A New Modular Approach for Tightly Coupled Fluid/Structure Analysis

GP Guruswamy[1]
NASA Advanced Supercomputing Division
Ames Research Center, Moffett Field, CA, U.S.A.
Guru.P.Guruswamy@nasa.gov

ABSTRACT
Static aeroelastic computations are made using a C++ executable developed for closely coupled fluid-structure analyses. The fluid flow is modeled using Euler/Navier-Stokes equations and the structure is modeled using finite elements. The FORTRAN-based fluid and structure codes are integrated in a C++ environment in which the flow and structural solvers are treated as separate object files, and the data exchange between them is accomplished using I/O. Use of this technique to solve transonic flow over a partially flexible surface is presented, and its ability to accurately predict flow-induced deformations associated with nonlinear structures is demonstrated.

NOMENCLATURE

A	area of triangular plate element
A_1, A_2, and A_3	areas of the sub-triangles
a_∞	free-stream speed of sound
c	length of the root chord
D	structural displacement at fluid grid point
d_i, d_j and d_k	structural displacements at i, j and k nodes
E, F and G	Euler flux vectors
e	total enthalpy
P	total aerodynamic load at fluid grid point
P_i, P_j, and P_k	contribution of total force P to structural nodal forces i, j and k
U, V, and W	structural displacements at nodes in x, y, and z directions
U_x, V_y and W_z	rotational degrees-of-freedom at node about x, y and z axes
u, v, and w	flow velocities in x, y and z directions
Q	conserved quantity vector
ρ_∞	free-stream density of air
γ	ratio of specific heats.

1. INTRODUCTION
Aeroelasticity involving strongly coupled fluids and structures is an important element in aerospace vehicle design. Monolithic (off-line) software that computes fluid and structure interactions using low-fidelity methods, such as linear aerodynamic flow equations coupled with modal structural equations, are well established.[1] Although these low-fidelity approaches are used for preliminary design, they are not adequate for the final design of aerospace vehicles, which requires high-fidelity analysis of complex flow-structure interactions. For example, modern supersonic transports with highly swept wings can experience flow-induced aeroelastic oscillations due to strong coupling of unsteady leading edge vortices and wing structures.[2] Figure 1 illustrates the sustained oscillation that occurs for a typical supersonic transport flying in the transonic regime at a moderate 8 degrees angle of attack, degrading the flight quality of the aircraft. In order to predict this phenomenon, direct coupling of high-fidelity Euler/Navier-Stokes (ENS) equations with modal structures equations was needed.[3] Strong coupling of fluids, structures and controls is also an important element in the analysis of space planes[4], which can experience instabilities dominated by complex, nonlinear flows coupled with structural motions soon after separating from their carriers. The results presented in reference 4 show that a low-fidelity method was not adequate to completely understand this type of instability phenomenon.

[1]Senior Scientist

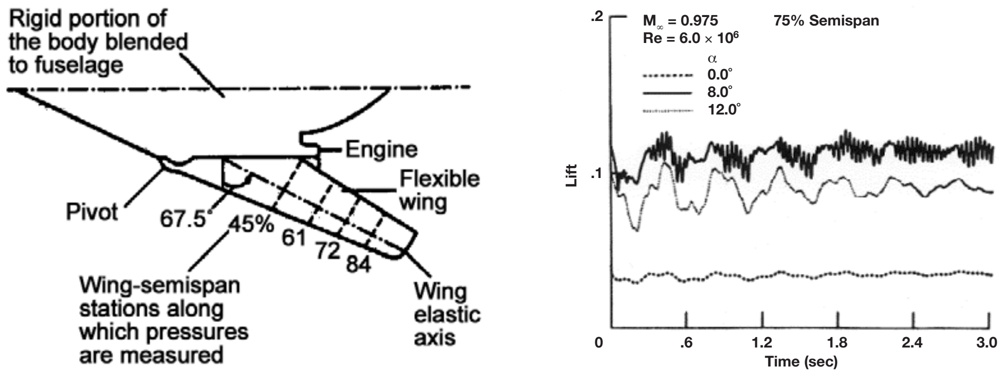

Figure 1.. Aeroelastic oscillations involving strongly coupled fluid-structure interactions

High-fidelity equations, such as the Euler/Navier-Stokes (ENS) for fluids directly coupled with finite elements (FE) for structures, are needed for accurate aeroelastic computations that involve these complex fluid-structure interactions. Because high-fidelity equations involve additional computational complexities from their higher-order terms, the coupling process is more elaborate when using high-fidelity methods than it is when using linear methods.

