BCS BOEING COMPUTER SERVICES COMPANY
A Division of The Boeing Company

EXPERIENCES WITH LARGE MSC/NASTRAN MODELS ON CYBER AND CRAY*

by

A. Mera, R. Cigel, B. Mukherji, and E. Herness
Boeing Computer Services Company
Seattle, Washington

ABSTRACT

As MSC/NASTRAN finite element models increase in size and complexity, the demand on computer software, hardware and other resources may exceed those initially allocated for the project. Successful completion of a large finite element analysis often requires understanding resource utilization as measured in terms of computation time, volume of data handled and stored, cost, and human endurance.

The engineer is usually relieved of the resource management task by MSC/NASTRAN's sophisticated program structure and data management technology. The program has the capability to estimate the necessary CP time, I/O time, primary and secondary storage, compare against resources available on the particular computer model, and then automatically trade off certain limiting resources against others. For large problems, however, situations arise when the NASTRAN user is forced to intervene. He finds it more efficient to schedule resources to match requirements. This competence is especially useful when computers with different characteristics are at one's disposal.

Several MSC/NASTRAN problems were run on Boeing Computer Services Company (BCS) CYBER 760 and CRAY-1S computers. The results provided some interesting insights into MSC/NASTRAN's numerical algorithms, its relative performance on the above computers, and its utilization of vector processing on the CRAY.

---

*Presented at the MSC/NASTRAN Users' Conference, March 18-19, 1982.

1.0 INTRODUCTION

Several years ago, finite element analysis was typically done on a single mainframe computer and one seldom had the option to use another computer. In those days the engineer-user's main concern was accurate finite element modeling and interpretation of displacement and stress results. Data generation, checking, and graphic display was a tedious task. MSC/NASTRAN could be relied upon to carry through the analysis and return a mountain of printout. The engineer was familiar with a few work-horse solution sequences for statics and dynamics like RF24 and RF25; he could trust automatic checkpoint/restart to reduce the cost of reanalysis; and he could even come back a year later and rerun the old deck on a newer, upward compatible version.

In training this engineer for MSC/NASTRAN, a three-day workshop on the finite element library, on solution sequences, and on input/output was sufficient. He had relatively little interest and spent relatively little time in examining the underlying matrix algebra, numerical algorithms, data management, and hardware aspects of NASTRAN. If he resisted the temptation of advanced features and special options, he got quick results without the need for specialist support to keep him out of trouble.

Since those days MSC/NASTRAN has expanded considerably in response to developments in computing power and demands for more capabilities. New data management concepts were implemented and special solution sequences were written for the superelement technique. The user/hardware interface has also changed drastically (Figure 1.1). Instead of carrying punched cards to the computer and waiting for printout, the present-day user can communicate through an intelligent terminal: a highly interactive local system equipped with sophisticated data generation, editing, and graphic display capabilities (reference 1). At BCS, this "engineering work station" is linked to the MAINSTREAM® computer network allowing submission of MSC/NASTRAN data for execution and retrieval of output in a distributed processing environment.

Growth pains have developed as the user, the program, and the hardware tackle ever larger finite element models. One such symptom is that program demands on hardware may reach the point where the available resources are no longer sufficient because certain bottlenecks and constraints manifest themselves. Now the user finds it advantageous and necessary to take corrective action. This requires that the analyst understand the resource management capabilities and limitations of the computer and of NASTRAN. He must also evaluate his own needs such as throughput, cost tradeoffs, and software availability. The solution may be to select the computer, e.g., VAX, CYBER, or CRAY, best suited to analyzing his structure.

This paper focuses on executing MSC/NASTRAN as a function of problem size and available hardware resources.

2.0 MSC/NASTRAN PROGRAM CHARACTERISTICS AND RESOURCE REQUIREMENTS

The major computation phases in NASTRAN are as shown in Figure 2.1. The program has the capability to estimate the necessary CP time, I/O time, primary and secondary storage, to compare against resources available on a particular computer model, and to automatically tradeoff certain limiting resources against others. For example, NASTRAN will calculate the amount of primary storage available and will automatically choose to decompose the active portion of the stiffness matrix either in core or with spill to secondary storage. In using the spill technique, the code ensures that the matrix will be decomposed but with the penalty of increased I/O activity.

Similarly, it will choose one of two methods for forward/backward substitution (FBS), attempting to select the one which minimizes computing time. These program characteristics have been implemented to minimize the need for the NASTRAN user to intervene in scheduling resources to match the requirements.

Alternate methods are available for other numerical algorithms. Some of the more important ones appear in Figure 2.2. Depending on the algorithm, the method used may be under program control or user control and is selected on the basis of matrix sparsity, symmetry, number of eigenvalues, etc. The family of matrix operations and numerical algorithms for the solution of large sparse matrix equations is described in the MSC/NASTRAN documentation (references 2, 3, 4, 5, and 6).

