# PERFORMANCE OF NASA EQUATION SOLVERS ON COMPUTATIONAL MECHANICS APPLICATIONS

Olaf O. Storaasli*
Computational Structures Branch
NASA Langley Research Center
Hampton VA

## Table of Contents

This paper describes the performance of a new family of NASA-developed equation solvers used for large-scale (i.e. 551,705 equations) structural analysis. To minimize computer time and memory, the solvers are divided by application and matrix characteristics (sparse/dense, real/complex, symmetric/ nonsymmetric, size: in-core/out of core) and exploit the hardware features of current and future computers. In this paper, the equation solvers, which are written in FORTRAN, and are therefore easily transportable, are shown to be faster than specialized computer library routines utilizing assembly code. Twenty NASA structural benchmark models with NASA solver timings reside on World Wide Web with a challenge to beat them.

# Introduction

The goal of this paper is to describe the performance of new NASA-developed general-purpose algorithms to solve systems of equations for static and vibration analyses of structures. The new equation solvers run at world-record speeds for structural analysis based on direct comparisons with other well-known equation solvers. They have been implemented on high-performance supercomputers (Cray C-90), workstations (IBM RS 6000, Sun) and a Pentium-Pro desktop system.

Previous NASA research [Refs. 1-4] described the fastest structural algorithms and equation solvers known at that time. These iterative, skyline and variable band linear equation solvers were written to exploit the full capabilities of parallel and vector supercomputers. Subsequent NASA equation solver development on parallel-vector computers (i.e. Intel Paragon and IBM SP-2) uncovered two major drawbacks that impede their use for structural analysis:

1. Significant interprocessor communication time that dominates as the number of processors increases.
2. Limited size of structural applications (number of equations) that can fit in processor memory.

As the size and complexity of structural models to be analyzed has increased beyond 100,000 equations, band and wavefront solution methods have been found to be too slow, and iterative methods often do not converge or are slow. Thus, a new family of Vector-Sparse matrix equation Solvers (VSS) was developed to solve structural analysis applications with increased speed and accuracy. VSS, is written entirely in FORTRAN and operates on supercomputers, low-cost workstations and a desktop Pentium-Pro system. This paper includes the following new information:

1. The methodology used in VSS is described and compared with methodology used in conventional solution procedures. The description and comparisons make it clear why VSS solves structural analysis applications faster than traditional iterative, skyline, and variable-band solution methods.

2. The full range of capabilities and limitations of VSS are described and compared to those of other solvers. These capabilities include the ability to solve complex, indefinite and even unsymmetric systems of equations, where many solvers fall short. Such a capability can reduce (by a factors of 10 or more) the computation time for electromagnetic and acoustic analysis compared to traditional iterative methods.

3. "Large" applications (exceeding 250,000 equations) which once required a supercomputer with specialized (costly) infrastructure are solved by VSS routinely on an engineering workstation (or a Pentium-Pro desktop system configured with sufficient memory).

4. VSS has an Xwindows real-time graphical interface which lets the user "watch" the equation solution process. Plots of the original matrix, matrix after reordering, reduced matrix, matrix after fill and during the solution process (as factoring proceeds down the diagonal) give the user added insight into the equation solution process. This is important because equation solution takes the majority of computation time for most engineering and scientific applications.

The paper addresses the four above items in detail with examples and results. First, VSS is described in terms of its methodology. Then the performance of VSS is evaluated using several structural analysis examples. Finally, VSS applications using more general matrices (complex, nonsymmetric, and indefinite) are discussed.

