AIAA JOURNAL
Vol. 43, No. 2, February 2005

Euler Solution Using Cartesian Grid with a Gridless Least-Squares Boundary Treatment

E. P. C. Koh* and H. M. Tsai†
National University of Singapore, Singapore 119260, Republic of Singapore
and
F. Liu‡
University of California, Irvine, Irvine, California 92697-3975

An approach that uses gridless or meshless methods to address the problem of boundary implementation associated with the use of Cartesian grid is discussed. This method applies the gridless concept only at the interface, whereas a standard structured grid method is used everywhere else. The Cartesian grid is used to specify and distribute the computational points on the boundary surface but not to define the geometrical properties. Euler fluxes for the neighbors of cut cells are computed using the gridless method involving a local least-squares curve fit of a "cloud" of grid points. The boundary conditions implemented on the surface points are automatically satisfied in the process of evaluating the surface values in a similar least-squares fashion. The present method does not require the use of halo points. Subsonic, transonic, and supersonic flows are computed for the NACA 0012 and RAE 2822 airfoils, and the results compare well with solutions obtained by a standard Euler solver on body-fitted grids. The method is also used to calculate the flow over a three-element airfoil configuration, and the result is compared with the exact solution for this configuration obtained by conformal mapping.

I. Introduction

A CONTINUING obstacle of computational fluid dynamics for configurations with complex geometry is the problem of mesh generation. Existing methods of flowfield discretization include using unstructured meshes, body-fitted curvilinear meshes, and Cartesian meshes. Unstructured meshes are typically constructed from triangles in two dimensions or tetrahedral cells in three dimensions (e.g., Refs. 1–4). The main advantage of traditional unstructured grids is the ease of grid generation about complex configurations because the cells can be oriented in any arbitrary way to conform to the geometry. However, the computational time and cost for unstructured mesh computations are generally higher than the structured mesh approach. This is especially so when applied to large-scale three-dimensional problems.

A widely used mesh for flow computation is the body-fitted, structured, curvilinear mesh. Its main advantage is the ease in implementing boundary conditions caused by the body-aligned nature of the mesh. However, a major drawback for this is the difficulty of mesh generation for complex geometry. This results in the occurrence of highly distorted or skewed cells in some regions of the domain that adversely influence the computation. Very often, it is not possible to use a single grid for highly complex geometries. This gives rise to more sophisticated methods such as multiblock, patched grids, Chimera, or hybrid-type grids. They are used in varying degrees of success (e.g., Refs. 5–7). However, these methods require transfer of information between the different meshes and the involvement of other complicated issues.

Used in an embedded grid manner, a Cartesian grid method offers great potential in dealing with complex configurations when grid generation becomes the limiting factor for body-fitted grid method. In recent years there is a renewed interest in the conceptually simple approach of Cartesian grids.[8–10] Both structured and unstructured methods of solution can be applied to grids created by this means. Besides the attractive advantages associated with structured grids, a structured Cartesian grid method offers ease of grid generation, lower computational storage requirements, and significantly less operational count per cell compared to body-fitted curvilinear grids. The convergence properties of the solver are also potentially better because there are no problems related to skewness or distortion of cells. Embedded grids, adaptive local grid refinement, and other grid-refinement strategies could be used to provide better resolution of geometry and flow features (e.g., Refs. 11–17). Multigrid acceleration can be implemented in the framework of embedded grids.[12] Furthermore, Cartesian grids offer considerable ease in implementing higher-order schemes. However, the main challenge in using a Cartesian method is in dealing with arbitrary boundaries. As the grids are not body aligned, Cartesian cells near the body can extend through surfaces of solid components. Hence, accurate means of representations for surface boundary conditions are essential for the success of Cartesian schemes.

Different methods have been proposed in the literature to resolve the boundary conditions either using unstructured or structured grid techniques. Broadly the methods proposed involve either cut cells for a finite volume treatment or grid points for a finite difference construction. A cut-cells method ensures conservation and uses the same flux computation algorithm as for the interior cells. However, the task of computing the volume and fluxes for all of the irregularly shaped cut cells entails a considerable increase in complexity. More crucially, the method can at times lead to the creation of very small cells at the boundary, which pose problems of numerical stability. To alleviate this problem, rule-based cell-merging techniques were used to combine cut cells that are too small with neighboring cells as in the work of Clarke et al.[18] and Ye et al.[19] Although this approach ensures strict conservation properties, cell formation and merging is not straightforward in three dimensions. Furthermore it is difficult to ensure that the local geometric properties are in full consistency with the original shape particularly when coarse grids are being used for multigrid implementation.

Presented as Paper 2003-1120 at the 41st Aerospace Sciences Meeting, Reno, NV, 6–9 January 2003; received 31 October 2003; revision received 9 August 2004; accepted for publication 19 August 2004.

*Associate Scientist, Temasek Laboratories, Kent Ridge Crescent.
†Principal Research Scientist, Temasek Laboratories, Kent Ridge Crescent. Member AIAA.
‡Professor, Department of Mechanical and Aerospace Engineering. Associate Fellow AIAA.

In a finite difference approach, however, boundary conditions can be handled by an extrapolation procedure.[11,12,20] Halo points might need to be used in such an approach. Unlike the cut-cells method, this method is not conservative and requires special treatment for regions with thin surfaces (e.g., a trailing edge) where multiple values are defined at a particular point.[11,12] To minimize the lack of conservation, higher-order representation for the boundary is used to improve solution accuracy as seen in the work of Forrer and Jeltsch.[21]

An alternative to reducing the effort associated with mesh generation is the use of a purely gridless approach for the entire computational domain (e.g., Refs. 22–27). Such methods typically incorporate either a direct least-squares fitting of the flux variables or can employ a moving least-squares fitting to establish trial and test functions in a finite element formulation. Such formulations have been implemented successfully within the field of fracture mechanics and crack propagation.[28] Batina pioneered the use of a pure gridless method for compressible fluid dynamics.[22] However, there are issues associated with global conservation of mass, momentum, and energy in the main flowfield, where shock waves can occur. In addition, such methods tend to be less efficient compared to conventional methods using structured meshes as a result of extra work in the construction of the least-squares fitting over clusters of grid points. In addition, it is not straightforward to implement acceleration techniques such as the multigrid method. Consequently, such pure gridless methods have not found wide use in the computational-fluid-dynamics community despite its purported flexibility.

The present paper attempts to combine the advantages of the conventional Cartesian grid method and the gridless method while avoiding their shortcomings. Unlike Batina's approach,[22] where the gridless technique is used for the whole flowfield, we use a Cartesian method to solve the flow equations for the grid points in the interior of the computational domain where the finite difference or finite volume stencils are complete. The gridless method, however, is used to implement boundary conditions and treat grid points for which the finite difference or finite volume stencil is not complete. The Cartesian method is used over the bulk of the computational domain for its efficiencies, whereas the gridless method is adopted for its flexibility in handling the complex arbitrary distribution of grid points near solid boundaries. The gridless method is only applied to a layer of cells around the object instead of the entire flowfield. In this way, we expect to arrive at a general solution method that is flexible, efficient, and accurate for problems with complex geometries. A similar approach to the present work was reported by Kirshman and Liu,[29] who used a finite difference scheme with Van Leer flux-splitting technique. In the present work, a finite volume formulation with central differencing for the Euler equations is used, and an alternative simpler gridless approach is employed in discretizing the surface. The Cartesian grid is used to define and distribute the computational points on the boundary surface to obtain a set of evenly distributed surface nodes with respect to the Cartesian grid. This helps to reduce the possibility of ill-conditioned matrices in the gridless computation with cloud points comprising of both surface nodes as well as Cartesian grid points. To preserve accuracy at the boundary, geometrical properties are defined using the original data of the geometry and not the surface computational points. Because of the unique way of surface definition, the problem of singularity at the trailing edge is not as pronounced in this method. An overall simplified implementation of boundary conditions is adopted. Euler fluxes for the neighbors of cut cells are computed using a gridless method with the resulting derivatives analytically determined for the flow equations while boundary conditions implemented on the surface nodes are automatically satisfied in the process of evaluating the surface values in a similar least-squares method.

