NASA Technical Memorandum 100643

Large-Scale Structural Analysis:

The Structural Analyst, The CSM Testbed, and The NAS System

Norman F. Knight, Jr., Susan L. McCleary,
Steven C. Macy, and Mohammad A. Aminpour

March 1989

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

Preface

A research activity named Computational Structural Mechanics, or CSM, at the NASA Langley Research Center is described. This activity is developing advanced structural analysis and computational methods that exploit high-performance computers. New methods are developed in the framework of the CSM Testbed software system and applied to representative complex structural analysis problems from the aerospace industry. An overview of the CSM Testbed methods development environment is presented. Selected application studies performed on the NAS CRAY-2 computer system are also summarized.

TABLE OF CONTENTS

List of Figures

List of Tables

LARGE-SCALE STRUCTURAL ANALYSIS:

The Structural Analyst, The CSM Testbed, and The NAS System

Norman F. Knight, Jr.†, Susan L. McCleary‡,
Steven C. Macy‡, and Mohammad A. Aminpour*
NASA Langley Research Center
Hampton, Virginia

Introduction

Over the past decade, the structural analyst has had to adapt to a changing computing environment. The computing environment includes software as well as hardware. Research in computational methods for structural analysis has been severely hampered by the complexity and cost of the software development process. Although usually interested in only a small aspect of the overall analysis problem, each researcher is often forced to construct much of the supporting software. This time-consuming and expensive approach is frequently required because existing software that the researcher could potentially exploit is not documented in sufficient detail internally, may not be suitable because of software architecture design, or both. After enduring this time-consuming software development effort, the researcher may find that a thorough evaluation of the new method is still impossible due to limitations of the supporting software. This scenario is true for many "research-oriented" finite element codes which have a limited element library or have a problem-size limit because of the use of a memory-resident equation solver. In addition, new computer architectures with vector and multiprocessor capabilities are being manufactured for increased computational power. Analysis and computational algorithms that can exploit these new computer architectures need to be developed. For credibility, these new algorithms should be developed and evaluated in a standard, general-purpose finite element structural analysis software system rather than in an isolated research software system.

At the NASA Langley Research Center, an intense effort is being directed towards developing advanced structural analysis methods and identifying the requirements of the next generation structural analysis software system which will exploit multiple vector processor computers

† Aerospace Engineer, Structural Mechanics Branch, Structural Mechanics Division.
‡ Structural Engineers, Planning Research Corporation.
* Research Scientist, Analytical Services and Materials, Inc.

(Blankenship and Hayduk[1]). This activity, called Computational Structural Mechanics or CSM (Knight and Stroud[2]), has developed the CSM Testbed software system (Lotts et al.[3], and Stewart[4,5,6]) to aid in the definition of these requirements and to serve as a "proving ground" for new methods on large-scale structural application problems. At the NASA Ames Research Center, the Numerical Aerodynamic Simulation (or NAS) Program (Bailey[7]) has developed the NAS System to provide a national computational capability and to serve as a pathfinder in advanced, large-scale computer systems.

This paper describes the evolving role of the structural analyst amidst a rapidly changing computing environment. The distributed nature of the computing hardware environment is described and its use demonstrated. The integrated nature of the computing software environment using the CSM Testbed is described and its use demonstrated. The flexibility of the CSM Testbed coupled with the computational facilities available through the NAS System make it possible for numerical analysts, methods developers, structural analysts, and computer scientists to integrate their research in a common, shared computing environment. The powerful problem-solving capability of this computing environment is demonstrated by solving several structural application problems involving linear and nonlinear stress analysis, buckling analysis, and transient dynamics analysis.

Computing Environments

During the 1970's and early 1980's, large-scale scientific computing was typically performed using stand-alone computing systems that were characterized by a single computer. Individual jobs were processed serially in a "batch" mode. During the mid-1980's, the concept of distributed computing was introduced. Distributed computing is characterized as several independent computers transparently linked together by network systems. Currently, the trend in computing environments is moving towards an integrated computing environment of shared resources (disks, networks, CPU's). The evolving computational hardware environment is described in the following sections.

Stand-Alone Systems

The stand-alone computing environment included three types of computer hardware systems. The first type was the minicomputer such as a DEC VAX 11/785 computer system using the VMS operating system. The second type was the mainframe computer such as a

CDC CYBER 175 computer system using the NOS operating system. The third type was the supercomputer such as a CDC VPS/32 supercomputer using the VSOS operating system. Each of these systems were used for a variety of structural analysis application studies and for the development of new structural analysis methods. However, the inter-machine communication link was limited to exchanging magnetic tapes between computer types. A knowledge of several computer operating systems was required for the user to work effectively on several computer types. The limitations of these stand-alone systems with vendor-specific operating systems and communication protocols (*e.g.*, VAX/VMS, CDC/NOS) were soon realized, and distributed computing environments were developed.

Distributed Systems

Distributed environments involve several types of computer systems; namely, personal computers (PC's), graphics workstations, minicomputers, mainframes, and supercomputers linked together through networks. An example of a distributed system is the coupling of a DEC VAX/VMS minicomputer through DECnet and a CRAY-station software system with a CRAY X-MP/48 supercomputer running the COS operating system. The changes occurring in the field of computer networking represent probably the most dramatic changes affecting structural analysts. Networking removed the constraint of physical distance. Working remotely from a supercomputer presented a new set of problems that, once solved, resulted in a unique new capability for the structural analysts.

The network at the NASA Langley Research Center uses Ethernet within buildings and a fiber optic Pronet 10 token-passing ring network called LARCnet between buildings. Initially the gateways between buildings would route only a Xerox XNS-based protocol developed at Langley. Connecting to one of the Ethernets was a Vitalink Bridge that routed both TCP/IP and DECnet to the NAS facility at Ames over a 256 kilobits per second satellite link. The evolution of this network has followed the networks developing in industry. Workstations in many buildings are supported with routing gateways through the LARCnet fiber optic system, through a Pronet P4200 gateway connected to the Vitalink directly. The communication link with Ames has been upgraded to a one megabit per second transfer rate (*i.e.*, T1 link), and it was discovered that a land line is preferable to a satellite link for interactive use. The result is that the miles between Ames and Langley are no longer a problem; Langley researchers can use the NAS system at Ames as easily as if it were located at Langley. The NAS CRAY-2

supercomputer appears to the structural analyst as if it were embedded in the local workstation.

The NAS CRAY-2 supercomputer uses the UNICOS† operating system, has four processors (each with a clock-cycle time of 4.1 nanoseconds), and has a total memory size of 256 million 64-bit words. The CRAY-2 supercomputer is capable of over one hundred times the computational capability of a VAX 11/785 minicomputer. In addition, the CRAY-2 computer system is a native 64-bit wordsize machine, and roundoff problems that can be a problem on 32-bit machines are usually eliminated. Even with 256 million words of main memory, finite element system matrices for large-scale structural analysis may not fit in memory. Auxiliary data storage requirements for these analyses is another concern. Single temporary files may require in excess of 500 megabytes of storage. Hence, coordination or scheduling of these runs by the analyst is necessary to avoid exceeding the available auxiliary storage.

Distributed computer environments are made up of stand-alone computers of different sizes, architectures, and vendors, with a common network protocol offering the user easy file transfer and remote login functions. Structural analysts require the diverse computer capabilities offered by a distributed environment (workstation–mainframe–supercomputer), but cannot afford the "overhead" of learning the operating system commands for each system they use. Software developers have a similar problem, but at a lower level. They cannot afford the "overhead" of learning a new set of system calls for each computer on which they wish to implement their application software. To alleviate this "overhead", integrated computing environments are evolving which exhibit a common operating system.

† The UNICOS operating system is derived from the AT&T UNIX System V operating system. UNICOS is also based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California.

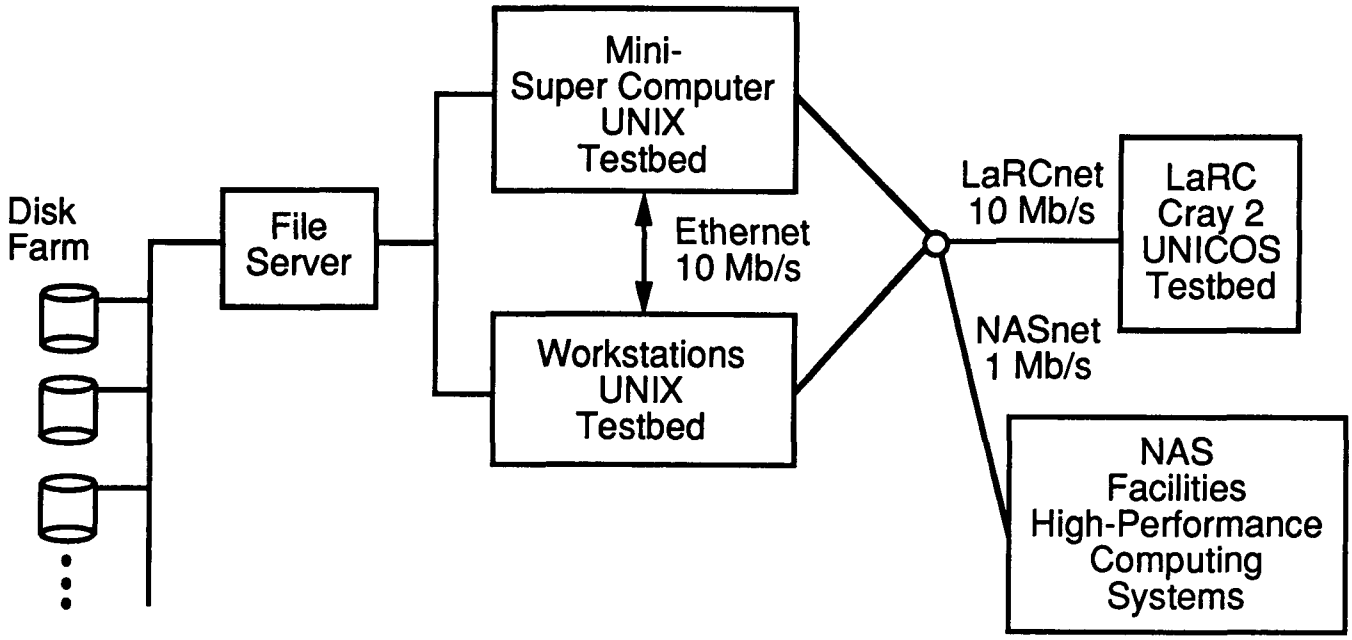


Fig. 1 Integrated computing environment.

Integrated Systems

Integrated computing environments are a natural step in the evolution of computer hardware systems. This step represents an environment dominated by an open-system architecture (*i.e.*, UNIX and TCP/IP). Integrated systems may involve several types of computer systems; namely, graphics workstations, mini-supercomputers, and supercomputers. In addition, integrated systems will also incorporate a tightly-coupled auxiliary storage subsystem (*i.e.*, disk farm) and high-speed wide-area and local-area networks. An integrated computing environment is shown schematically in figure 1. This integrated computing environment is a distributed, networked system with a common operating system (UNIX), a shared local file system (through NFS), and a high-speed wide-area network (NASnet).

Overview of CSM Testbed

The field of computerized structural analysis is dominated by two types of computer programs. One type is the huge, 2000 subroutine general purpose program (McLean[8]), that is the result of over a hundred man years of effort spanning more than a decade. The other type is the relatively small, special-purpose code resulting from a research environment that represents a one- to two-year effort for a specific research application. This dichotomy has resulted in long delays in making research technology available for critical structural analysis problems that NASA faces. To accelerate the introduction of successful research technology into large-scale

applications programs, a modular, public-domain, machine-independent, architecturally-simple, software development environment has been constructed. This system is denoted the CSM Testbed software system and its concept is depicted by a pyramid (see figure 2). The base of the pyramid is the computer and its operating system. The computer operating system is provided by the computer vendor and may be different for each vendor. Currently, the CSM Testbed is primarily targetted for UNIX-based systems in order to minimize these differences. The Testbed architecture insulates both the engineer and the methods developer from those differences by providing a consistent interface across various computer systems. The Testbed command language CLAMP procedures and application processors may be accessed as part of a methods research activity or as part of an application study. The methods development environment of the CSM Testbed is further described by Gillian and Lotts[9]. One goal of the CSM Testbed is to provide a common structural analysis environment for three types of users — engineers solving complex structures problems, researchers developing advanced structural analysis methods, and developers designing the software architecture to exploit multiprocessor computers.

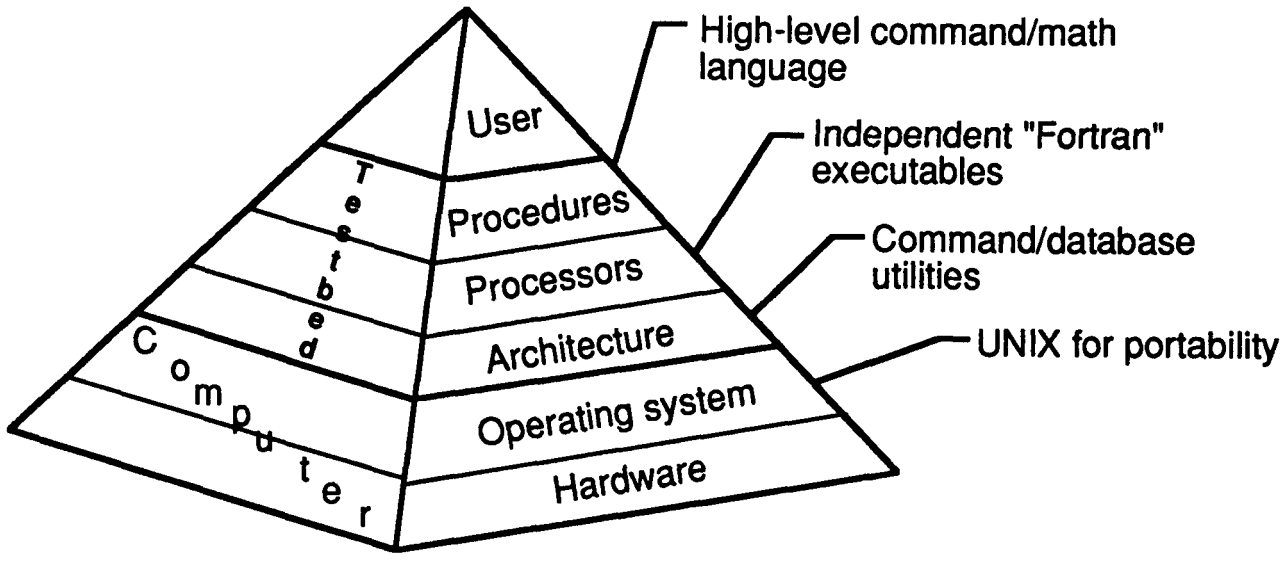


Fig. 2 Concept of the CSM Testbed software system.

The CSM Testbed software system is a highly modular and flexible structural analysis system for studying computational methods and for exploring new multiprocessor and vector computers. The CSM Testbed is used by a group of researchers from universities, industry, and government agencies. Unrestricted access to all parts of the code including the data manager and the command language is permitted. Research on these elements of software design is needed because deficiencies in the data management strategy can have a devastating impact on the performance of a large structural analysis code, totally masking the relative merits of competing computational techniques. Furthermore, software designs that exploit multiprocessor computers must be developed; in particular, techniques for handling parallel input/output (I/O) are required.

The initial CSM Testbed, called NICE/SPAR, began with the integration of the NICE system (Felippa[10] and Felippa and Stanley[11]) and Level 13 of SPAR (Whetstone[12]). Since then, new capabilities and improvements have been implemented in the CSM Testbed. Each step of the evolution of the CSM Testbed provides improved structural analysis capabilities to structural analysts. Implementation of new capabilities is done using the framework of the CSM Testbed as depicted in figure 3. A brief description of selected CSM Testbed processors is given in Table 1.

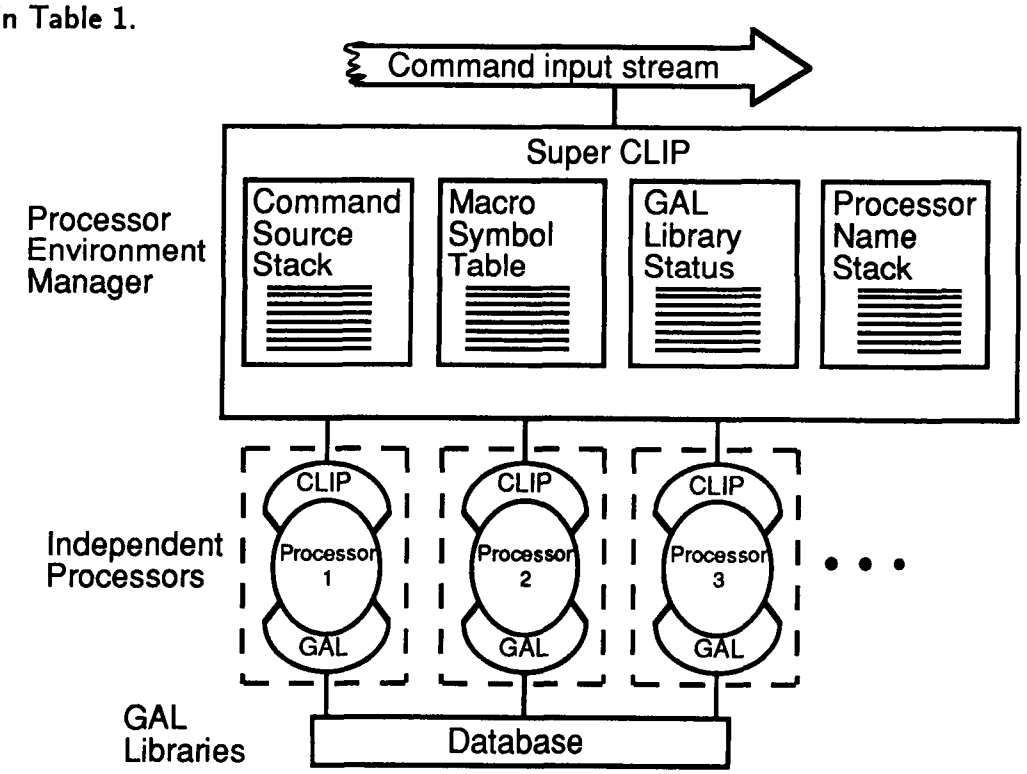


Fig. 3 Implementation of the CSM Testbed Software System.

Table 1. Selected CSM Testbed processors.

Processor	Description
ELD	Element definition (connectivity, material properties, etc)
LAU	Laminate analysis utility for 2-D and 3-D elements
E	Element-state initiation (build element information packets)
EKS	Compute the SPAR element intrinsic stiffness matrices
TOPO	Analyze the finite element mesh topology and build tables to drive assembly and factorization of system matrices
RSEQ	Reorder nodes for minimum fill or minimum bandwidth
AUS	Arithmetic utilities
K	Assembles unconstrained system stiffness matrix
M	Assembles unconstrained system mass matrix
INV	Applies constraints and factors assembled system matrix
SSOL	Performs forward reduction and back substitution
BAND	Factor and solve using profile or banded solvers
ITER	Factor and solve using iterative solvers
SPK	Factor and solve using sparse solver
KG	Form and assemble unconstrained system geometric stiffness matrix
EIG	Solves linear algebraic eigenproblems
ES	Generic element processor shell
VEC	Performs variety of vector algebra operations
IMP	Form initial geometric imperfections
NVAL	Perform global smoothing of element results
FPF	Perform first-ply failure analysis
T2PT	Testbed-to-PATRAN translator
PT2T	PATRAN-to-Testbed translator

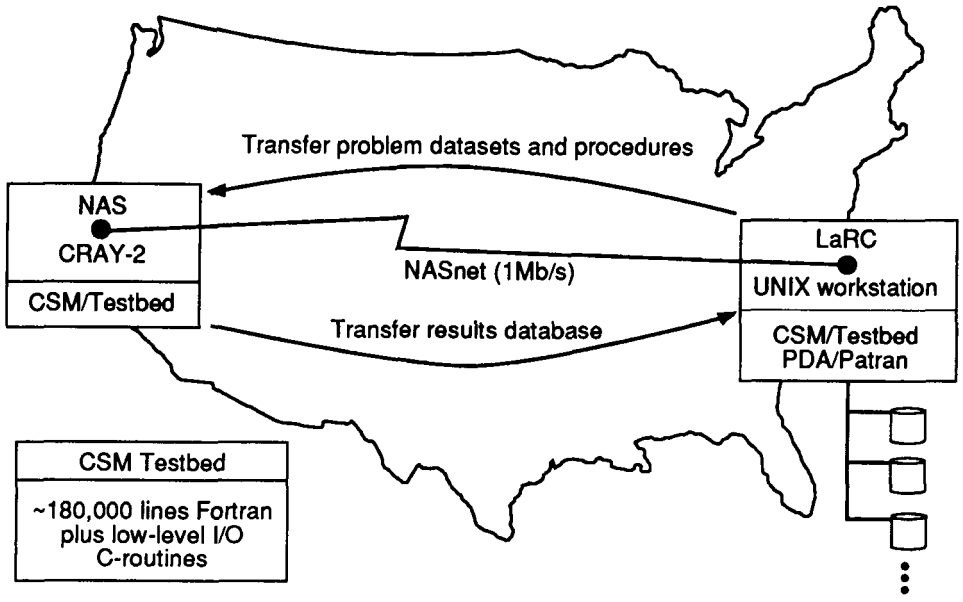


Fig. 4 Distributed computing environment of CSM.

CSM Computing Environment

The computing environment of the CSM activity is currently a distributed environment as shown in figure 4. Typically, a structural analyst will develop a finite element model of the structure either by using a preprocessing software system such as PATRAN or by using the CSM Testbed command language for "parameterizing" the model. Runstreams are the vehicle used to perform structural analyses with the CSM Testbed. The term "runstream" most commonly refers to the file (or files) of input data and commands used to perform a specific analysis, although it may also refer to input during an interactive session. Runstreams for the CSM Testbed are usually developed and verified on a workstation, and then transferred to the NAS CRAY-2 computer system for complete processing. Following a successful execution, the computational database may then be "unloaded" (*i.e.*, converted from the binary format of the NAS CRAY-2 computer system to ASCII format), transferred intact to the Langley Research Center using the NASnet wide-area network, and then "loaded" (*i.e.*, converted from ASCII format to the binary format of the desired workstation) back into a computational database which has the identical Testbed library format as on the NAS CRAY-2 computer system. Finally, postprocessing is done to help the structural analyst visualize the computed structural response. The sequence of steps just described depicts the computing environment to which the structural analyst must adapt in order to exploit the full potential of available computing systems.

To exploit this new computing environment, expertise is needed in the areas of computational strategies, numerical techniques, computer science, and communication networks, together with a firm understanding of the principles of structural mechanics. New computing hardware environments, like the NAS System, offer the computational power, memory, and disk space necessary for routine analysis of large structural models. New computing software environments, like the CSM Testbed, offer an integrated system with data management, a general command language, and many different application processors – features that enable the structural analyst to develop new analysis methods and to tailor the analysis for specific application needs.

CSM Testbed Architecture Features

The CSM Testbed is a Fortran program organized as a single executable file, called a macro-processor, which calls structural applications modules that have been incorporated as subroutines. The macro-processor and applications modules interface with the operating system for their command input and data management functions through a set of common "architectural utilities". Processors access the Testbed utilities by calling entry points implemented as Fortran-77 functions and subroutines which are available in the Testbed object libraries. Applications processors do not communicate directly with each other, but instead communicate by exchanging named data objects in a database managed by a data manager called GAL (Global Access Library). The user controls execution of applications processors using an interactive, or batch, command runstream written in a command language, called CLAMP (Command Language for Applied Mechanics Processors), which is processed by CLIP (Command Language Interpreter Program).

Command Language

The Testbed command language CLAMP is a generic language originally designed to support the NICE system and to offer program developers the means for building problem-oriented languages (Felippa[13-15]). It may be viewed as a stream of free-field command records read from an appropriate command source (the user's terminal, actual files, or processor messages). The commands are interpreted by a "filter" utility called CLIP, whose function is to produce object records for use by its user program. The standard operating mode of CLIP is the processor-command mode. Commands are directly supplied by the user, retrieved from ordinary card-image files, or extracted from the global database, and submitted to the running

processor. Special commands, called directives, are processed directly by CLIP; the processor is "out of the loop". Transition from processor-command to directive mode is automatic. Once the directive is processed, CLIP returns to the processor-command mode. Directives are used to dynamically change run-environment parameters, to process advanced language constructs such as macrosymbols and command procedures, to implement branching and looping, and to request services of the data manager. CLIP can be used in this way to provide data to a processor as well as to control the logic flow of the program through a single input stream. All command language directives are available to any processor that uses the CLIP-Processor interface entry points.

Directives are understood and processed by CLIP and provide the user with a means of defining command procedures. Command procedures, defined using the `*PROCEDURE` directive, bear some resemblance to Fortran subroutines. They may contain branching and looping constructs (implemented using the `*DO`, `*IF`, and `*WHILE` directives) as well as other directives and processor and macroprocessor commands. Command procedures may be given arguments which, unlike Fortran subroutine arguments, may be assigned default values. When a command procedure is called (using the `*CALL` directive), execution control shifts to the command procedure until the last directive (a `*END` directive) in the procedure is encountered. Once the `*END` directive is encountered, control returns to the input line in the calling procedure or runstream immediately following the call.

Command procedures, while extremely useful, are not a requirement for performing many types of simple analyses. A command procedure is only required if looping or branching constructs (*i.e.*, the `*DO`, `*IF`, and `*WHILE` directives) are used. Researchers may use procedures to study analysis methods (*e.g.*, problem-adaptive solution strategies) prior to implementing a specific strategy as a Fortran processor. However, computationally-intensive tasks may be inefficient as procedures and should be implemented as processors for use in a production analysis mode.

Data Manager

The data manager within the CSM Testbed was derived from the Global Access Library (GAL) concept developed at the Lockheed Palo Alto Research Laboratory (Wright et al.[16]). Methods for data management in structural analysis programs can be divided into three levels of complexity: file systems, file partition systems, and database systems (Hurst and Pratt[17]).

Since database files are subdivided or partitioned into datasets, the Testbed data manager is classified as a file partition manager. To a processor, a GAL data library is analogous to a file. It must be opened, written, read, closed, and deleted explicitly. The global access library resides on a direct-access disk file and contains a directory structure called a table of contents (TOC) through which specific datasets may be addressed. Low-level I/O routines access the GAL library file in a word-addressable scheme as described by Felippa[18]. The data management system is accessible to the user through the command language directives and to the running processors through the GAL-Processor interface.

The global database is made up of sets of data libraries residing on direct-access disk files. Data libraries are collections of named datasets, which are collections of dataset records. The data library format supported by the Testbed is called GAL/82, which can contain nominal datasets made up of named records. Some of the advantages of using this form of data library are: *i*) the order in which records are defined is irrelevant, *ii*) the data contained in the records may be accessed from the command level, and *iii*) the record data type is maintained by the manager; this simplifies context-directed display operations and automatic type conversion.

To provide the efficiency required to process the volume of data required for a complex structural analysis, all usual overhead associated with Fortran has been eliminated. The actual I/O interface between the GAL data manager and the UNIX operating system is accomplished through a set of block I/O routines written in the C programming language. For non-UNIX computer systems, this interface is accomplished through a set of assembly-language routines which are unique to each computer system.

User Interface

The user may develop runstreams using the high-level command language CLAMP for a specific engineering problem (*e.g.*, Felippa[13,14]). These runstreams may contain CLAMP directives and CLAMP procedures which are processed by the command language interpreter CLIP. Application processors are called using the `[XQT` command, or the global access library GAL (*e.g.*, Wright et al.[16]) may be interrogated. Engineers typically interact with the Testbed using simple runstreams or through CLAMP procedures. Researchers interact using CLAMP procedures (*e.g.*, to study nonlinear solution strategies) or through Fortran processors (*e.g.*, to implement new element formulations). Developers interact with the entire Testbed architecture, including the design of the command language, the data handling techniques for

large-scale analyses, and the strategy for I/O on parallel computers.

CSM Testbed Structural Analysis Features

The CSM Testbed presently provides structural analysis capabilities that permit an analyst to perform large-scale nonlinear stress analyses of shell-type structures. Three-dimensional stress analyses are presently limited to linear elastic orthotropic materials. Eigenvalue problems associated with either linear bifurcation buckling or linear vibration analyses may also be solved. Transient dynamic analyses are limited to linear elastic problems using either direct time integration or mode superposition to obtain the transient response. Some of the newly-developed engineering features of the CSM Testbed are the equation solvers, the element library, the material modeling, and the solution procedures. Interface utilities to and from the PATRAN graphics systems have been developed to support the modeling and analysis of large-scale structures. Access to such a preprocessing and postprocessing software system enhances the structural analyst's ability to understand the structural behavior through visualization of the computed results.

Equation Solvers

The system of equations that arise in static structural analysis applications has the general form $\boldsymbol{Ku} = \boldsymbol{f}$ where $\boldsymbol{K}$ is the symmetric, positive definite stiffness matrix, $\boldsymbol{f}$ is the load vector, and $\boldsymbol{u}$ is the vector of generalized displacements. Such linear systems can be as large as several hundred thousand degrees-of-freedom (*dof*) and often require significant computing resources, both memory and execution time. The structure of the stiffness matrices in these applications is often sparse, although in many applications an ordering of the nodes which minimizes the bandwidth makes banded or profile (skyline) type storage of these matrices practical. The choice of the particular method used to solve $\boldsymbol{Ku} = \boldsymbol{f}$ will depend on the non-zero structure of $\boldsymbol{K}$ and, in the case of the iterative methods, the condition number of $\boldsymbol{K}$. In addition, the architecture of the computer, particularly for modern vector and parallel computers, influences both the choice and implementation of methods used to solve these linear systems of equations. Ortega[19] presents a thorough description of these various methods and their implementations as applied to vector and parallel computers.

The data structure of the global stiffness matrix is a key factor in the design and implementation of equation solvers for the CRAY-2 architecture and the Testbed software (*e.g.*, Poole and Overman [20]). The generation of stiffness matrices is accomplished by several different

processors producing element stiffness matrices, defining boundary conditions, applied loads and ordering of nodes, and the assembling the global stiffness matrix. The stiffness matrix is stored in a nodal-block sparse form for which each block is dimensioned as $dof \times dof$ (usually 3×3 or 6×6) for each node. The original sparse out-of-core Choleski solver used by the Testbed (processors INV and SSOL) factors and solves the stiffness matrices using this data structure. A major source of inefficiency for this solver on a CRAY-2 computer system is that the operations carried out in factoring the stiffness matrix and solving the resulting triangular systems are carried out using these small $dof \times dof$ blocks. The vector length of these operations is therefore six or less, and the code is faster when run without vector optimization.

The new vectorized equation solvers (processors BAND and ITER) require $\boldsymbol{K}$ to be stored in one of several different sparse and banded storage schemes. Processor ITER contains three conjugate gradient iterative methods. These methods vary in their types of preconditioning, which include diagonal scaling, incomplete Choleski factorization with a sparse storage scheme, and incomplete Choleski factorization with a diagonal storage scheme. Processor BAND contains three basic algorithms that are all based on Choleski factorization of banded matrices. The first algorithm uses the standard LINPACK[21] routines for banded solvers; namely SPBFA and SPBSL. The second algorithm, *kji* Choleski, uses column storage of the lower triangular part of the symmetric matrix (to take advantage of vectors with a constant stride of one) and loop unrolling to level four. Loop unrolling reduces the number of memory references by holding vectors longer in the registers and increases the amount of vector computations within a loop. As a result, many of the multiplication and subtraction operations and memory references will overlap, leading to greater performance. In addition, the local memory of the CRAY-2 computer system is used to store up to four columns of the factored matrix to further decrease execution time. The third algorithm uses variable-bandwidth or profile storage of the matrix instead of banded storage and this type of storage results in a significant reduction in memory requirements and in the number of operations.

The strategy used for the vectorized equation solvers involves four steps. First, the coefficients of the unconstrained stiffness matrix are read from the global database into a temporary array. Second, the nodal constraint information and node ordering sequence information is retrieved from the global database. Third, the appropriate pointer arrays for the new storage scheme are formed. Finally, the coefficients of $\boldsymbol{K}$ are placed in a singly-dimensioned array and modifications are made to the right hand side $\boldsymbol{f}$ corresponding to any applied displacements.

For the direct Choleski methods, an additional storage scheme is included to reformat Testbed stiffness matrices into the standard LINPACK[21]-banded storage format. The reformatting procedure is essentially sequential, but the time to reformat the matrices is small compared with the time to solve the equations for large problems.

The capability to reorder the nodes automatically is an important part of the equation solving process in general-purpose finite element codes. The structure of the assembled stiffness matrices is determined by the node connectivities and node numbering scheme used in the finite element model. Although the node connectivity is fixed by the problem definition and discretization, many node orderings are possible. The Testbed software contains processor RSEQ, which uses four different algorithms to reorder nodes automatically. These algorithms are: nested dissection, minimum degree, reverse Cuthill-McKee, and Gibbs-Poole-Stockmeyer. The first two methods are used by sparse solvers and minimize fill in the factorization process. The last two are profile and bandwidth minimizing routines, respectively. The direct banded solvers implemented in processor BAND are most efficient with node orderings which minimize bandwidth, while the sparse out-of-core Choleski equation solver in processor INV is most efficient with orderings which minimize fill. For the various preconditioned conjugate gradient methods in processor ITER, the preconditioner used determines which ordering is best. Although the precise relationship between node ordering and the convergence rate of the Incomplete Choleski Conjugate Gradient (ICCG) is not known, preliminary results indicate that the ordering of nodes can have a great effect on the convergence rate. In the test problems used with the ICCG method, the convergence rate of ICCG is better for the sparse, minimum-fill orderings than for the bandwidth-minimizing orderings. However, in some cases, the ordering used to define the problem gives the best convergence rate. For the basic conjugate gradient method, the matrix structure has no effect on the convergence rate but the matrix structure is important for the storage requirements if diagonal storage is used. Orderings which minimize bandwidth also concentrate the coefficients near the main diagonal thereby minimizing the number of diagonals required for matrix storage. As a result, the vector lengths of the diagonals are longer, and the number of extra zeros added between non-zero coefficients is fewer; thus, the memory requirements are reduced, and the computation speed increased.

Generic Element Processor Template

The generic element processor template shown in figure 5 provides the element developer with a standard outer software "shell" that handles all user-command input and all I/O to and

from the global database. In addition, a standard set of "shell-to-kernel" interface routines (*e.g.*, ES_K, ES_M, ES_F) are provided as cover routines for the element developer's "kernel" routines. The function of the interface routines is to perform the transformation between the standard argument lists of the outer software "shell" and those of the element developer's personal code. The element developer's "kernel" routines are integrated with these interface routines using the convention that the interface subroutine names and argument lists are standardized. The independent structural element processors (*i.e.*, processor ES*i*, where *i*=1,2,...) are installed and readily accessible to all CSM researchers for small benchmark problems as well as large-scale application problems.

The generic element processor features a standard high-level procedure named ES that processes user commands such as

```
*call ES ( function='FORM STIFFNESS/MATL'; es_proc=ESi ).
```

All of the ES*i* processors are driven by a common set of commands through calls to the ES procedure and create the same data structure, regardless of how the element developer programmed the "kernel" routines (*e.g.*, the element stiffness calculations, element stress recovery). This approach provides an extendible and easy-to-use vehicle for integrated finite element research, development, and application within the CSM Testbed.

