J461

MITRE-Bedford

NASA CR 160382

MTR-4722

JSC# 14781

NASASTRAN Processing at Johnson Space Center

(NASA-CR-160382) NASTRAN PROCESSING AT JOHNSON SPACE CENTER (Mitre Corp., Houston, Tex.) 57 p HC A04/MF A01 CSCL 20K

N80-12440

Unclas
G3/39 46168

F. M. Richards, Jr.

March 1979

DEC 1979
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

MITRE Technical Report
MTR-4722

NASTRAN Processing at Johnson Space Center

F. M. Richards, Jr.

March 1979

CONTRACT SPONSOR	NASA/JSC
CONTRACT NO.	F19628-79-C-0001 T5295F
PROJECT NO.	8470
DEPT.	D-72

THE MITRE CORPORATION
HOUSTON, TEXAS

Department Approval: Edwin S

MITRE Project Approval: Edwin S

ABSTRACT

In the future, space structures much larger than those of today are expected. The analyses of such structures will have computational requirements considerably greater than what is currently being realized. To gain some understanding of those requirements, an investigation into the processing characteristics of NASA/JSC's primary structural analysis computer program, NASTRAN, has been conducted. Based on the outcome of that investigation which resulted in a model sensitive to various NASTRAN host systems and workload scenarios, a set of recommendations based on cost/performance considerations has been proposed.

ACKNOWLEDGEMENT

The author gratefully acknowledges the assistance of NASA/JSC's Richard Rosencranz who willingly and enthusiastically shared his considerable knowledge of NASTRAN and its world.

TABLE OF CONTENTS

		Page
List of Illustrations		ix
List of Tables		ix
SECTION 1	INTRODUCTION	1
1.0	BACKGROUND	1
1.1	Objectives	1
1.2	Organization of Report	2
SECTION 2	STRUCTURAL ANALYSIS PROCESSING	3
2.0	INTRODUCTION	3
2.1	Matrix Mathematics	3
2.2	Execution Characteristics of NASTRAN	4
2.2.1	SDCOMP CPU Requirements	5
2.2.2	SDCOMP I/O Requirements	6
2.3	Reducing the Impact of NASTRAN on the Computer	8
2.3.1	Bandwidth Reduction	8
2.3.2	Substructuring	9
2.4	Host Computing Systems for NASTRAN	10
2.4.1	CDC CYBER 17X	11
2.4.2	DEC VAX-11/780	12
2.4.3	IBM 370/303X	12
2.4.4	UNIVAC 1110 and 1100/8X	12
2.5	Versions of NASTRAN	13
2.6	NASTRAN User Survey	14
SECTION 3	ADDRESSING LIMITATIONS OF THE U1110	17
3.0	BACKGROUND	17
3.1	NASTRAN's Problems on the U1110	17
SECTION 4	JSC NASTRAN WORKLOAD DEFINITION	19
4.0	INTRODUCTION	19
4.1	Present NASTRAN Workload	19
4.2	Future NASTRAN Workload	19
4.2.1	Small Jobs	20
4.2.2	Large Jobs	22
4.2.3	Full NASTRAN Workload	26

(continued)

		Page
SECTION 5	COST/PERFORMANCE ANALYSIS	29
5.0	INTRODUCTION	29
5.1	Acquisition of Computer Configuration for NASTRAN Processing at JSC	29
5.1.1	Hardware	29
5.1.2	Software	30
5.2	Service Bureau	31
SECTION 6	CONCLUSIONS	35
APPENDIX I	A SIMPLE STATICS PROBLEM	39
APPENDIX II	NASTRAN PERFORMANCE MODEL	41
REFERENCES		53
DISTRIBUTION LIST		55

LIST OF ILLUSTRATIONS

Figure Number		Page
2-1	Stiffness Matrix	5
2-2	SDCOMP: CPU Time vs Problem Size	6
2-3	SDCOMP: No-Spill Memory Space vs Structure Size	7
2-4	SDCOMP Spill Behavior: I/O vs Memory	8
6-1	Cost/Performance Characteristics of Candidate Systems	36
I-1	Diagram of Structure	39

LIST OF TABLES

Table Number		Page
2-I	Host Systems for NASTRAN	10
2-II	NASTRAN User Survey	15
4-I	Relative CPU Power	20
4-II	Effect of Small Jobs on Candidate Systems	21
4-III	Large NASTRAN Job (Non-Substructured)	23
4-IV	Large NASTRAN Job (Substructured)	25
4-V	Elapsed Time (Hours) for Full NASTRAN Workload	28
5-I	Cost/Performance for Non-Substructured Workload	31
5-II	Cost/Performance for Substructured Workload	33
II-I	SDCOMP Timing and Memory Characteristics	42

SECTION 1
INTRODUCTION

1.0 BACKGROUND

NASTRAN (for NASA STRUCTURAL ANALYSIS) is a general purpose system of computer software used for solving a wide variety of engineering analysis problems by means of the finite element displacement method. At Johnson Space Center (JSC) NASTRAN is used primarily for performing static and dynamic structural analyses of the Space Shuttle.* The host system for JSC's processing is the UNIVAC 1110 (U1110). Because of NASTRAN's current impact on the U1110, and because even larger space structures than the Shuttle loom as future prospects, MITRE has been directed by the Institutional Data Systems Division (IDSD) at JSC to address these future structures in terms of their effect on future computing facilities.

1.1 Objectives

The objectives of this task are to:

(1) Gain a sufficient understanding of the processing behavior of NASTRAN in order that cost versus performance tradeoffs can be conducted for various computing systems.

(2) Study candidate systems in order to determine the most cost effective way for supporting NASTRAN at JSC in the future.

(3) Study the addressing constraints of UNIVAC systems and the effect of these constraints on the NASTRAN user and on job turnaround time.

---

*For a more complete description of NASTRAN's capabilities, see [1].

1.2 Organization of Report

In Section 2 the processing implications of structural analysis are addressed in order to lay a foundation for further developments. Section 3 addresses the current NASTRAN situation on the U1110. Section 4 describes the current JSC NASTRAN workload and estimates the effect of workload growth due to increased structure size. A cost/performance analysis is conducted in Section 5, and Section 6 summarizes the findings of this study.

SECTION 2
STRUCTURAL ANALYSIS PROCESSING

2.0 INTRODUCTION

The mathematical formulation of a structural analysis typically involves the use of large matrices of high precision data. In the following paragraphs, some basic comments about matrices are presented in order to lay the foundation for later developments. To illustrate the translation of a structural analysis problem into its mathematical state, the solution of a simple statics problem is discussed in Appendix I.

2.1 Matrix Mathematics

A matrix can be thought of as a rectangular arrangement of numbers; for example,

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 0 & 6 & 2 \\ 1 & 0 & 3 \end{bmatrix} , \quad B = \begin{bmatrix} 1 & 0 & 2 & 3 \\ 8 & 3 & 3 & 1 \\ 0 & 0 & 5 & 2 \end{bmatrix}$$

The size of a matrix is its order. The order of A is 3 rows by 3 columns (or 3 by 3); the order of B is 3 by 4. A set of rules for matrix arithmetic exists and includes such common operations as addition, subtraction, multiplication, and "division" (multiplicative inverse). For a discussion of matrix algebra/arithmetic the reader is referred to [2].

In terms of system storage, a matrix of order n by m has n x m data elements. Because of the large number of arithmetic operations performed in structural analysis (discussed in Section 2.2), accuracy problems due to truncation and error propagation require that floating point data values consist of about 54 bits [3]. This requirement generally implies double precision on most systems.

In terms of classical mathematics, the addition of two matrices, each of order n by m, requires a number of arithmetic operations proportional to n x m; the multiplication of two matrices of orders n by p and p by r requires a number of operations proportional to n x p x r. To invert (perform the multiplicative inverse of) a matrix of order n by n, the number of operations is proportional to n^3. When one interprets the storage and computational requirements associated with large matrices whose orders may easily be in the tens of thousands for static analyses, it becomes readily apparent that the problems can be of enormous size. Hundreds of millions of computer words and billions of operations are suggested by the above. However, in the world of NASTRAN (and other structural analysis programs), the characteristics of these matrices are usually exploited to make the situation much more tenable than it might appear. Specifically, these characteristics have to do with the very large proportion of zero valued elements in all structural analysis matrices. In fact, it is not uncommon for the density (of nonzero elements) of the matrices to be less than 1%. In such a situation, the matrix is said to be sparse. NASTRAN employs the use of sparse matrix data structures, thereby eliminating the need to physically represent zeros; and in conjunction with these data structures, utilizes sparse matrix computational techniques. Even though an "elimination" of 99% is suggested, the typical structural analysis requires several matrix operations and the amount of storage and the number of computer operations can still be quite formidable.

2.2 Execution Characteristics of NASTRAN

In general, NASTRAN has a multitude of capabilities and at JSC it is used in several ways. However, since JSC's workload is dominated (80%) by static analyses which, for large structures, are dominated (75%) by the symmetric decomposition (SDCOMP) of the stiffness matrix, the execution characteristics to be discussed will be those of SDCOMP.

The stiffness matrix is a banded, symmetrical matrix. The quality of symmetry means the matrix remains unchanged when the rows and columns are interchanged. A banded matrix is one in which the nonzero terms are clustered about the diagonal. Figure 2-1 illustrates the stiffness matrix.

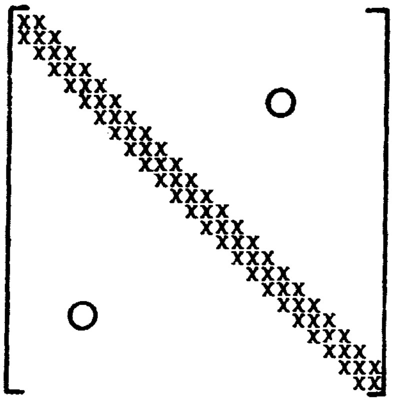


Figure 2-1 Stiffness Matrix

The decomposition of the stiffness matrix is directed to the solution of a system of linear equations. The symmetric decomposition algorithm used by NASTRAN works very efficiently with a memory scratch pad area whose size is dependent on the semi-bandwidth (or, in NASTRAN terminology, the number of active columns) of the matrix. This scratch pad can be of varying sizes; when the scratch pad is less than a certain size NASTRAN has to employ spill logic which, in effect, is a tradeoff between I/O activity and memory space.

