# High Reynolds Number Flow Simulations on Cartesian Cut-cell Grids

Alex Kleb[*,**], Krzysztof J. Fidkowski[*] and Joaquim R. R. A. Martins[*]
Corresponding author: akleb@umich.edu
[*] University of Michigan, USA.
[**] Analytical Mechanics Associates, USA.

**Abstract:** Aerodynamic design workflows are often bottlenecked by the user-intensive process of generating accurate computational meshes. Cartesian cut-cell methods circumvent this issue by automating the mesh generation procedure. However, Cartesian cut-cell methods still struggle to provide affordable solutions to the Reynolds-Averaged Navier–Stokes equations for high Reynolds number flows due to irregular stencils near the wall and a lack of anisotropic mesh stretching. In this work, we show that an ordinary differential equation (ODE) wall model incorporating pressure-momentum balance enables accurate high Reynolds number viscous flow predictions on coarse Cartesian cut-cell meshes. The ODE captures the transition from the viscous dominated near-wall region to the inviscid wake region, allowing forcing points to operate effectively at $y^+ > 600$. Unlike traditional wall functions, the ODE is not limited to the logarithmic layer and maintains accuracy in strong pressure gradient environments typical of aerodynamic applications. We demonstrate the effectiveness of the ODE wall model on a series of increasingly challenging test cases from the NASA turbulence modeling resource.

*Keywords:* Computational Fluid Dynamics, Turbulence Modeling, Cartesian, Cut Cells, ODE Wall Model.

## 1. Introduction

The Reynolds-Averaged Navier–Stokes (RANS) equations are commonplace in aircraft analysis and design. Compared to other computational fluid dynamics (CFD) models, RANS provides a reasonable accuracy to computational cost ratio at aircraft cruise conditions. Development of the RANS adjoint has encouraged its extensive use as the primary analysis tool in aerodynamic shape optimization [1, 2, 3].

One of the most important and time-consuming parts of a CFD workflow is the generation of the computational mesh. Cartesian cut-cell methods circumvent this problem by automatically generating meshes without user intervention. They allow for fully automated mesh generation for arbitrarily complex geometries [4]. Hexahedral background meshes are made up of right parallelepipeds aligned with the principal axes. A wetted surface geometry is cut out of a background mesh, and then adaptive mesh refinement is used to reduce discretization errors to adequate levels. This process generates a mesh that is not body-fitted.

Traditionally, cut-cell methods have struggled with high Reynolds number flows for two reasons. The first issue is the irregular cell sizes near walls due to cut cells. The irregular gradient stencils near the wall can lead to noise in gradient reconstruction and flux evaluations in the boundary layer where gradients change rapidly. This is a common problem for unstructured meshes [5]. However, this issue is amplified for cut cells where sliver cut cells are commonly orders of magnitude smaller than their neighbors.

The second issue is that the number of isotropic cells required for wall-resolved boundary layers becomes intractable for high Reynolds number flows. Wall-transverse resolution is only needed in specific areas such as boundary layer startup and shocks. Wall-normal resolution is needed throughout the domain. Boundary conforming meshes address these requirements by stretching cells along the wall, only placing extra points in the wall-normal direction. Various techniques such as separate near-body meshes [6, 7] and line-dropping [8, 9] attempted to resolve this issue for cut-cell meshes. However, they are ultimately no longer automatic meshing techniques. To retain the automatic and robust nature of the cut-cell mesh generation, extra burden must be passed onto the physical model to resolve these issues.

There are two practical avenues for solving high Reynolds number turbulent flows on Cartesian cut-cell meshes. The first approach is to change the behavior of the flow solution. The primary issue within the boundary layer is the nonlinear behavior of the wall-tangential velocity near the wall. If the wall-tangential velocity had a benign profile, then larger Cartesian cut-cells could be used without sacrificing accuracy. Tamaki et al. [10] developed a special form of the RANS equations with the Spalart–Allmaras turbulence model that has a linear velocity profile near the wall with a slip boundary condition. To ensure the solutions remain accurate, the turbulence model is adjusted to maintain the proper shear stress balance near the wall. Ursachi et al. [11] successfully performed a similar technique with an unstructured discontinuous Galerkin finite element scheme. They applied modifications to the RANS-SA equations to match the integrated shear-stress, with linear velocity profiles and a slip boundary condition.

The second method for alleviating Cartesian cut-cell meshing requirements is the interior forcing point approach [12, 13]. Background flow states are interpolated to interior forcing points and fit to functions that provide approximations of the boundary layer flow near the wall. They are coupled back into the background mesh to inform flux evaluations. By relying on these approximations to resolve the boundary layer, cell size requirements are relaxed, making the simulation of high Reynolds number flows feasible for isotropic cells. This approach does not detract from the automatic or robust grid generation process inherent to Cartesian cut-cell methods. Each cut cell can be assigned a unique interior forcing point defined using the local embedded face normal.

In a previous paper, we demonstrated an interior forcing point approach that relaxed near-wall meshing requirements [14]. That method produced noisy skin friction from the irregular gradient stencils in cut cells, and its ODE wall models stalled global convergence when their stiff source terms failed to solve. This paper resolves both. First, a forcing-point gradient interpolation that favors lower-order averaging over exact reconstruction suppresses the cut-cell noise corrupting viscous fluxes. Second, a continuation strategy retains a sufficient boundary layer approximation with partial convergence of the stiff ODE wall models, enabling global convergence of the RANS system. Together, on two-dimensional NASA turbulence modeling resource cases, these let an automatic Cartesian cut-cell solver match body-fitted FUN3D lift and drag convergence rates at $y^+ > 1000$ at forcing points.

## 2. Governing Equations

In this work, we focus on the compressible and steady form of the Reynolds-Averaged Navier–Stokes (RANS) equations known as the Favre-Averaged Navier–Stokes equations [15]. This distinction is usually ignored in the literature, and the term RANS equations will refer to the Favre-averaged Navier–Stokes in this work. The RANS equations are written,

$$\partial_j(\rho v_j) = 0, \tag{1}$$

$$\partial_j(\rho v_j v_i + \delta_{ij} p) - \partial_j \tau_{ij}^{\mathrm{RANS}} = 0, \tag{2}$$

$$\partial_j(\rho H v_j) - \partial_j(v_i \tau_{ij} - Q_i^{\mathrm{RANS}}) = 0, \tag{3}$$

where $\rho$ is the fluid density, $\vec{v}$ is the fluid velocity, $\delta_{ij}$ is the Kronecker delta function, $p$ is the pressure, $H$ is the specific total enthalpy, $\tau^{\mathrm{RANS}}$ is the effective viscous stress tensor accounting for Reynolds stresses, and $\vec{Q}^{\mathrm{RANS}}$ is the effective heat flux accounting for turbulent transport of energy. The state variables in the RANS equation are the mean flow quantities. The effective viscous stresses and heat flux can be written in terms of the mean flow quantities. In two dimensions, and using Stokes's hypothesis for the bulk viscosity,

