50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition
Jan. 9-12, 2012. Nashville, TN

AIAA 2012-1301

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

Marsha Berger
NYU
251 Mercer St.
NY, NY 10012
berger@cims.nyu.edu, Senior Member, AIAA.

Michael J. Aftosmis
NASA Ames Research Center
Moffett Field, CA 94035
michael.aftosmis@nasa.gov, Associate Fellow, AIAA.

with an Appendix by
Steven R. Allmaras
allmaras@adscfd.com, Senior Member, AIAA.

The World's Forum for Aerospace Leadership

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

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

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

with an Appendix by **Steven R. Allmaras**[1]

We present preliminary development of an approach for simulating steady viscous compressible flow in two space dimensions using a Cartesian cut-cell finite volume method. We consider laminar and turbulent flow with both low and high cell Reynolds numbers near the wall. The approach solves the full Navier-Stokes equations in all cells and uses a wall model to both address the resolution requirements near boundaries and to mitigate mesh irregularities in cut cells. We present a quadratic wall model for low cell Reynolds numbers. At high cell Reynolds numbers, the quadratic is replaced with a newly developed analytic wall model stemming from solution of a limiting form of the Spalart-Allmaras turbulence model which features an explicit evaluation for flow velocity, and exactly matches characteristics of the SA turbulence model in the field. We develop multigrid operators which attain convergence rates similar to inviscid multigrid. Investigations focus on preliminary verification and validation of the method. Flows over flat plates and compressible airfoils show good agreement with both theoretical results and experimental data. Mesh convergence studies on sub- and transonic airfoil flows show convergence of surface pressures with wall spacings as large as ∼0.1% chord. With the current analytic wall model, mesh converged skin friction requires near-wall cells two to four times smaller.

I. Introduction

Cartesian embedded-boundary grids have proven extremely useful for inviscid flow simulations around complex geometries. Their primary strengths include their accuracy, rapid turnaround time and level of automation. 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.

These same issues are less well understood for high Reynolds number 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 irregularity of the cut cells, along with loss of accuracy and numerical stiffness.[1–3] 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 the wall normal direction is clearly a 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 layers of conformal cells before switching to a background Cartesian mesh.[4–6] Some authors accomplish this through the use of body-fitted grids in the near-wall region and then blending 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.[7,8] Both methods rely on body-fitted near-wall layers to avoid the large cell counts associated with isotropically refined Cartesian grids, but give up the simplicity and robustness of a pure Cartesian approach. An alternate idea is to use the Cartesian

* *berger@cims.nyu.edu.* Senior Member, AIAA.
† *michael.aftosmis@nasa.gov.* Associate Fellow, AIAA.
[1] *allmaras@adscfd.com.* Senior Member, AIAA.

mesh down to the wall. This can be done using either a finite volume[9,10] or finite difference[11,12] scheme, or other integration methods[13,14] but all will have to contend with the non-aligned boundary. Some methods use an immersed boundary[15,16] instead of explicitly cutting the cells that intersect geometry. For any of these approaches to be ultimately practical, some additional technique must be used to alleviate the cell count problem of isotropic refinement for high Reynolds number flows.

Each of these approaches has 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. Fully Cartesian approaches preserve the decoupling between the volume mesh and the surface triangulation. This simplicity is one of the great strengths of embedded-boundary methods and is key for automation. Using only Cartesian faces also allows optimized numerics within the flow solver.

Recent papers present promising results using wall function or PDE-based wall model approaches to mitigate the inefficiency of isotropically refined Cartesian meshes in a boundary layer.[17–23] We adopt this same approach within the current work, however, it must be suitably modified for finite volume meshes with Cartesian cells explicitly cut by the embedded boundary. In this preliminary effort, we model near-wall behavior with either a simple quadratic representation of the solution (low cell Reynolds numbers) or with an analytic wall function (high cell Reynolds numbers) as our initial wall model for turbulent flow. While wall functions are sufficient for this initial exploratory work, ultimately we envision developing a more complete PDE-based subgrid wall model.

In this paper we lay the groundwork in several steps. We first develop a Navier-Stokes solver for laminar flows, by developing an accurate discretization for the cut cells and mesh interfaces. This is discussed in Section II and Appendix A. It entails an understanding of grid irregularity at the cut cells for second-derivative terms. Of particular importance is the use of a quadratic to compute the values in the cut cells (for both the solution and its derivatives) needed for the flux computations in laminar flow. This work builds upon our earlier efforts[24,25] where the governing equations are integrated to steady-state using a second-order cell-centered finite volume scheme. Section II also presents the RANS equations and the Spalart-Allmaras turbulence model. The coupling of the wall model to the finite volume scheme used in the turbulent flow computations is an important component. Section III discusses fortification of the basic multigrid solver for viscous flow. Section IV presents two-dimensional computational examples of both laminar and turbulent flow, including flat-plate boundary layers on both coordinate-aligned and non-aligned meshes, and compressible airfoils.

II. Discretization of the Navier-Stokes Equations

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

$$\frac{d}{dt}\iint_{\Omega} U\,dA + \oint_{\partial\Omega} (F\hat{\imath} + G\hat{\jmath}) \cdot \hat{n}\,dS = \oint_{\partial\Omega} (F_v\hat{\imath} + G_v\hat{\jmath}) \cdot \hat{n}\,dS \tag{1}$$

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

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

where the vector ∇q is proportional to the gradient of the temperature $\nabla q = -k\nabla T$, the Prandtl number is given by $Pr = \mu c_p/k$, and μ is computed using Sutherland's law.[26] 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.[25] 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 at the faces of these cut cells.

A. 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. Consider for simplicity an x face. We ensure stability and accuracy by taking $u_x \approx \frac{u_{i+1,j}-u_{i,j}}{\Delta x}$ at that face. This avoids the pitfall of simply averaging the left and right cell-centered gradients (already computed using second-order central differences for the inviscid terms in a second-order finite volume scheme). This would lead to an unstable 5-point approximation to the second derivative with a decoupled stencil. The transverse derivative however, in this case u_y at the x face, can stably be taken as the average from the adjacent cells, $u_y \approx .5(u_{y_{i,j}} + u_{y_{i+1,j}})$. This discretization is commonly found in the literature for regular Cartesian meshes.[27,28]

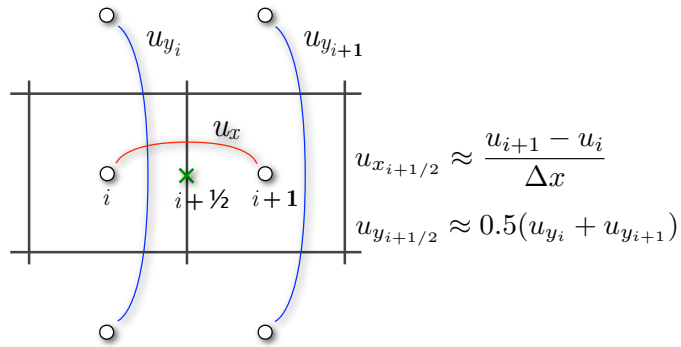


Figure 1. Illustration of stable viscous flux computation at face between cells i and $i+1$.

At mesh refinement interfaces, a least-squares gradient computation that uses only face neighbors is linearly exact, and so approximates the solution to second order. We use this gradient to recenter the solution to compute the fluxes at mesh interfaces in the same way as is done in the cut cells (described next). Since it is not centered, this difference approximation gives only a first-order accurate approximation to the derivative at an interface. The transverse component of the gradient is still taken to be an average of the transverse gradients on either side of the face, this time weighted by distance from the face.

B. Cut-Cell Discretization for Navier-Stokes

At cut faces, a recentering procedure illustrated in Fig. 2 is used to reconstruct the velocity from the cell centroid to the perpendicular line through the face centroid. Appendix A presents an accuracy study of the discretization with recentering compared with two other popular methods using a Poisson equation model problem. This study shows second-order accuracy in both the L_1 and max norms, which is on a par with the best of these methods.

When the face lies 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 one cut cell needs to recenter the solution, since the full Cartesian cell centroid and the face centroid must already be aligned. Pressure at the wall is obtained by simple reconstruction from the cell centroid to the surface centroid of the wall in each cut cell.

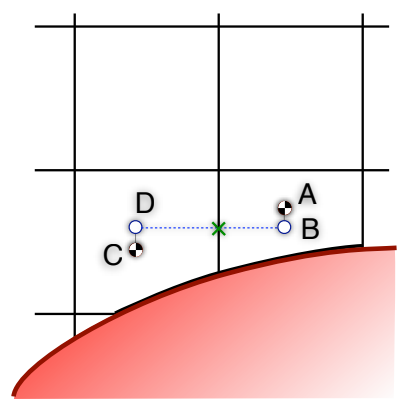


Figure 2. Recentering of the solution from cell centroids (A and C) to points (B and D) on the line perpendicular to the midpoint of the cut-cell edge. A simple difference (B-D)/Δx approximates the derivative.

Mesh irregularity through the cut cells introduces non-smooth truncation errors along walls. Accordingly, special procedures must be devised for accurate reconstructions of velocity derivatives at wall boundaries. For meshes with low cell Reynolds numbers at the wall, we obtain accuracy by using a higher-order (quadratic) polynomial for reconstruction of the data in the wall-normal direction. In RANS simulations (discussed later) with high cell Reynolds numbers, the quadratic is replaced by an appropriate wall model.

1. *Mesh Irregularity at 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 skin friction computed at the wall. Accurate skin friction, in particular, has been a challenge, and has stymied many viscous Cartesian efforts. Figure 3 displays the problem and is representative of many of the results found in the literature.[1–3] 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 face 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.

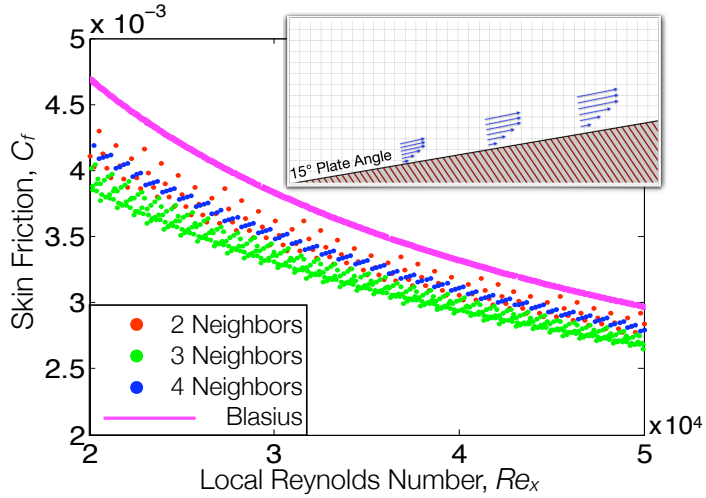


Figure 3. Noise in the skin friction profile along surface of the plate when using piecewise linear gradients in the cut cells, as in the literature.[1–3]

Figure 4 reveals the source of this irregularity. 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 $U'(d)$. With a non-linear profile, the slopes at a–d are obviously not $(dU/dy)_{wall}$, and the "noise" staircasing through the profile in Fig. 3 is not surprising. In the skin friction plot, the cell-centroid gradients are being projected along the wall, even though the centroids in this non-uniform region are at different distances from the wall.

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.

