

## ADVANCES IN FEFLO

Rainald Löhner<sup>1</sup>, Chi Yang<sup>1</sup>, Juan Cebal<sup>1</sup>, Orlando Soto<sup>1</sup>, Fernando Camelli<sup>1</sup>,  
Joseph D. Baum<sup>2</sup>, Hong Luo<sup>2</sup>, Eric Mestreau<sup>2</sup>, Dmitri Sharov<sup>2</sup>  
Ravi Ramamurti<sup>3</sup>, William Sandberg<sup>3</sup> and Choong Oh<sup>3</sup>

<sup>1</sup>School of Computational Sciences and Informatics  
M.S. 4C7, George Mason University, Fairfax, VA 22030-4444, USA

<sup>2</sup>Applied Physics Operations  
Science Applications International Corp., McLean, VA 22102, USA

<sup>3</sup>Laboratory for Computational Physics and Fluid Dynamics  
Naval Research Laboratory, Washington, D.C. 20375, USA

## ABSTRACT

This paper summarizes the major improvements and developments that have taken place during the last year for FEFLO, a general-purpose CFD code based on adaptive, unstructured grids. All aspects of a comprehensive CFD capability: gridding, solvers, mesh movement techniques and diagnostics saw important advances, and are treated.

## 1. INTRODUCTION

FEFLO was conceived as a general-purpose CFD code based on the following general principles:

- Use of unstructured grids (automatic grid generation and mesh refinement);
- Finite element discretization of space;
- Separate flow modules for compressible and incompressible flows;
- ALE formulation for moving grids;
- Edge-based data structures for speed;
- Optimal data structures for different architectures;
- Bottom-up coding from the subroutine level to assure an open-ended, expandable architecture.

The code has had a long history of relevant applications [Bau91, Bau93, Ram93, Bau94, Bau95, Bau96, Bau97, Bau99, Ram99]. Over the last three years, FEFLO has been ported to both shared memory [Löh98, Tus98, Sha00] and distributed memory [Ram93, Löh95, Ram96] machines.

This paper summarizes the major developments that have taken place during the year 2000 for FEFLO. The topics covered include:

- Grid generation:
  - Advances in the gridding of surfaces given as discrete data;
  - Improvements in parallel grid generation;
- Incompressible solver modules:
  - New implicit-advection projection solver;
  - Pressure boundary conditions for internal flows;
  - Arbitrary number of advective/diffusive species;
- Compressible solver modules:

- Two matrix-free implicit compressible solvers;
- Body/mesh movement modules:
  - Overlapping grids (unstructured chimera);
  - Avoidance of geometry recalculation for rigidly-moving parts;
  - Sink/Trim modules for free surface hydrodynamics; and
- Diagnostics:
  - Shock surfaces;
  - Vortex cores.

## 2. GRID GENERATION

## 2.1 Surfaces Defined by Discrete Data

In many instances, the computational domain is not defined by analytical surfaces (such as NURBS or Coon's patches), but by triangulations, i.e. discrete data. Examples where this is the case are:

- Offset data for ships [Yan00];
- Surfaces given by a NASTRAN, ANSYS, DYNA3D, or other CSD file;
- Geometric input data from imaging, such as CAT, MRA or PET scans for patient-specific hemodynamic simulations [Ceb99, Ceb00], or remote sensing input for climate, groundwater and seepage modelling;
- Multi-disciplinary problems, e.g. flow domain remeshing due to topology changes in the structure due to rupture or spallation [Löh99].

An advancing front technique for meshing discrete surfaces was first presented in [Löh96]. In principle, the technique follows the same steps as the classic advancing front procedure. The side forming the smallest new surface triangle is removed from the active

front. New points are placed on the discrete surface using interpolation. While this procedure works well for smooth surfaces, extensive research efforts have been devoted in order to make it able to deal with ‘dirty’ discrete surfaces. The following techniques were found to yield significant improvements in reliability and speed [Löh00]:

- 2-D crossing checks that avoid the ambiguities presented by 3-D crossing checks;
- Use of point and side normals to filter sides at ridges;
- Filtering of unallowed sides based on angles of visibility;
- Orientation checks to obtain the proper host face at ridges;
- Reachability criteria for close points and sides to distinguish topologically inconsistent points; and
- Octree-based search of host faces when near-neighbour search fails.

Figure 1 shows an MRA-based surface triangulation of cerebral arteries used for hemodynamic simulations [Ceb00]. The surface consisted of approximately 200 discrete surface patches with a total of 31,329 triangles. The background grid was adaptively refined to 6 levels, yielding approximately 60,000 elements. The generated surface mesh, shown in Figure 14, had approximately 61,800 triangles, and the volume mesh 826,701 tetrahedra. Without the improvements mentioned above, this case could not have been gridded with such a coarse surface mesh, rendering the ensuing simulations impractical due to high CPU requirements.

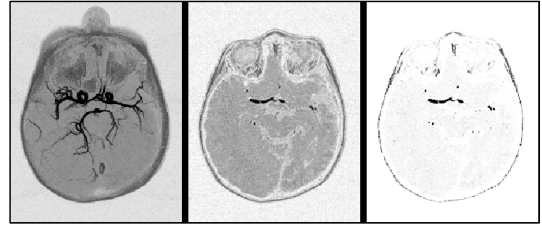


Figure 1a Cerebral Arteries: MRA Data



Figure 1b Cerebral Arteries: Discrete Surface Definition

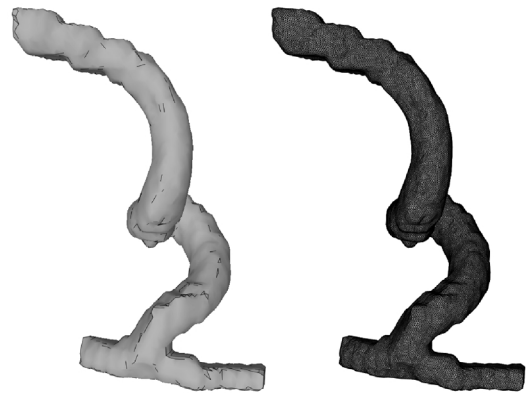


Figure 1c Cerebral Arteries: Surface Mesh

## 2.2 Improvements in Parallel Grid Generation

During the past two years, considerable effort was devoted to parallelize the advancing front grid generator that is used within FEFLO to generate initial grids and to regenerate grids globally or locally for moving/deforming body applications [Löh00]. The main improvements over the last year were:

- More reliable workload estimation for spatial domains;

- Introduction of a limit for the number of elements a processor may generate;
- Reduction of processors used if the number of remaining faces is too small; and
- Reduction of processors used in the post-generation/improvement phase.

Figure 2 shows timings for a pilot ejection case, a typical problem with moving bodies that requires several remeshings during the course of a simulation [Bau93, Bau95, Bau97, Sha00]. The outline of the domain is shown in Figure 2a.

