AD-A179 772 DAMAGE DIAGNOSIS FOR ELASTO-PLASTIC STRUCTURES(U) NEW MEXICO UNIV ALBUQUERQUE DEPT OF MECHANICAL ENGINEERING F D JU ET AL. DEC 86 ME-138(86)AFOSR-993-2 1/3

UNCLASSIFIED AFOSR-TR-87-0413 AFOSR-85-0085 F/G 20/11 NL


1.0
2.8
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

DTIC FILE COPY

AFOSR-TR- 87-0413

(2)

AD-A179 772

THE UNIVERSITY OF NEW MEXICO
COLLEGE OF ENGINEERING

...release;
distribution unlimited.

...FICE OF SCIENTIFIC RESEARCH (AFSC)
...ITTAL TO DTIC
...has been reviewed and is
...lic release IAW AFR 190-12.
...unlimited.
Chief, Technical Information Division

BUREAU OF ENGINEERING RESEARCH

DAMAGE DIAGNOSIS FOR ELASTO-PLASTIC STRUCTURES

BY

Fashin C. Chang

and

Frederick D. Ju

DTIC
ELECTE
APR 29 1987
S D
D

Technical Report No. ME-138(86)AFOSR-993-2

Work supported by Air Force Office of Scientific Research

Grant No. AFOSR-85-0085A

87 4 23 127

8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Office of Scientific Research	8b. OFFICE SYMBOL (If applicable) AFOSR/NA	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER AFOSR-85-0085

8c. ADDRESS (City, State and ZIP Code) Bld 410 Bolling AFB, DC 20332	10. SOURCE OF FUNDING NOS.			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
11. TITLE (Include Security Classification) Damage Diagnosis for Elasto-Plastic Structures	61102F	2302	C2	

12. PERSONAL AUTHOR(S)
Ju, Frederick D. and Fashin C. Chang

13a. TYPE OF REPORT Technical	13b. TIME COVERED FROM 01/86 TO 12/86	14. DATE OF REPORT (Yr., Mo., Day) December 1986	15. PAGE COUNT 211

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB. GR.	elastomers, plastics, Random Excitation, Random Stiffness, Nonlinear Elasto-Plastic, Damage Diagnosis
11			

19. Abstract (continued)

function for a system with random stiffness is significantly greater than the response autocovariance function for a system with deterministic stiffness. The probability of structural damage, or the structural reliability, is then estimated by the upper bound of the cumulative energy dissipation. It is noted that the damage diagnosis of structure is based upon the successful mathematical modeling of the structural system. The formalism of present report, therefore, enables us to assess the damage for a generic class of MDF non-linear system with Prandtl-Reuss material. In addition, the present investigation allows us the ready adaptation to finite element analysis.

DAMAGE DIAGNOSIS
FOR
ELASTO-PLASTIC STRUCTURES

by

Fashin C. Chang

and

Frederick D. Ju

Mechanical Engineering Department
The University of New Mexico
Albuquerque, NM 87131

Technical Report

ME-138(86)AFOSR-993-2

Work supported by Air Force Office of Scientific Research

Grant No. AFOSR-85-0085A

ABSTRACT

The present report established a probabilistic characterization of a damaged structure, which is modeled as a nonlinear multi-degree-of-freedom (MDF) system. The random excitation may be either stationary or nonstationary. The stiffness matrix is nonlinear to simulate the elasto-plastic behavior of a damaged structure. The stiffness matrix is also random to characterize the material and environmental variations. The governing stochastic differential equation is resolved into one for the mean response and another for its random component. The responses, their statistical moments and cross-moments are solved with discrete-time recurrence formulations. The omission of the higher order terms in obtaining the response statistics is admissible if the stiffness does not behave extremely random. Both errors in the mean and variance responses, which arise from neglecting the higher order terms, are of small order. The numerical computation shows two important results. (1) The response autocovariance function for the non-linear system has greater magnitude than the response autocovariance function for linear system. (2) The response autocovariance function for a system with random stiffness is significantly greater than the response autocovariance function for a system with deterministic stiffness. The probability of structural damage, or the structural reliability, is then estimated by the upper bound of the cumulative energy dissipation. It is noted that the damage diagnosis of structure is based upon the successful mathematical modeling of the structural system. The formalism of the present report, therefore, enables us to assess the damage for a generic class of MDF non-linear system with Prandtl-Reuss material. In addition, the present investigation allows us the ready adaptation to finite element analysis.

ACKNOWLEDGEMENT

The authors wish to acknowledge the support of the Air Force Office of Scientific Research in this study. The work was performed under AFOSR Contract No. 85-0085A, Col. Lawrence D. Hokanson was and Dr. Spencer Wu is the program manager.

TABLE OF CONTENTS

Page

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$[A_1]$, $[A_2]$, $[A_3]$	matrices, combinations of characteristic matrices $[m]$, $[c]$ and $[\lambda]$ -- eq. (2-15)
b	width
$[c]$	viscous matrix
c	position of the beam outer fibre
E, I	Young's Modulus, Area moment of Inertia
$E[\cdot]$	expected value of $(\cdot)$
E_d	dissipated energy due to damping
E_s	dissipated energy in a small element due to plastic deformation
E_v	dissipated energy due to plastic deformation in a system
E_t	total dissipated energy
$\{f\}$	random excitation
$\{F\}$	random component of excitation $\{f\}$
$[k]$, $[k_{rs}]$	random stiffness matrix of a linear system
$[K]$, $[K_{rs}]$	random component of stiffness
L	length of a beam component
$[m]$	mass matrix
M	resisting moment in a beam
$\{R(\mu)\}$	restoring force in non-linear system
t	time
$\{z\}$	random response (displacement)
$\{Z\}$	random component of $\{z\}$
α	random variable, stiffness
α, γ	Rayleigh damping coefficients
β	coefficient of variation (Std. deviation/mean)

ε	strain
ε_o	permanent set
$\{\phi\}$	expected value of $\{f\}$
$[\lambda]$	expected value of $[k]$
$\{\mu\}$	expected value of $\{z\}$
ω_n, ω_d	natural and damped frequencies
$[\ ]^{-1}$	inverse of $[\ ]$
$[\ ]^T$	transpose of $[\ ]$
$(\dot{\ })$	$d(\ )/dt$

Chapter 1

INTRODUCTION

1.1 STATEMENT OF PROBLEM

The purpose of the investigation is to assess the structural damage due to a random strong excitation. In the process, the research establishes a probabilistic characterization of the inelastic multi-degree-of-freedom (MDF) system and its response due to random excitation. The stiffness matrix is nonlinear to simulate the elasto-plastic behavior of a damaged structure. The stiffness matrix is also random to characterize the material and environmental variations. Based on the statistical results, the damage and the reliability of the structure can be assessed. This technical report constitutes as part of the theory of damage diagnosis.

The reliability of a system is defined as the probability that it will exhibit behavior that satisfies certain criteria over some preestablished time duration. The criteria may associate with the survival of the structure without catastrophic failure or the functioning of the structure to meet the design purpose even with cumulated amount of damage. The reliability of a structural system, however, is not known when the design, which is based on design codes, has been completed. The reasons are (1) structural design for dynamic environments is established using iterative procedures, (2) the dynamic environments are in random nature, (3) the uncertainty of the structural system. They are stated as follows.

(1) Structural design for dynamic environments is established using iterative procedures: The design procedures of a structure for dynamic environments can be summarized as below.

(a) The limitations on the design are defined.

(b) A preliminary concept on a safe design is proposed.

(c) The preliminary design is analyzed, and a model is tested, numerically, or in the laboratory, if possible.

(d) The response is assessed.

(e) The design is satisfactory if the response fits the requirements. Otherwise, the design must be modified, repeating steps (b)-(e) until a satisfactory response is achieved.

It can be seen that the reliability of the structural system is not known unless a proper theory is developed to assess the response. The theory discussed here includes two purposes, namely, (i) to predict the damage of the structural system, (ii) to assess the damaged structures for existing structures. Such theory is known as the damage diagnosis in structures. Therefore, it follows that the damage diagnosis not only can help us in determining the assessment of the structural response, but also enables us to establish the reliability for the structure under consideration. On the other hand, without the theory of damage diagnosis, the structural engineer has no way to know the reliability of the structure he has designed.

(2) The dynamic environments are in random nature: In the design process, it is possible to use deterministic analysis for some limited applications. This is true in situations where the structure under consideration is loaded only by known deterministic forces and where the parameters of the structure, its foundation, and the applied loads are nonrandom (or display

very little random variability). However, under most circumstances, especially for dynamic loadings, it is desirable to execute analyses that take into account random variability in the structural load and, possibly, the parameters of the structure under consideration. When structural loads are random, the computation and assessment of structural response in step (c), (d), of the procedure for structural design, as mentioned in (1), must account for this. Thus, a probabilistic measure, or an average measure of structural response, must be computed and used in the response assessment. The theory of damage diagnosis, hence, must take into consideration the probabilistic response characterization of structure with random loads. Examples of random dynamic loads are earthquake, blast, wind, ocean waves, aerodynamic turbulence, vibration induced in transportation by road roughness, dynamic load for aircraft during landing, etc. In order to execute the analysis of structural response using a probabilistic approach, the statistical properties of these loads must be known. These statistics are essential for the theory of damage diagnosis. According to the statistics of the random loads and the theory of damage diagnosis established, the reliability measures can be assessed based on the probabilistic results.

(3) The uncertainty of the structural system: Randomness may also occur in other important areas in the structural dynamic system. Particularly, the parameters that characterize structural behavior may be random. For example, in linear problem, the parameters that define stiffness, damping and mass may be random. In other words, the parameters that define modal frequencies and damping factors may be random. Further, in the elasto-plastic problems, the parameters that define the nonlinear plastic characteristics of structural behavior are definitely random. Structural parameter randomness arises from two sources. The first is randomness in material properties. The second is randomness in structural geometry and assembly.

Randomness in structural characteristics may be manifested in various ways. On a very general level, all material and geometric parameters may be considered as random process and many of these random processes may be correlated. On a simpler level, which is investigated for current research year, randomness may be limited to the variation in one or a few parameters, and these parameters are considered as random variables rather than random processes. The elasto-plastic material belongs to this category. However, for frictional material, such as concrete, the variation in stiffness may be time dependent since the material will show stiffness and strength degradation whose statistics are varied by time. In that case, a random process which is used to represent the randomness in structural characteristics is necessary. This will be investigated for next research year.

According to these statements as discussed above, the reliability of the structural system must be carried out by using the theory of damage diagnosis in order to obtain a satisfactory design. Further, the damage diagnosis is based on the successful modeling of the damageable structural system. Therefore, it is obvious that the theory of structural damage diagnosis is important because it can not only accurately describe and analyze data, but also predict behavior under conditions not covered by data. In particular, we use the theory of damage diagnosis in engineering first to assist in the design of components and devices to achieve a specified level of safety and reliability at a minimum cost, and second to assist in the optimum management of these structural systems in service.

In developing a diagnostic theory for damaged structures, the investigations have studied various structural models. The first model was a hyper-linear model, in which the nonlinear hysteretic structural behavior is modeled by using the method of higher-order equivalent linearization. By using this approach, the non-linear hysteresis of the restoring force can be reproduced. However, it can not address the structural phenomenon of permanent set. In order to overcome this, an elasto-plastic model that is able to take into account the permanent set is proposed for the present investigation. The objective of this report, therefore, is to establish a damage theory that enables us to assess the probabilistic characterization of a damaged structure, which is modeled as a nonlinear elasto-plastic MDF system.

1.2 LITERATURE REVIEW

During the past years, the research effort sponsored by the Air Force Office of Scientific Research has made significant progress toward developing a consistent structural theory in damage diagnosis and reliability assessment. Numerous papers have been published by Ju et al. Among these, [1,2] studied the diagnosis of symmetric edge cracks in simple structures, where the cracks are modeled as a fracture hinge. The fracture hinge model has recently been verified in an experimental study [3] to be reported later. The fracture damage diagnosis for structure has been investigated [4], wherein two nondestructive methods of damage diagnosis in simple and complex structures are studied. The damage assessment for nonlinear system which was modeled as higher-order equivalent linearization system was investigated [5-7].

There are also several areas of research interest related to the analysis of randomly excited system. The general problem of probabilistic analysis of structural system is treated in many text books. Among these are the book by Lin [8], Soong [9], Crandall and Mark [10], Newland [11], Crandall [12,13] and Clough and Penzien [14]. These books treat the problem of computation of response moments for structural system. The first passage problems, the fatigue problems and the Markov character of the response of systems excited by white noise are also presented in these books. In addition, there are numerous papers published discussed the statistical properties of the structural response [15-30], to name only a few. Among these, Bogdanoff [15,16] has developed a stochastic cumulative damage model that possesses the major sources of variability in life prediction as an inherent part of its structure. Similar works, however, by different approach have been investigated by Lin et al [17,18,19] where the cracking propagation is modeled as Fokker-Planck equation. Wiggins and Moran [20] developed a means for grading existing buildings. A more mathematical quantification of damage in structures has been used by Ang and Wen [21].

Several other first passage problems are solved. For example, [22-29] find the first passage probability, and peak response probability distribution for SDF system. Ang [30] approximately computes the first passage probability for MDF system.

There are some papers in the literature address the problem of damage assessment. Yao [31] examined various definitions of structural damage and reviewed available methods for damage assessment. Park et al [32] established a model for evaluating structural damage in reinforced concrete structures under earthquake ground motion.

Chapter 2

LINEAR SYSTEM

2.1 INTRODUCTION

The governing differential equation of motion for a structural framework modeled as a discrete multi-degree-of-freedom (MDF) system can be written:

$$[m]\{\ddot{z}\} + [c]\{\dot{z}\} + [k]\{z\} = \{f\}, \tag{2-1}$$

where [m], [c], [k] are the NxN mass, damping, and stiffness matrices, {f} is the external load vector; $\{z\}$, $\{\dot{z}\}$, $\{\ddot{z}\}$ are the displacement, velocity, and acceleration vectors of the system. The differential equation governing motion is random when some terms of the above equation are random variables or random processes.

In many applications, the external load vector, {f}, is assumed to be random. Less frequently, the coefficients of the above equation are also assumed to behave randomly. In this latter application, the mass, damping, and/or stiffness are not considered to be deterministic valued, but rather, they are considered to be random and to possess specific probability distributions. All the coefficients in Eqaution (2-1) play an important role in determining the dynamic response of a structure. However, if coefficients are to be considered as random, it is certainly clear that in practical applications to building structures the mass will have relatively low random variation and the stiffness and damping will have relatively high random variation. Because slight nonlinearities in material behavior can occur in practice even when the fundamental response is linear, the variation in stiffness is especially important. Therefore, in the present application only the stiffness matrix is considered to be a random coefficient. The other terms are considered to be deterministic. Specifically, the coefficient [k] is assumed to be a matrix of random variables, and the forcing function {f} is considered to be a vector stochastic process.

It is important to note that in the analysis of a random differential equation the initial conditions can strongly affect the results because the probability distribution of the response for a dynamic system is related to the initial conditions directly. However, in most structural dynamic systems the initial conditions can be set to zero because the structures start from rest in most situations of interest. Hence, the initial conditions of Equation (2-1) are given as zero and are deterministic.

Finally, note that Equation (2-1) can become nonlinear when the structural material responds in the plastic range. The nonlinear problem will be considered in the next chapter.

2.2 MATHEMATICL MODEL

As stated in Section 2.1, only {f} and [k] are treated as random in Equation (2-1). The response, {z}, then forms a random process. The quantities {f}, [k] and {z} can be decompo_ed as follows:

$$\{f\} = \{\phi\} + \{F\}, \quad [k] = [\lambda] + [K], \quad \{z\} = \{\mu\} + \{Z\}, \qquad (2\text{-}2)$$

where $\{\phi\}$, $[\lambda]$, and $\{\mu\}$ are the mean values of $\{f\}$, $[k]$, and $\{z\}$; $\{F\}$, $[K]$, and $\{Z\}$ represent the random components of $\{f\}$, $[k]$, and $\{z\}$. Taking the expected values of Equation (2-2) results in

$$
\begin{aligned}
E[\{f\}] &= \{\phi\} + E[\{F\}], \\
E[[k]] &= [\lambda] + E[[K]], \\
E[\{z\}] &= \{\mu\} + E[\{Z\}]. \qquad (2\text{-}3)
\end{aligned}
$$

Hence, the random terms $\{F\}$, $[K]$ and $\{Z\}$ must all have zero means, because $E[\{f\}] = \{\phi\}$, $E[[k]] = [\lambda]$, and $E[\{z\}] = \{\mu\}$. Equations (2-2) and (2-3) indicate that a random process can be separated into two parts, the deterministic part and random part. The deterministic part specifies the mean of the random process and the random part characterizes the chance fluctuation of the random process.

When it is possible to reduce a random differential equation into a simple form with the mean zero property, it simplifies the solution of the equation. In view of this, we substitute Equation (2-2) into Equation (2-1), to obtain

$$[m](\{\ddot{\mu}\}+\{\ddot{Z}\}) + [c](\{\dot{\mu}\}+\{\dot{Z}\}) + ([\lambda]+[K])(\{\mu\}+\{Z\}) = \{\phi\} + \{F\}. \qquad (2\text{-}4)$$

The mean response can be obtained from the above equation; then the remaining part characterizes the random component of Equation (2-1). As an approximation, it is assumed that the mean response is excited by the mean of the input with the stiffness equal to its mean value. A motivation for this approximation can be established as follows. Let g represent a measure of a response random process that is a function of the excitation and structural parameter random variables, α_i, $i = 1, \cdots N$. The functional expression is

$$g = f(\alpha_1, \alpha_2, \cdots \alpha_N). \tag{2-5}$$

Using a Taylor series, f can be expanded about the means of the random variables to obtain

$$g = f(\alpha_1, \alpha_2, \cdots \alpha_N) =$$

$$= f(\mu_1, \cdots \mu_N) + \sum_{i=1}^{N} (\alpha_i - \mu_i) \left. \frac{\partial f}{\partial \alpha_i} \right|_{\mu_i} + \cdots, \tag{2-6}$$

where μ_i, $i = 1, \cdots N$ denotes the mean of α_i. When the deviations from the means are small, the series can be truncated following its linear terms. The expected value of g is, therefore,

$$E[g] \simeq f(\mu_1, \cdots \mu_N). \tag{2-7}$$

Equation (2-7) shows that the mean value of a measure of a response random process can be approximated when all the random variables upon which the response depends take their mean values.

Based on Equation (2-7), the mean response can be obtained by solving

$$[m]\{\ddot{\mu}\} + [c]\{\dot{\mu}\} + [\lambda]\{\mu\} = \{\phi\}. \tag{2-8}$$

The remaining component of Equation (2-4) then can be obtained by subtracting Equation (2-8) from Equation (2-4),

$$[m]\{\ddot{Z}\} + [c]\{\dot{Z}\} + ([K] + [\lambda])\{Z\} = \{F\} - [K]\{\mu\}. \tag{2-9}$$

In Equation (2-9), $[K]\{Z\}$ is the only term that involves the product of two random quantities. In this sense it is a higher order term. Its values will be relatively small when random fluctuations in the stiffness and forcing function are small compared to their mean values. By neglecting this term, Equation (2-9) can be reduced to the following form:

$$[m]\{\ddot{Z}\} + [c]\{\dot{Z}\} + [\lambda]\{Z\} = \{F\} -[K]\{\mu\}. \tag{2-10}$$

Equation (2-10) approximately governs the random part of Equation (2-1), and Equation (2-8) approximately governs the mean of Equation (2-1). It is noted that in the above equation, $\{Z\}$ and $\{F\}$ both are zero-mean vector random processes and $[K]$ is a zero-mean matrix random variable.

The advantage of separating Equation (2-1) into Equations (2-8) and (2-10) is that Equations (2-8) and (2-10) have deterministic coefficients, which are relatively easier to handle than random ones. When Equation (2-8) represents the mean response of Equation (2-1), and Equation (2-10) represents the random part of Equation (2-1), the mean square response measures can be obtained directly from Equation (2-10). Equations (2-8) and (2-10) are not independent since Equation (2-10) includes the mean response on the right hand side. The techniques of solving Equations (2-8) and (2-10) will be discussed in Section 2.3.

It is noted that the higher order term is neglected in Equation (2-10). However, the higher order term can be treated and solved for a SDF system when the random stiffness K and the response random process Z(t) are both Gaussian distribution. The discussions about the omission of the higher order term are referred to in Article 7.1.

2.3 FINITE DIFFERENCE SOLUTION

2.3.1 Difference Formulation

In Section 2.2, it was established that the model of a random differential equation with random coefficients can be separated into two parts, the deterministic part and the random part. Equations (2-8) and (2-10) represent the model. In this section, techniques for solving Equations (2-8) and (2-10) are established. Equation (2-8) is considered first:

$$[m]\{\ddot{\mu}\} + [c]\{\dot{\mu}\} + [\lambda]\{\mu\} = \{\phi\}. \tag{2-8}$$

A solution to this equation is sought using a finite difference approach. This solution establishes the displacement response at the times $t_j = j\Delta t$, $j = 0, 1, 2, \cdots$, where Δt is a small positive time increment. At time t_j, the equation governing motion is

$$[m]\{\ddot{\mu}_j\} + [c]\{\dot{\mu}_j\} + [\lambda]\{\mu_j\} = \{\phi_j\} \tag{2-11}$$

where $\{\ddot{\mu}_j\}$ is the mean value of response acceleration at t_j, $\{\dot{\mu}_j\}$ is the mean value of response velocity at t_j. When the central difference method is used, the acceleration and velocity are approximated by

$$\{\ddot{\mu}_j\} = (\{\mu_{j+1}\} - 2\{\mu_j\} + \{\mu_{j-1}\})/\Delta t^2, \tag{2-12a}$$

$$\{\dot{\mu}_j\} = (\{\mu_{j+1}\} - \{\mu_{j-1}\})/2\Delta t. \tag{2-12b}$$

Substitution of Equation (2-12) into Equation (2-11) yields

$$\frac{[m](\{\mu_{j+1}\} - 2\{\mu_j\} + \{\mu_{j-1}\})}{\Delta t^2} + \frac{[c](\{\mu_{j+1}\} - \{\mu_{j-1}\})}{2\Delta t} + [\lambda]\{\mu_j\} = \{\phi_j\}. \tag{2-13}$$

This expression can be solved for $\{\mu_{j+1}\}$ to obtain

$$\{\mu_{j+1}\} = [A_1]^{-1}([A_2]\{\mu_j\} + [A_3]\{\mu_{j-1}\} + \Delta t^2\{\phi_j\}), \qquad j = 0, 1, 2, \cdots \tag{2-14}$$

in which

$$\begin{aligned} [A_1] &= [c]\Delta t/2 + [m], \\ [A_3] &= 2[m] - [\lambda]\Delta t^2, \\ [A_3] &= [c]\Delta t/2 - [m]. \end{aligned} \tag{2-15}$$

Equation (2-14) provides a recurrence relation for the displacement response. To start the calculation it is necessary to know $\{\mu_{-1}\}$. Because $\{\dot{\mu}_0\}$ and $\{\ddot{\mu}_0\}$ are known (usually, they are zero), the relations of Equation (2-12) can be used to obtain $\{\mu_{-1}\}$,

$$\{\mu_{-1}\} = \{\mu_0\} - \Delta t\{\dot{\mu}_0\} + \Delta t^2\{\ddot{\mu}_0\}/2. \tag{2-16}$$

Using Equations (2-16) and (2-14), the mean displacement solution then can be obtained.

Now Equation (2-10) is used to characterize the random response component,

$$[m]\{\ddot{Z}\} + [c]\{\dot{Z}\} + [\lambda]\{Z\} = \{F\} - [K]\{\mu\}. \qquad (2\text{-}10)$$

where [m], and [c] are the mass and stiffness matrices and [λ] is the deterministic mean of the stiffness matrix. Because the system under consideration is linear with constant coefficients, [λ] is constant throughout the time. At time step j, the response is governed by

$$[m]\{\ddot{Z}_j\} + [c]\{\dot{Z}_j\} + [\lambda]\{Z_j\} = \{F_j\} - [K]\{\mu_j\}. \qquad (2\text{-}17)$$

Again, apply the central difference approximation,

$$\frac{[m](\{Z_{j+1}\} - 2\{Z_j\} + \{Z_{j-1}\})}{\Delta t^2} + \frac{[c](\{Z_{j+1}\} - \{Z_{j-1}\})}{2\Delta t} + [\lambda]\{Z_j\} =$$
$$= \{F_j\} - [K]\{\mu_j\}. \qquad (2\text{-}18)$$

Solving for $\{Z_{j+1}\}$ yields

$$\{Z_{j+1}\} = [A_1]^{-1}([A_2]\{Z_j\} + [A_3]\{Z_{j-1}\} + \Delta t^2\{F_j\} - \Delta t^2[K]\{\mu_j\}) \qquad (2\text{-}19)$$

where $[A_1]$, $[A_2]$, $[A_3]$ are the constant matrices which are given in Equation (2-15). Equation (2-19) represents the random part of the response at time step j+1.

Let $\{Z_{j+1}\}^T$ denote the transpose of $\{Z_{j+1}\}$. The mean square response measure, $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$, can then be established. When $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$ is computed, it will contain the terms which involve the product of $\{F_j\}$ and $\{Z_j\}$, $\{Z_{j-1}\}$, etc. However, it is true that when the input is white noise excitation, the force at time t_j is independent of the displacement response at t_j. Therefore

$$E[\{F_j\}\{Z_j\}^T] = E[\{F_j\}]\ E[\{Z_j\}^T] = 0. \qquad (2\text{-}20a)$$

Another observation is that the force at time t_j is independent of the displacement response at t_{j-1} if the excitation is a sequence of independent and independently arriving random impulses [33],

$$E[\{F_j\}\{Z_{j-1}\}^T] = E[\{F_j\}]\ E[\{Z_{j-1}\}^T] = 0. \tag{2-20b}$$

Further, it is reasonable to assume that the force is independent of [K], i.e.,

$$E[[K]\{\mu_j\}\{F_j\}^T] = E[[K]\{\mu_j\}]\ E[\{F_j\}^T] = 0. \tag{2-21}$$

Using Equations (2-20) and (2-21), it can be shown that

$$\begin{aligned}
E[\{Z_{j+1}\}\{Z_{j+1}\}^T] = [A_1]^{-1}([A_2]E[\{Z_j\}\{Z_j\}^T]\ [A_2]^T & + \\
+ [A_3]E[\{Z_{j-1}\}\{Z_{j-1}\}^T][A_3]^T + \Delta t^4 E[\{F_j\}\{F_j\}^T] & + \\
+ \Delta t^4 E[[K]\{\mu_j\}\{\mu_j\}^T[K]^T] + [A_2]E[\{Z_j\}\{Z_{j-1}\}^T]\ [A_3]^T & + \\
+ [A_3]E[\{Z_{j-1}\}\{Z_j\}^T][A_2]^T & - \\
- \Delta t^2 E[[K]\{\mu_j\}\{Z_j\}^T][A_2]^T & - \\
- \Delta t^2 [A_2]E[\{Z_j\}\{\mu_j\}^T[K]^T] & - \\
- \Delta t^2 E[[K]\{\mu_j\}\{Z_{j-1}\}^T][A_3]^T & - \\
- \Delta t^2 [A_3]E[\{Z_{j-1}\}\{\mu_j\}^T[K]^T])[A_1]^{-T}. &
\end{aligned} \tag{2-22}$$

Note that, in the framework of a recursive solution, $E[\{Z_j\}\{Z_j\}^T]$ and $E[\{Z_{j-1}\}\{Z_{j-1}\}^T]$ can be obtained from the previous one and two steps of computation if the current analysis evaluates $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$. Further, the expression is simplified if it is noted that $[A_3]E[\{Z_{j-1}\}\{Z_j\}^T][A_2]^T$,

$[A_2]E[\{Z_j\}\{\mu_j\}^T[K]^T]$, and $[A_3]E[\{Z_{j-1}\}\{\mu_j\}^T[K]^T]$ are the transposes of $[A_2]E[\{Z_j\}\{Z_{j-1}\}^T][A_3]^T$, $E[[K]\{\mu_j\}\{Z_j\}^T][A_2]^T$ and $E[[K]\{\mu_j\}\{Z_{j-1}\}^T][A_3]^T$, respectively.

The autocorrelation function of the given excitation $E[\{F_j\}\{F_j\}^T]$ is assumed known or enumerable. The correlation terms $E[\{Z_j\}\{Z_{j-1}\}^T]$, $E[[K]\{\mu_j\}\{Z_j\}^T]$, $E[[K]\{\mu_j\}\{Z_{j-1}\}^T]$ and $E[[K]\{\mu_j\}\{\mu_j\}^T[K]^T]$ must be evaluated in order to compute $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$. These terms involve two vector or matrix random variables and are accompanied by one or two deterministic variables. Some special techniques must be provided to solve for these quantities. In the next several sections it will be shown how these terms can be treated and how methods for their evaluations are developed.

2.3.2 Evaluation of $E[[K]\{\mu_j\}\{\mu_j\}^T[K]^T]$

In this section, it will be shown how the term, $E[[K]\{\mu_j\}\{\mu_j\}^T[K]^T]$ can be generated in the step-by-step solution. The $E[[K]\{\mu_j\}\{\mu_j\}^T[K]^T]$ is an expectation of a form where the inner variables are deterministic and the outer variables are random. The general form of this type of problem can be expressed as $E[[K]\{p\}\{q\}^T[K]^T]$ where $[K]$ is the random matrix under consideration and $\{p\}$, $\{q\}$ are any deterministic column vectors. Let $[A] = E[[K]\{p\}\{q\}^T[K]^T]$, then

$$A_{rs} = E[K_{ri}p_iq_mK_{sm}] = p_iq_mE[K_{ri}K_{sm}], \qquad \text{for } r,s = 1, \cdots N \qquad (2\text{-}23)$$

where A_{rs} denotes the element in the r^{th} row and s^{th} column of $[A]$. The notation used here is the indicial notation, i.e., any repeated index means that a sum is executed over that index.

In order to establish the correlations between stiffness terms it is necessary to go back to the definition of the stiffness matrix. Recall from Equation (2-2) that [k] represents the random stiffness matrix. Each term in the stiffness matrix of a structural frame is a function of the cross sectional dimension of frame members, the member length, and Young's modulus. By noting this, if only Young's modulus is considered as random and the rest of variables as deterministic, the mathematical expression for k_{ri}, $r,i = 1, \cdots N$ then can be written as:

$$k_{ri} = f_1(E) = f_1(\alpha), \qquad \text{for } r,i = 1, \cdots N \tag{2-24}$$

where E denotes Young's modulus and α denotes the only underlying random variable which is E. It is important to note that [k] represents the total stiffness matrix (mean plus random components) and [K] represents the random component of the stiffness matrix (see Equation (2-2)).

When Taylor's expansion is used, Equation (2-24) can be expanded about the mean of its underlying random variable, i.e.,

$$k_{ri} = f_1(\mu_\alpha) + (\alpha - \mu_\alpha) \left.\frac{\partial f_1}{\partial \alpha}\right|_{\mu_\alpha} + \cdots, \qquad \text{for } r,i = 1, \cdots N \tag{2-25}$$

where μ_α denotes the mean of α. Use of Equation (2-2) leads to the following result:

$$K_{ri} = k_{ri} - E[k_{ri}] = (\alpha - \mu_\alpha) \left.\frac{\partial f_1}{\partial \alpha}\right|_{\mu_\alpha} + \cdots. \qquad \text{for } r,i = 1, \cdots N \tag{2-26}$$

For small deviation, the product of any two elements in [K] can be established by neglecting higher order terms. The mean value of a specific product is

$$E[K_{ri}K_{sm}] = \sigma_\alpha^2 \left.\frac{\partial f_1}{\partial \alpha}\right|_{\mu_\alpha} \left.\frac{\partial f_2}{\partial \alpha}\right|_{\mu_\alpha} ,$$

where σ_α denotes the standard deviation of α. Furthermore, if we postulate that $\sigma_\alpha = \beta\, E[\alpha]$ where β is the constant coefficient of variation, Equation (2-23) can be simplified,

$$E[K_{ri}p_i q_m K_{sm}] = \beta^2 \lambda_{ri} p_i q_m \lambda_{sm}, \tag{2-27}$$

where $[\lambda]$ is defined in Equation (2-2) and β is the coefficient of variation of Young's modulus. Equation (2-27) provides a general and simple solution when two random matrices are separated by any constant matrix. A more complicated form can be established in the same way when two or more random variables are included in Equation (2-24).

2.3.3 Evaluation of $E[[K]\{\mu_j\}\{Z_j\}^T]$

In this section, the term $E[[K]\{\mu_j\}\{Z_j\}^T]$ is discussed. Development of an expression for this term is a problem similar to the one discussed in Section 2.3.2 because the outer variables are random and the inner variable is deterministic. However, $[K]$ is negatively correlated with $\{Z_j\}$ and a different approach to the solution must be pursued. Noted that the middle variable, $\{\mu_j\}$, represents the mean response of the system at time step j and is a known quantity. Recall that for dynamic response analysis of a linear multi-degree-of-freedom system, a sometimes useful representation of the displacement is provided by the free vibration mode shapes. Any displacement vector, $\{\mu\}$, for a system can be developed by superposing suitable amplitudes of the modes of vibration. Let $\{b_i\}$, $i = 1, \cdots N$ represent the orthonormal mode shapes of a system which satisfies Equation (2-3) and $c_i^{(j)}$, $i = 1, \cdots N$ denote the corresponding amplitudes at time t_j. Then we can write

$$\{\mu_j\} = c_1^{(j)}\{b_1\} + \cdots + c_N^{(j)}\{b_N\} = \sum_{i=1}^{N} c_i^{(j)}\{b_i\}, \tag{2-28}$$

where the superscript (j) denotes the jth time step. It is noted that $c_i^{(j)}$, $i = 1, \cdots N$ are time dependent and if we choose $c_i^{(j)}$ properly then $\{\mu_j\}$ can be evaluated precisely. The $c_i^{(j)}$ can be obtained easily by noting that $\{b_i\}$ satisfies the orthogonality relation i.e.,

$$\{b_r\}^T[m]\{b_s\} = \begin{cases} 1 & \text{if } r = s \\ 0 & \text{else} \end{cases}, \tag{2-29}$$

where [m] is the system mass matrix. With the aid of Equation (2-28), an expression for $c_i^{(j)}$ can be developed.

$$\{b_r\}^T[m]\{\mu_j\} = \{b_r\}^T[m]\left(\sum_{i=1}^{N} c_i^{(j)}\{b_i\}\right) = c_r^{(j)}. \qquad \text{for } r = 1, \cdots N \tag{2-30}$$

The above equation shows how the modal amplitudes relate to the mean response. When $c_i^{(j)}$, $i = 1, \cdots N$ are evaluated in Equation (2-28), $E[[K]\{\mu_j\}\{Z_j\}^T]$ can be rewritten:

$$E[[K]\{\mu_j\}\{Z_j\}^T] = E[[K](\sum_{i=1}^{N} c_i^{(j)}\{b_i\})\{Z_j\}^T] =$$

$$= \sum_{i=1}^{N} c_i^{(j)}(E[[K]\{b_i\}\{Z_j\}^T]). \qquad (2\text{-}31)$$

$\{Z_j\}$ can be developed using Equation (2-19) at time step j. Use of Equation (2-21) leads to the result:

$$E[[K]\{b_i\}\{Z_j\}^T] = (E[[K]\{b_i\}\{Z_{j-1}\}^T][A_2]^T + E[[K]\{b_i\}\{Z_{j-2}\}^T][A_3^T] -$$

$$- \Delta t^2 E[[K]\{b_i\}\{\mu_j\}^T[K]^T])[A_1]^{-T}. \qquad (2\text{-}32)$$

Terms like $E[[K]\{b_i\}\{Z_{j-1}\}^T]$ and $E[[K]\{b_i\}\{Z_{j-2}\}^T]$ of the above equation can be obtained from the previous two steps when currently computing $E[[K]\{b_i\}\{Z_j\}^T]$. The last term on the right hand side of the above equation possesses the same form as that developed in Section 2.2.1. Equation (2-32) then provides a means for expressing $E[[K]\{\mu_j\}\{Z_j\}^T]$. It is noted that Equations (2-31) and (2-32) also provide $E[[K]\{\mu_j\}\{Z_{j-1}\}^T]$ when Equation (2-28) is used.

2.3.4 Evaluation of $E[\{Z_j\}\{Z_{j-1}\}^T]$

In this section, the term, $E[\{Z_j\}\{Z_{j-1}\}^T]$, is discussed. If $\{Z_j\}$ is replaced with the expression shown in Equation (2-19) at time t_j, another recurrence relationship will be obtained. Again, with the aid of Equation (2-21), it can be shown that,

$$E[\{Z_j\}\{Z_{j-1}\}^T] = [A_1]^{-1}([A_2]E[\{Z_{j-1}\}\{Z_{j-1}\}^T] + + [A_3]E[\{Z_{j-1}\}\{Z_{j-2}\}^T]^T - \Delta t^2 E[[K]\{\mu_{j-1}\}\{Z_{j-1}\}^T]). \tag{2-33}$$

Several observations are made regarding the above equation. The first term on the right hand side is known if, for example, at time t_{j+1} the term $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$ is computed. The second term on the right hand side can be obtained from the previous computation if $E[\{Z_j\}\{Z_{j-1}\}^T]$ is computed in the present step. The last term possesses the same form as was discussed in Section 2.3.3. Therefore, Equation (2-33) provides the expression for $E[\{Z_j\}\{Z_{j-1}\}^T]$ and all its component parts can be computed.

2.4 SUMMARY

The techniques described in Sections 2.3.2, 2.3.3 , 2.3.4 provide enough information to evaluate Equation (2-22). As stated before, $\{Z_j\}$ itself is a zero mean random process. The mean square response, together with the deterministic response, characterizes the statistical properties of the response random process. Furthermore, if the probability distribution is assumed to be Gaussian, the first and second moments computed above completely characterize the random process. Hence, the probabilistic description of the random system is established.

Chapter 3

NONLINEAR SYSTEM

3.1 INTRODUCTION

It was shown in Chapter 2 that the linear random differential equation with random coefficient can be separated into two parts, the deterministic part and the random part. Equations (2-8) and (2-10) were established to approximately define the model. The mean square response then was evaluated as Equation (2-22).

In this Chapter a technique is developed to compute the mean and mean square response of nonlinear structures. In this investigation it is assumed that only the stiffness matrix behaves nonlinearly. Specifically, the elasto-plastic property is assumed to govern material behavior. The type of system to be considered is a structural framework. For such a system the stiffness matrix is time dependent and is related to the displacement response time history. For this reason it is necessary to combine the stiffness matrix and displacement response to a vector which represents restoring forces, or internal forces. However, the displacement dependent stiffness matrix still needs to be evaluated in order to compute the mean square responses.

The technique for analyzing the nonlinear system is similar to linear problem because only the stiffness term needs to be reconsidered There-fore, Equations like (2-8) and (2-10) still can be used at each step to evaluate the mean and mean square responses. Some modifications are necessary. In the next section, the model of the nonlinear system is discussed.

3.2 NONLINEAR MODEL

The stiffness matrix in Equation (2-1) characterizes the restoring force of a structural framework. The stiffness matrix is formed by assembling the stiffness of individual beam elements relating element stiffness terms to specific elements in the global stiffness matrix. A beam element with specific deformations at each degree-of-freedom (DOF) is shown in Figure 3.1

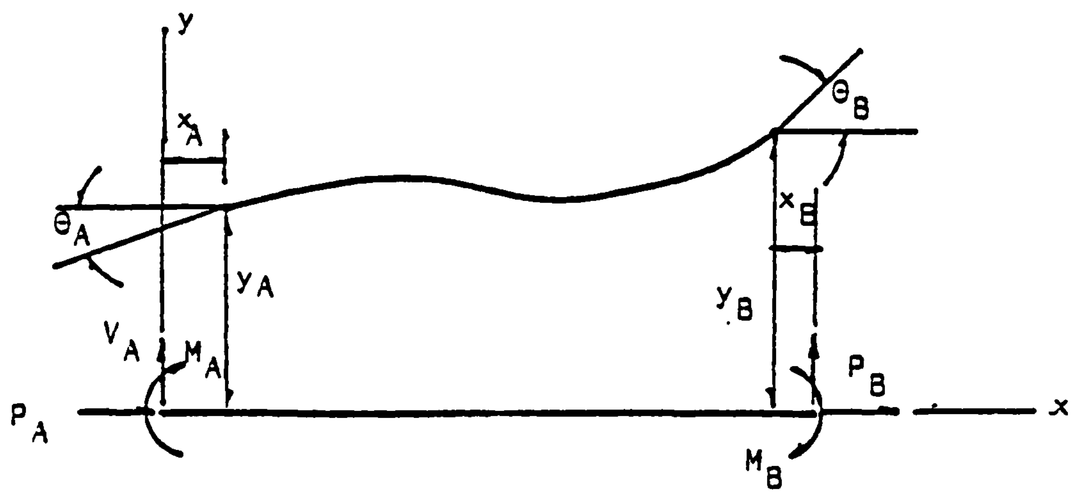

Figure 3.1 Typical beam element

where x_A, y_A, θ_A are the axial, vertical and rotational deformations at point A and P_A, V_A, M_A are the corresponding nodal forces. (The quantities at end B are defined similarly.) The relationships among the deformations and forces can be easily obtained when the stress and strain in the beam remain linear. However, when plastic response is allowed to occur, the linear theory no longer holds. In the present study the material response is assumed to be elasto-plastic. The relationship among nodal forces and deformations that accounts for this must be established. Before these relationships can be established, some preliminary theories must first be discussed.

First, the beam equation for an inelastic element will be derived. Consider a small length dx cut from a beam

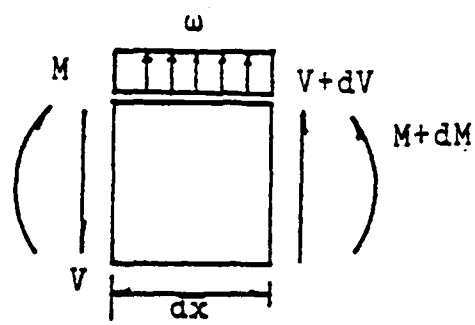

Figure 3.2 Small length cut from a beam

where ω is the external load. If the inertia effect of beam is neglected for the stiffness only, the equilibrium equations are to be satisfied, that is

$$V - \omega dx = V + dV, \qquad (3\text{-}1)$$

$$M + dM - M + Vdx = 0.$$

Therefore,

$$\frac{dV}{dx} = -\omega, \qquad \frac{dM}{dx} = -V, \qquad \frac{d^2M}{dx^2} = \omega. \qquad (3\text{-}2)$$

Equation (3-2) shows that the equilibrium equations for the deflection curve of a beam are the same regardless of whether or not the response is plastic.

Another important property which is not affected by the presence of plasticity is the relationship between strain and curvature since it relates to the geometry only,

$$\varepsilon_b(x,z) = \kappa z. \tag{3-3}$$

The $\varepsilon_b(x,z)$ is the bending strain in the beam at a point whose coordinates are x and z. The coordinate x is shown in Figure 3.1. The coordinate z is the vertical distance from the neutral axis. The strain is assumed to be constant across the width of the beam element. The κ represents the curvature of the neutral axis at point x.

Using the above notation it is possible to establish the relationships among nodal deformations and forces. First, the material under consideration shall satisfy the Prandtl-Reuss relationship. A typical stress strain curve for such material is shown in Figure 3.3.

If σ_y and ε_y denote the yield stress and strain respectively then Young's modulus is defined by $E = \sigma_y/\varepsilon_y$. Any strain larger than ε_y will force the material to experience permanent set which is defined by $\varepsilon_o = \varepsilon - \varepsilon_y$. The ε_o represents the permanent sets in the loaded material.

Before the elasto-plastic model can be used to analyze the stress and strain in a beam during a dynamic analysis, several problems must first be discussed. From Figure 3.1 it may be seen that any bar under consideration can be subjected to the simultaneous action of bending loads and axial forces. To understand how these operate simultaneously, consider the following figures which can represent the strain and stress distribution at any section in the bar.

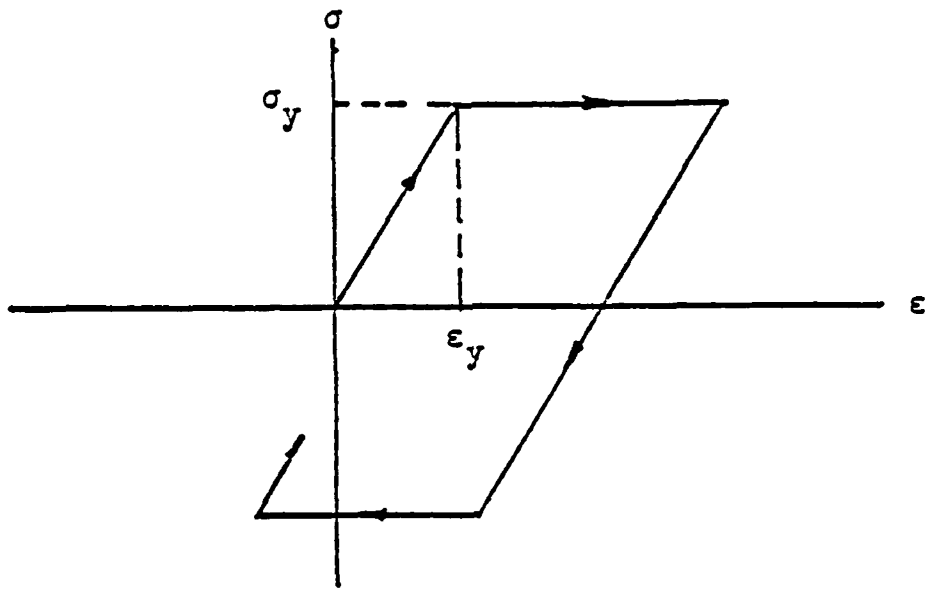

Figure 3.3 Stress-strain curve of elasto-plastic material

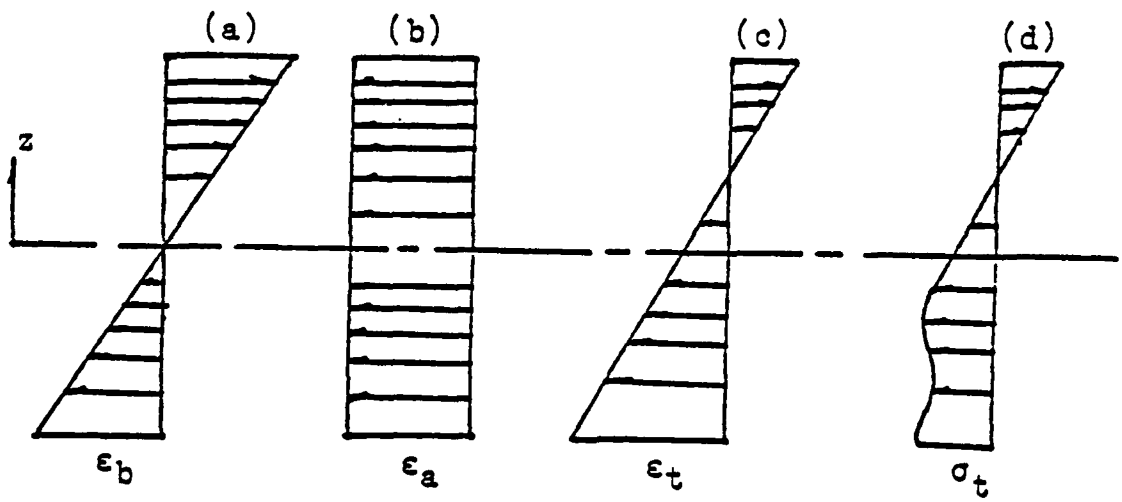

Figure 3.4 Combined bending and axial load

In Figure 3.4, ε_b, ε_a are the strain due to bending and axial forces respectively. The total strain, ε_t, is the superposition of ε_b and ε_a, hence, we can write $\varepsilon_b + \varepsilon_a = \varepsilon_t$. This strain is shown in Figure 3.4(c). Figure 3.4(d) is the corresponding stress distribution (after possibly several load and response cycles). Note that the stress is not linear in z because the material has been in plastic state and undergone permanent sets as shown in Figure 3.3. The permanent sets in the beam occur arbitrarily and the reason for this will be discussed in the next section. The total stress $\alpha_t(x,z)$ for an elastic-perfectly plastic material can be expressed as:

$$\sigma_t = E(\varepsilon_a(x) + \varepsilon_b(x,z) - \varepsilon_o(x,z)), \tag{3-4}$$

where $\varepsilon_o(x,z)$ is the permanent set at x-section and z fibre position. A symmetric I-beam, shown in Figure 3.5, is illustrated for analysis.