In recent years, significant single-discipline advances have been made individually in both computational fluid dynamics (CFD) using finite-difference approaches [5], and computational structural dynamics (CSD) using finite-element methods [6]. Both of these methods must be utilized together to obtain full aeroelastic solutions for coupled phenomena. The structures of aerospace vehicle are dominated by internal discontinuous members such as spars, ribs, panels, and bulkheads. The finite-element (FE) method, which is fundamentally based on discretization along physical boundaries of different structural components, is computationally efficient for solving these types of configurations. 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.

A major challenge for fluid-structure interaction studies is combining individual discipline codes into a single computational environment. To date, monolithic serial codes such as ENSAERO [7] have been developed using conventional FORTRAN-based programming techniques for mainframe supercomputers. In the last decade, several monolithic software tools that work on single image parallel systems, such as HiMAP [8], were also developed. These monolithic approaches, however, are typically limited to a specific class of problem and do not accommodate substitutions of alternate single discipline algorithms.

A new computational programming paradigm, known as problem solving environments (PSEs) [9], has evolved in the field of computer science during the last decade. The main purpose of the PSE approach is to provide an engineering workbench for designers to efficiently integrate single discipline codes. For example, SCIRun [9], a PSE developed at the University of Utah, provides an environment to integrate finite element based solvers for fluids and structures. PSEs are also designed to enable computations to run on either single image, parallel, or GRID (networks of remotely distributed computers)[10] computing environments. The primary attribute of a good PSE is that it requires minimal changes to the application codes while providing a seamless environment for the user.

This paper presents an approach to solving coupled static aeroelastic computations that can be ported to a PSE or GRID computing environment. The approach is based on a C++ executable that controls FORTRAN modules. A typical 3D aerospace problem of transonic flow over a thin flexible surface is analyzed using this method.

2. DESIGN OF C++ EXECUTABLE

In this application, it is assumed that fluid and structure solvers are independent executables. Interfaces from fluids to structures (FTOS) and structures to fluids (STOF) are also considered to be separate executables. All communications are made via I/O. This treatment facilitates the portability of the present development to distributed computing environments. Additional details about the C++ executable are given in reference 11 in addition to other than I/O based communications. A copy of I/O based C++ source code is given in Appendix A.

The C++ executable's analysis process includes two additional features that are important for coupled calculations. Firstly, to enable on-the-fly monitoring of convergence data during the computation, a 2D plotting function based on the open source software XMGRACE [12] is included. Secondly, the ability to save data in the FieldView [13] format at user-specified intervals is built in to allow necessary data to be saved for high-end graphic visualizations. Implementation of XMGRACE and FieldView are independent of new releases since they are based on primary input/output data formats. Figure 2 shows a flow diagram of the process.

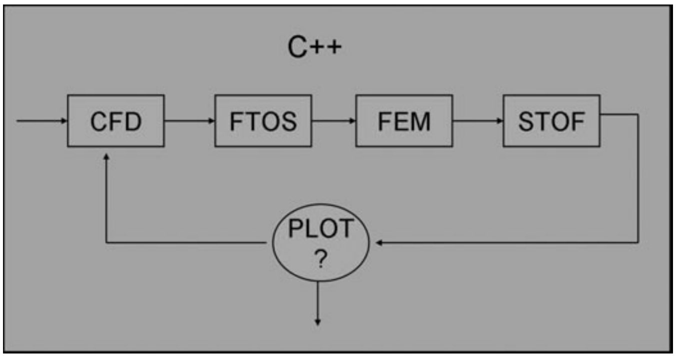

Figure 2. Flow diagram of analysis process.

