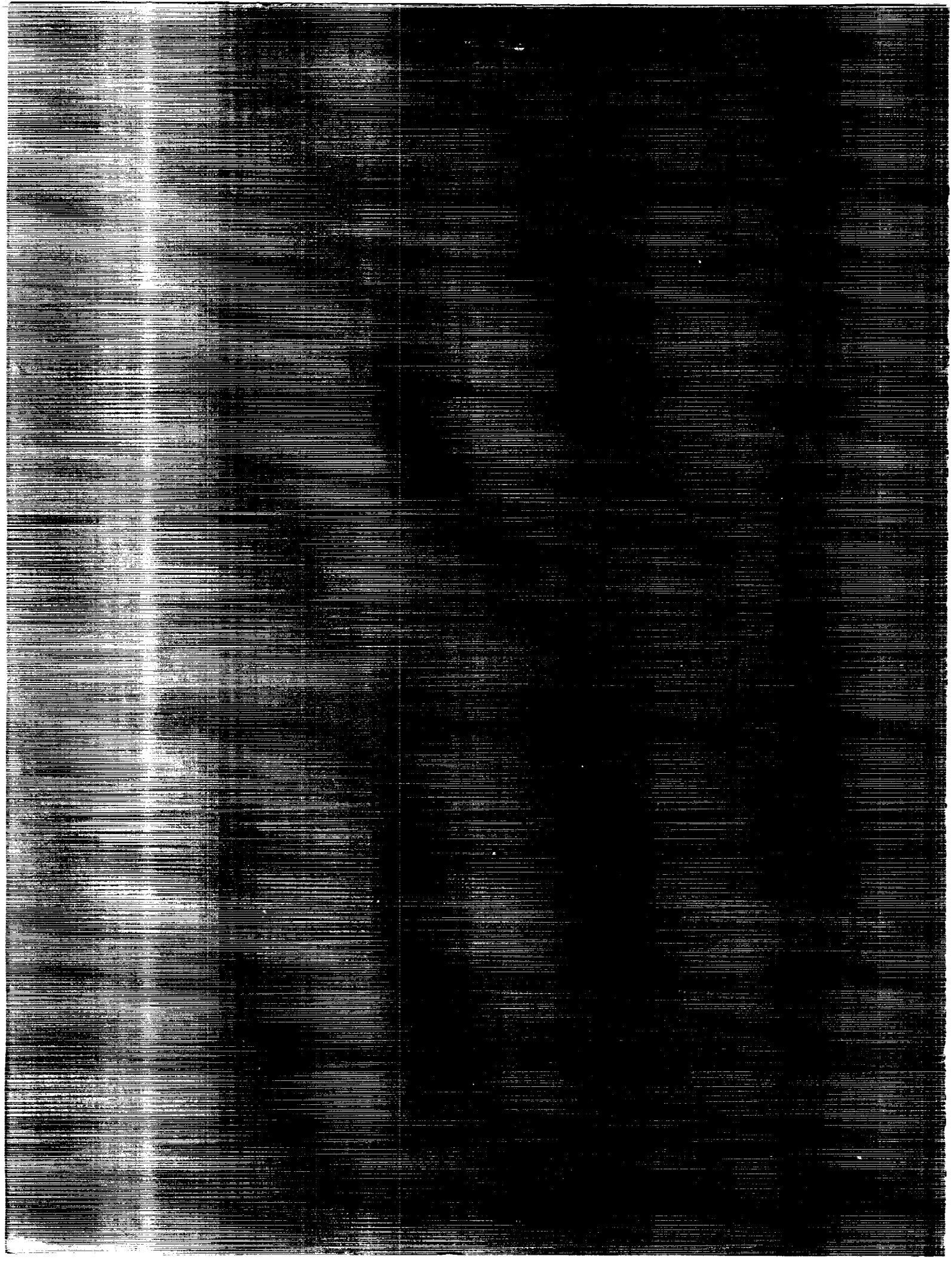
NASA Technical Memorandum 4156

Computational Aspects of Sensitivity Calculations in Linear Transient Structural Analysis

William H. Greene

MAY 1990



NASA Technical Memorandum 4156

Computational Aspects of Sensitivity Calculations in Linear Transient Structural Analysis

William H. Greene
Langley Research Center
Hampton, Virginia

NASA
National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

1990

Contents

List of Figures

List of Tables

List of Symbols

$\mathbf{C}$	system damping matrix
$\overline{\mathbf{C}}$	reduced damping matrix
C_x	cost in number of floating point operations for computational task x
E	Young's modulus
$\mathbf{f}$	vector of external forces
$\overline{\mathbf{f}}$	vector of reduced system forces
$g(t)$	scalar function representing time dependence of applied loading
g_i	ith constraint
h_i	thickness in ith span or section for five-span beam and stepped beam examples
$\mathbf{K}$	system stiffness matrix
$\overline{\mathbf{K}}$	reduced stiffness matrix
l_i	distance from beam root to end of ith section in stepped beam
$\mathbf{M}$	system mass matrix
$\overline{\mathbf{M}}$	reduced mass matrix
M	bending moment
m	mass design variable in beam example
$\mathbf{N}_{ij}$	submatrix i, j in matrix series expansion method
n_c	number of critical time points
n_g	number of degrees of freedom in finite element model
n_p	total number of physical response quantities to be calculated as function of time
n_r	number of equations in reduced system
n_t	number of time steps in numerical integration of differential equations
$\mathbf{q}$	vector of reduced system coordinates
$\mathbf{S}$	transformation matrix between element stresses and nodal displacements
t	time
t_c	critical time in critical point constraint
t_θ^i	thickness of θ degree lamina in ith region of skin for delta wing example
t_w^i	thickness of web in ith region for delta wing example
$\mathbf{u}$	vector of displacements
u_i	displacement at one-third point in ith span of five-span beam example
u_{allow}	maximum allowable value of displacement

u_{tip}	deflection at beam tip
V_i	shear force in ith span
$\mathbf{W}_{ij}$	submatrix i, j in matrix series expansion method
$\mathbf{x}$	vector of design variables
β	semibandwidth (excluding diagonal) of system matrices (e.g., $\mathbf{K}$, $\mathbf{M}$)
Δt	time step size in matrix series expansion integration technique
ρ	material density
$\boldsymbol{\sigma}$	vector of elemental stresses
σ_θ^i	fiber direction stress in θ degree lamina in ith region of skin for delta wing example
σ_{root}	stress at beam root
τ_w^i	shear stress in ith region of web for delta wing example
$\boldsymbol{\Phi}$	matrix with each column being basis vector
ω_j	jth vibration frequency

A dot over a symbol indicates derivative with respect to time. A superscript T indicates a transposed matrix.

Summary

A study has been performed focusing on the calculation of sensitivities of displacements, velocities, accelerations, and stresses in linear, structural, transient response problems. One significant goal of the study was to develop and evaluate sensitivity calculation techniques suitable for large-order finite element analyses. Accordingly, approximation vectors such as vibration mode shapes are used to reduce the dimensionality of the finite element model. Much of the research focused on the convergence of both response quantities and sensitivities as a function of the number of vectors used.

Two types of sensitivity calculation techniques were developed and evaluated. The first type of technique is an overall finite difference method where the analysis is repeated for perturbed designs. The second type of technique is termed semianalytical because it involves direct analytical differentiation of the equations of motion with finite difference approximation of the coefficient matrices. To be computationally practical in large-order problems, the overall finite difference methods must use the approximation vectors from the original design in the analyses of the perturbed models. This was found to result in poor convergence of stress sensitivities in several cases. To overcome this poor convergence, two semianalytical techniques were developed. The first technique accounts for the change in eigenvectors through approximate eigenvector derivatives. The second technique applies the mode acceleration method of transient analysis to the sensitivity calculations. Both result in very good convergence of the stress sensitivities. In both techniques the computational cost is much less than would result if the vibration modes were recalculated and then used in an overall finite difference method.

Chapter 1

Introduction

1.1. Overview

In the past 10 years there has been increasing interest in calculating the derivatives of structural behavior with respect to problem parameters or design variables (i.e., sensitivities). One of the main uses of these sensitivities is in automated design procedures where a numerical algorithm is used to improve a structure by modifying the design parameters while satisfying prescribed constraints on the structural behavior. Most of the numerical algorithms used in these procedures require both an initial design and a set of sensitivities in order to decide how to improve the structure. Many references address this sensitivity calculation question within the context of automated structural design, whereas others, such as this study, focus specifically on issues related to the calculation of sensitivities. Other uses of sensitivities in structures problems include the system identification problem in structural dynamics and statistical structural analysis. References 1 and 2 provide a comprehensive review of work on calculating sensitivities in structural systems.

It is clear from many references (e.g., ref. 1) that most of the emphasis in structural optimization and the associated sensitivity calculation methods has been on static problems. This is not surprising since most structural analyses themselves are static. The objective of the static analysis and sensitivity calculation problem, for linear systems, is to calculate the responses (e.g., displacements, stresses) and their derivatives with respect to structural parameters (e.g., member areas, thicknesses), which are assumed to be constant for all time. Techniques for both the analysis and sensitivity calculations have reached considerable maturity in the past 10 to 20 years.

In many problems, however, the loading on the structure varies with time, which causes the response of the structure also to vary as a function of time. Examples of such problems are a gust on an aircraft wing, an unbalanced engine in an automobile, or a building during an earthquake. In these cases, it is important to predict stresses accurately as well as displacements (and possibly velocities and accelerations) as a function of time. Often it is sufficient to predict the maximum and minimum values of these response quantities. Similarly, the goal of the sensitivity analysis is the calculation of derivatives of these response quantities with respect to the structural parameters as a function of time or at the time points where the maximum or minimum responses occur.

The introduction of the time parameter complicates the analysis in several ways. First, it changes the system of equations from a set of coupled algebraic equations to a set of coupled differential equations whose accurate solution may be difficult and computationally costly. Second, the amount of information that must be considered and evaluated to understand the response of the structure is increased by orders of magnitude.

Most practical static and dynamic analyses are currently performed with the finite element method. Since this technique replaces a continuum (infinite dimensional space) with a finite-degree-of-freedom approximation, the question of required mesh refinement is a natural one. This is not an easy question to answer because the convergence of the approximation as the mesh is refined depends on the quantity being considered. Usually, the fundamental unknowns are the displacements and rotations at the finite element nodes. In theses cases, the convergence of derivatives of displacements with respect to a spatial parameter (stresses), with respect to time (velocities, accelerations), or with respect to a structural parameter (sensitivities) will be worse than the convergence of the displacements themselves.

After the structure has been discretized with the finite element method, yet another approximation is usually introduced in linear dynamics problems. The behavior of the structure is represented by a reduced set of basis functions (frequently natural vibration modes) in order to simplify the solution of the transient response problem. This approximation introduces another set of concerns over accuracy of the response quantities and their sensitivities.

Other errors in transient analysis or sensitivity calculations, which rarely occur in static analyses, are due to the truncation error of finite difference operators. This problem occurs with the use of numerical integration techniques in solving the coupled differential equations in the transient problem. This problem also occurs when difference approximations are used in the calculation of sensitivities. Round-off errors, due to the finite precision arithmetic on digital computers, are also more of a concern in transient or sensitivity analyses than in simple static analyses.

All these complexities in transient analysis coupled with the problems of sensitivity analysis have slowed the progress in the development of sensitivity calculation techniques for transient response problems. However, substantial progress has been made. Some of the important previous work in optimization of structures under transient loads and calculation of sensitivities in transient response problems is discussed in section 1.2.

1.2. Review of Previous Pertinent Work

Reference 3 is one of the earliest papers dealing with optimization of structures under transient loads. In this paper, Fox and Kapoor consider the minimum mass design of frame structures under an applied base motion subject to constraints on deflections and stresses. The equations of motion are uncoupled by using vibration modes and solved for the maximum value of the modal response by using a shock spectrum approach. A considerable simplification is introduced by directly summing the maximum modal responses; therefore, time is removed as a parameter in the calculations.

In references 4 and 5, Cassis and Schmit present procedures for the automated design of plane frames under general transient loading. The dynamic analysis is performed with modal superposition, and only modal damping is allowed. Integral forms of the time-dependent constraints are used. Sensitivities are calculated with an explicit differentiation of the dynamic equations along with exact calculation of the required eigenvalue and eigenvector derivatives. Effects of finite element discretization and modal truncation on the sensitivities or final optimized designs were not considered.

In the past 10 years, other researchers have considered the application of general sensitivity theory to the problem of dynamic mechanical systems. Reference 1 summarizes this work and describes three basic approaches which have been employed. In the first method, called the direct method, the equations of motion are directly differentiated and solved. A second method offers the advantage of reduced computational cost when there are more design variables than constraints on response quantities. In this method, called the adjoint method, the sensitivity equations are rewritten in terms of a newly defined adjoint vector. After solving this new system for the adjoint vector, the calculation of the sensitivities of the response constraints with respect to each of the design variables is straightforward. In the third method, called the Green's function method, the derivatives are obtained in terms of the Green's function of the equations of motion. Although the results from all three methods are theoretically identical, their relative computational efficiency depends on the relative numbers of design variables, degrees of freedom, and constraints.

Haug, Arora, and their coworkers have made considerable progress in addressing many of the problems in the optimal design of mechanical systems under dynamic loadings. Much of their early work was spent studying a "state space" or adjoint variable approach to calculating sensitivities. References 6 through 9 should be noted. These references consider application to both elastic structural design and machine design problems that often have the additional complexity of nonlinear equations of motion. However, most of these examples have involved few degrees of freedom or design variables. A more recent paper by Haug (ref. 10) extended the sensitivity analyses of previous papers to include additional algebraic constraint equations that are often present in machine design problems. Also, sensitivity equations for second derivatives are presented.

The adjoint method is particularly attractive when a transient constraint is integrated over time to produce a single constraint because the total number of constraints is often small. However, the loss of information in this integral formulation and its disadvantages are noted in reference 11. Given the danger of having only a single "worst-case" value of the constraint function in time, reference 11 proposed including all local maximum points of the constraint function in the constraint set. A significant disadvantage of this approach is that for "jagged" response functions, there can be a large number of redundant local maxima. This important problem of constraint definition was also considered in reference 8, where several methods for obtaining a few important constraints at discrete points in time were proposed.

Both direct and adjoint sensitivity methods for a nonlinear hysteretic structure are presented in reference 12. Because of the nonlinearities, numerical integration of the full coupled system is required.

A recent approach in sensitivity analysis has been to write sensitivity expressions for the solid continuum prior to discretizing the system. This approach is especially attractive when shape-type design variables are being considered because the design variable itself often represents a continuous region on the surface of the body. Reference 13 uses the concept of the material derivative to calculate shape derivatives of a continuum under dynamic loads. In reference 14, expressions for shape sensitivities of a continuum considering material nonlinearities and dynamic effects are written with a variational approach.

1.3. Objectives and Scope

The purpose of the study reported herein is to investigate methods for calculating sensitivities in linear transient structural response problems. Very general forms of external loading on the structure and damping are permitted. In any numerical algorithm, both accuracy and computational efficiency are concerns. Errors in the sensitivities due to factors such as the finite element mesh, truncation of the basis vector set in the transient analysis, and finite difference approximations in the sensitivity and numerical integration procedures are considered. An objective of the study is to identify approaches to sensitivity analysis that are appropriate for large-scale structural analysis. This is emphasized in the selection of the algorithms and in a study of the relative computational efficiency of several competing methods.

Three transient response problems are considered in detail: a five-span, simply supported beam; a composite aircraft wing; and a cantilever beam with a cross section that varies along its length. None of these three problems are large. However, each problem includes ingredients which make the sensitivity analysis computationally difficult.

Chapter 2

Equations of Motion and Solution

2.1. Governing Equations

The equations of motion for a damped, linear structural system can be written as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{p}(t) \tag{2.1}$$

which is a set of n_g coupled differential equations and $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are the system mass, damping, and stiffness matrices, respectively. Frequently it is possible to separate the loading vector $\mathbf{p}$ into a product of a vector describing the spatial distribution of the loading $\mathbf{f}$ and a scalar function of time $g(t)$ as

$$\mathbf{p}(t) = g(t)\mathbf{f} \tag{2.2}$$

Often equations (2.1) are the result of a large finite element model and are therefore of large order. One way to characterize the behavior of this system is by examining the eigenvalues of the undamped system

$$\mathbf{K}\boldsymbol{\phi}_j - \omega_j^2\mathbf{M}\boldsymbol{\phi}_j = 0, \qquad (j = 1, \ldots, n_g) \tag{2.3}$$

For most large structural systems, equations (2.1) are "stiff"; the condition number $\omega_{n_g}^2/\omega_1^2$ is many orders of magnitude.

The external loading also has a major effect on the dynamic response of the system. Impulsive loads where $g(t)$ changes rapidly relative to the periods associated with the smallest ω_j tend to produce a response history with significant high-frequency components. Loads that are applied slowly relative to the vibration periods of the ω_j produce a predominantly low-frequency response history.

Two basic approaches are available for the solution of equations (2.1). The first approach is to numerically integrate the equations in a step-by-step manner. In implicit integration techniques, the time step must be a fraction of the period associated with the largest ω_j significantly excited by the loading in order to obtain an accurate solution. The well-known Newmark method (for example, ref. 15) is an example of such an integration technique. In explicit integration techniques, the time step must be a fraction of the period associated with ω_{n_g} in order for the solution process to be numerically stable. Using either technique, the computational work is large because equations (2.1) are of large order.

An alternative to directly solving equations (2.1) is to solve an approximate reduced-order problem instead. This is the preferred approach for most linear structural dynamics problems. The details of the techniques used to reduce the order of the dynamic system are discussed in section 2.2.

2.2. Reduction Techniques

The first step in applying a reduction technique to the solution of equations (2.1) is to approximate the solution by n_r basis functions

$$\mathbf{u} = \boldsymbol{\Phi}\mathbf{q} \tag{2.4}$$

where n_r is usually much less than n_g. Then a reduced set of equations can be written

$$\overline{\mathbf{M}}\ddot{\mathbf{q}} + \overline{\mathbf{C}}\dot{\mathbf{q}} + \overline{\mathbf{K}}\mathbf{q} = g(t)\bar{\mathbf{f}} \tag{2.5}$$

where

$$\overline{\mathbf{M}} = \boldsymbol{\Phi}^T\mathbf{M}\boldsymbol{\Phi} \tag{2.6}$$

$$\overline{\mathbf{C}} = \boldsymbol{\Phi}^T\mathbf{C}\boldsymbol{\Phi} \tag{2.7}$$

$$\overline{\mathbf{K}} = \boldsymbol{\Phi}^T\mathbf{K}\boldsymbol{\Phi} \tag{2.8}$$

$$\bar{\mathbf{f}} = \boldsymbol{\Phi}^T\mathbf{f} \tag{2.9}$$

If the number of vectors in $\boldsymbol{\Phi}$ is equal to the size of the original system n_g and the vectors in $\boldsymbol{\Phi}$ are linearly independent, the transformation of equation (2.4) is exact. Usually, though, $n_r << n_g$ and the solution to the full system (eqs. (2.1)) is only approximated by the solution to the reduced system (eqs. (2.5)). The quality of this approximation as the number of vectors in $\boldsymbol{\Phi}$ is increased is a key concern in evaluating the effectiveness of a particular reduction technique.

In all reduction methods considered herein, the first n_r vectors of the set are taken as the reduced basis. Alternate approaches are available for assessing the importance of a given vector prior to solution of the reduced system and then discarding the vector if its contribution is insignificant. These approaches are not considered here because the cost of generating the set of vectors $\boldsymbol{\Phi}$ is often high and the cost of solving equations (2.5) is often fairly low.

2.2.1. Mode Displacement Method

The most widely used reduction technique is the traditional mode displacement method. In this method, equations (2.3) are solved for the set of vibration modes with lowest n_r frequencies and modes. This set of vibration modes is used as the set of basis functions $\mathbf{\Phi}$. When the system is undamped ($\mathbf{C} = 0$ in eqs. (2.1)) or $\mathbf{C}$ can be expressed as a linear combination of $\mathbf{M}$ and $\mathbf{K}$, equations (2.5) represent a set of uncoupled differential equations which can be solved independently. If the eigenvectors are scaled so that $\phi_i^T \mathbf{M} \phi_i = 1$, the uncoupled equations can be written as

$$\ddot{q}_i + 2\xi_i \omega_i \dot{q}_i + \omega_i^2 q_i = g(t) \bar{f}_i \qquad (i = 1, \ldots, n_r) \tag{2.10}$$

where ξ_i is the modal damping ratio. For certain forms of external loading, such as $g(t)$ represented as a piecewise linear function of time, an exact explicit solution is available. This approach is described in reference 16 and is used in the NASTRAN®computer program (ref. 17).

Equations (2.1) are the result of a given finite element approximation designed to model the behavior of the dynamic system. The goal of the reduction methods discussed in this section is to achieve an accurate approximation to the solution of equations (2.1) with a small number of basis vectors. As discussed, the vibration modes are the most commonly used basis functions in linear structural dynamics. There are two cases, however, where a large number of modes are required for an accurate solution of equations (2.1), and therefore the performance of the mode displacement method is poor. In the first case, if the structure is loaded in an impulsive manner, many high-frequency modes tend to be excited. These high-frequency modes must be included in the analysis since their contribution to the total response is significant. In the second case, if the response of the structure contains a large static component, the linear combination of vibration modes can do a poor job of approximating the static deflection shape. The reduction methods discussed in sections 2.2.2, 2.2.3, and 2.2.4 alleviate this second accuracy problem with the mode displacement method.

2.2.2. Mode Acceleration Method

To alleviate the poor accuracy of the mode displacement method due to its poor representation of the static component in the response, a method was proposed by Williams and Jones (ref. 18) called the mode acceleration method, which is described in its modern computational forms in references 16 and 19. The mode acceleration method can be derived by rewriting equations (2.1) as

$$\mathbf{u}(t) = g(t)\mathbf{K}^{-1}\mathbf{f} - \mathbf{K}^{-1}\mathbf{C}\dot{\mathbf{u}} - \mathbf{K}^{-1}\mathbf{M}\ddot{\mathbf{u}} \tag{2.11}$$

The first term in equations (2.11) is the quasi-static solution with load amplitude determined by $g(t)$. This term is calculated by solving the equations

$$\mathbf{K}\mathbf{u}_s = \mathbf{f} \tag{2.12}$$

This solution is carried out in the standard way by first factoring $\mathbf{K}$ into a product of upper and lower triangular matrices and then performing a forward and backward substitution operation to obtain $\mathbf{u}_s$. The other two terms are calculated with the solution for $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ from the mode displacement solution. In these terms $\mathbf{K}^{-1}$ is calculated as follows. Equations (2.3) can be rewritten in matrix form as

$$\mathbf{K}\mathbf{\Phi}\mathbf{\Omega}^{-2} = \mathbf{M}\mathbf{\Phi} \tag{2.13}$$

where here $\mathbf{\Phi}$ is the full set of n_g eigenvectors and $\mathbf{\Omega^{-2}}$ is

$$\mathbf{\Omega}^{-2} = \begin{bmatrix} \frac{1}{\omega_1^2} & & & \\ & \frac{1}{\omega_2^2} & & \\ & & \ddots & \\ & & & \frac{1}{\omega_{n_r}^2} \end{bmatrix} \tag{2.14}$$

With the eigenvectors scaled so that

$$\mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} = \mathbf{I} \tag{2.15}$$

equation (2.13) can be written as

$$\mathbf{\Phi}^T \mathbf{K} \mathbf{\Phi} \mathbf{\Omega^{-2}} = \mathbf{I} \tag{2.16}$$

Premultiplying by $(\mathbf{\Phi}^T)^{-1}$ and postmultiplying by $\mathbf{\Phi}^T$ yields

$$\mathbf{K}\mathbf{\Phi}\mathbf{\Omega^{-2}}\mathbf{\Phi}^T = \mathbf{I} \tag{2.17}$$

or

$$\mathbf{K}^{-1} = \mathbf{\Phi}\mathbf{\Omega^{-2}}\mathbf{\Phi}^T \tag{2.18}$$

When $\mathbf{\Phi}$ contains less than the full n_g eigenvectors, this expression for $\mathbf{K}^{-1}$ is only approximate. However, since $\ddot{\mathbf{u}}$ is obtained from the mode displacement solution based on n_r modes ($\mathbf{\Phi}\ddot{\mathbf{q}}$), $\mathbf{K}^{-1}\mathbf{M}\ddot{\mathbf{u}}$ is exactly equal to $\mathbf{\Phi\Omega^{-2}}\ddot{\mathbf{q}}$, and no approximation results from introducing equations (2.18). For the damping term, introducing equations (2.18) with n_r vectors in $\mathbf{\Phi}$ is

not exact. However, this is a convenient approximation especially when modal damping is used. Consistent with these considerations, equations (2.11) can be rewritten as

$$\mathbf{u}(t) = g(t)\mathbf{K}^{-1}\mathbf{f} - \mathbf{\Phi}\mathbf{\Omega}^{-2}\overline{\mathbf{C}}\dot{\mathbf{q}} - \mathbf{\Phi}\mathbf{\Omega}^{-2}\ddot{\mathbf{q}} \quad (2.19)$$

The key to the effectiveness of this method is that the static solution is included explicitly in the solution. It is also simple to apply, since it essentially just superimposes the static and mode displacement solutions. Since $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are obtained from the mode displacement solution, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are identical to the values obtained in the mode displacement method.

2.2.3. Static Mode Method

An alternative approach to the mode acceleration method that accounts for the static solution slightly differently is termed the static mode method herein. In this method, the static solution is included as an additional "mode" in forming the reduced equations (2.5). The procedure begins by calculating a set of $n_r - 1$ eigenvectors $\mathbf{\Phi}$ with equations (2.3). Then the static solution is calculated as

$$\mathbf{K}\widehat{\mathbf{\Phi}}_1 = \mathbf{f} \quad (2.20)$$

To improve the orthogonality of the basis vectors, the components of the vibration modes are removed from the static solution by use of the Gram-Schmidt process

$$\overline{\phi}_1 = \widehat{\phi}_1 - \mathbf{\Phi}\mathbf{c} \quad (2.21)$$

where

$$\mathbf{c} = \mathbf{\Phi}^T\mathbf{M}\widehat{\phi}_1 \quad (2.22)$$

The vector $\overline{\phi}_1$ is then concatenated with $\mathbf{\Phi}$ to yield a new $\mathbf{\Phi}$ which is the complete basis. Equations (2.5) now become coupled and can be solved directly or reduced to an uncoupled form with the following procedure. First the reduced eigenvalue problem

$$\overline{\mathbf{M}}\mathbf{Z}\mathbf{\Lambda} + \overline{\mathbf{K}}\mathbf{Z} = \mathbf{0} \quad (2.23)$$

is solved for the $n_r \times n_r$ diagonal matrix of eigenvalues $\mathbf{\Lambda}$ and the $n_r \times n_r$ matrix of eigenvectors $\mathbf{Z}$. Now a new set of basis vectors can be written as

$$\widehat{\mathbf{\Phi}} = \mathbf{\Phi}\mathbf{Z} \quad (2.24)$$

When $\widehat{\mathbf{\Phi}}$ is substituted for $\mathbf{\Phi}$ in equations (2.6), (2.7), (2.8), and (2.9), an uncoupled system results when $\mathbf{C}$ is of the special form described in section 2.2.1.

The static mode method is similar to the mode acceleration method in that the static displacement vector is explicitly included in the solution. However, in the mode acceleration method the amplitude of the static displacement vector is not an unknown but is determined by $g(t)$, whereas in the static mode method the amplitude varies to possibly improve the solution. Also, this static displacement vector participates in the calculation of $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ to possibly improve them as well.

2.2.4. Ritz-Wilson-Lanczos Method

A fourth method, which has become popular in the past few years is termed the Ritz-Wilson-Lanczos (RWL) method and is described in references 20, 21, and 22. Instead of using eigenvectors of the structure, this method uses a set of Lanczos vectors to form the reduced equations. The algorithm used here follows that in reference 20. The first vector is obtained by solving the static equations (2.20) and then scaling so that

$$\phi_1 = \frac{\widehat{\phi}_1}{\left(\widehat{\phi}_1^T\mathbf{M}\widehat{\phi}_1\right)^{1/2}} \quad (2.25)$$

The vectors $i = 2, \ldots, m$ are obtained as follows. First,

$$\mathbf{K}\widehat{\phi}_i = \mathbf{M}\phi_{i-1} \quad (2.26)$$

is solved for $\widehat{\phi}_i$. Then $\widehat{\phi}_i$ is made $\mathbf{M}$-orthogonal with respect to all previously generated vectors by using a Gram-Schmidt process

$$\widetilde{\phi}_i = \widehat{\phi}_i - \sum_{j=1}^{i-1} c_{ij}\phi_j \quad (2.27)$$

where

$$c_{ij} = \phi_j^T\mathbf{M}\widehat{\phi}_i \quad (2.28)$$

and scaling gives

$$\phi_i = \frac{\widehat{\phi}_i}{\left(\widehat{\phi}_i^T\mathbf{M}\widehat{\phi}_i\right)^{1/2}} \quad (2.29)$$

It has been pointed out in many references (e.g., ref. 21) that the $\mathbf{M}$-orthogonalization (eq. (2.27)) is theoretically required only with respect to the two previously computed vectors. However, it is also well-known that round-off errors cause the Lanczos vectors to become less and less orthogonal. Performing the Gram-Schmidt operation with respect to all previously generated vectors will not ensure the $\mathbf{M}$-orthogonality of the vectors. However, it can improve the orthogonality in some cases.

Following reference 20, a final step is performed in generating the basis vectors to produce an uncoupled

dynamic system. Of course this is useful only when the system is undamped or $\overline{\mathbf{C}}$ is assumed to be diagonal. As in the static mode method, a reduced-order eigenvalue problem is solved (eqs. (2.23)), and a new set of basis vectors produced. The process of explicitly computing the reduced stiffness and mass matrices required in equations (2.23) helps alleviate the problems caused by the lack of orthogonality of the Lanczos vectors. The matrices $\overline{\mathbf{M}}$ and $\overline{\mathbf{K}}$ in equations (2.23) are assumed to be full. That is, no assumptions are made that particular terms in $\overline{\mathbf{M}}$ and $\overline{\mathbf{K}}$ are zero based on the properties of the vectors.

2.3. Transient Response Solution Method

When the reduction methods are used and general damping is included in the model, equations (2.5) are coupled. In principle any of the implicit or explicit numerical integration methods used for solving equations (2.1) could be used to solve equations (2.5). In contrast to equations (2.1), however, equations (2.5) are low order, not stiff, and the primary concern is accurately integrating every equation in the system. Therefore an integration method which reduces truncation errors in the solution is highly desirable. Accuracy is especially important in sensitivity analyses because errors in the solution process are usually magnified in the calculation of derivatives.

An approach that allows the use of moderately large time steps and makes the truncation error very small is called the matrix series expansion method in reference 23 and the transfer matrix method in references 24 and 25 when applied to structural dynamics problems and it is often referred to as a Taylor series method in numerical analysis texts (e.g., ref. 26). This method expands the solution in a Taylor series where the number of terms determines the accuracy of the approximation. With this series, an expression can be written for the solution at time $t+\Delta t$ in terms of the solution and load at time t as follows:

$$\begin{Bmatrix} \mathbf{q}(t+\Delta t) \\ \dot{\mathbf{q}}(t+\Delta t) \end{Bmatrix} = \begin{bmatrix} \mathbf{W}_{11} & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \begin{Bmatrix} \mathbf{q}(t) \\ \dot{\mathbf{q}}(t) \end{Bmatrix} + \begin{bmatrix} \mathbf{N}_{11} & \mathbf{N}_{12} \\ \mathbf{N}_{21} & \mathbf{N}_{22} \end{bmatrix} \begin{Bmatrix} g(t)\mathbf{f} \\ \dot{g}(t)\mathbf{f} \end{Bmatrix} \quad (2.30)$$

It has been assumed here that the time variation of the load $g(t)$ is approximated as a piecewise linear function of time, and therefore the second and higher order derivatives equal zero. Expressions for $\mathbf{W}_{ij}$ and $\mathbf{N}_{ij}$ are fairly complex and can be found in references 23 and 24. The values of the coefficients $\mathbf{W}_{ij}$ and $\mathbf{N}_{ij}$ depend on the number of terms taken in the series.

The convergence properties of the $\mathbf{W}_{ij}$ series for an undamped, single-degree-of-freedom case can be studied by considering the following Taylor series expansion:

$$\cos \omega\Delta t = 1 - \frac{(\omega\Delta t)^2}{2} + \frac{(\omega\Delta t)^4}{4!} - \frac{(\omega\Delta t)^6}{6!} + \frac{(\omega\Delta t)^8}{8!} - \cdots \quad (2.31)$$

It is well-known that round-off errors due to finite precision arithmetic will cause large errors in this series for "large" values of $\omega\Delta t$. Thus if ω is taken as ω_{n_r}, an upper bound on Δt can be estimated based on round-off error. In practice Δt usually needs to be much smaller than this upper bound value for two reasons. The first reason is that the input load history may be a complicated function of time, and Δt must be small enough to accurately sample this loading. The second, more important, reason is that Δt must be small enough to accurately sample the history of the output quantities. If Δt is larger than the smallest significant period of response, peak values of the response quantities will likely be missed. Accordingly, in the studies reported herein Δt was taken to be approximately one-eighth of the smallest period. Since the number of terms in the series has only a very small effect on the computational cost of the method, 50 terms were used in this study to make the truncations errors negligible.

Chapter 3

Critical Point Constraint

3.1. Constraint Formulation

The general form of the constraint equation is

$$g_i(\mathbf{x}, t) \leq 0 \tag{3.1}$$

where $\mathbf{x}$ is a vector of design variables and t is time. An effective approach for ensuring that this constraint is satisfied for all values of t is the "critical point constraint" approach described in reference 27, pages 168–169. In this approach a set of peak values of the function g_i (denoted critical points) is selected. An obvious point to include is the time with the "worst" value of g_i. However, if only this point is included, an optimization process modifying a structure based on this information might unknowingly produce a design where the constraint is violated at another time point. To guard against this possibility, a number of important peaks are selected. References 28 and 29 consider in detail the efficient location of critical points in large-scale structures problems with many constraints. This chapter presents a method for selecting the most important peaks as critical points.

In the work reported herein, constraints are assumed to be placed on the displacements, velocities, accelerations, and stresses in the structure. All these constraints are treated similarly. Thus the critical point constraint formulation is illustrated for the case of displacements. Constraints are placed on selected displacements such that

$$|u_i(\mathbf{x}, t)| \leq u_{\text{allow}} \tag{3.2}$$

where u_i are the displacements at specific points in the structure and u_{allow} is the absolute maximum allowable value of the displacement. The critical values of this constraint occur at points in time where u_i has the largest magnitude. These are identified by examining every value of u along the response history. In the implementation here, each constraint is assumed to have a specified number of critical points; five critical points for each u_i are selected. Values of u where $du/dt = 0$ or values of u at the end points of the time interval are local maxima of g_i and are termed candidate critical points.

3.2. Selection of Critical Points

The procedure for selecting the critical points from these candidates can best be explained by referring to an example displacement time history shown in figure 3.1. The critical points are labeled with numbers and a few of the many candidate critical points are labeled with letters. The selection criteria applied to every candidate critical point are explained as follows by considering these few candidate points. Candidate critical points a and c were discarded because the absolute values of the displacements at these points were smaller than those at the five other critical points. The criterion for discarding candidate points b, d, and e is slightly more complicated. From figure 3.1 it can be seen that all three candidate points have larger displacement magnitudes than that of critical point 1, for example. However, candidate points b, d, and e are all part of "major" peaks where a critical point is selected. A second criterion applied to the selection process is a requirement that only one critical point from each major peak be selected. This ensures that the critical points represent the total dynamic response rather than just the high-frequency undulations on, at worst, a single major peak.

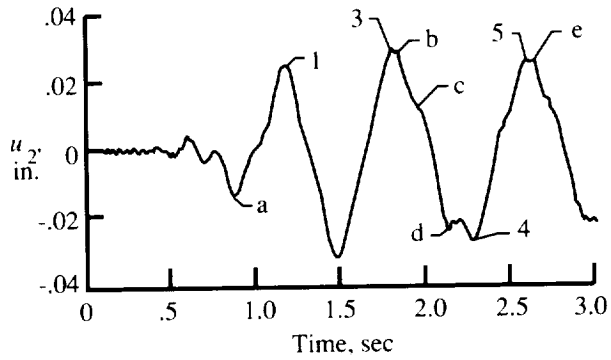


Figure 3.1. Example displacement time history illustrating critical point constraint selection process.

A major peak is identified with the following procedure. Whenever a critical point is selected after comparing its magnitude with that at other critical points, a special screening process is activated. This screening process tests the displacement at every subsequent time point to determine if it differs from that at this last selected critical point by at least a specified percentage (25 percent for the studies reported herein). If so, all subsequent time points are no longer considered part of the current major

peak. Any candidate critical points identified while this special screening process is in effect are compared only against the last selected critical point.

An example is the major peak in figure 3.1 which contains points d and 4. In the selection process, point d is initially selected as a critical point and the screening process is activated. The three points where $du/dt = 0$ between point d and point 4 are recognized to be part of the same major peak as d, but since the magnitude of the displacements at these points is smaller than at point d, they are discarded. Point 4 is also part of the same major peak as point d, but since the displacement magnitude there is larger than at point d, it replaces point d as a critical point. Before the next candidate critical point is considered, the displacement has changed from that at point 4 by more than 25 percent and therefore is considered to be on a new major peak.

3.3. Derivatives of Critical Point Constraints

Once the critical points have been identified for the nominal design, these can be used in calculating sensitivities. Reference 27 demonstrates that the change in time location of critical points can be neglected in calculating derivatives of peak values with respect to design variables by examining the expression for the total derivative of g_i with respect to a design variable x. Considering a constraint $g(x,t)$ at a critical time t_c gives

$$\frac{dg(x,t_c)}{dx} = \frac{\partial g}{\partial x} + \frac{\partial g}{\partial t}\frac{dt_c}{dx} \tag{3.3}$$

The last term in equation (3.3) is always zero because at interior critical points $\partial g/\partial t = 0$ and on the boundary $dt_c/dx = 0$. Accordingly, the sensitivity calculations need to be performed only at the specific times where the critical points have been identified. This can result in a considerable savings in computational time, especially when there are many constraints, many time points, or many basis functions used to represent the response. The details of each sensitivity calculation method are discussed in the next chapter.

Chapter 4

Methods For Calculating Sensitivities

4.1. Finite Difference Methods

4.1.1. Forward and Central Difference Operators

Both the forward difference and central difference methods have been used in this study to calculate sensitivities. The well-known forward difference approximation to du/dx,

$$\frac{\Delta u}{\Delta x} = \frac{u(x+\Delta x) - u(x)}{\Delta x} \tag{4.1}$$

and the central difference approximation,

$$\frac{\Delta u}{\Delta x} = \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} \tag{4.2}$$

are used. The truncation error for the forward difference approximation is

$$e_T(\Delta x) = \frac{\Delta x}{2}\frac{d^2u}{dx^2}(x+\zeta\Delta x) \qquad (0 \le \zeta \le 1) \tag{4.3}$$

and for the central difference approximation is

$$e_T(\Delta x) = \frac{(\Delta x)^2}{6}\frac{d^3u}{dx^3}(x+\zeta\Delta x) \qquad (0 \le \zeta \le 1) \tag{4.4}$$

In applying equations (4.1) and (4.2), the selection of difference step size Δx is a concern. Selection of a large step size results in errors in the derivative due to truncation of the operator (eqs. (4.3) and (4.4)). Selection of a small step size can lead to errors in the derivative due to the limited floating point precision of the computer or algorithmic inaccuracies in calculating u (condition errors). It is not uncommon with the forward difference method (eq. (4.1)) that no acceptable value exists for Δx to produce an accurate value of du/dx considering the conflicting requirements of minimizing truncation and condition errors. Because the truncation error associated with the operator of equation (4.2) is typically less than that of equation (4.1), it is possible to use a larger finite difference step size. The larger Δx reduces the condition error from the function evaluations and results in a more accurate value of du/dx. However, the necessity of two function evaluations needed for equation (4.2) makes the procedure computationally more costly.

4.1.2. Using Vibration Modes as Basis Functions

For many of the studies herein the natural vibration mode shapes are used as basis functions to represent the transient response. In calculating the response of the perturbed design in equation (4.1) and the two perturbed designs in equation (4.2), some computational savings are possible relative to the computations for the initial design.

If the mode shapes for the initial design are used to represent the perturbed design, the cost of resolving the eigenvalue problem is eliminated. However, the reduced set of equations for the perturbed system must still be formed and $\overline{\mathbf{M}}$, $\overline{\mathbf{C}}$, and $\overline{\mathbf{K}}$ are now full. This coupled system is then solved with the matrix series expansion method described in section 2.3.

If the updated mode shapes for the perturbed design are used in the analysis, many eigensolution procedures, such as the subspace iteration used here, can begin with the mode shapes from the initial design as approximations. Since the perturbation in the design is small, the subspace iteration procedure converges rapidly. However, at least one factorization of $\mathbf{K}$ is required. For large finite element models, this can be the largest part of the computational cost. For most of the studies in this paper, the forward difference method used the initial mode shapes to represent the perturbed design. Because the central difference method was used for reference values of derivatives, updated mode shapes were calculated for the two required perturbed designs. In both cases, because of the critical point constraint formulation, the transformation from modal coordinates to physical coordinates (e.g., displacements, stresses) is performed only at the critical times instead of at all time points.

4.2. Semianalytical Methods

The direct method for sensitivity calculation is derived by differentiating equations (2.1). The derivation presented here follows that in reference 27, pages 169–171. After differentiating equations (2.1) with respect to the design variable x the result is

$$\mathbf{M}\frac{d\ddot{\mathbf{u}}}{dx} + \mathbf{C}\frac{d\dot{\mathbf{u}}}{dx} + \mathbf{K}\frac{d\mathbf{u}}{dx} = \frac{d\mathbf{f}}{dx}g(t) - \frac{d\mathbf{M}}{dx}\ddot{\mathbf{u}} - \frac{d\mathbf{C}}{dx}\dot{\mathbf{u}} - \frac{d\mathbf{K}}{dx}\mathbf{u} \quad (4.5)$$

This system of differential equations of order n_g could be solved directly for the sensitivities $d\mathbf{u}/dx$, $d\dot{\mathbf{u}}/dx$, and $d\ddot{\mathbf{u}}/dx$. However, just as for the response equations, it is more efficient to consider a reduced form of the sensitivity equations which can be obtained by differentiating equations (2.5) with respect to x to yield

$$\overline{\mathbf{M}}\frac{d\ddot{\mathbf{q}}}{dx} + \overline{\mathbf{C}}\frac{d\dot{\mathbf{q}}}{dx} + \overline{\mathbf{K}}\frac{d\mathbf{q}}{dx} = \frac{d\bar{\mathbf{f}}}{dx}g(t) - \frac{d\overline{\mathbf{M}}}{dx}\ddot{\mathbf{q}} - \frac{d\overline{\mathbf{C}}}{dx}\dot{\mathbf{q}} - \frac{d\overline{\mathbf{K}}}{dx}\mathbf{q} \quad (4.6)$$

The first step in forming this equation is the calculation of the derivatives of $\bar{\mathbf{f}}$, $\overline{\mathbf{M}}$, $\overline{\mathbf{C}}$, and $\overline{\mathbf{K}}$ (eqs. (2.9), (2.6), (2.7), and (2.8)) with respect to x. Using equations (2.9) gives the derivative of $\bar{\mathbf{f}}$ with respect to x as follows:

$$\frac{d\bar{\mathbf{f}}}{dx} = \frac{d\mathbf{\Phi}^T}{dx}\mathbf{f} + \mathbf{\Phi}^T\frac{d\mathbf{f}}{dx} \quad (4.7)$$

The force $\mathbf{f}$ is frequently not a function of the design variables; this simplifies equation (4.7). Also, with equation (2.6), the derivative of $\overline{\mathbf{M}}$ with respect to x can be written as

$$\frac{d\overline{\mathbf{M}}}{dx} = \mathbf{\Phi}^T\frac{d\mathbf{M}}{dx}\mathbf{\Phi} + \frac{d\mathbf{\Phi}^T}{dx}\mathbf{M}\mathbf{\Phi} + \mathbf{\Phi}^T\mathbf{M}\frac{d\mathbf{\Phi}}{dx} \quad (4.8)$$

Similar expressions can be written for the derivatives of $\overline{\mathbf{C}}$ and $\overline{\mathbf{K}}$.

The derivative $d\mathbf{M}/dx$ in equation (4.8) (similarly for $d\mathbf{C}/dx$ and $d\mathbf{K}/dx$) is, in general, difficult to calculate because the finite element model may be composed of diverse element types whose properties are complicated functions of the design variables x. For this reason, these derivatives are often replaced with finite difference approximations. This combination of analytical differentiation of the response equations with finite difference matrix derivatives is known as a semianalytical approach. The semianalytical methods presented herein for calculating transient response quantities all use the forward difference operator to approximate $d\mathbf{M}/dx$, $d\mathbf{C}/dx$, $d\mathbf{K}/dx$, and $d\mathbf{f}/dx$. For several important classes of design variables, however, $\mathbf{M}$, $\mathbf{C}$, and $\mathbf{K}$ are linear functions of x. For example, $\mathbf{M}$ and $\mathbf{K}$ in a finite element model composed of truss members are linear functions of member cross-sectional area. In these cases, there are no truncation errors and large finite difference step sizes can be used to reduce the condition error and produce accurate derivatives.