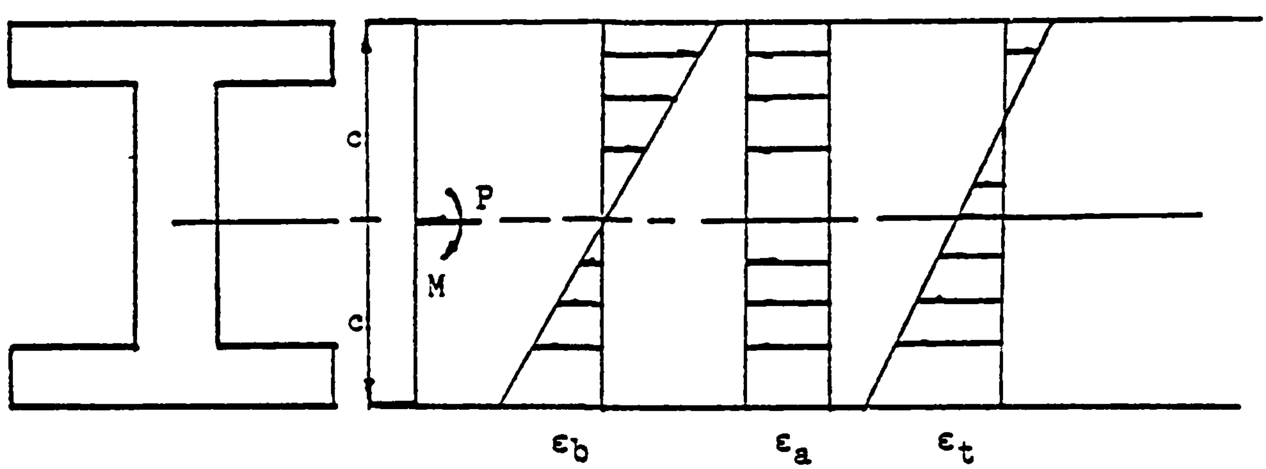

Figure 3.5 Cross section of an I-beam

Let the total load at x-section be resolved into an axial force P act at the neutral axis and a bending moment M(x). Then the moment for this section can be evaluated by

$$M(x) = \int_A \sigma_t w(z) z dz, \qquad (3\text{-}5)$$

where $w(z)$ is the width of the beam at z. When σ_t is replaced by (3-4),

$$M(x) = E \int_A (\varepsilon_a(x) + \varepsilon_b(x,z) - \varepsilon_o(x,z))\ w(z) z dz. \qquad (3\text{-}6)$$

Equation (3-3) can be used to simplify ε_b to

$$\varepsilon_b(x,z) = z\varepsilon_b(x,c)/c, \qquad (3\text{-}7)$$

where $\varepsilon_b(x,c)$ stands for the strain at upper fiber. By noting that $\int_A \varepsilon_a(x) w(z) z dz = 0$, Equation (3-6) then can be reduced to

$$M(x) = \frac{E}{c}\ \varepsilon_b(x,c) \int_{-c}^{c} z^2 w(z) dz - E \int_{-c}^{c} \varepsilon_o(x,z) z w(z) dz =$$

$$= EI\ \frac{\varepsilon_b(x,c)}{c} - E \int_{-c}^{c} \varepsilon_o(x,z) z w(z) dz,$$

where $I = \int_A z^2 w(z) dz$. When deflections in the beam are small, $\varepsilon_b/c \simeq y''$, therefore, the moment is

$$M(x) = EIy'' - E \int_{-c}^{c} \varepsilon_o(x,z) z w(z) dz. \qquad (3\text{-}8)$$

Note that if $\varepsilon_o(x,z)$ is zero the above equation reduces to a linear system, as it should. Rearranging Equation (3-8) yields

$$EIy'' = M(x) + E \int_{-c}^{c} \varepsilon_o(x,z) z w(z) dz \qquad (3\text{-}9)$$

Using Equation (3-2) a fourth order beam equation can be developed,

$$EIy'''' = \omega + E \int_{-c}^{c} \frac{\partial^2 \varepsilon_o(x,z)}{\partial x^2} zw(z)dz.$$

In the present analysis of frame structures it can always be assumed that the loads act at nodal points, therefore, we can let $\omega = 0$. Therefore,

$$EIy''' = E \int_{-c}^{c} \frac{\partial \varepsilon_o(x,z)}{\partial x} zw(z)dz + C_1, \tag{3-10}$$

$$EIy'' = E \int_{-c}^{c} \varepsilon_o(x,z)zw(z)dz + C_1x + C_2, \tag{3-11}$$

$$EIy' = E \int_0^x du \int_{-c}^{c} \varepsilon_o(u,z)zw(z)dz + \frac{1}{2} C_1x^2 + C_2x + C_3, \tag{3-12}$$

$$EIy = E \int_0^x dv \int_0^v du \int_{-c}^{c} \varepsilon_o(u,z)zw(z)dz + \frac{1}{6} C_1x^3 + \frac{1}{2} C_2x^2 + C_3x + C_4. \tag{3-13}$$

The constant C_i, $i = 1, \cdots 4$ can be obtained from the boundary condition, these are

$$\begin{aligned} y'(0) &= \theta_A, & y'(L) &= \theta_B, \\ y(0) &= y_A, & y(L) &= y_B. \end{aligned} \tag{3-14}$$

Use of Equation (3-14) in Equations (3-11) through (3-13) results in

$$\begin{aligned} C_1 &= \frac{12}{L^3}(q_1L/2 - q_2), & C_2 &= \frac{12}{L^2}(q_2/2 - q_1L/6), \\ C_3 &= EI\theta_A, & C_4 &= EIy_A, \end{aligned} \tag{3-15}$$

where

$$q_1 = EI(\theta_B - \theta_A) - E\int_0^L du \int_{-c}^{c} \varepsilon_o(u,z)zw(z)dz,$$

$$q_2 = EI(y_B - y_A) - EI\theta_A L - E\int_0^L dv \int_0^v du \int_{-c}^{c} \varepsilon_o(u,z)zw(z)dz.$$

After the constant C_i, $i = 1, \cdots 4$ are evaluated, the nodal forces can be established.

It can be shown that the moment in the beam is linear function of x by substituting Equation (3-8) into Equation (3-6).

$$M(x) = C_1x + C_2. \tag{3-16}$$

Using Equation (3-1),

$$V(x) = C_1. \tag{3-17}$$

Equations (3-16) and (3-17) give the relationship among nodal deformations and nodal forces. In other words, if the nodal deformations are provided then the moments and vertical shearing forces can be evaluated from Equations (3-16) and (3-17).

Now the axial forces are considered. Note that the axial forces also can be obtained using the results developed above. Referring to Figure 3.5,

$$P = \int_A \sigma_t w(z)dz. \tag{3-18}$$

Again, using Equation (3-4), and noting the $\int_A \varepsilon_b(x,z)w(z)dz = 0$ and $\int_A w(z)dz = A$ where A is the cross section area, we have

$$P = E\varepsilon_a(x)A - E\int_{-c}^{c}\varepsilon_o(x,z)w(z)dz. \tag{3-19}$$

Integrating the expression over the length yields the following result.

$$\int_0^L Pdx = EA\int_0^L \varepsilon_a(x)dx - E\int_0^L du\int_{-c}^{c}\varepsilon_o(u,z)w(z)dz.$$

Let $\int_A \varepsilon_o(x)dx = \delta$ be the absolute axial deformation. Then

$$PL = EA\delta - E\int_0^L du\int_{-c}^{c}\varepsilon_o(u,z)w(z)dz. \tag{3-20}$$

Therefore,

$$P = \frac{E}{L}\,(A\delta - \int_0^L du\int_{-c}^{c}\varepsilon_o(u,z)w(z)dz). \tag{3-21}$$

If $\varepsilon_o(u,z)$ is zero, the above equation reduces to linear result, as it should. Note that $\varepsilon_a(x)$ is not a constant over the length of the beam since the permanent sets in Equation (3-19) are not constant.

Equations (3-21), (3-16) and (3-17) provide the relationships among the nodal deformations and forces. At this point, the permanent sets, however, are still unknown. In order to evaluate the permanent sets an iteration method can be developed. This will be discussed in the next section.

3.3 ITERATION SCHEME FOR COMPUTATION OF MEAN RESPONSE

As mentioned in the previous section the permanent sets must be obtained using an iteration method since no closed form solution can be established. Before considering the iteration scheme, a property of the permanent sets is considered. It was stated earlier that the permanent sets accumulate arbitrarily. As an example consider the following beam cross section and stress-strain curve.

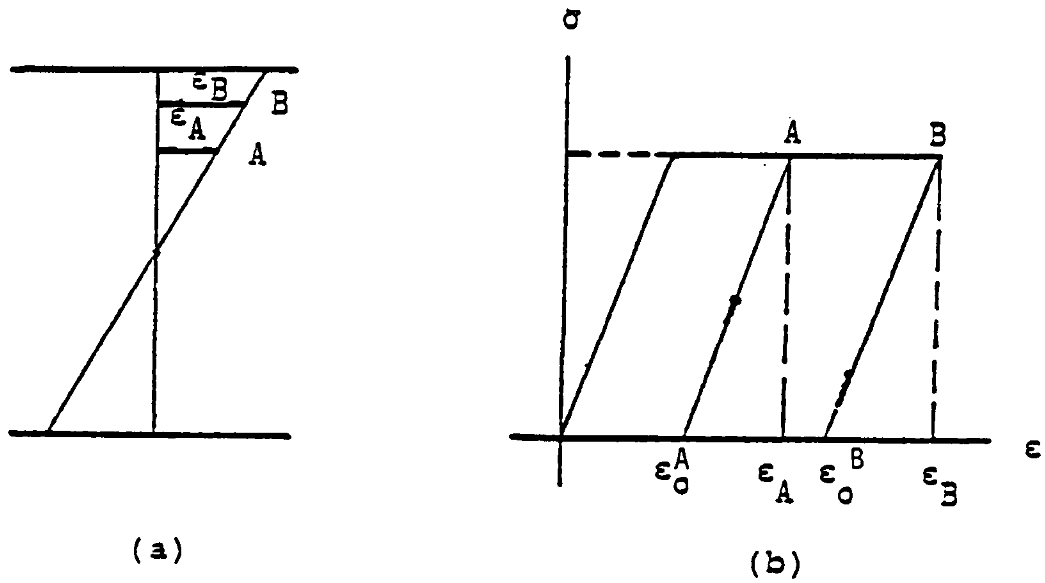

Figure 3.6 Permanent-set property

Figure 3.6(a) shows a typical strain curve of a cross section with bending only. Points A and B represent any two separate points with strains ε_A and ε_B, respectively. Since the inputs are random, the strain at points A and B may behave randomly, however, still possess linear relation as shown in Figure 3.6(a) if only bending is considered. The corresponding positions of ε_A and ε_B in the stress-strain curve may exist like Figure 3.6(b). At a specific time, if ε_A and ε_B start to decrease, the strain history paths of points A and B on the stress-strain curve will be different as shown in Figure 3.6(b). Therefore, at this stage points A and B will possess permanent sets ε_o^A and ε_o^B, respectively.

Furthermore, if the strain due to axial force is added, a more complicated form of the permanent sets will emerge. An important fact is that the strain due to bending and axial forces act simultaneously. The permanent sets in the beam can be evaluated only when the two quantities discussed above are added. According to the above statements, we conclude that the permanent sets in the beam exist in an arbitrary form and can be evaluated only from the total strain ε_t.

Reconsider now the equation of motion. Recall that Equation (2-8) approximately represents the mean response of a linear elastic system. For a nonlinear system, as discussed in the previous section, the equation of motion can be modified by replacing the restoring force term, $[\lambda]\{\mu\}$, in Equation (2-8) by $\{R(\mu)\}$ which is the restoring force of the nonlinear system. The equation governing the mean response is

$$[m]\{\ddot{\mu}\} + [c]\{\dot{\mu}\} + \{R(\mu)\} = \{\phi\}. \tag{3-22}$$

If the central difference method is used to solve this equation, then by means of Equation (2-12), the displacements at time t_{j+1} are evaluated using the formula

$$\{\mu_{j+1}\} = [A_1]^{-1}(2[m]\{\mu_j\} + [A_3]\{\mu_{j-1}\} + \Delta t^2(\{\phi_j\} - \{R(\mu_j)\})), \tag{3-23}$$

where $[A_1]$ and $[A_3]$ are given in Equation (2-15). The above equation shows that the displacement responses at time t_{j+1} are dependent on the restoring forces at time t_j only. By noting this, the numerical iteration sequence can be written as follows:

(1) The displacement responses at time t_{j+1} are computed from Equation (3-23). By the assumption that the system starts at rest, $\{\mu_o\}$, $\{\mu_{-1}\}$ and $\{R(\mu_o)\}$ are all zero.

(2) For each member, the nodal displacements can be obtained from $\{\mu_{j+1}\}$ which are in global coordinates. Hence, the coordinate transformation is required in order to obtain the deformations of each member in local coordinates.

(3) At the beginning of time t_{j+1}, it is assumed that the permanent sets for this time step are equal to the permanent sets at time t_j. After this assumption is made, terms such as the curvature $y''(x) = \kappa(x)$ and the axial strain $\varepsilon_a(x)$ can be computed from Equations (3-9) and (3-19):

$$y''(x) = \frac{C_1 x + C_2}{EI} + \frac{1}{I}\int_{-c}^{c} \varepsilon_o(x,z) z w(z) dz, \qquad (3\text{-}24)$$

$$\varepsilon_a(x) = \frac{1}{EA}\left(P + E\int_{-c}^{c} \varepsilon_o(x,z) w(z) dz\right), \qquad (3\text{-}25)$$

where C_1 and C_2 can be evaluated from Equation (3-15) and P can be evaluated from Equation (3-21). The $\varepsilon_b(x)$ can be computed from $y''(x)$ using Equation (3-3).

(4) By means of Equation (3-3), the total strain $\varepsilon_t(x)$, $0 \le x \le L$, can then be computed from $\varepsilon_t(x) = \varepsilon_b(x) + \varepsilon_a(x)$.

(5) Based on $\varepsilon_t(x)$ new permanent sets can be evaluated. According to the new permanent sets, new estimates of $y''(x)$, $\varepsilon_a(x)$ and $\varepsilon_t(x)$ can be evaluated. Steps (3) to (5) are repeated until convergence of the permanent sets occurs.

(6) After the final permanent sets are obtained, the restoring forces for each member can be evaluated by Equations (3-13), (3-14) and (3-19). Accordingly, the global restoring forces $\{R(\mu_{j+1})\}$ are assembled by the restoring forces of each member. Some elementary assembling techniques are required to form global restoring forces $\{R(\mu_{j+1})\}$.

Some modifications must be made in connection with the approximation that requires unchanged neutral axis, as stated in the previous section. In some cases the iteration scheme defined above will not converge because of the requirement that the neutral axis remain unchanged. Numerical investigations show that the permanent sets computed in a particular iteration cycle will sometimes converge, and then alternate between two modes during the iteration. In such cases, an approximation for the permanent set can be established by averaging the two modes. Some analyses show that this approximation yields good results.

3.4 FORMULATION

In Chapter 2, the model for separating a random differential equation was established for a linear system. Equations (2-8) and (2-10) represent the model. In the previous section it was shown that the mean response of a nonlinear system can be established by substitution of the nonlinear restoring force for the linear restoring force. The second order characteristics of the nonlinear response must now be established. The techniques for treating nonlinear problems are nearly the same as for linear problems. However, some modifications are necessary in treating the stiffness matrix. It was shown in the previous section that the mean response, $\{\mu\}$, can be represented using Equation (3-22) when the $[\lambda]\{\mu\}$ term is replaced by $\{R(\mu)\}$, the restoring forces of the system. Equation (3-23) then provides the solution of Equation (3-22) if the central difference approach is used.

To use the general approach of Section 2.2 to evaluate the mean square characteristics of a nonlinear system, two things are necessary. First, it is necessary to assume that during a single response computation time step the nonlinear system behaves approximately as a linear system. The reason for this is that the second order characteristics of the response of the nonlinear system will be computed using Equation (2-22). Second, it is necessary to evaluate the stiffness characteristics of a nonlinear system at each time step. The reason is that the stiffness term $[\lambda]$ appears in Equation (2-22). The mean displacement terms required in Equation (2-22) are established as described above.

The equivalent stiffness matrix $[\lambda]$ of this system must be provided in order to carry out the solution of Equation (2-10). This can be done if the restoring forces of the system are known. It was shown in previous section that the restoring forces can be computed when the displacements and the offsets are known. Hence, the equivalent stiffness matrix can be established. Let λ_{sm}, $s,m = 1, \cdots N$, denote the elements in the stiffness matrix. Recall that λ_{sm} is defined as the force developed at degree of freedom s due to a unit displacement at degree of freedom m when all other degrees of freedom are fixed. Let Q_s denote the restoring force at degree of freedom s. The functional expression for Q_s is

$$Q_s = h(z_1, \cdots z_m, \cdots z_N), \tag{3-26}$$

where z_i, $i = 1, \cdots N$, are the structural deformations at the degrees of freedom i and h is a function of these deformations. Then λ_{sm} can be expressed as

$$\lambda_{sm} = \frac{\partial Q_s}{\partial z_m} \simeq \tag{3-27}$$

$$\simeq \frac{h(z_1, \cdots z_m + \Delta z_m +, \cdots + z_N) - h(z_1, \cdots z_m - \Delta z_m +, \cdots z_N)}{2\Delta z_m}.$$

The above formula provides the general expression for the equivalent stiffness matrix element for the nonlinear system. Note that if no inelastic permanent sets are allowed in the beam, then h is a linear function of the z_i, $i = 1, \cdots N$. When permanent sets occur in the beam then h is a complicated nonlinear form that depends on the displacement history of the structure. This was discussed in the previous section.

It is observed that λ_{sm}, $s,m = 1, \cdots N$ can be evaluated column by column rather than element by element if Equation (3-27) is used since the restoring forces are calculated in global form. In other words, the restoring forces are formed in such a way that all elements are evaluated in a vector, $\{R\}$, rather than a single element, Q_s. Let $\{\lambda_m\}$ denote the m^{th} column in the equivalent stiffness matrix. Then Equation (3-27) can be modified to yield

$$\{\lambda_m\} = \frac{\partial \{R\}}{\partial z_m} \simeq \tag{3-28}$$

$$\simeq \frac{\{H(z_1 \cdots z_m + \Delta z_m + \cdots + z_N)\} - \{H(z_1 \cdots z_m - \Delta z_m + \cdots z_N)\}}{2\Delta z_m},$$

where $\{R\} = \{H(z_1, \cdots z_N)\}$ is the restoring forces vector.

Another important property is that the equivalent stiffness matrix is time dependent if permanent sets occur in the beam. If the central difference method is used, the stiffness matrix at time t_j is evaluated based on the displacement responses at time t_j. Equation (3-28) then can be used to establish the equivalent stiffness matrix at time t_j. Once this is done, Equation (2-10), which characterizes the random component of response, can then be modified to

$$[m]\{\ddot{Z}\} + [c]\{\dot{Z}\} + [\lambda(\mu(t))]\{Z\} = \{F\} - [K]\{\mu\}, \tag{3-29}$$

where the definitions of $[\lambda(\mu(t))]$ and $[K]$ are the same as before, i.e.,

$$[k(z(t))] = E[[k(z(t))]] + [K] = [\lambda(\mu(t))] + [K]. \tag{3-30}$$

Note that the random component of stiffness matrix is assumed to be a matrix random variable. Using the central difference method Equation (3-29) can be solved for Z_{j+1}, such that

$$\{Z_{j+1}\} = [A_1]^{-1}([A_{2_j}]\{Z_j\} + [A_3]\{Z_{j-1}\} + \Delta t^2(\{F_j\} - [K]\{\mu_j\})), \tag{3-31}$$

where $[A_1]$ and $[A_3]$ are given by Equation (2-15) and $[A_{2_j}]$ is given by

$$[A_{2_j}] = 2[m] - [\lambda_j]\Delta t^2. \tag{3-32}$$

Note that the difference between Equations (3-31) and (2-19) is that the term $[A_2]$ is time dependent in Equation (3-31). The mean square responses can then be obtained in a manner similar to that used in Section 2.3 where Equation (2-22) was used. Here, though, $[A_2]$ is replaced by $[A_{2_j}]$.

$$
\begin{aligned}
E[\{Z_{j+1}\}\{Z_{j+1}\}^T] = \\
&= [A_1]^{-1}(\Delta t^4 E[\{F_j\}\{F_j\}^T] + \Delta t^4 E[[K]\{\mu_j\}\{\mu_j\}^T[K]^T] + \\
&+ [A_{2_j}]E[\{Z_j\}\{Z_j\}^T][A_{2_j}]^T + [A_3]E[\{Z_{j-1}\}\{Z_{j-1}\}^T][A_3]^T + \\
&+ [A_{2_j}]E[\{Z_j\}\{Z_{j-1}\}^T][A_3]^T + [A_3]E[\{Z_{j-1}\}\{Z_j\}^T][A_{2_j}]^T - \\
&- \Delta t^2 E[[K]\{\mu_j\}\{Z_j\}^T][A_{2_j}]^T - \Delta t^2[A_{2_j}]E[\{Z_j\}\{\mu_j\}^T[K]^T] - \\
&- \Delta t^2 E[[K]\{\mu_j\}\{Z_{j-1}\}^T][A_3]^T - \\
&- \Delta t^2[A_3]E[\{Z_{j-1}\}\{\mu_j\}^T[K]^T])[A_1]^{-T}. \qquad (3\text{-}33)
\end{aligned}
$$

Each term shown in the above equation can be solved using the same techniques as developed in Sections 2.3.2, 2.3.3 and 2.3.4 except in all cases $[A_2]$ is replaced by $[A_{2_j}]$. Equations (3-23) and (3-33) then provide the mean and mean square responses at time t_{j+1}.

Chapter 4

AUTOCORRELATION AND CROSS CORRELATION OF RESPONSE MEASURES

4.1 INTRODUCTION

In Chapter 2 the techniques of establishing the mean and mean square displacement responses for a linear system were developed. In Chapter 3, the case of the nonlinear system was discussed. Note, however, that only displacement response was considered in these chapters. Sometimes, it is desirable to know the mean characteristics not only of the displacement terms, but also of the velocity and acceleration and of the cross terms such as the products between displacements, velocities, and/or accelerations. Moreover, the correlation between Z_j and Z_{j+n} for $n \geq 1$ still needs to be evaluated when the response autocorrelation function is important. In this chapter, the moments and cross moments, as described above, are established.

4.2 VELOCITY AUTOCORRELATION

In this section, the mean and mean square velocity response are considered. Recall that the mean velocity can be expressed as in Equation (2-12b) when the central difference method is used,

$$\{\dot{\mu}_j\} = (\{\mu_{j+1}\} - \{\mu_{j-1}\})/(2\Delta t), \qquad j = 0, 1, 2, \cdots \tag{2-12b}$$

where $\{\mu_j\}$, $j = 0, 1, 2, \cdots$, is the mean displacement response vector at time t_j. The random component of the velocity can be expressed using the same expression:

$$\{V_j\} = (\{Z_{j+1}\} - \{Z_{j-1}\})/2\Delta t, \qquad j = 0, 1, 2, \cdots \tag{4-1}$$

where $\{V_j\}$, $j = 0, 1, 2, \cdots$, represents the random component of the velocity response vector at time t_j, and $\{Z_j\}$, $j = 0, 1, 2, \cdots$, represents the random component of the displacement response vector at time t_j. Accordingly, the mean square velocity can be evaluated by

$$\begin{aligned} E[\{V_j\}\{V_j\}^T] = \frac{1}{4\Delta t^2} (&E[\{Z_{j+1}\}\{Z_{j+1}\}^T] + E[\{Z_{j-1}\}\{Z_{j-1}\}^T] - \\ &- E[\{Z_{j+1}\}\{Z_{j-1}\}^T] - E[\{Z_{j-1}\}\{Z_{j+1}\}^T]). \\ &j = 0, 1, 2, \cdots \end{aligned} \tag{4-2}$$

The element in the r^{th} row and s^{th} column of $E[\{V_j\}\{V_j\}^T]$ is the correlation between the velocity at time t_j at the r^{th} degree of freedom and the velocity at time t_j at the s^{th} degree of freedom. In the above equation, note that $E[\{Z_{j-1}\}\{Z_{j+1}\}^T]$ is the transpose of $E[\{Z_{j+1}\}\{Z_{j-1}\}^T]$. In order to evaluate $E[\{Z_{j+1}\}\{Z_{j-1}\}^T]$, Equation (2-19) which represents the expression of $\{Z_{j+1}\}$ in terms of $\{Z_j\}$ and $\{Z_{j-1}\}$ must be used. Postmultiplying Equation (2-19) by $\{Z_{j-1}\}^T$ and then taking the expectation of the result yields

$$\begin{aligned} E[\{Z_{j+1}\}\{Z_{j-1}\}^T] = [A_1]^{-1}(&[A_2]E[\{Z_j\}\{Z_{j-1}\}^T] + \\ &+ [A_3]E[\{Z_{j-1}\}\{Z_{j-1}\}^T] - \Delta t^2 E[[K]\{\mu_j\}\{Z_{j-1}\}^T]). \end{aligned} \tag{4-3}$$

Each term in the above equation can be evaluated by using the techniques discussed in Section 2.3.3 and 2.3.4. Accordingly, Equation (4-3) is solvable. The other terms in Equation (4-2) are known. Therefore, the mean square velocity can be computed immediately when $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$ is evaluated.

Note that for nonlinear systems, the equivalent stiffness matrix, $[\lambda]$, and the coefficient matrix $[A_2]$ which is a function of $[\lambda]$, are denoted by $[\lambda_j]$ and $[A_{2_j}]$, respectively. The j subscripts reflect the fact that the stiffness is time dependent. Consequently, for the case of nonlinear analysis, $[A_{2_j}]$ is used to replace $[A_2]$ in Equations (4-2) and (4-3); namely

$$E[\{Z_{j+1}\}\{Z_{j-1}\}^T] = [A_1]^{-1}([A_{2_j}]E[\{Z_j\}\{Z_{j-1}\}^T] + \tag{4-3a}$$

$$+ [A_3]E[\{Z_{j-1}\}\{Z_{j-1}\}^T] - \Delta t^2 E[[K]\{\mu_j\}\{Z_{j-1}\}^T].$$

Equation (4-3a) can then be used to compute the nonlinear mean square velocity.

4.3 ACCELERATION AUTOCORRELATION

In this section the mean square acceleration is discussed. Using Equation (2-12a), the random component of the acceleration at time t_j can be expressed as

$$\{A_j\} = (\{Z_{j+1}\} - 2\{Z_j\} + \{Z_{j-1}\})/\Delta t^2, \qquad j = 0, 1, 2, \cdots \tag{4-4}$$

where $\{A_j\}$ represents the random component of the acceleration response vector at time t_j. The mean square acceleration can then be evaluated as

$$E[\{A_j\}\{A_j\}^T] = \frac{1}{\Delta t^4} (E[\{Z_{j+1}\}\{Z_{j+1}\}^T] + 4E[\{Z_j\}\{Z_j\}^T] +$$

$$+ E[\{Z_{j-1}\}\{Z_{j-1}\}^T] - 2E[\{Z_{j+1}\}\{Z_j\}^T] - 2E[\{Z_j\}\{Z_{j+1}\}^T] -$$

$$- 2E[\{Z_j\}\{Z_{j-1}\}^T] - 2E[\{Z_{j-1}\}\{Z_j\}^T] + E[\{Z_{j+1}\}\{Z_{j-1}\}^T] +$$

$$+ E[\{Z_{j-1}\}\{Z_{j+1}\}^T]). \qquad j = 0, 1, 2, \cdots \qquad (4\text{-}5)$$

The element in the r^{th} row and s^{th} column of $E[\{A_j\}\{A_j\}^T]$ is the correlation between the acceleration at time t_j at the r^{th} degree of freedom and the acceleration at time t_j at the s^{th} degree of freedom. Note that $E[\{Z_j\}\{Z_{j+1}\}^T]$, $E[\{Z_{j-1}\}\{Z_j\}^T]$, $E[\{Z_{j-1}\}\{Z_{j+1}\}^T]$ are the transposes of $E[\{Z_{j+1}\}\{Z_j\}^T]$, $E[\{Z_j\}\{Z_{j-1}\}^T]$, $E[\{Z_{j+1}\}\{Z_{j-1}\}^T]$, respectively. The terms $E[\{Z_{j+1}\}\{Z_j\}^T]$ and $E[\{Z_j\}\{Z_{j-1}\}^T]$ can be evaluated by using the techniques described in 2.3.4. The term $E[\{Z_{j+1}\}\{Z_{j-1}\}^T]$ can be evaluated by using Equation (4-3) as discussed in the previous section. The rest of the terms in Equation (4-5) are known if $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$ is currently being evaluated. Therefore, the mean square acceleration at time t_j can be computed immediately after $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$ is computed.

For the case of nonlinear analysis, the acceleration response moments can be evaluated in a manner similar to the mean square velocity discussed in the previous section. Hence, use of $[A_{2_j}]$ to replace $[A_2]$ is necessary for nonlinear problems.

4.4 VELOCITY AND DISPLACEMENT CROSSMOMENTS

In this section, the covariance between velocity and displacement is considered. The covariance between velocity and displacement can be obtained easily if Equation (4-1) is used. Postmultiply Equation (4-1) by $\{Z_j\}^T$ then take the expectation to obtain

$$E[\{V_j\}\{Z_j\}^T] = \frac{1}{2\Delta t} E[(\{Z_{j+1}\} - \{Z_{j-1}\})\{Z_j\}^T] =$$

$$= \frac{1}{2\Delta t} (E[\{Z_{j+1}\}\{Z_j\}^T] - E[\{Z_{j-1}\}\{Z_j\}^T]). \qquad (4\text{-}6)$$

The element in the r^{th} row and s^{th} column of $E[\{V_j\}\{Z_j\}^T]$ is the correlation between the velocity at time t_j at the r^{th} degree of freedom and the displacement at time t_j at the s^{th} degree of freedom. Each term of the above equation can be evaluated by using the technhiques discussed in 2.3.4. Since the random processes $\{V_j\}$ and $\{Z_j\}$ are both zero mean, Equation (4-6) actually represents the cross covariance between velocity and displacement at time t_j. If the cross covariance between velocity and displacement of the nonlinear response are desired, then the expressions for $\{V_j\}$ and $\{Z_j\}$ that reflect the nonlinear response variation must be used.

Note that $\{V_j\}$ is the derivative of $\{Z_j\}$. Theoretically, it can be shown that $\{V(t)\}$ and $\{Z(t)\}$ are orthogonal when evaluated at the same time if $\{Z(t)\}$ is weakly stationary. To prove this, let x(t) be a weakly stationary random process. Consider the following partial derivative.

$$\frac{\partial}{\partial \tau} E[x(t)x(t+\tau)] = \frac{\partial}{\partial \tau} \int_{-\infty}^{\infty} S_{xx}(\omega)e^{i\omega\tau}d\omega = \int_{-\infty}^{\infty} (i\omega)S_{xx}(\omega)e^{i\omega\tau}d\omega,$$

where $S_{xx}(\omega)$ is the spectral density of x(t). This representation is valid since it is assumed that x(t) is weakly stationary. Also, it is assumed that x(t) is differentiable in a mean square sense. The left hand side of the above equation can be simplified by rewriting the expression,

$$\frac{\partial(t+\tau)}{\partial\tau}\frac{\partial}{\partial(t+\tau)}E[x(t)x(t+\tau)] = E[x(t)\dot{x}(t+\tau)].$$

By letting $\tau \to 0$,

$$E[x(t)\dot{x}(t)] = i\int_{-\infty}^{\infty}\omega S_{xx}(\omega)d\omega = 0.$$

The right side is zero since $S_{xx}(\omega)$ is an even function.

Now consider Equation (4-6). If $\{Z(t)\}$ is weakly stationary, the right hand side of Equation (4-6) vanishes, because $E[\{Z_{j+1}\}\{Z_j\}^T]$ and $E[\{Z_{j-1}\}\{Z_j\}^T]$ are equal. Therefore, $E[\{V_j\}\{Z_j\}^T] = 0$ and the orthogonality property is satisfied for the computed response. Consequently, the central difference assumption used in this study leads to an orthogonality condition that matches the orthogonality condition which occurs in a theoretical weakly stationary random process. Note that this is true even for a non-linear system.

4.5 ACCELERATION AND DISPLACEMENT CROSSMOMENTS

In this section the covariance between the acceleration and displacement response measures is discussed. The approach is the same as described in the previous section. If Equation (4-4) is used to replace $\{A_j\}$ which represents the random component of acceleration, we have

$$E[\{A_j\}\{Z_j\}^T] = \frac{1}{\Delta t^2}(E[\{Z_{j+1}\}\{Z_j\}^T] - 2E[\{Z_j\}\{Z_j\}^T] + \\ + E[\{Z_j\}\{Z_{j-1}\}^T]). \tag{4-8}$$

The element in the r^{th} row and s^{th} column of $E[\{A_j\}\{Z_j\}^T]$ is the correlation between the acceleration at time t_j at the r^{th} degree of freedom and the displacement at time t_j at the s^{th} degree of freedom. Each term of the above equation possesses the same form as discussed before. Hence, by using techniques as described in Section 2.3.4 and Equation (4-3), $E[\{A_j\}\{Z_j\}^T]$ can be computed easily. Again, since the random processes $\{A_j\}$ and $\{Z_j\}$ are both mean zero, Equation (4-8) represents the covariance of acceleration and displacement at time t_j. It is clear that, using an approach similar to previous section, Equation (4-8) can be used to represent the solution either for the linear or the nonlinear system.

4.6 ACCELERATION AND VELOCITY CROSSMOMENTS

In this section, the covariance between acceleration and velocity is discussed. By using the same approach as in Section 4.5, the covariance between acceleration and velocity can be obtained immediately. If Equations (4-1) and (4-4) are used then

$$\begin{aligned} E[\{A_j\}\{V_j\}^T] = \frac{1}{2\Delta t^3} (&E[\{Z_{j+1}\}\{Z_{j+1}\}^T] - E[\{Z_{j+1}\}\{Z_{j-1}\}^T] - \\ &- 2E[\{Z_j\}\{Z_{j+1}\}^T] + 2E[\{Z_j\}\{Z_{j-1}\}^T] + E[\{Z_{j-1}\}\{Z_{j+1}\}^T] - \\ &- E[\{Z_{j-1}\}\{Z_{j-1}\}^T]). \end{aligned} \tag{4-9}$$

The element in the r^{th} row and s^{th} column of $E[\{A_j\}\{V_j\}^T]$ is the correlation between the acceleration at time t_j at the r^{th} degree of freedom and the velocity at time t_j at the s degree of freedom. Note that

$E[\{Z_{j-1}\}\{Z_{j+1}\}^T]$ is the transpose of $E[\{Z_{j+1}\}\{Z_{j-1}\}^T]$ which can be evaluated by using Equation (4-3). The terms $E[\{Z_j\}\{Z_{j-1}\}^T]$ and $E[\{Z_j\}\{Z_{j+1}\}^T]$ can be evaluated by using the techniques discussed in Section 2.3.4. Therefore, the covariance of acceleration and velocity can be computed immediately after $E[\{Z_{j+1}\}\{Z_{j+1}\}^T]$ is known.

It is already proved in Section 4.4 that a weakly stationary random process is orthogonal to its derivative when evaluated at the same time. This property can also be seen from Equation (4-9) if $\{Z(t)\}$ is weakly stationary since the right hand side of Equation (4-9) vanishes under weakly stationary conditions. Consequently, the central difference assumption satisfies the orthogonality condition for the acceleration and velocity. This is true even for the nonlinear problem.

4.7 THREE OR MORE TIME INCREMENTS

It was mentioned earlier that sometimes it is important to evaluate the term $E[\{Z_{j+n}\}\{Z_j\}^T]$, $n \geq 1$, in order to obtain the displacement response autocorrelation functions. This expression was established in several previous sections for, specifically, $n = 1, 2$. However, for the case $n \geq 3$, the problem still can be solved. To do this, $\{Z_{j+n}\}$ must be expressed in terms of the displacement response at previous times. For example, consider a linear system $\{Z_{j+n}\}$ can generally be expressed as:

$$\{Z_{j+n}\} = [A_1]^{-1}([A_2]\{Z_{j+n-1}\} + [A_3]\{Z_{j+n-2}\} + \Delta t^2\{F_{j+n-1}\} - \Delta t^2[K]\{\mu_{j+n-1}\}). \tag{4-10}$$

When this expression is postmultiplied by $\{Z_j\}^T$ and the expected value is taken, the result is

$$E[\{Z_{j+n}\}\{Z_j\}^T] = [A_1]^{-1}([A_2]E[\{Z_{j+n-1}\}\{Z_j\}^T] + \qquad (4\text{-}11)$$
$$+ [A_3]E[\{Z_{j+n-2}\}\{Z_j\}^T] - \Delta t^2 E[[K]\{\mu_{j+n-1}\}\{Z_j\}^T]).$$

Note that $E[\{F_{j+n-1}\}\{Z_j\}^T]$ vanishes since $\{F_{j+n-1}\}$ is independent of $\{Z_j\}^T$ and both are zero mean. The terms in (4-11) like $E[\{Z_{j+n-1}\}\{Z_j\}^T]$ and $E[\{Z_{j+n-2}\}\{Z_j\}^T]$ can be reduced further to terms involving the crossmoments $E[\{Z_{j+n-2}\}\{Z_j\}^T]$, $E[\{Z_{j+n-3}\}\{Z_j\}^T]$ and $E[\{Z_{j+n-4}\}\{Z_j\}^T]$. The reduction can be continued until only the terms $E[\{Z_{j+2}\}\{Z_j\}^T]$, $E[\{Z_{j+1}\}\{Z_j\}^T]$ and $E[\{Z_j\}\{Z_j\}^T]$ appear in the expression. Then the crossmoment $E[\{Z_{j+n}\}\{Z_j\}^T]$ can be evaluated. Obviously, it will take more computational time and large storage capacity to obtain the results, especially when n is large.

An important property which might be of interest is whether the response is weakly stationary. This can be determined using Equation (4-11). A random process is said to be weakly stationary if its mean is constant and its autocovariance function depends only on the time lag between response variables considered. In terms of Equation (4-11) this means that if

$$E[\{Z_{j+n}\}\{Z_j\}^T] = E[\{Z_{k+n}\}\{Z_k\}^T], \qquad (4\text{-}12)$$

for all j and k and for arbitrary n then the second part of the above requirement is satisfied. If, in addition, $E[\{Z_j\}]$ is a constant, then the random process $\{Z_j\}$ is weakly stationary.

Clearly, the requirements set forth here can never be exactly satisfied because the random process has deterministic initial conditions. However, when j become large the requirement in Equation (4-12) may be approximately satisfied for $n \ll j$. In such a situation the random process is said to approach a weakly stationary state.

Chapter 5

ENERGY DISSIPATION AND DAMAGE DIAGNOSIS

5.1 INTRODUCTION

There are two types of structural failure which are the result of the dynamic response of a stable structure. The first type of structural failure occurs when an extreme value of some measures of structural response, such as $z(t)_{max}$ or $\ddot{z}(t)_{max}$, reaches an upper bound level or a lower bound level. The second type of structural failure occurs when the accumulated damage reaches a fixed level such that the structure diminishes in strength or resistance and a response that causes failure is realized. In many situations, the accumulated damage may lead to structural failure even when the input and response are of short duration. In a practical sense, the true criterion of structural failure may depend on both peak response and damage accumulation. However, at the present time there exists no universal measure of structural damage. Consequently, a postulated upper or lower bound for extreme responses, or a fixed level for accumulated damage is used in many applications.

Many studies [3, 34, 35] have considered the potential for the first type failure described above by using the threshold-crossing and the peak-distribution to establish the reliability of structural systems. It can be shown that the techniques discussed in Chapters 2 through 4 can be used to develop the joint probability for displacement and velocity so that the first passage problem can be solved. In the present study, only the second type of structural failure is considered. The cyclic damage may be defined by the cyclic permanent sets such as Miner [36], Coffin [37], and Ju and Yao [38], or the cyclic energy dissipation [39]. The criterion in the present chapter postulates that the accumulated damage is related to the energy dissipation in the system during the response.

5.2 ENERGY DISSIPATED RELATED TO DAMAGE PROBLEMS

Experimental investigations [34] have shown that the energy dissipated by a structural system due to load cycling is related to the residual strength of the structural material. The energy dissipation may be classified in two parts, namely, energy dissipated in the spring and energy dissipated in the damper. For a linear system, energy dissipation occurs only in the damping element because no energy is dissipated by the spring. However, in many situations, when the spring behaves nonlinearly, a hysteresis loop is formed in the material stress-strain curve during the response motion. The energy dissipated in the spring can then be defined by integrating the stress-strain curve.

Consider a single element system with uniaxial load. The total energy dissipated in the system can then be expressed

$$E_t = \int_0^\infty [c\dot{z} + R(z)]dz, \tag{5-1}$$

where z is the displacement response across the element, c is the element viscous damping and R(z) represents the restoring force in the element spring. First, consider the energy dissipated in the spring. If the single-element system referred to Equation (5-1) is assumed to represent a small element in a beam of a structural frame, the energy dissipated for this small element due to material nonlinearity can be expressed as

$$E_s = \Delta V \int_0^{\infty} R(\varepsilon) d\varepsilon, \tag{5-2}$$

where ΔV is the volume of the small element in a beam under consideration. Note that $R(\varepsilon)$ can be expressed by (see Figure 3.3)

$$R(\varepsilon) = E(\varepsilon(t) - \varepsilon_o(t)), \tag{5-3}$$

where E is Young's Modulus and $\varepsilon(t)$, $\varepsilon_o(t)$ are the total strain and permanent sets at time t, respectively. Substitution of Equation (5-3) into Equation (5-2) yields

$$E_s = \Delta V\ E \int_0^{\infty} (\varepsilon(t) - \varepsilon_o(t)) d\varepsilon. \tag{5-4}$$

Equation (5-4) deterministically defines the energy dissipation for a small element in a beam.

In the present study, the governing equation of motion of a nonlinear system is decomposed into two equations, namely Equations (3-22) and (3-29),

$$[m]\{\ddot{\mu}\} + [c]\{\dot{\mu}\} + \{R(\mu)\} = \{\phi\}, \tag{3-22}$$

$$[m]\{\ddot{Z}\} + [c]\{\dot{Z}\} + [\lambda(\mu(t))]\{Z\} = \{F\} - [k]\{\mu\}. \tag{3-29}$$

It will be illustrated in Chapter 7, that Equation (3-22) governs the mean to within 99 percent accuracy and Equation (3-29) characterizes the random component of the response within 93 percent accuracy. Let the total energy dissipation be decomposed into a mean and a fluctuating portion. The mean energy dissipation due to the material nonlinearity can then be evaluated based on Equation (3-22) which represents the mean characteristics of the response. The energy dissipation obtained from Equation (3-29) then represents the fluctuating portion of energy dissipation.

However, note that the Equations (3-22) and (3-29) are derived in such a way that the displacement, velocity and acceleration responses are represented only at the nodal points of the structure. In the present analysis of a structural frame system, the energy dissipation is computed along the beam rather than at the nodal points. Consequently, the fluctuation of energy dissipation can not be obtained directly from Equation (3-29) which governs the random component of the response. Therefore, in the present investigation, only the energy dissipation related to the mean response is considered. The energy dissipation, as computed from Equation (3-22), represents the mean energy dissipation. When Equation (5-4) is used to compute the mean energy dissipation for a small element in a beam, the total energy dissipation for a beam due to material nonlinearity can be obtained by summing up all the small elements. This can be stated mathematically in the following form.

$$
\begin{aligned}
E_v &= \sum_{\Delta V} E_s = \\
&= \sum_{\Delta V} \Delta V \, E \sum_{j} (\varepsilon(j\Delta t) - \varepsilon_o(j\Delta t)) \, (\varepsilon((j+1)\Delta t) - \varepsilon(j\Delta t)), \qquad (5\text{-}5)
\end{aligned}
$$

where E_v is the total energy dissipated in a beam. Equation (5-5) can be used as a measure of energy dissipation in a beam due to material nonlinearity.

Next, consider the energy dissipated due to damping. Let E_d denote the energy dissipated due to viscous damping in a single element,

$$E_d = c \int_0^\infty \dot{z} dz. \tag{5-6}$$

The integral can be transformed to

$$E_d = c \int_0^\infty \dot{z}^2 dt. \tag{5-7}$$

Under the assumption of response decomposition, we have

$$\dot{z} = \dot{\mu} + \dot{Z}. \tag{5-8}$$

Substitution of Equation (5-8) into Equation (5-7) yields

$$E_d = c \int_0^\infty (\dot{\mu}^2 + 2\dot{\mu}\dot{Z} + \dot{Z}^2) dt. \tag{5-9}$$

Taking the expectation on both sides of Equation (5-9), results in

$$E[E_d] = c \int_0^\infty \dot{\mu}^2 dt + c \int_0^\infty E[\dot{Z}^2] dt. \tag{5-10}$$

Use has been made of the fact that $\dot{\mu}$ is deterministic and $\dot{Z}$ is mean zero. The mean energy dissipated in the damper is clearly separated into two parts. The first term on the right side in Equation (5-10) represents the energy dissipation from mean velocity respopnse, the second term represents the energy dissipation from the variance of the velocity response. Note that the velocity autocorrelation of the response was developed in Section 4.2.

The mean energy dissipated in each element can be obtained using Equation (5-10). When the energy dissipation components are summed the total energy dissipated due to damping is obtained. The total mean energy dissipated in the beam can then be obtained as

$$E_t = E_d + E_s. \tag{5-11}$$

Equation (5-11) represents the mean energy dissipation measure. The mean square energy dissipation will include the cross term between spring and damper, and the square terms of spring and damper. However, as stated before, the Equation (3-29) can not be used to obtain the random component of the energy dissipation. The cross moments between damper and spring would need to be specified to execute the mean square energy dissipation analysis. Consequently, the mean square energy dissipation is very difficult to obtain.

The result from Equation (5-11) can be used to predict an upper bound of the energy dissipation measure if the Markov inequality is considerd [40].

$$P\{X \geq a\} \leq \frac{E[X]}{a}, \tag{5-12}$$

where a is a non-negative constant. In view of this, the above results obtained for mean energy dissipation can be used to make a probabilistic statement about damage; that is, X is the cumulative energy dissipation and a is the quarter cycle energy dissipation.

Numerical examples presented in Chapter 6 show the results of mean energy dissipation computations.

Chapter 6

NUMERICAL EXAMPLES

6.1 INTRODUCTION

In the previous chapters methods of establishing mean and mean square displacement responses for both linear and nonlinear systems were developed. The methods of computing the cross correlation between displacement, velocity and acceleration were also established. The equations developed to evaluate the statistical response properties (such as Equations (2-10), (2-22), etc.,) are all recursion relationships and can be computed directly by numerical methods. The capability to obtain solution moments of a random differential equation with nonlinear and random coefficients in terms of recursion equations, makes the methods discussed in the previous several chapters important. A computer code called FEDRANS (Finite Element Dynamic and Random Analysis for Nonlinear System) was programmed and applied to solve the equations in Chapters 2 through 5.

It is noted that for the problems involving nonlinear systems, a specially discretized beam is necessary in order to evaluate the permanent sets in Equations (3-13), (3-14) and (3-15). In the computer program FEDRANS, a beam is divided into 10 segments in the longitudinal direction and 10 layers through the thickness. A configuration of such beam is shown in Figure 6.1.

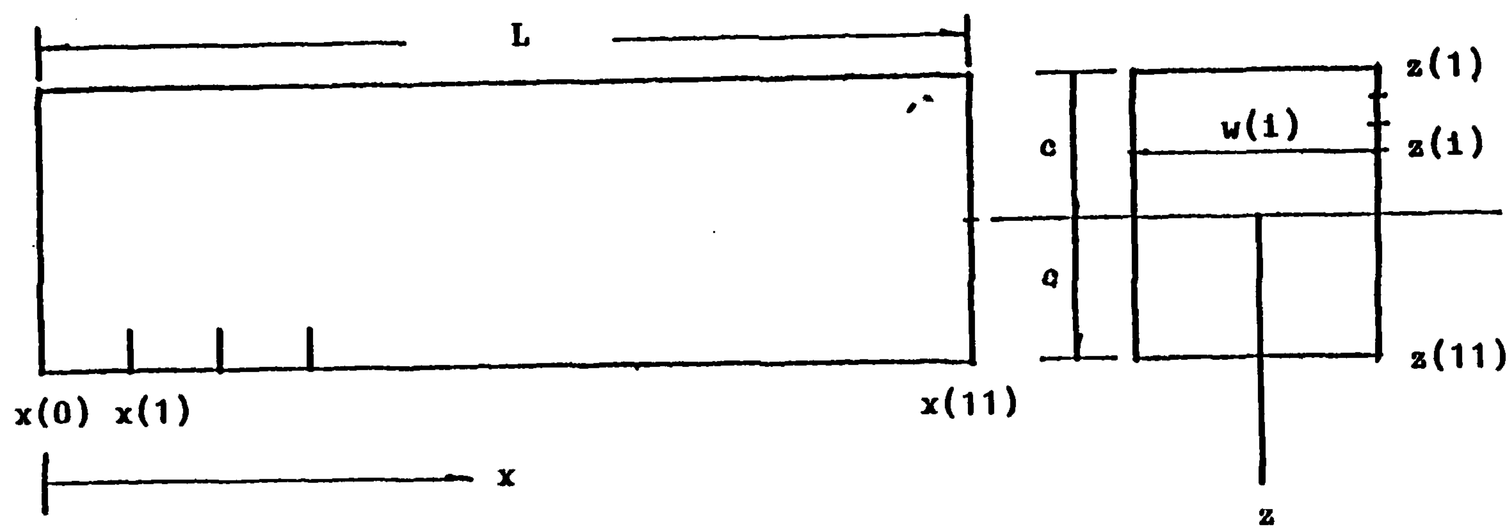

Figure 6.1 A configuration of a spatially discrete beam.

In Figure 6.1, along the longitudinal axis, x, the points where the beam is segmented are identified by the equi-spaced coordinate values x(i), i = 1, ··· 11. The beam is layered in the z direction in the same way, except that the layers may not have equal thickness. The w(i), i = 1, ··· 11 represent the widths at x(i) and z(i) locations. Note that x(0) = 0 and x(11) = L where L is the beam length as shown in Figure 6.1.

The integration is numerically computed with the trapezoid rule;

$$\int_{-c}^{c} \varepsilon_o(u,z) z w(z) dz \simeq$$

$$\simeq \sum_{i=2}^{10} \varepsilon_o(j,i) z(i) w(i) \Delta z + \frac{1}{2} (\varepsilon_o(j,1) + \varepsilon_o(j,11)) \Delta z = h_1(j), \qquad (6\text{-}1)$$

$$\int_{0}^{L} du \int_{-c}^{c} \varepsilon_o(u,z) z w(z) dz \simeq$$

$$\simeq \left\{ \begin{array}{ll} \sum_{i=2}^{j} h_1(i) \Delta x + \frac{1}{2} h_1(1) \Delta x & \text{for } j < 11 \\ \sum_{i=2}^{11} h_1(i) \Delta x + \frac{1}{2} (h_1(1) + h_1(11)) \Delta x & \text{for } j = 11 \end{array} \right. = h_2(j) \qquad (6\text{-}2)$$

$$\int_{0}^{L} dv \int_{0}^{v} du \int_{-c}^{c} \varepsilon_o(u,z) z w(z) dz \simeq$$

$$\simeq \sum_{j=2}^{10} h_2(j) \Delta x + \frac{1}{2} (h_2(1) + h_2(11)) \Delta x, \qquad (6\text{-}3)$$

where $\Delta x = L/10$ and $\Delta z = c/5$ and c is the half depth of the beam. Similarly,

$$\int_{-c}^{c} \varepsilon_o(u,z)w(z)dz \approx$$

$$\approx \sum_{i=2}^{10} \varepsilon_o(j,1)w(i)\Delta z + \frac{1}{2}(\varepsilon_o(j,1) + \varepsilon(j,11))\Delta z = h_3(j), \tag{6-4}$$

$$\int_{0}^{L} du \int_{-c}^{c} \varepsilon_o(u,z)w(z)dz \approx$$

$$\approx \sum_{j=2}^{10} h_3(j)\Delta x + \frac{1}{2}(h_3(1) + h_3(11))\Delta x. \tag{6-5}$$

Equations (6-1) through (6-5) are the approximate expression of the integrals in Chapter 3.

In Chapter 2 the eigenvectors for the vibrating system were used to represent the displacement response in the development of a recurrence relation. An iteration scheme was used to obtain the eigenvectors for the vibrating system. The displacement vector can be developed by superposing suitable amplitudes of the modes of free vibration. Since, the higher frequencies usually do not significantly contribute to the displacement response. It is sufficient to evaluate the lowest p modal frequencies rather than the entire collection of frequencies. Some experiments in the present study have shown that by choosing p = 3 it is possible to obtain good accuracy even for large structural systems. The method of subspace iteration was used because of its efficiency in obtaining the desired eigenvectors. A comprehensive discussion on the subspace iteration technique is given in reference [41].