Another dimension in coding sophistication is the technique implemented in NASTRAN for passing large amounts of data between primary and secondary memory. It is a characteristic of programs solving large matrices in limited central memory that the computationally intensive operations must be preceded by reorganization for more efficiency. When comparing NASTRAN arithmetic solution times with the rated performance of a large computer or with solution times of specialized matrix solvers, the following program and data structure characteristics are significant:

- All data resides on secondary storage between computational modules. Central memory is used as a scratch pad.
- Compact matrix storage is achieved through special matrix packing/unpacking routines which eliminate zeros.
- Compatibility of NASTRAN on many different computers requires that FORTRAN be the common coding language and that machine language coding is limited to arithmetic inner loops and I/O handlers.
- Unnecessary multiply/add operations are eliminated by operating primarily on non-zeros for sparse matrices. (While advantageous in scalar processing computers, this will usually hinder vector processing machines.)

In spite of this software sophistication the oft quoted statement that MSC/NASTRAN has no limitation on problem size should be interpreted judiciously. In practice, large problems will always tax computers, and the smaller the computer, the sooner the limit will be reached. Some symptoms of reaching the limit are:

- Wall clock time becomes many hours and the job won't finish overnight.
- Scratch disk space is insufficient and special handling by the operator is required.
- The engineer can't find room to store all the paper output and a microfiche reader may not be available.
- Special techniques and the assistance of a sophisticated NASTRAN user is required for job completion.

As this threshold is reached the experienced resource manager must intervene. He may makes special arrangements with the computer center operators, recommend superelements, or introduce checkpoint/restarts. He may decide that by sending the job to a more appropriate computer, a job which was complicated and costly becomes uncomplicated and less expensive.

3.0 BENCHMARK RESULTS ON CYBER AND CRAY

Four NASTRAN problems are presented in this study; they vary in size from small ones best suited for the CYBER or the VAX to problems large enough to be well-suited for the CRAY. Each model was run on two machines: a CYBER 760 with $400{,}000_8$ $(131{,}000_{10})$ words of primary storage and secondary storage on 885 disks (no spill across spindles) and a CRAY-1S with $2{,}000{,}000_{10}$ words of primary storage and secondary storage on DD19 and DD29 disks. The accompanying figures include a picture of each finite element model, size parameters, and resource utilization data on CYBER and CRAY.

The benchmark runs were evaluated according to the following criteria:

- Resource utilization.
- Total price units.
- Total residency time on the computer (for jobs exceeding one full night shift, the rollout time until the next night shift must be included).
- Need for special handling.

NASTRAN's performance on the VAX 11/780 has not been included as we have not yet gathered sufficient quantitative data, but enough qualitative experience is available for drawing conclusions. The VAX is an excellent machine for low-to-medium sized problems run in batch mode or interactively. At BCS most VAX machines are used in an interactive mode for pre- and post-processing activities. For large problems and heavy usage the machine becomes inefficient: long turnaround times, disk space not available, and "thrashing" when considerable virtual memory is used. The person who generates large NASTRAN models soon becomes rather unpopular. A way out is to restrict VAX activities to data generation and checkout and to the solution of smaller problems; the number crunching is transferred to CYBER or CRAY computers.

Characteristics for Model 1 are shown in Figure 3.1. It is a buckling eigenvalue extraction by the inverse power method: 85% of the CP time is spent in matrix modules, namely, 15 SDCOMP and 110 FBS operations. About 90% of the time in the matrix modules is spent in operations other than decomposition and forward/backward substitution, presumably packing and unpacking matrices. MSC/NASTRAN on the CRAY appears not yet tuned for these types of operations. The CRAY job residency time was only one-sixth of the CYBER time; this performance may be desirable in some circumstances even at the higher cost.

Model 2 results are shown in Figure 3.2. The major part of the cost is spent in the Guyan reduction, which is a series of SDCOMP, FBS, and MPYAD matrix operations. A significant difference was noted: MSC/NASTRAN chose to use FBS Method 2 in the CRAY run (42% of total CPU cost) versus Method 1 in the CYBER run (20% of total CPU cost).

Model 3 (Figure 3.3) is a static analysis of a solid modeled with 3055 grid points, 2440 solid elements, and a large number of rigid constraint relations. The major cost source in the CYBER can be attributed to multi-point constraint elimination (32%) and decomposition with spill of the K_{ff} matrix (26%). Checkpointing and a large amount of output created enough secondary storage requirement to necessitate special disk space requests. By contrast, the CRAY run spends only 18% of total CP time in decomposition.

Finally, Model 4 (Figure 3.4) shows the CRAY to be a clear winner. In this static analysis with 2 boundary conditions, the major operation is the decomposition of a large matrix with a substantial bandwidth ($C_{rms} = 627$) causing 50% spill at maximum central memory on the CYBER. When the job crashed after 10 hours wall clock time because of insufficient disk space on specially assigned units, the job was rerun on the CRAY. Results were out within 1 hour without any special handling requirements. If special system configuration of disk units were used, the job could have been run on the CYBER in two overnight sessions for a total time of about 28 hours. Job residency was therefore decreased by a factor of 26 and a cost saving of over 84% was achieved.

Performance data for all models appear in Figure 3.5. A graphical comparison of the four cases is shown in Figure 3.6.