Flow and Structural Equations

In this paper, computations are limited to Euler equations of motion. The strong conservative law form of the Euler equations is used for shock-capturing purposes. In non-dimensional[14] Cartesian coordinates form the equations can be written as

$$\frac{\partial Q}{\partial t}+\frac{\partial E}{\partial x}+\frac{\partial F}{\partial y}+\frac{\partial G}{\partial z}=\frac{\partial E_v}{\partial x}+\frac{\partial F_v}{\partial y}+\frac{\partial G_v}{\partial z} \tag{1}$$

where the conserved quantity vector, Q, and the Euler flux vectors, E, F, G, are:

$$Q=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ e\end{bmatrix}, E=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uw\\ u(e+p)\end{bmatrix}, F=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vw\\ v(e+p)\end{bmatrix}, G=\begin{bmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ w(e+p)\end{bmatrix} \tag{2}$$

In which a_∞ is the freestream speed of sound; density ρ is nondimensionlized by a_∞ the total energy per unit volume, e is nondimensionalized by $\rho_\infty a^2{}_\infty$; and the time is nondimensionalized by c/a_∞, where c is the characteristic length and is speed a_∞. Pressure can be computed from the ideal gas law as

$$p = (\gamma - 1)[e - 0.5\ \rho_\infty(u^2 + v^2 + w^2)] \tag{3}$$

where "γ" is the ratio of specific heats. Equation 1 is solved to determine the pressure p by using a streamwise upwind algorithm available in the GO3D [15] flow solver.

The finite element approach is used to solve structural equations. In this method an 18-degree-of-freedom (DOF) plate/shell element in TRIP3D [16] is used. The element details are shown in Figure 3. U, V, and W are displacement degrees-of-freedom and Ux, Vy and Wz are the corresponding rotational degrees-of-freedom at each node. The finite element equations are solved using a standard procedure described in Chapter 13 of Reference 6.

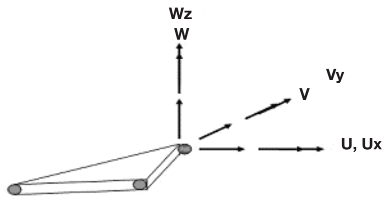

Figure 3. 18-DOF Triangular Plate/Shell Element

3. FLUID-STRUCTURE INTERFACES

The Euler/Navier Stokes flow solvers use either patched, overset, or unstructured grids. Figure 4 shows these three types of CFD surface grids. Surface grid data for the CFD flow solvers is triangulated in order to interface with the structures codes, which are based on irregular unstructured grids. Tools such as MIXSUR [17], developed in association with the CFD code OVERFLOW[5], are available for efficient triangulation of these complex surface grids.

Once triangulated data is available, it is interfaced with the structures data using the area coordinate approach [18]. In this robust method, fluid grid points are identified with respect to finite element structural nodes i, j and k, as shown in Figure 5. The force P at a given fluid grid point is computed from the control area associated with that point. The contribution of the total force P to the structural nodal forces P_i, P_j and P_k at the point are computed using

$$P_i = (P * A_1)/A \qquad P_j = (P * A_2)/\mathrm{A} \qquad P_k = (P * A_3)/A \tag{4}$$

where A_1, A_2, and A_3 are areas of the sub-triangles, and A is the total area, as shown in Figure 5. This procedure is repeated for all fluid grid points. The structural analysis displacements d_i, d_j, and d_k are interpolated to each fluid grid point using

$$\mathrm{D} = (d_i \times A_1 + d_j \times A_2 + d_k \times A_3)/A \tag{5}$$

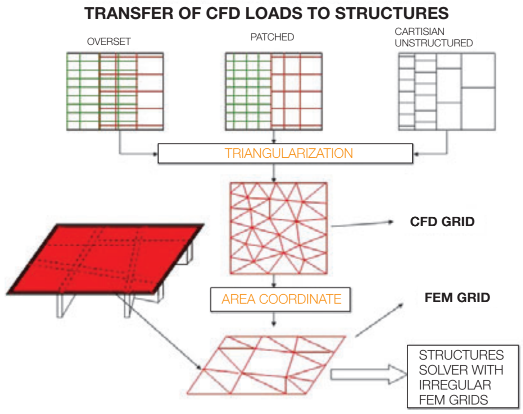

Figure 4. Surface grid topologies of fluid and structure.

