JOURNAL OF AIRCRAFT
Vol. 42, No. 4, July–August 2005

A General Purpose Euler and Navier–Stokes Solver for Structured and Unstructured Grids

M. Mani* and A. Cary†
The Boeing Company, St. Louis, Missouri 63166
and
S. V. Ramakrishnan‡
Hypercomp, Inc., Westlake Village, California 91362

An implicit, multizone, finite volume three-dimensional Euler and Navier–Stokes solver known as Boeing—computational-fluid-dynamics code has been developed for hybrid grids (structured, unstructured, or combinations). The code contains multispecies capabilities, different numerical algorithms, and turbulence models. It has been validated for subsonic through hypersonic flow regimes.

Introduction

COMPUTATIONAL-FLUID-DYNAMICS (CFD) algorithms and computer resources have matured to a level that CFD can be used in the aerospace design process as opposed to just for risk reduction and understanding of flow physics. However, the traditional approach with structured grid technology is often prohibitive of reaching the cycle time required (100x improvement) for CFD to be a design tool. The most time-consuming element in the application cycle time is the grid-generation process. To overcome this obstacle and make CFD available early in the design cycle, we have decided to move to the unstructured grid technology. In this manuscript we will discuss the development of the Navier–Stokes solver BCFD (Boeing-CFD) and show several validation cases.

The development of the unstructured grid technology at Boeing began in 1999. Since then, we have developed a process for faster turnaround of the CFD solutions. The grid-generation process has substantially reduced the CFD cycle time. To maintain our existing CFD capability and build the unstructured grid upon the existing technology, we decided to have both structured and unstructured grid capabilities within a single code, BCFD.

Since 1999, we have achieved over a magnitude reduction in our CFD cycle time. Although this is a substantial achievement, we believe it is possible to reduce the cycle time by yet another order of magnitude. We have reduced the cycle times from weeks to days; the goal is to obtain a Navier–Stokes solution within a few hours.

Navier–Stokes Solver (BCFD Code)

The unstructured-grid solver in BCFD solves the unsteady reynolds-averaged Navier–Stokes equations in a multizone hybrid-grid framework. Physical modeling capabilities of the hybrid-grid solver include one-equation pointwise and Spalart–Allmaras (S-A) turbulence models and multispecies flows with or without chemical reactions. BCFD employs a total-variation-diminishing (TVD)-based[1] second-order upwind finite volume scheme with backward Euler in time to discretize the governing RANS equations. Upwinding is achieved through the use of Roe's scheme[2] or a modified Rusanov's scheme.[3] The resulting algebraic equations are solved using a point-implicit scheme.

In BCFD, the structured and unstructured-grid solvers employ the same data structure for user input. English text is used for input specifications, and all capabilities have a default setting. BCFD is capable of solving URANS using a multizone mixed grid in the sense that the grid can consist of some structured-grid zones and some unstructured-grid zones. It can include unaligned zonal boundaries. Thus, BCFD offers the flexibility of employing the most appropriate grid topology for a region in the computational domain. For instance, the structured grid (hexahedral) is the most appropriate grid topology for viscous regions that involve boundaries with large curvatures, like the leading edge of a wing. Ability to employ the most appropriate grid topology for a region in the computational domain enables optimum distribution of grid cells.

The time-marching scheme employed in BCFD represents a compromise between computer memory and CPU time required for large-scale computations. Although the point-implicit scheme employed in BCFD is not the best scheme from the point of view of convergence, it has enabled the developers of BCFD to offer the users an unstructured-grid CFD code that uses only about 80 words of memory per grid cell. In spite of the fact that the cost and availability of computer memory have improved enormously in the last few years, considering the fact that currently 20–30 million grid-cell computations are performed routinely, a code that requires less than 1 kB of memory per grid cell makes it possible to perform such computations using available computing resources. The BCFD code is integrated into the CFD process from geometry acquisition to engineering results.

BCFD solves the unsteady, Reynolds-averaged, compressible, Navier–Stokes (RANS) equations on grids with arbitrary cell shapes. Parallel processing is achieved using message-passing interface/parallel virtual machine (PVM) and a master-slave paradigm. Flows involving equilibrium-air, frozen, and finite-rate chemistry can be simulated using BCFD. Ability to handle unaligned boundaries makes BCFD a powerful analysis tool.

A finite volume scheme employs the integral form of the governing equations for a control volume (grid cell) given by

$$V\frac{\partial \bar{\boldsymbol{q}}}{\partial t} + \iint\limits_{A} (\boldsymbol{F} + \boldsymbol{G})\,\mathrm{d}\boldsymbol{a} = \boldsymbol{s} \tag{1}$$

where t is time, V is the cell volume, A the cell boundary, $\boldsymbol{a}$ the area normal vector, and $\boldsymbol{s}$ is a source term vector. The vector of dependent variables $\bar{\boldsymbol{q}}$, the inviscid flux tensor $\boldsymbol{F} = (\boldsymbol{f}_1, \boldsymbol{f}_2, \boldsymbol{f}_3)$, and the viscous

Presented as Paper 2004-0524 at the AIAA 42nd Aerospace Sciences Meeting, Reno, NV, 5–8 January 2004; received 25 February 2004; revision received 4 July 2004; accepted for publication 28 July 2004.

*Boeing Technical Fellow. Associate Fellow AIAA.

†CFD Engineer. Senior Member AIAA.

