load transformation $(= I_P E_P I_P^T M_P T_P \quad)$

Note that we assume that no external forces are applied at the payload degrees of freedom. Therefore, PHINPR and TPR need not be calculated. However, because we are eventually interested in payload loads, we do need to generate the load transformations PL1 and PL2, in addition to T_P, according the Eqs. (18-21). Looking at Figure 9 we note that the quantities FREQPT, TP, PK1, PM2 and PM1 are calculated in the same way as are their counterparts in subroutine ZBOOST. The calculation of PL1 and PL2 does not present any unusual problem. Subroutines such as ZZERO, ZSLADR, ZTRANS, ZINV3 and ZMULT (Volume 3B) are used to accomplish this task. We also introduced two additional FLAG PARAMETERS: NEP (= 0, EP is not available, = 1, EP is available); and NLOAD (0 =, the load transformations PL1 and PL2 are not available, = 1, these transformations are available).

Again, subroutine ZPAYL does not require any subroutine arguments. The documentation in program PAYLOAD describes how the input quantities must be ordered in the input list.

ORIGINAL PAGE IS
OF POOR QUALITY

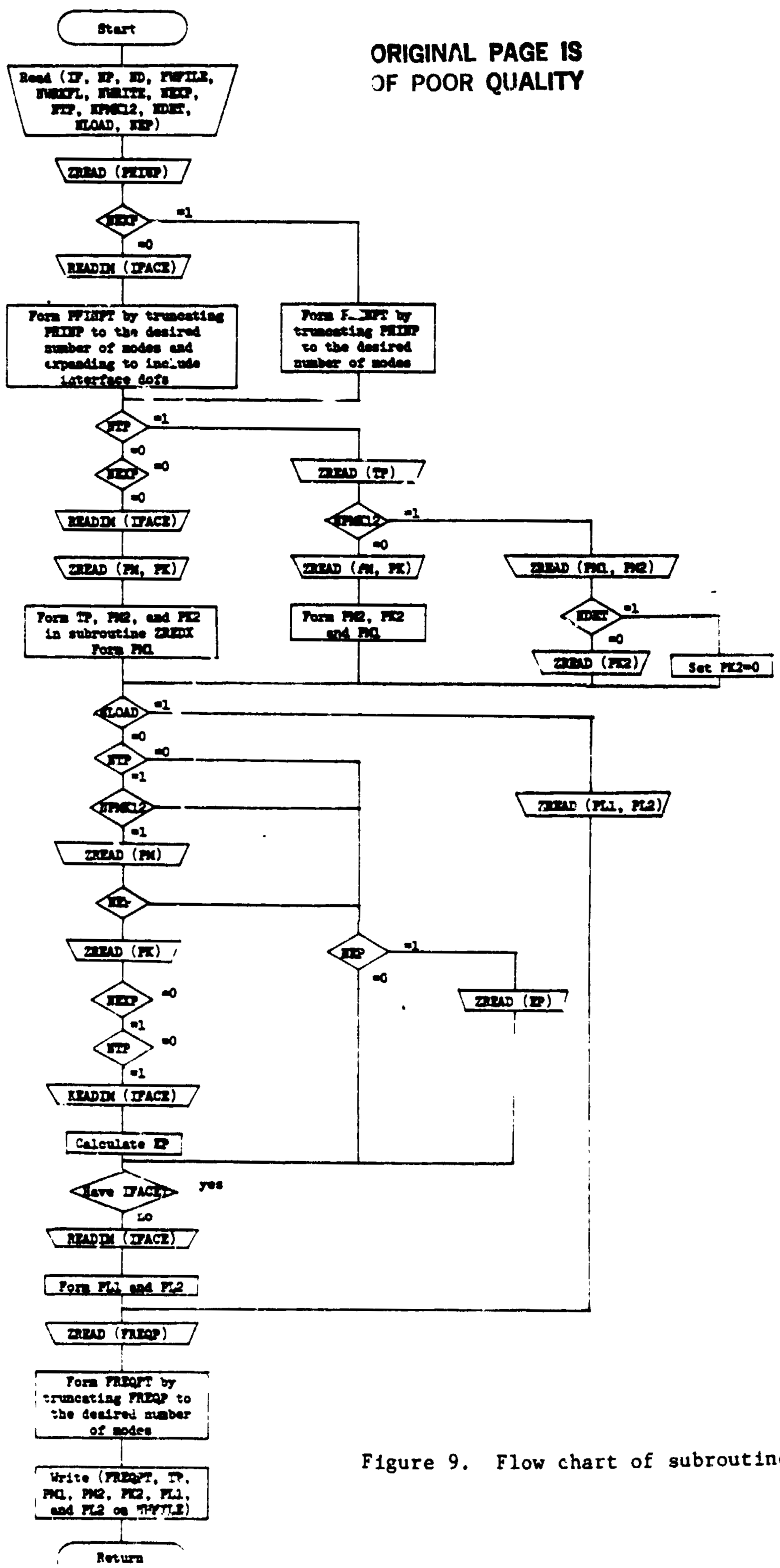


Figure 9. Flow chart of subroutine ZPAYL

c. Program INTFACE

Program INTFACE generates all interface quantities necessary to run program RESPONS. Also, it assembles quantities related to different payloads. Interface quantities involve both booster and payload data. In particular, program INTFACE calculates:

BPM2 = The sum of the interface reduced booster and payload mass matrices ($= M_{II}$)

BPK2 = The sum of the interface reduced booster and payload stiffness matrices ($= K_{II}$)

B2 = ${\phi_I^B}^T$ B , booster coupling mass matrix between interface and non-interface dofs. premultiplied by the transpose of the interface modes.

FREQPI = The interface frequencies ($= f_I$) These include the rigid body frequencies

FREQPA = The assembled payload frequency vector for all payloads ($= f_p = \{ f_{p1}, f_{p2}, \ldots, f_{pn} \}$)

PHIIB = The interface modes ($= \phi_I^B$)

P2 = ${\phi_I^B}^T$ P, the assembled payload coupling mass matrix as displayed in Eq. (10) premultiplied by the transpose of the interface modes.

Subroutine ZINTF starts by reading [illegible] of input parameters. These are as follows:

NPAY = Number of payloads to be coupled together

NWFILE = Logical file number to which the output is written (for example NWFILE = 11)

NWRKFL = Logical unit number of a required partition-logic work file (for example, NWRKFL = 10)

NWRITE = Flag parameter. If NWRITE = 0, the results will not be written on paper. If NWRITE = 1, the results will be written on paper.

IFB = Number of booster interface dofs. (including superfluous dofs.)

NPTOT = Total number of payload degrees of freedom

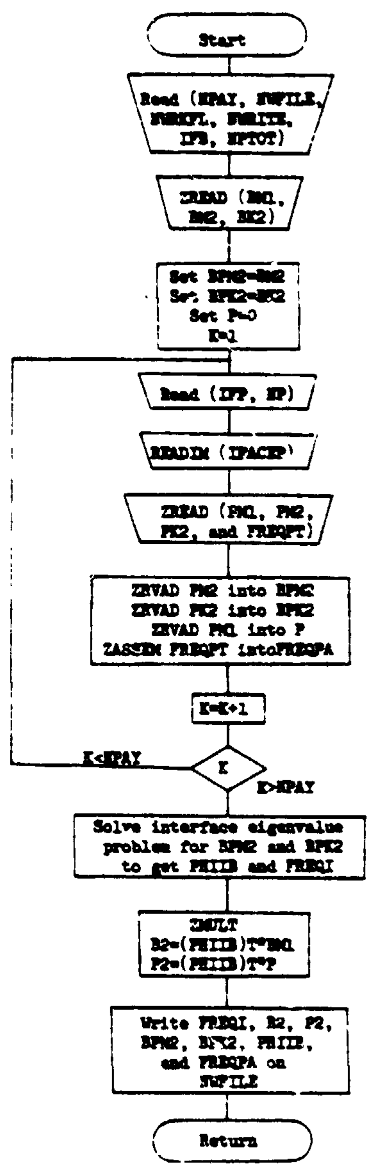


Figure 10. Flow chart of subroutine ZINTF

Next, the booster properties - BM1, BM2 and BK2 are input. BM2 and BK2 must be coupled with payload quantities to form the interface mass and stiffness. Therefore, we now need to read in the properties of each of the payloads. These properties are PM1, PM2, PK2 and FREQPT. We also need other descriptive information about the payloads so that they are handled correctly. Therefore, we must input parameters IFP, indicating how many dofs are in the payload interface, NP which tells how many payload modes were retained for this particular payload, and IFACEP which is a vector indicating which booster interface dofs correspond to which payload interface dofs.

Using IFACEP, PM2 is added to BM2 to form the interface mass matrix BPM2. Likewise, PK2 is added to BK2 to form the interface stiffness matrix BPK2. Also, PM1 is assembled into the total payload interface/non-interface coupling matrix, P, and a total payload frequency matrix, FREQPA, is assembled from FREQPT. This step is repeated for all payloads.

The interface eigenvalue problem is then solved and the interface modes PHIIB and frequencies, FREQI are formed. Note that we solve for all the modes. To get these properties in their final forms, P and BM1 are premultiplied by (PHIIB)T to get P2 and B2, respectively. Note that these properties are called P and B in Eq. (10).

To complete this program, all these quantities are written to the specified file, NWFILE, and on option, (NWRITE = 1) they are also written on paper.

Once again subroutine ZINTF has no subroutine arguments. All information given to the subroutine is in a data deck. The correct form of the data deck is given in the program documentation.

d. Program FORCE

Program FORCE generates the forcing function data in a form consistent with the requirements of program RESPONS. As stated earlier, it calculates the right-hand side of Eq. (7). Note tht these forcing terms contain only booster properties, so that this program need only be run once for each booster and force/time history. We have purposely waited until program RESPONS to premultiply the interface forcing term by $\phi_I^{B^T}$ as required by Eq. (10) for this reason. Subroutine ZFORCE does the actual calculation and generates a sequential file containing a header and the interpolated time-force data. A flow chart of subroutine ZFORCE is given in Figure 11.

Subroutine ZFORCE begins by reading the following set of input parameters:

DELTAT = The time step size used in the numerical integration scheme used to solve the response equation.

ORIGINAL PAGE IS
OF POOR QUALITY

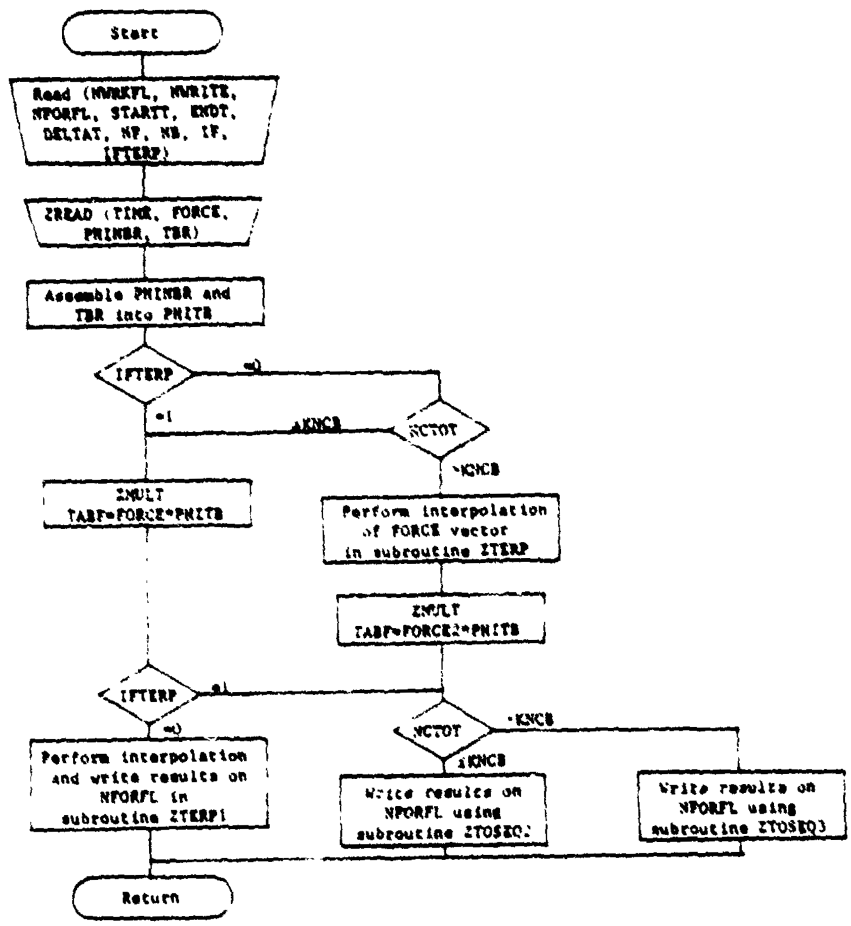


Figure 11. Flow chart of subroutine ZFORCE

ENDT = The end time for the interpolation time range

IF = The number of interface degrees of freedom

IFTERP = 0 The force data has not been previously interpolated
1 The force data has been interpolated

NB = The number of non-interface booster degrees of freedom

NF = The number of non-zero force components acting on the booster

NFORFL = Logical unit number of the output sequential file containing the time/force history

NWRITE = 0 Output time/force history should not be written on paper
= n Output is written on paper every n time points

NWRKFL = Logical unit number of a required partition-logic work file

STARTT = The beginning time for the interpolation time range

The only other input information needed are the partition-logic matrices TIME and FORCE which completely describe the response forcing function, and the matrices PHINBR and TBR which were output from program BOOSTER.

After all necessary data is known the subroutine ZFORCE goes about choosing the correct subroutines to generate the interpolated data on the sequential file NWFILE. To accomplish this goal with minimal time requirements, several special purpose subroutines were developed that combine partition and dense-logic.

If the input time/force data has not been previously processed, it is interpolated with either subroutine ZTERP or ZTERP1. ZTERP is a pure partition-logic linear interpolation routine. Output from this routine is in a matrix form that corresponds to the input. Subroutine ZTOSEQ3 then processes this matrix output data into the sequential file NWFILE. ZTOSEQ3 is also a pure partition-logic subroutine and therefore, has small storage space requirements and can handle large matrices.

Subroutine ZTERP1 is a special purpose subroutine which does the interpolation of the data and generates NWFILE at the same time. By combining these functions ZTERP1 runs approximately six times faster than the above combination. ZTERP1, however, is limited on the size of the matrices it can handle. This subroutine works by disassembling a row partition of the input matrix into a dense-logic matrix and then interpolates this section and writes the answers on NWFILE. The limits on size are determined by the dimensions of this dense-logic work matrix. Originally, this dimension was 600.

When interpolation of the force data is not needed either subroutine ZTOSEQ3 or ZTOSEQ2 converts the matrix data input into sequential output form. ZTOSEQ2 uses the same technique as ZTERP1 and the same matrix work space. It sections out a row partition of input matrix and converts it to a dense matrix and then generates NWFILE. Consequently, it suffers from the same size limitation as ZTERP1, but is again six times faster than ZTOSEQ3.

It should be pointed out that the use of these combinations of subroutines provides a balance between computation time and storage requirements. The hybrid routines run faster but require much more storage space. Therefore, the user might deem it worthwhile to change the limits on the size of the work space used by ZTOSEQ2 and ZTERP1 to fit his needs.

Subroutine ZFORCE does not have any subroutine arguments. All data is input through a data deck. The exact form of this data deck is given in the subroutine writeup.

e. Program RESPONS

Program RESPONS calculates the response of the coupled booster/payloads(s) system. The numerical technique used in this program is described by Eqs. (11-17) of Chapter I. These equations are programmed in subroutine ZRESP which does all of the calculations.

Subroutine ZRESP requires the following input:

NDAMPB and either DAMPB or ZETAB

NDAMPB = 0	indicates a constant value of booster modal damping. If NDAMP = 0 input ZETAB as this constant value
NDAMPB = 1	indicates a variable value of booster modal damping If NDAMP = 1 input DAMPB = $\lceil \bar{\zeta}_B \rfloor$ where DAMPB is a (1 x NB) row vector.

ORIGINAL PAGE IS
OF POOR QUALITY

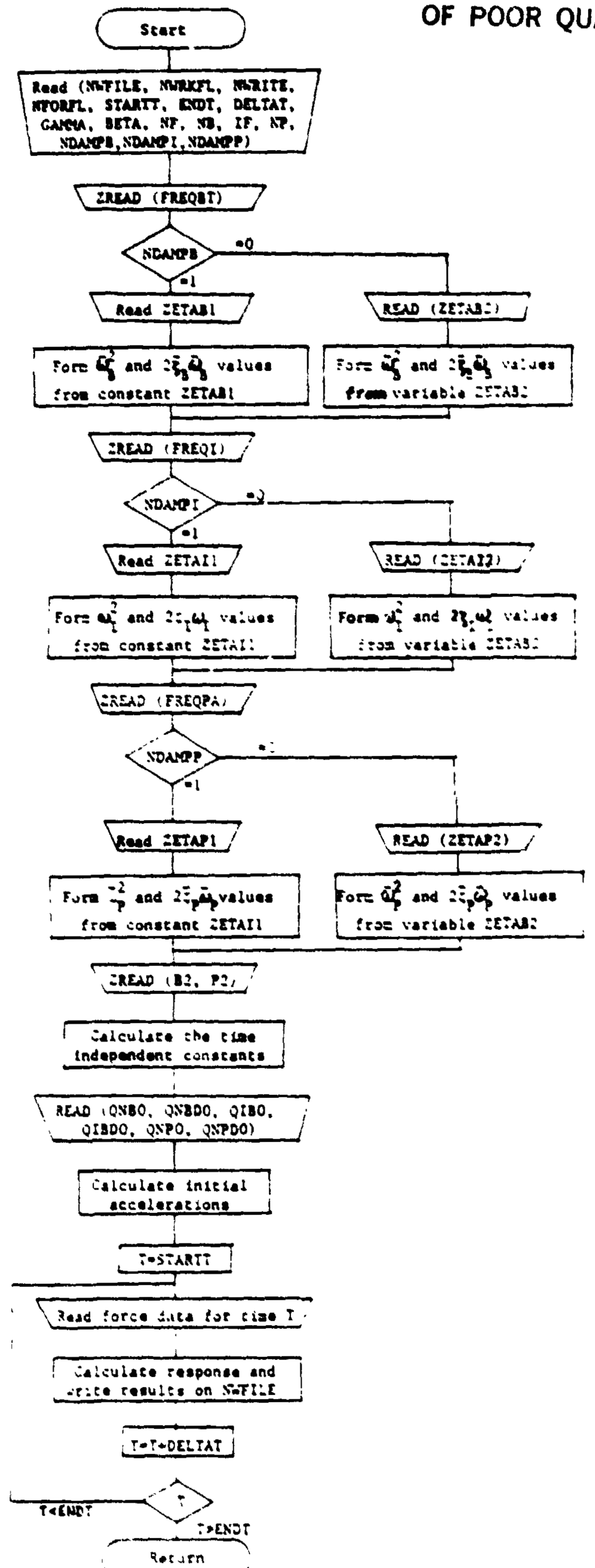


Figure 12. Flow chart of subroutine ZRESP

NDAMPI and either DAMPI or ZETAI
= The same parameters and variables as above for the interface modal damping.

NDAMPP and either DAMPP or ZETAP
= The same parameters and variables as above for the payload modal damping.

FREQBT = The assembled, truncated, cantilevered booster frequency vector output from ZBOOST

FREQI = The interface frequency vector output from ZINTF

FREQPT = The truncated, cantilevered payload frequency vector output from ZPAYL

B2 = The interface/booster mass coupling matrix, output from ZINTF

P2 = The interface/payload mass coupling matrix, output from ZINTF

PHIIB = The interface modes matrix output from ZINTF

QNBO, QNBDO, QIBO, QIBDO, QNPO , and QNPDO = the initial displacement and velocity patterns for the booster, interface and payload

STARTT, ENDT, DELTAT
= The start time, end time and time step for the numerical integration routine

GAMMA AND BETA
= Parameters in the Newmark-Chan-Beta numerical integration scheme. Good values are GAMMA = 0.5 and BETA = 0.25

IF = Number of interface degrees of freedom

NB = Number of non-interface booster degrees of freedom

NP = Number of non-interface payload degrees of freedom

NF = Number of forces acting on the booster

NFORFL = Logical file number containing the interpolated force data output from ZFORCE

NWFILE = Logical file number for output of the response data

NWRKFL = Logical file number of a required partition-logic work file

NWRITE = 0 when the results are not to be printed on paper
n the response data will be written on paper every n time steps

These input parameters and variables are then used to solve for the non-time dependent constants in Eqs. (13-15). Also, to totally describe the initial state of the system, we solve for the initial accelerations using Eq. (10) evaluated at the start time of the integration time loop. The response loop itself involves solving eqs. (12, 14 and 16) for each time step.

Once again subroutine ZRESP has no subroutine arguments. All input is done thru a data deck that is described in the documentation listed at the beginning of the subroutine.

f. Program LOADS

Program LOADS uses the response data and previously generated transformation matrices to calculate the internal loads. These loads can be calculated for an entire payload or any group of members upon request. This program must be run separately for each payload. All calculations are done in subroutine ZLOADS. A flow chart of ZLOADS is given in Figure 13.

Subroutine ZLOADS reads all input information from a data file, which is described in the subroutine documentation, it has no arguments. The subroutine begins by reading in a series of flag and input parameters. These parameters are:

IFB = The number of interface degrees of freedom in the booster

IFP = The number of interface degrees of freedom for the payload under study. (Note - this program is run separately for each payload, IFP is the number of degrees of freedom in this payload.)

ISELECT = 0 All rows of PKPSI are used in the load calculations

1 Only rows of PKPSI that are selected by IVSEL are used in the load calculations

MAXL = 0 Maximum/minimum loads calculation is not desired

1 Maximum/minimum loads calculation will be performed

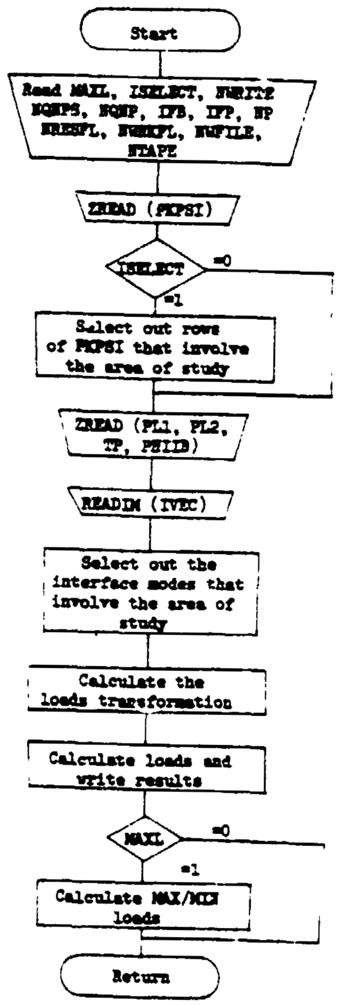


Figure 13. Flow chart of subroutine ZLOADS

NP = The total number of truncated, cantilevered modes for all payloads. The size of matrix (QNP)

NQNP = Number of non-interface dofs. in this payload

NQNPS = The position number in matrix (QNP) where the first degree of freedom for this payload occurs

NRESPFL = Logical unit number of the sequential file containing the response output from subroutine ZRESP

NTAPE = Logical unit number of a dense-logic forma file for the output of the maximum/minimum loads calculation results. If MAXL = 0 this file designation is ignored.

NWFILE = Logical unit number of a sequential file on which to write the loads calculation results.

NWRITE = 0 Loads results are not written on paper (if max/min loads are calculated, they are printed)

= n The loads results are written on paper every n time points

We then read in a load transformation matrix PKPSI which is given in Eq. (18). PKPSI is the entire free stiffness matrix of the payload (PKPSI $= K_p$) if internal node elastic forces is the desired output result. However, if individual member loads are desired, it will be the stiffness kernel for that member. The input flag ISELECT is consistent with this approach. If we are interested in studying a member of the body, then depending on the form of PKPSI we might desire to select out certain rows of PKPSI. An example of this feature is when the input form of PKPSI contains many zero rows or extraneous information. If we wish to use this selection process, we set ISELECT = 1 and input a vector IVSEL which when used in subroutine ZSLADR, picks out the desired rows.

Next a series of matrices that were output by subroutines ZPAYL and ZINTF are input. These matrices are PL1, PL2, TP and PHIIB. We must then select the rows of PHIIB which correspond to this payload. Therefore, an integer vector, IVEC, that designates which dofs of PHIIB are the interface dofs for the payload under study. These matrices are then used in the loads loop to calculate the loads history of the body or member. Also upon request, MAXL = 1, we calculate and output the maximum and minimum loads for the body.

4. SAMPLE PROBLEMS

In this section we shall discuss two sample problems. The first consists of a shell booster model that is to be integrated with two truss-like payloads. The second sample problem involves the more realistic case of the Space Transportation System (STS), Space Telescope (ST), and the OMS kit response and load analysis. We also included results on the short-cut version.

ORIGINAL PAGE IS
OF POOR QUALITY

Sample Problem 1

As stated previously, this sample problem consists of a booster and two payloads. FINEL, a Martin Marietta developed finite element code was used to generate these three models and their corresponding physical properties.

The booster, as shown in Figure 14 is a hexagonal cylinder consisting of 24 quadrilateral plate sections. The material properties of each of the plates are:

$\rho = 0.025$ lb/in^2 , $E = 10.6 \times 10^4$, $\nu = 0.334$ and thickness = 0.1 in

We joined the 24 elements at 30 nodes as described in Table 1. Each of these 30 nodes were assigned three translational degrees of freedom. The geometry and dof numbering system for the model are shown in Table 2. Twelve nodes (number 8, 9, 11, 12, 14, 15, 17, 18, 20, 21, 23 and 24) and all of their corresponding dofs were designated to make up the interface for possible coupling with payloads. Thus, for the booster we have 36 interface dofs and 54 non-interface dofs.

INPUT DATA FOR COMBINED MEMBRANE-BENDING QUADRILATERAL PLATE ELEMENTS

MASS = M1   STIF = K1
RO = .250E-01   E = .106E+06
T(MASS) = .100E+00   NU = .334E+00
T(MEMBRANE) = .100E+00
T(BENDING) = .100E+00

ELEMENT NUMBER	JOINT 1	JOINT 2	JOINT 3	JOINT 4
1	1	2	8	7
2	2	3	9	8
3	3	4	10	9
4	4	5	11	10
5	5	6	12	11
6	6	1	7	12
7	7	8	14	13
8	8	9	15	14
9	9	10	16	15
10	10	11	17	16
11	11	12	18	17
12	12	7	13	18
13	13	14	20	19
14	14	15	21	20
15	15	16	22	21
16	16	17	23	22
17	17	18	24	23
18	18	13	19	24
19	19	20	26	25
20	20	21	27	26
21	21	22	28	27
22	22	23	29	28
23	23	24	30	29
24	24	19	25	30

Table 1. Description of quadrilateral plate elements for the booster model used in sample problem 1

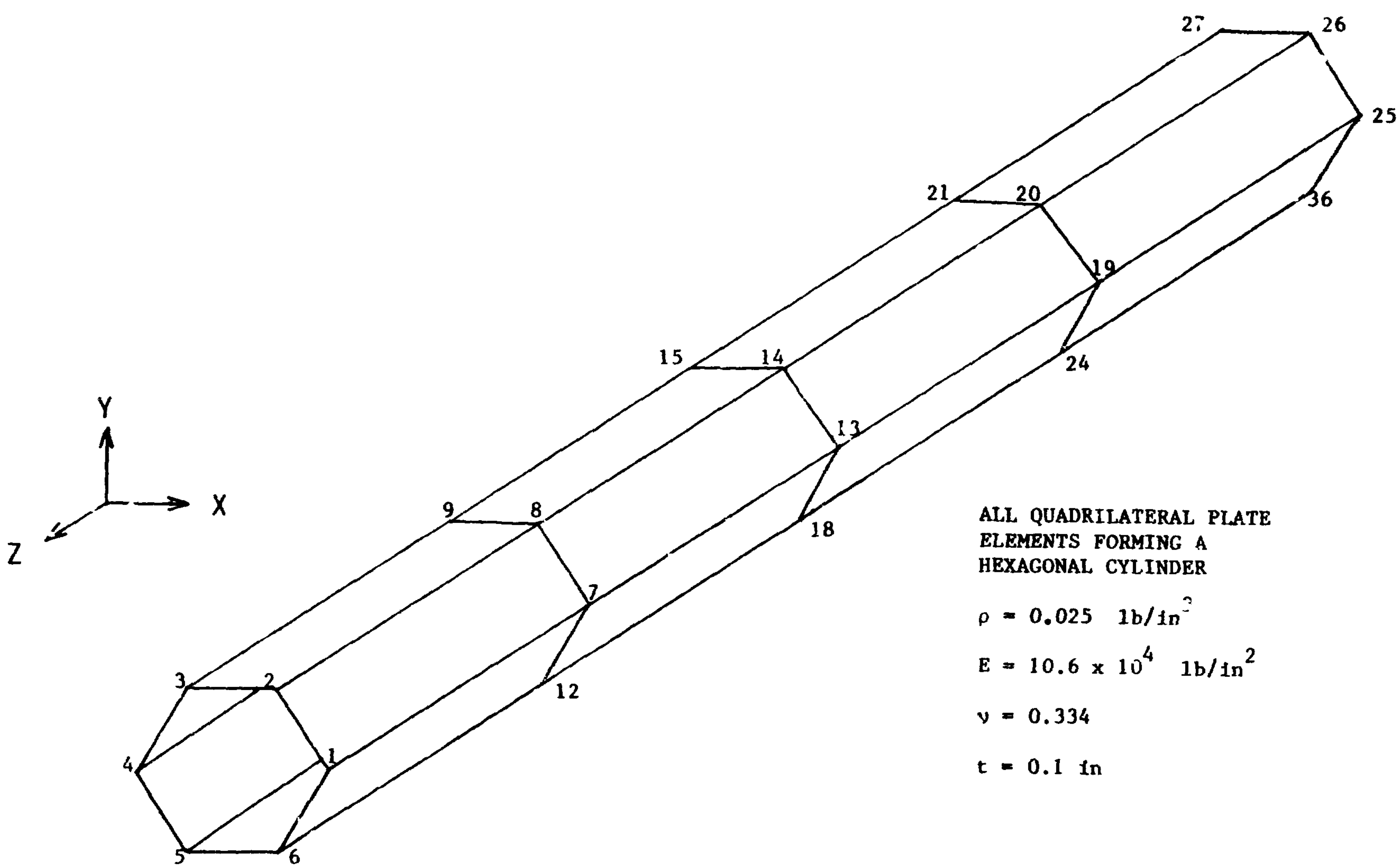


Figure 14. Booster model for Sample Problem 1

ORIGINAL PAGE IS
OF POOR QUALITY

JOINT DATA USED IN SUBROUTINE FEMKA

JOINT	DEGREES OF FREEDOM TRANSLATION U	V	W	ROTATION P	Q	R	GLOBAL CARTESIAN COORDINATES X	Y	Z
1	1	2	3	0	0	0	5.0000	0.0000	50.0000
2	4	5	6	0	0	0	2.5000	4.3300	50.0000
3	7	8	9	0	0	0	-2.5000	4.3300	50.0000
4	10	11	12	[illegible]	0	0	-5.0000	0.0000	50.0000
5	13	14	15	0	0	0	-2.5000	-4.3300	50.0000
6	16	17	18	0	0	0	2.5000	-4.3300	50.0000
7	19	20	21	0	0	0	5.0000	0.0000	25.0000
8	22	23	24	0	0	0	2.5000	4.3300	25.0000
9	25	26	27	0	0	0	-2.5000	4.3300	25.0000
10	28	29	30	0	0	0	-5.0000	0.0000	25.0000
11	31	32	33	0	0	0	-2.5000	-4.3300	25.0000
12	34	35	36	0	0	0	2.5000	-4.3300	25.0000
13	37	38	39	0	0	0	5.0000	0.0000	0.0000
14	40	41	42	0	0	0	2.5000	4.3300	0.0000
15	43	44	45	0	0	0	-2.5000	4.3300	0.0000
16	46	47	48	0	0	0	-5.0000	0.0000	0.0000
17	49	50	51	0	0	0	-2.5000	-4.3300	0.0000
18	52	53	54	0	0	0	2.5000	-4.3300	0.0000
19	55	56	57	0	0	0	5.0000	0.0000	-[illegible].0000
20	58	59	60	0	0	0	2.5000	4.3300	-25.0000
21	61	62	63	0	0	0	-2.5000	4.3300	-25.0000
22	64	65	66	0	0	0	-5.0000	0.0000	-25.0000
23	67	68	69	0	0	0	-2.5000	-4.3300	-25.0000
24	70	71	72	0	0	0	2.5000	-4.3300	-25.0000
25	73	74	75	0	0	0	5.0000	0.0000	-50.0000
26	76	77	78	0	0	0	2.5000	4.3300	-50.0000
27	79	80	81	0	0	0	-2.5000	4.3300	-50.0000
28	82	83	84	0	0	0	-5.0000	0.0000	-50.0000
29	85	86	87	0	0	0	-2.5000	-4.3300	-50.0000
30	88	89	90	0	0	0	2.5000	-4.3300	-50.0000

Table 2. Geometry description and degree of freedom table for the booster model used in sample problem 1

In order to run program BOOSTER for this model we generated four pieces of information about the body. First, the free-free mass and stiffness matrices were generated, then the interface dofs were constrained to generate the interface cantilevered modes and frequencies. Also, it was determined to force the model in the Z-direction at nodes 26, 27, 29 and 30 (dofs 78, 81, 87 and 90).

Using the above information program ZBOOST was run in order to generate all the booster quantities necessary to run the other programs in the software package. Figure 15 shows the input deck to program BOOSTER. The first three lines are input required by subroutine START. These determine the RUNNO and a descriptive title. All other lines are required by subroutine ZBOOST. For this model we have 36 interface dofs (IF = 36), we have retained all of the booster cantilevered modes and frequencies (NB = 54), we have a total of 90 dofs (ND = 90) and four force points (NF = 4). Since the only information we have about the booster is the free mass and stiffness, the non-expanded cantilevered modes and the cantilevered frequencies; NEXØ = 0, NTB = 0, NBMK12 = 0. Also, since there are more interface dofs than rigid body dofs, the interface is not determinate (NDET = 0). The results of subroutine ZBOOST's interpretation of this file is given in the sample output (Figure 16). A listing of the matrices generated by this subroutine and their sizes are given in Table 3.

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	ZBOOST	PHINB	90	54	10SE82	100%
2	ZBOOST	TB	90	36	10SE82	100%
3	ZBOOST	PHINBR	4	54	10SE82	100%
4	ZBOOST	FREQBT	1	54	10SE82	100%
5	ZBOOST	TBR	4	36	10SE82	100%
6	ZBOOST	BM1	36	54	10SE82	100%
7	ZBOOST	BM2	36	36	10SE82	100%
8	ZBOOST	BK2	36	36	10SE82	100%

Table 3. Listing of the matrices output by subroutine ZBOOST for sample problem 1

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZBOOST      TG SHANAHAN
PROGRAM ZBOOST   FOR BOOSTER 1 SAMPLE PROBLEM
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
   36   54   90    4                     * IF,NB,ND,NF *
   40   20    0                          * NWFILE,NWRKFL,NWRITE *
    0    0    0    0                     * NEXP,NTB,NBMK12,NDET *
MODES    0  -31BOOST          * NON-EXPANDED CANTILEVERED BOOSTER MODES *
IFACE    1   36               * VECTOR OF INTERFACE DOFS IN BOOSTER *
    1    1   22   23   24   25   26   27
    1    7   31   32   33   34   35   36
    1   13   40   41   42   43   44   45
    1   19   49   50   51   52   53   54
    1   25   58   59   60   61   62   63
    1   31   67   68   69   70   71   72
0000000000
NFORCE   1    4               * VECTOR OF FORCED DOFS *
    1    1   78   81   87   90
0000000000
MASS     0  -30UNDBOO         * FREE BOOSTER MASS *
STIF     0  -30UNDBOO         * FREE BOOSTER STIFFNESS *
FREQ     0  -31BOOST          * CANTILEVERED BOOSTER FREQUENCIES *
STOP
```

Figure 15. Input deck to program BOOSTER

```
                    INPUT VARIABLES TO ZBOOST
                    ------------------- --- -

                         NWFILE =   40
                         NWRKFL =   20
                         NWRITE =    0

                     DESCRIPTION OF BOOSTER
                     ----------------------

               NUMBER OF BOOSTER DOFS =   90          ND =   90
     NUMBER OF BOOSTER INTERFACE DOFS =   36          IF =   36
NUMBER OF TRUNCATED BOOSTER MODES RETAINED =   54     NB =   54
  NUMBER OF FORCES APPLIED TO THE BOOSTER =    4      NF =    4


THE MODES ARE NOT EXPANDED TO INCLUDE INTERFACE DOFS      (NEXP = 0)

           TB, BM1, BM2, AND BK2 ARE NOT GIVEN            (NTB = 0)

              THE INTERFACE IS NOT DETERMINANT            (NDET = 0)
```

Figure 16. Sample output from subroutine ZBOOST

Figure 17 shows the first payload (payload $P1$) which is made up of 18 bar elements. The bars are joined at 8 nodes to form this truss. The geometry of the structure is given in Table 4 and the material properties are given in Table 5. Once again all nodes were assigned translational degrees of freedom only. The four corner nodes and all of their corresponding degrees of freedom make up the interface. Thus we have the following for payload 1.

ND = 12, IF = 12,

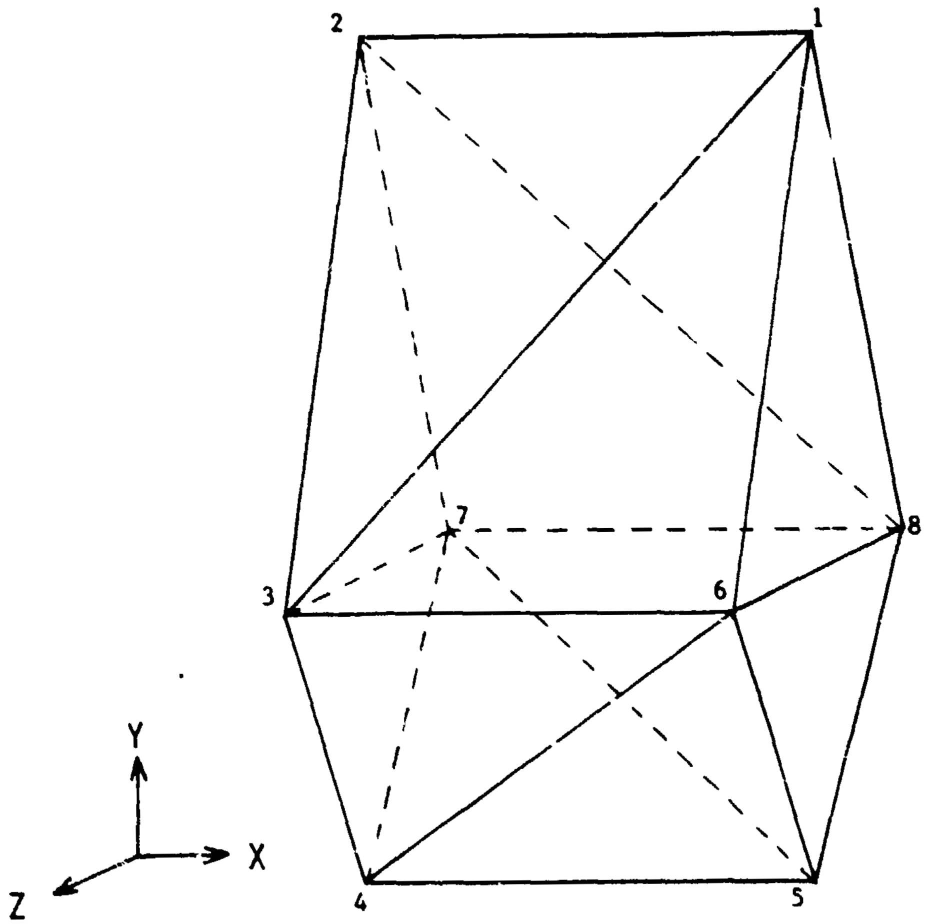
and, IFACE = dofs 1, 2, 3, 4, 5, 6, 10, 11, 12, 13, 14 and 15

If order to run program PAYLOAD for this structure we generated the free mass and stiffness and the interface cantilevered modes and frequencies. Program PAYLOAD was run using the input deck as shown in Figure 18. Note that using the limited information at hand; (namely, the free mass and stiffness and the cantilevered modes and frequencies) all the flag keys - NEXP, NTP, NPMK12, NDET, NLOAD and NEP are zero, and we retained all of the cantilevered modes and frequencies (NP = 12). Figure 19 shows how subroutine ZPAYL interpreted this input record, and Table 6 shows a listing of the matrices that were generated and saved on NWFILE.

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	PAYL1	PHINP	12	12	10SE82	100%
2	PAYL1	FREQPT	1	12	10SE82	100%
3	PAYL1	TP	24	12	10SE82	100%
4	PAYL1	PM1	12	12	10SE82	100%
5	PAYL1	PM2	12	12	10SE82	100%
6	PAYL1	PK2	12	12	10SE82	100%
7	PAYL1	PL1	24	12	10SE82	100%
8	PAYL1	PL2	24	12	10SE82	100%

Table 6. Listing of the matrices output by subroutine ZPAYL for Payload 1



ALL BAR ELEMENTS

$\rho = 0.0025 \ \ lb/in^3$

$E = 10.6 \times 10^3 \ \ lb/in^2$

$G = 3.84 \times 10^6 \ \ lb/in^2$

$A = 0.01 \ \ in^2$

$J_0 = 1.67 \times 10^{-5} \ \ in^4$

Figure 17. Payload 1

ORIGINAL PAGE IS
OF POOR QUALITY

JOINT DATA USED IN SUBROUTINE FEMKA

JOINT	DEGREES OF FREEDOM: TRANSLATION U	V	W	ROTATION P	Q	R	GLOBAL CARTESIAN COORDINATES: X	Y	Z
1	1	2	3	0	0	0	2.5000	4 3300	0 0000
2	4	5	6	0	0	0	-2.5000	4 3300	0 0000
3	7	8	9	0	0	0	-2.5000	-1 0000	1 0000
4	10	11	12	0	0	0	-2 5000	-4.3300	0.0000
5	13	14	15	0	0	0	2.5000	-4 3300	0.0000
6	16	17	18	0	0	0	2.5000	-1.0000	1 0000
7	19	20	21	0	0	0	-2 5000	-1.0000	-1 0000
8	22	23	24	0	0	0	2.5000	-1.0000	-1.0000

Table 4. Geometry and degree of freedom table for payload 1

INPUT DATA FOR BAR ELEMENTS
KODEK = KODEB =
MASS = M1 STIF = K1 LOAD TRANS = PAY3LT STRESS TRANS =
RO = .250E-02 E = .106E+05 ALPHA = 0.
G = .384E+07

ELEMENT NUMBER	JOINT 1	JOINT 2	REF POINT	AREA	POLAR INERTIA	TORSION CONST	Z BENDING INERTIA	Y BENDING INERTIA	SHEAR FACTOR
1	5	1	2	100E-01	.167E-04	.141E-04	.833E-05	.833E-05	833
2	1	2	3	100E-01	167E-04	.141E-04	833E-05	.833E-05	.833
3	2	3	1	100E-01	.167E-04	141E-04	833E-05	833E-05	833
4	3	6	1	.100E-01	167E-04	.141E-04	.833E-05	833E-05	833
5	3	1	2	100E-01	167E-04	.141E-04	.833E-05	833E-05	.833
6	8	2	1	.100E-01	167E-04	141E-04	833E-05	833E-05	.833
7	3	4	6	.100E-01	.167E-04	.141E-04	.833E-05	833E-05	.833
8	4	5	1	100E-01	167E-04	141E-04	833E-05	833E-05	.833
9	5	6	3	100E-01	167E-04	141E-04	833E-05	.833E-05	833
10	4	6	1	.100E-01	167E-04	141E-04	833E-05	.833E-05	.833
11	7	5	4	.100E-01	167E-04	141E-04	.833E-05	833E-05	833
12	6	8	5	.100E-01	167E-04	141E-04	.833E-05	833E-05	833
13	3	7	4	100E-01	167E-04	.141E-04	.833E-05	833E-05	833
14	8	7	2	100E-01	.167E-04	.141E-04	833E-05	833E-05	833
15	5	8	6	.100E-01	.167E-04	.141E-04	833E-05	833E-05	833
16	4	7	3	.100E-01	.167E-04	.141E-04	833E-05	833E-05	833
17	1	8	6	100E-01	167E-04	.141E-04	833E-05	833E-05	833
18	2	7	3	100E-01	.167E-04	141E-04	833E-05	833E-05	833

Table 5. Payload I material properties

ORIGINAL PAGE IS
OF POOR QUALITY

