THEORETICAL AND SOFTWARE CONSIDERATIONS FOR NONLINEAR DYNAMIC ANALYSIS

NASA-CR-174504
19840007521

By
Richard J. Schmidt
Robert H. Dodds Jr.

An Interim Report on Research Sponsored by the
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Research Grant
NAG 3-32

Structural Engineering and Engineering Materials
SM Report No. 8
February 1983

THE UNIVERSITY OF KANSAS CENTER FOR RESEARCH, INC.
2291 Irving Hill Drive—Campus West Lawrence, Kansas 66045

THEORETICAL AND SOFTWARE CONSIDERATIONS FOR NONLINEAR DYNAMIC ANALYSIS

by

Richard J. Schmidt

Robert H. Dodds Jr.

An Interim Report on Research Sponsored by the

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Research Grant NAG 3-32

University of Kansas

Lawrence, Kansas

February, 1983

REPORT DOCUMENTATION PAGE	1. REPORT NO.	2.	3. Recipient's Accession No.

4. Title and Subtitle

Theoretical and Software Considerations for Nonlinear Dynamic Analysis

5. Report Date

6.

7. Author(s)

R. J. Schmidt and R. H. Dodds, Jr.

8. Performing Organization Rept. No.

SM Report No. 8

9. Performing Organization Name and Address

University of Kansas Center for Research, Inc.
2291 Irving Hill Drive, West Campus
Cleveland, Ohio

10. Project/Task/Work Unit No.

11. Contract(C) or Grant(G) No.

(C)

(G)

12. Sponsoring Organization Name and Address

13. Type of Report & Period Covered

14.

15. Supplementary Notes

16. Abstract (Limit: 200 words)

In the finite element method for structural analysis, it is generally necessary to discretize the structural model into a very large number of elements in order to accurately evaluate displacements, strains, and stresses. As the complexity of the model increases, the number of degrees of freedom can easily exceed the capacity of present-day software system.

A solution to this problem lies in improvements of structural analysis software including more efficient use of existing hardware and improved structural modeling techniques. One modeling technique that is used successfully in static linear and nonlinear analysis is multilevel substructuring.

This research extends the use of multilevel substructure modeling to include dynamic analysis and defines the requirements for a general purpose software system capable of efficient nonlinear dynamic analysis.

This report contains a presentation of the multilevel substructuring technique, a review of the analytical formulations and computational procedures for dynamic analysis and nonlinear mechanics, and concludes with a presentation of an approach to the design and implementation of a general purpose structural software system.

17. Document Analysis a. Descriptors

dynamics, modal synthesis, finite elements, nonlinear analysis, multilevel substructures, transient analysis

b. Identifiers/Open-Ended Terms

c. COSATI Field/Group

18. Availability Statement	19. Security Class (This Report)	21. No. of Pages
Release Unlimited	Unlimited	292
	20. Security Class (This Page)	22. Price
	Unlimited	

This Page Intentionally Left Blank

ACKNOWLEDGEMENT

Portions of this report were extracted from a special project submitted by Richard J. Schmidt to the Civil Engineering Department, University of Kansas, in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering. The study was conducted under the direction of Dr. Robert H. Dodds Jr.

The research was supported by the National Aeronautics and Space Administration under Grant No. 3-32. Special thanks go to Dr. Murray S. Hirschbein, NASA Technical Monitor.

Numerical computations were performed on the Harris 500 computer at the Computer Aided Engineering Facility, School of Engineering, University of Kansas.

This Page Intentionally Left Blank

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF TABLES

This Page Intentionally Left Blank

CHAPTER 1

INTRODUCTION

1.1 General

Complex structural systems are most often modeled for analysis as assemblies of discrete structural components. The most generally applicable discretization approach is the finite element method (FEM). In this method it is often necessary to divide the structural model into a very large number of elements in order to accurately evaluate displacements, strains, and stresses. As the number of elements increases, the number of degrees of freedom (DOF) in the model can easily exceed the capacity of many present-day computer facilities (both hardware and software) or can make the solution of the large order matrix equations prohibitively expensive [1.1]. This problem becomes particularly acute in nonlinear analysis. The iterative nature of nonlinear analysis requires that the matrix equations be solved repetitively, thus compounding the computational expense.

The application of time-dependent (dynamic) loading to complex structural models imposes additional difficulty on current structural software systems. A linear dynamic analysis can be orders of magnitude more expensive than a static analysis of the same model. When nonlinear response is also considered, the computational effort required for analysis can quickly make the solution infeasible.

Recent advances in computer architecture, primarily in brute computational speed, have somewhat relieved the problems of excessively large and costly analyses. It is not expected, however, that new developments in hardware will keep pace with the growing demands, in terms of model size and complexity, made by structural analysts. Nor will the very expensive super-computers become widely available. Moreover, it appears unlikely that orders of magnitude increases in high-speed memory capacity and data transmission rates (which analysts have come to expect every few years) can be extracted from current technology. If any short term relief is to come, it must derive from improvements in the structural analysis software. Such improvements lie in more efficient use of existing hardware and in improved structural modeling techniques. The focus of this work is on the improvement of structural modeling techniques.

One procedure that is used successfully in static analysis is multilevel substructured modeling [1.2]. This approach allows the various major structural units, or substructures, to be treated independently prior to final assembly. With the use of condensation techniques, a reduction in total model size can be achieved while exactly retaining the original system characteristics for static analysis. Substructuring techniques also find broad applicability to the various types of computer hardware used by engineers today. Efficient use of both mainframes and virtual memory minicomputers with either serial or pipeline processors has been demonstrated [1.3]. In view of independent substructures, the possible adaptation of the software to a system of independently operating processors (or computers) under the logical con-

trol of a single machine becomes quite attractive.

Many investigators have presented extensions of the substructuring approach from static analysis to dynamic analysis of finite element models. However, these efforts are limited to one level of substructures. No attempt has yet been made to formulate and implement multilevel substructuring for dynamic analysis in a general purpose FEM system. Thus, the techniques are not proven effective from the practical viewpoint of large scale structural analysis. The need exists for a comprehensive dynamic analysis system capable of processing multilevel substructured models and incorporating nonlinear response into the solution.

1.2 Objectives and Scope

The objectives of this work are fourfold:

1. To review the state-of-the-art of the multilevel substructure methodology and modeling procedures for the static analysis of complex structures by the FEM. Included is a presentation of the design and implementation of the required software and an illustration of the modeling technique by way of example problems.

2. To review the analytical formulations and computational procedures available for the analysis of complex structural systems subjected to time-dependent loads and capable of linear or nonlinear response. Emphasis is placed on methods for reducing the size of the finite element model for dynamic analysis. Also studied are eigenproblem solution procedures, solution of the equations of motion, and formulations for tracking nonlinear response.

3. To identify the reduction and computational procedures most suitable for incorporation into a general FEM software system with multilevel substructuring capabilities.

4. To discuss the implementation considerations of the above relative to a multilevel substructured environment.

The discussion of dynamic analysis methods is based on a review and interpretation of the open literature. While the authors, along with several other researchers, have experience in large scale linear and nonlinear static analysis with substructuring, large scale dynamic analysis has been performed using only the simplest of techniques.

The remainder of this report is divided into chapters which identify the major topics covered. Chapter 2 is a presentation of the methodology and implementation procedures of multilevel substructuring in a general purpose software system. Methods for reducing the order of the coefficient matrices in the differential equations of motion are reviewed and evaluated in Chapter 3. Chapter 4 considers the various computational algorithms required in dynamic analysis. Included are eigenproblem solution methods, procedures for solving the differential equations of motion, and selected minor topics. The nonlinear continuum mechanics equations for finite deformation, cast in matrix form, are presented in Chapter 5. Exact forms for both the Total Lagrangian and the Updated Lagrangian approaches are described in addition to the finite deformation theories of elasto-plasticity. Chapter 6 presents specific forms of the finite element matrices for the Total and Updated Lagrangian approaches. Details of the transient solution procedure based on an implicit integration operator with the effects of substructuring are discussed. Chapter 7 describes the input language designed to provide a convenient user interface with the application software. A summary of work performed thus far is presented in Chapter 8 along with proposed future activities necessary for successful dynamic analysis of multilevel substructured models.

1.3 Notation

Most of the notation used in the following chapters is defined as it is introduced. However, the following list may prove to be a useful reference. Consistency is maintained whenever possible and exceptions are noted in the text. The complexity of the notation used in Chapters 5 and 6 warrants special discussion in those chapters.

$\{\ \}$	a vector
$[\ ]$	a matrix
$p(\lambda)$	characteristic polynomial
$x_g - y_g$	global coordinate system
$x_l - y_l$	local coordinate system
$\{q\}$	vector of generalized displacement coordinates
$\{u\}$, $\{\dot{u}\}$, $\{\ddot{u}\}$	displacement, velocity, and acceleration vectors in geometric coordinates
$\{u^m\}$, $\{u^s\}$	displacement vectors for master and slave DOF
$[x_j]$	matrix of trial mode shapes for iteration "j"
K.E.	kinetic energy for a structure
S.E.	strain energy for a structure
$[A]$	a tridiagonal matrix
$[B]$, $[G]$	an interaction matrix used in simultaneous iteration methods
$[C(t)]$	time dependent damping matrix
$[D]$	substructure dynamic stiffness matrix
$[K]$, $[M]$	stiffness and mass matrices or submatrices

$[K_G]$, $[M_G]$	Guyan reduced stiffness and mass matrices
$[K_F]$, $[M_F]$	fixed-interface substructure reduced stiffness and mass matrices
$[K_i^{jk}]$, $[M_i^{jk}]$	submatrices of generalized stiffness and mass for substructure "i"
$[\overset{*}{K}]$, $[\overset{*}{M}]$	stiffness and mass matrices assembled from lower level substructures
$[\overset{*}{K}^{mm}]$, $[\overset{*}{M}^{mm}]$	stiffness and mass submatrices for the system master DOF within $[\overset{*}{K}]$ and $[\overset{*}{M}]$
$[L]$	lower triangular matrix of Choleski factors of a stiffness matrix
$[M(\omega)]$	dynamic mass matrix
$[P]$, $[Q]$	orthogonal transformation matrices
$\{P(t)\}$	time dependent load vector
$[R]$	an upper triangular matrix
$[S]$	symmetric coefficient matrix for an eigenproblem in standard form
$[T_G]$, $[\phi^c]$	Guyan transformation matrix (static constraint modes)
$[T_\omega]$	frequency dependent transformation matrix
ω_i^2	square of the substructure natural frequency for mode "i"
$[\omega^2]$	diagonal matrix of substructure natural frequencies squared
Ω^2	unknown system frequency
λ, $[\lambda]$	an eigenvalue, diagonal matrix of eigenvalues
$[\phi]$	matrix of substructure normal modes of vibration
$[\bar{\phi}^n]$	set of retained substructure normal modes

1.4 References

1.1 Craig, Roy R. Jr. and Bampton, Mervyn C. C., "Coupling of Substructures for Dynamic Analyses," AIAA Journal, Vol. 6, No. 7, pp. 1313-1319 (1968)

1.2 Dodds, Robert H. Jr. and Lopez, L. A., "Substructuring in Linear and Nonlinear Analysis," International Journal for Numerical Methods in Engineering, Vol. 15, pp. 583-597 (1980)

1.3 Noor, Ahmed K., Kamel, Hussein A., and Fulton, Robert E., "Substructuring Techniques -- Status and Projections," Computers and Structures, Vol. 8, pp. 621-632 (1978)

CHAPTER 2

MULTILEVEL SUBSTRUCTURE STATIC ANALYSIS

2.1 General

Complex structures frequently consist of repeated, identical components. This may be dictated by economic, construction, or symmetry constraints. The boundaries between components (either real or artificial) partition a complex structure into a natural system of simpler substructures. Each substructure may in turn be partitioned to exploit additional repetition. The associated finite element model generation process is considerably simplified through the repeated use of previously defined components. In many cases, the computational costs of analysis are reduced accordingly. This concept of substructured modeling has been termed the "superelement" technique in the literature due to the similarity of the substructure merging process with that used to connect finite elements. The term "user-defined" has also been employed to distinguish analyst specified substructuring from automatic partitioning of the equilibrium equations.

The structural frame of an aircraft provides the classic example to illustrate the concepts and advantages of a substructured analysis. Independent design groups develop the individual substructures, for example: the wing assembly, fusealage sections, and vertical stabilizer. The substructures interface through relatively small boundaries (in terms of the number of nodes). Even with such "first-level" substructuring, the number of nodes and elements may be too large for efficient processing of the substructures. The same substructuring process, in

theory, can be repeated within each first-level substructure to introduce second, third, fourth, ... level substructures. The engine nacelles, shrouds, flaps, ribs, and skin panel sections within a wing assembly may comprise substructures five or six levels removed from the "highest" level structure that represents the complete airframe. The conceptual organization for this type of structural model parallels that of a tree. The tree has a single root node (the highest level structure). Any number of substructure levels may be defined below the root node. No theoretical limit exists on the number of branches that enter a node (substructure) at level "l" from level "l-1". All terminal nodes of the tree are individual finite elements.

Substructure techniques have been utilized extensively by the Norweigan ship building industry [2.3] to construct and analyse finite element models of oil tankers. Repeated bulkheads and common stiffener arrangements in ships are well suited for substructuring. Without multilevel substructures, typical analyses would involve 100-150 thousand degrees of freedom. Problems of this size remain impractical to solve despite the availability of super-computers. Both the airframe and ship building examples clearly demonstrate the usefulness of multi-level substructuring to support practical analyses.

In the context of linear, static analysis, a substructured model yields the same solution (to within round-off errors) as a "standard" model that considers the structure as a single collection of nodes and elements. The solution equivalence remains valid for static, nonlinear analysis provided the substructures experiencing nonlinear behavior are included in the iterative solution. Normally, regions of a structure

that remain linear are substructured and eliminated via static condensation from the iterative solution. For example, it is a simple matter to anticipate the plastic deformation that occurs near a stress concentration. Portions of the structure removed from the stress concentration are defined as linear substructures and condensed. Nonlinear finite elements and reduced linear substructures comprise the highest level structure for the iterative solution. Linear substructures simply provide elastic restraint on the nonlinear region. In such cases, the standard and substructured models again yield identical solutions. However, the substructured model generally requires much less computational effort as a consequences of its reduced size.

This chapter provides detailed background information on substructured modeling and solution procedures for static analysis. The advantages of substructured analysis relative to the standard procedure are first described. The literature concerned with static substructuring is reviewed. Various techniques adopted to address the user-software interface are described and a simple example is presented to illustrate the techniques used in the POLO-FINITE system. Computational and software issues that arise in programs for general substructured analysis are surveyed. The chapter concludes with a study of two example problems that illustrate the computational savings possible with substructured models for linear and nonlinear static analysis.

2.2 Substructured vs Standard Models

Compared with a standard modeling and solution procedure, a multi-level substructured approach offers a number of advantages. These include:

1. Input data requirements are reduced. Geometric and topologic descriptions of a substructure are specified only once. When the same substructure appears repeatedly at higher levels, input data that must be provided by the user is significantly reduced.

2. The impact of design changes on reanalysis costs can be reduced. Stiffness matrices and loading vectors for only the modified substructures are computed during reanalysis.

3. Models may be generated independently. Because substructures have clearly defined interfaces, the design and modeling groups may work almost independently. Element numbering, node numbering, and load case naming schemes are usually independent across substructures which simplifies model generation.

4. Isolated substructures may be independently verified. Each substructure may be constrained, loaded, and analyzed to study the response of the isolated model. Checkout analyses are usually inexpensive relative to the cost of a complete structure analysis. The higher execution priority assigned to jobs that request fewer machine resources also decreases the total analysis time. A complete analysis performed in smaller segments frequently costs less, and requires less residency time, than a comparable standard analysis performed in a single execution.

5. Identical substructures may be used repeatedly. This is the most often cited advantage of a substructured model. Substructure quantities, for example the reduced stiffness matrix, are computed but once and used repeatedly to form other structure stiffness matrices. The degree of computational savings is highly problem dependent with cost reductions in the range of 2 to 100 having been reported [2.2]. The level of savings in nonlinear analysis depends on the degree of size reduction and the frequency of tangent stiffness updates.

6. Numerical conditioning problems are often reduced. Certain types of numerical problems may be remedied through the use of a substructured modeling and solution procedure. A common

situation involves the joining of a very stiff substructure to a comparatively flexible one which results in a loss of precision during stiffness assembly (a very large number overwhelms the much smaller term). Condensation of a very stiff substructure frequently reduces the magnitude of stiffness coefficients for the remaining nodes to a level comparable with those of the adjacent, more flexible component. The loss of precision in the important stiffness assembly process can thus be minimized without extended precision arithmetic.

7. Exposure to machine failure is reduced. The solution of a structure with a very large number of DOF may require long residency times on multipurpose computer systems. During this time, the executing program is exposed to the possibility of a machine failure that would require restarting the analysis. Procedures have been developed, termed checkpoint restart, that save snapshots of the program status on disk files at specified intervals. This process often requires extensive machine dependent coding which restricts the software portability. Substructured models provide a more natural solution to the failure protection problem. Substructures are processed in a logically independent, sequential manner during execution. Natural breakpoints occur between each substructure at which the the execution may be terminated and the databases saved on tape. If a failure occurs while the next substructure is being processed, the databases on tape are simply restored to disk and the analysis resumed.

These advantages of a substructured approach are equally applicable in all types of analyses -- static, dynamic, linear, and nonlinear.

2.3 Literature Survey

Although the concepts and equations of substructured analysis are generally well known for static analysis, relatively few papers on the subject have appeared in the literature. A contributing factor to this apparent lack of interest is the enormous software complexity required to support substructuring, coupled with the limited availability of software to researchers. For static linear and nonlinear analysis, the governing equations are straightforward and simple to derive. The few researchers who have examined substructuring have focused on improving

the computational efficiency and on the computer implementation problems. Because only approximate substructured solutions are feasible in dynamic analysis, the literature concerned with improving the algorithms continues to expand and is reviewed in the next chapter. This section reviews previous studies that addressed static, linear and nonlinear substructured analyses.

In the early 1960s, Przemieniecki [2.15] presented the first comprehensive formulation for substructured analysis using the conventional displacement method. Taig [2.17] described an attempt to implement these procedures for general analysis. These two early studies viewed substructuring as automatic partitioning of the equilibrium equations to overcome computer size limitations. Interest in substructured analysis declined during the late sixties when commercial programs using sparse matrix techniques became operational on third generation computers. The non-computational aspects of analysis, including model generation and technical coordination, replaced computer limitations as the major problem areas. Substructure techniques regained appeal as an approach to solve these problems but in the form of analyst defined, rather than automatic partitioning. Thus the term "user-defined" substructuring was coined. In this same period, Williams [2.18] showed through operation counts that equation solving with sparse matrix techniques can never be more computationally efficient than a substructured solution, when the substructure arrangement is suitably defined.

Furuike [2.5] described a software system capable of processing three levels of substructures. The root and the next two levels of the tree may contain only substructures. Fourth level structures consist entirely of finite elements. The analyst supplies, through input data, the order of substructure processing and the proper sequencing of nodal DOF across substructure boundaries. Substructure stiffness condensation is performed with the inefficient inversion algorithm described in Section 2.5.1. The author presents several example solutions with and without substructuring that demonstrate the numerical equivalence of the results. Unfortunately, no comparisons of computer resources are provided.

Egelan and Araldsen [2.3] briefly surveyed the substructuring capabilities of the SESAM-69 program. No details of the user-interface for describing the substructure connectivity and orientation are provided. SESAM supports multilevel substructured models. Finite elements and substructures cannot be mixed at the same level in SESAM. This restriction is imposed by the adoption of separate assembly processors for structures composed of only elements and those composed of only substructures. The classical inversion algorithm for stiffness condensation is implied in the survey.

Descriptions of substructuring capabilities in NASTRAN are given by MacNeal and McCormick [2.11] and in ASKA by Schrem [2.16]. Each system initially supported only modest substructuring; the major emphasis being directed toward efficient sparse matrix techniques. Proprietary versions of NASTRAN have been expanded to support more comprehensive substructuring, including some techniques for dynamic analysis beyond Guyan

reduction. Software details are not publicly available. To perform a substructured analysis, users must write "driver" programs in a special language to manipulate disk files and to control the execution of processing modules. Completely automated solution procedures for substructured analysis are not available.

More recently, Peterson and Popov [2.14] addressed the computational penalty that occurs with stiffness matrix rearrangement prior to condensation. They propose a scheme to eliminate nodal freedoms at the element level before assembly of the condensed stiffness. The technique is promoted as more efficient than the conventional method of rearrangement although no comparisons of computer time are stated for the example problems.

Lopez [2.9] presented POLO-FINITE as the first major system to support multi-level substructuring as a natural approach to finite element modeling. Major advances incorporated in POLO-FINITE include a very simple substructure definition process and fully automated solution procedures. The dependencies between substructures in the hierarchy are determined completely by the system from basic input. Any number of related and/or independent hierarchies may be defined within a single database, with up to 20 levels of substructuring permitted in each hierarchy. Substructures and finite elements may be mixed at any level; the system processors treat finite elements and substructures identically. The extensive system logic that automatically controls the solution process also enables intelligent reanalysis to incorporate substructure modifications. As input data describing the modifications are processed, substructure results dependent upon the modifications are in-

validated. For example, changing a load case definition invalidates existing displacements for that load case but not the structure stiffness matrix. Prior to reanalysis, the system traverses the hierarchy to determine the type and order of computations required. During the traversal, dependent results for other substructures are destroyed and tagged for recomputation. When a substructure at level "I" is marked for computation, dependent substructures at level "I-1" are also marked for computation. In this manner, the effects of substructure modifications are automatically propagated upward through the hierarchy. POLO-FINITE equation solvers are based on the hypermatrix techniques first introduced in the ASKA system. An efficient "partial decomposition" algorithm is utilized to condense substructure stiffness matrices.

The most recently publicized substructure system, MISA [2.7], was developed by the Japanese ship building industry. MISA incorporates several unique concepts. Wavefront, rather than variable bandwidth, equation solvers are used to condense substructure stiffnesses although no computational advantage is claimed. The software logic deduces the substructure hierarchy from parent-child relations input by the user. A "copy" function facilitates repeated use of previously defined substructures. MISA does not support mixed substructures and finite elements at the same level. It currently analyses linear structures for static and thermal loads, and performs steady-state heat conduction analyses.

Dodds and Lopez [2.1] extended the POLO-FINITE system to support multilevel substructured models for static, nonlinear analysis. The analyst defines linear regions that are substructured and condensed to form effectively elastic supports surrounding the nonlinear, highest

level structure. The reduced size of the nonlinear structure analyzed with the iterative technique frequently yields significant cost savings. Currently, the nonlinear region must be defined prior to beginning the analysis. This is a major disadvantage of the approach. For the initial loading levels, this proves inefficient as a much smaller nonlinear region develops than is actually declared. To improve the situation, it is necessary to consider substructures that can be made nonlinear and brought into the iterative solution as the loading levels increase. This requires yet another level of sophistication in the control logic and has not been attempted.

2.4 User-Software Interface

The user-software interface for general purpose systems must provide sufficient flexibility to invoke the options, and yet it should not discourage the infrequent user with unnecessary details. Substructuring further complicates this interface with the introduction of more elaborate topology, geometry, and computational algorithms. The developers of most software to support substructuring have not made the interface particularly simple for the user. These systems usually incorporate substructuring as an extension of the original software design. Substructuring is viewed as a last resort to obtain a solution, rather than as a natural modeling approach. Users are required to "program" the substructured solution by directly or indirectly invoking computational modules and by manipulating disk files that store substructure data. Consequently, only the most experienced users attempt substructured solutions; casual users are told to avoid substructuring.

The process of defining a substructured model and conducting the analysis consists of five logical steps. These are:

1. The definition of each independent substructure (elements, constraints, and loadings).

2. The elimination of substructure "slave" (condensed) nodal freedoms. "Master" nodal freedoms remain after condensation.

3. The connection of individually defined substructures in a topologic and geometric sense. This requires matching of substructure boundaries to insure displacement compatibility and may also include coordinate transformations of substructure matrices from their local system to a common global reference frame. Additional boundary constraints may also be required.

4. The creation of a loading set hierarchy that parallels the substructure hierarchy. Equivalent nodal loads computed for the loading cases defined in (1) for the individual substructures are also reduced through condensation. Loads reduced to the master nodes serve to define a hierarchy of loading sets on the complete model. For example, the reduced loads may be applied to selected copies of a substructure at the next higher level to create a desired pattern of loading. The loads may also be carried up through the hierarchy with scalar multipliers and possibly combined with other reduced load cases to form new loading cases on a higher level substructure.

5. Requests for computation and output. The complexity of computational requests depends on the level of procedural detail required by the software to effect the solution. Output requests become complex when the user designates the hierarchy level for which results are desired, for which loading cases, elements, nodes, coordinate reference frames, etc.

The common technique to approach these problems involves the use of multiple programs or a single program executed many times. The programs communicate through data stored on disk files (and/or tapes). The analyst is responsible for coordinating the program executions to produce the desired results. A typical analysis with one level of substructures might require that the following tasks be performed by the analyst:

1. Elements, nodal coordinates, topology, constraints and loads are defined to a finite element program for each individual substructure. The analyst provides special instructions indicating a condensation is desired and the list of master (or slave) DOF. Disk files are attached to the program onto which the reduced stiffness matrices and load vectors are written. Usually each substructure must be processed in a separate program execution and with unique data files.

2. A substructure processing program is executed with all substructure disk files attached. This program is provided with disk file numbers for the substructures and the order in which they are to be processed. The connectivity relations between substructure master DOF and the global numbering system are also specified. Additional complexities arise when substructure reference axes are not all parallel. Since only substructures can be processed, it is not possible with such a scheme to mix substructures and simple finite elements at the same level, which is inconvenient in nonlinear analysis. The substructure processing program assembles the reduced stiffnesses and loads to form a final set of equilibrium equations for solution, then computes the displacements for the highest level. Master DOF displacements for user selected substructures are extracted and written onto another set of disk files.

3. Finally, the finite element program is again invoked with proper disk files attached that contain displacements for the substructure master nodes. Backcondensation procedures may then be performed to recover slave displacements and element strains-stresses. This process must be repeated for each copy of the substructure since the computed displacements, unlike the stiffness matrix, are unique.

Each of the above steps involve considerable manipulation of disk files and several program executions by the user. The manpower costs can easily approach those for model generation. The process is almost intractable when more than one level of substructuring is used (if permitted at all by the software). Nonlinear problems require looping over the second and third operations for each load increment. To place numbers on the amount of user effort involved with this approach, a linear model containing four substructures at the first level was analyzed with one of the most successful commercial finite element

programs. Following the recommended procedures in the user documentation, nine separate computer jobs were required to obtain a solution. Absolutely no automatic solution techniques are available in the program. Changes to one of the substructures required a complete reanalysis of the entire structure due to the arrangement of data on tape storage.

Demands placed on the user are greatly simplified when the software is designed to support substructured analysis in an integrated fashion. Standard analyses become the simplest default procedure in such a system. The following example, analyzed with the POLO-FINITE system, illustrates the degree of simplification possible with a user-oriented approach to substructured analyses. This example is not intended to demonstrate savings of computational effort; the computational advantages are demonstrated in more complex examples at the end of this chapter.

The structure is a simple two span, planar truss as shown in Fig. 2.1. The generally non-symmetric loads requires that a full model (both spans) be analyzed. Components of the substructured model are shown in Fig. 2.2, with names assigned to each component for identification in the model description. Figure 2.3 illustrates the substructure hierarchy in tree form for this simple example. One span is defined and condensed to the four nodes necessary for connection with the adjacent substructure. The bridge is defined using two copies of the condensed substructure and three rod elements to complete the model.

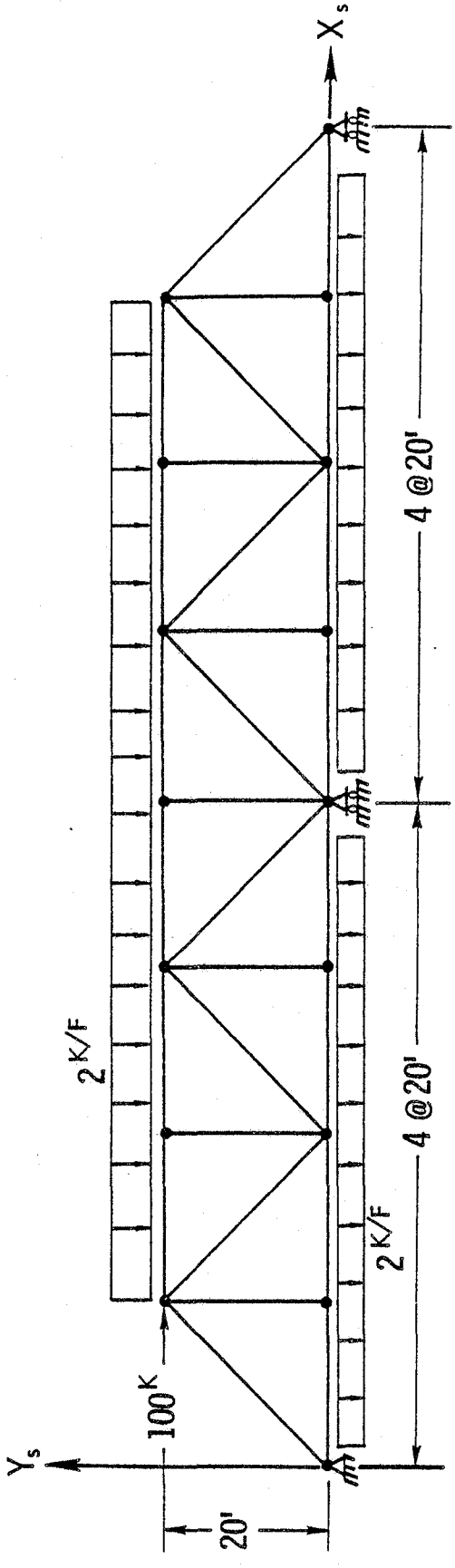


Figure 2.1 -- Bridge Structure to Illustrate Substructuring

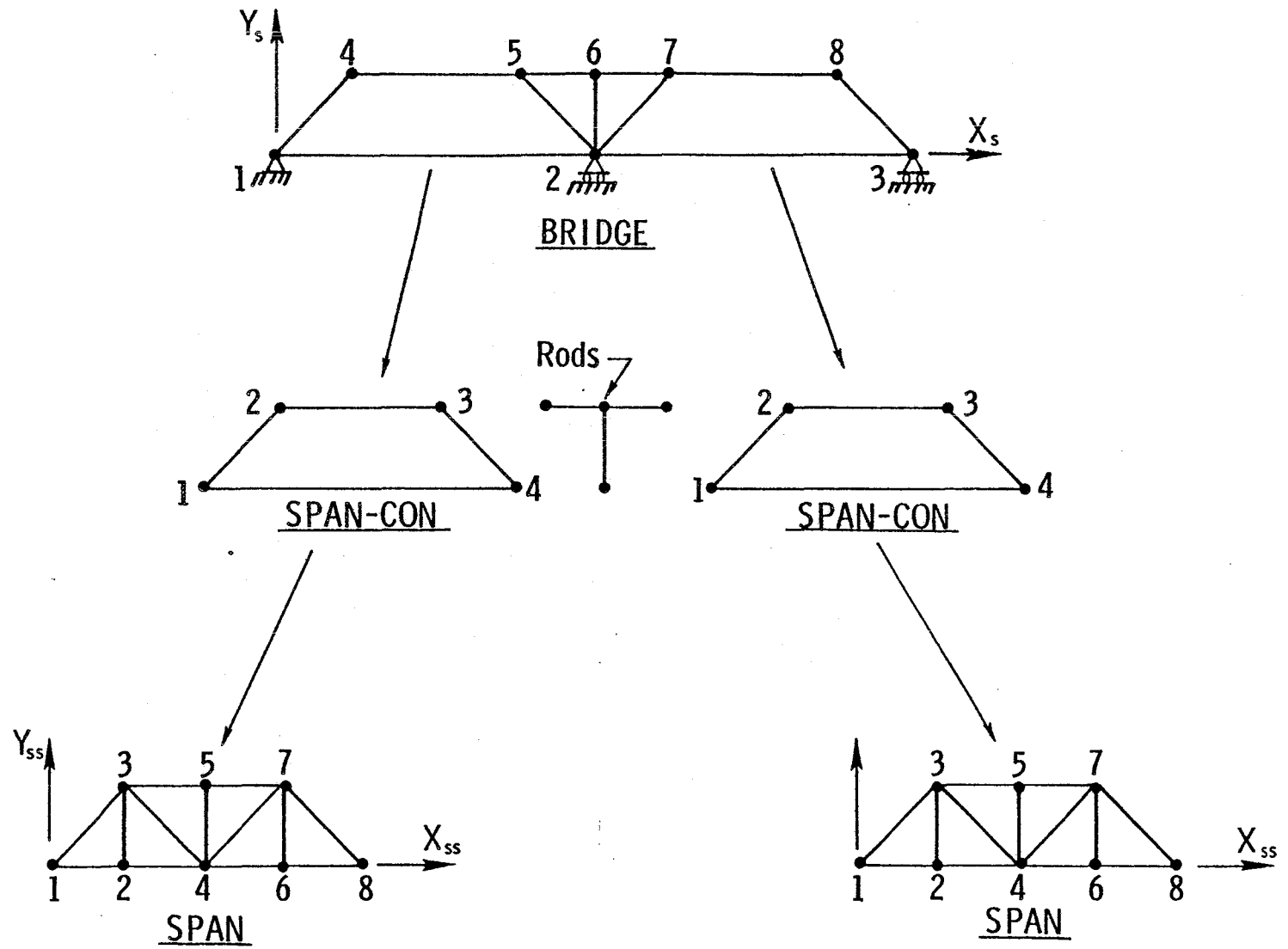


Figure 2.2 -- Substructured Model for Bridge Example

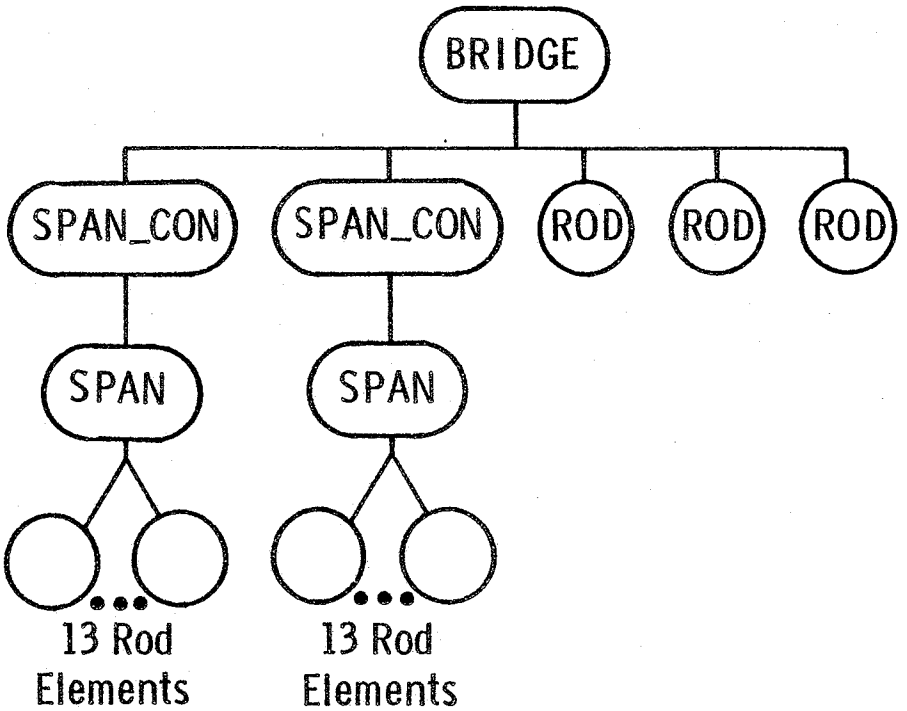


Figure 2.3 -- Tree Representation of Structural Hierarchy for Bridge Example

The lowest level (sub)structure, SPAN, consists of 8 nodes (each with 2 DOF) and 13 rod elements. Input data describing structure SPAN to FINITE are listed in Fig. 2.4. Element types, properties, topology, and nodal coordinates are first specified. The problem oriented language (POL) input eases data entry by eliminating column and command order restrictions. No natural boundary conditions occur at the nodes eliminated by condensation. Constraints are thus omitted on this lowest level substructure. Three independent loading cases are defined to act on SPAN. These represent a uniform load applied over the bottom chord, a uniform load on the top chord, and a lateral load acting on the top chord. The magnitude of each loading is unity to simplify the definition of actual loading magnitudes in the higher level structure.

Structure SPAN_CON is defined as the statically condensed version of SPAN. Figure 2.5 lists the input data describing this structure. Nodes 1, 3, 7, and 8 are retained after condensation. SPAN_CON in this example corresponds to a "super-element" in the terminology used by some researchers. Analysts explicitly introduce condensed substructures into the hierarchy through intermediate structures such as SPAN_CON. Structure SPAN is referred to as the "child" of the "parent" structure, SPAN_CON, which resides at the next higher level in the structure tree. This technique has proven to be a natural means of incorporating the condensed version of the substructure into the hierarchy. It eliminates confusion on the analyst's part and maintains a consistent definition of structures in the database. Some structures are simply tagged as "condensed" which serves to control execution of the processors. The incidences specified for element one of SPAN_CON designate the nodes

```
C
C       PROBLEM UNITS ARE KIPS, FEET
C
STRUCTURE SPAN
  NUMBER OF NODES 8 ELEMENTS 13
  ELEMENTS ALL TYPE ROD E 4.32E06 AX 0.0347
  COORDINATES
              X       Y
    1         0       0
    2        20       0
    3        20      20
    4        40       0
    5        40      20
    6        60       0
    7        60      20
    8        80       0
  INCIDENCES
    1     1  3
    2     2  3
    3     3  4
    .     .  .
    .     .  .
  LOADING UNIT_TOP
    NODAL LOADS
       3 7 FORCE Y -10
       5   FORCE Y -20
  LOADING UNIT_BOTTOM
    NODAL LOADS
       2 4 6 FORCE Y -20
  LOADING UNIT_SWAY
    NODAL LOADS
       3 FORCE X 1.0
```

Figure 2.4 -- POLO-FINITE Input Data for Structure SPAN

```
STRUCTURE SPAN_CON
  NUMBER OF NODES 4 ELEMENTS 1
  ELEMENT 1 TYPE SPAN CONDENSED
  INCIDENCES
    1    1 3 7 8   $ BECOME NODES 1-4
  LOADING UNIT_TOP
    EXTERNAL ELEMENT LOADS
       1     UNIT_TOP
  LOADING UNIT_BOTTOM
    EXTERNAL ELEMENT LOADS
       1     UNIT_BOTTOM
  LOADING UNIT_SWAY
    EXTERNAL ELEMENT LOADS
       1     UNIT_SWAY
```

Figure 2.5 -- POLO-FINITE Input Data for Structure SPAN_CON

retained from substructure SPAN. FINITE currently requires that all DOF at a node be either eliminated or retained during the condensation process. Loading cases are carried up through the structural hierarchy with a loading type designated EXTERNAL ELEMENT LOADS. An EXTERNAL loading specifies the names of loading cases on the child substructure. The loads are condensed and placed on the parent substructure nodes under the declared load case. There is a one-to-one correspondence between loading case names for the parent and child substructures in this example, but this is not required. Only loading case names within a structure must be unique. Any number of loading cases, with optional scalar multipliers, may be selected from the child substructure to construct loadings on the parent substructure. After the stiffness and loadings on SPAN are condensed during solution, SPAN_CON has the major characteristics of any other structure; it has a stiffness matrix and loading cases stored in a standard format.

Structure BRIDGE is modeled from two copies of SPAN_CON with three additional bar elements to complete the model. Figure 2.6 lists the input data describing structure BRIDGE. Copies of SPAN_CON (elements 1 and 2) are placed in BRIDGE with the same orientation relative to the coordinate system in which they are defined. In this model, substructure reference axes X_l-Y_l and the highest level structure axes X_g-Y_g are parallel. In other cases, a rotational transformation of substructure stiffness matrices and nodal loads may be required for proper geometric alignment. Coordinates are required for nodes of the three additional bar elements. Coordinate values are used to determine element size and orientation, thus the origin location is immaterial. For linear

```
STRUCTURE BRIDGE
  NUMBER OF NODES 8 ELEMENTS 5
  ELEMENTS
    1 2 TYPE SPAN_CON ROTATION SUPPRESSED
    3-5 TYPE ROD E 4.32E06 AX 0.0694
  COORDINATES
          X     Y
  2       0     0
  5     -20    20
  6       0    20
  7      20    20
  INCIDENCES
    1   1 4 5 2
    2   2 7 8 3
    3   5 6
    4   6 7
    5   2 6
  CONSTRAINTS
    1-3 V = 0.0
    1   U = 0.0
  LOADING FULL_TOP
    EXTERNAL ELEMENT LOADS
       1 2   UNIT_TOP 2
    NODAL LOADS
       5 7   FORCE Y -20
       6     FORCE Y -40
  LOADING LEFT_BOTTOM
    EXTERNAL ELEMENT LOADS
       1     UNIT_BOTTOM
       .
       .
  COMPUTE DISPLACEMENTS FOR STRUCTURE BRIDGE
  [ output requests ]
  STOP
```

Figure 2.6 -- POLO-FINITE Input Data for Structure BRIDGE

analysis, mixing of substructures and elements is more a convenience for model definition than a computational necessity. Constraints imposed on the nodes of structure BRIDGE model the simple support boundary conditions. EXTERNAL loads are again used to apply loadings from SPAN_CON to BRIDGE. Load cases on BRIDGE are defined by selecting external loads on elements 1 and 2. Nodal loads, standard element loads (e.g., a distributed force on an element), and external element loads may be combined in any manner to define the loading cases on BRIDGE.

A request for analysis has the simple form COMPUTE DISPLACEMENTS ... as shown in Fig. 2.6. FINITE processors traverse the hierarchy to automatically determine the order of processing required and to check for topological consistency. The solution then proceeds to completion without user intervention. A solution in this context implies the computation of displacements for structure BRIDGE. The hierarchy traversal performed for each COMPUTE DISPLACEMENTS command insures that only needed quantities are actually generated.

Output requests provide the capability to designate the unique copy of a substructure for which results are desired in addition to other information such as loading cases, element and node numbers, and coordinate systems for stresses-strains. The "substructure stack" is a part of the structure name in the OUTPUT command. Thus, to obtain stresses in all bar elements for the right span of BRIDGE for all loading cases, the following command is sufficient:

OUTPUT STRESSES FOR STRUCTURE BRIDGE/2/1

in which the 2/1 designates subelement 2 of structure BRIDGE, which is a structure named SPAN_CON, then subelement 1 of SPAN_CON, which is a structure named SPAN. The list of elements and/or nodes (in this case an implied "all" elements) refers to the final structure listed in the stack. FINITE processors examine the substructure stack and automatically determine the type and order of backcondensation processes required to satisfy the request.

The OUTPUT command provides considerable flexibility for requesting substructure results. The above command represents an extreme case in which the stack points to a unique occurence of a substructure in the hierarchy. Alternatively, the command

OUTPUT STRESSES FOR STRUCTURE BRIDGE

requests the printing of results for all elements in structure BRIDGE. In this case, FINITE processors automatically traverse the complete hierarchy below structure BRIDGE, recovering displacements, strains, and stresses at every level. The output processor traverses the same stack to print the results from the top (BRIDGE) down.

The major advantages of the above approach to substructured model definition and solution are now apparent. The complete problem definition and solution is accomplished in a single computer run with no user interaction concerning the placement of substructure information onto data files. In effect, the analyst has access to a structural model database which can be defined and modified at a very high (logical) level. Substructure model description is a completely natural extension

of standard model description procedures. When implemented in this manner, computation requests are identical with and without substructuring. The traversal processes are performed automatically regardless of the model complexity. Output commands are quite flexible in that results for a wide or limited range of substructures may be requested. The POL input, while not required, considerably simplifies data entry and provides ready documentation for the model. It is mandatory for interactive processing.

FINITE currently requires that the structural hierarchy be defined in an inverted order. For example, structure SPAN must exist in the database at the time structure SPAN_CON is defined. Similarly, structure BRIDGE cannot be defined unless structure SPAN_CON exists. The structural tree must be defined from the bottom up; however, each branch need not be completely defined before beginning another branch. This restriction is sometimes inconvenient in that descriptions of a tree from the top down may be more natural.

2.5 Computational and Software Issues

Multilevel substructuring creates a number of computational and software problems not encountered in conventional finite element systems. This section provides a brief survey of the major computational and software issues. The algorithmic details of stiffness condensation, load reduction, and displacement recovery for an isolated substructure are first reviewed. The logical control techniques and data structures required to automatically process complex substructured models are then examined. These non-numerical aspects of the software

determine, to a large extent, the eventual generality and user-acceptance of the system. The final topic discussed in this section concerns data structures and algorithms for the solution of very large sets of linear equations. The less widely known hypermatrix procedure is described. Hypermatrix computations offer a number of advantages compared to other techniques, including skyline and wavefront, especially when the implications of a paged virtual memory system and parallel-pipeline hardware are considered. Hypermatrix techniques are also advocated later in this report for eigenproblem solution of very large systems.

2.5.1 Substructure Reduction

The three major computational activities associated with processing an isolated substructure include:

1. Condensation of the substructure stiffness matrices to eliminate the slave nodal freedoms;
2. Condensation of the equivalent nodal loads applied to the slave nodes;
3. Recovery of slave node displacements once master nodal displacements are known from solutions of higher level structures.

The classical equations for these operations were first presented by Przeimienicki [2.15] and are included here for completeness. The formal procedures are inefficient and seldom followed. Two additional condensation procedures, referred to as coordinate transformation and partial decomposition, have better efficiency and are discussed in detail.

Equilibrium equations for an isolated substructure are first partitioned into two sets corresponding to the slave and master nodes as

$$\begin{bmatrix} K^{ss} & K^{sm} \\ K^{ms} & K^{mm} \end{bmatrix} \begin{Bmatrix} u^s \\ u^m \end{Bmatrix} = \begin{Bmatrix} p^s \\ p^m \end{Bmatrix} \tag{2.1}$$

where the superscripts m and s denote master and slave nodes respectively. The number of slave DOF is designated by "p", the number of master DOF by "q". Constraints forcing a slave displacement to have a prescribed value and multi-point constraints between slave displacements are assumed to have been imposed by modification of the coefficient matrix and load vector(s). The solution of Eq. (2.1) in partitioned form follows the standard procedures summarized below:

$$\{u^s\} = [K^{ss}]^{-1}\{p^s\} - [K^{ss}]^{-1}[K^{sm}]\{u^m\}, \tag{2.2}$$

$$\left([K^{mm}] - [K^{ms}][K^{ss}]^{-1}[K^{sm}]\right)\{u^m\} = \{P_{eff}\}, \tag{2.3}$$

$$\{P_{eff}\} = \{P^m\} - [K^{ms}][K^{ss}]^{-1}\{P^s\}. \tag{2.4}$$

These equations reveal the form of the condensed stiffness matrix, Eq. (2.3), the reduced load vector, Eq. (2.4), and the procedure to recover slave node displacements from known master node values, Eq. (2.2). Their inefficiency derives from the computation of $[K^{ss}]^{-1}$. The computational penalty increases dramatically when $[K^{ss}]$ has a narrow bandwidth but a fully populated inverse (the most common case). Operation counts for the computations are given later in this section for comparing the reduction methods.

In the second method for condensation, the slave node displacements are related to the master node displacements through a coordinate transformation matrix, [T], such that

$$\{u^s\} = [T]\{u^m\}. \tag{2.5}$$

Each column of [T] contains the displacements of the slave nodes for a unit value of one master node displacement component, all other master DOF displacements being held zero. Because these displacements represent deformed substructure shapes that are analgous to mode shapes in dynamics, they are often referred to as "static constraint modes". The matrix [T] is evaluated by substituting Eq. (2.5) into the first row partition of Eq. (2.1) which yields, in the absence of external loading,

$$[K^{ss}][T]\{u^m\} + [K^{sm}]\{u^m\} = \{0\}. \tag{2.6}$$

After eliminating $\{u^m\}$ from both sides of this equation, the product of $[K^{ss}]$ with each column of [T] equals the corresponding column of $-[K^{sm}]$.

Equation solving techniques are thus applicable to compute the columns of $[T]$. The matrix $[K^{ss}]$ is triangulated only once using the front, Choleski or some other decomposition scheme. The procedure may be relatively efficient since $[K^{ss}]$ is often banded (the natural ordering of slave DOF is not altered in forming these equations). The columns of $[T]$ are obtained by successive forward and backward substitution over the columns of $-[K^{sm}]$. An expression for the condensed stiffness matrix is obtained by equating the strain energy of the substructure with and without the coordinate transformation. Thus,

$$[\overset{*}{K}] = [T]^T[K^{ss}][T] + [T]^T[K^{sm}] + [K^{ms}][T] + [K^{mm}]. \quad (2.7)$$

Using the symmetry of off-diagonal submatrices, $[K^{sm}]$ and $[K^{ms}]$, and the symbolic form of $[T]$, the right side of Eq. (2.7) can be expanded and simplified. The form of the condensed stiffness for computation becomes

$$[\overset{*}{K}] = [K^{mm}] + [K^{ms}][T]. \quad (2.8)$$

Eq. (2.8) is often written in an unexpanded form that includes an identity matrix in the definition of $[T]$. The condensed load vector for the master DOF is related to the slave DOF load vector by $[T]$ in a similar manner using contragredience

$$\{\overset{*}{P}\} = \{P^m\} + [T]^T\{P^s\}. \quad (2.9)$$

A third technique for substructure condensation, which has been widely adopted, employs a "partial decomposition" of the stiffness matrix. Wilson [2.19] provides a detailed explanation of the computational procedures. Equations for an isolated substructure are assembled and partitioned into slave and master nodal DOF as in Eq. (2.1). Gauss or the more efficient Choleski decomposition is applied to completely eliminate the first "p" rows corresponding to the slave DOF. Row-wise storage and decomposition of the lower triangle is assumed in this discussion. Similarly, the master-slave coupling terms of $[K^{ms}]$ are reduced following standard procedures for off-diagonal terms. The Choleski reduction formulas applicable to the slave DOF are

$$l_{ii} = \left(k_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2} \quad ; \quad i \leq p$$

$$l_{ij} = \frac{k_{ij} - \sum_{k=1}^{j-1} l_{ik}\, l_{jk}}{l_{jj}} \quad ; \quad \begin{matrix} i \leq p \\ j < i \end{matrix} \qquad (2.10)$$

Lower limits of k=1 on the summations imply a fully populated coefficient matrix. Extension to accomodate a variable bandwidth is straightforward.

A partial decomposition is then performed on the remaining $[K^{mm}]$ submatrix of master DOF coefficients to eliminate the coupling effect of slave DOF in the matrix $[K^{ms}]$. This is accomplished by terminating the normal summations in Eq. (2.10) at column "p" for the terms i,j > p. The master DOF stiffness terms are modified by this process to reflect the original coupling between slave-slave DOF and slave-master DOF entirely within the master-master DOF terms. The modified submatrix $[K^{mm}]$

is the desired condensed stiffness matrix for the substructure.

Final elimination of the master DOF occurs during solution of the highest level structure. The basis for partial decomposition is that slave nodes within a substructure do not have a topological connection with nodes elsewhere in the hierarchy. Thus, standard decomposition of the complete structure stiffness, generated without substructuring, simply skips the summations that involve zero coupling terms between slave DOF in the equivalently substructured model.

Condensation of substructure loading vectors is accomplished through a forward reduction using the partially triangulated stiffness. Summations for master DOF terms are again terminated at column "p". Condensed loads for the master DOF reside in the last "q" rows of the load vectors. The first "p" rows contain partial displacements of slave DOF ready for backsubstitution. These are termed the "partial slave displacements."

Recovery of slave DOF displacements is performed by completing the backward substitution over the first "p" rows once master DOF displacements are available for the next higher structure. Before the back substitution is begun, the partial slave displacements generated and saved during load condensation are placed in the first "p" rows of the load vector. Although this appears to be a trivial task, it becomes a very complex logic problem when users are permitted to define load cases on higher level structures as combinations of condensed loading cases. In effect, users may implicitly define new load cases on the lower level substructure for which partial slave displacements are not computed during load condensation. The displacement recovery logic must

traverse the hierarchy again from the top, down to the level being processed to generate the loading combinations defined implicitly. Only then can the correct partial slave displacements be computed by summing the separate load case values with the implicitly defined multipliers.

Computational efficiency is always of concern when large matrices are manipulated. Experience with a large number of analyses has shown that generation of the condensed stiffness matrix requires nearly all of the computational effort associated with processing an individual substructure. Load case condensations and slave displacement recovery combined seldom require 10% of the effort for stiffness condensation. Each of the algorithms described above involve extensive summations to generate resultant terms. The number of multiplications performed in evaluating these summations provides a measure of their relative efficiency. Operation counts were developed for the stiffness condensation phase of the three algorithms based on the following assumptions:

1. Matrix $[K^{ss}]$ is symmetric, has "p" rows, and an average half bandwidth of "r";
2. Matrix $[K^{ms}] = [K^{sm}]^T$ and is fully populated as a consequence of DOF reordering necessary to partition slave and master DOF for condensation;
3. Matrix $[K^{mm}]$ is fully populated.

In the best possible case matrix $[K^{ms}]$ is upper triangular; the fully populated worst case is assumed here. Matrix $[K^{ss}]$ will nearly always be banded. With these assumptions, it is a simple task to estimate the number of multiplications required for each algorithm. The results are tabulated below:

1. Classical Inversion: $p(0.5r^2 + pr + pq + q^2)$

2. Coordinate Transformation: $p(0.5r^2 + qr + pq + q^2)$

3. Partial Decomposition: $0.5p(r^2 + pq + q^2)$

To illustrate the effort required for each algorithm, consider a substructure for which p=500, q=50, and r=100; only 10% of the nodal DOF are retained after condensation. The operation counts for each algorithm are listed below:

1. Classical Inversion: 4.125×10^7

2. Coordinate Transformation: 1.875×10^7

3. Partial Decomposition: 0.940×10^7

The results demonstrate clearly the inefficiency of the classical inversion algorithm employed in early substructure software. The coordinate transformation algorithm, which is explicitly required for dynamic reduction, and the partial decomposition algorithm are considerably more efficient. The effort for coordinate transformation rapidly approaches that for classical inversion when "q" nears the value for "p", that is, when a larger percentage of substructure DOF are retained after condensation. The results also demonstrate that the partial decomposition procedure should always be used for static analysis.

2.5.2 Logical Control and Data Structures

The numerical algorithms described in the previous section are applicable for an isolated substructure. Before the computations for a substructure may begin, the software logic must determine the correct order in which to process the substructures. The proper ordering depends on the type of operations to be performed (such as stiffness assembly or displacement recovery) and the topologic relationships between substructures, as represented by the hierarchical tree. This is not trivial task for models that consist of forty or fifty substructures distributed through five and six levels in a hierarchy. Finite element models with such characteristics are quite common for the analysis of large, geometrically complex structures. This section first outlines the familiar techniques used in standard analysis programs. An overview is then given of the techniques required to support general substructured analysis as illustrated in the POLO-FINITE example of Section 2.4.

Programs that support only standard, static analysis have comparatively simple control and data structure problems. A solution is accomplished by initiating a set of processing modules in a completely predetermined order that does not vary from one analysis to another. The typical sequence is: (1) read the input data, (2) compute the element stiffnesses, (3) assemble the structure stiffness and triangulate, (4) generate the loading vectors and perform a load pass to obtain displacements, (5) compute element strains and stresses, and (6) output results. Extensions to accomodate new loading cases are quite simple. Modification of the structural model necessitates a completely new solu-

tion.

The data is stored in a sequential manner that reflects the simple access mechanism required in the processing modules. A few sequential disk files suffice for most programs. Random files are occasionally adopted to facilitate stiffness assembly and triangulation for large structures processed with out-of-core techniques.

In contrast, the automatic solution of a general substructured model requires a dynamic control capability. Each model solved with a general system may potentially require a unique order of module execution. The order of execution cannot be "pre-programmed" in the software as it is for standard analysis. Rather, the software logic must use a description of the substructure hierarchy to determine the flow of execution for each particular analysis (or reanalysis). Dynamic control logic easily accomodates modifications to substructures that alter the flow of execution during reanalysis. More sophisticated data structures and access schemes are needed to support the dynamic nature of the solution procedure. The requirement for equal access to the data for any substructure eliminates consideration of sequential file storage (unless literally hundreds of files are available). The topologic dependencies between substructures suggest the use of a hierarchical data organization that parallels the natural substructure hierarchy. Formal data base management techniques are necessary to define, maintain, and access the data structures. With this approach it becomes feasible to efficiently maintain data for any substructured model, regardless of size and complexity, within a single disk file.

For purposes of discussion, the substructured solution is considered to have two computational phases. These are: 1) processing the model through the computation of displacements for the highest level structure, and 2) recovery of substructure displacements and element strains-stresses. The separation into these two phases follows from the data structures and processing logic naturally suited for each task. Output generation is also important but does not impact the computational processes or data organization. The first phase involves stiffness and load vector assembly, and requires that substructures be processed upward from the bottom of the hierarchy. Data structures must support the repeated use of a substructure stiffness matrix and loads to form similar matrices for higher level structures. In the second computational phase, processing of the hierarchy occurs from the top-down and then only along user designated paths. Displacements, strains, and stresses are not normally recovered for all substructures. Data structures must reflect the uniqueness of displacements, strains, and stresses for each occurence of a substructure in the hierarchy.

