AIAA Paper-2002-3285, presented at the 32nd AIAA Fluid Dynamics Conference, June 23-26, 2002, St. Louis, MO.

GRIDLESSS BOUNDARY CONDITION TREATMENT FOR A NON-BODY-CONFORMING MESH

D. J. Kirshman* and F. Liu†

Department of Mechanical & Aerospace Engineering
University of California, Irvine

Abstract

A numerical method for the treatment of boundary conditions in non-body conforming computational grids is proposed. The method incorporates a "gridless" algorithm in which numerical boundary conditions may be imposed in such a manner that explicit connectivity between surface geometry and field mesh is not required. As a result, solutions to problems in computational physics are greatly facilitated since definition of the body geometry may be generated independently from the field mesh. Furthermore, perturbations in body geometry may be introduced into the simulation without the requirement for regeneration of the field mesh (i.e. wing flap deflection). The method is particularly suited for Cartesian mesh applications, in which convergence and stability issues associated with mesh skewness and distortion are eliminated. Furthermore, the method does not require special treatment for "thin body" problems from which many Cartesian methodologies suffer. The applicability of the method for two-dimensional Euler simulations with examples of transonic flow over a NACA 0012 airfoil is presented here. Results of the method are shown to compare very favorably with traditional body-fitted mesh techniques.

Introduction

The most labor intensive aspect of a computational fluid dynamics (CFD) simulation is the generation of a quality discretization mesh (or grid). Namely, the mesh must in some way allow for the representation of the various computational boundaries, particularly those between a body and surrounding fluid. Traditional strategies incorporate either a "structured" or "unstructured" mesh, or some combination of the two in a "multi-block" arrangement. In a structured grid, each grid point is uniquely defined by a set of two independent parameters (or three in 3D simulations). The grid is formed by an array of quadrilateral cells (or hexahedral in 3D), which may have curved boundaries in physical space, and may be stretched along any curvilinear coordinate. When considering flows about arbitrary body geometries, the nature of the structured mesh often results in highly distorted or skewed cells in some regions of the domain, which detracts significantly from accuracy, as well as convergence and stability of the numerical simulation. Additionally, some mappings between physical and computational space result in singularities that require special treatment when finite difference schemes are considered. In most industrial applications, geometries may be so complex that generation of a single structured mesh within the computational domain is not feasible. In such situations, the domain may be decomposed into several "blocks" which can be more easily meshed individually. However, a multi-block arrangement can also lead to inaccuracies should the grids be significantly mismatched at block interfaces.

A second approach is the "unstructured" grid, which may be composed of both quadrilaterals and triangles (or hexahedra, tetrahedra, prisms, and pyramids in 3D). Unstructured grids have the advantage of increased flexibility for meshing around complex geometries. However, unstructured grids have several disadvantages such as increased computational time and significantly more memory requirements when compared to a structured mesh of the same resolution. Furthermore, mesh quality is not easily controlled, and alignment of the mesh with anticipated flow direction is certainly problematic. Another drawback is that the unstructured mesh is far less suitable for wave propagation phenomena important in many aspects of computational physics, particularly compressible fluid mechanics.

A third approach currently under consideration by many investigators is that of the Cartesian mesh. In this approach, all the advantages of a structured mesh are retained without issues related to skewness and distortion. A recurring issue with Cartesian methodologies, however, is the ability to treat "thin body" regions such as airfoils, turning vanes, or fins, in which the body geometry is thinner than the local cell

* Graduate Student, Member AIAA
† Associate Professor, Senior Member AIAA

size.[1] In this paper, a boundary treatment for non-body conforming grids is presented in which this problem is alleviated.

Cartesian Grid Methodologies

Early investigations on the applicability of Cartesian grids for compressible fluid dynamic analyses about arbitrary bodies considered the full potential equation[2,3]. Clarke, Salas and Hassan[4], who extended the potential flow methods of Wedan and South[3], considered Euler flow calculations for multi-element airfoils using Cartesian grids. They incorporated a finite volume procedure in which fractional cells are formed at the intersection of the mesh and the body geometry.

Though, for conformal meshes, finite volume methods have an advantage over finite difference schemes in that discretization on arbitrary mesh geometries can be performed without metric coefficients (i.e. conformal transformations), the introduction of partially cut cells near the boundary (for Cartesian meshes) presents considerable issues for convergence and stability[5]. In the work of Clarke[4], et. al., this issue was alleviated by merging small cells adjacent to the body (25 to 50% of the Cartesian cell size) into adjacent cells, though complicating the surface cell geometry and treatment. In an attempt to avoid the cut cell issue, Epstein, Luntz, and Nachshon[6] performed transonic full potential solutions using a finite difference formulation in conjunction with multigrid acceleration. Later, collaborating with Tidd and Strash[7], they extended the method to Euler solutions. In their method, however, the pressure on surface is determined in a very coarse, and complicated manner, in which a considerably fine grid is required in order to obtain reasonable accuracy, especially in regions of high curvature on the body.

More contemporary efforts to employ Cartesian grids for arbitrary body configurations incorporate complicated methodologies to address either the cut cell problem for finite volume discretizations, or difficulties associated with finite difference discretizations, while attempting to increase the accuracy of the boundary treatment. Though many of these treatments are very effective, in most instances the authors recognize some difficulty in applying their method to "thin" surfaces, and special treatment is required, especially methods in which "ghost" cells are incorporated. For instance, the recent work of Dadone and Grossman[8,9] must employ multi-valued nodes in order to satisfy the non-penetration condition at the wall.

Other investigators incorporate surface boundary conditions by means of a rotated Reimann problem in which flux contributions are evaluated considering reflected conditions across the wall. [10,11,12, 13,15] Typically, these methods rely on grid refinement to alleviate thin geometry issues, resulting in extremely fine grid cells. Other effective, though complicated, strategies include adaptive cell-splitting techniques in which sub-cells are created ensuring that only one side of a thin geometry occupies a given grid cell.[1,14]

Gridless Boundary Treatment

The most significant drawback of the above methods is that intricacies of the body geometry, such as the thin trailing edge of an airfoil, are difficult to capture on a Cartesian grid unless the feature falls exactly on a grid point. In this work, a method is proposed which alleviates this problem by incorporating, when necessary, additional surface nodes beyond those formed by the intersection of the body geometry and Cartesian mesh. Furthermore, additional "cloud" nodes unassociated with either the surface or field mesh may also be introduced in an adaptive fashion to increase the accuracy of the boundary treatment. Namely, a "gridless" approach is implemented by which surface properties and necessary gradients are evaluated based on weighted least squares approximation of the flow properties at a "cloud" of nodes in the vicinity of the surface location in question. Though nodes associated with the field grid are incorporated into the boundary treatment, their geometric connectivity with the field is not retained for purposes of obtaining surface properties. Increased order of accuracy at the boundary can be obtained by using high order curve fits. Batina[16] and other researchers[17,18] have proposed a purely gridless scheme for the entire computational domain. However, global conservation of mass, momentum, and energy for these schemes are not necessarily ensured.