A key feature of the generic element processor "shell" is the easy access to the utilities associated with an element-independent corotational formulation (Rankin and Brogan[22]). Through these utilities, element developers may readily attempt geometric nonlinear problems which exhibit large rotations. Only the basic element characteristics associated with linear strain-displacement relations are required from the element developer in the "kernel" routines. Extensions to include the nonlinear strain-displacement relations require the element developer to provide additional "kernel" routines (*e.g.*, internal force calculations).

Element Library

Presently only two-dimensional shell elements and three-dimensional solid elements have been installed in the CSM Testbed using the generic element processor template (see Table 2). Processor ES1 contains a family of 4- and 9-node continuum-based resultant (CBR) quadrilateral shell elements (Stanley[23]). This family of elements includes the assumed-natural coordinate strain (ANS) quadrilateral shell elements (Park and Stanley[24]) and the Lagrangian (LAG) quadrilateral shell elements with selectively-reduced integration (SRI). Processor ES2

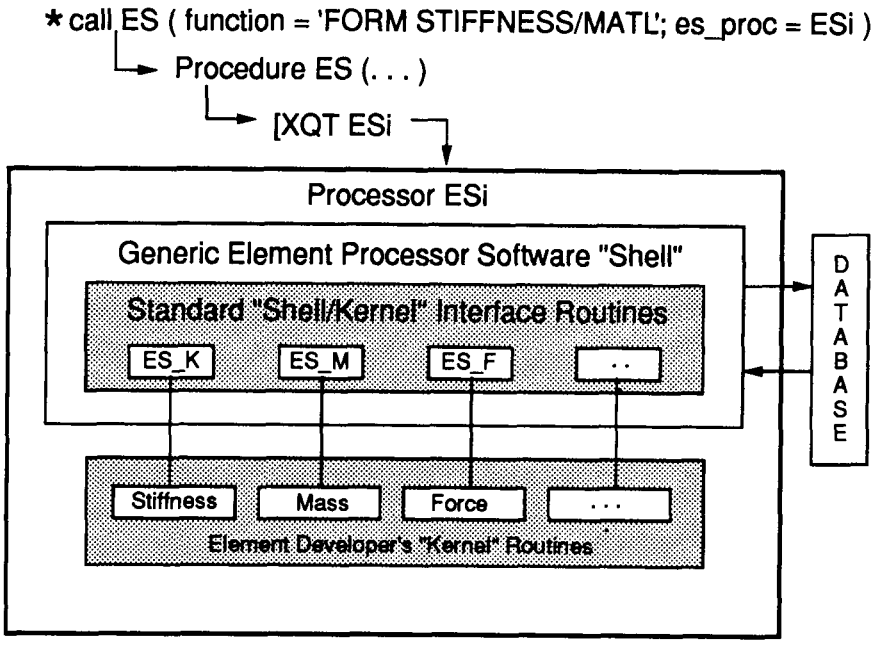


Fig. 5 Generic element processor template.

contains a new hybrid curved 4-node quadrilateral shell element (Kang and Pian[25]). Processor ES3 contains a family of three-dimensional hybrid solid elements including 8- and 20-node bricks (hexahedrons), 6- and 15-node wedges (pentahedrons), and 4- and 10-node pyramids (tetrahedrons), respectively (Aminpour[26]). Processor ES4 contains a family of hybrid plate/shell elements including 4-node quadrilateral and 3-node triangular elements (Aminpour[26]). Processor ES5 contains a displacement-based, 4-node quadrilateral plate/shell element from the STAGSC-1 computer code (Rankin, Stehlin, and Brogan[27]), denoted the 410-element. Processor ES10 contains five isoparametric displacement-based solid elements including 8-node, 16-node, 20-node, 24-node, and 32-node brick elements (Griffin[28]). Additional ES*i* processors are under development. In addition, elements in the original element library of Level 13 of SPAR[12] are currently still available for linear analyses.

Material Modeling

The material modeling features of the CSM Testbed are directed towards the analysis requirements of laminated composite structures. Constitutive relations for classical and shear flexible two-dimensional plate and shell models as well as for three-dimensional solids are evaluated and available to the element developer or structural analyst. Processor LAU is a laminate analysis utility for calculating the constitutive relations for 2-D and 3-D isotropic, orthotropic,

Table 2. Summary of current ES*i* processors

Processor	Description
ES1	4-, 9-node, curved, C^0, CBR (ANS and LAG), shell elements
ES2	4-node, curved, C^1, hybrid stress, shell elements
ES3	4-, 6-, 8-node,C^0, hybrid stress/strain, solid elements
	10-, 15-, 20-node, C^0, hybrid stress/strain, solid elements
ES4	3-, 4-node, flat, C^0, hybrid stress/strain, plate/shell elements
ES5	4-node, flat, C^1, displacement, STAGS 410 plate/shell element
ES10	8-, 16-, 20-, 24-, 32-node displacement, solid brick elements

and laminated structures. The formulation is based on the usual lamination theory (*e.g.*, Jones[29] and Whitney[30]) whereby the laminate constitutive relations are derived from the constitutive relations for each layer in the laminate. Having the midplane strains and curvatures, the inplane strains and corresponding stresses in each layer of the laminate may be calculated and used to evaluate selected stress- and strain-based failure criteria. The failure criteria implemented in processor FPF include maximum stress criteria, maximum strain criteria, and several quadratic polynomial failure criteria including Tsai[31]-Hill[32], Azzi-Tsai[33], Hoffman[34], and Tsai-Wu[35,36]. In addition, the failure criterion developed by Shuart[37] for zero-dominated laminates has been implemented.

Solution Procedures

Various types of analysis may be performed with the CSM Testbed through the use of either ordinary runstreams which execute various processors sequentially or CLAMP procedures which execute directives and processors and perhaps call other procedures. Linear stress analyses and eigenvalue analyses are both performed using simple analysis runstreams (*e.g.*, see Table 3). Solution procedures that require looping and branching are more complex procedures than linear analysis procedures. Two sets of solution procedures that require looping have been written and may be used to solve various application problems.

The first solution procedure is named NEWMARK and is based on the computational strategy outlined by Bathe[38]. Its function is to perform a linear transient dynamic analysis using the well-known Newmark-β method for direct time integration of the equations of motion. A listing of this procedure is given in reference 6. Parameters such as system stiffness and mass matrix names, the time step, and the total number of time steps in the analysis are formal

Table 3. Sample linear stress analysis runstream for the CSM Testbed.

```
$ testbed
*set echo=off
*add gen_util:utilities.prc
*open/new 1, flat_plate.101
*def/a es_name == EX97                    . Element name
*def/a es_proc == ES1                     . Element processor
*call ES ( function = 'DEFINE ELEMENTS'; es_proc = <es_proc> ;--
          es_name = <es_name>)
[xqt TAB
     START 25 1 2 6                       . 25 nodes; dof 1,2,6 constrained
     JOINT LOCATIONS                      . Enter joint locations
       1      0.0   0.0   0.0  2.5   0.0   0.0  5 1 5
        5      0.0   2.5   0.0  2.5   2.5   0.0
     CONSTRAINT DEFINITION 1              . Constraints:
       zero 3,4,5: 1,21,5
       nonzero 3 : 5,25,5                 . Apply displacement at x=lx edge
.
[XQT AUS                                  . Define material and section properties
     TABLE(NI=16,NJ=1): OMB DATA 1 1
     I=1,2,3,4,5,6,7,8,9
     J=1: 19.e+6 0.38   1.89e+6 0.93E+6 0.93E+6 0.93e+6 1.e-4 1.e-4 .01
.
     TABLE(NI=3,NJ=2,itype=0):  LAM OMB  1  1
     J=1:  1 .001  45.0
     J=2:  1 .001 -45.0
.
[xqt LAU                                  . Generate constitutive matrix
[xqt ELD                                  . Define elements
       <es_expe_cmd>
       NSECT = 1
       1  3 13 11  2  8  12  6  7    1 2 2
.
[xqt E                                    . Initialize element datasets
     stop
     *open 1
*call ES (function='INITIALIZE')          . Initialize element matrices
*call ES (function='FORM STIFFNESS/MATL') . Form intrinsic stiffness matrices
[xqt RSEQ                                 . Resequence
     reset maxcon=27
[xqt TOPO                                 . Create maps
[xqt K                                    . Assemble global stiffness matrix
[xqt INV
[xqt AUS                                  . Form applied displacement
     sysvec : appl moti : i=3:   j=5,25,5:    -0.01
[xqt SSOL                                 . Solve for static displacements
     stop
     *open 1
*call STRESS (direction=1; location= NODES)
[xqt VPRT                                 . Print static displacements
     print STAT DISP
[xqt exit
```

arguments to procedure NEWMARK. The CLAMP macro expression capability is used for calculating integration constants and controlling the algorithm. The initial acceleration at time $t = 0$ is calculated from the given initial displacement and velocity vectors. This initialization is done by using processor AUS to set up the equations of motion at $t = 0$, and processors INV and SSOL to solve for the initial accelerations. At each subsequent time step, processor AUS is used to set up the recursion relations, and processor SSOL is used to solve for the displacement vector at the next time step. Velocity and acceleration vectors may then be calculated and selectively printed.

The second set of procedures is named NL_STATIC_1. These procedures are used to perform a geometric nonlinear static analysis using a modified Newton-Raphson algorithm with corotational updates and an arc-length control strategy for either applied force or applied displacement problems (*e.g.*, Stanley and Felippa[39]). A listing of this set of procedures is also given in reference 6. Procedure NL_STATIC_1 provides a global load-stepping algorithm for advancing the nonlinear solution during a static analysis. Arc-length control algorithms treat the load factor as an additional unknown and augment the equilibrium equations with a constraint equation that ties the load factor to the generalized displacement unknowns. Implementations of the arc-length control strategy vary depending on how the augmented equation system is solved and the form of the arc-length constraint equation (*e.g.*, see Riks[40,41] and Crisfield[42]). The implementation in procedure NL_STATIC_1 involves a linearized version of the Crisfield quadratic arc-length constraint equation. At the beginning of each "arc-length" step, a new tangent stiffness matrix is formed and factored. This tangent stiffness matrix is used for all iterations at this step. Hence, this implementation may be viewed as a modified Newton-Raphson algorithm with simultaneous iteration on the generalized displacements and the load factor.

Application Studies Using CSM Testbed

Research in methods development for the CSM Testbed is driven in part by analysis deficiencies identified while solving various application problems. The Langley CSM activity employs the concept of focus problems to provide a common set of structural analysis problems for all CSM participants. Focus problems may be entire aerospace vehicles or various subcomponents that pose difficult structural mechanics problems. The problems selected as focus problems challenge our ability to predict their structural response and stretch our computing limits. These focus problems help guide methods research and development for generic classes of problems. New focus problems are selected as new technology evolves and computational structural mechanics methodology develops. To use large, complex structures as focus problems requires an understanding of the structure, its loading, and life cycle as well as an understanding of the underlying computational structural mechanics issues.

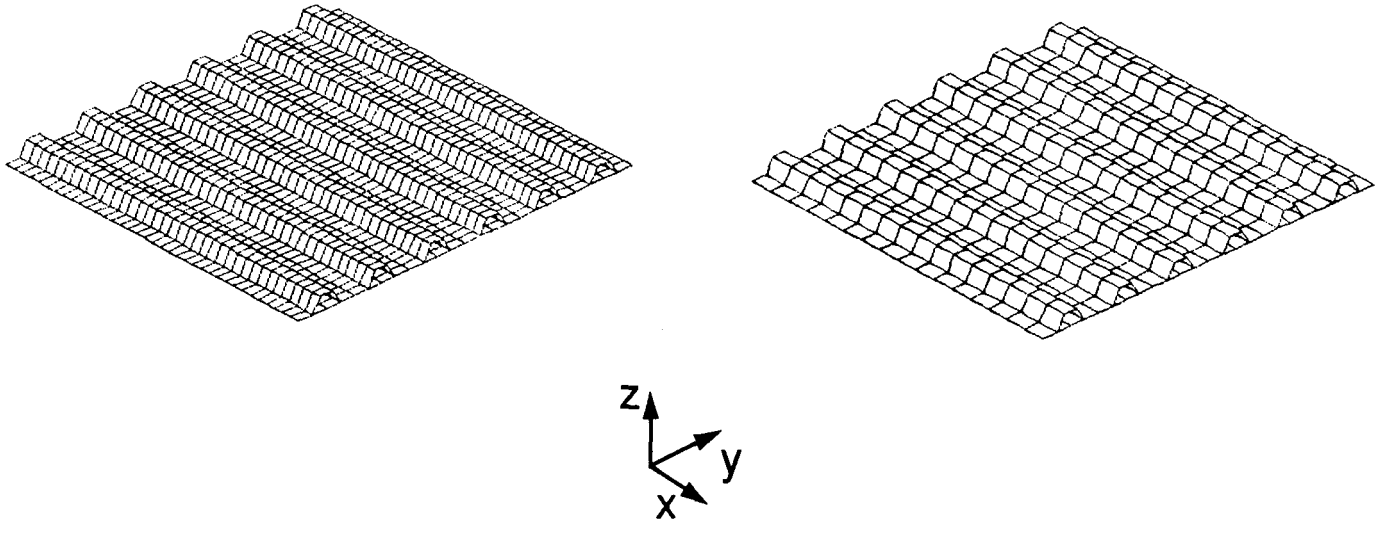
The application studies presented in this section represent a wide range of structural analysis problems. The problems selected for presentation here are:

- Composite hat-stiffened panel
- Composite blade-stiffened panel with discontinuous stiffener
- Three-dimensional composite analysis
- Circular cylindrical shell with cutouts
- Pear-shaped cylinder
- Impulsively loaded truncated conical shell
- SRM tang-clevis joint
- SRB global shell model

These application studies demonstrate the structural analysis capabilities of the CSM Testbed. The analyses presented herein utilize solution procedures implemented through the CLAMP language and various finite elements implemented through the generic element processor template. The execution times for selected CSM Testbed processors are compared for the various analysis problems considered. Postprocessing of the results, both deflections and stress resultants, are performed by first using the Testbed-to-PATRAN translator and then PATRAN for visualization of the computed results.

Composite Hat-Stiffened Panel

Aerospace structures generally involve stiffened panel construction. Understanding the structural response of composite stiffened panels to combined longitudinal compression and shear loading is necessary in order to tailor the design of the various subcomponents of aerospace vehicles. Various stiffened panel configurations were analyzed by Stroud, Greene, and Anderson[43]. These results provide accurate benchmark calculations to evaluate the performance of structural analysis methods for stiffened panel analysis. The stiffened panel configuration considered here is the hat-stiffened panel shown in figure 6.



(a) Mesh using 4-node elements. (b) Mesh using 9-node elements.

Fig. 6 Finite element model of composite hat-stiffened panel.

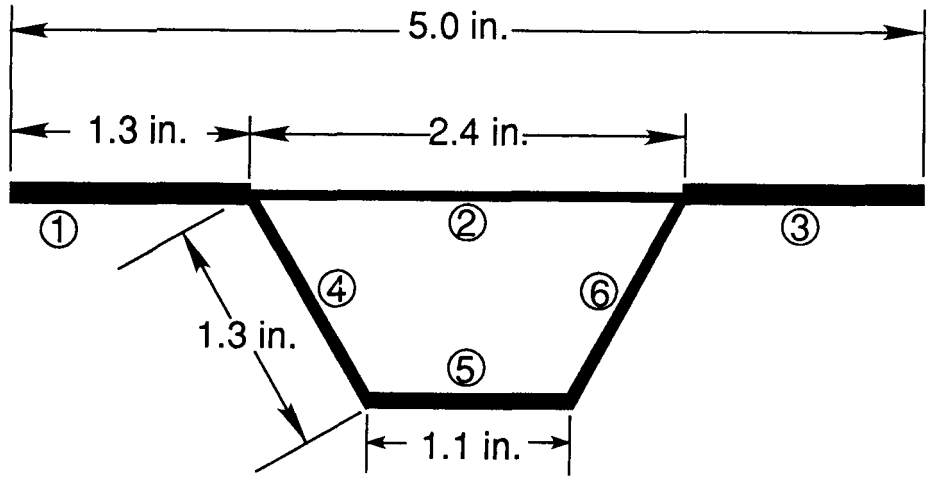


Fig. 7 Repeating element for composite hat-stiffened panel.

The finite element model of the hat-stiffened panel is "parameterized" using a CLAMP procedure given in reference 6. The modeling parameters define a repeating cross-sectional element (or repeating element) as shown in figure 7. This repeating element, composed of six sections or elements (denoted by the circled numbers in figure 7), is repeated along the length and width directions to create a finite element model of a panel 30-inches square with six hat-shaped stiffeners (*i.e.*, six repeating elements for the panel cross-section). The material system is graphite epoxy, and the laminate definitions for each section of the repeating element are given in references 6 and 43. The panel is simply supported on all four edges; these boundary conditions are applied to the stiffener ends as well as to the skin of the panel.

The finite element model shown in figure 6a has 36 4-node quadrilateral elements along the length. One 4-node quadrilateral element is used for each of the repeating element sections 1, 3, 4, 5, and 6 (*i.e.*, one element per section). Two finite elements are used for section 2 of the repeating element (*i.e.*, two elements per section). There are 1512 finite elements and 1369 nodes in the finite element model based on 4-node elements. The finite element model shown in figure 6b has 16 9-node quadrilateral elements along the length. One 9-node quadrilateral element is used for each of the repeating element sections 1, 3, 4, 5, and 6 (*i.e.*, two elements per section). Two 9-node elements are used for section 2 of the repeating element (*i.e.*, four elements per section). There are 672 finite elements and 2607 nodes in the finite element model based on 9-node elements.

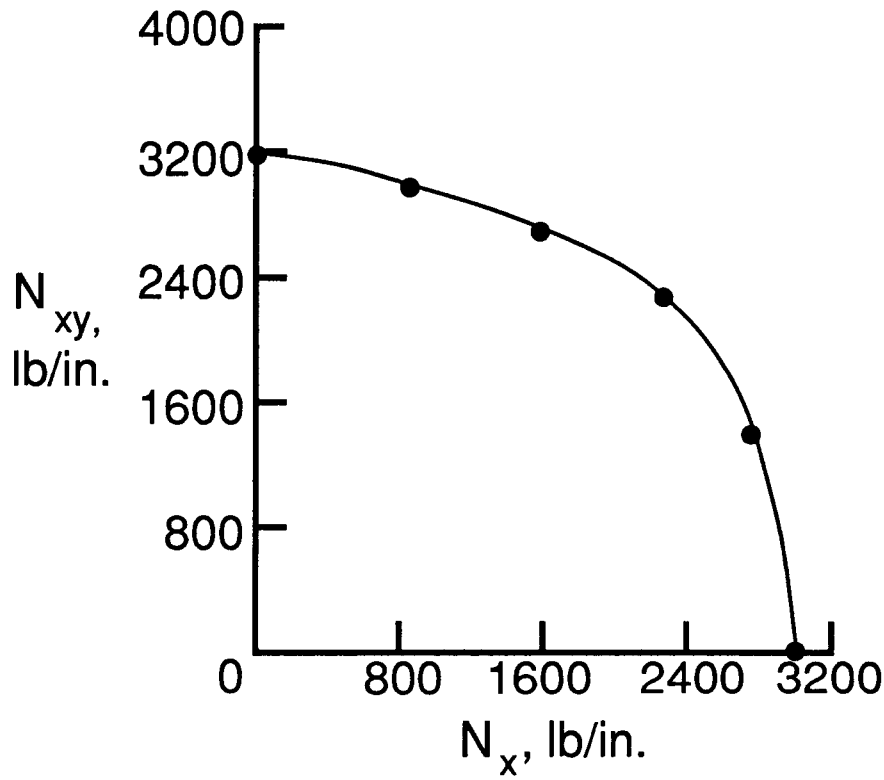


Fig. 8 Buckling load interaction diagram.

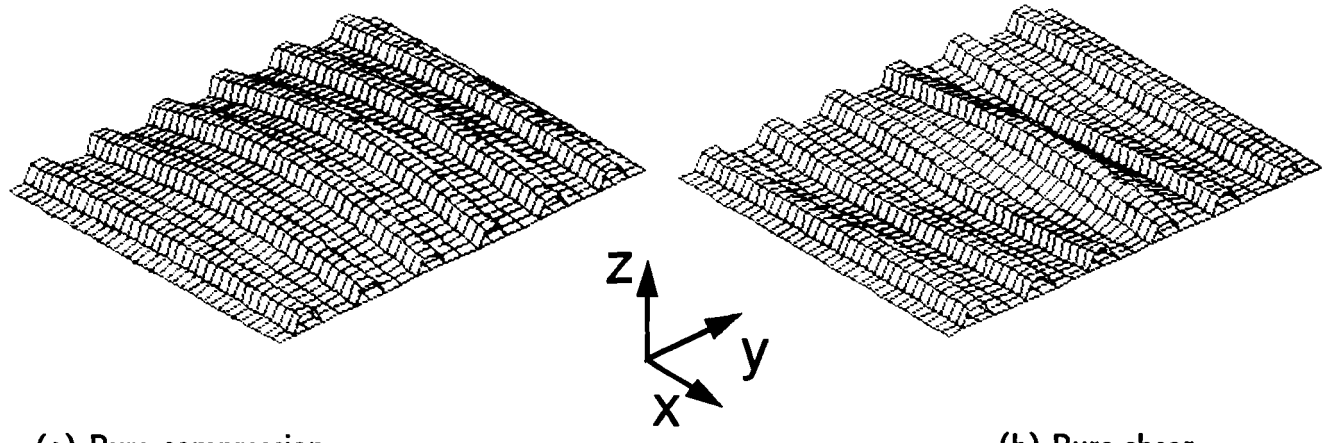
A comparison of the two finite element models reveals a major problem with the use of higher-order elements for this stiffened panel geometry. Nearly twice as many nodes are used in the 9-node element model as are used in the 4-node element model due to the modeling constraints imposed by both the hat-shaped stiffener and the skin of the panel. At least one finite element for each section of the stiffener and skin must be used regardless of the number of nodes in the element.

Linear bifurcation buckling analyses are performed for various combinations of longitudinal compression and shear and for different element types. The buckling load interaction diagram, shown in figure 8, is compiled by specifying the prebuckling stress state in each section of the repeating element (see figure 7) for selected combinations of compression and shear loading. For each prebuckling stress state, the linear algebraic eigenproblem is solved and only one converged eigenpair (buckling load and corresponding buckling mode shape) is extracted. These results, obtained for several load combinations, were used to determine the buckling load interaction shown in figure 8.

Table 4. Comparison of buckling results for composite hat-stiffened panel.

Applied Loading		Eigenvalue		
N_x lb/in	N_{xy} lb/in	NASA TP 2215[43]	Testbed	
			ES5/E410*	ES1/EX97**
0.0	1000.0	3.1920	3.2042	3.1906
300.0	1000.0	2.9320	—	2.9189
600.0	1000.0	2.6800	—	2.6683
1000.0	1000.0	2.3268	2.3317	2.3305
2000.0	1000.0	1.4062	—	1.4018
1000.0	0.0	3.0042	3.0000	2.9956

* ES5/E410 – Bending, membrane plate elements without transverse shear.
** ES1/EX97 – Bending, membrane plate elements with transverse shear.



(a) Pure compression. (b) Pure shear.

Fig. 9 Buckling mode shapes for composite hat-stiffened panel – 4-node model.

Buckling results obtained with the Testbed are compared to the results presented by Stroud, Greene, and Anderson[43] in Table 4. The Testbed results prove to be in excellent agreement with those benchmark values for both the 4-node (ES5/E410) element and the 9-node assumed natural-coordinate strain (ES1/EX97) element. The extremely small variations in the results for the two elements indicate that transverse shear deformations, included in the 9-node element but not in the 4-node element, do not significantly affect the buckling response of this particular panel. Oblique views of the buckling mode shapes for the pure compression case and the pure shear case are shown in figure 9 for the 4-node element model.

Table 5. Selected processor execution times for hat-stiffened panel (ES1/EX97, 1369 nodes, 7700 *dof*, average semi-bandwidth of 3811).

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	4.9	24.4
	E	1.4	35.6
	TOPO	7.8	86.7
Form and Factor Global Stiffness Matrices	ES	44.7	698.3
	K	7.7	120.2
	INV	55.1	2717.0
Eigenvalue Extraction	EIG	354.9	5098.5

Computation times to obtain one converged eigenvalue for one loading condition of the 4-node element model are given in Table 5 for selected Testbed processors. The majority of the CPU time is spent in processors INV and EIG performing the system matrix factorization and eigensolution, respectively. The estimated number of floating point operations to factor the system matrix is 370,206,288. This estimate is computed using the parameter IC1 output by processor TOPO. The parameter IC1 is the number of submatrix multiplications required to factor the system matrix. The estimated number of floating point operations is calculated by multiplying IC1 by the number of degrees-of-freedom at each joint cubed. The result is then multiplied by two to account for the two operations: multiply and add. The compute rates for processor INV are 6.72 and 0.14 MFLOPS (Million FLoating–point Operations Per Second) on the NAS CRAY-2 computer system and a VAX 11/785 computer system, respectively. The ratio of the overall execution time on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system is 17.1 to 1 for a single buckling analysis of the composite hat-stiffened panel.

Composite Blade-Stiffened Panel with Discontinuous Stiffener

Discontinuities and eccentricities are usually present in practical structures. In addition, potential damage of otherwise perfect structures is often an important design consideration. Predicting the structural response in the presence of discontinuities, eccentricities, and damage is particularly difficult when the component is built from graphite-epoxy materials or is loaded into the nonlinear range. Recent interest in applying graphite-epoxy materials to aircraft pri-

mary structures has led to several studies of postbuckling behavior and failure characteristics of graphite-epoxy components (Starnes, Dickson, and Rouse[44]). One goal of these studies has been the accurate prediction of the global response of the composite structural component in the postbuckling range. One study of composite stiffened panels tested a blade-stiffened panel (see Williams et al. [45]). A composite blade-stiffened panel was proof-tested and used as a "control specimen". It was not tested to failure, and was subsequently used in a study on discontinuities in composite blade-stiffened panels (see figure 10). The finite element modeling and analysis needed to predict accurately the nonlinear response of flat blade-stiffened graphite-epoxy panels loaded in axial compression is described in this section.

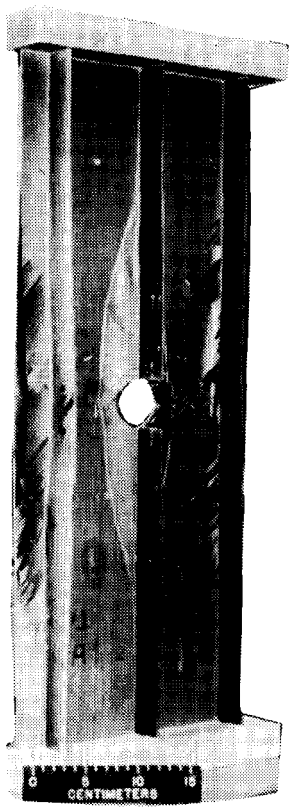
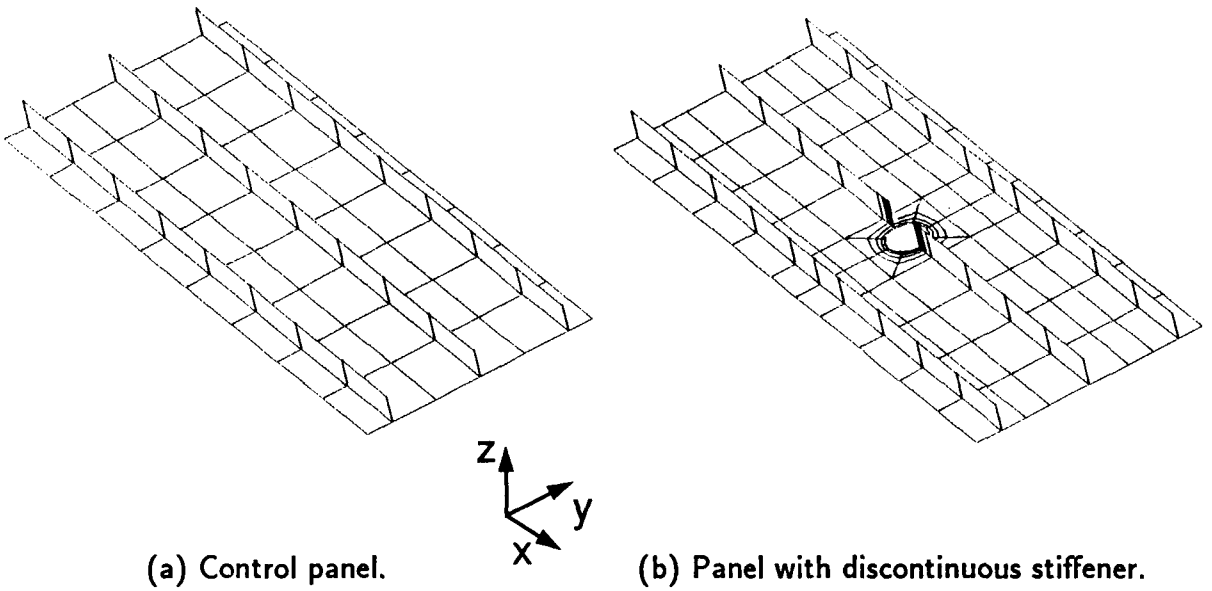


Fig. 10 Composite blade-stiffened panel with discontinuous stiffener.



(a) Control panel. (b) Panel with discontinuous stiffener.

Fig. 11 Finite element model of composite blade-stiffened panel.

The overall panel length is 30 in., the overall width is 11.5 in., the stiffener spacing is 4.5 in., the stiffener height is 1.4 in., and the hole diameter is 2 in. The three blade-shaped stiffeners are identical. The loading is uniform axial compression. The loaded ends of the panel are clamped and the sides are free. The material system for the panel is T300/5208 graphite-epoxy unidirectional tapes with a nominal ply thickness of 0.0055 in. Typical lamina properties for this graphite-epoxy system are 19,000 ksi for the longitudinal Young's modulus, 1,890 ksi for the transverse Young's modulus, 930 ksi for the shear modulus, and 0.38 for the Major Poisson's ratio. The ultimate strains for this material system are 0.110 for longitudinal tension, 0.0086 for longitudinal compression, 0.0036 for transverse tension, 0.0100 for transverse compression, and 0.0150 for shear. The blade stiffeners are 24-ply laminates $([\pm 45/0_{20}/\mp 45])$ and the panel skin is a 25-ply laminate $([\pm 45/0_2/\mp 45/0_3/\pm 45/0_3/\mp 45/0_3/\pm 45/0_2/\mp 45])$.

The finite element models of these composite blade-stiffened panels are shown in figure 11. The finite element model of the "control panel" is shown in figure 11a. A total of 72 9-node quadrilateral ANS shell elements (ES1/EX97) are used in the nonlinear analysis. This model has 323 nodes and 1425 active degrees-of-freedom. The finite element model of the panel with a discontinuous stiffener is shown in figure 11b. A total of 144 9-node quadrilateral ANS shell elements (ES1/EX97) are used in the nonlinear analysis. This model has 628 nodes and 2910 active degrees-of-freedom. The procedure NL_STATIC_1 is used to perform the nonlinear analysis.

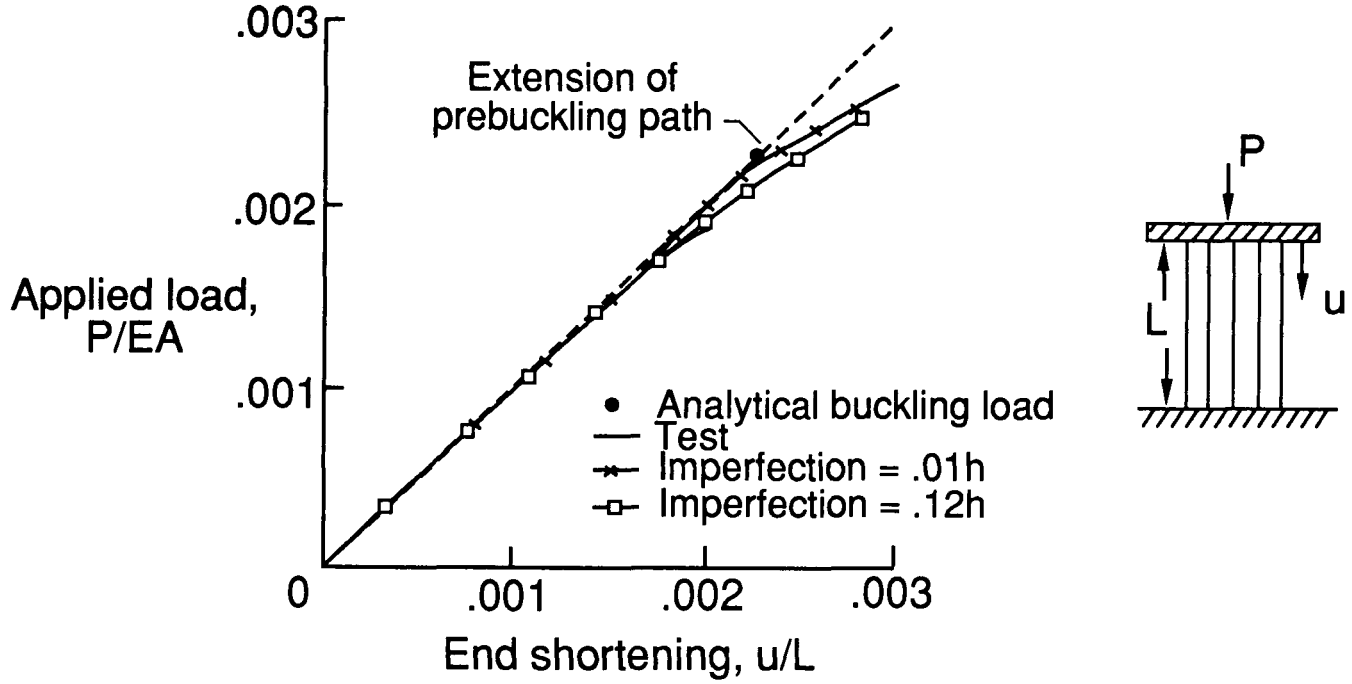
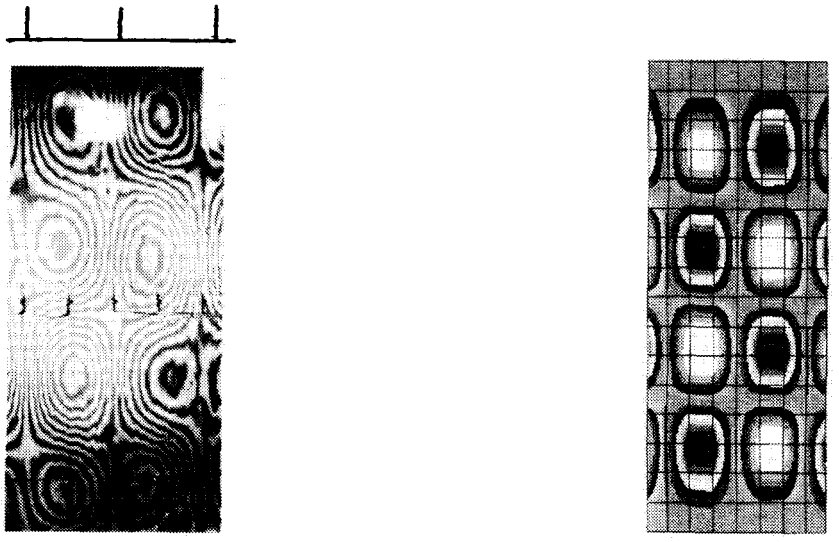


Fig. 12 Test and analysis correlation for end-shortening results for blade-stiffened panel.

