



Performance evaluation of viscoelastic finite element supercomputer algorithms

S. Yi, H.H. Hilton, M.F. Ahmad

*National Center for Supercomputing Applications
and Aeronautical and Astronautical Engineering
Department, University of Illinois at
Urbana-Champaign Urbana, IL 61801, USA*

ABSTRACT

The primary objectives of this study are performance evaluations of dynamic viscoelastic finite element procedures and codes on vector and parallel processing machines. Parametric studies on various CRAY machine architectures and benchmarks for various problem sizes are also undertaken. CRAY hardware performance monitors such as Flowtrace and Perftrace tools are used to obtain performance data for subroutine program modules and specified code segments. The performance of element stiffness computations is in the range of 121-183 mega floating-point operations per second (Mflops) on the NCSA CRAY-YMP. The performances of the Cray sparse matrix and the Feable solvers are also evaluated.

INTRODUCTION

The finite element method is a very attractive technique for solving boundary and initial value problems and it has been providing researchers with powerful versatile means for solving complex problems in science and engineering. However, evaluations of significantly large scale problems and/or analyzing rate dependent systems which are governed by hereditary integrals or by high-order differential equations require large memory and computational times. With the advent of vector and parallel processing architectures, it is possible to solve such large and complex problems

498 Applications of Supercomputers in Engineering

more effectively, since vector and parallel process computers can provide increased capabilities in both computational speed and memory.

High performance computing leads to accurate and efficient implementation of finite element analyses and utilization of vector and coarse grained parallel processing machine is expected to improve code performances. In order to optimize large codes, hardware performance analyses are needed to point out sections and/or subroutines of codes where vectorization and parallelization are both useful and feasible, and where programs need to be restructured. Codes can be vectorized and parallelized on coarse grained machines (CRAY Y-MP, CRAY Y-MP C-90) using the full utilization of compiler tools such as fpp and fmp as well as visual vectorization and parallelization tools, such as perfview and atexpert.

The most computer intensive areas in viscoelastic FEA are those which calculate the element stiffness matrices and those which factorize and solve global system matrices. Viscoelastic problem computations, of course, require much larger computational times than do corresponding elastic ones. Even under quasi-static loads large numbers of viscoelastic time solutions must be obtained, whereas only one solution is needed for equivalent elastic problems. Using variational principles, the present authors [1,2,3] previously developed numerical algorithms for analyzing dynamic responses of hygrothermo-viscoelastic laminated composites and/or viscoelastically damped composite structures in the real time domain. For transient analyses, recursion formulas have been obtained which reduce computer storage and additionally require only two previous time solutions to compute succeeding time solutions.

Generally, the solution of sparse systems is the most computationally expensive task in finite element analyses and, therefore, efficient solvers must be used. Liu [4] reviewed frontal and multifrontal solvers for finite element analyses and Irons [5] and Hood [6] developed frontal methods for symmetric and nonsymmetric systems respectively. Detailed information on available sparse solvers can also be found in Ref. 7. However, some solvers tailored to scalar machines may not be efficient on vector and parallel processing machines. Available on CRAY systems are the SPARSE algorithms, used as sparse matrix solvers for solutions of real sparse symmetric and positive definite systems.

In the present paper, studies are undertaken to evaluate the performance of the VISSHELL finite element code [2,3] developed for analyzing dynamic responses of viscoelastic composite structures on vector and parallel processing machines. The present study also involves parametric studies on various CRAY machine architectures and benchmarks for various problem sizes are given. The CRAY hardware performance monitor is used and automatic parallelization is expected. The performance of solvers such as the Feable subroutine developed for structural analysis purposes at MIT in the early 70's and the SPARSE packages developed for the CRAY systems is evaluated.