```
PAYL1       TG SHANAHAN
PROGRAM ZPAYL FOR PAYLOAD 1
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
   12   12   24                          * IF,NP,ND *
   40   20    0                          * NWFILE,NWRKFL,NWRITE *
    0    0    0    0    0    0           * NEXP,NTP,NPMK12,NDET,NLOAD,NEP *
MODES    0   -31PAYL1           * NON-EXPANDED CANTILEVERED PAYLOAD MODES *
IFACE    1   12                 * VECTOR OF INTERFACE DOFS *
    1    1    1    2    3    4    5    6
    1    7   10   11   12   13   14   15
0000000000
MASS     0   -30PAYL1           * FREE PAYLOAD MASS MATRIX *
STIF     0   -30PAYL1           * FREE PAYLOAD STIFFNESS MATRIX *
FREQ     0   -31PAYL1           * CANTILEVERED PAYLOAD FREQUENCIES *
STOP
```

Figure 18. Input deck to program PAYLOAD for payload 1

```
                    INPUT VARIABLES TO ZPAYL
                    ------------------------

                        NWFILE =   40
                        NWRKFL =   20
                        NWRITE =    0

                     DESCRIPTION OF PAYLOAD
                     ----------------------

                  NUMBER OF PAYLOAD DOFS =   24          ) =   24
        NUMBER OF PAYLOAD INTERFACE DOFS =   12          . =   12
NUMBER OF TRUNCATED PAYLOAD MODES RETAINED =   12        N. =   12


THE MODES ARE NOT EXPANDED TO INCLUDE INTERFACE DOFS        (NEXP = 0)

              TP, PM1, PM2, AND PK2 ARE NOT GIVEN           NTP = 0)

                 THE INTERFACE IS NOT DETERMINANT           (NDET = 0)

  THE LOAD TRANSFORMATIONS PL1 AND PL2 ARE NOT GIVEN        (NLOAD = 0)

     THE CANTILEVERED FLEXIBILITY - EP IS NOT GIVEN         (NEP = 0)
```

Figure 19. Sample output from subroutine ZPAYL for payload 1

The same scheme used in the development of payload 2. This model as shown in Figure 20 consists of 28 bar elements joined at 12 node points to form a truss. The geometry and the degree of freedom table for the model are given in Table 7 and the material properties and connections are given in Table 8. As in payload 1 all nodes were given three translational degrees of freedom and the corner nodes comprise the interface. Therefore, for payload 2:

ND = 36, IF = 12
and IFACE = dof's 7, 8, 9, 10, 11, 12, 19, 20, 21, 22, 23 and 24

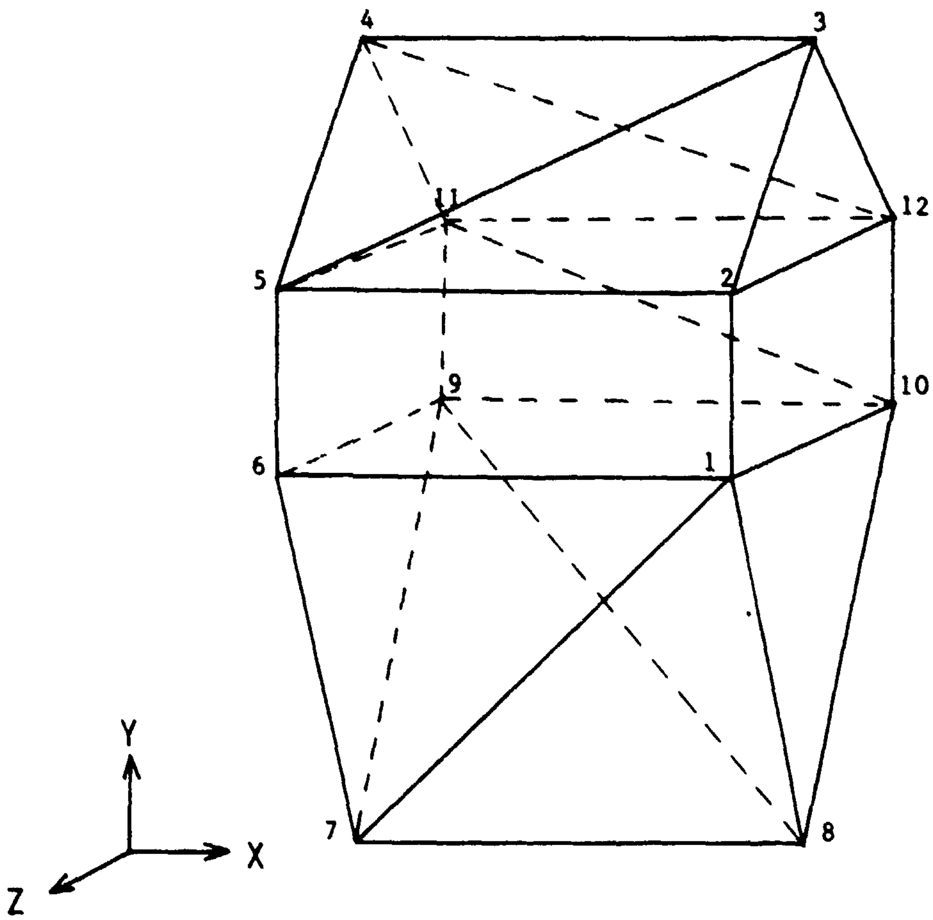
Models were created to generate the free-free mass and stiffness of the structure and the interface cantilevered modes and frequencies. With this information program PAYLOAD was run. Figure 21 shows the input deck to program PAYLOAD for payload 2. Note that all cantilevered modes and frequencies were retained (i.e. NP = 24), no other information other than that stated above was available (i.e. NEXP, NTP, NPMK12, NLOAD and NEP = 0), and the interface is indeterminate. Figure 22 is output from subroutine ZPAYL and shows how the subroutine interpreted the input deck. A listing of the resultant matrices from this subroutine is given in Table 9.

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	PAYL2	PHINP	24	24	11SE82	100%
2	PAYL2	FREQPT	1	24	11SE82	100%
3	PAYL2	TP	36	12	11SE82	100%
4	PAYL2	PM1	12	24	11SE82	100%
5	PAYL2	PM2	12	12	11SE82	100%
6	PAYL2	[illegible]	12	12	11SE82	100%
7	PAYL2	PL1	36	24	11SE82	100%
8	PAYL2	PL2	36	12	11SE82	100%

Table 9. Listing of the matrices output by subroutine ZPAYL for Payload 2

Now that all of the individual properties have been derived for the booster and each of the two payloads, the next step is to couple the three bodies together. Table 10 provides a map of how the booster/ payloads system is integrated.



ALL BAR ELEMENTS

ρ = 0.0025 lb/in^3

E = 10.6 x 10^3 lb/in^2

G = 3.84 x 10^6 lb/in^2

A = 0.01 in^2

J_0 = 1.67 x 10^{-5} in^4

Figure 20. Payload 2

ORIGINAL PAGE IS
OF POOR QUALITY

JOINT DATA USED IN SUBROUTINE FEMKA

JOINT	DEGREES OF FREEDOM TRANSLATION U	V	W	ROTATION P	Q	R	GLOBAL CARTESIAN COORDINATES X	Y	Z
1	1	2	3	0	0	0	2.5000	0.0000	1.0000
2	4	5	6	0	0	0	2.5000	2.0000	1.0000
3	7	8	9	0	0	0	2.5000	4.3300	0.0000
4	10	11	12	0	0	0	-2.5000	4.3300	0.0000
5	13	14	15	0	0	0	-2.5000	2.0000	1.0000
6	16	17	18	0	0	0	-2.5000	0.0000	1.0000
7	19	20	21	0	0	0	-2.5000	-4.3300	0.0000
8	22	23	24	0	0	0	2.5000	-4.3300	0.0000
9	25	26	27	0	0	0	-2.5000	0.0000	-1.0000
10	28	29	30	0	0	0	2.5000	0.0000	-1.0000
11	31	32	33	0	0	0	-2.5000	2.0000	-1.0000
12	34	35	36	0	0	0	2.5000	2.0000	-1.0000

Table 7. Geometry and degree of freedom table
for payload 2

INPUT DATA FOR BAR ELEMENTS
KODEK = KODEB =
MASS = M1 STIF = K1 LOAD TRANS = PAY4LT STRESS TRANS =
RO = .250E-02 E = .106E+05 ALPHA = 0.
G = .384E+07

ELEMENT NUMBER	JOINT 1	JOINT 2	REF POINT	AREA	POLAR INERTIA	TORSION CONST	Z BENDING INERTIA	Y BENDING INERTIA	SHEAR FACTOR
1	1	2	4	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
2	2	3	4	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
3	3	4	2	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
4	4	5	2	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
5	5	6	1	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
6	6	7	1	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
7	7	8	1	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
8	8	1	6	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
9	5	2	3	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
10	11	12	10	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
11	5	3	2	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
12	6	1	2	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
13	6	2	1	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
14	9	10	12	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
15	7	1	8	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
16	4	12	3	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
17	11	10	9	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
18	9	8	10	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
19	1	10	12	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
20	2	12	10	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
21	6	9	11	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
22	5	11	9	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
23	8	10	1	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
24	10	12	1	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
25	12	3	2	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
26	7	9	6	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
27	9	11	6	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833
28	11	4	5	.100E-01	.167E-04	.141E-04	.833E-05	.833E-05	.833

Table 8. Payload 2 material properties

ORIGINAL PAGE IS
OF POOR QUALITY

```
PAYL2       TG SHANAHAN
PROGRAM ZPAYL FOR PAYLOAD 2 FOR SAMPLE PAROBLEM
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
   12   24   36                       * IF,NP,ND *
   40   20    0                       * NWFILE,NWRKFL,NWRITE *
    0    0    0    0    0    0        * NEXP,NTP,NPMK12,NDET,NLOAD,NEP *
MODES    0  -31PAYL2                  * NON-EXPANDED CANILEVERED MODES *
IFACE    1   12                       * VECTOR OF INTERFACE DOFS *
    1    1    7    8    9   10   11   12
    1    7   19   20   21   22   23   24
0000000000
MASS     0  -30PAYL2                  * FREE MASS MATRIX *
STIF     0  -30PAYL2                  * FREE STIFFNESS MATRIX *
FREQ     0  -31PAYL2                  * CANTILEVERED FREQUENCIES *
STOP
```

Figure 21. Input deck to program PAYLOAD for payload 2

```
                    INPUT VARIABLES TO ZPAYL
                    ------------------------

                      NWFILE =   40
                      NWRKFL =   20
                      NWRITE =    0

                    DESCRIPTION OF PAYLOAD
                    ----------------------

                NUMBER OF PAYLOAD DOFS =   36          ND =   36
           NUMBER OF PAYLOAD INTERFACE DOFS =   12          IF =   12
  NUMBER OF TRUNCATED PAYLOAD MODES RETAINED =   24          NP =   24


THE MODES ARE NOT EXPANDED TO INCLUDE INTERFACE DOFS         (NEXP = 0)

               TP, PM1, PM2, AND PK2 ARE NOT GIVEN           NTP = 0)

                 THE INTERFACE IS NOT DETERMINANT            (NDET = 0)

  THE LOAD TRANSFORMATIONS PL1 AND PL2 ARE NOT GIVEN         (NLOAD = 0)

      THE CANTILEVERED FLEXIBILITY - EP IS NOT GIVEN         (NEP = 0)
```

Figure 22. Sample output from subroutine ZPAYL for payload 2

BOOSTER		PAYLOAD 1		PAYLOAD 2	
Interface dof no	Overall dof no	Interface dof no	Overall dof no	Interface dof no	Overall dof no
1	22	1	1		
2	23	2	2		
3	24	3	3		
4	25	4	4		
5	26	5	5		
6	27	6	6		
7	31	7	10		
8	32	8	11		
9	33	9	12		
10	34	10	13		
11	35	11	14		
12	36	12	15		
13	40				
14	41				
15	42				
16	43				
17	44		SUPERFLUOUS INTERFACE dofs		
18	45				
19	49				
20	50				
21	51				
22	52				
23	53				
24	54				
25	58			1	7
26	59			2	8
27	60			3	9
28	61			4	10
29	62			5	11
30	63			6	12
31	67			7	19
32	68			8	20
33	69			9	21
34	70			10	22
35	71			11	23
36	72			12	24

Table 10 Map of Booster and Payload Interface dofs

Program INTFACE does the actual coupling of booster and payload quantities. We already know the following inputs to subroutine ZINTF:

$$NPAY = 2 \;, \qquad IFB = 36$$

and,

$$NPTOT = NP_{payload\ 1} + NP_{payload\ 2}$$
$$= 12 + 24$$
$$= 36$$

All other quantities were output from subroutines ZBOOST or ZPAYL except for the IFACEP vectors. Each IFACEP vector te..ls how the payload is to be attached to the booster. To determine the IFACEP vector, we assign each booster dof a number from 1 to IFB in increasing order of overall dof number. Likewise, we do the same for each payload. Then, for each payload the IFACEP vector tells which booster interface dof number corresponds to the payload interface dof number. We now have all the information necessary to run program INTFACE. Figure 23 shows the input deck to run this program, and Figure 24 shows some ' mple output from subroutine ZINTF which tells how the input was interpreted. A listing of matrices output from this subroutine is located in Table 11.

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	ZINTF	FREQI	1	36	11SE82	100%
2	ZINTF	B2	36	54	11SE82	100%
3	ZINTF	P2	36	36	11SE82	100%
4	ZINTF	BPM2	36	36	11SE82	100%
5	ZINTF	BPK2	36	36	11SE82	100%
6	ZINTF	PHIIB	36	36	11SE82	100%
7	ZINTF	FREQPA	1	36	11SE82	100%

Table 11. Listing of the matrices output by subroutine ZINTF for sample problem 1

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZINTF        TG SHANAHAN
PROGRAM ZINTF WITH BOOSTER 1 AND PAYLOADS 1 AND 2 FOR SAMPLE PROBLEM
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
    2                                * NPAY *
   40    10    0                     * NWFILE,NWRKFL,NWRITE *
   36    36                          * IFB,NPTOT *
BM1      0   -30ZBOOST               * BOOSTER COUPLING MASS MATRIX *
BM2      0   -30ZBOOST               * BOOSTER MASS MATRIX REDUCED TO INTERFACE *
BK2      0   -30ZBOOST               * BOOSTER STIFFNESS REDUCED TO INTERFACE *
   12    12                          * IFP,NP - FOR PAYLOAD 1 *
IFACEP   1    12                     * VECTOR OF INTERFACE DOFS FOR PAYLOAD 1 *
    1    1     1    2    3    4    5    6
    1    7     7    8    9   10   11   12
0000000000
PM1      0   -31PAYL1                * PAYLOAD 1 COUPLING MASS MATRIX *
PM2      0   -31PAYL1                * PAYLOAD 1 MASS REDUCED TO INTERFACE *
PK2      0   -31PAYL1                * PAYLOAD 1 STIFFNESS REDUCED TO INTERFACE *
FREQPT   0   -31PAYL1                * PAYLOAD 1 TRUNCATED, CANTILEVERED FREQ *
   12    24                          * IFP,NP - FOR PAYLOAD 2 *
IFACEP   1    12                     * VECTOR OF INTERFACE DOF FOR PAYLOAD 2 *
    1    1    25   26   27   28   29   30
    1    7    31   32   33   34   35   36
0000000000
PM1      0   -32PAYL2                * PAYLOAD 2 COUPLING MASS MATRIX *
PM2      0   -32PAYL2                * PAYLOAD 2 MASS REDUCED TO INTERFACE *
PK2      0   -32PAYL2                * PAYLOAD 2 STIFFNESS REDUCED TO INTERFACE *
FREQPT   0   -32PAYL2                * PAYLOAD 2 TRUNCATED, CANTILEVERED FREQ *
STOP
```

Figure 23. Input deck to program INTFACE
for sample problem 1

ORIGINAL PAGE IS
OF POOR QUALITY

```
                         INPUT DATA TO ZINTF
                         -------------------


                          NWFILE  =   40
                          NWRKFL  =   10

                          NWRITE  =    0

      NUMBER OF BOOSTER INTERFACE DOFS  =  36        (IFB =   36)

   NUMBER OF TOTAL PAYLOAD MODES RETAINED  =  36     (NPTOT =   36)



                   INPUT DATA FOR THE    2 PAYLOADS
                   ------------------  -------------


                     INPUT DATA FOR PAYLOAD    1
                     --------------------------

             NUMBER OF PAYLOAD INTERFACE DOFS  =  12

          NUMBER OF PAYLOAD TRUNCATED MODES USED  =  12

          IVEC OF PAYLOAD INTERFACE DOF LOCATIONS IN THE BOOSTER INTERFACE
          ----------------------------------------------------------------


IFACEP (   1 X  12 )   /INPUT/            * VECTOR OF INTERFACE DOFS FOR PAYLOAD 1 *

    1    1    1    2    3    4    5    6    0    0    0    0    0    0
    1    7    7    8    9   10   11   12
END OF READIM.
*******************************************************************************************


                     INPUT DATA FOR PAYLOAD    2
                     --------------------------

             NUMBER OF PAYLOAD INTERFACE DOFS  =  12

          NUMBER OF PAYLOAD TRUNCATED MODES USED  =  24

          IVEC OF PAYLOAD INTERFACE DOF LOCATIONS IN THE BOOSTER INTERFACE
          ----------------------------------------------------------------


IFACEP (   1 X  12 )   /INPUT/            * VECTOR OF INTERFACE DOF FOR PAYLOAD 2 *

    1    1   25   26   27   28   29   30    0    0    0    0    0    0
    1    7   31   32   33   34   35   36
END OF READIM.
*******************************************************************************************
```

Figure 24. Sample output from subroutine ZINTF

Now that the models are couple together, the only other data needed to determine the response is the time/force history. Program FORCE takes input time/force data and outputs it in a form compatable with the response program. For this sample problem, there are four force points (NF = 4) 36 booster interface dof (IF = 36) and 54 booster modes retained (NB = 54). We have set the start time for the force data to zero seconds, the end time = 1.0 seconds and the time step = 0.01 seconds. The force data input to this program was a sampling over the time range of t = 0.0 seconds to t = 1.0 seconds for the functions:

$$F_{78} = 150000 \sin[3.1416\ (t - .001)]$$

$$F_{81} = 160000 \sin[2.8274\ (t - .002)]$$

$$F_{87} = 125000 \sin[3.7700\ (t - .02)]$$

$$F_{90} = 170000 \sin[3.4558\ (t - .01)]$$

The input deck to program FORCE is given in Figure 25, the interpretation of this data is echoed by subroutine ZFORCE in Figure 26. Output on NFORCE is sequential.

Program RESPONS then calculates the response by reading in all the booster, payload and force quantities and entering them in a direct numerical integration technique. Note that as input to program RESPONS the input start time and time step must be equal to that written on the input NFORFL, and the end time must be less than the end time on NFORFL. For our case we have STARTT = 0.0 seconds, ENDT = 0.90 seconds and DELTAT = 0.01 seconds. As well as properties derived in programs BOOSTER, PAYLOAD, INTFACE and FORCE, we must now input the modal damping. We choose to ignore damping for this model, therefore, we have a constant value of damping for booster, interface and payload (i.e. ZETAB = ZETAI = ZETAP = 0.0 and NDAMPB = NDAMPI = NDAMPP = 0). We also used the recommended values of γ and β for the Newmark-Chan-Beta numerical integration technique ($\gamma = 0.5$, $\beta = 0.25$). The input to program RESPONSE is listed in Figure 27. Notice that the only information needed to be read in are the coupling properties B2 and P2, the interface modes, PHIIB and frequencies, FREQI, the cantilevered frequencies for the payloads, FREQPA and booster, FREQBT, the damping terms as mentioned above and the initial velocities and displacements in modal coordinates. For our case the initial conditions have been set to zero. Subroutine ZRESP echoes the input data back as shown in Figure 28. Output data from ZRESP are the modal response for the interface and payload. These results are written on TAPE 40 = NWFILE.

```
ZFORCE      TG SHANAHAN
PROGRAM ZFORCE FOR BOOSTER 1 SAMPLE PROBLEM
PART OF COMPLETE BOOSTER-PAYLOAD INTEGRATION SOFTWARE PACKAGE
   20   10   40                   * NWRKFL,NWRITE,NFORFL *
    0          1.00     0.01      * STARTT,ENDT,DELTAT *
    4   54   36                   * NF,NB,IF *
    0                             * IFTERP *
TIME     0  -30SAMPLE             * TIME TABLE *
FORCE    0  -30SAMPLE             * FORCE TABLE *
PHINBR   0  -31ZBOOST             * REDUCED CANTILEVERED BOOSTER MODES *
TBP      0  -31ZBOOST             * REDUCED BOOSTER CONSTRAINT MODAL MAT *
STOP
```

Figure 25. Input deck to program FORCE

```
                    INPUT PARAMETERS
                    ----------------


                 NWRKFL  =   20
                 NFORFL  =   40

   DATA IS WRITTEN ON PAPER EVERY     10 TIME STEPS


                 STARTT  =  0.000000
                   ENDT  =  1.000000
                 DELTAT  =   .010000

   NUMBER OF FORCES APPLIED TO BOOSTER  =    4

     NUMBER OF TRUNCATED BOOSTER MODES  =   54

              NUMBER OF INTERFACE DOFS  =   36

INTERPOLATION OF THE FORCE DATA IS NEEDED     (IFTERP = 0)
```

Figure 26. Sample output from subroutine ZFORCE

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZRESP      TG SHANAHAN
PROGRAM ZRESP   - SAMPLE PROBLEM WITH BOOSTER 1, PAYL 1, AND PAYL2
PART OF A COMPLETE BOOSTER-PAYLOAD INTEGRATION SOFTWARE PACKAGE
   40   20    5   30              * NRESFL,NWRKFL,NWRITE,NFORFL *
    0.         0.90       0.01    * STARTT,ENDT,DELTAT *
    0.5        0.25               * GAMMA,BETA *
    4   54   36   36              * NF,NB,IF,NP *
    0    0    0                   * NDAMPB,NCAMPI,NDAMPP *
FREQBT    0   -31ZBOOST           * TRUNCATED, CANTILEVERED BOOSTER FREQ *
  0.000                           * ZETAB *
FREQI     0   -32ZINTF            * INTERFACE FREQUENCIES *
  0.000                           * ZETAI *
FREQPA    0   -32ZINTF            * ASSEMBLED PAYLOAD FREQUENCIES *
  0.000                           * ZETAP *
B2        0   -32ZINTF              INTERFACE,BOOSTER COUPLING MASS MATRIX *
P2        0   -32ZINTF            * INTERFACE,PAYLOAD COUPLING MASS MATRIX *
PHIIB     0   -32ZINTF            * INTERFACE MODES *
QNBO      1   54                  * INITIAL BOOSTER DISPLACEMENTS *
0000000000
QNBDO     1   54                  * INITIAL BOOSTER VELOCITIES *
0000000000
QIB       1   36                  * INITIAL INTERFACE DISPLACEMENTS *
0000000000
QIBDO     1   36                  * INITIAL INTERFACE VELOCITIES *
0000000000
QNPO      1   36                  * INITIAL PAYLOAD DISPLACEMENTS *
0000000000
QNPDO     1   36                  * INITIAL PAYLOAD VELOCITITES *
0000000000
STOP
```

Figure 27. Input deck to program RESPONS

```
                     INPUT PARAMETERS
                     ----------------

                   NWFILE  =   40
                   NWRKFL  =   20
                   NFORFL  =   30

DATA IS WRITTEN ON PAPER EVERY      5 TIME STEPS

                   STARTT  =   0.000000
                     ENDT  =    .900000
                   DELTAT  =    .010000

PARAMETERS FOR NEWMARK-CHAN-BETA INTEGRATION ROUTINE

                    GAMMA  =    .500000
                     BETA  =    .250000

VALUE OF BOOSTER MODAL DAMPING IS CONSTANT      (NDAMPB = 0)

VALUE OF PAYLOAD MODAL DAMPING IS A CONSTANT      (NDAMPP = 0)

NUMBER OF FORCES APPLIED TO BOOSTER  =    4

  NUMBER OF TRUNCATED BOOSTER MODES  =   54

          NUMBER OF INTERFACE DOFS  =   36

  NUMBER OF TRUNCATED PAYLOAD MODES  =   36
```

Figure 28. Sample output from subroutine ZRESP

Program LOADS can be used to generate many useful quantities. Using the ISELECT option we can choose an area of study for the body, and by choosing PKPSI judiciously, we can generate either elastic forces, member load, stresses, strains, etc. For this sample we choose to generate elastic node forces for the all nodes in payload 1, thus ISELECT = 0 and PKSPI = the free stiffness matrix for the structure. The payload non-interface dofs are ordered as payload 1 then payload 2, thus NQNPS = 1 for this calculation and NQNP = 12. The input deck to program LOADS is given in Figure 29. Subroutine ZLOADS interprets the input file, reads other information from NRESFL and prints it as given in Figure 30. Since we chose MAXL = 1, a max/min search then results of this are given in Table 12.

SAMPLE PROBLEM 2

As stated previously, this sample problem consists of a booster, the space shuttle (STS), and two payloads; the space telescope (ST) and the OMS kit. Various mass, stiffness and modal properties were received from NASA, from which all properties necessary to run the payload integration software pack can be derived:

The STS model received contained 759 overall dofs of which 33 are interface dofs. The following information about this model was received:

$$[PH] = [TB \mid I_{\beta}\ \Phi_N^{\beta}] \qquad (759 \times 333)$$

$$[MOCB] = [PH]^T [M] [PH] \qquad (333 \times 333)$$

$$[KOCB] = [PH]^T [K] [PH] \qquad (333 \times 333)$$

and the force and time tables [TABF] and [TABT]. From this data it was easy to section out or derive the following properties:

1) the expanded cantilevered modes PAINBX
2) the constraint modal matrix - TB
3) the STS mass and stiffness matrices reduced to the interface - BM2 and BK2
4) the STS coupling mass matrix - BM
5) the cantilevered frequencies - FREQ

Three hundred booster modes were retained. The force data reduced to 120 dofs that have non-zero forces. These are listed in Figure 33. Using the above data we have the following input to program BOOSTER

IF = 33, NB = 300, ND = 759, NF = 120

NEXP = 1, NTB = 1, NBMK12 = 1, NDET = 0

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZLOADS       TG SHANAHAN
PROGRAM ZLOADS FOR PAYLOAD 1
BOOSTER 1, PAYLOAD 1, PAYLOAD 2 SYSTEM
    1     0    10                    * MAXL,ISELECT,NWRITE *
    1    12                          * NONPS,NONP *
   36    12    36                    * IFB,IFP,NP *
   30    20    40    41              * NRESPFL,NWRKFL,NWFILE,NTAPE *
STIF      0   -32PAYL1               * PAYLOAD 1 FREE STIFFNESS MATRIX *
PL1       0   -31PAYL1               * PAYLOAD 1 LOAD TRANSFORMATION *
PL2       0   -31PAYL1               * PAYLOAD 1 LOAD TRANSFORMATION *
TP        0   -31PAYL1               * PAYLOAD 1 CONSTRAINT MODAL MATRIX *
PHIIB     0   -33ZINTF               * INTERFACE MODES *
IVEC      1    12                    * VECTOR OF PAYLOAD 1 INTERFACE LOCATIONS *
    1     1     1     2     3     4     5     6     7     8     9    10
    1    11    11    12
0000000000
```

Figure 29. Input deck to program LOADS

```
                     INPUT PARAMTERS TO ZLOADS
                     -------------------------

         MAXIMUN/MINIMUM LOAD CALCULATION WILL BE PERFORMED      (MAXL = 1)

       ALL ROWS OF PKPSI ARE USED IN THE LOAD CALCULATION      (ISELECT = 0)

                            NWRITE =    10

           THE NUMBER OF INTERFACE DOFS IN THIS PAYLOAD =    12

THE ROW NUMBER OF THE FIRST NON-INTERFACE DOF FOR THIS PAYLOAD IN THE BOOSTER =     1

          THE NUMBER OF NON-INTERFACE DOFS IN THIS PAYLOAD =    12

           THE NUMBER OF INTERFACE DOFS IN THE BOOSTER =    36

            THE TOTAL NUMBER OF PAYLOAD MODES RETAINED =    36

                           NRESPFL =    30

                            NWRKFL =    20

                            NWFILE =    40

                             NTAPE =    41



                           PAYLOAD LOADS
                           -------------

                      FOR THE TIME INTERVAL OF

                         STARTT =     0 000000

                           ENDT =       900000

                         DELTAT =       010000
```

Figure 30. Sample output from subroutine ZLOADS
for sample problem 1

ORIGINAL PAGE IS
OF POOR QUALITY

MAXIMUM AND MINIMUM LOADS

FORM MAXIMUM, TIME AT MAXIMUM, MINIMUM, TIME AT MINIMUM

MAXMIN (24 X 4) /OUTPUT/

		(1)	(2)	(3)	(4)	(5)
1	1	8.085E+01	8.900E-01	-7.888E+01	4.500E-01	
2	1	1.324E+02	9.000E-01	-1.278E+03	5.200E-01	
3	1	7.790E+01	4.900E-01	-2.439E-02	3.000E-02	
4	1	1.978E+02	5.300E-01	-1.382E+01	9.000E-02	
5	1	2.163E+01	9.000E-02	-1.405E+03	5.300E-01	
6	1	9.552E+00	9.000E-02	-4.443E+01	5.400E-01	
7	1	2.715E+00	5.900E-01	-2.998E+00	6.[illegible]00E-01	
8	1	2.612E+00	9.000E-01	-1.855E+00	5.400E-01	
9	1	3.529E-02	3.000E-02	-2.706E+01	4.900E-01	
10	1	7.956E+01	4.700E-01	-8.202E+01	9.000E-01	
11	1	1.210E+03	5.200E-01	-1.455E+02	9.000E-01	
12	1	8.983E+01	5.000E-01	-3.448E-02	3.000E-02	
13	1	1.352E+01	9.000E-02	-1.894E+02	5.400E-01	
14	1	1.473E+03	5.300E-01	-2.059E+01	9.000E-02	
15	1	1.666E+01	8.100E-01	-3.379E+01	5.300E-01	
16	1	2.311E+00	6.300E-01	-2.344E+00	6.100E-01	
17	1	2.454E+00	9.000E-01	-1.652E+00	4.400E-01	
18	1	3.887E-02	3.000E-02	-2.577E+01	4.900E-01	
19	1	2.424E+00	5.900E-01	-2.363E+00	8.800E-01	
20	1	2.060E+00	8.500E-01	-1.762E+00	4.200E-01	
21	1	3.167E-02	3.000E-02	-2.561E+01	4.900E-01	
22	1	2.916E+00	5.900E-01	-2.903E+00	6.100E-01	
23	1	1.876E+00	8.900E-01	-1.636E+00	4.200E-01	
24	1	3.901E-02	3.000E-02	-2.718E+01	4.900E-01	

END OF WRITE.

Table 12. Max/Min elastic forces for Sample problem 1. Output from program ZLOADS

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZBOOST      TG SHANAHAN
PROGRAM ZBOOST   FOR STS    MFSC MODEL
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
   33  300  759  120             * IF,NB,ND,NF *
   40   20    1                  * NWFILE,NWRKFL,NWRITE *
    1    1    1    0             * NEXP,NTB,NBMK12,NDET *
PHINB    0  -30BOSGEN            * EXPANDED CANTILEVERED BOOSTER MODES *
NODOR    0  -32NODDOF            * VECTOR OF FORCE LOCATIONS *
TB       0  -30BOSGEN            * CONSTRAINT MODAL MATRIX *
BM1      0  -30BOSGEN            * BOOSTER COUPLING MASS MATRIX *
BM2      0  -30BOSGEN            * BOOSTER INTERFACE MASS MATRIX *
BK2      0  -30BOSGEN            * BOOSTER COUPLING STIFFNESS MATRIX *
FREQB2   0  -31BOSGEN            * BOOSTER CANTILEVERED FREQUENCIES *
STOP
```

Figure 31. Input deck to program BOOSTER for the Space Shuttle (STS)

```
                         INPUT VARIABLES TO ZBOOST
                         -------------------------

                            NWFILE =   40
                            NWRKFL =   20
                            NWRITE =    1

                          DESCRIPTION OF BOOSTER
                          ----------------------

                   NUMBER OF BOOSTER DOFS =  759            ND =  759
           NUMBER OF BOOSTER INTERFACE DOFS =   33          IF =   33
  NUMBER OF TRUNCATED BOOSTER MODES RETAINED =  300         NB =  300
      NUMBER OF FORCES APPLIED TO THE BOOSTER =  120        NF =  120


THE MODES ARE EXPANDED TO INCLUDE INTERFACE DOFS        (NEXP = 1)

     TB - THE CONSTRAINT MODAL MATRIX IS GIVEN           (NTB = 1)

            BM1, BM2, AND BK2 ARE AVAILABLE              (NBMK12 = 1)

           THE INTERFACE IS NOT DETERMINANT              (NDET = 0)
```

Figure 32. Sample output from subroutine ZBOOST for the Space Shuttle (STS)

NFORCE (1 X 120) /OUTPUT/

		(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	(11)	(12)	(13)	(14)	(15)	(16)	(17)	(18)	(19)	(20)
1	1	14	15	26	27	38	39	50	51	55	56	57	58	79	80	113	114	118	120	132	134
1	21	143	145	186	187	210	211	212	213	214	215	219	220	221	222	223	224	231	232	233	234
1	41	235	236	256	257	265	266	274	275	292	294	298	299	300	326	501	502	504	505	508	521
1	61	522	530	537	538	542	550	572	573	582	583	584	585	586	587	588	590	591	592	594	595
1	81	596	598	599	600	601	·02	603	604	605	606	607	608	609	610	671	672	673	674	675	676
1	101	677	679	680	681	683	684	685	687	688	689	690	691	692	693	694	695	696	697	698	699

END OF WRITIA

Figure 33. Listing of *STS degrees of freedom* that have non-zero applied forces

A listing of the input file to program BOOSTER is given in Figure 31, and the program interpretation is given in Figure 32. Subroutine ZBOOST processes these quantities and output the matrices as listed in Table 13.

RUN NO ZBOOST

PROGRAM ZBOOST FOR STS MFSC MODEL
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	ZBOOST	PHINB	759	300	11SE82	100%
2	ZBOOST	TB	759	33	11SE82	88%
3	ZBOOST	PHINBR	120	300	11SE82	100%
4	ZBOOST	FREQBT	1	300	11SE82	100%
5	ZBOOST	TBR	120	33	11SE82	100%
6	ZBOOST	BM1	33	300	11SE82	100%
7	ZBOOST	BM2	33	33	11SE82	100%
8	ZBOOST	BK2	33	33	11SE82	100%

Table 13. Listing of matrices output from subroutine ZBOOST for the Space Shuttle

ORIGINAL PAGE IS
OF POOR QUALITY

The space telescope modal contains 214 dofs of which 6 are interface dofs, 175 cantilevered payload modes were retained. The following information was received from NASA:

$[M]$	= free mass matrix	(214 x 214)
$[K]$	= free stiffness matrix	(214 x 214)
$[I_B \bar{\Phi}_N^{\Pi}]$	= the interface expanded cantilevered modes	(214 x 175)
$[T_{P1}]$	= the constrain modal matrix	(214 x 6)
$\{\bar{f}_p\}$	= the cantilevered frequencies	(175)

In order to couple these properties with the booster it was necessary to reverse the direction of one of the interface dofs. This altered the free mass and stiffness matrices. Also, note that the interface is determinate (i.e. the number of interface dofs is less than or equal to the number of rigid body dofs). The interface consists of dofs 1, 2, 3, 4, 5 and 6. Thus, the inputs to program PAYLOAD for the space telescope are:

IF = 6, NP = 175, ND = 214
NEXP = 1, NATP = 0, NPMK12 = 0, NDET = 1,
NLOAD = 0, NEP = 0,
IFACE = dofs 1, 2, 3, 4, 5, 6

A listing of the input deck to this program is given in Figure 34. This data is interpreted and echoed back by subroutine ZPAYL as shown in Figure 35. Also, a listing of the output matrices from this subroutine is given in Table 14 . Note that since the interface is determinate PK2 = 0.

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	ZPAYL	PHINP	214	175	11SE82	100%
2	ZPAYL	FREQPT	1	175	11SE82	100%
3	ZPAYL	TP	214	6	11SE82	12%
4	ZPAYL	PM1	6	175	11SE82	100%
5	ZPAYL	PM2	6	6	11SE82	100%
6	ZPAYL	PK2	6	6	11SE82	0%
7	ZPAYL	PL1	214	175	11SE82	100%
8	ZPAYL	PL2	214	6	11SE82	100%

Table 14. Listing of the matrices output by subroutine ZPAYL for the Space Telescope

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZPAYL        TG SHANAHAN
      PROGRAM ZPAYL FOR THE SPACE TELESCOPE
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
   6  175  214                         * IF,NP,ND *
  40   20    1                         * NWFILE,NWRKFL,NWRITE *
   1    0    0    1    0    0          * NEXP,NTP,NPMK12,NDET,NLOAD,NEP *
PHIEXP    0  -3OSTDATA                 * EXPANDED CANILEVERED MODES *
IFACE     1   6                        * VECTOR OF INTERFACE DOFS *
   1    1    1    2    3    4    5    6
0000000000
MREV      0  -3OSTDATA                 * FREE MASS *
KREV      0  -3OSTDATA                 * FREE STIFFNESS *
FREQ      0  -3OSTDATA                 * CANILEVERED FREQUENCIES *
STOP
```

Figure 34. Input deck to program PAYLOAD for the Space Telescope (ST)

```
                    INPUT VARIABLES TO ZPAYL
                    ------------------------

                       NWFILE =   40
                       NWRKFL =   20
                       NWRITE =    1

                     DESCRIPTION OF PAYLOAD
                     ----------------------

              NUMBER OF PAYLOAD DOFS =  214          ND =  214
     NUMBER OF PAYLOAD INTERFACE DOFS =    6          IF =    6
 NUMBER OF TRUNCATED PAYLOAD MODES RETAINED =  175    NP =  175


 THE MODES ARE EXPANDED TO INCLUDE INTERFACE DOFS        (NEXP = 1)

           TP, PM1, PM2, AND PK2 ARE NOT GIVEN           NTP = 0)

                THE INTERFACE IS DETERMINANT             (NDET = 1)

THE LOAD TRANSFORMATIONS PL1 AND PL2 ARE NOT GIVEN       (NLOAD = 0)

   THE CANTILEVERED FLEXIBILITY - EP IS NOT GIVEN        (NEP = 0)
```

Figure 35 Sample output form subroutine ZPAYL for the Space Telescope (ST)

For the OMS kit, the free mass and stiffness matrices and a matrix map telling interface locations were received. From this information the interface cantilevered modes and frequencies were calculated. In total, the model contained 36 dofs of which 7 are interface dofs. Thus, with this bare minimum information about the OMS Kit, we were able to run the payload program. As inputs we have IF =7, NP = 14 (only 14 cantilevered modes were to be retained and ND = 36. The input deck to program PAYLOAD for the OMS kit is given in Figure 36. and the program echoed it as in Figure 37. The output matrices are listed in Table 15.

Using the output for these three previous runs construction of the input deck to program ZINTF as shown in Figure 38 is quite simple. We know there are two payloads (NPAY = 2) and 189 total payload modes ($NP + NP_{ST} + NP_{OMS\ KIT} = 175 + 14$). The IFACEP vectors were generated from Table 16. Figures 39 show how subroutine ZINTF interpreted the data deck and Table 17 lists the output matrices and their sizes.

