An Evaluation on the Efficient use of Supercomputers for Computational Aeroelasticity

G.P. Guruswamy[1]
NASA Advanced Supercomputing Division
Ames Research Center Moffett Field California, USA 94035
Guru.p.guruswamy@nasa.gov

ABSTRACT

This study evaluates the role of supercomputers in large-scale fluid/structure analyses of aerospace vehicles, with an emphasis on computational aeroelasticity dominated by complex fluid/structure interactions. The information presented is primarily based on responses to a web-based survey that was designed to capture the nuances of high-performance computational aeroelasticity and, in particular, its use of parallel computers. Survey responses were solicited from leading application engineers who use NASA and DoD supercomputing resources. A method of assigning a fidelity-complexity index (FCI) to each application is presented, and case studies of major applications using HPC resources are summarized.

1. INTRODUCTION

Aeroelasticity analysis involving strongly coupled fluids, structures and controls is an important element in aircraft design. Computational aeroelasticity based on low-fidelity methods, such as linear aerodynamic flow equations coupled with modal structural equations, is well established. Although these low-fidelity approaches are computationally less intensive, they are not adequate for the analysis of aircraft that can experience complex flow-structure interactions. Examples of such complex interactions include the vortex-induced aeroelastic oscillations of the B-1 aircraft [1], structural oscillations of the F-18A's vertical tails due to the burst of leading edge vortices [2], buffet-associated structural oscillations [3] and dips in flutter speed [4] experienced by aircraft that fly in the transonic regime, and abrupt wing stall experienced by modern fighters such as the F/A 18E aircraft that can be dominated by unsteady flows possibly associated with aeroelastic oscillations [5]. High-fidelity equations, such as the Euler/Navier-Stokes (ENS) for fluids and the finite-element method (FEM) for structures, are needed to produce accurate aeroelasticity computations for situations involving these complex fluid/structure interactions. Using high-fidelity methods, design quantities such as structural stresses can be directly computed.

Aeroelastic computations are typically orders of magnitude more expensive than steady calculations for rigid configurations due to the multidisciplinary complexities of the physics involved. Figure 1 shows a typical increase in the processor time required to compute a transonic flutter boundary for increasing geometric complexities (based on 5 modes, 5 frequencies, and 20 Mach numbers, with 10,000 timesteps per case). All computer time requirements are presented in terms of SGI Origin2000 single-processor hours. The growth in required CPU time is exponential. Hundreds of such computations are required for a complete aircraft design.

Continuous growth in computing resources is required to advance increasingly challenging new technologies in the aerospace industry. To sustain current leadership in aerospace technology, it is crucial for the US to also maintain its leadership in supercomputing. To help accomplish this, congress authorized the High Performance Computing and Communication (HPCC) program when it passed the High Performance Computing Act of 1991. Created as a foundation for the country's evolving R&D needs, this dynamic program provides the focus needed for continuing development of new computer technologies and applications. As leading participants in this program, the Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP) and the High Performance Computing and Communication (HPCC) project at NASA have promoted several state-of-the-art

[1]Senior Scientist

supercomputing platforms and advanced the development of parallel architectures. Users of those resources have begun to compute aeroelasticity associated with complex flow separation. Figure 2 shows the increase in GFLOPS rate (billion floating-point operations per second) achieved by production-type aeroelastic codes based on the Navier-Stokes equations coupled with the modal structural equations. Reference 6 shows vortex burst-induced structural response results for the F-18/A vertical tail, using a moderate grid that required approximately 10,000 node-hours for one set of flow parameters. However, unsteady aerodynamic and aeroelastic simulations of the F-18 E/F at abrupt wing stall, which is yet to be computed, will require millions of node-hours [7].

This report presents the status and impact of utilizing HPC resources for high-fidelity-based aeroelastic analysis including unsteady aerodynamics.