Calculation of the first term in equation (4.7) and the second and third terms in equation (4.8) depends on the particular choice of basis functions $\mathbf{\Phi}$. Considerable reduction in computational cost results if the vectors $\mathbf{\Phi}$ are taken to be independent of x that is fixed. Methods which are less costly than exact methods are also available to approximate $d\mathbf{\Phi}/dx$. Two semianalytical procedures which address these concerns are discussed in sections 4.2.1 and 4.2.2.

4.2.1. Fixed-Mode Semianalytical Formulation

If the basis vectors are assumed not to be functions of the design variables x, $d\mathbf{\Phi}/dx$ equals 0. This significantly simplifies equations (4.7) and (4.8). After forming the derivatives of $\bar{\mathbf{f}}$, $\overline{\mathbf{M}}$, $\overline{\mathbf{C}}$, and $\overline{\mathbf{K}}$, the right-hand side of equations (4.6) can be formed using $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ from the solution of equations (2.5). The matrix series expansion method ensures that accurate values of $\ddot{\mathbf{q}}$, $\dot{\mathbf{q}}$, and $\mathbf{q}$ are available for this step. Equations (4.6) can then be integrated to yield $d\ddot{\mathbf{q}}/dx$, $d\dot{\mathbf{q}}/dx$, and $d\mathbf{q}/dx$. Herein this fixed-mode, semianalytical implementation of the direct method is called the semianalytical method.

4.2.2. Variable-Mode Semianalytical Formulation

If the basis functions are assumed to be functions of x, the calculation of $d\mathbf{\Phi}/dx$ either exactly or approximately is required to form equations (4.7) and (4.8). Vibration modes are the most popular basis functions, and the calculation of their derivatives has been studied extensively. Reference 30, for example, surveys several methods for calculating derivatives of vibration mode shapes from a computational point of view. One of the most popular methods, Nelson's method (ref. 31), requires a factorization of the system equations for each eigenvector considered. This can be a considerable computational burden for large systems. Since the overall objective here is the accurate calculation of transient response sensitivities, not eigenvector sensitivities, it seems desirable to investigate cheaper approximate methods for calculating $d\mathbf{\Phi}/dx$.

One approximate method for calculating the eigenvector derivatives is similar to the modal approach for transient analysis. This modal method approximates each eigenvector derivative as a linear combination of the modes themselves. In many cases, however, a very large number of

eigenvectors are required for accurate derivatives. Furthermore, the eigenvector derivative approximation produced by this method cannot improve the transient response sensitivities because they are contained entirely within the span of the modes themselves.

A method proposed by Wang (ref. 32) to alleviate the poor performance of the modal method also improves the transient sensitivities. This modified modal method is derived by first differentiating equations (2.3) to yield

$$(\mathbf{K}-\omega_j^2\mathbf{M})\frac{d\mathbf{\Phi}_j}{dx}=\frac{d\omega_j^2}{dx}\mathbf{M}\mathbf{\Phi}_j-\frac{d\mathbf{K}}{dx}\mathbf{\Phi}_j+\omega_j^2\frac{d\mathbf{M}}{dx}\mathbf{\Phi}_j \tag{4.9}$$

This equation cannot be solved directly since the left-hand side is singular. Wang's approach, however, was to calculate a pseudostatic solution to this equation by neglecting $\omega_j^2\mathbf{M}$ on the left-hand side of equations (4.9). The solution to this pseudostatic equation introduces the change in basis associated with changes in the design variables and is significant in improving the transient response sensitivities. The mode shape derivative can then be written as

$$\frac{d\mathbf{\Phi}_j}{dx}=\left(\frac{d\mathbf{\Phi}_j}{dx}\right)_s+\sum_{k=1}^{n_r}A_{jk}\mathbf{\Phi}_k \tag{4.10}$$

where $(d\mathbf{\Phi}_j/dx)_s$ is the pseudostatic contribution. The coefficients A_{jk} are obtained by substituting equation (4.10) into equations (4.9), multiplying by $\mathbf{\Phi}_j^T$, and simplifying as

$$A_{jk}=\frac{\omega_j^2\mathbf{\Phi}_k^T\left(\frac{d\mathbf{K}}{dx}-\omega_j^2\frac{d\mathbf{M}}{dx}\right)\mathbf{\Phi}_j}{\omega_k^2(\omega_j^2-\omega_k^2)} \qquad (k\neq j) \tag{4.11}$$

or

$$A_{jj}=-\frac{1}{2}\mathbf{\Phi}_j^T\frac{d\mathbf{M}}{dx}\mathbf{\Phi}_j \qquad (k=j) \tag{4.12}$$

Given these approximate values of eigenvector derivatives, equations (4.7) and (4.8) can be formed. Then equations (4.6) can be solved for $d\ddot{\mathbf{q}}/dx$, $d\dot{\mathbf{q}}/dx$, and $d\mathbf{q}/dx$ just as in the fixed-mode, semianalytical method.

4.2.3. Recovery of Physical Sensitivities

Given $d\mathbf{q}/dx$, the derivative of the physical displacement vector $d\mathbf{u}/dx$ can be written as

$$\frac{d\mathbf{u}}{dx}=\mathbf{\Phi}\frac{d\mathbf{q}}{dx}+\frac{d\mathbf{\Phi}}{dx}\mathbf{q} \tag{4.13}$$

with similar expressions for $d\dot{\mathbf{u}}/dx$ and $d\ddot{\mathbf{u}}/dx$. The calculation of stresses begins with

$$\boldsymbol{\sigma}=\mathbf{S}\mathbf{u} \tag{4.14}$$

where $\mathbf{S}$ is the stress transformation matrix. Substituting equation (2.4) yields

$$\boldsymbol{\sigma}=\mathbf{S}\mathbf{\Phi}\mathbf{q} \tag{4.15}$$

when equation (4.15) is differentiated, the stress sensitivities can be written as

$$\frac{d\boldsymbol{\sigma}}{dx}=\mathbf{S}\mathbf{\Phi}\frac{d\mathbf{q}}{dx}+\frac{d\mathbf{S}}{dx}\mathbf{\Phi}\mathbf{q}+\mathbf{S}\frac{d\mathbf{\Phi}}{dx}\mathbf{q} \tag{4.16}$$

The matrix $d\mathbf{S}/dx$ is approximated with the forward difference operator. Because of the critical point constraint formulation, the transformation from these modal quantities to physical displacements, velocities, accelerations, and stresses is performed only at the critical times.

4.2.4. Mode Acceleration Method

The mode acceleration method was presented in chapter 2 as a technique for improving the dynamic displacements and stresses when the static component is significant. It is also possible that it can improve the sensitivities of displacements and stresses. An expression for the sensitivities using the mode acceleration method is obtained by first rearranging equations (4.5) to yield

$$\frac{d\mathbf{u}(t)}{dx}=\mathbf{K}^{-1}\left[\frac{d\mathbf{f}}{dx}g(t)-\frac{d\mathbf{K}}{dx}\mathbf{u}-\mathbf{C}\frac{d\dot{\mathbf{u}}}{dx}-\frac{d\mathbf{C}}{dx}\dot{\mathbf{u}}-\mathbf{M}\frac{d\ddot{\mathbf{u}}}{dx}-\frac{d\mathbf{M}}{dx}\ddot{\mathbf{u}}\right] \tag{4.17}$$

If a reduced basis approximation is applied uniformly to every term in equations (4.17), the resulting $d\mathbf{u}/dx$ would agree with that obtained from the solution of equations (4.6). The objective of a mode acceleration solution is to selectively apply the modal approximation to equations (4.17) with the goal of improving the values of $d\mathbf{u}/dx$. In applying the mode acceleration method to the transient response problem (eqs. (2.11)), $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are obtained from the mode displacement method. Here, in applying the mode acceleration method to the sensitivity equations, $\mathbf{u}$ is obtained from the solution to equations (2.19) and $d\dot{\mathbf{u}}/dx$ and $d\ddot{\mathbf{u}}/dx$ are obtained from the solution to the mode displacement, semianalytical equation (eqs. (4.6)). In the derivation here, the

modes $\mathbf{\Phi}$ are assumed to be fixed. Substituting equations (2.19) and $\dot{\mathbf{u}} = \mathbf{\Phi}\dot{\mathbf{q}}$, $\ddot{\mathbf{u}} = \mathbf{\Phi}\ddot{\mathbf{q}}$, $d\dot{\mathbf{u}}/dx = \mathbf{\Phi}d\dot{\mathbf{q}}/dx$, and $d\ddot{\mathbf{u}}/dx = \mathbf{\Phi}d\ddot{\mathbf{q}}/dx$ into equations (4.17) yields

$$\frac{d\mathbf{u}(t)}{dx} = \mathbf{K}^{-1}\left[\frac{d\mathbf{f}}{dx}g(t) - \frac{d\mathbf{K}}{dx}\left[g(t)\mathbf{K}^{-1}\mathbf{f} - \mathbf{\Phi}\mathbf{\Omega}^{-2}\overline{\mathbf{C}}\dot{\mathbf{q}} - \mathbf{\Phi}\mathbf{\Omega}^{-2}\ddot{\mathbf{q}}\right] - \mathbf{C}\mathbf{\Phi}\frac{d\dot{\mathbf{q}}}{dx} - \frac{d\mathbf{C}}{dx}\mathbf{\Phi}\dot{\mathbf{q}} - \mathbf{M}\mathbf{\Phi}\frac{d\ddot{\mathbf{q}}}{dx} - \frac{d\mathbf{M}}{dx}\mathbf{\Phi}\ddot{\mathbf{q}}\right] \quad (4.18)$$

The modal approximation for $\mathbf{K}^{-1}$ (eqs. (2.18)) is introduced into all terms in equation (4.18) that involve damping just as in the mode acceleration solution described in chapter 2. It was also pointed out in chapter 2 that $\mathbf{K}^{-1}\mathbf{M}\mathbf{\Phi}$ in equation (4.18) is exactly $\mathbf{\Phi}\mathbf{\Omega}^{-2}$. Based on these considerations, equation (4.18) can be simplified to yield the mode acceleration solution of the sensitivity equations

$$\frac{d\mathbf{u}(t)}{dx} = \mathbf{K}^{-1}\left[\frac{d\mathbf{f}}{dx} - \frac{d\mathbf{K}}{dx}\mathbf{K}^{-1}\mathbf{f}\right]g(t) + \mathbf{\Phi}\mathbf{\Omega}^{-2}\left[\frac{d\overline{\mathbf{K}}}{dx}\mathbf{\Omega}^{-2}\overline{\mathbf{C}} - \frac{d\overline{\mathbf{C}}}{dx}\right]\dot{\mathbf{q}} - \mathbf{\Phi}\mathbf{\Omega}^{-2}\overline{\mathbf{C}}\frac{d\dot{\mathbf{q}}}{dx} + \mathbf{K}^{-1}\left[\frac{d\mathbf{K}}{dx}\mathbf{\Phi}\mathbf{\Omega}^{-2} - \frac{d\mathbf{M}}{dx}\mathbf{\Phi}\right]\ddot{\mathbf{q}} - \mathbf{\Phi}\mathbf{\Omega}^{-2}\frac{d\ddot{\mathbf{q}}}{dx} \quad (4.19)$$

The key to the effectiveness of this mode acceleration sensitivity method is the usage of the exact $\mathbf{K}^{-1}$ in the calculation of the $\mathbf{K}^{-1}(d\mathbf{K}/dx)\mathbf{\Phi}\mathbf{\Omega}^{-2}$ and $\mathbf{K}^{-1}(d\mathbf{M}/dx)\mathbf{\Phi}$ terms. The explicit calculation of these terms expands the basis beyond the span of the modes in a manner similar to the pseudostatic term in the modified modal method described in the previous section. When equations (4.19) are used, the stress sensitivities can be calculated as

$$\frac{d\boldsymbol{\sigma}}{dx} = \mathbf{S}\frac{d\mathbf{u}}{dx} + \frac{d\mathbf{S}}{dx}\mathbf{u} \quad (4.20)$$

where $\mathbf{u}$ is obtained from equations (2.11).

It is worthwhile to contrast the sensitivity approach of equations (4.19) with an alternate approach of a fixed-mode, overall forward difference method with the response quantities calculated with a mode acceleration method. This overall forward difference approach has one obvious drawback. The mode acceleration method requires the costly factorization of $\mathbf{K}$ for the perturbed design. Therefore, much of the cost savings achieved by keeping the modes fixed is lost. A second defect of the overall forward difference method is that it is not as effective as the method of equations (4.19). The mode acceleration method in the overall finite difference procedure provides a good approximation to the first term (pseudostatic term) in equations (4.19). However, the key effects of the $\mathbf{K}^{-1}(d\mathbf{K}/dx)\mathbf{\Phi}\mathbf{\Omega}^{-2}$ and the $\mathbf{K}^{-1}(d\mathbf{M}/dx)\mathbf{\Phi}$ terms are completely neglected.

Chapter 5

Numerical Studies

The different transient response methods described in chapter 2 and the sensitivity calculation methods described in chapter 4 are applied to three example problems in this chapter. The three example problems are small but they all have certain characteristics which complicate the dynamics and sensitivity calculations. The first example is the five-span beam with relatively closely spaced frequencies and loaded with a moment applied at a single point. As a result, many modes participate in the dynamic response. The second example is the delta wing loaded with a uniform pressure load. Although the higher frequency modes are not significantly excited by this loading, the analysis is complicated by the laminated plate elements in the model and the sensitivity analysis is complicated by the lamina thickness design variables considered. The third example is a cantilever beam with a stepped cross section loaded with an applied rotation at the root. This loading is inertial, depends on the mass, and therefore also depends on the values of the design variables. The first two examples consider point mass and standard thickness design variables. The cantilever beam example also includes so-called shape design variables (section lengths) that are known to cause difficulties in the sensitivity analysis in some cases.

One of the key questions addressed in this chapter is how well a particular set of basis vectors represents the full system of order n_g. This full system, however, is the result of a particular finite element discretization. Thus the accuracy of the response or sensitivities as a function of the finite element mesh is also an appropriate question. This question is especially important when a large number of basis vectors (n_r close to n_g) are required for an accurate solution in a problem with a given finite element mesh. Either the basis vectors are doing a very poor job of spanning the solution space or the loading is legitimately exciting this high-frequency behavior.

In this chapter two terms are used to describe these studies which consider the dynamic response as a function of the number of basis vectors or the number of finite elements in the model. The effect of the number of basis vectors on the accuracy of the response or sensitivities for a given finite element mesh is called a "modal convergence" study. In this case, the goal is for the n_r basis vectors to provide an accurate solution to the approximate equations of order n_g. The question of whether the finite element model associated with this system is an accurate representation of the continuum is addressed in a "mesh convergence" study. In some cases, it will be shown that the modal convergence is strongly related to the mesh convergence. That is, when a large number of basis vectors are required for an accurate solution for a given finite element mesh, the finite element mesh is doing a poor job of representing the continuum. In other cases, even though the finite element mesh is providing an accurate representation of the continuum, some sets of basis vectors are doing a poor job of representing the response or sensitivities for this n_gth order system.

Several additional comments on the concept of a modal convergence study are in order. Clearly the use of the term convergence is imprecise because the accuracy of the approximate solution with different numbers of modes is compared only with the finite-degree-of-freedom solution rather than the continuum solution. However, it is assumed that an "acceptable" finite element model must do a good job of representing the low-frequency modes of the structure. Therefore, the accuracy of the dynamic solution with a small number of modes from the finite-degree-of-freedom model is a very reasonable approximation to the accuracy of the dynamic solution with a small number of modes calculated from a continuum model. Thus the convergence of the solution as a function of the number of modes calculated from the finite element model is a reasonable approximation to the true modal convergence obtained when the modes are calculated from the continuum model as long as the number of modes considered is small. Furthermore, if the number of modes required for an accurate calculation of either the response or sensitivities is not small, the basis vectors or the method will be considered poor.

5.1. Five-Span Beam Example

The first example considered is a five-span, planar beam example taken from reference 33 and shown in figure 5.1. An initial investigation of the displacement transient response of this problem was also considered in reference 34. In most of the studies, the beam is modeled with three beam finite elements per span resulting in 26 unconstrained degrees of freedom. The effect of finite element discretization is considered by developing alternate models with 6, 9, 12, ... elements per span. As

shown in figure 5.1, translational and rotational viscous dampers were also added to the beam. These devices are representative of velocity feedback controllers which might be added to flexible structures. Cases with and without dampers were considered. The numerical values of the damping coefficients from reference 33 of $c_1 = 0.008$ sec-lb/in. and $c_2 = 1.2$ sec-lb were used. In one example, modal damping with $\xi_i = 0.005$, which is intended to represent typical structural damping, was used instead of the discrete dampers. A case was also considered where a 1.0-lb mass (approximately 20 percent of the mass of the beam) was added to the beam at the location of the translational damper. The eigenvalues for three cases using the model with three elements per span are shown in table 5.1. The additional point mass has a significant effect on the frequencies, whereas the dampers have little effect. The effect on frequencies of increasing the number of elements per span in the finite element model is shown in table 5.2. It can be seen that the lowest 10 frequencies are fairly well converged even for the model with three elements per span. In the transient analysis, the applied loading for all problems consisted of a point moment of 0.04405 in-lb applied at the right end of the beam. Two different time functions for this load, a step and a ramp (shown in fig. 5.1), were considered.

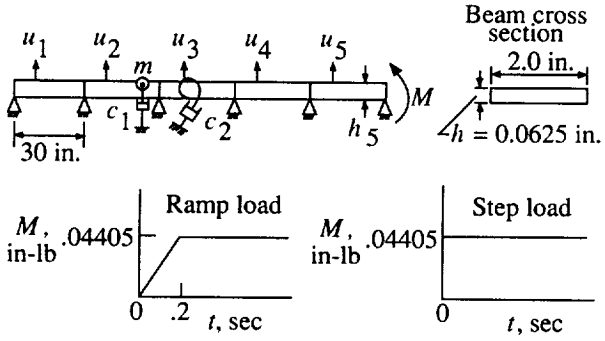


Figure 5.1. Five-span beam with applied end moment.

5.1.1. Beam Dynamic Response

The first part of this study focused on the transient response of the beam with the mode displacement, mode acceleration, static mode, and Ritz-Wilson-Lanczos (RWL) methods. Displacement, velocity, acceleration, and stress resultant response quantities are considered. For this beam example, all these response quantities are taken at a location 10.0 inches from the left end of each span. This point is the end of the first element in each span when three elements per span are used in the model.

5.1.1.1. Character of response. In the first case, the ramp loading was applied to the undamped beam modeled with three elements per span. All 26 modes were used in the analysis. Time histories of selected displacement, velocity, acceleration, and bending moment components are shown in figures 5.2, 5.3, 5.4, and 5.5, respectively. The displacement history (fig. 5.2) is relatively smooth indicating that only the low-frequency modes of the beam are contributing to the response. The velocity and bending moment response histories are more jagged indicating participation by higher frequency modes. The acceleration history (fig. 5.4) is extremely jagged with contributions from the highest frequency modes represented by the finite element model.

The impulsive nature of the step load makes the higher frequency modes much more important. This can be seen in figure 5.6 where the time history of velocity in the second span (fig. 5.1) $\dot{u}_2$ is

Table 5.1. Eigenvalues For Three Five-Span Beam Cases

	Undamped	Damped with point dampers		Undamped with point mass
Mode	Frequency, Hz	Frequency, Hz	Damping ratio	Frequency, Hz
1	1.1707	1.2210	0.0851	0.9401
2	1.2991	1.2926	.0352	1.2594
3	1.6254	1.6298	.0690	1.5445
4	2.0491	2.0910	.0590	1.8005
5	2.4628	2.5497	.0958	2.3729
6	4.7343	4.8426	.0044	4.2327
7	5.0105	4.9785	.0413	4.8858
8	5.6472	5.7703	.0126	5.6400
9	6.4153	6.4178	.0407	5.9261
10	7.1274	7.2229	.0193	6.8762

Table 5.2. Beam Frequencies With Different Numbers of Finite Elements Per Span

	Frequency, Hz, for—			
Mode	3 elements	6 elements	9 elements	12 elements
1	1.1707	1.1698	1.1698	1.1698
2	1.2991	1.2979	1.2978	1.2978
3	1.6254	1.6230	1.6229	1.6229
4	2.0491	2.0445	2.0442	2.0442
5	2.4628	2.4547	2.4542	2.4542
6	4.7343	4.6828	4.6798	4.6792
7	5.0105	4.9504	4.9469	4.9462
8	5.6472	5.5652	5.5601	5.5593
9	6.4153	6.3053	6.2980	6.2967
10	7.1274	6.9974	6.9874	6.9857

shown. By comparing this velocity history with that in figure 5.3 the increased importance of the high-frequency modes becomes obvious.

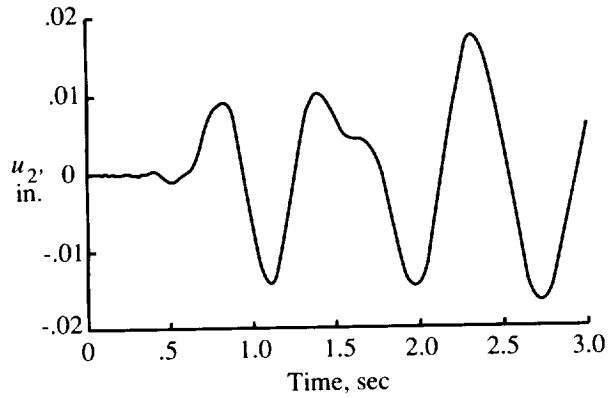


Figure 5.2. Time history of displacement u_2 for five-span beam subjected to transient end moment. Ramp load; undamped beam.

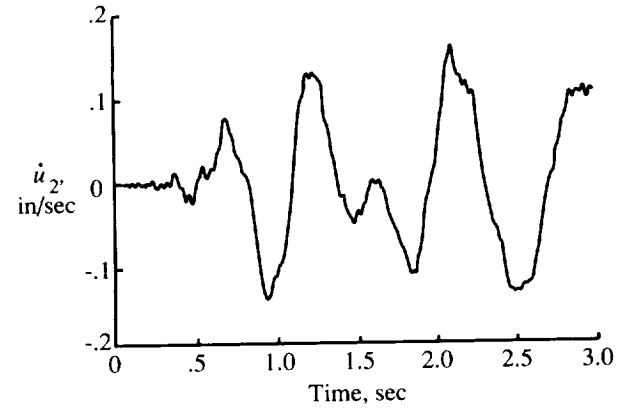


Figure 5.3. Time history of velocity $\dot{u}_2$ for five-span beam subjected to transient end moment. Ramp load; undamped beam.

The addition of the point dampers shown in figure 5.1, on the other hand, tends to reduce the importance of the high-frequency modes. This is shown in figure 5.7 where again $\dot{u}_2$ is shown. Comparing figures 5.7 and 5.3 shows that the velocity history for the damped case is significantly smoother than for the undamped case.

This changing character of the time histories with temporal or spatial differentiation of the response function or the addition of dampers is expected. The implications of this phenomenon on calculating the sensitivities of these response quantities are discussed below.

5.1.1.2. Modal convergence. When vibration modes or other functions are used to reduce the basis in a transient response problem (eq. (2.4)), the

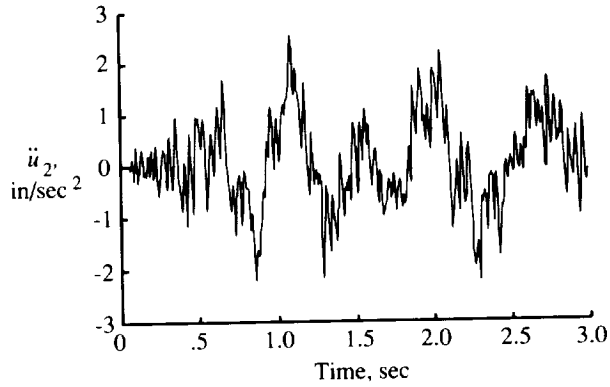


Figure 5.4. Time history of acceleration $\ddot{u}_2$ for five-span beam subjected to transient end moment. Ramp load; undamped beam.

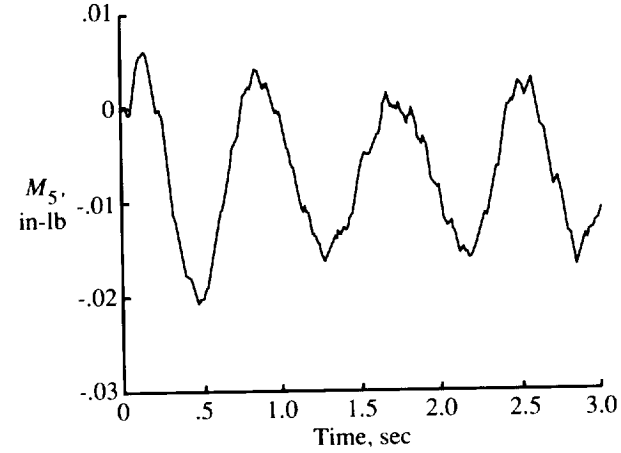


Figure 5.5. Time history of bending moment in span 5 for five-span beam subjected to transient end moment. Ramp load; undamped beam.

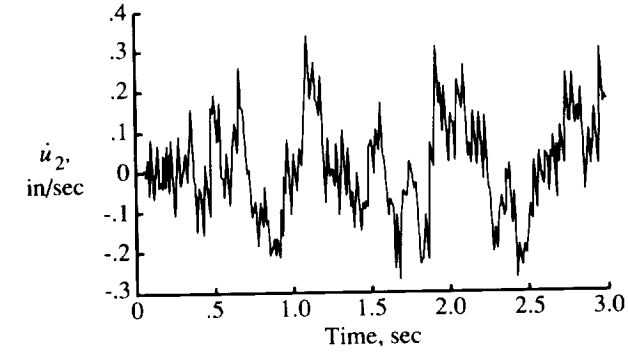


Figure 5.6. Time history of velocity $\dot{u}_2$ for five-span beam subjected to transient end moment. Step load; undamped beam.

key question is how many modes are required for an accurate solution. This section addresses that question for the five-span beam example with the response calculated with mode displacement, mode

acceleration, static mode, and RWL methods. Unless otherwise stated, all the response quantities are considered at critical times selected with the methods discussed in chapter 3.

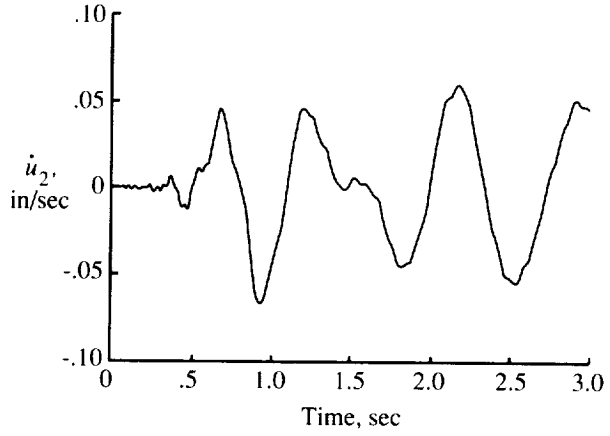


Figure 5.7. Time history of velocity $\dot{u}_2$ for five-span beam subjected to transient end moment. Ramp load; damped beam.

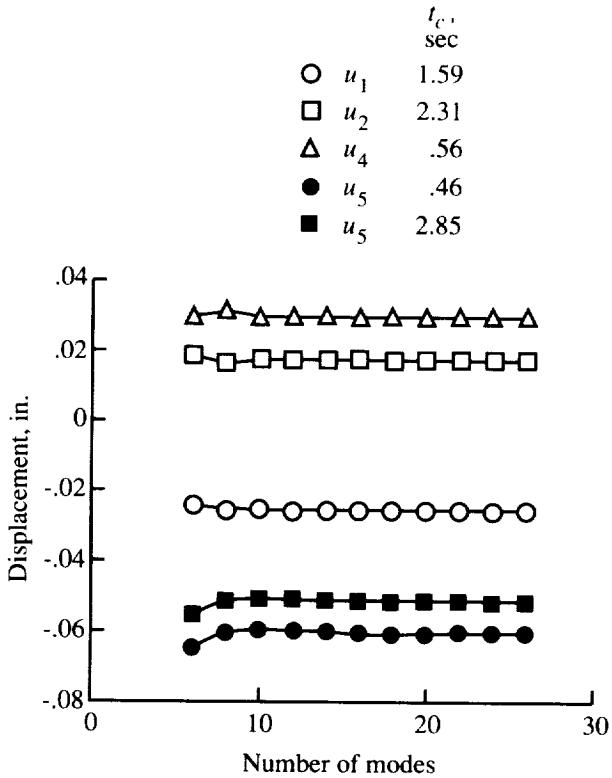


Figure 5.8. Modal convergence of selected displacements for five-span beam. Ramp load; undamped beam; mode displacement method.

The baseline case of ramp loading applied to the undamped beam modeled with three elements per

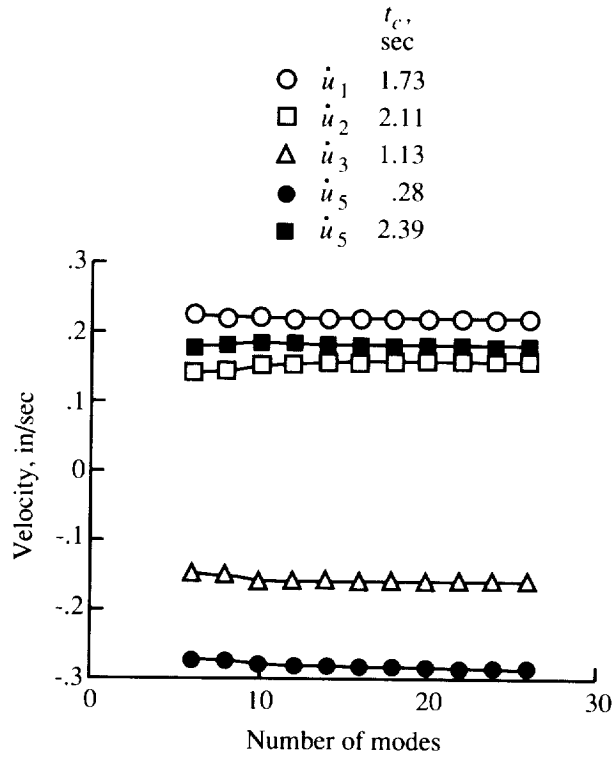


Figure 5.9. Modal convergence of selected velocities for five-span beam. Ramp load; undamped beam; mode displacement method.

span is considered first. Figure 5.8 shows the convergence of selected displacements at critical points as a function of number of modes. The displacement–critical point combinations were selected to be representative of both the largest and smallest critical values. In figure 5.8 and in all the other figures showing convergence of response quantities or sensitivities, the figure key indicates the quantity and the time of occurrence in seconds. In all cases the convergence is very good with approximately 10 modes yielding a converged solution. Figure 5.9 shows a similar plot for velocities. Again the convergence is good. The modal convergence for accelerations, however, is poor as shown in figure 5.10. Figures 5.11 and 5.12 show the modal convergence of selected bending moments and shear forces, respectively, and again, the convergence is poor.

To possibly alleviate this poor convergence, the alternate reduction methods discussed in chapter 2 were applied to this problem. The modal convergence for displacements calculated with the mode acceleration method (fig. 5.13) is even better than that found with the mode displacement method. The convergence of bending moments and shear forces has improved dramatically from the mode displacement results as can be seen in figures 5.14 and 5.15.

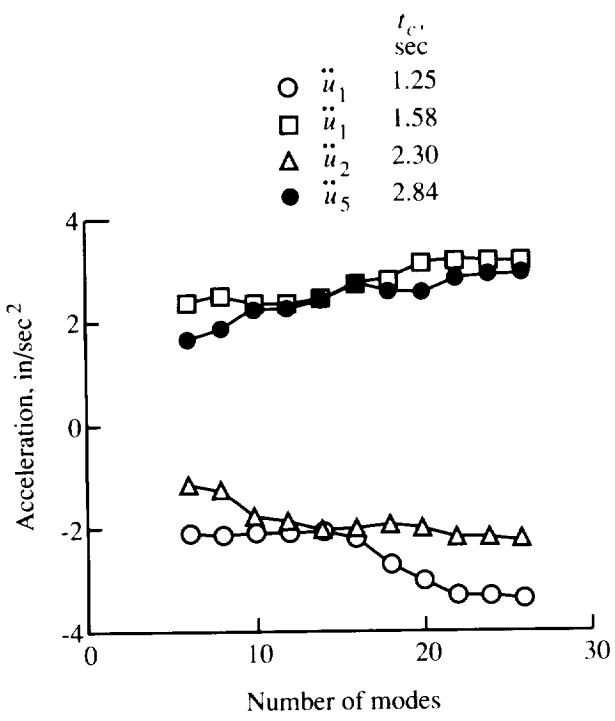


Figure 5.10. Modal convergence of selected accelerations for five-span beam. Ramp load; undamped beam; mode displacement method.

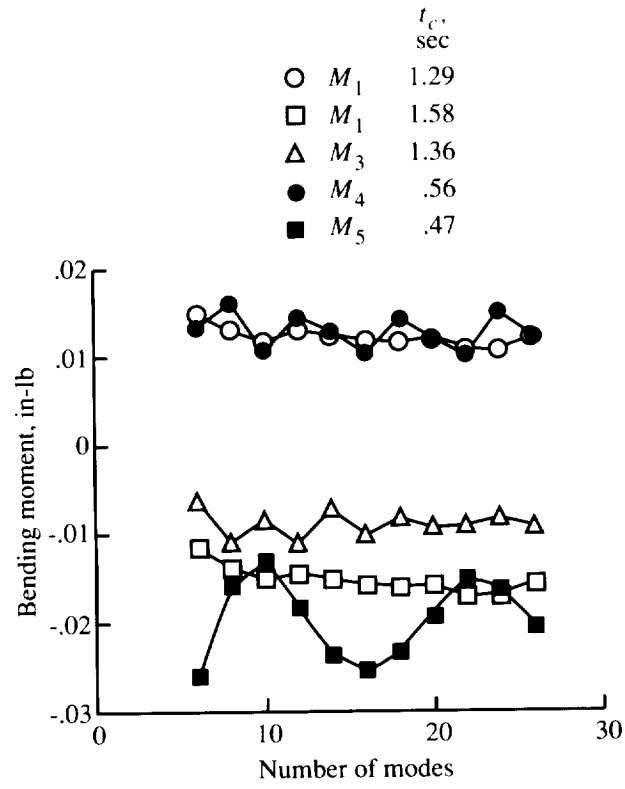


Figure 5.11. Modal convergence of selected bending moments for five-span beam. Ramp load; undamped beam; mode displacement method.

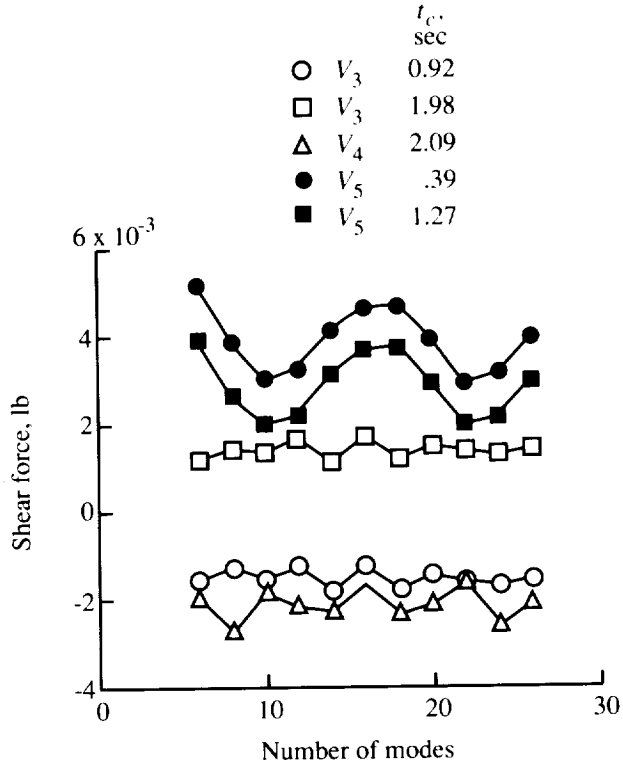


Figure 5.12. Modal convergence of selected shear forces for five-span beam. Ramp load; undamped beam; mode displacement method.

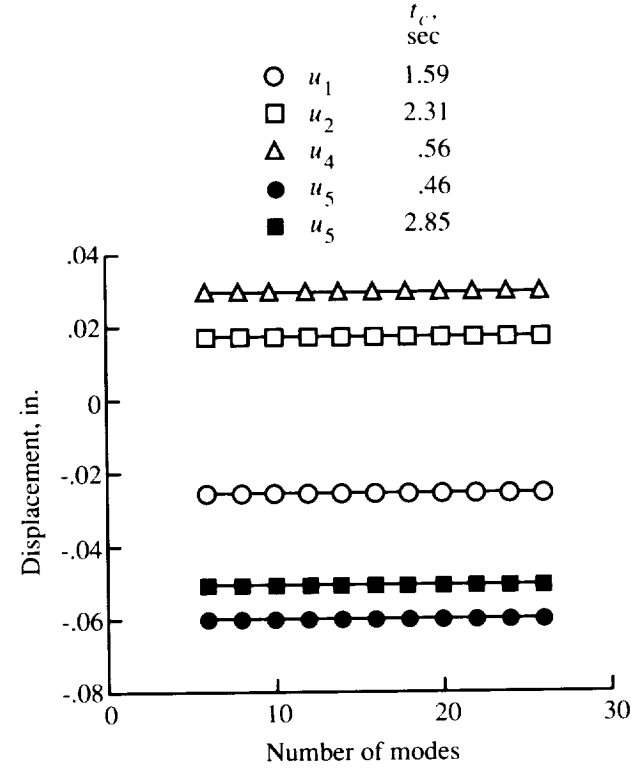


Figure 5.13. Modal convergence of selected displacements for five-span beam. Ramp load; undamped beam; mode acceleration method.

As mentioned in chapter 2, the mode acceleration method does not apply to the calculation of velocities and accelerations.

A similar improvement was noted with the static mode method. As an example, consider the excellent convergence of shear forces shown in figure 5.16. However, the addition of the static solution provides no improvement in the convergence of acceleration as shown in figure 5.17.

The RWL method is attractive because of the significantly reduced cost of calculating the vectors compared with solving the eigenproblem. In this five-span beam example, the modal convergence is also as good as the mode acceleration or static mode methods. The good convergence of the shear forces is shown in figure 5.18. Like the other reduction methods, however, the convergence of accelerations is poor (fig. 5.19).

The modal convergence of the response quantities for the step-loaded case is generally much poorer than for the ramp-loaded case. The convergence of the displacements is reasonably good. Convergence of velocities, accelerations, and stresses, however, is poor. This poor convergence is not surprising considering the "jaggedness" of the velocity time history shown in figure 5.6. As an example, two figures with the convergence of bending moments plotted as a function of the number of modes are shown. The first, figure 5.20, shows the bending moments calculated with the mode displacement method. Convergence is poor but this is not surprising since the convergence was poor with the mode displacement method for the ramp-loaded case (fig. 5.11). For this ramp-loaded case, the convergence of the bending moments improved dramatically when the mode acceleration method was used as can be seen in figure 5.14. Although convergence is improved for the step-loaded case by using the mode acceleration method (fig. 5.21), the convergence is still fairly poor.

Judging from the velocity time history in figure 5.7, it might be expected that including damping would improve the modal convergence of the response quantities. For the ramp-loaded, undamped case, the poorest convergence was for the accelerations (fig. 5.10). For the ramp-loaded case with discrete dampers, there is an improvement in modal convergence as seen in figure 5.22. For the case with 0.5 percent modal damping there is also a slight improvement in modal convergence. However, in neither case does the damping completely alleviate the poor convergence.

All the previous convergence results are at critical points located by the method described in chapter 3. When a different number of modes is used in the analysis, the critical time for a particular critical point usually shifts slightly. Consequently, the results for a given response quantity–critical point combination

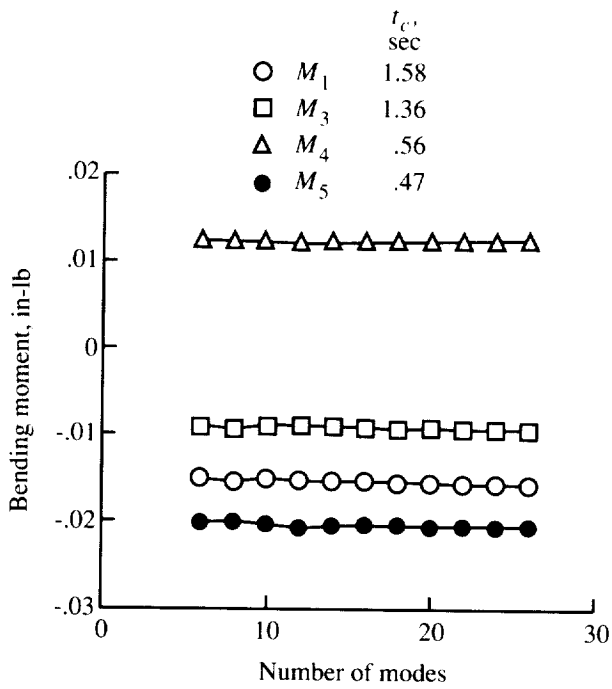


Figure 5.14. Modal convergence of selected bending moments for five-span beam. Ramp load; undamped beam; mode acceleration method.

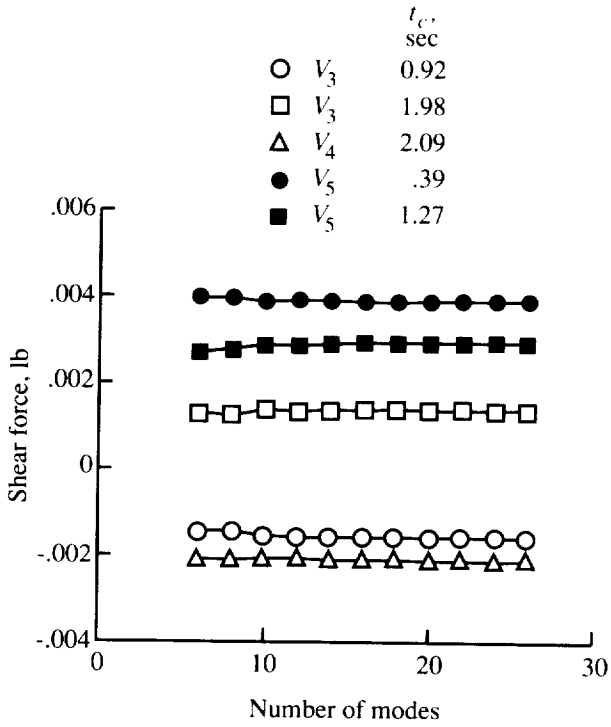


Figure 5.15. Modal convergence of selected shear forces for five-span beam. Ramp load; undamped beam; mode acceleration method.

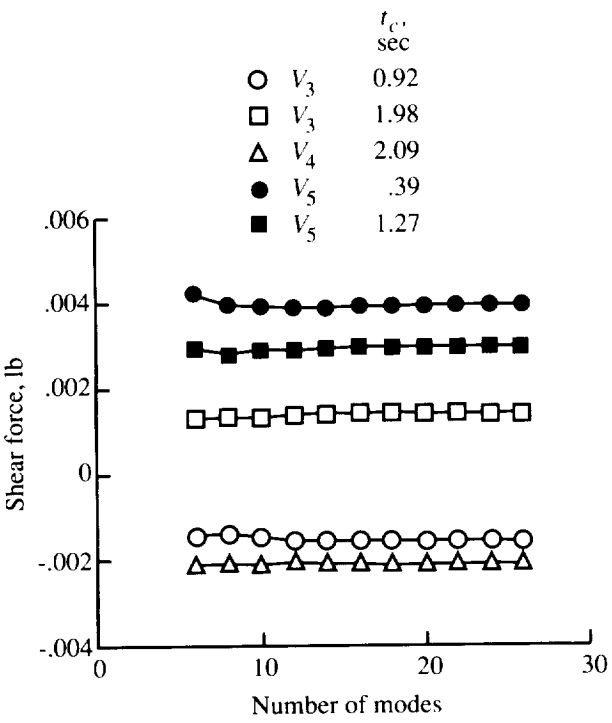


Figure 5.16. Modal convergence of selected shear forces for five-span beam. Ramp load; undamped beam; static mode method.

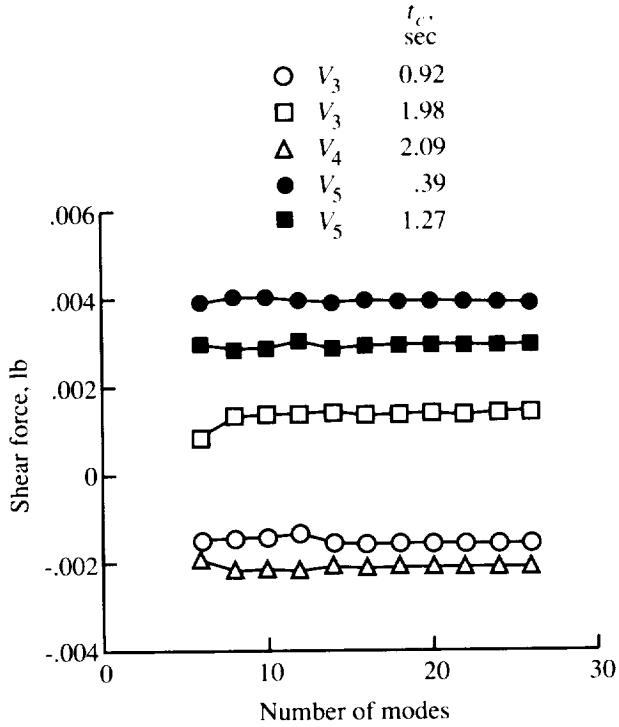


Figure 5.18. Modal convergence of selected shear forces for five-span beam. Ramp load; undamped beam; Ritz-Wilson-Lanczos method.

