FINITE ELEMENT ANALYSIS IN A MINICOMPUTER/MAINFRAME ENVIRONMENT

Olaf O. Storaasli
NASA Langley Research Center

Ronald C. Murphy
Joint Institute for Advancement of Flight Sciences
The George Washington University

INTRODUCTION

The advent of modular finite element systems provides the opportunity for engineers to solve a broad range of structures problems in a distributed computing environment. To maintain versatility in this changing computing environment (ref. 1), changes may be appropriate in the design concepts of future finite element systems. Recent exploratory studies (refs. 2-3) have shown that minicomputers offer great potential for solving structures problems. The purpose of this paper is to investigate design considerations for general purpose finite element systems to maximize performance when installed on distributed computer hardware/software systems. This paper explores how the features of current minicomputers complement those of a modular implementation of the finite element method. Central to this investigation is increasing the control, speed, and visibility (interactive graphics) by structural engineers in solving a broader range of structural problems at reduced cost. The approach used is to implement a finite element system in a distributed computer environment to solve structural problems and to explore alternatives in distributing finite element computations.

THE FINITE ELEMENT METHOD

To implement the finite element method on computers for typical static, dynamic, buckling, and thermal analyses, two approaches are commonly used. The first approach (fig. 1(a)), and the one which dominated software design concepts prior to the advent of virtual memory, is to use an executive program to connect and communicate with analysis overlays in a fixed, serial fashion. This method is known to lack modularity, portability, and efficiency and often requires significant effort to make minor software changes. The second approach (fig. 1(b)) is to implement each analysis activity of the finite element method as an independent processor and have all processors communicate through a common data base. The key to implementing such a modular approach is to have system and/or data base utility software (ref. 4) to open, read, and write to named files from within finite element processors. In the finite element procedure, the function of each processor can be selected to minimize computing time and memory. Such a modular approach is the basis of the implementation of the SPAR (ref. 5) finite element system on the PRIME computer* and is well suited for use in the current investigation.

It is important to identify how large a problem (degrees of freedom) may be solved conveniently on a minicomputer and which processors are the bottleneck with regard to computation time. Figure 2 shows minicomputer solution times for a range of problems vs. problem size (e.g., number of degrees of freedom). Also shown are projected times based on planned enhancement which should reduce solution times by a factor of two to four. Thus, to achieve solutions to static analysis problems on the minicomputer in less than 30 minutes, a reasonable problem size is about 2500 degrees of freedom. While large problems may be solved conveniently on the minicomputer mostly in a background mode, the 30 minutes shown in figure 2 (horizontal line) is probably a reasonable upper limit for engineering users to maintain thought continuity and work on other activities while background computations are in progress.

By analyzing the solution time for specific components (SPAR processors) of the finite element process, it is possible to identify functions which may be suited for mainframe or array processor calculations (see Results). A high-speed data link connecting the minicomputer to a CDC 6600 (roughly eight times the computation speed of the minicomputer) was explored to transfer "number crunching" activities. Use of this link is to minimize overall computation time and yet preserve the advantages of quick response and high-speed user interface provided by the minicomputer for structural engineering activity which involves interaction.

COMPUTER HARDWARE AND SOFTWARE

Today's minicomputers have similar capabilities to large mainframe computers, except the cost and CPU speed are about an order of magnitude less. (Table 1). Table 1 shows results of a benchmark test run on fifteen computers to simulate structures calculations (double precision matrix operations using nested DO loops). On the left are times to run the benchmark for seven large mainframes, and on the right are times to run the same benchmark for eight minicomputers. The table illustrates variations in both performance and cost

*The SPAR-minicomputer version, in use at NASA Langley Research Center and on several NASA contracts, is now available from COSMIC, the distribution center for NASA software.

for both mainframes and minicomputers. For interactive engineering calculations, many of the "mainframe" peripherals (high-speed card readers, line printers, punches, etc.) are not required.