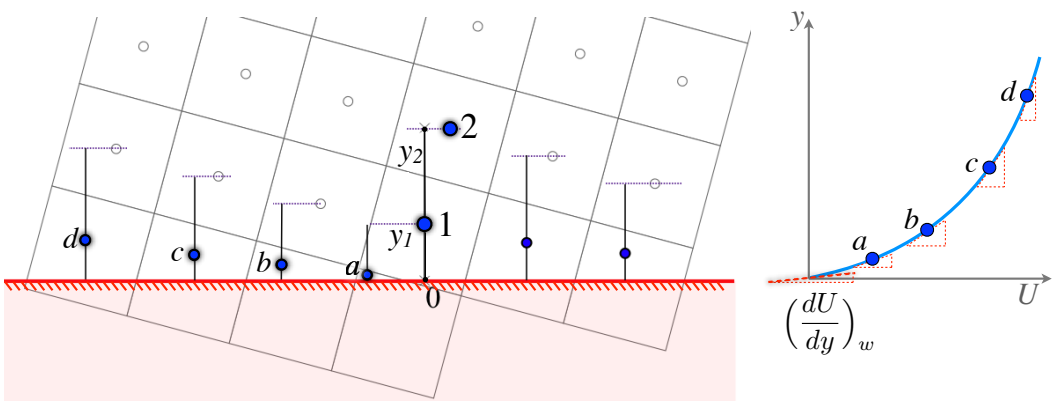


Figure 4. Illustration of a quadratic interpolant through the cut cells on a non-aligned mesh.

2. *Near-Wall Quadratic Reconstruction for Low Cell Reynolds Numbers*

As suggested above, it is a simple matter to fit a quadratic to three values, particularly since both the u and v velocity is zero at the wall. Using the notation of Figure 4, we construct a line through cell 1's centroid, normal to the patch of the wall contained in cell 1. Cell 2 is the neighbor across the face intersected by this normal. 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 as well. However, since streamwise gradients in high Reynolds number flows are generally several orders of magnitude smaller than wall-normal gradients, the current implementation uses data at 2 directly.

The resulting formula is

$$U(y) = yA + y(y - y_1)B \tag{3}$$

with

$$A = \frac{u_1}{y_1}, \quad \text{and} \quad B = \frac{\frac{u_2-u_1}{y_2-y_1} - A}{y_2 - y_0} \tag{4}$$

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

$$U'(y) = A + (2y - y_1)B. \tag{5}$$

This gives a slope at the wall of

$$\left(\frac{\partial U}{\partial y}\right)_{wall} \approx U'(0) = A - y_1 B \tag{6}$$

The slope at the wall provides the wall shear stress for both viscous fluxes and output skin friction, after rotating into the tangential direction. The quadratic also provides the u and v states for differencing at the cut faces, needed for the shear stress. In addition, the gradient of the quadratic evaluated at the cell centroid replaces the least-squares gradient 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 Fig. 4). Although these cells are full hexahedra, they also have boundary-dependent stencil irregularity since they are adjacent to the cut cells. The use of quadratic reconstruction mitigates the mesh irregularity while maintaining strict conservation.[2]

Of possible concern in using the quadratic to discretize wall shear is that it may adversely impact the allowable time step. Appendix B contains a stability analysis for a model 1D heat equation to determine the CFL limit for this quadratic boundary condition. Results using GKS stability analysis[29] show that using forward Euler in time and central differencing in space results in a CFL 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. Our choice of time step at the cut cells is sufficiently conservative and this minor reduction is not an issue.

C. RANS equations

Compressible high Reynolds number turbulent flows are modeled using the Favre-averaged Navier-Stokes equations (herein still referred to as the RANS equations). They are of the same form as (1) 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 an eddy-viscosity parameter μ_t and a turbulent Prandtl number Pr_t which we take to be 0.9, giving shear stresses

$$\begin{aligned} \tau_{xx} &= \frac{2}{3}(\mu + \mu_t)(2u_x - v_y) \\ \tau_{xy} &= (\mu + \mu_t)(u_y + v_x) = \tau_{yx} \\ \tau_{yy} &= \frac{2}{3}(\mu + \mu_t)(2v_y - u_x) \end{aligned} \tag{7}$$

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

We model μ_t using the Spalart-Allmaras turbulence model[30] in fully turbulent form (*i.e.* no transition or tripping terms). We use what is referred to as the *baseline model* described in Oliver's 2008 dissertation.[31]

[2] 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.

Added to this are terms for the so-called *negative model*, which modifies some of the functional forms in the production and destruction terms for negative values of $\tilde{\nu}$ to help push the turbulent viscosity back to the positive region.[31,32]

The resulting equation in the incompressible formulation is

$$\frac{\partial}{\partial t}(\tilde{\nu}) + \frac{\partial}{\partial x_j}(u_j\tilde{\nu}) = \frac{1}{\sigma}\left[(1+c_{b2})\Big(\frac{\partial\tilde{\nu}}{\partial x_j}\Big)^2 + (\nu+\tilde{\nu}f(\chi))\frac{\partial^2\tilde{\nu}}{\partial x_j^2}\right] \;+\; \text{Production} \;-\; \text{Destruction} \tag{8}$$

where we made the usual assumption that $\frac{\partial\nu}{\partial x_j}$ is small and can be neglected in rearranging the diffusion terms as above. For this somewhat modified negative model we take

$$f(\chi) = \begin{cases} 1 & \chi > 0 \\ 1+\frac{\chi}{2} & \text{otherwise} \end{cases}, \qquad \text{Production} = \begin{cases} c_{b1}\,\hat{S}\,\tilde{\nu} & \chi > 0 \\ c_{b1}\,\Omega\,g_n\,\tilde{\nu} & \text{otherwise} \end{cases}, \tag{9}$$

with

$$\hat{S} = \begin{cases} \Omega + \bar{S} & \bar{S} > -c_{v2}\,\omega \\ \Omega + \frac{\Omega\,(c_{v2}^2 + c_{v3}\bar{S})}{(\Omega\,(c_{v3}-2c_{v2})-\bar{S})} & \text{otherwise} \end{cases} \tag{10}$$

and $\bar{S} = \frac{\tilde{\nu}f_{v2}}{K^2d^2}$, where Ω is the magnitude of vorticity. For the destruction term we have

$$\text{Destruction} = \begin{cases} c_{w1}\,f_w(\frac{\tilde{\nu}}{d})^2 & \chi \geq 0 \\ -c_{w1}\,(\frac{\tilde{\nu}}{d})^2 & \text{otherwise} \end{cases} \tag{11}$$

with $\mu_t = \rho\tilde{\nu}f_{v1}$ and the usual definitions of f_w, g, r, χ, and the other constants.[30]

This negative model improves on and expands the robustness measures in the original and baseline models.[30,31] Oliver applied the negative model in a GMRES framework.[31] In our multigrid context, we found some of the modifications for negative $\tilde{\nu}$ overly steep and reduced them somewhat while still following the same design principles. Reducing them substantially improved the depth of convergence of our multigrid smoother. We have not done extensive study in arriving at the final functional forms. All equations – Navier Stokes plus turbulence model – are solved in a fully coupled manner as described in Section III.

One additional change was necessary that is not commonly discussed in the literature. The advective terms in the turbulence model are usually implemented using a first-order scheme to help preserve positivity. This was too diffusive on non-streamwise-aligned meshes. The first-order scheme was replaced by a second-order method with linear reconstruction to discretize the advective terms.

The Spalart-Allmaras turbulence model uses the normal distance to the wall in the destruction term – the d in (11) for each cell in the mesh. We use Sethian's fast marching method[33] to compute this additional quantity. This is mostly standard, with only a few modifications needed for our multi-level cut-cell mesh. This is incorporated into our mesh generator as a preprocessing step before computing the flow.

1. *Cut-Cell Wall Model for High Cell Reynolds Numbers*

One of the obvious challenges facing Cartesian methods for RANS simulations is the untenable cell counts associated with resolving the viscous stresses all the way down to the wall. Subgrid wall modeling is based on the fact that very near the wall, momentum transfer is governed by viscous stresses and the near wall flow is distinct from the outer flow in which inertial forces dominate. Wall functions, which are essentially algebraic wall models, are based on this observation. While wall-function development has been active[34] since the 80's, recent work has focused on extending them for numerical approaches using subgrid modeling for both body-fitted[18–20] and non-body-fitted grids.[21,22] An advantage of these approaches is that the streamwise flow gradients can be taken from the underlying grid cells, thus removing many of the restrictions of algebraic wall models with built-in assumptions about the underlying flow conditions (zero pressure gradient, etc.) These *two-layer* models can also couple more flexibly to the outer RANS flow. The the diffusion model of Bond and Blottner[23,35] is one example in which a system of ODEs is solved instead of assuming an analytic form for the wall function. The issues then become deciding how far from the wall the coupling occurs and which terms are retained in the equations that are solved in the wall layer.

While our eventual goal is to develop a complete subgrid model, in this paper we first examine the issues that arise when extending the use of wall functions to non-coordinate-aligned grids with cut cells at the wall. We use a new wall model recently developed by Steven Allmaras. The derivation of the model is described in Appendix C of this paper, written by Allmaras. We repeat the final equation here for convenience. (Note that the ArcTan arguments are the reverse of the usual, see Appendix C).

$$\begin{aligned} u^+(y^+) \;=\; \bar{B} \;+\; c_1 \log((y^+ + a_1)^2 + b_1^2) \;-\; c_2 \log((y^+ + a_2)^2 + b_2^2) \\ -\; c_3 \mathrm{ArcTan}(y^+ + a_1, b_1) \;-\; c_4 \mathrm{ArcTan}(\mathrm{y}^+ + \mathrm{a}_2, \mathrm{b}_2), \end{aligned} \tag{12}$$

where the constants are given in the Appendix. Several other researchers[16,36] have used Spalding's composite formula, an algebraic relation bridging the viscous sublayer and the log layer within a fully developed turbulent boundary layer, as a wall function. For completeness we give Spalding's formula here too,

$$y^+(u^+) = u^+ + e^{-\kappa B}\left(e^{\kappa u^+} - 1 - \kappa u^+ - \frac{1}{2}(\kappa u^+)^2 - \frac{1}{6}(\kappa u^+)^3\right) \tag{13}$$

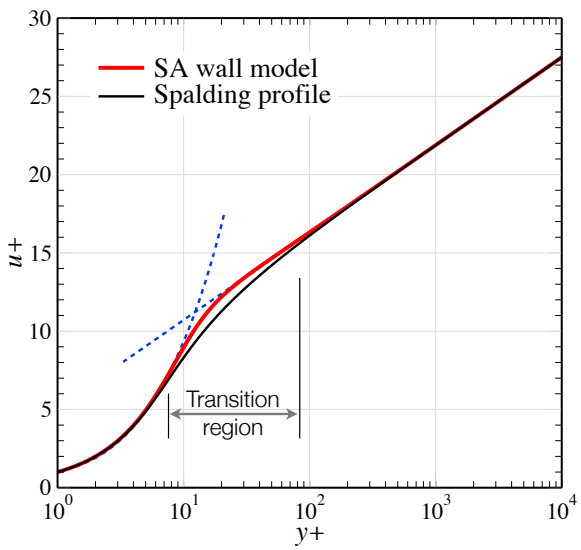


Figure 5. Comparison of new SA wall model with Spalding's formula. Most of the difference is in the transition region.

Equation (12) (henceforth called the SA wall model) has the advantage over Spalding's formula of matching the profiles actually computed in the flow field by the Spalart-Allmaras turbulence model. This is seen in the computational examples of Section C, where the computed profiles match the SA model through the transition region. Figure 5 shows the transition region, where the two formulas differ the most. In this figure, Spalding's formula was evaluated using $B = 5.033$ in (13) to match the asymptotic behavior of the SA model, which produces this value for the constant shift. Spalding's formula simply bridges this buffer zone with a functional form. Furthermore, (12) is more computationally efficient since given y^+, u^+ is explicitly evaluated rather than needing a nonlinear iteration as (13) does. (However Kalitzin et al.[18] have a nice approach that avoids iteration using pre-computed tables of inverses). By using a wall model that actually matches the background flow we can avoid the kinetic energy mismatch, described in Sondak and Pletcher.[37]

2. *Coupling*

Since the model is given in wall coordinates (y^+, u^+), $y^+ = \frac{y u_\tau}{\nu}$, $u^+ = \frac{u}{u_\tau}$, an estimate of the friction velocity u_τ (a.k.a. $\frac{\partial u}{\partial y}|_{wall}$) is needed for each cut cell. Wall functions have a well-known sensitivity to the location of the first point away from the wall,[34] and this issue must be directly addressed for the wildly irregular cut cells of an embedded-boundary mesh. Following Capizzano[21] we regularize the distance by using points a fixed distance h away from the wall instead of using the cut cell centroid.

