NASA
Technical
Memorandum

NASA TM - 103505

A TRANSIENT RESPONSE ANALYSIS OF THE SPACE SHUTTLE VEHICLE DURING LIFTOFF

By J.A. Brunty

Structures and Dynamics Laboratory
Science and Engineering Directorate

August 1990

National Aeronautics and
Space Administration

George C. Marshall Space Flight Center

National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM - 103505	2. Government Accession No.	3. Recipient's Catalog No.
4. Title and Subtitle A Transient Response Analysis of the Space Shuttle Vehicle During Liftoff		5. Report Date August 1990
		6. Performing Organization Code
7. Author(s) J.A. Brunty		8. Performing Organization Report No.
		10. Work Unit No.
9. Performing Organization Name and Address George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812		11. Contract or Grant No.
		13. Type of Report and Period Covered Technical Memorandum
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546		14. Sponsoring Agency Code

15. Supplementary Notes

Prepared by Structures and Dynamics Laboratory, Science and Engineering Directorate

16. Abstract

A proposed transient response method is formulated for the liftoff analysis of the space shuttle vehicle. The proposed method uses a power series approximation with unknown coefficients for the interface forces between the space shuttle and mobile launch platform. This allows the equations of motion of the two structures to be solved separately with unknown coefficients at the end of each time step. The unknown coefficients are obtained by enforcing the interface compatibility conditions between the two structures. Once the unknown coefficients are determined, the total response is computed for that time step. The method is validated by a numerical example of a cantilevered beam and by the liftoff analysis of the space shuttle vehicle. The proposed method is compared to an iterative transient response analysis method used by Martin Marietta for their space shuttle liftoff analysis. It is shown that the proposed method uses less computer time than the iterative method and does not require as small a time step for integration. The space shuttle vehicle model is reduced using two different types of component mode synthesis (CMS) methods, the Lanczos CMS method and the Craig and Bampton CMS method. By varying the cutoff frequency in the Craig and Bampton method it was shown that the space shuttle interface loads can be computed with reasonable accuracy. Both the Lanczos CMS method and Craig and Bampton CMS method give similar results. A substantial amount of computer time is saved using the Lanczos CMS method over that of the Craig and Bampton method. However, when trying to compute a large number of Lanczos vectors, input/output computer time increased and increased the overall computer time. The application of several liftoff release mechanisms that can be adapted to the proposed method are discussed.

17. Key Words (Suggested by Author(s)) Transient Response, Liftoff, Space Shuttle, Structural Dynamics	18. Distribution Statement Unclassified – Unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	107	NTIS

NASA FORM 1626 OCT 86

For sale by the National Technical Information Service, Springfield, VA 22161-2171

TABLE OF CONTENTS

		Page
I.	INTRODUCTION	1
	A. General Background	1
	B. Description of Space Shuttle Liftoff Release Mechanism	3
	C. Proposed Method	3
	D. Objectives	3
II.	GOVERNING EQUATIONS	7
	A. CMS Method	7
	B. Lanczos Vectors	10
	C. Martin Marietta's Transient Response Method With Changing Boundary Conditions	12
	D. Proposed Transient Response Method With Changing Boundary Conditions	18
III.	COMPUTATIONAL PROCEDURE	27
	A. Computational Procedure for the CMS Methods	27
	B. Computational Procedure for the Proposed Transient Response Method	28
IV.	NUMERICAL EXAMPLES	29
	A. Simple Beam	29
	B. Transient Response of Shuttle Liftoff	32
V.	CONCLUSIONS	56
REFERENCES		58
APPENDIX		61

LIST OF ILLUSTRATIONS

Figure	Title	Page
1.	Space shuttle and launch pad structures	4
2.	SRB holddown bolt and foot pad	5
3.	SRB/MLP post interface locations in orbiter coordinates	6
4.	Free-body diagram of two substructures	19
5.	Cantilevered beam with applied load $F(t)$	30
6a.	Two-dimensional finite element model of cantilevered beam	31
6b.	Element mass and stiffness for the cantilevered beam	31
7.	Frequency versus mode number for free-free beam Craig-Bampton CMS model comparisons	33
8.	Frequency versus mode number for cantilevered beam Craig-Bampton CMS model comparisons	34
9.	Frequency versus mode number for free-free beam Lanczos CMS model comparisons	35
10.	Frequency versus mode number of a cantilevered beam Lanczos CMS model comparisons	36
11.	Proposed transient response method tip displacement versus time (Craig and Bampton versus Lanczos)	37
12.	Proposed transient response method interface moment versus time (Craig and Bampton versus Lanczos)	38
13.	Frequency versus mode number of liftoff shuttle model (Lanczos vectors used for reduction)	40
14.	Proposed method versus iterative method post 1 interface forces (lb) versus time (seconds)	42
15.	Proposed method versus iterative method post 4 interface forces (lb) versus time (seconds)	43

LIST OF ILLUSTRATIONS (Continued)

Figure	Title	Page
16.	Proposed transient response method post 1 interface forces (lb) versus time (seconds)	45
17.	Proposed transient response method post 4 interface forces (lb) versus time (seconds)	46
18.	Proposed transient response method post 1 interface forces (lb) versus time (seconds)	47
19.	Proposed transient response method post 4 interface forces (lb) versus time (seconds)	48
20.	Proposed transient response method post 1 interface forces (lb) versus time (seconds)	49
21.	Proposed transient response method post 4 interface forces (lb) versus time (seconds)	50
22.	Proposed transient response method post 1 interface forces (lb) versus time (seconds)	51
23.	Proposed transient response method post 4 interface forces (lb) versus time (seconds)	52
24.	Proposed transient response method post 1 boundary displacement (inches) versus time (seconds)	53
25.	Proposed transient response method post 1 boundary displacement (inches) versus time (seconds)	54
26.	Proposed transient response method post 1 boundary displacement (inches) versus time (seconds)	55

LIST OF TABLES

Table	Title	Page
1.	Sizes of reduced vehicle models	39
2.	Computer time comparisons of the CMS methods and the proposed transient response method	44

NOMENCLATURE

$[C]$	damping coefficient matrix
$[\overline{C}]$	interface compatibility coefficient matrix, equation (51)
$\{F\}$	force vector
$[G]$	Guyan transformation matrix, equation (15)
$\{G_j\}$	term j in expansion of interface force vector, equation (34)
$[I]$	unity matrix
$[K]$	stiffness matrix
$[\hat{K}_c]$	coupling stiffness matrix, equation (16)
$[M]$	mass matrix
$\{q\}$	modal coordinate vector
t	time
t_i	time at ith time increment
$\{x\}$	physical coordinates
Δt	time increment of integration step
$[\Phi]$	mode shape matrix
$[\omega^2]$	diagonal matrix of square of frequency
ζ	damping factor

Subscripts

A, B	substructure A, B, respectively
CB	Craig and Bampton formulation
I	interface
L	Lanczos vector formulation

V, P	substructure vehicle, pad, respectively
i,b	interior, boundary
n	number of modes/vectors retained

Superscripts

T	transpose of matrix
-1	inverse of matrix
$*$	rows of interface degrees-of-freedom

TECHNICAL MEMORANDUM

A TRANSIENT RESPONSE ANALYSIS OF THE SPACE SHUTTLE VEHICLE DURING LIFTOFF

I. INTRODUCTION

The development of analytical tools for the design and analysis of complex structures has been a great achievement for structural engineers over the past 3 decades. With the advent of the computer age, new numerical analysis techniques have evolved utilizing the well-known finite element method. These techniques have developed into good representations for modeling the structural characteristics of complex structures; however, the models of today's complex structures can have several thousand degrees-of-freedom (DOF). These large models, therefore, become impractical to analyze on present-day computers from a monetary and computational sense. The problem concerned in this report is the liftoff dynamic transient response analysis of the space shuttle vehicle. The dynamic transient interaction between the launch pad and the space shuttle vehicle is a very complex phenomenon and requires detailed modeling of its structural components. This leads to models with thousands of DOF that represent the space shuttle vehicle. In order to analyze the liftoff event, the space shuttle models are reduced using component mode synthesis (CMS) methods. It is typical during liftoff that the maximum internal loads occur on the vehicle. These maximum internal loads on the space shuttle are a result of changing boundary conditions over a very short time span. The reduced model of the space shuttle, therefore, has an important effect on the accuracy of the computed internal loads. This report proposes a method which will compute the liftoff transient response of the space shuttle vehicle from its launch pad using a set of reduced models. The method is proposed to reduce the amount of computer cost of each liftoff analysis since there are over 300 individual sets of forcing environments that must be analyzed for each flight. The proposed method will be verified by comparing results with an iterative method used by Martin Marietta. The effects of the reduced CMS models used in the proposed analysis will be studied.

A. General Background

A structure with an infinite number of DOF is approximated by a finite number of DOF by using the finite element method. This approach offers a very good approximation when a reasonable number of DOF are retained in the structure's model. Of course, the more DOF a model has the more time consuming it is to analyze on a computer. The finite element formulation of a structure results in a set of coupled second-order matrix differential equations. The differential equations which represent the equations of motion of the structure can be solved by a number of numerical techniques. One approach is to numerically integrate the equations of motion. This may be impractical for models with several thousand DOF and limited computer resources. Another approach would be to use normal coordinates by solving the eigenvalue problem for the undamped and free motion. A reduction of the model could be performed through truncation of the vibrational modes. The normal coordinates have the advantage of uncoupling the differential equations. This approach loses its effectiveness if the size of the structural model is so large that it becomes impractical from a computational sense. Research in the area of vibration analysis of large order systems [1–3] has overcome some of these difficulties; however, researchers are continually searching for improvements. Another reduction technique referred to as static condensation, or Guyan reduction, is commonly applied to large size models. The method was originally developed for the reduction of the stiffness matrix and was extended to the mass matrix by Guyan [4]. This reduction technique reduces those DOF that are not significant for the dynamic analysis being performed (e.g., massless DOF). Proper selection of the DOF is required for accurate results. A model reduction technique which uses both the static condensation method in combination with an eigenvalue analysis is referred to as CMS method. There are a number of variations associated with CMS [5–12]. Recent research works [13–17] have shown that Lanczos vectors can be used as an efficient tool for CMS. The research has demonstrated that accurate results can be obtained for some small structural models. A reduction of large finite element models (i.e., equations of motion) must be accomplished before a transient response analysis can be performed. Therefore, the method used to reduce the structural models is important in both computational work and accuracy of solutions. One objective of this report will be to determine if the Lanczos CMS method can be used effectively on large complex structural models as compared to the Craig and Bampton CMS method. The space shuttle liftoff vehicle model will be used as an example.

The liftoff phase of an aerospace vehicle is a critical time period, because some of the maximum internal loads occur during this time. Several methods exist which have demonstrated acceptable accuracy and efficiency for the liftoff transient response analysis [18–22]. One method uses a Runge-Kutta numerical integration scheme used on the Titan rocket and reformulated for the space shuttle by Blejwas of Martin Marietta [18]. The boundary stiffness matrix of the vehicle is coupled to a stiffness matrix representing the launch pad. As the vehicle lifts off the launch pad, the interface loads between the vehicle and pad go from compression to tension. When this occurs, the introduced stiffness matrix between the vehicle and pad is reduced out and a new stiffness matrix is instituted. Thus, the vehicle is transformed from being in a fixed-boundary condition state to a free-flight environment. Another method proposed by White and Bodley of Martin Marietta [19] uses Lagrange multipliers in the formulation of equations of motion. These Lagrange multipliers, which represent the boundary forces, couple the vehicle equations of motion to the launch pad equations of motion. The Lagrange multipliers are determined iteratively at each time step of numerical integration. Once the Lagrange multipliers are determined for that time step, the corresponding response at that time step can be computed. During the separation phase of the analysis, the Lagrange multipliers are zeroed out as the vehicle lifts off the launch pad. A method proposed by Olberding [20] uses a coupling stiffness matrix between the vehicle and the pad. The coupling stiffness, when multiplied by the boundary displacements, represents the contact forces. The equations of motion are integrated using a numerical integration algorithm (Runge-Kutta or multistep), and the coupling stiffness matrix is modified as the vehicle lifts off. A different iterative method by Prabhakar [21] of Martin Marietta also uses a coupling stiffness matrix between the vehicle and pad. The coupling stiffness matrix is representative of the actual holddown studs used on the space shuttle. The contact forces are solved iteratively over one time step. They are then used to solve for the total response over that time step. During liftoff the coupling stiffness is modified allowing the vehicle free flight. The iterative method by Prabhakar [21] will be presented in detail later and will be used in comparison studies in this report.

B. Description of Space Shuttle Liftoff Release Mechanism

The transient response analysis of the space shuttle from the mobile launch platform (MLP) is performed after modifying the proposed transient response algorithm to include changes in boundary conditions. Figure 1 shows the space shuttle vehicle mounted to the MLP. The space shuttle liftoff vehicle is composed of two solid rocket boosters (SRB's), an external tank (ET), and the orbiter vehicle. The space shuttle vehicle is fixed to the MLP through the SRB aft skirts at eight points of contact. One of these connections is shown in figure 2. These eight points are shown relative to one another on the MLP in figure 3. Three DOF (X, Y, and Z directions) are retained for each one of these contact points, therefore, a total of 24 DOF are used to connect the liftoff vehicle to the MLP. Some assumptions have been made for the complex release mechanism of the space shuttle from the MLP, such as no frictional loads, lateral force feedback dynamics, bolt hangup mechanisms, or interface moment loads, etc., in the transient response analysis. The release mechanism employed on the space shuttle vehicle and MLP begins with the ignition of the SRB's. A signal is sent to the eight contact detonators after SRB ignition. The detonators then separate the eight flangible nuts (see fig. 2). These nuts are captured in blast containers. Holddown studs then drop into the MLP support posts due to gravity, and the vehicle lifts off from the MLP. This all occurs in about 0.25 s after SRB ignition. To simulate this effect in the transient response analysis, the interface axial forces are monitored after SRB ignition at each time step of 0.001 s. As soon as the interface axial forces became greater than zero, the constraint equations were modified which resulted in the axial and lateral forces at that contact point going to zero. This is accomplished independently for all eight contact points until the vehicle is separated from the pad. Recontact is not treated in the analysis.

C. Proposed Method

This report proposes a method which incorporates the effects of changing boundary conditions with a transient response analysis [23]. The proposed method uses substructures that are coupled together through interface boundary forces. The boundary forces are approximated by a power series in time with unknown coefficients. The equations of motion of the substructures are solved with unknown coefficients at each time step. The unknown coefficients are obtained by enforcing the compatibility equations of the substructure interfaces. The unknown coefficients can be obtained by a simple matrix multiplication. Once the unknown coefficients are computed, the total response is computed for that time step. Since the compatibility of the substructure's boundary is satisfied at each time step, the changing of boundary conditions can be easily managed by zeroing out the compatibility matrix as a change in constraints occurs.

D. Objectives

The objectives of this report are to formulate, program, and verify the proposed method for its use in the liftoff analysis of the space shuttle vehicle using reduced models. The proposed method will be verified by comparing results with the latest iterative method used by Martin Marietta Corporation. The amount of computer time it takes to perform the liftoff analysis is one important criterion for the evaluation of an analysis method. Two CMS methods to reduce the structural models will be studied using the proposed approach for dynamic response. One method

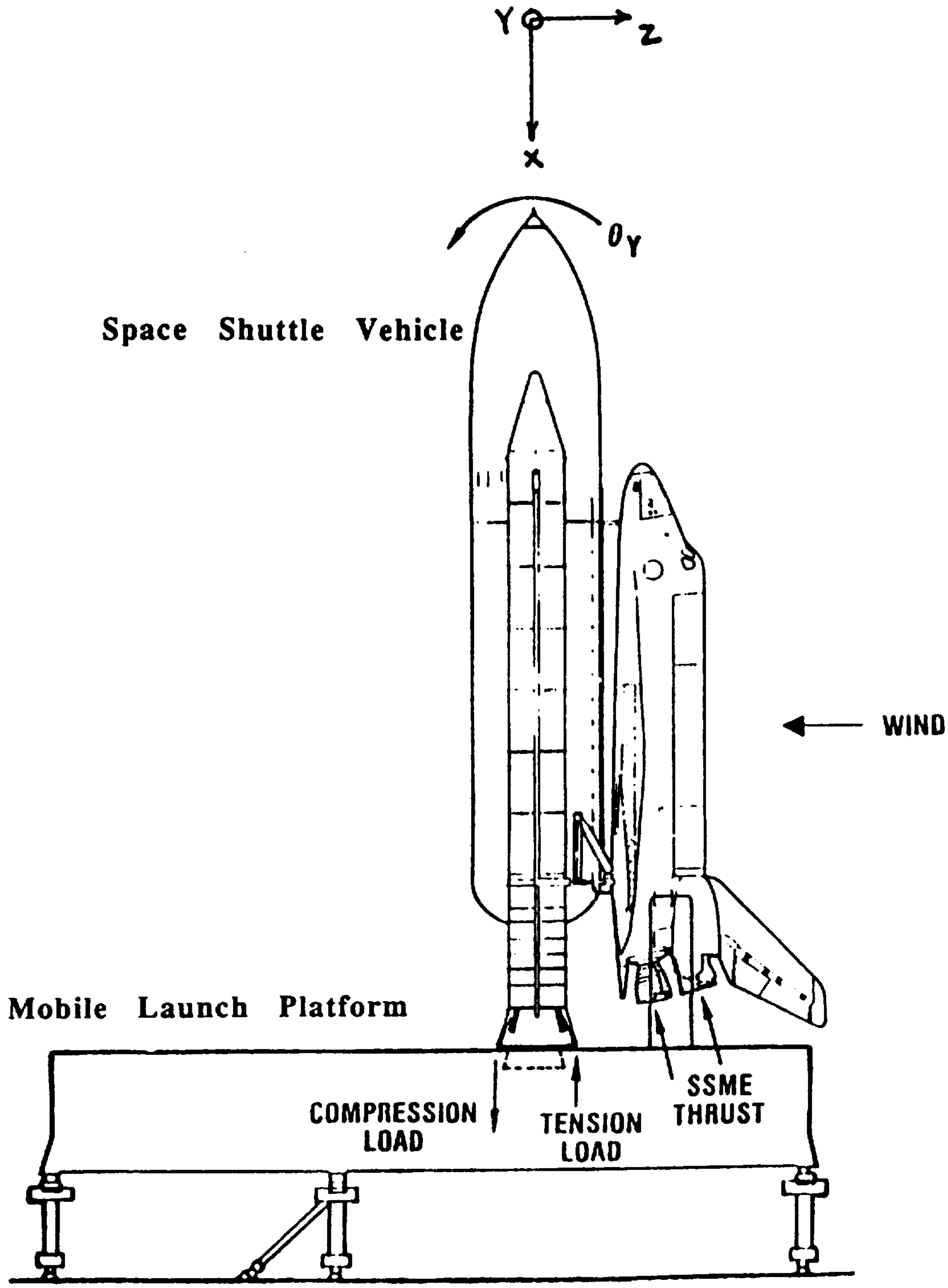


Figure 1. Space shuttle and launch pad structures.

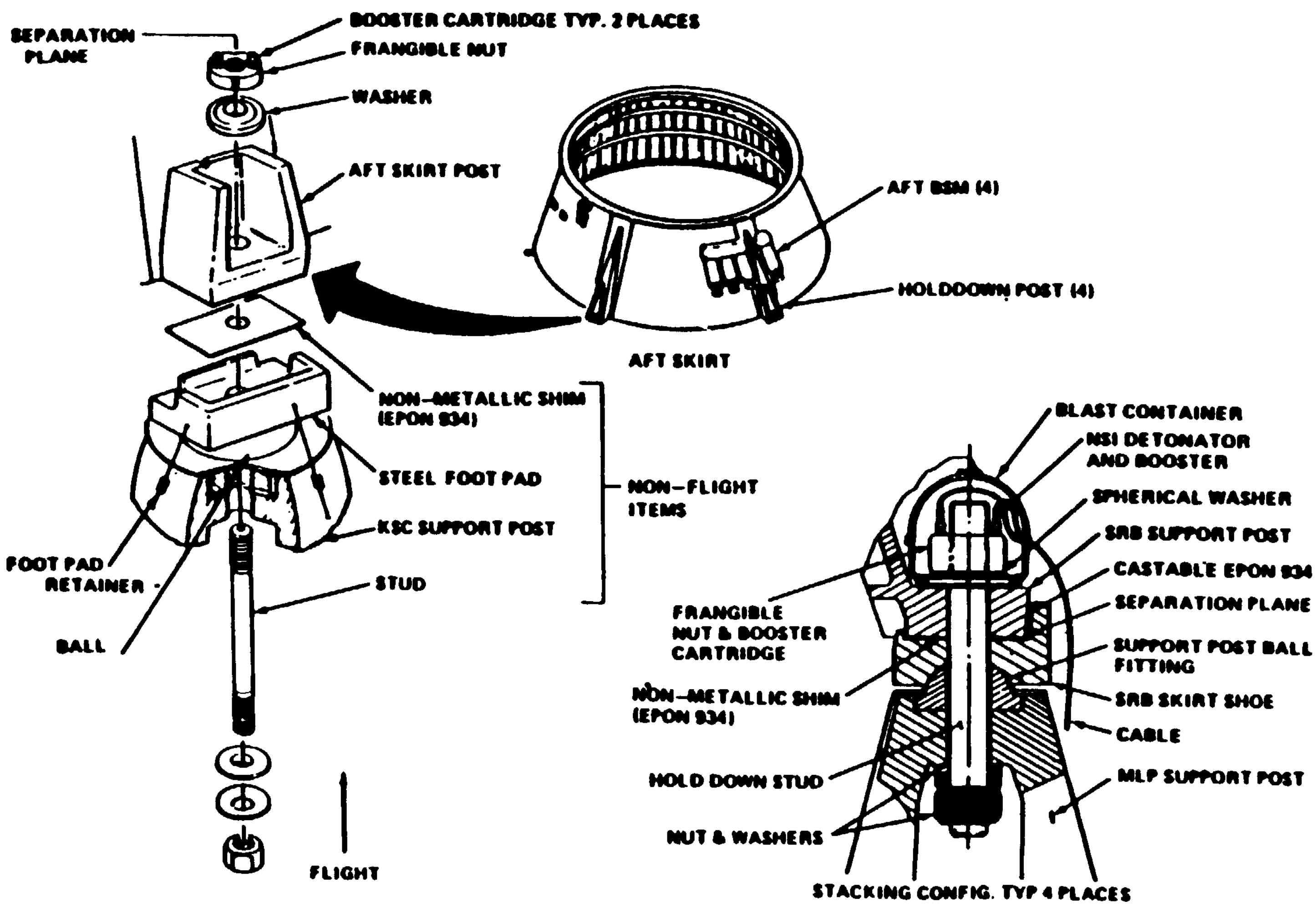
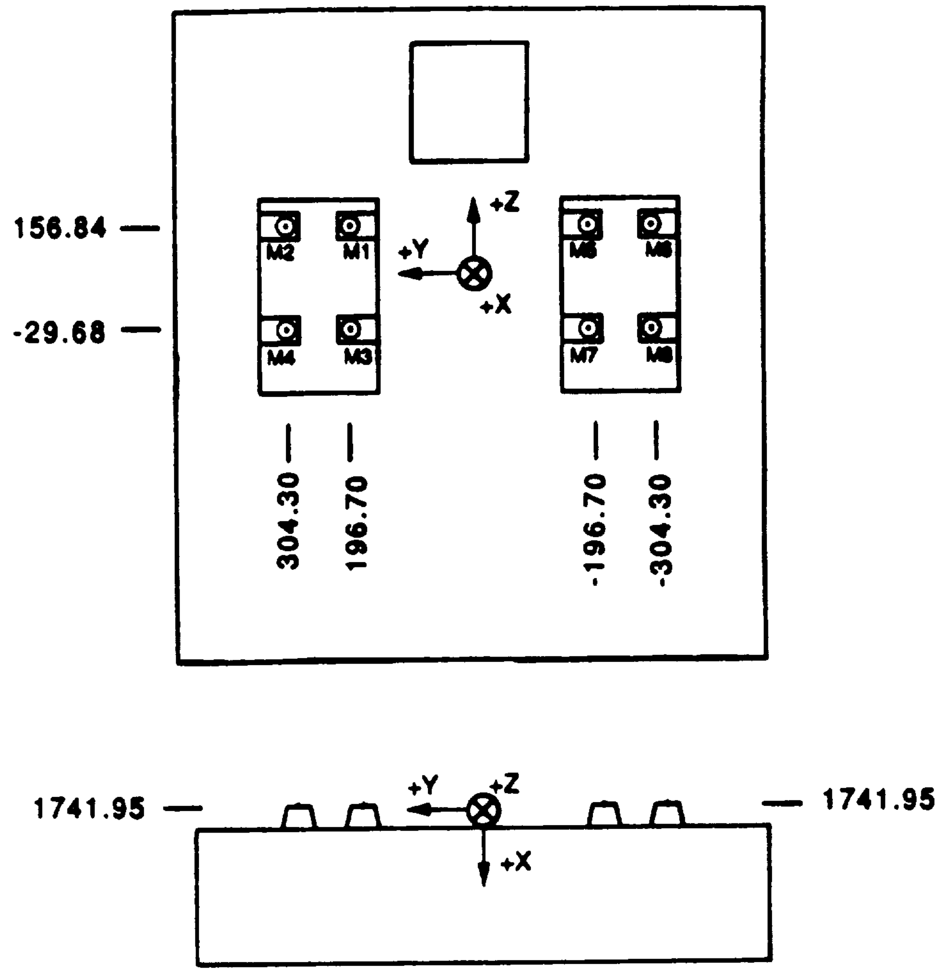


Figure 2. SRB holddown bolt and foot pad.



POST #	X	Y	Z
M1	1741.95	196.70	156.84
M2	1741.95	304.30	156.84
M3	1741.95	196.70	-29.68
M4	1741.95	304.30	-26.68
M5	1741.95	-196.70	156.84
M6	1741.95	-304.30	156.84
M7	1741.95	-196.70	-29.68
M8	1741.95	-304.30	-29.68

Figure 3. SRB/MLP post interface locations in orbiter coordinates.

(termed the Lanczos CMS method), which has been used very effectively on small structural models in recently published references, will be applied to the larger, more complex finite element model of the space shuttle vehicle. The other method (Craig and Bampton) has been used extensively and will serve as a base for comparison.

II. GOVERNING EQUATIONS

The most ardent chore for a dynamic problem is in the formulation of the mathematical model leading to the equations of motion. One way of doing this is through the use of the finite element method. The finite element method can be thought of as a mathematical idealization of the structure's mass, damping, and stiffness. The equations of motion of a structure are a set of linear second-order differential equations. Writing these differential equations in matrix notation, one has:

$$[M]\{\ddot{x}\}+[C]\{\dot{x}\}+[K]\{x\} = \{F\} \tag{1}$$

where $[M]$, $[C]$, and $[K]$ are the mass, damping, and stiffness matrices. When a computer is used to solve equation (1) numerically, the size of the system is a major concern. Reduction and/or uncoupling of equation (1) may be necessary depending on the amount of computer capacity and/or computer time that is available for the task undertaken. One commonly used approach is to break the structure into individual substructures. The substructures can then be reduced using a CMS method [5–12]. The reduced substructures are then rejoined together into a total structural model. This is often done where computer memory capacity is a problem. An example of this would be the space shuttle liftoff model which is comprised of the separate finite element models of the orbiter, ET, and two SRB's.

In this section, the Craig and Bampton CMS method used to reduce equation (1) will be presented. This is followed by a description of the Lanczos CMS method which can also be used in reducing equation (1). Next, the iterative transient response method of Prabhakar [21] will be presented. Finally, the proposed transient response method will be formulated.

A. CMS Method

There are several variations of the CMS method which have been developed [5–12]. One of the more versatile and efficient methods was developed in 1968 by Craig and Bampton [7]. The method has been widely used in structural analysis for reducing finite element models, and it can be easily used on substructures with both determinate and indeterminate interface connections. A brief summary of this method is presented here and the reader is referred to the references for other variations of the CMS method.

Neglecting damping for now, the mass and stiffness matrices of equation (1) can be arranged into interior and boundary coordinates. The equations of motion can then be written as:

$$
\begin{bmatrix} M_{ii} & M_{ib} \\ M_{bi} & M_{bb} \end{bmatrix} \begin{Bmatrix} \ddot{x}_i \\ \ddot{x}_b \end{Bmatrix} + \begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix} \begin{Bmatrix} x_i \\ x_b \end{Bmatrix} = \begin{Bmatrix} F_i \\ F_b \end{Bmatrix} , \tag{2}
$$

where i refers to the interior DOF and b refers to the boundary DOF. Normally, the interior DOF are much larger than the boundary DOF. The Craig and Bampton (CB) method incorporates the Guyan reduction [4] technique discussed earlier with the normal modes computed from the equations of motion of the interior coordinates. Neglecting inertial forces in equation (2) and setting $\{F_i\}$ equal to zero one has:

$$
\begin{bmatrix} K_{ii} & K_{ib} \\ K_{bi} & K_{bb} \end{bmatrix} \begin{Bmatrix} x_i \\ x_b \end{Bmatrix} = \begin{Bmatrix} 0 \\ F_b \end{Bmatrix} . \tag{3}
$$

Solving the top set of equations for $\{x_i\}$, a transformation can then be written as:

$$
\begin{Bmatrix} x_i \\ x_b \end{Bmatrix} = \begin{bmatrix} -[K_{ii}]^{-1}[K_{ib}] \\ I \end{bmatrix} \{x_b\} = \begin{bmatrix} G \\ I \end{bmatrix} \{x_b\} . \tag{4}
$$

This transformation is referred to as static condensation or Guyan reduction where the interior coordinates are expressed in terms of boundary coordinates. If the normal modes are computed for the top set of equations in equation (2), a transformation using the normal modes and the Guyan transformation can be formed. This transformation can be expressed as:

$$
\begin{Bmatrix} x_i \\ x_b \end{Bmatrix} = \begin{bmatrix} \Phi_n & G \\ 0 & I \end{bmatrix} \begin{Bmatrix} q_n \\ x_b \end{Bmatrix} = [T_{CB}] \begin{Bmatrix} q_n \\ x_b \end{Bmatrix} , \tag{5}
$$

where

$$
[T_{CB}] = \begin{bmatrix} \Phi_n & G \\ 0 & I \end{bmatrix}
$$

$\{q_n\}$ = the n generalized coordinates corresponding to $[\Phi_n]$

$[\Phi_n]$ = the n normal modes from the eigen analysis of the interior coordinates

n = the number of normal modes that are kept based on a given cutoff frequency, $n \leq i$

$[G] = -[K_{ii}]^{-1}[K_{ib}]$ = the constraint modes due to the Guyan reduction [4].

Substituting the $[T_{CB}]$ transformation of equation (5) into equation (2) and premultiplying equation (2) by the transpose of $[T_{CB}]$, a set of reduced equations of motion of the structure or substructure is obtained in the form:

$$[M_{CB}] \begin{Bmatrix} \ddot{q}_n \\ \ddot{x}_B \end{Bmatrix} + [K_{CB}] \begin{Bmatrix} q_n \\ x_b \end{Bmatrix} = [T_{CB}]^T \{F\} \quad , \tag{6}$$

where

$$[M_{CB}] = [T_{CB}]^T [M] [T_{CB}] = \begin{bmatrix} I_{qq} & \overline{M}_{qb} \\ \overline{M}_{bq} & \overline{M}_{bb} \end{bmatrix}$$

$$[K_{CB}] = [T_{CB}]^T [K] [T_{CB}] = \begin{bmatrix} \omega^2 & 0 \\ 0 & \overline{K}_{bb} \end{bmatrix}$$

and

$[I_{qq}]$ = unity matrix

$[\overline{M}_{qb}] = [\Phi_n]^T [M_{ii}][G] + [\Phi_n]^T [M_{ib}]$

$[\overline{M}_{bq}] = [G]^T [M_{ii}][\Phi_n] + [M_{ib}][\Phi_n]$

$[\overline{M}_{bb}] = [G]^T [M_{ii}][G] + [G]^T [M_{ib}] + [M_{bi}][G] + [M_{bb}]$

$[\omega^2]$ = diagonal matrix of square of frequency

$[\overline{K}_{bb}] = [G]^T [K_{ii}][G] + [G]^T [K_{ib}] + [K_{bi}][G] + [K_{bb}]$

The reduction of the equations of motion are formulated by using the truncated normal modes $[\Phi_n]$. This introduces some additional approximations into the model with respect to truncated normal modes. However, knowing the frequency content of the applied forces one can make an appropriate selection of the number of normal modes to keep. This will result in a good approximation of the structural dynamic loads from the transient response. It should be noted that the equations of motion are now in both modal and discrete coordinates, q_n and x_b.