$$\tau_{xx}^{\text{RANS}} = 2(\mu + \mu_t)\left[\frac{\partial v_x}{\partial x} - \frac{1}{3}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right)\right], \tag{4}$$

$$\tau_{yy}^{\text{RANS}} = 2(\mu + \mu_t)\left[\frac{\partial v_y}{\partial y} - \frac{1}{3}\left(\frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y}\right)\right], \tag{5}$$

$$\tau_{xy}^{\text{RANS}} = \tau_{yx}^{\text{RANS}} = (\mu + \mu_t)\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right), \tag{6}$$

$$Q_x^{\text{RANS}} = -\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\gamma}{\gamma - 1}\frac{\partial}{\partial x}\left(\frac{p}{\rho}\right), \tag{7}$$

$$Q_y^{\text{RANS}} = -\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\frac{\gamma}{\gamma - 1}\frac{\partial}{\partial y}\left(\frac{p}{\rho}\right), \tag{8}$$

where $\mu_t$ is the eddy viscosity that is computed by a turbulence model, $Pr$ is the Prandtl number and $Pr_t$ is the turbulent Prandtl number.

To close the system, we use the one-equation Spalart–Allmaras (SA) turbulence equation [16, 17]. A turbulent working variable, $\tilde{\nu}$, is introduced into the governing equations to calculate the eddy viscosity,

$$\mu_t = \begin{cases} \rho\tilde{\nu}f_{v1}, & \tilde{\nu} \geq 0 \\ 0, & \tilde{\nu} < 0 \end{cases}, \quad f_{v1} = \frac{\chi^3}{\chi^3 + c_{v1}^3}, \quad \chi = \frac{\tilde{\nu}}{\nu}, \tag{9}$$

where $\nu$ is the kinematic viscosity. To compute the turbulent working variable, the SA equation provides an additional conservation equation,

$$\partial_j(\rho\tilde{\nu}v_j) - \partial_j\left[\frac{1}{\sigma}\rho(\nu + \tilde{\nu}f_n)\partial_j\tilde{\nu}\right] = S_{\tilde{\nu}}, \tag{10}$$

where $S_{\tilde{\nu}}$ is a source term made up of production, destruction, and diffusion components,

$$S_{\tilde{\nu}} = P - D - \frac{1}{\sigma}(\nu + \tilde{\nu}f_n)\partial_j\rho\partial_j\tilde{\nu} + \frac{c_{b2}\rho}{\sigma}\partial_j\tilde{\nu}\partial_j\tilde{\nu}. \tag{11}$$

The definition of the remaining terms, taken from the SA-neg model (which admits negative working-variable values), can be found on the NASA turbulence modeling resource [18] or in the original paper by Allmaras et al. [17].

The turbulence variable is often multiple orders of magnitude lower than the rest of the state variables. It can also vary three to four orders of magnitude itself within a typical aerodynamic analysis. To ensure that the global system of RANS-SA equations is well conditioned, the turbulence variable is normalized to be closer in magnitude to the other state variables [19],

$$\tilde{\nu}_s = \frac{\tilde{\nu}}{ND}, \tag{12}$$

where $ND$ is a scaling factor. In this work, $ND = \sqrt{\nu_\infty}$. This brings the turbulence working variable in line with the other state variables, but it is also beneficial to have the residual magnitudes similarly adjusted. Equation 10 is divided by the scaling factor and $\tilde{\nu}$ is replaced with $\tilde{\nu}_s ND$,

$$\partial_j(\rho\tilde{\nu}_s v_j) - \partial_j\left[\frac{1}{\sigma}\rho(\nu + ND\tilde{\nu}_s f_n)\partial_j\tilde{\nu}_s\right] = \frac{S_{\tilde{\nu}}}{ND}. \tag{13}$$

This is still a dimensional set of equations because $ND$ is prescribed to have no units. Each component of the turbulent variable source term can be normalized independently, with all of the terms previously described computed using the transformation in equation 12.

In this work, we make use of two interior forcing point methods. The discussion of these methods will use a wall-aligned coordinate frame where $\xi$ is the wall-tangential direction and $\eta$ is the wall-normal direction. The discussions of velocity profiles in the boundary layer use non-dimensional velocity,

$$u^+ = \frac{q_t}{u_\tau}, \tag{14}$$

where $q_t$ is the wall-tangential velocity, and the non-dimensional wall-normal distance is

$$y^+ = \frac{y u_\tau}{\nu}, \tag{15}$$

where $u_\tau$ is the friction velocity,

$$u_\tau = \sqrt{\nu \left.\frac{\partial u}{\partial y}\right|_{\text{wall}}}. \tag{16}$$

## 2.1 Analytical Spalart–Allmaras Wall Function

The first boundary layer approximation is the analytical SA wall function derived by Allmaras in [13]. The wall function provides the tangential velocity as a function of the wall-normal distance and the friction velocity. The SA analytical wall function is derived with the law-of-the-wall assumptions: incompressible, no pressure gradient, constant edge velocity, negligible advection terms, and $|\partial/\partial\xi| \ll |\partial/\partial\eta|$,

$$\begin{aligned} u^+\left(y^+\right) = \bar{B} + c_1 \log\left(\left[y^+ + a_1\right]^2 + b_1^2\right) - c_2 \log\left(\left[y^+ + a_2\right]^2 + b_2^2\right) \\ - c_3 \mathtt{atan2}\left(b_1, y^+ + a_1\right) - c_4 \mathtt{atan2}\left(b_2, y^+ + a_2\right), \end{aligned} \tag{17}$$

where `atan2(y, x)` is an arc tangent of $y/x$ that chooses the correct quadrant and the other coefficients are given,

$$\bar{B} = 5.0333908790505579, \tag{18}$$
$$a_1 = 8.148221580024245, \tag{19}$$
$$a_2 = -6.9287093849022945, \tag{20}$$
$$b_1 = 7.4600876082527945, \tag{21}$$
$$b_2 = 7.468145790401841, \tag{22}$$
$$c_1 = 2.5496773539754747, \tag{23}$$
$$c_2 = 1.3301651588535228, \tag{24}$$
$$c_3 = 3.599459109332379, \tag{25}$$
$$c_4 = 3.6397531868684494. \tag{26}$$

This wall function can be fit to a forcing point using the density and tangential velocity. The density is needed to compute the local kinematic viscosity at the forcing point. In this work, constant viscosity is assumed, but if that were not the case, temperature would also be needed to compute the local kinematic viscosity according to Sutherland's Law. Plugging the kinematic viscosity, wall-tangential velocity, and wall-normal distance from the wall into equation 17 leaves only the friction velocity as an unknown. Newton's method is used to compute the friction velocity, which scales the SA wall function to match the forcing point data. Then, equation 17 can be used to compute velocities. Equation 17 can be differentiated to compute the gradient of the velocity at any $\eta$.