In an attempt to understand NASTRAN performance on large models on the CRAY, we also chart decomposition statistics for Models 3 and 4. Theoretically a banded, symmetric decomposition requires approximately $\frac{1}{2} NC_{rms}^2$ operations. Therefore one can compare the theoretical decomposition time with the actual time (which includes time for packing and I/O transfer)and develop timing estimates. The matrix characteristics and some formulas for the timing constant M are shown in Figure 3.7.

4.0 CONCLUSIONS

Our experience and the data presented lead us to the following conclusions:

- The VAX is the most user friendly computer for small to moderate problems. Large problems and heavy activity by other users will slow the VAX down and fill up the limited disk space; at BCS, this threshold becomes apparent for static problems at approximately 1000 grid points.

- The CYBER is presently the most user friendly large scale computer. For very large problems, considerable central memory spill, and secondary storage limitations occur. Abundant software is available for pre- and post-processing.

- The CRAY has been quite reliable. Cost savings relative to CYBER can be achieved for solutions dominated by large matrix operations where the two million word memory and vectorization can be used advantageously. Residency time reductions are substantial. A problem in which the C_{rms} is greater than 300 for SDCOMP will clearly run much faster and with less expense on the CRAY than the CYBER.

Comments on the benchmark problems:

- These were just a first set of runs. We plan to do additional studies with these problems. We will gather quantitative data on VAX runs and also examine how parameters can be changed to improve performance on the large computers.

Comments on NASTRAN performance on the CRAY:

- By default, estimation formulas will optimize wall clock time as the criteria for choosing among matrix algorithms. We need to exercise System Cell 84 to see the effect of optimizing CP or I/O.

- CRAY runs show considerable discrepancy between estimated and actual CP times for matrix operations. The MSC/NASTRAN estimation formula is too optimistic in many cases.

- We need to evaluate the cost and throughput benefits of dynamically allocating memory.

- Both MSC and CRAY Research recognize that the CRAY needs efficiency improvements in transferring large files (data base, checkpoint, printout) to backend storage and in postprocessing binary files.

- It appears that the CRAY matrix packing/unpacking code and sparse matrix decisions need examination and possible major algorithmic changes.

The experience gained from the above benchmarks demonstrate the importance of the resource estimation and management task on large NASTRAN problems. Simply choosing the appropriate computer can have a marked effect on the success, cost, and ease of analyzing a structure. The user gains considerable benefits by having:

- Solid grounding in the solution sequence proposed.
- Understanding of the fundamentals of matrix operations and numerical algorithms.
- Ability to estimate primary and secondary storage utilization.
- Familiarity with data base management.
- Knowing the available computer resources and being on good terms with data center operations.

REFERENCES

1. Herness, E., Robl, F. J., "MSC/NASTRAN Plotting Interface to Intelligent Systems," MSC/NASTRAN Users Conference, March 1980.

2. MacNeal, R. H., editor, MSC/NASTRAN Theoretical Manual, May 1974.

3. McLean, D., editor, MSC/NASTRAN Programmer's Manual, September 1981.

4. Joseph, J., editor, MSC/NASTRAN Applications Manual, January 1982.

5. Gockel, M., editor, MSC/NASTRAN Handbook of Superelement Analysis, November 1981.

6. McCormick, C. W., "Data Structure and Computation Techniques in MSC/NASTRAN," Symposium for CAD in the 80's, November 1980, Melbourne.