TABLE 1.- CPU TIME FOR STRUCTURES BENCHMARK

MAINFRAME	TIME (SEC)	MINICOMPUTER	TIME (SEC)
CDC CYBER 175	2	DEC VAX 11/780	23
IBM 360/95	3	DEC PDP 11/70	42
IBM 370/168	4	PRIME 500	47
CDC 6600	8	SEL 32/75	52
IBM 360/75	10	PRIME 400	65
CDC CYBER 173	12	SIGMA V	71
UNIVAC 1108	15	SEL 32/55	72
		MODCOMP IV	85
Cost Range ($2-6 Million)		Cost Range ($50-150 Thousand)	

Figure 3, upper left, shows one of the minicomputers at Langley Research Center on which the SPAR finite element code has been implemented. This PRIME 400 minicomputer contains 32-bit arithmetic registers, 192 000 16-bit words of real memory, and 80 million words of disk storage, and costs about $150 000. The virtual memory on the minicomputer permits each time-share user a working space in excess of 1 million words. Currently, seven high-speed (4800-9600 baud) data lines link seven Tektronix 4014 graphics terminals to the minicomputer.

Figure 3 also shows a 4800 baud intercomputer data link which permits lengthy iterative ("number crunching") activity to be transferred from the minicomputer to the large mainframe computer by entering a simple command from any Tektronix terminal. The primary reason for the minicomputer as the user interface (see refs. 2-3) was the increased capability available to users (through both hardware and software advances) at a significantly reduced cost when compared to time-shared computing on a large mainframe.

RESULTS

Approximately twenty smaller problems (less than 2500 degrees of freedom) were solved entirely on the minicomputer and each result was obtained within 30 minutes. For these problems, obtaining solutions on the minicomputer in a stand-alone mode was satisfactory with no need to off-load portions of calculations to faster computation devices. However, for three larger problems (figs. 4-6) the trends indicate that large finite element problems should be solved in a distributed computing environment which contains high-speed computing capabilities.

Figure 4 is a minicomputer plot of a finite element model of a current NASA flight project vehicle typical of a problem whose solution time on the minicomputer is less than 30 minutes. The model has 1120 degress of freedom and 450 structural nodes and consists of 728 two-node and 374 four-node elements. Symmetry constraints about the aircraft center line and y-constraints on the wing leading and trailing edges were imposed, and rigid masses were used in the fuselage. Load cases simulated were a fuel inertia relief maneuver condition, a cruise condition, and a taxi condition. The wing model has three degrees of freedom at each structural node point.

Figure 5 shows a finite element model of a launch umbilical tower with 2208 degrees of freedom, 372 structural nodes, 944 two-node elements, and six degrees of freedom per node. The model was subjected to a downward prestress load of 1 g unit.

Figure 6 is a minicomputer plot of a 4708-degree-of-freedom finite element model of the National Transonic Facility currently under construction at Langley. This cryogenic wind tunnel model is the largest finite element problem attempted thus far on a minicomputer at Langley; it has six degrees of freedom per node and requires 3 CP hours for the static solution.

The finite element models shown in figures 4-6 have distributions of solution time shown in figures 7-9, respectively. Shown on the abscissa of figures 7-9 are components of the finite element process for SPAR as they occur for static analysis. Shown on the ordinate of the figures are the central processor times (CP) in minutes. The processors TAB, ELD, TOPO, and E process the node point, element, topology, and elemental stiffness matrices. Figures 7-9 show that the model generation activity requires little CP time and is well suited for a minicomputer. Formation of the element data packets (EKS) involves significant CP time where a large number of three- and four-node elements are involved. The tower (fig. 8) requires less CP time for EKS (since it contains simple bar elements) even though it has more degrees of freedom than the wing model (fig. 7). Assembly of the global stiffness matrix (K) is the dominant CP activity for the wing model (fig. 7), while the decomposition processor (INV) is dominant for the larger tower and tunnel models (figs. 8-9). For the SPAR finite element system, the decomposition time (INV) is proportional to the cube of the degrees of freedom allowed at a structural node point. Thus, for large models, care should be exercised to include only those freedoms actually required. The remaining loads (AUS), static solution (SSOL), and stress (GSF) processors are less important from the standpoint of CP time for all models. Not shown in the figure are results obtained for free vibration analysis (EIG) which is a major CP user for large complex models. Recent improvements in solution time due to use of a virtual memory loader are shown by the dashed lines in figures 7-9.