# Description of VSS, Vector Sparse Solver

VSS was originally developed for the rapid solution of positive-definite structural analysis. However, soon a family of VSS solvers emerged, with versions tailored for different applications (matrix characteristics). Most sparse solvers, including the VSS family, contain three major components:

1. Matrix reordering
2. Matrix factoring
3. Forward/Backsolve

The speed of VSS results from techniques used to minimize the computation time (i.e. the number and types of computations) while maintaining accuracy. In particular, VSS reduces significantly the number of terms (and operations), required for general matrix solution

Matrix factoring dominates the computation time for small and medium applications when typical matrix reordering schemes (i.e. multiple-minimum degree) are used, while the forward/backsolve time is negligible. However, for larger applications (i.e., 250,000+ equations), the time for matrix ordering using the multiple-minimum degree method grows to equal or even exceed the factor time. Thus, a compressed multiple-minimum degree ordering algorithm was developed which uses a reduced matrix (i.e., a fraction of the full matrix size). This reduced reordering (and the related computations) take about 20 percent of the operations (time) otherwise required to solve the original matrix. No other matrix reordering algorithm has been found to be as fast. If a user insists on using a matrix previously reordered by a banded method, a VSS option compensates for such unnecessary reordering.

In matrix factoring, great pains were taken to avoid the costly overhead of indirect addressing. This was accomplished by referencing the equations in separate blocks, referred to as "blocking" the equations. Since the matrix factor time varies if the application contains solid (three dimensional) elements, the user may select one of three cases: mostly solid elements, few solid elements or a mixture of solid and other elements. This minimizes the solution time for all cases.

The time-consuming Choleski $LL^T$ factor method (which suffers from the expensive square root computations) was replaced by the more efficient (and more general) $LDL^T$ factor method. The so-called òloop unrollingó software technique (using levels 1, 3, 5 and 6) was found to significantly reduce the matrix factor time. Similar blocking and vectorization was used during the backsolve computations. VSS attempts to minimize "fill in" during matrix factorization. The following five equation solution steps may be viewed by the user via real time (X-window) matrix displays:

1. original matrix
2. reduced matrix
3. reordered matrix
4. matrix fill
5. "live" matrix factoring along the diagonal

VSS has options for complex, non-symmetric, out-of-core and indefinite matrix solution (i.e. for electromagnetic and interfaced structural analysis). The òout-of-coreó option permits the solution of even the largest applications on a workstation or desktop, but imposes a performance penalty due to increased disk I/O. The out-of-core version was found most attractive on the IBM RS 6000 and Intel Pentium-Pro computers. In addition to structural analysis, VSS reduces the analysis time for electromagnetic applications (discussed in a subsequent section)

To determine the accuracy of VSS displacement solutions, both an absolute and relative error norm are computed, and the largest displacement and sum of all displacements calculated for comparisons. VSS is written in FORTRAN and is therefore transportable to a wide variety of computers.

## VSS Performance

The performance of VSS was evaluated on several computers for 20 structural applications. Five of these applications are presented in this paper. These include adaptively-refined models of a High-Speed Civil Transport (HSCT), a wind tunnel, and an automobile. The number of equations for each application is:

16,152 equation Mach 2.4 HSCT (Fig 1)

88,404 equation Mach 3.0 HSCT (Fig 3)

217,918 equation wind tunnel model (Fig. 5)

263,574 equation automobile model (Fig. 6)

551,705 equation Mach 2.4 HSCT (refined Fig. 1)

The finite-element models (matrices and loads) were generated using the NASA COmputational MEchanics Testbed (COMET) software. The solutions were obtained using VSS and compared with solutions obtained for identical models using library solvers and commercial codes (see Figs 1-8). The factor time (including reordering) is reported because it represents approximately 99 percent of the total solution time.

# 16,152 Equation Mach 2.4 HSCT

A Mach 2.4 , 16,152 equation HSCT model (see Fig.1), was symmetrically loaded (upward) at both wingtips and fixed at its nose and tail.