B. Lanczos Vectors

The use of Lanczos vectors in the CMS formulation has gained much attention recently because of its less expensive solution time as compared to the eigenvalue problem normally used in CMS. Some simple structures have been tested; these studies have shown that very few Lanczos vectors are needed for good results. In transient response analysis, a major portion of computer time is spent on the reduction of the finite element models. Therefore, it is obvious that if the Lanczos vectors work for a complex structural model a substantial savings in computer time can be obtained.

The paper by Ojalvo [13] gives a brief history of the origin of the Lanczos vectors. Several other papers present various methods of implementing the Lanczos vectors into a reduction transformation [14–17]. The formulation and computational procedure used to compute the Lanczos vectors for this research are taken from the paper by Allen [16].

Lanczos vectors have similarity to the Ritz-type vector formulation. The first Lanczos vector is the static solution of the interior DOF to an applied interior force $\{f_i\}$. The force $\{f_i\}$ is either a unit applied force or can be a randomly generated force vector with values between zero and unity. This is expressed as:

$$\{L^*_i\}_1 = [K_{ii}]^{-1}\{f_i\} \ . \tag{7}$$

This vector is then normalized with respect to the interior mass matrix,

$$\{L_i\}_1 = \frac{\{L^*_i\}_1}{\sqrt{\{L^*_i\}_1^T[M_{ii}]\{L^*_i\}_1}} \ . \tag{8}$$

The next k Lanczos vectors $\{L_i\}_k$ are then computed using the recurrence relationship of:

$$\{L^*_i\}_k = [K_{ii}]^{-1}[M_{ii}]\{L_i\}_{k-1} \ , \tag{9}$$

and

$$\{L_i\}_k = \{L^*_i\}_k - \sum_{j=1}^{k-1} c_j\{L_i\}_j \ , \tag{10}$$

where

$k = 2,\ldots,n$ number of Lanczos vectors $n \leq i$

$c_j = \{L_i\}^T{}_j[M_{ii}]\{L^*{}_i\}_k$.

Once the Lanczos vector is computed it is then normalized with respect to the interior mass matrix as shown in equation (8). The number of Lanczos vectors to be generated will be less than or equal to the size of stiffness matrix, which in this case is i. After the n Lanczos vectors are generated they are assembled into the matrix $[L_n]$ and used in a Lanczos transformation matrix. The Lanczos transformation matrix can be expressed as:

$$\begin{Bmatrix} x_i \\ x_b \end{Bmatrix} = \begin{bmatrix} L_n & G \\ 0 & I \end{bmatrix} \begin{Bmatrix} q_n \\ x_b \end{Bmatrix} = [T_L] \begin{Bmatrix} q_n \\ x_b \end{Bmatrix} \tag{11}$$

where

$$[T_L] = \begin{bmatrix} L_n & G \\ 0 & I \end{bmatrix}$$

$\{q_n\}$ = the n generalized coordinates corresponding to $[L_n]$

$[L_n]$ = the n Lanczos vectors, $n \leq i$

$[G] = -[K_{ii}]^{-1}[K_{ib}]$ = the constraint modes due to the Guyan reduction [4].

The formulation of the Lanczos CMS vector transformation is similar to the Craig and Bampton CMS method. Note that the two transformations are identical in form with the transformation matrix being generated differently, i.e., the former (Craig and Bampton) is by normal modes and the latter (Lanczos) is by static response. Since the Lanczos vectors are computed from a recurrence formula, this eliminates the need to solve the eigenvalue problem for the equations of motion of the interior coordinates.

Substituting equation (11) into equation (2) and premultiplying the resulting equation by the transpose of $[T_L]$, a set of reduced equations of motion of the substructure is obtained in the form:

$$[M_L]\begin{Bmatrix} \ddot{q}_n \\ \ddot{x}_b \end{Bmatrix} + [K_L]\begin{Bmatrix} q_n \\ x_b \end{Bmatrix} = [T_L]^T\{F\} \ , \tag{12}$$

where

$$[M_L] = [T_L]^T[M][T_L] = \begin{bmatrix} I_{Lqq} & \overline{M}_{Lqb} \\ \overline{M}_{Lnq} & \overline{M}_{Lbb} \end{bmatrix}$$

$$[K_L] = [T_L]^T[K][T_L] = \begin{bmatrix} \overline{K}_{Lqq} & \overline{K}_{Lqb} \\ \overline{K}_{Lbq} & \overline{K}_{Lbb} \end{bmatrix}$$

and

$$[I_{Lqq}] = \text{unity matrix}$$

$$[\overline{M}_{Lqb}] = [L_n]^T[M_{ii}][G] + [L_n]^T[M_{ib}]$$

$$[\overline{M}_{Lbq}] = [G]^T[M_{ii}][L_n] + [M_{ib}][L_n]$$

$$[\overline{M}_{Lbb}] = [G]^T[M_{ii}][G] + [G]^T[M_{ib}] + [M_{bi}][G] + [M_{bb}]$$

$$[\overline{K}_{Lqq}] = [L_n]^T[K_{ii}][L_n]$$

$$[\overline{K}_{Lqb}] = [L_n]^T[K_{ii}][G] + [L_n]^T[K_{ib}]$$

$$[\overline{K}_{Lbq}] = [G]^T[K_{ii}][L_n] + [K_{bi}][L_n]$$

$$[\overline{K}_{Lbb}] = [G]^T[K_{ii}][G] + [G]^T[K_{ib}] + [K_{bi}][G] + [K_{bb}] \quad .$$

It should be noted that the off diagonal terms of the reduced stiffness matrix are not zeros as in the case of the Craig and Bampton reduced stiffness matrix. Like the Craig and Bampton method, the reduced equations of motion are in mixed vector and discrete coordinates.

C. Martin Marietta's Transient Response Method with Changing Boundary Conditions

A method of dealing with the transient response for the liftoff of the space shuttle vehicle used by Martin Marietta [21] is presented in this section.

The uncoupled equations of motion of the vehicle and launch pad for the shuttle system shown in figure 1 can be written as:

$$\begin{bmatrix} M_v & 0 \\ 0 & M_p \end{bmatrix} \begin{Bmatrix} \ddot{x}_v \\ \ddot{x}_p \end{Bmatrix} + \begin{bmatrix} K_v & 0 \\ 0 & K_p \end{bmatrix} \begin{Bmatrix} x_v \\ x_p \end{Bmatrix} = \begin{Bmatrix} F_v(t) \\ 0 \end{Bmatrix} \tag{13}$$

where the subscript v refers to the vehicle (space shuttle system) and p refers to the pad (MLP). If the interface nodal coordinates between the pad and vehicle are assumed to be massless, then the coupling stiffness represents the only physical attachments between the vehicle and pad. The justification for this is that the elastic forces at the attachments are assumed to be much larger than the inertia forces. Equation (13), with the added coupling stiffness overlaying the interface DOF for the vehicle and pad, can then be written as:

$$\begin{bmatrix} M_v & 0 \\ 0 & M_p \end{bmatrix} \begin{Bmatrix} \ddot{x}_v \\ \ddot{x}_p \end{Bmatrix} + \left[\begin{bmatrix} K_v & 0 \\ 0 & K_p \end{bmatrix} + [\hat{K}_c] \right] \begin{Bmatrix} x_v \\ x_p \end{Bmatrix} = \begin{Bmatrix} F_v(t) \\ 0 \end{Bmatrix} \tag{14a}$$

where $[\hat{K}_c]$ is the coupling stiffness between the vehicle and pad interfaces. The physical significance of the coupling stiffness is to constrain the contact DOF to move together. If equation (14a) is rewritten separating the interior DOF from the boundary (interface) DOF, it becomes clear how the coupling stiffness matrix couples the two structures together at the interface coordinates. Rewriting equation (14a) as:

$$\begin{bmatrix} M_{Vii} & M_{Vib} & 0 & 0 \\ M_{Vbi} & M_{Vbb} & 0 & 0 \\ 0 & 0 & M_{Pii} & M_{Pib} \\ 0 & 0 & M_{Pbi} & M_{Pbb} \end{bmatrix} \begin{Bmatrix} \ddot{x}_{Vi} \\ \ddot{x}_{Vb} \\ \cdots \\ \ddot{x}_{Pi} \\ \ddot{x}_{Pb} \end{Bmatrix} + \left[\begin{bmatrix} K_{Vii} & K_{Vib} & 0 & 0 \\ K_{Vbi} & K_{Vbb} & 0 & 0 \\ 0 & 0 & K_{Pii} & K_{Pib} \\ 0 & 0 & K_{Pbi} & K_{Pbb} \end{bmatrix} \right.$$

$$\left. + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & K_C & 0 & -K_C \\ 0 & 0 & 0 & 0 \\ 0 & -K_C & 0 & K_C \end{bmatrix} \right] \begin{Bmatrix} x_{Vi} \\ x_{Vb} \\ \cdots \\ x_{Pi} \\ x_{Pb} \end{Bmatrix} = \begin{Bmatrix} F_{Vi}(t) \\ F_{Vb}(t) \\ \cdots \\ 0 \\ 0 \end{Bmatrix} \tag{14b}$$

where x_{Vi} and x_{Pi} are the interior coordinates, x_{Vb} and x_{Pb} are the boundary (interface) coordinates, and

$$[\hat{K}_c] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & K_C & 0 & -K_C \\ 0 & 0 & 0 & 0 \\ 0 & -K_C & 0 & K_C \end{bmatrix}$$

The submatrices K_c are diagonal representing the stiffnesses between interface coordinates. The coupling stiffness matrix for the space shuttle can be represented by the bolt stiffnesses between the vehicle and pad. In figure 2, the bolt stiffnesses used in this study are $K_x = 31 \times 10^6$ lbf/in for x-direction, $K_y = 4 \times 10^6$ lbf/in for y-direction, and $K_z = 5.5 \times 10^6$ lbf/in for the z-direction. The same stiffness values are used for both the vehicle and pad sides. This means a 6-DOF stiffness matrix per contact point. A total of eight contact points are between the space shuttle vehicle and the MLP. Therefore, the coupling stiffness matrix is 48 by 48. The coupling stiffness matrix has the same form as a single spring matrix with 2 DOF. For example, in the x-direction at one attachment point, the coupling stiffness matrix has the form:

$$[K_{cx}] = \begin{bmatrix} K_x & -K_x \\ -K_x & K_x \end{bmatrix} .$$

The use of a coupling stiffness matrix has the advantage of allowing the vehicle to separate from the pad. This is accomplished by zeroing out the relevant attachment stiffness values in the coupling stiffness matrix $[\hat{K}_c]$ after the contact forces have gone into tension. This will become clearer after the iteration method is presented.

An eigen analysis of equation (13) results in the eigenvalues and eigenvectors of the vehicle and pad. A transformation of the coordinates can be written in terms of the eigenvectors as follows:

$$\begin{Bmatrix} x_v \\ x_p \end{Bmatrix} = \begin{bmatrix} \Phi_v & 0 \\ 0 & \Phi_p \end{bmatrix} \begin{Bmatrix} q_v \\ q_p \end{Bmatrix} , \tag{15}$$

where

$$\Phi_v^T M_v \Phi_v = I_v \quad ,$$

$$\Phi_p^T M_p \Phi_p = I_p \quad ,$$

$$\Phi_V^T K_V \Phi_V = \omega_V^2 \quad ,$$

$$\Phi_p^T K_p \Phi_p = \omega_p^2 \quad ,$$

and I_V and I_p are unity matrices. Substituting equation (15) into equation (14) and premultiplying the resulting equation by the transpose of the transformation matrix given by equation (15) gives the set of uncoupled differential equations:

$$\begin{bmatrix} I_V & 0 \\ 0 & I_P \end{bmatrix} \begin{Bmatrix} \ddot{q}_V \\ \ddot{q}_P \end{Bmatrix} + \begin{bmatrix} \omega_V^2 & 0 \\ 0 & \omega_P^2 \end{bmatrix} \begin{Bmatrix} q_V \\ q_P \end{Bmatrix} = \begin{bmatrix} \Phi_V^T & 0 \\ 0 & \Phi_P^T \end{bmatrix} \begin{Bmatrix} F_V(T) \\ 0 \end{Bmatrix}$$

$$\begin{bmatrix} \Phi_V^T & 0 \\ 0 & \Phi_P^T \end{bmatrix} [\hat{K}_C] \begin{bmatrix} \Phi_V & 0 \\ 0 & \Phi_P \end{bmatrix} \begin{Bmatrix} q_V \\ q_P \end{Bmatrix} \qquad (16)$$

where $[\hat{K}_C]$ is the coupling stiffness matrix.

The coupling stiffness term on the right hand side of equation (16) represents the contact forces between the vehicle and pad. If the contact force is denoted as $\{F_c\}$:

$$\{F_c\} = - \begin{bmatrix} \Phi_V^T & 0 \\ 0 & \Phi_P^T \end{bmatrix} [\hat{K}_C] \begin{bmatrix} \Phi_V & 0 \\ 0 & \Phi_P \end{bmatrix} \begin{Bmatrix} q_V \\ q_P \end{Bmatrix} \qquad (17)$$

and the applied forces are denoted as:

$$\{F_A\} = \begin{bmatrix} \Phi_V^T & 0 \\ 0 & \Phi_P^T \end{bmatrix} \begin{Bmatrix} F_V(t) \\ 0 \end{Bmatrix} \qquad (18)$$

After substituting equations (17) and (18) into equation (16) and making use of the orthogonality condition and then dropping the subscripts v and p for simplicity, equation (16) can be rewritten as:

$$\ddot{q}_j + 2\zeta_j\omega_j\dot{q}_j + \omega_j^2 q_j = F_{Aj} + F_{cj} \quad , \tag{19a}$$

for $(j = 1,\ldots,p+v)$, where ζ_j is the modal damping ratio. Equation (19a) can be simplified further by dropping the subscript j and this gives:

$$\ddot{q} + 2\zeta\omega\dot{q} + \omega^2 q = F_A + F_c \quad , \tag{19b}$$

for $(q = q_1,\ldots,q_{p+v})$. At a given time $t = t_0$, the initial conditions of the structure are known or can be computed. Initial conditions q_0,q_0 can be obtained from the coordinate transformation given by equation (15). The applied forces $\{F_{A0}\}$ are known at time $t = t_0$. The initial contact forces $\{F_{c0}\}$ are computed from the coupling stiffness matrix and the initial pad displacements. If the applied force (i.e., $F_A + F_c$) is approximated by $A + B\tau$, then equation (19) can be written as:

$$\ddot{q} + 2\zeta\omega\dot{q} + \omega^2 q = A + B\tau \tag{20}$$

for $(q = q_1,\ldots,q_{p+v})$.

Integrating equation (19) over the time interval of h where $t = t_0 + h$ and comparing equation (20) with equation (19b) at $\tau = 0$ the coefficient A is determined,

$$A = F_{A0} + F_{c0} \quad , \tag{21}$$

and for $\tau = h$ the coefficient B is determined,

$$\mathrm{B} = \frac{F_A - F_{A0}}{h} + \frac{F_c - F_{c0}}{h} \quad . \tag{22}$$

A closed-form solution of equation (20) can be obtained in terms of A and B. For those q's with $\omega = 0$ (rigid body motion) the solution is:

$$\ddot{q} = A + B\tau$$

$$\dot{q} = A\tau + \frac{B\tau^2}{2} + C_1$$

$$q = \frac{A\tau^2}{2} + \frac{B\tau^3}{6} + C_1\tau + C_2 \quad , \tag{23}$$

where

$$C_1 = q_0 \qquad \text{and} \qquad C_2 = \dot{q}_0 \quad ,$$

and for those q's where $\omega \neq 0$ the solution is:

$$q = e^{-\zeta\omega\tau}[K_1 \cos \omega_d\tau + K_2 \sin \omega_d\tau] + K_3 + K_4\tau \quad ,$$

$$q = (-\zeta\omega)e^{-\zeta\omega\tau}[K_1 \cos \omega_d\tau + K_2 \sin \omega_d\tau] + e^{-\zeta\omega\tau}[-K_1\omega_d \sin \omega_d\tau + K_2\omega_d \cos \omega_d\tau] + K_4 \quad ,$$

$$\ddot{q} = (-\zeta\omega)^2 e^{-\zeta\omega\tau}[K_1 \cos \omega_d\tau + K_2 \sin \omega_d\tau] + 2(-\zeta\omega)e^{-\zeta\omega\tau}[-K_1\omega_d \sin \omega_d\tau + K_2\omega_d \cos \omega_d\tau]$$
$$- e^{-\zeta\omega\tau}[K_1\omega_d^2 \cos \omega_d\tau + K_2\omega_d^2 \sin \omega_d\tau] \quad , \tag{24}$$

where

$$\omega_d = \omega\sqrt{1-\zeta^2}$$

$$K_1 = \left(q_0 - \frac{1}{\omega^2}\left(A - 2\zeta\omega\,\frac{B}{\omega^2}\right)\right)$$

$$K_2 = (\dot{q}_0 + \zeta\omega K_1 - K_4)$$

$$K_3 = \left(A - 2\zeta\omega\,\frac{B}{\omega^2}\right)$$

$$K_4 = \frac{B}{\omega^2}$$

$$q = q_1,\ldots,q_{p+v} \ , \quad \zeta = \zeta_1,\ldots,\zeta_{p+v} \ , \quad \omega = \omega_1,\ldots,\omega_{p+v} \quad .$$

The iterative procedure begins with an estimate of the contact force at $t = t_0+h$. Coefficients A and B can then be computed from equations (21) and (22). The coefficients are then used in equation (23) or (24) to obtain an estimate of the $\{q\}$'s. Contact forces between the vehicle and pad can then be computed. A check of the vehicle-to-pad separation is made and, if the vehicle has separated from the pad (i.e., bolt loads are in tension), the coupling stiffness is modified. The $\{q\}$'s along with the $[\hat{K}_c]$ stiffness matrix are then used to compute new values of the B coefficients (i.e., contact forces). The new B coefficients are then used again in equation (23) or (24) over the same time interval for a better estimate of the $\{q\}$'s. The process is continued until the change in contact force $\{F_c\}$ is within a specified tolerance. The procedure described has been used to simulate the space shuttle liftoff transient response and also some barge docking impact transient response analysis with success by Martin Marietta [21].

D. Proposed Transient Response Method With Changing Boundary Conditions

The proposed method is presented for a general structure and can be readily applied to the space shuttle liftoff transient response analysis. A transient response method dealing with the effects of changing boundary conditions for linearly coupled substructures [23] is proposed. The proposed method is applicable to any number of substructures as will become evident. In the following derivation only two substructures will be used. The term linear refers to the substructures which behave linearly and to all the forces, damping, applied, and interface, that are linear functions of the coordinate variables. The substructure equations of motion are assumed to be reduced by one of the CMS transformations in section II, either by equation (5) or by equation (11).

A structure can be divided into two substructures as shown in figure 4. The equations of motion for the undamped substructures A and B in matrix forms are respectively:

$$[M_A]\begin{Bmatrix} \ddot{x}_A \\ \ddot{x}_{AI} \end{Bmatrix} + [K_A]\begin{Bmatrix} x_A \\ x_{AI} \end{Bmatrix} = \begin{Bmatrix} F_A \\ F_{AI} \end{Bmatrix} + \begin{Bmatrix} 0 \\ F_I \end{Bmatrix}_A \tag{25}$$

$$[M_B]\begin{Bmatrix} \ddot{x}_B \\ \ddot{x}_{BI} \end{Bmatrix} + [K_B]\begin{Bmatrix} x_B \\ x_{BI} \end{Bmatrix} = \begin{Bmatrix} F_B \\ F_{BI} \end{Bmatrix} + \begin{Bmatrix} 0 \\ F_I \end{Bmatrix}_B \tag{26}$$

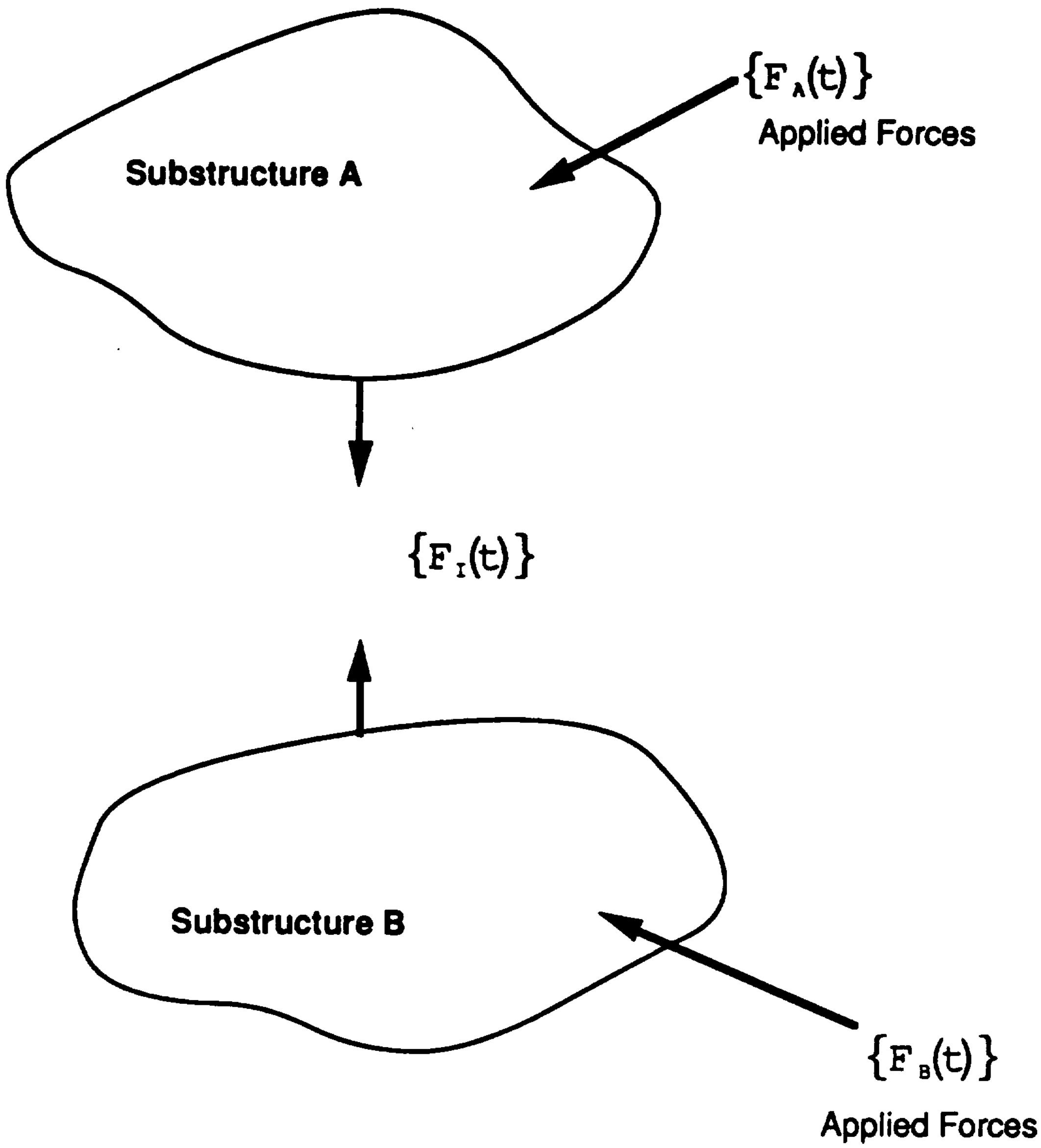


Figure 4. Free-body diagram of two substructures.

In the above equations

$$\begin{Bmatrix} 0 \\ F_I \end{Bmatrix}_A \quad \text{and} \quad \begin{Bmatrix} 0 \\ F_I \end{Bmatrix}_B$$

represent the interface forces acting on the interface coordinates of each substructure, and

$$\begin{Bmatrix} F_A \\ F_{AI} \end{Bmatrix} \quad \text{and} \quad \begin{Bmatrix} F_B \\ F_{BI} \end{Bmatrix}$$

are the applied forces. The subscript I refers to the interface coordinates. The compatibility conditions for the substructures shown in figure 4 can be written as:

$$\{x_{AI}\} \equiv \{x_{BI}\} \tag{27}$$

and

$$\{F_I\}_A + \{F_I\}_B = 0 \quad . \tag{28}$$

For simplicity, the interface forces will be designated as $\{F_I\}$, i.e., $\{F_I\}_A = -\{F_I\}_B = \{F_I\}$. Equations (25) and (26) are inertially and elastically coupled. To uncouple the equations, the normal modes of the substructures are computed and then a modal transformation is used. This can be expressed as:

$$\begin{Bmatrix} x_A \\ x_{AI} \end{Bmatrix} = [\Phi_A]\ \{q_A\} \tag{29}$$

and

$$\begin{Bmatrix} x_B \\ x_{BI} \end{Bmatrix} = [\Phi_B]\ \{q_B\} \tag{30}$$

This transformation will change the mixed coordinates of equations (25) and (26) into modal coordinates which are much easier to integrate from a computational sense. This step is similar to that done in the previous section. It is at this point that damping can be introduced into the equations of motion. In the numerical examples presented in section IV, the modal damping ratios ζ are assumed to be constant for all frequencies and for both substructures in the formulation. Then equations (25) and (26) can be rewritten as:

$$\{\ddot{q}_A\} + 2\zeta[\omega_A]\{\dot{q}_A\} + [\omega_A^2]\{q_A\} = [\Phi_A]^T \begin{Bmatrix} F_A \\ F_{AI} \end{Bmatrix} + [\Phi_A]^T \begin{Bmatrix} 0 \\ F_I \end{Bmatrix} , \tag{31}$$

and

$$\{\ddot{q}_B\} + 2\zeta[\omega_B]\{\dot{q}_B\} + [\omega_B^2]\{q_B\} = [\Phi_B]^T \begin{Bmatrix} F_B \\ F_{BI} \end{Bmatrix} - [\Phi_B]^T \begin{Bmatrix} 0 \\ F_I \end{Bmatrix} . \tag{32}$$

Within each time interval of integration step the interface forces between the substructures in equations (31) and (32) are treated like applied forces. If equations (31) and (32) are solved for in a time step manner, then an approximation can be made for the interface forces by the use of a power series which is valid for a time step Δt. This is expressed as:

$$\{F_I\} = \sum_{j=0}^{\infty} G_j(t-t_i)^j \qquad t_i \leq t \leq t_i + \Delta t \quad , \tag{33}$$

where G_j are unknown coefficients to be determined. A series is expected to converge rather rapidly for the size of Δt normally used in the integration of equations (31) and (32). It is sufficient that four terms of the series are kept, then the interface forces can be written as:

$$\{F_I\} = \{G_0\} + \{G_1\}(t-t_i) + \{G_2\}(t-t_i)^2 + \{G_3\}(t-t_i)^3 \quad , \tag{34}$$

for $t_i \leq t \leq t_i + \Delta t$. Substituting equation (34) into equations (31) and (32) results in the equations:

$$\{\ddot{q}_A\} + 2\zeta[\omega_A]\{\dot{q}_A\} + [\omega_A^2]\{q_A\} = [\Phi_A]^T \begin{Bmatrix} F_A \\ F_{AI} \end{Bmatrix} + [\Phi_A^*]^T(\{G_0\} + \{G_1\}(t-t_i)$$

$$+ \{G_2\}(t-t_i)^2 + \{G_3\}(t-t_i)^3) \tag{35}$$

and

$$\{\ddot{q}_B\} + 2\zeta[\omega_B]\{\dot{q}_B\} + [\omega_B^2]\{q_B\} = [\Phi_B]^T \begin{Bmatrix} F_B \\ F_{BI} \end{Bmatrix} + [\Phi_B^*]^T(\{G_0\} + \{G_1\}(t-t_i)$$

$$+ \{G_2\}(t-t_i)^2 + \{G_3\}(t-t_i)^3) \quad . \tag{36}$$

for $t_i \leq t \leq t_i + \Delta t$. The * superscript used on the substructure modes is defined as follows for substructures A and B:

$$[\Phi_A]^T \begin{Bmatrix} 0 \\ F_I \end{Bmatrix} = [[\overline{\Phi}_A]^T : [\Phi_A^*]^T] \begin{Bmatrix} 0 \\ \cdots \\ F_I \end{Bmatrix} = [\Phi_A^*]^T\{F_I\} \quad ,$$

$$[\Phi_B]^T \begin{Bmatrix} 0 \\ F_I \end{Bmatrix} = [[\overline{\Phi}_B]^T : [\Phi_B^*]^T] \begin{Bmatrix} 0 \\ \cdots \\ F_I \end{Bmatrix} = [\Phi_B^*]^T\{F_I\} \quad .$$

Using superposition of solutions, the modal coordinates of the substructures can be split up into two parts:

$$\{q_A\} \equiv \{q_{A1}\} + \{q_{A2}\} \quad , \tag{37}$$

and

$$\{q_B\} \equiv \{q_{B1}\} + \{q_{B2}\} \quad . \tag{38}$$

Substituting equations (37) and (38) along with their derivatives into equations (35) and (36) results in the following four sets of equations:

$$\{\ddot{q}_{A1}\} + 2\zeta[\omega_A]\{\dot{q}_{A1}\} + [\omega_A^2]\{q_{A1}\} = [\Phi_A]^T \begin{Bmatrix} F_A \\ F_{AI} \end{Bmatrix} + [\Phi_A^*]^T\{G_0\} \quad , \tag{39a}$$

$$\{\ddot{q}_{A2}\} + 2\zeta[\omega_A]\{\dot{q}_{A2}\} + [\omega_A^2]\{q_{A2}\} = [\Phi_A^*]^T(\{G_1\}(t-t_i) + \{G_2\}(t-t_i)^2 + \{G_3\}(t-t_i)^3) \quad , \tag{39b}$$

$$\{\ddot{q}_{B1}\} + 2\zeta[\omega_B]\{\dot{q}_{B1}\} + [\omega_B^2]\{q_{B1}\} = [\Phi_B]^T \begin{Bmatrix} F_B \\ F_{BI} \end{Bmatrix} - [\Phi_B^*]^T\{G_0\} \quad , \tag{39c}$$

$$\{\ddot{q}_{B2}\} + 2\zeta[\omega_B]\{\dot{q}_{B2}\} + [\omega_B^2]\{q_{B2}\} = -[\Phi_B^*]^T(\{G_1\}(t-t_i) + \{G_2\}(t-t_i)^2 + \{G_3\}(t-t_i)^3) \quad . \tag{39d}$$

Set the initial conditions for equation (39) as follows:

$$\{q_{A1}(t_i)\} = \{q_A(t_i)\} \qquad \{q_{A2}(t_i)\} = \{0\} \quad , \tag{40a}$$

$$\{\dot{q}_{A1}(t_i)\} = \{\dot{q}_A(t_i)\} \qquad \{\dot{q}_{A2}(t_i)\} = \{0\} \quad , \tag{40b}$$

$$\{q_{B1}(t_i)\} = \{q_B(t_i)\} \qquad \{q_{B2}(t_i)\} = \{0\} \quad , \tag{40c}$$

$$\{\dot{q}_{B1}(t_i)\} = \{\dot{q}_B(t_i)\} \qquad \{\dot{q}_{B2}(t_i)\} = \{0\} \quad . \tag{40d}$$

Substituting $t = t_i$ into equation (34) gives the first term in the power series as:

$$\{G_0\} = \{F_I(t_i)\} \quad . \tag{41}$$

Therefore, a closed-form solution of equations (39a) and (39c) can be obtained at $t_{i+1} = t_i + \Delta t$ using the initial conditions of equations (40) and (41). Thus, it allows one to compute the following quantities,

$$\{q_{A1}(t_{i+1})\} \ ; \qquad \{\dot{q}_{A1}(t_{i+1})\} \ ; \qquad \{\ddot{q}_{A1}(t_{i+1})\} \quad , \tag{42a}$$

$$\{q_{B1}(t_{i+1})\} \ ; \qquad \{\dot{q}_{B1}(t_{i+1})\} \ ; \qquad \{\ddot{q}_{B1}(t_{i+1})\} \quad , \tag{42b}$$

which will be needed in equation (47).