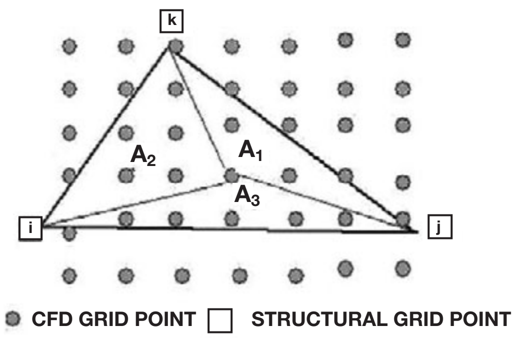

Figure 5. Interpolation based on triangular area coordinates.

4. COUPLED AEROELASTIC COMPUTATIONS

Coupled aeroelastic computations play an important role in the design of aerospace vehicles. Static divergence of launch vehicles is a typical example. In order to test the presented process for coupled static aeroelastic computations using the FORTRAN-based CFD and FEM codes, flow over a partially flexible flat plate was analyzed as a sample problem. The flow is modeled using the streamwise upwind 3D Euler/Navier-Stokes solver GO3D [15] and the structures code used is TRIP3D [16].

Initially, static aeroelastic computations are performed. Loads are computed using GO3D, and are then transferred to TRIP3D through an FTOS interface module. This interface interpolates loads from the CFD structured surface grids to finite element nodes. CFD structured surface grids are first mapped to FE triangular grids, and then the area coordinate approach is used to interface data between the fluids and structures grids (Eq. 4). Resulting structural stresses and deformations are computed using TRIP3D and then interpolated back to the CFD grids through an STOF interface module using the area coordinate method (Eq. 5). The results are monitored on the fly for convergence using the XMGRACE [11] plot utility.

The physical problem examined is flow over a partially flexible surface. Such configurations are common in space vehicles and coupled fluid-structure interaction computations are often essential in designing their components. A classic example is panels of lifting surfaces that can experience flutter [19]. Figure 6 shows the configuration considered for this analysis. The size of the flexible plate is assumed to be 1 inch long, 0.5 inches wide and 0.005 inches thick. The size of the flow surface is 5 inches long and 1.5 inches wide. A dynamic pressure of 1.4 psi is assumed for all computations.

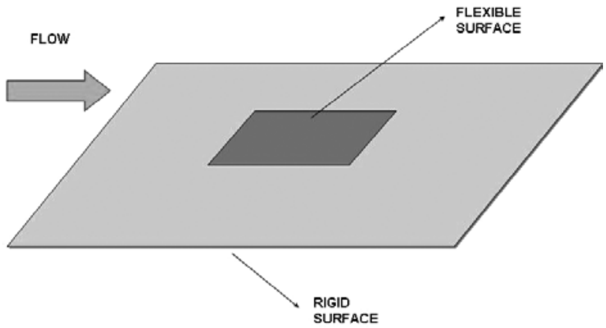

Figure 6. Geometric details of the configuration considered.

The 3D flow over the plate is calculated using the Euler option of the GO3D solver, with a grid size of 97×31×34 used to model the flow surface. The surface and sectional grid regions are shown in Figure 7. Structural deformations and stresses are computed using 64 plate elements for the TRIP3D FEM grid, as shown in Figure 8. Both GO3D and TRIP3D are well validated as independent solvers for this type of configuration [16].

