Progress towards a Cartesian Cut-Cell Method for Viscous Compressible Flow

Marsha Berger*
NYU, 251 Mercer St., NY, NY 10012

Michael J. Aftosmis†
NASA Ames, Moffett Field, CA 94035

The proposed paper reports advances in developing a method for high Reynolds number compressible viscous flow simulations using a Cartesian cut-cell method with embedded boundaries. This preliminary work focuses on accuracy of the discretization near solid wall boundaries. A model problem is used to investigate the accuracy of various difference stencils for second derivatives and to guide development of the discretization of the viscous terms in the Navier-Stokes equations. Near walls, quadratic reconstruction in the wall-normal direction is used to mitigate mesh irregularity and yields smooth skin friction distributions along the body. Multigrid performance is demonstrated using second-order coarse grid operators combined with second-order restriction and prolongation operators. Preliminary verification and validation for the method is demonstrated using flat-plate and airfoil examples at compressible Mach numbers. Simulations of flow on laminar and turbulent flat plates show skin friction and velocity profiles compared with those from boundary-layer theory. Airfoil simulations are performed at laminar and turbulent Reynolds numbers with results compared to both other simulations and experimental data (*included in final paper*).

I. Introduction

Cartesian embedded-boundary grids have proven to be extremely useful for inviscid flow simulations around complex geometries. There are many flow situations for which their rapid turnaround time, accuracy, and level of automation have had great success. The use of cut cells at the intersection of the grid and the geometry has been well-studied, and the numerical issues of discretizations, stiffness and convergence for these irregular cells are well understood.

Much less research has been done examining high Reynolds number viscous flows on cut-cell meshes. Since the Navier-Stokes equations involve one higher derivative, the numerical issues are more delicate. The literature shows difficulties extracting smoothly varying quantities such as skin friction due to grid irregularity of the cut cells, along with loss of accuracy and numerical stiffness. In addition, the resolution requirements needed to compute aerodynamic flows at Reynolds numbers of interest are daunting. The inability of Cartesian meshes to (easily) refine anisotropically in an arbitrary direction is an obvious challenge for these methods.

There are several approaches found in the literature for dealing with the mesh irregularity and resolution requirements of Cartesian meshes. The most obvious is to avoid cut cells altogether, and use a layer (or more) of conformal cells before switching to a background Cartesian mesh.[1–3] One way this can be done is by generating a body-fitted grid in the near-body region and then changing to a background Cartesian grid. Alternatively one can start with a cut-cell mesh, remove the layers of cells adjacent to the body, and then drop normals to the body to create the body-aligned mesh layers.[4,5] Both of these approaches give a body-fitted layer so that the Cartesian grid cell counts of isotropic

[*] berger@cims.nyu.edu. Member AIAA.
[†] michael.aftosmis@nasa.gov. Senior Member AIAA.

refinement can be avoided, but at the price of meshing simplicity. An alternative approach is to use the Cartesian mesh down to the wall. This can be done using either a finite-volume[6,7] or finite-difference[8,9] scheme, or other approaches[10,11] but all will all have to contend with the non-aligned boundary. Some methods use an immersed boundary[12,13] instead of explicitly cutting the cells that intersect geometry. For any of these methods to ultimately be practical, some additional technique, for example, wall functions or some form of subgrid wall model, must be used to to alleviate the cell count problem of isotropic refinement for high Reynold flows. Recent papers[14–17] present promising approaches of this type, and are the most closely related to our current work.

Each of the approaches above have benefits and drawbacks. We have chosen to use Cartesian cut cells so that all mesh faces remain Cartesian, placing emphasis on the ability to automatically generate meshes around complex configurations. Since all the faces (even cut faces) are Cartesian, the resulting volume mesh is decoupled from the surface triangulation. This simplicity is one of the great strengths of embedded boundary methods and is key for automation. Since all the faces remain Cartesian, this approach also allows highly optimized Cartesian flow solvers. In this work we build upon our earlier efforts[18,19] and integrate the governing equations using a second-order cell-centered finite-volume scheme. A multigrid-accelerated Runge-Kutta method with local time-stepping advances the solution to steady state.

The purpose of this paper is to present progress in extending our inviscid embedded-boundary algorithms towards the development of a cut-cell method to compute high Reynolds number compressible viscous flow. The first step is to develop accurate discretizations needed for the cut cells and the mesh interfaces. This is discussed in Section II. Of particular importance is the use of a quadratic polynomial to compute values of the solution and its derivatives needed for the flux computations in the cut cells. This section also presents the RANS equations and turbulence model used in our computational examples. Section III discusses our multigrid solver, and the modifications that were needed to be able to handle viscous flow. Section IV presents two-dimensional computational examples of both laminar and turbulent flow, including flat-plate boundary layers on non-coordinate aligned meshes, and compressible airfoil simulations. Conclusions are in Section V.

II. Numerical Discretizations

The development of an accurate discretization for the Navier-Stokes equations in a finite-volume context is much more delicate than for the inviscid equations. Two new stencils need to be developed to compute second-derivative terms at embedded boundaries – one at the cut faces between cut cells, and the other to compute the cut-cell boundary conditions at a solid wall. To start, we implemented several possible discretizations for an elliptic equation to test their accuracy on a problem with a known solution. Since the method has to eventually fit into our finite-volume flow solver, these tests do not rely on special properties of elliptic equations.

A. Preliminaries

To help guide our selection of a discretization for the Navier-Stokes equations, work we evaluated three second-derivative embedded-boundary discretizations to solve a Poisson equation in two dimensions.

Consider the model problem

$$\nabla^2 u = f \qquad (1)$$

with Dirichlet boundary conditions and exact solu-

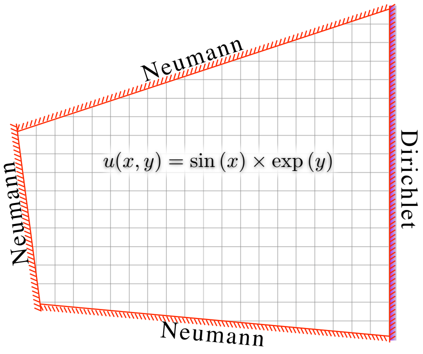


Figure 1. Domain for elliptic experiments in Table 1.

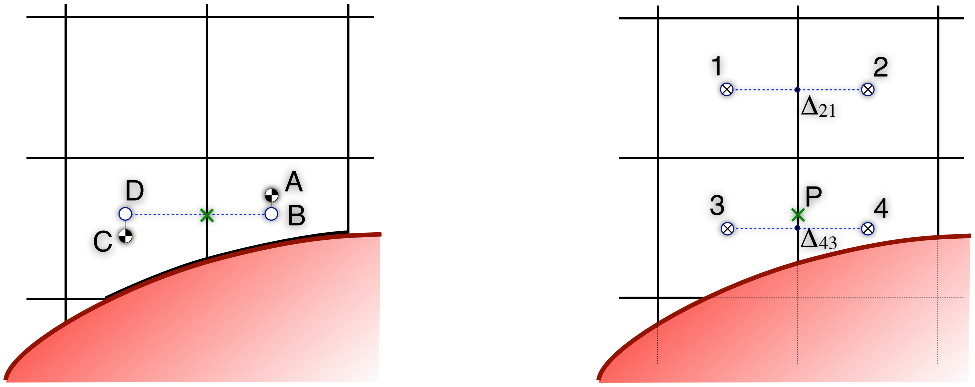
tion $u(x,y) = \sin(x)\exp(y)$. The domain is a quadrilateral with non-coordinate aligned sides, as shown in Figure 1. Three of the four sides use Neumann boundary conditions, with Dirichlet on the fourth. In finite-volume form the discretization of cell C is $\iint_C \nabla \cdot \nabla\, u\, dA = \int_{\partial C} \nabla u \cdot n\, dl = \iint_C f\, dA$. Thus the fluxes needed at the cell interfaces are the first derivatives, u_x at an x-face, and u_y at a y-face. This is in contrast to the Euler equations, where only the value of the solution u itself is needed at the cell edge. At interior cells the standard second-order accurate discretization was used.

