

Old Dominion University

## ODU Digital Commons

---

Civil & Environmental Engineering Theses & Dissertations

Civil & Environmental Engineering

---

Summer 1991

### Parallel-Vector Design Sensitivity Analysis in Structural Dynamics

Yongxing Zhang  
*Old Dominion University*

Follow this and additional works at: [https://digitalcommons.odu.edu/cee\\_etds](https://digitalcommons.odu.edu/cee_etds)



Part of the [Civil Engineering Commons](#)

---

#### Recommended Citation

Zhang, Yongxing. "Parallel-Vector Design Sensitivity Analysis in Structural Dynamics" (1991). Doctor of Philosophy (PhD), Dissertation, Civil & Environmental Engineering, Old Dominion University, DOI: 10.25777/ewck-qy91  
[https://digitalcommons.odu.edu/cee\\_etds/94](https://digitalcommons.odu.edu/cee_etds/94)

This Dissertation is brought to you for free and open access by the Civil & Environmental Engineering at ODU Digital Commons. It has been accepted for inclusion in Civil & Environmental Engineering Theses & Dissertations by an authorized administrator of ODU Digital Commons. For more information, please contact [digitalcommons@odu.edu](mailto:digitalcommons@odu.edu).

**PARALLEL-VECTOR DESIGN SENSITIVITY ANALYSIS  
IN STRUCTURAL DYNAMICS**

By

**Yongxing Zhang**

**B.S., January 1982, Shaanxi Institute of Mechanical Engineering**

**M.S., November 1986, Michigan Technological University**

**A Dissertation Submitted to the Faculty of  
Old Dominion University in Partial Fulfillment of the  
Requirements for the Degree of**

**DOCTOR OF PHILOSOPHY**

in

**CIVIL ENGINEERING**

**OLD DOMINION UNIVERSITY**

**June, 1991**

Approved by:

\_\_\_\_\_  
Dr. Duc T. Nguyen (Director)

\_\_\_\_\_  
Dr. Chuh Mei

\_\_\_\_\_  
Dr. Gene W. Hou

\_\_\_\_\_  
Dr. Leon R.L. Wang

## **ABSTRACT**

### **PARALLEL-VECTOR DESIGN SENSITIVITY ANALYSIS IN STRUCTURAL DYNAMICS**

Yongxing Zhang  
Old Dominion University, 1991  
Director: Dr. Duc T. Nguyen

In this study, the design sensitivity analysis is for the purpose of providing constraint derivative information for structural optimization under dynamic loads. Various existing formulations are reviewed, and the direct differentiation method is justified as the best one for design sensitivity analysis in structural dynamics. An alternative formulation for design sensitivity analysis with direct differentiation method is developed. The alternative formulation works efficiently with the reduced system of dynamic equations, and it eliminates the need for expensive and complicated eigenvector derivatives, which is required in the existing reduced system formulation. The relationship of the alternative formulation and the existing reduced system formulation is established originally, and it is proven analytically that the two approaches are identical, when the transformation is exact, i.e, when all the modes are included. The alternative approach is accurate, simple, and efficient.

Eigenvectors are used as the base vectors in system reduction for both dynamic response analysis and the design sensitivity analysis. Lanczos algorithm is

used for eigensystem solutions. A modified mode acceleration method is presented, thus, not only the displacements but also the velocities and accelerations are shown to be improved.

The accuracy of the dynamic response is checked by comparing with the original full system solution, and the accuracy of the sensitivity information is verified by comparing with the sensitivity information obtained by finite difference method of the original full system. Numerical studies have verified that the alternative formulation proposed could yield excellent accuracy. Numerical studies also show that the modal acceleration method could very effectively reduce the computation cost for both dynamic response analysis and design sensitivity analysis.

An efficient parallel-vector algorithm for design sensitivity analysis in large-scale structural dynamics is developed. Parallel computation can be achieved in both the global and local levels. The developed parallel-vector algorithm is then implemented in the Cray 2 and Cray Y-MP parallel computers using a parallel Fortran language called Force. The efficiency of the parallel-vector algorithm is illustrated by analyzing of large-scale structural systems and making comparison with the sequential version of the algorithm.

## **ACKNOWLEDGEMENT**

I wish to express my sincere gratitude to my advisor, Dr. Duc T. Nguyen, for his valuable advice and his encouragement through the course of this research. I would also like to thank the other members of my dissertation committee, Drs. Gene J. W. Hou, Chuh Mei, Leon R. L. Wang, for their suggestions, comments, and beneficial discussions.

The computer facility provided by the Computational Structural Mechanics (CSM) group at NASA Langley Research Center is also acknowledged. I wish to thank Ms. Martha A. Smith for her kind help in providing additional proof reading.

A special thanks goes to my wife Yen and my daughter Jiaxiao, their love, support, and understanding made this work possible.

Finally, I would like to thank my mother for her love, support, and encouragement.

## TABLE OF CONTENTS

	<u>Page</u>
LIST OF TABLES .....	vi
LIST OF FIGURES .....	ix
LIST OF SYMBOLS .....	xiii
CHAPTER	
1. INTRODUCTION .....	1
1.1 Overview .....	1
1.2 Review of Previous Pertinent Work .....	3
1.3 Objective and Scope .....	6
2. LARGE SCALE STRUCTURAL DYNAMIC ANALYSIS .....	11
2.1 Introduction .....	11
2.2 Eigensystem Solution in Structural Dynamics .....	12
2.2.1 The Subspace Iteration Method .....	13
2.2.2 The Lanczos Algorithm .....	15
2.2.3 The Justification of the Methods .....	18
2.3 The System Reduction Techniques .....	18

2.3.1	The Various Base Vectors .....	19
2.3.2	The Mode Displacement Method .....	22
2.3.3	The Mode Acceleration Method .....	26
2.3.4	The Modified Mode Acceleration Method .....	28
2.3.5	The Force Derivation Method .....	29
2.3.5	The Justification of the Methods .....	30
2.4	Direct Integration Schemes .....	31
3.	DESIGN SENSITIVITY ANALYSIS IN STRUCTURAL DYNAMICS	33
3.1	Introduction .....	33
3.2	The Direct Differentiation Method .....	35
3.3	The Adjoint Variable Method .....	37
3.4	The Methods Justification for DSA in Structural Dynamics .....	40
3.5	DSA Based on the Original Full System .....	41
3.6	The Reduced System Formulation .....	42
3.7	Sensitivity Calculation for Eigen-problems .....	44
3.7.1	Introduction .....	44
3.7.2	The Finite Difference Method .....	45
3.7.3	The Modal Method .....	46
3.7.4	The Modified Modal Method .....	47
3.7.5	The Nelson's Method .....	48
3.7.6	The Parallel-Vector Algorithm for the Nelson's Method ..	50

3.8 An Alternative DSA Formulation .....	51
3.9 Relationship Between the Reduced and Alternative Formulations .....	53
3.10 The Mode Acceleration Method in Design Sensitivity Analysis ..	56
3.11 Design Sensitivity Analysis of Stresses .....	57
3.12 Design Sensitivity for Non-linear Structural Dynamics .....	60
3.12.1 Formulation Based On the Original System .....	60
3.12.2 Formulation Based On the Reduced System .....	61
3.12.3 An Alternative Formulation Based On the Reduced System .....	62
 4. THE PARALLEL-VECTOR DSA ALGORITHM .....	 64
4.1 Introduction .....	64
4.2 The Parallel Fortran Language Force .....	66
4.3 Techniques and Skills Related to Vectorization .....	70
4.4 Overview of the Alternative DSA Algorithm .....	75
4.5 Designing of the Parallel-Vector Algorithm .....	77
4.5.1 Global Parallelization .....	77
4.5.2 Local Parallelization and Vectorization .....	79
4.5.3 The Parallel-Vector Equation Solver .....	79
4.5.4 The Parallel-Vector Eigensystem Solver .....	79
4.5.5 Parallel-Vector Matrix Vector Multiplication .....	80



5. NUMERICAL STUDIES .....	84
5.1 The Definition of Error Norms .....	84
5.2 Beam Examples .....	86
5.2.1 Three-Element Cantilever Beam .....	86
5.2.2 A 200-Element Cantilever Beam .....	101
5.3 Two-Dimensional Frame Examples .....	107
5.3.1 A Simple Two-Dimensional Frame .....	107
5.3.2 An 18-Bay 25-Story Two-Dimensional Frame .....	118
5.4 Three-Dimensional Frame Examples .....	132
5.4.1 A Flexible Offshore Structure .....	132
5.4.2 A Six-Story Eight-Bay Three-Dimensional Frame .....	141
5.4.3 Three Dimensional CSI Design Model .....	149
6. CONCLUSIONS AND FUTURE RESEARCH .....	160
6.1 Conclusion Remarks .....	160
6.2 Notes for Future Research .....	161
REFERENCES .....	164
APPENDIX A   RELATION BETWEEN THE ALTERNATIVE AND REDUCED SYSTEM FORMULATIONS FOR DSA IN NONLINEAR STRUCTURAL DYNAMICS .....	170

## LIST OF TABLES

<u>Table</u>	<u>Page</u>
5.1 Tip Deflection of the Three-Element Cantilever Beam (Duhamel Integral Method) . . . . .	89
5.2 Tip Deflection of the Three-Element Cantilever Beam (Newmark Method) . . . . .	90
5.3 Displacement, Velocity, and Acceleration Error Norm Analysis of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) . . . . .	95
5.4 DSA Error Norm Analysis of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) . . . . .	97
5.5 Error Norm Analysis on Stresses and Stress Derivatives of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) . . . . .	98
5.6 Algorithm Accuracy and Efficiency Analysis with the 200-Element Cantilever Beam . . . . .	103
5.7 Parallel-Vector Computation Efficiency with the 200-Element Cantilever Beam . . . . .	106
5.8 Dynamic Response Error Norm Analysis of the Simple Two-Dimensional Frame . . . . .	109
5.9 DSA Error Norm Analysis of the Simple Two-Dimensional Frame . . . . .	110

5.10 Stress and Stress Derivative Error Norm Analysis	
of the Simple Two-Dimensional Frame	114
5.11 Eigenvalues and Eigenvalue Derivatives	
of the Simple Two-Dimensional Frame	117
5.12a Displacement and Displacement Derivative Error Norm Analysis	
of the 18-Bay 25-Story Two-Dimensional Frame	121
5.12b Velocity and Acceleration Error Norm Analysis	
of the 18-Bay 25-Story Two-Dimensional Frame	121
5.13 Stress and Stress Derivative Error Norm Analysis of	
the 18-Bay 25-Story Two-Dimensional Frame	125
5.14 CPU Time Comparison of MAM Versus MDM with	
the 18-Bay 25-Story Two-Dimensional Frame	128
5.15 Eigenvalues and Eigenvalue Derivatives	
of the 18-Bay 25-Story Two-Dimensional Frame	130
5.16 Parallel-Vector Computation Efficiency	
of the 18-Bay 25-Story Two-Dimensional Frame	131
5.17 Dynamic Response Error Norm Analysis of the Offshore Steel Structure	135
5.18 DSA Error Norm of the Offshore Structure	
(With Respect to Design Variable One)	136
5.19 DSA Error Norm of the Offshore Structure	
(With Respect to Design Variable Two)	136
5.20 Eigenvalues and Eigenvalue Derivatives of the Offshore Structure	140

5.21	Algorithm Accuracy and Efficiency Analysis with the Six-Story Eight-Bay Three-Dimensional Frame .....	143
5.22	Eigenvalues and Eigenvalue Derivatives of the Six-Story Eight-Bay Three-Dimensional Frame .....	147
5.23	Parallel-Vector Computation Efficiency of the Six-Story Eight-Bay Three-Dimensional Frame (14 Modes, MAM) .....	148
5.24	Algorithm Accuracy and Efficiency Analysis with the Three-Dimensional CSI Design Model .....	152
5.25	Velocity and Acceleration Error Norms of the Three-Dimensional CSI Design Model .....	153
5.26	Eigenvalues of the CSI Design Model .....	158
5.27	Parallel-Vector Computation Efficiency with the CSI Design Model (40 Modes, MAM) .....	159

## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
4.1 Flow Chart of DSA in Structural Dynamics .....	76
4.2 Flow Chart of DSA in Structural Dynamics: Global and Local Parallelization .....	78
4.3 Storage Scheme and Block Dividing of the Symmetric, Banded Sparse Matrix .....	81
5.1 A Three-Element Cantilever Beam .....	88
5.2 Tip Deflections of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) .....	92
5.3 Displacement Error Norm of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) .....	93
5.4 Velocity Error Norm* of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) .....	94
5.5 Displacement Derivative Error Norm of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) .....	96
5.6 Stress Error Norm of the Three-Element Cantilever Beam ( $P = 10 \sin 100t$ ) .....	99

5.7	Stress Derivative Error Norm of the Three-Element Cantilever Beam	
	( $P = 10 \sin 100t$ )	100
5.8	A 200-Element Cantilever Beam	102
5.9	Displacement Error Norm of the 200-Element Cantilever Beam	104
5.10	Displacement Derivative Error Norm	
	of the 200-Element Cantilever Beam	105
5.11	A Simple Two-Dimensional Frame	108
5.12	Displacement Error Norm of the Simple Two-Dimensional Frame	111
5.13	Velocity Error Norm of the Simple Two-Dimensional Frame	112
5.14	Displacement Derivative Error Norm	
	of the Simple Two-Dimensional Frame	113
5.15	Stress Error Norm of the Simple Two-Dimensional Frame	115
5.16	Stress Derivative Error Norm of the Simple Two-Dimensional Frame	116
5.17	An 18-Bay 25-Story Two-Dimensional Frame	120
5.18a	Displacement Error Norm of	
	the 18-Bay 25-Story Two-Dimensional Frame	122
5.18b	Velocity and Acceleration Error Norm of	
	the 18-Bay 25-Story Two-Dimensional Frame	123
5.19	Displacement Derivative Error Norm of	
	the 18-Bay 25-Story Two-Dimensional Frame	124
5.20	Stress Error Norm of the 18-Bay 25-Story Two-Dimensional Frame	126

5.21	Stress Derivative Error Norm of the 18-Bay 25-Story Two-Dimensional Frame .....	127
5.22	CPU Time Comparison of MAM Versus MDM with the 18-Bay 25-Story Two-Dimensional Frame .....	129
5.23	A Flexible Steel Offshore Structure .....	133
5.24	A Simplified Model of the Flexible Offshore Structure .....	134
5.25	Displacement Error Norm of the Flexible Offshore Structure .....	137
5.26	Displacement Derivative Error Norm of the Flexible Offshore Structure (With Respect to Design Variable One) .....	138
5.27	Displacement Derivative Error Norm of the Flexible Offshore Structure (With Respect to Design Variable Two) .....	139
5.28	A Typical Three-Dimensional Frame .....	142
5.29	Displacement Error Norm of the Six-Story Eight-Bay Three-Dimensional Frame .....	144
5.30	Displacement Derivative Error Norm of the Six-Story Eight-Bay Three-Dimensional Frame .....	145
5.31	CPU Time Comparison of MAM Versus MDM for the Six-Story Eight-Bay Three-Dimensional Frame .....	146
5.32	CSI Design Finite Element Model .....	151
5.33	Displacement Error Norm of the CSI Design Model .....	154
5.34	Displacement Derivative Error Norm of the CSI Design Model .....	155
5.35	Velocity and Acceleration Error Norms of the CSI Design Model .....	156

5.36 CPU Time Comparison of MAM Versus MDM	
for the CSI Design Model .....	157



## LIST OF SYMBOLS

$C$	Original damping matrix
$K$	Original stiffness matrix
$M$	Original mass matrix
$Q(t)$	Nodal force vector for the original system
$\ddot{Z}$	Acceleration vector for the original system
$\dot{Z}$	Velocity vector for the original system
$Z$	Displacement vector for the original system
$\phi$	Eigenvector matrix
$\Lambda$	Diagonal matrix containing eigenvalues
$X_k$	Iteration vectors in the Subspace iteration method
$K_{k+1}$	Projection of the matrix of $K$ into $E_{k+1}$ space
$M_{k+1}$	Projection of the matrix of $M$ into $E_{k+1}$ space
$T_m$	Tridiagonal matrix in Lanczos algorithm
$\omega_i^2$	The $i^{\text{th}}$ eigenvalue, or, $\lambda_i$
$X$	Ritz vectors, or base vectors
$\ddot{u}$	Acceleration vector for the reduced system
$\dot{u}$	Velocity vector for the reduced system

$\mathbf{u}$	Displacement vector for the reduced system
$\bar{\mathbf{M}}$	Reduced system mass matrix
$\bar{\mathbf{C}}$	Reduced system damping matrix
$\bar{\mathbf{K}}$	Reduced system stiffness matrix
$\bar{\mathbf{Q}}(\mathbf{t})$	Reduced system loading vector
$m_i$	The $i^{\text{th}}$ mass element of the decoupled system
$c_i$	The $i^{\text{th}}$ damping element of the decoupled system
$k_i$	The $i^{\text{th}}$ stiffness element of the decoupled system
$p_i$	The $i^{\text{th}}$ loading element of the decoupled system
$\ddot{u}_i$	Acceleration for the $i^{\text{th}}$ equation in the decoupled system
$\dot{u}_i$	Velocity for the $i^{\text{th}}$ equation in the decoupled system
$u_i$	Displacement for the $i^{\text{th}}$ equation in the decoupled system
$\zeta_i$	Damping ratio
$\omega_i$	Frequency
$\omega_{Di}$	Damping frequency
$\ddot{\mathbf{Q}}(\mathbf{t}), \dot{\mathbf{Q}}(\mathbf{t})$	Derivatives of the loading function
$\varphi_0$	Objective function
$g_i$	Constraint functional
$\varphi_i$	Constraint functions
$\lambda(\mathbf{t})$	Adjoint variable vector in DSA
$\mathbf{b}_i$	Design variables

<b>E</b>	Young's modulus
<b>E</b>	Elastic Coefficient matrix
$\sigma$	Stress vector
$\epsilon$	Strain vector
<b>N</b>	Shape function matrix
<b>S</b>	Stress matrix
$\alpha, \beta$	Raleigh damping coefficients $C = \alpha M + K \beta$

## 1. INTRODUCTION

### 1.1 Overview

During the last twenty years, the rapid development of high speed digital computers, including parallel and vector computers, and of the finite element method have greatly increased the range and complexity of structural problems which can be solved. However, difficulties still remain in the computational solution of the large structural systems because of large storage requirements and long computational time. As a result, much research has been directed on the development of efficient techniques with structural analysis and design for large structural systems [1].

Research has indicated that computers with parallel and vector processing capability can perform computations much faster than the traditional scale sequential computers [2, 3, 4]. Because of this computational speed advantage, new algorithm and application software should incorporate computational methods that exploit such technology. In practice, it is very important to maintain reasonable computing cost for any analysis such that inexpensive reanalysis become possible. Thus, improvements in numerical techniques and developments of efficient parallel-vector algorithms, which significantly reduce the computational cost, are very useful.

The field of parallel-vector computing is relatively young. The first Cray 1, a pipelined vector processor, was delivered in 1976. Low-cost multiprocessors were not available before 1984. In the past ten years, parallel-vector computing has had a great impact on engineering computation. Many claim we are entering "the decade of parallel computer". Experts have pointed [5] that the emerging of supercomputers and parallel computers is going to cause a revolution in the scientific and engineering computations. And there is an urgent need to re-evaluate the existing sequential algorithms, to establish their suitability for parallel computers, and to develop numerical algorithms for parallel computers. It was also stated [ 5 ] that parallel computing is more than a different way of executing existing sequential programs, it offers a new challenge for numerical algorithm designers, applied mathematicians and computer scientists.

With regard to the topic of this study, sensitivity analysis is emerging as a fruitful area of engineering research. The reason for this interest is the recognition of the variety of uses for sensitivity derivatives, which range from automatic control theory to the analysis of large-scale psychological systems. Some of the areas where sensitivity analysis has been applied include: 1) system identification, 2) development of insensitive control systems, 3) use in gradient-based optimization algorithms, 4) approximation of system response to a change in a system parameter, and 5) assessment of design changes on system performance. The design sensitivity analysis ( DSA ) of this study is for the purpose of providing constraint derivative information for structural optimization under dynamic loads.

In the past twenty years, researchers have devoted much efforts in DSA of structural systems. The DSA for static structural systems have become considerable maturity, and a large amount of work has been carried out in structural optimization and DSA in static systems [6]. However, algorithms and theory of optimization and DSA in structural dynamics are still in the development stage.

It is well known that, in many practical problems, dynamic loading cases are more critical than the static ones. The time dependent loadings, which produce time dependent responses, increase the complexity and the level of computing expense for both the dynamic response analysis and DSA: instead of having a set of coupled algebraic equations in static system, a set of coupled second order differentiation equations occur, where accurate solutions are far more difficult to obtain and the solution procedure must be carried for various time steps, within the load duration, in an iterative manner.

Some of the important previous work in the area of optimal design and DSA in structural dynamics is briefly reviewed and discussed in Section 1.2.

## **1.2 Review of Previous Pertinent Work**

The DSA in structural dynamics has attracted considerable interest since the middle of 1970's, when the structural dynamic optimization emerged [7, 8, 9]. The first paper in structural optimal design under dynamic responses was published in 1970 by Fox and Kapoor [8]. In 1972, Pierson [7] conducted a survey of optimal

structural design under dynamic constraints. Feng, Arora, and Haug [10] proposed an algorithm for optimal design subject to dynamic loads in 1977, where, finite element, modal analysis and a generalized steepest descent method are employed, and point-wise dynamic constraints are treated as equivalent integral constraints.

Hsieh and Arora [11] presented a worst-case design procedure, in which the constraints are imposed at all local maximum responses, both direct differentiation and adjoint variable formulation are discussed. Structural DSA with general boundary conditions are presented by Hsieh and Arora for static as well as dynamic problems [12, 13]. A hybrid formulation for treatment of the dynamic constraints were proposed by Hsieh and Arora [14]. Sensitivity analysis of discrete structural system is reviewed by Adelman and Haftka [6], where both static and dynamic problems are discussed, variety of procedures of obtaining the DSA information, such as, the analytical method, adjoint variable method, Green's function method, finite difference method, and the FAST ( Fourier Amplitude Sensitivity Test ) method are presented. An active set RQP algorithm was presented for dynamic response optimization by Lim and Arora [15]. Sequential quadratic programming was applied to optimal design for dynamic systems by Tseng and Arora [16]. Shape design sensitivity of dynamic structures were presented by Meric [17]. Developments by Arora and Haug [18, 19] , Mroz, Haftka [20, 21], and Haug, Choi, KomKov [22] provided the mathematical foundation of structural DSA. Linearization method [23] was applied to dynamic system optimization in 1983. In some finite element codes a version of structural design sensitivity has been incorporated. For example, the

design sensitivity and optimization algorithm were developed using ADINA [ 24 ] for physically and geometrically non-linear structures. However, only static problems were treated. Ritz sequence is applied to continuum DSA of structural dynamics response by Choi and Wang [25]. Recent developments in DSA is reviewed by Haftka and Adelman [26].

Recently, William H. Greene [27] carried out a study of DSA in linear transient structural analysis. In his study, both forward finite difference and central finite difference methods are used to calculate the sensitivity information. System reduction techniques are applied, where mode displacement method, mode acceleration method, and the Ritz-Wilson-Lanczos method are used. Eigensystem solutions are obtained by subspace iteration method. In his work, eigenvector (or, base vector) derivatives are needed for DSA, which is very expensive and complicated. Greene proposed a fixed-mode semi-analytical formulation, where it is assumed that the base vectors are not a function of the design variables. The fixed-mode semi-analytical formulation significantly simplified the calculation, however, it suffers from the accuracy problem.

Little work for development of effective parallel-vector algorithm has been carried out for the sensitivity analysis as indicated in Nguyen and Niu's paper [28]. To the author's best knowledge, no information in the literature could be found on the proposed parallel-vector DSA in structural dynamics.



### 1.3 Objective and Scope

The major objective of this dissertation research is to develop an effective parallel-vector algorithm for DSA in large-scale structural dynamics. This dissertation research focusses at two levels. The first level is the primary focus on innovation in algorithms for the DSA in structural dynamics; The second level lies in the development of an efficient parallel-vector algorithm to deal with structure of sufficiently large-scale. The parallel-vector algorithm developed is implemented in Cray 2 (NASA, Langley) and Cray Y-MP (NASA, Ames) supercomputers using a parallel-Fortran language Force [29]. The accuracy of the algorithm is verified through numerical examples. The effectiveness and efficiency of the parallel-vector algorithm will be demonstrated through analysis of large-scale structural systems.

In the this research, linear dynamic structure systems are studied with finite element formulation [30, 31]. In large-scale structural design, a large number of degree-of-freedom (DOF) must be considered, to accurately describe the response of a complex structure to dynamic loads. This results in large number of equations of motion for the system. Therefore, system reduction techniques [30, 32, 33] are commonly used.

When applying the system reduction techniques, various base vectors, such as, eigenvectors [ 30 ], Ritz vectors [34] , Lanczos vectors [35], could be used. The Ritz vectors perform very well in structural dynamic response analysis when the loading can be separated as the function of space and time, and are also applied [25] to DSA

in structural dynamics. However, based on the author's study, it is understood that the Ritz vectors are load dependent vectors, due to the complete difference between the dynamic loading and the pseudo-loading term in the right hand side of the design sensitivity equation (see Chapter 3), the application of Ritz vectors to DSA in structural dynamics may lead to the inaccuracy problem. Even though the Ritz vectors cost less as compared to the eigenvectors obtained by the Lanczos algorithm, the eigenvectors are believed to be more accurate for DSA in structural dynamics. The most popular eigen-problem solving algorithms for large-scale structures are the subspace iteration method [30] and Lanczos method [3, 35]. The Lanczos algorithm is used in this study due to its efficiency. Since eigenvectors are used as base vectors for the reduction of the number of DOF in both structural dynamic response analysis and DSA, the reduced system equations are decoupled. The Duhamel integral formula [36] is the most suitable one to solve the decoupled system. As an alternative, the implicit integration techniques [37, 38, 39] could also be applied. Among those implicit integration methods, the Newmark method is the most promising candidate. The modal displacement method (MDM) [36], modal acceleration method (MAM) [36, 40], force derivation method (FDM) [40, 41, 42] are reviewed. A modified modal acceleration method ( MMAM ) is presented, which could improve not only the displacements, as the MAM does, also improves the velocities and the accelerations. However, due to the difficulty of applying MMAM to DSA, the MMAM is not recommended for DSA. Therefore, MAM is used for the solution of both structural dynamic responses and the DSA.

The DSA is bounded to elastic structures with fixed geometry. The design variables describe the cross-sectional properties, such as, cross-sectional area, moment of inertia, member thickness, element size, etc. The state variables arise from the dynamic analysis, which are time independent and include those such as, displacement, velocity, stress, etc., of the analyzed structure.

There are two different strategies [18, 20] for sensitivity analysis: the direct differentiation method (DDM) and adjoint variable methods (AVM). The selection of a DSA method for iterative optimal design, particularly for large-scale structures in the parallel-vector computation environment, is also very critical. The direct differentiation method is more suitable for developing parallel codes, since each set of sensitivity equations corresponding to a design variable is independent. In addition, it is believed that the DDM has its advantages over the AVM for DSA in structural dynamics.

The DSA could be carried out with either the original full system or the reduced system. It is obvious that, for large-scale structural systems, the DSA based on a reduced system is more computational advantage than the one based on the original structural system. However, the existing reduced system formulation requires eigenvector derivatives, which are very expensive and complicated, since system of equations with singularity has to be dealt with. Among the methods of eigenvector derivatives [42, 43, 44, 45, 46], which include the modal method, modified modal method and the Nelson's method. Nelson's method is the most promising algorithm

according to investigation of this study. Further, a parallel-vector Nelson's algorithm is developed in this study.

As mentioned above, the existing reduced system formulation leads to an efficient procedure of DSA in structural dynamics. However, it involves the eigenvector derivatives, which are complicated and very expensive in computation. It is desirable to develop a better approach. Here, an alternative formulation for DSA is developed in which the eigenvector derivatives are avoided. The relationship between the DDM DSA based on reduced system and the alternative formulation is originally established in this work. The equivalency of the two approaches could be analytically proved when the transformation from the original full system to the reduced system is exact, that is, if the number of eigenvectors used is equal to the size of the original full system. The alternative formulation has several advantages: 1) it works with reduced system, 2) it is efficient and suitable for parallel computing, 3) it is not only simple, it is also accurate, 4) it is valid for both linear and nonlinear cases provided the system reduction technique is applicable; it can be applied to FEM formulation as well as distributed parameter structural dynamic DSA.

The entire algorithm, from eigensystem solution, reduction of the structural dynamic system through dynamic response calculation to sensitivity analysis, is parallel-vectorized to an efficient one. To fulfill this objective, various standard techniques of achieving parallelism are used, which provides both global parallelization and local parallel-vectorization for the developed algorithm. Related matrix algebra algorithms are modified, since none of the existing sequential

algorithms could be adopted directly without modification, and some of them have been rewritten completely. Numerical examples are provided to demonstrate the accuracy of the algorithm developed. The efficiency is illustrated by analyzing large-scale structures.

## 2. LARGE-SCALE STRUCTURAL DYNAMIC ANALYSIS

### 2.1 Introduction

In structural analysis and design, the dynamic responses are often more critical than the static responses. The dynamic equilibrium equation from the finite element formulation [30] could be presented as:

$$\mathbf{M} \ddot{\mathbf{Z}} + \mathbf{C} \dot{\mathbf{Z}} + \mathbf{K} \mathbf{Z} = \mathbf{Q}(t) \quad (2.1)$$

where,  $\mathbf{M}$ ,  $\mathbf{C}$ , and  $\mathbf{K}$  are the mass matrix, damping matrix, and stiffness matrix respectively, each of them has the dimension of  $n \times n$ .  $\ddot{\mathbf{Z}}$ ,  $\dot{\mathbf{Z}}$ , and  $\mathbf{Z}$  represent the acceleration, velocity, and displacement vectors,  $\mathbf{Q}(t)$  is the nodal load vector.

Eq. (2.1) could be solved for the dynamic responses,  $\ddot{\mathbf{Z}}$ ,  $\dot{\mathbf{Z}}$ , and  $\mathbf{Z}$ , either by direct integration method, which include the central difference method, Houbolt method, Wilson method, and Newmark method, or by indirect method, such as the mode superposition method [30]. It is well known that system reduction techniques are extensively used in large-scale linear structural dynamics, which yield efficient solutions with desirable accuracy provided sufficient base vectors are applied.

## 2.2 Eigensystem Solution in Structural Dynamics

The solution of an eigensystem is very important in structural dynamics and vibration problems, which could provide the condition number of the stiffness matrix, give the fundamental frequencies for design needs. Particularly, in this study, the eigensystem solution provides frequency constraints and an eigenvector matrix which meets the needs of the system reduction in both dynamic response analysis and DSA. Besides, the solution of an eigensystem is also a very expensive task. A lot of research have been devoting in this research area.

The generalized eigen-problem could be presented as

$$\mathbf{K} \phi = \Lambda \mathbf{M} \phi \quad (2.2)$$

where,  $\mathbf{K}$  and  $\mathbf{M}$  is the stiffness and mass matrices.  $\Lambda$  is a diagonal matrix which contains the eigenvalues, and  $\phi$  is the eigenvector matrix.

A variety of eigensystem solution methods have been developed and well documented in the literature [47]. The methods can be mainly classified as three categories: transformation method, determinant search methods and iterative methods.

The transformation methods include those of Jacobi method, Householder's transformation, and QR transformation. In general, these methods are applied when the matrices involved are comparatively small in size and somewhat fully populated, or when they have a large bandwidth. Those methods find all the eigenvalues of the

eigensystem, while the eigenvectors could be obtained by a process of inverse transformation.

The determinant search method is suitable when the eigensystem has a very small bandwidth, and when only a few eigenvalues are required. The method is basically to solve the eigenvalue characteristic polynomial equations.

The iteration methods are very effective for the solution of large eigensystems with sparse and banded matrices involved, particularly in the situation of fewer eigenpairs are sought. These methods have been commonly applied in practice for large-scale structural dynamics and vibration problems. Vector iteration method, subspace method, and Lanczos method are in this group. The most widely applied ones are the subspace iteration method and the Lanczos method, which are further discussed in the coming sections.

### **2.2.1 The Subspace Iteration Method**

The subspace iteration method of the generalized eigen-problem was originally proposed by K. J. Bathe [30]. To find the lowest  $p$  eigenvalues and their corresponding eigenvectors, the basic subspace iteration method consists of the computational steps of:

**Step 1.** Establish  $q$  starting iteration vectors, which span the starting subspace  $E_1$ . The number of the starting vectors,  $q$ , should be greater than the numbers of the



eigenvalues,  $m$ , required. In practice, the selection of  $q$  is recommended as  $q = \min\{ 2p, p + 8 \}$ .

**Step 2.** Perform subspace iterations. The simultaneous inverse iteration is used on the  $q$  vectors, and Ritz analysis is employed to extract optimum eigenvalue and eigenvector approximations at the end of each inverse iteration.

a). For  $k = 1, 2, \dots$ , iterate from subspace  $E_k$  to subspace  $E_{k+1}$ :

$$K \bar{X}_{k+1} = M X_k \quad (2.3)$$

b). Calculate the projections of the matrices  $K$  and  $M$  onto  $E_{k+1}$ :

$$K_{k+1} = \bar{X}_{k+1}^T K X_{k+1} \quad (2.4)$$

$$M_{k+1} = \bar{X}_{k+1}^T M X_{k+1} \quad (2.5)$$

c). Solve for the eigensystem of the projected matrices:

$$K_{k+1} Q_{k+1} = M_{k+1} Q_{k+1} \Lambda_{k+1} \quad (2.6)$$

d) Calculate an improved approximation to the eigenvectors:

$$X_{k+1} = \bar{X}_{k+1} Q_{k+1} \quad (2.7)$$

**Step 3.** Convergence check and Sturm sequence check. If not converged, go to step 2. After iteration convergence, use the Sturm sequence check to verify that the required eigenvalues and corresponding eigenvectors have been obtained.