In the next section, the numerical method that combines a Cartesian method for the two-dimensional Euler equations and a gridless method for treating boundary conditions are presented. Determination of the gridless cloud points for the boundary and the spatial discretization for the flowfield and boundary nodes are detailed. The discretized Euler equations are solved using the modified four-stage Runge–Kutta scheme developed by Jameson et al.[30] and Jameson.[31] The method has been used to solve the flowfield over the NACA 0012 and the RAE 2822 airfoils in the subsonic, transonic, and supersonic regimes. Solutions are compared with results using a standard Euler solver FLO52 by Jameson over body-fitted curvilinear grids. In addition, the method is used to calculate the flow over a three-element airfoil configuration, and the result is compared with exact solution obtained by conformal mapping.

II. Numerical Method

A. Basic Finite Volume Method for the Euler Equations

The two-dimensional Euler equations consisting of the mass, momentum, and energy conservation laws that govern the motion of an inviscid fluid can be written in integral form as

$$\frac{\partial}{\partial t}\iint_{\Omega} U \,\mathrm{d}\Omega + \oint \boldsymbol{F}\cdot\boldsymbol{n}\,\mathrm{d}S = 0 \tag{1}$$

where Ω denotes the volume (area in two-dimensions); $\boldsymbol{F}$ is the flux vector; $\mathrm{d}S$ is a surface element, and $\boldsymbol{n}$ is the outward normal; and

$$U = \begin{Bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{Bmatrix}, \qquad f = \begin{Bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uH \end{Bmatrix}, \qquad g = \begin{Bmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho vH \end{Bmatrix} \tag{2}$$

Here, p, ρ, u, v, E, and H denote the pressure, density, Cartesian velocity components, total energy, and total enthalpy. For a perfect gas, $E = p/(\gamma - 1)\rho + \frac{1}{2}(u^2 + v^2)$ and $H = E + p/\rho$, where γ is the ratio of specific heats.

The computer program to solve the preceding system of equations is based on a version of FLO52 by Jameson. The basic numerical algorithms have been described in detail in Refs. 31 and 32. Major features are summarized next.

Spatial discretization of the Euler equations for an interior node in the Cartesian grid is performed using a second-order cell-centered finite volume method. Let (i, j) denote the cell center of such a grid with grid spacing Δx and Δy in the x and y directions, respectively. In a cell-centered finite volume scheme, the fluxes are evaluated by taking the average of the values in the cells on either side of each edge. On a Cartesian grid this reduces to the central-difference scheme

$$\frac{\mathrm{d}U_{i,j}}{\mathrm{d}t} + \frac{f_{i+1,j} - f_{i-1,j}}{2\Delta x} + \frac{g_{i,j+1} - g_{i,j-1}}{2\Delta y} = 0 \tag{3}$$

where f and g are the x and y components of the flux vector $\boldsymbol{F}$. The finite volume stencil for the cell center at (i, j) consists of points (cell centers) at (i, j), $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, and $(i, j+1)$.

The Euler equations, which are a set of nondissipative hyperbolic conservation laws, require artificial dissipation terms to be added to the discretized Euler equations. This is to suppress the tendency for odd-and-even decoupling and to prevent oscillation near shock waves. An adaptive scheme using second and fourth differences for adding dissipation is used. The Runge–Kutta multistage time integration[31] is applied to Eq. (3). The scheme is explicit with a Courant–Friedrichs–Lewy (CFL) condition of CFL $\leq 2\sqrt{2}$ and has second-order accuracy in time for a nonlinear equation. To increase the allowable stability bound, implicit residual smoothing and localtime stepping can be used to accelerate convergence to steady state.

B. Boundary and Surface Nodes Definition and Strategy of Mixed Cartesian and Gridless Method

Nodes (cell centers) near a solid body surface might not have a complete stencil needed for applying Eq. (3). Four types of nodes near the solid boundary are identified as shown in Fig. 1. Type 1

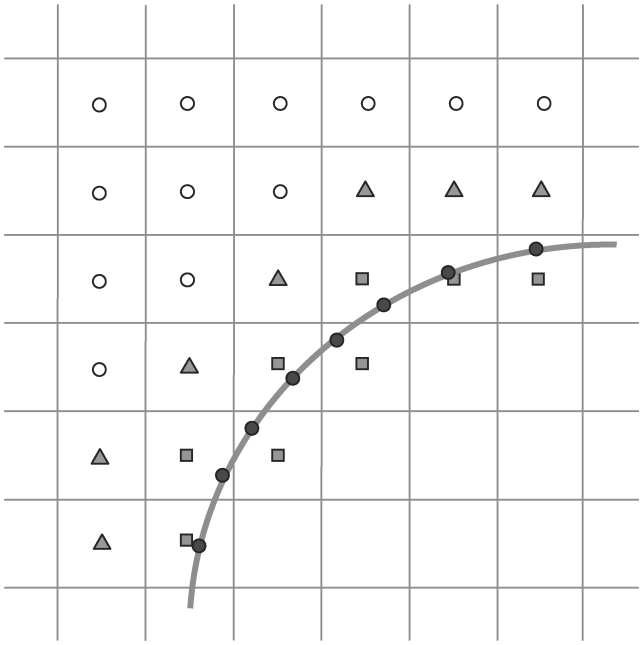


Fig. 1 Classification of grid nodes: ○, type 1, fluid nodes; △, type 2, boundary nodes; □, type 3, cut cell nodes; and ●, type 4, surface nodes.

nodes are nodes whose Euler fluxes can be computed easily in the finite volume method because the values of all four neighbors (left, right, top, and bottom) are defined, that is, the computational stencil for the finite volume is complete in the second-order scheme. These are termed fluid nodes. Type 2 nodes are nodes that have cut cells as neighbors and are termed boundary nodes. For these nodes, the finite volume method cannot be applied, and a gridless method with the use of a least-squares technique will be applied to compute the Euler fluxes. Type 3 nodes are nodes whose cells are cut by solid wall surfaces and are termed cut-cell nodes, and type 4 nodes are surface nodes that are defined on the body. The boundary and surface nodes are managed using an unstructured data approach.

The procedure for boundary and surface nodes determination is listed as follows:

1) Cut cells with reference to the Cartesian mesh and surface data points are identified. A feasible method for this is to use alternate-digital-tree approach whereby the intergrid information is classified in a tree structure for quick identification. With the cut cells found, solid cells that fall completely inside the object are marked out, and boundary nodes are identified.

2) A set of well-distributed points on the surface is obtained by "latching" the center of each cut cell onto the surface of the object to obtain the surface nodes. This is accomplished using a nonuniform rational B-splines (NURBS) program. The surface nodes obtained are the nearest point on the surface to the center of the cut cells. Details of the NURBS representation used can be found in Ray and Tsai.[32] With this latching method, the interval between the surface nodes will be approximately the same as the grid cell interval. Using a set of well-distributed points will help reduce the possibility of ill-conditioned matrices in the least-squares computation.

3) Normal and tangent vectors for the surface nodes are found using the original data inputs of the object instead of the computational points on the surface for better accuracy. Radius of curvatures at these nodes is also defined for boundary implementation. At the tip of airfoil trailing edge, these values are obtained by extrapolation to avoid singularity.

A cloud of points for each boundary node is set up to compute Euler fluxes, whereas a cloud of points for each surface node is used for boundary condition implementation and surface value determination. For each boundary node, a cloud of nine points with a mixture of fluid nodes, boundary nodes, and surface nodes are selected from its vicinity. This is determined using a 3×3 stencil over the boundary node as shown in Fig. 2a. Fluid nodes and boundary nodes that fall in the stencil are included, whereas cut cells are omitted. These are replaced by surface nodes that are closest to the boundary nodes and have outward normal pointing in the direction of the selected boundary nodes. This is to avoid choosing nodes from the wrong side of the profile at thin surfaces as illustrated in Fig. 2b. For each surface node, a one-sided cloud comprising only fluid nodes and boundary nodes is defined. The selected cloud points must fall on the same side as the outward normal direction of the surface node, and the selection is based on nearest distance to the node. As such, surface values are evaluated using computational nodes on the side of its outward normal as illustrated in Fig. 3. A mixture of different types of nodes in each cloud will ensure the transfer of information between these nodes.