When only one mode is used to approximate the response of the vibrating system, the method of inverse iteration can be used. Also, the highest modal frequency, ω_{max}, sometimes needs to be evaluated since the largest time increment Δt can not be chosen greater than $T_{min}/4$ where $T_{min} = 2\pi/\omega_{max}$. (Otherwise, stability problems will arise in using the central difference method). The method of forward iteration can be used here to obtain the highest frequency ω_{max}. Both the inverse and forward iteration methods are described in reference [41].

In the present analysis, the lumped mass approximation is used. Consider the uniform beam with six degrees-of-freedom as shown in Figure 6.2.

The mass values (elements in the diagonal element mass matrix) at the degrees-of-freedom i, i = 1, $\cdots$ 6 are

$$\begin{aligned} m(1) &= m(4) = 0.5\ \rho AL \\ m(2) &= m(5) = 0.5\ \rho AL \\ m(3) &= m(6) = 1/24\ \rho AL^3 \end{aligned} \tag{6-6}$$

where ρ, A, L are the mass density, cross sectional area and member length, respectively.

In order that the damped system is to possess normal mode oscillation, the damping in the present analysis is assumed to be Rayleigh damping; that is

$$[c] = \alpha[m] + \gamma[\lambda] \tag{6-7}$$

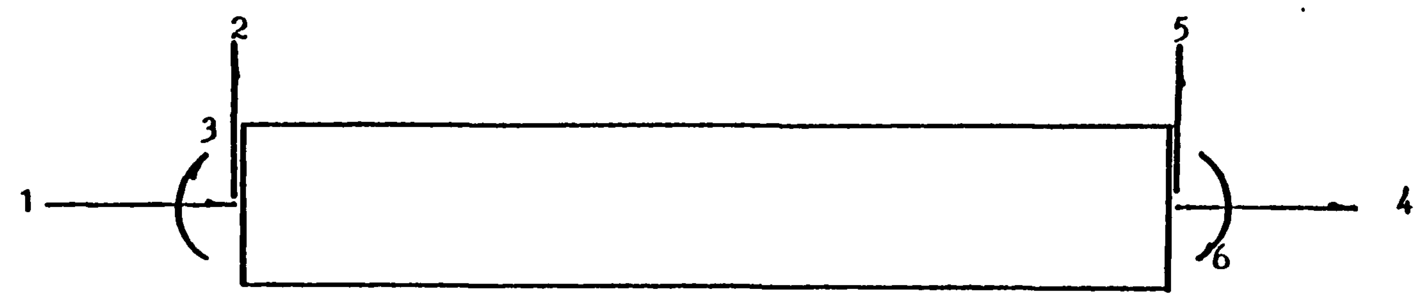

FIGURE 6.2 Uniform beam element with six degree of freedom.

where [c], [m], [λ] are the damping, mass, and mean linear stiffness matrices, respectively. The coefficients α, γ are constant coefficients that establish the damping. Note that the damping matrix is assumed to be a constant matrix even though the stiffness matrix behaves nonlinearly.

6.2 MATRIX ALGEBRA

The equations established to compute the statistics of structural response (such as Equations (2-22), (2-32), etc.,) involve matrix multiplication. In FEDRANS the matrices are established in such a way that only non-zero elements are stored. For instance, the stiffness matrix is assembled and stored in an NxH rectangular matrix rather than an NxN square matrix, where N is the total number degree-of-freedoms and H is the half bandwidth of the stiffness matrix. In view of this, when two matrices are multiplied together, the regular matrix multiplication needs to be modified.

Consider the following element in a matrix multiplication:

$$U_r = \sum_{s=1}^{N} Q_{r,s} V_s, \tag{6-8}$$

where [Q] is an NxN matrix and {V} is an Nx1 vector. The resulting matrix is {U}, which is Nx1. Let the matrix [Q] be stored in [W] which is NxH. The transformation between these two matrices can be defined as

$$\begin{aligned} W_{r,s-r+1} &= Q_{r,s} \qquad \text{for } s-r+1 > 0 \\ W_{s,r-s+1} &= Q_{s,r}. \qquad \text{for } s-r+1 \le 0 \end{aligned} \tag{6-9}$$

Since [Q] is symmetric. Substituting Equation (6-9) into Equation (6-8) yields the result:

$$U_r = \sum_{s=p}^{q} (W_{r,s-r+1}V_s + W_{s,r-s+1}V_s), \qquad (6\text{-}10)$$

where p and q are the lower and upper limits to be determined. Note that the first term on the right side in (6-10) exists only when $s-r+1 > 0$, and the second term exists only when $s-r+1 \leq 0$. Considering (6-9), p and q can be expressed

$$p = \begin{cases} 1 & \text{if} \quad r-H \leq 0 \\ r-H+1 & \text{if} \quad r-H \geq 1, \end{cases} \qquad (6\text{-}11)$$

$$q = \begin{cases} r+H-1 & \text{if} \quad q < N \\ N & \text{if} \quad q \geq N. \end{cases} \qquad (6\text{-}12)$$

Equations (6-10), (6-11) and (6-12) completely define the operation of matrix multiplication. Similarly, Gaussian elimination can be modified by an analogous approach.

6.3 EXAMPLE ONE

The first example considered is a single-degree-freedom structural system, shown in Figure 6.3(a). At modal excitation, the lateral motion of the beam is defined by its end motion z(t).

The beam cross sectional dimensions are defined as in Figure 6.3(b). The values of these cross sectional dimensions and the material properties are given in Table 6.1.

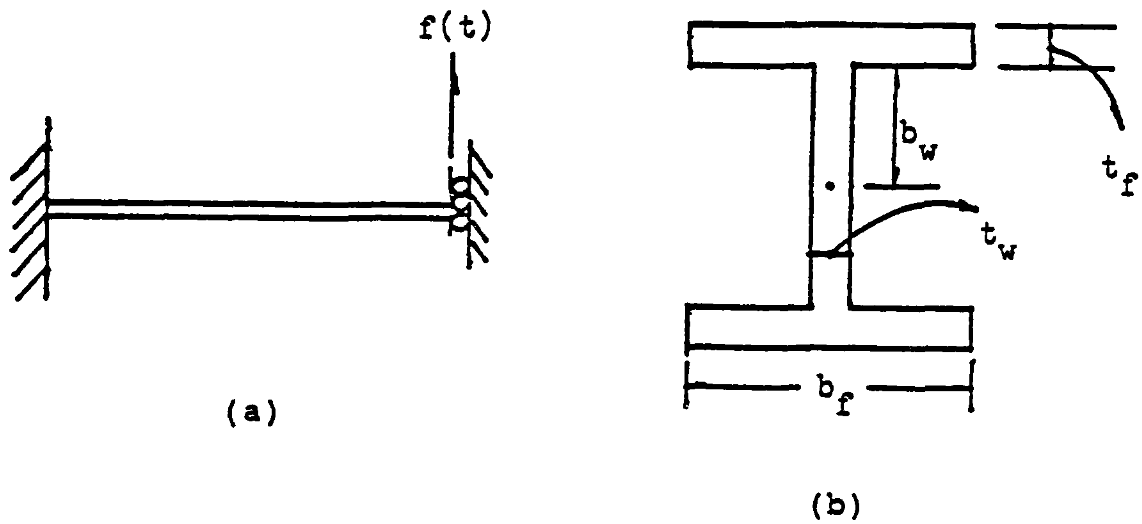

Figure 6.3 The single-degree-of-freedom system for Example 1

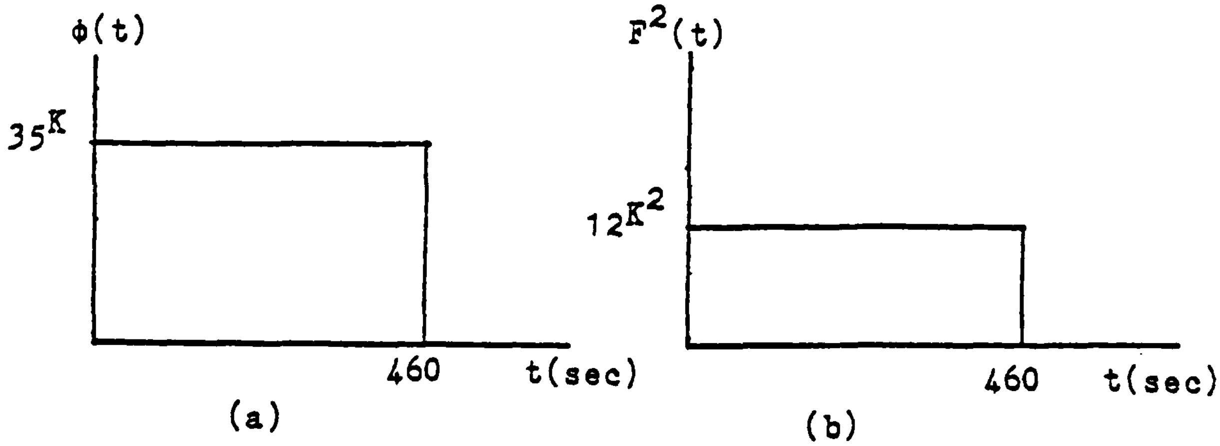

Figure 6.4 Mean and autocovariance of the input forcing function for Example 1

	b_f	t_f	b_w	t_w	E	σ_y	ν	ρ	α	γ	L
unit	in	in	in	in	ksi	ksi		$\frac{Ksec^2}{in^4}$			in
	10.	.56	4.43	.34	29000	40	.25	.2836	.1	0	100

Table 6.1 The cross sectional dimensions and the material properties for Example 1.

E is the mean value of Young's Modulus; σ_y is the yield stress; ν is the Possion's ratio; ρ is the mass density; α and γ are the coefficients of Rayleigh damping. The natural frequency, ω_n, and damping ratio, ζ, for system were computed from the above data. The results are

$$\omega_n = 0.68086 \text{ rad/sec}$$

$$\zeta = 0.0734$$

The mean and autocovariance of the input forcing function are graphically presented in Figure 6.4.

The mean and autocovariance of the response, which are given by Equations (2-14) and (2-22) for the linear system, Equations (3-23) and (3-33) for the nonlinear system, can then be computed using the data given above. In the linear problem solution the yield stress σ_y is assumed to be infinite. The mean and autocovariance of the response for both linear and nonlinear systems were determined using the computer program FEDRANS. There are two cases considered here for different values of β, the coefficient of variation of Young's modulus. (The coefficient of variation is the ratio of the standard deviation and the mean of a random variable. See Equation (2-27).) Figures 6.5 through 6.8 show the results for $\beta = 0$, and Figures 6.9 through 6.10 show the results for $\beta = 0.1$. Figures 6.5 and 6.6 show the mean structural responses for the linear and nonlinear systems, respectively, for the excitation shown in Figure 6.4. Figures 6.7 and 6.8 show the variance of the structural response for the linear and nonlinear systems, respectively, for the excitation in Figure 6.4. In both cases the stiffness is deterministic, i.e., $\beta = 0$.

When the stiffness is random the mean response is approximately equal to the mean response of the system whose stiffness is deterministic. However, their variances of the responses are different. Figures 6.9 and 6.10 show the variance of the structural responses for the linear and nonlinear systems, respectively, for the excitation shown in Figure 6.4 and for $\beta = 0.1$. The response variances are greater in both cases, when the structural stiffness are random than when the structural stiffnesses are deterministic.

In order to compare the results of this anlysis with known results, consider the response of a linear structure with $\beta = 0$, that is, the stiffness term is deterministic. In such a case, Equation (2-10) can be solved in closed form when the input is white noise with constant spectral density, S_o. The result is [33]

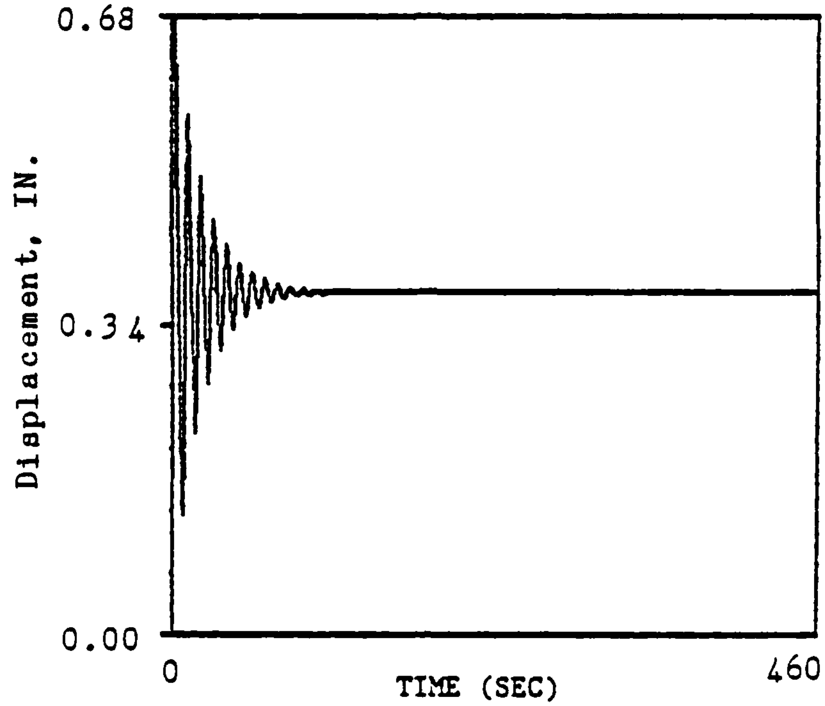

Figure 6.5 Mean of linear displacement response for Example 1

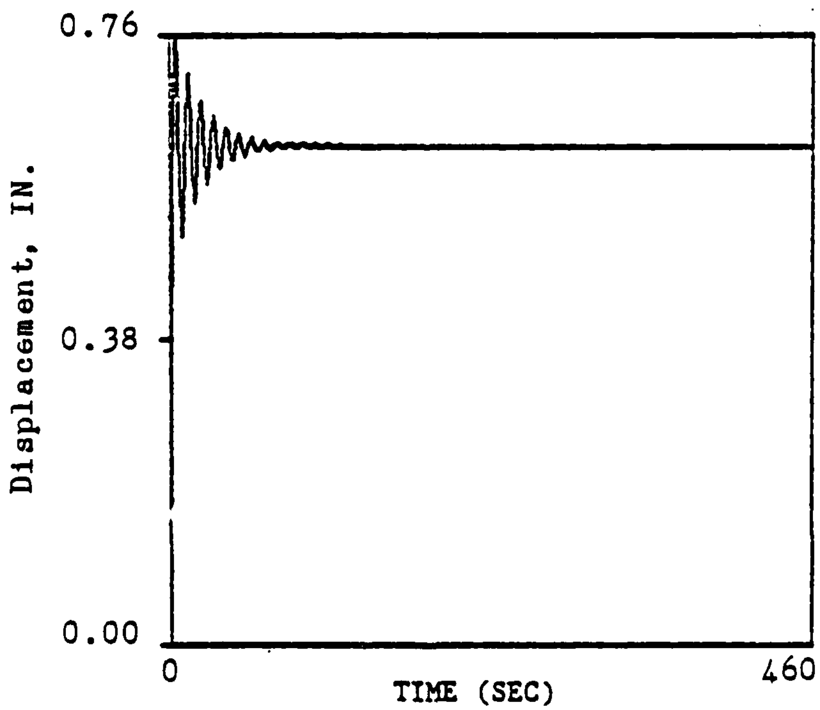

Figure 6.6 Mean of nonlinear displacement response for Example 1

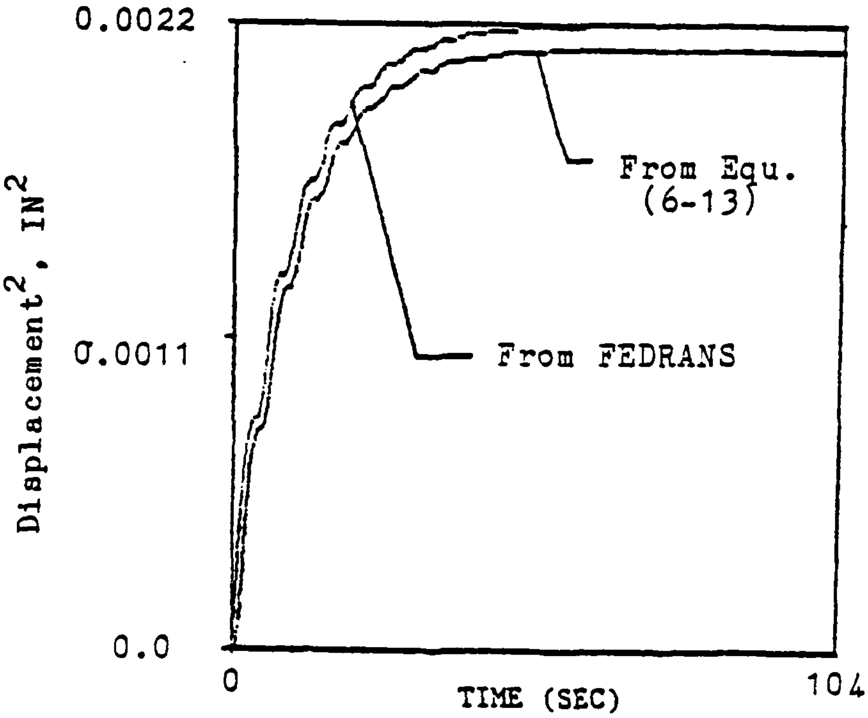

Figure 6.7 Variance of linear displacement response for coefficient of variation $\beta = 0$

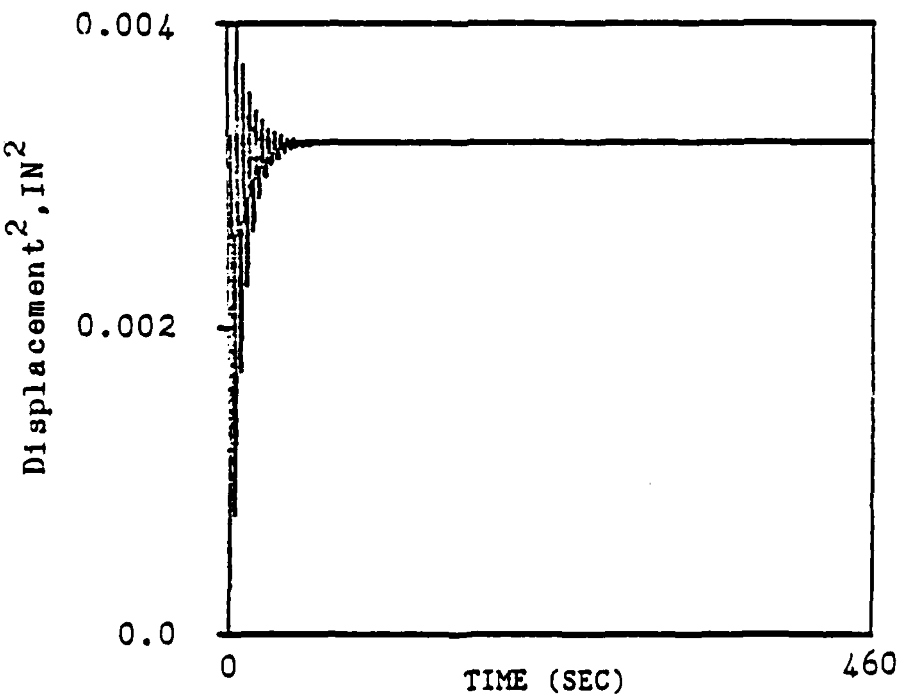

Figure 6.8 Variance of nonlinear displacement response for coefficient of variation $\beta = 0$

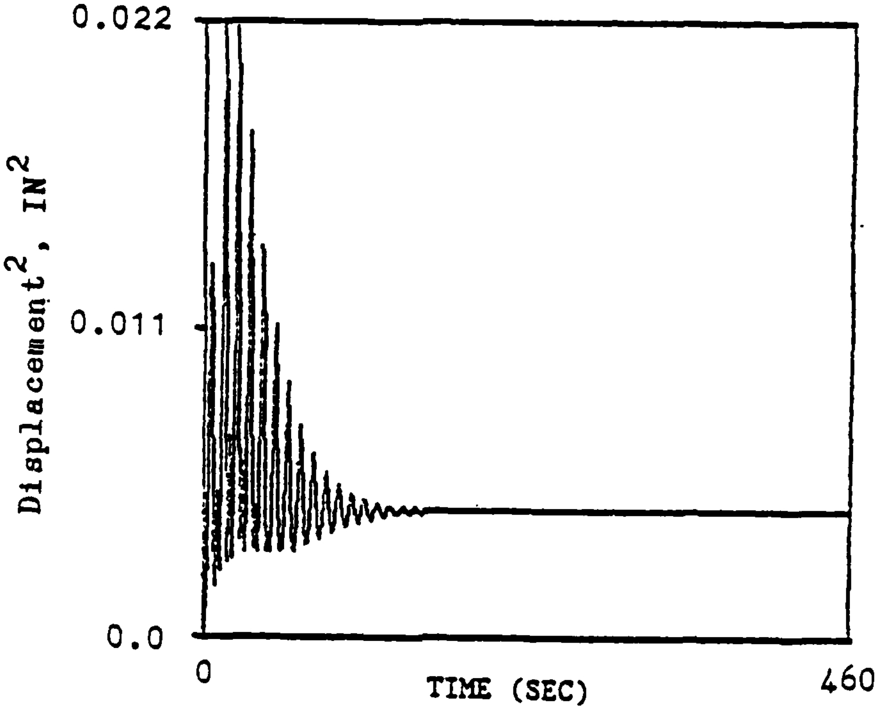

Figure 6.9 Variance of linear displacement response for coefficient of variation β = 0.1

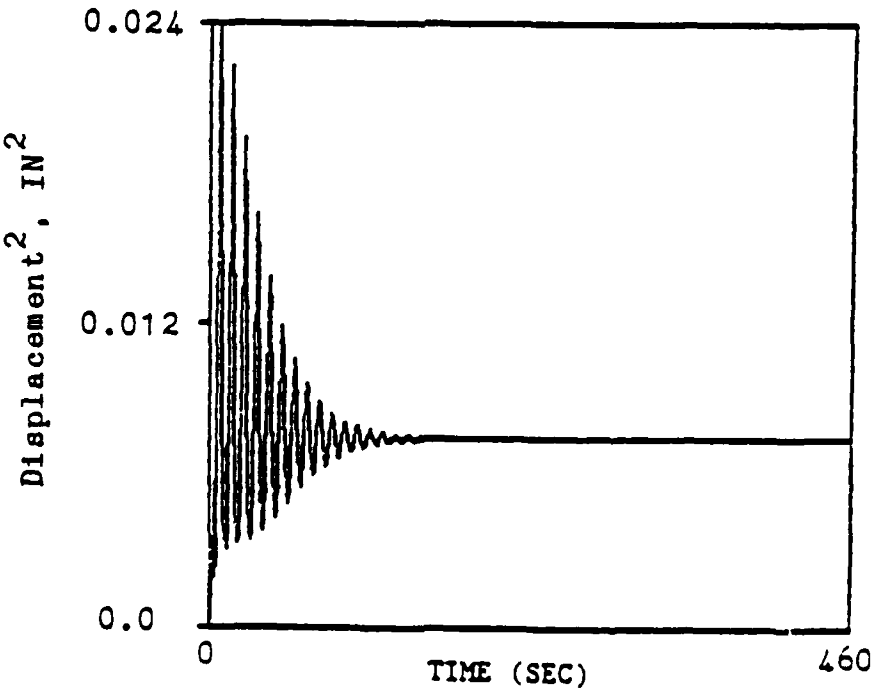

Figure 6.10 Variance of nonlinear displacement response for coefficient of variation β = 0.1

$$E[Z^2(t)] = \frac{\pi S_o}{2\zeta\omega^3 m^2}\{1 - \frac{\exp(-2\zeta\omega_n t)}{\omega_d^2}[\omega_d + 2(\zeta\omega_n \sin\omega_d t)^2 + \zeta\omega_n\omega_d \sin 2\omega_d t]\} \tag{6-13}$$

where ω_d is the damped frequency. The spectral density S_o can be determined in the following manner. Note that the mean square input is assumed to be a constant as shown in Figure 6.4(b). The constant mean square satisfies the property of white noise

$$E[F(t)F(s)] = R_{FF}(t-s) = 2\pi S_o \delta(\tau), \quad \tau = t-s$$

$$S_{FF}(\omega) = S_o \tag{6-14}$$

where $S_{FF}(\omega)$ and $R_{FF}(\tau)$ are the spectral density and the autocorrelation function for the input, respectively. In order to obtain the value S_o, consider a discrete time representation for the delta function, $\delta(\tau)$, as shown in Figure 6.11.

The ideal white noise can be obtained as $\Delta\tau \rightarrow 0$. However, in practical analysis, $\Delta\tau$ is chosen as a small value which never goes to zero. As a result, the signal which is represented as a band-limited white noise. The amplitude $\Delta\tau^{-1}$ remains finite for band-limited white noise. In such a case, the mean square value is evaluated at $\tau = 0$ and the result is

$$E[F^2(t)] = \frac{2\pi S_o}{\Delta\tau} \tag{6-15a}$$

$$S_o = \frac{E[F^2(t)]\Delta\tau}{2\pi} \tag{6-15b}$$

where $E[F^2(t)] = 12(k^2)$ as shown in Figure 6.4. (This relation can also be derived using frequency domain arguments and the graph of the spectral density of band-limited white noise.)

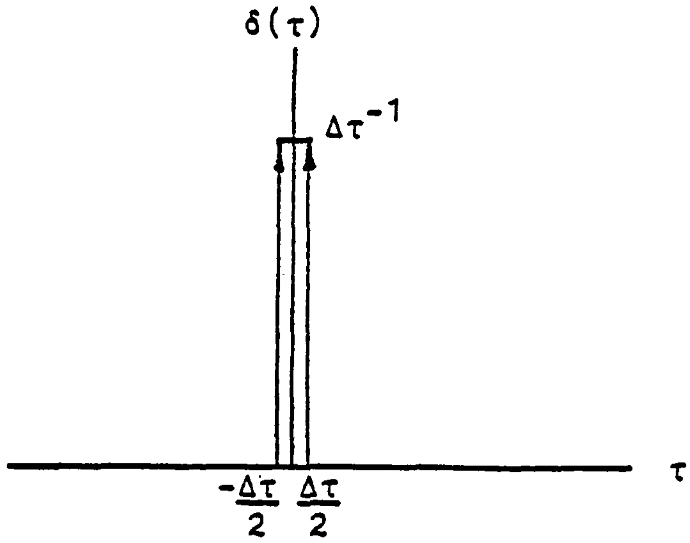

Figure 6.11 Delta function

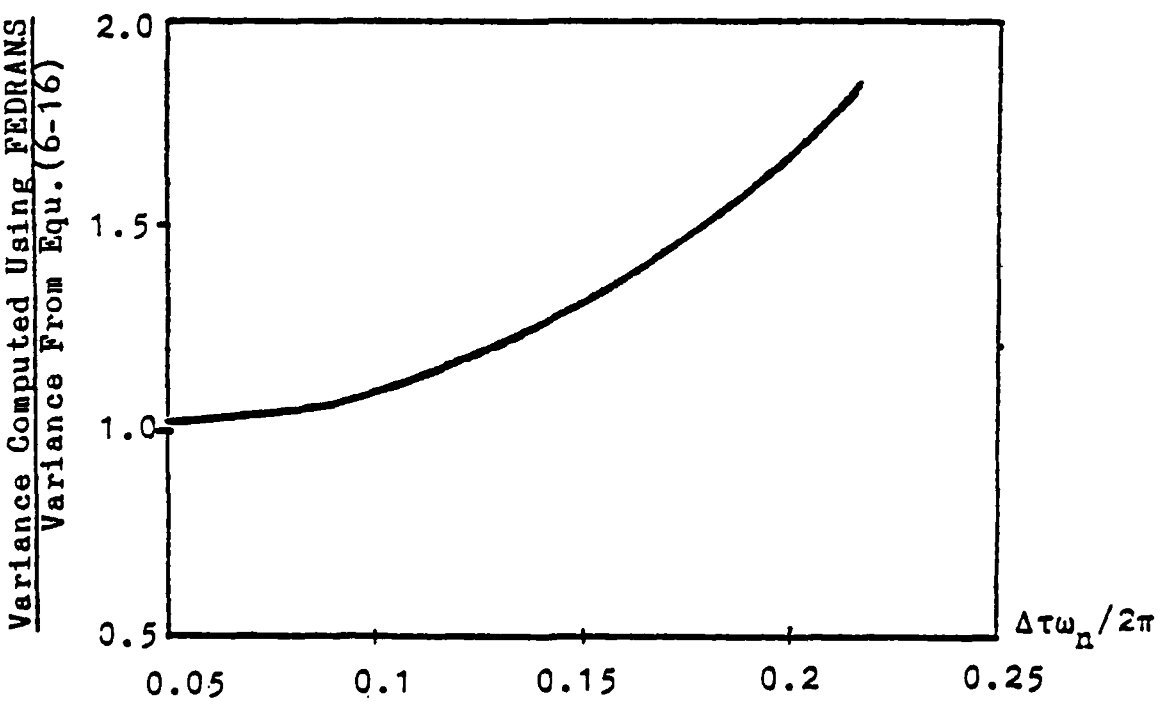

Figure 6.12 Comparisons of stationary displacement response variances for different small time increments, $\Delta\tau$

After the value S_o is obtained, Equation (6-13) can be evaluated. The comparison between Equation (6-13) and the corresponding result obtained using the formulas developed in this study is plotted in Figure 6.7. The agreement between these results is good. The evaluation of approximation in using FEDRANS as compared to the exact values can be made for various values of $\Delta\tau$ more readily at stationary values. The stationary response can be evaluated as $t\to\infty$ in Equation (6-13), that is

$$E[Z^2(t)] = \frac{E[F^2(t)]\Delta\tau}{2\zeta\omega_n^3 m^2}. \tag{6-16}$$

The ratio of FEDRANS approximation to the theoretical value (6-16) is shown in Figure 6.12 for several values of $\Delta\tau$ (normalized by the period of the single-degree-of-freedom system). When $\Delta\tau$ is small the ratio is near one and the error is very small. As $\Delta\tau$ increases the error increases. Specifically, the computed variance becomes greater than the theoretical variance. It can be established that the error is less than 8 percent when $\Delta\tau\omega_n/2\pi$ is less than 9 percent.

The variance of velocity and acceleration were also computed for the linear structural response. The results are plotted in Figures 6.13 and 6.14 for the case, $\beta = 0.1$. The cross correlations between displacement and velocity, between displacement and acceleration, and between velocity and acceleration were also computed and the results are plotted in Figures 6.15, 6.16 and 6.17. Note that E[VZ] and E[AV] approach zero after five or six response cycles. Also note that E[AZ] is negative as might be anticipated. The variances and correlations of displacement, velocity and acceleration for nonlinear system response were also computed. The results are plotted

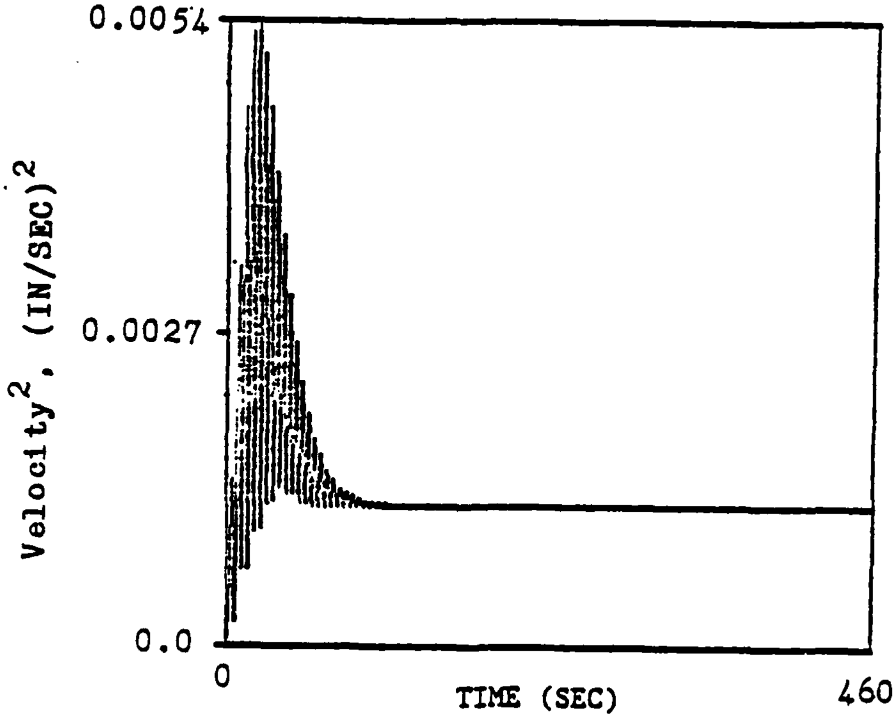

Figure 6.13 Variance of linear velocity response for Example 1, β = 0.1

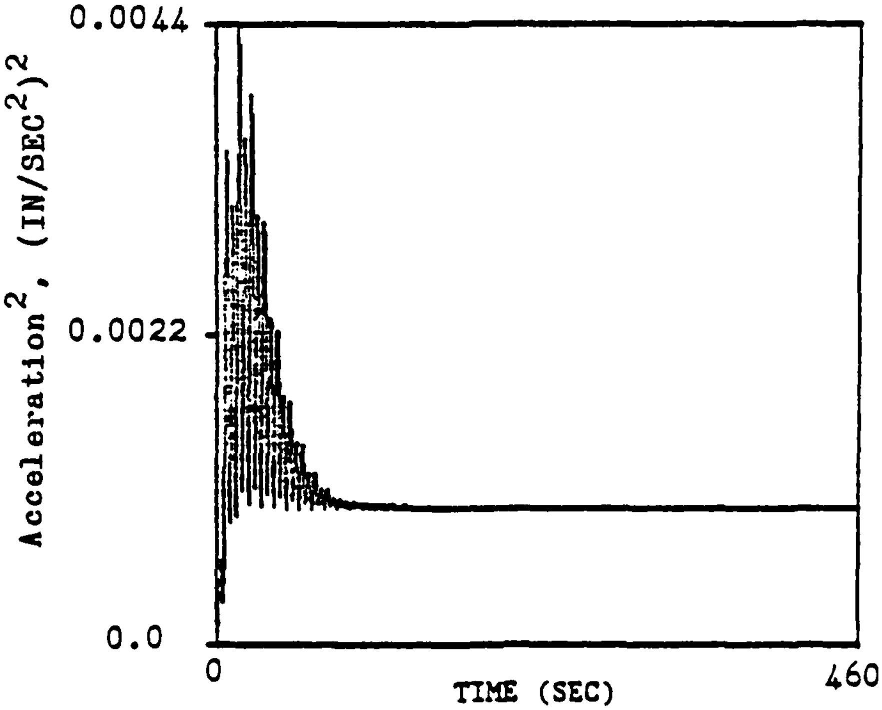

Figure 6.14 Variance of linear acceleration response for Example 1, β = 0.1

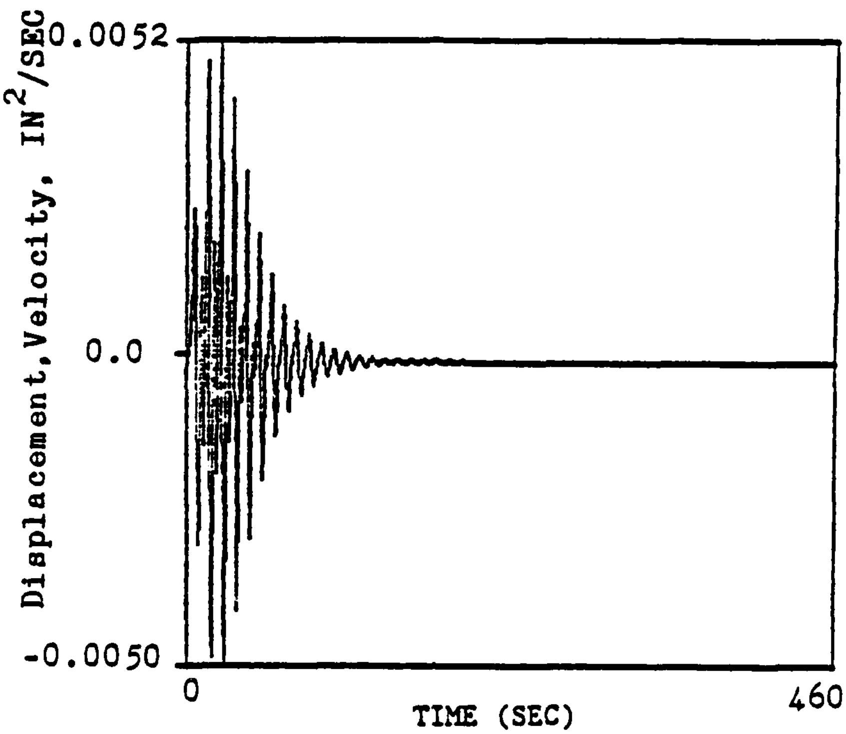

Figure 6.15 Crossmoment between linear velocity and displacement for Example 1, β = 0.1

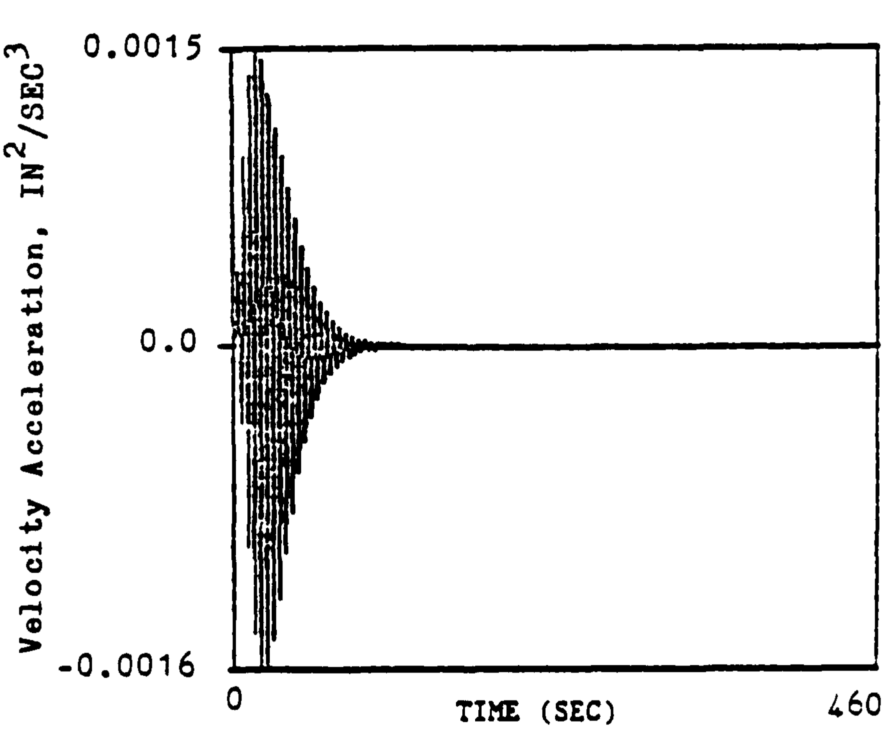

Figure 6.16 Crossmoment between linear acceleration and velocity for Example 1, β = 0.1

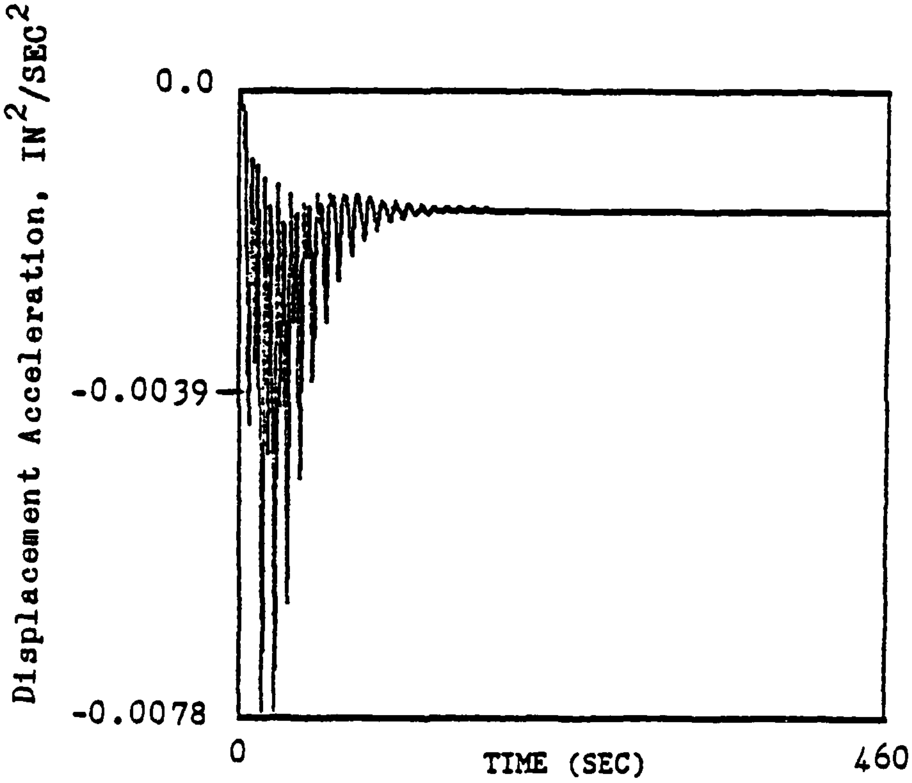

Figure 6.17 Crossmoment between linear acceleration and displacement for Example 1, $\beta = 0.1$

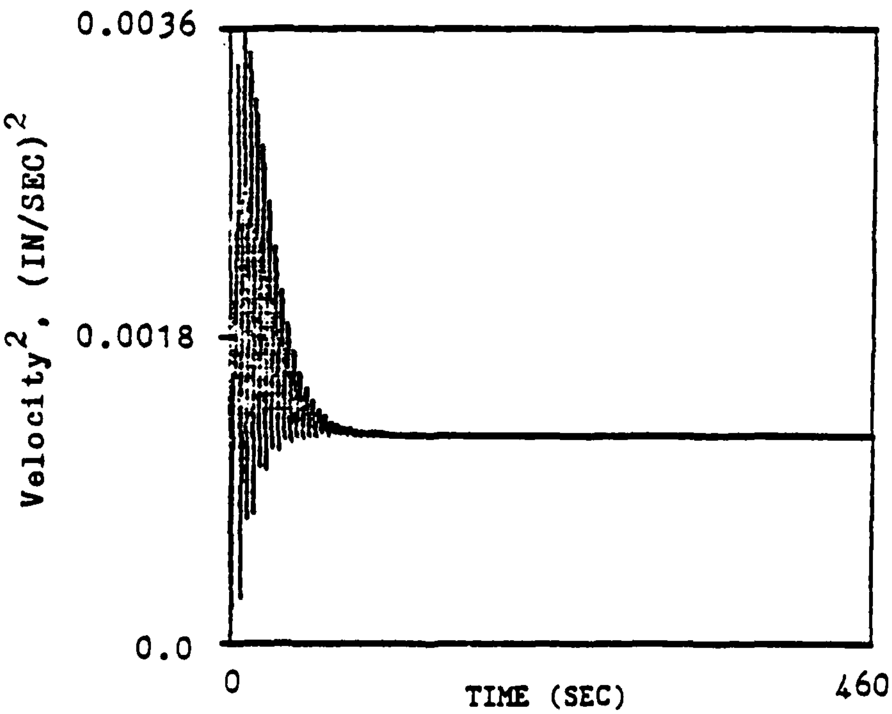

Figure 6.18 Variance of nonlinear velocity response for Example 1, $\beta = 0.1$

in Figures 6.18 through 6.22. Figures 6.18 and 6.19 show the variances of velocity and acceleration for the nonlinear system. Figures 6.20, 6.21 and 6.22 show the crossmoments between velocity and displacement, acceleration and velocity, acceleration and displacement, respectively, for the nonlinear system. The mean energy dissipation for the nonlinear system, Equation (5-5), is 0.281 (lb-in) which is small. This indicates that the yielding is not significant and the damage of the structure is small.

6.4 EXAMPLE TWO

The second example considered is a structural system with two degrees-of-freedom. The cantiliver beam, shown in Figure 6.23, with loading at the free end becomes a two-degrees-of-freedom system if the deflection and slope (lateral and rotational deformations) at the free end are independent.

The cross sectional dimensions and the material properties are given in Table 6.2.

	b_f	t_f	b_w	t_w	E	σ_y	ν	ρ	α	γ	L
unit	in	in	in	in	ksi	ksi	/	$\frac{Ksec^2}{in^4}$	/	/	in
	10.	.56	4.43	.34	29000	40	.25	.2386	.05	0	50

Table 6.2 The cross sectional dimensions and the material properties for Example 2

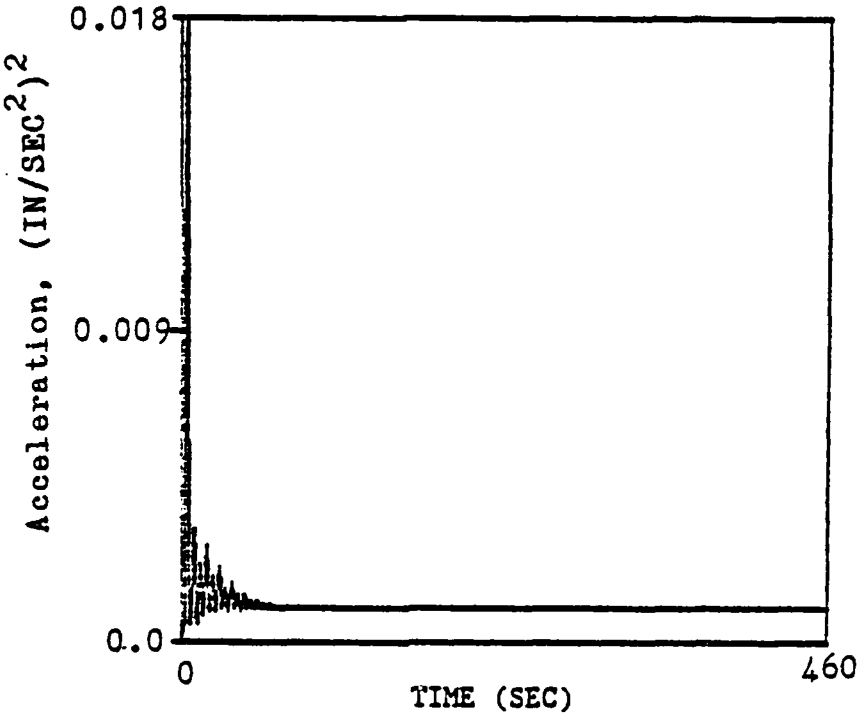

Figure 6.19 Variance of nonlinear acceleration response for Example 1, β = 0.1

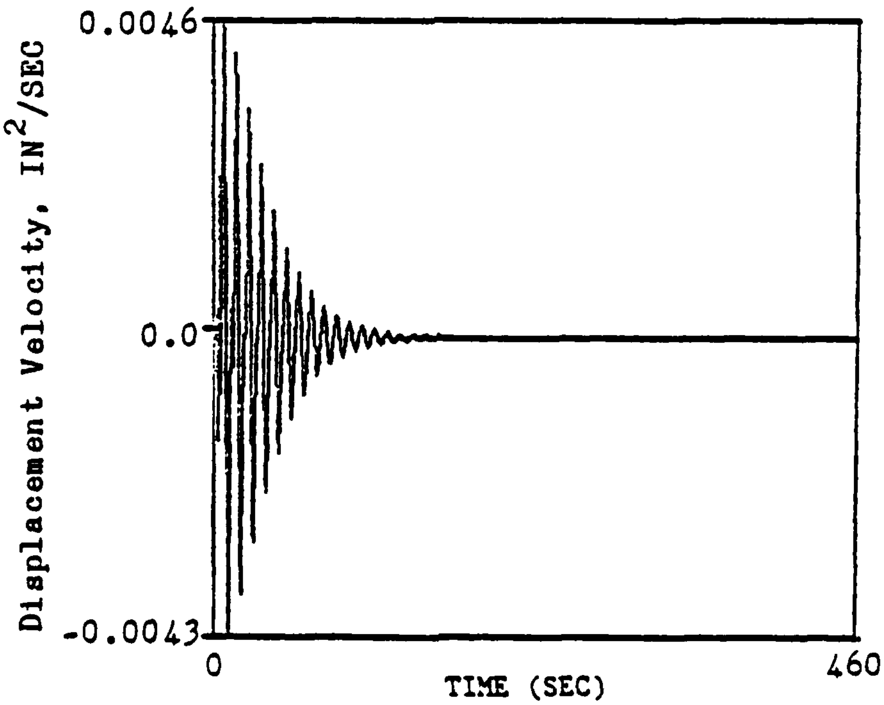

Figure 6.20 Crossmoment between nonlinear velocity and displacement for Example 1, β = 0.1

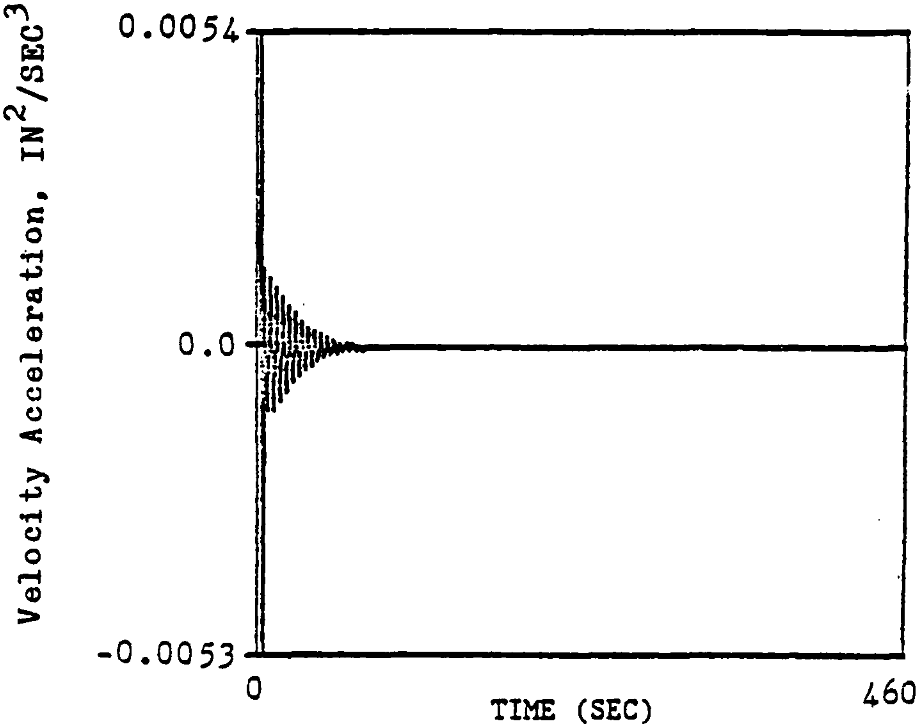

Figure 6.21 Crossmoment between nonlinear acceleration and velocity for Example 1, β = 0.1

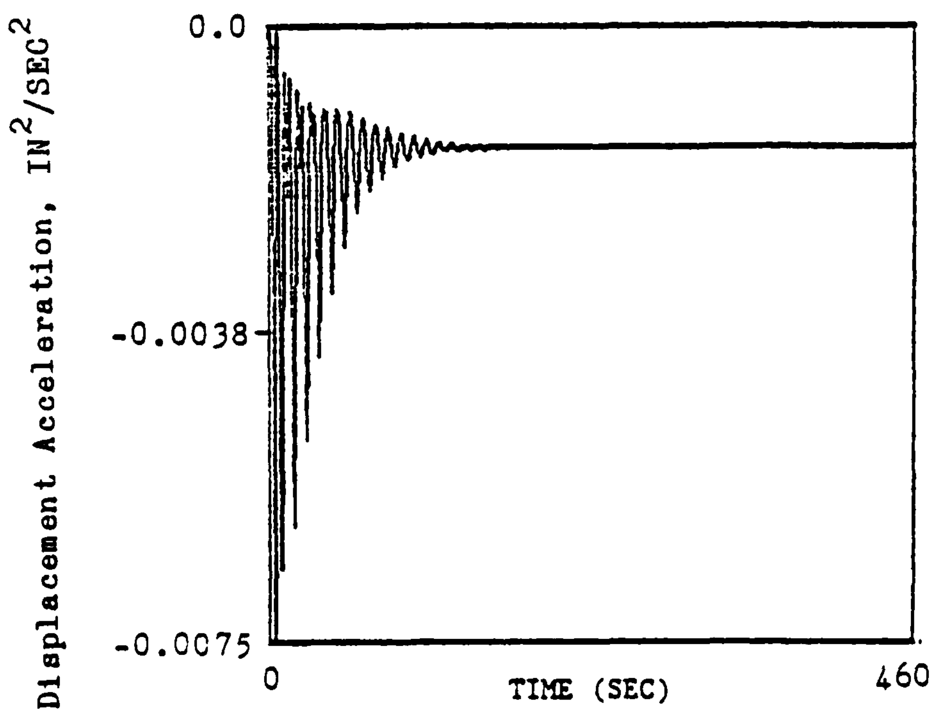

Figure 6.22 Crossmoment between nonlinear acceleration and displacement for Example 1, β = 0.1

The frequencies for this system can be computed based on the data given in Table 6.2. The results obtained from computer program FEDRANS are:

$\omega_1 = 1.244$ rad/sec,

$\omega_2 = 5.961$ rad/sec,

in which the subindices 1 and 2 represent the corresponding degree-of-freedom as shown in Figure 6.23. The time increment $\Delta\tau$ is then chosen to equal 0.2 sec.