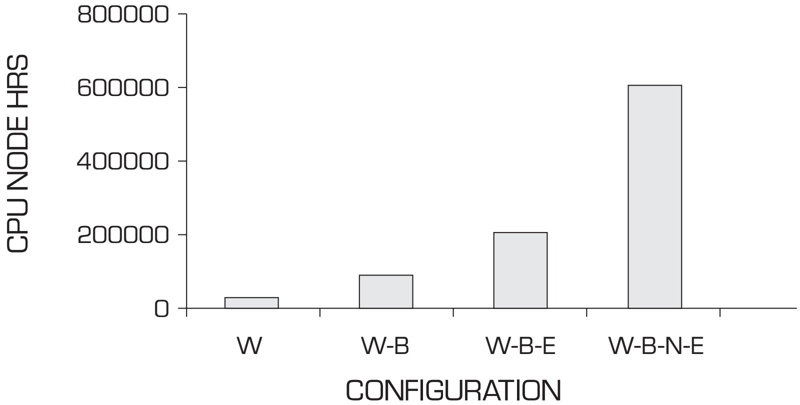

Figure 1. Computer time in SGI Origin2000 node-hours needed for a single design point transonic flutter boundary computation using coupled Navier-Stokes and modal equations. (W: Wing, B: Body, E: Empennage, N: Nacelle)

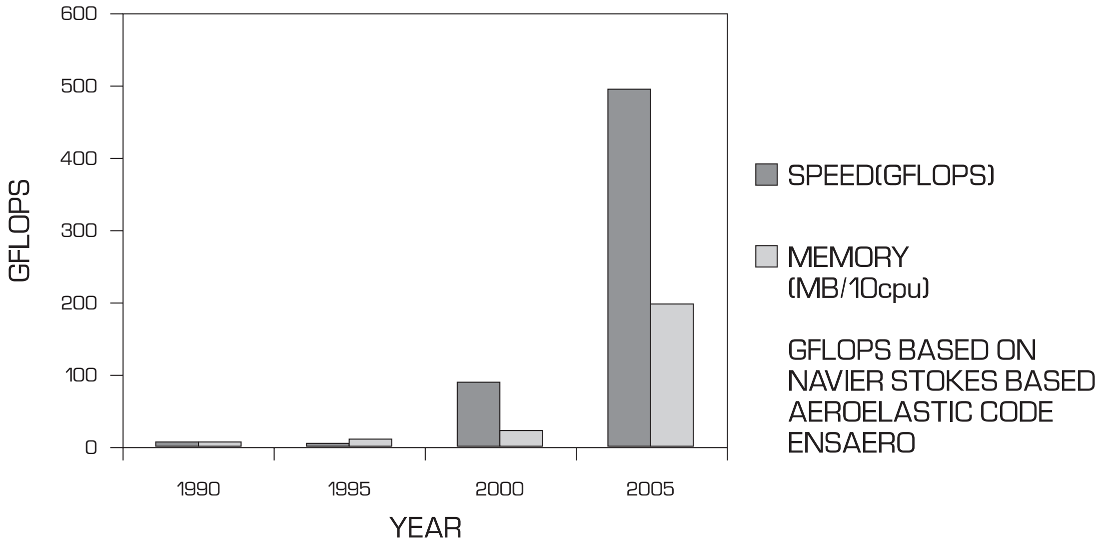

Figure 2. Growth in supercomputing power

2. COUPLED FLUID/STRUCTURE ANALYSIS

In recent years, significant single-discipline advances have been made individually in both computational fluid dynamics (CFD) using finite-difference approaches [8], and computational structural dynamics (CSD) using finite-element methods [9]. To achieve full aeroelastic solutions, both of these methods must be utilized together. Aerospace vehicle structures are dominated by internal, discontinuous members such as spars, ribs, panels, and bulkheads. The finite-element method (FEM), which is fundamentally based on discretization along physical boundaries of different structural components, is computationally efficient for solving these types of structures problems. The external aerodynamics of aerospace vehicles, on the other hand, is dominated by field discontinuities such as shock waves and flow separations, which are solved efficiently by finite-difference (FD) computational methods. Figure 3 illustrates a time-accurate coupled fluid-structure aeroelastic analysis process. It is a step-by-step time-integration procedure that independently computes the fluid and structural solutions and passes results between the two regimes at common boundaries. At every time step the pressure data (Cp) from the CFD computations are mapped onto structural grid points and the force vector {Z} is computed. Using {Z} the structural displacements are then computed from CSD analysis and resulting deflections are remapped onto fluid grids that move accordingly. The specific interface techniques used depend on the type of structural modeling being done.