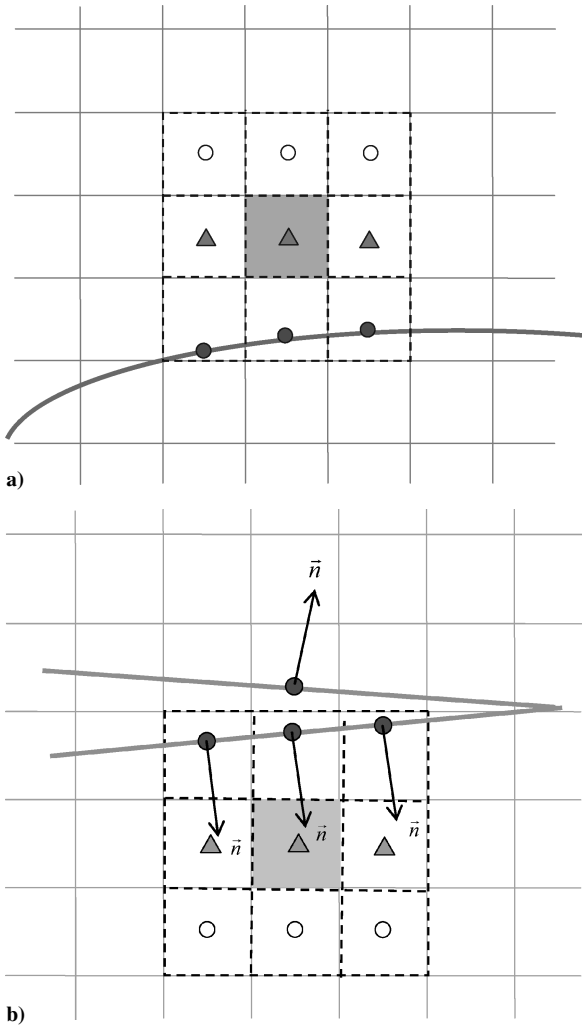


Fig. 2 Cloud points for a) boundary nodes and b) boundary nodes at thin surfaces.

1. Boundary Node Treatment

As boundary nodes do not have a complete stencil for the finite volume approximation, a gridless method is used to compute the Euler fluxes for these nodes. This is a very simple method for boundary treatment as it uses only clouds of points and does not require the points to have any prespecified connectivity among one another nor the formation of meshes or cells. The boundary nodes are treated separately. This makes the method easy to be implemented into any scheme.

Using the cloud points defined for each boundary node, the governing equation is solved through a local least-squares curve fit using these points. In each cloud, each term of the fluxes as defined in Eq. (2) is assumed to vary according to a function:

$$h(x, y) = a_0 + a_1 x + a_2 y + a_3 xy \tag{4}$$

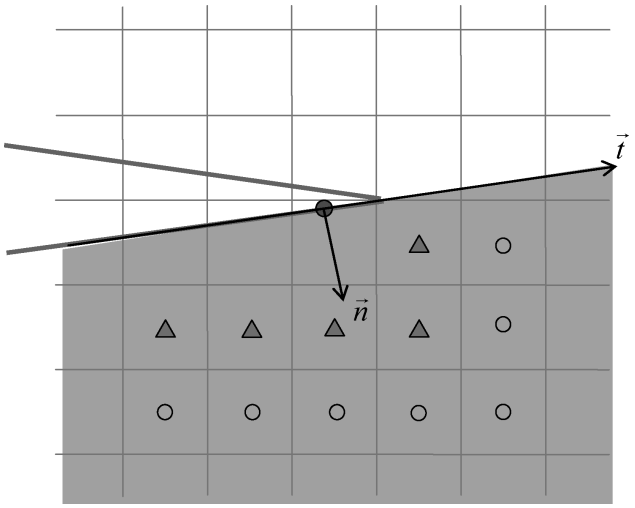


Fig. 3 Cloud points for surface nodes (closest eight points to surface node in shaded region).

Performing a least-squares curve fit in a given cloud results in a system of four equations:

$$\begin{bmatrix} n & \sum x_i & \sum y_i & \sum x_i y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i & \sum x_i^2 y_i \\ \sum y_i & \sum x_i y_i & \sum y_i^2 & \sum x_i y_i^2 \\ \sum x_i y_i & \sum x_i^2 y_i & \sum x_i y_i^2 & \sum x_i^2 y_i^2 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \sum h_i \\ \sum x_i h_i \\ \sum y_i h_i \\ \sum x_i y_i h_i \end{bmatrix} \tag{5}$$

The spatial derivatives can be found by differentiating Eq. (4) and are given as follows:

$$\frac{\partial h}{\partial x} = a_1 + a_3 y, \qquad \frac{\partial h}{\partial y} = a_2 + a_3 x \tag{6}$$

Hence by determining the values of a_1, a_2, and a_3, the spatial derivatives are found and the Euler fluxes evaluated. Although simple to implement, the gridless method is not a conservative method as pointed out by Batina.[22] By applying the gridless method only on boundary nodes, potential nonconservation is thus restricted to a layer of cells surrounding the object while the rest of the computational domain is computed using a fully conservative finite volume formulation. The size of this layer goes to zero, and thus the effect of the potential nonconservative effect also goes to zero as the grid size is reduced. The lack of conservativeness can be minimized with the use of sufficient resolution near the surface.

Artificial dissipative terms for boundary nodes are computed in the same manner as the fluid nodes. Special treatment is required to approximate the cut-cell values using the surface node values for the computation of local pressure gradient. Tests show that setting the cut-cell values to zero will result in large oscillations near the surface boundary as a result of an underestimation of the local pressure gradient.

Both the fluid nodes and the boundary nodes will be marched in time using the Runge–Kutta multistage time-stepping scheme while the values of the surface nodes are determined in a least-squares fashion. Surface boundary conditions are embedded in the computation of the flow variables on the surface of the body as described next.

2. *Implementation of Boundary Conditions*

Far-field boundary conditions are easily implemented in the same way as in any structured grid method. In the present work, the usual characteristic analysis based on one-dimensional Riemann invariants is used to determine the values of the inviscid flow variables. No special treatment is needed. The main challenge in using Cartesian grids is in the surface boundary condition implementation as the grids are not body aligned. A gridless method with least-squares technique is used in the present work to implement the surface boundary conditions. Surface node values are found in a similar least-squares fashion using a one-sided cloud of points. The boundary conditions are implemented such that in the process of solving the surface values the boundary conditions are satisfied.

For an inviscid condition the flow slips over the surface, and hence it is tangent to the surface. The slip condition is thus imposed on the boundary, and the other flow variables on the surface nodes are approximated as follows:

$$V_n = 0, \qquad \frac{\partial V_t}{\partial n} = 0, \qquad \frac{\partial \rho}{\partial n} = 0$$

$$\frac{\partial E}{\partial n} = 0, \qquad \frac{\partial p}{\partial n} = \frac{\rho V_t^2}{R} \tag{7}$$

where R is the local surface radius of curvature and the subscripts n and t stand for the normal and tangential directions, respectively, along the surface. Taking the curvature of the body into consideration will provide better accuracy. Strictly speaking, the condition $\partial V_t/\partial n = 0$ might not be consistent with the $\partial p/\partial n = \rho V_t^2/R$ condition in an irrotational flow. However, one must notice that only the pressure at wall is needed in a finite volume method for interior points. The $\partial V_t/\partial n = 0$ condition is used only to estimate the flow velocity at the wall, which is in turn used to calculate the pressure gradient at the wall. The error in the wall pressure itself will be second order with a first-order estimate of wall pressure gradient.

As shown in Fig. 4, the cloud for each surface node contains only one surface node, and the values of flow variables at the other cloud points are resolved to the tangential ξ and normal η directions of that surface node (here ξ and η are used in place of t and n.) This is for easy implementation of the boundary conditions because they are given in the ξ and η directions.

The variables are assumed to vary according to the function

$$f = b_0 + b_1\xi + b_2\eta + b_3\xi\eta \tag{8}$$

Taking V_t as an example and applying the Neumann condition in Eq. (7) on the surface node will result in the following:

$$\left(\frac{\partial V}{\partial \eta}\right)_{\text{bc}} = b_2 + b_3\xi_{\text{bc}} = 0, \qquad b_2 = -b_3\xi_{\text{bc}} \tag{9}$$

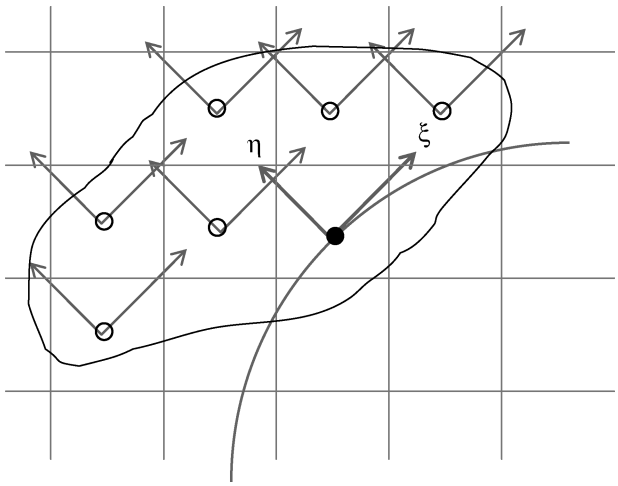


Fig. 4 Boundary implementation for surface nodes.