The postbuckling response of the composite blade stiffened panel (*i.e.*, the "control panel") is determined prior to analyzing the configuration with the discontinuous stiffener. End-shortening results are shown as a function of the applied compressive load in figure 12. The end-shortening u is normalized by the overall panel length L and the applied load is normalized by the panel's prebuckling extensional stiffness EA. The results indicate that the panel was loaded beyond its buckling load. The dashed curve represents an extension of the panel's prebuckling path. The analytically-predicted linear buckling load is denoted by the filled symbol on figure 12. Since the test buckling load is below the analytically-predicted value, it is suspected that the panel had some initial geometric imperfections that caused buckling below the predicted value.

The influence of initial geometric imperfections on the postbuckling response is also shown in figure 12. The shape of the imperfection is selected to be the first buckling mode shape scaled by a percentage of the panel skin thickness. Processor IMP is used to calculate and impose the initial geometric imperfection. The postbuckling response for two values of the imperfection amplitude are shown in figure 12. As the imperfection amplitude increases, the influence of geometric nonlinearities becomes more dominant at lower values of the applied compressive load. The analytical results obtained using a maximum imperfection amplitude of 12% of the panel skin thickness shows good correlation between test and analysis. Comparison



(a) Moire-fringe pattern. (b) Analytical contour plot.

Fig. 13 Comparison of moire-fringe pattern from test with contour plot of out-of-plane deflections from analysis.

of a moire-fringe pattern of the buckled panel skin and a contour plot of the analytically-obtained out-of-plane deflections for the last nonlinear solution is shown in figure 13. These results indicate that the postbuckling out-of-plane deflections from both test and analysis have the same shape over the entire panel.

End-shortening results are shown in figure 14 for the "control specimen" and for the configuration with a discontinuous stiffener. These results indicate that the presence of the discontinuity markedly changes the structural response of the panel. The structural response of the "control specimen" is typical of stiffened panels. Two equilibrium configurations are exhibited; namely, the prebuckling configuration and the postbuckling configuration. The structural response of the configuration with a discontinuous stiffener is nonlinear from the onset of loading due to the eccentric loading condition and the cutout. This problem has served as one of the CSM focus problems for identifying and resolving analysis deficiencies associated with the nonlinear global/local stress analysis of composite structures. This problem was selected as a focus problem because it has characteristics which often require a global/local analysis. These characteristics include a discontinuity, eccentric loading, large displacements, large stress gradients, high inplane loading, and a brittle material system. This problem represents a generic class of laminated composite structures with discontinuities in which the interlaminar stress state becomes important.

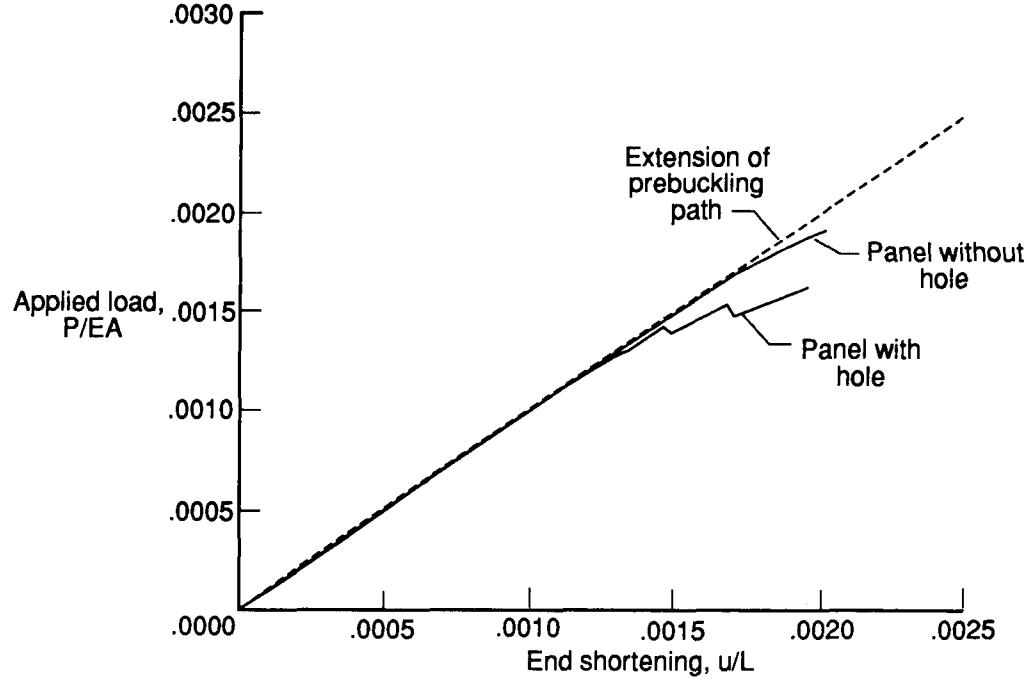


Fig. 14 End-shortening results for composite blade-stiffened panels.

End-shortening results are shown in figure 15 as a function of the applied compressive load. The blade-stiffened panel with a discontinuous stiffener was tested to failure. Local failures occurred prior to overall panel failure as evident from the end-shortening results shown in figure 15. Good agreement between test and analysis is shown up to the load where local failures occurred. The analytically-obtained out-of-plane deflection w at the edge of the hole and blade stiffener normalized by the panel skin thickness t is shown as a function of the applied load in figure 16. These large out-of-plane deflections indicate that the response is nonlinear from the onset of loading.

Oblique views of two deformed shapes with exaggerated deflections are shown in figure 17 for two values of applied compressive load. The two values are denoted in figure 15 by Load A and Load B. These deformed shapes are similar to each other indicating that the primary equilibrium path is being followed and buckling does not occur. Contour plots of the longitudinal inplane stress resultant N_x for Loads A and B are also shown in figure 17. These N_x distributions reveal several features of the global structural behavior of this panel. First, away from the discontinuity, the N_x distribution in the panel skin is nearly uniform and approximately half the value of the N_x in the outer two blade stiffeners. Second, load is diffused from the center discontinuous stiffener into the panel skin rapidly such that the center stiffener has essentially no N_x load at the edge of the hole. Third, the N_x load in the outer stiffeners

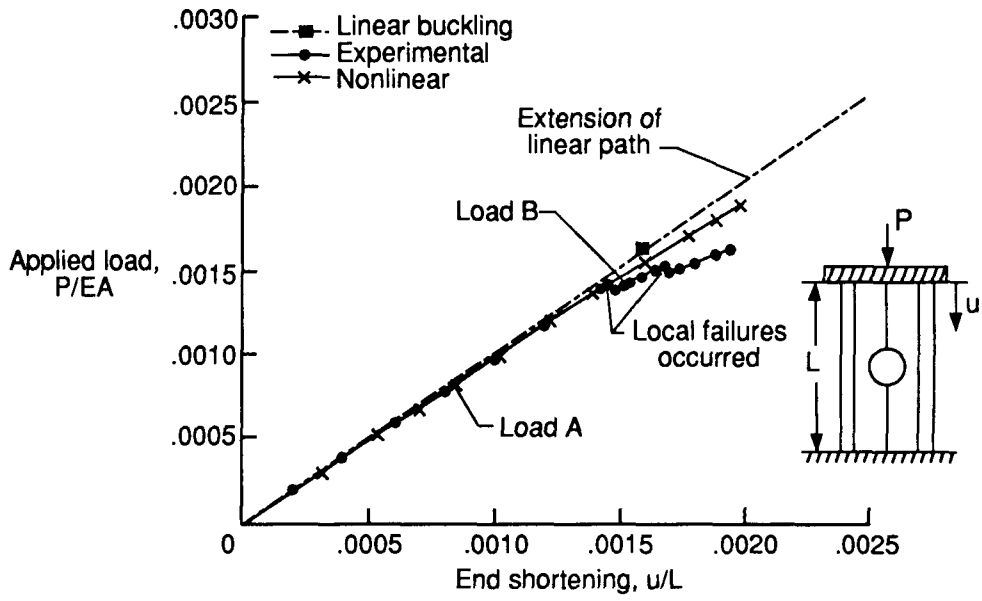


Fig. 15 Test and analysis correlation for end-shortening results for composite blade-stiffened panel with a discontinuous stiffener.

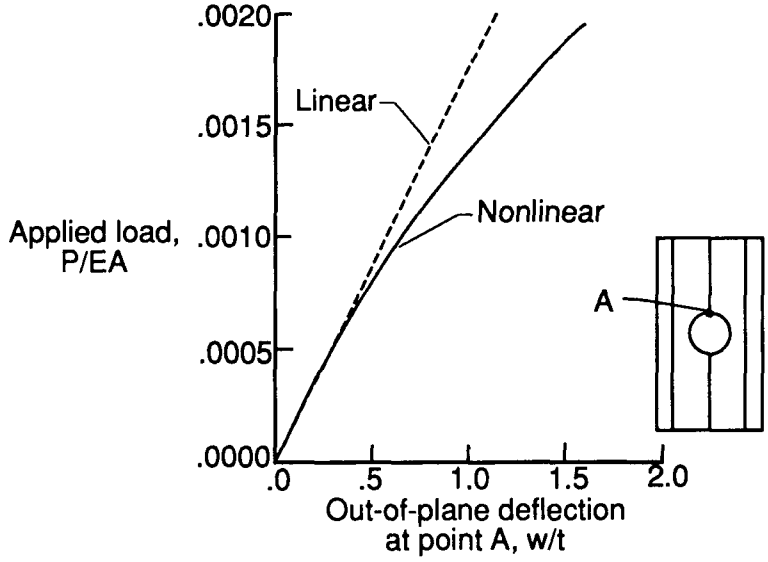
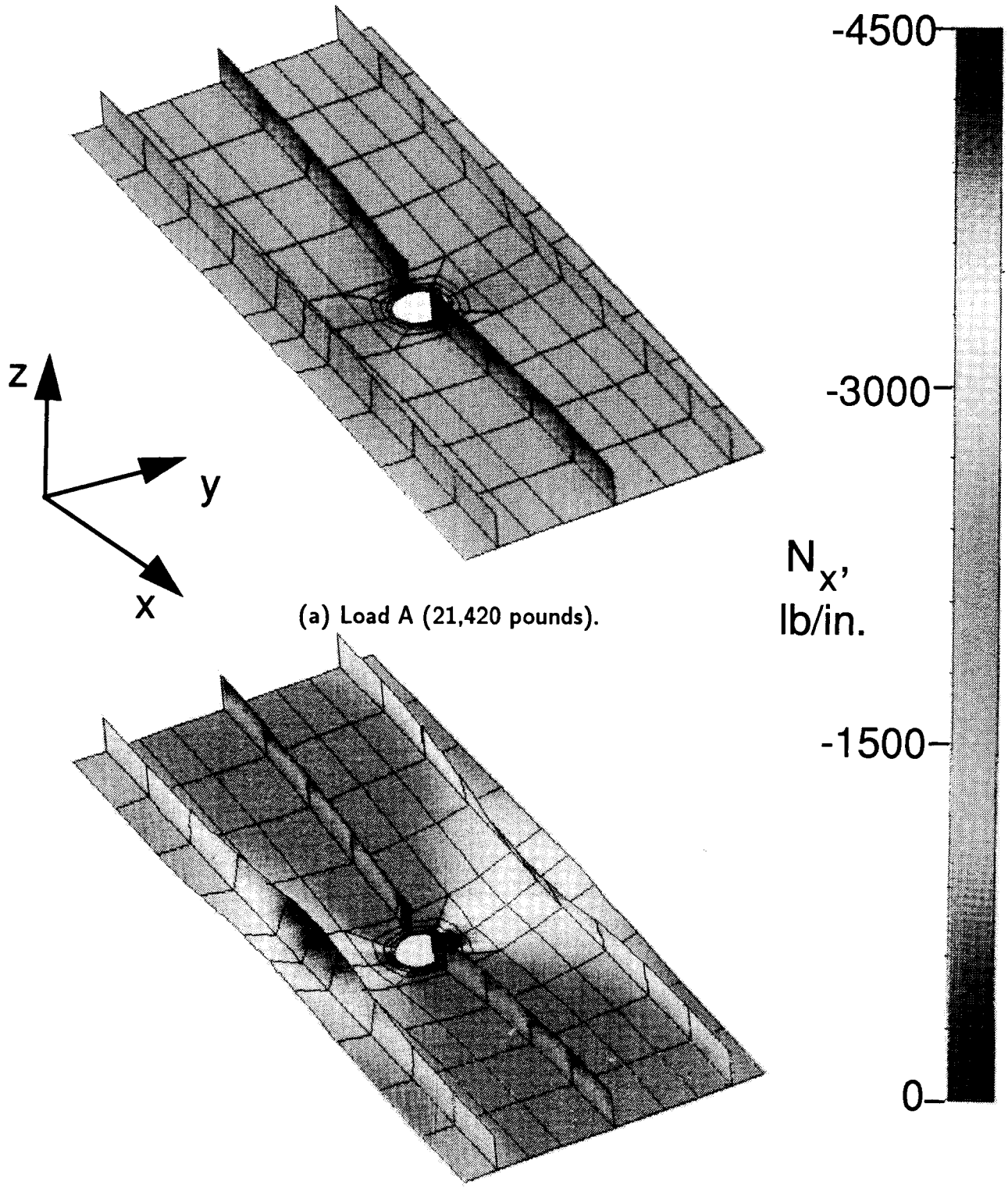


Fig. 16 Out-of-plane deflection at hole and blade stiffener.

increases towards the center of the panel and is concentrated in the blade tips (*i.e.*, away from the stiffener attachment line to the panel skin). Fourth, the N_x load in the panel skin near the center of the panel is much greater than the N_x load in other portions of the panel skin.



(a) Load A (21,420 pounds).

(b) Load B (37,800 pounds).

Fig. 17 Deformed geometry shapes with N_x distributions.
(see Appendix for color figures)

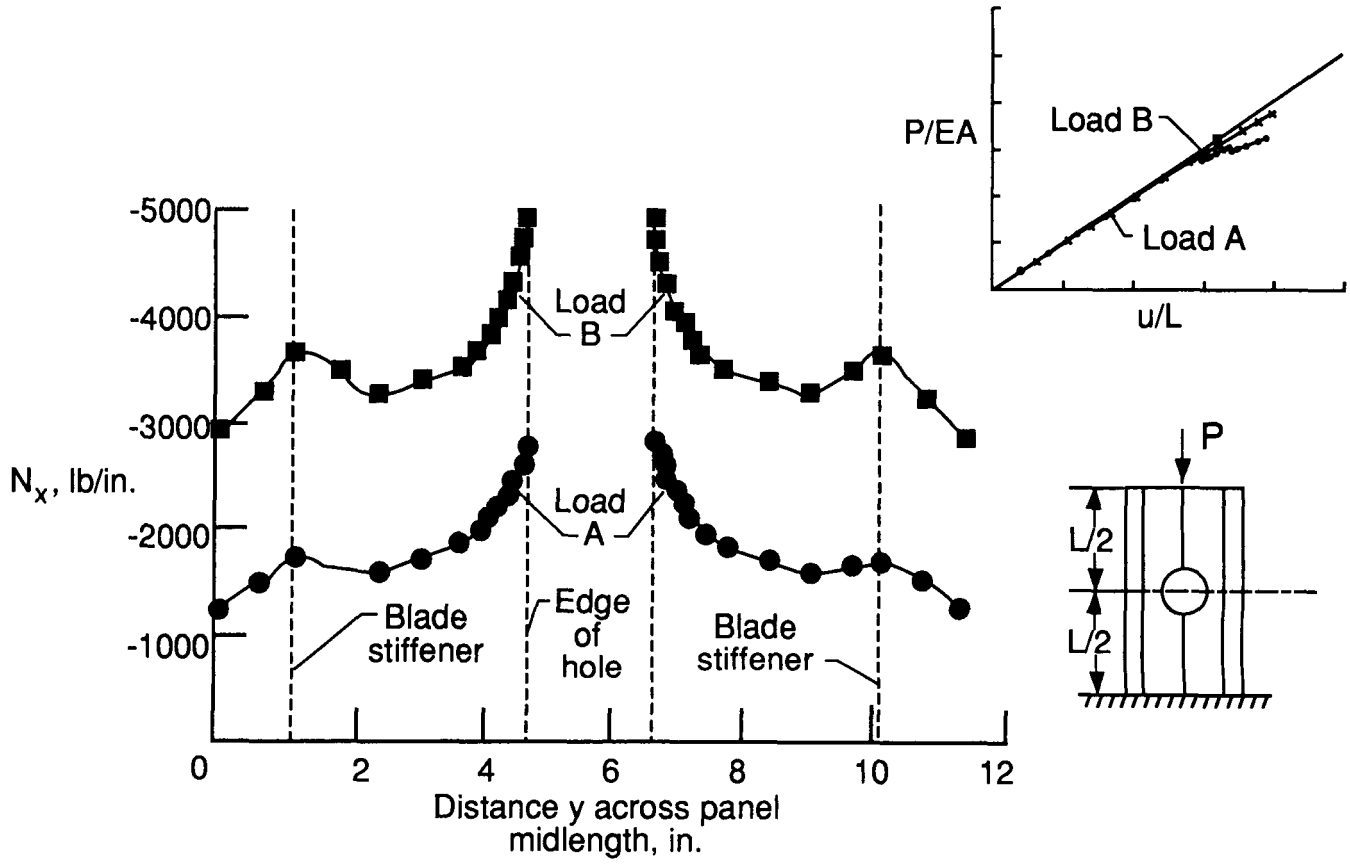
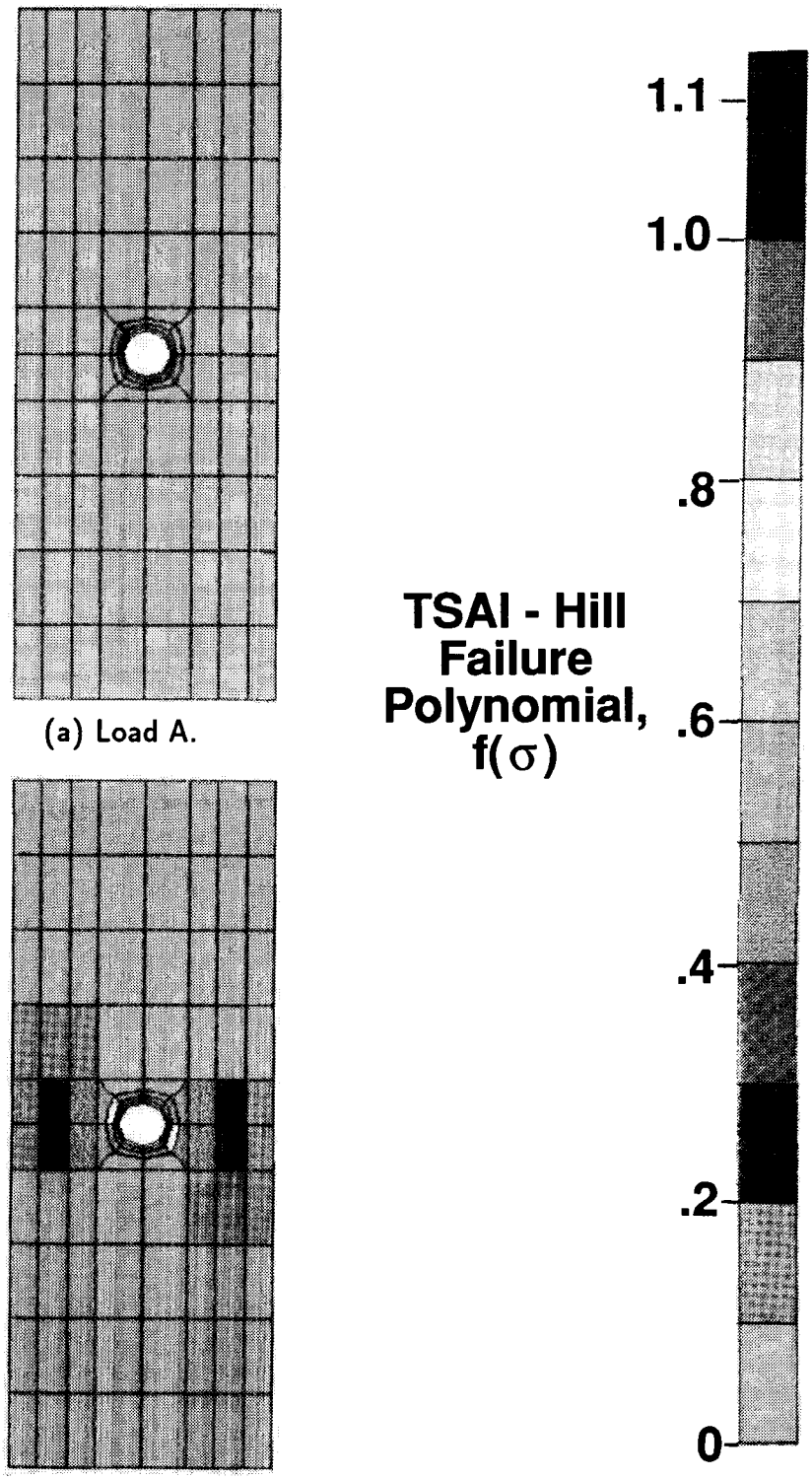


Fig. 18 Longitudinal inplane stress resultant N_x distributions at panel midlength.

Longitudinal inplane stress resultant N_x distributions at panel midlength are shown in figure 18 as a function of distance from the hole for Loads A and B. The results indicate that high inplane stresses and a high stress gradient exist near the hole. As the load increases, the longitudinal inplane stress resultant increases and the stress gradient increases near the hole and blade stiffeners. The outer blade stiffeners carry a larger percentage of the overall load at Load B than at Load A.

These high inplane stresses and stress gradients coupled with the large out-of-plane displacements and the free edge of the hole may cause material nonlinearities, local failures, and/or delaminations to develop in order to provide local stress relief mechanisms (like plasticity in metal structures) near the hole and blade stiffener. To determine when local failures initially occur, the analytically-obtained stress distributions may be used to assess various failure criteria to estimate when first-ply failure occurred.



(a) Load A.

(b) Load B.

Fig. 19 Tsai-Hill criterion for outer fiber surface of panel skin.
(see Appendix for color figures)

Table 6. Selected processor execution times for composite blade-stiffened panel with a discontinuous stiffener.
(ES1/EX97, 628 nodes, 2910 *dof*, average semi-bandwidth of 439).

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	3.2	6.7
	E	0.3	11.4
	TOPO	1.6	15.8
Form and Factor Global Stiffness Matrix	ES	50.1	821.0
	K	1.4	26.8
	INV	8.3	347.0
	VEC	2.9	18.4
	SSOL	0.9	28.8
Each Iteration	SSOL	0.9	35.3
	VEC	2.1	19.4
	ES	13.5	230.1

Processor FPF is used to perform this assessment by evaluating various failure criteria and creating a dataset containing the values of the failure criteria for each element. Color assignment of the elemental values of the Tsai-Hill failure criterion obtained for the outer fiber (+45-degree ply) surface of the panel skin are shown in figure 19 for Loads A and B. First-ply failures are detected near the edge of the hole. Values less than one indicate no failure; values greater than or equal to one indicate failure. Progressive failure analysis techniques are needed to predict accurately the nonlinear response.

Computation times for the nonlinear analysis of the composite blade-stiffened panel with a discontinuous stiffener are given in Table 6 for selected Testbed processors. Most of the CPU time was spent in processor ES which computes new elemental tangent stiffness matrices. The global tangent stiffness matrix was re-evaluated and factored fifteen times and a total of sixty-four iterations were required to predict the nonlinear structural response of this panel. The estimated number of floating point operations to factor the system matrix is 59,337,360. The compute rates for processor INV are 7.2 and 0.17 MFLOPS on the NAS CRAY-2 computer system and a VAX 11/785 computer system respectively. The ratio of the overall execution times on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system is 17.2 to 1 for the nonlinear response of the composite blade-stiffened panel with a discontinuous stiffener.

Table 7. Performance of direct solvers in processor BAND (628 nodes, 2910 *dof*, average semi-bandwidth of 439).

Method	NAS CRAY-2 (CPU seconds)	Compute Rate, (MFLOPS)
LINPACK	27.1	64.1
kji Choleski	27.4	63.4
kji Choleski*	17.7	98.2
kji Choleski**	12.7	136.9
kji Profile	12.7	57.1
kji Profile*	7.9	92.9
kji Profile**	5.6	129.4

* Loop unrolling to level 4.
**Loop unrolling to level 4 and use of local memory.

Performance results obtained using various direct solvers implemented in processor BAND are shown in Table 7. Increased performance is obtained by using "loop unrolling" to level 4, where a column of $\boldsymbol{K}$ is updated by four columns at a time rather than one, and by also exploiting the local memory of the CRAY-2 computer system. For this problem, only the profile method in processor BAND performs better than processor INV.

Three-Dimensional Composite Analysis

Detailed stress analysis of composite structures is often required to determine accurately through-the-thickness (or interlaminar) stress distributions. Some sources of interlaminar stress gradients are depicted in figure 20. These sources include free-edge effects, holes, ply drop-off (*e.g.*, tapered stiffener attachment flanges), bonded joints (*e.g.*, panels with secondarily-bonded stiffeners), and delaminations or damage. To study these effects, "quasi" three-dimensional formulations and two-dimensional finite element models have been used. To establish the quality and reliability of numerical solution procedures, researchers have studied the elastic response of finite-width composite plates (*e.g.*, see Pipes and Pagano[46] and Raju, Whitcomb, and Goree[47]). These studies developed an understanding of the mechanisms of interlaminar load transfer in fiber-reinforced composite materials. A symmetric four-ply laminate is shown in figure 21a. The laminate is assumed to be long in the x-direction. When this laminate is subjected to uniform axial strain ϵ_0 in the x-direction, all x =constant planes away from the ends deform in the same manner. Therefore, away from the ends, the displacements are

assumed to be

$$u(x,y,z) = U(y,z) + \epsilon_0 x$$

$$v(x,y,z) = V(y,z)$$

$$w(x,y,z) = W(y,z)$$

where, U, V, and W are displacement functions expressed in terms of y and z alone. These equations describe a "quasi" three-dimensional problem. The term "quasi" is used because, displacements occur in the three coordinate directions, but the gradients of U, V, and W with respect to the x-coordinate are zero. Thus, only an x =constant plane may be analyzed to obtain the stresses in the laminate. A typical x =constant plane is shown in figure 21b.

However, the analyses reported herein solve this problem using a three-dimensional finite element model. The hybrid natural-coordinate assumed-strain solid elements in the CSM Testbed (*i.e.*, ES3/EX20) are used for the analysis of the problem. The assumed strain field is chosen such that the standard isoparametric displacement-based formulation for a 20-node brick element is simulated. The quality of the results using the three-dimensional model are discussed. The problem considered in this section is a rectangular composite plate loaded by uniform axial extension as shown schematically in figure 21. The plate is a 4-ply $\left([0/90]_s\right)$ laminate with a nominal ply thickness of 0.005 in. Lamina properties used in this study are 20,000 ksi for the longitudinal Young's modulus, 2,100 ksi for the transverse Young's moduli, 850 ksi for the shear moduli, and 0.21 for the Poisson's ratios ν_{12}, ν_{13}, and ν_{23}. A uniform extensional strain ϵ_0 of 0.001 in./in. is applied.

The $[0/90]_s$ laminate is chosen to make the problem simple by being able to exploit symmetry. With this choice, only one-eighth of the problem needs to be analyzed due to the existence of three planes of symmetry. A $[\pm 45]_s$ laminate has only one plane of symmetry, and hence one half of the problem needs to be analyzed.

Three finite element models were developed. A typical model is shown in figure 22. Each layer is idealized as a homogeneous, elastic, orthotropic material. Two elements are used through the thickness of each layer for all models. The discretization along the length (*i.e.*, the long direction) consists of equally-spaced grid points. The discretization along the width of the plate is graded in a geometric progression.

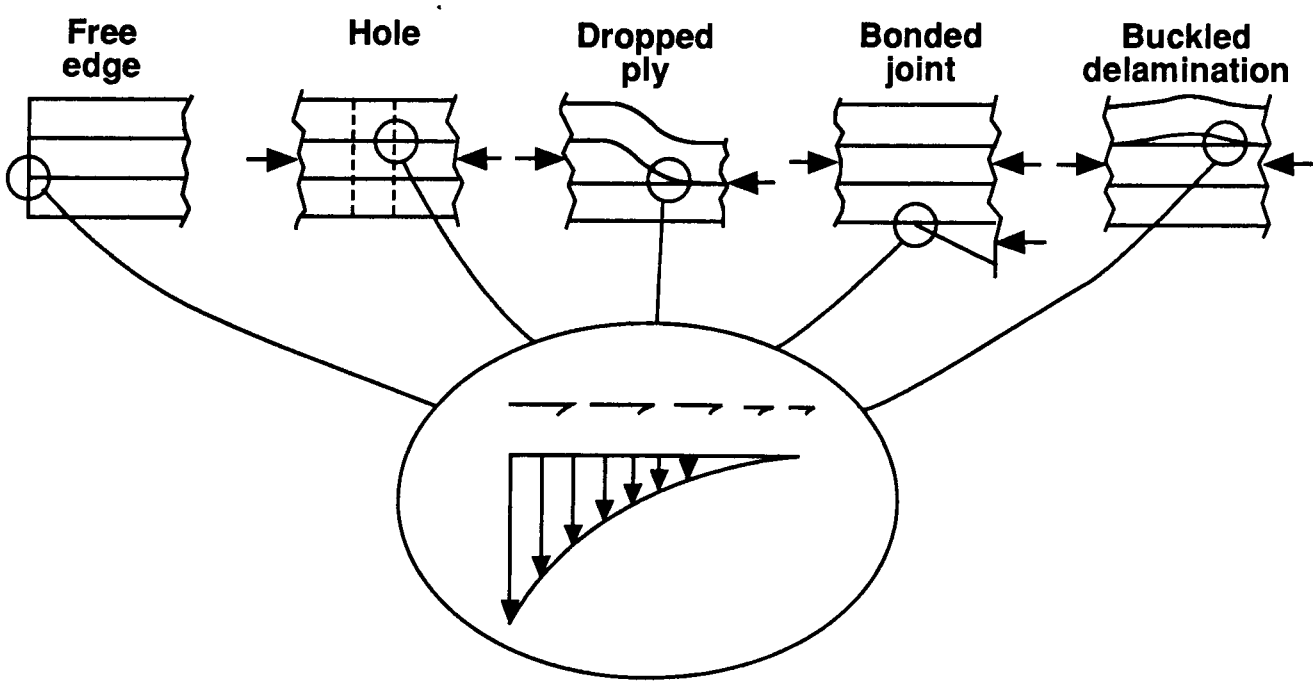
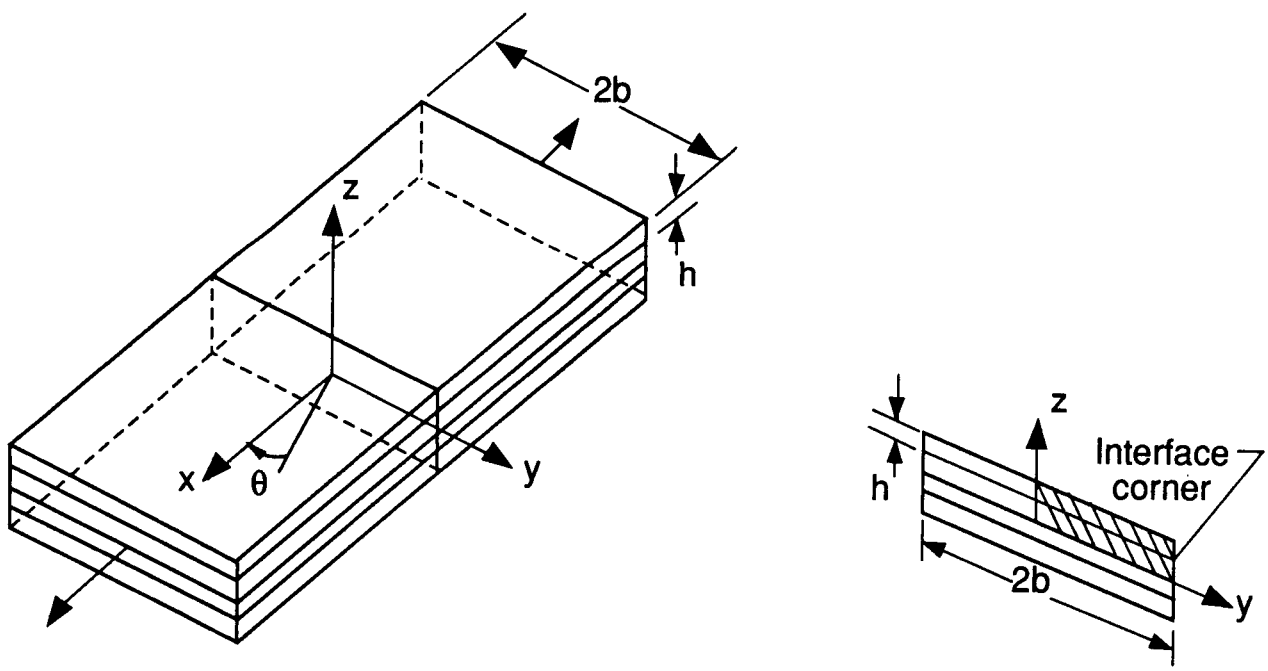
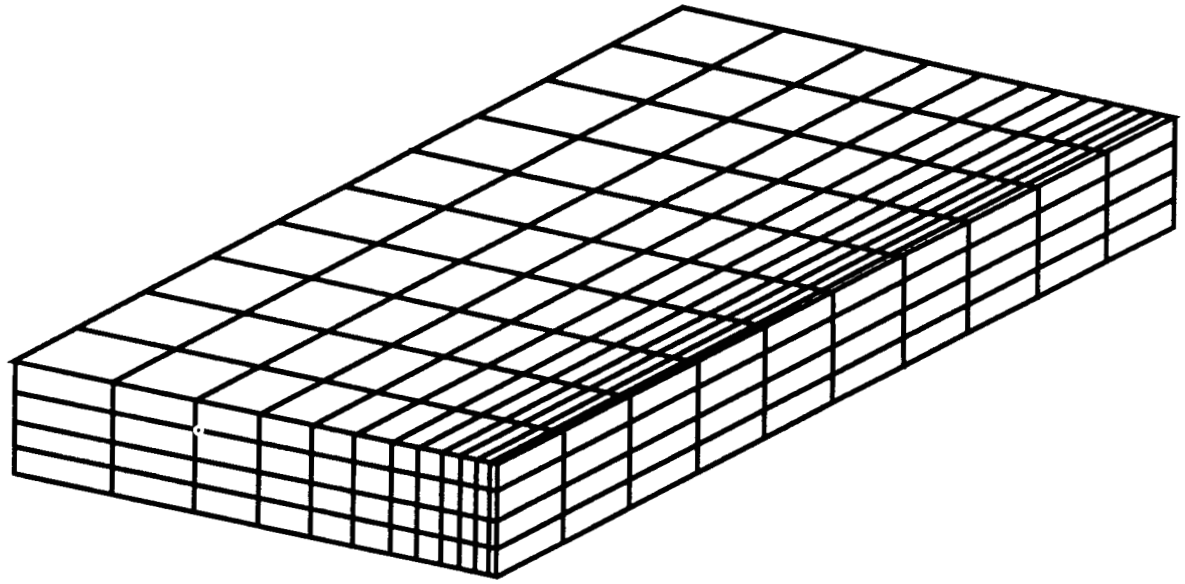


Fig. 20 Sources of interlaminar stress gradients.