Recentering: The first method uses a "recentering" idea,[20,21] illustrated in Figure 2a. The cell averages in the finite-volume scheme are considered to be the approximate solution at the cell centroids. At the cut cells, a least squares gradient is reconstructed using the solution from neighboring cells that share an edge. The gradient is used to reconstruct the solution from the cut-cell centroid to a line through the perpendicular to the face centroid. This is done on both sides of a cut face, so that the derivative can be approximated by a simple difference in the face normal direction. In the notation of Figure 2a, where the centroids are at points A and C, and the recentered values are located at B and D, we get

$$\begin{aligned} u_x &= \frac{u_B - u_D}{x_B - x_D} \\ &= \frac{(u_A + \nabla u_A \cdot d_{BA}) - (u_C + \nabla u_C \cdot d_{DC})}{x_B - x_D} \end{aligned} \tag{2}$$

where d_{BA} is the vector from A to B, and similarly for d_{DC} . Note that this divided difference is nominally only first order accurate, since it is not centered about the face centroid. However changing the recentering step to reconstruct to locations equally spaced about the centroid is no more accurate without using a more accurate reconstruction, so we do not include those results here.

Johansen-Colella: The second discretization uses the Johansen-Colella framework.[22] In this approach the solution u is thought of as being located in the center of the original uncut cell, not at the cut-cell centroid. First derivatives are then easily computed, and are located at the midpoint of an uncut face. The flux however is needed at the cut-face centroid, illustrated in Figure 2b. This is easily obtained by linear interpolation.



(a) **Recentering:** The solution is reconstructed from cell centroids (A and C) to new points in each cell (B and D) on the line perpendicular to the midpoint of the cut-cell edge. A simple difference (B-D)/Δx approximates the derivative.

(b) **Johansen-Colella:** The solution is considered to be located in the center of the uncut cell. The flux at the centroid of the cut face, P, is linearly interpolated from the uncut fluxes ($\triangle_{21}$ and $\triangle_{43}$) .

Figure 2. Two methods for computing the derivative at a cut-cell edge.

Polynomial Fit: The third discretization is also taken from the literature.[23] At each face, a least-squares polynomial that is linear in the face normal direction (e.g. the x direction for an x face) and quadratic in the transverse direction (e.g. y direction for an x face) can be constructed using 6 neighboring values. We use the cell centered solution to the left and right of the face, the boundary values in those cells, and the values one cell further away. The polynomial is differentiated to obtain the flux at the cut-face centroid.

In all three methods the wall flux was computed using the same discretization (modulo the location of the variables). In finite-volume form the flux at the wall $(\partial u/\partial n)$ is needed. For the sides with Neumann boundary conditions this is straightforward. On the Dirichlet side, a one-sided discretization using the cell average and the Dirichlet value at the boundary normal to the centroid was used.

Figure 3 and Table 1 give the results of computing the solution to (1). All methods show second-order convergence in the L1 norm. The convergence is not entirely smooth, since cut cells do not have a smooth asymptotic expansion for the error. We observe that the Johansen-Colella scheme has slightly better L_1 performance, and that the polynomial fit is somewhat less accurate (especially in the more finicky maximum norm). This conclusion is representative of a variety of test cases we ran. Since the accuracy of the the recentering approach and Johansen-Colella is similar, and the former fits better into our existing finite-volume cut-cell framework, the recentering approach was chosen for implementation of the Navier-Stokes equations.

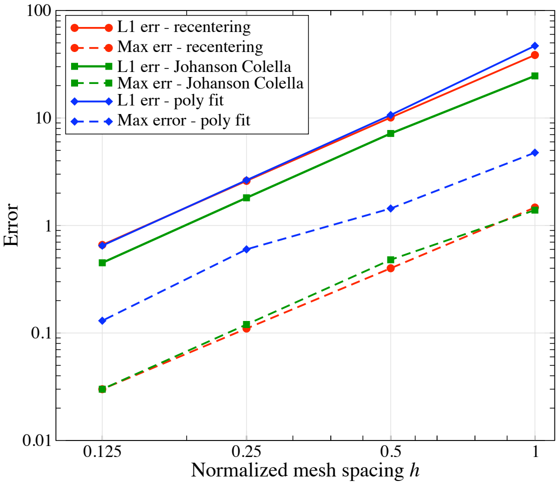


Figure 3. Convergence of 3 methods for computing cut-cell fluxes for the model Poisson equation $\nabla^2 u = f$, corresponding to the data in Table 1.

	Recentering		Johanson-Colella		Poly. fit	
Domain Size	1 norm	max. norm	1 norm	max. norm	1 norm	max. norm
9 × 9	38.53	1.47	24.64	1.39	46.93	4.76
18 × 18	10.10	.40	7.18	.48	10.66	1.44
36 × 36	2.60	.11	1.81	.12	2.64	.60
72 × 72	.66	.03	.45	.03	.65	.13

Table 1. Results of solving the Poisson equation $\nabla^2 u = f$ on an irregular domain using 3 different discretizations for the cut cells. Data corresponds to Figure 3.

Of course these results need to be interpreted carefully. In practical settings, the remaining terms of the Navier-Stokes equation that are not included in this model problem can dominate the error, and lead to nonconvergence if not handled carefully (the subject of the next subsection), even though the terms discussed here are second-order accurate. In addition, we are not looking at positivity, which has been the focus of several other studies.[2,24] Although positivity insures a maximum principle, blindly insisting on it may rule out potentially useful discretizations.[a]

[a] For example the 4th-order Laplacian derived by combining the Cartesian coordinate-aligned stencil and the rotated stencil has negative coefficients on the corner terms and yet it is symmetric positive definite.

B. Discretization of the Navier-Stokes Equations

In integral form in two space dimensions the compressible Navier-Stokes equations can be written

$$\frac{d}{dt}\iint_{\Omega} U\,dA + \oint_{\partial\Omega}(F\hat{i}+G\hat{j})\cdot\hat{n}\,dS = \oint_{\partial\Omega}(f_v\hat{i}+g_v\hat{j})\cdot\hat{n}\,dS \tag{3}$$

where $U = (\rho, \rho u, \rho v, \rho E)^T$, and

$$f = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho u v \\ u(\rho E + p) \end{pmatrix} \quad g = \begin{pmatrix} \rho v \\ \rho u v \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix} \quad f_v = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{xy} \\ u\tau_{xx} + v\tau_{xy} - q_x \end{pmatrix} \quad g_v = \begin{pmatrix} 0 \\ \tau_{yx} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{pmatrix}.$$

The shear stresses are

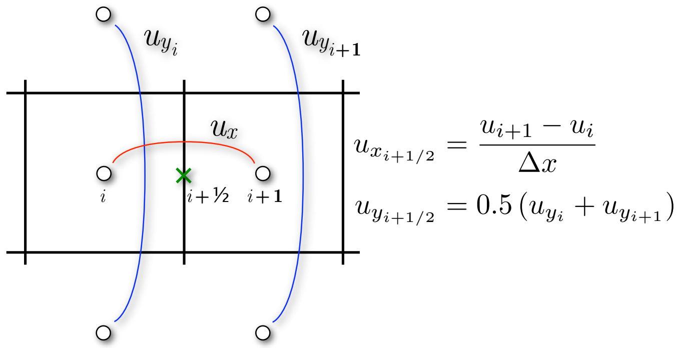
$$\begin{aligned} \tau_{xx} &= 2\mu u_x - 2/3\mu\nabla\cdot\mathbf{v} \\ \tau_{xy} &= \mu(u_y + v_x) \;=\; \tau_{yx} \\ \tau_{yy} &= 2\mu v_y - 2/3\mu\nabla\cdot\mathbf{v} \end{aligned} \tag{4}$$

where the vector ∇q is proportional to the gradient of the temperature $\nabla q = -k\nabla T$ for temperature T, the Prandtl number $Pr = \mu c_p/k$, and μ is computed using Sutherland's law.[25] We take $Pr = .72$.