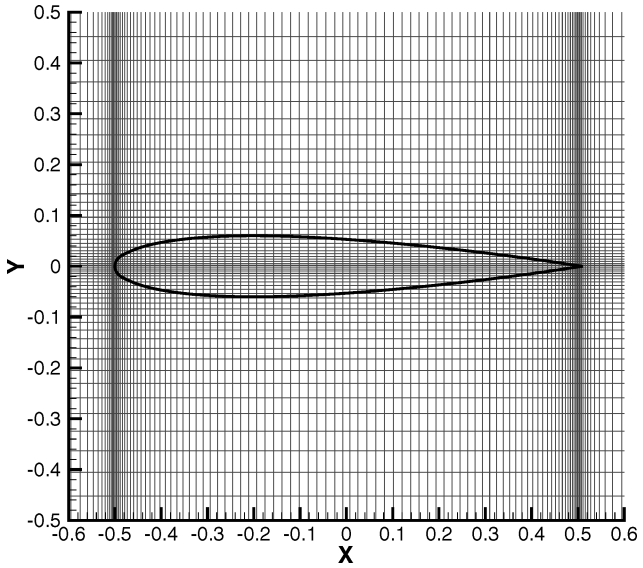


Fig. 5 Close-up view of a stretched grid for the NACA 0012 airfoil.

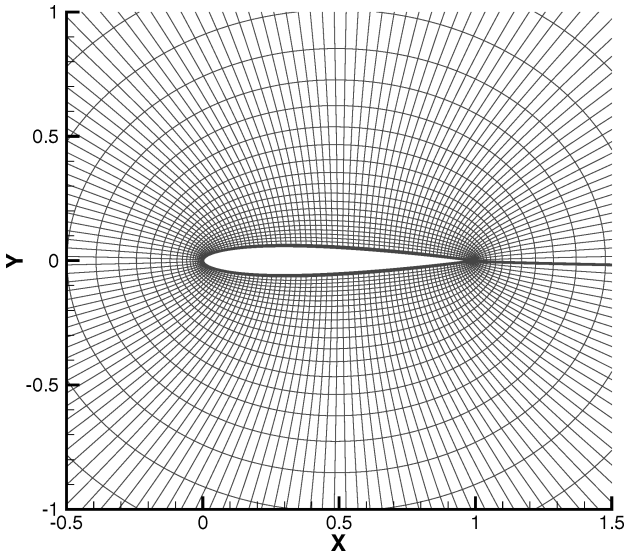


Fig. 6 Close-up of body-fitted grid with NACA 0012.

where the subscript bc denotes the value at the boundary point. Substituting Eq. (9) back into Eq. (8), a new function is obtained with the number of unknowns reduced by one:

$$V_t = b_0 + b_1\xi + b_3\eta(\xi - \xi_{\text{bc}}) \tag{10}$$

This is then solved using the least-squares method as described earlier to obtain b_0, b_1, and b_3, and the flow variable V_t at the surface node is thus defined. Because the boundary condition is incorporated into the system of equations to be solved, the value computed automatically satisfies the boundary condition in the least-squares sense. The other variables at the surface node are computed in a similar fashion.

III. Results and Discussion

Results are presented for two-dimensional airfoils using the method just described. To show the robustness of the method, a solution for a symmetric NACA 0012 airfoil and an asymmetric RAE 2822 airfoil are computed. All of the results are computed using stretched grids. A close-up view of a typical grid is shown in Fig. 5. A solution is first obtained on a coarse stretched grid with 144×130 cells. To improve the accuracy of the solution, the cells within the chord length are then refined to obtain a finer grid of 200×180. The coarse grid has 80 cells within the length of the airfoil, which results in approximately 160 surface nodes defined on the airfoil boundary. This resolution is deliberately chosen to be comparable to the reference body-fitted O-type grid (160×32) used in the solution by the FLO52 code. A close-up view of the body-fitted grid is shown in Fig. 6. The finer stretched Cartesian grid gives approximately 260 surface nodes defined on the boundary. Both the Cartesian grid and body-fitted grid extend to a distance of about 25 chords from the airfoil profile.

A. NACA 0012 Airfoil

For the symmetric airfoil, four flow cases with various combinations of freestream Mach number M_∞ and angle of attack α are computed: 1) $M_\infty = 0.5$, $\alpha = 3.0$ deg; 2) $M_\infty = 0.85$, $\alpha = 0$ deg; 3) $M_\infty = 0.8$, $\alpha = 1.25$ deg; and 4) $M_\infty = 2.0$, $\alpha = 0$ deg. All four cases were computed using a CFL number of 2.0 with local time stepping to accelerate convergence to steady state. For the present work, implicit residual smoothing has not been implemented because we are focused on demonstrating the potential of the gridless method for treating the boundary conditions. For each of the cases, the distributions of the pressure coefficient $C_p = (p - p_\infty)/\frac{1}{2}\rho_\infty V_\infty^2$ are plotted with the negative pressure upward, and the convergence histories show the mean rate of change of the density. All solutions are obtained using single-precision operations.

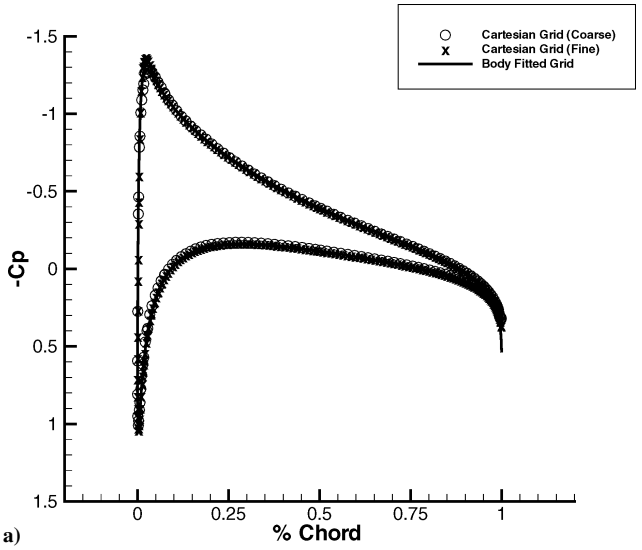
For case 1 with $M_\infty = 0.5$, $\alpha = 3.0$ deg, the C_p plot for the subsonic flow is presented in Fig. 7a and the convergence plot in Fig. 7b. From the C_p plot, the coarse grid solution is almost identical to the body-fitted grid solution. This shows that the Cartesian method just presented is able to provide an equally good solution for the subsonic case using the same number of cells as that in the body-fitted method. With the finer grid, the Cartesian method is able to resolve the peak at the leading edge better than the body-fitted grid. The convergence history for this case is shown in Fig. 7b.

Figure 8 shows the C_p plot for case 2 with $M_\infty = 0.85$, $\alpha = 0$ deg. The solutions for the three grids (coarse Cartesian grid, fine Cartesian grid, and body-fitted grid) are plotted on the same graph. Despite the fact that in the Cartesian method the edges are approximated with jagged cells, the coarse grid solution matches well with the body-fitted grid solution along the entire surface. There is a slight drop in the resolution of the shock wave by the present method on the coarse grid. This is caused by smearing near the shock region when the least-squares method is applied on the boundary and surface nodes. However, the finer Cartesian grid yields better resolution than that on the body-fitted grid. The Cartesian method uses four surface points to capture the shock.

