# CSM PARALLEL STRUCTURAL METHODS RESEARCH

Olaf O. Storaasli *

NASA Langley Research Center

## INTRODUCTION

For many years, the Structures and Dynamics Division at NASA Langley has conducted research on the forefront of parallel and high-speed scientific computing by developing innovative software and hardware to speed up finite element structural analysis computations.

Ten years ago, using some of the first microcomputers, Langley researchers performed some of the first, if not the first, parallel structural analysis computations using four computers (connected together and coded in assembly language) to solve the finite element beam bending equations. This early experience subsequently led to the development by Langley of the Finite Element Machine (ref. 5), a more comprehensive parallel computer with up to 36 processors. With this parallel computer, significant reductions in computation time were achieved for widespread applications including matrix equation solution, dynamic transient analysis, eigensolution and nonlinear elasto-plastic yield surface computations.

When the first commercial parallel computers appeared, Langley purchased a 20-processor FLEX/32 and began Computational Structural Mechanics (CSM) parallel methods research on it.

This presentation summarizes ~~our~~ CSM Parallel Structural Methods Research and provides an introduction for six members of our research team who will speak today (Drs. Robert Larson, Jim Ortega, Alan George, Harry Jordan, Terry Pratt and Merrell Patrick) and Phil Underwood who will speak tomorrow.

* Aerospace Engineer, Structural Mechanics Branch, Structures and Dynamics Division.

# LANGLEY CSM PROGRAM

Parallel Structural Methods Research - Olaf O. Storaasli

Testbed Development - Ronnie Gillian

Methods and Applications Studies - Norm Knight

# LANGLEY CSM PROGRAM

The Langley Computational Structural Mechanics (CSM) program consists of the three primary components shown on this slide. Today we will discuss the work of the Parallel Structural Methods Research team that I lead. Tomorrow the Testbed and Methods and Applications work led by Ronnie and Norm will be discussed.

In addition to these primary CSM research thrusts, some cooperative work is being conducted with the Structural Dynamics Branch which will be addressed Friday morning by Dr. Jerrold Housner.

# OUTLINE

Objective and Approach

Team Research Strategy

Parallel Architectures and Software

CSM Focus Problems

Typical Results

Future Directions

# OUTLINE

This talk is organized to address the six items shown on this outline. These items should cover the major aspects of our research work as well as provide a suitable background and introduction for the other team members who speak after me. Although I will present a sample of some of our typical results in selected areas, other team members will provide more detailed results from their specific research area.

# PARALLEL STRUCTURAL METHODS

Objective: To develop structural analysis methods for parallel computers.

Approach: Design, develop and implement computational utilities, solution methods and languages for a parallel processing environment.

Evaluate and compare parallel methods by solving CSM focus problems.

Incorporate methods in testbed software.

# PARALLEL STRUCTURAL METHODS

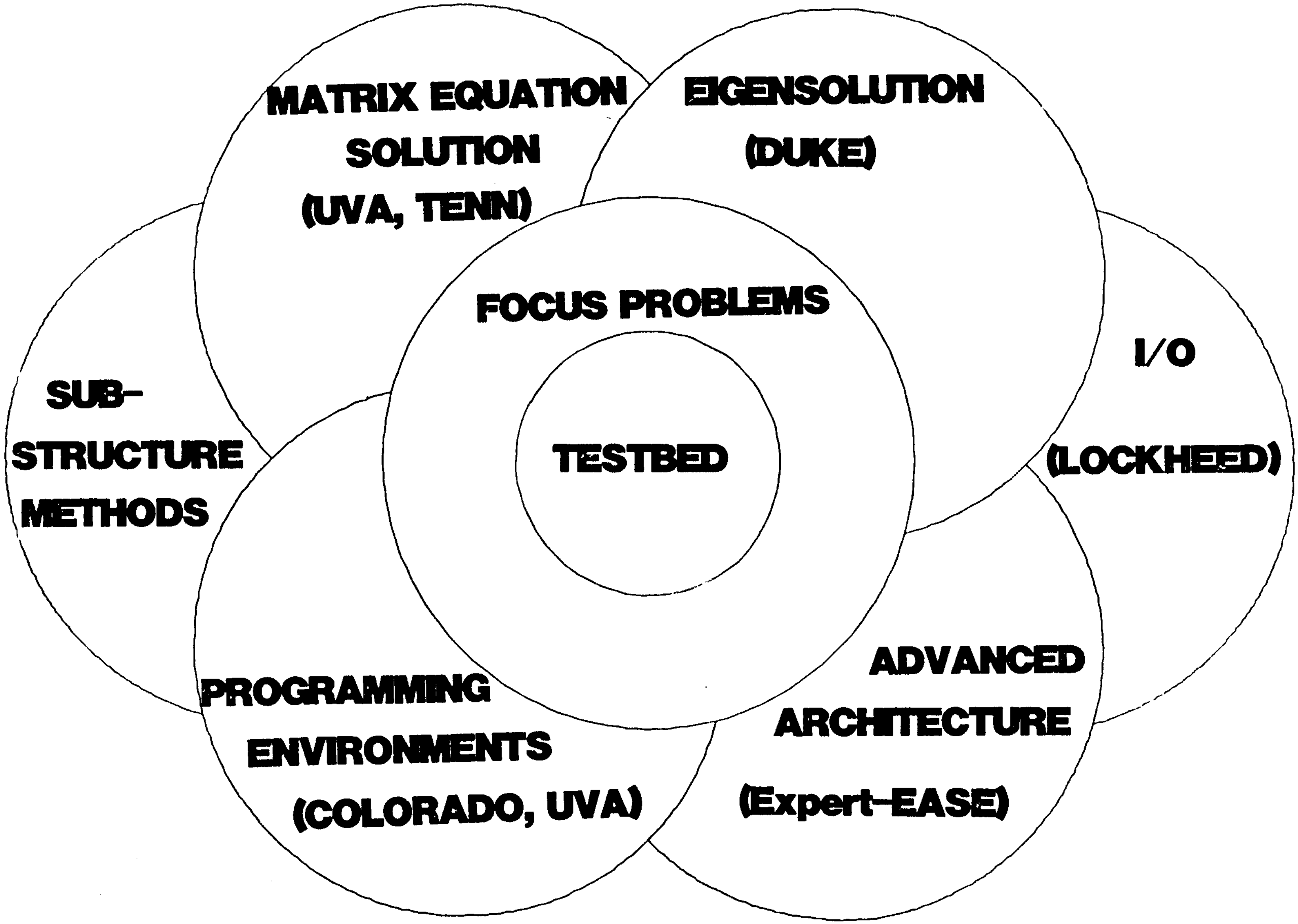
The advanced computer architectures of today and tomorrow, from CRAYs to Connection Machines share one common feature to achieve their high speed and performance: multiple processors. The limits of performance of single processor architectures are nearing their theoretical limit based on speed of light constraints, and before long, desktop versions of the fastest single processor supercomputers should be available.

However, the highest speed scientific computers both in the supercomputer and near supercomputer range will have many processors. The capability to develop structural analysis methods to take advantage of these parallel computers is the objective of the CSM Parallel Structural Methods Research.

The approach is to design, develop and implement in the CSM Testbed parallel solution algorithms for structural analysis. Certain computational utilities and languages have been developed to support the efficient development and performance of these new algorithms such that they are portable to other parallel computers.

The methods are incorporated in the Testbed software and their performance evaluated in their use to solve the CSM focus problems.