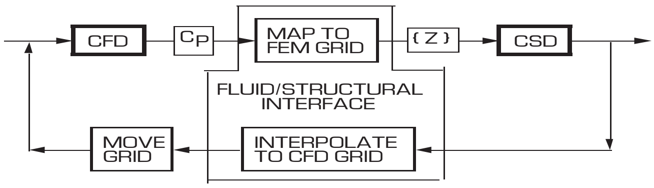

Figure 3. Coupled fluid structural analysis

Fluid and structure domains can be modeled at various levels of complexity, both in terms of the geometries and the physics employed. For design, aerodynamic data may be used at several levels of fidelity, from low-fidelity look-up tables to high-fidelity Navier-Stokes solutions. Similarly, structural data can range from low-fidelity assumed shape functions to detailed three-dimensional finite elements. As the modeling fidelity increases, it becomes more difficult to handle complex geometries. Figure 4 illustrates typical levels of modeling complexities involved in fluids and structures analyses. Techniques for interfacing the two computations depend on the levels of fidelity used for each domain. To date, general-purpose codes such as NASTRAN [9] can compute aeroelasticity for complex geometries using 3D finite element structures directly coupled with linear analytical aerodynamic methods. Codes based on CFD, such as HiMAP [10], can compute aeroelasticity using 3D Navier-Stokes equations coupled with simple 3D finite element structural equations. This study investigates coupled computations using different levels of fidelity on HPC resources.

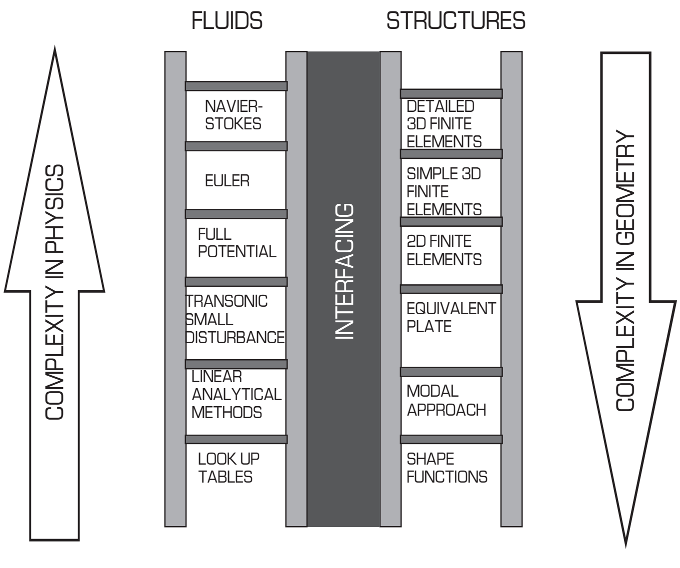

Figure 4. Varying levels of fidelity in modeling of fluids and structures.

3. INDEXING THE APPLICATIONS

Computational expense increases with the geometric complexity of the model and the fidelity of the equations solved. This paper provides a quantitative measure of the expense associated with these two factors in aeroelasticity analyses. A simpler approach to assigning an expense index has previously been presented by the author in Reference 11.

In this report, a fidelity-complexity index (FCI) is assigned to each application. It is assumed that the complexity of the problem is represented by the number of intersecting surfaces in the geometry and the grid sizes used for modeling flows and structures. It is also assumed that the complexity arising from intersecting surfaces has a strong impact on CFD grids and no impact on FEM grids. The numbers of intersecting surfaces for typical aerospace configurations considered by HPC users is given in Figure 5.

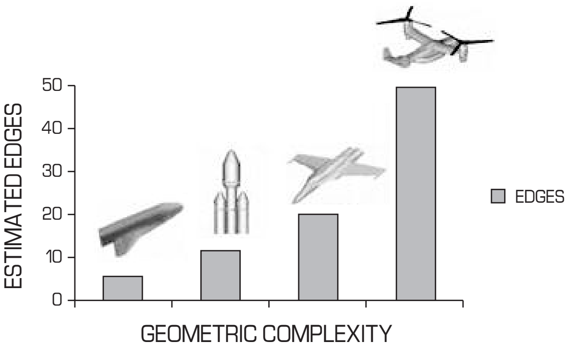

Figure 5. Number edges for different aerospace configurations.