We first present the basic finite-volume scheme used on the regular Cartesian cells that make up most of the volume mesh. Only the viscous terms will be discussed, since the inviscid terms have been previously described.[19] Following that we present the modifications needed for the viscous discretization in the irregular cells. Of particular importance here will be the method used to compute the gradient in these cut cells.

1. *Cartesian Cell Discretization*

The discretization of the viscous terms on uncut Cartesian cells is straightforward. At the midpoint of each Cartesian face, the gradient of both velocity components is needed. Each cell already computes a cell-centered gradient using a second-order accurate central difference formula. Consider for simplicity an x face. The pitfall to avoid is the unstable 5-point decoupled stencil that would result from simply averaging the left and right cell-centered gradients to obtain u_x at the wall. Instead we ensure that $u_x = \frac{u_{i+1,j}-u_{i,j}}{\Delta x}$ at that face. The transverse derivative however, in this case u_y at the x face, can be taken as the average $u_y = .5(u_{y_{i,j}} + u_{y_{i+1,j}})$. This discretization is commonly found in the literature for regular Cartesian meshes.[26,27]



u_{y_i}
$u_{y_{i+1}}$
u_x
i
$i+½$
$i+1$
$u_{x_{i+1/2}} = \frac{u_{i+1}-u_i}{\Delta x}$
$u_{y_{i+1/2}} = 0.5\,(u_{y_i} + u_{y_{i+1}})$

Figure 4. Illustration of stable viscous flux computation at face between cells i and $i+1$.

At mesh interfaces, a least-squares gradient computation that uses only face neighbors is linearly exact and first order accurate, though it is not centered. We use this gradient to recenter the fluxes at mesh interfaces in the same way it was previously described in (3) at cut cells. The transverse component of the gradient is still taken to be the average of the tranverse gradients on either side of the face, this time weighted by distance from the face.

The inviscid terms of the flow solver (which have previously been described[19]) are unchanged except for our implementation of the solid wall boundary condition. Instead of using a boundary Riemann problem to compute the pressure at a solid wall, we simply reconstruct the pressure to the centroid of the wall segment.

2. Cut Cell Discretization

For the cut faces, the same recentering procedure is used to reconstruct the velocity from the centroid to the perpendicular line through the face centroid. If the face is between two cut cells, both cells follow this procedure and the derivative can be computed. If one of the cells is uncut, the face itself must not be cut, and only the one cut cell needs to recenter the solution, since the full Cartesian cell and the face centroid must already line up. However due to the mesh irregularity in the cut cells, special procedures must be devised for accurate reconstructions at wall boundaries.

Mesh Irregularity Near Cut-Cells

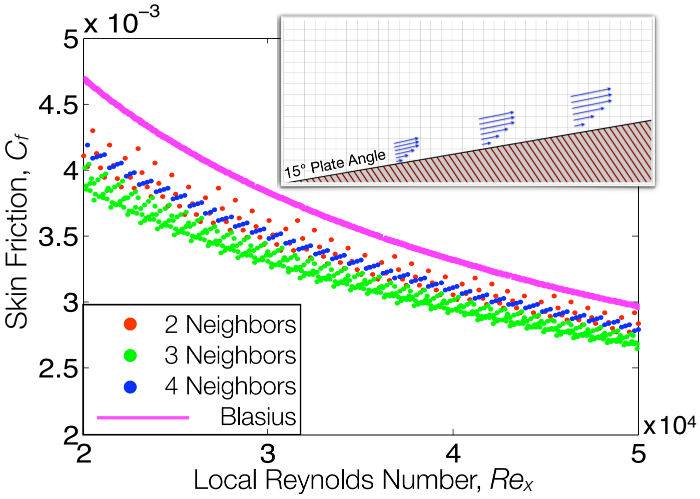


Figure 5. Noise in the skin friction profile along surface of the plate when using piecewise linear gradients in the cut cells

Mesh irregularity adjacent to geometric boundaries has been a major challenge for an accurate discretization of the viscous terms in the Navier-Stokes equations using Cartesian mesh methods. Since the viscous terms include derivative quantities, irregularity affects both the computed viscous fluxes and the output skin friction. Accurate skin friction, in particular, has been a challenge and has stymied many viscous Cartesian efforts. Figure 5 displays the problem and is representative of many of the results found in the literature.[24,28,29] This figure shows the skin friction C_f along a $Re_L = 5000$ flat plate rotated 15° to the Cartesian mesh at a free stream Mach number of 0.5. The magenta line indicates the Blasius solution, and the symbols represent the data extracted from cut cells along the plate's surface. Skin friction data using the viscous discretization method with linear reconstruction outlined in the preceding section have been colored by number of neighbors. Like similar examples in the literature, the skin friction data appear quite noisy around a mean which roughly follows the Blasius result for C_f. In this particular example, however, the sorting of data by number of neighbors reveals a stratification of the noise with cell type. On a 15° flat plate, 2-neighbor cells are always the smallest cut cells while 4-neighbor cells are always the largest. As a result, the patterns of the symbols associated with 2-, 3- and 4-neighbor cells graphically illustrate the link between mesh and stencil irregularity and skin friction noise. While figure 5 examines C_f, it is only one example of the link between accurate first derivatives and stencil irregularity. Similar inaccuracies are present in the viscous fluxes and other derivative quantities. In fact, with linear reconstruction for the gradient the numerical experiments show this noise is just barely convergent, and working in locally wall-aligned coordinates for the least squares gradient does not help.

Before proposing a remedy, it is worth examining more closely the functional approximation and reconstruction within the cut cells. Consider a quadratically varying scalar field on a non-aligned cut cell Cartesian mesh as sketched in frames (a) and (b) of Figure 6. In frame (c) the exact data has been reconstructed through each cell centroid using the exact cell-wise centroidal gradients. This reconstruction makes clear that even with *exact* data and *exact* cell-wise gradients, any measurement of the data along the wall will contain substantial noise simply because the function and its gradient are being sampled at different distances from the wall.

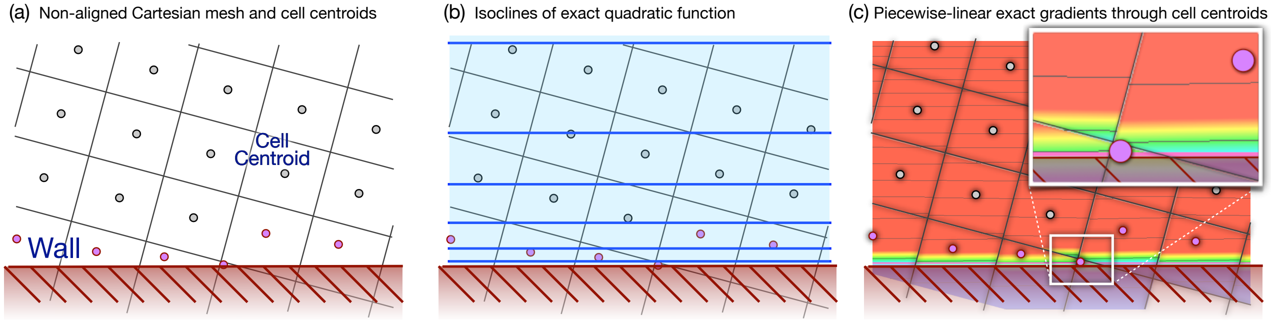

(a) Non-aligned Cartesian mesh and cell centroids
Cell Centroid
Wall
(b) Isoclines of exact quadratic function
(c) Piecewise-linear exact gradients through cell centroids

Figure 6. While isoclines of an exact quadratic are parallel to the wall, the piecewise linear reconstruction is discontinuous, even when using exact gradients at the cell centroids.