# PARALLEL METHODS RESEARCH TEAM



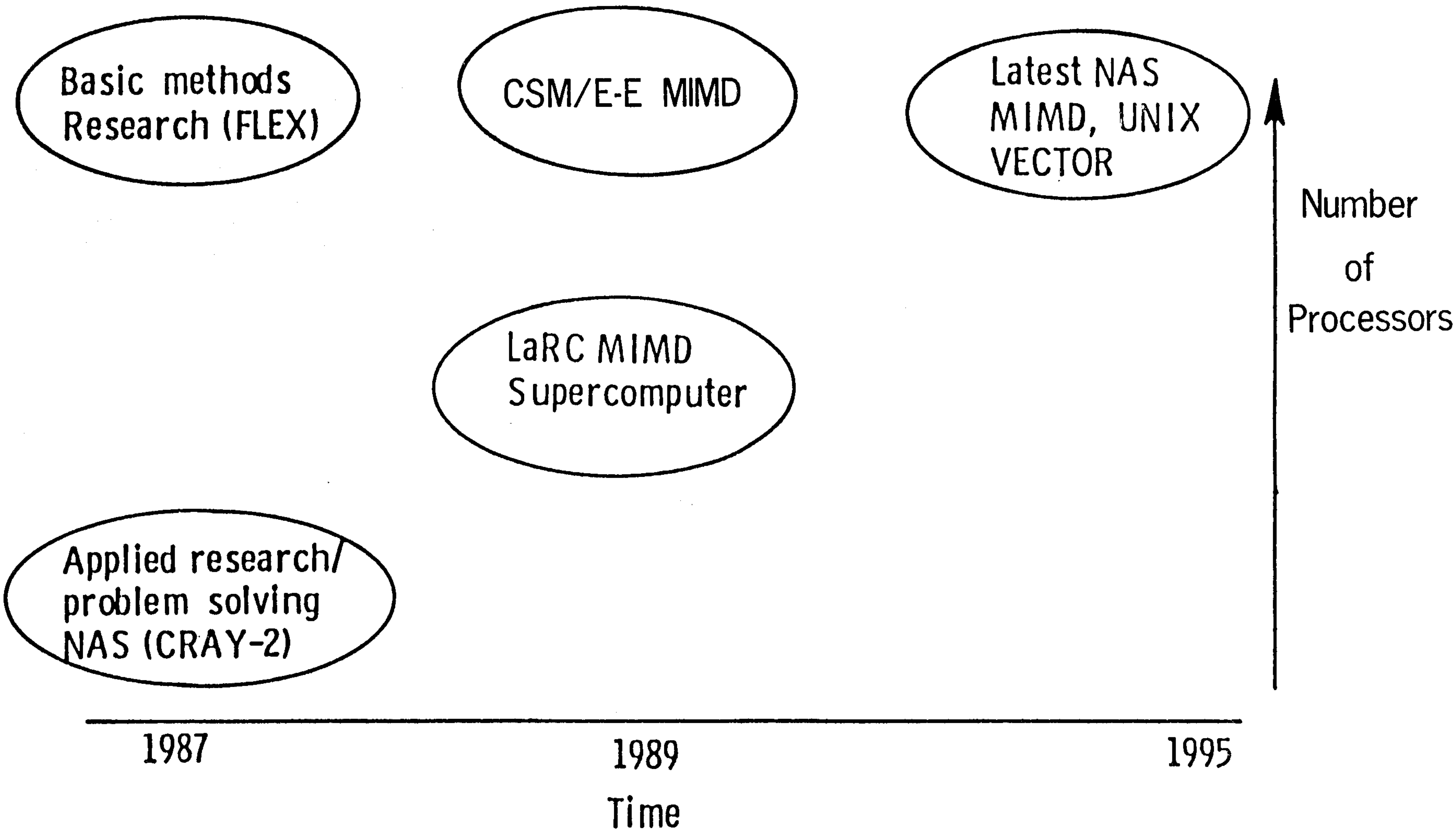
# PARALLEL METHODS RESEARCH TEAM

The parallel structural methods research team includes approximately 20 full and part-time NASA, grant and contract personnel located both on and off-site. All team members are working on parallel methods to improve the performance of the CSM Testbed for the CSM focus problems, as shown in the center of this slide. The areas being addressed by new algorithms being developed are indicated in the outer circles. Under Matrix Equation Solution, both direct (Choleski, Gauss Elimination) and iterative (Preconditioned Conjugate Gradient) parallel solution methods have been developed which significantly outperform the sequential testbed solver on multiprocessors and in some cases rival testbed performance even on one processor. For Eigensolution, three methods (Lanczos, Subspace Iteration, and Parallel Sectioning) have been developed and shown to speedup eigenvalue computation with the Lanczos giving the greatest time reductions (even for one processor). A limited substructuring capability has been developed and tested in the sequential Testbed, giving correct results, but as expected, longer solution times. These times are expected to be reduced when the matrix generation processors can be run in parallel, thus permitting parallel substructuring.

Two parallel programming environments (Force and Pisces) have been developed to simplify parallel programming and to offer portability to other parallel computers. These systems permit structural analysis software written on one parallel computer to run on other parallel computers with no changes required to the parallel software. The most promising algorithms are currently coded in Force to allow portability across Flexible, Alliant, Encore, Sequent, Cray and HEP parallel computers.

Work at both Lockheed and on grant with ICASE at Langley is addressing methods to minimize the time spent for data management and I/O by accomplishing these functions in parallel for large finite element applications. Finally, an advanced architecture parallel computer design based on a chordal ring of Inmos T-800 processors is planned for delivery to CSM in 1989. It should contain at least 15 processors each with a 64-bit floating point unit and a peak performance of approximately 90 million Whetstones for a total system peak performance of 34 MFLOPS.