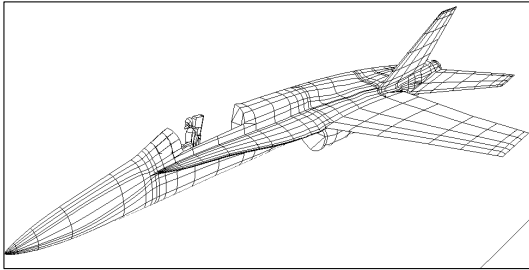


Figure 2a F18: Outline of Domain

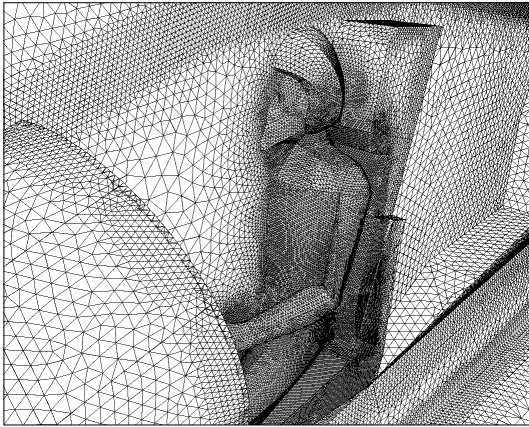


Figure 2b F18: Surface Mesh (Closeup)

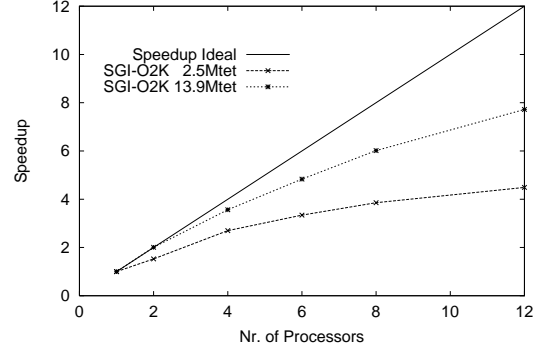


Figure 2c F18: Speedups Obtained

The surface triangulation of the final mesh, which had approximately 14 million tetrahedra, is shown in Figure 2b. The smallest and largest specified element side lengths were 0.65 cm and 250.00 cm respectively, i.e. an edge-length ratio of approximately  $1 : 4 \cdot 10^2$  and a volume ratio of  $1 : 5.6 \cdot 10^7$ . The spatial variation of element size was specified via approximately 110 sources [Löh96]. The speedup obtained for two different grid sizes is displayed in Figure 2c. One can see that the grid generator scales well with the number of processors and that scalability improves with the amount of work required.

### 3. INCOMPRESSIBLE FLOW SOLVERS

#### 3.1 Projection Scheme With Implicit Advection

To date, the advective terms present in the Navier-Stokes equations have been integrated explicitly [Mar92, Ram93, Ram96, Löh99, Ceb00]. This yields highly accurate results, but also imposes severe restrictions on the allowable timestep. For grids that exhibit a large variation in element size the timestep imposed by the smallest elements may be orders of magnitude smaller than the timestep required to obtain time-accurate results. This implies that for many classes of problems, e.g. heat release with buoyancy effects, tens of thousands of timesteps are required per simulation, rendering the scheme impractical. This prompted the quest for an implicit integration of the advective terms that could be incorporated easily into the existing, mature, explicit-advection projection scheme.

Using the notation:

$$u^\ominus = (1 - \Theta)u^n + \Theta u^{n+1} \quad , \quad (1)$$

which implies:

$$u^{n+1} - u^n = \frac{u^\ominus - u^n}{\Theta} \quad , \quad (2)$$

an implicit timestepping scheme can be formulated for the incompressible Navier-Stokes equations. Following similar approaches for compressible flow solvers ([Jam90], [Alo95]), this system can be interpreted as the steady-state solution of the following pseudo-time system:

$$\nabla \cdot \mathbf{v}^\ominus = 0 \quad , \quad (3a)$$

$$\rho \mathbf{v}_{,\tau}^\ominus + \rho \mathbf{v}^\ominus \nabla \mathbf{v}^\ominus + \nabla p^\ominus = \nabla \mu \nabla \mathbf{v}^\ominus + \rho \mathbf{g} - \beta \rho \mathbf{g} T^\ominus + \mathbf{s}_v^\ominus - \rho \frac{\mathbf{v}^\ominus - \mathbf{v}^n}{\theta \Delta t} \quad , \quad (3b)$$

$$\rho c_p T_{,\tau}^\ominus + \rho c_p \mathbf{v}^\ominus \nabla T^\ominus = \nabla k \nabla T^\ominus + \mathbf{s}_T^\ominus - \rho c_p \frac{T^\ominus - T^n}{\theta \Delta t} \quad . \quad (3c)$$

Here  $\rho, \mathbf{v}, p, T, \mu, \beta, k, c_p, \mathbf{s}_v$  and  $\mathbf{s}_T$  denote, respectively, the density, velocity, pressure, temperature, viscosity, expansion coefficient, thermal conductivity, specific heat capacitance, momentum and energy sources.  $\theta$  and  $\tau$  denote the degree of implicitness ( $\theta = 0$ : explicit,  $\theta = 1$ : fully implicit,  $\theta = 0.5$ : Crank-Nicholson) and the pseudo-timestep respectively. Observe that the only difference between Eqns.(3a-c) and the incompressible Navier-Stokes equations is the appearance of new source-terms. These source terms are pointwise dependent on the variables being integrated ( $\mathbf{v}, T$ ), and can therefore be folded into the left hand side of the advective-diffusive or temperature update without any difficulty. The idea is then to march Eqns.(3a-c) to steady state in the pseudo-time  $\tau$  using the original, explicit-advection projection scheme using and local timesteps. The possible gains as compared to the original explicit-advection scheme will depend strongly on the particular physics and mesh of a problem. These gains will be more pronounced the higher the ratio of local  $\tau$ - to global  $t$ -timesteps is, and the higher the variation of the local  $\tau$ -timestep is over the mesh. For very uniform grids, and a low ratio of  $\Delta t : \Delta \tau$  (e.g. detailed LES calculations), it may be advisable to revert back to the explicit-advection scheme.

We include two examples that demonstrate the gains achievable with the implicit-advection projection modules. The first is the well known vonKarman vortex street [Sch79]. The mesh had approximately 60,000 points and 300,000 elements, with special placement of points in the vicinity of the cylinder, was used for the simulation. The parameters were chosen such that the resulting Reynolds number was  $Re = 190$ . Figures 3a,b show the surface grid and the absolute value of the velocity in a cut plane.

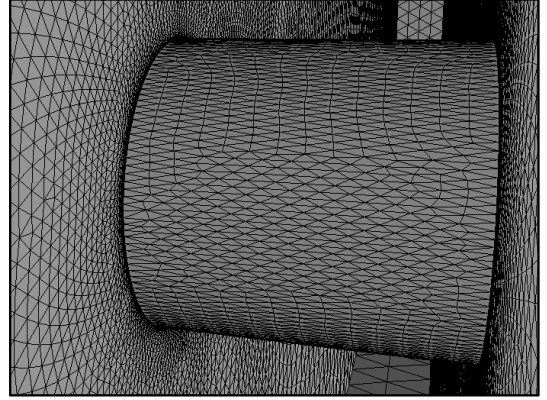


Figure 3a von Karman Vortex Street: Surface Mesh

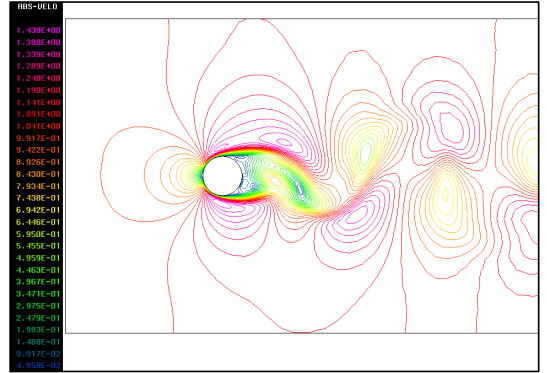


Figure 3b von Karman Vortex Street: Abs(Velocity) in Plane

The lift of the cylinder as a function of time for the original explicit-advection projection scheme, as well as the implicit-advection with implicitness parameters  $\Theta = 0.50, 0.60, 1.00$  is displayed in Figure 3c. Observe that the Strouhal number obtained is approximately  $S = 0.2$  for all cases, in good agreement with experiments [Sch79]. However, the lift changes markedly. Choosing  $\Theta = 1.0$  results in a lift force that is only 50% that of the correct solution. For  $\Theta = 0.50$  some oscillations appear. This is to be expected, as this is the limit of unconditionally stable schemes. In practice, we use  $\Theta = 0.52 - 0.55$ . The ratio of timestep size between the explicit and implicit schemes was approximately 1:100, resulting in overall saving of the order of 1:10.