‡Manager. Senior Member AIAA.

flux tensor $\boldsymbol{G} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3)$ are given by

$$\bar{\boldsymbol{q}} = \frac{1}{V}\iiint\limits_{\text{cell}} \boldsymbol{q}\,\mathrm{d}V = [\bar{\rho}, \overline{\rho u}, \overline{\rho v}, \overline{\rho w}, \bar{e}]^T$$

$$\boldsymbol{f}_1 = [\rho u, p + \rho u^2, \rho uv, \rho uw, (e+p)u]^T$$

$$\boldsymbol{f}_2 = [\rho v, \rho vu, p + \rho v^2, \rho vw, (e+p)v]^T$$

$$\boldsymbol{f}_3 = [\rho w, \rho wu, \rho wv, p + \rho w^2, (e+p)w]^T$$

$$\boldsymbol{g}_1 = \frac{1}{Re}\left[0, \tau_{xx}, \tau_{xy}, \tau_{xz}, C_1\kappa\frac{\partial T}{\partial x} + u\tau_{xx} + v\tau_{xy} + w\tau_{xz}\right]^T$$

$$\boldsymbol{g}_2 = \frac{1}{Re}\left[0, \tau_{yx}, \tau_{yy}, \tau_{yz}, C_1\kappa\frac{\partial T}{\partial y} + u\tau_{yx} + v\tau_{yy} + w\tau_{yz}\right]^T$$

$$\boldsymbol{g}_3 = \frac{1}{Re}\left[0, \tau_{zx}, \tau_{zy}, \tau_{zz}, C_1\kappa\frac{\partial T}{\partial z} + u\tau_{zx} + v\tau_{zy} + w\tau_{zz}\right]^T \qquad (2)$$

where the bar refers to cell average and p, ρ, T, and (u, v, w) represent nondimensional static pressure, density, temperature, and Cartesian velocity components in the x, y, z directions. Total energy per unit volume e can be related to p, ρ, and (u, v, w) using the equation

$$e = p/(\gamma - 1) + (\rho/2)(u^2 + v^2 + w^2)$$

where γ is the ratio of specific heats. The equation of state relating p, ρ, and T can be written as

$$T = p/\rho C_2$$

The parameters Re (Reynolds number) C_1, in Eq. (2), and C_2 in the equation of state can be computed from the reference quantities and the universal gas constant R_0 as

$$Re = \frac{\rho_{\text{ref}}U_{\text{ref}}L_{\text{ref}}}{\mu_{\text{ref}}}, \qquad C_1 = \frac{\kappa_{\text{ref}}T_{\text{ref}}}{\mu_{\text{ref}}U_{\text{ref}}^2}, \qquad C_2 = \frac{\rho_{\text{ref}}R_0T_{\text{ref}}}{p_{\text{ref}}W_{\text{ref}}}$$

where κ is the thermal conductivity, μ the coefficient of viscosity, and W the molecular weight. The subscript ref refers to reference quantities. In Eq. (2), τ_{xx}, τ_{xy}, τ_{xz}, τ_{yx}, τ_{yy}, τ_{yz}, τ_{zx}, τ_{zy} and τ_{zz} are the components of the nondimensional viscous tensor.

From Eq. (1), a discretized form of the governing equations for each control volume (grid cell) is obtained by replacing the integrals with the weighted sum of the integrants at quadrature points. For second-order accuracy, the integrants can be evaluated at face centers. Temporal derivative is approximated by a backward-difference formula. Temporally first-order discretization leads to the implicit scheme

$$\frac{\bar{\boldsymbol{q}}^{n+1} - \bar{\boldsymbol{q}}^{n}}{\Delta t}V + \sum_{\text{faces}}(\boldsymbol{F}^{n+1} + \boldsymbol{G}^{n+1})\,\mathrm{d}\boldsymbol{a} = \boldsymbol{s}^{n+1} \qquad (3)$$

where the superscripts n and $n+1$ refer to time levels and $\Delta t = t^{n+1} - t^n$ is the increment in time. Note that the scheme is implicit because both the fluxes and the source terms are evaluated at time level $n+1$ and not time level n.

Using the Taylor series to derive first-order approximation in time for $\boldsymbol{F}^{n+1}$, $\boldsymbol{G}^{n+1}$, and $\boldsymbol{s}^{n+1}$, in terms of the solution at time level n, we get

$$\boldsymbol{F}^{n+1} = \boldsymbol{F}^{n} + \boldsymbol{J}_i^{\text{cv}}\Delta\bar{\boldsymbol{q}}_{\text{cv}} + \boldsymbol{J}_i^{\text{nb}}\Delta\bar{\boldsymbol{q}}_{\text{nb}}$$

$$\boldsymbol{G}^{n+1} = \boldsymbol{G}^{n} + \boldsymbol{J}_v^{\text{cv}}\Delta\bar{\boldsymbol{q}}_{\text{cv}} + \boldsymbol{J}_v^{\text{nb}}\Delta\bar{\boldsymbol{q}}_{\text{nb}}, \qquad \boldsymbol{s}^{n+1} = \boldsymbol{s}^{n} + \boldsymbol{J}_s^{\text{cv}}\Delta\bar{\boldsymbol{q}}_{\text{cv}} \qquad (4)$$

where $\Delta\bar{\boldsymbol{q}} = \bar{\boldsymbol{q}}^{n+1} - \bar{\boldsymbol{q}}^{n}$; cv, and nb refer to the control-volume and neighbor cells; the superscripts n and $n+1$ to time levels; and subscripts i, v, and s to inviscid, viscous, and source terms. The Jacobians $\boldsymbol{J}$ are defined by

$$\boldsymbol{J}_i^{\text{cell}} = \frac{\partial\boldsymbol{F}^n}{\partial\boldsymbol{q}_{\text{cell}}}, \qquad \boldsymbol{J}_v^{\text{cell}} = \frac{\partial\boldsymbol{G}^n}{\partial\boldsymbol{q}_{\text{cell}}}$$

and

$$\boldsymbol{J}_s^{\text{cell}} = \frac{\partial\boldsymbol{s}^n}{\partial\boldsymbol{q}_{\text{cell}}} \qquad (5)$$

From Eqs. (3–5) we arrive at the following discretized form of the governing equations:

$$\left\{\frac{V\boldsymbol{I}}{\Delta t} + \sum_{\text{faces}}\left[\left(\boldsymbol{J}_i^{\text{cv}} + \boldsymbol{J}_v^{\text{cv}} + \boldsymbol{J}_s^{\text{cv}}\right)\mathrm{d}\boldsymbol{a}\right]\right\}\cdot\Delta\bar{\boldsymbol{q}}_{\text{cv}}$$

$$= \boldsymbol{s}^n - \sum_{\text{faces}}(\boldsymbol{F}^n + \boldsymbol{G}^n)\,\mathrm{d}\boldsymbol{a} - \sum_{\text{faces}}\left[\left(J_i^{\text{nb}} + J_v^{\text{nb}}\right)\mathrm{d}\boldsymbol{a}\right]\Delta\bar{\boldsymbol{q}}_{\text{nb}} \qquad (6)$$

where $\boldsymbol{I}$ is the unit tensor.