ANALYSIS FOR VISCOELASTIC SOLIDS

The theory of linear thermo-viscoelasticity leads to the following integral constitutive equations [8]

$$\sigma_i(T, M, \mathbf{x}, t) = \int_{-\infty}^t Q_{ij}[T_o, M_o, \zeta_{ij}(\mathbf{x}, t) - \zeta'_{ij}(\mathbf{x}, \tau)] \frac{\partial}{\partial \tau} \epsilon_j(\mathbf{x}, \tau) d\tau \quad (1)$$

where

$$\zeta_{ij}(\mathbf{x}, t) = \int_0^t b_{ij} \left[T(\mathbf{x}, s), M(\mathbf{x}, s) \right] ds \quad (2)$$

and

$$\zeta'_{ij}(\mathbf{x}, \tau) = \int_0^\tau b_{ij} \left[T(\mathbf{x}, s), M(\mathbf{x}, s) \right] ds$$

are reduced times which reflect material memory of temperature T and moisture M histories, the subscript o denotes reference conditions, $\mathbf{x}$ are principal material coordinates, and Q_{ij} are relaxation moduli. The composite shell is assumed to be in a state of plane stress and relaxation moduli for an orthotropic composite lamina in the principal material directions are

$$[\mathbf{Q} (T, M, t)] = \begin{bmatrix} Q_{11} & Q_{12} & 0 & 0 & 0 & 0 \\ Q_{21} & Q_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & Q_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & Q_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & Q_{66} \end{bmatrix} \quad (3)$$

500 Applications of Supercomputers in Engineering

where $Q_{12} = Q_{21}$. These relaxation moduli are related to viscoelastic Young's an shear moduli and to Poisson's ratios and each may be temperature, moisture and time dependent

$$\begin{aligned} Q_{11} &= \frac{E_{11}}{(1 - \nu_{12}\nu_{21})} \\ Q_{22} &= \frac{E_{22}}{(1 - \nu_{12}\nu_{21})} \\ Q_{12} &= \frac{\nu_{12}E_{22}}{(1 - \nu_{12}\nu_{21})} \\ Q_{44} &= G_{12} \\ Q_{55} &= K_{23}^s \cdot G_{23} \\ Q_{66} &= K_{31}^s \cdot G_{31} \end{aligned} \quad (4)$$

where the K_{23}^s and K_{31}^s are shear correction factors. The relaxation moduli $\bar{Q}_{ij}$ with respect to the laminate axes can be obtained from coordinate transformations.

Finite element procedures for the dynamic analyses of anisotropic viscoelastic composite shell structures can be formulated by using degenerated 3-D elements. Displacements in each shell element are expressed in terms of nodal degrees of freedom

$$\begin{Bmatrix} u(\bar{\mathbf{x}}, t) \\ v(\bar{\mathbf{x}}, t) \\ w(\bar{\mathbf{x}}, t) \end{Bmatrix} = \sum_{i=1}^n N_i(\bar{\mathbf{x}}) \begin{Bmatrix} u_i(t) \\ v_i(t) \\ w_i(t) \end{Bmatrix} + \frac{\chi}{2} \sum_{i=1}^n N_i(\bar{\mathbf{x}}) \cdot h_i [\hat{\mathbf{v}}_{1i}, -\hat{\mathbf{v}}_{2i}] \cdot \begin{Bmatrix} \theta_i^1(t) \\ \theta_i^2(t) \end{Bmatrix} \quad (5)$$

where n is the number of nodes per element; $\bar{\mathbf{x}}$ are the laminate coordinates; h_i is a lamina thickness at the i th node; χ is the local curvilinear coordinate through the lamina thickness direction; u_i , v_i , w_i are nodal displacements; θ_i^1 , θ_i^2 are rotations; $N_i(\bar{\mathbf{x}})$ are shape functions; and $\hat{\mathbf{v}}_{1i}$ and $\hat{\mathbf{v}}_{2i}$ are unit vectors which are tangent to the midsurface and define the directions of rotations θ_i^1 and θ_i^2 . By differentiating Eqs.(5) with respect to the laminate coordinates, strains can then be obtained in terms of nodal displacements $\{q(t)\}$ as

$$\{\bar{\epsilon}(\bar{\mathbf{x}}, t)\} = [B(\bar{\mathbf{x}})] \{q(t)\} \quad (6)$$

Applications of Supercomputers in Engineering 501

where $[B(\bar{\mathbf{x}})]$ is the element strain-displacement matrix and $\{q(t)\}$ is given by

$$\{q(t)\} = [u_1, v_1, w_1, \theta_1^1, \theta_1^2, \dots, u_i, v_i, w_i, \theta_i^1, \theta_i^2, \dots]^T \quad (7)$$