The mean and variance of the input are graphically shown in Figure 6.24.

The mean response for the linear system was computed and the result is graphically shown in Figure 6.25. The mean value of the response of the nonlinear system was also computed and the result is plotted in Figure 6.26.

The variances of the responses were also computed for the cases of $\beta = 0$ and 0.1. The system with $\beta = 0$ is considered first. The variance of the displacement response of the linear system was computed. The result is plotted in Figure 6.27. The variance of the displacement response of the nonlinear system was also computed. The result is plotted in Figure 6.28.

The system with $\beta = 0.1$ is considered next. The variances of the responses of the linear and nonlinear systems were computed, and the results are plotted in Figures 6.29 and 6.30, respectively. Comparison of Figure 6.27 and 6.28 shows that both the linear and nonlinear systems approach a state of stationary response with the same amplitude, and at the same time. When $\beta = 0.1$, the variances of both the linear and nonlinear responses

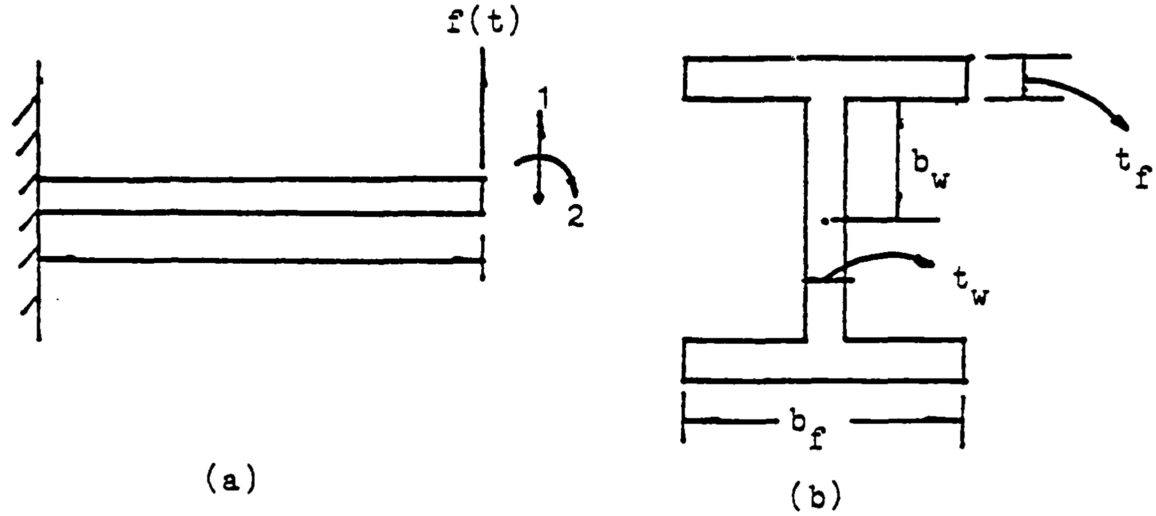

Figure 6.23 Cantilever beam for Example 2, two-degrees-of-freedom system

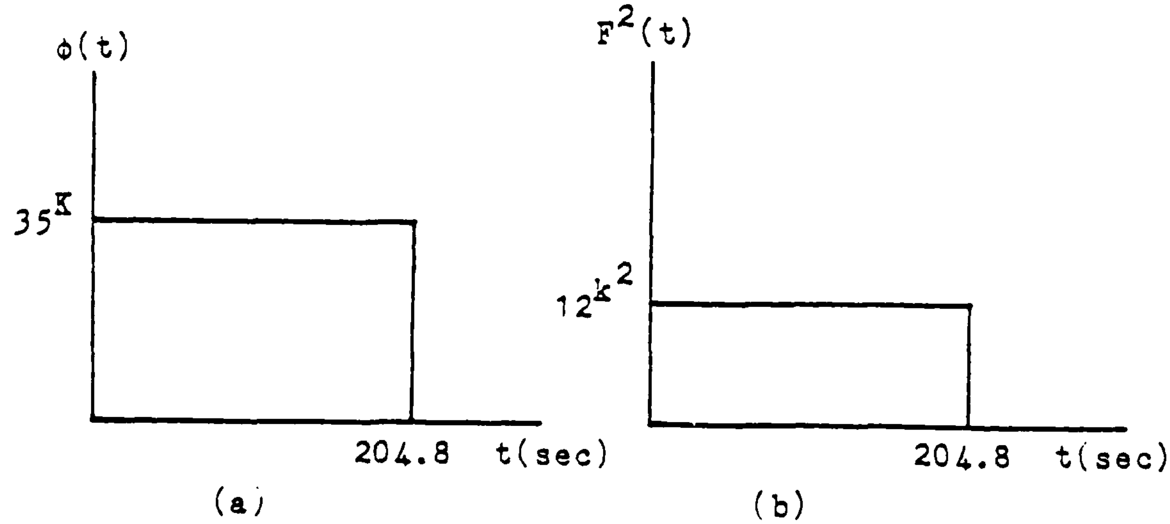

Figure 6.24 Mean and autocovariance of the input forcing function for Example 2

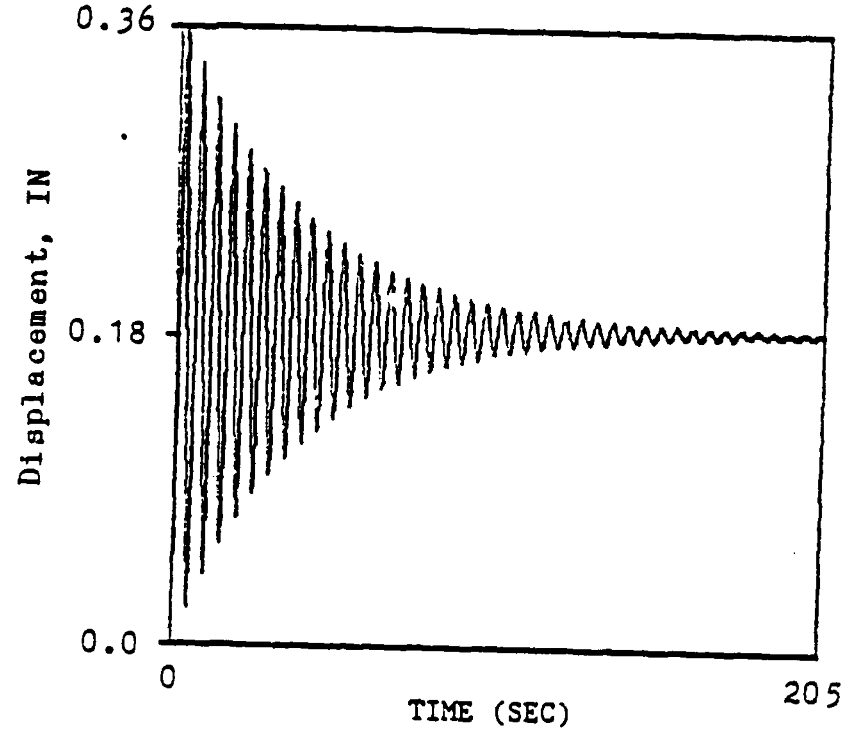

Figure 6.25(a) Mean displacement response for linear system, at DOF = 1

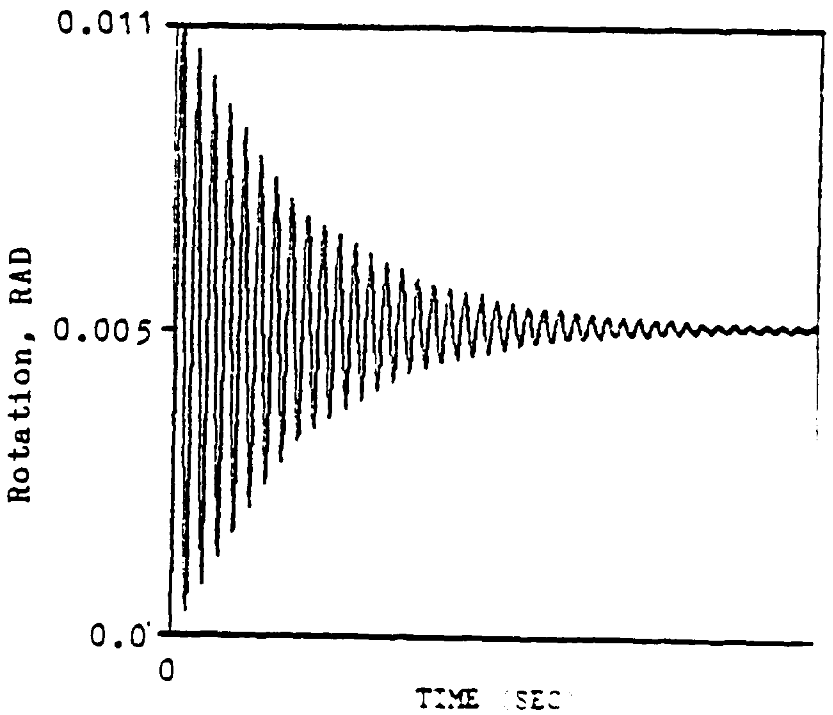

Figure 6.25(b) Mean [illegible]
linear [illegible]

AD-A179 772 DAMAGE DIAGNOSIS FOR ELASTO-PLASTIC STRUCTURES(U) NEW MEXICO UNIV ALBUQUERQUE DEPT OF MECHANICAL ENGINEERING F D JU ET AL. DEC 86 ME-138(86)AFOSR-993-2 2/3

UNCLASSIFIED AFOSR-TR-87-0413 AFOSR-85-0085 F/G 20/11 NL


1.0
2.8
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

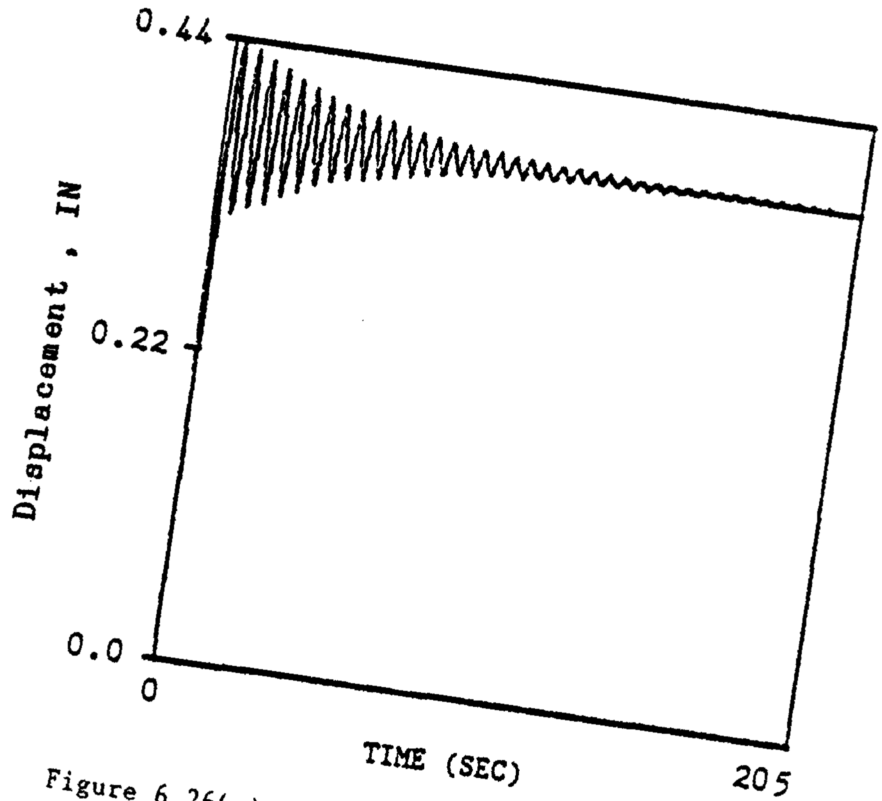

Figure 6.26(a) Mean displacement response for nonlinear system, at DOF = 1

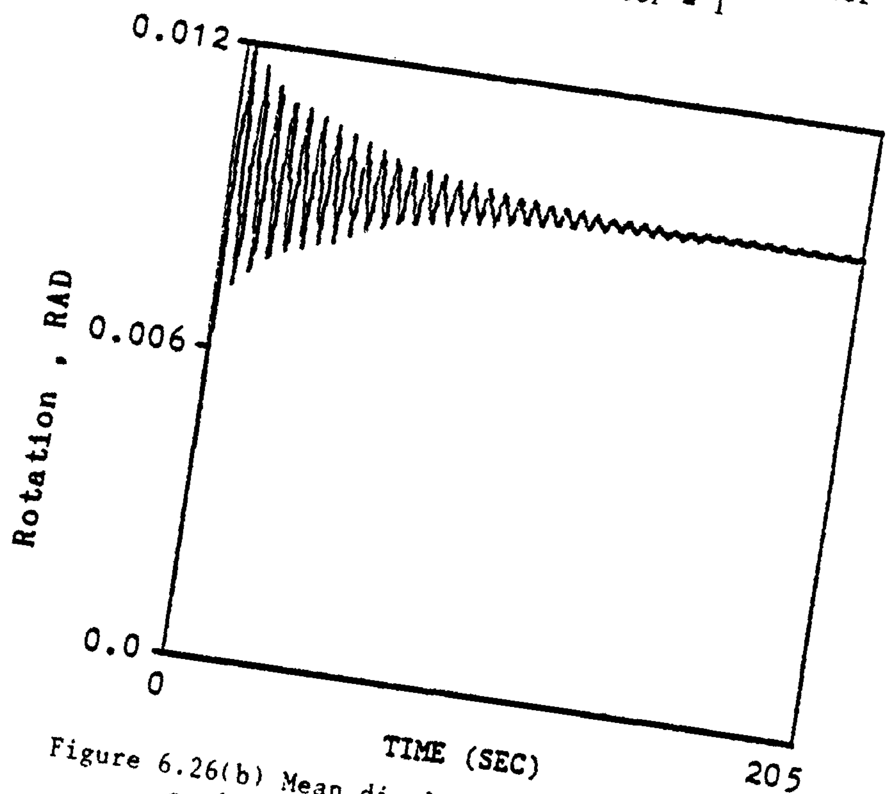

Figure 6.26(b) Mean displacement response for nonlinear system, at DOF = 2

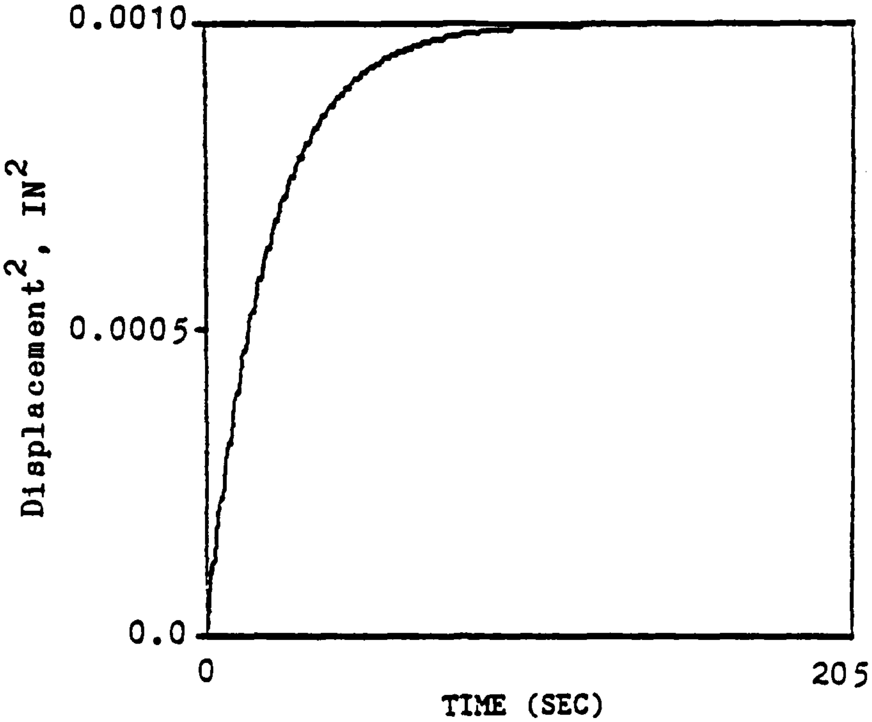

Figure 6.27(a) Variance of displacement response, linear system, β = 0, at DOF = 1

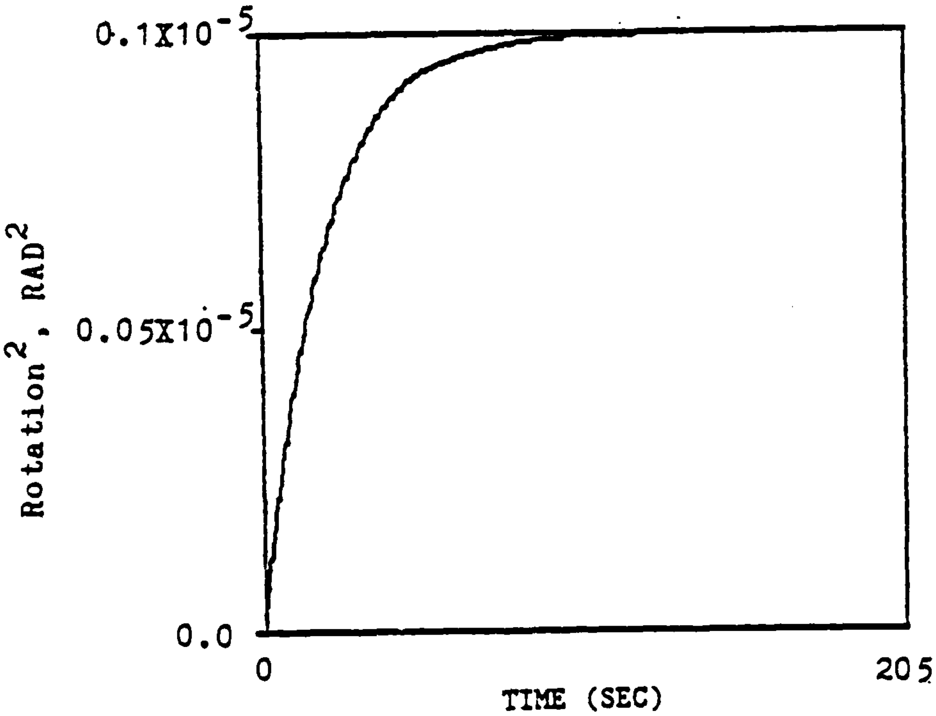

Figure 6.27(b) Variance of displacement response, linear system, β = 0, at DOF = 2

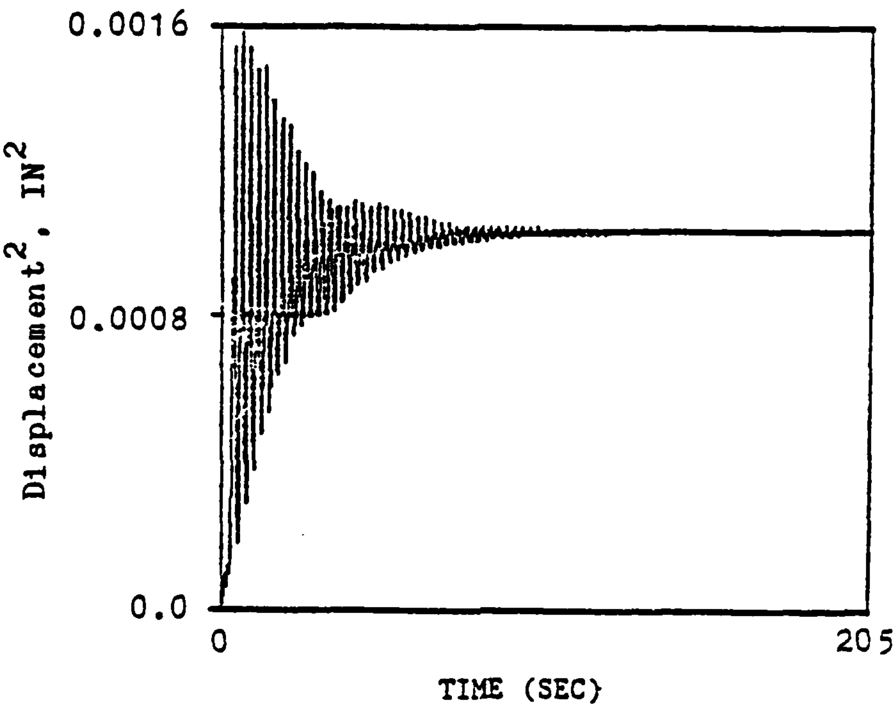

Figure 6.28(a) Variance of displacement response, nonlinear system, $\beta = 0$, at DOF = 1

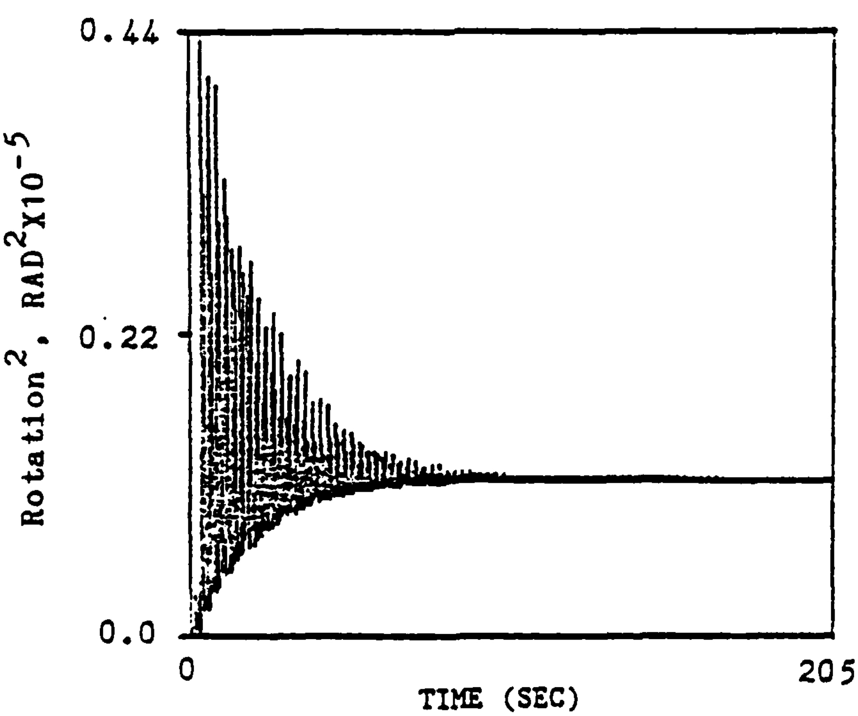

Figure 6.28(b) Variance of displacement response, nonlinear system, $\beta = 0$, at DOF = 2

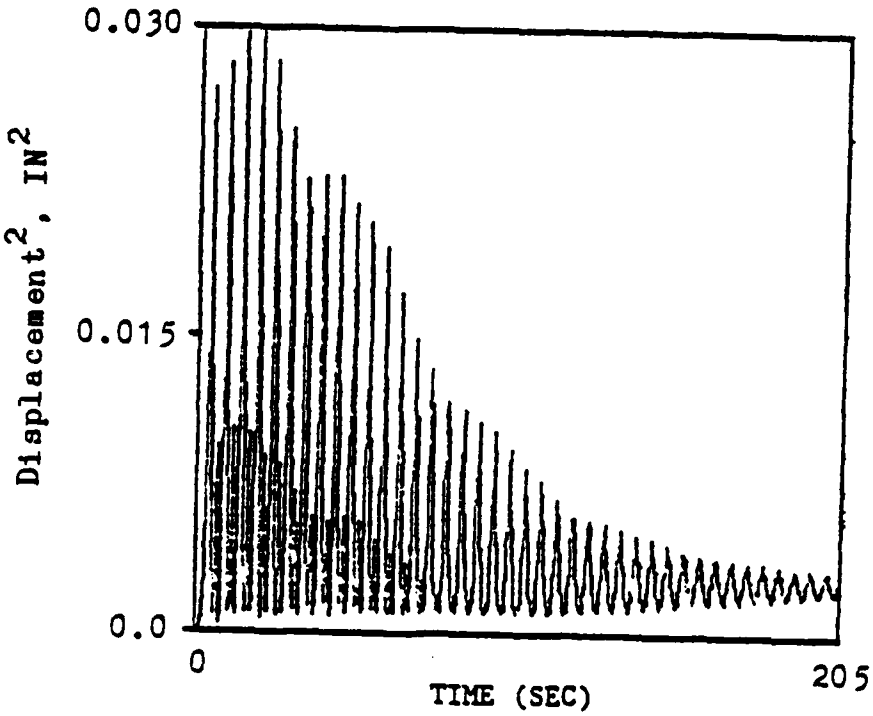

Figure 6.29(a) Variance of displacement response, linear system, β = 0.1, at DOF = 1

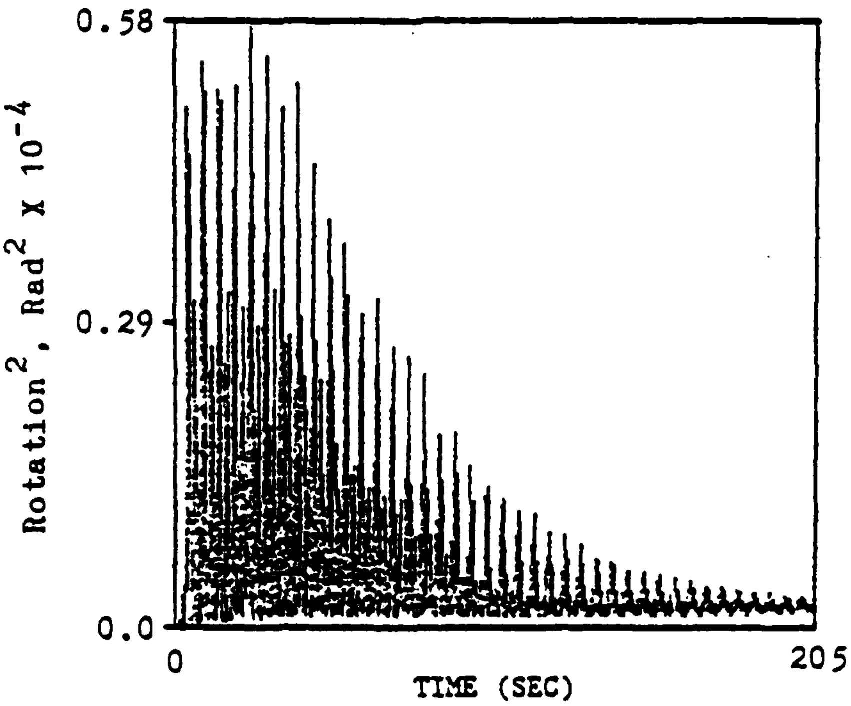

Figure 6.29(b) Variance of displacement response, linear system, β = 0.1, at DOF = 2

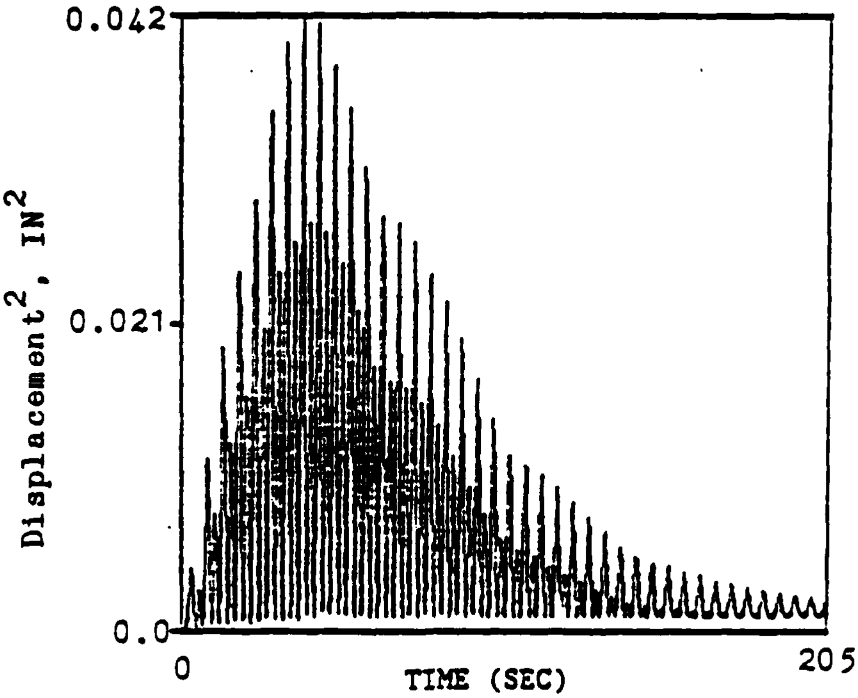

Figure 6.30(a) Variance of displacement response, nonlinear system, β = 0.1, at DOF = 1

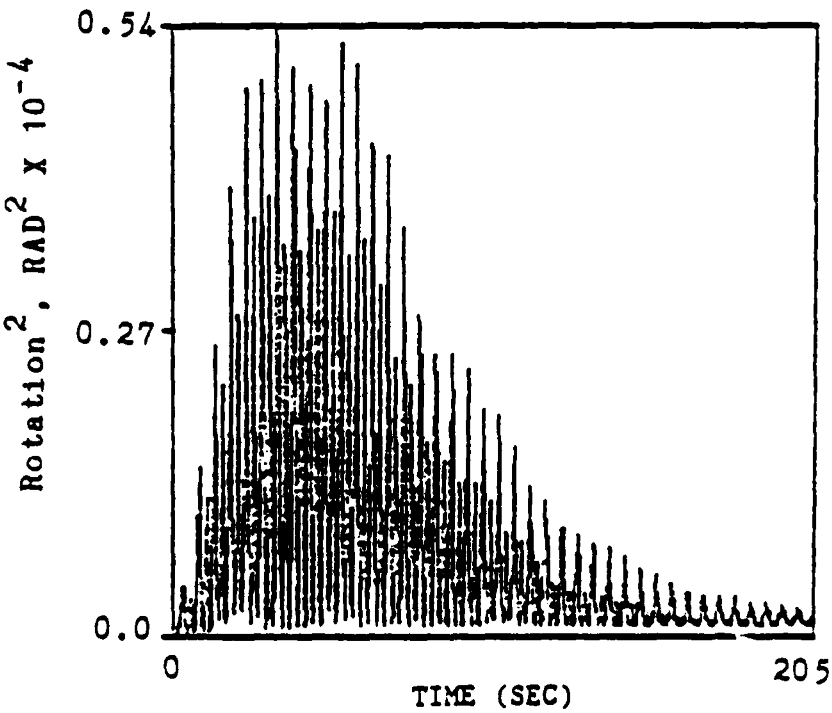

Figure 6.30(b) Variance of displacement response, nonlinear system, β = 0.1, at DOF = 2

approach a stationary state much more slowly than the case where $\beta = 0$. The energy dissipation for the nonlinear system, Equationn (5-5), is 0.89E-02 (lb-in) which still is small.

6.5 EXAMPLE THREE

The third example considered is a one-story building as shown in Figure 6.31. This structural system has six degrees-of-freedom. The load is assumed to act in the horizontal direction as shown.

The cross sectional dimensions and member properties for members number 1 and number 2 are given in Table 6.3.

	b_f	t_f	b_w	t_w	E	σ_y	ν	ρ	α	γ	L
unit	in	in	in	in	ksi	ksi		$k\frac{sec^2}{in^4}$			in
1	5.03	.42	6.54	.25	29000	40	.25	3.35×10^{-7}	15.	.0005	240.
2	6.75	.46	6.54	.34	"	"	"	"			"

Table 6.3 The cross sectional dimensions and the material properties for Example 3

The lowest and highest frequencies for this system were computed by using the computer program FEDRANS, and the results are:

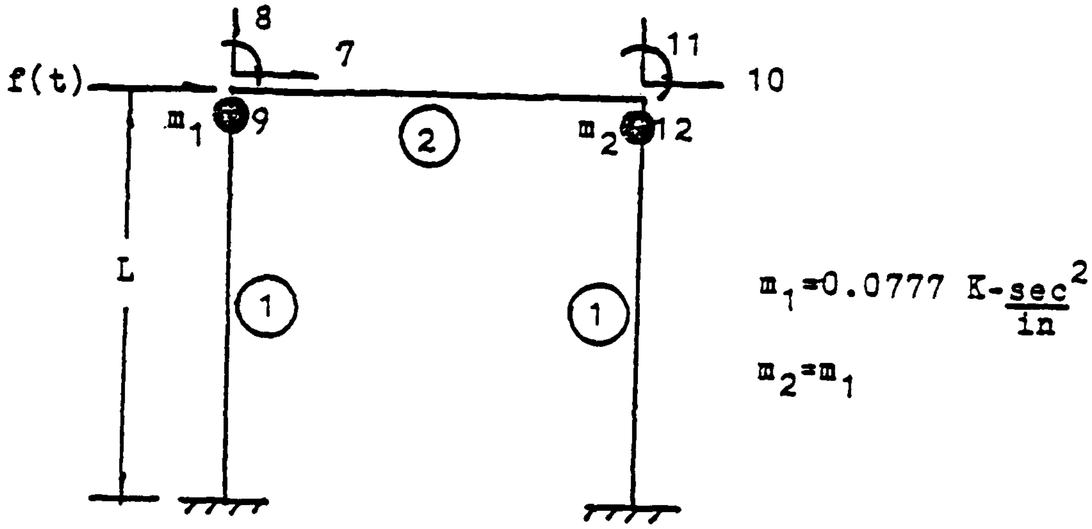

Figure 6.31 One story building frame structure for Example 3

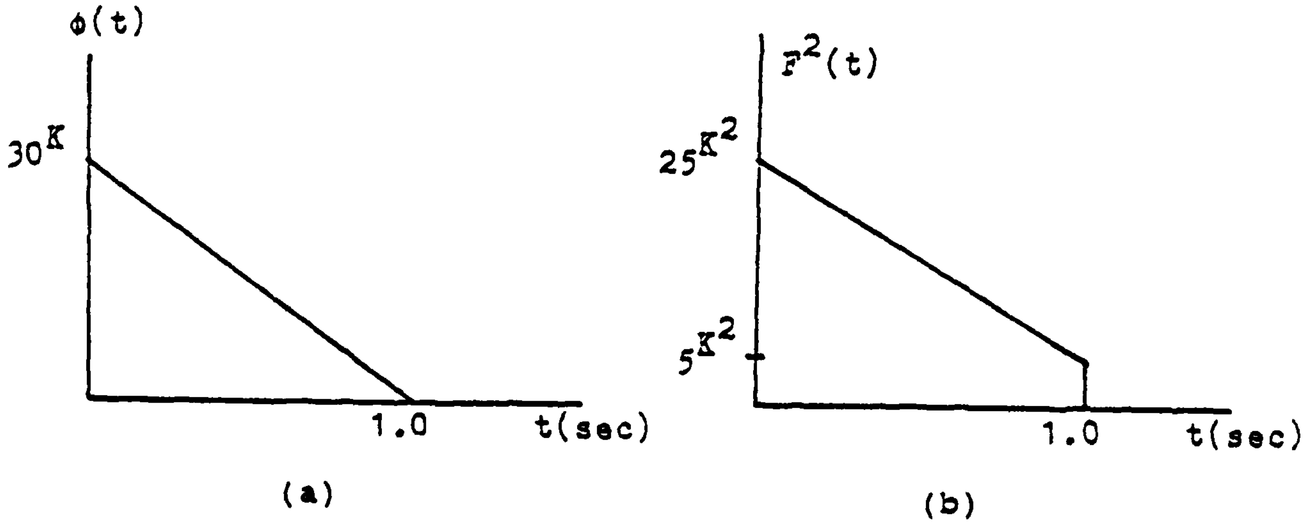

Figure 6.32 Mean and autocovariance of the input forcing function for Example 3

$$\omega_{min} = 76.07 \text{ (rad/sec)},$$

$$\omega_{max} = 1291.47 \text{ (rad/sec)}.$$

The time increment Δt is chosen to be 1 ms. The mean and mean square inputs are graphically given in Figure 6.32.

The mean displacement responses for the linear and nonlinear systems were computed using the computer program FEDRANS, and the results are plotted in Figures 6.33 and 6.34. Figure 6.33(a) shows the mean value of the response at degree-of-freedom 7. Figure 6.33(b) shows the mean value of the response at degree-of-freedom 8. Figure 6.33(c) shows the mean value of the response at degree-of-freedom 9. Figures 6.34(a), (b) and (c) present similar results for the nonlinear system.

The mean square displacement responses were computed for $\beta = 0$ and 0.01. For the case of $\beta = 0$, the variances of displacement responses for the degrees-of-freedom from 7 through 9 were computed. Figure 6.35 shows the results for the linear system, and Figure 6.36 shows the results for the nonlinear system. Figures 6.35(a), (b) and (c) show the variance of displacement (or rotation) response of the linear system at degrees-of-freedom 7, 8 and 9, respectively; Figures 6.36(a), (b) and (c) show the variance of displacement (or rotation) response of the nonlinear system at degrees-of-freedom 7, 8 and 9, respectively. For the case of $\beta = 0.01$, the variances of displacement responses are computed for the same degrees-of-freedom. Figures 6.37(a), (b) and (c) summarize results for the linear system, and Figures 6.41(a), (b) and (c) summarize results for the nonlinear system.

From Figures 6.35 through 6.38 it can be concluded that the envelopes of the variances of displacement responses at all the degrees-of-freedom possess approximately the same shape for the linear and nonlinear system.

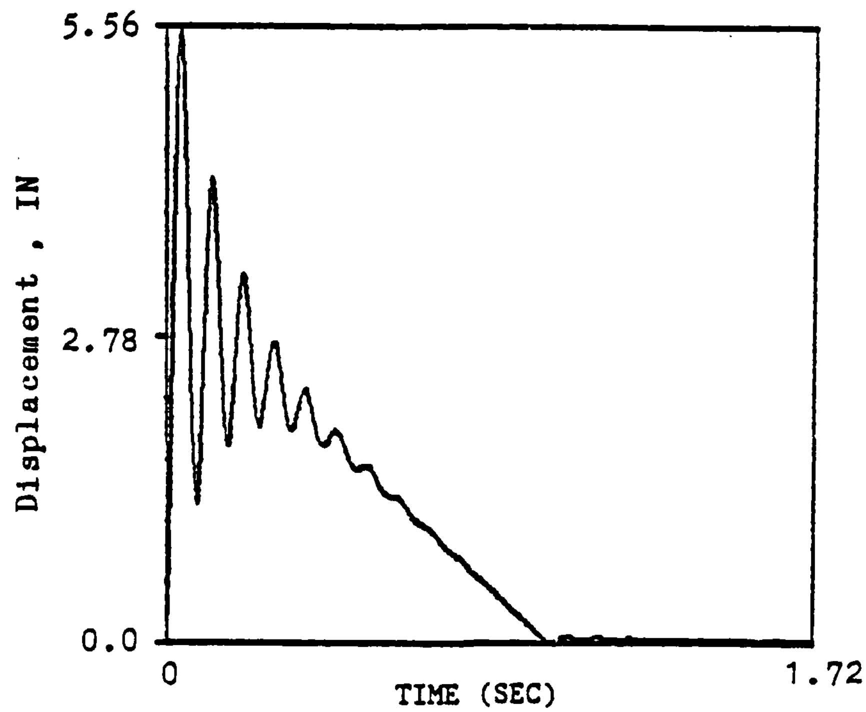

Figure 6.33(a) Mean displacement response for linear system at DOF = 7

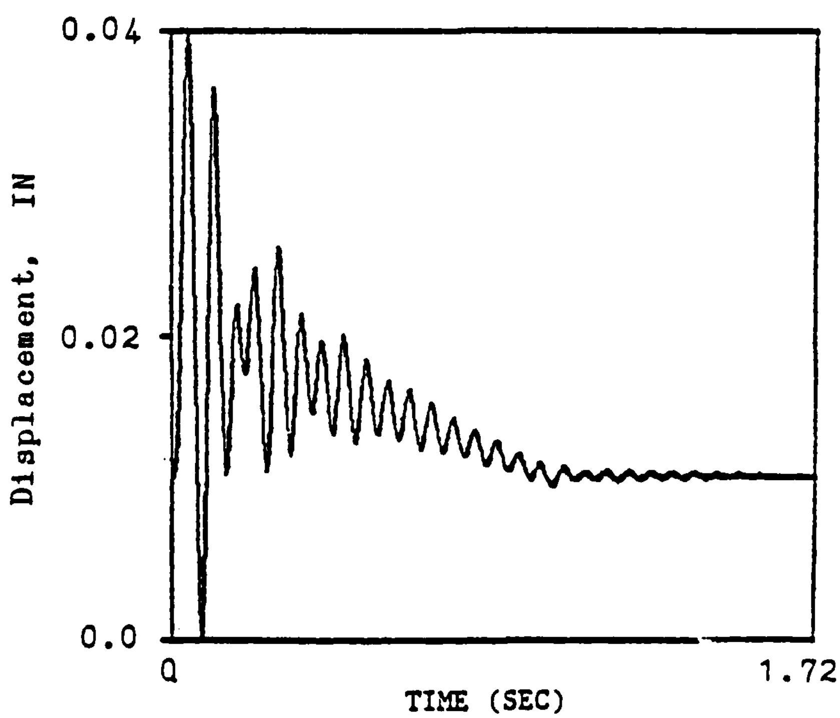

Figure 6.33(b) Mean displacement response for linear system at DOF = 8

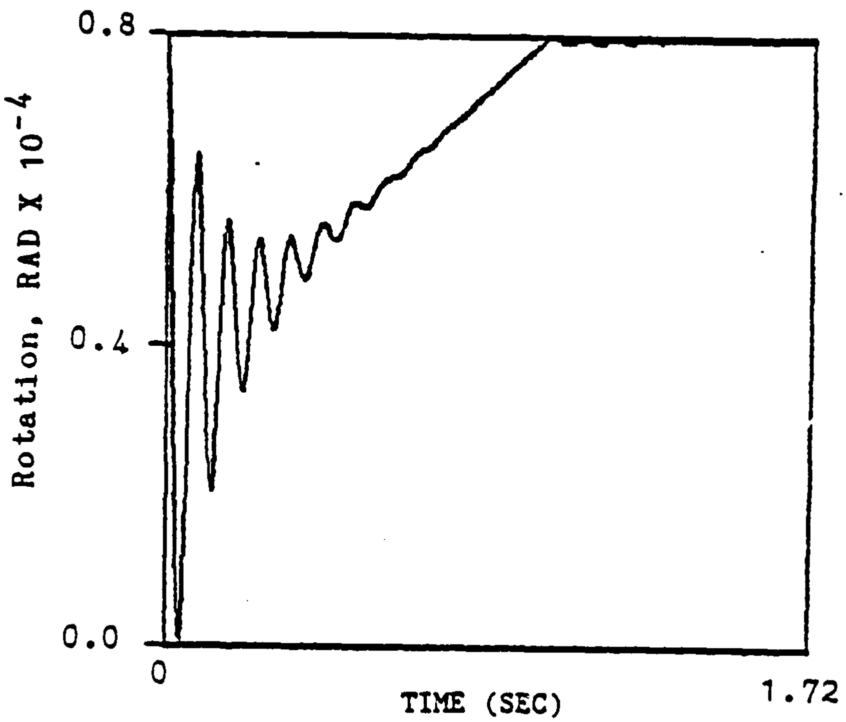

Figure 6.33(c) Mean displacement response for linear system at DOF = 9

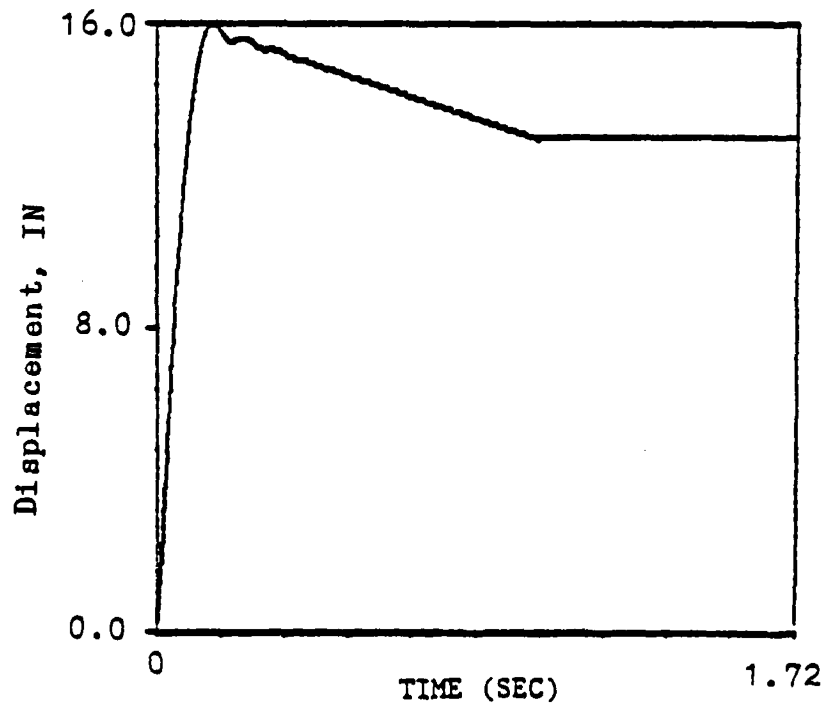

Figure 6.34(a) Mean displacement response for nonlinear system at DOF = 7

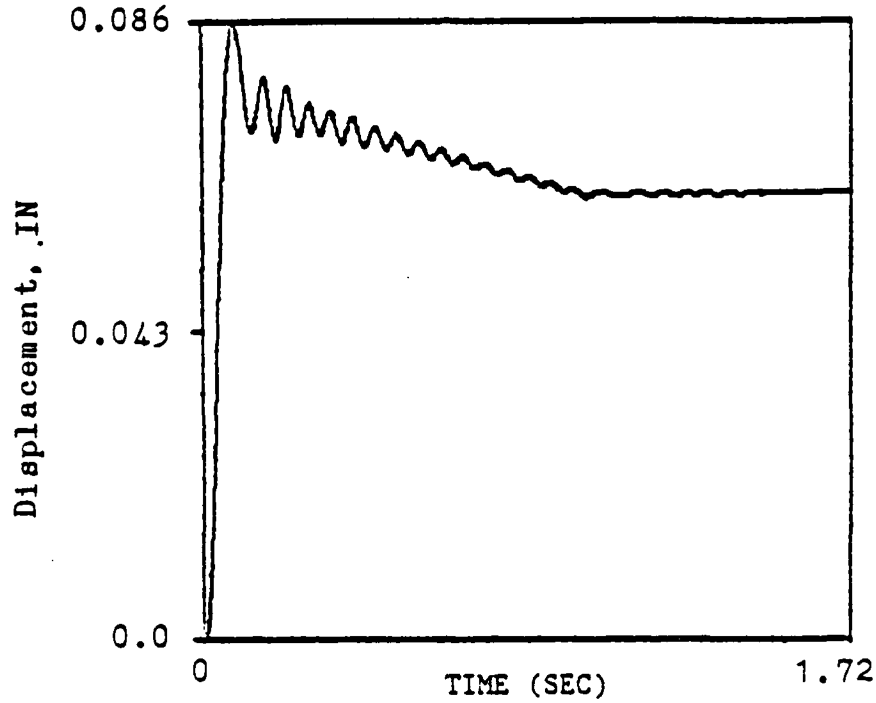

Figure 6.34(b) Mean displacement response for nonlinear system at DOF = 8

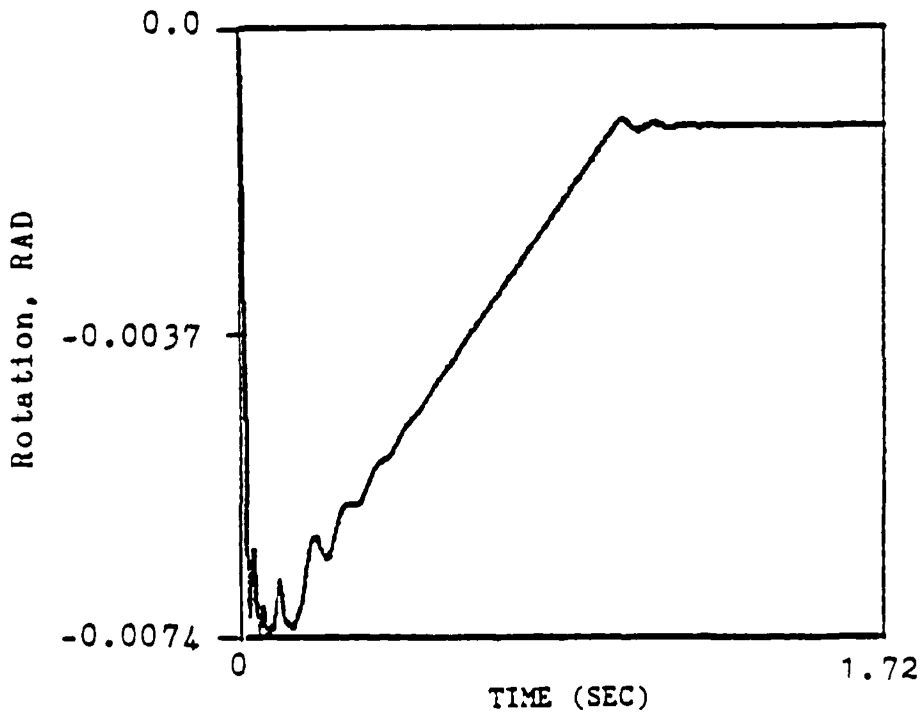

Figure 6.34(c) Mean displacement response for nonlinear system at DOF = 9

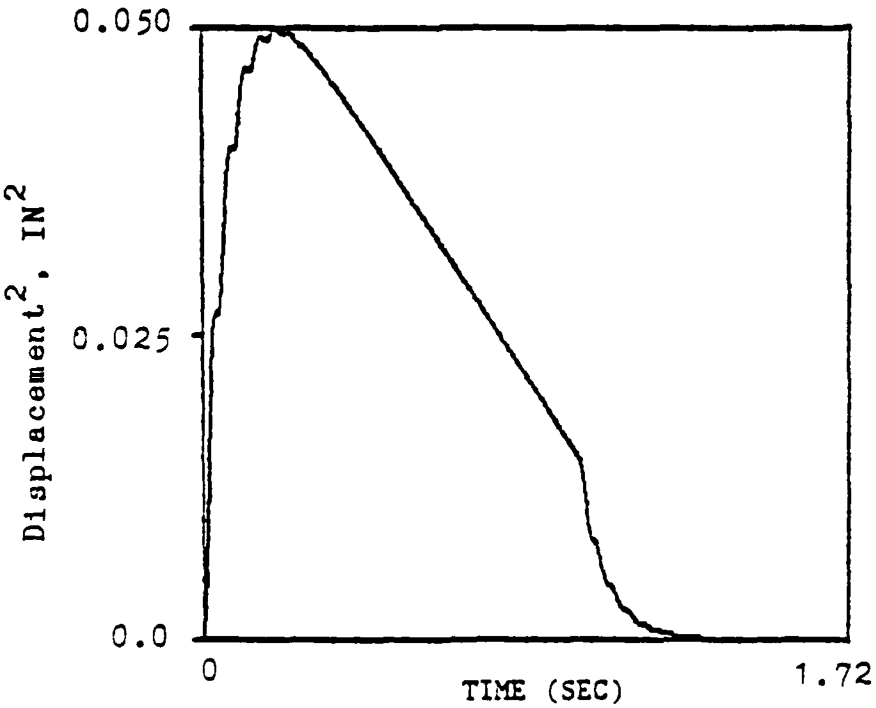

Figure 6.35(a) Variance of displacement response, linear system, $\beta = 0$, at DOF = 7