The fidelity of the fluids modeling depends on the types of equations solved and the turbulence model used. For structural computations, the fidelity depends on the type of element used. The degree of fidelity for both fluids and structures can be measured by the number of floating-point operations needed to solve the equations. In this report the fidelity of fluids computations is measured by the number of floating-point operations involved per grid point per local time step. The Euler option of the diagonal form of the Pulliam-Chaussee scheme [12], which requires about 1,400 floating-point operations per grid point, is used as a reference number. The fidelity for structures is measured using the number of degrees of freedom (DOF) per element, with the predominant 9-DOF triangular plate element used as a reference.

In addition to CPU and memory requirements, the need for other resources such as I/O is also significantly greater for dynamic aeroelastic computations than for steady-state computations. This factor is also accounted for in assessing FCI.

The Fidelity/Complexity Index (FCI) can be represented as

$$FCI = (Fp*Sp*Fg*Sg*Ne*SUA) \tag{1}$$

Fp = number of floating point operation per grid point per step divide by 1,400
Sp = number of degrees of freedom per element divided by 9
Fg = fluids grid size divided by 100,000.
Sg = structural elements divided by 1,000 (only for strongly coupled cases)
Ne = number of edges divided by 10
SUA = for steady = 1, unsteady = 2, static aeroelastic = 3, and dynamic aeroelastic = 4

For example, a dynamic aeroelastic computation (SUA = 4) over a typical wing-body configuration (10 edges) using an Euler solver based on the Pulliam-Chuassee [12] algorithm (Fp = 1) with 1M grid points and 1,000 triangular plate (9-DOF) elements yields an FCI of 40.

The only additional effort required for aeroelastic computations using modal structures is to solve a set of 5 to 20 ordinary differential equations. Additional time for these aeroelastic computations is negligible compared to that needed to solve the flow equations. As a result, Sg is set to unity when modal structures are used.

4. SURVEY RESULTS

A web-based questionnaire was sent to engineers/scientists who use HPC resources for computational aeroelasticity. The questionnaire was designed to capture the nuances of high-performance computational aeroelasticity, with an emphasis on use of parallel computers. The following sections are based on the responses to the questionnaire.

4.1 Parallel Methods.

Most of the CFD and FEM codes used MPI for parallelization. Use of OpenMP was limited to shared-memory configurations such as SGI. Some attempts to use UNIX native message passing instructions were reported, but this approach suffered portability issues.

In order to improve solution quality, it is often necessary to either refine the grid (h-method) or increase the order of accuracy (p-method). The p-method is generally preferred over the h-method due to its faster convergence. However, the computational time rapidly increases with use of the p-method.

The straightforward way to alleviate the increase in computational time is to utilize domain decomposition with MPI. The downside of this approach is that it requires rearrangement of grid zones, which can be time consuming. With shared-memory systems, however, MPI and OpenMP can be combined to avoid rezoning. Figure 6 illustrates the advantage of this strategy [13]. HPC applications use this technique for spectral and hp finite element discretizations [14]. The main drawback that may prevent further use of this technique is lack of portability.

Some applications use the UNIX-based message passing protocol MPL. Although it is less portable, MPL can perform better than either MPI or OpenMP if it is optimized for a particular hardware. Figure 7 shows the relative percentage use of these three different parallel protocols, based on a brief review of HPC applications.

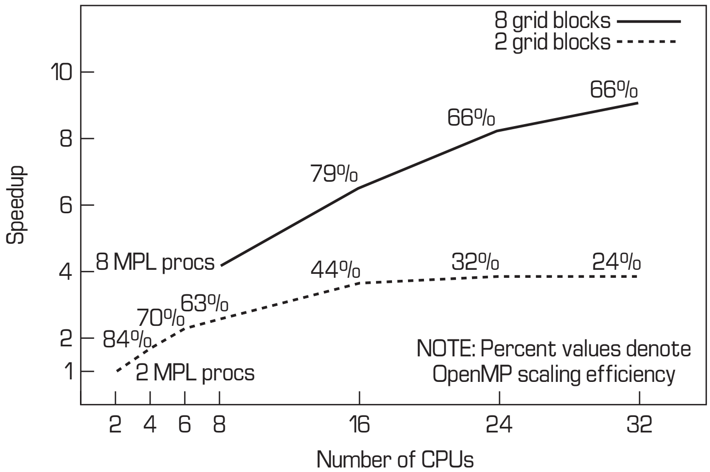

Figure 6. Performance of OpenMP on a Sun platform [13]