By using a variational formulation and the above expressions, the following finite element equilibrium equations are obtained for each element

$$M_{mn}^e(\bar{\mathbf{x}})q_n(t) + \int_{\tau=-\infty}^{\tau=t} k_{mn}^e(\zeta_r - \zeta_r') \frac{\partial q_n(\tau)}{\partial \tau} d\tau = f_m^e(t) \quad (8)$$

$(m, n = 1, 2, \dots, \mathfrak{R}^e)$

where $f_m^e(t)$ are element residual nodal forces and $\mathfrak{R}^e$ is the number of nodal degrees of freedom. Element mass matrices, time dependent element stiffness matrices and residual nodal force vectors can be expressed by

$$M_{mn}^e(\bar{\mathbf{x}}) = \int_V \gamma \bar{N}_m(\bar{\mathbf{x}}) \bar{N}_n(\bar{\mathbf{x}}) dV$$

$$k_{mn}^e(\zeta_r - \zeta_r') = \int_V B_{im} \bar{D}_{ij}(\bar{\mathbf{x}}, \zeta_r - \zeta_r') B_{jn} dV \quad (9)$$

$$f_m^e(t) = \int_V N_m(\bar{\mathbf{x}}) \bar{\Omega}_i dV$$

where γ is the mass density and $\bar{\Omega}_i$ are surface tractions. In the discretized time domain, Eqs.(8) can be reformulated by using Newmark's average acceleration method and Prony series representations for relaxation moduli, resulting in

$$\begin{aligned} & \sum_{r=1}^R \left\{ \left(\frac{2}{\Delta t} \right)^2 \cdot M_{mn} + K_{mnr} + \sum_{\rho=1}^{N_r} K_{mnr\rho} \cdot [S_{r\rho}^1(t_p) \right. \\ & \quad \cdot \Delta t - S_{r\rho}^2(\Delta t_p)] \frac{2}{\Delta t^2} \left. \right\} \cdot U_n(t_p) = F_m(t_p) \\ & + M_{mn} \left\{ \left(\frac{2}{\Delta t} \right)^2 [U_n(t_{p-1}) + \dot{U}_n(t_{p-1}) \cdot \Delta t] \right. \\ & \quad \left. + \ddot{U}_n(t_{p-1}) \right\} + \sum_{r=1}^R \sum_{\rho=1}^{N_r} \left[K_{mnr\rho} \left\{ S_{r\rho}^1(t_{p-1}) \right. \right. \end{aligned}$$

502 Applications of Supercomputers in Engineering

$$\begin{aligned}
 & \cdot \dot{U}_n(t_{p-1}) - S_{r\rho}^2(\Delta t_p) \cdot \frac{2}{\Delta t^2} \cdot [U_n(t_{p-1}) \\
 & + \dot{U}_n(t_{p-1}) \cdot \Delta t] + S_{r\rho}^1(t_p) \cdot [U_n(t_{p-1}) \cdot \frac{2}{\Delta t} \\
 & + \dot{U}_n(t_{p-1})] - \exp[-\zeta_r(t_p)/\lambda_{r\rho}] \cdot U_n(t_0) \Big\} \\
 & - R_{mr\rho}^c(t_p) \Big]
 \end{aligned} \tag{10}$$

where

$$\begin{aligned}
 S_{r\rho}^1(t_j) &= \int^{t_j} \exp[-(\zeta_r - \zeta_r')/\lambda_{r\rho}] ds \\
 S_{r\rho}^2(\Delta t_j) &= \int_{t_{j-1}}^{t_j} \int^s S_{r\rho}^1(s') ds' ds \\
 \Delta \zeta_r(t_j) &= \zeta_r(t_j) - \zeta_r(t_{j-1}) \\
 R_{mr\rho}^c(t_0) &= 0
 \end{aligned} \tag{11}$$

$$\begin{aligned}
 R_{mr\rho}^c(t_p) &= \exp[-\Delta \zeta_r(t_p)/\lambda_{r\rho}] \cdot \left[K_{mnr\rho} \cdot \left\{ S_{r\rho}^1(t_{p-1}) \right. \right. \\
 & \cdot \dot{U}_n(t_{p-1}) - S_{r\rho}^1(t_{p-2}) \cdot \dot{U}_n(t_{p-2}) \\
 & \left. \left. - S_{r\rho}^2(\Delta t_{p-1}) \cdot \ddot{U}_n^{av}(t_{p-1}) \right\} + R_{mr\rho}^c(t_{p-1}) \right]
 \end{aligned}$$

These recursion formulas require only two previous time solutions in order to be marched forward to the next time evaluation. For the formulation details see Yi *et al.* [2] and Yi [3].