Figures 7-9 show that for static analysis of large structures on a minicomputer, the EKS, K, and INV components of the finite element process dominate CP requirements and are prime candidates for being relegated to a mainframe or array processor. The EKS and K processors are less time consuming where two-node elements are used in the finite element process, and the decomposition (INV) processor is dominant for such models with more than 2500 degrees of freedom. Although the results shown in figures 7-9 are a function of the particular finite element system and its implementation on the PRIME 400 minicomputer, the general trends should be representative. In particular, they suggest some advantage to conducting finite element calculations in a distributed computer environment.

ISSUES INVOLVED IN DISTRIBUTING COMPUTATIONS

The above results suggest that, ideally, an automated selection of computer hardware for the EKS, K, and INV processors based on problem size and element complexity should be initiated to minimize the solution cost and time. However, for distributed structural computations, there is still a long way to go before such an automated system is achieved.

The current distributed capability (fig. 3) consists of both hardware and software. The hardware used in the data transfer is a disk on both the minicomputer and the mainframe, a modem on each, a synchronous multi-line controller (SMLC), and a telephone line. The software used includes the protocol supported by the mainframe computer (UT200), communications software on the minicomputer (COMET), and special software written to permit the transfer of SPAR binary data base files between computers. Use of this procedure soon exposed a basic deficiency in that excessive time was spent formatting data into 80-column card images and then reconstructing the data again after data transmission. This excessive formatting time will soon be overcome with the replacement of current protocol (UT200) by a better protocol (HASP) in the new release of the mainframe operating system, which supports the direct transfer of binary data at 9600 baud.

Another alternative being considered is to adapt the finite element software to permit the connection of an array processor directly to the minicomputer to overcome these hardware and software restrictions (i.e., 9600 baud, data transfer, and formatting). This approach looks promising from a technical point of view at present, as do increased CPU speed on minicomputers and the use of certain advanced computer linking devices (i.e., HYPERchannel, ref. 6), with transfer rates of 50 million bytes/sec. The current distributed configuration (fig. 3) at Langley permits computations on both the minicomputer and mainframe by using communications software to transfer SPAR data between the two computers at 4800 baud. However, the transfer process takes longer (in many cases) than the equivalent time for the minicomputer to perform the computations. Future software and hardware enhancements currently planned should, however, remove some of these restrictions.

The authors have already introduced several performance and efficiency improvements in the SPAR minicomputer version and it is clear that sometimes small subtle changes can lead to reductions in cost and time by factors of

2 to 3 or more. The mainframe version of SPAR has been optimized with judicious use of machine code (CDC COMPASS) and such improvements will continue as software packages are tuned to take advantage of specific hardware features. The distributed computations made thus far indicate that the above hardware and software configuration accomplishes the distribution of tasks with a moderate degree of success. However, a 50 million byte/sec computer communication link or judicious use of an array processor on the minicomputer could significantly improve the distributed solution of large finite element problems.

The modularity of the SPAR system made the combination of both mainframe and mini computing environments possible. Future finite element systems should have this feature, modularity in their design, so that time consuming number crunching tasks may be readily distributed to appropriate computing devices (i.e., mainframe computers, array processors, or specially tailored microprocessors) which are better suited for such tasks.

CONCLUDING REMARKS

This paper presents results of exploratory studies on how the modularity of the finite element process can complement the advantages of low-cost, quick-response minicomputers. The finite element process is separated into its basic building blocks (processors) for the SPAR finite element system, and minicomputer central processing (CP) times of each processor are shown for three finite element models. Results are then discussed for the case of a minicomputer linked to a remote mainframe host. It is shown that for problems up to about 2500 degrees of freedom, the performance of the minicomputer in solving the problem in a stand-alone mode is acceptable. While the virtual memory of the minicomputer removes any restriction on problem size, its slower CPU speed tends to place a practical limit on the size of interactive finite element solutions (approximately 2500 degrees of freedom). An initial distributed system is discussed in which computations are performed on both the minicomputer and the mainframe and data transferred between them. The deficiencies of this system are identified and a computer linking system is discussed which makes this distributed system practical. Array processors on minicomputers to carry out high-speed vector calculations may also be viable alternatives which, in many cases, may decrease the need for a high-speed link to the mainframe. Such strategies or combinations thereof should be developed and updated in future finite element systems. Most important, however, future finite element systems should be sufficiently modular to allow the interactive user the opportunity to take advantage of the capability offered by a wide variety of advanced computer hardware, either currently available, or likely to evolve in the near future.