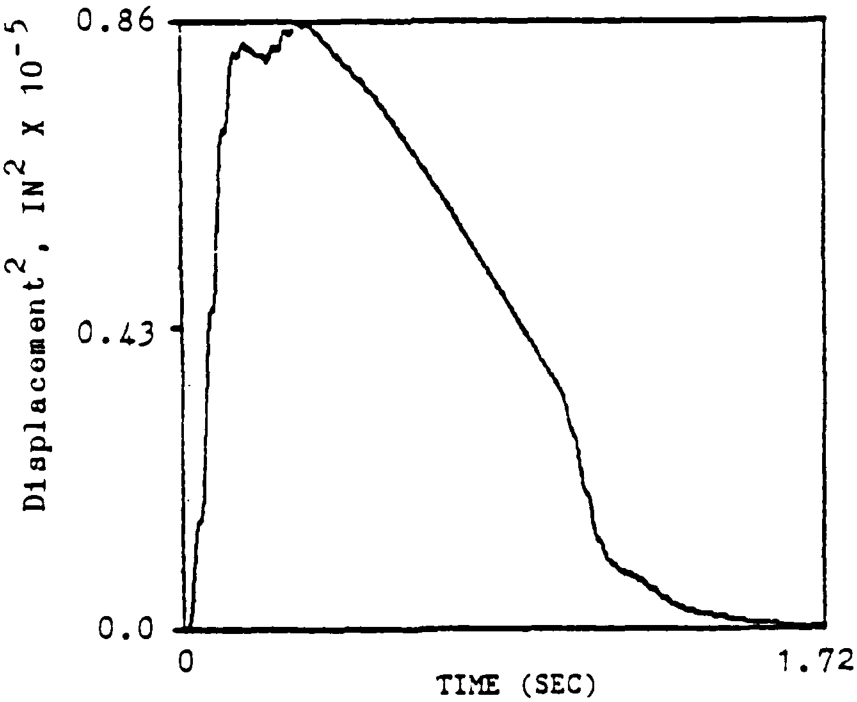

Figure 6.35(b) Variance of displacement response, linear system, $\beta = 0$, at DOF = 8

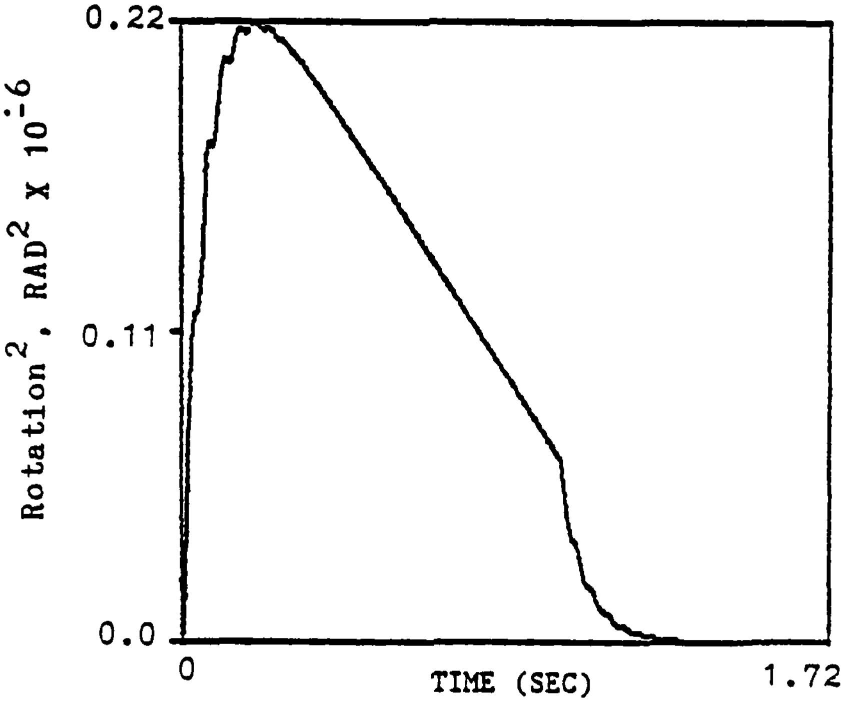

Figure 6.35(c) Variance of displacement response, linear system, β = 0, at DOF = 9

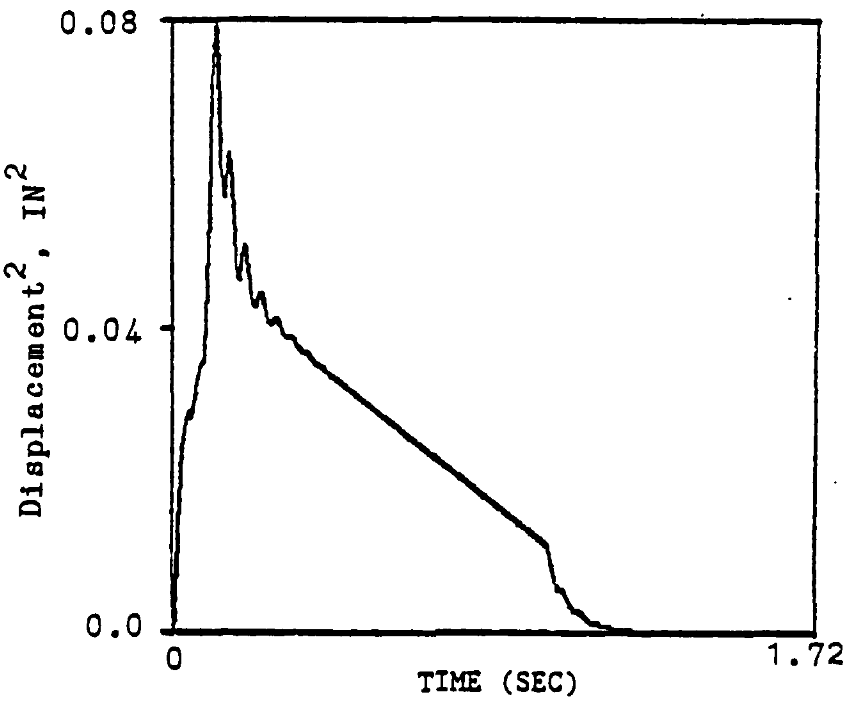

Figure 6.36(a) Variance of displacement response, nonlinear system, β = 0, at DOF = 7

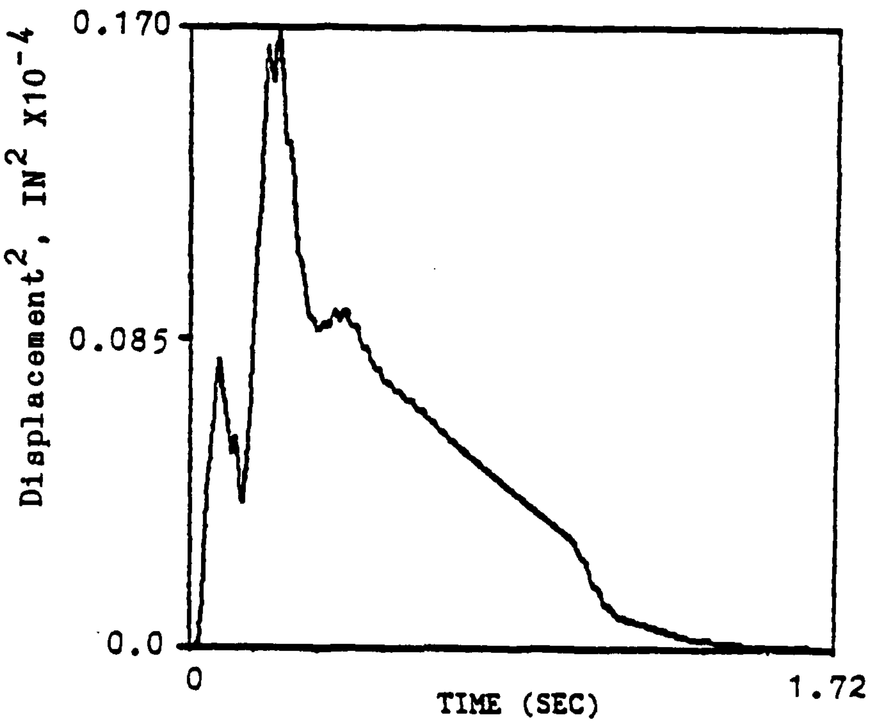

Figure 6.36(b) Variance of displacement response, nonlinear system, β = 0, at DOF = 8

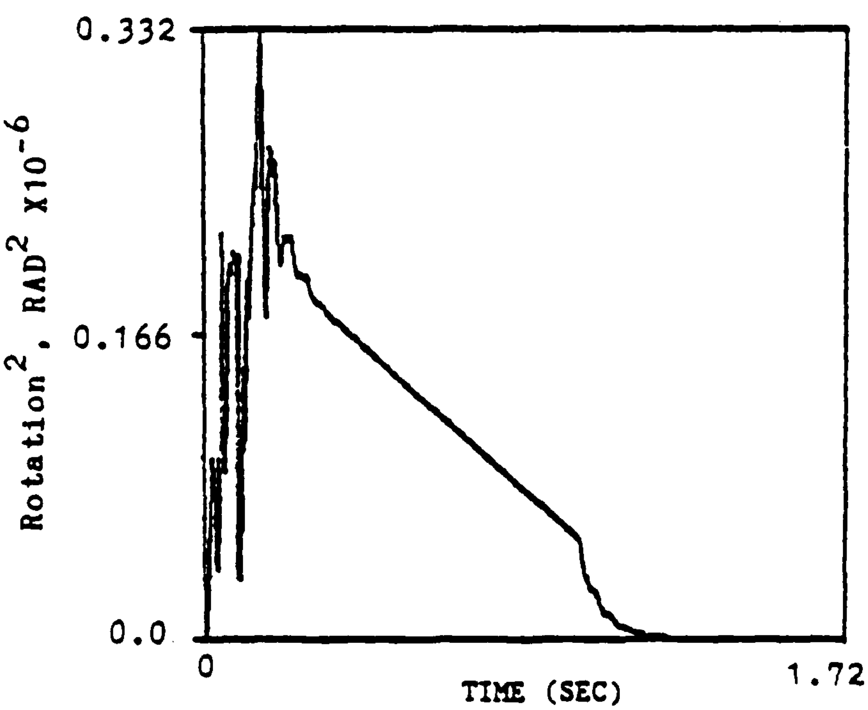

Figure 6.36(c) Variance of displacement response, nonlinear system, β = 0, at DOF = 9

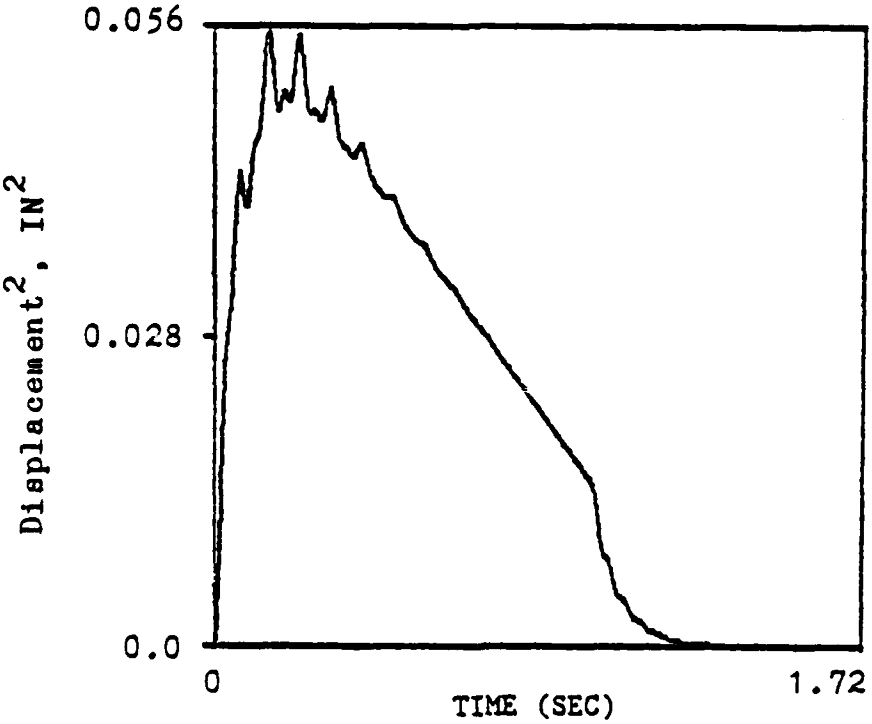

Figure 6.37(a) Variance of displacement response, linear system, $\beta = 0.01$, at DOF = 7

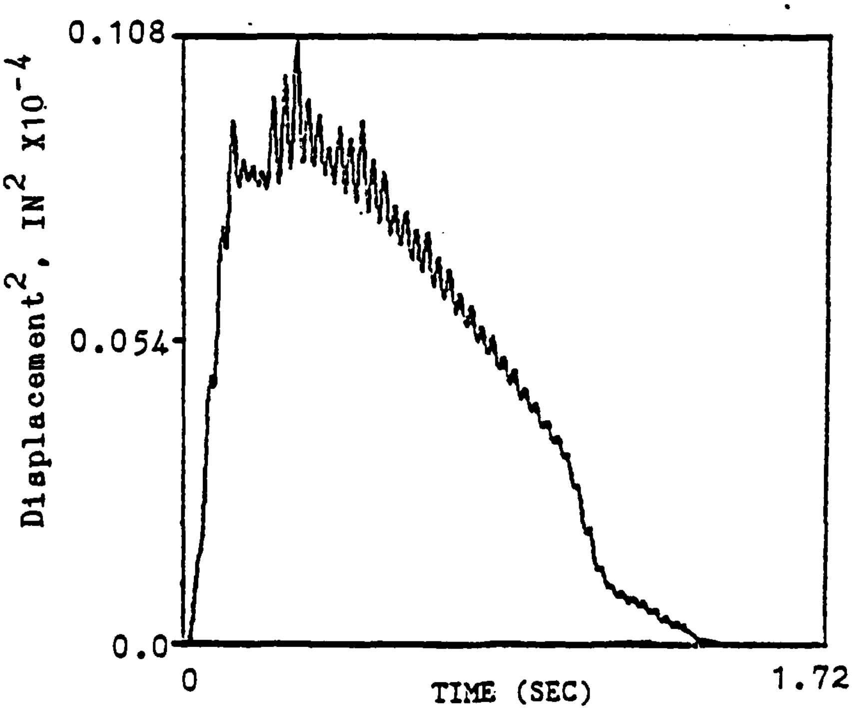

Figure 6.37(b) Variance of displacement response, linear system, $\beta = 0.01$, at DOF = 8

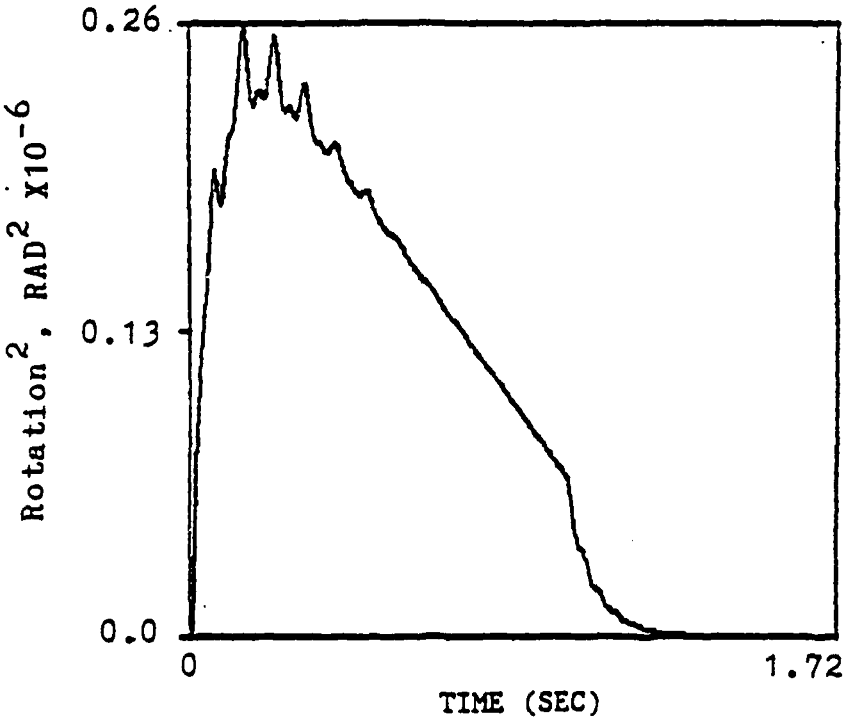

Figure 6.37(c) Variance of displacement response, linear system, β = 0.01, at DOF = 9

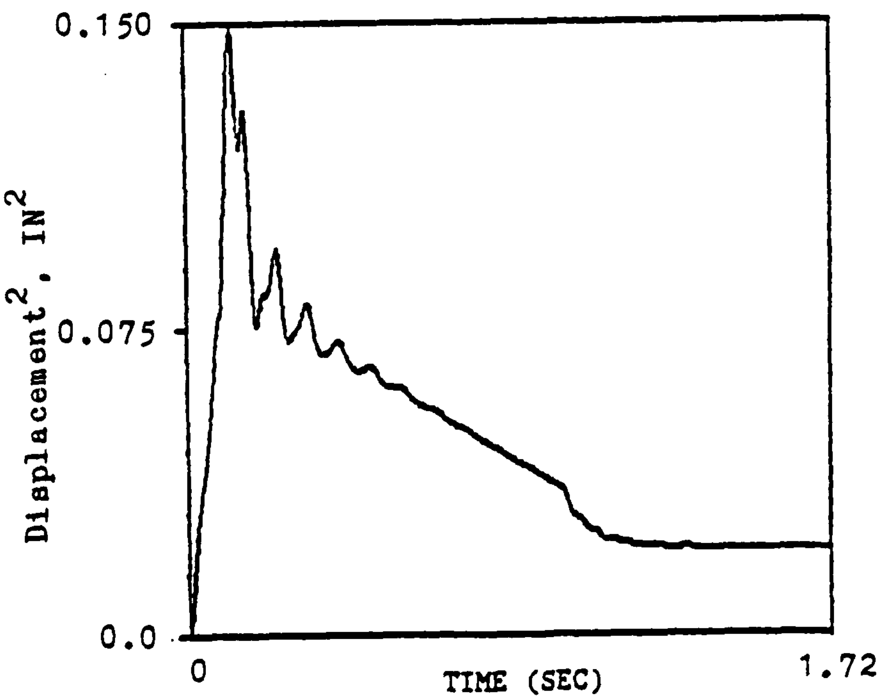

Figure 6.38(a) Variance of displacement response, nonlinear system, β = 0.01, at DOF = 7

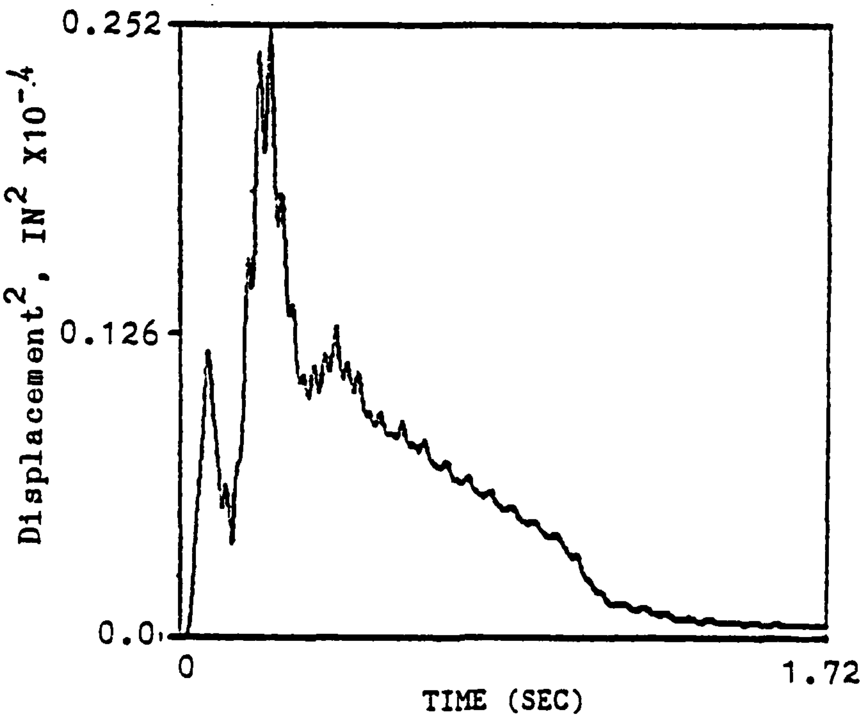

Figure 6.38(b) Variance of displacement response, nonlinear system, β = 0.01, at DOF = 8

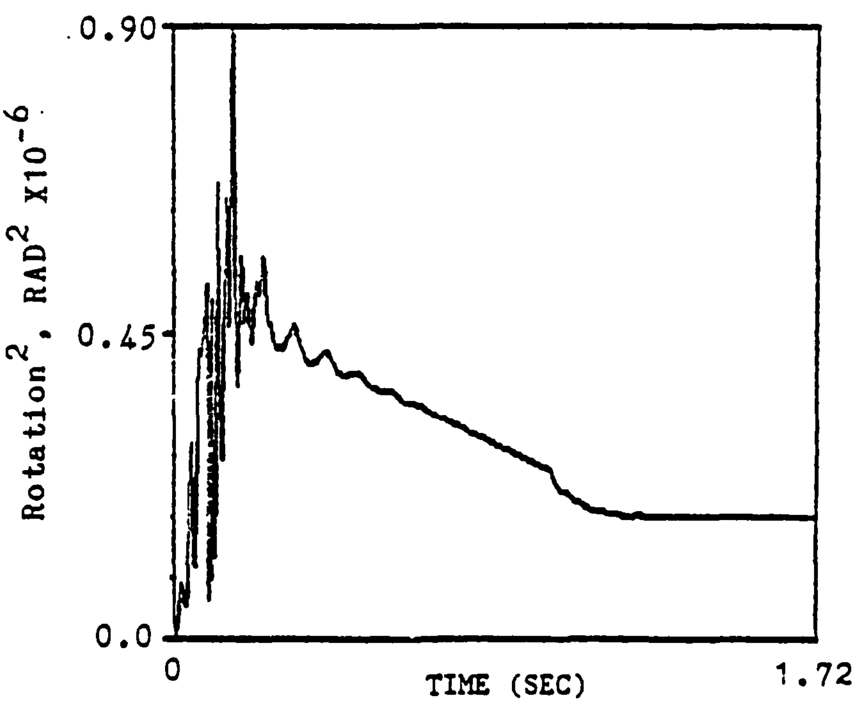

Figure 6.38(c) Variance of displacement response, nonlinear system, β = 0.01, at DOF = 9

When the stiffness behaves randomly, the envelopes of the variances of displacement responses at the various degrees-of-freedom still have the same general shapes, however, some early-time rapid variation is present in the response variance signals, especially for the nonlinear system.

Chapter 7

DISCUSSION AND CONCLUSION

7.1 DISCUSSION

It was mentioned in Chapter 2 that the higher order terms which were neglected in the present analysis can be included for SDF system when both the random stiffness K and the displacement response random process Z(t) are Gaussian. Consider for a linear, SDF system, the governing equation can be represented by Equation (2-4),

$$m(\ddot{\mu}+\ddot{Z}) + c(\dot{\mu}+\dot{Z}) + (\lambda+K)(\mu+Z) = \phi + F. \tag{2-4}$$

The mean response can be obtained by taking the expectation of the above equation,

$$m\ddot{\mu} + c\dot{\mu} + \lambda\mu + E[KZ] = \phi, \tag{7-1}$$

where use has been made the fact that Z and K are both zero mean. The remaining part which characterizes the random component of the response can be obtained by subtracting Equation (7-1) from Equation (2-4),

$$m\ddot{Z} + c\dot{Z} + \lambda Z = F - K\mu + E[KZ] - KZ. \tag{7-2}$$

The mean and random component of the displacement response at time step j+1 can be evaluated similarly to Equations (2-13) and (2-19),

$$\mu_{j+1} = (A_2\mu_j + A_3\mu_{j-1} + \Delta t^2\phi_j - \Delta t^2 E[KZ_j])/A_1, \tag{7-3}$$

$$Z_{j+1} = (A_2 Z_j + A_3 Z_{j-1} + \Delta t^2 F_j - \Delta t^2 K\mu_j + \Delta t^2 E[KZ_j] - \Delta t^2 KZ_j)/A_1, \tag{7-4}$$

where A_1, A_2, A_3 are the same as given by Equation (2-15). The term $E[KZ_j]$ can be evaluated by postmultiplying Equation (7-4) by K and then taking expectation,

$$E[KZ_j] = (A_2 E[KZ_{j-1}] + A_3 E[KZ_{j-2}] - \Delta t^2 E[K^2]\mu_j - \Delta t^2 E[K^2 Z_j])/A_1, \tag{7-5a}$$

where use has been made the fact that K is independent of F(t) and both are zero mean. It is noted that the last term in Equation (7-5a), $E[K^2 Z_j]$, is the higher order crossmoment between K and Z_j. However, when both K and Z(t) are Gaussian distribution, this term is zero [33], i.e.,

$$E[K^2 Z_j] = 0.$$

Consequently, Equation (7-5a) can be reduced to

$$E[KZ_j] = (A_2 E[KZ_{j-1}] + A_3 E[KZ_{j-2}] - \Delta t^2 E[K^2]\mu_j)/A_1. \tag{7-5b}$$

It is noted that the first and second term in Equation (7-5b) possess the recurrent form of $E[KZ_j]$ and the last term is assumed known. Therefore, Equation (7-5b) can be evaluated under the assumption that K, Z(t) are Gaussian distribution. Consequently, Equation (7-3) provides the exact solution of the mean displacement response.

The variance of the displacement response can be computed similarly to Equation (2-22),

$$
\begin{aligned}
E[Z_{j+1}^2] = {} & A_2^2E[Z_j^2] + A_3^2E[Z_{j-1}^2] + \Delta t^4E[K^2]\mu_j^2 + \Delta t^4E[F_j^2] - \\
& - \Delta t^4E[KZ_j]^2 + \Delta t^4E[K^2Z_j^2] + 2A_2A_3E[Z_jZ_{j-1}] - \\
& - 2\Delta t^2A_2E[KZ_j]\mu_j - 2\Delta t^2A_2E[KZ_j^2] - 2\Delta t^2A_3E[KZ_{j-1}]\mu_j - \\
& - 2\Delta t^2E[KZ_jZ_{j-1}]A_3 + 2\Delta t^4E[K^2Z_j]\mu_j)A_1^{-2}.
\end{aligned}
\qquad (7\text{-}6)
$$

The new terms introduced in Equation (7-6) are $E[KZ_j]$, $E[KZ_j^2]$, $E[K^2Z_j]$, $E[KZ_jZ_{j-1}]$ and $E[K^2Z_j^2]$. However, under the assumption that K and Z(t) are Gaussian, the following terms are zero;

$$E[K^2Z_j] = E[KZ_j^2] = E[KZ_jZ_{j-1}] = 0. \qquad (7\text{-}7)$$

The term $E[K^2Z_j^2]$ can be reduced to the following lower moments under Gaussian assumption,

$$E[K^2Z_j^2] = 2E[KZ_j]^2 + E[K^2]\ E[Z_j^2]. \qquad (7\text{-}8)$$

By using Equations (7-7) and (7-8), the variance of the displacement response which is given by Equation (7-6) can be solved. The comparison of the mean response between the one from equation (7-3) and the one from Equation (2-14) is made by using the same system and excitation of Example 1 and the result is plotted in Figure 7.1. The error is less than 0.5

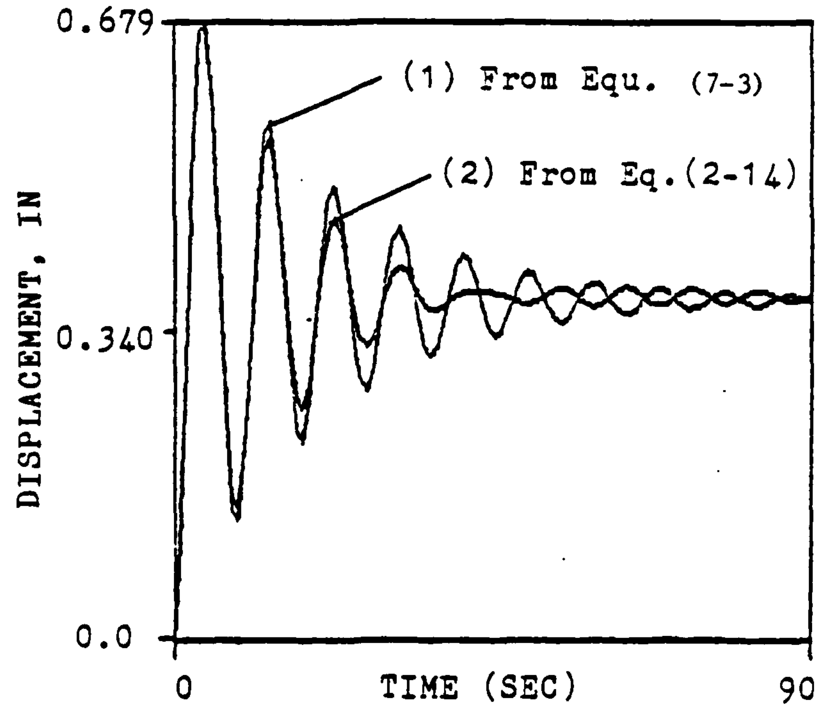

Figure 7.1 The comparison of mean response between (1) including the higher order terms and (2) omission of the higher order terms

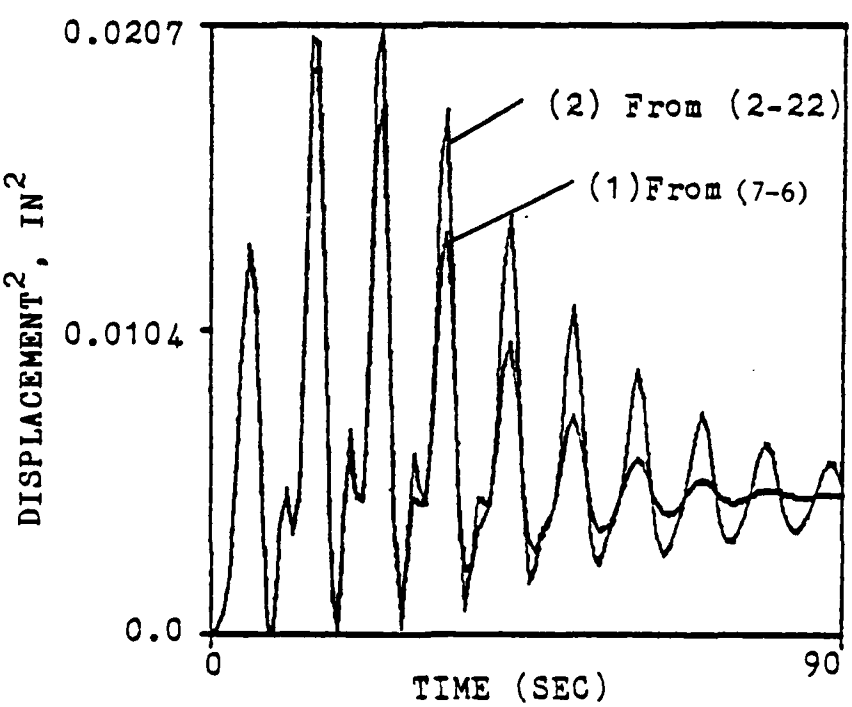

Figure 7.2 The comparison of variance between (1) including the higher order terms and (2) omission of the higher order terms

percent. The comparison of the variance response between the one from Equation (7-6) and the one from Equation (2-22) is also made and the result is plotted in Figure 7.2. The error is less than 8 percent.

From the above analysis, it can be concluded that the omission of the higher order term is admissible if the stiffness does not behave extremely random. The error occurred in variance response is higher than in the mean response.

Also, from Equation (7-5b), it is noted that the joint moment between K and Z(t) is related to the second moment of the stiffness only. In other words, the joint moment between K and Z(t) can be computed without any further assumption on the statistical property of the joint correlation between Z(t) and K. This satisfies the real situations which occurred in the nature since the correlation between stiffness and the response must exist in a natural sense which should be evaluated as long as the statistical properties of the stiffness and the excitation are known. Many studies [9] have assumed that the correlation between K and Z(t) is known which is not a reasonable assumption. The crossmoment between K and Z(t) is computed and the result is plotted in Figure 7.3. Figure 7.3 shows that K and Z(t) are negative correlated most of the time during the response cycle. This satisfies the physical phenomenon since the higher tendency of the stiffness will cause the lower magnitude of the response, as it should.

7.2 CONCLUSION

The present investigation has developed a structural damage theory to evaluate the mean and autocovariance functions, and the mean energy dissipation for the response of a structural system with random, potentially

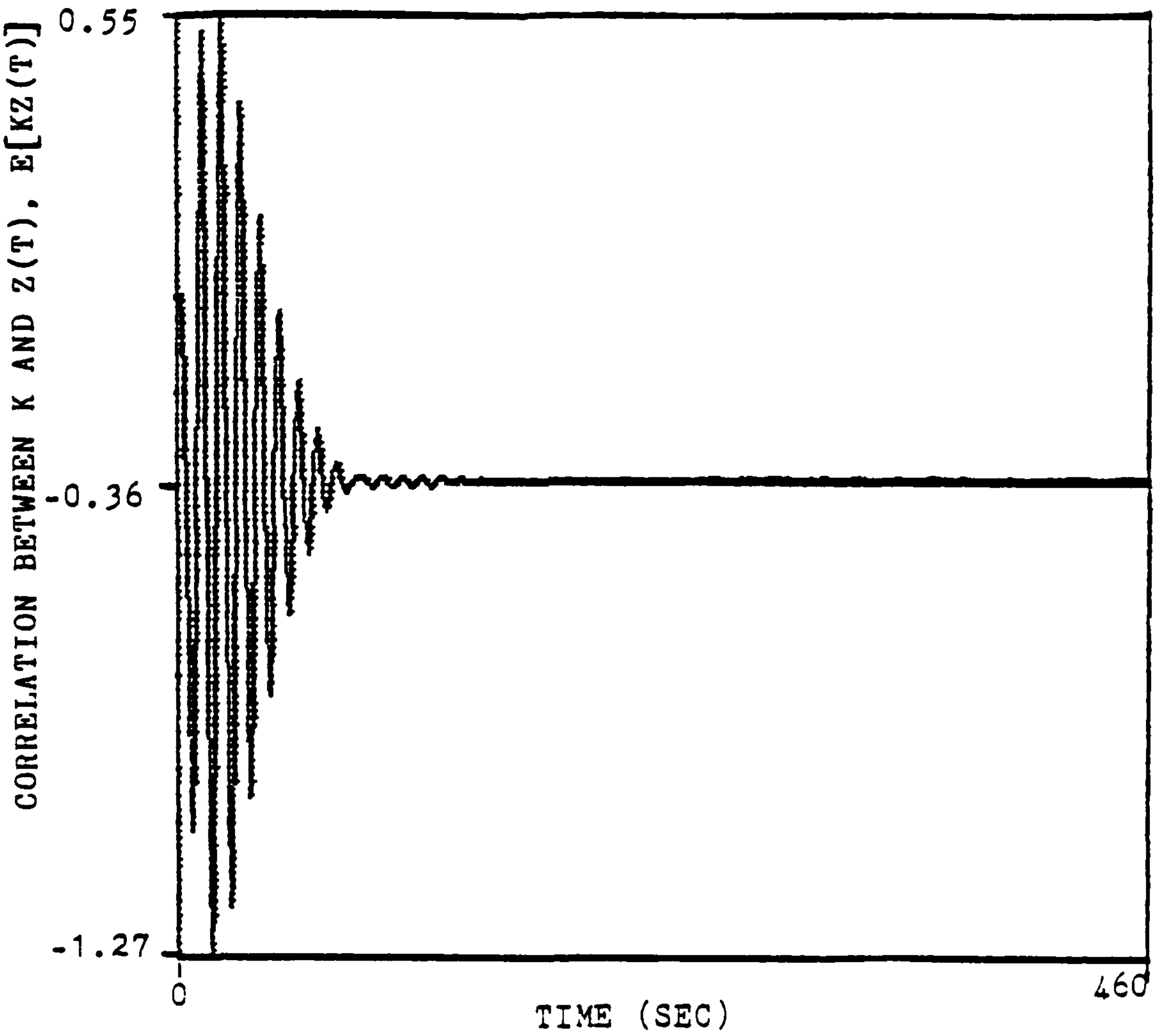

Figure 7.3 The correlation between K and Z(t)

elasto-plastic restoring force, subjected to random excitation. The problem is treated by shifting the stiffness matrix, the excitation and the response random processes about their means in such a way that the response is resolved into two parts with a differential equation governing each part. The first equation essentially governs the mean response. The second equation essentially characterizes the random component of the response. The response autocovariance function can be obtained using the equation that governs the random component of the response.

The measures of response considered in this investigation are displacement, velocity, acceleration and energy dissipation. The second order moments of displacement, velocity and acceleration are obtained. The cross-moments between displacement and velocity, velocity and acceleration, acceleration and displacement are also evaluated. The results are presented in discrete time. The first and second statistical moments in time history are plotted in specific examples. The theory developed here is applicable in the analysis of response to both stationary and nonstationary random excitations. It follows that the damage theory developed in this study can be used to assess the damaged, MDF structures. The computer program developed is based on the finite element method.

When nonlinearity appears in the structural restoring force function, it is assumed that the nonlinearity exists in material property. The material nonlinearity is assumed to be elasto-plastic. The cyclic elasto-plastic response of the structure is described in discrete time steps. At each step in the response computation, an equivalent stiffness matrix is obtained from the system restoring forces. These system restoring forces are obtained by iterating the permanent set at each time step.

The energy dissipation due to material nonlinearity is evaluated. However, only mean energy dissipation, which can be used to predict an upper bound on the total energy dissipation, is evaluated. When damage is assumed to be related to energy dissipated it can be assessed based on this analysis.

Several numerical examples are executed using the computer program FEDRANS which is written based on the theory developed in this study and appended in this report. The present theory can be applied to a generic class of random excitations. These include white and non-white types. A comparison of the present theory with published results is made for the case of stationary white noise excitation when the stiffness is deterministic. The results show good accuracy.

Other results of the numerical examples lead to the following conclusions. (1) The response autocovariance function for the nonlinear system has greater magnitude than the response autocovariance function for the linear system. For instance, while the mean response for the nonlinear system is 12 percent higher than that for the linear system, the standard deviation for the nonlinear system is 35 percent higher than that for the linear system. (2) The response autocovariance function for a system with random stiffness is greater than the response autocovariance function for a system with deterministic stiffness. For instance, while the stiffness shows 10 percent variation, the standard deviation of the response for a random stiffness is 5 times higher than that for a determinsitic stiffness system.

Therefore, the structural damage theory developed in this study provides a base to develop a method for damage diagnosis and reliability assessment of structural systems with random characteristics excited by random excitations. The formalism of the present report enables us to assess the damage for a generic class of MDF nonlinear system with elasto-plastic material. Further, the present investigation constitutes as part of the theory of damage diagnosis and makes the theory more satisfacotry.

The future study may include the following:

(1) Establishment of a damage model that allows the random component of the stiffness matrix to be a matrix of random processes rather than a matrix of random variables.

(2) Establishment of a damage model that involved the energy dissipation and maximum displacement. Also the establishment of the correlation between these two is desirable in the analysis of damage theory.

(3) Evaluation of the second moment of the energy dissipation.

(4) Consideration of system damping as a random quantity.

REFERENCES

1. Ju, F.D., Akgun, M., Wong, E.T., and Paez, T.L., "Modal Method in Diagnosis of Fracture Damage in Simple Structures," ASME 103rd WAM, 1982, Phoenix, AZ, Productive Applications of Mechanical Vibration, ASME publication, 1982, 113-125.

2. Ju, F.D., Akgun, M., Paez, T.L., and Wong, E.T., "Diagnosis of Fracture Damage in Simple Structures," UNM Technical Report, CE-62(82)AFOSR-993-1.

3. Ju, F.D. and Mimovich, M., "Experimental Diagnosis of Fracture Damage in Structures," 3rd Symposium on the Interaction of Non-nuclear Munitions with Structures, also M.S. Thesis, The University of New Mexico, Mechanical Engineering, 1985.

4. Ju, F.D., "Structural Dynamic Theories of Fracture Diagnosis," UNM Technical Report, ME-134(85)AFOSR-993-2, 1986.

5. Ju, F.D. et al., "Identification of Damage in Hysteretic Structures," UNM Technical Report, CE-72(84)AFOSR-993-1.

6. Ju, F.D. et al., "Diagnosis of Damage in SDF Structures," UNM Technical Report, CE-60(82)AFOSR-993-1, March 1982.

7. Ju, F.D. et al., "Mathematical Models for Damageable Structures," UNM Technical Report, CE-64(83)AFOSR-993-1, March 1983.

8. Lin, Y.K., Probabilistic Theory of Structural Dynamics, McGraw-Hill Book Company, New York, 1967.

9. Soong, T.T., Random Differential Equations in Science and Engineering, Academic Press, New York, 1973.

10. Crandall, S.H. and Mark, W.D., Random Vibration in Mechanical Systems, Academic Press, New York, 1973.

11. Newland, D.E., Random Vibrations and Spectral Analysis, Longman Group Limited, London and New York, 1975.

12. Crandall, S.H. (ed.), Random Vibration, MIT press, Cambridge, Mass., and John Wiley, 1958.

13. Crandall, S.H. (ed.), Random Vibration, Vol. 2, MIT press, Cambridge, Mass., 1963.

14. Clough, R.W. and Penzien, J., Dynamics of Structures, McGraw-Hill Book Company, New York, 1975.

15. Bogdanoff, J.L., "A New Cumulative Damage Model, Part 1," J. of Applied Mechanics, ASME, Vol. 45, 1978.

16. Bogdanoff, J.L., "A New Cumulative Damage Model, Part 2," J. of Applied Mechanics, ASME, Vol. 45, June, 1978.

17. Lin, Y.K. and Yang, J.N., "On Statistical Moments of Fatigue Crack Propagation," Journal of Engineering Fracture Mechanics, Vol. 18, 1983, pp. 243-256.

18. Lin, Y.K. and Yang, J.N., "A Stochastic Theory of Fatigue Crack Propagation," Proceedings, Part I, AIAA/ASME/AHS 24th Structures Structural Dynamics and Materials Conference, Lake Tahoe, NV, 1983, pp. 555-562. Also in AIAA Journal, Nov. 1984.

19. Lin, Y.K. and Wu, W.F., "Applications of Cumulant Closure to Random Vibration Problems," ASME publications, AMD-Vol. 65.

20. Wiggins, H.H., Jr., and Moran, D.F., "Earthquake Safety in the City of Long Beach Based on the Concept of Balanced Risk," J.H. Wiggins Company, Redondo Beach, CA, September 1971.

21. Ang, H-S. and Weng, Y.K., "Prediction of Structural Damage Under Random Earthquake Excitation," Earthquake Ground Motion and its effects on Structures, ASME, AMD, Vol. 53, November 1982.

22. Lin, Y.K., "On First-Excursion Failure of Randomly Excited Structures," II, AAA Journal, Vol. 8, No. 10, 1970, pp. 1868-1890.

23. Yang, J.N. and Shinozuka, M., "On the First Excursion Probability in Stationary Narrow-Band Random Vibration," Journal of Applied Mechanics, ASME, Vol. 38, No. 4, 1971.

24. Robert, J.B., "An Approach to the First Passage Problem in Random Vibration," Journal of Sound and Vibration, 8: 301-328 (1969).

25. Bernard, M.C. and Shipley, J.W., "The First Passage Problem for Stationary Random Structural Vibration," Journal of Sound and Vibration, 24: 121-130 (1972).

26. Lennox, W.C. and Fraser, D.A., "On the First Passage Distribution for the Envelope of Nonstationary Narrow Band Stochastic Process," Journal of Applied Mech., 41(3): 793-797 (1974).

27. Ariaratnam, S.T. and Pi, H.N., "On the First Passage Time for Envelope Crossing for a Linear Oscillator," Int. J. Control, 18(1): 89-96 (1973).

28. Sahay, B. and Lennox, W., "Moments of the First Passage for a Linear Oscillator," Journal of Sound and Vibration, 34(4): 449-458 (1974).

29. Paez, T.L. and Yao, J.T.P., "Probabilistic Analysis of Elasto-Plastic Structures," Journal of the Engineering Mechanics, ASCE, Bol. 102, No. EM1, Proc. Paper 11974, Feb. 1976, pp. 105-120.

30. Ang, A.H.S., "Probabilistic Concepts in Earthquake Eng.," ASME Special Conference, 1974, pp. 225-259.

31. Yao, J.T.P., "Damage Assessment and Reliability Evaluation of Existing Structures," Journal of Structural Engineering, Vol. 1, October 1979, pp. 245-251.

32. Park, Y.J. and Ang, A.H.S., "Mechanistic Seismic Damage Model for Reinforced Concrete," Journal of Structural Engineering, Vol. 111, No. 4, April 1985.

33. Lin, Y.K., Probablistic Theory of Structural Dynamics, McGraw-Hill Book Company, New York, 1967.

34. Rice, S.O., Mathematical Analysis of Random Noise, Bell System Tech. J., 23: 282-332 (1944); 24: 46-156 (1945). Reprinted in N. wax (ed.), Selected Papers on Noise and Stochastic Processes, New York, 1954.

35. Middleton, An Introduction to Statistical Communication Theory, p. 426, McGraw-Hill, New York, 1960.

36. Miner, M.A., Cumulative Damage in Fatigue, J. Appl. Mech., 12:A159-A164 (1945).

37. Coffin, L.F., Jr., Low-cycle Fatigue: A Review, Appl. Mater. Res., 1/3:129-141 (1962).

38. Ju, F.D., Yao, J.T.P. and Liu, T.T., Air Force Conference on Fatigue and Fracture of Aircraft Structures and Material, Miami Beach, Res., 1969. Proceeding AFFDL TR 70-144, Sep. 1970, pp. 265-270.