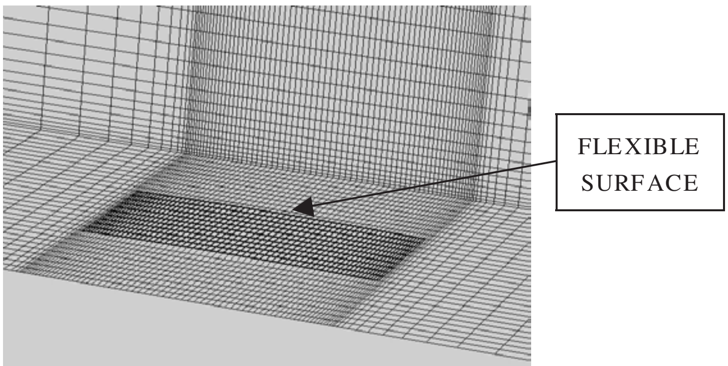

Figure 7. Portions of CFD grid

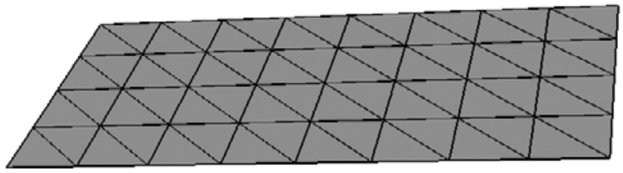

Figure 8. FEM grid.

To obtain accurate computations it is important to communicate data between fluids and structures at every timestep, particularly when either system is non-linear. Computations were made updating the fluid-structure information every 1, 2, and 4 timesteps for a test case at Mach = 0.85 with moderately non-linear flow and selected linear structural parameters. Figure 9 shows the effect of update frequency on the displacement results. Updating information every 2 time step introduced significant error, and updating information every 4 steps was not acceptable.

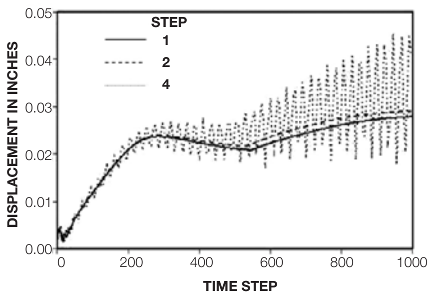

Figure 9. Effect structural update frequency on displacements.

Figure 10 shows the distribution of pressure coefficient (Cp) on half the surface and vertical plane at mid-span. Adaptation of the CFD grid to structural deformations can be seen in the figure. Due to the flexibility of the surface there is a significant change in pressure distribution. The structural deformation and corresponding stress distribution is shown in Figure 11. The variation of flow pressure (Cp) leads to large structural stresses. The peak structural stress occurs close to 65%-span near the area where Cp has large negative values.

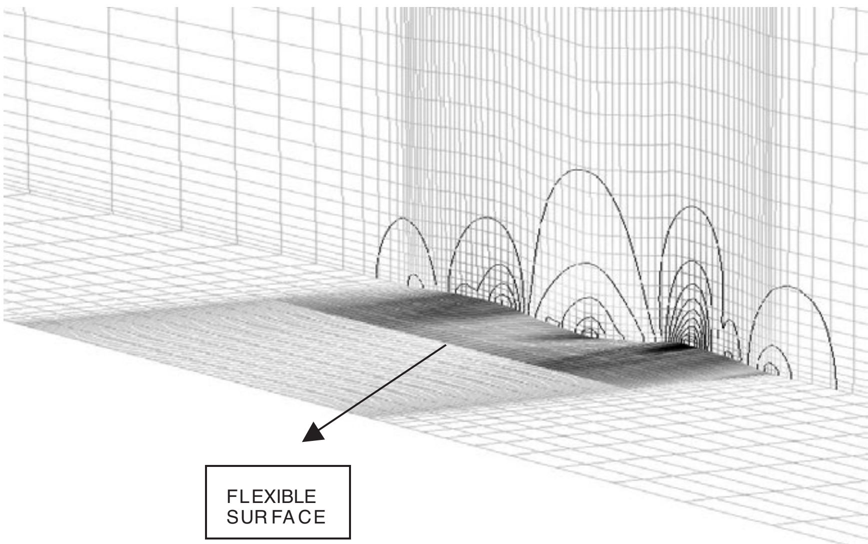

Figure 10. Distribution of pressure coefficient at M = 0.85.