REFERENCES

1. Noyce, R. N.: Microelectronics, Sci. American, vol. 237, no. 3, Sept. 1977, pp. 62-69.

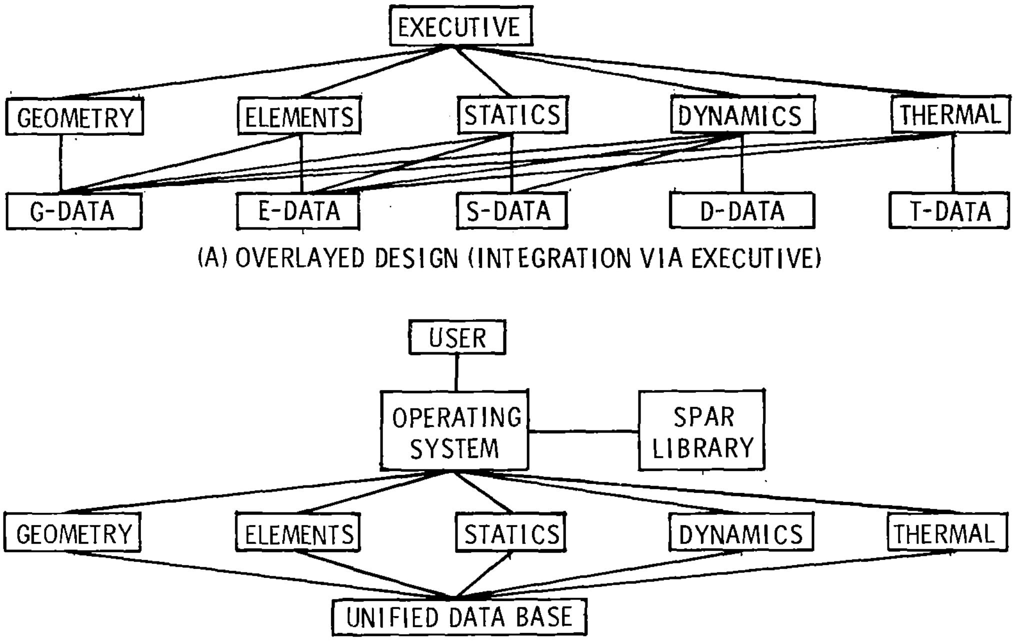
2. Storaasli, O. O.: On the Role of Minicomputers in Structural Design. Comput. & Struct. vol. 7, Feb. 1977, pp. 117-123.

3. Storaasli, O. O.; and Foster, E. P.: Cost Effective Use of Minicomputers To Solve Structural Problems. AIAA Paper No. 78-484 presented at the 19th AIAA/ASME Structures, Structural Dynamics and Materials Conference, Apr. 1978.

4. Giles, G. L.; and Haftka, R. T.: SPAR Data Handling Utilities. NASA TM-78701, 1978.

5. Whetstone, W. D.: SPAR Structural Analysis System Reference Manual, Vols. 1-3. NASA CR-145098-1-3, 1977.

6. McCusker, Tom: Talk Is Cheap Over Thornton's Channels. Datamation, vol. 22, no. 3, Mar. 1976, pp. 166-170.



(A) OVERLAYED DESIGN (INTEGRATION VIA EXECUTIVE)

(B) MODULAR PROCESSOR DESIGN (INTEGRATION VIA DATA BASE)

Fig. 1 Finite element software architectures.