### 2.2.2 The Lanczos Algorithm

Lanczos algorithm for the solution of generalized eigen-problem has been receiving considerable attention in recent years due to its computational efficiency [3,35]. The Lanczos method can be considered as a means of constructing an orthogonal set of Lanczos vectors, which is Krylov sequence with the Gram-schmidt orthogonalization at each step for use in the Rayleigh Ritz procedure. The Rayleigh Ritz procedure with M-orthonormal basis of the Lanczos vectors leads to a standard eigen-problem of a tridiagonal matrix,  $T_m$

$$T_m = \begin{bmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \beta_3 & \alpha_3 & \cdot & & \\ & & & \cdot & \cdot & \cdot \\ & & & & \cdot & \cdot & \beta_m \\ & & & & & \beta_m & \alpha_m \end{bmatrix} \quad (2.8)$$

through the following three-term recurrence:

$$r_j = \beta_{j+1} q_{j+1} = K^{-1} M q_j - \alpha_j q_j - \beta_j q_{j-1} \quad (2.9)$$

or, in the matrix form:

$$[K^{-1} M] Q_m - Q_m T_m = \{ 0, 0, \dots, T_m \} = r_m e_m^T \quad (2.10)$$

where  $e_m^T = (0, 0, 0, \dots, 1)$ ,  $Q_m$  is a  $n \times m$  orthogonal matrix with columns  $q_i$ ,  $i = 1, 2, 3, \dots, m$ . By solving the following reduced eigensystem

$$\mathbf{T}_m \mathbf{Z} = \theta \mathbf{Z} \quad (2.11)$$

the eigensolution of Eq. ( 2.2 ) can be obtained as

$$\omega_2 = \frac{1}{\theta} \quad (2.12)$$

$$\mathbf{X} = \mathbf{Q}_m \mathbf{Z} \quad (2.13)$$

For most structural engineering problems, only a few lowest frequencies and the corresponding eigenvectors are sought ( i.e.,  $m \ll n$  ), which leads to a significant savings in number of operations. The step-by-step computational Lanczos algorithm is given as follows:

1. Factorization stiffness matrix  $\mathbf{K} = \mathbf{L} \mathbf{L}^T$ , and form the starting vector:

$$\gamma_0 \neq 0; \quad \mathbf{q}_0 = \mathbf{0} \quad (2.14)$$

2. Compute

$$\mathbf{M} \gamma_0 \quad (2.15)$$

3. Compute

$$\beta_1 = \sqrt{\gamma_0^T \mathbf{M} \gamma_0}; \quad \mathbf{q}_1 = \frac{\gamma_0}{\beta_1} \quad (2.16)$$

4. Compute

$$\mathbf{P}_1 = \mathbf{M} \mathbf{q}_1 \quad (2.17)$$

5. Applying Lanczos iteration: For  $j = 1, 2, 3, \dots$ , Do

$$\text{a. } \bar{\gamma}_j = \mathbf{K}^{-1} \mathbf{P}_j \quad (2.18)$$

$$\text{b. } \gamma_j = \bar{\gamma}_j - \beta_j \mathbf{q}_{j-1} \quad (2.19)$$

$$\text{c. } \alpha_j = \mathbf{q}_j^T \mathbf{M} \gamma_j = \bar{\mathbf{P}}_j \gamma_j \quad (2.20)$$

$$\text{d. } \gamma_j = \gamma_j - \alpha_j \mathbf{q}_j \quad (2.21)$$

$$\text{e. } \bar{\mathbf{P}}_j = \mathbf{M} \gamma_j \quad (2.22)$$

$$\text{f. } \beta_{j+1} = \sqrt{\gamma_j^T \mathbf{M} \gamma_j} = \sqrt{\bar{\mathbf{P}}_j^T \gamma_j} \quad (2.23)$$

Reorthogonalization of  $\mathbf{q}_{j+1}$

$$\text{g. } \mathbf{q}_{j+1} = \frac{\gamma_j}{\beta_{j+1}}; \quad \mathbf{P}_{j+1} = \frac{\bar{\mathbf{P}}_j}{\beta_{j+1}} \quad (2.24)$$

6. If necessary solve  $\mathbf{T}_j \mathbf{Z} = \theta \mathbf{Z}$ , converged? (if no, go to step 5)

7. Eigenvector transformation  $\mathbf{X} = \mathbf{Q}_j \mathbf{Z}$

### **2.2.3 The Justification of the Methods**

Research about the justification of the two approaches of obtaining the eigensystem solution has been carried out in the literature, and some of the results are quoted below.

The first study [3] provides for an example to find the natural frequencies and mode shapes of a high-speed research aircraft. The finite element model has 1646 DOF, and a half-bandwidth of 323. For finding 20 eigen-pairs with 62 iteration steps, the CPU time in the Cray Y-MP computer for Lanczos method is 37.8 seconds while the subspace method takes 63.18 seconds. And some more examples are provided in reference [3].

Also, in other studies [48], it is pointed out that a good Lanczos algorithm is an order of magnitude faster and therefore less costly than basic subspace iteration in both the number of required matrix-vector operations and CPU time.

Therefore, the Lanczos algorithm is applied in this study for its efficiency.

### **2.3 The System Reduction Techniques**

As mentioned before, the dynamic analysis of a large-scale structural system is a very expensive and time consuming task. This is true because a huge number of DOF must be considered to accurately present the behavior of a structure under dynamic loads. A very large number of equations of motion for the system analyzed

results, and both the solution of large system of equations and the eigen-problem solutions are required. Besides, the solution is an iterative procedure for the time history of loading. It is inefficient to directly use time step integration techniques [49] to solve such a large system. Therefore, system reduction techniques are usually applied.

The base vectors used for the system reduction techniques could be eigenvectors, Ritz vectors, Lanczos vectors, etc.; those will be discussed in the coming section.

### **2.3.1 The Various Base Vectors**

For the reduction of the system equations in structural dynamics, various vectors, such as eigenvectors, Ritz vectors and Lanczos vectors could be applied.

The eigenvectors represent the mode shape of the structure. Procedures of obtaining the eigenvectors has been discussed in previous section (see section 2.2).

The key idea of the Ritz vector method [34, 50] is to select an orthonormalized Ritz basis in Krylov space that depends on the spatial distribution of the load. The advantages of this method is that no iteration is involved. The Ritz vector reduction method is declared as the one which has better efficiency and yields results of comparable accuracy or even better accuracy than those obtained from exact eigenvectors [34,50]. The disadvantage of the Ritz vectors is load dependency, which could cause inaccuracy under complicated load cases, especially when the

separation of spatial and temporal functions of the loading become difficult or impossible. Although the block Ritz vector method [50] were suggested to overcome this disadvantage, the effectiveness of the method needs further investigation. The procedure of obtaining the  $m^{\text{th}}$  lowest Ritz vectors could be presented as follows:

**Step 1. Factorization of the stiffness matrix**

$$\mathbf{K} = \mathbf{L} \mathbf{L}^T \quad (2.25)$$

**Step 2. Solve for the first Ritz-vector  $\mathbf{X}_1$**

a). solve for  $\mathbf{X}_1'$

$$\mathbf{K} \mathbf{X}_1' = \mathbf{f} \quad (2.26)$$

b). Normalize solution with respect to mass matrix

$$\mathbf{X}_1 = \frac{\mathbf{X}_1'}{\sqrt{\mathbf{X}_1'^T \mathbf{M} \mathbf{X}_1'}} \quad (2.27)$$

**Step 3. Solve for additional Ritz vectors ( $i = 2, 3, \dots, 2m$ )**

a). Solve for  $\mathbf{X}_i'$

$$\mathbf{K} \mathbf{X}_i' = \mathbf{M} \mathbf{X}_{i-1} \quad (2.28)$$

b). compute for  $j = 1, 2, \dots, i-1$

$$\mathbf{C}_j = \mathbf{X}_j^T \mathbf{M} \mathbf{X}_i' \quad (2.29)$$

c). make new vector orthogonal to previous Ritz vectors by

$$\mathbf{X}_i'' = \mathbf{X}_i' - \sum_{j=1}^{i-1} c_j \mathbf{X}_j \quad (2.30)$$

d). normalize vector with respect to mass matrix

$$\mathbf{X}_i = \frac{\mathbf{X}_i''}{\sqrt{\mathbf{X}_i''^T \mathbf{M} \mathbf{X}_i''}} \quad (2.31)$$

**Step 4.** (Optional) Orthogonalize Ritz vectors  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{2m})$  with respect to stiffness matrix

$$\mathbf{K}' = \mathbf{X}^T \mathbf{K} \mathbf{X} \quad (2.32)$$

$$\mathbf{M}' = \mathbf{X}^T \mathbf{M} \mathbf{X} = \mathbf{I} \quad (2.33)$$

Solve

$$(\mathbf{K}' - \omega_i^2 \mathbf{M}') \mathbf{Z}_i = 0 \quad (2.34)$$

The final Ritz vectors

$$\mathbf{X}^\circ = \mathbf{X} \mathbf{Z} \quad (2.35)$$



The Lanczos vectors are similar to the Ritz vectors. The generating of the Lanczos vectors is based on the Lanczos algorithm described in Section 2.2.2, detailed information is documented in reference [3, 35, 50].

### 2.3.2 The Mode Displacement Method

The mathematical understanding of system reduction methods, which are all base vector superposition methods, is recognized as a change of basis to a computationally more effective system of equations through the base vector matrix  $\phi$ . This base vector matrix  $\phi$  transforms the n-nodal point displacements to m-nodal generalized displacements, where  $m \ll n$ , prior to the applying of the step-by-step implicit integration. The method could be represented as follows:

First, we introduce a new variable  $u$  through the transformation of

$$Z = \phi u \quad (2.36)$$

with  $m \ll n$ .

Substituting Eq. (2.36) into Eq. (2.1) yields

$$M \phi \ddot{u} + C \phi \dot{u} + K \phi u = Q(t) \quad (2.37)$$

premultiplying both sides of the above equation by  $\phi^T$  gives

$$\phi^T M \phi \ddot{u} + \phi^T C \phi \dot{u} + \phi^T K \phi u = \phi^T Q(t) \quad (2.38)$$

which is then rewritten as

$$\bar{M} \ddot{u} + \bar{C} \dot{u} + \bar{K} u = \bar{Q}(t) \quad (2.39)$$

with the definition of

$$\bar{M} = \phi^T M \phi \quad (2.40)$$

$$\bar{C} = \phi^T C \phi \quad (2.41)$$

$$\bar{K} = \phi^T K \phi \quad (2.42)$$

$$\bar{Q} = \phi^T Q \quad (2.43)$$

Once the response of the reduced system is obtained, the response of the original system could be easily calculated by using the transformation Eq. (2.36).

If the transformation matrix  $\phi$  is an eigenvector matrix associated with the undamped free vibration problem of

$$M \ddot{Z} + K Z = 0 \quad (2.44)$$

Provided the eigenvectors are M-orthonormalized, we then have

$$\phi^T M \phi = \Lambda \quad (2.45)$$

$$\phi^T K \phi = I \quad (2.46)$$

where,  $\Lambda$  presents a diagonal matrix with the eigenvalues for diagonal elements, and  $I$  is a identity matrix. Thus, we have a decoupled system, then the reduced system of Eq. (2.39) could be rewritten as

$$m_i \ddot{u}_i + c_i \dot{u}_i + k_i u_i = p_i(t) \quad (2.47)$$

or,

$$\ddot{u}_i + 2 \zeta_i \omega_i \dot{u}_i + \omega_i^2 u_i = \frac{p_i}{m_i} \quad (2.48)$$

where,

$$\omega_i^2 = \frac{k_i}{m_i} \quad (2.49)$$

and,

$$2\zeta_i \omega_i = \frac{c_i}{m_i} \quad (i = 1,2,\dots,m) \quad (2.50)$$

The solution of Eq. (2.48) could be obtained by the Duhamel integral as

$$\begin{aligned}
u_i = & e^{-\zeta_i \omega_i t} \left( \frac{\dot{u}_i(0) + u_i(0) \zeta_i \omega_i}{\omega_{Di}} \sin \omega_{Di} t + u_i(0) \cos \omega_{Di} t \right) \\
& + \frac{1}{m_i \omega_{Di}} \int_0^t p_i(\tau) e^{-\zeta_i \omega_i (t-\tau)} \sin \omega_{Di} (t-\tau) d\tau
\end{aligned} \tag{2.51}$$

$$\begin{aligned}
\dot{u}_i = & -\zeta_i \omega_i u_i + e^{\zeta_i \omega_i t} [(\dot{u}_i(0) + u_i(0) \zeta_i \omega_i) \cos \omega_{Di} t \\
& - \omega_{Di} u_i(0) \sin \omega_{Di} t] \\
& + \frac{1}{m_i \omega_{Di}} \int_0^t [\omega_{Di} p_i(\tau) e^{-\omega_i \zeta_i (t-\tau)} \cos \omega_{Di}(t-\tau)] d\tau
\end{aligned} \tag{2.52}$$

$$\ddot{u}_i = \frac{1}{m_i} p_i(t) - 2 \zeta_i \omega_i \dot{u}_i - \omega_i^2 u_i \tag{2.53}$$

where

$$\omega_{Di} = \omega_i \sqrt{1 - \zeta_i^2} \tag{2.54}$$

Finally, the dynamic response of the original system could be obtained by

$$Z = \phi u \tag{2.55}$$

$$\dot{Z} = \phi \dot{u} \tag{2.56}$$

and,

$$\ddot{\mathbf{Z}} = \boldsymbol{\phi} \ddot{\mathbf{u}} \quad (2.57)$$

### 2.3.3 The Mode Acceleration Method

The mode displacement method may fail to give an accurate solution, even when static load is applied. Frequently, the convergence is slow and many modes would be needed to give an accurate mode displacement solution (see Chapter 5). This difficulty can be alleviated by using of the mode acceleration method. Because of the improved convergence properties of this method, fewer natural frequencies and modes are required from the eigen-solution. Obviously, the method itself requires more computational efforts than the mode displacement method. However, as shown latter in Chapter 5, this can be compensated, or, even over compensated by the less computation in the eigensolution, since obtaining of eigensolution is very expensive.

Let's recall Eq. (2.48)

$$\ddot{u}_i + 2 \zeta_i \omega_i \dot{u}_i + \omega_i^2 u_i = \frac{p_i}{m_i} = \frac{\phi_i^T Q}{m_i} \quad (2.58)$$

The static solution of the above problem is

$$u_i = \frac{\phi_i^T Q}{k_i} \quad (2.59)$$

or

$$u_i = \frac{\phi_i^T Q}{m_i \omega_i^2} \quad (2.60)$$

It is clear that if the spatial distribution of the forcing function is such that the higher modes are significantly excited, such modes must be included in the analysis. Meanwhile, if the higher-mode frequency is much larger than the highest frequency content of the applied loading, then the response in the higher mode is essentially static. Thus, the total response could be approximated by the addition of the dynamic response of the first  $m$  modes and the static response of the remaining  $m+1$  to  $n$  modes, i.e.,

$$Z = \sum_{i=1}^m \phi_i u_i + \sum_{i=m+1}^n \phi_i \frac{\phi_i^T Q}{k_i} \quad (2.61)$$

making use of Eq. (2.58), which could be rewritten as

$$Z = \sum_{i=1}^m \phi_i \left( \frac{\phi_i^T Q}{\omega_i^2 m_i} - \frac{\ddot{u}_i}{\omega_i^2} - \frac{2 \zeta_i \dot{u}_i}{\omega_i} \right) + \sum_{i=m+1}^n \phi_i \frac{\phi_i^T Q}{k_i} \quad (2.62)$$

$$Z = \sum_{i=1}^n \phi_i \frac{\phi_i^T Q}{k_i} - \sum_{i=1}^m \left( \phi_i \frac{\ddot{u}_i}{\omega_i^2} + \phi_i \frac{2 \zeta_i \dot{u}_i}{\omega_i} \right) \quad (2.63)$$

since

$$\sum_{i=1}^n \phi_i \frac{\phi_i^T Q}{m_i \omega_i^2} = \phi \Lambda^{-1} \phi^T Q \quad (2.64)$$

so,

$$Z = K^{-1} Q - \sum_{i=1}^m \left( \phi_i \frac{\ddot{u}_i}{\omega_i^2} + \phi_i \frac{2 \zeta_i \dot{u}_i}{\omega_i} \right) \quad (2.65)$$

or,

$$Z = K^{-1} Q - \phi (\Omega^{-2} \ddot{u} + \Omega^{-2} \bar{C} \dot{u}) \quad (2.66)$$

were,  $\Lambda$  is a diagonal matrix containing the eigenvalues.

Because the above equation involves the superposition of modal acceleration, the method is often referred to as the mode acceleration method. Numerical studies show that the MAM could significantly improve the accuracy of the displacement in some cases, which is to be presented in Chapter 5.

### 2.3.4 The Modified Mode Acceleration Method

In the mode acceleration method only displacements are modified. This section presents the expressions to modify also the velocities and accelerations. The motivation is an attempt to improve the results of DSA in structural dynamics through the improvements of velocities and accelerations also.

The Eq. (2.61) could be rewritten as

$$\mathbf{Z} = \sum_{i=1}^n \phi_i \frac{\phi_i^T \mathbf{Q}}{k_i} + \sum_{i=1}^m (\phi_i \mathbf{u}_i - \phi_i \frac{\phi_i^T \mathbf{Q}}{k_i}) \quad (2.67)$$

Thus,

$$\dot{\mathbf{Z}} = \sum_{i=1}^n \phi_i \frac{\phi_i^T \dot{\mathbf{Q}}}{k_i} + \sum_{i=1}^m \left( \phi_i \dot{\mathbf{u}}_i - \phi_i \frac{\phi_i^T \dot{\mathbf{Q}}}{k_i} \right) \quad (2.68)$$

$$\ddot{\mathbf{Z}} = \sum_{i=1}^n \phi_i \frac{\phi_i^T \ddot{\mathbf{Q}}}{k_i} + \sum_{i=1}^m \left( \phi_i \ddot{\mathbf{u}}_i - \phi_i \frac{\phi_i^T \ddot{\mathbf{Q}}}{k_i} \right) \quad (2.69)$$

Numerical studies (see Chapter 5) show that the proposed Eq. (2.68) could significantly improve the accuracy of velocity. However, Eq. (2.69) does not behavior so well, due to the loss of significance by a higher derivative. Fortunately, from numerical experiments we found that in MDM/MAM the velocity and the acceleration vectors are almost in the same accuracy. Thus by using the information of the improved velocity of MMAM, we then could improve the acceleration in MMAM. Numerical examples are shown in Chapter 5.

### 2.3.5 The Force Derivation Method

In addition to the MDM and MAM, there is the force derivation method (FDM). The FDM was originally proposed by Leung [40], and it was investigated that the MAM could be derived by integrating by parts with respect to time the integral form of the MDM. Then, integrating the integral formula two more times



yields a higher order model method than the MAM. The higher model-method is called the FDM, due to the formulation consists of a term of the forcing function as well as a term of time-derivatives of the forcing function, which provides successively higher-order approximations to the higher and neglected modes.

The detailed derivation and description of the method is given in references [40, 41]. It is declared that, for some cases, the FDM could provide more accurate solution, or could reduce the number of modes used.

### **2.3.5 The Justification of the Methods**

The MAM has been shown to be very effective in structural dynamics. It is also simple to apply, since it essentially superimposes the static and mode displacement solutions. The FDM requires much more computational effort, approximately  $24 \times B \times N \times N$  more than the MAM (with  $B$  = half-bandwidth,  $N$  = number of DOF). The MMAM presented could improve not only displacements, but also velocities and accelerations. However, MMAM could not be applied to DSA, like MAM is, due to the difficulty of obtaining the derivative for the pseudo-load for DSA. Because of this inconsistency, the DSA information could not be significantly improved. Therefore, MAM is select for the this study. Some numerical experiments have been conducted to illustrate the MMAM, which are presented in Chapter 5.

## 2.4 Direct Integration Schemes

The direct integration techniques could mainly be classified as explicit integration schemes [49], implicit integration schemes [39], and the mixed methods [37].

The most commonly used explicit integration scheme is the central difference method, which is a conditionally stable method. The condition of stability requires that the step size of time be less than  $T/\pi$ , where  $T$  is the period of the highest mode.

The implicit integration methods are similar to the explicit ones. The major difference is that the equation of motion is formed at next time point instead of the current one. Houbolt's method, Wilson- $\theta$  method, and Newmark method are typical examples.

A step-by-step solution procedure [ 30 ] using Newmark method is presented as follows:

A. Initial Calculations:

1. Form stiffness matrix  $K$ , mass matrix  $M$ , and damping matrix  $C$ .
2. Initialize  $Z_0$ ,  $\dot{Z}_0$ , and  $\ddot{Z}_0$ .
3. Select time step size  $\Delta t$ , parameters  $\alpha$  and  $\delta$ , and calculate integration constants:

$$\delta \geq 0.50; \alpha \geq 0.25(0.5 + \delta)^2$$

$$a_0 = 1/(\alpha \Delta t^2); a_1 = \delta/(\alpha \Delta t); a_2 = 1/(\alpha \Delta t); a_3 = 1/(2\alpha) - 1;$$

$$a_4 = \delta / \alpha - 1; a_5 = 0.5 \Delta t (\delta / \alpha - 2); a_6 = \Delta t (1 - \delta); a_7 = \delta \Delta t;$$

$$4. \text{ Form effective stiffness matrix } \hat{\mathbf{K}}: \hat{\mathbf{K}} = \mathbf{K} + a_0 \mathbf{M} + a_1 \mathbf{C}.$$

$$5. \text{ Factorize } \hat{\mathbf{K}}: \hat{\mathbf{K}} = \mathbf{LDL}^T$$

B. For each time step do:

1. Calculate effective loads at time  $t + \Delta t$ :

$$\begin{aligned} \hat{\mathbf{R}}_{t+\Delta t} = & \mathbf{R}_{t+\Delta t} + \mathbf{M} (a_0 \mathbf{Z}_t + a_2 \dot{\mathbf{Z}}_t + a_3 \ddot{\mathbf{Z}}_t) \\ & + \mathbf{C} (a_1 \mathbf{Z}_t + a_4 \dot{\mathbf{Z}}_t + a_5 \ddot{\mathbf{Z}}_t) \end{aligned}$$

2. Solve for displacements at time  $t + \Delta t$ :

$$\mathbf{LDL}^T \mathbf{Z}_{t+\Delta t} = \hat{\mathbf{R}}_{t+\Delta t}$$

3. Calculate accelerations and velocities at time  $t + \Delta t$ :

$$\ddot{\mathbf{Z}}_{t+\Delta t} = a_0 (\mathbf{Z}_{t+\Delta t} - \mathbf{Z}_t) - a_2 \dot{\mathbf{Z}}_t - a_3 \ddot{\mathbf{Z}}_t$$

$$\dot{\mathbf{Z}}_{t+\Delta t} = \dot{\mathbf{Z}}_t + a_6 \ddot{\mathbf{Z}}_t - a_7 \ddot{\mathbf{Z}}_{t+\Delta t}$$

### 3. DESIGN SENSITIVITY ANALYSIS IN STRUCTURAL DYNAMICS

#### 3.1 Introduction

A optimization problem of structural dynamics could be presented as follows,

$$\text{Minimize } \varphi_0( Z, \zeta, \mathbf{b} ) = g( \zeta, \mathbf{b} ) + \int_0^{\tau} f( Z, \mathbf{b}, t ) dt \quad (3.1)$$

where  $\varphi_0$  is the cost function presented in a generalized form,  $\zeta$  is the fundamental frequency of the structure, and  $\mathbf{b}$  is a vector of design variable.

Subject to:

a). Constraints:

$$\mathbf{g}_1 = \int \varphi_i( Z, \mathbf{b}, t ) dt \quad (3.2)$$

$$\mathbf{g}_2 = \varphi_2( Z, \mathbf{b}, t )|_{t=t_1} \quad (3.3)$$

b). State equations:

$$M \ddot{Z} + C \dot{Z} + K Z = Q( \mathbf{b}, t ) \quad (3.4)$$

with initial conditions of

$$\dot{Z}(0) = \dot{Z}_0, \quad Z(0) = Z_0 \quad (3.5)$$

and,

$$(K - \Lambda M) \phi = 0 \quad (3.6)$$

Taking the first variation of Eq. (3.2), we have

$$\delta g_1 = \int_0^T \left[ \frac{\partial \varphi_1(Z, b, t)}{\partial b} \delta b + \frac{\partial \varphi_1(Z, b, t)}{\partial Z} \delta Z \right] dt \quad (3.7)$$

By introducing the Dirac-delta function, Eq. (3.3) could be written as

$$g_2 = \int_0^T \varphi_2(Z, b, t) \delta(t-t_i) dt \quad (3.8)$$

A first variation of Eq. (3.8) yields

$$\delta g_2 = \int_0^T \left[ \frac{\partial \varphi_2(Z, b, t)}{\partial b} \delta b + \frac{\partial \varphi_2(Z, b, t)}{\partial Z} \delta Z \right] \delta(t-t_i) dt \quad (3.9)$$

or,

$$\delta g_2 = \frac{\partial \varphi_2(Z, b, t)}{\partial b} \Big|_{t-t_i} \delta b + \frac{\partial \varphi_2(Z, b, t)}{\partial Z} \Big|_{t-t_i} \delta Z \quad (3.10)$$

Here, both integral-type and point-wise type constraint functions are discussed.

Usually, there are two types of constraints in a structural dynamic optimization problem: the first type is time independent, such as, frequencies, lower and upper bounds; the second type is time dependent, which is generated from structural dynamic responses, such as displacements, velocities, accelerations and stresses, etc.

The sensitivity analysis of the first type constraints is easy and straight forward. However, obtaining the derivatives of the time dependent constraints requires much more efforts. Thus, only the time dependent constraints are further discussed, which are in the general form of Eqs. (3.2) and (3.3).

To obtain the DSA information in structural dynamics, the same approach of system reduction technique is used as to compute dynamic response, which is presented in Chapters 2. Thus, one always performs sensitivity analysis with the reduced system in order to produce efficient computational work.

There are basically two approaches [18, 20] available for DSA, i.e., the direct differentiation method ( DDM ) and the adjoint variable method ( AVM ). These methods are briefly discussed in the following sections for structural dynamic problems.

### 3.2 The Direct Differentiation Method

Taking the first variation of Eq. (3.4) gives

$$\mathbf{M} \delta \ddot{\mathbf{Z}} + \mathbf{C} \delta \dot{\mathbf{Z}} + \mathbf{K} \delta \mathbf{Z} = \mathbf{R}(t) \delta \mathbf{b} \quad (3.11)$$

where

$$\mathbf{R}(t) = \frac{\partial \mathbf{Q}}{\partial \mathbf{b}} - \left[ \frac{\partial \mathbf{M}}{\partial \mathbf{b}} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{b}} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{b}} \mathbf{Z} \right] \quad (3.12)$$

From first order Taylor series expansion, one has

$$\delta Z = \frac{\partial Z}{\partial \mathbf{b}} \delta \mathbf{b} \equiv \mathbf{q} \delta \mathbf{b} \quad (3.13)$$

substituting eq. (3.13) into Eq. (3.11) yields,

$$\mathbf{M} \ddot{\mathbf{q}} + \mathbf{C} \dot{\mathbf{q}} + \mathbf{K} \mathbf{q} = \mathbf{R}(t) \quad (3.14)$$

with the initial condition of

$$\mathbf{q}(0) = \mathbf{0}, \quad \dot{\mathbf{q}}(0) = \mathbf{0} \quad (3.15)$$

which are derived from the initial conditions of the dynamic equilibrium equations.

Obviously, Eq. (3.14) and (3.15) are in the same form as Eq. (3.4) and (3.5). Therefore any method used to solve for state variable  $Z$  in Eqs. (3.4) and (3.5) could be used to solve for  $\mathbf{q}$ .

Once we have obtained the  $\mathbf{q}$ , the design sensitivity could then be computed by Eqs. (3.7) and (3.10) for the constraints  $g_1$  and  $g_2$ , respectively,

$$\frac{dg_1}{db} = \int_0^T \left[ \frac{\partial \varphi_1(Z, \mathbf{b}, t)}{\partial \mathbf{b}} + \frac{\partial \varphi_1(Z, \mathbf{b}, t)}{\partial Z} \mathbf{q} \right] dt \quad (3.16)$$

and,

$$\frac{dg_2}{db} = \frac{\partial \varphi_2(Z, \mathbf{b}, t)}{\partial \mathbf{b}} \Big|_{t=t_1} + \frac{\partial \varphi_2(Z, \mathbf{b}, t)}{\partial Z} \mathbf{q} \Big|_{t=t_1} \quad (3.17)$$

### 3.3 The Adjoint Variable Method

For this method, an adjoint variable vector,  $\lambda(t)$ , is introduced. Pre-multiplying Eq. (3.11) by  $\lambda^T(t)$  and further integrating over the time interval 0 to T, one has,

$$\int_0^T \lambda^T [M \delta \ddot{Z} + C \delta \dot{Z} + K \delta Z] dt = \int_0^T \lambda^T R(t) \delta b dt \quad (3.18)$$

Integrating by parts the first and the second terms in Eq. (3.18) respectively,

$$\int_0^T \lambda^T M \delta \ddot{Z} dt = \lambda^T M \delta \dot{Z} - \dot{\lambda}^T M \delta Z|_0^T + \int_0^T \ddot{\lambda}^T M \delta Z dt \quad (3.19)$$

and

$$\int_0^T \lambda^T C \delta \dot{Z} dt = \lambda^T C \delta Z|_0^T - \int_0^T \dot{\lambda}^T C \delta Z dt \quad (3.20)$$

It should be noticed that  $\delta Z(0)$  and  $\delta \dot{Z}(0)$  vanish due to the initial condition of dynamic equilibrium equations.

Now we set,

$$\lambda^T M \delta \dot{Z}|^T = 0 \quad (3.21)$$

$$\lambda^T C \delta Z|^T = 0 \quad (3.22)$$

and



$$\dot{\lambda}^T M \delta Z|_T = 0 \quad (3.23)$$

Since  $\delta Z(0)$  and  $\delta \dot{Z}(0)$  are arbitrary, so

$$\lambda(T) = \dot{\lambda}(T) = 0 \quad (3.24)$$

Thus, Eq. (3.18) becomes,

$$\int_0^T [ \ddot{\lambda}^T M - \dot{\lambda}^T C + \lambda^T K ] \delta Z dt = \int_0^T \lambda^T R(t) \delta b dt \quad (3.25)$$

Let the adjoint variable vector be defined as the solution of

$$M \ddot{\lambda}(t) - C \dot{\lambda}(t) + K \lambda(t) = \left( \frac{\partial \varphi_1}{\partial Z} \right)^T \quad (3.26)$$

By setting

$$t = T - \tau, \quad \lambda(t) = y(\tau) \quad (3.27)$$

and

$$\varphi_1(Z, b, t) = \varphi'_1(Z, b, \tau) \quad (3.28)$$

Eq. (3.26) becomes

$$M \ddot{y}(\tau) + C \dot{y}(\tau) + K y(\tau) = \left[ \frac{\partial \varphi'_1(Z, b, \tau)}{\partial Z} \right]^T \quad (3.29)$$

with the initial condition of

$$y(0) = 0, \quad \dot{y}(0) = 0 \quad (3.30)$$

It should be noted that Eqs. (3.29) and (3.30) are in the same form as Eqs. (3.4), (3.5) as well as Eqs. (3.14) and (3.15). Thus, same techniques could be applied to solve for  $y(\tau)$ . As soon as the adjoint variable vector is obtained, the design sensitivity of constraints can be calculated without difficulties.

Substituting Eq. (3.26) into Eq. (3.25),

$$\int_0^T \left( \frac{\partial \varphi_1}{\partial Z} \right)^T \delta Z dt = \int_0^T \lambda^T R(t) \delta b dt \quad (3.31)$$

and, inserting Eq. (3.31) into Eq. (3.7) leads to,

$$\frac{d\varphi_1}{db} = \int_0^T \left[ \frac{\partial \varphi_1(Z, b, t)}{\partial b} + \lambda^T R(t) \right] dt \quad (3.32)$$

For the point-wise constraints  $\varphi_2$ , similarly as for the DDM, the adjoint system of equations could be derived as

$$M \ddot{\lambda}(t) - C \dot{\lambda}(t) + K \lambda(t) = \left[ \frac{\partial \varphi_2(Z, b, t)}{\partial Z} \right]^T \delta(t-t_i) \quad (3.33)$$

and

$$\lambda(T) = 0, \quad \dot{\lambda}(T) = 0 \quad (3.34)$$

then, apply the similar manipulation as in Eqs. (3.27) and (3.28), one has

$$\mathbf{M} \ddot{\mathbf{y}}(\tau) + \mathbf{C} \dot{\mathbf{y}}(\tau) + \mathbf{K} \mathbf{y}(\tau) = \left[ \frac{\partial \phi_2(\mathbf{Z}, \mathbf{b}, t)}{\partial \mathbf{Z}} \right]^T \quad (3.35)$$

with the initial condition of

$$\mathbf{y}(0) = \dot{\mathbf{y}}(0) = \mathbf{0} \quad (3.36)$$

Eqs. (3.35) and (3.36) are once more in the same form as Eqs. (3.4) and (3.5).

Once  $\lambda(t)$  is computed, the design sensitivity of  $\phi_2$  could be obtained by using Eq. (3.10),

$$\frac{d\phi_2}{d\mathbf{b}} = \left. \frac{\partial \phi_2}{\partial \mathbf{b}} \right|_{t=t_1} + \int_0^T \lambda^T \mathbf{R}(t) dt \quad (37)$$

It should be indicated that the solution of adjoint variable vector in AVM involves the backwards integration, since when  $t=0$ ,  $\tau = T$ , and when  $t = T$ ,  $\tau = 0$ . So, the integration is carried out from  $\tau = T$  to  $\tau = 0$ . While, in the DDM, the integration for obtaining  $\mathbf{q}$  is a forward integrating process.

### 3.4 The Methods Justification for DSA in Structural Dynamics

With the discussion from previous chapters, it can be seen that the DDM is more suitable for applying in parallel codes, since each set of sensitivity equations corresponding to a design variable is independent. In the DSA, it is found that the DDM is suitable for DSA in structural dynamics. The situation, here, of DSA in

structural dynamics is quite different from those in the static case, where, the choice of DDM or AVM is accomplished by comparing the number of design variables and the number of constraints. The adoption of DDM for this study is based on several advantages: 1) both DDM DSA and dynamic response analysis use forward integration, and, thus, have consistent accuracy consistency, while the AVM uses backward integration and there is inconsistent; 2) there is no need to store the time-dependent response history, which saves computer memory space; 3) no interpolation is required; and 4) DDM is suitable for parallel computation. Therefore, the DDM is adopted for DSA in this study.

### 3.5 DSA Based on the Original Full System

Let  $b \in R^k$  be the vector of design variables. Taking the derivatives of both sides of Eq. (3.4) with respect to the design variable  $b_i$ , one obtains

$$M \frac{d\ddot{Z}}{db_i} + C \frac{d\dot{Z}}{db_i} + K \frac{dZ}{db_i} = R(t) \quad (3.38)$$

where

$$R(t) \equiv \frac{dQ}{db_i} - \left( \frac{dM}{db_i} \ddot{Z} + \frac{dC}{db_i} \dot{Z} + \frac{dK}{db_i} Z \right) \quad (3.39)$$

with the initial conditions

$$\frac{d\dot{Z}}{db_i}(0) = 0, \quad \frac{dZ}{db_i}(0) = 0 \quad (3.40)$$

Thus, any time integration techniques can be directly applied to Eqs. (3.38) and (3.40) to solve for the design sensitivity vectors of

$$\frac{dZ}{db_i}, \quad \frac{d\dot{Z}}{db_i} \text{ and } \frac{d\ddot{Z}}{db_i}$$

### 3.6 The Reduced System Formulation

Taking the derivatives of both sides of Eq. ( 2.39 ) with respect to the design variable  $b_i$ , one obtains

$$\bar{M} \frac{d\ddot{u}}{db_i} + \bar{C} \frac{d\dot{u}}{db_i} + \bar{K} \frac{du}{db_i} = \bar{R}(t) \quad (3.41)$$

where:

$$\bar{R}(t) = \frac{d\bar{Q}}{db_i} - \left( \frac{d\bar{M}}{db_i} \ddot{u} + \frac{d\bar{C}}{db_i} \dot{u} + \frac{d\bar{K}}{db_i} u \right) \quad (3.42)$$

Realizing Eq. (2.40), one can calculate  $\frac{d\bar{M}}{db_i}$  in Eq. (3.42) as

$$\frac{d\bar{M}}{db_i} = \phi^T \frac{dM}{db_i} \phi + \frac{d\phi^T}{db_i} M \phi + \phi^T M \frac{d\phi}{db_i} \quad (3.43)$$

Similarly,

$$\frac{d\bar{K}}{db_i} = \phi^T \frac{dK}{db_i} \phi + \frac{d\phi^T}{db_i} K \phi + \phi^T K \frac{d\phi}{db_i} \quad (3.44)$$

and

$$\frac{d\bar{C}}{db_i} = \phi^T \frac{dC}{db_i} \phi + \frac{d\phi^T}{db_i} C \phi + \phi^T C \frac{d\phi}{db_i} \quad (3.45)$$

It should be pointed out that, when eigenvector matrix is used as base vector matrix, Eq. (3.33) gives a null matrix, and Eq. (3.34) equals the derivatives of eigenvalues.

The recovery of physical sensitivities of displacements is processed by

$$\frac{dZ}{db_i} = \phi \frac{du}{db_i} + \frac{d\phi}{db_i} u \quad (3.46)$$

The derivatives of velocities and accelerations for the original full system could be obtained with similar expressions as Eq. (3.46).

It should be noted that base vector derivatives is required in calculation of Eqs. (3.43) through (3.46). A lot of research effort have been devoted in the area of eigenvector derivatives [ 42, 43, 44, 45, 46 ]. Variety of the eigenvector derivative methods are reviewed in the following section. The calculation of eigenvector derivatives, however, is very tedious and requires a lot of computational time. Therefore, it is desirable to have an alternative formulation for DSA of the reduced dynamic equations without requiring the eigenvector derivatives, which is developed and presented in Section 3.8.