NUMERICAL RESULTS AND DISCUSSION

In Refs. 1-3, a number of numerical examples were presented in order to evaluate the accuracy of this FEM and to demonstrate its usefulness for and applicability to solutions of dynamic anisotropic shell and curved plate problems. In this study, the performance of the finite element code VISSHELL on CRAY supercomputers is examined. A clamped cylindrical roof shell (curved plate) under a uniformly distributed step load is considered. The geometry is shown in Fig. 1 with $r=64.52$ m, $L=12.90$ m,

$h=5.690$ cm, and $\beta = 11.46^\circ$. The time dependent transverse deflections for $[45/-45]_s$ angle ply laminates are calculated. Graphite/epoxy is used with elastic material properties of $E_{11} = 124.1$ GPa, $E_{22} = 96.53$ GPa, $G_{12} = G_{23} = G_{31} = 6.205$ GPa, $\nu_{12} = 0.34$, and $\gamma = 9134$ kg/m³. Time functions for the relaxation moduli and shift factors are given in Ref. 3. The shell is divided by using four noded or nine noded elements and the problem size is increased by increasing the number of edge divisions of the shell. Several examples with different degrees of freedom are computed on the CRAY-YMP machine. Computational times and vector floating point operations are noted. Flowtrace and Perftrace tools are used to report performance data for subroutine program modules and specified code segments. For four and nine node elements, Mflops (million floating point operations per second) and CPU times for the subroutine SHELLSTF, which calculates element stiffness matrices, are tabulated in Table 1. It should be noted that the theoretical maximum Mflops capability of the CRAY Y-MP is about 300 and therefore the nine node calculation of SHELLSTF reaches 61 % capacity. Total CPU times spent in SHELLSTF on the CRAY Y-MP for various problem sizes are plotted in Fig. 2. Since the vector lengths for vector floating point operations increase with increasing numbers of nodes per element, Mflops for nine noded elements is higher than one for four noded elements because of the larger number of operations. However, total CPU times for nine noded elements are greater than those for four noded elements. (See Table 2 for definitions of subroutine functions.)

Performances of the Cray sparse matrix solvers and the Feable solvers are discussed next. SSPOTR and SSPOTRS are subroutines developed by CRAY Inc. to factorize and to directly solve real sparse symmetric definite systems respectively, while the subroutines FACTRZ and SIMULQ (Feable subroutines) are MIT subroutines developed in the early 70's. In the subroutine FACTRZ the Cholesky decomposition method is used in order to factorize matrices. The variable bandwidth storage method is used for the Feable solvers to reduce storages of stiffness matrices. However, there may be zeros within the envelope of a band matrix. For the Cray sparse matrix solver, only nonzero entries in the stiffness matrix are required to be stored by using pointer vectors. The population densities in global

stiffness matrices are listed in Table 3. Suppose that an n -by- n global stiffness matrix has m nonzero entries, then for the Cray sparse matrix solver, a total of $2m + n + 1$ words of storage is required including pointers. The memory storage required for stiffness matrices and row and column indices is plotted in Fig. 3 for four and nine noded elements versus various problem sizes.

The results show that by using the Cray sparse matrix solver Mflops increase with increasing active degrees of freedom. Also as shown in Fig. 4, CPU times are not dramatically increased by augmenting active degrees of freedom. However, different from other wavefront solvers, the Feable solvers do not increase vector lengths for vector floating point operations and Mflops numbers. Consequently CPU times increase significantly with increasing problem sizes. CPU times and Mflops for the Cray and Feable solvers are illustrated in Figs. 4 and 5 respectively. The sparse matrix software for the Cray systems reduces memory storage and total arithmetic operations required by keeping track of only nonzero entries in the matrices. However, logical if statements must be used in order to find the nonzero values for stiffness matrices. Such logical scalar operations require much CPU time, which increase significantly with increasing matrix size. In Fig. 6, overhead CPU times needed to check and store only the nonzero values for stiffness matrices are calculated for various problem sizes by using four or nine node elements. A comparison of CPU times spent to solve these problems by using nine node elements on the different machines (CRAY Y-MP and CRAY Y-MP C-90) are presented in Fig. 7. As expected, CPU times for the C-90 model are significantly decreased by approximately a 2:1 factor.