2.2.1 SDCOMP CPU Requirements

The CPU requirements for symmetric decomposition can be approximated by the relationship, $T=0.5 \text{ x } m \text{ x } N \text{ x } \bar{C}^2$ [4], where m is a system-dependent, experimentally determined constant which represents the amount of CPU time required to make one pass through SDCOMP's tight multiply-add loop which dominates computation; N by N is the order of the matrix*, and $\overline{C}$ is the average number of active columns per row. Taking m to be 6 microseconds for the UNIVAC 1110 (see Appendix II), and plotting T with $\overline{C} = N/25$ (based on current workload characteristics), results in the curve shown in Figure 2-2.

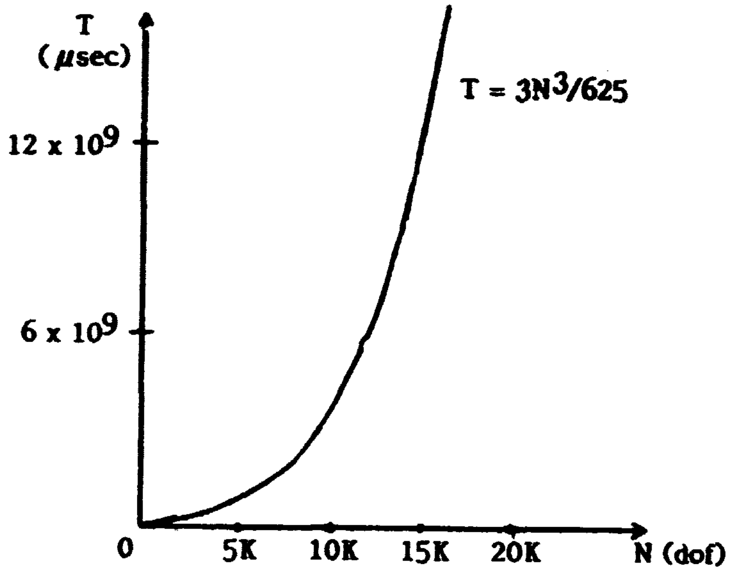


Figure 2-2 SDCOMP: CPU Time vs Problem Size

From Figure 2-2, it follows that if the structure size doubles, the CPU time required by SDCOMP increases by 700%; if the structure size triples, the CPU time required by SDCOMP increases by 2600%; and so on.

2.2.2 SDCOMP I/O Requirements

The decomposition of a symmetrical matrix can proceed very efficiently when a memory scratch pad area of approximately $C^2/2$ data values

---

*In structural analysis this is referred to as N degrees of freedom (dof).

is available [4], where C is the maximum number of active columns in any row. Assuming that NASTRAN's instruction space plus I/O buffers is 45000 words (see Appendix II), the no-spill, double precision memory requirements for the UNIVAC 1110 would be as shown in Figure 2-3 for the case where C = N/25.

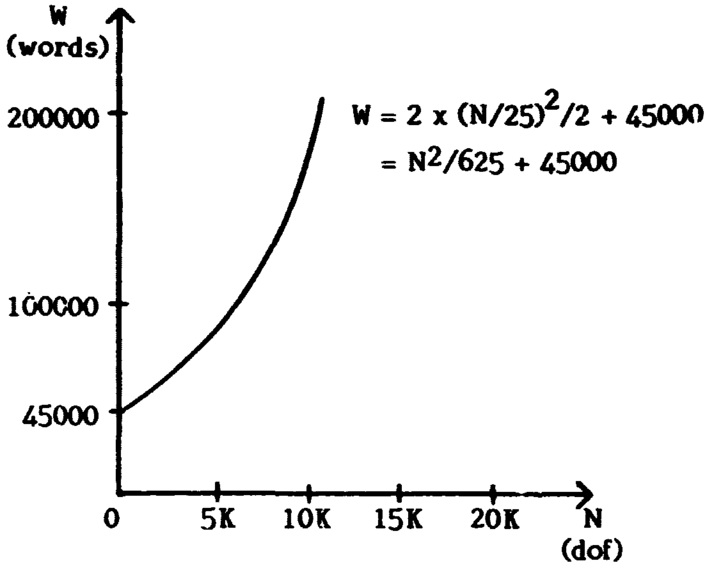


Figure 2-3 SDCOMP: No-Spill Memory Space vs. Structure Size

Before W exceeds the maximum address limit of the program, spill must be employed. For the above, spill becomes a necessity between 11000 dof and 12000 dof because of the U1110's limit of 262,000 words per program.

In the event that sufficient memory cannot be obtained to avoid spill, SDCOMP can proceed, in most cases, with whatever amount of memory is available. Again, the tradeoff is between I/O activity and memory space. The spill curve has the characteristics shown in Figure 2-4.

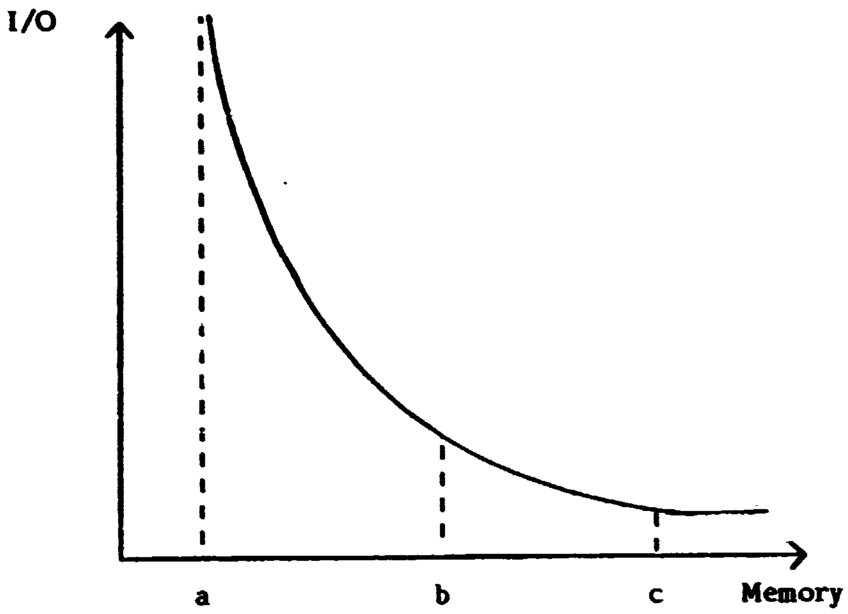


Figure 2-4 SDCOMP Spill Behavior: I/O vs Memory

The point 'a' represents the minimum amount of memory with which SDCOMP can proceed. The point 'b' is meant to suggest that there exists a system optimal amount of memory. And beyond point 'c' where spill is eliminated, excess memory would be dedicated to I/O buffers and might be less beneficial than the reduced I/O charges would justify.

2.3 Reducing the Impact of NASTRAN on the Computer

Based on the preceding discussion, it follows that the bandwidth of the stiffness matrix is the main determinant of system impact. The following paragraphs look at ways to reduce this impact.

2.3.1 Bandwidth Reduction

For the purposes of its mathematical description, the nodes of a structure can be numbered arbitrarily. For a given structure, all numbering schemes lead to the same size stiffness matrix and the same number of nonzero terms; however, different numbering schemes lead to different arrangements of nonzero terms. That arrangement which produces the minimum average bandwidth will result in the most favorable computing time. Two popular computer programs available to NASTRAN users for this purpose are BANDIT [5] and WAVEFRONT [6].

2.3.2 Substructuring

Applicable to both static and dynamic analyses, substructuring is the division of a large single structure into component structures. Each component (or substructure) is analyzed separately and finally all are reconnected for analysis of the total structure. In addition to providing an approach to analysis of a structure too large for efficient solution on the computer, substructuring offers the advantage of working with smaller and more manageable parts, each being independent of the others. [7]

The general procedures used in substructuring analysis can be divided into three distinct phases:

Phase I - NASTRAN analysis of each unique structure to produce, in matrix form, a description of each substructure in terms of a reduced set of degrees of freedom that, at a minimum, include the boundary degrees of freedom that connect to adjacent structures.

Phase II - Combination of matrices from Phase I with any additional structure that the user may wish to define, and the subsequent analysis of the "pseudo structure."

Phase III - The results from Phase I and Phase II are utilized to obtain detailed data from the individual substructures. One computer run is required for each substructure considered in Phase III.

The substructuring procedure described above involves the execution of 2k+1 computer runs, where k is the number of substructures.

2.4 Host Computing Systems for NASTRAN

In so far as NASTRAN host systems are concerned, there are several systems ranging from mini-computer to super computer which could be an adequate system for NASTRAN. There is currently a great deal of enthusiasm for the mini-computer [8], especially among engineering groups, and there is little doubt that super computers such as the CDC CYBER 203B (superceder of the STAR) and CRAY I, by virtue of their vector processing capabilities, would be powerful NASTRAN hosts. However, to qualify for consideration in this study, it was not whether the system had potential, but whether the system had an operational and supported version of NASTRAN. It was not considered reasonable for JSC to underwrite or perform the conversion of NASTRAN to another system. Such an activity would be very costly from the standpoint of both initial conversion and continuing maintenance/enhancement. In the past, NASA (Langley Research Center) has sponsored several studies [9,10,11,12] to determine the feasibility of converting NASTRAN to various super computers. None of the studies has resulted in NASA's funding such a conversion. One of the problems in justifying such a conversion and continuing maintenance is the cost of the effort versus the relatively small number of prospective host systems. (To give better perspective to this point, NASTRAN consists of between 400,000 and 500,000 FORTRAN statements.)

Table 2-I shows the host computing systems for NASTRAN considered in this study.

Table 2-I Host Systems for NASTRAN

VENDOR	COMPUTER SERIES
CDC	CYBER 17X: 172 → 176
DEC	VAX - 11/780
IBM	370/303X: 3031, 3032, 3033
UNIVAC	1110; 1100/8X: 1100/80→1100/84

At the present time the DEC VAX-11/780 version is under development and is not available for release. The VAX is represented in this study since it is expected that a version of VAX NASTRAN will be available from the MacNeal-Schwendler Corporation (MSC) during Spring 1979*. It should be noted that in the case of the IBM version, those systems which can execute IBM-compatible software would extend the vendor range of Table 2-I. The following paragraphs provide a brief description of the systems in Table 2-I. For more detailed descriptions see [13] and/or product information available from the vendors.