## ADVANCED ARCHITECTURE COMPUTERS FOR CSM PARALLEL STRUCTURAL METHODS RESEARCH



# ADVANCED ARCHITECTURE COMPUTERS FOR PARALLEL CSM RESEARCH

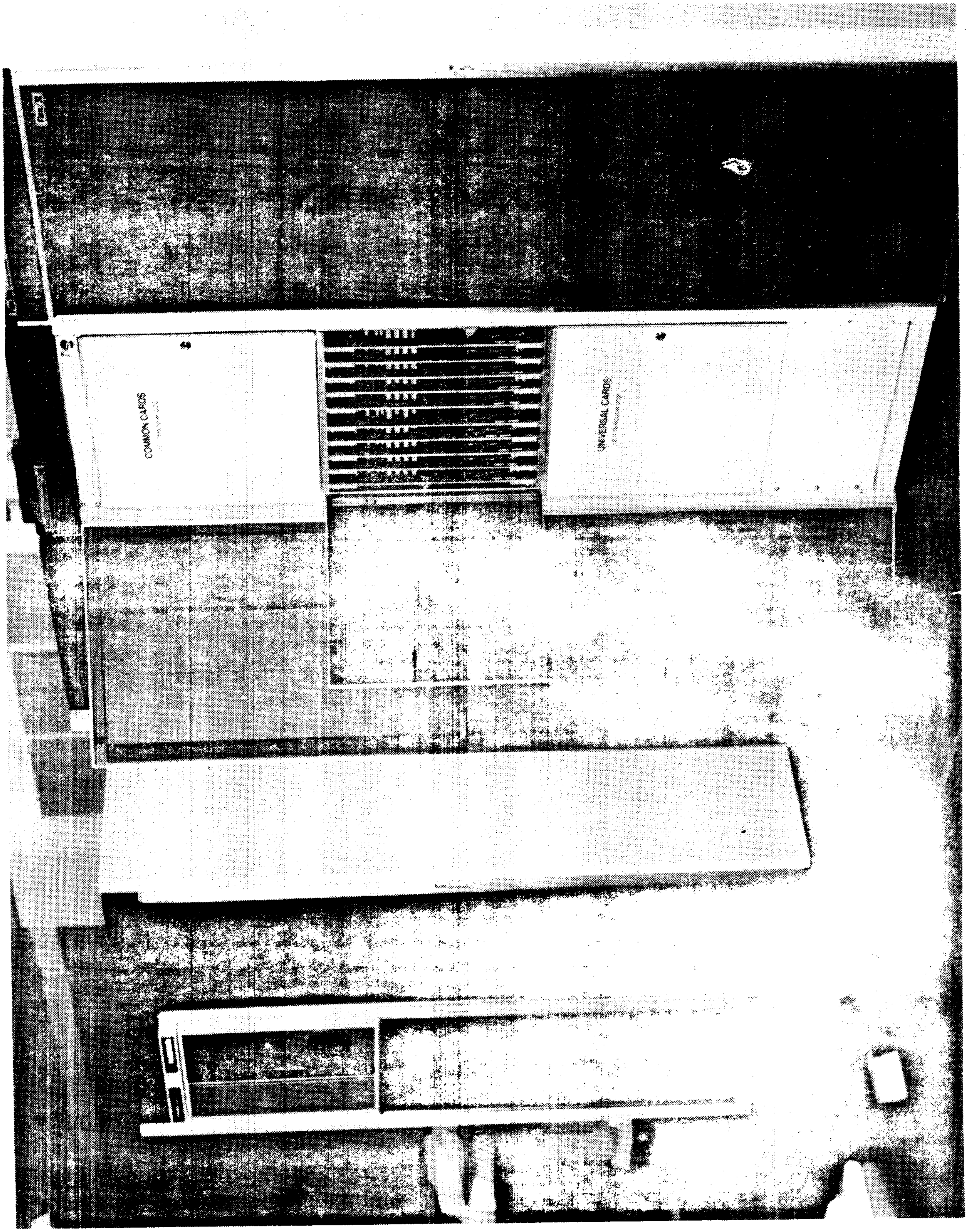
The intent is for the Parallel Structural Analysis Algorithms developed on today's parallel computers to migrate to the parallel and supercomputers of tomorrow. This slide shows today (1987), the near-term (1989) and the far-term future on the abscissa. On the ordinate is shown the number of processors (or level of parallelism) in advanced architecture computers.

The circle on the right labelled "Latest NAS" indicates the type and characteristics of what is anticipated to be the most advanced supercomputer at that time. There is little question that to achieve high-speed performance, supercomputers will be multiple instruction multiple data (MIMD) architecture with 16 or more vector processing units. To minimize development costs and maximize user acceptance, compatibility and portability, the UNIX operating system and utilities will be required.

On ~~the left are the two~~ **advanced architecture multiprocessor computers primarily used** in CSM parallel methods work. The CRAY-2 has four processors with a single path to shared memory and is useful for vectorization and timing studies and may become more useful for parallel research in the future when a compiler supporting parallel constructs is available. The FLEX/32 with 20 processors and 84.5 MB of memory (both local and global) supports research on algorithms exploiting a significant number of processors (what we might expect with CRAY in the 1990s).

At least two additional computers are planned to be delivered to Langley in 1989 to significantly enhance CSM research in Parallel Structural Methods. In addition to these, certain testing of algorithms is also performed on parallel computers at grantee, contractor and other sites.

It is expected that by maintaining the capability to explore methods exploiting a significant number of processors as well as implementing on computers exhibiting the maximum speed for today, we shall be in a position to have algorithms with the proper characteristics to run most efficiently on the fastest scientific computers in the future.

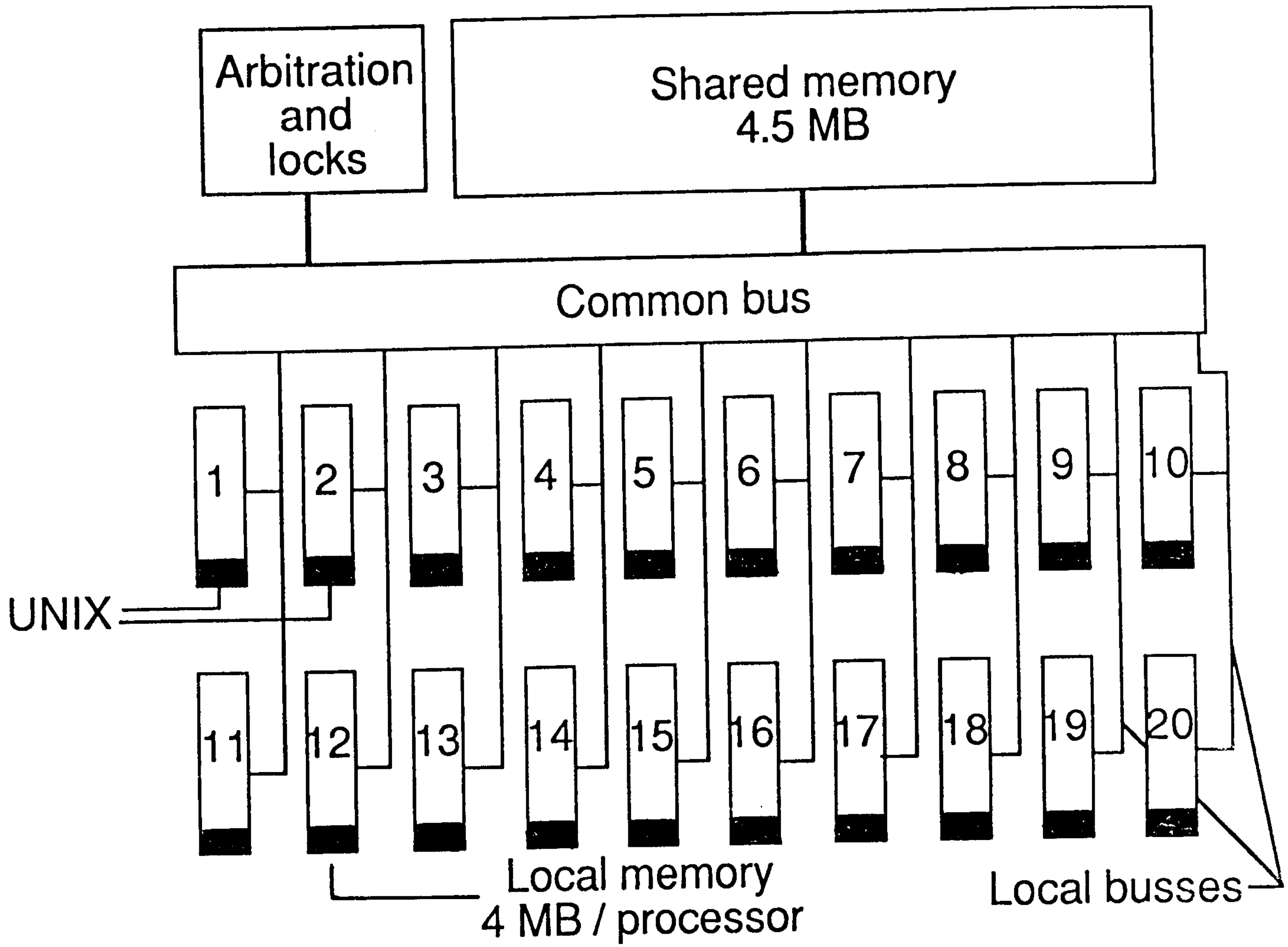

COMMON CARDS
UNIVERSAL CARDS

# FLEX/32 MULTICOMPUTER

This photo shows two FLEX/32 multicomputers. In the foreground is the primary National Semiconductor-based FLEX/32 on which nearly all the parallel structural methods research and development is taking place. It contains 20 processors, 10 of which can be seen by the open door and 10 more below them behind the door labelled Universal Cards. Each Universal card contains 4 MBytes of its own local memory for a total of 80 MBytes of local memory in the complete system. In addition there are 4.5 MBytes of common memory, shared by all processors, located behind the door labelled "Common Cards". The 8 disk drives, tape drive and communication hardware are located in the left third of the FLEX/32 in the foreground behind the large open door.