Stiffness matrix and load vector assembly proceed upward from the lowest levels of the hierarchy. Assembly of a structure at level "I" cannot begin until all dependent substructures at level "I-1" have been completed. Control of the assembly process through a stack driven traversal of the hierarchy appears, at first, to be a natural approach. However, a topologic sort of the hierarchy (performed using a stack) to determine the processing order before computations begin is far simpler. The topologic sort provides the stiffness module with a properly ordered list of substructures to process. Figure 2.7 presents a block flow

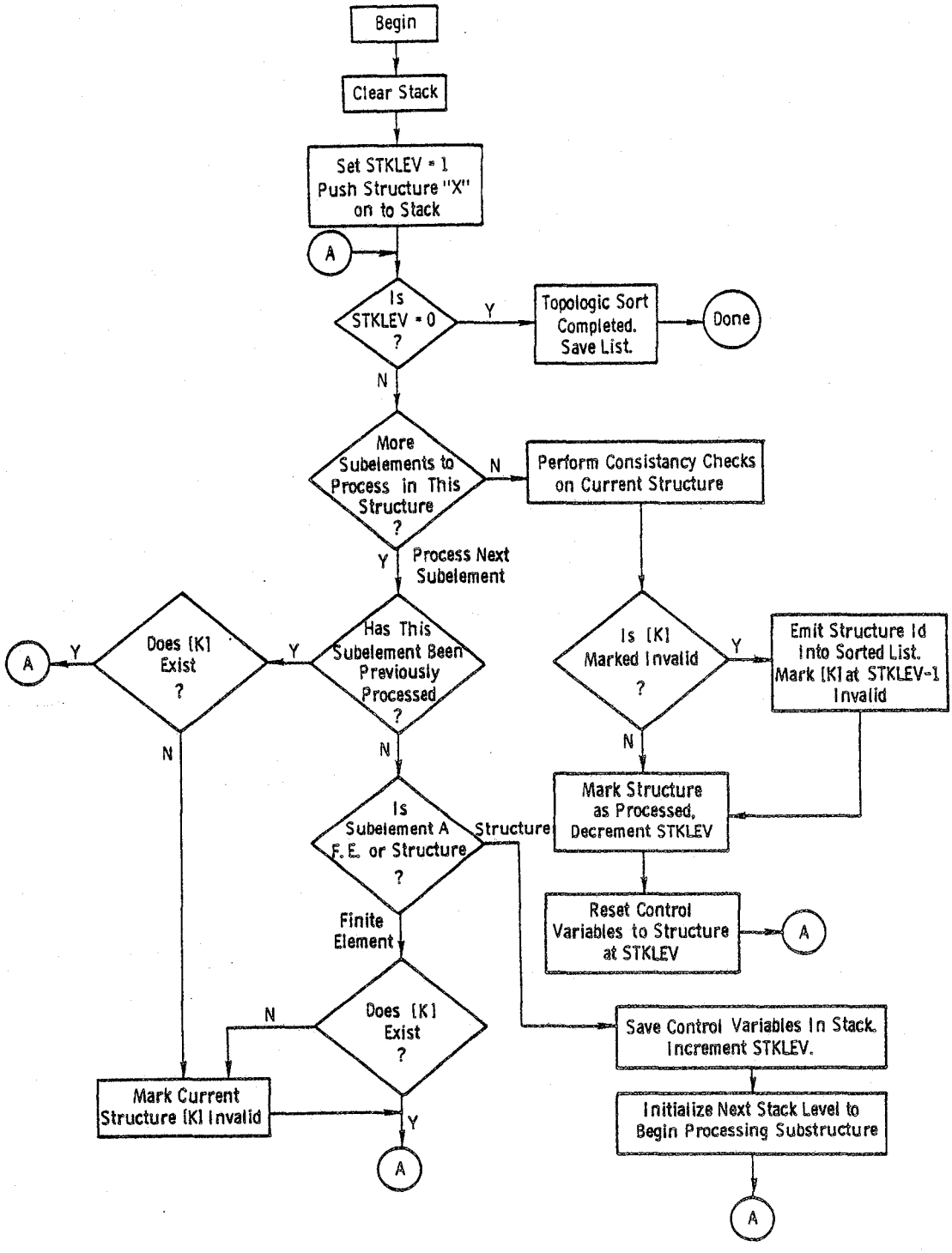


Figure 2.7 -- Topologic Sort to Order Substructures for Assembly

diagram for the topologic sort. A substructure appears only once in the sorted list regardless of its number of occurrences in the hierarchy. During reanalysis, the topologic sort simply omits all unmodified substructures from the list. Logical control within the assembly module becomes quite simple given the sorted list of substructures to process. Figure 2.8 illustrates the overall assembly logic, with the topologic sort shown as the first operation. The first structure in the list is extracted and its stiffness assembled, followed by the second structure in the list, etc. Only structures appear in the sorted list. The assembly module generates matrices for finite elements which occur in the structure being assembled.

When a "condensed" substructure is extracted from the sorted list, the assembler will have already computed the "uncondensed" stiffness matrix since the single child of the condensed substructure occurs one level lower in the hierarchy. The assembler suspends execution and invokes the equation solving module, passing it the identifier for the substructure to be condensed as indicated in Fig. 2.8. After the stiffness condensation is completed, the assembler resumes execution, retrieves the master node matrices, and saves them in data structures for the condensed structure. No traditional assembly operations are performed. Thereafter, the condensed substructure has a stiffness matrix with the same format as any other structure (or finite element). The assembler continues execution by selecting the next structure in the sorted list. Because the processing modules are dynamically invoked, condensation at any number of levels creates no logical control difficulties with this procedure.

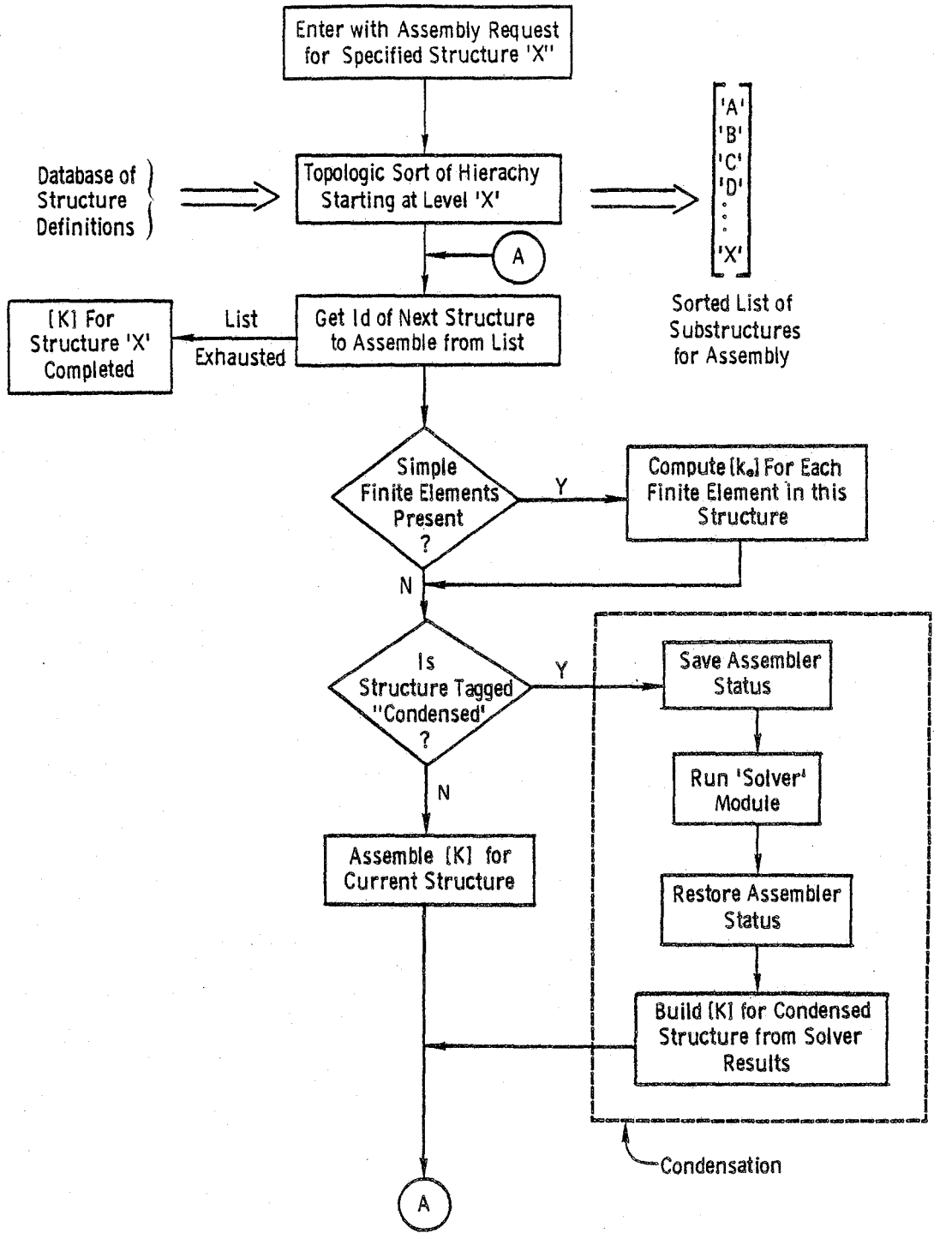


Figure 2.8 -- Multilevel Substructure Assembly Procedure

Figure 2.9 illustrates the essential features of a data structure for storage of the hierarchy description and stiffness matrices. The ELEMENTS table contains one column for every finite element and structure declared by the user. Rows of this table describe attributes for the element or structure, for example, the number of nodes "NUM_NODES". A STRUCTURE table exists only for columns that contain a structure. The STRUCTURE table stores pointers to lower level tables that contain data applicable only to a structure, for example, subelement incidences, subelement orientations, nodal coordinates, and components. The COMPONENTS table has one entry for each subelement in the structure. The data value for each subelement is the column number, denoted ECOL on the figure, in the ELEMENTS table that defines the subelement. The subelement may be a structure or a finite element. Given the ECOL for a structure, the hierarchy from that level downward may be easily traversed.

The stiffness matrix for a finite element and for a structure have identical storage formats as shown in Fig. 2.9. A stiffness matrix consists of nodal "submatrices" which are stored in a row-wise format. The column of the first non-zero submatrix in a nodal row is indicated by the value of FIRST_COL. For finite elements, this value is always one. For structures, the value of FIRST_COL reflects the half bandwidth for the nodal row. The lower triangle of the element stiffness is stored. All submatrices from column FIRST_COL through the diagonal are generated even though there may be intermediate null submatrices.

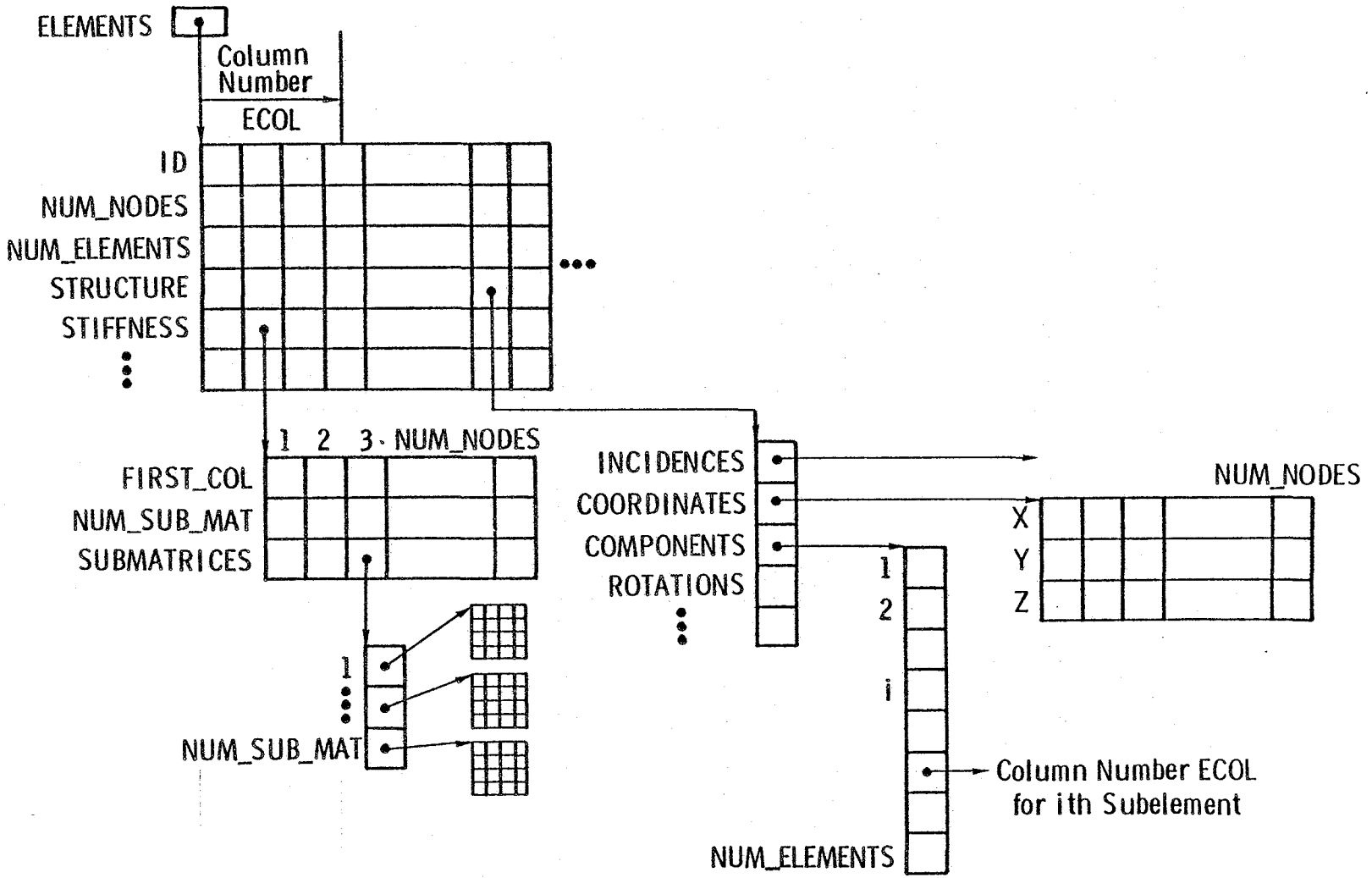


Figure 2.9 -- Data Structures for Hierarchy Representation of Stiffness Matrices

Assembly of the equivalent nodal loads for a multilevel substructure model is logically more complex than stiffness assembly. For example, when displacements for a load case defined on the highest level structure are requested, the equivalent loads processor must traverse the complete hierarchy while determining which substructures have load cases that contribute to the one specified on the highest level structure. The procedure is even more interesting when analysts define load cases in terms of other load cases from one substructure level to the next. A more flexible data structure than a simple vector of structure identifiers is necessary to accomodate the load case identifiers. A multiple vector, inverted list proves adequate. Condensation of the load vectors is performed in the same manner as stiffness condensation--as an interrupt in the normal procedure of transferring loads from one substructure level to another.

Phase two computations recover substructure slave displacements and element strains-stresses. The analyst generally specifies a path through the hierarchy that identifies the occurence of a substructure for which results are desired. The particular substructure may be at the lowest level of the hierarchy, in which case relatively few results are generated. Alternatively, it may be an intermediate level substructure, which can be used to imply displacement recovery for that level structure and all lower level substructures of the same branch.

Logical control using a stack technique is most convenient to drive substructure displacement recovery. Given the analyst supplied path and the COMPONENTS table shown in Fig. 2.9, the traversal procedure to reach the desired substructure(s) is quite simple. The back-condensation com-

putations to retrieve slave displacements from the master values are logically treated as an interrupt in processing down the hierarchy. In a manner paralleling the assembler condensation procedures, the displacement recovery module suspends execution, then initiates the equation solving module to compute slave DOF displacements for the current substructure. On resuming execution, the displacement recovery module extracts the slave DOF displacements in the solver data structures and reformats them to conform with standard displacement data structures. The next lower level in the substructure list may then be processed. Element strain-stress computations may be performed immediately after slave DOF displacements are recovered or a separate module may be invoked following the completion of all displacement recovery. A stack driven procedure also facilitates the computation of substructure strains-stresses.

The most interesting aspect of phase two involves the development of data structures for storage of substructure displacements, strains and stresses. Only one stiffness matrix exists in the database for a substructure regardless of its number of occurrences, but the displacements, strains, and stresses are unique for each occurrence. The data structure shown in Fig. 2.9 for phase one processing requires extension to support substructure displacement recovery. One solution for this problem is illustrated in Fig. 2.10. The uniqueness of substructure results is recognized only at the highest level structure. The LOADS table points to lower level tables that contain the definition and computed results for each load case. Displacements, strains, and stresses for the complete hierarchy for a load case are stored under the cor-

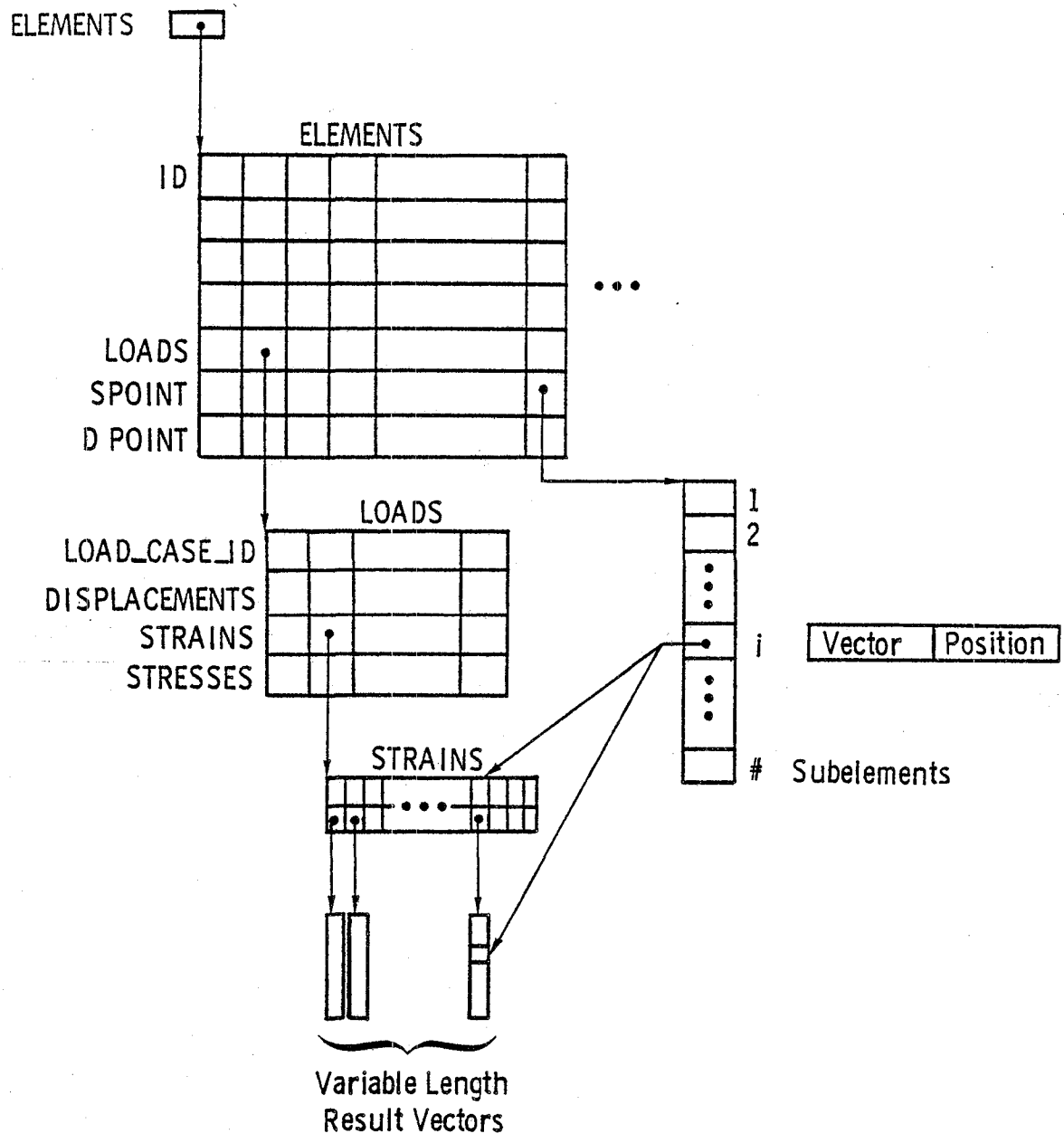


Figure 2.10 -- Data Structures for Storage of Substructure Results

responding column of the LOADS table (defined for the highest level structure). Results are packed into sets of variable length vectors that reside under the DISPLACEMENTS, STRAINS, and STRESSES pointer tables. The allocation and retrieval of result vectors is accomplished by a two-level pointer scheme. For example, vector SPOINT shown in the figure contains one entry for each subelement in the structure. For subelements that are finite elements, the "I" th entry in SPOINT defines the vector and position within the set of result vectors at which values for the element begin. For subelements that are substructures, the SPOINT entry defines a relative shift in the vector numbers from the current level structure's vector to the first relative vector for the specific occurrence of a substructure. SPOINT entries for the substructure refer to vector numbers that are relative to the first result vector for the substructure. Results for multiple occurrences of the same substructure thus appear in different stress-strain vectors. The absolute vector number for processing at any time is obtained by accumulating relative shifts for each substructure present in the traversal stack.

The most complicated aspect of this scheme involves construction of the SPOINT vectors for each substructure. Note that the SPOINT vectors are independent of any load cases. The mapping of substructure results onto vectors is identical for each load case. An identical scheme is used to store substructure nodal displacements, reactions, initial strains-stresses, and residual loads in nonlinear analysis.

2.5.3 Linear Equation Solving

Efficient procedures for linear equation solving are essential in finite element analysis. Reduction of the symmetric stiffness matrix to triangular form often requires 50 percent of the computational effort for linear analysis and a slightly smaller percentage for nonlinear analyses. Many efficient algorithms and the details of their computer implementation have been published. Meyer [2.12] has presented an extensive review of the subject.

The growing availability of computer hardware designed specifically for "number crunching" applications has spurred renewed interest in equation solving procedures. The details of data storage and access mechanisms comprise the key factors in the ability to utilize advanced hardware. Advanced machine architectures include parallel processors (the basis for most supercomputers) and virtual memory superminis. Some of the most recent supermini computers also have a pipeline design that greatly speeds up numerical operations compared to more conventional scalar processing. The pipeline concept uses a single processor but with separate prefetching of data to minimize processor wait times. It thus represents a intermediate design between a scalar processor and the parallel processing machine. The scalar design superminis can also be enhanced with the addition of an attached array processor.

The basic requirements for any equation solving procedure can be summarized as:

1. The number of equations should be limited only by the amount of disk storage available;
2. Use a reasonable amount of memory so as not to severely impact scheduling of the computer runs;
3. Minimize the storage of zero coefficients and operations on them;
4. Minimize data transfers to and from disk (I/O);
5. Exploit capabilities of modern computer architecture.

Substructure condensation by partial decomposition is essentially an equation solving procedure and therefore the same requirements listed above apply. However, stiffness condensation may adversely effect one decomposition procedure more than another. With band-based solvers, for example, reordering of the equations to facilitate condensation may produce a border-banded matrix of coefficients. The computation time required for partial decomposition is greatly increased compared to the time required for complete decomposition of the equations without reordering.

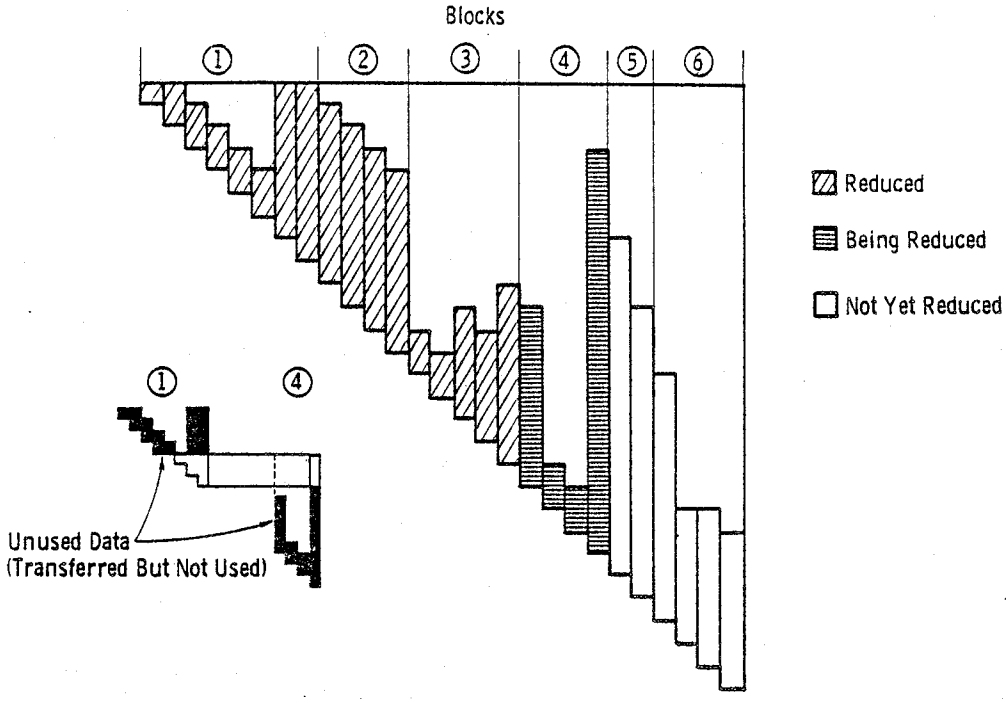
This section briefly surveys three standard equation solving procedures, namely (1) Choleski with variable band (skyline) storage, (2) the front method of solution, and (3) Choleski with hypermatrix storage. Each of these is a direct solution algorithm that involves a three step process: (1) reduction of the coefficient matrix to triangular form, (2) forward reduction of the load vector(s), and (3) recovery of displacements by a backward pass. The primary interests in the current discussion include the impact of reordering the equations prior to condensation and the prospects for adopting each procedure to new computer hardware.

Variable Bandwidth Procedure

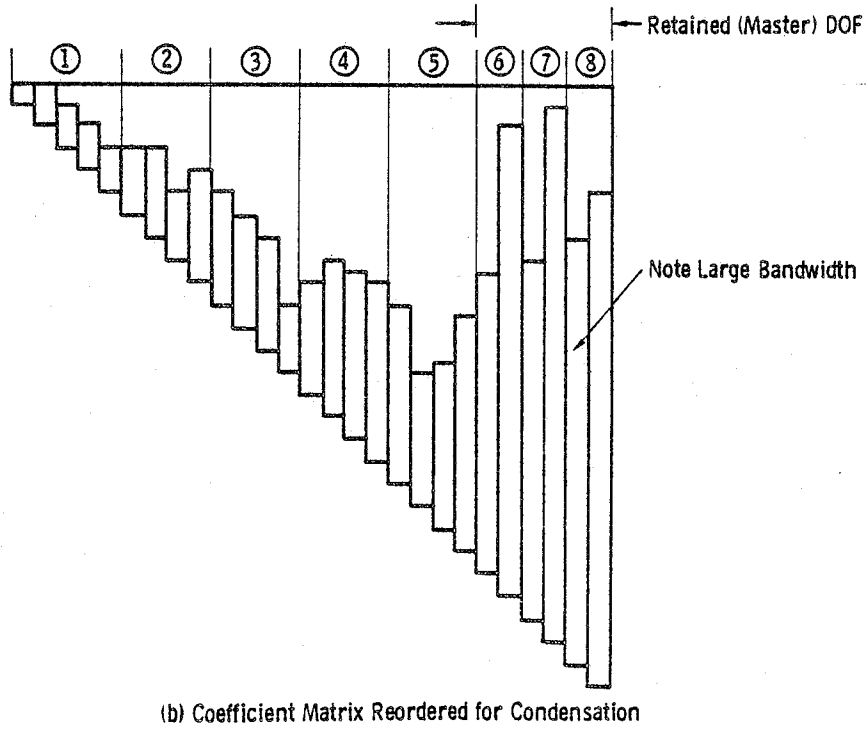
This tecnnique uses a Choleski or Crout triangulation algorithm with a compact storage scheme to accomodate wide variations in bandwidth. Jennings and Tuff [2.8] originally developed the procedure. Mondkar and Powell [2.13] later published the implementation details and results of timing studies. This procedure appears to have been widely adopted for "in-house" finite element systems because the data handling details are straightforward.

Contiguous columns of the upper triangle are grouped together in blocks (Fig. 2.11). All coefficients in a block are transferred to and from disk in a single, logical operation. The available memory determines the number of columns assigned to each block. The first non-zero row in each column is recorded during assembly. This information is simply appended after the last column of each block. All terms from the first non-zero row to the diagonal in a column are assumed to fill-in during triangulation.

Reduction of each term in column "i" requires that column "i" and at least one other column, "j", where $j < i$, reside in memory. Thus, at least two blocks of columns must fit simultaneously into the available memory. During forward load reduction, one coefficient block and one load block are required in memory (multiple load cases are handled very efficiently). The same memory requirements exist for the backward pass to recover displacements.



(a) Typical Variable Band Blocked Coefficient Matrix



(b) Coefficient Matrix Reordered for Condensation

Figure 2.11 -- Storage Scheme for Blocked, Variable Bandwidth Procedure

Over 80 percent of the effort in triangulating the matrix involves computing inner products of two non-contiguous columns. This makes the algorithm very attractive for all types of advanced hardware. The number of terms in the inner products is usually large relative to the break even point on most hardware (some overhead is involved in parallel processing to initially align data which requires that inner products have a minimum length to recover the overhead time--generally 5 to 10 terms).

The variable band procedure has a number of drawbacks. It can be very I/O inefficient during structure stiffness assembly if only one or two blocks fit into memory. Elements are usually processed in sequential order to assemble the structure stiffness. As a consequence, it is possible for a structure to have a very narrow bandwidth and yet require considerable swapping of blocks during assembly. The classic example is a narrow rectangular grid with nodes numbered in the short direction to minimize the bandwidth and elements numbered in the long direction (which does not affect the bandwidth). To assemble each sequence of elements in the long direction, each coefficient block must be brought into memory. Another drawback occurs when column heights vary significantly. Large numbers of inactive terms are transferred into memory during triangulation as a consequence of the blocking (Fig. 2.11a). Lastly, the storage scheme is biased towards accessing coefficients column-wise, which is natural for triangulation and forward load reduction. As noted above, each block is transferred to memory only once during the forward pass. During the backward pass, however, row-wise access is the most natural. A block of columns may be transferred into

memory many times depending on bandwidth variations.

When executed on virtual memory hardware, the logical memory space (that declared in FORTRAN DIMENSION statements) for blocks may be very large, for example, sufficient to store seven or eight blocks. The program issues fewer logical I/O requests to transfer blocks than a conventional approach in which space for only two blocks is allocated. The virtual memory operating system performs additional I/O required forcing the program to execute in a given amount of real memory (working set size). This process is normally handled through fragmentation of data arrays into pages which are transferred to and from memory as the array elements are referenced in the program. Considerable experimentation is necessary to balance the virtual paging and program block I/O, especially on virtual machines that permit users to explicitly declare the working set size.

Condensation causes difficulty only when the bandwidth increase due to reordering becomes excessive. Consider a structure which is numbered to minimize the average bandwidth for conventional analysis. In general, the reordering to accomodate condensation produces the border-banded matrix shown in Fig. 2.11b. This procedure requires that at least one complete column fit into a block. For large 3-D structures, this requirement may lead to large increases in block sizes to accomodate the very long columns corresponding to master degrees of freedom. During the partial decomposition process, no computational penalty occurs until the equations with the large band in the last blocks are reached. Most of the triangulation time may be involved in eliminating these terms. However, advanced hardware can be exploited to

the fullest in processing the very long columns.

Frontal Procedure

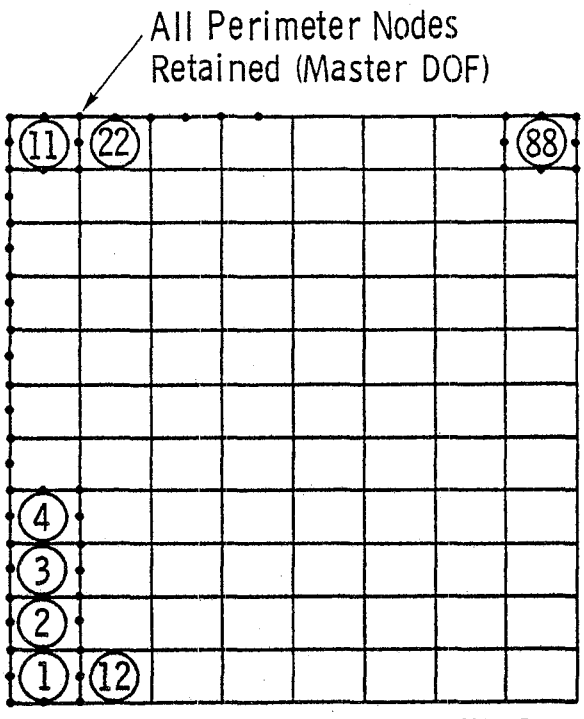
The highly touted frontal procedure was first introduced by Irons [2.6] and has since been extended to include condensation. The frontal solver is essentially Gauss elimination with extensive bookkeeping to minimize operations on zeroes. The structure stiffness matrix is never explicitly assembled. Rather, element stiffnesses are brought sequentially into memory, their terms added to the system matrix and then triangulated, all in one logical (but very complex) process. The ordering of equations in the system matrix is determined by the element, rather than the node, numbering scheme. As elements are processed, newly appearing DOF are simply appended as new equations. The memory occupied by active equation coefficents is referred to as the "front." The front storage space varies dynamically during solution as new equations are added and old ones are completely eliminated (eliminated coefficients are transferred to disk). The front is very similar to the "active triangle" concept in band-based solvers.

The frontal procedure eliminates much of the CPU and I/O costs associated with structure stiffness assembly. The triangulation aspect is no more efficient than the Choleski process. There are two major disadvantages of the frontal procedure for non-substructured analyses. First, the active front size becomes quite large for 3-D solid analyses. The bookkeeping logic is very complex when the complete front fits into memory; it appears intractable when coupled with a "spill" algorithm to accomodate a large front that is partially memory resident. Storage

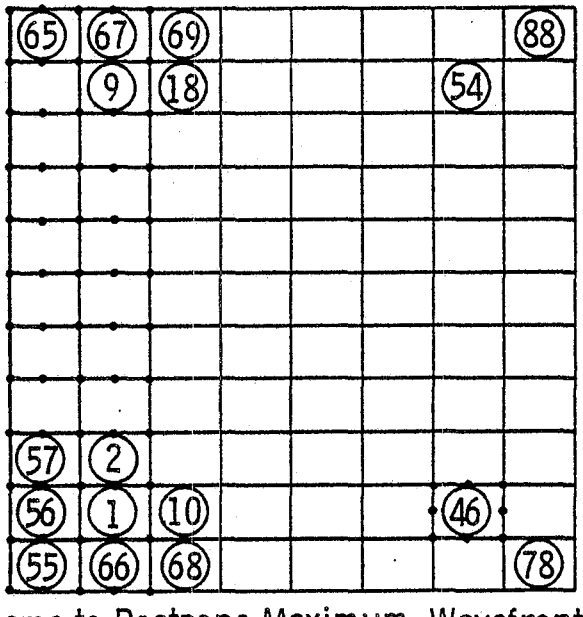
space of 100-200K words for an active front is common. This may cause scheduling problems on some machines and impose absolute structure size limitations on others. The second difficulty with the frontal method is its adaptation to advanced hardware. Unlike the Choleski procedure, inner products involving hundreds of terms are not common in the frontal solver. Operations are performed almost randomly over the active front. Parallel and pipeline hardware offer little advantage.

The frontal method appears ideally suited for virtual memory computers with scalar processors. The problem of handling a very large active front is relegated to the virtual memory operating system of the computer. An extremely large array space is dimensioned for the front. The operating system pages the array segments as needed to maintain a predefined working set. The random accessing into the front causes no penalty on a scalar (sequential) processor.

The absence of an explicitly assembled structure stiffness is a drawback of the frontal solver for substructured analyses. Substructure stiffness matrices (original and reduced forms) must always be available for use in defining other structural hierarchies. Stiffness condensation also affects the efficiency of the frontal solver as it does the variable band solver. Consider the element grid shown in Fig. 2.12a. All nodes on the perimeter are retained after condensation. Conventional node numbering for a band solver produces a border-banded matrix. Condensation in the frontal solver is accomplished by retaining all master nodes in the front until slave nodes are completely eliminated. For the mesh shown, the active front size increases each time a top and bottom row element is processed. The front size reaches a maximum when



(a) Worst Numbering for Condensation with Front Procedure -- Best Numbering for No Condensation



(b) Number Scheme to Postpone Maximum Wavefront Until Last Elements Processed

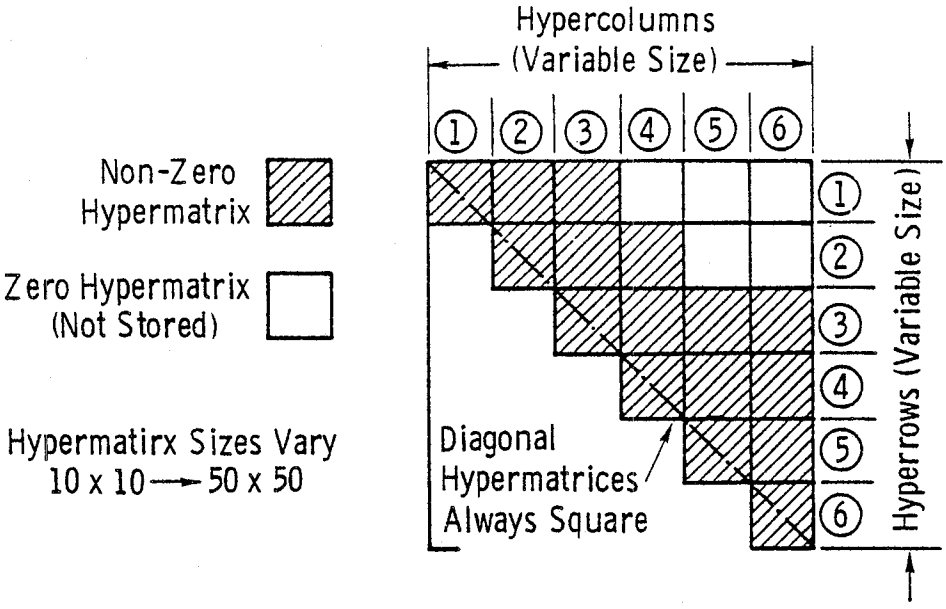
Figure 2.12 -- Effects of Element Numbering on Condensation by Frontal Technique

element 88 is processed. Alternatively, elements can be numbered as shown in Fig. 2.12b for front solution. The front size remains constant while all slave DOF are completely eliminated. It then grows rapidly to the same maximum size for the previous numbering scheme as elements coupling the master and slave DOF are processed. The only advantage of the second case is that a maximum front size exists for less execution time. This could drastically reduce the paging rate on virtual memory machines. The same effect occurs for the variable band solver in that the active triangle reaches a maximum size when the master DOF are encountered.

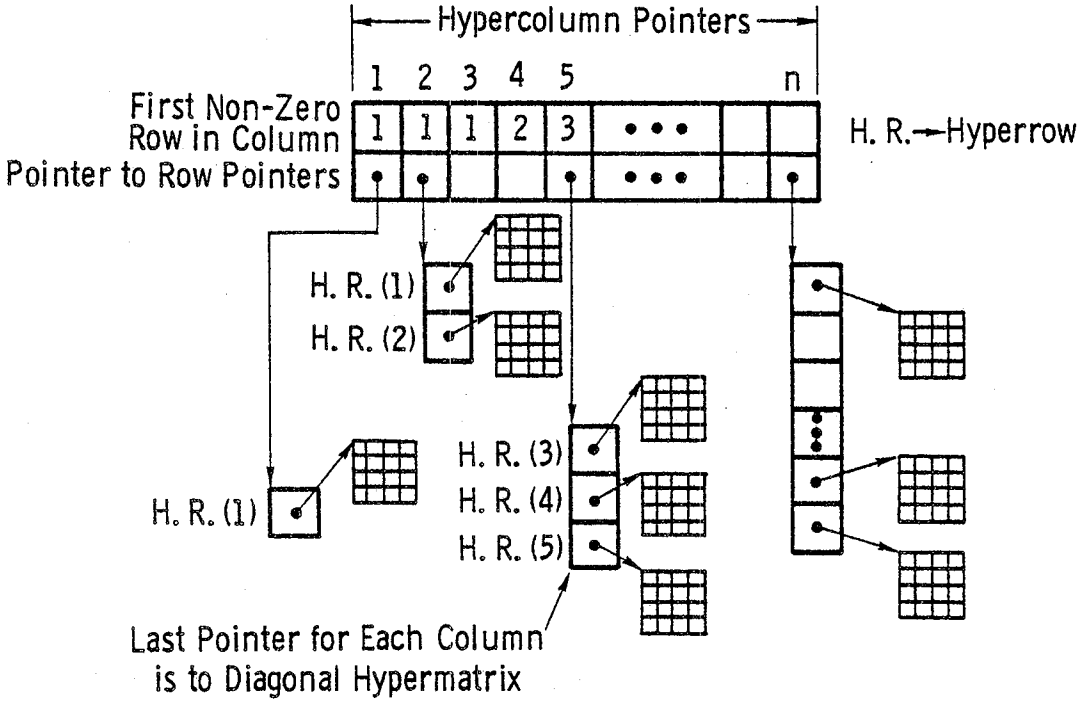
Hypermatrix Procedure

The basic concept of hypermatrix storage is illustrated in Fig. 2.13a. A block of contiguous columns defined in the variable band procedure is further partitioned row-wise to form rectangular "hypermatrices". This storage format overcomes the major problems with variable band storage--excessive column heights and the transfer of unused terms during triangulation.

Diagonal hypermatrices are always square; off-diagonal matrices are frequently rectangular. Hypermatrices with all zero terms are never created. It is thus a simple matter to omit large numbers of zero operations with such a storage format. The sizes of hypermatrices can easily be adjusted to fit a particular amount of memory available for processing. The use of larger blocks (i.e., greater than 50 x 50) increases the number of zero terms picked up at the periphery of the band. The use of smaller blocks improves the recognition of zero terms but in-



(a) Symmetric Partitioning of a Sparse Matrix into Hypermatrices



(b) Data Structure for Hypermatrix Storage

Figure 2.13 -- Hypermatrix Storage of Sparse Matrices

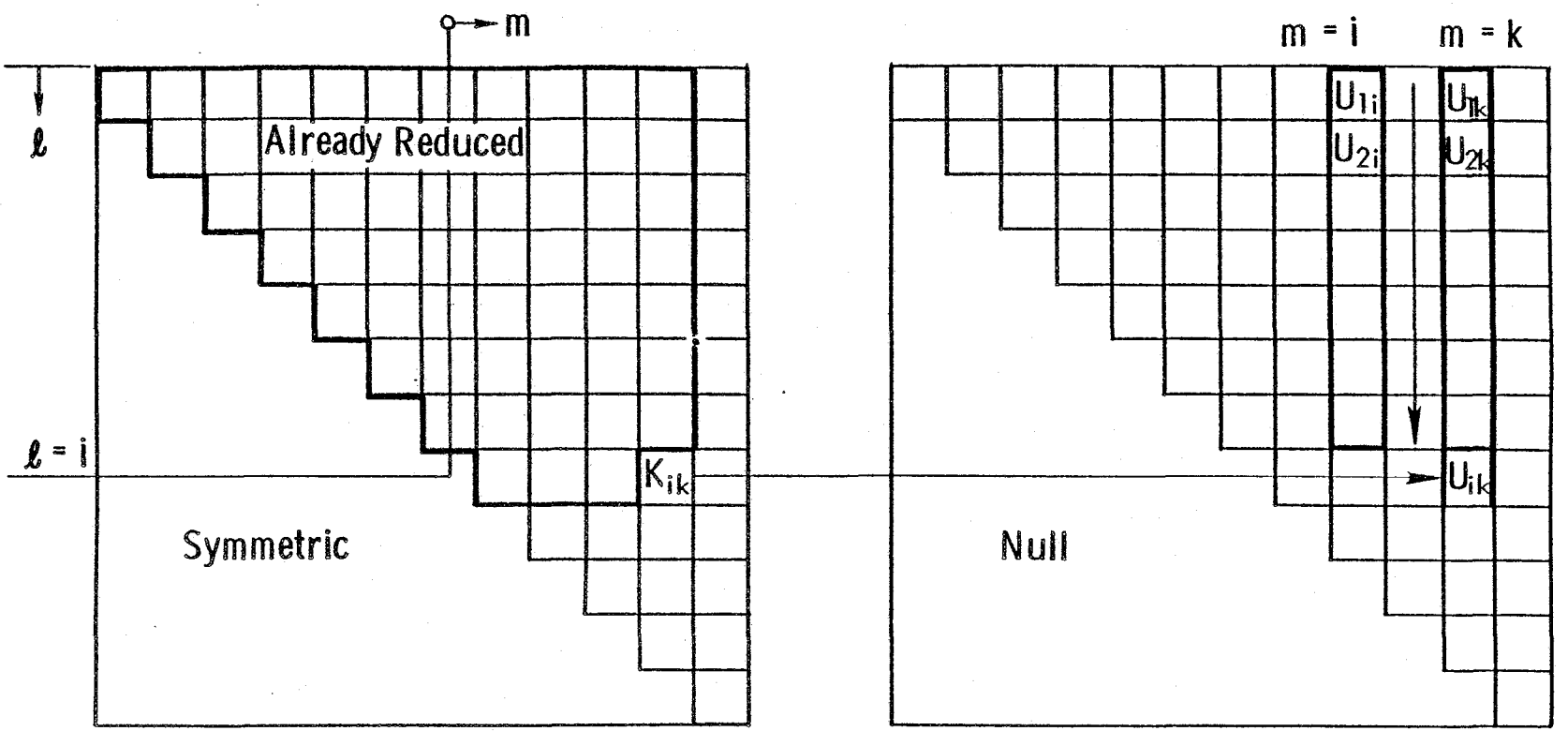
creases the data handling overhead.

A formal data structure for hypermatrix storage is shown in Fig. 2.13b. Individual hypermatrices are stored as separate entities (for example, a logical disk file record). Pointer vectors that locate all hypermatrices within a column eminate from the header table shown. Pointers to zero hypermatrices are not maintained in the vectors. Instead, the header table contains an offset locating the first non-zero hypermatrix in each column.

Hypermatrix partitioning has been studied extensively by computer scientists [2.10] in connection with array operations on virtual memory (paging) hardware. It has been demonstrated that hypermatrix partitioning requires the least working storage and incurs the least page faults for the operations of multiplication and triangulation. The usual scalar formulations for matrix multiplication and triangulation (Gauss, Cholesky, Crout) have very similar counterparts when cast in hypermatrix form. Cholesky decomposition in hypermatrix form requires the same number of floating point operations as in scalar form. Both multiplication and triangulation require that only three hypermatrices reside simultaneously in memory. Therefore, large problems may be solved in a very small memory space, for example, 7500 words if 50 x 50 blocks are used. Moreover, if memory for only three blocks is available it becomes a trivial problem to predict exactly the number of block transfers to and from disk. When space is available in memory for more than three blocks, it has been suggested [2.10] that a "least recently used" replacement algorithm efficiently utilizes the additional memory.

Fuchs and Schrem [2.4] devised a Choleski triangulation scheme in hypermatrix form and implemented the procedure in the ASKA program [2.16]. The data structure in ASKA employs a two level matrix of pointers to the hypermatrices instead of the scheme shown in Fig. 2.13b. The POLO-FINITE system uses the hypermatrix procedure with the storage format shown in Fig. 2.13b. Both ASKA and POLO-FINITE use demand paging, virtual memory management systems for which hypermatrices are ideally suited. Hypermatrix sizes are usually defined such that one hypermatrix fits onto a page. A page corresponds to a single random disk file record. The advantage of allocating one hypermatrix per page is that totally unbiased access to any block is obtained. Triangulation, forward pass, and backward pass operations thus have equal efficiency with respect to data access. As previously noted, the variable band scheme suffers a heavy access penalty during the backward pass.

As with the variable band procedure, the major effort in hypermatrix triangulation involves inner products of non-contiguous columns. Figure 2.14 illustrates this. However, in hypermatrix form, the inner product is actually a sequence of matrix multiplications as indicated on the figure. In effect, inner products over multiple scalar columns proceed simultaneously. Timing studies for large sets of equations have shown that the matrix multiplications usually represent 90% of the triangulation effort. The data handling overhead seldom exceeds 5% of the triangulation time. For example, after two 50 x 50 blocks are brought into memory, approximately 250,000 floating point multiplications and additions (plus subscripting) are performed to complete the matrix multiplication.



$$\text{Typical Operation: } [U_{ik}] = [K_{ik}] - \sum_{\ell=1}^{i-1} [U_{\ell i}]^T [U_{\ell k}]$$

Figure 2.14 -- Schematic of Hypermatrix Choleski Triangulation

Hypermatrix triangulation appears well suited for adaptation to parallel and pipeline hardware. The order of matrices in the multiply operation shown in Fig. 2.14 is particularly important. Because the first term is transposed, the matrix multiply is simply inner products of columns in the two matrices, rather than the traditional row multiplied into a column. Storage of the lower triangle in hypermatrix form eliminates this advantage of column-wise inner products. The equivalent matrix product requires inner products of two rows, rather than two columns. The choice of lower or upper triangle storage is immaterial for computations performed on scalar hardware. However, upper triangular storage is preferred for the most general case.

The large bandwidth increase that occurs with equation reordering prior to condensation presents no difficulties in the hypermatrix scheme. Large bandwidth fluctuations simply increase or decrease the number of non-zero hypermatrices in a column. The problem of fitting an entire column into memory that occurs with the variable band method does not occur with hypermatrix storage.

2.6 Examples of Substructured Analyses

Two example analyses are described in this section to illustrate typical computational savings achieved with substructured models. In the first example, a linear analysis is performed for a portion of a jet engine exhaust duct. Condensation is applied at two levels in the substructure hierarchy -- at the lowest level to reduce the bandwidth and at the next higher level to utilize repeated components. The exhaust duct represents an excellent test case for substructured dynamic

analysis with modal synthesis (discussed in Chapter 3) considering only linear response. Dynamic analyses using a full, unsubstructured model and some limited experimental data are available in the literature for comparison.

The second example problem consists of a slightly curved thick shell constructed of impact resistant acrylic. A uniform pressure is applied over a very small area at the apex to simulate the impact of a nondeformable object. The magnitude of the pressure is increased in each load increment but the loading area remains constant. Nonlinear response due to yielding of the material in the impact region is considered. The problem typifies a large class of structures for which substructured models reduce the computation time. The region of nonlinear response is easily estimated prior to the analysis. Standard and substructured models for this example are analyzed to provide data for comparisons of computational effort. This problem also provides an excellent test case for substructured, nonlinear dynamic analysis. A transient analysis is required to predict dynamic response following the impact of a high velocity projectile. Even under such loading, the nonlinear zone remains small relative to the overall structural dimensions. Condensation of the linear region should greatly reduce the computational effort for a transient analysis. For comparison, the computational effort within each time step of a transient analysis corresponds to that for a load increment in static analysis. The results presented here for the static solution provide a basis to estimate computational savings for transient analysis.

Each structure has been analyzed with the POLO-FINITE system. Comparisons between substructured and standard model solutions are made on the basis of CPU time and I/O. The POLO supervisor performs database and memory management functions for FINITE subsystems. POLO has extensive instrumentation that provides detailed summaries of CPU usage among the FINITE subsystems, for example: input, assembly, and triangulation. Within each subsystem, the CPU time expended on data management activities and on actual finite element computations is also available. CPU times are presented in non-dimensional form to eliminate dependencies on the processor execution speed. I/O activity is measured by the number of "page faults" executed by the POLO memory management system. A page fault represents the transfer of one 2500 word record from memory to disk followed by the transfer of another 2500 word vector to memory. For a given structural model, the number of page faults performed by POLO is independent of the computer hardware; it depends only on the dimensioned length of a data vector within POLO. Page faults are thus a very simple measure of I/O activity. Printer output and I/O transfers to sequential card image files are ignored.

2.6.1 Linear Example

Figure 2.15 shows a portion of a jet engine exhaust duct modeled for static stress analysis. The duct consists of a thin circular shroud connected to the central core with radial fins. Loadings of interest include torsion, uniform external pressure, and nonuniform temperature distributions. For this analysis, the inner edge of each fin is completely fixed.

DIMENSIONS

Shroud	Fins
R = 170 mm	L = 64 mm
t = 1.5 mm	t = 3.0 mm
w = 81 mm	w = 27 mm

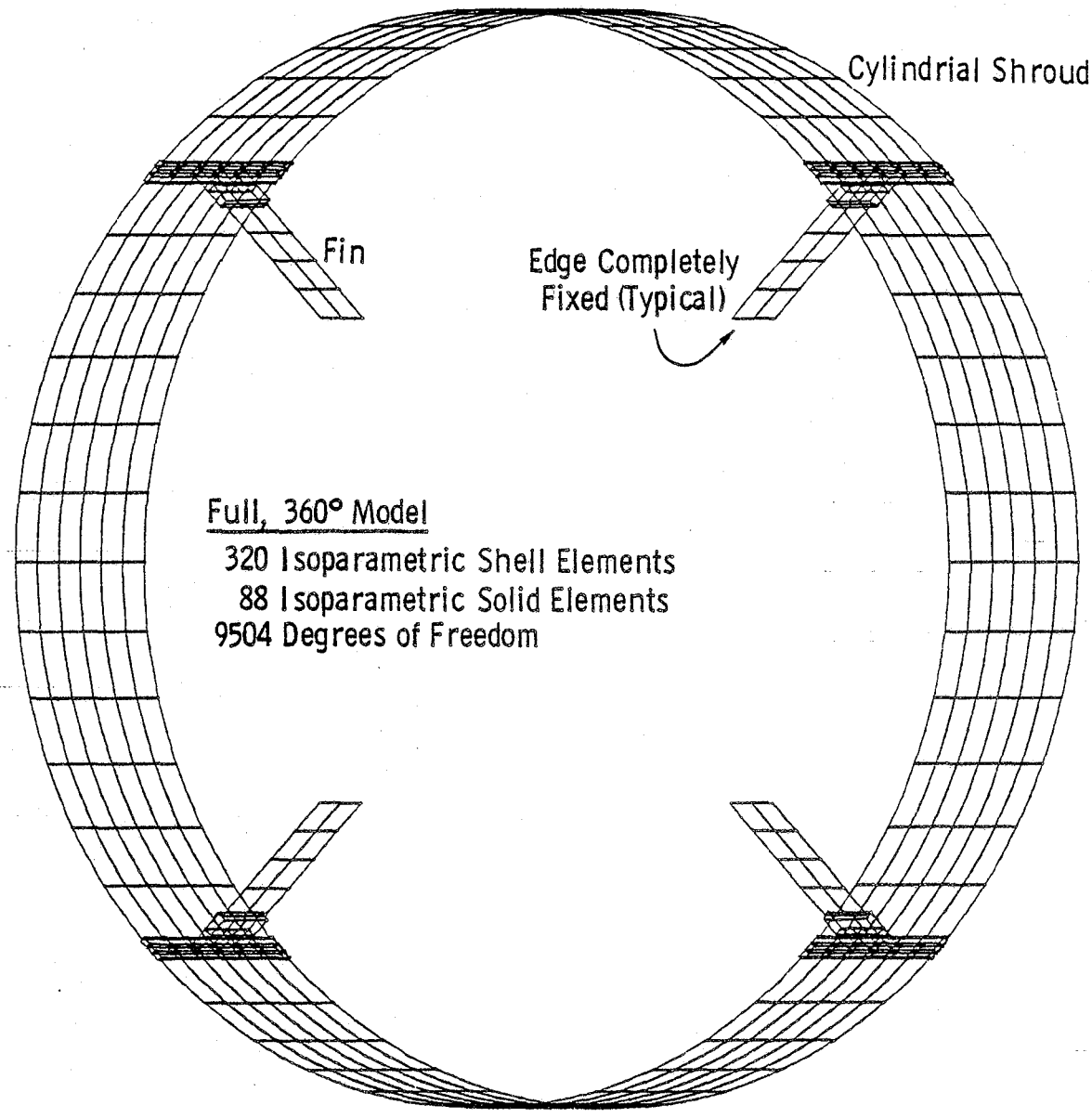
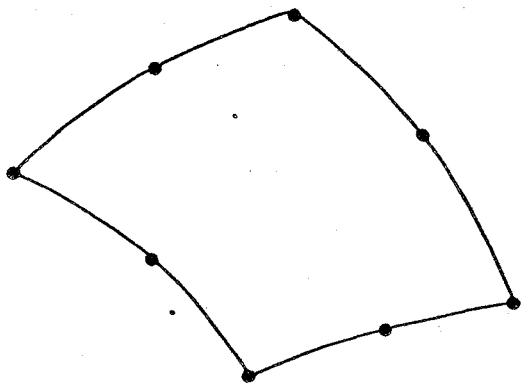


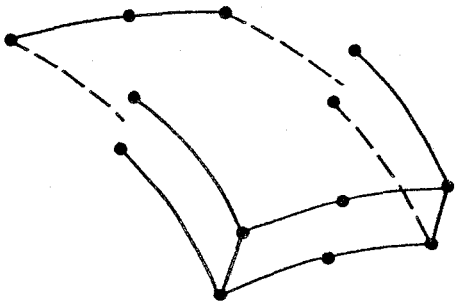
Figure 2.15 -- Jet Engine Exhaust Duct for Example Linear Analysis

Difficulties in constructing a proper finite element model arise at each junction of a fin with the shroud. Accurate determination of the bending stresses in the junction vicinity requires a relatively fine mesh and the maintenance of element displacement compatibility. Curved shell elements adequately model major portions of the fins and shroud. The 8 node isoparametric shell element is used here. Each junction is modeled as a 3-D solid with shell-to-solid, and solid-to-solid transition elements employed to maintain displacement compatibility. This model provides realistic predictions of stress distributions at each junction without an undue increase in the number of nodal DOF. Figure 2.16 illustrates the four types of elements employed in the analysis.

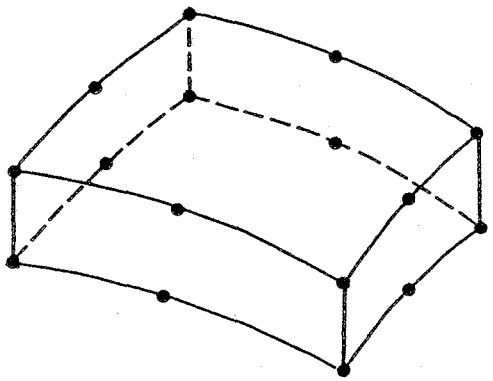
A 360 degree model is required for general analysis. For the grid shown in Fig. 2.15, the full model has 408 elements and approximately 9500 DOF. The generation of each element stiffness matrix requires numerical integration. And while the large number of DOF does not present major problems for general purpose systems, the grid topology does introduce some severe computational penalties. The traditional node numbering scheme follows the narrow (axial) direction then the circumferential direction. Each fin causes a large re-entrant area in the equation coefficients, but the most severe penalty arises because the first and last nodes are coupled. As a result the stiffness matrix is border-banded; the last 78 rows have a bandwidth of nearly 9500. This drastically increases the triangulation effort. Substructuring with condensation proves very computationally efficient for this structure as demonstrated below.