In this paper, the feasibility of a gridless approach for treating surface boundary conditions is demonstrated. The method is shown to agree favorably with body-fitted methods (FLO52 code) in regions of high surface curvature (i.e. nose of an airfoil) on a reasonably coarse grid (in direction of curvature) without mesh refinement. Efforts to incorporate cloud and field node adaption are currently under investigation in order to fully exploit the merits of the method.

The following discussion presents the numerical implementation of the gridless boundary treatment. Here, the implementation of the field solution is achieved using Van Leer flux vector splitting, though the boundary treatment could be incorporated into any numerical scheme for the solution of the field. It should be noted that, although the presence of thin geometry does not pose any special issue for the method, the trailing edge tip itself is problematic since a finite difference formulation is proposed here, and the tip of an airfoil represents a singularity (in any solver) in which surface radius vanishes, and the surface normal is not well defined.

Nonetheless, the method is able to handle this discontinuity, as discussed below.

Governing Equations

In this study, the two-dimensional flow of an inviscid, compressible gas is considered. Such flows are governed by the Euler equations, which provide for the conservation of mass, momentum, and energy, and are given by

$$\frac{\partial \mathbf{u}}{\partial t}+\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{g}}{\partial y}=0 \tag{1}$$

where t is time, x and y are the Cartesian coordinates, and

$$\mathbf{u}=\begin{bmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{bmatrix}\quad \mathbf{f}=\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho uv\\ \rho uH\end{bmatrix}\quad \mathbf{g}=\begin{bmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho vH\end{bmatrix} \tag{2}$$

where ρ is the density, u is the x-velocity component, v is the y-velocity component, p is the pressure, E is the total energy, and H is the total enthalpy. For an ideal gas, the total energy and total enthalpy can be written

$$E=\frac{p}{\rho(\gamma-1)}+\frac{1}{2}\left(u^2+v^2\right) \tag{3}$$

$$H=E+\frac{p}{\rho} \tag{4}$$

where γ is the ratio of specific heats. Additionally, the equation of state is given by

$$p=\rho\overline{R}T \tag{5}$$

where T is the static temperature and $\overline{R}$ is the ideal gas constant.

Numerical Algorithm

Field Discretization

The entire flow domain is discretized using a purely Cartesian mesh, which is generated independently of the body. In order to solve for the flow field, a finite difference scheme using Van Leer flux vector splitting is performed[19]. In this scheme, the convective flux vectors **f** and **g** are decomposed into "upwind" and "downwind" components based on the sign of the eigenvalues of the system of governing equations. Namely, at each grid point, the convective flux vectors are written as

$$\begin{aligned}\mathbf{f}&=\mathbf{f}^+ +\mathbf{f}^-\\ \mathbf{g}&=\mathbf{g}^+ +\mathbf{g}^-\end{aligned} \tag{6}$$

where the positive and negative superscripts indicate the portion of the total flux that travels in the positive and negative coordinate directions, respectively. For supersonic flow, there is no splitting and the entire flux quantity travels in the downstream direction. For subsonic flow (in a given coordinate direction), the flux vectors are given by

$$\mathbf{f}^{\pm}=\pm\frac{\rho}{4c(u\pm c)^2}\begin{bmatrix}1\\ \frac{1}{\gamma}\left[u(\gamma-1)\pm 2c\right]\\ v\\ \frac{\left[(\gamma-1)u\pm 2c\right]^2}{(\gamma^2-1)}+\frac{u^2+v^2}{2}\end{bmatrix} \tag{7}$$

$$\mathbf{g}^{\pm}=\pm\frac{\rho}{4c(v\pm c)^2}\begin{bmatrix}1\\ u\\ \frac{1}{\gamma}\left[v(\gamma-1)\pm 2c\right]\\ \frac{\left[(\gamma-1)v\pm 2c\right]^2}{(\gamma^2-1)}+\frac{u^2+v^2}{2}\end{bmatrix} \tag{8}$$

where c is the acoustic speed. The governing equations are then re-written as

$$\frac{\partial \mathbf{u}}{\partial t}+\frac{\partial \mathbf{f}^+}{\partial x}+\frac{\partial \mathbf{f}^-}{\partial x}+\frac{\partial \mathbf{g}^+}{\partial y}+\frac{\partial \mathbf{g}^-}{\partial y}=0 \tag{9}$$

which can be descretized into a finite difference formulation. Considering the grid location defined by the nodal indices i and j in the x and y directions respectively, a three point second order upwind discretization is written

$$\begin{aligned}\frac{\partial \mathbf{f}^+}{\partial x}&=\frac{1}{2\Delta x}\left(3\mathbf{f}_i^+-4\mathbf{f}_{i-1}^+ +\mathbf{f}_{i-2}^+\right)\\ \frac{\partial \mathbf{f}^-}{\partial x}&=\frac{1}{2\Delta x}\left(-3\mathbf{f}_i^-+4\mathbf{f}_{i+1}^- -\mathbf{f}_{i+2}^-\right)\end{aligned} \tag{10}$$

Generalizing to two dimensions, the semi discrete form of Equation (9) can be written in terms of a first order difference and a limited second order correction as

$$\begin{aligned}\frac{d\mathbf{u}_{i,j}}{dt}=&\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^+_{i+\frac{1}{2}}\right)\right)\delta_x^+\mathbf{f}^+_{i,j}-\Psi\left(r^+_{i-\frac{3}{2}}\right)\delta_x^+\mathbf{f}^+_{i-1,j}\right]\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^-_{i-\frac{1}{2}}\right)\right)\delta_x^-\mathbf{f}^-_{i,j}-\Psi\left(r^-_{i+\frac{3}{2}}\right)\delta_x^-\mathbf{f}^-_{i+1,j}\right]\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^+_{j+\frac{1}{2}}\right)\right)\delta_y^+\mathbf{g}^+_{i,j}-\Psi\left(r^+_{j-\frac{3}{2}}\right)\delta_y^+\mathbf{g}^+_{i,j-1}\right]\\ &-\frac{1}{2}\left[\left(2+\Psi\left(r^-_{j-\frac{1}{2}}\right)\right)\delta_y^-\mathbf{g}^-_{i,j}-\Psi\left(r^-_{j+\frac{3}{2}}\right)\delta_y^-\mathbf{g}^-_{i,j+1}\right]\end{aligned} \tag{11}$$

where the δ are first order upwind fluxes

$$\delta_x^+ f^+_{i,j}=\frac{f^+_{ij}-f^+_{i-1,j}}{\Delta x}\qquad \delta_x^- f^-_{i,j}=\frac{f^-_{i+1,j}-f^-_{ij}}{\Delta x} \tag{12}$$

and similarly for the y direction. The flux limiter, $\Psi(r)$ in the above is implemented to suppress oscillations of the second order solution in high gradient regions (i.e shocks). In the results presented here, the standard minmod limiter is incorporated in which

$$\Psi(r)=\begin{cases}1 & r\geq 1\\ r & 0\leq r\leq 1\\ 0 & r<0\end{cases} \qquad (13)$$

where

$$r^{+}_{i+\frac{1}{2}}=\frac{f^{+}_{i+2,j}-f^{+}_{i+1,j}}{f^{+}_{i+1,j}-f^{+}_{i,j}} \qquad r^{-}_{i+\frac{1}{2}}=\frac{f^{-}_{i,j}-f^{-}_{i-1,j}}{f^{-}_{i+1,j}-f^{-}_{i,j}}$$
$$r^{+}_{j+\frac{1}{2}}=\frac{g^{+}_{i,j+2}-g^{+}_{i,j+1}}{g^{+}_{i,j+1}-g^{+}_{i,j}} \qquad r^{-}_{j+\frac{1}{2}}=\frac{g^{-}_{i,j}-g^{-}_{i,j-1}}{g^{-}_{i,j+1}-g^{-}_{i,j}} \qquad (14)$$