## 3.7 Sensitivity Calculation for Eigen-problems

### 3.7.1 Introduction

The DSA for eigen-problems could be used in providing dynamic constraints sensitivity (as seen in the last section), in approximating a new vibration mode shape due to a perturbation in a design variable, determining the effect of design changes on the dynamic behavior of a structure, and in tailoring mode shapes to minimize displacements at certain points on a structure. The determination of eigenvalue derivatives is a straight forward and simple calculation. However, the calculation of the eigenvector derivatives is found to be much more involved and complicated. Mainly, there are four methods available: The finite difference method, the modal method, the modified modal method, and the Nelson's method. There are two other methods which are considered alternatives to Nelson's method, one is called the direct approach, while the other is called the iterative approach.

Before the introducing of various techniques, the technical background for eigen-system derivatives is first reviewed.

The generalized eigen-problem could be expressed as:

$$(K - \Lambda M) \phi = 0 \quad (3.47)$$

The condition of normalization:

$$\phi_j^T M \phi_j = 1 \quad (3.48)$$

The eigenvalue derivatives could be computed as

$$\frac{d\lambda_j}{db_i} = \phi_j^T \left( \frac{dK}{db_i} - \lambda_j \frac{dM}{db_i} \right) \phi_j \quad (3.49)$$

where  $b_i$  is the  $i^{\text{th}}$  design variable.

Differentiating Eq. (3.47) with respect to a design variable  $b_i$ , we have

$$\left[ K - \lambda_j M \right] \frac{d\phi_j}{db_i} = \frac{d\lambda_j}{db_i} M \phi_j - \frac{dK}{db_i} \phi_j + \lambda_j \frac{dM}{db_i} \phi_j \quad (3.50)$$

It is obvious that a direct solution of Eq. (3.50) is impossible since  $(K - \lambda M)$  is singular. However, we can use the following methods [42, 44] to overcome this difficulty and solve Eq. (3.50).

### 3.7.2 The Finite Difference Method

A step-by-step procedure of the finite difference method is presented as follows:

Do for each design variable:

- a. Solve Eq. (3.47) for  $\phi_{j(\text{old})}$
- b. perturbed the  $i^{\text{th}}$  design variable  $b_i$  by
 
$$b_{i(\text{new})} = b_{i(\text{old})} + \Delta b_i$$
- c. solve Eq. (3.47) again for  $\phi_{j(\text{new})}$



d. Approximate the eigenvector derivative by finite difference method

$$\frac{d\phi_j}{db_i} = \frac{\phi_{j(\text{new})} - \phi_{j(\text{old})}}{\Delta b_i}$$

It should be mentioned that, for this method, the choosing of  $\Delta b_i$  is very critical to the accuracy. And the re-calculation of the eigen-problem is required for each design variable, which is very time consuming.

### 3.7.3 The Modal Method

The modal method approximates the eigenvector derivatives as a linear combination of mode shapes (eigenvectors),

$$\frac{d\phi_j}{db_i} = \sum_{k=1}^m A_{ijk} \phi_k \quad (3.51)$$

where

$$A_{ijk} = \frac{\phi_k^T \left( \frac{dK}{db_i} - \lambda_j \frac{dM}{db_i} \right) \phi_j}{\lambda_j - \lambda_k} \quad \text{for } k \neq j \quad (3.52)$$

For  $k = j$ , Eq. (3.48) is differentiated to obtain

$$2 * \phi_j^T M \frac{d\phi_j}{db_i} + \phi_j^T \frac{dM}{db_i} \phi_j = 0 \quad (3.53)$$

substituting Eq. (3.51) into Eq. (3.53), we have

$$A_{ijk} = -\frac{1}{2} \phi_j^T \frac{dM}{db_i} \phi_j \quad \text{for } k = j \quad (3.54)$$

This method could be really expensive if a large number of modes are needed to accurately represent the mode shape derivatives.

### 3.7.4 The Modified Modal Method

The modified modal method was developed to reduce the number of modes needed to represent the derivative by including an additional term in the linear combination of the system mode shapes. The idea of the method, in fact, is adopted from mode acceleration method in structural dynamic response analysis.

First, by neglecting the term  $\lambda_j M \frac{d\phi_j}{db_i}$ , we solve Eq. (3.50) for  $\left\{ \frac{d\phi_j}{db_i} \right\}_s$ ,

which is the pseudo-static solution for  $\frac{d\phi_j}{db_i}$ , i.e.

$$\left( \frac{d\phi_j}{db_i} \right)_s = K^{-1} \left( \frac{d\lambda_j}{db_j} M - \frac{dK}{db_i} + \lambda_j \frac{dM}{db_i} \right) \phi_j \quad (3.55)$$

Then Eq. (3.51) is modified by adding the above pseudo-static solution,

$$\frac{d\phi_j}{db_i} = \frac{d\phi_j}{db_i} + \sum_{k=1}^m \bar{A}_{ijk} \phi_k \quad (57)$$

To obtain  $\bar{A}_{ijk}$ , the coefficients for the modified modal method, we substitute Eq. (3.56) into Eq. (3.50), and pre-multiply the results by  $\phi_k^T$ , we then have,

$$\bar{A}_{ijk} = \frac{\lambda_j \phi_k^T \left( \frac{dK}{db_i} - \lambda_j \frac{dM}{db_i} \right) \phi_j}{\lambda_k (\lambda_j - \lambda_{[k]})} \quad \text{for } k \neq j \quad (3.57)$$

and

$$\bar{A}_{ijk} = -\frac{1}{2} \phi_j^T \frac{dM}{db_i} \phi_j \quad \text{for } k = j \quad (3.58)$$

It is showed that the modified modal method converges faster than the modal method [44].

### 3.7.5 The Nelson's method

The method proposed by Professor Richard B. Nelson [42] expresses the eigenvector derivatives in terms of a complementary solution  $C_j \phi_j$  and a particular solution  $V_j$ , with  $C_j$  an undetermined coefficient, i.e.

$$\frac{d\phi_j}{db_i} = V_j + C_j \phi_j \quad (3.59)$$

Substituting Eq. (3.59) into Eq. (3.50) to obtain

$$(\mathbf{K} - \lambda_j \mathbf{M}) V_j = F_j \quad (3.60)$$

where

$$F_j \equiv \left( \frac{\partial \lambda_j}{\partial b_i} \mathbf{M} - \frac{\partial \mathbf{K}}{\partial b_i} + \lambda_j \frac{\partial \mathbf{M}}{\partial b_i} \right) \phi_j \quad (3.61)$$

Since  $(\mathbf{K} - \lambda_j \mathbf{M})$  is a singular matrix, we transform Eq. (3.60) into  $\tilde{\mathbf{K}} V_j = F_j$ , which is then a non-singular one, and we could solve for  $V_j$ . Here,  $\tilde{\mathbf{K}}$  is modified by zero the  $k^{\text{th}}$  row and column, except for the  $k^{\text{th}}$  diagonal element, and all the other elements remain unaltered, and  $F_j$  is obtained by zeroing the  $k^{\text{th}}$  element of  $F_j$ . Note that in the solution, the  $k^{\text{th}}$  element of  $V_j$  is zero, i.e.  $V_k = 0$ . Then, we substitute Eq. (3.59) into Eq. (3.53), which yields

$$\phi_j^T \mathbf{M} (V_j + C_j \phi_j) = \mathbf{b} \quad (3.62)$$

with

$$\mathbf{b} \equiv -\frac{1}{2} \phi_j^T \frac{d\mathbf{M}}{db_i} \phi_j \quad (3.63)$$

Thus

$$C_j = \mathbf{b} - \phi_j^T \mathbf{M} V_j \quad (3.64)$$

Finally, the eigenvector derivatives are obtained by Eq. (3.59).

Numerical computation showed [44] that the Nelson's method is the most efficient one among the methods presented. Although due to the use of the DDM alternative formulation developed, which avoids the requirement of the eigenvector derivatives, here a parallel-vector Nelson's algorithm is presented for future study and implementation, which is one of the author's future research target.

### 3.7.6 The parallel-vector algorithm for the Nelson's method

DO 1 J = 1, NUMODES

$$\tilde{\mathbf{K}} = \mathbf{K} - \lambda \mathbf{M} \quad (\text{Parallel/vector})$$

Search for the maximum value element in the  $j^{\text{th}}$  eigenvector, and label it "k".

(Parallel)

Modify  $\tilde{\mathbf{K}}$  to a non-singular matrix. (Parallel)

Factorize  $\tilde{\mathbf{K}}$  (Parallel-vector)

For each design variable DO (parallel):

Parallel DO 2 NV=1,NUMDV ( = No. of design variables)

Compute  $\mathbf{F}_j$  (Vectorization)

Solve for  $\tilde{\mathbf{K}} \mathbf{V}_j = \mathbf{F}_j$  (Vectorization)

Evaluate  $\mathbf{b} = \frac{1}{2} \phi_j^T \frac{d\mathbf{M}}{db_i} \phi_j$  (Vectorization)

Calculate  $\mathbf{C} = \mathbf{b} - \mathbf{V}_j^T \mathbf{M} \phi_j$  (Vectorization)

Obtain  $\frac{d\phi_j}{db_i} = \mathbf{V}_j + \mathbf{C}_j \phi_j$  (Vectorization)

2            End parallel DO  
 1        CONTINUE

### 3.8 An Alternative DSA Formulation

In this section, an alternative formulation for DSA in structural dynamics which avoids eigenvector derivatives in computation is developed and presented.

Let the derivative of the displacement response with respect to the design variable  $b_i$  be expressed as a linear combination of the eigenvectors,

$$\frac{dZ}{db_i} \equiv \Phi q \tag{3.65}$$

The first and second time derivatives of Eq. (3.65) are given as

$$\frac{d\dot{Z}}{db_i} = \Phi \dot{q} \tag{3.66}$$

$$\frac{d\ddot{Z}}{db_i} = \Phi \ddot{q} \tag{3.67}$$

Substituting Eqs. (3.65) through (3.67) into Eq. (3.4), one obtains

$$M \Phi \ddot{q} + C \Phi \dot{q} + K \Phi q = R ( t ) \tag{3.68}$$

in which  $R$  has been defined in Eq. (3.39). Pre-multiplying  $\Phi^T$  to both sides of Eq. (3.68), one has

$$\bar{\mathbf{M}} \ddot{\mathbf{q}} + \bar{\mathbf{C}} \dot{\mathbf{q}} + \bar{\mathbf{K}} \mathbf{q} = \hat{\mathbf{R}} (t) \quad (3.69)$$

where  $\bar{\mathbf{M}}$ ,  $\bar{\mathbf{C}}$  and  $\bar{\mathbf{K}}$  have been defined previously in Eqs. (2.40) through (2.42), and

$\hat{\mathbf{R}}$  is given by

$$\hat{\mathbf{R}} (t) = \Phi^T \mathbf{R}(t) \quad (3.70)$$

Substituting Eq. (3.39) into Eq. (3.70), one obtains

$$\hat{\mathbf{R}} (t) = \Phi^T \left[ \frac{dQ}{db_i} - \left( \frac{dM}{db_i} \ddot{\mathbf{z}} + \frac{dC}{db_i} \dot{\mathbf{z}} + \frac{dK}{db_i} \mathbf{z} \right) \right] \quad (3.71)$$

Substituting Eq. (2.36) into Eq. (3.71), one has

$$\hat{\mathbf{R}} (t) = \Phi^T \left[ \frac{dQ}{db_i} - \left( \frac{dM}{db_i} \Phi \ddot{\mathbf{u}} + \frac{dC}{db_i} \Phi \dot{\mathbf{u}} + \frac{dK}{db_i} \Phi \mathbf{u} \right) \right] \quad (3.72)$$

Thus, in this alternative formulation, the step-by-step procedure of calculating

$\frac{dZ}{db_i}$ ,  $\frac{d\dot{Z}}{db_i}$  and  $\frac{d\ddot{Z}}{db_i}$  is summarized as follows:

**Step 1:** Calculating the eigenvector matrix  $\Phi$ ;

**Step 2:** Computing  $\bar{\mathbf{M}}$ ,  $\bar{\mathbf{C}}$  and  $\bar{\mathbf{K}}$  according to Eqs. (2.40) through (2.42);

**Step 3:** Computing  $\hat{\mathbf{R}}$  according to Eq. (3.72);

**Step 4:** Using any time integration techniques to solve for  $\mathbf{q}$ ,  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  as

shown in Eq. (3.69);

**Step 5:** Calculating  $\frac{dZ}{db_i}$ ,  $\frac{d\dot{Z}}{db_i}$  and  $\frac{d\ddot{Z}}{db_i}$  according to Eq. (3.65) through

Eq. (3.67).

As can be seen from the above step-by-step procedure, the tedious computation of eigenvector derivatives  $\frac{d\phi}{db_i}$  is avoided.

### 3.9 Relationship Between the Reduced and Alternative Formulations

In this section, the relationship between the existing reduced system formulation and the alternative formulation presented in the previous section is established. The equivalency of the two formulations is analytically proved.

Pre-multiplying both sides of Eq. (3.65) by  $\phi^T M$ , one has

$$\mathbf{q} = \phi^T M \frac{dZ}{db_i} \quad (3.73)$$

Taking derivative of Eq. ( 2.36 ) with respect to the design variable  $b_i$ , one obtains

$$\frac{dZ}{db_i} = \frac{d\phi}{db_i} \mathbf{u} + \phi \frac{du}{db_i} \quad (3.74)$$

substituting Eq. (3.74) into Eq. (3.73), one obtains

$$\mathbf{q} = \phi^T M \left( \frac{d\phi}{db_i} \mathbf{u} + \phi \frac{du}{db_i} \right) \quad (3.75)$$

or



$$\mathbf{q} = \boldsymbol{\phi}^T \mathbf{M} \frac{d\boldsymbol{\phi}}{db_i} \mathbf{u} + \frac{d\mathbf{u}}{db_i} \quad (3.76)$$

Eq. (3.76) establishes the relationship between the conventional reduced system formulation and the alternative formulation.

In the following, it is analytically proved that Eqs. (3.41) presented in Section 3.6 are equivalent to Eqs. (3.69) derived in Section 3.8.

Substituting Eq. (3.76) into Eq. (3.69) gives

$$\bar{\mathbf{M}} \frac{d\ddot{\mathbf{u}}}{db_i} + \bar{\mathbf{C}} \frac{d\dot{\mathbf{u}}}{db_i} + \bar{\mathbf{K}} \frac{d\mathbf{u}}{db_i} = \mathbf{R}^* (t) \quad (3.77)$$

where,

$$\mathbf{R}^* (t) = \hat{\mathbf{R}} (t) - \left( \bar{\mathbf{M}} \boldsymbol{\phi}^T \mathbf{M} \frac{d\boldsymbol{\phi}}{db_i} \ddot{\mathbf{u}} + \bar{\mathbf{C}} \boldsymbol{\phi}^T \mathbf{M} \frac{d\boldsymbol{\phi}}{db_i} \dot{\mathbf{u}} + \bar{\mathbf{K}} \boldsymbol{\phi}^T \mathbf{M} \frac{d\boldsymbol{\phi}}{db_i} \mathbf{u} \right) \quad (3.78)$$

and the vector  $\hat{\mathbf{R}}(t)$  has been defined in Eq. (3.72).

Comparing Eq. (3.77) with Eq. (3.41), it can be seen that if one can prove that  $\mathbf{R}^*(t)$  in Eq. (3.77) is identical to  $\bar{\mathbf{R}}(t)$  in Eq. (3.41), then it follows that the two formulations are identical.

Taking the derivatives of both sides of Eq. (2.43) with respect to the design variable  $\mathbf{b}_i$ , one gets,

$$\frac{d\bar{\mathbf{Q}}}{db_i} = \frac{d\boldsymbol{\phi}^T}{db_i} \mathbf{Q} + \boldsymbol{\phi}^T \frac{d\mathbf{Q}}{db_i} \quad (3.79)$$

Substituting Eq. (3.79) and Eqs. (3.43) through (3.45) into Eq. (3.42) yields

$$\begin{aligned} \bar{\mathbf{R}}(t) = & \left( \frac{d\phi^T}{db_i} \mathbf{Q} + \phi^T \frac{d\mathbf{Q}}{db_i} \right) - \left[ \left( \frac{d\phi^T}{db_i} \mathbf{M} \phi + \phi^T \frac{d\mathbf{M}}{db_i} \phi + \phi^T \mathbf{M} \frac{d\phi}{db_i} \right) \ddot{\mathbf{u}} \right. \\ & + \left( \frac{d\phi^T}{db_i} \mathbf{C} \phi + \phi^T \frac{d\mathbf{C}}{db_i} \phi + \phi^T \mathbf{C} \frac{d\phi}{db_i} \right) \dot{\mathbf{u}} \quad (3.80) \\ & \left. + \left( \frac{d\phi^T}{db_i} \mathbf{K} \phi + \phi^T \frac{d\mathbf{K}}{db_i} \phi + \phi^T \mathbf{K} \frac{d\phi}{db_i} \right) \mathbf{u} \right] \end{aligned}$$

Regarding to the alternative formulation, now let's substituting Eq. (3.72) into Eq. (3.78) and realizing that  $\phi^T \mathbf{M} \phi = \mathbf{I}$  ( an identity matrix ), and  $\phi \mathbf{M} \phi^T = \mathbf{I}$  also, when the matrix  $\phi$  includes all the modes. Then we have

$$\begin{aligned} \mathbf{R}^*(t) = & \phi^T \frac{d\mathbf{Q}}{db_i} - \phi^T \left( \frac{d\mathbf{M}}{db_i} \phi \ddot{\mathbf{u}} + \frac{d\mathbf{C}}{db_i} \phi \dot{\mathbf{u}} + \frac{d\mathbf{K}}{db_i} \phi \mathbf{u} \right) \\ & - \phi^T \left( \mathbf{M} \frac{d\phi}{db_i} \ddot{\mathbf{u}} + \mathbf{C} \frac{d\phi}{db_i} \dot{\mathbf{u}} + \mathbf{K} \frac{d\phi}{db_i} \mathbf{u} \right) \quad (3.81) \end{aligned}$$

Eq. (3.80) can be re-arranged into

$$\begin{aligned} \bar{\mathbf{R}}(t) = & \phi^T \frac{d\mathbf{Q}}{db_i} + \frac{d\phi^T}{db_i} \mathbf{Q} - \frac{d\phi^T}{db_i} ( \mathbf{M} \phi \ddot{\mathbf{u}} + \mathbf{C} \phi \dot{\mathbf{u}} + \mathbf{K} \phi \mathbf{u} ) \\ & - \phi^T \left( \frac{d\mathbf{M}}{db_i} \phi \ddot{\mathbf{u}} + \frac{d\mathbf{C}}{db_i} \phi \dot{\mathbf{u}} + \frac{d\mathbf{K}}{db_i} \phi \mathbf{u} \right) \quad (3.82) \\ & - \phi^T \left( \mathbf{M} \frac{d\phi}{db_i} \ddot{\mathbf{u}} + \mathbf{C} \frac{d\phi}{db_i} \dot{\mathbf{u}} + \mathbf{K} \frac{d\phi}{db_i} \mathbf{u} \right) \end{aligned}$$

Since the second and the third terms in Eq. (3.82) are canceled each other, it is proved that Eq. (3.81) and Eq. (3.82) are therefore identical.

### 3.10 The Mode Acceleration Method in Design Sensitivity Analysis

The idea of applying MAM to DSA in structural dynamics is directly adopted from the dynamic response analysis, which was presented in Chapter 2, where the MAM was applied as a means of improving the displacement in the cases of static component is significant or when the higher modes are excited. The same logic is applied here to enhance the accuracy of the design sensitivities. The use of the MMAM to improve the velocities and the accelerations has been presented in Chapter 2. However, the application of MMAM will lead to inconvenient third and fourth time derivatives in the pseudo-load expression in DSA equations. Thus, the MAM is applied to DSA in structural dynamics, which is presented as follows:

By manipulating Eq. (3.38), one could have,

$$\begin{aligned}
 \frac{dZ}{db_i} &= K^{-1} \left( R - \left[ M \frac{d\ddot{Z}}{db_i} + C \frac{d\dot{Z}}{db_i} \right] \right) \\
 &= K^{-1} R - (\phi \Omega^{-2} \phi^T) M \phi \frac{d\ddot{u}}{db_i} - (\phi \Omega^{-2} \phi^T) (\alpha M + \beta K) \phi \frac{d\dot{u}}{db_i} \\
 &= K^{-1} R - \phi \Omega^{-2} \frac{d\ddot{u}}{db_i} - \alpha \phi \Omega^{-2} \frac{d\dot{u}}{db_i} - \beta \phi \frac{d\dot{u}}{db_i}
 \end{aligned} \tag{3.83}$$

Thus,

$$\frac{dZ}{db_i} = K^{-1} R - \phi \left[ \Omega^{-2} \frac{d\ddot{u}}{db_i} + (\alpha \Omega^{-2} + \beta I) \frac{d\dot{u}}{db_i} \right] \tag{3.84}$$

provided proportional damping is applied.

### 3.11 Design Sensitivity Analysis of Stresses

In finite element analysis [30, 31], when the nodal displacement vector has been determined, the element stresses could be calculated by using the stress-strain relationship as,

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon} \quad (3.85)$$

where,  $\boldsymbol{\sigma}$  is the stress vector,  $\mathbf{E}$  is the elastic-coefficient matrix, or, called elasticity matrix, and  $\boldsymbol{\varepsilon}$  is the element strain vector.

The displacement vector is presented as

$$\mathbf{u} = \mathbf{N} \mathbf{Z} \quad (3.86)$$

in which,  $\mathbf{N}$  is the shape function matrix, and  $\mathbf{Z}$  is the nodal displacement vector.

The strain vector is then obtained by

$$\boldsymbol{\varepsilon} = \mathbf{B} \mathbf{Z} \quad (3.87)$$

where,  $\mathbf{B}$  is obtained from the shape function matrix  $\mathbf{N}$  through appropriate differentiation.

Thus, the element stresses could be expressed as

$$\boldsymbol{\sigma} = \mathbf{E} \boldsymbol{\varepsilon} = \mathbf{E} \mathbf{B} \mathbf{Z} = \mathbf{S} \mathbf{Z} \quad (3.88)$$

in which,  $\mathbf{S}$  is defined as the stress matrix.

A. Stress DSA based on the original system:

Taking derivatives of Eq. (3.88) with respect to a design variable  $b_i$ , one has

$$\frac{d\sigma}{db_i} = \frac{dS}{db_i} Z + S \frac{dZ}{db_i} \quad (3.89)$$

which gives the stress derivatives for the formulation based on the original system.

B. Stress DSA based on the reduced system:

In the reduced system formulation, since

$$Z = \phi u \quad (3.90)$$

so,

$$\sigma = S \phi u \quad (3.91)$$

Taking derivatives of Eq. (3.91) with respect to a design variable yields,

$$\frac{d\sigma}{db_i} = S \phi \frac{du}{db_i} + \frac{dS}{db_i} \phi u + S \frac{d\phi}{db_i} u \quad (3.92)$$

It could be seen that in the stress derivatives for the reduced system formulation eigenvector derivatives are involved.

C. Stress DSA with the alternative formulation:

Substituting Eq. (3.65) into Eq. (3.89) yields

$$\frac{d\sigma}{db_i} = \frac{dS}{db_i} Z + S \phi q \quad (3.93)$$

or,

$$\frac{d\sigma}{db_i} = \frac{dS}{db_i} \phi u + S \phi q \quad (3.94)$$

in which, there are no eigenvector derivatives involved. Therefore, the alternative formulation also has advantage over the reduced system formulation in computation of the stress derivatives.

### 3.12 Design Sensitivity for Non-linear Structural Dynamics

For nonlinear structural dynamics problems the equation of motion can be represented as

$$\mathbf{M}(\mathbf{Z}, \mathbf{b}) \ddot{\mathbf{Z}} + \mathbf{C}(\mathbf{Z}, \mathbf{b}) \dot{\mathbf{Z}} + \mathbf{K}(\mathbf{Z}, \mathbf{b}) \mathbf{Z} = \mathbf{Q}(t) \quad (3.95)$$

In many practical situations, system reduction methods could be applied [ 51, 52, 53] to nonlinear structural dynamic problems. Substituting Eq. ( 2.36 ) into Eq. (3.95) and with some manipulating, one obtains

$$\bar{\mathbf{M}}(\mathbf{u}, \mathbf{b}) \ddot{\mathbf{u}} + \bar{\mathbf{C}}(\mathbf{u}, \mathbf{b}) \dot{\mathbf{u}} + \bar{\mathbf{K}}(\mathbf{u}, \mathbf{b}) \mathbf{u} = \bar{\mathbf{Q}}(t) \quad (3.96)$$

where, the definitions of the matrices  $\bar{\mathbf{M}}$ ,  $\bar{\mathbf{C}}$ , and  $\bar{\mathbf{K}}$ , as well as  $\bar{\mathbf{Q}}$ , could be refer to Eqs. (2.40) through ( 2.43 ).

#### 3.12.1 Formulation Based On the Original System

Taking the derivatives of both sides of Eq. (3.95) with respect to the design variable vector  $\mathbf{b}_i$ , one obtains

$$\mathbf{M} \frac{\partial \ddot{\mathbf{Z}}}{\partial \mathbf{b}_i} + \mathbf{C} \frac{\partial \dot{\mathbf{Z}}}{\partial \mathbf{b}_i} + \dot{\mathbf{K}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} = \mathbf{R}(t) \quad (3.97)$$

Where,

$$\hat{\mathbf{K}} = \mathbf{K} + \frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{Z}} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{Z}} \mathbf{Z} \quad (3.98)$$

and

$$\mathbf{R}(t) = \frac{\partial \mathbf{Q}}{\partial \mathbf{b}_i} - \left[ \frac{\partial \mathbf{M}}{\partial \mathbf{b}_i} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{b}_i} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{b}_i} \mathbf{Z} \right]$$

The above three equations present the DSA formulation with the original system in non-linear structural dynamics.

### 3.12.2 Formulation Based On the Reduced System

To obtain the DSA formulation based on the reduced system, we take the derivatives of both sides of Eq. (3.96) with respect to the design variable  $\mathbf{b}_i$ ,

$$\bar{\mathbf{M}} \frac{\partial \ddot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{C}} \frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{K}} \frac{\partial \mathbf{u}}{\partial \mathbf{b}_i} = \bar{\mathbf{R}}(t) \quad (3.100)$$

Where

$$\bar{\mathbf{K}} = \bar{\mathbf{K}} + \frac{\partial \bar{\mathbf{M}}}{\partial \mathbf{u}} \ddot{\mathbf{u}} + \frac{\partial \bar{\mathbf{C}}}{\partial \mathbf{u}} \dot{\mathbf{u}} + \frac{\partial \bar{\mathbf{K}}}{\partial \mathbf{u}} \mathbf{u} \quad (3.101)$$

$$\bar{\mathbf{R}} = \frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{b}_i} - \left( \frac{\partial \bar{\mathbf{M}}}{\partial \mathbf{b}_i} \ddot{\mathbf{u}} + \frac{\partial \bar{\mathbf{C}}}{\partial \mathbf{b}_i} \dot{\mathbf{u}} + \frac{\partial \bar{\mathbf{K}}}{\partial \mathbf{b}_i} \mathbf{u} \right) \quad (3.102)$$

and



$$\frac{\partial \bar{M}}{\partial b_i} = \Phi^T \left( \frac{\partial M}{\partial b_i} + \frac{\partial M}{\partial Z} \frac{\partial Z}{\partial b_i} \right) \Phi + \frac{\partial \Phi^T}{\partial b_i} M \Phi + \Phi^T M \frac{\partial \Phi}{\partial b_i} \quad (2.103)$$

$$\frac{\partial \bar{K}}{\partial b_i} = \Phi^T \left( \frac{\partial K}{\partial b_i} + \frac{\partial K}{\partial Z} \frac{\partial Z}{\partial b_i} \right) \Phi + \frac{\partial \Phi^T}{\partial b_i} K \Phi + \Phi^T K \frac{\partial \Phi}{\partial b_i} \quad (3.104)$$

$$\frac{\partial \bar{C}}{\partial b_i} = \alpha \frac{\partial \bar{M}}{\partial b_i} + \beta \frac{\partial \bar{K}}{\partial b_i} \quad (3.105)$$

in which, provided the Rayleigh damping is applied.

### 3.12.3 An Alternative Formulation Based On the Reduced System

Substituting Eqs. (3.66) through (3.68) into Eq. (3.97), and pre-multiplying  $\Phi^T$  to both sides of the equation, one obtains

$$\bar{M} \ddot{q} + \bar{C} \dot{q} + \bar{K} q = \bar{R}(t) \quad (3.106)$$

Where

$$\bar{K} = \Phi^T \hat{K} \Phi \quad (3.107)$$

$$\bar{R} = \Phi^T R \quad (3.108)$$

It should be noted that Eq. (3.76), which established the relationship between the reduced and alternative formulations, is still valid for nonlinear systems, provided that the system reduction techniques are applicable.

Thus for design sensitivity analysis of nonlinear structural dynamics problems, once can solve  $q$ ,  $\dot{q}$ , and  $\ddot{q}$  from Eq. (3.106). The unknown design sensitivity

variables,  $\frac{\partial z}{\partial b_i}$ ,  $\frac{\partial \dot{z}}{\partial b_i}$ , and  $\frac{\partial \ddot{z}}{\partial b_i}$  can then be solved from Eqs. (3.66), (3.67) and

(3.68). Thus with the alternative formulation for nonlinear structural dynamic the DSA could be conducted without the calculating the derivatives of eigenvectors.

For DSA of linear structural dynamic systems, it has been proved analytically in Section 3.9 that the alternative formulation and reduced system formulation are equivalent, provided the transformation from the original system to the reduced system is exact.

For DSA of nonlinear structural dynamic systems, it could also be proved analytically that the reduced and alternative formulations are equivalent, provided the transformation is exact. The proof is presented in Appendix A.

## **4. THE PARALLEL-VECTOR DSA ALGORITHM**

### **4.1 Introduction**

We are in a decade of supercomputers. Human beings have a very long history of use computing devices. Abacus was the original computer invented in ancient China. Since the first electronic computer was invented in the early 1950's, computer performance [54] has increased over the past three decades by a factor of 10 every five years. The electric computers have gone through five generations of developments. For the first generation, electro-mechanical relays or vacuum tubes were used to implement logical and memory, and all the programming was done in machine language. The second generation used transistors, printed circuits, and magnetic core memory, and assembly language was used. The third generation is characterized by the use of small-scale and medium-scale integrated circuits, and semi-conductor memory began to replace magnetic core memory. The IBM system/360 series are well known examples of this generation. The fourth generation uses large-scale integration and VLSI to construct logical and memory units. Most of the operation systems are time shared and the use of virtual memory is available. Then vectorizing compilers for pipelined vector processors appeared as the fifth

generation. Cray 1 is a typical example of this generation. It is believed that the Cray 2 and Cray Y-MP is for another generation, which has not only vectorization for each CPU, also, with multi-parallel processors.

In this study, the supercomputer Cray 2 and Cray Y-MP are used to implement the parallel-vector algorithm developed. Cray 2 and Cray Y-MP are shared memory multi-processor systems with 4 and 8 processors respectively. Each Cray-2 and Cray Y-MP CPU is a high-speed vector processor with specialized pipelined functional units which can be utilized in parallel to perform high-speed floating point computations.

Basically, there are three approaches [55] for designing a parallel-vector algorithm: a) detect and exploit any inherent parallelism in an existing sequential algorithm; b) invent a new parallel algorithm; c) adapt another parallel algorithm that solves a similar problem. As far as the author's knowledge, there is no parallel-vector algorithm available in the literature for DSA in structural dynamics. However, in the mathematical view point, all the numerical algorithms, for example, solution of the simultaneous system equations, solution of the eigensystem, multiplying a matrix with a vector, etc., are all existing sequential algorithms. Therefore, the first approach is adopted in this study. The parallel Fortran language Force [ 29 ] is used to implement the developed parallel-vector algorithm in Cray-2 and Cray Y-MP high performance computers.

## 4.2 The Parallel Fortran Language Force

The Force is a portable parallel fortran language developed by Jordan [29], with which the programmer, insulated from process management is left free to concentrate on the synchronization issues of parallel programming.

The following is a quick glance of the Force language, detailed information is given in reference [29].

The Force macro declares that start of a parallel main program has the following syntax:

```
Force < name > of < nproc> ident <me >
```

then followed by variable declarations, the body of the parallel program. Instead ending the main program with "Return" and "End", the Force main program ends with " Join" and " End". For example,

```
Force DSA of NP ident ME
  <declarations>
End declarations
C Force body
Join
End
```

Where, NP is a user named shared integer variable containing the number of processors executing the program. ME is a user named private variable which contains a unique index for each processor, numbered between 1 and NP.

The parallel subroutines are declared in the format of

Forcesub <name> (<parameter list>) of <nproc> ident <ME>

for example,

C .....Matrix multiplication subroutine:  $C = A*B$

Forcesub MULT(A,B,C,N1,N2,M1) of NP ident ME

INTEGER N1,N2,M1

REAL A(N1,N2),B(N2,M1),C(N1,M1)

Private INTEGER I,J,K

End declarations

C Initialize C

Pre2do 100 I=1,N1; J=1,M1

C(i,J)=0.0

100 End presched DO

C Multiplication process

Presched DO 300 I=1,N1

DO 200 J=1,M1

```

                                DO 200 K=1,N2
200                                C(I,J)=C(I,J)+A(I,K)*B(K,J)
300                                End presched DO

                                RETURN

                                END

```

To call the Force subroutine, Forcesub, instead of using "CALL", "Forcecall" is used. For example,

```

                                Shared REAL A(100,50),B(50,100),C(100,100)
                                Private N1,N2,M1
                                End declarations
                                .
                                .
                                Forcecall MULT(A,B,C,N1,N2,M1)

```

Most of the Fortran statements are valid in Force. Different from the Fortran, in a Force program, variables (including arrays) need to be declared as either "Shared" or "Private". When a variable is declared "Shared", only one copy of it is maintained by all the processors, i.e., all the processors communicate through shared memory locations. If a variable is "Private", then each processor has its own storage space for the variable, even though the variable is named only once in the

main program. Besides, there are synchronous variables in a Force program. The common format for variable declarations is

Declared-type <Type> <Variable list>

for example,

Shared Real A(1000), B(1000)

Private Real TT(1001)

Private Integer I, J

Shared Logical OK

Async Real X

The parallel Do-loops are identified by "Presched DO", "Selfsched DO", etc., for example,

```
Presched DO 2 I=1, LL
      DO 1 J=1,MM
      A(I,J) = FLOAT(I,J)
1      CONTINUE
2      End presched DO
```

```
Selfsched DO 3 J=1,LM
```



C(J)=0.0

IF ( J .GT. LM/2) CALL HARDJOB(C(J))

3 End Selsched DO

It should be mentioned that, in the parallel Do-loop, the computation work for the Do-loop variable *i* must be independent of each other, otherwise the parallel Do-loop is misused. The difference between the "Presched DO" and the "selsched DO" is in the way of assigning the computing task to be taken care of by which processor. One is pre-scheduled, and the other is determined during the executing of the parallel Do-loop, in order to assign evenly the computing loads to each processor.

Synchronization is realized through "Barrier", "Critical", "Consume", "Copy", etc.

#### **4.3 Techniques and Skills Related to Vectorization**

The performance of programs executing on vector computers could be significantly improved when the number of accesses to memory is reduced. Unrolling Fortran Do-loops [56], followed by substitutions and eliminations in the unrolled code, can reduce the number of loads and stores. Some other skills, as making the long vector length, eliminating If Statements from Do-loops, are also important.

Here, two subroutines are presented to illustrate the idea behind the techniques of loop-unrolling and the vector-unrolling.