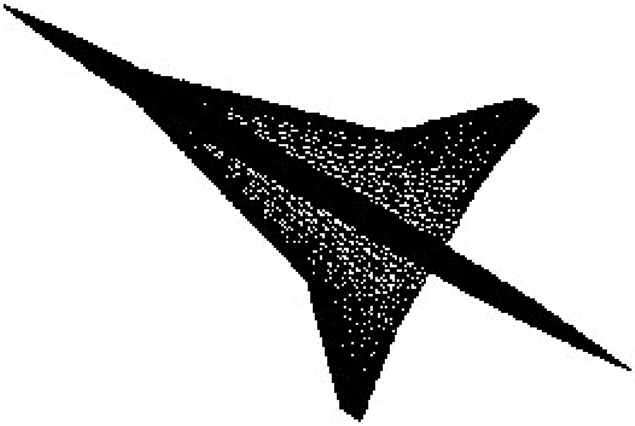
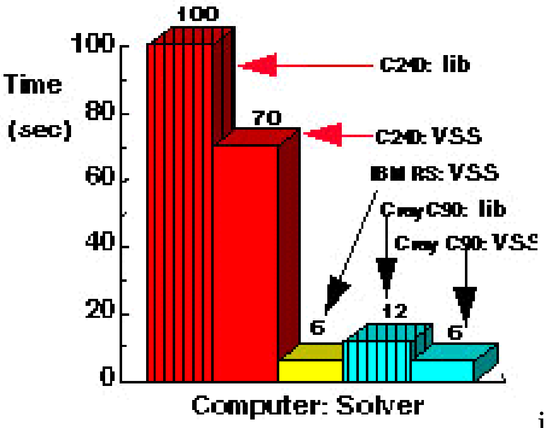


Fig. 1. Mach 2.4 HSCT Structural Model

The model contained 2,694 nodes and 7,868 triangular elements which produced a matrix with 12.5 million terms with a maximum and average bandwidth of 697 and 449 terms, respectively. Static displacements were computed on Convex C-240, Intel Pentium Pro, IBM RS 6000 and Cray C-90 computers using VSS and library solvers (denoted lib in Fig. 2.) for symmetric wingtip loads



i

Fig. 2. Mach 2.4 HSCT Solution Time (sec)

As the library equation solvers are tailored to specific computers, they often outperform user-developed equation solvers. However, as shown in Fig. 2, VSS clearly outperforms system library equation solvers for the Convex C240 and the Cray C-90.

The performance variation of the three computers is also evident in Fig. 2. The IBM RS 6000 and Cray C-90 clearly outperform the Convex C-240 whether VSS or the specialized system library equation solvers are used. For this structural model, the performance of the IBM RS 6000 matches the performance of the Cray C-90 using VSS. The VSS solution time on the IBM RS-6000 was twice as fast as the solution time on the Cray C-90 using the Cray library equation solver. The Pentium-Pro (P6) system took 16 seconds to obtain the solution. All computations for this model were accomplished in computer memory (no out-of-core computations).

# 88,404 Equation Mach 3 HSCT

A second, larger, more complex finite-element model of a Mach 3.0 HSCT (see Fig. 3) was next used to evaluate the VSS equation solver.