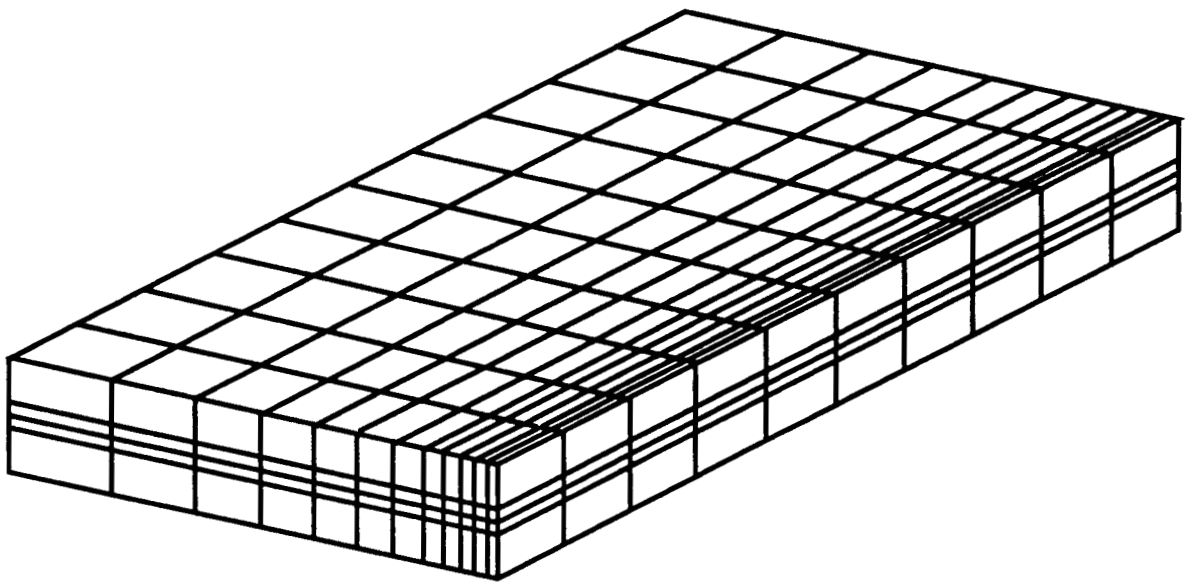


(a) Four-ply laminate (b) An $x =$ constant plane

Fig. 21 Three-dimensional composite problem.



(a) Model 1.



(b) Models 2 and 3.

Fig. 22 Finite element models of the three-dimensional composite plate.

The half-length of the plate is divided into ten equal segments for all models. The half-width of the plate is divided into twelve segments with a progression factor of 0.81 for all models. The discretizations maintain a minimum aspect ratio of 0.4 (*i.e.*, the length of the element in the width direction is 0.4 times the length of the element in the thickness direction) for elements on the free edge. This discretization results in 2597 nodes and 480 20-node brick elements (ES3/EX20) for all three models. The three models differ in their discretization. For Model 1, the discretization in the thickness direction consists of equally-spaced grid points. For Model 2, the discretization in the thickness direction is such that the elements around the interface are one-third as thick as the elements away from the interface. Model 3 is the same as Model 2 except that the midside nodes of the elements around the interface are moved to the quarter-point position from the interface. Also the midside nodes of the elements on the free-edge are moved to the quarter-point position from the free-edge. This process of moving the midside nodes to the quarter-point positions produces the so called "quarter-point singularity elements". Therefore, in Model 3 a singularity condition is induced at the interface (on both sides of the interface) in the thickness direction and also a singularity condition is induced at the free-edge in the width direction.

The interlaminar normal stress σ_z distributions along the interface between the 90-degree layer and the 0-degree layer are shown in figure 23 for all three models. The stresses shown are the averaged nodal values of stresses calculated directly at the nodes of each element. For all models, the interlaminar normal stress distribution along the interface between layers is virtually identical as shown in figure 23, except at the free-edge where the refined models (Models 2 and 3) give higher values, respectively. The accepted solution for this problem is that the interlaminar normal stress σ_z approach positive infinity at the free-edge. The finite element solution does not produce a value of infinity, but it should be rapidly approaching it as shown in figure 23. Much higher values can be obtained by successively refining the mesh (*e.g.*, see Raju, Whitcomb, and Goree[47]). The normal stress σ_z distribution along the interface for Model 3 exhibits a small anomaly near the free-edge as shown in figure 23 which may be attributed to the numerical difficulties caused by the relatively coarse mesh employed. Of course, the manner in which the stresses at the element nodes are calculated (*i.e.*, calculated directly at the nodes or calculated at the integration points and then extrapolated to the nodes) may alter the values of the stresses, and the amplitude of the anomaly may be attenuated or heightened when other methods of calculating the stresses are employed.

The normal stress σ_z distribution at the free-edge in the thickness direction are shown in figure 24 for all three models. Again, the stresses shown are the averaged nodal values of stresses calculated directly at the nodes of each element. For all models, the normal stress distributions at the free-edge in the thickness direction exhibit similar behaviors as shown in figure 24. However, the peak values near the interface are heightened for the refined models (Models 2 and 3), respectively. It is noted that the values of σ_z from all three models at $z = 0$ (*i.e.*, at the interface between the two 90–degree layers) exhibit little difference. However, at $z = .01$ inches (*i.e.*, the top surface of the plate), the normal stress σ_z must vanish. Model 1 very nearly satisfies this condition, but Models 2 and 3 give non-zero values at this point. Again, this behavior may be attributed to the numerical difficulties caused by the relatively coarse mesh employed. This behavior is particularly evident for Models 2 and 3, in which the discretization in the thickness direction is refined towards the interface at the expense of a coarser discretization away from the interface (*i.e.*, $z = .01$ inches). Again, it should be mentioned that the manner in which the stresses at the element nodes are calculated (*i.e.*, calculated directly at the nodes or calculated at the integration points and then extrapolated to the nodes) may alter the values of the stresses and the values of σ_z at $z = .01$ inches may or may not be closer to zero when other methods of calculating the stresses are employed.

These results demonstrate the analysis capabilities in the CSM Testbed and show that the solid elements recover the established results and trends for free-edge stress analysis problems in composite structures.

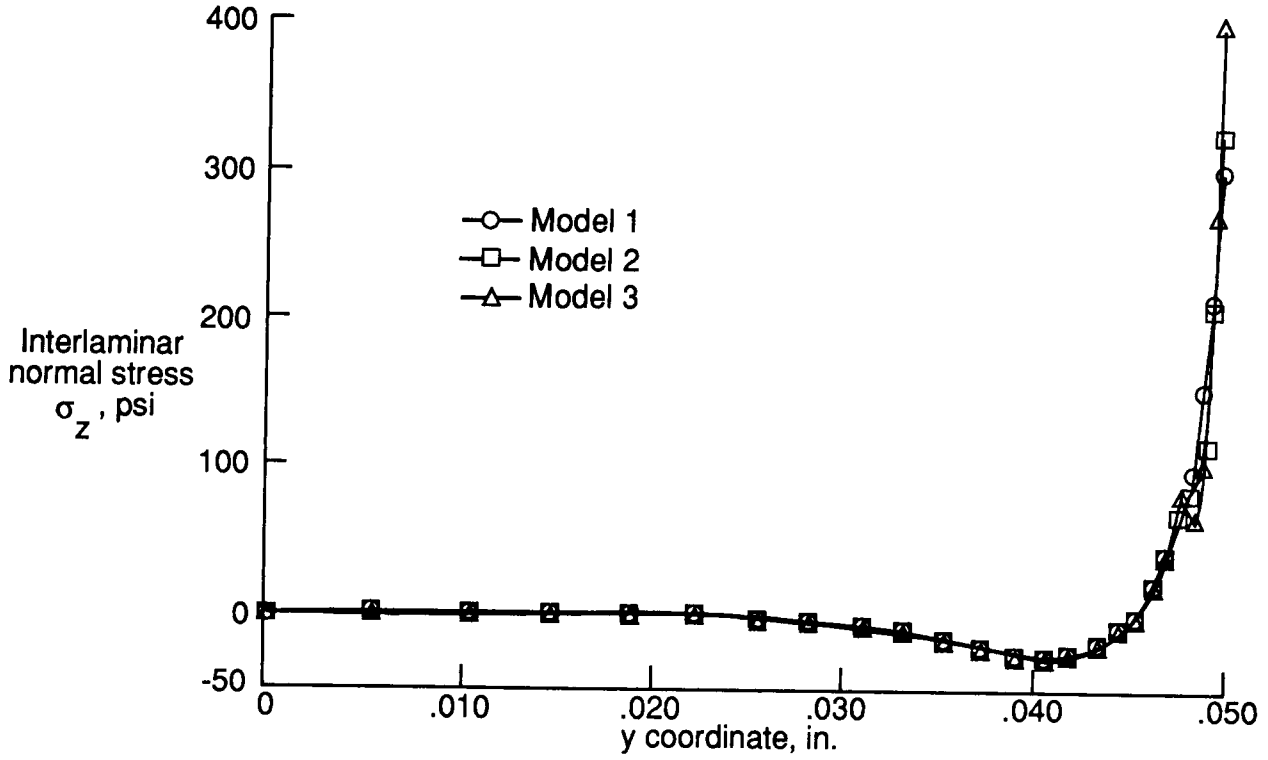


Fig. 23 Interlaminar normal stress σ_z distributions along the interface between the 90–degree and 0–degree layers.

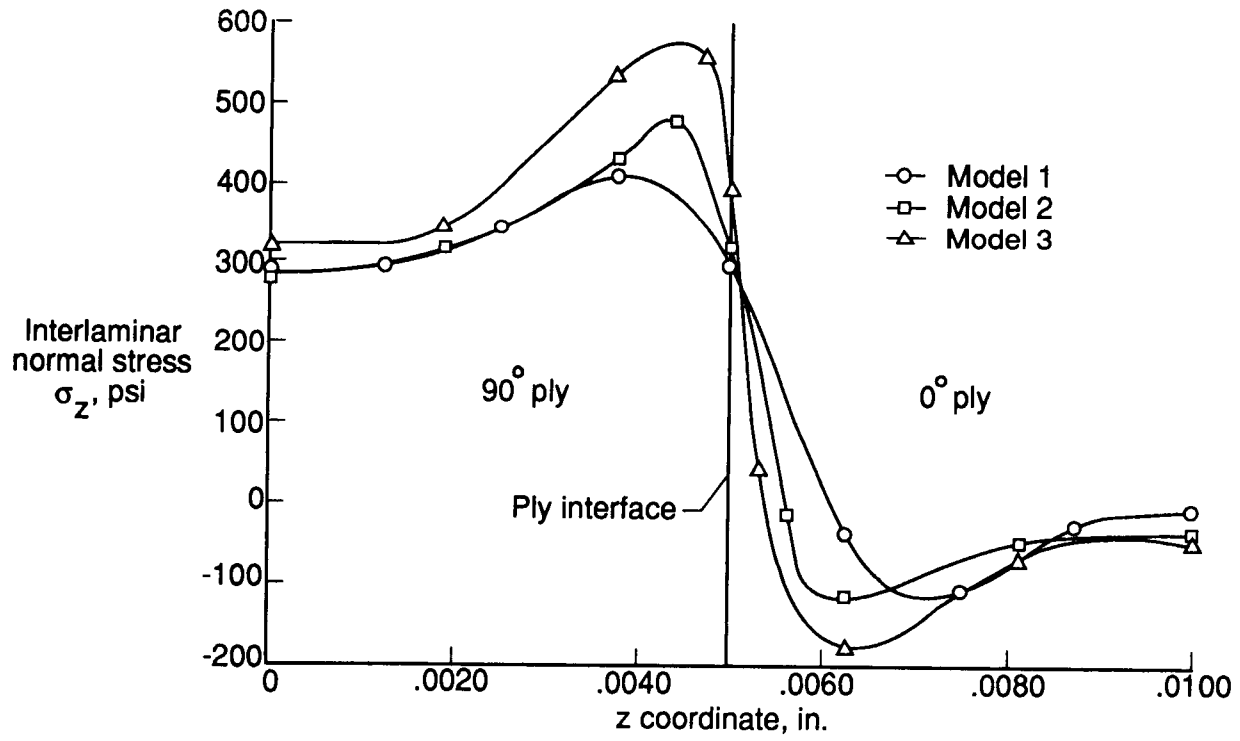


Fig. 24 Normal stress σ_z distributions at the free-edge in the thickness direction.

Table 8. Selected processor execution times for 3-D stress analysis (2597 nodes, 6883 *dof*, average semi-bandwidth of 663).

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)
Mesh Generation	ELD	3.9
	E	.5
	TOPO	2.3
Form and Factor Global Stiffness Matrices	ES	280.9
	K	7.1
	SPK	451.1
Stress Recovery	ES	278.4

Computation times for the free-edge stress analysis are given in Table 8 for selected Testbed processors. Most of the CPU time is spent in processor ES calculating the element stiffness matrices and the nodal values of stresses in the postprocessing phase. The remaining time is, for all practical purposes, spent in the processor SPK factoring the global stiffness matrix. The number of floating point operations required to factor the system stiffness matrix is 865,404,512. The compute rates for processor SPK is 2.2 MFLOPS on the NAS CRAY-2 computer system. Processor ELD used a very small amount of time because the model is generated using PATRAN, and processor ELD only read the data file. Processor TOPO also used a very small amount of time because only the KMAP dataset for guiding the assembly of the global stiffness matrix is generated. Processor SPK does not require the AMAP dataset for factoring the global stiffness matrix that is ordinarily generated by processor TOPO. These analyses have only been performed on the NAS CRAY-2 computer system because of the large memory requirement.

Circular Cylindrical Shell with Two Rectangular Cutouts

A common structural configuration is that of a cylindrical shell (*e.g.*, storage tanks, pipelines, aircraft fuselages, rocket motor cases). Shell-type structures are generally sensitive to initial geometric imperfections and to local discontinuities such as cutouts. Many aerospace vehicles contain large cutouts (*e.g.*, access holes, windows). The strength of these structures is limited to its static collapse load. Static collapse is characterized by two modes of behavior: bifurcation buckling and nonlinear collapse (see Bushnell[48]). Predicting the nonlin-

ear collapse behavior of these shell-type structures is a difficult and computationally-intensive analysis problem.

The circular cylindrical shell with two rectangular cutouts loaded by uniform end shortening shown in figure 25 is representative of this class of structures. This problem has also been used as a benchmark problem by Hartung and Ball[49] for shell analysis computer codes and by Almroth and Brogan[50] for assessing shell elements. These researchers considered only one-eighth of the shell in their analyses. The results reported herein are compared with their results, and hence only one-eighth of the shell is modeled. Three finite element models were considered in this study as shown in figure 26. Mesh 1 has 19 rows and 27 columns of nodes (including corner and midside nodes) and is composed of 101 9-node quadrilateral shell elements (ES1/EX97), 449 nodes, and 2012 active degrees-of-freedom as shown in figure 26a. Mesh 2 has 25 rows and 41 columns of nodes and is composed of 210 9-node quadrilateral shell elements (ES1/EX97), 905 nodes, and 4182 active degrees-of-freedom as shown in figure 26b. Mesh 3 has the same nodal layout as Mesh 1 but is composed of 404 4-node quadrilateral shell elements (ES1/EX47). That is, Mesh 1 uses higher-order (biquadratic shape function), curved shell elements while Mesh 3 uses low-order (bilinear shape function), flat shell elements with the two discretizations having the same number of nodes (compare figure 26a with figure 26c).

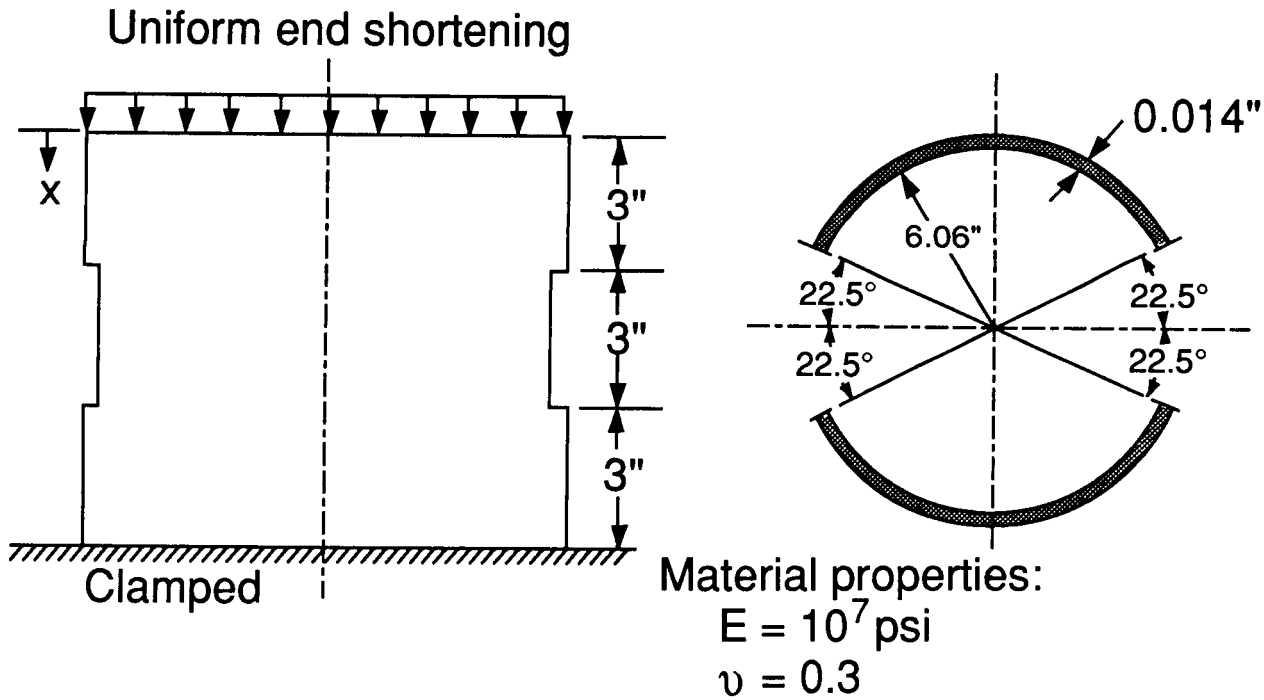
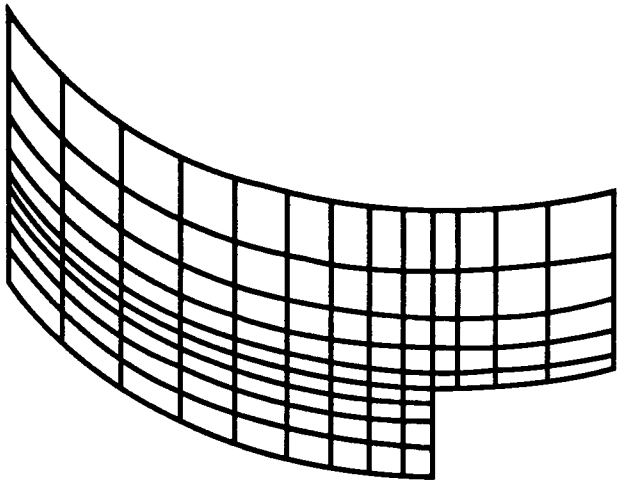
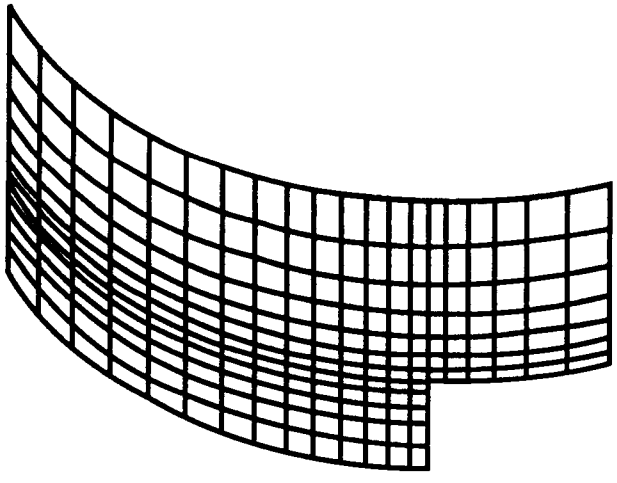


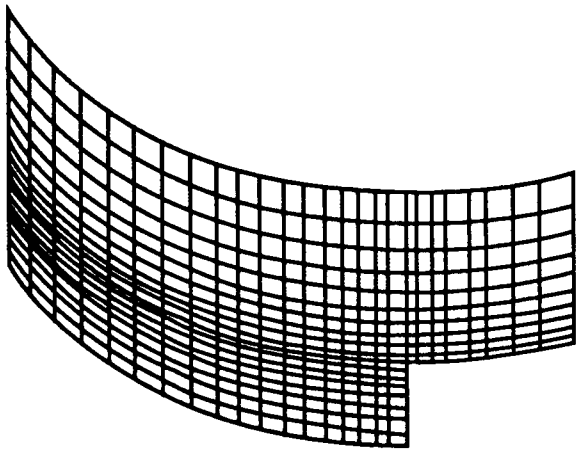
Fig. 25 Cylinder with cutouts – geometry, properties, and loading



(a) Mesh 1 (101 ES1/EX97 elements, 449 nodes).



(b) Mesh 2 (210 ES1/EX97 elements, 905 nodes).



(c) Mesh 3 (404 ES1/EX47 elements, 449 nodes).

Fig. 26 Finite element models of cylinder with cutouts.

Table 9. Elastic collapse loads for cylinder with cutouts – $\frac{1}{8}$-model results.

Source	Element Name	Mesh(1)	Elastic Collapse Load, pounds
Hartung and Ball[49]	(2)	17 × 22	2109
Almroth and Brogan[50] (STAGSC-1)	411	11 × 13	3698
		17 × 19	3060
		23 × 27	2922
		29 × 37	2750
	440	17 × 19	3733
		23 × 29	3098
STAGSC-1	410	17 × 19	3290
		23 × 29	2962
CSM Testbed	ES1/EX97	19 × 27	2846
		25 × 41	2828
	ES1/EX47	19 × 27	2966

(1) Mesh description is $n \times m$ meaning n rows by m columns of nodes.
(2) Finite difference version of STAGS.

A linear bifurcation buckling analysis is performed prior to the nonlinear elastic collapse analysis. The buckling loads computed using Meshes 1, 2, and 3 are 1016 pounds, 1008 pounds and 1179 pounds, respectively, which correlate well with the results presented by Hartung and Ball[49]. The buckling mode shape indicates that the vertical edges of the cutout buckle locally.

The nonlinear analysis of the cylinder with cutouts is performed using the procedure NL_STATIC_1. Out-of-plane deflections w are shown in figure 27 as a function of the applied load for two points (denoted points "a" and "b") for the three meshes considered. The nonlinear response shown in figure 27 is characterized by large local radial deflections along the straight edges of the cutouts at a load level of approximately 1000 pounds followed by overall shell collapse at a load level of approximately 3000 pounds. The elastic collapse loads predicted using the Testbed are summarized in Table 9. These collapse loads are each nearly three times the linear bifurcation buckling load. As the out-of-plane deflections near the vertical edges of the cutouts develop, the compressive axial stresses are redistributed away from these regions and the load is carried by the remaining portions of the shell as shown in figure 28.

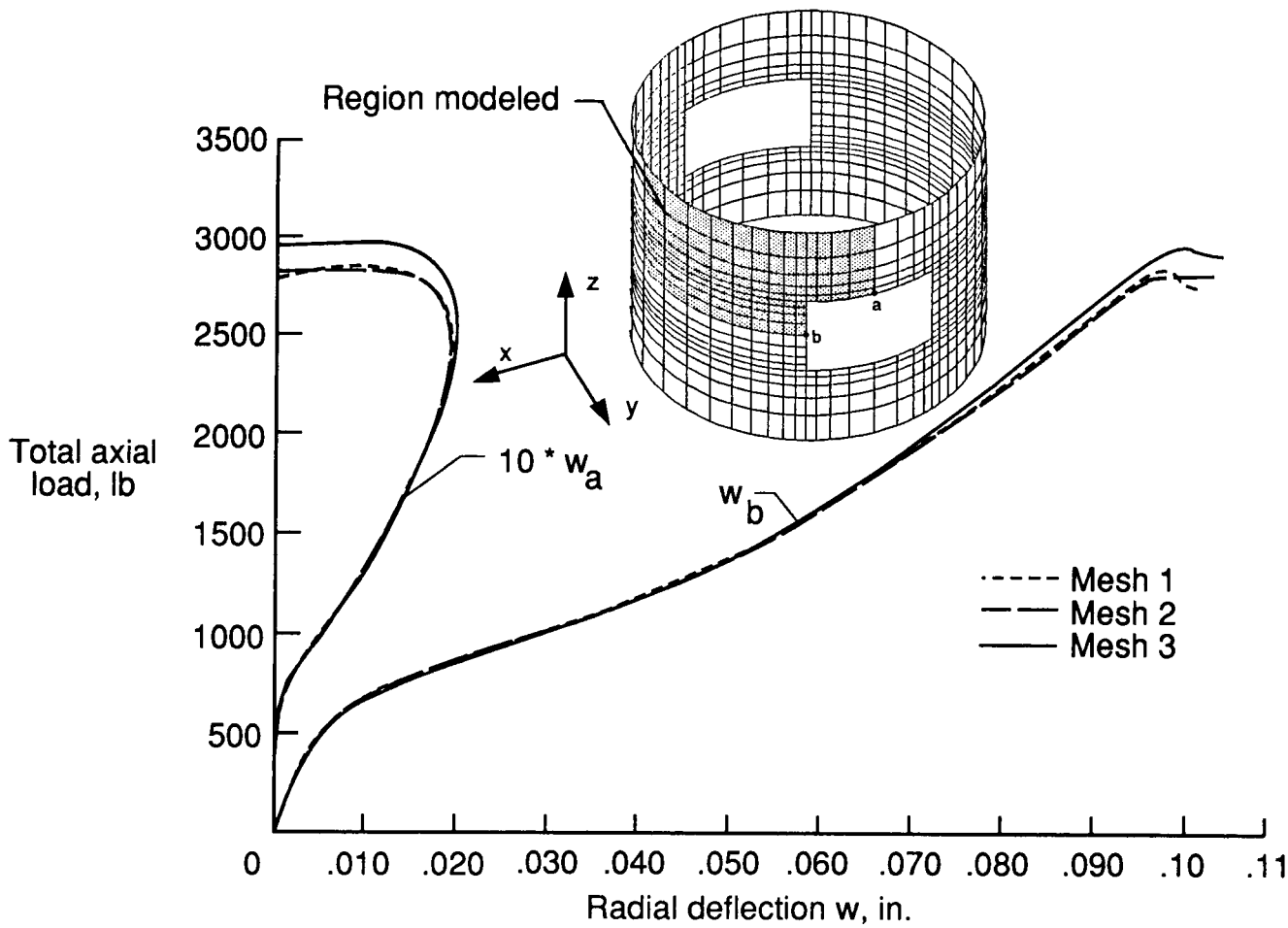
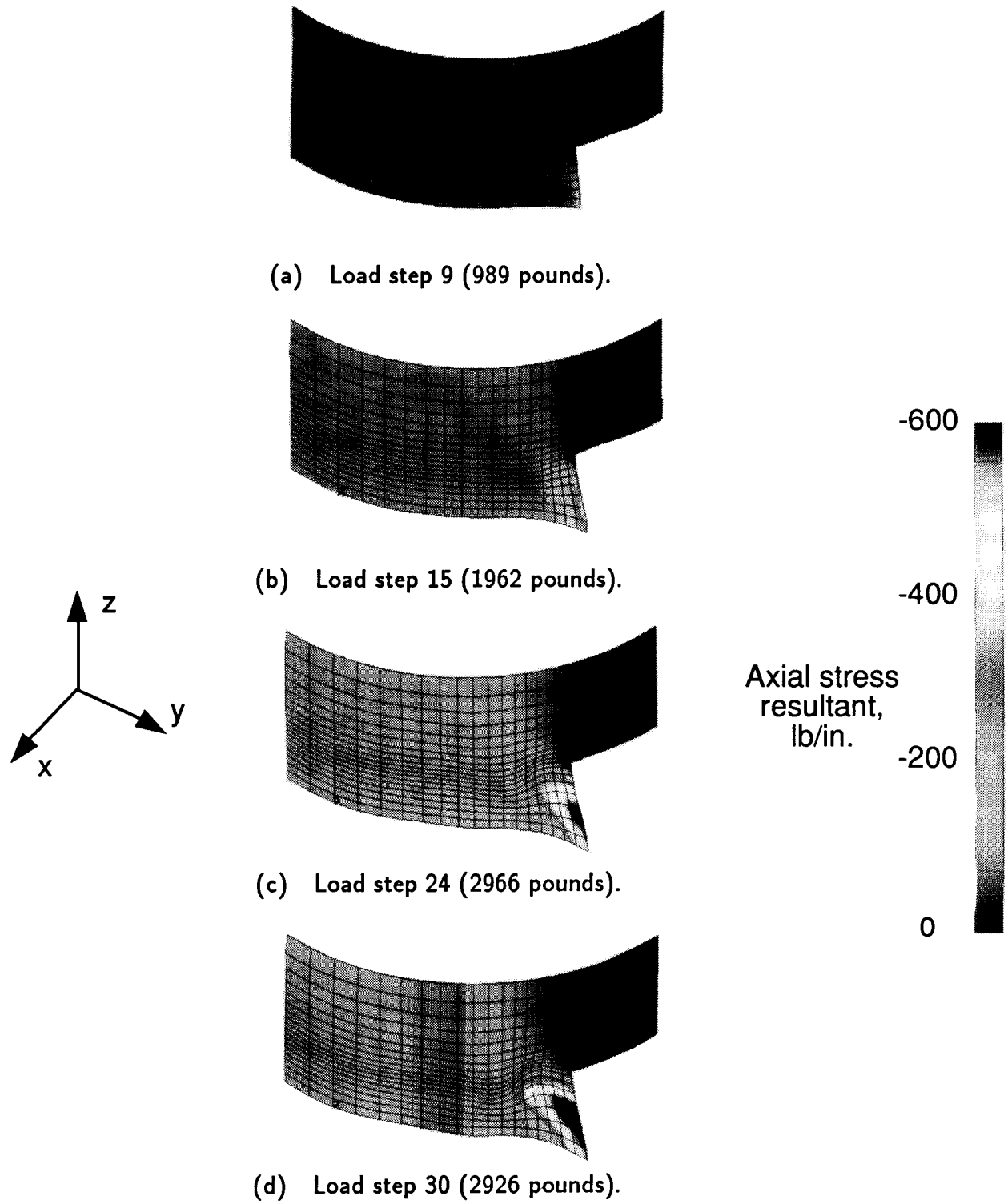


Fig. 27 Nonlinear response of cylinder with cutouts – Out-of-plane deflections.



(a) Load step 9 (989 pounds).

(b) Load step 15 (1962 pounds).

(c) Load step 24 (2966 pounds).

(d) Load step 30 (2926 pounds).

Fig. 28 Deformed geometry plots for several load steps – Mesh 3 results.
(see Appendix for color figures)

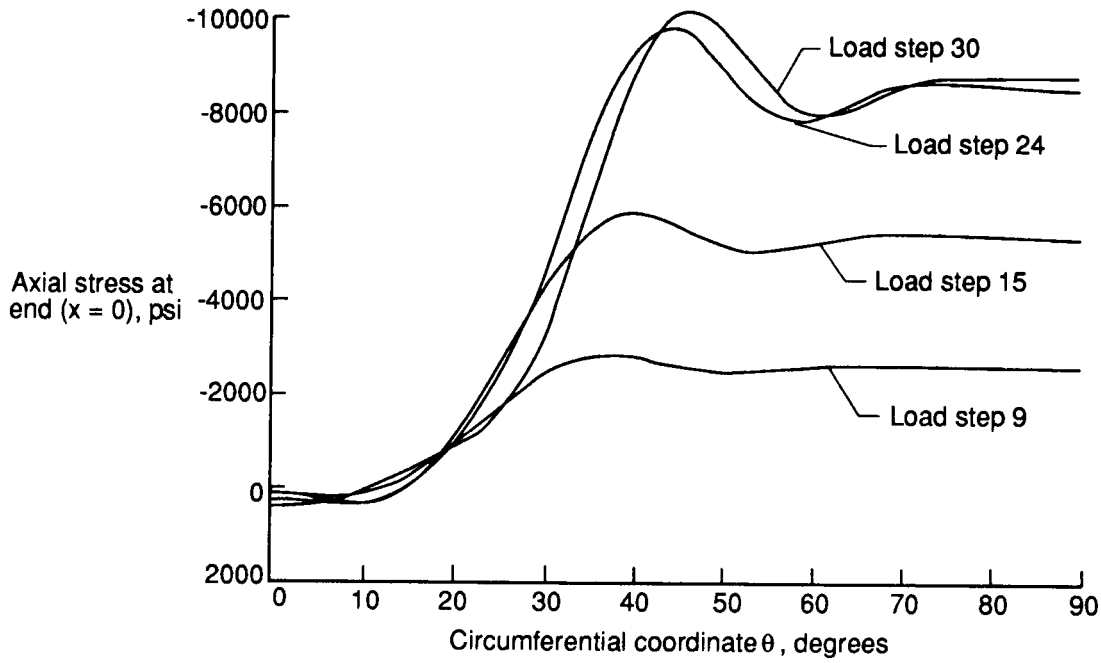


Fig. 29 Axial stress at $x = 0$ for various load steps – Mesh 3 results.

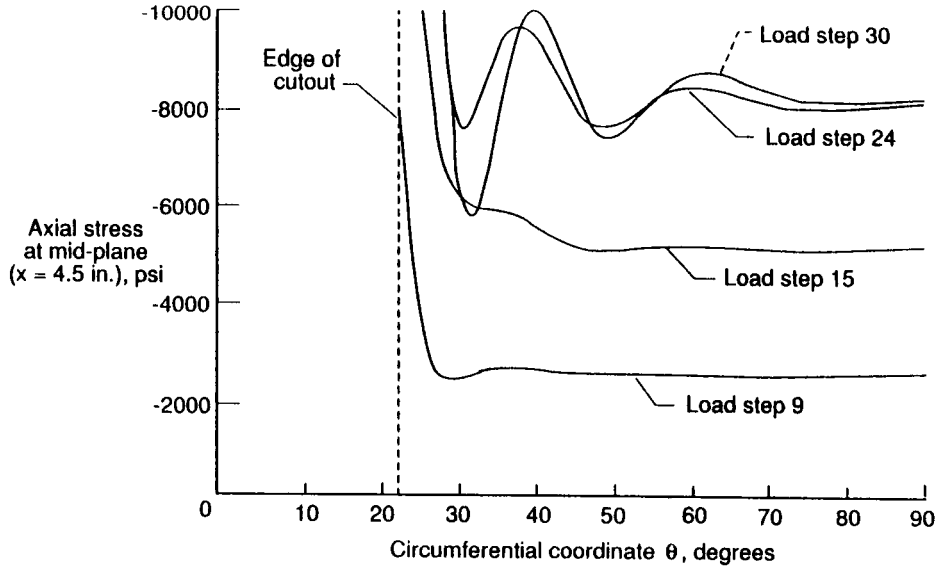


Fig. 30 Axial stress at $x = 4.5$ in. for various load steps – Mesh 3 results.