The force/time history for the forcing function was transmitted via three matrices:

1)	TABT -	table of break point times (1741 x 1)
2)	TABF -	table of forces (1741 x 185)
3)	IDFORM -	table of force locations (185 x 1)

TABF gives the time/force history. The 1741 rows each correspond to the time listed for the same element in matrix TABT. Each column of TABF corresponds to a location that is described by the appropriate element of matrix IDFORM. Each element of IDFORM contained a four digit integer number. The first three digits told the node number to which the force was to be applied, the last digit specified the direction. For example:

IDFORM (J) = 8583

means force number J is applied at node 858 in the 3 or z-direction. Using a provided matrix map of the STS model it was then possible to determine the degree of freedom number to which the force was to be applied. Careful inspection of IDFORM revealed that in several cases more than one force was to be applied to a particular degree of freedom. Thus, these forces were superimposed and resolved into one force vector. After processing this data, 120 independent force vectors remained. The forces were then arranged in order of increasing dof number which resulted in the NFORCE vector listed in Figure 33. Thus, the following matrices were input to program FORCE:

-	TABT3	time vector	(1741 x 1)
-	TABF3	force table	(1741 x 185)
-	PHINBR	reduced, truncated, cantilevered booster modes	(185 x 300)
-	TBR	booster constraint modal matrix	(185 x 33)

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZPAYL       TG SHANAHAN
      PROGRAM ZPAYL FOR THE OMS KIT
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
    7    14    36                          * IF,NP,ND *
   40    20     1                          * NWFILE,NWRKFL,NWRITE *
    0     0     0   0     0     0          * NEXP,NTP,NPMK12,NDET,NLOAD,NEP *
PHIOMS    0  -30EGVL1                      * NON-EXPANDED CANILEVERED MODES *
IFACE     1     7                          * VECTOR OF INTERFACE DOFS *
    1     .     1     3     6     7    9    12    14
0000000000
MSSOMS    0  -30EGVL1                      * FREE MASS *
STFOMS    0  -30EGVL1                      * FREE STIFFNESS *
FRCOMS    0  -30EGVL1                      * CANILEVERED FREQUENCIES *
STOP
```

Figure 36. Input deck to program PAYLOAD for the OMS Kit

```
                    INPUT VARIABLES TO ZPAYL
                    ------------------------

                       NWFILE =   40
                       NWRKFL =   20
                       NWRITE =    1

                     DESCRIPTION OF PAYLOAD
                     ----------------------

                  NUMBER OF PAYLOAD DOFS =   36          ND =   36
        NUMBER OF PAYLOAD INTERFACE DOFS =    -          IF =    7
 NUMBER OF TRUNCATED PAYLOAD MODES RETAINED =   14       NP =   14


THE MODES ARE NOT EXPANDED TO INCLUDE INTERFACE DOFS     (NEXP = 0)

               TP, PM1, PM2, AND PK2 ARE NOT GIVEN       NTP = 0)

                   THE INTERFACE IS NOT DETERMINANT      (NDET = 0)

  THE LOAD TRANSFORMATIONS PL1 AND PL2 ARE NOT GIVEN     (NLOAD = 0)

     THE CANTILEVERED FLEXIBILITY - EP IS NOT GIVEN      (NEP = 0)
```

Figure 37. Sample output from subroutine ZPAYL for the OMS Kit

STS (BOOSTER)		ST (PAYLOAD 1)		OMS KIT (PAYLOAD 2)	
Interface dof no	Overall dof no	Interface dof no	Overall dof no	Interface dof no	Overall dof no
1	385				
2	387				
3	390				
4	392				
5	395				
6	397				
7	400	5	5		
8	402	4	4		
9	405				
10	407				
11	410			1	1
12	412			2	3
13	415				
14	417			3	6
15	420				
16	422				
17	425				
18	427	1	1		
19	430				
20	432				
21	435	3	3		
22	437	2	2		
23	440				
24	442				
25	445			4	7
26	447			5	9
27	450				
28	452			6	12
29	456				
30	459				
31	462	6	-6		
32	465				
33	468			7	14

Table 16. Listing of Interface for ST-STS-OMS Kit Model

LISTING OF MATRICES ON LOGICAL UNIT 40

NO	RUN NO	NAME	NROWS	NCOLS	DATE	DENSITY
1	ZPAYL	PHINP	29	29	11SE82	100%
2	ZPAYL	FREQPT	1	14	1 SE82	100%
3	ZPAYL	TP	36	7	11SE82	100%
4	ZPAYL	PM1	7	14	11SE82	100%
5	ZPAYL	PM2	7	7	11SE82	100%
6	ZPAYL	PK2	7	7	11SE82	100%
7	ZPAYL	PL1	36	14	11SE82	100%
8	ZPAYL	PL2	36	7	11SE82	100%

Table 15. Listing of the matrices output by subroutine ZPAYL for the OMS Kit

LISTING OF MATRICES ON LOGICAL UNIT 40

NO.	RUN NO	NAME	NROWS	NCOLS	DATE	DENSI
1	ZINTF	FREQI	1	33	11SE82	100%
2	ZINTF	B2	33	300	11SE82	100%
3	ZINTF	P2	33	189	11SE82	100%
4	ZINTF	BPM2	33	33	11SE82	100%
5	ZINTF	BPK2	33	33	11SE82	100%
6	ZINTF	PHIIB	33	33	11SE82	100%
7	ZINTF	FREQPA	1	189	11SE82	100%

Table 17. Listing of the matrices output by subroutine ZINTF for the ST-STS-OMS Kit system

```
ZINTF        TG SHANAHAN
PROGRAM ZINTF FOR THS ST-STS-OMS KIT MODEL
PART OF PAYLOAD INTEGRATION SOFTWARE PACKAGE
   2                                 * NPAY *
  40    10     1                     * NWFILE.NWRKFL.NWRITE *
  33   189                           * IFB.NFTOT *
BM1      0   -30ZBOOST               * SHUTTLE COUPLING MASS MATRIX *
BM2      0   -30ZBOOST               * SHUTTLE MASS MATRIX REDUCED TO INTERFACE *
BK2      0   -30ZBOOST               * SHUTTLE STIFFNESS REDUCED TO INTERFACE *
   6   175                           * IFP.NP  - FOR THE SPACE TELESCOPE *
IFACEP   1     5                     * VECTOR OF ST INTERFACE DOFS *
   1     1    18    22    21     8     7    31
0000000000
PM1      0   -31ZPAYL                * ST COUPLING MASS MATRIX *
PM2      0   -31ZPAYL                * ST MASS MATRRIX REDUCED TO INTERFACE *
PK2      0   -31ZPAYL                * ST STIFFNESS REDUCED TO INTERFACE *
FREQPT   0   -31ZPAYL                * ST TRUNCATED. CANTILEVERED FREQUENCIES *
   7    14                           * IFP. NP FOR OMS KIT *
IFACEP   1     7                     * VECTOR OF OMS KIT INTERFACE DOFS *
   1     1    11    12    14    25    26    28    33
0000000000
PM1      0   -32ZPAYL                * OMS KIT COUPLING MASS MATRIX *
PM2      0   -32ZPAYL                * OMS KIT MASS MATRIX REDUCED TO INTERFACE *
PK2      0   -32ZPAYL                * OMS KIT STIFFNESS REDUCED TO INTERFACE *
FREQPT   0   -32ZPAYL                * OMS KIT TRUNCATED. CANTILEVERED FREQ *
STOP
```

Figure 38. Input deck to program INTFACE for the ST-STS-OMS Kit system

ORIGINAL PAGE IS
OF POOR QUALITY

```
                              INPUT DATA TO ZINTF
                              -------------------

                                NWFILE  =   40
                                NWRKFL  =   10

                                NWRITE  =    1

          NUMBER OF BOOSTER INTERFACE DOFS  =   33          (IFB =   33)

     NUMBER OF TOTAL PAYLOAD MODES RETAINED  =  189          (NPTOT =  189)


                        INPUT DATA FOR THE     2 PAYLOADS
                        ----------------------------------


                           INPUT DATA FOR PAYLOAD     1
                           ----------------------------

                 NUMBER OF PAYLOAD INTERFACE DOFS  =    6

             NUMBER OF PAYLOAD TRUNCATED MODES USED  =  175

             IVEC OF PAYLOAD INTERFACE DOF LOCATIONS IN THE BOOSTER INTERFACE
             ----------------------------------------------------------------


IFACEP (    1 X    6 )    /INPUT/                • VECTOR OF ST INTERFACE DOFS •

    1    1   18   22   21    8    7   31
END OF READIM.
*************************************************************************************


                           INPUT DATA FOR PAYLOAD     2
                           ----------------------------

                 NUMBER OF PAYLOAD INTERFACE DOFS  =    7

             NUMBER OF PAYLOAD TRUNCATED MODES USED  =   14

             IVEC OF PAYLOAD INTERFACE DOF LOCATIONS IN THE BOOSTER INTERFACE
             ----------------------------------------------------------------


IFACEP (    1 X    7 )    /INPUT/                • VECTOR OF OMS KIT INTERFACE DOFS •

    1    1   11   12   14   25   26   28   33
END OF READIM
*************************************************************************************
```

Figure 39. Sample output from subroutine ZINTF
for the ST-STS-OMS Kit system

along with the constants NF = 120, NB = 300 and IF = 33. Figure 40 shows the input deck to program FORCE and its interpretation is listed in Figure 41. This resultant data is written on NFORFL sequentially, resulting in relatively fast and easy use in the response program.

Using the quantities calculated in these previous programs, we then ran the response program. The input deck to program RESPONS is given in Figure 42 . The time interval of interest is t = 0.0 seconds to t = 10.0 seconds. The initial conditions were calculated by solving the response equation at time = 0.0 seconds with the velocities, elastic accelerations and the rigid body displacements equal to zero. Thereby, calculating the rigid body accelerations and elastic displacements. The results of this response run were then used with an acceleration transformation matrix (ATM), provided by NASA, to calculate the discrete accelerations of the space telescope. Figure 43 shows how subroutine ZRESP interpretes this input deck.

```
ZFORCE        TG SHANAHAN
PROGRAM ZFORCE FOR THE SPACE SHUTTLE MODEL   (MSFC MODEL)
PART OF A COMPLETE BOOSTER-PAYLOAD INTEGRATION SOFTWARE PACKAGE
   20   25   40
     0.       10.0        0.005
  120  300   33
    0
TABT3    0  -30STREV
TABF3    0  -30STREV
PHINBR   0  -31ZBOOST
TBR      0  -31ZBOOST
STOP
```

Figure 40. Input deck to program FORCE for the Space Shuttle

```
                    INPUT PARAMETERS
                    ----------------

                 NWRKFL  =   20
                 NFCRFL  =   40

DATA IS WRITTEN ON PAPER EVERY     25 TIME STEPS

                 STARTT  =  0.000000
                   ENDT  = 10.000000
                 DELTAT  =   .005000

NUMBER OF FORCES APPLIED TO BOOSTER  =  120

  NUMBER OF TRUNCATED BOOSTER MODES  =  300

          NUMBER OF INTERFACE DOFS   =   33

INTERPOLATION OF THE FORCE DATA IS NEEDED     (IFTERP = 0)
```

Figure 41. Sample output from subroutine ZFORCE for the Space Shuttle

ORIGINAL PAGE IS
OF POOR QUALITY

```
ZRESP       TG SHANAHAN
PROGRAM ZRESP FOR ST-STS-OMS KIT MODELS
PART OF A COMPLETE BOOSTER-PAYLOAD INTEGRATION SOFTWARE PACKAGE
   40    20     1    30                   * NWFILE,NWRKFL,NWRITE,NFORFL *
    0.         1C.0         0.005         * STARTT,ENDT,DELTAT *
     0 5        0.25                      * GAMMA,BETA *
   120  300    33   189                   * NF,NB,IF,NP *
    0     0     0                         * NDAMPB,NCAMPI,NDAMPP *
FREQBT    0   -31ZBOOST                   * BOOSTER TRUNCATED,CANTILEVERED FREQ *
   0.01                                   * ZETAB *
FREQI     0   -32ZINTF                    * INTERFACE FREQUENCIES *
   0.01                                   * ZETAI *
FREQPA    0   -32ZINTF                    * ASSEMBLED PAYLOAD FREQUENCIES *
   0.01                                   * ZETAP *
B2        0   -32ZINTF                    * INTERFACE/BOOSTER COUPLING MASS MAT *
P2        0   -32ZINTF                    * INTERFACE/PAYLOAD COUPLING STIFFNESS*
PHIIB     0   -32ZINTF                    * INTERFACE MODES *
QNBO      0    33STINTL                   * INITIAL BOOSTER DISPLACEMENTS *
QNBDO     1   300                         * INITIAL BOOSTER VELOCITIES *
0000000000
QIBO      0    33STINTL                   * INITIAL INTERFACE DISPLACEMENTS *
QIBDO     1    33                         * INITIAL INTERFACE VELOCITIES *
0000000000
QNPO      0    33STINTL                   * INITIAL PAYLOAD DISPLACEMENTS *
QNPDO     1   189                         * INITIAL PAYLOAD VELOCITIES *
0000000000
STOP
```

Figure 42. Input deck to program RESPONS
for the ST-STS-OMS Kit system

```
                        INPUT PARAMETERS
                        ----------------

                      NWFILE  =   40
                      NWRKFL  =   20
                      NFORFL  =   30

   DATA IS WRITTEN ON PAPER EVERY      1 TIME STEPS

                      STARTT  =  0.000000
                        ENDT  = 10.000000
                      DELTAT  =    005000

   PARAMETERS FOR NEWMARK-CHAN-BETA INTEGRATION ROUTINE

                       GAMMA  =    500000
                        BETA  -    250000

   VALUE OF BOOSTER MODAL DAMPING IS CONSTANT      (NDAMPB = 0)

   VALUE OF PAYLOAD MODAL DAMPING IS A CONSTANT     (NDAMPP = 0)

   NUMBER OF FORCES APPLIED TO BOOSTER  =  120

     NUMBER OF TRUNCATED BOOSTER MODES  =  300

              NUMBER OF INTERFACE DOFS  =   33

    NUMBER OF TRUNCATED PAYLOAD MODES   =  189
```

Figure 43. Sample output from subroutine ZRESP
for the ST-STS-OMS Kit system

We also implemented the short-cut version which is based on a base drive or open loop technique and was explained in section 5 of Chapter I.

First, a new program must be developed in order to solve Equations (24) through Equations (25-30). This program is called ZSCBRES and is listed in section 6 of Chapter II. Its use is very similar to that of ZRESP and it is believed that enough comment cards are included so that the user should be able to apply it without difficulty. The Input deck is described in Figure 44. Note that we used discrete interface displacements. This Input deck is echoed by program ZSCBRES as shown in Figure 45.

The other change is the creation of a new response routine ZSCPRES. Here we used interface modes $[\Phi_I^B]$ which is not a requirement, but the user should be aware that in the present version these interface modes are used. The Input deck to subroutine ZSCPRES is illustrated in Figure 46 and its interpretation is shown in Figure 47. The results for some test problems are discussed in section 3 of Chapter IV of the Final Report.

```
ZSCBRES         TG SHANAHAN
 SHORT CUT NOMINAL BOOSTER RESPONSE WITHOUT PAYLOADS FOR BOOSTER 1
 OUTPUT TO BE USED IN DIRECT OR COUPLED BASE DRIVE ROUTINE
    40   10   25   30          * NWFILE,NWRKFL,NWRITE,NFORFL *
    0.     0.9      C..        * STARTT,ENDT,DELTAT *
    0.5    0.25                * GAMMA,BETA *
     0    4   54   36          * NDAMPB,NF,NB,IF *
 FREQBT   0  -31ZBOO           * TRUNCATED, CANTILEVERED BOSTER FREQ *
    0.0                        * ZETAB *
 BM1      0  -31ZBOOST         * BOOSTER/INTERFACE COUPLING MASS MAT *
 BM2      0  -31ZBOOST         * BOOSTER MASS REDUCED TO THE INTERFACE *
 BK2      0  -31ZBOOST         * BOOSTER STIFFNESS REDUCED TO INTERFACE *
 QNBO     1   54               * INITIAL BOOSTER NON-IFACE MODAL DISP *
 0000000000
 QNBDO    1   54               * INITIAL BOOSTER NON-IACE MODAL VEL *
 0000000000
 XIBO     1   36               * INITIAL INTERFACE DISCRETE DISP *
 0000000000
 XIBDO    1   36               * INITIAL INTERFACE DEISCRETE VELOCITIES *
 0000000000
 STOP
```

Figure 44. Input deck to program ZSCBRES

ORIGINAL PAGE IS
OF POOR QUALITY

```
                    INPUT PARAMETERS
                    ----------------


                    NWFILE  =   40
                    NWRKFL  =   10
                    NFORFL  =   30

DATA IS WRITTEN ON PAPER EVERY    25 TIME STEPS


                    STARTT  =  0.000000
                      ENDT  =   .900000
                    DELTAT  =   .010000


PARAMETERS FOR NEWMARK-CHAN-BETA INTEGRATION ROUTINE

                     GAMMA  =   .500000
                      BETA  =   .250000

VALUE OF BOOSTER MODAL DAMPING IS CONSTANT     (NDAMPB = 0)


NUMBER OF FORCES APPLIED TO BOOSTER  =    4

  NUMBER OF TRUNCATED BOOSTER MODES  =   54

           NUMBER OF INTERFACE DOFS  =   36
```

Figure 45. Sample output from subroutine ZSCPRES for sample problem 1

```
ZSCPRES      TG SHANAHAN
 PROGRAM SCPRES  - SHORT CUT RESPONSE VARYING BETWEEN A DIRECT BASE
 DRIVE AND A COUPLED BASE DRIVE FOR BOOSTER 1, PAYL 1, PAYL 2 SYSTEM
   40   20   25   30                 * NWFILE,NWRKFL,NWRITE,NRESOFL *
     0.         0.9        0.01      * STARTT, ENDT,DELTAT *
     0.5        0.25                 * GAMMA, BETA *
      1.0                            * EPSILON *
    4   54   36   36                 * NF,NB,IF,NP *
     0    0    0                     * NDAMPB,NDAMPI,NDAMPP *
 FREQBT    0   -31ZBOOST             * TRUNCATED, CANTILEVERED BOOSTER FREQ *
   0.00                              * ZETAB *
 FREQI     0   -32ZINTF              * INTERFACE FREQUENCIES *
   0.00                              * ZETAI *
 FREQPA    0   -32ZINTF              * ASSEMBLED, TRUNC, CANT PAYLOAD FREQ *
   0.00                              * ZETAP *
           0   -32ZINTF              * BOOSTER/INTERFACE COUPLING MASS MAT *
           0   -32ZINTF              * PAYLOAD/INTERFACE COUPLING MASS MAT *
           0   -33ZBOOST             * BOOSTER MASS REDUCED TO INTERFACE *
 BK2       0   -33ZBOOST             * BOOSTER STIFF REDUCED TO INTERFACE *
 BPM2      0   -32ZINTF              * BOOSTER/PAYLOAD COUPLING MASS MATRIX *
 BPK2      0   -32ZINTF              * BOOSTER/PAYLOAD COUPLING STIFFNESS *
 PHIIB     0   -32ZINTF              * INTERFACE MODES *
 QNPO      1    36                   * INITIAL PAYLOAD MODAL DISPLACEMENTS *
 0000000000
 QNPDO     1    36                   * INITIAL PAYLOAD MODAL VELOCITIES *
 0000000000
 STOP
```

Figure 46. Input deck to program ZSCPRES

```
                INPUT PARAMETERS
                ----------------

             NWFILE  =  40
             NWRKFL  =  20
            NRESOFL  =  30

DATA IS WRITTEN ON PAPER EVERY    25 TIME STEPS

             STARTT  =  0.000000
               ENDT  =   .900000
             DELTAT  =   .010000

PARAMETERS FOR NEWMARK-CHAN-BETA INTEGRATION ROUTINE

              GAMMA  =   .500000
               BETA  =   .250000

VALUE OF BOOSTER MODAL DAMPING IS CONSTANT     (NDAMPB = 1)

VALUE OF PAYLOAD MODAL DAMPING IS A CONSTANT     (NDAMPP = 1)

NUMBER OF FORCES APPLIED TO BOOSTER  =   4

  NUMBER OF TRUNCATED BOOSTER MODES  =  54

          NUMBER OF INTERFACE DOFS   =  36

  NUMBER OF TRUNCATED PAYLOAD MODES  =  36
```

Figure 47. Sample output from subroutine ZSCIRES for sample problem 1

5. REFERENCES

1 Engels, R. C., "Structural Dynamics Payload Loads Estimates," Methodology Assessment Report, MMC, MCR-80-553, August 1980.

2 Chen, J. C., Garba, J. A., Salama, M., and Trubert, M., "A Survey of Load Methodology for Shuttle Orbiter Payloads," AIAA/ASME/ASCE/AMS 22nd Structures, Structural Dynamics & Materials Conference, April 6-8, 1981, Atlanta, Georgia. Paper No. 81-0570-CP.

3 Engels, R. C., and Harcrow, H. W., "A New Payload Integration Method," AIAA/ASME/ASCE/AMS 22nd Structures, Structural Dynamics & Materials Conference, April 6-8, 1981, Atlanta, Georgia. Paper No. 81-0571-CP.

4 Craig, R. R., and Bampton, M. C., "Coupling of Structures for Dynamic Analysis," AIAA Journal, Vol. 6, No. 7, July 1968, pp. 1313-1319.

5 Hruda, R. F., and Jones, P. J., "Load Transformation Development Consistent with Modal Synthesis," Shock and Vibration Bulletin, No. 48, September 1978.

6. LISTINGS

First, we list the 6 major programs :

BOOSTER
PAYLOAD
INTFACE
RESPONS
FORCE
LOADS

Then, we list the short-cut routines :

ZSCBRES
ZSCPRES

Finally, we also list the special purpose subroutines,

ZABDI
ZMULTCD
ZMULTDD
ZTERP
ZTERP1
ZTOSEQ2
ZTOSEQ3

```
      PROGRAM BOOSTER (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +                 TAPE20,TAPE30,TAPE31,TAPE32,TAPE40)
C
C
    1 CALL START
                                               CALL TIMCHK (6HTBEGIN)
                                               CALL TIMCHK (6HZBOOST)
      CALL ZBOOST
                                               CALL TIMCHK (6HZBOOST)
                                               CALL TIMCHK (6HTPRINT)
      GO TO 1
      END
C
C
C
C
C
C
C     **************
C     *            *
C     *   ZBOOST   *
C     *            *
C     **************
C
C
C
      SUBROUTINE  ZBOOST
C
C
C    SUBROUTINE ZBOOST PRODUCES ALL BOOSTER RELATED INPUT DATA
C    NECESSARY TO RUN PROGRAM INTFACE.
C
C    DEVELOPED BY RC ENGELS AND TG SHANAHAN , FEBRUARY 1981.
C
C
C    COMMENTS
C    --------
C
C    1.  SUBROUTINE ZBOOST IS PART OF A COMPLETE BOOSTER/PAYLOAD
C        INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C        INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE CAN
C        BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS ESTIMATES
C        FINAL REPORT, JUNE 1982".  ALSO, A USER GUIDE IS CONTIAINED
C        IN "STRUCTURAL DYNAMICS PAYLOAD LOADS ESTIMATES - USER
C        GUIDE, JUNE 1982".
C
C    2.  THE FOLLOWING DATA IS PUT ON NWFI .
C        -BK2         = TBT*BK*TB.  BOOSTER STIFFNESS MATRIX REDUCED TO  HE
C                       INTERFACE.  IF THE INTERFACE IS
C                       DETERMINATE THEN BK2=0 .   SIZE (IF,IF).
C        -BM1         = TBT*BM*IB*PHINB.  BOOSTER COUPLING MASS MATRIX
C                       BETWEEN INTERFACE AND NON-INTERFACE DOFS.
C                       SIZE (IF,NB).
C        -BM2         = TBT*BM*TB.  BOOSTER MASS MATRIX REDUCED TO THE
C                       INTERFACE.  SIZE (IF,IF).
C        -FREQBT      = TRUNCATED CANTILEVERED BOOSTER FREQUENCY  ECTOR
C                       SIZE (1,NB).
C        -PHINBR      = THE INTERFACE EXPANDED SET OF TRUNCATED
C                       CANTILEVERED BOOSTER MODES WHERE ROWS
C                       CORRESPONDING TO ZERO APPLIED FORCES ARE DELETED.
C                       SIZE (NF,NB).
C        -TBR         = THE CONSTRAINT MODAL MATRIX WHERE ROWS
```

```
C                    ROWS CORRESPONDING TO ZERO APPLIED FORCES ARE
C                    DELETED.  SIZE (NF,IF).
C
C    3.  SUBROUTINE ZBOOST CALLS THE FOLLOWING FORMA SUBROUTINES:
C        - PAGEHD, READIM, TIMCHK, ZATXB, ZDISA, ZREAD, ZREDX, ZSAVEL,
C          ZSAVEW, ZSIZE, ZSLADR, ZWRITE, ZWRKFL, ZZBOMB, ZZERO
C
C    SUBROUTINE ARGUMENTS
C    --------------------
C
C       SUBROUTINE ZBOOST HAS NO SUBROUTINE ARGUMENTS.
C
C
C
C    EXAMPLE OF CALLING PROGRAM ON CDC 172/720/730
C    ---------------------------------------------
C
C       FILE ASSUMPTIONS :
C         -TAPE1      = WORK FILE REQUIRED BY ZBOOST.  NWRKFL=1
C         -TAPE10     = FORMA FILE (FOR INPUT DATA).  NOTE THAT MORE
C                       TNAN ONE FORMA FILE MAY BE NECESSARY IF INPUT
C                       DATA IS RECORDED ON SEVERAL FORMA FILES.
C         -TAPE 11    = FORMA FILE (FOR OUTPUT DATA).  NWFILE=11
C
C
C
C           PROGRAM BOOSTER (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C          *                 TAPE1,TAPE10,TAPE11)
C         1 CALL START
C                           CALL TIMCHK (6HTBEGIN)
C                           CALL TIMCHK (6HZBOOST)
C           CALL ZBOOST
C                           CALL TIMCHK (6HZBOOST)
C                           CALL TIMCHK (6HTPRINT)
C           GO TO 1
C           END
C
C
C
C
C    INPUT FORM
C    ----------
C      CALLING PROGRAM MUST -CALL START-
C
C           READ (4I5)  IF,NB,ND,NF
C           READ (3I5)  NWFILE,NWRKFL,NWRITE
C           READ (4I5)  NEXP,NTB,NBMK12,NDET
C           CALL ZREAD (PHINB)
C           IF (NEXP .LQ. 1) GO TO 1
C           CALL READIM (IFACE,NR1,NC1,1,K1)
C         1 CONTINUE
C           CALL READIM (NFORCE,NR2,NC2,1,K2)
C           IF (NTB .EQ. 1) GO TO 3
C           IF (NEXP .EQ. 0) GO TO 2
C           CALL READIM (IFACE,NR1,NC1,1,K1)
C         2 CONTINUE
C           CALL ZREAD (BM)
C           CALL ZREAD (BK)
C           GO TO 5
C         3 CONTINUE
C           CALL ZREAD (TB)
C           IF (NBMK12 .EQ. 1) GO TO 4
```

```
C           CALL ZREAD (BM)
C           CALL ZREAD (BK)
C           GO TO 5
C         4 CONTINUE
C           CALL ZREAD (BM1)
C           CALL ZREAD (BM2)
C           IF (NDET .EQ. 1) GO TO 5
C           CALL ZREAD (BK2)
C         5 CONTINUE
C           CALL ZREAD (FREQB)
C           RETURN
C
C
C     THEREFORE THE FOLLOWING CASES OF INPUT ARE ALLOWED :
C
C       IF,NB,ND,NF                       (ALWAYS INPUT)
C       NWFILE,NWRKFL,NWRITE              (ALWAYS INPUT)
C       NEXP,NTB,NBMK12,NDET              (ALWAYS INPUT)
C       PHINB                             (ALWAYS INPUT)
C       IFACE                             (IS INPUT WHEN NEXP=0 OR NTB=0)
C       NFORCE                            (ALWAYS INPUT)
C       BM,BK                             (ARE INPUT WHEN NTB=0 OR NPMK12=0)
C       TB                                (IS INPUT WHEN NTB=1)
C       BM1,BM2                           (ARE INPUT ONLY WHEN NBMK12=1)
C       BK2                               (IS INPUT WHEN NBMK12=1 AND NDET=0)
C       FREQB                             (ALWAYS INPUT)
C
C         NOTE THAT THE ORDER OF APPEARANCE IN THE INPUT FILE OF THE
C     ABOVE QUANTITIES IS IMPORTANT.  ALSO, NOTE THAT THE CASE NTB=0
C     AND NBMK12=1 IS CONSIDERED IMPOSSIBLE I.E. IF TB IS NOT GIVEN IT IS
C     ASSUMED THAT BM2, BK2, AND BM1 ARE NOT GIVEN EITHER AND MUST BE
C     CALCULATED (I.E. IF NTB=0 THEN NECESSARILY NBMK12=0)
C
C
C
C
C
C     DEFINITION OF INPUT VARIABLES
C     -----------------------------
C       BK       = DISCRETE FREE BOOSTER STIFFNESS MATRIX IN PARTITION-
C                  LOGIC.  SIZE (ND,ND).
C       BK2      = TBT*BK*TB.  BOOSTER STIFFNESS MATRIX REDUCED TO THE
C                  INTERFACE IN PARTITION-LOGIC.  WHEN THE
C                  INTERFACE IS DETERMINATE, BK2=0.  SIZE (IF,IF).
C       BM       = DISCRETE FREE BOOSTER MASS MATRIX IN PARTITION-LOGIC.
C                  SIZE (ND,ND).
C       BM1      = TBT*BM*IB*PHINB.  BOOSTER COUPLING MASS MATRIX BETWEEN
C                  INTERFACE AND NON-INTERFACE DOFS IN PARTITION-LOGIC.
C                  SIZE (IF,NB).
C       BM2      = TBT*BM*TB.  BOOSTER MASS MATRIX REDUCED TO THE INTERFACE
C                  IN PARTITION-LOGIC.  SIZE (IF,IF).
C       IF       = NUMBER OF INTERFACE DOFS
C       IFACE    = VECTOR OF INTEGERS TO INDICATE INTERFACE LOCATIONS IN
C                  DENSE-LOGIC. SIZE (1,IF).  IFACE(I)=N.  N= LOCATION OF
C                  I-TH INTERFACE DOF IN DISCRETE DISPLACEMENT VECTOR OF
C                  BOOSTER.
C       NB       = NUMBER OF TRUNCATED BOOSTER MODES.  NUMBER OF COLUMNS I
C                  FREQBT OF NUMBER OF ROWS IN PHINBR.
C       NBMK12   = 0  BM2, BK2, AND BM1 ARE NOT AVAILABLE.
C                     (BK2 IS NECESSARY ONLY IF NDET=0)
C                = 1  BM2, BK2, AND BM1 ARE AVAILABLE.
C       ND       = NUMBER OF ROWS AND COLUMNS IN BM AND BK (INCLUDING
```

```
C               INTERFACE DOFS).
C     NDET     = 0  THE INTERFACE IS NOT DETERMINATE.
C              = 1  THE INTERFACE IS DETERMINATE.
C     NEXP     = 0  THE CANTILEVERED BOOSTER MODES ARE AVAILABLE BUT
C                   ARE NOT EXPANDED TO INCLUDE THE INTERFACE DOFS.
C              = 1  THE INTERFACE EXPANDED CANTILEVER BOOSTER MODES ARE
C                   AVAILABLE.
C     NF       = NUMBER OF NON-ZERO APPLIED DISCRETE BOOSTER FORCE
C                COMPONENTS.  NUMBER OF COLUMNS IN PHINBR AND TBR.
C     NFORCE   = VECTOR OF INTEGERS IN DENSE-LOGIC TO INDICATE DOFS IN
C                THE BOOSTER WHERE NON-ZERO FORCES ARE APPLIED.
C                NFORCE(I) = DOF WHERE THE I-TH FORCE COMPONENT IN
C                TABLE OF FORCES IS APPLIED.  THE TABLE OF FORCES
C                SHOULD BE ORDERED FROM LOWEST DOF TO THE HIGHEST
C                DOF CORRESPONDING TO NON-ZERO FORCE COMPONENTS, SO
C                THE NFORCE VECTOR SHOULD ASLO BE INCREASING.
C                SIZE (1,NF).
C     NTB      = 0  TB IS NOT AVAILABLE
C              = 1  TB IS AVAILABLE
C     NWFILE   = LOGICAL FILE NUMBER FOR OUTPUT DATA. (E.G. NTAPE=11)
C     NWRITE   = 0  RESULTS ARE NOT PRINTED ON PAPER.
C              = 1  RESULTS ARE PRINTED ON PAPER.
C     NWRKFL   = LOGICAL FILE NUMBER FOR WORKFILE. (E.G. NWRKFL=1)
C     FREQB    = CANTILEVERED BOOSTER FREQUENCY VECTOR IN
C                PARTITION-LOGIC.  SIZE (1,NT) WITH NT .GE. NB.
C     PHINB    = CANTILEVERED BOOSTER MODES IN PARTITION-LOGIC.
C                IF NEXP=0 : INTERFACE DOFS ARE NOT INCLUDED.
C                            SIZE (ND-IF,NT).
C                IF NEXP=1 : INTERFACE DOFS ARE INCLUDED, I.E. ZERO ROWS
C                WHERE INTERFACE DOFS OCCUR.  SIZE (ND,NT).
C     TB       = THE CONSTRAINT MODAL MATRIX IN PARTITION-LOGIC.
C                SIZE (ND,IF).
C
C
C     NERROR EXPLANATIONS
C     -------------------
C      1  =  INPUT SIZE OF THE NON-INTERFACE EXPANDED PHINB
C            IS INCORRECT.  (NRPHI .NE. (ND-IF))
C      2  =  INPUT SIZE OF IFACE IS INCORRECT.  (NC1 .NE. IF)
C      3  =  INPUT SIZE OF THE INTERFACE EXPANDED PHINB
C            IS INCORRECT.   (NRPHI .NE. ND)
C      4  =  INPUT SIZE OF NFORCE IS INCORRECT.  (NC2 .NE. NF)
C      5  =  THE NFORCE VECTOR DOES NOT MONOTOMICALLY INCREASE.
C
C    DIMENSION,COMMON,DATA,FORMAT
C    ----------------------------
      DIMENSION IFACE (200), NFORCE (500)
      COMMON   /NITNOT/      NIT,NOT
      DATA     K1,K2         /200,500/
C
 1000 FORMAT (10I5)
 2000 FORMAT(9(/),49X,*INPUT VARIABLES TO ZBOOST*,/,
     *          ,49X,*-------------------------*,///,
     *          ,53X,*NWFILE =*,I5,/,53X,*NWRKFL =*,I5,/,
     *          ,53X,*NWRITE =*,I5,///,
     *          ,50X,*DESCRIPTION OF BOOSTER*,/
     *          ,50X,*----------------------*,///
     *                ,45X,*NUMBER OF BOOSTER DOFS =*,I5,10X,*ND =*,I5,/
     *        35X,*NUMBER OF BOOSTER INTERFACE DOFS =*,I5,10X,*IF =*I5/25X
     *  ,*NUMBER OF TRUNCATED BOOSTER MODES RETAINED =*,I5,10X,*NB =*,I5
     */28X,*NUMBER OF FORCES APPLIED TO THE BOOSTER =*,I5,10X,*NF =*,I5
     *        ///)
```

```
 2001 FORMAT (19X,*THE MODES ARE NOT EXPANDED TO INCLUDE INTERFACE DOFS*
     *         ,10X,*(NEXP = 0)*,/)
 2002 FORMAT (23X,*THE MODES ARE EXPANDED TO INCLUDE INTERFACE DOFS*,10X
     *          ,*(NEXP = 1)*,/)
 2003 FORMAT (36X,*TB, BM1, BM2, AND BK2 ARE NOT GIVEN*,10X,*(NTB = 0)*
     *          ,/)
 2004 FORMAT (30X,*TB - THE CONSTRAINT MODAL MATRIX IS GIVEN*,10X,
     *          *(NTB = 1)*,/)
 2005 FORMAT (36X,*BM1, BM2, AND BK2 ARE NOT AVAILABLE*,10X,
     *          *(NBMK12 = 0)*,/)
 2006 FORMAT (40X,*BM1, BM2, AND BK2 ARE AVAILABLE*,10X,*(NBMK12 = 1)*/)
 2007 FORMAT (39X,*THE INTERFACE IS NOT DETERMINANT*,10X,*(NDET = 0)*,/)
 2008 FORMAT (43X,*THE INTERFACE IS DETERMINANT*,10X,*(NDET = 1)*,/)
C
C
C
C    A.  BEGINNING OF PROGRAM
C    ------------------------
      READ (NIT,1000) IF,NB,ND,NF
      READ (NIT,1000) NWFILE,NWRKFL,NWRITE
      READ (NIT,1000) NEXP,NTB,NBMK12,NDET
C
      CALL PAGEHD
      WRITE (NOT,2000) NWFILE,NWRKFL,NWRITE,ND,ND,IF,IF,NB,NB,NF,NF
      IF (NEXP .NE. 1)   WRITE (NOT,2001)
      IF (NEXP .EQ. 1)   WRITE (NOT,2002)
      IF (NTB .NE. 1)    WRITE (NOT,2003)
      IF (NTB .EQ. 1)    WRITE (NOT,2004)
      IF (NBMK12 .NE. 1 .AND. NTB .EQ. 1) WRITE (NOT,2005)
      IF (NBMK12 .EQ. 1) WRITE (NOT,2006)
      IF (NDET .EQ. 0)   WRITE (NOT,2007)
      IF (NDET .NE. 0)   WRITE (NOT,2008)
C
      CALL ZWRKFL (NWRKFL)
C
C    B.  COMPUTE PHINBR
C    ------------------
                                                CALL TIMCHK (6HPHINBR)
                                                CALL TIMCHK (6HREAD  )
      CALL ZREAD  (PHINB)
                                                CALL TIMCHK (6HREAD  )
      CALL ZSIZE (PHINB,NRPHI,NCPHI)
      NN=ND-IF
C
      IF (NEXP .EQ. 1) GO TO 100
                                                     NERROR=1
      IF (NRPHI .NE. NN)                             GO TO 999
C
C       B1.  TRUNCATE THE CANTILEVERED MODES AND THEN
C            EXPAND THEM TO INCLUDE INTERFACE DOFS
C       ---------------------------------------------
                                                 CALL TIMCHK (6HREAD  )
      CALL READIM (IFACE,NR1,NC1,1,K1)
                                                 CALL TIMCHK (6HREAD  )
                                                      NERROR=2
      IF (NC1 .NE. IF)                                GO TO 999
C
      J=1
      DO 20 I=1,ND
      IF (J .GT. IF) GO TO 10
      IF (IFACE(J) .NE. I) GO TO 10
      NFORCE(I)=0
```

```
      J=J+1
      GO TO 20
   10 NFORCE(I)=I-J+1
   20 CONTINUE
C
      CALL ZZERO (ZZ,ND,NB)
      NN=ND-IF
      CALL ZDISA (PHINB,1,1,NN,NB,TT)
      CALL ZSLADR (1.,TT,NFORCE,ND,ZZ)
      GO TO 120
C
  100 CONTINUE
C
C       B2.  TRUNCATE THE EXPANDED MODES
C       -------------------------------
                                                           NERROR=3
      IF (NRPHI .NE. ND)                                   GO TO 999
      CALL ZDISA (PHINB,1,1,ND,NB,ZZ)
C
  120 CONTINUE
C ****************************************************
      CALL ZSAVEW (ZZ,6HPHINB ,NWFILE)
C ****************************************************
C
C       B3.  DELETE ROWS THAT CORRESPOND TO DEGREES
C            OF FREEDOM WITH ZERO FORCES APPLIED
C       ------------------------------------------
                                                      CALL TIMCHK (6HREAD   )
      CALL READIM (NFORCE,NR2,NC2,1,K2)
                                                      CALL TIMCHK (6HREAD   )
                                                            NERROR=4
      IF (NC2 .NE. NF)                                      GO TO 999
                                                            NERROR=5
      DO 140 KK=2,NC2
      IF (NFORCE(KK) .LE. NFORCE(KK-1))                     GO TO 999
  140 CONTINUE
C
      CALL ZZERO (PHINBR,NF,NB)
      CALL ZSLADR (1.,ZZ,NFORCE,NF,PHINBR)
                                                      CALL TIMCHK (6HPHINBR)
C
C   C.  COMPUTE TBR,BM2,BK2,AND BM1
C   ------------------------------
                                                       CALL TIMCHK (6HTBMK12)
      IF (NTB .EQ. 1) GO TO 280
C
C       C1.  TB IS NOT GIVEN
C       -------------------
                                                       CALL TIMCHK (6HREAD   )
      IF (NEXP .EQ. 0) GO TO 210
      CALL READIM (IFACE,NR1,NC1,1,K1)
                                                              NERROR=2
      IF (NC1 .NE. IF)                                        GO TO 999
  210 CONTINUE
      CALL ZREAD (BM)
      CALL ZREAD (BK)
                                                       CALL TIMCHK (6HREAD   )
C
C             CALCULATE TB, BM2, AND BK2 IN ZREDX
C             ----------------------------------
                                                       CALL TIMCHK (6HZREDX )
      CALL ZREDX (BM,BK,IFACE,IF,TB,BM2,BK2)
```

```
                                                            CALL TIMCHK (6HZREDX )
      IF (NDET .EQ. 0)  GO TO 260
      CALL ZZERO (BK2,IF,IF)
C
  260 CONTINUE
C
C             CALCULATE BM1                          ORIGINAL PAGE IS
C             -------------                          OF POOR QUALITY
      CALL ZATXB (TB,BM,Z)
      CALL ZMULT (Z,ZZ,BM1)
C
      GO TO 500
C
  280 CONTINUE
C
C        C2.  TB IS GIVEN
C        ---------------
                                                           CALL TIMCHK (6HREAD  )
      CALL ZREAD (TB)
                                                           CALL TIMCHK (6HREAD  )
      IF (NRMK12 .EQ. 1) GO TO 380
                                                           CALL TIMCHK (6HREAD  )
      CALL ZREAD (BM)
      CALL ZREAD (BK)
                                                           CALL TIMCHK (6HREAD  )
C
C             CALCULATE BM1, BM2, AND BK2
C             -------------------------
      CALL ZATXB (TB,BM,Z)
      CALL ZMULT (Z,TB,BM2)
      CALL ZMULT (Z,ZZ,BM1)
      IF (NDET .EQ. 1) GO TO 330
      CALL ZATXB (TB,BK,Z)
      CALL ZMULT (Z,TB,BK2)
      GO TO 340
  330 CONTINUE
      CALL ZZERO (BK2,IF,IF)
  340 CONTINUE
C
      GO TO 500
C
  380 CONTINUE
C
C        C3.  READ BM1, BM2, AND BK2
C        -------------------------
                                                          CALL TIMCHK (6HREAD  )
      CALL ZREAD (BM1)
      CALL ZREAD (BM2)
      IF (NDET .EQ. 1) GO TO 400
      CALL ZREAD (BK2)
      GO TO 410
  400 CONTINUE
      CALL ZZERO (BK2,IF,IF)
  410 CONTINUE
                                                          CALL TIMCHK (6HREAD  )
C
  500 CONTINUE
C
C   D.  DELETE ROWS OF TB THAT CORRESPOND TO DOFS
C       WITH ZERO FORCES APPLIED
C   ------------------------------------------
C   ***************************************************************
```

```
      CALL ZSAVEW (TB,6HTB     ,NWFILE)
C    ******************************************************************
      CALL ZZERO (TBR,NF,IF)
      CALL ZSLADR (1.,TB,NFORCE,NF,TBR)
                                                       CALL TIMCHK (6HTBMK12)
C
C    E.  READ IN AND TRUNCATE THE BOOSTER FREQUENCIES
C    ------------------------------------------------
                                                       CALL TIMCHK (6HREAD  )
      CALL ZREAD (FREQB)
                                                       CALL TIMCHK (6HREAD  )
      CALL ZDISA (FREQB,1,1,1,NB,FREQBT)
C
C
C    F.  WRITE OUTPUT ON PAPER
C    -------------------------
                                                       CALL TIMCHK (6HWRITE )
C
      IF (NWRITE .EQ. 0) GO TO 510
C
      CALL ZWRITE (PHINBR,6HPHINBR)
      CALL ZWRITE (FREQBT,6HFREQBT)
      CALL ZWRITE (TBR,6HTBR    )
      CALL ZWRITE (BM1,6HBM1    )
      CALL ZWRITE (BM2,6HBM2    )
      CALL ZWRITE (BK2,6HBK2    )
C
  510 CONTINUE
C
C   G.  WRITE OUTPUT ON NWFILE
C   --------------------------
      CALL ZSAVEW (PHINBR,6HPHINBR,NWFILE)
      CALL ZSAVEW (FREQBT,6HFREQBT,NWFILE)
      CALL ZSAVEW (TBR,6HTBR    ,NWFILE)
      CALL ZSAVEW (BM1,6HBM1    ,NWFILE)
      CALL ZSAVEW (BM2,6HBM2    ,NWFILE)
      CALL ZSAVEW (BK2,6HBK2    ,NWFILE)
C
      CALL ZSAVEL (NWFILE)
                                                     CALL TIMCHK (6HWRITE )
      RETURN
C
  999 CALL ZZBOMB (6HZBOOST,NERROR)
C
      END
```