The SA analytical wall function's primary drawback is that the forcing point must lie within the log region of the boundary layer to remain valid. Without convection terms or the pressure gradient, the wake region of the boundary layer cannot be modeled accurately. Additionally, these assumptions result in a linear turbulence variable profile, which can lead to overpredictions of the viscosity when the forcing point is too far from the wall.

### 2.2 Ordinary Differential Equation Wall Model

The ordinary differential equation (ODE) wall model developed by Berger and Aftosmis [20] extends the valid region for the wall model into the wake region of the boundary layer. To extend the validity of the wall model into the wake region, the assumptions made for the SA wall function are relaxed. Tangential momentum and pressure gradient balance are reintroduced. The final wall model is a coupled two-equation system,

$$\begin{aligned}\frac{\partial}{\partial \eta}\left[(\mu+\mu_t)\frac{\partial q_t}{\partial \eta}\right] &= \frac{\partial p}{\partial \xi} + \rho\left[q_t\frac{\partial q_t}{\partial \xi} + q_n\frac{\partial q_t}{\partial \eta}\right]\\ \frac{\partial}{\partial \eta}\left[(\nu+\nu_t)\frac{\partial \tilde{\nu}}{\partial \eta}\right] &= -c_{b2}\left(\frac{\partial \tilde{\nu}}{\partial \eta}\right)^2 - \sigma(P-D),\end{aligned} \tag{27}$$

where $q_n$ is the wall-normal velocity, $P$ is the production term from equation 11, and $D$ is the destruction term from equation 11. The system is closed by the following boundary conditions,

$$q_t(0) = 0, \tag{28}$$
$$q_t(L) = q_t|_F, \tag{29}$$
$$\tilde{\nu}(0) = 0, \tag{30}$$
$$\tilde{\nu}(L) = \tilde{\nu}|_F, \tag{31}$$

where $(\cdot)|_F$ denotes a quantity interpolated at the forcing point and $L$ is the distance of the forcing point from the no-slip wall.

This is a coupled system because the turbulence variable depends on the vorticity generated by the momentum equation and the viscosity in the momentum equation is determined by the turbulence variable. There is no simple analytical solution to this set of equations analogous to that of the SA wall function. This two-equation ODE needs to be solved on the 1D domain between the no-slip wall and the forcing point.

The wall-tangential gradients in the $\xi$ direction introduce unwanted complication. To solve this system of equations, the $\xi$ gradients would need to be computed along the entire 1D domain in a direction perpendicular to the domain. This would require coupling adjacent wall model linelets together because the background mesh resolution cannot provide the necessary fidelity.

To keep each ODE wall model localized to only the wall-normal direction, an activation function is introduced to determine how strong the convective forces are through the boundary layer. Near the wall, the molecular viscous forces should balance out the pressure gradient. Farther, from the wall into the log layer and wake region of the boundary layer, the convective forces balance out the pressure gradient. At a given distance from the wall, $\eta$, the activation function is the ratio between the SA wall function's velocity at that distance to the velocity at the forcing point,

$$\Phi(\eta) = q_{t,SA}(\eta)/q_{t,SA}(L), \tag{32}$$

where $q_{t,SA}$ is the wall-tangential velocity from the SA wall function fit using the forcing point interpolated density and wall-tangential velocity. Then, the $\xi$ derivatives in equation 27 can be approximated with values at the forcing point and the activation function,

$$\begin{aligned}\frac{\partial}{\partial \eta}\left[(\mu+\mu_t)\frac{\partial q_t}{\partial \eta}\right] &= S_{\mathrm{ODE}}(\eta)\\ \frac{\partial}{\partial \eta}\left[(\nu+\nu_t)\frac{\partial \tilde{\nu}}{\partial \eta}\right] &= -c_{b2}\left(\frac{\partial \tilde{\nu}}{\partial \eta}\right)^2 - \sigma(P-D),\end{aligned} \tag{33}$$

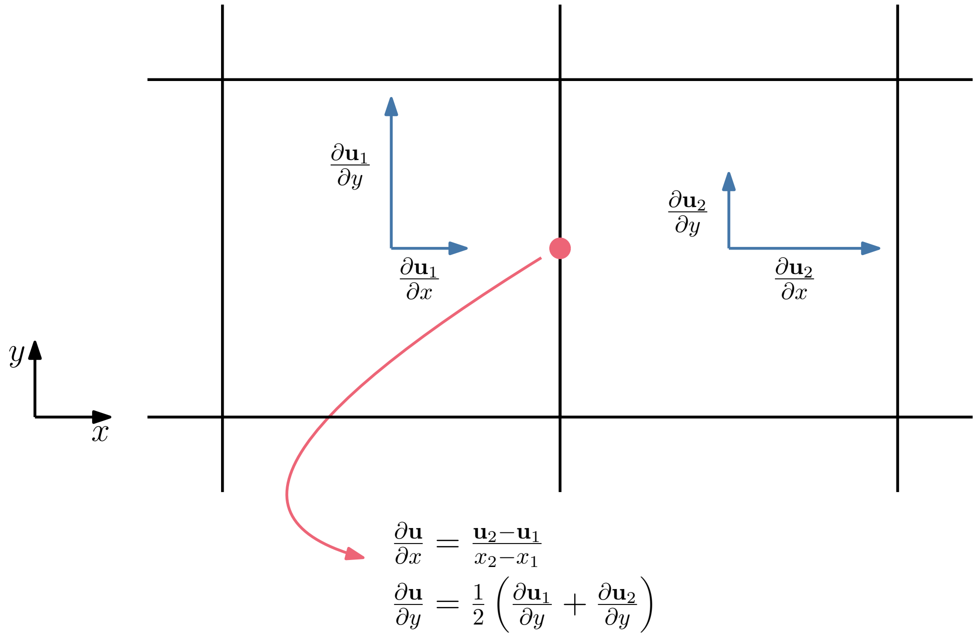



Figure 1: Gradients at interior faces are computed by averaging in the direction of the face normal and differencing in the direction perpendicular to the face normal.

where $S_{\text{ODE}}(\eta)$ is the momentum source term,

$$S_{\text{ODE}}(\eta) = \left.\frac{\partial p}{\partial \xi}\right|_F + \Phi(\eta)\rho|_F \left[ q_t|_F \left.\frac{\partial q_t}{\partial \xi}\right|_F + q_n|_F \left.\frac{\partial q_t}{\partial \eta}\right|_F \right], \tag{34}$$

which only varies with $\eta$. All other quantities are the interpolated values at the forcing point. The activation function captures the blending between viscous and convective pressure balance. This new system depends only on the forcing point values and its own 1D domain. The procedure for solving this system requires careful consideration to ensure that it is both accurate and robust enough to be used within a global flow solver without significant computational cost.