CONCLUSIONS

The subroutine SHELLSTF is dominated by vector operations and achieves about 121 to 183 million floating-point operations per second during its executions for four noded and nine noded elements respectively. Computational intensity studies (computational intensity is the ratio of flops to memory references) also indicate that it is an efficient algorithm. Since the vector lengths for vector floating point operations increase with increasing numbers of nodes per element, Mflops for nine noded elements

are higher than those for four noded elements. However, total CPU times for nine noded elements are greater than those for four noded elements. The results show that when using the Cray sparse matrix solver Mflops increase with increasing active degrees of freedom, but CPU times are not dramatically increased by augmenting the number of active degrees of freedom. However, overhead CPU times to check and store only the non-zero values for stiffness matrices are significant for the Cray sparse matrix solver. The Feable subroutines do not increase vector lengths for vector floating point operations and Mflops numbers.

REFERENCES

- [1] Hilton, H. H. and Yi, S. 'Anisotropic Viscoelastic Finite Element Analysis of Mechanically and Hygrothermally Loaded Composites', *Composites Engineering*, Vol. 3, No. 2, pp. 123-135, 1993.
- [2] Yi, S., Pollock, G. D., Ahmad, M. F., and Hilton, H. H. 'Time Dependent Analysis of Viscoelastic Composite Shell Structures', *Computing Systems in Engineering*, Vol. 3, pp.457-467, 1992.
- [3] Yi, S. *Finite Element Analysis of Anisotropic Viscoelastic Composite Structures and Analytical Determination of Optimum Viscoelastic Material Properties*, Ph.D. dissertation, Aeronautical and Astronautical Engineering Department, University of Illinois at Urbana-Champaign, 1992.
- [4] Liu, J. W. H. 'The Multifrontal Solution of Indefinite Sparse Matrix Solution: Theory and Practice', *SIAM Review*, Vol. 34, pp. 82-109, 1992.
- [5] Irons, B. M. 'A Frontal Solution Program for Finite Element Analysis', *International Journal for Numerical Methods in Engineering*, Vol. 10, pp. 379-399, 1976.
- [6] Hood, P. 'Frontal Solution Program for Unsymmetric Matrices', *International Journal for Numerical Methods in Engineering*, Vol. 2, pp. 5-32, 1970.
- [7] Heath, M. T. (Ed). *Sparse Matrix Software Catalog*, Mathematics and Statistics Research Department, Oak Ridge National Laboratory, Oak Ridge, Tennessee, 1982.
- [8] Hilton, H. H. 'Viscoelastic Analysis', *Engineering Design for Plastics*, Baer, E. (Ed), Reinhold Pub. Corp., New York, pp. 199-276, 1964.



Table 1 Mflops and CPU times for the element stiffness matrix computation.

node	Mflops	CPU Time/Element (secs)
4	121	0.45×10^{-3}
9	183	0.3×10^{-2}

Table 2 Subroutines and their functions.

Program Name	Subroutine Name	Function
CRAY SPARSE SOLVER	SSPOTR	factorization
	SSPORTS	solution
VISSHELL	SHELLSTF	stiffness computation
	FACTRZ	factorization
	SIMULQ	solution

Table 3 Population densities of non-zero values in global stiffness matrices.

Storage Method	81 nodes	625 nodes	961 nodes	1225 nodes
variable bandwidth	2.34×10^{-1}	8.18×10^{-2}	6.57×10^{-2}	5.81×10^{-2}
non-zero entries	5.74×10^{-2}	7.41×10^{-3}	4.8×10^{-3}	3.77×10^{-3}