Equations (39b) and (39d) can be solved in a closed-form solution using the initial conditions in equation (40), however their solutions contain the unknown coefficients $\{G_1\}$, $\{G_2\}$, and $\{G_3\}$. By assigning a unit value to the coefficients one at a time and solving equations (39b) and (39d), a solution is obtained in terms of the coefficients. The results can be written in matrix notation as:

$$\{q_{A2}(t_{i+1})\} = [\bar{C}_{A11}]\{G_1\} + [\bar{C}_{A12}]\{G_2\} + [\bar{C}_{A13}]\{G_3\} \quad ,$$

$$\{\dot{q}_{A2}(t_{i+1})\} = [\bar{C}_{A21}]\{G_1\} + [\bar{C}_{A22}]\{G_2\} + [\bar{C}_{A23}]\{G_3\} \quad ,$$

$$\{\ddot{q}_{A2}(t_{i+1})\} = [\bar{C}_{A31}]\{G_1\} + [\bar{C}_{A32}]\{G_2\} + [\bar{C}_{A33}]\{G_3\} \quad , \tag{43}$$

and

$$\{q_{B2}(t_{i+1})\} = [\bar{C}_{B11}]\{G_1\} + [\bar{C}_{B12}]\{G_2\} + [\bar{C}_{B13}]\{G_3\} \quad ,$$

$$\{\dot{q}_{B2}(t_{i+1})\} = [\bar{C}_{B21}]\{G_1\} + [\bar{C}_{B22}]\{G_2\} + [\bar{C}_{B23}]\{G_3\} \quad ,$$

$$\{\ddot{q}_{B2}(t_{i+1})\} = [\bar{C}_{B31}]\{G_1\} + [\bar{C}_{B32}]\{G_2\} + [\bar{C}_{B33}]\{G_3\} \quad . \tag{44}$$

Each element in the $[\bar{C}]$ matrices, which is referred to as the interface compatibility matrix, represents the solution to an assigned unit value of the coefficient.

The coefficients $\{G_1\}$, $\{G_2\}$, and $\{G_3\}$ can be evaluated from the interface compatibility condition stated in equation (27) at the end of each time step Δt. It gives:

$$\{x_{AI}(t_{i+1})\} = \{x_{BI}(t_{i+1})\} \quad , \tag{45a}$$

$$\{\dot{x}_{AI}(t_{i+1})\} = \{\dot{x}_{BI}(t_{i+1})\} \quad , \tag{45b}$$

$$\{\ddot{x}_{AI}(t_{i+1})\} = \{\ddot{x}_{BI}(t_{i+1})\} \quad . \tag{45c}$$

Equation (45) can be expressed in terms of the unknown coefficients by using the modal transformation given by equations (29) and (30) along with equations (37) and (38). Thus, equation (45) can be written as:

$$[\Phi_A^*](\{q_{A1}(t_{i+1})\} + \{q_{A2}(t_{i+1})\}) = [\Phi_B^*](\{q_{B1}(t_{i+1})\} + \{q_{B2}(t_{i+1})\}) \quad , \tag{46a}$$

$$[\Phi_A^*](\{\dot{q}_{A1}(t_{i+1})\} + \{\dot{q}_{A2}(t_{i+1})\}) = [\Phi_B^*](\{\dot{q}_{B1}(t_{i+1})\} + \{\dot{q}_{B2}(t_{i+1})\}) \quad , \tag{46b}$$

$$[\Phi_A{}^*](\{\ddot{q}_{A1}(t_{i+1})\}+\{\ddot{q}_{A2}(t_{i+1})\}) = [\Phi_B{}^*](\{\ddot{q}_{B1}(t_{i+1})\}+\{\ddot{q}_{B2}(t_{i+1})\}) \quad . \tag{46c}$$

Rearranging equation (46) so that the terms due to response of the externally applied forces on the left hand side, yields:

$$[\Phi_A{}^*]\{q_{A1}(t_{i+1})\}-[\Phi_B{}^*]\{q_{B1}(t_{i+1})\} = [\Phi_B{}^*]\{q_{B2}(t_{i+1})\}-[\Phi_A{}^*]\{q_{A2}(t_{i+1})\} \quad , \tag{47a}$$

$$[\Phi_A{}^*]\{\dot{q}_{A1}(t_{i+1})\}-[\Phi_B{}^*]\{\dot{q}_{B1}(t_{i+1})\} = [\Phi_B{}^*]\{\dot{q}_{B2}(t_{i+1})\}-[\Phi_A{}^*]\{\dot{q}_{A2}(t_{i+1})\} \quad , \tag{47b}$$

$$[\Phi_A{}^*]\{\ddot{q}_{A1}(t_{i+1})\}-[\Phi_B{}^*]\{\ddot{q}_{B1}(t_{i+1})\} = [\Phi_B{}^*]\{\ddot{q}_{B2}(t_{i+1})\}-[\Phi_A{}^*]\{\ddot{q}_{A2}(t_{i+1})\} \quad . \tag{47c}$$

The terms on the left hand side which represent the difference in displacement, velocity, and acceleration of the two substructures at their interface due to externally applied forces can be obtained from equation (42). The left hand side terms of equation (47) can be rewritten as:

$$\{\delta(t_{i+1})\} = [\Phi_A{}^*]\{q_{A1}(t_{i+1})\}-[\Phi_B{}^*]\{q_{B1}(t_{i+1})\} \quad , \tag{48a}$$

$$\{\dot{\delta}(t_{i+1})\} = [\Phi_A{}^*]\{\dot{q}_{A1}(t_{i+1})\}-[\Phi_B{}^*]\{\dot{q}_{B1}(t_{i+1})\} \quad , \tag{48b}$$

$$\{\ddot{\delta}(t_{i+1})\} = [\Phi_A{}^*]\{\ddot{q}_{A1}(t_{i+1})\}-[\Phi_B{}^*]\{\ddot{q}_{B1}(t_{i+1})\} \quad . \tag{48c}$$

Substituting equations (43), (44), and (48) into equation (47) gives:

$$\begin{aligned}\{\delta(t_{i+1})\} = {} & ([\Phi_B^*][\overline{C}_{B11}]-[\Phi_A^*][\overline{C}_{A11}])\{G_1\} \\ & + ([\Phi_B^*][\overline{C}_{B12}]-[\Phi_A^*][\overline{C}_{A12}])\{G_2\} \\ & + ([\Phi_B^*][\overline{C}_{B13}]-[\Phi_A^*][\overline{C}_{A13}])\{G_3\} \quad ,\end{aligned} \tag{49a}$$

$$\begin{aligned}\{\dot{\delta}(t_{i+1})\} = {} & ([\Phi_B^*][\overline{C}_{B21}]-[\Phi_A^*][\overline{C}_{A21}])\{G_1\} \\ & + ([\Phi_B^*][\overline{C}_{B22}]-[\Phi_A^*][\overline{C}_{A22}])\{G_2\} \\ & + ([\Phi_B^*][\overline{C}_{B23}]-[\Phi_A^*][\overline{C}_{A23}])\{G_3\} \quad ,\end{aligned} \tag{49b}$$

$$\{\ddot{\delta}(t_{i+1})\} = ([\Phi_B^*][\overline{C}_{B31}] - [\Phi_A^*][\overline{C}_{A31}])\{G_1\}$$
$$+ ([\Phi_B^*][\overline{C}_{B32}] - [\Phi_A^*][\overline{C}_{A32}])\{G_2\}$$
$$+ ([\Phi_B^*][\overline{C}_{B33}] - [\Phi_A^*][\overline{C}_{A33}])\{G_3\} \quad . \tag{49c}$$

Equation (49) can be combined into a single matrix equation as:

$$\begin{Bmatrix} \{\delta(t_{i+1})\} \\ \{\dot{\delta}(t_{i+1})\} \\ \{\ddot{\delta}(t_{i+1})\} \end{Bmatrix} = \begin{bmatrix} [\overline{C}_{11}] & [\overline{C}_{12}] & [\overline{C}_{13}] \\ [\overline{C}_{21}] & [\overline{C}_{22}] & [\overline{C}_{23}] \\ [\overline{C}_{31}] & [\overline{C}_{32}] & [\overline{C}_{33}] \end{bmatrix} \begin{Bmatrix} \{G_1\} \\ \{G_2\} \\ \{G_3\} \end{Bmatrix} \quad . \tag{50}$$

or

$$\begin{Bmatrix} \{\delta(t_{i+1})\} \\ \{\dot{\delta}(t_{i+1})\} \\ \{\ddot{\delta}(t_{i+1})\} \end{Bmatrix} = [\overline{C}] \begin{Bmatrix} \{G_1\} \\ \{G_2\} \\ \{G_3\} \end{Bmatrix} \quad . \tag{51}$$

By inverting the interface compatibility matrix $[\overline{C}]$, the unknown coefficients can be computed as:

$$\begin{Bmatrix} \{G_1\} \\ \{G_2\} \\ \{G_3\} \end{Bmatrix} = [\overline{C}]^{-1} \begin{Bmatrix} \{\delta(t_{i+1})\} \\ \{\dot{\delta}(t_{i+1})\} \\ \{\ddot{\delta}(t_{i+1})\} \end{Bmatrix} \quad . \tag{52}$$

The size of the interface compatibility matrix is directly related to the number of interface coordinates between the substructures and the number of terms kept in the power series that approximates the interface forces. Thus, the interface compatibility matrix will be relatively small and can easily be inverted. It is important to note that the interface compatibility matrix does not change in time as long as the same time step Δt is used. Therefore, the interface compatibility matrix and inversion need to be computed only once at the beginning of the integration.

From the development of these equations, it is obvious that a change in boundary conditions can be performed during the integration of the equations of motion. This change can be accomplished for the liftoff analysis of the space shuttle vehicle by a modification of the compatibility equation given in equation (52). As a constraint is released, the interface forces go to zero and the interface displacement, velocity, and acceleration become unequal. Thus, during the integration of the equations of motion, the inverted interface compatibility matrix $[\overline{C}]$ can be changed by zeroing out the row and column of the released interface DOF. This approach will be used to simulate the liftoff transient analysis of the space shuttle in section IV.

III. COMPUTATIONAL PROCEDURE

In this section, computational procedures are developed which make it more convenient for implementing the reduction transformations of the two CMS methods presented in section II. A computational procedure for the proposed transient response method with changing boundary conditions is also developed. Part of the procedure for the proposed method has been presented in reference 23.

A. Computational Procedure for the CMS Methods

First, the models of the substructures must be reduced in size for computational purposes. This is accomplished by using one of the CMS methods described in the previous section. To better understand the reduction procedure, a series of steps are listed for both of the CMS methods. Steps 1 through 3 are identical for both CMS methods. The steps are:

Step 1 – Partition the mass and stiffness matrices into interior and boundary (interface) coordinates.

Step 2 – Compute the $[G]$ Guyan transformation from equation (4).

Step 3 – Form CMS transformation matrix of equation (4).

For the Craig and Bampton CMS method, the following steps are followed:

Step 4 – Compute the normal modes for the fixed interior coordinates of the substructures and normalize the modes with respect to the interior mass matrix.

Step 5 – Truncate the eigenvectors to a specified cutoff frequency that gives satisfactory results to the transient response.

Step 6 – Form transformation $[T_{CB}]$ of equation (5) using truncated eigenvectors.

Step 7 – Compute reduced mass and stiffness matrices using $[T_{CB}]$ from step 6, i.e., perform matrix triple product as in equation (6).

For the Lanczos CMS method the following steps are followed:

Step 4 – Compute n Lanczos vectors of the interior coordinates of the substructures using equations (7) through (10).

Step 5 – Form transformation of equation (11) using the Lanczos vectors.

Step 6 – Compute reduced mass and stiffness matrices using $[T_L]$ from step 5, i.e., perform matrix triple product as in equation (11).

B. Computational Procedure for the Proposed Transient Response Method

The proposed transient response method computational procedure is presented next. This is given also as a list of steps to help in its implementation:

Step 1 – Compute the normal modes and frequencies of the reduced substructure from either step 7 or step 6 depending on which CMS method has been used.

Step 2 – Select an integration time step Δt that is consistent with the highest substructure normal frequency.

Step 3 – Compute the interface compatibility matrix $[\overline{C}]$, as defined in equation (51).

Step 4 – Compute the inverse of the interface compatibility matrix $[\overline{C}]^{-1}$.

Step 5 – Set t_i = integration start time, with $i = 1$.

Step 6 – Compute initial conditions at integration start time given in equations (40) and (41).

Step 7 – Set $t_{i+1} = t_i + \Delta t$.

Step 8 – Compute the response of substructures due to applied loads at $t = t_{i+1}$ by solving equations (39a) and (39c).

Step 9 – Compute difference of interface displacements, velocities, and accelerations due to applied forces at $t = t_{i+1}$ using equation (48).

Step 10 – Compute the coefficients G_1, G_2, and G_3 at $t = t_{i+1}$ using equation (52).

Step 11 – Compute the response of the substructures due to the interface forces at $t = t_{i+1}$ using equations (43) and (44).

Step 12 – Compute the total response of the substructures at $t = t_{i+1}$ using equations (37) and (38).

Step 13 – If a change in constraint or compatibility is necessary, modify the inverted interface compatibility matrix.

Step 14 – Compute the interface forces using equation (34) at $t = t_{i+1}$.

Step 15 – Reset the initial conditions for the next time step using equations (40) and (41).

Step 16 – Set $t_i = t_{i+1}$ and return to step 7 if t_i equals the end of integration time, then stop.

All of the normal modes and frequencies of each substructure done in step 1 of the proposed transient response method should be computed if computer capacity is available. This will result in more accurate internal loads on the substructures. The loss of accuracy will then be limited to only the reduced CMS method used.

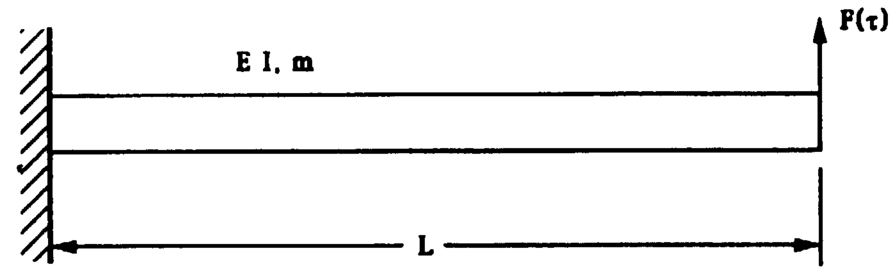
The computer algorithms for the two CMS methods are listed in the appendix. The proposed transient response method with changing boundary conditions is also listed in the appendix. All the computer routines have been written in FORTRAN computer code. The FORTRAN library called FORtran Matrix Analysis (FORMA) [27] is used throughout all of the algorithms developed.

IV. NUMERICAL EXAMPLES

In this section, two numerical examples are presented. One example uses a simple cantilevered beam with an applied force at the free end of the beam. The other example uses transient response analysis of the space shuttle liftoff from the MLP. The simple beam example is used for the purpose of checking out the computer algorithms for the CMS methods and the proposed transient response method. The liftoff transient response analysis includes the effects of changing boundary conditions as the vehicle goes from a fixed-base configuration to a free-flight configuration. All computations have been performed on a Cray XMP computer.

A. Simple Beam

A simple cantilevered beam is selected for the check out of the proposed transient response method and algorithms. The beam is also used to study and check out the Lanczos CMS method and algorithms. The cantilevered beam with an applied tip force is shown in figure 5, along with its material and geometric properties. The beam was modeled using the finite element method. It was first broken up into two substructures (one free-free and the other cantilevered) of equal length. Finite element mass and stiffness matrices are then generated for the two substructures. A total of 50 DOF for the free-free substructure and 48 DOF for the cantilevered substructure are used. The finite element representation of the beam is shown in figure 6a. Finite element mass and stiffness matrices are given in figure 6b. Reduction of the substructures is performed using the CMS transformations of sections A and B of section II. The substructure interface DOF are kept in discrete coordinates.



E I = 52080 lbs-in.^{2}
m = 2.4055E-05 lbs-sec.^{2}/in.^{2}
L = 48.0 in.

$$F(\tau) = \begin{cases} 1.0 \sin(\pi \tau / 0.2) & 0. \le \tau \le 0.2 \\ 0.0 & \tau > 0.2 \end{cases}$$

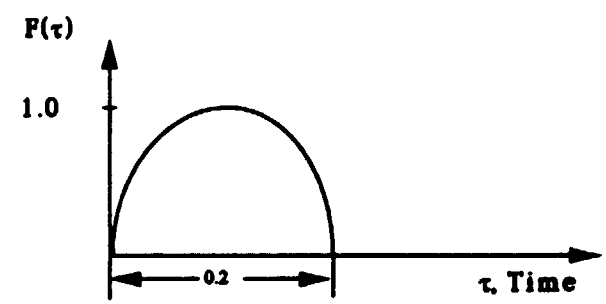


Figure 5. Cantilevered beam with applied load $F(t)$.

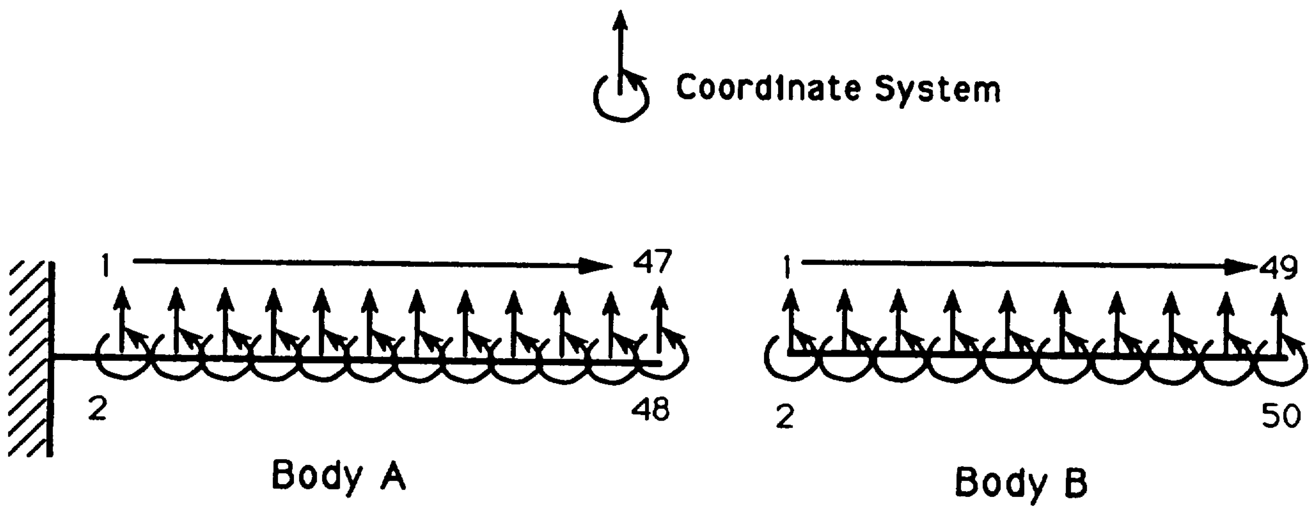


Figure 6a. Two-dimensional finite element model of cantilevered beam.

$$K_e = \frac{EI}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \qquad M_e = \frac{mL}{420}\begin{bmatrix} 156 & 22L & 54 & -13L \\ 22L & 4L^2 & 13L & -3L^2 \\ 54 & 13L & 156 & -22L \\ -13L & -3L^2 & -22L & 4L^2 \end{bmatrix}$$

Figure 6b. Element mass and stiffness for the cantilevered beam.

For the proposed transient response method, the accuracy of eigenvalues used to represent the beam is important for satisfactory results. Therefore, the discrete beam model eigenvalues (frequencies) are compared to the reduced beam models for the two CMS methods. Frequency versus mode number comparisons are shown in figures 7 through 10 for several reduced models of the free-free and cantilevered beams. From figures 7 through 10 it is seen that the Craig and Bampton reduced models more accurately represented the original discrete model. The Lanczos reduced models consistently lost accuracy depending on the number of vectors retained in the reduction transformation. For the Lanczos reduced models to achieve the same accuracy in frequency as the Craig and Bampton reduced models, more Lanczos vectors need to be generated for the reduction transformation.

To check out the algorithms of the proposed method, a transient response analysis of the beam substructure models is performed. The transient response uses both the reduced Craig and Bampton CMS models and Lanczos CMS models. A closed-form solution of the discrete finite element model is computed for comparison. The finite element discrete model consists of the mass and stiffness matrices of the two substructures coupled together using the direct stiffness method. A eigen analysis of the discrete beam model is performed. Only modes up to 100 Hz are retained and used in the closed-form transient response analysis. The length of the transient response analysis is set to a time interval from 0.0 to 2.0 s and an integration time step of 0.001 s is used. Damping is neglected for this study. The displacement at the end of the beam (tip of the beam) and the beam's interface moment (between substructures) are computed for comparison studies. The tip displacement is plotted versus time in figure 11 and the interface moment is plotted versus time in figure 12. From figures 11 and 12 it can be seen that there is very little difference between the Craig and Bampton CMS and Lanczos CMS models. The differences that are present can be attributed to the time step used in the numerical integration. One characteristic of the Lanczos CMS reduced model is a higher frequency for the last mode. Thus, for good results a smaller time step is needed in the integration process. The Craig and Bampton method has the advantage of picking a cutoff frequency and thus limiting the size of the time step of integration. The analyst must choose the number of vectors to be generated for the Lanczos CMS method without prior knowledge of what the last modal frequency will be.

B. Transient Response of Shuttle Liftoff

A study of the proposed method using Lanczos CMS models versus Craig and Bampton CMS models has been performed. Computer usage is studied for the CMS reduction methods and for the proposed transient response method. Described in the following sections are the models, forcing functions, and results of the simulation. The results are compared with the iterative transient response method used by Martin Marietta.

1. Models

Free-free models of the substructures that make up the total liftoff space shuttle vehicle are obtained from Rockwell International [24]. The models were in a mixed discrete and modal form. The total liftoff vehicle model includes the orbiter (excluding a payload), ET, and two SRB's. The individual substructure models are coupled together using the direct stiffness method. The size of

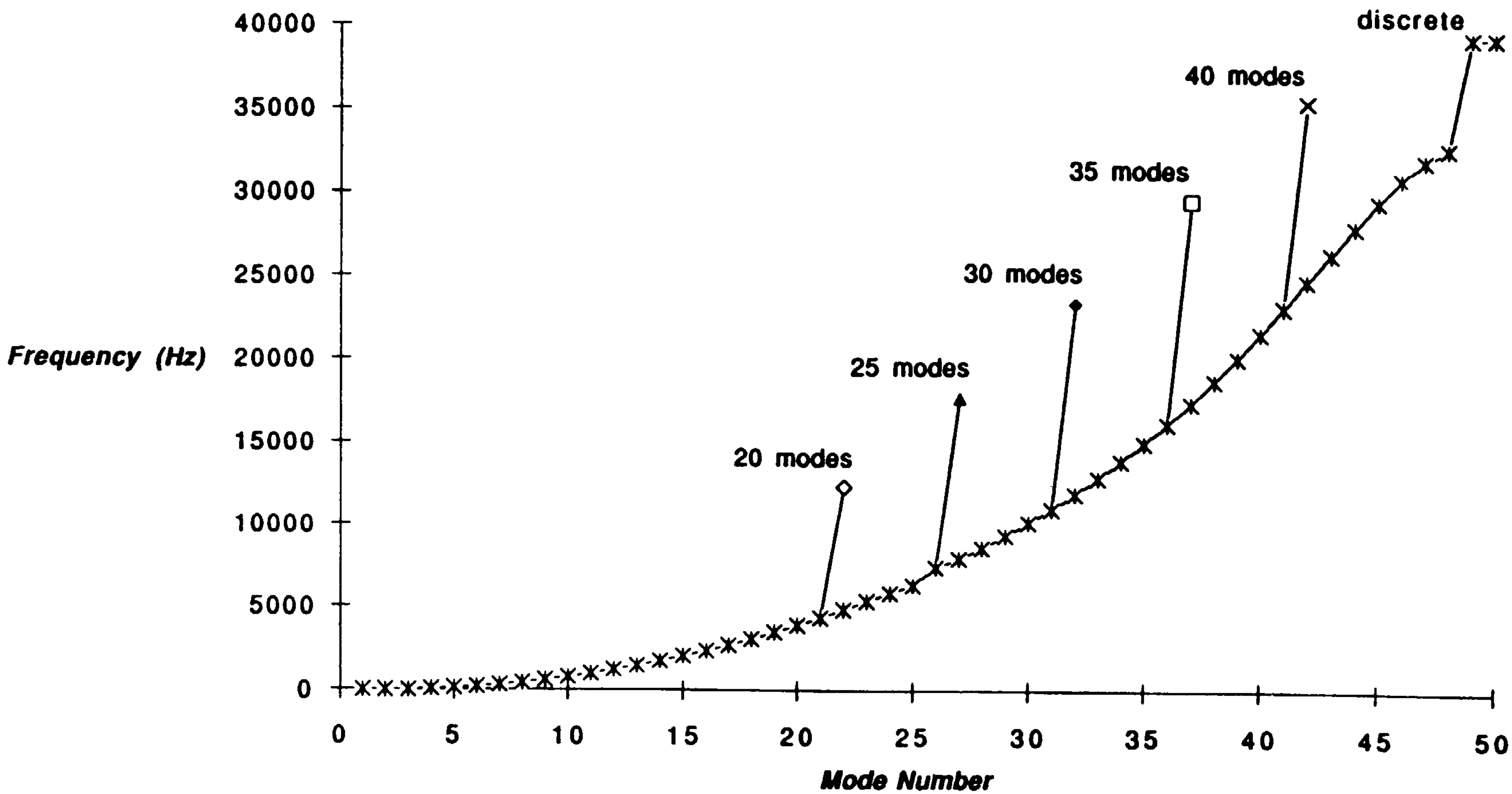


Figure 7. Frequency versus mode number for free-free beam Craig-Bampton CMS model comparisons.

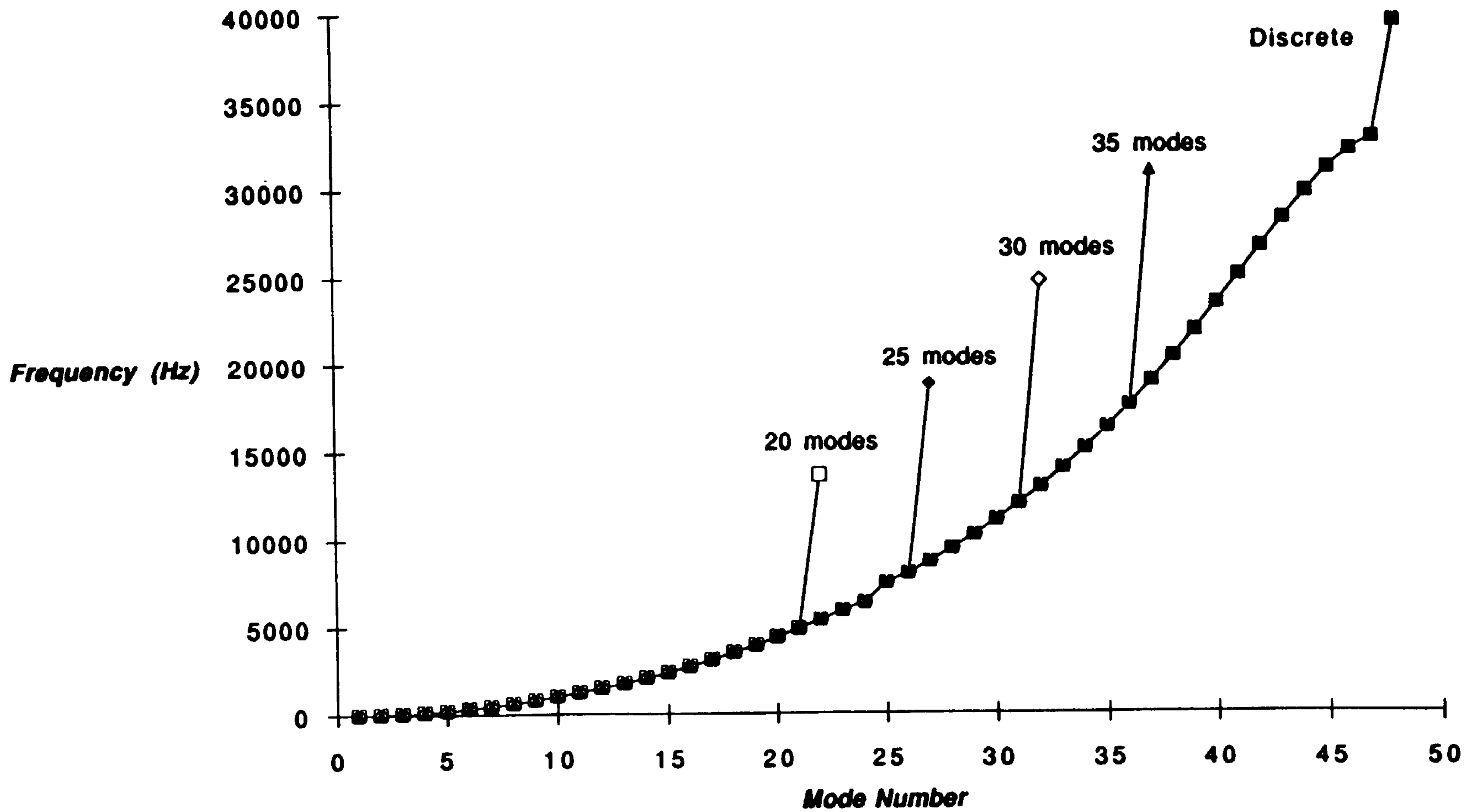


Figure 8. Frequency versus mode number for cantilevered beam Craig-Bampton CMS model comparisons.