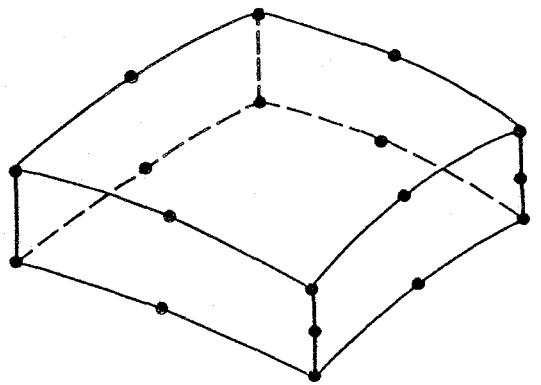
a) 8 Node Shell (QSHELL)



b) Shell-to-Solid Transition (SHELL-TRANS)



c) 16 Node Solid (TS16ISOP)



d) 18 Node Solid Transition (TS18ISOP)

Figure 2.16 -- Element Types for Exhaust Duct Model

Division of the full model into identical 90 degree substructures is obvious. The selected partitions place the fin at the center of the substructure as shown in Fig. 2.17. The location of each type of finite element is also indicated on this figure. Condensation of this substructure reduces the number of nodes to 26; 13 nodes across each end for connection with adjacent copies of the same substructure. Within the 90 degree substructure, the fin is also modeled as a substructure and condensed to the 13 nodes that connect to the shroud. Condensation of the fin eliminates the re-entrant area in the coefficients described above for the full model. Figure 2.18 shows the same 90 degree section model without substructuring.

Considering the analysis of an isolated 90 degree section, the fin condensation reduces CPU time by 14% and I/O transfers by 21% compared to an analysis without fin condensation. (Fin stiffness generation and condensation time are included in the comparison.) However, condensation to eliminate all but the 26 boundary nodes increases the CPU by 53% and the I/O transfers by 65%. These increases reflect directly the penalty incurred by reordering the equations prior to condensation. Reordering for this substructure has the same effect on the bandwidth as closing the ring on the full model. The first and last nodes are effectively coupled by the reordering, but final 156 rows have a bandwidth of only 2400 compared to 9500 for the final 78 rows of the 360 degree model. The large bandwidth difference produces the major savings for the full model generated with substructures. (Recall that the triangulation effort is proportional to bandwidth squared.)

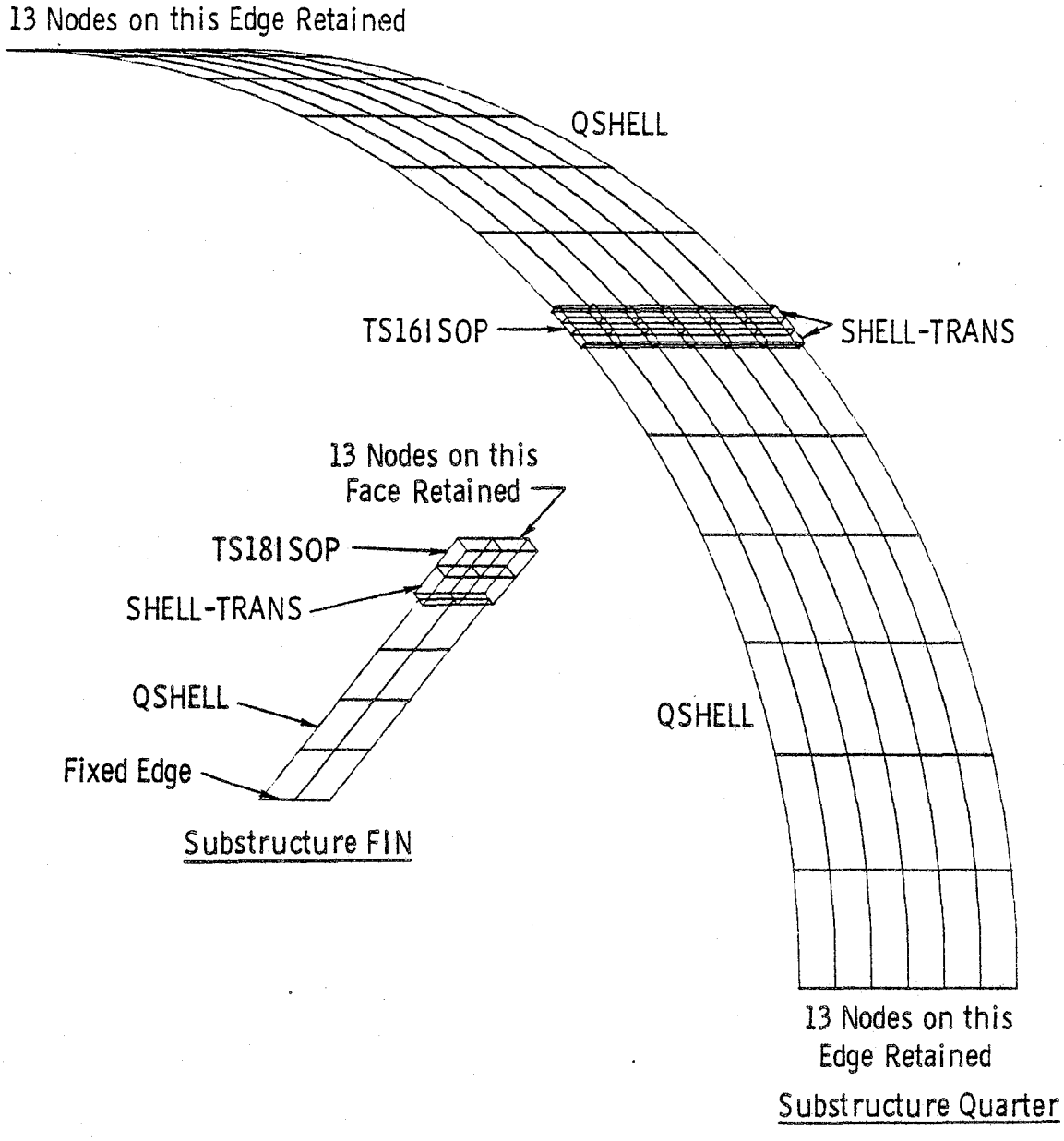


Figure 2.17 -- Substructured 90° Model for Exhaust Duct

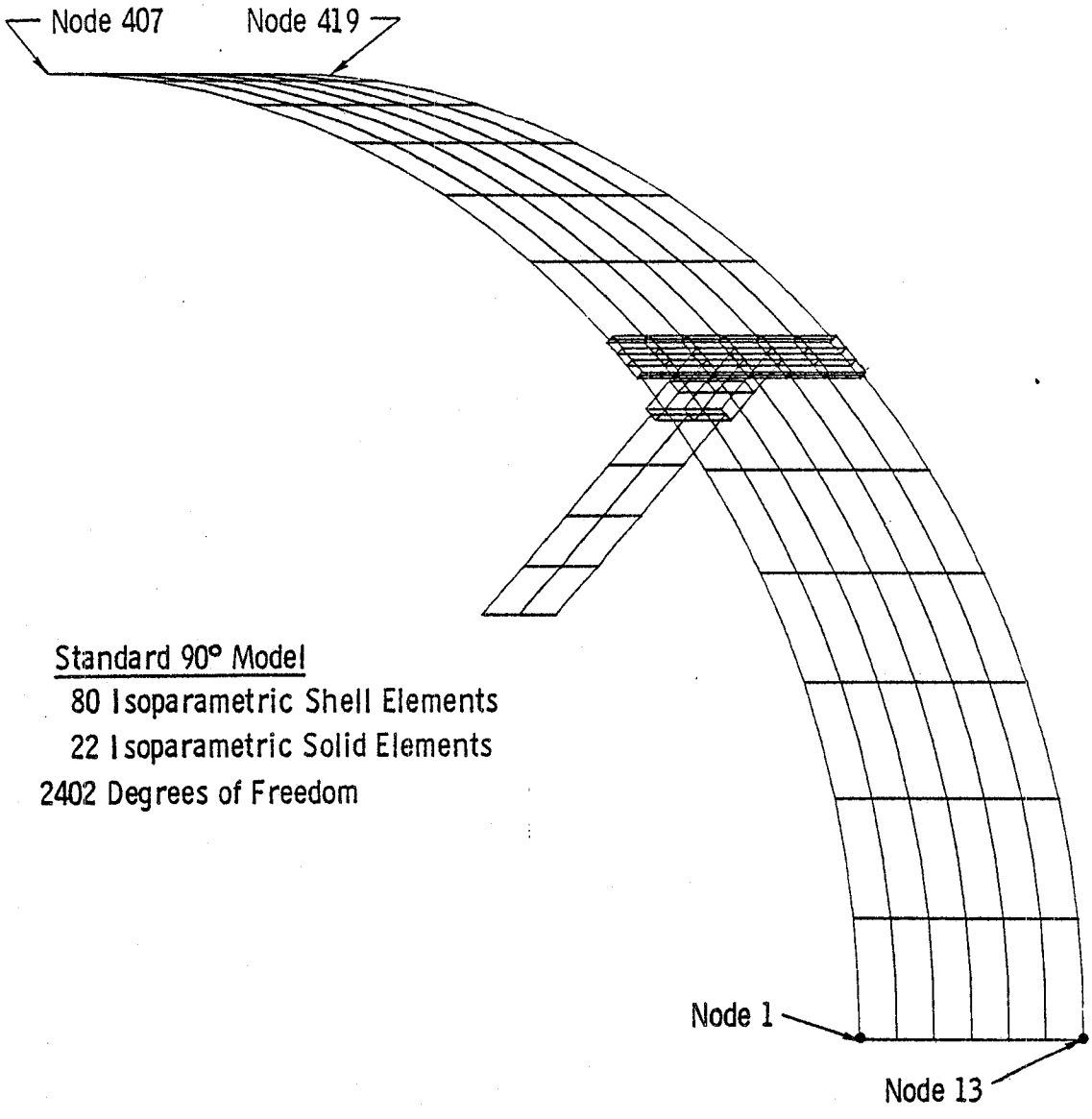


Figure 2.18 -- 90° Section of Exhaust Duct Without Substructuring

Using four properly oriented copies of the reduced 90 degree substructure, the 360 degree model was generated and solved. The final structure has only 312 nodal DOF with effectively a full bandwidth. (The condensed substructure stiffness is fully populated.) Figure 2.19 illustrates the substructure hierarchy for the 360 degree model. Five levels of substructures are present in the model.

Table 2.1 compares the CPU and I/O for the various solutions. The CPU and I/O required for complete solution of the substructured 360 degree model are assigned values of 1.0. Relative times are given for other models. Over 90% of the full model solution effort is expended in the substructure reduction process. This is not too surprising given the very large DOF reduction and the very small number of DOF at the highest level structure. It was not possible to analyze the unsubstructured 360 degree model with POLO-FINITE given the current limitation on the number of nodes in a structure (the system limits the number of nodes in a single substructure to 833). However, accurate estimates of the CPU and I/O time are possible using the exact timings obtained for solution of the isolated 90 degree structure without fin condensation. The estimated CPU and I/O consider the factor of 4 in element stiffness generation effort, the factor of 4 in number of DOF, and and the quadratic increase with bandwidth for the final 78 equations. The relative savings with the substructured model are in the 13-14 range for CPU and 18-20 for I/O transfers. Most of the saving results from the bandwidth reduction noted above, but a minimum factor of 4 is realized from the reduction in nodal DOF and element stiffness generation.

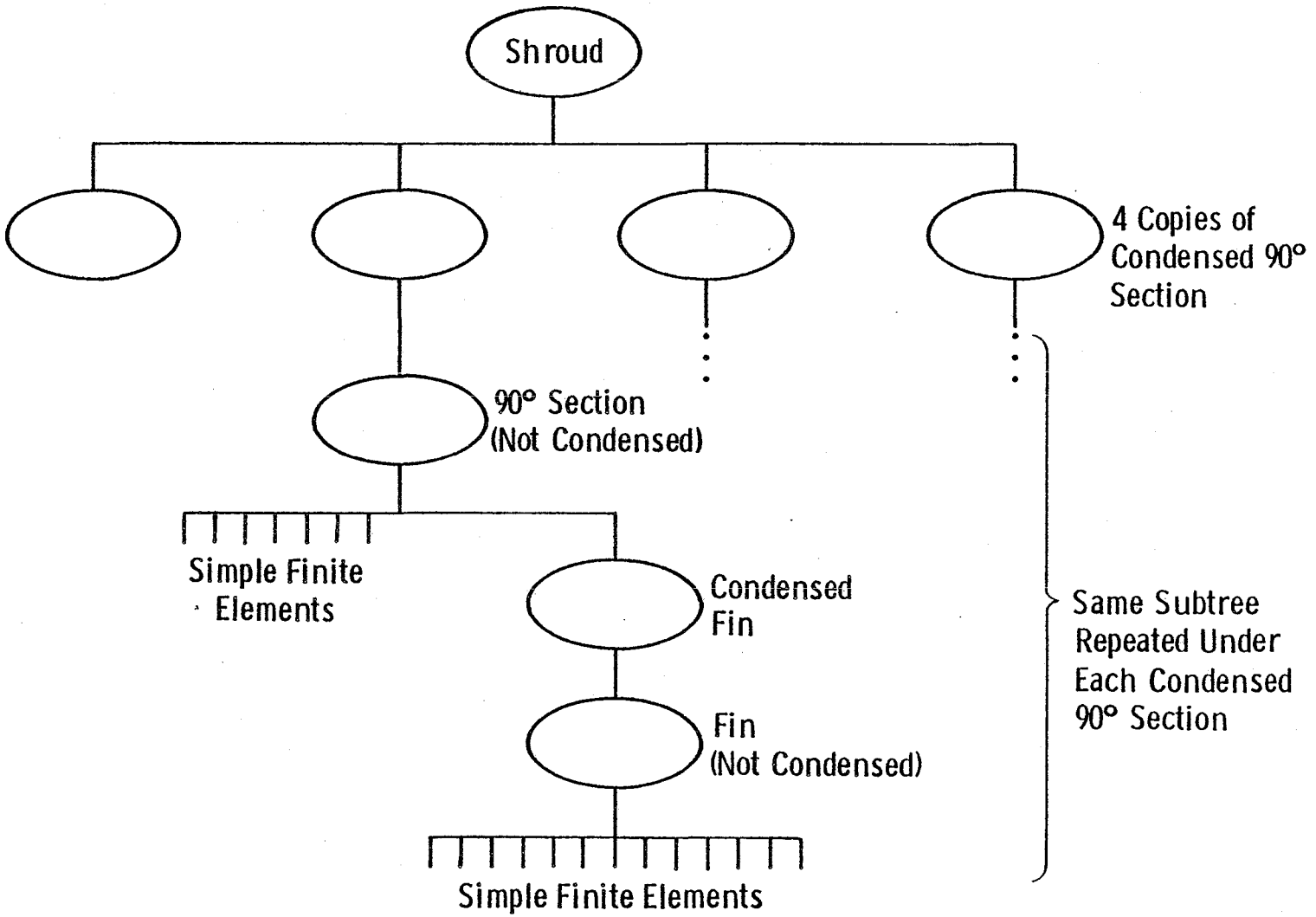


Figure 2.19 -- Structural Model Hierarchy for Exhaust Duct

Model	CPU	I/O
(1) 90° Standard	0.69	0.63
(2) 90° Substructured	0.60	0.54
(3) 90° Condensation	0.92	0.89
(4) 360° Substructured	1.0	1.00
(5) 360° Standard	13.6*	18.92*

*Estimated using CPU and I/O measures for the 90° standard model

Table 2.1 -- CPU Times for the Exhaust Duct Analysis

Figure 2.20 provides a detailed breakdown of CPU among the various POLO-FINITE subsystems for solution of the 360 degree substructured model. Not surprisingly, the major effort is expended in the assembler and equation solver modules. It is interesting to note that a relatively small percentage of the total job time (14%) is expended in data management activities. The percentage of data management time in the assembler is larger due to the very large number of small matrices that must be manipulated. In contrast, the equation solver accesses data in larger blocks (50x50) and performs a significant amount of numerical computation on them. Consequently, percentage of data management time in the solver is very small.

2.6.2 Nonlinear Example

Figure 2.21 shows one quarter of a thick, shallow shell structure constructed of impact resistant acrylic. A monotonically increasing pressure is applied over a small region at the apex to simulate the impact of a projectile. Yielding of material in and around the impact zone is the nonlinear behavior of interest. Large geometry changes are ignored in the present analysis.

Figure 2.22 shows the details of the element grids. The square panel represents the element grid projection onto the global X-Y plane. The middle surface of the shell lies on the surface of a sphere having a radius of 60 in. (152.4 cm). The rise along diagonal A-D is 3.15 in. (8 cm). Edges C-D and B-D lie on symmetry planes; edges A-C and A-B are completely fixed (no translation). A simple von Mises yield criterion with associated flow rule is adopted for the nonlinear material

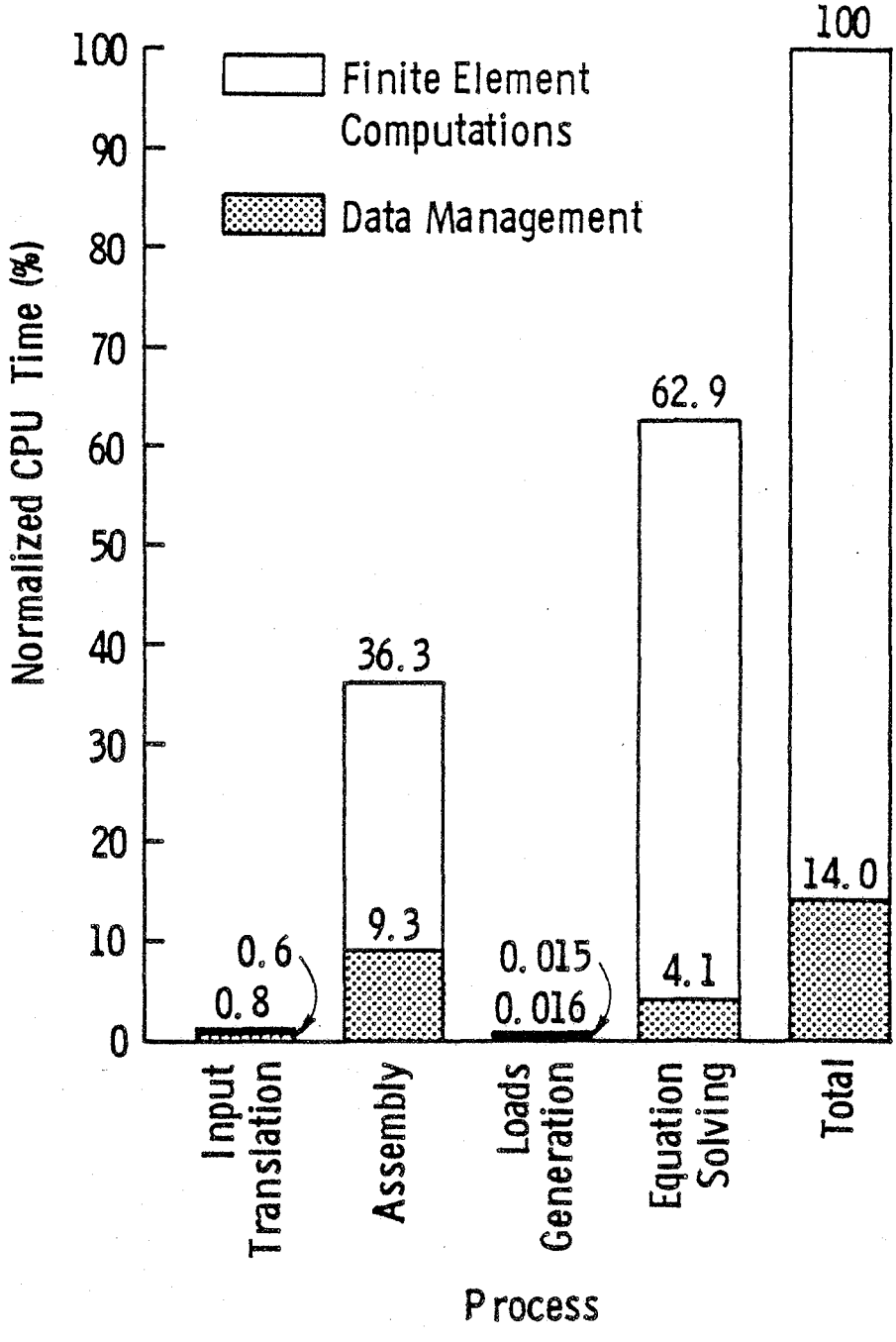


Figure 2.20 -- CPU Time Distribution for Exhaust Duct Analysis, 360° Substructured Model

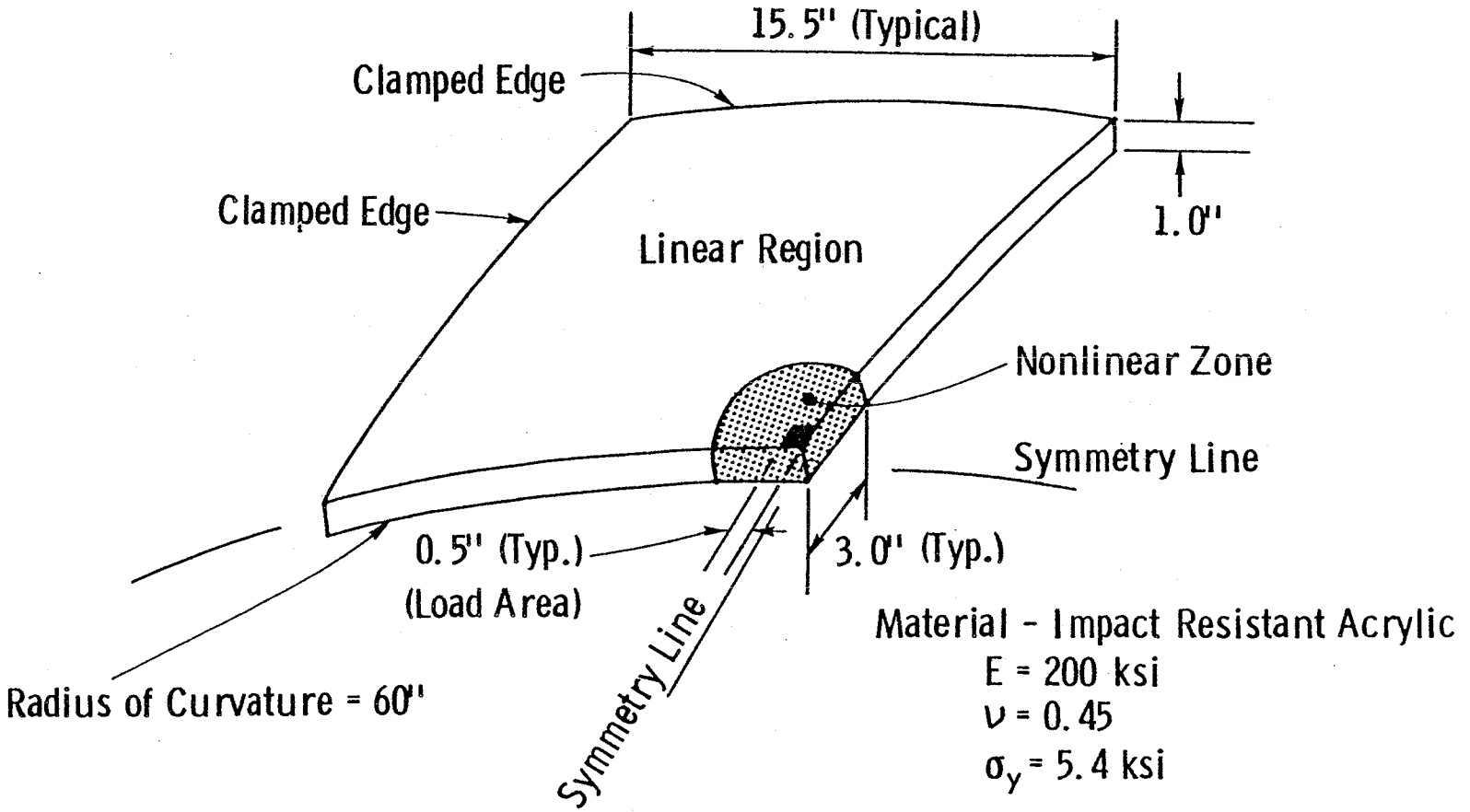


Figure 2.21 -- Shallow Shell Subjected to Projectile Impact

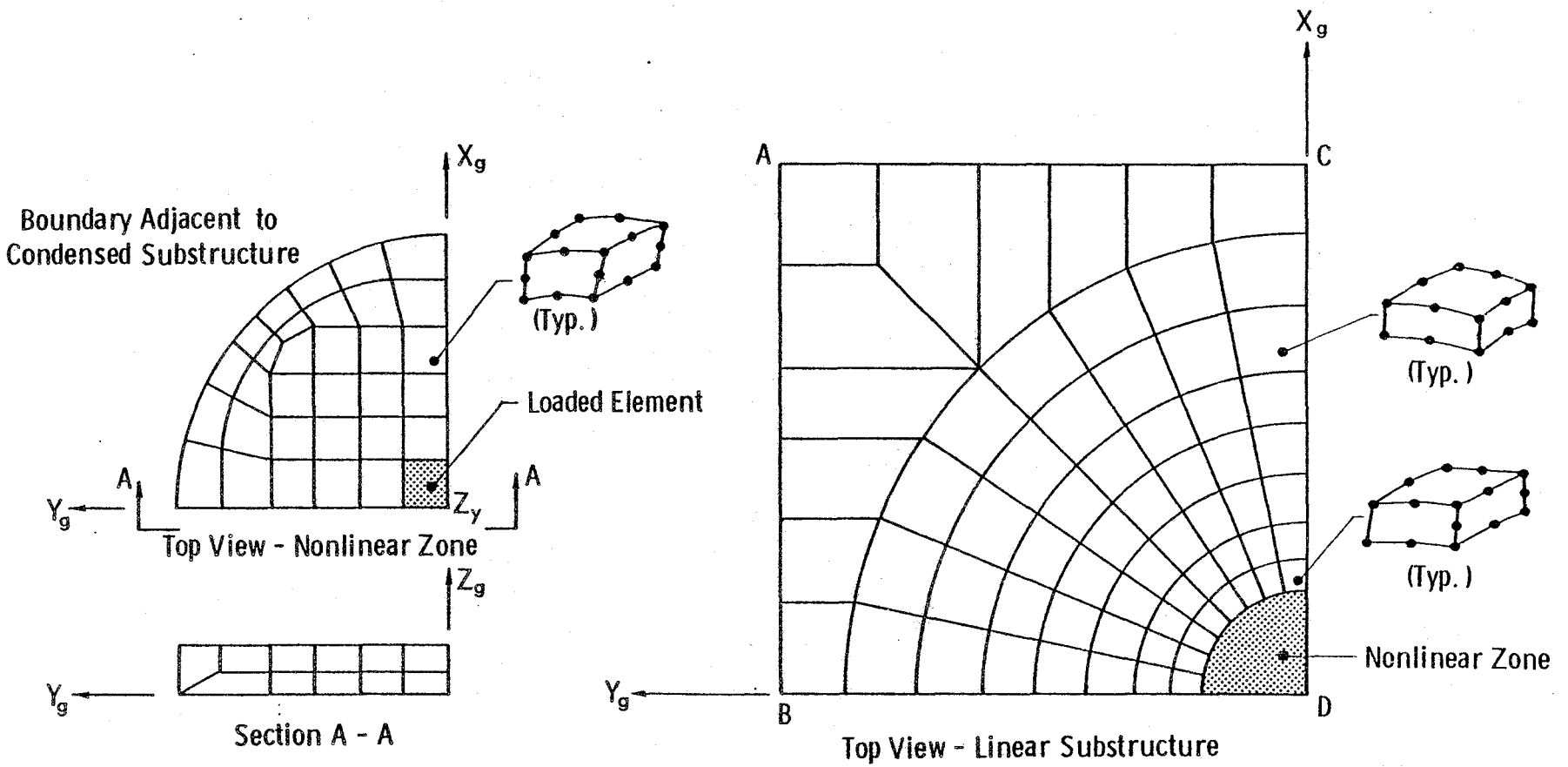


Figure 2.22 -- Substructures for Nonlinear Impact Example

response. The uniaxial stress-strain curve is idealized as elastic-perfectly plastic for simplicity.

Three types of 3-D solid isoparametric elements are used to model the shell. Each node of these elements has 3 translational degrees of freedom (u,v,w). In the impact zone, 20 node parabolic elements are employed as shown in Fig. 2.22. Two elements are used in the thickness direction very near the loaded region. Away from the impact point, one 16 node thick shell element is adequate through the thickness. A ring of 18 node transition elements connect the 20 node and 16 node elements without loss of displacement compatibility. A 2x2x2 Gauss integration order is used in all elements.

Figure 2.22 defines the substructured model components. The outer linear region has 483 nodes (1449 DOF), 59 thick shell elements, and 8 transition elements. Condensation reduces the number of nodes to the 43 (129 DOF) that interface with the nonlinear region. The natural node numbering scheme places retained nodes last in [K], which eliminates reordering. The highest level structure consists of 56 nonlinear parabolic elements in addition to the condensed linear substructure. The final nonlinear model has 376 nodes (1128 DOF). The standard model for this structure has exactly the same element grid but without substructuring. This model has 818 nodes (2454 DOF) and 123 elements on which the nonlinear solution is performed. The substructured model thus reduces the number of DOF by 54%.

A total of four load increments were applied to each model with the modified Newton-Raphson procedure to distribute residual forces. Iterations at constant external load were conducted until the Euclidean norm of the residual force vector fell below 1% of the applied load vector norm. Tangent stiffness updates were performed before iterations 2,5, and 8 in each step. Table 2.2 provides the number of yielded integration points, the number of stiffness updates, and the number of iterations for each load increment. The fourth load increment propagated the plastic zone into the third element band from the apex. Sufficient CPU timing data was collected during the first four load steps for the desired comparisons.

Figure 2.23 compares CPU time for the standard and substructured models. The normalized CPU time used through each load step is expressed as a percentage of the total time required for the standard model analyzed through load step four. Solution times for load step one are nearly equal. The standard model requires slightly more time due to the larger number of elements for which strains and stresses are computed. The results for step one clearly demonstrate the negligible overhead for controlling the substructured solution. The major CPU savings with the substructured model begin to occur in load steps three and four. At the end of step four, the substructured model used 72% of the CPU required for the standard model. Both solutions appear to have reached a steady-state condition (constant slopes) at load step three. Linear extrapolation of these curves to one hundred load steps shows that solution of the substructured model would require 64% of the standard model solution time. The improvement from 72% to 64% arises from

Load Increment	Number of Yielded Points	Number of Iterations	Number of $[K_T]$ Updates
1	1	2	1
2	6	5	2
3	11	4	1
4	8	3	1

Table 2.2 -- Computation Summary for Nonlinear Example (Identical Results for Substructured and Standard Models)

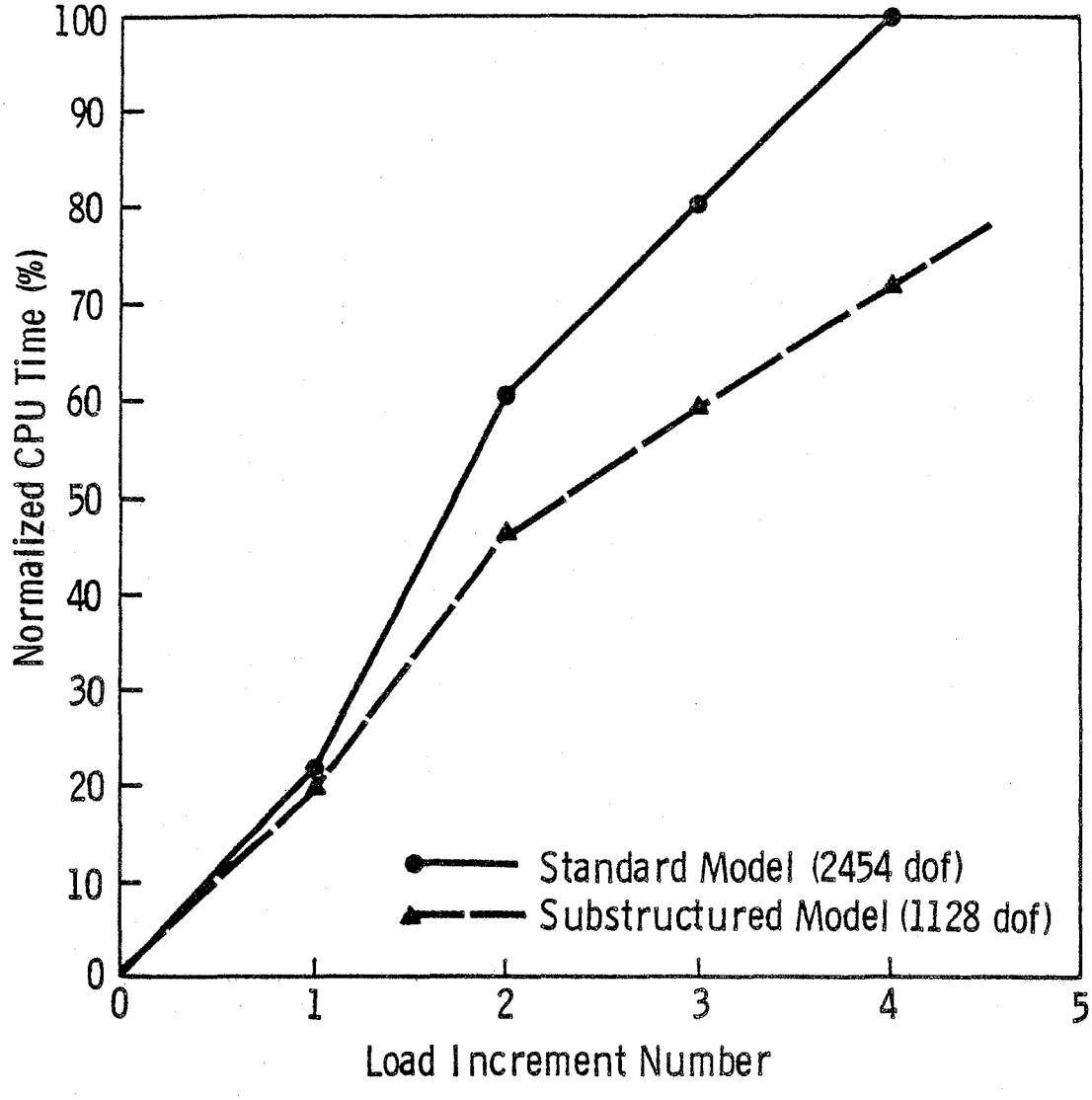


Figure 2.23 -- CPU Time Comparison for the Example Nonlinear Analysis

the decreasing significance of equal times in step one and equation solver savings in the substructured model.

The bar graph of CPU distribution among processing modules shown in Fig. 2.24 indicates the source of CPU savings. Input translation and strain-stress computation (which includes material updating) consume an insignificant percentage of the CPU time. The equation solver is most dominant followed by the stiffness assembler (which includes element computations) and the residual loads generator. Savings in the equation solver for the substructured model are approximately 25% at the end of step four. This figure would increase to 40% with additional stiffness updates during subsequent load steps. The substructured nonlinear region with 1128 DOF requires 40% less CPU to triangulate than the full 2454 DOF nonlinear model.

Little savings accrue in the assembler for the substructured model; however, this is to be expected. After step one, there is very little difference in the operations performed by the stiffness assembler for each model. The same number of nonlinear element stiffnesses are generated regardless of whether or not the model is substructured. The small savings result from the reduced number of element stiffnesses assembled in the substructured model. Moreover, this is an I/O rather than CPU intensive activity.

Figure 2.25 compares I/O activity for the two models. The I/O activity through each load step is again normalized by the total I/O througn load step four of the standard model. The curves follow the same trend as the CPU time in Fig. 2.23. The substructured model savings are 33% at the end of step four, which is slightly better than

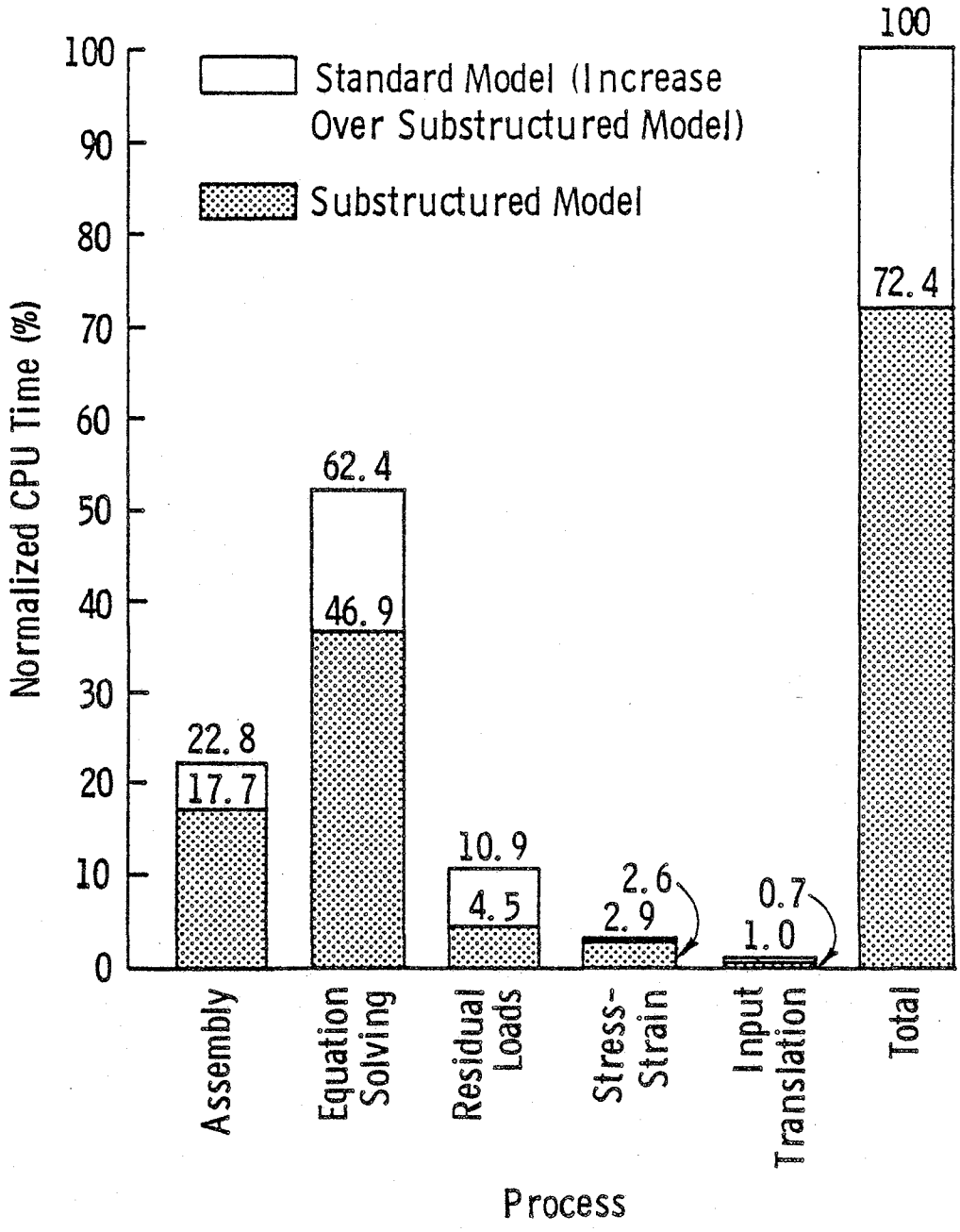


Figure 2.24 -- CPU Time Distribution for the Example Nonlinear Analysis

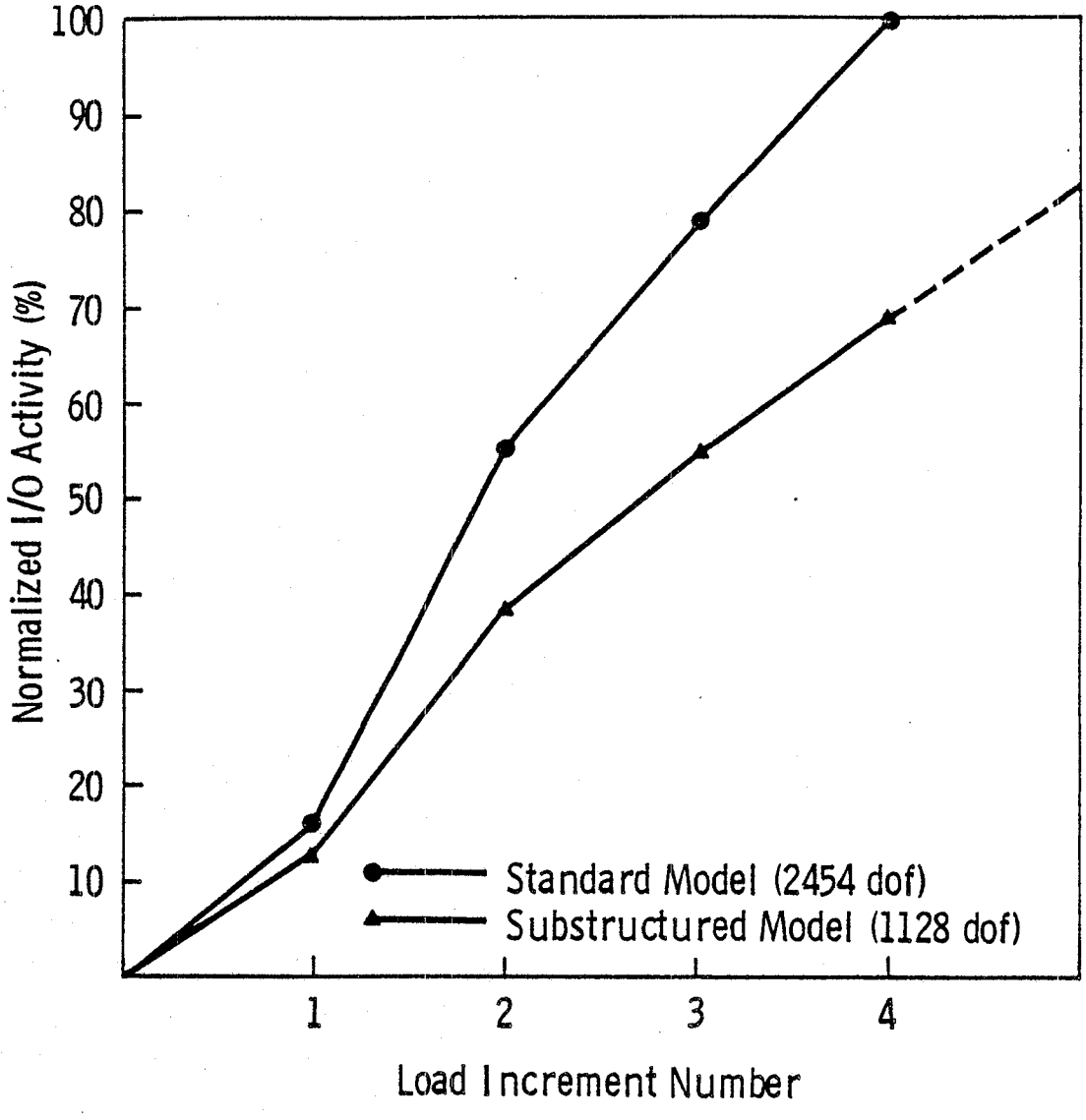


Figure 2.25 -- I/O Comparison for the Example Nonlinear Analysis

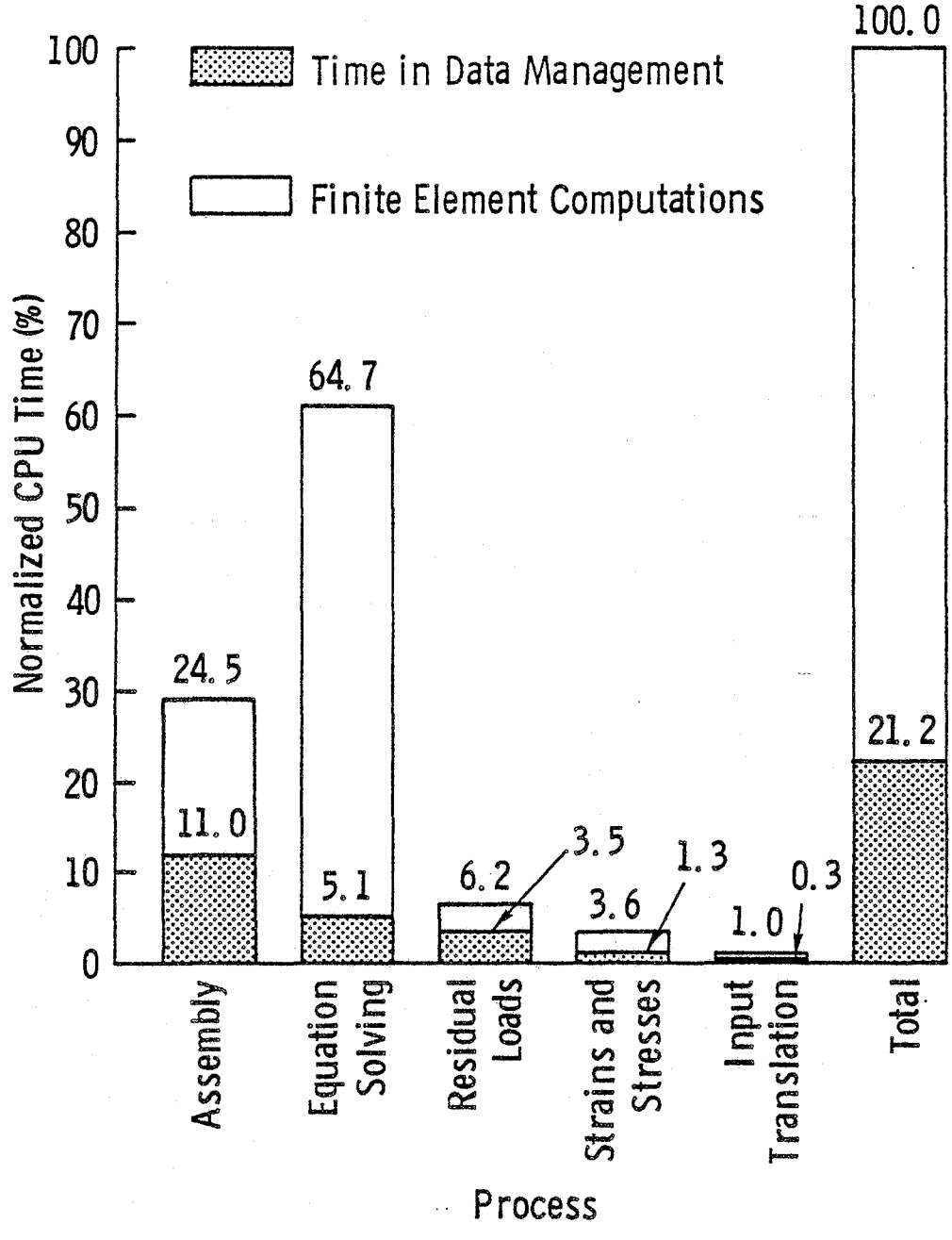


Figure 2.26 -- CPU Time Distribution for the Substructured Nonlinear Example

the CPU savings. Most of the I/O savings occurred in the equation solving file.

Figure 2.26 provides a breakdown of the CPU time distribution among POLO-FINITE subsystems for the substructured solution. A similar comparision for the linear example is given in Fig. 2.20. These two figures illustrate the increased percentage of job CPU time expended in data management activities for the nonlinear analysis. While it is not strictly valid to compare a linear analysis of one structure with the nonlinear analysis of another structure, the same trend of increased data management time has been observed in other analyses performed with POLO-FINITE. For most linear analyses, data management activities represent 10-25% of the total CPU time. In contrast, data management activities consistently require a larger percentage of the CPU time for nonlinear analyses. The increase is attributed to the much larger data base sizes (generally a factor of 10), the larger number of data structures that must be accessed to obtain nonlinear data, and the large increase in solution logic that requires movement and duplication of data without computation.

2.7 References

2.1 Dodds, R.H., and Lopez, L.A., "Generalized Software for Nonlinear Analysis," International Journal for Advances in Engineering Software, Vol. 2, No. 4, 1981, pp. 161-168.

2.2 Dodds, R.H., and Lopez, L.A., "Substructuring in Linear and Nonlinear Analysis," International Journal for Numerical Methods in Engineering, Vol. 15, 1980, pp. 583-597.

2.3 Egeland, O., and Araldsen, P., "SESAM-69: A General Purpose Finite ELement Program," Computers and Structures, Vol. 4, No. 1, 1974, pp. 41-68.

2.4 Fuchs, G. von, and Schrem, E., "Hypermatrix Solution of Large Sets of Symmetric, Positive-Definite Linear Equations." Computer Methods in Applied Mechanics and Engineering, No. 1, 1972, pp. 197-216.

2.5 Furuike, T., "Computerized Multiple Level Substructured Analysis," Computers and Structures. Vol. 2. 1972. pp. 695-712.

2.6 Irons, B.M., "A Frontal Solution Program for Finite Element Analysis," International Journal for Numerical Methods in Engineering, Vol. 2, No. 1, 1970, pp. 5-32.

2.7 Iwaki, T., Maeda, A., and Ishii, T., "MISA: A General Purpose FEM Program," Computers and Structures, Vol. 10, 1979, pp. 311-322.

2.8 Jennings, A., and Tuff, A.D., "A Direct Method for the Solution of Large Sparse Symmetric Simultaneous Equations," in Large Sparse Sets of Linear Equations, Ed. J.K. Reid (Academic Press, N.Y.), 1971.

2.9 Lopez, L.A., "FINITE: An Approach to Structural Mechanics Systems," International Journal for Numerical Methods in Engineering, Vol. 11, No. 5, 1977, pp. 851-866.

2.10 McKellar, A.C., and Coffman, E.G., "Organizing Matrices and Matrix Operations for Paged Memory Systems," Communications of the ACM, Vol. 12, No. 3, March, 1969, pp. 153-165.

2.11 MacNeal, R. and McCormick, C.W., "The NASTRAN Computer Program for Structural Analysis," Computers and Structures, Vol. 1, No. 3, 1971, pp. 389-412.

2.12 Meyer, C., "Solution of Linear Equations--State of the Art," Journal of the Structural Division, ASCE, Vol. 99,No. ST7, 1973, pp. 1507-1526.

2.13 Mondkar, D.P., and Powell, G.H., "Large Capacity Equation Solver For Structural Analysis," Computers and Structures, Vol. 4, 1974, pp. 699-728.

2.14 Peterson, H. and Popov, E.P., "Substructuring and Equation System Solutions in Finite Element Analysis," Computers and Structures, Vol. 7, 1977, pp. 197-206.

2.15 Przemieniecki, J.S., "Matrix Structural Analysis of Substructures," AIAA Journal, 1963, pp. 138-147.

2.16 Schrem, E., "From Program Systems to Programming Systems for Finite Element Analysis," Proceedings, U.S.-German Symposium on Formulational and Computational Methods in Finite Element Analysis, 1976, MIT Press.

2.17 Taig, I.C., "Automated Stress Analysis Using Substructures," Proceedings, 1st Wright-Patterson Conference, AFFDL TR, 1966, pp. 66-80.

2.18 Williams, F.W., "Comparison of Sparse Matrix and Substructure Methods," International Journal for Numerical Methods in Engineering, Vol. 5, 1973, pp. 383-394.

2.19 Wilson, E.L., "The Static Condensation Algorithm," International Journal for Numerical Methods in Engineering, Vol. 8, No. 1, 1974, pp. 198-203.

CHAPTER 3

DYNAMIC REDUCTION OF STIFFNESS AND MASS MATRICES

3.1 General

As demonstrated in Chapter 2, multilevel substructuring provides an economical approach to static analysis of very large linear and nonlinear structural models. Size reduction of the substructures, through nodal condensation, yields an exact and economical solution to the statically loaded problem. Since dynamic analysis of a finite element model requires significantly more computational effort than a static analysis of the same model, an analogous reduction scheme would be useful in dynamics.

As an illustration of the usefulness of substructured models in dynamics, consider the following example. Assume that a particular structural model contains 5000 DOF and has a half-bandwidth of 500. Computation of the 50 lowest natural frequencies and corresponding mode shapes by a method suitable to the problem characteristics requires roughly $7.6(10^9)$ operations. Now suppose that the model can be divided into five identical substructures, each containing 1200 DOF. Reduction of a substructure to 100 independent DOF while retaining the 10 lowest natural frequencies and mode shapes requires roughly $7.5(10^8)$ operations. Since all five of the substructures are identical, the reduction must be performed only once. Assembly of the substructures into final form results in the reduced model containing only 300 independent DOF. Since the equations for the substructured model are fully populated, a different procedure may be appropriate for computation of natural fre-

quencies. Computation of the 50 lowest frequencies and mode shapes would require only $2.3(10^7)$ operations. Thus, a savings in required operations of a full order of magnitude can be realized by using substructuring on this hypothetical model. These operation counts are, of course, highly dependent upon the algorithms used and the model being analysed, but the computational savings is well illustrated. Additional savings can be gained in the solution of the equations of motion and in the recovery of substructure displacements, strains, and stresses.

The goal of dynamic reduction methods is to generate stiffness and mass matrices that accurately represent the stiffness and inertia characteristics of the substructure with the minimum number of DOF. As previously stated, reduction in static analysis is exact and can be mathematically viewed as an equation solving technique. In dynamic analysis, however, exact dynamic reduction of an individual substructure is dependent upon the unknown frequencies of the total structural system. Since these system frequencies are actually objectives of the analysis, the analyst must use reduction methods which are either iterative or frequency independent (and therefore approximate).

Two classes of methods for dynamic reduction have evolved for use with the FEM. The first class, known as Guyan reduction [3.6], is an extension of static condensation. It is currently the technique most widely used to reduce the number of DOF prior to frequency or transient analysis of standard (nonsubstructured) finite element models. The method involves elimination of DOF that are assumed to have a negligible effect on mode shapes and thus vibration response of the structure. Dynamic results, especially strains and stresses, are generally quite

sensitive to the choice of DOF to be eliminated. Although the method has proven useful for smaller models, its extension to multilevel substructuring is expected to be only marginally successful.

The second class of dynamic reduction techniques, termed modal synthesis, contains methods that rely on a Rayleigh-Ritz transformation of each substructure's geometric coordinates to a smaller set of generalized coordinates. This transformation is usually based on the natural frequencies and mode shapes of the isolated substructures. The reduced stiffness and mass matrices for the substructure are defined in terms of the generalized coordinates. Assembly of the stiffness and mass matrices for the next higher level structure is based on displacement compatibility. The reduction process can then be successively repeated for each additional level of substructuring. Many techniques in the category of modal synthesis have been devised in an attempt to select the best combination of substructure modes and displacement compatibility conditions.

This chapter provides detailed descriptions of the initial formulations of Guyan reduction and modal synthesis. Derivations of the governing equations of the methods are followed by a brief review of their respective extensions and modifications. Although many more of these alternate techniques have been proposed in the open literature than are presented here, those discussed effectively encompass the breadth of the topic. The chapter concludes with an evaluation and comparison of the dynamic reduction techniques and recommendations for implementation in a general purpose software system.

3.2 Guyan Reduction

3.2.1 Basic Formulation

Consider an isolated substructure consisting of simple elements, such as the FIN substructure used in the exhaust duct example problem of Chapter 2 (see Figure 3.1). Let the internal boundary of the substructure identify its interface with other substructures while an external boundary corresponds to the physical boundary of the entire structure. The undamped, free vibration equation of the substructure, partitioned to seperate master (m) and slave (s) DOF, is

$$\begin{bmatrix} K^{mm} & K^{ms} \\ K^{sm} & K^{ss} \end{bmatrix} \begin{Bmatrix} u^m \\ u^s \end{Bmatrix} - \omega_i^2 \begin{bmatrix} M^{mm} & M^{ms} \\ M^{sm} & M^{ss} \end{bmatrix} \begin{Bmatrix} u^m \\ u^s \end{Bmatrix} = \{0\}. \tag{3.1}$$

Master DOF are those that will remain after condensation and are usually chosen to lie on the internal boundary of the substructure. They are used for connectivity to adjacent substructures. The slave DOF are those to be eliminated and usually lie in the interior of the substructure or on its external boundary. The natural frequency ω_i is that of the complete structural system, not just the isolated substructure. The presence of nonzero off-diagonal blocks $[M^{sm}]$ and $[M^{ms}]$ in Eq. (3.1) implies the use of a consistent mass formulation. When a lumped mass model is used, the mass matrix is diagonal.