2.4.1 CDC CYBER 17X

The CYBER 170-series which spans the medium-to-large class of computers is the latest evolution of an architecture that began with the CDC 6600. The 171 and 172 can be either unit processors or dual processors; the 174 is a dual processor 173, while the 175 and 176 are unit processors. For all of these systems the architectural philosophy is to perform computation with the central processor(s) (CP) and to distribute the slower, lower-level functions such as input/output and system control among multiple peripheral processors. The central memory (CM) of these systems is made up of 60-bit words and can consist of as many as 262,144 words of which no more than 131,072 can be used by a single FORTRAN program. Optionally available is an extended core storage (ECS) which can consist of up to 2,097,152 words. On the 176 a minimum ECS complement of 524,288 words is required. ECS is different from CM in that it cannot be executed from directly; in general, the system uses ECS as an ultra-high-speed I/O device. The 176 can use ECS in a more flexible way than can the other systems, but still ECS on the 176 is nowhere near a full-fledged extension of CM. From 171 through 174, the CPs consist of a single unified arithmetic unit that executes all instructions. The 175 and 176 CPs employ nine independent functional units

---

*Concurrently with and independent of MSC, another version of VAX NASTRAN is being developed by the Goddard Space Flight Center (GSFC). This version is scheduled for completion during FY80.

that perform specialized operations thereby allowing a high degree of concurrency. Also, on the CPs of the 175 and 176 there is an instruction stack similar to a cache memory which can hold between 14 and 48 instructions. For loops which can fit entirely within the stack, very good performance can result.

2.4.2 DEC VAX-11/780

The DEC VAX-11/780, which is an extension of the PDP-11 family of computers, is generally regarded as a "supermini". The VAX/VMS virtual memory operating system provides a virtual address space of over 4 billion bytes (byte = 8 bits). The real memory complement of the VAX can presently be as large as 8 million bytes. The processor includes an 8K byte cache memory.

2.4.3 IBM 370/303X

The IBM 370/303X family of computers includes the 3031, 3032, and 3033. These virtual memory systems can accommodate programs whose individual virtual sizes can be as large as 16 million bytes (byte = 8 bits) when operating under MVS (Multiple Virtual Storages). The real memory complement of the 3031 and 3032 can be as large as 6 million bytes, whereas the real memory size of the 3033 can go up to 16 million bytes. For the 3031 and 3032, a buffer storage of 32,768 bytes is available. On a unit processor 3033 this buffer storage can be as large as 65,536 bytes.

2.4.4 UNIVAC 1110 and 1100/8X

Although the 1110 is a discontinued system as far as the current UNIVAC product line is concerned, it was included in this study because it will remain as part of the IDSD Central Computing Facility (CCF) for the foreseeable future. The CCF U1110 is a 2x2 configuration (i.e., 2 CPUs (CAUs) and 2 I/O processors) with 524,288 words of memory (262,144 primary and 262,144 extended.) Due to an 18-bit address limitation, a single program can, at a given instant, access no more than 262,144 words (262K words). While access to more than 262K words is possible through a system feature called program banking, the maximum instantaneous address space is still 262K words.

The UNIVAC 1100/8X-series (including /80 through /84) represents UNIVAC's latest offering in the large system category. Except in the case of /80 which is a unit processor, the x (of 8x) signifies the number of CPUs on the system. An optionally available Scientific Accelerator Module (SAM) significantly increases the execution speeds of floating point arithmetic operations. One of the most distinguishable differences between the /8X-series and its predecessors (1106, 1108 and 1110) is the use of a high-speed buffer storage in the Storage Interface Unit (SIU) connecting a large backing store of moderate speed. The maximum buffer storage size is 32,768 words while the backing store can be as large as 16 million words. However, as with the U1110, the single program can, at a given instant, access no more than 262,144 words.

2.5 Versions of NASTRAN

NASTRAN's original development was sponsored by NASA's Goddard Space Flight Center (GSFC) during the late 1960's. Once completed, NASTRAN's maintenance and enhancement became the responsibility of the NASTRAN System Management Office at the Langley Research Center. During early 1979, Langley will cease this function to be assumed by COSMIC (NASA's COmputer Software Management and Information Center located at the University of Georgia) which for several years, has had the role of distributing NASTRAN. Other versions of NASTRAN have been spawned from the earlier NASA version. Perhaps the best known of these versions is that marketed by the MacNeal-Schwendler Corporation (MSC) which was part of the original development team. Universal Analytics Incorporated (UAI) market feature and performance enhancement packages for the COSMIC version. JSC is currently using the COSMIC version with local performance modifications resulting from an earlier MITRE study [14].

It has not been a goal of this study to compare the different versions of NASTRAN. However, a task within IDSD is currently concerned with selecting the most cost effective version of NASTRAN for the U1110.

2.6 NASTRAN User Survey

To determine what the NASTRAN world outside JSC was like, a limited survey of other NASTRAN user groups was conducted. The intent was to find out what computing combination (computer system and version of NASTRAN) was used by other engineering groups which are generally involved with structures similar in size to those studied at JSC. Table 2-11 presents the findings of that survey. It is of some interest to note the absence of UNIVAC systems and the predominance of MSC NASTRAN among the private industry component of those surveyed.

Table 2-11 NASTRAN User Survey

USER		NASTRAN		COMPUTER SYSTEM(s)		
		COSMIC*	MSC	CDC	IBM	UNIVAC
PRIVATE INDUSTRY	Boeing Company		X	CYBER 175	370/168	
	Ford Motor Company		X	CYBER 175		
	General Motors		X		370/168	
	General Dynamics		X	CYBER 172		
	Grumman Aerospace Corporation	X		CYBER 72	370/168	
	Lockheed - California	X			3033	
	Martin Marietta		X	CYBER 172	370/xxx	
	McDonnell Douglas		X	CYBER 175	370/168	
	Northrop		X		3033	
	Rockwell	X		CYBER 176	3033	
NASA	Ames Research Center		X	7600		
	Dryden Flight Research Center	X		6600		
	Goddard Space Flight Center	X	X		360/95	
	Jet Propulsion Laboratory	X				1108
	Johnson Space Center	X				1110
	Langley Research Center	X		CYBER 175		
	Lewis Research Center	X				1110
	Marshall Space Flight Center	X				1108

*Includes derivatives of COSMIC version.

SECTION 3
ADDRESSING LIMITATIONS ON THE U1110

3.0 BACKGROUND

At the outset of this task the poor turnaround time typically realized by large NASTRAN jobs at JSC was blamed to a large extent on the addressing limitations of the U1110. The purpose of this section is to better explain those problems. As background to this explanation, it is useful to realize that many of the programs executed on the U1110 are quite large in size, and that the aggregate workload is considered heavy.

3.1 NASTRAN's Problems on the U1110

Since the memory requirements of current large NASTRAN jobs are approximately half of the 262,144 word limit allowing for moderate spill, it follows that addressing limitations are not responsible for any current problems. These problems appear to be more related to the competition for memory which is a function of (a) the aggregate memory requirements of the active job set, (b) the amount of memory available to the active job set, and possibly (c) the scheduling philosophy of the operating system, EXEC 8. This competition results in a large amount (often days) of system wall clock time for the job. Since the combined CPU and I/O requirements of a large NASTRAN job are currently less than three (3) SUP[*] hours, the greatest part of that job's life is spent waiting. If a job is at some stage of processing short of completion, and if the system should go down for any reason (e.g., system crash, preventative maintenance, or block time), whatever work had been done in behalf of the job is lost and will have to be repeated through a restart of the job. NASTRAN's users say that restarts are a common occurrence. To test the premise that a higher priority would "push" the job through the system more quickly, a fairly large job (100K words of memory, 80 SUP minutes)

---

[*] Standard Unit of Processing. Used for resource accounting.

PAGE 16 INTENTIONALLY BLANK

was run as deadline batch* with a goal completion time of the then current time. The experiment took place on a normally loaded system and completed in about three hours of wall clock time. While no assessment can be made of the adverse effect, if any, on the other jobs in the system, it does seem reasonable to suspect that a higher priority for NASTRAN would not only improve the turnaround of large NASTRAN jobs, but would also reduce the wasted computation due to restarts, and generally contribute to greater overall system productivity. For further details on deadlining to improve system productivity, the reader is referred to [15] which discusses a UNIVAC 1108-related performance analysis.

---

*Deadline batch is a type of batch run whose priority is dynamically adjusted so as to realize a preset job completion time. The priority is a function of the run card-specified maximum SUPs, the current time, and the specified completion time.

SECTION 4
JSC NASTRAN WORKLOAD DEFINITION

4.0 INTRODUCTION

To accomplish the planning aspects of this study, it was necessary to identify future NASTRAN processing requirements. Since little was known about future space structures except that they would be significantly larger than the Space Shuttle, arbitrarily defined workloads had to be used. The following paragraphs describe the workload modeling approach used in this study and the processing expectations for these workloads on the prospective host systems.

4.1 Present NASTRAN Workload

Presently, the NASTRAN workload on the U1110 can be described as a set of small NASTRAN jobs and large NASTRAN jobs. Each of the small jobs consumes approximately 20 SUP minutes at a CPU:IO ratio of about 3:1 and requires about 65,000 words of memory. Approximately 20 to 25 small jobs per week are submitted. The large job class consists of 2 to 3 jobs per week. Since static analyses account for 80% of these jobs, the large job class can be represented by static analysis jobs whose individual characteristics involve a stiffness matrix of 10,000 dof and 400 active columns.*

4.2 Future NASTRAN Workload

If the structure being studied increases in size by a factor of m, it is expected that the small jobs will not increase in size, but rather, will increase in number by a factor of m. In the case of the large jobs, their quantity (2 to 3 per week) will remain fixed, but the size of the stiffness

---

*The dynamic analyses accounting for the other 20% have similar execution characteristics, in terms of memory requirements and SUP consumption, to those of the static analyses.

matrix and the number of active columns will be affected by the factor of m. This treatment of the active columns is felt to be conservative.

Thus, for an eventual structure whose size is m times the current size, the workload will be as follows:

Small Jobs: 20 x m to 25 x m per week, each requiring

- 65,000 words of memory
- 15 U1110 CPU minutes (3/4 of 20)
- 5 U1110 I/O minutes (1/4 of 20)

Large Jobs: 2 to 3 per week, each involving a stiffness matrix of m x 10,000 dof and m x 400 active columns.

4.2.1 Small Jobs

In order to evaluate the impact of small jobs on host systems other than the U1110, the following assumptions were made:

(1) An amount of memory equivalent to 65,000 U1110 words would be required,

(2) The I/O power on the other systems being considered would be the same as the I/O power of the U1110, and

(3) The CPU requirements on the other systems would be related to the U1110 in terms of the assumed power ratio ranges shown in Table 4-I.