The FLEX/32 in the background was recently installed as part of a Phase 2 Small Business Initiative Research project. It contains only two Motorola 68020 processors each with high speed floating point units and the necessary disk and communications hardware to make the system operational for test purposes and to make comparisons with the primary FLEX computer system.

# FLEX/32 20-PROCESSOR CONFIGURATION



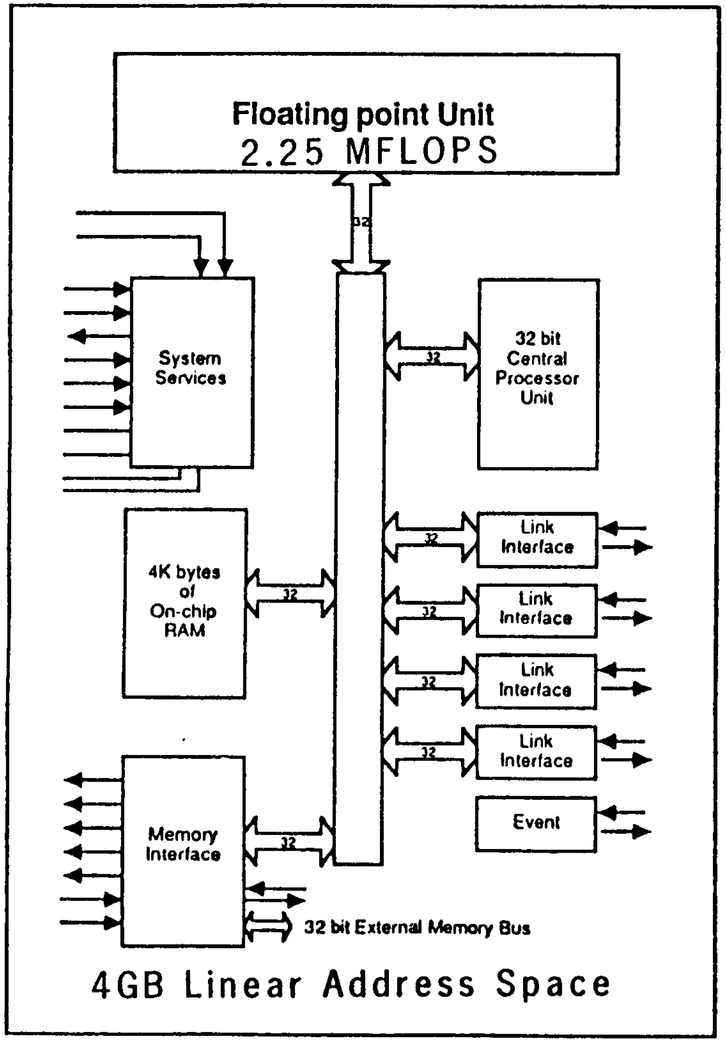
# FLEX/32 20-PROCESSOR CONFIGURATION

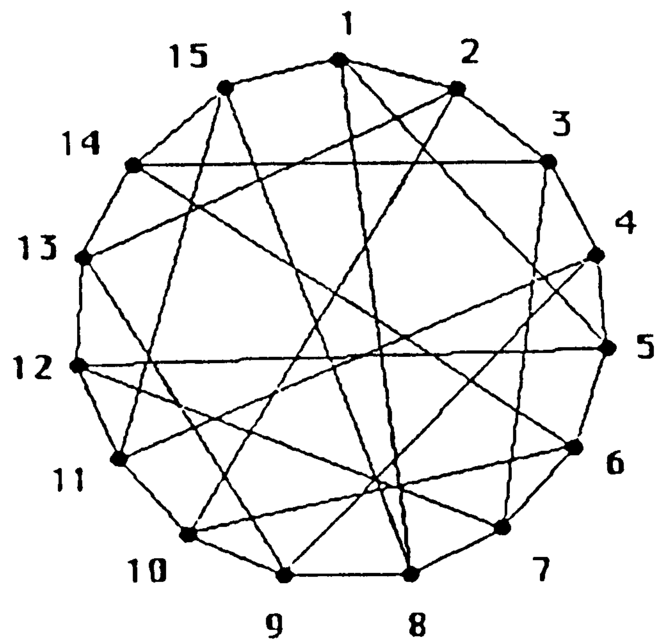
This slide shows the primary components of the primary FLEX/32 multicomputer. The 20 processors (labelled 1-20) are each shown to contain 4MB memory (black shading) and are connected to a common bus in addition to being paired via local busses. Two processors (labelled 1 and 2) are used for user program development using the UNIX operating system while the remaining 18 processors are available to run parallel programs. Two parallel programs may run simultaneously if they each require nine or fewer processors and programs requiring from 9 to 18 processors take the entire parallel array, although the two UNIX processors may still be used simultaneously for program development. All processors can access the shared memory with the restriction that the arbitration and lock mechanism used prevents simultaneous contention for the same memory location by more than one processor. Disks are available on both the UNIX and parallel processors and a "virtual I/O" capability exists to access disks from a processor not connected to that disk.

# MULTI-MICROPROCESSOR COMPUTER ARCHITECTURE

## - FAMILY OF LOW-COST SUPERCOMPUTERS -

IMS T800




| processor | | Whetstones/second single length |
|---|---|---|
| Intel 80286/80287 | 8 MHz | 300K |
| IMS T414-20 | 20 MHz | 663K |
| NS 32332-32081 | 15 MHz | 728K |
| MC 68020/68881 | 16/12 MHz | 755K |
| ATT 32000/32100 | | 1000K |
| Fairchild Clipper | 33 MHz | 2220K |
| IMS T800-20 | 20 MHz | 4000K |
| IMS T800-30 | 30 MHz | 6000K |