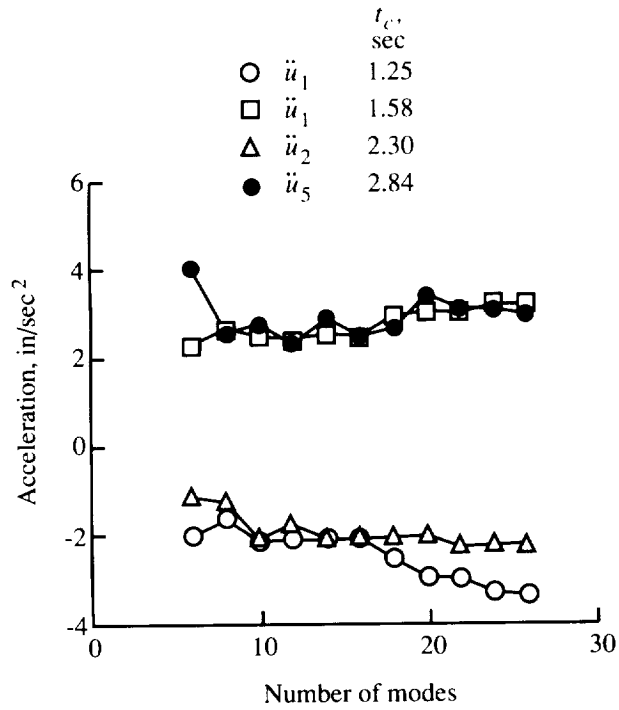


Figure 5.17. Modal convergence of selected accelerations for five-span beam. Ramp load; undamped beam; static mode method.

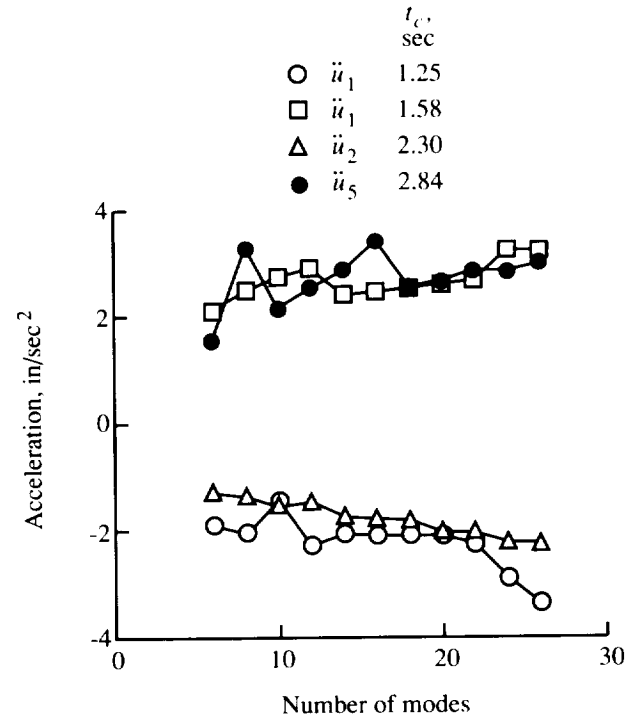


Figure 5.19. Modal convergence of selected accelerations for five-span beam. Ramp load; undamped beam; Ritz-Wilson-Lanczos method.

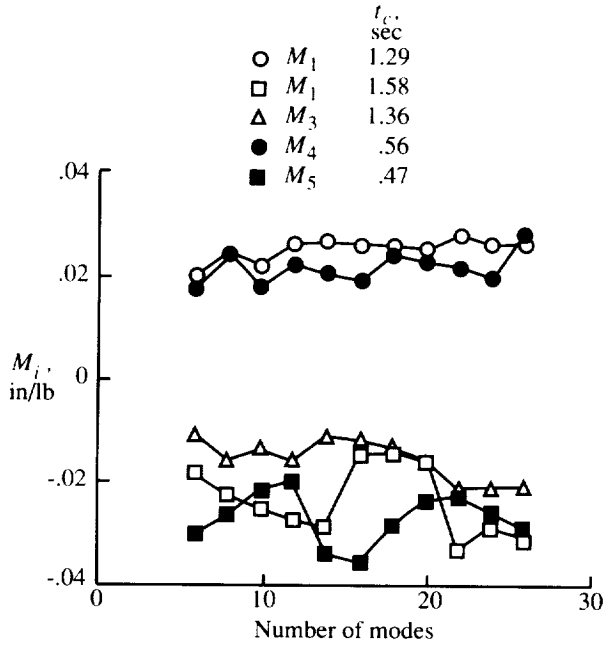


Figure 5.20. Modal convergence of selected bending moments for five-span beam. Step load; undamped beam; mode displacement method.

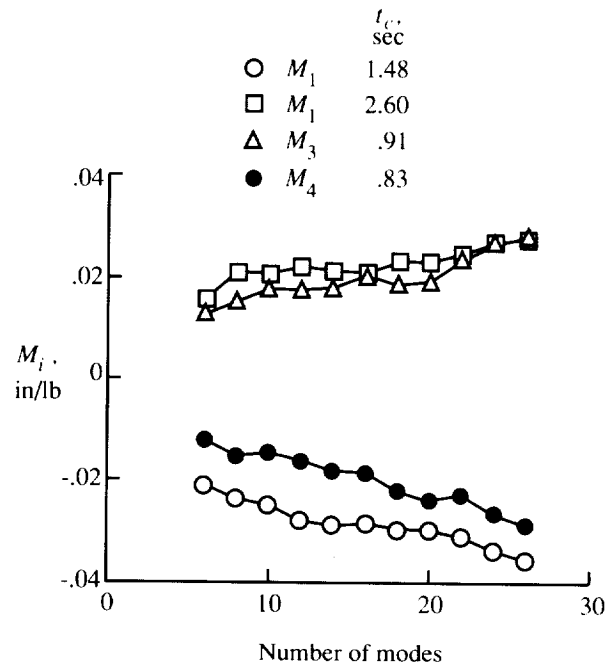


Figure 5.21. Modal convergence of selected bending moments for five-span beam. Step load; undamped beam; mode acceleration method.

occur at different times depending on the number of modes used in the analysis (the values shown in parentheses in the figures are for the most refined solution). It is natural to ask whether response quantities at fixed times plotted as a function of number of modes would show similar convergence. Figures 5.23 and 5.24 show the modal convergence of selected velocities and bending moments, respectively, at fixed times. The mode displacement method was used in the analyses. The particular response quantities and times were selected to span the range between largest positive and negative values. As can be seen in figure 5.23, the convergence of velocities is good. From figure 5.24, it can be seen that the convergence of the bending moments is poor and remarkably similar to the critical point convergence results (fig. 5.11). Thus it would appear that the critical point constraint formulation does not significantly affect the modal convergence of the response.

5.1.1.3. Mesh convergence. Table 5.2 shows the convergence of the lowest 10 frequencies as a function of the number of elements used to model each span of the beam. The convergence of these lower frequencies is rapid. The convergence of various response quantities as a function of the mesh is also a concern.

The modal convergence cannot be uncoupled from the mesh convergence. This was discussed in reference 33 for the derivatives of damping ratios

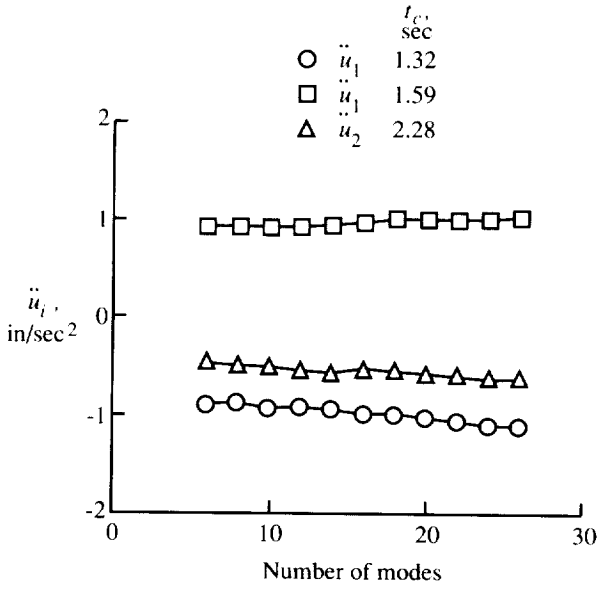


Figure 5.22. Modal convergence of selected accelerations for five-span beam. Ramp load; discretely damped beam; mode displacement method.

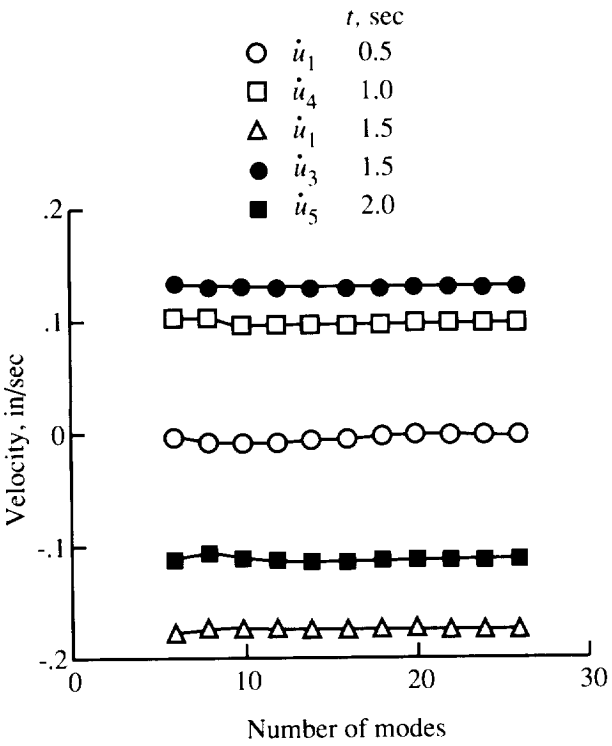


Figure 5.23. Modal convergence of selected velocities for five-span beam at fixed time points. Ramp load; undamped beam; mode displacement method.

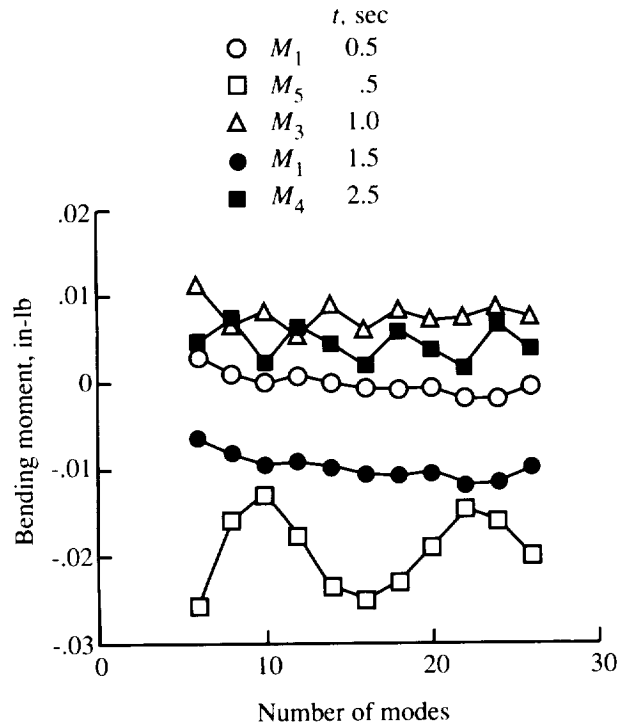


Figure 5.24. Modal convergence of selected bending moments for five-span beam at fixed time points. Ramp load; undamped beam; mode displacement method.

calculated with undamped vibration modes. For several cases, the modal convergence of the derivatives was poor. As the mesh was refined, convergence was achieved only when almost all the available modes were used in calculating the damping ratio. Clearly, this is an example where the modal basis provides a very poor approximation to the actual solution.

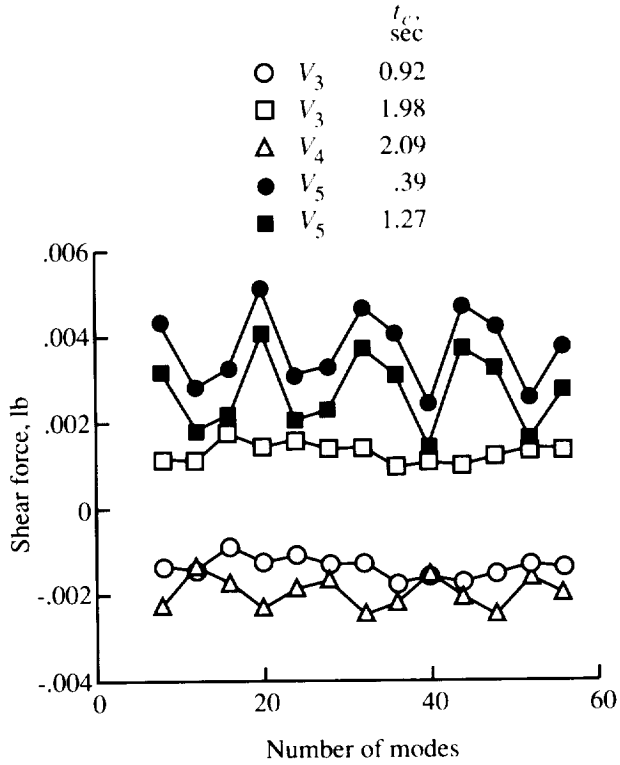


Figure 5.25. Modal convergence of selected shear forces for five-span beam modeled with six elements per span. Ramp load; undamped beam; mode displacement method.

Figure 5.25 shows the modal convergence of shear force for the five-span beam modeled with six elements per span and the transient analysis performed with the mode displacement method. The convergence for this case is just as poor as for the case with three elements per span shown in figure 5.12. However, a plot of convergence of this shear force as a function of the number of elements per span, when all modes are used in each analysis (figure 5.26), shows good convergence. Clearly, the convergence of shear forces for the ramp-loaded five-span beam is similar to that reported for derivatives of damping ratios in reference 33; the vibration modes are simply doing a poor job of representing the solution.

Figure 5.27, which shows the convergence of accelerations for the step-loaded beam as a function of elements per span, indicates a different behavior,

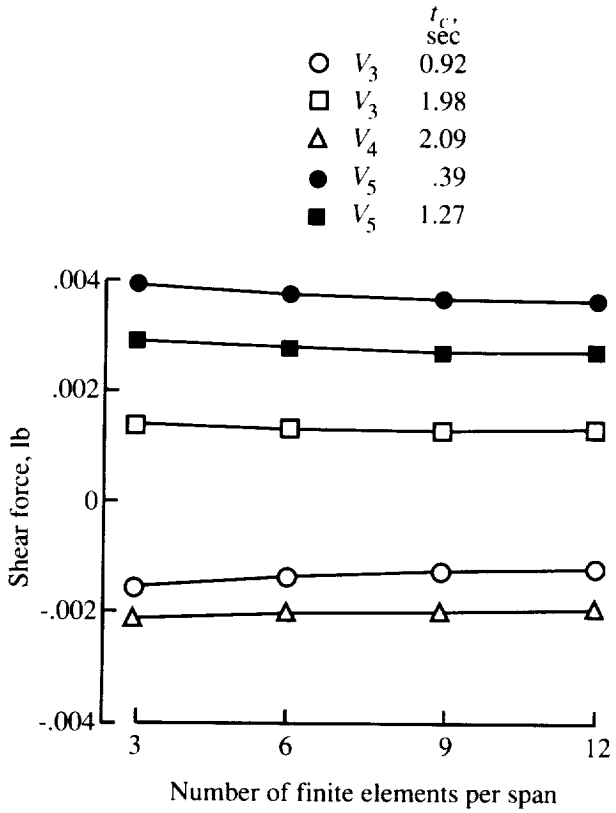


Figure 5.26. Convergence of selected shear forces as function of number of finite elements per span for five-span beam. Ramp load; undamped beam; mode displacement method.

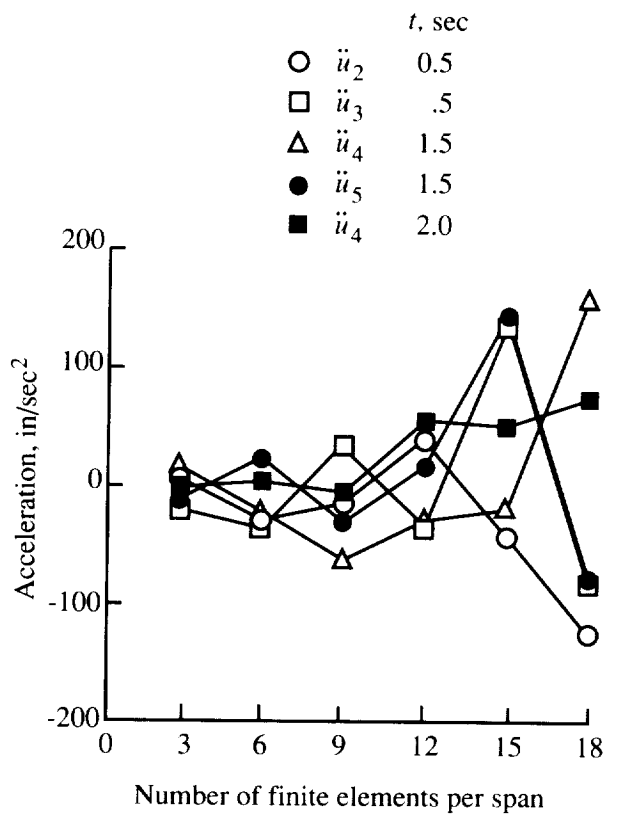


Figure 5.27. Convergence of selected accelerations as function of number of finite elements per span for five-span beam. Step load; undamped beam; mode displacement method.

however. Here, the very poor convergence is due to the higher frequency modes being excited by the step loading. As the mesh is refined, the number of high-frequency modes increases, and these continue to have a significant contribution to the acceleration.

In evaluating the accuracy of the sensitivity calculation procedures in section 5.1.2, particular attention must be paid to the convergence characteristics. Some convergence problems such as those caused by the use of vibration mode shapes can be improved by the use of alternate basis functions. However, other convergence problems, such as for the accelerations in the step-loaded case, are inherent in the problem definition.

5.1.2. Sensitivities of Beam Dynamic Response

In the previous section, the transient response of the five-span beam was considered in detail. In this section, the calculation of sensitivities of displacements, velocities, accelerations, and stresses with respect to various design variables is considered.

5.1.2.1. Design variables. Two different classes of design variables were considered. The first design

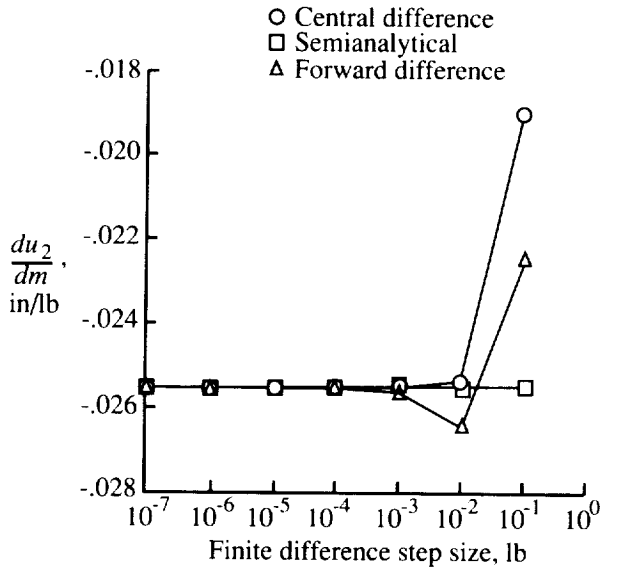


Figure 5.28. Effect of finite difference step size on accuracy of displacement derivative approximation with respect to point mass design variable. Ramp load; undamped beam.

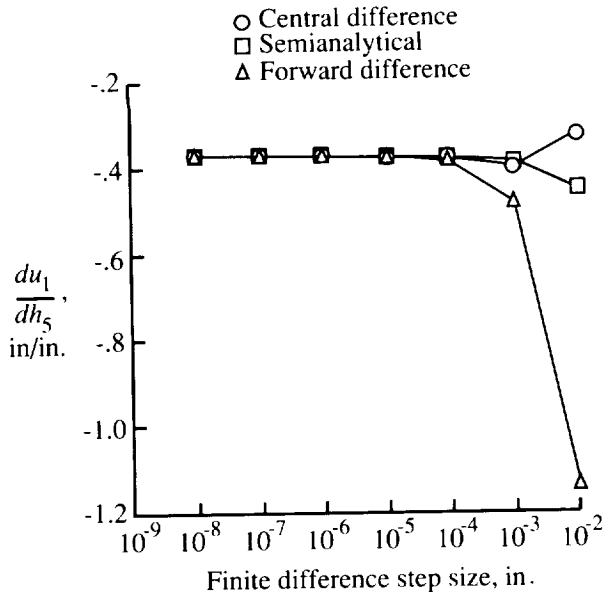


Figure 5.29. Effect of finite difference step size on accuracy of displacement derivative approximation with respect to thickness design variable. Ramp load; undamped beam.

variable is a concentrated mass m (initially zero) at the location of the translational damper. This design variable was also considered in reference 33. The derivatives of the system mass and stiffness matrices with respect to this design variable are constant and zero, respectively. As a consequence, the derivatives of the system matrices required in the semianalytical methods can be calculated exactly by a simple forward difference operator. The beam thicknesses in each of the five spans were also design variables. Derivatives with respect to the five thickness design variables showed similar characteristics. Herein, results for derivatives with respect to the thickness in the rightmost span h_5 along with derivatives with respect to the point mass m are presented.

5.1.2.2. Effect of finite difference step size. The methods described in chapter 4 for calculating sensitivities all rely on finite difference operators at some stage in the algorithm. The forward and central difference methods rely on the operators in equations (4.1) and (4.2) to calculate derivatives of response quantities. In the semianalytical methods, the derivatives of the system matrices are calculated with the forward difference operator in equation (4.1). In all cases the finite difference step size must be selected so that the operator provides a reasonable approximation to the derivative. If the step size is too large, the error due to truncating the series approximation of the derivative is large. If the step size is too small, the numerical condition error in performing the function evaluations (dynamic analyses) becomes large.

To assess the effect of step size on the calculation of sensitivities for the five-span beam, derivatives were calculated with the three methods with various step sizes. In this study the beam was undamped and the ramp loading was applied. All 26 vibration modes were included in the analysis. Figure 5.28 shows the estimated derivative of displacement u_2 at critical point 5 with respect to the point mass design variable m as a function of step size. As mentioned, the derivatives of the system matrices with respect to this design variable can be calculated exactly in the semianalytical method. As a result, the derivative estimated with the semianalytical method is approximately constant for the six-order-of-magnitude change in step size shown in the figure. The central difference method uses the higher order operator and provides good accuracy over most of the step size range shown in the figure. The forward difference operator provides good accuracy with the smaller step sizes but begins to diverge earlier for the larger step sizes than the central difference method.

Figure 5.29 shows the estimated derivative of displacement u_1 at critical point 5 with respect to the rightmost span thickness h_5 as a function of step size. In this case, the system mass matrix is a linear function of this design variable and its derivative can be represented exactly by the forward difference operator. The system stiffness matrix is a cubic function of this design variable and its derivative can only be approximated by the forward difference operator. Still, the derivative approximation computed by the semianalytical method is very accurate except for the largest step size and is no worse for this case than the much more costly central difference method. Again, the forward difference operator results in substantial errors for the larger step sizes.

Because this example has a relatively small number of degrees of freedom, there is little condition error when small step sizes are used. To assess the effects of condition error which would occur for larger problems, the derivative approximations for the five-span beam problem were also calculated with 32-bit floating point precision compared with the 60-bit precision used in the studies described above. The estimated derivative of displacement u_2 at critical point 5 with respect to the point mass is plotted as a function of finite difference step size in figure 5.30. Derivative approximations are calculated using the semianalytical method, the central difference method, and the forward difference method. For the larger step sizes, the results from all three methods agree well with those calculated with 60-bit precision. For step sizes smaller than 10^{-4} in., there is

considerable error in the derivative approximations calculated with the forward and central difference methods. The derivative approximations calculated with the semianalytical method, however, are highly accurate over the entire range of step sizes shown in the figure.

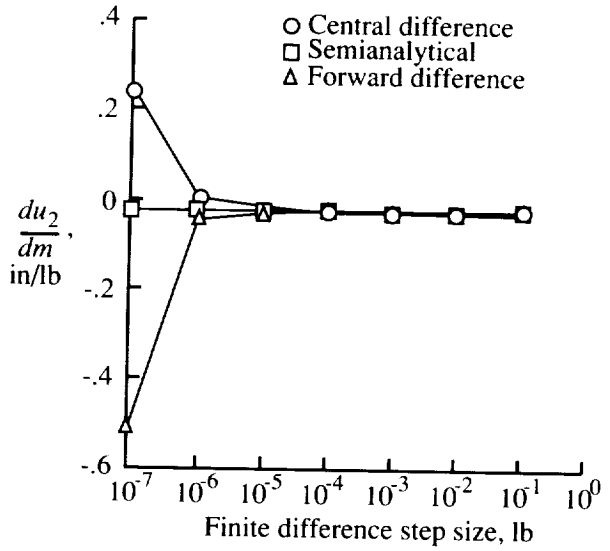


Figure 5.30. Effect of finite difference step size on accuracy of displacement derivative with respect to point mass design variable. Calculations performed with 32-bit precision; ramp load; undamped beam.

5.1.2.3. Modal convergence of sensitivities. The first case considered is the undamped beam with the ramp load. Figure 5.31 shows the convergence of selected estimated derivatives of displacements with respect to m at various critical points. The mode displacement method was used and the derivative approximations were calculated with the central difference operator with updated modes. The convergence is good although slightly poorer than the convergence of the displacements themselves (fig. 5.8). The convergence of the estimated displacement derivatives with respect to the thickness design variable is similar.

Although the modal convergence of the velocities for this case is good (fig. 5.9), the convergence of selected estimated derivatives of velocity with respect to m is generally poor (fig. 5.32). Given the poor convergence of the accelerations shown in fig. 5.10, it is not surprising that the convergence of the sensitivities of the accelerations is also very poor. From figure 5.33 it can be seen that the derivative approximations of the four selected critical point accelerations with respect to the thickness design variable are essentially not converging with increasing number of modes. It should be pointed out again that

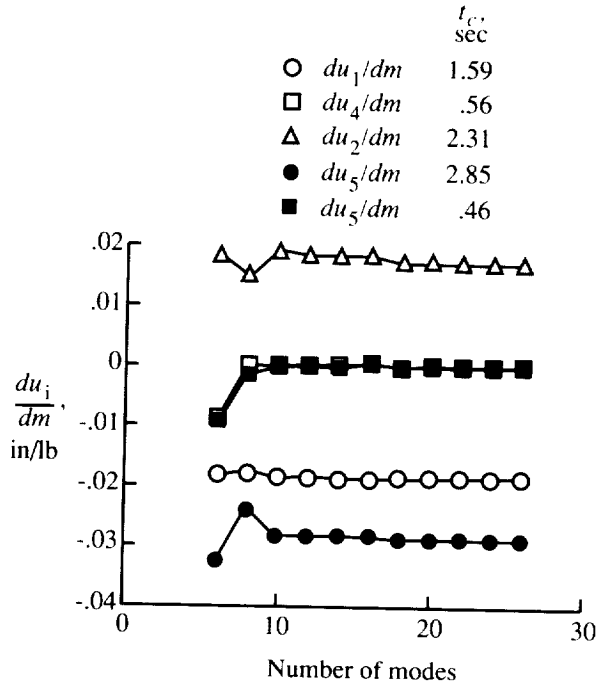


Figure 5.31. Modal convergence of derivatives of selected displacements with respect to mass design variable. Ramp load; undamped beam; mode displacement method; central difference operator.

these derivative approximations of velocity and acceleration are calculated with the central difference method and updated mode shapes; thus, the numerical errors are minimized. The poor convergence exhibited in figures 5.32 and 5.33 is due to the poor approximation of the sensitivities by the mode shapes.

Similar modal convergence behavior is observed for sensitivities of the stress resultants. This is consistent with the poor convergence of the stress resultants calculated with the mode displacement method (figs. 5.11 and 5.12). Figure 5.34 shows the poor convergence of derivative approximations of selected bending moments with respect to the thickness design variable. It can be seen that the convergence of the bending moment derivative approximation in the rightmost span with respect to the thickness in the rightmost span (dM_5/dh_5) is especially poor.

It was shown in the previous section that several approaches are available for overcoming the poor convergence of bending moments and shear forces in this beam example. The mode acceleration, static mode, and RWL methods all produced good modal convergence of bending moments and shears as shown in figures 5.14, 5.15, 5.16, and 5.18. Unfortunately this type of dramatic improvement does not occur for the sensitivities of the stress resultants. Figure 5.35 shows the convergence of the bending moment derivative approximations with respect to

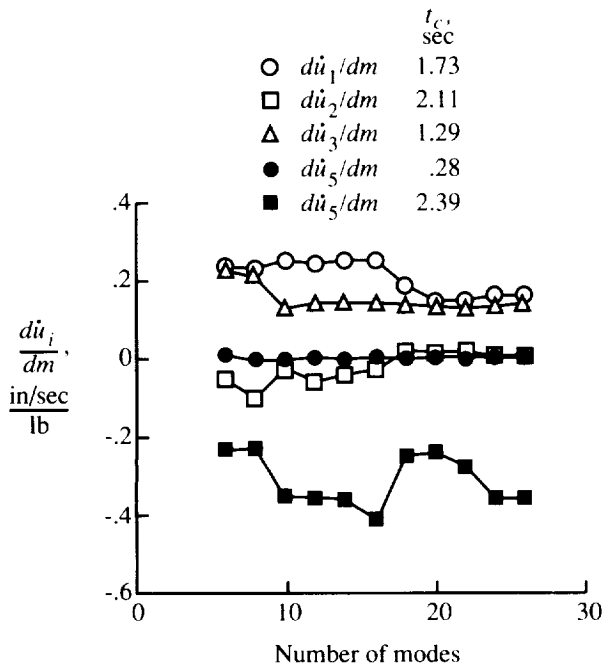


Figure 5.32. Modal convergence of derivative approximations of selected velocities with respect to mass design variable. Ramp load; undamped beam; mode displacement method; central difference operator.

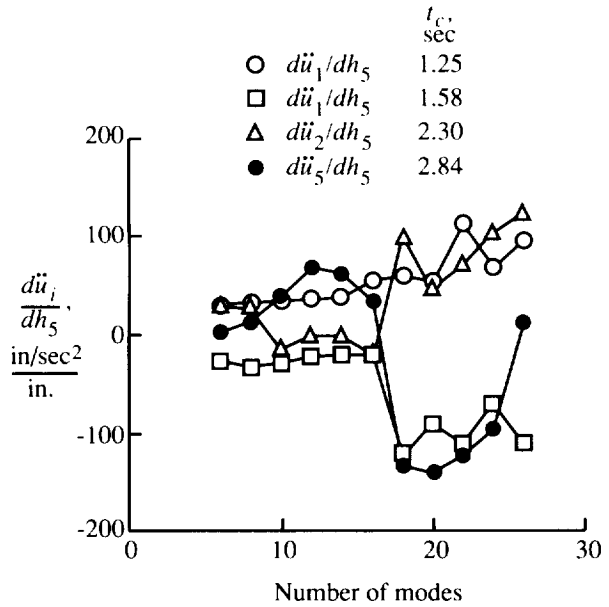


Figure 5.33. Modal convergence of derivative approximations of selected accelerations with respect to thickness design variable. Ramp load; undamped beam; mode displacement method; central difference operator.

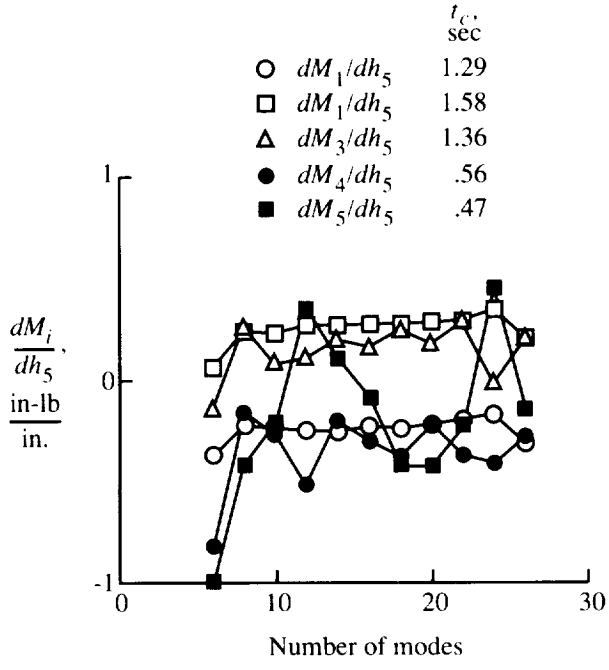


Figure 5.34. Modal convergence of derivative approximations of selected bending moments with respect to thickness design variable. Ramp load; undamped beam; mode displacement method; central difference operator.

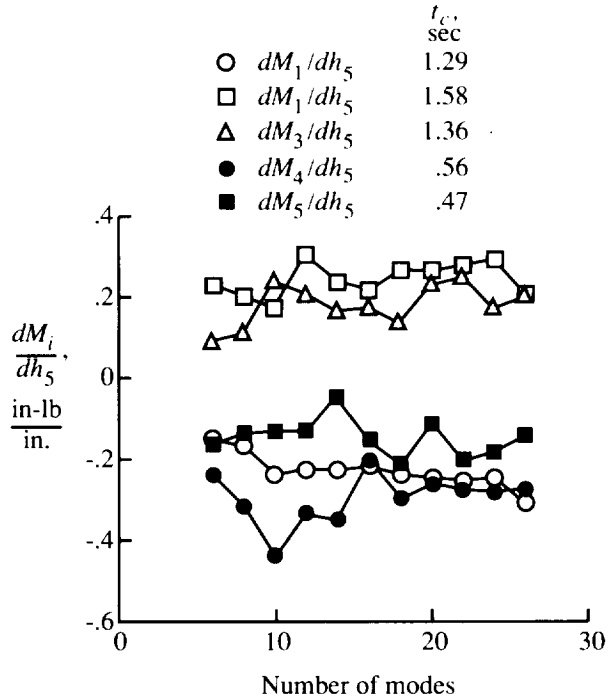


Figure 5.35. Modal convergence of derivative approximations of selected bending moments with respect to thickness design variable. Ramp load; undamped beam; RWL method; central difference operator.

the thickness design variable where the analysis was done with the RWL method. As in the studies discussed that used the mode displacement method, the sensitivities were calculated by the central difference operator with the basis vectors updated for the perturbed design. Convergence of dM_5/dh_5 is somewhat improved compared with the mode displacement case. Other quantities show convergence similar to the mode displacement case; none of these convergence histories can be described as good. Convergence of the shear force derivative approximations with the RWL method is considerably worse than for the bending moments.

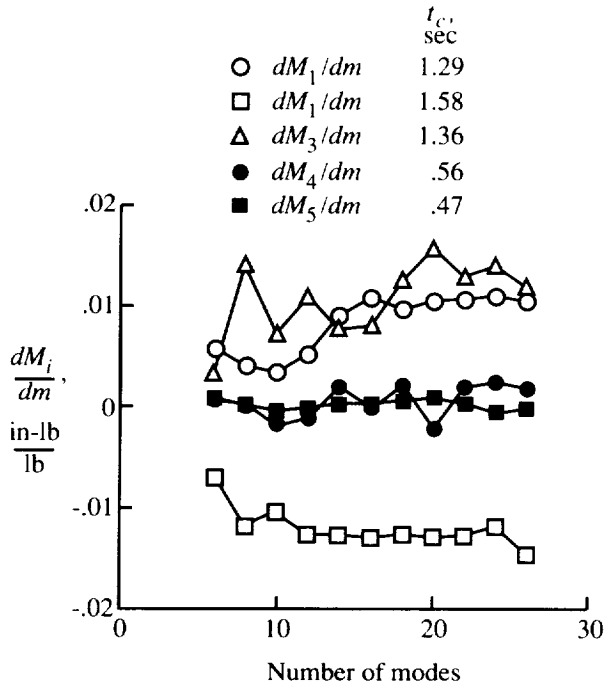


Figure 5.36. Modal convergence of derivative approximations of selected bending moments with respect to mass design variable. Ramp load; undamped beam; RWL method; semianalytical formulation.

The semianalytical methods have also been used for calculating sensitivities of stress resultants. Figure 5.36 shows the convergence of bending moment derivative approximations with respect to the mass design variable calculated with the fixed-mode, semianalytical method and RWL vectors. The convergence is very similar, especially for larger numbers of modes, to that of the central difference method. The mode acceleration, semianalytical method, and the semianalytical method with approximate $d\Phi/dx$ were also tried. Again, the modal convergence curves had the same jaggedness as for previous cases.

Considering these difficulties with modal convergence for the ramp-loaded cases, especially poor convergence would be expected for the step-loaded case. For the ramp-loaded case, the convergence of displacement derivative approximations with respect to the mass design variable was reasonably good (fig. 5.31). For the step-loaded case, the modal convergence of the same displacement sensitivities is poor as shown in figure 5.37. The modal convergence of higher order sensitivities (velocities, accelerations, and stresses) is extremely poor.

Adding damping slightly improved the modal convergence of the response quantities but did not completely alleviate the convergence problem. The result for sensitivities is similar. Figure 5.38 shows the convergence of velocity sensitivities for the discretely damped case. The convergence is slightly improved over the undamped case shown in figure 5.32, but the curves are still fairly jagged. Convergence of sensitivities for the case with modal damping is also very similar to that in the undamped case.

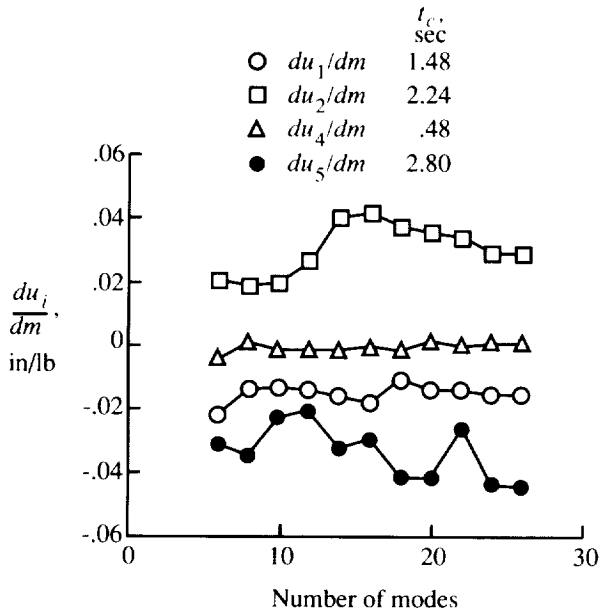


Figure 5.37. Modal convergence of derivative approximations of selected displacements with respect to mass design variable. Step load; undamped beam; mode acceleration method; semianalytical method.

5.1.2.4. Mesh convergence of sensitivities. Just as for the response quantities, additional insight can be obtained by considering the convergence of their sensitivities with increasing number of elements per bay. The case of the stress resultants will be considered, since they were shown to converge well with mesh refinement (fig. 5.26) but poorly as a function

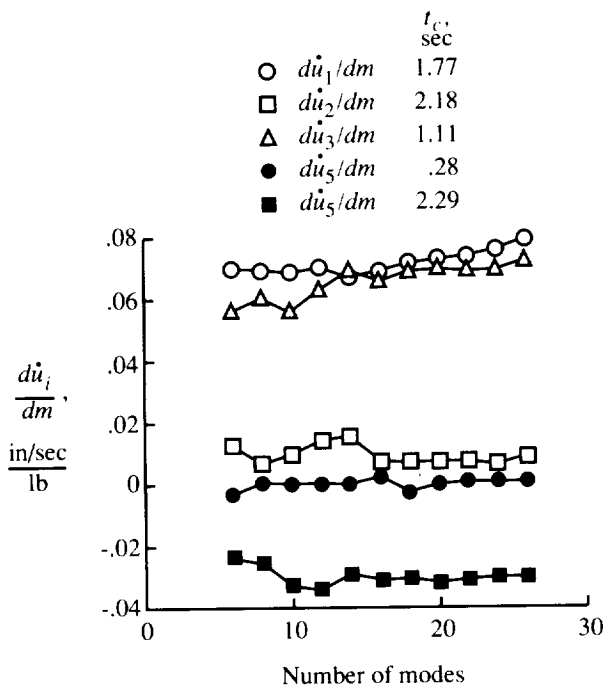


Figure 5.38. Modal convergence of derivative approximations of selected velocities with respect to mass design variable. Ramp load; discretely damped beam; mode acceleration method; semianalytical method.

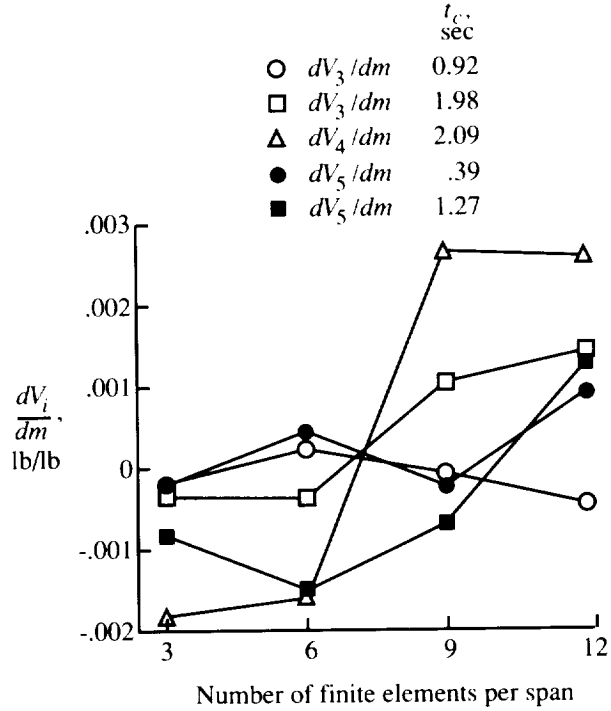


Figure 5.39. Convergence of derivative approximations of selected shear forces with respect to mass design variable as function of number of elements per span. Ramp load; undamped beam; mode displacement method; central difference operator.

of number of vibration modes used in the analysis. Figure 5.39 shows the convergence of derivative approximations of shear force with respect to the mass design variable as a function of number of elements per bay. Surprisingly, the convergence is extremely poor.

5.1.2.5. Fixed versus updated modes in sensitivity calculations. As mentioned, the computational cost of updating the vibration modes for the perturbed analyses is substantial. The question of whether the modes from an initial design can be used in a finite-difference-based procedure to calculate sensitivities of the transient behavior has received considerable attention in the literature. In reference 35, it was shown that there is a substantial difference in the derivatives of aircraft flutter speeds when fixed modes are used rather than the updated modes. In reference 33, however, there was little difference in the derivatives of damping ratios for the five-span beam when either fixed or updated modes were used. This was investigated here with the same five-span, undamped beam under the step load. As shown in figure 5.37 where the derivative approximations were calculated with the semianalytical, mode acceleration method, convergence with respect to the number of modes is very slow. Figure 5.40 shows the modal convergence of derivative approximations of selected displacements with respect to h_5 calculated with forward difference procedures. Results with both fixed and updated vibration modes are shown. Again, the convergence as a function of number of modes is poor. However, for all three derivative approximations, the results are nearly the same for both the fixed mode and updated mode cases.

5.2. Composite Delta Wing Example

The second example considered is an aircraft delta wing with laminated composite cover skins taken from reference 36 and described in detail in reference 37. The finite element model of this structure is shown in figure 5.41. Since the wing is geometrically symmetric about the midplane through its thickness (X-Y plane), only the upper half of the wing is modeled, and boundary conditions enforcing antisymmetric motion are imposed on the joints lying in the X-Y plane. The wing is also cantilevered at the root. The model contains a total of 88 joints with a total of 140 unconstrained degrees of freedom. The webs in the wing are made of titanium and are modeled with 70 shear panel finite elements along with rod elements through the thickness of the wing. The cover skins are made of a moderate-modulus

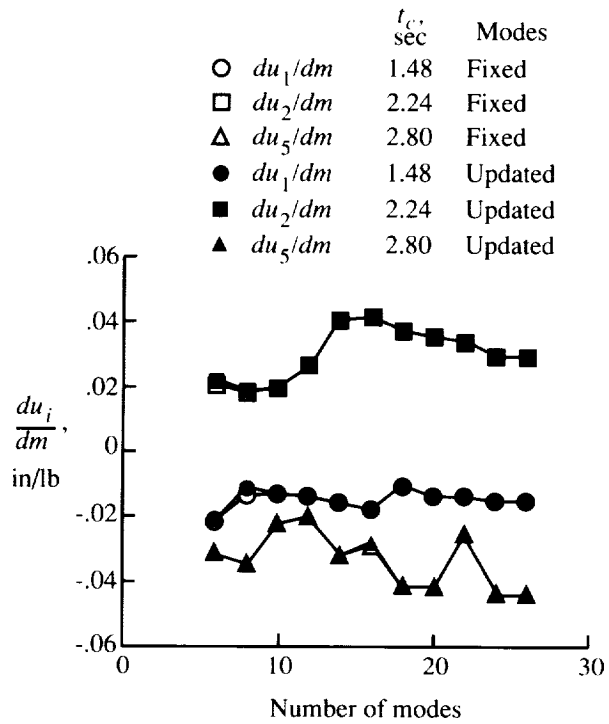


Figure 5.40. Modal convergence of derivative approximations of selected displacements with respect to mass design variable calculated with both fixed and updated vibration modes. Step load; undamped beam; forward difference methods.