The distribution of the axial stress at the midsurface across the end of the cylinder ($x = 0$) is shown in figure 29 for various load levels. The distribution of the axial stress at the midsurface across the midplane of the cylinder ($x = 4.5$ inches) is shown in figure 30 for these same load levels. These distributions indicate that substantial redistribution of the axial stress occurs as the collapse load is approached.

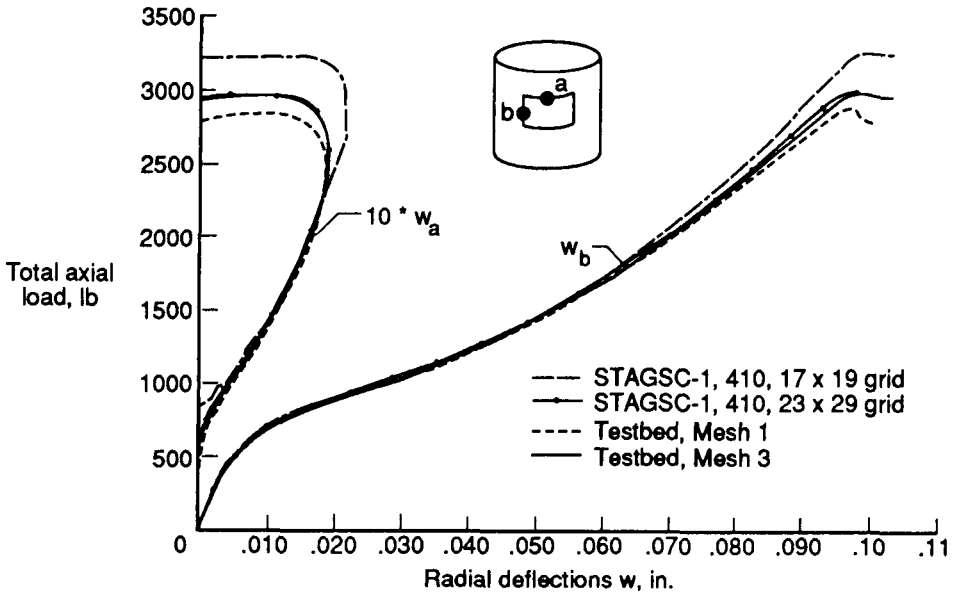


Fig. 31 Comparison of Testbed and STAGSC-1 results for the nonlinear response of cylinder with cutouts – Out-of-plane deflections.

Hartung and Ball[49] reported a collapse load of 2109 pounds using a finite difference version of the STAGS computer code. Later, Almroth and Brogan[50], in a convergence study using the finite element version of STAGS, reported a "nearly" converged collapse load of 2750 pounds. Independent elastic collapse analyses using the STAGSC-1 computer code were performed to compare with the nonlinear response predictions obtained using the CSM Testbed. These elastic collapse loads are summarized in Table 9. The nonlinear response shown in figure 31 indicate that the Testbed and STAGSC-1 analyses correlate well. The Testbed finite element models include transverse shear deformations and the ES1/EX97 elements are curved 9-node elements. The STAGSC-1 element is a 4-node flat, shell element which neglects transverse shear flexibility. These results indicate that the influence of transverse shear deformations on the nonlinear shell response is very slight. However, approximating the shell geometry as a curved surface rather than a facetted surface lowers the nonlinear collapse by approximately five percent.

One additional analysis is performed for the cylinder with cutouts. This analysis used a half model of the shell as shown in figure 32. This finite element model is composed of 1616 4-node quadrilateral shell elements (ES1/EX47), 1736 nodes and 8010 active degrees-of-freedom. The elastic collapse load for the half model is 2943 pounds compared to 2966 pounds for the eighth model (Mesh 3 in figure 26c). The nonlinear shell response for the half model is compared with the response from the eighth model in figure 33.

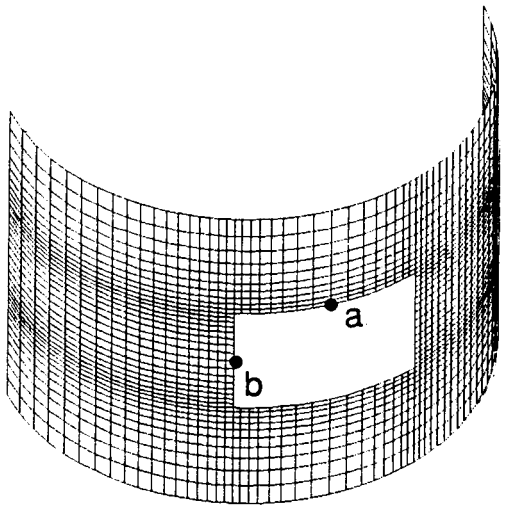


Fig. 32 Half model of cylinder with cutouts using 4-node elements. (1616 ES1/EX97 elements, 1736 nodes).

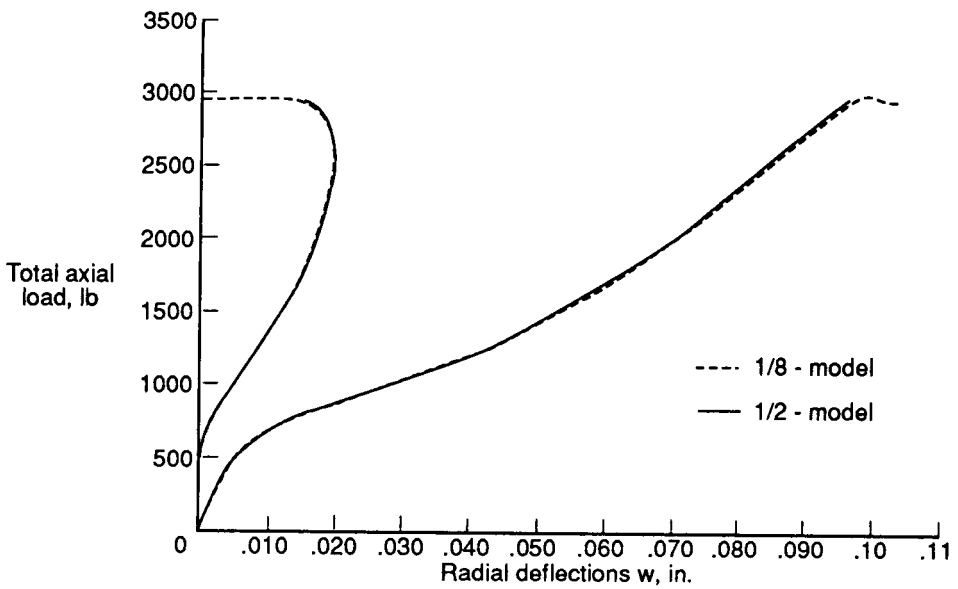
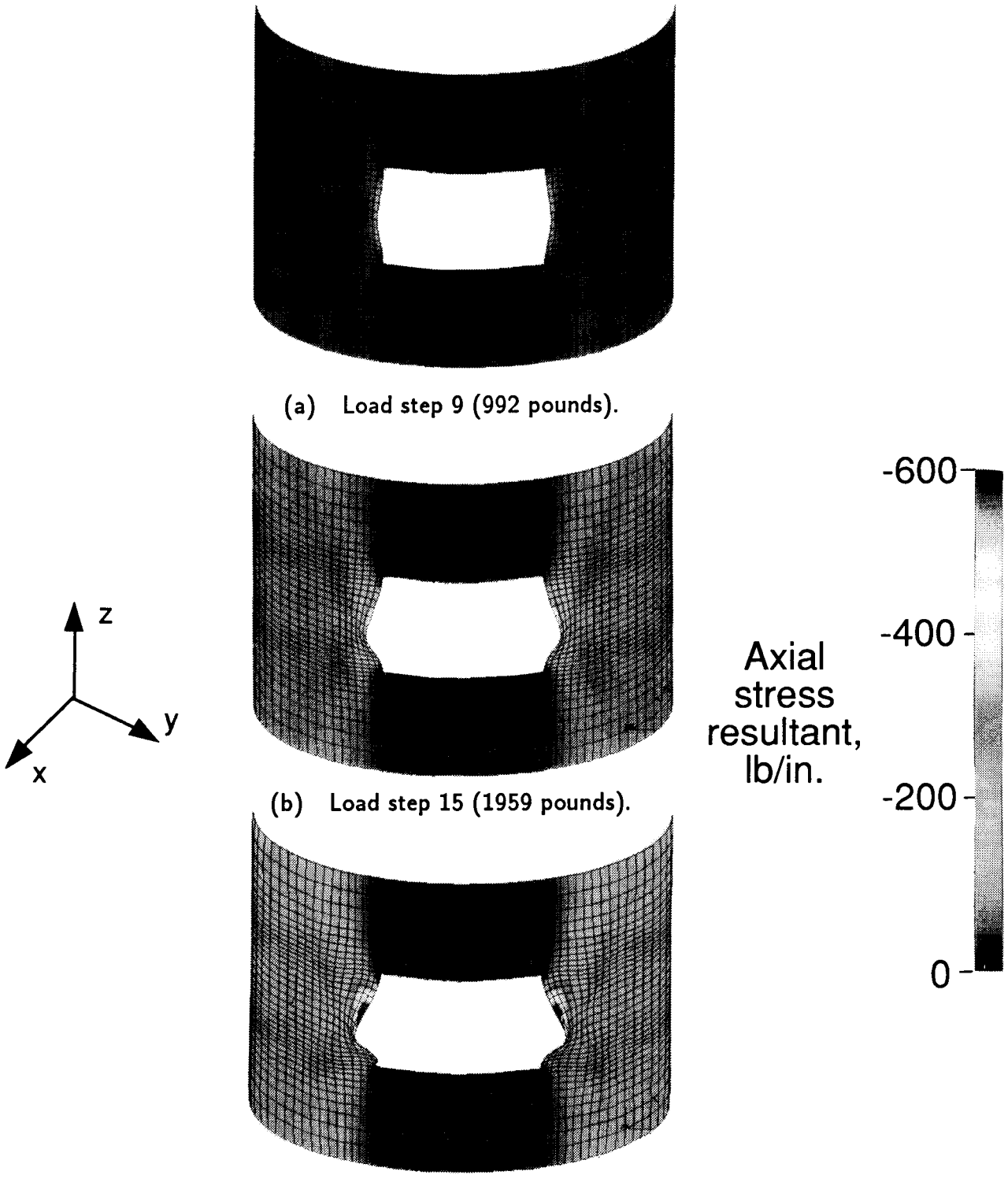


Fig. 33 Comparison of eighth-model and half-model results for nonlinear response of cylinder with cutouts – Out-of-plane deflections.

Contour plots of the axial stress resultant distributions for several load steps superposed on the corresponding deformed geometries with exaggerated deflections are shown in figure 34. Note that complete symmetry in the half-model response is exhibited up to the collapse load.



(a) Load step 9 (992 pounds).

(b) Load step 15 (1959 pounds).

(c) Load step 25 (2943 pounds).

Fig. 34 Deformed geometry plots for several load steps – Half-model results.
(see Appendix for color figures)

Table 10. Selected processor execution times for cylinder with cutouts (Mesh 1: ES1/EX97 – 449 nodes, 2012 *dof*, average semi-bandwidth of 148).

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	1.00	5.10
	E	0.30	5.30
	TOPO	1.70	14.20
Form and Factor Global Stiffness Matrix	ES	60.19	971.58
	K	0.94	17.10
	INV	9.17	448.55
	VEC	12.01	46.82
	SSOL	3.14	75.74
Each Iteration	SSOL	0.80	19.37
	VEC	3.06	11.98
	ES	15.34	248.50

Table 11. Selected processor execution times for cylinder with cutouts (Mesh 3: ES1/EX47 – 449 nodes, 2012 *dof*, average semi-bandwidth of 93).

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	2.40	13.3
	E	0.60	8.6
	TOPO	1.50	11.6
Form and Factor Global Stiffness Matrix	ES	49.37	800.07
	K	1.14	16.67
	INV	5.00	221.98
	VEC	12.05	46.35
	SSOL	2.32	54.24
Each Iteration	SSOL	0.61	14.15
	VEC	3.14	12.09
	ES	12.88	208.72

Computation times for the nonlinear analysis of the cylinder with cutouts are given in Tables 10 and 11 for selected Testbed processors. Most of the CPU time was spent in processor ES which computes new elemental tangent stiffness matrices. The global tangent stiffness matrix was re-evaluated and factored 23 times for Mesh 1 (9-node elements) and 25 times for Mesh 3 (4-node elements). Mesh 1 required 102 iterations while Mesh 3 required 92 iterations to

predict the nonlinear structural response of this cylinder. On the NAS Cray-2 computer system, processor INV ran at a compute rate of 8.3 MFLOPS for Mesh 1 and 7.9 MFLOPS for Mesh 3. On the VAX/VMS computer system processor INV ran at a compute rate of 0.17 MFLOPS for Mesh 1 and 0.18 MFLOPS for Mesh 3. The ratio of the overall execution time on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system to obtain the nonlinear response of the cylinder is 15.4 to 1 for Mesh 1 and 16.2 to 1 for Mesh 3.

Pear-Shaped Cylinder

The nonlinear shell response of cylindrical shells with a noncircular cross-section became the subject of intense research in the early 1970's. Early Space Shuttle fuselage configurations were noncircular, and the "pear-shaped" cross-section was a leading candidate. The pear-shaped cylinder shown in figure 35 has been adopted by many researchers and the behavior of this shell subject to a uniform end-shortening investigated (*e.g.*, refs. 49 and 50). The results in this report are compared with results reported by Hartung and Ball[49] and by Almroth and Brogan[50]. In all cases, only one fourth of the cylinder is modeled. The shell is isotropic with a uniform thickness of 0.01 inches. The boundaries are simply supported.

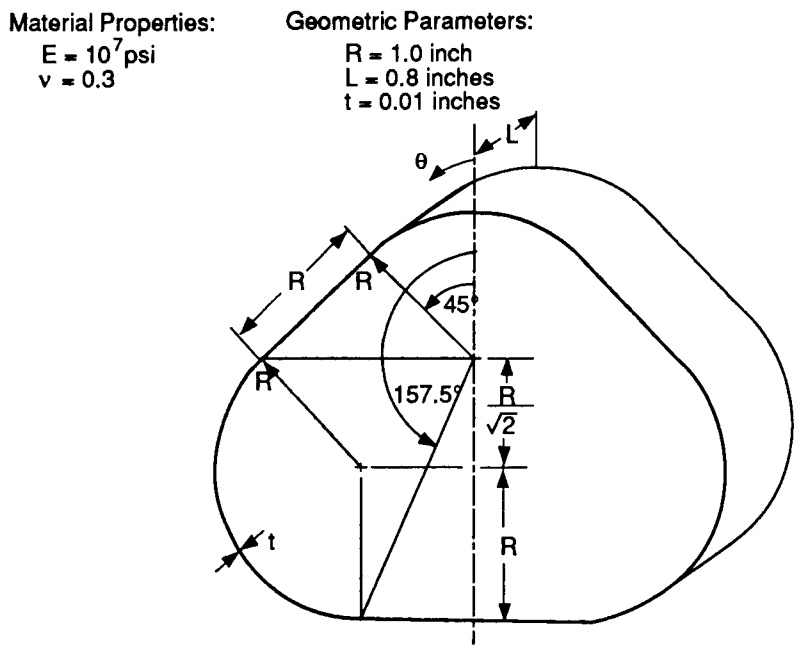
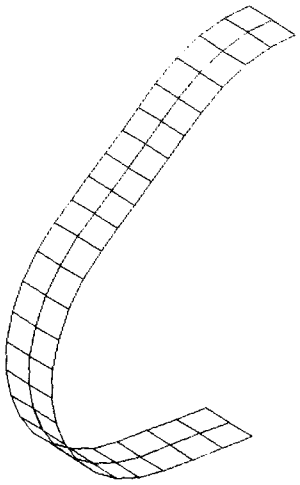


Fig. 35 Pear-shaped cylinder – geometry, properties, and loading.

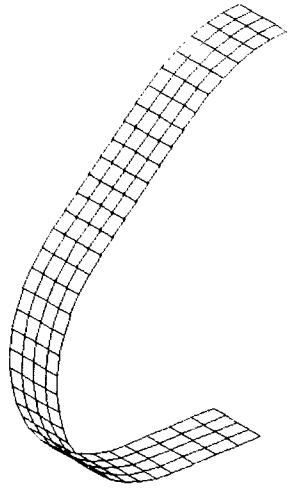
Three finite element models were considered in this study as shown in figure 36. Mesh 1 (3 $\times$ 26) has three nodes through the cylinder depth and 26 nodes around the half-cylinder circumference (see figure 36a). The mesh is composed of 50 4-node quadrilateral shell elements (ES1/EX47 or ES5/E410) and a total of 78 nodes. Mesh 2 (5 $\times$ 37) has five nodes through the cylinder depth and 37 nodes around the half-cylinder circumference (see figure 36b). This mesh accommodates both 4- and 9-node elements and will contain either 144 4-node (ES1/EX47 or ES5/E410) or 36 9-node (ES1/ES97) quadrilateral shell elements and 185 nodes. Mesh 3 (7 $\times$ 51) has seven nodes through the cylinder depth and 51 nodes around the half-cylinder circumference (see figure 36c). Mesh 3 accommodates both 4- and 9-node elements and will contain either 300 4-node (ES1/EX47 or ES5/E410) or 75 9-node (ES1/EX97) quadrilateral shell elements and 357 nodes.

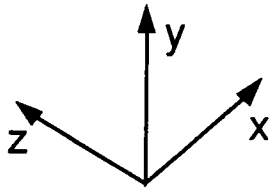
The pear-shaped cylinder is representative of a shell-type structure with a complex nonlinear collapse behavior. The shell response becomes nonlinear at a very low value of applied end-shortening, and the normal deflections of the flat portions of the shell increase rapidly. The nonlinear analysis of the cylinder is performed using the procedure NL_STATIC_1. The distribution of the normal displacement (normal to the shell surface) is shown in figure 37 as a function of the applied load for the point of maximum normal displacement (at x =0, θ =180°) and for all meshes and element types considered. These results indicate that a converged solution is provided by the ES1/EX97 elements as the curves for Mesh 2 and Mesh 3 in figure 37c are nearly identical. These three solutions correspond to a discretization with 185 nodes with either 144 4-node flat classical (ES5/E410) elements, 144 4-node flat shear-flexible (ES1/EX47) elements, or 36 9-node curved shear-flexible (ES1/EX97) elements.

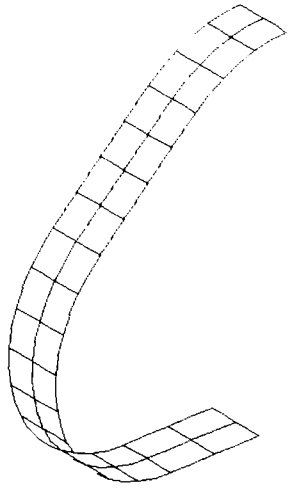
A comparison of the three solutions obtained using Mesh 2 is provided in figure 38. The effects of transverse shear flexibility, present in the ES1/EX97 and ES1/EX47 elements but not present in ES5/E410 elements, are apparent near collapse. The response curves begin to separate slightly at approximately half the elastic collapse load with the shear flexible elements exhibiting a consistent, yet slightly lower, stiffness than the response obtained using the ES5/E410 element. The elastic collapse loads obtained using the 4-node flat elements is nearly the same. However, results obtained using higher-order curved elements to model the shell and its response are approximately 10% lower than the collapse loads obtained using the flat elements.



(a) Mesh 1.



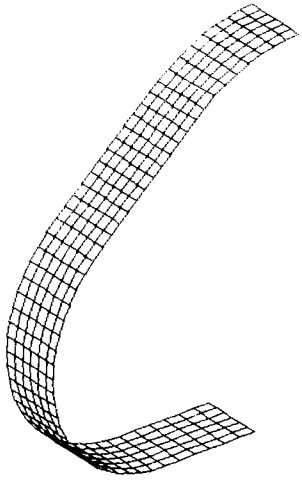


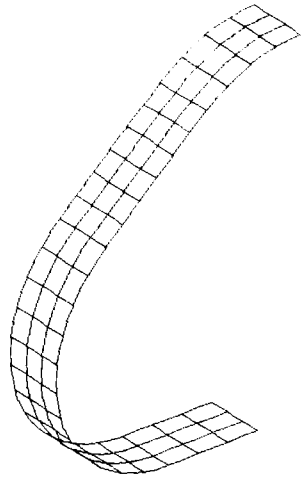


4-node elements

9-node elements

(b) Mesh 2.



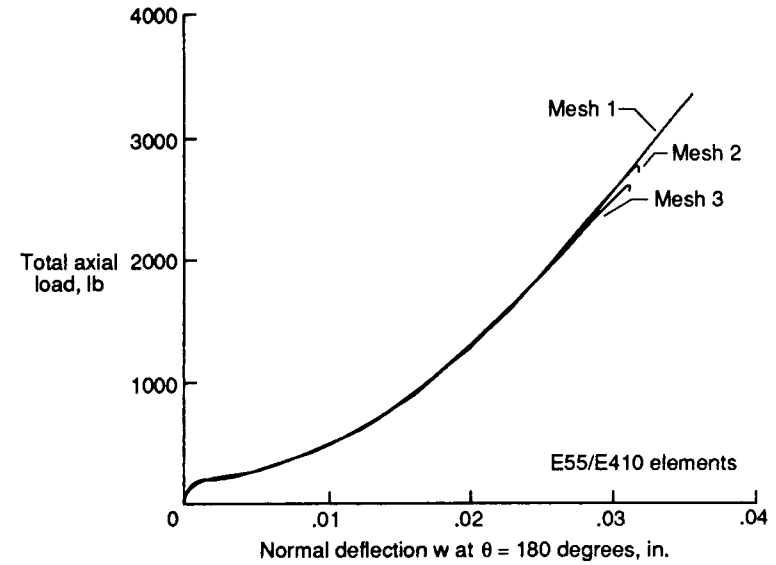


4-node elements

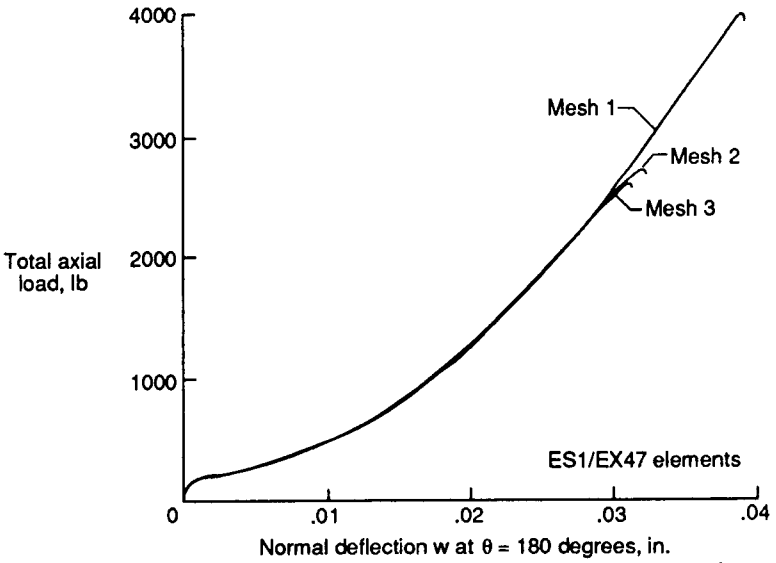
9-node elements

(c) Mesh 3.

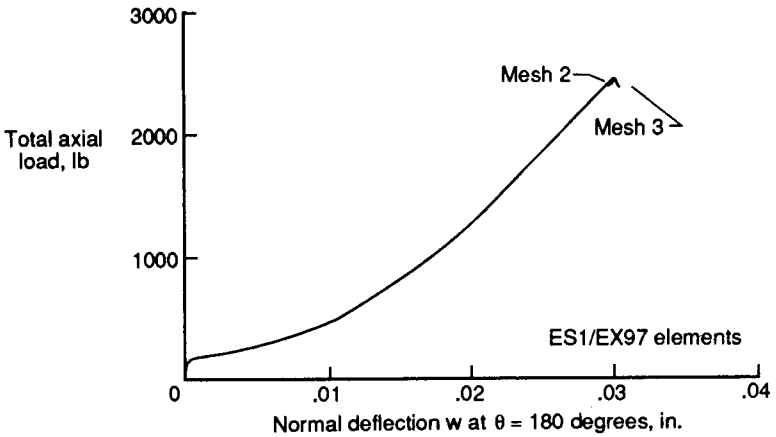
Fig. 36 Finite element models of pear-shaped cylinder.



(a) Models using 4-node flat classical (ES5/E410) elements.



(b) Models using 4-node flat shear-flexible (ES1/EX47) elements.



(c) Models using 9-node curved shear-flexible (ES1/EX97) elements.

Fig. 37 Nonlinear response of pear-shaped cylinder.

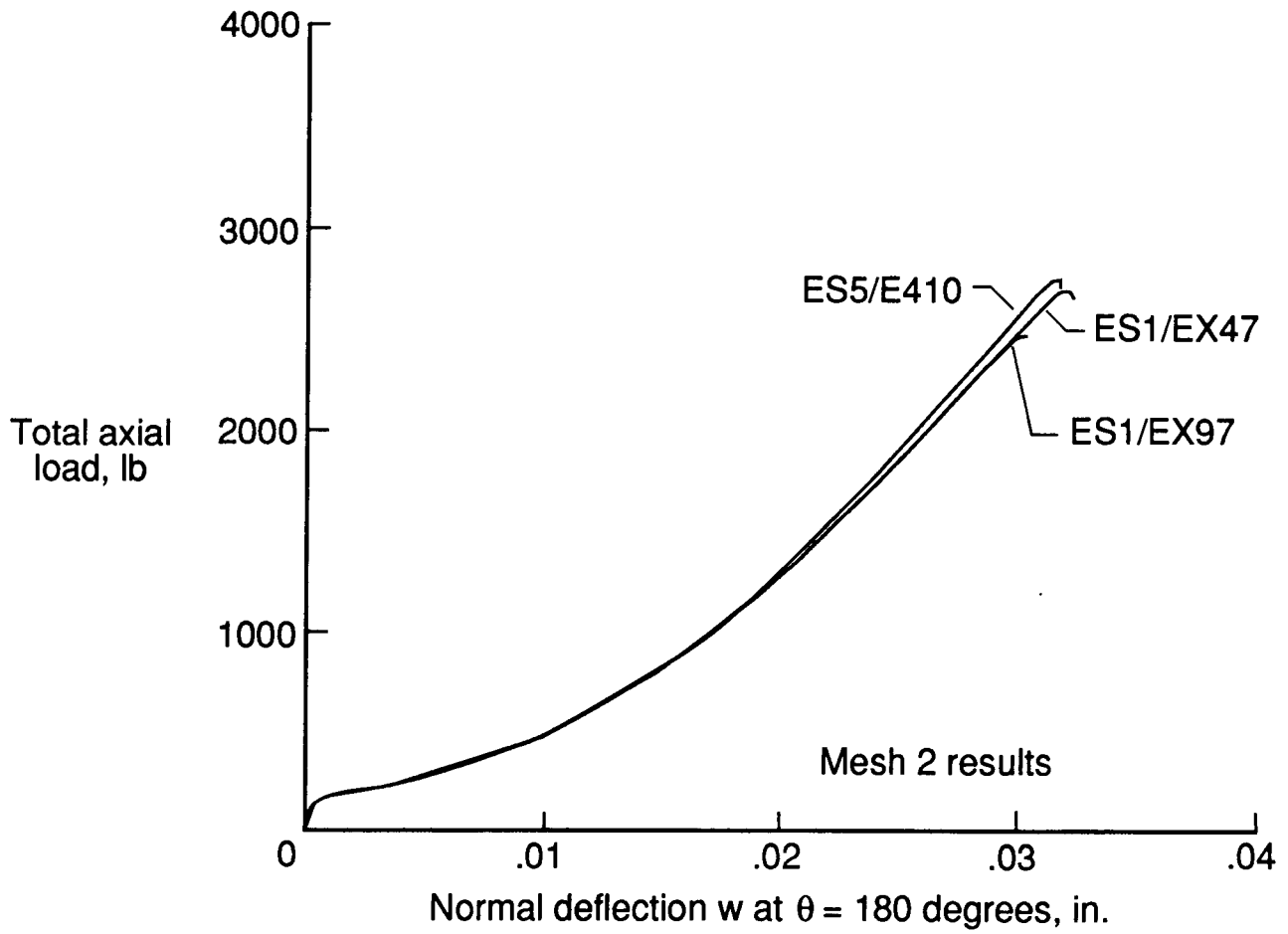


Fig. 38 Nonlinear response of pear-shaped cylinder – Mesh 2.

Hartung and Ball[49] reported an elastic collapse load of 2372 pounds using a finite difference version of STAGS. Several years later, Almroth and Brogan[50] performed a convergence study using the finite element version of STAGS and estimated that the elastic collapse load was between 2300 and 2400 pounds. These results, along with the Testbed results, and results from independent elastic collapse analyses using the STAGSC-1 computer code, are summarized in Table 12.

Table 12. Elastic collapse loads for pear-shaped cylinder.

Source	Element Name	Mesh(1)	Elastic Collapse Load, pounds
Hartung and Ball[47]	(2)	4 × 40	2372
Almroth and Brogan[48]	411	3 × 27	3586
		5 × 37	2731
		7 × 47	2586
	440	5 × 37	2657
		7 × 47	2530
STAGSC-1	410	3 × 26	3570
		5 × 37	2734
CSM Testbed	ES5/E410	3 × 26	3343
		5 × 37	2753
		7 × 51	2577
	ES1/EX97	5 × 37	2475
		7 × 51	2466
	ES1/EX47	3 × 26	3945
		5 × 37	2696
		7 × 51	2568

(1) Mesh description is $n \times m$ meaning n rows by m columns of nodes.
(2) Finite difference version of STAGS.

In figure 39, the normal deflection at $x = 0$ is plotted as a function of the circumferential coordinate θ for four different levels of applied load: 154, 300, 1689, and 2464 pounds (load steps 10, 20, 30 and 40 respectively). The final load step (step 40 at 2464 pounds) occurs just after collapse. The flat portions of the shell, from $45° \leq \theta \leq 90°$ and $157.5° \leq \theta \leq 180°$, show a rapid growth in normal deflections. Associated with this growth is a redistribution of the longitudinal stress indicating that the curved portions begin to take up a larger percentage of the total axial load. This type of behavior can be seen in figure 40 which plots the longitudinal stress resultant as a function of the circumferential coordinate for load steps 10, 20, 30 and 40. Contour plots of the longitudinal inplane stress resultant, superimposed on the deformed geometry with exaggerated deflections, are provided for the same four load steps in figure 41.

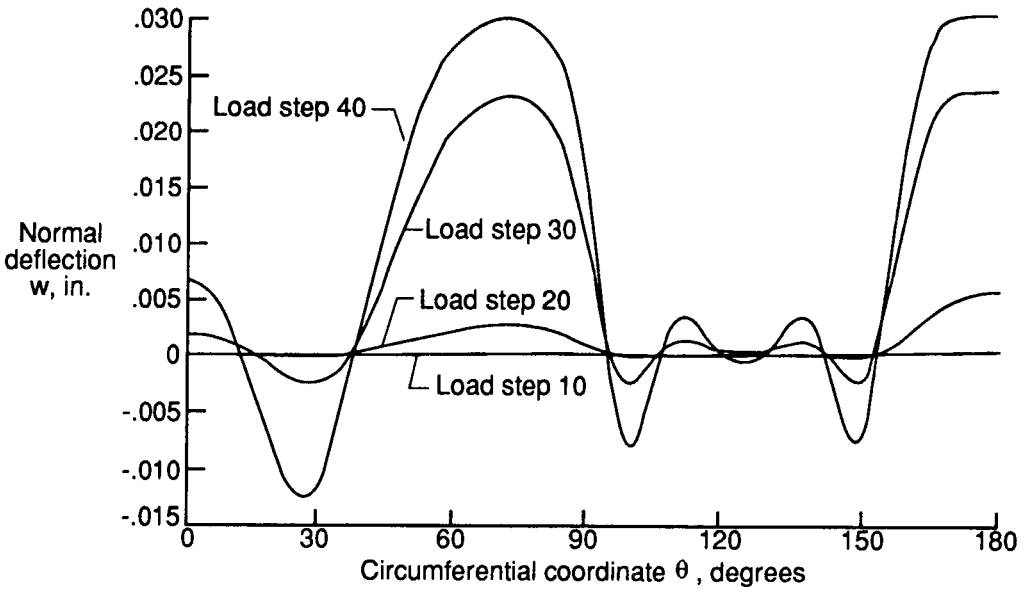


Fig. 39 Normal deflection distribution at cylinder midlength for various load steps.