"Gridless" Node Definition

In order to treat the region of the flow field in the vicinity of the body, the locations where the surface geometry intersects the Cartesian grid are first evaluated. To facilitate this process, the body is described by a series of surface definition nodes through which a spline curve fit is created. Once the intersection of the body with the Cartesian grid has been determined, the field nodes are then categorized into three types: those far from the body which can be treated as field points using the finite difference expression, Equation (11); those embedded within the body, or located very close to the surface, which are removed from the calculation; and those near the body which do not have a complete computational stencil to apply Equation (11). This last category of nodes is to be solved with the gridless method using least squares approximation. Also, nodes associated with the body geometry are treated using the gridless method. Namely, the surface nodes formed by the intersection of the body and field mesh, as well any surface definition (spline) nodes that may be desired to ensure that intricacies in the body geometry not captured by the intersection nodes are retained in the solution. Finally, adaptive "cloud" nodes may be introduced into the solution to facilitate the least squares approximation and enhance robustness and accuracy of the simulation. Figure 1 provides a representation of the various types of nodes described.

Derivation of Gridless Shape Functions

Implementation of the gridless boundary method requires the definition of nodal shape functions. Namely, the effect of each node in the vicinity of a particular location is weighted by the value of its shape function. These shape functions are determined using a weighted least squares approximation in each of the coordinate directions. Flow properties at a particular location can then be evaluated by the sum of the products of the shape function and flow properties at the surrounding nodes. Derivation of these shape functions follows. Belytschko et al[20] present a similar derivation for application to structures analysis using element-free Galerkin methods.

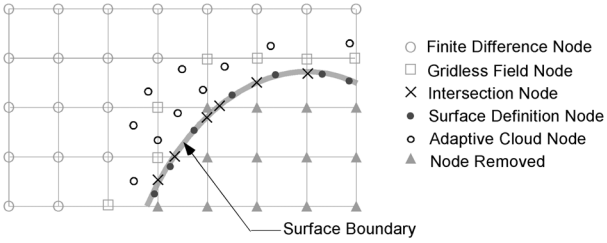

Figure 1 Nodal Assignments for Gridless Method

In order to evaluate the variation in flow field properties at any *p*-th gridless node, a "cloud" of *N* nodes in the vicinity are judiciously chosen so that a weighted least squares approximation can be made. In order to establish the approximation, a polynomial basis is first chosen. Thus, any particular component of a conserved flux variable, $f(x,y)$, can by approximated by the least squares interpolant $f^h(x,y)$ of *M* terms where

$$f^h(x,y)=\sum_{m=1}^{M} p_m(x,y)a_m=\mathbf{p}^{\mathrm{T}}(x,y)\mathbf{a} \qquad (15)$$

where each p_m is a monomial in the space coordinates providing an *M* term basis vector given by

$$\mathbf{p}^{\mathrm{T}}(x,y)=\left[1,x,y,xy,x^2,y^2\ldots\right] \qquad (16)$$

and **a** is the vector of polynomial coefficients. The basis vector **p** can have as many terms as desired. Increasing the number of terms yields higher order approximations at the expense of an increased number of cloud nodes required to perform the least squares fit, as well as an increased computational effort. At a minimum, one must have as many cloud nodes as there are terms in **p**. It is also required that the points not be collinear, in which case the least squares matrix becomes singular. Increasing the number of cloud nodes beyond that of terms in **p** yields lower order resolution as well as diffusion into the solution. Additionally, numerical experiments have indicated that using additional nodes (beyond the order of the basis) can lead to spurious results at the leading edge of an airfoil where gradients are high.

A weighted least squares interpolant is found at the location of any *p*-th node by minimizing

$$I=\sum_{n=1}^{N} w(x_n-x_p,y_n-y_p)\left[\mathbf{p}^{\mathrm{T}}(x_{n,}y_n)\mathbf{a}-f_n\right]^2 \qquad (17)$$

where *N* is the number of nodes in the cloud of points considered, and f_n is the value of the flux variable at node *n*. The weight function, *w*, is arbitrary and should decrease in magnitude with the relative distance between the location of the node of interest (x_p,y_p) and the location of any given cloud node(x_n,y_n). For *w* of unity, one retains a standard least squares formulation.

The stationarity of *I* with respect to the coefficient vector **a** leads to a set of *M* equations given by

$$\frac{\partial I}{\partial a_m} = \sum_{n=1}^{N} w_n \left[\mathbf{p}^{\mathrm{T}}(x_n, y_n)\mathbf{a} - f_n\right] p_m(x_n, y_n) = 0 \quad (18)$$

where $w_n \equiv w(x_n - x_p, y_n - y_p)$. The solution for the coefficient vector is then,

$$\mathbf{a} = \mathbf{A}^{-1}\mathbf{B}\mathbf{f} \quad (19)$$

where

$$\mathbf{a}^{\mathrm{T}} = [a_1 \quad \cdots \quad a_M]$$

$$\mathbf{A} = \sum_{n=1}^{N} w_n \mathbf{p}^{\mathrm{T}}(x_n, y_n)\mathbf{p}(x_n, y_n) \quad (20)$$

$$\mathbf{B} = [w_1\mathbf{p}(x_1, y_1) \quad \cdots \quad w_N\mathbf{p}(x_N, y_N)]$$

$$\mathbf{f}^{\mathrm{T}} = [f_1 \quad \cdots \quad f_N]$$

The interpolant can then be written at location (x_p,y_p) as

$$\begin{aligned} f^h(x_p, y_p) &= \mathbf{p}^{\mathrm{T}}(x_p, y_p)\mathbf{a} = \mathbf{p}^{\mathrm{T}}(x_p, y_p)\mathbf{A}^{-1}\mathbf{B}\mathbf{f} \\ &= \sum_{n=1}^{N}\sum_{m=1}^{M} p_m(x_p, y_p)\left[\mathbf{A}^{-1}\mathbf{B}\right]_{mn} f_n \end{aligned} \quad (21)$$

or in terms of a shape function ϕ as

$$\begin{aligned} f^h(x_p, y_p) &= \sum_{n=1}^{N} \phi_n(x_p, y_p) f_n \\ \phi_n(x_p, y_p) &\equiv \sum_{m=1}^{M} p_m(x_p, y_p)\left[\mathbf{A}^{-1}\mathbf{B}\right]_{mn} \end{aligned} \quad (22)$$