We fit the one-dimensional model in (12) using the boundary condition of zero velocity on the wall and the solution at a fixed point F (to use the same terminology[21]) located a distance h from the boundary in the normal direction. Let the point F be in cell D; the approximate solution there is obtained using linear reconstruction from D's centroid. The velocity is then rotated into q_{tang}, the tangential velocity, using the directions defined by the portion of the boundary in cut cell C. A Newton iteration is used to find the friction velocity u_τ that transforms the point $(h, q_{tang}(F))$ into $(h^+, q^+_{tang}(F))$ lying on the SA wall model curve (12). Gradients of the model provide the viscous flux needed at the wall for the finite volume scheme. The model also gives values of the tangential velocity needed to compute the difference at the cut faces (after rotating back to Cartesian velocity components u and v). Finally, as with the quadratic we replace the cut-cell least squares gradient with a model gradient (and set the tangential component to zero). This coupling strategy makes use of the fact that the friction velocity is constant through the inner portion of the boundary layer. The procedure is illustrated in Fig. 6.

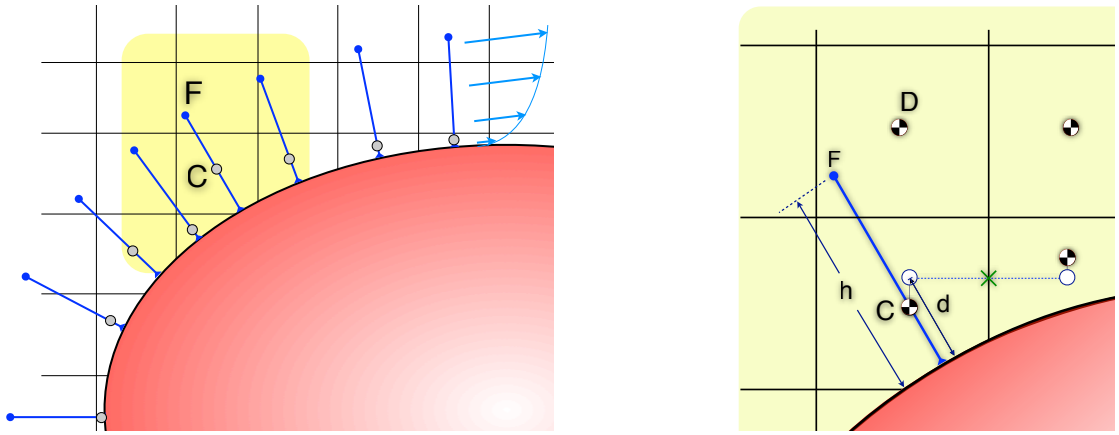


Figure 6. Illustration of construction for wall model coupling via constant height "forcing" points through the centroids of the cut cells.

Currently we take the distance $h = 1.5 \times \min(dx, dy)$. Since the cell sizes in the coordinate directions are approximately equal, this distance is slightly larger than a diagonal in a two dimensional cell, and so is sufficient to situate the point F out of the cut cell and into the neighboring cell. This procedure does impose some restrictions on the mesh resolution, since the point F must lie in the logarithmic region of the boundary layer to get valid estimates of the friction velocity. Eventually this could be used as a criteria for adaptation, or alternatively the location of the point F could vary depending on the local flow conditions.

Section IV.C shows results of coordinate-aligned turbulent flat plate simulations with Reynolds number $Re_L = 5 \times 10^5$ using both approaches – integrating down to the wall and using the wall model. The latter needs substantially less resolution. We will also show a comparison to the more difficult case where the flat plate is rotated with respect to the mesh. The wall model computation needs an extra level of refinement in the non-coordinate-aligned case. These preliminary studies show savings commensurate with the savings found in body-fitted grid implementations of wall models.[19,38]

III. Multigrid Acceleration and Time-Stepping

Our steady-state solution strategy uses an FAS multigrid method with a multi-stage Runge-Kutta smoother applied on each multigrid level. It is common for viscous flows on highly anisotropic grids to cause convergence problems for multigrid in getting to steady-state. However, since the Cartesian cells in this work are essentially isotropic, appropriately strengthened multigrid operators should yield Navier-Stokes performance on a par with our inviscid multigrid algorithm.[25] In RANS simulations, the equation for the turbulence model also needs time advancement until a steady state is reached. Since this equation has stiff source terms that may restrain the rest of the system, we use a separate timestep for this equation as described below. The system is solved fully coupled, with the Runge-Kutta smoother operating on all equations, but using one timestep for the flow equations and another for the turbulence model.

A. Time-Stepping

We compute directional timesteps for the flow equations following the usual advection-diffusion model.

$$\Delta t_x = \frac{h_x}{|u| + c + \frac{K_v}{h_x}}, \qquad \Delta t_y = \frac{h_y}{|v| + c + \frac{K_v}{h_y}}, \tag{14}$$

where c is the local sound speed and

$$K_v = 2 \max\left(\frac{4}{3}(\mu + \mu_\tau),\ \gamma \frac{\mu + \mu_\tau}{Pr}\right) \tag{15}$$

The timestep within the cell is a harmonic blend of the directional Δt's.

$$\Delta t = \frac{\Delta t_x \Delta t_y}{\Delta t_x + \Delta t_y} \tag{16}$$

Following Pierce,[39] the use of a separate timestep for the turbulence model in a fully coupled time-advance scheme amounts to a diagonal timestep formulation, essentially applying a scalar preconditioning to the turbulence model. The timestep for the SA working variable, $\tilde{\nu}$, may be written concisely by replacing K_ν with $K_{\nu\mathrm{SA}}$

$$K_{\nu\mathrm{SA}} = 2(\nu + |\tilde{\nu}|)\Big(\frac{1 + C_{b2}}{\sigma}\Big) \tag{17}$$

where σ is 2/3. This results in the directional timesteps

$$\Delta t_x = \frac{h_x}{|u| + \frac{c}{8} + \frac{K_{\nu\mathrm{SA}}}{h_x}}, \qquad \Delta t_y = \frac{h_y}{|v| + \frac{c}{8} + \frac{K_{\nu\mathrm{SA}}}{h_y}}. \tag{18}$$

The factor $\frac{c}{8}$ is an *ad-hoc* inclusion for robustness that prevents the hyperbolic contribution from vanishing.[39] The directional timesteps for the turbulence model are again blended via the harmonic mean in (16).

The source term ("Production - Destruction") is treated implicitly by penalizing the timestep when its Jacobian, $Q_{\tilde{\nu}}$, is negative.[30,39]

$$\Delta t_{\mathrm{imp}} = \frac{\Delta t}{1 - \Delta t \min(0, Q_{\tilde{\nu}})} \tag{19}$$

B. Second-Order Coarse Grid Operator

Inviscid multigrid solvers commonly use a first-order spatial differencing on coarser grids. This saves the expense of computing gradients, since the coarse grid solution does not affect the final solution. Without gradients, the only geometric information required is the area of the cut faces, the cell volumes, and the wall normal vector in each coarse cell. By contrast, a second-order coarse operator needs solution gradients to perform linear reconstruction to coarse grid cell faces. Near the wall, this requires the cut-cell centroids, cut-face centroids, and surface (wall) centroids on the coarser grids. Most of this information is easily computed from the fine grid and can be done concurrently with coarse mesh generation. For example, coarse grid cell centroids are weighted averages of all the fine grid cells that restrict to that cell. One of the difficulties is assigning the triangles associated with cut cells on the fine grid to the proper coarsened Cartesian cells.

Specifically, the most difficult part of the coarse grid generation is organizing split cells on-the-fly during mesh coarsening. Split cells are cut Cartesian cells that are split into multiple control volumes by the geometry. Figure 7 shows two interesting examples. Cut cells on the fine grid may agglomerate into split cells on the coarse mesh (Fig. 7a). Alternatively, split cells on the fine grid can yield a cut (but unsplit) coarse cell (Fig. 7b). Specific situations like these may be quite complex in three dimensions since they can involve any number of control volumes. Nevertheless, all such cases can be resolved using a simple integer matching algorithm. This algorithm scans for common triangles on sorted lists of intersected triangles maintained by each cut cell, along with the face lists that connect cut cells with their neighbors. Since it is based on integer comparisons of sorted lists, this matching procedure is very fast and robust.

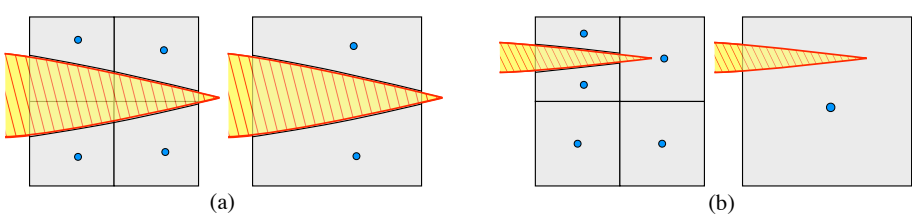


Figure 7. (a) Four fine cut cells that make one coarse split cell containing two control volumes. (b) A fine split cell, a cut cell, and two full cells that make up one coarse cut (but not split) coarse cell.

C. Linear Prolongation Operator

The other major change in the multigrid algorithm is the formulation of the prolongation operation. Multigrid theory shows that higher-order derivatives need better prolongation than the piecewise constant approach

typical of inviscid flow with cell-centered schemes.[40] Our restriction operator, a volume-weighted average of the solution and residual on the fine grid, is already second-order, and a second order coarse grid discretization was also implemented. With gradients now defined on the coarse levels it is easy to 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 modified solution after recursively smoothing on the coarser levels. Thus, in addition to the computational expense, there is some minor additional memory overhead from storing gradients on the coarser levels.

Figure 8 compares convergence of the multigrid scheme before and after improvement for the Navier-Stokes equations (1). The black curve shows the original formulation without coarse gradients and using only piecewise constant prolongation. The blue curve shows the improved operator using a second-order coarse grid scheme combined with a linear prolongation operator. While this particular data is for the laminar ($Re_L = 5000$) flow over a NACA 0012 airfoil at $M_\infty = 0.5$ (presented in Section IV.B) the improvement is characteristic of the scheme's performance. In this example, six levels of multigrid were used with a W-cycle consisting of one pre- and one post-sweep of a 5 stage Runge-Kutta smoother on each level. With linear prolongation and coarse grid gradients, the scheme remained stable with a CFL number of about 1.4. In contrast, the unmodified scheme in Fig. 8 had a maximum stable Courant number of only about 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 improve multigrid performance. Figure 8 shows that when used in combination we can achieve convergence behavior nearly on a par with that of the base inviscid scheme previously reported.[25]

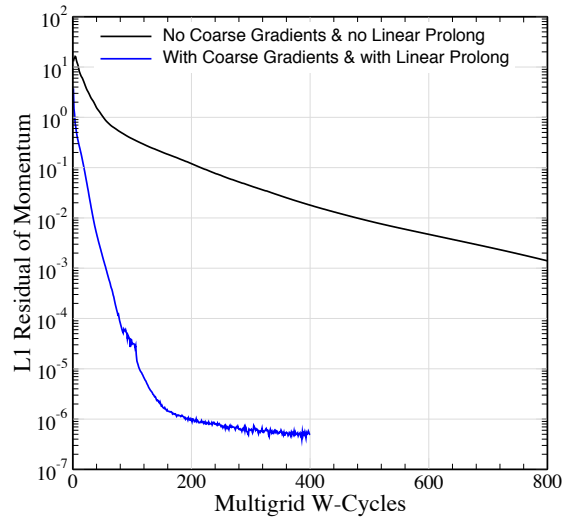


Figure 8. Multigrid convergence for NACA 0012 with $M_\infty = 0.5$, and $Re_L = 5000$ at zero angle of attack. The figure compares the use of linear prolongation and coarse grid gradients with piecewise constant prolongation and no coarse gradients. Six levels of multigrid were used.