Figure 7 clarifies further. The figure shows a non-aligned body with cut cells a, b, c and d. To the right, we sketch a nonlinear profile $U(y)$ marked with the cell-centroid data for cells a through d and slopes $U'(a)$ through $U'(d)$. Obviously, none of these slopes are $(dU/dy)_{wall}$. In light of this, the apparent "noise" in the profile in Figure 5 is not surprising. Cell-centroid gradients are being taken along the wall, even though the cell-centroids in this non-uniform region are at different distances from the wall.

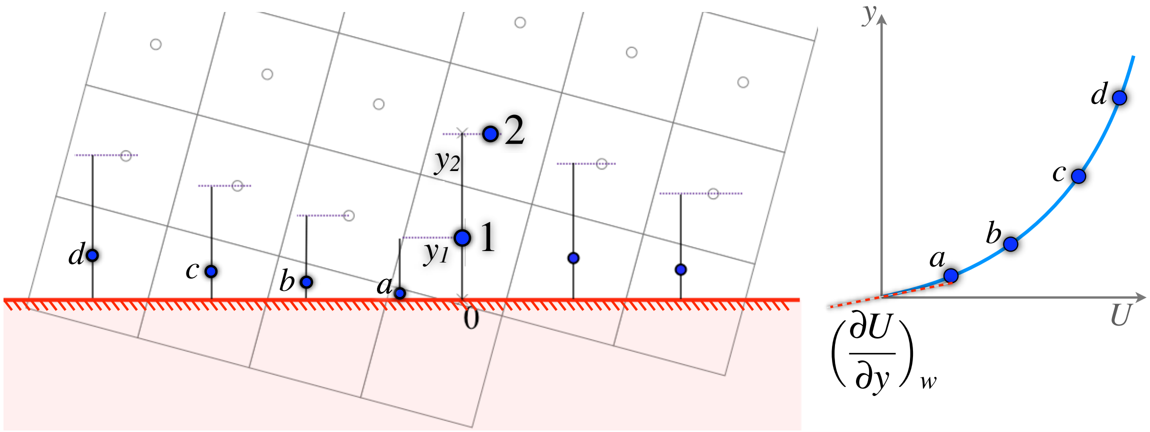


Figure 7. Illustration of a quadratic interpolant through the cut cells on a non-aligned mesh.

This observation suggests an obvious path forward: given cell centroid data at the first and second cells off the wall, reconstruct a quadratic function through these data, and evaluate the wall slope using this reconstruction. Indeed, this reconstruction can provide all relevant slopes needed by the discretization stencil outlined in the preceding section – the wall slopes as well as gradients at cell centroids.

Near-Wall Quadratic Reconstruction

There are numerous approaches for quadratically reconstructing the wall-normal variation in the data. In this case, the fact that the velocities are zero at the wall makes reconstruction using the

Newton form of the polynomials particularly attractive. Consider cell 1 in Figure 7. We construct a line through cell 1's centroid and normal to the patch of the wall contained in cell 1. Point 0 is the wall point on this line a distance y_1 from cell 1's volume centroid. At point 0 the u and v velocities are both zero. Cell 2 is the neighbor across the face intersected by this normal and its centroid is a distance y_2 in the normal direction from the point 0 at the wall. Strictly speaking, data in cell 2 should be recentered (tangentially) but given the relatively slow tangential variation of the data, the current implementation uses data at 2 directly.

The Newton from of the quadratic for the U through points 0, 1 and 2 can be written

$$U(y) = [U_0] + (y - y_0)[U_0, U_1] + (y - y_0)(y - y_1)[U_0, U_1, U_2] \tag{5}$$

where the bracket notation indicates the divided differences of the indicated values. Expanding and taking advantage of the fact that at the wall both the distance y_0 and the function U_0 are zero gives

$$U(y) = yA + y(y - y_1)B \tag{6}$$

with

$$A = \frac{u_1}{y_1}, \quad \text{and} \quad B = \frac{\frac{u_2 - u_1}{y_2 - y_1} - A}{y_2 - y_0} \tag{7}$$

$U(y)$ can be differentiated in the wall normal direction giving

$$U'(y) = A + (2y - y_1)B \tag{8}$$

giving a slope at the wall of

$$U'(y) = A - y_1 B \tag{9}$$

In the current implementation, we recenter the velocity in the Cartesian directions independently for u and v. However since the reconstruction is quadratic in the wall-normal direction only, there may be advantages to operating in a wall-aligned system. *This topic will be investigated for the final paper.* The slope at the wall provides the wall shear stress for both viscous fluxes and output skin friction. The gradient of the quadratic evaluated at the cell centroid replaces the least-squares gradients previously computed and is used in both inviscid and viscous flux balances. The quadratic reconstruction is essentially compensating for the stencil irregularity near the wall, and it is used both in the cut cells and also in the first layer of interior cells (cell 2 in Figure 7). Although these cells are full hexahedra, they also have boundary-dependent stencil irregularity since they are adjacent to the cut cells. Using quadratic reconstruction mitigates the mesh irregularity while maintaining strict conservation. [b]

One might worry that use of a quadratic to discretize wall shear could impact the allowable time step. The Appendix contains a stability analysis for a model 1D heat equation to determine if the quadratic reduces the CFL limit. Results show that using forward Euler in time and central differencing in space results in a timestep limit of 0.43. The standard scheme using linear reconstructions is stable with a CFL limit of 0.5, so the impact on the timestep is minimal.

3. *RANS equations*

Compressible high-Reyolds number turbulent flows are modeled using the Favre-averaged Navier-Stokes equations (still referred to as the RANS equations). They are of the same form as (3) except for the stresses, which are empirically modeled in turbulent flow. Here we use the Boussinesq assumption relating the Reynolds stress tensor to known flow properties. The net effect of this is the addition of

[b] One area for potential improvement is that the quadratic is fit to the cell-averaged value as if it were the pointwise solution, as is common for second-order finite-volume schemes. Strictly speaking a higher order scheme should account for this, but this error is small.

an eddy-viscosity parameter μ_t and a turbulent Prandtl number Pr_t, giving shear stresses

$$\begin{aligned}
\tau_{xx} &= \frac{2}{3}\,(\mu+\mu_t)(2u_x - v_y) \\
\tau_{xy} &= \quad (\mu+\mu_t)(u_y + v_x) \;=\; \tau_{yx} \\
\tau_{yy} &= \frac{2}{3}\,(\mu+\mu_t)(2v_y - u_x)
\end{aligned} \tag{10}$$

and turbulent heat flux $\nabla q = (\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t})\nabla T$.

In these preliminary investigations we model μ_t using the simple 2-layer algebraic turbulence model of Cebici-Smith.[30] This model assumes local equilibrium of turbulence production and destruction/dissipation and is sufficient for the very simple wall-bounded flows used in this early stage of developmentt. This model will be replaced with a more general formulation as this research proceeds. For the turbulent Prandtl number we use $Pr_t = 0.9$.

III. Multigrid Acceleration

Viscous flows on highly anisotropic grids can often cause convergence problems for multigrid algorithms. However since our Cartesian cells are essentially isotropic we expect better performance when multigrid is appropriately beefed up for viscous flow. Since our inviscid multigrid algorithm has been fully described,[19,31] in this section we present only the modifications that were needed for the Navier-Stokes equations.