Table 4-I Relative CPU Power

SYSTEM		CPU POWER RANGE
CDC	CYBER 173	0.8 to 1.3
	CYBER 175	2.4 to 5.3
DEC	VAX-11/780	0.3 to 0.5
IBM	3031	0.8 to 1.2
	3032	2.1 to 3.1
	3033	3.6 to 5.4
UNIVAC	1110	1
	1110/80	1 to 1.2
	1100/80 + SAM	1 to 1.8

For a system whose CPU power range is a to b [abbreviated as (a,b)], the CPU requirements for processing the small job subset of the current NASTRAN workload are

$$(15 \text{ U1110 CPU Minutes}) \times (1/b,\ 1/a) \times (20,\ 25)$$

or

$$20 \times 15/b \text{ to } 25 \times 15/a = 300/b \text{ to } 375/a \text{ CPU Minutes}$$

Table 4-II summarizes the CPU and I/O requirements of the small job subset of the NASTRAN workload for those systems considered in this study.

Table 4-II. Effect of Small Jobs on Candidate Systems

		Small Jobs — Weekly Population	
		CPU (Min)	I/O (Min)
CDC	CYBER 173	231 to 469	100 to 125
	CYBER 175	57 to 156	100 to 125
DEC	VAX-11/780	600 to 1250	100 to 125
IBM	3031	250 to 469	100 to 125
	3032	97 to 179	100 to 125
	3033	56 to 104	100 to 125
UNIVAC	1110	300 to 375	100 to 125
	1100/80	250 to 375	100 to 125
	1100/80 + SAM	167 to 375	100 to 125

4.2.2 Large Jobs

Because of the nonlinear effect (see Section 2.2) of large NASTRAN job processing with respect to a change in structure size, it was necessary to develop a mathematical model which could predict this nonlinear behavior. This model, which is more specifically a model of NASTRAN's symmetric decomposition activity, is described in Appendix II. Table 4-III shows the model's predictions for single, non-substructured jobs varying in size from 10000 dof to 60000 dof. The estimates, expressed in terms of memory, CPU and I/O requirements, are given for the considered host systems which are modeled as single processor systems essentially dedicated to the processing of the indicated NASTRAN jobs. The amount of memory shown in the table was selected to be the smaller of: a) the system limit for a single job and b) the smallest amount of memory greater than the spill point for the problem. For the DEC and IBM systems which utilize the concept of virtual memory, it was assumed that the problem was executed in a totally "real" region.* Each of these systems, except for the IBM 3033, was assumed to have available for a single program, the maximum amount of system memory less 2 Mbytes (1 Mbyte = million bytes) which would be available to the operating system and/or other programs. Although it can be configured as a 16 Mbyte system, the IBM 3033 was modeled as an 8 Mbyte system.

Except for showing system-specific data; Table 4-III is essentially a replay of Figures 2-2, 2-3, and 2-4. Noteworthy is the magnitude of processing required for the larger NASTRAN jobs and the spill phenomena. None of the systems are affected by spill at the 10K dof level. However, for the CDC and UNIVAC systems, spill comes into play between the 10K dof and 20K dof levels. By 30K dof, none of the systems processed the problem without spill. As reflected by the CPU:I/O ratios, the advantages of the more powerful systems are reduced as the problem size increases.

---

*Because of the nature of the SDCOMP algorithm [3], it is felt that this manner of execution, realizable in practice, shows the virtual memory system in its best light.

Table 4-111. Large NASTRAN Job (Non-Substructured)

Structure Size (dof) / System		10K			20K			30K			60K		
		Memory*	CPU	I/O	Memory	CPU	I/O	Memory	CPU	I/O	Memory	CPU	I/O
CDC	CYBER 173	116K	2.4	.2	130K	19.4	4.9	130K	66.1	24.0	130K	547	386
	CYBER 175	116K	.2	.2	130K	1.5	4.9	130K	5.4	24.0	130K	49	386
DEC	VAX-11/780	893K	3.9	.2	2837K	30.9	.9	6000K	104.4	3.6	6000K	839	52
IBM	3031	893K	2.4	.2	2837K	19.0	.9	4000K	64.6	5.3	4000K	521	73
	3032	893K	.9	.2	2837K	7.1	.9	4000K	24.3	5.3	4000K	196	73
	3033	893K	.5	.2	2837K	4.3	.9	6000K	14.5	3.4	6000K	117	49
UNIVAC	1110	206K	1.8	.3	260K	14.6	5.1	260K	50.0	24.0	260K	416	371
	1100/80	206K	1.5	.3	260K	12.2	5.1	260K	41.8	24.0	260K	349	371
	1100/80 + SAM	206K	1.2	.3	260K	9.7	5.1	260K	33.1	24.0	260K	274	371

* Memory space is expressed in words (CDC & UNIVAC) and bytes (DEC & IBM).
CPU & I/O are expressed in hours per week.

Section 2.3.2 introduced the technique of substructuring as an alternative to the monolithic model. To model the effect of substructuring on the various large job sizes studied, the following assumptions* were made:

a) Substructuring would eliminate all spill I/O,

b) All jobs would be substructured to 5000 dof jobs resulting in k (of 2k + 1 jobs, as discussed in paragraph 2.3.2) being equal to (dof of structure size) ÷ 5000,

c) The aggregate CPU requirements for the 2k + 1 jobs would be 80% that of what the non-substructured model required, and

d) The aggregate I/O requirements for the 2k + 1 jobs would be the total non-spill I/O of the non-substructured job plus two (2) minutes of I/O for each of the 2k + 1 submodels.

Table 4-IV shows the expected system requirements based on the above assumptions. Even if assumptions (c) and (d) are disregarded because of their softness, the significant reduction in I/O of Table 4-IV over that of Table 4-III is due mostly to the elimination of spill I/O. Also, not to be overlooked is that system memory has ceased to be a critical performance factor. This judgement is based on the relatively small amount of memory space required by a 5000 dof job and the relatively high CPU:I/O ratios suggesting that it will take only a few such jobs to saturate the CPU and thus effectively, the system.

---

[*] Based on the arbitrarily selected submodel size of (b), (a) seems to be a reasonable expectation; (c) and (d) were based on discussions with NASTRAN users and, in the absence of actual experimentation, should be regarded as soft.

Table 4-IV. Large NASTRAN Job (Substructured)

System	Structure Size (dof)	10K		20K		30K		60K	
		CPU*	I/O	CPU	I/O	CPU	I/O	CPU	I/O
CDC	CYBER 173	1.9	.4	15.2	1.2	51.3	2.3	410	8
	CYBER 175	.14	.4	1.1	1.2	3.9	2.3	31	8
DEC	VAX-11/780	3.1	.4	24.7	1.2	83.4	2.5	666	9
IBM	3031	1.9	.4	15.2	1.2	51.3	2.3	410	8
	3032	.7	.4	5.7	1.2	19.3	2.3	154	8
	3033	.4	.4	3.4	1.2	11.6	2.3	92	8
UNIVAC	1110	1.4	.4	11.4	1.3	38.4	2.7	307	10
	1100/80	1.2	.4	9.5	1.3	32.1	2.7	256	10
	1100/80 + SAM	1.0	.4	7.6	1.3	25.6	2.7	205	10

*CPU and I/O are expressed in hours per week.

4.2.3 Full NASTRAN Workload

In the above paragraphs, the separate components of the NASTRAN workload were discussed for different workload scenarios on the various host systems. The purpose of the following paragraphs is to look at the system impact of the full NASTRAN workload for different structure sizes. For each of the considered systems, the amount of elapsed time for processing the full workload (20 m to 25 m small jobs and 2 to 3 10,000 x m dof large jobs) in a multiprogramming environment will be determined. A multiprogramming analysis is generally difficult to accomplish in a simplistic fashion because of the resource scheduling which takes place by the operating system. Still such an approach is being taken since to do otherwise would be inconsistent with the level of detail available with the workload data.

If the active job mix is taken to be such that no more than a single large job and a single small job are active at a given instant, then the number of system hours, SYSTIME, for the full NASTRAN workload can be approximated as:

$$\mathrm{SYSTIME} = \mathrm{MAX}(\mathrm{TLJCPU},\mathrm{TSJIOT}) + \mathrm{MAX}(\mathrm{TSJCPU},\mathrm{TLJIOT}),$$

where TLJCPU is the total amount of CPU time required by the large job population, TSJIOT is the total amount of I/O time required by the small jobs, TSJCPU is the total amount of CPU time required by the small jobs, and TLJIOT is the total amount of I/O time required by the large jobs. The assumptions are that the system is fully dedicated to NASTRAN processing until the NASTRAN workload is completed and that a sufficient backlog of NASTRAN jobs will be available so that a large job and small job will always be executing while representatives of either class remain to be processed.

For non-substructuring, the assumption that only a single large job and a single small job will be active is reasonable when one considers that memory availability will essentially force this condition. In the case of substructuring, where memory would not force this restriction, SYSTIME, as estimated above, should be regarded as slightly conservative since a higher degree of multiprogramming could be accomplished with the smaller components of the large job class. In this case, SYSTIME would be closer to the sum of TLJCPU and TSJCPU resulting in only a relatively small difference when compared to the SYSTIME computation used.

Being conservative with respect to both the large and small job subpopulations and taking 3 large jobs and 25 x m small jobs, as well as the slower end of the power range in Table 4-I, produces the results of Table 4-V. Of particular interest is the processing penalty incurred by not employing substructuring; further interpretation is deferred to Section 5 where the same data is looked at in an operational context.

Table 4-V Elapsed Time (Hours) for Full NASTRAN Workload

System	Structure Size (dof)	10K		20K		30K		60K	
		w/o sub-structuring	sub-structuring	w/o sub-structuring	sub-structuring	w/o sub-structuring	sub-structuring	w/o sub-structuring	sub-structuring
CDC	CYBER 173	15.0	13.5	73.8	61.2	270.3	177.4	2799	1277
	CYBER 175	4.7	4.7	19.2	9.4	88.2	19.5	1305	117
DEC	VAX-11/780	32.5	30.1	134.4	115.8	375.7	312.7	2673	2123
IBM	3031	15.0	13.5	72.6	61.2	217.3	177.4	1782	1277
	3032	5.7	5.1	27.3	23.1	88.8	66.9	807	486
	3033	3.8	3.8	16.4	13.8	53.7	41.7	498	300
UNIVAC	1110	11.7	10.5	59.1	46.7	222.0	134.0	2361	959
	1100/80	10.8	9.9	51.9	41.0	197.4	115.1	2160	806
	1100/80 + SAM	9.9	9.3	44.4	35.3	171.3	95.6	1935	653

SECTION 5
COST/PERFORMANCE ANALYSIS

5.0 INTRODUCTION

From Table 4-V, it is apparent which are the most powerful systems for doing NASTRAN processing. It is now important to look at that power in terms of cost and to compute the more meaningful system selection criterion of cost/performance. Because the U1110 is not a current product offering, it will not be a part of the analysis in this section.