```
      PROGRAM PAYLOAD (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE20,
     +                 TAPE30,TAPE31,TAPE32,TAPE40)
C
    1 CALL START
                             CALL TIMCHK (6HTBEGIN)
                             CALL TIMCHK (6HZPAYL )
      CALL ZPAYL
                             CALL TIMCHK (6HZPAYL )
                             CALL TIMCHK (6HTPRINT)
      GO TO 1
      END
C
C
C
C    ***********
C    *         *
C    *  ZPAYL  *
C    *         *
C    ***********
C
C
C
      SUBROUTINE ZPAYL
C
C    SUBROUTINE ZPAYL PRODUCES ALL PAYLOAD RELATED INPUT DATA
C    NECESSARY TO RUN PROGRAM RESPONS.
C
C    DEVELOPED BY RC ENGELS AND TG SHANAHAN, NOVEMBER 1981.
C
C
C
C    COMMENTS
C    --------
C    1.  SUBROUTINE ZPAYL IS PART OF A COMPLETE BOOSTER/PAYLOAD
C        INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C        INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE CAN
C        BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS ESTIMATES -
C        FINAL REPORT, JUNE 1982".
C
C    2.  THE FOLLOWING DATA IS PUT ON NWFILE :
C
C        - FREQPT  = TRUNCATED CANTILEVERED PAYLOAD FREQUENCY VECTOR.
C                    (1,NP)
C        - PK2     = TPT*PK*TP.  PAYLOAD STIFFNESS MATRIX REDUCED TO
C                    THE INTERFACE.  (IF,IF).  IF THE INTERFACE IS
C                    DETERMINATE THEN PK2=0.
C        - PL1     = IP*EP*IPT*MP*IP*PHINPT.  (ND,NP)   A LOADS
C                    TRANSFORMATION.  WHERE IP*PHINPT ARE THE TRUNCATED
C                    EXPANDED CANTILEVERED PAYLOAD MODES.
C        - PL2     = IP*EP*IPT*MP*TP.  (ND.IF)  A LOADS TRANSFORMATION
C        - PM1     = TPT*PM*IP*PHINPT.  PAYLOAD COUPLING MASS MATRIX BETWE
C                    INTERFACE AND NON-INTERFACE DOFS.  (IF,NP).
C        - PM2     = TPT*PM*TP.  PAYLOAD MASS MATRIX REDUCED TO THE
C                    INTERFACE.  (IF,IF)
C        - TP      = THE CONSTRAINT MODAL MATRIX.  (ND,IF)
C
C    3.  SUBROUTINE ZPAYL CALLS THE FOLLOWING FORMA SUBROUTINES:
C        - PAGEHD, READIM,TIMCHK, ZATXB, ZDISA, ZINV3, ZMULT, ZREAD,
C          ZREDX, ZSAVEL, ZSAVEW, ZSIZE, ZSLADR, ZTRANS, ZWRITE, ZWRKFL,
C          ZZBOMB, ZZERO
C
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C
C     SUBROUTINE ARGUMENTS
C     --------------------
C
C     SUBROUTINE ZPAYL HAS NO ARGUMENTS
C
C
C
C     EXAMPLE OF CALLING PROGRAM ON CDC 172/720/730
C     ---------------------------------------------
C
C     FILE ASSUMPTIONS :
C
C         - TAPE1    = WORK FILE REQUIRED BY ZPAYL.  NWRKFL=1.
C         - TAPE10   = FORMA FILE (FOR INPUT DATA).  NOTE THAT MORE THAN
C                      ONE FORMA FILE MAY BE NECESSARY IF INPUT DATA IS
C                      RECORCED ON SEVERAL FORMA FILES.
C         - TAPE11   = FORMA FILE (FOR OUTPUT DATA).  NWFILE=11.
C
C
C       PROGRAM PAYLOAD (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C      *                 TAPE1,TAPE10,TAPE11)
C      1 CALL START
C                      CALL TIMCHK (6HTBEGIN)
C                      CALL TIMCHK (6HZPAYL )
C       CALL ZPAYL
C                      CALL TIMCHK (6HZPAYL )
C                      CALL TIMCHK (6HTPRINT)
C       GO TO 1
C       END
C
C
C
C
C     INPUT FORM
C     ----------
C     CALLING PROGRAM MUST -CALL START-
C
C         READ (3I5) IF, NP, ND
C         READ (3I5) NWFILE, NWRKFL, NWRITE
C         READ (6I5) NEXP, NTP, NPMK12, NDET, NLOAD, NEP
C         CALL ZREAD (PHINP)
C         IF (NEXP .EQ. 1)   GO TO 1
C         CALL READIM (IFACE, NR1, NC1, 1, K1)
C       1 CONTINUE
C         IF (NTP .EQ. 1)   GO TO 3
C         IF (NEXP .EQ. 0)   GO TO 2
C         CALL READIM (IFACE,NR1,NC1,1,K1)
C       2 CONTINUE
C         CALL ZREAD (PM)
C         CALL ZREAD (PK)
C         GO TO 5
C       3 CONTINUE
C         CALL ZREAD (TP)
C         IF (NPMK12 .EQ. 1) GO TO 4
C         CALL ZREAD (PM)
C         CALL ZREAD (PK)
C         GO TO 5
C       4 CONTINUE
C         CALL ZREAD (PM1)
C         CALL ZREAD (PM2)
```

```
C        IF (NDET .EQ. 1)   GO TO 5
C        CALL ZREAD (PK2)
C      5 CONTINUE
C        IF (NLOAD .EQ. 1)   GO TO 9
C        IF (NTP .EQ. 0 .OR. NPMK12 .EQ. 0)   GO TO 6
C        CALL ZREAD (PM)
C        IF (NEP .EQ. 1)   GO TO 6
C        CALL ZREAD (PK)
C        GO TO 7
C      6 CONTINUE
C        IF (NEP .EQ. 0)  GO TO 7
C        CALL ZREAD (EP)
C        GO TO 8
C      7 CONTINUE
C        IF (NEXP .EQ. 0 .OR. NTP .EQ. 0)   GO TO 8
C        CALL READIM (IFACE,NR1,NC1,1,K1)
C      8 CONTINUE
C        IF ((NEXP .EQ. 0 .OR. NTP .EQ. 0)
C            .OR. NEP .EQ. 0)             GO TO 10
C        CALL READIM (IFACE,NR1,NC1,1,K1)
C        GO TO 10
C      9 CONTINUE
C        CALL ZREAD (PL1)
C        CALL ZREAD (PL2)
C     10 CONTINUE
C        CALL ZREAD (FREQP)
C        RETURN
C
C
C    THEREFORE THE FOLLOWING CASES OF INPUT ARE ALLOWED :
C
C    IF,NP,ND                            (ALWAYS INPUT)
C    NWFILE,NWRKFL,NWRITE                (ALWAYS INPUT)
C    NEXP,NTP,NPMK12,NDET,NLOAD,NEP      (ALWAYS INPUT)
C    PHINP                               (ALWAYS INPUT)
C    IFACE                               (IS INPUT WHEN NEXP=0 OR NTP=0
C                                                  OR (NEP=0 AND NLOAD=0))
C    PM                                  (IS INPUT WHEN NTP=0 OR NPMK12=0
C                                                  OR NLOAD =0)
C    PK                                  (IS INPUT WHEN NTP=0 OR NPMK12=0
C                                                  OR (NLOAD=1 AND NEP=0))
C    TP                                  (IS INPUT ONLY WHEN NTP=1)
C    PM1,PM2                             (ARE INPUT ONLY WHEN NTP=1 AND
C                                                  NPMK12=1)
C    PK2                                 (IS INPUT ONLY WHEN NTP=1,
C                                                  NPMK12=1 AND NDET=0)
C    PL1                                 (IS INPUT WHEN NLOAD=1)
C    PL2                                 (IS INPUT WHEN NLOAD=1)
C    EP                                  (IS INPUT ONLY WHEN NLOAD=0 AND
C                                                  NEP=1)
C    FREQ                                (ALWAYS INPUT)
C
C        NOTE THAT THE ORDER OF APPEARENCE IN THE INPUT FILE OF THE ABOVE
C    QUANTITIES IS IMPORTANT. ALSO, NOTE THAT THE CASE NTP=0 AND NPMK12=1
C    IS CONSIDERED IMPOSSIBLE I.E. IF TP IS NOT GIVEN IT IS ASSUMED THAT
C    PM2,PK2 AND PM1 ARE NOT GIVEN EITHER AND MUST BE CALCULATED (I.E. IF
C    NTP=0 THEN NECESSARILY NPMK12=0)
C
C
C
C
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C
C     DEFINITION OF INPUT VARIABLES
C     ----------------------------
C     EP         = (IPT*PK*IP)**-1  THE INVERSE OF THE CANTILEVERED
C                  STIFFNESS MATRIX.
C     FREQP      = CANTILEVERED PAYLOAD FREQUENCY VECTOR IN PARTITION-
C                  LOGIC. (1,NT) WITH NT .GE. NP.
C     IF         = NUMBER OF INTERFACE DOFS.
C     IFACE      = VECTOR OF INTEGERS TO INDICATE INTERFACE LOCATIONS
C                  DENSE-LOGIC.  (1,IF).  IFACE(I)=N.  N=LOCATION OF I-TH
C                  INTERFACE DOF.  IN DISCRETE DISPLACEMENT VECTOR OF
C                  PAYLOAD.
C     ND         = NUMBER OF ROWS AND COLUMNS IN PM AND PK (INCLUDING
C                  INTERFACE DOFS).
C     NDET       = 0   THE INTERFACE IS NOT DETERMINATE.
C                = 1   THE INTERFACE IS DETERMINATE
C     NEP        = 0   EP IS NOT AVAILABLE
C                = 1   EP IS AVAILABLE
C     NEXP       = 0   A SET OF CANTILEVERED PAYLOAD MODES IS AVAILABLE BUT
C                      IS NOT EXPANDED TO INTERFACE DOFS.
C                = 1   A SET OF INTERFACE EXPANDED PAYLOAD MODES IS
C                      AVAILABLE
C     NLOAD      = 0   THE LOAD TRANSFORMATIONS, PL1, PL2, ARE NOT AVAILABLE
C                = 1   THE LOAD TRANSFORMATIONS, PL1, PL2, ARE AVAILABLE
C     NP         = NUMBER OF TRUNCATED PAYLOAD MODES.  NUMBER OF COLUMNS IN
C                  FREQP OF NUMBER OF ROWS IN PHINPR.
C     NPMK12     = 0   PM2,PK2 AND PM1 ARE NOT AVAILABLE.
C                = 1   PM2, PK2, AND PM1 ARE AVAILABLE.
C     NTP        = 0   TP IS NOT AVAILABLE
C                = 1   TP IS AVAILABLE
C     NWFILE     = LOGICAL FILE NUMBER FOR OUTPUT DATA.  (E.G. NTAPE=11)
C     NWRITE     = 0   RESULTS ARE NOT PRINTED ON PAPER.
C                = 1   RESULTS ARE PRINTER ON PAPER.
C     NWRKFL     = LOGICAL FILE NUMBER FOR WORKFILE.  (E.G. NWRKFL=1)
C     PHINP      = CANTILEVERED PAYLOAD MODES IN PARTITION-LOGIC.
C                  IF NEWXP=0 : INTERFACE DOFS ARE NOT INCLUDED. (ND-IF, NT)
C                  IF NEXP=1 : INTERFACE DOFS ARE INCLUDED, I.E. ZERO ROWS
C                  WHERE INTERFACE DOFS OCCUR.  (ND,NT)
C     PK         = DISCRETE FREE PAYLOAD STIFFNESS MATRIX IN PARTITION-
C                  LOGIC. (ND,ND)
C     PL1        = IP*EP*IPT*MP*IP*PHINPR (ND,ND) A LOAD TRANSFORMATION
C                  WHERE, IP*PHINPR IS THE TRUNCATED EXPANDED CANTILEVERED
C                  MODES.
C     PL2        = IP*EP*IPT*MP*TP (ND,IF) A LOAD TRANSFORMATION.
C     PM         = DISCRETE FREE PAYLOAD MASS MATRIX IN PARTITION-LOGIC
C                  (ND,ND)
C     TP         = THE CONSTRAINT MODAL MATRIX.  (ND,IF)
C
C
C     NERROR EXPLANTAIONS
C     ------------------
C      1  =  SIZE OF THE INPUT NON-INTERFACE EXPANDED MODES IS
C            INCORRECT.  (NEXP=0 BUT NRPHI .NE. (ND-IF))
C      2  =  INPUT SIZE OF IFACE IS  INCORRECT.  (NC1 .NE. IF)
C      3  =  SIZE OF THE INPUT INTERFACE EXPANDED MODES IS
C            INCORRECT.  (NEXP=1 BUT NRPHI .NE. ND)
C
C
C
C
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C    DIMENSION, COMMON, DATA, FORMAT
C    -------------------------------
      DIMENSION IFACE (200), NVEC (500)
C
      COMMON  /NITNOT/      NIT,NOT
      DATA          K1                /200/
C
 1000 FORMAT (10I5)
 2000 FORMAT(9(/),49X,*INPUT VARIABLES TO ZPAYL*,/,
     *             ,49X,*------------------------*,///,
     *             ,53X,*NWFILE =*,I5,/,53X,*NWRKFL =*,I5,/,
     *             ,53X,*NWRITE =*,I5,///,
     *             ,50X,*DESCRIPTION OF PAYLOAD*,/
     *             ,50X,*----------------------*,///
     *                 ,45X,*NUMBER OF PAYLOAD DOFS =*,I5,10X,*ND =*,I5,/
     *        35X,*NUMBER OF PAYLOAD INTERFACE DOFS =*,I5,10X,*IF =*I5/25X
     *   ,*NUMBER OF TRUNCATED PAYLOAD MODES RETAINED =*,I5,10X,*NP =*,I5
     *          ///)
 2001 FORMAT (19X,*THE MODES ARE NOT EXPANDED TO INCLUDE INTERFACE DOFS*
     *          ,10X,*(NEXP = 0)*,/)
 2002 FORMAT (23X,*THE MODES ARE EXPANDED TO INCLUDE INTERFACE DOFS*,10X
     *             ,*(NEXP = 1)*,/)
 2003 FORMAT (36X,*TP, PM1, PM2, AND PK2 ARE NOT GIVEN*,10X,*NTP = 0)*/)
 2004 FORMAT (30X,*TP - THE CONSTRAINT MODAL MATRIX IS GIVEN*,10X,
     *             *(NTP = 1)*,/)
 2005 FORMAT (36X,*PM1, PM2, AND PK2 ARE NOT AVAILABLE*,10X,
     *             *(NPMK12 = 0)*,/)
 2006 FORMAT (40X,*PM1, PM2, AND PK2 ARE AVAILABLE*,10X,*(NPMK12 = 1)*/)
 2007 FORMAT (39X,*THE INTERFACE IS NOT DETERMINANT*,10X,*(NDET = 0)*,/)
 2008 FORMAT (43X,*THE INTERFACE IS DETERMINANT*,10X,*(NDET = 1)*,/)
 2009 FORMAT (21X,*THE LOAD TRANSFORMATIONS PL1 AND PL2 ARE NOT GIVEN*,
     *          10X,*(NLOAD = 0)*,/)
 2010 FORMAT (25X,*THE LOAD TRANSFORMATIONS PL1 AND PL2 ARE GIVEN*,
     *          10X,*(NLOAD = 1)*,/)
 2011 FORMAT (25X,*THE CANTILEVERED FLEXIBILITY - EP IS NOT GIVEN*,
     *          10X,*(NEP = 0)*,/)
 2012 FORMAT (29X,*THE CANTILEVERED FLEXIBILITY - EP IS GIVEN*,
     *          10X *(NEP = 1)*,/)
C
C
C    BEGINNING OF PROGRAM
C    --------------------
C
      READ (NIT,1000) IF, NP, ND
      READ (NIT,1000) NWFILE, NWRKFL, NWRITE
      READ (NIT,1000) NEXP, NTP, NPMK12, NDET, NLOAD, NEP
C
      CALL PAGEHD
      WRITE (NOT,2000) NWFILE,NWRKFL,NWRITE,ND,ND,IF,IF,NP,NP
      IF (NEXP .NE. 1)   WRITE (NOT,2001)
      IF (NEXP .EQ. 1)   WRITE (NOT,2002)
      IF (NTP .NE. 1)    WRITE (NOT,2003)
      IF (NTP .EQ. 1)    WRITE (NOT,2004)
      IF (NPMK12 .NE. 1 .AND. NTP .EQ. 1) WRITE (NOT,2005)
      IF (NPMK12 .EQ. 1) WRITE (NOT,2006)
      IF (NDET .EQ. 0)   WRITE (NOT,2007)
      IF (NDET .NE. 0)   WRITE (NOT,2008)
      IF (NLOAD .EQ. 1)  GO TO 6
      WRITE (NOT,2009)
      IF (NEP .NE. 1)    WRITE (NOT,2011)
      IF (NEP .EQ. 1)    WRITE (NOT,2012)
      GO TO 8
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
    6 WRITE (NOT,2010)
    8 CONTINUE
C
C
C
C
C
C   A.  COMPUTE PHINPT = EXPANDED TRUNCATED MODES
C   ------------------------------------------
                                                    CALL TIMC''  (6HF   'IPT)
      CALL ZWRKFL (NWRKFL)
                                                    CALL TIMCHK (6HREAD   )
      CALL ZREAD (PHINP)
                                                    CALL TIMCHK (6HREAD   )
      CALL ZSIZE (PHINP,NRPHI,NCPHI)
      NN=ND-IF
      IF (NEXP .EQ. 1) GO TO 100
                                            NERROR=1
      IF (NRPHI .NE. NN)                    GO TO 999
C
C       A1.  TRUNCATE THE CANTILEVERED MODES THEN EXPAND
C               TO INCLUDE THE INTERFACE DOFS
C       ----------------------------------------------
                                                    CALL TIMCHK (6HREAD   )
      CALL READIM (IFA E,NR1,NC1,1,K1)
                                                    CALL TIMCHK (6HREAD   )
                                            NERROR=2
      IF (NC1 .NE. IF)                      GO TO 999
      J=1
      DO 20 I=1,ND
      IF (J .GT. IF) GO TO 10
      IF (IFACE(J) .NE. I) GO TO 10
      NVEC(I) = 0
      J=J+1
      GO TO 20
   10 NVEC(I) = I-J+1
   20 CONTINUE
C
      CALL ZZERO (ZZ,ND,NP)
      CALL ZDISA (PHINP,1,1,NN,NP,TT)
      CALL ZSLADR (1.,TT,NVEC,ND,ZZ)
      GO TO 110
  100 CONTINUE
C
C       A2.  TRUNCATE THE EXPANDED MODES
C       ------------------------------
                                                 NERROR=3
      IF (NRPHI .NE. ND)                         GO TO 999
      CALL ZDISA (PHINP,1,1,ND,NP,ZZ)
C
  110 CONTINUE
C ****************************************************
      CALL ZSAVEW (PHINP,6HPHINP ,NWFILE)
C ****************************************************
                                                    CALL TIMCHK (6HPHINPT)
C
C   B.  COMPUTE TP,PM2,PK2,AND PM1  (IF NECCESARY)
C   ------------------------------------------
                                                    CALL TIMCHK (6HTPMK12)
      IF (NTP .EQ.1) GO TO 280
C
C       B1.  TP IS NOT GIVEN
```

```
C       -------------------
                                                      CALL TIMCHK (6HREAD   )
      IF (NEXP .EQ. 0) GO TO 210
      CALL READIM (IFACE,NR1,NC1,1,K1)
  210 CONTINUE
      CALL ZREAD (PM)
      CALL ZREAD (PK)
                                                      CALL TIMCHK (6HREAD   )
                                                      CALL TIMCHK (6HZREDX )
C
C             CALCULATE TP, PM2, AND PK2 IN ZREDX
C             -------------------- ------------
      CALL ZREDX (PM,PK,IFACE,IF,TP,PM2,PK2)
                                                      CALL TIMCHK (6HZREDX )
C
      IF (NDET .EQ. 0) GO TO 240
      CALL ZZERO (PK2,IF,IF)
  240 CONTINUE
      CALL ZATXB (TP,PM,Z)
      CALL ZMULT (Z,ZZ,PM1)
C
      GO TO 500
C
  280 CONTINUE
C
C       B2.  TP IS GIVEN
C       ----------------
                                                      CALL TIMCHK (6HREAD   )
      CALL ZREAD (TP)
                                                      CALL TIMCHK (6HREAD   )
      IF (NPMK12 .EQ. 1) GO TO 380
                                                      CALL TIMCHK (6H       )
      CALL ZREAD (PM)
      CALL ZREAD (PK)
                                                      CALL TIMCHK (6HREAD   )
C
C             CALCULATE PM1, PM2 AND PK2
C             --------------------------
      CALL ZATXB (TP,PM,Z)
      CALL ZMULT (Z,TP,PM2)
      IF (NDET .EQ. 1) GO TO 330
      CALL ZATXB (TP,PK,Z)
      CALL ZMULT (Z,TP,PK2)
      GO TO 340
  330 CONTINUE
      CALL ZZERO (PK2,IF,IF)
  340 CONTINUE
      CALL ZATXB (TP,PM,Z)
      CALL ZMULT (Z,ZZ,PM1)
C
      GO TO 500
C
C
  380 CONTINUE
                                                      CALL TIMCHK (6HREAD   )
C
C             READ PM1, PM2, AND PK2
C             ----------------------
      CALL ZREAD (PM1)
      CALL ZREAD (PM2)
      IF (NDET .EQ. 1) GO TO 400
      CALL ZREAD (PK2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      GO TO 500
  400 CONTINUE
      CALL ZZERO (PK2,IF,IF)
                                                  CALL TIMCHK (6HREAD   )
  500 CONTINUE
                                                  CALL TIMCHK (6HTPMK12)
C
C
C   C.  COMPUTE THE LOADS TRANSFORMATIONS PL1 AND PL2
C   ---------------------------------------------------
                                                  CALL TIMCHK (6HLOADTR)
      IF (NLOAD .EQ. 1) GO TO 600
                                                  CALL TIMCHK (6HEPEX   )
      IF (NTP .EQ. 0 .OR. NPMK12 .EQ. 0) GO TO 510
                                                  CALL TIMCHK (6HREAD   )
      CALL ZREAD (PM)
                                                  CALL TIMCHK (6HREAD   )
      IF (NEP .EQ. 1) GO TO 510
                                                  CALL TIMCHK (6HREAD   )
      CALL ZREAD (PK)
                                                  CALL TIMCHK (6HREAD   )
      GO TO 520
  510 CONTINUE
      IF (NEP .EQ. 0) GO TO 520
                                                  CALL TIMCHK (6HREAD   )
      CALL ZREAD (EP)
                                                  CALL TIMCHK (6HREAD   )
      GO TO 550
  520 CONTINUE
C
C       C1.  CALCULATE EP = INV(IPT*KP*IP) - THE CANTILERED FLEXIBILITY
C       ----------------------------------------------------------------
      IF (NEXP .EQ. 0 .OR. NTP .EQ. 0)   GO TO 525
      CALL READIM (IFACE,NR1,NC1,1,K1)
                                             NERROR=2
      IF (NC1 .NE. IF)                       GO TO 999
  525 CONTINUE
C
C            FORM THE CANTILEVERED STIFFNESS
C            -------------------------------
      J=1
      K=1
      DO 540 I=1,ND
      IF (IFACE(J) .EQ. I) GO TO 530
      NVEC(K)=I
      K=K+1
      GO TO 540
  530 CONTINUE
      J=J+1
  540 CONTINUE
C
      CALL ZZERO (PKC,NN,ND)
      CALL ZSLADR (1.,PK,NVEC,NN,PKC)
      CALL ZTRANS (PKC,PKCT)
      CALL ZZERO (PKNN,NN,NN)
      CALL ZSLADR (1.,PKCT,NVEC,NN,PKNN)
C
C            INVERT THE CANTILEVERED STIFFNESS TO GET EP
C            -------------------------------------------
      CALL ZINV3 (PKNN,EP,1)
C
  550 CONTINUE
```

```
C
C       C2.  EXPAND EP TO INCLUDE THE INTERFACE
C            ------------------------------------
      IF (NEXP .EQ. 0 .OR. NTP .EQ. 0
     +    .OR. NEP .EQ. 0)                     GO TO 560
      CALL READIM (IFACE,NR1,NC1,1,K1)
                                           NERROR=2
      IF (NC1 .NE. IF)                     GO TO 999
  560 CONTINUE
C
      J=1
      DO 580 I=1,ND
      IF (J .GT. IF) GO TO 570
      IF (IFACE(J) .NE. I) GO TO 570
      NVEC(I)=0
      J=J+1
      GO TO 580
  570 NVEC(I)=I-J+1
  580 CONTINUE
C
      CALL ZZERO (EPC,ND,NN)
      CALL ZSLADR (1.,EP,NVEC,ND,EPC)
      CALL ZTRANS (EPC,EPCT)
      CALL ZZERO (EPEX,ND,ND)
      CALL ZSLADR (1.,EPCT,NVEC,ND,EPEX)
                                                 CALL TIMCHK (6HEPEX   )
      CALL ZMULT (EPEX,PM,EPEXPM)
C
C       C3.  CALCULATE PL1 AND PL2
C            -------------------------
      CALL ZMULT (EPEXPM,ZZ,PL1)
      CALL ZMULT (EPEXPM,TP,PL2)
      GO TO 650
C
  600 CONTINUE
C
C       C4.  READ PL1 AND PL2
C            --------------------
      CALL ZREAD (PL1)
      CALL ZREAD (PL2)
C
  650 CONTINUE
                                                 CALL TIMCHK (6HLOADTR)
C
C       D.  READ IN AND TRUNCATE THE FREQUENCIES
C           ----------------------------------------
                                                 CALL TIMCHK (6HREAD   )
      CALL ZREAD (FREQP)
                                                 CALL TIMCHK (6HREAD   )
      CALL ZDISA (FREQP,1,1,1,NP,FREQPT)
                                                 CALL TIMCHK (6HWRITE  )
C
      IF (NWRITE .EQ. 0) GO TO 700
C
C         E.  WRITE DATA ON PAPER
C             ----------------------
      CALL ZWRITE (FREQPT,6HFREQPT)
      CALL ZWRITE (TP,6HTP    )
      CALL ZWRITE (PM1,6HPM1   )
      CALL ZWRITE (PM2,6HPM2   )
      CALL ZWRITE (PK2,6HPK2   )
      CALL ZWRITE (PL1,6HPL1   )
```

```
      PROGRAM INTFACE (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +                 TAPE10,TAPE30,TAPE31,TAPE32,TAPE40)
C
    1 CALL START
                                          CALL TIMCHK (6HTBEGIN)
                                          CALL TIMCHK (6HZINTF )
      CALL ZINTF
                                          CALL TIMCHK (6HZINTF )
                                          CALL TIMCHK (6HTPRINT)
      GO TO 1
      END
C
C
C
C
C
C
C     *************
C     *           *
C     *   ZINTF   *
C     *           *
C     *************
C
C
C
        SUBROUTINE ZINTF
C
C     SUBROUTINE ZINTF PRODUCES ALL BOOSTER/PAYLOAD INTERFACE RELATED
C     INPUT DATA NECESSARY TO RUN PROGRAM RESPONS.
C
C     DEVELOPED BY RC ENGELS AND TG SHANAHAN, JULY 1981.
C
C     COMMENTS
C     --------
C
C     1.  SUBROUTINE ZINTF IS PART OF A COMPLETE BOOSTER/PAYLOAD
C         INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C         INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE
C         CAN BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS ESTIMATES-
C         FINAL REPORT, MAY 1981".
C
C     2.  THE FOLLOWING DATA IS PUT ON NWFILE :
C                B2  =  THE INTERFACE/BOOSTER MASS COUPLING MATRIX.
C                       SIZE (IFB,NB).
C              BPK2  =  THE INTERFACE COUPLING STIFFNESS MATRIX BETWEEN
C                       THE BOOSTER AND THE PAYLOADS.  SIZE (IFB,IFB).
C              BPM2  =  THE INTERFACE COUPLING MASS MATRIX BETWEEN
C                       THE BOOSTER AND THE PAYLOADS.  SIZE (IFB,IFB).
C            FREQPA  =  THE ASSEMBLED FREQUENCY MATRIX FOR ALL THE
C                       PAYLOADS IN PARTITION-LOGIC.
C                       FREQPA = (F1(PAY1), F2(PAY1), ........, FN(PAY1),
C                                 F1(PAY2), F2(PAY2), ........, FN(PAY2),
C                                     .                ........       .
C                                 F1(PAYM), F2(PAYM), ........, FN(PAYM))
C             FREQI  =  THE INTERFACE FREQUENCY VECTOR.  SIZE (1,IFB)
C                P2  =  THE INTERFACE/PAYLOAD MASS COUPLING MATRIX.
C                       SIZE (IFB,NP)
C             PHIIB  =  THE INTERFACE MODES MATRIX FORMED BY SOLVING THE
C                       EIGENVALUE PROBLEM FOR BPM2 AND BPK2.
C                       SIZE (IFB,IFB).
C
C     3.  SUBROUTINE ZINTF CALL THE FOLLOWING FORMA SUBROUTINES:
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C       - PAGEHD, READIM, TIMCHK, ZASSEM, ZATXB, ZM2A, ZREAD, ZRVAD
C         ZSAVEL, ZSAVEW, ZSIZE, ZWRITE, ZWRKFL, ZZBOMB, ZZERO
C
C   SUBROUTINE ARGUMENTS
C   --------------------
C
C   SUBROUTINE ZINTF HAS NO SUBROUTINE ARGUMENTS.
C
C
C
C   EXAMPLE OF CALLING PROGRAM ON CDC 172/720/730.
C   ----------------------------------------------
C
C   FILE ASSUMPTIONS :
C        -TAPE1     = WORK FILE REQUIRED BY ZINTF.  NWRKFL=1
C        -TAPE10    = FORMA FILE (FOR INPUT DATA).  NOTE THAT MORE
C                     THAN ONE FORMA FILE MAY BE NECESSARY IF INPUT
C                     DATA IS RECORDED ON SEVERAL FORMA FILES.
C        -TAPE11    = FORMA FILE (FOR OUTPUT).  NWFILE=11
C
C
C       PROGRAM INTFACE (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C      +                   TAPE1,TAPE10,TAPE11)
C     1 CALL START
C                                   CALL TIMCHK (6HTBEGIN)
C                                   CALL TIMCHK (6HZINTF )
C       CALL ZINTF
C                                   CALL TIMCHK (6HZINTF )
C                                   CALL TIMCHK (6HTPRINT)
C       GO TO 1
C       END
C
C
C
C
C   INPUT FORM
C   ----------
C   CALLING PROGRAM MUST - CALL START -
C
C       READ   NPAY                          (I5)
C       READ   NWFILE,NWRKFL,NWRITE          (3I5)
C       READ   IFB, NPTOT                    (2I5)
C       CALL ZREAD (BM1)
C       CALL ZREAD (BM2)
C       CALL ZREAD (BK2)
C       DO 1 K=1,NPAY
C       READ   IFP,NP                        (2I5)
C       CALL READIM (IFACEP,1,NR1,1,K1)
C       CALL ZREAD (PM1)
C       CALL ZREAD (PM2)
C       CALL ZREAD (PK2)
C       CALL ZREAD (FREQPT)
C     1 CONTINUE
C       RETURN
C
C   DEFINITION OF INPUT VARIABLES
C   -----------------------------
C
C         BK2  =  TBT*BK*TB.  BOOSTER STIFFNESS MATRIX REDUCED TO THE
C                 INTERFACE IN PARTITION-LOGIC.  SIZE (IFB,IFB).
C                 WHEN THE INTERFACE IS DETERMINATE, BK2=0.
C         BM1  =  BM1=TBT*BM*IB*PHINB.  BOOSTER COUPLING MASS MATRIX
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                    BETWEEN INTERFACE AND NON-INTERFACE DOFS IN PARTITION-
C                    LOGIC.  SIZE (IFB,NB).
C          BM2  =    TBT*BM*TB.  BOOSTER MASS MATRIX REDUCED TO THE
C                    INTERFACE IN PARTITION-LOGIC.  SIZE (IFB,IFB).
C       FREQPT  =    THE TRUNCATED, CANTILEVERED FREQUENCY VECTOR FOR
C                    THE KTH PAYLOAD IN PARTITION-LOGIC.  SIZE (1,NP(K)).
C          IFB  =    NUMBER OF INTERFACE DOFS ON THE BOOSTER SIDE.
C                    (INCLUDING THE SUPERFLUOUS INTERFACE DOFS)
C          IFP  =    NUMBER OF INTERFACE DOFS IN PAYLOAD K.
C       IFACEP  =    VECTOR OF INTEGERS THAT TELL THE LOCATION OF
C                    PAYLOAD K INTERFACE DOFS IN THE BOOSTER INTERFACE.
C                    IFACEP(I) = LOCATION OF PAYLOAD K INTERFACE DOF I
C                    IN INPUT MATRIX BM1.  SIZE(IFP(K)).
C           NP  =    NUMBER OF TRUNCATED PAYLOAD MODES.  NUMBER OF COLUMNS
C                    IN FREQPT, AND NUMBER OF ROWS IF PM1 FOR PAYLOAD K.
C        NPTOT  =    THE TOTAL NUMBER OF PAYLOAD DEGREES OF FREEDOM.
C                    THE SUM OF ALL NP'S FOR PAYLOADS 1 THRU NPAY.
C         NPAY  =    NUMBER OF PAYLOADS.
C       NWFILE  =    LOGICAL FILE NUMBER FOR OUTPUT DATA. (E.G. NTAPE=11)
C       NWRITE  =    0 RESULTS ARE NOT PRINTED ON PAPER.
C                    1 RESULTS ARE PRINTED ON PAPER.
C       NWRKFL  =    LOGICAL FILE NUMBER FOR WORKFILE. (E.G. NWRKFL=1)
C                    THE INTERFACE AND NON-INTERFACE DOFS IN PARTITION-
C                    LOGIC.  SIZE (IFP(K),IFP(K)) FOR THE KTH PAYLOAD.
C          PM1  =    TPT*PM*IP*PHINP.  PAYLOAD COUPLING MASS MATRIX BETWEEN
C                    THE PAYLOAD AND INTERFACE.  SIZE (IFP(K),NP(K)).
C          PM2  =    TPT*PM*TP.  PAYLOAD MASS MATRIX REDUCED TO THE
C                    INTERFACE IN PARTITION-LOGIC.  SIZE (IFP(K),IFP(K)).
C          PK2  =    TPT*PK*TP.  PAYLOAD STIFFNESS MATRIX REDUCED TO THE
C                    INTERFACE IN PARTITION-LOGIC.  WHEN THE INTERFACE
C                    IS DETERMINATE, PK2=0.  SIZE (IFP(K),IFP(K)).
C
C    NERROR EXPLANATION
C    ------------------
C     1  =  NUMBER OF ALLOWABLE PAYLOADS HAS BEEN EXCEEDED.
C           THIS CAN BE CORRECTED BY INCREASING KP AND CHANGING
C           THE DIMENSION STATEMENT ACCORDINGLY.
C     3  =  SIZE OF IFACE FOR PAYLOAD K IS NOT EQUAL TO IFP(K).
C     4  =  ROW SIZE OF PM2 FOR PAYLOAD K IS NOT EQUAL TO IFP(K).
C     5  =  SIZES OF PM1, PM2 AND PK2 ARE NOT CONSISTENT.
C     6  =  SIZES OF PM1 AND FREQPT ARE NOT CONSISTENT.
C     7  =  COLUMN SIZE OF PM1 FOR PAYLOAD K IS NOT EQUAL TO NP(K).
C     8  =  INPUT SIZE OF IFACE OF SUPERFLUOUS INTERFACE DOFS
C           IS NOT EQUAL TO IFS.
C     9  =  ASSEMBLED SIZE OF PM2 AND PK2 IS NOT CORRECT.
C
C    A.  DIMENSION,COMMON,DATA,FORMAT
C    ---- ---------------------------
      COMMON    /NITNOT/  NIT,NOT
      COMMON    /LSTRT4/   NLINE,NLPP
      DIMENSION IVEC(200),IFACEP(200)
C
      DATA      K1, KR          /500,200/
C
 1000 FORMAT (10I5)
 1100 FORMAT (10E10.0)
 2001 FORMAT (/////,50X,*INPUT DATA TO ZINTF*,/,
     +              50X,*-------------------*,
     +       /////,55X,*NWFILE  =*,I5,/,55X,*NWRKFL  =*,I5,//,
     +       55X,*NWRITE  =*,I5,//,
     +       29X,*NUMBER OF BOOSTER INTERFACE DOFS  =*,I5
     +              ,10X,*(IFB =*,I5,*)*,//,
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     +       23X,*NUMBER OF TOTAL PAYLOAD MODES RETAINED  =*
     +       ,I5,10X,*(NPTOT =*,I5,*)*)
 2002 FORMAT (//,40X,*INPUT DATA FOR PAYLOAD*,I5,/
     +              ,40X,*----------------------------*,//
     +       ,28X,*NUMBER OF PAYLOAD INTERFACE DOFS  =*,I5,//,
     +         22X,*NUMBER OF PAYLOAD TRUNCATED MODES USED  =*,I5,//,
     +         22X,*IVEC OF PAYLOAD INTERFACE DOF LOCATIONS IN THE*,
     +             * BOOSTER INTERFACE*,/,
     +         22X,*-----------------------------------------------*,
     +             *-----------------*,//)
 2003 FORMAT (//,22X,*IVEC OF SUPERFLUOUS INTERFACE DOF LOCATIONS*
     +         ,* IN THE BOOSTER*,/,
     +              22X,*------------------------------------------*
     +         ,*----------------*,////)
 2004 FORMAT (///,35X,*INPUT DATA FOR THE *,I5,* PAYLOADS*,/,
     +              ,35X,*----------------------------------*,//)
C
C                    *********************
C
C                    BEGINNING OF PROGRAM
C
C                    *********************
C
C
C    B.  READ SUBROUTINE INPUT PARAMETERS
C    ------------------------------------
      READ (NIT,1000)  NPAY
      READ (NIT,1000)  NWFILE, NWRKFL, NWRITE
      READ (NIT,1000)  IFB, NPTOT
      CALL PAGEHD
      WRITE (NOT,2001) NWFILE,NWRKFL,NWRITE,IFB,IFB,NPTOT,NPTOT
C
      CALL ZWRKFL (NWRKFL)
C
C    C   READ AND AND CHECK THE SIZE OF BOOSTER QUANTITIES
C    -----------------------------------------------------
      CALL ZREAD (BM1)
      CALL ZREAD (BPM2)
      CALL ZREAD (BPK2)
C
      CALL ZSIZE (BM1,IF,NB)
      CALL ZSIZE (BPM2,IF2,IF3)
      CALL ZSIZE (BPK2,IF4,IF5)
                                                          NERROR=1
      IF (IFB .NE. IF)                                    GO TO 999
      IF (IFB .NE. IF2 .OR. IF2 .NE. IF3)                 GO TO 999
      IF (IFB .NE. IF4 .OR. IF4 .NE. IF5)                 GO TO 999
C
C    D.  READ IN AND CHECK THE SIZES OF PAYLOAD QUANTITIES
C        ASSEMBLE FREQPA AND  P; AND COUPLE THE BOOSTER
C        BOOSTER AND PAYLOAD INTERFACE QUANTITIES
C    -----------------------------------------------------
      NPS=0
      CALL ZZERO (FREQPA,1,NPTOT)
      CALL ZZERO (P,IFB,NPTOT)
C
      DO 40 K=1,NPAY
      CALL PAGEHD
      WRITE (NOT,2004)    NPAY
      NLINE=NLINE+15
      READ (NIT,1000)  IFP, NP
      WRITE (NOT,2002)  K,IFP,NP
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL READIM (IFACEP,1,IVSIZE,1,K1)
                                                                 NERROR=2
      IF (IFP .NE. IVSIZE)                                       GO TO 999
      CALL ZREAD (PM1)
      CALL ZREAD (PM2)
      CALL ZREAD (PK2)
      CALL ZREAD (FREQPT)
      CALL ZSIZE (PM2,IF1,IF2)
      CALL ZSIZE (PK2,IF3,IF4)
      CALL ZSIZE (PM1,IF5,NP1)
      CALL ZSIZE (FREQPT,NO1,NP2)
                                                                 NERROR=3
      IF (IF1 .NE. IFP)                                          GO TO 999
                                                                 NERROR=4
      IF (IF1 .NE. IF2 .OR. IF1 .NE. IF3 .OR.
     +    IF1 .NE. IF4 .OR. IF1 .NE. IF5)                        GO TO 999
                                                                 NERROR=5
      IF (NP1 .NE. NP2)                                          GO TO 999
                                                                 NERROR=6
      IF (NP1 .NE. NP)                                           GO TO 999
C
      DO 20 I=1,NP
      IVEC(I)=NPS+I
   20 CONTINUE
C
      CALL ZRVAD (1.0,PM2,IFACEP,IFACEP,IVSIZE,IVSIZE,BPM2)
      CALL ZRVAD (1.0,PK2,IFACEP,IFACEP,IVSIZE,IVSIZE,BPK2)
      CALL ZRVAD (1.0,PM1,IFACEP,IVEC,IVSIZE,NP,P)
      CALL ZASSEM (FREQPT,1,NPS+1,FREQPA)
C
      NPS=NPS+NP
C
   40 CONTINUE
C
                                                                 NERROR=7
      IF (NPS .NE. NPTOT)                                        GO TO 999
C
C   F.  SOLVE THE EIGENVALUE PROBLEM FOR THE INTERFACE
C       MASS, BPM2, AND THE INTERFACE STIFFNESS, BPK2.
C   -----------------------------------------------------
                                                     CALL TIMCHK (6HZM2A   )
      CALL ZM2A (BPM2,BPK2,PHIIB,FREQI,NWRITE,6HPHIIB ,6HFREQI
     +          ,MATUI,MATDI,MAT3I,MAT4I,MAT5I,MAT6I,
     +           IFB,IFB,1.0,20)
                                                     CALL TIMCHK (6HZM2A   )
C
C   G.  CALCULATE B2 AND P2
C   -----------------------
      CALL ZATXB (PHIIB,BM1,B2)
      CALL ZATXB (PHIIB,P,P2)
C
      IF (NWRITE .EQ. 0)   GO TO 180
C
C   H.  WRITE MATRICES ON PAPER
C   ---------------------------
      CALL ZWRITE (BPM2,6HBPM2  )
      CALL ZWRITE (BPK2,6HBPK2  )
      CALL ZWRITE (B2,6HB2    )
      CALL ZWRITE (P2,6HP2    )
      CALL ZWRITE (FREQPA,6HFREQPA)
C
C   I.  WRITE MATRICES ON NWFILE
```

ORIGINAL PAGE IS OF POOR QUALITY