The spatial derivative of the flux in the *k*-th direction can also be written in terms of a shape function as

$$\begin{aligned} \frac{\partial f^h(x_p, y_p)}{\partial x_k} &= \sum_{n=1}^{N} \frac{\partial \phi_n(x_p, y_p)}{\partial x_k} f_n \\ \frac{\partial \phi_n(x_p, y_p)}{\partial x_k} &= \sum_{m=1}^{M} \frac{\partial p_m(x_p, y_p)}{\partial x_k}\left[\mathbf{A}^{-1}\mathbf{B}\right]_{mn} \end{aligned} \quad (23)$$

It is pointed out that the shape functions have the following property

$$\sum_{n=1}^{N} \phi_n = 1; \quad \sum_{n=1}^{N} \frac{\partial \phi_n}{\partial x_k} = 0 \quad (24)$$

When deriving shape functions with higher order polynomials, a trial and error period is typically required in which various combinations of fitted nodes are selected and the minimum pivot resulting from the inversion of matrix **A** should be as large as possible to ensure a convergent solution.

It is noted that the "weighted" least squares method described above provides for dominance in the effect of nodes located close to the point of interest through the implementation of a weight function *w*. In the results provided here, a weight function of unity was considered (i.e. standard least squares fitting). Effects of other weight function formulation are under consideration, but numerical experiments to date have not indicated more a suitable choice.

Gridless Field and Adaptive Cloud Node Treatment

With the above specification of least squares shape functions, the time variation of flow properties at the gridless field nodes or adaptive cloud nodes can be obtained using a finite difference formulation. Returning to the conservation statement given by Equation (9), the variation in the conserved variable vector **u** results a total of four fluxes, one from each of the coordinate directions. Depending on the location of the gridless node with respect to the Cartesian mesh, some of the flux contributions can be obtained from components of the field finite difference expression, Equation (11). Namely, these are directions in which there exists a Cartesian field node located within the flow domain (i.e., typically in directions away from the body) in the appropriate flux direction. For flux components resulting from directions toward the body, a cloud of nodes is chosen, and the contribution of the flux is specified using the gridless technique. Thus, flux contributions for directions that require gridless treatment are written

$$\begin{aligned} \frac{\partial \mathbf{f}^{\pm}}{\partial x} &= \sum_{n=1}^{N_x^{\pm}} \frac{\partial \phi_n^{\pm}}{\partial x}\mathbf{f}_n^{\pm} \\ \frac{\partial \mathbf{g}^{\pm}}{\partial y} &= \sum_{n=1}^{N_y^{\pm}} \frac{\partial \phi_n^{\pm}}{\partial y}\mathbf{g}_n^{\pm} \end{aligned} \quad (25)$$

where the ± superscripts on the shape function indicate that, in selecting the *N* cloud nodes, only points in the appropriate flux direction are chosen. That is, for positive flux contributions, only nodes located in negative direction along the appropriate coordinate are considered.

Surface Node Treatment

For the nodes located on the body surface (either from geometry specification or those resulting from intersection with the Cartesian mesh) all flux contributions are derived using the gridless technique. However, two variations for the surface node treatment are available: time marching treatment, and surface gradient treatment. In either case, a local wall coordinate system is established in which the positive $\tilde{y}$ direction is oriented along the surface normal, pointing into the flow field.

Time Marching Boundary Method

For the time marching treatment, flux components are evaluated using the gridless technique in the local coordinate system for the positive $\tilde{x}$, negative $\tilde{x}$, and positive $\tilde{y}$ directions. Since the negative $\tilde{y}$ direction is directed into the body surface, there is no flux contribution from that direction. Thus, in semi-discrete form, the time variation of flow properties at the surface are written

$$\frac{d\tilde{\mathbf{u}}}{dt} = -\sum_{n=1}^{N_{\tilde{x}}^{+}} \frac{\partial \phi_n^{+}}{\partial \tilde{x}}\tilde{\mathbf{f}}_n^{+} - \sum_{n=1}^{N_{\tilde{x}}^{-}} \frac{\partial \phi_n^{-}}{\partial \tilde{x}}\tilde{\mathbf{f}}_n^{-} - \sum_{n=1}^{N_{\tilde{y}}^{-}} \frac{\partial \phi_n^{-}}{\partial \tilde{y}}\tilde{\mathbf{g}}_n^{-} \quad (26)$$

where the tilde indicates that the flux vectors have been rotated into the local wall coordinate system by appropriately adjusting the velocity components. In the local wall coordinate system, flow tangency requires that the velocity normal to wall, *v*, vanish. Thus, the above equation is only integrated for 3 components of the local conserved variable vector, while the normal component of velocity, $\tilde{v}$, is set to zero. The conserved variable vector is then rotated back into the global coordinate orientation so that their flux contributions to other nodes may properly be evaluated.

Wall Boundary Gradient Boundary Method

Another approach to obtaining the flow properties at surface nodes is to specify gradients of flow properties as boundary conditions. Namely,

$$\begin{aligned} \frac{\partial \rho}{\partial \tilde{y}} &= 0 \\ \frac{\partial \tilde{u}}{\partial \tilde{y}} &= 0 \\ \tilde{v} &= 0 \\ \frac{\partial p}{\partial \tilde{y}} &= \frac{\rho \tilde{u}^2}{R} \end{aligned} \tag{27}$$

where the last equation represents a balance between the pressure in the fluid and the centrifugal force associated with the fluid motion along a curved path defined by the local surface radius of curvature, *R*. Again, the tildes here indicate that the velocity components are in the local (surface normal) coordinate system. Note that for this method, only gradients normal to the wall need be evaluated. Thus, shape functions for surface tangent directions are not required. The surface boundary conditions of Equations (27) can be written in terms of gridless shape functions at surface node *p* by

$$\begin{aligned} \rho_p &= -\frac{1}{\partial \phi_p^- / \partial \tilde{y}} \left(\sum_{\substack{n=1 \\ n \neq p}}^{N_y^-} \frac{\partial \phi_n^-}{\partial \tilde{y}} \rho_p \right) \\ \tilde{u}_p &= -\frac{1}{\partial \phi_p^- / \partial \tilde{y}} \left(\sum_{\substack{n=1 \\ n \neq p}}^{N_y^-} \frac{\partial \phi_n^-}{\partial \tilde{y}} \tilde{u}_p \right) \\ \tilde{v}_p &= 0 \\ p_p &= \frac{1}{\partial \phi_p^- / \partial \tilde{y}} \left(\frac{\rho_p \tilde{u}_p^2}{R_p} - \sum_{\substack{n=1 \\ n \neq p}}^{N_y^-} \frac{\partial \phi_n^-}{\partial \tilde{y}} p_p \right) \end{aligned} \tag{28}$$

The above equations can be evaluated subsequent to evaluation of the field nodes after each time step. This boundary method provides for a higher order description of the surface geometry since the surface radius of curvature is incorporated in addition to the discrete locations of the boundary points.