The first example is implementing the operation of a matrix-vector multiplication with level-8 loop-unrolling, where, the matrix is full populated and unsymmetric.

```
Forcesub MVMULP(A,B,C,IROWA,JCOLA,LL,MM) of NP ident ME
REAL A(LL,MM),B(1),C(1)
INTEGER IROWA,JCOLA,LL,MM
Shared INTEGER ND,NEND,NL1
Private INTEGER J
End declarations
C
Barrier
ND=JCOLA/8
NEND=(ND-1)*8+1
NL1=ND*8+1
DO 1 I=1,IROWA
1  C(I)=0.0
End barrier
C ** Level-8 loop-unrolling, with parallel do:
Presched DO 3 J=1,NEND,8
```

```

        DO 2 I=1,IROWA
        C(I)=C(I)+A(I,J)*B(J)
1          +A(I,J+1)*B(J+1)
2          +A(I,J+2)*B(J+2)
3          +A(I,J+3)*B(J+3)
4          +A(I,J+4)*B(J+4)
5          +A(I,J+5)*B(J+5)
6          +A(I,J+6)*B(J+6)
7          +A(I,J+7)*B(J+7)
2          CONTINUE
3      End presched DO
C
    Barrier
    End barrier
C ** Taking care left-over:
    Presched DO 5 J=NL1,JCOLA
        DO 4 I=1,IROWA
            C(I)=C(I)+A(I,J)*B(J)
4          CONTINUE
5      End presched DO
    Barrier

```

End barrier

RETURN

END

The second example implements the multiplication of a transposed matrix with a vector with level-8 vector unrolling, provided the matrix is un-symmetric and full populated.

Forcesub MTVP(A,B,C,IROWA,JCOLA,LL,MM) of NP ident ME

REAL A(LL,MM), B(1), C(1)

INTEGER IROWA,JCOLA,LL,MM

Shared INTEGER ND,NEND,NL1

Private INTEGER I

End declarations

C

Barrier

ND=JCOLA/8

NEND=(ND-1)\*\* + 1

NL1=ND\*8+1

DO 1 I=1,JCOLA

1 C(I)=0.0

End barrier

C \*\* Level-8 vector unrolling, with parallel do:

Presched DO 3 I=1,NEND,8

DO 2 J=1,IROWA

$C(I) = C(I) + A(J,I) * A(J,I) * B(J)$

$C(I+1) = C(I+1) + A(J,I+1) * A(J,I+1) * B(J)$

$C(I+2) = C(I+2) + A(J,I+2) * A(J,I+2) * B(J)$

$C(I+3) = C(I+3) + A(J,I+3) * A(J,I+3) * B(J)$

$C(I+4) = C(I+4) + A(J,I+4) * A(J,I+4) * B(J)$

$C(I+5) = C(I+5) + A(J,I+5) * A(J,I+5) * B(J)$

$C(I+6) = C(I+6) + A(J,I+6) * A(J,I+6) * B(J)$

$C(I+7) = C(I+7) + A(J,I+7) * A(J,I+7) * B(J)$

2 CONTINUE

3 End presched DO

Barrier

End barrier

C \*\* Taking care of left-over:

Presched DO 5 I=NL1,JCOLA

DO 4 J=1,IROWA

4  $C(I) = C(I) + A(J,I) * B(J)$

5 End presched DO

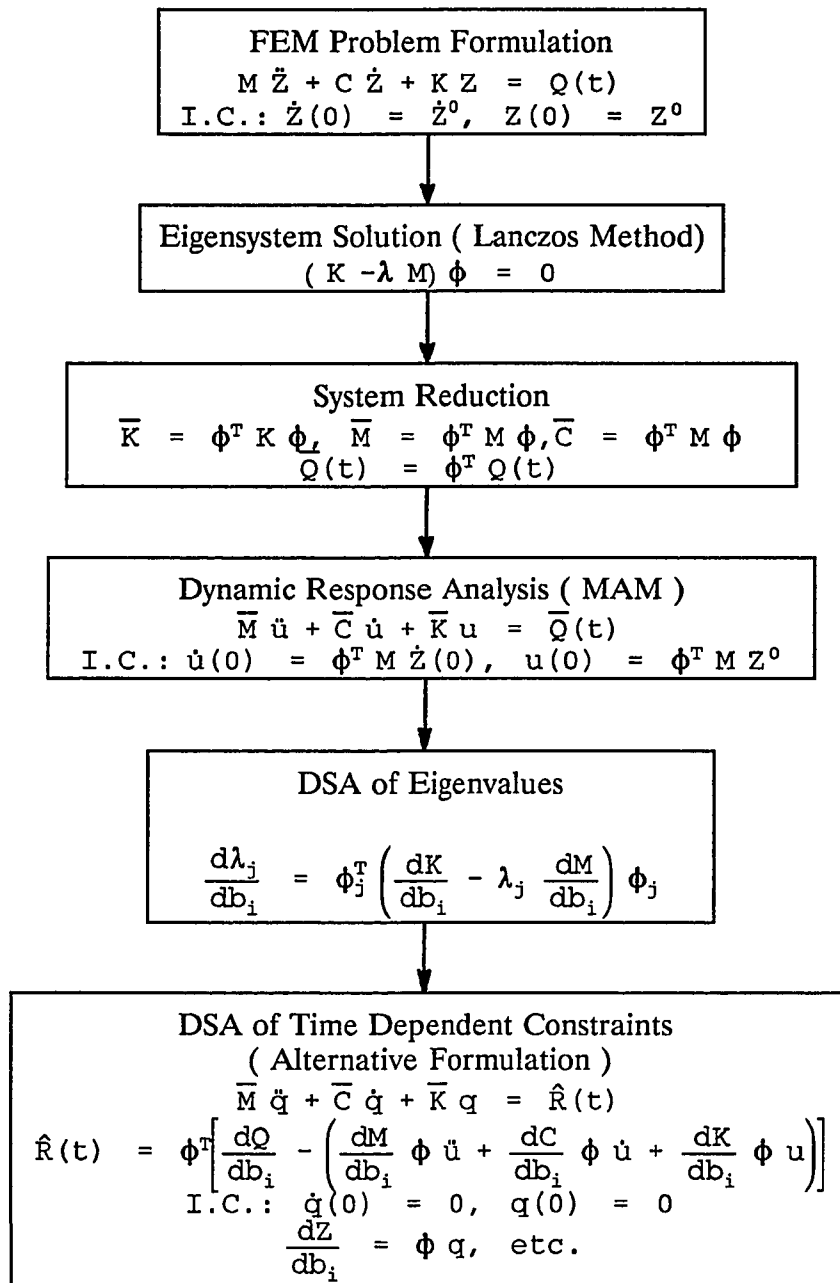
C

Barrier

```
End barrier
C  RETURN
   END
```

#### **4.4 Overview of the Alternative DSA Algorithm**

An effective algorithm for DSA in structural dynamics has been built up through the studies presented in the previous chapters. The algorithm is presented here in a flow chart as shown in Fig. 4.1.



**Fig.4.1 Flow Chart of DSA in Structural Dynamics**

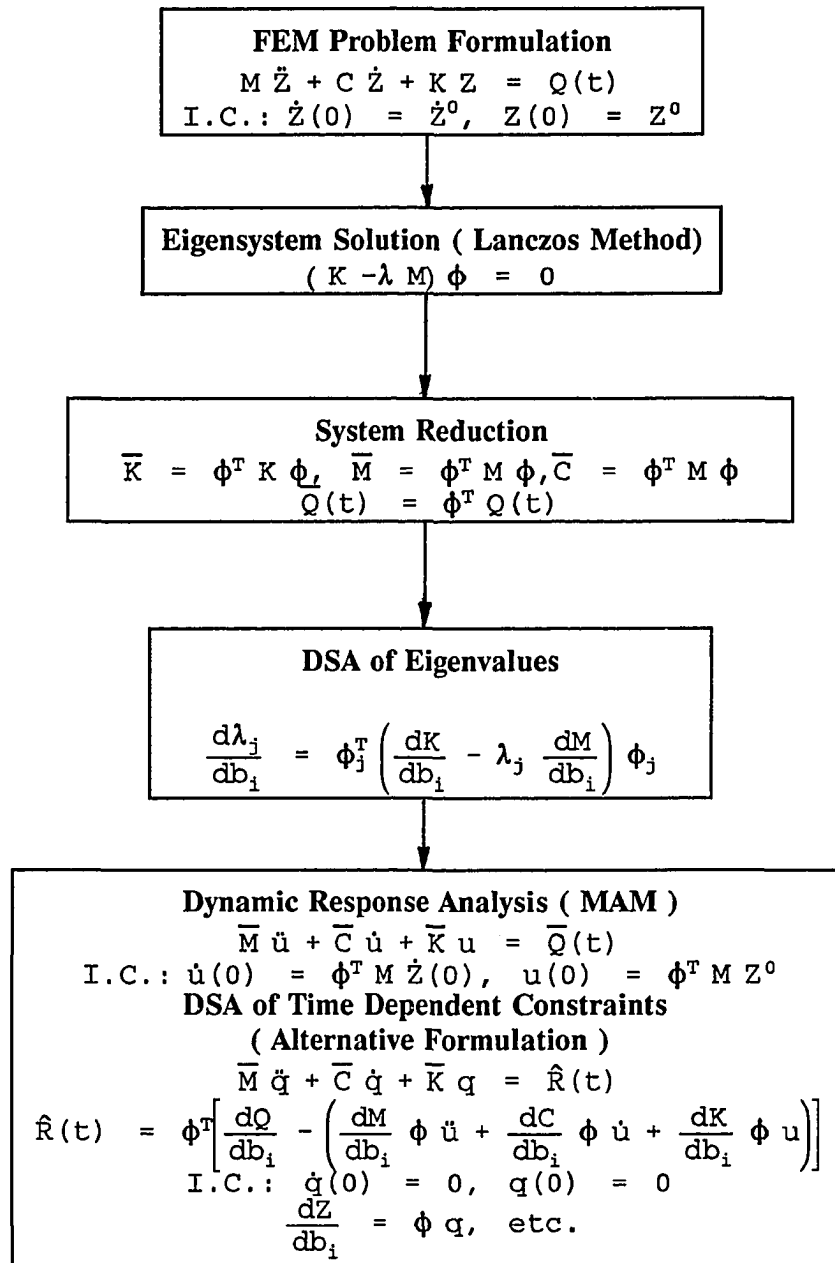
## **4.5 Designing of the Parallel-Vector Algorithm**

The parallel-vector algorithm is developed by exploring the parallelism of the design sensitivity algorithm in both global and local levels. Also, the vector computing is effectively employed in the algorithm to achieve high computational efficiency.

### **4.5.1 Global Parallelization**

The idea of realizing the global parallelization is illustrated by the flow chart shown in Fig. 4.2. In a sequential algorithm, the dynamic response analysis is carried out first, and then followed by the DSA computation. Here, in the parallel algorithm, the dynamic response analysis and the DSA are applied at the same time. Thus, if 2 processors are applied, the global parallelization could be realized, i.e., processor 1 and 2 could carry out dynamic response analysis and DSA in parallel.





**Fig.4.2 Flow Chart of DSA in Structural Dynamics: Global and Local Parallelization**

#### **4.5.2 Local Parallelization and Vectorization**

Besides the global parallelization, the local parallelization is applied at different levels of computation in various portion of the DSA algorithm. Vectorization techniques are practiced through all the subroutines in order to assist achieving the goal of developing an efficient parallel-vector algorithm ( see Fig. 4.2).

#### **4.5.3 The Parallel-Vector Equation Solver**

A parallel-vector equation solver [2] is incorporated with this parallel-vector algorithm. The parallel-vector solver is developed based on Choleski method for the solution of symmetric, sparse, large scale system of equations. The matrix involved in the system of equations are stored in row-wise skyline form. The solver has taken use of the variable-band storage scheme to reduce the number of the operations in the Choleski factorization. The algorithm employs parallel computation in the outermost Do-loop and vector computation via the loop unrolling techniques in the innermost Do-loop.

#### **4.5.4 The Parallel-Vector Eigensystem Solver**

Parallel-vector eigen-solver [3] developed based on the Lanczos algorithm is applied. The Lanczos method has been presented in Chapter 2. The parallel-vector

version of the Lanczos algorithm is developed by exploring the inherent parallelism and make use of the vector computation.

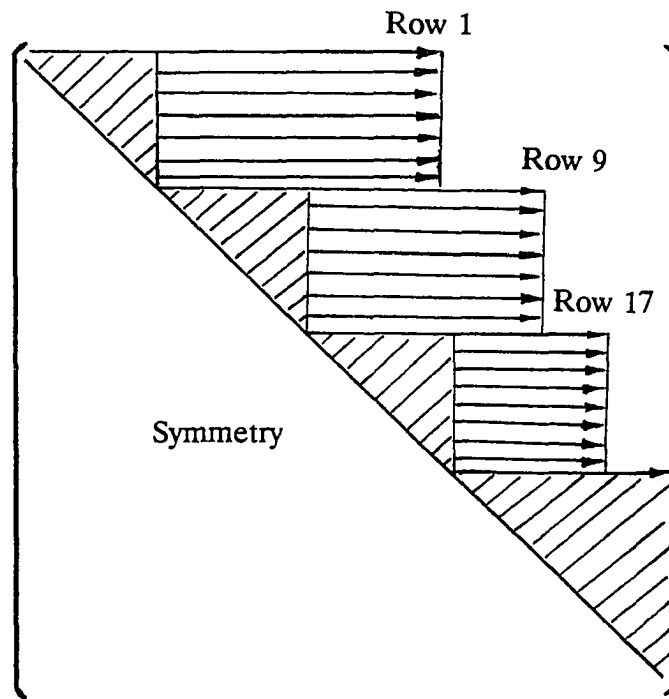
#### **4.5.5 Parallel-Vector Matrix Vector Multiplication**

The matrix vector multiplication is also a very commonly used operation in most of the structural engineering problems. Especially, for large-scale structure systems in which matrices with huge size are involved, the matrix-vector multiplication becomes a very costly computation task. Therefore, it is of great importance to develop parallel-vector matrix-vector multipliers also.

There are basically three types of matrix-vector multiplications, which is classified according to the storage of the matrices.

##### **A. Symmetric, Banded Sparse Matrix:**

For this type of matrices, the row-wise skyline storage is used. In order to apply the level-8 loop-unrolling technique, the matrix is virtually divided into many blocks, and each block is modified to have block height of 8. By doing this, the algorithm could be well vectorized with both vector unrolling and loop unrolling. The storage scheme and the block dividing are shown in Fig. 4.3. The small shaded triangular areas are called the left-overs, which is then manipulated with parallelization and vectorization.



$$M = [ \text{Row 1, Row 2, \dots, Row n} ]$$

**Figure 4.3 Storage Scheme and Block Dividing of the Symmetric, Banded Sparse Matrix**

## B. Full Populated, Unsymmetric Matrices

When the matrices involved are full populated and unsymmetric, the level-eight loop-unrolling techniques are applied. The matrices are divided column-wise into many strips of width equal to eight, and saxpy operations applied. Unlike in the conventional multiplication of a matrix with a vector, where the result vector are obtained directly by multiplying each row of the matrix to the vector (dot-product operations). Instead, here, the result vector are obtained by keep adding of the partial results (saxpy operations). If the total number of the columns could not divide evenly by eight, then the remainder is called left-over, which should be taken cared.

## C. The Matrix Transpose Multiply to a Vector

Here, obviously, we are talking about unsymmetric, full populated matrices. We still divide the width of the matrix into many blocks with width of eight, and then apply vector-unrolling. Instead of performing dot-product one column each time and get one element of the result vector, in the vector-unrolling method, the vector is fetched once and made use for eight times by performing the eight columns in the block to multiply the vector, thus, we obtained eight elements for the result vector at each time.

In this chapter, parallel-vector computing background and a parallel Fortran language, Force, are reviewed; important techniques for vector computing are presented. A parallel-vector algorithm for DSA in structural dynamics is developed. The parallel-vector algorithm is then implemented in a Force code, PVDSASD (Parallel-Vector DSA in Structural Dynamics) [57]. The numerical studies are presented in the following chapter, with various examples to illustrate the accuracy, efficiency, and effectiveness of the algorithm.

## 5. NUMERICAL STUDIES

The parallel-vector algorithm for DSA in structural dynamics presented in previous chapters is implemented in a parallel Fortran ( Force ) code, PVDSASD. Beam, two-dimensional frame, and three-dimensional frame examples are analyzed in this chapter to illustrate the accuracy, and efficiency of the algorithm developed.

Before discussing the numerical examples, let's first define the error norms.

### 5.1 The Definition of Error Norms

To evaluate quantitatively the accuracy of the dynamic response and DSA information obtained with the alternative formulation developed, error norms are defined in this section.

The relative displacement error norm is defined as

$$\epsilon_u = \left( \frac{\delta^T \delta}{u_f^T u_f} \right)^{\frac{1}{2}} \quad (5.1)$$

where,

$$\delta = \mathbf{u}_f - \mathbf{u}_a \quad (5.2)$$

in which  $\mathbf{u}_f$  is the displacement vector obtained from the solution of the original full system, and  $\mathbf{u}_a$  is the approximate displacement vector calculated from the reduced system.

Similarly, the error norms for velocity, acceleration, and stress vectors are defined by replacing the displacement vector with the velocity, acceleration, and the stress vector respectively.

The error norms for DSA results are also defined in the similar manner as the dynamic responses. For example, the error norm for displacement derivatives is defined as,

$$\varepsilon_{\frac{du}{db}} = \left( \frac{\bar{\delta}^T \bar{\delta}}{\left(\frac{du}{db}\right)_{\text{ffdm}}^T \left(\frac{du}{db}\right)_{\text{ffdm}}} \right)^{\frac{1}{2}} \quad (5.3)$$

in which

$$\bar{\delta}^T = \left(\frac{du}{db}\right)_{\text{ffdm}} - \left(\frac{du}{db}\right)_a \quad (5.4)$$

where,  $(du/db)_{\text{ffdm}}$  is the derivative of displacement vector obtained using finite difference method, by perturbing the design variable when solving the original full system. And  $(du/db)_a$  is the approximate displacement derivative obtained by the alternative DSA algorithm developed, which works with the reduced system.



The error norms of stress derivatives are defined similarly as the one of the displacement derivatives, by replacing the displacement derivative vector with the stress derivative vector.

## **5.2 Beam Examples**

In this section two beam examples are illustrated.

### **5.2.1 Three-Element Cantilever Beam**

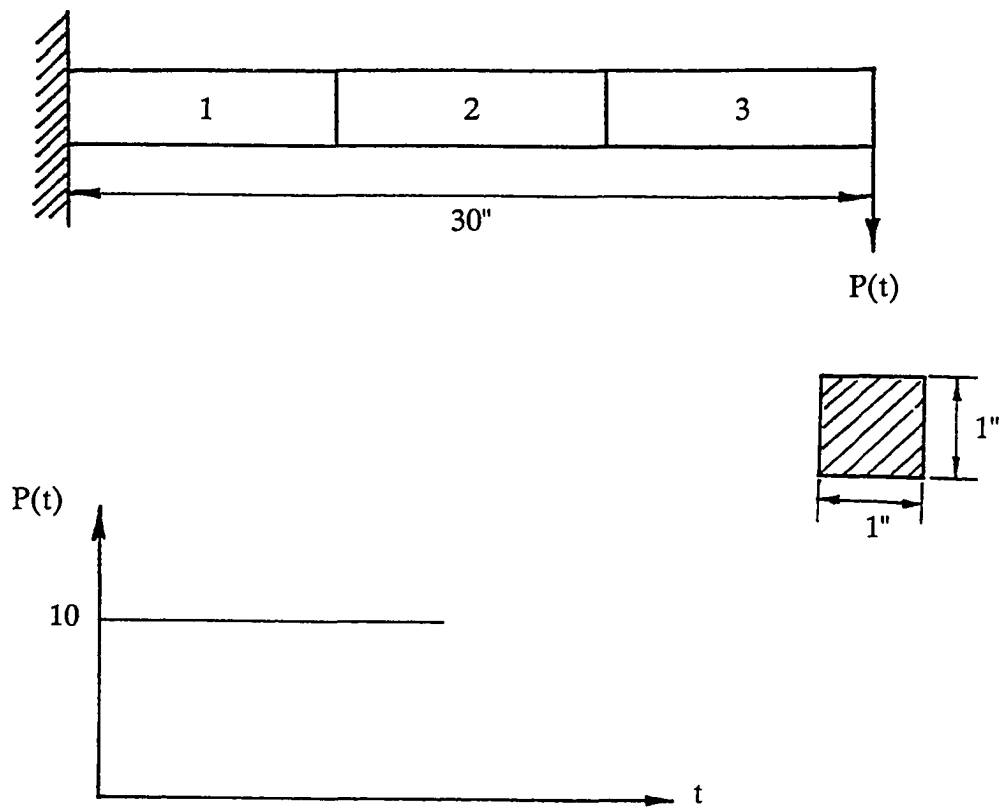
An aluminum cantilever beam [36] is modeled with three finite elements, as shown in Fig. 5.1. The beam is at rest at  $t = 0$  when a 10 lb concentrated mass is suddenly attached at the tip of the beam.

This small problem is mainly used to verify the correctness of the code developed for dynamic analysis. It also demonstrates the effectiveness of the MAM in comparing with the MDM. Table 5.1 gives the deflections at the tip of the cantilever for the first 20 time steps. The second column presents the solution quoted from Reference [26], which is obtained with one mode by linear acceleration method. The third, fourth, and the fifth columns show the results obtained by MDM with one, three and nine modes are applied respectively, where Duhamel integral method is used. The sixth column presents the solution by one mode MAM using

Duhamel integral. The correctness is verified by comparing with the deflections given in column one. Results also show the effectiveness of the MAM.

Table 5.2 also presents the tip deflections of the cantilever, which are obtained by Newmark method. The second column lists the results by solving the full system using Newmark integration method, which is then compared with the solutions by MDM with one mode and three modes, and the solution by MAM with one mode.

The CPU time for using the Newmark method and the Duhamel integral method is about the same, and a quantitative comparison will be given in Section 5.3.2.



$$E = 1.0 \times 10^6 \text{ psi}, \quad \rho = 2.591 \times 10^{-4} \text{ lb-sec}^2/\text{in}^4, \quad m = 0.02591 \text{ lb-sec}^2/\text{in}$$

$$A = bh = 1 \text{ in}^2, \quad I = 8.333 \times 10^{-2} \text{ in}^4$$

$$\text{Time step } \Delta t = 0.005 \text{ second.}$$

**Figure 5.1 A Three-Element Cantilever Beam**

**Table 5.1 Tip Deflection of the Three-Element Cantilever Beam  
( Duhamel Integral Method )**

Time (Second)	1 Mode Linear Acc.[26]	1 Mode MDM	3 Modes MDM	9 Modes MDM	1 Modes MAM
0.005	-0.0443	-0.0442565	-0.04437877	-0.0443489	-0.044348919
0.010	-0.17406	-0.1733987	-0.17357099	-0.173572	-0.17357219
0.015	-0.37822	-0.37684161	-0.37691373	-0.376914	-0.37691389
0.020	-0.64013	-0.63791043	-0.63791404	-0.637915	-0.63791478
0.025	-0.93821	-0.93520713	-0.93531971	-0.935321	-0.93532096
0.030	-1.24796	-1.2443643	-1.2445334	-1.24453	-1.2445338
0.035	-1.54388	-1.5400425	-1.5400963	-1.54010	-1.5400969
0.040	-1.80162	-1.7980069	-1.7980175	-1.79802	-1.7980186
0.045	-1.99998	-1.9971139	-1.9972443	-1.99724	-1.9972448
0.050	-2.12263	-2.1210442	-2.1212031	-2.12120	-2.1212038
0.055	-2.15949	-2.1596399	-2.159675	-2.15968	-2.1596758
0.060	-2.10751	-2.1097377	-2.1097579	-2.10976	-2.1097585
0.065	-1.97098	-1.9754277	-1.9755731	-1.97557	-1.9755740
0.070	-1.76113	-1.7677184	-1.7678641	-1.76786	-1.7678649
0.075	-1.49523	-1.5036343	-1.5036551	-1.50366	-1.5036554
0.080	-1.19516	-1.2048205	-1.2048563	-1.20486	-1.2048572
0.085	-0.88560	-0.89576888	-0.89592858	-0.895930	-0.8959296
0.090	-0.59203	-0.60181021	-0.60194136	-0.601942	-0.60194151
0.095	-0.33860	-0.34703830	-0.34704933	-0.347050	-0.34705026
0.100	-0.14617	-0.15233507	-0.15238875	-0.152390	-0.15238997

**Table 5.2 Tip Deflection of the Three-Element Cantilever Beam  
( Newmark Method )**

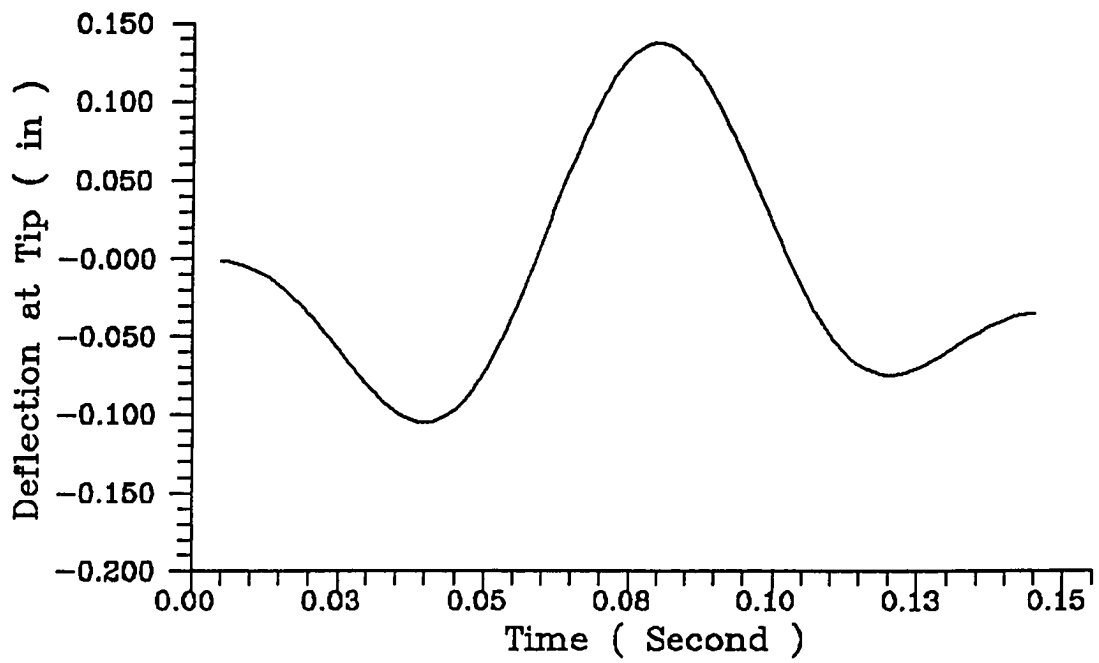
Time ( Second)	Full System Solution	1 Mode MDM	3 Modes MDM	1 Modes MAM
0.005	-0.0438102	-0.0436613	-0.0438087	-0.043794
0.010	-0.171206	-0.171115	-0.171206	-0.171203
0.015	-0.372078	-0.372054	-0.372077	-0.372142
0.020	-0.630405	-0.630232	-0.630405	-0.630320
0.025	-0.924807	-0.924771	-0.924806	-0.924859
0.03	-1.243193	-1.23186	-1.23193	-1.23194
0.035	-1.52681	-1.52665	-1.52681	-1.52674
0.040	-1.78533	-1.78533	-1.78533	-1.78541
0.045	-1.98709	-1.98696	-1.98709	-1.98705
0.050	-2.11536	-2.11525	-2.11536	-2.11534
0.055	-2.15983	-2.15982	-2.15983	-2.15991
0.060	-2.11724	-2.11707	-2.11724	-2.11716
0.065	-1.99050	-1.99045	-1.99050	-1.99053
0.070	-1.79026	-1.79021	-1.79026	-1.79026
0.075	-1.53269	-1.53253	-1.53269	-1.53262
0.080	-1.23826	-1.23825	-1.23826	-1.23826
0.085	-0.931285	-0.931168	-0.931284	-0.931283
0.090	-0.636242	-0.636115	-0.636240	-0.636202
0.095	-0.376952	-0.376947	-0.376952	-0.377034
0.100	-0.174783	-0.174768	-0.174781	-0.174709

Now, for the same cantilever beam, we alter the loading as  $Q(t) = 10 \sin(100t)$ . The purpose of this example is to illustrate the effectiveness of MMAM, which could not only improve the displacements, as the MAM does, but also improve the velocities and accelerations. The deflection at the tip is plotted in Fig. 5.2. Fig. 5.3 shows the displacement error norm for MDM and MAM (or, MMAM) with different numbers of modes used.

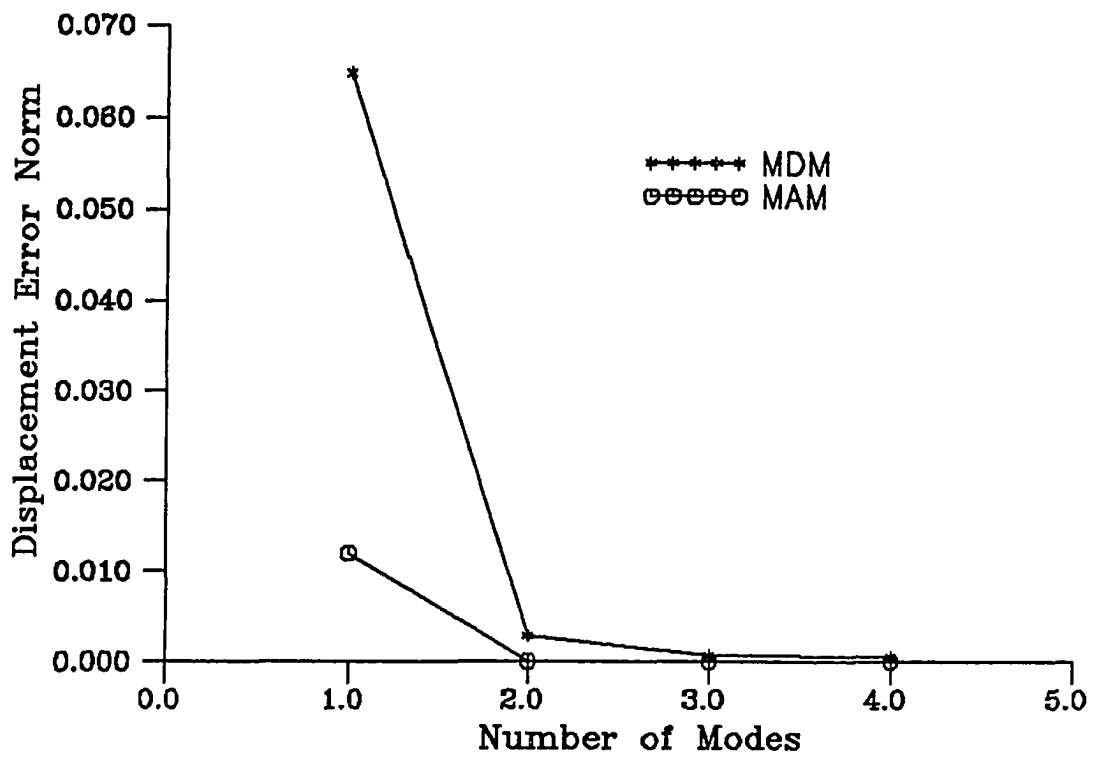
The velocity and acceleration error norms are presented in Fig. 5.4, which shows the effectiveness of MMAM in comparing with the MDM/MAM. The dynamic response error analysis results discussed above are also given in Table 5.3.

The displacement derivative error norm is shown in Fig. 5.5. It can be seen clearly that the MAM works well for improving the displacement derivatives. However, the improvement by MMAM on the derivatives of the velocity and accelerations is about 1.0%, which is not significant. This is because that the MMAM involves time derivatives of the loading function, which is difficult to apply with the DSA, i.e. we used MMAM for computing dynamic responses, and MAM for DSA. This inconsistency results in the unsatisfactory improvement of the DSA information for velocities and accelerations by the MMAM. The DSA error norm analysis results are also listed in Table 5.4.

The Error Norm analysis results for stresses and the stress derivatives are presented in Table 5.5, and plotted in Figs. 5.6 and 5.7. The effectiveness of the MAM in improving the stresses and stress derivatives are shown clearly.

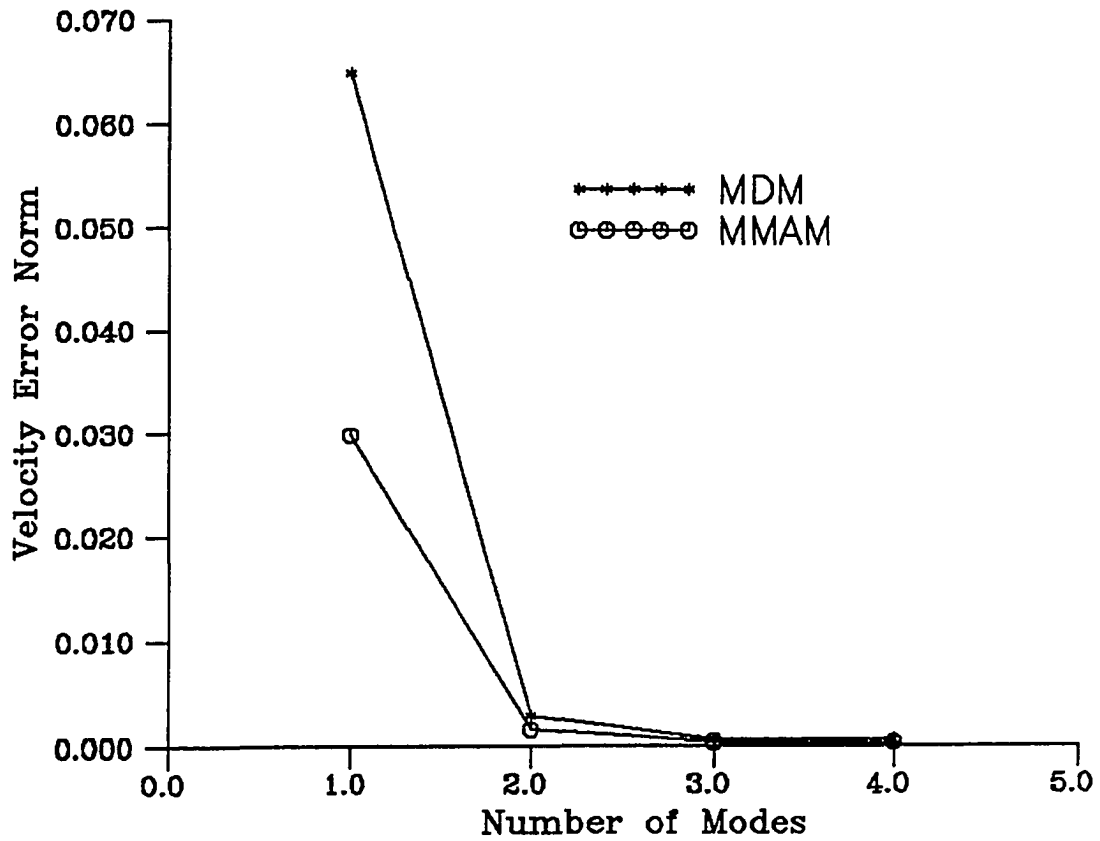


**Figure 5.2 Tip Deflections of the Three-Element Cantilever Beam  
(  $P = 10 \sin 100 t$  )**



**Figure 5.3 Displacement Error Norm of the Three-Element Cantilever Beam  
(  $P = 10 \sin 100 t$  )**



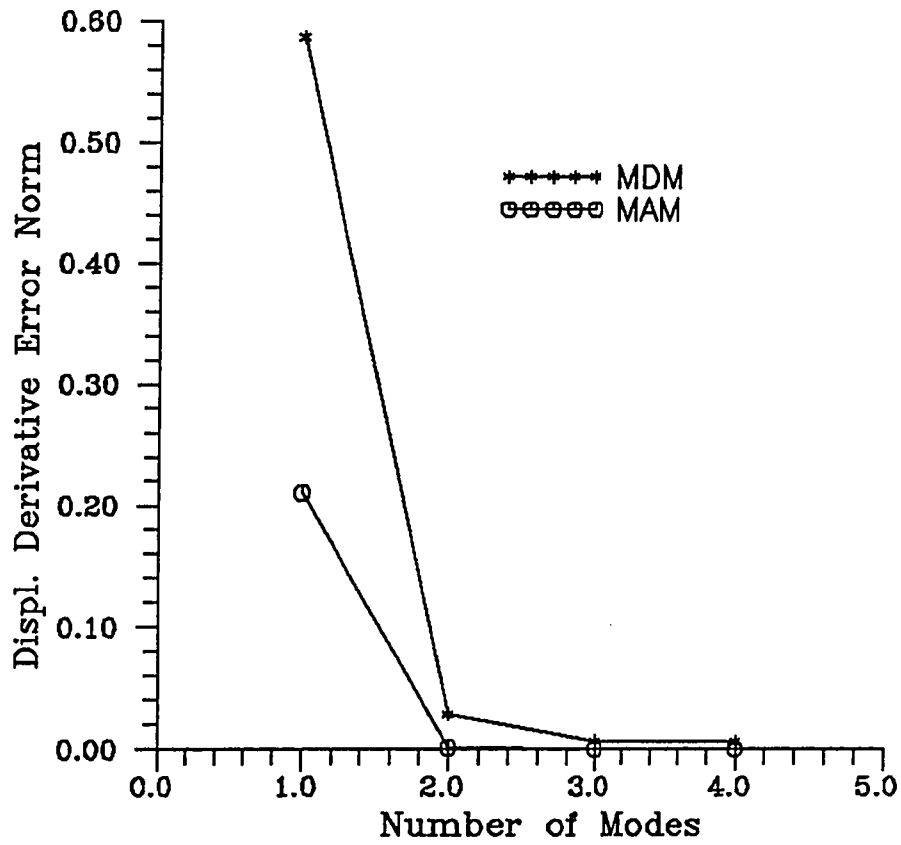


**Figure 5.4 Velocity Error Norm\* of the Three-Element Cantilever Beam  
(  $P = 10 \sin 100 t$  )**

\* The acceleration error norm is approximately equal to the velocity error norm.