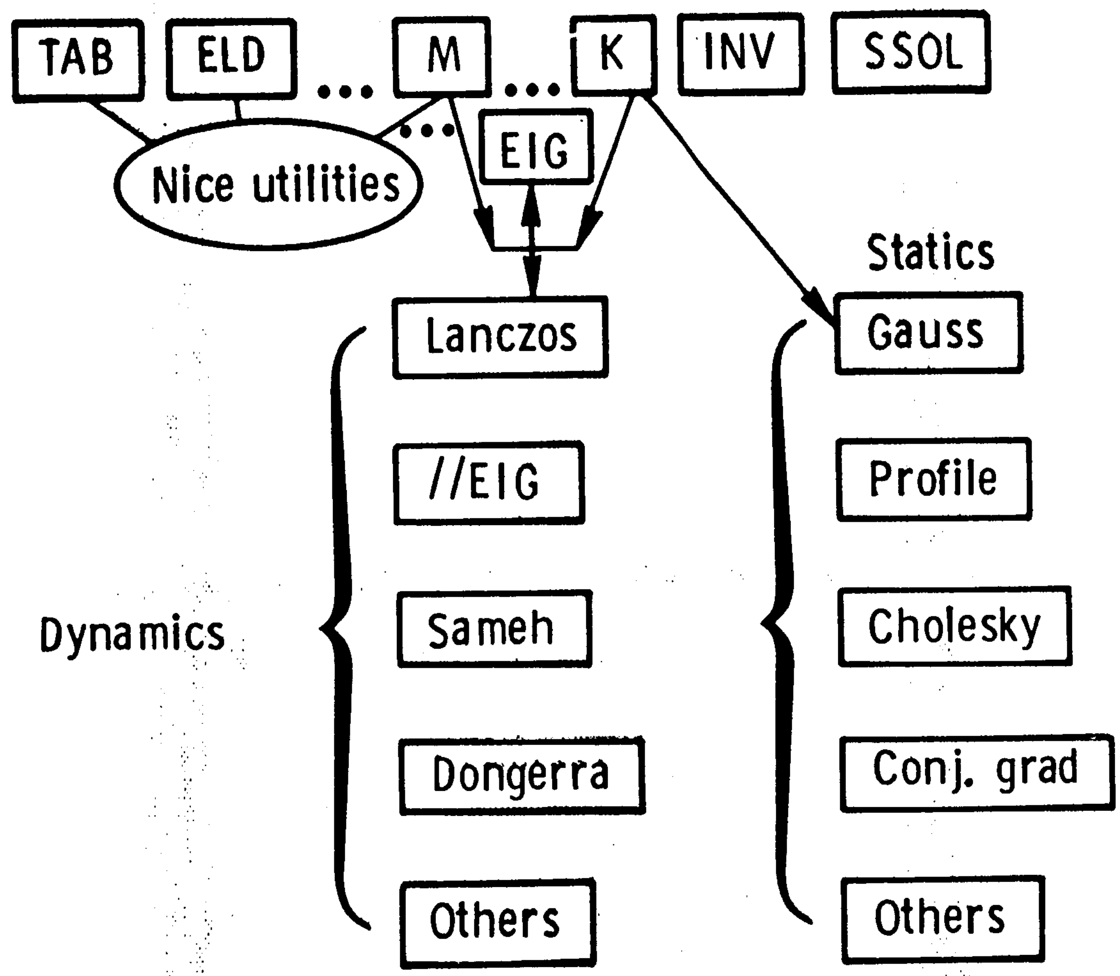
# MULTI-MICROPROCESSOR COMPUTER ARCHITECTURE

This slide shows three aspects of a new concept referred to as "chordal ring" for the design of a new family of low-cost supercomputers of the future. Each node of the chordal ring consists of an Inmos T800 computer (left), complete with 2.25 MFLOPS floating point unit, four links to companion computers, limited on-chip memory, four memory interface links to companion processors and on-chip system software.

The Inmos T800 is the fastest single chip computer on the market, (see comparison-lower right) and it surpasses the most frequently used processors by nearly an order of magnitude by performing 6 million Whetstones per second. The Whetstone benchmark, like LINPACK and other benchmarks, is typical of the programs in use by scientific and engineering users and is found to be more meaningful than millions of floating point operations per second (MFLOPS) or million instructions per second (MIPS) frequently quoted by computer manufacturers but of little credence for most "real" engineering applications.

On the upper right is shown a chordal ring with 15 nodes each sharing connections with 4 neighboring nodes and having a minimum hop length of 3 to travel from one node to any other node. The simulations performed to date show the chordal ring superior to the hypercubes in common use. Plans are to exploit their use by evaluating parallel structures algorithms from the CSM Testbed on them.

# PARALLEL STRUCTURAL ANALYSIS TESTBED



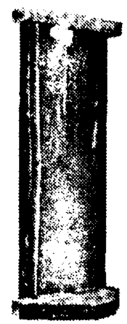
# PARALLEL STRUCTURAL ANALYSIS TESTBED

The organization of the CSM sequential Testbed (above) and the strategy used to add new parallel modules to it (below) is shown in this slide. The CSM sequential Testbed with several typical (SPAR) modules is shown from left to right at the top (TAB...SSOL). The data base and command utilities (labelled NICE utilities) used by the Testbed processors are shown in the oval on the left just below the processors. The method by which the SPAR processors communicate is via data sets written to and read from the data library (a disk file). Thus, for example, the processor M generates the mass matrix while K generates the stiffness matrix. Both the K and M processors perform their respective functions as a result of a user command which causes them to access data sets (containing geometric, material and element data) already written in the data library by previously run processors such as TAB and ELD. Processor INV performs a forward decomposition of K, and SSOL performs a back substitution to calculate the static solution for displacements.

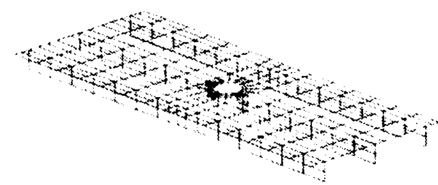
The new parallel algorithms for dynamics (eigensolutions for vibration analysis) and statics (matrix equation solution) are shown at the bottom to the left and right, respectively. Each new parallel dynamics method (shown by one box) replaces both the INV and EIG processors of the sequential Testbed. The new parallel solution methods read the K and M matrices directly from the Testbed data library just as other sequential testbed processors. Thus, on a parallel computer, the user has a choice of running the sequential algorithm used by the CSM Testbed or a parallel algorithm offering equivalent accuracy and a reduced computation time.

# BLADE-STIFFENED GRAPHITE-EPOXY WITH A DISCONTINUOUS STIFFENER

- Graphite-epoxy (T300/5208)
- Flat panel with three blade stiffeners
- 30 in. long
- 11.5 in. wide
- 2.0 in. diameter hole
- 25-ply panel skin
- 24-ply blade stiffeners
- Axially loaded with loaded ends clamped and sides free



FINITE ELEMENT MODEL



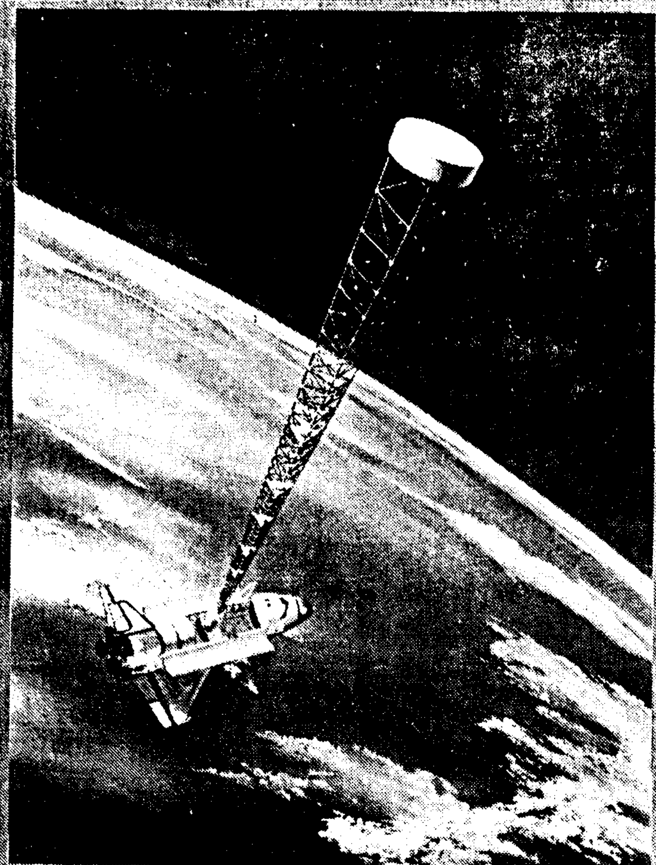
ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

# STIFFENED PANEL WITH CIRCULAR CUTOUT

A model of a 76.2 cm by 29.2 cm rectangular blade-stiffened aluminum panel with a 5.1 cm hole in the center is shown on this slide. It contains three 3.56 cm high stiffeners spaced 11.43 cm apart. The thicknesses of the plate and stiffeners are 0.254 cm. A more detailed description of the finite element model used (including input data) is contained in Appendix C of reference 10.