Inviscid and viscous flux tensors $\boldsymbol{F}$ and $\boldsymbol{G}$ for a face are in general complex functions of $\boldsymbol{q}_{\text{cv}}$ and $\boldsymbol{q}_{\text{nb}}$, and hence exact expressions for the Jacobians are not employed in practice. For steady flows, it is not necessary to employ exact expressions because $\Delta\bar{\boldsymbol{q}} \to 0$ on convergence. Following Ref. 4, we define the inviscid Jacobians to be

$$\boldsymbol{J}_i^{\text{cv}} = \frac{1}{2}\left[\left(\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{q}}\right)_{\text{face}} + \beta|\lambda|_{\max}\right] \qquad (7a)$$

$$\boldsymbol{J}_i^{\text{nb}} = \frac{1}{2}\left[\left(\frac{\partial\boldsymbol{F}}{\partial\boldsymbol{q}}\right)_{\text{face}} - \beta|\lambda|_{\max}\right] \qquad (7b)$$

where $|\lambda|_{\max}$ is the maximum magnitude of the eigenvalue at the face center and β is a relaxation parameter, usually set equal to 1.5.

The discretized governing equations for cell averages [Eq. (6)] involve evaluation of fluxes at cell boundaries. Such an evaluation requires smart interpolation to ensure accuracy and stability. Inviscid and viscous fluxes are computed using different interpolation schemes. For second-order inviscid fluxes, the interpolation scheme employed in BCFD requires evaluation of gradients of the dependent variables at cell centers. Computation of gradients at cell centers involves the two steps described in the next two paragraphs.

In the first step, an initial approximation for the gradients is computed using surface integrals with values of the dependent variables at face centers computed as weighted means of their values at neighboring cell centers. That is,

$$\nabla q = \frac{1}{V_{\text{cell}}}\sum_{\text{faces}} q\boldsymbol{a} = \frac{1}{V_{\text{cell}}}\sum_{\text{faces}}(w_1q_{\text{cv}} + w_2q_{\text{nb}})\boldsymbol{a} \qquad (8)$$

where $\boldsymbol{q}$ refers to a component of $\boldsymbol{q}$ and the weights w_1 and w_2 can be computed either based on the cell volumes or on the distance between face center and cell centers.

In the next step, a slope-limiting procedure is employed to modify the gradients to ensure that when the gradients are used in the computation of face-center values from cell-center values, no new maxima or minima are created. Schemes that employ such a slope-limiting procedure are referred to as TVD schemes. Slope-limited gradients $\nabla\hat{q}$ are evaluated by comparing a cell gradient, component by component, with corresponding quantities from its neighbors using a minmod operator. The expression for the minmod operator is given by

$$|\nabla q_{\min}|_i = \min(|\nabla q_j|_i)\,j\varepsilon \qquad \text{neighbor cell}$$

$$(\nabla\hat{q})_i = \begin{cases} 0 \quad \text{when} \quad (\nabla q_{\text{cell}})_i\cdot(\nabla q_{\min})_i \le 0 \\ \text{sign}(\nabla q_{\text{cell}})_i\cdot\min[|\nabla q_{\text{cell}}|_i, cmp|\nabla q_{\min}|_i] \\ \text{when} \quad |\nabla q_{\text{cell}}|_i > |\nabla q_{\min}|_i \\ (\nabla q_{\text{cell}})_i \quad \text{when} \quad |\nabla q_{\text{cell}}|_i < |\nabla q_{\min}|_i \end{cases} \qquad (9)$$

where subscript i refers to a component of Δq and *cmp* is an input parameter > 1. A larger value for *cmp* yields more accurate solution. Smaller value increases stability.

The next step in the solution process is computation of inviscid fluxes at cell boundaries. An upwind scheme is employed for this purpose. Computation of upwind fluxes requires evaluation of a "left" and a "right" state at the centroid of a face using cell averages and slope-limited gradients of the dependent variables for the two neighboring cells. The expression for the two states can be written as

$$q_{\text{left}} = q_{\text{cv}} + \boldsymbol{r}_{\text{cf}} \cdot \nabla q_{\text{cv}} \tag{10a}$$

$$q_{\text{right}} = q_{\text{nb}} + \boldsymbol{r}_{\text{nf}} \cdot \nabla q_{\text{nb}} \tag{10b}$$

where $\boldsymbol{r}_{\text{cf}}$ and $\boldsymbol{r}_{\text{nf}}$ are radius vectors from corresponding cell centers to face center. The expression for the upwind flux resulting from Roe's scheme can be written as

$$\boldsymbol{F}^{Roe} = \tfrac{1}{2}[\boldsymbol{F}(\boldsymbol{q}_{\text{left}}) + \boldsymbol{F}(\boldsymbol{q}_{\text{right}})] - \tfrac{1}{2}|\boldsymbol{J}|(\boldsymbol{q}_{\text{right}} - \boldsymbol{q}_{\text{left}}) \tag{11}$$

where $\boldsymbol{J}$ is the Jacobian matrix computed at facecenter using $\boldsymbol{q}_{\text{left}}$ and $\boldsymbol{q}_{\text{right}}$ and Roe's approximate Riemann solver. In the case of the modified Rusanov's scheme, the flux can be computed from

$$\boldsymbol{F}^{Rusanov} = \tfrac{1}{2}[\boldsymbol{F}(\boldsymbol{q}_{\text{left}}) + \boldsymbol{F}(\boldsymbol{q}_{\text{right}})] - \tfrac{1}{2}\delta|\lambda|_{\max}(\boldsymbol{q}_{\text{right}} - \boldsymbol{q}_{\text{left}}) \tag{12}$$

where δ is an input parameter. The original Rusanov's scheme corresponds to $\delta = 1$. While $\delta = 0.5$ results in a stable scheme for many problems with only half the dissipation as the original Rusanov's scheme, it has been possible to obtain stable solutions with values for δ as low as 0.05. In most cases the value employed is between 0.1 and 0.25.