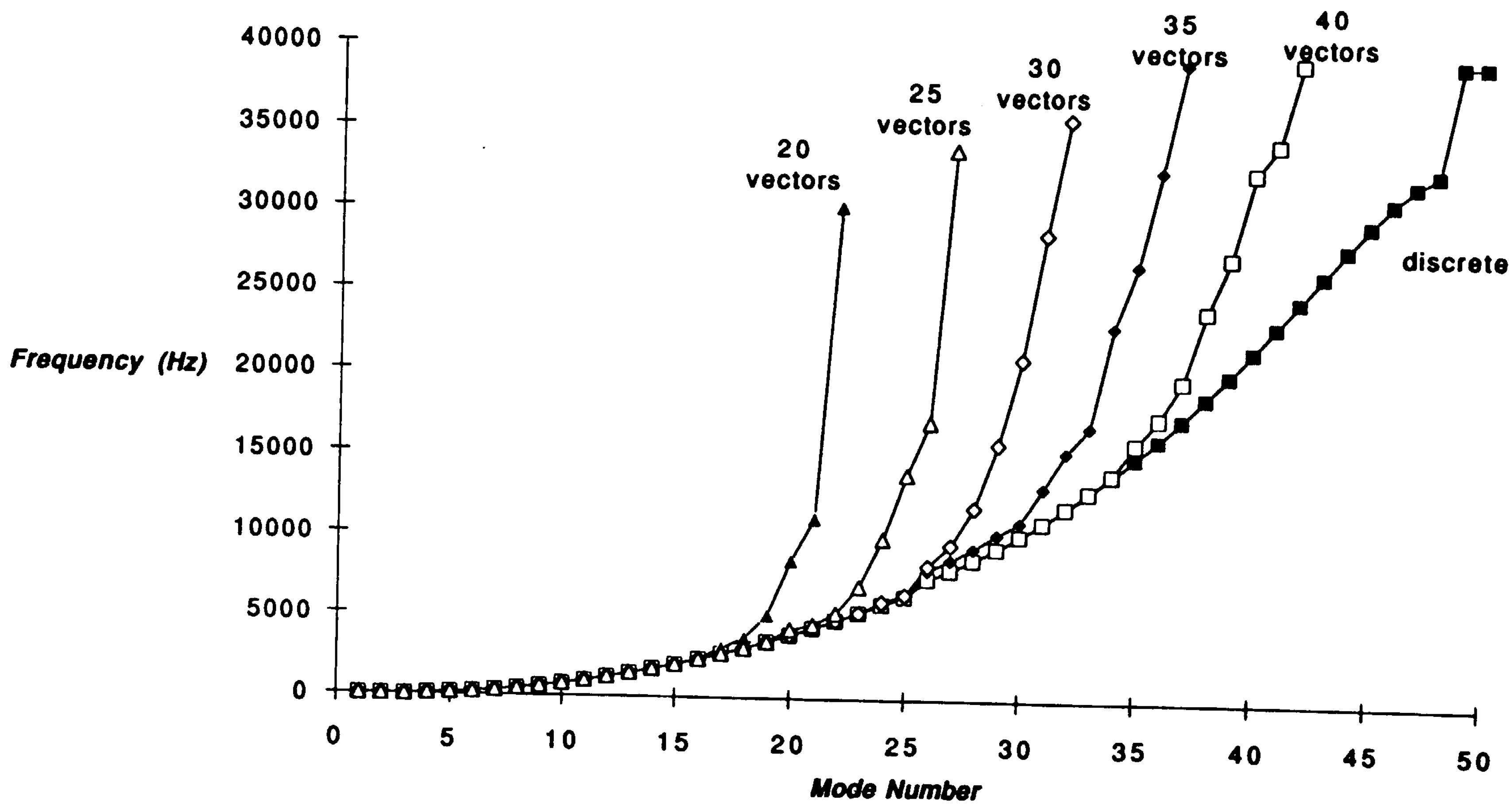


Figure 9. Frequency versus mode number for free-free beam Lanczos CMS model comparisons.

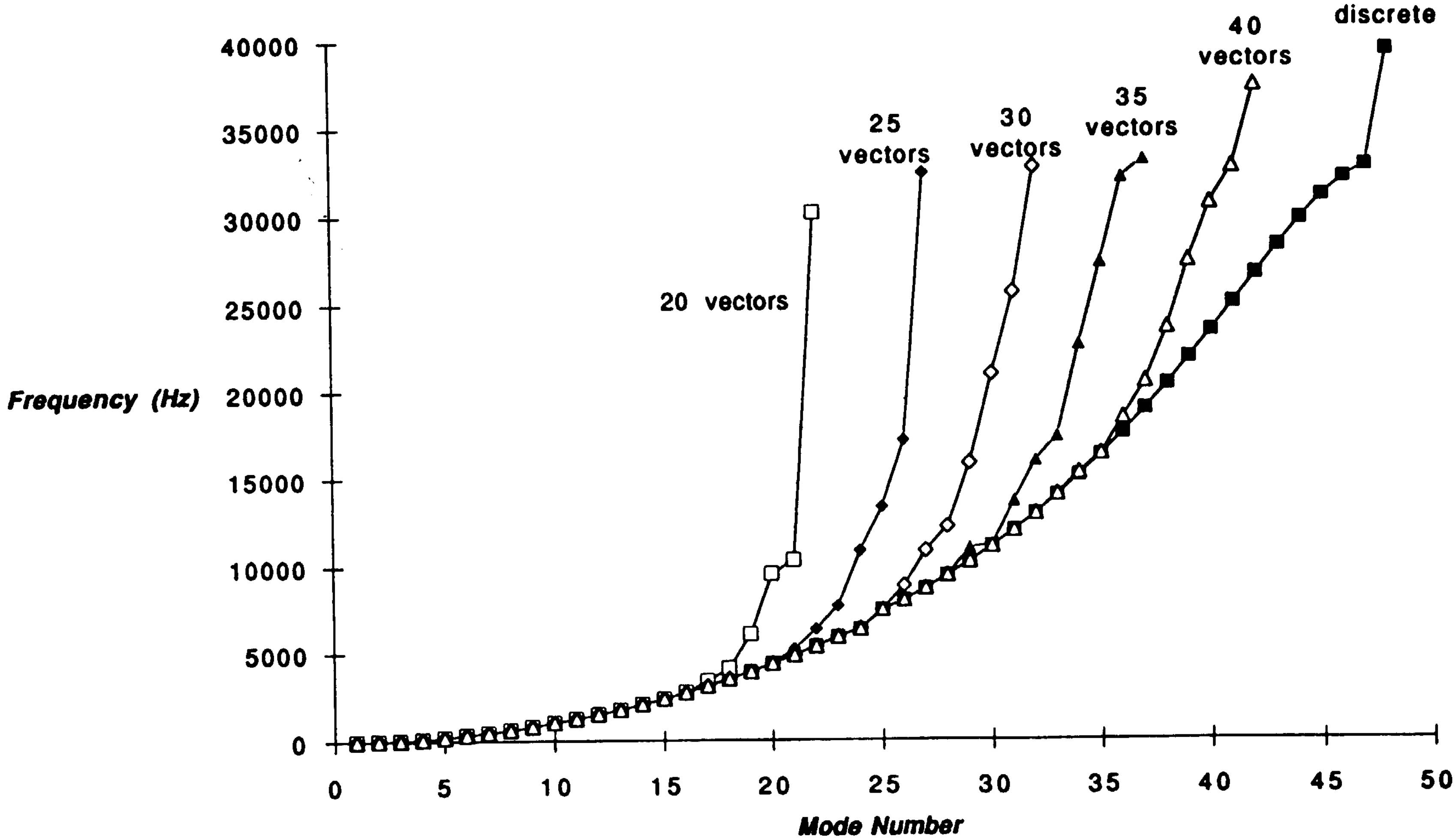


Figure 10. Frequency versus mode number of a cantilevered beam Lanczos CMS model comparisons.

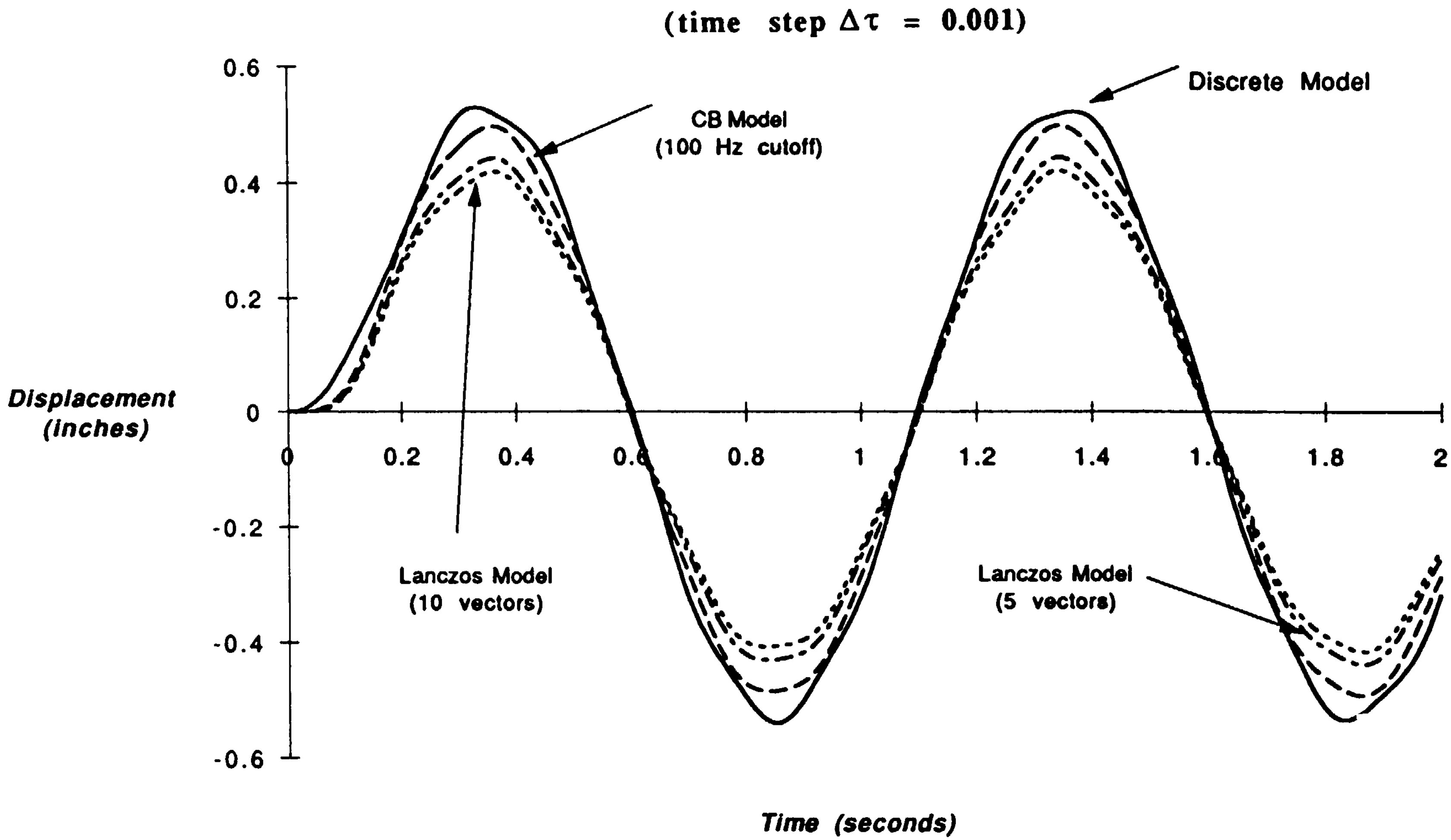


Figure 11. Proposed transient response method tip displacement versus time
(Craig and Bampton versus Lanczos).

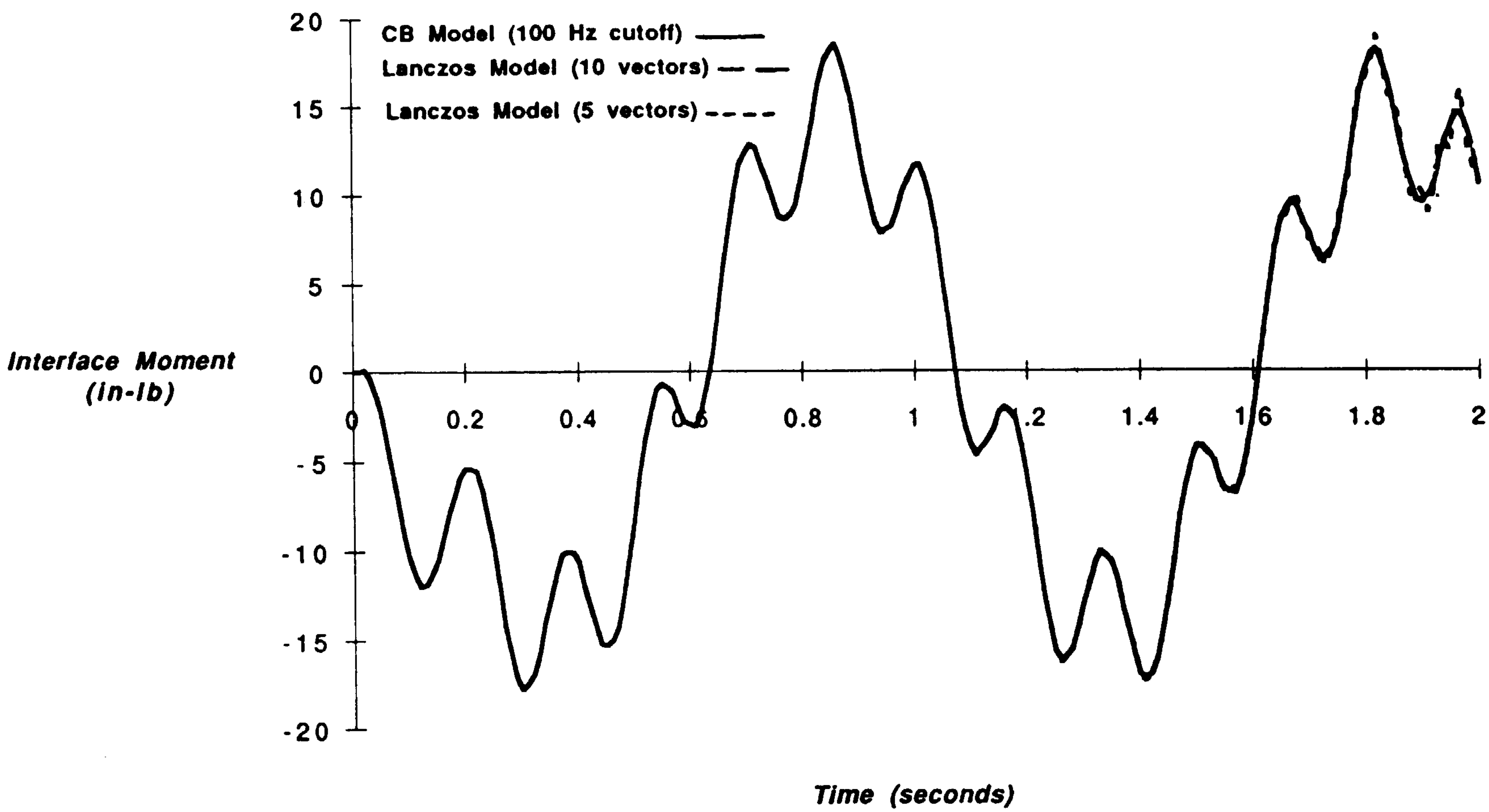


Figure 12. Proposed transient response method interface moment versus time (Craig and Bampton versus Lanczos).

the total liftoff model has 1,725 DOF. An MLP model is obtained from NASA Johnson Space Center [25]. It has 109 DOF and is in discrete coordinates. The MLP is reduced using the Craig and Bampton CMS method using a cutoff frequency of 100 Hz. The same MLP model is used for all the simulations. Liftoff vehicle models are reduced using both the Craig and Bampton and Lanczos CMS methods. Four Craig and Bampton liftoff vehicle models and three Lanczos liftoff vehicle models have been studied. Table 1 gives the sizes of the reduced models. A comparison between the computed eigenvalues of the reduced Lanczos CMS vehicle models and the unreduced vehicle model has been performed and is shown in figure 13. The results are similar to what was observed previously in the simple beam model. The more vectors retained in the Lanczos transformation the more accurate the eigenvalues of the reduced model.

Table 1. Sizes of reduced vehicle models.

Craig and Bampton		**Lanczos**	
Cutoff Frequency	Size	Vectors Retained	Size
70 Hz	664 dof	300 vectors	324 dof
50 Hz	495 dof	100 vectors	124 dof
35 Hz	365 dof	10 vectors	34 dof
20 Hz	206 dof		

Following the steps outlined in section III, an eigen analysis of the reduced models is accomplished next. All eigenvalues and eigenvectors are kept for the reduced models in table 1. Damping is neglected in the transient response analysis for both the vehicle and MLP models. A time step of 0.001 s is used in all the transient response analyses using the proposed method. For the iterative method a time step of 0.0001 s is needed for convergence of solution.

2. Forcing Functions

Forces acting on the space shuttle vehicle during liftoff include gravity, wind loads, space shuttle main engine (SSME) thrust forces, solid rocket motor (SRM) ignition overpressure loads, SRM thrust and pressure loads, and foot pad loads. There are over 300 sets of these forces which are developed by Rockwell International [26]. The set of forces used in the transient response example is designated LR2019. These forces applied to a free-free vehicle model with the foot pad loads simulating the effect of the MLP during liftoff are normally used for payload liftoff loads response analysis. For this simulation, the pad loads are zeroed out of the forcing functions since they are part of the results. A total of 166 applied forces per 680 time points are used in the liftoff siimulation. The forcing functions are interpolated using the integration time step before they are applied in the integration of the equations of motion. The SRB's are ignited at t = 6.548 s, therefore, the transient response analysis of the reduced models is accomplished over the time interval of 0 to 7.0 s.

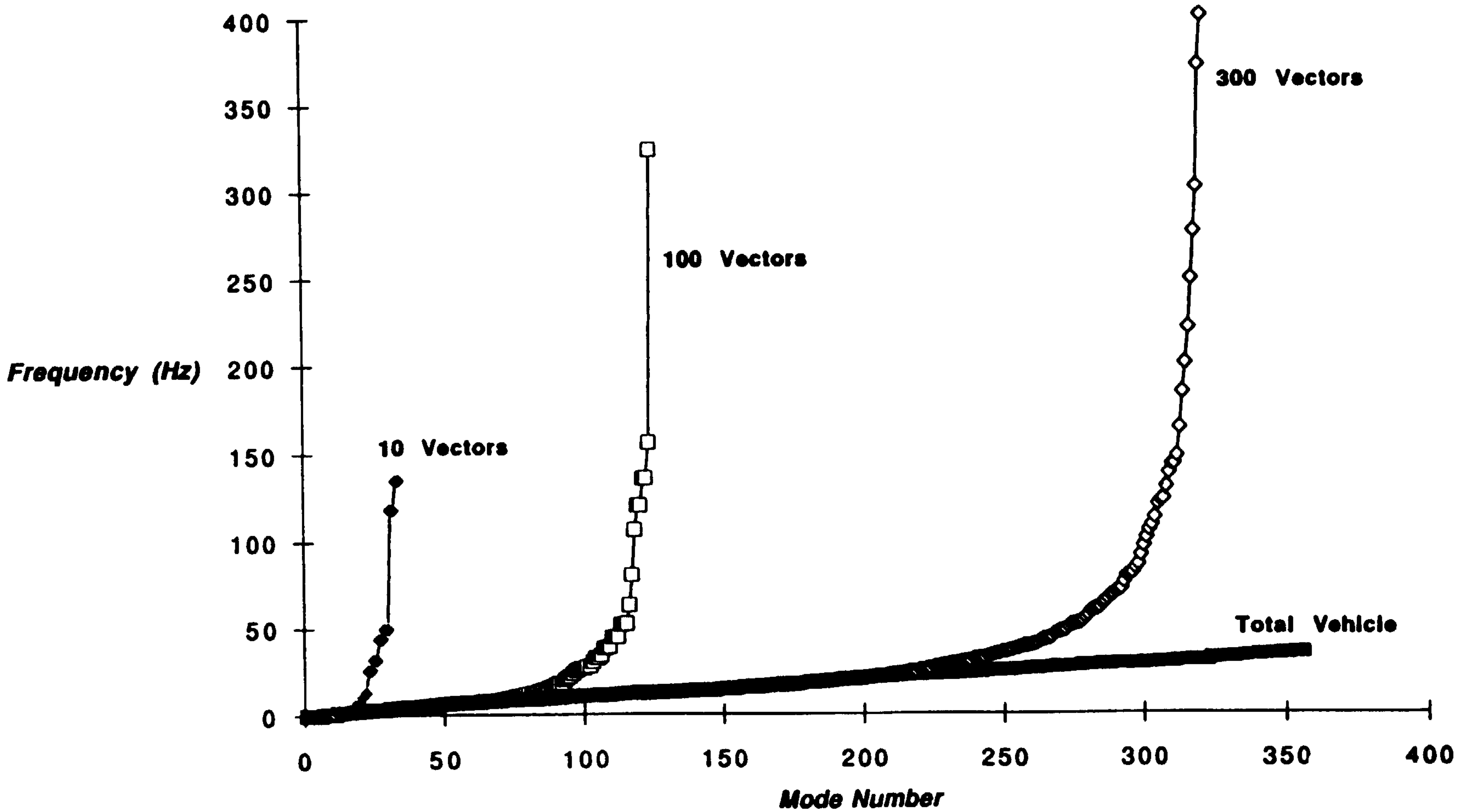


Figure 13. Frequency versus mode number of liftoff shuttle model
(Lanczos vectors used for reduction).

3. Results

MLP to vehicle interface forces and displacements are computed for the reduced models shown in table 1. To verify the proposed transient response method, the same reduced model (Craig-Bampton 70 Hz) is used for the iterative method and the proposed method. Figures 14 and 15 show the interface forces of posts 1 and 4 using both methods. It is seen that the proposed method compares very well with the iterative method. The differences in results that do appear in the two methods shown in figures 14 and 15 (especially in the *y* and *z* directions) can be attributed to the coupling stiffness matrix used to represent the holddown bolts and to the separation criterion in Martin Marietta's iterative method. Martin Marietta's separation criterion is to release the contact point as soon as the holddown bolt goes into tension. The holddown bolts are modeled using the coupling stiffness matrix, therefore, the bolt loads go into tension when the contact points separate. This accounts for post-4 interface loads computed by Martin Marietta's method going to zero before the proposed method as shown in figure 15.

A summary of the computer time needed for the reduction of models and the transient response methods is given in table 2. A substantial savings in computer time is shown for the Lanczos reduced models over that of the Craig and Bampton reduced models. It is observed that the iterative method used considerably more computer time than the proposed method for the same solution. This can be attributed to the smaller time step required and also due to the number of iterations needed for solution convergence.

Figures 16 and 17 show the interface forces for posts 1 and 4, using the proposed method. Different reduced Craig and Bampton CMS models are used for comparison studies. It is noted that good results can be obtained using the smaller Craig and Bampton CMS vehicle model (20-Hz cutoff frequency). A closer look at the interface loads during changes in the boundary conditions is shown in figures 18 and 19. Only for post 4 are there any large deviations in loads. These deviations can be attributed to the cutoff frequency used on the reduced models.

The reduced Lanczos vehicle models are compared against the Craig and Bampton (70-Hz cutoff) vehicle model. Interface forces for posts 1 and 4 are shown in figures 20 and 21. Good results are obtained of the interface forces for the Lanczos models. After SRB ignition, however, several deviations are noticed. Shown in figures 22 and 23 are the interface forces during SRB ignition and subsequent liftoff. The smaller Lanczos model (10 vectors) was not able to respond to the applied loads at post 4 as well as the higher fidelity models.

Displacements at the interface of post 1 are computed and are shown in figures 24 through 26 for the Craig and Bampton (70-Hz cutoff) model. The *x*-direction displacement (direction of flight) in figure 24 shows the MLP and vehicle connected right up to SRB ignition. After SRB ignition the two structures are separated. From the data it appears that the two structures reconnect a short time after SRB ignition. It is possible that this reconnection or chatter of the two structures is going on, however, the present routine does not deal with the reconnection event.

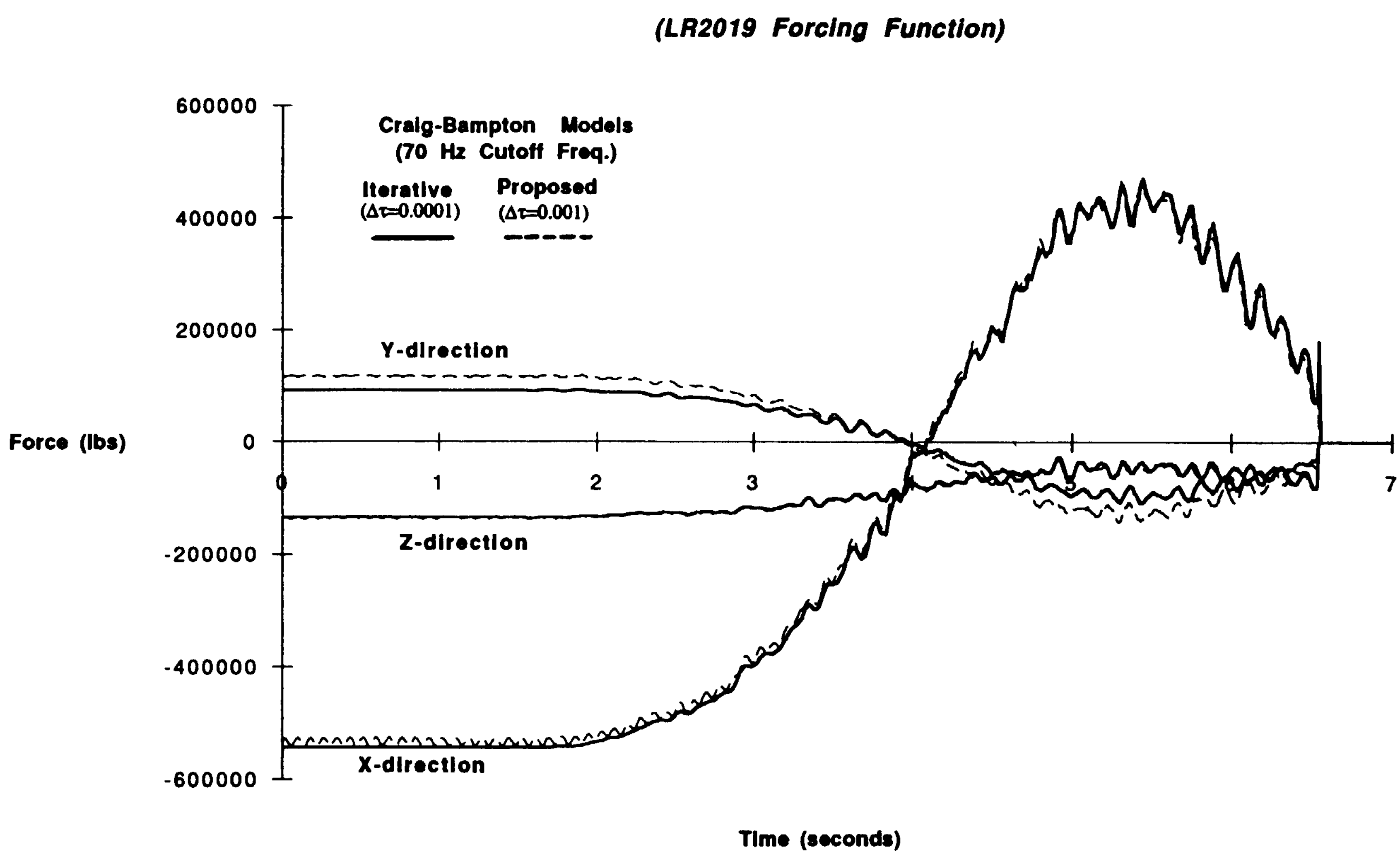


Figure 14. Proposed method versus iterative method post 1 interface forces (lb) versus time (seconds).

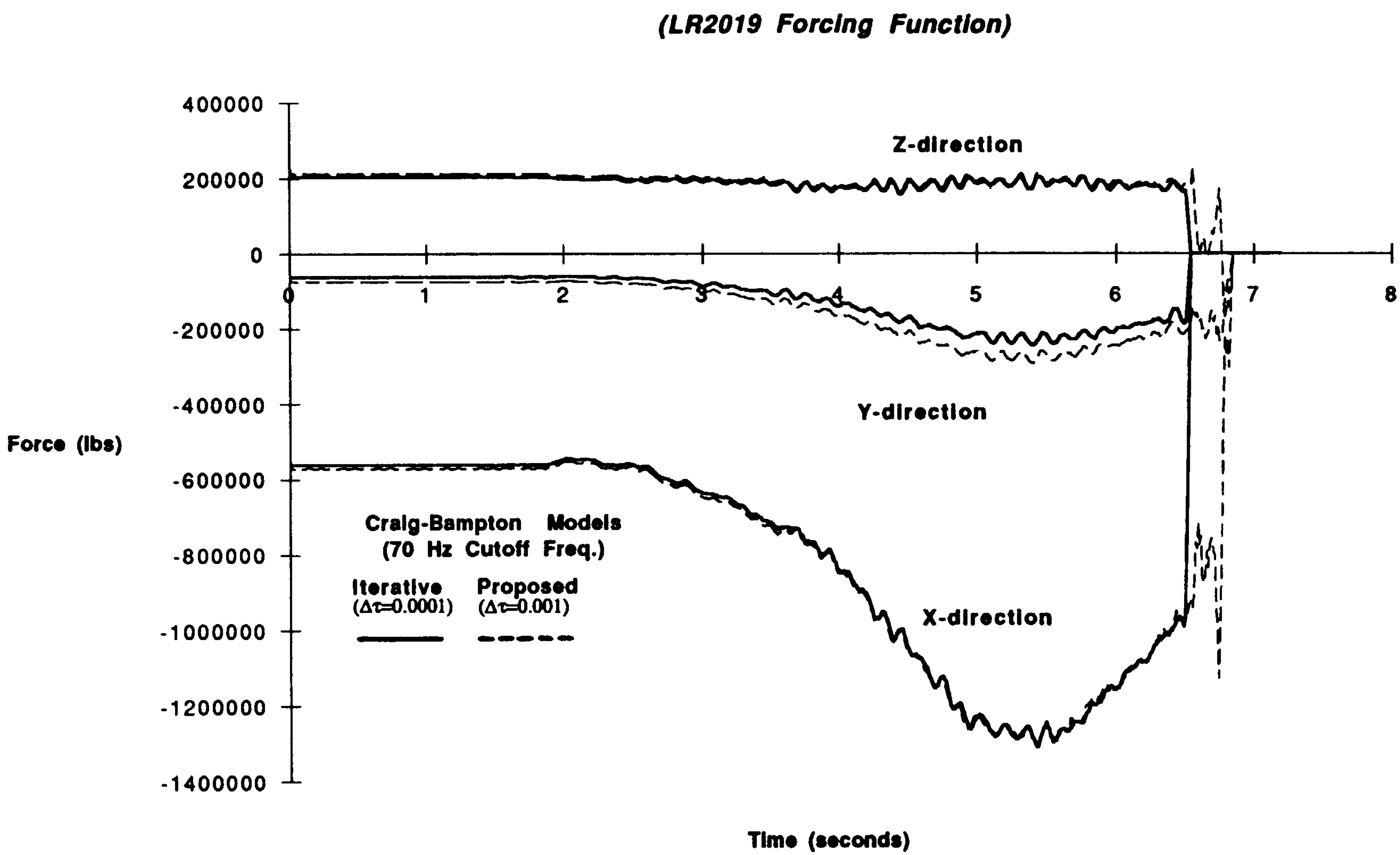


Figure 15. Proposed method versus iterative method post 4 interface forces (lb) versus time (seconds).

Table 2. Computer time comparisons of the CMS methods and the proposed transient response method.

Martin Marietta's Iterative Transient Response Method

			CPU Time		
Craig-Bampton Models	Cutoff Freq. Hz	Reduced Size	CB Reduction	System Modes	Iterative Response
MLP	100	79 x 79	3 sec.	-	-
Space Shuttle Liftoff	70	664 x 664	6183 sec.	200 sec.	1733 sec.

Proposed Transient Response Method

			CPU Time		
Craig-Bampton Models	Cutoff Freq. Hz	Reduced Size	CB Reduction	System Modes	Proposed Response
MLP	100	79 x 79	3 sec.	-	-
Space Shuttle Liftoff	70	664 x 664	6183 sec.	200 sec.	328 sec.
Space Shuttle Liftoff	50	495 x 495	4367 sec.	94 sec.	257 sec.
Space Shuttle Liftoff	35	365 x 365	2933 sec.	44 sec.	207 sec.
Space Shuttle Liftoff	20	206 x 206	2028 sec.	13 sec.	139 sec.