The lower half of Eq. (3.1) can be expanded to

$$([K^{sm}] - \omega_i^2[M^{sm}])\{u^m\} + ([K^{ss}] - \omega_i^2[M^{ss}])\{u^s\} = \{0\}. \tag{3.2}$$

Solving for $\{u^s\}$ in terms of $\{u^m\}$ yields a coordinate transformation

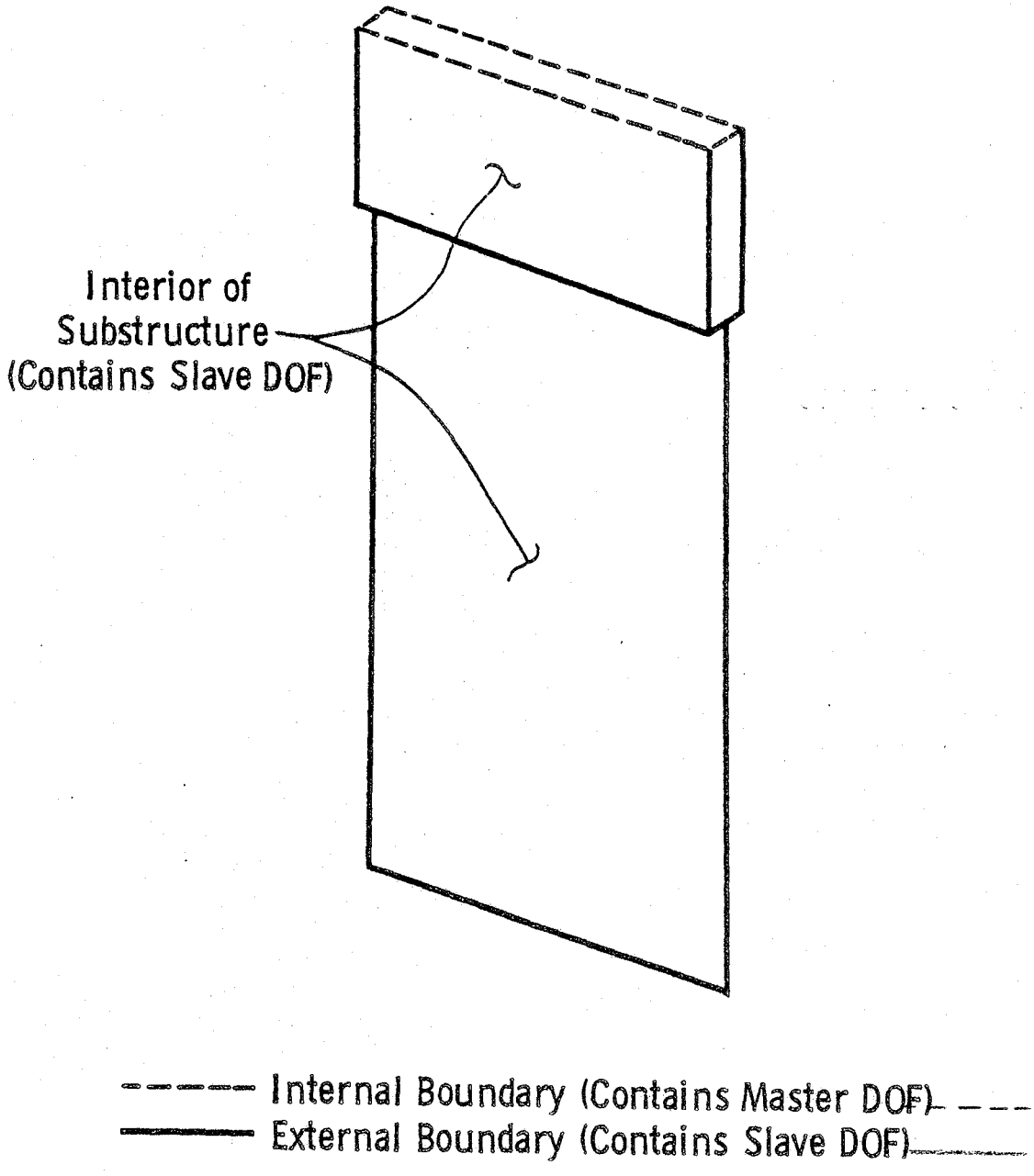


Figure 3.1 -- Internal and External Boundaries for Substructure FIN from Chapter 2 Exhaust Duct Example

which is dependent on the unknown system vibration frequency ω_i. If it is assumed that the inertia forces on the slave DOF are small compared to the static forces, the former can be neglected. Thus, the frequency dependence is eliminated and Eq. (3.2) simplifies to

$$[K^{ss}]\ \{u^s\} = -[K^{sm}]\ \{u^m\}. \tag{3.3}$$

Defining the coordinate transformation $[T_G]$ from $\{u^m\}$ to $\{u^s\}$ as

$$\{u^s\} = [T_G]\ \{u^m\} \tag{3.4}$$

$\{u^s\}$ can be eliminated from Eq. (3.3) resulting in

$$[K^{ss}]\ [T_G] = -[K^{sm}]. \tag{3.5}$$

As discussed in Chapter 2, $[T_G]$ is evaluated by standard equation solving techniques. It is important to recall that the columns of the transformation matrix are composed of the previously defined static constraint modes. The complete substructure displacement vector can be re-expressed, using Eq. (3.4), as

$$\{u\} = \left\{\begin{array}{c} u^m \\ \hline u^s \end{array}\right\} = \left[\begin{array}{c} I \\ \hline T_G \end{array}\right] \{u^m\}. \tag{3.6}$$

The substructure strain and kinetic energies are given by

$$S.E. = 1/2 \left\{ \begin{array}{c} u^m \\ \hline u^s \end{array} \right\}^T \left[\begin{array}{c|c} K^{mm} & K^{ms} \\ \hline K^{sm} & K^{ss} \end{array} \right] \left\{ \begin{array}{c} u^m \\ \hline u^s \end{array} \right\} \quad , \text{ and} \tag{3.7a}$$

$$K.E. = 1/2 \left\{ \begin{array}{c} \dot{u}^m \\ \hline \dot{u}^s \end{array} \right\}^T \left[\begin{array}{c|c} M^{mm} & M^{ms} \\ \hline M^{sm} & M^{ss} \end{array} \right] \left\{ \begin{array}{c} \dot{u}^m \\ \hline \dot{u}^s \end{array} \right\} \quad ; \tag{3.7b}$$

where $\{\dot{u}\}$ is the first time derivative of $\{u\}$. These expressions are rewritten in terms of the master DOF by substituting Eq. (3.6) into Eq. (3.7). The resulting Guyan reduced stiffness $[K_G]$ and mass $[M_G]$ are given by

$$[K_G] = [K^{mm}] + [K^{ms}]\,[T_G] \quad \text{and} \tag{3.8}$$

$$[M_G] = [M^{mm}] + [T_G]^T[M^{ss}]\,[T_G] + [T_G]^T[M^{sm}] + [M^{ms}][T_G]. \tag{3.9}$$

For the simpler case of a lumped mass model, Eq. (3.9) reduces to

$$[M_G] = [M^{mm}] + [T_G]^T[M^{ss}]\,[T_G]. \tag{3.10}$$

In Guyan reduction, the inertia effects of the slave DOF are not lost. Instead, the contribution to the mass of these eliminated DOF is distributed to the master DOF. It is assumed that the dynamic response of the slave DOF of the substructure is adequately approximated from that of the master DOF by linear combinations of the static constraint modes. Note that regardless of which mass matrix formulation is used, consistent or lumped, the reduced mass matrix, $[M_G]$, will be fully populated. This situation must be considered when choosing algorithms

for solving the free vibration and transient response problems for the assembled structure.

When loads are applied to the slave DOF, they too must be condensed. If the substructure is subjected to an arbitrary virtual displacement, $\{\delta u\}$, the work done by the substructure forces $\{P\}$ is

$$\delta W = \{\delta u\}^T \{P\}. \tag{3.11}$$

The statically equivalent condensed forces, $\{F\}$, applied at the master DOF must do the same work during a virtual displacement consistent with $\{\delta u\}$, so

$$\{\delta u^m\}^T \{F\} = \{\delta u\}^T \{P\}. \tag{3.12}$$

Recalling Eq. (3.6), the condensed force vector becomes

$$\{F\} = \begin{bmatrix} I \\ \hline T_G \end{bmatrix}^T \{P\}. \tag{3.13}$$

Each substructure has its stiffness, mass, and loads similarly partitioned and reduced. Assembly of both the reduced substructure mass and stiffness into the next higher level follows the procedures outlined in Chapter 2. Geometric compatibility between substructures is automatically assured by the use of the master DOF as generalized coordinates.

The extension of Guyan reduction to multilevel substructuring is straightforward. Referring to the terminology of Chapter 2, assume that all substructures at level "i" have been assembled either from simple elements or level "i+1" substructures. The level "i-1" substructures

are built up by selecting slave and master DOF for each substructure at level "1", condensing these substructures using Eq. (3.8) and Eq. (3.9), and assembling as described above. When the highest level structure is reached, the final condensed stiffness and mass matrices can be used to form the equations of motion for the entire structure.

After a free vibration problem has been solved for the highest level structure, it may be necessary to recover the portion of the system mode shapes contained within lower level condensed substructures. This is achieved by simply applying Eq. (3.4) recursively to each substructure to recover the components of the vibration modes at the slave DOF from the master DOF. Recovery of the displacement patterns after a transient analysis is not as elementary. A procedure analogous to the computation of partial slave displacements discussed in Chapter 2 must be adopted. During the condensation process, loads applied at the slave DOF are transformed into work-equivalent forces at the master DOF. Thus, application of Eq. (3.4) to displacement patterns at the master DOF does not yield the total response of the slave DOF. A separate transient analysis of the substructure with its master DOF fixed is required. The displacements of the slave DOF from this analysis must then be superposed with those from Eq. (3.4) to obtain the total response of the substructure. Of course, this procedure is unnecessary when the internal response of condensed substructures is of no interest in the analysis of the structure.

3.2.2 Automatic Selection of Master DOF

To insure complete geometric compatibility, all DOF at the internal boundaries of a substructure must be included in the set of master DOF. However, the set need not be limited to the DOF on the internal boundary of the substructure. Other substructure DOF are possible candidates for retention.

Henshell and Ong [3.7] have suggested a simple method for automating the process of selecting master DOF. The method is based on an assumption fundamental to the development of Guyan reduction; that the mass terms corresponding to the slave DOF have a negligible effect on the mode shapes. This can be rephrased by saying that k_{ss}/m_{ss} is large relative to k_{mm}/m_{mm} for any pair of slave and master DOF. Thus the obvious candidates for master DOF, in addition to those on the substructure internal boundary, are those with the smallest ratios k_{ii}/m_{ii}; i=1,2,...,n; n=number of substructure internal DOF.

The method presented in Ref. [3.7] was oriented towards standard finite element models. Its extension to multilevel substructuring requires only the addition of the internal boundary DOF in the set of master DOF.

The above decision criterion is equivalent to preserving the lowest vibration modes of the substructure. The ratio k_{ii}/m_{ii} is interpreted as the square of the vibration frequency associated with the "i" th DOF while all other substructure DOF are held fixed. Retention of the DOF with the smallest values of k_{ii}/m_{ii} has shown to produce better accuracy in the lower modes than random selection of master DOF.

3.2.3 Improved Displacement Recovery

The frequency dependent transformation from $\{u^m\}$ to $\{u^s\}$, when cast in a modified form, is useful for improving recovery of substructure mode shapes and displacements. The technique was initially presented by Kidder [3.15] and was later reintroduced with the addition of some numerical results by Miller [3.20]. The technique is applied after the structure equations have been reduced by standard Guyan reduction. The frequency dependent transformation matrix resulting directly from Eq. (3.2) is

$$[T_\omega] = -(-\omega^2[M^{ss}] + [K^{ss}])^{-1} (\omega^2[M^{sm}] - [K^{sm}]). \qquad (3.14)$$

Expansion of the inverse term gives

$$(-\omega^2[M^{ss}] + [K^{ss}])^{-1} = [K^{ss}]^{-1} + \omega^2[K^{ss}]^{-1}[M^{ss}][K^{ss}]^{-1} + \ldots \qquad (3.15)$$

Ignoring as small the terms containing ω to powers greater than two, the transformation becomes

$$[T_\omega] = [K^{ss}]^{-1}(-[K^{sm}] + \omega^2([M^{sm}] - [M^{ss}][K^{ss}]^{-1}[K^{sm}])). \qquad (3.16)$$

As with $[T_G]$, $[T_\omega]$ can be evaluated by equation solving rather than by computing $[K^{ss}]^{-1}$.

The transformation $[T_\omega]$ replaces $[T_G]$ in recovering the substructure mode shapes and displacements during a transient analysis of the uncoupled linear equations of motion. Upon solution of the modal equations at the highest level, mode shape recovery is achieved via

$$\{u^s\} = [T_\omega]\{u^m\}. \tag{3.17}$$

This transformation is performed for each modal frequency considered so as to yield individual mode shapes within the substructure. Displacement recovery follows the procedure outlined in Sec. 3.2.1 with the total displacement vector equal to the sum of its modal components.

The improved displacement recovery technique has no effect on the computed system frequencies and highest level structure mode shapes. The argument for its use is that frequencies determined from substructure mass and stiffness matrices computed with $[T_G]$ are generally realistic. However, improvements are needed in the substructure mode shapes from which strains and stresses are derived.

Miller's numerical results demonstrated greatly improved mode shape vectors over standard Guyan reduction for frame structures. However, the effectiveness of the method is limited to the lower frequencies of vibration. This is because the size of the truncated terms of the series in Eq. (3.15), and thus the error in $[T_\omega]$, grows with increasing values of ω.

In order to use improved displacement recovery, the equations of motion of the highest level structure must be solved in their uncoupled form so that modal vibration frequencies can be computed. When a transient analysis of coupled equations is performed, as in nonlinear analysis, the displacements are computed directly and thus, this method is not applicable.

3.2.4 Evaluation of Guyan Reduction Techniques

Guyan reduction techniques have yet to be applied to multilevel substructured models. Therefore, their performance in such an application remains unknown. Nevertheless, Guyan reduction has some important advantages that could make it an attractive approach to dynamic reduction in particular cases. First, its development is taken directly from the static condensation approach. This allows the reduction scheme to be readily added to existing software which is currently capable of handling multilevel substructured models for static analysis. Secondly, reasonably good numerical results have been achieved with Guyan reduction in computing system frequencies for small models. This feature makes the technique attractive for preliminary vibration analysis. Lastly, Guyan reduction is the least expensive of all dynamic reduction techniques.

Some drawbacks of Guyan reduction are evident and must be considered. The success of the method is highly dependent upon the choice of master DOF. This is further complicated by the need to include internal boundary DOF in the set of master DOF for substructured models. The result is likely to be a decrease in the degree of reduction capable for large, multilevel substructured models. Another problem is the quality of the mode shapes. Accurate prediction of strains and stresses requires that displacement vectors and mode shapes be well formed. The ability to achieve this goal with Guyan reduction is still in doubt. However, there are circumstances in which it is not necessary to recover strains and stresses within a condensed substructure. In such cases, i.e. when the response of only the highest level structure is of major

concern, Guyan reduction may perform quite well.

Guyan reduction techniques provide an important first step in the development of more sophisticated dynamic reduction methods. As will be seen later, the procedures derived above actually represent a degenerated case of modal synthesis. Guyan reduction does not appear adequate for general application to dynamic analysis of multilevel substructured models. However, with automatic selection of master DOF and improved displacement recovery, it does hold potential as an economical approach to dynamic reduction of certain models.

3.3 Modal Synthesis

3.3.1 Introduction

Modal synthesis was developed expressly for use with substructured models. Although this discussion is limited to reduction of finite element models, the techniques have also been applied to distributed systems. All modal synthesis transformations are based on Rayleigh-Ritz arguments. The procedure involves the derivation of a transformation matrix composed of a truncated set of mode shape vectors that adequately describe the dynamic characteristics of the substructure. This set is fewer in number than the number of independent DOF contained in the substructure. With the transformation matrix, individual substructure mass and stiffness matrices are converted from geometric coordinates into a reduced set of generalized coordinates. The generalized mass and stiffness matrices for each substructure are then synthesized while maintaining geometric compatibility along internal boundaries to form similar matrices for the next higher level structure. In a multilevel substructured environment the transformation and assembly processes are performed recursively at each level.

There are two basic operations that must be performed with any application of modal synthesis. First, an approach must be chosen for selecting the set of substructure mode shapes from which the reduced substructure matrices are computed. Second, a procedure is needed to enforce geometric compatibility along substructure internal boundaries. The numerous modal synthesis techniques proposed in the literature vary in how these concepts are implemented.

The initial formulation of modal synthesis, accredited to Hurty [3.11, 3.12], has been extensively modified and enhanced. The following section presents Hurty's formulation, commonly referred to as the fixed-interface method, while the later sections describe suggested improvements or alternatives to the method.

3.3.2 Fixed-Interface Method

The origin of modal synthesis techniques lies in Hurty's fixed-interface method. However, details of the procedure presented here parallel the development by Craig and Bampton [3.3]. Hurty's development required a distinction between "statically determinate" and "redundant" constraints. Statically determinate and redundant constraints are artificial constraints applied at all master DOF. They are imposed independently of any actual physical constraints that may already exist on the substructure at slave DOF. The set of statically determinate constraints serve to restrain any rigid-body motion that may be possible. Redundant constraints are those applied to the remaining master DOF. Figure 3.2 illustrates the application of these constraints to a simple two-dimensional plate. Three statically determinate constraints are needed to restrain translation and rotation. The five remaining master DOF have redundant constraints applied. Craig and Bampton treat the two sets of constraints simply as boundary constraints.

Consider again the undamped, free vibration equation for an isolated substructure composed of simple elements and partitioned to separate master and slave DOF

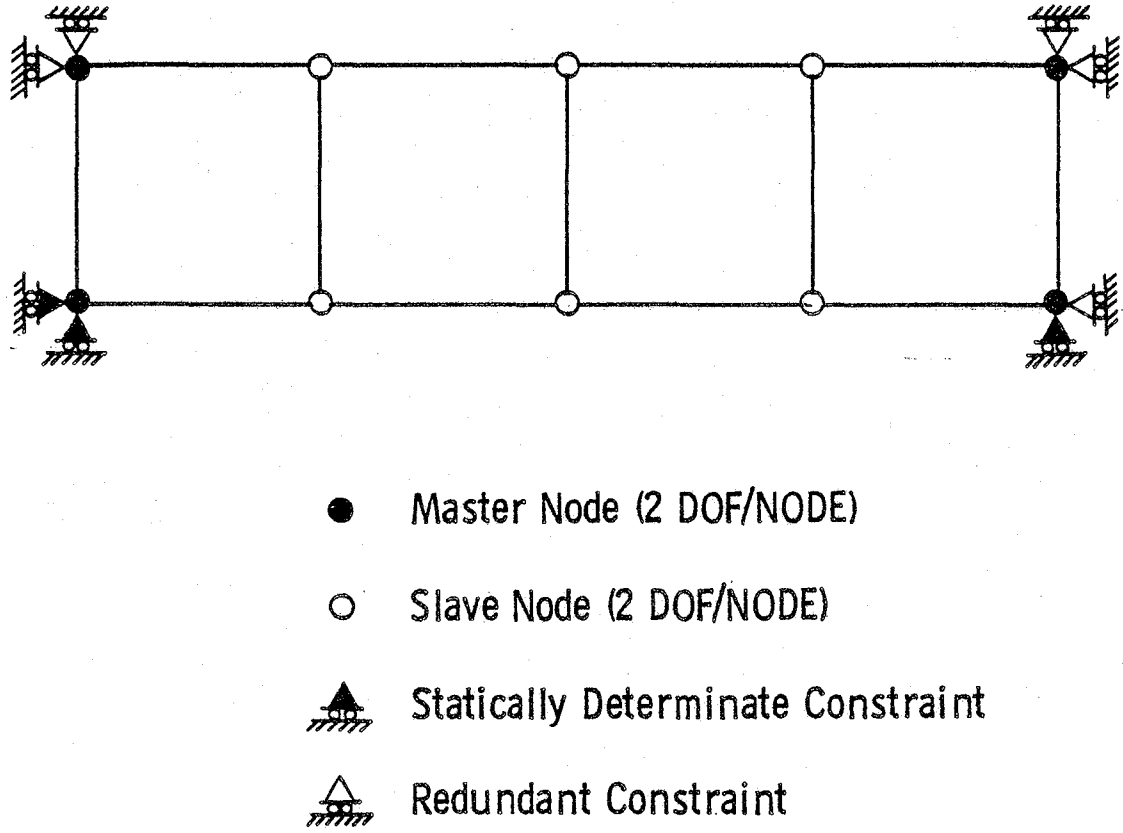


Figure 3.2 -- Statically Determinate and Redundant Constraints on a Simple Two Dimensional Substructure

$$\left[\begin{array}{c|c} K^{mm} & K^{ms} \\ \hline K^{sm} & K^{ss} \end{array}\right] \left\{\begin{array}{c} u^m \\ \hline u^s \end{array}\right\} - \omega_i^2 \left[\begin{array}{c|c} M^{mm} & M^{ms} \\ \hline M^{sm} & M^{ss} \end{array}\right] \left\{\begin{array}{c} u^m \\ \hline u^s \end{array}\right\} = \{0\}. \tag{3.18}$$

As in Guyan reduction, a static transformation from the master to the slave DOF can be written

$$\{u^s\} = [\phi^c]\{u^m\}. \tag{3.19}$$

As with simple Guyan reduction, the set of master DOF in modal synthesis methods is not limited to those DOF on internal boundaries. DOF in the interior of the substructure may be retained as well and may possibly improve the solution. Inspection of Eq. (3.2-3.4) reveals that $[\phi^c]$ is identical to $[T_G]$, the static constraint modes. For consistency in the following development, $[\phi^c]$ is used to represent the static constraint modes.

If the set of master DOF is restrained from displacement, Eq. (3.18) reduces to

$$[K^{ss}]\{u^s\} - \bar{\omega}_i^2[M^{ss}]\{u^s\} = \{0\}. \tag{3.20}$$

The solution of this eigenvalue problem yields the matrix of fixed-fixed normal modes of vibration, $[\phi^n]$, having the same order as $[K^{ss}]$ and $[M^{ss}]$. The computed vibration frequencies, $\bar{\omega}_i$, are those of the isolated substructure.

To reduce the substructure mass and stiffness matrices, a transfor-

mation to generalized coordinates, $\{q\}$, is defined as

$$\{u\} = \left\{\begin{array}{c} u^m \\ \hline u^s \end{array}\right\} = [T_F]\{q\} = [T_F]\left\{\begin{array}{c} q^m \\ \hline q^n \end{array}\right\} \quad . \tag{3.21}$$

The fixed-interface transformation, $[T_F]$, is derived from the static constraint and the normal modes as

$$[T_F] = \left[\begin{array}{c|c} I & 0 \\ \hline \phi^c & \bar{\phi}^n \end{array}\right] \quad , \tag{3.22}$$

in which $[\bar{\phi}^n]$ is a rectangular matrix of retained modes from $[\phi^n]$. In general, the modes corresponding to the lowest natural frequencies, $\bar{\omega}_i$, are retained in $[\bar{\phi}^n]$. The slave displacements, $\{u^s\}$, are now dependent on both the static constraint modes and the retained normal modes of the isolated substructure.

Two observations regarding Eq. (3.21) are noteworthy. First, the generalized coordinate subvector, $\{q^m\}$, corresponds precisely to the master set of geometric coordinates, $\{u^m\}$. This proves useful in guaranteeing geometric compatibility with adjacent substructures. Secondly, as the number of mode shapes in $[\bar{\phi}^n]$ is reduced, the transformation shrinks to just the static constraint modes and thus, the fixed-interface method degenerates to Guyan reduction.

The reduced stiffness and mass matrices in generalized coordinates are obtained by maintaining equivalence of kinetic and potential ener-

gies between the two coordinate systems. The resulting forms are

$$[K_F] = [T_F]^T[K][T_F] = \left[\begin{array}{c|c} K_G & 0 \\ \hline 0 & \bar{\omega}_n^2 \end{array}\right] \quad \text{, and} \tag{3.23}$$

$$[M_F] = [T_F]^T[M][T_F] = \left[\begin{array}{c|c} [M_G] & [M^{ms}][\bar{\phi}^n]^T + \\ & [\phi^c]^T[M^{ss}][\bar{\phi}^n] \\ \hline [\bar{\phi}^n]^T[M^{ss}][\phi^c] & \\ + [\bar{\phi}^n]^T[M^{sm}] & [I] \end{array}\right] . \tag{3.24}$$

For a lumped mass formulation of the substructure

$$[M_F] = \left[\begin{array}{c|c} [M_G] & [\phi^c]^T[M^{ss}][\bar{\phi}^n] \\ \hline [\bar{\phi}^n]^T[M^{ss}][\phi^c] & [I] \end{array}\right] . \tag{3.25}$$

$[K_G]$ and $[M_G]$ are the Guyan reduced stiffness and mass matrices. The identity submatrix in $[M_F]$ and the submatrix $[\bar{\omega}_n^2]$ in $[K_F]$ result from the orthonormality of $[\bar{\phi}^n]$. $[\bar{\omega}_n^2]$ is a diagonal matrix of natural frequencies corresponding to the modes retained in $[\bar{\phi}^n]$.

Reduction of substructure loads, $\{P\}$, follows the same virtual work argument used in Guyan reduction. The resulting generalized force vector, $\{F\}$, is

$$\{F\} = [T_F]^T\{P\}. \tag{3.26}$$

Before proceeding, a simplified notation is introduced. Let

$$[K_F] = \left[\begin{array}{c|c} \bar{K}^{mm} & 0 \\ \hline 0 & \bar{K}^{nn} \end{array}\right] \quad \text{and} \quad [M_F] = \left[\begin{array}{c|c} \bar{M}^{mm} & \bar{M}^{mn} \\ \hline \bar{M}^{nm} & I \end{array}\right] , \tag{3.27a,b}$$

with the relationships to Eq. (3.23) and Eq. (3.24) established by inspection.

Although assembly of the reduced substructure stiffness and mass matrices is routine, an illustration of their form is useful. For an assembly of "r" substructures with the master DOF entered last

$$[\overset{*}{K}] = \begin{bmatrix} \bar{K}_1^{nn} & 0 & \cdots & 0 & 0 \\ 0 & \bar{K}_2^{nn} & & & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & \bar{K}_r^{nn} & 0 \\ 0 & 0 & \cdots & & \overset{*}{K}^{mm} \end{bmatrix} ; \quad [\overset{*}{M}] = \begin{bmatrix} I & 0 & \cdots & 0 & \bar{M}_1^{nm} \\ 0 & I & & 0 & \bar{M}_2^{nm} \\ \vdots & & \ddots & & \\ 0 & 0 & & I & \bar{M}_r^{nm} \\ \bar{M}_1^{mn} & \bar{M}_2^{mn} & \cdots & \bar{M}_r^{mn} & \overset{*}{M}^{mm} \end{bmatrix} . \tag{3.28a}$$

(3.28b)

Since the master DOF do not participate in the normal modes, no coupling between substructures exists outside the submatrices $[\overset{*}{K}^{mm}]$ and $[\overset{*}{M}^{mm}]$, which are the assembled Guyan stiffness and mass. From a data storage and computational viewpoint, these forms for $[\overset{*}{K}]$ and $[\overset{*}{M}]$ are ideally suited to hypermatrix methods.

The synthesis process for one level of substructuring is now complete. If $[\overset{*}{K}]$ and $[\overset{*}{M}]$ are the stiffness and mass matrices for the highest level structure, the differential equation of motion can be written and solution for displacements can proceed. If the highest level structure has not yet been reached, $[\overset{*}{K}]$ and $[\overset{*}{M}]$ are reduced just like any other substructure.

In summary, the fixed-interface method employs static constraint modes and a truncated set of fixed-fixed normal modes to achieve a reduction in the number of independent substructure DOF. Geometric coordinates at internal boundaries are retained in the set of generalized coordinates to assure displacement compatibility between substructures during assembly. This mixture of geometric and generalized coordinates in the substructure equations has resulted in no reports of numerical difficulties and none are expected.

3.3.3 Free-Interface Method with Interface Loading

In the free-interface method of modal synthesis [3.5, 3.10], the transformation to a reduced set of generalized coordinates relies on a truncated set of free-free normal modes. These mode shapes are computed for the isolated substructure with geometric constraints applied only as they occur in the actual structure. Internal boundary (master) DOF are not arbitrarily identified and fixed. This approach allows rigid-body modes to appear in the set of substructure normal modes unless sufficient physical constraints actually exist. The transformation to generalized coordinates also neglects the static constraint modes common to both Guyan reduction and the fixed-interface method. The transformation is simply

$$\{u\} = [\bar{\phi}^n]\{q\} \tag{3.29}$$

where $[\bar{\phi}^n]$ contains the truncated set of free-free normal modes corresponding to the lowest substructure natural frequencies.

The development by Goldman [3.5] includes a distinction between rigid-body modes and free-free elastic modes which is not required in Hou's method [3.10]. These two authors use similar approachs to substructure assembly. Geometric compatibility between substructures is enforced by writing equations of constraint in modal coordinates for internal boundary DOF. These constraint equations are used to combine the generalized displacement vectors for all isolated substructures into a single generalized displacement vector for the final structure. The corresponding reduced stiffness and mass matrices are thus generated.

A significant difficulty arises with the extension of the free-interface method to multilevel substructuring. In achieving one level of substructuring, all geometric coordinates are transformed to generalized coordinates. Further substructuring is complicated by the absence of geometric coordinates which are useful in assuring continuity of displacements at substructure internal boundaries. Beyond the lowest level, it will be necessary to modify the process of developing constraint equations to link together the substructures. Rather than defining the constraints in terms of the generalized coordinates at the level being assembled, they must be written in terms of the geometric coordinates at the lowest level. Such a task has yet to be investigated and the requirements for its implementation remain unknown.

The mode shapes of a substructure with free (or fixed) boundaries are not totally representative of the substructure's response in the assembled structure. This follows because the stiffness and inertia effects of the adjoining substructures have not been included in computing the dynamic modes of the isolated substructure. Interface loading [3.2,

3.9] is a technique which incorporates some of these effects in an attempt to make the isolated substructure modes more like the modes for the entire structural system when the free-interface method is used. The approach is to modify the stiffness and mass of each substructure prior to extracting its free-free normal modes.

Let the DOF of the substructure under consideration reside in set A and let all remaining DOF in the structure reside in set B. In general, set B may contain more than one substructure. Upon assembly, set A will intersect set B over the subset AB. Consider subset AB as the master DOF for both sets A and B. Figure 3.3 illustrates this approach as applied to the shallow shell example of Chapter 2. The substructure under consideration is the linear zone and it forms set A. The remainder of the structure, the nonlinear zone, falls in set B. Subset AB is the interface between the two sets.

By the same transformation as used in Guyan reduction, the displacements of DOF in set B can be written in terms of those in subset AB. Recognizing that subset AB also defines the master DOF of set A, the displacements of DOF in set B are effectively expressed in terms of the master DOF of set A. From these relationships, the interface loaded stiffness and mass for set A are derived. The effect of the interface loading is to add the Guyan reduced stiffness and mass of set B to the internal boundary terms (master DOF) of A's stiffness and mass.

The free-free normal modes of the substructure in set A are computed using the modified stiffness and mass matrices. Each substructure that will be reduced is identified as set A with the remaining substructures lumped into set B. A new interface loading effect is computed and

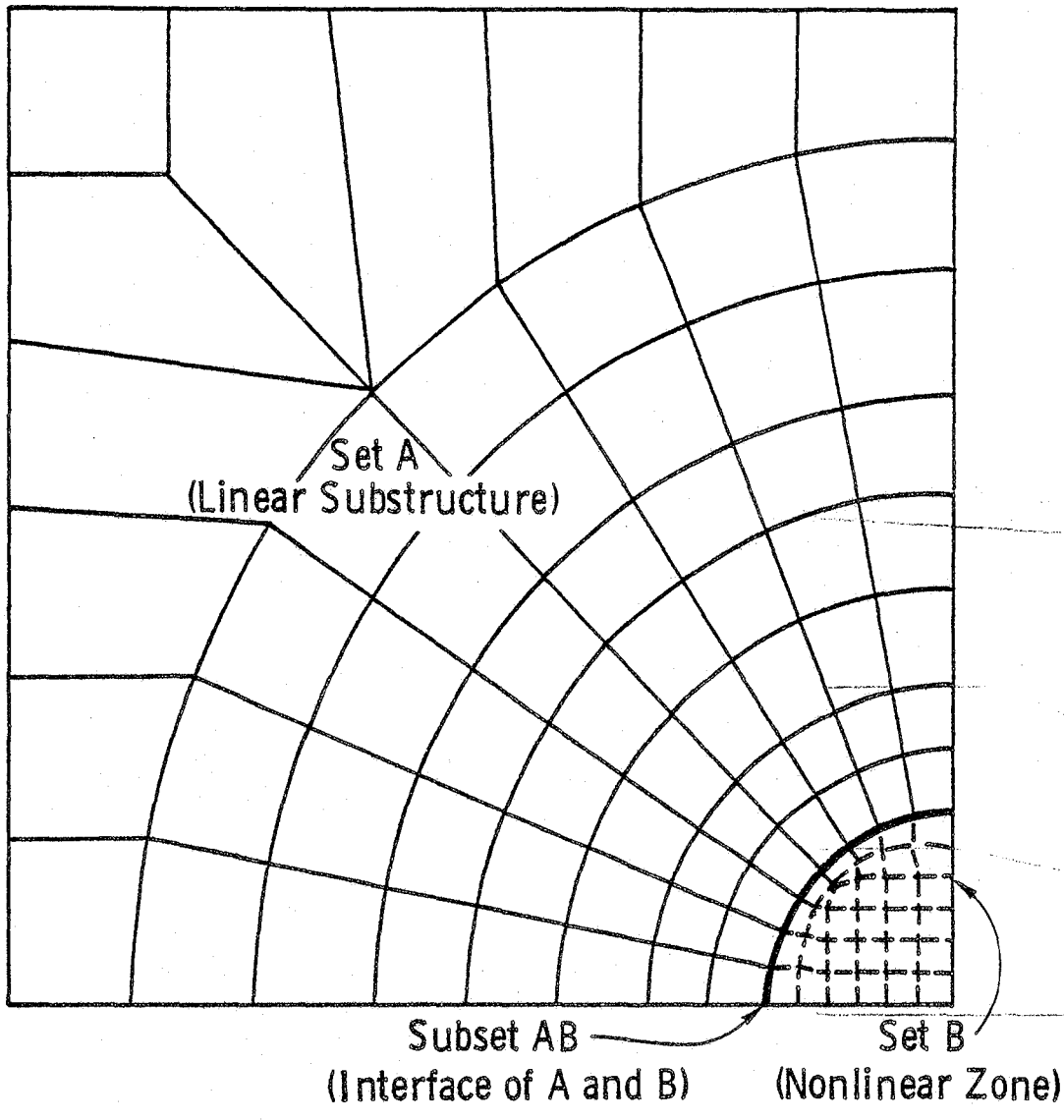


Figure 3.3 -- DOF Sets for Interface Loading of the Linear Substructure from the Chapter 2 Shell Example

the free-free analysis performed. Actual geometric constraints are applied as they occur in either set. After solution for the normal modes of each substructure is complete, the substructures are assembled using the truncated mode sets and the original, unmodified substructure stiffness and mass matrices. The interface loaded stiffness and mass are not used in the assembly process. Their use is limited to computing free-free normal modes for the substructures.

3.3.4 Branch Mode Analysis

Branch mode analysis [3.4, 3.9] is a hybrid method of modal synthesis which incorporates features of both the fixed-interface and the free-interface methods. The following procedure is one of several that fall under the class of branch mode analysis [3.13].

In branch mode analysis, one substructure is selected as the main body and all adjacient substructures are designated as branches. The analysis starts with a determination of the free-free component modes of the main body. Interface loading may be employed prior to computing the free-free modes. The topology of the main body is then expanded to include all substructures on its boundaries. This is done by writing the displacements for the main body and its adjacent branches in terms of the free-free modes of the main body and the fixed-fixed normal modes from each branch. Using the transformation to modal coordinates in the structure kinetic and potential energy equations, the reduced branch stiffness and mass matrices are computed.

When substructures are remote from the initially chosen main body, a number of solution schemes may be used. One approach is to identify one main body and its branches and perform the foregoing reduction. This reduced substructure group is then treated as a branch to a newly selected main body. Two more eigenproblems are solved and the process is repeated until each substructure has been joined.

3.3.5 Dynamic Stiffness Matrix

An iterative method for obtaining an exact solution to the frequency dependent dynamic reduction problem has been developed by Leung [3.17]. The method is closely related to the fixed-interface method but includes the frequency dependence in the transformation to the reduced set of coordinates. The first step of the method is to partition the stiffness and mass of each substructure and compute the Guyan reduced matrices $[K_G]$ and $[M_G]$. With the internal boundary DOF identified as master DOF, the fixed-fixed normal modes, $\{\phi^n\}$, and associated frequencies, $\bar{\omega}_i^2$, are computed for each substructure. The exact substructure dynamic stiffness matrix, $[D]$, can then be derived as

$$[D] = [K_G] - \Omega^2[M_G] - \Omega^2[G][\lambda][G]^T \tag{3.30}$$

where Ω^2 = an unknown system frequency

$[G] = [M^{ms}][\phi^n] - [K^{ms}][\phi^n][\bar{\omega}_i^2]^{-1}$

$[\bar{\omega}_i^2]$ = a diagonal matrix of substructure frequencies

$[\lambda]$ = a diagonal matrix of elements $\dfrac{\Omega^2}{\bar{\omega}^2 - \Omega^2}$

With no effect on the order of $[D]$, all normal modes in $[\phi^n]$ may be used or the set may be truncated to include only those associated with the

lowest frequencies.

Using a theorem stating that the dynamic mass matrix, $[M(\bar{\omega})]$, equals the partial derivative of $[D]$ with respect to $\bar{\omega}^2$, the substructure reduced mass is given by

$$[M(\bar{\omega})] = [M_G] + [G][Q][G]^T \tag{3.31}$$

where $[Q]$ = a diagonal matrix of elements $1 - \dfrac{\bar{\omega}_i^4}{(\bar{\omega}_i^2 - \Omega^2)^2}$.

The matrices $[K_G]$ and $[M(\bar{\omega})]$ for each substructure at the same level are assembled by the method used in Guyan reduction to obtain synthesized stiffness and mass matrices for the next higher level. The process is then repeated until the highest level structure is reached.

The solution process for this formulation requires iteration at each level of substructures. The unknown system frequency, Ω, must be initially estimated and then iteratively improved until convergence is attained. One iteration involves building $[M(\Omega)]$ and $[D]$ for each substructure until the highest level structure is reached. Then the eigenproblem at the highest level is solved,

$$[D]\{\psi\} = \{0\}. \tag{3.32}$$

From this solution, the next value of Ω is chosen for use in the following iteration. Although the fixed-fixed normal modes need be computed only once for each substructure, each iteration requires that the mass matrix for the intermediate level substructures be recondensed.

It is stated that a partial vibration case results when $\Omega = \bar{\omega}_i$ for a substructure and thus drives the dynamic mass to infinity. However, there are no clear guidelines given for handling this potential instability in the method.

The method is shown to yield good accuracy in computing system frequencies but results for mode shapes are not given. Also, no comparison of computational efficiency is made with the more standard methods of modal synthesis.

3.3.6 Attachment Modes and Interface Mode Sets

Bamford, Wada, Garba, and Chisholm [3.1] introduced the concept of attachment modes as an additional set of static modes used in modal synthesis. An attachment mode defines the response of the substructure to a unit force applied at an internal boundary DOF while all other internal boundary DOF remain free. The motivation for using attachment modes is that their use can be expected to reduce the number of normal modes necessary to accurately describe the displacement behavior of the substructure.

In the transformation from generalized to geometric coordinates for the substructure, attachment modes are combined with static constraint and normal modes. The normal modes may be computed with internal boundary DOF fixed, free, or a combination of the two. Bamford's transformation matrix includes a set of rigid-body modes, however it is noted that no distinction between rigid-body modes and static constraint modes is required. A drawback in the use of attachment modes is that they are not always linearly independent of the static constraint and normal

modes.

In an effort to establish a more rational approach to substructure synthesis, Hintz [3.8] grouped combinations of the four mode classes: rigid-body, static constraint, normal, and attachment, into five different interface mode sets. The four mode classes are illustrated in Figure 3.4 as they apply to the FIN substructure of Chapter 2. Each of the five sets is claimed to be complete in that it precisely represents the original finite element model for static and dynamic response. Implications of truncating a selected interface mode set are considered and thus guidelines are developed for retaining accuracy with a reduced size model. Since each of the five mode sets represents the substructure differently, truncation of each set has varying impacts on the substructure model. The guidelines are directed toward establishing a mode set that will allow maximum truncation of the normal modes while retaining the detail of the original finite element model with respect to statically imposed interface forces and displacements.

In numerical results presented by Hintz, the fixed-interface method (defined by one of the five interface mode sets) gave good results over a broader range of frequencies than did those methods using attachment modes. However, use of attachment modes did produce more accurate vibration frequencies in the low frequency range.

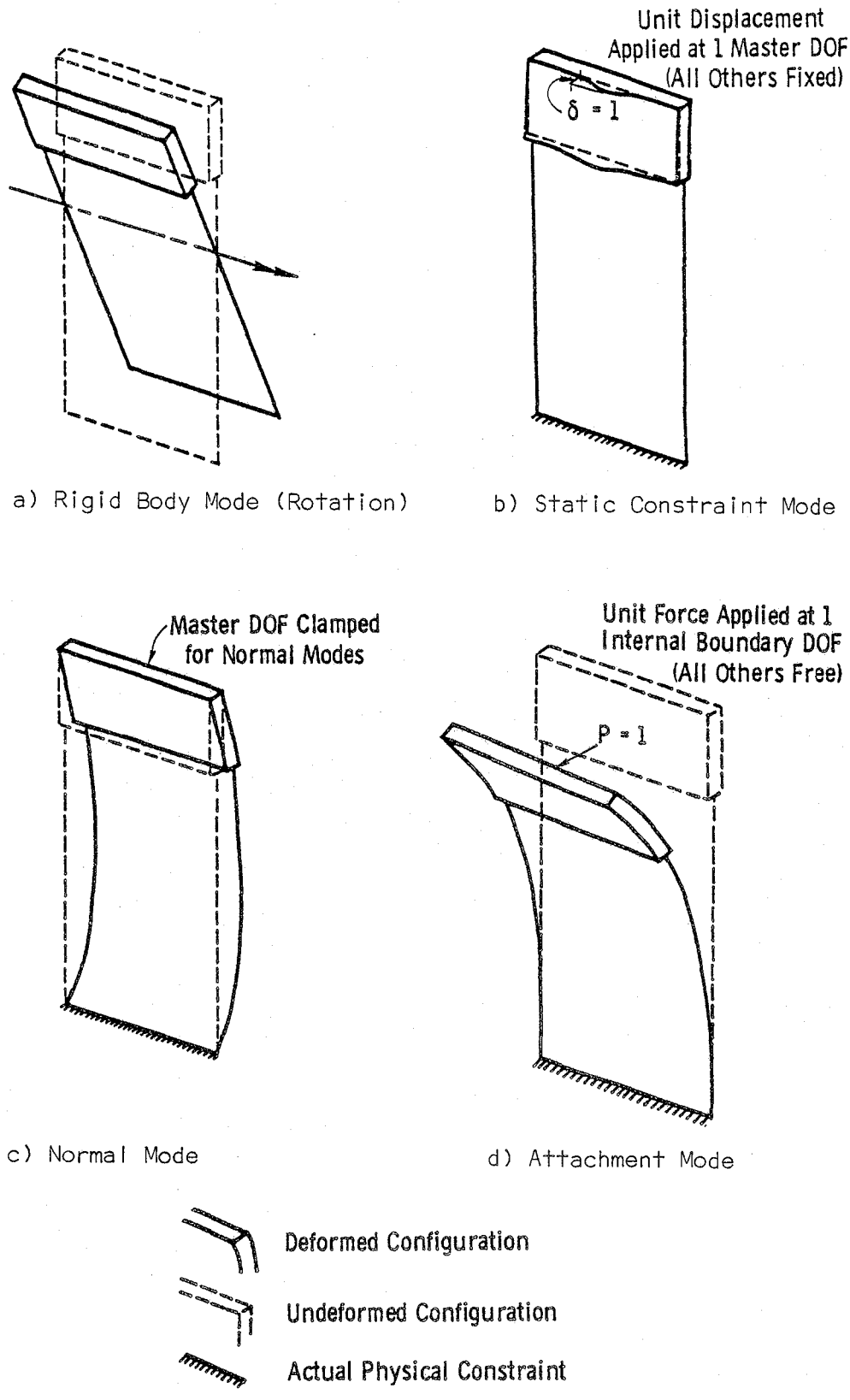


Figure 3.4 -- Mode Classes for Interface Mode Sets Illustrated for Substructure FIN

3.3.7 Improved Displacement Recovery in Modal Synthesis

The frequency dependent transformation previously discussed with regard to Guyan reduction, Eq. (3.14), has been applied in a limited fashion to modal synthesis by Kuhar and Stahle [3.16]. The method consists of reducing the order of the eigenvalue problem for the highest level structure after modal synthesis has been performed. If Ω is a frequency about which the eigenproblem for the highest level is to be reduced, the synthesized matrices can be reduced with the transformation

$$[T] = -([K^{ss}] - \Omega^2[M^{ss}])^{-1}([K^{sm}] - \Omega^2[M^{sm}]). \qquad (3.33)$$

The reduced matrices form an eigenvalue problem of smaller size than that defined by the synthesized matrices.

As with Guyan reduction, the eigenproblem is evaluated more accurately at the frequencies near the reduction value, Ω. (In Guyan reduction, the reduction frequency is $\Omega = 0$.) Thus, the choice of master DOF for the highest level structure should be directed towards preserving the frequency content of the model around Ω. Although selection of the master DOF is initially a matter of judgement, it can be improved in the second and later iterations if necessary. If the frequency range of interest is broad, it may be advantageous to re-solve the problem for different values of Ω.

Upon solution of the reduced eigenproblem at the highest level, the set of mode shapes, $[\phi^m]$, will be in terms of the master DOF. If a computed frequency ω_i is equal to the reduction frequency Ω, then the cor-

responding mode shape is obtained exactly by

$$\{\phi_i\} = \begin{bmatrix} I \\ \hline T \end{bmatrix} \{\phi_i^m\}. \tag{3.34}$$

Likewise, the full set of mode shapes is obtained from

$$[\phi] = \begin{bmatrix} I \\ \hline T \end{bmatrix} [\phi^m]. \tag{3.35}$$

but the vectors in $[\phi]$ are only approximate. Based on the coefficients in the mode shape vectors corresponding to frequencies within the range of interest, new master DOF can be identified to improve the accuracy of the reduction.

3.3.8 Residual Mass and Residual Flexibility

A hybrid method of substructure reduction with the capability of modeling a more general condition of internal boundary constraint was developed by MacNeal [3.18] and implemented to a limited extent in NASTRAN. The normal modes of the substructure are computed with internal boundary DOF free, fixed, or a combination of the two. The decision regarding boundary DOF fixity is based on the nature of the adjacent substructures in the model. If an interface DOF in one substructure is held fixed, the corresponding DOF in the adjacent substructure must remain free in order to avoid an overconstraint problem at the interface. This procedure requires that information on the topology of the substructures be provided by the analyst prior to modal analysis of any

individual substructure. MacNeal's formulation also requires that the mass at fixed boundary DOF be distributed to free DOF in the substructure since fixed DOF do not participate in the vibration response.

The representation of the substructure can be improved by including static approximations to the effects of the truncated higher modes. The improvement is in the form of a residual mass matrix for the synthesis involving fixed-fixed normal modes and a residual flexibility matrix when free-free normal modes are used. For the case of hybrid modes (some internal boundary DOF fixed, others free), both matrices are formulated. Since these matrices are obtained by a static derivation ($\omega = 0$), they are valid only at vibration frequencies that are low compared to the lowest mode of the substructure.

Rubin [3.21] proposed an improvement to MacNeal's residual flexibility for substructures with free-free normal modes only. The approach is shown to have better convergence than the methods of MacNeal and Hurty but like MacNeal's, its application is strictly limited to the low frequency range. Rubin's method is further restricted to the use of free-interface normal modes and rigid-body modes. Thus, MacNeal's residual mass is not included in the development.

3.3.9 Low-Order Polynomial Transformations

One of the primary differences among the reduction techniques discussed thus far lies in the definition of the transformation to generalized coordinates. Each method that has been reviewed uses some combination of static constraint and substructure normal modes. Meirovitch and Hale [3.19] have recognized that it is not necessary to

consider only substructure modes for use in the transformation. They have shown that it is sufficient for substructures to be represented by admissible functions that are from any complete set. One such set considered useful for continuous systems is a set of low-order polynomials. As applied to finite element models, the polynomials are defined over the domain of the substructure and satisfy the geometric boundary conditions. The shape vectors are then built by sampling the polynomials at the spatial coordinates of each substructure DOF.

The transformation to generalized coordinates is

$$\{u\} = [T_p]\{q\}, \tag{3.36}$$

where $[T_p]$ is the set of shape vectors derived from the low-order polynomials. As with the free-interface method, the generalized coordinates, $\{q\}$, contain no geometric equivalents for substructure assembly. Compatibility between substructures is enforced by a weighted residual method using spatial Dirac delta functions for continuous models or equivalently, unit vectors for discrete models. For a finite element model, the effect is achieved by writing equations of constraint matching the displacement at the point shared by two substructures. Using the constraint equations, a constraint matrix is produced which is used to link together the otherwise isolated substructures. From this operation is obtained the synthesized stiffness and mass matrices in generalized coordinates.

A difficulty lies in development of the constraint matrix. In the elementary example problems presented, the matrix was built by inspection. This is not an acceptable procedure for general application. As observed by the authors, an alternative formulation can be defined which relieves the problem of constructing the constraint matrix. The approach is simply to use the same transformation matrix as used in the fixed-interface method but with fixed-fixed normal modes replaced by low-order polynomial shapes. Thus

$$[T_p] = \begin{bmatrix} I & 0 \\ \hline \phi^c & \phi^p \end{bmatrix}, \tag{3.37}$$

where $[\phi^p]$ is the set of low-order polynomials. For this approach, polynomials which have zero values at the internal boundary nodes must be chosen to simulate the fixed-fixed boundary conditions. Geometric compatibility is enforced by the presence of the static constraint modes $[\phi^c]$ and retained internal boundary nodes in geometric coordinates.

There are two trade-offs that must be made in using the transformation of Eq. (3.37) instead of that in Eq. (3.36). The static constraint modes must be evaluated causing some additional computational expense. Also, the size of the set of admissible polynomial functions is reduced by the requirement that the functions be zero-valued at substructure internal boundaries.

In contrast to individual finite elements, the geometry of substructures is not of a nature that can be easily classified. Thus, selection of the appropriate polynomials is a matter of experience and judgement. Because no automated selection method has yet been devised,

this approach to modal synthesis is currently not capable of performing dynamic reduction in a general finite element system. However, the procedure remains one of the more actively pursued techniques and does hold potential.

3.3.10 Evaluation of Modal Synthesis Techniques

In the study of modal synthesis techniques, several advantageous characteristics of an ideal method can be identified. A discussion of these characteristics as evaluation criteria will be useful in comparing the various methods presented.

1. Efficiency of the Reduction Method

   The efficiency of a dynamic reduction method is influenced by a number of factors. First, the method must result in an accurate reduction of the substructure stiffness and mass. An efficient method will yield synthesized stiffness and mass matrices that accurately maintain the dynamic characteristics of the substructure with the minimum number of DOF. Second, the degree of analyst participation should be limited to simply the definition of the model and specification of the solution type. A method should be automatic once the solution process begins, hence eliminating the need for the analyst to interpret intermediate results and restart the process. This is not to imply that the analyst should surrender control of the solution process. Instead, he should be relieved of the burdensome task of supervising the computational process. Third, the synthesis method should be efficient in its use of the computer. Given the problem size, algorithms should be chosen that minimize the required computer resources, including CPU time and I/O. The number of arithmetic operations performed should be predictable rather than dependent upon an arbitrary test for convergence of an iterative process.

2. Applicability to General Problems

   A wide variety of dynamics problems exists for which modal synthesis is needed to achieve an economical and accurate solution. A synthesis method used in a general purpose FEM system should be capable of representing substructures over a broad range of geometries and with various types of boundary constraint. Also helpful would be the ability to incorporate experimental data, such as mode shapes and natural

frequencies, into the substructure model.

By necessity, finite element analyses of nonlinear structures are performed incrementally. As the effects of nonlinear materials and geometry occur, the model must be reformulated to accurately track the true response of the structure. Therefore, dynamic reduction methods must lend themselves to this incremental solution process.

3. Independence of Individual Substructures

Often times the analysis and design responsibilities of the various components of a structure are distributed among different organizational groups. This separation of responsibilities has many advantages and should not be encumbered by the synthesis method. Therefore, the method used should consider each substructure as an isolated entity in evaluating its dynamic response prior to system synthesis. Topology of the substructures should not be considered until the equations at the next higher level are ready for assembly.

4. Ease of Reanalysis

The most reliable test for convergence of a dynamic reduction method is to re-solve the problem with a more highly refined model (more independent DOF). The addition of more DOF to the model can be a relatively simple task, achieved at little expense, or it could be as difficult and expensive as a complete reanalysis of each substructure. The ideal approach allows simply the addition of previously neglected terms to improve the accuracy of the reduction. These terms generally take the form of truncated substructure normal modes.

5. Accuracy and Stability

Accuracy of results is important in two respects. Well defined modal response data is needed to accurately synthesize the higher level structures for frequency and transient analysis. Also, in returning to the lower levels for recovery of strains and stresses, the quality of the displacement vectors is critical. Accurate stresses require that displacement gradients be well formed. Intimately tied to accuracy of results is the numerical precision with which computations must be performed. Operations such as triangulation in equation solving can have a significant impact on final accuracy and the need for such operations should be considered in selecting the reduction method.

The potential for problems with the stability of operations in the reduction methods can often be identified by close examination of the development of the methods. Typical problem areas are linear dependence of the vectors comprising a transformation matrix and the divide-by-zero singularity.

With the criteria established, each synthesis method can be evaluated. The goal of this evaluation is to isolate one or two methods of modal synthesis that will be most useful in the general purpose FEM system.

The fixed-interface method successfully satisfies four of the above five criteria. The method is simple to apply and results in a significant size reduction of properly substructured models. It does not require consideration of substructure topology prior to assembly thus preserving substructure independence. Repetitive tests for convergence are straightforward, requiring simply the addition of more normal modes in selected substructures. The entire problem need not be re-solved to incorporate the additional modes. Thus the cost of convergence tests is low and predictable. Orthogonality of the transformation matrices ensures stability of the method and accuracy has proven favorable for many problems. One drawback of the method is its limited capability to use experimental data. Since all master DOF are fixed during the computation of normal modes, the structural components must be tested in a like fashion.

Interface loading has proven necessary to the use of the free-interface method. The approximate inertial effects of adjacent substructures serve to improve the displacement gradients at the internal boundaries, thus yielding more accurate strains and stresses [3.14]. Although interface loading is helpful in computing accurate stresses, its use eliminates the ability to maintain substructure independence. As previously mentioned, extension of the free-interface method to multilevel substructuring might be a difficult task to accomplish.

Branch mode analysis was originally developed for chain-type structures, e.g. piping networks. As such, its application to analysis of more general structures is cumbersome. The method requires solution of twice as many eigenproblems as the fixed-interface method and it is intimately dependent upon substructure topology. Convergence testing using a more refined model becomes impractical as complete reanalysis is necessary.

Leung's iterative method appears limited to eigenvalue extraction over a narrow band of frequencies or analysis of transient response to harmonic forcing functions. This is because the system equations must be synthesized for each modal frequency considered. For multilevel substructured models, the need for repetitive condensations is expected to make the method prohibitively expensive when the frequency range of interest is broad. As mentioned in the development of the method, some special precautions would also be necessary to maintain stability when the system frequency is very close or equal to a substructure modal frequency.

Although attachment modes are useful in reducing the number of retained substructure normal modes, their potential failings are not well illustrated in the literature. It is noted that attachment modes may cause ill-conditioning in the transformation to generalized coordinates [3.1]. What must be recognized is the difficulty with which linearly independent attachment modes are chosen. The selection of attachment DOF for a substructure is by no means an intuitive process. It is expected that even the experienced analyst will require a trial-and-error approach to their selection.

Efficient methods for extracting a limited number of eigenpairs in large systems effectively make the highest level structure reduction proposed by Kuhar and Stahle unnecessary. Their process yields a reduced size equation of motion via an iterative procedure. The same results can easily be achieved by solving for selected modes of the unreduced synthesized equations and then formulating a transformation with these modes and appropriate static modes. This transformation can then be used to reduce the size of the synthesized equations.

MacNeal's residual mass and flexibility method provides an automatic approach to modal synthesis which includes general boundary constraint capabilities. However, the method does have some limitations. It is correctly argued that fixed coordinates do not participate in the vibration behavior of the substructure. What is not considered is that in a multilevel substructured model, the coordinates held fixed at level "I+1" may be considered free at level "I". If the mass components of the constrained DOF are redistributed as suggested, the inertia characteristics of those DOF will not be accurately preserved. Also, the need to define substructure topology and properties prior to assigning internal boundary constraints hampers the effort to retain independence in computing individual substructure response. Another drawback is the potential ill-conditioning of the residual flexibility matrix when many modes are retained in the transformation. The advantages of using residual mass and flexibility matrices appear marginal in that they improve only the low frequency response and that residual mass is implicitly contained in Hurty's method [3.18].

Polynomial transformation techniques hold tremendous potential as dynamic reduction methods. A great savings in computational effort is possible by evaluating low-order polynomials rather than computing substructure vibration modes. Unfortunately though, the approach to selecting the polynomials has not yet been standardized. As such, the techniques are not immediately useful for general application in FEM software.

3.4 Selection of Methods for Dynamic Reduction

In view of the above discussion, selection of dynamic reduction schemes becomes simple. The two broad types of structural models, those with and without substructuring, exemplify the need for two distinct methods for dynamic reduction. The two methods considered most suitable are Guyan reduction and the fixed-interface method of modal synthesis. Perhaps the greatest advantage of both of these methods is their conceptual simplicity. This point cannot be overemphasized. It is imperative that the analyst have a complete understanding of the analysis methods he uses. Without this understanding the chances of achieving a meaningful dynamic analysis are remote. The obvious disadvantage of this philosophy is the sacrifice in sophistication that the experienced analyst makes if he is to use the general system. One might expect, however, that special purpose FEM systems could be used when the need for more refined techniques is not satisfied elsewhere.