```
C    ----------------------------
  180 CALL ZSAVEW (FREQI,6HFREQI  ,NWFILE)
      CALL ZSAVEW (B2,6HB2     ,NWFILE)
      CALL ZSAVEW (P2,6HP2     ,NWFILE)
      CALL ZSAVEW (BPM2,6HBPM2   ,NWFILE)
      CALL ZSAVEW (BPK2,6HBPK2   ,NWFILE)
      CALL ZSAVEW (PHIIB,6HPHIIB  ,NWFILE)
      CALL ZSAVEW (FREQPA,6HFREQPA,NWFILE)
C
C   J.  WRITE LIST OF MATRICES ON NWFILE
C    ----------------------------------
      CALL ZSAVEL (NWFILE)
C
      RETURN
  999 CALL ZZBOMB (6HZINTF  ,NERROR)
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM FORCE (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +               TAPE20,TAPE30,TAPE31,TAPE32,TAPE40)
C
C
    1 CALL START
                                              CALL TIMCHK (6HTBEGIN)
                                              CALL TIMCHK (6HZFORCE)
      CALL ZFORCE
                                              CALL TIMCHK (6HZFORCE)
                                              CALL TIMCHK (6HTPRINT)
C
      GO TO 1
      END
C
C
C             ************************************
C             *                                  *
C             *            ZFORCE                *
C             *                                  *
C             ************************************
C
      SUBROUTINE ZFORCE
C
C      SUBROUTINE ZFORCE TAKES INPUT TIME AND FOR DATA AND PROCESSES
C      IT TO GET THE FINAL FORCE VECTORS FOR USE IN A NUMERICAL
C      INTEGRATION RESPONSE ROUTINE.
C
C      DEVELOPED BY TG SHANAHAN, FEB. 1982.
C
C      COMMENTS
C      --------
C      1.  SUBROUTINE ZFORCE IS PART OF A COMPLETE BOOSTER/PAYLOAD
C          INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C          INTEGRATION TECHNIQUE.  A DISCUSION OF THIS TECHNIQUE CAN
C          BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS ESTIMATES -
C          FINAL REPORT, MAY 1982".
C      2.  DATA IS OUTPUT ON NFORFL SEQUENTIALLY.
C      3.  NFORFL CONTAINS:
C          A.  A HEADER WITH RUN NO., DATE, START TIME, TIME STEP SIZE,
C              END TIME, NUMBER OF FORCE POINTS,AND THE NUMBER OF TIME
C              POINTS.
C          B.  THE TIME AND THE CORRESPONDING FORCE DATA FOR THAT TIME:
C                  T1,FB1(T1),FB2(T1),...,FBN(T1),FI1(T1), ...,FIF(T1),
C                  T2,FB1(T2),    .    ...    .        .    ...,FIF(T2),
C                  .    .         .    ...    .        .    ...    .
C                  .    .         .    ...    .        .    ...    .
C                  TE,FB1(TE),FB2(TE),...,FBN(TE),FI1(TE),...,FIF(TE)
C
C                  WHERE,  FB1 THRU FBN ARE THE FORCE COMPONENTS ACTING
C                          ON THE BOOSTER.  FB=(PHINBR)T*F.
C                    AND,  FI1 THRU FIF ARE THE FORCE COMPONENTS ACTING
C                          ON THE INTERFACE.  FI=(TBR)T*F.
C      4.  THIS SUBROUTINE USES VARIOUS COMBINATIONS OF SUBROUTINES TO
C          GENERATE NFORFL DEPENDING ON THE SIZE OF THE INPUT SYSTEM.
C          BECAUSE OF THE NATURE OF PARTITION-LOGIC SUBROUTINES A
C          PURE PARTITION LOGIC INTERPOLATION SUBROUTINE SUCH AS ZTERP
C          IS VERY SLOW COMPARED TO A DENSE-LOGIC VERSION.  THEREFORE,
C          SEVERAL SPECIALIZED SUBROUTINES WERE CODED FOR USE WITHIN
C          THIS SUBROUTINE.  THESE ROUTINES ARE ZTERP1 AND ZTOSEQ2.
C          HOWEVER, WHILE THESE SUBROUTINES ARE FASTER THAN THEIR
C          STRAIGHT PARTITION-LOGIC COUNTERPARTS, ZTERP AND ZTOSEQ3,
C          THEY ARE SIZE LIMITED AND THEREFORE CAN NOT BE USED ALL THE
```

```
C          TIME.  THIS LIMITING SIZE IS SET BY THE PARAMETER KCNB
C          WHICH IS THE DIMENSION SIZE OF A MATRIX IN THE COMMON BLOCK
C          /LWORK1/ THAT IS USED IN THESE SUBROUTINES.  THE
C          ORIGINAL VALUE OF KCNB IS 600, HOWEVER, THIS MAY BE
C          INCREASED OR DECREASED AS STORAGE REQUIREMENTS DEMAND
C          BY CHANGING ITS VALUE IN THE CORRECT DATA STATEMENTS
C          AND IN THE COMMON BLOCK DIMENSIONS.
C
C      5.  SUBROUTINE ZFORCE CALLS THE FOLLOWING FORMA SUBROUTINES:
C          - DTOZ, PAGEHD, TIMCHK, WRITE, ZASSEM, ZMULT, ZREAD, ZSIZE,
C            ZTOD, ZTRANS, ZWRKFL, ZZBOMB, ZZERO
C
C      6. IT ALSO CALL THE FOLLOWING SPECIAL PURPOSE SUBROUTINES THAT
C         WERE DEVELOPED ESPECIALLY FOR THIS SOFTWARE PACKAGE:
C         - ZTERP, ZTERP1, ZTOSEQ2, ZTOSEQ3
C
C      INPUT FORM
C      ----------
C       READ    NWRKFL, NWRITE, NFORFL                (3I5)
C       READ    STARTT, ENDT, DELTAT                  (3E10.0)
C       READ    NF,NB,IF                              (3I5)
C       READ    IFTERP                                (I5)
C       CALL ZREAD (TIME)
C       CALL ZREAD (FORCE)
C       CALL ZREAD (PHINBR)
C       CALL ZREAD (TBR)
C
C .    DEFINITION OF INPUT VARIABLES
C
C      -----------------------------
C         DELTAT  =  TIME STEP BETWEEN OUTPUT INTERPOLATED TIME POINTS.
C           ENDT  =  END TIME FOR INTERPOLATION TIME RANGE.
C          FORCE  =  INPUT MATRIX OF FORCE VECTORS IN PARTITION-LOGIC.
C                    SIZE (NTP,NF).
C             IF  =  NUMBER OF BOOSTER INTERFACE POINTS.
C         IFTERP  =  0  IF THE FORCE DATA HAS NOT BEEN PREVIOUSLY
C                       INTERPOLATED.
C                 =  1  IF THE FORCE DATA HAS BEEN INTERPOLATED.
C             NB  =  NUMBER OF TRUNCATED BOOSTER MODES RETAINED.
C                    NUMBER OF ROWS IN PHINBR.
C             NF  =  NUMBER OF FORCE POINTS ON THE BOOSTER.
C                    NUMBER OF COLUMNS IN FORCE.
C         NFORFL  =  LOGICAL UNIT NUMBER FOR OUTPUT. CONTAINS
C                    INTERPOLATED DATA SEQUENTIALLY.
C            NTP  =  NUMBER OF TIME POINTS IN FORCE TABLE.
C                    NUMBER OF ROWS IN FORCE.
C         NWRITE  =  0  RESULTS ARE NOT PRINTED ON PAPER.
C                    N  RESULTS ARE PRINTED EVERY N TIME STEPS.
C         NWRKFL  =  LOGICAL UNIT NUMBER OF A WORK FILE REQUIRED
C                    BY ZFORCE.
C         PHINBR  =  THE TRUNCATED EXPANDED CANTILEVERED BOOSTER MODES
C                    MATRIX IN PARTITION-LOGIC WITH ROWS CORRESPONDING
C                    TO ZERO APPLIED FORCES HAVE BEEN DELETED.
C                    SIZE (NB,NF).
C           TIME  =  INPUT VECTOR OF TIME DATA IN PARTITION LOGIC.
C                    CONTAINS DATA CORRESPONDING TO THE ROWS IN THE
C                    FORCE MATRIX.  SIZE (NTP,1).
C            TBR  =  THE BOOSTER CONSTRAINT MODAL MATRIX IN PARTITION-
C                    LOGIC WHERE ROWS CORRESPONDING TO ZERO APPLIED
C                    FORCES HAVE BEEN DELETED.  SIZE (NF,IF).
C
C   A.  COMMONS, DIMENSIONS, AND FORMATS
C   ------------------------------------
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      COMMON    /LSTRT4/   NLINE, NLPP
      COMMON    /NITNOT/   NIT,NOT
      COMMON    /LWORK1/   F(6000),FILLER(13200)
C
C
      DATA       KCNB       /600/
      DATA       KF         /6000/
      DATA       BUF        /-1.E50/
C
 1000 FORMAT (10I5)
 1100 FORMAT (10E10.0)
 1200 FORMAT (//,4X,*TIME*,55X,*FORCE POINTS*/
     +            ,4X,*----*,55X,*----- ------*)
 1201 FORMAT (20X,10(*(*I2*)*,7X)/)
 1202 FORMAT (//1X,1PE11.4,4X,10(1PE11.3))
 1203 FORMAT (16X,10(1PE11.3))
 1204 FORMAT (//,48X,*STARTT  =*,F10.6,/,50X,*ENDT  =*,F10.6,/,
     +            48X,*DELTAT  =*,F10.6)
 1205 FORMAT (/////,50X,*INPUT PARAMETERS*,/,50X,16(1H-),///,
     +          48X,*NWRKFL  =*,I5,/,
     +          48X,*NFORFL  =*,I5,//,
     +          30X,*DATA IS WRITTEN ON PAPER EVERY *,I5,* TIME STEPS*)
 1206 FORMAT (//30X,*NUMBER OF FORCES APPLIED TO BOOSTER  =*,I5,//,
     +            32X,*NUMBER OF TRUNCATED BOOSTER MODES  =*,I5,//,
     +            41X,*NUMBER OF INTERFACE DOFS  =*,I5)
 1207 FORMAT (//,27X,*INTERPOLATION OF THE FORCE DATA IS NEEDED*,5X,
     +              *(IFTERP = 0)*)
 1208 FORMAT (//,25X,*INTERPOLATION OF THE FORCE DATA IS NOT NEEDED*,
     +            ,5X,*(IFTERP = 1)*)
 2000 FORMAT (10(/),40X,*FORCE - TIME HISTORY*,/,40X,20(1H-),//,
     +                40X,*STARTT =*,1PE13.3,//,42X,*ENDT =*,1PE13.3,//,
     +                40X,*DELTAT =*,1PE13.3,//,
     +                14X,*NUMBER OF FORCE POINTS (NB + IF) =*,I6,//,
     +                26X,*NUMBER OF TIME STEPS =*,I6)
 2001 FORMAT (4(/),40X,*TIME =*,1PE13.3,/,40X,19(1H-),/)
C
C       ***************************
C
C            BEGINING OF PROGRAM
C
C       ***************************
C
C    B.  INPUT VARIABLES
C    -------------------
      READ (NIT,1000) NWRKFL,NWRITE,NFORFL
      READ (NIT,1100) STARTT,ENDT,DELTAT
      READ (NIT,1000) NF,NB,IF
      READ (NIT,1000) IFTERP
C
      CALL PAGEHD
      WRITE (NOT,1205)    NWRKFL,NFORFL,NWRITE
      WRITE (NOT,1204)    STARTT,ENDT,DELTAT
      WRITE (NOT,1206)    NF,NB,IF
      IF (IFTERP .EQ. 0)   WRITE (NOT,1207)
      IF (IFTERP .EQ. 1)   WRITE (NOT,1208)
C
      CALL ZWRKFL (NWRKFL)
C
C    C.  READ TIME AND FORCE DATA
C    ----------------------------
      CALL ZREAD (TIME)
      CALL ZREAD (FORCE)
```

```
      CALL ZREAD (PHINBR)
      CALL ZREAD (TBR)
C
C   D.  PERFORM ERROR CHECKS AND FORM THE FORCE MULTIPLICATION MATRIX
C   -----------------------------------------------------------------
                                                  CALL TIMCHK (6HASSEM )
      CALL ZSIZE (FORCE,NR1,NC1)
      CALL ZSIZE (PHINBR,NR2,NC2)
      CALL ZSIZE (TBR,NR3,NC3)
                                                              NERROR=1
      IF (NF .NE. NC1 )                                      GO TO 999
                                                              NERROR=2
      IF (NR2 .NE. NR3 .OR. NR2 .NE. NF)                     GO TO 999
                                                              NERROR=3
      IF (NC2 .NE. NB)                                       GO TO 999
                                                              NERROR=4
      IF (NC3 .NE.IF)                                        GO TO 999
C
      IF (IFTERP .NE. 1)   GO TO 30
                                                              NERROR=5
      CALL ZTOD (TIME,F,NT,1,KF,1)
      EPS=DELTAT/1.E08
      DO 20 I=1,NT
      IF ((STARTT+FLOAT(I-1)*DELTAT-F(I)) .GT. EPS)          GO TO 999
   20 CONTINUE
C
   30 NCTOT=NB+IF
      CALL ZZERO (PHITB,NR2,NCTOT)
      CALL ZASSEM (PHINBR,1,1,PHITB)
      CALL ZASSEM (TBR,1,NB+1,PHITB)
                                                  CALL TIMCHK (6HASSEM )
C
      IF (IFTERP .EQ. 0 .AND. NCTOT .GT. KCNB)   GO TO 50
C
                                                  CALL TIMCHK (6HF*PHTB)
      CALL ZMULT (FORCE,PHITB,TABF)
                                                  CALL TIMCHK (6HF*PHTB)
C
      IF (IFTERP .EQ. 1)    GO TO 100
C
C   E.  PERFORM INTERPOLATION USING SUBROUTINE ZTERP1
C   ------------------------------------------------
                                                  CALL TIMCHK (6HZTERP1)
      CALL ZTERP1 (TIME,TABF,STARTT,ENDT,DELTAT,NFORFL)
                                                  CALL TIMCHK (6HZTERP1)
C
      GO TO 200
C
C   F.  PERFORM INTERPOLATION USING SUBROUTINE ZTERP
C   -----------------------------------------------
   50 CONTINUE
                                                  CALL TIMCHK (6HPRETER)
      NTP=(ENDT-STARTT)/DELTAT+1.1
C
      DO 70 I=1,NTP
      F(I)=STARTT+FLOAT(I-1)*DELTAT
   70 CONTINUE
C
      CALL DTOZ (F,T2,1,NTP,1,KF)
C
      CALL ZTRANS (TIME,T1)
      CALL ZTRANS (FORCE,F1)
```

```
      CALL DTOZ (F,TIME,NTP,1,KF,1)
                                                            CALL TIMCHK (6HPRETER)
C
                                                            CALL TIMCHK (6HZTERP )
      CALL ZTERP (T1,T2,F1,F2)
                                                            CALL TIMCHK (6HZTERP )
C
                                                            CALL TIMCHK (6HT-MULT)
      CALL ZTRANS (F2,F2T)
      CALL ZMULT (F2T,PHITB,TABF)
                                                            CALL TIMCHK (6HT-MULT)
C
C   G. WRITE INTERPOLATION RESULTS ON NFORFL
C   ---------------------------------------
  100 IF (NCTOT .GT. KCNB)   GO TO 120
                                                            CALL TIMCHK (6HZTOS2 )
      CALL ZTOSEQ2 (TIME,TABF,NFORFL)
                                                            CALL TIMCHK (6HZTOS2 )
      GO TO 200
C
  120 CONTINUE
                                                            CALL TIMCHK (6HZTOS3 )
      CALL ZTOSEQ3 (TIME,TABF,NFORFL)
                                                            CALL TIMCHK (6HZTOS3 )
C
C
  200 IF (NWRITE .EQ. 0)    RETURN
C
C   H.  WRITE INTERPOLATED RESULTS ON PAPER
C   ---------------------------------------
                                                            CALL TIMCHK (6HWRITE )
      REWIND NFORFL
      NW=NWRITE
C
      READ (NFORFL) IRUNNO,IDATE,STARTT,ENDT,DELTAT,NBIF,NTP,
     +              (BUF,I=1,10)
C
      CALL PAGEHD
      WRITE (NOT,2000)  STARTT,ENDT,DELTAT,NBIF,NTP
      CALL PAGEHD
C
      DO 300 J=1,NTP
      READ (NFORFL) T,(F(I),I=1,NBIF)
      IF (NW .LT. NWRITE .AND. J .NE. NTP)  GO TO 290
      NLINE=NLINE+7
      IF (NLINE .LE. NLPP)  GO TO 280
      CALL PAGEHD
      NLINE=NLINE+7
  280 WRITE (NOT,2001) T
      CALL WRITE (F,1,NBIF,6HFORCES,1,KF)
      NW=0
  290 NW=NW+1
  300 CONTINUE
                                                            CALL TIMCHK (6HWRITE )
C
C   I. EXIT
C   -------
      RETURN
C
  999 CALL ZZBOMB (6HZFORCE,NERROR)
      END
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      PROGRAM RESPONS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +                 TAPE20,TAPE30,TAPE31,TAPE32,TAPE33,TAPE40)
C
C
    1 CALL START
                                                  CALL TIMCHK (6HTBEGIN)
                                                  CALL TIMCHK (6HZRESP )
      CALL ZRESP
                                                  CALL TIMCHK (6HZRESP )
                                                  CALL TIMCHK (6HTPRINT)
      GO TO 1
      END
C
C
C
C
C        ********************************
C        *                              *
C        *            ZRESP             *
C        *                              *
C        ********************************
C
C
C
      SUBROUTINE ZRESP
C
C
C    SUBROUTINE ZRESP PRODUCES THE RESPONSE OF THE COUPLED
C    BOOSTER/PAYLOAD SYSTEM.
C
C    DEVELOPED BY RC ENGELS AND TG SHANAHAN, JANUARY 1982.
C
C
C    COMMENTS
C    --------
C    1.  SUBROUTINE ZRESP IS PART OF A COMPLETE BOOSTER/PAYLOAD
C        INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C        INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE CAN
C        CAN BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS
C        ESTIMATES - FINAL REPORT, MAY 1982".
C
C    2.  SUBROUTINE ZRESP USES A MODIFIED FORMA OF THE NEWMARK-
C        CHAN-BETA INTEGRATION SCHEME.
C
C    3.  THE FOLLOWING DATA IS PUT ON NWFILE:
C        A.  A HEADER CONTAINING - RUNNO, DATE, STARTT, ENDT, DELTAT, IF
C                                          AND NP
C        B.  THE RESPONSE DATA :
C                 T1, (QIB(I),I=1,IF),    (QNP(I),I=1,NP),
C                     (QIBD(I),I=1,IF),   (QNPD(I),I=1,NP),
C                     (QIBDD(I),I=1,IF),  (QNPDD(I),I=1,NP),
C
C            ALL OF THESE PROPERTIES ARE EVALUATED AT TIME = T1.
C            THIS STEP IS REPEATED FOR ALL TIME STEPS.
C
C    4.  SUBROUTINE ZRESP CALL S THE FOLLOWING FORMA SUBROUTINES:
C        - PAGEHD, READ, TIMCHK, ZAABB, ZINV3, ZMULT, ZREAD, ZSIZE,
C          ZTOD, ZTRANS, ZWRKFL, ZZBOMB
C
C    5.  SUBROUTINE ZRESP ALSO CALLS THE FOLLOWING SPECIAL PURPOSE
C        SUBROUTINES THAT WERE DEVELOPED ESPECIALLY FOR THIS SOFTWARE
C        PACKAGE:
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C       - ZABDI, ZMULTCD, ZMULTDD
C
C    SUBROUTINE ARGUMENTS
C    --------------------
C
C         SUBROUTINE ZRESP HAS NO SUBROUTINE ARGUMENTS.
C
C
C    EXAMPLE OF A CALLING PROGRAM ON CDC 172/720/730
C    -----------------------------------------------
C
C    FILE ASSUMPTIONS:
C       - TAPE1    =  WORK FILE REQUIRED BY ZRESP.  NWRKFL=1.
C       - TAPE10   =  FORMA FILE (FOR INPUT DATA).  NOTE THAT
C                     MORE THAN ONE FORMA FILE MAY BE NECESSARY
C                     IF  NPUT DATA IS RECORDED ON SEVERAL FORMA
C       - TAPE11   =  SEQUENTIAL FILE (FOR OUTPUT DATA).  NWFILE=11.
C
C       PROGRAM RESPNS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C      *                TAPE1,TAPE10,TAPE11)
C
C     1 CALL START
C                                              CALL TIMCHK (6HTBEGIN)
C                                              CALL TIMCHK (6HZRESP )
C       CALL ZRESP
C                                              CALL TIMCHK (6HZRESP )
C                                              CALL TIMCHK (6HTPRINT)
C       GO TO 1
C       END
C
C
C
C    INPUT FORM
C    ----------
C       CALLING PROGRAM MUST CALL START
C
C       READ   NWFILE,NWRKFL,NWRITE,NFORFL      (4I5)
C       READ   STARTT, ENDT, DELTAT             (3E10.0)
C       READ   GAMMA, BETA                      (2E10.0)
C       READ   NF, NB, IF, NP                   (4I5)
C       READ   NDAMPB, NDAMPI, NDAMPP           (3I5)
C       CALL ZREAD (FREQBT)
C       IF (NDAMPB .EQ. 0)  GO TO 1
C       CALL READ (DAMPB,NR1,NC1,1,KB)
C       GO TO 2
C     1 READ   ZETAB                            (E10.0)
C     2 CALL ZREAD (FREQI)
C       IF (NDAMPI .EQ. 0)  GO TO 3
C       CALL READ (DAMPI,NR2,NC2,1,KI)
C       GO TO 4
C     3 READ   ZETAI                            (E10.0)
C     4 CALL ZREAD (FREQPA)
C       IF (NDAMPP .EQ. 0)  GO TO 5
C       CALL READ (DAMPP,NR2,NC2,1,KP)
C       GO TO 6
C     5 READ   ZETAP                            (E10.0)
C     6 CALL ZREAD (B2)
C       CALL ZREAD (P2)
C       CALL ZREAD (PHIIB)
C       CALL READ (QNBO,NR3,NC3,1,KB)
C       CALL READ (QNBDO,NR4,NC4,1,KB)
C       CALL READ (QIBO,NR5,NC5,1,KI)
```

```
C       CALL READ (QIBDO,NR6,NC6,1,KI)
C       CALL READ (QNPO,NR7,NC7,1,KP)
C       CALL READ (QNPDO,NR8,NC8,1,KP)
C
C
C    DEFINTITION OF INPUT VARIABLES
C    ------------------------------
C
C             B2  =  THE INTERFACE/BOOSTER MASS COUPLING MATRIX IN
C                    PARTITION-LOGIC.  SIZE (NB,IF).
C           BETA  =  INPUT PARAMETER FOR NEWMARK-CHAN-BETA NUMERICAL
C                    INTEGRATION TECHNIQUE.  A GOOD VALUE IS BETA= 0.25
C          DAMPB  =  BOOSTER MODAL DAMPING MATRIX IN DENSE-LOGIC.
C                    DIAGONAL MATRIX INPUT AS A ROW VECTOR.
C                    SIZE (1,NB).
C          DAMPI  =  INTERFACE MODAL DAMOPING MATRIX IN DENSE-LOGIC.
C                    DIAGONAL MATRIX INPUT AS A ROW VECTOR.
C                    SIZE (1,IF).
C          DAMPP  =  PAYLOAD MODAL DAMPING MATRIX IN DENSE-LOGIC.
C                    DIAGONAL MATRIX INPUT AS A ROW VECTOR.
C                    SIZE (1,NP)
C         DELTAT  =  TIME STEP FOR INTEGRATION ROUTINE.
C           ENDT  =  STOPPING TIME FOR THE NUMERICAL INTEGRATION
C                    ROUTINE.
C         FREQBT  =  TRUNCATED, CANTILEVERED BOOSTER FREQUENCY VECTOR
C                    IN PARTITION-LOGIC.  SIZE (1,NB)
C          FREQI  =  INTERFACE FREQUENCY VECTOR IN PARTITION-LOGIC.
C                    SIZE (1,IF)
C         FREQPA  =  TRUNCATED, CANTILEVERED, ASSEMBLED PAYLOAD
C                    FREQUENCY VECTOR FOR ALL PAYLOADS IN PARTITION-
C                    LOGIC.  SIZE (1,NP).
C          GAMMA  =  INPUT PARAMETER FOR NEWMARK-CHAN-BETA NUMERICAL
C                    INTEGRATION TECHNIQUE.  A GOOD VALUE IS GAMMA=0.5
C             IF  =  NUMBER OF INTERFACE DOFS.
C             NB  =  NUMBER OF TRUNCATED, CANTILERED BOOSTER MODES.
C         NDAMPB  =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                       BOOSTER MODAL DAMPING MATRIX.
C                 =  1  FOR A VARIABLE VALUE OF MODAL DAMPING IN THE
C                       BOOSTER MODAL DAMPING MATRIX.
C                       (MUST INPUT DAMPB FOR NDAMPB = 1).
C         NDAMPI  =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                       INTERFACE MODAL DAMPING MATRIX.
C                    1  FOR A VARIABLE VALUE OF INTERFACE MODAL
C                       DAMPING.  MUST INPUT DAMPI WHEN NDAMPI=1.
C         NDAMPP  =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                       PAYLOAD MODAL DAMPING MATRIX.
C                 =  1  FOR A VARIABLE VALUE OF MODAL DAMPING.
C                       MUST INPUT DAMPP FOR NDAMPP = 1.
C             NF  =  NUMBER OF DOFS IN BOOSTER WHERE FORCES ARE APPLIED.
C         NFORFL  =  LOGICAL FILE NUMBER CONTAINING THE INTERPOLATED
C                    FORCE DATA.  THIS DATA IS SEQUENTIAL.
C             NP  =  TOTAL NUMBER OF TRUNCATED, CANTILEVERED PAYLOAD
C                    MODES.
C         NWFILE  =  LOGICAL FILE NUMBER FOR OUTPUT DATA.
C         NWRITE  =  0  RESULTS ARE NOT PRINTED ON PAPER.
C                 =  1  RESULTS ARE PRINTED ON PAPER.
C         NWRKFL  =  LOGICAL FILE NUMBER FOR WORK FILE.
C             P2  =  THE PAYLOAD/INTERFACE COUPLING MASS MATRIX.
C                    SIZE (IF,NP).
C          PHIIB  =  THE INTERFACE MODES MATRIX.  SIZE (IF,IF).
C           QIBO  =  DENSE-LOGIC VECTOR OF INITIAL MODAL DISPLACEMENTS
C                    OF THE INTERFACE DOFS AT STARTT.  SIZE (1,IF)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C          QIBDO   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                     OF THE INTERFACE DOFS AT STARTT.  SIZE (1,IF).
C           QNBO   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL
C                     DISPLACEMENTS OF THE BOOSTER DOFS AT THE STARTT.
C                     SIZE (1,NB).
C          QNBDO   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                     OF THE BOOSTER DOFS AT STARTT.  SIZE (1,NB)
C           QNPO   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL
C                     DISPLACEMENTS OF THE PAYLOAD DOFS.  SIZE (1,NP).
C          QNPDO   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                     OF THE PAYLOAD DOFS AT STARTT.  SIZE (1,NP).
C          ZETAB   =  VALUE OF BOOSTER MODAL DAMPING FOR ALL DOFS WHEN
C                     NDAMPB = 0.
C          ZETAI   =  VALUE OF INTERFACE DAMPING FOR ALL DOFS WHEN
C                     NDAMPI =0.
C          ZETAP   =  VALUE OF PAYLOAD MODAL DAMPING FOR ALL DOFS WHEN
C                     NDAMPP = 0.
C
C                                                      ORIGINAL PAGE IS
C                                                      OF POOR QUALITY
C    A.  DIMENSION, COMMON, DATA, FORMAT
C    -----------------------------------
      COMMON     /NITNOT/   NIT,NOT
      COMMON     /LSTRT4/   NLINE,NLPP
      DIMENSION D1(500),D2(200),D3(500),D4(500),D5(200),D6(500),D7(500),
     +           D8(200),D9(500),QNBO(500),QNB(500),QNBDO(500),QNBD(500),
     +           QNBDDO(500),QNBDD(500),QIBO(200),QIB(200),QIBDO(200),
     +           QIBD(200),QIBDDO(200),QIBDD(200),QNPO(500),QNP(500),
     +           QNPDO(500),QNPD(500),QNPDDO(500),QNPDD(500),OMB2(500),
     +           OMP2(500),FB(500),FI(200),FP(500),
     +           HELP11(200),OMI2(200)
C
      DATA       KB,KI,KP             /500,200,500/
      DATA       BUF                  /-1.E50/
C
 1000 FORMAT (10I5)
 1100 FORMAT (10E10.0)
 1200 FORMAT (//9X,8H TIME = ,F10.6)
 1250 FORMAT (//10X,*TIME = *,F10.6,4X,*(CONTINUED)*)
 1300 FORMAT (//9X,15H APPLIED FORCES /(10X,1P5E16.8))
 1400 FORMAT (//5X,*INTERFACE RESPONSE  (MODAL COORDINATES)*,/,
     +           5X,*---------  --------  -------------------*,
     +          //9X,4H ROW,6X,13H ACCELERATION,8X,9H VELOCITY,10X,
     +       13H DISPLACEMENT//,(10X,I3,1P3E20.8))
 2002 FORMAT (//,48X,*STARTT  =*,F10.6,/,50X,*ENDT  =*,F10.6,/,
     +             48X,*DELTAT  =*,F10.6)
 2003 FORMAT (//,30X,*PARAMETERS FOR NEWMARK-CHAN-BETA INTEGRATION*,
     +           * ROUTINE*,//,49X,*GAMMA  =*,F10.6,/,
     +           50X,*BETA  =*,F10.6)
 1500 FORMAT (//4X,* NON-INTERFACE RESPONSE  (MODAL COORDINATES)*,/,5X,
     +         13(1H-),1X,8(1H-),2X,19(1H-),//9X,* ROW*,7X,
     +         *ACCELERATION*,9X,*VELOCITY*,11X,*DISPLACEMENT*//
     +         ,(10X,I3,1P3E20.8))
 2001 FORMAT (/////,50X,*INPUT PARAMETERS*,/,50X,16(1H-),///,
     +         48X,*NWFILE  =*,I5,/,48X,*NWRKFL  =*,I5,/,
     +         48X,*NFORFL  =*,I5,//,
     +         30X,*DATA IS WRITTEN ON PAPER EVERY *,I5,* TIME STEPS*)
 2004 FORMAT (/30X,*VALUE OF BOOSTER MODAL DAMPING IS CONSTANT*,
     +          5X,*(NDAMPB = 1)*)
 2005 FORMAT (/30X,*VALUE OF PAYLOAD MODAL DAMPING IS A CONSTANT*,
     +          5X,*(NDAMPP = 1)*)
 2006 FORMAT (//30X,*NUMBER OF FORCES APPLIED TO BOOSTER  =*,I5,//,
```

```
     +          32X,*NUMBER OF TRUNCATED BOOSTER MODES  =*,I5,//,
     *          41X,*NUMBER OF INTERFACE DOFS  =*,I5,//,
     +          32X,*NUMBER OF TRUNCATED PAYLOAD MODES  =*,I5)
C
C     ******************************
C       BEGINNING OF PROGRAM
C     ******************************
C
C   B.  READ INPUT DATA
C   -------------------
      READ (NIT,1000) NWFILE,NWRKFL,NWRITE,NFORFL
      READ (NIT,1100) STARTT,ENDT,DELTAT
      READ (NIT,1100) GAMMA,BETA
      READ (NIT,1000) NF,NB,IF,NP
      READ (NIT,1000) NDAMPB,NDAMPI,NDAMPP
      WRITE (NOT,2001) NWFILE,NWRKFL,NFORFL,NWRITE
      WRITE (NOT,2002) STARTT,ENDT,DELTAT
      WRITE (NOT,2003) GAMMA,BETA
      IF (NDAMPB .EQ. 0)    WRITE(NOT,2004)
      IF (NDAMPP .EQ. 0)    WRITE(NOT,2005)
      WRITE (NOT,2006) NF,NB,IF,NP
C
      CALL ZWRKFL (NWRKFL)
C
C   C.  CALCULATION OF CONSTANTS
C   ----------------------------
      C0=DELTAT*DELTAT
      C1=GAMMA*DELTAT
      C2=BETA*C0
      C3=(1.-GAMMA)*DELTAT
      C4=(0.5-BETA)*C0
C
C   D.  CALCULATION OF THE VECTORS D1,D4, AND D7
C   --------------------------------------------
                                                  CALL TIMCHK (6HDVECS )
      CALL ZREAD (FREQBT)
      CALL ZTOD (FREQBT,QNB,1,NB2,1,KB)
                                                              NERROR=1
      IF (NB .NE. NB2)                                        GO TO 999
      IF (NDAMPB .EQ. 0) GO TO 10
      CALL READ (QNBD,NR1,NC1,1,KB)
      IF (NB .NE. NC1)                                        GO TO 999
C
      DO 20 I=1,NB
      QNBD(I)=12.56637061*QNBD(I)*QNB(I)
      OMB2(I)=39.4784176*QNB(I)*QNB(I)
   20 CONTINUE
      GO TO 30
   10 CONTINUE
      READ (NIT,1100) ZETAB
      DO 40 I=1,NB
      QNBD(I)=12.56637061*ZETAB*QNB(I)
      OMB2(I)=39.4784176*QNB(I)*QNB(I)
   40 CONTINUE
   30 CONTINUE
C
      DO 50 I=1,NB
      D1(I)=1.+C1*QNBD(I)+C2*OMB2(I)
      D4(I)=QNBD(I)+DELTAT*OMB2(I)
      D7(I)=C3*QNBD(I)+C4*OMB2(I)
   50 CONTINUE
C
```

```
C    E.  CALCUATION OF THE VECTORS D2, D5, AND D8
C    -----------------------------------------
        CALL ZREAD (FREQI)
        CALL ZTOD (FREQI,QIB,1,IF2,1,KI)
                                                               NERROR=2
        IF (IF .NE. IF2)                                       GO TO 999
        IF (NDAMPI .EQ. 0)   GO TO 55
        CALL READ (QIBD,NR3,NC3,1,KI)
        IF (IF .NE. NC3)                                       GO TO 999
C
        DO 52 I=1,IF
        QIBD(I)=12.56637061*QIBD(I)*QIB(I)
        OMI2(I)=39.4784176*QIB(I)*QIB(I)
     52 CONTINUE
        GO TO 58
     55 READ (NIT,1100)  ZETAI
        DO 56 I=1,IF
        QIBD(I)=12.56637061*ZETAI*QIB(I)
        OMI2(I)=39.4784176*QIB(I)*QIB(I)
     56 CONTINUE
     58 CONTINUE
        DO 59 I=1,IF
        D2(I)=1.+C1*QIBD(I)+C2*OMI2(I)
        D5(I)=QIBD(I)+DELTAT*OMI2(I)
        D8(I)=C3*QIBD(I)+C4*OMI2(I)
     59 CONTINUE
C
C
C    F.  CALCULATION OF THE VECTORS D3, D6, AND D9
C    -------------------------------------------
        CALL ZREAD (FREQPA)
        CALL ZTOD (FREQPA,QNP,1,NP2,1,KP)
                                                               NERROR=3
        IF (NP .NE. NP2)                                       GO TO 999
C
        IF (NDAMPP .EQ. 0) GO TO 60
        CALL READ (QNPD,NR2,NC2,1,KP)
        IF (NP .NE. NC2)                                       GO TO 999
C
        DO 70 I=1,NP
        QNPD(I)=12.56637061*QNPD(I)*QNP(I)
        OMP2(I)=39.4784176*QNP(I)*QNP(I)
     70 CONTINUE
        GO TO 80
     60 CONTINUE
        READ(NIT,1100) ZETAP
        DO 90 I=1,NP
        QNPD(I)=12.5663706 * ZETAP * QNP(I)
        OMP2(I) = 39.4784176*QNP(I)*QNP(I)
     90 CONTINUE
C
     80 CONTINUE
C
        DO 100 I=1,NP
        D3(I)=1.+C1*QNPD(I)+C2*OMP2(I)
        D6(I)=QNPD(I)+DELTAT*OMP2(I)
        D9(I)=C3*QNPD(I)+C4*OMP2(I)
    100 CONTINUE
C
                                              CALL TIMCHK(6HDVECS )
C
C    G.  CALCULATION OF A1, A2, AND A3 MATRICES
```

```
C    ---------------------- --------------
                                                    CALL TIMCHK(6HA123   )
      CALL ZREAD (B2)
      CALL ZREAD (P2)
      CALL ZREAD (PHIIB)                           ORIGINAL PAGE IS
      CALL ZTRANS (PHIIB,PHIIBT)                   OF POOR QUALITY
C
      CALL ZSIZE (B2,IF2,NB2)
                                                                NERROR=4
      IF (IF.NE.IF2 .OR. NB.NE.NB2)                             GO TO 999
      CALL ZSIZE (P2,IF2,NP2)
                                                                NERROR=5
      IF (IF.NE.IF2 .OR. NP.NE.NP2)                             GO TO 999
      CALL ZSIZE (PHIIB,IF2,IF3)
                                                                NERROR=6
      IF (IF.NE.IF2 .OR. IF.NE.IF3)                             GO TO 999
C
      DO 110 I=1,NB
      D1(I)=-1./D1(I)
  110 CONTINUE
      DO 120 I=1,NP
      D3(I)=-1./D3(I)
  120 CONTINUE
      CALL ZMULTDD (B2,D1,NB,A2)
      CALL ZMULTDD (P2,D3,NP,A3)
      CALL ZTRANS (B2,B2T)
      CALL ZTRANS (P2,P2T)
      CALL ZMULT (A2,B2T,A2B2T)
      CALL ZMULT (A3,P2T,A3P2T)
      CALL ZAABB (1.,A2B2T,1.,A3P2T,HELP2)
      CALL ZABDI (HELP2,D2,IF,A1I)
      CALL ZINV3 (A1I,A1,1)
      CALL ZTRANS (A2,A2T)
      CALL ZTRANS (A3,A3T)
                                                    CALL TIMCHK(6HA123   )
C
C    H.  CALCULATION OF INITIAL ACCELERATIONS - QNBDD0, QIBDD0, QNPDD0
C    --------------------------------------------------------------
                                                    CALL TIMCHK (6HINITL )
      CALL ZMULT (B2,B2T,HELP3)
      CALL ZMULT (P2,P2T,HELP4)
      CALL ZAABB (-1.,HELP4,-1.,HELP3,HELP5)
      DO 125 I=1,IF
      FI(I)=1.
  125 CONTINUE
      CALL ZABDI (HELP5,FI,IF,AI)
      CALL ZINV3 (AI,A,1)
C
      CALL READ (QNB0,NR3,NC3,1,KB)
      CALL READ (QNBD0,NR4,NC4,1,KB)
      CALL READ (QIB0,NR5,NC5,1,KI)
      CALL READ (QIBD0,NR6,NC6,1,KI)
      CALL READ (QNP0,NR7,NC7,1,KP)
      CALL READ (QNPD0,NR8,NC8,1,KP)
C
C
      REWIND NFORFL
      READ (NFORFL) IRUNNO,IDATE,START2,ENDT2,DELTAT2,(BUFCH,I=1,10)
      READ (NFORFL) T2,(FB(I),I=1,NB),(FI(I),I=1,IF),BUFCH
C
      DO 130 I=1,NB
      FB(I)=FB(I)-QNBD(I)*QNBD0(I)-OMB2(I)*QNB0(I)
```

```
  130 CONTINUE
C
      CALL ZMULTCD (PHIIBT,FI,FP,IF,KP)
C
      DO 140 I=1,IF
      FI(I)=FP(I)-QIBD(I)*QIBCO(I)-OMI2(I)*QIBO(I)
  140 CONTINUE
C
      DO 150 I=1,NP
      FP(I)=-QNPD(I)*QNPDO(I)-OMP2(I)*QNPO(I)
  150 CONTINUE
C
      CALL ZMULTCD (B2,FB,QNBD,NB,KB)
      CALL ZMULTCD (P2,FP,HELP11,NP,KI)
C
      DO 160 I=1,IF
      FI(I)=FI(I)-QNBD(I)-HELP11(I)
  160 CONTINUE
C
      CALL ZMULTCD (A,FI,QIBDDO,IF,KI)
      CALL ZMULTCD (B2T,QIBDDO,QNBDDO,IF,KB)
      CALL ZMULTCD (P2T,QIBDDO,QNPDDO,IF,KP)
C
      DO 170 I=1,NB
      QNBDDO(I)=FB(I)-QNBDDO(I)
  170 CONTINUE
C
      DO 180 I=1,NP
      QNPDDO(I)=FP(I)-QNPDDO(I)
  180 CONTINUE
                                                  CALL TIMCHK (6HINITL )
C
C    I.  CALCULATE THE NUMBER OF TIME POINTS TO BE USED
C    ------------------------------------------------
      NTP=(ENDT-STARTT)/DELTAT+1.1
      NW=NWRITE
      NX=NB+IF+NP
C
      REWIND NWFILE
      WRITE (NWFILE) IRUNNO,IDATE,STARTT,ENDT,DELTAT,IF,NP,(BUF,I=1,10)
C
C    J.  RESPONSE LOOP
C    ----------------
                                                  CALL TIMCHK (6HRESP   )
      DO 500 ITP=1,NTP
      T=STARTT+FLOAT(ITP-1)*DELTAT
C
      READ (NFORFL) T2,(FB(I),I=1,NB),(FI(I),I=1,IF)
C
      DO 190 I=1,NB
      FB(I)=FB(I)-D4(I)*QNBDO(I)-D7(I)*QNBDDO(I)-OMB2(I)*QNBO(I)
  190 CONTINUE
C
      CALL ZMULTCD (PHIIBT,FI,FP,IF,KP)
C
      DO 200 I=1,IF
      FI(I)=FP(I)-D5(I)*QIBDO(I)-D8(I)*QIBDDO(I)-OMI2(I)*QIBO(I)
  200 CONTINUE
C
      DO 210 I=1,NP
      FP(I)=-D6(I)*QNPDO(I)-D9(I)*QNPDDO(I)-OMP2(I)*QNPO(I)
  210 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                       ORIGINAL PAGE IS
C        J1.  CALCULATION OF RESPONSE                   OF POOR QUALITY
C             --------------------------
      CALL ZMULTCD (A2,FB,QIBDD,NB,KI)
      CALL ZMULTCD (A3,FP,HELP11,NP,KI)
C
      DO 220 I=1,IF
      FI(I)=FI(I)+QIBDD(I)+HELP11(I)
  220 CONTINUE
C
      CALL ZMULTCD (A1,FI,QIBDD,IF,KI)
C
      DO 230 I=1,IF
      QIBD(I)=QIBDO(I)+C3*QIBDDO(I)+C1*QIBDD(I)
      QIB(I)=QIBO(I)+DELTAT*QIBDO(I)+C4*QIBDDO(I)+C2*QIBDD(I)
  230 CONTINUE
C
      CALL ZMULTCD (A2T,QIBDD,QNBDD,IF,KB)
C
      DO 240 I=1,NB
      QNBDD(I)=-D1(I)*FB(I)+QNBDD(I)
      QNBD(I)=QNBDO(I)+C3*QNBDDO(I)+C1*QNBDD(I)
      QNB(I)=QNBO(I)+DELTAT*QNBDO(I)+C4*QNBDDO(I)+C2*QNBDD(I)
  240 CONTINUE
C
      CALL ZMULTCD (A3T,QIBDD,QNPDD,IF,KP)
C
      DO 260 I=1,NP
      QNPDD(I)=-D3(I)*FP(I)+QNPDD(I)
      QNPD(I)=QNPDO(I)+C3*QNPDDO(I)+C1*QNPDD(I)
      QNP(I)=QNPO(I)+DELTAT*QNPDO(I)+C4*QNPDDO(I)+C2*QNPDD(I)
  260 CONTINUE
C
C        J2.  WRITE ANSWERS ON NWFILE FOR LATER USE
C             ----------------------------------------
      WRITE (NWFILE) T,      (QIBO(I),I=1,IF),     (QNPO(I),I=1,NP),
     +                       (QIBDO(I),I=1,IF),    (QNPDO(I),I=1,NP),
     +                       (QIBDDO(I),I=1,IF),   (QNPDDO(I),I=1,NP),
     +                       BUF
C
C        J3.  SEE IF DATA SHOULD BE PRINTED
C             ------------------------------
      IF (ITP .LT. NTP .AND. NW .LT. NWRITE) GO TO 300
                                                CALL TIMCHK (6HWRITE )
      CALL PAGEHD
      NLINE=13
      WRITE (NOT,1200) T
      NXSI=1
      NXEI=IF
      IF (NXEI .GT. (NLPP-NLINE))     NXEI=NLPP-NLINE
  310 WRITE (NOT,1400) (I,QIBDDO(I),QIBDO(I),QIBO(I),I=NXSI,NXEI)
      NLINE=NLINE+NXEI-NXSI+1
      IF (IF .EQ. NXEI) GO TO 315
      CALL PAGEHD
      WRITE (NOT,1250)  T
      NLINE=13
      NXSI=NXEI+1
      NXEI=IF
      IF ((NXEI-NXSI) .GT. (NLPP-NLINE))    NXEI=NXSI+NLPP-NLINE
      GO TO 310
  315 CONTINUE
      NXSP=1
```