## 3. Methodology

Our flow solver, Viscous Aerodynamic Cartesian Cut cells (VACC), is a 2D second-order finite volume Cartesian cut-cell method [21]. The flow is driven to steady-state using a Jacobian-free Newton–Krylov method with pseudo-transient continuation [14]. We retain second-order accuracy for the turbulence variable. Cell gradients are computed using a least-squares gradient reconstruction technique. Directional limiters are computed by solving a linear-programming problem in each cell to retain maximum gradient components subject to monotonicity and total variation diminishing constraints.

The introduction of viscous terms require gradients at face centroids. At interior faces between two equally sized Cartesian cells, cell centroids are aligned with face centroids. The gradient in the direction perpendicular to the face normal is averaged. However, the gradient in the direction of the face normal must be re-differenced to avoid even-odd decoupling (Figure 1). Sometimes cell centroids do not align, either at refinement boundaries or in cut cells. When this occurs, the same strategy is employed, but with an additional step. The cell centroid states are reconstructed perpendicular to the face normal to align with the face centroid (Figure 2) before differencing.

In cut cells, the interior forcing point boundary layer approximation computes the velocity gradient at face centroids. The nonlinear behavior of the velocity in the boundary layer makes the normal approach a poor fit. The forcing point method in this work uses the strategy shown in Figure 3. Each boundary

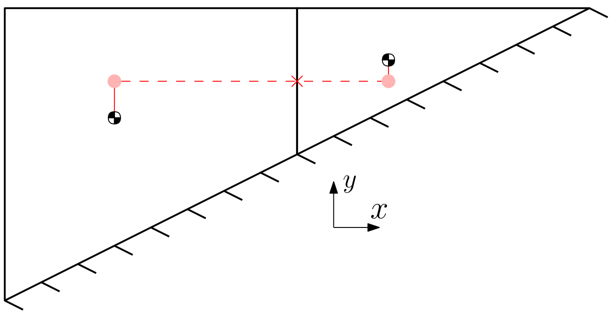



Figure 2: When cell centroids are not aligned, the averaged gradient is computed identically. The differenced gradient recenters the cell centroid state with the least-squares gradient to align with the face centroids before differencing.

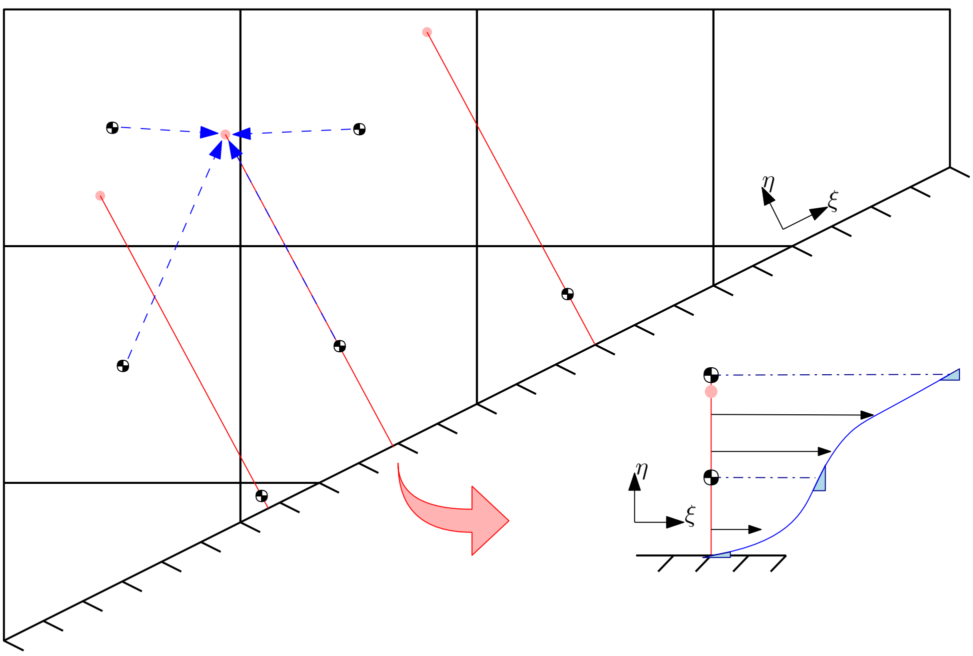
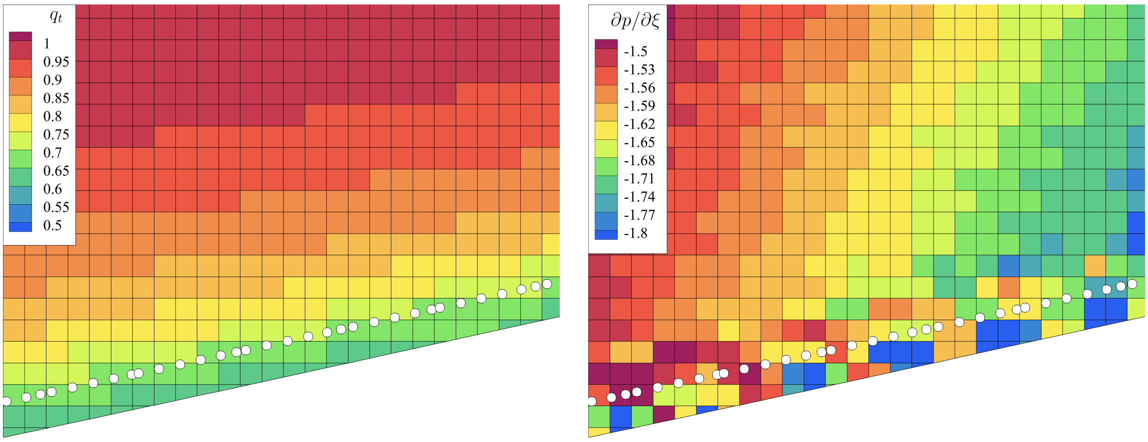



Figure 3: The forcing point constructs an interpolation stencil from nearby cell centroids. The states at the forcing point are interpolated using a least-squares approach. The resulting boundary layer approximation is used to inform velocities and velocity gradients for viscous fluxes at all faces adjacent to cut cells. This approximation also informs the turbulence variable source term through the vorticity.



(a) Wall-tangential velocity. (b) Wall-tangential pressure gradient.

Figure 4: Wall-tangential velocity and pressure gradients on a 2D turbulent bump case. White dots indicate forcing point locations. Irregular cut cells near the wall result in a pressure gradient field that is much noisier than the velocity field. This behavior is visible even on inviscid solutions.