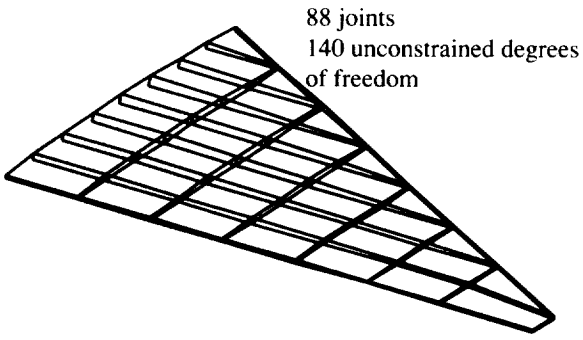


Figure 5.41. Finite element model of composite delta wing.

($E_1 = 21 \times 10^6$ psi), graphite-epoxy material with 0°, ±45°, and 90° lamina where the 0° material runs spanwise along the wing. These cover skins are modeled with membrane finite elements; thus, only the total thicknesses (and not the stacking sequence of plies) of each lamina are important. The structural mass is 6003 lb, but most of the wing mass is due to a fuel mass of 93 650 lb distributed over the joints. The spatial distribution of the load is the same as the static load from reference 37 and is roughly equivalent to a 144-psf pressure load on the wing skin. A step loading function was used as the time function for all cases. The lowest 10 vibration frequencies for the wing are shown in table 5.3. Damping is accounted for by assuming 0.5 percent of critical damping for all modes.

Table 5.3. Lowest 10 Vibration Frequencies for Delta Wing

Mode	Frequency, Hz
1	2.055
2	2.765
3	4.104
4	4.913
5	5.920
6	6.944
7	7.451
8	8.421
9	9.583
10	9.880

5.2.1. Wing Dynamic Response

The character of the dynamic response of the delta wing is considerably different than that of the five-span beam. Shown in figure 5.42 is a time history of acceleration at the wingtip. Although 64 modes were included in the analysis, it is evident from figure 5.42 that only the low-frequency modes are being excited. The same is true for stresses as shown in the time history of figure 5.43. Shown in figure 5.43 is τ_w^6 which is a typical shear stress in a web. As can be seen, there is a small amount of higher frequency response superimposed on the predominant response frequency. However, the time history exhibits none of the high-frequency response present in the five-span beam.

In contrast to the five-span beam example, the modal convergence of all response quantities considered for the delta wing is quite good. Shown in figures 5.44 and 5.45 are modal convergence plots for selected accelerations and stresses at critical points

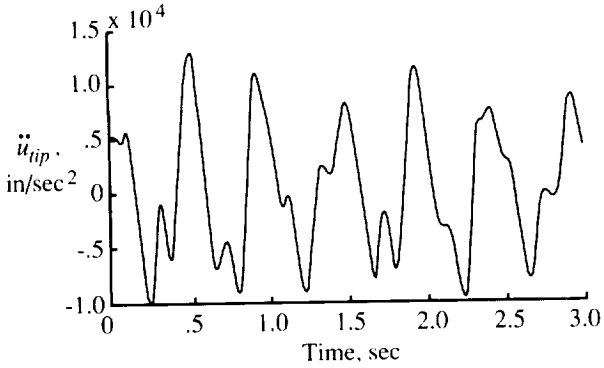


Figure 5.42. Time history of tip acceleration for delta wing.

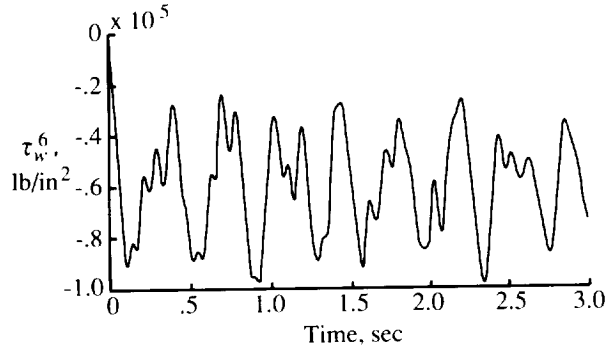


Figure 5.43. Time history of shear stress in web in region 6 for delta wing.

calculated with the mode displacement method. A converged solution is reached with approximately 20 or less modes for all the response quantities shown. Convergence is also good for response quantities when the mode acceleration or RWL methods are used instead of the mode displacement method. Shown in figure 5.46 is a convergence plot for the same stresses shown in figure 5.45 but calculated with RWL vectors.

5.2.2. *Sensitivities of Wing Dynamic Response*

5.2.2.1. Design variables. The design variable definitions are the same as those in reference 37 and are shown in figure 5.47. As can be seen in figure 5.47, the skin is broken up into 16 regions. In each region there are three design variables— the thickness of the 0° lamina, the thickness of the 90° lamina, and the thickness of the ±45° lamina. These design variables will be denoted t_θ^i where i denotes the region of the wing skin, and θ is either 0°, 90°, or 45° depending on the lamina orientation. Also shown in figure 5.47 are the 12 design variables controlling the thickness of the webs. These will be denoted t_w^i where i denotes the particular web region.

In calculating sensitivities of various response quantities, only a small subset of these design variables were considered. Specifically, derivatives of selected displacement, velocity, acceleration, and stress quantities were calculated with respect to t_0^2, t_{90}^{13}, t_0^{16}, t_{90}^{16}, t_w^6, and t_w^{10}.

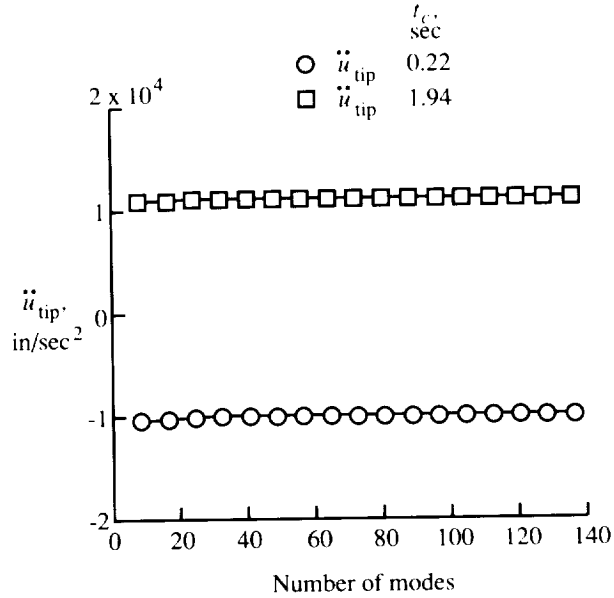


Figure 5.44. Modal convergence of tip accelerations for delta wing.

5.2.2.2. Effect of finite difference step size. Compared with the five-span beam example, the system matrices for the delta wing are larger and have a more complicated connectivity. Since many of the significant operations in the transient response analysis operate directly on these matrices, there is considerable potential for accumulating round-off error. This round-off error along with the truncation error in the finite difference expressions is a concern in selecting a step size for a finite difference approximation to a derivative.

A study was performed to consider the effect of step size in the forward difference and central difference methods for the delta wing. Figure 5.48 shows derivative approximations of the wingtip acceleration at critical points with respect to selected thickness design variables as a function of the finite difference step size used. As seen in the figure, the step size was varied by factors of 10 from 10^{-7} to 10^{-2}. The central difference method was not used with the 10^{-2} step size because the backward perturbation from the nominal design would result in negative member thickness. One significant observation is that the acceptable step size range for the forward difference method is small—approximately 2 decades. A second observation is that the behavior of the central difference method as a function of step size is surprisingly good. It is expected that, for larger step sizes

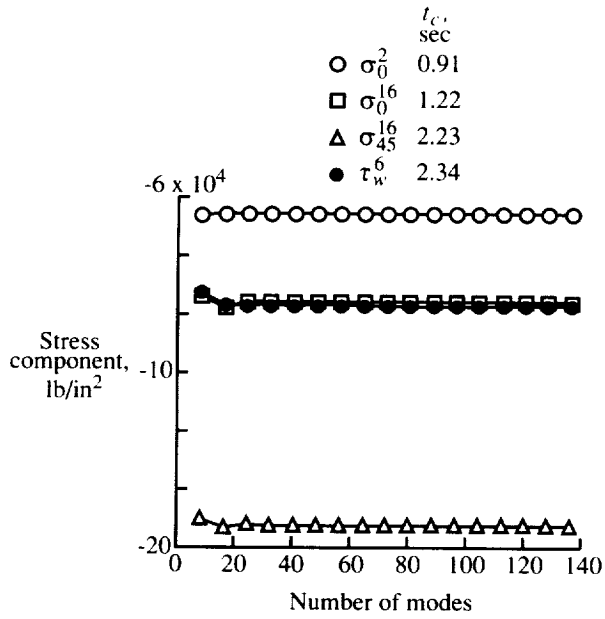


Figure 5.45. Modal convergence of selected stresses for delta wing calculated with mode displacement method.

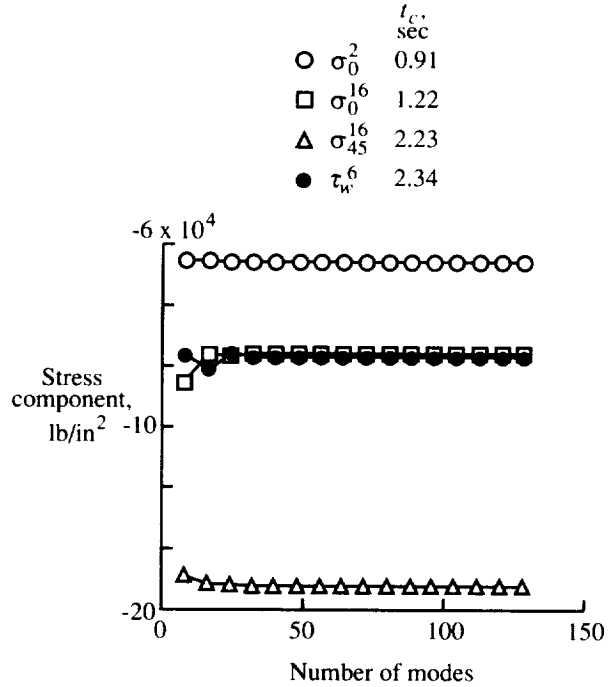


Figure 5.46. Modal convergence of selected stresses for delta wing calculated with RWL method.

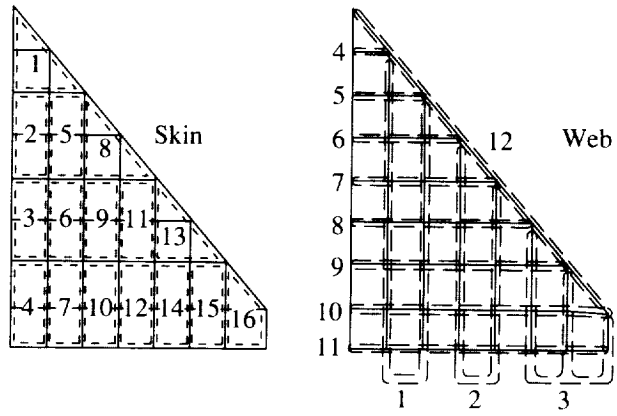


Figure 5.47. Design variable definitions for delta wing example.

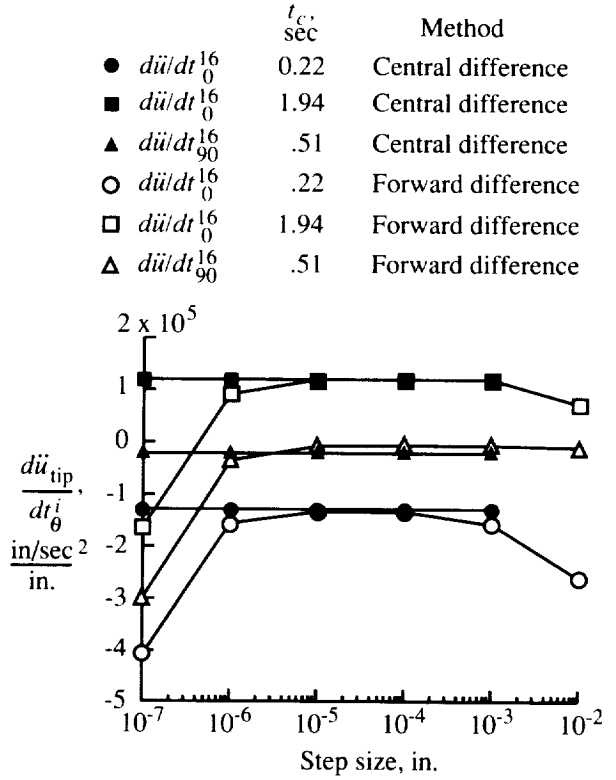


Figure 5.48. Effect of finite difference step size on accuracy of tip displacement derivative approximations calculated with forward and central difference methods with fixed modes. Delta wing example.

(10^{-3}), the central difference method results in less error than the forward difference method. The unexpected superior performance of the central difference method for the smaller step sizes is probably due to the symmetry of the difference operator. The round-off errors that occur with the positive and negative perturbations tend to cancel each other and thus produce the better than expected accurate values for the sensitivities.

The situation is similar for selected stress sensitivities shown in figure 5.49. Most of the curves for the forward difference case have a small acceptable step size range. This is especially obvious for the derivative $d\sigma_{45}^{16}/dt_{90}^{16}$ where 10^{-5} is the apparent choice for step size. It should also be mentioned that these calculations were performed by using 64-bit arithmetic. In the five-span beam example, the effect of step size on displacement derivatives was not as severe even though, for one case, these were calculated with predominantly 32-bit arithmetic (fig. 5.30).

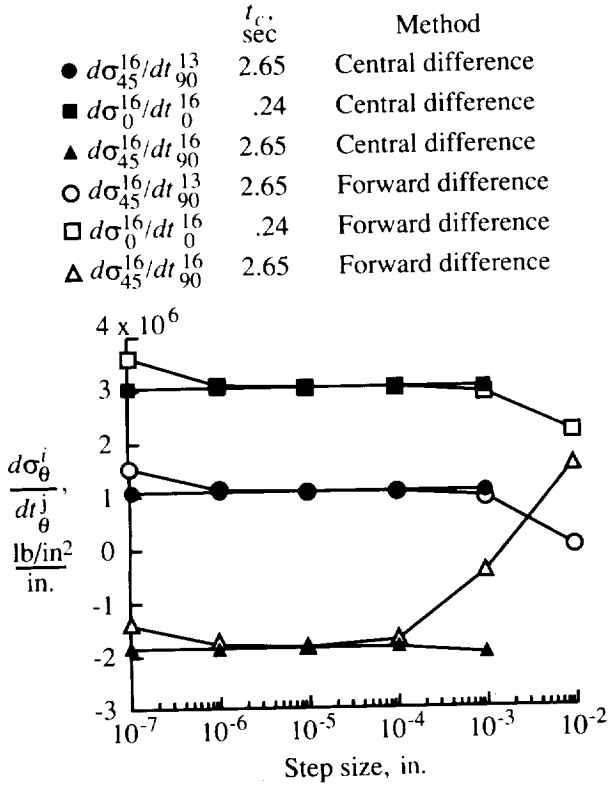


Figure 5.49. Effect of finite difference step size on accuracy of stress derivative approximations calculated with forward and central difference methods with fixed modes. Delta wing example.

The simple approach of selecting a single step size for use with all response quantities and all design variables was used here. This approach has the obvious advantage of simplicity but very questionable validity for the forward difference method and this delta wing example. From figure 5.49, there is significant error in $d\sigma_{45}^{16}/dt_{90}^{16}$ if greater than 10^{-4} is used as the step size. However, if less than 10^{-5} is used instead, $d\ddot{u}_{tip}/dt_0^{16}$ is in error.

As noted the central difference method improves the range of acceptable step sizes but at the cost of an additional analysis for each design variable. Alternatively, the semianalytical method is particularly attractive for this delta wing example. The stiffness matrices of the membrane and shear panel finite elements are linear functions of the thickness design variables. Thus, large values of the step size can be used to effectively eliminate the round-off errors in generating the derivative approximations of the stiffness and mass matrices required for the semianalytical method.

5.2.2.3. Modal convergence of sensitivities. Unlike the five-span beam example, the modal convergence of the displacement, velocity, and acceleration derivatives for the delta wing example is good. As an example, consider the reference case of acceleration sensitivities calculated with the central difference method with updated modes shown in figure 5.50. For all derivative approximations, convergence is achieved with 32 or less modes. Modal convergence for acceleration sensitivities is equally good when the simple forward difference method with fixed modes is used as shown in figure 5.51. Figure 5.52 shows the convergence of acceleration sensitivities calculated with the semianalytical method with RWL vectors instead of vibration modes. Convergence is also good although slightly poorer than when modes are used. For example, approximately 40 RWL vectors are required for a converged value of $d\ddot{u}_{tip}/dt_0^{16}$ compared with approximately 32 vibration mode shapes.

The modal convergence of stress derivatives, however, depends dramatically on whether fixed or updated modes are used in the calculation. The reference case with the central difference operator uses updated modes, and as shown in figure 5.53, the modal convergence for all stress sensitivities is very good. Also the convergence of the stress sensitivities with the forward difference operator with updated modes as shown in figure 5.54 is very good with 24 or less modes yielding a converged solution. However, when the forward difference operator with fixed modes is used the modal convergence of the stress sensitivities is very poor as shown in figure 5.55. For one derivative approximation, $d\sigma_{45}^{16}/dt_w^6$, the convergence is fairly good with approximately 24 modes yielding a converged solution. Especially poor convergence is observed for $d\sigma_0^{16}/dt_0^{16}$ (derivative of stress in the lamina with respect to its own thickness)

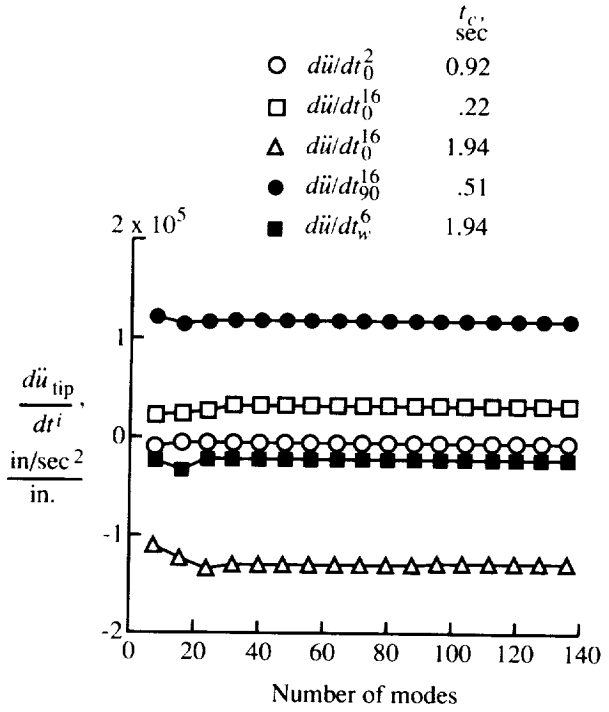


Figure 5.50. Modal convergence of tip acceleration sensitivities for delta wing calculated with central difference method.

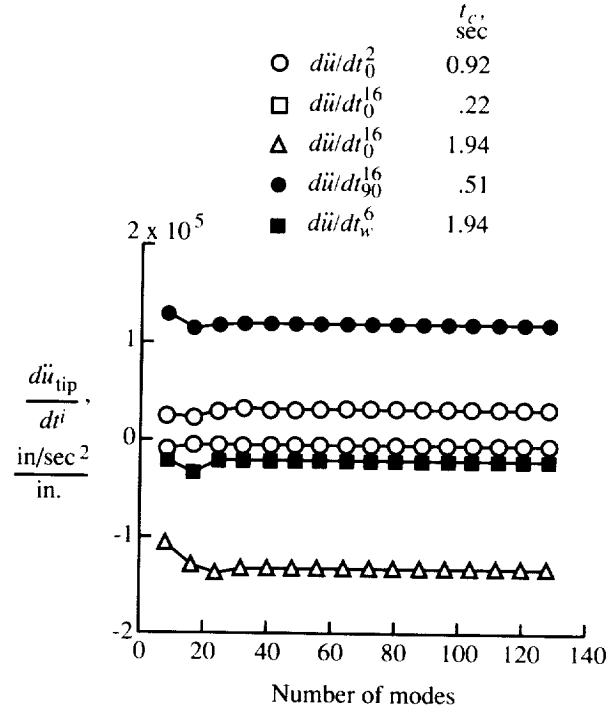


Figure 5.51. Modal convergence of tip acceleration sensitivities for delta wing calculated with forward difference method with fixed modes.

where approximately 100 modes are required for convergence.

Using the semianalytical method with fixed modes does not improve the modal convergence of the stress sensitivities. Figure 5.56 shows the modal convergence of the same stress sensitivities as in the previous figures but calculated with the semianalytical method with RWL vectors. The convergence behavior for each derivative approximation here is very similar to that for the forward difference method with fixed modes.

However, when the basis vectors are assumed to vary with the design variables and the modified modal method (see section 4.2.2) is used to approximate $d\Phi/dx$, the results are significantly improved. Figure 5.57 shows the modal convergence of the same stress derivative approximations as shown in previous figures. Here, the convergence is good with only around 24 modes required for convergence of the stress sensitivities.

It was mentioned in chapter 4 that the modal method for approximating $d\Phi/dx$ produces no improvement in the values of transient response sensitivities. This implies that including the modes in the modified modal method (see eq. (4.10)) may also not significantly improve the transient response sensitivities. This implication was tested by studying the modal convergence of the stress

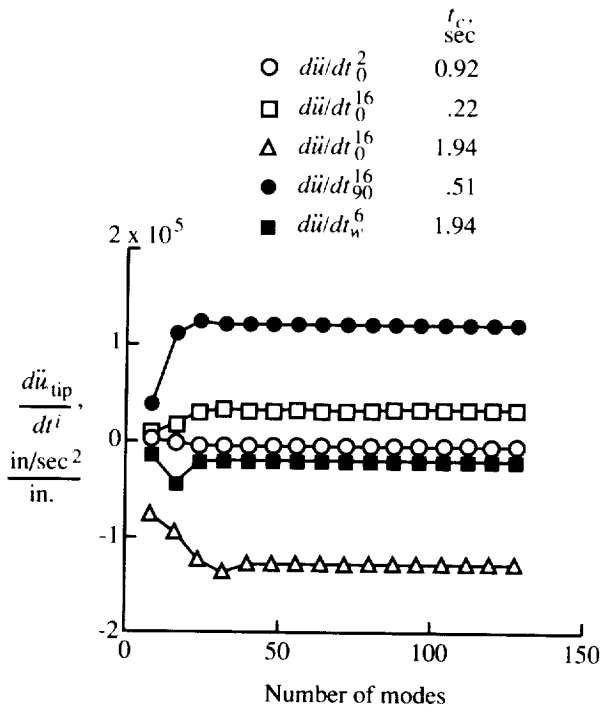


Figure 5.52. Modal convergence of tip acceleration sensitivities for delta wing calculated with semianalytical method with RWL vectors.

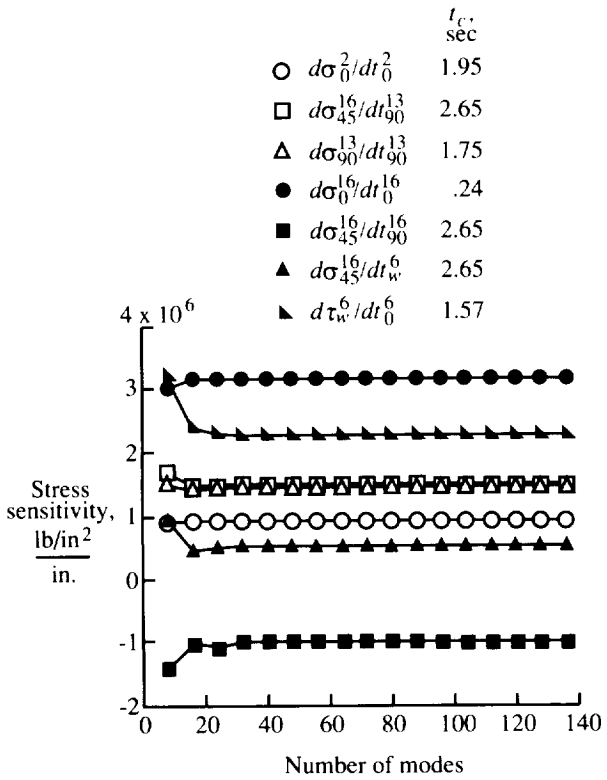


Figure 5.53. Modal convergence of selected stress sensitivities for delta wing calculated with central difference method.

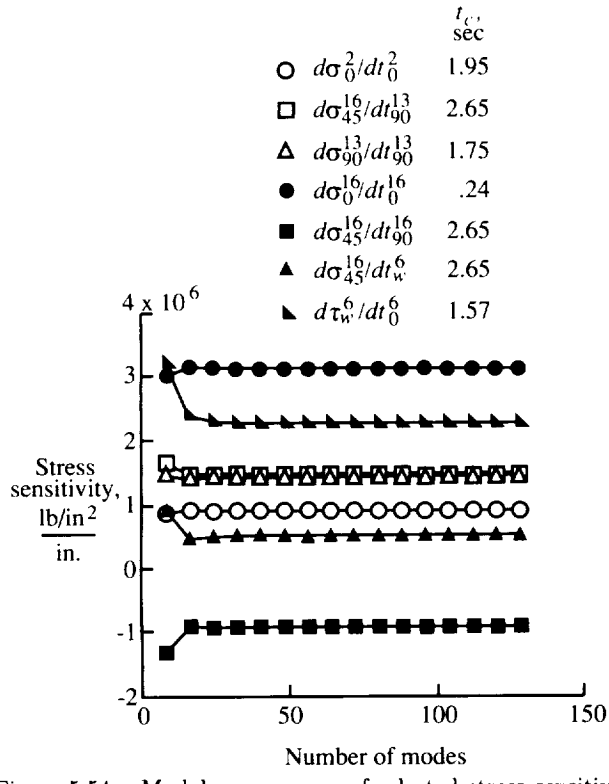


Figure 5.54. Modal convergence of selected stress sensitivities for delta wing calculated with forward difference method with updated modes.

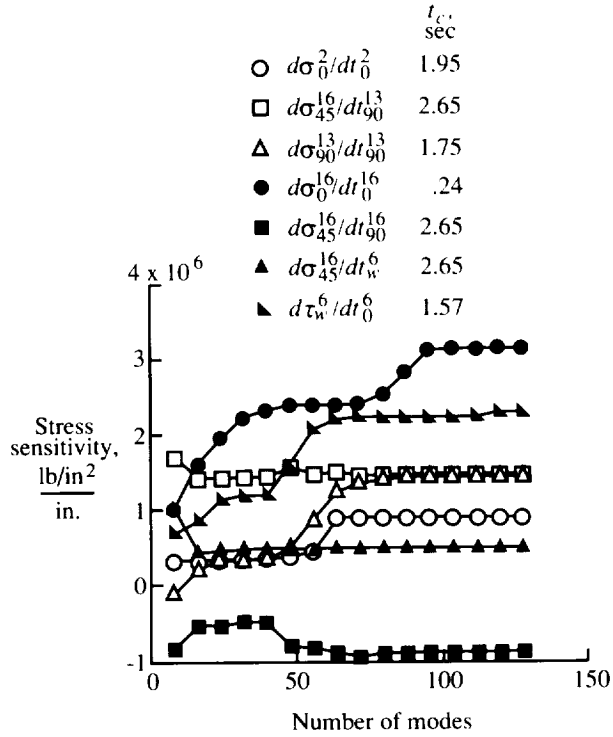


Figure 5.55. Modal convergence of selected stress sensitivities for delta wing calculated with forward difference method with fixed modes.

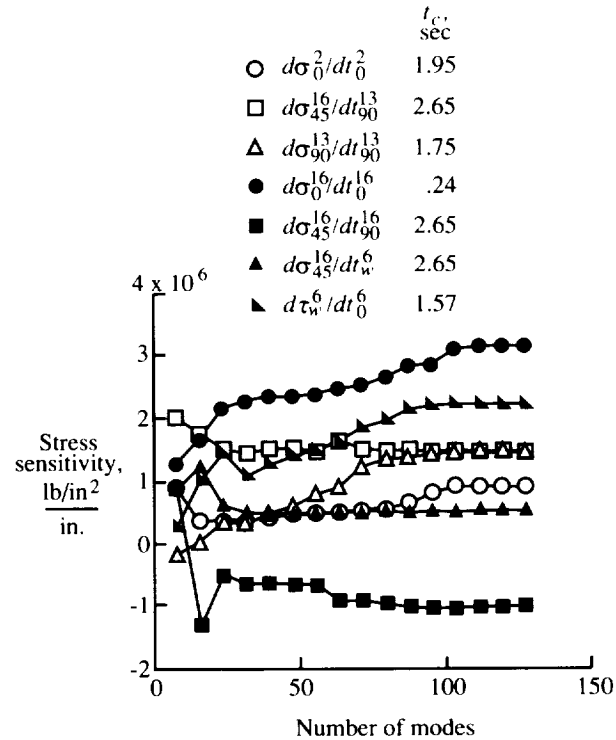


Figure 5.56. Modal convergence of selected stress sensitivities for delta wing calculated with semianalytical method with RWL vectors.

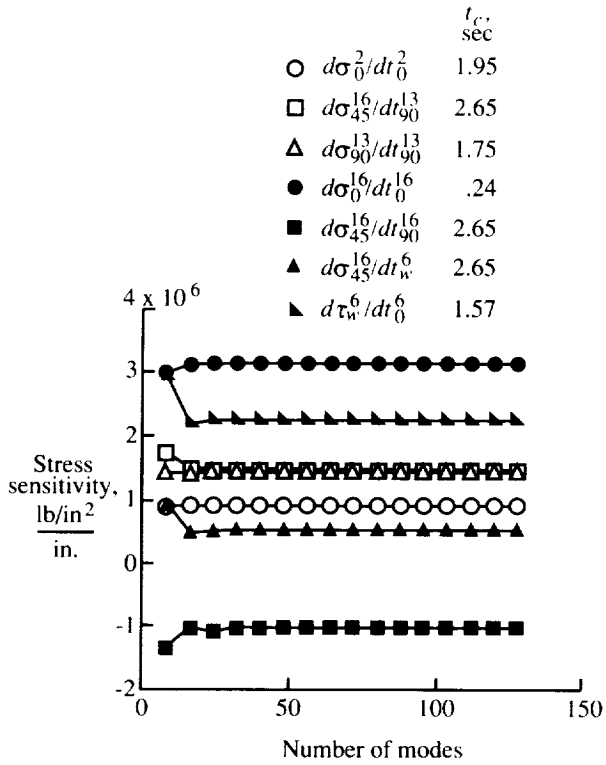


Figure 5.57. Modal convergence of selected stress sensitivities for delta wing calculated with semianalytical method with $d\Phi/dx$ approximated with modified modal method.

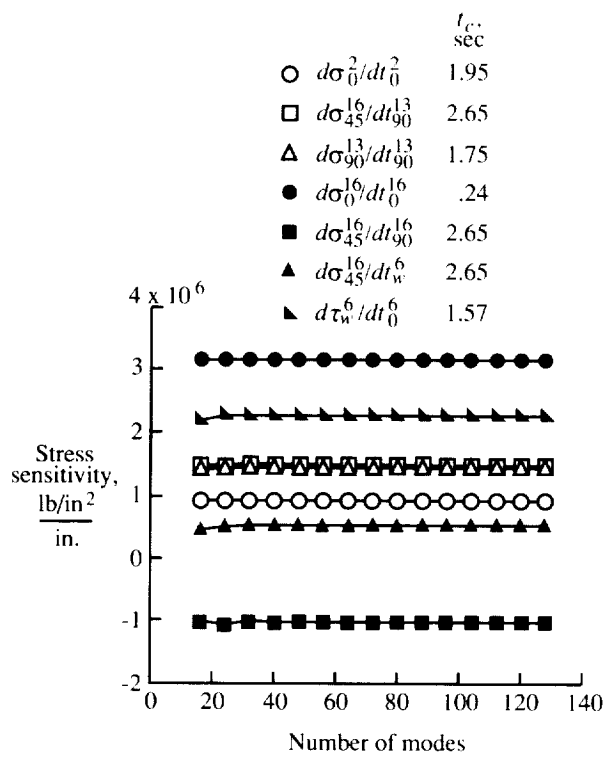


Figure 5.58. Modal convergence of selected stress sensitivities for delta wing calculated with semianalytical method with $d\Phi/dx$ approximated with only pseudostatic solution.

sensitivities with the use of the modified modal method but approximating $d\Phi/dx$ with only the pseudostatic term in equation (4.10). These results are shown in figure 5.58. Comparing this figure with figure 5.57 shows that for more than eight modes the results are nearly identical. It appears that a cheap, effective approximation to $d\Phi/dx$ in the semianalytical formulation can be obtained with only the pseudostatic term from the modified modal method.

For the five-span beam example, the convergence of the stresses was substantially improved by including the static solution via either the mode acceleration method, the static mode method, or the RWL method. The RWL method is attractive because it is cheaper to calculate n_r RWL vectors than n_r vibration mode shapes. However, incorporating the modified modal method in the sensitivity calculations with RWL vectors would seem to be impossible because it is derived to calculate the derivatives of vibration eigenvectors. (See eq. (4.10).) Regardless, it seems like a worthwhile numerical experiment to try using RWL vectors along with the pseudostatic correction term from the modified modal method in the variable mode, semianalytical formulation. One legitimate argument for doing this is the well-known observation that the basis spanned by the RWL vectors is an excellent approximation to the basis spanned by the eigenvectors. The results of this experiment for the modal convergence of the stresses in the delta wing are shown in figure 5.59. The convergence here is quite good also. For small numbers of modes the convergence is a bit erratic but in all cases the results are good for more than 32 modes. The benefit of combining the RWL vectors with the pseudostatic approximation to $d\Phi/dx$ is that the RWL vectors add the often important static displacement component to the basis, whereas the pseudostatic term adds components reflecting the change in the design variable to the basis.

As mentioned, the benefit of the mode acceleration method is that it also includes this pseudostatic term. The semianalytical, mode acceleration, sensitivity method described in chapter 4 was also applied to this delta wing example. Again, the modal convergence of the stress sensitivities shown in the previous figures is considered. Figure 5.60 shows the excellent convergence of the stress sensitivities. Clearly, the mode acceleration method provides the same improvement in stress sensitivities as the semianalytical method with a modified modal approximation to $d\Phi/dx$.

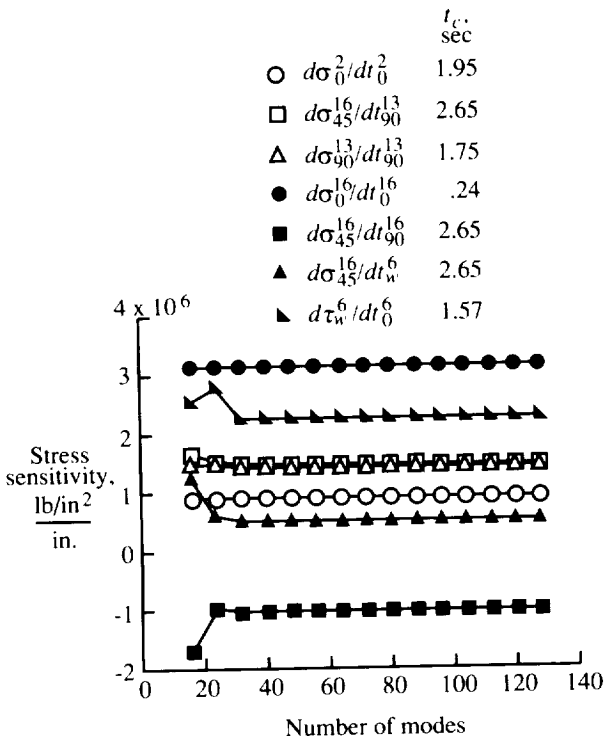


Figure 5.59. Modal convergence of selected stress sensitivities for delta wing calculated with semianalytical method with RWL vectors and pseudostatic approximation to $d\Phi/dx$.

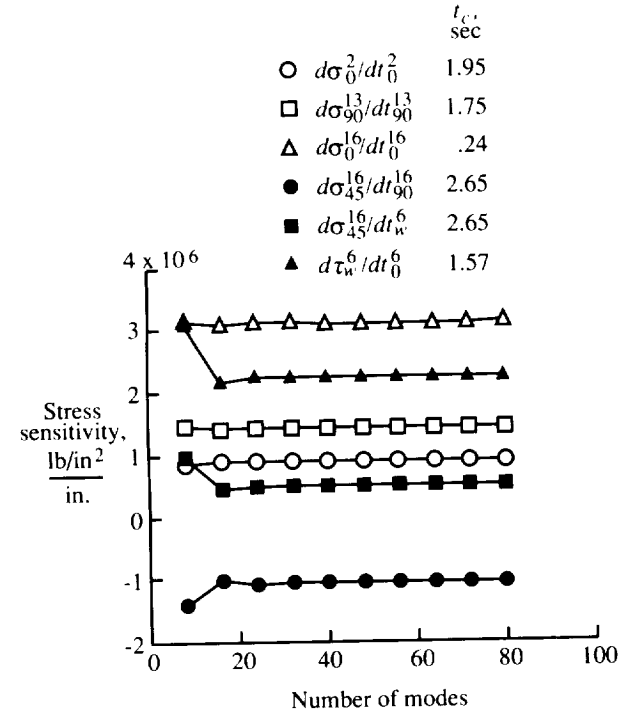
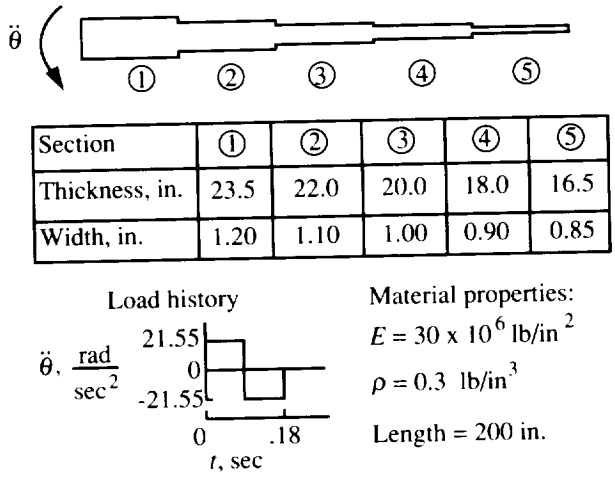


Figure 5.60. Modal convergence of selected stress sensitivities for delta wing calculated with semianalytical mode acceleration method. .

5.3. Stepped Cantilever Beam Example

The third example considered is a cantilever beam with five different rectangular cross sections along the length. (See fig. 5.61.) This example is taken from reference 38 where its minimum mass design under a static tip load was considered. The thickness and width of the beam cross section in each of the five sections are given in the table insert on figure 5.61 and represent an optimized design from reference 38. The beam is 200 in. long and, in the nominal case, each of the five sections has the same length. The material properties for the beam are also shown in figure 5.61.



Section	①	②	③	④	⑤
Thickness, in.	23.5	22.0	20.0	18.0	16.5
Width, in.	1.20	1.10	1.00	0.90	0.85

Figure 5.61. Stepped cantilever beam with applied rotational acceleration at root.

In most of the analyses, the beam is modeled with three finite elements per section. The transverse displacement and rotation are the nodal unknowns resulting in a total of 30 degrees of freedom for this case. The effect of different numbers of elements per section on the lowest 10 beam natural frequencies (with the beam clamped at the root) is shown in table 5.4. In the transient response analyses 0.5 percent of critical damping is included for each mode.

5.3.1. Loading

The loading for this stepped beam example is significantly different than for the first two examples. First, the load results from prescribing the acceleration at the beam root rather than by applying a force to the beam, and second, the time history as shown in figure 5.61 is more complicated than the simple step and ramp histories in the previous examples. The objective of this particular loading condition is to simulate the rotation of an appendage attached

Table 5.4. Lowest Frequencies for Stepped Cantilever Beam

Mode	Frequency, Hz, for			
	3 elements	4 elements	5 elements	6 elements
1	22.67	22.67	22.67	22.67
2	102.67	102.66	102.66	102.65
3	249.72	249.62	249.80	249.55
4	440.57	440.04	439.80	439.67
5	652.50	650.82	650.04	649.62
6	878.48	874.48	872.58	871.54
7	1093.36	1086.15	1082.61	1080.64
8	1296.61	1285.63	1279.94	1276.72
9	1479.81	1465.74	1457.74	1453.09
10	1641.44	1625.75	1615.41	1609.19

to a relatively large mass (e.g., robotic arm). The acceleration history in figure 5.61 rotates the root of the beam through 10° in 0.18 sec. After 0.18 sec, the beam root is motionless while other points in the beam are undergoing dynamic motion. Beam displacements, velocities, and accelerations in the following sections are with respect to the rotating coordinate system.

This type of applied acceleration can be handled as an equivalent external force given as

$$\mathbf{p} = -\mathbf{M}\mathbf{r}g(t) \tag{5.1}$$

where $\mathbf{r}$ is a vector describing the linear rigid body rotation of the beam about its root and $g(t)$ is the prescribed acceleration history given in figure 5.61. It should be noted that the applied force in this case depends on the system mass matrix; this must be considered in the sensitivity calculations.

5.3.2. Stepped Beam Dynamic Response

The transient behavior of the beam is strongly affected by the period of the loading. From table 5.4, the period of the lowest vibration mode is 0.044 sec, whereas the period of the square-wave loading is 0.18 sec. From figure 5.62, it can be seen that in the time history of the beam tip displacement, this first mode predominates, and almost exactly four cycles occur during the period of the loading. After the load is removed, the displacement response at the tip is relatively small. The bending stress at the root has a time history similar to that of the tip displacement as can be seen in figure 5.63 but with slightly more participation from higher frequency modes. As expected, the acceleration time history for the tip as shown in figure 5.64 is considerably more jagged; this indicates the participation of many higher frequency modes. This behavior is largely due to the abrupt changes in loading in the square-wave input. Significant accelerations exist at the tip after the loading is removed.

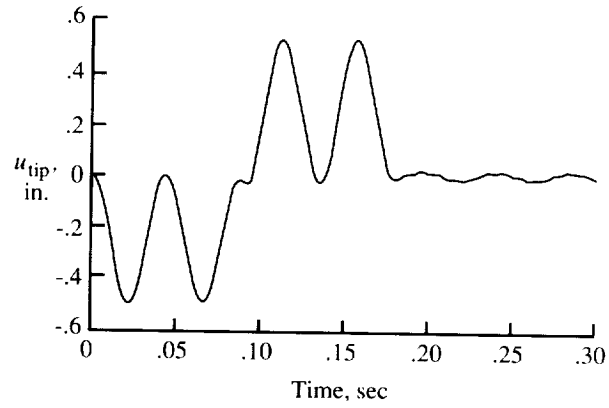


Figure 5.62. Time history of tip displacement for stepped cantilever beam.

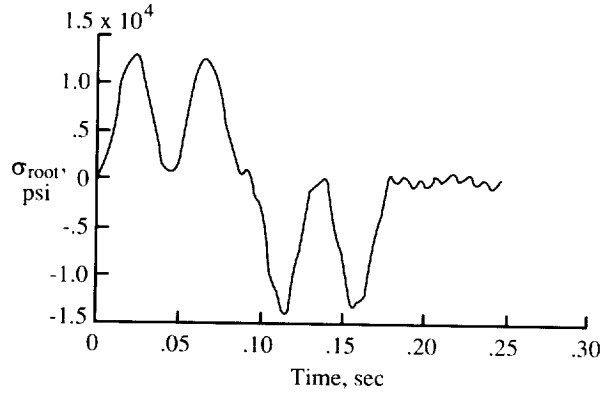


Figure 5.63. Time history of root stress for stepped cantilever beam.

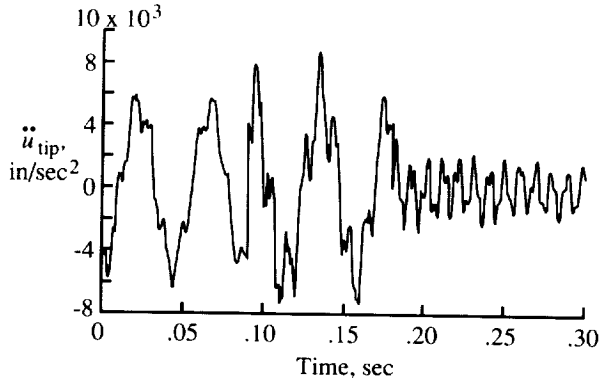


Figure 5.64. Time history of tip acceleration for stepped cantilever beam.

5.3.2.1. Modal convergence. The first convergence study considered the effect of the number of finite elements per section on the convergence of the critical point displacements, velocities, and accelerations at the beam tip and stresses at the root. For all these quantities, the convergence is excellent. For example, the peak acceleration changes by less than 1 percent when the number of finite elements per section is varied from 3 to 8. Then, for the beam modeled with three elements per section, the effect of the number of modes used in the analysis was considered. Generally the convergence was better than expected. Figure 5.65 shows the modal convergence of the tip acceleration at two different critical time points calculated with the mode displacement method. The values are essentially converged with five modes.