```
      NXEP=NP
      NLINE=NLINE+13
      IF ((NLINE+10) .LT. NLPP)    GO TO 318
      CALL PAGEHD
  318 WRITE (NOT,1250)  T
      NLINE=NLINE+10
      IF (NXEP .GT. (NLPP-NLINE))   NXEP=NLPP-NLINE
  320 WRITE (NOT,1500) (I+IF,QNPDDO(I),QNPDO(I),QNPO(I),I=NXSP,NXEP)
      IF (NP .EQ. NXEP) GO TO 330
      NXSP=NXEP+1
      NXEP=NP
      CALL PAGEHD
      WRITE (NOT,1250)  T
      NLINE=10
      IF ((NXEP-NXSP) .GT. (NLPP-NLINE)) NXEP=NXSP+NLPP-NLINE
      GO TO 320
C
  330 NW=0
                                                   CALL TIMCHK (6HWRITE )
  300 NW=NW+1
C
C
C       J4.  REASSIGN
C       -------------
      DO 400 I=1,NB
      QNBO(I)=QNB(I)
      QNBDO(I)=QNBD(I)
      QNBDDO(I)=QNBDD(I)
  400 CONTINUE
      DO 410 I=1,IF
      QIBO(I)=QIB(I)
      QIBDO(I)=QIBD(I)
      QIBDDO(I)=QIBDD(I)
  410 CONTINUE
      DO 420 I=1,NP
      QNPO(I)=QNP(I)
      QNPDO(I)=QNPD(I)
      QNPDDO(I)=QNPDD(I)
  420 CONTINUE
C
  500 CONTINUE
                                                    CALL TIMCHK (6HRESP   )
C
C    K. EXIT
C    -------
      RETURN
  999 CONTINUE
      CALL ZZBOMB (6HZRESP ,NERROR)
      END
```

```
      PROGRAM LOADS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +               TAPE20,TAPE30,TAPE31,TAPE32,TAPE33,TAPE40,TAPE41)
C
C
    1 CALL START
                                                    CALL TIMCHK (6HTBEGIN)
                                                    CALL TIMCHK (6HZLOADS)
      CALL ZLOADS
                                                    CALL TIMCHK (6HZLOADS)
                                                    CALL TIMCHK (6HTPRINT)
C
      GO TO 1
      END
C
C
C     ***********************
C     *                     *
C     *       ZLOADS        *
C     *                     *
C     ***********************
C
C
      SUBROUTINE ZLOADS
C
C    SUBROUTINE ZLOADS PRODUCES ALL THE NECESSARY PAYLOAD MEMBER LOADS.
C    MAXIMUM AND MINIMUM LOADS ARE CALCULATED ON REQUEST.
C
C    DEVELOPED BY RC ENGELS AND TG SHANAHAN, FEBRUARY 1982.
C
C
C
C
C          COMMENTS
C          --------
C          1.  SUBROUTINE ZLOADS IS PART OF A COMPLETE BOOSTER/PAYLOAD
C              INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C              INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE
C              CAN BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS
C              ESTIMATES - FINAL REPORT, MAY 1982".
C
C          2.  PROGRAM ZLOADS SHOULD BE RUN FOR EACH PAYLOAD
C              SEPERATELY.
C
C          3.  THE FOLLOWING INFORMATION IS PUT ON NWFILE :
C              A.  A HEADER CONTAINING THE RUN NUMBER, DATE, START TIME
C                  OF THE LOADS CALCULATION, END TIME OF THE LOADS
C                  CALCULATION, TIME STEP, NUMBER OF DEGREES OF FREEDOM
C                  USED, AND THE NUMBER OF TIME STEPS.
C              B.  THE TIME AND LOADS RESULTS IN SEQUENTIAL ORDER :
C                    T1,  (LOADS(I),I=1,NUMBER OF DOFS),
C                    T2,  (LOADS(I),I=1,NUMBER OF DOFS),
C                    .,   .      .       .       .    .
C                    ENDT,(LOADS(I),I=1,NUMBER OF DOFS)
C
C          4.  IF MAXL=1 THE MAXIMUM/MINIMUM LOADS MATRIX IS WRITTEN
C              ON NTAPE.  NTAPE IS A DENSE-LOGIC FORMA FILE.
C
C          5.  SUBROUTINE ZLOADS USES THE FOLLOWING FORMA SUBROUTINES:
C              -  LTAPE, PAGEHD, READIM, TIMCHK, WRITE, WTAPE, ZMULT,
C                 ZREAD, ZSIZE, ZSLADR, ZWRKFL, ZZBOMB, ZZERO
C
C          6.  IT ALSO USES SPECIAL PURPOSE SUBROUTINE ZMULTCD
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C       SUBROUTINE ARGUMENTS
C       --------------------
C       SUBROUINE ZLOADS HAS NO SUBROUTINE ARGUMENTS
C
C       EXAMPLE OF A CALLING PROGRAM ON THE CDC 172/720/730
C       ----------------------- ------------------------
C       FILE ASSUMPTIONS
C        -  TAPE1  =  WORK FILE REQUIRED BY ZLOADS.  NWRKFL=1.
C        - TAPE10  =  SEQUENTIAL INPUT FILE CONTAINING RESPONSE DATA.
C                     NRESPFL=10.
C        - TAPE11  =  SEQUENTIAL OUTPUT FILE CONTAINING LOADS DATA.
C                     NWFILE=11.
C        - TAPE12  =  FORMA OUTPUT FILE CONTAINING MAXIMUM/MINIMUM
C                     LOADS DATA.  NTAPE=12
C
C
C     PROGRAM LOADS (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C    +               TAPE1,TAPE10,TAPE11,TAPE12)
C
C   1 CALL START
C                                            CALL TIMCHK (6HTBEGIN)
C                                            CALL TIMCHK (6HZLOADS)
C     CALL ZLOADS
C                                            CALL TIMCHK (6HZLOADS)
C                                            CALL TIMCHK (6HTPRINT)
C     GO TO 1
C     END
C
C       INPUT FORM
C       ----------
C       READ  MAXL,ISELECT,NWRITE                        (3I5)
C       READ  NQNPS,NQNP                                 (2I5)
C       READ  IFB,IFP,NP                                 (3I5)
C       READ  NRESPFL,NWRKFL,NWFILE,NTAPE                (4I5)
C       CALL ZREAD (PKPSI)
C       IF (ISELECT .EQ. 0)  GO TO 1
C       CALL READIM (IVSEL,NR1,IVSIZE,1,K1)
C     1 CALL ZREAD (PL1)
C       CALL ZREAD (PL2)
C       CALL ZREAD (TP)
C       CALL ZREAD (PHIIB)
C       CALL READIM (IVEC,NRI,IF1,1,KI)
C       RETURN
C
C
C       DEFINITION OF INPUT VARIABLES
C       -----------------------------
C           IFB  =  NUMBER OF BOOSTER INTERFACE DOFS.
C           IFP  =  NUMBER OF INTERFACE DOFS FOR THIS PAYLOAD.
C       ISELECT  =  0  ALL ROWS OF PKPSI ARE USED IN THE LOAD
C                      CALCULATIONS.
C                =  1  ONLY ROWS OF PKPSI THAT ARE SELECTED IN IVSEL
C                      ARE USED IN THE LOAD CALCULATIONS.
C          IVEC  =  VECTOR OF THE ROW LOCATIONS IN PHIIB THAT
C                   CORRESPOND TO INTERFACE DOFS OF THIS PAYLOAD.
C                   IVEC(I) = ROW LOCATION IN PHII OF THE I-TH
C                   INTERFACE DOF IN THIS PAYLOAD.  SIZE (IFP).
C         IVSEL  =  VECTOR OF ROWS OF PKPSI THAT INVOLVE THIS PAYLOAD.
C                   SIZE (1,IVSIZE).
C          MAXL  =  0  MAXIMUM/MINIMUM LOADS CALCULATION IS NOT
C                      DESIRED
```

```
C                     =  1  MAXIMUM/MINIMUM LOADS CALCULATION IS DESIRED.
C              ND  =  NUMBER OF DOFS IN THIS PAYLOAD.
C              NP  =  THE TOTAL NUMBER OF TRUNCATED CANTILEVERED
C                     MODES FOR ALL THE PAYLOADS.
C            NQNP  =  NUMBER OF NON-INTERFACE DOFS IN THE PAYLOAD.
C           NQNPS  =  THE POSITION NUMBER IN MATRIX (QNP) WHERE THE
C                     FIRST DOF FOR THE PAYLOAD OCCURS.
C         NRESPFL  =  LOGICAL FILE NUMBER OF THE SEQUENTIAL FILE
C                     CONTAINING THE RESPONSE DATA FOR THE SYSTEM.
C           NTAPE  =  LOGICAL FILE NUMBER OF A FORMA FILE FOR OUTPUT
C                     MAXIMUM/MINIMUM LOADS DATA.  IF MAXL=0 THIS TAPE
C                     IS NOT USED.
C          NWFILE  =  LOGICAL UNIT NUMBER OF SEQUENTIAL TAPE ON WHICH
C                     TO WRITE LOADS DATA.
C          NWRKFL  =  LOGICAL UNIT NUMBER OF PARTITION-LOGIC WORK
C                     FILE REQUIRED BY ZLOADS.
C          NWRITE  =  0  THE LOADS CALCULATION RESULTS ARE NOT PRINTED
C                        ON PAPER.
C                  =  N  THE LOADS RESULTS ARE PRINTED ON PAPER EVERY N
C                        TIME STEPS.
C           PHIIB  =  THE INTERFACE MODES MATRIX.  SIZE (IFB,IFB).
C           PKPSI  =  A LOADS TRANSFORMATION MATRIX.  TRANSFORMS DISRETE
C                     DISPLACEMENTS TO LOADS.  SIZE (ND,ND).
C             PL1  =  THE NON-INTERFACE LOADS TRANSFORMATION FOR THIS
C                     PAYLOAD.  SIZE (ND,ND).
C             PL2  =  THE INTERFACE LOADS TRANSFORMATION FOR THIS
C                     PAYLOAD.  SIZE (ND,IFP).
C              TP  =  THE CONSTRAINT MODAL MATRIX FOR THIS PAYLOAD.
C                     SIZE (ND,IFP).
C
C         DIMENSION, COMMON, DATA, AND FORMAT
C         ----------------------------------
      COMMON     /NITNOT/   NIT,NOT
      COMMON     /LSTRT4/   NLINE,NLPP
      COMMON     /LSTART/   IRUNNO,IDATE,NPAGE,UNAME(3),T1(12),T2(12)
C
      DIMENSION IVSEL(500),QIB(200),QNP(500),QIBDD(200),QNPDD(500),
     +          AL1(600),AL2(600),AL3(600),ALMM(600,4),QIBD(200),
     +          QNPD(500)
C
      DATA   K1,K2,KI  /500,600,200/
      DATA   BUF         /-1.E50/
C
 1000 FORMAT (10I5)
 2001 FORMAT (///,55X,*PAYLOAD LOADS*,/,
     +              55X,*-------------*,///,
     +              50X,*FOR THE TIME INTERVAL OF :*//,52X,*STARTT =*,
     +           F12.6,//,54X,*ENDT =*,F12.6,//,52X,*DELTAT =*,F12.6,//)
 2002 FORMAT (5(/),40X,*INPUT PARAMTERS TO ZLOADS*,/,
     +              40X,*-------------------------*,//)
 2003 FORMAT (18X,*MAXIMUM/MINIMUM LOAD CALCULATION WILL NOT BE *
     +              *PERFORMED*,5X,*(MAXL = 0)*,/)
 2004 FORMAT (20X,*MAXIMUN/MINIMUM LOAD CALCULATION WILL BE PERFORMED*,
     +           5X,*(MAXL = 1)*,/)
 2005 FORMAT (18X,*ALL ROWS OF PKPSI ARE USED IN THE LOAD CALCULATION*,
     +           5X,*(ISELECT = 0)*,/)
 2006 FORMAT ( 2X,*ONLY ROWS OF PKPSI THAT ARE SPECIFIED IN IVSEL ARE *,
     +              *USED IN THE LOAD CALCULATIONS*,5X,*(ISELECT = 1)*,//)
 2007 FORMAT (46X,*NWRITE =*,I5,///,
     +          26X,*THE NUMBER OF INTERFACE DOFS IN THIS PAYLOAD =*,I5,//
     +         ,10X,*THE ROW NUMBER OF THE FIRST NON-INTERFACE DOF FOR *,
     +              *THIS PAYLOAD IN THE BOOSTER =*,I5,//,
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
     +         24X,*THE NUMBER OF NON-INTERFACE DOFS IN THIS PAYLOAD =*,
     +            ,I5,//
     +         26X,*THE NUMBER OF INTERFACE DOFS IN THE BOOSTER =*,I5,//
     +         27X,*THE TOTAL NUMBER OF PAYLOAD MODES RETAINED =*,I5,///,
     +         45X,*NRESPFL =*,I5,//,46X,*NWRKFL =*,I5,//,46X,*NWFILE =*,
     +         I5,//,47X,*NTAPE =*,I5,//)
 2010 FORMAT (///,50X,*TIME =*,F12.6,//)
 2011 FORMAT (///,50X,*MAXIMUM AND MINIMUM LOADS*,/,
     +            ,50X,*------------------------*,//,
     +            ,30X,*FORM :  MAXIMUM, TIME AT MAXIMUM, MINIMUM,*,
     +                ,* TIME AT MINIMUM*,//)
C
C
C         BEGINNING OF PROGRAM                                  ORIGINAL PAGE IS
C         --------------------                                  OF POOR QUALITY
      READ (NIT,1000)  MAXL,ISELECT,NWRITE
      READ (NIT,1000)  NQNPS,NQNP
      READ (NIT,1000)  IFB,IFP,NP
      READ (NIT,1000)  NRESPFL,NWRKFL,NWFILE,NTAPE
      CALL PAGEHD
      WRITE (NOT,2002)
      IF (MAXL .EQ. 0)     WRITE (NOT,2003)
      IF (MAXL .NE. 0)     WRITE (NOT,2004)
      IF (ISELECT .EQ. 0)   WRITE (NOT,2005)
      IF (ISELECT .NE. 0)   WRITE (NOT,2006)
      WRITE (NOT,2007)  NWRITE,IFP,NQNPS,NQNP,IFB,NP,NRESPFL,
     +                  NWRKFL,NWFILE,NTAPE
C
C         COMPUTE MATRICES A=K*PSI*PL1 AND B=K*PSI*PL2
C         --------------------------------------------
                                                      CALL TIMCHK (6HAB
      CALL ZWRKFL (NWRKFL)
      IF (ISELECT .EQ. 0)  GO TO 100
      CALL ZREAD (PKPSI)
      CALL READIM (IVSEL,NR1,IVSIZE,1,K1)
      CALL ZSIZE (PKPSI,NR2,NPKPSI)
      CALL ZZERO (Z,IVSIZE,NPKPSI)
      CALL ZSLADR (1.,PKPSI,IVSEL,IVSIZE,Z)
      GO TO 200
C
  100 CONTINUE
      CALL ZREAD (Z)
      CALL ZSIZE (Z,IVSIZE,NCZ)
C
  200 CONTINUE
      CALL ZREAD (PL1)
      CALL ZREAD (PL2)
      CALL ZREAD (TP)
      CALL ZREAD (PHIIB)
C
C        EXTRACT INTERFACE MODES THAT INVOLVE THIS PAYLOAD
C        -------------------------------------------------
      CALL READIM (IVSEL,NRI,IF1,1,KI)
                                                          NERROR=1
      IF (IFP .NE. IF1)                                   GO TO 999
C
      CALL ZZERO (PHIIR,IFP,IFB)
      CALL ZSLADR (1.,PHIIB,IVSEL,IFP,PHIIR)
C
      CALL ZMULT (Z,PL1,A)
      CALL ZMULT (Z,PL2,B)
      CALL ZMULT (B,PHIIR,B2)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CALL ZMULT (Z,TP,C)
      CALL ZMULT (C,PHIIR,C2)
                                                    CALL TIMCHK (6HAB     )
C
C        CALCULATE MEMBER LOADS   L=A*ACCEL+B*DISPL
C        -----------------------------------------
                                                    CALL TIMCHK (6HLOADS )
      REWIND NRESPFL
      NW=NWRITE
C
      READ (NRESPFL) IRUNN2,IDAT2,STARTT,ENDT,DELTAT,IF2,NP2,
     +                         (BUFCH,I=1,10)
                                                             NERROR=2
      IF (NP2 .NE. NP)                                       GO TO 999
                                                             NERROR=3
      IF (IF2 .NE. IFB)                                      GO TO 999
C
      NTP=(ENDT-STARTT)/DELTAT+1.1
C
      REWIND NWFILE
      WRITE (NWFILE) IRUNNO,IDATE,STARTT,ENDT,DELTAT,IVSIZE,NTP,
     +                      (BUF,I=1,10)
C
      WRITE (NOT,2001) STARTT,ENDT,DELTAT
      DO 500 KK=1,NTP
      READ (NRESPFL) T,(QIB(I),I=1,IFB),(QNP(I),I=1,NP)
     +                ,(QIBD(I),I=1,IFB),(QNPD(I),I=1,NP)
     +                ,(QIBDD(I),I=1,IFB),(QNPDD(I),I=1,NP),BUFCH
C
C
      DO 300 J=1,NQNP
      QNPDD(J)=QNPDD(NQNPS+J-1)
  300 CONTINUE
      CALL ZMULTCD (A,QNPDD,AL1,NQNP,K2)
      CALL ZMULTCD (B2,QIBDD,AL2,IFB,K2)
      CALL ZMULTCD (C2,QIB,AL3,IFB,K2)
      DO 350 J=1,IVSIZE
      AL1(J)=-AL1(J)-AL2(J)+AL3(J)
  350 CONTINUE
C
      WRITE (NWFILE) T,(AL1(I),I=1,IVSIZE)
C
      IF (NWRITE .EQ. 0)    GO TO 500
      IF (KK .LT. NTP .AND. NW .LT. NWRITE)  GO TO 450
C
      CALL PAGEHD
      WRITE (NOT,2010) T
      NLINE=NLINE+10
      CALL WRITE (AL1,1,IVSIZE,6HLOADS ,1)
      NW=0
C
  450 CONTINUE
      NW=NW+1
C
  500 CONTINUE
C
                                                    CALL TIMCHK (6HLOADS )
      IF (MAXL .EQ. 0) GO TO 900
                                                    CALL TIMCHK (6HMAXMIN)
      REWIND NWFILE
      READ (NWFILE) IRUNNO,IDATE,STARTT,ENDT,DELTAT,IVSIZE,NTP,
     +                      (BUFCH,I=1,10)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
      READ (NWFILE) T,(AL1(I),I=1,IVSIZE)
C
      DO 800 I=1,IVSIZE
      ALMM(I,1)=AL1(I)
      ALMM(I,2)=T
      ALMM(I,3)=AL1(I)
      ALMM(I,4)=T
  800 CONTINUE
C
      DO 840 KK=2,NTP
      READ (NWFILE) T,(AL1(I),I=1,IVSIZE)
C
      DO 840 J=1,IVSIZE
      IF (AL1(J) .LE. ALMM(J,1)) GO TO 830
      ALMM(J,1)=AL1(J)
      ALMM(J,2)=T
  830 IF (AL1(J) .GE. ALMM(J,3)) GO TO 840
      ALMM(J,3)=AL1(J)
      ALMM(J,4)=T
  840 CONTINUE
C
      CALL PAGEHD
      WRITE (NOT,2011)
      NLINE=NLINE+10
      CALL WRITE (ALMM,IVSIZE,4,6HMAXMIN,K2)
C
      CALL WTAPE (ALMM,IVSIZE,4,6HMAXMIN,K2,NTAPE)
      CALL LTAPE (NTAPE)
                                                  CALL TIMCHK ( [illegible]AXMIN)
C
  900 CONTINUE
      RETURN
  999 CALL ZZBOMB (6HZLOADS,NERROR)
      END
```

```
      PROGRAM SCBRES (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +                TAPE10,TAPE30,TAPE31,TAPE40,TAPE41)
C
C
C           THIS PROGRAM GENERATES THE NOMINAL RESPONSE OF THE
C           BOOSTER WITHOUT PAYLOADS.  THE OUTPUT OF THIS PROGRAM
C           CAN THEN BE USED IN A DIRECT OR COUPLED BASE DRIVE
C           RESPONSE ANALYSIS OF A PAYLOAD.
C
C           CREATED   JU:  6
C
C
    1 CALL START
                                             CALL TIMCHK (6H ,EGIN)
                                             CALL TIMCHK (6H∠SCBR )
      CALL ZSCBRES
                                             CALL TIMCHK (6HZSCBR )
                                             CALL TIMCHK (6HTPRINT)
      GO TO 1
      END
C
C
C         ***************************************
C         *                                     *
C         *              ZSCBRES                *
C         *                                     *
C         ***************************************
C
C
      SUBROUTINE ZSCBRES
C
C
C        THIS SUBROUTINE DETERMINES THE NOMINAL BOOSTER RESPONSE TO A
C        GIVEN FORCING FUNCTION.  THE OUTPUT RESPONSE CAN THEN BE USED
         IN A COUPLED OR DIRECT BASE DRIVE RESPONSE ANALYSIS FOR A
C        PAYLOAD
C
C        DEVELOPED BY TG SHANAHAN AND RC ENGELS, MAY 1982.
C
C        COMMENTS
C        --------
C        1.  SUBROUTINE ZSCBRES IS PART OF A COMPLETE BOOSTER/PAYLOAD
C            INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C            INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE
C            CAN BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS
C            ESTIMATES - FINAL REPORT, SEPTEMBER 1982".
C        2.  SUBROUTINE ZSCBRES USES A MODIFIED FORM OF THE NEWMARK-
C            CHAN- BETA INTEGRATION SCHEME.
C        3.  THE FOLLOWING OUTPUT IS PUT ON NWFILE:
C            - A HEADER CONTAINING THE RUN NUMBER, DATE, STARTT, ENDT,
C              DELTAT, NB AND IF
C            - THEN FOR EACH TIME STEP:
C                 THE TIME, T, AND BOOSTER AND INTERFACE RESPONSE,
C                 QNB, QNBD, QNBDD, XIB, XIBD, AND XIBDD
C        4.  SUBROUTINE ZSCBRES USES THE FOLLOWING FORMA SUBROUTINES:
C            - PAGEHD, READ, TIMCHK, ZAA, ZABB, ZINV3, ZMULT, ZREAD,
C              ZTOD, ZTRANS, ZWRKFL, ZZBOMB
C        5.  SUBROUTINE ZSCBRES ALSO USES THE FOLLOWING SPECIAL PURPOSE
C            SUBROUTINES THAT WERE DEVOLOPED ESPECIALLY FOR THIS SOFTWARE
C            PACKAGE:
C            - ZMULTCD, ZMULTDD
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C        SUBROUTINE ARGUMENTS
C        --------------------
C
C        SUBROUTINE ZSCBRES HAS NO SUBROUTINE ARGUMENTS
C
C   EXAMPLE OF A CALLING PROGRAM ON CDC 172/720/730
C   -----------------------------------------------
C
C   FILE ASSUMPTIONS:
C       - TAPE1    =  WORK FILE REQUIRED BY ZSCBRES.  NWRKFL=1.
C       - TAPE10   =  FORMA FILE (FOR INPUT DATA).  NOTE THAT
C                     MORE THAN ONE FORMA FILE MAY BE NECESSARY
C                     IF INPUT DATA IS RECORDED ON SEVERAL FORMA
C       - TAPE11   =  SEQUENTIAL FILE (FOR OUTPUT DATA).  NWFILE=11.
C
C       PROGRAM SCBRES (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C     +                 TAPE1,TAPE10,TAPE11)
C
C    1 CALL START
C                                                  CALL TIMCHK (6HTBEGIN)
C                                                  CALL TIMCHK (6HZSCBR )
C      CALL ZSCBRES
C                                                  CALL TIMCHK (6HZSCBR )
C                                                  CALL TIMCHK (6HTPRINT)
C      GO TO 1
C      END
C
C
C
C   INPUT FORM
C   ----------
C       CALLING PROGRAM MUST CALL START
C
C       READ   NWFILE,NWRKFL,NWRITE,NFORFL        (4I5)
C       READ   STARTT, ENDT, DELTAT               (3E10.0)
C       READ   GAMMA, BETA                        (2E10.0)
C       READ   NDAMPB, NF, NB, IF                 (4I5)
C       CALL ZREAD (FREQBT)
C       IF (NDAMPB .EQ. 0)  GO TO 1
C       CALL READ (DAMPB,NR1,NC1,1,KB)
C       GO TO 2
C     1 READ   ZETAB                              (E10.0)
C     2 CALL ZREAD (BM1)
C       CALL ZREAD (BM2)
C       CALL ZREAD (BK2)
C       CALL READ (QNBO,NR3,NC3,1,KB)
C       CALL READ (QNBDO,NR4,NC4,1,KB)
C       CALL READ (XIBO,NR5,NC5,1,KI)
C       CALL READ (XIBDO,NR6,NC6,1,KI)
C
C
C   DEFINTITION OF INPUT VARIABLES
C   ------------------------------
C
C            BK2  =  THE BOOSTER STIFFNESS MATRIX REDUCED TO THE
C                    INTERFACE.  (TB)T*BK*TB.  SIZE (IF,IF).
C            BM1  =  THE BOOSTER COUPLING MASS MATRIX BETWEEN INTERFACE
C                    AND NON-INTERFACE DOFS.  SIZE (IF,NP).
C            BM2  =  THE BOOSTER MASS MATRIX REDUCED TO THE INTERFACE.
C                    (TB)T*MB*TB.  SIZE (IF,IF).
C           BETA  =  INPUT PARAMETER FOR NEWMARK-CHAN-BETA NUMERICAL
C                    INTEGRATION TECHNIQUE.  A GOOD VALUE IS BETA= 0.25
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C           DAMPB   =  BOOSTER M  L DAMPING MATRIX IN DENSE-LOGIC.
C                      DIAGONAL   TRIX INPUT AS A ROW VECTOR.
C                      SIZE (1,NL .
C          DELTAT   =  TIME STEP FOR INTEGRATION ROUTINE.
C            ENDT   =  STOPPING TIME FOR THE NUMERICAL INTEGRATION
C                      ROUTINE.
C          FREQBT   =  TRUNCATED, CANTILEVERED BOOSTER FREQUENCY VECTOR
C                      IN PARTITION-LOGIC.  SIZE (1,NB)
C           GAMMA   =  INPUT PARAMETER FOR NEWMARK-CHAN-BETA NUMERICAL
C                      INTEGRATION TECHNIQUE.  A GOOD VALUE IS GAMMA=0.5
C              IF   =  NUMBER OF INTERFACE DOFS.
C              NB   =  NUMBER OF TRUNCATED, CANTILERED BOOSTER MODES.
C          NDAMPB   =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                         BOOSTER MODAL DAMPING MATRIX.
C                   =  1  FOR A VARIABLE VALUE OF MODAL DAMPING IN THE
C                         BOOSTER MODAL DAMPING MATRIX.
C                         (MUST INPUT DAMPB FOR NDAMPB = 1).
C              NF   =  NUMBER OF DOFS IN BOOSTER WHERE FORCES ARE APPLIED.
C          NFORFL   =  LOGICAL FILE NUMBER CONTAINING THE INTERPOLATED
C                      FORCE DATA.  THIS DATA IS SEQUENTIAL.
C          NWFILE   =  LOGICAL FILE NUMBER FOR OUTPUT DATA.
C          NWRITE   =  0  RESULTS ARE NOT PRINTED ON PAPER.
C                   =  1  RESULTS ARE PRINTED ON PAPER.
C          NWRKFL   =  LOGICAL FILE NUMBER FOR WORK FILE.
C            XIB0   =  DENSE-LOGIC VECTOR OF INITIAL DISCRETE DISPLACEMENTS
C                      OF THE INTERFACE DOFS AT STARTT.  SIZE (1,IF)
C           XIBD0   =  DENSE-LOGIC VECTOR OF THE INITIAL DICRETE VELOCITIES
C                      OF THE INTERFACE DOFS AT STARTT.  SIZE (1,IF).
C            QNB0   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL
C                      DISPLACEMENTS OF THE BOOSTER DOFS AT THE STARTT.
C                      SIZE (1,NB).
C           QNBD0   =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                      OF THE BOOSTER DOFS AT STARTT.  SIZE (1,NB)
C           ZETAB   =  VALUE OF BOOSTER MODAL DAMPING FOR ALL DOFS WHEN
C                      NDAMPB = 0.
C
C
C
C           A.  DIMENSION, COMMON, DATA, AND FORMAT STATEMENTS
C           --------------------------------------------------
      DIMENSION      QNB(500),QNBD(500),QNBDD(500),
     +               QNB0(500),QNBD0(500),QNBDD0(500),
     +               XIB(200),XIBD(200),XIBDD(200),
     +               XIB0(200),XIBD0(200),XIBDD0(200),
     +               D1(500),D3(500),D5(500).OMB2(500),
     +               FB(500),FI(200)
C
C
      COMMON    /NITNOT/    NIT,NOT
      COMMON    /LSTRT4/    NLINE,NLPP
C
C
      DATA    KB,KI       /500,200/
      DATA    BUF         /-1.E50/
C
C
 1000 FORMAT (10I5)
 1100 FORMAT (10E10.0)
 2001 FORMAT (/////,50X,*INPUT PARAMETERS*,/,50X,16(1H-),///,
     +        48X,*NWFILE  =*,I5,/,48X,*NWRKFL  =*,I5,/,
     +        48X,*NFORFL  =*,I5,//,
     +        30X,*DATA IS WRITTEN ON PAPER EVERY *,I5,* TIME STEPS*)
```

```
 2002 FORMAT (//,48X,*STARTT  =*,F10.6,/,50X,*ENDT  =*,F10.6,/,
     +            48X,*DELTAT  =*,F10.6)
 2003 FORMAT (//,30X,*PARAMETERS FOR NEWMARK-CHAN-BETA INTEGRATION*,
     +           * ROUTINE*,//,49X,*GAMMA  =*,F10.6,/,
     +           50X,*BETA  =*,F10.6)
 2004 FORMAT (/30X,*VALUE OF BOOSTER MODAL DAMPING IS CONSTANT*,
     +           5X,*(NDAMPB = 0)*)
 2006 FORMAT (//30X,*NUMBER OF FORCES APPLIED TO BOOSTER  =*,I5,//,
     +           32X,*NUMBER OF TRUNCATED BOOSTER MODES  =*,I5,//,
     +           41X,*NUMBER OF INTERFACE DOFS  =*,I5)
 2010 FORMAT (//9X,8H TIME = ,F10.6)
 2011 FORMAT (//10X,*TIME = *,F10.6,4X,*(CONTINUED)*)
 2012 FORMAT (//5X,*BOOSTER RESPONSE  (MODAL COORDINATES)*,/,
     +           5X,*-------- --------  ------------------*,
     +           //9X,4H ROW,6X,13H ACCELERATION,8X,9H VELOCITY,10X,
     +        13H DISPLACEMENT//,(10X,I3,1P3E20.8))
 2013 FORMAT (//5X,*INTERFACE RESPONSE  (DISCRETE COORDINATES)*,/,
     +           5X,*--------- --------  --------- -----------*,
     +         //9X,* ROW*,7X,
     +         *ACCELERATION*,9X,*VELOCITY*,11X,*DISPLACEMENT*//
     +         ,(10X,I3,1P3E20.8))
C
C     ******************************
C       BEGINNING OF PROGRAM
C     ******************************
C
C
C          B.  READ INPUT DATA
C              ---------------
      READ (NIT,1000)   NWFILE,NWRKFL,NWRITE,NFORFL
      READ (NIT,1100)   STARTT,ENDT,DELTAT
      READ (NIT,1100)   GAMMA,BETA
      READ (NIT,1000)   NDAMPB,NF,NB,IF
C
      CALL PAGEHD
      WRITE (NOT,2001)   NWFILE,NWRKFL,NFORFL,NWRITE
      WRITE (NOT,2002)   STARTT,ENDT,DELTAT
      WRITE (NOT,2003)   GAMMA,BETA
      IF (NDAMPB .EQ. 0)   WRITE (NOT,2004)
      WRITE (NOT,2006)   NF,NB,IF
C
      CALL ZWRKFL (NWRKFL)
C
C          C.  CALCULATION OF CONSTANTS
C              ------------------------
      C0=DELTAT*DELTAT
      C1=GAMMA*DELTAT
      C2=BETA*C0
      C3=(1.0-GAMMA)*DELTAT
      C4=(0.5-BETA)*C0
C
C          D.  CALCULATION OF VECTORS  D3, D5, AND -D1INV
C              ------------------------------------------
                                            CALL TIMCHK (6HDVECS )
      CALL ZREAD (FREQB)
      CALL ZTOD (FREQB,QNB,1,NC2,1,KB)
                                                                NERROR=1
      IF (NC2 .NE. NB)                                          GO TO 999
C
      IF (NDAMPB .EQ. 0)   GO TO 40
C
      CALL READ (QNBD,NR1,NC1,1,KB)
```

ORIGINAL PAGE IS
OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

```
      IF (NC1 .NE. NB)                                       NERROR=2
                                                             GO TO 999
C
      DO 20 I=1,NB
      QNBD(I)=12.56637061*QNBD(I)*QNB(I)
      OMB2(I)=39.4784176*QNB(I)*QNB(I)
   20 CONTINUE
C
      GO TO 80
C
   40 READ (NIT,1100)  ZETAB
C
      DO 60 I=1,NB
      QNBD(I)=12.56637061*ZETAB*QNB(I)
      OMB2(I)=39.4784176*QNB(I)*QNB(I)
   60 CONTINUE
C
   80 CONTINUE
C
      DO 100 I=1,NB
      D1(I)=-1./(1.0+C1*QNBD(I)+C2*OMB2(I))
      D3(I)=QNBD(I)+DELTAT*OMB2(I)
      D5(I)=C3*QNBD(I)+C4*OMB2(I)
  100 CONTINUE
C
C          E.  CALCULATION OF D2, D4, AND D6
C          ---------------------------------
      CALL ZREAD (BM1)
      CALL ZREAD (BM2)
      CALL ZREAD (BK2)
C
      CALL ZAABB (1.0,BM2,C2,BK2,D2)
      CALL ZAA (DELTAT,BK2,D4)
      CALL ZAA (C4,BK2,D6)
                                   CALL TIMCHK (6HDVECS )
C
C          F.  CALCULATION OF A1 AND A2
C          ----------------------------
                                   CALL TIMCHK (6HA12    )
      CALL ZMULTDD (BM1,D1,NB,A2)
      CALL ZTRANS (BM1,BM1T)
      CALL ZMULT (A2,BM1T,H1)
      CALL ZAABB (1.0,H1,1.0,D2,A1I)
      CALL ZINV3 (A1I,A1,1)
      CALL ZTRANS (A2,A2T)
                                   CALL TIMCHK (6HA12    )
C
C          G.  CALCULATION OF INITIAL ACCELERATIONS  QNBDD0 AND XIBDD0
C          -----------------------------------------------------------
                                   CALL TIMCHK (6HINITL )
      CALL ZMULT (BM1,BM1T,H2)
      CALL ZAABB (-1.0,H2,1.0,BM2,H3)
      CALL ZINV3 (H3,H3I,1)
C
      CALL READ (QNB0,NR3,NC3,1,KB)
      CALL READ (QNBD0,NR4,NC4,1,KB)
      CALL READ (XIB0,NR5,NC5,1,KI)
      CALL READ (XIBD0,NR6,NC6,1,KI)
      IF (NC3 .NE. NB .OR. NC4 .NE. NB)                      NERROR=3
                                                             GO TO 999
      IF (NC5 .NE. IF .OR. NC6 .NE. IF)                      NERROR=4
                                                             GO TO 999
```

```
C
      REWIND NFORFL
      READ (NFORFL) IRUNNO,IDATE,STARTT2,ENDT2,DELTAT2,(BUFCH,I=1,10)
      READ (NFORFL) T2,(FB(I),I=1,NB),(FI(I),I=1,IF)
C
                                                               NERROR=5
      IF (T2 .NE. STARTT)                                      GO TO 999
C
      DO 120 I=1,NB
      FB(I)=FB(I)-QNBD(I)*QNBDO(I)-OMB2(I)*QNBO(I)
  120 CONTINUE
C
      CALL ZMULTCD (BK2,XIBO,XIB,IF,KI)
C
      DO 130 I=1,IF
      FI(I)=FI(I)-XIB(I)
  130 CONTINUE
C
      CALL ZMULTCD (BM1,FB,XIBD,NB,KI)
C
      DO 140 I=1,IF
      XIBDD(I)=FI(I)-XIBD(I)
  140 CONTINUE
C
      CALL ZMULTCD (H3I,XIBDD,XIBDDO,IF,KI)
      CALL ZMULTCD (BM1T,XIBDDO,QNBDD,IF,KB)
C
      DO 160 I=1,NB
      QNBDDO(I)=FB(I)-QNBDD(I)
  160 CONTINUE
                                               CALL TIMCHK (6HINITL )
C
C          H.  CALCULATE NUMBER OF TIME POINTS TO BE USED
C              ------------------------------------------
      NTP=(ENDT-STARTT)/DELTAT+1.1
      NW=NWRITE
      REWIND NWFILE
      WRITE (NWFILE)  IRUNNO,IDATE,STARTT,ENDT,DELTAT,NB,IF,(BUF,I=1,10)
C
C         I.  RESPONSE LOOP
C             -------------
                                               CALL TIMCHK (6HRESPON)
      DO 500 ITP=1,NTP
C
      T=STARTT+FLOAT(ITP-1)*DELTAT
      READ (NFORFL) T2,(FB(I),I=1,NB),(FI(I),I=1,IF)
C
      DO 220 I=1,NB
      FB(I)=FB(I)-D3(I)*QNBDO(I)-D5(I)*QNBDDO(I)-OMB2(I)*QNBO(I)
  220 CONTINUE
C
      CALL ZMULTCD (D4,XIBDO,XIBD,IF,KI)
      CALL ZMULTCD (D6,XIBDDO,XIBDD,IF,KI)
      CALL ZMULTCD (BK2,XIBO,XIB,IF,KI)
C
      DO 240 I=1,IF
      FI(I)=FI(I)-XIBD(I)-XIBDD(I)-XIB(I)
  240 CONTINUE
C
C                I2.  CALCULATION OF ACCELERATIONS - QNBDD AND XIBDD
C                     -----------------------------------------------
      CALL ZMULTCD (A2,FB,XIB,NB,KI)
```

```
C
      DO 260 I=1,IF                              ORIGINAL PAGE IS
      XIB(I)=XIB(I)+FI(I)                        OF POOR QUALITY
  260 CONTINUE
C
      CALL ZMULTCD (A1,XIB,XIBDD,IF,KI)
C
      CALL ZMULTCD (A2T,XIBDD,QNBDD,IF,KB)
C
      DO 280 I=1,NB
      QNBDD(I)=QNBDD(I)-D1(I)*FB(I)
  280 CONTINUE
C
C                I3.  CALCULATION OF QNB, QNBD, XIB, AND XIBD
C                ------------------------------------------
      DO 300 I=1,NB
      QNBD(I)=QNBD0(I)+C3*QNBDD0(I)+C1*QNBDD(I)
      QNB(I)=QNB0(I)+DELTAT*QNBD0(I)+C4*QNBDD0(I)+C2*QNBDD(I)
  300 CONTINUE
C
      DO 320 I=1,IF
      XIBD(I)=XIBD0(I)+C3*XIBDD0(I)+C1*XIBDD(I)
      XIB(I)=XIB0(I)+DELTAT*XIBD0(I)+C4*XIBDD0(I)+C2*XIBDD(I)
  320 CONTINUE
C
C                I4.  WRITE ANSWERS ON NWFILE FOR LATER USE
C                ----------------------------------------
      WRITE (NWFILE) T,(QNB0(I),I=1,NB),   (XIB0(I),I=1,IF),
     +                 (QNBD0(I),I=1,NB),  (XIBD0(I),I=1,IF),
     +                 (QNBDD0(I),I=1,NB), (XIBDD0(I),I=1,IF),BUF
C
C                I5.  SEE IF DATA SHOULD BE PRINTED
C                --------------------------------
      IF (ITP .LT. NTP .AND. NW .LT. NWRITE) GO TO 400
      CALL PAGEHD
      NLINE=10
      WRITE (NOT,2010) T
      NXSI=1
      NXEI=NB
      IF (NXEI .GT. (NLPP-NLINE))     NXEI=NLPP-NLINE
  360 WRITE (NOT,2012) (I,QNBDD0(I),QNBD0(I),QNB0(I),I=NXSI,NXEI)
      NLINE=NLINE+NXEI-NXSI+1
      IF (NB .EQ. NXEI) GO TO 370
      CALL PAGEHD
      WRITE (NOT,2011)  T
      NLINE=10
      NXSI=NXEI+1
      NXEI=NB
      IF ((NXEI-NXSI) .GT. (NLPP-NLINE))    NXEI=NXSI+NLPP-NLINE
      GO TO 360
C
  370 CONTINUE
      NXSP=1
      NXEP=IF
      IF ((NLINE+20) .LT. NLPP)    GO TO 380
      CALL PAGEHD
  380 WRITE (NOT,2011)  T
      NLINE=NLINE+10
      IF (NXEP .GT. (NLPP-NLINE))   NXEP=NLPP-NLINE
  390 WRITE (NOT,2013) (I+NB,XIBDD0(I),XIBD0(I),XIB0(I),I=NXSP,NXEP)
      IF (IF .EQ. NXEP) GO TO 395
      NXSP=NXEP+1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NXEP=IF
      CALL PAGEHD
      WRITE (NOT,2011)  T
      NLINE=10
      IF ((NXEP-NXSP) .GT. (NLPP-NLINE)) NXEP=NXSP+NLPP-NLINE
      GO TO 390
C
  395 NW=0
  400 NW=NW+1
C
C
C        I5.  REASSIGN
C        -------------
      DO 420 I=1,NB
      QNBO(I)=QNB(I)
      QNBDO(I)=QNBD(I)
      QNBDDO(I)=QNBDD(I)
  420 CONTINUE
C
      DO 440 I=1,IF
      XIBO(I)=XIB(I)
      XIBDO(I)=XIBD(I)
      XIBDDO(I)=XIBDD(I)
  440 CONTINUE
C
  500 CONTINUE
                                                 CALL TIMCHK (6HRESPON)
C
      RETURN
C
  999 CONTINUE
      CALL ZZBOMB (6HZRESP ,NERROR)
C
      END
```