39. Wang, M.L., Identification of Damage in Hysteretic Structures, Dissertation, UNM, Albuquerque, NM, 1983.

40. Ross, S., A First Course in Probability, 2nd ed., p. 320, Macmillan Publishing Company, New York, 1984.

41. Bathe, K.J. and Wilson, E.L., Numerical Method in Finite Element Analysis, Prentice-Hall, Inc., 1976.

APPENDIX A

USER MANUAL OF FEDRANS

1. TITLE (20A4)

1-80	TITLE	Any alphanumeric identifier

2. CONTROL PARAMETERS (11I5)

1-5	METHOD	Static = 1, Dynamic = 2
6-10	KIND	1 Plane truss
		2 Plane frame
		3 Space truss
		4 Space frame
11-15	NUMNP	Total # of nodes(including supports)
16-20	NSTRUT	# of Two-force members
21-25	NBEAM	# of Beam members
26-30	NMTYPE	# OF MEMBER TYPES IN MEMBER LIBRARY (A, I,...)
31-35	NUMEM	Total # of members
36-40	NUMAT	# OF DIFFERENT MATERIALS (E, POISSION RATIO, MASS DENSITY,...)
41-45	NRNP	# of constraint (support) nodes
46-50	NMREL	# of member releases
51-55	NLC	# of loading conditions = 0, For dynamic analysis (no use for dynamic analysis, but still need input)

3. PRINT OUT CONTROL CARD (3I5)

1-5	WSTIF	Print out stiffness matrix(0=NO,1=YES)

Also control the printing out for all the calculation of intermediate steps in random analysis. Also in nonlinear analysis, it controls the offset print out. For every case, 0: NO, 1: Yes

6-10	WRSTIF	Print out reduced stiffness matrix Also control the stiffness print out for nonlinear analysis.
11-15	WLOD	Print out Loading conditions Also control the final offset print out for every case, 0: No, 1: Yes

Card 4-6 are read only for METHOD = 2

4. DYNAMIC CONTROL CARD (3I5)

(This card read only for METHOD=2, otherwise, skip this card, go directly to card 7)

1-5	MASS	0: No concentrated mass added >1: add MASS times concentrated mass
6-10	INLOD	1=Loading function read from data card 2=Read from data file
11-15	MDI	Method of direct integration 1=Newmark Method 2=Central difference method

5. CONCENTRATED MASS CARD (4(i5,e10.5))

(read only for mass >0, otherwise, skip this card)

6. DAMPING CONTROL CARD (2F10.4)

1-10	ALPHA	C = ALPHA * M + BETA * K
11-20	BETA	

7. MEMBER PROPERTY TABLE (6E10.4)

Repeat NMTYPE times

1-10	tf(im)	(For rectangular
11-20	bf(im)	crossection, use
		bw=4*tf; tw=bf)
21-30	tw(im)	
31-40	bw(im)	
41-50	XJ(im)	Torsional constant Ix
51-60	AVY(im)	Effective shear area for shear in Y-Dir
61-70	AVZ(im)	Effective shear area for shear in Z-Dir
71-72	MCURV(im)	Member curvature index.
		0=straight
		1=curve

8. MATERIAL DATA (3E10.4)

Repeat NUMAT times, one card for each different material

1-10	E(im)	Young's modulus of elasticity
11-20	sigma(im)	yield stress
21-30	PR(im)	Poisson's ratio
31-40	RHO(im)	Mass density

9. NODAL POINT COORDINATES (i5,3E10.4)

Repeat NUMNP times (one card for each nodal point)

1-5	NN	Number of nodal point
6-15	x(nn)	x-coordinate (2-dim, use x-y plane)
16-25	Y(nn)	Y-coordinate
26-35	Z(nn)	Z-coordinate

10. MEMBER DESCRIPTIONS (6I5,3E10.4)

Repeat NUMEM times- one card for each member

1-5	ID	Member I.D., (L.E. NUMEM)
6-10	ND(id,1)	Nodal point number, node i
11-15	ND(id,2)	Nodal point number, node j
16-20	MTYPE(ID)	Type of member (from library)
21-25	MID(id)	Material I.D. number
26-30	MLC(id)	Not presently used
31-40	ALFA(id)	Angle of rotation of the principal y-axis
41-50	THETAI(id)	Specified rotation about z-axis
51-60	THETAJ(id)	Specified rotation about Z-axis

11. B.C. (I3,I1,,6I1,6F10.4)

1-3	N	Number of restrained node
4	NTYPE(n)	B.C. Type 0= No I.C.
5-10	IR(n,j)	A six digit integer identifying type of constraint. 1= constrain, 0=free (for plane frame, only 3 was read which is (x,y,zz))
11-20	UI(n,1)	Specified IC(translation in global x-dir)
21-30	UI(n,2)	in y-direction (Use only for static
31-40	UI(n,3)	in z-direction analysis, In dynamic
41-50	UI(n,4)	in x rotation analysis, read from
51-60	UI(n,5)	in y rotation card 23)
61-70	UI(n,6)	in z rotation

12. MEMBER RELEASE (I3,2I1)

Only for NMREL GT 0

These data are required for members with initially specified releases only. Repeat NMREL times (one card for each released member)

1-3	IM	Member number
4	MREL(im,1)	Release code for node i of member as specified im member data
5	MREL(IM,2)	Release code for node j of member The release code is specified by a two digit integer: either 1 or 0 is used depending on whether the I or J nodes of

the member are released or not released respectively.

13. LOADING PARAMETERS (2I5, 3F10.4)

Only for static analysis, otherwise, skip this card.

This card repeated NLC times, one for each load data set

1-5	NLND	Number of nodes with concentrated loads
6-10	NLMEM	Number of members having loads along their length between nodes
11-20	AX	Blank
21-30	AY	g-Acceleration in y-direction
31-40	AZ	Blank

14. CONCENTRATED NODAL LOADS (I5,6F10.4)

Only for static analysis, otherwise, skip this card

One card per loaded node for each loading condition is required.

1-5	NL	Node number
6-15	P(1)	Component of concentrated force in global x-direction
16-25	P(2)	Force in global y-direction
26-35	P(3)	Focce in global z-direction
36-45	P(4)	Component of concentrated couple about global x-axis
46-55	P(5)	Couple about global y-axis

56-65	P(6)	Couple about global z-axis

15. Member Load

Only for static analysis, otherwise, skip this card.

Two cards per loaded member (NLNEM members) are required for each loading dondition involing members with intermediate loads.

1-5	MN	Member number
6-10	I	Node number
11-20	P(1)	Fixed end axial force at node I
21-30	P(2)	Fixed end y-shear at node I
31-40	P(3)	Fixed end z-moment at node I
41-50	P(4)	Fixed end x-moment (torque) at node I
51-60	P(5)	Fixed end y-moment at node I
61-70	P(6)	Fixed end z-moment at node I

SECOND CARD (5x,I5,6f10.4)

1-5		Blank
6-10	I	Node number
11-20	P(7)	Fixed end axial force at node J
21-30	P(8)	Fixed end y-shear at node J
31-40	P(9)	Fixed end z-shear at node J
41-50	P(10)	Fixed end x-moment (torque) at node J
51-60	P(11)	Fixed end y-moment at node J
61-70	P(12)	Fixed end z-moment at node J

(Card 16 to 25, read only for dynamic anlaysis)

16. DYNAMIC LOAD INPUT (2I5,f10.4)

1-5	IRANDM	Random analysis, =1: No, =2: yes
6-10	LON	Linear or nonlinear analysis
		=1: Linear, 2: Nonlinear
11-15	NDOFL	Total # dof that have load
16-20	NTI	Total # of time increments
21-30	DT	Delta t

17. LOAD DESCRIPTION (repeat NDOFL times) (2I5)

1-5	NDOF(i)	# of dof that has load acting
6-10	NTPDL	# of points when card is applied.
		(see next card for the meaning of ns(i,j))

18. LOADS DESCRIPTION 5(I5,f10.5)

Repeat NTPDL TImes

1-5	NS(ndof(i),1)	# increment of time when load is specified.
6-15	DL(ndof(i),1)	corresponding amplitude
16-20	NS(ndof(i),2)	same as above
21-30	DL(ndof(i),2)	
31-35	NS(ndof(i),3)	(if necessary)
36-46	DL(ndof(i),3)	

19. CROSS TERM CONTROL CARD

This card read only when IRANDM=2

1-5	ICROS	Flag indicating compute cross moment 0:NO, 1:YES
6-10	IVV	Flag indicating compute E(VV) 0:NO, 1:YES
11-15	IAA	Flag indicating compute E(AA) 0:NO, 1:YES
16-20	IVZ	Flag indicating compute E(VZ) 0:NO, 1:YES
21-25	IAZ	Flag indicating compute E(AZ) 0:NO, 1:YES
26-30	IAV	Flag indicating compute E(AV) 0:NO, 1:YES

Card 20-22, read only for IRANDM=2, otherwise, skip it

20. COVARIANCE MATRIX INPUT CONTROL CARD AND MODAL ANALYSIS CONTROL CARD

1-5	NCVDF	Total pairs # DOF in covariance matrix
6-10	NROOT	Number of eigenvectors required to represent the mean response
11-15	NSMAX	Maximum number of sweeps allowed in subroutine JACOBI
16-25	CORRI	Coefficient of variation of Young's modulus (beta)

21. PAIRS DESCRIPTION

Repeat NCVDF times, each time read one card

1-5	N1(I)	Pair for DOF
6-10	N2(I)	
11-20	NTACV	Total # needed to describe Ax in time domain

22. AMPLITUDE DESCRIPTION IN COVARIANCE MATRIX

Repeat NTACV Times

1-5	NT(n1(i),n2(i),j)	Time increment that have amplitude change
6-15	AX(n1(i),n2(i),j)	The corresponding amplitude

23. PRINT OUT DETAIL CONTROL CARD (7I5)

1-5	INPT	Initial # dof for print out
6-10	IFPT	Finial # dof for print out
11-15	JPNP	Jump number for Print out
16-20	INTPT	Initial # for Time print out
21-25	IFTPT	Final # for time print out
26-30	JPNT	Jump increment for print out
31-35	INIT	Initial condition 0=Generate by program 1=Read from data card
36-40	IPLOT	Plot control 0=direct print out 1=write into a data file

24. I.C. (only for INIT=1)

Repeat NUMEQ times

1-10	xdp(ii,1)
11-20	x1d(ii,1)
21-30	x2d(ii,1)

25. This card inputs a trial number in sub-space iteration. Usually, set it to unity. If the object matrix is not positive define, change to another value until the object stiffness matrix is positive definite. A better way in second try is to set this number as negative.

(Random format) TRIX

APPENDIX B

COMPUTER PROGRAM "FEDRANS"