Figure 1.1

BCS Craypower

Fully Integrated Service

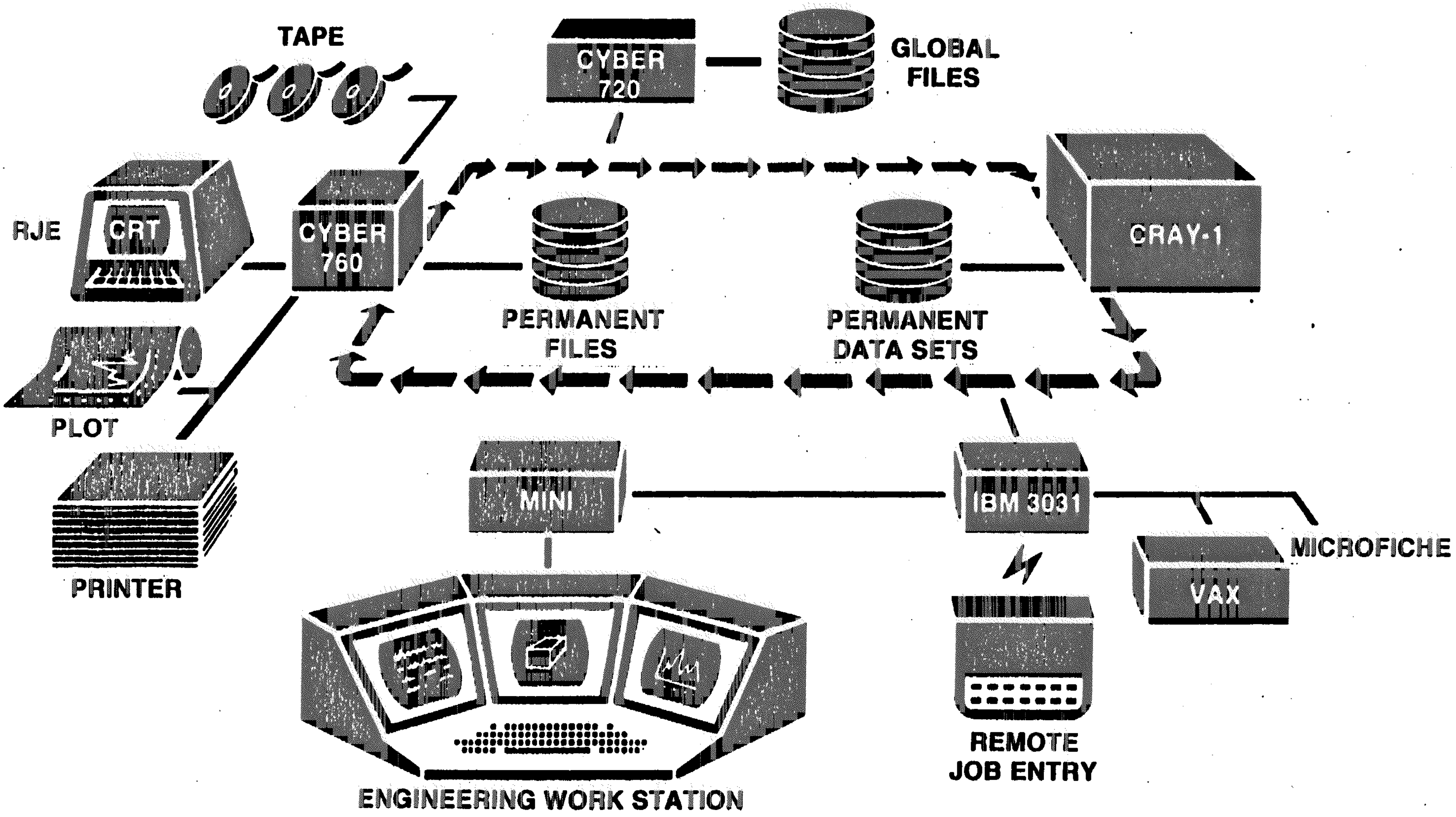


Figure 2.1

MSC/NASTRAN Solution Flow

Computation Phases		% Total
Input Data Processing Tables, Sorting, Resequencing	SCALAR	10-30%
Matrix Generation K, B, M, P, etc.	SCALAR	10-30%
Matrix Solution MPYAD, SDCOMP, FBS, UDCOMP	VECTOR	20-60%
Data Recovery Formatting, Sorting, Plotting Max/Min Search	SCALAR	20-50%

Figure 2.

Matrix Algebra and Numerical Methods in MSC/NASTRAN

- **Symmetric Decomposition of Sparse Matrices (SDCOMP)**
    - Active/Passive Columns
    - Cholesky
    - Spill
    - Error Analysis

- **Unsymmetric Decomposition (UDCOMP)**

- **Forward-Backward Substitution (FBS)**
    - Method 1
    - Method 2

- **Multiply/Add (MPYAD)**

$$D = A^T \cdot B + C \text{ or } A \cdot B + C$$

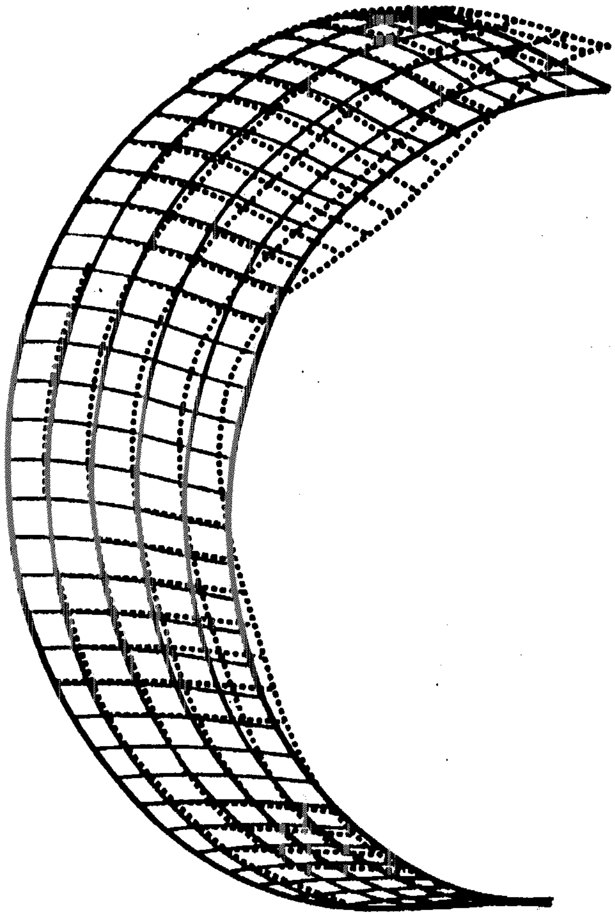
$$A\begin{bmatrix}\text{Sparse}\\ \text{Dense}\end{bmatrix} \cdot B\begin{bmatrix}\text{Sparse}\\ \text{Dense}\end{bmatrix}$$