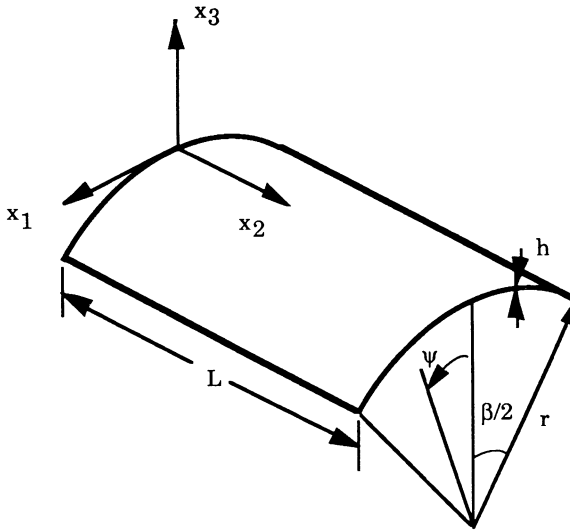


Fig. 1 Cylindrical roof shell.

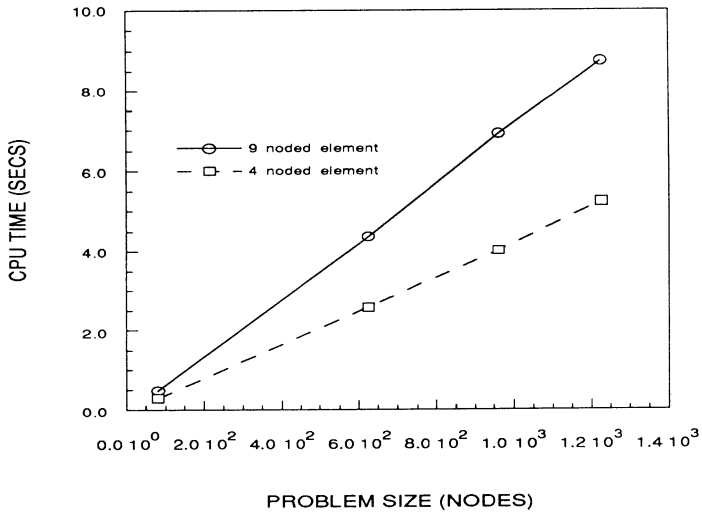


Fig. 2 CPU time spent in the subroutine SHELLSTF.



508 Applications of Supercomputers in Engineering

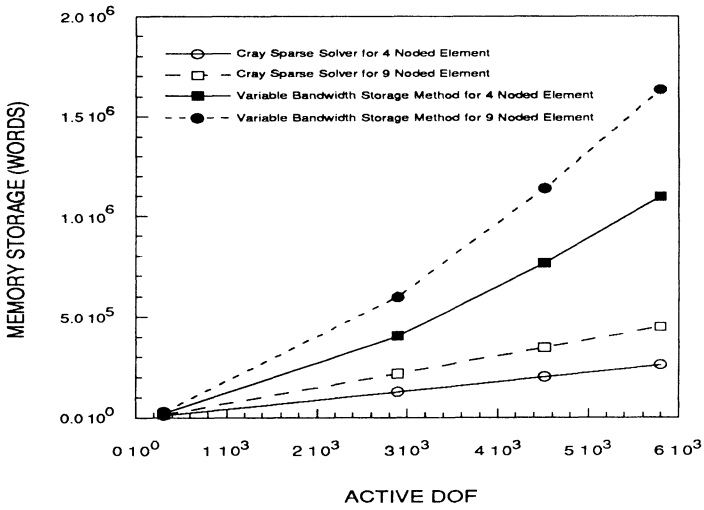


Fig. 3 Memory storage for stiffness matrices and pointers.

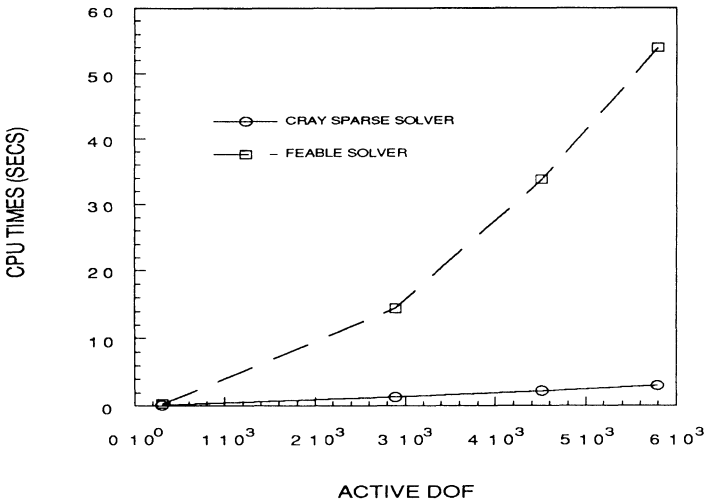


Fig. 4 CPU time for the Cray and Feable solvers.