```
      PROGRAM SCRESP6 (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
     +                 TAPE20,TAPE30 TAPE31,TAPE32,TAPE33,TAPE34,TAPE40)
C
    1 CALL START
                                               CALL TIMCHK (6HTBEGIN)
                                               CALL TIMCHK (6HSCPRES)
      CALL ZSCPRES
C
C
                                               CALL TIMCHK (6HSCPRES)
                                               CALL TIMCHK (6HTPRINT)
      GO TO 1
      END
C
C
C
C
C        ******************************
C        *                            *
C        *          ZSCPRES           *
C        *                            *
C        ******************************
C
C
C
      SUBROUTINE ZSCPRES
C
C     SUBROUTINE ZSCRES PRODUCES THE RESPONSE OF THE COUPLED
C     BOOSTER/PAYLOAD SYSTEM USING A COMBINATION BASE DRIVE / COUPLED
C     BASE DRIVE FORCING FUNCTION.
C
C     DEVELOPED BY RC ENGELS AND TG SHANAHAN, MAY 1982.
C
C
C     COMMENTS
C     --------
C     1.  SUBROUTINE ZSCPRES IS PART OF A COMPLETE BOOSTER/PAYLOAD
C         INTEGRATION SOFTWARE PACKAGE USING A DIRECT NUMERICAL
C         INTEGRATION TECHNIQUE.  A DISCUSSION OF THIS TECHNIQUE CAN
C         CAN BE FOUND IN "STRUCTURAL DYNAMICS PAYLOAD LOADS
C         ESTIMATES - FINAL REPORT, SEPTEMBER 1982".
C     2.  SUBROUTINE ZCSPRES USES A MODIFIED FORM OF THE NEWMARK-
C         CHAN-BETA INTEGRATION SCHEME.
C     3.  THE FOLLOWING DATA IS PUT ON NEWFILE:
C         - A HEADER
C         - THE INTERFACE AND PAYLOAD(S) MODAL RESPONSE DATA
C     4.  SUBROUTINE ZSCPRES USES THE FOLLOWING FORMA SUBROUTINES:
C         -  PAGEHD, READ, READIM, TIMCHK, ZAABB, ZERO, ZINV3, ZMULT,
C            ZREAD, ZRVAD, ZSIZE, ZTOD, ZTRANS, ZWRKFL, ZZBOMB, ZZERO
C     5.  SUBROUTINE ZCSPRES ALSO USES THE FOLLOWING SPECIAL PURPOSE
C         SUBROUTINES THAT WERE DEVELOPED ESPECIALLY FOR THIS SUBROUTINE
C         PACKAGE:
C         -  ZABDI, ZMULTCD, ZMULTDD
C
C     SUBROUTINE ARGUMENTS
C     --------------------
C
C          SUBROUTINE ZSCPRES HAS NO SUBROUTINE ARGUMENTS.
C
C
C     EXAMPLE OF A CALLING PROGRAM ON CDC 172/720/730
C     -----------------------------------------------
```

```
C
C    FILE ASSUMPTIONS:
C        - TAPE1    =  WORK FILE REQUIRED BY ZRESP.  NWRKFL=1.
C        - TAPE10   =  FORMA FILE (FOR INPUT DATA).  NOTE THAT
C                      MORE THAN ONE FORMA FILE MAY BE NECESSARY
C                      IF INPUT DATA IS RECORDED ON SEVERAL FORMA
C        - TAPE11   =  SEQUENTIAL FILE (FOR OUTPUT DATA).  NWFILE=11.
C
C        PROGRAM SCPRES (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,
C      +                   TAPE1,TAPE10,TAPE11)
C
C     1 CALL START
C                                                   CALL TIMCHK (6HTBEGIN)
C                                                   CALL TIMCHK (6HSCPRES)
C       CALL ZSCPRES
C                                                   CALL TIMCHK (6HSCPRES)
C                                                   CALL TIMCHK (6HTPRINT)
C       GO TO 1
C       END
C
C
C
C    INPUT FORM
C    ----------
C        CALLING PROGRAM MUST CALL START
C
C        READ    NWFILE,NWRKFL,NWRITE,NRESOFL     (4I5)
C        READ    STARTT, ENDT, DELTAT              (3E10.0)
C        READ    GAMMA, BETA                       (2E10.0)
C        READ    EPSILON                           (E10.0)
C        READ    NPAY                              (I5)
C        READ    NF, NE, IF, NP                    (4I5)
C        READ    NDAMPB,NDAMPI,NDAMPP              (3I5)
C        CALL ZREAD (FREQB)
C        IF (NDAMPB .EQ. 0)  GO TO 1
C        CALL READ (DAMPB,NR1,NC1,1,KB)
C        GO TO 2
C      1 READ    ZETAB                             (E10.0)
C      2 CALL ZREAD (FREQI)
C        IF (NDAMPI .EQ. 0)  GO TO 3
C        CALL READ (DAMPI,NR2,NC2,1,KI)
C        GO TO 4
C      3 READ    ZETAI                             (E10.0)
C      4 CALL ZREAD (FREQP)
C        IF (NDAMPP .EQ. 0)  GO TO 5
C        CALL READ (DAMPP,NR2,NC2,1,KP)
C        GO TO 6
C      5 READ    ZETAP                             (E10.0)
C      6 CALL ZREAD (B2)
C        CALL ZREAD (P2)
C        CALL READ (QNBO,NR3,NC3,1,KB)
C        CALL READ (QNBDO,NR4,NC4,1,KB)
C        CALL READ (QIBO,NR5,NC5,1,KI)
C        CALL READ (QIBDO,NR6,NC6,1,KI)
C        CALL READ (QNPO,NR7,NC7,1,KP)
C        CALL READ (QNPDO,NR8,NC8,1,KP)
C        RETURN
C
C
C    DEFINTITION OF INPUT VARIABLES
C    ------------------------------
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C               B2  =  THE INTERFACE/BOOSTER MASS COUPLING MATRIX IN
C                      PARTITION-LOGIC.  SIZE (NB,IF).
C             BETA  =  INPUT PARAMETER FOR NEWMARK-CHAN-BETA NUMERICAL
C                      INTEGRATION TECHNIQUE.  A GOOD VALUE IS BETA= 0.25
C            DAMPB  =  BOOSTER MODAL DAMPING MATRIX IN DENSE-LOGIC.
C                      DIAGONAL MATRIX INPUT AS A ROW VECTOR.
C                      SIZE (1,NB).
C            DAMPI  =  INTERFACE MODAL DAMPING MATRIX IN DENSE-LOGIC.
C                      DIAGONAL MATRIX INPUT AS A ROW VECTOR.
C                      SIZE (1,IF).
C            DAMPP  =  PAYLOAD MODAL DAMPING MATRIX IN DENSE-LOGIC.
C                      DIAGONAL MATRIX INPUT AS A ROW VECTOR.
C                      SIZE (1,NP)
C           DELTAT  =  TIME STEP FOR INTEGRATION ROUTINE.
C             ENDT  =  STOPPING TIME FOR THE NUMERICAL INTEGRATION
C                      ROUTINE.
C          EPSILON  =
C            FREQB  =  TRUNCATED, CANTILEVERED BOOSTER FREQUENCY VECTOR
C                      IN PARTITION-LOGIC.  SIZE (1,NB)
C            FREQI  =  INTERFACE FREQUENCY VECTOR IN PARTITION-LOGIC.
C                      SIZE (1,IF)
C            FREQP  =  TRUNCATED, CANTILEVERED PAYLOAD FREQUENCY VECTOR
C                      IN PARTITION-LOGIC.  SIZE (1,NP)
C            GAMMA  =  INPUT PARAMETER FOR NEWMARK-CHAN-BETA NUMERICAL
C                      INTEGRATION TECHNIQUE.  A GOOD VALUE IS GAMMA=0.5
C               IF  =  NUMBER OF INTERFACE DOFS.
C               NB  =  NUMBER OF TRUNCATED, CANTILERED BOOSTER MODES.
C           NDAMPB  =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                         BOOSTER MODAL DAMPING MATRIX.
C                   =  1  FOR A VARIABLE VALUE OF MODAL DAMPING IN THE
C                         BOOSTER MODAL DAMPING MATRIX.
C                         (MUST INPUT DAMPB FOR NDAMPB = 1).
C           NDAMPI  =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                         INTERFACE MODAL DAMPING MATRIX.
C                      1  FOR A VARIABLE VALUE OF INTERFACE MODAL
C                         DAMPING.  MUST INPUT DAMPI WHEN NDAMPI=1.
C           NDAMPP  =  0  FOR A CONSTANT VALUE OF MODAL DAMPING IN THE
C                         PAYLOAD MODAL DAMPING MATRIX.
C                   =  1  FOR A VARIABLE VALUE OF MODAL DAMPING.
C                         MUST INPUT DAMPP FOR NDAMPP = 1.
C               NF  =  NUMBER OF DOFS IN BOOSTER WHERE FORCES ARE APPLIED.
C           NFORFL  =  LOGICAL FILE NUMBER CONTAINING THE INTERPOLATED
C                      FORCE DATA.  THIS DATA IS SEQUENTIAL.
C               NP  =  TOTAL NUMBER OF TRUNCATED, CANTILEVERED PAYLOAD
C                      MODES.
C           NWFILE  =  LOGICAL FILE NUMBER FOR OUTPUT DATA.
C           NWRITE  =  0  RESULTS ARE NOT PRINTED ON PAPER.
C                   =  1  RESULTS ARE PRINTED ON PAPER.
C           NWRKFL  =  LOGICAL FILE NUMBER FOR WORK FILE.
C               P2  =  THE PAYLOAD/INTERFACE COUPLING MASS MATRIX.
C                      SIZE (IF,NP).
C             QIBO  =  DENSE-LOGIC VECTOR OF INITIAL MODAL DISPLACEMENTS
C                      OF THE INTERFACE DOFS AT STARTT.  SIZE (1,IF)
C            QIBDO  =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                      OF THE INTERFACE DOFS AT STARTT.  SIZE (1,IF).
C             QNBO  =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL
C                      DISPLACEMENTS OF THE BOOSTER DOFS AT THE STARTT.
C                      SIZE (1,NB).
C            QNBDO  =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                      OF THE BOOSTER DOFS AT STARTT.  SIZE (1,NB)
C             QNPO  =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL
C                      DISPLACEMENTS OF THE PAYLOAD DOFS.  SIZE (1,NP).
```

```
C           QNPDO  =  DENSE-LOGIC VECTOR OF THE INITIAL MODAL VELOCITIES
C                     OF THE PAYLOAD DOFS AT STARTT.  SIZE (1,NP).
C           ZETAB  =  VALUE OF BOOSTER MODAL DAMPING FOR ALL DOFS WHEN
C                     NDAMPB = 0.
C           ZETAI  =  VALUE OF INTERFACE DAMPING FOR ALL DOFS WHEN
C                     NDAMPI =0.
C           ZETAP  =  VALUE OF PAYLOAD MODAL DAMPING FOR ALL DOFS WHEN
C                     NDAMPP = 0.
C
C
C                                                          ORIGINAL PAGE IS
C    A.  DIMENSION, COMMON, DATA, FORMAT                   OF POOR QUALITY
C    ---------------------------------
      COMMON     /NITNOT/   NIT,NOT
      COMMON     /LSTRT4/   NLINE,NLPP
      DIMENSION D1(500),D2(200),D3(500),D4(500),D5(200),D6(500),D7(500),
     +           D8(200),D9(500),
     +           FB(500),FI(200),FP(500),
     +           DAMPI(200),
     +           QNB(500),QNBD(500),QNBDD(500),
     +           QNBO(500),QNBDO(500),QNBDDO(500),
     +           QNBRO(500),QNBRDO(500),QNBRDDO(500),
     +           QNBR(500),QNBRD(500),QNBRDD(500),
     +           QNBN(500),QNBND(500),QNBNDD(500),
     +           QIBO(200),QIBDO(200),QIBDDO(200),
     +           QIB(200),QIBD(200),QIBDD(200),
     +           XIBN(200),XIBND(200),XIBNDD(200),
     +           QIBRO(200),QIBRDO(200),QIBRDDO(200),
     +           QIBR(200),QIBRD(200),QIBRDD(200),
     +           QIBN(200),QIBND(200),QIBNDD(200),
     +           QNPO(500),QNPDO(500),QNPDDO(500),
     +           QNP(500),QNPD(500),QNPDD(500),
     +           OMB2(500),OMI2(200),OMP2(500),
     +           IVEC(200)
C
      DATA       KB,KI,KP             /500,200,500/
      DATA       BUF                  /-1.E50/
C
 1000 FORMAT (10I5)
 1100 FORMAT (10E10.0)
 1200 FORMAT (//9X,8H TIME = ,F10.6)
 1250 FORMAT (//10X,*TIME = *,F10.6,4X,*(CONTINUED)*)
 1300 FORMAT (//9X,15H APPLIED FORCES /(10X,1P5E16.8))
 1400 FORMAT (//5X,*INTERFACE RESPONSE  (MODAL COORDINATES)*,/,
     +           5X,*--------- --------  ------------------*,
     +          //9X,4H ROW,6X,13H ACCELERATION,8X,9H VELOCITY,
     +             //,(10X,I3,1P3E20.8))
 1450 FORMAT (//5X,*INTERFACE RESPONSE (MODAL COORDINATES)*,/,
     +           5X,*--------- -------- ------------------*,//,
     +          27X,*TOTAL*,32X,*RESIDUAL*,31X,*NOMINAL*,/,
     +           2X,*ROW*,5X,3(*ACC*,10X,*VEL*,10X,*DIS*,10X),/,
     +           (2X,I3,3(1P3E13.4,* / *)))
 1500 FORMAT (//4X,* NON-INTERFACE RESPONSE  (MODAL COORDINATES)*,/,5X,
     +          13(1H-),1X,8(1H-),2X,19(1H-),//9X,* ROW*,7X,
     +          *ACCELERATION*,9X,*VELOCITY*,11X,*DISPLACEMENT*//
     +          ,(10X,I3,1P3E20.8))
 2001 FORMAT (/////,50X,*INPUT PARAMETERS*,/,50X,16(1H-),///,
     +          48X,*NWFILE  =*,I5,/,48X,*NWRKFL  =*,I5,/,
     +          47X,*NRESOFL  =*,I5,//,
     +          30X,*DATA IS WRITTEN ON PAPER EVERY *,I5,* TIME STEPS*)
 2002 FORMAT (//,48X,*STARTT  =*,F10.6,/,50X,*ENDT  =*,F10.6,/,
     +            48X,*DELTAT  =*,F10.6)
```

```
 2003 FORMAT (//,30X,*PARAMETERS FOR NEWMARK-CHAN-BETA INTEGF TION*,
     +            * ROUTINE*,//,49X,*GAMMA  =*,F10.6,/,
     +            50X,*BETA  =*,F10.6)
 2004 FORMAT (/30X,*VALUE OF BOOSTER MODAL DAMPING IS CONSTANT*,
     +           5X,*(NDAMPB = 1)*)
 2005 FORMAT (/30X,*VALUE OF PAYLOAD MODAL DAMPING IS A CONSTANT*,
     +           5X,*(NDAMPP = 1)*)
 2006 FORMAT (//30X,*NUMBER OF FORCES APPLIED TO BOOSTER  =*,I5,//,
     +            32X,*NUMBER OF TRUNCATED BOOSTER MODES  =*,I5,//,
     +            41X,*NUMBER OF INTERFACE DOFS  =*,I5,//,
     +            32X,*NUMBER OF TRUNCATED PAYLOAD MODES  =*,I5,//,
     +            47X,*NUMBER OF PAYLOADS  =*,I5)
 3005 FORMAT (/////,20X,* FOR TIME = *,F12.4,* DIRECT BASE DRIVE USED*)
C
C     ******************************
C       BEGINNING OF PROGRAM
C     ******************************
C
C   B.  READ INPUT DATA
C   -------------------
      READ (NIT,1000)    NWFILE,NWRKFL,NWRITE,NRESOFL
      READ (NIT,1100)    STARTT,ENDT,DELTAT
      READ (NIT,1100)    GAMMA,BETA
      READ (NIT,1100)    EPSILON
      READ (NIT,1000)    NPAY
      READ (NIT,1000)    NF,NB,IF,NP
      READ (NIT,1000)    NDAMPB,NDAMPI,NDAMPP
C
      CALL PAGEHD
      WRITE (NOT,2001)    NWFILE,NWRKFL,NRESOFL,NWRITE
      WRITE (NOT,2002)    STARTT,ENDT,DELTAT
      WRITE (NOT,2003)    GAMMA,BETA
      IF (NDAMPB .EQ. 0)   WRITE(NOT,2004)
      IF (NDAMPP .EQ. 0)   WRITE(NOT,2005)
      WRITE (NOT,2006)    NF,NB,IF,NP,NPAY
C
      CALL ZWRKFL (NWRKFL)
C
C   C.  CALCULATION OF CONSTANTS
C   ----------------------------
      C0=DELTAT*DELTAT
      C1=GAMMA*DELTAT
      C2=BETA*C0
      C3=(1.-GAMMA)*DELTAT
      C4=(0.5-BETA)*C0
C
C   D.  CALCULATION OF THE VECTORS -D1INV, D4, AND D7
C   -------------------------------------------------
                                              CALL TIMCHK (6HDVECS )
      CALL ZREAD (FREQB)
      CALL ZTOD (FREQB,QNB,1,NB,1,KB)
      IF (NDAMPB .EQ. 0)   GO TO 10
      CALL READ (QNBD,NR1,NC1,1,KB)
C
      DO 20 I=1,NB
      QNBD(I)=12.56637061*QNBD(I)*QNB(I)
      OMB2(I)=39.4784176*QNB(I)*QNB(I)
   20 CONTINUE
      GO TO 30
   10 CONTINUE
      READ (NIT,1100) ZETAB
      DO 40 I=1,NB
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      QNBD(I)=12.56637061*ZETAB*QNB(I)
      OMB2(I)=39.4784176*QNB(I)*QNB(I)
   40 CONTINUE
   30 CONTINUE
C
      DO 50 I=1,NB
      D1(I)=-1./(1.+C1*QNBD(I)+C2*OMB2(I))
      D4(I)=QNBD(I)+DELTAT*OMB2(I)
      D7(I)=C3*QNBD(I)+C4*OMB2(I)
   50 CONTINUE
C
C    E.  CALCUATION OF THE VECTORS D2, D5, AND D8
C    ------------------------------------------
      CALL ZREAD (FREQI)
      CALL ZTOD (FREQI,QIB,1,IF,1,KI)
      IF (NDAMPI .EQ. 0)   GO TO 55
      CALL READ (DAMPI,NR3,NC3,1,KI)
      DO 52 I=1,IF
      DAMPI(I)=12.56637061*DAMPI(I)*QIB(I)
      OMI2(I)=39.4784176*QIB(I)*QIB(I)
   52 CONTINUE
      GO TO 58
   55 READ (NIT,1100)  ZETAI
      DO 56 I=1,IF
      DAMPI(I)=12.56637061*ZETAI*QIB(I)
      OMI2(I)=39.4784176*QIB(I)*QIB(I)
   56 CONTINUE
   58 CONTINUE
      DO 59 I=1,IF
      D2(I)=1.+C1*DAMPI(I)+C2*OMI2(I)
      D5(I)=DAMPI(I)+DELTAT*OMI2(I)
      D8(I)=C3*DAMPI(I)+C4*OMI2(I)
   59 CONTINUE
C
C
C    F.  CALCULATION OF THE VECTORS -D3INV, D6, AND D9
C    ------------------------------------------------
      CALL ZREAD (FREQP)
      CALL ZTOD (FREQP,QNP,1,NP,1,KP)
C
      IF (NDAMPP .EQ. 0)   GO TO 60
      CALL READ (QNPD,NR2,NC2,1,KP)
C
      DO 70 I=1,NP
      QNPD(I)=12.56637061*QNPD(I)*QNP(I)
      OMP2(I)=39.4784176*QNP(I)*QNP(I)
   70 CONTINUE
      GO TO 80
   60 CONTINUE
      READ(NIT,1100) ZETAP
      DO 90 I=1,NP
      QNPD(I)=12.5663706 * ZETAP * QNP(I)
      OMP2(I) = 39.4784176*QNP(I)*QNP(I)
   90 CONTINUE
C
   80 CONTINUE
C
      DO 100 I=1,NP
      D3(I)=-1./(1.+C1*QNPD(I)+C2*OMP2(I))
      D6(I)=QNPD(I)+DELTAT*OMP2(I)
      D9(I)=C3*QNPD(I)+C4*OMP2(I)
  100 CONTINUE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C                                                   CALL TIMCHK (6HDVECS )
C
C   G.  CALCULATION OF A1, A2, AND A3 MATRICES
C   ------------------------------------------
                                                    CALL TIMCHK (6HA123  )
      CALL ZREAD (B2)
      CALL ZREAD (P2)
C
      CALL ZMULTDD (B2,D1,NB,A2)
      CALL ZMULTDD (P2,D3,NP,A3)
      CALL ZTRANS (B2,B2T)
      CALL ZTRANS (P2,P2T)
      CALL ZMULT (A2,B2T,A2B2T)
      CALL ZMULT (A3,P2T,A3P2T)
      CALL ZAABB (1.,A2B2T,1.,A3P2T,HELP2)
      CALL ZABDI (HELP2,D2,IF,A1I)
      CALL ZINV3 (A1I,A1,1)
      CALL ZTRANS (A2,A2T)
      CALL ZTRANS (A3,A3T)
                                                    CALL TIMCHK (6HA123  )
C
C   H.  CALCULATION OF MPI2 , KPI2, AND PHIINV=(PHIIB)T*BPM2
C   -------------------------------------------------------
      CALL ZREAD (PHIIB)
                                                        NERROR=1
      CALL ZSIZE (PHIIB,NRI,NCI)
      IF (NRI .NE. NCI .OR. NRI .NE. IF)                GO TO 999
C
      CALL ZREAD (BPM2)
C
      CALL ZZERO (MPI,IF,IF)
      CALL ZZERO (KPI,IF,IF)
C
      DO 122 I=1,NPAY
C
      READ (NIT,1000)     IFP
      CALL ZREAD (PM2)
      CALL ZREAD (PK2)
      CALL READIM (IVEC,NRP,NCP,1,KI)
C
      CALL ZSIZE (PM2,NRM,NCM)
      CALL ZSIZE (PK2,NRK,NCK)
                                                        NERROR=2
      IF (NRM .NE. NCM .OR. NRK .NE. NCK)               GO TO 999
                                                        NERROR=3
      IF (NRM .NE. NRK .OR. NRM .NE. IFP)               GO TO 999
                                                        NERROR=4
      IF (NCP .NE. IFP)                                 GO TO 999
C
      CALL ZRVAD (1.0,PM2,IVEC,IVEC,NCP,NCP,MPI)
      CALL ZRVAD (1.0,PK2,IVEC,IVEC,NCP,NCP,KPI)
C
  122 CONTINUE
C
      CALL ZTRANS (PHIIB,PHIIBT)
      CALL ZMULT (PHIIBT,MPI,MPI2)
      CALL ZMULT (PHIIBT,KPI,KPI2)
C
      CALL ZMULT (PHIIBT,BPM2,PHIINV)
C
C              CHECK THE ACCURACY OF THE INVERSE
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C             ------------------------------
C             CALL ZMULT (PHIINV,PHIIB,UNITY)
C             CALL ZWRITE (UNITY,6HUNITY )
C
C
C  I.  CALCULATION OF INITIAL ACCELERATIONS - QNBDDO, QIBDDO, QNPDDO
C  -----------------------------------------------------------------
                                              CALL TIMCHK (6HINITL )
      CALL ZMULT (B2,B2T,HELP3)
      CALL ZMULT (P2,P2T,HELP4)
      CALL ZAABB (-1.,HELP4,-1.,HELP3,HELP5)
      DO 125 I=1,IF
      FI(I)=1.
  125 CONTINUE
      CALL ZABDI (HELP5,FI,IF,AI)
      CALL ZINV3 (AI,A,1)
C
C          I1.  READ IN INITIAL PAYLOAD DISPLACEMENTS AND VELOCITIES
C          -----------------------------------------------------
      CALL READ (QNPO,NR7,NC7,1,KP)
      CALL READ (QNPDO,NR8,NC8,1,KP)
C
      CALL ZERO (QNBRO,1,NB,1)
      CALL ZERO (QNBRDO,1,NB,1)
      CALL ZERO (QIBRO,1,IF,1)
      CALL ZERO (QIBRDO,1,IF,1)
C
C          I2.  READ THRU NRESOFL UNTIL THE STARTT IS REACHED
C          ---------------------------------------------
      REWIND NRESOFL
      NTP=(ENDT-STARTT)/DELTAT+1.1
      NW=NWRITE
      NBPIF=NB+IF
C
      READ (NRESOFL)  IRUNNO,IDATE,STARTT2,ENDT2,DELTAT2,NB2,IF2,
     +                 (BUFCH,I=1,10)
C
                                                  NERROR=7
      IF (NB2 .NE. NB)                            GO TO 999
                                                  NERROR=8
      IF (IF2 .NE. IF)                            GO TO 999
                                                  NERROR=4
      IF (DELTAT .NE. DELTAT2)                    GO TO 999
                                                  NERROR=5
      IF (STARTT2 .LT. STARTT)                    GO TO 999
                                                  NERROR=6
      IF (STARTT .EQ. STARTT2)   GO TO 180
C
      DELST=STARTT2-STARTT
      NTS=DELST/DELTAT
                                                  NERROR=6
      IF ((NTS*DELTAT) .NE. DELST)                GO TO 999
C
      DO 160 K=1,NTS
      READ (NRESOFL)   BUF,((BUF,I=1,NBPIF),J=1,3),BUF
  160 CONTINUE
C
  180 CONTINUE
C
      READ (NRESOFL) TO,(QNBN(I),I=1,NB),  (XIBN(I),I=1,IF),
     +                  (QNBND(I),I=1,NB), (XIBND(I),I=1,IF),
     +                  (QNBNDD(I),I=1,NB),(XIBNDD(I),I=1,IF),BUF
```

```
C
C          I3.  CALCULATE THE NOMINAL INTERFACE MODAL RESPONSE
C          ------------------------------------------------
      CALL ZMULTCD (PHIINV,XIBN,QIBN,IF,KI)
      CALL ZMULTCD (PHIINV,XIBND,QIBND,IF,KI)
      CALL ZMULTCD (PHIINV,XIBNDD,QIBNDD,IF,KI)
C
C          I4.  CALCULATE THE BASE DRIVE FORCING FUNCTONS
C          ------------------------------------------
      CALL ZMULTCD (P2T,QIBNDD,FP,IF,KP)
C
      DO 220 I=1,NP
      FP(I)=-FP(I)-QNPD(I)*QNPDO(I)-OMP2(I)*QNPO(I)
  220 CONTINUE
C
      CALL ZMULTCD (MPI2,XIBNDD,FI,IF,KI)
      CALL ZMULTCD (KPI2,XIBN,XIBNDD,IF,KI)
      CALL ZMULTCD (P2,FP,QNPDD,NP,KP)
C
      DO 240 I=1,IF
      FI(I)=-FI(I)-XIBNDD(I)-DAMPI(I)*QIBND(I)-QNPDD(I)
  240 CONTINUE
C
C          I5.  CALCULATION OF THE RESPONSE AT T=STARTT
C          ------------------------------------------
C               I5A.  CALCULATION OF BASE DRIVE INTERFACE RESPONSE
C               ---------------------------------------------
C
      CALL ZMULTCD (A,FI,QIBRDDO,IF,KI)
C
      DO 250 I=1,IF
      QIBDDO(I)=QIBRDDO(I)+QIBNDD(I)
      QIBDO(I)=QIBND(I)
      QIBO(I)=QIBN(I)
  250 CONTINUE
C
C               I5B.  CALCULATION OF BASE DRIVE BOOSTER RESPONSE
C               -------------------------------------------
      CALL ZMULTCD (B2T,QIBRDDO,QNBRDDO,IF,KB)
C
      DO 260 I=1,NB
      QNBRDDO(I)=-QNBRDDO(I)
  260 CONTINUE
C
C               I5C.  CALCULATION OF PAYLOAD RESPONSE
C               ---------------------------------
      CALL ZMULTCD (P2T,QIBRDDO,QNPDDO,IF,KP)
C
      DO 280 I=1,NP
      QNPDDO(I)=FP(I)-QNPDDO(I)
  280 CONTINUE
C
                                                CALL TIMCHK (6HINITL )
C
C    *****************************************************************
C
                                                CALL TIMCHK (6HRESP   )
C
C    J.  RESPONSE LOOP
C    ---------------
      REWIND  NWFILE
      WRITE (NWFILE)  IRUNNO,IDATE,STARTT,ENDT,DELTAT,IF,NP,(BUF,I=1,10)
```

```
      EPS2=EPSILON*EPSILON
      IDBD=0
      IDBDP=1
C
      DO 600 ITP=1,NTP
C
      READ (NRESOFL)    T,(QNBN(I),I=1,NB),  (XIBN(I),I=1,IF),
     +                    (QNBND(I),I=1,NB), (XIBND(I),I=1,IF),
     +                    (QNBNDD(I),I=1,NB),(XIBNDD(I),I=1,IF),BUF
C
C          J1.  CONVERT DISCRETE NOMINAL BOOSTER RESPONSE TO
C                          NORMAL COORDINATES
      CALL ZMULTCD (PHIINV,XIBNDD,QIBNDD,IF,KI)
      CALL ZMULTCD (PHIINV,XIBND,QIBND,IF,KI)
C
C         CALCULATE THE BASE DRIVE FORCING FUNCTIONS
C         ------------------------------------------
      CALL ZMULTCD (MPI2,XIBNDD,FI,IF,KI)
      CALL ZMULTCD (KPI2,XIBN,QIBR,IF,KI)
C
      CALL ZMULTCD (P2T,QIBNDD,FP,IF,KP)
C
      DO 300 I=1,IF
      FI(I)=-FI(I)-DAMPI(I)*QIBND(I)-QIBR(I)
  300 CONTINUE
C
      IF (IDBD .EQ. 1)    GO TO 340
C
C          IF SIGNIFICANT ADD IN THE FEEDBACK TERMS TO THE FORCING TERMS
C          -------------------------------------------------------------
      DO 320 I=1,NB
      FB(I)=-D4(I)*QNBRDO(I)-D7(I)*QNBRDDO(I)-OMB2(I)*QNBRO(I)
  320 CONTINUE
C
      CALL ZMULTCD (A2,FB,QIBRDD,NB,KI)
C
      DO 322 I=1,IF
      FI(I)=FI(I)-D5(I)*QIBRDO(I)-D8(I)*QIBRDDO(I)-OMI2(I)*QIBRO(I)
     +                    +QIBRDD(I)
  322 CONTINUE
C
      DO 324 I=1,NP
      FP(I)=FP(I)+D6(I)*QNPDO(I)+D9(I)*QNPDDO(I)+OMP2(I)*QNPO(I)
  324 CONTINUE
C
C               CALCULATE THE RESIDUAL INTERFACE ACCELERATIONS
C            AND COMPARE THE SIZE WITH THE NOMINAL ACCELERATIONS
C            ---------------------------------------------------
  340 CALL ZMULTCD (A3,FP,QIBRD,NP,KI)
C
      DO 350 I=1,IF
      FI(I)=FI(I)-QIBRD(I)
  350 CONTINUE
C
      CALL ZMULTCD (A1,FI,QIBRDD,IF,KI)
C
      AMAGN=0.
      AMAGR=0.
      IDBDP=IDBD
      IDBD=1
C
      DO 380 I=1,IF
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      AMAGN=AMAGN+QIBNDD(I)*QIBNDD(I)
      AMAGR=AMAGR+QIBRDD(I)*QIBRDD(I)
  380 CONTINUE
C
      IF (AMAGN .EQ. 0.0)       GO TO 385
      IF ((AMAGR/AMAGN) .LE. EPS2)     GO TO 450
C
C        CALCULATE THE COUPLED BASE DRIVE RESPONSE
C                (FEEDBACK IS SIGNIFICANT)
C        -----------------------------------------
  385 IDBD=0
C
      CALL ZMULTCD (A2T,QIBRDD,QNBRDD,IF,KB)
      CALL ZMULTCD (A3T,QIBRDD,QNPDD,IF,KP)
C
      DO 390 I=1,NP
      QNPDD(I)=QNPDD(I)+D3(I)*FP(I)
  390 CONTINUE
C
      IF (IDBDP .EQ. 1)       GO TO 420
C
      DO 400 I=1,NB
      QNBRDD(I)=QNBRDD(I)-D1(I)*FB(I)
  400 CONTINUE
C
      GO TO 450
C
  420 DO 430 I=1,NB
      QNBRDD(I)=-D1(I)*FB(I)
  430 CONTINUE
C
C
C          J4.  WRITE ANSWERS ON NWFILE FOR LATER USE
C          ----------------------------------------
  450 WRITE (NWFILE)  TO,(QIBO(I),I=1,IF),  (QNPO(I),I=1,NP)
     +                   ,(QIBDO(I),I=1,IF), (QNPDO(I),I=1,NP)
     +                   ,(QIBDDO(I),I=1,IF),(QNPDDO(I),I=1,NP),BUF
C
C          J5.  SEE IF DATA SHOULD BE PRINTED
C          ----------------------------------
      IF (ITP .LT. NTP .AND. NW .LT. NWRITE) GO TO 500
                                          CALL TIMCHK (6HWRITE ).
      CALL PAGEHD
      NLINE=13
      WRITE (NOT,1200)  TO
      NXSI=1
      NXEI=IF
      IF (NXEI .GT. (NLPP-NLINE))     NXEI=NLPP-NLINE
  470 WRITE (NOT,1400) (I,QIBDDO(I),QIBDO(I),QIBO(I),I=NXSI,NXEI)
      NLINE=NLINE+NXEI-NXSI+1
      IF (IF .EQ. NXEI) GO TO 480
      CALL PAGEHD
      WRITE (NOT,1250)  TO
      NLINE=13
      NXSI=NXEI+1
      NXEI=IF
      IF ((NXEI-NXSI) .GT. (NLPP-NLINE))    NXEI=NXSI+NLPP-NLINE
      GO TO 470
  480 CONTINUE
      NXSP=1
      NXEP=NP
      NLINE=NLINE+13
```

```
      IF ((NLINE+10) .LT. NLPP)    GO TO 485
      CALL PAGEHD
  485 WRITE (NOT,1250)  TO
      NLINE=NLINE+10
      IF (NXEP .GT. (NLPP-NLINE))   NXEP=NLPP-NLINE
  490 WRITE (NOT,1500) (I+IF,QNPDDO(I),QNPDO(I),QNPO(I),I=NXSP,NXEP)
      IF (NP .EQ. NXEP) GO TO 495
      NXSP=NXEP+1
      NXEP=NP
      CALL PAGEHD
      WRITE (NOT,1250)  TO
      NLINE=10
      IF ((NXEP-NXSP) .GT. (NLPP-NLINE)) NXEP=NXSP+NLPP-NLINE
      GO TC 490
C
  495 NW=0
                                                 CALL TIMCHK (6HWRITE )
  500 NW=NW+1
C
C
C            J6.  CALCULATE VELOCITY AND DISPLACEMENT TERMS
C                         AND REASSIGN THE VALUES
C            ---------------------------------------------------
      TO=T
C
      IF (IDBD .EQ. 1)   GO TO 540
C
C       CALCULATE COUPLED BASE DRIVE RESPONSE
C       -------------------------------------
                                                 CALL TIMCHK (6HCOUPLE)
      DO 520 I=1,NB
      QNBRO(I)=QNBRO(I)+DELTAT*QNBRDO(I)+C4*QNBRDDO(I)+C2*QNBRDD(I)
      QNBRDO(I)=QNBRDO(I)+C3*QNBRDDO(I)+C1*QNBRDD(I)
      QNBRDDO(I)=QNBRDD(I)
      QNBO(I)=QNBN(I)+QNBRO(I)
      QNBDO(I)=QNBND(I)+QNBRDO(I)
      QNBDDO(I)=QNBNDD(I)+QNBRDDO(I)
  520 CONTINUE
C
      DO 522 I=1,IF
      QIBRO(I)=QIBRO(I)+DELTAT*QIBRDO(I)+C4*QIBRDDO(I)+C2*QIBRDD(I)
      QIBRDO(I)=QIBRDO(I)+C3*QIBRDDO(I)+C1*QIBRDD(I)
      QIBRDDO(I)=QIBRDD(I)
      QIBO(I)=QIBO(I)+DELTAT*QIBDO(I)+C4*QIBDDO(I)
     +                     +C2*(QIBNDD(I)+QIBRDDO(I))
      QIBDDO(I)=QIBNDD(I)+QIBRDDO(I)
      QIBDO(I)=QIBND(I)+QIBRDO(I)
  522 CONTINUE
C
      DO 524 I=1,NP
      QNPO(I)=QNPO(I)+DELTAT*QNPDO(I)+C4*QNPDDO(I)+C2*QNPDD(I)
      QNPDO(I)=QNPDO(I)+C3*QNPDDO(I)+C1*QNPDD(I)
      QNPDDO(I)=QNPDD(I)
  524 CONTINUE
                                                 CALL TIMCHK (6HCOUPLE)
C
      GO TO 600
C
C          CALCULATE THE DIRECT BASE DRIVE RESPONSE
C          ----------------------------------------
  540 CONTINUE
                                                 CALL TIMCHK (6HDIRECT)
```

```
C
      DO 542 I=1,NB
      QNBO(I)=QNBN(I)
      QNBDO(I)=QNBND(I)
      QNBDDO(I)=QNBNDD(I)
  542 CONTINUE
C
      DO 544 I=1,IF
      QIBO(I)=QIBO(I)+DELTAT*QIBDO(I)+C4*QIBDDO(I)+C2*QIBNDD(I)
      QIBDO(I)=QIBND(I)
      QIBDDO(I)=QIBNDD(I)
  544 CONTINUE
C
      DO 546 I=1,NP
      QNPDD(I)=D3(I)*FP(I)
      QNPO(I)=QNPO(I)+DELTAT*QNPDO(I)+C4*QNPDDO(I)+C2*QNPDD(I)
      QNPDO(I)=QNPDO(I)+C3*QNPDDO(I)+C1*QNPDD(I)
      QNPDDO(I)=QNPDD(I)
  546 CONTINUE
                                                CALL TIMCHK (6HDIRECT)
C
  600 CONTINUE
                                                CALL TIMCHK (6HRESP  )
C
C    K.  EXIT
C    --------
      RETURN
C
  999 CONTINUE
C
      CALL ZZBOMB (6HZRESP ,NER.2")
C
      END
```

```
C
C
C        ********************************
C        *                              *
C        *            ZABDI             *
C        *                              *
C        ********************************
C
C
      SUBROUTINE ZABDI (MA,B,NRB,MZ)
C
      COMMON    /LZWRKF/     NWRK,INDWRK(100),MWRKSP(10)
      COMMON    /LZ1/        MHA(10),INDRPA(201),INDCPA(201),SA(30,30)
      COMMON    /LZ2/        MHZ(10),INDRPZ(201),INDCPZ(201),SZ(30,30)
      DIMENSION   B(1)
      DATA        KRCPRT     /30/
C
C      SUBROUTINE ZABDI ADDS A PARTITION-LOGIC MATRIX (A) TO A DENSE-
C      LOGIC COLUMN VECTOR (B) WHICH REPRESENTS THE ELEMENTS OF A
C      DIAGONAL MATRIX.
C
C        ARGUMENT DEFINITION
C        -------------------
C         MA - INPUT PARTITION-LOGIC MATRIX (A).
C          B - INPUT DENSELOGIC COLUMN VECTOR (B).  (B) REPRESENTS
C              THE ELEMENTS OF A DIAGONAL MATRIX.
C        NRB - NUMBER OF ELEMENTS IN MATRIX (B).
C         MZ - OUTPUT PARTITION-LOGIC MATRIX (Z)=(A)+(B).
C
C      FIND POSITION OF MATRIX (Z) IN INDWRK
C      -------------------------------------
      IF (MZ .GE. 1 .AND. MZ .LE. 99) GO TO 5
      DO 2 MZ=1,99
      IF (INDWRK(MZ+1) .EQ. 0)    GO TO 5
    2 CONTINUE
                                                         NERROR=1
                                                         GO TO 999
C
C     READ MATRIX (A) HEADER
C     ----------------------
    5 CALL READRA (NWRK,MHA,10,MA,INDWRK,100)
      NRA=MHA(1)
      NCA=MHA(2)
      NRPA=MHA(3)
      NCPA=MHA(4)
      NRLA=MHA(5)
      NCLA=MHA(6)
                                                         NERROR=2
      IF (NRA .NE. NRB .OR. NCA .NE. NRB)                GO TO 999
C
C      FORM MATRIX (Z) HEADER
C      ----------------------
      DO 20 I=1,6
   20 MHZ(I)=MHA(I)
      DO 30 I=7,10
   30 MHZ(I)=0
C
C      ADD MATRICES (A) AND (B)
C      ------------------------
      CALL READRA (NWRK,INDRPA,201,1,MHA(9),2)
      CALL ZERO (INDRPZ,1,201,1)
C
```

```
      DO 200 IRPA=1,NRPA
      CALL READRA (NWRK,INDCPA,201,IRPA,INDRPA,201)
      CALL ZERO (INDCPZ,1,201,1)
      NCSA=KRCPRT
      NEL=NCSA*KRCPRT
C
      DO 100 JCPA=1,NCPA
      IF (JCPA .NE. NCPA)      GO TO 35
      NCSA=NCLA
      NEL =NCSA*KRCPRT
   35 IF (INDCPA(JCPA+1) .NE. 0) GO TO 40
      IF (JCPA .NE. IRPA)    GO TO 100
      CALL ZERO (SA,NCSA,NCSA,KRCPRT)
      GO TO 60
   40 CALL READRA (NWRK,SA,NEL,JCPA,INDCPA,201)
      IF (JCPA .NE. IRPA)      GO TO 90
   60 CONTINUE
      DO 80 I=1,NCSA
      SA(I,I)=SA(I,I)+B((IRPA-1)*KRCPRT+I)
   80 CONTINUE
   90 CALL WRITRA (NWRK,SA,NEL,JCPA,INDCPZ,201)
  100 CONTINUE
      CALL WRITRA (NWRK,INDCPZ,201,IRPA,INDRPZ,201)
  200 CONTINUE
      CALL WRITRA (NWRK,INDRPZ,201,1,MHZ(9),2)
      CALL WRITRA (NWRK,MHZ,10,MZ,INDWRK,100)
C
C     RESTORE MASTER INDEX ON NWRK
C     --------------------------
      CALL STINDX (NWRK,INDWRK,100)
      RETURN
C
  999 CALL ZZBOMB (6HZABDI ,NERROR)
      END
```