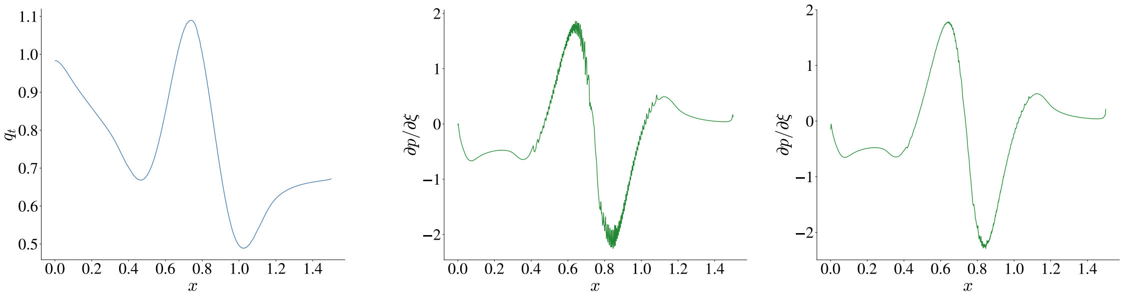
face computes a forcing point by drawing a line that starts at the boundary and passes through the cut-cell centroid. The forcing point is placed a distance $h = 1.6 \times \max(dx, dy)$ from the boundary along this line, where $dx$ and $dy$ are the $x$ and $y$ widths of the cell, respectively. The constant 1.6 was chosen to ensure that the forcing point lies outside the cut cell that contains the no-slip wall boundary condition. The distance $h$ is a constant to compute smooth skin frictions even with varying cut-cell sizes [12]. During a residual evaluation, a state is reconstructed to the forcing point using a stencil of nearby cells. The reconstructed state is used to approximate the boundary layer between the forcing point and the no-slip wall. Finally, the boundary layer approximation is used to inform flux evaluations in and around cut cells.

### 3.1 Forcing Point Interpolation

Both the SA wall function and the ODE wall model must interpolate quantities from the background mesh to approximate the boundary layer. The SA wall function interpolates $\rho$ and $q_t$. In addition to these quantities, the ODE must also interpolate $\partial p/\partial \xi$, $q_t \partial q_t/\partial \xi + q_n \partial q_t/\partial \eta$, and $\tilde{\nu}$. The primal quantities used by the SA wall function are smooth (Figure 4a), but the gradient quantities used by the ODE wall model are noisy (Figure 4b). Interpolating the least-squares gradients at forcing points can lead to additional noise in the system if not handled properly.

Gradients in cut cells have irregular truncation errors, which result in noisy least-squares gradients in and around cut cells. This is a phenomenon that is visible even for inviscid cases where cut cells are used extensively [22, 23]. The problem is not realized in inviscid flows because the gradient quantities are not directly needed to evaluate fluxes. Limiters often reduce gradients to negligible values in problematic areas to enforce stringent monotonicity constraints. This smoothes out inviscid solutions, but reduces the accuracy of viscous flux evaluations. Velocities and their gradients change rapidly inside the boundary layer and must be accurate for a large range of wall distances.

The forcing point data are interpolated using a least-squares gradient reconstruction technique similar to the cell centroid gradients. The forcing point interpolation stencil comprises the cell containing the forcing point along with all of that cell's nearest and corner neighbors. To provide a better fit within the




(a) Wall-tangential velocity. (b) Original pressure gradient. (c) Corrected pressure gradient.

Figure 5: Interpolated tangential velocity and tangential pressure gradient around a two-dimensional bump. Oscillations are clearly visible in the pressure gradient not visible in the tangential velocity. These oscillations can cause oscillations in the final skin friction. A less accurate forcing point reconstruction stencil for the pressure gradient smooths the oscillations.

boundary layer, we use a quadratic reconstruction in the wall-normal direction,

$$\begin{bmatrix} 1/||\vec{r}_1|| & r_{\xi,1}/||\vec{r}_1|| & r_{\eta,1}/||\vec{r}_1|| & 0.5r_{\eta,1}^2/||\vec{r}_1|| \\ 1/||\vec{r}_2|| & r_{\xi,2}/||\vec{r}_2|| & r_{\eta,2}/||\vec{r}_2|| & 0.5r_{\eta,2}^2/||\vec{r}_2|| \\ \vdots & \vdots & \vdots & \vdots \\ 1/||\vec{r}_n|| & r_{\xi,n}/||\vec{r}_n|| & r_{\eta,n}/||\vec{r}_n|| & 0.5r_{\eta,n}^2/||\vec{r}_n|| \end{bmatrix} \begin{bmatrix} C|_F \\ \partial C/\partial \xi|_F \\ \partial C/\partial \eta|_F \\ \partial^2 C/\partial \eta^2|_F \end{bmatrix} = \begin{bmatrix} C_1/||\vec{r}_1|| \\ C_2/||\vec{r}_2|| \\ \vdots \\ C_n/||\vec{r}_n|| \end{bmatrix}, \tag{35}$$

where $C$ is the quantity being interpolated, $\vec{r}$ is the vector pointing from the forcing point to a stencil point, and $n$ is the number of stencil points.

When interpolating states in the boundary layer, changing even a single cell in the interpolation stencil can have a huge impact on the outcome. Each equation in the linear system is weighted by the inverse distance between the forcing point and that stencil point to make states closer to the forcing point more influential than those farther away. This prevents sudden jumps in the interpolated states when neighboring forcing points shift across parent cell boundaries, changing the interpolation stencil.

Figure 5a demonstrates that this interpolation procedure works well for the primal quantities. However, Figure 5b shows that accurately interpolating the gradient quantities in this way can lead to noise. This noise will persist through the ODE wall model boundary layer approximation resulting in noisy skin friction and other quantities of interest.

To provide smooth approximations of gradient quantities near the wall, a different interpolation procedure is used. The gradient interpolation reverts to a linear reconstruction in the wall-normal direction,

$$\begin{bmatrix} 1 & r_{\xi,1} & r_{\eta,1} \\ 1 & r_{\xi,2} & r_{\eta,2} \\ \vdots & \vdots & \vdots \\ 1 & r_{\xi,n} & r_{\eta,n} \end{bmatrix} \begin{bmatrix} C|_F \\ \partial C/\partial \xi|_F \\ \partial C/\partial \eta|_F \end{bmatrix} = \begin{bmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{bmatrix}. \tag{36}$$

Additionally, the inverse distance weighting is removed, which reduces the importance of any individual stencil point. Figure 5c demonstrates the improvements this interpolation procedure provides for the gradient quantities.

Recent work done by Zhang et al. [24] suggests that using the reconstructed pressure gradient to approximate the convective terms can alleviate issues with noise in the forcing point quantities. We saw minor improvements when using the quadratic reconstruction in Equation 35 for the pressure gradient in conjunction with the approach presented by Zhang et al. [24] for the convective term. However, switching to the linear reconstruction in Equation 36 for all gradient quantities provided significantly smoother results.

### 3.2 ODE Wall Model Discretization

Thousands of ODEs will be solved each global flow residual evaluation. If any of the ODEs fail, then the residual cannot be evaluated. Almost all of the runtime for an explicit or matrix-free implicit global flow solution is spent evaluating residuals. Therefore, ensuring that the ODE solutions are computed efficiently and robustly is of the utmost importance.