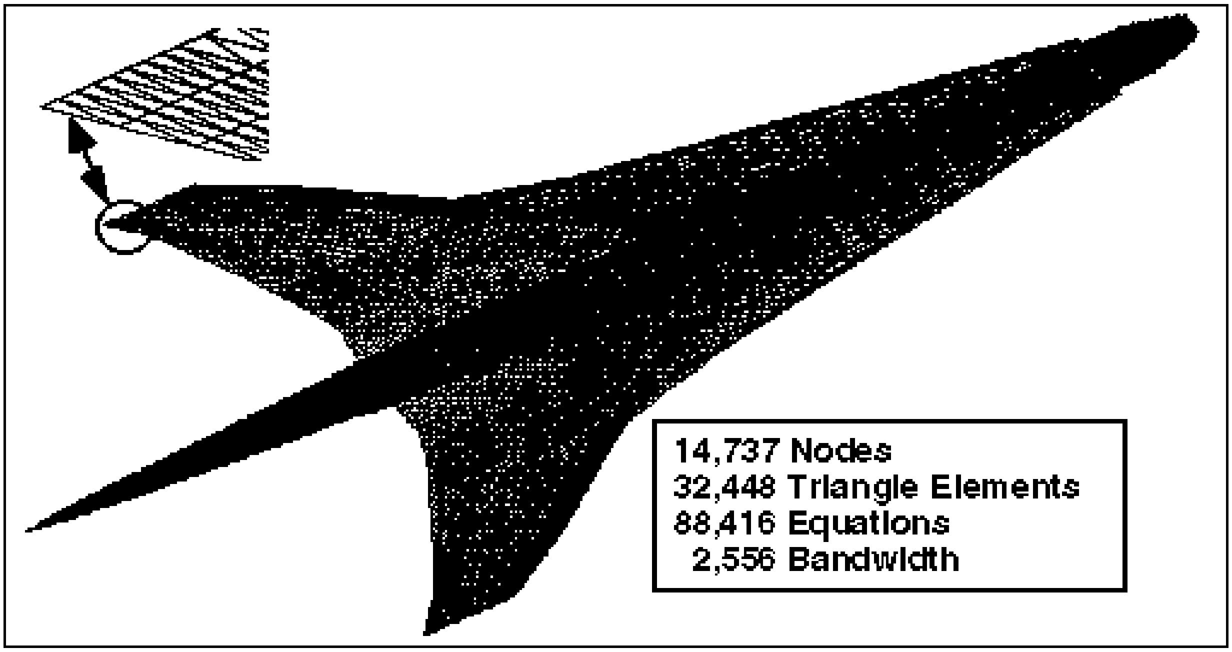


Fig 3. Mach 3.0 HSCT structural model

Symmetric wingtip loads were applied and a static structural analysis performed using the equation solver just as in the Mach 2.4 HSCT analysis. This Mach 3.0 model consists of 14,737 nodes and 32,448 triangular elements as shown in Fig. 3. The assembled global stiffness matrix has 88,416 unconstrained (88,404 constrained) equations with a maximum and average bandwidth of 2,557 and 1095 terms, respectively

Static displacements for this HSCT structural model were obtained on the Convex C-240, Convex C-3820, IBM RS 6000 and Cray C-90 (see Fig. 4) for VSS and system library equation solvers.

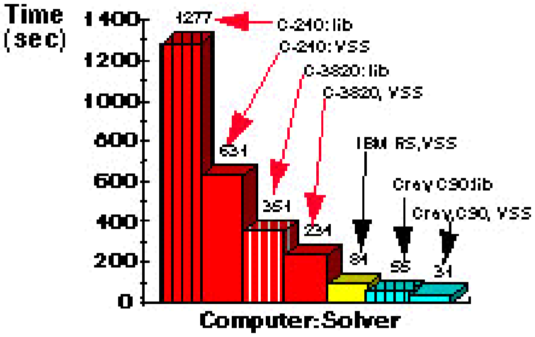


Fig. 4. Mach 3 HSCT Factor Time (sec)

The Intel Pentium-Pro system used had 64 MB memory which was too small to solve this model with either the in-core or out-of-core version of VSS. VSS computes static displacements much faster than the system library equation solvers on the Convex C-240, C3820 and Cray C-90. VSS was approximately 2.7 times faster on the Cray C-90 than on the IBM RS 6000.

## 217,918 Equation Wind Tunnel

The 217,918 equation structural model of a NASA Ames wind tunnel, shown in Fig. 5, has a 12 foot diameter test section and cross-sections whose inner and outer diameters vary from 12 to 68 feet. The operating pressure is 73.5 PSI and the tunnel was hydo-tested to 101 PSI using ASME Section-9, Division 2 design code. Stiffened columns and stiffener rings are located along the wind flow direction. The water weight for the hydro-test was 55 million pounds. The tunnel was designed to withstand earthquake loads and loss of half of its blades to impart an unbalanced load of 300,000 kips on the tunnel at operating speed. The tunnel contains large access penetrations concentrated in the sphere settling chamber and the test section.