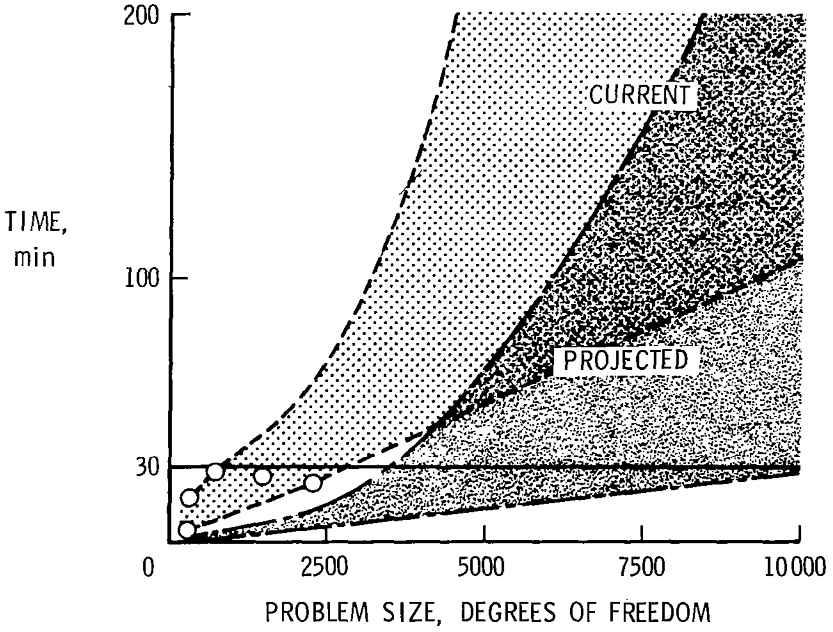


Fig. 2 Minicomputer solution time vs. problem size.