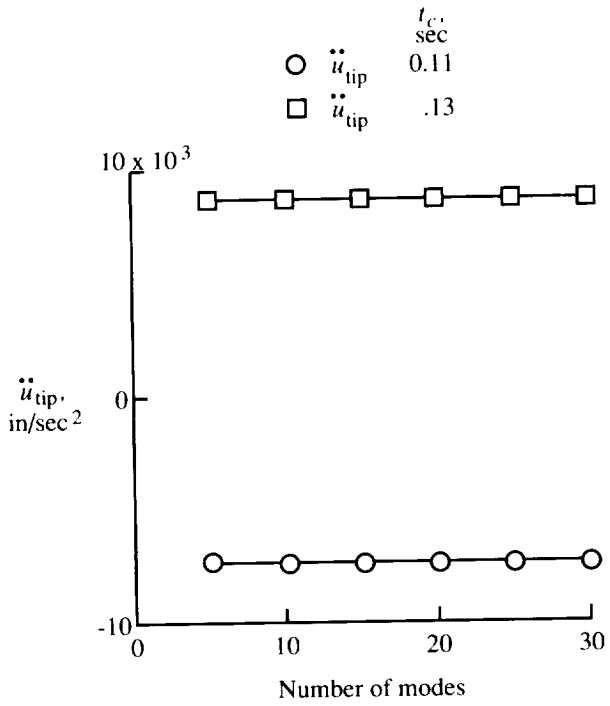


Figure 5.65. Modal convergence of critical point tip accelerations for stepped cantilever beam. Mode displacement method.

The modal convergence of the stress at the beam root is also rapid. Figure 5.66 shows the convergence of the root stress at two critical points calculated with the mode displacement method. No more than five modes are required for convergence. It was mentioned that there is a strong static component in the beam response during the period while the load is applied. Usually this requires the use of the mode acceleration method or RWL vectors for acceptable convergence of the stresses. Evidently the lowest vibration mode is close enough to the static displacement shape for this cantilever beam that the mode displacement method gives good values for the stresses.

5.3.2.2. Use of RWL vectors in analysis. In the stepped beam and delta wing examples, the convergence with RWL vectors in analysis and sensitivity calculations was generally as good or better than with vibration modes. The modal convergence in the stepped cantilever beam example when RWL vectors are used is very good also as seen in figure 5.67 for accelerations.

As can be seen in figure 5.67, the largest number of RWL vectors used in the analysis is 20. In the convergence studies considering vibration modes (e.g., fig. 5.65), the full set of 30 modes was used. A complete set of RWL vectors could not be generated for this example because of ill-conditioning inherent in the numerical process (eqs. (2.26) through (2.29)). As additional RWL vectors are generated, round-off errors cause the vectors to become less and less orthogonal. Eventually, the vectors become linearly dependent; this results in a singular reduced system. In most practical applications of this RWL method, this singularity problem would not occur because the number of RWL vectors generated would be much smaller than the total number of degrees of freedom.

5.3.3. Sensitivities of Stepped Beam Dynamic Response

5.3.3.1. Design variables. Two different classes of design variables are considered in this example. The first class is the set of beam thicknesses in each of the five sections. They are denoted h_i, where i is the section number from figure 5.61. These are similar to thickness design variables considered in the five-span beam and delta wing examples. Sensitivity results are presented in the next sections with h_1 and h_5 considered from this set.

The second class of design variables is the set of lengths of the five sections in the beam. The beam length is fixed at 200 in.; thus, only four design variables determine the lengths of the five sections. The four design variables are denoted l_i, where l_i is the distance from the beam root to the end of the ith section. Sensitivity results are presented in the next sections with l_1 and l_4 considered from this set. In the structural optimization field this type of design variable is often referred to as a "shape" design variable and is studied separately from member thickness-type design variables. A recent study (ref. 39) considered the calculation of static response sensitivities with respect to shape design variables with the semianalytical method. It was found that numerical difficulties in the semianalytical method resulted

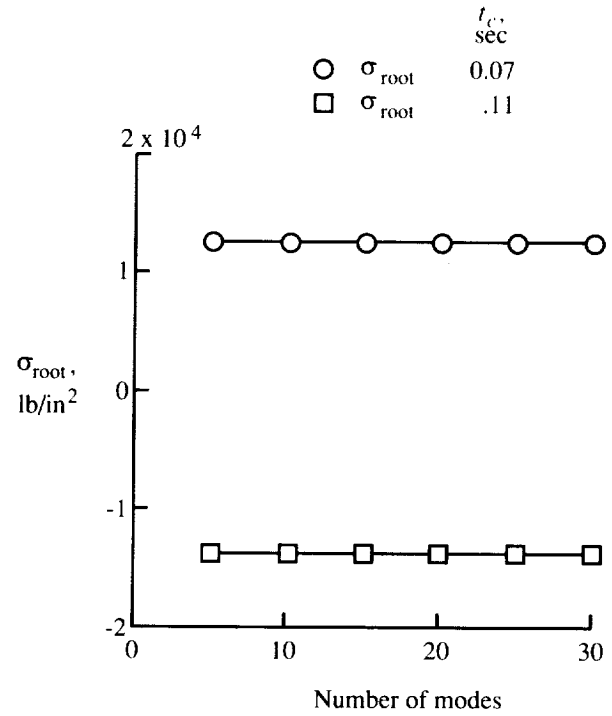


Figure 5.66. Modal convergence of critical point root bending stresses for stepped cantilever beam. Mode displacement method.

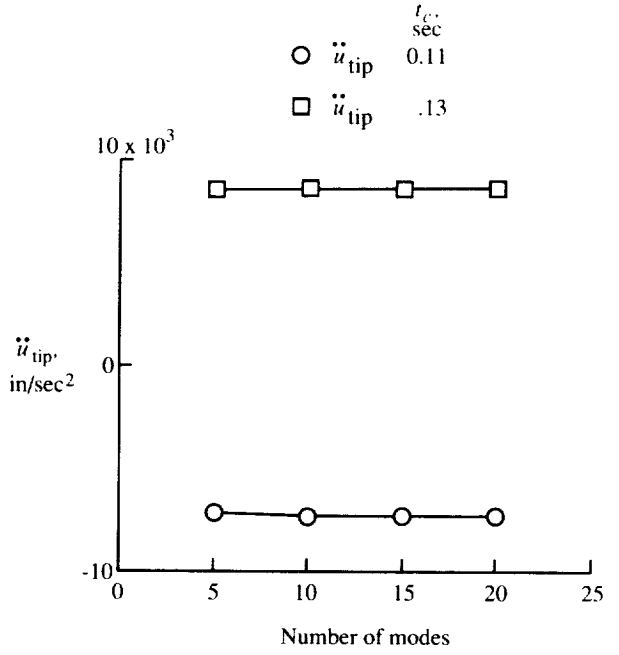


Figure 5.67. Modal convergence of critical point tip accelerations for stepped cantilever beam. RWL vectors.

in very large errors in sensitivities. This difficulty is addressed in sections 5.3.3.2 and 5.3.3.3 for the transient case.

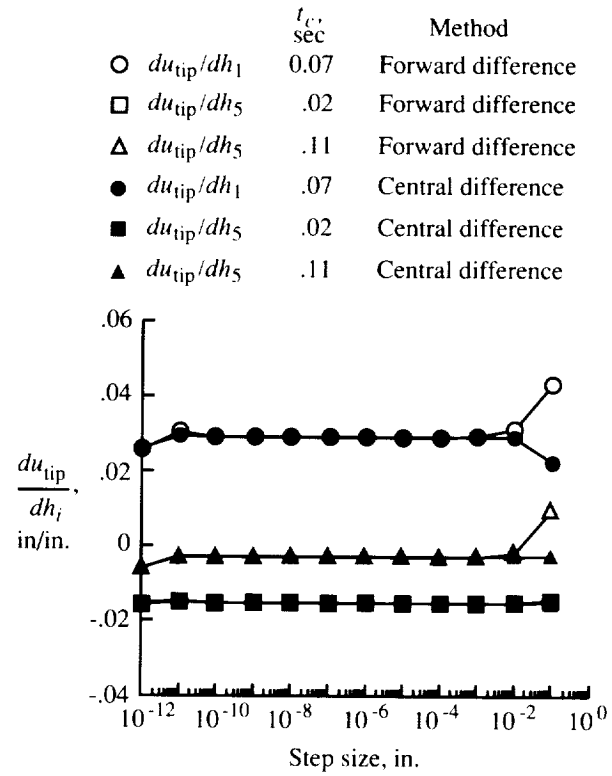


Figure 5.68. Effect of finite difference step size on accuracy of tip displacement derivatives with respect to thickness design variables for stepped cantilever beam. Forward and central difference operators.

5.3.3.2. Effect of finite difference step size. It is shown in this section that, practically, the selection of finite difference step size is not a concern for this stepped beam example. A series of studies was performed to consider the effect of step size on both thickness and length sensitivities calculated with finite difference and semianalytical methods. The finite element model with three elements per section was used and all 30 modes were included. Figure 5.68 presents approximate derivatives of tip displacement with respect to section thicknesses calculated with overall forward and central difference methods. A key point to be made is that both methods give excellent results for approximately an 8-decade step size range. For the large step size of 10^{-1} in., the central difference operator generally gives better results than the forward difference operator as would be expected. The results are nearly as good for sensitivities of the root stress with respect to the section thicknesses as shown in figure 5.69.

Compared with figure 5.68, there is slightly more error for the smallest and largest step sizes but the sensitivities are still accurate over a very broad range of step sizes. If sensitivities of stresses with respect to the length design variables are considered, the results are also very good. Figure 5.70 shows sensitivities calculated with the forward and central difference methods. Again there is a broad range of step sizes that provide accurate sensitivities. For the smaller step sizes the results are generally less accurate than in figure 5.69, but for the 10^{-2} step sizes they are more accurate.

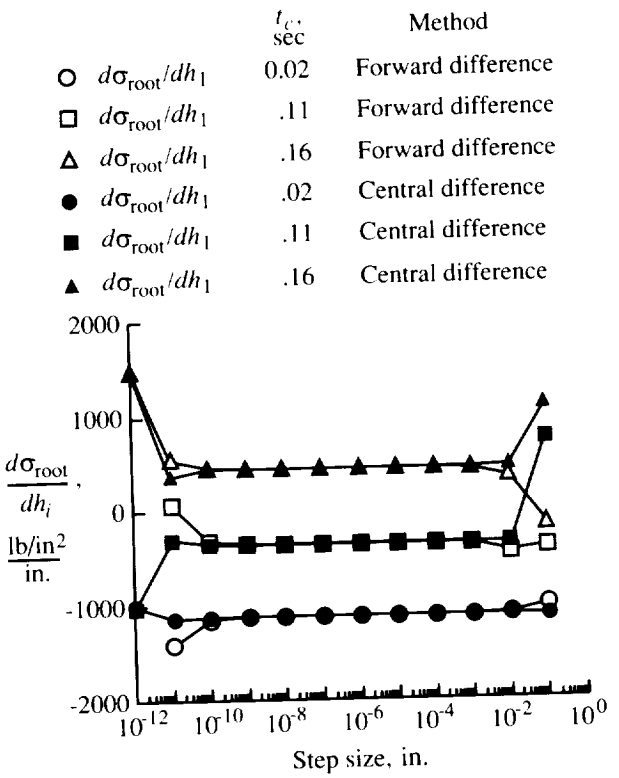


Figure 5.69. Effect of finite difference step size on accuracy of root stress derivatives with respect to thickness design variables for stepped cantilever beam. Forward and central difference operators.

It has been mentioned that severe numerical difficulties were uncovered in reference 39 when sensitivities of static response were calculated with respect to shape design variables. The result of this numerical ill-conditioning could be seen by calculating the sensitivities with different finite difference step sizes used for approximating the derivatives of the stiffness matrix. For very small step sizes, the error in the sensitivities is due to round-off. For the larger step sizes, however, the errors in sensitivities were much larger than those due to truncation of the finite difference operator and were found to be caused by basic ill-conditioning in solving the semianalytical equations.

This same phenomenon occurs when sensitivities are calculated with a semianalytical method for the transient case. Figure 5.71 shows approximate derivatives of root stress with respect to the length design variables calculated with the forward difference and semianalytical methods. Again, all 30 modes are used in the analyses. For the smaller step sizes, the accuracy is significantly better for the semianalytical method than for the overall forward difference method. For the 10^{-2} step size, however, the results from the forward difference method are excellent, whereas several of the sensitivities calculated with the semianalytical method exhibit extremely large errors. This result is completely consistent with that in reference 39. Although in this example, there is a large range of step sizes where accurate sensitivities can be obtained, in general, this would not be the case. Especially as the problem becomes larger it is desirable to use larger step sizes in a semianalytical method, but this is severely restricted for shape design variables by the type of error shown in figure 5.71.

5.3.3.3. Modal convergence of sensitivities. Most of the sensitivities exhibit the same good modal convergence as the response quantities. For example, the modal convergence of approximate derivatives of tip displacement with respect to h_1 and h_5 at different critical points is shown in figure 5.72. The sensitivities were calculated with the central difference method with updated modes and, as can be seen, the convergence is excellent. The convergence of tip acceleration derivative approximations is not as good as the displacement derivative approximations but is still acceptable as seen in figure 5.73. Again, these sensitivities are with respect to h_1 and h_5 and are calculated with the central difference method with updated modes.

Convergence is also good when sensitivities with respect to the length design variables are considered. Figure 5.74 shows the modal convergence of approximate derivatives of acceleration with respect to l_1 and l_4 calculated with the central difference method. A step size of 10^{-5} was used to avoid the problem shown in figure 5.71. As can be seen in figure 5.74, convergence is achieved with approximately 10 modes.

The modal convergence of stress sensitivities is similar to that for the delta wing example. When updated modes are used with an overall finite difference method, the convergence is excellent. An example of this is shown in figure 5.75 where approximate derivatives of the root stress with respect to h_1 and h_5 calculated with the forward difference

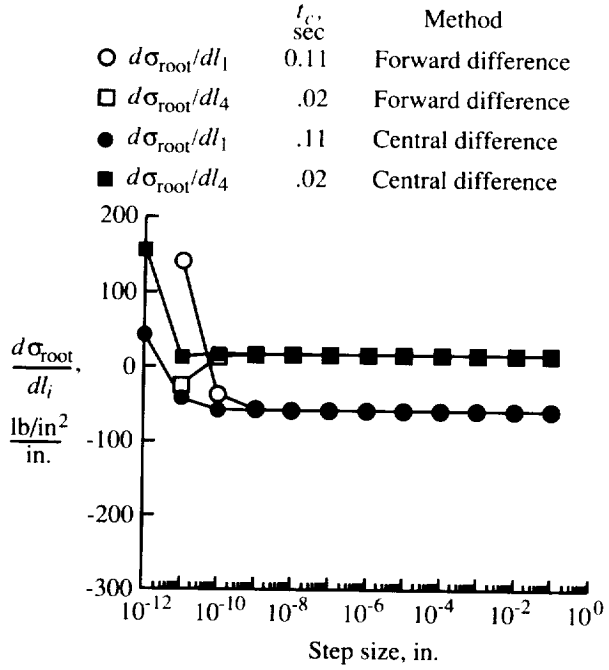


Figure 5.70. Effect of finite difference step size on accuracy of approximate root stress derivatives with respect to length design variables for stepped cantilever beam. Forward and central difference operators.

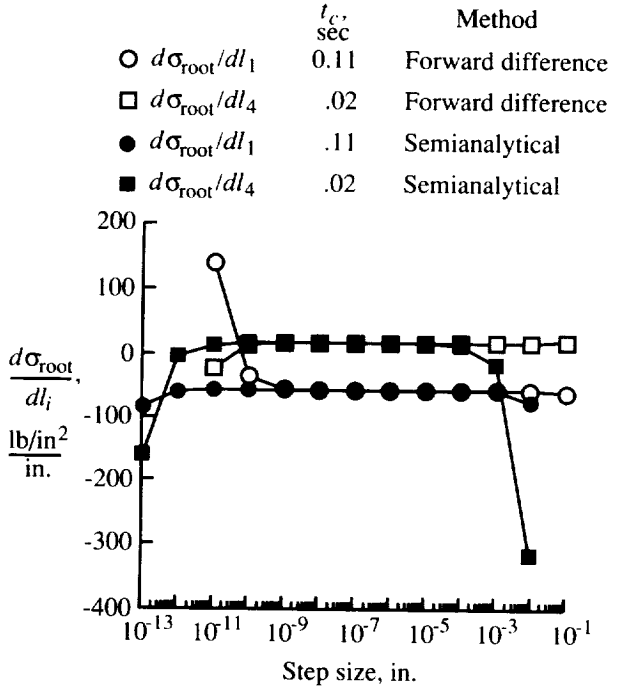


Figure 5.71. Effect of finite difference step size on accuracy of approximate root stress derivatives with respect to length design variables for stepped cantilever beam. Overall forward difference and semianalytical methods.

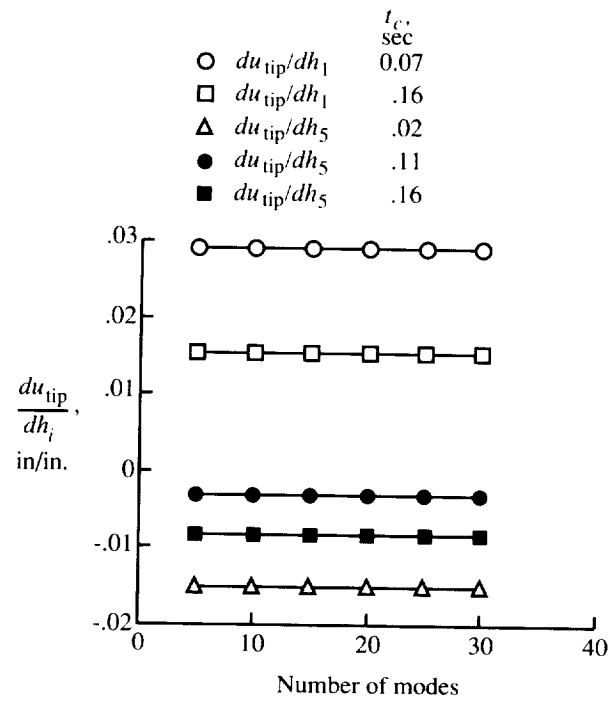


Figure 5.72. Modal convergence of approximate derivatives of tip displacement with respect to thickness design variables for stepped cantilever beam. Mode displacement method; central difference operator.

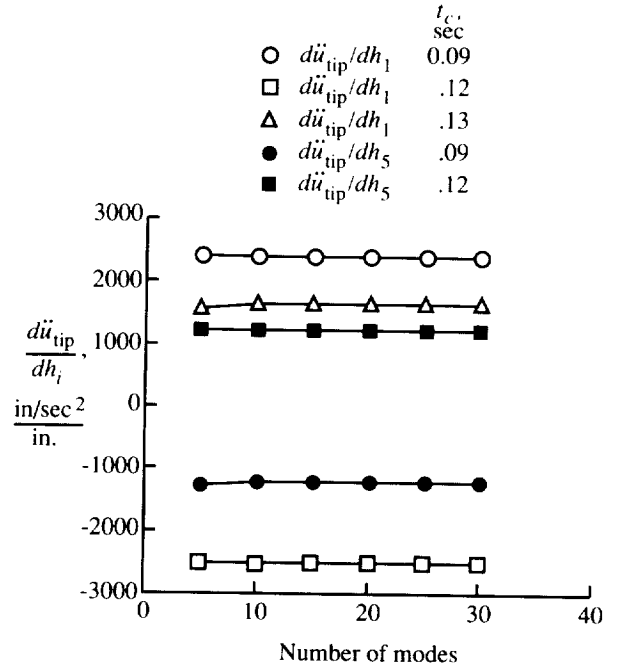


Figure 5.73. Modal convergence of approximate derivatives of tip acceleration with respect to thickness design variables for stepped cantilever beam. Mode displacement method; central difference operator.

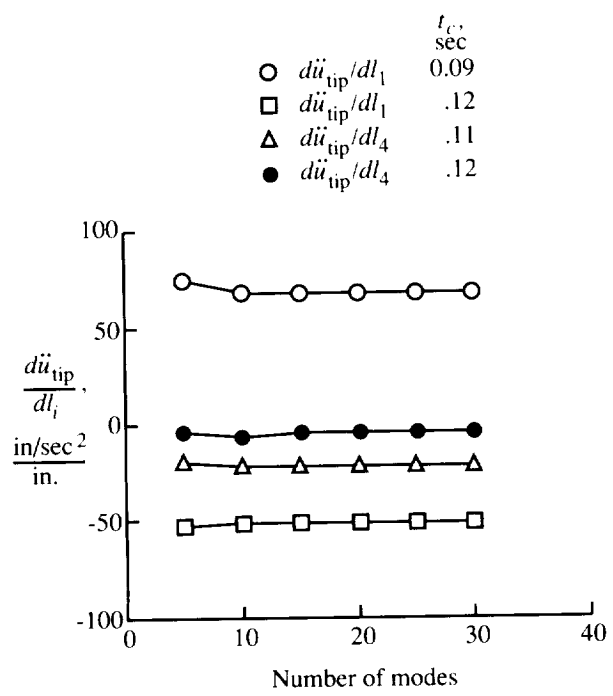


Figure 5.74. Modal convergence of approximate derivatives of tip acceleration with respect to length design variables for stepped cantilever beam. Mode displacement method; central difference operator.

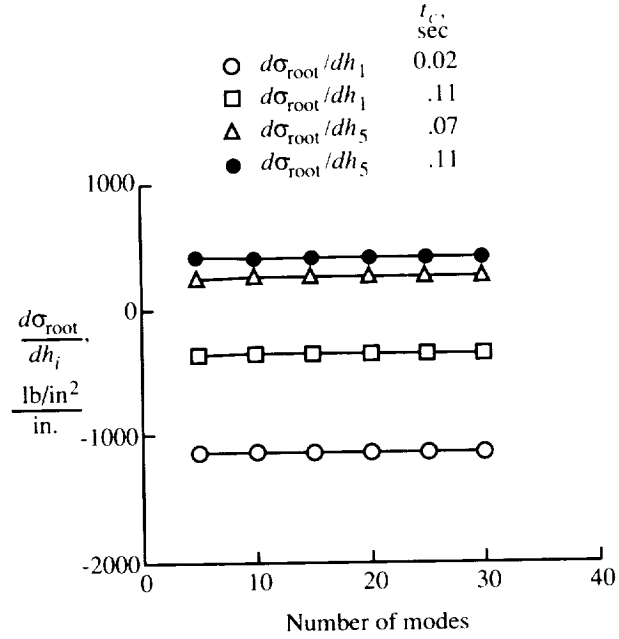


Figure 5.75. Modal convergence of approximate derivatives of root stress with respect to thickness design variables for stepped cantilever beam. Mode displacement method; forward difference operator; updated modes.

operator are shown. However, if fixed modes are used in a finite difference procedure, the modal convergence is much worse. Figure 5.76 shows an example of this for the same sensitivities as in figure 5.75. Also, if sensitivities of the root stress with respect to the length design variables (l_1, l_4) are considered, the modal convergence is very poor. An example of this poor convergence is shown in figure 5.77. The convergence is similarly bad if the fixed-mode semianalytical method is used instead of a finite difference method. Figure 5.78 shows the poor modal convergence of the same sensitivities as figure 5.77 but calculated with the fixed-mode, semianalytical method.

For the delta wing example, remedies for the poor convergence of stress sensitivities in the semianalytical method were based on approximating the mode shape derivatives $d\Phi/dx$. These semianalytical methods including approximations for $d\Phi/dx$ were also applied to this stepped beam example. First $d\Phi/dx$ was approximated with the modified modal method. The modal convergence of the stress sensitivities is now excellent as can be seen in figure 5.79. The degree of improvement can best be appreciated by comparing figures 5.78 and 5.79 and noting that the range of the ordinate in figure 5.78 is much broader than in figure 5.79. Using only the first pseudo-static term from the modified modal method as an approximation to $d\Phi/dx$ was also tried. As can be seen in figure 5.80, the convergence is adequate though not quite as good as when the complete modified modal method is used.

Just as in the delta wing example, a case was also considered where RWL vectors were used instead of vibration modes but their derivatives were computed by the modified modal method (version with pseudo-static term plus modes). Again, somewhat surprisingly, the modal convergence of the stress sensitivities is good as seen in figure 5.81.

The semianalytical mode acceleration method was also tried as a remedy for the poor convergence of the stress sensitivities. Again, the very poor convergence is eliminated as can be seen in figure 5.82.

5.4. Summary

A number of different methods for calculating sensitivities of transient response quantities have been exercised on three example problems: a five-span beam, a composite aircraft wing, and a variable-cross-section beam. Two of the methods are overall finite difference methods where the analysis is repeated for perturbed designs. The other methods are termed semianalytical methods because they involve direct, analytical differentiation of the equations of motion with finite difference approximations

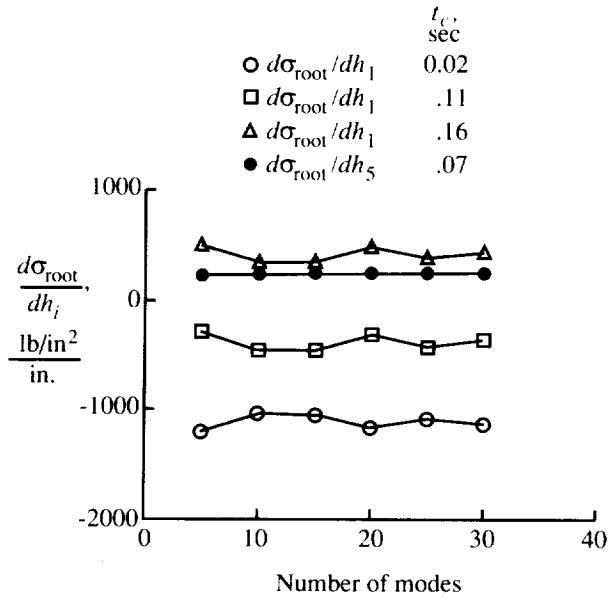


Figure 5.76. Modal convergence of approximate derivatives of root stress with respect to thickness design variables for stepped cantilever beam example. Mode displacement method; forward difference operator; fixed modes.

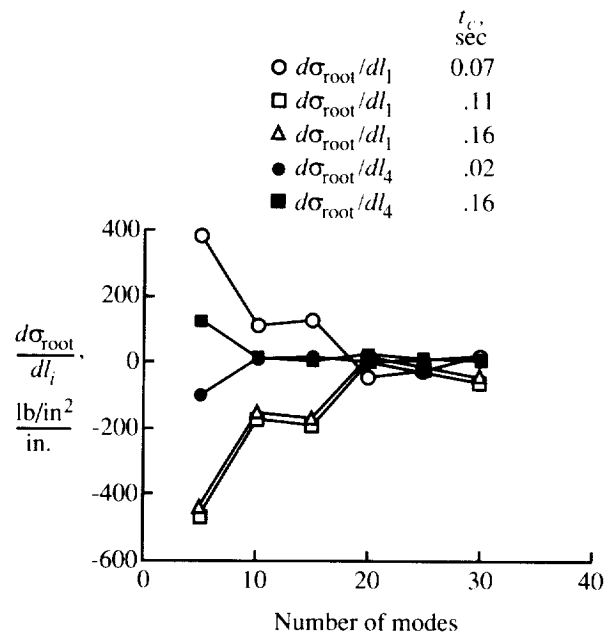


Figure 5.78. Modal convergence of approximate derivatives of root stress with respect to length design variables for stepped cantilever beam. Mode displacement method; semianalytical method.

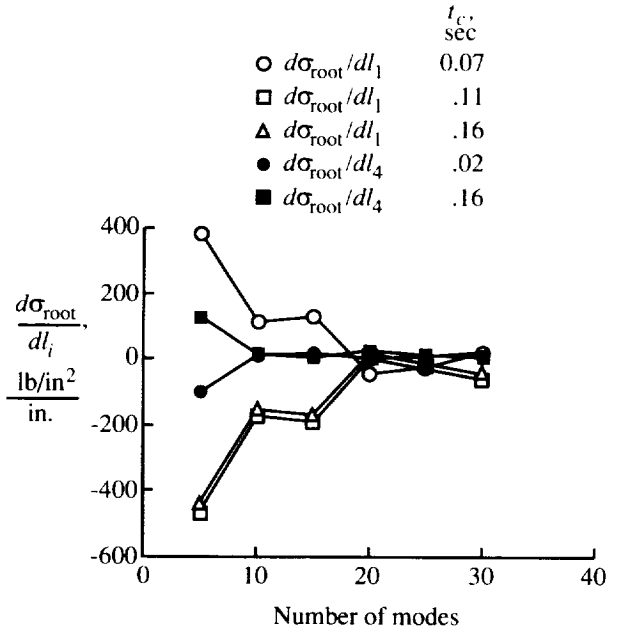


Figure 5.77. Modal convergence of approximate derivatives of root stress with respect to length design variables for stepped cantilever beam. Mode displacement method; forward difference operator; fixed modes.

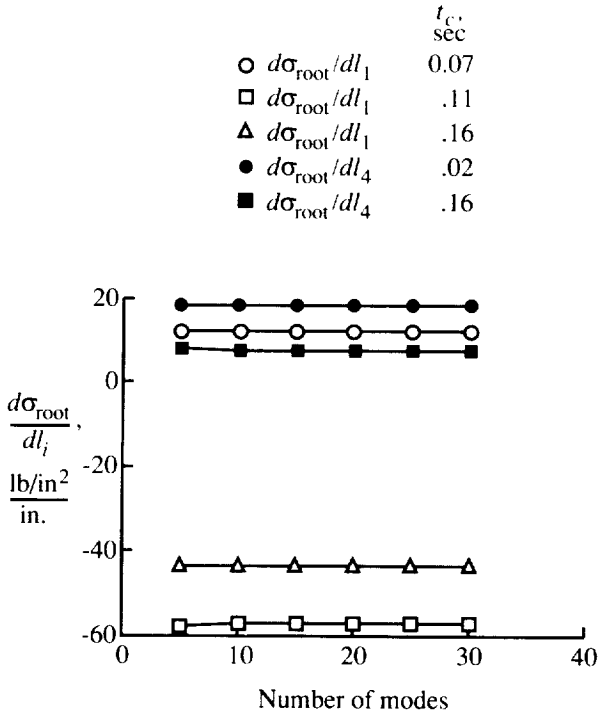


Figure 5.79. Modal convergence of approximate derivatives of root stress with respect to length design variables for stepped cantilever beam. Mode displacement method; semianalytical modified modal method.

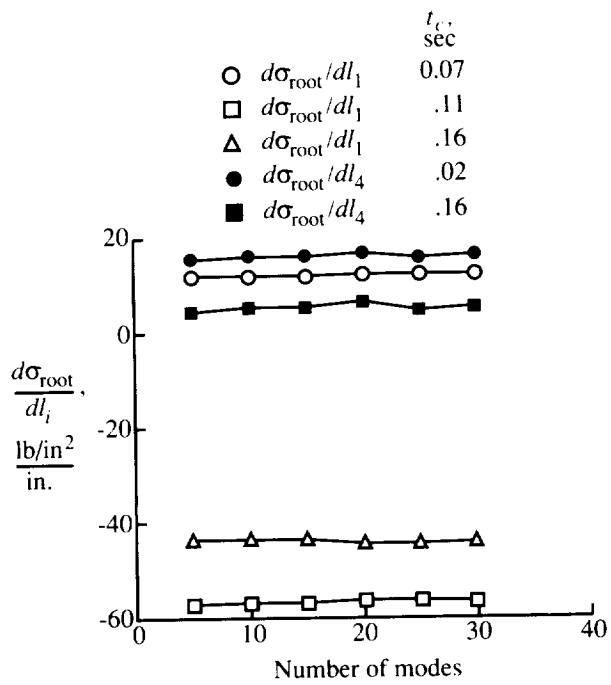


Figure 5.80. Modal convergence of approximate derivatives of root stress with respect to length design variables for stepped cantilever beam. Mode displacement method; semianalytical one-term modified modal method.

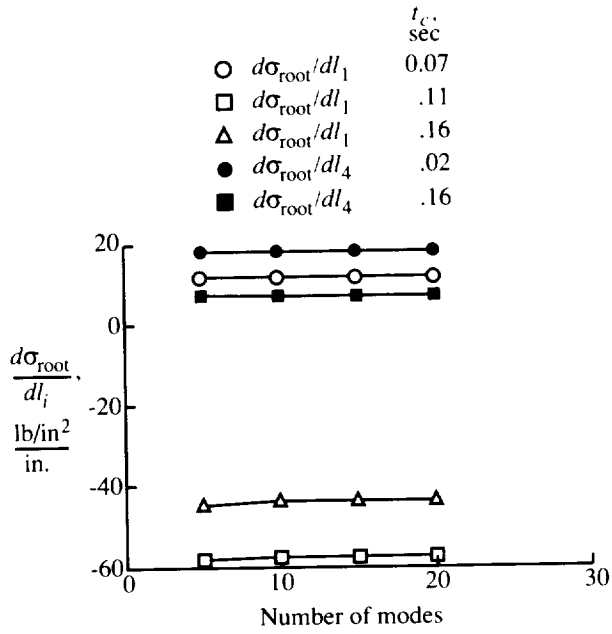


Figure 5.81. Modal convergence of approximate derivatives of root stress with respect to length design variables for stepped cantilever beam. RWL vectors; semianalytical modified modal method.

of the coefficient matrices. All the methods use basis vectors to reduce the dimensionality of the problem. Accordingly, the convergence of both the transient response quantities and their sensitivities as a function of number of basis vectors was a key concern in this chapter.

In the delta wing and stepped cantilever beam examples, the convergence of the response quantities was consistently very good. However, this was not true with the five-span beam. With the five-span beam under a concentrated end moment and ramp time history, the convergence of displacements and velocities was adequate. However, the convergence of accelerations was poor. The convergence of stress resultants for this example depended on how they were calculated. When the mode displacement method was used, the convergence was quite poor. However, when the mode acceleration method, the Ritz-Wilson-Lanczos vector method, or the static mode method was used, the convergence was good. In cases where convergence was poor for the five-span beam, the addition of modal or discrete damping improved the convergence somewhat. However, it did not eliminate the convergence problems.

The modal convergence of the sensitivities in the three examples is consistent with the convergence of the response quantities themselves. For the delta

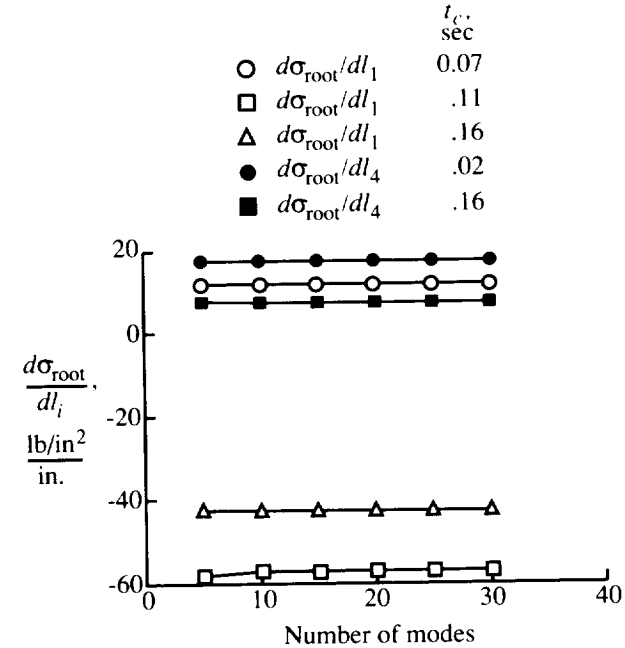


Figure 5.82. Modal convergence of approximate derivatives of root stress with respect to length design variables for stepped cantilever beam. Mode acceleration; semianalytical method.

wing and stepped cantilever beam examples, the convergence of sensitivities was generally good. For the five-span beam example, the convergence of displacement sensitivities was adequate but the convergence of velocities, accelerations, and stress resultants was generally poor. This poor convergence was observed for all the sensitivity calculation methods. Furthermore, it appears to be associated with the structure and loading because no improvement was observed as the number of finite elements per span was increased.

In certain cases poor convergence of sensitivities was also observed for the delta wing and stepped cantilever beam examples. When sensitivities of stresses were calculated with the fixed-mode overall finite difference methods or the fixed-mode semianalytical methods, the convergence was very poor. In large problems, however, updating the vibration modes in the overall finite difference methods or rigorously calculating derivatives of the mode shapes is very expensive. The mode acceleration version of the semianalytical method and the semianalytical method with mode shapes approximated with the modified modal method were devised to alleviate this poor convergence with lower computational cost. When both methods are applied to delta wing and stepped cantilever beam examples, the modal convergence of sensitivities is excellent.

All the sensitivity calculation methods considered herein rely on finite difference operators. Thus step size selection is an important concern. The system stiffness and mass matrices are linear functions of many of the design variables in the three example problems. This allowed large step sizes to be used in the semianalytical methods to minimize the round-off errors and produce accurate derivatives of the stiffness and mass matrices. Also there is less opportunity for round-off error in calculating finite difference derivatives of just the coefficient matrices compared with finite difference derivatives of the overall response quantities. For these reasons, the semianalytical methods were consistently less sensitive to finite difference step size than the overall finite difference methods.

Chapter 6

Computational Costs

A consideration of the computational costs is essential for evaluating any numerical method. This is especially difficult in large-scale, finite-element-based procedures because there is often considerable "overhead" required in the practical implementation of a given numerical method. For example, most finite element codes require only a small portion of a system matrix to be resident in central memory during factorization at any given time. The other portions of the matrix are read from disk and the factored portions are written to disk as required. A similar situation can exist on virtual memory machines where the disk operations are transparent to the implementor. In these cases, the computer resources required are very implementation dependent.

An approach that is common in the formal study of numerical methods is to evaluate the computational cost by counting the number of floating point operations. There are some pitfalls to this approach. Sometimes, even for large problems, because of the required overhead it is impossible to achieve a practical implementation that will execute as fast as the predictions from the operation count. At other times, especially on vector machines, it is possible for a method with a higher operation count to be faster than a method with a lower operation count.

Nevertheless, this approach is used here, primarily to indicate the major trends in the costs of the methods, not to make fine distinctions between them. Following common practice, a floating point operation (or "flop") is defined as the combination of a floating point multiply, add, and associated array indexing. In the rest of this chapter a floating point operation is often referred to simply as an operation.

6.1. Costs of Basic Matrix Manipulations

Multiplication of full matrices occurs in several places in the transient response and sensitivity methods. The approximate number of floating point operations required to multiply a full $l \times m$ matrix and a $m \times n$ matrix is given as

$$C_{\text{fmul}} = l\,m\,n \tag{6.1}$$

Solution of the reduced eigenproblem (eqs. (2.23)) is important in solving the system eigenproblem with subspace iteration (eqs. (2.3)) and in uncoupling the reduced system when basis vectors other than the eigenvectors are used. In both cases, it is necessary to solve a full, generalized eigenproblem for all n_r eigenvalues and eigenvectors. Since eigenvalue solution techniques are inherently iterative, the number of operations required for a converged solution can only be estimated. Reference 15 estimates the number of operations for the complete solution of a generalized eigenproblem with the Jacobi method as

$$C_{\text{reig}} = 18n_r^3 + 36n_r^2 \tag{6.2}$$

Other techniques for solving this eigenproblem may have a significantly different cost. However, it is shown that the cost of this eigensolution is small relative to other tasks in the sensitivity calculations.

6.2. Costs of System Matrix Manipulations

For the purpose of considering the computational costs of operations on system matrices (e.g., $\mathbf{K}$, $\mathbf{M}$), these matrices are considered to be stored in a banded form. In a banded form, only matrix elements located near the diagonal are stored; the matrix elements outside this "bandwidth" of the matrix are zero and are not stored or considered in operations. Finite element problems yielding a stiffness matrix with a constant bandwidth are rarely encountered in practice so most finite element codes use more sophisticated and efficient schemes for storing system matrices. However, having a single, easily understood number to characterize the sparsity of a system matrix (the bandwidth) is convenient in approximating computational costs. Although few finite element problems have precisely a constant bandwidth, this assumption is accurate enough in many cases to get reasonable estimates for a relative number of operations in a numerical procedure.

From reference 40 the cost in number of operations of factoring a banded system matrix of order n_g is given as

$$C_{\text{bfac}} = \frac{1}{2}\beta(\beta+3)n_g - \frac{\beta^3}{3} - \beta^2 - \frac{2}{3}\beta \tag{6.3}$$

where β is one half the bandwidth (excluding the diagonal). Also from reference 40, the cost of a single solution of a banded system, given the factored matrix, is given as

$$C_{\text{bsol}} = 2(\beta+1)n_g - \beta(\beta+1) \tag{6.4}$$

The cost of multiplying a banded system matrix and a single vector is given as

$$C_{\text{bmul}} = (2\beta + 1)n_g \tag{6.5}$$

6.3. Cost of Basis Reduction

The process of reducing the degrees of freedom from n_g to n_r requires the matrix triple product operations shown in equations (2.6), (2.7), and (2.8). Since this process is used in all the sensitivity methods, the cost is considered separately here. In performing this operation, n_r system vectors are multiplied by a system matrix. Then $n_r(n_r + 1)/2$ inner products (for a symmetric system matrix) between system vectors are performed. The total number of required floating point operations is

$$C_{\text{red}} = n_r C_{\text{bmul}} + \frac{n_g\, n_r(n_r + 1)}{2} \tag{6.6}$$

6.4. Cost of System Eigensolution

The cost of solving the generalized vibration eigenvalue problem is even more difficult to estimate than the cost of solving the reduced eigenproblem. The numerical techniques vary widely among different analysis codes. Furthermore, a technique used for one problem might be totally inappropriate when applied to a different problem. Nevertheless, some assumptions are made here that will hopefully lead to a reasonable estimate of computational costs for a fairly broad class of problems.

First, it is assumed that the eigenvalue problem is solved with a subspace iteration technique with shifts (for example, ref. 15). In recent years, software based on this approach has become common. Also, the eigensolver, E4, in the EAL software used in this study (ref. 23) is based on this approach. It is also necessary to make assumptions about the number of vectors used in the subspace and the number of iterations at a given shift point required to converge some subset of these vectors to eigenvectors. The following numbers were used for these quantities with the realization that they may be optimum for only a few problems. Also, it is assumed that the eigensolution is being performed for a slightly perturbed model and the eigenvectors from the initial model are available as the initial subspace. At each shift point $n_{\text{tss}} = 16$ vectors are included in the subspace. After $n_{\text{it}} = 2$ iterations, $n_{\text{css}} = 8$ of these vectors have converged to eigenvectors.

The number of shifts or number of factorizations required is approximately

$$n_{\text{shift}} = \frac{n_r}{n_{\text{css}}} + 1 \tag{6.7}$$

At each iteration, the inverse power operation requires a matrix product between $\mathbf{M}$ and n_{tss} vectors followed by n_{tss} solutions of the system equations based on the current factored $\mathbf{K}$. The basis reduction operation for both $\mathbf{K}$ and $\mathbf{M}$ requires $n_{\text{tss}}(n_{\text{tss}} + 1)$ system vector inner products. Next, the eigenproblem of reduced order n_{tss} must be solved. This cost is given in equation (6.2) with n_r replaced by n_{tss}. Finally, the updated set of approximate system eigenvectors must be formed as a linear combination of the current approximation. This requires $n_{\text{tss}}^2\, n_g$ operations. The approximate cost of solving the system eigenproblem for n_r modes and frequencies can be written as

$$\begin{aligned} C_{\text{eig}} = n_{\text{shift}} C_{\text{bfac}} + n_{\text{shift}} n_{\text{it}} [&n_{\text{tss}} C_{\text{bmul}} \\ &+ n_{\text{tss}} C_{\text{bsol}} + n_{\text{tss}}(n_{\text{tss}} + 1) n_g \\ &+ 18 n_{\text{tss}}^3 + 36 n_{\text{tss}}^2 + n_{\text{tss}}^2\, n_g] \end{aligned} \tag{6.8}$$

6.5. Cost of Generating RWL Vectors

As has been demonstrated, RWL vectors are an attractive alternative to vibration mode shapes for basis reduction in transient response analysis. It has been mentioned previously that generation of the RWL vectors is considerably cheaper than vibration modes. An estimate of this cost in number of floating point operations is derived here.

First, a factorization of the system $\mathbf{K}$ is required. The system equations are solved n_r times based on the factored $\mathbf{K}$. The generation of right-hand-side vectors requires $n_r - 1$ matrix products between $\mathbf{M}$ and a vector. Another key step in the process is the Gram-Schmidt orthogonalization as indicated in equation (2.27). For all vectors, this requires $n_r - 1$ multiplications of a vector by $\mathbf{M}$ and $n_r(n_r - 1)/2$ vector inner products. The scaling of each vector requires n_r vector inner products and n_r divisions of a system vector by a scalar. Writing the total number of floating point operations in expanded form yields

$$\begin{aligned} C_{\text{RWL}} = C_{\text{bfac}} + n_r C_{\text{bsol}} + 2(n_r - 1) C_{\text{bmul}} \\ + \frac{n_r(n_r - 1)}{2} n_g + 2 n_r n_g \end{aligned} \tag{6.9}$$

6.6. Cost of Model Generation

The generation of the finite element model requires processing of the input, forming elemental matrices, and forming global system matrices. Most of the sensitivity calculation methods require generation of a single perturbed model for each design variable. The central difference method, however, requires the generation of two perturbed models. Thus to compare the central difference method with the other methods, an estimate of the model generation cost is required. This cost is difficult to calculate in general. For the purposes herein this cost is estimated empirically with EAL by observing the execution time for model generation relative to matrix multiplication for a number of models. From these experiments it was observed that the predominant element type in the model substantially affects the cost. That is, forming the element matrices in a model composed of three-dimensional solid elements is much more costly than in a model composed of rod elements. The estimate for model generation cost used here,

$$C_{\text{model}} = 100\beta n_g \tag{6.10}$$

roughly approximates the cost for a model with two-dimensional, plate-type elements in EAL but would be significantly in error for predominantly one-dimensional or three-dimensional models.