**Table 5.3 Displacement, Velocity, and Acceleration Error Norm Analysis of the Three-Element Cantilever Beam  
(  $P = 10.0 \sin 100 t$  )**

Numbers of Modes	MDM		MAM		MMAM	
	Displ. Error Norm	Velo.& Accel. Error Norm	Displ. Error Norm	Velo.& Accel. Error Norm	Displ. Error Norm	Velo.& Accel. Error Norm
1	6.4825032E-02	6.4825032E-02	1.1939333E-02	6.4825032E-02	1.1939333E-02	2.970076E-02
2	2.8286435E-03	2.8286435E-03	6.1673865E-05	2.8286435E-03	6.1673865E-05	1.506388E-03
3	6.3388509E-04	6.3388509E-04	3.0787227E-06	6.3388509E-04	3.0787227E-06	3.424297E-04
4	5.5817412E-04	5.5817412E-04	2.2844017E-06	5.5817412E-04	2.2844017E-06	3.016930E-04
6	3.0000000E-06	3.0000000E-06	-----	3.0000000E-06	-----	-----



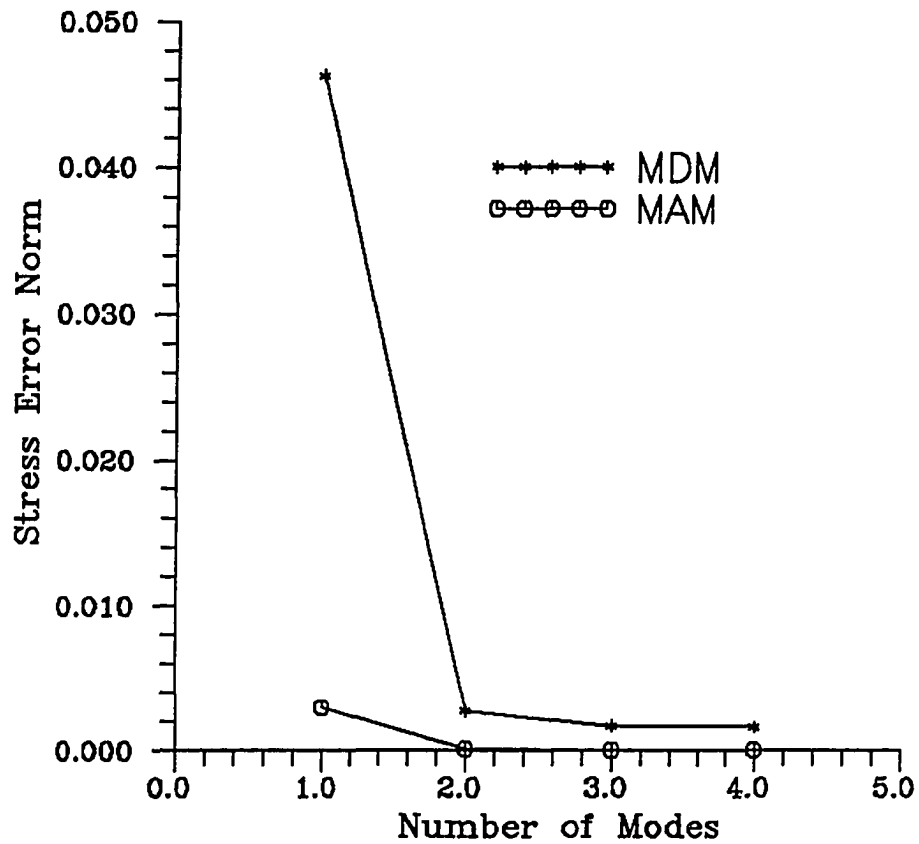
**Figure 5.5 Displacement Derivative Error Norm of the Three-Element Cantilever Beam (  $P = 10 \sin 100 t$  )**

**Table 5.4 DSA Error Norm Analysis of the Three-Element Cantilever Beam  
( P = 10 sin 100 t )**

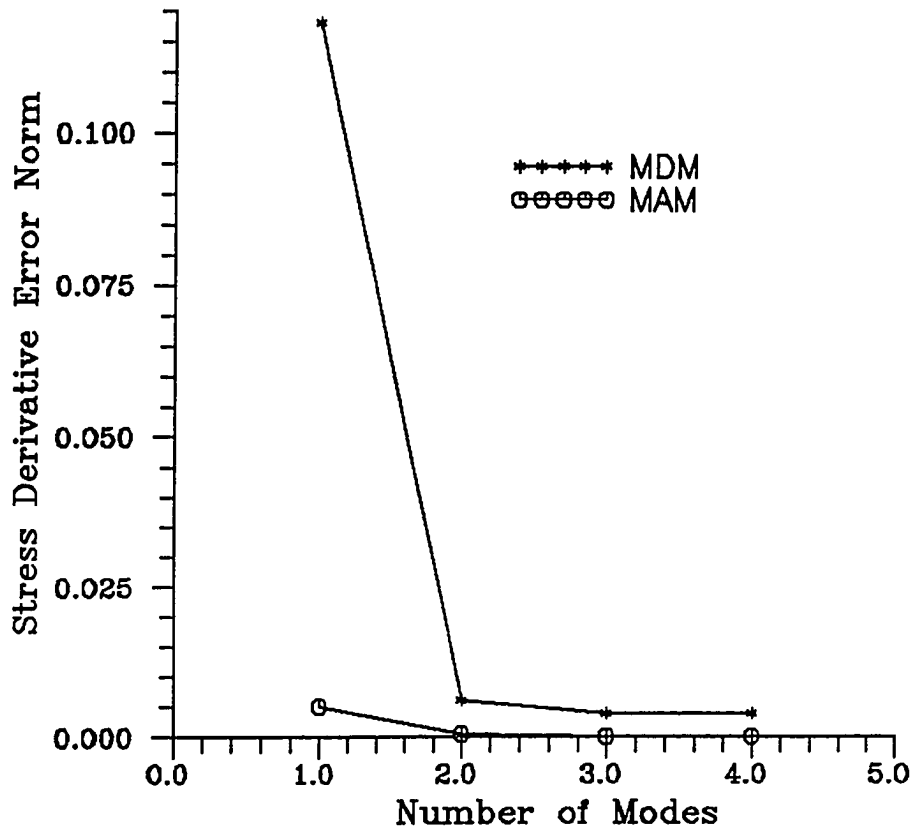
Number of Modes	MDM		MAM	
	Error Norm dZ/db	Error Norm d $\dot{Z}$ /db;d $\ddot{Z}$ /db	Error Norm dZ/db	Error Norm d $\dot{Z}$ /db;d $\ddot{Z}$ /db
1	5.8629844E-01	5.8626306E-01	2.1025986E-01	6.4825032E-02
2	2.8676460E-02	2.8686689E-02	1.3268246E-03	2.8286435E-03
3	6.5579695E-03	6.5587518E-03	3.2010480E-04	6.3388509E-04
4	5.7858866E-03	5.7829854E-03	2.2844017E-06	3.0426885E-04
6	-----	-----	-----	-----

**Table 5.5 Error Norm Analysis on Stresses and Stress Derivatives of the Three-Element Cantilever Beam  
(  $P = 10 \sin 100 t$  )**

Number of Modes	MDM		MAM	
	$\epsilon_{\sigma}$	$\epsilon \frac{d\sigma}{db}$	$\epsilon_{\sigma}$	$\epsilon \frac{d\sigma}{db}$
1	4.6251949E-02	1.1793927E-01	2.9993000E-03	5.0919000E-03
2	2.7495382E-03	6.0807772E-03	1.0481478E-04	4.7678016E-04
3	1.7129520E-03	3.8662389E-03	1.3262279E-05	6.7624600E-05
4	1.6269400E-03	3.8465666E-03	1.0318229E-05	6.4257013E-05



**Figure 5.6 Stress Error Norm of the Three-Element Cantilever Beam  
(  $P = 10 \sin 100 t$  )**



**Figure 5.7 Stress Derivative Error Norm of the Three-Element Cantilever Beam  
(  $P = 10 \sin 100 t$  )**

### 5.2.2 A 200-Element cantilever Beam

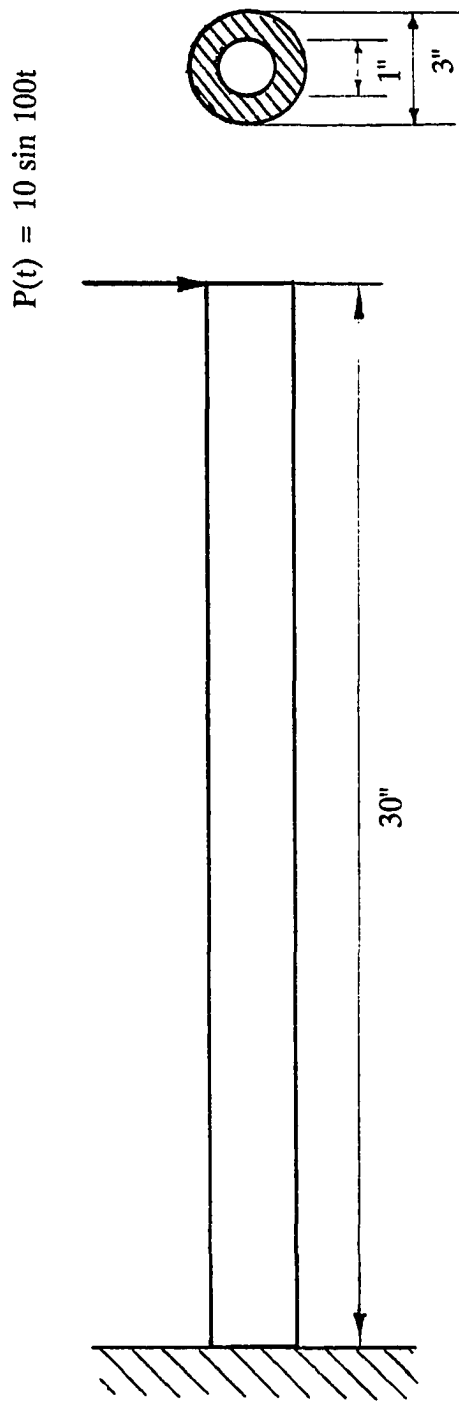
To illustrate also the computation efficiency, it is necessary to analyze larger size problems. In this section, a cantilever beam modeled with 200 elements is analyzed, which is shown in Fig. 5.8.

It can be seen from Table 5.6 that MAM with two modes could provide an accurate DSA result with only 0.18% error ( CPU time = 0.62 second); while it needs 12 modes to be applied for MDM to achieve an approximate same accuracy of 0.22% error for DSA information ( CPU time = 1.028 second ). The displacement error norm is plotted in Fig. 5.9, and the displacement derivative error norm is shown in Fig. 5.10. The MAM is very effective for this problem in both dynamic response analysis and DSA.

The lowest two frequencies are:  $\omega_1 = 54.2414$  Hz,  $\omega_2 = 459.927$  Hz. The corresponding eigenvalue derivatives are:  $\frac{d\lambda_1}{db} = 6466.5097547$ , and  $\frac{d\lambda_2}{db} = 2758407.695274$ .

The parallel-vector computing efficiency is presented in Table 5.7.





$E = 1.0 \times 10^6 \text{ psi}, \rho = 2.591 \times 10^{-4} \text{ lb-sec}^2/\text{in}^4, m = 0.02591 \text{ lb-sec}^2/\text{in}$

**Figure 5.8 A 200-Element Cantilever Beam**

**Table 5.6 Algorithm Accuracy and Efficiency Analysis with the 200-Element Cantilever Beam**

Number of modes	ERROR NORM ( MDM )				ERROR NORM ( MAM )			
	Z	dZ/db	Operation Counts	CPU Time ( Sec. )	Z	dZ/db	Operation Counts	CPU Time ( Sec. )
1	2.37908E-2	2.37982E-2	5.3075M	0.539734	4.61268E-4	8.44013E-3	5.4408M	0.56144
2	1.78398E-3	2.52691E-3	9.1297M	0.596162	7.85988E-6	1.77720E-3	9.28478M	0.61979
4	3.28058E-4	2.26059E-3	16.9044M	0.682011	-	-	-	-
6	3.66098E-5	2.25162E-3	24.8527M	0.784407	-	-	-	-
8	1.72651E-5	2.25161E-3	32.9747M	0.869184	-	-	-	-
10	1.00314E-5	2.25161E-3	41.2703M	0.940331	-	-	-	-
12	7.29572E-6	2.24220E-3	49.7397M	1.028230	-	-	-	-

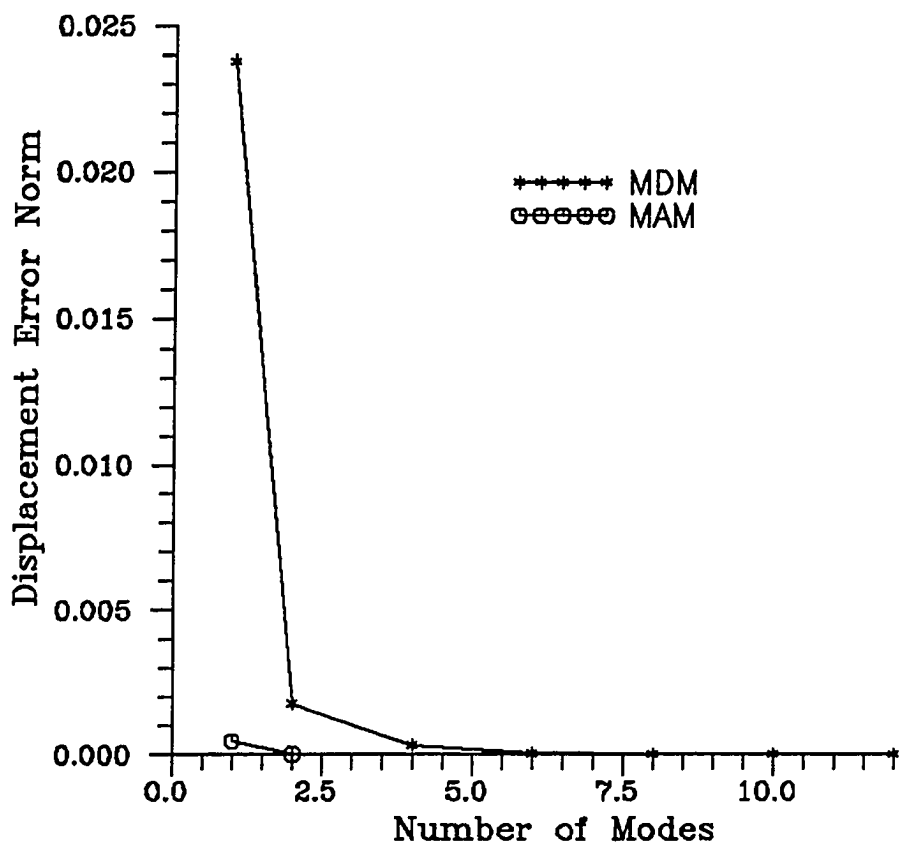
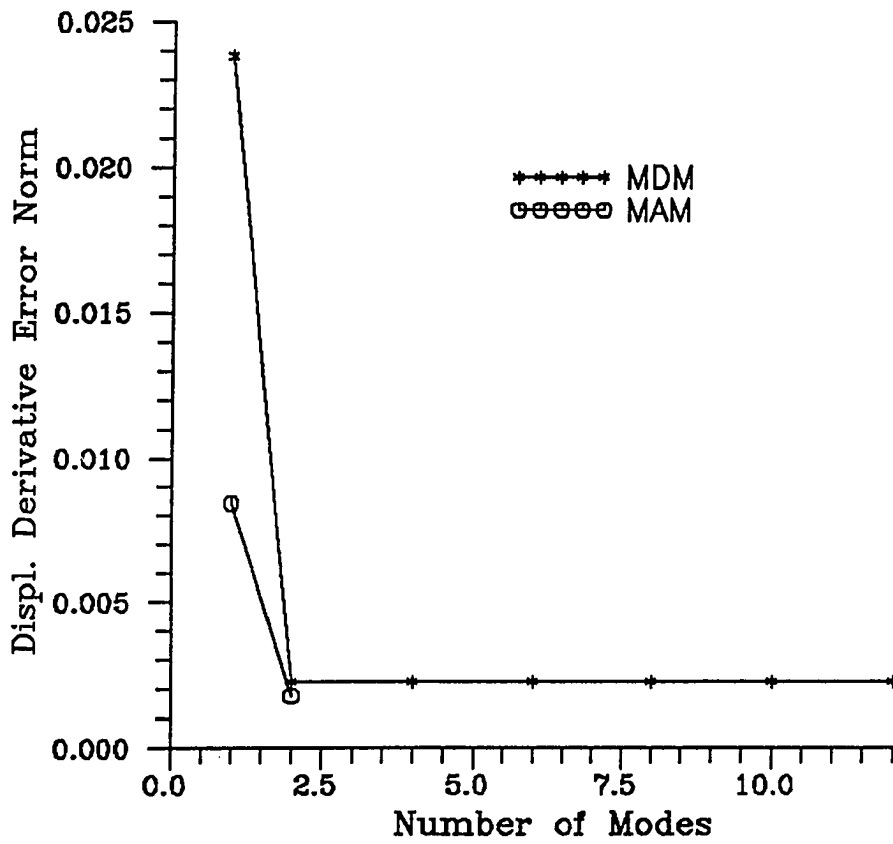


Figure 5.9 Displacement Error Norm of the 200-Element Cantilever Beam



**Figure 5.10 Displacement Derivative Error Norm of the 200-Element Cantilever Beam**

**Table 5.7 Parallel-vector Computation Efficiency  
with the 200-Element Cantilever Beam**

NP	Time (Second)	Efficiency	Speed Up
1	0.56144	100.00%	1.0
2	0.32689	85.88%	1.72

Total Number of Operation = 5.441 M

Computer Used: Cray Y-MP ( Sabre )

Time Measured By: Tsecond

### 5.3 Two-Dimensional Frame examples

Two two-dimensional frames are analyzed in this section.

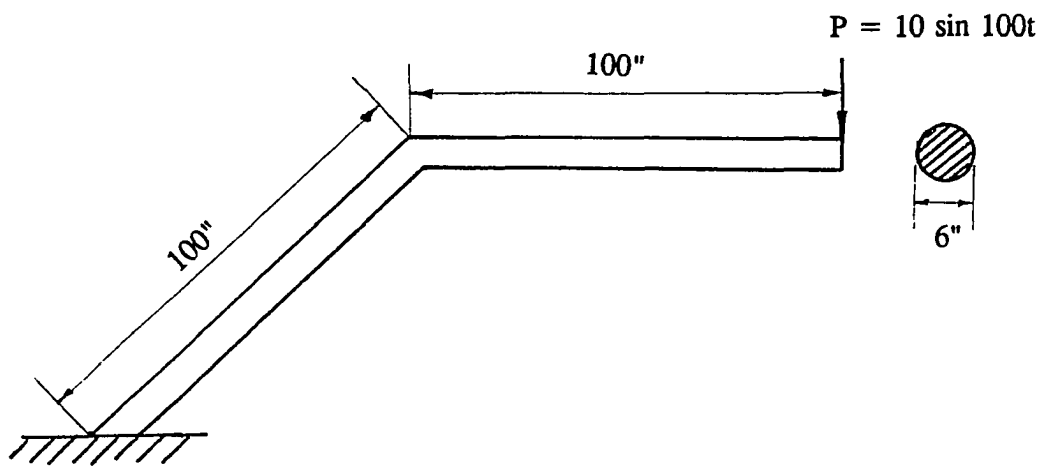
#### 5.3.1 A Simple Two-Dimensional Frame

A simple two-dimensional frame, as shown in Fig. 5.11, is modeled with eight finite elements.

Error norm analysis was conducted and the results are shown in Table 5.8 and Table 5.9 for dynamic response and DSA respectively. Fig. 5.12 shows the displacement error norm, Fig. 5.13 plots the velocity error norm. The error norm for displacement derivatives are shown in Fig. 5.14. Results presented in this example show that the MAM is a very effective method, and the proposed alternative formulation yields excellent accuracy.

The error norms for stresses and the stress derivatives are listed in Table 5.10, are plotted in Fig. 5.15 and Fig. 5.16, respectively. It is illustrated clearly that the MAM improves the accuracy of stresses and stress derivatives significantly through the improvement of the displacements.

The lowest four eigenvalues and their corresponding derivatives with respect to the design variable  $r$ , which is the radius of the circular cross-section, are presented in Table 5.11.



$$E = 10.0 \times 10^6 \text{ psi}, \quad \rho = 4.0 \times 10^{-5} \text{ lb-sec}^2/\text{in}^4$$

**Figure 5.11 A Simple Two-dimensional Frame**

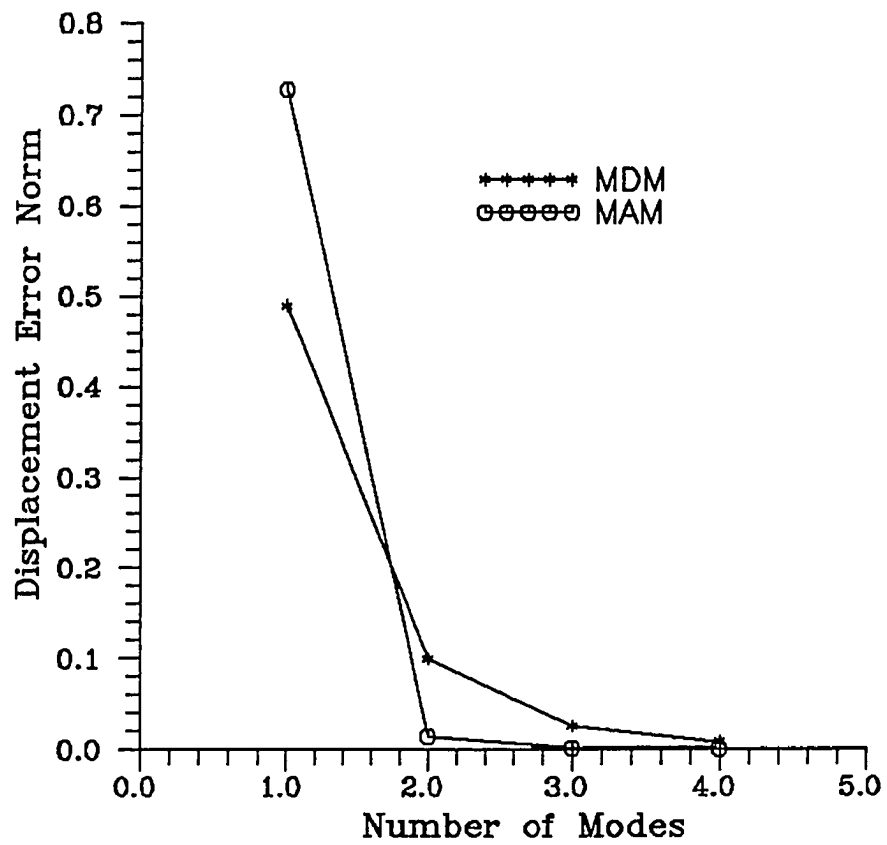
**Table 5.8 Dynamic Response Error Norm Analysis of the Simple Two-Dimensional Frame**

Numbers of Modes	MDM		MAM		MMAM	
	Z Error Norm	$\dot{z}$ Error Norm	Z Error Norm	$\dot{z}$ Error Norm	Z Error Norm	$\dot{z}$ Error Norm
1	0.4892	0.4892	0.7271	0.4892	0.7271	0.0865
2	0.0990	0.0990	0.01354	0.0990	0.01354	0.04761
3	0.02558	0.02558	0.00131	0.02558	0.00131	0.01329
4	0.007662	0.007662	0.000124	0.007662	0.000124	0.0041

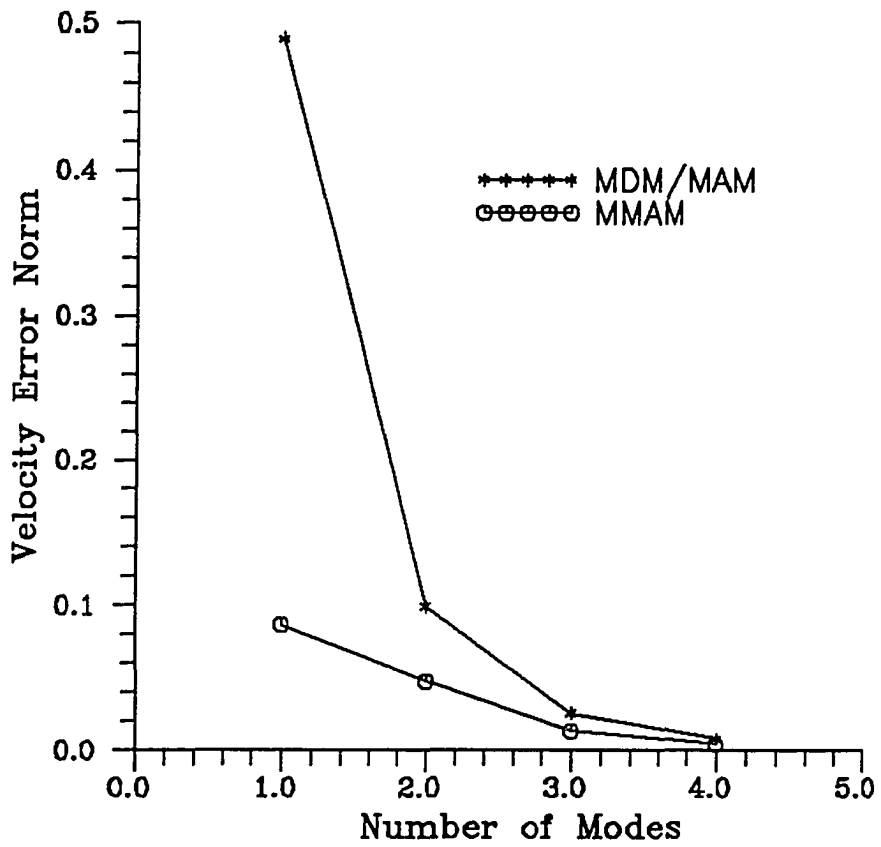


**Table 5.9 DSA Error Norm Analysis of the Simple Two-Dimensional Frame**

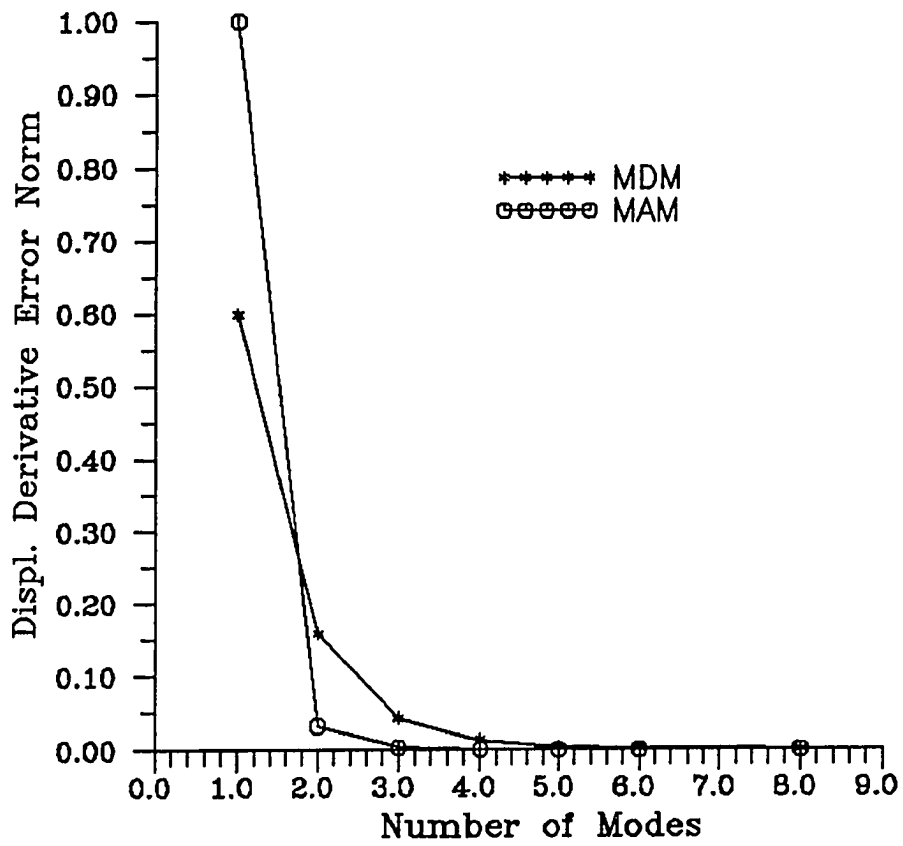
Numbers of modes	MDM		MAM		MMAM	
	Error Norm $dZ/db$	Error Norm $d\dot{Z}/db = d\ddot{Z}/db$	Error Norm $dZ/db$	Error Norm $d\dot{Z}/db = d\ddot{Z}/db$	Error Norm $dZ/db$	Error Norm $d\dot{Z}/db = d\ddot{Z}/db$
1	0.5987426	0.5987426	1.000000	0.5987425	1.000000	0.5987425
2	0.1595410	0.1595410	0.0320652	0.1595410	0.0320652	0.1595410
3	0.0427056	0.0427056	0.0032651	0.0427010	0.0032651	0.0427010
4	0.0131046	0.0131046	0.0003120	0.0130907	0.0003120	0.0130907
5	0.0037604	0.0037604	0.0000500	0.0037607	0.0000500	0.0037607
6	0.0018047	0.0018047	0.0000108	0.0017992	0.0000108	0.0017992
8	0.0006504	0.0006504	0.0000016474	0.00064365	0.00000146	0.00064395



**Figure 5.12 Displacement Error Norm of the Simple Two-Dimensional Frame**



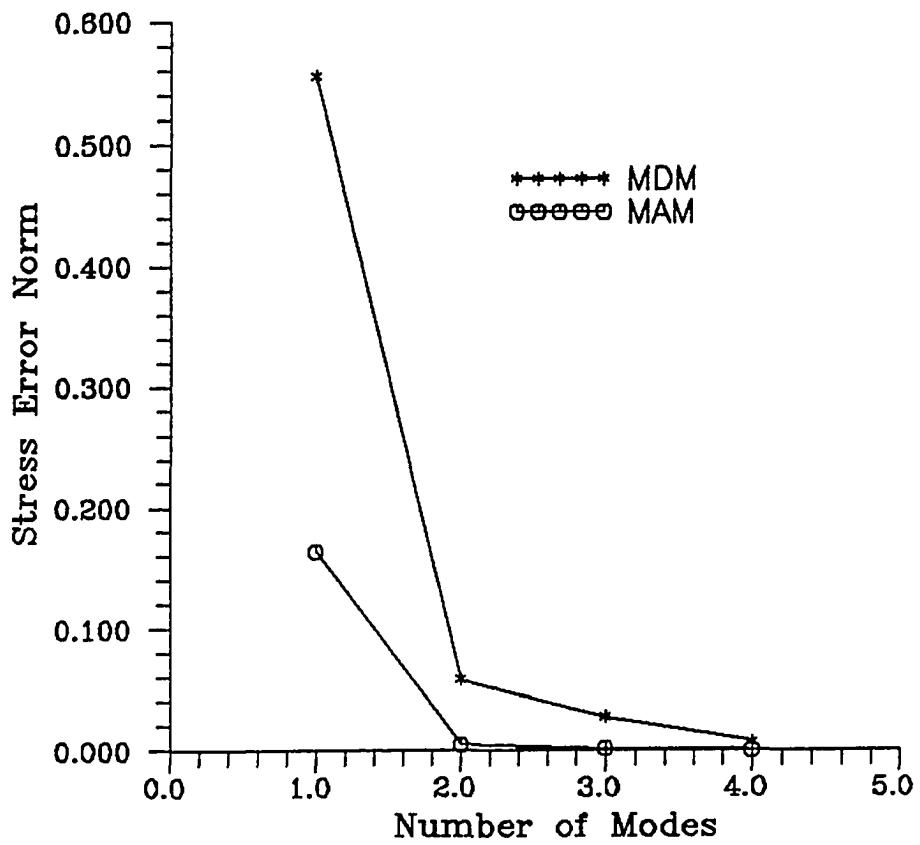
**Figure 5.13** Velocity Error Norm of the Simple Two-Dimensional Frame



**Figure 5.14 Displacement Derivative Error Norm of the Simple Two-Dimensional Frame**

**Table 5.10 Stress and Stress Derivative Error Norm Analysis of the Simple Two-Dimensional Frame**

Number of Modes	MDM		MAM	
	$\epsilon_{\sigma}$	$\epsilon_{\frac{d\sigma}{db}}$	$\epsilon_{\sigma}$	$\epsilon_{\frac{d\sigma}{db}}$
1	5.55209651E-01	1.26152981E+00	1.63996264E-01	3.70340746E-01
2	5.88482531E-02	1.40313660E-01	4.87675678E-03	3.53684330E-02
3	2.77103538E-02	5.59812578E-02	1.65608339E-03	3.34960236E-02
4	6.95740350E-03	3.75760283E-02	1.82951845E-04	3.25964212E-02



**Figure 5.15** Stress Error Norm of the Simple Two-Dimensional Frame

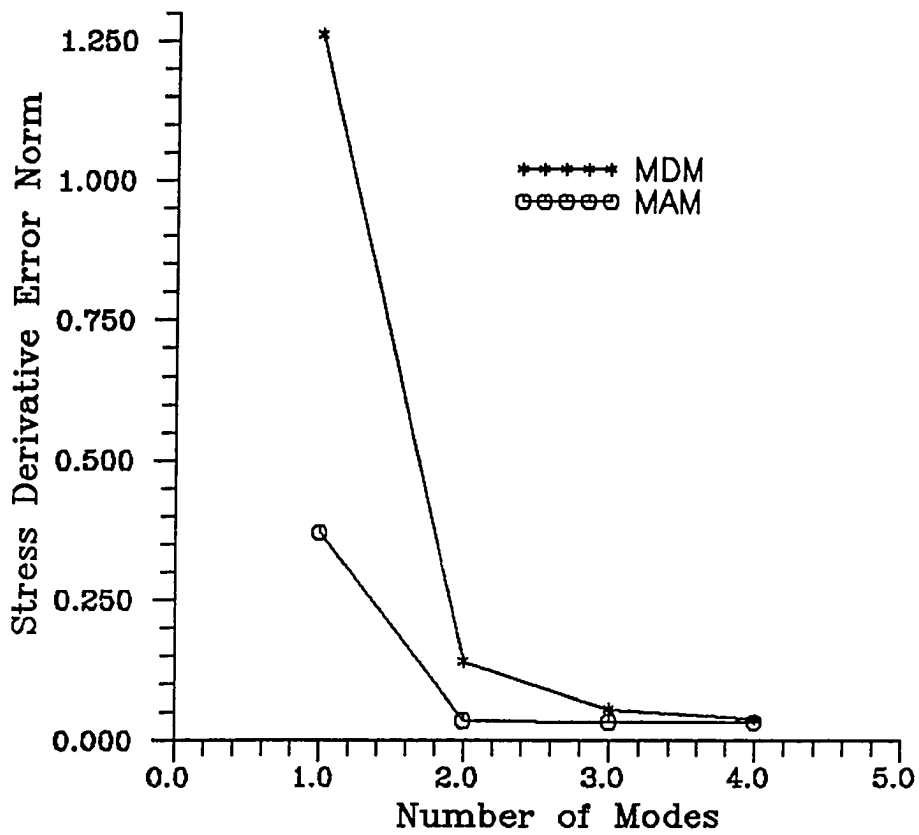


Figure 5.16 Stress Derivative Error Norm of the Simple Two-Dimensional Frame

**Table 5.11 Eigenvalues and Eigenvalue Derivatives  
of the Simple Two-Dimensional Frame**

Number	Eigenvalue	Frequency ( Hz )
1	4869.91206	11.1066
2	101982.961	50.8258
3	1106121.49	167.3871
4	9358851.94	273.4389

$\frac{d\lambda_1}{db} = 324.55517,$	$\frac{d\lambda_2}{db} = 67810.3844,$
$\frac{d\lambda_3}{db} = 732854.7861,$	$\frac{d\lambda_4}{db} = 1911436.6938$



### 5.3.2 An 18-Bay 25-Story Two-Dimensional Frame

An 18-bay 25-story two-dimensional frame, shown in Fig. 5.17 is analyzed in this section. This frame has 494 nodes, consists of 925 beam elements. Each nodes has three DOF. Both dynamic response analysis and DSA are carried out, and the error norm analysis is conducted.