Surface Singularity (Trailing Edge Tip)

Although the method does not require any special treatment for thin geometry, the presence of surface singularities such as the tip of an airfoil trailing edge are problematic since the surface radius of curvature and normal direction cannot be well defined. In the results presented below, two approaches are considered. One is to assign a unit normal direction parallel to the chord line. and a small (but finite) radius of curvature. In this case, the solution at the trailing edge is seen to have an unphysical drop in pressure attributed to the discontinuous definition of surface properties (i.e unit normal direction). Another approach is to extrapolate flow properties from within the flow field using the shape functions described above. Although this is an undesirable feature of the method, it does not detract significantly from the overall solution, and a more appropriate treatment is currently under investigation.

Multistage Time Stepping

Time integration of the flow field (evolution) is achieved using Runge-Kutta time stepping, as first presented by Jameson, et al[21]. This explicit multi-stage scheme is ideally suited for multigrid acceleration and provides an increased (though limited) allowable time step compared to computational work. Considering separate temporal and spatial discretization (i.e. method of lines), the semi-discrete form of the governing equations can be written

$$\frac{d\mathbf{u}}{dt} = -\mathrm{Res}_j \tag{29}$$

where the residual at node *j*, Res_n, is a discretized representation of the spatial derivative (e.g. Equations (11) and (26)). The Runge-Kutta integrations is a sequence of updates which takes the solution from time level *n* to time level *n*+1. The *k* stages, the scheme reads

$$\begin{aligned} u_j^{(0)} &= u_j^n \\ u_j^{(1)} &= u_j^{(0)} - \alpha_1 \Delta t_j \mathrm{Res}_j^{(0)} \\ u_j^{(2)} &= u_j^{(0)} - \alpha_2 \Delta t_j \mathrm{Res}_j^{(1)} \\ &\vdots \\ u_j^{(k)} &= u_j^{(0)} - \alpha_k \Delta t_j \mathrm{Res}_j^{(k\text{-}1)} \\ u_j^{n+1} &= u_j^{(k)} \end{aligned} \tag{30}$$

where α_k are the stage coefficients, and Δt is the local allowable time step at node *j*. The stage coefficients can be optimized for various spatial discretizations. For instance, Van Leer, et al[22], provides stage coefficients for 3, 4, and 5 stage schemes optimized for first order and second order upwind schemes. For a 4 stage first order upwind scheme, the coefficients read

$$\alpha_k = \{0.0833, 0.2069, 0.4265, 1.0000\} \quad (31)$$

It is noted that in solutions containing shocks, stage coefficients optimized for first order schemes is preferred regardless of the spatial discretization[23].

A disadvantage of explicit time marching schemes is the limitation of the allowable time step based on the CFL condition (ratio of acoustic speed to propagation distance per time step.). This results from the fact that acoustic propagation cannot travel farther than the distance of the basic grid dimension in one time step without causing instabilities in the solution. Implicit time marching schemes do not suffer from this limitation since communication between all grid cells is achieved during each time step. However, explicit schemes are more appropriate for embedded multi-grid applications, as considered here. The optimal CFL for the stage coefficients presented above is reported to be 2.0.[22]

For steady state solutions where time accuracy is not required, the use of local time stepping can also significantly accelerate convergence. Here, the local time step varies throughout the computational domain based on applying the CFL condition to the local flow conditions and grid spacing.

Multigrid

A full approximation storage (FAS) multgrid scheme based on the work of Jameson[24] is incorporated for communication and convergence acceleration of the flow solution on an embedded mesh hierarchy. Here, embedded meshes use coarse meshes beneath them to accelerate convergence of the solution. The multigrid method accelerates the solutions to convergence primarily by means of two effects. First, larger time steps can be applied on coarser grid with reduced numerical effort, and much of the computational effort is then distributed. Secondly, iterative schemes are most efficient at reducing the high frequency components of the solution error, whereas low frequency error is barely affected. Thus, upon transferring the solution from a fine grid to a course grid, the low frequency components of error now become high frequency components and are therefore more effectively damped. A correction to the fine grid is then obtained from the coarse grid solution by transfer of the residuals.

As mentioned, the work presented here incorporates an embedded mesh hierarchy in which the finest mesh is a host to the body geometry, beneath which is a coarser mesh that extends farther toward the domain boundaries, as indicated in Figure 2 which depicts a geometry host grid for a cylindrical geometry, and one coarser grid below it. The portion of the coarse grid that lies beneath the finer grid is used for corrections to the finer grid in the multi-grid cycle.

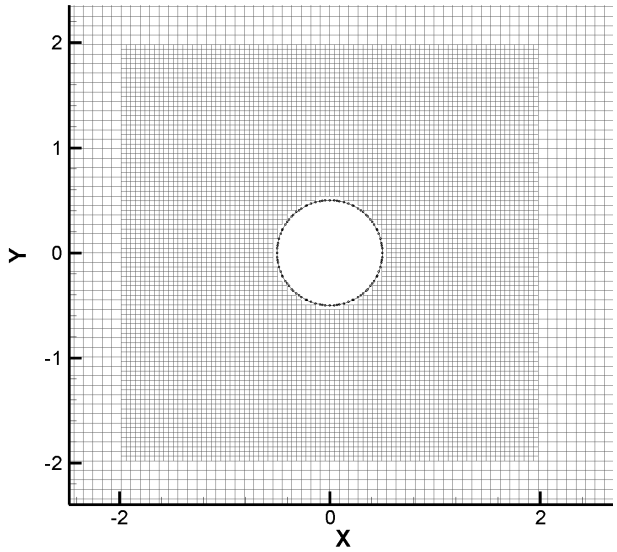

Figure 2 Embedded Mesh Example

Results

Some results of the gridless boundary condition treatment presented above are given. Although at this writing incorporation of the adaptive cloud nodes has not yet been completed, several test cases have been completed in which the merits of the method can be demonstrated. Initial computations considered a transonic flow about the NACA 0012 airfoil for a Mach number of 0.8 and 1.25 degrees angle of attack, in which the single stretched grid of Figure 3 was employed. In these calculations, a comparison was made between the pressure coefficient distribution resulting from using the surface intersection nodes or the surface definition (spline) nodes. Figure 4 presents the pressure coefficient distribution for each. As shown there is little visible difference between the results with the exception of a small discrepancy in the upper surface shock location. These results indicate that surface definition nodes can be used in lieu of those defined by the intersection of the body (and aligned with the Cartesian mesh). The Mach number distribution for the case using the surface definition nodes is presented in Figure 5. It is pointed out that the preliminary results of Figure 4 and Figure 5 do not incorporate far field vorticity correction[25] and, due to the stretching of the mesh, have a far field location less than 10 chord lengths away. Thus the results of Figure 4 and Figure 5 are applicable for mutual comparison only. The boundary treatment for these cases used a 3 term polynomial basis in conjunction with the surface gradient method presented above (see Equation (16)). The effect of discontinuous slope at the trailing edge is evident in pressure coefficient plots for both cases, resulting in the sudden pressure drop at the tip. The

following cases incorporate field extrapolation for properties at the airfoil tip, as mentioned above. As a result, the sudden pressure drop is not present.