```
C
C
C                ***********************************
C                *                                 *
C                *            ZMULTCD              *
C                *                                 *
C                ***********************************
C
      SUBROUTINE ZMULTCD(MA,B,Z,NRB,KRZ)
      COMMON /LZWRKF/ NWRK, INDWRK(100),MWRKSP(10)
      COMMON /LZ1/ MHA(10),INDRPA(201),INDCPA(201),SA(30,30)
      DIMENSION B(1),Z(1)
      DATA KRCPRT /30/
C
C       HYBRID PARTITION-LOGIC,DENSE-LOGIC MULTIPICATION SUBROUTINE.
C       MULTIPLIES A PARTITION-LOGIC MATRIX (A), WITH A DENSE-LOGIC
C       COLUMN VECTOR (B).  RESULTS IN A DENSE-LOGIC COLUMN VECTOR (Z).
C       (A) * (B) = (Z).
C
C       DEVELOPED BY TG SHANAHAN.  FEB, 82.
C
C       SUBROUTINE ARGUMENTS
C       --------------------
C        MA   - INPUT PARTITION-LOGIC MATRIX (A).
C         B   - INPUT DENSE-LOGIC COLUMN VECTOR (B).
C         Z   - OUTPUT DENSE-LOGIC COLUMN VECTOR (Z)
C       NRB   - NUMBER OF ELEMENTS IN (B) AND COLUMNS IN (A).
C       KRZ   - DIMENSION SIZE OF (Z) IN THE CALLING PROGRAM.
C
C       COMMON EXPLANATIONS
C       -------------------
C       /LZWRKF/ - SHOULD HAVE BEEN INITIALIZED WITH SUBROUTINE ZWRKFL.
C                  DATA COMES IN AND GOES OUT.
C          /LZ1/ - WORK SPACE.  NO DATA COMES IN OR GOES OUT.
C
C       NERROR EXPLANATIONS
C       --------------------
C        1 = DIMENSION SIZE OF (Z) IS NOT SUFFICENTLY LARGE.
C        2 = MATRICES (A) AND (B) ARE NOT COMPATABLE.
C
C       READ IN MATRIX (A) HEADER
C       -------------------------
      CALL READRA (NWRK,MHA,10,MA,INDWRK,100)
      NRA=MHA(1)
      NCA=MHA(2)
      NRPA=MHA(3)
      NCPA=MHA(4)
      NRLA=MHA(5)
      NCLA=MHA(6)
C
                                                             NERROR=1
      IF (KRZ .LT. NRA)                                      GO TO 999
                                                             NERROR=2
      IF (NCA .NE. NRB)                                      GO TO 999
C
C       PERFORM MULTIPICATION
C       ---------------------
      CALL READRA(NWRK,INDRPA,201,1,MHA(9),2)
      CALL ZERO(Z,NRA,1,KRZ)
      DO 20 IRPA=1,NRPA
      NRSA=KRCPRT
      IF(IRPA.EQ.NRPA) NRSA=NRLA
```

```
      CALL READRA(NWRK,INDCPA,201,IRPA,INDRPA,201)
      DO 20 JCPA=1,NCPA
      IF(INDCPA(JCPA+1).EQ.0) GO TO 20
      NCSA=KRCPRT
      IF(JCPA.EQ.NCPA) NCSA=NCLA
      CALL READRA (NWRK,SA,KRCPRT*NCSA,JCPA,INDCPA,201)
      DO 10 I=1,NRSA
      NR=(IRPA-1)*KRCPRT+I
      DO 10 J=1,NCSA
      NC=(JCPA-1)*KRCPRT+J
      Z(NR)=Z(NR)+SA(I,J)*B(NC)
   10 CONTINUE
   20 CONTINUE
C
C        RESTORE MASTER INDEX ON NWRK
C        ----------------------------
      CALL STINDX (NWRK,INDWRK,100)
C
      RETURN
  999 CALL ZZBOMB(6HZMULTC,NERROR)
      END
```

```
C
C
C      ************************************
C      *                                  *
C      *             ZMULTDD              *
C      *                                  *
C      ************************************
C
C
      SUBROUTINE ZMULTDD(MA,D,NCD,MZ)
      COMMON /LZWRKF/ NWRK,INDWRK(100),MWRKSP(10)
      COMMON /LZ1/ MHA(10),INDRPA(201),INDCPA(201),SA(30,30)
      COMMON /LZ2/ MHZ(10),INDRPZ(201),INDCPZ(201),SZ(30,30)
      DIMENSION D(1)
      DATA KRCPRT/30/
C
C      HYBRID PARTITION-LOGIC, DENSE-LOGIC MATRIX MULTIPICATION
C      ROUTINE.   (A) * (D) = (Z)
C      WHERE, MATRIX (A) IS A PARTITION-LOGIC MATRIX,
C             MATRIX (D) IS A DIAGONAL MATRIX REPRESENTED AS A DENSE
C             COLUMN VECTOR,
C      AND, MATRIX (Z) IS THE PARTITION-LOGIC RESULT OF THE
C           MULTIPICATION.
C
C      DEVELOPED BY TG SHANAHAN, JAN 1982.
C
C      SUBROUTINE ARGUMENTS
C      --------------------
C      MA   = INPUT MATRIX (A) IN PARTITION-LOGIC.
C       D   = INPUT DENSE VECTOR WHICH CONTAINS THE DIAGONAL ELEMENTS
C             OF DIAGONAL MATRIX (D).
C     NCD   = NUMBER OF ROWS AND COLUMNS IN MATRIX (D).
C      MZ   = OUTPUT MATRIX (Z) IN PARTITION-LOGIC.
C
C      COMMON EXPLANATIONS
C      -------------------
C      /LZWRKF/  = SHOULD HAVE BEEN INITIALIZED WITH SUBROUTINE ZWRKFL.
C                  DATA COMES IN AND GOES OUT.
C      /LZ1/     = WORK SPACE.  NO DATA COMES IN OR GOES OUT.
C      /LZ2/     = WORK SPACE.  NO DATA COMES IN OR GOES OUT.
C
C      NERROR EXPLANATIONS
C      -------------------
C      1 = NUMBER OF MATRICES ON WORK FILE (NWRK) EXCEEDS LIMIT.
C      2 = MATRICES (A) AND (D) ARE NOT COMPATABLE.
C
C
C
C           FIND POSITION OF MATRIX (Z) IN INDWRK
C           -------------------------------------
      IF(MZ.GE.1.AND.MZ.LE.99) GO TO 5
      DO 2 MZ=1,99
      IF(INDWRK(MZ+1).EQ.0) GO TO 5
    2 CONTINUE
                                                      NERROR=1
                                                      GO TO 999
C
C           READ IN MATRIX (A) HEADER
C           ------------------------
    5 CALL READRA(NWRK,MHA,10,MA,INDWRK,100)
      NRA=MHA(1)
      NCA=MHA(2)
```

```
      NRPA=MHA(3)
      NCPA=MHA(4)
      NRLA=MHA(5)
      NCLA=MHA(6)
      IF(NCD.EQ.NCA) GO TO 7
                                                    NERROR=2
                                                    GO TO 999
C
C          FORM MATRIX (Z) HEADER
C          ---------------------
    7 DO 10 I=1,6
   10 MHZ(I)=MHA(I)
      DO 20 I=7,10
   20 MHZ(I)=0
C
C           MULTIPLY MATRICES (A) AND (D)
C           ----------------------------
      CALL READRA(NWRK,INDRPA,201,1,MHA(9),2)
      CALL ZERO(INDRPZ,1,201,1)
      NRSA=KRCPRT
      DO 60 IRPA=1,NRPA
      IF(NRPA.EQ.IRPA) NRSA=NRLA
      CALL READRA(NWRK,INDCPA,201,IRPA,INDRPA,201)
      CALL ZERO(INDCPZ,1,201,1)
      NCSA=KRCPRT
      NEL=NCSA*KRCPRT
      DO 50 JCPA=1,NCPA
      IF (INDCPA(JCPA+1).EQ.0) GO TO 50
      IF(JCPA.NE.NCPA) GO TO 30
      NCSA=NCLA
      NEL=NCSA*KRCPRT
   30 CALL READRA(NWRK,SA,NEL,JCPA,INDCPA,201)
      DO 40 J=1,NCSA
      NC=(JCPA-1)*KRCPRT+J
      DO 40 I=1,NRSA
      SZ(I,J)=SA(I,J)*D(NC)
   40 CONTINUE
      CALL WRITRA(NWRK,SZ,NEL,JCPA,INDCPZ,201)
   50 CONTINUE
C
      CALL WRITRA(NWRK,INDCPZ,201,IRPA,INDRPZ,201)
   60 CONTINUE
C
      CALL WRITRA(NWRK,INDRPZ,201,1,MHZ(9),2)
      CALL WRITRA(NWRK,MHZ,10,MZ,INDWRK,100)
C
C            RESTORE MASTER INDEX ON NWRK
C            ----------------------------
      CALL STINDX (NWRK,INDWRK,100)
C
      RETURN
C
  999 CALL ZZBOMB(6HZMULDD,NERROR)
      END
```

```
C
C
C
C
C      ******************************************
C      *                                        *
C      *                ZTERP                   *
C      *                                        *
C      ******************************************
C
C
      SUBROUTINE ZTERP (MA,MAI,MB,MBI)
C
      COMMON   /LZWRKF/  NWRK,INDWRK(100),MWRKSP(10)
      COMMON   /LZ1/  MHA(10),INDRPA(201),INDCPA(201),SA(30,30)
      COMMON   /LZ2/  MHAI(10),INDRPAI(201),INDCPAI(201),SAI(30,30)
      COMMON   /LZ3/  MHB(10),INDRPB(201),INDCPB(201),SB(30,30)
      COMMON   /LZ4/  MHBI(10),INDRPBI(201),INDCPBI(201),SBI(30,30)
      COMMON   /LZ5/  MHV(10),INDCPSV(201),INDCPMV(201),MV(2,30) SV(840)
C
      DIMENSION     SBOLD(30), IPART1(2)
C
      DATA     KRCPRT      /30/
C
C       LINEAR INTERPOLATION SUBROUTINE IN PARTITION-LOGIC.
C
C        DEVELOPED BY TG SHANAHAN, JULY 1982.
C
C          SUBROUTINE ARGUMENTS
C          --------------------
C        MA - INPUT MATRIX (A) CONTAINING THE ORIGINAL X-COORDINATES
C             FOR THE ROWS OF (B).  SIZE (1,NTP).
C       MAI - INPUT MATRIX (AI) CONTAINING THE X-COORDINATES FOR THE
C             INTERPOLATE VALUES.  SIZE (1,NT).
C        MB - INPUT MATRIX (B) CONTAINING THE Y-COORDINATES TO BE
C             TO BE INTERPOLATED.  SIZE (NF,NTP).
C       MBI - OUTPUT MATRIX (BI) CONTAINING THE INTERPOLATED
C             Y-COORDINATES.  EACH COLUMN OF (MB) COORESPONDS TO THE
C             COLUMN ELEMENT OF (AI).  SIZE (NF,NT).
C
C          COMMON EXPLANATIONS
C          -------------------
C       /LZWRKF/ - SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE ZWRKFL.
C                  DATA COMES IN AND GOES OUT.
C       /LZ1/, /LZ2/, /LZ3/, LZ4/, AND /LZ5/ ARE ALL WORK SPACES.
C                  NO DATA COMES IN OR GOES OUT OF THESE COMMON BLOCKS.
C
C          NERROR EXPLANATIONS
C          -------------------
C       1 = MATRIX (A) IS NOT A ROW VECTOR.
C       2 = MATRIX (AI) IS NOT A ROW VECTOR.
C       3 = NUMBER OF COLUMNS OF MATRIX (A) IS NOT EQUAL TO THE NUMBER
C           OF COLUMNS IN MATRIX (B).
C       4 = NUMBER OF MATRICES ON WORK FILE (NWRK) EXCEEDS LIMIT.
C       5 = MATRIX (A) IS NOT IN INCREASING ORDER.
C       6 = MATRIX (AI) IS NOT IN INCREASING ORDER.
C       7 = TRYING TO REACH OUT OF RANGE OF MATRIX (B)
C
C       NOTE - ZERO PARTITIONS OF (A) AND (AI) ARE NOT ALLOWED.
C
C     READ IN MATRIX (A) HEADER
C     ------------------------
```

```
      CALL READRA (NWRK,MHA,10,MA,INDWRK,100)
      NRA=MHA(1)
      NCA=MHA(2)
      NRPA=MHA(3)
      NCPA=MHA(4)
      NRLA=MHA(5)
      NCLA=MHA(6)
                                                            NERROR=1
      IF (NRA .NE. 1)                                      GO TO 999
C
C    READ IN MATRIX (AI) HEADER
C    --------------------------
      CALL READRA (NWRK,MHAI,10,MAI,INDWRK,100)
      NRAI=MHAI(1)
      NCAI=MHAI(2)
      NRPAI=MHAI(3)
      NCPAI=MHAI(4)
      NRLAI=MHAI(5)
      NCLAI=MHAI(6)
                                                            NERROR=2
      IF (NRAI .NE. 1)                                     GO TO 999
C
C    READ IN MATRIX (B) HEADER
C    --------------------------
      CALL READRA (NWRK,MHB,10,MB,INDWRK,100)
      NRB=MHB(1)
      NCB=MHB(2)
      NRPB=MHB(3)
      NCPB=MHB(4)
      NRLB=MHB(5)
      NCLB=MHB(6)
                                                            NERROR=3
      IF (NCA .NE. NCB)                                    GO TO 999
C
C    LOCATE MATRIX (BI) AND FORM HEADER
C    ----------------------------------
      DO 10 MBI=1,99
      IF (INDWRK(MBI+1) .EQ. 0)       GO TO 20
   10 CONTINUE
                                                            NERROR=4
                                                           GO TO 999
C
   20 MHBI(1)=NRB
      MHBI(2)=NCAI
      MHBI(3)=NRPB
      MHBI(4)=NCPAI
      MHBI(5)=NRLB
      MHBI(6)=NCLAI
C
      DO 25 I=7,10
      MHBI(I)=0
   25 CONTINUE
C
C    READ IN INDICES OF ROW PARTITIONS
C    ---------------------------------
      CALL READRA (NWRK,INDRPA,201,1,MHA(9),2)
      CALL READRA (NWRK,INDRPAI,201,1,MHAI(9),2)
      CALL READRA (NWRK,INDRPB,201,1,MHB(9),2)
C
      CALL ZERO (INDRPBI,1,201,1)
      CALL ZERO (INDCPSV,1,201,1)
      CALL ZERO (INDCPMV,1,201,1)
```

```
C
C   FORM VECTORS (MV) AND (SV) WHICH TELL WHICH POINTS THE INTERPOLATED
C   POINTS LIE BETWEEN AND THEIR RELATIVE WEIGHTING
C   ----------------------------------------------------------------
      JA=1
      JCPA=1
      IRPA=1
      NCSA=KRCPRT
      IF (NCPA .EQ. 1) NCSA=NCLA
      NELPA=KRCPRT*NCSA
C
      CALL READRA (NWRK,INDCPA,201,1,INDRPA,201)
      CALL READRA (NWRK,INDCPAI,201,1,INDRPAI,201)
      CALL READRA (NWRK,SA,NELPA,JCPA,INDCPA,201)
C
                                                                NERROR=5
      DO 30 I=2,NCSA
      IF (SA(1,I) .LE. SA(1,I-1))                              GO TO 999
   30 CONTINUE
C
      NCSAI=KRCPRT
      NELPAI=KRCPRT*NCSAI
C
      DO 200 JCPAI=1,NCPAI
C
      IF (JCPAI .NE. NCPAI)  GO TO 40
      NCSAI=NCLAI
      NELPAI=NCSAI*KRCPRT
C
   40 CALL READRA (NWRK,SAI,NELPAI,JCPAI,INDCPAI,201)
C
                                                                NERROR=6
      DO 45 I=2,NCSAI
      IF (SAI(1,I) .LE  SAI(1,I-1))                            GO TO 999
   45 CONTINUE
C
      DO 100 JAI=1,NCSAI
C
   50 IF (SA(1,JA) .GT. SAI(1,JAI))      GO TO 70
C
      JA=JA+1
      IF (JA .LE. NCSA)   GO TO 50
C
      IF (JCPA .GE. NCPA)   GO TO 60
      JA=1
      JCPA=JCPA+1
      SAOLD=SA(1,NCSA)
C
      IF (JCPA .NE. NCPA)         GO TO 55
      NCSA=NCLA
      NELPA=NCSA*KRCPRT
   55 CALL READRA (NWRK,SA,NELPA,JCPA,INDCPA,201)
C
                                                                NERROR=5
      DO 52 I=2,NCSA
      IF (SA(1,I) .LE. SA(1,I-1))                              GO TO 999
   52 CONTINUE
      IF (SA(1,1) .LE. SAOLD)                                  GO TO 999
C
      GO TO 50
C
   60 MV(1,JAI)=NCPA
```

```
      MV(2,JAI)=NCLA
      JA=NCLA
      IF (NCLA .EQ. 1)   GO TO 80
      GO TO 90
C
   70 MV(1,JAI)=JCPA
      MV(2,JAI)=JA
      IF (JA .NE. 1)   GO TO 90
C
      IF (JCPA .EQ. 1)   GO TO 85
C
   80 SV(JAI)=(SAI(1,JAI)-SAOLD)/(SA(1,JA)-SAOLD)
      GO TO 100
C
   85 JA=2
      MV(2,JAI)=2
C
   90 SV(JAI)=(SAI(1,JAI)-SA(1,JA-1))/(SA(1,JA)-SA(1,JA-1))
C
  100 CONTINUE
C
      CALL WRITRA (NWRK,MV,2*NCSAI,JCPAI,INDCPMV,201)
      CALL WRITRA (NWRK,SV,NCSAI,JCPAI,INDCPSV,201)
C
  200 CONTINUE
C
C
C    INTERPOLATE THE DATA
C    --------------------
      CALL READRA (NWRK,IPART1,1,1,INDCPMV,201)
      IF (IPART1(2) .EQ. 1)        IPART1(1)=IPART1(1)-1
C
      NRSB=KRCPRT
C
      DO 400 IRPB=1,NRPB
C
      IF (IRPB .EQ. NRPB)        NRSB=NRLB
C
      CALL READRA (NWRK,INDCPB,201,IRPB,INDRPB,201)
C
      JCPB=IPART1(1)
      NCSB=KRCPRT
      IF (NCPB .EQ. IPART1(1))   NCSB=NCLB
      NELPB=NCSB*KRCPRT
C
      IF (INDCPB(JCPB+1) .NE. 0)    GO TO 205
      CALL ZERO (SB,NRSB,NCSB,KRCPRT)
      GO TO 207
C
  205 CALL READRA (NWRK,SB,NELPB,JCPB,INDCPB,201)
C
  207 NCSV=KRCPRT
      NELPBI=NCSV*KRCPRT
      CALL ZERO (INDCPBI,1,201,1)
C
      DO 350 JCPV=1,NCPAI
C
      IF (JCPV .NE. NCPAI)    GO TO 210
C
      NCSV=NCLAI
      NELPBI=NCSV*KRCPRT
C
```

```
  210 CALL READRA (NWRK,SV,NCSV,JCPV,INDCPSV,201)
      CALL READRA (NWRK,MV,2*NCSV,JCPV,INDCPMV,201)
C
      DO 300 J=1,NCSV
C
  215 IF (MV(1,J) .EQ. JCPB)   GO TO 260
C
      DO 220 I=1,NRSB
      SBOLD(I)=SB(I,NCSB)
  220 CONTINUE
C
      JCPB=JCPB+1
                                                          NERROR=7
      IF(JCPB .GT. NCPB)                                 GO TO 999
C
      IF (JCPB .NE. NCPB)  GO TO 230
      NCSB=NCLB
      NELPB=NCSB*KRCPRT
C
  230 IF (INDCPB(JCPB+1) .NE. 0)  GO TO 240
      CALL ZERO (SB,NRSB,NCSB,KRCPRT)
      GO TO 215
C
  240 CALL READRA (NWRK,SB,NELPB,JCPB,INDCPB,201)
      GO TO 215
C .
  260 LV=MV(2,J)
C
      IF (LV .EQ. 1)     GO TO 285
C
      LV1=LV-1
C
      DO 280 I=1,NRSB
      SBI(I,J)=SB(I,LV1)+(SB(I,LV)-SB(I,LV1))*SV(J)
  280 CONTINUE
C
      GO TO 300
C
  285 DO 290 I=1,NRSB
      SBI(I,J)=SBOLD(I)+(SB(I,LV)-SBOLD(I))*     )
  290 CONTINUE
C
  300 CONTINUE
C
      CALL WRITRA (NWRK,SBI,NELPBI,JCPV,INDCPBI,201)
C
  350 CONTINUE
C
      CALL WRITRA (NWRK,INDCPBI,201,IRPB,INDRPBI,201)
C
  400 CONTINUE
C
      CALL WRITRA (NWRK,INDRPBI,201,1,MHBI(9),2)
      CALL WRITRA (NWRK,MHBI,10,MBI,INDWRK,100)
C
C   RESTORE MASTER INDEX ON NWRK
C   -------------------------
      CALL STINDX (NWRK,INDWRK,100)
      RETURN
  999 CALL ZZBOMB (6HZTERP ,NERROR)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C
C            *************************************
C            *                                   *
C            *              ZTERP1               *
C            *                                   *
C            *************************************
C
      SUBROUTINE ZTERP1 (MA,MB,STARTT,ENDT,DELTAT,NTAPE)
C
      COMMON     /LSTART/   IRUNNO,IDATE,NPAGE,UNAME(3),T1(12),T2(12)
      COMMON     /LZWRKF/  NWRK,INDWRK(100),MWRKSP(10)
      COMMON     /LZ1/     MHA(10),INDRPA(201),INDCPA(201),SA(900)
      COMMON     /LZ2/     MHB(10),INDRPB(201),INDCPB(201),SB(30,30)
      COMMON     /LWORK1/  Z(600),DB(31,600)
C
      DATA           KRCPRT,KCDB    /30,600/
      DATA           BUF            /-1.E50/
C
C       LINEAR INTERPOLATION SUBROUTINE IN PARTITION LOGIC.
C       INTERPOLATES MATRIX (MB) TO EVENLY SPACED POINTS.
C       THE INTERPOLATED ANSWERS FOR EACH NEW X-COORDINATE
C       ARE WRITTEN ON NTAPE SEQUENTIALLY ALONG WITH THE NEW
C       X-COORDINATE.   INTERPOLATION SCHEME INVOLVES DISASSEMBLING
C       A ROW PARTITION OF MATRIX (MB) AND CONVERTING THIS TO DENSE-
C       LOGIC.  IT IS THEREFORE SIZE LIMITED TO THE COLUMN DIMENSION
C       SIZE OF (DB) IN COMMON BLOCK /LWORK1/, WHICH IS KCDB=600.
C       NOTE THAT THIS IS NOT A PURE PARTITION-LOGIC SUBROUTINE
C       AND ITS SPACE REQUIREMENTS ARE LARGER THAN A PARTITION-LOGIC
C       SUBROUTINE.
C
C         SUBROUTINE ARGUMENTS
C         --------------------
C
C           MA - INPUT MATRIX (A) OF X-COORDINATES FOR THE ROWS OF
C                MATRIX (B).  SIZE (NRA).
C           MB - INPUT MATRIX (B) OF Y-COORDINATES TO BE INTERPOLATED.
C                SIZE (NRA,NCA).  NCA IIS THE NUMBER OF DIFFERENT Y
C                VECTORS TO BE INTERPOLATED.
C       STARTT - STARTING POINT OF X-COORDINATES THAT MATRIX (B) IS
C                ITERPOLATED TO.
C       DELTAT - INCREMENT OF X-COORDINATE.
C         ENDT - LAST X-COORDINATE FOR INTERPOLATION OF MATRIX (B).
C        NTAPE - NUMBER OF SEQUENTIAL TAPE ON WHICH INTERPOLATED ANSWERS
C                ARE WRITTEN.  NTAPE ALSO INCLUDES A HEADER.
C
C         COMMON EXPLANATIONS
C         -------------------
C        /LZSTART/ - SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE START.
C                    DATA COMES IN ONLY.
C         /LZWRKF/ - SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE ZWRKFL.
C                    DATA COMES IN AND GOES OUT.
C            /LZ1/ - WORK SPACE.  NO DATA COMES IN OR GOES OUT.
C            /LZ2/ - WORK SPACE.  NO DATA COMES IN OR GOES OUT.
C         /LWORK1/ - WORK SPACE.  NO DATA COMES IN OR GOES OUT.
C
C         NERROR EXPLANATIONS
C         -------------------
C        1 = MATRIX (A) IS NOT A COLUMN VECTOR
C        2 = MATRICES (A) AND (B) DO NOT HAVE THE SAME ROW SIZE.
C        3 = COLUMN SIZE OF (B) - THE NUMBER OF VECTORS TO BE
C            INTERPOLATED - IS GREATER THAN KCNB.  ORIGINAL    KCNB=600.
C            MORE VECTORS CAN BE USED BY REDIMENSIONING THE COLUMN SIZE
```

```
C            OF MATRIX (DB) AND CHANGING THE VALUE OF KCNB TO THIS NEW
C            SIZE.
C
C       FORM MATRIX (A) HEADER
C       ----------------------
      CALL READRA (NWRK,MHA,10,MA,INDWRK,201)
      NRA=MHA(1)
      NCA=MHA(2)
      NRPA=MHA(3)
      NCPA=MHA(4)
      NRLA=MHA(5)
      NCLA=MHA(6)
C
                                                                NERROR=1
      IF (NCA .NE. 1)                                          GO TO 999
C
C       FORM MATRIX (B) HEADER
C       ----------------------
      CALL READRA (NWRK,MHB,10,MB,INDWRK,201)
      NRB=MHB(1)
      NCB=MHB(2)
      NRPB=MHB(3)
      NCPB=MHB(4)
      NRLB=MHB(5)
      NCLB=MHB(6)
C
                                                                NERROR=2
      IF(NRA .NE. NRB)                                         GO TO 999
                                                                NERROR=3
      IF (NCB .GT. KCDB)                                       GO TO 999
C
      T=STARTT
      STOPT=ENDT+DELTAT
      NTP=(STOPT-STARTT)/DELTAT
C
      REWIND NTAPE
      WRITE (NTAPE) IRUNNO,IDATE,STARTT,ENDT,DELTAT,NCB,NTP,(BUF,I=1,10)
C
      CALL READRA (NWRK,INDRPA,201,1,MHA(9),2)
      CALL READRA (NWRK,INDRPB,201,1,MHB(9),2)
C
      IRPA=1
      NRSA=KRCPRT
C
C       READ IN A ROW PARTITION OF MATRICES (A) AND (B)
C       -----------------------------------------------
   20 IF(IRPA .EQ. NRPA) NRSA=NRLA
      CALL READRA (NWRK,INDCPA,201,IRPA,INDRPA,201)
      CALL READRA (NWRK,SA(2),NRSA,1,INDCPA,201)
      CALL READRA (NWRK,INDCPB,201,IRPA,INDRPB,201)
      NCSB=KRCPRT
      NEL=NCSB*KRCPRT
      DO 60 JCPB=1,NCPB
      IF (JCPB.NE.NCPB) GO TO30
      NCSB=NCLB
      NEL=NCSB*KRCPRT
   30 IF (INDCPB(JCPB+1) .NE. 0) GO TO 40
      CALL ZERO (SB,NRSA,NCSB,KRCPRT,KRCPRT)
      GO TO 50
   40 CALL READRA (NWRK,SB,NEL,JCPB,INDCPB,201)
C
C       ASSEMBLE THIS PARTITION OF (B) INTO (DB)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
C         -------------------------------------
   50 DO 60 J=1,NCSB
      NC=(JCPB-1)*KRCPRT+J
      DO 60 I=1,NRSA
      DB(I+1,NC)=SB(I,J)
   60 CONTINUE
C
C        FIND TIME POINTS TO INTERPOLATE BETWEEN
C        -------------------------------------
      I=1
   80 IF (T .LE. SA(I+1)) GO TO 100
      I=I+1
      IF (I .LE. NRSA) GO TO 80
      IF (IRPA .EQ. NRPA) GO TO 200
C
C        WRITE THE LAST ROW (DB) INTO THE FIRST ROW
C        FOR INTERPOLATION BETWEEN ROW PARTITIONS
C        ----------------------------------------
      SA(1)=SA(NRSA+1)
      DO 90 J=1,NCB
      DB(1,J)=DB(NRSA+1,J)
   90 CONTINUE
C
      IRPA=IRPA+1
      GO TO 20
C
  100 IF (I .EQ. 1 .AND. IRPA .EQ. 1) GO TO 140
C
C        NORMAL INTERPOLATION
C        --------------------
      DELT=(T-SA(I))/(SA(I+1)-SA(I))
      DO 120 J=1,NCB
      Z(J)=DB(I,J)+DELT*(DB(I+1,J)-DB(I,J))
  120 CONTINUE
      GO TO 180
C
C        EXTRAPOLATION PRCEEDING DATA POINTS
C        ---------------------------------
  140 DELT=(T-SA(2))/(SA(3)-SA(2))
      DO 160 J=1,NCB
      Z(J)=DB(2,J)+DELT*(DB(3,J)-DB(2,J))
  160 CONTINUE
C
C        WRITE TIME AND INTERPOLATED VALUES ONTO NTAPE SEQUENTIALLY
C        ----------------------------------------------------------
  180 WRITE (NTAPE) T,(Z(J),J=1,NCB),BUF
C
C        CHECK TO SEE IF NEXT TIME POINT IS REQUIRED
C        -----------------------------------------
      T=T+DELTAT
      IF (T .LE.STOPT) GO TO 80
      ENDFILE NTAPE
      RETURN
C
C        EXTRAPOLATION AFTER DATA POINTS
C        -----------------------------
  200 DELT=(T-SA(NRSA))/(SA(NRSA+1)-SA(NRSA))
      DO 220 J=1,NCB
      Z(J)=DB(NRSA,J)+DELT*(DB(NRSA+1,J)-DB(NRSA,J))
  220 CONTINUE
C
C        WRITE EXTRAPOLATED VALUES ONTO NTAPE
```

```
C         ------------------------------------
      WRITE (NTAPE) T,(Z(J),J=1,NCB),BUF
C
C       CHECK TO SEE IF THE NEXT TIME POINT IS REQUIRED
C       ---------------------------------------------------
      T=T+DELTAT
      IF (T.LE.STOPT) GO TO 200
      ENDFILE NTAPE
      RETURN
C
  999 CALL ZZBOMB (6HZTERP1,NERROR)
      END
```

```
C
C
C
C      ****************************************
C      *                                      *
C      *               ZTOSEQ2                *
C      *                                      *
C      ****************************************
C
C
      SUBROUTINE ZTOSEQ2 (MA,MB,NTAPE)
C
      COMMON /LSTART/ IRUNNO,IDATE,NPAGE,UNAME(3),T1(12),T2(12)
      COMMON /LZWRKF/ NWRK,INDWRK(100),MWRKSP(10)
      COMMON /LZ1/    MHA(10),INDRPA(201),INDCPA(201),SA(900)
      COMMON /LZ2/    MHB(10),INDRPB(201),INDCPB(201),SB(30,30)
      COMMON /LWORK1/ DA(600),DB(600,30),FILLER(600)
C
      DATA    KC         /600/
      DATA    KRCPRT     /30/
      DATA    BUF        /-1.E50/
C
C     THIS SUBROUTINE TAKES TO MATRICES, MATCHES AN ELEMENT FROM
C     COLUMN VECTOR (A) WITH A ROW OF MATRIX (B) AND WRITES THESE
C     ELEMENTS OUT SEQUENTIALLY.  THIS SUBROUTINE OPERATES BY
C     DISASSEMBLING OUT A ROW PARTITION OF MATRIX (B) AND CONVERTING
C     IT TO A DENSE-LOGIC MATRIX.  IT IS THEREFORE LIMITED TO HANDLING
C     MATRICES THAT HAVE A COLUMN DIMENSION SIZE SMALLER THAN THE
C     DIMENSION OF (DB) WHICH WAS ORIGINALLY 600.
C
C      DEVELOPED BY TG SHANAHAN,  JULY 1982.
C      -------------------------------------
C
C
C
C       SUBROUTINE ARGUMENTS
C       --------------------
C          MA = INPUT COLUMN MATRIX (A) GIVING THE X-COORDINATES THAT
C               CORRESPOND TO THE ROWS OF MATRIX (B).  SIZE (NTP,1).
C          MB = INPUT MATRIX (B) CONTAINING THE Y-COORDINATES FOR
C               EACH X-COORDINATE IN MATRIX (A).  SIZE (NTP,NF).
C       NTAPE = INPUT LOGICAL UNIT NUMBER OF THE SEQUENTIAL TAPE THAT
C               THE DATA CONTAINED IN MATRICES (A) AND (B) IS TO BE
C               WRITTEN.
C
C       COMMON EXPLANATIONS
C       -------------------
C       /LSTART/ - SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE START.
C                  DATA COMES IN.
C       /LZWRKF/ - SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE ZWRKFL.
C                  DATA COMES IN.
C          /LZ1/ - WORK SPACE. NO DATA COMES IN OR GOES OUT.
C          /LZ2/ - WORK SPACE. NO DATA COMES IN OR GOES OUT.
C       /LWORK1/ - WORK SPACE. NO DATA COMES IN OR GOES OUT.
C
C       NERROR EXPLANATIONS
C       -------------------
C       1 = MATRIX (A) IS NOT A COLUMN VECTOR.
C       2 = NUMBER OF ROWS IN MATRIX (A) IS NOT EQUAL TO THE NUMBER OF
C           ROWS IN MATRIX (B).
C       3 = COLUMN SIZE OF MATRIX (B) IS TOO LARGE FOR THIS PROGRAM.
C
C       NOTE - ZERO PARTITIONS ARE NOT ALLOWED IN MATRIX (A).
```

```
C
C   READ IN MATRIX (A) HEADER
C   ------------------------
      CALL READRA (NWRK,MHA,10,MA,INDWRK,100)
      NRA=MHA(1)
      NCA=MHA(2)
      NRPA=MHA(3)
      NCPA=MHA(4)
      NRLA=MHA(5)
      NCPA=MHA(6)
                                                         NERROR=1
      IF (NCA .NE. 1)                                   GO TO 999
C
C   READ IN MATRIX (B) HEADER
C   ------------------------
      CALL READRA (NWRK,MHB,10,MB,INDWRK,100)
      NRB=MHB(1)
      NCB=MHB(2)
      NRPB=MHB(3)
      NCPB=MHB(4)
      NRLB=MHB(5)
      NCLB=MHB(6)
                                                         NERROR=2
      IF (NRB .NE. NRA)                                 GO TO 999
                                                         NERROR=3
      IF (NCB .GT. KC)                              .   GO TO 999
C
C  READ IN THE INDICES OF THE ROW PARTITIONS OF THE TWO MATRICES
C   ------------------------------------------------------------
      CALL READRA (NWRK,INDRPA,201,1,MHA(9),2)
      CALL READRA (NWRK,INDRPB,201,1,MHB(9),2)
C
C   FORM THE HEADER FOR NTAPE
C   ------------------------
      CALL READRA (NWRK,INDCPA,201,1,INDRPA,201)
      CALL READRA (NWRK,AFIRST,1,1,INDCPA,201)
      CALL READRA (NWRK,INDCPA,201,NRPA,INDRPA,201)
      CALL READRA (NWRK,SA,NRLA,1,INDCPA,201)
      ALAST=SA(NRLA)
      ADELTA=(ALAST-AFIRST)/(NRA-1)
C
      REWIND NTAPE
      WRITE (NTAPE) IRUNNO,IDATE,AFIRST,ALAST,ADELTA,NCB,NRA,
     +                    (BUF,I=1,10)
C
C   WRITE DATA OUT ONTO NTAPE
C   ------------------------
      NRSA=KRCPRT
C
      DO 100 IRPA=1,NRPA
C
      IF (IRPA .EQ. NRPA)      NRSA=NRLA
C
      CALL READRA (NWRK,INDCPA,201,IRPA,INDRPA,201)
      CALL READRA (NWRK,SA,NRSA,1,INDCPA,201)
C
      CALL READRA (NWRK,INDCPB,201,IRPA,INDRPB,201)
C
      NCSB=KRCPRT
      NELPB=NCSB*KRCPRT
C
      DO 50 JCPB=1,NCPB
```

```
C
      IF (JCPB .NE. NCPB)    GO TO 10
      NCSB=NCLB
      NELPB=NCSB*KRCPRT
C
   10 IF (INDCPB(JCPB+1) .NE. 0)    GO TO 20
      CALL ZERO (SB,KRCPRT,NCSB,KRCPRT)
      GO TO 25
C
   20 CALL READRA (NWRK,SB,NELPB,JCPB,INDCPB,201)
C
   25 JPOS=(JCPB-1)*KRCPRT
      DO 45 I=1,NRSA
      DO 45 J=1,NCSB
      DB(JPOS+J,I)=SB(I,J)
   45 CONTINUE
C
   50 CONTINUE
C
      DO 80 I=1,NRSA
      WRITE (NTAPE)  SA(I),(DB(J,I),J=1,NCB),BUF
   80 CONTINUE
C
  100 CONTINUE
C
      RETURN
C
  999 CALL ZZBOMB (6HZTOSE2,NERROR)
C
      END
```

```
C
C
C     *************************************
C     *                                   *
C     *              ZTOSEQ3              *
C     *                                   *
C     *************************************
C
C
      SUBROUTINE ZTOSEQ3 (MA,MB,NTAPE)
C
      COMMON  /LSTART/  IRUNNO,IDATE,NPAGE,UNAME(3),TITLE1(12),TITLE2(12)
      COMMON  /LZWRKF/  NWRK,INDWRK(100),MWRKSP(10)
      COMMON  /LZ1/     MHA(10),INDRPA(201),INDCPA(201),SA(900)
      COMMON  /LZ2/     MHB(10),INDRPB(201),INDCPB(201),SB(30,30)
      COMMON  /LZ3/     MHZ(10),INDCPZ(201),INDRZ(201),Z(900)
      COMMON  /LWORK1/  ZA(6000),FILLER(13200)
C
      DATA    KRCPRT    /30/
      DATA    BUF       /-1.E50/
C
C       THIS SUBROUTINE MATCHES AN ELEMENT OF COLUMN VECTOR (A) AND
C       A ROW OF MATRIX (B) AND WRITES THEM OUT TOGETHER SEQUENTIALLY
C       ON NTAPE.
C
C       THE FORM OF NTAPE IS AS FOLLOWS :
C       HEADER - CONTAINING RUNNO, DATE, ETC.
C       THE DATA -  A(1),  B(1,1),  B(1,2),  B(1,3)  ,....,B(1,NCB),
C                   A(2),  B(2,1),  B(2,2),  B(2,3)  ,....,B(2,NCB),
C                     .  ,    .   ,    .   ,    .    ,.....,  .  ,
C                     .  ,    .   ,    .   ,    .    ,.....,  .  ,
C                   A(NRA),B(NRA,1),B(NRA,2),B(NRA,3),.....,B(NRA,NCB)
C
C     DEVELOPED BY TG SHANAHAN, JULY 1982.
C     ------------------------------------
C
C       SUBROUTINE ARGUMENTS
C       --------------------
C       MA = INPUT MATRIX (A) WHICH CONTAINS THE X-COORDINATES THAT
C            CORRESPOND TO THE ROWS OF Y-COORDINATES IN MATRIX (B).
C            SIZE (NRA,1).
C       MB = INPUT MATRIX (B) WHICH CONTAINS THE Y-COORDINATES.
C            EACH ROW OF MATRIX (B) CORRESPONDS TO AN ELEMENT OF
C            MATRIX (A).  SIZE (NRA,NCB).
C    NTAPE = INPUT LOGICAL UNIT NUMBER TO WHICH THE DATA WILL BE
C            WRITTEN.
C
C        COMMON EXPLANATIONS
C        -------------------
C      /LSTART/ SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE START.
C               DATA COMES IN ONLY.
C      /LZWRKF/ SHOULD HAVE BEEN INITIALIZED IN SUBROUTINE ZWRKFL.
C               DATA COMES IN ONLY.
C         /LZ1/ WORK SPACE.  DATA COMES IN AND GOES OUT.
C         /LZ2/ WORK SPACE.  DATA COMES IN AND GOES OUT.
C         /LZ3/ WORK SPACE.  DATA COMES IN AND GOES OUT.
C
C        NERROR EXPLANATIONS
C        -------------------
C
C      1 = MATRIX (A) IS NOT A COLUMN VECTOR.
C      2 = NUMBER OF ROWS OF MATRIX (A) DOES NOT EQUAL THE NUMBER
C          OF ROWS OF MATRIX (B).
```

```
C
C        NOTE - ZERO PARTITIONS OF MATRRIX (A) ARE NOT ALLOWED
C
C
C     READ IN MATRIX (A) HEADER
C     -------------------------
        CALL READRA (NWRK,MHA,10,MA,INDWRK,100)
        NRA=MHA(1)
        NCA=MHA(2)
        NRPA=MHA(3)
        NCPA=MHA(4)
        NRLA=MHA(5)
        NCLA=MHA(6)
C
        IF (NCA .NE. 1)                                          NERROR=1
                                                                 GO TO 999
C
C     READ IN MATRIX (B) HEADER
C     -------------------------
        CALL READRA (NWRK,MHB,10,MB,INDWRK,100)
        NRB=MHB(1)
        NCB=MHB(2)
        NRPB=MHB(3)
        NCPB=MHB(4)
        NRLB=MHB(5)
        NCLB=MHB(6)
        IF (NRB .NE. NRA)                                        NERROR=2
                                                                 GO TO 999
C
        CALL READRA (NWRK,INDRPA,201,1,MHA(9),2)
        CALL READRA (NWRK,INDRPB,201,1,MHB(9),2)
C
C     DETERMINE THE FIRST AND LAST VALUE OF (A)
C     -----------------------------------------
        CALL READRA (NWRK,INDCPA,201,1,INDRPA,201)
        CALL READRA (NWRK,AFIRST,1,1,INDCPA,201)
        CALL READRA (NWRK,INDCPA,201,NRPA,INDRPA,201)
        CALL READRA (NWRK,SA,NRLA,1,INDCPA,201)
        ALAST=SA(NRLA)
C
        ADELTA=(ALAST-AFIRST)/(NRB-1)
C
C     WRITE HEADER ON NTAPE
C     ---------------------
        REWIND NTAPE
        WRITE (NTAPE)  IRUNNO,IDATE,AFIRST,ALAST,ADELTA,NCB,NRB,
       +               (BUF,I=1,10)
C
C     WRITE DATA ON NTAPE
C     -------------------
        NRSB=KRCPRT
C
        DO 500 IRPB=1,NRPB
C
        IF (IRPB .EQ. NRPB)   NRSB=NRLB
        CALL READRA (NWRK,INDCPA,201,IRPB,INDRPA,201)
        CALL READRA (NWRK,SA,NRSB,1,INDCPA,201)
        CALL READRA (NWRK,INDCPB,201,IRPB,INDRPB,201)
C
        CALL ZERO (INDCPZ,1,201,1)
C
        NCSB=KRCPRT
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      NELPB=NCSB*KRCPRT
C
      DO 300 JCPB=1,NCPB
C
      IF (JCPB .NE. NCPB)   GO TO 40
      NCSB=NCLB
      NELPB=NCSB*KRCPRT
C
   40 IF (INDCPB(JCPB+1) .NE. 0)   GO TO 50
      CALL ZERO (SB,NRSB,NCSB,KRCPRT)
      GO TO 60
C
   50 CALL READRA (NWRK,SB,NELPB,JCPB,INDCPB,201)
   60 CALL ZERO (INDRZ,1,31,1)
C
      DO 250 I=1,NRSB
      DO 240 J=1,NCSB
      ZA(J)=SB(I,J)
  240 CONTINUE
      CALL WRITRA (NWRK,ZA,NCSB,I,INDRZ,31)
  250 CONTINUE
C
      CALL WRITRA (NWRK,INDRZ,31,JCPB,INDCPZ,201)
C
  300 CONTINUE
C
      DO 400 I=1,NRSB
      NCSB=KRCPRT
C
      DO 350 JCPB=1,NCPB
      IF (JCPB .EQ. NCPB)   NCSB=NCLB
      JPOS=(JCPB-1)*KRCPRT+1
      CALL READRA (NWRK,INDRZ,31,JCPB,INDCPZ,201)
      CALL READRA (NWRK,ZA(JPOS),NCSB,I,INDRZ,31)
  350 CONTINUE
C
      WRITE (NTAPE)  SA(I),(ZA(J),J=1,NCB),BUF
C
  400 CONTINUE
C
  500 CONTINUE
C
C     RESTORE MASTER INDEX ON NWRK
C     ----------------------------
      CALL STINDX (NWRK,INDWRK,100)
      RETURN
C
  999 CALL ZZBOMB (6HZTOSEQ,NERROR)
C
```