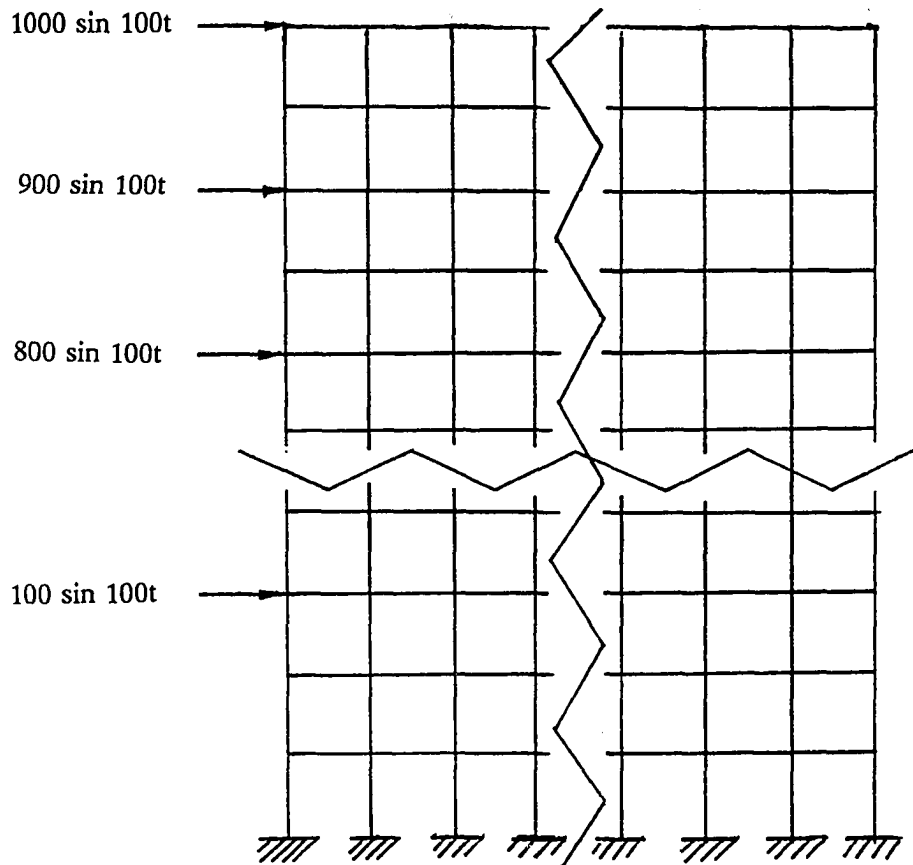
Error norms for displacements and displacement derivatives are presented in Table 5.12a. When one mode is used to approximate the response, the error norm associated with MAM is larger than the one by MDM. If more than one mode are used, the MAM yields much better accuracy. For instance, MAM with four modes gives only 0.19% error for displacement vector, while, MDM with 20 modes produces 1.79% error. The displacements and displacement derivatives error norms are further plotted in Fig. 5.18a and Fig. 5.19 respectively. Table 5.12b shows the velocity and acceleration error norms. Table 5.12b illustrates that MMAM significantly improves the velocity and acceleration vectors. The velocity and acceleration error norm are also plotted in Fig. 5.18b.

The stress and stress derivative error norms are presented in Table 5.13, and plotted in Figs. 5.20 and 5.21.

To illustrate the efficiency of MAM versus MDM, the CPU time for both cases are recorded, which are shown in Table 5.14, and plotted in Fig. 5.22. It can be seen clearly that the MAM gives much better accuracy with fewer modes used, and it also significantly reduces the computation time.

The lowest 20 eigenvalues and the eigenvalue derivatives of the lowest eigenvalue are presented in Table 5.15.

The parallel-vector computation efficiency is presented in Table 5.16, which is conducted in Cray Y-MP high performance computer. The wall-clock time measured using Timef for one processor is close to the time measured by Tsecond: for example, 2.336 seconds by using Tsecond, while 2.355 seconds by using Timef. When multi-processors are request, the CPU time recorded by using Tsecond is roughly the same for different runs. For instance, 1.227 seconds, 1.333 seconds, 1.367 seconds, for using two processors; 0.8909 seconds, 0.9334 seconds, 0.9402 seconds for three processors. However, the CPU time recorded by using Timef varies a lot for different runs. For example, when two processors are used, with different three runs, the time recorded are 1.80 seconds, 1.9534 seconds, 2.3278 seconds; when three processors are request, 1.784 seconds, 2.1092 seconds, and 3.3782 seconds. Thus, using Timef to measure the CPU time will not be accurate unless under dedicate computing environment. However, it is expected that in the truly dedicated computing environment, using either Tsecond or Timef should give about the same CPU time. The computational time by using Newmark method and the Duhamel integral method is roughly the same. For instance, when 20 modes are applied, in Cray 2 (Voyager), the Newmark method takes 3.5345 seconds, while the Duhamel integral takes 3.5942 seconds. And their parallel performance is about the same.



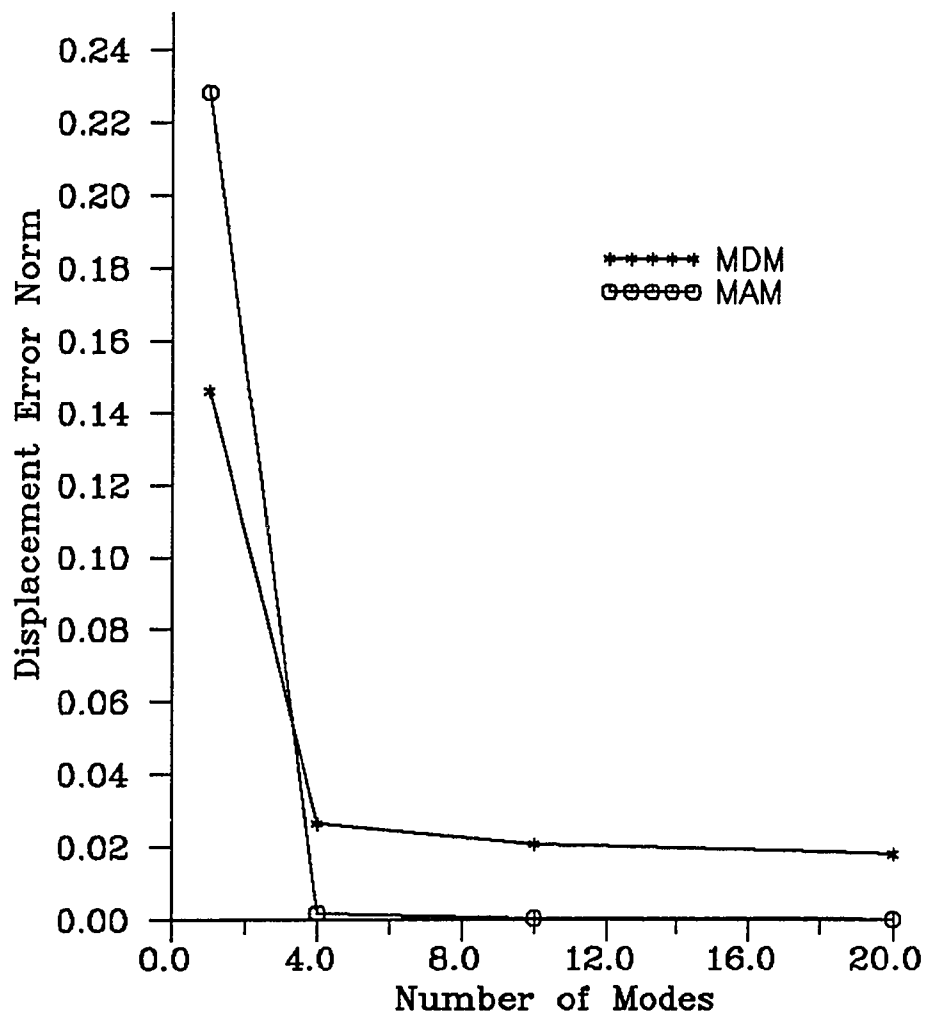
**Figure 5.17** An 18-Bay 25-Story Two-Dimensional Frame  
 (  $P(t) = 1000 \sin 100 t$  )

**Table 5.12a Displacement and Displacement Derivative Error Norm Analysis of the 18-Bay 25-Story Two-Dimensional Frame ( 494 nodes, 925 elements, 1425 DOF )**

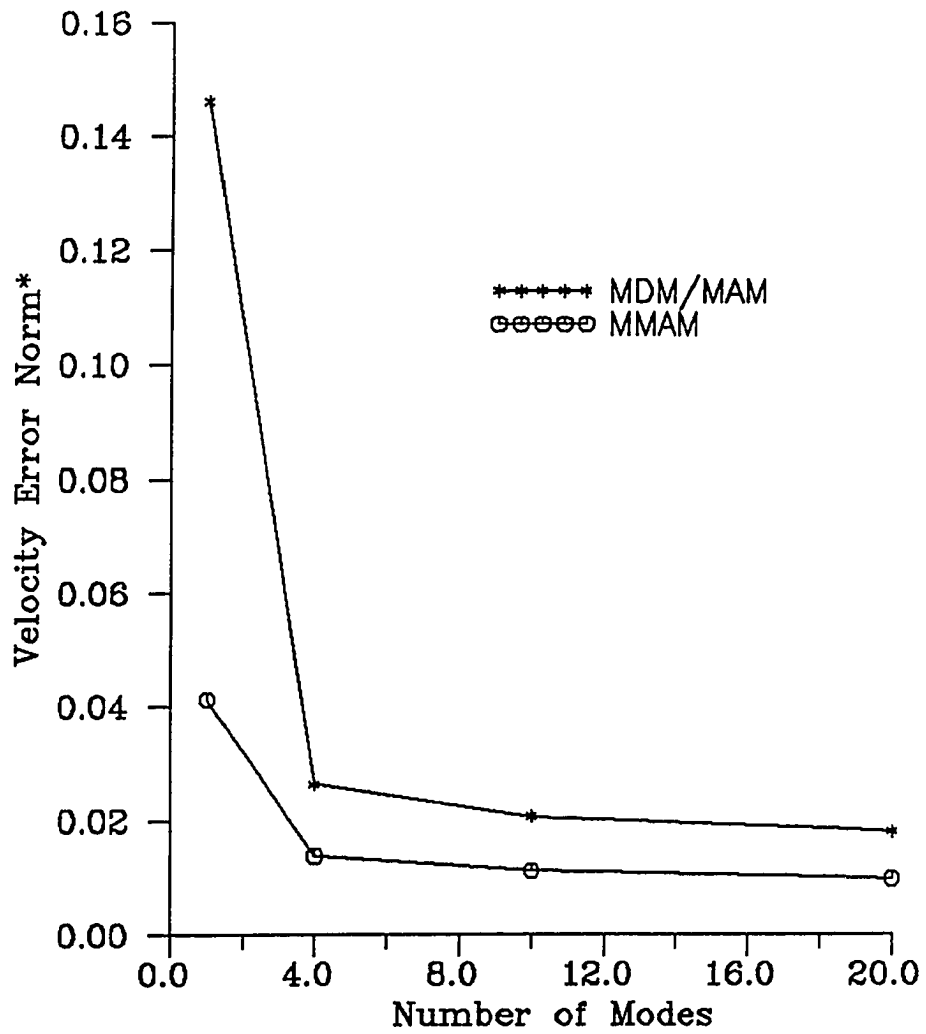
Number of Modes	MDM		MAM	
	$\epsilon_z$	$\epsilon_{dz/db}$	$\epsilon_z$	$\epsilon_{dz/db}$
1	1.46015E-01	1.48038E-01	2.28155E-01	5.07135E-01
4	2.63286E-02	2.63481E-02	1.85422E-03	2.66839E-03
10	2.06270E-02	2.07616E-02	3.30111E-04	4.57390E-04
20	1.79353E-02	1.81785E-02	1.39492E-04	1.99745E-04

**Table 5.12b Velocity and Acceleration Error Norm Analysis of the 18-Bay 25-Story Two-Dimensional Frame ( 494 nodes, 925 elements, 1425 DOF )**

Number of Modes	Velo. & Accel. Error Norm	
	MDM & MAM	MMAM
1	1.46015E-01	4.11475E-02
4	2.63286E-02	1.36999E-02
10	2.06270E-02	1.10691E-02
20	1.79353E-02	9.66767E-03



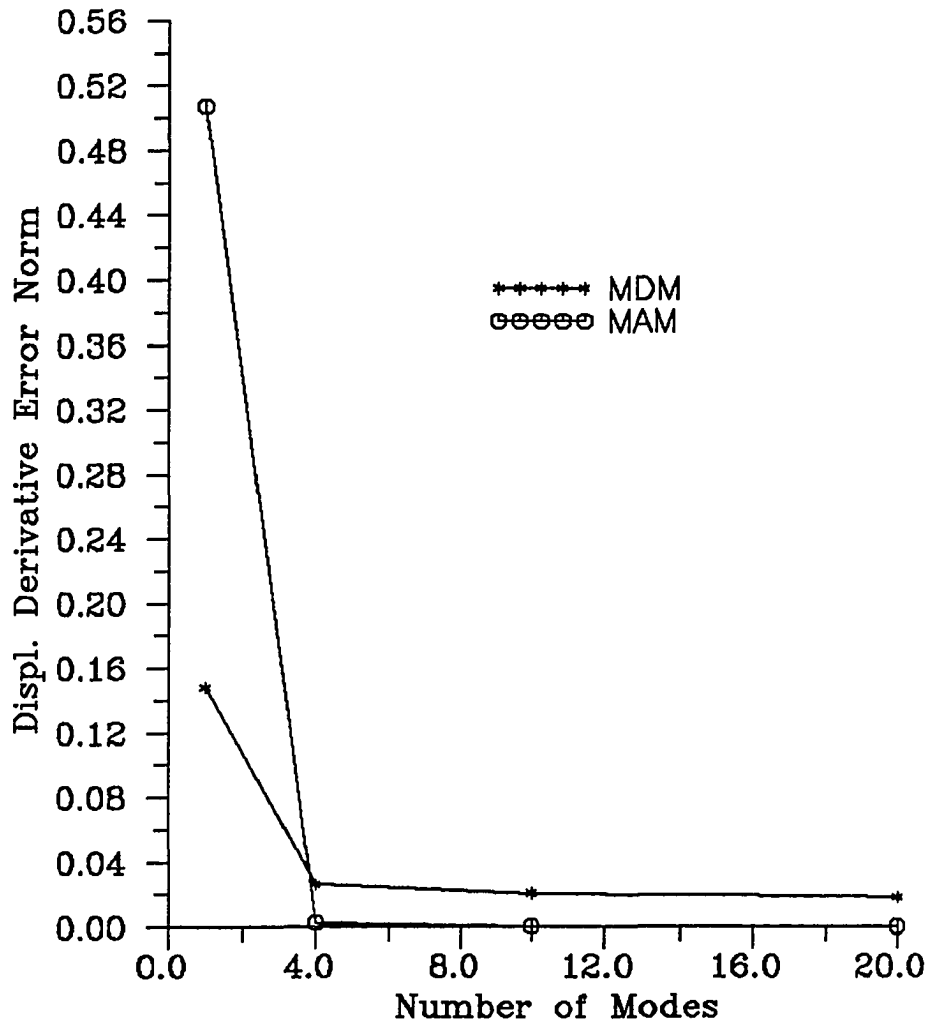
**Figure 5.18a Displacement Error Nor of the 18-Bay 25-Story Two-Dimensional Frame**



**Figure 5.18b Velocity & Acceleration Error Norm  
of the 18-Bay 25-Story Two-Dimensional Frame**

---

\* Velocity Error norm and the acceleration error norm are the same.



**Figure 5.19 Displacement Derivative Error Norm of the 18-Bay 25-Story Two-Dimensional Frame**

**Table 5.13 Stress and Stress Derivative Error Norm Analysis of the 18-Bay 25-Story Two-Dimensional Frame ( 494 nodes, 925 elements, 1425 DOF )**

Number of Modes	MDM		MAM	
	$\epsilon_{\sigma}$	$\epsilon \frac{d\sigma}{db}$	$\epsilon_{\sigma}$	$\epsilon \frac{d\sigma}{db}$
1	3.22185E-01	5.07589E-01	3.38276E-01	6.03404E-01
4	1.14610E-01	2.67104E-01	4.44931E-03	1.03542E-02
10	1.07666E-01	2.44914E-01	1.73338E-03	4.15386E-03
20	9.76454E-02	2.11302E-01	8.51913E-03	2.52437E-03



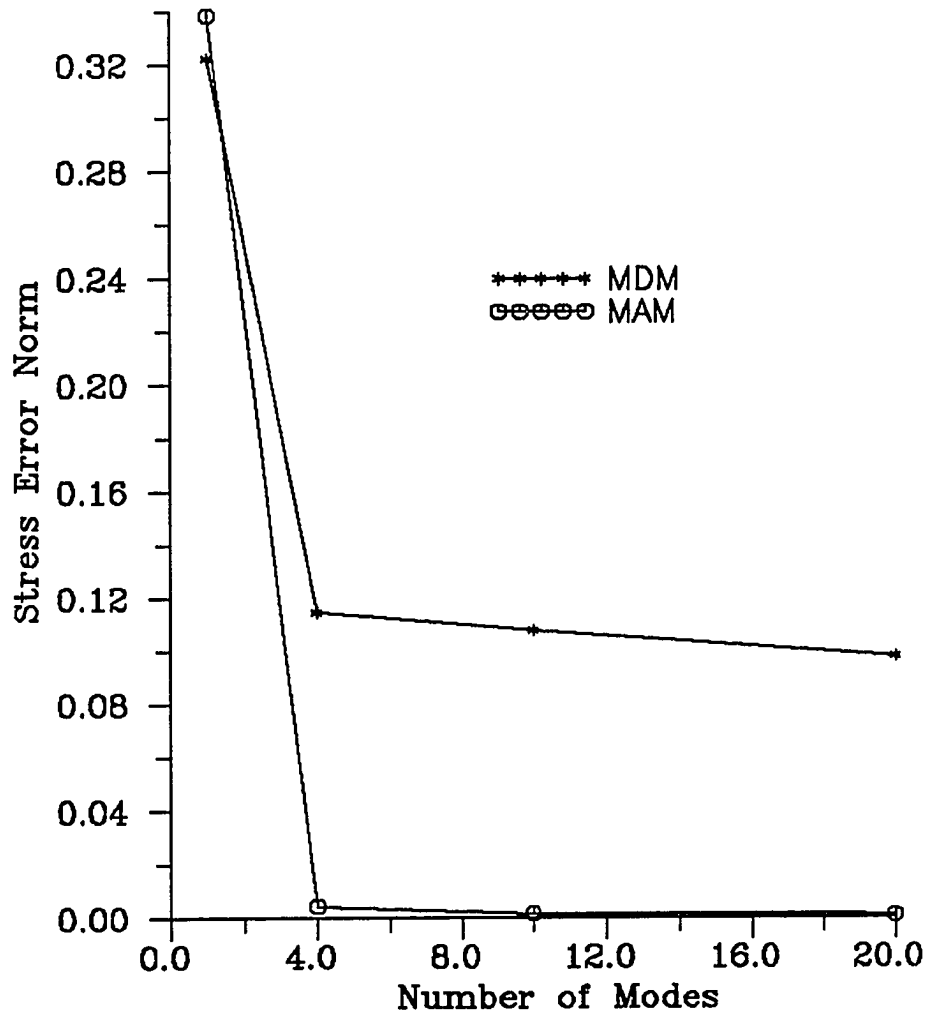
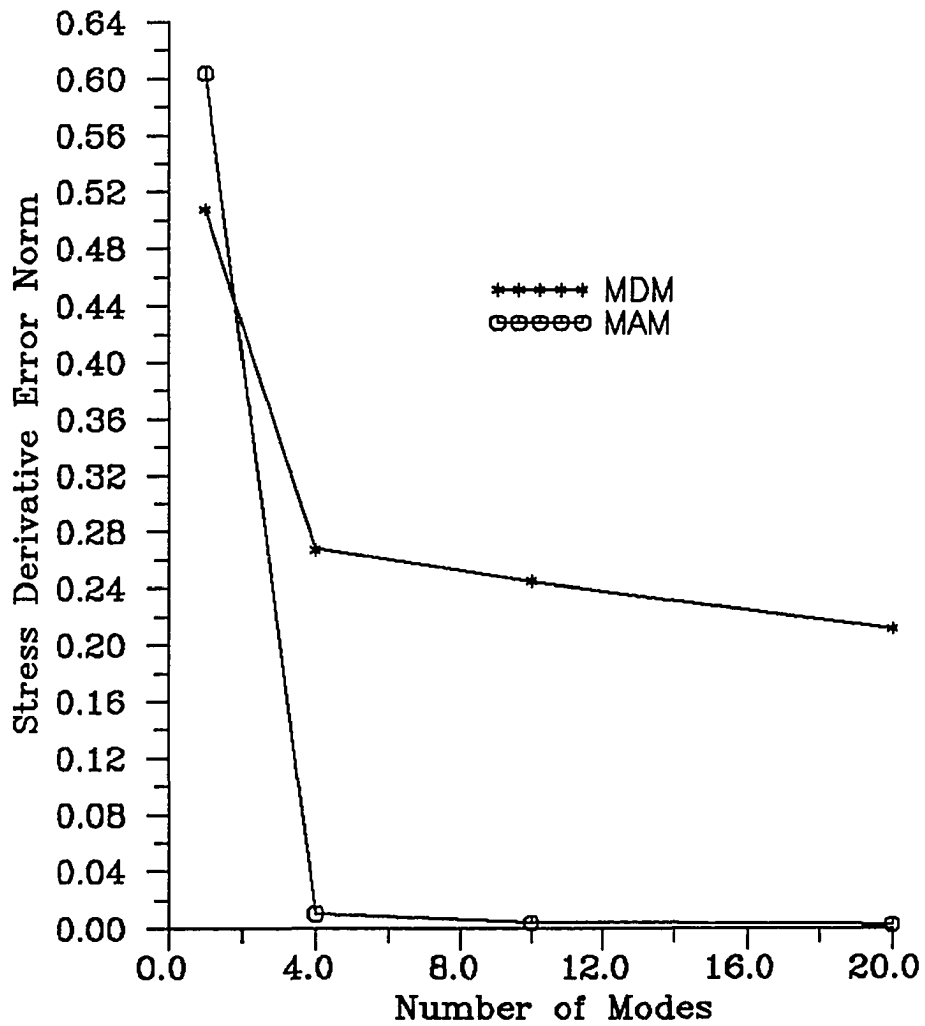


Figure 5.20 Stress Error Norm of the 18-Bay 25-Story Two-Dimensional Frame

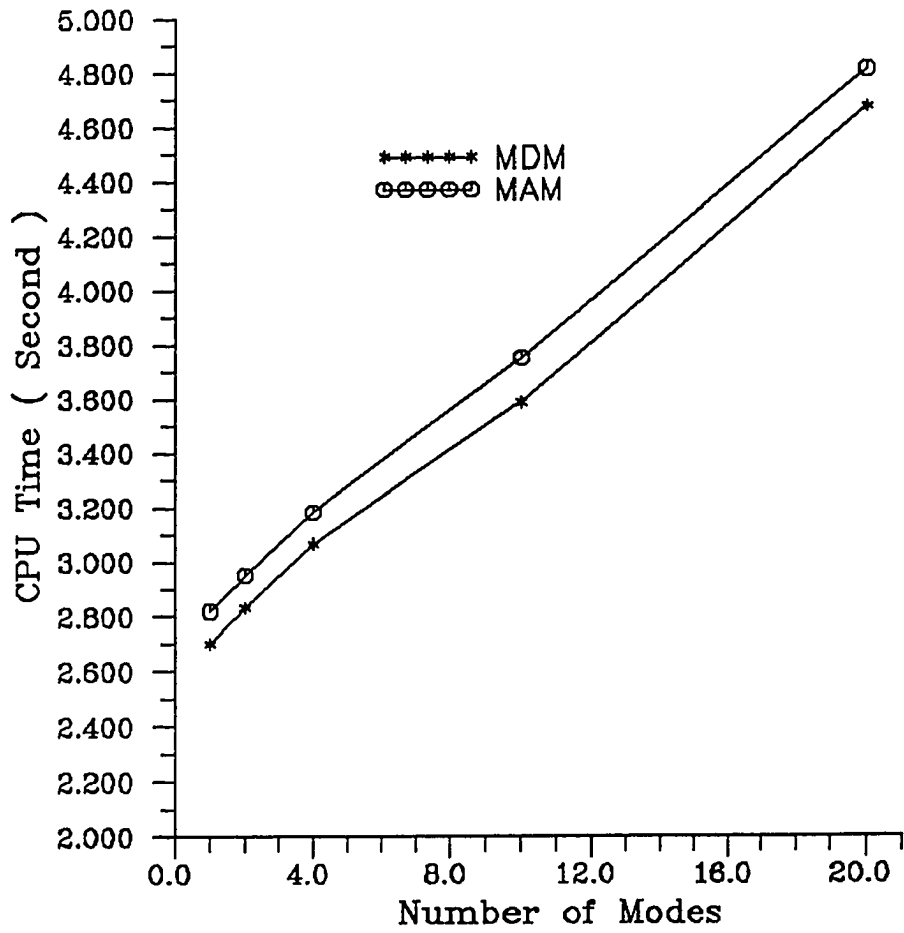


**Figure 5.21** Stress Derivative Error Norm of the 18-Bay 25-Story Two-Dimensional Frame

**Table 5.14 CPU Time Comparison of MAM Versus MDM  
with the 18-Bay 25-Story Two-Dimensional Frame  
( 494 nodes, 925 elements, 1425 DOF )**

Computer used: Cray-2 (Voyager )

Number of Modes	CPU Time ( Tsecond )	
	MDM	MAM
1	2.6992884	2.8205487
2	2.8333591	2.9509871
4	3.0656696	3.1828010
10	3.5871777	3.7515778
20	4.6694767	4.8081796



**Figure 5.22 CPU Time Comparison of MAM Versus MDM with the 18-Bay 25-Story Two-Dimensional Frame**

**Table 5.15 Eigenvalues and Eigenvalue Derivatives  
of the 18-Bay 25-Story Two-Dimensional Frame**

Mode Number	Eigen-frequency ( Hz )
1	0.147601E+02
2	0.478619E+02
3	0.906611E+02
4	0.129870E+03
5	0.144571E+03
6	0.169977E+03
7	0.204021E+03
8	0.222585E+03
9	0.252227E+03
10	0.290483E+03
11	0.328321E+03
12	0.344924E+03
13	0.366085E+03
14	0.402779E+03
15	0.432956E+03
16	0.438629E+03
17	0.461602E+03
18	0.475123E+03
19	0.492173E+03
20	0.510032E+03

$\frac{d\lambda_1}{db_1} = 2575.45135$
$\frac{d\lambda_2}{db_1} = 34280.5312$

**Table 5.16 Parallel-Vector Computation Efficiency  
of the 18-Bay 25-Story Two-Dimensional Frame**

NP	Time (Second)	Efficiency	Speed-up
1	2.3358	100%	1.00
2	1.2278	95.12%	1.90
3	0.8909	87.39%	2.62

Compiled in Cray Y-MP (Saber, NASA Langly R.C.), run at Cray Y-MP  
(Reynolds, NASA, Ames);  
Time measured by: Tsecond

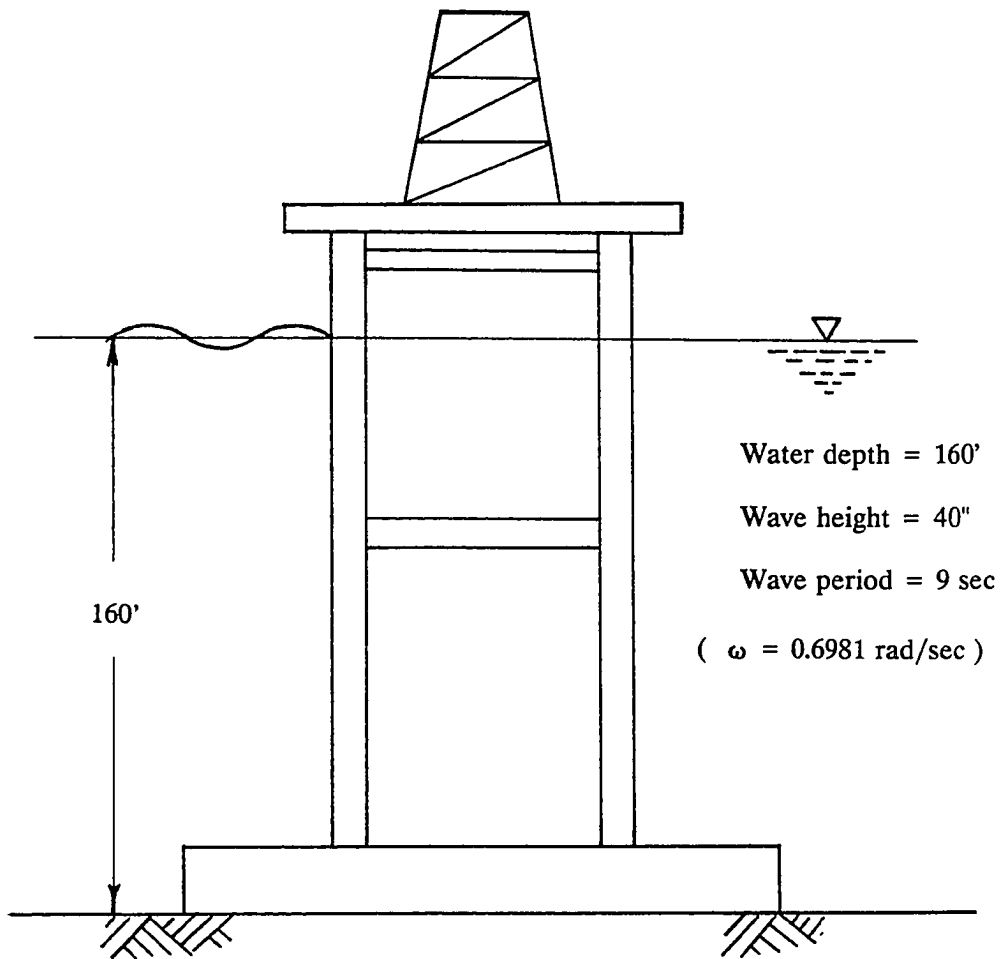
## 5.4 Three-Dimensional Frame Examples

In this section, a flexible offshore structure, a typical three-dimensional frame, and a CSI design [57] model are studied.

### 5.4.1 A Flexible Offshore Structure

A flexible offshore platform shown in Fig. 5.23 is modeled as a three-dimensional frame with 48 DOF [50], as shown in Fig. 5.24.

Table 5.17 shows error norms of the displacements, velocities for MDM and MAM with different numbers of modes applied. This structure is of 48 DOF, when 10 modes are applied, the error norms are 100% for both MDM and MAM. While when 12 modes are used, the MDM yields a error norm of 71.0%, and the MAM with a error norm of 20.8%, for the displacements. The error norm for DSA information are presented in Tables 5.18 and 5.19. It can be seen that 12 to 14 modes are needed to obtain a acceptable accuracy of the DSA information. The error norm information is also plotted in Fig. 5.25, Fig. 5.26, and Fig. 5.27. The lowest 20 eigenvalues and the eigenvalue derivatives of the two lowest eigenvalues are listed in Table 5.20. It should be noticed that the eigen-frequency has a big jump (from 0.404Hz to 99.547Hz) from the 11th lower frequency to the 12th one, which represents the change from bending to axial modes of vibration. This is due to insufficient elements used in the finite element model. Increasing number of elements for the vertical members would eliminate this kind of phenomenon.