The inviscid solver used a first-order spatial differencing on coarser grids, so no geometric information about the cut cells was needed beyond cut face areas and the wall normal vector in each cut cell. For a second-order scheme, the cut cell centroids, cut face centroids, and surface (wall) centroid are needed. These permit both the gradient computation and a second-order stencil on the coarser grids. Most of this information is easily computed from the fine grid when the coarser grids are formed; for example coarse grid cell centroids are weighted averages of all the fine grid cells that restrict to that cell. Difficulties arise in recognizing split cells on-the-fly during mesh coarsening. These are Cartesian cut cells split into multiple control volumes by the geometry. As Figure 8 shows, split cells on the fine grid can yield a cut (but unsplit) coarse cell; alternatively, cut (but unsplit) fine cells can be agglomerated into two or more coarse control volumes. These cases can be resolved using an integer matching algorithm which looks for common triangles on the sorted lists of intersected triangles each cut cell maintains, and by using the face lists that connect cut cells with their neighbors.

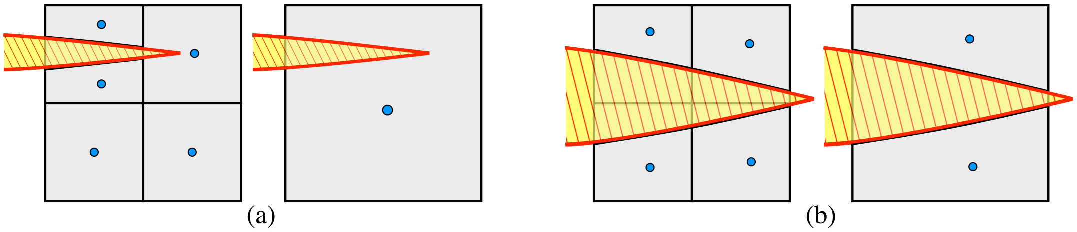


Figure 8. (a) A fine split cell, a cut cell, and two full cells that make up one coarse cut (but not split) cell. (b) Four fine cut cells that make one coarse split cell containing two control volumes.

In addition to using a second-order coarse grid operator with coarse grid gradients, the other major change is the prolongation operator. Theory suggests that higher-order derivatives need a better prolongation than the piecewise constant approach typically used for inviscid flow with cell centered schemes. Our restriction operator, a volume-weighted average of the solution and residual on the fine grid, is already second-order. With gradients now defined on the coarse levels we can use a linear prolongation to bring the change in the solution back to the finer levels. Two copies of the solution were already needed - the initial restriction to the coarse grid and the solution after smoothing (and recursive calls to even coarser levels). Thus, in addition to the computational expense, there is some minor additional memory overhead from storing gradients on the coarser levels.

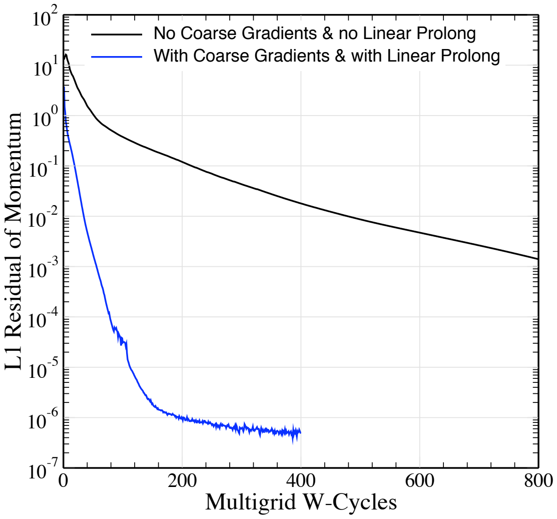


Figure 9. Full multigrid scheme comparing the use of linear prolongation and coarse grid gradients with piecewise constant prolongation and no coarse gradients. Six levels of multigrid were used.

Figure 9 shows the convergence history using second-order coarse meshes and the improved prolongation, for the NACA 0012 computation presented in Section III. Six levels of multigrid were used, with a W-cycle consisting of one pre- and one post-sweep of a 5 stage Runge-Kutta scheme on each level. When multigrid was used with linear prolongation and coarse grid gradients, a CFL of 1.4 was used. If gradients on the coarse grids were used with only piecewise constant prolongation, the stability limit decreased with the number of levels. For the six level computation in Figure 9 the stability limit was reduced to 0.1. Note that the improved convergence and stability required the combination of both linear prolongation and the improved coarse grid spatial scheme − using either alone was insufficient to noticeably improve convergence. Figure 9 shows that when used in combination we can achieve convergence behavior nearly on a par with that of the base inviscid scheme previously reported.[19]

IV. Computational Examples

In this early investigation, computational results are principally confined to verification and validation exercises. Our goal is to demonstrate that accurate solutions can be achieved on Cartesian cut cell meshes despite the grid irregularity. We do not focus on performance, since we have not yet begun to deploy any sub-grid or wall-layer modeling to attack the cell count or meshing efficiency issues.

A. Flat Plate Boundary Layer

The first test is a two-dimensional flat plate boundary layer for $M_\infty = 0.5$ and $Re_L = 5000$. As in Section II.B the plate is oriented 15° to the mesh. The plate starts at $x = 0$ in a domain with x spanning -9 to 20 and y from -3 to 30. The HLLC Riemann solver[32] was used, with wave speeds from ref. [[33]]. Figure 10 shows a sketch of the problem setup (top left), and isoclines of the computed tangential velocity, which are much smoother than the discontinuous contours shown in Figure 6c. The figure also shows the normal and tangential velocity profiles, with and without limiting. The velocity profiles are taken at 3 stations corresponding to $Re_x = 5000, 10000$, and 50000 along the plate, plotted in self-similar coordinates. As expected, the tangential velocity profiles collapse on each other in these variables. The normal velocity profiles show some effects of the plate leading edge and the mesh resolution. The calculation on the left did not use limiters, and shows some viscous overshoot at the first two stations. Velocity profiles at the right of Figure 10 used the van Leer limiters and have no viscous overshoot. Isotropic Cartesian cells were used with a cell size at the wall of $h = 5.9 \times 10^{-3}$, giving approximately 13, 17, and 40 cells in the boundary layer at the three stations respectively. These resolution requirements are similar to standard second-order finite-volume codes.[34] Note that since the profiles are taken in the direction normal to the wall and not a coordinate direction, each profile intersects both x and y grid lines along the way, so the number of symbols does not correspond to the number of cells.