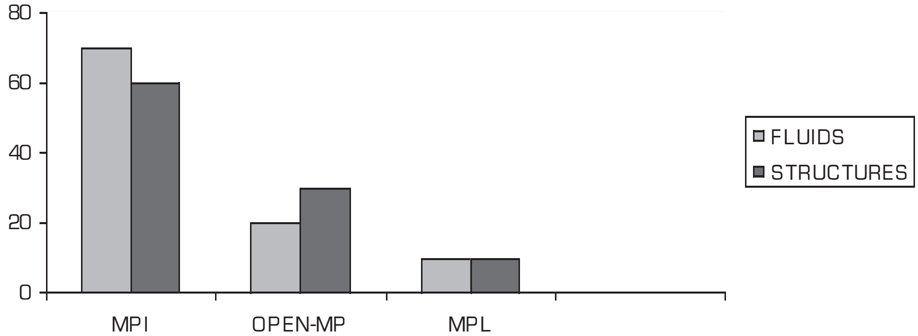

Figure 7. Percentage use distribution of MPI, OpenMP, and MPL

4.2 Load Balancing

Zonal grids for both CFD and FEM codes are designed to accurately model a specific configuration. Depending on the configuration, the size of the grid in each zone can vary significantly. Even larger variations occur when using the Navier-Stokes equations to solve viscous flows. Figure 8 shows the grid zone distribution for a full aircraft, which has 35 zones and a total of 9.3M grid points. The number of grid points in the largest zone is 15 times the number in the smallest zone. Simplistic grid zone mapping, which places each grid zone into a separate processor, leads to inefficient parallel performance because it does not incorporate load balancing. The efficiency percentage factor, *e*, due to lack of load balancing can be expressed as

$$e = 100t /\{p*m*r\} \qquad (2)$$

where *t* is the total grid size, *m* is the number of processors used, *p* is the optimum grid size per processor, and *r* is the ratio of largest to smallest grid size. For the configuration shown in Figure 8, the efficiency is 6 percent where the optimum size per processor is 300K grid points on an Origin2000 system. It is assumed that each zone is assigned to one processor.

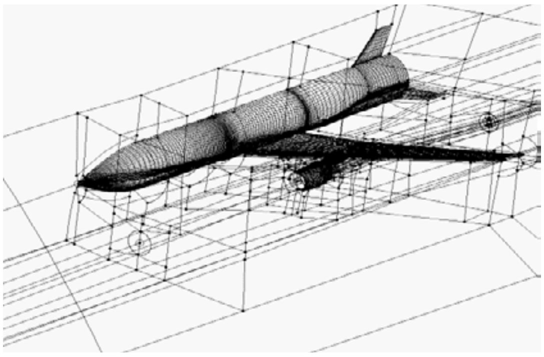

Figure 8. Complex grid arrangement for a typical transport aircraft

When using complex configurations, a load balancing scheme is needed. Procedures to load-balance using a simplified zone-splitting approach and an advanced zone-coalescing approach are given in two references 15 and 16, respectively. Results of applying a load balancing scheme to the configuration shown in Figure 8 are given in Figure 9. In this example, the ratio of largest to smallest zone *r* is decreased to 1.23 with the rearrangement. Also, zones are grouped such that they can be assigned to 28 processors instead of 35. As a result, the efficiency factor *e* is increased from 6 percent to 87 percent.

Load balancing procedures that can retain the characteristics of the original zonal arrangement are still under development. Patched structured grids are highly suitable for load balancing, as illustrated in Ref. 16, both for MPI and OpenMP architectures. None of the survey responses reported use of load balancing techniques for aeroelasticity computations. Some single-discipline users stated that they employed manual load balancing, but details of these approaches were not available. Based on the data received, typical efficiency factor values *e* computed for different grid sizes are shown in Figure 10.