Computation of viscous fluxes requires evaluation of gradients at face centers. In BCFD, gradients at face centers are computed in such a way that the components of gradients in the direction normal to a face are obtained using the cell averages from neighboring cells, whereas the components in the plane of the face are computed as a weighted mean of the corresponding components of the two cell gradients. The expressions for the viscous flux Jacobians employed in BCFD are the same as in Ref. 5.

With the computation of fluxes, Eq. (6) can now be solved to obtain $\Delta\bar{\boldsymbol{q}}$. BCFD employs a point-implicit scheme for solving Eq. (6). Usually four iterations are performed for every time step. A pair of iterations corresponds to a forward sweep followed by a backward sweep. A forward sweep corresponds to solving for $\Delta\bar{\boldsymbol{q}}$, starting with the first cell and proceeding in the ascending order. The backward sweep corresponds to starting from the last cell and proceeding in the descending order. In the forward sweep, starting with an initial approximation for $\Delta\bar{\boldsymbol{q}}$, Eq. (6) is solved to obtain $\Delta\bar{\boldsymbol{q}}_{\text{cell}}$, cell = 1, no of cells. Then, $\Delta\bar{\boldsymbol{q}}_{\text{nb}}$ is obtained from the initial approximation $\Delta\bar{\boldsymbol{q}}$. In the case of the backward sweep, $\Delta\bar{\boldsymbol{q}}_{\text{nb}}$ computed from the forward sweep is employed. In BCFD, an initial approximation for $\Delta\bar{\boldsymbol{q}}$ is obtained using a diagonal scheme.

Validation and Results

Several cases have been run to validate the unstructured grid algorithms implementations in the BCFD code. The BCFD code is operational on structured and hybrid unstructured grids. The viscous grids for all validation cases were prismatic in the boundary layer and tetrahedral outside the boundary layer and away from the surfaces.

Verification of Implementation on a Flat Plate

The first case used for validation is subsonic flow at Mach = 0.2, $Re = 2.25 \times 10^6$ over a flat plate. The y^+ for this case was set at 0.5, and as a result it took over 30,000 iterations to converge. In general, for $y^+ < 1$ the pressure field and over all viscous forces (90%) are established quickly, but the last 10% convergence of the viscous forces takes a substantial number of iterations. To overcome this problem, we are implementing grid-agglomeration capability in the BCFD code. However, it is possible to get accurate solutions with y^+ of 3 to 5. The skin-friction coefficients are shown in Fig. 1 for CFD and experimental data. The unstructured grid CFD result matches the data very well.

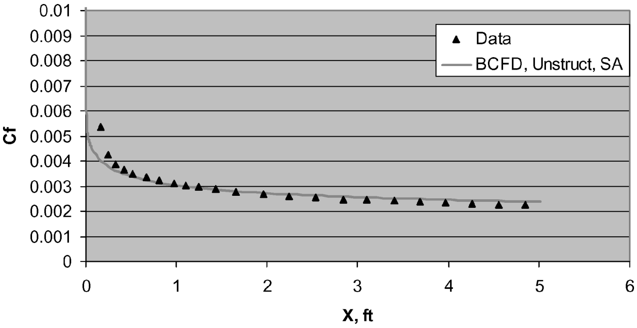


Fig. 1 Skin-friction coefficients for a flat plate at Mach = 0.2, $y^+ = 0.5$.

Solution Demonstration on an ONERA Wing

The CFD results were obtained for the ONERA wing at Mach = 0.84, angle of attack (AOA) = 5.06, $Re = 11.78 \times 10^6$ with both structured and unstructured grid. Both sets of CFD results were obtained using the one-equation S-A model. The surface-pressure coefficients are compared with each other and experimental data at several spanwise locations in Figs. 2a–2f. The CFD results match each other and data very well.

Solution Demonstration with a Generic Weapon

The unstructured grid enables detailed modeling of a configuration without any additional time and negative impacts on the Navier–Stokes solver. In this case fins and strakes were modeled. The grid contains 4.65 million cells. The solutions were obtained for freestream Mach = 0.8, pressure = 6.75 psi, and temperature = 447.4^0R. The one-equation S-A turbulence model and Rusanov algorithm were employed for this study. The CFD solutions were obtained for six different angles of attack between 4 and 30 deg. The configuration and surface-pressure coefficient contours are shown in Figs. 3 and 4. The CFD results are compared against the data for pitching moments and normal-force coefficients in Figs. 5 and 6, respectively. The CFD results match the data very well at all angles of attack.

Generic Commercial Aircraft

A simplified configuration representing a commercial aircraft was chosen to evaluate the code performance. Two grids were generated, one representative of a wall function and the other solving to the wall. The wall-function grid contains 4.3 million cells and the grid for solving to the wall 5.0 million. The surface grid on the wing is shown in Fig. 7, and a cross section of the volume grid on the wing is depicted in Fig. 8. Prismatic grid is used in the boundary layer, pyramid in the transition layer from prism to tetrahedral, and tetrahedral in outer layer.