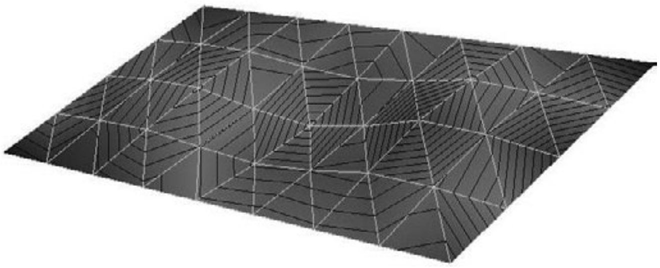

Figure 11. Distribution of bending stresses (shaded area) and Cp (contour lines) on deformed structural surface.

5. NON-LINEAR STRUCTURES

Flow-structure interactions become increasingly important during structural failures, when structural behavior is nonlinear. To demonstrate the ability of the present development to accurately predict deformations when both flows and structures are nonlinear, a computation with assumed nonlinear structures is presented. For this test case, the structures are assumed to become nonlinear after a certain number of timesteps. The effective thickness of the plate is given by

$$T_{\mathbf{new}} = T_{\mathbf{org}}\,(d)^{n} \tag{7}$$

where T is the thickness of the plate, d is a decay factor, and n is the number timesteps after decay has started.

For the test case, the configuration shown in Figure 6 is modeled at Mach 0.85, with a decay factor of d = 0.9995 starting after 1,000 steps. Figure 12 shows the resulting deformations at midpoint, both with and without structural decay. For comparison, the displacement at the midpoint is also computed assuming that there is no coupling between fluids and structures. The resulting value is 35% lower than that obtained by a fully coupled calculation because the uncoupled computation does not account for changes in the flow prior to computing the final displacement. Errors associated with uncoupled computations, as demonstrated in Figure 9, are higher when both fluids and structures are non-linear.

Similar studies were conducted with simply supported boundary conditions. A summary of results is presented in Table 1. Both for fixed and simply supported boundary conditions, the percentage error due to uncoupled computations is greater when the structure is nonlinear. Also, errors due to uncoupled computations occur more for the simply supported plate than for the fixed plate.

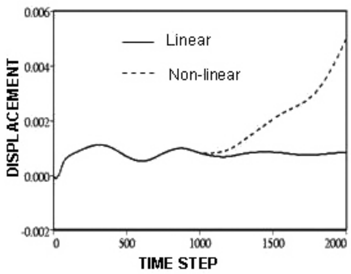

Figure 12. Comparison of displacements between linear and nonlinear structures.

Table 1. Effect of Boundary Conditions

Case	FEM	Coupled	Uncoupled	Diff
Fixed	L	0.831E-03	0.723E-03	–13%
Fixed	NL	4.999E-03	3.227E-03	–36%
SS	L	3.282E-03	2.401E-03	–27%
SS	NL	1.423E-02	7.558E-03	–46%

SS: Simply Supported, L: Linear, NL: Nonlinear

6. CONCLUSIONS

A procedure for solving tightly coupled fluid-structure interactions using distributed computing systems is presented. The infrastructure accommodates general CFD and FEM codes. Data management and communication is done using a C++ interface, which helps port the present development to distributed computing system. The need to update fluid-structure interface data when one of the systems is nonlinear has been demonstrated for a 3D problem. Use of the present development to predict flow-induced deformations is demonstrated for cases with strong coupling due to nonlinear structural behavior.