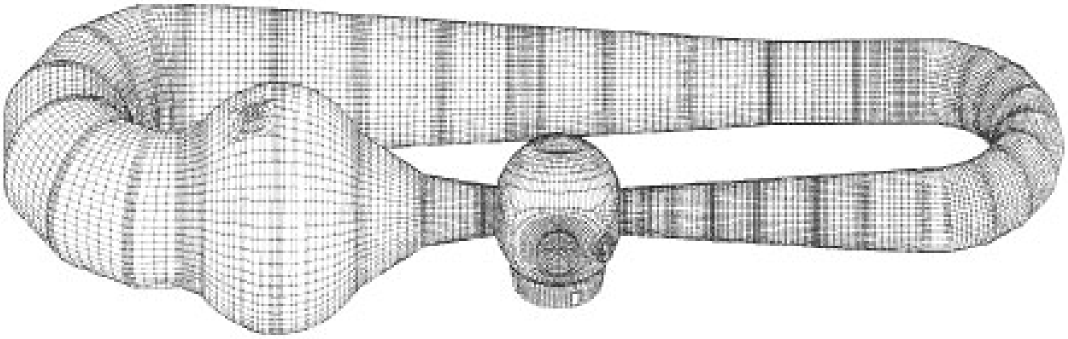


Fig. 5. NASA Ames Wind Tunnel Model

Static displacements, resulting from the 101 PSI internal pressure loading, were computed using MSC/ NASTRAN (has only an out-of-core solver) and both in-core and out-of-core versions of VSS. The equation solution times were compared on a Cray C-90 for two versions of VSS (òin-coreó and òout-of-coreó) and MSC/NASTRAN on the Cray C-90. The solution time was 5.5, 3.4 and 1.7 minutes for out-of-core VSS, NASTRAN (sparse solver), and VSS (in-core), respectively. Two additional comparisons were made: the NASTRAN sparse solver took 2.3 minutes on the Cray T-90 compared to 1.7 minutes for incore VSS on the Cray C-90. This indicates great potential for an optimized VSS in finite element codes. The time to solve the same wind tunnel application on the IBM RS 6000 was 14.6 minutes, showing that such large-scale structural applications no longer require an expensive supercomputer.

## 263,574 Equation Automobile

A 263,574 equation structural model of a Ford Thunderbird automobile is shown in Fig. 6.