Solutions were obtained at a freestream Mach number of 0.2, $Re = 2.36 \times 10^6$/chord and AOA = 4,10, and 16 deg. It is significantly faster to obtain a solution with the wall function than to solve to the wall. The wall-function approach is not accurate for all flow conditions such as massively separated flows. However, with care one can use the wall-function grid to obtain accurate and fast turnaround time. To make sure that there are sufficient points in the boundary layer, the grid needs to be restretched from the second point onward. The surface-pressure coefficient for AOA = 4 is shown in Fig. 9. The lift and drag coefficients are shown in Figs. 10a and 10b, and the results compare well with the data. The solutions were converged within 12,000 iterations and approximately 60 h on 18 CPUs. Figure 11 shows the history of the lift convergence. As one can see, the lift was about 90% converged at about 4000 iterations. However, it took another 8000 iteration to fully converge the solution. This is representative of most cases we have obtained.

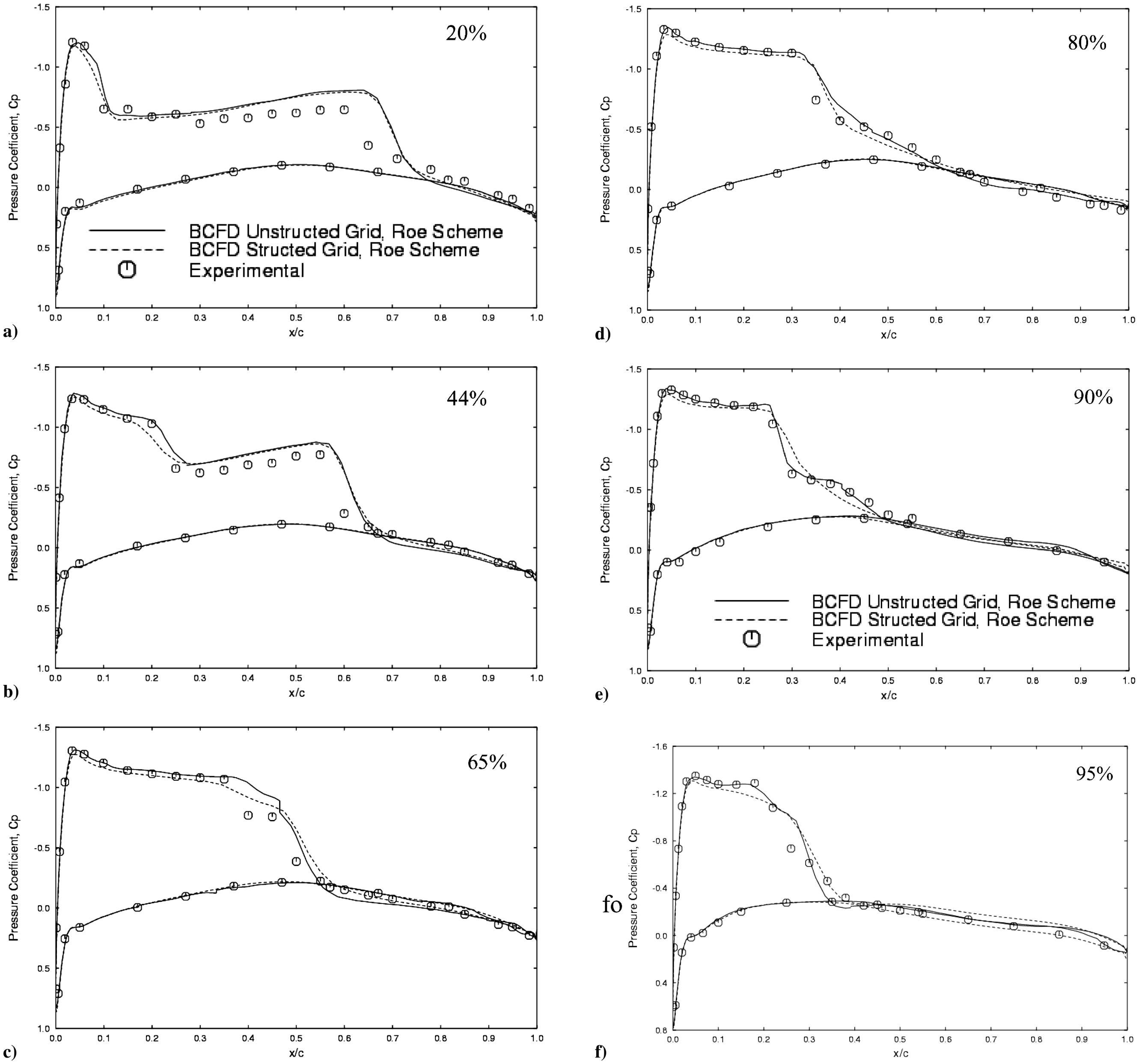


Fig. 2 Surface-pressure coefficients for ONERA wing at a–d) two different wing-spans and e) and f) four different wing spans.

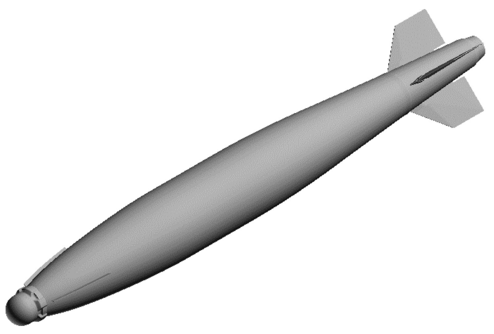


Fig. 3 Weapon configuration.

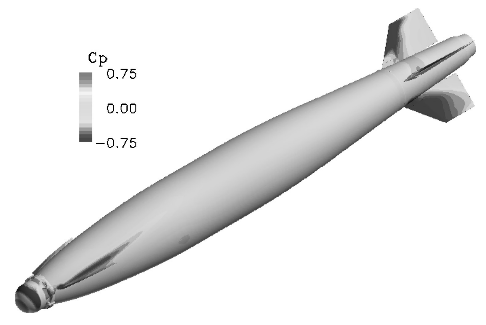


Fig. 4 Surface-pressure coefficient.