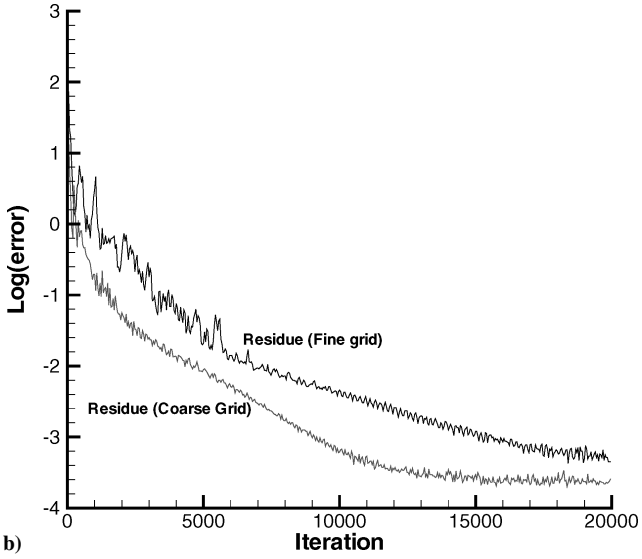
In case 3, the transonic flow of $M_\infty = 0.8$ with $\alpha = 1.25$ deg is computed, and the C_p is shown in Fig. 9a. Similar trends are observed for the C_p plot for cases 2 and 3. A weak shock is developed on the lower surface of the airfoil for case 3. The weak shock captured by the fine-grid solution appears to be sharper than that by the body-fitted grid method. Figure 9b shows the Mach-number contours on the fine Cartesian grid for this case. The shock wave in the flowfield is well resolved. Because the conventional finite volume method is used in the interior flowfield on a Cartesian grid, the scheme is strictly conservative. The preceding transonic cases demonstrate that this method gives the correct shock speed and strength as expected.

The convergence properties for the preceding transonic cases are similar. The convergence histories for case 3 on both the fine and coarse Cartesian grids are shown in Fig. 9c. The convergence for the transonic cases is slightly faster than that for subsonic computations, which is also typical of FLO52.

Supersonic flow of $M_\infty = 2.0$ with $\alpha = 0$ deg is tested in case 4, and the results are presented in Figs. 10a and 10b. The C_p plots for both the coarse and fine Cartesian grids match the body-fitted grid solution. No distinguishable difference is observed from the graph. However, a comparison of the contour plots for the fine Cartesian grid with the body-fitted grid reveals some differences. Close-up views of the Mach contours on the two grids are shown in Figs. 10c and 10d. Note that in the construction of the fine Cartesian grid, refinement of cells was only done for cells within the airfoil, and the intervals between the cells away from the airfoil remain the same



a)



b)

Fig. 7 Solution for NACA 0012 with $M_\infty = 0.5$, $\alpha = 3$ deg: a) C_p plot and b) convergence plot.

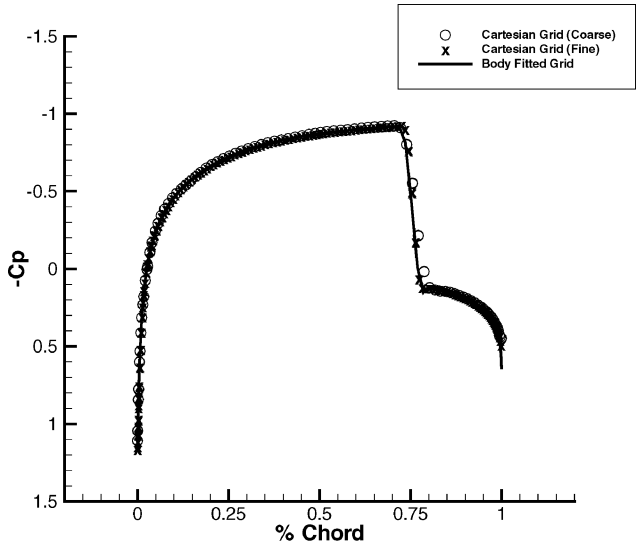
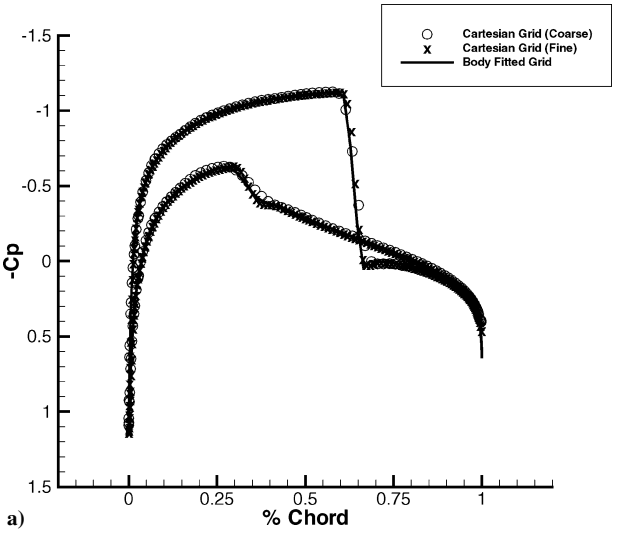
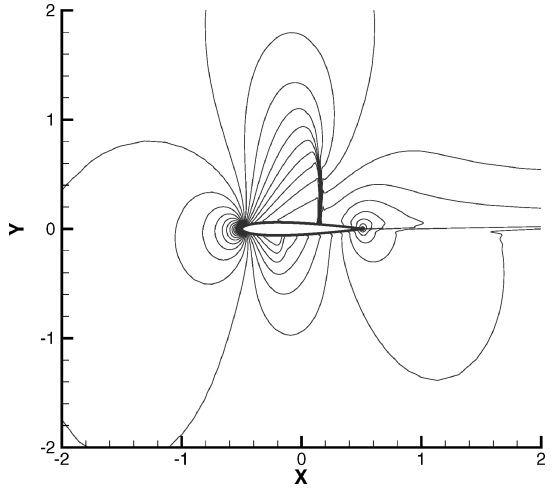


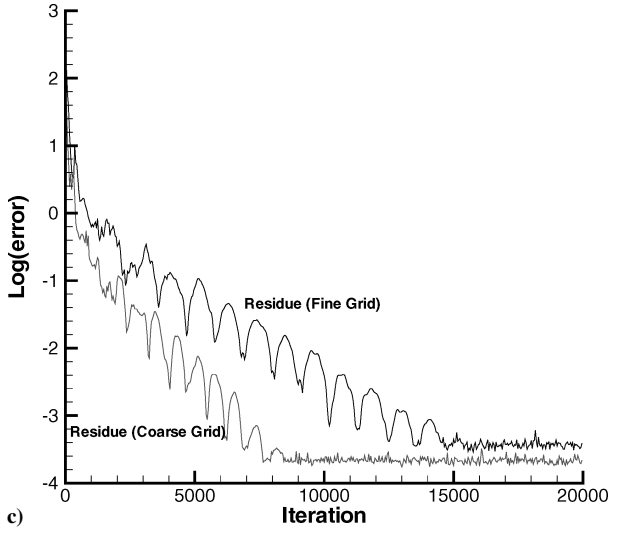
Fig. 8 Solution for NACA 0012 with $M_\infty = 0.85$, $\alpha = 0$ deg.



a)



b)

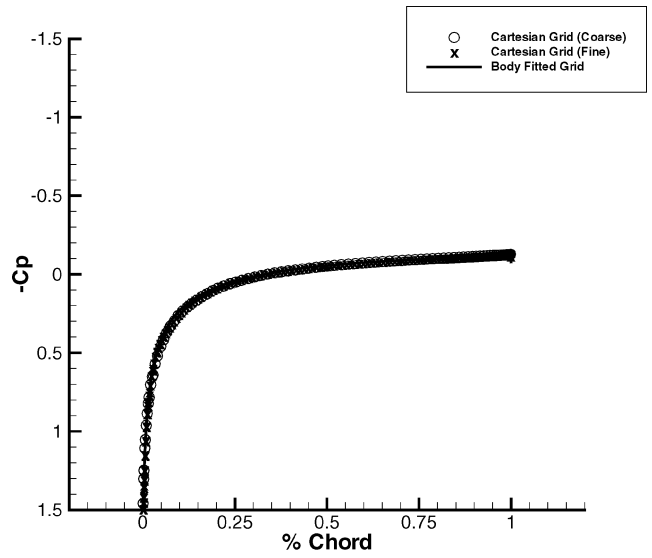


c)

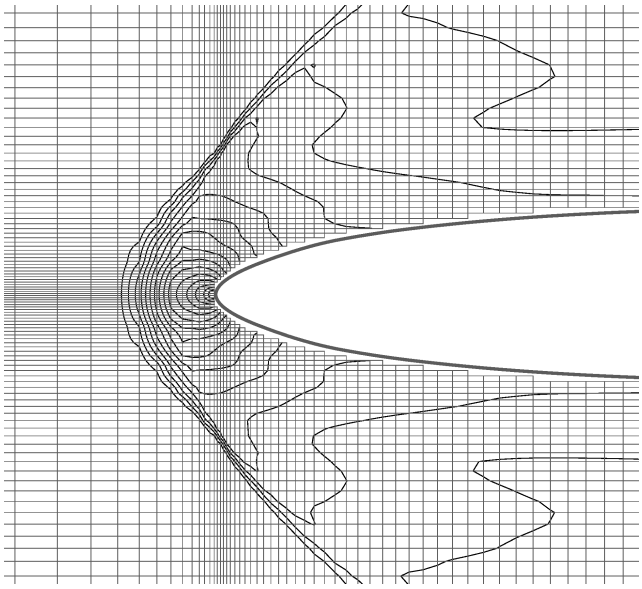
Fig. 9 Solution for NACA 0012 with $M_\infty = 0.8$, $\alpha = 1.25$ deg: a) C_p plot; b) Mach contour plot with contour $M_{\min} = 0.1$, $M_{\max} = 1.1$ and interval = 0.026; and c) convergence plot.

as those of the body-fitted grid. The solution obtained on the body-fitted grid offers slightly better resolution of the bow shock in the region immediately ahead of the airfoil nose. On the other hand, the solution obtained on the Cartesian grid offers sharper resolution of the slanted shoulders of the bow shock because of the higher grid densities of the Cartesian grid in those areas.