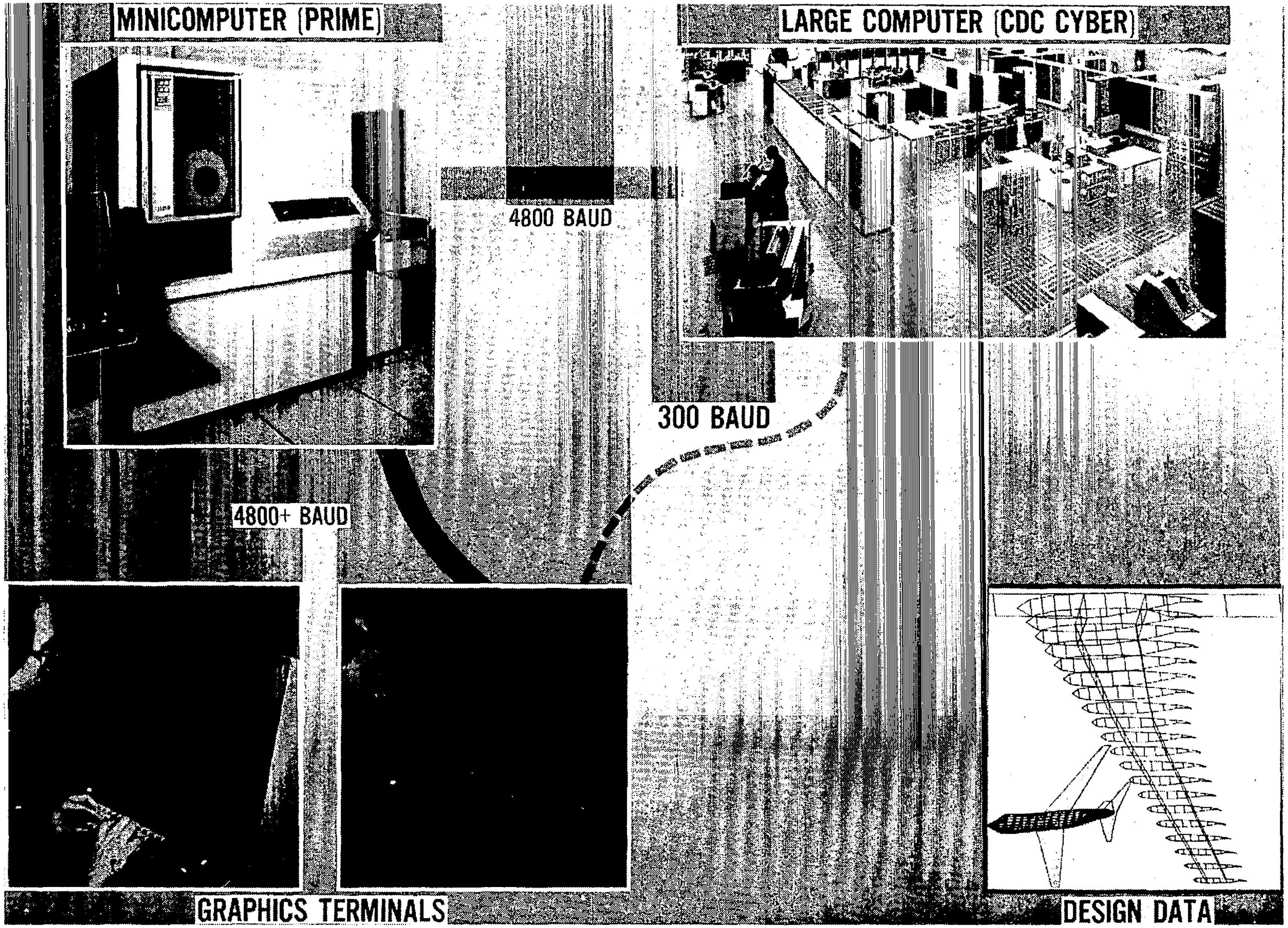
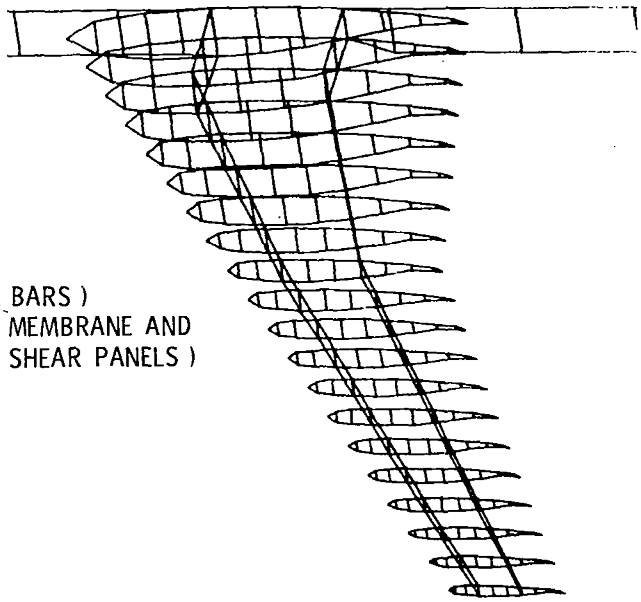


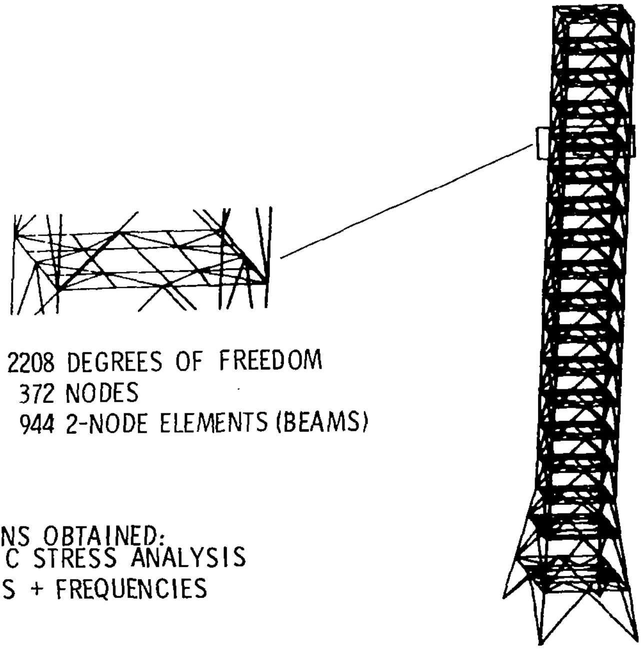
Fig. 3 Interactive terminal access to minicomputer and mainframe computer.