Solution Demonstration with F-18 Store Carriage

The CFD Challenge II cases, summarized in Ref. 6, assess the accuracy of CFD codes in predicting the loads on a Mk-84 joint direct attack munition (JDAM) carried on the outboard wing pylon of a CF/A-18C. Carriage loads have been predicted using CFD by a number of authors (Refs. 7–13) at two conditions: $M = 0.962$ at 6382 ft altitude and an angle of attack of 0.46 deg and $M = 1.055$ at 10,832 ft altitude and an angle of attack of -0.65 deg. Wind-tunnel and flight measurements are also available at these conditions. Figure 12 depicts the supersonic solution computed with BCFD on this configuration. The grid had 6.7 million cells consisting of both tetrahedral and prisms elements. Prism layers were generated around the Mk-84 JDAM and its pylon, whereas the rest of the aircraft and volume used approximately isotropic tetrahedral cells. Slip walls were used everywhere except for the store and pylon of interest to reduce the grid size and computational cost. This grid and the resulting solution cost

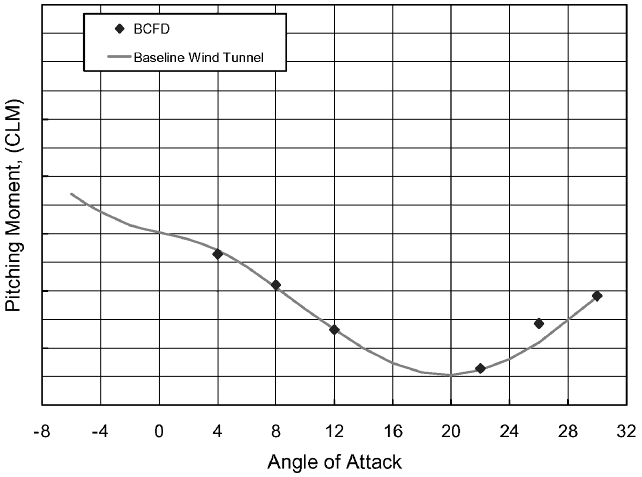


Fig. 5 Pitching-moments coefficients.

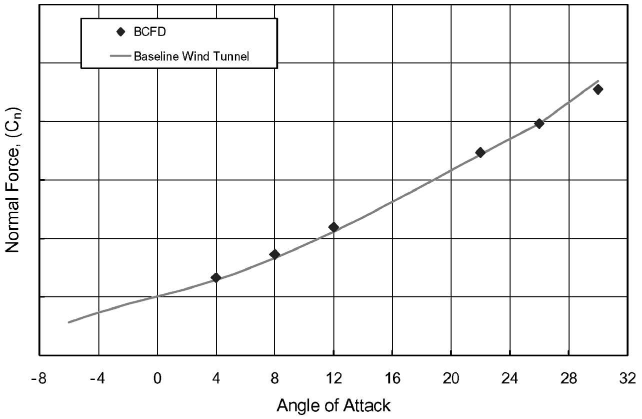


Fig. 6 Normal-force coefficients.

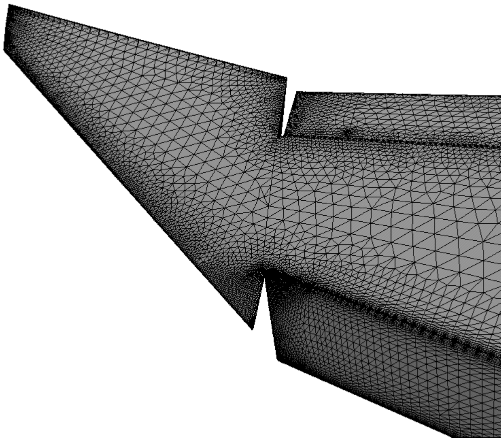


Fig. 7 Surface grid.

were quite comparable to data that had been reported for comparable flow solvers on this problem (Refs. 6 and 13). Table 1 compares the wind-tunnel and flight data reported in Ref. 14 to the BCFD force-and-moment coefficients. Although there is as much as 25% difference in some of the force-and-moment data, the BCFD results fell well within the scatter in the results of the other reported codes.

Table 1 Force-and-moment coefficient comparison between experiment and BCFD for weapon carriage simulations

Data source	C_Y	C_N	C_m	C_n
		$M = 0.962$ *case*		
Wind tunnel	0.3105	0.1088	−2.3208	2.7642
Flight test	0.3140	0.1491	−2.50	2.8019
BCFD	0.2416	0.0856	−2.120	2.092
		$M = 1.055$ *case*		
Wind tunnel	0.2351	0.0246	−2.0660	2.5566
Flight test	0.2509	0.0491	−2.00	2.2075
BCFD	0.2126	0.0115	−2.197	1.9387

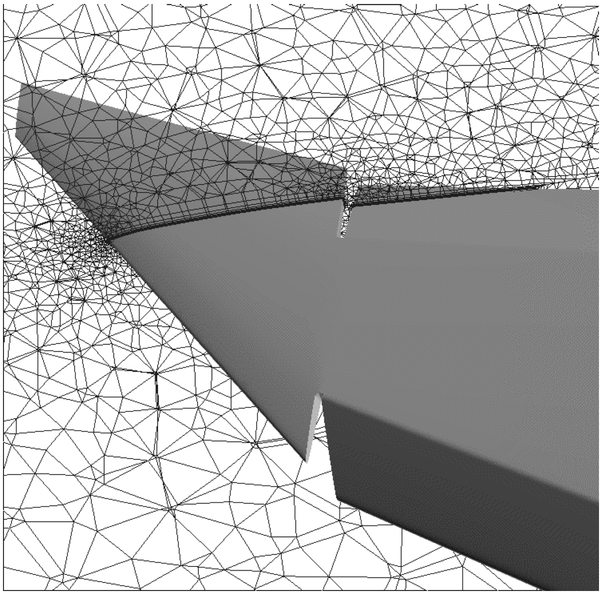


Fig. 8 Sectional cut of the volume grid.