3.5 References

3.1 Bamford, R., Wada, B.K., Garba, J.A., and Chisholm, J., "Dynamic Analysis of Large Structural Systems," Synthesis of Vibrating Systems (ASME Booklet Nov. 1971), pp. 57-71

3.2 Benfield, W.A. and Hruda, R.F., "Vibration Analysis of Structures by Component Mode Sunstitution," AIAA Journal, Vol. 9 pp. 1255-1261 (1971)

3.3 Craig, R.R. Jr. and Bampton, M.C.C., "Coupling of Substructures for Dynamic Analysis," AIAA Journal, Vol. 6 pp. 1313-1319 (1968)

3.4 Gladwell, G.M.L., "Branch Mode Analysis of Vibrating Systems," Journal of Sound and Vibration, Vol. 1 pp. 41-59 (1964)

3.5 Goldman, R.L., "Vibration Analysis by Dynamic Partitioning," AIAA Journal, Vol. 7 pp. 1152-1154 (1969)

3.6 Guyan, R.J., "Reduction of Stiffness and Mass Matrices," AIAA Journal, Vol. 3 pp. 380 (1965)

3.7 Henshell, R.D. and Ong, J.H., "Automatic Masters for Eigenvalue Economization," Earthquake Engineering and Structural Dynamics, Vol. 3 pp. 375-383 (1975)

3.8 Hintz, R.M., "Analytical Methods in Component Modal Synthesis," AIAA Journal, Vol. 13 pp. 1007-1016 (1975)

3.9 Holze, G.H. and Boresi, A.P., "Free Vibration Analysis Using Substructuring," Journal of the Structural Division (ASCE), Vol. 101 pp. 2627-2639 (1975)

3.10 Hou, S.N., "Review of Modal Synthesis Techniques and a New Approach," Shock and Vibration Bulletin, 40(4) pp. 25-39 (1969)

3.11 Hurty, W.C., "Vibrations of Structural Systems by Component Mode Synthesis," Journal of the Engineering Mechanics Division (ASCE), Vol. 86 pp. 51-69 (1960)

3.12 Hurty, W.C., "Dynamic Analysis of Structural Systems Using Component Modes," AIAA Journal, Vol. 3 pp. 678-685 (1965)

3.13 Hurty, W.C., "Introduction to Modal Synthesis Techniques," Synthesis of Vibrating Systems (ASME Booklet Nov 1971), pp. 1-13

3.14 Hurty, W.C., Collins, J.D., and Hart, G.C., "Dynamic Analysis of Large Structures by Modal Synthesis Techniques," Computers and Structures, Vol. 1 pp. 535-563 (1971)

3.15 Kidder, R.L., "Reduction of Structural Frequency Equations," AIAA Journal, Vol. 11 p. 892 (1973)

3.16 Kuhar, E.J. and Stahle, C.V., "Dynamic Transformation Method for Modal Synthesis," AIAA Journal, Vol. 12 pp. 672-678 (1974)

3.17 Leung, Y-T, "An Accurate Method of Dynamic Substructuring with Simplified Computation," International Journal for Numerical Methods in Engineering, Vol. 14 pp. 1241-1256 (1979)

3.18 MacNeal, R.H., "A Hybrid Method of Component Mode Synthesis," Computers and Structures, Vol. 1 pp. 581-601 (1971)

3.19 Meirovitch, L. and Hale, A.L., "On the Substructure Synthesis Method," AIAA Journal, Vol. 19 pp. 940-947 (1981)

3.20 Miller, C.A., "Dynamic Reduction of Structural Models," Journal of the Structural Division (ASCE), Vol. 106 pp. 2097-2108

3.21 Rubin, S., "Improved Component-Mode Representation for Structural Dynamic Analysis, AIAA Journal, Vol. 13 pp. 995-1006 (1975)

CHAPTER 4

COMPUTATIONAL ALGORITHMS FOR DYNAMIC ANALYSIS

4.1 General

The solution of structural dynamics problems by the FEM requires computational capabilities which are not necessary in static analysis. The two most important of these, in terms of computational efficiency, are eigenproblem solution and transient response analysis. These two operations form the core of the dynamic analysis process. Their proper implementation and use is essential to the success of the dynamic analysis. Other less important features include mass matrix and damping matrix formulation and the use of experimental data. Little computational effort is expended on these operations but their use adds generality and completeness to the solution strategy.

Eigenproblem solution is the single most expensive operation in the modal synthesis process. While the approach to modal synthesis will control the quality of the solution, eigenvalue analysis can be expected to control the cost. An understanding of the various methods for eigenvalue analysis is, therefore, necessary if the analyst is to achieve an economical and accurate solution.

Solution of the differential equations of motion yields the complete structural response to the transient loading. This solution can be obtained by either of two approaches: mode superposition for linear systems and time-history integration for both linear and nonlinear systems.

The remainder of this chapter is divided into three sections. The first presents a review of eigenproblem solution techniques. The methods are evaluated and a selection is made of those methods deemed necessary for incorporation in the general FEM system that employs modal synthesis for substructure reduction. The second section discusses the methods used to solve the differential equations of motion. Emphasis is placed on the effects that nonlinear structural response and multilevel substructured modeling have on the solution processes. The final section briefly discusses mass matrix formulation, damping, and the use of experimental data.

4.2 Eigenproblem Solution

4.2.1 Effects of Multilevel Substructuring

The introduction of multilevel substructuring into the FEM has added significantly to the requirements for a versatile eigenproblem solution package in the general FEM system. The importance of accuracy in computing substructure modes and the frequency with which the eigenproblem must be solved become major considerations in the selection of a solution process. Accurate synthesis of the highest level structure stiffness and mass is dependent upon the quality of the retained modes from each substructure. In a model with multilevel substructuring, the stiffness and mass matrices at the highest level are synthesized using vibration modes from the lower levels. As each higher level substructure is assembled from the lower levels, repeated vibration analysis of synthesized substructures is performed. This process will tend to compound any errors that may exist in the vibration modes. Eventual degradation of the highest level stiffness and mass can be avoided by maintaining good accuracy in the computed mode shapes at each level. In the modal synthesis process, a free vibration analysis is performed on each substructure. In nonlinear problems, the vibration analysis of nonlinear substructures is repeated with each update of system properties. For large substructures these requirements for frequent vibration analyses could become prohibitively expensive unless very efficient solution methods are used.

In the evaluation of eigenproblem solution methods, one must consider more than the computational efficiency with which a solution is achieved. Also of prime concern is the stability of the method for a particular problem, given the form of the matrices involved. If the solution method is inappropriate for the given problem, convergence to the wrong solution or divergence can occur. This stability is of particular interest for finite element models using multilevel substructuring. As the synthesis progresses upward through the various levels of substructuring, the nature of the eigenproblem changes as outlined below. Thus the analyst must have the capability to select the solution method that most closely matches the characteristics of the new problem.

The changes in the nature of the eigenproblem include variations in the characteristics of the substructure stiffness and mass matrices and in the number of eigenpairs needed. For lowest level substructures, the mass matrix may be diagonal or banded, positive definite or semi-definite, well formed or ill-conditioned. Although the stiffness matrix is normally banded, it may be singular, such as when rigid-body modes are included. Further complications arise when static condensation is used to eliminate massless degrees of freedom. This may cause a loss of bandform for both the stiffness and the mass. At the lowest substructure level, usually only the lowest frequencies and mode shapes are necessary to progress with the synthesis process. However, depending upon the eigenproblem solution method used, this requirement may involve finding all eigenpairs of a reduced system.

When a structure's stiffness and mass are assembled from lower level substructures, the resulting matrices can take on a special form. If Guyan reduction is used at the lowest level, the structure matrices are generally full or widely banded. If modal synthesis is used, the stiffness matrix will be in block diagonal form and the mass matrix will be block diagonal and border banded; see Eq. (3.25) and (3.26). It is most desirable to have eigenproblem solution methods that can deal effectively with these special forms when the total number of DOF is large.

No single eigenvalue solution method has been developed which is most efficient for all of the expected forms taken by the eigenproblem. Within a particular class of problems (e.g. all eigenpairs are required and the system matrices are narrowly banded) superior solution methods can be identified. An adequate selection of these methods must be available in a general FEM system, thus permitting the analyst to tailor the solution to the particular characteristics of his problem.

4.2.2 Solution Methods

For discussion purposes, eigenproblem solution methods are seperated into three categories. These are simultaneous iteration, transformation, and polynomial/vector iteration methods. Solution techniques within each of these categories rely on one or more of the fundamental properties of eigenpairs. Some of these properties are expressed in equation form as

1. $[K]\{\phi\} = \lambda[M]\{\phi\}$. (4.1)

   This form suggests an iterative solution by assuming a mode shape and successively improving the approximation until convergence to the true mode shape is attained.

2. $[\phi]^T[K][\phi] = [\lambda]$ and (4.2)

$[\phi]^T[M][\phi] = [I]$. (4.3)

This form suggests that, since the vectors in $[\phi]$ are unique within a scalar multiple of themselves, successive transformations of [K] and [M] to bring them to diagonal form will yield the matrices $[\phi]$ and $[\lambda]$.

3. $\det(K - \lambda M) = 0$. (4.4)

This form shows that the roots of the characteristic polynomial define the eigenvalues of the problem $[K]\{\phi\} = \lambda[M]\{\phi\}$.

4. $\lambda_1^{(m)} \leq \lambda_1^{(m+1)} \leq \lambda_2^{(m)} \leq \lambda_2^{(m+1)} \leq \ldots \leq \lambda_{n-m-1}^{(m+1)} \leq \lambda_{n-m}^{(m)}$ (4.5)

where $\lambda_i^{(m)}$ and $\lambda_i^{(m+1)}$ define the eigenvalues of the "m" th and the "m+i" th constraint problem for mode "i". This eigenvalue separation property states that the characteristic polynomials $p(\lambda^{(m)})$; m=1,2,...,(n-1) form a Sturm sequence.

Details regarding these properties may be found in the references [4.2, 4.25, & 4.27].

4.2.2.1 Simultaneous Iteration

Simultaneous iteration methods involve successive improvement of a set of orthogonal vectors chosen to approximate the mode shapes. Each of the methods uses one of the basic forms of inverse iteration:

$[\bar{x}_{j+1}] = [A][x_j]$, or (4.6)

$[K][\bar{x}_{j+1}] = [M][x_j]$. (4.7)

In these forms [K] is the (nxn) structure stiffness, [M] is the (nxn) structure mass matrix, $[x_j]$ is the (nxq) matrix of assumed mode shapes for iteration "j" and in symbolic form:

$[A] = [L]^{-1}[M][L]^{-T}$, where (4.8)

$[K] = [L][L]^T$. (4.9)

The differences in the methods lie in the processes used to orthogonalize $[\bar{x}_{j+1}]$ and to accelerate convergence.

Most of the simultaneous iteration methods reviewed use the standard form of the eigenvalue problem, Eq. (4.6). Jennings [4.12] presented the first such practical method in which the coupling among the vectors of $[x_j]$ is evaluated and reduced by an interaction analysis. The process involves examining the off-diagonal terms of the product

$$[B] = [x_j]^T[\bar{x}_{j+1}], \tag{4.10}$$

then decoupling the vectors in $[\bar{x}_{j+1}]$ and reorthogonalizing prior to the next iteration.

Rutishauser [4.23] introduced a procedure in which the matrix of modal vectors is improved by solving the reduced (qxq) eigenproblem using the matrix

$$[G] = [\bar{x}_{j+1}]^T[\bar{x}_{j+1}]. \tag{4.11}$$

No explicit orthogonalization is then required. The remaining variations of these methods, primarily attributable to Jennings and his colleagues [4.5, 4.7, & 4.13], represent modifications to enhance convergence or to improve the orthogonalization process.

Dong [4.8] and Bathe [4.2] presented independent developments of the method most commonly known as subspace iteration. This method uses the form of Eq. (4.7) and relies on the solution of a reduced eigenproblem similar to that of Rutishauser. This method has a significant advantage in that it does not require conversion of the eigenproblem to the standard form of Eq. (4.6). This feature affords the potential to

solve a wider range of problems ([K] and/or [M] ill-conditioned) than was previously feasible. Several methods have been proposed to accelerate the convergence of the subspace iteration process [4.1, & 4.28]. These include shifting, overrelaxation, and the use of Chebyshev polynomials. An improved method for selecting starting vectors is also available. Although these techniques were introduced for application to subspace iteration, they are applicable to other simultaneous iteration methods.

4.2.2.2 Transformation Methods

Transformation methods are based on the orthogonality property of eigenvectors. The approach is to generate a sequence of orthogonal transformation matrices, $[P_i]$, of order (nxn), that drive the symmetric matrix, [A], also of order (nxn), to diagonal form. For a case requiring "k" iterations to achieve convergence, the operation takes the form

$$[P_k]^T[P_{k-1}]^T \ldots [P_2]^T[P_1]^T[A][P_1][P_2]\ldots[P_k] = [\lambda]. \qquad (4.12)$$

The diagonal matrix $[\lambda]$ contains the complete set of eigenvalues. The corresponding eigenvectors are computed by

$$[\phi] = [P_1][P_2] \ldots [P_k] \, . \qquad (4.13)$$

The eigenpairs are not ordered as in the case of simultaneous iteration. Further, since the complete eigensystem is solved, the resulting diagonal matrix and the product in Eq. (4.13) are unique. This uniqueness exists only when the transformation matrices are of order (nxn).

The generalized Jacobi method is recognized to be the most efficient, truely iterative, transformation method for solving the general eigenproblem

$$[K]\{\phi\} = \lambda[M]\{\phi\}. \tag{4.14}$$

Each transformation matrix, $[P_i]$, performs a rotation which zeros corresponding off-diagonal terms in [K] and [M]. One sweep is completed when each off-diagonal term in [K] and [M] has been individually zeroed. Since zeroing one off-diagonal term tends to make a previously zeroed term nonzero, multiple sweeps are required to achieve convergence.

If the eigenproblem can be transformed to standard form:

$$[A]\{\phi\} = \lambda\{\phi\}, \tag{4.15}$$

where [A] is defined by Eq. (4.8), transformation of [A] to tridiagonal form can be performed without iteration. Givens introduced a method of plane rotations [4.25] in which, for step "j", zeros are introduced in row "j" and column "j" without destroying the zeros from previous steps. The eigenpairs of the resulting tridiagonal matrix can then be found with relative ease. Householder developed an improved approach to tridiagonalization using reflection matrices to perform the transformation. Like that of Givens, Householder's method requires (n-2) steps to complete the reduction but each step of Householder's method involves roughly half as many multiplications.

A third method for tridiagonalization is available and can be applied to the general eigenvalue problem of Eq. (4.14). The generalized Lanczos method [4.24] is applicable to problems when [K] and [M] are symmetric, equally banded, and at least one is positive definite. In the method, a single orthogonal transformation matrix is built one column at a time. With the evaluation of each new column, $\{r_i\}$, reorthogonalization to all previous columns, $\{r_1\}$ to $\{r_{i-1}\}$, is required to ensure stability.

Solution for the eigenpairs of a tridiagonal matrix is effectively attained by using QR iteration to find the eigenvalues and inverse vector iteration to find the eigenvectors. A discussion of inverse vector iteration follows in the next section.

QR iteration [4.2] is simply a factorization of the tridiagonal matrix, [S], into the product of an orthogonal matrix, [Q], and an upper triangular matrix, [R].

$$[S] = [Q][R] \tag{4.16}$$

Pre- and post-multiplying Eq. (4.16) by [Q] and [Q] respectively gives

$$[Q]^T[S][Q] = [R][Q] \tag{4.17}$$

where the product on the right-hand side is the diagonal matrix of eigenvalues. Jacobi rotation matrices are often used to reduce [S] to uppertriangular form of [R], with [Q] being the product of those rotation matrices. A shifting strategy, discussed below, can be used with QR iteration to accelerate convergence.

4.2.2.3 Polynomial/Vector Iteration

Polynomial iteration techniques involve an iterative solution of the characteristic polynomial

$$p(\lambda) = \det([K] - \lambda[M]) = 0. \tag{4.18}$$

In practice the polynomial is never explicitly evaluated. Instead an estimate of the exact eigenvalue is made and then the determinant is evaluated using Gauss or Choleski factorization. From this evaluation, the estimate of the eigenvalue is improved and the process is repeated.

Two popular procedures aid in this evaluation: accelerated secant iteration and bisection. Accelerated secant iteration is a superlinear interpolation method from which an improved estimate of the eigenvalue, λ_{k+1}, is determined using two previous estimates, λ_{k-1} and λ_k. The improved estimate is

$$\lambda_{k+1} = \lambda_k - \alpha\left(\frac{p(\lambda_k)(\lambda_k - \lambda_{k-1})}{p(\lambda_k) - p(\lambda_{k-1})}\right) \quad ; \tag{4.19}$$

where α is an acceleration constant. The polynomial is evaluated at λ_{k+1} and if required, another iteration is performed. Bisection is an iterative process in which the interval between the previously determined upper and lower bounds of λ_k is halved and the sign of the polynomial at that midpoint evaluated. On the basis of the evaluation, the new interval containing λ_k is identified and the process repeated until convergence is achieved.

Vector iteration is a special version of simultaneous iteration in which only one eigenpair is computed at a time. The iteration sequence takes the form:

$$[K]\{\bar{x}_{j+1}\} = [M]\{x_j\}. \tag{4.20}$$

If convergence of the eigenvector is gained in iteration "k", the eigenvalue can be computed from Rayleigh's quotient:

$$\lambda = \frac{\{x_k\}^T[K]\{x_k\}}{\{x_k\}^T[M]\{x_k\}} \quad . \tag{4.21}$$

The above form of vector iteration, termed inverse iteration, will initially converge to the most dominant eigenpair (smallest eigenvalue).

To force convergence to a subdominant eigenpair, the influence of the previously computed eigenvectors must be eliminated. If an estimate of the eigenvalue is known, shifting can be used to make the unknown eigenpair dominant. Letting μ be the value of the shift, a modified eigenproblem is defined

$$[K - \mu M]\{\phi\} = \nu[M]\{\phi\} \tag{4.22}$$

where $\nu = \lambda - \mu$. Inverse iteration applied to this problem produces the eigenpair closest to the shift μ. When a good estimate to the desired eigenvalue is unavailable, shifting is not effective. The iteration vector must be chosen to be orthogonal to the previously computed eigenvectors. When eigenvalues are multiple or clustered, it is necessary to reorthogonalize the iteration vector to avoid convergence to mode shapes which have already been computed.

Effective eigenproblem solution techniques using a combination of polynomial iteration and vector iteration have been developed for limited applications. In Gupta's method [4.10] the individual intervals containing each eigenvalue are defined using Sturm sequence properties and bisection. A shift is made to the middle of each interval and then the eigenpair is determined by inverse iteration and Rayleigh's quotient. The upper bound on the interval for λ_i is used as the lower bound for λ_{i+1}. Bathe and Wilson presented a method called determinant search [4.2] in which accelerated secant iteration is used to accurately determine the eigenvalue. Inverse iteration with shifts is then used to find the eigenvector. If the eigenvalue is accurately evaluated, a shift to that value will result in the eigenvector being found in no more than two vector iterations. An advantage of these combined techniques is that when eigenvalues are not multiple or clustered, each eigenpair is independently evaluated with no need for orthogonalization.

Robinson and Harris [4.22] introduced a Newton-Raphson iterative approach to vector iteration. For iteration "i", the process involves choosing a vector increment, $\{\Delta x_i\}$, and a frequency increment, $\Delta\lambda_i$, which will eliminate the residual vector

$$\{r_i\} = [K]\{x_i\} - \lambda_i[M]\{x_i\}, \tag{4.23}$$

and thus force convergence to a true eigenpair. As an additional side condition, the vector increment, $\{\Delta x_i\}$, must be orthogonal to the approximate eigenvector, $\{x_i\}$, with respect to $[M]$, i.e.

$$\{x_i\}^T[M]\{\Delta x_i\} = 0. \tag{4.24}$$

For [K] and [M] of order (nxn), a set of (n+1) simultaneous equations is then solved for $\{\Delta x_i\}$ and $\Delta\lambda_i$. The process is repeated until some convergence criterion is satisfied. Upon convergence, estimates of a different eigenpair are taken and the process is restarted. Initial estimates to the eigenpairs are taken from another source, such as subspace iteration.

The above method requires a complete triangulation of an (n+1) order set of linear equations for each iteration. Lee and Robinson [4.17] presented an alternative method which requires only one triangulation for each eigenpair. Although the convergence rate is less than for the Robinson-Harris method, the overall computational efficiency is improved. When eigenvalues are multiple or clustered, all eigenpairs in the group are found simultaneously by a method using Lagrange multipliers and the stationary property of the Rayleigh quotient. Illustrative examples show the Lee-Robinson method to require between 36% and 60% as many operations as needed by subspace iteration. Comparison of operation counts indicates that the method is also more efficient than determinant search.

4.2.3 Evaluation of Eigenproblem Solution Methods

As with the evaluation of modal synthesis, it is useful to establish criteria for the evaluation of eigenproblem solution methods. Although the categories listed below are similar to those previously discussed, they differ in the application to eigenproblems.

1. Computational Effort

The computational effort required by a solution algorithm is most effectively measured by the number of arithmetic operations performed. For eigenproblem solutions, this number is dependent upon the order and bandwidth of the matrices and the number of eigenpairs computed. For large problems, in-core solutions are not always possible. Therefore, efficiency is further measured by the capability to solve the eigen-problem with the use of out-of-core data storage. Controlling parameters are the number of data blocks required to simul-taneously reside in core and the frequency with which these blocks must be replaced.

2. Applicability to General Problems

Generality of eigenproblem solution methods implies the ability to solve problems of the form given by Eq. (4.14). Often the transformation to standard form, Eq. (4.15), is ineffective because of an ill-conditioned mass matrix. Such a situation is common with a lumped mass formulation in which rotational DOF are assigned zero inertia. This limitation is avoided when an eigenproblem solution method is available that can solve the general problem. Generality further applies to dealing with the various forms taken by the stiffness and mass matrices. As shown in Chapter 3, the dynamic reduction process can lead to coefficient matrices that are diagonal, banded, full, block-diagonal, or block-diagonal and border banded.

3. Ease of Reanalysis

The need for reanalysis of an eigenproblem arises in two instances. First, the analysis may need to be continued with a tighter convergence limit to attain greater precision in the frequencies and mode shapes. Secondly, as the physical characteristics of the model change during nonlinear response, the vibration characteristics also change. When a previous eigensolution is available, significant cost savings are realized if the method takes advantage of this information in reanalysis.

4. Accuracy and Stability

As with modal synthesis, stability of the operations and accuracy of results of the eigenproblem solution methods are vital. Stability problems tend to be caused by mode shape vectors that are not orthogonal while accuracy is controlled by the convergence criteria.

The superior eigenproblem solution methods from each of the three categories are identified below. The use of these methods and their interdependencies will be discussed in the next section.

Subspace iteration emerges as the superior process in the category of simultaneous interation. The primary advantage over the Jennings and Rutshauser approaches is its applicability to the general problem form. The potentially ill-conditioned transformation to standard form is avoided, thus making the method generally more stable. Other advantages, which are common to all methods in this category, include the ability to use previous modal information in reanalysis and the ease with which acceleration schemes are incorporated.

The most efficient solution process of the transformation methods category is that employing the Householder transformation to tridiagonal form followed by QR iteration (the HQRI method). This method is effective only when the eigenproblem can be written in standard form and when all eigenpairs must be calculated. Although the method does not take advantage of the bandform of the equations, this is not necessarily a drawback since the conversion to standard form often results in a fully populated coefficient matrix. Such is the case when a consistent or a synthesized mass matrix is used.

When the above conversion cannot be made, generalized Jacobi iteration makes an attractive alternative to the HQRI process. Although Jacobi iteration is not as computationally efficient as other transformation methods, it has two distinct advantages. First, as the off-diagonal terms become smaller, the process becomes more efficient, i.e. convergence is more rapid on systems that are almost diagonal.

Secondly, Jacobi iteration is the only transformation method which effectively utilizes a previous solution in reanalysis. Rather than starting the process with a standard rotation matrix to zero an off-diagonal element, the previous mode shape set is used to rotate the coefficient matrices close to diagonal form.

In general, as the order of the eigenproblem and/or its bandwidth increases, polynomial/vector iteration methods become less economical because of their reliance on computing the determinant in Eq. (4.4). An exception is the method presented by Lee and Robinson [4.17]. In this method only one complete triangulation is performed for each eigenpair that is evaluated. The required starting solution, in the form of a prior modal analysis, can be used to advantage in reanalysis.

As mentioned previously, operation counts provide the best evidence regarding the efficiency of a solution algorithm. For comparison purposes, Table 4.1 lists the operation counts for the methods discussed above. It is assumed that no computational penalty is paid by implementing the methods with hypermatrix data structures.

Table 4.I

OPERATION COUNTS FOR EIGENPROBLEM SOLUTION

Method	Number of operations required [4.2, 4.17]
Subspace Iteration.(1)	$nm^2 + nm(4 + 2p) + 4np + 20nq(2m + q + 1.5)$
HQRI.(2)	$0.67n^3 + 10.5n^2 + pn^2 + 9pn$
Generalized Jacobi Iteration.(2,3)	$3n^3 + 6n^2$
Lee-Robinson Vector Iteration	$0.5np(m^2 + 5m + 2) + nT(7m + 6)$

Notes

(1) Assumes 10 iterations required

(2) Assumes fully populated [K] and [M]

(3) Total for one sweep only

Notation

n = order of [K] and [M]

m = half-bandwidth of [K] and [M]

p = number of eigenpairs computed

q = min(2p, p + 8)

T = number of iterations required

4.2.4 Choice of Eigenproblem Solution Methods

It has been demonstrated that a wide variety of eigenproblems can arise in dynamic analysis by the FEM. As mentioned earlier, no single method can be effectively applied to the solution of each of these problems. Therefore, a selection of methods must be available that permits the analyst to choose the one most suited to his problem. The three controlling variables in making the choice are the number of DOF in the model, the bandwidth of the matrices, and the number of eigenpairs to be computed.

In general solution methods from the three catagories are most efficient when applied to a particular class of problem. When all eigenpairs are required of large, fully populated matrices, a transformation method should be used. When only a few frequencies and mode shapes are needed and the equations are narrowly banded, polynomial/vector iteration methods are most efficient. Simuntaneous iteration exhibits its superiority in the range of problems between the two mentioned above.

To fulfill the selection requirements, it is proposed that the superior methods identified in the preceding section be incorporated into the general purpose FEM system. These methods are: subspace iteration, HQRI, generalized Jacobi iteration, and the Lee-Robinson vector iteration. Conveniently, several of these methods can be used in concert. That is, a starting solution for Lee-Robinson vector iteration may be obtained from subspace iteration. Also, subspace iteration effectively utilizes generalized Jacobi iteration in the solution of the reduced eigenproblem. When the transformation to standard form, required by the HQRI approach, is not effective, generalized Jacobi itera-

tion becomes the next best alternative.

Although operation counts provide useful information in choosing a solution method, other factors must be included. The availability of a starting or initial solution may significantly increase the efficiency of some methods. Also, the number of iterations required by some methods is highly dependent on the convergence tolerance that is established. These factors force selection of the appropriate eigenproblem solution method to be made more on the basis of the analyst's experience and judgement rather than on published operation counts.

4.3 Solution of the Equations of Motion

4.3.1 Introduction

Consider the form of the differential equation of motion for a damped finite element system:

$$[M(t)]\{\ddot{u}\} + [C(t)]\{\dot{u}\} + [K(t)]\{u\} = \{P(t)\}. \tag{4.25}$$

For many problems the coefficient matrices of Eq. (4.25) are constant with respect to time, resulting in a set of coupled linear differential equations. When material and/or geometric nonlinearities do arise, they are most often limited to the stiffness matrix, $[K(t)]$. The mass of the structure seldom varies with time and the damping characteristics generally are not sufficiently well understood to warrant any time dependent change in the modal damping ratios.

When multilevel substructured models are used and the coefficient matrices are derived by modal synthesis, the mass and damping matrices can also become time dependent. These dependencies enter the synthesized mass through the coordinate transformations performed on each condensed nonlinear substructure. Likewise, the relationship between damping and the structure mass (see Section 4.4) causes a corresponding variation in the damping matrix. Therefore, in its most general form Eq. (4.25) has all three of its coefficient matrices varying with time.

The transient response of a substructured model is described by the displacements $\{u\}$, velocities $\{\dot{u}\}$, and accelerations $\{\ddot{u}\}$ for the DOF in the highest level structure. Two general strategies are available for obtaining this solution. They are mode superposition and time-history

integration. Both strategies are fundamental to the dynamic analysis of structures and as such are familiar to most analysts.

The selection of a particular solution strategy is highly problem dependent. The decision variables include the degree of nonlinearity (if any), the number of modes participating in the response, the length of the time interval over which the response must be evaluated, and the nature of the transient load. A discussion of these variables and of the effects of multilevel substructured modeling on the solution procedures is presented in the following sections.

4.3.2 Mode Superposition

Mode superposition is an economical approach to solving the equations of motion when the structural response is linear and is limited to the lower modes of the frequency spectrum. For linear systems, the basic approach is to uncouple the equations by a transformation to modal coordinates. The resulting set of equations describe the dynamic response of the structure in each mode when excited by the corresponding modal component of the applied load. These modal equations are solved individually by a direct integration method and then the total response is computed as the sum of the modal responses.

A computational advantage with modal superposition over time-history integration methods (see Section 4.3.3) is realized when the dominant portion of the structural response is contained in the lower modes. This characteristic is evident when the loading function has very little high frequency content. Examples of such loading types might include earthquake excitation and mechanical vibrations. The

number of modes that participate in the response and thus must be included in the modal analysis is easily estimated by examining the loading function after it has been transformed to modal coordinates. Loadings that typically have more high frequency content include blast or shock loading and impact. These later examples are more economically analysed by time-history integration of Eq. (4.25).

The extension of mode superposition to nonlinear systems has received only limited attention in the literature. Morris [4.18] has applied the concept to cable systems and framed structures. His results are not particularly encouraging. The method performed poorly when applied to structures experiencing more than just mild nonlinearities. Nickell [4.21] presented a more complete development of nonlinear mode superposition. Although mode superposition performed very well in his example problems, it showed no particular numerical advantage over direct integration schemes. Nickell does note that some physical insight into the behavior of the structure is gained by observing the changes in the modal spectrum as deformation proceeds.

The procedure for solving Eq. (4.25) by mode superposition remains unchanged when multilevel substructured models are introduced. The transformations performed on the coefficient matrices are independent of the methods used to build those matrices. Although both geometric and modal DOF from the condensed substructures may exist in the structure stiffness, damping, and mass matrices, no complication is expected in making further transformations to uncouple the equations of motion. In fact, the transformations performed in mode superposition are quite similar to those used in modal synthesis. However, no studies have been

located that document the effects of substructuring on the mode superposition solution strategy.

4.3.3 Time-History Integration

When the dynamic loading on a structure excites many modes of vibration, time-history integration, also known as temporal or direct integration, is often a more economical solution procedure than mode superposition. The integration operators work directly on the coupled equations of motion without regard for the modal content of the response.

Operators for time-history integration vary in the relationships between the known and the unknown solution variables. These variables are the displacements, velocities, and accelerations of the structure DOF at various points in time throughout the response period. The assumed relationship between the known and unknown solution variables defines whether the operator is explicit or implicit. In explicit methods [4.2], the displacements at the end of the interval, $\{u_{t+\Delta t}\}$, are based on the equilibrium conditions at the start of the interval: Eq. (4.25) evaluated at time t. After a starting solution has been established, it is possible to progress through the solution without solving a set of simultaneous equations. This feature of explicit methods presents an advantage in computational efficiency over the implicit methods discussed below. However, since explicit methods are extrapolatory in terms of the satisfying equilibrium, they are only conditionally stable [4.15] at best. A stable integration operator is one for which the error introduced in each time step is bounded for the

chosen step size and thus the total error does not grow without bound as the solution progresses. A conditionally stable operator is one which remains stable for only a limited time step size. The central difference method [4.9], when used with a diagonal mass matrix, is considered to be an accurate and economical explicit solution method [4.16].

Implicit operators rely on the solution of Eq. (4.25) at time $t + \Delta t$ to obtain the solution $\{u_{t+\Delta t}\}$. Two significant consequences result. First, a set of linear equations must be solved at each increment in time considered in the solution. Secondly, it is possible to develop implicit operators that are unconditionally stable (stable for any size of time step Δt). The two most popular implicit operators are the Newmark-β method [4.19] and the Wilson-θ method [4.26].

The implicit operators which exhibit unconditional stability when used on linear problems may not always maintain this feature when general nonlinear response occurs. Proofs of unconditional stability of the Newmark-β operator, when applied to nonlinear problems, have been presented [4.4 & 4.7] but contradictory evidence has also been documented [4.20]. The question of unconditional stability is not necessarily a drawback to the use of implicit (or even explicit) operators. For nonlinear analysis relatively small time steps (which lead to small displacement changes) must be taken to accurately detect and incorporate nonlinear behavior. As a result, frequent equilibrium iterations with updates of the coefficient matrices will be required to maintain accuracy, regardless of which integration operator is employed. This process can succeed in arresting any instability that may develop

from using a particular integration operator [4.20].

Explicit integration of the equations of motion has proven to be effective for nonlinear problems that do not involve substructured models. Since the frequent equilibrium iterations prevent instability in the solution, an implicit operator that may be unconditionally stable will not show any advantage over explicit operators. Further, it is not always necessary to use the structure stiffness matrix with explicit operators to compute the internal nodal forces. Computation of the elastic force vector can be performed directly by integration of the following expression for each element in the structure [4.3]:

$$\{F_e\} = \sum_{1}^{n} [L]^T \int_V [B]^T \{\sigma\} dV, \tag{4.26}$$

in which $\{F_e\}$ = structure elastic force vector,

$[L]$ = connectivity matrix for the element,

$[B]$ = strain-displacement relation,

$\{\sigma\}$ = element stress vector,

V = volume of the element, and

n = number of elements in the structure.

In substructured models the element stress vector is not available for condensed lower level substructures. As a result, computation of the elastic force vector must incorporate at least a portion of the structure stiffness matrix. The resulting elastic forces are

$$\{F_e\} = \{F_e\}_H + [K]_c \{u\}_c \tag{4.27}$$

where $\{F_e\}$ = structure elastic force vector,

$\{F_e\}_H$ = internal forces for uncondensed elements at the

highest level, computed by Eq. (4.26),

$[K]_c$ = stiffness matrix for the adjoining condensed substructures, and

$\{u\}_c$ = condensed DOF displacement vector.

Computation of elastic forces by Eq. (4.27) requires that the stiffness matrix for the condensed substructures be brought into memory during each time step. It is yet unknown whether this requirement will reduce the computational effectiveness of explicit integration of the nonlinear equations of motion.

Since implicit methods require equation solving with the full structure stiffness matrix, no complications in the algorithms due to the use of substructured models are anticipated.

4.4 Additional Computational Considerations

Several minor computational details regarding the dynamic analysis of structures are complicated by the use of substructured finite element models. Details considered in this study include mass matrix formulation, damping, and the use of experimental data.

Two approaches are used to assemble the mass matrix of a substructure containing only simple elements [4.6]. The most elementary approach is the lumped mass formulation. With this approach, the mass of each element is lumped at the nodal points. Upon assembly, a diagonal substructure mass matrix is achieved. This form has many computational advantages as previously noted. The alternate approach is the consistent mass formulation. In this approach, accelerations within the element are interpolated by the same shape functions used to describe the

element displacements. Thus, the assembled substructure mass matrix has the same banded nature as the substructure stiffness. Regardless of initial formulation, when multilevel substructuring is introduced, the resulting structure mass matrix is of banded form. It is anticipated that as the number of levels of substructuring increases, the differences between the lumped and consistent formulations will diminish. The reduction process used, through the various transformations performed, is expected to mask the original mass matrix formulation and thus make the two formulations indistinguishable.

Structural damping is a phenomenon which is not easily modelled by analysts. Usually, damping can only be evaluated by experimental testing thus making its specification for a general finite element model difficult. However, there are procedures designed to include the effects of damping in a computationally efficient manner [4.6]. Normally, modal damping ratios are known (or selected) and used directly in computing structural response in modal coordinates. If it is desirable to have an explicit damping matrix in geometric coordinates, one can be formulated which is proportional to both the stiffness and mass matrices. In this form, termed Rayleigh damping, the system vibration mode shapes are orthogonal to the damping as well as the mass and stiffness matrices. In substructured models, application of damping is delayed until the equations of motion for the highest level structure are formulated. The methods of Guyan reduction and modal synthesis do not provide for damping at the substructure level.

Pre-existing experimental data regarding the free vibration characteristics of a substructure could be useful for reducing the computational expenses of a finite element analysis. It is a simple matter to adapt input translators to accept this data and thus eliminate unnecessary vibration analyses. An important consideration in using experimental data is the consistency in boundary restraint between the test fixture and the finite element model.

4.5 References

4.1 Bathe, K.J., and Ramaswamy, S., "An Accelerated Subspace Iteration Method," Computer Methods in Applied Mechanics and Engineering, Vol. 23 pp. 313-331 (1980)

4.2 Bathe, K.J., and Wilson, E.L., Numerical Methods in Finite Element Analysis, Prentice Hall, 1976

4.3 Belytschko, T., Chiapetta, R.L., and Bartel, H.D.. "Efficient Large Scale Nonlinear Transient Analysis by Finite Elements," International Journal for Numerical Methods in Engineering, Vol. 10 pp. 579-596 (1976)

4.4 Belytschko, T., and Schoeberle, D.F., "On the Unconditional Stability of an Implicit Algorithm for Nonlinear Structural Dynamics," Journal of Applied Mechanics, Vol. 97 pp. 865-869 (1975)

4.5 Clint, M. and Jennings, A., "The Evaluation of Eigenvalues and Eigenvectors of Real Symmetric Matrices by Simultaneous Iteration," The Computer Journal, Vol. 13 pp. 76-80 (1970)

4.6 Clough, R.W. and Penzien, J., Dynamics of Structures, McGraw Hill, 1975

4.7 Corr, R.B. and Jennings, A., "A Simultaneous Iteration Algorithm for Symmetric Eigenvalue Problems." International Journal for Numerical Methods in Engineering Vol. 10 pp. 647-663 (1976)

4.8 Dong, S.B., Wolf, J.A., and Peterson, F.E., "On a Direct-Iterative Eigensolution Technique." International Journal for Numerical Methods in Engineering, Vol. 4 pp. 155-161 (1972)

4.9 Goudreau, G.L. and Taylor, R.L., "Evaluation of Numerical Integration Methods in Elastodynamics," Computer Methods in Applied Mechanics and Engineering, Vol. 2 pp. 69-97 (1972)

4.10 Gupta, K.K., "Eigenproblem Solution by a Combined Sturm Sequence and Inverse Iteration Technique." International Journal for Numerical Methods in Engineering, Vol. 7 pp. 17-42 (1973)

4.11 Hughes, T.J.R., "A Note on the Stability of Newmark's Algorithm in Nonlinear Structural Dynamics," International Journal for Numerical Methods in Engineering, Vol. 11 pp. 383-386 (1977)

4.12 Jennings, A., "A Direct Iteration Method of Obtaining Latent Roots and Vectors of a Symmetric Matrix," Proceedings of the Cambridge PhilosophicalSociety, Vol. 63 pp. 755-765 (1967)

4.13 Jennings, A. and Orr, D.R.L., "Application of the Simultaneous Iteration Method to Undamped Vibration Problems," International Journal for Numerical Methods in Engineering, Vol. 3 pp. 13-24 (1971)

4.14 Johnsen, T.L., "On the Computation of Natural Modes of an Unsupported Vibrating Structure by Simultaneous Iteration," Computer Methods in Applied Mechanics and Engineering, Vol. 2 pp. 305-322 (1973)

4.15 Krieg, R.D., "Unconditional Stability in Numerical Time Integration Methods, Journal of Applied Mechanics, Vol. 40 pp. 417-421 (1973)

4.16 Krieg, R.D. and Key, S.W., "Transient Shell Response by Numerical Time Integration," International Journal for Numerical Methods in Engineering, Vol. 7 pp. 273-286 (1973)

4.17 Lee, I-W, and Robinson, A.R., "Solution Techniques for Large Eigenvalue Problems in Structural Dynamics," Technical Report No. UILU ENG-79-2006, Dept. of Civil Engineering, Univ. of Ill., Urbana, (1979)

4.18 Morris, N.F., "The Use of Modal Superposition in Nonlinear Dynamics," Computers and Structures, Vol. 7 pp. 65-72 (1977)

4.19 Newmark, N.M., "A Method of Computation for Structural Dynamics," Journal of the Engineering Mechanics Division (ASCE), Vol. 85 pp. 67-94 (1959)

4.20 Nickell, R.E., "Direct Integration Methods in Structural Dynamics," Journal of the Engineering Mechanics Division (ASCE), Vol. 99 pp. 303-316 (1973)

4.21 Nickell, R.E., "Nonlinear Dynamics by Mode Superposition," Computer Methods in Applied Mechanics and Engineering, Vol. 7 pp. 107-129 (1976)

4.22 Robinson, A.R. and Harris, J.F., "Improving Approximate Eigenvalues and Eigenvectors," Journal of the Engineering Mechanics Division (ASCE), Vol. 97 pp. 457-475 (1971)

4.23 Rutishauser, H., "Computational Aspects of F. L. Bauer's Simultaneous Iteration Method," Numerische Mathematik, Vol. 13 pp. 4-13 (1969)

4.24 Weaver, W. Jr. and Yoshida, D.M. "The Eigenvalue Problem for Banded Matrices," Computers and Structures, Vol. 1 pp. 651-664 (1971)

4.25 Wilkinson, J.H., The Algebraic Eigenvalue Problem, Oxford University Press, 1965

4.26 Wilson, E.L., Farhoomand, I., and Bathe, K.J., "Nonlinear Dynamic Analysis of Complex Structures," Earthquake Engineering and Structural Dynamics, Vol. 1 pp. 241-252 (1973)

4.27 Wylie, C.R., Advanced Engineering Mathematics, 4th Edition, McGraw Hill, 1975

4.28 Yamamoto, Y. and Ohtsubo, H., "Subspace Iteration Accelerated by Using Chebyshev Polynomials for Eigenvalue Problems with Symmetric Matrices," International Journal for Numerical Methods in Engineering, Vol. 10 pp. 935-944 (1976)

CHAPTER 5

MATRIX FORM OF NONLINEAR CONTINUUM MECHANICS

5.1 Introduction

The dynamic solution procedures described in the preceeding chapters are independent of the continuum theory used to derive the governing nonlinear equations of motion. This chapter focuses on the essential aspects of nonlinear continuum theories. This information is included in the report for three reasons: (1) the formulations are not readily available in matrix notation familiar to finite element workers, (2) the exact transformations in matrix form will be required during the implementation phase, and (3) a relative assessment of efficiency of the formulations is desired. The finite element formulations are developed and contrasted as to their suitability for substructured dynamic analysis in the next chapter.

Initial work on a matrix formulation for nonlinear continuum theory was conducted by Nayak [5.1]. Zienkiewicz [5.2] has been the primary advocate of this approach which has been adopted by most researchers at Swansea University. This chapter summarizes their work and presents an extension of the approach to include an Updated Lagrangian formulation.

Nonlinear behavior includes that due to constitutive relationships (material nonlinearity) and strain-displacement relationships (geometric nonlinearity). To achieve maximum generality in the discussion, structures are assumed to exhibit both types of nonlinearity simultaneously. The discussion of material nonlinearity is directed towards metal plasticity at both infinitesimal and finite strain magnitudes. While it

is recognized that strain-rate dependent behavior may be of importance in dynamic analysis, these effects are not considered in this discussion. Few procedures in the literature include strain-rate dependencies, creep and thermal dependencies of material properties. Most often, rate effects are incorporated into the basic concepts of incremental plasticity and thus do not affect the comparison of nonlinear formulations.

The dynamic analysis of structures in which nonlinearity is limited to the material stress-strain behavior presents no complications in formulating the equations of motion. This follows as all quantities are referred to the initial configuration of the structure, with displacements and rotations assumed to be infinitesimal. The challenge is developing realistic constitutive models for the material behavior under cyclic loading. Geometric nonlinear effects are important when the structure undergoes large rotations and/or finite strain magnitudes. This occurs, for example, in the following situations: crashworthiness of aircraft structures and components, metal forming processes, necking of structural components subjected to tensile overloads, geometry changes under service loading that affect performance (e.g., turbine blades) and the localized deformation of material in the vicinity of stress concentrations -- especially notches and cracks.

The governing equations of nonlinear solid mechanics are summarized in the following sections in terms of Cartesian reference axes. The notation adopted here is a modification of that first used by Bathe [5.3]. Each quantity retains the same symbol throughout the deformation process. Superscripts and subscripts are used to indicate the configuration of the structure in which the variable occurs and to which con-

figuration it is referenced, respectively. A single left subscript implies an increment of the variable over time Δt with the subscript indicating the reference configuration. A summary of the notation is presented in Section 5.9.

All equations are derived using matrix, rather than tensor, notation. The theory of finite deformation is elegantly expressed in tensor notation. However, all tensor operations require transformation to equivalent matrix form for computer implementation. This, combined with the exclusive use of matrix notation in finite element research, encourages the adoption of matrices at the start. The reader already familiar with tensor notation will readily recognize the appearance of symmetric strain and stress "matrices."

The following sections contain discussions of coordinate systems, strain-displacement relations, stress definitions, and virtual displacements for finite deformations. Particular emphasis is placed on the finite deformation strain and stress rates that are required for virtual work arguments. Two nonlinear formulations arise in the discussion. These are termed Total Lagrangian (T. L.) and Updated Lagrangian (U. L.). In the T. L. formulation all quantities are referenced to the configuration at time $t=0$. The difficulty arises with constitutive relations for the material which are naturally expressed in terms of the deformed configuration, e.g., true stress and true strain. The correct transformations of strain, stress, strain rates, stress rates, and incremental constitutive relationships between the initial and instantaneous configurations are derived in this chapter. The U. L. formulation refers all quantities at time $t+\Delta t$ to the configuration at time t. Numerous attempts have been made to derive an U. L. formulation

for finite element analysis. Unfortunately, very few have followed the rigorous equations of continuum mechanics; most formulations are derived by assuming certain nonlinear terms are small during the motion from t to $t+\Delta t$. This has led to considerable misunderstanding in the literature. The exact U. L. formulation is rigorously derived in this chapter. Only then are several commonly used simplifying assumptions described.

An extensive discussion of constitutive models for finite deformation elasto-plasticity is included in this chapter. The various strain and stress rate definitions and their transformations are presented in matrix notation. This material will enable the realistic modeling of large deformation effects that occur near impact zones. It is anticipated that the use of substructuring prior to a transient analysis will show a significant computational advantage in such problems.

The chapter concludes with a qualitative assessment of the computation efficiency of the two major formulations.

5.2 Coordinate Systems and Transformations

The configuration of the body is considered at three times; namely, 0, t, and $t+\Delta t$. For static analysis, the parameter t may be associated with a loading state rather than time. Motion of the body is described with respect to fixed Cartesian axes having unit vectors $\{i\}$. The position vector of any material point at the three times is denoted ${}^{0}\{x\}$, ${}^{t}\{x\}$, and ${}^{t+\Delta t}\{x\}$. Coordinates ${}^{0}\{x\}$ remain fixed for a particular material point and thus are convected in the present usage. No subscripts are necessary for the position vectors and their corresponding differentials.

Let ${}^{0}\{dx\}$ represent the components of an initial line segment, ${}^{0}ds$, which deforms into line segment ${}^{t}ds$ at time t. Components ${}^{t}\{dx\}$ then are given by

$$ {}^{t}\{dx\} = {}^{t}_{0}[J]\ {}^{0}\{dx\} \tag{5.1} $$

in which ${}^{t}_{0}[J]$ is the deformation Jacobian with terms defined by

$$ {}^{t}_{0}J_{ij} = {}_{0}({}^{t}x_{i,j}) \tag{5.2} $$

The subscript outside the ( ) explicitly indicates the reference configuration for differentiation. The deformation Jacobian fully characterizes motion in the differential neighborhood of a point that is displaced from ${}^{0}P$ to ${}^{t}P$. Geometric considerations [5.1] show that differential volume and area changes may be expressed as

$$ {}^{t}dV = {}^{t}_{0}|J|\ {}^{0}dV \tag{5.3} $$

$$ {}^{t}\{dA\} = {}^{t}_{0}|J|\ {}^{t}_{0}[J]^{-T}\ {}^{0}\{dA\} \tag{5.4} $$

in which ${}^{0}dV$ is a differential volume and ${}^{0}\{dA\}$ is the vector of components for a differential area ${}^{0}dA$, both at $t=0$. Vertical bars are used to indicate a determinant.

The position vector ${}^{t}\{x\}$ is written in terms of a displacement vector ${}^{t}\{u\}$ with components referred to base vectors $\{i\}$ as

$$ {}^{t}\{x\} = {}^{0}\{x\} + {}^{t}\{u\} \tag{5.5} $$

The deformation Jacobian may then be written in the form

$$ {}^{t}_{0}[J] = [I] + {}^{t}_{0}[j] \tag{5.6} $$

where ${}^{\tau}_{o}[j]$ is termed the displacement Jacobian and [I] is an identity matrix. Terms of ${}^{\tau}_{o}[j]$ are defined by

$$ {}^{\tau}_{o}j_{i,j} = {}_{o}({}^{\tau}u_{i,j}) \tag{5.7} $$

Equations (5.1) - (5.7) provide the basis of strain-stress transformations required in the T. L. formulation.

The finite magnitude changes in volume and area of differential elements that occur during the motion from τ to $\tau+\Delta\tau$ may be measured with respect to the configuration at time 0 or τ. First, the configuration az $\tau+\Delta\tau$ is expressed in terms of the configuration at τ and the displacement increment (deformation plus rotation). No restrictions are imposed on the magnitude of the displacement increment during $\Delta\tau$. In analogy with (5.1) an expression of the form

$$ {}^{\tau+\Delta\tau}\{dx\} = {}^{\tau+\Delta\tau}_{\tau}[J]\,{}^{\tau}\{dx\} \tag{5.8} $$

is sought in which ${}^{\tau+\Delta\tau}_{\tau}[J]$ is the increment of the deformation Jacobian over $\Delta\tau$ but measured with respect to the configuration at time τ. The terms of the Jacobian increment are given by

$$ {}^{\tau+\Delta\tau}_{\tau}J_{ij} = {}_{\tau}({}^{\tau+\Delta\tau}x_{i,j}) \tag{5.9} $$

Let ${}^{\tau}\{\Delta u\}$ represent the incremental displacement vector: ${}^{\tau}\{\Delta u\}^{T} = \lfloor {}^{\tau}\Delta u,\ {}^{\tau}\Delta v,\ {}^{\tau}\Delta w \rfloor$ during motion from τ to $\tau+\Delta\tau$. Then

$$ {}^{\tau+\Delta\tau}_{\tau}(J_{ij}) = {}_{\tau}({}^{\tau}x_{i,j}) + {}_{\tau}({}^{\tau}\Delta u_{i,j}) \tag{5.10} $$

More simply, (5.10) may be written

$$ {}^{\tau+\Delta\tau}_{\tau}[J] = [I] + {}^{\tau+\Delta\tau}_{\tau}[v] \tag{5.11} $$

in which ${}^{t+\Delta t}_{t}[v]$ represents the mean displacement gradient with respect to the current (t) configuration (or equivalently the mean velocity gradient if divided by Δt). Increments of differential volumes and areas from t to $t+\Delta t$ may be similarly expressed in analogy with (5.3) and (5.4) as

$$ {}^{t+\Delta t}dV = {}^{t+\Delta t}_{t}|J| \; {}^{t}dV \tag{5.12} $$

$$ {}^{t+\Delta t}\{dA\} = {}^{t+\Delta t}_{t}|J| \; {}^{t+\Delta t}_{t}[J]^{-T} \; {}^{t}\{dA\} \tag{5.13} $$

Equations (5.8) - (5.13) provide the basis for transformations required in the U. L. formulation as first presented by Yaghmai and Popov [5.4]. The U. L. formulation is generally attributed to Bathe [5.3, 5.5] due to the numerous papers he has published on the formulation.

Variables for the configuration at $t+\Delta t$ may also be related to that at t=0 through a sequence of transformations -- first from $t+\Delta t$ to t, then from t to t=0. Consider first that at $t+\Delta t$,

$$ {}^{t+\Delta t}\{dx\} = {}^{t+\Delta t}_{o}[J] \; {}^{o}\{dx\} \tag{5.14} $$

Subtraction of (5.1) from the above expression yields

$$ {}^{t+\Delta t}\{dx\} - {}^{t}\{dx\} = {}_{o}[\Delta J] \; {}^{o}\{dx\} = \left[{}^{t+\Delta t}_{o}[J] - {}^{t}_{o}[J] \right] {}^{o}\{dx\} \tag{5.15} $$

in which ${}_{o}[\Delta J]$ is the incremental deformation Jacobian but referred to the configuration at t=0. In (5.8), ${}^{t+\Delta t}_{t}[J]$ is the incremental deformation Jacobian referred to the configuration at time t. Using the incremental displacement vector ${}^{t}\{\Delta u\}$ over the time increment Δt, the terms of ${}_{o}[\Delta J]$ are

$$ {}_{o}(\Delta J_{ij}) = {}_{o}({}^{t}x_{i} + {}^{t}\Delta u_{i})_{i,j} - {}_{o}({}^{t}x_{i,j}) \tag{5.16} $$

or, more simply

$$ {}_{o}(\Delta J_{ij}) = {}^{\dagger}_{o}(\Delta u)_{i,j} \tag{5.17} $$

Using the chain rule, the derivatives in (5.17) with respect to ${}^{o}\{x\}$ are transformed to those with respect to ${}^{\dagger}\{x\}$. The operations readily show that the result may be written as

$$ {}_{o}[\Delta J] = {}^{\dagger+\Delta\dagger}_{\dagger}[v]\ {}^{\dagger}_{o}[J] \tag{5.18} $$

Substitution of (5.18) into (5.15) shows that

$$ {}^{\dagger+\Delta\dagger}\{dx\} = \left[\ [I] + {}^{\dagger+\Delta\dagger}_{\dagger}[v]\ \right] {}^{\dagger}[J]\ {}^{o}\{dx\} \tag{5.19} $$

and since ${}^{\dagger}\{dx\} = {}^{\dagger}_{o}[J]\ {}^{o}\{dx\}$, the above expression simplifies to

$$ {}^{\dagger+\Delta\dagger}\{dx\} = {}^{\dagger+\Delta\dagger}_{\dagger}[J]\ {}^{\dagger}\{dx\} \tag{5.20} $$

which is the same result obtained in (5.8). This procedure illustrates the multiplicative decomposition of deformation that parallels simple coordinate transformation (which must be the case when the displacement gradients represent a rigid rotation).

Consider now the imposition of virtual, rather than finite, displacement increments on the configuration at time $\dagger$. Corresponding virtual changes in differential areas and volumes are desired. Given the differential volumes and areas at time $\dagger$, the expressions previously derived for finite changes during $\Delta\dagger$ are re-examined for the limiting case as $\Delta\dagger \to 0$, i.e., the virtual displacement occurs during a vanishing increment of the time (or loading) parameter. Introducing the first variational operator, δ, the variation of line segments is given by

$$ \delta\,{}^{\dagger}\{dx\} = {}^{\dagger}_{o}[\delta J]\ {}^{o}\{dx\} \tag{5.21} $$

in which terms of the variation of the deformation Jacobian appearing above are defined by

$$ {}^{\dagger}_{o}\delta J_{ij} = {}_{o}({}^{\dagger}\delta x_{i,j}) = {}_{o}({}^{\dagger}\delta u_{i,j}) \tag{5.22} $$

since the variation of the initial coordinates, ${}^{o}\{x\}$, is zero under the imposed virtual displacement. If the virtual displacements, ${}^{\dagger}\{\delta u\}$, are interpreted as occurring over a time dt, the derivatives in (5.22) may be considered virtual velocity gradients. Using the chain rule, the virtual displacement gradients are referred to the configuration at time t with the resulting form

$$ {}^{\dagger}_{o}[\delta J] = {}^{\dagger}_{\dagger}[\delta v]\ {}^{\dagger}_{o}[J] \tag{5.23} $$

in which the variational form of the displacement gradient matrix defined in (5.17) and (5.18) is used. Substitution of (5.23) into (5.21) provides the variation of the deformation Jacobian with respect to the current configuration in analogy with (5.8)

$$ \delta^{\dagger}\{dx\} = {}^{\dagger}_{\dagger}[\delta v]\ {}^{\dagger}_{o}[J]\ {}^{o}\{dx\} \tag{5.24} $$

but ${}^{\dagger}_{o}[J]\ {}^{o}\{dx\}$ is simply ${}^{\dagger}\{dx\}$, thus the above simplifies to

$$ \delta^{\dagger}\{dx\} = {}^{\dagger}_{\dagger}[\delta v]\ {}^{\dagger}\{dx\} \tag{5.25} $$

which implies that

$$ {}^{\dagger}_{\dagger}[\delta J] = {}^{\dagger}_{\dagger}[\delta v] \tag{5.26} $$

Using Euler's theorem for homogeneous functions, it is shown that the variation of the deformation Jacobian determinant is given by

$$ {}^{\dagger}_{o}|\delta J| = \mathrm{Tr}({}^{\dagger}_{\dagger}[\delta v])\ {}^{\dagger}_{o}|J| \tag{5.27} $$

in which Tr( ) denotes the trace of the matrix. It is customary to express the virtual displacement gradient matrix, ${}^{\dagger}_{\dagger}[\delta v]$, as the sum of a symmetric deformation matrix, ${}^{\dagger}_{\dagger}[\delta e]$, and an anti-symmetric spin matrix, ${}^{\dagger}_{\dagger}[\delta w]$, as

$$ {}^{\dagger}_{\dagger}[\delta v] = {}^{\dagger}_{\dagger}[\delta e] + {}^{\dagger}_{\dagger}[\delta w] \tag{5.28} $$

in which

$$ {}^{\dagger}_{\dagger}[\delta e] = \frac{1}{2} \; {}^{\dagger}_{\dagger}[\; [\delta v] + [\delta v]^{T} \;] \tag{5.29} $$

and

$$ {}^{\dagger}_{\dagger}[\delta w] = \frac{1}{2} \; {}^{\dagger}_{\dagger}[\; [\delta v] - [\delta v]^{T} \;] \tag{5.30} $$

Both the deformation and spin components are linear in the virtual displacement derivatives. Biot [5.6] has demonstrated that ${}^{\dagger}_{\dagger}[\delta w]$ represents the rotation of rigid-convected axes attached to point ${}^{\dagger}P$ during the small (infinitesimal) virtual displacement. Similarly, ${}^{\dagger}_{\dagger}[\delta e]$ is shown to constitute the pure deformation caused by the virtual displacements. A null deformation matrix is therefore a necessary and sufficient condition for a rigid virtual displacement. The 6 x 1 vector representation of the symmetric deformation matrix is denoted ${}^{\dagger}_{\dagger}\{\delta e\}$ (shear terms are doubled to form the vector representation).

Using the above procedure, the following relationships are derived for the U. L. formulation.

$$ \delta {}^{\dagger+\Delta\dagger}\{dx\} = {}^{\dagger+\Delta\dagger}_{\dagger}[\delta J]^{\dagger}\{dx\} \tag{5.31} $$

$$ {}^{\dagger+\Delta\dagger}_{\dagger}[\delta J] = {}^{\dagger+\Delta\dagger}_{\dagger+\Delta\dagger}[\delta v] \; {}^{\dagger+\Delta\dagger}_{\dagger}[J] \tag{5.32} $$

These relationships are analogous to (5.21) and (5.23). The virtual displacement gradients at time $t+\Delta t$ may again be expressed as the sum of a symmetric deformation matrix, ${}^{t+\Delta t}_{t+\Delta t}[\delta e]$, and an anti-symmetric spin matrix, ${}^{t+\Delta t}_{t+\Delta t}[\delta w]$, as in (5.28-5.30).

5.3 Strain-Displacement Relations

Strain measures valid for arbitrarily large deformations are presented in this section. Corresponding expressions for the variations of these measures are derived for subsequent use in virtual work equations. The strain measures and variations are written in terms of the configuration at times 0 and t corresponding to the T. L. and U. L. approaches.