Finally, Fig. 9 shows multigrid convergence behavior of a fully turbulent RANS example with the Spalart-Allmaras turbulence model. The fully coupled time advance is using the diagonal timestep described earlier in combination with the second-order coarse discretization and linear prolongation operator. The case shown corresponds to the $M_\infty = 0.676$ fully turbulent flow over an RAE 2822 airfoil at a Reynolds number of 5.7×10^6 presented in Section IV.D (medium mesh, AGARD Case 1) computed with 5 levels of multigrid. Cells at the wall are about three times finer than in the NACA example of Fig. 8. While the additional stiffness introduced by the turbulence model has clearly impacted convergence, the scheme is still delivering roughly 9 orders-of-magnitude reduction in the L_1 norm of momentum in under 1000 (fine-grid) multigrid cycles for both the flow and turbulence model. This corresponds to a multigrid convergence rate of about 0.98. While this is noticeably slower than the rate of roughly 0.92 shown in laminar case (Fig. 8), it is still fast enough to support our numerical investigations. With approximately 115,000 cells in the mesh, this example took about 5 minutes on a current generation desktop computer. The literature on block-Jacobi preconditioning and matrix time-stepping offers many suggestions for further improvement of these results.

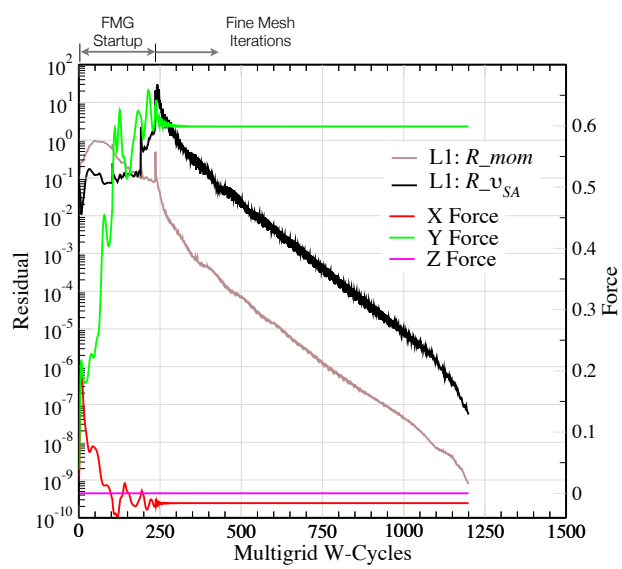


Figure 9. Multigrid convergence for fully turbulent flow over an RAE 2822 airfoil at $M_\infty = 0.676$ and Reynolds number 5.7×10^6 (see Section IV.D) using a five level multigrid W-cycle.

IV. Computational Examples

In this early investigation, computational results are principally confined to verification and validation exercises. Our goal is to investigate the accuracy of the viscous discretization operator on cut-cell Cartesian meshes and examine fundamental issues surrounding stencil irregularity, wall modeling and resolution requirements. We do not focus on performance, since we have not yet seriously begun to attack the cell count or meshing efficiency issues.

A. Laminar 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 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. As discussed in Section II.2, these laminar computations all used quadratic reconstruction in the wall-normal direction to obtain the necessary derivatives in the cut cells.

Figure 10 shows a sketch of the problem setup (top left) along with profiles of both tangential and normal velocity which were extracted from solutions 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 similarity 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 on the right used the van Leer limiter and have no viscous overshoot. Limiters can help control the overshoot, but the extra dissipation can adversely affect skin friction. These calculations used the HLLC Riemann solver,[41] with wave speeds from Batten *et al.*,[42] since this is less dissipative than the van Leer flux function.

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.[43] 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.

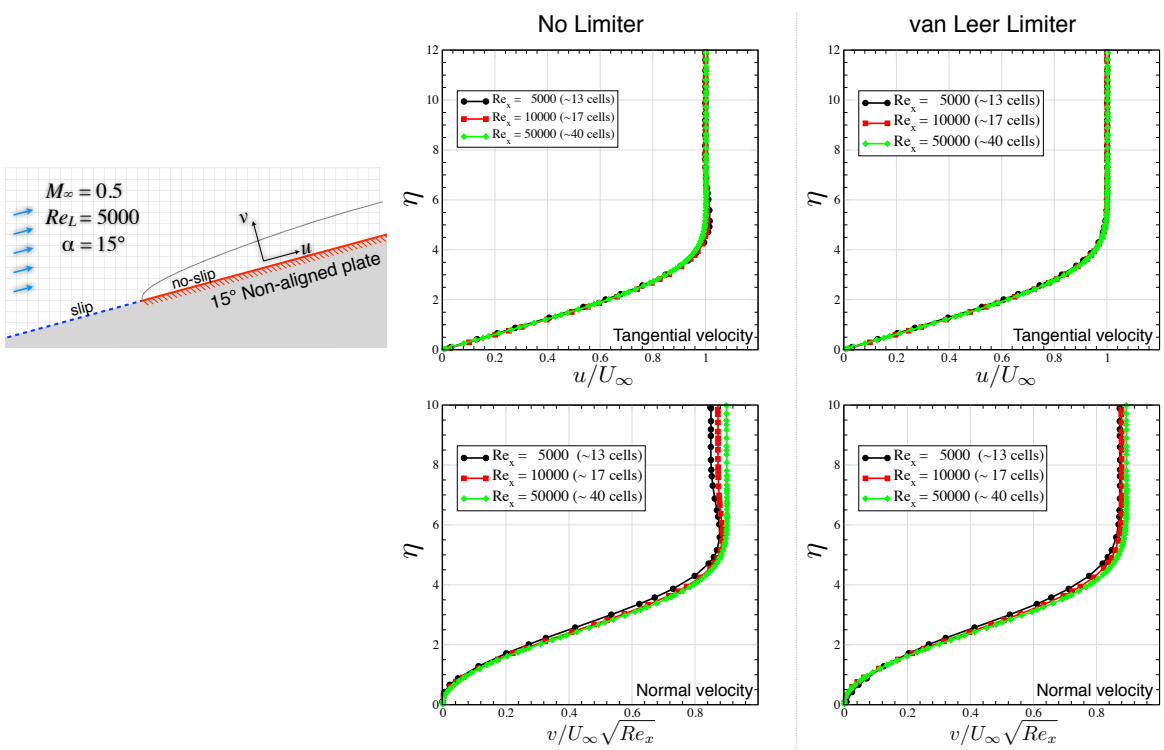


Figure 10. Three profiles at $Re_x = 5K, 10K$ and $50K$ from the flat plate boundary layer for $M_\infty = 0.5$, $Re_L = 5000$. The plate is rotated 15° to the mesh, shown on the left. In the middle figures, the profiles were not limited; on the right the van Leer limiter was used.

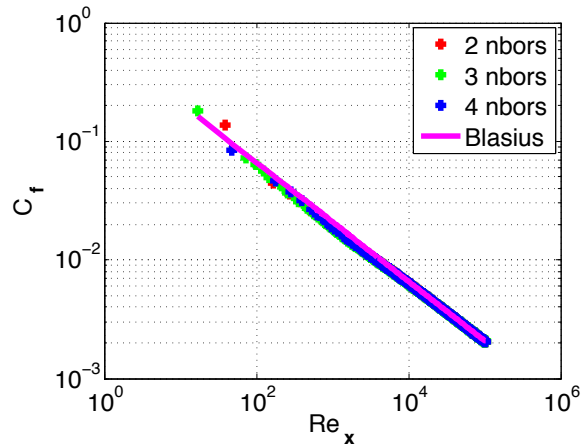


Figure 11. Skin friction for flat plate at 15° to the mesh using the quadratic is well predicted by $Re_x = 100$.

Figure 11 shows the skin friction distribution from the unlimited computation for this case, discussed in detail in Section II. C_f is well predicted by $Re_x = 100$, and compares nicely with coordinate-aligned results.[43] Note the smooth C_f distribution provided by the quadratic wall-normal reconstruction.

B. NACA 0012

The next 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. The results are stable regardless of the direction of the stencil. Figure 12 shows an overview of the discrete solution along with some examples of the quadratic's stencil near the leading and trailing edges. What sometimes appears to be 3 connected points in the stencil are really 2 stencils of 2 points each for adjacent cut cells. Figure 12b shows the velocity magnitude, computed on a grid with leading edge resolution of 0.0016c obtained 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.[43]

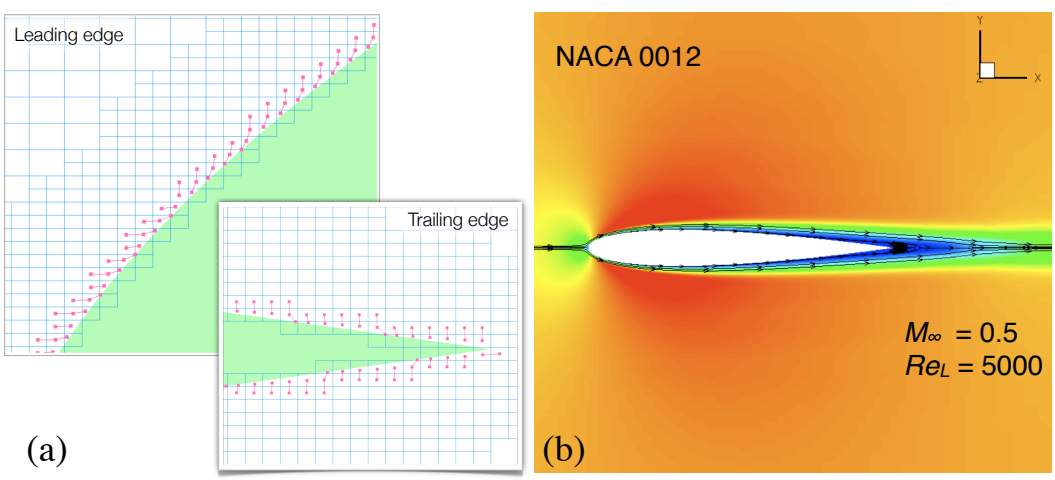


Figure 12. (a) Cells used in quadratic stencil for computing wall fluxes for NACA0012 example. (b) Velocity magnitude.

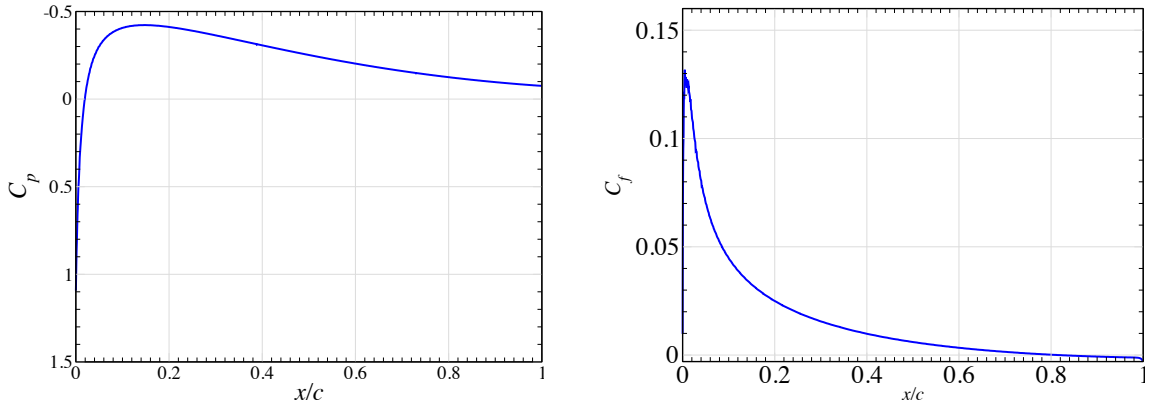


Figure 13. C_p **(left) and** C_f **(right) for the results in Fig. 12b for flow around a NACA 0012.**

C. Turbulent Flat Plate

While turbulent boundary layers are thicker than laminar boundary layers, the higher wall shear stress creates a greater demand for resolution near the wall. In this example we use the RANS equations to simulate a flat plate with free stream Mach number 0.5 and fully turbulent flow at $Re_L = 5 \times 10^5$. We will first compare wall treatments with and without the wall model for a plate that is coordinate-aligned. We then compare the use of the wall model in the aligned case with that of a plate angled at 15° to the mesh. All cases uses the Spalart-Allmaras turbulence model; the cases with the wall model use the SA wall model.