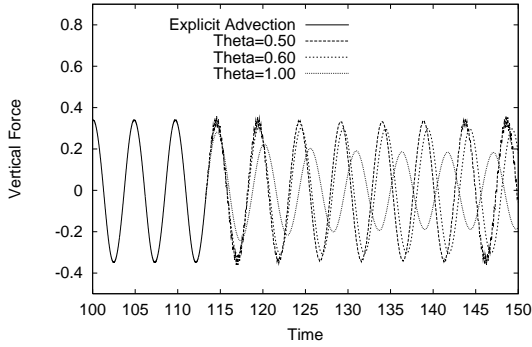


Figure 3c von Karman Vortex Street: Vertical Forces

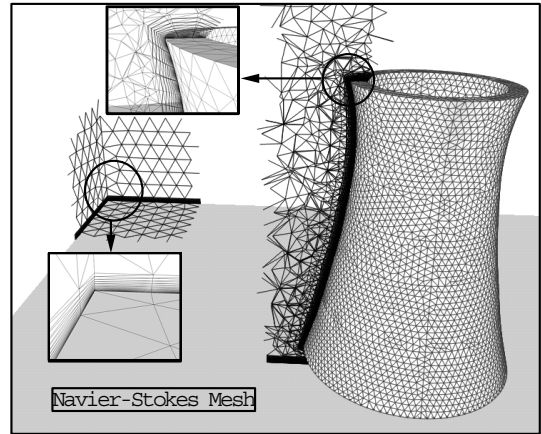


Figure 4b Cooling Tower: Mesh Used

The second example is more of industrial nature. It consist of a cooling tower, where buoyancy effects are important. The geometry of the problem, as well as the boundary conditions, are illustrated in Figure 4a. The Smagorinsky model was used for modeling turbulence effects. The mesh had approximately 300,000 points and 1,200,000 elements, with special placement of points in the vicinity of the tower to capture the separations. Figures 4b-4e show the grid, temperatures and velocities near the tower. For this time-accurate run, the implicit scheme runs approximately 10 times faster an the explicit scheme.

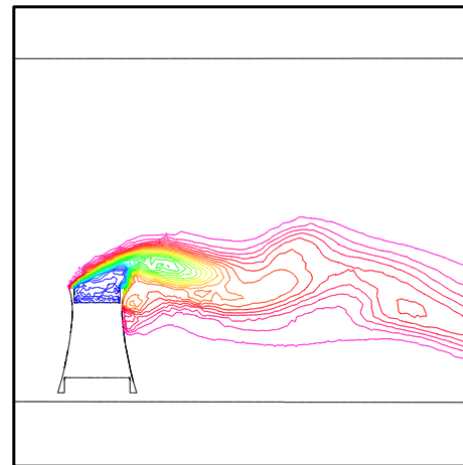


Figure 4c Cooling Tower: Temperature in Mid-Plane

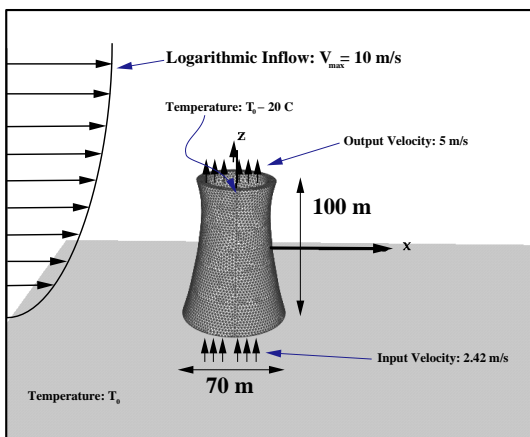


Figure 4a Cooling Tower: Boundary Conditions

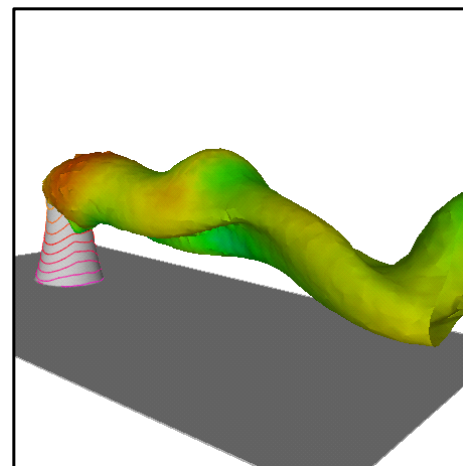


Figure 4d Cooling Tower: Iso-Surface of Temperature

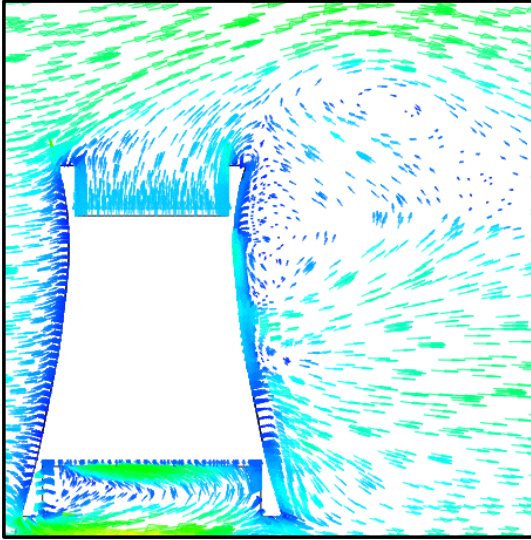


Figure 4e Cooling Tower: Velocities in Mid-Plane

#### 4.2 Pressure Boundary Conditions

For external aerodynamics, the imposition of pressure at outflow boundaries presents no further difficulties. The situation is very different for internal flows, such as pipe networks or arteries in the human body. In these cases, the pressure or the pressure gradient will depend on the outflow flux  $q$ , i.e.

$$p = p_0 + R \cdot (q_n - q_0) \quad , \quad (4)$$

or

$$p_{,n} = g_0 + R' \cdot q_n \quad , \quad (5)$$

where  $R, R'$  are resistance coefficients and  $n$  the outward normal. For a rational determination of  $R, R'$ , see [Ceb00]. Both of these boundary conditions were implemented in FEFLO. Figure 5, taken from [Ceb00], shows the results for the Circle of Willis in the brain, where these boundary conditions are the only way to simulate with some degree of realism the complex hemodynamic patterns that arise. A comparison of average volume flows in different parts of the arterial branches for normal flow and clipped flow is shown in Figure 5c.