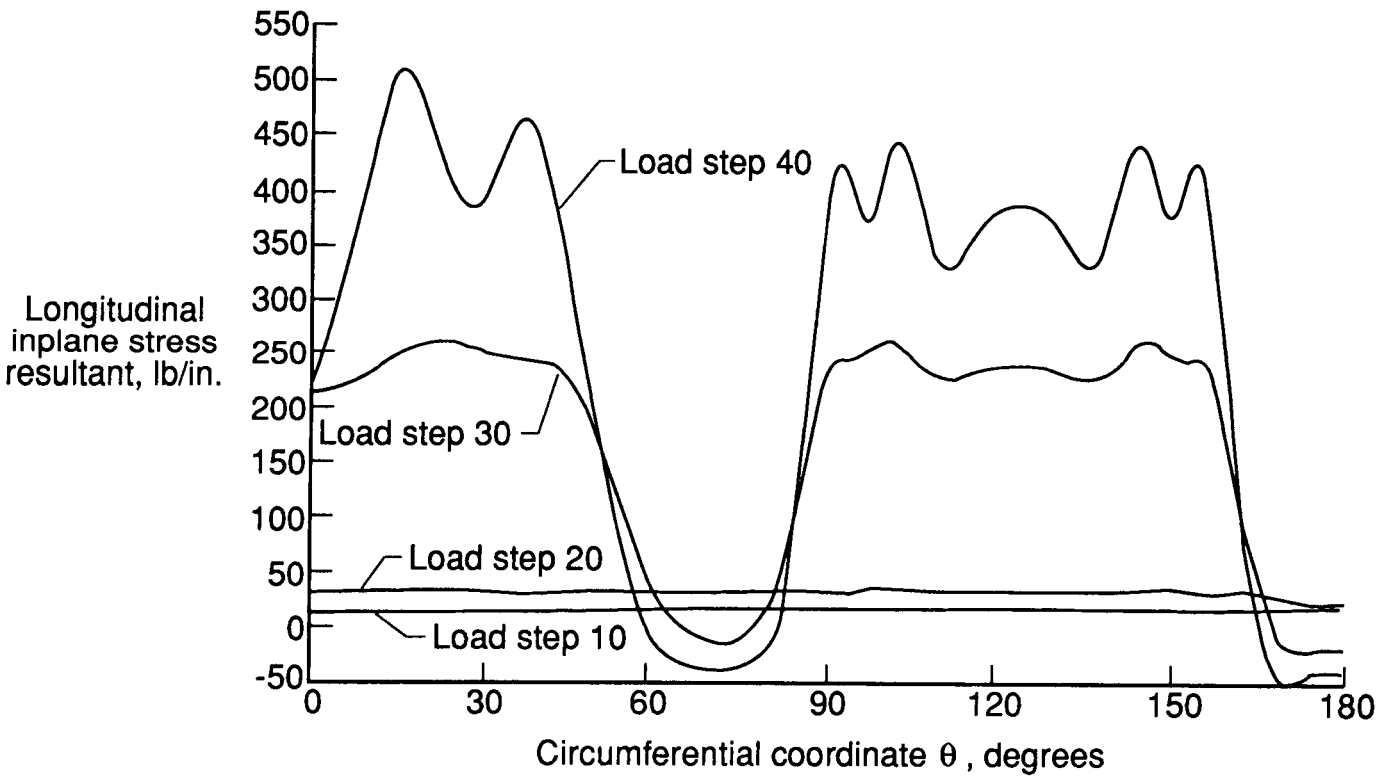
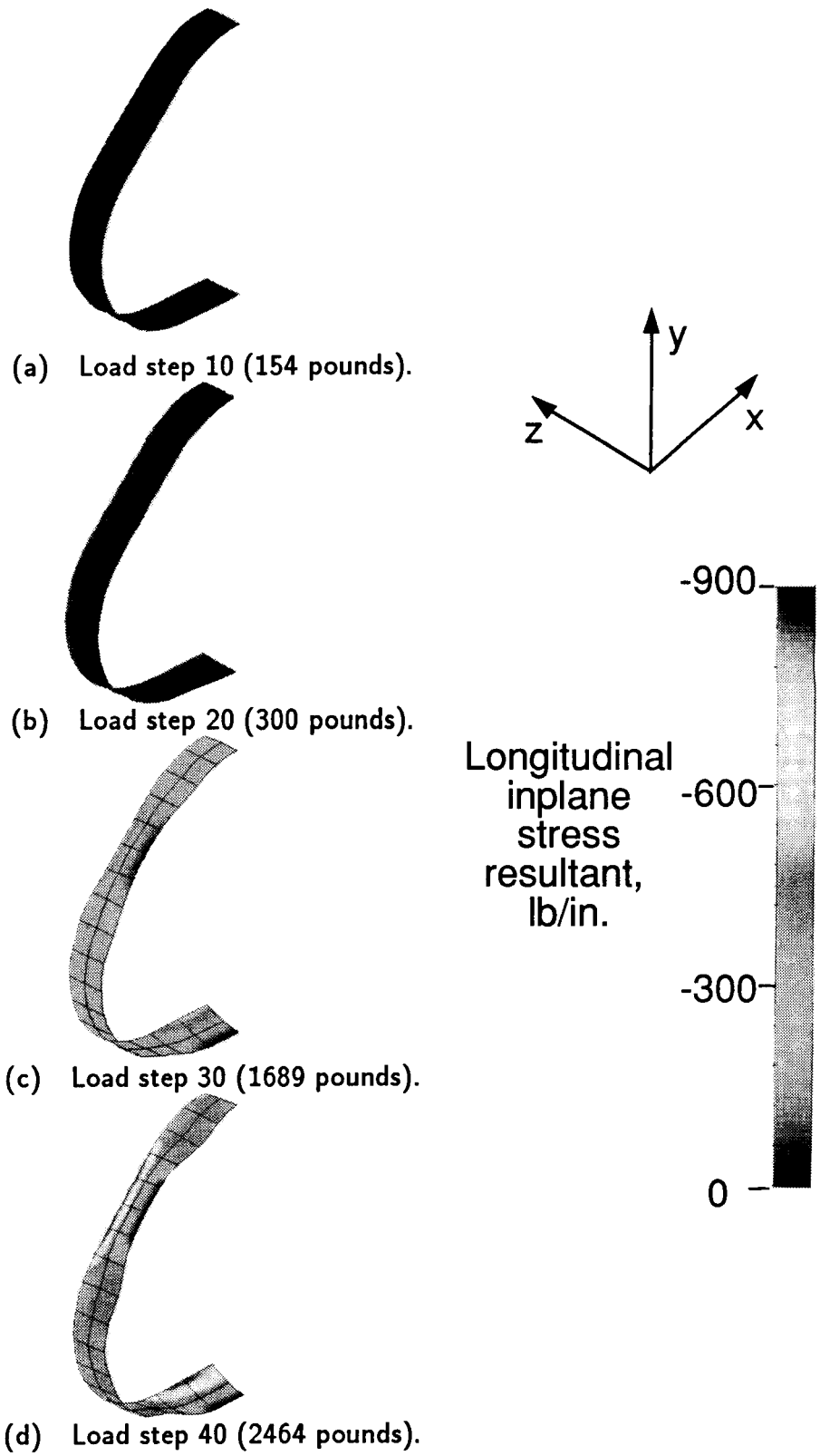


Fig. 40 Longitudinal inplane stress resultant N_x distribution at cylinder midlength for various load steps.



(a) Load step 10 (154 pounds).

(b) Load step 20 (300 pounds).

(c) Load step 30 (1689 pounds).

(d) Load step 40 (2464 pounds).

Fig. 41 Deformed geometry plots for several load steps.
(see Appendix for color figures)

Table 13. Selected processor execution times for pear-shaped cylinder (ES1/EX97, 185 nodes, 687 *dof*, average semi-bandwidth of 37.)

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	0.3	1.1
	E	0.2	1.9
	TOPO	0.6	4.3
Form and Factor Global Stiffness Matrix	ES	9.2	44.9
	K	0.4	7.4
	INV	0.5	19.7
	VEC	1.9	11.1
	SSOL	0.2	4.9
Each Iteration	SSOL	0.3	4.9
	VEC	1.8	4.0
	ES	3.4	46.1

This study appears to indicate that the curved element, ES1/EX97, performs much better than its flat 4-node counterparts. The wide range of values predicted for the collapse load suggests that addition research in shell element technology and in nonlinear solution strategies is required in order to provide reliable analysis tools.

Computation times for the nonlinear analysis of the isotropic pear-shaped cylinder (Mesh 2, ES1/EX97) are given in Table 13 for selected Testbed processors. Most of the CPU time was spent in processor ES which computes new elemental tangent stiffness matrices. The global tangent stiffness matrix was re-evaluated and factored 51 times and a total of 186 iterations were required to predict the nonlinear structural response of this cylindrical shell. The estimated number of floating point operations to factor the system matrix is 4,204,656. The compute rates for processor INV are 8.41 and 0.21 MFLOPS on the NAS CRAY-2 and VAX 11/785 computer system, respectively. The ratio of the overall execution time on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system to obtain the nonlinear response of the pear-shaped cylinder is 10.2 to 1.

Impulsively Loaded Truncated Conical Shell

A number of important engineering problems are associated with the prediction of the response of a shell to high-energy, short-duration dynamic loads. Examples include reentry

vehicles, space vehicles subjected to pyrotechnic separation loads, and vehicles subjected to blast or impulse environments (*e.g.*, water impact). Sometimes these high-energy loads only generate a rapidly varying linear elastic stress state, but in other cases the loads may be sufficiently high or of sufficient duration that the structural response is nonlinear.

The linear elastic transient response of a truncated conical shell subjected to an impulse load (initial velocity) shown in figure 42 is selected as representative of these transient dynamic shell analysis problems. This problem also has been used as a benchmark problem by Hartung and Ball[49]. The finite element model shown in figure 43 is composed of 540 4-node quadrilateral elements, 589 nodes, and 2569 active degrees-of-freedom. The predicted transient response shown in figure 44 for the normal deflections at two points on the shell correlates well with the results presented in reference 49. Both points are located at 6.5 inches from the clamped, small diameter edge; one at $\theta = 0°$ (point "a") and one at $\theta = 180°$ (point "b"). The transient response was calculated for 1500 microseconds using the Newmark method with a time step of 2 microseconds. Oblique views of the deformed shape with exaggerated deflections at various points in time are shown in figure 45. Contour plots of the hoop stress resultants are superposed on these deformed shapes.

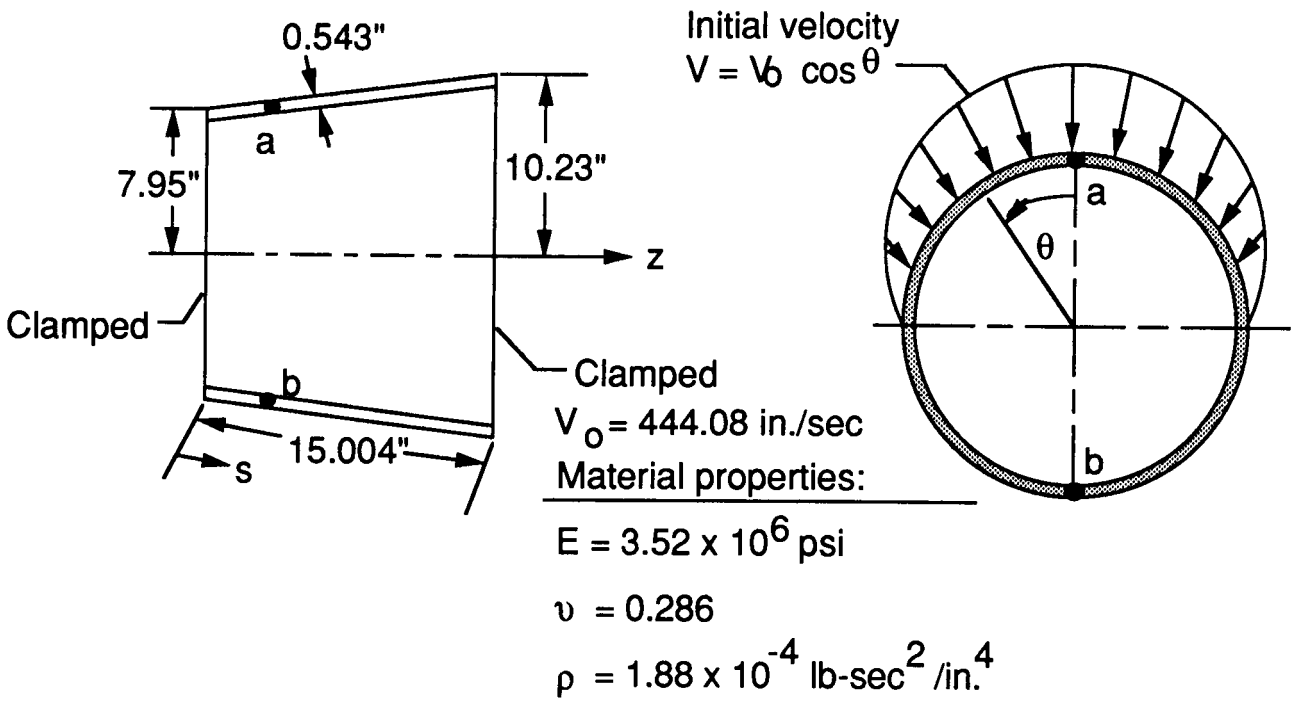


Fig. 42 Truncated conical shell – geometry, properties, and loading.

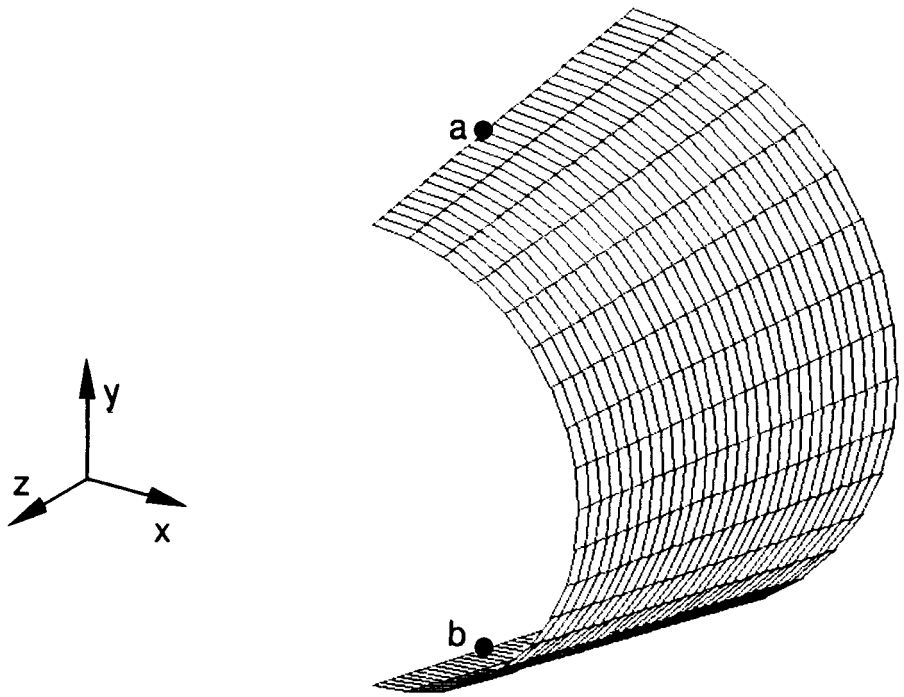


Fig. 43 Finite element model of truncated conical shell.

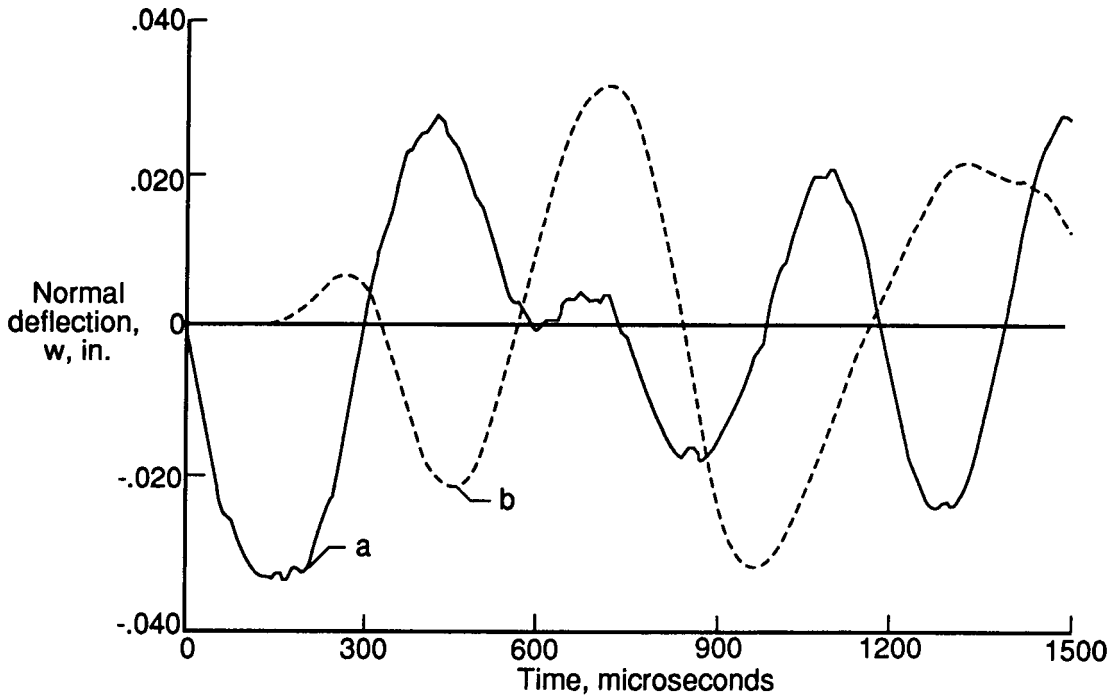
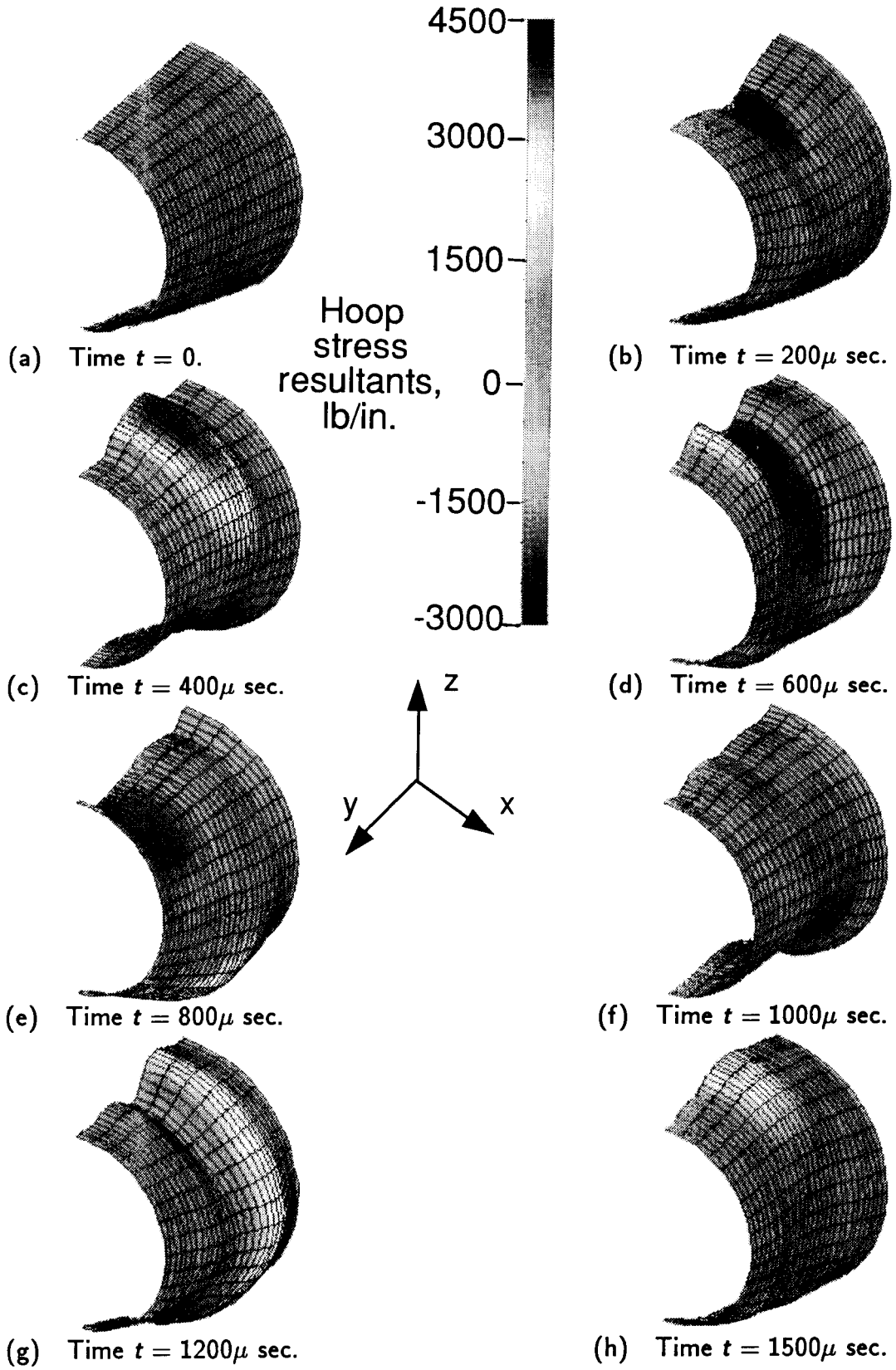


Fig. 44 Normal deflections at points "a" and "b" on truncated conical shell.



(a) Time $t = 0$.

(b) Time $t = 200\mu$ sec.

(c) Time $t = 400\mu$ sec.

(d) Time $t = 600\mu$ sec.

(e) Time $t = 800\mu$ sec.

(f) Time $t = 1000\mu$ sec.

(g) Time $t = 1200\mu$ sec.

(h) Time $t = 1500\mu$ sec.

Fig. 45 Deformed shapes for truncated conical shell during the transient response.
(see Appendix for color figures)

Table 14. Selected processor execution times for truncated conical shell (589 nodes, 2146 *dof*, average semi-bandwidth of 78).

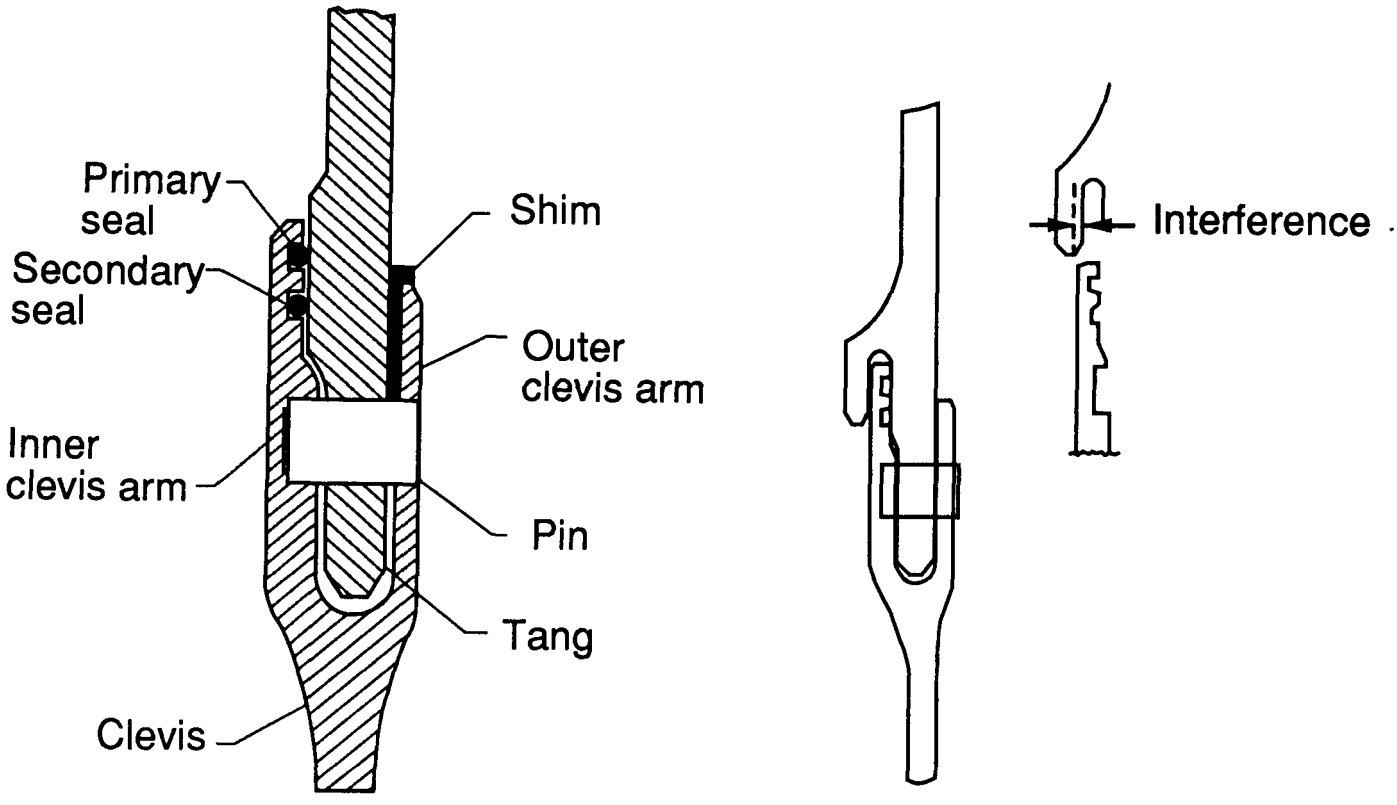
Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	1.1	9.0
	E	0.6	8.5
	TOPO	1.5	16.1
Form and Factor Global Mass and Stiffness Matrices	EKS	8.1	77.4
	K	1.5	21.1
	M	2.9	36.8
	INV	5.6	235.1
Each Time Step	AUS	0.5	5.4
	SSOL	0.7	12.0

Computation times for the nonlinear analysis of the truncated conical shell are given in Table 14 for selected Testbed processors. While each time step requires a relatively small amount of CPU time, 751 time steps are required in this analysis, thus the majority of time is spent in the Newmark algorithm. The estimated number of floating point operations in factoring the matrix is 32,493,250. The compute rates for processor INV are 5.4 MFLOPS on the CRAY-2 computer system and 0.14 MFLOPS on the VAX 11/785 computer system, respectively. The ratio of the overall execution time on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system is 14.5 to 1 for the complete transient response prediction of the truncated conical shell.

SRM Tang-Clevis Joint

The Space Shuttle Challenger accident investigation focused on the failure of a tang-clevis joint on the right Solid Rocket Motor (SRM). The existence of relative motion between the inner arm of the clevis and the O-ring sealing surface on the tang has been identified as a potential contributor to this failure. Finite element structural analyses were performed to predict both deflections and stresses in the joint under the primary, pressure loading condition (*e.g.*, see Greene, Knight, and Stockwell[51] and McConnaughey, Lee, and Moore[52]). These analyses have demonstrated the difficulty of accurately predicting the structural behavior of the tang-clevis joint. Stresses in the vicinity of the connecting pins, obtained from elastic analyses, considerably exceed the material yield allowables indicating that inelastic analyses are probably

necessary. The original design of the joint has been modified to resolve issues raised in the failure investigation; namely, to control the relative motion between the inner clevis arm and the tang at the O-ring sealing surface. The modification, referred to as the "capture feature", uses additional material on the inside of the tang and an interference fit to restrict the motion of the inner clevis arm. A cross section of the original and modified tang-clevis joints are shown in figure 46. The upper end of the lower cylindrical, motor segment forms the clevis. The lower end of the upper cylindrical, motor segment forms the tang which mates with the lower clevis. Around the circumference of both tang and clevis ends are 180 holes into which one-inch-diameter connecting pins are inserted. Three of the pin holes on the tang end are used as alignment slots to facilitate assembly of the SRM segments. The seal between two motor segments is provided by two O-rings in the "inner arm" of the clevis. The O-rings are compressed by a flat sealing surface on the tang.



(a) Original design. (b) Modified design.

Fig. 46 SRM tang-clevis joints.

(a) Entire model. (b) Close-up of local joint.

Fig. 47 Finite element models of the modified SRM tang-clevis joint.

In order to model the details of the contact between tang, pin, and clevis and also to predict the general three-dimensional stress state in the joint, three-dimensional elastic, solid finite elements are selected for the analysis. The joint geometry is assumed to be identical at each of the 180 pin locations around the circumference of the case segment. The effects of the three alignment slots are ignored in the analysis reported in this paper. This assumption implies that the structural behavior will be symmetric about a plane through the shell axis and pin centerline, and a plane through the shell axis and the centerline between two pins. The finite element model is therefore restricted to a one-degree circumferential slice of the case segment as shown in figure 47.

The CLAMP language features, along with the capability to easily add new, user-supplied computational processors, are exploited in the study. The finite element models for each component use six- and eight-node elastic, solid elements. These elements are based on an assumed-stress hybrid formulation. The elements have elastic material properties for all analyses performed in this study. The finite element model of the redesigned SRM tang-clevis joint has 2477 joints, 148 6-node elements, 1256 8-node elements, and 178 contact points (89 nonlinear spring elements).

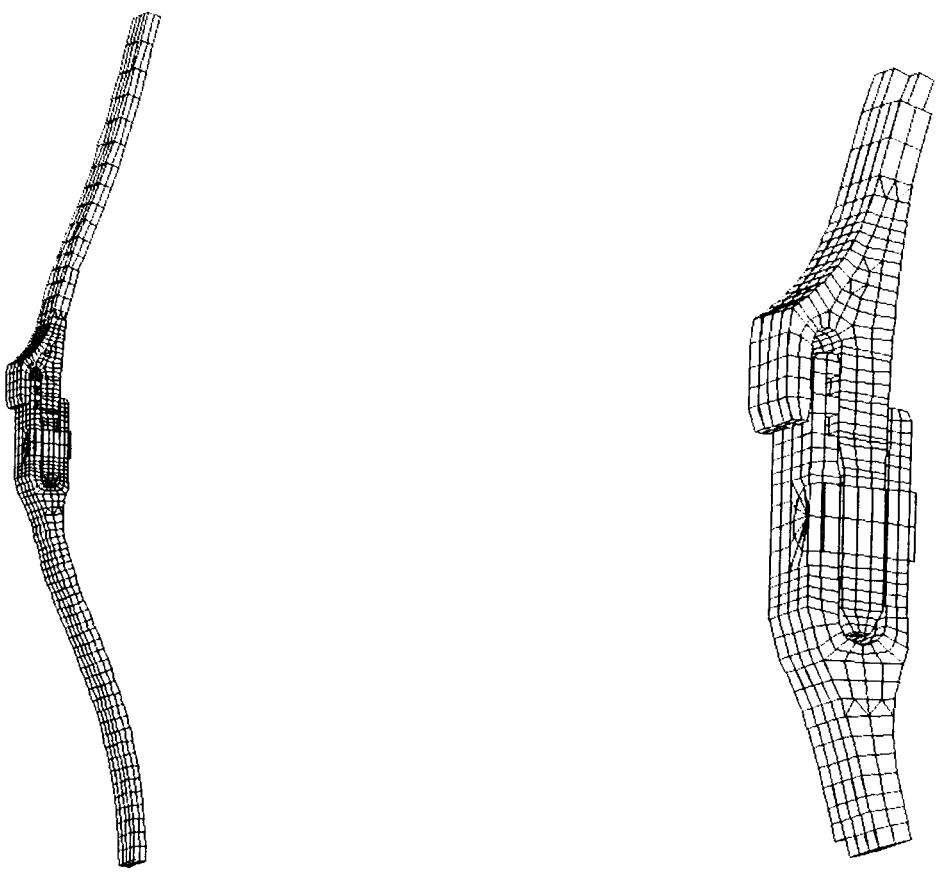
A key ingredient of the joint modeling approach is the method used for connecting the

separate finite element models of the tang, clevis, and pin. It was recognized from the outset that the contact regions between these components would change as a function of loading, leading to a nonlinear analysis problem. This nonlinearity occurs, for example, when a gap that exists between unloaded components closes when loaded. The actual contact is modeled by adding a nonlinear spring between two adjacent contact nodes (*e.g.*, adjacent nodes on the tang pin hole and the pin or adjacent nodes on the inner clevis arm and the tang). The nonlinear spring stiffness curves are piecewise-linear functions of the relative displacement and are generated, for example, such that a high stiffness results for any compression of the spring and a low or zero stiffness results if the spring stretches. To reduce the computational cost of the overall analysis, a substructure approach based on a "unit motion" solution technique is adopted (see reference 51). In this approach, all nonlinear behavior in the structure is assumed to be confined to specific contact nodes on the boundaries of the components. Therefore, a reduced, nonlinear problem can be formed which involves the equilibrium equations only at these contact nodes. For the finite element model shown in figure 47, 178 unit-motion vectors are calculated, and the reduced, nonlinear system has 178 unknowns. The reduced, nonlinear problem is solved using a full Newton-Raphson strategy.

Several characteristics of the original SRM joint design have been identified as potential contributors to the failure. One characteristic is the behavior of the joint under internal pressure load. The motor case expands radially outward due to the pressure. Because the joint has a higher hoop stiffness than the case wall on either side of the joint, its radial expansion is less than that of the case wall. This nonuniform radial expansion shown in figure 48 is the primary cause of relative motion between the inner clevis arm and the sealing surface on the tang. This relative motion can cause the O-rings to become unseated and therefore lose their sealing capability. Away from the tang-clevis joint, the structural response approaches that obtained using membrane shell theory.

The purpose of the analysis reported herein is to examine the sensitivity of the average gap motion for the modified tang-clevis joint with the capture feature to deviations from an interference fit. For new SRM cases, the nominal interference between the capture feature and the inner clevis arm is 0.019 inches. For reuse, the SRM cases are refurbished and proof-tested. A parametric study is performed to determine the structural response of the redesigned SRM joint to the initial clearance between the capture feature and the inner clevis arm. The effect of this clearance on the gap motion measured midway between the primary and secondary O-rings

is shown in figure 49. With a metal-to-metal fit, the average gap motion is approximately 0.004 inches. As the initial clearance becomes larger, the average gap motion increases. This trend continues until the initial clearance between the capture feature and the inner clevis arm is larger than the deflection of the inner clevis arm (*i.e.*, contact does not occur). For an initial clearance greater than approximately 0.050 inches, the average gap motion is approximately that of the original tang-clevis joint.



(a) Overall response. (b) Local joint response.

Fig. 48 Deformed geometry of the modified SRM tang-clevis joint.

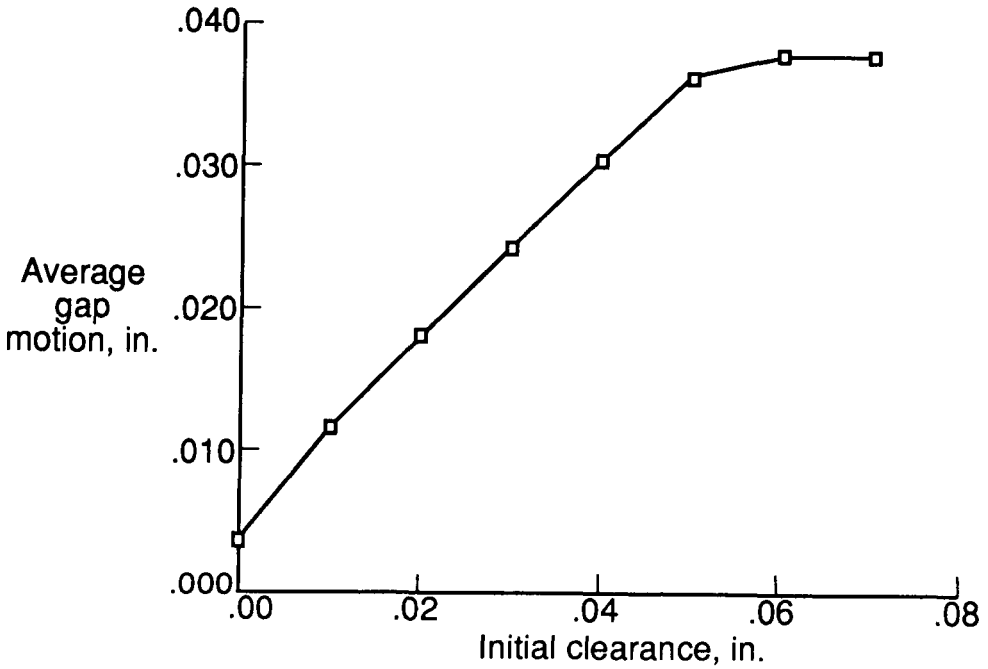


Fig. 49 Effect of initial clearance on average gap motion.

Computation times for the SRM tang-clevis joint analysis are given in Table 15 for selected Testbed processors. Most of the CPU time was spent in processor SSOL generating the unit-motion vectors needed at each contact point in order to generate the reduced nonlinear system. However, several other processors (EKS, INV, AUS) also used a sizeable amount of CPU time. All of these processors need to be studied and improved for large-scale analysis problems. The estimated number of floating point operations to factor the system matrix is 74,862,252. The compute rates for processor INV are 6.72 and 0.14 MFLOPS on the NAS CRAY-2 computer system and a VAX 11/785 computer system, respectively. The ratio of the overall execution time on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system is 13.7 to 1 for the eight initial clearance cases reported in figure 49. That is, the unit-motion vectors are formed once and the reduced, nonlinear system of equations is solved once for each initial clearance case considered.