The baseline case of integrating the governing equations down to the wall using 13 levels of refinement is shown in Fig. 14. The tangential profiles (left) are taken from 4 stations along the plate. They are all well-resolved and show no viscous overshoot. We also compare the velocity profile at $x = 10$ corresponding to $Re_x = 5 \times 10^6$ in plus coordinates with the SA wall model (middle). With an initial cell with centroid at $y^+ = 8.8$ the results match well, and are virtually identical to 14 level results (not shown). Skin friction is shown on the right, compared to the asymptotic formula for a flat plate with zero pressure gradient taken from White.[44] The solution is not noisy, since the wall is aligned with the mesh and all cell centroids are the same distance from the wall. After 13 levels of refinement at the wall, the Cartesian cells have a mesh spacing of $h = 3.6 \times 10^{-4}$.

The computational results using the SA wall model are shown in Fig. 15. The figure compares wall model results in both the aligned and non-aligned case. With the wall model, accurate results are achieved on an aligned mesh with only 10 levels of refinement. No attempt has yet been made to refine the mesh in a solution-adaptive manner; it is only refined next to the flat plate. Nevertheless we can report that our

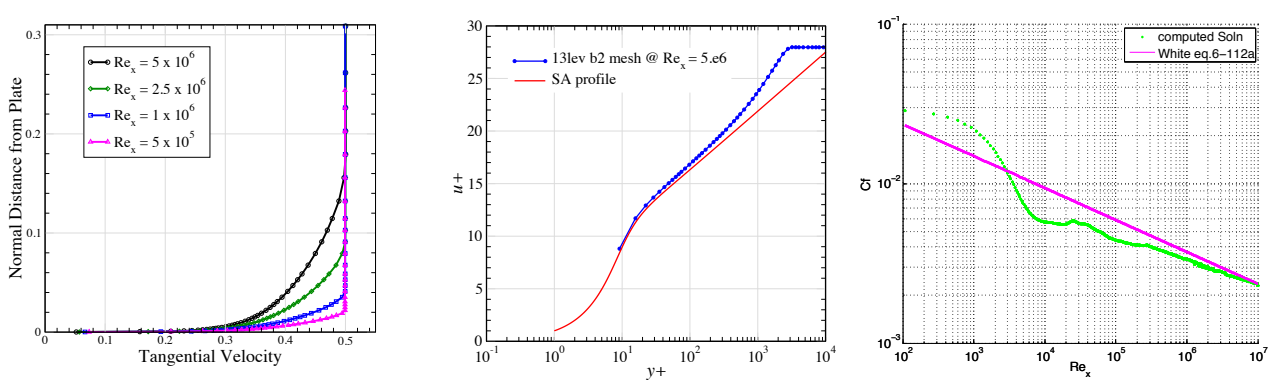


Figure 14. Coordinate-aligned turbulent flat plate with $Re_L = 5 \times 10^5, M_\infty = 0.5$**. Left figure shows well resolved profiles with no viscous overshoot. Calculations were done integrating down to the wall using the quadratic on a 13 level mesh. Middle figure shows the velocity profile at** $Re_x = 5 \times 10^6$ **plotted in wall coordinates and compared to the SA formula for a zero pressure gradient flat plate. For this profile the first cut-cell centroid corresponds to** $y^+ \approx 8.8$**. Right figure shows skin friction compared with asymptotic formula from White.[44]**

non-optimized 13 level mesh had 1.6M cells, and the 10 level mesh had 180K cells.

The computation is much more difficult on the non-aligned plate, since the difference scheme for the cut cells degenerates to first-order. Results on the flat plate oriented at 15° to the mesh needed one finer level of mesh refinement. In addition to the two results with the wall model, Fig. 15 also includes the results of integrating down to the wall. The aligned 13 level result integrating to the wall ($y^+ = 8.8$), the aligned 10 level wall model calculation ($y^+ \approx 70$), and the non-aligned 11 level results ($y^+ \approx 35$) all agree quite well for small y^+. In the wake region the overshoots in the solution clearly show the effect of the larger truncation error when the streamwise direction is not aligned with a coordinate direction. Additional mesh refinement would certainly help in this region of high curvature around the knee of the profile. Alternatively, future research can try to improve the accuracy of the discretization to fourth-order, an especially attractive idea on Cartesian meshes. (Note that the first point in the plot of y^+ versus u^+ is taken from the forcing point F, and so does not necessarily correspond to the finest mesh spacing reported at the wall.)

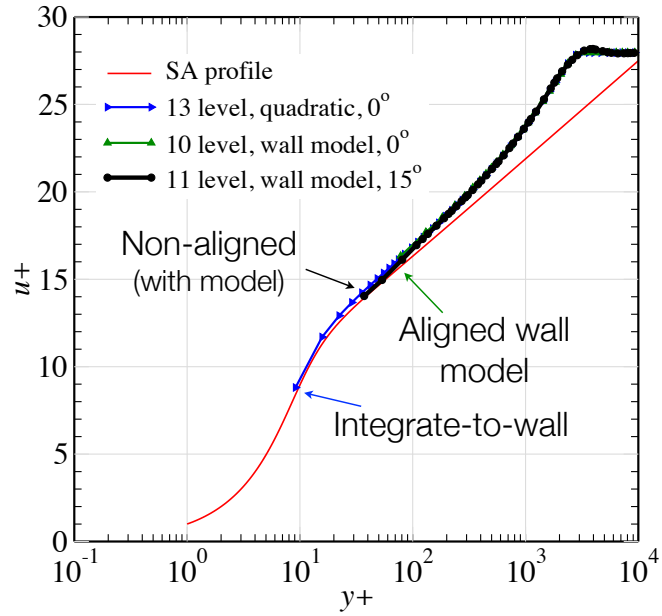


Figure 15. Non-aligned turbulent flat plate using the wall model on an 11 level mesh, compared with the aligned cases, using both the wall model and the quadratic.

D. RAE Airfoil

To assess performance and resolution requirements of the numerical scheme and wall model for high-Reynolds number turbulent flows, we consider three cases using the RAE 2822 airfoil. We present mesh convergence studies on a sequence of three meshes for AGARD Cases 1, 6 and 10.[45] The *coarse* mesh in this sequence has a wall spacing of 0.1% of the airfoil chord, c, and a total of about 50K cells. For these cases, this wall spacing gives values of y^+ over the airfoil largely in the range 200–350. The *medium* mesh has a wall spacing of 0.05%c and a total of ~115K cells with y^+ of 100–170. The *fine* mesh has a wall spacing of 0.025%c and ~150K cells with y^+ values largely between 50 and 80. The medium and fine grids were constructed by subdivision of the near-wall cells of the coarse mesh, thus the three grids are nesting, The domain extends a distance 30c from the airfoil. The simulation had no circulation correction, and all simulations are fully turbulent.

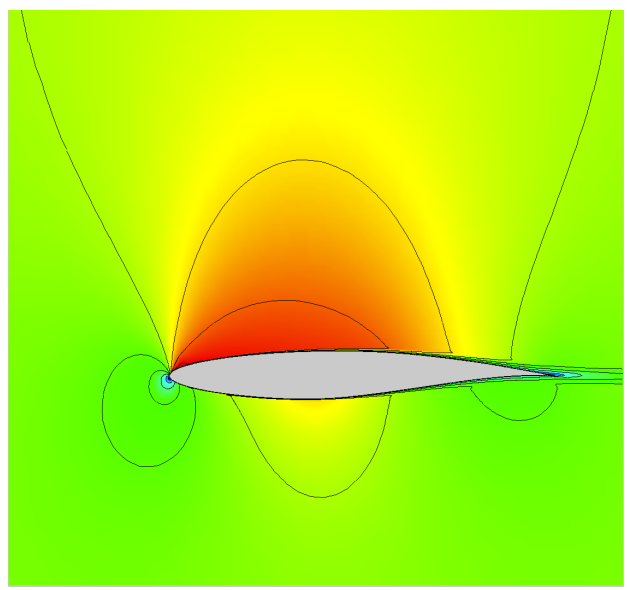


Figure 16. Mach contours of flow around RAE 2822 airfoil at AGARD Case 1 conditions. $M_\infty = 0.676$, $Re_L = 5.7 \times 10^6$ **and** $C_L = 0.566$ ($\alpha = 1.85°$) **using the SA wall model.**

1. *AGARD Case 1*

AGARD Case 1[45] is a subcritical flow at $M_\infty = 0.676$ and a Reynolds number of $Re_c = 5.7 \times 10^6$. In wind tunnel experiments, the target lift coefficient of $C_L = 0.566$ was achieved at an uncorrected $\alpha = 2.4°$. Most simulations however match this value with about half a degree less incidence angle.[46]

Figure 16 illustrates this flow through Mach contours on the fine mesh. The simulation matched the experimental value of lift at $\alpha = 1.85°$. Subsequent simulations on the coarse and medium meshes were run

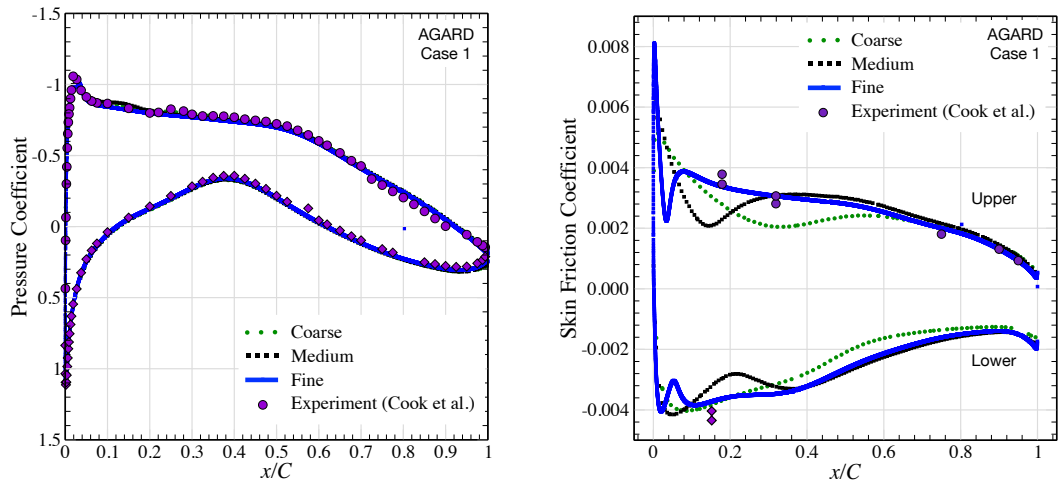


Figure 17. Surface pressure coefficient and skin friction on RAE 2822 airfoil for AGARD Case 1 conditions using the SA wall model on coarse, medium and fine meshes with experimental data from Cook *et al.*[45]

at the same incidence angle. Figure 17 displays the surface pressure coefficient and skin friction distribution for all three meshes, and contains the experimental data for comparison. Agreement between simulation and experiment is on a par with other published solutions.[46] Simulation data from three meshes makes it possible to evaluate the level of mesh convergence. The pressure distribution (Fig. 17 left) nicely captures the suction peak, the rooftop region and the characteristic "duck-tail" in the highly-loaded aft-portion of this airfoil. Overall, the pressures are remarkably invariant with mesh resolution, indicating that these pressures are essentially mesh converged even on the coarsest grid. Since the aft recovery region depends upon prediction of the displacement thickness, it is worth noting that even the coarsest mesh seems to accurately capture this feature despite local y^+ values of approximately 200 on this grid.