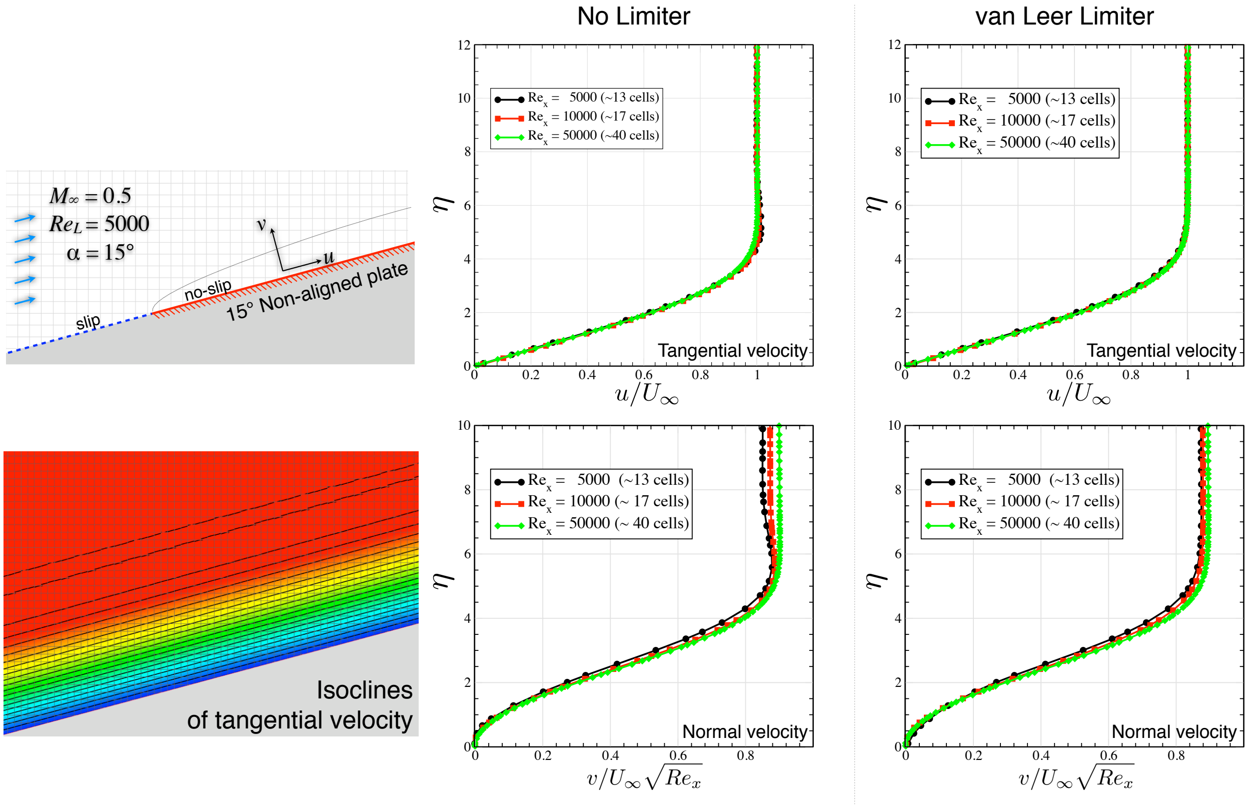


Figure 10. Three profiles at $Re_x = 5K, 10K$ and $50K$ from the flat plate boundary layer compared to the Blasius solution for $M_\infty = 0.5$, $Re_L = 5000$. The plate is rotated 15° to the mesh, shown on the left. In the middle figures, the profiles were not limited; on the right the van Leer limiter was used.

Figure 11 shows the skin friction distribution for this case, discussed in detail in Section II.B. C_f is well predicted by $Re_x = 100$, and compares nicely with coordinate-aligned results.[34] To the right of Figure 11 we show a close-up on a scale similar to that of Figure 5 showing the remarkably smooth C_f distribution provided by the quadratic wall-normal reconstruction. *The final paper will show skin friction for other flat plate calculations at different angles to the mesh.*

B. NACA0012

Another test is laminar flow about a NACA 0012, also at $M_\infty = 0.5$, Reynolds number 5000, and zero angle of attack. Unlike the flat plate, in this case the stencil of the quadratic used in each cut cell is not in a fixed direction, but rotates around the airfoil. Figure 12 shows an overview of the discrete solution along with some examples of the quadratic's stencil near the leading and trailing edges. Figure 12b shows the velocity magnitude, computed on a grid with leading edge resolution of 0.0016c obtained

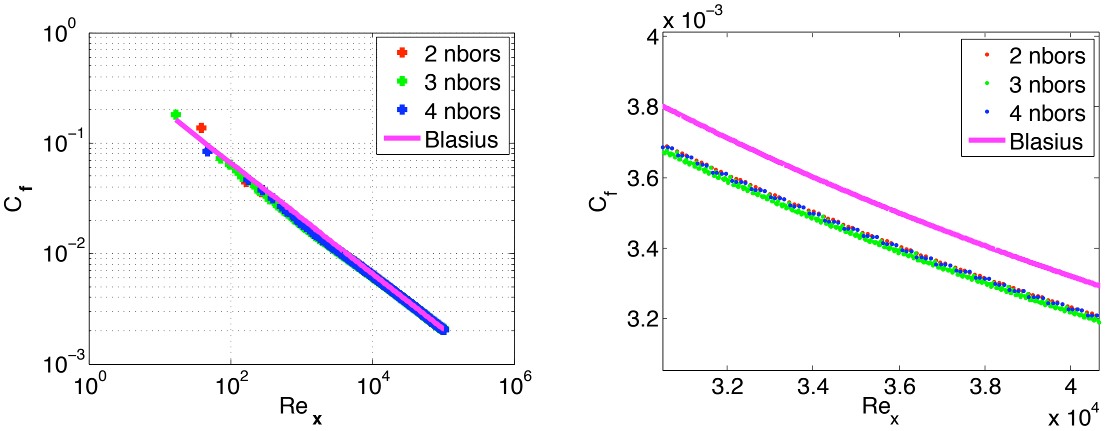


Figure 11. Skin friction for flat plate at 15^o to the mesh using the quadratic is well predicted by $Re_x = 100$. The zoom on the right shows much smoother results than is obtained with a linear wall boundary condition.

with 10 levels of refinement along the airfoil. Figures 13a and b show the pressure and skin friction coefficients around the body. Again the skin friction is smooth along the airfoil. Separation here occurs at $x = 0.816$, very close to the mesh converged values found in the literature.[34] *The final paper will also look at the angle of separation.*

C. Turbulent Flat Plate

While turbulent boundary layers are thicker than laminar boundary layers, the higher shear stress creates a greater demand for resolution near the wall. To verify that the approach taken so far carries over to the RANS equations and that no new showstoppers appear, we simulate a fully turbulent flat plate, again at 15° to the wall with $Re_L = 10^5$ using our simple eddy-viscosity model. The equations are integrated down to the wall – no wall functions are used. The isotropic Cartesian cells for this case have a mesh spacing of $h = 7.4 \times 10^{-4}$ at the wall. This is a factor of 8 finer at the wall than was used in the laminar case in example A.

Velocity profiles along the plate were sampled at $Re_x = 1 \times 10^5, 2 \times 10^5$ and 1×10^6. Figure 14a shows the three profiles, all nicely resolved on this mesh with no viscous overshoot. No limiters were used in this calculation. Figure 14b shows the velocity profile at $Re_x = 10^6$ plotted in wall coordinates and compared to Spalding's formula for a zero pressure gradient flat plate. The plus variables were calculated using the computed friction velocity to normalize the results. For this profile the first cut cell centroid corresponds to a $y^+ \approx 4.3$.

Figure 15 shows the skin friction, and also plots the expected asymptotic form $C_f \approx 0.0592 Re_x^{-1/5}$ from White.[35] The skin friction looks good by $Re_x = 10^4$. It is quite smooth, and does not show much grid irregularity. It does fall somewhat below the curve from White, primarily due to the relatively low Reynolds numbers. These results are comparable to other results in the literature.[36,37]

D. Final Examples

The final paper will include an RAE case 9 computation and a three dimensional example. We will use