Next, a more critical test on the accuracy of the method was investigated with the incorporation of embedded Cartesian meshes in addition to far field vorticity correction applicable for lifting bodies as proposed by Usab and Murman[25]. In this case, the NACA 0012 is again considered, but the solution is computed on an embedded hierarchy of 8 Cartesian meshes for two freestream conditions: Mach 0.8 at 1.25 degrees angle of attack, and Mach 0.5 at 3 degrees angle of attack. These results are shown to compare very favorably to Jameson's FLO52 code even though mesh refinement in the high curvature nose of the airfoil is not incorporated. Figure 6 presents an overview of the entire domain, whereas Figure 7 presents a closer view of the airfoil nose region. For comparison purposes, the FLO52 mesh in the vicinity of the airfoil nose is presented in Figure 8.

Here, both the time marching and boundary gradient methods are investigated using a six term polynomial basis for surface nodes, and 3 term for field nodes flux directions not having a full second order finite difference stencil. Figure 9 and Figure 10 present a comparison between the FLO52 results for surface pressure coefficient and those resulting from the boundary gradient and time marching treatments, respectively. As shown, the boundary gradient method is able to match the FLO52 results more precisely. This is attributed to the fact that the boundary gradient method incorporates the body radius of curvature into the solution. The time marching method relies more heavily on a fine mesh near regions of high curvature in order to retain surface curvature information. An obvious advantage of the time marching method would be in three dimensional applications since then the surface curvature is much more difficult to specify than in the two dimensional case. It is pointed out that, although the mesh of Figure 6 appears relatively fine, in terms of nodal distribution along the high curvature region it is not adequate to capture the surface curvature, as indicated by distance between solution points in Figure 9 and Figure 10. Such is the nature of the Cartesian mesh. Nonetheless, encouraging results are obtained. Finally, Figure 11 presents Mach number contours consistent with the results of Figure 9 for the gridless method.

Pressure coefficient results for the Mach 0.5, 3-degree angle of attack case are presented in Figure 12 and Figure 13. A similar trend between the two gridless boundary treatments can be seen, although for the prediction of lift coefficient both treatments are in much closer agreement with the results of FLO52. Although the trailing edge stagnation point is not closely matched to the FLO52 results using the field extrapolation, excellent agreement is still apparent.

Conclusion and Future Work

A new method for treating boundary conditions for non-body conforming computational grids has been presented. The method has significant advantages over other formulations in that it can esily incorporate additional surface geometry nodes into the solution, and does not require special treatment for thin geometry. Surface discontinuities remain an issue that is currently being addressed.

Results presented here have demonstrated the feasibility of the concept and its applicability to transonic thin body problems. Comparisons with body fitted results have shown to be in good agreement considering the lack of adaption and resolution in regions of high curvature. The primary goal of future work is to implement grid adaptation techniques to increase the solution accuracy in high curvature regions and facilitate evaluation of nodal shape functions. In addition to refinement of the Cartesian mesh, a fundamental aspect in the grid adaption will be the incorporation of "cloud" nodes which naturally fit into the gridless technique. For instance, they can be introduced along the direction of the surface normal, or concentrated in trouble spots like the trailing edge tip.

Finally, a detailed study on the convergence and accuracy of the method is also planned, as compared to traditional CFD approches. The method may then be extended to more complex geometries and/or higher dimensions.

References

1. Lahur, P. R, Nakamura, Y., " A New Method for Thin Body Problem in Cartesian Grid Generation", AIAA Paper 99-0919, 1999.
2. Purvis, J.W., Burkhalter, J. E., "Prediction of Critical Mach Number for Store Configurations", AIAA Journal, Vol. 17, pp. 1170-1177, November 1979.
3. Wedan, B., South, J. C. Jr., "A Method for Solving the Transonic Full-Potential Equation for General Configurations", AIAA Paper 83-1889, 1983.
4. Clarke, D. K., Salas, M. D, Hassan, H. A., "Euler Calculataions for Multielement Airfoils Using Cartesian Grids", *AIAA Journal*, Vol. 24, , pp. 353-358 March 1986.
5. Gaffney, R. L., Hassan, H. A., Salas, M. D., "Euler Calculations for Wings Using Cartesian Grids", AIAA Paper 87-03563, January 1987.
6. Epstein, B., Luntz, A. L., Nachshon, A., "Multigrid Transonic Computations About Arbitrary Aircraft Configurations", *Journal of Aircraft*, Vol. 26, pp.751-759, August 1989.

7. Tidd, D. M., Strash, D. J., et. al., "Multigrid Euler Calculations over Complete Aircraft", *Journal of Aircraft*, Vol. 29, pp. 1080-1085, Nov.-Dec. 1992.
8. Dadone, A., Grossman, B., "An Immersed Body Methodology for Inviscid Flows on Cartesian Grids", AIAA Paper 2002-1059, January 2002.
9. Dadone, A., Grossman, B., "Surface Boundary Conditions for the Numerical Solution of the Euler Equations", *AIAA Journal*, Vol. 32, pp. 285-293, 1995.
10. Leveque, R. J., Berger, M. J., "A Rotated Difference Scheme for Cartesien Grids in Complex Geometries", AIAA Paper CP-91-1602, 1991.
11. Pember, R. B., Bell, J. B., et. al., "An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions", *Journal of Computational Physics*, Vol. 120, pp. 278-304, 1995.
12. Forrer, H., Jeltsch, R., "A Higher Order Boundary Treatment for Cartesian-Grid Methods", *Journal of Computational Physics*, Vol. 140, pp. 259-277, 1998.
13. De Zeeuw, D., Powell, K. G., "An Adaptively Refined Cartesian Mesh Solver for he Euler Equations", *Journal of Computational Physics*, Vol. 104, pp. 56-68, 1993.
14. Melton, J. E., Berger, M. J., Aftosmis, M. J., Wong, M. D., "3D Applications of A Cartesian Grid Euler Method", AIAA Paper 95-0853, 1995.
15. Aftosmis, M. J., Berger, M. J., Adomavicius, G., "A Parallel Multilevel Method for Adaptively Refined Cartesian Grids with Embedded Boundaries", AIAA Paper 2000-0808, *38th Aerospace Sciences Meeting and Exhibit*, January 2000..
16. Batina, J. T., "A Gridless Euler/Navier-Stokes Solution Algorithm for Complex Two-Dimensional Applications", NASA-TM-107631, June 1992.
17. Shih, S.-C., Lin, S.-Y., "A Weighted Least Squares Method for Euler and Navier-Stokes Equations", AIAA Paper 94-0522, 1994.
18. Liu, J. L., Su, S. J., "A Potentially Gridless Solution Method for the Compressible Euler/Navier-Stokes Equations", AIAA Paper 96-0526, 1996.
19. Van Leer, B., "Flux-Vector Splitting for the Euler Equations", *Proc. 8th International Conference on Numerical Methods in Fluid Dynamics*, Springer Verlag, pp. 507-512, 1982
20. Belytschko, T., Lu, Y. Y., Gu, L., "Element-Free Galerkin Methods", *Int. Journal Num. Methods in Engineering*, Vol. 37, pp. 229-256, 1994.
21. Jameson, A., Schmidt, W., Turkel, E., "Numerical Solutions of the Euler Equations by Finit Volume Methods using Runge-Kutta Time-Stepping", AIAA Paper 81-1259, *AIAA 14th Fluid and Plasma Dynamics Conference*, June, 1981.
22. Van Leer, B. Tai, C.-H., Powell, K. G., "Design of Optimally Smoothing Multi-Stage Schemes for the Euler Equations", AIAA Paper 89-1933, 1989.
23. Blazek, J., *Computational Fluid Dynamics: Principles and Applications*, Elsevier Science Ltd publisher, 2001.
24. Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method", *Applied Mathematics and Computation*, Vol. 13, pp. 327-356, 1983.
25. Usab, W. J., Murman, E. M., "Embedded Mesh Solution of the Euler Equations Using a Multiple-Grid-Method", AIAA Paper 83-1946, 1983..