Since it is a derivative quantity, prediction of skin friction is more challenging (Fig. 17 right). The experimental drag was measured at 85 counts. The drag values from the simulations are a bit higher, with $C_D = \{0.0099, 0.0097, 0.0099\}$ on the coarse, medium and fine meshes respectively. Despite similarities in the skin friction distributions and the consistency in the integrated drag, there are significant differences between the profiles. At the trailing edge, where the boundary layer is the thickest, local values on the three meshes are in good agreement with both each other and experimental data. Moving upstream from the trailing edge, differences become apparent. As the boundary layer thins, the coarse mesh results start to differ. The fine and medium meshes agree over most of the surface, however upstream of about 30%c, even the medium mesh appears to be insufficient as it starts to differ from both the fine mesh solution and the experimental data. Near the leading edge, only the fine mesh remains credible and shows good agreement with the data upstream of 20% chord. As in the experimental dataset, skin friction data in this figure was normalized using the dynamic pressure at the outer-edge of the boundary layer to facilitate direct comparison.

2. AGARD Cases 6 & 10

The final two examples consider supercritical flow over the same RAE 2822 airfoil section. They are characterized by increasingly strong shock-boundary layer interactions. AGARD Case 6 was tested at $M_\infty = 0.725$, $Re_c = 6.5 \times 10^6$ and produced a measured lift coefficient of $C_L = 0.743$ with an uncorrected incidence angle of $\alpha = 2.92°$. Case 10 was tested at the same lift coefficient, but under somewhat more aggressive conditions: $M_\infty = 0.750$ and $Re_c = 6.2 \times 10^6$. $C_L = 0.743$ was measured at an uncorrected incidence of $\alpha = 3.19°$.[45] Over the past 30 years, these two cases have been the subject of countless numerical experiments. As in Case 1, most numerical simulations achieve the experimental values of lift with around half a degree less than the experimentally reported incidence angle.[46]

Figure 18 illustrates these two flows through isoclines of Mach number with case 6 on the left and case 10 on the right. The fine mesh simulation data of Case 6 matched the experimental lift coefficient of $C_L = 0.743$ at $\alpha = 2.44°$ while the Case 10 data matched this same value at $\alpha = 2.54°$. As before, simulations on all meshes were performed using these values. Despite identical lift coefficients, the higher Mach number in the Case 10 flow produces a stronger shock which is shifted noticeably aft.

Figure 19 contains a comparison of surface pressure distributions for these two transonic cases. Agreement

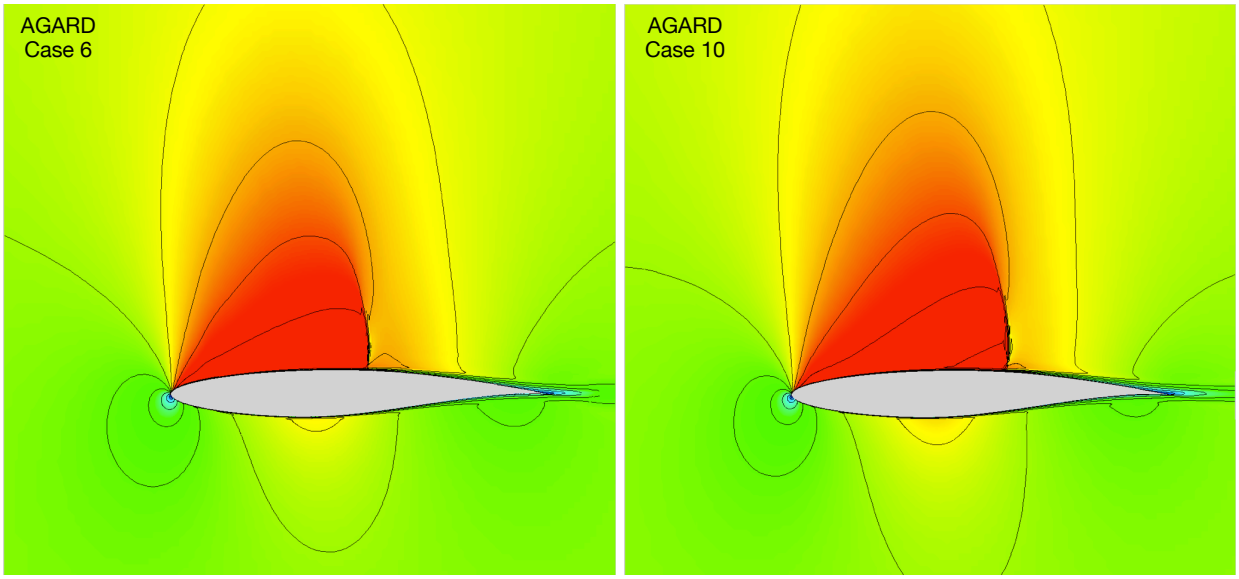


Figure 18. Mach contours of flow around RAE 2822 airfoil at AGARD Case 6 (left) and Case 10 (right) conditions using the SA wall model on the fine mesh. Case 6: $M_\infty = 0.725$**,** $Re_c = 6.5 \times 10^6$ **@** $C_L = 0.743$ **(**$\alpha = 2.44°$**). Case 10:** $M_\infty = 0.750$**,** $Re_c = 6.2 \times 10^6$ **@** $C_L = 0.743$ **(**$\alpha = 2.54°$**).**

with the experimental data is comparable to others in the literature.[30,46,47] In Case 6, the computed pressures on the lee side do not dip as much following the suction peak as the experimental data, and consequently the shock is slightly forward to yield the same lift coefficient. In Case 10, the numerical solution places the shock slightly farther aft than in the experiment, and shows small discrepancies in the aft loading. Similar differences have been noted by numerous authors.[30,46,47] As in the earlier subcritical example, both of these cases show little variation of the pressure distribution as the mesh is refined, and even the coarse mesh with a wall spacing of 0.1%c and only 50K cells seems sufficient to produce mesh-converged pressures. The shock position in both of these examples is critically dependent on the displacement effect of the boundary layer and the shock-boundary layer interaction, and the performance of the coarse mesh is noteworthy in this regard.

Figure 20 displays the the evolution of C_f for both Case 6 and Case 10 as the mesh is refined. The profiles display similar behavior to that discussed in the earlier subcritical example (Fig. 17). Integrated drag on the fine mesh was 133 counts which compares well with the experimental value of 127 counts.[45] As before, agreement is strongest where the boundary layer is the thickest. Tracing upstream from the trailing edge

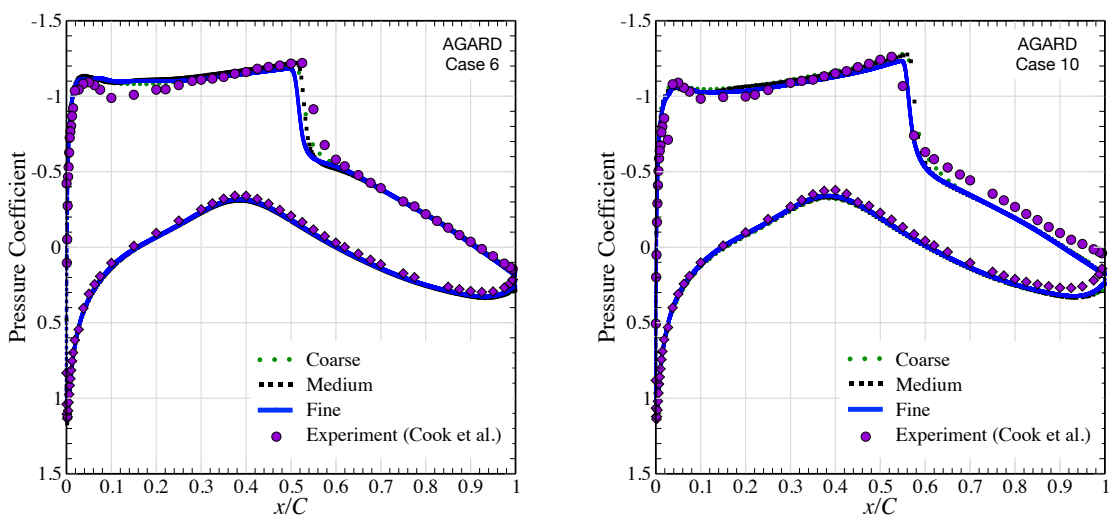


Figure 19. Surface pressure coefficients on RAE 2822 airfoil at AGARD Case 6 (left) and Case 10 (right) conditions using the SA wall model on coarse, medium and fine meshes with comparison to experimental data from Cook ***et al.*****[45].**

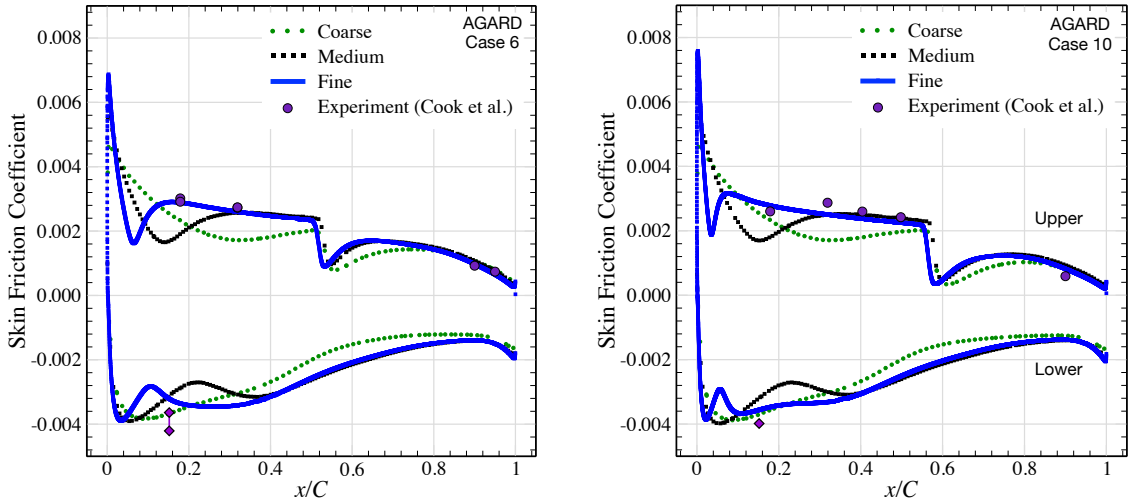


Figure 20. Skin friction distributions on RAE 2822 airfoil at AGARD Case 6 (left) and Case 10 (right) conditions using the SA wall model on coarse, medium and fine meshes with comparison to experimental data from Cook *et al.*[45].

reveals that the coarse mesh starts to differ at about 80%c. The medium mesh shows discrepancy upstream of 30%c, and only the finest mesh accurately predicts the experimental data near the leading edge where the boundary layer is the thinnest. This case is characterized by the strong shock-boundary layer interaction on the suction side. While results in the literature are mixed, many authors report a small recirculation bubble following this shock.[46] The present simulations are showing very small but positive values of skin friction, indicating only incipient separation. Detailed investigation of the analytic wall model's behavior near the shock-boundary layer interaction remains an important topic of investigation.