5.1 Acquisition of Computer Configuration for NASTRAN Processing at JSC

A computer obtained for local use can be acquired by either of two means - lease or purchase. The following analysis assumes that the system(s) will be purchased.

Based on the number of hours expected of each system for the different workload scenarios, it is a simple matter to compute the system's cost effectiveness when the cost of the system is known. In order to arrive at the needed cost information the following configuration has been selected. This near-minimal configuration has resulted from the need to compare the wide range of computers considered, and is not necessarily the recommended configuration. Final configuration selection should take into consideration any other needs expected to be satisfied by the system.

5.1.1 Hardware

a) Single CPU

b) Central Memory

- CDC: 262K words
- DEC: 8 Mbytes
- IBM: 6 Mbytes (3031, 3032)
  8 Mbytes (3033)
- UNIVAC: 524K words

c) Two(2) 9-track magnetic tape drives

d) Five Hundred (500) Mbytes of online disk storage

e) One (1) 600 LPM line printer

f) One (1) card reader

g) Four (4) CRT terminals

5.1.2 Software

a) Standard operating system generated for a low level of multiprogramming

b) FORTRAN compiler and runtime library

c) Basic system utilities

d) NASTRAN

The monthly costs of the various systems, configured as above and amortized over seven (7) years, are shown in Table 5-1 which corresponds to the non-substructured approach to problem solution. The costs include only hardware, software, and their maintenance. Operational costs such as personnel, cost of supplies, energy, floor space, etc. are assumed to be equal for all configurations and therefore are not a part of this comparative analysis. A system week of 120 hours (instead of 168 hours) is used to allow for operating system overhead, preventative maintenance, and quality of service. In some cases, as indicated by %SYSTEM, multiple systems can result.

Some points to be made about Table 5-1 are:

- The better cost/performances of the DEC and IBM systems over those of the CDC and UNIVAC systems are due mainly to program memory considerations.
- At the point where %SYSTEM exceeds 300, that system is unable to handle a week's work since a large, nonsubstructured job cannot be split among several systems.

Table 5-I. Cost/Performance for Non-Substructured Workload

	SYSTEM	MONTHLY COST ($)	STRUCTURE SIZE (DOF)							
			10K		20K		30K		60K	
			%SYSTEM	CPF	%SYSTEM	CPF	%SYSTEM	CPF	%SYSTEM	CPF
CDC	CYBER 173	32539	12.5	41	61.5	200	225.2	733	2331	7585
	CYBER 175	49892	3.9	19	16.2	81	73.6	367	1086	5420
DEC	VAX-11/780	11916	27.0	32	112.0	133	313.1	373	2230	2657
IBM	3031	28019	12.5	35	60.7	170	181.2	508	1485	4162
	3032	42258	4.7	20	22.9	97	73.8	312	672	2839
	3033	61143	3.2	19	13.6	83	44.6	273	413	2528
UNIVAC	1100/80	26756	8.9	24	43.1	115	164.4	440	1800	4817
	1100/80 + SAM	28410	8.2	23	36.8	105	142.6	405	1611	4577

%SYSTEM = (No. of System Hours) / 120

Cost Performance Factor: CPF = (%SYSTEM) x (Monthly Cost) / 100

- Excess capacities result when %SYSTEM is rounded up to a whole number.

Subject to the same costing considerations as above, Table 5-II corresponds to the substructured approach to problem solution. The monthly costs in that table have changed in some cases to reflect the need for less memory. Noteworthy points about Table 5-II are:

- The clear supremacy of the CYBER 175 over all above competition.
- A week's work at any workload level can now be handled provided enough systems are available. This is due to the ability to now split the large job according to its substructured components and simultaneously compute on several systems.
- Again, excess capacities will result when %SYSTEM is rounded up to a whole number.

5.2 Service Bureau

Although it has been assumed in the above cost analysis that a system would be acquired for local use at JSC, there also exists the service bureau option. With a service bureau, the user is more immune to the job turnaround problems which can affect the user of local facilities. And commensurate with this higher quality of service is a higher cost. Because so many variables come into play - accounting algorithms, priority of service, charges for permanent file space, terminal connect time, communications costs, remote job entry station(s), NASTRAN surcharge, quantity discounts, etc. - it is difficult at this level of analysis to judge the cost of using the service bureau in lieu of the local computer. Should a service bureau cost be estimated, it is important to recall, for purposes of comparison, that Tables 5-I and 5-II do not include operational costs which are included in the service bureau charges.

Table 5-11. Cost/Performance For Substructured Workload

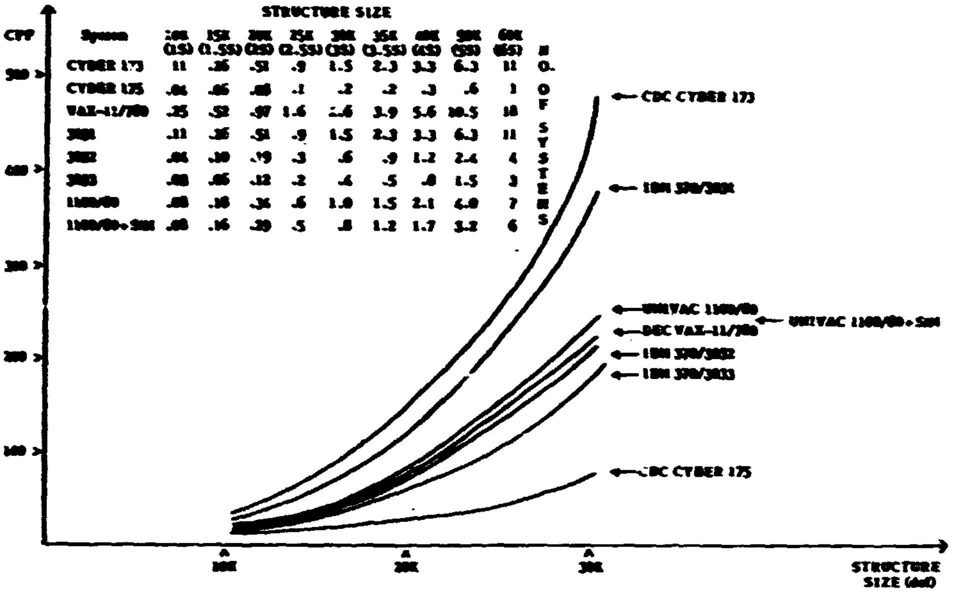
SYSTEM		MONTHLY COST ($)	STRUCTURE SIZE (DOF)							
			10K		20K		30K		60K	
			%SYSTEM	CPF	%SYSTEM	CPF	%SYSTEM	CPF	%SYSTEM	CPF
CDC	CYBER 173	32539	11.3	37	51.0	166	147.7	481	1064	3461
	CYBER 175	49892	3.9	19	7.8	39	16.2	81	98	489
DEC	VAX-11/780	8733	25.1	22	96.5	84	260.5	227	1770	1545
IBM	3031	25715	11.3	29	51.1	132	147.9	380	1064	2737
	3032	40111	4.3	17	19.3	77	55.6	223	406	1627
	3033	56912	3.2	18	11.5	65	34.7	198	252	1433
UNIVAC	1100/80	26756	8.2	22	34.2	91	95.7	256	672	1797
	1100/80 + SAM	28410	7.6	22	29.4	84	79.7	226	543	1544

%SYSTEM = (No. of System Hours) / 120
Cost Performance Factor: CPF = (%SYSTEM) x (Monthly Cost) / 100

SECTION 6
CONCLUSIONS

This study has delved into the many facets of the NASTRAN situation at Johnson Space Center. The current situation, as well as those which might come about in the future, have been addressed. The most immediate conclusions to be drawn from this study are that:

- Any growth in structure size will have increasingly nonlinear implications in terms of the computer processing requirements on any NASTRAN host system.
- At some workload level, depending on the system, NASTRAN model substructuring has a practical value. For the CDC and UNIVAC systems, this level occurs between 10K and 20K dof and for the DEC and IBM systems, before 30K dof.
- The NASTRAN user can improve his own situation, as well as that of the system, by using bandwidth reduction techniques such as BANDIT and WAVEFRONT.
- While the UNIVAC 1110 does have program addressing limitations, these limitations are not responsible for any current NASTRAN problems on the U1110 at JSC, nor should such become a problem if substructuring is employed for problem sizes involving more than 11,000 dof (see Figure 2-3.)
- Except for those cases where inordinate I/O (i.e., heavy spill) prevails, the faster mainframes, as shown in Figure 6-1, offer the best cost/performance when measured in terms of hardware, software and maintenance, and assuming that substructuring will be employed.



STRUCTURE SIZE

System	10K (1S)	15K (1.5S)	20K (2S)	25K (2.5S)	30K (3S)	35K (3.5S)	40K (4S)	50K (5S)	60K (6S)
CYBER 173	11	.26	.51	.9	1.5	2.3	3.3	6.3	11
CYBER 175	.04	.05	.08	.1	.2	.2	.3	.6	1
VAX-11/780	.25	.52	.97	1.6	2.6	3.9	5.6	10.5	18
3081	.11	.26	.51	.9	1.5	2.3	3.3	6.3	11
3082	.04	.10	.19	.3	.6	.9	1.2	2.4	4
3083	.03	.05	.12	.2	.4	.5	.8	1.5	3
1100/80	.08	.18	.36	.6	1.0	1.5	2.1	4.0	7
1100/80+SIM	.08	.16	.29	.5	.8	1.2	1.7	3.2	6

OF SYSTEMS

Figure 6-1. Cost/Performance Characteristics of Candidate Systems

The NASTRAN workload associated with the current structure sizes (1S) does not appear to pose a serious problem to the CCF since this workload is equivalent to less than 20 SUP hours per week. This opinion applies to the U1110 (assuming a solution to the present turnaround difficulties can be found) as well as to the replacement system expected during FY82. In the unlikely event that a version of NASTRAN does not exist for that replacement system, then the NASTRAN workload could remain on the U1110 which is expected to be retained.