REFERENCES

1. NASTRAN User's Manual: NASA SP-222 (08), June 1986.
2. Dobbs, S., and Miller, G. D., Self-Induced Oscillation Wind Tunnel Test of a Variable Sweep Wing, *AIAA Structural Dynamics Conf*, AIAA Paper 85-0739, Kissimmee, FL Apr. 1985.
3. Guruswamy, G. P., Vortical Flow Computations on a Blended Wing-Body Configuration, *AIAA Journal*, Vol. 30, No. 10, 1992. pp. 2077-2084.
4. Blades, E. L., Ruth, M., and Fuhrmann, H. D., Aeroelastic Analysis of the X-34 Launch Vehicle, AIAA Structural dynamics Conf, St. Luis, MO, AIAA-99-135, Apr. 99.
5. 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, 2001, pp. 820-826.
6. Yang, T. Y., *Finite Element Structural Analysis*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1986
7. Guruswamy, G. P., ENSAERO – A Multidisciplinary Program for Fluid/Structural Interaction Studies, *Computing Systems Engineering*, Vol. 1 Nos. 2-4, 1990.
8. Guruswamy, G.P., Development and Applications of a Large Scale Fluids/Structures Simulation Process on Clusters, *Computers & Fluids* 36 (2007) 530-539, March 2007.
9. Parker, S., Weinstein, D., and Johnson, C., *Modern Software Tools in Scientific Computing* 1-44 Birkhuaser press, 1997.
10. Foster, I., Kesselman, C., *The GRID Blue Print for a new Computing Infrastructure*. Morgan Kaufmann Publisher Inc, San Francisco 2002.
11. Guruswamy, G.P. and Cheung, C., 2D Static Fluid/Structure Analysis Using PSE, *The 2003 International Conference on Parallel and Distributed Computing*, Las Vegas, June 2003.
12. XMGRACE , *Free Software Foundation, Inc.,* 675 Mass Ave, Cambridge, MA 02139, UK.
13. Fieldview Version 6 Reference Manual, Intelligent Light, Lyndhurst, NJ, May 1999.
14. Peyret, R. and Viviand, H., Computation of Viscous Compressible Flows Based on Navier-Stokes Equations, AGARD-AG-212, 1975.
15. Obayashi, S., Guruswamy, G. P., and Goorjian P. M., Streamwise Upwind Algorithm for Computing Unsteady Transonic Flows Past Oscillating Wings, *AIAA Jl.,* Vol. 29, No. 10, Oct. 1992. pp. 1668-1677.
16. Byun, C. and Guruswamy, G. P., Wing-Body Aeroelasticity on Parallel Computers, *Jl of Aircraft*, Vol 33, No 2, Mar-April 96, pp. 421- 428.
17. Chan, W. M., and Buning, P.G., User's Manual for FOMOCO Utilities-Force and Moment Computation Tools for Overset Grids, *National Aeronautics and Space Administration* NASA TM July 1996.
18. Bhardwaj, M., Kapania, R., Reichenbach and Guruswamy, G.P. , Computational Fluid Dynamics/Computational Structural Dynamics Interaction Methodology for Aircraft Wings, *AIAA Jl* Vol. 36, No. 12, 1998, pp. 2179-2186.
19. Korenecki, A., Dowell, E. H., and O'Brien, J., On the Aeroelastic Instability of Two-Dimensional Panels in Uniform Incompressible Flow, *Jl of Sound and Vibration,* Vol. 47 (1976) pp. 163-178.