- **Eigenvalue Extraction**
    - Givens, Modified Givens
    - Inverse Power
    - Generalized Dynamic Reduction

- **Solution of a System of Differential Equations**
    - Linear
    - Nonlinear

Figure 3.1

Buckling Analysis



Grids	234
Elements	180 QUAD 4

Eigenvalue Extraction, Inverse Power

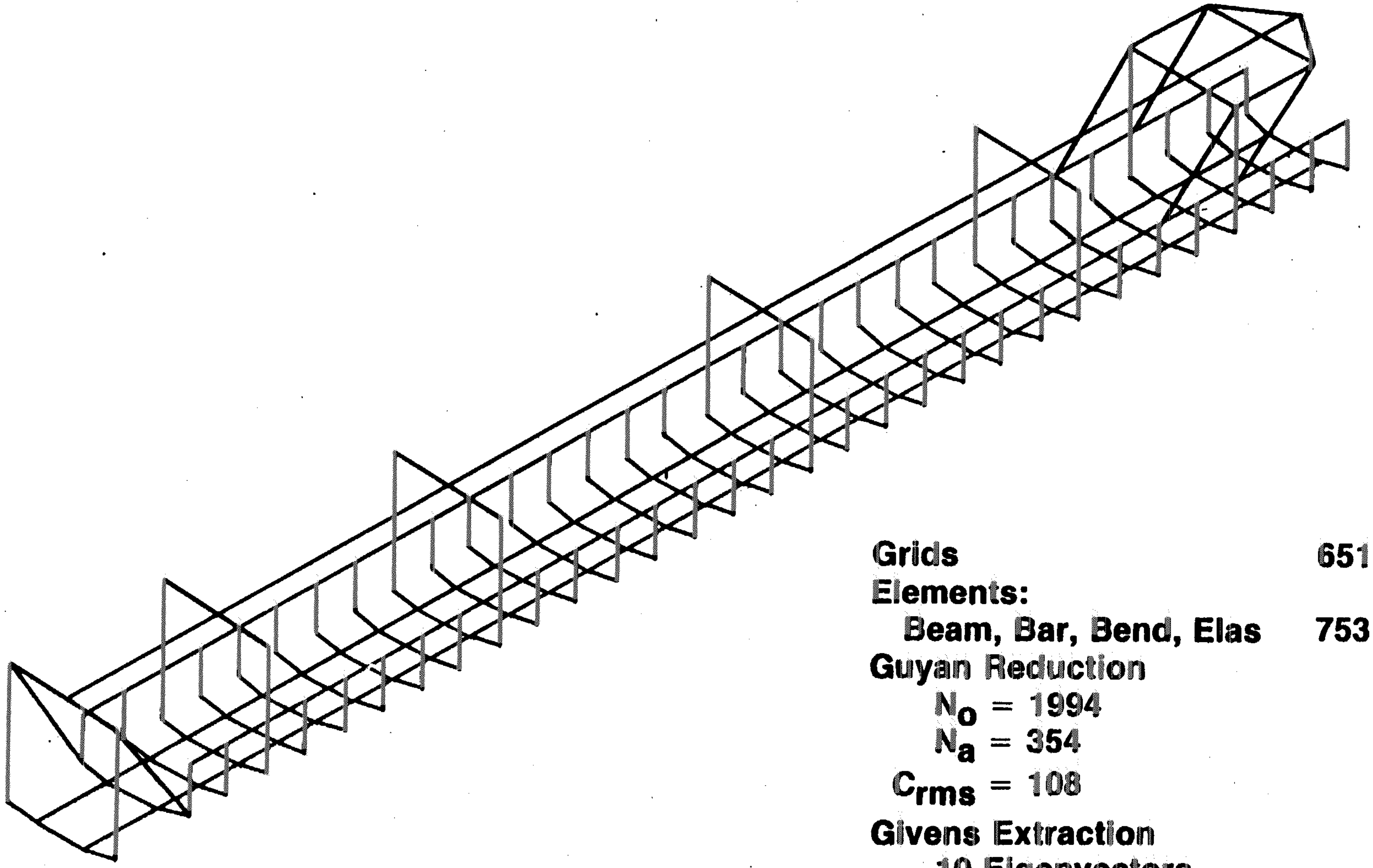
$N_a = 1224 \quad C_{rms} = 41$

15 SDCOMP + 110 FBS

11 Eigenvectors

Figure 3.2

Transporter, Vibration Modes



Grids	651
Elements:	
Beam, Bar, Bend, Elas	753
Guyan Reduction	
$N_o = 1994$	
$N_a = 354$	
$C_{rms} = 108$	
Givens Extraction	
10 Eigenvectors	

Figure 3.3

PIN-BOSS STATIC ANALYSIS

Model characteristics:

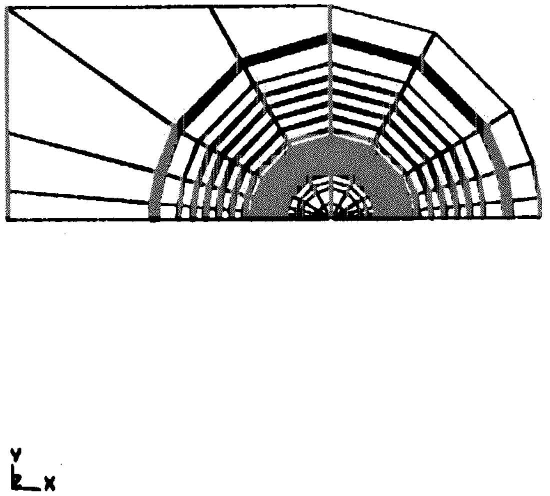
3,055 grids
2,440 solid elements
427 MPC equations
6 load cases

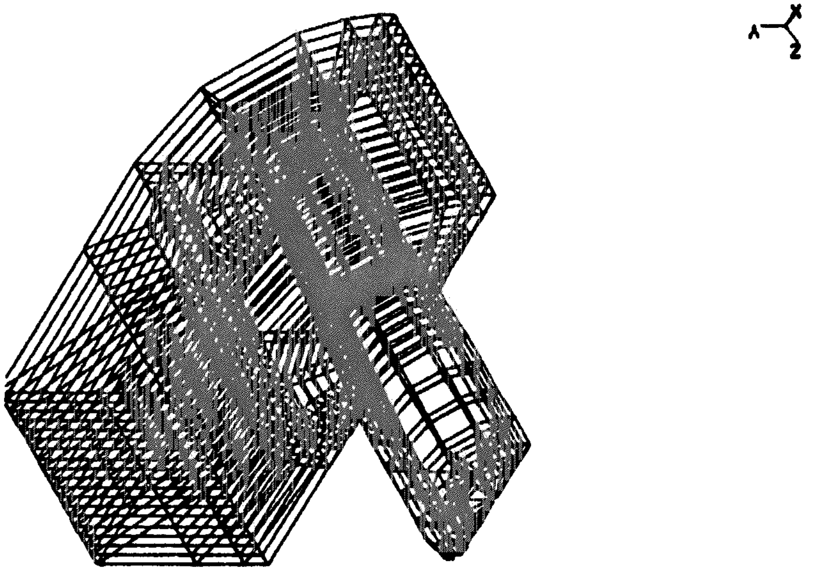
Decomposition:

Na=8,738 Crms=399

Output: Displacements, stresses, grid point force balance, etc.