V. Conclusions

In this paper we outlined initial development of a method for solution of the Reynolds Averaged Navier-Stokes equations on embedded-boundary Cartesian meshes using a wall model in the cut cells. First we developed accurate discretizations based on recentering for use in the highly irregular cut cells. We also introduced an analytic wall model for turbulent flow simulations based on a limiting solution of the Spalart-Allmaras turbulence model which provides accurate representation of attached flows in the laminar sub-layer, buffer and log-layer regions of the boundary layer. This model has distinct advantages over standard Spalding-based wall functions in that it automatically matches the turbulence model used away from the wall. Additionally, it entails an explicit evaluation for the velocities as a function of distance and does not require an iterative solution. We use an enhanced multigrid algorithm that includes geometry and gradients on coarser grids to achieve multigrid convergence nearly as good as in the inviscid case. Finally, we showed that the resulting scheme can provide accurate pressure distributions for high Reynolds number flow over subsonic and transonic airfoils with wall spacings of about 0.1%c, even when the pressure distribution is strongly dependent on the displacement of the boundary layer. Drag values for attached flows are reasonable at this same resolution, but detailed skin friction distributions require at least one or two additional cell refinements.

Resolution requirements of the current method are driven by the assumptions underlying the analytic wall model. The new SA wall model, like a Spalding-based wall function, is still based on a simple diffusion model of the near-wall flow and so it cannot accurately predict values outside the log layer. Both the non-aligned turbulent flat plate and 2D airfoil examples showed that accurate local skin friction values required resolution to the log layer. This is currently the limiting factor in determining the lowest feasible Cartesian mesh resolution. This provides strong motivation for work towards a more comprehensive PDE-based wall model which retains more physics from the governing equations.

The present results are an important first step, but clearly a long list of questions remain. Our immediate goal is development of a more complete, non-analytic wall model which includes streamwise momentum and pressure gradient terms that become important outside the log layer and in separated flows. Robust

convergence and coupling of such a model with the underlying Cartesian method is likely to be another challenge. A successful three-dimensional implementation will clearly depend on the wall model to mitigate the resolution requirements that would lead to untenable cell counts. Nevertheless, the simplicity and automation of Cartesian mesh generation continue to provide persuasive motivation for further algorithmic development.

Appendix A: Numerical Discretizations of Second Derivatives at Cut Cells

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. We consider three possible discretizations for an elliptic equation to evaluate their accuracy on a problem with an analytic solution, chosen because they are nominally second-order, conservative and can be implemented in a finite volume framework. These tests do not rely on special properties of elliptic equations.

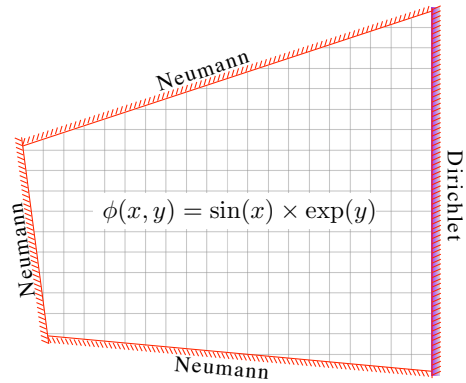


Figure 21. Domain for elliptic model problem in Table 1.

Consider the model Poisson equation in two dimensions

$$\nabla^2 \phi = f \tag{20}$$

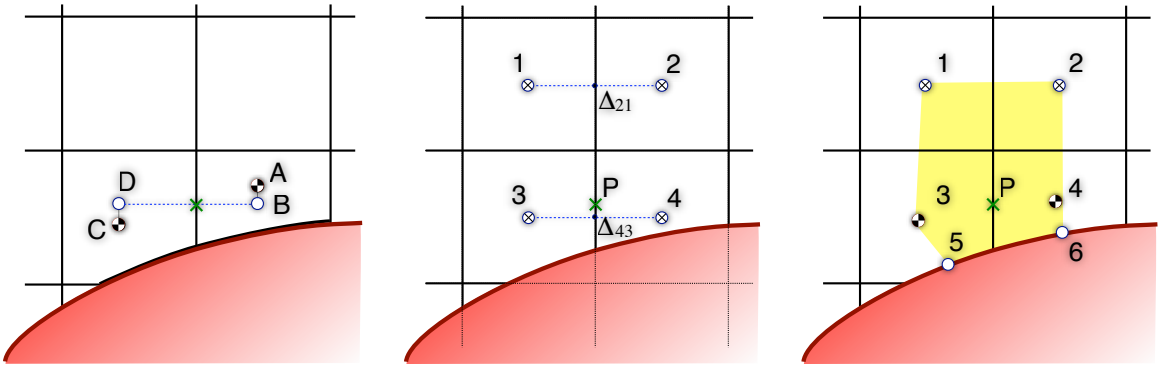
with exact solution $\phi(x, y) = \sin(x)\exp(y)$. The domain is a quadrilateral with three non-coordinate-aligned sides, as shown in Fig. 21. Three of the four sides use Neumann boundary conditions, with Dirichlet on the fourth.This is in contrast to the Euler equations, where only the value of the solution itself is needed at the cell edge. At interior cells, the standard central difference approximation for the first derivative, resulting in the second-order accurate 5-point Laplacian is used. Three alternatives are tested for the cut cells.

Recentering: The first method uses a "recentering" idea[48,49] illustrated in Fig. 22a. Recentering operates on the integral cell average, which for a second-order finite volume scheme can also be used for the pointwise value of the state vector at the cell centroid. Within each cut cell, a least squares gradient is reconstructed using the solution from each cell's stencil of face neighbors. The gradient is used to reconstruct the solution from the cut-cell centroid to a line normal to the face through the face centroid (see Fig. 22a). 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. Using the notation of Fig. 22a, with cell centroids at A and C, we recenter to locations B and D, defining

$$\begin{aligned} \phi_x &= \frac{\phi_B - \phi_D}{x_B - x_D} \\ &= \frac{(\phi_A + \nabla\phi_A \cdot d_{BA}) - (\phi_C + \nabla\phi_C \cdot d_{DC})}{x_B - x_D} \end{aligned} \tag{21}$$

where d_{BA} is the vector from A to B, and similarly for d_{DC}. Note that this divided difference is nominally first order accurate, since it is not centered about the face centroid. Changing the recentering step to use locations that are equally spaced about the face centroid, so that a centered difference approximation can be used, is no more accurate than the one-sided scheme without using a more accurate reconstruction, so we do not include those results here.

Johansen-Colella: The second discretization uses the Johansen-Colella framework.[50] In this approach the solution ϕ 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 at the midpoint of the uncut face. The flux however is needed at the cut-face centroid, illustrated in Fig. 22b at the point P. This is easily obtained by linear interpolation along the edge.



(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 $\triangle_{43}$) .

(c) **Polynomial Fit:** A least squares polynomial that is linear in x and quadratic in y is fit using 6 neighboring values. The polynomial is differentiated and evaluated at the point P to approximate the flux.

Figure 22. Three methods for computing the derivative at a cut-cell edge.

Polynomial Fit: The third discretization is similar to the method described in Ye et al.[51] 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 centroidal values. As illustrated in Fig. 22c, we use the 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 and evaluated at the cut-face centroid to obtain the flux.

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 ($\partial\phi/\partial n$) is needed. For the sides with Neumann boundary conditions this becomes an evaluation of the boundary condition. On the Dirichlet side we use a one-sided discretization, taking the cell average, subtracting the Dirichlet boundary condition evaluated at the point normal to the centroid (or cell center, in the case of Johansen-Colella), and dividing by the distance between them.

	Recentering		Johanson-Colella		Polynomial fit	
Size	1 norm	max norm	1 norm	max norm	1 norm	maxnorm
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
conv. rate	3.8	3.7	3.4	2.9	4.4	3.3
36×36	2.60	.11	1.81	.12	2.64	.60
conv.rate	3.9	3.6	4.0	4.0	4.0	2.4
72×72	.66	.03	.45	.03	.65	.13
conv.rate	3.9	3.7	4.02	4.0	4.0	4.6

Table 1. Results of solving the Poisson equation $\nabla^2\phi = f$ on an irregular domain using 3 different discretizations for the cut cells. Data corresponds to Fig. 23.

Figure 23 and Table 1 report the error in computing the solution to (20). Results include both the maximum norm of the error, $||e||_\infty = \max_{ij} \ |\phi_{exact_{ij}} - \phi_{computed_{ij}}|$, and the L_1 norm of the error, $||e||_1 = \sum_{ij} \ A_{ij}|\phi_{exact_{ij}} - \phi_{computed_{ij}}|$, where the A_{ij} are the cell areas. (Note that the L_1 error is not normalized by the domain size, so it is larger than the maximum norm errors.) All methods show second-order convergence in the L_1 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 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. These results come with a caveat, and need to be interpreted carefully. There are terms in the Navier-Stokes equations that are not included in this model problem. In practical settings, these remaining terms can dominate the error, and lead to non-convergence if not handled carefully. In addition, we are not looking at positivity of the stencil, which has been the focus of several other studies.[1,5] Although positivity ensures a maximum principle, blindly insisting on it may rule out potentially useful discretizations.[3]

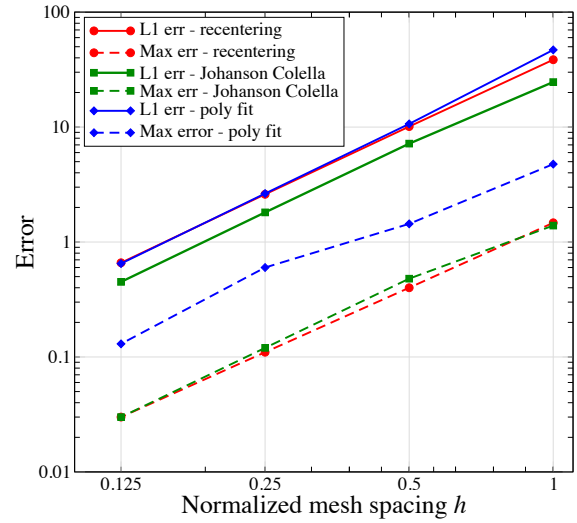


Figure 23. Convergence of 3 methods for computing cut-cell fluxes for the model Poisson problem $\nabla^2\phi = f$, corresponding to data in Table 1.

Appendix B: Stability of Quadratic Boundary Condition

This analysis examines 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[29] 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[52] of the polynomial, the quadratic interpolant of the three values u_b, u_1 and u_2 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 $x = 0$ even though it is zero. 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 the update of u_1 is

$$u_1^{n+1} = u_1^n + \frac{\Delta t}{h}((u_2 - u_1)/h - p_2') \tag{22}$$

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

$$z = 1 + \lambda(\kappa - 2 + 1/\kappa) \tag{23}$$

where we are interested in solutions with $\kappa \le 1$. The characteristic equation for the boundary condition (22) is

$$z = 1 + \frac{\lambda}{3}(4\kappa - 12). \tag{24}$$

Equating (23) and (24) gives an l_2 solution $\kappa = 3 - 2\sqrt{3}$.

Substituting this root into (24) and solving for $z \le 1$ 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

[3]For example, the 4th-order Laplacian derived using Richardson extrapolation by combining the 3-point h-sized stencil and the $2h$ stencil has negative coefficients on the points $2h$ away from the center yet is symmetric positive definite.

equations only reduce the allowable time step by a small fraction, the impact on the time step in practice would be correspondingly less.