1120 DEGREES OF FREEDOM
450 NODES
1102 ELEMENTS
{728 2-NODE ELEMENTS (BARS)
{374 4-NODE ELEMENTS (MEMBRANE AND SHEAR PANELS)

SOLUTIONS OBTAINED:
STATIC STRESS ANALYSIS

Fig. 4 Plot (bars shown) of finite element wing model.



2208 DEGREES OF FREEDOM
372 NODES
944 2-NODE ELEMENTS (BEAMS)

SOLUTIONS OBTAINED:
STATIC STRESS ANALYSIS
MODES + FREQUENCIES

Fig. 5 Plot of launch umbilical tower model.

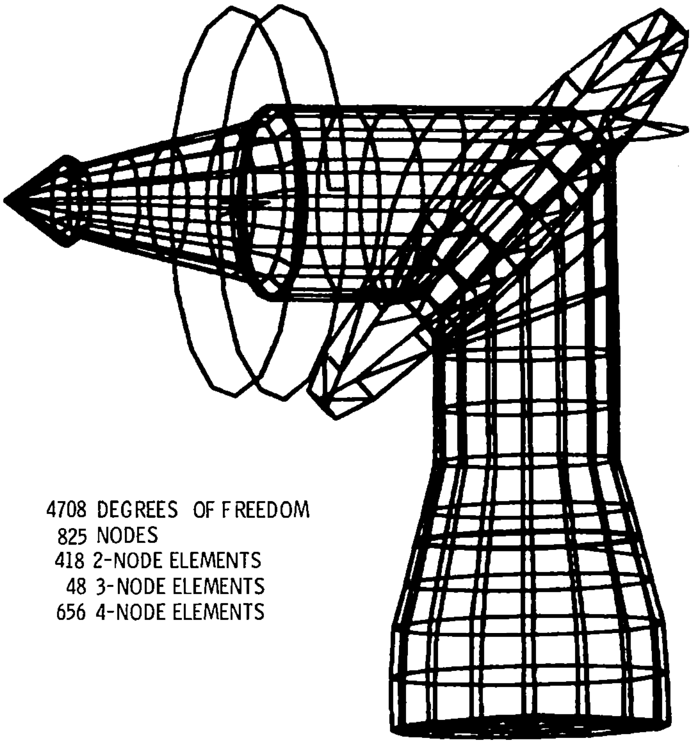


Fig. 6 Plot of National Transonic Facility (cryogenic wind tunnel) model

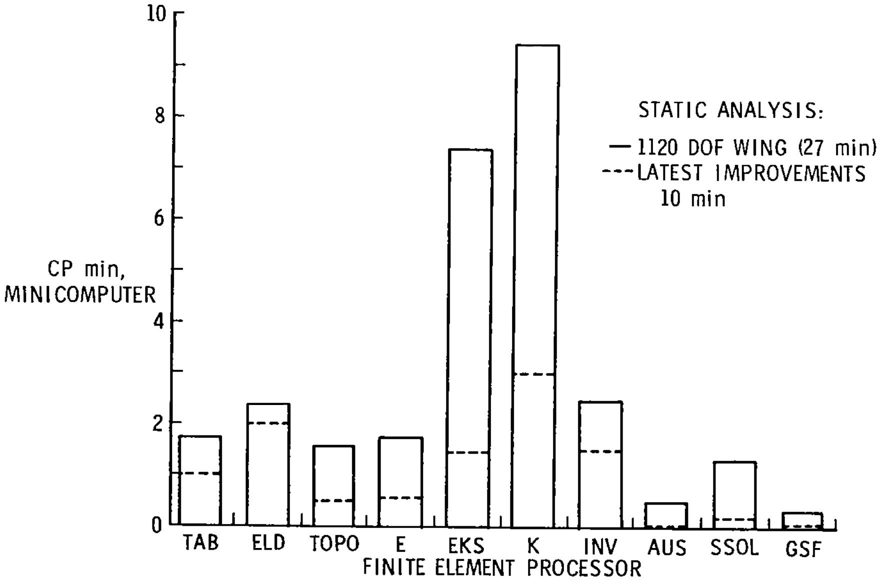


Fig. 7 Finite element processor time distribution (wing model).