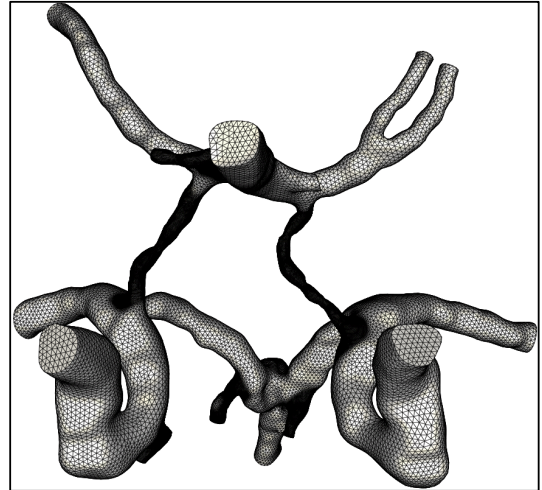


Figure 5a Circle of Willis: Surface Mesh

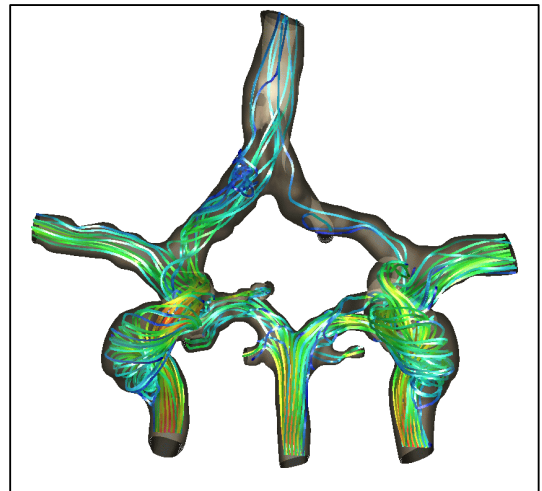


Figure 5b Circle of Willis: Instantaneous Streamlines

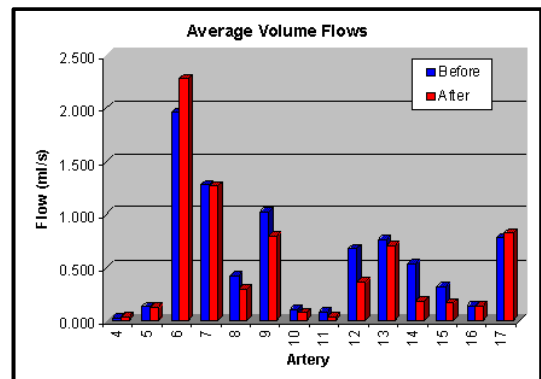


Figure 5c Circle of Willis: Average Volume Flows

### 4.3 Arbitrary Number of Advected Species

In order to simulate the effects of contaminants, combustion and for diagnostics purposes, the option of advecting an arbitrary number of species was added. The equations solved are of the form:

$$u_{,t} + \mathbf{v} \cdot \nabla u = \nabla k \nabla u + S \quad . \quad (6)$$

Both FEM-FCT [Löh87] and upwind [Luo94] solvers were implemented. Figure 6 shows the advection of a marker for the same von Karman cylinder case as shown in Figure 3. The integration of the marker was carried out using the FEM-FCT option. Note the sharpness of the resolution, enabling detailed mixing diagnostics.

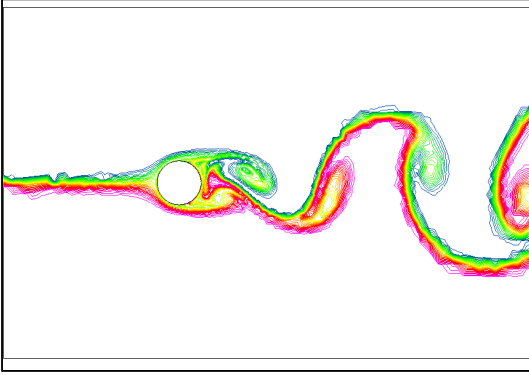


Figure 6 Vortex Street: Passive Scalar

## 4. COMPRESSIBLE FLOW SOLVERS

If we consider a steady flow problem, the fastest solvers to date include multigrid and GMRES-LU-SGS solvers. For supersonic flows, space-marching and blocking represent an interesting alternative [Löh98]. During the past year, the GMRES-LU-SGS solvers developed in prototype versions of FEFLO [Luo94, Luo98, Luo99, Luo00] were migrated into the production code.

The spatial discretization of the compressible Euler equations yields the following system of ODEs:

$$\mathbf{M}_i^i \mathbf{u}_{,t}^i = \mathbf{r}^i = \sum \mathbf{C}^{ij} \cdot \mathbf{F}_{ij} \quad , \quad (7)$$

where  $i$  denotes the index of any given point, and the  $j$  the indices of all points surrounding  $i$ . Intergrating in time using a backward Euler scheme, and linearizing with the Jacobians of the fluxes  $\mathbf{A} = \mathbf{F}_{,\mathbf{u}}$ , one obtains

$$\left[ \frac{1}{\Delta t} \mathbf{M}_i^i - \sum \mathbf{C}^{ij} \cdot \mathbf{A}_{ij} \right] \Delta \mathbf{u}^i = \mathbf{r}^i \quad . \quad (8)$$

This may be rewritten as

$$\mathbf{K} \cdot \Delta \mathbf{u} = (\mathbf{L} + \mathbf{D} + \mathbf{U}) \cdot \Delta \mathbf{u} = \mathbf{r} \quad , \quad (9)$$

where  $\mathbf{L}, \mathbf{D}, \mathbf{U}$  denote the lower, diagonal and upper diagonal entries of  $\mathbf{K}$ . Classic relaxation schemes to solve this system of equations include:

a) Gauss-Seidel, given by:

$$\begin{aligned} (\mathbf{L} + \mathbf{D}) \cdot \Delta \mathbf{u}^1 &= \mathbf{r} - \mathbf{U} \cdot \Delta \mathbf{u}^0 \quad , \\ (\mathbf{D} + \mathbf{U}) \cdot \Delta \mathbf{u} &= \mathbf{r} - \mathbf{L} \cdot \Delta \mathbf{u}^1 \quad ; \end{aligned} \quad (10)$$

b) Lower-Upper Symmetric Gauss-Seidel (LU-SGS), given by:

$$(\mathbf{L} + \mathbf{D}) \cdot \mathbf{D}^{-1} \cdot (\mathbf{D} + \mathbf{U}) \cdot \Delta \mathbf{u} = \mathbf{r} \quad . \quad (11)$$

These relaxation schemes have been optimized over the years, resulting in very efficient solvers [Luo98, Luo99, Luo00, Sha00]. Key ideas include:

- Using the spectral radius of  $\mathbf{A}$  for the diagonal entries  $\mathbf{D}$ , resulting in:

$$\mathbf{D} = \left[ \frac{1}{\Delta t} \mathbf{M}_i^i - 0.5 \sum \mathbf{C}^{ij} \rho_A \right] \mathbf{I} \quad ; \quad (12)$$