Four faces of the structure are identical

Vertical members:  $D = 4''$ ,  $t = 1.5''$

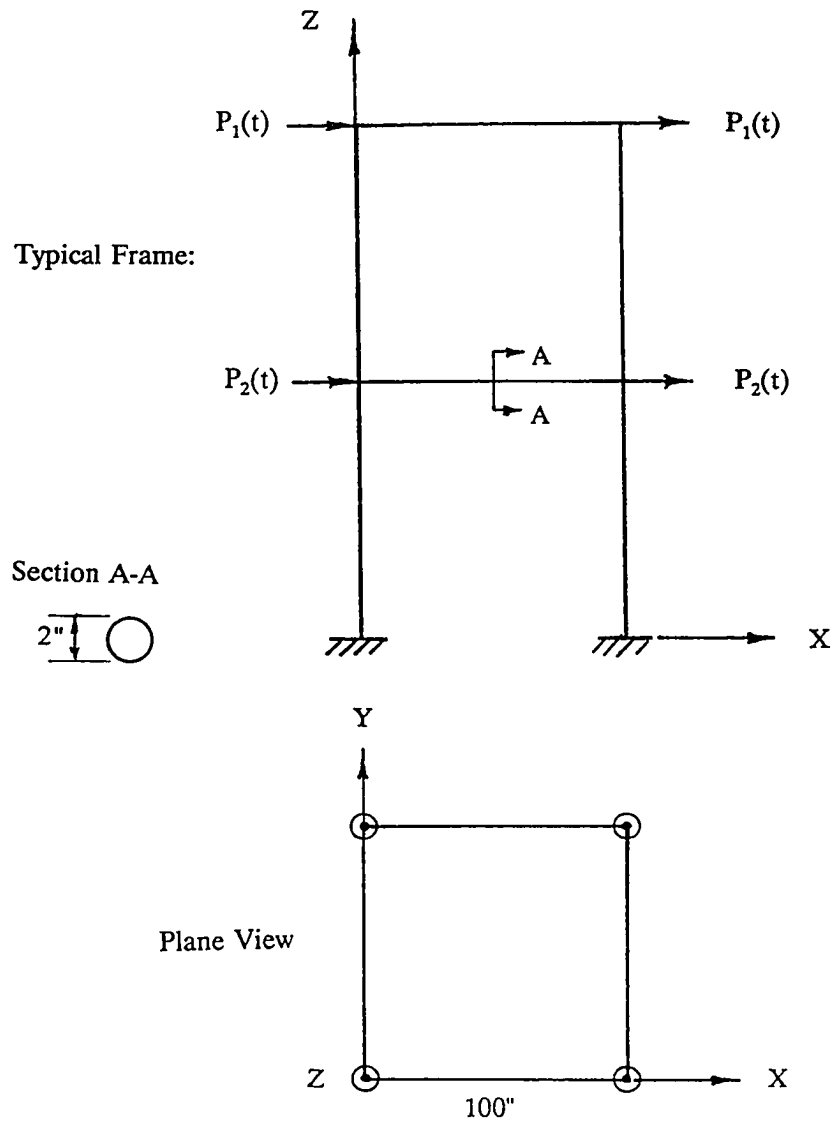
Horizontal members:  $D = 2''$ ,  $t = 0.5''$

$E = 29000$ ksi

Deck weight = 2000 kips ( Asymmetric )

**Figure 5.23 A Flexible Steel Offshore Structure**





$$P_1(t) = 59.5 \sin(-\omega t + 2.0) \text{ k}$$

$$P_2(t) = 19.3 \sin(-\omega t + 2.0) \text{ k}$$

**Figure 5.24 A Simplified Model of the Flexible Offshore Structure**

**Table 5.17 Dynamic Response Error Norm Analysis of the Offshore Steel Structure**

Numbers of Modes	MDM		MAM		MMAM	
	Error Norm of Displ.	Error Norm of Veloc.	Error Norm of Displ.	Error Norm of Veloc.	Error Norm of Displ.	Error Norm of Veloc.
10	1.0	1.0	1.0	1.0	1.0	1.0
12	7.07039E-01	7.07039E-01	2.08538E-01	7.07039E-01	2.08538E-01	7.06639E-01
14	2.67366E-03	2.67366E-03	1.26325E-03	2.67366E-03	1.26325E-03	2.67255E-03
16	2.67365E-03	2.67365E-03	1.26325E-03	2.67365E-03	1.26325E-03	2.67255E-03
18	1.75654E-04	1.75654E-04	8.73305E-05	1.75654E-04	8.73305E-05	1.75654E-04
20	1.74582E-04	1.74582E-04	8.68329E-05	1.74582E-04	8.68329E-05	1.74511E-04

**Table 5.18 DSA Error Norm of the Offshore Structure  
( With Respect to Design Variable One )**

Number of modes	MDM		MAM	
	Error Norm of dZ/db	Error Norm of dŻ/db	Error Norm of dZ/db	Error Norm of dŻ/db
10	1.0	1.0	1.0	1.0
12	7.0716567E-01	7.0716536E-01	7.00262E-01	7.07165E-01
14	2.3138859E-02	2.3120125E-02	2.31388E-02	2.31196E-02
16	2.3138860E-02	2.3120144E-02	3.99325E-03	2.31196E-02
18	1.4597171E-03	1.4751136E-03	4.64125E-04	1.47511E-03
20	1.3636388E-03	1.3801533E-03	4.59564E-04	1.37986E-04

**Table 5.19 DSA Error Norm of the Offshore Structure  
( With Respect to Design Variable Two )**

Numbers of Modes	MDM		MAM	
	Error Norm of dZ/db	Error Norm of dŻ/db	Error Norm of dZ/db	Error Norm of dŻ/db
14	9.6451044E-03	9.6680986E-03	1.155956E-03	9.668360E-03
18	8.2814553E-04	8.2838469E-04	6.9250216E-05	8.283228E-04
20	8.2642312E-04	8.2666201E-04	6.9250123E-05	8.266021E-04

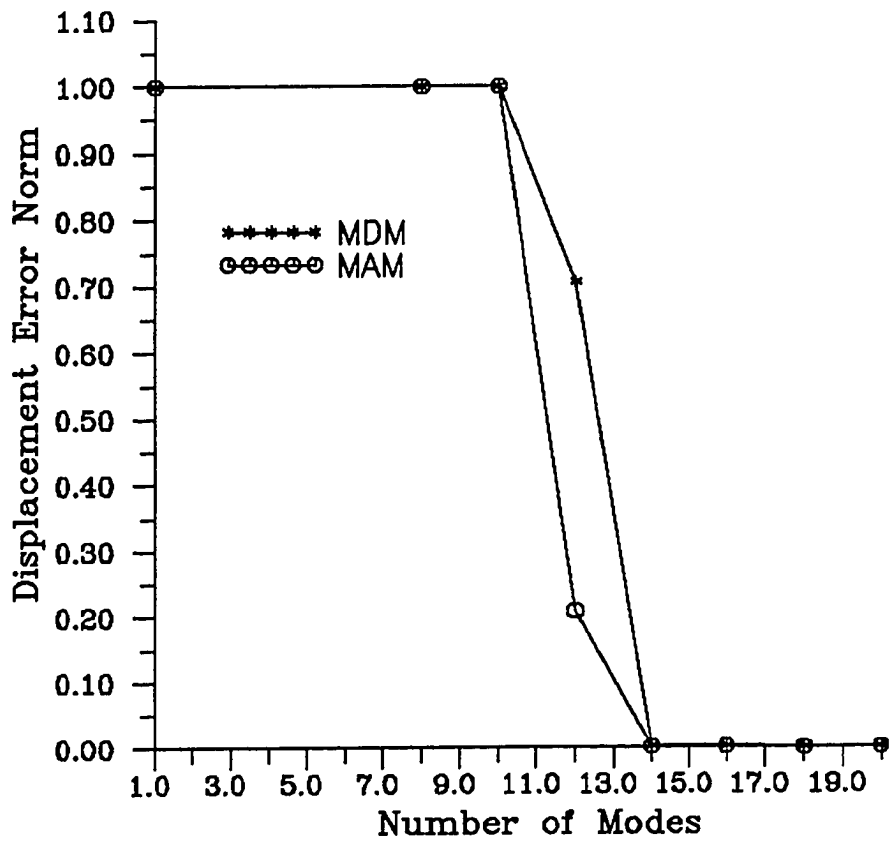
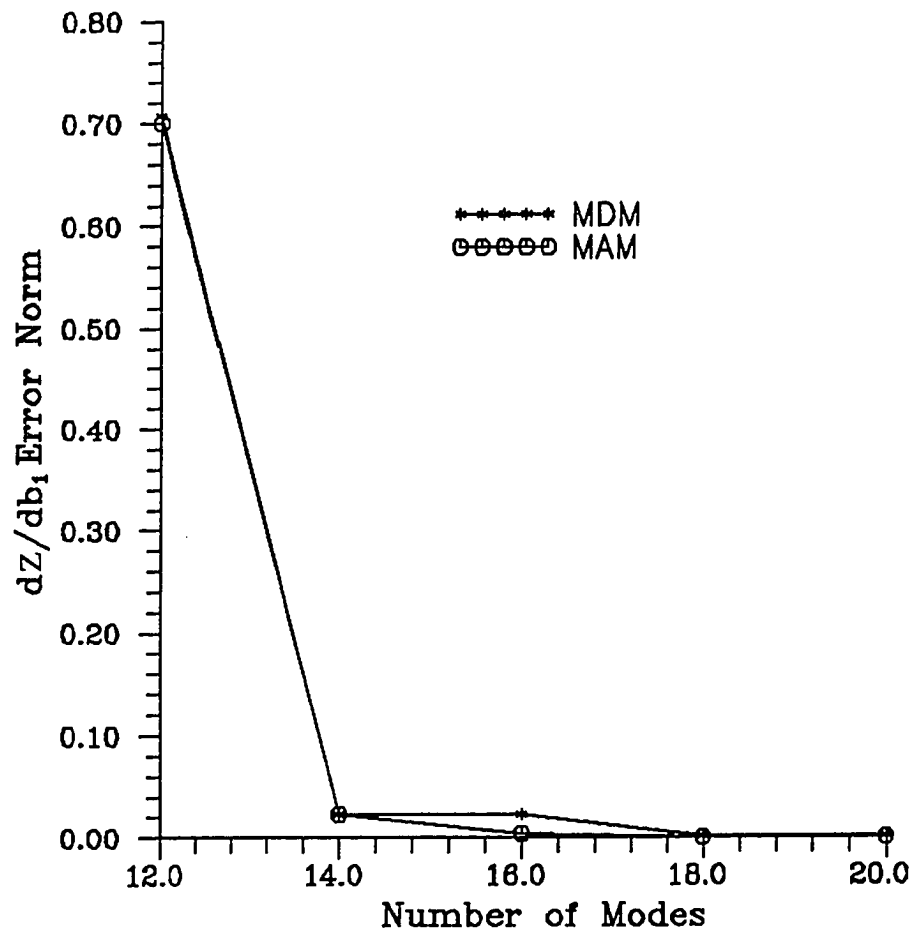
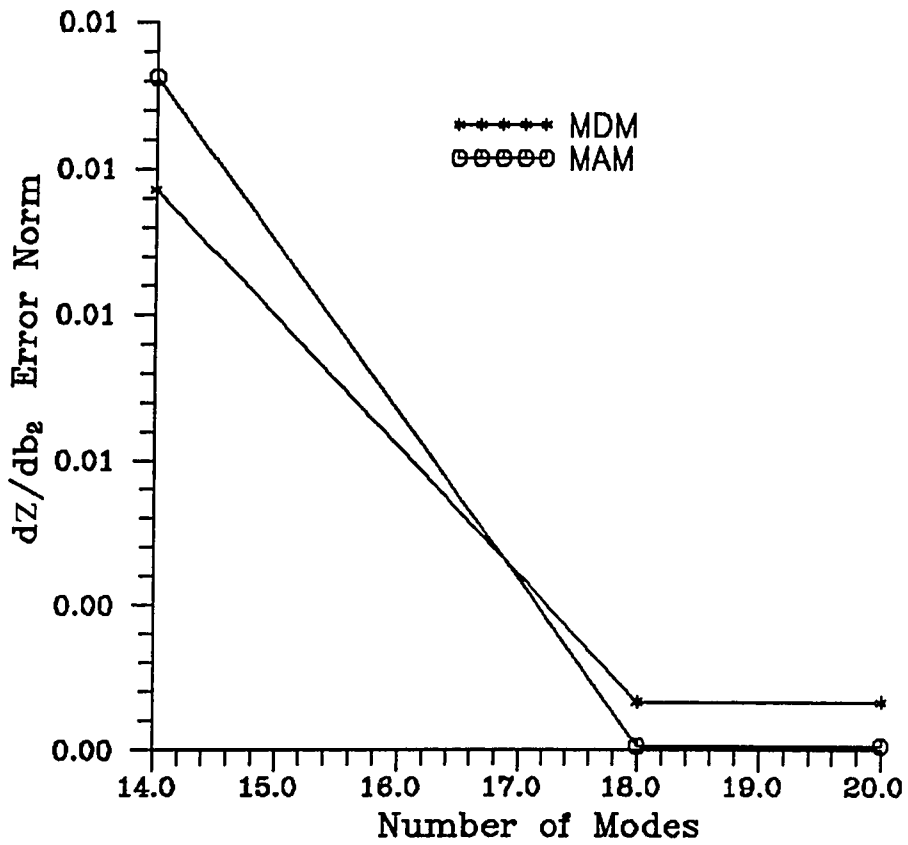


Figure 5.25 Displacement Error Norm of the Flexible Offshore Structure



**Figure 5.26 Displacement Derivative Error Norm of the Flexible Offshore Structure ( With Respect to Design Variable One )**



**Figure 5.27 Displacement Derivative Error Norm of the Flexible Offshore Structure ( With Respect to Design Variable Two )**

**Table 5.20 Eigenvalues and Eigenvalue Derivatives of the Offshore Structure**

Number	Eigenvalue	Frequency ( Hz )
1	0.936090104E-01	0.486943919E-01
2	0.104328885E+00	0.514070167E-01
3	0.120123943E+00	0.551613529E-01
4	0.686421969E+00	0.131860800E+00
5	0.277109582E+01	0.264938999E+00
6	0.288713960E+01	0.270429466E+00
7	0.329843880E+01	0.289050906E+00
8	0.331244816E+01	0.289664094E+00
9	0.335007894E+01	0.291304800E+00
10	0.595928577E+01	0.388523440E+00
11	0.646588479E+01	0.404700835E+00
12	0.391212421E+06	0.995466013E+02
13	0.391329959E+06	0.995615544E+02
14	0.392087165E+06	0.996578314E+02
15	0.119278831E+07	0.173820825E+03
16	0.293910254E+07	0.272852218E+03
17	0.293971441E+07	0.272880618E+03
18	0.293997465E+07	0.272892696E+03
19	0.404791830E+07	0.320210810E+03
20	0.417676288E+07	0.325267019E+03

$\frac{d\lambda_1}{db_1} = -5.455322746E-03,$	$\frac{d\lambda_1}{db_2} = -9.674249656E-02$
$\frac{d\lambda_2}{db_1} = -3.702377222E-03,$	$\frac{d\lambda_2}{db_2} = -1.199867522E-01$

#### 5.4.2 A Six-Story Eight-Bay Three-Dimensional Frame

### 5.4.2 A Six-Story Eight-Bay Three-Dimensional Frame

In this section, a typical three-dimensional frame, as shown in Fig. 5.28, with six stories and eight bays, and 756 degree of freedom, is studied.

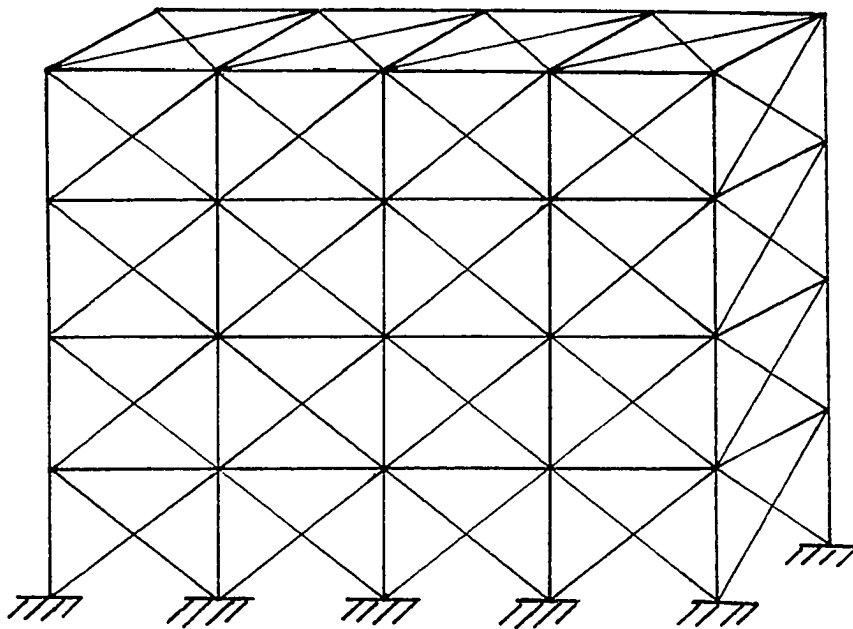
Table 5.21 shows the error norm for dynamic response and DSA for both MDM and MAM, with different numbers of nodes applied. CPU time for some typical runs are recorded for comparing the efficiency of MDM with MAM. The error norms are also plotted in Fig. 5.29 and Fig. 5.30 for displacements and the derivatives of displacements respectively. The CPU time, in Cray Y-MP ( Sabre, NASA Langley ), for MDM and MAM with different modes applied is shown in Fig. 5.31.

It can be clearly seen that the MAM could provide better accuracy with less modes used than the MDM, and MAM gives better efficiency also. For instance, when 40 modes are applied, the MDM gives 50.8% error norm for displacements, and 50.1% error norm for the displacement derivatives, the CPU time used is 2.88 seconds. While, with 40 modes applied, the MAM yields much better accuracy, the error norms are 7.3% and 7.0% respectively for the displacements and the derivatives of the displacements, and the CPU time is 2.976 seconds.

The lowest 70 eigenvalues and the eigenvalue derivatives of the two lowest eigenvalues are presented in Table 5.22.

The parallel vector computation efficiency of analyzing this problem is shown in Table 5.23. The computer used is Cray Y-MP, the time is measured using Tsecond.

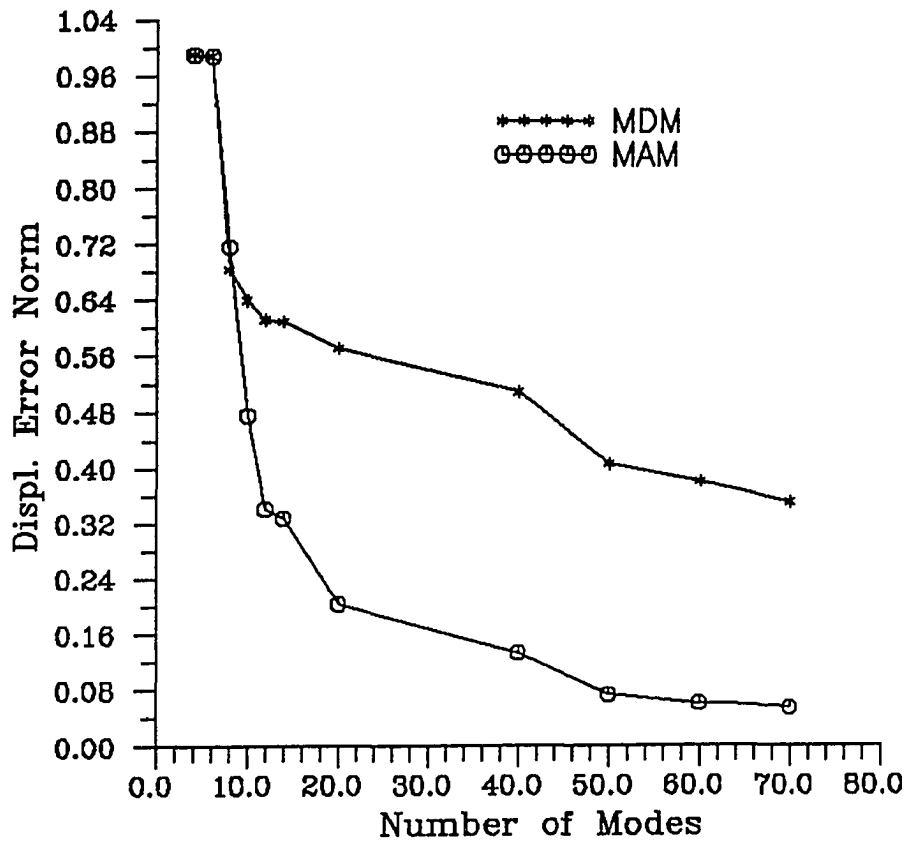




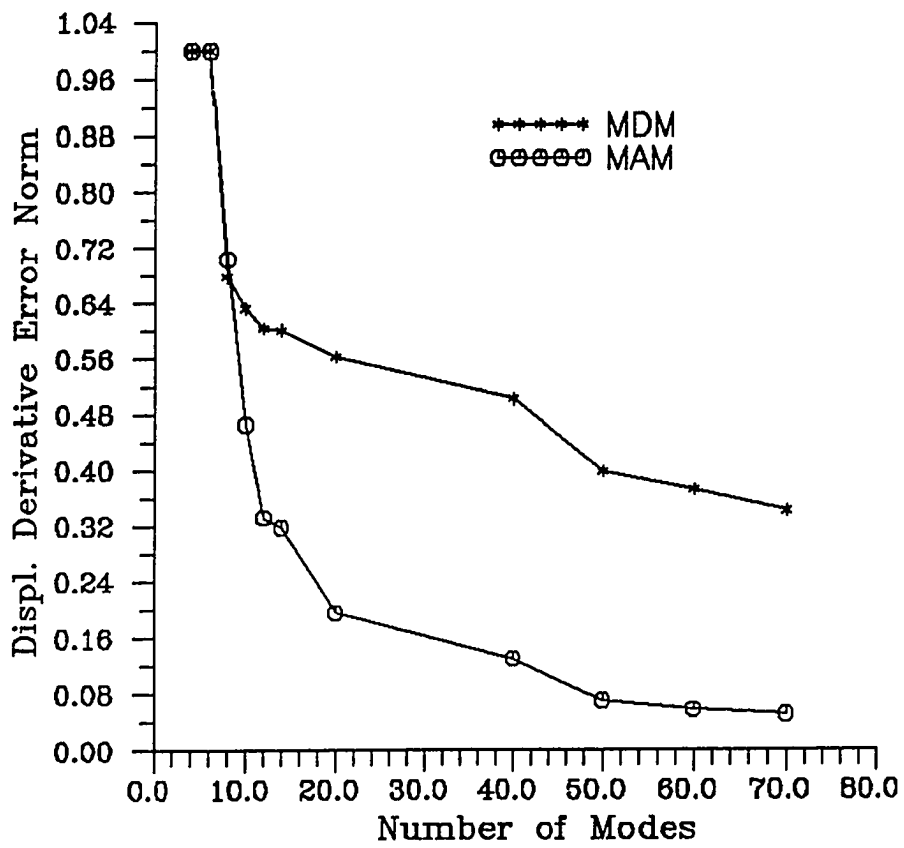
**Figure 5.28 A Typical Three-Dimensional Frame**

**Table 5.21** Algorithm Accuracy and Efficiency Analysis with the Six-Story Eight-Bay Three-Dimensional Frame ( DOF = 700 )

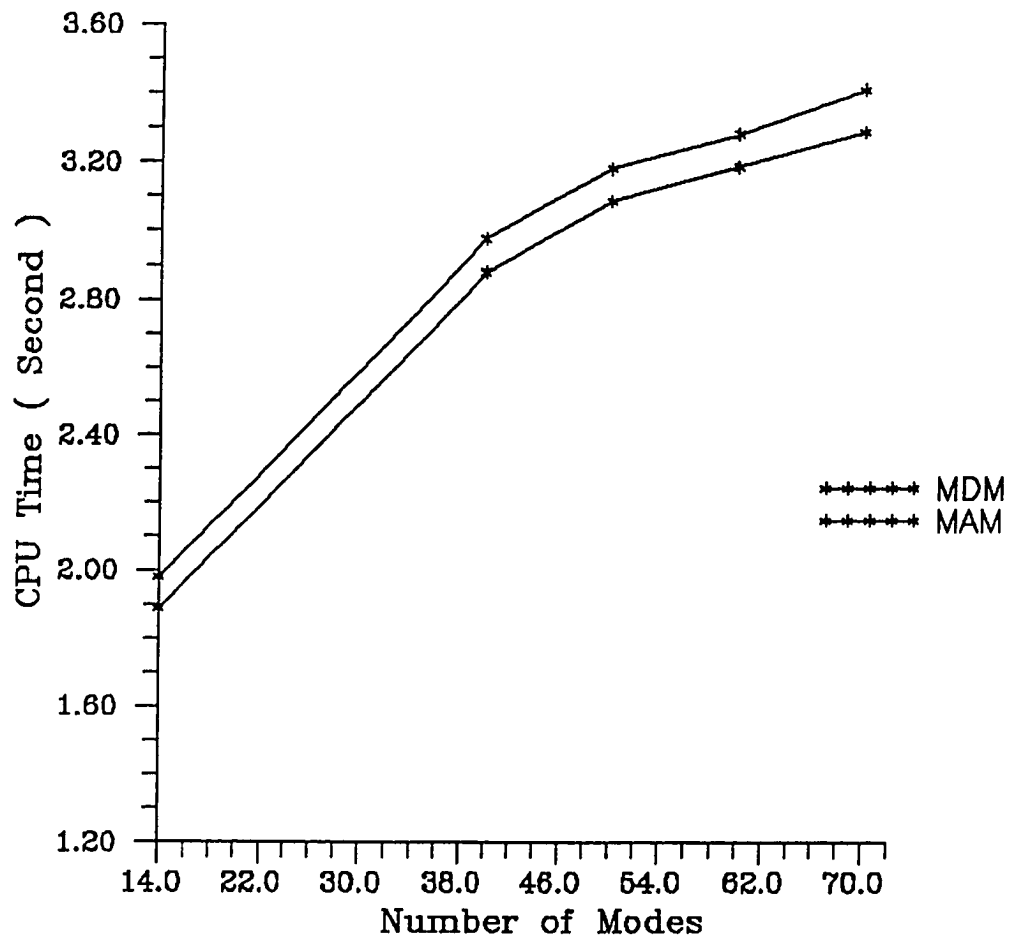
Number of modes	ERROR NORM ( MDM )				ERROR NORM ( MAM )			
	Z	dZ/db	Operation Counts	CPU Time ( Sec. )	Z	dZ/db	Operation Counts	CPU Time ( Sec. )
6	0.988330	1.0	—	—	0.988330	1.0	67.42100M	1.630024
8	0.684358	0.6783	—	—	0.716320	0.704161	79.86822M	1.723475
10	0.639893	0.6328	—	—	0.475347	0.465665	92.60575M	1.825641
12	0.612137	0.6044	—	—	0.341039	0.332155	105.6336M	1.903676
14	0.609188	0.6014	105.7871M	1.887691	0.327966	0.318315	118.9517M	1.980821
20	0.570868	0.5626	—	—	0.203264	0.195735	160.4794M	2.205218
40	0.508044	0.5012	292.6026M	2.878850	0.132882	0.129853	318.5051M	2.975665
50	0.405746	0.3977	347.2525M	3.083454	7.27891E-2	6.96631E-2	408.3201M	3.178930
60	0.380164	0.3718	461.3456M	3.185253	6.01111E-2	5.71532E-2	505.3927M	3.281309
70	0.350048	0.3417	553.8819M	3.286124	5.36730E-2	5.06626E-2	609.7229M	3.408275



**Figure 5.29 Displacement Error Norm of the Six-Story Eight-Bay Three-Dimensional Frame**



**Figure 5.30 Displacement Derivative Error Norm of the Six-Story Eight-Bay Three-Dimensional Frame**



**Figure 5.31 CPU Time Comparison of MAM Versus MDM for the Six-Story Eight-Bay Three-Dimensional Frame**

**Table 5.22 Eigenvalues and Eigenvalue Derivatives of the Six-Story Eight-Bay Three-Dimensional Frame**

Number	Eigen-Frequency(Hz)	Number	Eigen-Frequency(Hz)
1	0.572611E+01	36	0.875992E+02
2	0.829902E+01	37	0.886950E+02
3	0.111081E+02	38	0.913655E+02
4	0.184677E+02	39	0.931257E+02
5	0.265546E+02	40	0.952263E+02
6	0.306910E+02	41	0.960364E+02
7	0.345493E+02	42	0.985877E+02
8	0.355950E+02	43	0.101159E+03
9	0.388953E+02	44	0.101978E+03
10	0.416819E+02	45	0.103179E+03
11	0.450212E+02	46	0.104649E+03
12	0.458277E+02	47	0.106149E+03
13	0.472004E+02	48	0.107032E+03
14	0.493625E+02	49	0.108682E+03
15	0.497159E+02	50	0.109527E+03
16	0.541969E+02	51	0.114825E+03
17	0.581508E+02	52	0.115120E+03
18	0.605358E+02	53	0.117124E+03
19	0.625017E+02	54	0.118449E+03
20	0.631007E+02	55	0.119285E+03
21	0.651839E+02	56	0.119791E+03
22	0.670239E+02	57	0.121779E+03
23	0.694373E+02	58	0.124463E+03
24	0.717013E+02	59	0.126171E+03
25	0.720210E+02	60	0.129554E+03
26	0.726173E+02	61	0.130917E+03
27	0.739466E+02	62	0.133768E+03
28	0.745699E+02	63	0.136344E+03
29	0.756607E+02	64	0.140725E+03
30	0.768734E+02	65	0.142837E+03
31	0.790121E+02	66	0.146310E+03
32	0.813189E+02	67	0.151438E+03
33	0.827175E+02	68	0.154684E+03
34	0.830653E+02	69	0.158034E+03
35	0.846688E+02	70	0.162699E+03

$\frac{d\lambda_1}{db} = 46.377344$	$\frac{d\lambda_2}{db} = 24.133193$
-------------------------------------	-------------------------------------

**Table 5.23 Parallel-Vector Computation Efficiency of the Six-Story Eight-Bay Three-Dimensional Frame (14 Modes, MAM)**

NP	Time	Efficiency	Speed Up
1	1.98082113	100%	1.00
2	1.08386757	91.38%	1.83
3	0.75541916	87.41%	2.622

Total Number of Operation = 160.648 M

Computer Used: Cray Y-MP ( Sabre )

Time Measured By: Tsecnd

### 5.4.3 Three-Dimensional CSI Design Model

A three-dimensional finite element model to study Control-Structure Interaction (CSI) [58] is shown in Fig. 5.32. The structure has 1647 beam elements, 537 nodes with six DOF per node, with totally 3096 DOF and 17 design variables.

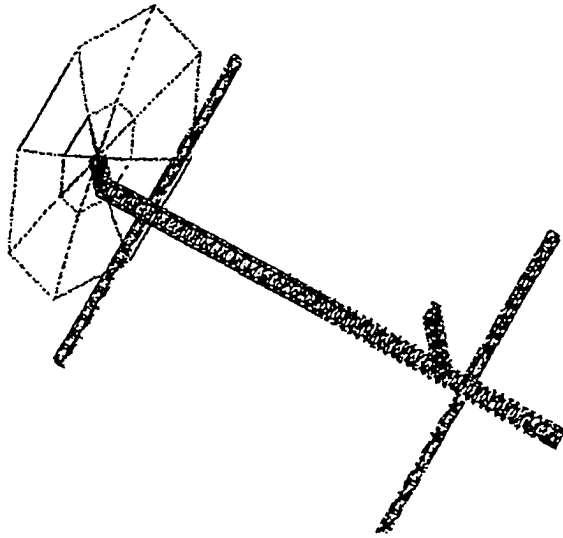
The original data for this model is slightly altered by the author: Some of the DOF are fixed to avoid rigid body motion; Most of the cross-sections are ring shape in the original data, the other types of cross-sections (i.e. angles, etc.) are converted into ring sections.

The error norm analysis is conducted for MDM, MAM, and MMAM, with different number of modes applied. The operation counts and CPU time are also recorded for each runs. The results are presented in Tables 5.24 and 5.25. Error Norms of displacement, velocity, and acceleration vectors are plotted in Figs. 5.33 and 5.35. Displacement derivative error norm for a typical design variable is presented in Fig. 5.34. From table 5.24, it can be seen that when 30 modes are applied: MDM gives error norm of 15.89% for the displacements and 16.77% for the displacement derivatives, and the CPU time used is 6.146 seconds; while, MAM yields error norm of 1.84% for displacements and 2.60% for the derivatives of the displacements, and the CPU time used is 6.308 seconds. The CPU time for the DSA computation with MDM versus MAM is also plotted in Fig. 5.36. It is illustrated that, by Table 5.25 and Fig. 5.35, the MMAM developed in this study can significantly improve the accuracy of velocity and acceleration vectors.



The lowest 40 eigenvalues are shown in Table 5.26.

The parallel-vector computing efficiency is shown in Table 5.27. The computer used is Cray Y-MP, time is measured using Tsecond.