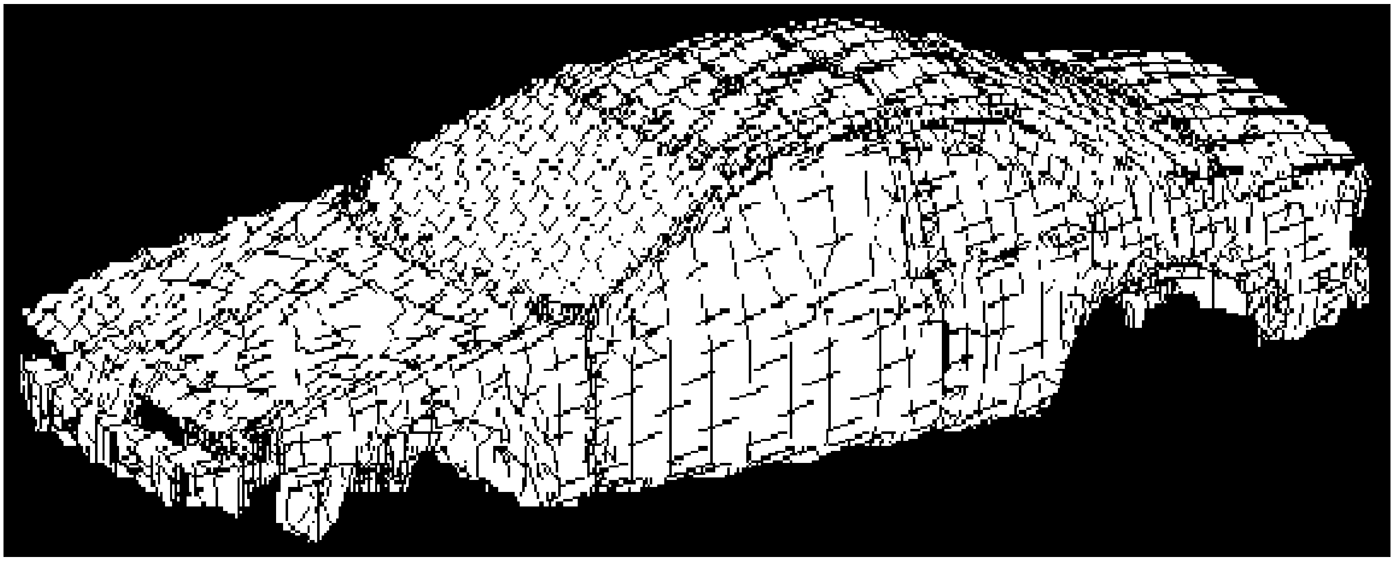


Fig. 6. Ford Thunderbird Model

This model contains 44,188 nodes, 48,894 shell elements and its global stiffness matrix has a average bandwidth of 3,374. Static displacements were calculated for a front impact load for this structural model using VSS on the IBM RS 6000 and Cray C-90. The solution times are shown in Fig 7 for both VSS and system library equation solvers.

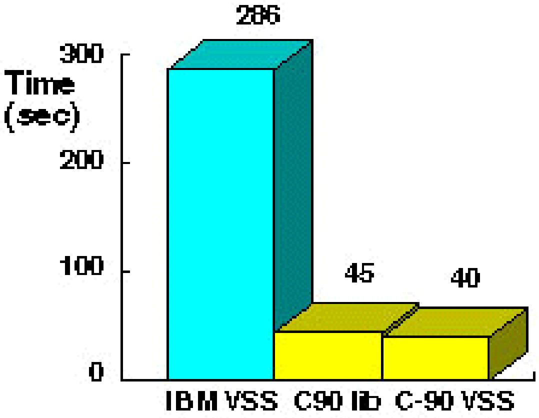


Fig. 7. Automobile Factor Time (sec)

VSS is 7.1 times faster on the Cray C-90 than on the IBM SP-2 and slightly faster than the system library equation solver on the Cray C-90.

## 551,705 Equation Mach 2.4 HSCT

The largest HSCT model attempted to date contained 551,705 degrees of freedom, 121,524 nodes and 39,942 elements. Such applications require a sparse solver as even the best banded solver, running at nearly ideal MFLOP rates, would, at a minimum, be 100 times slower (5 vs 500) as illustrated at the left of Fig. 8.

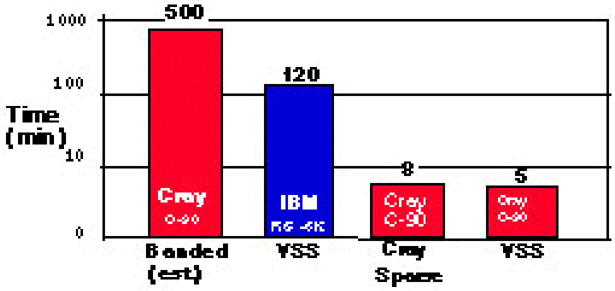


Fig. 8. Large HSCT Factor Time (min.)

With the rapid VSS reordering, most of the solution time is associated with the matrix factorization for all structural models evaluated.

# VSS Extensions and Applications

The NASA solvers are being applied and extended in a natural way to the solution of eigenvalues and applications involving indefinite, complex and unsymmetric matrices (i.e. electromagnetics). A brief description of the extension and application follows. To avoid confusion and maintain a high level of performance for each application, a separate solver version is maintained for each application.

## Symmetric Lanczos Eigensolver

VSS was used in a Lanczos eigensolver and found to reduce the eigenvalue computation time in direct proportion to the speedup of the linear solver. Since most (typically 90+%) of Lanczos eigenvalue computations are associated with the linear solver run repetitively, the overall reductions in eigenvalue analysis time were dramatic (directly proportional to the reduction in linear solve time). Since most Lanczos eigensolvers have the linear solver clearly identified as a subroutine call, replacement with the faster VSS solver was accomplished with minimal effort. In addition to replacing the linear solver, a fast matrix-vector multiplication method was also incorporated in the Lanczos procedure [Ref. 4].