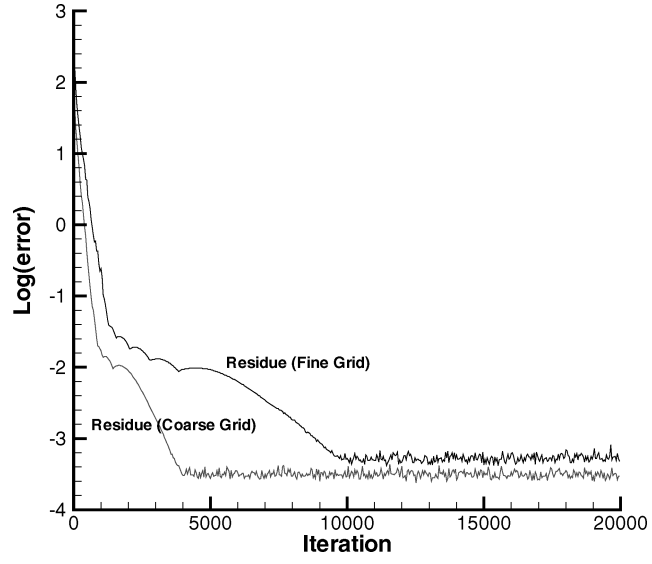
A further test (not shown here) is run to access the stability of the proposed Cartesian method by computing the flow at a high



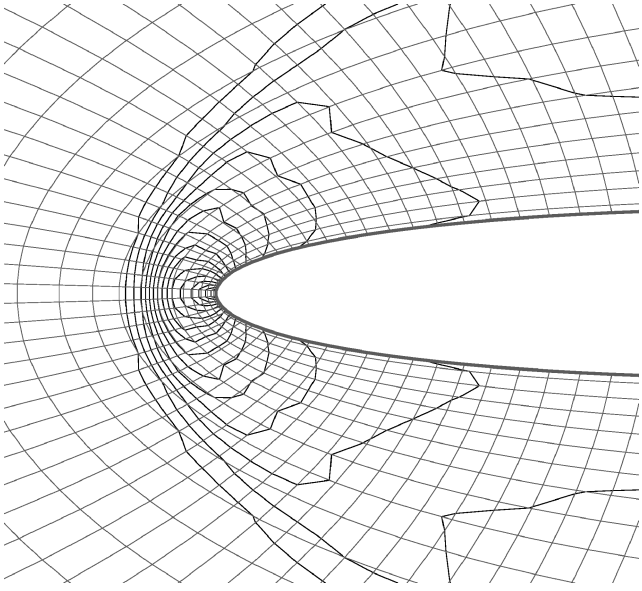
a)



c)



b)



d)

Fig. 10 Solution for NACA 0012 with M_∞ = 2.0, α = 0 deg: a) C_p plot; b) convergence plot; c) close-up Mach contour plot with contour $M_{\min}$ = 0.25, $M_{\max}$ = 1.9, and interval = 0.12; and d) close-up Mach contour plot using body-fitted grid with contour $M_{\min}$ = 0.25, $M_{\max}$ = 1.9, and interval = 0.12.

Mach number $M_\infty = 5.0$ with $\alpha = 0$ deg. A converged solution can be obtained with a lower CFL number of 0.9 but without having to change the artificial dissipation coefficient. This demonstrates the robustness of the present method for a wide Mach-number range.

B. RAE 2822 Airfoil

To further test the present method, subsonic and transonic flows over the RAE 2822 airfoil are calculated. The solution for the subsonic case is not shown here to limit the size of the paper. However, the behavior of the solution is very similar to that for the subsonic NACA 0012 case just shown. The solution obtained using the coarse Cartesian grid matches very closely to the body-fitted grid solution except at the peak location on the upper surface of the airfoil where it has a slightly lower value. Using the finer Cartesian grid, the C_p plot obtained matches exactly the body-fitted solution. At the peak location, it is able to predict a slightly higher value than the body-fitted solution.

Results for the transonic flow with $M_\infty = 0.75$ and $\alpha = 3$ deg are presented in Figs. 11a–11c. The coarse grid solution gives a slightly thicker shock on the upper surface. Second, the solution near the leading edge with high curvature shows a lower suction peak. The fine-grid solution on the other hand matches the body-fitted grid solution for both upper and lower surfaces of the airfoil. The Mach-number contours on the fine grid for this case are presented in Fig. 11b. Again it demonstrate the fine resolution of the shock wave although details of the pressure distribution shows that slightly finer grids on the airfoil surface are needed to obtain the same accuracy as the body-fitted grid for transonic flows. Figure 11c shows the convergence history for this case, which is similar to, and in fact faster in this case than, computations for the NACA 0012 case.

The present Cartesian grid code uses the same coding for the interior flow points as that of FLO52 for body-fitted curvilinear grids. Consequently, the computational time for both methods is comparable for the same mesh size. In fact, the gridless boundary