Should the future NASTRAN workload increase to 2S (2 times current structure size), NASTRAN's appetite for computing resources will approximately quadruple. While this resulting workload can be done on either the U1110 or the replacement system (assuming it hosts NASTRAN), the impact of NASTRAN will be felt by the system. In this workload range the mini-computer appears to be a good alternative. The mini's attraction is due to the following reasons:

a) If the mini is user operated, its relative CPF should be less than suggested by Figure 6-1,

b) An acquisition strategy phased to workload growth can be implemented, and

c) The mini can probably be obtained more quickly than a maxi, and thus result in avoiding significant costs if a service bureau would otherwise have to be used.

The attractiveness of the mini continues until some point before the 4S workload level. At that point the number of mini's (4 at 3½S, 6 at 4S) probably becomes impractical for the users to operate, and thus the minicomputer option might be avoided.

At the 4S workload level, the choice of using the CCF is still viable assuming that this much heavier NASTRAN workload can be dealt with economically (CPF of replacement system). If such is not the case, or if the workload goes beyond the 4S level, then procurement of a large-scale dedicated NASTRAN system such as the CYBER 175 (or CYBER 176*) appears to be a good choice assuming that any excess capacity can be utilized.

The workload of the future cannot be presently estimated since it will depend on as yet undefined future space structures. The conclusions of this report are based on as accurate an understanding of the future workload as can be obtained at the present time. It would be a relatively simple matter, through use of the approach described and the model used in this study, to reassess the situation at a future time should a better definition of the workload become available.

---

*Even though the CYBER 176 has not been addressed in this report, its ability to process NASTRAN jobs is approximately 30% better than that of the CYBER 175.

F. M. Richards Jr.

F. M. Richards, Jr.
Technical Staff

APPENDIX I

A SIMPLE STATICS PROBLEM

As an example of the solution of a structural analysis problem, consider the structure in Figure I-1 which is made up of a rigid block (on frictionless rollers) and four springs whose stiffnesses (in lbs/inch) are a, b, c, and d, as shown. The block is constrained so that it can only move left or right. Applying forces F_1, F_2 and F_3 at the points ①, ② and ③, respectively, produces displacements of U_1, U_2 and U_3 as shown.

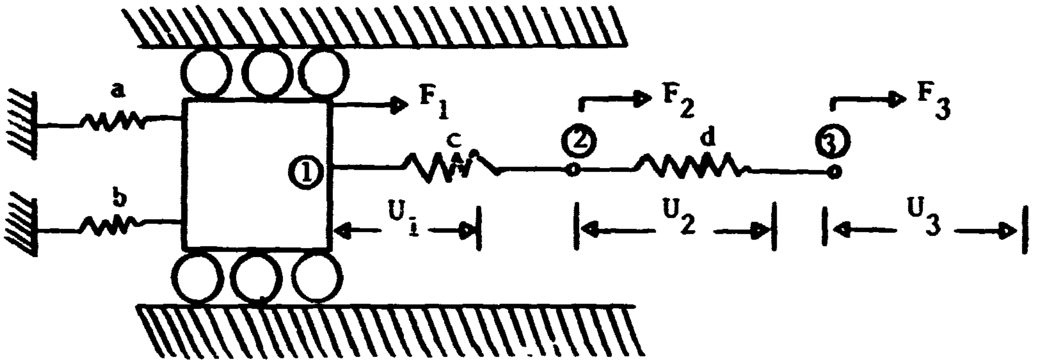


Figure I-1 Diagram of Structure

The mathematical formulation of the problem proceeds as follows.

Since the system is to remain in equilibrium, the following algebraic relationships must hold:

$$F_1 - aU_1 - bU_1 - cU_1 + cU_2 = 0$$
$$F_2 + cU_1 - cU_2 - dU_2 + dU_3 = 0$$
$$F_3 + dU_2 - dU_3 = 0$$

which is equivalent to:

$$F_1 = (a + b + c)\,U_1 + (-c)\,U_2 + (0)\,U_3$$
$$F_2 = (-c)\,U_1 + (c + d)\,U_2 + (-d)\,U_3$$
$$F_3 = (0)\,U_1 + (-d)\,U_2 + (d)\,U_3$$

Expressed in matrix notation, the above set of equations becomes

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} a+b+c & -c & 0 \\ -c & c+d & -d \\ 0 & -d & d \end{bmatrix} \begin{bmatrix} U_1 \\ U_2 \\ U_3 \end{bmatrix}$$

or F = KU, where F is the <u>Force Vector</u>, K is the <u>Stiffness Matrix</u>, and U is the <u>Displacement Vector</u>.

APPENDIX II

NASTRAN PERFORMANCE MODEL

As explained in Section 2, the execution characteristics of NASTRAN increase nonlinearly with respect to an increase in structure size. Therefore, to study the effects of larger structures it became necessary to represent this behavior mathematically. As is often the case with mathematical models of complex systems, certain simplifying assumptions are necessary; for this model those assumptions are:

(a) The symmetric decomposition activity represents 75% of the total job's requirements in terms of both CPU and I/O.

(b) The stiffness matrix, K, is tightly banded.

(c) The semi-bandwidth (or number of active columns), C, of K is related to the order, N, of K as $C = .04 \times N$.

(d) Where they were available, the MSC/NASTRAN timing and memory characteristics for SDCOMP were used:

t_1 ≡Arithmetic time for multiply-add loop.

t_2 ≡Time to pack (unpack) one term in a string of nonzero matrix terms.

t_3 ≡Time to pack (unpack) one element in a column of the matrix

t_4 ≡Average time required to read (write) one block.

n_1 ≡Number of words (or bytes) per floating point datum.

n_2 ≡Number of words (or bytes) per integer datum.

WB≡Amount of storage required for each I/O buffer.

WN≡Size of NASTRAN's instruction space including working storage.

Table II-I shows the values used.

Table 11-1. SDCOMP Timing and Memory Characteristics

SYSTEM		t_1 (μsec)	t_2 (μsec)	t_3 (μsec)	t_4(4) (sec)	n_1	n_2	WB(5)	WN
CDC	CYBER 173	8	6	10	.05	1	1	1792	25000
	CYBER 175	0.6	1.5	3	.05	1	1	1792	25000
DEC	VAX - 11/780(1)	13	16	17	.044	8	4	9400	192000
IBM	3031(1)	8	14	14	.041	8	4	9400	192000
	3032(1)	3	6	6	.041	8	4	9400	192000
	3033(2)	1.8	3	3	.041	8	4	9400	192000
UNIVAC	1110	6	5	11	.038	2	1	1796	34000
	1100/80(3)	5	5	10	.038	2	1	1796	34000
	1100/80 + SAM(3)	4	3	6	.038	2	1	1796	34000

(1) Estimated. Compare with IBM 370/158's t_1, t_2, t_3, WB, and WN = 10, 12, 13, 9400 and 192000, respectively.

(2) Estimated. Compare with AMDAHL 470/V6's t_1, t_2, t_3, WB, and WN = 1.8, 3, 3, 9400 and 192000, respectively.

(3) Estimated. Compare with UNIVAC 1110's t_1, t_2, t_3, WB, and WN = 6, 5, 11, 1796 and 34000, respectively.

(4) Based on WB and characteristics of devices configured for this study.

(5) Five (5) buffers are used.

The mathematics of system behavior expressed in terms of CPU, I/O, and memory requirements are as follows:

- The working space, WS, available for symmetric decomposition is

  $$WS = REGION - WN - 5 \times WB,$$

  where REGION is the amount of program memory space.

- If $WS < C \times (n_1 + 2n_2)$, then sufficient memory does not exist for SDCOMP to execute.

- If $C \times (C-1) < 2 \times (WS - C \times (n_1 + 2n_2))/n_1$, then SDCOMP can execute with no spill.

- If spill is to occur, the number of rows, S, which can be contained in a spill group are

  $$S = C - \sqrt{C \times (C-1) - 2 \times (WS - C \times (n_1+2n_2))/n_1}$$

- Exclusive of spill, the amount of CPU time for SDCOMP is

  $$CPU \left| SDCOMP \right| = N \times (0.5 \times t_1 \times C^2 + t_2 \times C)$$

- If spill occurs, increment CPU $\left| SDCOMP \right|$ by $N \times t_3 \times C^2/S$

- The Input/Output activity associated with SDCOMP consists of:

  (a) $R \left| K \right| \equiv$ Total time for reading K, the stiffness matrix.

  $$= t_4 \times \lceil (N \times (N \times \text{density}) \times n_1 + \text{header})/WB \rceil$$

  Where $\lceil x \rceil$ is the smallest integer $\geq x$, "density" is the ratio of nonzero terms to total terms of K, and "header" is the number of words (bytes) representing I/O record control information (header $= 8 \times n_2$.)

(b) $W\,|AC, PR| \equiv$ Total time for writing the active column vector/pivotal row file

$= t_4 \times \lceil N \times (C \times (n_1 + n_2) + \text{header})/WB \rceil$

(c) $R\,|AC, PR| \equiv$ Total time for reading the active column vector/pivotal row file

$= W\,|AC, PR|$

(d) $W\,|RESULTS| \equiv$ Total time for writing SDCOMP results

$= t_4 \times \lceil N \times ((C + 1) \times n_1 + \text{header})/WB \rceil$

and should spill occur,

(e) $W\,|SPILL| \equiv$ Total time for writing to the spill files

$$= t_4 \times \lceil (\lceil C/S - 1 \rceil \times \lceil C/S \rceil \times (S/2) + (\lceil N/S \rceil - \lceil C/S \rceil) \times (C-1)) \times C \times (n_1 + n_2)/WB \rceil$$

$R\,|SPILL| \equiv$ Total time for reading from the spill files

$= W\,|SPILL|$

(f) The total I/O for SDCOMP is

$$IO\,|SDCOMP| = R\,|K| + W\,|AC, PR| + R\,|AC, PR| + W\,|RESULTS| + W\,|SPILL| + R\,|SPILL|$$

- Finally, based on the 75% relationship of SDCOMP to total job,

$$CPU\,|JOB| = (4/3) \times CPU\,|SDCOMP|$$
$$IO\,|JOB| = (4/3) \times IO\,|SDCOMP|$$

The above algebra plus some of the other calculations involved in this study were coded as a FORTRAN program to facilitate computation. Documentation of that program is provided on the following pages in the form of:

(a) Program Symbol Dictionary
(b) Program Listing
(c) Sample Input
(d) Sample Output