The ODE wall model is dominated by viscous diffusion, with no advection operator. The convection terms in the ODE are constants with respect to the spatial dimension of the ODE. To solve the ODE in equation 33, a high-order continuous Galerkin discretization was implemented. The solver uses a Newton–Krylov method with pseudo-transient continuation similar to the background flow solver to drive the residuals to zero. All results in this work use seven $p = 4$ elements. The optimal distribution of mesh points is found using mesh optimization via error sampling and synthesis (MOESS) [25]. This ensures solutions are as accurate as possible given the computational cost of the wall model. All of the ODE meshes are put through ten mesh adaptation cycles before starting the background flow solution (Figure 6).

This solution strategy is able to handle many different boundary conditions. However, the stiffness of the momentum source terms, particularly in wall models near stagnation points, can result in ODE solution failures. In these circumstances, it is still essential that the wall model is able to provide a boundary layer approximation to avoid stalling global flow convergence.

To address this issue, we developed a continuation-based method that scales the momentum source term based on the difficulty of computing solutions to the ODE wall model. The momentum equation in equation 33 is modified by a $\beta$ parameter,

$$\frac{\partial}{\partial \eta}\left[(\mu + \mu_t)\frac{\partial u}{\partial \eta}\right] = \beta S_{\mathrm{ODE}}(\eta), \tag{37}$$

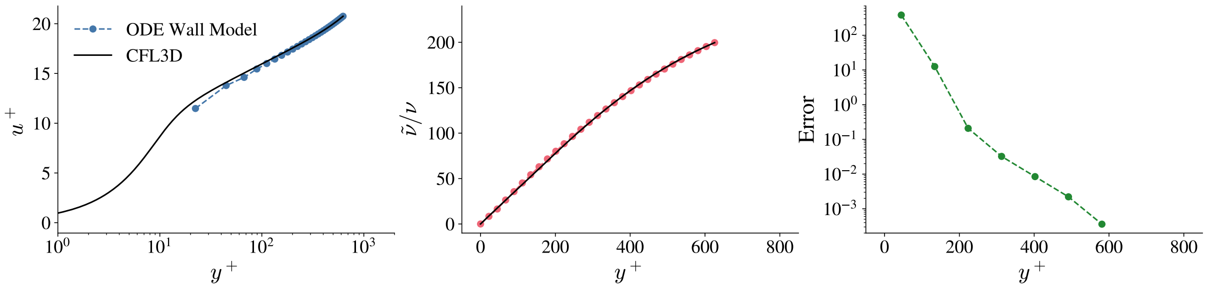
where $\beta \in [0, 1]$. When an ODE solution fails, $\beta$ is reduced by 0.01. The balance between the pressure gradient and convection is crucial for this wall model, and therefore $\beta$ is applied across the entire source term. This is sufficient to protect the ODE solver except on particularly coarse meshes. Even without the convection source term, the ODE is theoretically more accurate than the SA wall function because it allows for nonlinear variation in the turbulence variable. In practice this difference is minuscule compared to the change once the convection source term is added. As a last resort, when the wall model does not converge with $\beta = 0$, the ODE is discarded entirely for the analytical SA wall function.

## 4. Results

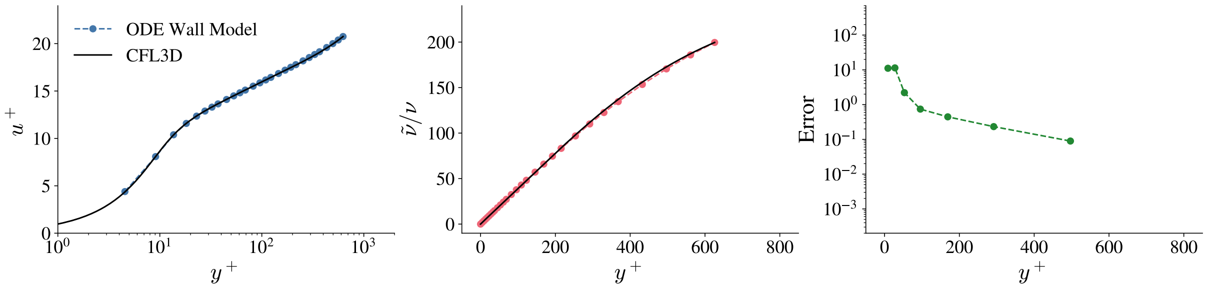
To demonstrate the capability of the described method, we present results from a series of increasingly complex NASA turbulence modeling resource (TMR) problems [18]. The TMR is a database that contains state-of-the-art results for a variety of RANS turbulence models on a variety of test problems. The TMR also provides reference material for implementing many RANS turbulence models. CFL3D [26] and FUN3D [27] results from the TMR are considered *the reference* solutions to match [28].

### 4.1 2D Bump

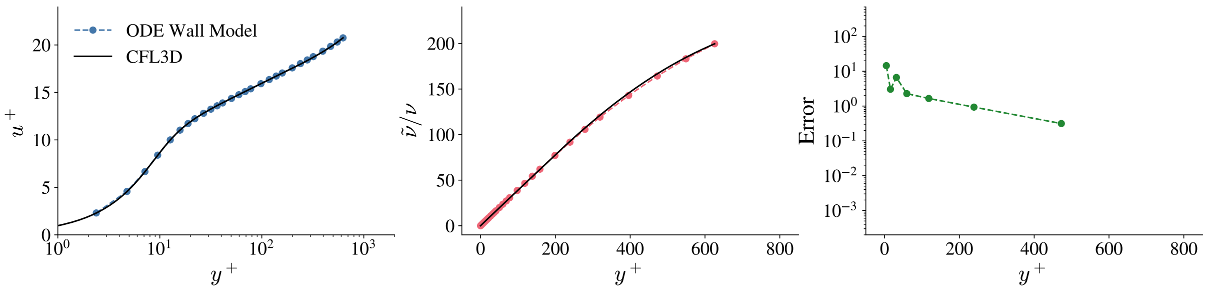
The first test case is the 2D bump. It clearly demonstrates the differences between the SA wall function and the ODE wall model and highlights the importance of the gradient interpolation procedure. The 2D bump is run at Mach 0.2 and Reynolds number $3 \times 10^6$. The results are presented against the CFL3D results provided by the TMR.



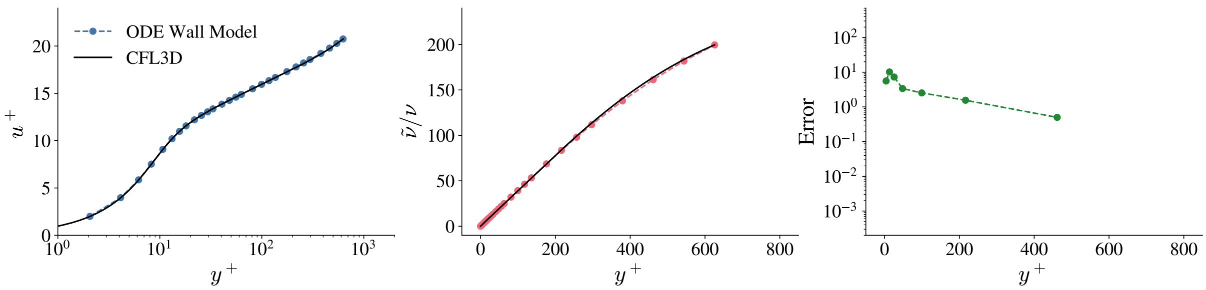

(a) Initial mesh with 7 linearly spaced $p = 4$ elements.