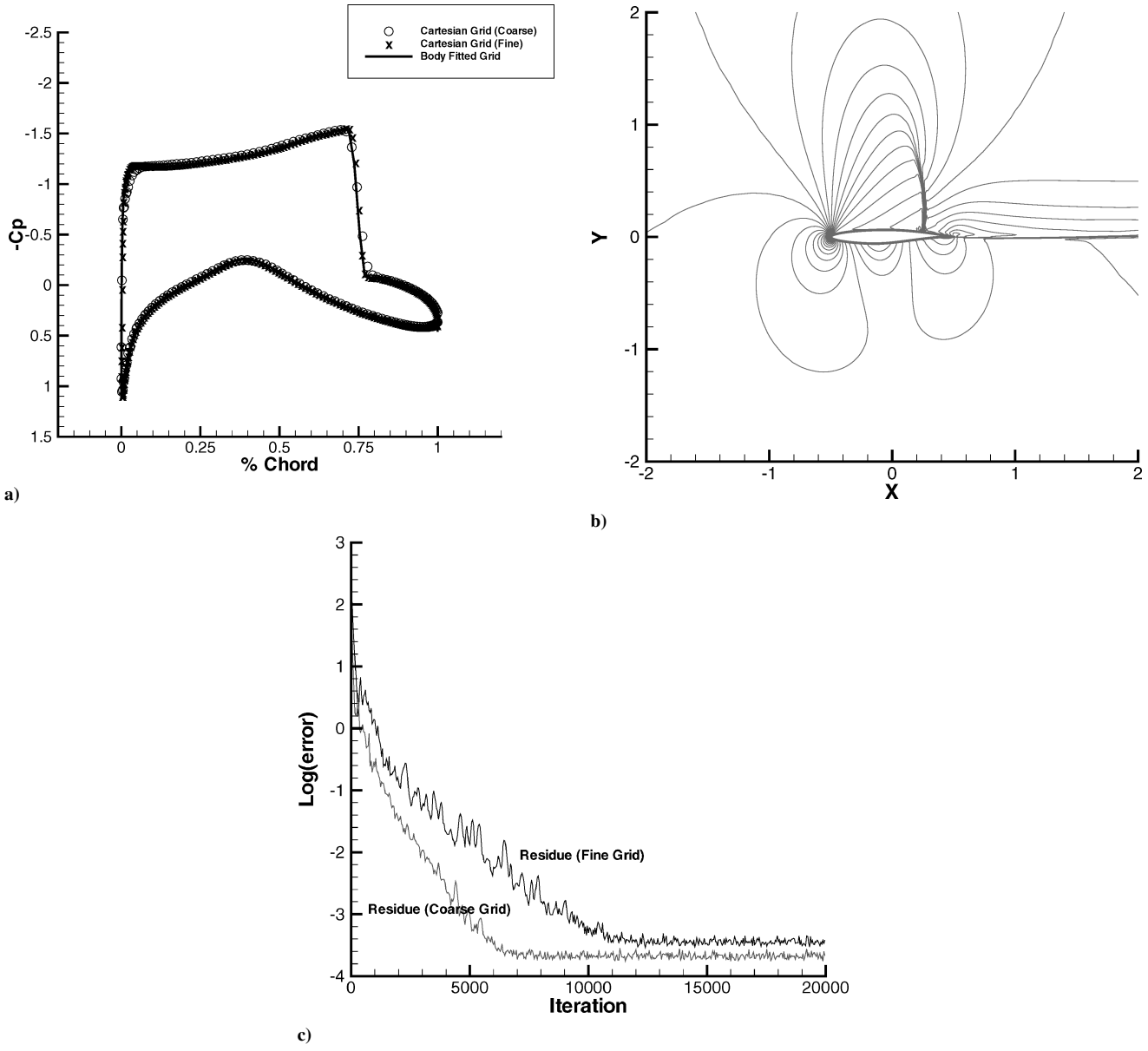


Fig. 11 Solution for RAE 2822 with M_∞ = 0.75, α = 3 deg: a) C_p plot; b) Mach contour plot with contour $M_{\min}$ = 0.1, $M_{\max}$ = 1.1, and interval = 0.025; and c) convergence plot.

treatment method takes a little longer because of the boundary condition treatments are more involved than those in the traditional body-fitted code. Computational efficiency of a Cartesian grid method will show up if the code is written to avoid the use of mapping the grids. This is not done here as the purpose of this paper is to prove the concept, and therefore the code still goes through a grid-mapping procedure even though it is not needed on the Cartesian grid. The use of the same code with grid mapping in this study, however, provides a direct comparison in terms of the accuracy of the new gridless boundary condition treatment method. The advantages of a Cartesian grid method go beyond computational efficiency only; it reduces the man-hours needed to generate a body-fitted mesh.

C. Three-Element Airfoil

To demonstrate the robustness and accuracy of the present method for complex configurations where the generation of straightforward body-fitted grids is difficult, we consider a multi-element airfoil for which an analytical solution is available. Suddhoo and Hall[33] provide an analytical incompressible potential flow solution for a three-element airfoil configuration at an angle of attack of 20 deg. To make a direct comparison with the analytical solution, the current Euler code is run at $M = 0.2$ on a Cartesian stretched grid of 270×320 shown in Fig. 12a. The Mach contours of the computed solution are shown in Fig. 12b. High-gradient flow structures are concentrated on the upper surface of the slat portion and on the leading edge of the airfoil. Figure 12c compares the distribution of surface-pressure coefficient over each element with the incompressible analytical solution by Suddhoo and Hall.[33] The computed result is in good agreement with the incompressible analytical solution except near the two suction peaks, where the Euler solution underpredicts the surface-pressure coefficient. The convergence plot is presented in Fig. 12d. The convergence rate for this test case is slower because of the low Mach number ($M = 0.2$) of the flow and high stretching and high aspect ratios of the grid, which might also have been responsible for the underprediction of the suction peaks on the airfoil. Use of locally refined embedded mesh as adopted in Ref. 29 would improve both convergence and accuracy of the computations. Nevertheless, this three-element test case demonstrates the flexibility offered by the Cartesian method with the present gridless boundary condition implementation.

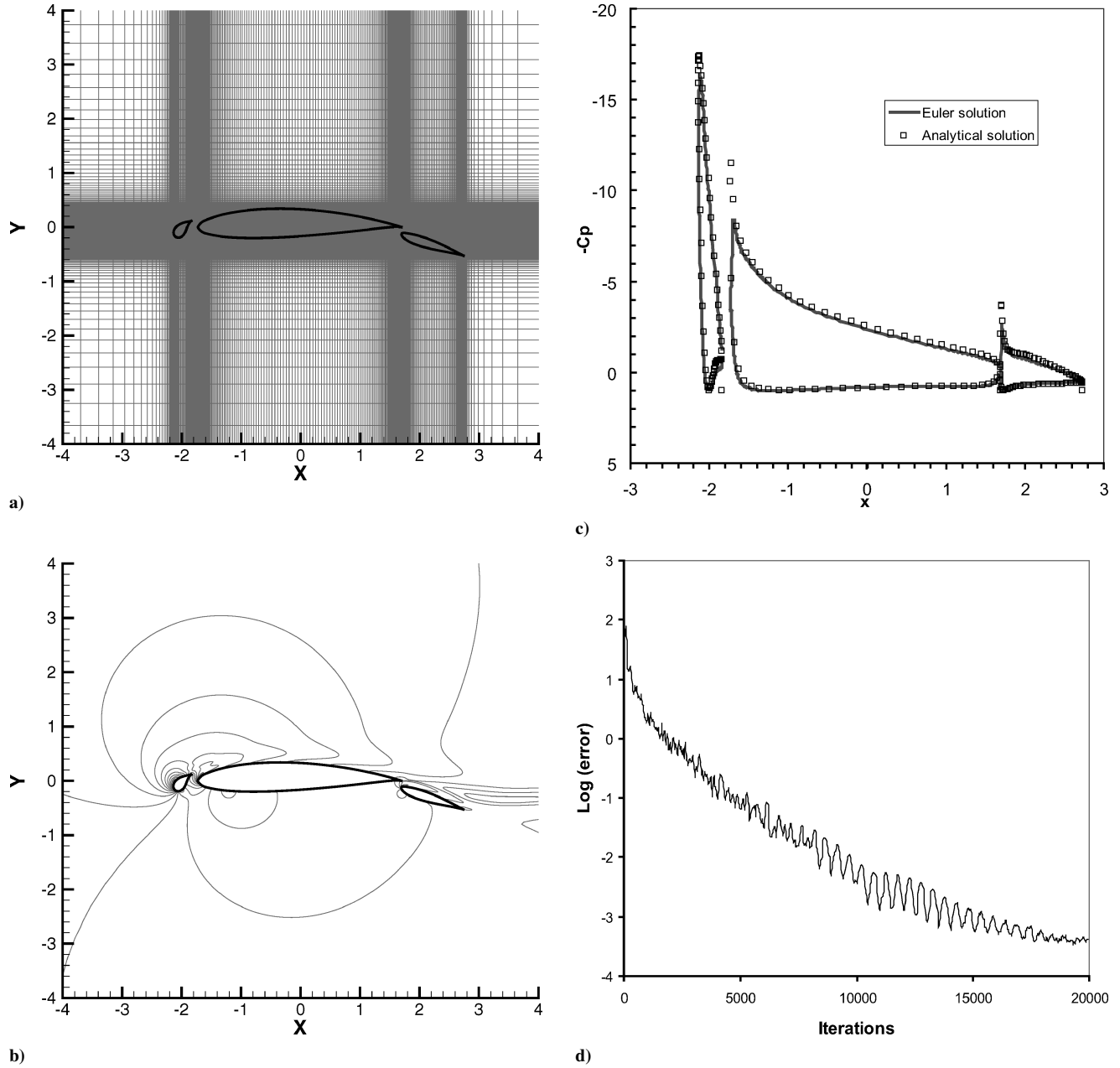


Fig. 12 Solution for three-element airfoil with $M_\infty = 0.2$, $\alpha = 20$ deg: a) close-up view of stretched grid; b) Mach contour plot with contour $M_{\min} = 0.048$, $M_{\max} = 0.72$, and interval = 0.048; c) C_p plot; and d) convergence plot.