SYMBOL DICTIONARY

BIGREG : "BIG REGION." USED FOR ELIMINATION OF SPILL.
BYPASS : PRINT SWITCH. TRUE==> PRINT SYSTEM CHARACTERISTICS
C : NUMBER OF ACTIVE COLUMNS.
CF1 : CONVERSION FACTOR. RATIO OF JOB CPU TIME TO SDCOMP CPU TIME
CF2 : CONVERSION FACTOR. RATIO OF JOB I/O TIME TO SDCOMP I/O TIME
COST : COST OF SPECIFIED SYSTEM
CPFH : COST PERFORMANCE FACTOR FOR SYSTEM. HIGH END OF RANGE.
CPFL : COST PERFORMANCE FACTOR FOR SYSTEM. LOW END OF RANGE.
CPU : SDCOMP CPU TIME (IN SECONDS.)
CS : MAXIMUM NO. OF ACTIVE COLUMNS WHICH CAN BE HELD IN WS WITHOUT SPILLING.
DENSTY : DENSITY OF STIFFNESS MATRIX.
GSI : GSI(X) = SMALLEST INTEGER .GE. X
HCPU : JOB CPU TIME EXPRESSED IN HOURS.
HEADER : NO. OF WORDS(BYTES) REPRESENTING I/O DATA BLOCK CONTROL INFORMATION.
HIOT : JOB I/O TIME EXPRESSED IN HOURS.
HOSTS : NAMELIST PROVIDING RUNTIME OPPORTUNITY FOR MODIFYING SYSTEM CHARACTERISTICS
HWCT : JOB WALL CLOCK TIME IN HOURS. EQUALS HCPU+HIOT.
IOB : NO. OF I/O BLOCKS TRANSFERRED DURING SDCOMP.
IOT : SDCOMP I/O TIME (IN SECONDS). BASED ON IOB & T4.
JOB : NAMELIST USED FOR DESCRIBING MODEL EXPERIMENTS.
JOBCPU : LARGE JOB CPU TIME (IN SECONDS).
JOBIOT : LARGE JOB I/O TIME (IN SECONDS).
JOBWCT : LARGE JOB WALL CLOCK TIME (IN SECONDS). EQUALS JOBCPU+JOBIOT.
N : NO. OF DOF IN STIFFNESS MATRIX.
NAME : ALPHANUMERIC IDENTIFICATION OF SYSTEM.
NLH : HIGH END OF RANGE FOR NUMBER OF LARGE JOBS.
NLL : LOW END OF RANGE FOR NUMBER OF LARGE JOBS.
NSREGN : APPROXIMATE MINIMUM MEMORY REGION RESULTING IN NO SPILL
NSYS : NO. OF SYSTEMS REPRESENTED BY MODEL
N1 : NO. OF WORDS(BYTES) PER FLOATING POINT DATUM.
N2 : NO. OF WORDS(BYTES) PER INTEGER DATUM.
REGION : USER IMPOSED MEMORY REGION SIZE.
RIN : NO. OF BLOCKS READ FROM FILE CONTAINING ACTIVE COLUMN/PIVOTAL ROWS.
RKLL : NO. OF BLOCKS READ FROM STIFFNESS MATRIX FILE DURING SDCOMP
RSI : NO. OF BLOCKS READ FROM SPILL FILE DURING SDCOMP.
S : WHEN SPILL OCCURS, S IS THE NUMBER OF ROWS WHICH CAN BE CAN BE CONTAINED IN A SPILL GROUP.
SJCPH : HIGH END OF RANGE FOR SMALL JOB CPU TIME (IN MINUTES).
SJCPL : LOW END OF RANGE FOR SMALL JOB CPU TIME (IN MINUTES).
SJIOH : HIGH END OF RANGE FOR SMALL JOB I/O TIME (IN MINUTES).
SJIOL : LOW END OF RANGE FOR SMALL JOB I/O TIME (IN MINUTES).
SPILL : LOGICAL FLAG. TRUE ==> SPILL IN EFFECT FOR THIS PROBLEM AND SYSTEM.
SQTARG : TEMPORARY VARIABLE. USED FOR TESTING SIGN OF QUANTITY BEFORE TAKING SQUARE ROOT.
SS : LOGICAL VARIABLE. TRUE ==> PROBLEM IS SUBSTRUCTURED.
SS$: COST OF SYSTEM ON WHICH SUBSTRUCTURED WORKLOAD IS PROCESSED
SUFH : HIGH END OF RANGE FOR SYSTEM UTILIZATION FRACTION.
SUFL : LOW END OF RANGE FOR SYSTEM UTILIZATION FRACTION.
SYSTEM : NUMERICAL INDEX CORRESPONDING TO COMPUTER SYSTEM MODELED.
T1 : TIMING CHARACTERISTIC OF MODELED SYSTEM. SEE TABLE II-I
T2 : TIMING CHARACTERISTIC OF MODELED SYSTEM. SEE TABLE II-I
T3 : TIMING CHARACTERISTIC OF MODELED SYSTEM. SEE TABLE II-I
T4 : TIMING CHARACTERISTIC OF MODELED SYSTEM. SEE TABLE II-I
WB : SYSTEM'S I/O BUFFER SIZE (IN WORDS(BYTES))
WIN : NO. OF BLOCKS WRITTEN TO ACTIVE COLUMN/PIVOTAL ROW FILE
WLLL : NO. OF BLOCKS WRITTEN TO FINAL OUTPUT FILE.
WN : SIZE OF NASTRAN INSTRUCTION SPACE. (IN WORDS(BYTES))
WOSS$: COST OF SYSTEM CONFIGURED FOR NON-SUBSTRUCTURING

```
WS      : WORKING STORAGE AVAILABLE FOR SDCOMP SCRATCH PAD
WSO     : NO. OF BLOCKS WRITTEN TO SPILL FILE.
XN1     : ARRAY FROM WHICH N1 IS OBTAINED. N1=XN1(SYSTEM)
XN2     : ARRAY FROM WHICH N2 IS OBTAINED. N2=XN2(SYSTEM)
XT1     : ARRAY FROM WHICH T1 IS OBTAINED. T1=XT1(SYSTEM)
XT2     : ARRAY FROM WHICH T2 IS OBTAINED. T2=XT2(SYSTEM)
XT3     : ARRAY FROM WHICH T3 IS OBTAINED. T3=XT3(SYSTEM)
XT4     : ARRAY FROM WHICH T4 IS OBTAINED. T4=XT4(SYSTEM)
XWB     : ARRAY FROM WHICH WB IS OBTAINED. WB=XWB(SYSTEM)
XWN     : ARRAY FROM WHICH WN IS OBTAINED. WN=XWN(SYSTEM)
Z       : FLOATING POINT EQUIVALENT OF S.
```

NASTRAN COST/PERFORMANCE CALCULATIONS
(UNIVAC FORTRAN V)

```
      COMPILER (GEN=LIBRY)
      IMPLICIT INTEGER(A-Z)
      PARAMETER NSYS=8,BIGREG=1000000000
      DIMENSION XWB(NSYS),XN1(NSYS),XN2(NSYS),XT4(NSYS),XWN(NSYS),
     -          XT1(NSYS),XT2(NSYS),XT3(NSYS),NAME(3,NSYS),
     -          WOSS$(NSYS),SS$(NSYS),SJCPL(NSYS),SJCPH(NSYS)
      REAL XT1,XT2,XT3,XT4,T1,T2,T3,T4,Z,DENSTY,CF1,CF2,SUFL,SUFH,
     -     CPFL,CPFH,JOBCPU,JOBIOT,JOBWCT,HCPU,HIOT,HWCT,SJIOL,SJIOH
      LOGICAL BYPASS,SPILL,SS
      DATA BYPASS/T/ SS/F/
      DATA CF1/1.333/ CF2/1.333/ NLL/2/ NLH/3/
      DATA (SJCPL(J),SJCPH(J),J=1,NSYS)/
     -     231,469,    57,156,   600,1250,    250,469,    97,179,
     -      56,104,   250,375,    167,375/
      DATA SJIOL/100./  SJIOH/125./
      DATA ((NAME(J,I),J=1,3),WOSS$(I),SS$(I),XT1(I),XT2(I),
     -      XT3(I),XT4(I),XN1(I),XN2(I),XWN(I),XWB(I),I=1,NSYS)/
     -'CDC CYBER 173     ',32539,32539,8.0,6.0,10.,.050,1,1, 25000,1792,
     -'CDC CYBER 175     ',49892,49892,0.6,1.5,3.0,.050,1,1, 25000,1792,
     -'DEC VAX-11/780    ',11916, 8733,13.,16.,17.,.044,8,4,192000,9400,
     -'IBM 3031          ',28019,25715,8.0,14.,14.,.041,8,4,192000,9400,
     -'IBM 3032          ',42258,40111,3.0,6.0,6.0,.041,8,4,192000,9400,
     -'IBM 3C33          ',61143,56912,1.8,3.0,3.0,.041,8,4,192000,9400,
     -'UNIVAC 1100/80    ',26756,26756,5.0,5.0,10.,.038,2,1, 34000,1796,
     -'UNIVAC 1100/80+SAM',28410,28410,4.0,3.0,6.0,.038,2,1, 34000,1796/
      NAMELIST/HOSTS/XT1,XT2,XT3,XT4,XN1,XN2,XWN,XWB,
     -               WOSS$,SS$,SJCPL,SJCPH,BYPASS
      NAMELIST/JOB/SYSTEM,REGION,C,N,DENSTY,SS
      DEFINE G$I(X)=INT(X+0.99999999)
      READ(5,HOSTS,END=300)
      IF(BYPASS) GO TO 200
      PRINT 1000
      DO 100 I=1,NSYS
      PRINT 1100,(NAME(J,I),J=1,3),I,XT1(I),XT2(I),XT3(I),XT4(I),
     -           XN1(I),XN2(I),XWN(I),XWB(I)
  100 CONTINUE
  200 PRINT 1200
  300 READ(5,JOB,END=400)
      IF(SYSTEM.EQ.0) GO TO 300
      IF(SS) REGION=BIGREG
      T1=XT1(SYSTEM)*1E-6
      T2=XT2(SYSTEM)*1E-6
      T3=XT3(SYSTEM)*1E-6
      T4=XT4(SYSTEM)
      N1=XN1(SYSTEM)
      N2=XN2(SYSTEM)
      HEADER=8*N2
      WN=XWN(SYSTEM)
      WB=XWB(SYSTEM)
      NSREGN=1.01*(N1*C*(C-1)/2.+WN+5*WB+C*(N1+2*N2))
      IF(.NOT.SS) REGION=MIN0(REGION,NSREGN)
      WS=REGION-WN-5*WB
      SQTARG=2*(WS-C*(N1+2*N2))/N1
      IF(SQTARG.LT.0) GO TO 500
      CS=SQRT(SQTARG)
      S=0
```

```
      SQTARG=C*(C-1)-CS**2
      IF(SQTARG.GE.0) S=C-SQRT(SQTARG)
      Z=MAX0(1,S) @ REALLY MEANS Z=S. AVOIDING OPTIMIZER PROBLEMS
      SPILL=.FALSE.
      IF(S.NE.0) SPILL=.TRUE.
      W=0.5*N1*C**2
      CPU=N*(0.5*T1*C**2+T2*C) + 0.5
      IF(SPILL) CPU=CPU+N*T3*C**2/Z
      RKLL=G$I(N*(N*DENSTY*N1+HEADER)/FLOAT(WB))
      WIN=G$I((N*(C*(N1+N2)+HEADER))/FLOAT(WB))
      RIN=WIN
      WLLL=G$I((N*((C+1)*N1+HEADER))/FLOAT(WB))
      WSO=0
      IF(SPILL) WSO=G$I(((G$I(C/Z)-1)*G$I(C/Z)*S/2.+
     -              (G$I(N/Z)-G$I(C/Z))*(C-1))*C*(N1+N2)/FLOAT(WB))
      RSI=WSO
      IOB=RKLL+WIN+RIN+WLLL+WSO+RSI
      IOT=TA*IOB + 0.5
      JOBCPU=CF1*CPU
      IF(SS) JOBCPU=0.80*JOBCPU
      JOBIOT=CF2*IOT
      IF(SS) JOBIOT=CF2*IOT+(2*N/5000.+1)*120. @ 120 = 2 MINUTES OF I/O
      JOBWCT=JOBCPU+JOBIOT
      SUFL=(MAX(NLL*JOBCPU/60.,FLOAT(N)/10000*SJIOL)+
     -      MAX(FLOAT(N)/10000*SJCPL(SYSTEM),NLL*JOBIOT/60.))/72.
C ... ABOVE & BELOW: /72. ==> (/60/120)*100. WHERE 60 IS USED TO
C ... CONVERT MIN TO HR. 120 TO ADJUST TO 120-HR SYSTEM WEEK, AND
C ... 100 IS USED TO CONVERT TO A PERCENTAGE.
      SUFH=(MAX(NLH*JOBCPU/60.,FLOAT(N)/10000*SJIOH)+
     -      MAX(FLOAT(N)/10000*SJCPH(SYSTEM),NLH*JOBIOT/60.))/72.
      COST=WOSS$(SYSTEM)
      IF(SS) COST=SS$(SYSTEM)
      CPFL=((SUFL/100.)*COST)/100.
      CPFH=((SUFH/100.)*COST)/100.
      HIOT=JOBIOT/3600.
      HCPU=JOBCPU/3600.
      HWCT=JOBWCT/3600.
      WRITE(6,2000) (NAME(J,SYSTEM),J=1,3),COST,N,C,REGION,
     -              SUFL,CPFL,SUFH,CPFH,HCPU,HIOT,HWCT
      GO TO 300
  400 STOP
  500 WRITE(6,1500)
      GO TO 300
 1000 FORMAT('1SYSTEM CHARACTERISTICS:'///20X,'ID        T1     T2',
     -        '      T3         T4         N1       N2          WN         WB '//)
 1100 FORMAT(1X,3A6,I3,F10.1,F6.1,F7.1,F9.3,I7,I7,I9,I8)
 1200 FORMAT(////T24,'COST',5X,'N',5X,'C',8X,'REGION',7X,'SUFL',
     -        4X,'CPFL',7X,'SUFH',4X,'CPFH',14X,'LCPU',4X,
     -        'LIOT',4X,'LWCT'//)
 1300 FORMAT(T6,' SYSTEM             REGION     C       N'/1X3A6,I8,I5,I7)
 1500 FORMAT('0REGION SIZE TOO SMALL FOR EXECUTION.')
 2000 FORMAT(1X,3A6,'  $',I5,I8,I6,3X,I9,F10.1,'%',F8.0,F10.1,
     -        '%',F8.0,10X,3F8.2)
      END
```