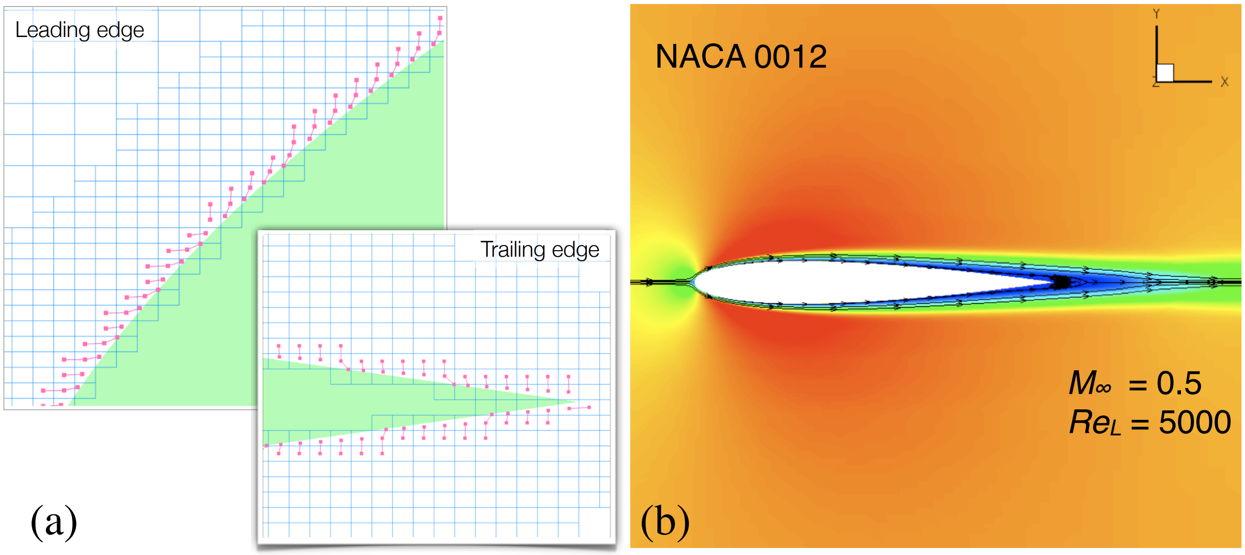


Figure 12. (a) Cells used in quadratic stencil for wall fluxes. (b) Velocity magnitude for NACA 0012.

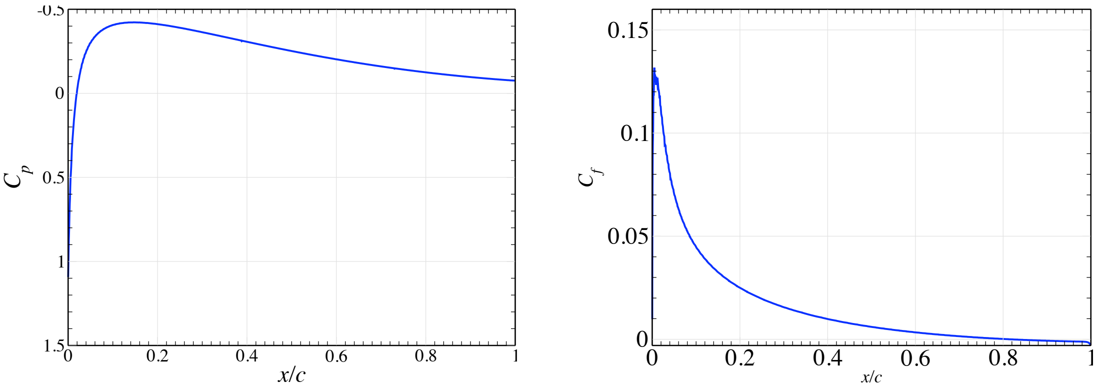


Figure 13. Cp (left) and Cf (right) for the results in Figure 12b for flow around a NACA 0012.

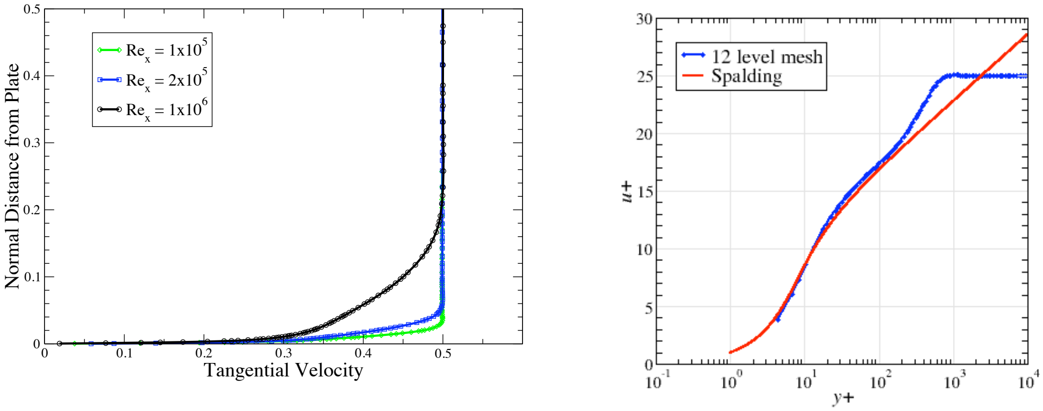


Figure 14. Turbulent flat plate with $Re_L = 10^5, M_\infty = 0.5$. Well-resolved profiles with no viscous overshoot (left) and overlaid with Spalding's formula in plus coordinates (right).

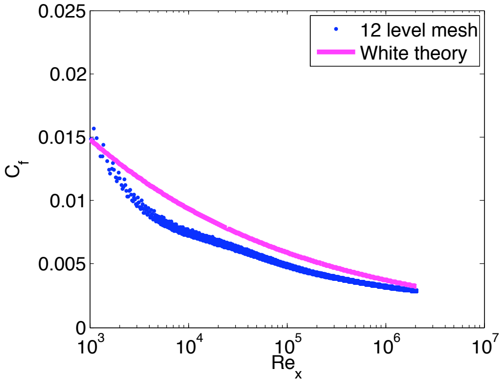


Figure 15. Skin friction for turbulent flat plate example at $M_\infty = 0.5$ and $Re_L = 1.0 \times 10^5$

the Spalart-Allmaras turbulence model instead of an algebraic model.

V. Conclusions

The final paper will contain additional calculations (a flat plate boundary layer at varying degrees to the mesh, a turbulent airfoil using the SA turbulence model, and a 3D example), as well as more algorithmic investigations (for example the use of normal and tangential instead of Cartesian coordinates in the recentering procedure). The final paper will contain preliminary results with a wall model to alleviate resolution requirements at the wall. The final paper will draw some conclusions.

Appendix

In this analysis we examine the effect on the allowable time step of using a quadratic instead of a linear interpolant to discretize the boundary condition for a model problem. We will use GKS theory[38] and consider the heat equation $u_t = u_{xx}$ on [0,1], with $u(0) = u(1) = 0$. Using forward Euler in time and central differencing in space gives $u_j^{n+1} = u_j^n + \frac{\Delta t}{h}((u_{j+1} - u_j)/h - (u_j - u_{j-1})/h)$, where the flux term is written in finite-volume form. The boundary condition $u(x = 0) = 0$ is usually implemented as $.5(u_1 + u_0) = 0$ where u_0 is a ghost cell. A simple calculation shows that this boundary conditions does not reduce the stability limit $\lambda = \Delta t/h^2 = .5$ of the initial value problem.

Using divided difference notation and the Newton form of the polynomial as in (5), the quadratic can be written

$$p_2(x) = [u_b] + [u_b, u_1]x + [u_b, u_1, u_2]\, x\, (x - x_1)$$

where we have written u_b to explicitly represent the boundary value at $x = 0$. The left flux at the first cell will be discretized using p_2' at the left cell edge, ${p_2}'(0) = (-u_2 + 9u_1)/3h$ so that the equation for u_1 is

$$u_1^{n+1} = u_1^n + \frac{\Delta t}{h}((u_2 - u_1)/h - p_2') \tag{11}$$

It is sufficient to consider the stability of the left half plane problem, and look for solutions of the form $u_j^n = \kappa^j z^n$. Substituting this into the difference schemes gives the characteristic equation

$$z = 1 + \lambda(\kappa - 2 + 1/\kappa) \tag{12}$$

where we are interested in solutions with $\kappa \leq 1$. The characteristic equation for the boundary condition (11) is

$$z = 1 + \frac{\lambda}{3}(4\kappa - 12). \tag{13}$$

Equating (12) and (13) give an l_2 solution $\kappa = 3 - 2\sqrt{3}$.