Finite

Element

Dynamic and

Random

Analysis for

Nonlinear

Systems

```
      subroutine beam (id,i,j)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      double precision s,r,t,st,tf,s1,s2,s3,s4,s5,sl,dsqrt,x1,x2,y1,
     1                 y2,z1,z2
      x1=x(i)
      x2=x(j)
      y1=y(i)
      y2=y(j)
      z1=z(i)
      z2=z(j)
      sl=dsqrt((x2-x1)**2+(y2-y1)**2+(z2-z1)**2)
c          generate element stiffness matrix (S)
      m=mid(id)
      n=mtype(id)
      ym=e(m)
      sm=g(m)
      s1=xa(n)*ym/sl
      s2=sm*xj(n)/sl
      s3=ym*zi(n)/sl
      s4=ym*yi(n)/sl
      s5=3.*ym*zi(n)/(sl**3)
      do 10 ii=1,neq
      do 10 jj=1,neq
   10 s(ii,jj)=0.0
   12 s(1,1)=s1
      s(1,7)=-s1
      s(3,3)=12.*s4/(sl**2)
      s(3,5)=-6.*s4/sl
      s(3,9)=-s(3,3)
      s(3,11)=s(3,5)
      s(4,4)=s2
      s(4,10)=-s2
      s(5,5)=4.*s4
      s(5,9)=6.*s4/sl
      s(5,11)=2.*s4
      s(7,7)=s1
      s(9,9)=12.*s4/(sl**2)
      s(9,11)=6.*s4/sl
      s(10,10)=s2
      s(11,11)=4.*s4
      if (mrel(id,1).ne.0.and.mrel(id,2).ne.0) go to 35
      if (mrel(id,1).ne.0.and.mrel(id,2).eq.0) go to 30
      if (mrel(id,1).eq.0.and.mrel(id,2).ne.0) go to 25
c          full continuity
      s(2,2)=12.*s3/(sl**2)
      s(2,6)=6.*s3/sl
```

```
      s(2,8)=-s(2,2)
      s(2,12)=s(2,6)
      s(6,6)=4.*s3
      s(6,8)=-6.*s3/sl
      s(6,12)=2.*s3
      s(8,8)=12.*s3/(sl**2)
      s(8,12)=-6.*s3/sl
      s(12,12)=4.*s3
      go to 35
c          hinge right end
   25 s(2,2)=s5
      s(2,6)=s5*sl
      s(2,8)=-s5
      s(6,6)=s(2,6)*sl
      s(6,8)=-s(2,6)
      s(8,8)=s5
      go to 35
c          hinge left end
   30 s(2,2)=s5
      s(2,8)=-s5
      s(2,12)=s5*sl
      s(8,8)=s5
      s(8,12)=-s(2,12)
      s(12,12)=s(2,12)*sl
c          symmetrize (S)
   35 do 15 ii=1,neq
      do 15 jj=ii,neq
   15 s(jj,ii)=s(ii,jj)
c          transform (S) to global coordinates
   14 call rotate (1,id)
      return
      end
```

```
      subroutine beam2 (id,i,j)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1             avz(25),mcurv(25)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      common /xmss/ xms(12)
      double precision s, r,t,st,tf,s1,s2,s3,sl,dsqrt,xp,yp,xms
      xp=x(j)-x(i)
      yp=y(j)-y(i)
      sl=dsqrt(xp**2+yp**2)
c        generate the element stiffness matrix (s)
      m=mid(id)
      n=mtype(id)
      ym=e(m)
      area=xa(n)
      rh=rho(m)
      s1=xa(n)*Ym/sl
      s2=zi(n)*ym/sl
      s3=3.*ym*zi(n)/(sl**3)
      do 10 ii=1,neq
      do 10 jj=1,neq
   10 s(ii,jj)=0.
      s(1,1)=s1
      s(1,4)=-s1
      s(4,4)=s1
      if (mrel(id,1).ne.0.and.mrel(id,2).ne.0) go to 35
      if (mrel(id,1).ne.0.and.mrel(id,2).eq.0) go to 30
      if (mrel(id,1).eq.0.and.mrel(id,2).ne.0) go to 25
c           full continuity
      s(2,2)=12.*s2/(sl**2)
      s(2,3)=6.*s2/sl
      s(2,5)=-s(2,2)
      s(2,6)=s(2,3)
      s(3,3)=4.*s2
      s(3,5)=-6.*s2/sl
      s(3,6)=2.*s2
      s(5,5)=s(2,2)
      s(5,6)=-s(2,3)
      s(6,6)=4.*s2
      go to 35
c       hinge reight end
   25 s(2,2)=s3
      s(2,3)=s3*sl
      s(2,5)=-s3
      s(3,3)=s(2,3)*sl
      s(3,5)=-s(2,3)
      s(5,5)=s3
c        hinge left end
   30 s(2,2)=s3
```

```
      s(2,5)=-s3
      s(2,6)=s3*sl
      s(5,5)=s3
      s(5,6)=-s(2,6)
      s(6,6)=s(2,6)*sl
c         symmertize (s)
 35   do 15 ii=1,neq
      do 15 jj=ii,neq
 15   s(jj,ii)=s(ii,jj)
c         transform  (s) to global coordinates
      call rotate (1,id)
c         Form Mass matrix
      go to (99,101) ,method
  101 do 100 ii=1,neq
      xms(ii)=0.
 100  continue
      c1=rh*area*sl*.5
      c2=rh*area*sl**3/24.
      xms(1)=c1
      xms(2)=c1
      xms(4)=c1
      xms(5)=c1
      xms(3)=c2
      xms(6)=c2
 99   return
      end
```

```
      subroutine bound(ibnd,high)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      double precision a,b,xmass,damp
      go to (20,25), ibnd
   20 nhw=mband-1
      do 60 n=1,numnp
c        compute control counters
      do 5 kk=1,nq
      k=nq*n-(nq-kk)
      j1=k-nhw
      if(j1) 10,10,15
  10  ii=1
      go to 30
  15  ii=j1
c        reduce row of (A)
  30  if (ir(n,kk).eq.0)  go to 5
      if (high.Le.1.) go to 81
      a(k,1)=a(k,1)*high
      go to 5
  81  go to (35,34), method
  34  damp(k,1)=1.0
  35  a(k,1)=1.0
c        reduce mass matrix
      go to (37,36) ,method
  36  xmass(k)=1.0
  37  do 40 j=1,nhw
      l=j+1
      go to (40,48), method
  48  damp(k,l)=0.0
  40  a(k,l)=0.0
c           reduce column of (A)
      if(k-1) 5,5,50
   50 jj=k-1
      do 55 j=ii,jj
      k1=k-j+1
      go to (55,54), mehtod
   54 damp(j,k1)=0.
   55 a(j,k1)=0.0
    5 continue
   60 continue
      go to 85
c           reduce (B)
   25 do 80 n=1,numnp
   65 do 70 kk=1,nq
      k=nq*n-(nq-kk)
      if (ir(n,kk).eq.0) go to 70
   75 b(k)=ui(n,kk)
   70 continue
   80 continue
   85 continue
```

```
return
end
```

```
      subroutine choles(it)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/chol/xmc(50,25)
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      double precision xmc,diag1,air1,sum1,dsqrt,xdp,x1d,x2d
c
      if (it.ge.2) go to 123
      nhw=mband-1
      n=numeq
  100 nred=0
      itrig=0
      lim=mband
  101 if(nred+1-n) 102,500,500
  102 nred=nred+1
      diag1=xmc(nred,1)
      if (diag1-1.0d-30) 601,601,110
  110 diag1=dsqrt(diag1)
c           go to 601 if matrix is singular or not positive define
c           divide roe by square root of diagonal element
  111 do 113 j=1,lim
  113 xmc(nred,j)=xmc(nred,j)/diag1
c           reduce remaining block of numbers
  201 do 251 i=1,nhw
      l=nred+i
      if(l-n) 211,211,251
  211 air1=xmc(nred,i+1)
c           skip this row if multiplier air is zero
      if(air1) 221,251,221
  221 do 231 j=i,nhw
      m=l+j-i
  231 xmc(l,m)=xmc(l,m)-air1*xmc(nred,j+1)
  251 continue
      go to 101
  601 itrig=nred
  500 if(itrig) 600,610,600
c           singular matrix
  600 write(6,602) itrig
  602 format(1x,'Singular Matrix at Cholesk@a NRED =',i4)
  610 continue
c           reduce the right hand sides
  123 continue
      nred=0
  301 if(nred+1-n) 302,401,401
  302 nred=nred+1
c           divide row by square root of diagonal element
      xdp(nred,3)=xdp(nred,3)/xmc(nred,1)
c           reduce remaining block of numbers
      do 351 i=1,nhw
      l=nred+i
      if(l-n) 311,311,351
  311 xdp(l,3)=xdp(l,3)-xmc(nred,i+1)*xdp(nred,3)
  351 continue
```

```
      go to 301
c            back substitution
  401 xdp(n,3)=xdp(n,3)/xmc(n,1)
      n1=n-1
      do 451 ii=1,n1
      i=n-ii
      sum1=0.0
      do 421 jj=1,nhw
      m=jj+i
      if(n-m) 451,421,421
  421 sum1=sum1+xmc(i,jj+1)*xdp(m,3)
  451 xdp(i,3)=(xdp(i,3)-sum1)/xmc(i,1)
   25 continue
  700 continue
      return
      end
```

```
      subroutine curvbm (id,i,j)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mlib/xa(25),zi(25),yi(25),rr(25),avy(25),
     1            avz(25),mcurv(25)
      common/mem/mtype(50),no(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mat/ee(5),sigma(5),epsiln(5),pr(5),gg(5),rho(5)
      common/stif/ s(12,12),q(3,3),t(12,12),st(12,12),tf(12),p(12)
      double precision s,q,t,st,tf,r,beta,cb,sb,a1,b1,c1,d1,e1,
     1                 a,b,c,d,e,f,g,s1,dsin,dcos,thi,thj,dabs
      m=mid(id)
      n=mtype(id)
      ym=ee(m)
      r=rr(n)
      thi=thetai(id)/57.2957795
      thj=thetaj(id)/57.2957795
      beta=dabs(thj-thi)
      do 10 ii=1,neq
      do 10 jj=1,neq
   10 s(ii,jj)=0.0
c                generate element stiffness matrix (S)
      cb=dcos(beta)
      sb=dsin(beta)
      a1=beta-sb
      b1=cb+0.5*sb*sb-1.0
      c1=1.5*beta-2.0*sb+(dsin(2.0*beta))/4.0
      d1=1.5*beta-(dsin(2.0*beta))/4.0
      e1=cb-1.0
      a=e1*e1/beta-d1
      b=b1-a1*e1/beta
      c=a1*d1-b1*e1
      d=a1*a1/beta-c1
      e=c1*e1-a1*b1
      f=b1*b1-c1*d1
      g=b1*(b1-2.*a1*e1/beta)+c1*(e1*e1/beta-d1)+a1*a1*d1/beta
      s1=(ym*zi(n)/g)/(r**3)
      s(1,1)=s1*a
      s(1,2)=s1*b
      s(1,3)=s1*c*r/beta
      s(1,4)=-s1*(a*cb+b*sb)
      s(1,5)=s1*(a*sb-b*cb)
      s(1,6)=s1*(a*(cb-1.)+b*sb-c/beta)*r
      s(2,2)=s1*d
      s(2,3)=s1*e*r/beta
      s(2,4)=-s1*(b*cb+d*sb)
      s(2,5)=s1*(b*sb-d*cb)
      s(2,6)=s1*(b*(cb-1.)+d*sb-e/beta)*r
      s(3,3)=s1*f*r*r/beta
      s(3,4)=-s1*(c*cb+e*sb)*r/beta
      s(3,5)=s1*(c*sb-e*cb)*r/beta
      s(3,6)=s1*(c*(cb-1.)+e*sb-f)*r*r/beta
      s(4,4)=s1*a
```

```
      s(4,5)=-s1*b
      s(4,6)=s1*c*r/beta
      s(5,5)=s1*d
      s(5,6)=-s1*e*r/beta
      s(6,6)=s1*f*r*r/beta
c        symmetrize (S)
      do 15 ii=1,6
      do 15 jj=1,6
      s(jj,ii)=s(ii,jj)
   15 continue
c        Transform (S) to global coordinates
      st1=dsin(thi)
      ct1=dcos(thi)
      st2=dsin(thj)
      ct2=dcos(thj)
      do 20 ii=1,6
      do 20 jj=1,6
   20 t(ii,jj)=0.
      t(1,1)=ct1
      t(1,2)=st1
      t(2,1)=-st1
      t(2,2)=ct1
      t(3,3)=1.
      t(4,4)=ct2
      t(4,5)=st2
      t(5,4)=-st2
      t(5,5)=ct2
      t(6,6)=1.
      call rotate (1,id)
      return
      end
```

```
      subroutine d1 (i)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      double precision a,b,xmass,damp,xdp,x1d,x2d,t1,t2,f
      double precision df,xam1,xam2
c        Find the forcing fn at i'th step
      do 150 ii=1,numeq
c        Pick out # DOF that has load
      do 160 jj=1,ndofl
      if (ii.eq.ndof(jj)) go to 161
  160 continue
      f(ii)=0.
      go to 150
c       Pick out n'th point that has amplitude change
  161 lc=la(ii)
      if (i.eq.ns(ii,lc)) go to 170
      df=(dl(ii,lc)-dl(ii,lc-1))/(ns(ii,lc)-ns(ii,lc-1))
      f(ii)=dl(ii,lc-1)+df*(i-ns(ii,lc-1))
      go to 150
  170 f(ii)=dl(ii,lc)
      la(ii)=la(ii)+1
  150 continue
      if (wstif.eq.0) go to 155
      write (6,2)
    2 format (' the forcing function is ')
      write (6,1) (f(ii),ii=1,numeq)
    1 format (6e13.5)
c       Form effective load vector
c       {[ K ]-2/dt**2*[M]}, and 1/dt** [M]-.5*dt*[C]
  155 do 200 ii=1,numeq
      t1(ii)=0.
      t2(ii)=0.
      call range (ii,numeq,mband,mg,mbd)
      do 200 ij=mg,mbd
      ms=ij-ii+1
      if (ms) 201,201,202
  201 t2(ii)=t2(ii)+a(ij,ii-ij+1)*xdp(ij,2)
      t1(ii)=t1(ii)-damp(ij,ii-ij+1)*xdp(ij,1)*.5/dt
      go to 200
  202 if (ms.eq.1) go to 203
      xam2=a(ii,ms)
      xam1=-damp(ii,ms)*.5/dt
      go to 204
  203 xam2=a(ii,ms)-2.*xmass(ii)/dt**2
      xam1=-damp(ii,ms)*.5/dt+xmass(ii)/dt**2
  204 t2(ii)=t2(ii)+xam2*xdp(ij,2)
      t1(ii)=t1(ii)+xam1*xdp(ij,1)
  200 continue
```

```
      do 210 ii=1,numeq
      xdp(ii,3)=f(ii)-t1(ii)-t2(ii)
  210 continue
      call  choles(i)
c           calculate velocity and acceleration
      do 220 ii=1,numeq
      x2d(ii,1)=(xdp(ii,3)-2.*xdp(ii,2)+xdp(ii,1))/dt**2
      x1d(ii,1)=(xdp(ii,3)-xdp(ii,1))/(2.*dt)
  220 continue
      return
      end
```

```
      subroutine d2 (i)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),ax(50,50,10),
     1               nt(50,50,10),lon,nroot,nsmax,corri
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndof1
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      double precision a1,a2,a3,xla,rf,rz,rz1,rz12,zz,z12,z2,ddx
c        Pick out Rff at time step i
      icount=1
      do 410 ii=1,numeq
      do 410 jj=1,numeq
      do 471 kk=icount,ncvdf
      if (ii.eq.n1(kk).and.jj.eq.n2(kk)) go to 411
  471 continue
      rf(ii,jj)=0.
      go to 410
  411 icount=icount+1
      Lc1=La2(ii,jj)
      if (i.eq.nt(ii,jj,lc1)) go to 412
c               Linear interpolation
      ddx=(ax(ii,jj,Lc1)-ax(ii,jj,Lc1-1))/
     1    (nt(ii,jj,Lc1)-nt(ii,jj,Lc1-1))
      rf(ii,jj)=(ax(ii,jj,Lc1-1)+ddx*(i-nt(ii,jj,Lc1-1)))*dt**4
      go to 410
  412 rf(ii,jj)=ax(ii,jj,lc1)*dt**4
  470 la2(ii,jj)=la2(ii,jj)+1
  410 continue
c          Evaluate A2 zz(i,j,2) A2(T)
      do 420 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 420 jj=1,numeq
      xla(ii,jj)=0.
      if (a2(ii,ii).eq.1.0 .or.a2(jj,jj).eq.1.) go to 420
      do 421 kk=mg,mbd
      xla(ii,jj)=xla(ii,jj)+a2(ii,kk)*zz(kk,jj,2)
  421 continue
  420 continue
      do 425 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 425 jj=ii,numeq
      rz(ii,jj)=0.
      if (a2(ii,ii).eq.1.0.or.a2(jj,jj).eq.1.) go to 425
      do 426 kk=mg,mbd
      rz(ii,jj)=rz(ii,jj)+a2(ii,kk)*xla(jj,kk)
  426 continue
  425 continue
      if (wstif.eq.0) go to 150
      write (6,1)
    1 format ('rz(i,j)')
      do 3 ii=1,numeq
```

```
    3 write (6,2) (rz(ii,jj),jj=1,numeq)
    2 format (6e13.5)
c           Evaluate A3 zz(i,j,1) A3(T)
  150 do 430 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 430 jj=1,numeq
      xla(ii,jj)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.) go to 430
      do 431 kk=mg,mbd
      xla(ii,jj)=xla(ii,jj)+a3(ii,kk)*zz(kk,jj,1)
  431 continue
  430 continue
      do 435 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 435 jj=ii,numeq
      rz1(ii,jj)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.) go to 435
      do 436 kk=mg,mbd
      rz1(ii,jj)=rz1(ii,jj)+a3(ii,kk)*xla(jj,kk)
  436 continue
  435 continue
      if (wstif.eq.0) go to 151
      write (6,4)
    4 format ('rz1(i,j)')
      do 5 ii=1,numeq
    5 write (6,2) (rz1(ii,jj),jj=1,numeq)
c           temporary sum
  151 do 520 ii=1,numeq
      do 520 jj=ii,numeq
  520 rf(ii,jj)=rf(ii,jj)+rz(ii,jj)+rz1(ii,jj)
      return
      end
```

```
      subroutine d3
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),ax(50,50,10),
     1               nt(50,50,10),lon,nroot,nsmax,corri
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/slv/ a(50,25),b(50)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndof1
      common/matrix/ la(50),t1(50),t2(50),f(50)
      common/kuz/ rx(50,50,3,8),ck(8)
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      double precision a,b,xdp,x1d,x2d,t1,t2,f,rx,eigvec,rho,a4,ck
      double precision a1,a2,a3,xla,rf,rz,rz1,rz12,zz,z12,z2
c        Evaluate K b Z(j)T
      do 465 np=1,nroot
c        first: K b xdp(i,1)T K
      do 445 ii=1,numeq
      t1(ii)=0.
      call range (ii,numeq,mband,mg,mbd)
      do 445 jj=mg,mbd
      if (xdp(jj,1).eq.0.) go to 445
      ms=jj-ii+1
      if (ms) 446, 446, 447
  446 t1(ii)=t1(ii)+a(jj,ii-jj+1)*eigvec(jj,np)
      go to 445
  447 t1(ii)=t1(ii)+a(ii,ms)*eigvec(jj,np)
  445 continue
      do 300 ii=1,numeq
      do 300 jj=1,numeq
  300 xla(ii,jj)=t1(ii)*xdp(jj,1)
      do 310 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 310 jj=1,numeq
      rz(ii,jj)=0.
      if (a(ii,1).eq.1..or.a(jj,1).eq.1.) go to 310
      do 311 kk=mg,mbd
      ms=jj-kk+1
      if (ms) 315,315,316
  315 rz(ii,jj)=rz(ii,jj)+xla(ii,kk)*a(jj,kk-jj+1)*(corri*dt)**2
      go to 311
  316 rz(ii,jj)=rz(ii,jj)+xla(ii,kk)*a(kk,ms)*(corri*dt)**2
  311 continue
  310 continue
c        second: K b Z(j-1)T A2(j-1)(T) and K b Z(j-2)T A3(T)
      do 450 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 450 jj=1,numeq
      xla(jj,ii)=0.
      if (a4(ii,ii).eq.1.0.or.a4(jj,jj).eq.1.) go to 450
      do 451 kk=mg,mbd
```

```
      xla(jj,ii)=xla(jj,ii)+a4(ii,kk)*rx(jj,kk,2,np)
  451 continue
  450 continue
      do 150 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 150 jj=1,numeq
      z12(jj,ii,2)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.) go to 150
      do 151 kk=mg,mbd
      z12(jj,ii,2)=z12(jj,ii,2)+a3(ii,kk)*rx(jj,kk,1,np)
  151 continue
  150 continue
c         third: sum of above
      do 455 ii=1,numeq
      do 455 jj=1,numeq
      xla(ii,jj)=xla(ii,jj)+z12(ii,jj,2)-rz(ii,jj)
  455 continue
c         fourth: sum * A1(-T)
      do 460 ii=1,numeq
      do 460 jj=1,numeq
      rx(ii,jj,3,np)=0.
      if (a1(jj,jj).eq.1.0.or.a1(ii,ii).eq.1.) go to 460
      do 461 kk=1,numeq
      rx(ii,jj,3,np)=rx(ii,jj,3,np)+xla(ii,kk)*a1(jj,kk)
  461 continue
  460 continue
  465 continue
c         Sum of above
      do 510 ii=1,numeq
      do 510 jj=1,numeq
      z2(ii,jj,2)=0.
      rz(ii,jj)=0.
      do 510 np=1,nroot
      z2(ii,jj,2)=z2(ii,jj,2)+rx(ii,jj,3,np)*ck(np)
  510 rz(ii,jj)=rz(ii,jj)+rx(ii,jj,2,np)*ck(np)
      if (wstif.eq.0) return
      write (6,1)
    1 format ('z2(i,j,2) matrix, K xdp Z(j)')
      do 4 ii=1,numeq
    4 write (6,2) (z2(ii,jj,2),jj=1,numeq)
    2 format (6e13.5)
      write (6,3)
    3 format ('rz(i,j) matrix, K xdp Z(j-1)')
      do 5 ii=1,numeq
    5 write (6,2) (rz(ii,jj),jj=1,numeq)
      return
      end
```

```
      subroutine d4
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),ax(50,50,10),
     1               nt(50,50,10),lon,nroot,nsmax,corri
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),d1(50,10),ndof(10),ndof1
      common/slv/ a(50,25),b(50)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      double precision a,b,xdp,x1d,x2d.t1,t2,f,eigvec,rho,a4
      double precision a1,a2,a3,xla,rz,rz1,rz12,z12,zz,rf,z2
c           Evaluate K xdp xdp(T) K(T)
      do 515 ii=1,numeq
      t2(ii)=0.
      call range (ii,numeq,mband,mg,mbd)
      do 515 jj=mg,mbd
      if (xdp(jj,2).eq.0.) go to 515
      ms=jj-ii+1
      if (ms) 516,516,517
  516 t2(ii)=t2(ii)+a(jj,ii-jj+1)*xdp(jj,2)
      go to 515
  517 t2(ii)=t2(ii)+a(ii,ms)*xdp(jj,2)
  515 continue
      do 300 ii=1,numeq
      do 300 jj=1,numeq
  300 xla(ii,jj)=t2(ii)*xdp(jj,2)
      do 310 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 310 jj=1,numeq
      rz1(ii,jj)=0.
      if (a(ii,1).eq.1..or.a(jj,1).eq.1.) go to 310
      do 311 kk=mg,mbd
      ms=jj-kk+1
      if (ms) 315,315,316
  315 rz1(ii,jj)=rz1(ii,jj)+xla(ii,kk)*a(jj,kk-jj+1)*(corri*dt**2)**2
      go to 311
  316 rz1(ii,jj)=rz1(ii,jj)+xla(ii,kk)*a(kk,ms)*(corri*dt**2)**2
  311 continue
  310 continue
      if (wstif.eq.0) go to 333
      write (6,13)
   13 format (' K XDP XDP(T) K(T) :')
      do 303 ii=1,numeq
  303 write (6,2) (rz1(ii,jj),jj=1,numeq)
c           Evaluate E[z(i)z(i-1)]
  333 do 470 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 470 jj=1,numeq
      xla(ii,jj)=0.
```

```
      if (a4(ii,ii).eq.1.0.or.a4(jj,jj).eq.1.) go to 470
      do 471 kk=mg,mbd
      xla(ii,jj)=xla(ii,jj)+a4(ii,kk)*zz(kk,jj,1)
  471 continue
  470 continue
      do 570 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 570 jj=1,numeq
      z12(ii,jj,2)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.) go to 570
      do 571 kk=mg,mbd
      z12(ii,jj,2)=a3(ii,kk)*z12(jj,kk,1)+z12(ii,jj,2)
  571 continue
  570 continue
      do 480 ii=1,numeq
      do 480 jj=1,numeq
  480 xla(ii,jj)=z12(ii,jj,2)+xla(ii,jj)-z2(ii,jj,1)*dt**2
      do 482 ii=1,numeq
      do 482 jj=1,numeq
      z12(ii,jj,2)=0.
      if (al(ii,ii).eq.1.0.or.al(jj,jj).eq.1.) go to 482
      do 483 kk=1,numeq
      z12(ii,jj,2)=z12(ii,jj,2)+al(ii,kk)*xla(kk,jj)
  483 continue
  482 continue
      if (wstif.eq.0) go to 150
      write (6,1)
    1 format ('z12 (i,j) matrix')
      do 22 ii=1,numeq
   22 write (6,2) (z12(ii,jj,2),jj=1,numeq)
    2 format (6e13.6)
c         Evaluate rz12(i,j) = A2 z12(i,j,2) A3(T)
  150 do 485 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 485 jj=1,numeq
      xla(ii,jj)=0.
      if (a2(ii,ii).eq.1.0.or.a2(jj,jj).eq.1.) go to 485
      do 486 kk=mg,mbd
      xla(ii,jj)=xla(ii,jj)+a2(ii,kk)*z12(kk,jj,2)
  486 continue
  485 continue
      do 490 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 490 jj=1,numeq
      rz12(jj,ii)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.) go to 490
      do 491 kk=mg,mbd
      rz12(jj,ii)=rz12(jj,ii)+a3(ii,kk)*xla(jj,kk)
  491 continue
  490 continue
      if (wstif.eq.0) go to 152
      write (6,5)
    5 format ('rz12 (i,j) matrix')
      do 6 ii=1,numeq
```

```
    6 write (6,2) (rz12(ii,jj),jj=1,numeq)
c       Evaluate K xdp(i,2) Z(j)T A2(T) and K xdp(i,2) Z(j-1)T A3(T)
  152 do 495 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 495 jj=1,numeq
      xla(jj,ii)=0.
      if (a2(ii,ii).eq.1.0.or.a2(jj,jj).eq.1.) go to 495
      do 496 kk=mg,mbd
      xla(jj,ii)=xla(jj,ii)+a2(ii,kk)*z2(jj,kk,2)*dt**2
  496 continue
  495 continue
      do 595 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 595 jj=1,numeq
      z12(jj,ii,1)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.) go to 595
      do 596 kk=mg,mbd
      z12(jj,ii,1)=z12(jj,ii,1)+a3(ii,kk)*rz(jj,kk)*dt**2
  596 continue
  595 continue
      if (wstif.eq.0) return
      write (6,9)
    9 format (' K xdp Z(j) A2(t)')
      do 10 ii=1,numeq
   10 write (6,2) (xla(ii,jj),jj=1,numeq)
      write (6,11)
   11 format (' K xdp Z(j-1) A3(t)')
      do 12 ii=1,numeq
   12 write (6,2) (z12(ii,jj,1),jj=1,numeq)
      return
      end
```

```
      subroutine d5 (iv)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/xmd/ xmass(50),damp(50,25)
      common/matrix/ La(50),t1(50),t2(50),f(50)
      common/rest/ res(50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      double precision xdp,x1d,x2d,xmass,damp,t1,t2,f,df
      double precision a1,a2,a3,xLa,res
c        Find the forcing fn at i'th step
      do 150 ii=1,numeq
c        Pick out # DOF that has load
      do 160 jj=1,ndofl
      if (ii.eq.ndof(jj)) go to 161
  160 continue
      f(ii)=0.
      go to 150
c        Pick out n'th point that has amplitude change
  161 Lc=la(ii)
      if (iv.eq.ns(ii,Lc)) go to 170
      df=(dl(ii,Lc)-dl(ii,lc-1))/(ns(ii,lc)-ns(ii,lc-1))
      f(ii)=dl(ii,lc-1)+df*(iv-ns(ii,lc-1))
      go to 150
  170 f(ii)=dl(ii,lc)
      la(ii)=la(ii)+1
  150 continue
c        form effective load vector
      do 200 ii=1,numeq
      t1(ii)=0.
      call range (ii,numeq,mband,mg,mbd)
      do 200 jj=mg,mbd
      if (xdp(jj,1).eq.0..or.a3(ii,jj).eq.0.) go to 200
      t1(ii)=t1(ii)+a3(ii,jj)*xdp(jj,1)
  200 continue
      do 205 ii=1,numeq
  205 t2(ii)=2.*xmass(ii)*xdp(ii,2)
      do 210 ii=1,numeq
  210 xdp(ii,3)=t2(ii)+t1(ii)+(f(ii)-res(ii))*dt**2
      call  choles(iv)
c         calculate velocity and acceleration
      do 220 ii=1,numeq
      x2d(ii,1)=(xdp(ii,3)-2.*xdp(ii,2)+xdp(ii,1))/dt**2
      x1d(ii,1)=(xdp(ii,3)-xdp(ii,1))/(2.*dt)
  220 continue
      return
      end
```

```
      subroutine d6 (iv,irandm)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1           nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/xmd/ xmass(50),damp(50,25)
      common/rest/ res(50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      dimension res1(50,2)
      double precision xdp,x1d,x2d,xmass,damp,aaa,eigvec,rho,a4
      double precision a1,a2,a3,xLa,dxdp,zz,z12,z2,res,res1
c        evaluate the restoring force
      if (irandm.eq.2) go to 320
      do 290 ii=1,numeq
  290 res(ii)=0.
      do 310 id=1,numem
      call restor (id,iv,1)
  310 continue
      call rbound
      return
c        Move A2 to A4; where A4 = A2(j-1)
  320 do 150 ii=1,numeq
      do 150 jj=1,numeq
  150 a4(ii,jj)=a2(ii,jj)
c        evaluate the equivalent stifness matrix
      do 330 it=1,numnp
      do 330 is=1,nq
      if (ir(it,is).eq.1) go to 330
      ik=it*nq-(nq-is)
      aaa=xdp(ik,3)
  329 dxdp=aaa/333.
      if (dxdp.eq.0.) go to 380
      do 340 nn=1,2
      do 361 im=1,numeq
  361 res(im)=0.
      if (nn.eq.2) go to 341
      xdp(ik,3)=aaa+dxdp
      go to 350
  341 xdp(ik,3)=aaa-dxdp
  350 do 360 id=1,numem
      call restor (id,iv,nn)
  360 continue
      do 370 ii=1,numeq
  370 res1(ii,nn)=res(ii)
  340 continue
      if (wstif.eq.0) go to 380
      write (6,7) ik
    7 format ('dof = ',i4,'  res1(ii,1), and res1(ii,2) are:')
      do 465  j5=1,2
  465 write (6,1) (res1(i4,j5),i4=1,numeq)
```

```
  380 do 390 ii=1,numeq
      if (dxdp.eq.0.) go to 395
      a2(ii,ik)=(res1(ii,1)-res1(ii,2))/(2.*dxdp)
      go to 390
  395 a2(ii,ik)=a2(ii,ik)
  390 continue
      do 385 ii=1,numeq
  385 x2d(ii,3)=(res1(ii,1)+res1(ii,2))/2.
      xdp(ik,3)=aaa
  330 continue
      do 430 ii=1,numeq
  430 res(ii)=x2d(ii,3)
c        write down the stiffness matrix
      if (wstif.eq.0) go to 405
      do 415 ii=1,numeq
  415 write (6,1) (a2(ii,jj),jj=1,numeq)
  405 call sbound(1)
      call rbound
      if (wrstif.eq.0) go to 410
      write (6,2)
      do 420 ii=1,numeq
  420 write (6,1) (a2(ii,jj),jj=1,numeq)
c        compute a2 matrix
  410 do 400 ii=1,numeq
      do 400 jj=1,numeq
      a2(ii,jj)=-a2(ii,jj)*dt**2
      if (ii.ne.jj) go to 400
      a2(ii,jj)=a2(ii,jj)+2.*xmass(ii)
  400 continue
    1 format (6e13.6)
    2 format (/6x,'THE EQUIVALENT STIFFNESS MATRIX IS :')
      return
      end
```

```
      subroutine d7
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      double precision zz,z12,z2,rf,rz,rz1,rz12,a1,a2,a3,xLa
c        This subroutine evaluate   E [Z(j+1) Z(j-1) ]
      do 200 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 200 jj=1,numeq
      xla(ii,jj)=0.
      if (a2(ii,ii).eq.1.0.or.a2(jj,jj).eq.1.0) go to 200
      do 210 kk=mg,mbd
      xla(ii,jj)=xla(ii,jj)+a2(ii,kk)*z12(kk,jj,2)
  210 continue
  200 continue
      do 220 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 220 jj=1,numeq
      rz(ii,jj)=0.
      if (a3(ii,ii).eq.1.0.or.a3(jj,jj).eq.1.0) go to 220
      do 230 kk=mg,mbd
      rz(ii,jj)=rz(ii,jj)+a3(ii,kk)*zz(kk,jj,1)
  230 continue
  220 continue
      do 240 ii=1,numeq
      do 240 jj=1,numeq
  240 rz(ii,jj)=rz(ii,jj)+xla(ii,jj)-z2(ii,jj,1)*dt**2
      do 250 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 250 jj=1,numeq
      rf(ii,jj)=0.
      if (a1(ii,ii).eq.1.0.or.a1(jj,jj).eq.1.0) go to 250
      do 260 kk=mg,mbd
      rf(ii,jj)=rf(ii,jj)+a1(ii,kk)*rz(kk,jj)
  260 continue
  250 continue
      return
      end
```

```
      subroutine damping
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/dya/ mass,inlod,alpha,beta,mdi
      double precision a,b,xmass,damp
      do 100 jj=1,mband
      do 100 ii=1,numeq
      if (jj.eq.1) go to 101
      damp(ii,jj)=beta*a(ii,jj)
      go to 100
  101 damp(ii,jj)=alpha*xmass(ii)+beta*a(ii,jj)
  100 continue
      return
      end
```

```fortran
      subroutine dicd
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/control/ ivelo,iacc
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),ax(50,50,10),
     1               nt(50,50,10),loh,nroot,nsmax,corri
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/chol/xmc(50,25)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      common/kuz/ rx(50,50,3,8),ck(8)
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      common/cross/ vv(50,50),aa(50,50),vz(50,50),az(50,50),av(50,50)
      common/icross/ icros,ivv,iaa,ivz,iaz,iav
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      double precision a,b,xmass,damp,xdp,x1d,x2d,t1,t2,f,xmc
      double precision a1,a2,a3,xla,rz,rz1,rz12,z12,zz,rf
      double precision eigvec,rho,a4,z2,ck,rx,vv,aa,vz,az,av
c          This subroutine proceed the direct integration by using
c           Central Difference method
      open(unit=10,file='plotout.mean.cd',status='new')
      open(unit=11,file='plotout.var.cd',status='new')
      read (5,1) inpt,ifpt,jpnp,intpt,iftpt,jpnt,init,iplot,meanw,iwvar
      write (6,6) inpt,ifpt,jpnp,intpt,iftpt,jpnt
      write (6,7) init,iplot
      write (6,8) meanw,iwvar
      if (init.eq.0) go to 100
      do 110 i=1,numeq
      read (5,2) xdp(i,2),x1d(i,2),x2d(i,2)
  110 continue
      go to 111
  100 do 120 ii=1,numeq
      xdp(ii,2)=0.
      x1d(ii,2)=0.
      x2d(ii,2)=0.
  120 continue
c        Evaluate the Initial displacement
  111 do 140 ii=1,numeq
      xdp(ii,1)=xdp(ii,2)-dt*x1d(ii,2)+.5*x2d(ii,2)*dt**2
  140 continue
c        Evaluate [C]*.5/dt+[M]/dt**2
      do 300 ii=1,numeq
      do 300 jj=1,mband
      if (jj.eq.1) go to 301
      xmc(ii,jj)=damp(ii,jj)*.5/dt
      go to 300
  301 xmc(ii,jj)=damp(ii,jj)*.5/dt+xmass(ii)/dt**2
  300 continue
```

```
      write (6,12)
c        find the highest eigenvalue
      do 148 ii=1,numeq
      do 148 jj=1,mband
  148 rz1(ii,jj)=a(ii,jj)
      call higheig (nmc,numeq,mband)
      write (6,14) rho(8),nmc
      write (6,11) (eigvec(ii,8),ii=1,numeq)
      nwr=0
      go to (145,146) irandm
c        Modal analysis
  146 call bound (1,1.0E+15)
      if (wstif.eq.0) go to 200
      do 210 ii=1,numeq
  210 write (6,11) (a(ii,jj),jj=1,mband)
  200 if (nroot.gt.1) go to 340
      call loweig (nmc,numeq,mband)
      write (6,15) rho(1),nmc
      write (6,11) (eigvec(ii,1),ii=1,numeq)
      go to 147
  340 call sspace (nroot,nsmax)
      write (6,16)
      write (6,11) (rho(ii),ii=1,nroot)
      write (6,17)
      do 342 ii=1,nroot
      write (6,11) (eigvec(jj,ii),jj=1,numeq)
  342 continue
  147 do 360 ii=1,numeq
      do 360 jj=1,mband
  360 a(ii,jj)=rz1(ii,jj)
c          evaluate A1,A2,A3
      do 402 ii=1,numeq
      do 402 jj=1,mband
      a1(ii,jj+ii-1)=damp(ii,jj)*dt*.5
      a2(ii,jj+ii-1)=-a(ii,jj)*dt**2
      a3(ii,jj+ii-1)=a1(ii,jj+ii-1)
      a4(ii,jj+ii-1)=a2(ii,jj+ii-1)
      if (jj.ne.1) go to 402
  403 a1(ii,jj+ii-1)=a1(ii,jj+ii-1)+xmass(ii)
      a2(ii,jj+ii-1)=a2(ii,jj+ii-1)+2.*xmass(ii)
      a3(ii,jj+ii-1)=a3(ii,jj+ii-1)-xmass(ii)
      a4(ii,jj+ii-1)=a2(ii,jj+ii-1)
  402 continue
c          symmetric
      do 404 ii=1,numeq
      do 404 jj=ii+1,numeq
      a1(jj,ii)=a1(ii,jj)
      a2(jj,ii)=a2(ii,jj)
      a3(jj,ii)=a3(ii,jj)
      a4(jj,ii)=a4(ii,jj)
  404 continue
c        A inverse
      call matinv (numeq,wstif)
      call sbound (4)
```

```
  145 do 130 ii=1,numeq
      la(ii)=1
  130 continue
      do 131 ii=1,numeq
      do 131 jj=1,numeq
  131 la2(ii,jj)=1
c         Start the iteration
      do 1000 i=1,nti
      call d1 (i)
      go to (400,408), irandm
  408 call d2 (i)
c        Evaluate ck(np)
      do 440 np=1,nroot
      ck(np)=0.
      do 440 ii=1,numeq
      if (xdp(ii,2).eq.0.) go to 440
      ck(np)=ck(np)+eigvec(ii,np)*xmass(ii)*xdp(ii,2)
  440 continue
      call d3
      call d4
c        TOTAL SUM
      do 500 ii=1,numeq
      do 500 jj=ii,numeq
      rf(ii,jj)=rf(ii,jj)+rz12(ii,jj)+rz12(jj,ii)
      rf(ii,jj)=rf(ii,jj)-xla(ii,jj)-xla(jj,ii)
      rf(ii,jj)=rf(ii,jj)-z12(ii,jj,1)-z12(jj,ii,1)+rz1(ii,jj)
  500 continue
      do 505 jj=1,numeq
      do 505 ii=jj+1,numeq
      rf(ii,jj)=rf(jj,ii)
  505 continue
      if (wstif.eq.0) go to 515
      write (6,9)
    9 format (' the total sum matrix')
      do 525 ii=1,numeq
  525 write (6,11) (rf(ii,jj),jj=1,numeq)
c       Evaluate zz(ii,jj,3)
  515 do 540 ii=1,numeq
      do 540 jj=1,numeq
      xla(ii,jj)=0.
      if (al(ii,ii).eq.1.0.or.al(jj,jj).eq.1.) go to 540
      do 541 kk=1,numeq
      xla(ii,jj)=xla(ii,jj)+al(ii,kk)*rf(kk,jj)
  541 continue
  540 continue
      do 550 ii=1,numeq
      do 550 jj=ii,numeq
      zz(ii,jj,3)=0
      if (al(jj,jj).eq.1.0.or.al(ii,ii).eq.1.) go to 550
      do 551 kk=1,numeq
      zz(ii,jj,3)=zz(ii,jj,3)+xla(ii,kk)*al(jj,kk)
  551 continue
  550 continue
      do 555 jj=1,numeq
```

```
      do 555 ii=jj,numeq
      if (ii.eq.jj) go to 555
      zz(ii,jj,3)=zz(jj,ii,3)
  555 continue
      if (icros.eq.0) go to 400
      if (ivz.eq.1) call svz (i,nwr) .
      if (iaz.eq.1) call saz (i,nwr)
      call d7
      if (iav.eq.1) call sav (i,nwr)
      if (iaa.eq.1) call saa (i,nwr)
      if (ivv.eq.1) call svv (i,nwr)
c         write down the results
  400 call write (i,nwr)
 1000 continue
    1 format (10i5)
    2 format (3f10.5)
    6 format (9(/),5x,' INITIAL # DOF FOR  PRINT OUT [INPT] =',i5,
     1  ///5x,' FINIAL # DOF FOR PRINT OUT [IFPT] = ',i5,
     2  ///5x,' INCREMENT OF DOF FOR PRINT OUT [JPNP] =',i5,
     3  ///5x,' INITIAL TIME NUMBER FOR  PRINT OUT [INTPT] =',i5,
     4  ///5x,' FINIAL TIME NUMBER FOR PRINT OUT [IFTPT] =',i5,
     5  ///5x,' INCREMENT OF TIME FOR PRINT OUT [JPNT] = ',i5)
    7 format (9(/),5x,'INITIAL CONDITION [INIT] = ',i5,
     1        /5x,': 0 GENERATE TO BE EQUAL ZERO'
     2        /5x,': 1 READ FROM DATA CARD'
     3      ///5x,'[ IPLOT ] = ',i5,
     4        /5x,'  0 DIRECTLY WRITE DOWN FOR PRINT OUT'
     5        /5x,'  1 WRITE DOWN FOR PLOTTING MANNER')
    8 format (//5x,'FLAG FOR WRITING DOWN MEAN RESPONSE [ MEANW ] =',
     1        i5,/5x,': 0  NO',/5x,': 1  YES',
     2      //5x,'FLAG FOR WRITING DOWN VARIANCE RESPONSE [ IWVAR ] =',
     3        i5,/5x,': 0  NO',/5X,': 1  YES')
   11 format (6e13.5)
   12 format (10(/),10x,' THIS IS SOLUTION OF C.D. METHOD')
   14 format (///3x,'THE HIGHEST FREQUENCY IS :  ',e20.9,
     1  3x'NUMBER OF ITERATION OF FINDING HIGHEST FREQUENCY IS :',i5,
     2  15x,'THE CORRESPONDING EIGENVECTOR IS :')
   15 format (///3x,'THE LOWEST FREQUENCY IS :  ',e20.9,
     1  3x'NUMBER OF ITERATION OF FINDING LOWEST FREQUENCY IS :',i5,
     2  15x,'THE CORRESPONDING EIGENVECTOR IS :')
   16 format (///5x,'THE EIGENVALUES ARE :')
   17 format (///5x,'THE EIGENVECTOR ARE  :')
      return
      end
```

```
      subroutine dicdn
      integer wstif,wrstif,wlod
      common/engy/ strain(11,11,10,2),energd(10),dv(2)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),ax(50,50,10),
     1               nt(50,50,10),lon,nroot,nsmax,corri
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/chol/xmc(50,25)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      common/kuz/ rx(50,50,3,8),ck(8)
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      common/flagw/inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      common/icross/ icros,ivv,iaa,ivz,iaz,iav
      double precision a,b,xmass,damp,xdp,x1d,x2d,t1,t2,f,xmc
      double precision a1,a2,a3,xla,rz,rz1,rz12,z12,zz,rf
      double precision eigvec,rho,a4,z2,ck,rx
c          This subroutine proceed the direct integration by using
c           Central Difference method
      open(unit=10,file='plotout.mean.cd.n',status='new')
      open(unit=11,file='plotout.var.cd.n',status='new')
      read (5,1) inpt,ifpt,jpnp,intpt,iftpt,jpnt,init,iplot,meanw,iwvar
      write (6,6) inpt,ifpt,jpnp,intpt,iftpt,jpnt
      write (6,7) init,iplot
      write (6,8) meanw,iwvar
      if (init.eq.0) go to 100
      do 110 i=1,numeq
      read (5,2) xdp(i,2),x1d(i,2),x2d(i,2)
  110 continue
      go to 111
  100 do 120 ii=1,numeq
      xdp(ii,2)=0.
      x1d(ii,2)=0.
      x2d(ii,2)=0.
  120 continue
c        Evaluate the Initial displacement
  111 do 140 ii=1,numeq
      xdp(ii,1)=xdp(ii,2)-dt*x1d(ii,2)+.5*x2d(ii,2)*dt**2
  140 continue
c        Evaluate [C]*.5/dt+[M]/dt**2
      do 300 ii=1,numeq
      do 300 jj=1,mband
      xmc(ii,jj)=damp(ii,jj)*dt/2.
      if (jj.ne.1) go to 300
      xmc(ii,jj)=damp(ii,jj)*dt/2.+xmass(ii)
  300 continue
      write (6,12)
c        Find the highest eigenvalue
```

```
      do 148 ii=1,numeq
      do 148 jj=1,mband
  148 rz1(ii,jj)=a(ii,jj)
      call higheig (nmc,numeq,mband)
      write (6,14) rho(8),nmc
      write (6,11) (eigvec(ii,8),ii=1,numeq)
      nwr=0
      go to (145,146) irandm
c        Modal analysis
  146 call bound (1,1.0E+15)
      if (wstif.eq.0) go to 210
      do 220 ii=1,numeq
  220 write (6,11) (a(ii,jj),jj=1,mband)
  210 if (nroot.gt.1) go to 340
      call loweig (nmc,numeq,mband)
      write (6,15) rho(1),nmc
      write (6,11) (eigvec(ii,1),ii=1,numeq)
      go to 147
  340 call sspace (nroot,nsmax)
      write (6,16)
      write (6,11) (rho(ii),ii=1,nroot)
      write (6,17)
      do 342 ii=1,nroot
      write (6,11) (eigvec(jj,ii),jj=1,numeq)
  342 continue
  147 do 360 ii=1,numeq
      do 360 jj=1,mband
  360 a(ii,jj)=rz1(ii,jj)
c           evaluate A1,A2,A3; The reason we evaluate the
c        A2 matrix is because that at time step (1) the stiffness
c        matrix is equal to the primary stiffness we establish
  145 do 402 ii=1,numeq
      do 402 jj=1,mband
      a1(ii,jj+ii-1)=damp(ii,jj)*dt*.5
      a2(ii,jj+ii-1)=-a(ii,jj)*dt**2
      a3(ii,jj+ii-1)=a1(ii,jj+ii-1)
      a4(ii,jj+ii-1)=a2(ii,jj+ii-1)
      if (jj.ne.1) go to 402
  403 a1(ii,jj+ii-1)=a1(ii,jj+ii-1)+xmass(ii)
      a2(ii,jj+ii-1)=a2(ii,jj+ii-1)+2.*xmass(ii)
      a3(ii,jj+ii-1)=a3(ii,jj+ii-1)-xmass(ii)
      a4(ii,jj+ii-1)=a2(ii,jj+ii-1)
  402 continue
c           symmetric
      do 404 ii=1,numeq
      do 404 jj=ii+1,numeq
      a1(jj,ii)=a1(ii,jj)
      a2(jj,ii)=a2(ii,jj)
      a3(jj,ii)=a3(ii,jj)
      a4(jj,ii)=a4(ii,jj)
  404 continue
      if (irandm.eq.1) go to 149
c        A inverse
      call matinv (numeq,wstif)
```

```
      call sbound (4)
  149 do 130 ii=1,numeq
      La(ii)=1
  130 continue
      do 131 ii=1,numeq
      do 131 jj=1,numeq
  131 La2(ii,jj)=1
c        Start the iteration
      do 1000 i=1,nti
      call d5 (i)
      go to (400,408), irandm
  408 call d2 (i)
c        Evaluate ck(np)
      do 440 np=1,nroot
      ck(np)=0.
      do 440 ii=1,numeq
      if (xdp(ii,2).eq.0.) go to 440
      ck(np)=ck(np)+eigvec(ii,np)*xmass(ii)*xdp(ii,2)
  440 continue
      call d3
      call d4
c        TOTAL SUM
      do 500 ii=1,numeq
      do 500 jj=ii,numeq
      rf(ii,jj)=rf(ii,jj)+rz12(ii,jj)+rz12(jj,ii)
      rf(ii,jj)=rf(ii,jj)-xla(ii,jj)-xla(jj,ii)
      rf(ii,jj)=rf(ii,jj)-z12(ii,jj,1)-z12(jj,ii,1)+rz1(ii,jj)
  500 continue
      do 505 jj=1,numeq
      do 505 ii=jj+1,numeq
      rf(ii,jj)=rf(jj,ii)
  505 continue
      if (wstif.eq.0) go to 515
      write (6,9)
    9 format (' the total sum matrix')
      do 525 ii=1,numeq
  525 write (6,11) (rf(ii,jj),jj=1,numeq)
c     Evaluate zz(ii,jj,3)
  515 do 540 ii=1,numeq
      do 540 jj=1,numeq
      xla(ii,jj)=0.
      if (al(ii,ii).eq.1.0.or.al(jj,jj).eq.1.) go to 540
      do 541 kk=1,numeq
      xla(ii,jj)=xla(ii,jj)+al(ii,kk)*rf(kk,jj)
  541 continue
  540 continue
      do 550 ii=1,numeq
      do 550 jj=ii,numeq
      zz(ii,jj,3)=0.
      if (al(jj,jj).eq.1.0.or.al(ii,ii).eq.1.) go to 550
      do 551 kk=1,numeq
      zz(ii,jj,3)=zz(ii,jj,3)+xla(ii,kk)*al(jj,kk)
  551 continue
  550 continue
```

```
      do 555 jj=1,numeq
      do 555 ii=jj,numeq
      if (ii.eq.jj) go to 555
      zz(ii,jj,3)=zz(jj,ii,3)
  555 continue
      if (icros.eq.0) go to 400
      if (ivz.eq.1) call svz (i,nwr)
      if (iaz.eq.1) call saz (i,nwr)
      call d7
      if (iav.eq.1) call sav (i,nwr)
      if (ivv.eq.1) call svv (i,nwr)
      if (iaa.eq.1) call saa (i,nwr)
  400 call d6(i,irandm)
c        write down the results
      call write (i,nwr)
 1000 continue
    1 format (10i5)
    2 format (3f10.5)
    6 format (9(/),5x,' INITIAL # DOF FOR  PRINT OUT [INPT] =',i5,
     1  ///5x,' FINIAL # DOF FOR PRINT OUT [IFPT] = ',i5,
     2  ///5x,' INCREMENT OF DOF FOR PRINT OUT [JPNP] =',i5,
     3  ///5x,' INITIAL TIME NUMBER FOR  PRINT OUT [INTPT] =',i5,
     4  ///5x,' FINIAL TIME NUMBER FOR PRINT OUT [IFTPT] =',i5,
     5  ///5x,' INCREMENT OF TIME FOR PRINT OUT [JPNT] = ',i5)
    7 format (9(/),5x,'INITIAL CONDITION [INIT] = ',i5,
     1       /5x,': 0 GENERATE TO BE EQUAL ZERO'
     2       /5x,': 1 READ FROM DATA CARD'
     3     ///5x,'[ IPLOT ] = ',i5,
     4       /5x,': 0 DIRECTLY WRITE DOWN FOR PRINT OUT'
     5       /5x,': 1 WRITE DOWN FOR PLOTTING MANNER')
    8 format (///5x,'FLAG FOR WRITING DOWN MEAN RESPONSE [ MEANW ] =',
     1       i5,/5x,': 0  NO',/5x,': 1  YES',
     2   ///5x,'FLAG FOR WRITING DOWN VARIANCE RESPONSE [ IWVAR ] =',
     3       i5,/5x,': 0  NO',/5X,': 1  YES')
   11 format (6e13.5)
   12 format (10(/),10x,' THIS IS SOLUTION OF C.D. METHOD')
   14 format (///3x,'THE HIGHEST FREQUENCY IS : ',e20.9,
     1//3x'NUMBER OF ITERATION OF FINDING HIGHEST FREQUENCY IS :',i5,
     2//15x,'THE CORRESPONDING EIGENVECTOR IS :')
   15 format (///3x,'THE LOWEST FREQUENCY IS : ',e20.9,
     1//3x'NUMBER OF ITERATION OF FINDING LOWEST FREQUENCY IS :',i5,
     2//15x,'THE CORRESPONDING EIGENVECTOR IS :')
   16 format (//5x,'THE EIGENVALUES ARE :')
   17 format (//5x,'THE EIGENVECTOR ARE :'/)
      return
      end
```

```
      subroutine dinm
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      double precision a,b,xmass,damp,xdp,x1d,x2d,t1,t2,f
      double precision ax1,ax2,err1,error
c   This subroutine proceeed direct integration using New-Mark method
c          inipt: initial #do of print outy
c          ifpt: final #dof of print out
c          jpnp: jump increment for #dof of print out
c          intpt: intial time for number for print
c          jpnt: jump increment for time print out
c          fiftpt: final tmeime number for print
c          init: initial condition 0: generate zero
c                                  1: read from data card
      read (5,1) inpt,ifpt,jpnp,intpt,iftpt,jpnt,init,iplot
      write (6,6) inpt,ifpt,jpnp,intpt,iftpt,jpnt
      write (6,7) init,iplot
      if (init.eq.0) go to 100
      do 110 i=1,numeq
      read (5,2) xdp(i,1),x1d(i,1),x2d(i,1)
 110  continue
      go to 111
 100  do 120 ii=1,numeq
      xdp(ii,1)=0.
      x1d(ii,1)=0.
      x2d(ii,1)=0.
 120  continue
      open (unit=10,file='poutout.fda.nm',status='new')
 111  do 130 ii=1,numeq
      la(ii)=1
 130  continue
      write (6,9)
   9  format (10(/),36x,'THIS IS THE SOLUTION FOR N.M. METHOD')
      do 1000 i=2,nti
c        Find the forcing fn at i'th step
      do 150 ii=1,numeq
c       Pick out # DOF that has load
      do 160 jj=1,ndofl
      if (ii.eq.ndof(jj)) go to 161
 160  continue
      f(ii)=0.
      go to 150
c        Pick out n'th point that has amplitude change
 161  lc=la(ii)
      if (i.eq.ns(ii,lc)) go to 170
      df=(dl(ii,lc)-dl(ii,lc-1))/(ns(ii,lc)-ns(ii,lc-1))
      f(ii)=dl(ii,lc-1)+df*(i-ns(ii,lc-1))
      go to 150
 170  f(ii)=dl(ii,lc)
```

```
      la(ii)=la(ii)+1
 150  continue
c           M inverse *F
      do 190 ii=1,numeq
      f(ii)=f(ii)/xmass(ii)
 190  continue
c           start the iteration
      if (i.eq.2) go to 132
      do 131 ii=1,numeq
      x2d(ii,2)=x2d(ii,1)
 131  continue
      go to 999
 132  do 133 ii=1,numeq
      x2d(ii,2)=1.
 133  continue
 999  do 140 ii=1,numeq
      xt=x2d(ii,1)+x2d(ii,2)
      xdp(ii,2)=xdp(ii,1)+dt*x1d(ii,1)+.25*xt*dt**2
      x1d(ii,2)=x1d(ii,1)+.5*xt*dt
 140  continue
c             K*xdp
      do 209 ii=1,numeq
      t2(ii)=0.
 209  t1(ii)=0.
      do 200 ii=1,numeq
      mbd=mband+ii-1
      if (mbd.gt.numeq) mbd=numeq
      mg=ii-mband+1
      if (mg.le.0) mg=1
      do 200 ij= mg,mbd
      ms=ij-ii+1
      if (ms) 201,201,202
 201  t1(ii)=t1(ii)+a(ij,ii-ij+1)*xdp(ij,2)
      t2(ii)=t2(ii)+damp(ij,ii-jj+1)*x1d(ij,2)
      go to 200
 202  t1(ii)=t1(ii)+a(ii,ms)*xdp(ij,2)
      t2(ii)=t2(ii)+damp(ii,ms)*x1d(ij,2)
 200  continue
      do 300 ii=1,numeq
      t1(ii)=t1(ii)/xmass(ii)
      t2(ii)=t2(ii)/xmass(ii)
 300  continue
      do 320 ii=1,numeq
      x2d(ii,3)=f(ii)-t1(ii)-t2(ii)
 320  continue
      do 330 ii=1,numeq
      ax1=dabs(x2d(ii,2))
      ax2=dabs(x2d(ii,3))
      if (ax1.eq.ax2) go to 330
      if (ax1.eq.0.0.and.ax2.ne.0.0) go to 340
      if (ax2.eq.0.0.and.ax1.ne.0.0) go to 340
      err1=ax1/ax2
      error=dabs(err1-1.)
      if (error.gt.0.005) go to 340
```

```
330  continue
     go to 998
340  do 350 ii=1,numeq
     x2d(ii,2)=x2d(ii,3)
350  continue
     go to 999
998  if (i.gt.iftpt) go to 995
     if (i.eq.intpt) go to 997
     nwr=nwr+1
     if (nwr.eq.jpnt) go to 997
     go to 995
997  if (iplot.ne.0) go to 955
     write (6,3) i
     write (6,4)
     do 996 ii=inpt,ifpt,jpnp
     write (6,5) ii, xdp(ii,2),x1d(ii,2),x2d(ii,2),f(ii)
996  continue
     go to 954
955  do 956 ii=inpt,ifpt,jpnp
     write (10,8) i,xdp(ii,2),x1d(ii,2),x2d(ii,2),f(ii)
956  continue
954  nwr=0
995  do 994 ii=1,numeq
     xdp(ii,1)=xdp(ii,2)
     x1d(ii,1)=x1d(ii,2)
     x2d(ii,1)=x2d(ii,2)
994  continue
1000 continue
   1 format (8i5)
   2 format (3f10.5)
   3 format (///20x,'  NUMBER OF TIME STEP',3x,i5)
   4 format (//'# DOF ',2x,'  DISPLACEMENT ',6x,'  VELOCITY',7x,
    1  ' ACCELERATION',7x,' FORCING FUNCTION')
   5 format (i5,4x,e15.8,4x,e15.8,4x,e15.8,4x,e15.8)
   6 format (9(/),20x,'INITIAL # DOF FOR  PRINT OUT [INPT]',
    1  5x,i5/20x,' FINIAL # DOF FOR PRINT OUT [IFPT]',5x,i5,
    2  /20x,' INCREMENT OF DOF FOR PRINT OUT [JPNP]',5x,i5,
    3  /20x,' INITIAL TIME NUMBER FOR  PRINT OUT [INTPT]',5X,i5,
    4  /20x,' FINIAL TIME NUMBER FOR PRINT OUT [IFTPT]',5x,i5,
    5  /20x,' INCREMENT OF TIME FOR PRINT OUT [JPNT]',5x,i5)
   7 format (9(/),20x,' INITIAL CONDITION ',15x,i5,
    1       /20x,' = 0 GENERATE TO BE EQUAL ZERO'
    2       /20x,' = 1 READ FROM DATA CARD'
    3       /20x,' IPLOT =  ',i5
    4       /20x,'       = 0  DIRECTLY WRITE DOWN FOR PRINT OUT'
    5       /20x,'       = 1  WRITE DOWN FOR PLOTTING MANNER')
  8  format(i4,2x,e15.8,2x,e15.8,2x,e15.8,2x,e15.8)
     return
     end
```

```fortran
      subroutine dld (ild,w,yp,xzp,sp)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      double precision s,r,t,st,tf
      go to (5,10,15,20), ild
c          fixed - fixed ends
    5 p(1)=0.5*w*yp
      p(2)=0.5*w*xzp
      p(6)=w*sp*xzp/12.
      p(7)=p(1)
      p(8)=p(2)
      p(12)=-p(6)
      p(12)=-p(6)
      go to 25
c          hinge - hinge ends
   10 p(1)=0.5*w*yp
      p(2)=0.5*w*xzp
      p(7)=p(1)
      p(8)=p(2)
      go to 25
c          hinge left end
   15 p(1)=0.5*w*yp
      p(2)=0.375*w*xzp
      p(7)=p(1)
      p(8)=0.625*w*xzp
      p(12)=-0.125*w*sp*xzp
      go to 25
c          hinge right end
   20 p(1)=0.5*w*yp
      p(2)=0.625*w*xzp
      p(6)=0.125*w*sp*xzp
      p(7)=p(1)
      p(8)=0.375*w*xzp
   25 continue
      return
      end
```

```fortran
      subroutine dload
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/random/ irandm,ncvdf,ntacv,nl(10),n2(10),ax(50,50,10),
     1               nt(50,50,10),lon,nroot,nsmax,corri
      common/icross/ icros,ivv,iaa,ivz,iaz,iav
c     this subroutine read dynamic load
c       irandm: indicate random analysis
c               1: yes; 2: No
c       ndofl : total number dof that have load acting
c       dt    : time increment
c       nti   : number of time increment
c    ndof(20) : #th  dof that has load acting
c    dl(ndo,j): dynamic load amplitude
c     ns(200,j) : ns 'th point that has changing amplitude
c ntpdl :  amount number of point of ns(i,j) [for one time function]
c    ncvdf: Total # of pairs in covariance matrix for forcing fn
c    LOn  : linear of nonlinear analysis
c           1 : Linear analysis; 2: Nonlinear analysis
c    nl(i) & n2(i) : pair # in covariance forcing fn for input
c    ntacv: total # of amplitude change in time hystery for N1,N2 pair
c    nt(nl(i),n2((i),j), and ax(nl(i),n2(i),j) are the pair for time
c    and corresponding amplitude change point
c
      read (5,1) irandm,lon,ndofl,nti,dt
      write (6,10) irandm,lon,ndofl,nti,dt
      do 100 i=1,ndofl
      read (5,2) ndof(i),ntpdl
      ndo=ndof(i)
      read (5,3) (ns(ndo,j),dl(ndo,j), j=1,ntpdl)
      write (6,11) ndo,(ns(ndo,j),dl(ndo,j), j=1,ntpdl)
  100 continue
      if (irandm.eq.1) return
c        Read variance matrix for input forcing fn
      read (5,13) icros,ivv,iaa,ivz,iaz,iav
      write (6,14) icros,ivv,iaa,ivz,iaz,iav
   20 read (5,4)  ncvdf,nroot,nsmax,corri
      write (6,5) ncvdf,nroot,nsmax,corri
      do 200 i=1,ncvdf
      read (5,6) nl(i),n2(i),ntacv
      write (6,7) nl(i),n2(i),ntacv
      nxl=nl(i)
      nx2=n2(i)
      read (5,9) (nt(nxl,nx2,j),ax(nxl,nx2,j),j=1,ntacv)
      write (6,8) (nt(nxl,nx2,j),ax(nxl,nx2,j),j=1,ntacv)
  200 continue
    1 format (4i5,f10.5)
    2 format (2i5)
    3 format (5(i5,f10.5))
    4 format (3i5,f10.7)
    5 format (//1x,'TOTAL # IN COVARIANCE MATRIX FOR FORCING FUNCTION ',
     1'[ NCVDF ] = ',i5///1x,'NUMBER OF EIGENVECTORS REQUIRE IN REPRESEN
     2TING THE MEAM RESPONSE [ NROOT ] =',i2///1x,'NUMBER OF SWEEPING ',
     3'REQUIRE IN SUBSPACE ITERATION [ NSMAX ] = ',i5///1x,
     4'FRACTION OF STANDARD DEVIATION OF STIFFNESS [ CORRI ] = ',f10.5)
```

```
 6 format (4(i5,i5,i5))
 7 format(2(/),1x,' PAIR [',I2,',',I2,']  ','TOTAL POINTS OF ',
  1'CHANGING AMPLITUDE [ NTACV ] =',I5)
 9 format (5(i5,f10.4))
 8 format (' TIME INCREMENT ',2x,i5,2x,' AMPLITUDE ',e13.6)
10 format (10(/),5x,'FLAG INDICATING RANDOM ANALYSIS [IRANDM] = ',
  1  i5,/5x,' :1  NO',/5x,' :2 YES'///5x,' LINEAR OR NONLINEAR',
  2 ' ANALYSIS [LON] = ',i5,/5x,':1 LINEAR'/5x,':2 NONLINEAR',
  3      ///5x,'TOTAL # DOF THAT HAVE LOAD [ NDOFL ] =',i5,
  4             ///5x,'TOTAL TIME INCREMENT [ NTI ] =',i5,
  5             ///5x,'TIME INCREMENT [ DT ] =',F10.5)
11 format (5(/),5x,i5,'  DOF INPUT LOADING FUNCTION ECHO',
  1  /(5x,'time increment',2x,i5,5x,' amplitude',3x,e10.5))
12 format (////5x,'FLAG INDICATING COMPUTING MEAN SQUARE VELOCITY =',
  1           i5/5x,'0 : NO'/5x,'1 : YES',
  2     ////5x,'FLAG INDICATING COMPUTING MEAN SQUARE ACCELERATION =',
  3           i5/5x,'0 : NO'/5x,'1 : YES')
13 format (6i5)
14 format (////5x'FLAG INDICATE COMPUTING CROSS MOMENT [ ICROS ] =',
  1           I5/5X,'0 : NO'/5X,'1 : YES',//5X,'[ IVV ] =',
  2           I5/5X,'0 : NO'/5X,'1 : YES',//5X,'[ IAA ] =',
  3           I5/5X,'0 : NO'/5X,'1 : YES',//5X,'[ IVZ ] =',
  4           I5/5X,'0 : NO'/5X,'1 : YES',//5X,'[ IAZ ] =',
  5           I5/5X,'0 : NO'/5X,'1 : YES',//5X,'[ IAV ] =',
  6           I5/5X,'0 : NO'/5X,'1 : YES')
   return
   end
```

```
      subroutine eigen
      common /modal/sk(16,16),sm(16,16),qk(16,16),xLam(16,2)
      dimension beta(2),phi(2)
      double precision sk,sm,qk,xLam,beta,phi
      real judge
      a=sm(1,1)*sm(2,2)-sm(1,2)*sm(1,2)
      b=2.*sk(1,2)*sm(1,2)-sm(1,1)*sk(2,2)-sm(2,2)*sk(1,1)
      c=sk(1,1)*sk(2,2)-sk(1,2)*sk(1,2)
      judge=b*b-4.*a*c
      if (judge.ge.0.) go to 500
      write (6,1)
      stop
  500 judge=judge**.5
      if (a.ne.0.) go to 110
      write (6,8)
      stop
  110 xLam(1,1)=(-b-judge)/(2.*a)
      xLam(2,1)=(-b+judge)/(2.*a)
      do 100 i=1,2
      deno=sk(2,2)-xLam(i,1)*sm(2,2)
      if (deno.ne.0.) go to 120
      write (6,9)
      stop
  120 beta(i)=(xLam(i,1)*sm(1,2)-sk(1,2))/deno
  100 phi(i)=(sm(1,1)+2.*sm(1,2)*beta(i)     1       sm(2,2)*beta(i)*beta(i))**-
5
      qk(1,1)=phi(1)
      qk(2,1)=beta(1)*phi(1)
      qk(1,2)=phi(2)
      qk(2,2)=phi(2)*beta(2)
    1 format (//5x,'B**2 - 4 A C < 0  IN SUBROUTINE EIGEN, STOP !!')
    8 format (//5x,'A IS ZERO IN SUBROUTINE EIGEN, ERROR, STOP!!')
    9 format (//5x,'DENO IS ZERO IN SUBROUTINE EIGEN, ERROR, STOP!!')
      return
      end
```

```
      program fda
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/dya/ mass,inlod,alpha,beta,mdi
      common/lod/ fi(50,12),ax,ay,az
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),axx(50,50,10),
     1                nt(50,50,10),lon,nroot,nsmax,corri
      common/rest/res(50)
      double precision a,b,fi,xmass,damp,res
c          read and write structure data
      call input
c          write band width
      write(6,1060) mband
 1060 format(10(/),13h Band Width =,i5,//)
c          initialize stiffness matrix (a)
      numeq=numnp*nq
      do 5 i=1,numeq
      do 5 j=1,mband
      a(i,j)=0.0
    5 continue
c          generate stiffness matrix (a)
      call stiff
c          write stiffness matrix (A), and mass matrix
      if (wstif.eq.0) go to 101
      write(6,1005)
 1005 format(46x,35h The Structure Stiffness Matrix (A)/)
      do 10 i=1,numeq
   10 write(6,1010) i,(a(i,j),j=1,mband)
 1010 format(/i5,6e19.7/(5x,6e19.7))
c                   Form Damping Matrix
  101 go to (11,102), method
  102 call damping
c        Add concentrated mass
      if (mass.eq.0) go to 500
      do 510 ii=1,numeq
  510 xmass(ii)=xmass(ii)+res(ii)
  500 continue
      if (wstif.eq.0) go to 11
   11 write (6,1004)
      format (10(/),46x,'The Structure Mass Matrix')
      do 9 i=1,numeq
      write (6,1010) i,xmass(i)
      write (6,1031)
      format (10 ,46x,'The Structure DAMPING Matrix')
              = numeq
              i, damp(i,j),j=1,mband)
              conditions to stiffness matrix (a) and mass matrix

                  ed stiffness matrix (a)
                    4    22
```

```
      write(6,1020)
 1020 format(10(/),21x,42hThe Reduced Structure Stiffness Matrix (A))
      do 20 i=1,numeq
   20 write(6,1010) i,(a(i,j),j=1,mband)
      go to (12,23),method
   23 write(6,1006)
 1006 format(10(/),21x,'The Reduced Mass Matrix (XMASS)')
      do 27 i=1,numeq
   27 write(6,1010) i,xmass(i)
      write (6,200)
 200  format (10(/),21x,'The Reduced damping Matrix (DAMP)')
      do 31 i=1,numeq
 31   write (6,1010) i,(damp(i,j),j=1,mband)
 22   go to (12,90), method
c           reduce system stiffness matrix (a)
 12   call solver(1)
      do 1050 jk=1,nlc
c           initialize load vector (b)
      do 45 i=1,numeq
      b(i)=0.0
   45 continue
c           read structure loads
      read (5,1140) nlnd,nlmem,ax,ay,az
 1140 format (2i5,3e10.4)
      call load
c           write reduced load vector (b)
      if (wlod.eq.0) goto 13
      write(6,1015)
 1015 format(    47x,27hThe Reduced Load Vector (B),/)
      do 15 i=1,numnp
      n=nq*i-(nq-1)
      nn=nq*i
      write (6,1025) i,(b(k),k=n,nn)
   15 continue
c           compute nodal displacements
   13 call solver(2)
c           write nodal displacements
      write(6,1035)
 1035 format (10(/),37x,44hS T R U C T U R E   D I S P L A C E M E N T S/
     1/)
      write(6,1070)
 1070 format(2x,4hNode,6x,6hU (in),13x,6hV (in),13x,6hW (in),13x,
     1          12hThetax (rad),7x,12hThetay (rad),7x,12hThetaz (rad)/)
      do 30 i=1,numnp
      n=nq*i-(nq-1)
      nn=nq*i
      write (6,1025) i,(b(k),k=n,nn)
   30 continue
c           compute and write member forces
      call force
 1050 continue
 1025 format (i5,6e19.7)
      go to (91,90),method
   90 call dload
```

```
      go to (699,698) ,mdi
 698  go to (691,692) ,lon
 691  call dicd
      go to 697
 692  call dicdn
      go to 697
 699  call  dinm
 697  continue
  91  write (6,1036)
1036  format (//,10x,'* * * END OF OUTPUTS * * *',10(/))
  92  stop
      end
```

```
      subroutine force
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/slv/ a(50,25),b(50)
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      common /lod/ fi(50,12),ax,ay,az
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      dimension u(12),f(12),ndf(12)
      double precision a,b,s,r,t,st,tf,u,f,fi
      mprint=0
      do 5 id=1,numem
c          compute control counters
      i=nd(id,1)
      j=nd(id,2)
      m=mtype(id)
      do 2 k=1,nq
      kk=k+nq
      ndf(k)=nq*i-(nq-k)
      ndf(kk)=nq*j-(nq-k)
    2 continue
      do 10 ii=1,neq
      il=ndf(ii)
      u(ii)=b(il)
   10 continue
c          compute member forces for structure axes
      go to (11,12,11,16), kind
   11 call strut  (id,i,j)
      go to 14
   12 if (mcurv(m).ne.0) call curvbm (id,i,j)
      if (mcurv(m).ne.0) go to 14
      call beam2 (id,i,j)
      go to 14
   16 call beam (id,i,j)
   14 do 15 ii=1,neq
      f(ii)=0.0
      do 15 jj=1,neq
      f(ii)=f(ii)+s(ii,jj)*u(jj)
   15 continue
      do 20 ii=1,neq
      p(ii)=f(ii)+fi(id,ii)
   20 continue
c          transform forces to member axes
      call rotate (3,id)
c          write member forces
      if (mprint.ne.0) go to 25
      mprint=54
      write(6,1030)
 1030 format (10(/),47x,25hM E M B E R   F O R C E S//)
      write(6,2000)
 2000 format(7h member,3x,4hnode,5x,8hforce(x),9x,8hforce(y),9x,
```

```
     18hforce(z),9x,9hmoment(x),8x,9hmoment(y),8x,9hmoment(z)/)
  25 mprint=mprint-1
     nf=nif/2
     write (6,1005) id,i,(tf(lk),lk=1,nf)
1005 format(/i5,3x,i5,6e17.7)
     nf1=nf+1
     write (6,1006) j,(tf(lk),lk=nf1,nif)
1006 format(8x,i5,6e17.7/)
   5 continue
     return
     end
```

```
      subroutine gauss (isolve,nord)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      double precision rf,rz,rz1,rz12,a1,a2,a3,xLa
      go to (800,850), isolve
 800  do 500 i=1,numeq
      do 500 j=1,numeq
      if (i.gt.j) go to 515
      if((i-1).Le.0) go to 510
      a2(i,j)=a1(i,j)
      do 505 k=1,i-1
 505  a2(i,j)=a2(i,j)-a3(i,k)*a2(k,j)
      go to 520
 510  a2(i,j)=a1(i,j)
      go to 520
 515  a3(i,j)=a2(j,i)/a2(j,j)
      go to 500
 520  if (i.ne.j) go to 500
      a3(i,i)=1.
 500  continue
      go to 888
c        foward substitution
 850  do 900 nn=1,nord
      do 600 i=1,numeq
      if (i.eq.1)  go to 605
      xLa(i,nn)=rz12(i,nn)
      do 610 j=1,i-1
 610  xLa(i,nn)=xLa(i,nn)-a3(i,j)*xLa(j,nn)
      go to 600
 605  xLa(1,nn)=rz12(1,nn)
 600  continue
c        backward substitution
      do 620 i=1,numeq
      it=numeq-i+1
      if (it.eq.numeq) go to 625
      rz12(it,nn)=xLa(it,nn)
      do 630 j=it+1,numeq
 630  rz12(it,nn)=rz12(it,nn)-a2(it,j)*rz12(j,nn)
      rz12(it,nn)=rz12(it,nn)/a2(it,it)
      go to 620
 625  rz12(numeq,nn)=xLa(numeq,nn)/a2(numeq,numeq)
 620  continue
 900  continue
 888  return
      end
```

```
      subroutine higheig (nmc,numeq,mband)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/xmd/ xmass(50),damp(50,25
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      common/matrix/ La(50),t1(50),t2(50),f(50)
      common/slv/ a(50,25),b(50)
      dimension xrho(2)
      double precision rf,rz,rz1,rz12,a,b,xmass,damp,t1,t2,f
      double precision eigvec,rho,a4
      nmc=0
      tole=1.0 E-6
c        transfer stiffness matrix
      do 150 i=1,numeq
      do 150 j=1,mband
      t2(i)=1.
  150 rf(i,j+i-1)=rz1(i,j)
      do 155 I=1,numeq
      do 155 j=I,numeq
      if (i.eq.j) go to 155
      rf(j,i)=rf(i,j)
  155 continue
      do 160 i=1,numeq
      call range (i,numeq,mband,mg,mbd)
      b(i)=0.
      do 160 j=mg,mbd
  160 b(i)=b(i)+rf(i,j)*t2(j)
  165 nmc=nmc+1
      do 170 i=1,numeq
  170 t1(i)=b(i)/xmass(i)
      do 180 i=1,numeq
      call range (i,numeq,mband,mg,mbd)
      t2(i)=0.
      do 180 j=mg,mbd
  180 t2(i)=t2(i)+rf(i,j)*t1(j)
      xrho1=0.
      xrho2=0.
      do 185 i=1,numeq
      xrho1=xrho1+t1(i)*t2(i)
  185 xrho2=xrho2+t1(i)*b(i)
      xrho(1)=xrho1/xrho2
      if (nmc.eq.1) go to 200
      if ((1.-abs(xrho(2)/xrho(1))).Le.tole) go to 210
  200 do 205 i=1,numeq
  205 b(i)=t2(i)/(xrho2**.5)
      xrho(2)=xrho(1)
      go to 165
  210 call solver (1)
      call solver (2)
      do 220 i=1,numeq
  220 eigvec(i,8)=b(i)
      rho(8)=xrho(1)
c        Transfer the stiffness back
      do 230 i=1,numeq
      do 230 j=1,mband
```

```
230 a(i,j)=rz1(i,j)
    return
    end
```

```
      subroutine input
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dya/ mass,inlod,alpha,beta,mdi
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      common /lod/ fi(50,12),ax,ay,az
      common/mlib2/ tf(25),bf(25),tw(25),bw(25)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common/rest/res(50)
      dimension title(20)
      double precision fi,sl,dsqrt,x1,x2,y1,y2,z1,z2,res
c          read and write control parameters
      read (5,990) title
  990 format(20a4)
      write(6,995) title
  995 format(15(/),43x,'Run Title: ',20a4//)
      read (5,1000) method,kind,numnp,nstrut,nbeam,nmtype,numem,numat,
     1             nrnp,nmrel,nlc
      read (5,999) wstif,wrstif,wlod
      go to (54,55,56,57), kind
   54 neq=4
      nq=2
      nif=2
      go to 58
   55 neq=6
      nq=3
      nif=6
      go to 58
   56 neq=6
      nq=3
      nif=2
      go to 58
   57 neq=12
      nq=6
      nif=12
   58 continue
 1000 format(11i5)
  999 format (3i5)
    6 write(6,1005)
 1005 format (43x,34hC O N T R O L   P A R A M E T E R S//)
      write (6,1010) method,numnp,nstrut,nbeam
      write (6,1009) numem,nmtype,numat,nrnp,nmrel,nlc
 1010 format(42x,'Type of problem [ method ]',i5/
     1       42x,'=1  static analysis'/
     2       42x,'=2  dynamic analysis'//
     3       42x,32hNumber of Nodal Points          =,i5//
     4       42x,32hNumber of Two-force Members     =,i5//
     5       42x,32hNumber of Beam Members          =,i5//)
 1009 format(42x,32hTotal Number of Members         =,i5//
     1       42x,32hNumber of Members in Library    =,i5//
```

```
    2        42x,32hNumber of Materials             =,i5//
    3        42x,32hNumber of Boundary Constraints =,i5//
    4        42x,32hNumber of Member Releases       =,i5//
    5        42x,32hNumber of Loading Conditions   =,i5)
      write  (6,1011) wstif,wrstif,wlod
 1011 format (///36x,45hStiffness Matrix Print Flag (1=yes,0=no)=     ,
    1   i5//36x,45hReduced Stiffness Matrix Matrix (1=yes,0=no)=,i5//
    2        36x,45hReduced Load Vector (1=yes,0=no)=           ,i5//)
      go to ( 7,8) ,method
c        read and write the flag of method of solving
c        problem for dynamic analysis
  8   read (5,1129) mass,inlod,mdi
1129  format (3i5)
      write (6,1131) mass,inlod
 1131 format (10(/),36x,' MASS = ',i5/
    1 36x,' = 0  NO CONCENTRATED MASS WAS ADDED '/
    2 36x,' > 0  ADD CONCENTRATED MASS FOR [ MASS ] TIMES'//
    3 36x,' INLOD = ',i5/
    4 36x,' = 1   READ FROM DATA  CARD '/
    5 36x,' = 2   READ FROM FILE')
      write (6,1192) mdi
1192  format (/36x,' MDI = ',i5,
    1        /36x,'     = 1  Newmark method ',
    2        /36x,'     = 2  Central Difference Method')
      if (mass.eq.0) go to 150
      do 155 in=1,mass
 155  read(5,156) mndof, res (mndof)
 156  format (4(i5,e10.5))
 150  continue
c  read damping (alpha and beta) [C]= alpha[M] + beta[K]
      read (5,1132) alpha,beta
 1132 format (2f10.5)
      write (6,1133) alpha,beta
 1133 format (10(/), 36x,' DAMPING MATRIX = ALPHA [M] + BETA [K] '/
    1                36x,' ALPHA = ',f15.8/
    2                36x,' BETA  = ',f15.8)
c          read and write of member property table
 7    write (6,1036)
 1036 format(10(/),25x,51hT A B L E   O F   M E M B E R   P R O P E R T
    1I E S//)
      write (6,1037)
 1037 format(42x,18hMoments of Inertia,18x,21hEffective Shear Areas//1x,
     16hMember,10x,8hX - Sect,7x,8hZ - Axis,7x,8hY - Axis,7x,8hX - Axis,
     27x,8hY - Axis,7x,8hZ - Axis/2x,4hType,14x,4hArea,90x,'Half depth')
      do 11 im=1,nmtype
      read(5,1038) tf(im),bf(im),tw(im),bw(im),xj(im),avy(im),
    1               avz(im),mcurv(im)
   11 continue
      do 303 im=1,nmtype
      xa(im)=bf(im)*tf(im)*2.+bw(im)*tw(im)*2.
      zi(im)=bf(im)*(bw(im)+tf(im))**3-(bf(im)-tw(im))*bw(im)**3
      zi(im)=zi(im)*2./3.
  303 yi(im)=(tf(im)*bf(im)**3+bw(im)*tw(im)**3)/6.
      do 12 im=1,nmtype
```

```
      write (6,1039) im,xa(im),zi(im),yi(im),xj(im)
     1,avy(im),avz(im),mcurv(im),bw(im)
   12 continue
 1038 format (7e10.3,i2)
 1039 format (i5,5x,6f15.3,5x,i3,f15.3)
c          read and write of material constants
      write(6,1015)
 1015 format (10(/),42x,34hM A T E R I A L  C O N S T A N T S//)
    2 do 10 im=1,numat
      read (5,1020) e(im),sigma(im),pr(im),rho(im)
      g(im)=.5*e(im)/(1.+pr(im))
      epsiln(im)=sigma(im)/e(im)
      write (6,1024) im
 1024 format (48x,'Properties for Material ',i2)
      write (6,1025)e(im),sigma(im),epsiln(im),pr(im),g(im),rho(im)
   10 continue
 1020 format(4e10.2)
 1025 format(      42x,21hYoungs Modulus      =,e16.7/
     1             42x,21hYield stress        =,e16.7/
     2             42x,21hYield strain        =,e16.7/
     3             42x,21hPoissons Ratio      =,e16.7/
     4             42x,21hShear Modulus       =,e16.7/
     5             42x,21hMass Density        =,e16.7/)
c          read and write nodal point coordinates
      do 15 n=1,numnp
      read (5,1120) nn,x(nn),y(nn),z(nn)
 1120 format(i5,3e10.4)
      if (n.gt.1) go to 14
      write(6,1030)
 1030 format (10(/),43x,32hJ O I N T  C O O R D I N A T E S//)
      write(6,1031)
 1031 format(35x,4hNode,11x,1hX,15x,1hY,15x,1hZ/
     1        48x,4h(in),12x,4h(in),12x,4h(in)/)
   14 write(6,1035) nn,x(nn),y(nn),z(nn)
 1035 format(33x,i5,1x,3f16.4)
   15 continue
c          read and write of member descriptions
      nhw=0
      do 20 ii=1,numem
      read (5,1040) id,(nd(id,k),k=1,2),mtype(id),mid(id),mlc(id),
     1               alfa(id),thetai(id),thetaj(id)
 1040 format (6i5,3f10.4)
      i=nd(id,1)
      j=nd(id,2)
      x1=x(i)
      x2=x(j)
      y1=y(i)
      y2=y(j)
      z1=z(i)
      z2=z(j)
      sl=dsqrt((x2-x1)**2+(y2-y1)**2+(z2-z1)**2)
      mt=mtype(id)
      if (ii.gt.1) go to 22
      write (6,1045)
```

```
1045 format(10(/),41x,37hM E M B E R   D E S C R I P T I O N S//)
     write (6,1101)
1101 format(2x,6hMember,3x,4hLeft,3x,5hRight,5x,6hLength,7x,4hArea,8x,
    12hIx,9x,2hIy,9x,2hIz,8x,3hAvy,9x,3hAvz,4x,5hMat'l,3x,7hLoading/2x,
    26hNumber,3x,4hNode,3x,4hNode,8x,1hL,10x,4hL**2,7x,4hL**4,7x,4hL**4
    3,7x,4hL**4,7x,4hL**2,6x,4hL**2/)
  22 write (6,1100) id,(nd(id,k),k=1,2),sl,xa(mt),xj(mt),yi(mt),zi(mt),
    1               avy(mt),avz(mt),mid(id),mlc(id)
1100 format (1x,i5,2i8,f13.2,6f11.2,i8,3x,i8)
c          compute half band width mband
     if(j-i) 35,40,45
  40 write(6,1055) id
1055 format(42h Identical End Nodal Points for Member no.,i4)
  35 ji=nq*j-nq+1
     ij=nq*i
     go to 50
  45 ij=nq*i-nq+1
     ji=nq*j
  50 nbd=iabs(ji-ij)
     if(nbd-nhw) 20,20,25
  25 nhw=iabs(ji-ij)
  20 continue
     mband=nhw+1
c          read boundary restraint codes
     do 34 nn=1,numnp
     ntype(nn)=0
     do 34 j=1,nq
     ui(nn,j)=0.0
  34 ir(nn,j)=0
     write (6,1135)
1135 format (10(/),45x, 'B O U N D A R Y   C O N D I T I O N S',/,
    1    2x,'Node',3x,'Init.',3x,'Boundary',6x,'X Initial',6x,'Y Initia
    2l',6x,'Z Initial',4x,'X Init. Rotat. Y Init. Rotat. Z Init. Rotat.
    3',/,' Number  Disp.  Constraint',7x,'(L)',12x,'(L)',12x,'(L)',13x,
    4'(Deg)',10x,'(Deg)',10x,'(Deg)',/)
     do 60 nb=1,nrnp
     read (5,1125) n,ntype(n),(ir(n,j),j=1,6),(ui(n,j),j=1,6)
     write(6,1130) n,ntype(n),(ir(n,j),j=1,6),(ui(n,j),j=1,6)
  60 continue
1125 format (i3,i1,6i1,6f10.4)
1130 format (5x,i3,5x,i1,2x,6i1,5x,6e15.6)
c          read and write member releases
     do 31 im=1,numem
     do 32 j=1,2
  32 mrel(im,j)=0
  31 continue
     if (nmrel.eq.0) go to 36
     write (6,1060)
1060 format (10(/),40x,39hM E M B E R   R E L E A S E   C O D E S/)
     do 33 mr=1,nmrel
     read (5,1065) im,(mrel(im,j),j=1,2)
     write(6,1070) im,(mrel(im,j),j=1,2)
  33 continue
1065 format (i3,/,i1,/,i1)
```

```
1070 format (53x,i5,5x,i1,2x,i1)
36    continue
 53   return
      end
```

```
      subroutine jacobi (n,nsmax)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,ifpr,wrstif,wlod
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),d(50,50),La2(50,50)
      common/modal/ a(16,16),b(16,16),x(16,16),eigv(16,2)
      double precision a1,a2,a3,d,a,b,x,eigv
      iout=6
      rtol=1.0 E-6
c        initialize iegenvalue and eigenvector mtrices
      do 10 i=1,n
      if (a(i,i).gt.0..and.b(i,i).gt.0.) go to 4
      write (iout,2040)
      stop
    4 d(i,1)=a(i,i)/b(i,i)
   10 eigv(i,1)=d(i,1)
      do 30 i=1,n
      do 20 j=1,n
   20 x(i,j)=0.
   30 x(i,i)=1.
      if (n.eq.1) return
c        Initialize sweep counter and begin iteration
      nsweep=0
      nr=n-1
   40 nsweep=nsweep+1
      if (ifpr.eq.1) write (iout,2000) nsweep
c        check if present off-diagonal element is large enough to
c        reguire zeroing
      eps=(.01**nsweep)**2
      do 210 j=1,nr
      jj=j+1
      do 210 k=jj,n
      eptola=(a(j,k)*a(j,k))/(a(j,j)*a(k,k))
      eptolb=(b(j,k)*b(j,k))/(b(j,j)*b(k,k))
      if ((eptola.Lt.eps).and.(eptolb.Lt.eps)) go to 210
c       if zeroing is required , calculate the botation matrix elements
c       ca and cg
      akk=a(k,k)*b(j,k)-b(k,k)*a(j,k)
      ajj=a(j,j)*b(j,k)-b(j,j)*a(j,k)
      ab=a(j,j)*b(k,k)-a(k,k)*b(j,j)
      check=(ab*ab+4.*akk*ajj)/4.
      if (check) 50,60,60
   50 write (iout,2020)
      stop
   60 sqch=sqrt(check)
      d1=ab/2.+sqch
      d2=ab/2.-sqch
      den=d1
      if (abs(d2).gt.abs(d1)) den=d2
      if (den) 80,70,80
   70 ca=0.
      cg=-a(j,k)/a(k,k)
      go to 90
   80 ca=akk/den
```

AD-A179 772 DAMAGE DIAGNOSIS FOR ELASTO-PLASTIC STRUCTURES(U) NEW MEXICO UNIV ALBUQUERQUE DEPT OF MECHANICAL ENGINEERING F D JU ET AL. DEC 86 ME-138(86)AFOSR-993-2 3/3