Three finite element models of this stiffened panel were used in this study: a coarse model (648 degrees-of-freedom), a medium-sized model (2328 degrees-of-freedom), and a detailed model (4392 degrees-of-freedom). The stiffness matrix for the coarse 108-node model of the panel contains 476 rows with a semi-bandwidth of 118, while that for the 2328 degree-of-freedom model has 1824 rows with a semi-bandwidth of 240. The behavior of these three stiffened panel models as well as a complementary space mast problem were used to evaluate the performance of the linear equation solvers and eigenvalue solvers developed for use on parallel computers.

# SPACE MAST PROBLEM



60 meter three-longeron truss
54 bays
165 nodes
486 degrees of freedom

# SPACE MAST PROBLEM

The space Mast problem is based on a proposed space shuttle experiment to explore the dynamic characteristics of a 60-meter, 54-bay, 3-longeron deployable truss beam shown at the left. The finite element model contained 165 nodes and 486 degrees-of-freedom. Details of the model definition are contained in reference 8. The stiffness matrix for this model had a semi-bandwidth of only 18, considerably smaller than that for the stiffened panel.

A second, more detailed one-third length beam model of the space Mast with mid-point hinges, referred to as the Mini-Mast, was deployed and tested for static and dynamic characteristics at the NASA Langley Research Center. The Mini-Mast (ref. 9) consists of 18 bays containing thin graphite-epoxy tubular longerons, battens and diagonal members each with titanium tip connectors. The mass of the 111 titanium joints (0.7775 kg) is significant when compared to the light-weight tubular members. Thus the Mini-Mast is referred to as a "joint-dominated structure". The Mini-Mast is fixed at the three base points leaving 1980 of the total 1998 degrees-of-freedom active in the solution. Examination of the element interconnection pattern for joints reveals that the Mini-Mast has a small bandwidth (60) when compared to the panel problems (118 and 240).

# SPACE STATION: POTENTIAL FOCUS PROBLEM

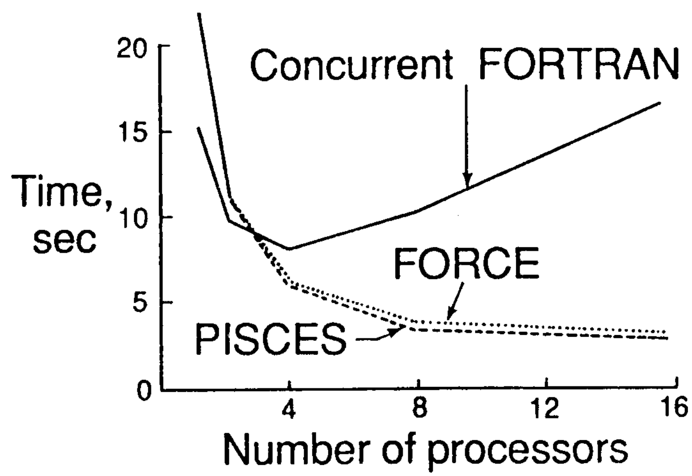
The blade-stiffened panel and space Mast focus problems contain sufficient complexity to evaluate new methods and software. However, additional potential focus problems are being evaluated including a model of the space station shown in this slide. A finite element model was converted from NASTRAN format to Testbed format with the exception of some material properties which were defined in an non-standard manner in the NASTRAN model.

The finite element model is useful for both static and dynamic analysis and contains 2328 degrees of freedom. The lightweight solar arrays and certain other appendages are modeled as beams with equivalent properties. Although the space station model does not contain a large number of degrees of freedom, it is a natural step beyond the smaller space Mast problems.

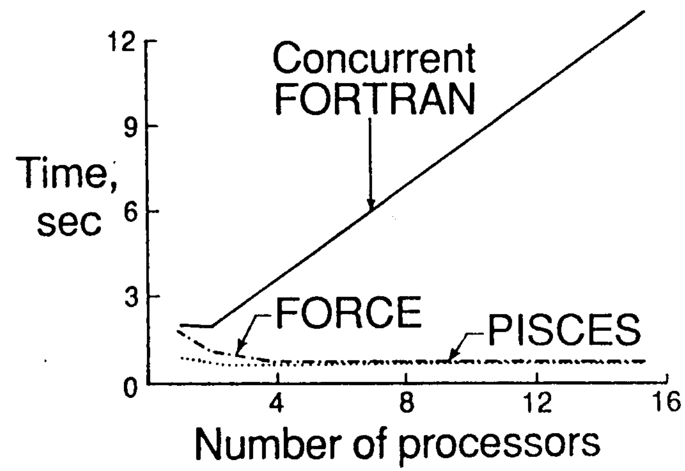
# PISCES AND FORCE REDUCE MATRIX EQUATION SOLUTION TIME

Problem: No parallel language standards or portability
*FORCE: FORTRAN extensions (U. of Colorado)
PISCES: Parallel programming environment with FORTRAN extensions (UVa)
Solve: 200 x 200 matrix with semi-bandwidth 50 (Duke, ICASE)

**FACTORIZATION**




**BACK SUBSTITUTION**




* Offers portability across FLEX, ENCORE, ALLIANT, HEP, SEQUENT

# PISCES AND FORCE REDUCE SOLUTION TIME

This slide describes an important way to achieve significant performance improvements on parallel computers. Until the development of Force and Pisces, there has been little accomplished to assist software developers on parallel computers to achieve efficient code while maintaining portability. In fact many feel significant efficiency must be sacrificed to achieve portability.

There is currently no standard for parallel language constructs and probably won't be until FORTRAN 9X is defined and agreed upon. In the meantime, Force was developed by Harry Jordan at the University of Colorado which maps a common parallel language (Force) into the differing parallel languages existing on different parallel computers. The method used the UNIX SED (stream editor) as a precompiler to convert the common parallel language to the corresponding parallel language (i.e. concurrent FORTRAN) supplied by each vendor for their computer.

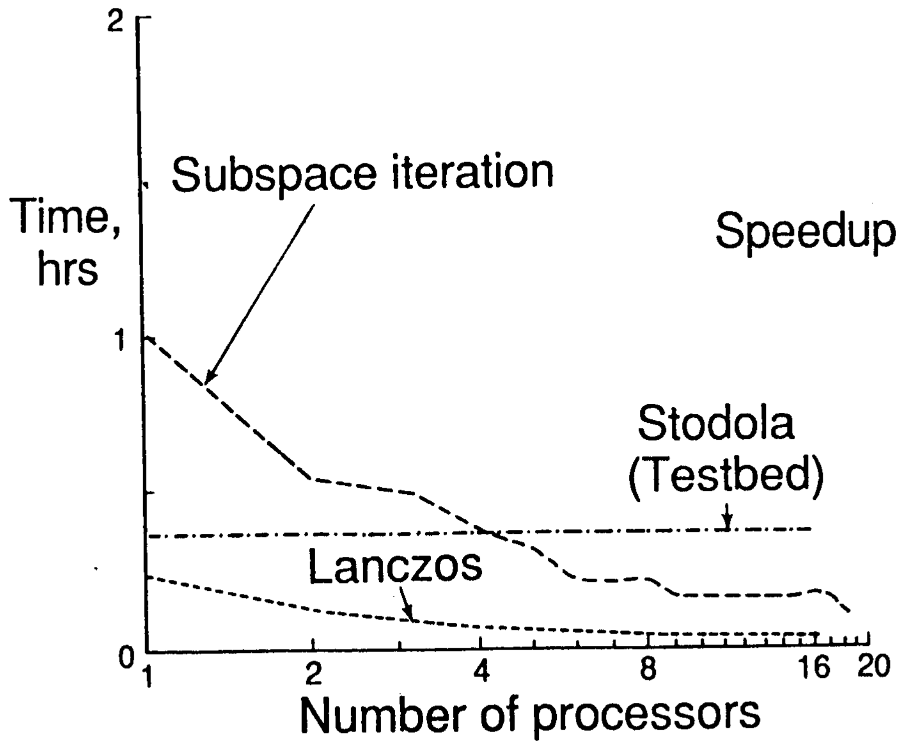
The slide shows that in addition to gaining portability across computers (including Flexible, Encore, Alliant, HEP, Sequent and Cray) the performance achieved by Force and Pisces exceeded the performance of the vendor's Concurrent FORTRAN by a significant amount for both the factorization and back substitution portions of the matrix equation solution. The trends actually degrade seriously for concurrent FORTRAN for beyond just a few processors, while the situation continues to improve for Force and Pisces.