(b) Mesh after three MOESS cycles with 7 elements.




(c) Mesh after six MOESS cycles with 7 elements.




(d) Mesh after nine MOESS cycles with 7 elements.

Figure 6: MOESS sequence on the ODE with $p = 4$ elements. The initial mesh is seven linearly spaced elements. These meshes are adapted on the gradient of the velocity at $y = 0$. A nodal basis is used, so the mesh points correspond to the basis function locations. The $u^+$ and $\tilde{\nu}$ plots show points at each basis function's global space location. The error plots show points at the middle of each element. The analytic SA wall function shown is the one used in the $x$-momentum source in equation 33. It is not necessarily accurate given the forcing point location.

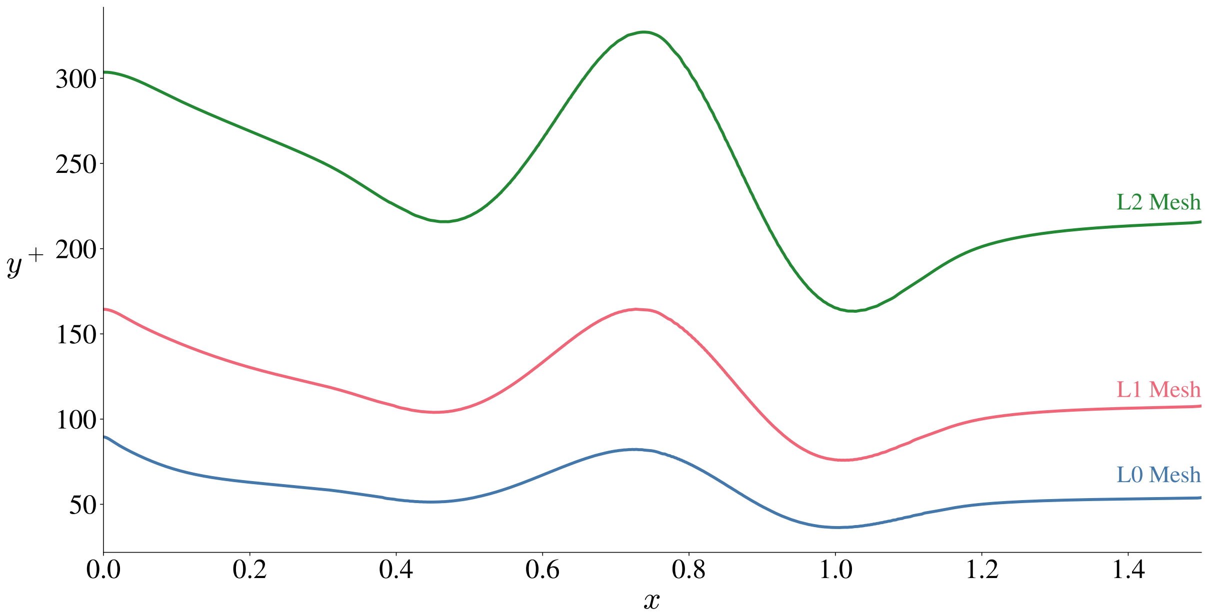



Figure 7: Forcing point $y^+$ values on the ODE wall model solution for the turbulent bump case. The forcing points are well into the wake region of the boundary layer, particularly on the coarse mesh.

The L0 mesh is the finest and the L2 mesh is the coarsest. The L0 mesh has approximately 220k isotropic cells, and the L2 mesh has approximately 59k isotropic cells. Figure 7 shows the $y^+$ values of the wall-function forcing points along the surface of the bump. It is clear, particularly on the coarse mesh, that the forcing point is entering the wake region of the boundary layer. Figure 8 presents a comparison of the eddy viscosity generated by the bump. The eddy viscosity profiles from VACC are computed on the L1 mesh with the SA wall function. VACC exhibits good agreement with CFL3D.

Figure 9 shows two velocity profiles on the top of the bump ($x = 0.75$) and in the wake region downstream ($x = 1.2$). In all cases, the mesh-converged solution matches the CFL3D solution.

Figure 10 presents the pressure coefficient distributions along the bump. Both the SA wall function and the ODE wall model are able to predict the correct pressures on all of the mesh levels. There are very small spurious pressure spikes around triangle cut cells that can barely be seen at this plotting resolution. This is a known phenomenon that exists for Cartesian cut-cell solutions for inviscid problems. Cut cells result in poor gradient stencils and nonuniform truncation errors (Figure 4b).

Figure 11 presents the skin friction distributions along the bump. The SA analytical wall function matches the CFL3D solution well. The phase shift seen in the skin friction distributions occurs because the SA wall function assumes no pressure gradient. As the mesh resolution increases, the skin friction approaches the expected profile. The ODE wall model does not exhibit the same phase shift but is noisy. As the mesh is refined, the magnitude of the noise is reduced. The ODE allows the coarser meshes to fix the phase shift without needing additional mesh resolution.

The downturn in the ODE skin friction near the front of the no-slip wall is due to the localized nature of the wall functions. Near the front of the bump, the boundary layer is extremely thin and the forcing points lie well into the inviscid region of the flow. This makes all forcing point data look identical even though the boundary layer thickness is not. Since the wall functions do not communicate with those in front of or behind them in the wall-tangential direction, they have no notion of how thick the boundary layer should be when given data that lies outside the boundary layer. The SA and ODE wall functions have the same issue. In both cases, the skin friction at the leading edge of the bump converges under mesh refinement.

The ODE wall model and SA wall function both provide credible results. The ODE wall model predicts skin frictions more accurately when the boundary layer is thin. The smooth gradient interpolation