Postprocess for stress contour plots





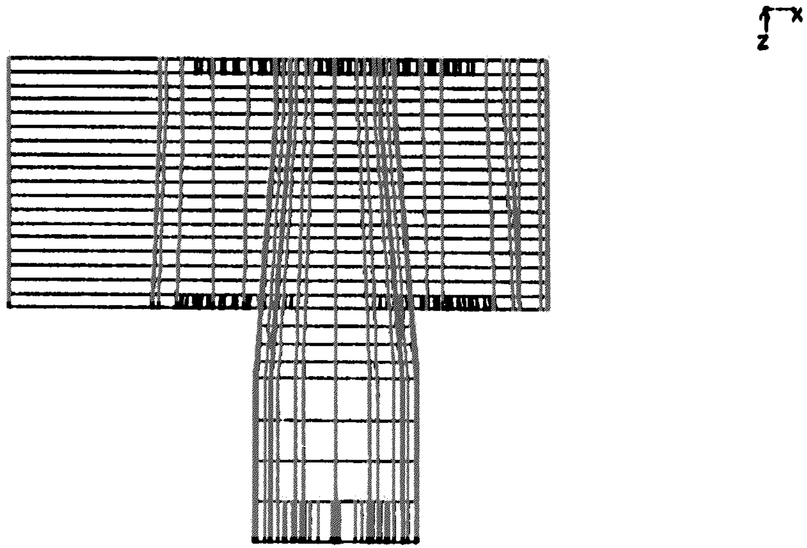
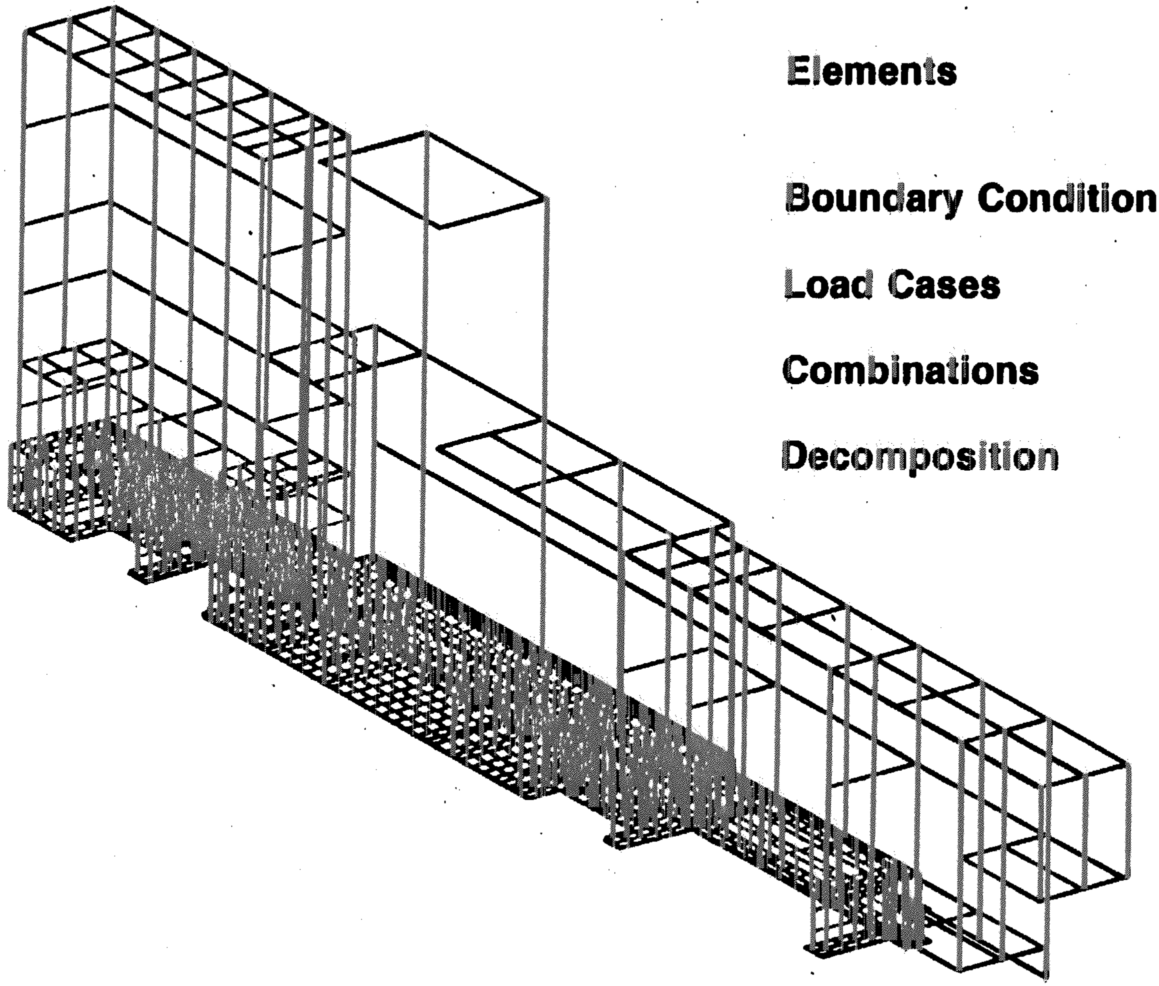


Figure 3.4

Gas Generator (Offshore Platform)

Support Frame, Static Analysis



Grids	5752	
Elements	232 QUAD 4, 1764 QUAD 8 27 TRIA6, 85 BAR	
Boundary Conditions	2	
Load Cases	4	
Combinations	4	
Decomposition	N_a	28,924
	C_{max}	1, 305
	C_{rms}	627

Undeformed Shape

Figure 3.5

PERFORMANCE COMPARISON

Model 1: Cylinder, Buckling

Resource	CYBER 760	CRAY-1S	Ratio CYB:CRAY
CP_{total}	221 seconds	124 seconds	1.8:1
% Matrix Oper's	89%	90%	
CM	170 K_8 wds	340 K_8 wds	
I/O	39.1 Mwds	46.2 Mwds	1:1.2
Residency	49.2 minutes	7.9 minutes	6.2:1

Model 2: Transporter, Eigenvalues

Resource	CYBER 760	CRAY-1S	Ratio CYB:CRAY
CP_{total}	374 seconds	264 seconds	1.4:1
% Matrix Oper's	85%	57%	
CM	300 K_8 wds	450 K_8 wds	
I/O	23.8 Mwds	27 Mwds	1:1.3
Residency	33.6 minutes	9.0 minutes	3.7:1