- Replacing:

$$\mathbf{A} \cdot \Delta \mathbf{u} \approx \Delta \mathbf{F} \quad ,$$

resulting in:

$$\Delta \mathbf{F} = \mathbf{F}(\mathbf{u} + \Delta \mathbf{u}) - \mathbf{F}(\mathbf{u}) \quad . \quad (13)$$

The combined effect of these simplifications is a family of schemes that are matrix free, require no extra storage as compared to explicit schemes, and per relaxation sweep are faster than conventional explicit schemes. For the LU-SGS scheme, each pass over the mesh proceeds as follows:

a) Forward Sweep:

$$\Delta \hat{\mathbf{u}}^i = \mathbf{D}^{-1} \left[ \mathbf{r}^i - 0.5 \sum_{j < i} \mathbf{C}^{ij} \cdot (\Delta \hat{\mathbf{F}}_{ij} - \rho_A \Delta \hat{\mathbf{u}}_j) \right] \quad (14a)$$

b) Backward Sweep:

$$\mathbf{r} = \mathbf{D} \cdot \Delta \hat{\mathbf{u}}$$

$$\Delta \mathbf{u}^i = \mathbf{D}^{-1} \left[ \mathbf{r}^i - 0.5 \sum_{j > i} \mathbf{C}^{ij} \cdot (\Delta \mathbf{F}_{ij} - \rho_A \Delta \mathbf{u}_j) \right] \quad (14b)$$

Luo [Luo98] has shown that no discernable difference could be observed when taking central or upwind discretizations for  $\Delta \mathbf{F}$ . As the CPU requirements of upwind discretizations are much higher, all relaxation

passes are carried out using central schemes. Given that the same loop structure ( $\mathbf{L}$ ,  $\mathbf{D}$ ,  $\mathbf{U}$ ) is required for both the Gauss-Seidel, the LU-SGS and the GMRES matrix-vector products, it is possible to write a single ‘sweep’ subroutine that encompasses all of these cases. The initialization of the Gauss-Seidel loop is accomplished with an LU-SGS pass. Figure 7 shows the convergence results for .....

Figure 7 LUSGS Convergence results

## 5. BODY/MESH MOVEMENT MODULES

The ability to move an arbitrary number of bodies immersed in a flowfield has been used by the authors for over a decade [Löh90, Bau93, Bau94, Bau95, Bau97, Bau99, Löh99, Ram99]. The last year saw a number of important extensions that are summarized below.

### 5.1 Overlapping Grids

A large body of work exists on overlapping grids; in fact, a whole series of conferences is devoted to the subject. In a series of papers, Nakahashi and co-workers [Nak99, Tog00] developed and applied overlapping grids with unstructured grids, showing good performance for the overall scheme. A general ‘distance to wall’ criterion was used to assign which points should be interpolated between grids. For each grid system, the distance to the body walls was computed. Any point from a given grid falling into the element of another grid was interpolated if its distance to wall was larger than that of the points of the other grid. The interpolation information was obtained from a nearest neighbour technique. Given the success of this overlapping grid implementation for the simulation of flows with rigidly moving bodies, similar techniques were incorporated into FEFLO. The key ideas may be summarized as follows (for a detailed description, see [Löh01]):

- For the ‘dominant mesh criterion’ that determines which points interpolate to other points, both the distance to walls and the element size are considered;
- To compute the distance to walls, a fast algorithm based on heap lists and an advancing front concept is used;
- The search for interpolation candidates between grids is carried out using octrees and bounding boxes;
- The algorithms used have the ability to treat bodies that are partially outside grids;
- For applications with moving bodies, incremental interpolation procedures with nearest neighbour search are employed.

The overlapping grid technique has been implemented for both the compressible and incompressible flow modules. Figure 8 shows a generic spinning missile at  $Ma = 2.5$  and  $\alpha = 5^\circ$  degree angle of attack. Note that the shocks neither dissipate nor reflect across the overlap zone.

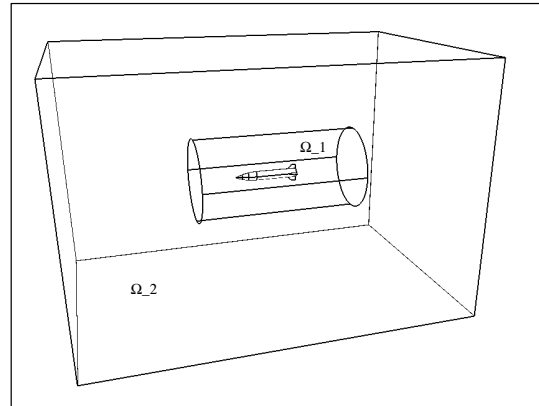


Figure 8.1 Spinning Missile: Computational Domain

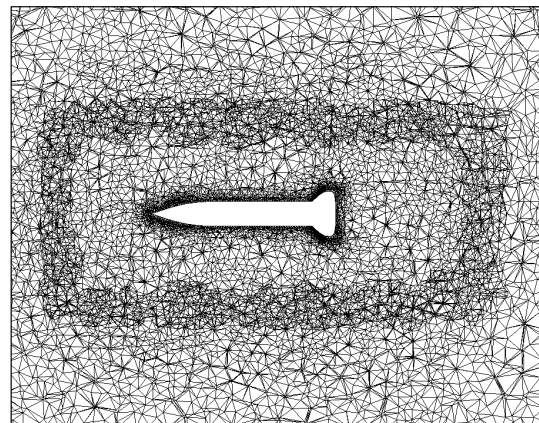


Figure 8.2 Spinning Missile: Plane  $z=0$ : Mesh

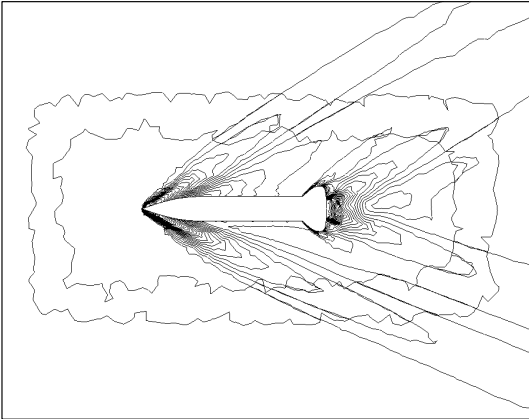


Figure 8.3 Spinning Missile: Plane  $z=0$ : Pressure Contours

5.2 Avoidance of Geometry Recalculation for Rigidly-Moving Parts

Any moving mesh application requires the recalculation of geometry parameters, such as shape-function derivatives, boundary normals, etc. For explicit solvers, the recalculation of these geometry parameters can consume as much CPU time as advancing the solution one timestep. In some cases, the fact that the mesh is deforming only in a small portion of the domain can be used to avoid a complete recalculation of these geometry parameters at every timestep. The idea is to identify the number of elements that actually deform. If the ratio of these elements to the overall mesh is small enough, only the edge-parameters that require change are updated. If not, the geometry parameters are recalculated for the whole mesh. For elements that are rotating rigidly, the face and edge normals are rotated accordingly. This technique has been used for simulations of rotating machinery, and moving body simulations with overlapping grids.

5.3 Mesh Movement from Prescribed CAD Motion

The ability to move the mesh based on a prescribed deforming surface opens the possibility to study actuators, compliant surface devices and animal locomotion in water and air. Such a capability was implemented in FEFLO. The user prescribes with tables the motion of CAD data in time. The code then identifies the parametric coordinates  $\xi, \eta$  of each surface point. While  $\xi, \eta$  are kept constant during the simulation for surface points, their Cartesian coordinates  $x, y, z$  change in time according to the prescribed motion. Figure 9 shows the pressure for a bird wrasse, together with 3 positions of the fins. The fin motion was taken from measurements.