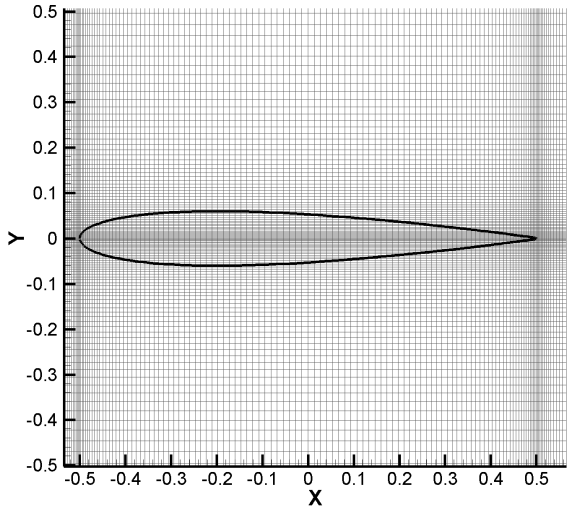

Figure 3. NACA 0012 Surface Geometry and Stretched Cartesian Mesh

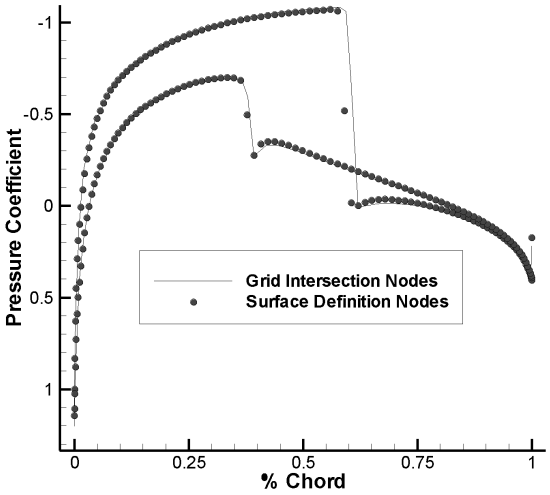

Figure 4. NACA 0012 Cp for Stretched Mesh, Boundary Gradient (M=0.8, α=1.25)

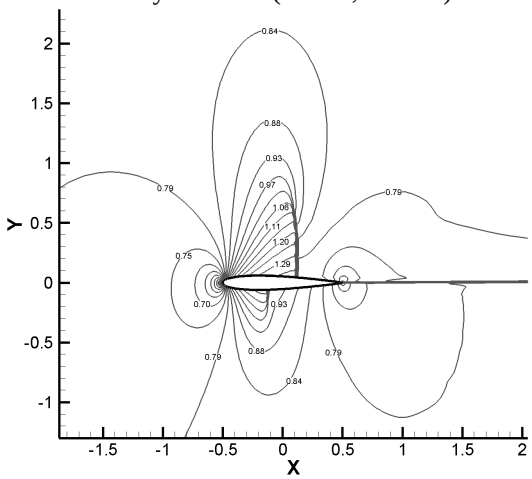

Figure 5. NACA 0012 Mach Number for Stretched Mesh for Gridless Boundary Gradient (M=0.8, α=1.25)