The measure of finite deformation is taken as the difference in the squared lengths before and after deformation of a line segment (ds) in the infinitesimal neighborhood of a point.

$$\text{deformation} = {}^{t}ds^2 - {}^{0}ds^2 = {}^{t}\{dx\}^T\,{}^{t}\{dx\} - {}^{0}\{dx\}^T\,{}^{0}\{dx\} \tag{5.33}$$

The deformation may be written in terms of the initial line segment components, ${}^{0}\{dx\}$, using the deformation Jacobian to yield

$${}^{t}ds^2 - {}^{0}ds^2 = 2\,{}^{0}\{dx\}^T\,{}^{t}_{0}[\varepsilon]\,{}^{0}\{dx\} \tag{5.34}$$

The symmetric, Green strain matrix, ${}^{t}_{0}[\varepsilon]$, appearing above is defined by

$${}^{t}_{0}[\varepsilon] = \frac{1}{2}\left({}^{t}_{0}[J]^T\,{}^{t}_{0}[J] - [I]\right) \tag{5.35}$$

Alternatively, the strain may be referred to the final line segment components, ${}^{t}\{dx\}$, using the inverse deformation Jacobian in the form

$${}^{t}ds^2 - {}^{0}ds^2 = 2\,{}^{t}\{dx\}^T\,{}^{t}_{t}[\varepsilon]\,{}^{t}\{dx\} \tag{5.36}$$

The symmetric Almansi strain matrix, ${}^{t}_{t}[\varepsilon]$, appearing above is defined by

$$ {}^{t}_{t}[\varepsilon] = \frac{1}{2} \left([I] - {}^{o}_{t}[J]^{T} \, {}^{o}_{t}[J] \right) \tag{5.37} $$

in which

$$ {}^{o}_{t}[J] = {}^{t}_{o}[J]^{-1} \tag{5.38} $$

Finally, it is verified by direct substitution that the Green and Almansi strains are related by

$$ {}^{t}_{o}[\varepsilon] = {}^{t}_{o}[J]^{T} \, {}^{t}_{t}[\varepsilon] \, {}^{t}_{o}[J] \tag{5.39} $$

The Almansi strain matrix is often referred to as the "true" finite strain measure since the deformed configuration provides the reference state. When ${}^{t}_{o}[J]$ represents a rigid rotation, the transformation in (5.39) is recognized as a simple change of reference axes.

The symmetric strain matrix may be written in 6x1 vector form as ${}^{t}_{o}\{\varepsilon\}$ and ${}^{t}_{t}\{\varepsilon\}$ for the Green and Almansi definitions respectively. Shear strains from the matrix definition are doubled in the vector representation. The Almansi to Green strain transformation in (5.39) is written in vector form as

$$ {}^{t}_{o}\{\varepsilon\} = {}^{t}_{o}[T] \, {}^{t}_{t}\{\varepsilon\} \tag{5.40} $$

in which the terms of the transformation matrix ${}^{t}_{o}[T]$ are given in the appendix to this chapter (Section 5.9).

Both the Green and Almansi strain components correctly describe the finite deformation due to a nonhomogeneous displacement field within the infinitesimal neighborhood of a point ${}^{o}P$ that displaces to ${}^{t}P$ during the motion. The Green strain components are employed in both the T. L. and

U. L. formulations. The incremental form of the Almansi strain is used to develop the material constitutive models.

It is useful to separate the Almansi strain into linear and non-linear components by first writing

$$ {}^{o}_{t}[J] = [I] - {}^{t}_{t}[j] \tag{5.41} $$

in which

$$ {}^{t}_{t}(j_{ij}) = {}_{t}({}^{t}u_{i,j}) \tag{5.42} $$

Substitution of (5.41) into (5.37) yields

$$ {}^{t}_{t}[\varepsilon] = \frac{1}{2} \left({}^{t}_{t}[j] + {}^{t}_{t}[j]^{T} - {}^{t}_{t}[j]^{T} \, {}^{t}_{t}[j] \right) \tag{5.43} $$

The linear displacement gradient terms are now obvious and are denoted by

$$ {}^{t}_{t}[e] = \frac{1}{2} \, {}^{t}_{t}\left[[j] + [j]^{T} \right] \tag{5.44} $$

The matrix ${}^{t}_{t}[e]$ is often termed the linear "true" strain. The similarity of the above expression with (5.29) is noted. If a virtual Almansi strain is derived from a virtual displacement field, the linear term of ${}^{t}_{t}[\delta\varepsilon]$ is simply ${}^{t}_{t}[\delta e]$, the virtual deformation matrix.

In the U. L. formulation it is necessary to define the strain at time $t+\Delta t$ with respect to the configuration at time t. By the same procedure used to develop (5.33), the deformation increment is given as

$$ {}^{t+\Delta t}_{t}ds^{2} - {}^{t}_{t}ds^{2} = {}^{t+\Delta t}_{t}\{dx\}^{T} \, {}^{t+\Delta t}_{t}\{dx\} - {}^{t}_{t}\{dx\}^{T} \, {}^{t}_{t}\{dx\} \tag{5.45} $$

Substitution of (5.8) for ${}^{t+\Delta t}_{t}\{dx\}$ in terms of the incremental deformation Jacobian yields

$$ {}^{t+\Delta t}ds^2 - {}^{t}ds^2 = 2\ {}^{t}\{dx\}^T\ {}^{t+\Delta t}_{t}[\varepsilon]\ {}^{t}\{dx\} \tag{5.46} $$

in which

$$ {}^{t+\Delta t}_{t}[\varepsilon] = \frac{1}{2}\left({}^{t+\Delta t}_{t}[J]^T\ {}^{t+\Delta t}_{t}[J] - [I]\right) \tag{5.47} $$

represents the symmetric Green strain matrix during the motion from t to t+Δt but referred to the configuration at time t. The 6x1 vector form of this strain is denoted ${}^{t+\Delta t}_{t}\{\varepsilon\}$. This expression is <u>exact</u>; no restrictions are placed on the magnitude of the displacement increment ${}^{t}\{\Delta u\}$.

The strain defined in (5.47) cannot be added to that defined in (5.35) to determine the Green strain at t+Δt referred to time 0. However, using (5.8) it is easily shown that

$$ {}^{t+\Delta t}_{0}[\varepsilon] = {}^{t}_{0}[\varepsilon] + {}^{t}_{0}[J]^T\ {}^{t+\Delta t}_{t}[\varepsilon]\ {}^{t}_{0}[J] \tag{5.48} $$

or in vector form

$$ {}^{t+\Delta t}_{0}\{\varepsilon\} = {}^{t}_{0}\{\varepsilon\} + {}^{t}_{0}[T]\ {}^{t+\Delta t}_{t}\{\varepsilon\} \tag{5.49} $$

This transformation implies that, relative to the initial configuration, the strain increment employed in U. L. is in actuality an Almansi strain increment referred to the configuration at t.

When the body is subjected to a virtual displacement field, a corresponding virtual strain field is produced. Application of the first variational operator, δ, on the Green strain matrix of (5.35) yields

$$ {}^{t}_{0}[\delta\varepsilon] = \frac{1}{2}\left({}^{t}_{0}[J]^T\ {}^{t}_{0}[\delta J] + {}^{t}_{0}[\delta J]^T\ {}^{t}_{0}[J]\right) \tag{5.50} $$

Substitution of the variation of the deformation Jacobian from (5.23) permits the above equation to be written in the form

$$ {}^{t}_{o}[\delta\varepsilon] = \frac{1}{2} \left({}^{t}_{o}[J]^{T} \, {}^{t}_{t}[\delta v] \, {}^{t}_{o}[J] + {}^{t}_{o}[J]^{T} \, {}^{t}_{t}[\delta v]^{T} \, {}^{t}_{o}[J] \right) \tag{5.51} $$

which may be simplified using (5.28) to provide

$$ {}^{t}_{o}[\delta\varepsilon] = {}^{t}_{o}[J]^{T} \, {}^{t}_{t}[\delta e] \, {}^{t}_{o}[J] \tag{5.52} $$

The Green strain rate matrix measured with respect to the initial configuration, is symmetric and is given by the transformed, symmetric deformation matrix ${}^{t}_{t}[\delta e]$ (which is the linear portion of the virtual Almansi strain). Since ${}^{t}_{t}[\delta e]$ vanishes under a rigid virtual displacement, the virtual Green strain also vanishes. Using vector forms of the virtual quantities, the above transformation is written as

$$ {}^{t}_{o}\{\delta\varepsilon\} = {}^{t}_{o}[T] \, {}^{t}_{t}\{\delta e\} \tag{5.53} $$

Inverting the above relationship yields an expression for the deformation vector

$$ {}^{t}_{t}\{\delta e\} = {}^{t}_{o}[T]^{-1} \, {}^{t}_{o}\{\delta\varepsilon\} \tag{5.54} $$

Using a differential operator matrix which acts on the virtual displacement vector, the virtual Green strain vector is easily evaluated as

$$ {}^{t}_{o}\{\delta\varepsilon\} = {}^{t}_{o}[B] \, {}^{t}\{\delta u\} \tag{5.55} $$

The terms of this operator matrix are defined in the appendix to this chapter.

For the U. L. formulation, the virtual strain at time $t+\Delta t$ referred to the configuration at time t is required. Application of the variational operator on (5.47) followed by a substitution for the variation of the incremental deformation Jacabian from (5.32) yields

$$ {}^{t+\Delta t}_{t}[\delta\varepsilon] = {}^{t+\Delta t}_{t}[J]^{T}\ {}^{t+\Delta t}_{t+\Delta t}[\delta e]\ {}^{t+\Delta t}_{t}[J] \tag{5.56} $$

Use of the vector form of the deformation and strain matrices permits the above transformation to be written as

$$ {}^{t+\Delta t}_{t}\{\delta\varepsilon\} = {}^{t+\Delta t}_{t}[T]\ {}^{t+\Delta t}_{t+\Delta t}\{\delta e\} \tag{5.57} $$

in which ${}^{t+\Delta t}_{t}[T]$ is defined in the appendix. An expression analagous to (5.55) for U. L. is described in Chapter 6.

5.4 Stress Measures and Rates

The most common stress measures employed in finite deformation theory and their corresponding rates are described in this section. In the deformed configuration, the "true" or Cauchy stress components provide the natural measure. As in classical elasticity theory, these components are defined by considering the equilibrium of a differential tetrahedron extracted from the body at any time t and acted upon by the differential force vector ${}^{t}\{dF\}$. The tetrahedron has three surfaces parallel to the fixed reference axes. The resulting stress matrix, denoted ${}^{t}[\sigma]$, is symmetric in the absence of body couples due to the orthogonality of the reference planes.

Alternatively, stresses acting on the undeformed areas ${}^{0}\{dA\}$ may be defined as those resisting the same differential force vector ${}^{t}\{dF\}$.

Using (5.4), the stresses can be transformed from deformed to undeformed areas such that of the

$$ {}^{t}\{dF\} = {}^{t}_{o}[\sigma]\ {}^{t}\{dA\} = {}^{t}_{o}|J|\ {}^{t}_{o}[\sigma]\ {}^{t}_{o}[J]^{-T}\ {}^{o}\{dA\} \tag{5.58} $$

in which ${}^{t}_{o}[\sigma]$ is the 1st Piola-Kirchoff or Lagrange stress. This matrix is in general nonsymmetric since the reference areas on which the stresses act are not orthogonal. A symmetric stress matrix is obtained if the actual force vector, ${}^{t}\{dF\}$, is considered to be defined by a transformed pseudo force vector ${}^{o}\{dF\}$. Force ${}^{o}\{dF\}$ is assigned a transformation to ${}^{t}\{dF\}$ in the same manner that ${}^{t}\{dx\}$ transforms to ${}^{o}\{dx\}$, i.e.,

$$ {}^{t}\{dF\} = {}^{t}_{o}[J]\ {}^{o}\{dF\} \tag{5.59} $$

Substitution of the transformed force vector into (5.58) yields

$$ {}^{t}\{dF\} = {}^{t}_{o}[S]\ {}^{t}\{dA\} = {}^{t}_{o}|J|\ {}^{t}_{o}[J]^{-1}\ {}^{t}[\sigma]\ {}^{t}_{o}[J]^{-T}\ {}^{t}\{dA\} \tag{5.60} $$

in which ${}^{t}_{o}[S]$ represents the 2nd Piola-Kirchoff (2nd P-K) or "pseudo" stress referred to areas at t=0. The above transformation shows that this stress matrix is symmetric; however, it cannot be physically interpreted as stress in the usual manner since it resists a transformed force vector and does not act on deformed areas. Defining 6x1 stress vectors, ${}^{t}\{\sigma\}$ and ${}^{t}_{o}\{S\}$, and noting the similarity of the above equations with (5.39), the transformation in (5.60) may be written in vector form as

$$ {}^{t}_{o}\{S\} = {}^{t}_{o}|J|\ {}^{t}_{o}[T]^{-T}\ {}^{t}\{\sigma\} \tag{5.61} $$

$$ {}^{t}\{\sigma\} = {}^{t}_{o}|J|^{-1}\ {}^{t}_{o}[T]^{T}\ {}^{t}_{o}\{S\} \tag{5.62} $$

in which matrix ${}^{t}_{0}[T]$ was introduced in (5.39).

Now consider the configuration at time $t+\Delta t$ in which Cauchy stresses ${}^{t+\Delta t}\{\sigma\}$ act. These stresses may be referred to corresponding areas at $t=0$ or alternatively at time t as follows

$$ {}^{t+\Delta t}_{0}\{S\} = {}^{t+\Delta t}_{0}|J| \; {}^{t+\Delta t}_{0}[T]^{-T} \; {}^{t+\Delta t}\{\sigma\} \tag{5.63} $$

$$ {}^{t+\Delta t}_{t}\{S\} = {}^{t+\Delta t}_{t}|J| \; {}^{t+\Delta t}_{t}[T]^{-T} \; {}^{t+\Delta t}\{\sigma\} \tag{5.64} $$

in which ${}^{t+\Delta t}_{t}[T]$ was introduced in (5.57).

Stress rates and increments are employed in the development of path dependent, incremental constitutive models. The 2nd P-K rate is obtained by differentiating (5.60) with respect to time (or the pseudo load parameter). The formal stress increment is obtained by multiplying the rate by the time increment. The differentiation process parallels that used in (5.50) for the variational operator. Denoting a rate (as distinguished from a variation) by a dot superscipt, and using (5.26-5.30) with a dot replacing the variational operator, the 2nd P-K stress rate is given, after some manipulation, by

$$ {}^{t}_{0}[\dot{S}] = {}^{t}_{0}|J| \; {}^{t}_{0}[J]^{-1} \; {}^{t}_{t}[\dot{\sigma}_{T}] \; {}^{t}_{0}[J]^{-T} \tag{5.65} $$

in which

$$ {}^{t}_{t}[\dot{\sigma}_{T}] = {}^{t}[\dot{\sigma}] - {}^{t}[\sigma] \; [\dot{w}]^{T} - [\dot{w}] \; {}^{t}[\sigma] \tag{5.66} $$

$$ - {}^{t}[\sigma] \; [\dot{e}]^{T} - [\dot{e}]^{t} \; [\sigma] + Tr([\dot{e}]) \; {}^{t}[\sigma] $$

In (5.66), $[\dot{w}]$ and $[\dot{e}]$ are at time t. The term ${}^{t}[\dot{\sigma}]$ is the actual Cauchy stress rate. The combined terms in (5.66), denoted ${}^{t}_{t}[\dot{\sigma}_{T}]$, are referred to as the Truesdell stress rate in the literature. Because the

2nd P-K stress and, therefore, its rate are symmetric, the Truesdell rate is symmetric by virtue of the symmetric transformation in (5.65). In addition, the Truesdell stress rate transforms to an equivalent 2nd P-K rate in the same way that the Cauchy stress transforms to 2nd P-K stress, see (5.60). Under a rigid rotation, ${}^{\dagger}_{o}[\dot{S}]$ must vanish since no change of the basis unit vectors occurs. The 2nd P-K stresses are thus invariant under a rigid rotation. This requirement necessarily implies that ${}^{\dagger}_{\dagger}[\dot{\sigma}_T]$ must vanish under a rigid rotation, i.e.,

$$[0] = {}^{\dagger}[\dot{\sigma}] - {}^{\dagger}[\sigma]\,[\dot{w}]^T - [\dot{w}]\,{}^{\dagger}[\sigma] \tag{5.67}$$

Jaumann [5.7] adopted the above terms within the Truesdell stress rate as a suitable spin-invariant stress rate for use in constitutive laws at finite deformation, i.e.,

$${}^{\dagger}_{\dagger}[\dot{\sigma}_J] = {}^{\dagger}[\dot{\sigma}] - {}^{\dagger}[\dot{\sigma}]\,[\dot{w}]^T - [\dot{w}]\,{}^{\dagger}[\sigma] \tag{5.68}$$

In the case of a pure deformation increment, $[\dot{w}] = 0$ and the Jaumann rate is the actual Cauchy rate. The Jaumann stress rate is also symmetric. In vector form, the above expressions may be written

$${}^{\dagger}_{o}\{\dot{S}\} = {}^{\dagger}_{o}|J|\;{}^{\dagger}_{o}[T]^{-T}\;{}^{\dagger}_{\dagger}\{\dot{\sigma}_T\} \tag{5.69}$$

$${}^{\dagger}_{\dagger}\{\dot{\sigma}_T\} = {}^{\dagger}_{\dagger}\{\dot{\sigma}_J\} - {}^{\dagger}_{\dagger}[Q]\;{}^{\dagger}_{\dagger}\{\dot{e}\} \tag{5.70}$$

Matrix ${}^{\dagger}_{\dagger}[Q]$ is defined in the appendix to this chapter. The Jaumann stress rate finds use in finite strain plasticity as the quantity that is linearly related to the rate of the deformation matrix. As discussed by Hill [5.8], this is equivalent to a true stress vs. logarithimic strain relationship in simple tension. Use of the Jaumann stress rate

as the resultant of the constitutive model yields a zero Treusdell and 2nd P-K rate under rigid rotation.

In the course of finite element computations, the equilibrium configuration at time $t+\Delta t$ is sought using the equilibrium configuration at time t as the initial condition. Increments of the computed stress components are accumulated as the iterative solution converges toward the correct equilibrium configuration at $t+\Delta t$. In the T. L. approach, this process may be written as

$$ {}^{t+\Delta t}_{o}\{S\} = {}^{t}_{o}\{S\} + \Sigma\ {}_{o}\{\Delta S\} \tag{5.71} $$

As prevously noted, formal stress increments, $\{\Delta S\}$, are obtained from the corresponding stress rate multiplied by the time increment, Δt. In actual computations, the stress increments are determined from strain increments which are induced by a displacement increment $\{\Delta u\}$. The displacement increment can be thought of as a velocity multiplied by the time increment.

The total Cauchy stress at time $t+\Delta t$ may be computed using the transformation in (5.62). The incremental decomposition of stress above implies that material constitutive models produce increments of 2nd P-K stress. However, 2nd P-K stresses are generally unsuitable for material constitutive relationships which are naturally cast in terms of true stress-logarithmic strain. Use of a T. L. formulation necessitates some additional strain-stress transformations to obtain the ${}_{o}\{\Delta S\}$ needed in (5.71). These transformations are described in Section 5.6.

Equation (5.71) also serves as the starting point in developing an appropriate incremental stress decomposition for use with the U. L. approach. Both sides of (5.71) are multiplied by ${}^{t}_{o}|J|^{-1}\ {}^{t}_{o}[T]^{T}$ to yield

$$ {}^{t+\Delta t}_{t}\{S\} = {}^{t}\{\sigma\} + \Sigma\ {}_{t}\{\Delta S\} \tag{5.72} $$

in which ${}_{t}\{S\}$ are 2nd P-K stresses at time $t+\Delta t$ referred to the configuration at time t. The Truesdell stress increments (${}_{t}\{\Delta S\}$) appear naturally in this transformation. The inverse of (5.64) is applied to obtain the Cauchy stresses at time $t+\Delta t$ from (5.72). The above relationship is appealing from a physical viewpoint if the motion from t to $t+\Delta t$ is a simple rigid rotation. In such a case, the accumulated Truesdell increments in (5.72) are zero. The Cauchy stress at $t+\Delta t$, obtained using the inverse of (5.64), becomes ${}^{t}\{\sigma\}$ transformed through the rigid rotation, i.e., [T] is formed from an orthogonal deformation Jacobian.

5.5 Principle of Virtual Displacements

The principle of virtual displacements is applied to derive the equilibrium conditions for a body undergoing arbitrarily large displacements. The basis of the argument is that, for an imposed virtual displacement field, the virtual work remains invariant with respect to the configuration in which the variables are measured [5.9].

Consider a body at time $t+\Delta t$ which occupies a volume ${}^{t+\Delta t}V$ and is bounded by the surface area ${}^{t+\Delta t}A$. Let the body be in equilibrium under a set of body forces $\{p\}$ and surface tractions $\{q\}$. For dynamic analysis, the inertia effects are included in the body forces according to D'Alembert's principle. If a virtual displacement field $\{\delta u\}$ consistent with the kinematic boundary conditions is applied to the body, the virtual work done is given by (ignoring thermodynamic effects)

$$ {}^{t+\Delta t}\delta W = \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho \; \{\delta u\}^T \; {}^{t+\Delta t}\{\rho\} \; {}^{t+\Delta t}dV + \int_{{}^{t+\Delta t}A} \{\delta u\}^T \; {}^{t+\Delta t}_{t+\Delta t}\{q\} \; {}^{t+\Delta t}dA \tag{5.73} $$

and must be zero. The mass density is denoted by ρ. The body forces are in terms of unit mass; the surface tractions are in terms of the force per unit area at time $t+\Delta t$. The applied surface tractions $\{q\}$ are written in terms of the equilibrating Cauchy stresses. The classical procedure [5.9] is then employed to convert these integrals to the virtual work form using Gauss' theorem. The result is

$$ {}^{t+\Delta t}\delta W = {}^{t+\Delta t}\delta W_{ext} + {}^{t+\Delta t}\delta W_{int} = 0 \tag{5.74} $$

in which the external virtual work is given by

$$ {}^{t+\Delta t}\delta W_{ext} = \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\rho \; \{\delta u\}^T \; {}^{t+\Delta t}_{o}\{\rho\} \; {}^{t+\Delta t}dV + \int_{{}^{t+\Delta t}A} \{\delta u\}^T \; {}^{t+\Delta t}_{t+\Delta t}\{q\} \; dA \tag{5.75} $$

The internal virtual work is given by

$$ {}^{t+\Delta t}\delta W_{int} = - \int_{{}^{t+\Delta t}V} Tr(\; {}^{t+\Delta t}[\sigma] \; {}^{t+\Delta t}[\delta e] \;) \; {}^{t+\Delta t}dV \tag{5.76} $$

The internal virtual work measured from the instantaneous configuration involves the Cauchy stress and the variation of the deformation matrix (which is the linear part of the Almansi strain variation). The above expression may be simplified using vector forms of the strain and stress to yield

$$ {}^{t+\Delta t}\delta W_{int} = - \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\{\delta e\}^T \; {}^{t+\Delta t}\{\sigma\} \; {}^{t+\Delta t}dV \tag{5.77} $$

To apply this equation, the terms must be expressed as functions of variables in a known equilibrium configuration. The obvious choices are

the configuration at time 0 (T. L.) and time t (U. L.). To develop virtual work expressions for the Total Lagrangian formulation, all variables are referred to the undeformed configuration. Use is made of the following transformations previously derived

$$ {}^{t+\Delta t}\{\sigma\} = {}^{t+\Delta t}_{0}|J|^{-1}\ {}^{t+\Delta t}_{0}[T]^{T}\ {}^{t+\Delta t}_{0}\{S\} \tag{5.78} $$

$$ {}^{t+\Delta t}\{\delta e\} = {}^{t+\Delta t}_{0}[T]^{-1}\ {}^{t+\Delta t}_{0}\{\delta\varepsilon\} \tag{5.79} $$

$$ {}^{t+\Delta t}dV = {}^{t+\Delta t}_{0}|J|\ {}^{0}dV \tag{5.80} $$

Upon direct substitution of these transformations into (5.76), the internal virtual work becomes

$$ {}^{t+\Delta t}\delta W_{int} = -\int_{{}^{0}V} {}^{t+\Delta t}_{0}\{\delta\varepsilon\}^{T}\ {}^{t+\Delta t}_{0}\{S\}\ {}^{0}dV \tag{5.81} $$

The internal virtual work is in terms of the 2nd Piola-Kirchoff stresses and the variation of the Green strain. In [5.9] the above internal virtual work expression is derived by direct transformation of the integrals in (5.76) to time 0 before Gauss's theorem is applied.

To obtain the U. L. formulation, the terms in (5.76) are written as functions of the known equilibrium configuration at time t. The previously derived transformations are

$$ {}^{t+\Delta t}\{\sigma\} = {}^{t+\Delta t}_{t}|J|^{-1}\ {}^{t+\Delta t}_{t}[T]^{T}\ {}^{t+\Delta t}_{t}\{S\} \tag{5.82} $$

$$ {}^{t+\Delta t}\{\delta e\} = {}^{t+\Delta t}_{t}[T]^{-1}\ {}^{t+\Delta t}_{t}\{\delta\varepsilon\} \tag{5.83} $$

$$ {}^{t+\Delta t}dV = {}^{t+\Delta t}_{t}|J|\ {}^{t}dV \tag{5.84} $$

Upon substitution of these transformations into (5.76), the internal virtual work expression becomes

$$ {}^{t+\Delta t}\delta W_{int} = - \int_{{}^{t}V} {}^{t+\Delta t}_{t}\{\delta\varepsilon\}^T \, {}^{t+\Delta t}_{t}\{S\} \, {}^{t}dV \tag{5.85} $$

which is analogous to (5.81) except that integration is performed over the configuration at time t rather than time 0.

Transformation of the external virtual work integrals (5.75) to either T. L. or U. L. forms introduces another level of complexity. However, the volume integral for the body forces presents no difficulties and is readily transformed to

$$ {}^{t+\Delta t}\delta W^{BF}_{ext} = \int_{{}^{o}V} {}^{o}\rho \, \{\delta u\}^T \, {}^{t+\Delta t}\{p\} \, {}^{o}dV \tag{5.86} $$

for T. L. and to

$$ {}^{t+\Delta t}\delta W^{BF}_{ext} = \int_{{}^{t}V} {}^{t}\rho \, \{\delta u\}^T \, {}^{t+\Delta t}\{p\} \, {}^{t}dV \tag{5.87} $$

for U. L. These transformations are particularly simple since the directions of the body force components are not altered under the differential volume conversions. Body forces are conservative by definition. Inertia effects are simple to incorporate in (5.86) by the substitution of $\{-\ddot{u}\}$ for $\{p\}$. Similarly, for U. L., inertia effects are incorporated by transforming the integral in (5.87) to time 0 by noting that ${}^{t}\rho \, {}^{t}dV = {}^{o}\rho \, {}^{o}dV$ (conservation of mass) and letting $\{p\} = \{-\ddot{u}\}$. The transformed integral is identical to (5.86). This is of major practical consequence as the mass matrix for the U. L. formulation is identical to that for T. L. and remains constant for all times throughout the response.

The transformation of external virtual work for the applied surface tractions in (5.75) is more complex due to the different orientation of ${}^{t+\Delta t}dA$ when referred to times 0 and t. This dependence of external work on the displacements leads to non-conservative loading. An

interesting case of applied surface tractions is a normal pressure loading as considered by Oden [5.10], Nayak [5.1], and Mackay [5.11]. Thus,

$$ {}^{t+\Delta t}_{t+\Delta t}\{q\}\, {}^{t+\Delta t}dA = {}^{t+\Delta t}q_n\, {}^{t+\Delta t}\{n\}\, {}^{t+\Delta t}dA \tag{5.88} $$

where ${}^{t+\Delta t}\{n\}$ contains the components of a unit outward normal to the deformed area ${}^{t+\Delta t}dA$ and ${}^{t+\Delta t}q_n$ is the intensity of the applied normal pressure (suction is positive). Transforming (5.88) to time 0 using (5.4) yields

$$ {}^{t+\Delta t}q_n\, {}^{t+\Delta t}\{n\}\, {}^{t+\Delta t}dA = {}^{t+\Delta t}q_n\, {}^{t+\Delta t}_{0}|J|\, {}^{t+\Delta t}_{0}[J]^{-T}\, {}^{0}\{n\}\, {}^{0}dA \tag{5.89} $$

where ${}^{0}\{n\}$ is the unit outward normal at the same material point on the undeformed element surface. The surface traction integral of (5.75) becomes

$$ {}^{t+\Delta t}\delta W^{ST}_{ext} = \int_{{}^{0}A} \{\delta u\}^T\, {}^{t+\Delta t}_{0}[J]^{-T}\, {}^{0}\{q^*\}\, {}^{t+\Delta t}_{0}|J|\, {}^{0}dA \tag{5.90} $$

in which ${}^{0}\{q^*\}$ has been substituted for ${}^{t+\Delta t}q_n\, {}^{0}\{n\}$. The analogous expression for Updated Lagrangian is

$$ {}^{t+\Delta t}_{\Delta}W^{ST}_{ext} = \int_{{}^{t}A} \{\delta u\}^T\, {}^{t+\Delta t}_{t}[J]^{-T}\, {}_{t}\{q^*\}\, {}^{t+\Delta t}_{t}|J|\, {}^{t}dA \tag{5.91} $$

with ${}^{t}\{q^*\} = {}^{t+\Delta t}q_n\, {}^{t}\{n\}$. When the displacements represent large rotations and/or finite geometry changes, the use of (5.90) or (5.91) is essential to correctly generate residual loads in finite element computations. When only concentrated (non-follower) forces and body forces are applied, the surface integral above vanishes. Similarly, if the pressure loading is applied on a surface that undergoes infinitesimal displacements, equation (5.91) degenerates to (5.90) in

mechanics type problems with loading far removed from the intensely deformed plastic zone near the crack tip.

5.6 Constitutive Models for Finite Deformation Elasto-Plasticity

Constitutive models provide a connection between incremental strain and stress changes. There exist no generally applicable constitutive relations for materials subjected to finite strain magnitudes. However, two special cases have received considerable attention by finite element researchers and have resulted in reasonable engineering approximations for material behavior.

Oden [5.10] has reviewed formulations for nonlinear elastic (hyper-elastic) materials similar to rubber. For such materials, it may be possible to develop an elastic energy function in terms of the total Green strain components. The 2nd P-K stress is obtained by partial differentiation of the energy function with respect to strain. Thus,

$$W = W\left({}^{o}\{x\},\ {}^{t}_{o}\{\varepsilon\},\ \text{material constants}\right) \tag{5.92}$$

$${}^{t}_{o}\{S\} = \frac{\partial W}{\partial\, {}^{t}_{o}\{\varepsilon\}} \tag{5.93}$$

The relationship between increments of strain and stress is obtained by again differentiating (5.93) which yields

$$d\ {}^{t}_{o}\{S\} = {}^{t}_{o}[D]\ d\ {}^{t}_{o}\{\varepsilon\} \tag{5.94}$$

in which

$${}^{t}_{o}[D] = \frac{\partial^2 W}{\partial\, {}^{t}_{o}\{\varepsilon\}\ \partial\, {}^{t}_{o}\{\varepsilon\}^{T}} \tag{5.95}$$

The matrix ${}^{t}_{o}[D]$ contains the tangent elastic moduli. When the material is isotropic, the function W may be simplified through the use of strain invariants. The commonly used Mooney-Rivlin model for rubber assumes incompressible behavior with an energy function defined by the first and second strain invariants and two material constants.

The hyper-elastic material models are of little use for most engineering materials which experience permanent deformation when subjected to stress states outside an elastic domain. The remainder of this section focuses on an approximate theory for finite deformation, elasto-plasticity. This theory has been adopted by a number of finite element researchers, especially for applications involving the elastic-plastic fracture of metals. The fundamental assumptions of this theory are:

1. the material is initially isotropic;
2. the material work hardens isotropically;
3. there exists a linear relation between stress and strain increments;
4. additive decomposition of strain increments into elastic and plastic parts is valid;
5. recoverable elastic strains are infinitesimal;
6. the stress increment is independent of the rate of rigid body rotation.

The basis for this material modeling theory is provided in Hill's work [5.8]. More recently, Nemat-Nasser [5.12] reviewed the status of developments of finite deformation plasticity and provided additional arguments in support of assumption (4). His argument is that infinitesimal incremental deformations within a crystalline solid occurring from a finitely deformed configuration must satisfy additivity as in the infinitesimal deformation case. He further argued that the symmetric part of the velocity gradients (the deformation vector ${}^{t}_{t}\{\dot{e}\}$) may be decomposed into elastic and plastic components.

Given the above restrictions, the Prandtl-Reuss equations are adopted as usual, but in terms of true stress and true strain. Strain rate effects, creep, and thermal dependent material constants may be incorporated in this theory following the same techniques used in small strain plasticity, although little of this work has been reported in the literature.

Following Hill [5.8], Nemat-Nasser [5.12] and Hibbitt, et al. [5.13], the Jaumann stress increment and the increment of the deformation vector provide appropriate stress and strain increments which vanish under rigid body motion. The incremental relation becomes

$$ {}^{\dagger}_{\dagger}\{\Delta\sigma_J\} = {}^{\dagger}_{\dagger}[D_T]\ {}^{\dagger}_{\dagger}\{\Delta e\} \qquad (5.96) $$

in which ${}^{\dagger}_{\dagger}[D_T]$ is formed exactly as in the case of infinitesimal strain plasticity [5.14], but using true, Cauchy stresses. The above relationship is strictly valid for differential changes of the deformation vector. The von Mises yield criterion and associated flow rule are adopted for metals. For stress states inside the yield surface, the tangential constitutive matrix, ${}^{\dagger}_{\dagger}[D_T]$, is composed of simple elastic constants. For an elastic-plastic state, the terms of ${}^{\dagger}_{\dagger}[D]$ are derived from the elastic constants, the current stress state, the material strain hardening characteristics, and the history of Cauchy stresses. For uniaxial tension, it is simple to show that (5.96) represents a true stress vs. logarithmic strain relationship. Numerical refinements, such as the subincrement method, used in small strain plasticity are equally applicable in (5.96) to assure satisfaction of the flow rule.

The incremental relation in (5.96) is adopted in both U. L. and T. L. formulations through appropriate transformation of the terms to the re-

quired reference configuraiton. Considering first the T. L. formulation, the deformation increment is obtained from the Green strain increment using (5.54).

$$ {}^{t}_{t}\{\Delta e\} = {}^{t}_{o}[T]^{-1}\ {}^{t}_{o}\{\Delta\varepsilon\} \tag{5.97} $$

The assumption here is that the Green strain increment, ${}^{t}_{o}\{\Delta\varepsilon\}$, is actually of differential, not finite, magnitude (see equations 5.50 - 5.55). The resulting Jaumann stress increments obtained from (5.96) are transformed to Truesdell increments with (5.70).

$$ {}^{t}_{t}\{\Delta\sigma_T\} = {}^{t}_{t}\{\Delta\sigma_J\} - {}^{t}_{t}[Q]\ {}^{t}_{t}\{\Delta e\} \tag{5.98} $$

The Truesdell increment is transformed to the required 2nd P-K increment with (5.69).

$$ {}^{t}_{o}\{\Delta S\} = {}^{t}_{o}|J|\ {}^{t}_{o}[T]^{-T}\ {}^{t}_{t}\{\Delta\sigma_T\} \tag{5.99} $$

To compute a tangential stiffness, matrix, it is necessary to have a direct relationship between increments of 2nd P-K stress and Green strain in the form

$$ {}^{t}_{o}\{\Delta S\} = {}^{t}_{o}[D_T]\ {}^{t}_{o}\{\Delta\varepsilon\} \tag{5.100} $$

Combining (5.96-5.99), the tangent material matrix is given by

$$ {}^{t}_{o}[D_T] = {}^{t}_{o}|J|\ {}^{t}_{o}[T]^{-T}\ [\ {}^{t}_{t}[D_T] - {}^{t}_{t}[Q]\ ]\ {}^{t}_{o}[T]^{-1} \tag{5.101} $$

However, because ${}^{t}_{t}[Q]$ is not symmetric, the resulting constitutive relation is non-symmetric which leads to an undesirable non-symmetric stiffness matrix. Osias [5.15] arrived at this same relationship (5.101) and retained the non-symmetric terms. Upon examination of the [Q] matrix, the

non-symmetric terms are found to arise from the trace of the deformation matrix which represnts the incremental volume change, see Eq. (5.66). If these terms are neglected, then ${}^{t}_{t}[Q]$ becomes symmetric. Recalling assumption (5), the rationale to neglect these terms is that: (1) under elastic conditions, the volume change multiplied by the stresses is a very small term compared to the elastic moduli, and (2) under perfect plasticity conditions the incompressibility constraint forces a zero volume change. McMeeking and Rice [5.16] omitted the trace term as did Nagtegaal and de Jong [5.17], although neither group noted this assumption. Further complications arise when the material has a very low hardening modulus (in terms of true stress-logarithmic strain). The intermediate result, ${}^{t}_{t}[D_T] - {}^{t}_{t}[Q]$, may develop negative terms on the diagonal for large tensile strains. One possible remedy is to simply omit ${}^{t}_{t}[Q]$ in (5.101) when forming a suitable constitutive relation for use in computing a tangent stiffness. The writers have demonstrated that this technique is acceptable provided the exact stress increment is computed using (5.96-5.99).

Consider now constitutive relations for use with the U. L. formulation. The deformation increment at time $t+\Delta t$ for use in (5.96) is computed from the Green strain increment referred to the configuration at time t using (5.57).

$$ {}^{t+\Delta t}_{t+\Delta t}\{\Delta e\} = {}^{t+\Delta t}_{t}[T]^{-1}\ {}^{t+\Delta t}_{t}\{\Delta \varepsilon\} \tag{5.102} $$

Again, it is implied that ${}^{t+\Delta t}_{t}\{\Delta \varepsilon\}$ is actually of differential magnitude. The Jaumann stress increment is obtained from (5.96) and transformed to a Truesdell increment by

$$ {}^{t+\Delta t}_{t+\Delta t}\{\Delta \sigma_T\} = {}^{t+\Delta t}_{t+\Delta t}\{\Delta \sigma_J\} - {}^{t+\Delta t}_{t+\Delta t}[Q]\ {}^{t+\Delta t}_{t+\Delta t}\{\Delta e\} \tag{5.103} $$

The Truesdell increment is transformed to an increment of 2nd P-K stress referred to the configuration at time t (which is just the Truesdell stress transformed to time t)

$$ {}^{t+\Delta t}_{t}\{\Delta S\} = {}^{t+\Delta t}_{t}|J| \; {}^{t+\Delta t}_{t}[T]^{-T} \; {}^{t+\Delta t}_{t+\Delta t}\{\Delta \sigma_T\} \tag{5.104} $$

In U. L., the tangent stiffness matrix is formed for the configuration at time t and generally held constant during the increment Δt. Thus, the material tangent matrix is given by

$$ {}^{t}_{t}[D^*_T] = {}^{t}_{t}[D_T] - {}^{t}_{t}[Q] \tag{5.105} $$

in which both matrices on the right side are generated in terms of actual Cauchy stresses at time t. The ${}^{t}_{t}[D^*]$ above relates increments of Green strain and Truesdell stress at time t. If the stiffness is updated between t and $t+\Delta t$ (to accelerate convergence) the new constitutive matrix is evaluated as in (5.105) using the most current values of Cauchy stress. The same difficulty that arises due to the non-symmetry of $[Q]$ for T. L. also occurs here for U. L.

Approximating assumptions for (5.96-5.105) have been introduced by various investigators as summarized below. Alturi, et. al. [5.18] employed an U. L. approach with the approximation

$$ {}^{t+\Delta t}_{t}\{\Delta S\} = {}^{t+\Delta t}_{t+\Delta t}[D_T] \; {}^{t+\Delta t}_{t}\{\Delta \varepsilon\} \tag{5.106} $$

which neglects changes in the structural configuration over the interval Δt. In forming the constitutive matrix, $[Q]$ was simply ignored. Hibbitt, Marcal, and Rice [5.13] fully developed the T. L. equations (5.96-5.101) but implemented a simplified form valid only for small strains by neglecting $[Q]$ in <u>all</u> equations. The determinant of the

deformation Jacobian was also assumed to remain unity which effectively reduces [T] to a rotational transformation. Based on these assumptions, they claim that the implemented formulation is valid for the case of large rotations but infinitesimal strains. McMeeking and Rice [5.16] developed an U. L. approach in which the configuration is continuously updated. Under such a procedure, the transformation in (5.102) is unnecessary, i.e., [T] is an identity matrix. In addition, the transformation of Truesdell stress in (5.104) becomes unnecessary. Bathe, et al. [5.5] presented both T. L. and U. L. formulations. For T. L., the material constitutive relation in (5.100) was formulated using classical plasticity theory but in terms of total 2nd P-K stresses. The proposed U. L. constitutive model (5.106) is that adopted by Atluri, et al. [5.18].

With exception of the Jaumann to Truesdell transformation in (5.98), differences in the various U. L. approaches vanish for sufficiently small motion over the interval Δt. However, the choice of a "sufficiently" small Δt cannot be assured for complex structures undergoing large displacements; thus, the more exact transformation in (5.104) is preferred. With this approach, the T. L. formulation is recovered as the limiting case when no updates of the deformed configuration are performed. The use of (5.100) with ${}^{t}_{o}[D_T]$ formed in terms of 2nd P-K stresses, as proposed in [5.5], is clearly unacceptable for general applications.

There continues to exist some question as to the need for the [Q] terms appearing in the various transformations. These terms arise from the Jaumann to Truesdell stress increment transformation. In the elastic range, these terms are on the order of stress, which when multiplied by a strain increment are clearly negligible compared with elastic moduli. In the plastic range, the terms of [Q] become more comparable in magnitude to

the plastic moduli (which may be on the same order as the stress) unless the deformation increment is truly of differential magnitude. Hibbitt, et al. [5.13] acknowledges the role of [Q] then chose to ignore it under the restructions of small strain-large rotation. McMeeking and Rice [5.16] include [Q] in their continuously updated formulation as an initial stress matrix and do not report any numerical difficulties. They conclude that whenever the slope of the Cauchy (true) stress vs. logarithmic strain curve has a magnitude comparable to the current stress level, the predicted tangent stiffness cannot be accurate. Both [5.13] and [5.16] conclude that the relative importance of [Q] terms is not fully known and requires further study.

5.7 Summary and Comparisons of the U.L. and T.L. Formulations

Using matrix notation, the equations of nonlinear continuum mechanics that provide a rational basis for finite element analysis have been presented in this chapter. Two formulations are described in detail; namely, the Total Lagrangian (T. L.) and Updated Lagrangian (U. L.) The initial configuration of the body at time $t=0$ serves as the reference state for all variables in the T. L. formulation. The configuration at time t serves as the reference state to describe all variables at time $t+\Delta t$ in the U. L. formulation. Full details of the stress measures, rates, transformations and nonlinear constitutive models have been described that enable analyses involving both infinitesimal and finite strain magnitudes. Finite magnitude strains commonly appear in structures that experience transient response after being subjected to localized blast or impact loadings.

The two nonlinear formulations, T. L. and U. L., are shown to derive from a common definition of the rate of work per unit mass which leads to equivalent virtual work expressions when the corresponding reference states are introduced. Analyses derived from each formulation are identical to within truncation errors introduced in the numerical procedures. However, as noted throughout this chapter, various investigators have introduced simplifying approximations into the formulations that limit their applicability to the most general class of problems. The U. L. formulation offers the most temptation to simplify the equations since one might argue that the difference in configuration between time step t and $t+\Delta t$ is sufficiently small that all second order nonlinear terms may be discarded. Stress rates and transformations in the corresponding constitutive models are simplified similarly. No approximations of any type are permissible in the T. L. formulation since the deformed and reference configuration are seldom similar.

The differences in computational efficiency of the two formulations lies in the stress rate transformations and the strain-displacement relations. The mass representation is identical in both formulations and thus does not enter into the discussion. Similarly, computations associated with a nonlinear constitutive model are identical as the computed results required by the model are identical in both formulations. As shown in Section 5.6, the same number and type of transformations to account for stress rates are required in each formulation. The U. L. approach appears to gain a computational advantage if the simplifying approximation is employed that eliminates the Truesdell to 2nd P-K stress transformation. However, this advantage is offset by two factors: (1) the number of time steps to assure that the change in configuration over a single time step

is "small" remains unknown, and (2) the stresses at the beginning of the step and the increments that occur during the step must be transformed to the configuration at the end of the step. This last process is unnecessary for the T. L. formulation in which all stress increments are transformed to the initial configuration as computed. The stress increments are simply added to the existing stress state. Therefore, in terms of stress computation, the T. L. approach actually has a slight efficiency advantage over the exact U. L. formulation.

The computation of an increment of the deformation vector is accomplished in the same symbolic form (5.53 and 5.57) for both formulations. The differences lie in the effort required to construct the [T] matrix in each case. For U. L., the formulation of derivatives in [T] requires that updated coordinates be available, whereas in T. L., derivatives are always formed with respect to the initial configuration. These derivatives could be computed once and saved for re-use. Some effort is necessary to continuously update the coordinates. More importantly, the strain increments produced by U. L. cannot be simply accumulated to form the total strain, as shown in (5.48). This transformation to a common reference configuration requires additional computational effort that is unnecessary in the T. L. formulation.

In summary, an examination of the computational efficiency of the transformations necessary in each approach, reveals that T. L. appears to have a slight advantage over U. L. This does not include any factors that are introduced when the governing equations are cast into a form suitable for finite element analysis. In a practical sense, the advantages of either the U. L. or T. L. continuum theory appear computationally

insignificant given other costs of computation, for example, equation solving and massive data transfers between memory and disk.

5.8 References

5.1 Nayak, G. C., "Plasticity and Large Deformation Problems by the Finite Element Method," unpublished Ph.D. Thesis, University of Wales, Swansea, 1971 (C/PH/15/1971).

5.2 Zienkiewicz, O.C., The Finite Element Method in Engineering Science, 3rd edition, McGraw-Hill, London, 1978.

5.3 Bathe, K. J., Ramm, E., and Wilson, E. L., "Finite Element Formulations for Large Deformation Dynamic Analysis," Int. J. for Num. Meth. in Engr., Vol. 9, 1975, pp. 353-386.

5.4 Yaghmai, S. and Popov, E. P., "Incremental Analysis of Large Deflections of Shells of Revolution," Int. J. Solids and Structures, Vol. 7, 1971, pp. 1375-1393.

5.5 Bathe, K. J. and Ozdemir, H., "Elastic-Plastic Large Deformation Static and Dynamic Analysis," Computers and Structures, Vol. 6, 1976, pp. 81-92.

5.6 Boit, M. A., Mechanics of Incremental Deformations, Wiley, N.Y., 1965.

5.7 Prager, W., Introduction to Mechanics of Continuna, Gin and Co., 1961.

5.8 Hill, R., "On Constitutive Inequalities for Simple Materials," J. Mech. Phys. Solids, Vol. 16, 1968, pp.229-242.

5.9 Dym, C. L. and Shames, I., Solid Mechanics: A Variational Approach, McGraw-Hill, 1973.

5.10 Oden, J. T., "Numerical Formulation of Nonlinear Elasticity Problems," J. Struct. Div. Am. Soc. Civ. Engr., Vol. 93, 1967, pp. 235-255.

5.11 Mackay, D. C., "Finite Element Formulations for Large Strain Plasticity Problems," unpublished M.S. thesis, University of Wales, Swansea (C/M/146/80).

5.12 Nemat-Nasser, S., "On Finite Deformation Elasto-Plasticity," Int. J. Solids and Structures, Vol. 18, 1982, pp. 857-872.

5.13 Hibbitt, H. D., Marcal, P. V., and Rice, J. R., "A Finite Element Formulation for Problems of Large Strain and Large Displacement," Int. J. Solids and Structures, Vol. 6, 1972, pp. 1069-1086.

5.14 Nayak, G. C. and Zienkiewicz, O. C., "Elasto-Plastic Stress Analysis, A Generallization for Various Constitutive Relations Including Strain Softening," Int. J. Num. Meths. Engr., Vol. 5,1972, pp. 113-135.

5.15 Osias, J. R. and Swedlow, J. L., "Finite Elasto Plastic Deformation - I: Theory and Numerical Examples," Int. J. Solids and Structures, Vol. 10, 1974, pp. 321-339.

5.16 McMeeking, R. M. and Rice, J. R., "Finite Element Formulations for Problems of Large Elastic-Plastic Deformation," Int. J. Solids and Structures, Vol. 11, 1975, pp. 601-616.

5.17 Nagtegaal, J. C. and de Jong, T. E., "Some Computational Aspects of Elastic-Plastic Large Strain Analysis," Report No. M.S.10.910. MARC Analysis Research Corp.

5.18 Alturi, Nakagaki, and Chea, "Fracture Analysis Under Large-Scale Plastic Yielding," Flaw Growth and Fracture, ASTM STP631, American Society of Testing Materials, 1977, pp. 42-61.

5.9 Appendix -- Notation

The following convention for vector and matrix subscripts and superscripts is used:

1. a left superscript indicates the discrete time configuration in which the variable occurs,
2. a left subscript in conjunction with a left superscript indicates the discrete time configuration with respect to which the variable is measured,
3. a left subscript by itself indicates an increment from time t to $t+\Delta t$ referred to the configuration at the specified time,
4. a dot over a symbol denotes a rate quantity. Left subscripts and superscripts indicate the time and reference configuration,
5. a "δ" symbol denotes a variation of the quantity. Left subscripts and superscripts indicate the time and reference configuration.

$[\ ]$	Square brackets denote a matrix
$\{\ \}$	Curly braces denote a column vector
$[\ ]^T$, $\{\ \}^T$	Right superscript "T" denotes the transpose
${}^0\{x\}$, ${}^t\{x\}$, ${}^{t+\Delta t}\{x\}$	Cartesian coordinates at time 0, t, and $t+\Delta t$
$\{i\}$	unit vectors
${}^t_0[J]$	Deformation Jacobian matrix
${}^t\{u\}$, ${}^{t+\Delta t}\{u\}$	Vectors of displacements from 0 to t and from 0 to $t+\Delta t$
${}^0\{n\}$, ${}^t\{n\}$	Vector components of the surface normal at times 0, t
${}^0\rho$, ${}^t\rho$	Mass density at times 0, t
${}^0A$, ${}^tA$	Area at times 0, t

${}^{0}V$, ${}^{t}V$	Volume at time 0 and t
${}^{t}_{0}\{q\}$, ${}^{t}_{t}\{q\}$	Vectors of surface force components at time t per unit area at 0 andd t
${}^{t}_{0}\{p\}$	Vector of body force components per unit mass at time t referenced to configuration at 0
${}^{t+\Delta t}\{\sigma\}$, ${}^{t+\Delta t}[\sigma]$	Cauchy stress vector and matrix at time t+Δt. (Note that ${}^{t+\Delta t}[\sigma] = {}^{t+\Delta t}_{t+\Delta t}[\sigma]$).
${}^{t}_{0}\{S\}$, ${}^{t}_{0}[S]$	2nd Piola-Kirchoff stress vector and matrix at time t referenced to the configuration at 0
${}^{t+\Delta t}_{t}\{S\}$, ${}^{t+\Delta t}_{t}[S]$	2nd Piola-Kirchoff stress vector and matrix at time t+Δt, referred to configuration at t
${}^{t}_{t}\{\dot{\sigma}_{T}\}$, ${}^{t}_{t}[\dot{\sigma}_{T}]$	Truesdell stress vector and matrix at time t referred to the configuration at t
${}^{t}_{t}\{\dot{\sigma}_{J}\}$, ${}^{t}_{t}[\dot{\sigma}_{J}]$	Rates of Jaumann stress vector and matrix at time t referred to the configuration at t
${}^{t+\Delta t}\{\varepsilon\}$, ${}^{t+\Delta t}[\varepsilon]$	Almansi strain vector and matrix at time t+Δt (note that ${}^{t+\Delta t}[\varepsilon] = {}^{t+\Delta t}_{t+\Delta t}[\varepsilon]$).
${}^{t+\Delta t}_{0}\{\varepsilon\}$, ${}^{t+\Delta t}_{0}[\varepsilon]$	Green strain vector and matrix at time t+Δt referenced to configuration at 0
${}^{t}_{t}[v]$	Incremental displacement gradient matrix at t referenced to configuration t
${}^{t}_{t}[e]$	Deformation matrix at t referred to configuration t (symmetric part of ${}^{t}_{t}[v]$)
${}^{t}_{t}[w]$	Spin matrix at t referred to configuration t (anti-symmetric part of ${}^{t}_{t}[v]$)
${}^{t}_{0}[T]$	Transformation matrix to convert Almansi strains at time t to Green strains

${}^{t}_{o}[j]$	Displacement Jacobian at time t referred to configuration at 0
${}^{t}_{o}[B]$	Differential operator to yield Green strain increments from displacement increments
Tr( )	Denotes the trace of a square matrix (sum of diagonal terms)
${}^{t}_{t}[Q]$	Transformation matrix that converts Jaumann stress increments at time t to Truesdell stress increments at time t
δW	Virtual work quality
W	Elastic energy density function
${}^{t}_{o}[D]$	Elasticity matrix at time t referred to configuration at 0
${}^{t}_{t}[D_T]$	Elastic-plastic matrix at time t referred to configuration at t

Key Transformation Matrices (2-D Case)

$$ {}^{\dagger}_{o}[J] = \left[\begin{array}{c|c} \bar{X}_x & \bar{X}_y \\ \hline \bar{Y}_x & \bar{Y}_y \end{array}\right] ; \quad \begin{array}{l} \bar{x} = x + u \\ \bar{y} = y + v \end{array} ; \quad \bar{x}_x = \frac{\partial \bar{x}}{\partial x} $$

$$ {}^{\dagger}_{o}[T] = \left[\begin{array}{c|c|c} J_{11}^2 & J_{21}^2 & J_{11}J_{21} \\ \hline J_{12}^2 & J_{22}^2 & J_{12}J_{22} \\ \hline 2J_{11}J_{12} & 2J_{21}J_{22} & J_{11}J_{12} + J_{12}J_{21} \end{array}\right] $$

$$ {}^{\dagger}_{\dagger}[Q] = \left[\begin{array}{c|c|c} \sigma_{11} & -\sigma_{11} & \sigma_{12} \\ \hline -\sigma_{22} & \sigma_{22} & \sigma_{12} \\ \hline 0 & 0 & \frac{1}{2}(\sigma_{11} + \sigma_{22}) \end{array}\right] ; \quad \sigma_{ij} = {}^{\dagger}_{\dagger}\sigma_{ij} $$

$$ d\,{}^{\dagger}_{o}\{\varepsilon\} = d \begin{Bmatrix} \varepsilon_x \\ \varepsilon_y \\ \gamma_{xy} \end{Bmatrix}^{\dagger}_{o} = {}^{\dagger}_{o}[B]\, d\begin{Bmatrix} u \\ v \end{Bmatrix} $$

$$ \delta\,{}^{\dagger}_{o}\{\varepsilon\} = {}^{\dagger}_{o}[B]\,\delta\{u\} $$

$$
{}^{\dagger}_{o}[B] = \begin{bmatrix} \bar{x}_x \frac{\partial}{\partial x} & \bar{y}_x \frac{\partial}{\partial x} \\ \bar{x}_y \frac{\partial}{\partial y} & \bar{y}_y \frac{\partial}{\partial y} \\ \bar{x}_y \frac{\partial}{\partial x} + \bar{x}_x \frac{\partial}{\partial y} & \bar{y}_y \frac{\partial}{\partial x} + \bar{y}_x \frac{\partial}{\partial y} \end{bmatrix}
$$

CHAPTER 6

NONLINEAR FINITE ELEMENT EQUATIONS

6.1 General

The finite element concept of discretization is combined with the continuum mechanics theory in matrix form in this chapter. With the addition of solution procedures for nonlinear transient equations, a very general analysis capability results. The form of the finite element equations of motion are first expressed without regard to a particular choice of reference configuration. The use of such a general form makes the detailed discussion of nonlinear transient solution procedures equally applicable for an arbitrary reference configuration. Specific forms of the element stiffness matrices and internal resisting force vectors are derived for the Total and Updated Lagrangian approaches. The relative computational effort required for each approach and the implications of a substructured modeling procedure are also examined. Recommendations are made for appropriate formulations and solution procedures necessary to support the analysis of a broad class of problems.

A number of investigators have contributed to the current state of rigorous finite element formulations derived from the continuum theory of the previous chapter. Three approaches appeared almost simultaneously in the literature of the early 1970's. Hibbitt, Marcal, and Rice [6.1] first presented a comprehensive formulation using the configuration at time t=0 as the reference state (Total Lagrangian).

Matrices required for finite element analysis, including the non-symmetric stiffness for displacement dependent loading, were described. Using matrix notation throughout, Nayak [6.2] independently derived the same element matrices. Both of these studies addressed only static analysis and employed the symmetric 2nd P-K stress and Green strain as conjugate measures in the configuration at time t=0. Nemat-Nasser and Shatoff [6.3] used a combination of current (time t) and initial reference configurations with the non-symmetric 1st P-K stress measure.

Investigators advocating the use of the current configuration (convected coordinates) for the reference state included Hartzman and Hutchinson [6.4], Belytschko and Hsieh [6.5], and Key [6.6]. In each of these studies, the motion of the body over Δt was decomposed into a rigid motion and a pure deformation. The procedures worked well in each case even though different schemes were devised for the decomposition procedure. Only Key [6.6] focused on the finite strain case; the other two studies were concerned primarily with large rotation effects. Interestingly, each of these studies addressed the problem of nonlinear wave propagation using explicit integration procedures. McMeeking and Rice [6.7] presented a rigorous formulation based on a continuously updated reference configuration applicable for finite deformation and rotation.

Yaghmai and Popov [6.8] were the first to describe an Updated Lagrangian approach which attempts to combine the Total Lagrangian and the updated coordinate approaches. This was done by adopting the configuration at any time t prior to $t+\Delta t$ as a reference state. The existing Green strains and 2nd P-K stresses in the reference

configuration are simply treated as initial strains and stresses to which are added increments of 2nd P-K stress and Green strain. This approach captures the spirit of a convected coordinate approach but is more appealing for four reasons: (1) the reference configuration does not need to be continuously updated, (2) the decomposition of motion into deformation and rotation follows Lagrangian mechanics, (3) the Total Lagrangian approach is recovered exactly if the reference configuration is not updated, and (4) no limits on the deformation magnitude over Δt are implied due to linearization if the exact U. L. equations of the previous chapter are utilized. Nagarajan and Popov [6.9] subsequently used the U. L. approach to study viscoplastic response of thin shell structures. Bathe, et. al. [6.10, 6.11] formalized the U. L. procedure for general transient analysis and published a large number of papers using the procedure.