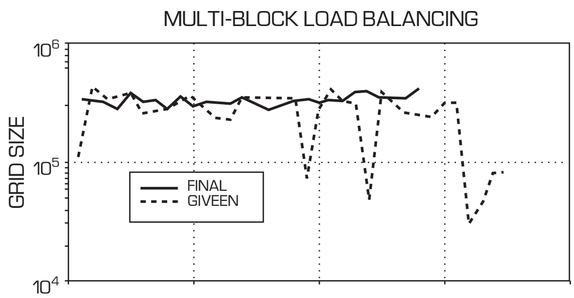

Figure 9. Grid points per processor with and without load balancing scheme.

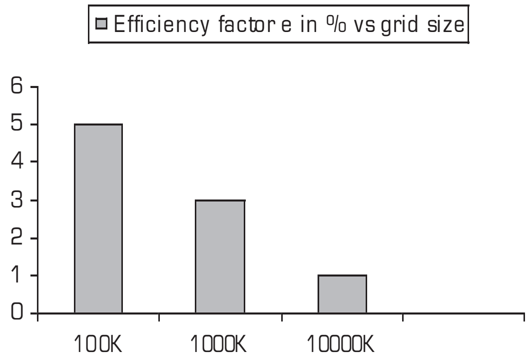

Figure 10. Grid sizes and efficiency factors.

For unstructured grids, automated load balancing tools such as METIS [17] were used by some CSM and CFD applications, but users did not report performance improvements. Using this type of automated tool could slow solution convergence, particularly for structural computations, if the topological constraints of the original grid are not accounted for. For example, this can occur if zone sections involving both wing and body fall into same computational domain. Since the wing is flexible and the fuselage is rigid, a single stiffness matrix for both may become ill-conditioned.

5. CASE STUDIES

Based on the survey responses received from HPC resource users, several case studies were conducted. For each case, a FCI value was determined from the information given by the authors. A summary of these results is shown in Figure 11, along with results for an advanced aeroelastic application [18] formerly sponsored by NASA under the High Performance Computing and Communication (HPCC) program, which is shown for comparison. Use of lower-fidelity structural equations led to a lower FCI for most applications. As demonstrated for NASA High Speed Civil Transport (HSCT) project [19], an FCI of 12,000 (based on static aeroelastic computations on a configuration with 10M fluid grid points, 20 edges and 20,000 structural finite elements) is possible on a 256-node SGI Origin2000 system. Even larger FCI values can be achieved on current HPC resources. Rapid progress is being made in using unstructured grids tends to achieve increasing values of FCI with more efficient utilization of advancing HPC resources.

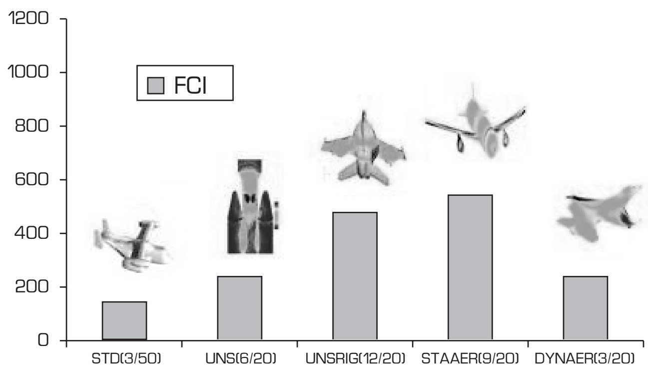

Figure 11. Fidelity-Complexity Index (FCI) for typical aerospace applications. STD = Steady, UNS = Unsteady, UNSRIG = Unsteady flow over moving rigid configuration, STAAER = Static Aeroelasticity. DYNAER = Dynamic Aeroelasticity. (Grid in M/edges)

6. CONCLUSIONS

1. HPC resources have significantly advanced computational aeroelasticity.
2. Parallel performance of most fluid solvers, particularly those using MPI, is almost linear.

3. For computational aeroelasticity, flow solvers have advanced in fidelity faster than structural solvers. Flow solvers typically use the Navier-Stokes equations, while structural computations use either modal or simple 2D plate elements.
4. Use of load balancing to significantly improve parallel computational efficiency was seldom used, particularly for fluids. Few structural applications used an automated partitioning tool. Significant increase in use of load balancing tools is needed.
5. Use of lower-fidelity structural equations produced lower FCI values for most HPC applications. However, rapid progress is taking place in this area using unstructured grids for fluids and finite elements for structures