Applications of Supercomputers in Engineering 509

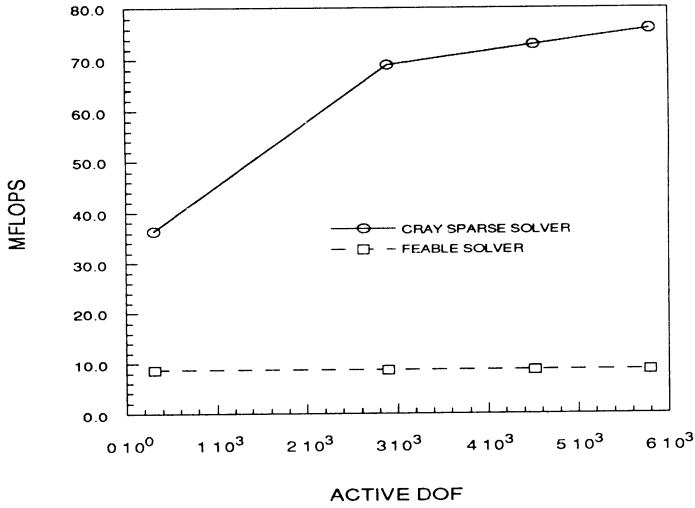


Fig. 5 Mflops for the Cray and Feable solvers.

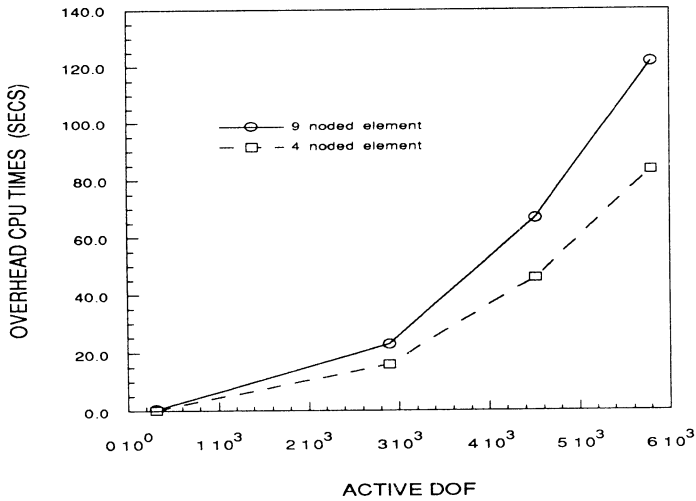


Fig. 6 Overhead CPU time for the Cray sparse solvers.



510 Applications of Supercomputers in Engineering

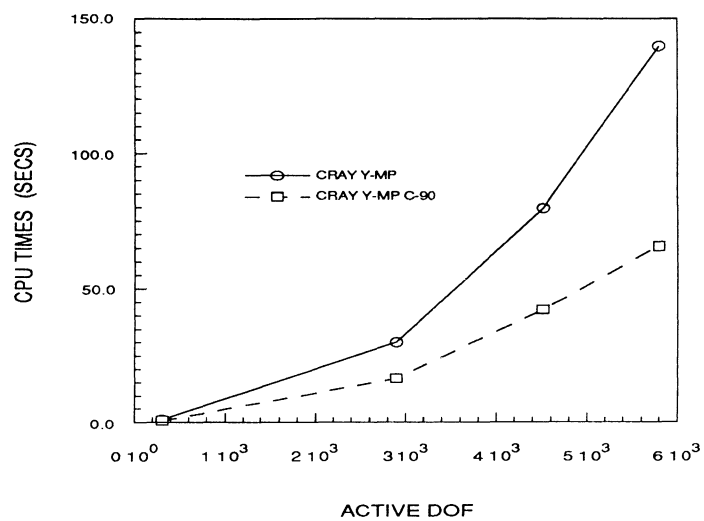


Fig. 7 Total CPU time for nine noded elements using CRAY sparse solvers.