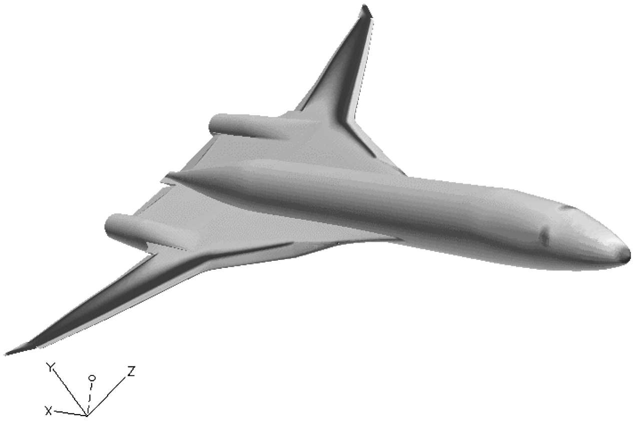


Fig. 9 Surface-pressure coefficient.

Convergence Acceleration

Solution convergence in the BCFD code can be tracked with residual or the force and moments. The residual usually drops by about two to three orders of magnitude before it levels off. The force and moments are usually a better gauge for the convergence in the BCFD code. There are a couple of ways to accelerate the solution: one is by reordering cells to a string of lines 1000 point each. This would allow the use of higher Courant–Friedrichs–Lewy (CFL) and, as a result, faster convergence. Figure 13a shows the convergence history of the pressure coefficients with reorder cells and CFL = 50 and no reordering cells with CFL = 5. There is about a factor of two improvements in convergence. The second approach is to use the reorder-cell grid with the implicit line Gauss-sidle. The line implicit improves the convergence by another factor of two over the point implicit at the same CFL number (Fig. 13). It appears that by increasing the line from 5 points to 1000 points and CFL from 50 to 500, the solution convergence increased only slightly (by about 500 iterations). Of course by increasing the number of points per line, the memory requirements go up.

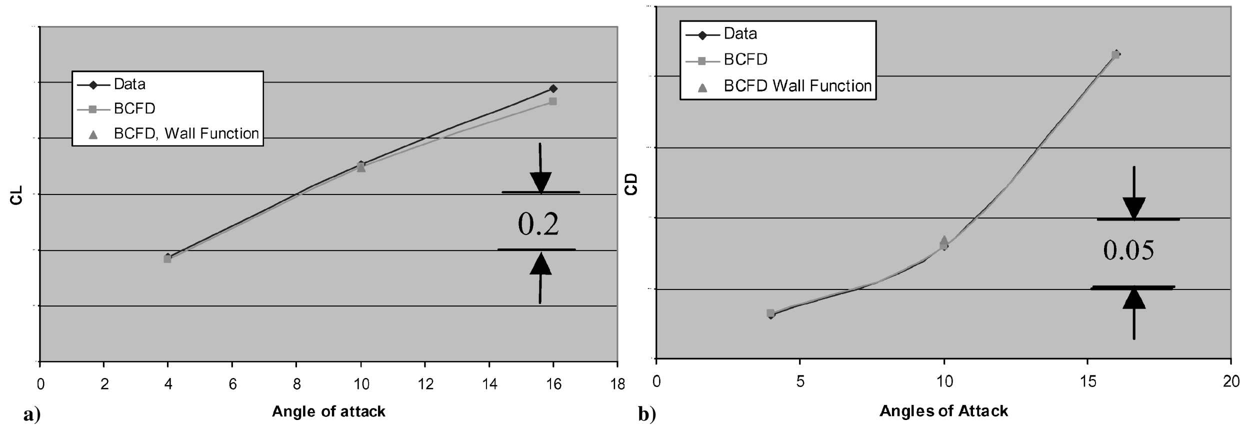


Fig. 10 Comparison of CFD and data for lift and drag coefficients.

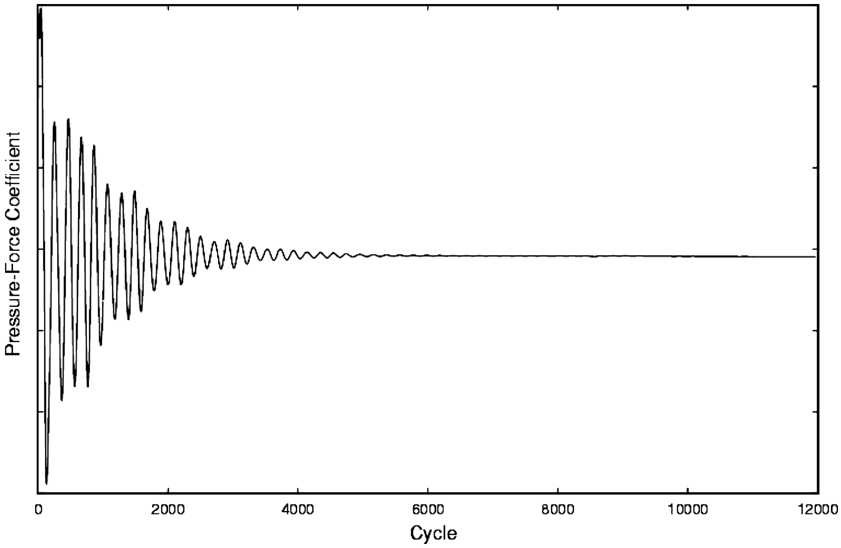
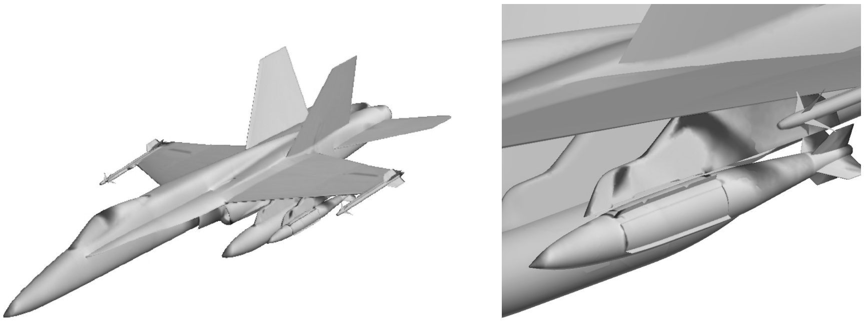