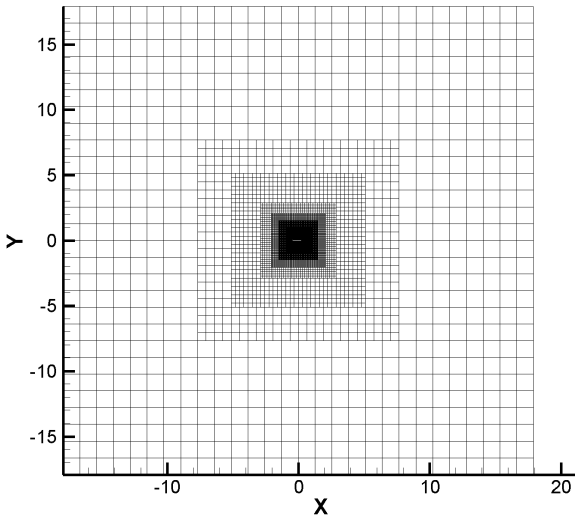

Figure 6. Cartesian Mesh Domain for NACA 0012

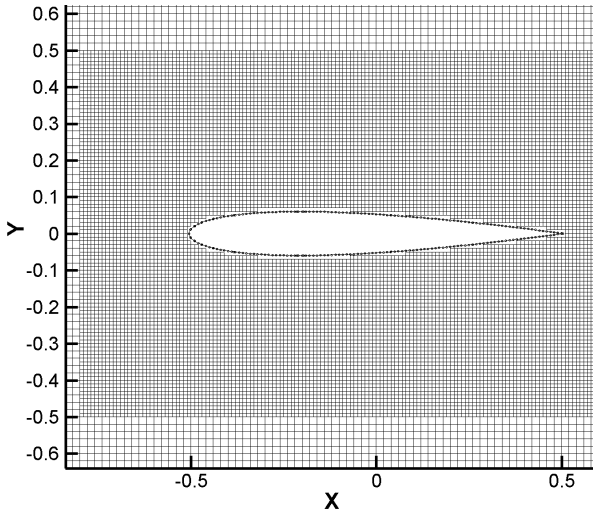

Figure 7. Near Field Cartesian Mesh for NACA 0012

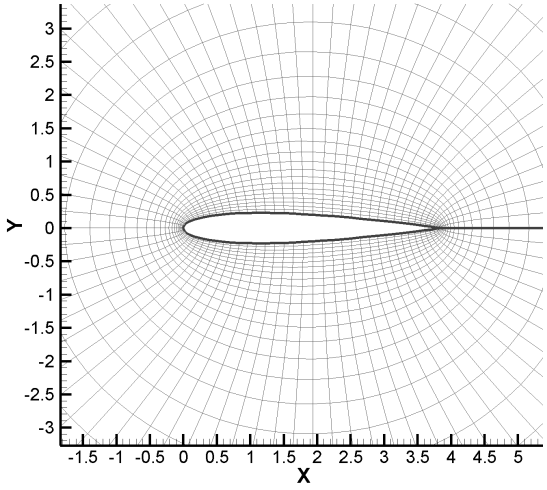

Figure 8 Near Field Body Fitted Mesh (FLO52) for NACA 0012

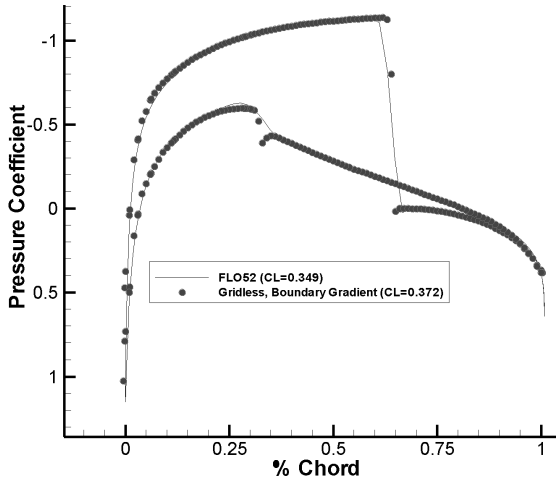

Figure 9 Comparison of Cp for Gridless Boundary Gradient Method with FLO52 (M=0.8, α=1.25)

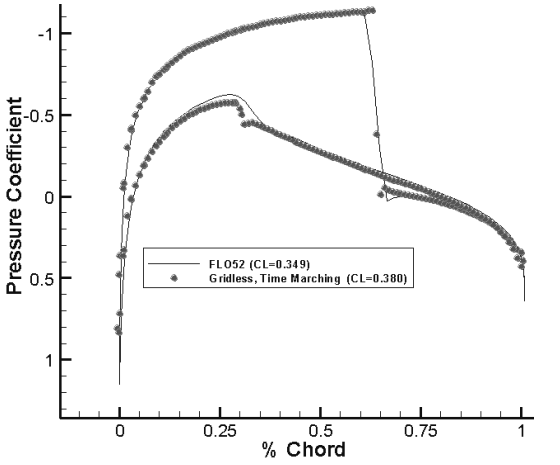

Figure 10. Comparison of Cp for Gridless Time Marching Method with FLO52 (M =0.8, α=1.25)

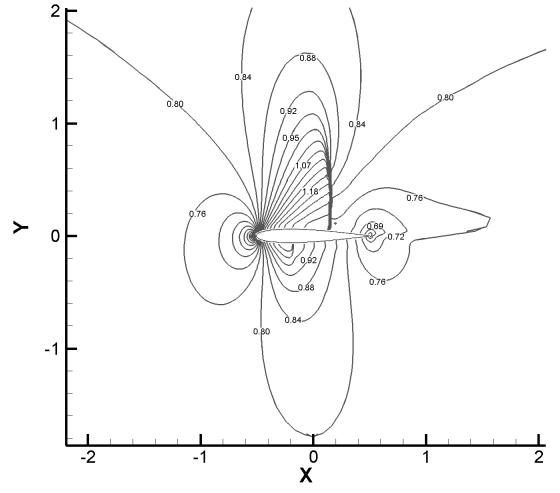

Figure 11. NACA 0012 Mach Number Contours for Cartesian Mesh using Gridless Boundary Gradient (M=0.8, α=1.25)

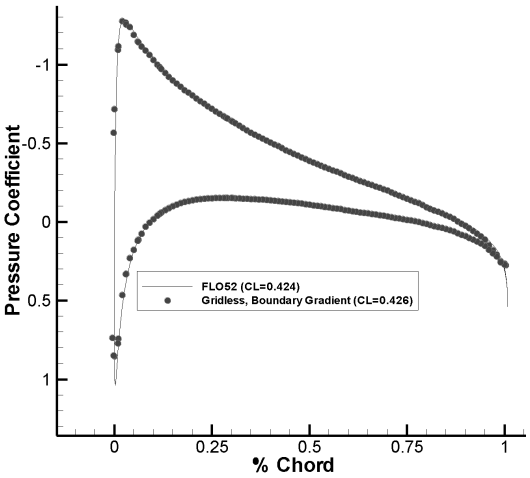

Figure 12 Comparison of Cp for Gridless Boundary Gradient Method with FLO52 (M=0.5, α=3.0)

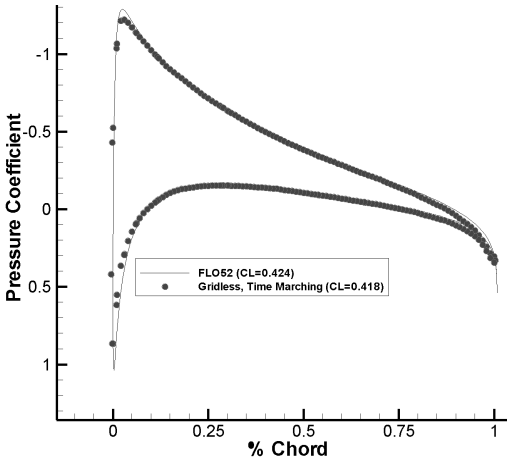

Figure 13. Comparison of Cp for Gridless Time Marching Method with FLO52 (M=0.5, α=3.0)

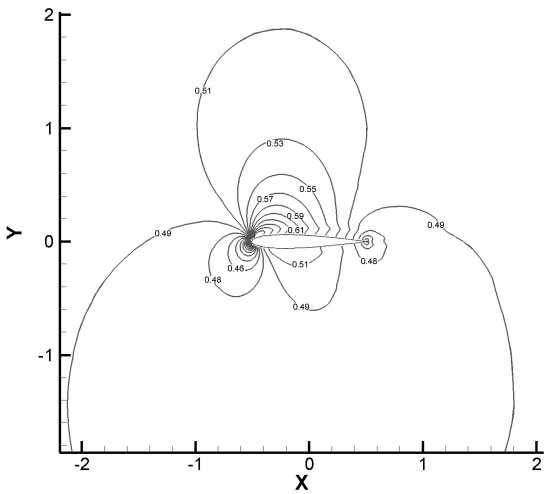

Figure 14. NACA 0012 Mach Number Contours for Cartesian Mesh using Gridless Boundary Gradient (M=0.5, α=3.0)