Appendix C: Analytic Law of the Wall Solution for the Spalart-Allmaras Turbulence Model

by Steven R. Allmaras

We make the usual assumptions for law of the wall analysis: incompressible, zero pressure gradient, constant outer (edge) velocity, advection terms are negligible and $\partial/\partial x \ll \partial/\partial y$, where x is streamwise and y normal to the wall. With these assumptions, $u = u(y)$ and $\tilde{\nu} = \tilde{\nu}(y)$, and the x-momentum equation reduces to a statement of constant total shear stress,

$$\frac{d}{dy}\left[(\nu+\nu_t)\frac{du}{dy}\right] = 0, \quad \rightarrow \quad (\nu+\nu_t)\frac{du}{dy} = \text{const} = u_\tau^2, \tag{25}$$

where u_τ is the wall shear stress velocity. With these same assumptions, Spalart-Allmaras (SA) reduces to,

$$\frac{1}{\sigma}\frac{d}{dy}\left[(\nu+\tilde{\nu})\frac{d\tilde{\nu}}{dy}\right] + \frac{c_{b2}}{\sigma}\left[\frac{d\tilde{\nu}}{dy}\right]^2 + c_{b1}\left(1-f_{t2}\right)\widetilde{S}\tilde{\nu} - \left(c_{w1}f_w - \frac{c_{b1}}{\kappa^2}f_{t2}\right)\left[\frac{\tilde{\nu}}{y}\right]^2 = 0, \tag{26}$$

where

$$\nu_t = \tilde{\nu}\, f_{v1}, \qquad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \qquad \chi \equiv \frac{\tilde{\nu}}{\nu}, \tag{27}$$

and

$$\widetilde{S} = \frac{du}{dy} + \frac{\tilde{\nu}}{\kappa^2 y^2} f_{v2}, \qquad f_{v2} = 1 - \frac{\chi}{1+\chi f_{v1}}, \qquad f_{t2} = c_{t3}\exp\left(-c_{t4}\chi^2\right). \tag{28}$$

By construction, these equations have the simple solution,

$$\tilde{\nu} = \kappa u_\tau y, \qquad \widetilde{S} = \frac{u_\tau}{\kappa y} \tag{29}$$

Transforming to wall units, $y^+ \equiv y u_\tau/\nu$, $u^+ \equiv u/u_\tau$, this solution becomes,

$$\chi = \kappa y^+, \qquad \widetilde{S}^+ = \frac{\nu}{u_\tau^2}\widetilde{S} = \frac{1}{\kappa y^+}. \tag{30}$$

This solution is the extension of well known log-law behavior to the entire inner layer from the wall through the viscous sublayer and into the log-law region. In the log layer, Reynolds' stresses dominate molecular stresses. With the introduction of the Boussinesq approximation and the log-law velocity distribution, both velocity gradient and eddy viscosity can be determined,

$$\frac{\nu_t}{\nu} = \kappa y^+, \qquad \frac{du^+}{dy^+} = \frac{1}{\kappa y^+}, \qquad y^+ \gg 1. \tag{31}$$

In developing the near-wall or viscous sublayer components of SA, this simple behavior was retained for the new solution variable $\tilde{\nu}$ and the modified vorticity $\widetilde{S}$ by introducing the eddy viscosity correlation function f_{v1}, the definition of modified vorticity (via f_{v2}) and the re-definition of r (not shown). Note that the presence of the laminar suppression term, f_{t2}, in SA is passive with respect to the simple solution; contributions from production and wall destruction cancel in (26).

Substituting the simple solution (30) into either x-momentum or the definition for $\widetilde{S}$ then gives,

$$\frac{du^+}{dy^+} = \frac{c_{v1}^3 + (\kappa y^+)^3}{c_{v1}^3 + (\kappa y^+)^3(1+\kappa y^+)} \tag{32}$$

This equation can be integrated (via Mathematica[53]) and the constant of integration determined from the no-slip boundary condition $u(0) = 0$. Using complex arithmetic, the solution is,

$$u^+(y^+) = \sum_{i=1}^{4} \frac{c_{v1}^3 + z_i^3}{\kappa z_i^2 (3 + 4z_i)} \left[\log\left(\kappa y^+ - z_i\right) - \log\left(-z_i\right)\right], \tag{33}$$

where z_i are the four solutions to the quartic equation,

$$c_{v1}^3 + z_i^3 + z_i^4 = 0 \tag{34}$$

The solution can be simplified and rewritten using real arithmetic,

$$u^+(y^+) = \overline{B} + c_1 \log\left((y^+ + a_1)^2 + b_1^2\right) - c_2 \log\left((y^+ + a_2)^2 + b_2^2\right) \\ - c_3 \mathrm{ArcTan}[y^+ + a_1,\, b_1] - c_4 \mathrm{ArcTan}[y^+ + a_2,\, b_2], \tag{35}$$

where $\mathrm{ArcTan}[x,\, y]$ is the Mathematica function equivalent to the Fortran function $\mathrm{atan2}(y,\, x)$. For the values of $\kappa = 0.41$ and $c_{v1} = 7.1$, the constants are given by,

$$\begin{aligned} &\overline{B} = 5.0333908790505579 \\ &a_1 = 8.148221580024245 && b_1 = 7.4600876082527945 \\ &a_2 = -6.9287093849022945 && b_2 = 7.468145790401841 \\ &c_1 = 2.5496773539754747 && c_2 = 1.3301651588535228 \\ &c_3 = 3.599459109332379 && c_4 = 3.6397531868684494 \end{aligned}$$

Analysis of this solution for large y^+ reveals that SA asymptotically produces a log-law with a shifted origin compared to the conventional formulas,

$$u^+ \sim \frac{1}{\kappa} \log\left(y^+ + 1/\kappa\right) + \overline{B}, \qquad \frac{du^+}{dy^+} \sim \frac{1}{\kappa y^+ + 1}, \quad \text{as} \quad y^+ \to \infty. \tag{36}$$

The shift in the asymptotic gradient can also be derived from (32). The origin shift is minor, but is easily noticeable in law of the wall velocity plots.

Figure 24 shows the law of the wall velocity profile for SA (35) compared to its asymptotic form (36) and to Spalding's composite formula,

$$y^+ = u^+ + \exp(-\kappa B)\left[\exp(\kappa u^+) - 1 - \kappa u^+ - \frac{1}{2}(\kappa u^+)^2 - \frac{1}{6}(\kappa u^+)^3\right]. \tag{37}$$

Here the values of $\kappa = 0.41$ and $B = 5$ have been plotted; this value of B is consistent with the calibration of SA (and in particular $c_{v1} = 7.1$). SA differs from Spalding by about 3.4% at $y^+ = 50$.

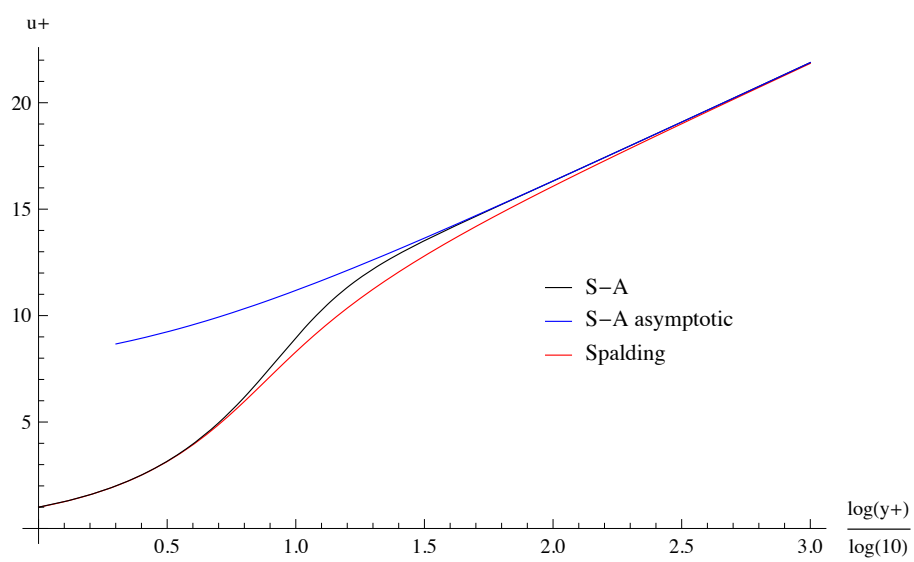


Figure 24. Law of the Wall velocity profile

Acknowledgments

Marian Nemec has been very helpful in advising us on this work throughout. We thank Tom Pulliam for providing ARC2D results along with fruitful and thought-provoking discussions. Mike Olsen was instrumental in getting started on this research. M.J. Berger was supported in part by DOE grant DE-FG02-88ER25053 and by AFOSR grants FA9550-06-1-0203 and FA9550-10-1-0158. Steve Allmaras's research was conducted while the author was under contract with the NASA Ames Educational Associates Program.

References

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

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

[3]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*, 2010.

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

[5]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.

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

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

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

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

[10]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.

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

[12]Kupiainen, M. and Sjogreen, B., "A Cartesian Embedded Boundary Method for the Compressible Navier-Stokes Equations," *J. Scientific Comp.*, Vol. 41, No. 1, 2009, pp. 94–117.

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

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

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

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

[17]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.

[18]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.

[19]Craft, T., Grant, S. E., Iacovides, H., and Launder, B. E., "A New Wall Function Strategy for Complex Turbulent Flows," *Numerical Heat Transfer, Part B*, 2004, pp. 301–318.

[20]Myers, S. H. and Walters, D. K., "A One-dimensional Subgrid Near Wall Treatment for Turbulent Flow CFD Simulation," *Proceedings of IMECE 2005*, Paper IMCE 2005-81712, November 2005.

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

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

[23]Bond, R. B. and Blottner, F. G., "A Compressible Wall-Layer Approach Compatible with Various Turbulence Models," *AIAA-2007-1410*, 2007.

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

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

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

[27]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.

[28]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.

[29]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.

[30]Spalart, P. and Allmaras, S., "A one equation turbulence model for aerodynamic flows," *AIAA-92-0439*, 1992.

[31]Oliver, T. A., *A High-Order, Adaptive, Discontinuous Galerkin Finite Element Method for the Reynolds-Averaged Navier-Stokes Equations*, Ph.D. thesis, Massachusetts Institute of Technology, 2008.

[32]Allmaras, S., Private communication.

[33]Sethian, J. A., "A fast marching level set method for monotonically advancing fronts," *Proc. Natl. Acad. Sci. USA*, Vol. 93, 1996, pp. 1591–1595.

[34]Wilcox, D. C., *Turbulence Modeling for CFD*, DCW Industries, Inc., La Cañada, CA, 91011, 3rd ed., 2006.

[35]Bond, R. B. and Blottner, F. G., "Derivation, implementation, and initial testing of a compressible wall-layer model," *Intl. J. Num. Methods Fluids*, Vol. 66, 2011, pp. 1183–1206.

[36]Frink, N. T., "Tetrahedral Unstructured Navier-Stokes Method for Turbulent Flows," *AIAA Journal*, Vol. 36, No. 11, 1998.

[37]Sondak, D. and Pletcher, R., "Application of Wall Functions to Generalized Nonorthogonal Curvilinear Coordinate Systems," *AIAA Journal*, Vol. 33, No. 1, 1995.

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

[39]Pierce, N. A., *Preconditioned Multigrid Methods for Compressible Flow Calculations on Stretched Meshes*, Ph.D, University of Oxford, August 1997.

[40]Hackbusch, W., *Multi-Grid Methods and Applications*, Springer-Verlag, 1985.

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

[42]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.

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

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

[45]Cook, P. H., MacDonald, M. A., , and Firmin, M. C. P., "Aerofoil RAE 2822 - Pressure Distributions, and Boundary-Layer and Wake Measurements," *AGARD Advisory Report No. 138: Experimental Data Base for Computer Program Assessment*, North Atlantic Treaty Organization Advisory Group for Aerospace Research and Development, May 1979.

[46]Holst, T. L., "Viscous Transonic Airfoil Workshop Compendium of Results," *AIAA Paper 1987-1460*, June 1987.

[47]Capizzano, F., "Turbulent Wall Model for Immersed Boundary Methods," *AIAA Journal*, Vol. 49, No. 11, 2011.

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

[49]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.

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

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

[52]"Newton Polynomial," *http://en.wikipedia.org/Newton_polynomial*, 2010.

[53]Wolfram Research, I., Mathematica Edition: version 7.0, 2008.