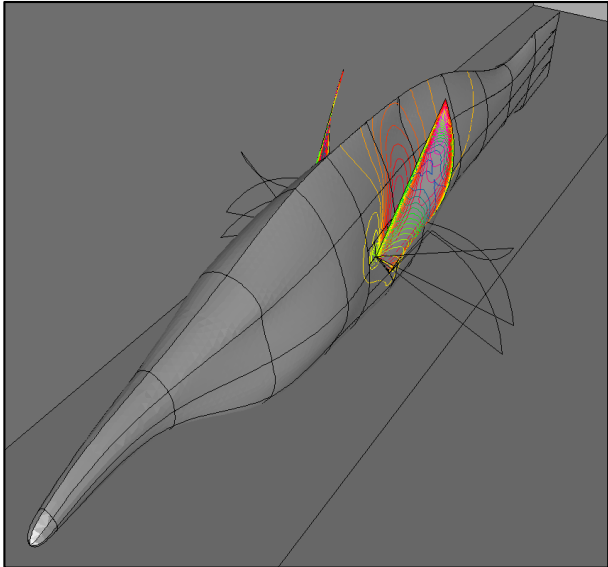


Figure 9 Bird Wrasse: Pressure

Tying the motion of surface points to prescribed CAD-data produces very smooth deforming grids.

5.4 Sink/Trimm Modules for Free Surface Hydrodynamics

Over the last years, FEFLO has been used extensively for the calculation of free surface hydrodynamics.

sound of the gas. The normal vector may be computed in a variety of ways. One option that works well is the absolute value of the velocity vector, i.e.:

$$\mathbf{n} = \frac{\nabla|\mathbf{v}|}{|\nabla|\mathbf{v}||} . \tag{16}$$

A plot of the resulting isosurface reveals, for each shock, the desired shock surface plus close isosurfaces that are false. An explanation for these false shock surfaces can be discerned from Figure 11. As one moves along the path A-B, the normal Mach-number passes twice through the desired value of  $Ma_n = -1$ . The solution is to retain only the isosurface faces that are facing in the direction of the flow, i.e.

$$Ma_{n_f} = \frac{\mathbf{v} \cdot \mathbf{n}_f}{c} > c_f , \tag{17}$$

Figure 10 Wigley Hull: Sink/Trimm Results

6. DIAGNOSTICS

Given that most of the large production runs are carried out remotely and are of transient nature, all of the diagnostics that a CFD code can produce must be obtained remotely. In this way, only the extracted information, which is only a very small fraction of a typical restart dump, is transferred back for display purposes. Over the last year, two important diagnostics were added to FEFLO: shock surfaces and vortex core identification [Ceb01].

6.1 Shock Surfaces

For many complex flowfields, the visualization of shock surfaces can yield useful insight into relevant physics [Lov99]. Shock surfaces could also be used as effective adaptive refinement indicators. The shock surface detection technique implemented in FEFLO to date is only applicable to steady flows. At a steady shock, the Mach-number of the normal velocity must be unity and decrease in the direction of the flow:

$$Ma_n = \frac{|v_n|}{c} = 1 , \quad v_n = \mathbf{v} \cdot \mathbf{n} , \quad c = \sqrt{\frac{\gamma p}{\rho}} , \tag{15}$$

where  $Ma$ ,  $\mathbf{n}$ ,  $\mathbf{v}$ ,  $\gamma$ ,  $p$ ,  $\rho$  and  $c$  denote, respectively, the Mach-number, unit normal vector, velocity vector, ratio of specific heats, pressure, density and speed of

where  $n_f$  denotes the unit normal of the isosurface face and  $c_f = O(0.4)$ .

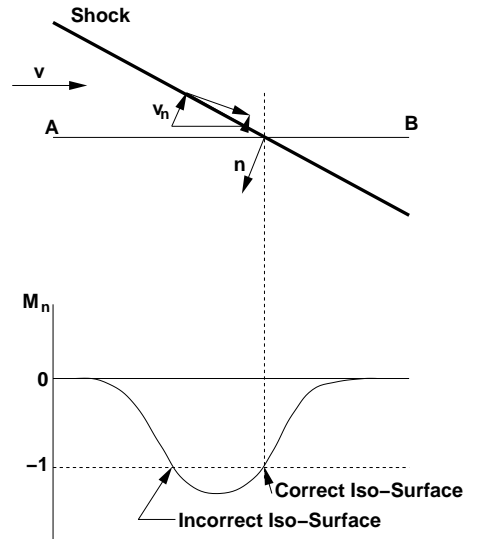


Figure 11 Identifying The Correct Shock Surface

Figure 12, taken from [Ceb01], shows the shock surfaces for a supersonic inlet problem.

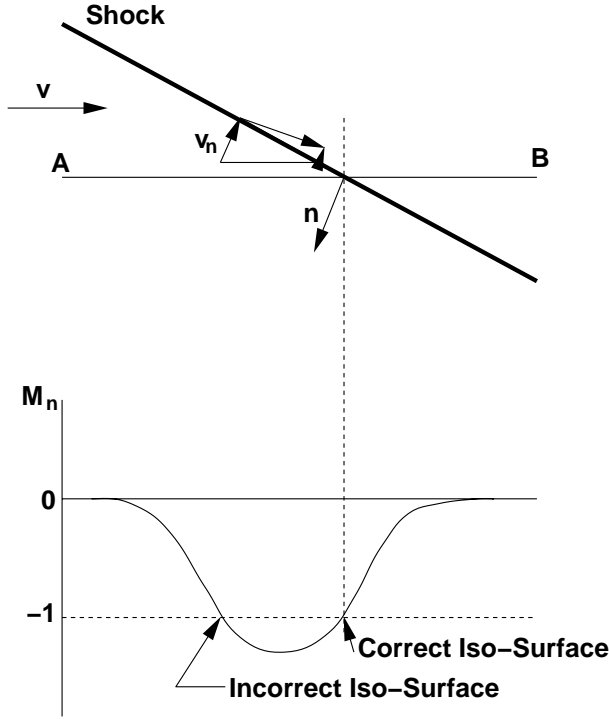


Figure 11 Identifying The Correct Shock Surface

## 6.2 Vortex Core Identification

Vortex centerlines are useful to visualize complex flowfields [Saw95, Suj95, Hai99], and have also been used as an effective adaptive refinement indicator [Mur00]. A point on the vortex centerline is defined as the location where the velocity in the plane normal to the vorticity vanishes. In other words, given a measurable vorticity vector  $\omega$ :

$$\omega = \nabla \times \mathbf{v} \quad , \quad (18)$$

and the velocity normal to it:

$$\mathbf{v}' = \mathbf{v} - \frac{(\mathbf{v} \cdot \omega)\omega}{|\omega|^2} \quad , \quad (19)$$

a point on the vortex centerline is given by:

$$\mathbf{v}' = 0 \quad . \quad (20)$$

Determining whether a vortex centerline lies in an element is particularly simple for linear tetrahedral elements. In this case, the vorticity is constant, and the projected values  $\mathbf{v}'_i$  for the plane normal to  $\omega$  can be written as:

$$\mathbf{v}' = \mathbf{v}'_1 + [\mathbf{v}'_{21}, \mathbf{v}'_{31}, \mathbf{v}'_{41}] \cdot \xi = \mathbf{v}'_1 + \mathbf{V} \cdot \xi \quad . \quad (21)$$

with  $\xi = (\xi, \eta, \zeta)$  and  $\mathbf{v}'_{ij} = \mathbf{v}'_i - \mathbf{v}'_j$ . The vortex centerline **point** for the element is then given in local coordinates by:

$$\xi_0 = -\mathbf{V}^{-1} \cdot \mathbf{v}'_1 \quad . \quad (22)$$

An element is marked as being on a vortex centerline if

$$0 \leq \xi, \eta, \zeta, 1 - \xi - \eta - \zeta \leq 1 \quad . \quad (23)$$

For graphical display purposes, the intersection of the **line** given by  $\xi_0$  and the direction of the vorticity vector with the faces of the element is evaluated. This will lead to a discontinuous vortex core line, but leads to acceptable displays. The present vortex core identification technique may also be used as an error indicator [Mur00]. The elements that contain a vortex filament are marked for refinement, together with layers of elements surrounding them. Figure 10 shows the incompressible flow past a NACA0012 wing. Three levels of refinement, were used to capture the vortex. The vortex core obtained using the present procedure is clearly visible.

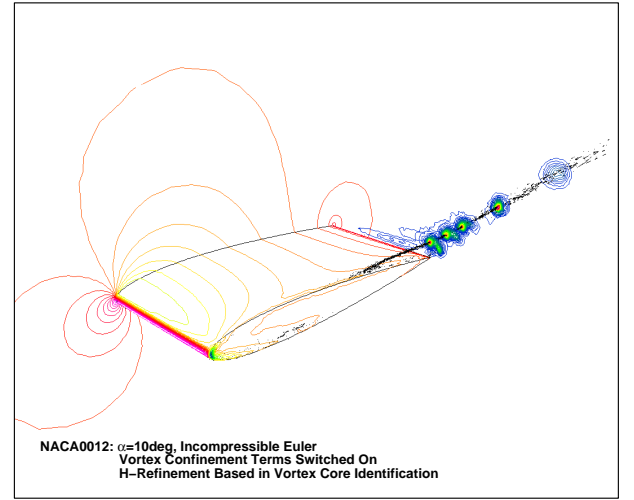


Figure 10 NACA0012: Vortex Core Visualization

## 7. CONCLUSIONS AND OUTLOOK

The year 2000 has seen a number of important developments and improvements for FEFLO. Notable among these are:

- Advances in the gridding of surfaces given as discrete data;
- Improvements in parallel grid generation;
- A new implicit-advection projection solver;
- Additional boundary conditions for internal flows;
- The ability to advect/diffuse an arbitrary number of species;

- Two matrix-free implicit compressible solvers;
- Overlapping grids (unstructured chimera);
- The avoidance of geometry recalculation for rigidly-moving parts;
- Sink/trimm modules for free surface hydrodynamics; and
- Shock surface and vortex core identification modules.

Planned future developments include:

- Improved Navier-Stokes gridding;
- Arbitrary-gas equation of state lookup;
- Combustion modules to simulate chemically reacting flows;
- Inclusion of some form of vorticity confinement to track vortices over large distances; and
- Local remeshing for simulations with topology change.

We are also actively optimizing FEFLO for vector supercomputers. This perhaps surprising move is prompted by the observation that it takes 50-100 SGI Origin processors to achieve the same MFLOP-rate as one processor on the NEC-SX5. It appears that every human endeavour has its inherent Moore's law. It not only applies to personal computers but also to vector-machines. The constant may be lower (even this is not clear at present), but the numbers are compelling.

## 8. ACKNOWLEDGEMENTS

This work was partially supported by AFOSR, DTRA and SAIC IR&D. The parallelization of FEFLO was partially supported by the DoD HPC Modernization Office under the CHSSI FEFLO project.

## 9. REFERENCES

- [Alo95] J. Alonso, L. Martinelli and A. Jameson - Multigrid Unsteady Navier-Stokes Calculations with Aeroelastic Applications; *AIAA-95-0048* (1995).
- [Bau91] J.D. Baum and R. Löhner - Numerical Simulation of Shock Interaction with a Modern Main Battlefield Tank; *AIAA-91-1666* (1991).
- [Bau93] J.D. Baum and R. Löhner - Numerical Simulation of Pilot/Seat Ejection from an F-16; *AIAA-93-0783* (1993).
- [Bau93] J.D. Baum, H. Luo and R. Löhner - Numerical Simulation of a Blast Inside a Boeing 747; *AIAA-93-3091* (1993).
- [Bau94] J.D. Baum, H. Luo and R. Löhner - A New ALE Adaptive Unstructured Methodology for the Simulation of Moving Bodies; *AIAA-94-0414* (1994).
- [Bau95] J.D. Baum, H. Luo and R. Löhner - Numerical Simulation of Blast in the World Trade Center; *AIAA-95-0085* (1995).
- [Bau95] J.D. Baum, H. Luo and R. Löhner - Validation of a New ALE, Adaptive Unstructured Moving Body Methodology for Multi-Store Ejection Simulations; *AIAA-95-1792* (1995).
- [Bau96] J.D. Baum, H. Luo, R. Löhner, C. Yang, D. Pelessone and C. Charman - A Coupled Fluid/Structure Modeling of Shock Interaction with a Truck; *AIAA-96-0795* (1996).
- [Bau97] J.D. Baum, H. Luo, R. Löhner, E. Goldberg and A. Feldhun - Application of Unstructured Adaptive Moving Body Methodology to the Simulation of Fuel Tank Separation From an F-16 C/D Fighter; *AIAA-97-0166* (1997).
- [Bau97] J.D. Baum, R. Löhner, T.J. Marquette and H. Luo - Numerical Simulation of Aircraft Canopy Trajectory; *AIAA-97-1885* (1997).
- [Bau99] J.D. Baum, H. Luo, E. Mestreau, R. Löhner, D. Pelessone and C. Charman - A Coupled CFD/CSD Methodology for Modeling Weapon Detonation and Fragmentation; *AIAA-99-0794* (1999).
- [Ceb99] J. Cebal and R. Löhner - From Medical Images to CFD Meshes; *Proc. 8th Int. Meshing Roundtable*, South Lake Tahoe, October (1999).
- [Ceb00] J.R. Cebal and R. Löhner - Automatic Grid Generation for Anatomically Accurate Computational Hemodynamics Calculations; *Proc. ICMMB-11*, April 2-5, Hawaii (2000).
- [Ceb00] J.R. Cebal and R. Löhner - Image-Based Computational Hemodynamics; *Proc. World Congress in Medical Physics and Biomedical Engineering*, Chicago, Illinois, July 23-28 (2000).
- [Ceb00] J.R. Cebal, R. Löhner and J. Burgess - Computer Simulation of Cerebral Artery Clipping: Relevance to Aneurysm Neuro-Surgery Planning; *Proc. of ECCOMAS 2000 Conf.*, Barcelona, Spain, September (2000).
- [Ceb01] J.R. Cebal and R. Löhner - Flow Visualization On Unstructured Grids Using Geometrical Cuts, Vortex Detection and Shock Surfaces; *AIAA-01-0915* (2001).
- [Hai99] R. Haimes and D. Kenwright - On The Velocity Gradient Tensor and Fluid Feature Extraction; *AIAA-99-3288-CP* (1999).
- [Löh87] R. Löhner, K. Morgan, J. Peraire and M. Vahdati - Finite Element Flux-Corrected Transport (FEM-FCT) for the Euler and Navier-Stokes Equations; ICASE Rep. 87-4, *Int. J. Num. Meth. Fluids* 7, 1093-1109 (1987).
- [Löh90] R. Löhner - Three-Dimensional Fluid-Structure Interaction Using a Finite Element Solver and Adaptive Remeshing; *Computer Systems in Engineering* 1, 2-4, 257-272 (1990).