REFERENCES

1. Dobbs, S. K. and Miller, G. D., Self-Induced Oscillation Wind Tunnel Test of a Variable Sweep Wing. *Structural Dynamics Conference*, AIAA Paper 85-0739, Kissimmee FL, Apr. 1985.
2. Rizk, Y. and Guruswamy, G. P., Numerical Investigation of Tail Buffet of F-18 Aircraft. *AIAA 10th Applied Aerodynamics Conference*, AIAA-92-2673, Palo Alto, CA, June 1992
3. Eckstrom, C. V, Seidel, D. A and Stanford, M. C., Measurements of Unsteady Pressure and Structural Response for and Supercritical wing, *National Aeronautics and Space Administration* NASA TP 3443, Hampton VA, Nov. 1994.
4. Farmer, M. G and Hanson, P. W., Comparison of supercritical and Conventional Wing Flutter Characteristics, *National Aeronautics and Space Administration,* NASA TM X-72837, Hampton VA, May 1976.
5. Grove, D., Computational Fluid Dynamics Study of Abrupt Wing Stall on an F/A-18E, *High Performance Computing Modernization Program, Department of Defense, USA User's Group conference,* Monterey CA June2002.
6. Findlay, D., Numerical Analysis of Aircraft High Angle of Attack Unsteady Flows, *AIAA/CEAS Aeroacoustics Conf*, Kauai, Hawaii AIAA 2000-1946, June 2000.
7. Jones, K., Platzer, M., Rodriguez, D. and Guruswamy, G. P., On the effect of Area ruling on Transonic Abrupt Wing Stall. *Symposium Transsonicum IV* Gottingen, Sept. 2002.
8. Buning, P. G., Wong, T. C., Dilley, A. D., and Pao, J. L. , Computational Fluid Dynamics Prediction of Hyper-X Stage Separation Aerodynamics, *Journal of Spacecraft and Rockets*, Vol. 38, No. 6, Nov.-Dec. 2001, pp. 820-826.
9. NASTRAN User's Manual, *National Aeronautics and Space Administration,* NASA SP-222 (08), June 1986.
10. Guruswamy, G. P., Byun, C., Potsdam, M., and Farhangnia, M., User's Manual for HiMAP - A Multilevel Parallel Multidisciplinary Analysis Process. *National Aeronautics and Space Administration,* NASA TM-1999-209578, Sept 1999.
11. Guruswamy, G. P and Obayashi, S., Use of High Fidelity Methods in Multidisciplinary Optimization – A Preliminary Study. *AIAA MDO meeting* AIAA paper 2002-5638, Atlanta Sept. 2002.
12. Pulliam, T. H. and Chaussee, D.S, A Diagonal Form of an Implicit Approximate Factorization Algorithm. *Jl. Of Comp Physics*, Vol. 39, No. 2, 1981, pp 347-361.
13. Byun, C., Porting Experience With HiMAP On Sun HPC Environment. *SUN Users meeting*, Honolulu, Hawaii, April 2002.
14. Dong, S., Lucor, D., Symeonidis, V, Xu, J., and Karniadakis, G.E., Multilevel Parallelization Models: Application to VIV. *Proceedings of the 2003 Users Group Conference, IEEE Computer Society*, Los Alamitos, CA, June 2003.
15. Hatay, F., Byun, C., Guruswamy, G.P., Jespersen, D., and Rizk, Y., A Multi-Level Parallelization Concept for High Fidelity Multiblock Solvers, *Supercomputing 97*, San Jose, CA, Nov. 1997.
16. Guruswamy, G.P., Development and Applications of a Large Scale Fluids/Structures Simulation Process on Clusters, *Computers & Fluids*, 36 (2007), March 2007, 530-539.
17. Schloegel, K., Karypis, G. and Kumar, V., Unified Algorithm for Load-balancing Adaptive Scientific Simulations. *Supercomputing 2000,* San Jose CA, Nov 2000.

18. Potsdam, M.A. and Guruswamy, G. P., A Parallel Multiblock Mesh Movement Scheme for Complex Aeroelastic Applications. *AIAA Aerospace Sciences Conf* AIAA 2001-0716, Reno NE Jan. 2001.
19. Guruswamy, G. P., HiMAP- A portable Super Modular Multilevel Parallel Multidisciplinary Process for Large Scale Analysis. *Elsevier Advances in Engineering Software* Vol. 31 2000.