Substituting this root into (13) and solving for $z \leq 1$ gives $\lambda < \sqrt{3}/4 \approx .43$. This is only a small reduction in the stability limit for this model heat equation. Since viscous terms in the Navier-Stokes equations only reduce the allowable time step by a small fraction, the impact on the time step in practice would be correspondingly less.

Acknowledgements

Marian Nemec has been instrumental in advising us on this work throughout. We thank Tom Pulliam for providing ARC2D results along with fruitful and thought-provoking discussions. M.J. Berger was supported in part by DOE grant DE-FG02-88ER25053 and by AFOSR grant FA9550-06-1-0203.

References

[1]Wang, Z., "A Quadtree-Based Adaptive Cartesian/Quad Grid Flow Solver for Navier-Stokes Equations," *Computers Fluids*, Vol. 27, No. 4, 1998, pp. 529–549.

[2]Delanaye, M., Aftosmis, M., Berger, M., Liu, Y., and Pulliam, T., "Automatic Hybrid-Cartesian Grid Generation for High-Reynolds Number Flows around Complex Geometries," *AIAA-99-0777*, Jan. 1999.

[3]Jr., S. K., "SPLITFLOW: A 3D Unstructured Cartesian Prismatic Grid CFD Code for Complex Geometries," *AIAA-1995-343*.

[4]Wang, Z. and Chen, R., "Anisotropic Solution-Adaptive Viscous Cartesian Grid Method for Turbulent Flow Simulation," *AIAA Journal*, Vol. 40, Oct. 2002, pp. 1969–1978.

[5]Dawes, W., Harvey, S., Fellows, S., Favaretto, C., and Velivelli, A., "Viscous Layer Meshes from Level Sets on Cartesian Meshes," *AIAA-2007-555*.

[6]Nakahashi, K. and Kim, L., "Building-Cube Method for Large-Scale, High Resolution Flow Computations," *AIAA-2004-0423*, 2004.

[7]Gao, F., Ingram, D., Causon, D., and Mingham, C., "The Development of a Cartesian Cut Cell Method for Incompressible Viscous Flows," *Intl. J. Num. Methods in Fluids*, Vol. 54, 2007, pp. 1033–1053.

[8]Ghias, R., Mittal, R., and Dong, H., "A sharp interface immersed boundary method for compressible viscous flows," *J. Comp. Phys.*, Vol. 225, 2007.

[9]Kupiainen, M. and Sjogreen, B., "A Cartesian Embedded Boundary Method for the Compressible Navier-Stokes Equations," Tech. Rep. LLNL-JRNL-402504, 2009.

[10]Ollivier-Gooch, C., *Solution of the Navier-Stokes Equations on Locally-Refined Cartesian Meshes*, Ph.D. thesis, Stanford University, 1993.

[11]Fidkowski, K. J. and Darmofal, D., "An Adaptive Simplex Cut-Cell Method for Discontinuous Galerkin Discretiations of the Navier-Stokes Equations," *AIAA-2007-3941*.

[12]Iaccartino, G., Kalitzin, G., and Khalighi, B., "Towards an Immersed Boundary RANS Flow Solver," *AIAA-2003-0770*.

[13]Lee, J.-D. and Ruffin, S. M., "Development of a Turbulent Wall-Function based Viscous Cartesian-Grid Methodology," *AIAA-2007-1326*.

[14]Knopp, T., Alrutz, T., and Schwamborn, D., "A Grid and Flow Adaptive Wall-Function Method for RANS Turbulence Modelling," *J. Comp. Phys.*, Vol. 220, 2006, pp. 19–40.

[15]Kalitzin, G., Goradz, M., Iaccarino, G., and Durbin, P., "Near Wall Behavior of RANS Turbulence Models and Implications for Wall Functions," *J. Comp. Phys.*, Vol. 204, 2005, pp. 265–291.

[16]Capizzano, F., "A Turbulent Wall Model for Immersed Boundary Methods," *AIAA-2010-712*.

[17]Dadone, A., "Towards a Ghost Cell Method for Analysis of Viscous Flows on Cartesian Grids," *AIAA-2010-709*.

[18]Aftosmis, M., Melton, J., and Berger, M., "Robust and Efficient Cartesian Mesh Generation for Component-Based Geometry," *AIAA Journal*, Vol. 36, 1998.

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

[20]Berger, M., Aftosmis, M., and Murman, S., "Analysis of Slope Limiters on Irregular Grids," *AIAA-2005-0490*, Jan. 2005.

[21]Ji, H., Lien, F.-S., and Yee, E., "An Efficient Second-Order Accurate Cut-Cell Method for Solving the Variable Coefficient Poisson Equation with Jump Conditions on Irregular Domains," *Intl. J. Num. Methods in Fluids*, Vol. 52, 2006.

[22]Johansen, H. and Colella, P., "A Cartesian Grid Embedded Bounary Method for Poisson's Equation on Irregular Domains," *J. Comp. Phys.*, Vol. 147, 1998.

[23]Ye, T., Mittal, R., Udaykumar, H., and Shyy, W., "An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boudnaries," *J. Comp. Phys.*, Vol. 156, 1999.

[24]Coirier, W., *An Adaptively Refined, Cartesian, Cell-Based Scheme for the Euler and Navier-Stokes Equations*, Ph.D. thesis, University of Michigan, 1994.

[25]Fay, J. and Riddell, F., "Theory of Stagnation Point Heat Transfer in Dissociated Air," *J. Aeronaut. Sci.*, Vol. 25, 1958, pp. 73–85.

[26]Steinthorsson, E., Modiano, D., and Colella, P., "Computations of Unsteady Viscous flows Using Solution Adaptive Mesh Refinement in Curvilinear Body-Fitted Grid Systems," *AIAA-94-2330*, June 1994.

[27]Dragojlovic, Z., Najmabadi, F., and Day, M., "An Embedded Boundary Method for Viscous Conducting Compressible Flow," *J. Comp. Phys.*, Vol. 216, 2006, pp. 37–51.

[28]Marshall, D., *Extending the functionalities of Cartesian Grid Solvers: Viscous Effects, Modeling and MPI Parallelization*, Ph.D. thesis, Georgia Institute of Technology, 2002.

[29]Hu, P., Zhao, H., Kamakoti, R., Dittakavi, N., Xue, L., Ni, K., Mao, S., Marshall, D., and Aftosmis, M., "Towards Efficient Viscous Modeling Based on Cartesian Methods for Automated Flow Simulation," *AIAA-2010-1472*.

[30]Cebeci, T. and Smith, A., *Analysis of Turbulent Boundary Layers*, Academic Press, 1974.

[31] Aftosmis, M., Berger, M., and Murman, S., "Applications of Space-filling Curves to Cartesian Methods for CFD," *AIAA-2004-1232*.

[32] Toro, E., *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer, 1997.

[33] Batten, P., Clarke, N., Lambert, C., and Causon, D., "On the Choice of Wave Speeds for the HLLC Riemann Solver," *SIAM J. Sci. Computing*, Vol. 18, No. 6, Nov. 1997, pp. 1553–1570.

[34] Aftosmis, M., "Upwind Method for Simulation of Viscous Flow on Adaptively Refined Meshes," *AIAA Journal*, Vol. 32, No. 2, Feb. 1994.

[35] White, F. M., *Viscous Fluid Flow*, McGraw Hill, 1974.

[36] Nichols, R. and Nelson, C. C., "Wall Function Boundary Conditions Including Heat Transfer and Compressibility," *AIAA Journal*, Vol. 42, No. 6, 2004.

[37] NPARC Consortium, `http://www.grc.nasa.gov/WWW/wind/archive.html`.

[38] Gustafsson, B., Kreiss, H.-O., and Sundström, A., "Stability Theory of Difference Approximations for Mixed Initial Boundary Value Problems. II," *Math. Comp.*, Vol. 26, 1972, pp. 649–686.