Table 15. Selected processor execution times for SRM tang-clevis joint (2477 nodes, 6314 *dof*, average semi-bandwidth of 129.)

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	9.2	52.7
	E	3.4	45.1
	TOPO	11.5	119.0
Form and Factor Global Stiffness Matrix	EKS	20.5	249.5
	K	9.4	104.1
	INV	21.3	884.4
Generate Reduced Stiffness Matrices	SSOL	223.1	2712.2
	AUS	87.0	1692.6
Solve Reduced Nonlinear System	SSNA	28.8	661.0

SRB Global Shell Model

The basic elements of the Space Shuttle system are the Orbiter, the External Tank (ET), and the two reusable Solid Rocket Boosters (SRB's) as shown in figure 50. The SRB's provide the primary Shuttle ascent boost for the first two minutes of flight with an assist from the three Space Shuttle Main Engines (SSME's) on the Orbiter. The SRB structural subsystems are described in reference 53. A major subsystem of the SRB is the Solid Rocket Motor (SRM) which consists of four lined, insulated rocket motor segments. These segments are connected using pinned tang-clevis joints. Each SRB is approximately 144 feet long and 12 feet in diameter.

The modeling philosophy adopted for the SRB was substantially influenced by the size of the structure to be analyzed and the resulting number of the degrees-of-freedom in the equations to be solved. The underlying philosophy was to construct a finite element model that would accurately reflect the global load transfer in the SRB in a manner such that nonlinear shell collapse and shell ovalization under prelaunch loads could be assessed.

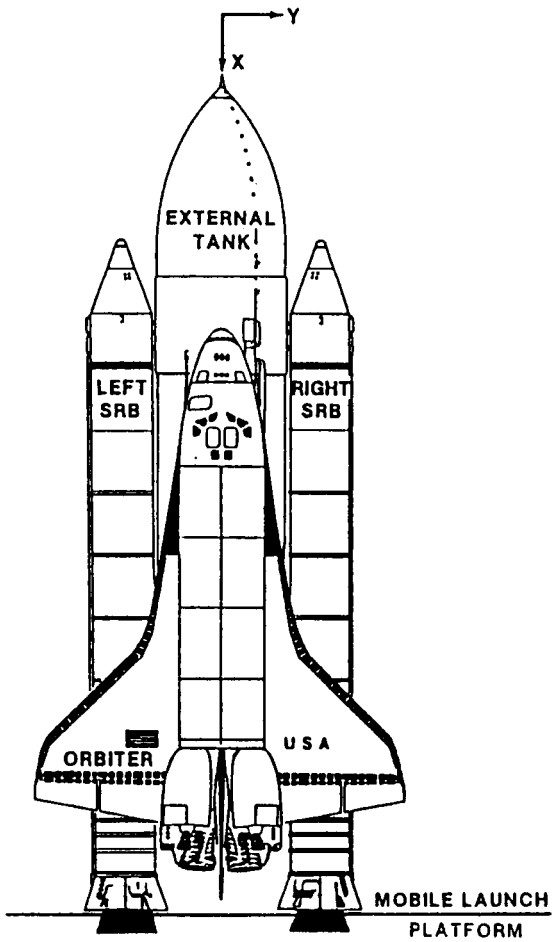


Fig. 50 Overview of Space Shuttle system.

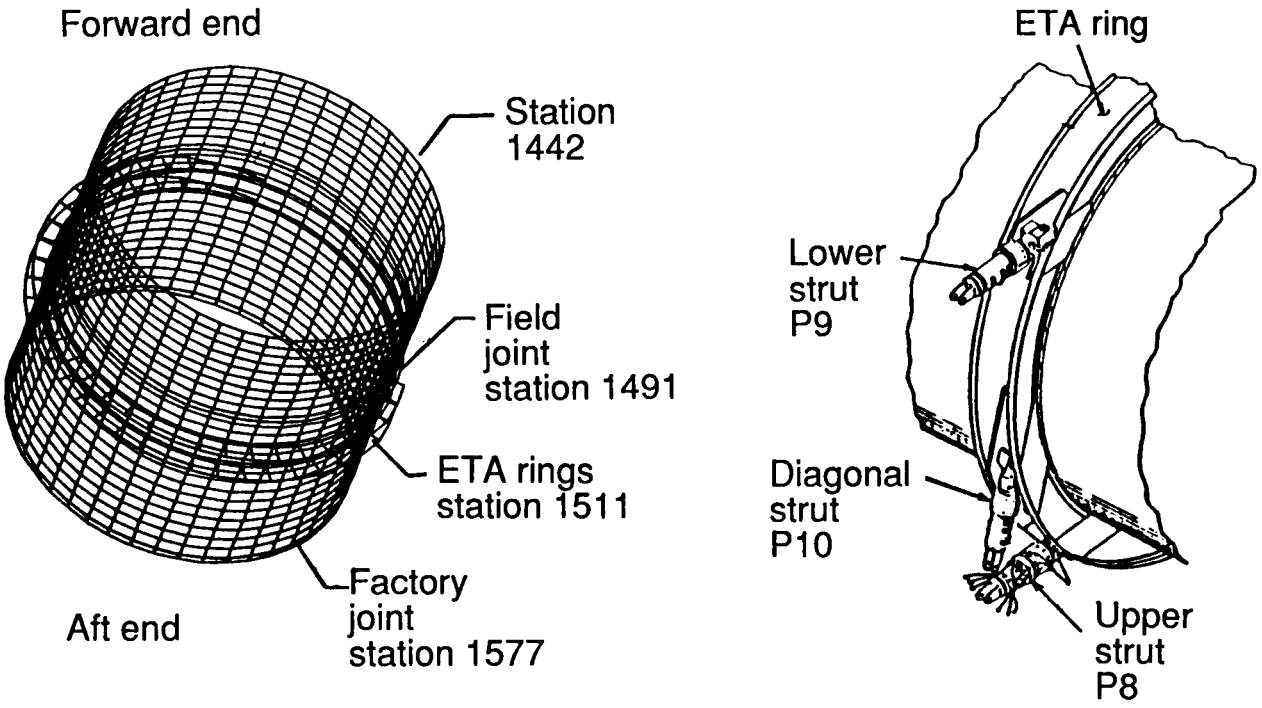


Fig. 51 Overview of SRB/ET attachment ring interface.

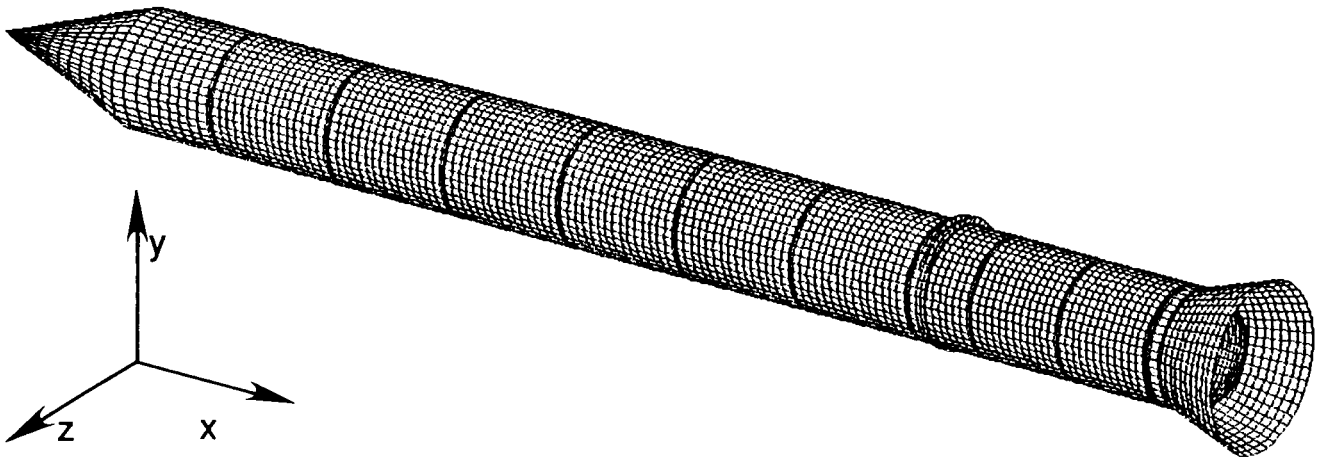
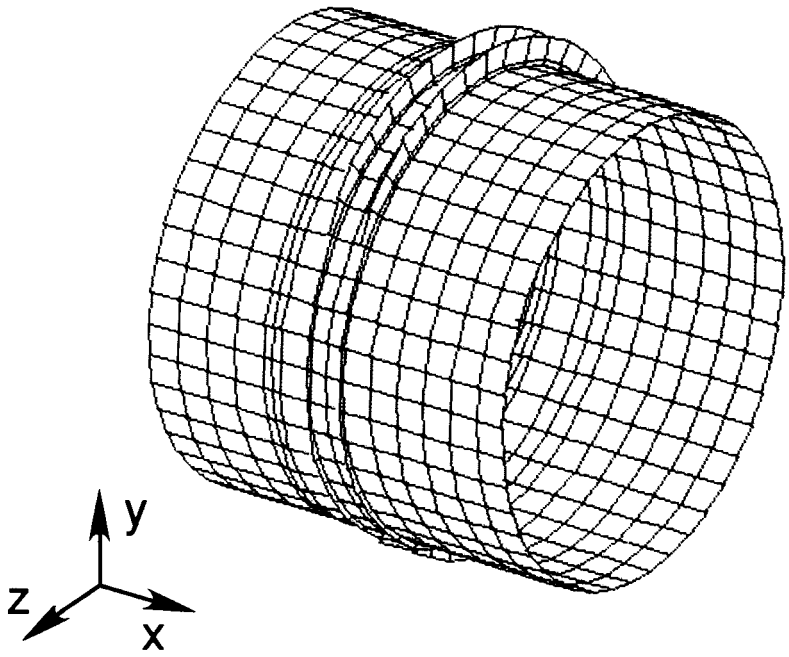
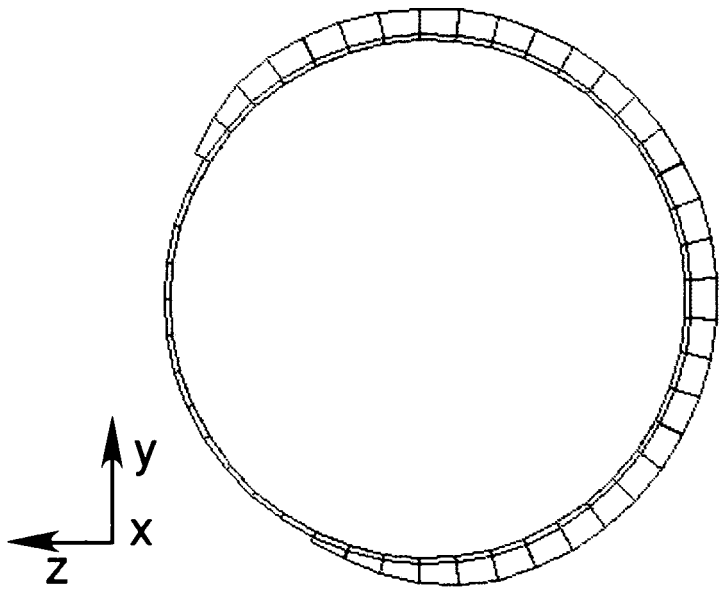


Fig. 52 Finite element model of SRB.



(a) Oblique view.



(b) Top view.

Fig. 53 Close-up view of finite element model of SRB/ET attachment ring interface.

The SRB/ET attachment (SRB/ETA) ring interface region shown in figure 51 includes both of the ETA rings (ring webs are approximately 12 inches apart), a portion of the SRM aft attachment segment including the factory joint at station 1577 (approximately sixty inches of shell), and a portion of the aft center segment including the field joint at station 1491 (approximately 64 inches of shell). The center of the SRB/ETA ring interface region is located at station 1511, approximately twenty inches below the aft attachment segment field joint. The ETA ring assembly is comprised of two tapered, partial rings, H-fittings to attach the ET struts, cover plates, and various other intercostals and brackets. The ETA ring assembly extends only 270-degrees circumferentially around the SRM segment. Three struts attach the aft ends of the SRB and the ET. These three attachment struts are designated the lower strut (P9), the diagonal strut (P10), and the upper strut (P8).

A preliminary assessment of the SRB global shell response to selected prelaunch loads has been performed Knight, Gillian and Nemeth[54] using the STAGSC-1 computer code[27]. The STAGSC-1 two-dimensional shell finite element model was translated into a format compatible with the CSM Testbed. This finite element model shown in figure 52 involves 9205 nodes with 1273 two-node beam elements, 90 three-node triangular elements, and 9156 four-node quadrilateral elements. A close-up view of the finite element model of the SRB/ET attachment ring interface is shown in figure 53. Although the resulting finite element model involves nearly 54,870 degrees-of-freedom, it does not have the fidelity necessary to determine detailed stress distributions in particular SRB subsystems. In this global shell model, the field and factory joints are modeled by using equivalent stiffness joints instead of detailed models of the joint. As such, local joint behavior cannot be obtained from this global model. Additional details of the finite element modeling for the entire SRB shell structure are described in reference 54.

The linear stress analysis considered herein involves only the loading case of a uniform SRM internal pressure of 1000 psi. An oblique view of the deformed geometry with exaggerated deflections is shown in figure 54. The deflection pattern exhibits a "pressure pillowing" behavior in the vicinity of the joints. The influence of the partial (270-degree) SRB/ET attachment ring on the SRB shell response is shown in figure 55. An abrupt change in the deflection pattern near the ends of the ET attachment ring can be seen. Further studies on the SRB/ET attachment ring are reported by Knight[55].

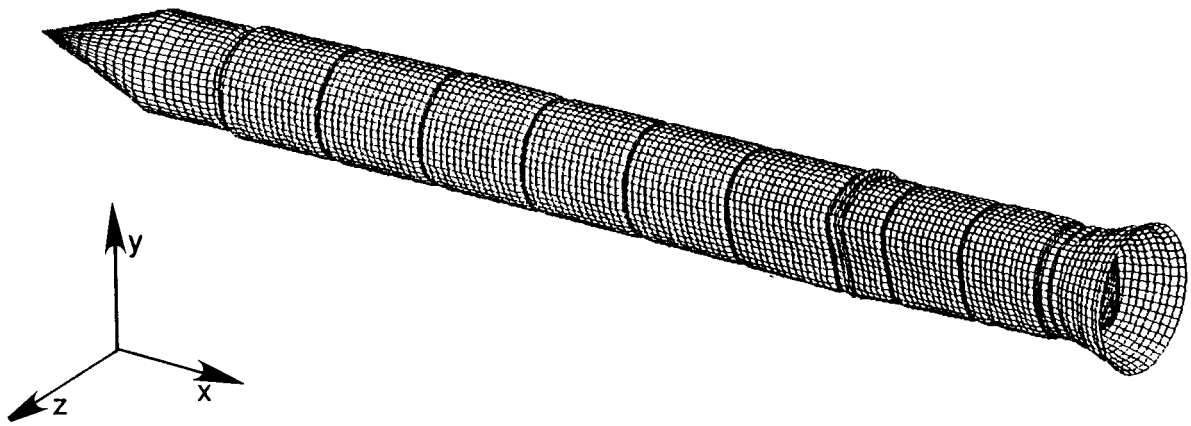
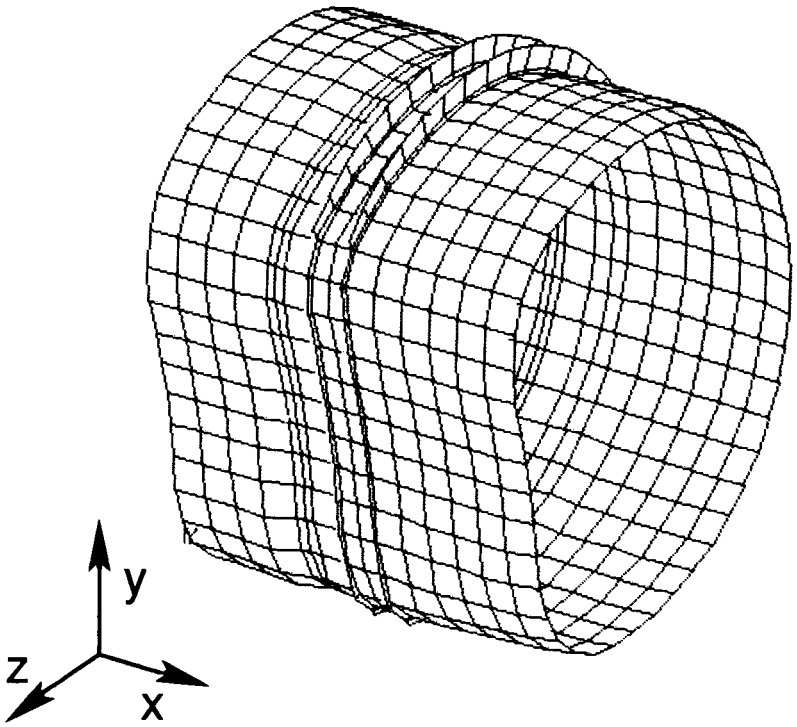
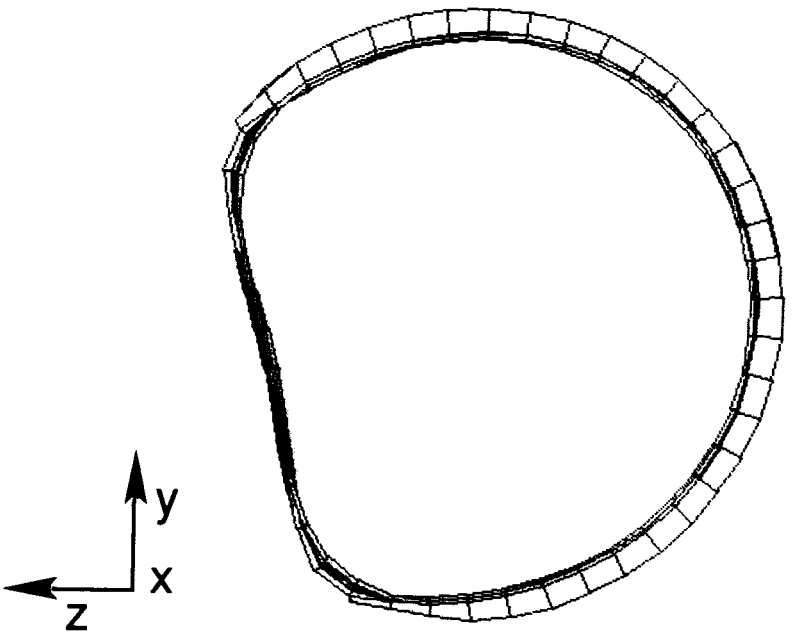


Fig. 54 Deformed geometry plot of global SRB shell model.



(a) Oblique view.



(b) Top view.

Fig. 55 Close-up view of deformed geometry at SRB/ET attachment ring interface.

Table 16. Selected processor execution times for global SRB shell model (9205 nodes, 54870 *dof*, average semi-bandwidth of 382).

Solution Phase	Processor Name	NAS CRAY-2 (CPU seconds)	VAX 11/785 (CPU seconds)
Mesh Generation	ELD	248.6	460.8
	E	2.0	70.7
	TOPO	94.3	1678.4
Form and Factor Global Stiffness	EKS	168.3	1625.0
	K	46.9	472.3
	INV	804.1	51185.1
	SSOL	17.2	295.6

Computation times for the SRB global shell analysis are given in Table 16 for selected Testbed processors. Most of the CPU time was spent in processor INV factoring the global stiffness matrix. However, several other processors (ELD,EKS, TOPO) also used a sizeable amount of CPU time. All of these processors need to be studied and improved for large-scale analysis problems. The estimated number of floating point operations required to factor the system stiffness matrix is 5,652,771,408. The compute rates for processor INV are 7.03 and 0.11 MFLOPS on the NAS CRAY-2 computer system and a VAX 11/785 computer system, respectively. The ratio of the overall execution time on a VAX 11/785 computer system to the execution time on the NAS CRAY-2 computer system is 35.6 to 1 for a single linear stress analysis of the SRB global shell model.

The new Testbed equation solvers implemented in processors BAND and ITER have also been applied to this problem. Using the skyline method in processor BAND with loop unrolling to level 4 and exploiting local memory, the solution time to factor and solve was 74.8 CPU seconds on the NAS CRAY-2 computer system (a compute rate of 127.9 MFLOPS). Processor BAND factors the global stiffness matrix in less than one-tenth the CPU time required by processor INV on the NAS CRAY-2 computer system. Using the incomplete Choleski conjugate gradient method with a sparse storage scheme, the solution was obtained after 562 iterations. The solution time was 455 seconds which corresponds to a compute rate of 20 MFLOPS.

CSM Research Directions

The research directions of the CSM activity at Langley Research Center aim to resolve deficiencies noted in today's computational structural mechanics technology. The broad objective of the CSM activity is to develop advanced structural analysis technology that will exploit modern and emerging scientific computers – such as computers having vector and/or parallel processing capabilities. The evolving computational environment (both hardware and software) is providing new opportunities to the structural analysts that enable them to study the structural behavior of complex nonlinear systems.

The current research direction of the Langley CSM activity for structural analysis technology is focused on methodology for predicting the nonlinear structural response of large-scale composite primary aircraft structures. Many of the structural analysis software systems available today can predict the nonlinear structural response of composite components. However, the lack of progressive failure analysis techniques in large-scale structural analysis systems limits the analyst in the design of composite aerospace structures. A capability to model and analyze damaged composite structures is needed in the aerospace community. In addition, designers needs analysis tools that can be used to assess the sensitivity of variations in material properties or loads on selected response parameters for complex structural systems. Finally, error sensing and control strategies for finite element solutions are needed in order to provide quantitative as well as qualitative information about the quality of the results from such calculations.

To complement the transition to multiple-instruction, multiple-data (MIMD) computers, numerical analysts are preparing a wealth of new algorithms designed to take advantage of the vector processing capability offered by many modern computers. In the past, the sparse nature of the matrices that dominate the structural analysis task has made vector processors of limited use. It is anticipated that work will continue on the development of numerical algorithms that will take full advantage of both the vector capabilities and the MIMD capabilities of future computer architectures. Such algorithms will be developed within the Testbed framework and will be evaluated on challenging structural analysis focus problems.

The Langley CSM methods research emphasizes applied structural mechanics research to accelerate the transfer of methods technology to industry. Structural analysis and computational methods that have reached a level of maturity to demonstrate high potential for solving realistic, practical structural problems will continue to be a focus. Problem-adaptive solution

strategies with error control for routine global/local stress analysis is a goal of the CSM methods research.

Summary

The CSM Testbed is a powerful methods development environment for developing structural analysis and computational methods. With enhancements and extensions for MIMD computers, the Testbed should continue to be a useful research environment for the forseeable future. It is currently being used by researchers developing structural analysis methods, numerical algorithms, and evaluating MIMD I/O strategies. The Testbed application environment provides the mechanism to allow researchers concentrating on different parts of the structural analysis problem to communicate about solutions to problems directly related to current NASA needs. The transfer of technology among researchers in structural engineering, computer science, and numerical analysis can now be accomplished more effectively than was previously possible.

An overview of the CSM activity at the NASA Langley Research Center has been presented. The CSM Testbed software system serves as a framework for structural analysis and computational methods research for high performance computers. The CSM Testbed has been described and its use demonstrated by solving selected structural analysis problems. Future directions for CSM research using the Testbed have been outlined. These future developments will take full advantage of both vector processors and parallel methods on the CRAY-2 computer system and on anticipated supercomputers of the 1990's.

REFERENCES

1. Blankenship, Charles P. and Hayduk, Robert J. (1987). "Potential Supercomputer Needs for Structural Analysis," *Supercomputing '87: Volume 2 – Industrial Supercomputer Applications and Computations*, L. P. Kartashev and S. I. Kartashev (Editors), International Supercomputing Institute, Inc., pp. 180-202.

2. Knight, Norman F., Jr. and Stroud, W. Jefferson (1985). *Computational Structural Mechanics: A New Activity at the NASA Langley Research Center*, NASA TM-87612.

3. Lotts, C. G., Greene, W. H., McCleary, S. L., Knight, N. F., Jr., Paulson, S. S., and Gillian, R. E. (1987). *Introduction to the Computational Structural Mechanics Testbed*, NASA TM-89096.

4. Stewart, Caroline B., Compiler (1989). *The Computational Structural Mechanics Testbed User's Manual*, NASA TM-100644.

5. Stewart, Caroline B., Compiler (1988). *The Computational Structural Mechanics Testbed Data Library Description*, NASA TM-100645.

6. Stewart, Caroline B., Compiler (1989). *The Computational Structural Mechanics Testbed Procedures Manual*, NASA TM-100646.

7. Bailey, F. R. (1987). "NAS — Current Status and Future Plans," *Supercomputing in Aerospace*, NASA CP-2454, pp. 13-21.

8. McLean, Donald M. (1983). *MSC/NASTRAN Programmer's Manual*, MSC NASTRAN Version 63. MSR-50, pp. 1.1-4.

9. Gillian, R. E. and Lotts, C. G. (1988). *The CSM Testbed Software System – A Development Environment for Structural Analysis Methods on the NAS CRAY-2*, NASA TM-100642.

10. Felippa, C. A. (1981). "Architecture of a Distributed Analysis Network for Computational Mechanics," *Computers and Structures*, Vol. 13, pp. 405-413.

11. Felippa, C. A. and Stanley, G. M. (1985). "NICE: A Utility Architecture for Computational Mechanics," *Proceedings of US-Europe Symposium on Finite Element Methods for Nonlinear Problems*, Bergan, P. G., Bathe, K. J., and Wunderlich, W. (Editors), University of Trondheim, Norway, pp. 447-463.

12. Whetstone, W. D. (1978). *SPAR Structural Analysis System Reference Manual, Vols. 1-4*, NASA CR-158970-1.

13. Felippa, Carlos A. (1988). *The Computational Structural Mechanics Testbed Architecture: Volume 1 - The Language*, NASA CR-178384.

14. Felippa, Carlos A. (1989). *The Computational Structural Mechanics Testbed Architecture: Volume 2 - Directives*, NASA CR-178385.

15. Felippa, Carlos A. (1988). *The Computational Structural Mechanics Testbed Architecture: Volume 3 - The Interface*, NASA CR-178386.

16. Wright, Mary A., Regelbrugge, Marc E., and Felippa, Carlos A. (1989). *The Computational Structural Mechanics Testbed Architecture: Volume 4 - The Global-Database Manager GAL-DBM.* NASA CR-178387.

17. Hurst, P. W. and Pratt, T. W. (1985). *Executive Control Systems in the Engineering Design Environment*, AIAA Paper No. 85-0619.

18. Felippa, Carlos A. (1982). "Fortran-77 Simulation of Word-Addressable Files," *Advanced Engineering Software*, Vol. 4, pp. 156-162.

19. Ortega, James M. (1988). *Introduction to Parallel and Vector Solution of Linear Systems*, Plenum Press, New York.

20. Poole, E. L. and Overman, A. L. (1988). *The Solution of Linear Systems of Equations in a Structural Analysis Code on the NAS CRAY-2*, NASA CR-4159.

21. Dongarra, J. J., Bunch, J. R., Moler, C. B., and Stewart, G. W. (1979). *LINPACK Users' Guide*, SIAM Publications, Philadelphia.

22. Rankin, C. C. and Brogan, F. A. (1986). "An Element-Independent Corotational Procedure for the Treatment of Large Rotations," *Journal of Pressure Vessel Technology*, Vol. 108, pp. 165-174.

23. Stanley, G. M. (1985). *Continuum-Based Shell Elements*, PhD Dissertation, Stanford University, Stanford, CA.

24. Park, K. C. and Stanley, G. M. (1986). "A Curved C^0 Shell Element Based on Assumed Natural-Coordinate Strains," *Journal of Applied Mechanics*, Vol. 108, pp. 278-290.

25. Kang, D. S. and Pian, T. H. H. (1987). *A Versatile and Low Order Hybrid Stress Element for General Shell Geometry*, AIAA Paper No. 87-0840.

26. Aminpour, Mohammad A. (1989). "Assessment of SPAR Elements and Formulation of Some Basic 2-D and 3-D Elements for Use with Testbed Generic Element Processor," *Proceedings of NASA Workshop on Computational Structural Mechanics – 1987*, NASA CP-10012-Part 2, Nancy P. Sykes, (Editor).

27. Rankin, C. C., Stehlin, P., and Brogan, F. A. (1986). *Enhancements to the STAGS Computer Code*, NASA CR-4000.

28. Griffin, O. Hayden, Jr. (1989). "Development and Verification of Local/Global Techniques for Laminated Composites," *Proceedings of NASA Workshop on Computational Structural Mechanics – 1987*, NASA CP-10012-Part 2, Nancy P. Sykes, (Editor).

29. Jones, R. M. (1975). *Mechanics of Composite Materials.* Scripta Book Company.

30. Whitney, J. M. (1987). *Structural Analysis of Laminated Anisotropic Plates.* Technomic Publishing Co., Inc.

31. Tsai, S. W. (1965). *Strength Characteristics of Composite Materials*, NASA CR-224.

32. Hill, R. (1948). "A Theory of the Yielding and Plastic Flow of Anisotropic Metals," *Proceedings of the Royal Society of London*, Series A, Vol. 193, pp. 281-297.

33. Azzi, V. D. and Tsai, S. W. (1965). "Anisotropic Strength of Composites," *Experimental Mechanics*, Vol. 9, pp. 283-288.

34. Hoffman, O. (1967). "The Brittle Strength of Orthotropic Materials," *Journal of Composite Materials*, Vol. 1, pp. 200-206.

35. Tsai, S. W. and Wu, E. M. (1971). "A General Theory of Strength for Anisotropic Materials,". *Journal of Composite Materials*, Vol. 5, pp. 58-80.

36. Wu, R. Y. and Stachurski, Z. (1984). "Evaluation of the Normal Stress Interaction Parameter in the Tensor Polynomial Strength Theory for Anisotropic Materials," *Journal of Composite Materials*, Vol. 18, pp. 456-463.

37. Shuart, M. J. (1985). *Short-Wavelength Buckling and Shear Failures for Compression-Loaded Composite Laminates*, NASA TM-87640.

38. Bathe, K. J. (1982). *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.

39. Stanley, G. M. and Felippa, C. A. (1986). "Computational Procedures for Postbuckling of Composite Shells," *Proceedings of US-Europe Symposium on Finite Element Methods for Nonlinear Problems*, Bergan, P. G., Bathe, K. J., and Wunderlich, W. (Editors), University of Trondheim, Norway, pp. 359-385.

40. Riks, E. (1970). *On the Numerical Solution of Snapping Problems in the Theory of Elastic Stability*, SUDDAR Report No. 410, Stanford University, Stanford, CA.

41. Riks, E. (1987). "Progress in Collapse Analysis," *ASME Journal of Pressure Vessel Technology*, Vol. 109, pp. 33-41.

42. Crisfield, M. A. (1983). "A Fast Incremental/Iterative Solution Procedure that Handles Snap-Through," *Computers and Structures*, Vol. 13, pp. 55-62.

43. Stroud, W. Jefferson, Greene, William H., and Anderson, Melvin S. (1984). *Buckling Loads of Stiffened Panels Subjected to Combined Longitudinal Compression and Shear: Results Obtained with PASCO, EAL, and STAGS Computer Programs*, NASA TP-2215.

44. Starnes, James H., Jr., Dickson, John N., and Rouse, Marshall (1984). "Postbuckling Behavior of Graphite-Epoxy Panels," *ACEE Composite Structures Technology: Review of Selected NASA Research on Composite Materials and Structures*, NASA CP-2321, pp. 137-159.

45. Williams, Jerry G., Anderson, Melvin S., Rhodes, Marvin D., Starnes, James H., Jr., and Stroud, W. Jefferson (1979). *Recent Developments in the Design, Testing, and Impact-Damage Tolerance of Stiffened Composite Panels*, NASA TM-80077.

46. Pipes, R. B. and Pagano, N. J. (1970). "Interlaminar Stresses in Composite Laminates Under Uniform Axial Extension," *Journal of Composite Materials*, Vol. 4, pp. 538-548.

47. Raju, I. S., Whitcomb, J. D., and Goree, J. G. (1980). *A New Look at Numerical Analyses of Free-Edge Stresses in Composite Laminates*, NASA TP-1751.

48. Bushnell, David (1985). "Static Collapse: A Survey of Methods and Modes of Behavior," *Finite Elements in Analysis and Design*, Vol. 1, pp. 165-205.

49. Hartung, R. F. and Ball, R. E. (1973). *A Comparison of Several Computer Solutions to Three Structural Shell Analysis Problems*, AFFDL TR-73-15.

50. Almroth, B. O. and Brogan, F. A. (1981). "Computational Efficiency of Shell Elements," *Nonlinear Finite Element Analysis of Plates and Shells*, Hughes, T. J. R., Pifko, A., and Jay, A. (Editors), AMD Vol. 48, ASME, pp. 147-165.

51. Greene, William H., Knight, Norman F., Jr., and Stockwell, Alan E. (1986). *Structural Behavior of the Space Shuttle SRM Tang-Clevis Joint*, NASA TM-89018.

52. McConnaughey, Paul K., Lee, Henry M., and Moore, Carleton J. (1988). "Supercomputing and the Redesign of Space Shuttle Propulsion Elements," *CRAY Channels*, Vol. 10, No. 1, pp. 8-13.

53. SRB Integration Branch, Space Shuttle Systems Division (1977). *SRB Systems Data Book, Volume 1*. Report No. SE-019-083-2H, NASA George C. Marshall Space Flight Center, (Revision A).