6.7. Cost of Integration of Reduced System

The basic operation for integrating the reduced system is shown in equation (2.30). The two matrix multiplications shown in equation (2.30) are performed at every time step. If equations (2.5) are coupled, $\mathbf{W}_{ij}$ and $\mathbf{N}_{ij}$ are full and the explicit matrix multiplication must be performed. In this case, the total number of floating point operations for integration of the system is given as

$$C_{\text{inte}} = 8n_r^2 n_t \tag{6.11}$$

where n_t is the number of time steps in the analysis. If equations (2.5) are not coupled, $\mathbf{W}_{ij}$ and $\mathbf{N}_{ij}$ are diagonal, and this fact can be exploited to substantially reduce the cost of integrating the system. The number of floating point operations in this case is given as

$$C_{\text{inte}} = 8n_r n_t \tag{6.12}$$

When the number of equations in the reduced system n_r is large, the difference between C_{inte} in equations (6.11) and (6.12) is very large. For the comparisons of sensitivity methods in this chapter, equation (6.12) is used to estimate the integration cost. When vectors other than vibration modes are used or vibration modes for an initial model are used with a perturbed model, the equations are first uncoupled by solving the reduced-order eigenproblem.

6.8. Cost of Back Transformation for Physical Response Quantities

After the reduced equations have been solved, it is necessary to recover the physical displacements, velocities, accelerations, and stresses (or stress resultants) of interest. Usually the quantities of interest are only a subset of all possible quantities available from the finite element model. In the critical point constraint formulation described in chapter 3 it is necessary to recover the physical response quantities only at the critical times. That is, the back transformation is performed at only 5 to 20 critical points rather than at thousands of time steps. The cost of the basic back transformation operation is

$$C_{\text{back}} = n_p n_r n_c \tag{6.13}$$

where n_p is the number of physical quantities being recovered from the modal values and n_c is the number of critical points. The costs of back transformation in the specific sensitivity methods will be expressed as a multiple of this basic cost.

6.9. Cost of Sensitivity Calculation Methods

Because all the sensitivity calculation methods require the dynamic analysis of the initial model, this component of the cost can be neglected in comparing the different methods. In addition, in all the methods the basic operations are repeated for each design variable so the costs estimated below are per design variable. Also, to simplify the cost analysis, the models are assumed to be undamped so that any operations dealing with modal damping or system damping matrices are not included.

6.9.1. Finite Difference Methods

Both forward and central difference methods for calculating sensitivities were considered in chapter 4. In the central difference method, the basic operations of the forward difference method are performed twice; therefore the cost is approximately twice that of the forward difference method. Costs are derived here for the forward difference method. In both finite difference methods, the basis vectors can be the same as for the original model (fixed) or recalculated for the perturbed model (updated). The cost with updated modes presented herein is based on using

natural vibration modes. An alternative of using RWL vectors is considered separately.

The first step in the forward difference method is evaluation of the perturbed model. If the modes are being updated, the eigenproblem is solved. Otherwise, the original modes are used to reduce the basis, and the reduced-order eigenproblem is solved to uncouple the transient equations. The uncoupled equations for the perturbed system are then integrated and the n_p physical quantities calculated at the n_c critical time points. The cost of the actual difference operation is very small and is therefore neglected. For the fixed-mode case, the total cost is

$$C_{\text{fdfix}} = C_{\text{model}} + 2C_{\text{red}} + C_{\text{reig}} + n_r^2 n_g + C_{\text{inte}} + C_{\text{back}} \tag{6.14}$$

For the updated mode case, the total cost is

$$C_{\text{fdupd}} = C_{\text{model}} + C_{\text{eig}} + C_{\text{inte}} + C_{\text{back}} \tag{6.15}$$

6.9.2. Semianalytical Method With Fixed Modes

The semianalytical method begins by evaluating the perturbed model. Then $d\mathbf{M}/dx$ and $d\mathbf{K}/dx$ are formed using a forward difference operator. Each derivative requires about βn_g operations. Then the basis reduction operation is applied to both derivative matrices. Formation of the right-hand-side pseudo load (eqs. (4.6)) is a fairly costly operation and the two matrix products $(d\overline{\mathbf{M}}/dx)\ddot{\mathbf{q}}$ and $(d\overline{\mathbf{K}}/dx)\mathbf{q}$ require about $n_r^2 n_t$ operations each. Finally, the uncoupled equations are integrated and the physical sensitivities recovered. For the purposes of cost estimation, a single quantity n_p is used as the total number of required physical sensitivities. In the semianalytical methods, however, the specific procedure for recovering the sensitivities depends on whether the quantity is a displacement, velocity, acceleration, or stress sensitivity. In estimating the costs of this back transformation operation these differences are ignored. One justification for this approach is that the cost of back transformation is usually small relative to other costs in the sensitivity calculation. In this fixed-mode, semianalytical method, approximately the same number of operations is required for the recovery of physical sensitivities as in the finite difference methods. The total number of floating point operations can be written as

$$C_{\text{safix}} = C_{\text{model}} + 2\beta n_g + 2C_{\text{red}} + 2n_r^2 n_t + C_{\text{inte}} + C_{\text{back}} \tag{6.16}$$

6.9.3. Semianalytical Method With Approximate $d\mathbf{\Phi}/dx$

Just as in the fixed-mode, semianalytical method, evaluating the perturbed model and forming $d\mathbf{M}/dx$ and $d\mathbf{K}/dx$ is the first step. The next step is using the modified modal method to approximate $d\mathbf{\Phi}/dx$.

The procedure for the modified modal method is given in equations (4.9), (4.10), (4.11), and (4.12). The calculation of the n_r pseudostatic contributions requires the formation of n_r right-hand-side vectors and n_r solutions of the system equations. The formation of the A_{jk} participation factors requires approximately n_r system matrix additions plus the equivalent of a triple product basis reduction operation. Forming the linear combination of pseudostatic term and eigenvectors requires $n_r n_g$ operations. The total cost in number of floating point operations for the modified modal method is

$$C_{\text{mmod}} = 2n_r n_g + n_r C_{\text{bsol}} + n_r \beta n_g + C_{\text{red}} + n_r n_g \tag{6.17}$$

Given $d\mathbf{\Phi}/dx$, the derivatives of the reduced system matrices can be formed. For both $d\overline{\mathbf{M}}/dx$ and $d\overline{\mathbf{K}}/dx$, two triple product, basis reductions plus n_r vector inner products (for the $d\mathbf{\Phi}^T/dx\mathbf{M}\mathbf{\Phi}$ term since $\mathbf{M}\mathbf{\Phi}$ is already available) are required as shown in equation (4.8). The right-hand-side formation and integration of the reduced sensitivity equations are identical to the fixed-mode semianalytical method. Because of the nonzero $d\mathbf{\Phi}/dx$, recovery of the physical sensitivities is more complicated than in the fixed-mode case. Approximately twice the number of operations is required in the back transformation since both $\mathbf{\Phi}$ and $d\mathbf{\Phi}/dx$ terms must be considered as shown in equations (4.13). The total cost for the variable-mode semianalytical method is

$$C_{\text{saupd}} = C_{\text{model}} + 2\beta n_g + C_{\text{mmod}} + 4C_{\text{red}} + 2n_r n_g + 2n_r^2 n_t + C_{\text{inte}} + 2C_{\text{back}} \tag{6.18}$$

6.9.4. Semianalytical Mode Acceleration Method

Since $d\ddot{\mathbf{q}}/dx$ and $d\mathbf{q}/dx$ are obtained from the fixed-mode semianalytical method, the operations in equation (6.16) (except C_{back}) are required in applying the mode acceleration method. The back transformation operations for displacement and stress sensitivities are more complicated as seen in equations (4.19) and (4.20). The cost of forming the coefficients in equations (4.19) is dominated by multiplying a vector by a system matrix, adding $n_r + 1$

system vectors, and solving the system equations for $n_r + 1$ pseudostatic vectors. Again, the assumption is made that the model is undamped so that the $\overline{\mathbf{C}}$ and the $d\overline{\mathbf{C}}/dx$ terms in equations (4.19) are zero.

The back transformation procedure for displacement and stress sensitivities involves application of equations (4.19) and (4.20) for each quantity at the critical times. Velocity and acceleration terms are calculated as in the fixed mode, semianalytical method. Again, with only a single quantity for the number of back transformed quantities n_p, the cost can only be roughly estimated as $4C_{\text{back}}$. The total cost for the semianalytical, mode acceleration method can then then be written as

$$\begin{aligned} C_{\text{samacc}} = {} & C_{\text{model}} + 2\beta n_g + 2C_{\text{red}} + 2n_r^2 n_t \\ & + C_{\text{bmul}} + (n_r + 1)n_g + (n_r + 1)C_{\text{bsol}} \\ & + C_{\text{inte}} + 4C_{\text{back}} \end{aligned} \tag{6.19}$$

6.10. Analysis of Cost For Various Models

With the expressions for computational cost in the previous sections, it is now possible to evaluate the use of the sensitivity calculation methods on various examples. The first three examples are those considered in chapter 5. These three examples, however, are all rather small compared with the class of problems envisioned for the production use of the sensitivity methods. Accordingly, two other hypothetical problems with a larger number of degrees of freedom have been included.

The key parameters from the five problems required for the cost analysis are shown in table 6.1. Several points should be made about these parameters. The two beam problems have a small number of degrees of freedom and a very small bandwidth and, as a result, a small cost for system matrix factorization. This is unusual in finite element analysis. Medium model A and large model B represent a typical medium size linear dynamics problem and a rather large ambitious problem, respectively. Medium model A also is complicated by the fact that 100 vectors are assumed to be required in the transient analysis. In all five examples, a relatively large number of time steps are used in the transient analysis.

6.10.1. Cost of Computational Subtasks

Table 6.2 shows the number of floating point operations required for different computational tasks for the five example problems. Examining the costs for these subtasks gives some clues to the costs of different sensitivity calculation methods. For the first three examples, the cost of system matrix factorization is low. For the two larger hypothetical examples the factorization cost is much higher relative to that of other tasks. In the first three examples, the cost of integrating the reduced equations is substantial even though the equations are uncoupled. For models A and B, the integration cost is 1 to 2 orders of magnitude less than the other subtask costs in table 6.2. Consistently, in all five examples, the cost of performing the triple product basis reduction is high. For the three small problems, this cost is significantly higher than the factorization cost. For medium model A, this cost is also much higher than the factorization cost, but this is primarily due to the requirement of 100 vectors in the reduced system. Even in model B, however, the basis reduction cost is only a little less than one half the factorization cost. One conclusion is that the number of vectors in the reduced system substantially affects the cost of the analysis even if the vectors are not updated for the current model.

Table 6.1. Parameters Governing Computational Costs

Model	n_g	β	n_r	n_t	n_p	n_c
Five-span beam	32	3	18	6 000	25	10
Delta wing	264	30	20	30 000	13	5
Stepped beam	32	3	20	30 000	4	5
Medium model A	3 000	100	100	10 000	50	10
Large model B	12 000	300	30	20 000	200	10

The use of RWL vectors in the transient and sensitivity analyses was considered in chapter 5. Here, the cost of generating RWL vectors compared with vibration modes is considered. Table 6.2 shows the cost of system matrix eigensolution C_{eig} and RWL vector generation C_{RWL} for the five example problems. In every case the generation of RWL vectors is cheaper than the eigensolution. In the beam examples, C_{RWL} is more than an order of magnitude less than C_{eig}. This results from the unusual situation in which the number of required eigenvectors is nearly the same as the total number of degrees of freedom. In this case, the solution of the reduced eigenproblem artificially raises the cost of the system eigensolution. The other three examples show C_{eig} to be three or four times C_{RWL}. This is probably a much more accurate estimate of the cost savings obtained by using RWL vectors instead of eigenvectors.

Table 6.2. Number of Operations for Selected Computational Subtasks

Model	C_{bfac}	C_{red}	C_{eig}	C_{RWL}	C_{inte}
Five-span beam	2.7×10^2	9.5×10^3	6.6×10^5	1.8×10^4	8.6×10^5
Delta wing	1.2×10^5	3.8×10^5	4.8×10^6	1.1×10^6	4.8×10^6
Stepped beam	2.7×10^2	1.1×10^4	6.6×10^5	2.1×10^4	4.8×10^6
Medium model A	1.5×10^7	7.5×10^7	7.4×10^8	2.1×10^8	8.0×10^6
Large model B	5.4×10^8	2.2×10^8	4.0×10^9	1.2×10^9	4.8×10^6

6.10.2. Comparison of Costs for Five Sensitivity Methods

The primary objective of this chapter is to compare the costs in number of floating point operations of the sensitivity methods. This is summarized for five sensitivity methods, for the five examples in table 6.3. It is believed that these five sensitivity methods are all practical alternatives for large-order problems. This belief is substantiated by the fact that for all five examples the difference among the five costs is less than 1 order of magnitude.

Table 6.3. Overall Operation Costs for Five Sensitivity Methods

Model	C_{fdfix}	C_{fdupd}	C_{safix}	C_{saupd}	C_{samacc}
Five-span beam	1.0×10^6	1.5×10^6	4.8×10^6	4.8×10^6	4.8×10^6
Delta wing	6.5×10^6	1.0×10^7	3.0×10^7	3.2×10^7	3.1×10^7
Stepped beam	5.0×10^6	5.5×10^6	2.9×10^7	2.9×10^7	2.9×10^7
Medium model A	2.4×10^8	7.8×10^8	3.9×10^8	6.8×10^8	4.5×10^8
Large model B	8.2×10^8	4.4×10^9	8.5×10^8	1.7×10^9	1.1×10^9

The forward difference method with fixed modes is consistently the cheapest method. However, this low computational cost must be weighted against the pitfalls of the method discussed in chapter 5. The cost of a fixed-mode central difference method which is approximately twice the forward difference cost would also be quite competitive with the other methods and would lessen the sensitivity to finite difference step size. For the two larger problems, the forward difference method with updated modes is relatively expensive and an updated-mode central difference method would be extremely expensive for larger problems.

In the three smaller problems the semianalytical methods require significantly more operations than the finite difference methods. This is primarily because the larger number of time steps makes the calculation of the right-hand-side pseudo load relatively large. For the two larger problems, however, the fixed-mode semianalytical method is quite competitive with the forward difference method. For model A, it is less than twice the cost of the finite difference method, and for model B, it is essentially the same.

The number of basis vectors used is a key parameter in both the analysis and sensitivity calculations. Table 6.1 shows the number of modes used in the baseline cost analyses for the five examples. Here, the effect of the number of modes on the overall sensitivity costs is considered. First, the delta wing example, which is representative of a typical small problem, is considered. Shown in figure 6.1 is the cost for the five methods plotted as a function of number of modes used. The number of modes ranges from 20 to 100. The values of the other parameters in the problem are in table 6.1. The key result from figure 6.1 is that the semianalytical methods are much more costly than the finite difference methods for large numbers of modes. There are two reasons for this: first, because the problem is small, calculation of the vibration modes is relatively cheap, and second, because there are a large number of time steps, formation of the right-hand side in the sensitivity equations for the semianalytical methods is quite costly when the number of modes used is large.

For the large model B example, the result of varying the number of modes is very different. For this example, the cost of the five sensitivity methods plotted as a function of number of modes is shown in figure 6.2. In this example, the calculation of the modes is a very costly operation. Accordingly, the forward difference method with updated modes is substantially more costly than the other methods for large numbers of modes. The fixed-mode forward difference and semianalytical methods show only moderate increases in cost as the number of modes is increased.

It was mentioned above that a relatively large number of time steps are used in the five examples. The effect of the number of time steps on the overall sensitivity calculation costs is considered here. The delta wing example is considered as representative of a small problem; the large model B example, of a large problem. For the delta wing, the computational costs for the five sensitivity methods are plotted as a function of the number of time steps in figure 6.3. The values of the sensitivity calculation costs here are similar to those in figure 6.1; the forward difference methods show only moderate cost increases for larger numbers of time steps and the semianalytical methods show substantial cost increases. Again, the reason is that the right-hand-side formation in the semianalytical methods is a substantial part of the total cost in small problems.

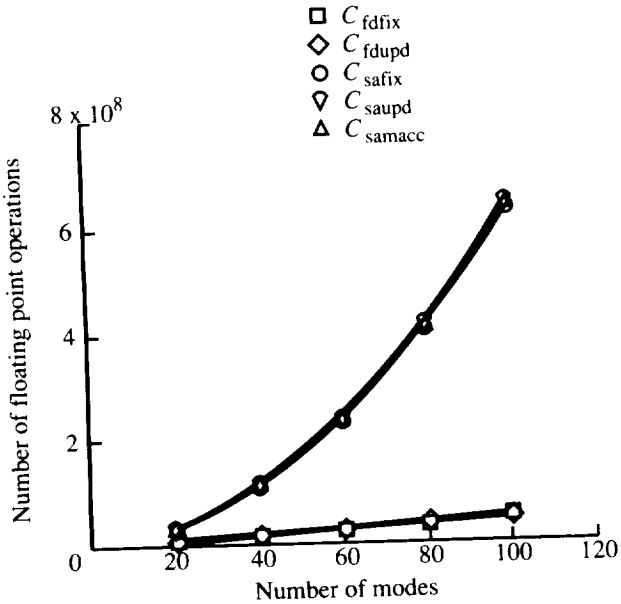


Figure 6.1. Cost of sensitivity calculation methods as function of number of modes for delta wing example.

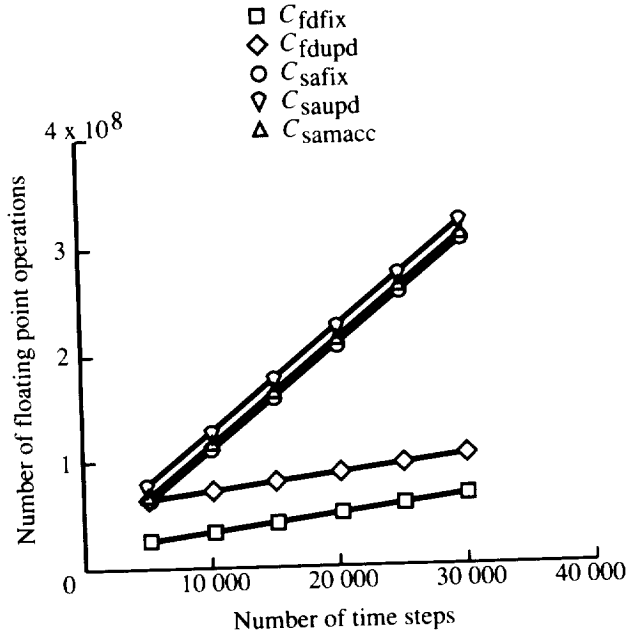


Figure 6.3. Cost of sensitivity calculation methods as function of number of time steps for delta wing example.

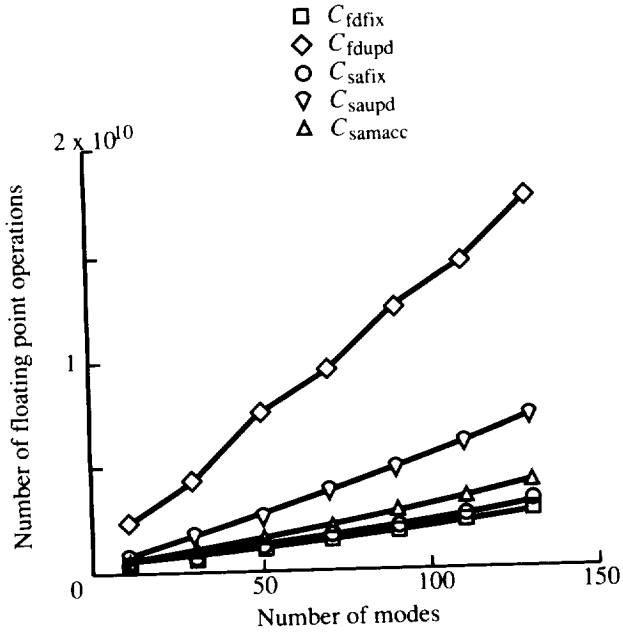


Figure 6.2. Cost of sensitivity calculation methods as function of number of modes for model B example.

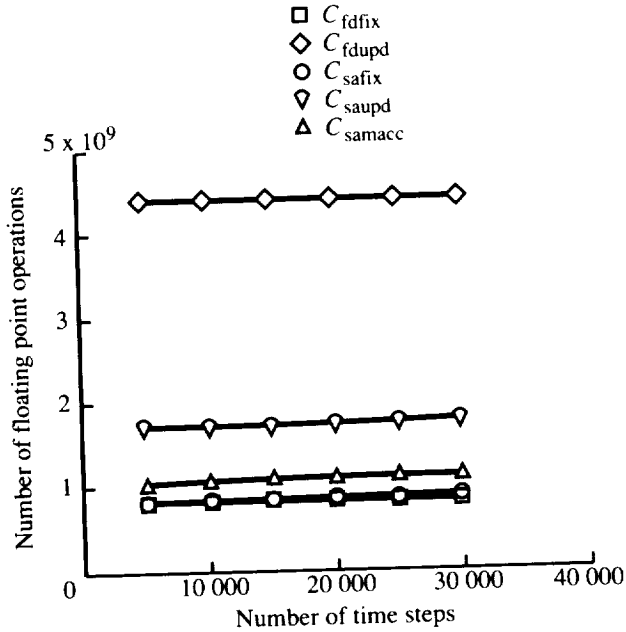


Figure 6.4. Cost of sensitivity calculation methods as function of number of time steps for model B example.

The cost results from a large problem, the model B example, are very different, however. The cost as a function of number of time steps is plotted in figure 6.4. Obviously, there is practically no change in cost for any of the methods as a function of number of time steps. For this large problem, both the cost of integrating the uncoupled equations and the cost of forming the right-hand side in the semianalytical methods are 2 to 3 orders of magnitude less than the total cost.

6.11. Summary

The main objective of this chapter is a comparison of the computational costs in number of floating point operations of the sensitivity calculation methods. Five example problems were considered—the three example problems from chapter 5, which are all fairly small and two larger hypothetical examples.

Many of the results depend significantly on whether the problem is one of the three smaller examples or one of the two larger hypothetical examples. In the three smaller examples, the cost of system matrix factorization is low, whereas in the larger problems, this cost is quite high. When the cost of factorization is high, the system eigenproblem is especially costly. In the smaller problems, operations repeated for the reduced problem at each time step (such as integration of the uncoupled equations) are a significant percentage of the total sensitivity calculation cost. For large problems, the relative cost of these operations is small.

For all five examples, the forward difference method with fixed modes was the cheapest. For the smaller problems the forward difference method with updated modes had a relatively low cost, but for the larger problems the cost was quite high. For the larger problems the semianalytical method with fixed modes and the semianalytical mode acceleration method have costs that are relatively competitive with the fixed-mode forward difference method. In all cases, the semianalytical method with approximate eigenvector derivatives was one of the more costly methods.

It was shown in chapter 5 that for two examples the accuracy of the stress sensitivities for small numbers of basis vectors was extremely poor. It was demonstrated that the semianalytical mode acceleration method was one means of dramatically improving this accuracy. From the results of this chapter, the semianalytical mode acceleration method is only slightly more costly than the fixed-mode forward difference and semianalytical methods. Given the unacceptable accuracy of these fixed-mode methods for two of the examples, the semianalytical mode acceleration method appears to be the method of choice.

Chapter 7

Concluding Remarks

Several methods have been developed and evaluated for calculating sensitivities of displacements, velocities, accelerations, and stresses in linear, structural, transient response problems. Two of the methods are overall finite difference methods where the analysis is repeated for perturbed designs. The other methods are termed semianalytical methods because they involve direct analytical differentiation of the equations of motion with finite difference approximations of the coefficient matrices. The different sensitivity methods were evaluated by applying them to three example problems: a five-span simply supported beam loaded with an end moment, an aircraft wing loaded with a distributed pressure, and a cantilever beam with a stepped cross section loaded with an applied root angular acceleration.

An important issue in calculating transient response sensitivities for use in formal optimization procedures is how to define the constraints. Two common approaches are to integrate the response quantity over time or to pick the maximum (or minimum) value of the response quantity in time. Both these approaches have drawbacks. An alternative critical point constraint approach was implemented which identifies the most important response points along the time history. A method for identifying these critical points was devised that, based on the three examples considered, appears to be very effective even for very jagged response histories.

All the analyses and sensitivity methods considered use approximation vectors to reduce the number of degrees of freedom in the analysis. Vibration mode shapes, Ritz-Wilson-Lanczos vectors, and static displacement shapes were used in the analysis and sensitivity calculations. The key question when an approximate reduced basis is used in an analysis is how many basis vectors are required for an accurate approximation to the finite element solution. It was generally found that, if the accuracy of the response quantities was poor, the accuracy of the sensitivities was extremely poor. In a number of cases, however, even though the accuracy of the response quantities was adequate, the accuracy of sensitivities was poor. This is discussed further below. In all cases considered herein, the accuracy as a function of the number of vectors for both the response quantities and sensitivities with Ritz-Wilson-Lanczos vectors was as good or better than with vibration modes. Since the generation of Ritz-Wilson-Lanczos vectors is cheaper than vibration modes, they appear to provide a more cost-effective alternative to modes in many cases.

A goal in considering sensitivity methods in this study is that they be suitable for very large-order finite element analysis. In these types of problems, a complete vibration analysis for each perturbed model is impractical because of the high computational cost. To reduce this cost, one approach which was studied herein is to use the basis vectors from the initial model to approximate the response in the perturbed model. This often provides an effective solution. In two of the three examples problems considered, however, using the initial vectors in an overall finite difference method or assuming fixed modes in a semianalytical method resulted in very poor modal convergence for stress sensitivities. Two methods were devised to improve this poor performance.

The first method retains the derivatives of the basis vectors in the sensitivity equations but approximates these derivatives rather than using a very costly exact computation. One well-known method for approximating eigenvector derivatives, the modal method, was found to be completely ineffective because it adds no new information to the existing modal basis. Another technique, the modified modal method, adds a pseudostatic contribution to the eigenvectors in approximating the eigenvector derivatives. This technique, along with the semianalytical method, was found to be very effective in improving the poor accuracy of the stress sensitivities.

A second method for improving the accuracy of the stress sensitivities as a function of the number of modes is to use a mode acceleration version of the semianalytical method. The key to the mode acceleration method in the transient analysis is that it supplements the modal basis with a static contribution calculated from the complete model. The key to the mode acceleration implementation of the semianalytical sensitivity method is that it supplements the modal basis with pseudostatic sensitivity terms calculated from the complete model. This technique produced the same dramatic improvement in the accuracy of stress sensitivities as the semianalytical modified modal method.

As mentioned, computational cost was an overriding concern in considering the sensitivity analysis methods. To estimate this cost, expressions for the number of floating point operations in each of the methods were derived. Although this approach does not include important effects such as overhead

operations or disk input/output that would be present in a practical implementation of these methods, it does provide a mechanism for an approximate coarse ranking of the methods by computational cost. The overall forward difference method with fixed basis vectors was found to be the cheapest method for all cases considered. This technique, however, suffers from the accuracy problems previously mentioned. One approach to alleviating these accuracy problems is to recalculate the modes for the perturbed model (updated modes) in the overall forward difference method. This forward difference method with updated modes was found to be very costly for large models, however. The fixed-mode semianalytical method is only slightly more costly than the overall forward difference method with fixed modes but suffers from the same accuracy problem as the fixed-mode overall forward difference method. Two techniques with reasonable costs that alleviate the accuracy problem are the mode acceleration implementation of the semianalytical method and the semianalytical method with approximate mode shape derivatives. Of these two methods, the semianalytical mode acceleration method is slightly cheaper.

Given the high accuracy of the semianalytical mode acceleration method for a relatively small number of modes and its reasonable computational cost, this appears to be the method of choice. In the three examples considered herein, this method consistently performed as well as the much more costly, updated-mode overall finite difference methods. Furthermore, the insensitivity of this and the other semianalytical methods to finite difference step size makes this semianalytical mode acceleration method especially attractive.

NASA Langley Research Center
Hampton, VA 23665-5225
February 1, 1990

Acknowledgment

I would like to thank Professor Raphael T. Haftka of the Virginia Polytechnic Institute and State University for his numerous suggestions throughout the course of this research.

Appendix

Computer Implementation

The methods for calculating sensitivities and the example problems have been implemented with the general purpose finite element code, EAL (ref. 23). EAL includes general language constructs for controlling execution flow as well as general and specific utilities for manipulating data stored as named entities in a data base. It also allows procedures (called "runstreams") to be defined and then explicitly executed. Most of the implementation was done with EAL runstreams. However, some parts of the implementation could not be conveniently done with runstreams and were coded as Fortran additions to EAL. The Fortran additions are described in the next section. The runstreams for the algorithms and example problems are included and described also.

Additions to EAL

The transient response module in EAL version 312 solves the uncoupled form of equations (2.5) with the matrix series expansion method. A modification was made to allow equations (2.5) to be fully coupled. In the semianalytical method, the right-hand-side pseudo loading of equations (4.6) can be easily formed with EAL. However, a slight modification to the transient response module was required to permit solution of equations (4.6) with this general form of loading. In addition, a special purpose module was added to EAL to perform the task of identifying the critical points on each response function.

Runstream for Stepped Beam Example Problem

The runstream for the stepped beam is included to illustrate how the sensitivity calculation runstreams are used. At the beginning of the runstream, the data set `XFLG ADS` indicates which subset of the possible design variables will be considered in the sensitivity analysis. The data sets `X ADS` and `XNAME ADS` contain the initial values and the register names of all the design variables, respectively. Various parameters controlling the analysis and sensitivity calculations are defined in runstream data sets `TR PARAMETERS`, `DXDV PARAMETERS`, and `BACK METHOD`. The runstream data set `MODEL` defines the model in terms of the design variables in `X ADS`. It is called before the initial dynamic analysis and at least once for each design variable considered in the sensitivity analysis. The runstream data set `DYNAM SOLN` is called once to perform the dynamic analysis of the initial model. The runstream data set `PLOT RESP` illustrates the interface to a useful utility runstream `TR PLOT` for automatically generating plots of response quantities as a function of time. `TR PLOT` is called once for each class (e.g., accelerations) of response quantity to be plotted. The actual sensitivity analysis is performed by calling the runstream `TR DXDV` n where the n is associated with the particular sensitivity calculation method.

```
*CM=120000
$-----------------------------------------------------------------
$ RUNSTREAM FOR STEPPED BEAM EXAMPLE
$-----------------------------------------------------------------
*XQT EXTE
! SYST = SSP(4,5) $ GET SYSTEM TYPE
*XQT U1
```

```
 *INF=7
 *CLIB=29
 *(ALL) ALL
 *XQT AUS
  TABLE(NI=1,NJ=9,TYPE=0) : XFLG ADS
  J=1 : 1
  J=5 : 1
  J=6 : 1
  J=9 : 1
 *XQT U1
 *TI(X ADS)
  23.5
  22.0
  20.0
  18.0
  16.5
 $
  40.0
  80.0
  120.0
  160.0
 *TI(XNAME ADS)
  H1 : H2 : H3 : H4 : H5
  XL1 : XL2 : XL3 : XL4
 *(TR,PARAMETERS)
  QLIB=1
  MNAME=CEM
  NMODES=5
  DT=1.0E-5
  T2 = .3
  DRFORMAT=DIAG
  METHOD=MODES
  DXDV=0
  EIGEN=1
  PRINT=1
  VLIB=1
  NCRIT=5
  CONV=1.E-10
  BLKSIZE=2000
  NTERMS=50
 *(DXDV,PARAMETERS)
  FDCH=1.0E-5
  FDMCH=1.0E-6
  DXMD=FIXED
 *TI(BACK METHOD)
  2 $ DISP
  1 $ VELOCITIES
  1 $ ACCELERATIONS
  1 $ REACTIONS
  2 $ STRESSES
 *(MODEL)                          END
 ! LEN=200.
 ! NEPS = 3
 ! NEL = NEPS*5
 ! NNODE = NEL + 1
 *XQT AUS
  TABLE(NI=1,NJ=5) : XX1
  I=1 : J=1,5 : 0. "XL1" "XL2" "XL3" "XL4"
  TABLE(NI=1,NJ=5) : XX2
  I=1 : J=1,5 : "XL1" "XL2" "XL3" "XL4" "LEN"
  D1 = SUM(XX2 -1.0 XX1)
 ! RNEP = 1.0/NEPS
  DELX = UNION("RNEP" D1)
 *XQT TAB
  START "NNODE"  1 3 4 5
  JLOC
```

```
$1 0. 0. 0. 200. 0. 0. "NNODE"
! J = 1 : ! X = 0.0
! I1 = 1 : ! N1 = 5
*LABEL 5
! DELX = DS,1,"I1",1(1 DELX AUS 1 1)
! I2 = 1 : ! N2 = NEPS
*LABEL 8
 "J" "X" 0.0 0.0
! J = J + 1 : ! X = X + DELX
*JGZ,-1(N2,8)
! I1 = I1 + 1
*JGZ,-1(N1,5)
 "NNOD" "LEN" 0. 0.
 CON 1
 ZERO 1,2,6 : 1
 MATC
 1 30.+6 .3 .3
 BA
 RECT 1 1.20 "H1"
 RECT 2 1.10 "H2"
 RECT 3 1.00 "H3"
 RECT 4  .90 "H4"
 RECT 5  .85 "H5"
 RECT 6 1.0 20.0
 MREF
 1 1 2 1 1.0
*XQT ELD
 E21
! N = 5
! I = 1
! N1 = 1
*LABEL 20
 NSECT = "I"
! N2 = N1 + 1
 "N1" "N2" 1 "NEPS"
! N1 = N1 + NEPS
! I = I + 1
*JGZ,-1(N,20)
*XQT E
 RESET G=386.
*XQT EKS
*XQT TAN
*XQT K
*XQT M
 RESET G=386.
*XQT AUS
 R = RIGID(1)
 DEFINE R6 = R AUS 1 1 6,6
 APPL FORC = PROD(-1.0 CEM R6)
*END
*(DYNAM,SOLN) END
*DCALL(TR,VECTORS)
*XQT U1
*TI(SEL DISP)
 "NNODE" 2
*TI(SEL VELO)
 "NNODE" 2
*TI(SEL ACCE)
 "NNODE" 2
*TI(SEL STRE)
 E21 1 1 SZ1 1 0
*XQT AUS
$ DEFINE SOME MODAL DAMPING
 TABLE(NI=1,NJ="NMODES") : DRAT
 I=1 : J=1,"NMODES" : .005
*DCALL(SQUARE LOAD) RTIME=.18 RANG=10.0
```

```
 *DCALL(TR,MAIN)
 *                                    END
 *(SQUARE,LOAD) END
 *XQT AUS
 ! RT2 = RTIME/2.0
 ! EPS = 0.0
 ! RT2M = RT2 - EPS
 ! RT2P = RT2 + EPS
 ! RTM = RTIME - EPS
 ! D2R = 3.1415926/180.
 ! RRAD = RANG*D2R
 ! AMP = 4.0*RRAD/RTIME/RTIME
 ! MAMP = -AMP
  TABLE(NJ=6) : TIME
  J=1,6 : 0. "RT2M" "RT2P" "RTM" "RTIME" 10000.0
  TABLE(NJ=6) : CA
  J=1,6 : "AMP" "AMP" "MAMP" "MAMP" 0.0 0.0
*END
*(PLOT RESP) END
*XQT DCU
 CHANGE 1 A AUS MASK MASK HIST CA 1 1
*XQT AUS
 ALPHA : FTITLE
1' HISTORY OF FORCE MULTIPLIER G(T)
 ALPHA : DTITLE
1' TIP DISPLACEMENT HISTORY FOR CANTILEVER BEAM
 ALPHA : TVTITLE
1' TIP VELOCITY HISTORY FOR CANTILEVER BEAM
 ALPHA : TATITLE
1' TIP ACCELERATION HISTORY FOR CANTILEVER BEAM
 ALPHA : STITLE
1' BENDING STRESS AT THE ROOT FOR THE CANTILEVER BEAM
! TLIB = 15
*XQT U1
*(TRPLOT OPTIONS)
! YNAME = 'CA
! TITLE = 'FTITLE
! ID = 1
*DCALL(TR,PLOT)
*XQT U1
*(TRPLOT OPTIONS)
! YNAME = 'DISP
! IDJK = 'DISP
! TITLE = 'DTITLE
! ID = "NMODES"
*DCALL(TR,PLOT)
*XQT U1
*(TRPLOT OPTIONS)
! YNAME = 'VELO
! IDJK = 'VELO
! TITLE = 'TVTITLE
! ID = "NMODES"
*DCALL(TR,PLOT)
*XQT U1
*(TRPLOT OPTIONS)
! YNAME = 'ACCE
! IDJK = 'ACCE
! TITLE = 'TATITLE
! ID = "NMODES"
*DCALL(TR,PLOT)
*XQT U1
*(TRPLOT OPTIONS)
! YNAME = 'STRE
! IDQ = 'STRE
! TITLE = 'STITLE
! ID = "NMODES"
```

```
*DCALL(TR,PLOT)
*END
*RGI
*DCALL(TR,PARAMETERS)
*DCALL(SENS,DVUP)
*DCALL(MODEL)
*DCALL(DYNAM,SOLN)
$*DCALL(PLOT,RESP)
*IF("DXDV" NE 0): *DCALL(TR,DXDV,"DXDV")
*ALL
*IF("SYST" EQ CDC ): *PRT(ALL)
*IF("SYST" EQ CNVX): *PRT(ALL)
*DCALL(ALL)
*XQT EXIT
```

Runstreams for Sensitivity Methods

Runstream TR MAIN

TR MAIN is the main runstream for performing the transient response analysis and is based on a similar runstream produced by EISI. It is used only for the transient analysis of the initial model and not for the sensitivity calculations.

```
$-----------------------------------------------------------------
$ (TR MAIN) - MAIN DRIVER FOR TRANSIENT RESPONSE ANALYSIS
$-----------------------------------------------------------------
* XQT U1
* REGISTER STORE(29 TR REGISTERS 1 1)
* REGISTER RETR (29 TR REGISTERS 1 1)
* RGI
$       DEFAULT REGISTERS:
           QLIB  = 2    $ SOURCE FOR EXCITATION, DESTINATION LIB FOR RESPONSE
           VLIB  = 1    $ SOURCE LIB FOR VIBR MODE AND VIBR EVAL DATASETS
           VSET  = 1    $ USE "VLIB" VIBE MODE "VSET" "VCON" FOR THE RITZ
           VCON  = 1    $     FUNCTIONS
           KNAME = K    $ STIFFNESS MATRIX
           MNAME = DEM  $ MASS MATRIX
           DAMP  = DAMP $ NAME OF SPAR FORMAT DAMPING MATRIX
           FSET  = 1    $ EXCITATION SET NUMBER
           NAME  = AUS  $ 2ND WORD OF TIME "NAME" AND CA "NAME"
           DT    = 0.   $ SET TIME INCREMENT
           T2    = 0.   $ FINAL INTEGRATION TIME
           DRFORM= DIAG $ FORMAT FOR THE REDUCED MATRICES (DIAG,FULL,RITZ)
           DRMETH= 0    $ TIME INTEGRATION METHOD (0=MSE)
           NTER  = 50   $ SET NUMBER OF TERMS IN MATRIX SERIES EXPANSION
           NMODES= 0    $ NUMBER OF MODES USED IN DYNAMIC ANALYSIS (DEFAULT=ALL)
           BLKSIZ= 6000 $ BLOCK SIZE FOR OUTPUT DATASETS
           EIGEN = 0    $ EIGENVALUE ANALYSIS OF DAMPED SYSTEM
           PRINT = 0    $ PRINT FLAG FOR DTEX
           OPT=0,  PROC=MAIN,  NERR=0
 * DCALL,OPT (TR PARAMETERS)
 ! ZERO = SSP(0,10)
 *IF("NMODES" EQ 0): ! NMODE=TOC,NBLOCK("VLIB" VIBR MODE "VSET" "VCON")
 $
 $ COMPUTE DATASETS REQUIRED FOR DR/DTEX, /TR1, AND /BACK:
 $
 * CALL (TR PREP)
 $
 $ COMPUTE THE MODAL RESPONSE:
 $
 * XQT DRX
 * IF("DT" GT 1.E-20): *GOTO 20
           DTEX(INLI="QLIB",N2="NAME",OUTL="QLIB",EIGEN="EIGEN",>
                PRINT="PRINT")
 *GOTO 30
```

```
*LABEL 20
          DTEX(INLI="QLIB",N2="NAME",OUTL="QLIB",DT="DT",>
               NTER="NTER",EIGEN="EIGEN",PRINT="PRINT")
*LABEL 30
                    TR1 (INLIB="QLIB",N2="NAME",CASE="FSET",ALIB="QLIB">
                    QXLIB="QLIB",QX1LIB="QLIB",QX2LIB="QLIB",>
                    T2="T2",LB="BLKSIZ")
$
$ BACK TRANSFORMATION:
$
*XQT AUS
! NBCK = 5
 TABLE(NI="NBCK", NJ=1, TYPE=4) : "QLIB" BACK LIST
 J=1 : DISP VELO ACCE REAC STRE
$
EXIT
! I = 1 : ! N = NBCK
*LABEL 50
! BKMETH = DS,1,"I",1("QLIB" BACK METH 1 1)
! NM = DS,"I",1,1("QLIB" BACK LIST 1 1)
! IERR = TOC,IERR("QLIB" SEL "NM" MASK MASK)
*IF("IERR" EQ 0): *CALL (TR "NM" "BKMETH")
! I = I + 1
*JGZ,-1(N,50)
$
$ EXIT:
* XQT U1
* REGISTER RETRIEVE(29 TR REGISTERS 1 1)
*END
```

Runstream TR PREP

This Runstream TR PREP is used to define the reduced equations for the transient analysis and also to prepare data for the back transformation phase. It is based on an EISI runstream of the same name. It is used only for the transient analysis of the initial model and not in the sensitivity calculations.

```
$-----------------------------------------------------------------
$ (TR PREP) - PREPARATION OF REQUIRED DATASETS
$-----------------------------------------------------------------
* XQTC U1
* RGI
          PROC=PREP, NERR=0, MOTI=MOTI, FORC=FORC
$
$ CHECK FOR THE REQUIRED DATASETS AND DETERMINE THE TYPE OF EXCITATION:
$ !TYPE=0 FOR APPLIED FORCE.  !TYPE=1 FOR APPLIED DISPLACEMENT.
$
         !TYPE=2
         !IERR=TOC,IERR("QLIB" APPL FORC "FSET" MASK)
* IF("IERR" NE 0): *GO TO 100
         !TYPE=0: !FNAME='FORC   $$ APPLIED FORCE EXCITATION
* LABEL 100
         !IERR=TOC,IERR("QLIB" APPL MOTI "FSET" MASK)
* IF("IERR" NE 0): *GO TO 200
* IF("TYPE" EQ 0): !NERR=1          $$ FORCE & DISP SPECIFIED          $ NERR=1
         !TYPE=1: !FNAME='MOTI   $$ APPLIED DISPLACEMENT EXCITATION
* LABEL 200
* IF("TYPE" EQ 2):!NERR=2          $$ NO EXCITATION SPECIFIED         $ NERR=2
* IF("NERR" NE 0):*CALL (TR ERROR)
         !IERR=TOC,IERR("VLIB" VIBR MODE "VSET" "VCON"): !NERR=3
* IF("IERR" NE 0):*CALL (TR ERROR)                                    $ NERR=3
         !IERR=TOC,IERR("VLIB" VIBR EVAL "VSET" "VCON"): !NERR=4
* IF("IERR" NE 0):*CALL (TR ERROR)                                    $ NERR=4
         !IERR=TOC,IERR("QLIB" TIME "NAME" "FSET" MASK): !NERR=5
* IF("IERR" NE 0):*CALL (TR ERROR)                                    $NERR=5
```

```
         !IERR=TOC,IERR("QLIB" CA   "NAME" "FSET" MASK): !NERR=6
* IF("IERR" NE 0):*CALL (TR ERROR)                                    $NERR=6
$
$ COMPUTE XTMX, XTKX, XTDX, & XTF FOR DR/DTEX & TR1:
$
          !IERR=TOC,IERR(1 INV "KNAME" "VCON" MASK)
* IF ("IERR" EQ 0):*GO TO 250
* XQT DRSI
           RESET K="KNAME",CON="VCON"
* LABEL 250
$
*XQT AUS
           OUTLIB="QLIB": INLIB="QLIB"
           DEFINE X="VLIB" VIBR  MODE   "VSET" "VCON"
           DEFINE E="VLIB" VIBR  EVAL   "VSET" "VCON"
           DEFINE F="QLIB" APPL "FNAME" "FSET"
           DEFINE K=  1   "KNAME"
           DEFINE M=  1   "MNAME"
* IF("TYPE" EQ 0): *GO TO 300
           KS=PROD("KNAME" -1. F): DEFINE F=KS
* LABEL 300
           XTF "NAME" "FSET"= XTY(X,F)
$
! IDMD = TOC,IERR("VLIB" DRAT MASK MASK MASK)
*IF("IDMD" NE 0): *GOTO 400
           DEFINE D = "VLIB" DRAT
           OMEG = SQRT(E)
           DMPD = PROD(2.0 D OMEG)
*LABEL 400
          !IERR=TOC,IERR("QLIB" XTMX "NAME" MASK MASK)
* IF("IERR" EQ 0): *GO TO 500
$
*CALL(TR,REDM)
* LABEL 500
$
$ COMPUTE ("QLIB" STAT DISP "FSET" "VCON"):
$
          !IERR=TOC,IERR("QLIB" STAT DISP "FSET" "VCON")
* IF("IERR" EQ 0): *GO TO 700
* XQT SSOL
           RESET SET="FSET",CON="VCON",QLIB="QLIB"
* LABEL 700
$
*LABEL 1000
*END
```