6.2 Nonlinear Equations of Motion

The virtual work principles derived in Section 5.4 provide the basis to generate approximate equations of motion using finite element concepts. Considering the motion of the body over the interval from t to $t+\Delta t$, and without regard to a particular reference coordinate system, the principal of virtual work provides that at $t+\Delta t$ (see 5.74-5.76)

$$\delta W = \delta W_{ext} + \delta W_{int} = 0 \tag{6.1}$$

where the $t+\Delta t$ left superscript has been dropped in this section for brevity. All quantities are at time $t+\Delta t$ unless otherwise noted.

The internal and external virtual work terms are given symbolically by

$$\delta W_{int} = -\int_{*V} {}_{*}\{\delta\varepsilon\}^{T}\,{}_{*}\{\sigma\}^{*}\,dV \tag{6.2}$$

$$\delta W_{ext} = \int_{*V} {}_{*}\rho\{\delta u\}^{T}\{p\}^{*}\,dV + \int_{*A}\{\delta u\}^{T}\,{}_{*}\{q\}^{*}\,dA \tag{6.3}$$

The asterik (*) is adopted to indicate an unspecified reference configuration for the variables. As shown in Section (5.4), the general forms of the virtual work equations for both T. L. and U. L. are identical. The differences derive from the choice of reference system (*) and the requirement of conjugate strain and stress definitions. This notation permits a completely general discussion of the finite element process and solution techniques prior to introducing specific matrices for the T. L. and U. L. approaches.

The finite element concept is invoked at this point to provide a spatial discretization of the structure. Individual elements are connected at discrete points termed nodes. At time $t+\Delta t$, the continuous displacement field, $\{u\}$, within each element is approximated by a set of interpolation functions, $[N]$, which act on the nodal displacements $\{a^{e}\}$. Thus,

$$\{u\} = [N]\,\{a^{e}\} \tag{6.4}$$

In 2-D problems, $\{u\}$ is a 2 x 1 vector containing u and v displacement components as a function of position within the element. Vector $\{a^{e}\}$ then contains (2 * number of element nodes) terms. Terms of $[N]$ are usually simple functions of position within the element and do not depend on its deformed shape or time.

To facilitate the development of elements with initially curved edges and faces, the shape functions are expressed in terms of an intrinsic, convected coordinate system. A one-to-one mapping from the parent configuration (usually square) to the actual shape in the structure is provided by a set of interpolating functions and the Cartesian coordinates of the element nodes. As will be shown, the use of initial or updated Cartesian coordinates for this mapping plays a major role in the differences between T. L. and U. L. When interpolation functions for the element mapping and those for the displacement interpolation are identical (6.4), the popular isoparametric element family is obtained. Fortunately, the details of transformations between the intrinsic isoparametric coordinate and global Cartesian coordinates do not affect the basic finite element solution procedure.

At time $t+\Delta t$ virtual displacements, $\{\delta a^e\}$, consistent with the kinematic constraints are imposed on the nodes. Corresponding variations of the element displacement field, $\{\delta u\}$, are given by

$$\{\delta u\} = [N]\ \{\delta a^e\} \tag{6.5}$$

Virtual strains caused by the virtual displacement field are denoted ${}_*\{\delta\varepsilon\}$, in which the definition of strain appropriate for the selected reference configuration is implied. However, it is completely general to express the virtual strains as

$${}_*\{\delta\varepsilon\} = {}_*[L]\ \{\delta u\} \tag{6.6}$$

in which ${}_*[L]$ is a differential operator matrix that may be a nonlinear function of the deformed element configuration at time $t+\Delta t$. Combining (6.5) and (6.6) yields

$$ {}_*\{\delta\varepsilon\} = {}_*[L]\ [N]\ \{\delta a^e\} = {}_*[B]\ \{\delta a^e\} \tag{6.7} $$

in which the conventional notation ${}_*[B]$ of the virtual strain-nodal displacement relationship is introduced. The general form of the internal virtual work for an element is written

$$ \delta W^e_{int} = -\{\delta a^e\}^T \int_{*V} {}_*[B]^T {}_*\{\sigma\}^* dV_e = -\{\delta a^e\}^T \{F^e\} \tag{6.8} $$

in which the notation $\{F^e\}$ is introduced to denote equivalent nodal forces induced by the deformed element. The (*) left subscript is not required on such terms since these forces are directed along global coordinate axes.

Consider now the virtual work of body forces within an element,

$$ \delta W^e_{BF} = \int_{*V} {}_*\rho\ \{\delta u\}^T \{p\}^* dV_e \tag{6.9} $$

in which ${}_*\rho$ is the mass density and $\{p\}$ are the Cartesian components of body force per unit mass. Two types of body forces are considered: (1) self-weight and centrifugal, in which the body force intensity is written in terms of current density as ${}_*\{F\}$; (2) inertia effects using D'Alembert's principle with $\{\ddot{u}\}$ defining the element acceleration field. Using (6.4), the body force virtual work may be written in the form

$$ \delta W^e_{BF} = \{\delta a^e\}^T \int_{*V} [N]^T {}_*\{F\}^* dV_e - \{\delta a^e\}^T \int_{*V} {}_*\rho [N]^T \{\ddot{u}\}^* dV_e \tag{6.10} $$

To simplify this expression, let $\{BF^e\}$ represent the first integral (equivalent nodal loads due to body force). Again the (*) left subscript can be omitted. The second integral is simplified by noting that in analogy with (6.4)

$$\{\ddot{u}\} = [N]\{\ddot{a}^e\} \tag{6.11}$$

and that the second integral of (6.10) must be invariant with the deformed configuration (conservation of mass--see Section 5.4). The virtual work of element inertia forces may thus be written in the form

$$\delta W_I^e = -\{\delta a^e\}^T [M^e]\{\ddot{a}^e\} \tag{6.12}$$

in which

$$[M^e] = \int_{{}^oV} [N]^T [N]\, {}^o\rho\, {}^o dV \tag{6.13}$$

Matrix $[M^e]$ is usually termed the consistent element mass; it is computed once at the beginning of the solution for the initial configuration and recalled from secondary storage whenever needed.

In a similar manner, the virtual work of external tractions applied over the element surfaces may be written in the form

$$\delta W_{ST}^e = \{\delta a^e\}^T \int_{{}^*A} [N]^T {}^*\{q\}^*\, dA = \{\delta a^e\}^T \{TF^e\} \tag{6.14}$$

The total virtual work for an element is thus written

$$\delta W^e = \{\delta a^e\}^T \left\{-\{F^e\} + \{BF^e\} + \{TF^e\} - [M^e]\{\ddot{a}_e\}\right\} \tag{6.15}$$

Virtual work for the complete structure is obtained by summing the contributions from each element. Element nodal displacements are related to the structural nodal displacement vector through a simple

Boolean connectivity matrix, which accomplishes the symbolic assembly process. The resulting virtual work expression is analogous to (6.15) except that vectors are of structural, rather than elemental, size.

$$\delta W = 0 = \{\delta a\}^T \left\{-\{F\} + \{BF\} + \{TF\} - [M]\{\ddot{a}\}\right\} \tag{6.16}$$

The choice of nodal virtual displacements is arbitrary; thus the summed terms in { } must vanish for dynamic equilibrium at time $t+\Delta t$. The result is a set of nonlinear, simultaneous equations in the nodal displacements and accelerations which may be written in the form

$$\{R\} = \{P\} - \{F\} - [M]\{\ddot{a}\} = \{0\} \tag{6.17}$$

Vectors {F} and {P} are implicit functions of the nodal displacements and generally cannot be written as a matrix multiplied into the current displacement vector. For simplicity, the external applied load effects are combined into {P}. Likewise, damping effects have been neglected; however they could be included by a term of the form $[C]\{\dot{a}\}$ analagous to the inertia term.

At this point it is instructive to review the sources of nonlinearity in the dynamic equations of motion (6.17):

1) the change in surface orientation and magnitude under loading causes {P} to become a function of the displacements;
2) the internal resisting force vector, {F}, may be nonlinear due to the material constitutive relationship between total stress and strain ($_*\{\sigma\}$ and $_*\{\varepsilon\}$);
3) the internal resisting force vector may be nonlinear due to the dependence of $_*[B]$ on the nodal displacements.

The following section considers general solution procedures.

6.3 Transient Analysis with Substructuring

At any given time, 0, Δt, $2\Delta t$,.....t, $t+\Delta t$..... the nodal displacements, velocities, and accelerations are sought such that the nonlinear equilibrium equations represented by (6.17) are satisfied. A temporal integration operator is employed for the time discretization. Both implicit and explicit operators are applicable. With explicit operators, computations are performed directly on the vector form of the dynamic equations in (6.17) with {R} taken as zero. Component terms of each vector are computed directly from the integrals in (6.8, 6.10 and 6.12). No stiffness matrices or mass matrices are ever assembled. The advantage of an explicit procedure is that it provides the computational efficiency needed to render feasible the solution of wave propagation problems (which require exceptionally small Δt). The computations over each time step are relatively simple. The major disadvantage is that nonlinear effects due to the spatial variables cannot be "iterated" out at constant dynamic load, i.e., no equilibrium iterations within a time step are possible.

With implicit integration operators, the displacement increment over Δt is used to predict the accelerations at $t+\Delta t$. The displacement increment is obtained by using an effective stiffness matrix to form a set of simultaneous equations. The advantage of an implicit approach lies in the capability to completely correct for nonlinear effects in the spatial variables that occur over Δt through equilibrium iterations. In addition, corrections in the estimated displacement increment over Δt during iterations are utilized to improve the predicted acceleration at $t+\Delta t$. At $t+\Delta t$, the dynamic equilibrium equations can be satisfied to within a specified tolerance with an

implicit procedure. The major disadvantage of the implicit scheme is the increased computational effort required to assemble and triangulate the effective stiffness matrix. Thus, a solution procedure based on an implicit scheme is more appropriate for problems in which wave propogation is not as important as gross inertia effects. Much larger time steps are permitted thus reducing computational costs. If necessary, the implicit scheme can be used to compute wave propagation effects.

The implicit solution scheme is adopted for detailed discussion here and will be adopted in the software. The implicit scheme builds upon the experience gained in the solution of static nonlinear problems. Static analysis is recovered as the degenerate case when the mass matrix is zero.

The nonlinear equations of motion (6.17) are solved iteratively to determine displacements and accelerations at $t+\Delta t$ beginning with a known solution for time t. Suppose that 'm' such iterations have been performed and the current estimate of the total nodal displacement field is ${}^{t+\Delta t}\{a_m\}$ and the nodal acceleration field is ${}^{t+\Delta t}\{\ddot{a}_m\}$. If ${}^{t+\Delta t}\{R_m\}$ computed using (6.17) vanishes for these displacements and accelerations, the solution has converged. Computations for the next time step are begun. If ${}^{t+\Delta t}\{R_m\}$ does not vanish, a generalized force imbalance exists at the nodes and is given by the terms in ${}^{t+\Delta t}\{R_m\}$. These are usually termed the residual loads; they must be eliminated by suitable changes of nodal displacements and accelerations during the m+1 iteration. Let $\{\Delta R_m\}$ represent the change of the residual loads necessary to remove ${}^{t+\Delta t}\{R_m\}$; then the following equation is to be solved

$$^{t+\Delta t}\{R_m\} + {}^{t+\Delta t}\{\Delta R_m\} = \{0\} \tag{6.18}$$

The increment of the residual load vector is simply determined by summing the increment of each vector in (6.17).

$$^{t+\Delta t}\{\Delta R_m\} = {}^{t+\Delta t}\{\Delta P_m\} - {}^{t+\Delta t}\{\Delta F_m\} - [M]\,{}^{t+\Delta t}\{\Delta \ddot{a}_m\} \tag{6.19}$$

in which $\{P\}$ and $\{F\}$ are implicit functions of the nodal displacements. Corrections in the nodal accelerations, $\{\Delta \ddot{a}_m\}$, are coupled indirectly to the nodal displacement corrections through the integration operator. The finite size corrections implied in (6.19) are also nonlinear functions of the spatial variables. By assuming that an approximate correction can be obtained using the differential of $\{R\}$,

$$^{t+\Delta t}\{\Delta R_m\} \approx {}^{t+\Delta t}\{dR_m\}, \tag{6.20}$$

a one term Taylor series about the configuration $\{a_m\}$ yields a set of linearized correction terms for use in (6.19)

$$^{t+\Delta t}\{\Delta P_m\} \approx {}^{t+\Delta t}\{dP_m\} = \left[\frac{dP_m}{da}\right]\{a_m\} = [K_\ell]_m\{a_m\} \tag{6.21}$$

$$^{t+\Delta t}\{\Delta F_m\} \approx {}^{t+\Delta t}\{dF_m\} = \left[\frac{dF_m}{da}\right]\{a_m\} = [K_T]\{a_m\} \tag{6.22}$$

Vector $\{a_m\}$ denotes an increment of nodal displacement. The symbolic differentiations inside the [ ] lead to the initial load stiffness, $[K_\ell]$, and the conventional tangent stiffness, $[K_T]$. These two matrices are the Jacobians of the functions $\{P\}$ and $\{F\}$, which are nonlinear in the nodal displacements. Explicit forms of $[K_T]$ for the T. L. and U. L. formulations are provided in following sections. The 'm' right subscript outside the [ ] denotes the nodal displacements at which the stiffnesses are evaluated. The influence of $\{\Delta \ddot{a}_m\}$ on $\{\Delta R_m\}$ is

simply a function of the integration operator and need not be considered in (6.21) or (6.22). As discussed in Section 5.4, the $[K_\ell]$ matrix arises due to non-conservative loading. It is generally non-symmetric and thus ignored in forming $\{\Delta R_m\}$. Corrections in the effective loading, $\{P\}$, due to geometry changes are incorporated by occasionally recomputing $\{P\}$ using the current geometry. Symmetric forms of $[K_T]$ may be derived for both T. L. and U. L. formulations as shown in subsequent sections.

Substitution of (6.22) and (6.20) into (6.18) and then into (6.17) yields the incremental-iterative equations of motion

$$[K_T]\{a_m\} = {}^{t+\Delta t}\{P_m\} - {}^{t+\Delta t}\{F_m\} - [M]\,{}^{t+\Delta t}\{\ddot{a}_m\} \tag{6.23}$$

which can be solved for $\{a_m\}$ using linear equation solving techniques. The integration operator enters (6.23) through the ${}^{t+\Delta t}\{\ddot{a}_m\}$ term. Each vector on the right hand side of (6.23) is computed for the m^{th} iteration estimate of the nodal displacements at $t+\Delta t$ using (6.8), (6.10), (6.14) at the element level and (6.16) at the structure level. The improved estimate of nodal displacements for $t+\Delta t$ is

$${}^{t+\Delta t}\{a_{m+1}\} = {}^{t+\Delta t}\{a_m\} + \{a_m\} \tag{6.24}$$

Corresponding accelerations ${}^{t+\Delta t}\{\ddot{a}_{m+1}\}$ are computed using the integration operator. The right side of (6.23) is then evaluated for the improved displacements and accelerations (the R.H.S. is simply ${}^{t+\Delta t}\{R\}$). If this vanishes, the solution has converged; otherwise a new correction $\{a_m\}$ is computed and the process repeated.

Simplifications in solving (6.23) involve the use of various forms of the Newton-Raphson procedure. It is not necessary to recompute and triangulate $[K_T]$ during each iteration. Any previously computed $[K_T]$,

including the linear elastic [K], may be employed. The convergence rate of the iterative procedure will be affected accordingly. Over-relaxation techniques used in static analysis may also be used here to enhance the convergence characteristics.

In a multilevel substructure model, (6.23) is applied only at the highest level structure which contains the synthesized linear substructures and nonlinear finite elements. The effects of substructuring are reflected in each term of (6.23). Modal synthesis techniques discussed in previous chapters provide procedures to formulate [M] and [K] at the outset of the transient response computation. The external dynamic loading ${}^{t+\Delta t}\{P_m\}$ is generally independent of the nodal displacements (conservative loading) but not independent of time. Thus, if time dependent loads are applied inside a synthesized substructure, the equivalent dynamic loads on the nodes (and modes) remaining after synthesis must be recomputed each time step. A simplification occurs when the spatial loading pattern applied inside the substructure is synthesized once, then a specified time function is applied to the equivalent loads to yield the variation with time. The internal resisting forces, ${}^{t+\Delta t}\{F\}$, are the sum of contributions from the individual nonlinear elements and from the synthesized substructures that appear in the highest level structure. Nonlinear element contributions are given by (6.8); synthesized substructure contributions are obtained from the product of their stiffness matrix with the m^{th} estimate of the total nodal displacements ${}^{t+\Delta t}\{a_m\}$. This computation requires the retrieval of each synthesized stiffness matrix from secondary storage during every equilibrium iteration. The major computational savings derive from the greatly reduced size

Initial Calculations

1. For each linear substructure (repeat at each level for nested substructures):

   a. Form $[K]$ using standard techniques,
   b. Form $[M]$ as consistent or lumped.
   c. Reduce $[K]$ and $[M]$ using any of the dynamic reduction techniques described in Chapter 3.

2. Form the synthesized stiffness, $[K]$, and mass, $[M]$, for the highest level structure.

3. Set initial displacements ${}^{0}\{a\}$, and velocities, ${}^{0}\{\dot{a}\}$.

4. Compute initial accelerations, ${}^{0}\{\ddot{a}\}$, from equilibrium equation: $[M]\ {}^{0}\{\ddot{a}\} + [K]\ {}^{0}\{a\} = {}^{0}\{P\}$.

5. Define constants for the specific integration operator, e.g., Wilson-θ, Newmark-β. Denote these constants α_i, α_j, α_k, etc.

6. Compute the contribution of the mass to the effective stiffness for the highest level structure: $[\bar{K}] = \alpha_i[M]$.

7. Triangulate effective stiffness, $[K_E] = [K] + [\bar{K}]$, of the highest level structure, using Choleski decomposition: $[K_E] = [L][L]^T$.

Table 6.1 -- Procedure for Transient Analysis with Substructuring (Damping Neglected)

For Each Time Step $t \rightarrow t+\Delta t$:

1. If specified by the user, update the tangent stiffness, $[K_T]$, of the highest level structure. Only elements that are currently nonlinear are updated. Condensed substructure stiffnesses are re-used. Triangulate the new $[K_T] = [L][L]^T$.

2. Compute the effective load increment vector. Place it in ${}^{t+\Delta t}\{R\}$.

   ${}^{t+\Delta t}\{R\} = {}^{t+\Delta t}\{P\} - {}^{t}\{P\} - [M](\alpha_i {}^{t}\{\dot{a}\} + \alpha_j {}^{t}\{\ddot{a}\})$.

   This expression is derived by subtracting (6.17) evaluated at time t from the same equations at time $t+\Delta t$ as follows:

   $[M]\,{}^{t+\Delta t}\{\ddot{a}\} + {}^{t+\Delta t}\{F\} - [M]\,{}^{t}\{\ddot{a}\} - {}^{t}\{F\} = {}^{t+\Delta t}\{P\} - {}^{t}\{P\}$.

   Now substitute $[K_T]\{\Delta a\} = {}^{t+\Delta t}\{F\} - {}^{t}\{F\}$ and ${}^{t+\Delta t}\{\ddot{a}\} =$

   $\alpha_i\{\Delta a\} + \alpha_j {}^{t}\{\dot{a}\} + \alpha_k {}^{t}\{\ddot{a}\}$ from the integration operator to yield

   $([K_T] + \alpha_i[M])\{\Delta a\} = {}^{t+\Delta t}\{P\} - {}^{t}\{P\} + [M](\alpha_j {}^{t}\{\dot{a}\} + \alpha_k {}^{t}\{\ddot{a}\})$

   The right hand side is simply ${}^{t+\Delta t}\{R\}$. It is not necessary to include any remaining residual load, ${}^{t}\{R\}$, in the effective load vector for the new step if equilibrium iterations will be performed. Otherwise ${}^{t}\{R\}$ should be added to the ${}^{t+\Delta t}\{R\}$ derived above.

3. Solve for the displacement increment vector $[L][L]^T\{a\} = {}^{t+\Delta t}\{R\}$.

4. Proceed to step 5 if no equilibrium iterations are specified to correct the acceleration or to eliminate residual loads due to nonlinear response in the spatial variables.

Table 6.1 -- (continued)

4. (continued) Otherwise, begin equilibrium iterations:

   a. Set m = 1

   b. Compute improved estimates of the <u>total</u> nodal displacements and accelerations given values at time t and the most recent estimates of change over the step.

   $$^{t+\Delta t}\{a_m\} = {}^{t}\{a\} + \{a_m\}; \quad {}^{t+\Delta t}\{\ddot{a}_m\} = \alpha_i {}^{t+\Delta t}\{a_m\} - \alpha_j {}^{t}\{\dot{a}\} - \alpha_k {}^{t}\{\ddot{a}\}.$$

   c. For the improved <u>total</u> displacements and accelerations, compute the <u>total</u> residual load vector: $^{t+\Delta t}\{R_m\} = {}^{t+\Delta t}\{P_m\} - {}^{t+\Delta t}\{F_m\} - [M]\ {}^{t+\Delta t}\{\ddot{a}_m\}$.

   d. Solve for a correction to the displacement change over the step:

   $$[L][L]^T\{\Delta a\} = {}^{t+\Delta t}\{R_m\}.$$

   e. Update the estimated change in nodal displacements for the time step: $\{a_{m+1}\} = \{a_m\} + \{\Delta a\}$.

   f. Perform checks on convergence using $^{t+\Delta t}\{R_m\}$, $\{\Delta a\}$, $^{t+\Delta t}\{a_m\}$. If converged, jump to step 5. If not converged and iteration number m is less than the maximum allowed, increment m and go to g; otherwise, terminate analysis as a nonconvergent system exists.

   g. If specified by the user, update $[K_T]$ and triangulate. Go to b.

5. Compute new accelerations, velocities, and displacements for $t+\Delta t$ to serve as initial conditions for the next step:

$$^{t+\Delta t}\{\ddot{a}\} = \alpha_i {}^{t}\{a\} + \alpha_j {}^{t}\{\dot{a}\} + \alpha_k {}^{t}\{\ddot{a}\}.$$

$$^{t+\Delta t}\{\dot{a}\} = {}^{t}\{\dot{a}\} + \alpha_l {}^{t}\{\ddot{a}\} + \alpha_m {}^{t+\Delta t}\{\ddot{a}\}.$$

$$^{t+\Delta t}\{a\} = {}^{t}\{a\} + \{a\}.$$

Stresses, strains, reactions, etc. can be computed for the highest level structure. Similar results may be recovered for condensed substructures.

Table 6.1 -- (continued)

of $[K_T]$ possible with substructuring. The frequent triangulations of $[K_T]$ required to enhance convergence each consume much less time than does the reduction of a $[K_T]$ representing an equivalent unsubstructured model.

Table 6.1 provides a detailed flow of the transient response computation for a multilevel substructured model based on an implicit integration operator.

6.4 Total Lagrangian Stiffness

Specific forms of the elemental stiffnesses for the T. L. approach are described in this section. Once element stiffnesses are generated, a substructure stiffness is assembled using standard, well documented techniques [6.12].

The approach adopted here derives from the work of Nayak [6.2], McKay [6.13], and Dodds [6.14]. As shown in (6.8), the element nodal force vector due to internal stress and strain is given by

$$\{F^e\} = \int_{*V} {}_*[B]_*\{\sigma\}^* dV_e \tag{6.25}$$

in which the above integral may be evaluated at any time t. With the configuration at time t known, an approximation for $\{F^e\}$ at $t+\Delta t$ is given by

$${}^{t+\Delta t}\{F^e\} \approx {}^t\{F^e\} + d^t\{F^e\} = {}^t\{F^e\} + {}^t_o[K_T^e]d\{a\} \tag{6.26}$$

in which $d\{a\}$ represents the change in element nodal displacements over Δt.

For the T. L. approach, the (*) configuration in (6.25) is taken to be that at time 0. The integrand is evaluated at time t but referred to

the configuration at time 0 using the results derived in Chapter 5, specifically (5.81). After substituting into (6.25) the differential, internal force vector may be written as

$$d^{\dagger}\{F^{e}\} = d\left[\int_{{}^{0}V} {}^{\dagger}_{0}[B]^{T}\, {}^{\dagger}_{0}\{S\}\, {}^{0}dV\right] \tag{6.27}$$

Using the product rule, differentiation under the integral sign yields

$$d^{\dagger}\{F^{e}\} = \int_{{}^{0}V} {}^{\dagger}_{0}[B]^{T} d^{\dagger}_{0}\{S\} + d^{\dagger}_{0}[B]^{T}\, {}^{\dagger}_{0}\{S\}\, {}^{0}dV \tag{6.28}$$

From (5.100), the increment of 2nd P-K stress is given by

$$d^{\dagger}_{0}\{S\} = {}^{\dagger}_{0}[D_{T}]\, d^{\dagger}_{0}\{\varepsilon\} \tag{6.29}$$

with the tangent modulus matrix provided by (5.101). The differential of the Green strain is given by (5.55) as

$$d^{\dagger}_{0}\{\varepsilon\} = {}^{\dagger}_{0}[B]\, d\{a\} \tag{6.30}$$

in which the form for ${}^{\dagger}_{0}[B]$ is repeated in Table 6.2.

The differential of the ${}^{\dagger}_{0}[B]$ matrix in (6.28) with respect to the element nodal displacements has been derived in a convenient form by Nayak [6.2] and Dodds [6.14]. The ${}^{\dagger}_{0}[B]$ matrix is split into two components, the first being independent of the nodal displacements which vanishes under differentiation with respect to the nodal displacements. Differentiation of the nonlinear terms yields an expression of the form

$$d^{\dagger}_{0}[B]^{\dagger}_{0}\{S\} = {}^{\dagger}_{0}[G]^{T}\, {}^{\dagger}_{0}[M]\, {}^{\dagger}_{0}[G]\; d\{a\} \tag{6.31}$$

following rather lengthy but straightforward differentiation of the summations. Complete details of the process are given by Dodds

[6.14]. Matrix ${}^{\tau}_{o}[G]$ simply contains derivatives of the element shape functions with respect to the initial, ${}^{o}\{x\}$, coordinate system. The ${}^{\tau}_{o}[M]$ matrix contains a regular arrangement of the 2nd P-K stress components, ${}^{\tau}_{o}\{S\}$. Details of the [G] and [M] matrices for the 2-D case are given in Table 6.2.

The complete element tangent stiffness is obtained by combining the effects of (6.28-6.31) as

$$ {}^{\tau}\{\Delta F^{e}\} = {}^{\tau}_{o}[K_{T}]\{\Delta a\} = [{}^{\tau}_{o}[K_{NL}] + {}^{\tau}_{o}[K_{\sigma}]]\{\Delta a\} \tag{6.32} $$

in which finite increments of the nodal displacements, $\{\Delta a\}$, are used during computation to replace the differential values.

The initial stress stiffness, $[K_{\sigma}]$, is given by (6.31). The nonlinear stiffness, $[K_{NL}]$, is given by

$$ {}^{\tau}_{o}[K_{NL}] = \int_{{}^{o}V} {}^{\tau}_{o}[B]^{T\,\tau}_{o}[D_{T}]^{\tau}_{o}[B]\,{}^{o}dV^{e} \tag{6.33} $$

and contains the usual linear stiffness, in addition to displacement dependent contributions, i.e., if the nodal displacements are zero, ${}^{\tau}_{o}[B]$ simplifies to the standard linear strain-displacement relation.

6.5 Updated Lagrangian Stiffness

The procedures adopted in the previous section also enable the derivation of stiffness matrices for the U. L. formulation. The resulting stiffness matrices for U. L. have a form very similar to that for T. L. which simplifies the associated programming when both approaches are implemented.

The Green strain increments in 2-D are:

$$ {}^{\tau}_{o}\{\delta\varepsilon\} = {}^{\tau}_{o}[B]\{\delta a\}, \qquad d\,{}^{\tau}_{o}\{\varepsilon\} = {}^{\tau}_{o}[B]\; d\{a\} $$

$$ {}^{\tau}_{o}\{\varepsilon\}^{T} = \left\lfloor {}^{\tau}_{o}\varepsilon_x \;;\; {}^{\tau}_{o}\varepsilon_y \;;\; {}^{\tau}_{o}\gamma_{xy} \right\rceil \qquad \{\delta a\}^{T} = \left\lfloor \delta u \;;\; \delta v \right\rceil ; \text{ variation of total displacements} $$

$$ {}^{\tau}_{o}\varepsilon_x = {}^{\tau}_{o}u_x + 0.5\left[{}^{\tau}_{o}u_x^2 + {}^{\tau}_{o}v_x^2 \right]; \quad x = {}^{o}x; \quad {}^{\tau}_{o}u_x = \frac{\partial\, {}^{\tau}u}{\partial\, {}^{o}x} $$

$$ {}^{\tau}_{o}\varepsilon_y = {}^{\tau}_{o}v_y + 0.5\left[{}^{\tau}_{o}u_y^2 + {}^{\tau}_{o}v_y^2 \right]; \quad y = {}^{o}y $$

$$ {}^{\tau}_{o}\gamma_{xy} = {}^{\tau}_{o}u_y + {}^{\tau}_{o}v_x + {}^{\tau}_{o}u_x\, {}^{\tau}_{o}u_y + {}^{\tau}_{o}v_x\, {}^{\tau}_{o}v_y $$

A virtual or differential change of the strain ${}^{\tau}_{o}\varepsilon_x$ is given by

$$ \delta\, {}^{\tau}_{o}\varepsilon_x = \delta\, {}^{\tau}_{o}u_x + {}^{\tau}_{o}u_x\, \delta\, {}^{\tau}_{o}u_x + {}^{\tau}_{o}v_x\, \delta\, {}^{\tau}_{o}v_x \text{ with other terms derived similarly.} $$

$$ {}^{\tau}_{o}[B] = \left[\begin{array}{c|c} (1 + {}^{\tau}_{o}u_x)\dfrac{\partial}{\partial x} & {}^{\tau}_{o}v_x \dfrac{\partial}{\partial x} \\ \hline {}^{\tau}_{o}u_y \dfrac{\partial}{\partial y} & (1 + {}^{\tau}_{o}v_y)\dfrac{\partial}{\partial y} \\ \hline {}^{\tau}_{o}u_y \dfrac{\partial}{\partial x} + (1 + {}^{\tau}_{o}u_x)\dfrac{\partial}{\partial y} & (1 + {}^{\tau}_{o}v_y)\dfrac{\partial}{\partial x} + {}^{\tau}_{o}v_x \dfrac{\partial}{\partial y} \end{array}\right] $$

Table 6.2 -- Summary of Total Lagrangian Matrices

Substituting shape functions into the above, the corresponding matrices for the Jth node are:

$$
{}^{t}_{o}[B^{J}] = \begin{bmatrix}
(1 + {}^{t}_{o}u_{x})\dfrac{\partial N_J}{\partial x} & {}^{t}_{o}v_{x}\dfrac{\partial N_J}{\partial x} \\
{}^{t}_{o}u_{y}\dfrac{\partial N_J}{\partial y} & (1 + {}^{t}_{o}v_{y})\dfrac{\partial N_J}{\partial y} \\
{}^{t}_{o}u_{y}\dfrac{\partial N_J}{\partial x} + (1 + {}^{t}_{o}u_{x})\dfrac{\partial N_J}{\partial y} & (1 + {}^{t}_{o}v_{y})\dfrac{\partial N_J}{\partial x} + {}^{t}_{o}v_{x}\dfrac{\partial N_J}{\partial y}
\end{bmatrix}
$$

$$
{}^{t}_{o}[G^{J}] = [\ {}^{t}_{o}G^{1}\ ;\ {}^{t}_{o}G^{2}\ ;\ \ldots\ ;\ {}^{t}_{o}G^{n}\ ]
$$

$$
{}^{t}_{o}[G^{J}]^{T} = \begin{bmatrix}
\dfrac{\partial N_J}{\partial x} & \dfrac{\partial N_J}{\partial y} & 0 & 0 \\
0 & 0 & \dfrac{\partial N_J}{\partial x} & \dfrac{\partial N_J}{\partial y}
\end{bmatrix}
$$

$$
{}^{t}_{o}[M] = \begin{bmatrix}
S_{xx} & S_{xy} & 0 & 0 \\
S_{xy} & S_{yy} & 0 & 0 \\
0 & 0 & S_{xx} & S_{xy} \\
0 & 0 & S_{xy} & S_{yy}
\end{bmatrix} ;\ S_{ij} = {}^{t}_{o}S_{ij}
$$

Table 6.2 -- (continued)

The basic equation of the U. L. formulation defines the internal virtual work of an element at time $t+\Delta t$ relative to the configuration at time t as

$$ {}^{t+\Delta t}\delta W^{e}_{int} = -\int_{{}^{t}V} {}^{t+\Delta t}_{t}\{\delta\varepsilon\}\, {}^{t+\Delta t}_{t}\{S\}\, {}^{t}dV \tag{6.34} $$

Defining a suitable differential operator matrix, ${}^{t+\Delta t}_{t}[B]$, to yield virtual strains from virtual nodal displacements, (6.34) may be written in the form of (6.8) as

$$ {}^{t+\Delta t}\delta W^{e}_{int} = -\{\delta a^{e}\}^{T}\int_{{}^{t}V} {}^{t+\Delta t}_{t}[B]^{T}\, {}^{t+\Delta t}_{t}\{S\}\, {}^{t}dV = -\{\delta a^{e}\}^{T}\, {}^{t+\Delta t}\{F^{e}\} \tag{6.35} $$

in which ${}^{t+\Delta t}\{F^{e}\}$ in (6.35) has the identical meaning as the same quantity in (6.26).

The element resisting force vector at $t+\Delta t$ is expressed in terms of a known value for the configuration at time t plus a differential change over Δt as

$$ {}^{t+\Delta t}\{F^{e}\} \approx {}^{t}\{F^{e}\} + d^{t}\{F^{e}\} \tag{6.36} $$

in which

$$ d^{t}_{t}\{F^{e}\} = {}^{t}_{t}[K_{T}]\, d\{a\} \tag{6.37} $$

The difference between (6.36-37) and (6.26) is simply the reference configuration of the tangent stiffness. The incremental force, $d^{t}\{F^{e}\}$, and nodal displacement increment, $d\{a\}$, have the identical physical interpretations in both T. L. and U. L. formulations.

The specific form of ${}^{t}_{t}[K_{T}]$ is obtained by evaluating the differential of (6.35) with respect to the displacements at the value of nodal displacements corresponding to time t. Symbolically,

$$d^{\dagger}\{F^{e}\} = \int_{\dagger_V} {}^{\dagger}_{\dagger}[B]^{T} d^{\dagger}_{\dagger}\{S\} + d^{\dagger}_{\dagger}[B]^{T\dagger}_{\dagger}\{S\}^{\dagger} dV \tag{6.38}$$

From Chapter 5, the following relationships are used to expand (6.38)

$${}^{\dagger}_{\dagger}\{S\} = {}^{\dagger}\{\sigma\} \text{ (the Cauchy stress)} \tag{6.39}$$

$$d^{\dagger}_{\dagger}\{S\} = {}_{\dagger}\{S\} = {}^{\dagger}_{\dagger}[D_T] d^{\dagger}_{\dagger}\{\varepsilon\} \tag{6.40}$$

The tangent modulus matrix, referred to the current (time $\dagger$) configuration, ${}^{\dagger}_{\dagger}[D_T]$, was derived in Chapter 5. The differential strain, measured from the current configuration, is

$$d^{\dagger}_{\dagger}\{\varepsilon\} = {}^{\dagger}_{\dagger}[B] d\{a\} \tag{6.41}$$

in which ${}^{\dagger}_{\dagger}[B]$ is simply ${}^{\dagger+\Delta\dagger}_{\dagger}[B]$ for a zero increment of displacement (see Table 6.3). Notice that the resulting form is the conventional linear [B] matrix containing derivatives taken with respect to total displacements at time $\dagger$. Using (6.40) and (6.41), the first integrand term of (6.38) becomes the "nonlinear" stiffness matrix ${}^{\dagger}_{\dagger}[K_{NL}]$ and is given by

$${}^{\dagger}_{\dagger}[K_{NL}] = \int_{\dagger_V} {}^{\dagger}_{\dagger}[B_L]^{T} {}^{\dagger}_{\dagger}[D_T] {}^{\dagger}_{\dagger}[B_L]^{\dagger} dV \tag{6.42}$$

in which the "L" subscript is used to denote the simple, linear form taken by [B].

To obtain the second integrand term of (6.38), an expression for $d^{\dagger}_{\dagger}[B]$ is required. This is found by differentiating ${}^{\dagger+\Delta\dagger}_{\dagger}[B]$, with respect to the nodal displacements, then evaluating the terms for a zero displacement increment. A comparison of ${}^{\dagger+\Delta\dagger}_{\dagger}[B]$ in Table 6.3 with ${}^{\dagger}_{o}[B]$ in Table 6.2 shows a very similar form. The same operations that yield $d^{\dagger}_{o}[B]$, readily yield $d^{\dagger}_{\dagger}[B]$ as

The Green strain increments in 2-D are:

$$ {}^{t+\Delta t}_{t}\{\delta\varepsilon\} = {}^{t+\Delta t}_{t}[B]\{\delta a\}, \qquad d\,{}^{t+\Delta t}_{t}\{\varepsilon\} = {}^{t+\Delta t}_{t}[B]\; d\{a\} $$

$$ {}^{t+\Delta t}_{t}\{\varepsilon\}^T = {}_{t}\{\varepsilon\}^T = \left\lfloor {}_{t}\varepsilon_x \; ; \; {}_{t}\varepsilon_y \; ; \; {}_{t}\gamma_{xy} \right\rceil $$

$$ \{\delta a\}^T = \left\lfloor \delta_{t}u \; ; \; \delta_{t}v \right\rceil \; ; \text{ variation of incremental displacements that produce configuration at } t+\Delta t. $$

$$ {}_{t}\varepsilon_x = {}_{t}u_{\bar{x}} + 0.5\left[{}_{t}u_{\bar{x}}^2 + {}_{t}v_{\bar{x}}^2 \right]; \qquad \bar{x} = {}^{o}x + u \; ; \quad {}_{t}u_{\bar{x}} = \frac{\partial_{t}u}{\partial\bar{x}} $$

$$ {}_{t}\varepsilon_y = {}_{t}v_{\bar{y}} + 0.5\left[{}_{t}u_{\bar{y}}^2 + {}_{t}v_{\bar{y}}^2 \right]; \qquad \bar{y} = {}^{o}y + v $$

$$ {}_{t}\gamma_{xy} = {}_{t}u_{\bar{y}} + {}_{t}v_{\bar{x}} + {}_{t}u_{\bar{x}}\,{}_{t}u_{\bar{y}} + {}_{t}v_{\bar{x}}\,{}_{t}v_{\bar{y}} $$

A virtual or differential change of the incremental strain ${}_{t}\varepsilon_x$ is given by

$$ \delta_{t}\varepsilon_x = \delta_{t}u_{\bar{x}} + {}_{t}u_{\bar{x}}\,\delta_{t}u_{\bar{x}} + {}_{t}v_{\bar{x}}\,\delta_{t}v_{\bar{x}} \; ; \text{ with other terms derived similarly.} $$

$$ {}^{t+\Delta t}_{t}[B] = \left[\begin{array}{c|c} (1 + {}_{t}u_{\bar{x}})\dfrac{\partial}{\partial\bar{x}} & {}_{t}v_{\bar{x}}\dfrac{\partial}{\partial\bar{x}} \\ \hline {}_{t}u_{\bar{y}}\dfrac{\partial}{\partial\bar{y}} & (1 + {}_{t}v_{\bar{y}})\dfrac{\partial}{\partial\bar{y}} \\ \hline {}_{t}u_{\bar{y}}\dfrac{\partial}{\partial\bar{x}} + (1 + {}_{t}u_{\bar{x}})\dfrac{\partial}{\partial\bar{y}} & (1 + {}_{t}v_{\bar{y}})\dfrac{\partial}{\partial\bar{x}} + {}_{t}v_{\bar{x}}\dfrac{\partial}{\partial\bar{y}} \end{array}\right] $$

Setting ${}_{t}u = {}_{t}v = 0$;

$$ {}^{t}_{t}[B] = \left[\begin{array}{c|c} \dfrac{\partial}{\partial\bar{x}} & 0 \\ \hline 0 & \dfrac{\partial}{\partial\bar{y}} \\ \hline \dfrac{\partial}{\partial\bar{y}} & \dfrac{\partial}{\partial\bar{x}} \end{array}\right] $$

Table 6.3 -- Summary of Updated Lagrangian Matrices

Substituting shape functions into the forgoing, the corresponding matrices for the Jth node are:

$$ {}^{t+\Delta t}_{t}[B^J] = \begin{bmatrix} (1 + {}^{t}u_x)\frac{\partial N_J}{\partial x} & {}^{t}v_x\frac{\partial N_J}{\partial x} \\ {}^{t}u_y\frac{\partial N_J}{\partial y} & (1 + {}^{t}v_y)\frac{\partial N_J}{\partial y} \\ {}^{t}u_y\frac{\partial N_J}{\partial x} + (1 + {}^{t}u_x)\frac{\partial N_J}{\partial y} & (1 + {}^{t}v_y)\frac{\partial N_J}{\partial x} + {}^{t}v_x\frac{\partial N_J}{\partial y} \end{bmatrix}; \quad {}^{t}_{t}[B] = \begin{bmatrix} \frac{\partial N_J}{\partial x} & 0 \\ 0 & \frac{\partial N_J}{\partial y} \\ \frac{\partial N_J}{\partial y} & \frac{\partial N_J}{\partial x} \end{bmatrix} $$

$$ {}^{t}_{t}[G] = [\ {}^{t}_{t}G^1\ ;\ {}^{t}_{t}G^2\ ;\ \ldots\ ;\ {}^{t}_{t}G^n\ ] $$

$$ {}^{t}_{t}[G^J]^T = \begin{bmatrix} \frac{\partial N_J}{\partial x} & \frac{\partial N_J}{\partial y} & 0 & 0 \\ 0 & 0 & \frac{\partial N_J}{\partial x} & \frac{\partial N_J}{\partial y} \end{bmatrix} $$

$$ {}^{t}_{t}[M] = \begin{bmatrix} \sigma_{xx} & \tau_{xy} & 0 & 0 \\ \tau_{xy} & \sigma_{yy} & 0 & 0 \\ 0 & 0 & \sigma_{xx} & \tau_{xy} \\ 0 & 0 & \tau_{xy} & \sigma_{yy} \end{bmatrix} \qquad \sigma_{ij} = {}^{t}_{t}\sigma_{ij} $$

Table 6.3 -- (continued)

$$d\,{}^{t}_{t}[B]\;{}^{t}_{t}\{S\} = {}^{t}_{t}[G]^{T}\;{}^{t}_{t}[M]\;{}^{t}_{t}[G]\;d\{a\} \tag{6.43}$$

in which the details of ${}^{t}_{t}[G]$ and ${}^{t}_{t}[M]$ are shown in Table 6.3. Matrix ${}^{t}_{t}[M]$ contains the Cauchy stresses at time t arranged in the identical form that the 2nd P-K stresses appear in ${}^{t}_{o}[M]$. Similarly, the form of ${}^{t}_{t}[G]$ is identical to that of ${}^{t}_{o}[G]$ except that derivatives are taken with respect to the displacements at time t rather than time 0. The integral of the product shown in (6.43) is again termed an initial stress stiffness.

The complete element tangent stiffness is obtained by combining (6.42) and (6.43) to yield

$${}^{t}\{\Delta F^{e}\} = {}^{t}_{t}[K_{T}]\{\Delta a\} = [{}^{t}_{t}[K_{NL}] + {}^{t}_{t}[K_{\sigma}]]\{\Delta a\} \tag{6.44}$$

6.6 Comparison of Formulations

Details of a transient solution procedure based upon an implicit integration scheme have been described in this chapter. An implicit scheme is recommended over an explicit scheme for three reasons. First, the procedures for static linear and nonlinear analysis are recovered from the implicit scheme by simply omitting the mass matrix; the degenerate case of an explicit scheme does not yield a formulation suitable for static analysis. The capability to perform both static and transient analyses with the same software is particularly attractive to engineers since a static analysis invariably precedes a dynamic analysis. The second reason to select an implicit scheme is the more general class of structures that may be analyzed. An implicit scheme may be used to compute the details of localized wave propagation under very high velocity impact as well as the vibration characteristics of a

massive structure subjected to time dependent loads. The explicit scheme may be computationally more efficient for wave propagation studies but not necessarily more efficient than an implicit approach for the latter class of problems. Thirdly, the equilibrium imbalance due to nonlinear response in the spatial variables can be "iterated out" to within a specified tolerance using the implicit scheme. Dynamic equilibrium can thus be assured at the end of each time increment.

Complete details of the element tangent stiffness matrices have been presented in the 2-D case for both the T. L. and U. L. formulations. For both approaches, stiffness matrices have the identical symbolic form, which is conveniently expressed as the sum of an initial stress stiffness, $[K_\sigma]$, and a nonlinear stiffness, $[K_{NL}]$. Given the common rate of work per unit mass expression from which each formulation is derived, it is expected that identical results for each solution would be obtained (provided the full stress rate transformations described in Chapter 5 are utilized). Some computational evidence [6.10, 6.11] does demonstrate the agreement of overall load-deflection curves for static and dynamic solutions which include plasticity effects. Unfortunately, no comparisons are provided of the internal stress distributions, which must be identical if equivalence of the formulations is to be demonstrated. Global agreement of load-deflection does not imply identical internal stress distributions.

The computational efficiency of each approach is addressed at two levels; namely, the effort required to compute the element matrices, and the number of iterations required to attain the equilibrium configuration at time $t+\Delta t$, assuming Δt is the same in both formulations.

In each formulation the construction of $[K_\sigma]$ requires the same number of operations if the expense of coordinate updating in U. L. is neglected. In contrast, construction of the nonlinear stiffness, $[K_{NL}]$, in U. L. requires the same number of operations as required for a conventional linear stiffness. The T. L. nonlinear stiffness requires more operations due to the absence of zeroes in the [B] matrix (refer to Table 6.2). The increase in the operation count depends on the element type but easily exceeds by a factor of three the number of operations required to form $[K_{NL}]$ in U. L. Evaluation of the internal resisting force vector during iterations requires the identical number of operations in each case; the U. L. matrix ${}^{t+\Delta t}_{t}[B]$ has the identical form of ${}^{t}_{0}[B]$ in T. L. (no zero terms). The reduced operation count of U. L. compared to T. L. during element stiffness generation is an important consideration only when all matrices associated with the analysis reside in memory during execution. Once the swapping of element stiffnesses to and from disk begins, it would appear that the I/O overhead overwhelms any advantage of one formulation over the other. Furthermore, the timing results shown in Chapter 2 for several large linear and nonlinear analyses reveal that element stiffness generation times do not represent a major portion of the total solution time. Thus, a reduction of stiffness generation time obtained by the selection of U.L over T. L. does not yield an equal percentage reduction in total analysis time.

Perhaps the most important efficiency comparison between the two formulations is the number of equilibrium iterations required for convergence during each time step. Using a common program that has both U. L. and T. L. capabilities, a structure could be analyzed with each formulation for identical time steps and with an identical convergence

criterion based on total force equilibrium. The results of such a comparison would demonstrate any inherent computational advantage of one formulation over the other, a least for a particular problem. Unfortunately, no such comparisons have been found in the open literature. A systematic study of several problem classes using this approach is necessary to form the basis for any general efficiency statements.

The distorting of elements has led to numerical problems in some U. L. finite element analyses. For example, an element that is square at $t=0$ may become a badly distorted quadrilateral at time $t+\Delta t$. Element response characteristics are known to degenerate rapidly as aspect ratios increase; this will influence the response in an U. L. approach as shape function derivatives are dependent on the deformed shape at time t. In T. L., all such effects are incorporated in the pre-multiplier terms of shape function derivatives in the ${}^{t}_{0}[B]$ matrix. The shape function derivatives in T. L. are always computed relative to the configuration at time $t=0$ and thus remain well-behaved if the aspect ratio at $t=0$ is acceptable. The interesting case that demonstrates the problem with U. L. is an 8-node, 2-D isoparametric element which is square at $t=0$ but which at time t has the mid-side node displaced toward a corresponding corner node. Any movement of the mid-side node toward the corner produces a singular point in the corresponding shape function when the derivative is evaluated with respect to the current element shape. No such singularities are introduced in shape function derivatives in T. L. if the mid-side node is properly positioned at $t=0$.

A final comparison of the two formulations considers their applicability in a substructured modeling and solution procedure. At all times

t, the stiffness matrix of a linear substructure and its synthesized form remain unchanged from that at time t=0. The configuration of the structure at t=0 provides the basis to which all strains and stresses are referred. The linear, strain-displacement relations and constitutive matrix that is independent of nodal displacements at time t are utilized. Consider the common nodes along the boundary of a nonlinear region that is shared by an adjacent linear substructure. In an U. L. formulation, the coordinates of these common nodes are updated at time t to reflect the incremental displacements over the previous increment Δt. The tangent stiffness of the nonlinear region is regenerated to reflect the new geometry. Thus, there exists a discontinuity of nodal positions along the boundary. If the substructure stiffness is recomputed and synthesized for the updated geometry along the boundary (and the consequent internal repositioning of nodes) all the advantages of a substructured model are lost. The alternative is to assume that nodes along the interface are sufficiently remote from any effects causing nonlinear behavior that the response in nonlinear elements on the boundary is actually linear, i.e., displacements and strains are infinitesimal and the matrices involved at the element level revert to the linear analysis form. If such an assumption is actually verified in the analysis, the integrity of the results is assured.

In a substructured T. L. solution, discontinuities of nodal positions along a boundary do not occur since nodal coordinates are never updated for the displacement increments. However, the effects of deformation and rotation in the nonlinear elements adjacent to the boundary are implicitly incorporated in the tangent stiffness through the displacement dependent terms in ${}^{t}_{o}[B]$. Thus, the same difficulties

described above for U. L. appear to arise for T. L. However, the maintenance of external geometric continuity is more appealing than allowing the discontinuities to develop. It could possibly be argued that the T. L. formulation would introduce less disturbance in the strain field across the boundary than U. L., but there exists no computational evidence to collaborate this speculation.

In view of the considerations described above, a Total Lagrangian approach is recommended to form the basis of a general software system having broad applicability. If computational evidence in support of U. L. as a more efficient approach should be forthcoming, only minor changes in software to support a T. L. formulation are required. In fact, nearly all U. L. dependent computations can be isolated within element dependent routines. The reverse situation of implementing a T. L. approach in a system designed only for U. L. is not nearly as simple.

6.7 References

6.1 Hibbitt, H., Marcal, P., and Rice, J., "A Finite Element Formulation for Problems of Large Strain and Large Displacement," Int. J. Solids Structures, Vol. 6, 1970, pp. 1069-1086.

6.2 Nayak, G., "Plasticity and Large Deformation Problems by the Finite Element Method," unpublished Ph.D. Thesis, University of Wales, Swansea, 1971. (C/PH/15/1971).

6.3 Nemat-Nasser, S. and Shatoff, H., "A Consistent Numerical Method for the Solution of Nonlinear Elasticity Problems at Finite Strains," SIAM J. of Appl. Math, Vol. 20, 1971, pp. 462-481.

6.4 Hartzman, M. and Hutchinson, J., "Nonlinear Dynamics of Solids by the Finite Element Method," Computers of Structures, Vol. 2, 1972, pp. 47-77.

6.5 Belytschko, T. and Hsieh, B., "Non-Linear Transient Finite Element Analysis with Convected Coordinates," Int. J. Num. Meth. Engr., Vol. 7, 1973, pp. 255-271.

6.6 Key, S.W., "A Finite Element Procedure for the Large Deformation Dynamic Response of Axisymmetric Solids," Comp. Meth. App. Mech. Engr., Vol. 4, 1974, pp. 195-218.

6.7 McMeeking, R. and Rice, J., "Finite Element Formulations for Problems of Large Elastic-Plastic Deformation," Int. J. Solids Structures, Vol. 11, 1975, pp. 601-616.

6.8 Yaghmai, S. and Popov, E., "Incremental Analysis of Large Deflections of Shells of Revolution," Int. J. Solids Structures, Vol. 7, 1971, pp. 1375-1393.

6.9 Nagarajan, S. and Popov, E., "Plastic and Viscoplastic Analysis of Axisymmetric Shells," Int. J. Solids Structures, Vol. 11, 1975, pp. 1-19.

6.10 Bathe, K, Ramm, E., and Wilson E., "Finite Element Formulations for Large Deformation Dynamic Analysis," Int. J. Num. Meth. Engr., Vol. 9, 1975, pp. 353-386.

6.11 Bathe, K. and Ozdemir, H., "Elastic-Plastic Large Deformation Static and Dynamic Analysis," Computers & Structures, Vol. 6, 1976, pp. 81-92.

6.12 Cook, R.D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, N.Y., N.Y., 1981, 2nd Ed.

6.13 McKay, D., "Finite Element Formulations for Large Strain Plasticity Problems," unpublished M.S. Thesis, University of Wales, Swansea, 1980, (C/M/146/80).

6.14 Dodds, R.H., "Numerical and Software Requirements for General Nonlinear Finite Element Analysis," Civil Engineering Studies, Structural Research Series No. 454, University of Illinois at Urbana-Champaign, 1978.

CHAPTER 7

USER INTERFACE -- INPUT DESIGN

7.1 General

The most popular approach to user communication with structural analysis software is the problem oriented language (POL). Virtually all successful software systems use the POL approach, either by initial design or by the use of pre-processors to translate POL input into fixed format, card images. The POL approach provides the user with greater flexibility by placing him in control of the process rather than forcing him to conform to rigid formats and sequences. The self-documenting nature of the input reduces the need for reference to manuals and provides a concise description of the structural model for other analysts. The POL is essential for interactive processing where error recovery is often necessary.

Dynamic analysis with multilevel substructuring will be implemented as an extension of the present POLO-FINITE structural mechanics system. The philosophy established during the development of POLO-FINITE was to maintain as much independence as possible among the various components of a complete structural model. These components include nonlinear material model specification, geometric definition of structures, parameters controlling the nonlinear algorithms, and requests for computation and output. The primary reasons for choosing this approach are to provide maximum flexibility in using condensed substructures as elements in higher level structures and to minimize the effect of changes

in the structural model throughout the analysis-design sequence.

Wherever possible, this philosophy is maintained in the extension to dynamics. However, one area exists in which dynamic solution parameters must be tied directly to the geometric definition of a substructure. This is the frequency analysis of a substructure that is to be condensed by modal synthesis. Since economical frequency analysis depends upon the type of structure, the number of eigenpairs required, and the solution method, it is not appropriate to select just one solution algorithm for all substructures in a complex model. Various substructures will have differing characteristics and may require an unequal number of retained modes for condensation. It is also possible that one substructure could be condensed two or more times in differing ways, with varying geometric and generalized DOF, for use in separate, higher level structures. Thus, it is necessary to tie the selection of the eigenproblem solution method to the structure definition.

The capabilities selected for general purpose dynamic analysis, along with the various options and parameters that control the solution, must be defined accurately and unambiguously by the POL. Section 7.2 presents an explanation of the capabilities to be incorporated into POLO-FINITE. Section 7.3 lists the command structure for dynamics. Examples illustrating the use of these commands are given in section 7.4.

7.2 Description of the POL

Structure and Element Mass

The mass of a structure can be divided into two parts: primary and secondary. Primary mass is the mass of the load-carrying components (elements) of the structure. Its definition is easily added to the specification of an element through two new element properties. The first defines the type of mass formulation: LUMPED or CONSISTENT. The second is the DENSITY of the material of which the element is composed. The element mass matrix can then be formed using existing shape functions. Assembly of primary mass for a structure will follow a procedure identical to that used in stiffness assembly. The current FINITE system accepts up to thirty DOF at each node of an element. These are the displacement DOF: U, V, and W, plus their first and second derivatives: UX, VX, WX, UY, etc. Depending upon the particular element formulation, it is possible for mass to be assigned to any or all of these DOF.

Secondary mass is the mass of non-load-carrying components, such as concentrated and distributed live-loads, that are supported by the structure. Secondary mass is defined in a manner similar to the application of gravity loads. The secondary mass is resolved into equivalent nodal masses via the appropriate shape functions. The result will always be a lumped mass matrix which is added to the primary mass of the structure.

There are three types of secondary mass: nodal, element, and pattern. Nodal mass is mass that is concentrated at a structure node. Element mass is concentrated or distributed on the surface of and element. As with primary mass, secondary mass may be assigned to any of the thirty nodal DOF. The pattern mass is provided as a convenience to the user. It enables the definition of secondary mass in terms of a previously defined loading condition, usually gravity loading. The user need specify only the name of the loading condition to be used as the pattern and a value for the acceleration of gravity to support the appropriate conversion from force to mass.

Substructure loads can be defined at the lower levels and then applied selectively at the higher levels of the structure hierarchy through the "EXTERNAL ELEMENT LOADS" facility. It would be advantageous to have a similar capability with respect to substructure mass. The analyst may wish to use several copies of a particular substructure, each with a different mass distribution (described by secondary mass input). This analogy to substructure loading implies the need for an "EXTERNAL ELEMENT MASS" facility. This is not possible in dynamic analysis since the change in the mass of a substructure changes the natural frequencies and mode shapes. Physically distinct substructures exist when the mass distribution varies. Each of these distinct substructures must be uniquely defined at the lower levels of the hierarchy. The mass is then automatically carried through the hierarchy via the condensation or synthesis process.

The commands for computation (assembly) and output of the mass matrix for a structure or stand-alone element follow directly from those for the stiffness matrix.

Structure Damping

Since the dynamic reduction process recommended here does not include the substructure damping matrix, damping is defined only for the highest level structure. Two methods are available for defining structural damping: modal and Rayleigh. Definition of modal damping requires input of the modal damping ratio for each vibration mode under consideration. Modal damping is applicable only to transient analysis by mode superposition. Rayleigh damping involves the definition of two damping ratios at two selected frequencies; the frequencies need not be eigenvalues of the structure. Rayleigh damping is applicable to transient analysis by either mode superposition or time-history integration. Use of Rayleigh damping requires that a frequency analysis be performed in order to compute the modal damping ratios for mode superposition or to explicitly form the damping matrix for time-history integration.

Depending upon the method used to define damping, either the damping matrix or modal ratios can be output for the structure.

Frequency Analysis

As previously mentioned, the parameters controlling the frequency analysis (computation of natural frequencies and mode shapes) must be defined explicitly for each structure for which the analysis is to be performed. No default analysis method is adopted. The syntax for

specification of the solution method is similar to that for a nonlinear material. That is, the TYPE of solution procedure is identified followed by a listing of PROPERTIES unique to that type. When appropriate the range of frequencies and the maximum number of modes to consider are specified at this time rather than in the computation request. Solution method properties can be changed via analysis restart and the ACCESS STRUCTURE... sequence. If a substructure is to be condensed by Guyan reduction, no frequency analysis specification is required.