UNCLASSIFIED AFOSR-TR-87-0413 AFOSR-85-0085 F/G 20/11 NL


1.0
2.8
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

```
      cg=-ajj/den
c        perform generalize rotation to zero the present off-diagonal
c        element
 90   if (n-2) 100,190,100
 100  jp1=j+1
      jm1=j-1
      kp1=k+1
      km1=k-1
      if (jm1-1) 130,110,110
 110  do 120 i=1,jm1
      aj=a(i,j)
      bj=b(i,j)
      ak=a(i,k)
      bk=b(i,k)
      a(i,j)=aj+cg*ak
      b(i,j)=bj+cg*bk
      a(i,k)=ak+ca*aj
 120  b(i,k)=bk+ca*bj
 130  if (kp1-n) 140,140,160
 140  do 150 i=kp1,n
      aj=a(j,i)
      bj=b(j,i)
      ak=a(k,i)
      bk=b(k,i)
      a(j,i)=aj+cg*ak
      b(j,i)=bj+cg*bk
      a(k,i)=ak+ca*aj
 150  b(k,i)=bk+ca*bj
 160  if (jp1-km1) 170,170,190
 170  do 180 i=jp1,km1
      aj=a(j,i)
      bj=b(j,i)
      ak=a(i,k)
      bk=b(i,k)
      a(j,i)=aj+cg*ak
      b(j,i)=bj+cg*bk
      a(i,k)=ak+ca*aj
 180  b(i,k)=bk+ca*bj
 190  ak=a(k,k)
      bk=b(k,k)
      a(k,k)=ak+2.*ca*a(j,k)+ca*ca*a(j,j)
      b(k,k)=bk+2.*ca*b(j,k)+ca*ca*b(j,j)
      a(j,j)=a(j,j)+2.*cg*a(j,k)+cg*cg*ak
      b(j,j)=b(j,j)+2.*cg*b(j,k)+cg*cg*bk
      a(j,k)=0.
      b(j,k)=0.
c        update the eigenvector matrix after each rotation
      do 200 i=1,n
      xj=x(i,j)
      xk=x(i,k)
      x(i,j)=xj+cg*xk
 200  x(i,k)=xk+ca*xj
 210  continue
c        update the iegenvector after each sweep
```

```
      do 220 i=1,n
      if (a(i,i).gt.0..and.b(i,i).gt.0.) go to 220
      write (iout,2050)
      stop
 220  eigv(i,1)=a(i,i)/b(i,i)
      if (ifpr.eq.0) go to 230
      write (iout,2030)
      write (iout,2010) (eigv(i,1),i=1,n)
c        check for convergence
 230  do 240 i=1,n
      toL=rtoL*d(i,1)
      dif=dabs(eigv(i,1)-d(i,1))
      if (dif.gt.toL) go to 280
 240  continue
c        check all off-diagonal elements to see
c        if another sweep is required
      eps=rtoL**2
      do 250 j=1,nr
      jj=j+1
      do 250 k=jj,n
      epsa=(a(j,k)*a(j,k))/(a(j,j)*a(k,k))
      epsb=(b(j,k)*b(j,k))/(b(j,j)*b(k,k))
      if ((epsa.Lt.eps).and.(epsb.Lt.eps)) go to 250
      go to 280
 250  continue
c        fill out bottom triangle of resultant
c        matrices and scale eigenvectors
 255  do 260 i=1,n
      do 260 j=1,n
      a(j,i)=a(i,j)
 260  b(j,i)=b(i,j)
      do 270 j=1,n
      bb=dsqrt(b(j,j))
      do 270 k=1,n
 270  x(k,j)=x(k,j)/bb
      return
c        update D matrix and start new sweep,if allowed
 280  do 290 i=1,n
 290  d(i,1)=eigv(i,1)
      if (nsweep.Lt.nsmax) go to 40
      go to 255
 2000 format (5x,'SWEEP NUMBER IN JACOBI = ',i5)
 2010 format (6e20.12)
 2020 format (10x,'MATRIX NOT POSITIVE DEFINE ( B**2-4AC < 0), STOP!')
 2040 format (10x,'MATRIX NOT POSITIVE DEFINE (DIAGONAL < 0), STOP!')
 2030 format (10x,'CURRENT EIGENVALUE IN JACOBI ARE :')
 2050 format (10x,'MATRIX NOT POSITIVE DEFINE (DIAGONAL < 0,FOR UPDATE'
     1       ,' STIFFNESS AND MASS MATRIX), STOP!')
      end
```

```
      subroutine load
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/slv/ a(50,25),b(50)
      common/mem/ mtype(50),ij(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      common /lod/ fi(50,12),ax,ay,az
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      dimension u(12)
      double precision a,b,s,r,t,st,tf,u,fi,sl,dsqrt,xp,yp,zp,xzp
c        initialize routine
      nq1=nq+1
      do 5 k=1,numem
      do 5 l=1,neq
      p(l)=0.0
      u(l)=0.0
      fi(k,l)=0.0
    5 continue
      if (nlnd) 410,410,360
c          read and write concentrated nodal loads
  360 write(6,361)
  361 format(    37x,45hC O N C E N T R A T E D  J O I N T  L O A D S///
     116x,5hJoint,6x,8hForce(X),6x,8hForce(Y),6x,8hForce(Z),7x,
     29hMoment(X),5x,9hMoment(Y),5x,9hMoment(Z)/)
      do 405 l=1,nlnd
      read (5,1120) nl,(p(lk),lk=1,6)
 1120 format(i5,/,5(f10.4,/),f10.4)
      write(6,362) nl,(p(lk),lk=1,nq)
  362 format(14x,i5,2x,3f14.4,3f14.2)
c          compute load vector (B)
      do 405 kk=1,nq
      k=nq*nl-(nq-kk)
      b(k)=p(kk)
  405 continue
c.         read and/or compute and write member loads
c          member loads due to gravity
  410 do 300 im=1,numem
      m=mid(im)
      in=mtype(im)
      w=ay*xa(in)*rho(m)
      i=ij(im,1)
      j=ij(im,2)
      xp=x(j)-x(i)
      yp=y(j)-y(i)
      zp=z(j)-z(i)
      sl=dsqrt(xp**2+yp**2+zp**2)
      xzp=dsqrt(sl**2-yp**2)
      do 10 ii=1,12
   10 p(ii)=0.0
      if (mrel(im,1).eq.0.and.mrel(im,2).eq.0) go to 100
```

```
      if (mrel(im,1).gt.0.and.mrel(im,2).gt.0) go to 105
      if (mrel(im,1).gt.0.and.mrel(im,2).eq.0) go to 110
      if (mrel(im,1).eq.0.and.mrel(im,2).gt.0) go to 115
  100 call dld (1,w,yp,xzp,sl)
      go to 120
  105 call dld (2,w,yp,xzp,sl)
      go to 120
  110 call dld (3,w,yp,xzp,sl)
      go to 120
  115 call dld (4,w,yp,xzp,sl)
  120 continue
      go to (121,122,123,124), kind
  121 p(3)=p(7)
      p(4)=p(8)
      go to 124
  122 p(3)=p(6)
      p(4)=p(7)
      p(5)=p(8)
      p(6)=p(12)
      go to 124
  123 p(4)=p(7)
      p(5)=p(8)
      p(6)=p(9)
  124 continue
c          transform (P) to global coordinates.
      n=im
      call rotate (2,n)
      do 306 jj=1,neq
      fi(im,jj)=tf(jj)
  306 tf(jj)=-tf(jj)
c          merge gravity loads
      call merge (i,j,2)
  300 continue
  420 if (nlmem) 490,490,425
  425 write (6,600)
  600 format(10(/),29x,58hJ O I N T  F O R C E S  F R O M  L O A D E D M
     1 E M B E R S///11x,6hMember,3x,4hNode,5x,8hForce(X),6x,8hForce(Y),
     26x,8hForce(Z),7x,9hMoment(X),5x,9hMoment(Y),5x,9hMoment(Z)/)
      do 423 l=1,nlmem
      read(5,1121) mn,i,(p(lm),lm=1,6)
 1121 format(i5,/,i5,/,5(f10.4,/),f10.4)
      read(5,1122) j,(p(lm),lm=7,12)
 1122 format(i5,/,5(f10.4,/),f10.4)
  411 write(6,601) mn,i,(p(lk),lk=1,nq)
  601 format(10x,i5,3x,i5,3f14.4,3f14.2)
      write(6,602) j,(p(lk),lk=nq1,neq)
  602 format(18x,i5,3f14.4,3f14.2)
      n=mn
      call rotate (2,n)
      do 412 ii=1,neq
      fi(mn,ii)=fi(mn,ii)+tf(ii)
      tf(ii)=-tf(ii)
  412 continue
c          merge forces due to member loads
```

```
      call merge (i,j,2)
  423 continue
c          impose initial displacement boundary conditions
  490 do 424 l=1,numem
      i=ij(l,1)
      j=ij(l,2)
      m=mtype(l)
      if (ntype(i).lt.1) go to 30
      do 35 jj=1,nq
   35 u(jj)=ui(i,jj)
   30 if(ntype(i).lt.1.and.ntype(j).lt.1) go to 25
      do 40 jj=1,nq
      kk=jj+nq
   40 u(kk)=ui(j,jj)
      id=1
      go to (41,42,41,44), kind
   41 call strut  (id,i,j)
      go to 46
   42 if (mcurv(m).ne.0) call curvbm (id,i,j)
      if (mcurv(m).ne.0) go to 46
      call beam2 (id,i,j)
      go to 46
   44 call beam (id,i,j)
   46 do 50 jj=1,neq
      tf(jj)=0.0
      do 45 kk=1,neq
   45 tf(jj)=tf(jj)+s(jj,kk)*u(kk)
      tf(jj)=-tf(jj)
   50 continue
c          compute load vector (b)
      call merge (i,j,2)
   25 continue
  424 continue
      write (6,1024)
 1024 format (8(/),'The System Load Vector (B)',//)
      do 500 ii=1,numnp
      n=nq*ii-(nq-1)
      nn=nq*ii
      write (6,1025) ii,(b(k),k=n,nn)
  500 continue
 1025 format (i5,6e19.7)
c          reduce the load vector (B)
      call bound(2)
      return
      end
```

```
      subroutine loweig (nmc,numeq,mband)
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      double precision a,b,xmass,damp,t1,t2,f
      double precision rf,rz,rz1,rz12,eigvec,rho,a4
      nmc=0
      tole=1.00 E-06
      do 305 ii=1,numeq
      f(ii)=xmass(ii)
  305 b(ii)=f(ii)
  306 call solver (1)
  307 call solver (2)
      rho1=0.
      do 310 ii=1,numeq
  310 rho1=rho1+b(ii)*f(ii)
      do 315 ii=1,numeq
  315 f(ii)=b(ii)*xmass(ii)
      rho2=0.
      do 320 ii=1,numeq
  320 rho2=rho2+b(ii)*f(ii)
      nmc=nmc+1
      if (nmc.gt.1) go to 330
      rho(2)=rho1/rho2
      do 321 ii=1,numeq
      f(ii)=f(ii)/rho2**.5
  321 b(ii)=f(ii)
      go to 307
  330 rho(1)=rho1/rho2
      if (abs(1.-rho(1)/rho(2)).Le.tole) go to 350
      do 331 ii=1,numeq
      f(ii)=f(ii)/rho2**.5
  331 b(ii)=f(ii)
      rho(2)=rho(1)
      go to 307
  350 do 351 ii=1,numeq
  351 eigvec(ii,1)=f(ii)/(xmass(ii)*rho2**.5)
      return
      end
```

```
      subroutine matinv(n,wstif)
c        matrix inverse c: input, output still is c
c        where the matrix d is operating matrix
      common/rr3/c(50,50),a2(50,50),a3(50,50),d(50,50),la2(50,50)
      double precision c,d,a2,a3,p2,p3
      do 10 j=1,n
      do 10 k=1,n
 10   d(j,k)=0.
      do 11 k=1,n
 11   d(k,k)=1.
      do 55 i=1,n
      p2=c(i,i)
      do 40 j=1,n
      c(i,j)=c(i,j)/p2
 40   d(i,j)=d(i,j)/p2
      do 51 ic=1,n
      p3=-c(ic,i)
      do 50 k=1,n
      if (ic-i) 21,51,21
 21   c(ic,k)=c(i,k)*p3+c(ic,k)
      d(ic,k)=d(i,k)*p3+d(ic,k)
 50   continue
 51   continue
 55   continue
      do 70 it=1,n
      do 70 is=1,n
 70   c(it,is)=d(it,is)
      if (wstif.eq.0) return
      do 800 ii=1,n
 800  write (6,1000) (c(ii,jj),jj=1,n)
 1000 format (6e13.6)
      return
      end
```

```
      subroutine merge (i,j,imerge)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      common /xmss/ xms(12)
      dimension ndf(12)
      double precision a,b,s,r,t,st,tf,xmass,damp,xms
      go to (5,16), imerge
c          form structure stiffness matrix
    5 do 2 k=1,nq
      kk=k+nq
      ndf(k)=nq*i-(nq-k)
      ndf(kk)=nq*j-(nq-k)
    2 continue
      do 15 ii=1,neq
      k1=ndf(ii)
      do 15 jj=1,neq
      k2=ndf(jj)
      if(k2-k1) 15,10,10
   10 k3=k2-k1+1
      a(k1,k3)=a(k1,k3)+s(ii,jj)
      if (k3.eq.1) go to 51
      go to 15
   51 go to (15,26) ,method
   26 xmass(k1)=xmass(k1)+xms(ii)
   15 continue
      go to 25
c          form structure load vector
   16 do 20 ii=1,nq
      k=nq*i-(nq-ii)
      kk=nq*j-(nq-ii)
      inq=ii+nq
      b(k)=b(k)+tf(ii)
      b(kk)=b(kk)+tf(inq)
   20 continue
   25 continue
      return
      end
```

```
      subroutine offs (id,eps,nx,nz,ipass)
      common/engy/ strain(11,11,10,2),energd(10),dv(2)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common/off/ offset(11,11,10,2),toffs(11,11)
      m=mid(id)
      if (ipass.ne.10) go to 200
      strain(nx,nz,id,1)=strain(nx,nz,id,2)
      strain(nx,nz,id,2)=eps
      if (nz.eq.11) return
      avstn=.5*(strain(nx,nz,id,1)+strain(nx,nz+1,id,1))
      avoff=.5*(offset(nx,nz,id,1)+offset(nx,nz,id,1))
      if (nz.eq.1.or.nz.eq.10) goto 130
      delv=dv(2)
      go to 131
  130 delv=dv(1)
  131 energd(id)=energd(id)+(avstn-avoff)
     1           *(eps-strain(nx,nz,id,1))*delv
      return
  200 x0=(eps-offset(nx,nz,id,2))*e(m)
      x1=sigma(m)
      x2=-sigma(m)
      if (x0.gt.x1) go to 100
      if (x0.Lt.x2) go to 110
      toffs(nx,nz)=offset(nx,nz,id,2)
      return
100   toffs(nx,nz)=eps-epsiln(m)
      return
110   toffs(nx,nz)=eps+epsiln(m)
120   return
      end
```

```
      subroutine order (n,s,numeq)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/matrix/ La(50),t1(50),t2(50),f(50)
      double precision rf,rz,rz1,rz12,t1,t2,f
      do 100 ii=1,numeq
      if (s.Le.rz(ii,1)) go to 110
100   continue
      write (6,1)
      stop
110   if (ii-numeq) 115,125,125
115   it=numeq
      do 120 in=ii,numeq-1
      it=it-1
      rz(it+1,1)=rz(it,1)
120   La(it+1)=La(it)
      if (ii-1) 135,135,125
125   do 130 im=1,ii-1
      rz(im,1)=rz(im,1)
130   La(im)=La(im)
135   rz(ii,1)=s
      La(ii)=n
  1   format (10x,' S IS TOO BIG IN SUBROUTINE ORDER')
      return
      end
      subroutine range (ii,numeq,mband,mg,mbd)
      mbd=ii+mband-1
      if (mbd.gt.numeq) mbd=numeq
      mg=ii-mband+1
      if (mg.Le.0) mg=1
      return
      end
      subroutine rbound
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common /rest/ res(50)
      double precision res
      do 60 n=1,numnp
      do 60 kk=1,nq
      k=nq*n-(nq-kk)
      if (ir(n,kk).eq.0) go to 60
      res(k)=0.
60    continue
      return
      end
```

```
      subroutine restor (id,itt,nn)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      common/mlib2/ tf(25),bf(25),tw(25),bw(25)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common /off/ offset(11,11,10,2),toffs(11,11)
      common/rest/res(50)
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/engy/ strain(11,11,10,2), energd(10),dv(2)
      dimension epa(11),epz(11),epc(11),iw1(11),iw2(11),tryoff(11,11)
      dimension nyie(2),width(10),zx(5),dz(10)
      double precision sL,dsqrt,xp,yp,xdp,x1d,x2d,res
      double precision costhi,sinthi,c1,c2,c3,c4,c5,c6,q1,q2
      double precision xaa,xbb,ya,yb,thitaa,thitab,paxi
      iL=nd(id,1)
      jr=nd(id,2)
      xp=x(jr)-x(iL)
      yp=y(jr)-y(iL)
      sL=dsqrt(xp**2+yp**2)
      m=mid(id)
      n=mtype(id)
      dz1=bw(n)/4.
      dx=sL/10.
      ym=e(m)
      costhi=(x(jr)-x(iL))/sL
      sinthi=(y(jr)-y(iL))/sL
      dz(1)=tf(n)
      width(1)=bf(n)
      zx(1)=bw(n)+tf(n)/2.
      do 140 k1=2,5
      xk1=float (k1)
      width(k1)=tw(n)
      zx(7-k1)=(xk1-1.5)*dz1
  140 dz(7-k1)=dz1
      do 145 k1=1,5
      width(k1+5)=width(6-k1)
  145 dz(k1+5)=dz(6-k1)
      dv(1)=dx*dz(1)*width(1)
      dv(2)=dx*dz(2)*width(2)
      xAa=xdp(iL*3-2,3)*costhi+xdp(iL*3-1,3)*sinthi
      ya=-xdp(iL*3-2,3)*sinthi+xdp(iL*3-1,3)*costhi
      thitaA=xdp(iL*3,3)
      xBb=xdp(jr*3-2,3)*costhi+xdp(jr*3-1,3)*sinthi
      yb=-xdp(jr*3-2,3)*sinthi+xdp(jr*3-1,3)*costhi
      thitaB=xdp(jr*3,3)
      paxi=ym*xa(n)*(xBb-xAa)
      nii=0
      izero=0
```

```
      ipass=1
260   if (itt.ne.1) go to 100
      h3=0.
      h4=0.
      h5=0.
      go to 200
c        Evaluate h3 ,h4 and h5
100   if (nii.ne.0) go to 101
      if (nn.eq.2) go to 105
      do 103 ii=1,11
      do 103 jj=1,11
103   tryoff(ii,jj)=2.*offset(ii,jj,id,2)-offset(ii,jj,id,1)
      go to 101
105   do 106 ii=1,11
      do 106 jj=1,11
106   tryoff(ii,jj)=offset(ii,jj,id,2)
101   h3=0.
      do 110 jj=1,11
      epa(jj)=0.
      do 110 ii=1,5
      haxx1=(tryoff(jj,7-ii)+tryoff(jj,6-ii))*.5
      haxx2=(tryoff(jj,ii+5)+tryoff(jj,ii+6))*.5
110   epa(jj)=epa(jj)+(haxx1-haxx2)*zx(6-ii)*width(6-ii)*dz(6-ii)
      do 111 jj=1,11
      if (jj.eq.1.or.jj.eq.11) go to 112
      h3=h3+epa(jj)
      go to 111
112   h3=h3+.5*epa(jj)
111   continue
      h3=h3*ym*dx
      h4=0.
      do 121 ii=1,11
      epc(ii)=0.
      do 121 jj=1,ii
      if (jj.eq.1.or.jj.eq.ii) go to 122
      epc(ii)=epc(ii)+epa(jj)
      go to 121
122   epc(ii)=epc(ii)+epa(jj)*.5
121   continue
      do 123 ii=1,11
      if (ii.eq.1.or.ii.eq.11) go to 124
      h4=h4+epc(ii)
      go to 123
124   h4=h4+epc(ii)*.5
123   continue
      h4=h4*ym*dx**2
      do 310 jj=1,11
      epc(jj)=0.
      do 310 kk=1,10
      epc(jj)=epc(jj)+(tryoff(jj,kk)+tryoff(jj,kk+1))*.5
     1        *width(kk)*dz(kk)
310   continue
      h5=0.
      do 330 ii=1,11
```

```
      if (ii.eq.1.or.ii.eq.11) go to 331
      h5=h5+epc(ii)
      go to 330
331   h5=h5+epc(ii)*.5
330   continue
      h5=h5*dx*ym
200   pax=(paxi-h5)/sL
      q1=ym*zi(n)*(thitaB-thitaA)-h3
      q2=ym*zi(n)*(yB-yA-thitaA*sL)-h4
      c1=12.*(.5*sL*q1-q2)/(sL**3)
      c2=12.*(q2*.5-q1*sL/6.)/(sL**2)
      do 210 nx=1,11
      if (itt.eq.1) go to 216
      h1=epc(nx)*ym
      go to 217
216   h1=0.
217   if (itt.eq.1) go to 218
      h2=epa(nx)/zi(n)
      go to 202
218   h2=0.
202   cappa=(c1*dx*(nx-1)+c2)/(ym*zi(n))+h2
      epaa=(pax+h1)/(ym*xa(n))
      epcc=cappa*(bw(n)+tf(n))
      do 270 nz=1,4
      epz(6-nz)=epcc*dz1*nz/(bw(n)+tf(n))
270   epz(6+nz)=-epz(6-nz)
      epz(1)=epcc
      epz(11)=-epcc
      epz(6)=0.
      do 212 nz=1,11
212   epz(nz)=epz(nz)+epaa
      do 213 nz=1,11
      eps=epz(nz)
      call offs (id,eps,nx,nz,ipass)
213   continue
210   continue
      if (ipass.eq.10) go to 380
      do 220 i=1,11
      do 220 j=1,11
      if (toffs(i,j).eq.tryoff(i,j)) go to 220
      if (toffs(i,j).eq.0.and.tryoff(i,j).ne.0.)goto 221
      if (toffs(i,j).ne.0.and.tryoff(i,j).eq.0.)goto 222
      erx=(toffs(i,j)-tryoff(i,j))/tryoff(i,j)
      erx=abs(erx)
      izero=izero+1
      if (erx.Le.0.05) go to 220
      go to 240
221   erx=abs(tryoff(i,j))
      izero=izero+1
      if (erx.Le.0.05) go to 220
      go to 240
222   erx=abs(toffs(i,j))
      izero=izero+1
      if (erx.Le.0.05) go to 220
```

```
      go to 240
 220  continue
      if (izero.eq.0) go to 380
      go to 300
 240  nii=nii+1
c        print out the intermediate offset
      if (wstif.eq.0) go to 340
 350  write (6,24) id,nii
      do 360 i=1,11
      jx=0
      do 370 j=1,11
      if (toffs(i,j).eq.0.) go to 370
      jx=jx+1
      epc(jx)=toffs(i,j)
      iw2(jx)=j
      iw1(jx)=i
 370  continue
      if (jx.eq.0) go to 360
      write (6,22) (iw1(jj),iw2(jj),epc(jj),jj=1,jx)
 360  continue
 340  if (nii.eq.5) go to 300
      do 250 jj=1,11
      do 250 ii=1,11
 250  tryoff(ii,jj)=toffs(ii,jj)
      go to 260
c         Move the offset to the next step
 300  do 320 i=1,11
      do 320 j=1,11
      epc(j)=0.
      offset(i,j,id,1)=offset(i,j,id,2)
      offset(i,j,id,2)=(toffs(i,j)+tryoff(i,j))*.5
 320  tryoff(i,j)=offset(i,j,id,2)
      ipass=10
      go to 260
c       Restoring forces
 380  c3=(ym*xa(n)*(xbb-xaa)-h5)/sL
      c4=c1*sL+c2
      c5=-c1*sinthi-c3*costhi
      c6=c1*costhi-c3*sinthi
      res(3*iL-2)=res(3*iL-2)+c5
      res(3*iL-1)=c6+res(3*iL-1)
      res(3*iL)=-c2+res(3*iL)
      res(3*jr-2)=res(3*jr-2)-c5
      res(3*jr-1)=-c6+res(3*jr-1)
      res(3*jr)=c4+res(3*jr)
c         check plastic hinge
      if (nn.ne.1) return
      mm=0
      do 451 j=1,11,10
      mm=mm+1
      nyield=0
      do 450 i=1,11
      if (offset(j,i,id,2).eq.0.) go to 450
      nyield=nyield+1
```

```
 450   continue
       nyie(mm)=nyield
 451   continue
       do 452 is=1,2
       if (nyie(is).Lt.10) go to 452
 453   write (6,25) nd(id,is),id
 452   continue
c           Write down the final offset
 460   if (wlod.eq.0) return
       if (izero.eq.0) go to 440
 50    write (6,23) id
       do 431 i=1,11
       jx=0
       do 430 j=1,11
       if (offset(i,j,id,2).eq.0.) go to 430
       jx=jx+1
       epc(jx)=offset(i,j,id,2)
       iw2(jx)=j
       iw1(jx)=i
 430   continue
       if (jx.eq.0) go to 431
       write (6,22) (iw1(jj),iw2(jj),epc(jj),jj=1,jx)
 431   continue
       return
 440   write (6,20) id,nii
 20    format (' THE ITERATION TIMES OF OFFSET  AT ',i3,x,
      1'th MEMBER ARE',i5)
 22    format (6('[',i2,',',i2,'] =',e11.4,'; '))
 23    format (' THE FIANL OFFSET OF THE BEAM AT ',I3,' MEMBER')
 24    format (' THE ITERATED OFFSET AT',I4,' MEMBER FOR NII=',I5,
      1' ARE:')
 25    format (' PLASTIC HINGE OCCURS AT NODAL POINT',i3,' MEMBER #',I3)
 999   return
       end
```

```
      subroutine rotate (nrot,id)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1             nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1             mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1             avz(25),mcurv(25)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      double precision s,r,t,st,tf,cx,cy,cz,rad,alpha,sina,cosa,
     1                  sl,dsqrt,x1,x2,y1,y2,z1,z2,dsin,dcos
      m=mtype(id)
      if (mcurv(m).ne.0) go to 99
c          initialize routine
      i=nd(id,1)
      j=nd(id,2)
      x1=x(i)
      x2=x(j)
      y1=y(i)
      y2=y(j)
      z1=z(i)
      z2=z(j)
      sl=dsqrt((x2-x1)**2+(y2-y1)**2+(z2-z1)**2)
c          compute member direction cosines
      cx=(x2-x1)/sl
      cy=(y2-y1)/sl
      cz=(z2-z1)/sl
c          generate transformation matrix (t)
   14 do 20 ii=1,neq
      do 20 jj=1,neq
   20 t(ii,jj)=0.0
      go to (106,85,105,21), kind
c          three dimensional frame transformation
   21 alpha=alfa(id)/57.2957795
      alpha=alfa(id)/57.2957795
      sina=dsin(alpha)
      cosa=dcos(alpha)
      r(1,1)=cx
      r(1,2)=cy
      r(1,3)=cz
      if(cx**2+cz**2) 25,30,25
   30 if(y2-y1) 45,45,50
   45 cy=-1.
      go to 55
   50 cy=1.
   55 r(2,1)=-cy*cosa
      r(2,2)=0.0
      r(2,3)=sina
      r(3,1)=cy*sina
      r(3,2)=0.0
      r(3,3)=cosa
      go to 35
   25 rad=dsqrt(cx**2+cz**2)
      r(2,1)=(-cx*cy*cosa-cz*sina)/rad
```

```
      r(2,2)=rad*cosa
      r(2,3)=(-cy*cz*cosa+cx*sina)/rad
      r(3,1)=(cx*cy*sina-cz*cosa)/rad
      r(3,2)=rad*sina
      r(3,3)=(cy*cz*sina+cx*cosa)/rad
   35 do 40 k=1,3
      k1=k+3
      k2=k+6
      k3=k+9
      do 40 l=1,3
      t(k,l)=r(k,l)
      l1=l+3
      t(k1,l1)=r(k,l)
      l2=l+6
      t(k2,l2)=r(k,l)
      l3=l+9
      t(k3,l3)=r(k,l)
   40 continue
      go to 99
c          two dimensional frame transformation
   85 do 90 ii=1,nq
      do 90 jj=1,nq
   90 r(ii,jj)=0.0
      r(1,1)=cx
      r(1,2)=cy
      r(2,1)=-cy
      r(2,2)=cx
      r(3,3)=1.0
      do 95 k=1,nq
      k1=k+nq
      do 95 l=1,nq
      t(k,l)=r(k,l)
      l1=l+nq
      t(k1,l1)=r(k,l)
   95 continue
      go to 99
c          three dimensional truss transformation
  105 t(1,1)=cx
      t(1,2)=cy
      t(1,3)=cz
      t(2,4)=cx
      t(2,5)=cy
      t(2,6)=cz
      go to 99
c          two dimensional truss transformation
  106 t(1,1)=cx
      t(1,2)=cy
      t(2,3)=cx
      t(2,4)=cy
   99 go to (100,200,300), nrot
c          transform (S) to structure coordinates
  100 do 60 ii=1,nif
      do 60 jj=1,neq
      st(ii,jj)=0.0
```

```
      do 65 kk=1,nif
   65 st(ii,jj)=st(ii,jj)+s(ii,kk)*t(kk,jj)
   60 continue
      do 70 ii=1,neq
      do 70 jj=1,neq
      s(ii,jj)=0.0
      do 75 kk=1,nif
   75 s(ii,jj)=s(ii,jj)+t(kk,ii)*st(kk,jj)
   70 continue
      go to 80
c          transform loads to structure axes
  200 do 205 ii=1,neq
      tf(ii)=0.0
      do 110 jj=1,nif
  110 tf(ii)=tf(ii)+t(jj,ii)*p(jj)
  205 continue
      go to 80
c          transform forces to member axes
  300 do 305 ii=1,nif
      tf(ii)=0.0
      do 305 jj=1,neq
      tf(ii)=tf(ii)+t(ii,jj)*p(jj)
  305 continue
   80 continue
      return
      end
```

```
      subroutine saa (i,nwr)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/rrl/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/cross/ vv(50,50),aa(50,50),vz(50,50),az(50,50),av(50,50)
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      dimension aa1(50,50,2)
      double precision vv,aa,vz,az,av,aa1
      double precision rf,rz,rz1,rz12,zz,z12,z2
      open (unit=13,file='plotaa',status='new')
c        This subroutine evaluate aa(i,j)= E [ a a(T) ]
c        Calculate the next pre-computation
      do 200 i=1,numeq
      do 200 j=1,numeq
      aa1(i,j,1)=(zz(i,j,3)+4.*zz(i,j,2)+zz(i,j,1)-2.*z12(i,j,2)
     1 -2.*z12(j,i,2)+rf(i,j)+rf(j,i))
  200 continue
c        Calculate the present aa
      do 210 i=1,numeq
      do 210 j=1,numeq
      aa(i,j)=(aa1(i,j,2)-2.*z12(i,j,2)-2.*z12(j,i,2))/dt**4
  210 continue
      do 220 ii=1,numeq
      do 220 jj=1,numeq
  220 aa1(ii,jj,2)=aa1(ii,jj,1)
      if (i.gt.iftpt) return
      if (i-intpt) 405, 420, 430
  430 nwr=nwr+1
      if (nwr.eq.jpnt) go to 420
      return
  420 if (iplot.ne.0) go to 440
      write (6,3) i
      write (6,4)
      do 460 ii=inpt,ifpt,jpnp
  460 write (6,5) (aa(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
      return
  440 do 490 ii=inpt,ifpt,jpnp
  490 write (13,5) (aa(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
    3 format (/5x,'  NUMBER OF TIME STEP',3x,i5)
    4 format ('  aa(ii,jj) ')
    5 format (6e15.8)
  405 return
      end
```

```
      subroutine sav (i,nwr)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar.
      common/rrl/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rzl(50,50),rz12(50,50)
      common/cross/ vv(50,50),aa(50,50),vz(50,50),az(50,50),av(50,50)
      dimension aa2(50,50,2)
      double precision vv,aa,vz,az,av,aa2
      double precision rf,rz,rzl,rz12,zz,z12,z2
      open (unit=14,file='plotav',status='new')
c        This subroutine evaluate av(i,j)= E [ a v(T) ]
c        Calculate the next pre-computation
      do 200 i=1,numeq
      do 200 j=1,numeq
      aa2(i,j,1)=(zz(i,j,3)-zz(i,j,1)+2.*z12(i,j,2)
     1 -rf(i,j)+rf(j,i))
  200 continue
c        Calculate the present aa
      do 210 i=1,numeq
      do 210 j=1,numeq
      av(i,j)=(aa2(i,j,2)-2.*z12(j,i,2))/(2.*dt**3)
  210 continue
      do 220 i=1,numeq
      do 220 j=1,numeq
  220 aa2(i,j,2)=aa2(i,j,1)
      if (i.gt.iftpt) return
      if (i-intpt) 405, 420, 430
  430 nwr=nwr+1
      if (nwr.eq.jpnt) go to 420
      return
  420 if (iplot.ne.0) go to 440
      write (6,3) i
      write (6,4)
      do 460 ii=inpt,ifpt,jpnp
  460 write (6,5) (av(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
      return
  440 do 490 ii=inpt,ifpt,jpnp
  490 write (14,5) (av(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
    3 format (/5x,'  NUMBER OF TIME STEP',3x,i5)
    4 format ('  av(ii,jj) ')
    5 format (6e15.8)
  405 return
      end
```

```
      subroutine saz (i,nwr)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      common/rrl/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rzl(50,50),rz12(50,50)
      common/cross/ vv(50,50),aa(50,50),vz(50,50),az(50,50),av(50,50)
      dimension aa4(50,50,2)
      double precision vv,aa,vz,az,av,aa4
      double precision rf,rz,rzl,rz12,zz,z12,z2
      open (unit=15,file='plotaz',status='new')
c         This subroutine evaluate az(i,j)= E [ a z(T) ]
c         Calculate the next pre-computation
      do 200 i=1,numeq
      do 200 j=1,numeq
      aa4(i,j,1)=z12(j,i,2)-2.*zz(i,j,2)
  200 continue
c        Calculate the present aa
      do 210 i=1,numeq
      do 210 j=1,numeq
      az(i,j)=(aa4(i,j,2)+z12(i,j,2))/(dt**2)
  210 continue
      do 220 ii=1,numeq
      do 220 jj=1,numeq
  220 aa4(ii,jj,2)=aa4(ii,jj,1)
      if (i.gt.iftpt) return
      if (i-intpt) 405, 420, 430
  430 nwr=nwr+1
      if (nwr.eq.jpnt) go to 420
      return
  420 if (iplot.ne.0) go to 440
      write (6,3) i
      write (6,4)
      do 460 ii=inpt,ifpt,jpnp
  460 write (6,5)  (az(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
      return
  440 do 490 ii=inpt,ifpt,jpnp
  490 write (15,5) (az(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
    3 format (/5x,'  NUMBER OF TIME STEP',3x,i5)
    4 format ('  az(ii,jj) ')
    5 format (6e15.8)
  405 return
      end
```

```
      subroutine sbound (nbound)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1             nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common /rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),La2(50,50)
      common /eign/ eigvec(50,24),rho(24),a4(50,50)
      double precision a1,a2,a3,xLa,eigvec,rho,a4
      do 60 n=1,numnp
      do 60 kk=1,nq
      if (ir(n,kk).eq.0) goto 60
      k=nq*n-(nq-kk)
      do 70 ii=1,numeq
      if (ii.eq.k) go to 71
      a2(k,ii)=0.
      a2(ii,k)=0.
      go to 70
 71   a2(k,k)=1.
 70   continue
 60   continue
      if (nbound.eq.1) return
      do 160 n=1,numnp
      do 160 kk=1,nq
      if (ir(n,kk).eq.0) goto 160
      k=nq*n-(nq-kk)
      do 170 ii=1,numeq
      if (ii.eq.k) go to 171
      a1(k,ii)=0.
      a1(ii,k)=0.
      go to 170
 171  a1(k,k)=1.
 170  continue
 160  continue
      do 260 n=1,numnp
      do 260 kk=1,nq
      if (ir(n,kk).eq.0) goto 260
      k=nq*n-(nq-kk)
      do 270 ii=1,numeq
      if (ii.eq.k) go to 271
      a3(k,ii)=0.
      a3(ii,k)=0.
      go to 270
 271  a3(k,k)=1.
 270  continue
 260  continue
      do 360 n=1,numnp
      do 360 kk=1,nq
      if (ir(n,kk).eq.0) goto 360
      k=nq*n-(nq-kk)
      do 370 ii=1,numeq
      if (ii.eq.k) go to 371
      a4(k,ii)=0.
      a4(ii,k)=0.
      go to 370
 371  a4(k,k)=1.
```

```
370 continue
360 continue
    return
    end
```

```
      subroutine solver(isolve)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /slv/ a(50,25),b(50)
      double precision a,b,diag,air,sum,dsqrt
      nhw=mband-1
      n=numeq
      go to (100,300), isolve
  100 nred=0
      itrig=0
      lim=mband
  101 if(nred+1-n) 102,500,500
  102 nred=nred+1
      diag=a(nred,1)
      if (diag-1.0d-30) 601,601,110
  110 diag=dsqrt(diag)
c          go to 601 if matrix is singular or not positive definite
c          divide roe by square root of diagonal element
  111 do 113 j=1,lim
  113 a(nred,j)=a(nred,j)/diag
c          reduce remaining block of numbers
  201 do 251 i=1,nhw
      l=nred+i
      if(l-n) 211,211,251
  211 air=a(nred,i+1)
c          skip this row if multiplier air is zero
      if(air) 221,251,221
  221 do 231 j=i,nhw
      m=l+j-i
  231 a(l,m)=a(l,m)-air*a(nred,j+1)
  251 continue
      go to 101
  601 itrig=nred
  500 if(itrig) 600,610,600
c          singular matrix
  600 write(6,602) itrig
  602 format(1x,35hSingular Matrix at Reduction NRED =,i4)
  610 continue
      go to 700
c          reduce the right hand sides
  300 continue
      nred=0
  301 if(nred+1-n) 302,401,401
  302 nred=nred+1
c          divide row by square root of diagonal element
      b(nred)=b(nred)/a(nred,1)
c          reduce remaining block of numbers
      do 351 i=1,nhw
      l=nred+i
      if(l-n) 311,311,351
  311 b(l)=b(l)-a(nred,i+1)*b(nred)
  351 continue
      go to 301
c          back substitution
```

```
401 b(n)=b(n)/a(n,1)
    n1=n-1
    do 451 ii=1,n1
    i=n-ii
    sum=0.0
    do 421 jj=1,nhw
    m=jj+i
    if(n-m) 451,421,421
421 sum=sum+a(i,jj+1)*b(m)
451 b(i)=(b(i)-sum)/a(i,1)
 25 continue
700 continue
    return
    end
```

```
      subroutine sspace (nroot,nsmax)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/modal/ stk(16,16),stm(16,16),qk(16,16),xLam(16,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/rr3/ a1(50,50),a2(50,50),a3(50,50),xLa(50,50),la2(50,50)
      common/xmd/ xmass(50),damp(50,25)
      common/eign/ eigvec(50,24),rho(24),a4(50,50)
      common/matrix/ La(50),t1(50),t2(50),f(50)
      common/slv/ a(50,25),b(50)
      double precision rf,rz,rz1,rz12,a,b,xmass,damp,t1,t2,f
      double precision a1,a2,a3,xLa,stk,stm,qk,xLam,eigvec,rho,a4
c      This subroutine evaluate the eigenvalue and eigenvector
c      by using subspace iteration, the trial vector is rz12(i,j),
c      the object matrices is K* and M* and the output eivalues are
c      rho(i),i=1,nroot,the output eigenvectors are eigvec(i,j),j=1,nroot
c      Determine the order of iteration vector
      read (5,*) trix
      nord=nroot
      rz(1,1)=1.0 E+20
      tole=1.0 E-06
      nit=0
      do 110 ii=1,numeq
      do 110 jj=2,nord
 110  rz12(ii,jj)=trix
      do 115 ii=1,numeq
      do 115 jj=1,numeq
      a1(ii,jj+ii-1)=a(ii,jj)
      rf(ii,jj+ii-1)=a(ii,jj)
 115  continue
      do 116 ii=1,numeq
      do 116 jj=1,numeq
      if (ii.eq.jj) go to 116
      rf(jj,ii)=rf(ii,jj)
      a1(jj,ii)=a1(ii,jj)
 116  continue
c        Determine the starting iteration vector
      do 120 ii=1,numeq
 120  rz12(ii,1)=xmass(ii)
c        ordering the a(ii,1)/xmass(ii)
      do 220 ii=1,numeq
      s=a(ii,1)/xmass(ii)
      call order (ii,s,numeq)
 220  continue
      do 225 ii=1,nord-1
      n2=La(ii)
 225  rz12(n2,ii+1)=rz(ii,1)
c          Decomposing a1 = [ K ]
 145  call gauss (1,nord)
 135  call gauss (2,nord)
c          Form K* amd M*
      do 140 ii=1,numeq
      call range (ii,numeq,mband,mg,mbd)
      do 140 jj=1,nord
```

```
      xLa(ii,jj)=0.
      do 140 kk=mg,mbd
  140 xLa(ii,jj)=xLa(ii,jj)+rf(ii,kk)*rz12(kk,jj)
      do 150 ii=1,nord
      do 150 jj=ii,nord
      stk(ii,jj)=0.
      do 150 kk=1,numeq
  150 stk(ii,jj)=stk(ii,jj)+rz12(kk,ii)*xLa(kk,jj)
      do 160 ii=1,nord
      do 160 jj=ii,nord
      stm(ii,jj)=0.
      do 160 kk=1,numeq
      stm(ii,jj)=stm(ii,jj)+rz12(kk,ii)*rz12(kk,jj)*xmass(kk)
  160 continue
c        Symmetric
      do 165 ii=1,nord
      do 165 jj=ii,nord
      if (ii.eq.jj) go to 165
      stm(jj,ii)=stm(ii,jj)
      stk(jj,ii)=stk(ii,jj)
  165 continue
c        Find the eigenpairs for K* and M*
      nit=nit+1
      if (nroot.gt.2) go to 170
      call eigen
      go to 180
  170 call jacobi(nord,nsmax)
c        Check for convergence
  180 if (nit.eq.1) go to 190
      do 250 ii=1,nord
      er1=abs(1.-xLam(ii,2)/xLam(ii,1))
      if (er1.gt.tole) go to 190
  250 continue
      go to 200
  190 do 195 ii=1,nord
  195 xLam(ii,2)=xLam(ii,1)
      do 210 ii=1,numeq
      do 210 jj=1,nord
      xLa(ii,jj)=0.
      do 210 kk=1,nord
  210 xLa(ii,jj)=xLa(ii,jj)+rz12(ii,kk)*qk(kk,jj)
      do 215 ii=1,numeq
      do 215 jj=1,nord
  215 rz12(ii,jj)=xLa(ii,jj)*xmass(ii)
      go to 135
  200 do 230 ii=1,numeq
      do 230 jj=1,nroot
      eigvec(ii,jj)=0.
      do 230 kk=1,nroot
  230 eigvec(ii,jj)=eigvec(ii,jj)+rz12(ii,kk)*qk(kk,jj)
      do 240 ii=1,nroot
  240 rho(ii)=xLam(ii,1)
      write (6,1) nit
    1 format (3x,'NUMBER OF ITERATION IN SSPACE',2x,i5)
```

```
return
end
```

```
      subroutine stiff
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/slv/ a(50,25),b(50)
      common/xmd/ xmass(50),damp(50,25)
      common/mem/ mtype(50),ij(50,2),mid(50),mlc(50),alfa(50),
     1            mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      double precision a,b,xmass,damp
      do 5 id=1,numem
      i=ij(id,1)
      j=ij(id,2)
      n=id
      m=mtype(id)
      go to (15,20,15,30), kind
   15 call strut  (n,i,j)
      go to 10
   20 if (mcurv(m).ne.0) call curvbm (n,i,j)
      if (mcurv(m).ne.0) go to 10
      call beam2 (n,i,j)
      go to 10
   30 call beam (n,i,j)
   10 continue
c          merge element stiffness matrix
      call merge (i,j,1)
    5 continue
      return
      end
```

```
      subroutine strut (id,i,j)
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1           nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common /nod/ x(25),y(25),z(25),ntype(25),ir(25,6),ui(25,6)
      common/mem/ mtype(50),nd(50,2),mid(50),mlc(50),alfa(50),
     1           mrel(50,2),thetai(50),thetaj(50)
      common/mlib/ xa(25),zi(25),yi(25),xj(25),avy(25),
     1            avz(25),mcurv(25)
      common/mat/ e(5),sigma(5),epsiln(5),pr(5),g(5),rho(5)
      common /stif/ s(12,12),r(3,3),t(12,12),st(12,12),tf(12),p(12)
      common /xmss/ xms(12)
      double precision s,r,t,st,tf,sl,sl,dsqrt,xp,yp,zp,xms
      i=nd(id,1)
      j=nd(id,2)
      xp=x(j)-x(i)
      yp=y(j)-y(i)
      zp=z(j)-z(i)
      sl=dsqrt(xp**2+yp**2+zp**2)
c       generate element stiffness matrix (s)
      m=mid(id)
      n=mtype(id)
      ym=e(m)
      area=xa(n)
      rh=rho(m)
      sl=xa(n)*ym/sl
      s(1,1)=sl
      s(1,2)=-sl
      s(2,1)=-sl
      s(2,2)=sl
c       transform (s) to global coordinates
      call rotate (1,id)
c       Form mass matrix
      go to (99,101), method
 101  do 100 ii=1,neq
      xms(ii)=0.
 100  continue
      cl=rh*area*sl/2.
      xms(1)=cl
      xms(2)=cl
  99  return
      end
```

```
      subroutine svv (i,nwr)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/cross/ vv(50,50),aa(50,50),vz(50,50),az(50,50),av(50,50)
      double precision zz,z12,z2,rf,rz,rz1,rz12
      double precision vv,aa,vz,az,av
      open (unit=16,file='plotvv',status='new')
c        This subroutine evaluate  vv(i,j)= E [v v(T) ]
      do 270 ii=1,numeq
      do 270 jj=1,numeq
      vv(ii,jj)=(zz(ii,jj,3)+zz(ii,jj,1)-rf(ii,jj)-rf(jj,ii))/
     1          (4.*dt**2)
  270 continue
      if (i.gt.iftpt) return
      if (i-intpt) 405, 420, 430
  430 nwr=nwr+1
      if (nwr.eq.jpnt) go to 420
      return
  420 if (iplot.ne.0) go to 440
      write (6,3) i
      write (6,4)
      do 460 ii=inpt,ifpt,jpnp
  460 write (6,5) (vv(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
      return
  440 do 490 ii=inpt,ifpt,jpnp
  490 write (16,5) (vv(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
    3 format (/5x,'  NUMBER OF TIME STEP',3x,i5)
    4 format ('  vv(ii,jj) ')
    5 format (6e15.8)
  405 return
      end
```

```
      subroutine svz (i,nwr)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/dlod/ dt,nti,ns(50,10),dl(50,10),ndof(10),ndofl
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/rr2/ rf(50,50),rz(50,50),rz1(50,50),rz12(50,50)
      common/cross/ vv(50,50),aa(50,50),vz(50,50),az(50,50),av(50,50)
      dimension aa3(50,50,2)
      double precision vv,aa,vz,az,av,aa3
      double precision rf,rz,rz1,rz12,zz,z12,z2
      open (unit=17,file='plotvz',status='new')
c        This subroutine evaluate vz(i,j)= E [ v z(T) ]
c        Calculate the next pre-computation
      do 200 i=1,numeq
      do 200 j=1,numeq
      aa3(i,j,1)=-z12(j,i,2)
  200 continue
c        Calculate the present aa
      do 210 i=1,numeq
      do 210 j=1,numeq
      vz(i,j)=(aa3(i,j,2)+z12(i,j,2))/(2.*dt)
  210 continue
      do 220 i=1,numeq
      do 220 j=1,numeq
  220 aa3(i,j,2)=aa3(i,j,1)
      if (i.gt.iftpt) return
      if (i-intpt) 405, 420, 430
  430 nwr=nwr+1
      if (nwr.eq.jpnt) go to 420
      return
  420 if (iplot.ne.0) go to 440
      write (6,3) i
      write (6,4)
      do 460 ii=inpt,ifpt,jpnp
  460 write (6,5) (vz(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
      return
  440 do 490 ii=inpt,ifpt,jpnp
  490 write (17,5) (vz(ii,jj),jj=inpt,ifpt,jpnp)
      nwr=0
    3 format (/5x,'  NUMBER OF TIME STEP',3x,i5)
    4 format ('  vz(ii,jj) ')
    5 format (6e15.8)
  405 return
      end
```

```
      subroutine write(i,nwr)
      integer wstif,wrstif,wlod
      common/par/ method,kind,numnp,nstrut,nbeam,numem,nrnp,nmrel,mband,
     1            nlnd,nlmem,neq,nq,numeq,nlc,nif,wstif,wrstif,wlod
      common/random/ irandm,ncvdf,ntacv,n1(10),n2(10),ax(50,50,10),
     1                nt(50,50,10),lon,nroot,nsmax,corri
      common/sol/ xdp(50,3),x1d(50,2),x2d(50,3)
      common/matrix/ la(50),t1(50),t2(50),f(50)
      common/rest/ res(50)
      common/kuz/ rx(50,50,3,8),ck(8)
      common/rr1/ zz(50,50,3),z12(50,50,2),z2(50,50,2)
      common/flagw/ inpt,ifpt,jpnp,intpt,iftpt,jpnt,iplot,meanw,iwvar
      double precision xdp,x1d,x2d,t1,t2,f,zz,z12,z2,rx,ck,res
  400 if (i.gt.iftpt) go to 410
      if (i-intpt) 410, 420, 430
  430 nwr=nwr+1
      if (nwr.eq.jpnt) go to 420
      go to 410
  420 if (iplot.ne.0) go to 440
      if (meanw.eq.0) go to 450
      write (6,3) i
      if (lon.eq.2) go to 7
      write (6,4)
      do 460 ii=inpt,ifpt,jpnp
      write (6,5) ii, xdp(ii,3),x1d(ii,1),x2d(ii,1),f(ii)
  460 continue
      go to 450
    7 write (6,1)
      do 6 ii=inpt,ifpt,jpnp
    6 write (6,5) ii, xdp(ii,3),x1d(ii,1),x2d(ii,1),res(ii)
  450 go to (470,480), irandm
  480 if (iwvar.eq.0) go to 470
      do 490 ii=inpt,ifpt,jpnp
      write (6,9) (zz(ii,jj,3),jj=inpt,ifpt,jpnp)
  490 continue
      go to 470
  440 if (meanw.eq.0) go to 500
      if (lon.eq.2) go to 540
      do 550 ii=inpt,ifpt,jpnp
      write (10,15) xdp(ii,3),f(ii)
  550 continue
      go to 500
  540 do 560 ii=inpt,ifpt,jpnp
      write (10,15) xdp(ii,3),res(ii)
  560 continue
  500 go to (470,510), irandm
  510 if (iwvar.eq.0) go to 470
      do 520 ii=inpt,ifpt,jpnp
      write (11,9)  (zz(ii,jj,3),jj=inpt,ifpt,jpnp)
  520 continue
  470 nwr=0
c         Move 3rd and 2nd points to 2nd and 1st point
  410 do 399 ii=1,numeq
      xdp(ii,1)=xdp(ii,2)
```

```
      xdp(ii,2)=xdp(ii,3)
  399 continue
      if(irandm.eq.1) return
      do 380 ii=1,numeq
      dc 380 jj=1,numeq
      zz(ii,jj,1)=zz(ii,jj,2)
      zz(ii,jj,2)=zz(ii,jj,3)
      z2(ii,jj,1)=z2(ii,jj,2)
      z12(ii,jj,1)=z12(ii,jj,2)
  380 continue
      do 385 ii=1,numeq
      do 385 jj=1,numeq
      do 385 np=1,nroot
      rx(ii,jj,1,np)=rx(ii,jj,2,np)
  385 rx(ii,jj,2,np)=rx(ii,jj,3,np)
    1 format (/'# DOF ',3x,'  DISPLACEMENT ',4x,'  VELOCITY',8x,
     1  ' ACCELERATION',4x,' RESTORING FORCE')
    3 format (///5x,'  NUMBER OF TIME STEP',3x,i5)
    4 format (/'# DOF ',3x,'  DISPLACEMENT ',4x,'  VELOCITY',8x,
     1  ' ACCELERATION',4x,' FORCING FUNCTION')
    5 format (i5,4x,e15.8,4x,e15.8,2x,e15.8,4x,e13.6)
    9 format (6e13.5)
   15 fcrmat (2e15.6)
      return
      end
```

END

5-87

DTIC