APPENDIX-A

```
/* This will run the Trip3D and GO3d modules */
#include <stdlib.h>
#include <stdio.h>
#include <iostream.h>
#include <fstream.h>
#include <grace_np.h>
/* function to invoke graphics */
int main(){
 int prnfre,titer,strupd;
 cout<<"ENTER ITERATIONS : ";
 cin >> titer;
 cout<<" FREQUECY for PLOT : ";
 cin>> prnfre;
 cout<<"ENTER FREQUECY OF STRUCTURES UPDATE : ";
  cin>>strupd;
 /*   to initialize data */
      system("sh /u/wk/guru/PSEFW/FLOW/init");
      system("sh /u/wk/guru/PSEFW/STRU/init");
      system("sh /u/wk/guru/PSEFW/RESULTS/remove");
//////Initialize XMGRACE/////////////
 if(GraceOpen(2048)==-1){
  cout<<"Can't run Grace."<<endl;
 }
 GracePrintf("world xmax 2000");
 GracePrintf("world ymax 0.1");
 GracePrintf("world ymin -0.1");
 GracePrintf("xaxis tick major 200");
 GracePrintf("xaxis tick minor 5");
 GracePrintf("yaxis tick major 0.02");
 GracePrintf("yaxis tick minor 0.01");
 GracePrintf("s0 on");
 GracePrintf("s0 symbol 0");
 GracePrintf("s0 symbol size 0.3");
 GracePrintf("s0 symbol fill pattern 1");
system("cp -f /u/wk/guru/PSEFW/feminput.dat /u/wk/guru/PSEFW/STRU/input.dat");
system("cp -f /u/wk/guru/PSEFW/cfdinput.dat /u/wk/guru/PSEFW/FLOW/pltdyn.inp");
//////END INITIALIZE///////////////
for ( int i = 0 ; i < titer ; i++ ){ //iterations
 cout<<" ITERATION NO = "<<i<<endl;
        system("sh /u/wk/guru/PSEFW/FLOW/run");
       system("cp -f /u/wk/guru/PSEFW/FLOW/cfdload.dat /u/wk/guru/PSEFW/FTOS/cfdload.dat");
       int chkstr= (i/strupd)*strupd;
       if(chkstr==i) {
        cout <<"STRUCTURES UPDATED :" << endl;
       system("sh /u/wk/guru/PSEFW/FTOS/run");
       system("cp -f /u/wk/guru/PSEFW/FTOS/femload.dat /u/wk/guru/PSEFW/STRU/femload.dat");
       system("sh /u/wk/guru/PSEFW/STRU/run");
       system("cp -f /u/wk/guru/PSEFW/STRU/femdisp.dat /u/wk/guru/PSEFW/STOF/femdisp.dat");
       system("sh /u/wk/guru/PSEFW/STOF/run");
       system("cp -f /u/wk/guru/PSEFW/STOF/cfddisp.dat /u/wk/guru/PSEFW/FLOW/cfddisp.dat");
       ////////////////////////
     ifstream infile;
 infile.open("/u/wk/guru/PSEFW/STRU/fort.9",ifstream::in);
 float displacement;
 infile>>displacement;
```

```
  infile.close();
 cout<<displacement<<endl;
 if(GraceIsOpen()){
  GracePrintf("g0.s0 point %d, %f", i, displacement);
  GracePrintf("redraw");
 }
              ////////////////////////
        system("sh /u/wk/guru/PSEFW/STOF/run");
        system("cp -f /u/wk/guru/PSEFW/STOF/cfddisp.dat /u/wk/guru/PSEFW/FLOW/cfddisp.dat");
        }
    //////////Copy FILES fort.2 and fort.3 //////////////
    char COPYFORT2[256] = "cp -f /u/wk/guru/PSEFW/FLOW/fort.2
/u/wk/guru/PSEFW/RESULTS/grid";
    char COPYFORT3[256] = "cp -f /u/wk/guru/PSEFW/FLOW/fort.3
/u/wk/guru/PSEFW/RESULTS/q";
    char COPYFORT4[256] = "cp -f /u/wk/guru/PSEFW/STRU/fort.4
/u/wk/guru/PSEFW/RESULTS/strs";
    char suffix[20];
    for(int r; r < 20 ; r++){
     suffix[r]= '\0';
    }
    int length = 10;
    int digits =1;
    while (length <= i) {
        length = length *10;
        digits = digits+1;
    }
    length = length/10;
    int tempi = i;
    for(int k = 0; k < digits; k++){
        suffix[k] = (int)(tempi/length) + 48;
        tempi = tempi % length
        length = length / 10
    }
    int chkfre = int(i/prnfre)*prnfre;
    if(chkfre==i)
     {
          system(strcat(&COPYFORT2[0],&suffix[0]));
          system(strcat(&COPYFORT3[0],&suffix[0]));
    if(chkstr==i){system(strcat(&COPYFORT4[0],&suffix[0]));}
    cout << " PLOTTED FILES : " << chkfre<<endl;
     }
    system("rm -f /u/wk/guru/PSEFW/FLOW/fort.2");
    system("rm -f /u/wk/guru/PSEFW/FLOW/fort.3");
    if(chkstr==i){system("rm -f /u/wk/guru/PSEFW/STRU/fort.4");}
////////////END COPY////////////////////////////////
}//for
    return 0;
} //main
```