In the request for computation the analyst may select a nonlinear dynamic loading and time step at which the frequency analysis is to be performed. This allows the user to interrupt a transient analysis after some nonlinear behaviour has occured and compute natural frequencies and mode shapes of the structure. Standard output includes natural frequencies and modes shapes.

Prior to a transient analysis by mode superposition, the user may examine the modal content of a particular dynamic loading condition. A special output request facilitates selection of the modes that participate in the dynamic response. After a frequency analysis the user can request output of MODAL LOADS for the loading condition. The frequency content of the loading can then be examined and the appropriate modes selected for superposition.

Dynamic Reduction

The procedure to request dynamic reduction parallels that for static condensation. The reduction method is defined at the intermediate substructure level; i.e., the substructure with only one element of type: CONDENSED. Guyan reduction is the default method. Automatic selection of master nodes, in addition to the interface nodes, is included by specifying the number of additional nodes to be retained. The fixed-interface method is invoked by specifying which substructure normal modes to retain. The modes specified must be within the range computed in the frequency analysis of the lower-level substructure. The retained modes need not be consecutively numbered. As an alternative to using normal modes, user-supplied modes can be included in the synthesis process. These modes could be derived from an experimental analysis or some other source, such as low-order polynomials. Input data describing these modes must be included with the definition of the structure to be condensed.

Dynamic reduction can be explicitly invoked with a COMPUTE STIFFNESS... or COMPUTE MASS... command for the intermediate level substructure. Reduction is performed automatically when required to satisfy a request at a higher level.

Initial Conditions

Initial conditions can be defined for a structure prior to transient analysis. They define a starting solution, in terms of displacements and velocities, for the unconstrained physical DOF at time $t = 0$. For all other times the displacements and velocities from the previous time step are used automatically.

The user can specify initial conditions in one of two ways. First, he can define numerical values for each DOF with non-zero displacement or velocity. The default initial conditions are zero displacement and velocity for all unconstrained DOF. The second method uses the static equilibrium configuration from a previous linear or nonlinear analysis. This method allows the structure to be released from some deflected initial shape with zero initial velocity. A dynamic loading can then be optionally applied.

Dynamic Loading

The dynamic loading function, $P(x,y,z,t)$, is defined such that it has a spatially-varying component, $F(x,y,z)$, and a time-varying component, $G(t)$:

$$P(x,y,z) = F(x,y,z) * G(t). \qquad (7.1)$$

Simply stated, the pattern of the load is fixed and its magnitude changes with time.

The load pattern, F(x,y,z), can be described as either actual forces applied to the structure or as support accelerations. The former can be best defined as a static linear loading condition, while the latter requires an additional loading type: NODAL ACCELERATIONS. No special provisions are necessary for input of out-of-phase support accelerations. They can be recognized and handled automatically.

The time-varying component, G(t), is combined with other loading data to form a dynamic loading condition. The G(t) vs. t relation may be harmonic, impulsive, or general. The dynamic loading condition must also include the loading pattern, F(x,y,z), which is to be used. More than one static linear loading condition can be combined to form the complete pattern of the dynamic load. Other necessary input includes the values of time t at which displacements are to be computed (thus defining the step size) and values of time t at which computed results are to be retained in the data base. This last item is important because a transient analysis of any significant duration could result in more data than could be effectively stored. Also, it is likely that stresses and strains would be required at only a few of the many time points for which displacements are computed.

Transient Analysis

Transient analysis yields the response of the structure, in terms of displacements and possibly velocities, when it is subjected to time-varying loading or support accelerations. Two approaches are available for performing transient analysis: mode superposition and time-history integration. Mode superposition requires that a frequency analysis be

performed so the equations of motion can be uncoupled. This implies that an appropriate frequency analysis method must be selected prior to requesting the transient analysis. The resulting set of independent equations is easily solved using one of the Lagrange interpolation formulas. Time-history integration is performed by any one of a number of explicit or implicit operators. Specification of the transient analysis method is similar to that for frequency analysis; the TYPE of method is defined followed by the PROPERTIES list.

The request for computation includes the loading condition, time steps, and optionally initial conditions and a mode list. The mode list is used with mode superposition to specify which modes participate in the response. Results available for output include displacements, velocities, strains, and stresses.

Shock Spectrum Analysis

The analysis of shock spectrum response is currently restricted to linear structures. The shock spectrum is input by defining the functional relationship between a spatial coordinate and a time coordinate. The spatial coordinate can be chosen as displacement, velocity, or acceleration, while the time coordinate can be either period or frequency. The user inputs discrete points from the spectrum and the remainder of the curve is constructed by linear interpolation in four-way logarithmic coordinates. The direction of application of the shock is defined using direction cosines for the translational DOF (U, V, and W for 3-D structures). The nodes at which the shock is applied are also defined.

Prior to computing the spectral response, a frequency analysis of the structure must be performed. Spectral response quantities are computed only after the corresponding output request has been made. Results available for output include spectral displacements, spectral velocities, spectral strains, and spectral stresses. These quantities can be output on a mode by mode basis or in some combined form. Methods used to combine the modal quantities include SRSS (square root of the sum of the squares) and PEAK_SRSS (peak response mode plus SRSS of the remaining modes). PEAK_SRSS is also known as the Naval sum. As a measure of the portion of the total mass responding to the shock in each mode, the modal PARTICIPATION_FACTORS can also be output.

Utility Commands

The dynamic solution process can be monitored by invoking the TRACE... command. Messages listing the currently executing module and elapsed CPU time are output at various checkpoints.

To eliminate unnecessary data from the data bases, the DESTROY... command is expanded to include results from frequency, transient, and spectral analyses.

7.3 POL Structure

7.3.1 Symbolic Conventions of the Syntax

The following is a description of the conventions used in this section to illustrate the FINITE command syntax.

A descriptor is used to identify the position and class of a data item in a particular FINITE command line. The descriptor is delimited by the characters "< >." The command

NUMBER OF NODES <integer>

implies that the word NODES is to be followed by an integer. An appropriate example is:

NUMBER OF NODES 100

The following are definitions of the descriptors used within the POL.

<integer> -- a series of digits optionally preceded by a plus or minus sign. Examples are 121, +300, -8.

<real> -- a representation of a floating point number in either decimal or exponential form. Real numbers must contain a decimal points and may be optionally signed. Examples are 1.0, -3.5, 5.24E-08.

<number> -- either an integer or a real number can be input. The data item is converted to a real number.

<integer list> -- a sequence of integers. The sequence may be listed explicitly or defined over a range of integers with a constant increment. The default increment is 1. Examples of integer lists are: 1, 2, 4, 5, 8, 11; 1-10; 2 TO 20 BY 2.

<real list> -- a sequence of real numbers. Real lists have the same form as integer lists except that there is no default increment. Examples are: 1.0, 1.5, 2.0, 3.0; 0.0-2.5 BY 0.25.

<number list> -- either an integer list or a real list is input. The data is converted to real.

<label> -- a series of letters and digits beginning with a letter. Examples are: PLANEFRAME, DEADLOAD10.

<string> -- any text enclosed within single or double quotes. An example is: "THIS IS A STRING".

In some instances a description of the physical meaning of the data item is added to the class in the syntax of a descriptor. This is helpful in clarifying the use of the data item. For example a command of the form

STRUCTURE <structure name:label>

implies that the data item following the word STRUCTURE is a label defining the name of the structure.

It is not always necessary to completely spell out every word on a command line in order to have the command correctly translated. Many words can be abbreviated and these are identified in the command syntax by underlining. The underlined portions of words identify the minimum input necessary for proper command translation. Descriptors are not underlined but are replaced by an item of the specified class when applicable. If the command syntax has the form:

NUMBER OF NODES <integer>

the following is acceptable as input:

NUM OF NODE 10

When only one word from a group of words may be selected as input, the choices are listed one above the other and enclosed in braces, "{ }". The command syntax

$$\text{COMPUTE} \begin{Bmatrix} \text{STIFFNESS} \\ \text{DISPLACEMENTS} \end{Bmatrix}$$

implies that any of the following commands are acceptable:

COMPUTE STIFFNES

COMPUTE DISPLACEMENTS

COMPUTE DISPL

When an entire word or phrase in the command is optional, it is enclosed within parentheses. The command with the syntax

NUMBER (OF) NODES <integer>

can be written as

NUM NODES 100

When more than one word from a group of words may be selected, the group is enclosed in brackets, "[ ]". The command

OUTPUT [DISPLACEMENTS / STRAINS / STRESSES]

implies that the user may request

OUTPUT DISPL STRAINS

Brackets and braces are combined to allow more flexibility in designing commands. The command syntax

<integer>[{X / Y / Z} <number>]

implies that the user may enter data of the form:

1 X 0.0 Y 0.0 Z 5.0

2 X 1.0 Z 5.0

Continuation of an input line onto a second physical line is accomplished by placing a comma at the end of the line to be continued.

Comments may be placed in the data by placing a C in column 1 and a blank in column 2 of the comment line.

One method for line termination is to place a dollar sign "$" on the line. Space on the line following the "$" is ignored by the translator and and may be used for comments.

7.3.2 Command Syntax

Specification of Mass

Example of the command to specify primary mass:

```
ELEMENT 1 TYPE CSTRIANGLE CONSISTENT E 1 NU .3 DENSITY .00074
```

Example of the commands to specify secondary mass:

```
MASS
  NODAL
    2 U V W  20.0   THETAX THETAY   5.0
  ELEMENT MASS FOR TYPE PLANEFRAME
    3 LINEAR  U V W  FRACTIONAL LA 0.25 LB 0.75 WA 3.0 WB 8.0
    1 CONCENTRATED  U V W  L 3.6 M 5.0
    2 CONCENTRATED  THETAZ     L 3.6 M 3.0
  USE LOADING DEAD_LOAD  G 386.4
```

Assembly command:

COMPUTE MASS (FOR) { STRUCTURE / ELEMENT } <label>

Output command:

OUTPUT MASS (FOR) { STRUCTURE / ELEMENT } <label>

Specification of Damping

Modal damping:

DAMPING MODAL { RATIOS / PERCENTS } [<mode list> <number>]

Rayleigh damping:

DAMPING RAYLEIGH [{ FREQUENCIES / PERIOD } <number> <number>
{ RATIOS / PERCENTS } <number> <number>]

Output command:

OUTPUT DAMPING { MATRIX / RATIOS / PERCENTS } ((FOR) STRUCTURE <label>) (,)

((FOR) MODES <integer list>)

Units of seconds for time and rad/sec for frequencies will be required.

Specification of Frequency Analysis

Definition of the frequency analysis method:

FREQUENCY ANALYSIS (TYPE) { HQRI / JACOBI / SUBSPACE / NEWTON / . / . / . }

PROPERTIES <define properties unique to each type>

Computation request:

COMPUTE (NONLINEAR) [(NATURAL) FREQUENCIES / (MODE) SHAPES] ((FOR) STRUCTURE (,)

<label>) ((FOR) LOADING <label> (TIME) STEPS <integer list>)

Standard output request:

```
OUTPUT (NONLINEAR)[(NATURAL) FREQUENCIES]((FOR) STRUCTURE <label> (,)
                  [(MODE) SHAPES       ]

      [(FOR) LOADING <label> (TIME) STEPS <integer list>]
      [                                                 ]
      [(FOR) MODES <integer list>                       ]
```

Modal loads output request:

```
OUTPUT MODAL LOADS ((FOR) STRUCTURE <label>)  (,)

      [(FOR) LOADING <label>     ]
      [(FOR) MODES   <integer list>]
```

Specification of User-supplied Mode Shapes

Command sequence:

```
ALTERNATE (MODES)  <label>  ((TITLE) <string>)

 <specification of DOF order: U V W UX ...>

[MODE <mode number:integer>                          ]
[                                                    ]
[[<node number:integer> [<DOF value:number>]]        ]
```

Specification of Dynamic Reduction

Element declaration for intermediate level structure:

```
ELEMENT 1 TYPE <structure name> CONDENSED (,)

{ RETAINED {MODES  <integer list>} }
{          {NODES  <integer>      } }
{                                   }
{ USE ALTERNATE (MODES) <label>     }
```

Specification of Initial Conditions

Command sequence:

```
INITIAL CONDITIONS <label> ((TITLE) <string>)
{ [ DISPLACEMENTS
        [<node list> <DOF list> = <number>] ]
  [ VELOCITIES
        [<node list> <DOF list> = <number>] ]
  USE (NONLINEAR) DISPLACEMENTS ((FOR) STRUCTURE <label>) (,)
        (FOR) LOADING <label> (STEP <integer>) }
```

Specification of Dynamic Loading Condition

Input of support accelerations as F(x,y,z):

```
LOADING <label> ((TITLE) <string>)
  (NODAL) ACCELERATIONS
    [<node list> <DOF list> <number>]
```

Definition of loading condition:

```
LOADING <label> ((TITLE) <string>)
  [ DYNAMIC  ]
  [ NONLINEAR ]
```

Definition of G(t):

For a harmonic variation of G(t):

```
HARMONIC PERIOD <number> (PHASE (ANGLE) <number>) (,)
   (COMBINE) [ <label> (FACTOR) <number> (.) ]
```

For a general variation of G(t):

```
GENERAL (COMBINE) [ <label> [ { TIMES   } <number list> ] ]
                            { FACTORS }
```

For an impulsive variation of G(t):

```
IMPULSIVE (SHAPE) { HALF_SINE       } DURATION <number>  (,)
                  { RECTANGULAR     }
                  { POS_TRIANGULAR  }
                  { NEG_TRIANGULAR  }

    (COMBINE)  [ <label> (FACTOR) <number> ]
```

Step size definition:

```
[ (TIME) STEPS <integer list> ((TITLE) <string>) <number list>  (,)

    (SECONDS) ]
```

Definition of results saved in the data base:

```
(SAVE (TIME) STEPS <integer list>)
```

Note that the last step computed is always saved, even if not in the integer list or if the command is not given.

Specification of Transient Analysis

Definition of the transient analysis method:

```
TRANSIENT ANALYSIS (TYPE) { MODE_SUPERPOSITION }
                          { NEWMARK            }
                          { CENTRAL_DIFFERENCE }
                          {        .           }
                          {        .           }
                          {        .           }

    PROPERTIES <define properties unique to each type>
```

Computation request:

```
COMPUTE [NONLINEAR] DISPLACEMENTS ((FOR) STRUCTURE <label>)  (,)
        [DYNAMIC  ]

    [ LOADING <label> (TIME) STEPS <integer list>  ]
    [                                              ]
    [ INITIAL CONDITIONS <label>                   ]
    [                                              ]
    [ INCLUDE MODES <integer list>                 ]
```

Output request:

```
OUTPUT [DYNAMIC  ] [DISPLACEMENTS] (<integer list>)    (,)
       [NONLINEAR]  [VELOCITIES   ]
                    [STRAINS      ]
                    [STRESSES     ]

    ((FOR) STRUCTURE <label>)      (,)

    (FOR) LOADING <label> (TIME) STEPS <integer list>
```

Specification of Shock Spectrum Analysis

Definition of the spectrum:

```
(SHOCK) SPECTRUM <label> ((TITLE) <string>)

    [ { DISPLACEMENTS } <number list> ]
    [ { VELOCITIES    }               ]
    [ { ACCELERATIONS }               ]
    [                                 ]
    [ { PERIODS     } <number list>   ]
    [ { FREQUENCIES }                 ]

  DIRECTIONS (,)

        <node list> [ { U } <direction cosine:number> ]
                    [ { V }                           ]
                    [ { W }                           ]
```

Output request:

```
OUTPUT DYNAMIC [DISPLACEMENTS      ] (<integer list>) (,)
               [VELOCITIES         ]
               [STRESSES           ]
               [STRAINS            ]
               [PARTICIPATION_FACTORS]

  ((FOR) STRUCTURE <label>) (,)

  [(FOR) (SHOCK) SPECTURM <label>]
  [                               ]
  [(FOR) MODES [<integer list>]   ]
  [            [SRSS          ]   ]
  [            [PEAK_SRSS     ]   ]
```

Specification of Utility Commands

Trace command:

TRACE [NONLINEAR / DYNAMIC] (SOLUTION)

Destroy command:

DESTROY [NONLINEAR / DYNAMIC] RESULTS (FOR) STRUCTURE <label> (,)

{ (FOR) LOADING <label> (TIME) STEPS <integer list> / (FOR) (SHOCK) SPECTRUM <label> / (FOR) MODES <integer list> }

7.4 Sample Input

The following sections contain example input data illustrating the use of the foregoing commands for dynamic analysis of some simple plane structures. Each example problem is liberally commented to explain the analysis process. The substructured nonlinear analysis of Section 7.4.5 deserves additional discussion and is described in detail in that section.

7.4.1 Standard Linear Structure - Vibration Analysis

```
*RUN FINITE
C
C
C               EXAMPLE INPUT NO. 1 FOR DYNAMIC ANALYSIS
C               ========================================
C
C               (REFERENCE FIGURE 7.1)
C
C               THIS EXAMPLE ILLUSTRATES THE INPUT NECESSARY TO DESCRIBE
C               A LINEAR, THREE ELEMENT, PLANE FRAME AND TO PERFORM
C               A FREQUENCY ANALYSIS OF THE STRUCTURE.  IN ANTICIPATION
C               OF A TRANSIENT ANALYSIS BY MODE SUPERPOSITION. THE
C               LOADS ARE OUTPUT IN MODAL COORDINATES.  THE PROBLEM IS
C               RESTARTED AND THE TRANSIENT ANALYSIS IS INVOKED WITH ONLY
C               SELECTED MODES INCLUDED.  INITIAL CONDITIONS ARE ALSO
C               DEFINED AND INCORPORATED INTO THE TRANSIENT ANALYSIS.
C
C
  STRUCTURE FRAME
   NUMBER OF ELEMENTS 3 NODES 4
     ELEMENTS 1,3 TYPE PLANEFRAME LUMPED E 30000 G 12000 AX 20. ,
                  AY 5.877 IZ 724  DENSITY 0.00074
     ELEMENT 2 TYPE PLANEFRAME CONSISTENT E 30000 G 12000 AX 14.4,
                  AY 3.4 IZ 273.   DENSITY 0.00074
C
   COORDINATES
     1    0.0    0.0
     2    0.0   96.0
     3   96.0   96.0
     4   96.0    0.0
C
   INCIDENCES
     1  1  2
```

```
     2  2  3
     3  3  4
C
   CONSTRAINTS
     1,4  ALL = 0.0
C
   LOADING MOTOR
    ELEMENT LOADS FOR TYPE PLANEFRAME
      2  CONCENTRATED FORCE Y  P 300.0  L 48.0
C
   MASS
      NODAL MASS
        2,3  U V W   1.0
      ELEMENT MASS FOR TYPE PLANEFRAME
        2 UNIFORM  U V W  W  0.05    $ THE SECOND W INDICATES AN INTENSITY
      USE LOADING MOTOR  G 386.4
C
C              STATIC LOADING PATTERN:  F(X,Y,Z)
C
   LOADING PATTERN
    ELEMENT LOADS FOR TYPE PLANEFRAME
     1 LINEAR FORCE Y LA 0.0  LB 1.0  WA 0.0 WB 1.0
C
C              DYNAMIC LOADING CONDITION:  G(T)
C
   LOADING SHAKE
    DYNAMIC
     HARMONIC PERIOD 0.04 PHASE 0.0 COMBINE PATTERN 6.0
     TIME STEPS 1-100  0.0 TO 10.0 BY 0.1
     SAVE STEPS 10-100 BY 10
C
C              SPECIFY FREQUENCY ANALYSIS PARAMETERS
C
   FREQUENCY ANALYSIS TYPE SUBSPACE
       PROPERTIES
         CONVERGENCE TOLERANCE  1.0E-08,
         MAXIMUM ITERATIONS 13,
         SHIFT EVERY 2 MODES,
         MAXIMUM MODES 4,
         RANGE MIN 0.0  MAX 50.0
C
C              REQUEST COMPUTATION OF FREQUENCIES AND MODES
C
   COMPUTE NATURAL FREQUENCIES MODE SHAPES STRUCTURE FRAME
C
   OUTPUT FREQUENCIES SHAPES STRUCTURE FRAME MODES 1-4
   OUTPUT MODAL LOADS STRUCTURE FRAME LOADING SHAKE /MODES 1-4
C
   STOP
C
C
C
*RUN FINITE FILES = 20,,22,23
C
```

```
C
C               RESTART OF THE LINEAR PLANEFRAME TO PERFORM THE
C               TRANSIENT ANALYSIS AFTER EXAMINING THE MODAL LOADS.
C
C
  ACCESS STRUCTURE FRAME
C
C               SPECIFY TRANSIENT ANALYSIS PARAMETERS
C                    (USE DEFAULT PROPERTIES)
C
   TRANSIENT ANALYSIS TYPE MODE_SUPERPOSITION
C
C
C
C          DEFINE INITIAL CONDITIONS FOR DISPLACEMENTS.
C          (VELOCITIES NOT REQUIRED SINCE WE HAVE NO DAMPING.)
C
     INITIAL CONDITIONS DEAD_SHAPE
       DISPLACEMENTS
         2,3  U = 0.1
         2    THETAZ = -0.085
         3    THETAZ =  0.085
C
    COMPUTE DYNAMIC DISPLACEMENTS STRUCTURE FRAME LOADING SHAKE,
            TIME STEPS 1-25 INITIAL CONDITIONS DEAD_SHAPE INCLUDE,
            MODES 1, 3, 4
C
    OUTPUT DYNAMIC DISPLACEMENTS 2,3 STRUCTURE FRAME LOADING SHAKE,
        STEPS 10, 20, 25      $ RECALL THAT STEP 25 IS SAVED EVEN THOUGH
C                               I DIDN'T REQUEST IT.
C
    OUTPUT WIDE BY ELEMENT DYNAMIC STRESSES STRAINS ALL STRUCTURE FRAME,
           LOADING SHAKE STEPS 10, 20, 25
C
    STOP
```

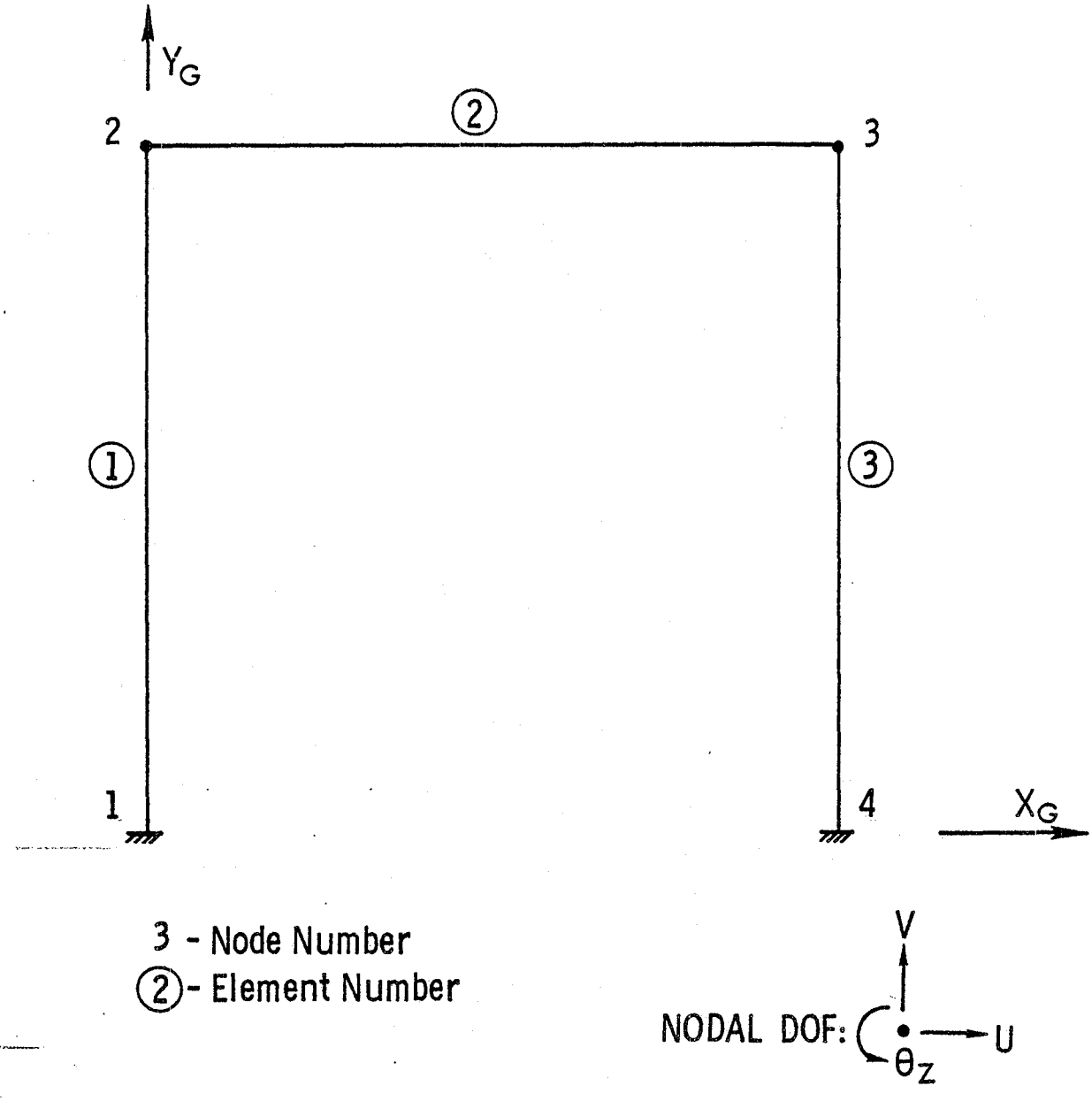


Figure 7.1 -- Three Element, Plane Frame (Example Input #1-3)

7.4.2 Standard Linear Structure - Shock Spectrum Analysis

```
*RUN FINITE
C
C
C               EXAMPLE INPUT NO. 2 FOR DYNAMIC ANALYSIS
C               ========================================
C
C               (REFERENCE FIGURE 7.1)
C
C               THIS EXAMPLE ILLUSTRATES THE INPUT NECESSARY TO DESCRIBE
C               A LINEAR, THREE ELEMENT, PLANE FRAME AND TO PERFORM
C               A SHOCK SPECTRUM ANALYSIS OF THE STRUCTURE.  THE SHOCK
C               SPECTRUM CAN CONTAIN BOTH HORIZONTAL AND VERTICAL
C               COMPONENTS.  WE CAN GET THE PARTICIPATION FACTORS
C               (ALSO KNOWN AS EFFECTIVE MODAL MASS) PRIOR TO REQUESTING
C               THAT SPECTRAL DISPLACEMENTS, STRESSES, AND STRAINS BE
C               COMPUTED.  TWO METHODS ARE AVAILABLE FOR SUMMING THE
C               SPECTRAL VALUES FOR EACH MODE:  SRSS AND PEAK_SRSS.
C               SRSS IS THE SQUARE-ROOT-OF-THE-SUM-OF-THE-SQUARES METHOD.
C               PEAK_SRSS TAKES THE MODE WITH THE LARGEST PARTICIPATION
C               FACTOR AND ADDS TO THAT THE SRSS OF THE REMAINING MODES
C               THAT ARE INCLUDED IN THE ANALYSIS.
C
C
C
C
  STRUCTURE FRAME
   NUMBER OF ELEMENTS 3 NODES 4
     ELEMENTS 1,3 TYPE PLANEFRAME LUMPED E 30000 G 12000 AX 20. ,
                  AY 5.877 IZ 724   DENSITY 0.00074
     ELEMENT 2 TYPE PLANEFRAME CONSISTENT E 30000 G 12000 AX 14.4 ,
                  AY 3.4 IZ 273.   DENSITY 0.00074
C
   COORDINATES
     1   0.0   0.0
     2   0.0   96.0
     3   96.0   96.0
     4   96.0   0.0
C
   INCIDENCES
     1  1  2
     2  2  3
     3  3  4
C
   CONSTRAINTS
     1,4  U V = 0.0
C
   LOADING MOTOR
    ELEMENT LOADS FOR TYPE PLANEFRAME
      2  CONCENTRATED FORCE Y  P 300.0  L 48.0
C
   MASS
```

```
      NODAL MASS
        2,3  U V W  1.0
      ELEMENT MASS FOR TYPE PLANEFRAME
        2 UNIFORM   U V W    W   0.05
      USE LOADING MOTOR  G 386.4
C
   SHOCK SPECTRUM FAIL_SAFE "5 PERCENT DAMPING"
         VELOCITIES  2  30  30 12
         PERIODS   .05 .60 4.5 10.
C
         DIRECTIONS 1,4  U 0.866  V 0.5
C
C
C
C              SPECIFY FREQUENCY ANALYSIS PARAMETERS
C
   FREQUENCY ANALYSIS TYPE SUBSPACE
       PROPERTIES
         CONVERGENCE TOLERANCE  1.0E-08,
         MAXIMUM ITERATIONS 13,
         SHIFT EVERY 2 MODES,
         MAXIMUM MODES 4,
         RANGE MIN 0.0  MAX 50.0
C
C              REQUEST COMPUTATION OF FREQUENCIES AND MODES
C
   COMPUTE NATURAL FREQUENCIES MODE SHAPES STRUCTURE FRAME
C
   OUTPUT FREQUENCIES SHAPES STRUCTURE FRAME MODES 1-4
   OUTPUT DYNAMIC PARTICIPATION FACTORS STRUCTURE FRAME SHOCK,
          SPECTRUM FAIL_SAFE  MODES ALL
C
   STOP
C
C
C
*RUN FINITE FILES = 20,,22,23
C
C
C              RESTART OF THE LINEAR PLANEFRAME TO PERFORM THE
C              SHOCK SPECTRUM ANALYSIS AND COMPUTE SPECTRAL STRESS
C              AND STRAINS.  AT THIS POINT, WE HAVE HAD AN OPPORTUNITY
C              TO EXAMINE THE PARTICIPATION FACTORS AND SEE THAT ONLY
C              THREE OF THE MODES HAVE ANY SIGNIFICANT CONTRIBUTION TO
C              THE SPECTRAL RESPONSE.
C
C
  ACCESS STRUCTURE FRAME
C
C
     OUTPUT DYNAMIC DISPLACEMENTS STRESSES STRAINS STRUCTURE FRAME,
            SPECTRUM FAIL SAFE MODES 1-3 SRSS PEAK SRSS
C
    STOP
```

7.4.3 Standard Nonlinear Structure

```
*RUN FINITE
C
C
C              EXAMPLE INPUT NO. 3 FOR DYNAMIC ANALYSIS
C              ========================================
C
C              (REFERENCE FIGURE 7.1)
C
C              THIS EXAMPLE ILLUSTRATES THE INPUT NECESSARY TO DESCRIBE
C              A NONLINEAR, THREE ELEMENT, PLANE FRAME AND TO PERFORM
C              A TRANSIENT ANALYSIS OF THE STRUCTURE.  AT SOME SELECTED
C              TIME STEP, THE TRANSIENT ANALYSIS IS SUSPENDED AND A
C              FREQUENCY ANALYSIS IS PERFORMED WITH THE CURRENT STRUCTURE
C              STIFFNESS AND MASS.  THE GENERALIZED NEWMARK OPERATOR IS
C              USED FOR THE TIME-HTORY INTEGRATION.  THIS OPERATOR
C              CAN BE USED AS EITHER AN EXPLICIT OR IMPLICIT INTEGRATOR
C              AND HAS THE ABILITY TO CONTROL SPURIOUS DAMPING.
C
C
   MATERIAL STEEL TYPE VON_MISES
     PROPERTIES  SIGNAL_YIELD
     USE STRESS-STRAIN FUNCTION SEGMENTAL
         PROPERTIES E 30000. NU 0.3 STRAIN_HARDENING TENSYIELD 30,
            COMPYIELD 30. TENSION_SLOPE 3000. COMP_SLOPE 3000.
C
C
  STRUCTURE FRAME
   NUMBER OF ELEMENTS 3 NODES 4
     ELEMENTS 1,3 TYPE PLANEFRAME LUMPED E 30000 G 12000 AX 20. ,
                    AY 5.877 IZ 724   DENSITY 0.00074
     ELEMENT 2 TYPE PLANEFRAME CONSISTENT MATERIAL STEEL E 30000 ,
         G 12000 AX 14.4 AY 3.4 IZ 273. DENSITY 0.00074
C
   COORDINATES
     1   0.0   0.0
     2   0.0   96.0
     3   96.0   96.0
     4   96.0   0.0
C
   INCIDENCES
     1  1  2
     2  2  3
     3  3  4
C
   CONSTRAINTS
     1.4  ALL = 0.0
C
   LOADING MOTOR
    ELEMENT LOADS FOR TYPE PLANEFRAME
      2  CONCENTRATED FORCE Y  P 300.0  L 48.0
C
```

```
   MASS
      NODAL MASS
        2,3   U V W   1.0
      ELEMENT MASS FOR TYPE PLANEFRAME
        2 UNIFORM   U V W    W   0.05
      USE LOADING MOTOR  G 386.4
C
   LOADING PATTERN
    ELEMENT LOADS FOR TYPE PLANEFRAME
     1 LINEAR FORCE Y LA 0.0  LB 1.0  WA 0.0 WB 1.0
C
   LOADING SHAKE
    NONLINEAR DYNAMIC
     HARMONIC PERIOD 0.04 PHASE 0.0 COMBINE PATTERN 6.0
     TIME STEPS 1-100  0.0 TO 10.0 BY 0.1
     SAVE STEPS 10-100 BY 10
C
C              SPECIFICATION OF NONLINEAR SOLUTION PARAMETERS
C
   TRACE NONLINEAR SOLUTION
   CONVERGENCE TEST NORM RESIDUAL LOADS TOLER 1.5 INCLUDE TOTAL,
       REACTIONS
   UPDATE STIFFNESS EVERY STEP
   TERMINATE IF NONCONVERGENT
C
C              SPECIFICATION OF TRANSIENT ANALYSIS SOLUTION PARAMETERS
C
   TRANSIENT ANALYSIS TYPE NEWMARK
     PROPERTIES ALPHA 0.0 BETA 0.5 GAMMA 0.25
C
C              REQUESTS FOR COMPUTATION AND OUTPUT.
C
  COMPUTE NONLINEAR DYNAMIC DISPLACEMENTS STRUCTURE FRAME LOADING,
      SHAKE TIME STEPS 1-30
C
   OUTPUT DYNAMIC NONLINEAR STRESSES 2 STRUCTURE FRAME LOADING SHAKE,
       STEPS 10,20,30
C
   STOP
C
C
C
*RUN FINITE FILES = 20,,22,23
C
C
C              RESTART OF NONLINEAR PLANE FRAME TO PERFORM THE
C              FREQUENCY ANALYSIS AND GET THE NATURAL FREQUENCIES
C              AND MODE SHAPES.
C
C
  ACCESS STRUCTURE FRAME
C
C
C              SPECIFICATION OF FREQUENCY ANALYSIS
```

```
C
  FREQUENCY ANALYSIS TYPE JACOBI
   PROPERTIES
    MAX_SWEEPS 15,
    CONVERGENCE TOLERANCE 1.0E-08
C
C
C               REQUEST COMPUTATION AND OUTPUT OF FREQUENCIES AND MODES
C
   COMPUTE NONLINEAR NATURAL FREQUENCIES MODE SHAPES STRUCTURE FRAME,
       LOADING SHAKE TIME STEP 20
C
   OUTPUT NONLINEAR FREQUENCIES SHAPES STRUCTURE FRAME MODES ALL,
       LOADING SHAKE STEP 20
C
C               AT THIS POINT, WE CAN "ACCESS STRUCTURE..." AGAIN
C               AND CONTINUE WITH THE TRANSIENT ANALYSIS
C
   STOP
```

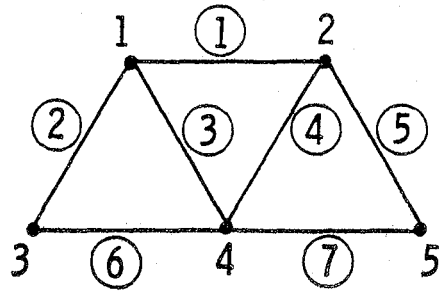
7.4.4 Substructured Linear Analysis

```
*RUN FINITE
C
C
C               EXAMPLE INPUT NO. 4 FOR DYNAMIC ANALYSIS
C               ========================================
C
C               (REFERENCE FIGURE 7.2)
C
C               THIS EXAMPLE ILLUSTRATES THE INPUT NECESSARY TO DESCRIBE
C               A LINEAR, MULTILEVEL SUBSTRUCTURED MODEL AND TO PERFORM
C               A FREQUENCY ANALYSIS OF THE HIGHEST LEVEL STRUCTURE.
C               FEATURES OF DYNAMICS ILLUSTRATED ARE GUYAN REDUCTION AND
C               MODAL SYNTHESIS FOR CONDENSING THE SUBSTRUCTURES AND THE
C               METHOD FOR CARRYING FORWARD SUBSTRUCTURE MASS (IE. NO
C               SPECIAL CONSIDERATION IS GIVEN TO BRINGING UP MASS).
C
C               THE ENTIRE STRUCTURAL SYSTEM IS BUILT OUT OF ONE STAND-
C               ALONE ELEMENT. A PLANE FRAME ELEMENT.
C
C
  ELEMENT BAR TYPE PLANEFRAME CONSISTENT E 30000. G 12000. AX 20.0,
      AY 5.877  IZ 724 DENSITY 0.00074
C
  COORDINATES
    1   0.0   0.0
    2  96.0   0.0
C
C               DEFINE LOWEST LEVEL STRUCTURE, A TRUSS WITH THREE BAYS.
C
  STRUCTURE THREE_BAY
   NUMBER OF ELEMENTS 7 NODES 5
    ELEMENTS ALL TYPE BAR
     1, 6, 7  ROTATION SUPPRESSED
     2, 4     ROTATION Z  60.
     3. 5     ROTATION Z -60.
C
  INCIDENCES
    1   1 2
    2   3 1
    3   1 4
    4   4 2
    5   2 5
    6   3 4
    7   4 5
C
C               ADD MASS TO ONE OF THE LOWER CHORDS.
C
   MASS
    ELEMENT MASS FOR TYPE PLANEFRAME
     7  UNIFORM    U V W     W  0.0003
C
```

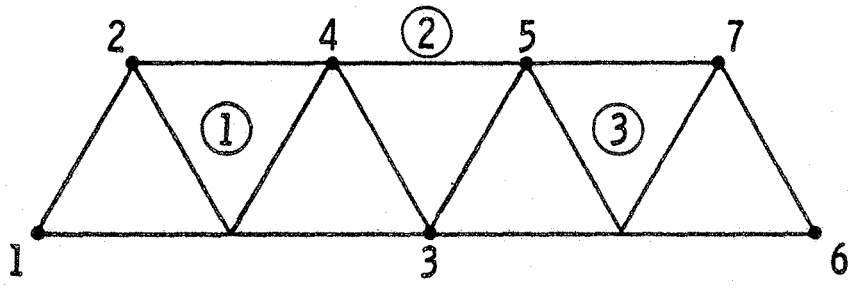
```
C
C               CONDENSE OUT NODE 4 USING GUYAN REDUCTION
C
  STRUCTURE THREE_CON
   NUMBER OF ELEMENTS 1 NODES 4
    ELEMENT 1 TYPE THREE_BAY CONDENSED
C
   INCIDENCES
    1   3 1 5 2
C
C
C               STICK TWO OF THE FRAMES TOGETHER AND CLOSE THE GAP
C               AT THE TOP WITH A BAR ELEMENT.
C
   STRUCTURE SPAN
    NUMBER OF ELEMENTS 3 NODES 7
     ELEMENTS
      1, 3 TYPE THREE_CON ROTATION SUPPRESSED
      2    TYPE BAR       ROTATION SUPPRESSED
C
     INCIDENCES
       1  1 2 3 4
       2  4 5
       3  3 5 6 7
C
C               ADD A LITTLE MORE MASS TO SOME SELECTED NODES
C
  MASS
   NODAL MASS
    4, 5   U V W     0.5
C
C               DEFINE FREQUENCY ANALYSIS PARAMETERS FOR THIS STRUCTURE
C               SINCE IT WILL BE CONDENSED USING MODAL SYNTHESIS
C
   FREQUENCY ANALYSIS TYPE SUBSPACE
       PROPERTIES
          CONVERGENCE TOLER 1.0E-08,
          MAX ITERATIONS  10,
          MAX MODES 5
C
C
C               CONDENSE SPAN VIA MODAL SYNTHESIS
C
  STRUCTURE SPAN_CON
     NUMBER OF ELEMENTS 1 NODES 2
       ELEMENT 1 TYPE SPAN CONDENSED RETAINED MODES 5
C
  INCIDENCES
    1   1 6
C
C
C               BUILD THE TWO SPAN BRIDGE
C               NOTE THAT THE THETAZ DOF AT THE MIDDLE SUPPORT KEEPS
C               THIS STRUCTURE FROM BECOMING TWO SIMPLE SPANS.
```

```
C
C               ALSO NOTE THAT ALTHOUGH THIS STRUCTURE ONLY HAS 3 NODES
C               WITH 3 GEOMETRIC DOF EACH, IT HAS 10 GENERALIZED DOF FROM
C               THE RETAINED NORMAL MODES FOR A TOTAL OF 19 DOF.
C
   STRUCTURE TWO_SPAN
    NUMBER OF ELEMENTS 2 NODES 3
     ELEMENTS ALL TYPE SPAN_CON ROTATION SUPPRESSED
C
   INCIDENCES
    1   1  2
    2   2  3
C
  CONSTRAINTS
    1   U V = 0.0
    2 3   V = 0.0
C
  FREQUENCY ANALYSIS TYPE SUBSPACE
    PROPERTIES
      CONVERGENCE TOLER 1.0E-08,
      MAX ITERATIONS 10,
      MAX MODES      10
C
   COMPUTE FREQUENCIES  STRUCTURE TWO_SPAN
C
   OUTPUT FREQUENCIES SHAPES STRUCTURE TWO_SPAN MODES ALL
C
   STOP
```

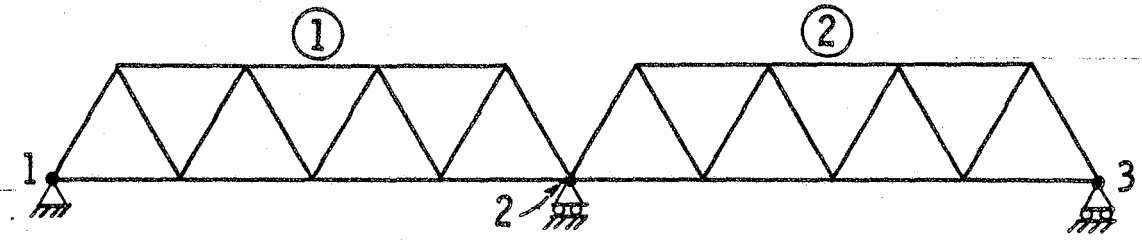
Element Bar



Structure Three - Bay



Structure Span



Structure Two-Span

4 - Node Number

(6) - Element Number

NODAL DOF:

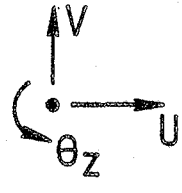


Figure 7.2 -- Two Span Bridge (Example Input #4)

7.4.5 Substructured Nonlinear Analysis

The input listing in this section describes a simple, two-span bridge. Nonlinearity is introduced into the example by addition of a nonlinear bar element over the center support. The example uses six levels of substructures with the nonlinear element added at the highest level. To facilitate the description of the input sequence, line numbers are placed before each FINITE command line. The numbers are not part of the commands. They serve only as reference numbers. Comment lines are not numbered. The structure is illustrated in Figure 7.3.

Lines 2-6 define a stand-alone element, which is used to construct the majority of the final structure. Lines 7-23 describe the lowest level substructure called THREE_BAY. No frequency analysis parameters are defined since this structure will be condensed using Guyan reduction. The condensed version of THREE_BAY is named structure PIECE. No additional input is required to invoke the condensation process; Guyan reduction is the default procedure adopted by the system.

Lines 29-48 describe structure HALF which contains two copies of substructure PIECE and one copy of stand-alone element BAR. To illustrate its use in substructures, secondary mass is applied in lines 38-40. A frequency analysis method is defined, lines 44-48, so the structure can be condensed by modal synthesis. Structure HALF_CON. lines 49-56, is the condensed version of structure HALF. It is necessary to carry forward the loads from HALF but the mass is automatically included in the reduction process.

A nonlinear material model is defined in lines 57-61. This model is required for the nonlinear element used in structure BRIDGE. This highest level structure is composed of two condensed substructures, HALF_CON. and one simple element, TYPE PLANEFRAME (see lines 62-67). Damping and dynamic loading are defined in lines 78-88 with the transient analysis specification and requests for computation and output in lines 89-96.

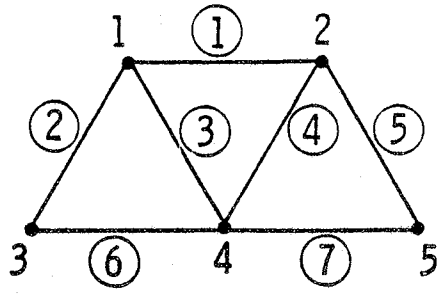
```
 1  *RUN FINITE
    C
    C
    C            EXAMPLE INPUT NO. 5 FOR DYNAMIC ANALYSIS
    C            ========================================
    C
    C            (REFERENCE FIGURE 7.3)
    C
    C            THIS EXAMPLE ILLUSTRATES THE INPUT NECESSARY TO DESCRIBE
    C            A NONLINEAR, MULTILEVEL SUBSTRUCTURED MODEL AND TO PERFORM
    C            A TRANSIENT ANALYSIS OF THE STRUCTURE WHEN IT IS SUBJECTED
    C            TO A GENERAL DYNAMIC LOADING.  THE NONLINEARITY IS
    C            RESTRICTED TO A MATERIALLY NONLINEAR ELEMENT AT THE HIGHEST
    C            LEVEL.  THE STRUCTURE IS THE TRUSS FROM EXAMPLE NO. 4 WITH
    C            THE GAP OVER THE CENTER SUPPORT CLOSED BY THE NONLINEAR
    C            ELEMENT.
    C
    C
 2    ELEMENT BAR TYPE PLANEFRAME CONSISTENT E 30000. G 12000. AX 20.0,
 3        AY 5.877  IZ 724 DENSITY 0.00074
    C
 4    COORDINATES
 5      1   0.0   0.0
 6      2  96.0   0.0
    C
    C              DEFINE LOWEST LEVEL STRUCTURE, A TRUSS WITH THREE BAYS.
    C
 7    STRUCTURE THREE_BAY
 8     NUMBER OF ELEMENTS 7 NODES 5
 9      ELEMENTS ALL TYPE BAR
10       1, 6, 7  ROTATION SUPPRESSED
11       2, 4     ROTATION Z  60.
12       3, 5     ROTATION Z -60.
    C
13    INCIDENCES
14      1   1 2
15      2   3 1
16      3   1 4
```

```
17        4   4 2
18        5   2 5
19        6   3 4
20        7   4 5
      C
      C                ADD MASS TO ONE OF THE LOWER CHORDS.
      C
21      MASS
22       ELEMENT MASS FOR TYPE PLANEFRAME
23        7  UNIFORM  U V W     W 0.0003
      C
      C
      C                CONDENSE OUT NODE 4 USING GUYAN REDUCTION
      C
24    STRUCTURE PIECE
25     NUMBER OF ELEMENTS 1 NODES 4
26      ELEMENT 1 TYPE THREE_BAY CONDENSED
      C
27      INCIDENCES
28       1   3 1 5 2
      C
      C
      C                STICK TWO OF THE FRAMES TOGETHER AND CLOSE THE GAP
      C                AT THE TOP WITH A BAR ELEMENT.
      C
29      STRUCTURE HALF
30       NUMBER OF ELEMENTS 3 NODES 7
31        ELEMENTS
32         1, 3 TYPE PIECE ROTATION SUPPRESSED
33         2    TYPE BAR   ROTATION SUPPRESSED
      C
34        INCIDENCES
35          1  1 2 3 4
36          2  4 5
37          3  3 5 6 7
      C
      C                ADD A LITTLE MORE MASS TO SOME SELECTED NODES
      C
38    MASS
39     NODAL MASS
40      4, 5  U V W    0.5
      C
      C                ADD THE PATTERN OF LOAD TO BE USED IN THE DYNAMIC
      C                LOADING
      C
41    LOADING CENTER_SPAN
42     NODAL LOADS
43      3  FORCE Y P -1.0
      C
      C
      C                DEFINE FREQUENCY ANALYSIS PARAMETERS FOR THIS STRUCTURE
      C                SINCE IT WILL BE CONDENSED USING MODAL SYNTHESIS
      C
44      FREQUENCY ANALYSIS TYPE SUBSPACE
```

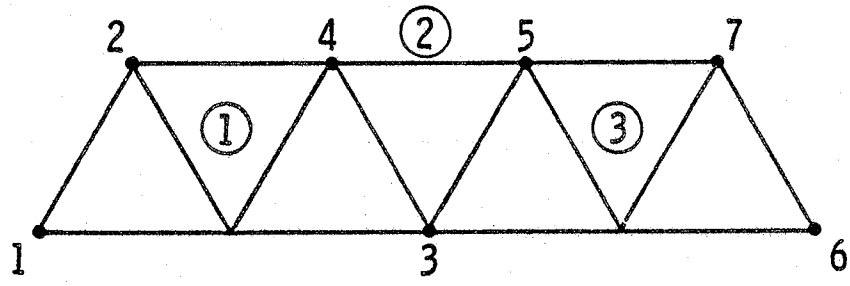
```
45          PROPERTIES
46             CONVERGENCE TOLER 1.0E-08,
47             MAX ITERATIONS  10.
48             MAX MODES 5
    C
    C
    C                CONDENSE HALF VIA MODAL SYNTHESIS
    C
49     STRUCTURE HALF_CON
50        NUMBER OF ELEMENTS 1 NODES 4
51          ELEMENT 1 TYPE HALF CONDENSED RETAINED MODES 3
    C
52     INCIDENCES
53       1    1 2 6 7
    C
    C                CARRY FORWARD THE LOADS FROM HALF
    C
54     LOADING CENTER_CON
55      EXTERNAL ELEMENT LOADS
56       1    CENTER_SPAN  1.0
    C
    C
    C
    C                DEFINE THE NONLINEAR MATERIAL
    C
57     MATERIAL STEEL TYPE VON_MISES
58      PROPERTIES  SIGNAL YIELD
59      USE STRESS-STRAIN FUNCTION SEGMENTAL
60       PROPERTIES  E 1 NU 0 STRAIN HARDENING,
61          TENSYIELD 1 TENSION_SLOPE .1
    C
    C
    C                BUILD THE TWO SPAN BRIDGE
    C
    C                CLOSE THE GAP OVER THE CENTER SUPPORT WITH A
    C                NONLINEAR BAR.
    C
62      STRUCTURE BRIDGE
63       NUMBER OF ELEMENTS 3 NODES 7
64        ELEMENTS
65           1, 3 TYPE HALF_CON ROTATION SUPPRESSED
66           2    TYPE PLANEFRAME MATERIAL STEEL CONSISTENT AX 20.0 ,
67                     AY 5.877  IZ 724 DENSITY 0.00074
    C
68      COORDINATES
69        4   0.0  0.0
70        5  96.0  0.0
    C
71      INCIDENCES
72        1    1 2 3 4
73        2    4 5
74        3    3 5 6 7
    C
    C
```

```
75     CONSTRAINTS
76       1    U V = 0.0
77       3 6    V = 0.0
       C
       C                APPLY DAMPING TO THE STRUCTURE BY USING RAYLEIGH DAMPING
       C
78     DAMPING  RAYLEIGH  FREQUENCIES  2.0  12.0  PERCENTS  1.0  3.4
       C
       C
       C           APPLY THE LOAD PATTERN TO EACH SPAN
       C
79     LOADING PATTERN
80      EXTERNAL ELEMENT LOADS
81       1,3  CENTER_CON 1.0
       C
       C                DEFINE THE DYNAMIC, NONLINEAR LOAD.
       C
82     LOADING SHAKE
83      DYNAMIC NONLINEAR
       C
84       GENERAL COMBINE PATTERN,
85          FACTORS 0.0  100.  40. -50. 100.   0.  50. -90.  0.0,
86          TIMES   0.0   0.2  0.4  0.6  0.8  1.0  1.2  1.4  1.6
       C
87       TIME STEPS 1-100  0.0 TO 1.0 BY .01
88       SAVE STEPS 5-100 BY 5
       C
       C                DEFINE THE TRANSIENT ANALYSIS
       C
89      TRANSIENT ANALYSIS TYPE NEWMARK
90        PROPERTIES
91          ALPHA 0.0,
92          BETA 0.5,
93          GAMMA 0.25
       C
94     COMPUTE NONLINEAR DYNAMIC DISPLACEMENTS STRUCTURE BRIDGE,
95       LOADING SHAKE TIME STEPS 1-25
       C
96     OUTPUT NONLINEAR DISPLACEMENTS  LOADING SHAKE TIME STEPS 5-25 BY 5
       C
       C
97     STOP
```

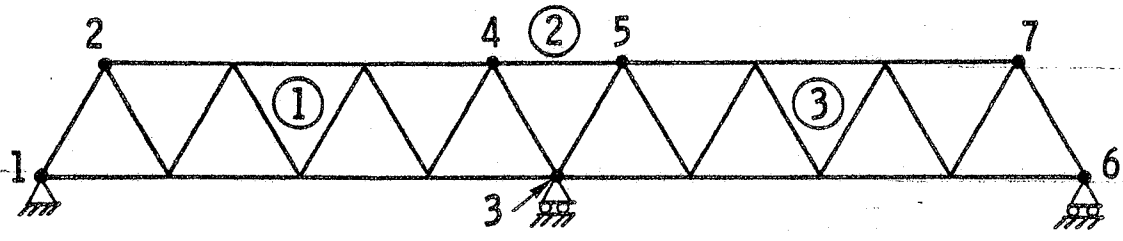
Element Bar



Structure Three - Bay



Structure Half



Structure Bridge

5 — Node Number

(3) - Element Number

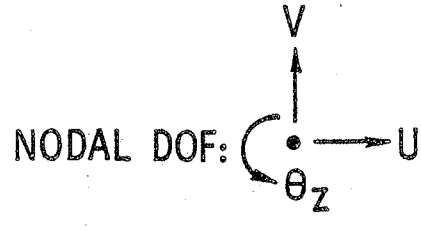


Figure 7.3 -- Two Span Nonlinear Bridge (Example Input #5)

CHAPTER 8

SUMMARY AND TOPICS FOR FURTHER STUDY

8.1 Summary

Comprehensive dynamic analysis of complex structural systems by the finite element method can be an expensive, if not impossible undertaking. Existing software systems capable of achieving some economy suffer a limited scope. The need exists for a general purpose FEM system which is capable of dynamic analysis of arbitrary structures. This capability includes structures experiencing geometric and material nonlinearities. In order to achieve an economical solution, multilevel substructuring is seen as a requisite modeling approach. It is the purpose of this work to bring together the individual, isolated topics of multilevel substructured modeling, dynamic analysis by the FEM, and nonlinear continuum mechanics into the design of a comprehensive, general purpose, finite element package. The resulting software will be used to perform numerical experiments to explore the behavior of the proposed modal synthesis technique in a multilevel substructured environment. The factors studied will include the economics, accuracy, and analyst interaction required to perform modal synthesis.

Implementation of multilevel substructuring for static analysis of linear and nonlinear structures has been discussed in detail. The success of the effort is dependent upon the schemes used for data storage and retrieval, equation solving, and user definition of the model. It was shown that static results are equivalent for both substructured and standard models. Economy in the solution via substructured modeling was

demonstrated in two numerical examples.

Dynamic reduction of the stiffness and mass matrices has been identified as the pivotal process in accurately representing complex structures as simplified models for dynamic analysis. A number of the various methods currently available for dynamic reduction have been identified by a review of the open literature. Guyan reduction and the fixed-interface method have been chosen for incorporation into the general purpose FEM software system.

Eigenproblem solution and transient response analysis are the most computationally expensive operations in the dynamic analysis of structural systems. Their proper implementation and use is essential to the success of the dynamic analysis. A brief review of these processes and an examination of their use in a multilevel substructured environment was given. The most effective eigenproblem solution methods have been identified while transient response analysis was discussed in more general terms.

Using matrix notation, the nonlinear equations of continuum mechanics were derived. Two formulations, Total Lagrangian (T. L.) and Updated Lagrangian (U. L.) were described in detail. Both formulations were shown to derive from a common definition of the rate of work per unit mass and thus should provide identical analysis results. Differences in the computational efficiency of the two formulations were shown to arise in the stress rate transformations and in the complexity of the nonlinear strain-displacement relations. It was concluded that the T. L. formulation has a slight advantage in that no question arises regarding the significance of certain nonlinear terms, i.e., all non-

linear terms must be included in the formulation.

The details of a transient solution procedure for a substructured nonlinear model based upon an implicit integration operator were presented. An implicit scheme was recommended to support dynamic analysis since a static solution procedure can be obtained as the degenerate case of dynamic analysis. Details of the elemental stiffness matrices were derived for both the T. L. and U. L. formulations. Specific matrices were listed for the general 2-D case. Qualitative comparisons of computational efficiency were made and a T. L. approach was recommended for a general software system. The current absence of computational evidence in the literature regarding the performance of a finite element solution based on each approach does not enable the superior approach to be identified. However, an U. L. approach can be easily embedded within a T. L. software system. A T. L. approach cannot be as easily incorporated into a U. L. based system. Thus, the choice of T. L. provides some flexibility for future modifications.

The POLO-FINITE input language has been extended to include the computational features recommended in this report for general purpose dynamic analysis. Wherever possible, consistency has been maintained in the philosophy and method of defining the substructured model. The complete command structure was detailed and examples of its use were presented.

8.2 Topics for Further Study

With the definition of the basic requirements for general dynamic analysis now available, efforts can be directed to software design, implementation, and verification.

The next task to be performed is the design of a prototype software system with the specific goal of demonstrating the applicability of multilevel substructuring in nonlinear dynamic analysis. During the literature review, no evidence was found of this having been attempted at any level of sophistication. Additional software design topics include design of the data structures and processing modules necessary for performing the analysis and specification of the formats for convenient and selective output of results.

Later activities include implementation and testing of the system in the POLO-FINITE structural mechanics software system. The performance of the system will be evaluated over a broad range of structural types including general substructure geometry and linear/nonlinear response.

End of Document