Model 3: Pin-Boss, Static Analysis

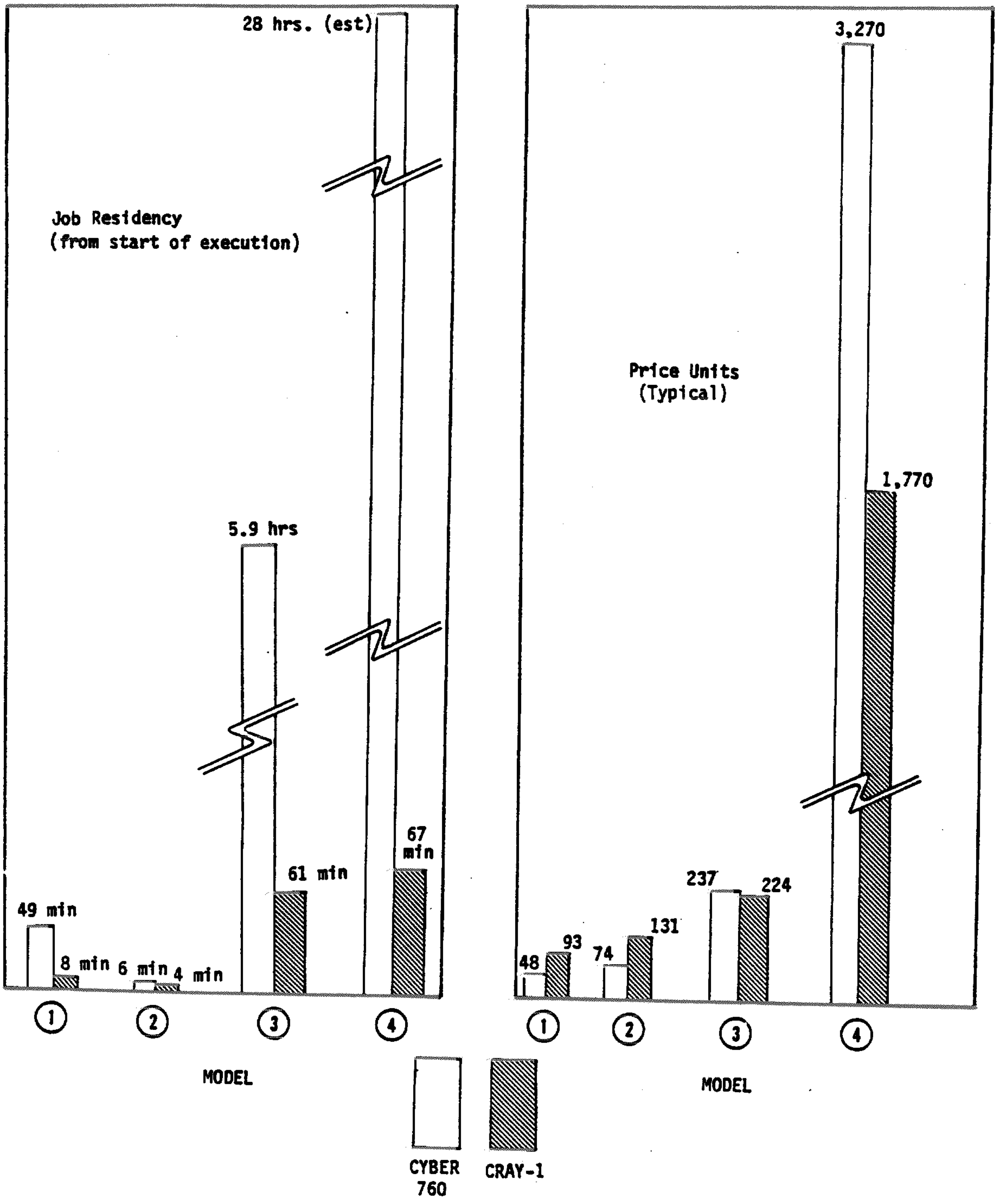
Resource	CYBER 760	CRAY-1S	Ratio CYB:CRAY
CP_{total}	1,017 seconds	338 seconds	3:1
% Matrix Oper's	66%	56%	
CM	377 K_8 wds	600 K_8 wds	
I/O	92.2 Mwds	72.6 Mwds	1.3:1
Residency	351.6 minutes	61.2 minutes	5.8:1

Model 4: Generator Frame, Statics

Resource	CYBER 760 (est.)	CRAY-1S	Ratio CYB:CRAY
CP_{total}	14,147 seconds	2,679 seconds	5.1:1
% Matrix Oper's	90%	68%	
CM	370 K_8 wds	1,400 K_8 wds	
I/O	1,458 Mwds	360 Mwds	4.1:1
Residency	28 hours	1.1 hours	26:1

Figure

PERFORMANCE SUMMARY



BOEING COMPUTER SERVICES COMPANY
A Division of The Boeing Company

SYMMETRIC DECOMPOSITION PERFORMANCE ON THE CRAY

MATRIX CHARACTERISTICS

MODEL		# 3	# 4
Matrix Size :	N	8,738	28,924
Active Columns :	C_{max}	483	1,304
	C_{rms}	399	627
Spill :	Groups	0	11
	Rows	0	1,389
Passive Columns :	Groups	6	0
	Max	297	0
	Ave	216	0
Theoretical Number of Decomposition Operations $= \frac{1}{2}C^2_{rms} N$		700×10^6	$5{,}690 \times 10^6$

CRAY DECOMPOSITION TIMES (seconds)

MODEL	#3	#4
Actual	61	777
Estimate used by NASTRAN $M \approx \left[0.05 + \frac{10}{C_{rms}}\right] \times 10^{-6}$ sec/operation	57	436
Estimate based on BCS CRAY experience $M=0.1\times10^{-6}$	70	569
Estimate based on nominal CRAY vector speed $M=0.033 \times 10^{-6}$ sec/operation	23	190