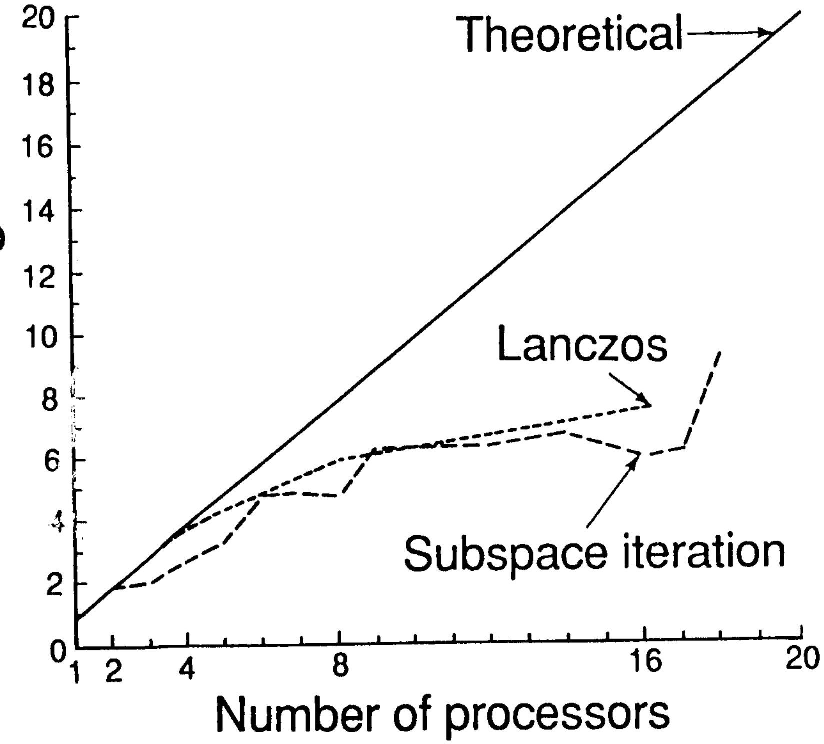
Code written in Force requires no changes to run on another parallel computer on which Force is running. Thus, new solution methods developed on the Flexible/32 research parallel computer can be transferred to and run on the Cray 2 without changes. All those developing new parallel algorithms for CSM are encouraged to use Force for both ease and commonality of coding and portability to other parallel computers.

# PARALLEL EIGENSOLVERS REDUCE SOLUTION TIME

Solve: $Kx = \lambda Mx$

Example: 10 lowest vibration frequencies of graphite-epoxy panel by Lanczos (in-house) and sub-space iteration (Duke University)



# PARALLEL EIGENSOLVERS REDUCE SOLUTION TIME

The determination of the fundamental (lowest) natural vibration frequencies and associated mode shapes is a key step used to uncover and correct potential failures or problem areas in most complex structures. However, the computation time taken by finite element codes to evaluate these natural frequencies is significant, often the most computationally intensive part of structural analysis calculations. There is a continuing need to reduce this computation time. This slide shows significant reductions in computation time achieved by two parallel eigensolution methods. The objective of both the Lanczos and Subspace Iteration method is to solve the eigenvalue problem $Kx = xMx$ for the ten lowest vibration frequencies for the stiffened panel CSM focus problem.

The plot on the left shows the reduction in computation time achieved by the Lanczos method and the subspace iteration as the number of processors increases. Despite the slightly greater reduction achieved by the subspace iteration method, the Lanczos method took less time overall than the subspace iteration method and the sequential Testbed solver (horizontal line) regardless of the number of processors.

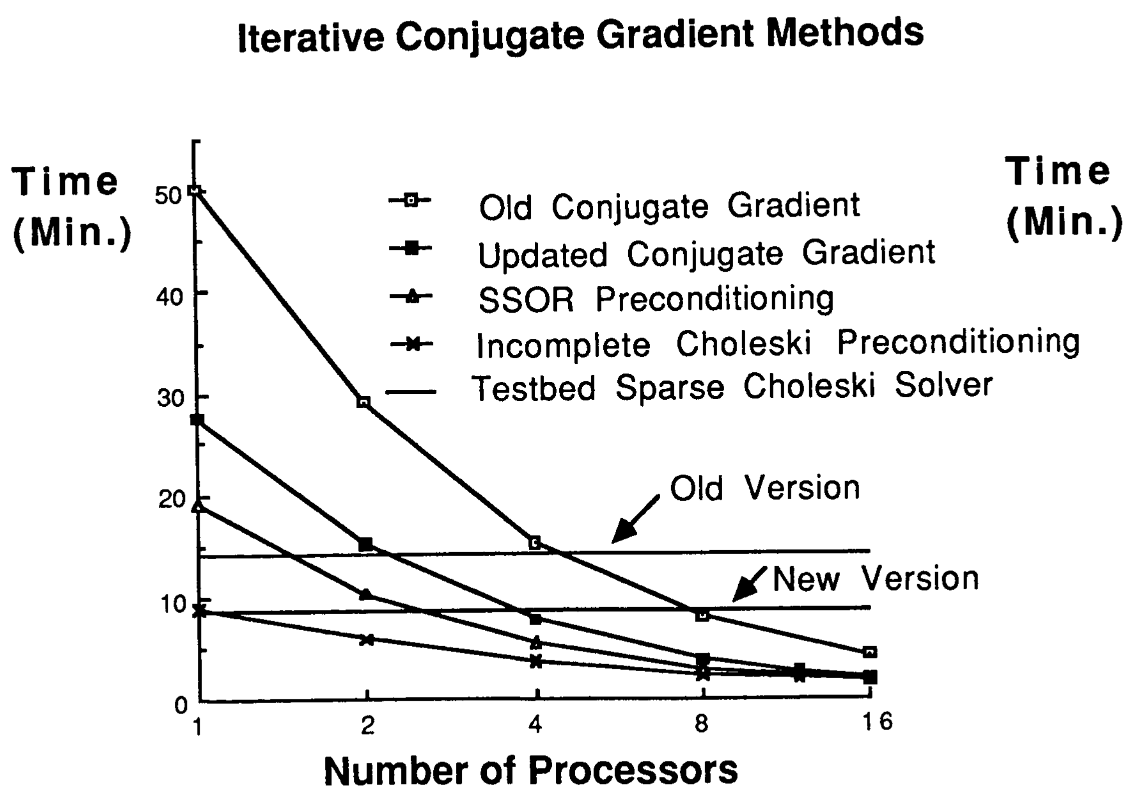
The computation speedup for each method is shown on the right plot with an increasing number of processors along the abscissa. Both methods fall below the theoretical limit (maximum speedup) which indicates further reductions in computation time may be possible by introducing further refinements to the parallel methods.