			CPU Time		
Lanczos Models	# Lanczos Vectors	Reduced Size	Lanczos Reduction	System Modes	Proposed Response
Space Shuttle Liftoff	10	34 x 34	355 sec.	3 sec.	67 sec.
Space Shuttle Liftoff	100	124 x 124	696 sec.	6 sec.	105 sec.
Space Shuttle Liftoff	300	324 x 324	2951 sec.	40 sec	198 sec.

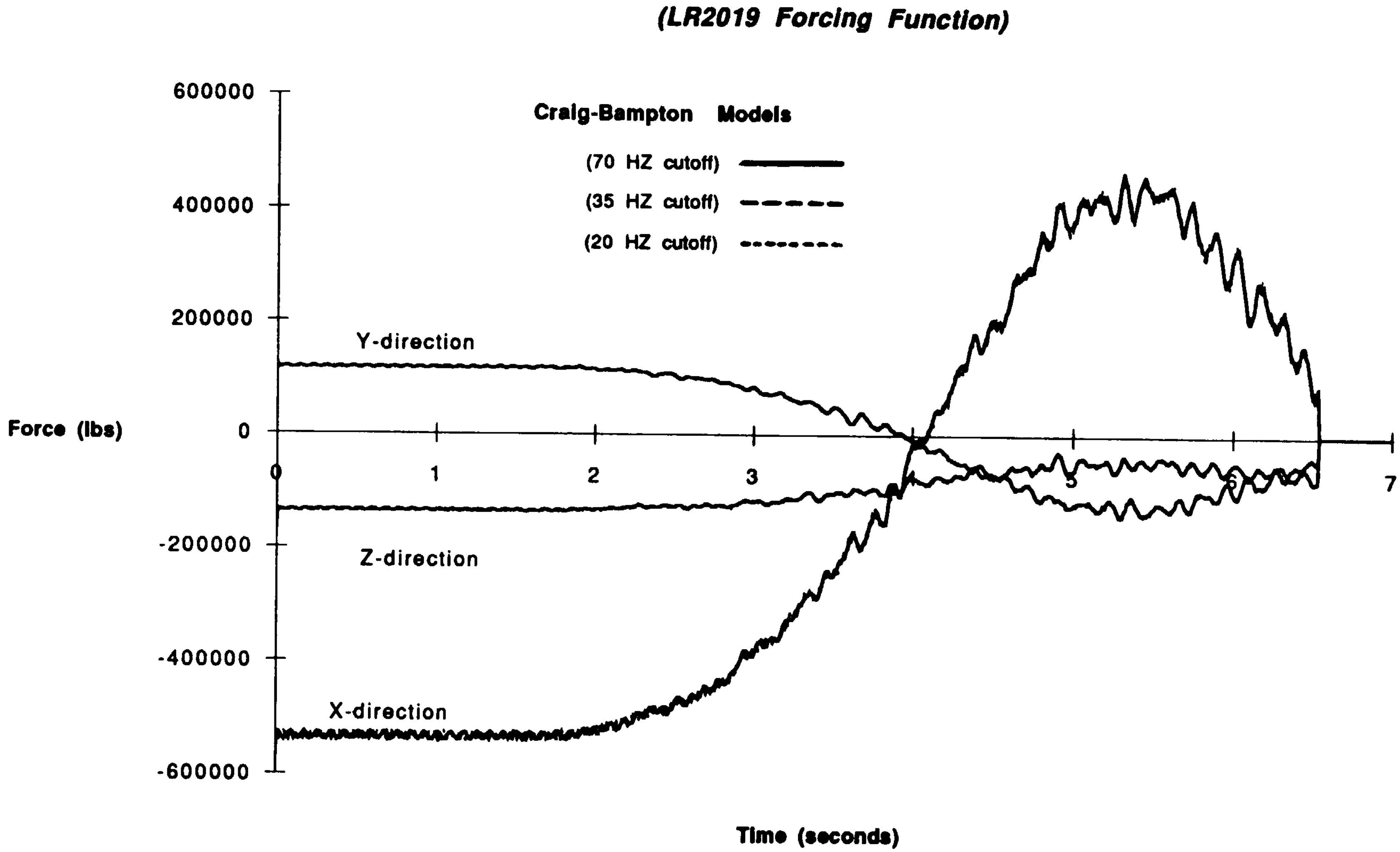


Figure 16. Proposed transient response method post 1 interface forces (lb) versus time (seconds).

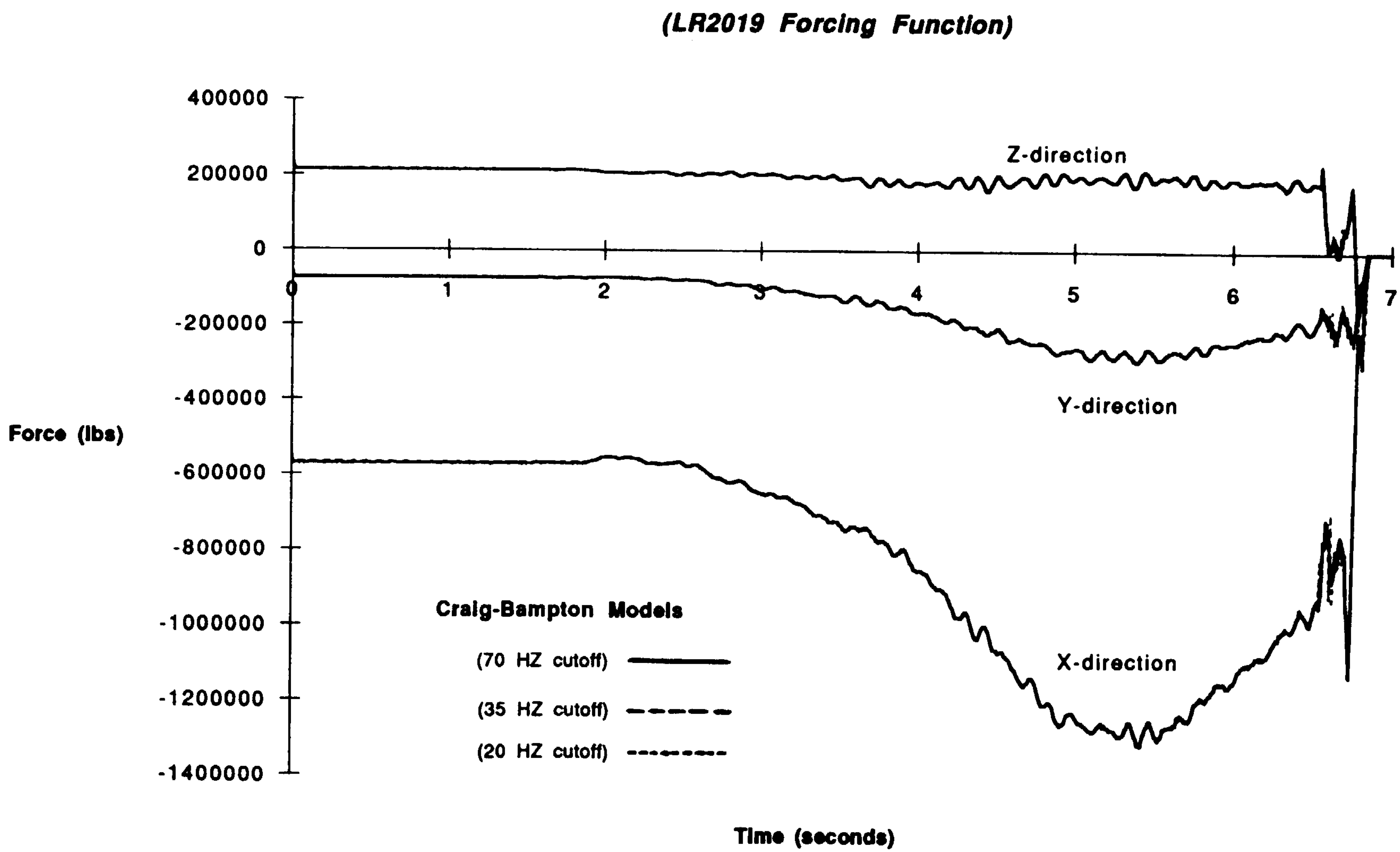


Figure 17. Proposed transient response method post 4 interface forces (lb) versus time (seconds).

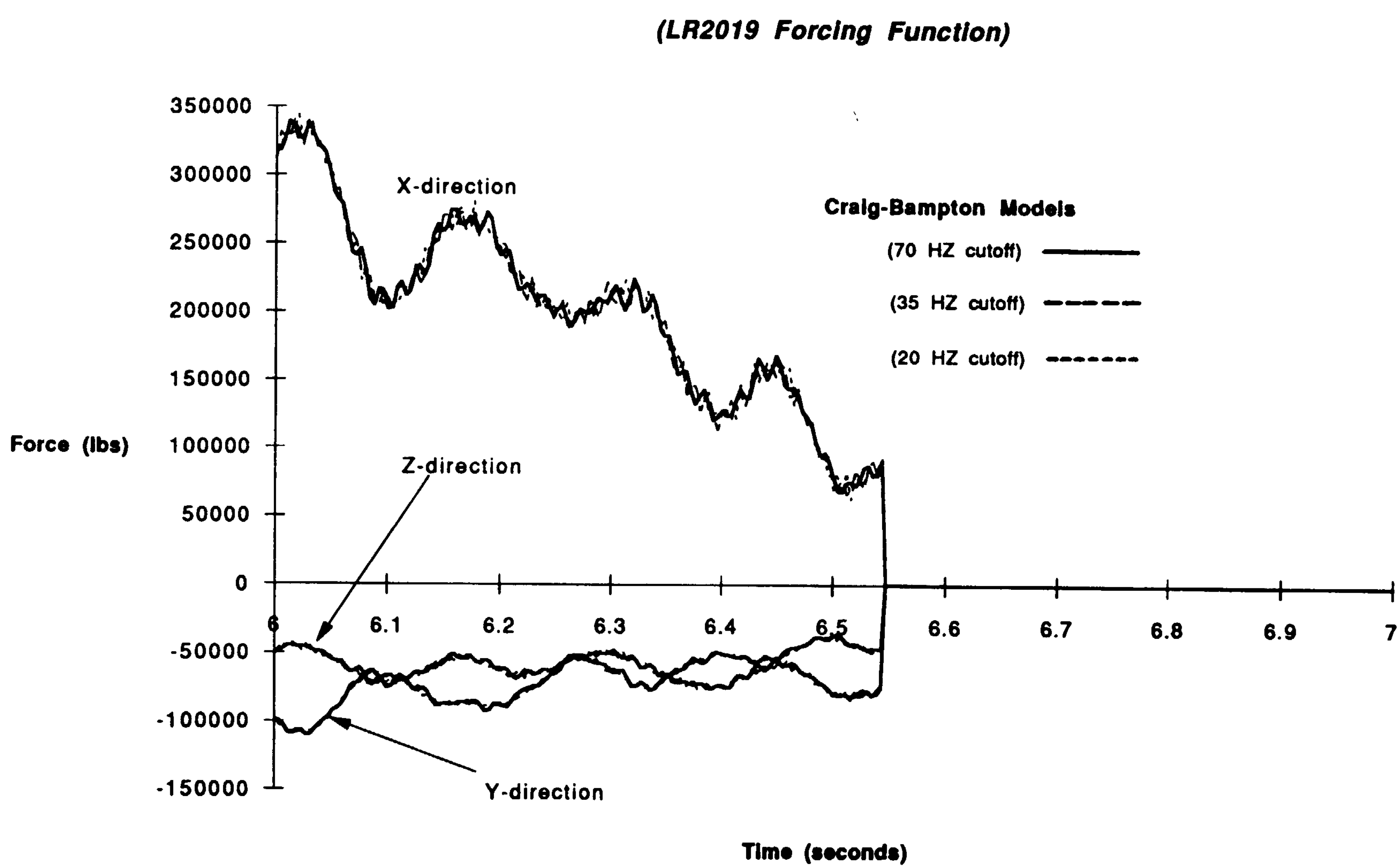


Figure 18. Proposed transient response method post 1 interface forces (lb) versus time (seconds).

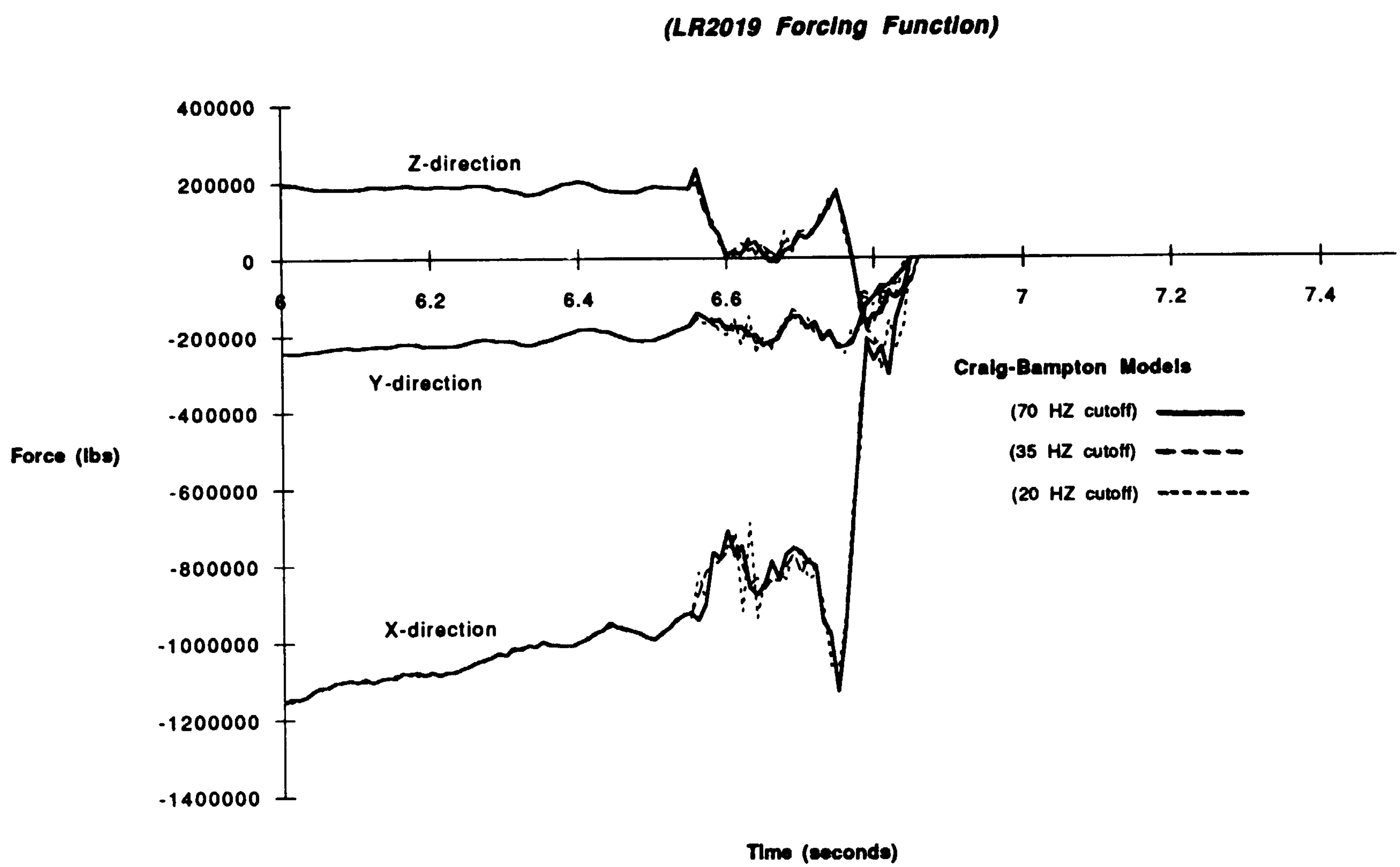


Figure 19. Proposed transient response method post 4 interface forces (lb) versus time (seconds).

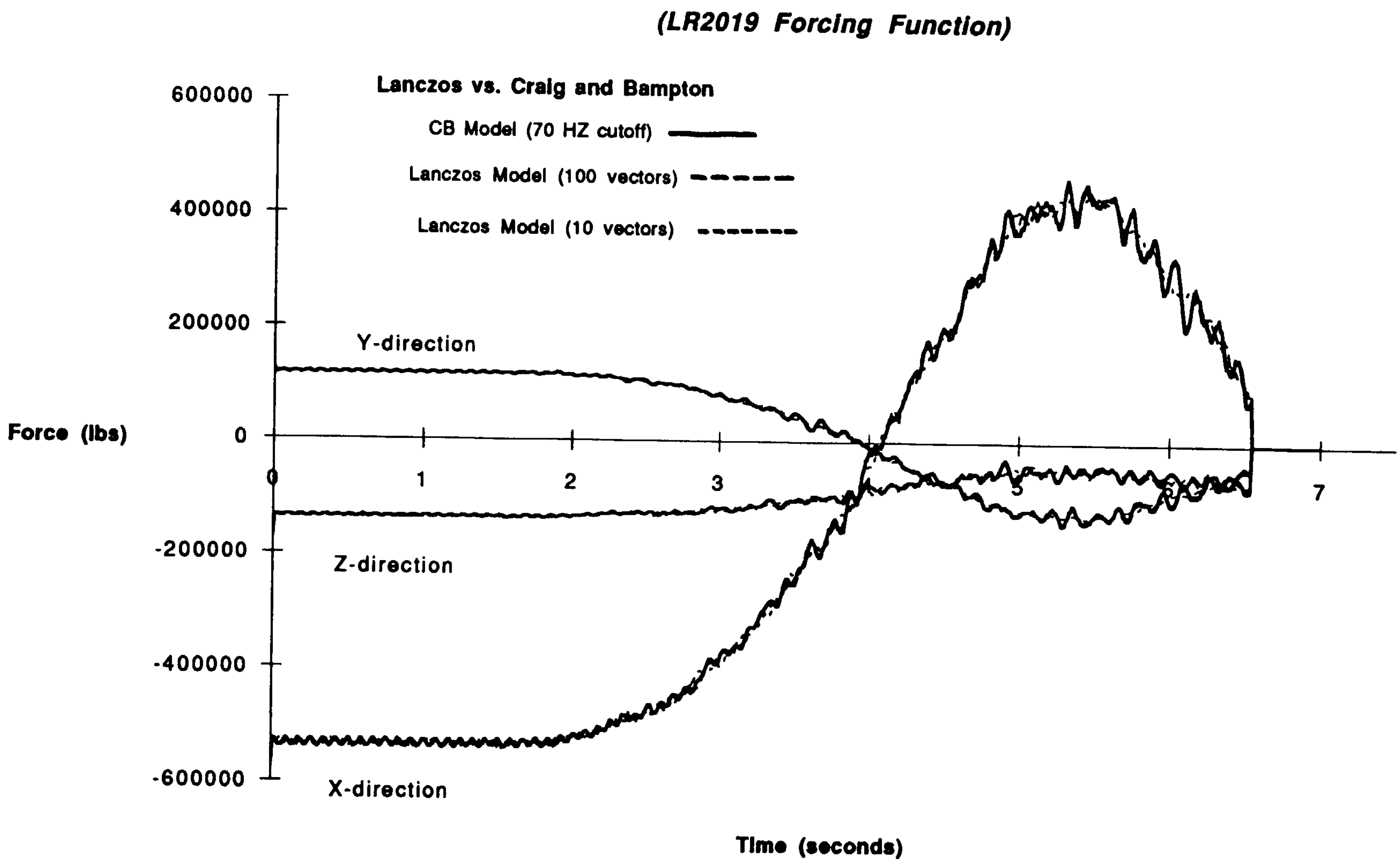


Figure 20. Proposed transient response method post 1 interface forces (lb) versus time (seconds).

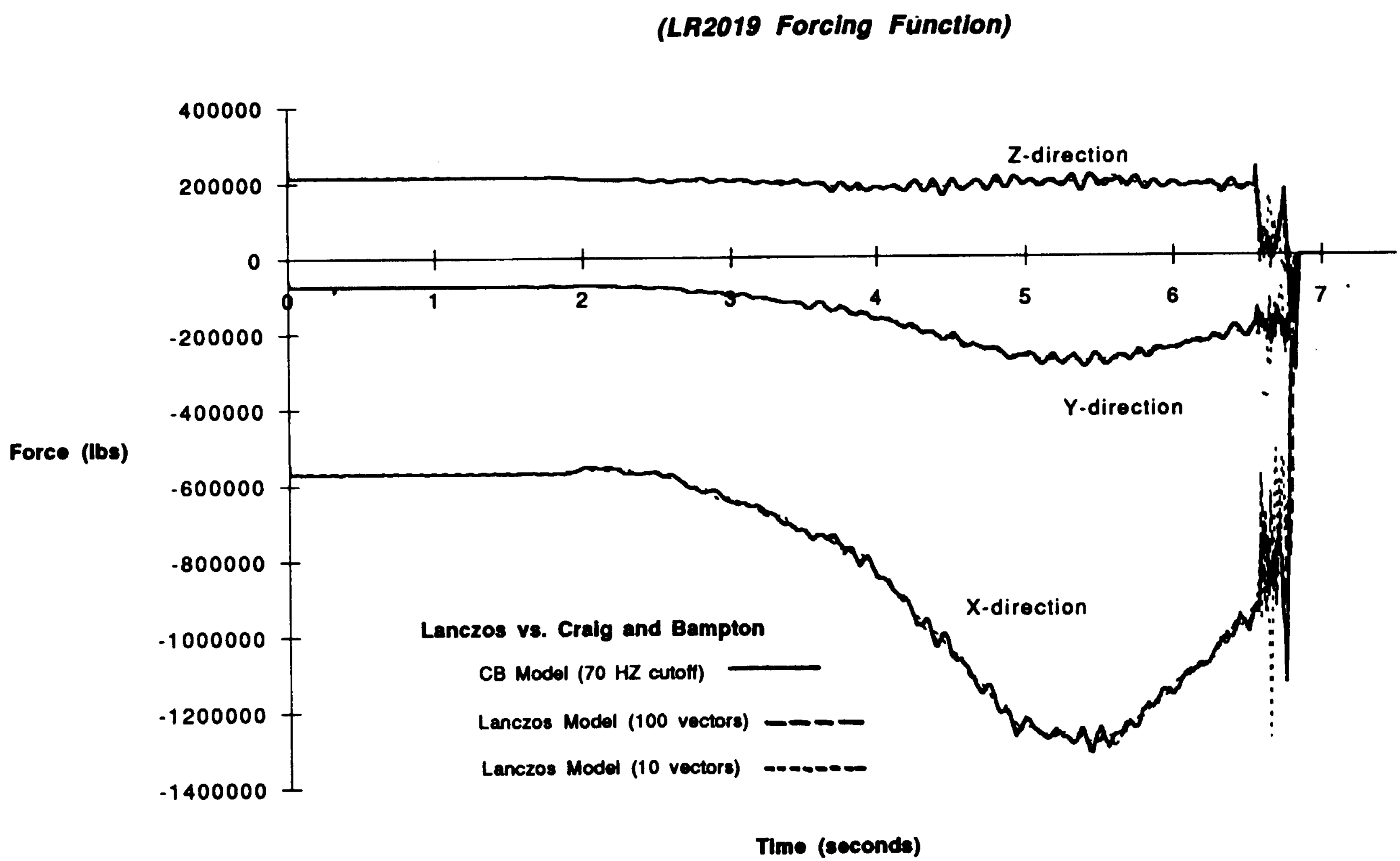


Figure 21. Proposed transient response method post 4 interface forces (lb) versus time (seconds).

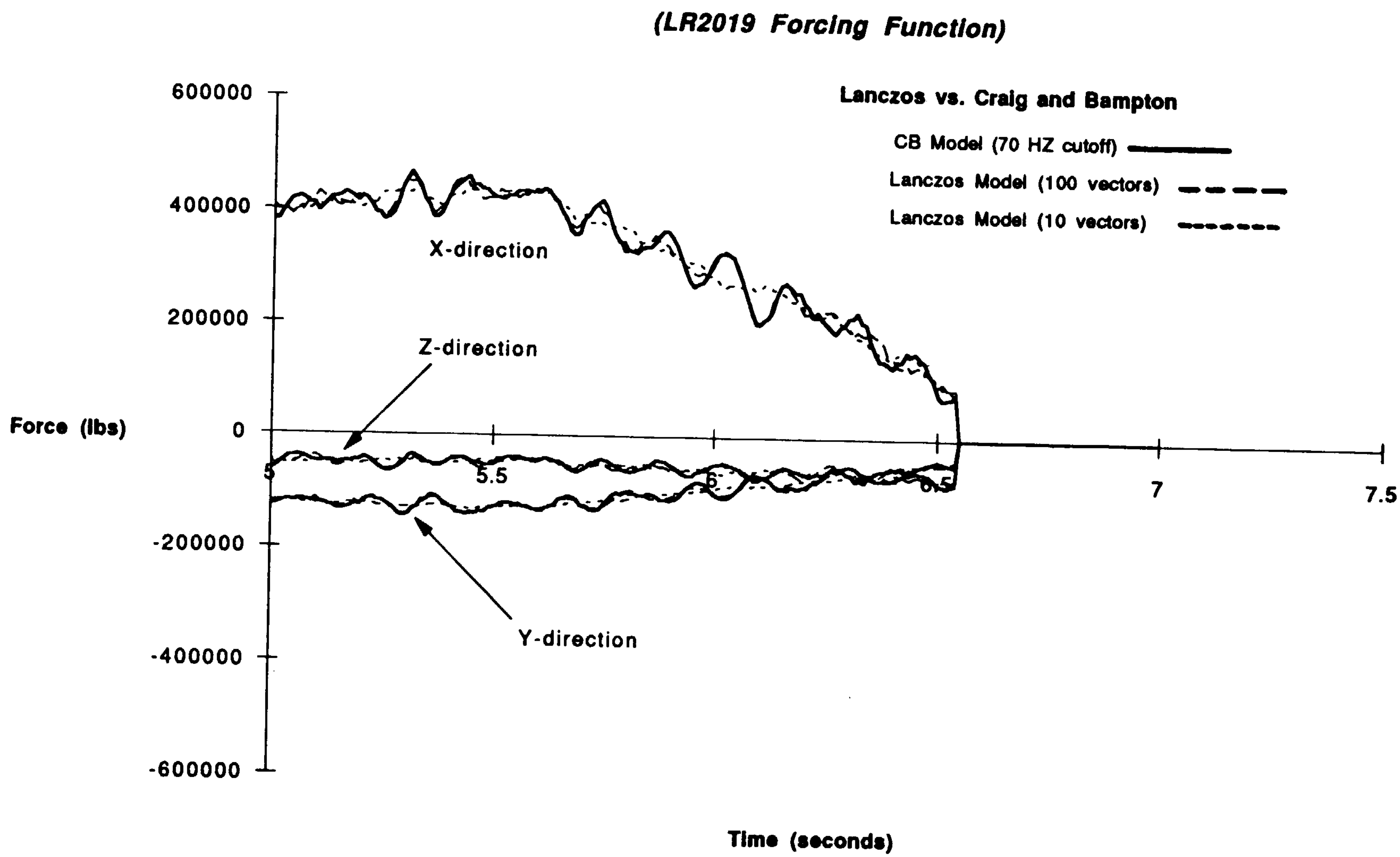


Figure 22. Proposed transient response method post 1 interface forces (lb) versus time (seconds).

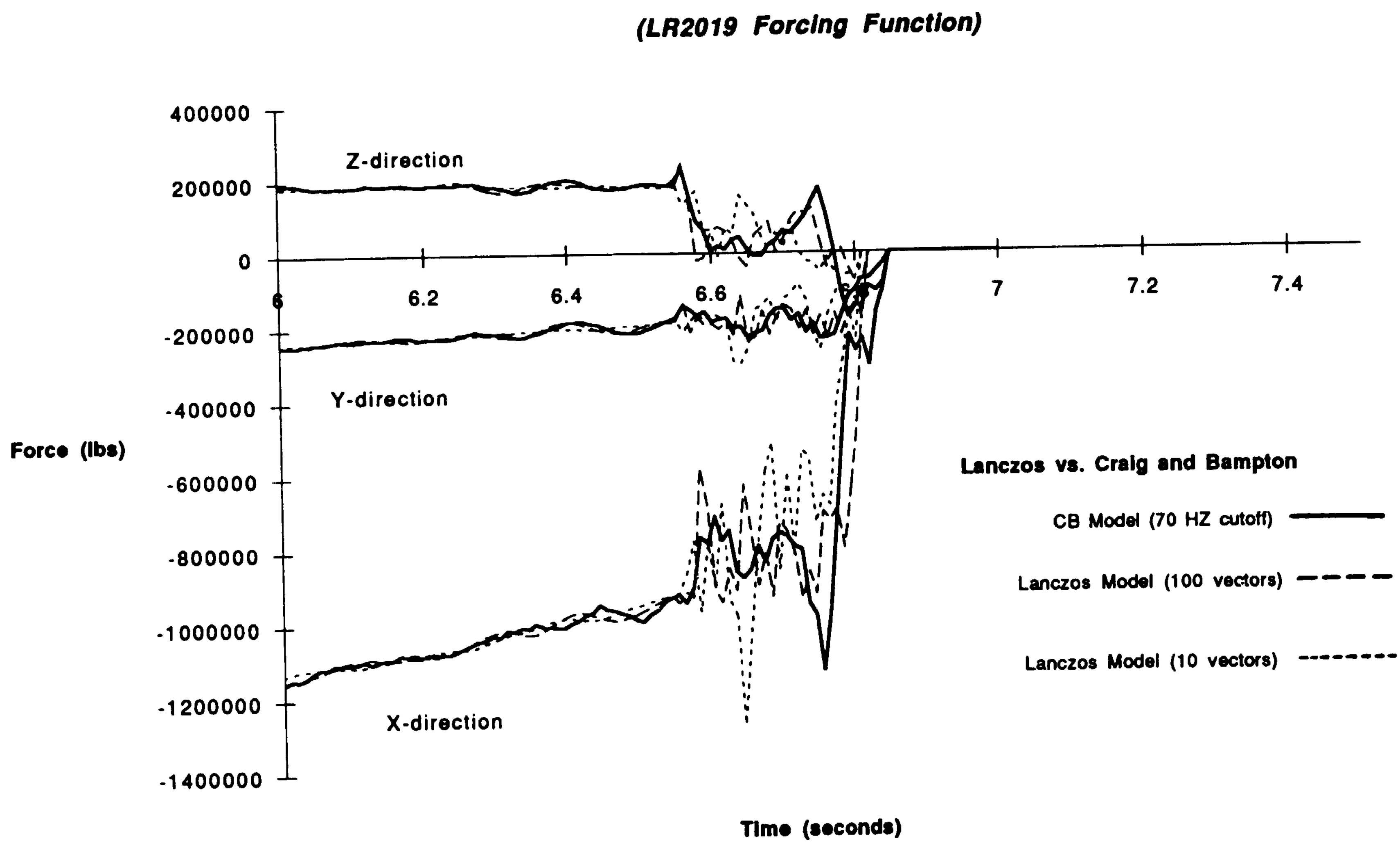


Figure 23. Proposed transient response method post 4 interface forces (lb) versus time (seconds).