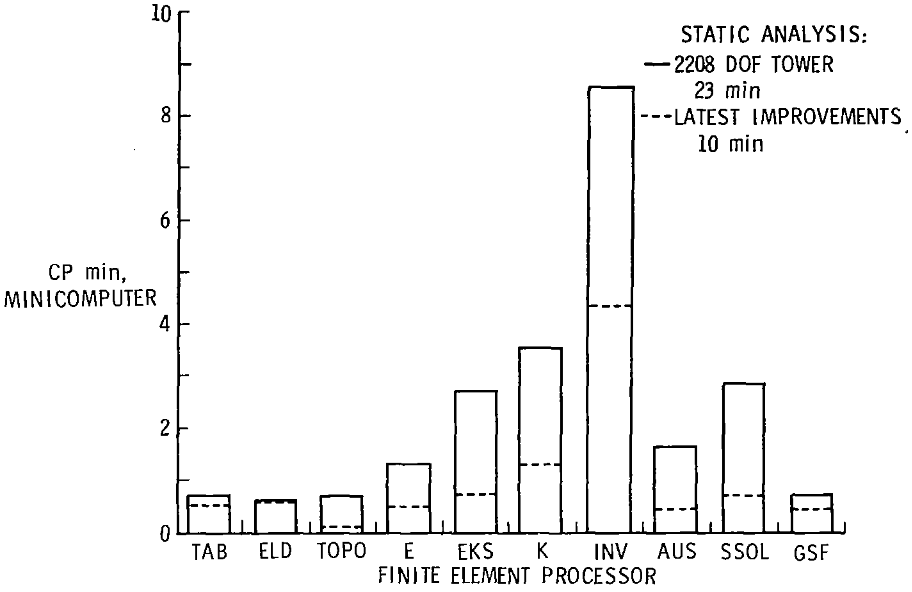


Fig. 8 Finite element processor time distribution (tower model).

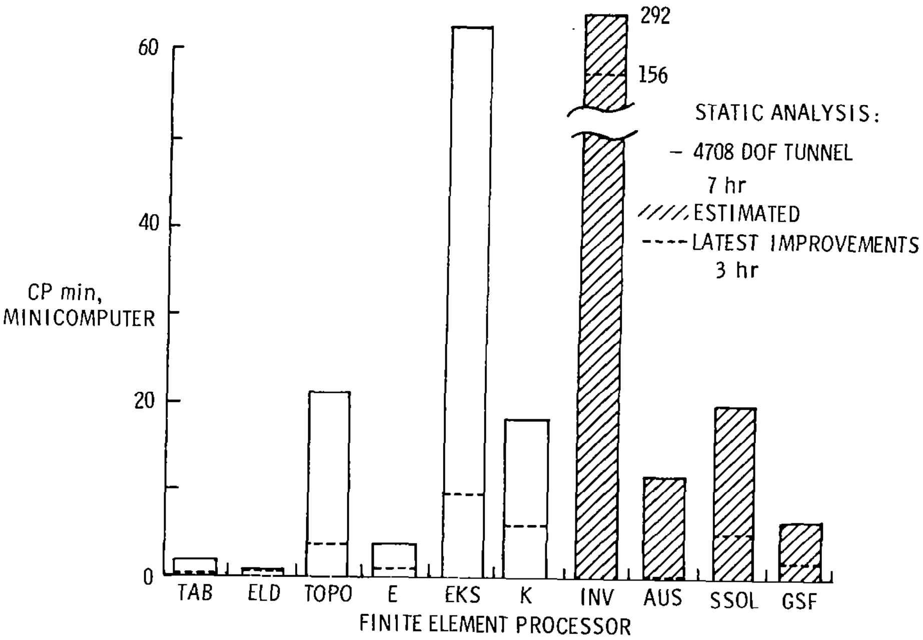


Fig. 9 Finite element processor time distribution (tunnel model).