Runstream TR DISP

The two TR DISP runstreams do the back transformation for displacements. TR DISP 1 performs the back transformation with the mode displacement method and TR DISP 2 uses the mode acceleration method. This naming convention is used for the other runstreams that perform the back transformation operation for other response quantities. The TR DISP runstreams and companion runstreams for velocities, accelerations, and stresses perform the back transformation at all time steps. Accordingly they are used only in the dynamic analysis of the initial model and not in the sensitivity analysis.

```
$-----------------------------------------------------------------
$ (TR DISP 1) - BACK TRANSFORMATION FOR DISPLACEMENTS
$               MODE DISPLACEMENT METHOD
$-----------------------------------------------------------------
*XQTC AUS
           OUTLIB="QLIB": INLIB="QLIB"
           DEFINE IDJK      = "QLIB" SEL   DISP
           DEFINE X = "VLIB" VIBR MODE "VSET" "VCON"
```

```
           TMAT VMOD=SVTRAN(IDJK,X)
 *XQT DRX
           BACK(LRZ="BLKSIZE")
           T = +1.0 "QLIB" TMAT VMOD : Y = "QLIB" QX
           Z= "QLIB" HIST DISP
           EXT =   "QLIB" EXT  DISP "FSET"
 *END

 $-----------------------------------------------------------------
 $ (TR DISP 2) - BACK TRANSFORMATION FOR JOINT DISPLACEMENTS
 $             MODE ACCELERATION METHOD
 $-----------------------------------------------------------------
 $
 $ TRANSIENT RESPONSE: BACK TRANSFORMATION FOR JOINT DISPLACEMENTS
 $ APPLICABLE FOR APPLIED FORCE OR DISPLACEMENT EXCITATION
 $
 $ REGISTERS: QLIB, NAME, FSET, VCON
 $
 * XQT AUS
 ! ID = TOC,IERR("QLIB" XTDX MASK MASK MASK)
           OUTLIB="QLIB": INLIB="QLIB"
           DEFINE E = "VLIB" VIBR EVAL
           ROMG = RECIP(E)
           DEFINE IDJK      = "QLIB" SEL   DISP
           DEFINE XS        = "QLIB" STAT  DISP "FSET" "VCON"
           DEFINE X         = "VLIB" VIBR  MODE    1   "VCON"
           DEFINE DACC = TMAT DACC
           XOME = CBD(X,ROMG)
           TMAT DACC  = SVTRAN(IDJK,XOME)
           TMAT DS    = SVTRAN(IDJK,XS)
*JNZ(ID,30)
$ DAMPING TERM
! NJ = TOC,NJ("QLIB" XTDX MASK MASK MASK)
! NINJ = TOC,NINJ("QLIB" XTDX MASK MASK MASK)
*IF("NJ" NE "NINJ"): *GOTO 10
$ MODAL DAMPING
          XOMD = CBD(XOME,XTDX)
          TMAT DVEL = SVTRAN(IDJK,XOMD)
*GOTO 30
*LABEL 10
$ GENERAL DAMPING
          TMAT DVEL = RPROD(DACC,XTDX)
*LABEL 30
* XQT DRX
           BACK(LRZ="BLKSIZE")
           T = +1. "QLIB" TMAT DS : Y = "QLIB"  A  "NAME" "FSET"
*IF("ID" EQ 0): T=-1. "QLIB" TMAT DVEL : Y="QLIB" QX1 "NAME" "FSET"
           T = -1. "QLIB" TMAT DACC : Y = "QLIB" QX2 "NAME" "FSET"
           Z   =   "QLIB" HIST DISP "FSET"
           EXT =   "QLIB" EXT  DISP "FSET"
*END
```

Runstream TR VELO

```
$-----------------------------------------------------------------
$ (TR VELO 1) - BACK TRANSFORMATION FOR VELOCITIES
$               MODE DISPLACEMENT METHOD
$-----------------------------------------------------------------
*XQTC AUS
           OUTLIB="QLIB": INLIB="QLIB"
           DEFINE IDJK      = "QLIB" SEL   VELO
           DEFINE X = "VLIB" VIBR MODE "VSET" "VCON"
           TMAT VVEL=SVTRAN(IDJK,X)
*XQT DRX
           BACK(LRZ="BLKSIZE")
           T = +1.0 "QLIB" TMAT VVEL : Y = "QLIB" QX1
```

```
          Z= "QLIB" HIST VELO
          EXT =   "QLIB" EXT  VELO "FSET"
*END
```

Runstream TR ACCE

```
$-----------------------------------------------------------------
$ (TR ACCE 1) - BACK TRANSFORMATION FOR ACCELERATIONS
$               MODE DISPLACEMENT METHOD
$-----------------------------------------------------------------
*XQTC AUS
           OUTLIB="QLIB": INLIB="QLIB"
           DEFINE IDJK      = "QLIB" SEL   ACCE
           DEFINE X = "VLIB" VIBR MODE "VSET" "VCON"
           TMAT VACC=SVTRAN(IDJK,X)
*XQT DRX
           BACK(LRZ="BLKSIZE")
           T = +1.0 "QLIB" TMAT VACC : Y = "QLIB" QX2
           Z= "QLIB" HIST ACCE
           EXT =   "QLIB" EXT  ACCE "FSET"
*END
```

Runstream TR STRESS

```
$-----------------------------------------------------------------
$ (TR STRESS 1)
$-----------------------------------------------------------------
$
$ MODE DISPLACEMENT STRESS BACK TRANSFORMATION
$
*XQT ES
 RESET OPER=T
 IDQ= "QLIB" SEL STRESS
 U = "VLIB" VIBR MODE 1 "VCON" 1,"NMODE"
 T = "QLIB" TMAT VSTRE "FSET"
*XQT DRX
           BACK(LRZ="BLKSIZE")
 T = +1. "QLIB" TMAT VSTRE "FSET" : Y = "QLIB" QX "NAME" "FSET"
 Z = "QLIB" HIST STRESS "FSET"
 EXT = "QLIB" EXT STRESS "FSET"
*END
$-----------------------------------------------------------------
$ (TR STRESS 2)
$-----------------------------------------------------------------
$
$ TRANSIENT RESPONSE: BACK TRANSFORM FOR ELEMENT STRESSES
$ APPLICABLE FOR APPLIED FORCE OR DISPLACEMENT EXCITATION
$
$ REGISTERS: QLIB, NAME, FSET, VCON
$
* XQT AUS
! ID = TOC,IERR("QLIB" XTDX MASK MASK MASK)
           OUTLIB="QLIB": INLIB="QLIB"
           DEFINE E = "VLIB" VIBR EVAL
           ROMG = RECIP(E)
           DEFINE IDJK      = "QLIB" SEL   DISP
           DEFINE XS        = "QLIB" STAT  DISP "FSET" "VCON"
           DEFINE X         = "VLIB" VIBR  MODE    1   "VCON"
           XOME = CBD(X,ROMG)
*JNZ(ID,30)
$ DAMPING TERM
! NJ = TOC,NJ("QLIB" XTDX MASK MASK MASK)
! NINJ = TOC,NINJ("QLIB" XTDX MASK MASK MASK)
*IF("NJ" NE "NINJ"): *GOTO 10
$ MODAL DAMPING
```

```
          XOMD = CBD(XOME,XTDX)
 *GOTO 30
 *LABEL 10
 $ GENERAL DAMPING
          XOMD=CBR(XOME,XTDX)
 *LABEL 30
 *XQT ES
           RESET OPER=T
           IDQ = "QLIB" SEL STRESS
           U= "QLIB" STAT DISP "FSET" "VCON" : T= "QLIB" TMAT SF
 *IF("ID" EQ 0): U= "QLIB" XOMD :                T= "QLIB" TMAT SD
           U= "QLIB" XOME :                      T= "QLIB" TMAT SP
 * XQT DRX
           BACK(LRZ="BLKSIZE")
                T = +1. "QLIB" TMAT SF : Y = "QLIB"  A  "NAME" "FSET"
 *IF("ID" EQ 0): T=-1. "QLIB" TMAT SD :   Y = "QLIB" QX1 "NAME" "FSET"
                T = -1. "QLIB" TMAT SP : Y = "QLIB" QX2 "NAME" "FSET"
           Z   =   "QLIB" HIST STRE "FSET"
           EXT =   "QLIB" EXT  STRE "FSET"
 *END
```

Runstream TR ERROR

```
$-----------------------------------------------------------------
$ (TR ERROR)
$-----------------------------------------------------------------
$
$ THIS PROCEDURE PRINTS FATAL ERROR MESSAGES FOR THE TR PROCS.
$
* XQT U3
          RP2: NUMBER OF FORMATS=10
          FORM 1'(33H1*** TR FATAL ERROR: PROC, NERR= ,A4,1H,,I4)
          PRINT(1) "PROC" "NERR"
* GO TO "PROC"
$
* LABEL PREP
$
          FORM 1'(10X,47H .BOTH (APPL FORC  FSET ) AND (APPL MOTI  FSET )/
               ' 10X,46HARE SPECIFIED IN  QLIB , ONLY ONE IS PERMITTED)
          FORM 2'(10X,48HNEITHER (APPL FORC  FSET ) NOR (APPL MOTI  FSET )/
               ' 10X,20HIS PRESENT IN  QLIB )
          FORM 3'(10X,47HVIBR MODE  VSET   VCON  IS NOT PRESENT IN  VLIB )
          FORM 4'(10X,47HVIBR EVAL  VSET   VCON  IS NOT PRESENT IN  VLIB )
          FORM 5'(10X,43HTIME  NAME   FSET  IS NOT PRESENT IN  QLIB )
          FORM 6'(10X,43HCA    NAME   FSET  IS NOT PRESENT IN  QLIB )
          PRINT("NERR")
$
* GO TO FINIS
$
* LABEL FINIS
* XQT U1
* SHOW
* XQT DCU
          TOC 1: TOC "QLIB": TOC "VLIB"
* XQT EXIT
*END
```

Runstream TR RITZ

Runstream TR RITZ calculates RWL vectors following equations (2.25) through (2.29). Then the reduced system is optionally uncoupled by solving the reduced-order eigenproblem. This runstream is substantially based on one written at EISI.

```
$-----------------------------------------------------------------
$ (TR RITZ)
$-----------------------------------------------------------------
```

```
* XQT U1
$
$$$ REGISTERS: MAXHZ, AFLIB, VLIB, MNAME, NMD, SCALE
$
* REGISTER STORE (29 REGISTER HOUSE 1 1)
* REGISTER RETR  (29 REGISTER HOUSE 1 1)
  !IERR=TOC IERR(1 INV K MASK MASK)
*ONLINE=0
* IF("IERR" EQ 0): *GO TO 109
* XQT DRSI
* LABEL 109
* XQT SSOL
      RESET QLIB="AFLIB"
* XQT AUS
      OUTLIB=10: INLIB=10
      DEFINE M=1 "MNAME"
$
$ SCALE THE FIRST VECTOR
$
      DEFINE X="AFLIB" STAT DISP 1 MASK 1
      MX=PROD(M,X)
      XTMX=XTY(X,MX)
      RECI=RECI(XTMX)
      SCAL=SQRT(RECI)
      11 RITZ VECT=CBD(X,SCAL)
      DEFI RITZ=11 RITZ VECT
      12 MX=PROD(M,RITZ)
      !NSET=TOC NBLOCKS("AFLIB" STAT DISP 1 MASK)
      !N=NSET-1: !N1=1 : !N2=0
*     IF ("N" EQ 0): *GO TO 104
$
$ M-ORTHONORMALIZE VECTORS 2 THROUGH NSET
$
*     LABEL 105
            !N1=N1+1: !N2=N2+1
            DEFI U="AFLIB" STAT DISP 1 MASK "N1"
            DEFI MX=12 MX MASK MASK MASK 1 "N2"
            XTMU=XTY(MX,U)
            DEFI X=11 RITZ VECT MASK MASK 1 "N2"
            A=CBR(X,XTMU)
            U1=SUM(U,-1. A)
            MU=PROD(M,U1)
            UTMU=XTYD(U1,MU)
            RECI=RECI(UTMU)
            SCAL=SQRT(RECI)
            VECT=CBD(U1,SCAL)
            11 RITZ VECT=UNION,U(VECT)
            TEMP=PROD(M,VECT)
            12 MX =UNION,U(TEMP)
*     JGZ -1 (N 105)
* LABEL 104
* XQT DCU
      ERASE 10
! NDO=NMD-NSET: !SET=NSET+1: !SET1=NSET
* IF ("NDO" LE 0): *GO TO 1002
* XQT AUS
      TABLE(NJ=1): 13 SCALE
      J=1: "SCALE"
$
$ GENERATE REMAINING VECTORS ORTHONORMAL TO STATIC SOLUTION RITZ VECTORS
$
* LABEL 1000
*       XQT AUS
            OUTLIB=10: INLIB=10
            DEFINE M=1 "MNAME"
            DEFINE X=11 RITZ VECT MASK MASK "SET1"
```

```
            TEMP=PROD(M,X)
            NORM=NORM(TEMP)
            DEFI SCALE=13 SCALE
            APPL FORC=CBD(NORM,SCALE)
 *          IF("SET1" EQ "NSET"): *GO TO 106
            12 MX=UNION,U(TEMP)
 *          LABEL 106
 *       XQT SSOL
            RESET QLIB=10
 *       XQT AUS
            OUTLIB=10: INLIB=10
            DEFI U=STAT DISP
            DEFI MX=12 MX MASK MASK MASK 1 "SET1"
            XTMU=XTY(MX,U)
            DEFI X=11 RITZ VECT MASK MASK 1 "SET1"
            A=CBR(X,XTMU)
            U1=SUM(U,-1. A)
            DEFI M=1 "MNAME"
            MU=PROD(M,U1)
            UTMU=XTYD(U1,MU)
            RECI=RECI(UTMU)
            SCAL=SQRT(RECI)
            VECT=CBD(U1,SCAL)
            11 RITZ VECT=UNION,U(VECT)
            !SET1=SET: !SET=SET+1
* XQT DCU
      ERASE 10
* JGZ,-1(NDO,1000)
* LABEL 1002
*IF("DRFORMAT" EQ DIAG): *GOTO 10020
* XQT AUS
      DEFI X=11 RITZ VECT
      "VLIB" VIBR MODE 1 1=UNION(X)
 TABLE(NI=1,NJ="NMD") : VIBR EVAL
*RETURN
*LABEL 10020
* XQT AUS
      OUTLIB=10: INLIB=10
      DEFINE K=1 K: DEFI M=1 "MNAME"
      DEFINE X=11 RITZ VECT
      IJCODE=10000
      !NMODE=NMD
      KX=PROD(K,X): SYN K 10000 "NMODE" = XTYS(X,KX)
      MX=PROD(M,X): SYN M 10000 "NMODE" = XTYS(X,MX)
! ZERO=NMODE-1
* JZ (ZERO,1003)
*XQT DCU
 TOC 10
* XQT STRP
      RESET SOURCE=10, DEST=10, FRQ2="MAXHZ"
* JGZ (ZERO,1004)
* LABEL 1003
* XQT AUS
      OUTLIB=10: INLIB=10
      !K=DS 2 1 1(10 SYN K MASK MASK)
      !M=DS 2 1 1(10 SYN M MASK MASK)
      !EVAL=K/M
      TABLE(NI=1,NJ=1): SYS EVEC: J=1: 1.0
      TABLE(NI=1,NJ=1): SYS EVAL: J=1: "EVAL"
* LABEL 1004
* XQT AUS
      OUTLIB=10: INLIB=10
      DEFINE E=SYS EVEC
      DEFI X=11 RITZ VECT
      X ORTH 1 1=CBR(X,E)
      DEFINE X=X ORTH 1 1
```

```
      "VLIB" VIBR MODE 1 1=UNION(X)
      DEFINE E=SYS EVAL
      "VLIB" VIBR EVAL 1 1=UNION(E)
      "VLIB" VIBR HZ   1 1=SQRT(.0253303 E)
*ONLINE=1
* XQT DCU
      PRINT "VLIB" VIBR HZ 1 1
* XQT U1
* REGISTER RETR (29 REGISTER HOUSE 1 1)
*END
```

Runstream TR REDM

TR REDM is a utility runstream for generating the reduced equations given a set of basis vectors. Depending on the input register DRFORMAT the equations can be coupled or uncoupled. If the equations are uncoupled, it is assumed that $\mathbf{\Phi}^T\mathbf{M}\mathbf{\Phi}$ is the identity matrix and $\mathbf{\Phi}^T\mathbf{K}\mathbf{\Phi}$ is a diagonal matrix with the eigenvalues along the diagonal.

```
$-------------------------------------------------------------
$ (TR REDM) - FORM REDUCED K AND M MATRICES FOR TRANSIENT RESP.
$-------------------------------------------------------------
$
$ REGISTERS:
$           DRFO = 'FULL, 'RITZ, OR 'DIAG
$           NMODE = NUMBER OF MODES
$           MNAME = MASS MATRIX NAME
$           VLIB  = LIBRARY FOR VIBRATIONAL MODES AND FREQS
$           QLIB  = DESTINATION LIBRARY FOR MATRICES
$
*XQTC AUS
 OUTLIB="QLIB"
 DEFINE X = "VLIB" VIBR MODE 1 1 1,"NMODES"
 DEFINE E = "VLIB" VIBR EVAL 1 1
 DEFINE DAMP = 1 DAMP SPAR
 DEFINE DMPD = 1 DMPD
! IDSP = TOC,IERR(1 DAMP SPAR MASK MASK)
! IDMD = TOC,IERR(1 DMPD MASK MASK MASK)
! DRFO
*IF("DRFO" NE FULL): *GOTO 100
$
$ FULL MATRICES, X IS A SET OF EIGENVECTORS
$
! N = NMODES
! I = 1
 TABLE(NI="NMODES",NJ="NMODES") : XTMX
*LABEL 10
 I="I" : J="I" : 1.0
! I = I + 1
*JGZ,-1(N,10)
! N = NMODES
! I = 1
 TABLE(NI="NMODES",NJ="NMODES") : XTKX
*LABEL 20
! K = DS,"I",1,1("VLIB" VIBR EVAL 1 1)
 I="I" : J="I" : "K"
! I = I + 1
*JGZ,-1(N,20)
*IF("IDSP" NE 0): *GOTO 85
 OUTL=22 : INLI=22
 DX = PROD(DAMP,X)
 "QLIB" XTDX = XTY(X,DX)
*LABEL 85
*IF("IDMD" NE 0): *RETURN
! N = NMODES : ! I = 1
 *IF("IDSP" NE 0): TABLE(NI="NMODES",NJ="NMODES") : "QLIB" XTDX
```

```
*IF("IDSP" EQ 0): TABLE,U : "QLIB" XTDX
*LABEL 90
! D = DS,"I",1,1("VLIB" DMPD MASK MASK MASK)
 I="I" : J="I" : "D"
! I = I + 1
*JGZ,-1(N,90)
*RETURN
*LABEL 100
$
$ FULL REDUCED MATRICES (X NOT EIGENVECTORS)
$
*IF("DRFO" NE RITZ): *GOTO 200
 OUTLIB=22 : INLIB=22
 DEFINE K = 1 K SPAR
 DEFINE M = 1 "MNAME"
 KX = PROD(K,X)
 MX = PROD(M,X)
 "QLIB" XTKX = XTY(X,KX)
 "QLIB" XTMX = XTY(X,MX)
*IF("IDSP" NE 0): *GOTO 130
 OUTL=22 : INLI=22
 DX = PROD(DAMP,X)
 "QLIB" XTDX = XTY(X,DX)
*LABEL 130
*IF("IDMD" NE 0): *RETURN
! N = NMODES : ! I = 1
*IF("IDSP" NE 0): TABLE(NI="NMODES",NJ="NMODES") : "QLIB" XTDX
*IF("IDSP" EQ 0): TABLE,U : "QLIB" XTDX
*LABEL 140
! D = DS,"I",1,1("VLIB" DMPD MASK MASK MASK)
 I="I" : J="I" : "D"
! I = I + 1
*JGZ,-1(N,140)
*RETURN
*LABEL 200
$
$ SIMPLE DIAGONAL CASE (X EIGENVECTORS)
$
 TABLE(NI=1,NJ="NMODES") : XTKX : TRAN(SOUR=E)
 TABLE(NI=1,NJ="NMODES") : XTMX : J=1,"NMODES" : 1.0
*IF("IDSP" NE 0): *GOTO 210
 OUTL=22 : INLI=22
 DX = PROD(DAMP,X)
 "QLIB" XTDX = XTYD(X,DX)
*LABEL 210
*IF("IDMD" NE 0): *RETURN
! N = NMODES : ! I = 1
*IF("IDSP" NE 0): TABLE(NI=1,NJ="NMODES") : "QLIB" XTDX
*IF("IDSP" EQ 0): TABLE,U : "QLIB" XTDX
*LABEL 220
! D = DS,"I",1,1("VLIB" DMPD MASK MASK MASK)
 I=1 : J="I" : "D"
! I = I + 1
*JGZ,-1(N,220)
*RETURN
*END
```

Runstream TR PLOT

TR PLOT is a utility runstream for producing plots of response quantities as a function of time. Its use is demonstrated in the stepped beam example runstream.

```
$-------------------------------------------------------------
$ (TR PLOT)
$-------------------------------------------------------------
*XQT U1
```

```
$
$ PLOTS TRANSIENT, TIME HISTORY DATA PRODUCED BY DR/TR1
$
*REGISTER EXCEPTIONS TLIB
*REGISTER STORE ("TLIB" TR REG 1 1)
*REGISTER RETRIEVE("TLIB" TR REG 1 1)
$                  DEFAULT REGISTER ASSIGNMENTS
! INLIB=1
! IDJK = 'NONE
! IDQ = 'NONE
! YNAME = 'DISP
! N3 = 1
! TITLE = 'TITLE
! ID = 1
! OPT=0
*DATA,OPT(TRPLOT OPTIONS)
$
! NS1 = TOC,NI("INLIB" HIST "YNAME"" N3" MASK)
! NW1 = TOC,NWDS("INLIB" HIST "YNAME" "N3" MASK)
! NBLK = TOC,NBLOCKS("INLIB" HIST "YNAME" "N3" MASK)
! NJBL = TOC,NJ("INLIB" HIST "YNAME" "N3" MASK)
! NS1 : ! NW1 : ! NBLK : ! NJBL
! NSTE=NW1/NS1
! NPPT=NSTE
! DT=DS,1,1,1("INLIB" DT MASK MASK MASK) $ TIME STEP
! TIDQ = TOC,IERR("INLIB" SEL "IDQ" MASK MASK)
! TIDJ = TOC,IERR("INLIB" SEL "IDJK" MASK MASK)
! TTIT = TOC,IERR("INLIB" "TITLE" MASK MASK MASK)
*ONLINE=0
*XQT AUS
 TABLE(NI=1,NJ="NPPT") : "TLIB" XTAB
  DDATA="DT"
  J=1,"NPPT" : 0.0
$
$ LOOP OVER ALL RESPONSE QUANTITIES
$
! NJLS = 1-NBLK*NJBL + NSTE $ NJ OF LAST BLOCK
! KBLK = NBLK - 1
! DBLS = KBLK*NJBL
! JBLK = KBLK
! NSM1 = NS1 - 1
! SBASE = 0
! NJLS
! I1=1 : ! N1=NS1
*LABEL                                                  L1
 DEFINE Y = "INLIB" HIST "YNAME" "N3" 1 1,"JBLK"
 TABLE(NJ="NSTE") : "TLIB" YTAB AUS "I1"
*JZ(KBLK,L2)
 TRAN(SOUR=Y,SBAS="SBAS",SSKIP="NSM1",ILIM=1,JLIM="NJBL")
*LABEL L2
 DEFINE Y = "INLIB" HIST "YNAME" "N3" 1 "NBLK","NBLK"
 TABLE,U          : "TLIB" YTAB AUS "I1"
 TRAN(SOUR=Y,SBAS="SBAS",SSKIP="NSM1",ILIM=1,JLIM="NJLS",DBAS="DBLS")
! SBASE = SBASE + 1
! I1=I1+1
*JGZ,-1(N1,L1)
$
$ GENERATE AN X,Y PLOT FOR EACH RESPONSE QUANTITY
$
*ONLINE=1
*XQT PXY
 RESET DEVICE=META
 RESET NDEV=4014
 FONT XNUM=1 : FONT YNUM=1
 FONT XLAB=1 : FONT YLAB=1
 FONT TEXT=1
```

```
  X = "TLIB" XTAB
  XLABEL' TIME (SECONDS)
 *JNZ(TIDJ,110)
  YLFORMAT,72>
  '(4H J =,I2,9H JOINT = ,I5,8H COMP = ,I2,8H HIST = ,A4,6H ID = ,I6)
 *LABEL 110
 *JNZ(TIDQ,120)
  YLFORMAT,72>
 '(4H J= ,I2,1X,A4,6H GRP= ,I2,6H IND= ,I5,7H COMP= ,A4,5H ID= ,I6)
 *LABEL 120
  XAXIS=3,5,10
  YAXIS=4,5,10
  TPOS=0,0
 ! I1=1 : ! N1=NS1
 *JNZ(TTIT,L3)
  TEXT = "TITLE"
 *LABEL                                          L3
  ADVANCE
  BOUNDARIES = .01 .99 .04 .1
 *JNZ(TTIT,L4)
  PLOT TEXT
 *LABEL                                          L4
  BOUNDARIES=.01 .99 .15 .85
  Y = "TLIB" YTAB AUS "I1"
 *JNZ(TIDJ,210)
 ! JOINT = DS,1,"I1",1("INLIB" SEL "IDJK" MASK MASK)
 ! COMP  = DS,2,"I1",1("INLIB" SEL "IDJK" MASK MASK)
  YLABEL "I1"  "JOINT" "COMP" "YNAME" "ID"
 *LABEL 210
 *JNZ(TIDQ,220)
 ! ENAME = DS,1,"I1",1("INLIB" SEL "IDQ" MASK MASK)
 ! EGRP = DS,2,"I1",1("INLIB" SEL "IDQ" MASK MASK)
 ! EINDX = DS,3,"I1",1("INLIB" SEL "IDQ" MASK MASK)
 ! ECOMP = DS,4,"I1",1("INLIB" SEL "IDQ" MASK MASK)
  YLABEL "I1"  "ENAME" "EGRP" "EINDX" "ECOMP" "ID"
 *LABEL 220
  INIT
  PLOT CURV
 ! I1=I1+1
 *JGZ,-1(N1,L3)
 *XQT U1
 *REGISTER RETRIEVE ("TLIB" TR REG 1 1)
 *FREE "TLIB"
 *RETURN
 *END
```

Runstream TR VECTORS

Runstream **TR VECTORS** generates basis vectors by calling the system eigensolver, calling runstream **TR RITZ**, using the static mode method, or by other experimental techniques.

```
$-------------------------------------------------------------
$ (TR,VECTORS) - COMPUTE VECTORS FOR USE IN DYNAMIC ANALYSIS
$-------------------------------------------------------------
*XQT AUS
 R = RIGID(1)
 CR = PROD("MNAME",R)
 Z = NDDF,1(CR)
! NDDF = DS,1,1,1(1 Z AUS 1 1)
! NDDF
*IF("METHOD" NE MODE): *GOTO 100
*XQT E4
 RESET NMODES="NMODES"
 RESET M="MNAME"
 RESET NDDF="NDDF"
 RESET CONV="CONV"
```

```
*DCALL,OPT(E4 PARAMETERS)
*GOTO 200
$
*LABEL 100
*IF("METHOD" NE RITZ): *GOTO 105
! MAXHZ = 1.03E+10
! NMD = NMODES
! SCALE = 1.0
! AFLIB = 1
! VLIB  = 1
*DCALL(TR RITZ )
*GOTO 200
$
*LABEL 105
*IF("METHOD" NE OLD): *GOTO 110
*XQT DCU
 COPY 3 1 VIBR MODE
 COPY 3 1 VIBR EVAL
*GOTO 200
*LABEL 110
*IF("METHOD" NE ONES): *GOTO 115
*DCALL(TEST NEB3)
*GOTO 200
*LABEL 115
*IF("METHOD" NE STAT): *GOTO 120
*XQT E4
 RESET NMODES="NMODES"
 RESET M="MNAME"
 RESET NDDF="NDDF"
 RESET CONV="CONV"
*DCALL,OPT(E4 PARAMETERS)
*XQT DRSI
*XQT SSOL
$ ORTHOGONALIZE STATIC SOLUTION AND APPEND TO SET OF MODE SHAPES
*DCALL(TR,GRAM)
$ MAKE THE VECTORS ORTHOGONAL WITH RESPECT TO BOTH K AND M
*DCALL(TR DIAG)
*XQT VPRT
 PRINT S1 AUS
*XQT DCU
 TOC 1
 PRINT 1 VIBR EVAL
*LABEL 120
*IF("METHOD" NE UMOT): *GO TO 200
*XQT AUS
 Z = NDDF,1,2(CR)
*XQT DRSI
 RESET CON=2
*XQT AUS
 UDF = 1
 SSPREP(K,2)
*XQT DCU
 CHANGE 1 BNF MASK MASK MASK VIBR MODE 1 1
$ UNCOUPLE THE SYSTEM
*DCALL(TR,DIAG)
$ END OF METHODS OPTIONS
*LABEL 200
*END
```

Runstream **TR GRAM**

TR GRAM is a utility runstream for performing a Gram-Schmidt orthogonalization of a set of vectors.

```
$-------------------------------------------------------------
$ (TR,GRAM) - PERFORM GRAM-SCHMIDT PROCESS TO M-ORTHOGONALIZE
```

```
$             STAT DISP WITH RESPECT TO VIBR MODE AND THEN
$             REPLACE THE LAST VIBR MODE WITH THE NEW VECTOR
$------------------------------------------------------------
*XQT AUS
! NMM1 = NMODES - 1
 DEFINE X = VIBR MODE 1 1 1,"NMM1"
 DEFINE S = STAT DISP 1 1
 DEFINE M = "MNAME"
 INLIB=10 : OUTLIB=10
$ ORTHOGONALIZE THE STATIC SOLUTION WITH RESPECT TO THE MODE SHAPES
 MX = PROD(M,X)
 XTMS = XTY(MX,S)
 A = CBR(X,XTMS)
 S1 = SUM(S, -1.0 A)
$ NOW SCALE THE VECTOR
 MS = PROD(M,S1)
 STMS = XTY(S1,MS)
! STMS = DS,1,1,1(10 STMS AUS 1 1) : ! STMS = STMS**.5 : ! STMS = 1.0/STMS
 TEMP MODE 1 1 = UNION("STMS" S1,X)
*XQT DCU
 CHANGE 10 TEMP MODE 1 1 VIBR MODE 1 1
 COPY 10 1 VIBR MODE 1 1
*DELETE 10
*END
```

Runstream SENS DVUP

SENS DVUP is a utility runstream for updating the design variable registers based on the data sets X ADS and XNAME ADS. It is always called immediately before calling MODEL so the current values of the design variables are available for use.

```
$------------------------------------------------------------
$ (SENS DVUP) - UPDATE DESIGN VARIABLE REGISTERS FROM DATASET
$------------------------------------------------------------
*XQT U1
! N = TOC,NJ(1 XNAME ADS 1 1)
! I = 1
*LABEL 10
! RNAME = DS,1,"I",1(1 XNAME ADS 1 1)
! RVAL  = DS,1,"I",1(1 X     ADS 1 1)
! "RNAME" = "RVAL"
! "RNAME"
! I = I + 1
*JGZ,-1(N,10)
*END
```

Runstream TR DXDV 1

The TR DXDV n runstreams implement the different sensitivity methods. The structure of all these runstreams is similar. In each case there is a loop over the designated design variables, and sensitivities are calculated of the required response quantities at the set of critical points. Within this loop there is at least one call to runstream MODEL to form a perturbed design, a call to form a set of new reduced equations, and a call to processor DRX to integrate the reduced equations in time. Runstream TR DXDV 1 implements the forward difference method with either fixed or updated basis vectors.

```
$------------------------------------------------------------
$ (TR DXDV 1) - CALCULATES DERIVATIVES OF TRANSIENT RESPONSE
$               USING THE FORWARD DIFFERENCE OPERATOR AND
$               EITHER FIXED OR UPDATED MODES
$ UPDATE HISTORY
$ 6/28/88 WHG - MODIFIED FOR VELO, ACCE, STRESSES
$------------------------------------------------------------
*XQT U1
*REGISTER STORE(1 DXDV REGISTERS 1 1)
```

```
*REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
*RGI
     FDCH = .001
     FDMCH = .0001
     XLIB = 5
     OPT  = 0
     RLIB = 14
     DRMETHOD=0
     DXMD=UPDATED
*DCALL,OPT(DXDV PARAMETERS)
*SHOW
*DCALL(TR,DPREP)
$
$ LOOP OVER ALL DESIGN VARIABLES
$
! NDV = TOC,NJ(1 X ADS 1 1)
! NCNT = NDV
! IDV = 1
*LABEL 10
*XQT U1
! IFLG = DS,1,"IDV",1(1 XFLG ADS 1 1)
*JZ(IFLG,100)
*LIBS "XLIB" 2 3 4 1 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT DCU
 COPY "XLIB" 1 XNAME ADS 1 1
*XQT U3
 RP2
 FORMAT 1'(1H1,20X,41HBEGINNING SENSITIVITY CALCULATION. DV NO.,I3)
 PRINT(1) "IDV"
*XQT AUS
 DEFINE X = "XLIB" X ADS 1 1
 TABLE(NI=1,NJ="NDV") : X ADS 1 1
 J=1,"NDV" : 1.0
 J="IDV" : "FDCH"
 TRAN(SOURCE=X, OPERATION=MULT)
$ CHECK FOR TOO SMALL A STEP
! X = DS,1,"IDV",1("XLIB" X ADS 1 1)
! DX = FDCH*X
*IF("DX" GT "FDCHM"): *GOTO 20
! DX = FDMCH
! X = X + FDMCH
 TABLE,U : X ADS 1 1
 OPER = XSUM
 J="IDV" : "X"
*LABEL 20
*CALL(SENS,DVUP)
$ FORM PERTURBED MODEL
*ONLINE=0
*DCALL(MODEL)
*ONLINE=1
*XQT DCU
 COPY "XLIB" 1 TIME
 COPY "XLIB" 1 CA
 COPY "XLIB" 1 DMPD
*IF("DXMD" NE FIXE): *GOTO 30
 COPY "XLIB" 1 VIBR MODE
*DCALL(TR,DIAG)
*GO TO 40
*LABEL 30
*IF("DXMD" NE UPDA): *GOTO 40
*DCALL(TR,VECTORS)
*LABEL 40
*XQT AUS
 DEFINE X = VIBR MODE 1 1 1,"NMODES"
 DEFINE F = APPL FORC 1
 XTF = XTY(X,F)
```

```
 *DCALL(TR,REDM) VLIB=1
 *XQT DRX
  DTEX(DT="DT",METHOD="DRMETHOD",NTERMS="NTERMS")
  TR1(QXLIB=1,QX1L=1,QX2L=1,T2="T2",LB="BLKSIZE")
 *DCALL(TR,DBACK 1)
 $
 $ COMPUTE DERIVATIVES USING FORWARD DIFFERENCE OPERATOR
 $
 ! OVDX = 1.0/DX
 ! MOVD = - OVDX
 ! I = 1 : ! N = NBCK
 *LIBS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
 *LABEL 80
 *XQT AUS
 ! NM = DS,"I",1,1(1 BACK LIST 1 1)
 ! IERR = TOC,IERR(1 SEL "NM" MASK MASK)
 *JNZ(IERR,90)
  DEFINE CP1 = "XLIB" CRPT "NM"
  DEFINE CP0 =  CRPT "NM"
  DXDV "NM" "IDV" = SUM("OVDX" CP1 "MOVD" CP0)
 *XQT DCU
  PRINT 1 DXDV "NM" "IDV"
 *LABEL 90
 ! I = I + 1
 *JGZ,-1(N,80)
 ERASE "XLIB"
 *LABEL 100
 ! IDV = IDV + 1
 *JGZ,-1(NCNT,10)
 $
 *XQT U1
 *REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
 *RETURN
 *END
```

Runstream TR DXDV 3

Runstream TR DXDV 3 implements the fixed-mode semianalytical sensitivity method. Depending on the call to runstream TR DBACK, either the mode displacement or mode acceleration method is used to recover the physical sensitivities.

```
$-------------------------------------------------------------
$ (TR DXDV 3) - CALCULATES DERIVATIVES OF TRANSIENT RESPONSE
$               SEMIANALYTICALLY
$-------------------------------------------------------------
*XQT U1
*REGISTER STORE(1 DXDV REGISTERS 1 1)
*REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
*RGI
     FDCH = .001
     FDMCH = .0001
     XLIB = 5
     OPT  = 0
     RLIB = 14
     DRMETHOD=0
*DCALL,OPT(DXDV PARAMETERS)
*SHOW
$
$ INITIALIZATION FOR DERIVATIVE CALCULATIONS
$
*DCALL(TR,DPREP)
*XQT U1
$
$ LOOP OVER ALL DESIGN VARIABLES
$
! NDV = TOC,NJ(1 X ADS 1 1)
```

```
! NCNT = NDV
! IDV = 1
*LABEL 10
*LIBS "XLIB" 2 3 4 1 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT U1
! IFLG = DS,1,"IDV",1("XLIB" XFLG ADS 1 1)
*JZ(IFLG,100)
*XQT DCU
 COPY "XLIB" 1 XNAME ADS 1 1
*XQT U3
 RP2
 FORMAT 1'(1H1,20X,41HBEGINNING SENSITIVITY CALCULATION. DV NO.,I3)
 PRINT(1) "IDV"
*XQT AUS
 DEFINE X = "XLIB" X ADS 1 1
 TABLE(NI=1,NJ="NDV") : X ADS 1 1
 J=1,"NDV" : 1.0
 J="IDV" : "FDCH"
 TRAN(SOURCE=X, OPERATION=MULT)
$ CHECK FOR TOO SMALL A STEP
! X = DS,1,"IDV",1("XLIB" X ADS 1 1)
! DX = FDCH*X
*IF("DX" GT "FDCHM"): *GOTO 20
! DX = FDMCH
! X = X + FDMCH
 TABLE,U : X ADS 1 1
 OPER = XSUM
 J="IDV" : "X"
*LABEL 20
*CALL(SENS,DVUP)
$ FORM PERTURBED MODEL
*ONLINE=0
*DCALL(MODEL)
*ONLINE=1
*XQT AUS
 DEFINE X = "XLIB" VIBR MODE 1 1 1,"NMODES"
 DEFINE K0 = "XLIB" K SPAR
 DEFINE K1 =        K SPAR
 DEFINE M0 = "XLIB" "MNAME"
 DEFINE M1 =        "MNAME"
 DEFINE D0 = "XLIB" DAMP SPAR
 DEFINE D1 =        DAMP SPAR
 DEFINE F0 = "XLIB" APPL FORC 1
 DEFINE F1 =        APPL FORC 1
! IDSP = TOC,IERR(1 DAMP SPAR MASK MASK)
! OVDX = 1.0/DX
! MOVD = -OVDX
 DKDV = SUM("OVDX" K1 "MOVD" K0)
 DMDV = SUM("OVDX" M1 "MOVD" M0)
 DFDV = SUM("OVDX" F1 "MOVD" F0)
*IF("IDSP" EQ 0): DDDV = SUM("OVDX" D1 "MOVD" D0)
 DKX = PROD(DKDV,X)
 DMX = PROD(DMDV,X)
*IF("IDSP" EQ 0): DDX = PROD(DDDV,X)
 RDKX = XTY(X,DKX)
 RDMX = XTY(X,DMX)
*IF("IDSP" EQ 0): RDDX = XTY(X,DDX)
 XTF AUS = XTY(X,DFDV)
*XQT DCU
 COPY "XLIB" 1 TIME
 COPY "XLIB" 1 CA
 COPY "XLIB" 1 DT AUS
 COPY "XLIB" 1 DTEX AUS
 COPY "XLIB" 1 DCON AUS $ CONSTANTS FOR NEWMARK METHOD
$
$ FORM THE RIGHT-HAND-SIDE PSEUDO LOAD VECTOR
```

```
$
*XQT DRX
          BACK(LRZ="BLKSIZE",PRINT=0)
 T = -1.0 RDMX : Y = "XLIB" QX2 AUS
*IF("IDSP" EQ 0): T = -1.0 RDDX : Y = "XLIB" QX1 AUS
 T = -1.0 RDKX : Y = "XLIB" QX  AUS
 Z = FH AUS
*XQT DRX
 TR1(QXLIB=1,QX1LIB=1,QX2LIB=1,T2="T2",FHLIB=1,LB="BLKSIZE")
$
$ BACK TRANSFORM FOR NECESSARY SENSITIVITIES OF PHYSICAL
$ QUANTITIES
$
*DCALL(TR,DBACK,4)
*XQTC DCU
 ERASE 1
*LABEL 100
! IDV = IDV + 1
*JGZ,-1(NCNT,10)
$
*LIBS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT U1
*REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
*RETURN
*END
```

Runstream TR DXDV 5

Runstream TR DXDV 5 implements the overall central difference method using either fixed or updated basis vectors.

```
$-----------------------------------------------------------------
$ (TR DXDV 5) - CALCULATES DERIVATIVES OF TRANSIENT RESPONSE
$               USING TWO POINT CENTRAL DIFFERENCE OPERATOR
$               WITH UPDATED OR FIXED MODES
$-----------------------------------------------------------------
*XQT U1
*REGISTER STORE(1 DXDV REGISTERS 1 1)
*REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
*RGI
     FDCH = .001
     FDMCH = .0001
     XLIB = 5
     YLIB = 6
     OPT  = 0
     RLIB = 14
     DRMETHOD=0
     DXMD=UPDATED
*DCALL,OPT(DXDV PARAMETERS)
*SHOW
$
$ INITIALIZATION FOR DERIVATIVE CALCULATIONS
$
*DCALL(TR,DPREP)
$
$ LOOP OVER ALL DESIGN VARIABLES
$
! NDV = TOC,NJ(1 X ADS 1 1)
! NCNT = NDV
! IDV = 1
*LABEL 10
*XQT U1
! IFLG = DS,1,"IDV",1(1 XFLG ADS 1 1)
*JZ(IFLG,100)
*XQT U3
 RP2
```

```
 FORMAT 1'(1H1,20X,41HBEGINNING SENSITIVITY CALCULATION. DV NO.,I3)
 PRINT(1) "IDV"
*LIBS "XLIB" 2 3 4 1 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT AUS
! IS = 1
! NST = 2
! SIGN = 1.0
$
$ DO ANALYSIS FOR BOTH POSITIVE AND NEGATIVE STEPS
$
*LABEL 15
*XQT DCU
 COPY "XLIB" 1 XNAME ADS 1 1
*XQT AUS
 DEFINE X = "XLIB" X ADS 1 1
 TABLE(NI=1,NJ="NDV") : X ADS 1 1
 TRAN(SOURCE=X)
$ DIFFERENCE APPROPRIATE DESIGN VARIABLE
! X = DS,1,"IDV",1("XLIB" X ADS 1 1)
! DX = FDCH*X
*IF("DX" GT "FDCHM"): *GOTO 20
! DX = FDMCH
*LABEL 20
! X = DX*SIGN + X
 TABLE,U : X ADS 1 1
 OPER = XSUM
 J="IDV" : "X"
*CALL(SENS,DVUP)
$ FORM PERTURBED MODEL
*ONLINE=0
*DCALL(MODEL)
*ONLINE=1
*XQT DCU
 COPY "XLIB" 1 TIME
 COPY "XLIB" 1 CA
 COPY "XLIB" 1 DMPD
*IF("DXMD" NE FIXE): *GOTO 30
 COPY "XLIB" 1 VIBR MODE
*DCALL(TR,DIAG)
*GO TO 40
*LABEL 30
*IF("DXMD" NE UPDA): *GOTO 40
*DCALL(TR,VECTORS)
*LABEL 40
*XQT AUS
 DEFINE X = VIBR MODE 1 1 1,"NMODES"
 DEFINE F = APPL FORC 1
 XTF = XTY(X,F)
*DCALL(TR,REDM) VLIB=1
*XQT DRX
 DTEX(DT="DT",METHOD="DRMETHOD",NTERMS="NTERMS")
 TR1(QXLIB=1,T2="T2",QX1LIB=1,QX2LIB=1,LB="BLKSIZE")
*DCALL(TR,DBACK 1)
! SIGN = -1.0
*LIBS "YLIB" 2 3 4 1 5 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*JGZ,-1(NST,15)
$
$ COMPUTE DERIVATIVES USING CENTRAL DIFFERENCE OPERATOR
$
! TWDX = 2.0*DX
! OVDX = 1.0/TWDX
! MOVD = - OVDX
! I = 1 : ! N = NBCK
*LIBS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*LABEL 80
*XQT AUS
```

```
 ! NM = DS,"I",1,1(1 BACK LIST 1 1)
 ! IERR = TOC,IERR(1 SEL "NM" MASK MASK)
 *JNZ(IERR,90)
  DEFINE CP1 = "XLIB" CRPT "NM"
  DEFINE CP0 = "YLIB" CRPT "NM"
  DXDV "NM" "IDV" = SUM("OVDX" CP1 "MOVD" CP0)
 *XQT DCU
  PRINT 1 DXDV "NM" "IDV"
 *LABEL 90
 ! I = I + 1
 *JGZ,-1(N,80)
 ERASE "XLIB"
 ERASE "YLIB"
 *LABEL 100
 ! IDV = IDV + 1
 *JGZ,-1(NCNT,10)
 $
 *XQT U1
 *REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
 *RETURN
 *END
```