## Indefinite Solver

Although structural matrices generated by finite element analysis are positive definite, some applications, such as interfacing regions where Lagrange multipliers are imposed or combining finite element, boundary element and other regions into a single indefinite matrix, an indefinite solver is required. Since most of the structural matrix is sparse, for performance reasons, it is desirable to use VSS. However, additional code was inserted to detect any zeros on the diagonal and move them down to the lower right via reordering, where pivoting is limited only to that section. The indefinite solver takes nearly the same time as VSS to solve applications where the number of indefinite terms is small (all structural applications attempted). The results for the indefinite matrix solution agreed with those obtained for the same application using the Cray-Boeing indefinite solver (uses Cray assembly code, limited to Cray computers). VSS, written entirely in FORTRAN, took about 20% less time to obtain the solutions when compared to the Cray-Boeing indefinite solver.

## Complex Unsymmetric (Electromagnetics)

VSS was extended to solve complex matrices by changing the data type from REAL to COMPLEX and modifying the error norm to measure both the real and imaginary part of the complex solution. This change allows the solution of a large subset of symmetric electromagnetic problems that formerly used iterative solvers taking 3-15 times longer. For low-observable electromagnetic applications, the majority of the matrix is symmetric (non-metal reflections), with a smaller non-symmetric matrix (metallic reflections). VSS was extended to solve such electromagnetic applications with less than a 20% performance penalty when compared to complex symmetric VSS solutions.

# Conclusions and Recommendations

A family of highly-efficient sparse equation solvers and analysis algorithms have recently been developed by NASA Langley. All these modular subroutines have been tested on a variety of engineering applications on supercomputers, workstations and on a desktop Pentium-Pro system. The sparse equation solver appears to solve structural analysis applications faster than any known solver. It permits the solution of nearly all complex engineering applications on workstations with sufficient memory.

# Software and Data Availability

The NASA software operates in either a stand-alone mode, or integrated into existing finite element codes. To obtain copies of NASA equation solvers, structural analysis software or finite element benchmark data, send a formal request to the author (i.e. on Letterhead stationery by letter or FAX) specifying the target computer(s), operating system(s) and ftp address for receipt of the software. Once approved, the software, documentation files and sample cases will be sent electronically. A sample request letter and performance comparisons for twenty NASA benchmarks are maintained on WorldWide Web: olaf.larc.nasa.gov, by the author (O.O.Storaasli@larc.nasa.gov).

# Acknowledgments

The author is grateful to Dr. Majdi Baddourah, (currently with Cray Research, Inc.) for programming assistance and to the Northop-Grumman B-2 Division. for electromagnetic matrix data. The author is also grateful Intel for use of one of the first Pentium-Pro (P6) pre-production system as part of the Intel P6 User Test Program.

# References

1.Storaasli, O. O., Bostic, S. W., Patrick, M., Mahajan, U., and Ma, S., "Three Parallel Computation Methods for Structural Vibration Analysis", AIAA Paper No. 88-2391 *Proc. of the 29th AIAA/ASME ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*,Williamsburg, VA, April 18-20, 1988, pp. 1401-1411.

2*. Storaasli, O. O., Nguyen, D. T. and Agarwal, T.K., "A Parallel-Vector Algorithm for Rapid Structural Analysis on High-Performance Computers", *NASA TM 102614*, April 1990.

3. Baddourah, Majdi A. , Storaasli, O. O., Carmona, Edward A., and Nguyen, D. T., "A Fast Parallel Algorithm for Generation and Assembly of Finite Element Stiffness and Mass Matricesó, *Proceedings of the 32nd AIAA/ASMEASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference,* Baltimore, MD April 8-10, 1991

4*. Storaasli, O. O., Nguyen, D. T., Baddourah, M. A. and Qin, J., "Computational mechanics analysis tools for parallel-vector super-computers", Computing Systems In Engineering, Volume 4, Numbers 4-6, pp 349-354, 1993. (Also AIAA Paper No. 1387, AIAA 34th Structures, Structural Dynamics and Materials Conference, La Jolla, CA, April 19-22, 1993)

* On Langley Technical Report Server on World Wide Web.