IV. Conclusions

A method that combines the advantages of the efficiency, accuracy, and ease of grid generation of a Cartesian grid method and the flexibility in handling complex geometry of a gridless method while avoiding their shortcomings is presented. The approach uses the gridless or meshless method to address the boundary or interface while a standard structured grid method is used everywhere else. Boundary conditions are implemented on the geometry surface and are automatically satisfied as the surface values are solved in a least-squares fashion. The method is tested on the calculation of the subsonic, transonic, and supersonic flows over the NACA 0012 and RAE 2822 airfoils. The results are compared to those obtained by FLO52 using body-fitted curvilinear grids. It converges well for a range of Mach numbers tested including at high supersonic Mach numbers. The method is fully conservative in the flowfield because of the use of a conservative Cartesian method and therefore offers accurate resolution of shock waves although a slightly larger number of grid points might be needed to obtain the same resolution compared to a body-fitted grid method. In addition, the method is used to calculate the incompressible flow over a three-element airfoil configuration, and the result compares well with the exact solution obtained by conformal mapping. The flexibility in grid generation for complex geometries offers great potential for problems beyond isolated airfoils. The proposed gridless method for handling the boundary conditions can also be easily implemented in other basic flow solvers with non-body-fitted grids. Future work can explore embedded grids, multigrid, and extension to three dimensions.

Acknowledgments

The support of the Directorate of Research and Development, Defence Science and Technology Agency, Singapore (under POD 0001820, Program Monitors, Pek Beng Tit and Ng Keng Bee), is gratefully acknowledged. The authors thank Yeo Khoon Seng of the Mechanical and Production Engineering Department at the National University of Singapore and Cai Jinsheng of Temasek Laboratories for their suggestions and helpful discussions.

References

[1]Batina, J. T., "Unsteady Euler Algorithm with Unstructured Dynamic Mesh for Complex Aircraft Aerodynamic Analysis," *AIAA Journal*, Vol. 29, No. 3, 1991, pp. 327–333.

[2]Weatherill, N. P., Hassan, O., Marchant, M. J., and Marcum, D. L., "Adaptive Inviscid Flow Solutions for Aerospace Geometries Using Efficiently Generated Unstructured Tetrahedral Meshes," AIAA Paper 93-3390, 1993.

[3]Jameson, A., and Mavriplis, D., "Finite Volume Solution of the Two-Dimensional Euler Equations on a Regular Triangular Mesh," *AIAA Journal*, Vol. 24, No. 4, 1986, pp. 611–618.

[4]Frink, N. T., and Pirzadeh, S. Z., "Tetrahedral Finite-Volume Solutions to the Navier–Stokes Equations on Complex Configurations," 10th International Conf. on Finite Elements in Fluids, Jan. 1998.

[5]Meakin, R. L., "Automatic Off-Body Grid Generation for Domains of Arbitrary Size," AIAA Paper 2001-3105, June 2001.

[6]Liu, F., Cai, J., Zhu, Y., Wong, A. S. F., and Tsai, H. M., "Calculation of Wing Flutter by a Coupled Fluid-Structure Method," *Journal of Aircraft*, Vol. 38, No. 2, 2001, pp. 334–342.

[7]Meakin, R. L., "Object X-Rays for Cutting Holes in Composite Overset Structured Grids," AIAA Paper 2001-2537, June 2001.

[8]Coirier, W. J., and Powell, K. G., "An Accuracy Assessment of Cartesian Mesh Approaches for the Euler Equations," *Journal of Computational Physics*, Vol. 117, No. 1, 1995, pp. 121–131.

[9]Berger, M. J., and LeVeque, R. J., "An Adaptive Cartesian Mesh Algorithm for the Euler Equations in Arbitrary Geometries," AIAA Paper 89-1930, June 1989.

[10]Clarke, D. K., Salas, M. D., and Hassan, H. A., "Euler Calculations for Multielement Airfoils Using Cartesian Grids," *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 353–358.

[11]Epstein, B., Luntz, A., and Nachson, A., "Multigrid Euler Solver About Arbitrary Aircraft Configurations, with Cartesian Grids and Local Refinement," AIAA Paper 89-1960, June 1989.

[12]Tidd, D. M., Strash, D. J., Epstein, B., Luntz, A., Nachshon, A., and Rubin, T., "Multigrid Euler Calculations over Complete Aircraft," *Journal of Aircraft*, Vol. 29, No. 6, 1992, pp. 1080–1085.

[13]Wu, Z. N., "Anisotropic Cartesian Grid Method for Viscous Flow Computation," *Computational Fluid Dynamics Review 1998*, edited by M. Hafez and K. Oshima, World Scientific, 1998.

[14]Aftosmis, M. J., Berger, M. J., and Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries," AIAA Paper 2000-0808, Jan. 2000.

[15]Zeeuw, D. D., and Powell, K. G., "An Adaptively Refined Cartesian Mesh Solver for the Euler Equations," *Journal of Computational Physics*, Vol. 104, No. 1, 1993, pp. 56–68.

[16]Durbin, P. A., and Iaccarino, G., "An Approach to Local Refinement of Structured Grids," *Journal of Computational Physics*, Vol. 181, No. 2, 2002, pp. 639–653.

[17]Ham, F. E., Lien, F. S., and Strong, A. B., "A Cartesian Grid Method with Transient Anisotropic Adaptation," *Journal of Computational Physics*, Vol. 179, No. 2, 2002, pp. 469–494.

[18]Clarke, D. K., Salas, M. D., and Hassan, H. A., "Euler Calculations for Multi-Element Airfoils Using Cartesian Grids," *AIAA Journal*, Vol. 24, No. 3, 1986, pp. 353–358.

[19]Ye, T., Mittal, R., Udaykumar, H. S., and Shyy, W., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries," *Journal of Computational Physics*, Vol. 156, No. 2, 1999, pp. 209–240.

[20]"A Cartesian Multigrid Euler Code for Flow Around Arbitrary Configurations," MGAERO User's Manual, Ver. 3.1.4, Analytical Methods, Inc., Redmond, WA, 2001.

[21]Forrer, H., and Jeltsch, R., "A Higher-Order Boundary Treatment for Cartesian-Grid Methods," *Journal of Computational Physics*, Vol. 140, No. 2, 1998, pp. 259–277.

[22]Batina, J. T., "A Gridless Euler/Navier–Stokes Solution Algorithm for Complex-Aircraft Applications," AIAA Paper 93-0333, Jan. 1993.

[23]Batina, J. T., "A Gridless Euler/Navier–Stokes Solution Algorithm for Complex Two-Dimensional Applications," NASA TM-107631, June 1992.

[24]Sridar, D., and Balakrishnan, N., "An Upwind Finite Difference Scheme for Meshless Solvers," *Journal of Computational Physics*, Vol. 189, No. 1, 2003, pp. 1–29.

[25]Liu, J. L., and Su, S. J., "A Potentially Gridless Solution Method for the Euler/Navier–Stokes Equations," AIAA Paper 96-0526, Jan. 1996.

[26]Vanka, S. P., and Ploplys, N., "Meshless Methods for Navier–Stokes Equations Using Radial Basis Functions," *Advances in Computational Engineering and Sciences*, Vol. 2, ICES, 2000.

[27]Subrata, R., and Fleming, M., "Nonlinear Subgrid Embedded Element-Free-Galerkin Method for Monotone CFD Solutions," Proceedings of the 3rd ASME/JSME Joint Fluids Engineering Conf., FEDSM99-7023, July 1999.

[28]Belytschko, T., Lu, Y. Y., and Gu, L., "Element-Free Galerkin Methods," *International Journal of Numerical Methods in Engineering*, Vol. 37, No. 2, 1994, pp. 229–256.

[29]Kirshman, D. J., and Liu, F., "A Gridless Boundary Condition Method for the Solution of the Euler Equations on Embedded Cartesian Meshes with Multigrid," *Journal of Computational Physics*, Vol. 201, No. 1, 2004, pp. 119–147.

[30]Jameson, A., Schmidt, W., and Turkel, E., "Numerical Solutions of the Euler Equations By Finite Volume Methods Using Runge–Kutta Time-Stepping Schemes," AIAA Paper 81-1259, June 1981.

[31]Jameson, A., "Transonic Flow Calculation for Aircraft," *Numerical Methods in Fluid Dynamics*, edited by F. Brezzi, Vol. 1127, Lecture Notes in Mathematics, Springer-Verlag, 1985, pp. 156–242.

[32]Ray, T., and Tsai, H. M., "Some Issues in NURBS Representation of Airfoil Shapes for Optimization," 8th International Conf. on Numerical Grid Generation in Computational Field Simulations, June 2002.

[33]Suddhoo, A., and Hall, I. M., "Test Cases for the Plane Potential Flow past Multi-Element Aerofoils," *Aeronautical Journal*, Vol. 89, Dec. 1985, pp. 403–414.

S. Mahalingam
Associate Editor