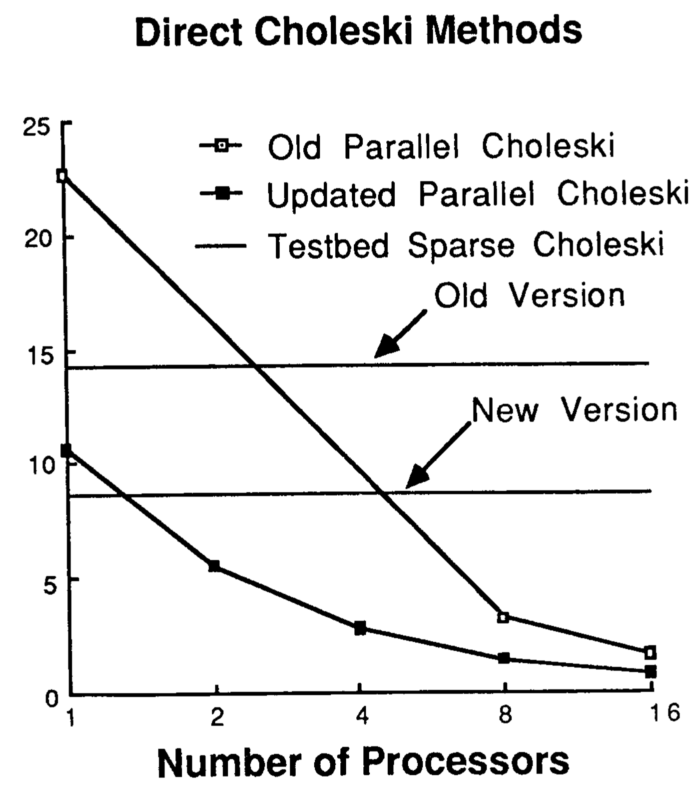
Further details of these results can be found in reference 2.

# Preconditioners and Refined Code Improve Performance of Equation Solvers

Solve Ku = f

Comparison of Execution Times for Panel Example Problem with 1824 Equations



# PARALLEL EQUATION SOLVERS REDUCE SOLUTION TIME

The solution of the system of load-displacement equations, $K u = f$ is often the most time consuming portion of the solution of linear finite element structural analysis problems. As the size of the stiffness matrix, $K$, increases, the proportion of time spent in equation solution increases at a superlinear rate, approaching in excess of 90 percent of the time spent for larger structural analysis problems. Reductions achieved in computation of the structural displacements have the direct benefit of permitting the solution of larger more complex problems, without resorting to the simplifying assumptions used by many to bypass large computation times for the static structural analysis.

Results obtained for two methods, Conjugate gradient and Choleski are shown in the plot at the lower left. For this 1824 degree-of-freedom stiffened panel problem, the best times obtained by iterative (left) and direct (right) solvers for one processor are competitive with the sparse Choleski solver in the new version of the Testbed.. More significant is that all parallel solvers developed eventually are faster than the new Testbed solver, with the Incomplete Choleski Preconditioned Conjugate Gradient method performing the best for the iterative methods and the Updated Parallel Choleski performing best for the direct methods. These results are typical in that smaller problems with smaller bandwidths may not perform as well in parallel while larger problems with the same or increased bandwidths perform even better.

# PARALLEL STRUCTURAL METHODS: FUTURE

- Solve Nonlinear & Buckling Problems
- Parallel Matrix Generation (Element, Stiffness & Mass)
- Parallel Substructuring Methods
- Portability across Parallel Computers

# PARALLEL STRUCTURAL METHODS: FUTURE

Our future work in parallel structural methods includes many facets ranging from parallel and vector structural analysis methods on the Cray 2 to development of a new architecture (chordal ring) 15-processor parallel computer based on high-speed transputer processors. However, the four items shown on this slide represent major directions the parallel structural methods group plans for the future.

Based on the success achieved for the parallel solution of linear static and vibration analysis problems, we plan to extend our methods to include nonlinear analysis and buckling problems. The nonlinear analysis consists of repeated linear analyses to exploit our new efficient linear solvers controlled by an arc-length (or other) search methods. The buckling analysis algorithm is quite similar to the vibration analysis methods (Lanczos and Subspace iteration) where the Kg matrix is used in place of the mass matrix.

A second direction is to extend the benefits of parallel solution to the generation of elemental and global stiffness matrices and mass matrices. Although major computation time reductions are not expected for this, it is a step towards total parallelism and achieving improved efficiency. Providing a parallel substructuring capability in which different substructures can be generated in parallel and the solution obtained either via the parallel solvers or a new parallel substructuring solver is a challenging objective. Plans are to add such parallel substructuring capability to the CSM testbed, based on the primitive "hooks" in the testbed used for modal synthesis.

The final item is to demonstrate portability of typical testbed processors across several parallel computers. Work is currently underway aimed at achieving this goal by using the Force extensions to concurrent Fortran (ref. 7).

# REFERENCES


1. Storaasli, O., Poole, E., Ortega, J., Cleary, A.and Vaughan, C., "Solution of Structural Analysis Problems on a Parallel Computer," AIAA Paper 88-2287-CP, Proceedings of the 29th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Williamsburg, Virginia, April 18-20,1988.

2. Storaasli, O., Bostic, S., Patrick, M., Mahajan, U. and Ma, S., "Three Parallel Computation Methods for Structural Vibration Analysis," AIAA Paper 88-2391-CP, Proceedings of the 29th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Williamsburg, Virginia, April 18-20, 1988.

3. Cleary, A., Harrar, D. and Ortega, J., "Gaussian Elimination and Choleski Factorization on the FLEX/32,", University of Virginia Applied Mathematics Report RM-86-13, 1986.

4. Bostic, S. and Fulton, R., "A Lanczos Eigenvalue Method on a Parallel Computer", AIAA Paper 87-0725-CP, Proceedings of the 28th AIAA/ASME/ASCE/AHS Structures, Structural Dynamics and Materials Conference, Monterey, California, April 6-8, 1987.

5. Storaasli, O., Ransom, J. and Fulton, R., "Structural Dynamic Analysis on a Parallel Computer: The Finite Element Machine," Computers and Structures, Vol. 26, No. 4, pp. 551-559, 1987.

6. Storaasli, O., Peebles, S., Knott, J., Crockett, T. and Adams, L., "The Finite Element Machine: An Experiment in Parallel Processing. NASA TM 84514, July, 1982, also in Research in Structural and Solid Mechanics, NASA CP 2245, pp. 201-217, 1982.

7. Jordan, H., Benten, M., Arenstorf, N. and Ramanan, A., "Force User's Manual," Department of Electrical and Computer Engineering Technical Report 86-1-4R, University of Colorado, Boulder, CO, October, 1987.

8. Horta, L., Walsh, J., Horner, G. and Bailey, J., "Analysis and Simulation of the MAST (COFS-1 Flight Hardware)," NASA CP 2447, Part 1, Nov. 18-21, 1986, pp. 515-532.

9. Adams, L., "Design, Development and Fabrication of a Deployable/Retractable Truss Beam Model for Large Space Structures Application," NASA CR 178287, June 1987.

10. Lotts, T. et al., "Introduction to the CSM Testbed: NICE/SPAR," NASA TM-89096, 1986.