Number of D.O.F	3096
Maximum bandwidth	108
Number of elements	1647
Number of nodes	537

**Figure 5.32 The CSI Design Finite Element Model**

**Table 5.24 Algorithm Accuracy and Efficiency Analysis with the Three-Dimensional CSI Design Model**

Number of modes	ERROR NORM ( MDM )				ERROR NORM ( MAM )			
	Z	dZ/db	Operation Counts	CPU Time ( Sec. )	Z	dZ/db	Operation Counts	CPU Time ( Sec. )
4	1.0	1.0	---	---	1.0	1.0	---	---
8	0.589504	0.643229	245.5907M	3.49062	0.343344	0.513812	874.9878M	3.68806
12	0.403425	0.416261	1275.602M	4.45026	0.139704	0.173171	1294.869M	4.64660
16	0.391966	0.400710	1695.791M	4.77253	0.126149	0.152441	1717.221M	4.97816
20	0.331218	0.353003	2117.833M	5.20209	9.53464E-2	0.121760	2142.042M	5.43492
30	0.158937	0.167747	3181.044M	6.14599	1.83897E-2	2.60357E-2	3214.900M	6.30769
40	0.103614	0.102164	4255.832M	7.54099	6.01313E-3	6.79009E-3	4303.196M	7.72477

**Table 5.25 Velocity and Acceleration Error Norms  
of the Three-Dimensional CSI Design Model  
(MMAM Versus MDM/MAM)**

Number of Modes	Velo. & Accel. Error Norm	
	MDM & MAM	MMAM
8	5.89504E-01	1.84500E-01
20	3.31218E-01	1.40078E-01
40	1.03614E-01	5.37279E-02

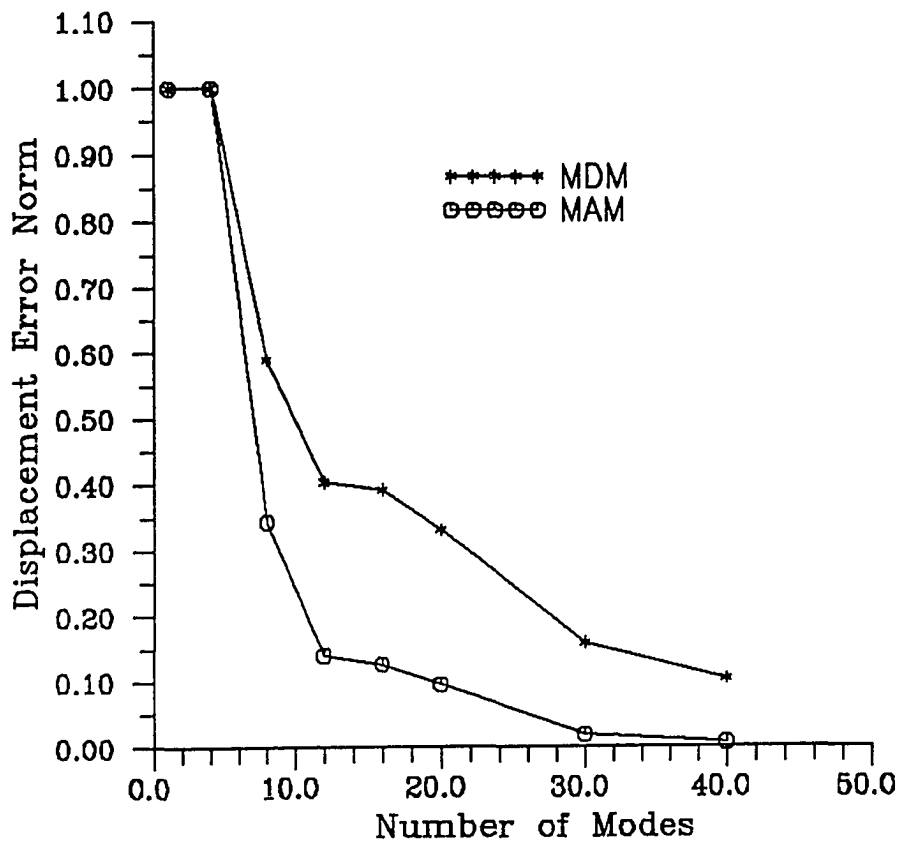
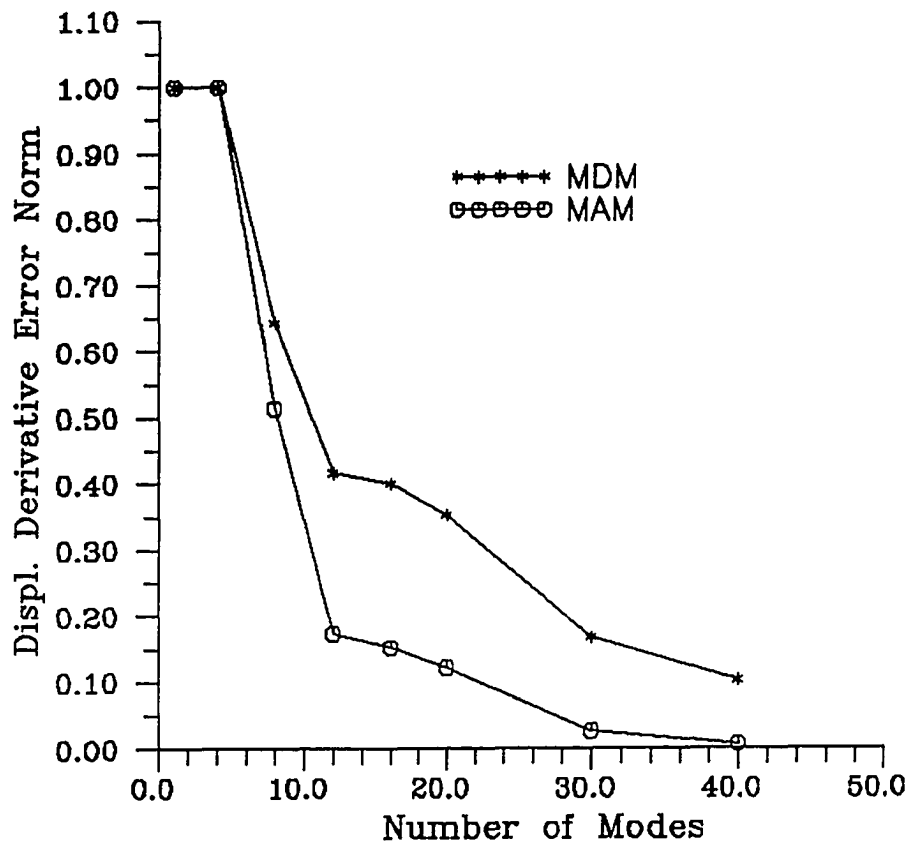
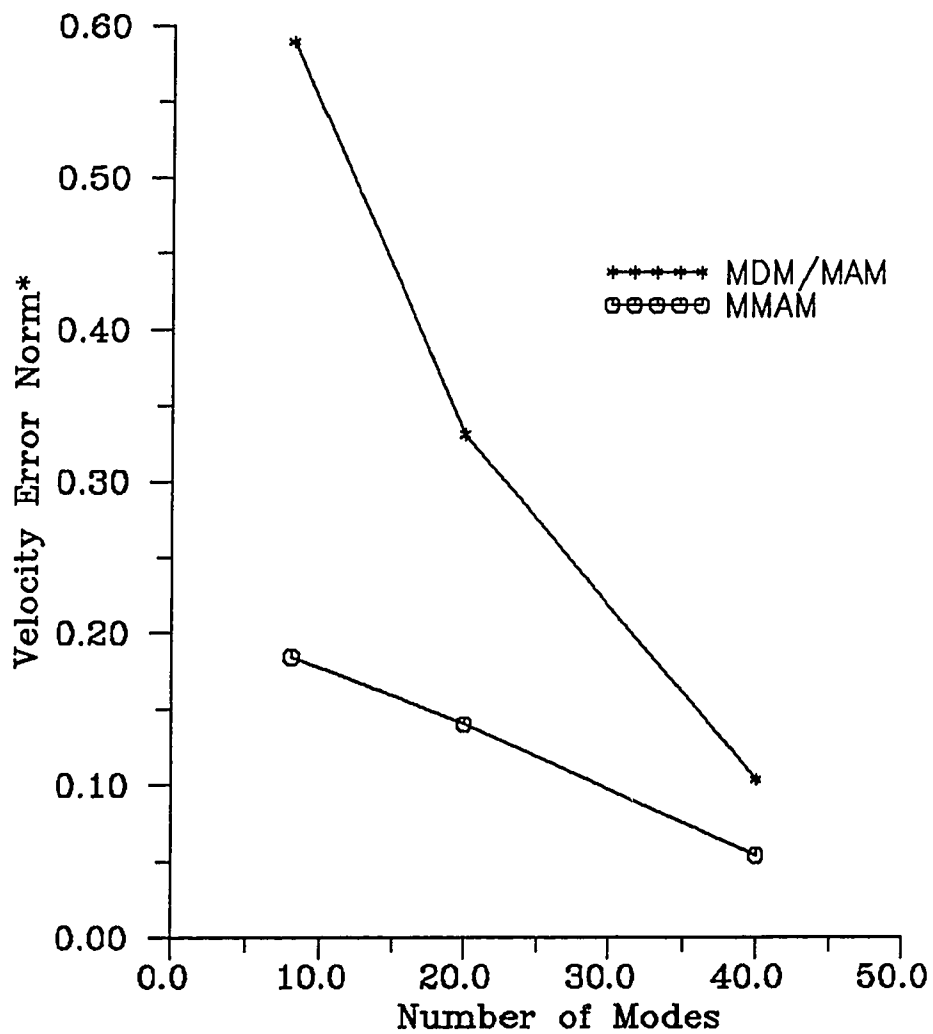


Figure 5.33 Displacement Error Norm of the CSI Design Model



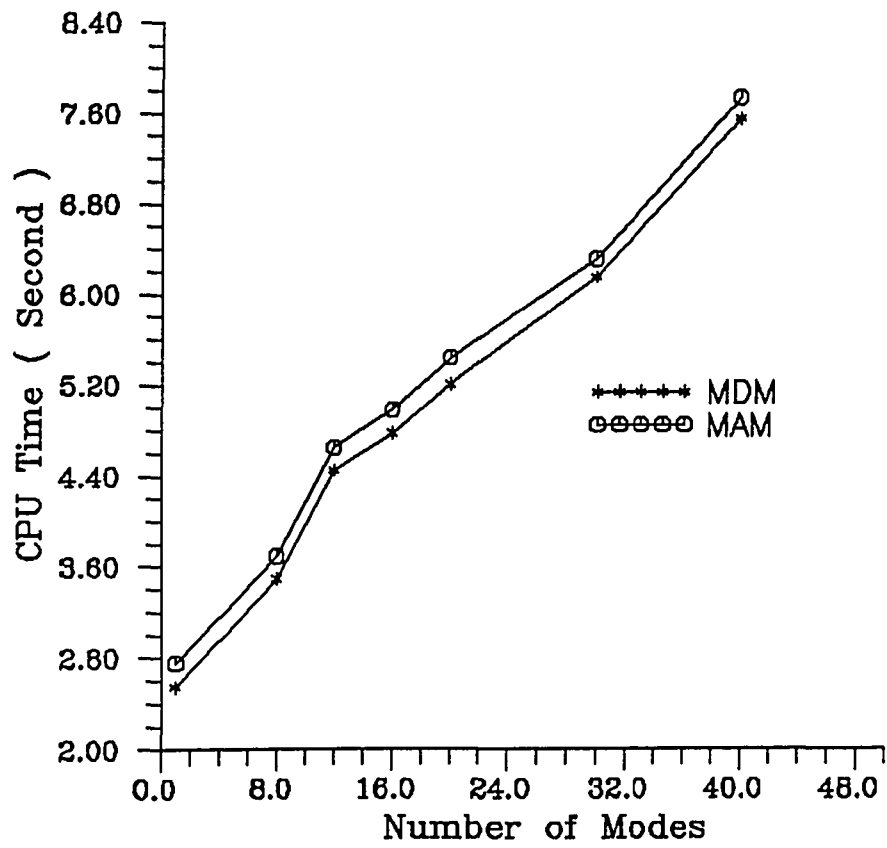
**Figure 5.34 Displacement Derivative Error Norm of the CSI Design Model**



**Figure 5.35 Velocity and Acceleration Error Norm of the CSI Design Model**

---

\* Velocity error norm and the acceleration error norm are the same.



**Figure 5.36 CPU Time Comparison of MAM Versus MDM for the CSI Design Model**



**Table 5.26 Eigenvalues of the CSI Design Model**

---

Number	Eigen-Frequency(Hz)	Number	Eigen-Frequency(Hz)
1	0.116070E+02	21	0.105169E+03
2	0.143949E+02	22	0.108988E+03
3	0.145239E+02	23	0.115615E+03
4	0.157429E+02	24	0.117691E+03
5	0.381830E+02	25	0.121541E+03
6	0.394871E+02	26	0.126516E+03
7	0.508478E+02	27	0.127841E+03
8	0.542062E+02	28	0.134522E+03
9	0.624059E+02	29	0.145428E+03
10	0.648400E+02	30	0.150936E+03
11	0.734292E+02	31	0.157011E+03
12	0.755228E+02	32	0.161472E+03
13	0.760329E+02	33	0.169400E+03
14	0.793875E+02	34	0.178729E+03
15	0.818006E+02	35	0.182048E+03
16	0.888392E+02	36	0.184844E+03
17	0.928983E+02	37	0.192432E+03
18	0.949426E+02	38	0.198713E+03
19	0.101190E+03	39	0.199323E+03
20	0.101916E+03	40	0.206673E+03

---

**Table 5.27 Parallel-Vector Computation Efficiency  
with the CSI Design Model ( 40 Modes, MAM )**

NP	Time	Efficiency	Speed Up
1	6.9804118	100%	1.00
2	3.8493594	90.67%	1.81
3	2.7916512	83.35%	2.50
4	2.4137	72.30%	2.89

Total Number of Operation = 44589.723 M

Computer Used: Cray Y-MP ( Reynolds )

Time Measured By: Tsecond

## **6. CONCLUSIONS AND FUTURE RESEARCH**

### **6.1 Conclusion Remarks**

The direct differentiation method is justified as the best for design sensitivity analysis in structural dynamics. An alternative formulation for design sensitivity analysis with direct differential method is developed in this study. The alternative formulation works efficiently with the reduced system, it avoids the expensive, complicated, and tedious computation of the eigenvector derivatives which is required in the existing reduced system formulation. It is demonstrated that the alternative formulation is accurate, simple, and very efficient. The relationship between the alternative formulation and the existing reduced system formulation is initially established by the author. It is further analytically proved that the alternative formulation and the reduced system formulation are mathematically equivalent, when the transformation from the original full system to the reduced system is exact, i.e., when all the modes are included.

System reduction technique is applied with eigen-vectors as the base vectors. The eigenvectors are obtained by an efficient Lanczos algorithm. Both the dynamic response analysis and design sensitivity analysis are conducted efficiently working

with the reduced system. It is shown that mode acceleration method performs well in both dynamic response analysis and the design sensitivity analysis. A modified modal acceleration method is presented, which improves not only the displacements, as the mode acceleration method does, but also the velocities and the accelerations.

A parallel-vector algorithm for design sensitivity in structural dynamics is developed, which serves both global and local parallelization, also makes use the advantage of the vector computing. The effective algorithm developed is then illustrated through several examples for its accuracy and efficiency. The accuracy of the algorithm developed is appreciated by comparing with the information from the solution of the original full system. The efficiency is showed by analyzing large-scale structure in high performance computers Cray 2 and Cray Y-MP.

## **6.2 Notes for Future Research**

Some of the possible extensions of this work and some ideas for future research are discussed in this section.

- 1) It has been shown that the alternative formulation is accurate, efficient, and simple to apply. Further research could be conducted to compare quantitatively the efficiencies and accuracies of the alternative formulation versus the existing reduced system formulation. The alternative formulation does not require eigenvector derivatives, thus, it obviously has much better efficiency than the existing reduced system formulation. Further research could be conducted to compare quantitatively

the computational efficiency of the two approaches. It is analytically proven that the two formulations are equivalent, when the transformation is exact. However, besides the approximation introduced by system reduction, both formulations involve some other different kinds of approximation. For the existing reduced system formulation, error could be produced in the process of obtaining eigenvector derivatives. It is well known that the eigenvectors converge in a much slower rate than the eigenvalue do. On the other hand, when the alternative formulation is applied, error could be introduced from the assumption of Eq. (3.64), i.e. the displacement derivative of the original system is approximated as matrix-vector product of the eigenvector matrix with the reduced system displacement derivatives. Thus, the numerical accuracy comparison of the two approaches could be a valuable research topic. Research on quantitative comparison of both accuracy and efficiency of the two formulations could provide practical guidance for DSA in structural dynamics. Also, it deserves further investigation under the parallel-vector computing environment.

2) In this study, the eigenvector derivatives are avoided since the alternative formulation is selected. The study of the eigenvector derivatives could be a potential research topic also. This area has attracted great amount of research interests [42, 43, 44, 45]. Numerical comparison [44] has been made for various techniques of eigenvector derivatives. However, it is valuable to re-evaluate their performance in parallel-vector computation environment.

3) For both alternative formulation and existing reduced system formulation, which could be applied in conjunction with the MAM, certain number of modes

(eigenvectors) is required to obtain DSA solutions with an acceptable accuracy. Although the number of modes required is problem dependent, and the convergence criteria could be very complicated to establish, further research work on mode convergence is needed to provide some criteria to serve as a guidance for DSA in structural dynamics.

4) With the study of the DSA in structural dynamics, it is natural for the future research to conduct dynamic structural optimization. More important, there are many interesting topics in structural optimization under dynamic loads which deserve further study, as for example, the treatment of point-wise time dependent state variable constraints.

5) Only linear structures are discussed in this study. The alternative formulation could be valid for non-linear dynamic structural system also, provided the modal reduction technique could be applied. This study is limited to fixed geometry problem. DSA for Shape optimization under dynamic loads is another interesting topic for the author to learn, and hopefully conduct some research work on it in the future.

## REFERENCES

- [ 1 ] Ahmed K. Noor and Satya N. Alturi, Advances and Trends in Computational Structural Mechanics, AIAA Journal Vol. 25, No. 7, July 1987, pp. 977-995.
- [ 2 ] Olaf O. Storaasli, Duc T. Nguyen, and Tarun K. Agarwal The Parallel Solution of Large-Scale Structural Analysis Problems on Supercomputers, Proceedings of the AIAA/ASME/ASCE/AHS 30th Structures, Structural Dynamics and Materials Conference, Mobile, AL., April 3-5, 1989. Also appeared in the AIAA Journal, Vol. 28, No. 7, July 1990, pp. 1211-1216.
- [ 3 ] Jiangning Qin, Duc T. Nguyen and Yongxing Zhang, A Parallel-vector Lanczos Eigen-Solver for Structural Vibration Problems, proceedings of the Fourth International Conference on Recent Advances in Structural Dynamics, 15 - 18 July, 1991, Southampton, Great Britain.
- [ 4 ] Duc T. Nguyen, Olaf O. Storaasli, Moayad Al-Nasra, Yongxing Zhang, Majdi Baddourah, and Tarun K. Agarwal, Parallel-vector Computation for Nonlinear Unconstrained Optimization Problems, proceedings of the Symposium on Parallel Methods on Large-Scale Structural Analysis and Physics Applications, 5 - 6 February, 1991, NASA Langley Research Center, Hampton, Virginia. Also to appear in Computing Systems in Engineering Journal.
- [ 5 ] J. S. Kowalik, Parallel Computers and Their Impact on Numerical Algorithms and Related Softwares, Nonlinear Optimization 1981, edited by M. J. D. Powell, NATO Conference Series, Series II: System Science, 1982.
- [ 6 ] Howard M. Adelman and R. T. Haftka, Sensitivity Analysis of Discrete Structural Systems, AIAA Journal, Vol. 24, 1986, pp. 823 -832.
- [ 7 ] B. L. Pierson, A Survey of Optimal Structural Design Under Dynamic Constraints, Int. J. Numer. Meths. Engrg, Vol. 4, 1972, pp. 491-499.
- [ 8 ] R. L. Fox and M. P. Kapoor, structural Optimization in Dynamic Response Regime: A Computational Approach. AIAA J. Vol. 8, No. 10, 1970, pp. 1798-1804.

- [ 9 ] Juan H. Cassis, Lucien A. Schmit, Jr., Optimum Structural Design with Dynamic Constraints, J. Struct. Div., ASCE, Vol. 102, No. ST10, 1976, pp. 2053-2071.
- [10 ] T. T. Feng, J. S. Arora and E. J. Haug, Optimal Structural Design Under Dynamic Loads, Int. J. Numer. Meths. Engrg., 11, 1977, pp. 39-52.
- [11 ] C. C. Hsieh and J. S. Arora, Design Sensitivity Analysis and Optimization of Dynamic Response, Computer Meth. Appl. Mech. and Engrg, 43, 1984, pp. 195-219.
- [12 ] C. C. Hsieh and J. S. Arora, Structural Design Sensitivity Analysis with General Boundary Conditions: Static Problem, Int. J. for Num. Meths. in Eng., Vol. 20, 1984, pp. 1661-1670.
- [13 ] C. C. Hsieh and J. S. Arora, Structural Design Sensitivity Analysis with General Boundary Conditions: Dynamic Problem, Int. J. for Num. Meth. in Eng., Vol. 20, 1984, pp. 1661-1670.
- [14 ] C. C. Hsieh and J. S. Arora, A Hybrid Formulation for Treatment of Point-wise State Variable Constraints in Dynamic Response Optimization, Computer Methods in Applied Mechanics and Engineering, Vol. 48, 1985, pp. 171-189.
- [15 ] O.K. Lim, J. S. Arora, Dynamic Response Optimization Using An Active SET RQP Algorithm, Int. J. Numer. Meths. Eng., 1987, pp. 1827-1840.
- [16 ] C. H. Tseng and J. S. Arora, Optimal Design of Systems for Dynamic and Controls Using Sequential Quadratic Programming, AIAA Journal, Vol. 27, No. 12, December 1989, pp. 1793-1800.
- [17 ] R. A. Meric, Shape Design Sensitivity Analysis of Dynamic Structures, AIAA Journal, Vol. 26, No. 2, 1988, pp. 206-212.
- [18 ] E. J. Haug and J. S. Arora, Applied Optimal Design, Willey, New York, 1979.
- [19 ] J. S. Arora, Introduction to Optimum Design, McGraw-Hill Book Comp., New York, 1989.
- [20 ] R. T. Haftka, Z. Gurdal and M.P. Kamat, Elements of Structural Optimization, second revised edition, Kluwer Academic Publisher, 1990.
- [21 ] Z. Moroz and R. T. Haftka, First and Second Order Sensitivity Analysis of Linear and Nonlinear Systems. AIAA Journal, Vol. 24, 1986, pp. 1887-1892.



- [22 ] E. L. Haug, K. K. Choi, and V. Komkov, Design Sensitivity Analysis of Structural Systems, Academic Press, 1986.
- [23 ] K. K. Choi, E. J. Haug, J. W. Hou, and V. N. Sohoni, Pshenichny's Linearization Method for Mechanical System Optimization, ASME J. of Mech., Transmissions, and Automation in Design, No. 1, Vol 105, 1983, pp. 97-103.
- [24 ] M. Haririan, J. B. Cardoso and J. S. Arora, Use of ADINA for Design Optimization of Nonlinear Structures. Comput. Struct. 26 (1/2), 1987, pp. 123-133.
- [25 ] K. K. Choi and S. Wang, Continuum Design Sensitivity Analysis of Structural Dynamic Response Using Ritz Sequence, Proceeding of the AIAA/ASME /ASCE /AHS 31st SDM Conference, Long Beach, CA., April 2-4, 1990.
- [26 ] R. T. Haftka and H. M. Adelman, Recent Developments in Structural Sensitivity Analysis, Struct. Optim., Vol. 1, No. 3, 1989, pp. 137 - 151.
- [27 ] William H. Greene, Computational Aspects of Sensitivity Calculations in Linear Transient Structural Analysis, NASA Technical Memorandum 4156, May 1990.
- [28 ] Duc T. Nguyen, K. T. Niu, A Parallel Algorithm for Structural Sensitivity Analysis on FLEX/32 Multicomputer, Proceedings of The 6th ASCE Structures Congress, Orlando, Florida, August 17-20, 1987, pp. 98-112.
- [29 ] H. F. Jordan, M. S. Benten, N. S. Arenstorf, and A. V. Ramann, Force User's Manual: A Portable Parallel Fortran, NASA CR 4265, January 1990.
- [30 ] K. J. Bathe, Finite Element Procedures in Engineering Analysis, Prentice Hall, Inc., New York, 1982.
- [31 ] O. C. Zienkiewicz, The Finite Element in Engineering Science, McGraw-Hill, New York, 1971.
- [32 ] Duc T. Nguyen, J. S. Shim, and Yongxing Zhang, The Component-Mode Method in A Parallel Computer Environment, Proceedings of The AIAA/ASME /ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, Va. April 18-20, 1988. pp 1705-1710, AIAA Paper No. 88-2438.

- [33 ] Yongxing Zhang, Duc T. Nguyen, Olaf Storaasli, Parallel-Vector Modal Superposition Algorithm by Ritz Vectors in Large Structural Dynamics, to be submitted to the 33rd SDM Conference, Dallas, Texas, April 13-15, 1992.
- [34 ] P. Leger, E. Wilson, M. Yuan, and J. M. Dikens, Dynamic Analysis by Direct Superposition of Ritz Vectors, *Earthquake Engrg. and Struc. Dyna.*, 10, 1882, pp. 813-821.
- [35 ] J. R. Hughes, The Finite Element Method: Linear Static and Dynamic Finite Element Analysis, Prentice-Hall, 1987.
- [36 ] Roy R. Craig, Jr. Structural Dynamics, An Introduction to Computer Methods, John Wiley & Sons, New York, 1981.
- [37 ] S. N. Penry, W. H. Wood, Comparison of Some Single-Step Methods for the Numerical Solution of The Structural Dynamics Equation, *Int. J. Numer. Meths. Engrg.*, 21, 1985, pp. 1941-1955.
- [38 ] R.E. Cornwell, R. R. Craig Jr. and C. P. Johnson, On the Application of the Mode-Acceleration Method to Structural Engineering Problem. *Earthquake Engineering and Structural Dynamics*, Vol. 11, 1983, pp. 679-688.
- [39 ] K. Subbaraj, M. A. Dokainish, A Survey of Direct Time-Integration Methods in Computational Structural Dynamics II: Implicit Methods, *Compt., Struc.*, 32, 1989, pp. 1387-1401.
- [40 ] Y. T. Leung, Fast Response Method for Undamped Structures, *Engineering Structures*, Vol. 5, April 1983, 141-149.
- [41 ] Charles J. Camarda, Raphael T. Haftka and Michael F. Riley, An Evaluation of Higher-order Modal Methods for Calculating Transient Structural Response, NASA Technical Memorandum 89134, NASA Langley Research Center, April 1987.
- [42 ] Richard B. Nelson, Simplified Calculation of Eigenvector Derivatives, *AIAA Journal*, Vol.14, No. 9, 1976, pp. 1201-1205.
- [43 ] I. U. Ojalvo, Efficient Computation of Mode Shape Derivatives for Large Dynamic Systems, *AIAA Journal*, Vol. 25, No. 10, 1986, pp. 1386-1396.
- [44 ] Thomas R. Sutter, Charles J. Camarda, Joanne L. Walsh, Howard M. Adelmans, Comparison of Several Methods for Calculating Vibration Mode Shape Derivatives, *AIAA Journal*, Vol. 26, No. 12, 1988, pp. 1506-1511.

- [45 ] Gene J. W. Hou, Sean P. Kenny, Eigenvalue and Eigenvalue Sensitivity and Approximate Analysis for Repeated Eigenvalue Problems, Proceedings of the AIAA/ASME/ASCE/AHS 32nd Structures, Structural Dynamics and Materials Conference, Baltimore, MD, April 8-10, 1991, pp. 373-383, AIAA Paper No. AIAA-91-1089-CP.
- [46 ] R.L. Dailey, Eigenvector Derivatives with Repeated Eigenvalues, AIAA Journal, Vol. 27, No. 4, 1988, pp. 486-491.
- [47 ] N. B. Parlett, The Symmetric Eigenvalue Problem, Prentice-Hall, Englewood Cliffs, N.J. 1980.
- [48 ] B. Nour-Omid, B. N. Parlett, and R. L. Taylor, Lanczos versus Subspace Iteration for Solution of Eigenvalue Problems, Int. J. for Numer. Methods in Engineering, Vol. 19, 1983, pp. 859-871.
- [49 ] M.A. Dokainish, K. Subbaraj, A Survey of Direct Time-Integration Methods in Computational Structural Dynamics I: Explicit Methods, Compt. & Struct., 32, 1989, pp. 1371-1386.
- [50 ] Pierre Leger, Edward L. Wilson and Ray W. Clough, The Use of Load Dependent Vectors for Dynamic and Earthquake Analysis, Rport to NSF, Report No. UCB/EERC-86/04, March 1986.
- [51 ] K. J. Bathe and Sheryl Gracewski, On Nonlinear Dynamic Analysis Using Substructuring and Modes Superposition, Computers and Structures, Vol. 13, 1981, pp. 699-707.
- [52 ] R. E. Nickell, Nonlinear Dynamics by Mode Superposition, J. Compt. Method Appl. Mech. Engg., Vol. 7, 1976, pp. 107-129.
- [53 ] A. K. Noor, Recent Advances in Reduction Methods for Nonlinear Problems, J. Compt. Structures, Vol. 13, 1980, pp. 31-44.
- [54 ] Dimitri P. Bertsekas, John N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Prentice-Hall, 1989.
- [55 ] Michael J. Quinn, Designing Efficient Algorithms for Parallel Computers, McGraw-Hill Book Company, New York, 1987.
- [56 ] Wayne R. Cowell, and Christopher P. Thompson, Transforming Fortran Do Loops to Improve Performance on Vector Architectures, ACM Transactions on Mathematical Software, Vol. 12, No. 4, pp. 324-353, December 1986.

- [57 ] Yongxing Zhang, User's Manual for the Parallel-Vector Code PVDSASD, Department of Civil Engineering, Old Dominion University, July 1991.
- [58 ] W. K. Belvin, K. E. Elliott, A. Bruner, J. Sulla, and J. Bailey, The LaRc CSI Phase-O Evolutionary Model Tested: Design and Experimental Results, Proceedings of the Fourth Annual NASA/AOD Conference On Control /Structure Interaction Technology, Orlando, Florida, November 5-7, 1990.

**APPENDIX A RELATION BETWEEN THE ALTERNATIVE AND REDUCED  
SYSTEM FORMULATIONS FOR DSA IN STRUCTURAL DYNAMICS**

In this appendix, it is to be proven that the alternative and the reduced system formulations are equivalent, provided the transformation from the original system to the reduced system is exact.

Recalled Eq. (3.100)

$$\bar{\mathbf{M}} \frac{\partial \ddot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{C}} \frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{K}} \frac{\partial \mathbf{u}}{\partial \mathbf{b}_i} = \bar{\mathbf{R}}(t) \quad (\text{A.1})$$

which is the DSA equation for the reduced system formulation.

Substituting Eqs. (3.102) through (3.105) into Eq. (A.1), and using the relation of Eq. (3.78), (3.38), and (2.36), one obtains the right hand side ( R.H.S. ) of Eq. (A.1) as

$$\begin{aligned} \text{R.H.S.} &= \phi^T \left[ \frac{\partial \mathbf{Q}}{\partial \mathbf{b}_i} - \left( \frac{\partial \mathbf{M}}{\partial \mathbf{b}_i} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{b}_i} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{b}_i} \mathbf{Z} \right) \right] \\ &- \phi^T \left[ \frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} \mathbf{Z} \right] \\ &- \phi^T \left[ \mathbf{M} \frac{\partial \phi}{\partial \mathbf{b}_i} \ddot{\mathbf{u}} + \mathbf{C} \frac{\partial \phi}{\partial \mathbf{b}_i} \dot{\mathbf{u}} + \mathbf{K} \frac{\partial \phi}{\partial \mathbf{b}_i} \mathbf{u} \right] \end{aligned} \quad (\text{A.2})$$

Recalling Eq. (3.101) and noticing the relation in Eq. (2.43), the left hand side (L.H.S.) of Eq. (A.1) can be written as,

$$\text{L.H.S.} = \bar{\mathbf{M}} \frac{\partial \ddot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{C}} \frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{K}} \frac{\partial \mathbf{u}}{\partial \mathbf{b}_i} \quad (\text{A.3})$$

Thus, Eq. ( A.1 ) becomes,

$$\begin{aligned} \bar{\mathbf{M}} \frac{\partial \ddot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{C}} \frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{K}} \frac{\partial \mathbf{u}}{\partial \mathbf{b}_i} &= \boldsymbol{\phi}^T \left[ \frac{\partial \mathbf{Q}}{\partial \mathbf{b}_i} - \left( \frac{\partial \mathbf{M}}{\partial \mathbf{b}_i} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{b}_i} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{b}_i} \mathbf{Z} \right) \right] \\ &- \boldsymbol{\phi}^T \left[ \frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{Z}} \frac{\partial \mathbf{Z}}{\partial \mathbf{b}_i} \mathbf{Z} \right] \\ &- \boldsymbol{\phi}^T \left[ \mathbf{M} \frac{\partial \phi}{\partial \mathbf{b}_i} \ddot{\mathbf{u}} + \mathbf{C} \frac{\partial \phi}{\partial \mathbf{b}_i} \dot{\mathbf{u}} + \mathbf{K} \frac{\partial \phi}{\partial \mathbf{b}_i} \mathbf{u} \right] \end{aligned} \quad (\text{A.4})$$

Recall Eq. ( 3.106 ), which is the DSA equation for the alternative formulation,

$$\bar{\mathbf{M}} \ddot{\mathbf{q}} + \bar{\mathbf{C}} \dot{\mathbf{q}} + \bar{\mathbf{K}} \mathbf{q} = \bar{\mathbf{R}}(\mathbf{t}) \quad (\text{A.5})$$

where,  $\bar{\mathbf{K}}$  and  $\bar{\mathbf{R}}$  are defined in Eqs. ( 3.107 ) and ( 3.108 ).

Substituting the expressions for  $\bar{\mathbf{K}}$  and  $\bar{\mathbf{R}}$  into Eq. ( A.5 ) yields,

$$\begin{aligned} \bar{\mathbf{M}} \frac{\partial \ddot{\mathbf{u}}}{\partial \mathbf{b}_i} + \bar{\mathbf{C}} \frac{\partial \dot{\mathbf{u}}}{\partial \mathbf{b}_i} + \boldsymbol{\phi}^T \left( \mathbf{K} + \frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{Z}} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{Z}} \mathbf{Z} \right) \boldsymbol{\phi} \frac{\partial \mathbf{u}}{\partial \mathbf{b}_i} \\ = \boldsymbol{\phi}^T \left[ \frac{\partial \mathbf{Q}}{\partial \mathbf{b}_i} - \left( \frac{\partial \mathbf{M}}{\partial \mathbf{b}_i} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{b}_i} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{b}_i} \mathbf{Z} \right) \right] \\ - \boldsymbol{\phi}^T \left[ \mathbf{M} \frac{\partial \phi}{\partial \mathbf{b}_i} \ddot{\mathbf{u}} + \mathbf{C} \frac{\partial \phi}{\partial \mathbf{b}_i} \dot{\mathbf{u}} + \left( \mathbf{K} + \frac{\partial \mathbf{M}}{\partial \mathbf{Z}} \ddot{\mathbf{Z}} + \frac{\partial \mathbf{C}}{\partial \mathbf{Z}} \dot{\mathbf{Z}} + \frac{\partial \mathbf{K}}{\partial \mathbf{Z}} \mathbf{Z} \right) \frac{\partial \phi}{\partial \mathbf{b}_i} \mathbf{u} \right] \end{aligned} \quad (\text{A.6})$$

Eq. ( A.6 ) then could further be rewritten as

$$\begin{aligned}
& \bar{M} \frac{\partial \ddot{u}}{\partial b_i} + \bar{C} \frac{\partial \dot{u}}{\partial b_i} + \bar{K} \frac{\partial u}{\partial b_i} \\
&= \phi^T \left[ \frac{\partial Q}{\partial b_i} - \left( \frac{\partial M}{\partial b_i} \ddot{z} + \frac{\partial C}{\partial b_i} \dot{z} + \frac{\partial K}{\partial b_i} z \right) \right] \\
&- \phi^T \left[ \left( \frac{\partial M}{\partial Z} \ddot{z} + \frac{\partial C}{\partial Z} \dot{z} + \frac{\partial K}{\partial Z} z \right) \left( \frac{\partial \phi}{\partial b_i} u + \phi \frac{\partial u}{\partial b_i} \right) \right] \\
&- \phi^T \left[ M \frac{\partial \phi}{\partial b_i} \ddot{u} + C \frac{\partial \phi}{\partial b_i} \dot{u} + K \frac{\partial \phi}{\partial b_i} u \right]
\end{aligned} \tag{A.7}$$

Since Eq. ( 3.73 ), Eq. ( A.7 ) could be written as

$$\begin{aligned}
& \bar{M} \frac{\partial \ddot{u}}{\partial b_i} + \bar{C} \frac{\partial \dot{u}}{\partial b_i} + \bar{K} \frac{\partial u}{\partial b_i} \\
&= \phi^T \left[ \frac{\partial Q}{\partial b_i} - \left( \frac{\partial M}{\partial b_i} \ddot{z} + \frac{\partial C}{\partial b_i} \dot{z} + \frac{\partial K}{\partial b_i} z \right) \right] \\
&- \phi^T \left[ \frac{\partial M}{\partial Z} \frac{\partial Z}{\partial b_i} \ddot{z} + \frac{\partial C}{\partial Z} \frac{\partial Z}{\partial b_i} \dot{z} + \frac{\partial K}{\partial Z} \frac{\partial z}{\partial b_i} z \right] \\
&- \phi^T \left[ M \frac{\partial \phi}{\partial b_i} \ddot{u} + C \frac{\partial \phi}{\partial b_i} \dot{u} + K \frac{\partial \phi}{\partial b_i} u \right]
\end{aligned} \tag{A.8}$$

Eq. ( A.8 ) is identical to Eq. ( A.3 ). Thus, we have proved that the reduced and the alternative formulations are equivalent.

## AUTOBIOGRAPHY

Yongxing Zhang was born in Wubu, Shaanxi, China, on October 15, 1956. In January 1982, he graduated from Shaanxi Institute of Mechanical Engineering, Xi'an, China, with a Bachelor of Engineering Science. He worked as a faculty member in Department of Civil & Irrigation Engineering, Northwestern Agricultural University, China. He studied structural engineering in Michigan Technological University, and graduated with a Master of Science degree, in November 1986. He joined the Civil Engineering Ph.D. program at Old Dominion University (ODU), in 1987, and became a Ph.D. candidate in 1990. During his study in ODU, he worked on a project granted by the National Aeronautics and Space Administration (NASA) at Langley Research Center. He also lectured an engineering mechanics course, Statics. He is a student member of ASCE. He has published the following articles:

"A Parallel-Vector Lanczos Eigen-solver For Structural Vibration Problems" (with J. Qin, Duc T. Nguyen), Proceedings of the Fourth International Conference on Recent Advances in Structural Dynamics, July 15-18, 1991, Southampton, Great Britain.

"Parallel-Vector Computation for Large-scale Linear Structural Analysis and Nonlinear Unconstrained Optimization Problems" (with Duc T. Nguyen, Olaf Storaasli, Moayyad Al-Nasra, Majdi Baddourah, and Tarun Agrwal), Proceedings of the Symposium on Parallel Methods on Large-scale Structural Analysis and Physical Applications, February 1991, NASA Langley Research Center, Hampton, Virginia.

"An Alternative Formulation For Design Sensitivity Analysis of Linear Structural Dynamic Systems" (with Duc T. Nguyen, Gean W. Hou), submitted to Computers & Structures Journal.

"The Component-mode Method in a Parallel Computer Environment" (with Duc T. Nguyen and J. Shim), Proceedings of AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, Virginia, 1988.

Stability and Dynamic Response of Railroad Tracks Under Stochastic Loads, Vol. III: Combined Finite Elements and Boundary Elements for Track and Foundation (with F. Arbabi), Interim Report, NSF Grant MEA-8400317, 1987.

"A Combined Boundary Element Finite Element Micro-Computer Program" (with F. Arbabi), Proceedings of the Fourth National Conference on Micro-Computers in Civil Engineering, Orlando, November 1986, PP.1-5.