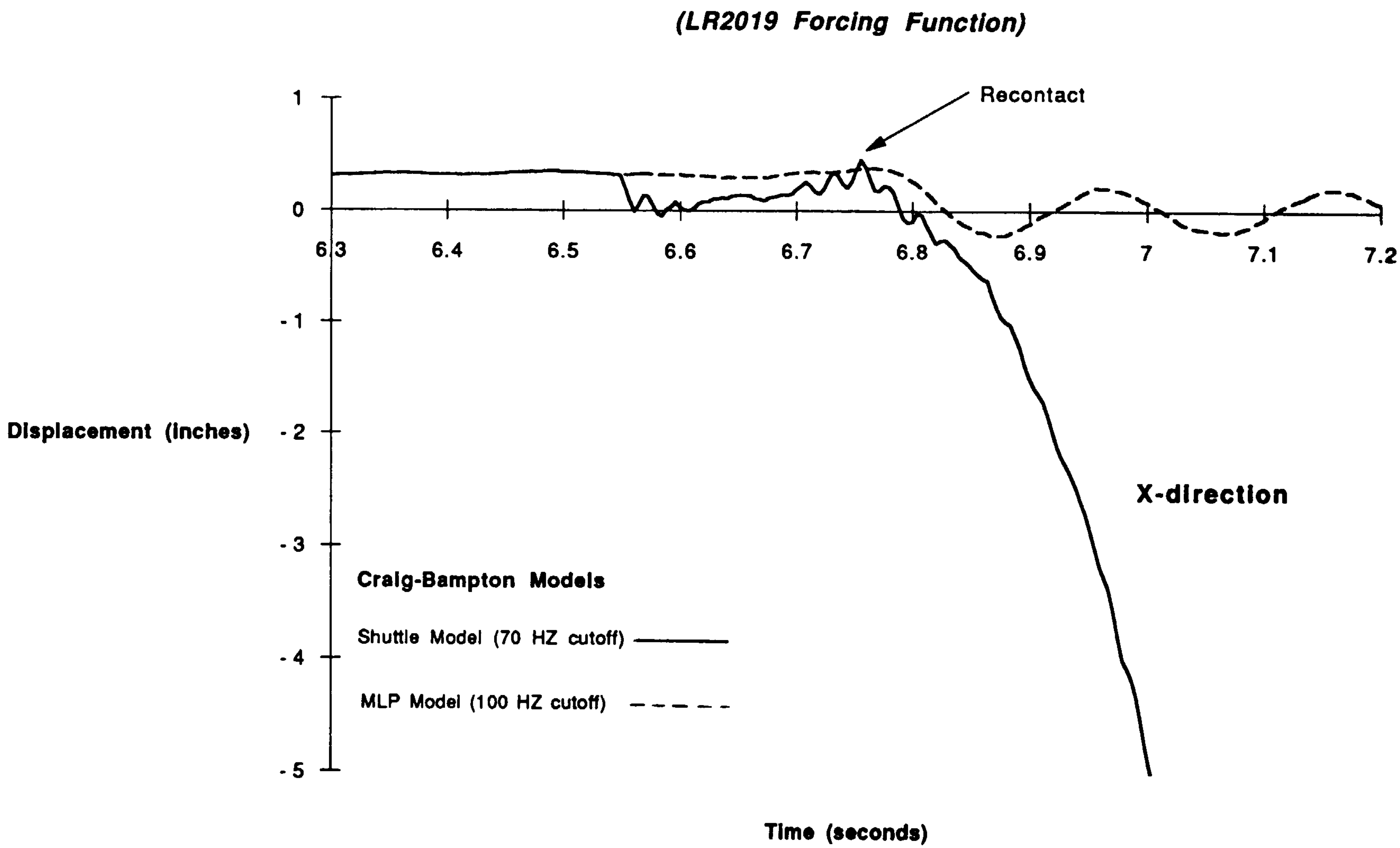


Figure 24. Proposed transient response method post 1 boundary displacement (inches) versus time (seconds).

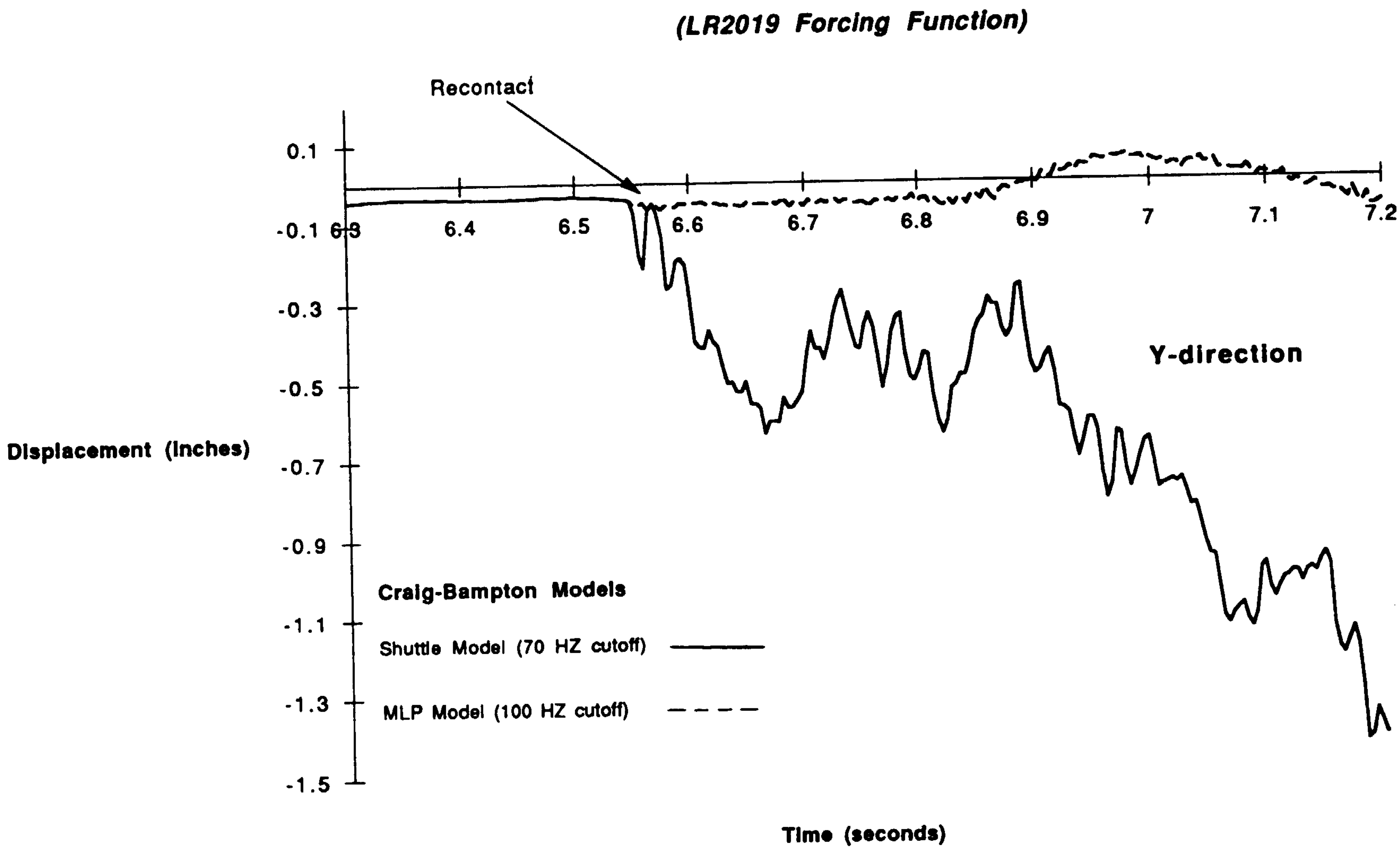


Figure 25. Proposed transient response method post 1 boundary displacement (inches) versus time (seconds).

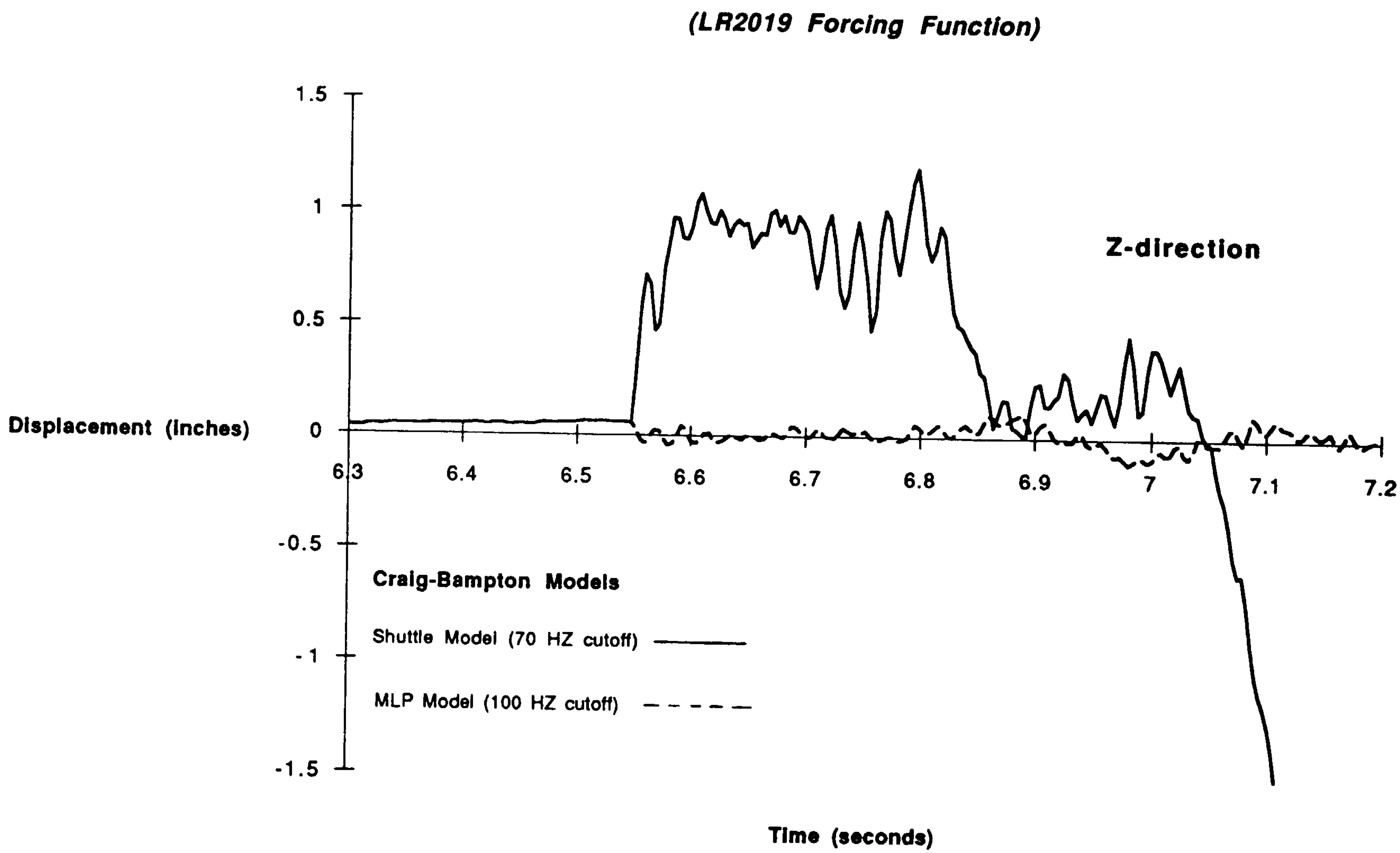


Figure 26. Proposed transient response method post 1 boundary displacement (inches) versus time (seconds).

V. CONCLUSIONS

A proposed method has been presented for the liftoff transient response analysis of the space shuttle vehicle using reduced models. The proposed method is validated with the numerical examples of a simple beam problem and the liftoff simulation of the space shuttle vehicle. Several different reduced models of the space shuttle liftoff vehicle model (four by the Craig and Bampton CMS method and three by the Lanczos CMS method) have been analyzed and studied. The following observations are made for the simple beam problem:

1. A beam model reduced by the Craig and Bampton CMS method gives accurate frequencies.

2. A beam model reduced by the Lanczos CMS method gives accurate frequencies depending on the number of Lanczos vectors computed.

3. The proposed transient response method gives good results for the beam problem analyzed. The integration time step is critical for accurate results. Therefore, the number of modes or vectors used to reduce the beam model is important, since this determines the cutoff frequency which then determines the integration time step.

The transient response analysis of the space shuttle vehicle during liftoff resulted in the following observations and conclusions:

1. The larger complex space shuttle liftoff model reduced by the Lanczos CMS method gives accurate frequencies depending on the number of Lanczos vectors computed.

2. A substantial savings in computer time is gained for the reduction of the space shuttle liftoff model using the Lanczos CMS method over that of the Craig and Bampton CMS method.

3. Computer time increased substantially when a large number of Lanczos vectors were computed. This is due to input/output computer time increasing.

4. Comparisons of the proposed transient response method for the space shuttle vehicle with the iterative method used by Martin Marietta have been made. The following conclusions are noted.

a. The differences observed between the two methods in the lateral directions and during separation can be attributed to the coupling stiffness matrix and the separation criterion used in the iterative method by Martin Marietta.

b. The proposed transient response method can save computer time over the iterative method used by Martin Marietta, since the iterative method requires a smaller time step and numerous iterations for convergence.

5. The proposed transient response analysis using reduced space shuttle vehicle models by the Craig and Bampton CMS method gives good results for the interface loads, even for reduced space shuttle models using a 20-Hz cutoff frequency.

6. The proposed transient response analysis using a reduced space shuttle vehicle model by the Lanczos CMS method is shown to give adequate interface loads as compared to the reduced models by the Craig and Bampton CMS method (70-Hz cutoff frequency), even for the smaller 10 Lanczos vector model.

7. The interface loads computed using models that have been reduced by the Lanczos CMS method have less frequency content in their responses as compared to the models reduced using the Craig and Bampton CMS method.

8. As the vehicle leaves the pad the possibility of chatter or reconnections needs to be investigated. It appears that, based on this analysis, this phenomenon does occur during liftoff. However, it is not incorporated in the present space shuttle liftoff analysis.

Areas of possible improvement to the proposed method include: recontact dynamics (chatter effect), improved separation criterion that has lateral load release, and modeling of the physical release mechanism. Another concern which has occurred on several space shuttle flights is the holddown bolts hanging up. The proposed method could be modified to analyze this phenomenon and determine its effects on the vehicle during liftoff separation.

REFERENCES

1. Admire, J.R.: "Modal Analysis of Structures by an Iterative Rayleigh-Ritz Technique." NASA Technical Memorandum, NASA TM X-64528, 1970, pp. 5–12.

2. Bathe, K.J.: "Solution Methods of Large Generalized Eigenvalue Problems in Structural Engineering." Civil Engineering Department, University of California, Berkeley, California, Report UC SESM 71-20, 1971.

3. Bathe, K.J., and Wilson, E.L.: "Large Eigenvalue Problems in Dynamic Analysis." A.S.C.E., Journal of Engineering Mechanics Division, Vol. 98, 1972, pp. 1471–1485.

4. Guyan, R.J.: "Reduction of Stiffness and Mass Matrices." AIAA Journal, Vol. 3, February 1965, p. 380.

5. Gladwell, G.M.L.: "Branch Mode Analysis of Vibrating Systems." Journal of Sound and Vibration, Vol. 1, 1964, pp. 41–59.

6. Hurty, W.C.: "Dynamic Analysis of Structural Systems Using Component Mode Synthesis." AIAA Journal, Vol. 3, No. 4, April 1965, pp. 678–685.

7. Craig, R.R., Jr., and Bampton, M.C.C.: "Coupling of Substructures for Dynamic Analyses." AIAA Journal, Vol. 6, July 1968, pp. 1313–1319.

8. Benfield, W.A., and Hruda, R.F.: "Vibration Analysis of Structures by Component Mode Substitution." AIAA Journal, Vol, 9, July 1971, pp. 1255–1261.

9. Hintz, R.M.: "Analytical Methods in Component Modal Synthesis." AIAA Journal, Vol. 13, August 1975, pp. 1007–1016.

10. MacNeal, R.M.: "A Hybrid Method of Component Mode Synthesis." Computers and Structures Journal, Vol. 1, 1971, pp. 581–601.

11. Rubin, S.: "Improved Component-Mode Representation for Structural Dynamic Analysis." AIAA Journal, Vol. 13, August 1975, pp. 995–1006.

12. Hale, A.L., and Meirovitch, L.: "A General Substructure Synthesis Method for the Dynamic Simulation of Complex Structures." Journal of Sound and Vibration, Vol. 69, No. 2, 1980, pp. 209–326.

13. Ojalvo, I.U.: "Origins and Advantages of Lanczos Vectors for Large Dynamic Systems." Proceedings of the SEM Sixth International Modal Analysis Conference, Vol. 1, February 1988, pp. 489–494.

14. Nour-Omid, B., and Regelbrugge, M.E.: "Lanczos Method for Dynamic Analysis of Damped Structural Systems." Proceedings fo the SEM Sixth International Modal Analysis Conference, Vol. 1, February 1988, pp. 495–500.

15. Craig, R.R., Jr., Kim, H.M., and Su T.: "Some Applications of Lanczos Vectors in Structural Dynamics." Proceedings of the SEM Sixth International Modal Analysis Conference, Vol. 1, February 1988, pp. 501–506.

16. Allen, J.J.: "A Component Synthesis Method Using Lanczos Vectors." Proceedings of the SEM Sixth International Modal Analysis Conference, Vol. 1, February 1988, pp. 507–513.

17. Leger, P.: "Application of Load-Dependent Vectors Bases for Dynamic Substructure Analysis." AIAA Journal, Vol. 28, No. 1, January 1990, pp. 177–179.

18. Blejwas, T.E.: "Shuttle System Dynamic Loads Analysis." Martin Marietta Denver, MCR-76-161, February 1976.

19. White, C.W., and Bodley, C.: "Boundary Interchanges Between Redundantly Connected Complex Structures." MMC Denver, Report Number R78-48628-001, 1978.

20. Olberding, D.J.: "Simulating the Dynamic Separation of Launch Vehicles From Their Support Structures." AIAA Journal, Paper Number 84-0948, 1984, pp. 180–182.

21. Prabhakar, A.: "Development of a Liftoff Response Program to Obtain Launch Loads for ACC-OTV." MMC Michoud IOM 3524-85-260, November 1985.

22. Flannigan, C.C.: "A Residual Flexibility Approach to Launch Vehicle Liftoff Loads Analysis." AIAA/ASME/ASCE/AHS/ASC 30th Structures, Structural Dynamics, and Material Conference, 89-1336-CP, April 1989, pp. 1541–1545.

23. Admire, J.R., and Brunty, J.A.: "A Transient Response Method for Linear Coupled Substructures." NASA Technical Paper 2926, December 1989.

24. Rockwell International, STS Dynamic Math Models (M6.0ZA) for Payload Loads Analysis, STS81-0641f, July 1988.

25. Computer Data Transmission of Hybrid Mobile Launch Platform (MLP) Finite Element Model, NASA/JSC, January 1988.

26. Computer Tape and Description, Rockwell International, LR2000 Series Liftoff Forcing Functions, April 1988.

27. FORtran Matrix Analysis (FORMA) Library, MSFC NASA, updated for the Cray XMP computer, August 1988.

APPENDIX

```
      N1=IIVEC(I)
      IF(N1.GT. NSYS)GO TO 999
      IVEC(N1)=I
      NB=NB+1
    2 CONTINUE
      NI=N-NB
      NERROR=3
      IF(NI.NE.NII)GO TO 999
      N1=0
      N2=0
      DO 3 I = 1,NSYS
      IF(IVEC(I).EQ.0)GO TO 7
      N2=N2+1
      IVEC(I)=IVEC(I)+NI
      GO TO 3
    7 N1=N1+1
      IVEC(I)=N1
    3 CONTINUE
      NERROR=4
      IF(N1.NE.NI  .OR. N2.NE.NB) GO TO 999
C REORDER M AND K
      CALL ZREAD(NMSK)
      CALL ZSIZE(NMSK,'K',N1,N2,N,N)
      NERROR=5
      IF(N1.NE.N2)GO TO 999
      NERROR=6
      IF(N1.NE.NSYS)GO TO 999
      CALL ZZERO(NMSKB,N,N)
      CALL ZRVAD(1.,NMSK,IVEC,IVEC,N,N,NMSKB,NMS2)
C
      CALL ZREAD(NMSM)
      CALL ZSIZE(NMSM,'M',N1,N2,N,N)
      NERROR=7
      IF(N1.NE.N2)GO TO 999
      NERROR=8
      IF(N1.NE.NSYS)GO TO 999
      CALL ZZERO(NMSMB,N,N)
      CALL ZRVAD(1.,NMSM,IVEC,IVEC,N,N,NMSMB,NMS2)
C CHECK DIAGONAL OF MASS MATRIX
C TO ENSURE POSITIVE DEFINITE (I.E. NO ZERO'S ON DIAGONAL)
      CALL ZDIAGR(NMSMB,NMS1)
C TAKE OUT DIAG OF MASS MATRIX
      CALL ZDIAGR(NMS1,NMSWK1)
```

```
      CALL ZAABB(-1.0,NMSWK1,1.0,NMSMB,NMSWK2)
      CALL ZAA(1.0,NMSWK2,NMSMB)
      CALL ZTOD(NMS1,W,N1,N2,1,KR)
      IFTEST=0
      NSUM=0
      DO 9 I=1,N
      IF(W(I).GT.0)GO TO 9
      IF(W(I).EQ.0)NSUM=NSUM+1
      IFTEST=1
      PRINT *,'NEGATIVE OR ZERO ON DIAGONAL OF MASS MATRIX'
      PRINT *,'LOCATED AT ROW/COL'
      PRINT,*,I
      PRINT *,'REPLACED WITH .000002 VALUE'
      W(I)=.000002
    9 CONTINUE
      PRINT *,'SUM OF ZEROS ON DIAG.=',NSUM
      IF(IFTEST.EQ.0)GO TO 31
      CALL DTOZ(W,NMS1,1,N,1,KR)
   31 CONTINUE
      CALL ZDIAGR(NMS1,NMS2)
      CALL ZAABB(1.,NMS2,1.,NMSMB,NMS1)
      CALL ZAA(1.,NMS1,NMSMB)
C REORDER TRANSFORMATION MATRIX
      CALL ZUNITY(NMS1,N)
      CALL ZZERO(NMSTB,N,N)
      CALL ZRVADC(1.,NMS1,IVEC,N,NMSTB)
C COMPUTE FIXED MODES
      CALL ZDISA(NMSKB,1,1,NI,NI,NMS1)
      CALL ZDISA(NMSMB,1,1,NI,NI,NMS2)
      IF(NSYS.LE.100)THEN
      NUT1=4
      CALL ZTOD(NMS1,AK,NI,NI,KR,KR)
      CALL ZTOD(NMS2,AM,NI,NI,KR,KR)
C     CALL WRITE(AK,NI,NI,'K',KR)
C     CALL WRITE(AM,NI,NI,'M',KR)
      CALL MCHCPU(TSEC)
      PRINT *,'CPU TIME BEFORE MODE1 ',TSEC,' SEC.'
      CALL MODE1(AM,AK,W2,W,FREQ,NI,-1.,KR,NUT1)
      CALL MCHCPU(TSEC)
      PRINT *,'CPU TIME AFTER MODE1 ',TSEC,' SEC.'
      ELSE
      NERROR=99
      GO TO 999
```

```
      END IF
C  TRUNCATE MODES AND BUILD TCB
      NERROR=9
      DO 21 I=1,NI
      IF(W2(I).LE.O)GO TO 999
C     IF(W2(I).LE.CUTW2)NQ=I
   21 CONTINUE
      NQ=NVEC
      CALL WRITE(W2,1,NQ,'W2',1)
      M=NQ+NB
      NQP=NQ+1
      CALL DTOZ(AM,NMS3,NI,NI,KR,KR)
      CALL ZDISA(NMS3,1,1,NI,NQ,NMS4)
      CALL ZZERO(NMSTCB,N,M)
      CALL ZASSEM(NMS4,1,1,NMSTCB)
      CALL ZUNITY(NMS4,NB,NB)
      CALL ZASSEM(NMS4,NQP,NQP,NMSTCB)
      CALL ZSRED2(NMSKB,NMS1,NMS2,NB,1,NMS3)
      CALL ZASSEM(NMS2,1,NQP,NMSTCB)
C  CALCULATE KCB AND MCB AND CHECK
      CALL ZBTAB(NMSKB,NMSTCB,NMS1,NMS2)
      CALL ZBTAB(NMSMB,NMSTCB,NMS2,NMS3)
      CALL ZAA(1.,NMS1,NMSKCB)
      CALL ZAA(1.,NMS2,NMSMCB)
      CALL ZUNITY(NMS3,NQ,NQ)
      CALL ZASSEM(NMS3,1,1,NMSMCB)
      CALL DTOZ(W2,NMS4,1,NI,1,KR)
      CALL ZDISA(NMS4,1,1,1,NQ,NMS3)
      CALL ZDIAGR(NMS3,NMS4)
      CALL ZASSEM(NMS4,1,1,NMSKCB)
      CALL ZMULT(NMSTB,NMSTCB,NMS3)
      CALL ZAA(1.,NMS3,NMSTCB)
      CALL ZWDISK(NMSKCB,'KCB',NDISK)
      CALL ZWDISK(NMSMCB,'MCB',NDISK)
      CALL ZWDISK(NMSTCB,'TCB',NDISK)
      CALL ZLDISK(NDISK)
      CALL ZCOMPR(NMS1,NMSKCB,3,1.OE-04,'STIF','NMSKCB',50)
      CALL ZCOMPR(NMS2,NMSMCB,3,1.OE-04,'MASS','NMSMCB',50)
      CALL ZTOD(NMSKCB,AK,M,M,KR,KR)
      CALL ZTOD(NMSMCB,AM,M,M,KR,KR)
C     CALL WRITE(AK,M,M,'K',KR)
C     CALL WRITE(AM,M,M,'M',KR)
      CALL MODE1(AM,AK,W2,W,FREQ,M,-1.,KR,NUT1)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
CRAIGB      PAGE 5                   ON=CELPQRSUV                  02/21/90-16:45:43    CFT 1.16BFO(06/29/89)   PAGE 5
     173     173.         PRINT *,'FREQ.(HZ)'
     174     174.         DO 501 I=1,M
     175     175.           PRINT *,FREQ(I)
     176     176.     501 CONTINUE
     177     177.         GO TO 10
     178     178.     999 CALL ZZBOMB('CBRUN1',NERROR)
     179     179.         END
CRAIGB                         VECTOR LOOP BEGINS AT SEQ. NO.            37, P=      72c
CRAIGB                LOOP USES VECTOR LENGTH OF 8 AT SEQ. NO.          133, P=     523c
```