Fig. 11 History of lift coefficient convergence.



a) Aircraft pressure coefficient contours **b) Store and pylon detail pressure coefficient contours**

Fig. 12 Pressure coefficients on CF/A-18C and Mk-84 JDAM at $M = 1.055$, 10,832 ft altitude, and −0.65 angle of attack.

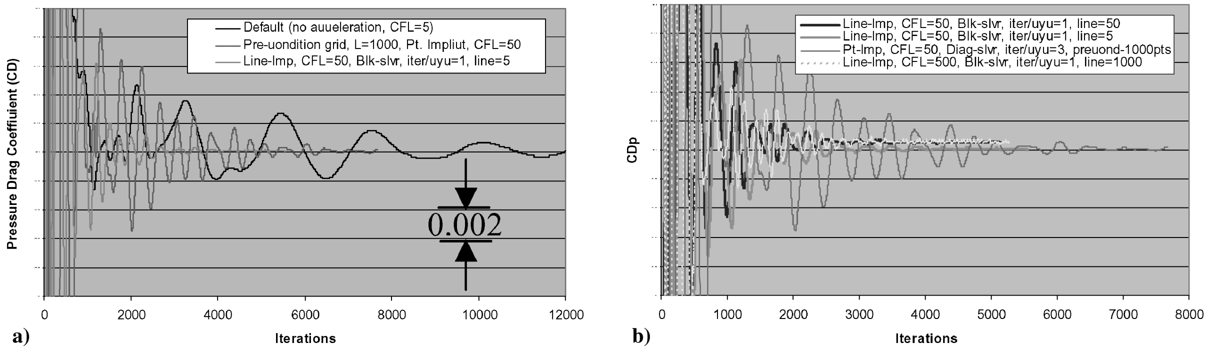


Fig. 13 Convergence history of pressure-drag coefficients.

Conclusions

A general geometry, three-dimensional, implicit, Euler, and Navier–Stokes solver for structured and hybrid-unstructured grids has been developed. Two one-equation turbulence models for the unstructured grid have been implemented. The Liu–Vinokur equilibrium air curve fits and finite-rate chemistry for high-speed and chemically reacting flow have been implemented. The code has been validated for a wide range of applications from low subsonic through hypersonic. To achieve the fast turnaround time required by all projects, it is necessary to implement a multigrid capability in the BCFD code. We have begun the implementation of the two-level grid agglomeration with the ultimate goal of n-level multigrid capability.

References

[1] Chakravarthy, S. R., and Szema, K. Y., "Unified Nose-To-Tail Computational Method for Hypersonic Vehicle Applications," AIAA Paper 88-2564, 1988.

[2] Roe, P. L., "Approximate Riemann Solvers, Parameter Vectors, and Difference Schemes," *Journal of Computational Physics*, Vol. 43, 1981, pp. 357–372.

[3] Zhining, Liu, Ramakrishnan, S. V., Olling, C., and Rajagopal, K. R., "Aeroheating Prediction Using the Hybrid Flow Solver, ICAT," AIAA Paper 02-2936, 2002.

[4] Jameson, A., and Turkel, E., "Implicit Schemes and LU Decompositions," *Mathematics of Computation*, Vol. 37, No. 156, Oct. 1986, pp. 385–397.

[5] Merkle, C. L., et al., "Computation of Flows with Arbitrary Equations of State," *AIAA Journal*, Vol. 36, No. 4, 1998.

[6] Cenko, A., Gowanlock, D., Lutton, M., and Tutty, M., "F/A-18C/JDAM Applied Computational Fluid Dynamics Challenge II Results," AIAA Paper 00-0795, Jan. 2000.

[7] Tomaro, R. F., Witzeman, F. C., and Strang, W. Z., "A Solution on the F-18C for Store Separation Using Cobalt$_{60}$," AIAA Paper 99-0122, Jan. 1999.

[8] Woodson, S. H., and Bruner, C. W. S., "Analysis of Unstructured CFD Codes for the Accurate Prediction of Aircraft Trajectories," AIAA Paper 99-0123, Jan. 1999.

[9] Welterlen, T. J., "Store Release Simulation on the F/A-18C Using Splitflow," AIAA Paper 99-0124, Jan. 1999.

[10] Fairlie, B. D., and Caldeira, R. H., "Prediction of JDAM Separation Characteristics from the F/A-18C Aircraft," AIAA Paper 99-0126, Jan. 1999.

[11] Fortin, F., Benmeddour, A., and Jones, D. J., "Application of the Canadian Code to the F/A-18C JDAM Separation," AIAA Paper 99-0127, Jan. 1999.

[12] Noack, R. W., and Jolly, B., "Fully Time Accurate CFD Simulations of JDAM Separation from an F-18C Aircraft," AIAA Paper 00-0794, Jan. 2000.

[13] Sickles, W. L., Denny, A. G., and Nichols, R. H., "Time-Accurate CFD Predictions of the JDAM Separation from an F-18C Aircraft," AIAA Paper 00-0796, Jan. 2000.

[14] Cenko, A., "F/A-18C/JDAM CFD Challenge Wind Tunnel and Flight Test Results," AIAA Paper 99-0120, Jan. 1999.