- [Löh95] R. Löhner and R. Ramamurti - A Load Balancing Algorithm for Unstructured Grids; *Comp. Fluid Dyn.* 5, 39-58 (1995).
- [Löh96] R. Löhner - Regridding Surface Triangulations; *J. Comp. Phys.* 126, 1-10 (1996).
- [Löh96] R. Löhner - Extensions and Improvements of the Advancing Front Grid Generation Technique; *Comm. Num. Meth. Eng.* 12, 683-702 (1996).
- [Löh98] R. Löhner - Renumbering Strategies for Unstructured-Grid Solvers Operating on Shared-Memory, Cache-Based Parallel Machines; *Comp. Meth. Appl. Mech. Eng.* 163, 95-109 (1998).
- [Löh98] R. Löhner - Computational Aspects of Space-Marching; *AIAA-98-0617* (1998).
- [Löh99] R. Löhner, Chi Yang, E. Oñate and S. Idelsohn - An Unstructured Grid-Based, Parallel Free Surface Solver; *Appl. Num. Math.* 31, 271-293 (1999).
- [Löh99] R. Löhner, Chi Yang, J.D. Baum, H. Luo, D. Pelessone and C. Charman - The Numerical Simulation of Strongly Unsteady Flows With Hundreds of Moving Bodies; *Int. J. Num. Meth. Fluids* 31, 113-120 (1999).
- [Löh99] R. Löhner, C. Yang, J. Cebal, J.D. Baum, H. Luo, E. Mestreau, D. Pelessone and C. Charman - Fluid-Structure Interaction Algorithms for Rupture and Topology Change; *Proc. 1999 JSME Computational Mechanics Division Meeting*, Matsuyama, Japan, November (1999).
- [Löh00] R. Löhner - A Parallel Advancing Front Grid Generation Scheme; *AIAA-00-1005* (2000).
- [Löh00] R. Löhner, O. Soto and F. Camelli - On Implicit Projection Schemes; *4th Annual GMU Transport and Dispersion Modeling Workshop*, Fairfax, VA, July (2000).
- [Löh00] R. Löhner - Advances in Unstructured Grid Generation; *Proc. of ECCOMAS 2000 Conf.*, Barcelona, Spain, September (2000).
- [Löh01] R. Löhner, D. Sharov, H. Luo and R. Ramamurti - Overlapping Unstructured Grids; *AIAA-01-0439* (2001).
- [Lov99] D. Lovely and R. Haimes - Shock Detection from Computational Fluid Dynamics Results; *AIAA-99-3285-CP* (1999).
- [Luo94] H. Luo, J.D. Baum and R. Löhner - Edge-Based Finite Element Scheme for the Euler Equations; *AIAA J.* 32, 6, 1183-1190 (1994).
- [Luo94] H. Luo, J.D. Baum, R. Löhner and J. Cabello - Implicit Finite Element Schemes and Boundary Conditions for Compressible Flows on Unstructured Grids; *AIAA-94-0816* (1994).
- [Luo98] H. Luo, J.D. Baum and R. Löhner - A Fast, Matrix-Free Implicit Method for Compressible Flows on Unstructured Grids; *J. Comp. Phys.* 146, 664-690 (1998).
- [Luo99] H. Luo, J.D. Baum and R. Löhner - An Accurate, Fast, Matrix-Free Implicit Method for Computing Unsteady Flows on Unstructured Grids; *AIAA-99-0937* (1999).
- [Luo00] H. Luo, D. Sharov, J.D. Baum and R. Löhner - A Class of Matrix-free Implicit Methods for Compressible Flows on Unstructured Grids; *First International Conference on Computational Fluid Dynamics*, Kyoto, Japan, July 10-14 (2000).
- [Mar92] D. Martin and R. Löhner - An Implicit Linelet-Based Solver for Incompressible Flows; *AIAA-92-0668* (1992).
- [Mur00] M. Murayama, K. Nakahashi and K. Sawada - Numerical Simulation of Vortex Breakdown Using Adaptive Grid Refinement With Vortex-Center Identification; *AIAA-00-0806* (2000).
- [Nak99] K. Nakahashi, F. Togashi and D. Sharov - An Intergrid-Boundary Definition Method for Over-set Unstructured Grid Approach; *AIAA-99-3304-CP* (1999); see also *AIAA J.* 38, 11, 2077-2084 (2000).
- [Ram93] R. Ramamurti and R. Löhner - Simulation of Flow Past Complex Geometries Using a Parallel Implicit Incompressible Flow Solver; pp. 1049,1050 in *Proc. 11th AIAA CFD Conf.*, Orlando, FL, July (1993).
- [Ram96] R. Ramamurti and R. Löhner - A Parallel Implicit Incompressible Flow Solver Using Unstructured Meshes; *Computers and Fluids* 25, 2, 119-132 (1996).
- [Ram99] R. Ramamurti, W.C. Sandberg and R. Löhner - Computation of Unsteady Flow Past Deforming Geometries; *Int. J. Comp. Fluid Dyn.*, 83-99 (1999).
- [Saw95] K. Sawada - A Convenient Visualization Method for Identifying Vortex Centers; *Transactions of the Japan Society for Aeronautical and Space Sciences* 38, 120 (1995).
- [Sch79] H. Schlichting - *Boundary Layer Theory*; McGraw-Hill (1979).
- [Sha00] D. Sharov, H. Luo, J.D. Baum and R. Löhner - Implementation of Unstructured Grid GMRES+LU-SGS Method on Shared-Memory, Cache-Based Parallel Computers; *AIAA-00-0927* (2000).
- [Sha00] D. Sharov, H. Luo, J.D. Baum and R. Löhner - Time-Accurate Implicit ALE Algorithm for Shared-Memory Parallel Computers; *First International Conference on Computational Fluid Dynamics*, Kyoto, Japan, July 10-14 (2000).

[Suj95] D. Sujudi and - Identification of Swirling Flow in 3-D Vector Fields; *AIAA-95-1715* (1995).

[Tog00] F. Togashi, K. Nakahashi and Y. Ito - Flow Simulation of NAL Experimental Supersonic Airplane/Booster Separation Using Overset Unstructured Grids; *AIAA-00-1007* (2000).

[Tus98] J. Tuszynski and R. Löhner - Parallelizing the Construction of Indirect Access Arrays for Shared-Memory Machines; *Comm. Appl. Num. Meth. Eng.* 14, 773-781 (1998).

[Yan00] Chi Yang, R. Löhner, F. Noblesse and T.T. Huang - Calculation of Ship Sinkage and Trim Using Unstructured Grids; *Proc. of ECCOMAS 2000 Conf.*, Barcelona, Spain, September (2000).