ORIGINAL PAGE IS
OF POOR QUALITY

Computer Routine for the Reduction
of a Mass and Stiffness Matrix
using Lanczos CMS Method

```
      PROGRAM MAIN
      DIMENSION IIVEC(1750),IVEC(1750),W(1750)
C     DIMENSION AK(1750,1750),AM(1750,1750),W2(1750),FREQ(1750)
C    *          XPHI(1750,1750)
      CHARACTER*8 NDNAM
      DATA KR,KV/1750,1750/
      DATA NUT1/4/
      DATA NMSK,NMSM,NMS1,NMS2,NMSMLA,NMSKLA/
     *      .21,  22.  23.  24.   25.   26/
      DATA NMS3,NMS4,NMSTB,NMSTLA,NMSTC/
     *       27.  28.  29.  30.   31/
      DATA NMSMB,NMSKB,NMSWK1,NMSWK2,NMXPHI/
     *         32.  33.   34.     35.      36/
      CALL ZFIRST(21,'WORKFL  ')
   10 CALL START
      READ(5,*) NUMDSK
      IF(NUMDSK.LE.0)GO TO 30
      DO 20 II=1,NUMDSK
      READ(5,13) NDNAM,NDNUM
   13 FORMAT(A8,2X,I5)
      CALL ZOPNFL(NDNUM,NDNAM,1)
      CALL ZLDISK(NDNUM)
   20 CONTINUE
   30 CONTINUE
      READ(5,15) NDNAM,NDISK,NBB,NII,NVEC
   15 FORMAT(A8,2X,4I5,F8.3)
      PRINT *,'NDNAM,NDISK,NBB,NII,NVEC'
      PRINT *,NDNAM,NDISK,NBB,NII,NVEC
      NDNUM=IABS(NDISK)
      CALL ZOPNFL(NDNUM,NDNAM,1)
      IF(NDISK .GT. 0) GO TO 16
      NDISK=IABS(NDISK)
      CALL INZSAV(NDISK)
   16 CONTINUE
      NSYS=NBB+NII
      PRINT *,'NSYS = ',NSYS
C READ IN IVEC OF INTERFACES REVERSE ORDER
      CALL READIM(IIVEC,N1,N,1,KV)
      NERROR=1
      IF(N.NE.NSYS)GO TO 999
      DO 1 I=1,NSYS
    1 IVEC(I)=0
      NB=0
```

```
      NERROR=2
      DO 2 I=1,NSYS
      IF(IIVEC(I) .EQ. 0) GO TO 2
      IF(IIVEC(I) .LT. 0)GO TO 999
      N1=IIVEC(I)
      IF(N1.GT. NSYS)GO TO 999
      IVEC(N1)=I
      NB=NB+1
    2 CONTINUE
      NI=N-NB
      NERROR=3
      IF(NI.NE.NII)GO TO 999
      N1=0
      N2=0
      DO 3 I = 1,NSYS
      IF(IVEC(I).EQ.0)GO TO 7
      N2=N2+1
      IVEC(I)=IVEC(I)+NI
      GO TO 3
    7 N1=N1+1
      IVEC(I)=N1
    3 CONTINUE
      NERROR=4
      IF(N1.NE.NI .OR. N2.NE.NB) GO TO 999
C REORDER M AND K
      CALL ZREAD(NMSK)
C     PRINT *,'STIFFNESS MATRIX'
C     DO 550 I=1,NSYS
C     READ *, (AK(I,J),J=1,NSYS)
C     PRINT *,(AK(I,J),J=1,NSYS)
C50   CONTINUE
C     CALL DTOZ(AK,NMSK,NSYS,NSYS,KR,KR)
C     CALL ZTOD(NMSK,AK,NSYS,NSYS,KR,KR)
      CALL ZSIZE(NMSK,'K',N1,N2,N,N)
      NERROR=5
      IF(N1.NE.N2)GO TO 999
      NERROR=6
      IF(N1.NE.NSYS)GO TO 999
      CALL ZZERO(NMSKB,N,N)
      CALL ZRVAD(1.,NMSK,IVEC,IVEC,N,N,NMSKB,NMS2)
C
      CALL ZREAD(NMSM)
C     PRINT *,'MASS MATRIX'
```

```
C       DO 551 I=1,NSYS
C       READ *, (AM(I,J),J=1,NSYS)
C       PRINT *,(AM(I,J),J=1,NSYS)
C       CONTINUE
C
C
C       CALL DTOZ(AM,NMSM,NSYS,NSYS,KR,KR)
C       CALL ZTOD(NMSM,AM,NSYS,NSYS,KR,KR)
        CALL ZSIZE(NMSM,'M',N1,N2,N,N)
        NERROR=7
        IF(N1.NE.N2)GO TO 999
        NERROR=8
        IF(N1.NE.NSYS)GO TO 999
C
C  RUN EIGEN-VALUE PROBLEM FOR COMPARISON STUDY
C
C  REMEMBER AM IS CHANGED TO PHI AFTER MODE1
C       CALL MCHCPU(TSEC)
C       PRINT *,'CPU TIME BEFORE MODE1 = ',TSEC,' SEC.'
C       CALL MODE1(AM,AK,W2,W,FREQ,NSYS,-1.,KR,NUT1)
C       CALL MCHCPU(TSEC)
C       PRINT *,'CPU TIME AFTER MODE1 = ',TSEC,' SEC.'
C       PRINT *,'UNREDUCED SYSTEM FREQUENCIES (HZ)'
C       DO 552 I=1,NSYS
C         PRINT *,FREQ(I)
C52     CONTINUE
C CHECK DIAGONAL OF MASS MATRIX
C TO ENSURE POSITIVE DEFINITE (I.E. NO ZERO'S ON DIAGONAL)
        CALL ZZERO(NMSMB,N,N)
        CALL ZRVAD(1.,NMSM,IVEC,IVEC,N,N,NMSMB,NMS2)
        CALL ZDIAGR(NMSMB,NMS1)
C TAKE OUT DIAG OF MASS MATRIX
        CALL ZDIAGR(NMS1,NMSWK1)
        CALL ZAABB(-1.0,NMSWK1,1.0,NMSMB,NMSWK2)
        CALL ZAA(1.0,NMSWK2,NMSMB)
        CALL ZTOD(NMS1,W,N1,N2,1,KR)
        IFTEST=0
        NSUM=0
        DO 9 I=1,N
        IF(W(I).GT.0)GO TO 9
        IF(W(I).EQ.0)NSUM=NSUM+1
        IFTEST=1
        PRINT *,'NEGATIVE OR ZERO ON DIAGONAL OF MASS MATRIX'
```

```
MAIN      PAGE 4               ON=CELPQRSUV               04/14/90-20:12:44   CFT 1.16RFO(06/29/89)
      PRINT *,'LOCATED AT ROW/COL
      PRINT *,I
      PRINT *,'REPLACED WITH .000002 VALUE'
      W(I)=.000002
    9 CONTINUE
      PRINT *,'SUM OF ZEROS ON MASS MATRIX DIAG.=',NSUM
      IF(IFTEST.EQ.0)GO TO 31
      CALL DTOZ(W,NMS1,1,N,1,KR)
   31 CONTINUE
      CALL ZDIAGR(NMS1,NMS2)
      CALL ZAABB(1.,NMS2,1.,NMSMB,NMS1)
      CALL ZAA(1.,NMS1,NMSMB)
C REORDER TRANSFORMATION MATRIX
      CALL ZUNITY(NMS1,N)
      CALL ZZERO(NMSTB,N,N)
      CALL ZRVADC(1.,NMS1,IVEC,N,NMSTB)
C COMPUTE LANCZOS VECTORS
      CALL ZDISA(NMSKB,1,1,NI,NI,NMS1)
      CALL ZDISA(NMSMB,1,1,NI,NI,NMS2)
      IF(NSYS.LE.1750)THEN
      CALL MCHCPU(TSEC)
      PRINT *,'CPU TIME BEFORE LANCZOS VECTORS = ',TSEC,' SEC.
C     IF(NI.LE.300)THEN
C       CALL ZTOD(NMS1,AK,NI,NI,KR,KR)
C       CALL ZTOD(NMS2,AM,NI,NI,KR,KR)
C       CALL WRITE(AK,NI,NI,'K',KR)
C       CALL WRITE(AM,NI,NI,'M',KR)
C       PRINT *,'NI, NVEC, AND KR =',NI,NVEC,KR
C       CALL LANCZ(AM,AK,XPHI,NI,NVEC,KR)
C     ELSE
        CALL ZLANCZ(NMS2,NMS1,NMXPHI,NI,NVEC,NDISK)
C       CALL ZTOD(NMXPHI,XPHI,N1,N2,KR,KR)
        CALL ZSIZE(NMXPHI,'LPHI',N1,N2,NI,NVEC)
        NERROR=90
        IF(N1.NE.NI)GO TO 999
        NERROR=91
        IF(N2.NE.NVEC)GO TO 999
C     ENDIF
      CALL MCHCPU(TSEC)
      PRINT *,'CPU TIME AFTER LANCZOS VECTORS = ',TSEC,' SEC.
      ELSE
      NERROR=99
      GO TO 999
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
          END IF
C     BUILD TLA TRANSFORMATION
         NQ=NVEC
         M=NQ+NB
         NQP=NQ+1
         PRINT *,'NQ  NB  NQP  M  NVEC'
         PRINT *,NQ,NB,NQP,M,NVEC
C        CALL DTOZ(XPHI,NMS3,NI,NQ,KR,KR)
         CALL ZDISA(NMXPHI,1,1,NI,NQ,NMS4)
         CALL ZZERO(NMSTLA,N,M)
         CALL ZASSEM(NMS4,1,1,NMSTLA)
         CALL ZSRED2(NMSKB,NMS1,NMS2,NB,1,NMS3)
C        CALL ZWRITE(NMS2,'T')
C        CALL ZWRITE(NMS1,'K')
         CALL ZASSEM(NMS2,1,NQP,NMSTLA)
C     CALCULATE KLA AND MLA AND CHECK
         CALL ZBTAB(NMSKB,NMSTLA,NMS1,NMS2)
         CALL ZBTAB(NMSMB,NMSTLA,NMS2,NMS3)
         CALL ZAA(1.,NMS1,NMSKLA)
         CALL ZAA(1.,NMS2,NMSMLA)
C        CALL ZUNITY(NMS3,NQ,NQ)
C        CALL ZASSEM(NMS3,1,1,NMSMLA)
         CALL ZMULT(NMSTB,NMSTLA,NMS3)
         CALL ZAA(1.,NMS3,NMSTLA)
         CALL ZWDISK(NMSKLA,'KLA',NDISK)
         CALL ZWDISK(NMSMLA,'MLA',NDISK)
         CALL ZWDISK(NMSTLA,'TLA',NDISK)
         CALL ZLDISK(NDISK)
         CALL ZCOMPR(NMS1,NMSKLA,3,1.OE-04,'STIF','NMSKLA',50)
         CALL ZCOMPR(NMS2,NMSMLA,3,1.OE-04,'MASS','NMSMLA',50)
C
C     COMPUTE EIGENVALUES FOR COMPARISON STUDY
C
C     REMEMBER AM IS CHANGED TO PHI AFTER MODE1
C        CALL ZTOD(NMSKLA,AK,N1,N2,KR,KR)
C        CALL ZTOD(NMSMLA,AM,N1,N2,KR,KR)
C        CALL WRITE(AM,N1,N2,'MASS-RED',KR)
C        CALL WRITE(AK,N1,N2,'STIF RED',KR)
C        CALL ZERO(FREQ,NSYS,1,KR)
C        CALL MODE1(AM,AK,W2,W,FREQ,N1,-1.,KR,NUT1)
C        PRINT *,'ALTERED MASS AND STIFFNESS FREQUENCIES (HZ)'
C        DO 553 I=1,NSYS
C          PRINT *,FREQ(I)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
MAIN       PAGE 6                ON=CELPQRSUV              04/14/90-20:12:44   CFT 1.16BFO(06/29/89)  PAGE 6

    216    216. C53   CONTINUE
    217    217.       GO TO 10
    218    218.   999 CALL ZZBOMB('MAIN',NERROR)
    219    219.       END
MAIN                       VECTOR LOOP BEGINS AT SEQ. NO.          41, P=      72c
```

```
      SUBROUTINE ZLANCZ (NMSM,NMSK,NMXPHI,N,NVEC,NDISK)
      DIMENSION F(1750,1),C1(1),CC1(1750),Z1(1)
C     DIMENSION S(300,300),SINV(300,300)
      DATA KA,KS/1750,300/
      DATA NMKINV,NMX1,NMZ,NMZZ,NMC1/
     *           1,   2,  3,   4,   9/
      DATA NMSUM,NMF,NMAA,NMSU,NMSD,NMS1,NMS2,NMS3,NMS4/
     *         13, 14,  15,  16,  17,  18,  19,  20,  21/
      DATA NMZ1,NMCC1/
     *       22,  23/
C
C  CALCULATE LANCZOS VECTORS FOR A GIVEN MASS AND
C  STIFFNESS MATRICES.
C
C  REFERENCE:  E.L. WILSON, M. YUAN AND J.M DICKENS,
C              "DYNAMIC ANALYSIS BY DIRECT SUPERPOSITION
C              OF RITZ VECTORS", EARTHQUAKE ENG. STRUCT.
C              DYN., 10, 813-821, 1979.
C
C  THE MASS (NMSM) MATRIX MUST BE REAL, SYMMETRIC, POSITIVE DEFINITE.
C  THE STIF (NMSK) MATRIX MUST BE REAL, SYMMETRIC.
C  CALLS FORMA SUBROUTINES ZBTAB,ZMULT,ZDISA,ZATXB,ZASSEM,ZZERO
C                          ZZBOMB,ZAASUM,ZDCOM2,ZBSOL2,ZUNITY
C
C  CODED BY  J.A. BRUNTY FEB. 1990
C
C     SUBROUTINE ARGUMENTS
C  NMSM     = INPUT  MASS MATRIX. SIZE(N,N).
C  NMSK     = INPUT  STIFFNESS MATRIX. SIZE(N,N).
C  NMXPHI   = OUTPUT MATRIX OF LANCZOS VECTORS. SIZE(N,NVEC).
C  N        = INPUT  ROW AND COLUMN DIMENSION OF NMSM,NMSK
C  NVEC     = INPUT  NUMBER OF LANCZOS VECTORS WANTED FROM NMSM,NMSK
C
      PRINT *,'******* COMPUTING LANCZOS VECTORS *******'
      NERROR = 1
      IF(N.GT.KA)GO TO 999
      NERROR = 2
      IF(NVEC.GT.N)GO TO 999
C
C INVERT STIFFNESS MATRIX USING PARTITION LOGIC
C
C     IF(N.LE.300)THEN
C       CALL ZTOD(NMSK,S,N1,N2,KS,KS)
```

```
C         CALL INV2(S,SINV,N1,KS)
C         CALL DTOZ(SINV,NMKINV,N1,N2,KS,KS)
C       ELSE
          CALL ZUNITY(NMS1,N,N)
          CALL ZDCOM2(NMSK,NMSU,NMSD)
          CALL ZBSOL2(NMSU,NMSD,NMS1,NMKINV)
          CALL ZWDISK(NMKINV,'KINV',NDISK)
C       ENDIF
C
C  COMPUTE K¬-1 M USED IN LANCZOS VECTOR COMPUTATIONS
C
        CALL ZMULT(NMKINV,NMSM,NMAA)
C
C  SET UP INITIAL VECTOR USING STATIC DISPLACEMENT
C
        CALL ZERO(F,N,1,KA)
        DO 100 I=1,N
          F(I,1)=RANF()
  100   CONTINUE
        CALL DTOZ(F,NMF,N,1,KA,1)
        CALL ZMULT(NMKINV,NMF,NMS3)
C
C  ORTHOGONALIZE (NORMALIZE) THE INITIAL VECTOR USING MASS MATRIX
C
        CALL ZBTAB(NMSM,NMS3,NMS1,NMS2)
        CALL ZTOD(NMS1,Z1,N1,N2,1,1)
        IF(Z1(1).LE.0)THEN
          PRINT *,'Z1 = ',Z1(1)
        ENDIF
        C1(1) = 1/SQRT(Z1(1))
        CALL ZAA(C1,NMS3,NMX1)
        CALL ZZERO(NMXPHI,N,NVEC)
        CALL ZASSEM(NMX1,1,1,NMXPHI)
C
C  START RECURRENCE COMPUTATION OF LANCZOS VECTORS
C
        DO 20 I=2,NVEC
          CALL ZMULT(NMAA,NMX1,NMZZ)
            CALL ZMULT (NMSM,NMZZ,NMC1)
            CALL ZATXB (NMXPHI,NMC1,NMCC1)
            CALL ZTOD (NMCC1,CC1,N1,N2,KA,1)
          CALL ZZERO(NMSUM,N,1)
          DO 30 J=1,I-1
```

```
                CALL ZDISA (NMXPHI,1,J,N,1,NMZ)
                CALL ZAASUM (CC1(J),NMZ,NMSUM)
   30         CONTINUE
              CALL ZAABB(1.0,NMZZ,-1.0,NMSUM,NMS2)
C
C  ORTHOGONALIZE (NORMALIZE) THE VECTORS USING MASS MATRIX
C
              CALL ZBTAB (NMSM,NMS2,NMZ1,NMS1)
              CALL ZTOD(NMZ1,Z1,N1,N2,1,1)
              IF(Z1(1).LE.O)THEN
                PRINT *,'Z1 =',Z1(1)
                STOP
              ENDIF
              Z1(1) = 1/SQRT(Z1(1))
              CALL ZAA(Z1,NMS2,NMX1)
C
C  ASSEMBLE VECTORS INTO XPHI
C
              CALL ZASSEM (NMX1,1,I,NMXPHI)
              PRINT *,'LANCZOS VECTOR #',I,' COMPUTED'
   20       CONTINUE
            PRINT *,'CHECK OF LANCZOS VECTORS'
            CALL ZBTAB (NMSM,NMXPHI,NMAA,NMS1)
            CALL ZWRITE(NMXPHI,'LPHI')
            CALL ZWRITE(NMAA,'LT M L')
            PRINT *,'******* DONE COMPUTING LANCZOS VECTORS *******'
C
C  END OF SUBROUTINE
C
            RETURN
  999       CALL ZZBOMB('ZLANCZ',NERROR)
            END
```

ZLANCZ VECTOR LOOP BEGINS AT SEQ. NO. 60, P= 5b

```
      SUBROUTINE LANCZ (A,S,XPHI,N,NVEC,KR)
      DIMENSION A(KR,1),S(KR,1),XPHI(KR,1),SINV(300,300),
     1 X1(300,1),Z(300,1),ZZ(300,1),C1(300,1),SUM(300,1),F(300,1),
     * AA(300,300),Z1(1,1)
      DATA KA/300/
C
C CALCULATE LANCZOS VECTORS FOR A GIVEN MASS AND
C STIFFNESS MATRICES.
C
C REFERENCE:  E.L. WILSON, M. YUAN AND J.M DICKENS,
C             "DYNAMIC ANALYSIS BY DIRECT SUPERPOSITION
C             OF RITZ VECTORS", EARTHQUAKE ENG. STRUCT.
C             DYN., 10, 813-821, 1979.
C
C THE MASS (A) MATRIX MUST BE REAL, SYMMETRIC, POSITIVE DEFINITE.
C THE STIF (S) MATRIX MUST BE REAL, SYMMETRIC.
C CALLS FORMA SUBROUTINES BTAB,INV2,MULT,DISA,ATB1,ASSEM,ZERO,(ZZBUMB)
C THE MAXIMUM SIZE IS
C    N = 2500  (BASED ON BTABA, BTABA2).
C
C CODED BY  J.A. BRUNTY FEB. 1990
C
C    SUBROUTINE ARGUMENTS
C A      = INPUT  MASS MATRIX. SIZE(N,N).
C S      = INPUT  STIFFNESS MATRIX. SIZE(N,N).
C XPHI   = OUTPUT MATRIX OF LANCZOS VECTORS. SIZE(N,NVEC).
C N      = INPUT  ROW AND COLUMN DIMENSION OF A,S
C NVEC   = INPUT  NUMBER OF LANCZOS VECTORS WANTED FROM A,S
C KR     = INPUT  ROW DIMENSION OF A,S IN CALLING PROGRAM.
C
      PRINT *,'******* COMPUTING LANCZOS VECTORS *******'
      NERROR = 1
      IF(N.GT.KR .OR. NVEC.GT.KR)GO TO 999
      NERROR = 2
      IF(KR.NE.KA)GO TO 999
      CALL ZERO(XPHI,N,NVEC,KA)
C
C INVERT STIFFNESS MATRIX
C
      CALL INV2(S,SINV,N,KA)
C
C COMPUTE K-1 M USED IN LANCZOS VECTOR COMPUTATIONS
C
```

```
      CALL MULT(SINV,A,AA,N,N,N,KA,KA)
C
C  SET UP INITIAL VECTOR USING STATIC DISPLACEMENT
C
      CALL ZERO(F,N,1,KA)
      DO 100 I=1,N
        F(I,1)=RANF()
100   CONTINUE
      CALL MULT(SINV,F,X1,N,N,1,KA,KA)
C
C  ORTHOGONALIZE (NORMALIZE) THE INITIAL VECTOR USING MASS MATRIX (A)
C
      CALL BTAB (A,X1,Z1,N,1,KR,1)
      DO 10 II=1,N
        IF(Z1(1,1).LE.O.)THEN
          PRINT *,'Z1 =',Z1(1,1)
          STOP
        ENDIF
        X1(II,1) = X1(II,1) / SQRT(Z1(1,1))
10    CONTINUE
      CALL ASSEM(X1,1,1,XPHI,N,1,N,NVEC,KA,KA)
C
C  START RECURRENCE COMPUTATION OF LANCZOS VECTORS
C
      DO 20 I=2,NVEC
        CALL MULT(AA,X1,ZZ,N,N,1,KA,KA)
        CALL ZERO(SUM,N,1,KA)
        DO 30 J=1,I-1
          CALL DISA (XPHI,1,J,Z,N,NVEC,N,1,KA,KA)
          CALL MULT (A,ZZ,C1,N,N,1,KR,KA)
          CALL ATB1 (Z,C1,CC1,N,1,1,KA,KA,1)
          DO 40 JJ=1,N
            SUM(JJ,1) = SUM(JJ,1) + CC1 * Z(JJ,1)
40        CONTINUE
30      CONTINUE
        DO 50 J=1,N
          X1(J,1) = ZZ(J,1) - SUM(J,1)
50      CONTINUE
C
C  ORTHOGONALIZE (NORMALIZE) THE VECTORS USING MASS MATRIX (A)
C
C       CALL WRITE(X1,N,1,'X1',KA)
        CALL BTAB (A,X1,Z1,N,1,KR,1)
```

```
            DO 60 II=1,N
              X1(II,1) = X1(II,1) / SQRT(Z1(1,1))
   60       CONTINUE
C
C  ASSEMBLE VECTORS INTO XPHI
C
            CALL ASSEM(X1,1,I,XPHI,N,1,N,NVEC,KA,KA)
   20     CONTINUE
          PRINT *,'CHECK OF LANCZOS VECTORS'
          CALL BTAB(A,XPHI,AA,N,NVEC,KR,KA)
          CALL WRITE(XPHI,N,NVEC,'LPHI',KA)
          CALL WRITE(AA,NVEC,NVEC,'LTML',KA)
          PRINT *,'******* DONE COMPUTING LANCZOS VECTORS *******'
C
C  END OF SUBROUTINE
C
          RETURN
  999     CALL ZZBOMB('LANCZ',NERROR)
          END
LANCZ                 VECTOR LOOP BEGINS AT SEQ. NO.        49, P=      5b
LANCZ                 VECTOR LOOP BEGINS AT SEQ. NO.        75, P=    213c
LANCZ                 VECTOR LOOP BEGINS AT SEQ. NO.        79, P=    267a
LANCZ                 VECTOR LOOP BEGINS AT SEQ. NO.        87, P=    305c
```