SAMPLE INPUT

```
FD2-X14669*TPFS(1).EXPERIMENT
     1          $HOSTS BYPASS=.F. $
     2          $JOB N=10000,C=400,DENSTY=.003,SYSTEM=1,REGION=118000 $
     3          $JOB SYSTEM=2 $
     4          $JOB REGION=900000,SYSTEM=3 $
     5          $JOB SYSTEM=4 $
     6          $JOB SYSTEM=5 $
     7          $JOB SYSTEM=6 $
     8          $JOB REGION=210000,SYSTEM=7 $
     9          $JOB SYSTEM=8 $
    10          $JOB SYSTEM=0,SS=T $
    11          $JOB N=10000,C=400,DENSTY=.003,SYSTEM=1,REGION=118000 $
    12          $JOB SYSTEM=2 $
    13          $JOB REGION=900000,SYSTEM=3 $
    14          $JOB SYSTEM=4 $
    15          $JOB SYSTEM=5 $
    16          $JOB SYSTEM=6 $
    17          $JOB REGION=210000,SYSTEM=7 $
    18          $JOB SYSTEM=8 $
```

SAMPLE OUTPUT

```
SYSTEM CHARACTERISTICS:

                        ID      T1    T2     T3        T4      N1     N2       WN      WB


CDC CYBER 173           1      8.0   6.0   10.0      .050      1      1     25000    1792
CDC CYBER 175           2       .6   1.5    3.0      .050      1      1     25000    1792
DEC VAX-11/780          3     13.0  16.0   17.0      .044      8      4    192000    9400
IBM 3031                4      8.0  14.0   14.0      .041      8      4    192000    9400
IBM 3032                5      3.0   6.0    6.0      .041      8      4    192000    9400
IBM 3033                6      1.8   3.0    3.0      .041      8      4    192000    9400
UNIVAC 1100/80          7      5.0   5.0   10.0      .038      2      1     34000    1796
UNIVAC 1100/80+SAM      8      4.0   3.0    8.0      .038      2      1     34000    1796



                          COST      N     C         REGION       SUFL    CPFL      SUFH    CPFH               LCPU     LIOT     LWCT


CDC CYBER 173           $32539   10000   400        116109       7.2%     23.     12.5%     41.               2.38      .21     2.59
CDC CYBER 175           $49892   10000   400        116109       2.2%     11.      3.9%     19.                .18      .21      .39
DEC VAX-11/780          $11916   10000   400        892638      14.8%     18.     27.0%     32.               3.87      .23     4.10
IBM 3031                $28019   10000   400        892638       7.5%     21.     12.5%     35.               2.39      .21     2.60
IBM 3032                $42259   10000   400        892638       2.8%     12.      4.7%     20.                .90      .21     1.11
IBM 3033                $61143   10000   400        892638       2.2%     13.      3.2%     19.                .54      .21      .75
UNIVAC 1100/80          $26756   10000   400        206221       6.0%     16.      8.9%     24.               1.49      .26     1.75
UNIVAC 1100/80+SAM      $28410   10000   400        206221(1)    4.3%     12.      8.2%     23.               1.19      .26     1.45
CDC CYBER 173           $32539   10000   400    **********       6.4%     21.     11.3%     37.               1.90      .38     2.28
CDC CYBER 175           $49892   10000   400    **********       2.2%     11.      3.9%     19.                .14      .38      .52
DEC VAX-11/780          $ 8733   10000   400    **********      13.5%     12.     25.1%     22.               3.10      .40     3.49
IBM 3031                $25715   10000   400    **********       6.7%     17.     11.3%     29.               1.91      .38     2.29
IBM 3032                $40111   10000   400    **********       2.7%     11.      4.3%     17.                .72      .38     1.10
IBM 3033                $56912   10000   400    **********       2.2%     12.      3.2%     18.                .43      .38      .81
UNIVAC 1100/80          $26756   10000   400    **********       5.5%     15.      8.2%     22.               1.19      .42     1.62
UNIVAC 1100/80+SAM      $28410   10000   400    **********       3.9%     11.      7.6%     22.                .96      .42     1.38
```

(1) "**********" indicates substructuring

REFERENCES

[1] Butler, T. G. and Michel, D., NASTRAN - A Summary of the Functions and Capabilities of the NASA Structural Analysis Computer System, NASA, SP-260, 1971.

[2] Davis, P. J, The Mathematics of Matrices, Blaisdell Publishing Co., New York, NY, 1965.

[3] The NASTRAN Theoretical Manual, NASA, SP-221(03), 1976.

[4] MSC/NASTRAN Application Manual, The MacNeal-Schwendler Corporation, 1974.

[5] Everstine, G. C., BANDIT User's Guide, David W. Taylor Naval Ship Research and Development Center, TM-184-77-03, 1977.

[6] Levy, R., Structural Stiffness Matrix Wavefront Resequencing Program (WAVEFRONT), Jet Propulsion Laboratory, JPL Report 32-1526, Vol. XIV, 1972.

[7] Cook, R, D., Concepts and Applications of Finite Element Analysis, John Wiley & Sons, Inc., New York, NY, 1974.

[8] Storaasli, O. O. and Murphy, R. C., Finite Element Analysis in a Minicomputer/Mainframe Environment, NASA Conf. Pub. 2059, October 1978.

[9] Field, E. I., Feasibility Study for the Implementation of NASTRAN on the ILLIAC IV Parallel Processor, Universal Analytics, Inc., NASA CR-132702, 1975(?).

[10] Schwendler, R. G. et al, Study of the Modifications Needed for Efficient Operation of NASTRAN on the CDC 7600 Computer, The MacNeal-Schwendler Corporation, NASA CR-144940, 1975(?).

[11] Golden, M. E and Hurwitz, M. M., NASTRAN on the Texas Instruments Advanced Scientific Computer: Feasibility Study and Experimentation, David W. Taylor Naval Ship Research and Development Center, DTNSRDC Report 76-0092, 1976.

[12] ---, Study of the Modifications Needed for Efficient Operation of NASTRAN on the Control Data Corporation STAR-100 Computer, Aerospace Division of Control Data Corporation, NASA CR-132644, 1974.

[13] DATAPRO, Datapro Research Corporation, Delran, N.J., Vol. 1, 1979.

REFERENCES (Concluded)

[14] Hennigan, M. L. and Lenker, M. D., Application Tuning of NASA's NASTRAN Program, The MITRE Corporation, MTR-4704 (Rev. 1), November 1977.

[15] Patton, J. V., Results of U1108-8 Throughput Analysis, The MITRE Corporation, WP-5197, September 1978.