Runstream TR DXDV 6

Runstream TR DXDV 6 implements the semianalytical method with nonzero $d\Phi/dx$. The called procedure TR DPHI determines how the basis vector derivatives are calculated.

```
$-----------------------------------------------------------------
$ (TR DXDV 6) - CALCULATES DERIVATIVES OF TRANSIENT RESPONSE
$               SEMIANALYTICALLY BUT WITH THE EFFECT OF CHANGING
$               MODES INCLUDED
$-----------------------------------------------------------------
*XQT U1
*REGISTER STORE(1 DXDV REGISTERS 1 1)
*REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
*RGI
     FDCH = .001
     FDMCH = .0001
     XLIB = 5
     OPT  = 0
     RLIB = 14
     DRMETHOD=0
*DCALL,OPT(DXDV PARAMETERS)
*SHOW
$
$ INITIALIZATION FOR DERIVATIVE CALCULATIONS
$
*DCALL(TR,DPREP)
*XQT U1
$
$ LOOP OVER ALL DESIGN VARIABLES
$
! NDV = TOC,NJ(1 X ADS 1 1)
! NCNT = NDV
! IDV = 1
*LABEL 10
*LIBS "XLIB" 2 3 4 1 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT U1
! IFLG = DS,1,"IDV",1("XLIB" XFLG ADS 1 1)
*JZ(IFLG,100)
*XQT DCU
 COPY "XLIB" 1 XNAME ADS 1 1
*XQT U3
 RP2
 FORMAT 1'(1H1,20X,41HBEGINNING SENSITIVITY CALCULATION. DV NO.,I3)
 PRINT(1) "IDV"
```

```
*XQT AUS
 DEFINE X = "XLIB" X ADS 1 1
 TABLE(NI=1,NJ="NDV") : X ADS 1 1
 J=1,"NDV" : 1.0
 J="IDV" : "FDCH"
 TRAN(SOURCE=X, OPERATION=MULT)
$ CHECK FOR TOO SMALL A STEP
! X = DS,1,"IDV",1("XLIB" X ADS 1 1)
! DX = FDCH*X
*IF("DX" GT "FDCHM"): *GOTO 20
! DX = FDMCH
! X = X + FDMCH
 TABLE,U : X ADS 1 1
 OPER = XSUM
 J="IDV" : "X"
*LABEL 20
*CALL(SENS,DVUP)
! OVDX = 1.0/DX
! MOVD = -OVDX
$ FORM PERTURBED MODEL
*ONLINE=0
*DCALL(MODEL)
$ CALCULATE DERIVATIVES OF MODES SHAPES
*XQT AUS
 DEFINE K0 = "XLIB" K SPAR
 DEFINE K1 =        K SPAR
 DEFINE M0 = "XLIB" "MNAME"
 DEFINE M1 =        "MNAME"
 DKDV = SUM("OVDX" K1 "MOVD" K0)
 DMDV = SUM("OVDX" M1 "MOVD" M0)
*DCALL(TR,DPHI,3)
*XQT DCU
 COPY "XLIB" 1 DT AUS
 COPY "XLIB" 1 DTEX AUS
 COPY "XLIB" 1 DCON AUS $ CONSTANTS FOR NEWMARK METHOD
*XQT AUS
 DEFINE X0 = "XLIB" VIBR MODE 1 1 1,"NMODES"
 DEFINE K0 = "XLIB" K SPAR
 DEFINE K1 =        K SPAR
 DEFINE M0 = "XLIB" "MNAME"
 DEFINE M1 =        "MNAME"
 DEFINE D0 = "XLIB" DAMP SPAR
 DEFINE D1 =        DAMP SPAR
 DEFINE F0 = "XLIB" APPL FORC 1 1
 DEFINE F1 =        APPL FORC 1 1
 DEFINE DXDV = DXDV AUS "IDV"
 $
 $ CALCULATE DERIVATIVE TERMS INVOLVING THE STIFFNESS MATRIX
 $
  DKX1 = PROD(DKDV,X0)
  DKX2 = PROD(K0,X0)
  DKX3 = PROD(K0,DXDV)
  XDK1 = XTY(X0,DKX1)
  XDK2 = XTY(DXDV,DKX2)
  XDK3 = XTY(X0,DKX3)
  XTMP = SUM(XDK1,XDK2)
  XDKX = SUM(XTMP,XDK3)
 $
 $ CALCULATE DERIVATIVE TERMS INVOLVING THE MASS MATRIX
 $
  DMX1 = PROD(DMDV,X0)
  DMX2 = PROD(M0,X0)
  DMX3 = PROD(M0,DXDV)
  XDM1 = XTY(X0,DMX1)
  XDM2 = XTY(DXDV,DMX2)
  XDM3 = XTY(X0,DMX3)
```

```
 XTMP = SUM(XDM1,XDM2)
 XDMX = SUM(XTMP,XDM3)
$
$ CALCULATE DERIVATIVE TERMS INVOLVING THE DAMPING MATRIX
$
! IDSP = TOC,IERR(1 DAMP SPAR MASK MASK)
*IF("IDSP" NE 0): *GOTO 30
 DDDV = SUM("OVDX" D1 "MOVD" D0)
 DDX1 = PROD(DDDV,X0)
 DDX2 = PROD(D0,X0)
 DDX3 = PROD(D0,DXDV)
 XDD1 = XTY(X0,DDX1)
 XDD2 = XTY(DXDV,DDX2)
 XDD3 = XTY(X0,DDX3)
 XTMP = SUM(XDD1,XDD2)
 XDDX = SUM(XTMP,XDD3)
*LABEL 30
$
$ CALCULATE DERIVATIVE TERMS INVOLVING THE FORCE VECTOR
$
 DFDV = SUM("OVDX" F1 "MOVD" F0)
 XF1 = XTY(DXDV, F0)
 XF2 = XTY(X0, DFDV)
 XTF AUS = SUM(XF1, XF2)
*XQT DCU
 PRINT 1 XTF AUS
 COPY "XLIB" 1 TIME AUS
 COPY "XLIB" 1 CA AUS
$
$ FORM THE RIGHT-HAND-SIDE PSEUDO LOAD VECTOR
$
*XQT DRX
          BACK(LRZ="BLKSIZE",PRINT=0)
 T = -1.0 XDMX : Y = "XLIB" QX2 AUS
*IF("IDSP" EQ 0): T = -1.0 XDDX : Y = "XLIB" QX1 AUS
 T = -1.0 XDKX : Y = "XLIB" QX  AUS
 Z = FH AUS
*XQT DRX
 TR1(QXLIB=1,QX1LIB=1,QX2LIB=1,T2="T2",FHLIB=1,LB="BLKSIZE")
*XQT DCU
 TOC 1
$
$ BACK TRANSFORM FOR NECESSARY SENSITIVITIES OF PHYSICAL
$ QUANTITIES
$
*DCALL(TR,DBACK,3)
*XQTC DCU
 ERASE 1
*LABEL 100
! IDV = IDV + 1
*JGZ,-1(NCNT,10)
$
*LIBS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*ONLINE=1
*XQT U1
*REGISTER RETRIEVE(1 DXDV REGISTERS 1 1)
*RETURN
*END
```

Runstream TR DPREP

TR DPREP is a utility runstream used by all the sensitivity calculation runstreams. Its main task is to locate the critical points for all required response quantities.

```
$-----------------------------------------------------------------
$ (TR,DPREP) - PREPARATION FOR SENSITIVITY CALCULATIONS
```

```
$-------------------------------------------------------------
$
$ FORM CRITICAL POINT TABLES FOR RESPONSE QUANTITIES
$
! NBCK = TOC,NI("QLIB" BACK LIST 1 1)
! I = 1 : ! N = NBCK
*LABEL 10
! NM = DS,"I",1,1("QLIB" BACK LIST 1 1)
! IERR = TOC,IERR("QLIB" SEL "NM" MASK MASK)
*JNZ(IERR,20)
*XQT U10
  CRIT(Y="QLIB" HIST "NM",DT="DT", NCRIT="NCRIT", &
       CRPT="QLIB" CRPT "NM",CRTI="QLIB" CRTI "NM", &
       PCH=.25)
*XQT DCU
 PRINT "QLIB" CRTI "NM"
 PRINT "QLIB" CRPT "NM"
*LABEL 20
! I = I + 1
*JGZ,-1(N,10)
*XQT U1
*(E4 PARAMETERS) EOFX
$RESET NFCT="NMODES", NLIM="NMODES"
 RESET NIF="NMODES"
 IFSOURCE= "XLIB" VIBR MODE 1 1
*EOFX
*END
```

Runstream TR DBACK 1

TR DBACK n runstreams implement the different procedures for recovering the physical sensitivities. They all rely heavily on runstream TR CRPT which recovers a specific physical quantity at the critical points. Runstream TR DBACK 1 recovers the sensitivities with the mode displacement method and is used in the overall finite difference procedures.

```
$-------------------------------------------------------------
$ (TR DBACK 1) - BACK TRANSFORMATION FOR DERIVATIVES USING MODE
$                DISPLACEMENT METHOD
$-------------------------------------------------------------
! IIB = 1 : ! NNB = NBCK
*LABEL 10
*XQT AUS
 DEFINE X = VIBR MODE 1 1 1,"NMODES"
! NM = DS,"IIB",1,1("XLIB" BACK LIST 1 1)
! IERR = TOC,IERR("XLIB" SEL "NM" MASK MASK)
*JNZ(IERR,200)
$
$ DISPLACEMENTS
$
*IF("NM" NE DISP): *GOTO 30
 DEFINE IDJK = "XLIB" SEL DISP
 TMAT VMOD = SVTRAN(IDJK,X)
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3=1 TNAME=VMOD CNAME=DISP Q=QX
*GOTO 200
*LABEL 30
$
$ VELOCITIES
$
*IF("NM" NE VELO): *GOTO 50
 DEFINE IDJK = "XLIB" SEL VELO
 TMAT VVEL = SVTRAN(IDJK,X)
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3=1 TNAME=VVEL CNAME=VELO Q=QX1
*GOTO 200
*LABEL 50
$
```

```
 $ ACCELERATIONS
 $
 *IF("NM" NE ACCE): *GOTO 70
  DEFINE IDJK = "XLIB" SEL ACCE
  TMAT VACC = SVTRAN(IDJK,X)
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3=1 TNAME=VACC CNAME=ACCE Q=QX2
 *GOTO 200
 *LABEL 70
 $
 $ REACTIONS
 $
 *IF("NM" NE REAC): *GOTO 90
 *GOTO 200
 *LABEL 90
 $
 $ STRESSES
 $
 *IF("NM" NE STRE): *GOTO 110
 *XQT ES
  RESET OPER=T
  IDQ = "XLIB" SEL STRESS
  U = VIBR MODE 1 1 1,"NMODES"
  T = TMAT VSTRE
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3=1 TNAME=VSTRE CNAME=STRE Q=QX
 *GOTO 200
 *LABEL 110
 *LABEL 200
 ! IIB = IIB + 1
 *JGZ,-1(NNB,10)
 ! IIB = FREE() : ! NNB = FREE() : ! LIB1 = FREE() : ! LIB2 = FREE()
 ! LIB3 = FREE() : ! TNAME = FREE() : ! CNAME = FREE()
 ! Q = FREE()
 *END
```

Runstream TR DBACK 2

Runstream TR DBACK 2 recovers sensitivities in the fixed-mode, mode displacement version of the semianalytical method.

```
$-----------------------------------------------------------------
$ (TR DBACK 2) - BACK TRANSFORMATION FOR DERIVATIVES USING
$                SEMIANALYTICAL METHOD
$ UPDATE HISTORY
$ --------------
$ 6/22/88 WHG - MODIFIED FOR SEMIANALYTICAL METHOD
$ 6/21/88 WHG - CREATED FROM (TR,DBACK,1) FOR UPDATED MODES
$-----------------------------------------------------------------
! IIB = 1 : ! NNB = NBCK
*LABEL 10
*XQT AUS
! NM = DS,"IIB",1,1("XLIB" BACK LIST 1 1)
! IERR = TOC,IERR("XLIB" SEL "NM" MASK MASK)
*JNZ(IERR,300)
$
$ DISPLACEMENTS
$
*IF("NM" NE DISP): *GOTO 30
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VMOD CNAME=DISP Q=QX
*GOTO 200
*LABEL 30
$
$ VELOCITIES
$
*IF("NM" NE VELO): *GOTO 50
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VVEL CNAME=VELO Q=QX1
*GOTO 200
```

```
*LABEL 50
$
$ ACCELERATIONS
$
*IF("NM" NE ACCE): *GOTO 70
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VACC CNAME=ACCE Q=QX2
*GOTO 200
*LABEL 70
$
$ REACTIONS
$
*IF("NM" NE REAC): *GOTO 90
*GOTO 200
*LABEL 90
$
$ STRESSES
$
*IF("NM" NE STRE): *GOTO 110
$ FORM [S] [DQ/DV]
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VSTRE CNAME=STRE Q=QX
*XQT DCU
 CHANGE 1 CRPT STRE 1 1 CRPT STR1 1 1
$ FORM [DS/DV] [Q]
*XQT ES
 RESET OPER=T
 IDQ = "XLIB" SEL STRESS
 U = "XLIB" VIBR MODES 1 1 1,"NMODES"
 T = TMAT VSTRE
*XQT AUS
 DEFINE S0 = "XLIB" TMAT VSTRESS
 DEFINE S1 =        TMAT VSTRESS
 TMAT DSDV = SUM("OVDX" S1 "MOVD" S0)
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DSDV CNAME=STRE Q=QX
*XQT AUS
 DEFINE STR1 = CRPT STR1
 DEFINE STR2 = CRPT STRE
 CRPT STRE = SUM(STR1,STR2)
*GOTO 200
*LABEL 110
*GOTO 300
*LABEL 200
*XQT DCU
 CHANGE 1 CRPT "NM" 1 1 DXDV "NM" "IDV" 1
 COPY 1 "XLIB" DXDV "NM" "IDV" 1
 PRINT "XLIB" DXDV "NM" "IDV" 1
*LABEL 300
! IIB = IIB + 1
*JGZ,-1(NNB,10)
! IIB = FREE() : ! NNB = FREE() : ! LIB1 = FREE() : ! LIB2 = FREE()
! LIB3 = FREE() : ! TNAME = FREE() : ! CNAME = FREE()
! Q = FREE()
*END
```

Runstream TR DBACK 3

Runstream TR DBACK 3 recovers sensitivities in the semianalytical method with nonzero $d\Phi/dx$.

```
$-----------------------------------------------------------------
$ (TR DBACK 3) - BACK TRANSFORMATION FOR DERIVATIVES USING
$                SEMIANALYTICAL METHOD WITH CHANGING MODES
$ UPDATE HISTORY
$ --------------
$ 6/22/88 WHG - MODIFIED FOR SEMIANALYTICAL METHOD
$ 6/21/88 WHG - CREATED FROM (TR,DBACK,1) FOR UPDATED MODES
$-----------------------------------------------------------------
```

```
 ! IIB = 1 : ! NNB = NBCK
 *LABEL 10
 *XQT AUS
  DEFINE DX = DXDV AUS "IDV" 1 1,"NMODES"
 ! NM = DS,"IIB",1,1("XLIB" BACK LIST 1 1)
 ! IERR = TOC,IERR("XLIB" SEL "NM" MASK MASK)
 *JNZ(IERR,300)
 $
 $ DISPLACEMENTS
 $
 *IF("NM" NE DISP): *GOTO 30
  DEFINE IDJK = "XLIB" SEL DISP
  TMAT DVMX = SVTRAN(IDJK,DX)
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VMOD CNAME=DISP Q=QX
 *XQT DCU
  CHANGE 1 CRPT DISP 1 1 CRPT DSP1 1 1
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DVMX CNAME=DISP Q=QX
 *XQT AUS
  DEFINE D1 = CRPT DSP1
  DEFINE D2 = CRPT DISP
  CRPT DISP = SUM(D1,D2)
 *GOTO 200
 *LABEL 30
 $
 $ VELOCITIES
 $
 *IF("NM" NE VELO): *GOTO 50
  DEFINE IDJK = "XLIB" SEL VELO
  TMAT DVMX = SVTRAN(IDJK,DX)
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VVEL CNAME=VELO Q=QX1
 *XQT DCU
  CHANGE 1 CRPT VELO 1 1 CRPT VEL1 1 1
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DVMX CNAME=VELO Q=QX1
 *XQT AUS
  DEFINE V1 = CRPT VEL1
  DEFINE V2 = CRPT VELO
  CRPT VELO = SUM(V1,V2)
 *GOTO 200
 *LABEL 50
 $
 $ ACCELERATIONS
 $
 *IF("NM" NE ACCE): *GOTO 70
  DEFINE IDJK = "XLIB" SEL ACCE
  TMAT DVMX = SVTRAN(IDJK,DX)
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VACC CNAME=ACCE Q=QX2
 *XQT DCU
  CHANGE 1 CRPT ACCE 1 1 CRPT ACC1 1 1
 *DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DVMX CNAME=ACCE Q=QX2
 *XQT AUS
  DEFINE A1 = CRPT ACC1
  DEFINE A2 = CRPT ACCE
  CRPT ACCE = SUM(A1,A2)
 *GOTO 200
 *LABEL 70
 $
 $ REACTIONS
 $
 *IF("NM" NE REAC): *GOTO 90
 *GOTO 200
 *LABEL 90
 $
 $ STRESSES
 $
 *IF("NM" NE STRE): *GOTO 110
 $ FORM [S] [DQ/DV]
```

```
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VSTRE CNAME=STRE Q=QX
*XQT DCU
 CHANGE 1 CRPT STRE 1 1 CRPT STR1 1 1
$ FORM [DS/DV] [Q]
*XQT ES
 RESET OPER=T
 IDQ = "XLIB" SEL STRESS
 U = "XLIB" VIBR MODES 1 1 1,"NMODES"
 T = TMAT VSTRE
*XQT AUS
 DEFINE S0 = "XLIB" TMAT VSTRESS
 DEFINE S1 =        TMAT VSTRESS
 TMAT DSDV = SUM("OVDX" S1 "MOVD" S0)
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DSDV CNAME=STRE Q=QX
*XQT DCU
 CHANGE 1 CRPT STRE 1 1 CRPT STR2 1 1
$ FORM S [D PHI / DV] [Q]
*XQT ES
 RESET OPER=T
 IDQ = "XLIB" SEL STRESS
 U = DXDV AUS "IDV" 1 1,"NMODES"
 T = TMAT DSTRE
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DSTRE CNAME=STRE Q=QX
*XQT AUS
 DEFINE STR1 = CRPT STR1
 DEFINE STR2 = CRPT STR2
 DEFINE STR3 = CRPT STRE
 TMP = SUM(STR1,STR2)
 CRPT STRE = SUM(TMP,STR3)
*GOTO 200
*LABEL 110
*GOTO 300
*LABEL 200
*XQT DCU
 CHANGE 1 CRPT "NM" 1 1 DXDV "NM" "IDV" 1
 COPY 1 "XLIB" DXDV "NM" "IDV" 1
 PRINT "XLIB" DXDV "NM" "IDV" 1
*LABEL 300
! IIB = IIB + 1
*JGZ,-1(NNB,10)
! IIB = FREE() : ! NNB = FREE() : ! LIB1 = FREE() : ! LIB2 = FREE()
! LIB3 = FREE() : ! TNAME = FREE() : ! CNAME = FREE()
! Q = FREE()
*END
```

Runstream TR DBACK 4

Runstream TR DBACK 4 recovers sensitivities in the fixed-mode semianalytical method with the mode acceleration method.

```
$-----------------------------------------------------------------
$ (TR DBACK 4) - BACK TRANSFORMATION FOR DERIVATIVES USING
$                SEMIANALYTICAL METHOD
$                WITH THE MODE ACCELERATION METHOD
$-----------------------------------------------------------------
*XQT AUS
 DEFINE X = "XLIB" VIBR MODE 1 1 1,"NMODES"
 DEFINE E = "XLIB" VIBR EVAL 1 1
 DEFINE DKX = DKX AUS
 DEFINE DMX = DMX AUS
 DEFINE ROMG = "XLIB" ROMG AUS
 DEFINE XTDX = "XLIB" XTDX
 DEFINE XOMD = "XLIB" XOMD
 DEFINE XOME = "XLIB" XOME
*IF("IDSP" NE 0): TABLE(NI="NMODES",NJ="NMODES") : RDDX
$ CALCULATE VELOCITY TERM
```

```
 XOM1 = CBR(XOME,RDKX)
 XOMK = CBD(XOM1,ROMG)
! NJDM = TOC,NJ("XLIB" XTDX MASK MASK MASK)
! NBDM = TOC,NINJ("XLIB" XTDX MASK MASK MASK)
*IF("NJDM" NE "NBDM"): XOKC = CBR(XOMK,XTDX)
*IF("NJDM" EQ "NBDM"): XOKC = CBD(XOMK,XTDX)
 XOMC = CBR(XOME,RDDX)
 XQD = SUM(XOKC, -1.0 XOMC)
$ CALCULATE ACCELERATION TERM
 DKXO = CBD(DKX, ROMG)
 APPL FORC 887 = SUM(DKXO, -1.0 DMX)
$ CALCULATE DERIVATIVE OF THE PSEUDOSTATIC TERM
 DEFINE USTAT = "XLIB" STAT DISP 1 1
 FSL1 = PROD(DKDV,USTAT)
 APPL FORC 888 = SUM(DFDV,-1.0 FSL1)
*XQT SSOL
 RESET SET=887, KLIB="XLIB", KILIB="XLIB", REAC=0
*XQT SSOL
 RESET SET=888, KLIB="XLIB", KILIB="XLIB", REAC=0
$
$ LOOP OVER ALL RESPONSE QUANTITY TYPES
$
! IIB = 1 : ! NNB = NBCK
*LABEL 10
*XQT AUS
! NM = DS,"IIB",1,1("XLIB" BACK LIST 1 1)
! IERR = TOC,IERR("XLIB" SEL "NM" MASK MASK)
*JNZ(IERR,300)
$
$ DISPLACEMENTS
$
*IF("NM" NE DISP): *GOTO 30
 DEFINE IDJK = "XLIB" SEL DISP
 DEFINE DAC1 = STAT DISP 887
 DEFINE DUST = STAT DISP 888
 DEFINE XTDX = "XLIB" XTDX
 DEFINE DACC = "XLIB" TMAT DACC
 TMAT DUST = SVTRAN(IDJK,DUST)
 TMAT DAC1 = SVTRAN(IDJK,DAC1)
$ VELOCITY TERMS
 TMAT DVL1 = SVTRAN(IDJK,XQD)
$
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DUST CNAME=DISP Q=A
 TOCC(1 CRPT DISP 1 1) : N2=DSP1
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DVL1 CNAME=DISP Q=QX1
 TOCC(1 CRPT DISP 1 1) : N2=DSP2
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=DVEL CNAME=DISP Q=QX1
 TOCC(1 CRPT DISP 1 1) : N2=DSP3
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DAC1 CNAME=DISP Q=QX2
 TOCC(1 CRPT DISP 1 1) : N2=DSP4
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=DACC CNAME=DISP Q=QX2
 TOCC(1 CRPT DISP 1 1) : N2=DSP5
*XQT U4
 VU
 DEFINE D1 = CRPT DSP1
 DEFINE D2 = CRPT DSP2
 DEFINE D3 = CRPT DSP3
 DEFINE D4 = CRPT DSP4
 DEFINE D5 = CRPT DSP5
 CRPT DISP = SUM(D1, D2, -1.0*D3, D4, -1.0*D5)
*GOTO 200
*LABEL 30
$
$ VELOCITIES
$
*IF("NM" NE VELO): *GOTO 50
```

```
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VVEL CNAME=VELO Q=QX1
*GOTO 200
*LABEL 50
$
$ ACCELERATIONS
$
*IF("NM" NE ACCE): *GOTO 70
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=VACC CNAME=ACCE Q=QX2
*GOTO 200
*LABEL 70
$
$ REACTIONS
$
*IF("NM" NE REAC): *GOTO 90
*GOTO 200
*LABEL 90
$
$ STRESSES
$
*IF("NM" NE STRE): *GOTO 110
$
$ FORM [S] [DU/DV]
$
*LIBS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT ES
 RESET OPER=T
 IDQ = SEL STRESS
 U="XLIB" STAT DISP 888 : T = "XLIB" TMAT DTM1
 U="XLIB" XQD AUS       : T = "XLIB" TMAT DTM2
 U="XLIB" STAT DISP 887 : T = "XLIB" TMAT DTM4
*LIBS "XLIB" 2 3 4 1 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
*XQT AUS
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DTM1 CNAME=STRE Q=A
 TOCC(1 CRPT STRE 1 1) : N2=STR1
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DTM2 CNAME=STRE Q=QX1
 TOCC(1 CRPT STRE 1 1) : N2=STR2
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=SD CNAME=STRE Q=QX1
 TOCC(1 CRPT STRE 1 1) : N2=STR3
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DTM4 CNAME=STRE Q=QX2
 TOCC(1 CRPT STRE 1 1) : N2=STR4
*DCALL(TR,CRPT) LIB1="XLIB" LIB2="XLIB" LIB3=1 TNAME=SP CNAME=STRE Q=QX2
 TOCC(1 CRPT STRE 1 1) : N2=STR5
$
$ FORM [DS/DV] [U]
$
*XQT ES
 RESET OPER=T
 IDQ = "XLIB" SEL STRESS
 U = "XLIB" STAT DISP 1 1  : T = TMAT SF
 U = "XLIB" XOMD AUS       : T = TMAT SD
 U = "XLIB" XOME AUS       : T = TMAT SP
*XQT AUS
 DEFINE S0 = "XLIB" TMAT SF : DEFINE S1 = TMAT SF
 TMAT DSF = SUM("OVDX" S1 "MOVD" S0)
 DEFINE S0 = "XLIB" TMAT SD : DEFINE S1 =        TMAT SD
 TMAT DSD = SUM("OVDX" S1 "MOVD" S0)
 DEFINE S0 = "XLIB" TMAT SP : DEFINE S1 =        TMAT SP
 TMAT DSP = SUM("OVDX" S1 "MOVD" S0)
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DSF CNAME=STRE Q=A
 TOCC(1 CRPT STRE) : N2=STR6
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DSD CNAME=STRE Q=QX1
 TOCC(1 CRPT STRE) : N2=STR7
*DCALL(TR,CRPT) LIB1="XLIB" LIB2=1 LIB3="XLIB" TNAME=DSP CNAME=STRE Q=QX2
 TOCC(1 CRPT STRE) : N2=STR8
*XQT U4
 VU
```

```
 DEFINE S1 = CRPT STR1
 DEFINE S2 = CRPT STR2
 DEFINE S3 = CRPT STR3
 DEFINE S4 = CRPT STR4
 DEFINE S5 = CRPT STR5
 DEFINE S6 = CRPT STR6
 DEFINE S7 = CRPT STR7
 DEFINE S8 = CRPT STR8
 CRPT STRE = SUM(S1, S2, -1.0*S3, S4, -1.0*S5, S6, -1.0*S7, -1.0*S8)
*GOTO 200
*LABEL 110
*GOTO 300
*LABEL 200
*XQT DCU
 CHANGE 1 CRPT "NM" 1 1 DXDV "NM" "IDV" 1
 COPY 1 "XLIB" DXDV "NM" "IDV" 1
 PRINT "XLIB" DXDV "NM" "IDV" 1
*LABEL 300
! IIB = IIB + 1
*JGZ,-1(NNB,10)
! IIB = FREE() : ! NNB = FREE() : ! LIB1 = FREE() : ! LIB2 = FREE()
! LIB3 = FREE() : ! TNAME = FREE() : ! CNAME = FREE()
! Q = FREE()
*END
```

Runstream TR DPHI 3

Runstream TR DPHI 3 implements the modified modal method for calculating eigenvector derivatives and is called from sensitivity calculation runstream TR DXDV 6.

```
$-----------------------------------------------------------------
$ (TR,DPHI,3) - CALCULATE EIGENVECTOR DERIVATIVES USING THE
$               MODIFIED MODAL METHOD
$-----------------------------------------------------------------
*XQT AUS
 INLIB=10 : OUTLIB=10
 DEFINE MO = "XLIB" "MNAME"
 DEFINE DK = 1 DKDV SPAR
 DEFINE DM = 1 DMDV SPAR
 DEFINE XO = "XLIB" VIBR MODE 1 1 1,"NMODES"
 DEFINE AJK = AJK
 DEFINE EV = "XLIB" VIBR EVAL 1 1
 MX = PROD(DM,XO)
 AKK = XTYD(-.5 XO,MX)
 TABLE(NI="NMODES",NJ="NMODES") :  AJK
 TABLE(NI=1,NJ="NMODES") :  UNIT : J=1,"NMODES" : 1.0
! J = 1 : ! NJ = NMODES
*LABEL 10
! EJ = DS,"J",1,1("XLIB" VIBR EVAL 1 1)
! MEJ = -EJ
 DEFINE XJ = "XLIB" VIBR MODE 1 1 "J","J"
 DKDM = SUM(DK,"MEJ" DM)
 MX = PROD(DKDM,XJ)
 DLAM = XTY(XJ,MX)
! DLAM = DS,1,1,1(10 DLAM AUS 1 1)
 AF1 = PROD("DLAM" MO, XJ)
 AF2 = SUM(AF1 -1.0 MX)
*IF("J" EQ 1): 11 APPL FORC = UNION(AF2)
*IF("J" NE 1): 11 APPL FORC = UNION,U(AF2)
 AA = XTY(XO,MX)
 DEN1 = SUM(-1.0 EV, "EJ" UNIT)
 DEN2 = PROD(EV,DEN1)
 TABLE,U : DEN2 : I="J" : J=1 : 1.0
 FACT = RECIP(DEN2)
 AAB = PROD("EJ" FACT,AA)
 DE1 : OPER=XSUM : DEST,U=AJK AUS
```

```
      SOURCE=AAB : JS=1 : JD="J" : EX1
      SOURCE=AKK : IS="J" : JS=1 : ID="J" : JD="J" : EX1
! J = J + 1
*JGZ,-1(NJ,10)
 DXDV = CBR(XO,AJK)
*XQT SSOL
 RESET KLIB="XLIB", KILIB="XLIB", QLIB=11, REAC=0, EP=0
*XQT AUS
 DEFINE D1 = 10 DXDV AUS
 DEFINE D2 = 11 STAT DISP
 DXDV AUS "IDV" = SUM(D1,D2)
*DELETE 10
*DELETE 11
*RETURN
*END
```

Runstream TR CRPT

TR CRPT is a utility runstream which performs the transformation from modal to physical basis for a single response quantity at a set of critical times. It is heavily used by the TR DBACK runstreams.

```
$-----------------------------------------------------------------
$ (TR,CRPT) - FORM CRITICAL POINT RESPONSE TABLE
$-----------------------------------------------------------------
*ONLINE=0
*XQT AUS
! NCRIT = TOC,NI("LIB1" CRTI "CNAME" 1 1) $ NUMBER OF TIMES
! ND    = TOC,NI("LIB2" TMAT "TNAME" 1 1) $ NUMBER OF RESP. QUANTITIES
! NQ    = TOC,NJ("LIB2" TMAT "TNAME" 1 1) $ NUMBER OF MODES
! NJQ = TOC,NJ("LIB3" "Q" AUS MASK MASK)
! ISTP = 0
$ LOOP OVER ALL RESPONSE QUANTITIES
! I = 1
! N = ND
 INLIB = 21 : OUTLIB = 21
*LABEL 20
 DE1
 SOURCE="LIB2" TMAT "TNAME"
 ID = 1 : IS="I"
 DEST=TONE "TNAME" "I" 1
 EX1
 TABLE(NI="NQ", NJ="NCRIT") : XBAR CRIT "I"
$ LOOP OVER NUMBER OF CRITICAL POINTS
! II = 1
! NN = NCRIT
*LABEL 40
 DE1
! TIME = DS,"II",1,"I"("LIB1" CRTI "CNAME" 1 1)
! ISTP = TIME/DT + .5
! ISTP = ISTP + 1
! IB = ISTP/NJQ
! IST = IB*NJQ
*IF("IST" NE "ISTP"): ! IB = IB + 1
! J = IB - 1 * NJQ : ! J = ISTP - J
 SOURCE = "LIB3" "Q" AUS MASK MASK "IB" : JS="J" : DEST,U=XBAR CRIT "I"
 JD = "II" : EX1
! II = II + 1
*JGZ,-1(NN,40)
 DEFINE T = TONE "TNAME" "I"
 DEFINE XBAR = XBAR CRIT "I"
 CI AUS "I" = RPROD(T,XBAR)
 TOCC(CI AUS "I") : NJ=1
 DEFINE CI = CI AUS "I"
*IF("I" EQ 1): 1 CRPT "CNAME" = UNION(CI)
*IF("I" GT 1): 1 CRPT "CNAME" = UNION,U(CI)
```

```
! I = I + 1
*JGZ,-1(N,20)
*ONLINE=1
! NCRIT = FREE() : ! ND = FREE() : ! NQ = FREE() ! NJQ = FREE()
! ISTP = FREE() : ! IB = FREE() : ! IST = FREE()
*END
```

Runstream TR DIAG

TR DIAG is a utility runstream that solves the reduced-order eigenproblem based on a given set of basis vectors to uncouple a reduced system.

```
$-----------------------------------------------------------
$(TR DIAG) - MAKE AN ARBITRARY VECTOR SET M AND K ORTHONORMAL
$-----------------------------------------------------------
* XQT AUS
      OUTLIB=10: INLIB=10
      DEFINE K=1 K: DEFI M=1 "MNAME"
      DEFINE X = "VLIB" VIBR MODE
      IJCODE=10000
      KX=PROD(K,X): SYN K 10000 "NMODE" = XTYS(X,KX)
      MX=PROD(M,X): SYN M 10000 "NMODE" = XTYS(X,MX)
! ZERO=NMODE-1
* JZ (ZERO,1003)
* XQT STRP
      RESET SOURCE=10, DEST=10
* JGZ (ZERO,1004)
* LABEL 1003
* XQT AUS
      OUTLIB=10: INLIB=10
      !K=DS 2 1 1(10 SYN K MASK MASK)
      !M=DS 2 1 1(10 SYN M MASK MASK)
      !EVAL=K/M
      TABLE(NI=1,NJ=1): SYS EVEC: J=1: 1.0
      TABLE(NI=1,NJ=1): SYS EVAL: J=1: "EVAL"
* LABEL 1004
* XQT AUS
      OUTLIB=10: INLIB=10
      DEFINE E=SYS EVEC
      DEFI X = "VLIB" VIBR MODE
      X ORTH 1 1=CBR(X,E)
      DEFINE X=X ORTH 1 1
      "VLIB" VIBR MODE 1 1=UNION(X)
      DEFINE E=SYS EVAL
      "VLIB" VIBR EVAL 1 1=UNION(E)
*END
```

References

1. Adelman, Howard M.; and Haftka, Raphael T.: Sensitivity Analysis of Discrete Structural Systems. *AIAA J.*, vol. 24, no. 5, May 1986, pp. 823–832.
2. Haftka, R. T.; and Adelman, H. M.: Recent Developments in Structural Sensitivity Analysis. *Struct. Optim.*, vol. 1, no. 3, Sept. 1989, pp. 137–151.
3. Fox, R. L.; and Kapoor, M. P.: Structural Optimization in the Dynamics Response Regime: A Computational Approach. *AIAA J.*, vol. 8, no. 10, Oct. 1970, pp. 1798 1804.
4. Cassis, Juan H.; and Schmit, Lucien A., Jr.: Optimum Structural Design With Dynamic Constraints. *J. Struct. Div., American Soc. Civ. Eng.*, vol. 102, no. ST10, Oct. 1976, pp. 2053–2071.
5. Cassis, Juan H.: *Optimum Design of Structures Subjected to Dynamic Loads.* UCLA-ENG-7451, Mechanics and Structures Dep., Univ. of California, Los Angeles, June 1974.
6. Hsiao, M. H.; Haug, E. J., Jr.; and Arora, J. S.: Mechanical Design Optimization for Transient Dynamic Response. 76-WA/DE-27, American Soc. of Mechanical Engineers, Dec. 1976.
7. Haug, Edward J.; and Arora, Jasbir S.: Design Sensitivity Analysis of Elastic Mechanical Systems. *Comput. Methods in Appl. Mech. & Eng.*, vol. 15, no. 1, July 1978, pp. 35–62.
8. Feng, T-T.; Arora, J. S.; and Haug, E. J., Jr.: Optimal Structural Design Under Dynamic Loads. *Int. J. Numer. Methods Eng.*, vol. 11, no. 1, 1977, pp. 39–52.
9. Haug, E. J.; Wehage, R.; and Barman, N. C.: Design Sensitivity Analysis of Planar Mechanism and Machine Dynamics. *J. Mech. Design*, vol. 103, no. 3, July 1981, pp. 560 570.
10. Haug, Edward J.: Design Sensitivity Analysis of Dynamic Systems. *Computer Aided Optimal Design: Structural and Mechanical Systems*, Carlos A. Mota Soares, ed., Springer-Verlag, c.1987, pp. 705–755.
11. Hsieh, C. C.; and Arora, J. S.: Design Sensitivity Analysis and Optimization of Dynamic Response. *Comput. Methods Appl. Mech. & Eng.*, vol. 43, no. 2, Apr. 1984, pp. 195 219.
12. Ray, D.; Pister, K. S.; and Polak, E.: Sensitivity Analysis for Hysteretic Dynamic Systems: Theory and Applications. *Comput. Methods Appl. Mech. & Eng.*, vol. 14, no. 2, May 1978, pp. 179–208.
13. Meric, R. A.: Shape Design Sensitivity Analysis of Dynamic Structures. *AIAA J.*, vol. 26, no. 2, Feb. 1988, pp. 206–212.
14. Tortorelli, Daniel A.; Haber, Robert B.; and Lu, Stephen C-Y.: Shape Sensitivities for Nonlinear Dynamic Thermoelastic Structures. *A Collection of Technical Papers, Part 3—AIAA/ASME/ASCE/AHS/ASC 30th Structures, Structural Dynamics and Materials Conference*, Apr. 1989, pp. 1321–1329. (Available as AIAA-89-1311-CP.)
15. Bathe, Klaus-Jürgen: *Finite Element Procedures in Engineering Analysis.* Prentice-Hall, Inc., c.1982.
16. Craig, Roy R., Jr.: *Structural Dynamics—An Introduction to Computer Methods.* John Wiley & Sons, Inc., c.1981.
17. *The NASTRAN® Theoretical Manual.* NASA SP-221(06), 1981.
18. Williams, D.; and Jones, R. P. N.: *Dynamic Loads in Aeroplanes Under Given Impulsive Loads With Particular Reference to Landing and Gust Loads on a Large Flying Boat.* R. & M. No. 2221, British Aeronautical Research Council, 1948.
19. Léger, P.; and Wilson, E. L.: Modal Summation Methods for Structural Dynamic Computations. *Earthq. Eng. & Struct. Dyn.*, vol. 16, no. 1, Jan. 1988, pp. 23–27.
20. Wilson, Edward L.; Yuan, Ming-Wu; and Dickens, John M.: Dynamic Analysis by Direct Superposition of Ritz Vectors. *Earthq. Eng. & Struct. Dyn.*, vol. 10, no. 6, Nov.–Dec. 1982, pp. 813–821.
21. Nour-Omid, Bahram; and Clough, Ray W.: Dynamic Analysis of Structures Using Lanczos Co-ordinates. *Earthq. Eng. & Struct. Dyn.*, vol. 12, no. 4, July Aug. 1984, pp. 565–577.
22. Kline, Kenneth A.: Dynamic Analysis Using a Reduced Basis of Exact Modes and Ritz Vectors. *AIAA J.*, vol. 24, no. 12, Dec. 1986, pp. 2022–2029.
23. Whetstone, W. D.: *User Instructions—EISI-EAL Version 312.08.* Engineering Information Systems, Inc., Aug. 1985.
24. Brenneman, Benjamin, III: An Analytic Solution of Systems of Linear Ordinary Differential Equations Useful for Numerical Integration. M.S. Thesis, Virginia Polytechnic Inst. and State Univ., 1973.
25. Melosh, Robert J.: Integration of Linear Equations of Motion. *J. Struct. Div., American Soc. Civ. Eng.*, vol. 101, no. ST7, July 1975, pp. 1551 1558.
26. Kahaner, David; Moler, Cleve; and Nash, Stephen: *Numerical Methods and Software.* Prentice-Hall, Inc., c.1989.
27. Haftka, Raphael T.; and Kamat, Monohar P.: *Elements of Structural Optimization.* Kluwer Academic Publ., 1985.
28. Grandhi, Ramana V.; Haftka, Raphael T.; and Watson, Layne T.: Efficient Identification of Critical Stresses in Structures Subject to Dynamic Loads. *Comput. & Struct.*, vol. 22, no. 3, 1986, pp. 373–386.
29. Grandhi, Ramana V.; Haftka, Raphael T.; and Watson, Layne T.: Design-Oriented Identification of Critical Times in Transient Reponse. *AIAA J.*, vol. 24, no. 4, Apr. 1986, pp. 649–656.
30. Sutter, Thomas R.; Camarda, Charles J.; Walsh, Joanne L.; and Adelman, Howard M.: Comparison of Several Methods for Calculating Vibration Mode Shape Derivatives. *AIAA J.*, vol. 26, no. 12, Dec. 1988, pp. 1506–1511.
31. Nelson, Richard B.: Simplified Calculation of Eigenvector Derivatives. *AIAA J.*, vol. 14, no. 9, Sept. 1976, pp. 1201–1205.
32. Wang, B. P.: An Improved Approximate Method for Computing Eigenvector Derivatives. University of Texas at Arlington paper presented at 26th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference (Orlando, Florida), Apr. 1985.
33. Sandridge, Chris A.; and Haftka, Raphael T.: Accuracy of Derivatives of Control Performance Using a Reduced

Structural Model. *A Collection of Technical Papers, Part 2B—AIAA Dynamic Specialist Meeting*, Apr. 1987, pp. 622–628. (Available as AIAA-87-0905.)

34. Greene, William H.; and Haftka, Raphael T.: Computational Aspects of Sensitivity Calculations in Transient Structural Analysis. *Comput. & Struct.*, vol. 32, no. 2, 1989, pp. 433–443.
35. Haftka, Raphael T.; and Yates, E. Carson, Jr.: Repetitive Flutter Calculations in Structural Design. *J. Aircr.*, vol. 13, no. 7, July 1976, pp. 454–461.
36. Schmit, L. A.; and Miura, H.: An Advanced Structural Analysis/Synthesis Capability—Access 2. *Int. J. Numer. Methods Eng.*, vol. 12, no. 2, 1978, pp. 353–377.
37. Fleury, Claude; and Schmit, Lucien A., Jr.: *Dual Methods and Approximation Concepts in Structural Synthesis.* NASA CR-3226, 1980.
38. Vanderplaats, Garret N.: An Efficient Feasible Directions Algorithm for Design Synthesis. *AIAA J.*, vol. 22, no. 11, Nov. 1984, pp. 1633–1640.
39. Barthelemy, Bruno: Accuracy Analysis of the Semi-Analytical Method for Shape Sensitivity Analysis. Ph.D. Thesis, Virginia Polytechnic Inst. and State Univ., 1987.
40. George, Alan; and Liu, Joseph W-H: *Computer Solution of Large Sparse Positive Definite Systems.* Prentice-Hall, Inc., c.1981.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-4156		

4. Title and Subtitle	5. Report Date
Computational Aspects of Sensitivity Calculations in Linear Transient Structural Analysis	May 1990
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
William H. Greene	L-16643
	10. Work Unit No.
9. Performing Organization Name and Address	506-43-41-02
NASA Langley Research Center Hampton, VA 23665-5225	11. Contract or Grant No.
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Technical Memorandum
National Aeronautics and Space Administration Washington, DC 20546-0001	14. Sponsoring Agency Code

15. Supplementary Notes

The information presented in this report was offered as a dissertation in partial fulfillment of the requirements for the Degree of Doctor of Philosophy in Aerospace and Ocean Engineering, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, August 1989.

16. Abstract

A study has been performed focusing on the calculation of sensitivities of response quantities in linear, structural, transient response problems. One significant goal of the study was to develop and evaluate sensitivity calculation techniques suitable for large-order finite element analyses. Accordingly, approximation vectors such as vibration mode shapes are used to reduce the dimensionality of the finite element model. Overall finite difference methods where the analysis is repeated for perturbed designs and semianalytical methods using direct differentiation of the equations of motion are considered. To be computationally practical in large-order problems, the overall finite difference methods must use the approximation vectors from the original design in the analyses of the perturbed models. In several cases, the fixed-mode approach resulted in very poor approximations of the stress sensitivities. To overcome this poor accuracy, two semianalytical techniques were developed. The first technique accounts for the change in eigenvectors through approximate eigenvector derivatives. The second technique applies the mode acceleration method of transient analysis to the sensitivity calculations. Both result in accurate values of the stress sensitivities with a small number of modes and low computational cost.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Structural dynamics Sensitivity analysis Automated design Finite element analysis Structural optimization Structural design	Unclassified--Unlimited Subject Category 39

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	107	A06

NASA-Langley, 1990