Computer Routine for the Transient Response of the Space Shuttle Vehicle During Liftoff using the Proposed Method

```
      PROGRAM MAIN
      DATA NMGMS,NMGCS,NMGKS,NMQSO,NMQDSO,NMPHIS,NMDS/
     *     10,    11,    12,    13,    14,    15,   16/
      DATA NMGMP,NMGCP,NMGKP,NMQPO,NMQDPO,NMPHIP,NMDP/
     *     17,    18,    19,    20,    21,    22,   23/
      DATA NMTTBS,NMFTBS,NMTTBP,NMFTBP,NMFBO/
     *      24,    25,    26,    27,    28/
      DATA   NMST,  NMFB,  NMFGS,  NMFGP/
     *        29,    30,    31,     32/
      DATA   NMS1,NMS2,NMS3/
     *        33,  34,  35/
      CHARACTER*8 NDNAM
   10 CALL START
      CALL ZFIRST(21,'WORKFL  ')
      READ(5,11) NUMDSK
   11 FORMAT(I5)
      IF(NUMDSK .LE. 0) GO TO 30
      DO 20 I=1,NUMDSK
      READ(5,12) NDNAM,NDNUM
   12 FORMAT(A8,2X,I5)
      CALL ZOPNFL(NDNUM,NDNAM,1)
      CALL ZLDISK(NDNUM)
   20 CONTINUE
   30 CONTINUE
      READ(5,12) NDNAM,NDISK
      NDNUM=IABS(NDISK)
      CALL ZOPNFL(NDNUM,NDNAM,1)
      IF(NDISK .GT. 0) GO TO 40
      NDISK=IABS(NDISK)
      CALL INZSAV(NDISK)
   40 CONTINUE
      CALL ZLDISK(NDISK)
      CALL ZREAD(NMGMS)
      CALL ZREAD(NMGCS)
      CALL ZREAD(NMGKS)
      CALL ZREAD(NMQSO)
      CALL ZREAD(NMQDSO)
      CALL ZREAD(NMPHIS)
      CALL ZREAD(NMDS)
      CALL ZREAD(NMTTBS)
      CALL ZREAD(NMFTBS)
      CALL ZREAD(NMGMP)
      CALL ZREAD(NMGCP)
```

```
      CALL ZREAD(NMGKP)
      CALL ZREAD(NMQPO)
      CALL ZREAD(NMQDPO)
      CALL ZREAD(NMPHIP)
      CALL ZREAD(NMDP)
      CALL ZREAD(NMTTBP)
      CALL ZREAD(NMFTBP)
      CALL ZREAD(NMS1)
      CALL ZTRANS(NMS1,NMFBO)
      CALL ZREAD(NMFGS)
      CALL ZREAD(NMFGP)
      READ(5,*) TSTART,TEND,DT,NPRT
      CALL LIFOFF(TSTART,TEND,DT,NMGMS,NMGCS,NMGKS,NMQSO,
     *                    NMQDSO,NMPHIS,NMDS,NMTTBS,NMFTBS,NMGMP,
     *                    NMGCP,NMGKP,NMQPO,NMQDPO,NMPHIP,NMDP,
     *                    NMTTBP,NMFTBP,NMFBO,NPRT,
     *                    NMST,NMFB,NMFGS,NMFGP)
      CALL ZWDISK(NMST,'NMST',NDISK)
      CALL ZWDISK(NMFB,'NMFB',NDISK)
      CALL ZLDISK(NDISK)
      GO TO 10
  999 CALL ZZBOMB('MAIN',NERROR)
      END
```

```
      SUBROUTINE LIFOFF(TSTART,TEND,DT,NMGMS,NMGCS,NMGKS,NMQSO,
     *                  NMQDSO,NMPHIS,NMDS,NMTTBS,NMFTBS,NMGMP,
     *                  NMGCP,NMGKP,NMQPO,NMQDPO,NMPHIP,NMDP,
     *                  NMTTBP,NMFTBP,NMFBO,NPRT,
     *                  NMST,NMFB,NMFGS,NMFGP)
      PARAMETER (KQS=900,KB=24,KFS=175,KTS=700,KQP=100,
     *           KFP=50,KTP=600,KTIM=10001,KFSP=175,K3B=72)
CCCCC
      DIMENSION AA(50,5),SS(12,1),SM(KTIM),SXDD(KTIM),NFLAG(KB)
CCCCC
      DIMENSION GMS(KQS),GCS(KQS),GKS(KQS),QSO(KQS),QDSO(KQS),
     *          PHIS(KB,KQS),DS(KQS,KFS),TTBS(KFS,KTS),
     *          FTBS(KFS,KTS),GMP(KQP),GCP(KQP),GKP(KQP),
     *          QPO(KQP),QDPO(KQP),PHIP(KB,KQP),DP(KQP,KFP),
     *          TTBP(KFP,KTP),FTBP(KFP,KTP),FBO(KB),
     *          GRAVS(KQS),GRAVP(KQP)
      DIMENSION NCTS(KFS),NCTP(KFP),FWK(KFSP),GFS1(KQS),
     *          GFS2(KQS),GFP1(KQP),GFP2(KQP),GFSB(KQS),
     *          GFPB(KQP),TV(KTIM),QS(KQS),QDS(KQS),
     *          QDDS(KQS),QP(KQP),QDP(KQP),QDDP(KQP),
     *          FWK1(KQS),FWK2(KQP)
      DIMENSION COSA(KQS,KB),COSB(KQS,KB),COSC(KQS,KB),
     *          CQDSA(KQS,KB),CQDSB(KQS,KB),CQDSC(KQS,KB),
     *          CQDDSA(KQS,KB),CQDDSB(KQS,KB),CQDDSC(KQS,KB),
     *          COPA(KQP,KB),COPB(KQP,KB),COPC(KQP,KB),
     *          CQDPA(KQP,KB),CQDPB(KQP,KB),CQDPC(KQP,KB),
     *          CQDDPA(KQP,KB),CQDDPB(KQP,KB),CQDDPC(KQP,KB),
     *          COEMAT(K3B,K3B),CIVMAT(K3B,K3B),XBS(KB),XBP(KB),
     *          XDBS(KB),XDBP(KB),XDDBS(KB),XDDBP(KB)
CCCCC
      DATA NMJOE1,NMJOE2/
     *        71,    72/
CCCCC
      CALL PAGEHD
      WRITE(6,2000) TSTART,TEND,DT,NPRT
 2000 FORMAT(5X,'TSTART=',E12.5,'  TEND=',E12.5,'  DT=',E12.5,
     *       '  NPRT=',I4,///)
      IF(KFS .GT. KFSP .OR. KFP .GT. KFSP) GO TO 999
      CALL ZSIZE(NMGMS,'NMGMS',N1,NQS,1,0)
      CALL ZTOD(NMGMS,GMS,N1,N2,1,KQS)
      CALL ZTOD(NMGCS,GCS,N1,N2,1,KQS)
                                                   NERROR=1
      IF(N2 .NE. NQS) GO TO 999
```

```
      CALL ZTOD(NMGKS,GKS,N1,N2,1,KQS)
                                                  NERROR=2
      IF(N2 .NE. NQS) GO TO 999
      CALL ZTOD(NMQSO,QSO,N1,N2,1,KQS)
                                                  NERROR=3
      IF(N2 .NE. NQS) GO TO 999
      CALL ZTOD(NMQDSO,QDSO,N1,N2,1,KQS)
                                                  NERROR=4
      IF(N2 .NE. NQS) GO TO 999
      CALL ZTOD(NMPHIS,PHIS,NB,N1,KB,KQS)
                                                  NERROR=5
      IF(N1 .NE. NQS) GO TO 999
      CALL ZTOD(NMDS,DS,N1,NFS,KQS,KFS)
                                                  NERROR=6
      IF(N1 .NE. NQS) GO TO 999
      CALL ZSIZE(NMGMP,'NMGMP',N1,NQP,1,0)
      CALL ZTOD(NMGMP,GMP,N1,N2,1,KQP)
      CALL ZTOD(NMGCP,GCP,N1,N2,1,KQP)
                                                  NERROR=7
      IF(N2 .NE. NQP) GO TO 999
      CALL ZTOD(NMGKP,GKP,N1,N2,1,KQP)
                                                  NERROR=8
      IF(N2 .NE. NQP) GO TO 999
      CALL ZTOD(NMQPO,QPO,N1,N2,1,KQP)
                                                  NERROR=9
      IF(N2 .NE. NQP) GO TO 999
      CALL ZTOD(NMQDPO,QDPO,N1,N2,1,KQP)
                                                  NERROR=10
      IF(N2 .NE. NQP) GO TO 999
      CALL ZTOD(NMPHIP,PHIP,NB,N1,KB,KQP)
                                                  NERROR=11
      IF(N1 .NE. NQP) GO TO 999
      CALL ZTOD(NMDP,DP,N1,NFP,KQP,KFP)
                                                  NERROR=12
      IF(N1 .NE. NQP) GO TO 999
      CALL ZTOD(NMFBO,FBO,N1,N2,1,KB)
                                                  NERROR=13
      IF(N2 .NE. NB) GO TO 999
      CALL ZTOD(NMFGS,GRAVS,N1,N2,KQS,1)
                                                  NERROR=14
      IF(N1.NE.NQS) GO TO 999
      CALL ZTOD(NMFGP,GRAVP,N1,N2,KQP,1)
                                                  NERROR=15
```

```
      IF(N1.NE.NQP) GO TO 999
C
C     CONVERT FORCE AND TIME TABLES TO DENSE MATRICES
C     ALSO CHECK DATA AND FORM TIME VECTOR TV.
C
      CALL ZTOD(NMTTBS,TTBS,N1,NTS,KFS,KTS)
                                                NERROR=16
      IF(N1 .NE. NFS) GO TO 999
      CALL ZTOD(NMFTBS,FTBS,N1,N2,KFS,KTS)
                                                NERROR=17
      IF(N1 .NE. NFS .OR. N2 .NE. NTS) GO TO 999
                                                NERROR=18
      DO 20 I=1,NFS
      DO 10 J=1,NTS-1
      JP=J+1
      IF(TTBS(I,JP) .LE. 0.) GO TO 20
      IF(TTBS(I,J) .GE. TTBS(I,JP)) GO TO 999
   10 CONTINUE
   20 CONTINUE
      CALL ZTOD(NMTTBP,TTBP,N1,NTP,KFP,KTP)
                                                NERROR=19
      IF(N1 .NE. NFP) GO TO 999
      CALL ZTOD(NMFTBP,FTBP,N1,N2,KFP,KTP)
                                                NERROR=20
      IF(N1 .NE. NFP .OR. N2 .NE. NTP) GO TO 999
                                                NERROR=21
      DO 40 I=1,NFP
      DO 30 J=1,NTP-1
      JP=J+1
      IF(TTBP(I,JP) .LE. 0.) GO TO 40
      IF(TTBP(I,J) .GE. TTBP(I,JP)) GO TO 999
   30 CONTINUE
   40 CONTINUE
      TMIN=TTBS(1,1)
      DO 50 I=NTS,1,-1
      IF(TTBS(1,I) .LE. 0.) GO TO 50
      TMAX=TTBS(1,I)
      GO TO 60
   50 CONTINUE
                                                NERROR=22
      GO TO 999
   60 CONTINUE
      DO 70 I=1,NFS
```

```
      IF(TTBS(I,1) .GT. TMIN) TMIN=TTBS(I,1)
   70 CONTINUE
      DO 80 I=1,NFP
      IF(TTBP(I,1) .GT. TMIN) TMIN=TTBS(I,1)
   80 CONTINUE
      DO 110 I=1,NFS
      DO 90 J=NTS,2,-1
      IF(TTBS(I,J) .LE. 0.) GO TO 90
      IF(TTBS(I,J) .LT. TMAX) TMAX=TTBS(I,J)
      GO TO 100
   90 CONTINUE
                                                    NERROR=23
      GO TO 999
  100 CONTINUE
  110 CONTINUE
      DO 140 I=1,NFP
      DO 120 J=NTP,2,-1
      IF(TTBP(I,J) .LE. 0.) GO TO 120
      IF(TTBP(I,J) .LT. TMAX) TMAX=TTBP(I,J)
      GO TO 130
  120 CONTINUE
                                                    NERROR=24
      GO TO 999
  130 CONTINUE
  140 CONTINUE
      IF(TSTART .LT. TMIN) TSTART=TMIN
      IF(TEND .GT. TMAX) TEND=TMAX
      CALL COECAL(GMS,GCS,GKS,NOS,GMP,GCP,GKP,NOP,NB,DT,
     *            PHIS,PHIP,CQSA,CQSB,CQSC,CQDSA,CQDSB,
     *            CQDSC,CQDDSA,CQDDSB,CQDDSC,COPA,COPB,
     *            COPC,CQDPA,CQDPB,CQDPC,CQDDPA,CQDDPB,
     *            CQDDPC,COEMAT,KQS,KQP,KB,K3B)
      NB3=NB*3
      CALL INV2(COEMAT,CIVMAT,NB3,K3B)
      NTIM=IFIX((TEND-TSTART)/DT+.00001)+1
                                                    NERROR=25
      IF(NTIM .GT. KTIM) GO TO 999
      DO 150 I=1,NTIM
  150 TV(I)=TSTART+DT*(I-1)
      TSTART=TV(1)
      TEND=TV(NTIM)
      TS1=TV(1)
      TSO=TS1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
LIFOFF     PAGE 5              ON=CELPQRSUV              04/16/90-13:49:08    CFT 1.16BFO(06/29/89)   PAGE 8
```

```
      CALL FINDF(TS1,TTBS,FTBS,FWK,NCTS,NFS,NTS,KFS,KTS)
      CALL MULT(DS,FWK,FWK1,NQS,NFS,1,KQS,KFS)
      CALL AABB(1.0,GRAVS,1.0,FWK1,GFS1,NQS,1,KQS)
      CALL FINDTB(TS1,TTBS,NCTS,TS2,NFS,NTS,KFS,KTS)
      CALL FINDF(TS2,TTBS,FTBS,FWK,NCTS,NFS,NTS,KFS,KTS)
      CALL MULT(DS,FWK,FWK1,NQS,NFS,1,KQS,KFS)
      CALL AABB(1.0,GRAVS,1.0,FWK1,GFS2,NQS,1,KQS)
      TP1=TV(1)
      TPO=TP1
      CALL FINDF(TP1,TTBP,FTBP,FWK,NCTP,NFP,NTP,KFP,KTP)
      CALL MULT(DP,FWK,FWK2,NQP,NFP,1,KQP,KFP)
      CALL AABB(1.0,GRAVP,1.0,FWK2,GFP1,NQP,1,KQP)
      CALL FINDTB(TP1,TTBP,NCTP,TP2,NFP,NTP,KFP,KTP)
      CALL FINDF(TP2,TTBP,FTBP,FWK,NCTP,NFP,NTP,KFP,KTP)
      CALL MULT(DP,FWK,FWK2,NQP,NFP,1,KQP,KFP)
      CALL AABB(1.0,GRAVP,1.0,FWK2,GFP2,NQP,1,KQP)
      CALL GFB(1.,PHIS,FBO,GFSB,NB,NQS,1,KB,KB,KQS)
      CALL GFB(-1.,PHIP,FBO,GFPB,NB,NQP,1,KB,KB,KQP)
      NPRINT=1
      CALL DTOZ(TV,NMST,1,NTIM,1,KTIM)
      CALL ZZERO(NMFB,24,NTIM)
      CALL DASSMZ(FBO,24,1,1,1,NMFB,KB)
CCCCC
C     CALL ZREAD(NMJOE1)
C     CALL ZTRANS(NMJOE1,NMJOE2)
C     CALL ZDISA(NMJOE1,49,1,1,5,NMJOE2)
C     CALL ZTOD(NMJOE2,AA,NRAA,NCAA,50,5)
CCCCC
      NBCHK = 0
      DO 500 L=1,NTIM
      T=TV(L)
************************
      IF(T.GE.6.548)THEN
       IF(T.LT.6.549)THEN
        PRINT *,'SRB''S ARE IGNITED'
        PRINT *,'HOLD DOWN BOLTS ARE BLOWN'
        PRINT *,'START CHECKING THE PAD FORCES FOR LIFTOFF'
       ENDIF
C
C  START OF LIFTOFF CHECK FOR FX TO GO TO ZERO
C
       DO 3200 I = 1,NB,3
       IF(FBO(I).GE.O. .AND. NFLAG(I).EQ.O)THEN
```

```
            FBO(I) = O.
            FBO(I+1) = O.
            FBO(I+2) = O.
            II1 = I
            II2 = I + NB
            II3 = I + 2*NB
            NFLAG(I) = 1
C
C  ZERO OUT ROWS & COLUMNS OF C**-1 MATRIX AT
C  FBO DOF'S THAT HAVE POSITIVE X-DIR FORCES
C
            DO 3300 K=1,3
            DO 3000 J=1,NB3
               CIVMAT(II1,J) = O.
               CIVMAT(J,II1) = O.
               CIVMAT(II2,J) = O.
               CIVMAT(J,II2) = O.
               CIVMAT(II3,J) = O.
               CIVMAT(J,II3) = O.
 3000       CONTINUE
              II1 = II1 + 1
              II2 = II2 + 1
              II3 = II3 + 1
 3300       CONTINUE
            ENDIF
 3200    CONTINUE
         ENDIF
************************
  200                                                  NERROR=26
         IF(TSO .LT. TS1 .OR. T .LT. TSO) GO TO 999
         IF(T .GT. TS2) GO TO 210
         CALL SOLEQ(TS1,TS2,TSO,T,QSO,QDSO,NQS,GMS,GCS,GKS,GFSB,GFS1,
        *             GFS2,QS,QDS,QDDS)
         GO TO 240
  210    CALL SOLEQ(TS1,TS2,TSO,TS2,QSO,QDSO,NQS,GMS,GCS,GKS,GFSB,GFS1,
        *             GFS2,QS,QDS,QDDS)
         TS1=TS2
         TSO=TS2
         DO 220 I=1,NQS
         QSO(I)=QS(I)
         QDSO(I)=QDS(I)
  220    GFS1(I)=GFS2(I)
         CALL FINDIB(TS1,TTBS,NCTS,TS2,NFS,NTS,KFS,KTS)
```

```
      CALL FINDF(TS2,TTBS,FTBS,FWK,NCTS,NFS,NTS,KFS,KTS)
      CALL MULT(DS,FWK,FWK1,NQS,NFS,1,KQS,KFS)
      CALL AABB(1.0,GRAVS,1.0,FWK1,GFS2,NQS,1,KQS)
      GO TO 200
  240 CONTINUE
  250                                                 NERROR=27
      IF(TPO .LT. TP1 .OR. T .LT. TPO) GO TO 998
      IF(T .GT. TP2) GO TO 260
      CALL SOLEQ(TP1,TP2,TPO,T,QPO,QDPO,NQP,GMP,GCP,GKP,GFPB,GFP1,
     *           GFP2,QP,QDP,QDDP)
      GO TO 290
  260 CALL SOLEQ(TP1,TP2,TPO,TP2,QPO,QDPO,NQP,GMP,GCP,GKP,GFPB,GFP1,
     *           GFP2,QP,QDP,QDDP)
      TP1=TP2
      TPO=TP2
      DO 270 I=1,NQP
      QPO(I)=QP(I)
      QDPO(I)=QDP(I)
  270 GFP1(I)=GFP2(I)
      CALL FINDTB(TP1,TTBP,NCTP,TP2,NFP,NTP,KFP,KTP)
      CALL FINDF(TP2,TTBP,FTBP,FWK,NCTP,NFP,NTP,KFP,KTP)
      CALL MULT(DP,FWK,FWK2,NQP,NFP,1,KQP,KFP)
      CALL AABB(1.0,GRAVP,1.0,FWK2,GFP2,NQP,1,KQP)
      GO TO 250
  290 CONTINUE
      IF(L .EQ. 1) GO TO 295
      CALL BOUNDF(QS,QDS,QDDS,NQS,QP,QDP,QDDP,NQP,NB,FBO,
     *            PHIS,PHIP,CQSA,CQSB,CQSC,CQDSA,CQDSB,
     *            CQDSC,CQDDSA,CQDDSB,CQDDSC,CQPA,CQPB,
     *            CQPC,CQDPA,CQDPB,CQDPC,CQDDPA,CQDDPB,
     *            CQDDPC,CIVMAT,DT,KQS,KQP,KB,K3B)
      CALL GFB(1.,PHIS,FBO,GFSB,NB,NQS,1,KB,KB,KQS)
      CALL GFB(-1.,PHIP,FBO,GFPB,NB,NQP,1,KB,KB,KQP)
      CALL DASSMZ(FBO,24,1,1,L,NMFB,KB)
  295 CONTINUE
      CALL MULT(PHIS,QS,XBS,NB,NQS,1,KB,KQS)
      CALL MULT(PHIP,QP,XBP,NB,NQP,1,KB,KQP)
      CALL MULT(PHIS,QDS,XDBS,NB,NQS,1,KB,KQS)
      CALL MULT(PHIP,QDP,XDBP,NB,NQP,1,KB,KQP)
      CALL MULT(PHIS,QDDS,XDDBS,NB,NQS,1,KB,KQS)
      CALL MULT(PHIP,QDDP,XDDBP,NB,NQP,1,KB,KQP)
      NPRINT=NPRINT-1
      IF(NPRINT .GT. 0) GO TO 400
```

```
      WRITE(6,111) TV(L)
  111 FORMAT(/,5X,'T=',F8.4)
      WRITE(6,112) (XBS(I),I=1,NB)
  112 FORMAT(3X,'XBS=',5E17.8)
      WRITE(6,113) (XBP(I),I=1,NB)
  113 FORMAT(3X,'XBP=',5E17.8)
      WRITE(6,114) (XDBS(I),I=1,NB)
  114 FORMAT(2X,'XDBS=',5E17.8)
      WRITE(6,115) (XDBP(I),I=1,NB)
  115 FORMAT(2X,'XDBP=',5E17.8)
      WRITE(6,116) (XDDBS(I),I=1,NB)
  116 FORMAT(1X,'XDDBS=',5E17.8)
      WRITE(6,117) (XDDBP(I),I=1,NB)
  117 FORMAT(1X,'XDDBP=',5E17.8)
      WRITE(6,118) (FBO(I),I=1,NB)
  118 FORMAT(3X,'FBO=',5E17.8)
      NPRINT=NPRT
  400 CONTINUE
      TSO=T
      DO 300 I=1,NQS
      QSO(I)=QS(I)
  300 QDSO(I)=QDS(I)
      TPO=T
      DO 310 I=1,NQP
      QPO(I)=QP(I)
  310 QDPO(I)=QDP(I)
CCCCC
C     CALL MULT(AA,QS,SS,1,NQS,1,12,KQS)
C     SXDD(L) = SS(1,1)
C     SM(L) = FBO(2)
CCCCC
  500 CONTINUE
CCCCC
C     PRINT *,'TIP DISPLACEMENT'
C     PRINT *,'TIME(SEC)   XDD'
C     DO 96 II=1,NTIM
C       PRINT *, TV(II),SXDD(II)
C  96 CONTINUE
C     PRINT *,'TIME(SEC)         MOMENT(IN-LB)
C     DO 97 II=1,NTIM
C       PRINT *, TV(II),SM(II)
C  97 CONTINUE
CCCCC
```

```
LIFOFF     PAGE 9                  ON-CELPQRSUV                 04/16/90-13:49:08   CFT 1.16BFO(06/29/89)  PAGE 12

    411    345.       RETURN
    412    346.    999 CALL ZZBOMB('LIFOFF',NERROR)
    413    347.       END
LIFOFF               LOOP USES VECTOR LENGTH OF 8 AT SEQ. NO.          100, P=     361b
LIFOFF               LOOP USES VECTOR LENGTH OF 8 AT SEQ. NO.          114, P=     475b
LIFOFF               LOOP USES VECTOR LENGTH OF 8 AT SEQ. NO.          121, P=     551a
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          129, P=     610d
LIFOFF               LOOP USES VECTOR LENGTH OF 8 AT SEQ. NO.          136, P=     672d
LIFOFF               LOOP USES VECTOR LENGTH OF 8 AT SEQ. NO.          146, P=     744b
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          167, P=    1011c

   AT SEQUENCE NUMBER -    229.
PRNAME LIFOFF   COMMENT -     DEPENDENCY INVOLVING ARRAY "CIVMAT" IN SEQUENCE NUMBER 232
EXPLANATION:  AMBIGUOUS OR CONFLICTING SUBSCRIPTS

   AT SEQUENCE NUMBER -    229.
PRNAME LIFOFF   COMMENT -     DEPENDENCY INVOLVING ARRAY "CIVMAT" IN SEQUENCE NUMBER 234
EXPLANATION:  AMBIGUOUS OR CONFLICTING SUBSCRIPTS

   AT SEQUENCE NUMBER -    230.
PRNAME LIFOFF   COMMENT -     DEPENDENCY INVOLVING ARRAY "CIVMAT" IN SEQUENCE NUMBER 231
EXPLANATION:  AMBIGUOUS OR CONFLICTING SUBSCRIPTS
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          228, P=    1512a
LIFOFF             CONDITIONAL VECTOR LOOP BEGINS AT SEQ. NO.          228, P=    1512a
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          254, P=    1613a
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          274, P=    1757d
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          321, P=    2536c
LIFOFF                         VECTOR LOOP BEGINS AT SEQ. NO.          325, P=    2557a
```

```
      SUBROUTINE COECAL(GMS,GCS,GKS,NQS,GMP,GCP,GKP,NQP,NB,DT,
     *                  PHIS,PHIP,CQSA,CQSB,CQSC,CQDSA,CQDSB,
     *                  CQDSC,CQDDSA,CQDDSB,CQDDSC,CQPA,CQPB,
     *                  CQPC,CQDPA,CQDPB,CQDPC,CQDDPA,CQDDPB,
     *                  CQDDPC,COEMAT,KQS,KQP,KB,K3B)
      DIMENSION GMS(1),GCS(1),GKS(1),GMP(1),GCP(1),GKP(1),
     *          PHIS(KB,1),PHIP(KB,1),CQSA(KQS,1),CQSB(KQS,1),
     *          CQSC(KQS,1),CQDSA(KQS,1),CQDSB(KQS,1),CQDSC(KQS,1),
     *          CQDDSA(KQS,1),CQDDSB(KQS,1),CQDDSC(KQS,1),
     *          CQPA(KQP,1),CQPB(KQP,1),CQPC(KQP,1),CQDPA(KQP,1),
     *          CQDPB(KQP,1),CQDPC(KQP,1),CQDDPA(KQP,1),CQDDPB(KQP,1),
     *          CQDDPC(KQP,1),COEMAT(K3B,1)
      COMMON /LOWK1/ QO(900),QDO(900),VNUL(900),GFV(900),Q(900),
     *               QD(900),QDD(900)
                                                    NERROR=1
      IF(KQS .GT. 900 .OR. KQP .GT. 900) GO TO 999
                                                    NERROR=2
      IF(NQS .GT. KQS) GO TO 999
                                                    NERROR=3
      IF(NQP .GT. KQP) GO TO 999
                                                    NERROR=4
      IF(NB .GT. KB) GO TO 999
      DO 10 I=1,NQS
      QO(I)=0.
      QDO(I)=0.
   10 VNUL(I)=0.
      DO 50 I=1,NB
      CALL DISA(PHIS,I,1,GFV,NB,NQS,1,NQS,KB,1)
      CALL SOLPR3(0.,QO,QDO,DT,NQS,GMS,GCS,GKS,VNUL,GFV,VNUL,VNUL,
     *            Q,QD,QDD)
      DO 20 J=1,NQS
      CQSA(J,I)=Q(J)
      CQDSA(J,I)=QD(J)
   20 CQDDSA(J,I)=QDD(J)
      CALL SOLPR3(0.,QO,QDO,DT,NQS,GMS,GCS,GKS,VNUL,VNUL,GFV,VNUL,
     *            Q,QD,QDD)
      DO 30 J=1,NQS
      CQSB(J,I)=Q(J)
      CQDSB(J,I)=QD(J)
   30 CQDDSB(J,I)=QDD(J)
      CALL SOLPR3(0.,QO,QDO,DT,NQS,GMS,GCS,GKS,VNUL,VNUL,VNUL,GFV,
     *            Q,QD,QDD)
      DO 40 J=1,NQS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      CQSC(J,I)=Q(J)
      CQDSC(J,I)=QD(J)
   40 CQDDSC(J,I)=QDD(J)
   50 CONTINUE
      DO 60 I=1,NQP
      QO(I)=0.
      QDO(I)=0.
   60 VNUL(I)=0.
      DO 100 I=1,NB
      CALL DISA(PHIP,I,1,GFV,NB,NQP,1,NQP,KB,1)
      CALL ALPHAA(-1.,GFV,GFV,1,NQP,1)
      CALL SOLPR3(0.,QO,QDO,DT,NQP,GMP,GCP,GKP,VNUL,GFV,VNUL,VNUL,
     *            Q,QD,QDD)
      DO 70 J=1,NQP
      CQPA(J,I)=Q(J)
      CQDPA(J,I)=QD(J)
   70 CQDDPA(J,I)=QDD(J)
      CALL SOLPR3(0.,QO,QDO,DT,NQP,GMP,GCP,GKP,VNUL,VNUL,GFV,VNUL,
     *            Q,QD,QDD)
      DO 80 J=1,NQP
      CQPB(J,I)=Q(J)
      CQDPB(J,I)=QD(J)
   80 CQDDPB(J,I)=QDD(J)
      CALL SOLPR3(0.,QO,QDO,DT,NQP,GMP,GCP,GKP,VNUL,VNUL,VNUL,GFV,
     *            Q,QD,QDD)
      DO 90 J=1,NQP
      CQPC(J,I)=Q(J)
      CQDPC(J,I)=QD(J)
   90 CQDDPC(J,I)=QDD(J)
  100 CONTINUE
      N=3*NB
      DO 110 I=1,N
      DO 110 J=1,N
  110 COEMAT(I,J)=0.
      DO 200 I=1,NB
      I2=I+NB
      I3=I+2*NB
      DO 200 J=1,NB
      J2=J+NB
      J3=J+2*NB
      DO 120 K=1,NQP
      COEMAT(I,J)=COEMAT(I,J)+PHIP(I,K)*CQPA(K,J)
      COEMAT(I,J2)=COEMAT(I,J2)+PHIP(I,K)*CQPB(K,J)
```

```
COECAL      PAGE 3                 DN=CELPQRSUV                 04/16/90-13:49:08   CFT 1.16BFO(06/29/89)  PAGE 16
                                                                                                            10.
    500    87.          COEMAT(I,J3)=COEMAT(I,J3)+PHIP(I,K)*CQPC(K,J)
    501    88.          COEMAT(I2,J)=COEMAT(I2,J)+PHIP(I,K)*CQDPA(K,J)
    502    89.          COEMAT(I2,J2)=COEMAT(I2,J2)+PHIP(I,K)*CQDPB(K,J)
    503    90.          COEMAT(I2,J3)=COEMAT(I2,J3)+PHIP(I,K)*CQDPC(K,J)
    504    91.          COEMAT(I3,J)=COEMAT(I3,J)+PHIP(I,K)*CQDDPA(K,J)
    505    92.          COEMAT(I3,J2)=COEMAT(I3,J2)+PHIP(I,K)*CQDDPB(K,J)
    506    93.          COEMAT(I3,J3)=COEMAT(I3,J3)+PHIP(I,K)*CQDDPC(K,J)
    507    94.      120 CONTINUE
    508    95.          DO 130 K=1,NQS
    509    96.          COEMAT(I,J)=COEMAT(I,J)-PHIS(I,K)*CQSA(K,J)
    510    97.          COEMAT(I,J2)=COEMAT(I,J2)-PHIS(I,K)*CQSB(K,J)
    511    98.          COEMAT(I,J3)=COEMAT(I,J3)-PHIS(I,K)*CQSC(K,J)
    512    99.          COEMAT(I2,J)=COEMAT(I2,J)-PHIS(I,K)*CQDSA(K,J)
    513   100.          COEMAT(I2,J2)=COEMAT(I2,J2)-PHIS(I,K)*CQDSB(K,J)
    514   101.          COEMAT(I2,J3)=COEMAT(I2,J3)-PHIS(I,K)*CQDSC(K,J)
    515   102.          COEMAT(I3,J)=COEMAT(I3,J)-PHIS(I,K)*CQDDSA(K,J)
    516   103.          COEMAT(I3,J2)=COEMAT(I3,J2)-PHIS(I,K)*CQDDSB(K,J)
    517   104.          COEMAT(I3,J3)=COEMAT(I3,J3)-PHIS(I,K)*CQDDSC(K,J)
    518   105.      130 CONTINUE
    519   106.      200 CONTINUE
    520   107.          RETURN
    521   108.      999 CALL ZZBOMB('COECAL',NERROR)
    522   109.          END
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          23, P=      5b
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          31, P=     54b
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          37, P=    131c
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          43, P=    176a
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          48, P=    245d
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          57, P=    272a
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          63, P=    355c
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          69, P=    422b
COECAL                          VECTOR LOOP BEGINS AT SEQ. NO.          76, P=    477d

   AT SEQUENCE NUMBER -     85.
PRNAME COECAL    COMMENT -      DEPENDENCY INVOLVING ARRAY "COEMAT" IN SEQUENCE NUMBER 86
EXPLANATION:  NO CII WAS FOUND IN ARRAY REFERENCES

   AT SEQUENCE NUMBER -     85.
PRNAME COECAL    COMMENT -      DEPENDENCY INVOLVING ARRAY "COEMAT" IN SEQUENCE NUMBER 86
EXPLANATION:  NO CII WAS FOUND IN ARRAY REFERENCES

   AT SEQUENCE NUMBER -     85.
PRNAME COECAL    COMMENT -      DEPENDENCY INVOLVING ARRAY "COEMAT" IN SEQUENCE NUMBER 87
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
        PAGE              ON=CELPORSUV              04/16/90-13:49:08   CFT 1 16BFO(06/29/89)  PAGE 27

  667     1.          SUBROUTINE GFB(ALPHA,A,B,Z,NRA,NCA,NCB,KRA,KRB,KRZ)
  668     2.          DIMENSION A(KRA,1),B(KRB,1),Z(KRZ,1)
  669     3.          DO 20 I=1,NCA
  670     4.          DO 20 J=1,NCB
  671     5.          Z(I,J)=0.
  672     6.          DO 10 K=1,NRA
  673     7.       10 Z(I,J)=Z(I,J)+A(K,I)*B(K,J)
  674     8.       20 Z(I,J)=ALPHA*Z(I,J)
  675     9.          RETURN
  676    10.          END
GFB                        VECTOR LOOP BEGINS AT SEQ. NO.          6. P=       20b
```

PAGE ON=CELPORSUV 04/16/90-13:49:08 CFT 1.16BFO(06/29/89) PAGE 25

```
      SUBROUTINE SOLEQ(T1,T2,TO,T,QO,QDO,N,GM,GC,GK,GFB,GF1,GF2,
     *                 Q,QD,QDD)
      DIMENSION QO(1),QDO(1),GM(1),GC(1),GK(1),GFB(1),GF1(1),
     *          GF2(1),Q(1),QD(1),QDD(1)
      DT=T-TO
      DT2=DT*DT
      DT3=DT2*DT
      DO 500 I=1,N
      R1=(GF2(I)-GF1(I))/(T2-T1)
      RO=GFB(I)+GF1(I)+R1*(TO-T1)
      IF(GK(I) .LE. O.) GO TO 400
      A1=GC(I)/(2.*GM(I))
      A2=SQRT(GK(I)/GM(I)-A1*A1)
      H1=R1/GK(I)
      HO=(RO-H1*GC(I))/GK(I)
      B1=QO(I)-HO
      B2=(QDO(I)-H1+A1*QO(I)-A1*HO)/A2
      C1=-A1*B1+B2*A2
      C2=-A1*B2-B1*A2
      D1=-A1*C1+C2*A2
      D2=-A1*C2-C1*A2
      EA1T=EXP(-A1*DT)
      CA2T=COS(A2*DT)
      SA2T=SIN(A2*DT)
      Q(I)=EA1T*(B1*CA2T+B2*SA2T)+HO+H1*DT
      QD(I)=EA1T*(C1*CA2T+C2*SA2T)+H1
      QDD(I)=EA1T*(D1*CA2T+D2*SA2T)
      GO TO 500
  400 CONTINUE
      Q(I)=QO(I)+QDO(I)*DT+(RO*DT2*.5+R1*DT3/6.)/GM(I)
      QD(I)=QDO(I)+(RO*DT+R1*DT2*.5)/GM(I)
      QDD(I)=(RO+R1*DT)/GM(I)
  500 CONTINUE
      RETURN
      END
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      SUBROUTINE SOLPR3(TO,QO,QDO,T,N,GM,GC,GK,GFO,GF1,GF2,GF3,
     *                  Q,QD,QDD)
      DIMENSION QO(1),QDO(1),GM(1),GC(1),GK(1),GFO(1),GF1(1),
     *          GF2(1),GF3(1),Q(1),QD(1),QDD(1)
      DT=T-TO
      DT2=DT*DT
      DT3=DT2*DT
      DT4=DT2*DT2
      DT5=DT3*DT2
      DO 500 I=1,N
      IF(GK(I) .LE. O.) GO TO 400
      A1=GC(I)/(2.*GM(I))
      A2=SQRT(GK(I)/GM(I)-A1*A1)
      H3=GF3(I)/GK(I)
      H2=(GF2(I)-3.*GC(I)*H3)/GK(I)
      H1=(GF1(I)-2.*GC(I)*H2-6.*GM(I)*H3)/GK(I)
      HO=(GFO(I)-H1*GC(I)-2.*GM(I)*H2)/GK(I)
      B1=QO(I)-HO
      B2=(QDO(I)-H1+A1*QO(I)-A1*HO)/A2
      C1=-A1*B1+B2*A2
      C2=-A1*B2-B1*A2
      D1=-A1*C1+C2*A2
      D2=-A1*C2-C1*A2
      EA1T=EXP(-A1*DT)
      CA2T=COS(A2*DT)
      SA2T=SIN(A2*DT)
      Q(I)=EA1T*(B1*CA2T+B2*SA2T)+HO+H1*DT+H2*DT2+H3*DT3
      QD(I)=EA1T*(C1*CA2T+C2*SA2T)+H1+2.*H2*DT+3.*H3*DT2
      QDD(I)=EA1T*(D1*CA2T+D2*SA2T)+2*H2+6.*H3*DT
      GO TO 500
  400 CONTINUE
      Q(I)=QO(I)+QDO(I)*DT+(GFO(I)*DT2*.5+GF1(I)*DT3/6.+
     *     GF2(I)*DT4/12.+GF3(I)*DT5/20.)/GM(I)
      QD(I)=QDO(I)+(GFO(I)*DT+GF1(I)*DT2*.5+GF2(I)*DT3/3.+
     *      GF3(I)*DT4*.25)/GM(I)
      QDD(I)=(GFO(I)+GF1(I)*DT+GF2(I)*DT2+GF3(I)*DT3)/GM(I)
  500 CONTINUE
      RETURN
      END
```

SOLPR3 VECTOR LOOP BEGINS AT SEQ. NO. 10. P= 5b

```
      SUBROUTINE BOUNDF(QS,QDS,QDDS,NQS,QP,QDP,QDDP,NQP,NB,FBO,
     *                  PHIS,PHIP,COSA,COSB,COSC,CODSA,CODSB,
     *                  CQDSC,CQDDSA,CQDDSB,CQDDSC,COPA,COPB,
     *                  CQPC,CQDPA,CQDPB,CQDPC,CQDDPA,CQDDPB,
     *                  CQDDPC,CIVMAT,DT,KQS,KQP,KB,K3B)
      DIMENSION QS(1),QDS(1),QDDS(1),QP(1),QDP(1),QDDP(1),FBO(1),
     *          PHIS(KB,1),PHIP(KB,1),COSA(KQS,1),COSB(KQS,1),
     *          COSC(KQS,1),CQDSA(KQS,1),CQDSB(KQS,1),CQDSC(KQS,1),
     *          CQDDSA(KQS,1),CQDDSB(KQS,1),CQDDSC(KQS,1),
     *          CQPA(KQP,1),CQPB(KQP,1),CQPC(KQP,1),CQDPA(KQP,1),
     *          CQDPB(KQP,1),CQDPC(KQP,1),CQDDPA(KQP,1),CQDDPB(KQP,1),
     *          CQDDPC(KQP,1),CIVMAT(K3B,1)
      COMMON /LOWK1/ DEL(72),ABC(72),DUM(6156)
      DT2=DT*DT
      DT3=DT2*DT
      N3B=NB*3
                                                     NERROR=1
      IF(N3B .GT. 72) GO TO 999
      DO 30 I=1,NB
      I1=I
      I2=I1+NB
      I3=I2+NB
      DEL(I1)=0.
      DEL(I2)=0.
      DEL(I3)=0.
      DO 10 J=1,NQS
      DEL(I1)=DEL(I1)+PHIS(I,J)*QS(J)
      DEL(I2)=DEL(I2)+PHIS(I,J)*QDS(J)
   10 DEL(I3)=DEL(I3)+PHIS(I,J)*QDDS(J)
      DO 20 J=1,NQP
      DEL(I1)=DEL(I1)-PHIP(I,J)*QP(J)
      DEL(I2)=DEL(I2)-PHIP(I,J)*QDP(J)
   20 DEL(I3)=DEL(I3)-PHIP(I,J)*QDDP(J)
   30 CONTINUE
      DO 40 I=1,N3B
      ABC(I)=0.
      DO 40 J=1,N3B
   40 ABC(I)=ABC(I)+CIVMAT(I,J)*DEL(J)
      DO 50 I=1,NB
      I1=I
      I2=I1+NB
      I3=I2+NB
   50 FBO(I)=FBO(I)+ABC(I1)*DT+ABC(I2)*DT2+ABC(I3)*DT3
```

```
566   44.         DO 60 I=1,NQS
567   45.         DO 60 J=1,NB
568   46.         J1=J
569   47.         J2=J1+NB
570   48.         J3=J2+NB
571   49.         QS(I)=QS(I)+CQSA(I,J)*ABC(J1)+CQSB(I,J)*ABC(J2)+
572   50.        *      CQSC(I,J)*ABC(J3)
573   51.         QDS(I)=QDS(I)+CQDSA(I,J)*ABC(J1)+CQDSB(I,J)*ABC(J2)+
574   52.        *.     CQDSC(I,J)*ABC(J3)
575   53.         QDDS(I)=QDDS(I)+CQDDSA(I,J)*ABC(J1)+CQDDSB(I,J)*ABC(J2)+
576   54.        *      CQDDSC(I,J)*ABC(J3)
577   55.      60 CONTINUE
578   56.         DO 70 I=1,NQP
579   57.         DO 70 J=1,NB
580   58.         J1=J
581   59.         J2=J1+NB
582   60.         J3=J2+NB
583   61.         QP(I)=QP(I)+CQPA(I,J)*ABC(J1)+CQPB(I,J)*ABC(J2)+
584   62.        *      CQPC(I,J)*ABC(J3)
585   63.         QDP(I)=QDP(I)+CQDPA(I,J)*ABC(J1)+CQDPB(I,J)*ABC(J2)+
586   64.        *      CQDPC(I,J)*ABC(J3)
587   65.         QDDP(I)=QDDP(I)+CQDDPA(I,J)*ABC(J1)+CQDDPB(I,J)*ABC(J2)+
588   66.        *      CQDDPC(I,J)*ABC(J3)
56)   67.      70 CONTINUE
590   68.         RETURN
591   69.     999 CALL ZZBOMB('BOUNDF',NERROR)
592   70.         END

   AT SEQUENCE NUMBER -     27.
PRNAME BOUNDF   COMMENT -      DEPENDENCY INVOLVING ARRAY "DEL" IN SEQUENCE NUMBER 28
EXPLANATION:  NO CII WAS FOUND IN ARRAY REFERENCES

   AT SEQUENCE NUMBER -     27.
RRNAME BOUNDF   COMMENT -      DEPENDENCY INVOLVING ARRAY "DEL" IN SEQUENCE NUMBER 28
EXPLANATION:  NO CII WAS FOUND IN ARRAY REFERENCES

   AT SEQUENCE NUMBER -     27.
PRNAME BOUNDF   COMMENT -      DEPENDENCY INVOLVING ARRAY "DEL" IN SEQUENCE NUMBEP 29
EXPLANATION:  NO CII WAS FOUND IN ARRAY REFERENCES

   AT SEQUENCE NUMBER -     31.
PRNAME BOUNDF   COMMENT -      DEPENDENCY INVOLVING ARRAY "DEL" IN SEQUENCE NUMBER 32
EXPLANATION:  NO CII WAS FOUND IN ARRAY REFERENCES
```

APPROVAL

A TRANSIENT RESPONSE ANALYSIS OF THE SPACE SHUTTLE VEHICLE DURING LIFTOFF

By J.A. Brunty

The information in this report has been reviewed for technical content. Review of any information concerning Department of Defense or nuclear energy activities or programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

JAMES C. BLAIR
Director, Structures and Dynamics Laboratory

☆ U.S. GOVERNMENT PRINTING OFFICE 1990—731—061/20185