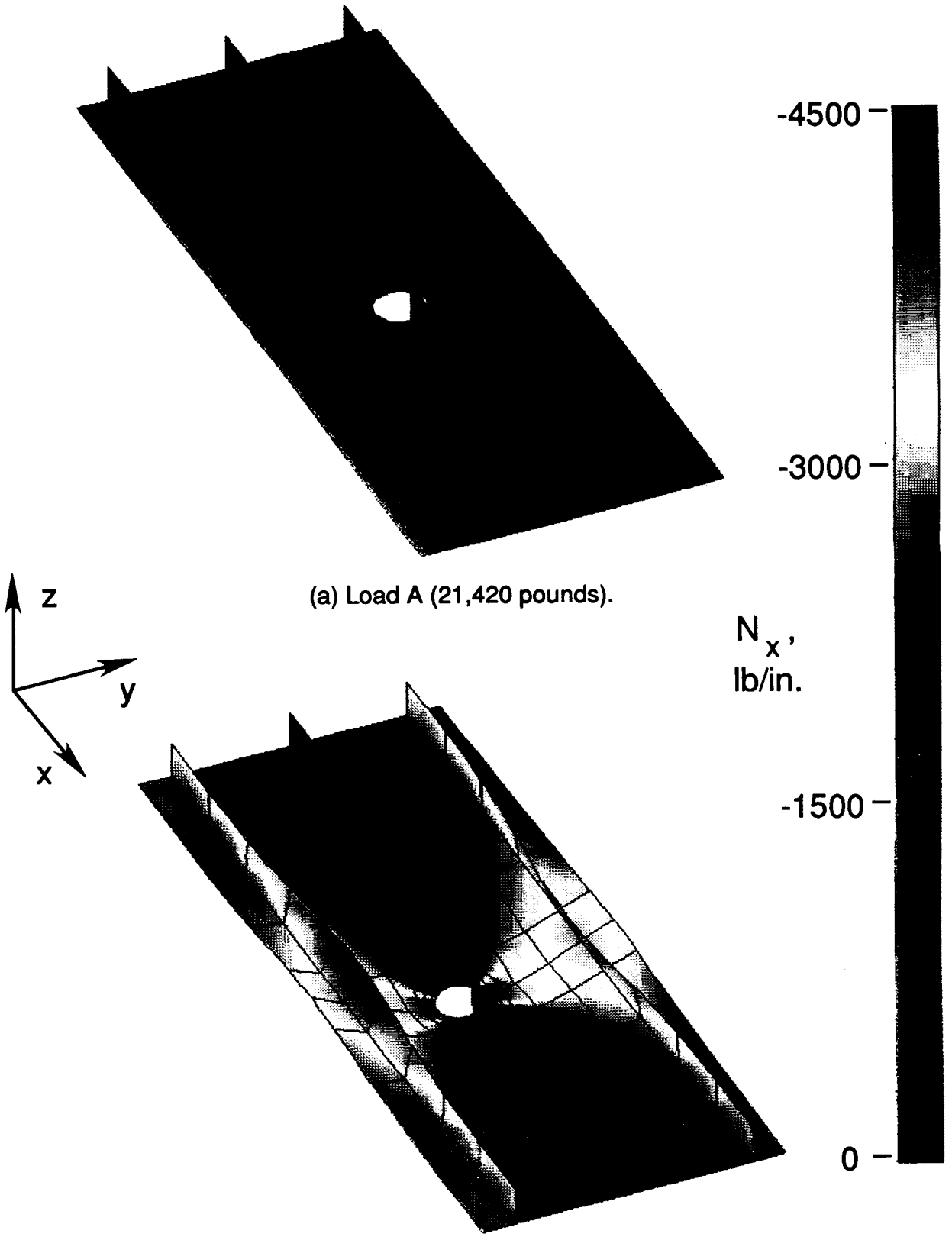
54. Knight, Norman F., Jr., Gillian, Ronnie E., and Nemeth, Michael P. (1987). *Preliminary 2-D Shell Analysis of the Space Shuttle Solid Rocket Boosters*, NASA TM-100515.

55. Knight, Norman F., Jr. (1987). *Nonlinear Shell Analyses of the SRB/ETA Ring Interface*, NASA TM-89164.

TRADEMARKS

APPENDIX

Color Photographs of Figures 17, 19, 28, 34, 41, and 45

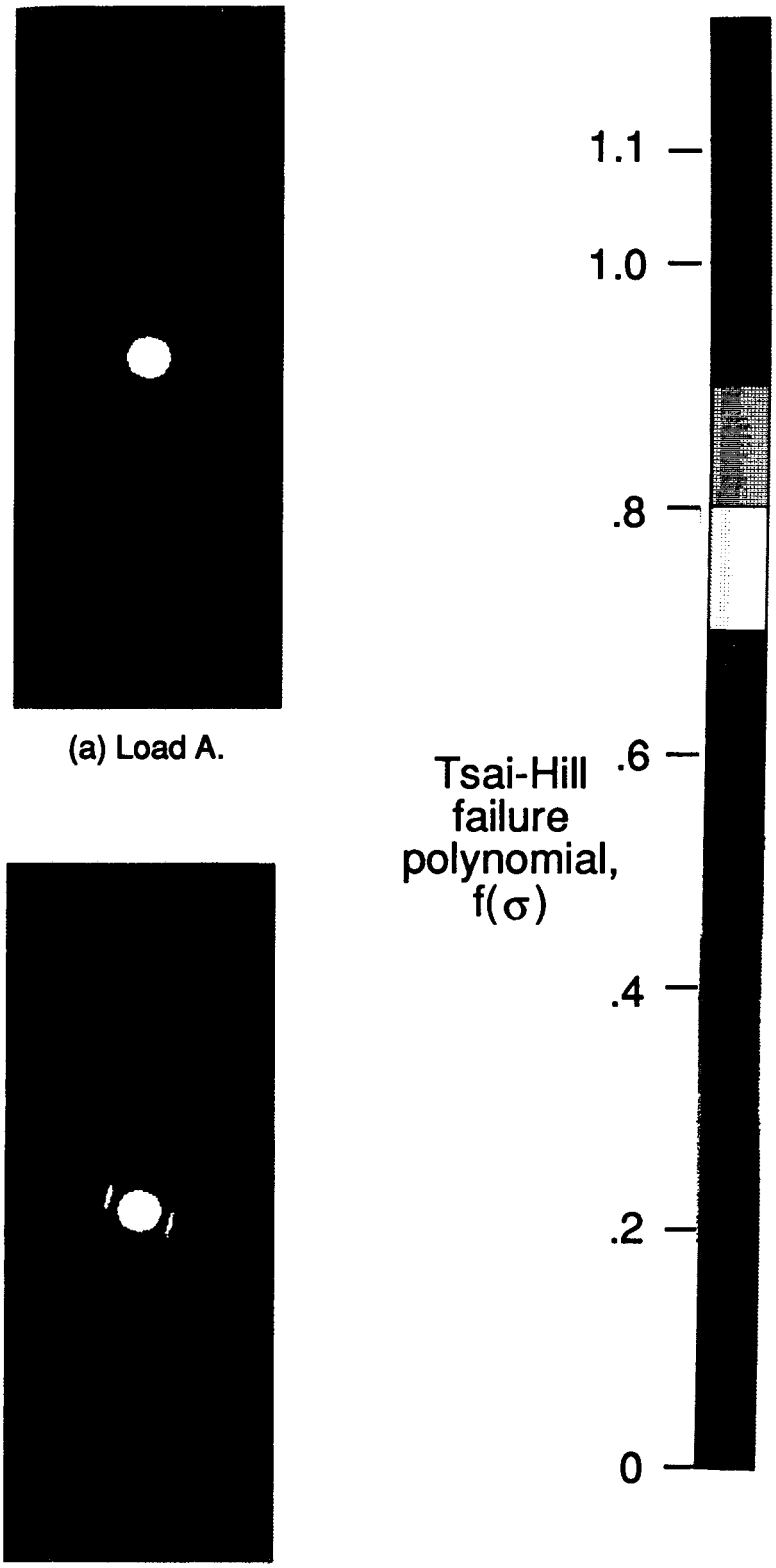


(a) Load A (21,420 pounds).

(b) Load B (37, 800 pounds).

Fig. 17 Deformed geometry shapes with N_x distributions.

ORIGINAL PAGE
COLOR PHOTOGRAPH

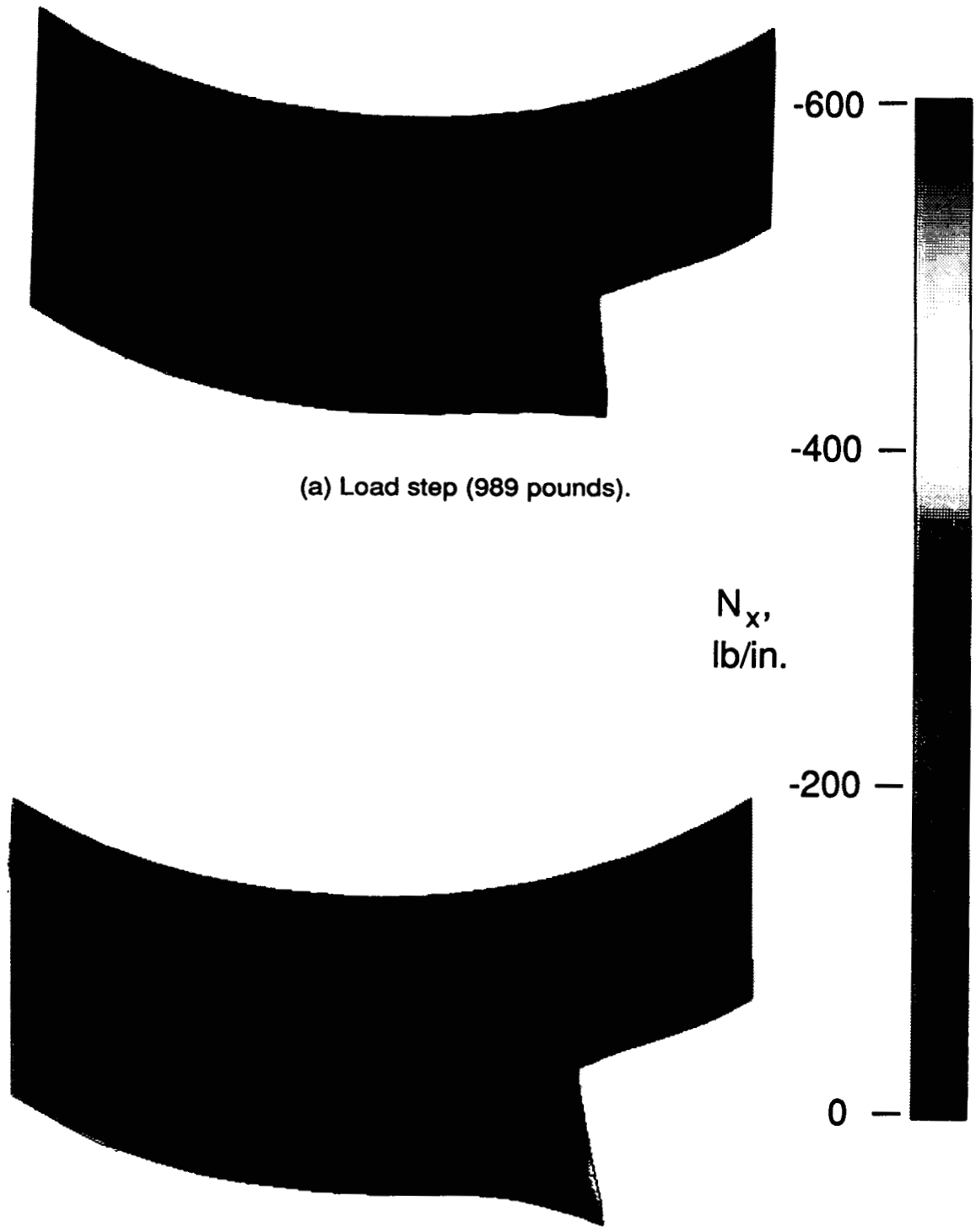


(a) Load A.

(b) Load B.

Fig. 19 Tsai-Hill criterion for outer fiber surface of panel skin.

ORIGINAL PAGE
COLOR PHOTOGRAPH

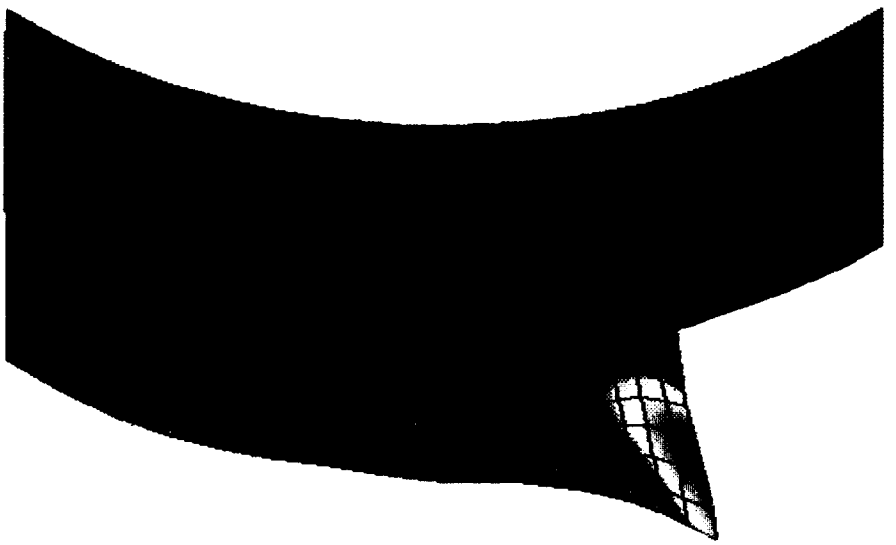


(a) Load step (989 pounds).

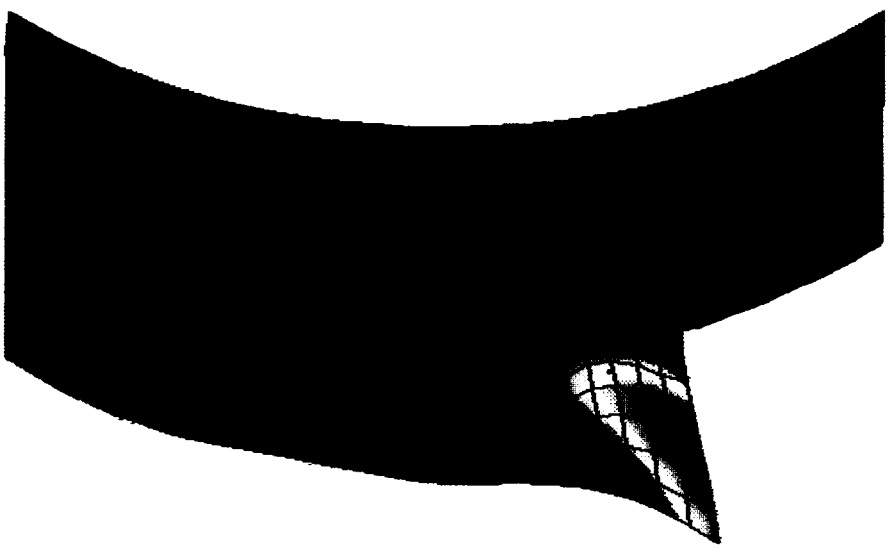
(b) Load step 15 (1962 pounds).

Fig. 28 Deformed geometry plots for several load steps - Mesh 3 results.

ORIGINAL PAGE
COLOR PHOTOGRAPH



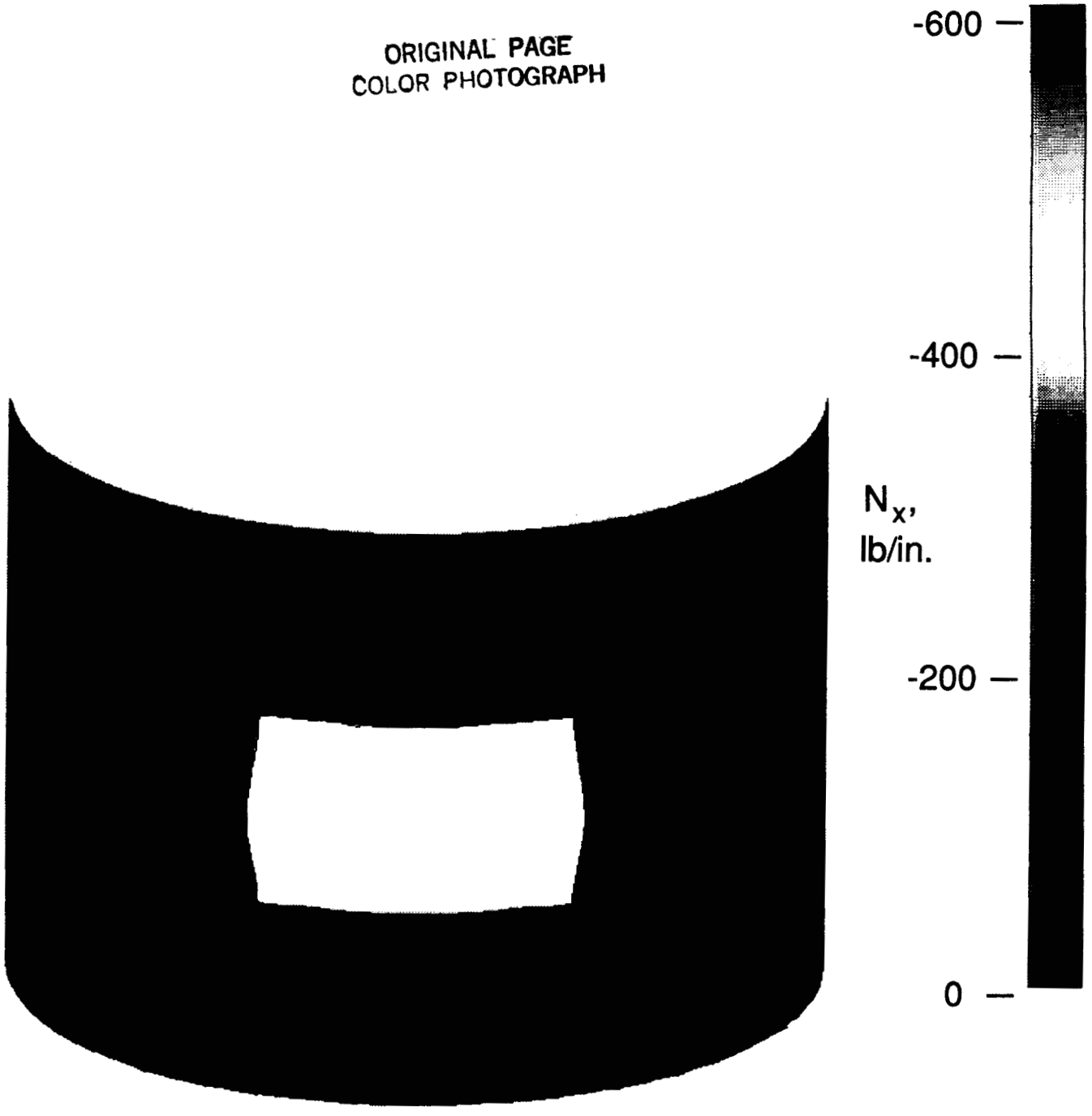
(c) Load step 24 (2966 pounds).



(d) Load step 30 (2926 pounds).

Fig. 28 Concluded.

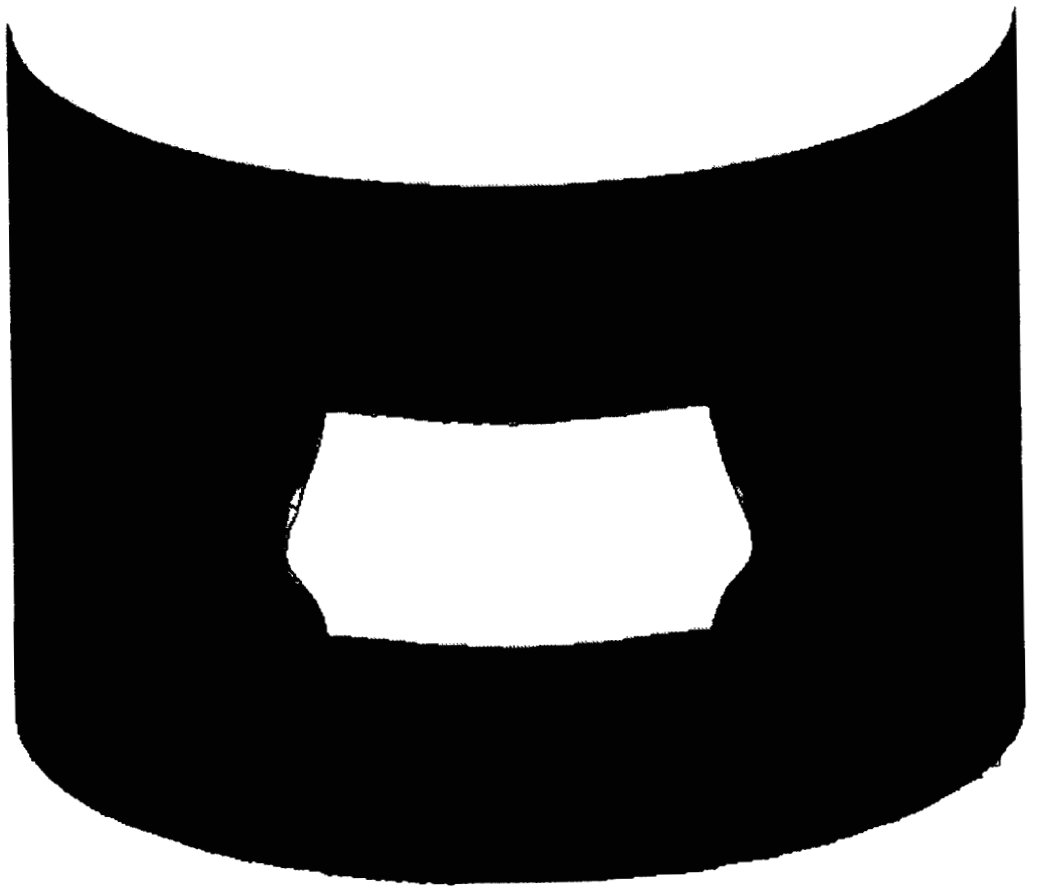
C-2



(a) Load step 9 (992 pounds).

Fig. 34 Deformed geometry plots for several load steps - Half-model results.

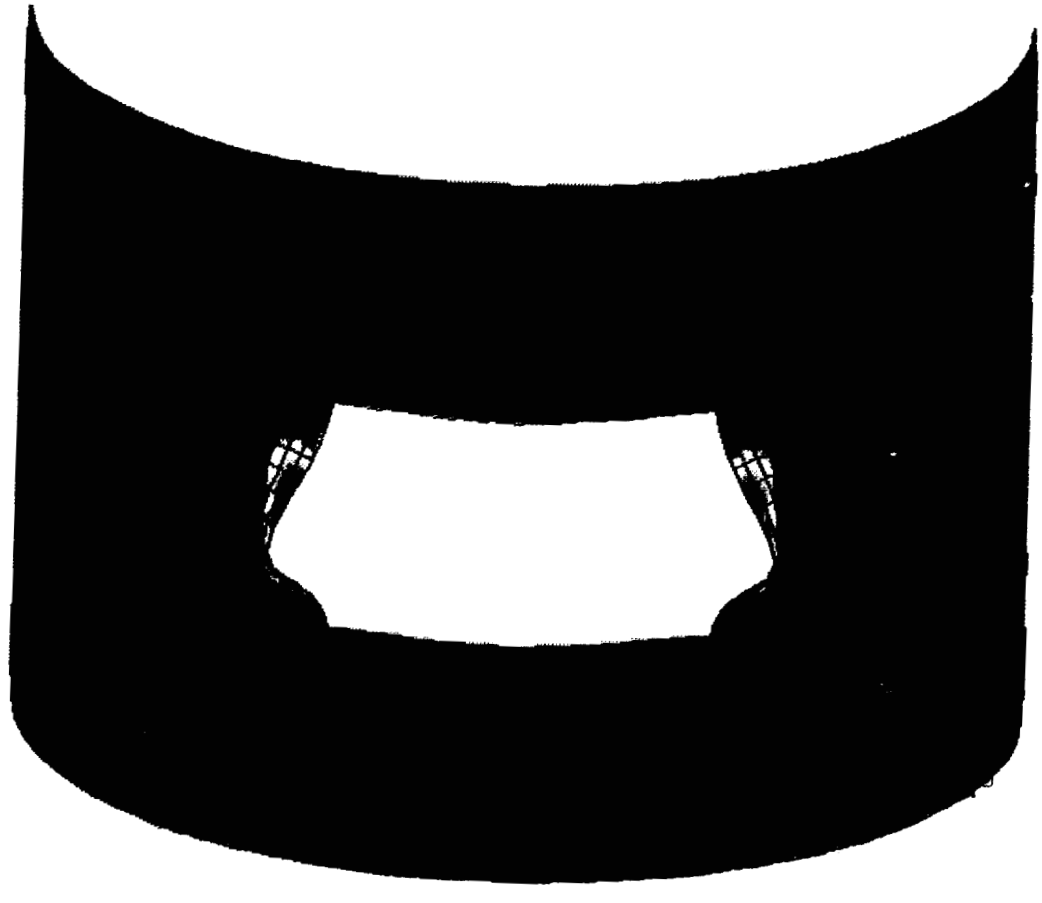
ORIGINAL PAGE
COLOR PHOTOGRAPH



(b) Load step 15 (1959 pounds).

Fig. 34 Continued.

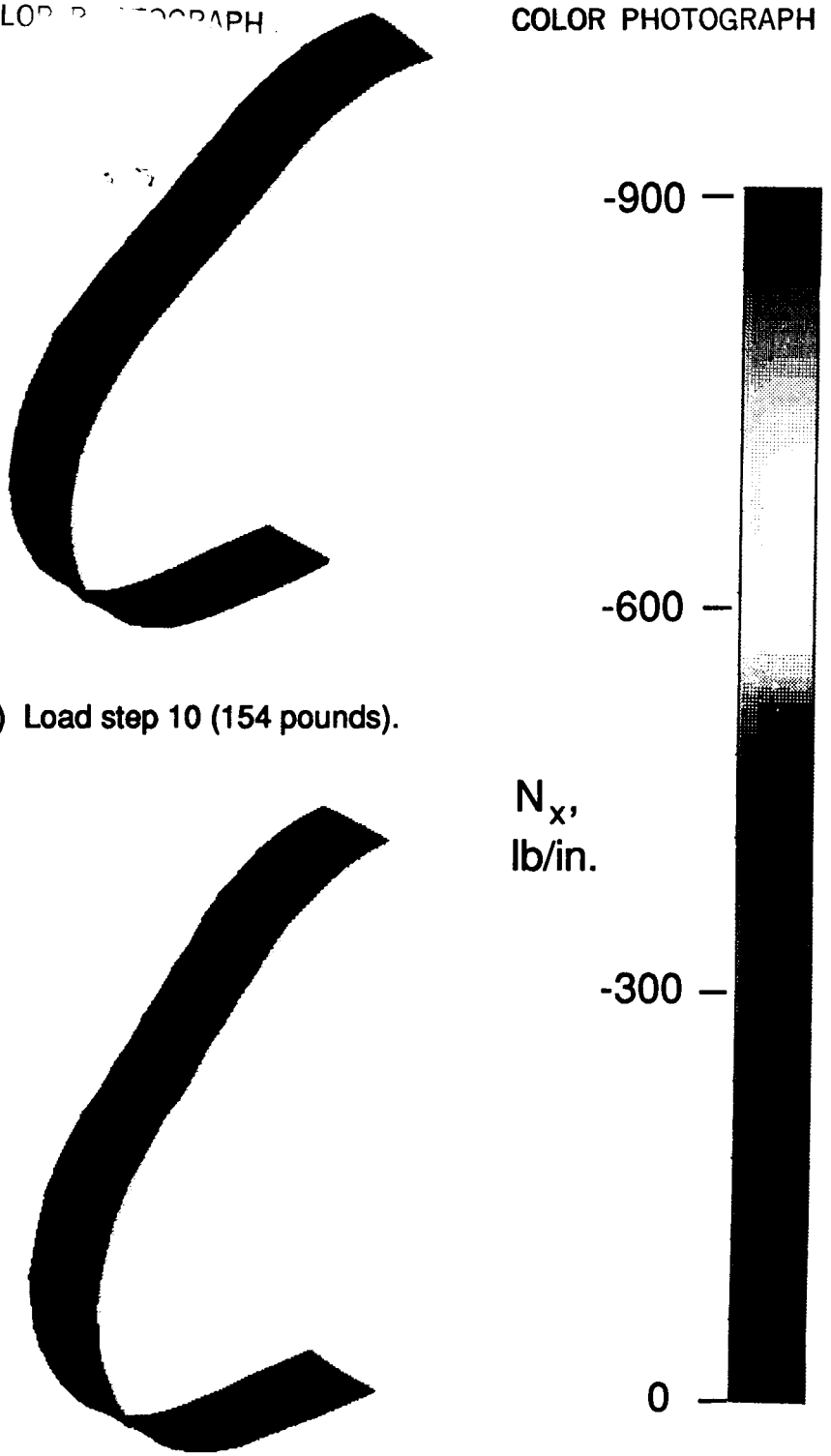
ORIGINAL PAGE
COLOR PHOTOGRAPH



(c) Load step 25 (2943 pounds).

Fig. 34 Concluded.

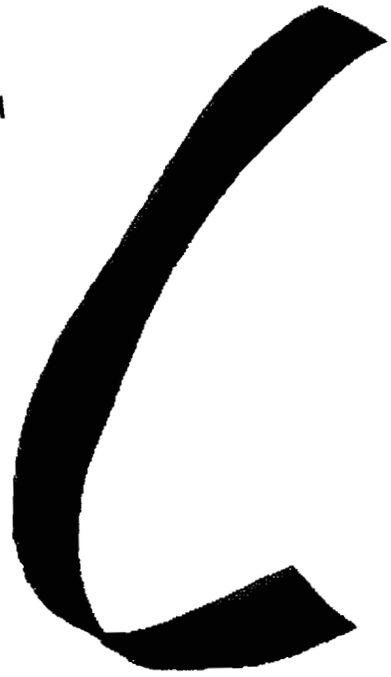
ORIGINAL PAGE
COLOR PHOTOGRAPH



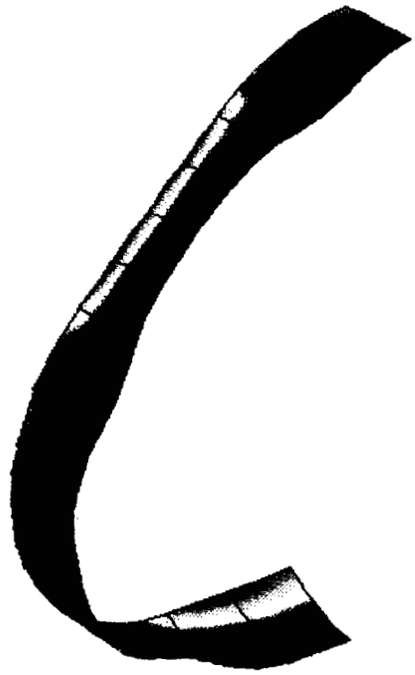
(a) Load step 10 (154 pounds).

(b) Load step 20 (300 pounds).

Fig. 41 Deformed geometry plots for several load steps.



(c) Load step 30 (1689 pounds).



(d) Load step 40 (2464 pounds).

Fig. 41 Concluded.

ORIGINAL PAGE
COLOR PHOTOGRAPH

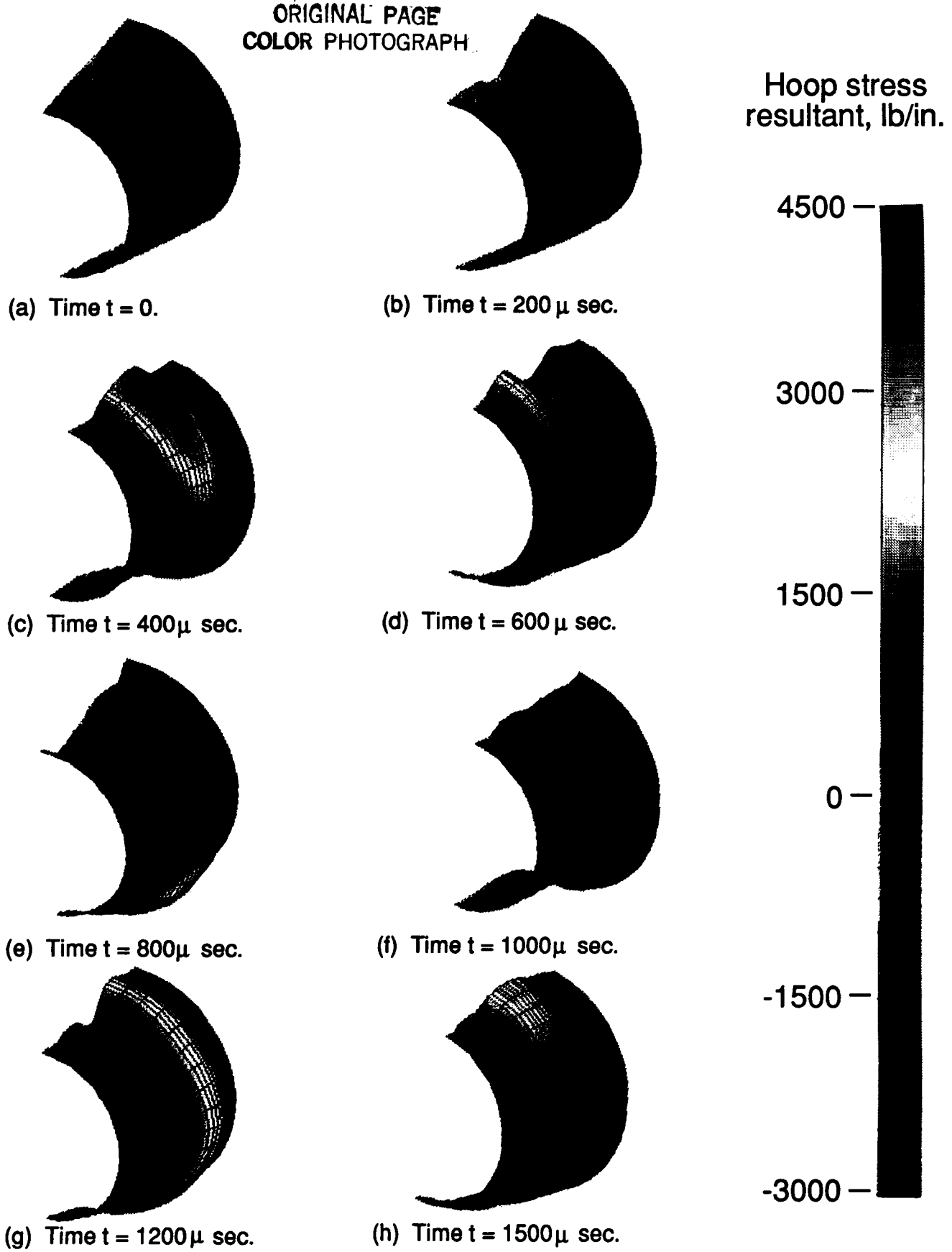


Fig. 45 Deformed shapes for truncated conical shell during the transient response.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM-100643		

4. Title and Subtitle
Large-Scale Structural Analysis – The Structural Analyst, the CSM Testbed, and the NAS System

5. Report Date
March 1989

6. Performing Organization Code

7. Author(s)
Norman F. Knight, Jr., Susan L. McCleary, Steven C. Macy, and Mohammad A. Aminpour

8. Performing Organization Report No.

9. Performing Organization Name and Address
NASA Langley Research Center
Hampton, VA 23665-5225

10. Work Unit No.
505-63-01-10

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, DC 20546-0001

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes
N.F. Knight, Jr., Langley Research Center, Hampton, VA; S.L. McCleary and S.C. Macy, Planning Research Corporation, Hampton, VA; and M.A. Aminpour, Analytical Services and Materials, Inc., Hampton, VA.

16. Abstract

A research activity named Computational Structural Mechanics or CSM at the NASA Langley Research Center is described. This activity is developing advanced structural analysis and computational methods that exploit high-performance computers. New methods are developed in the framework of the CSM Testbed software system and applied to representative complex structural analysis problems from the aerospace industry. An overview of the CSM Testbed methods development environment is presented and some new numerical methods developed on a CRAY-2 are described. Selected application studies performed on the NAS CRAY-2 are also summarized.

17. Key Words (Suggested by Authors(s))	18. Distribution Statement
Computational Structural Mechanics Structural Analysis	Unclassified—Unlimited Subject Category 39

19. Security Classif.(of this report)	20. Security Classif.(of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	104	A06

NASA FORM 1626 OCT 86
For sale by the National Technical Information Service, Springfield, Virginia 22161-2171