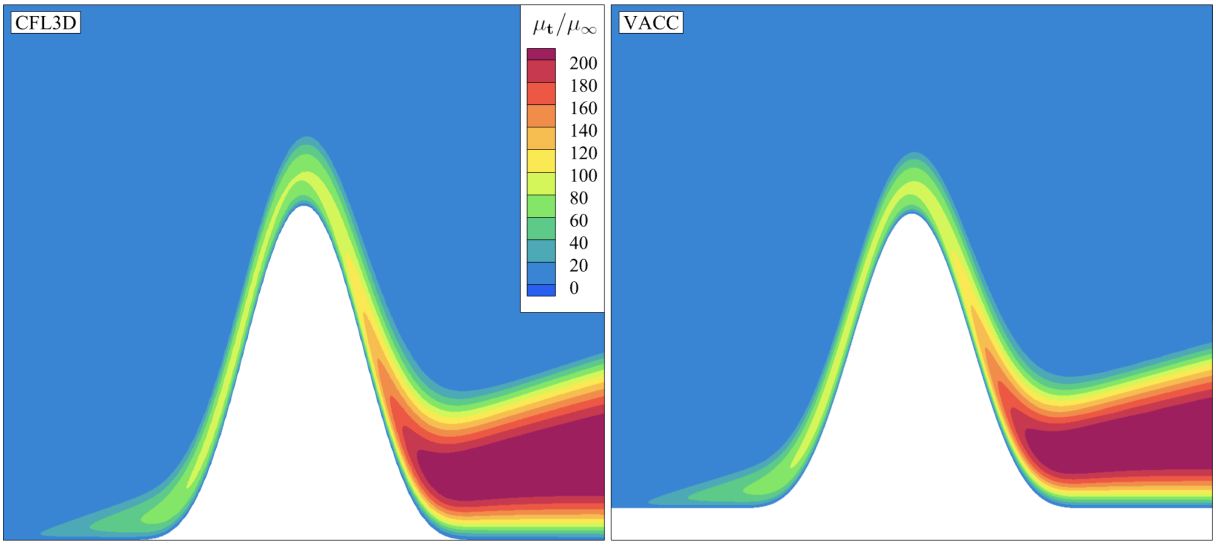
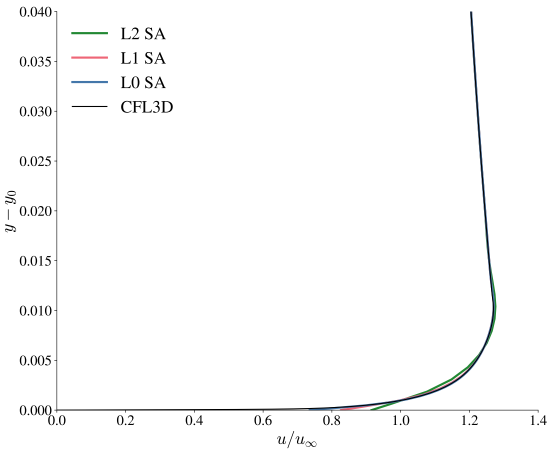
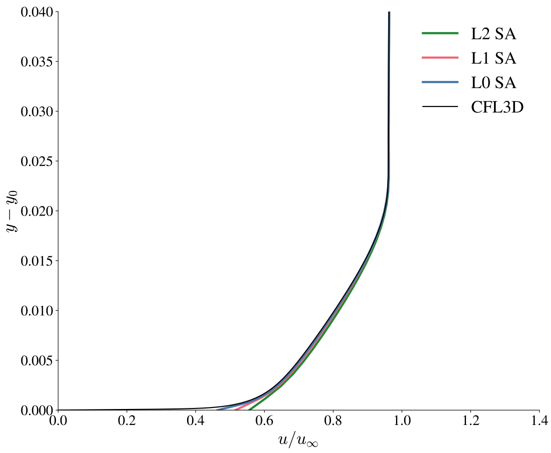


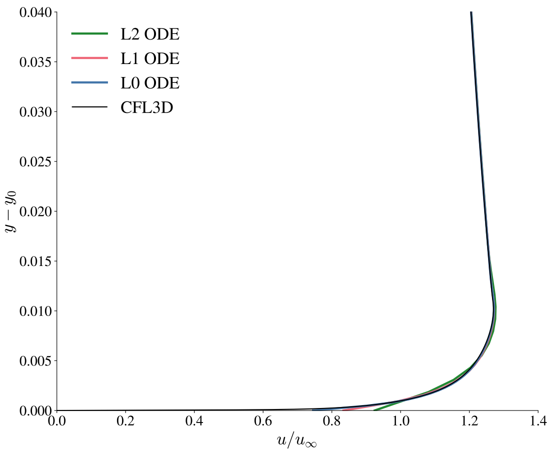

Figure 8: Turbulent eddy viscosity contours around the bump. The conditions for this case come from the NASA turbulence modeling resource. It was run at Mach 0.2 with a Reynolds number of $3 \times 10^6$. The CFL3D solution matches the SA wall function on the L1 mesh. The $y$-axis is scaled to be approximately twenty times larger than the $x$-axis.



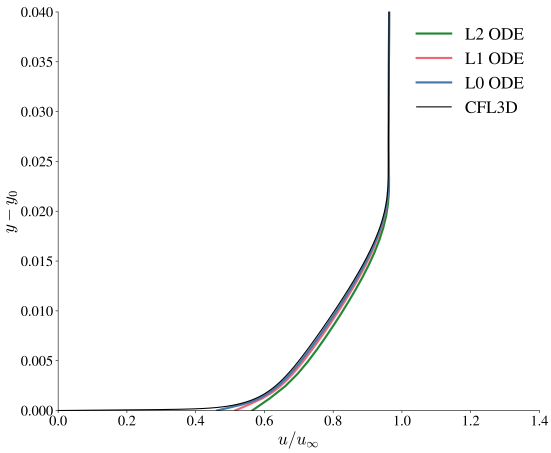

(a) SA wall function at $x = 0.75$




(b) SA wall function at $x = 1.20148$

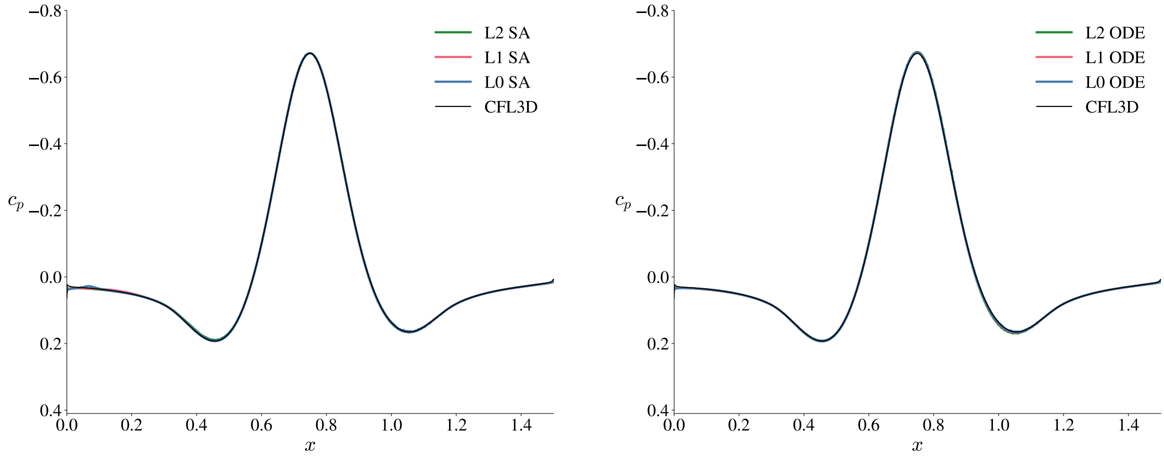



(c) ODE wall model at $x = 0.75$




(d) ODE wall model at $x = 1.20148$

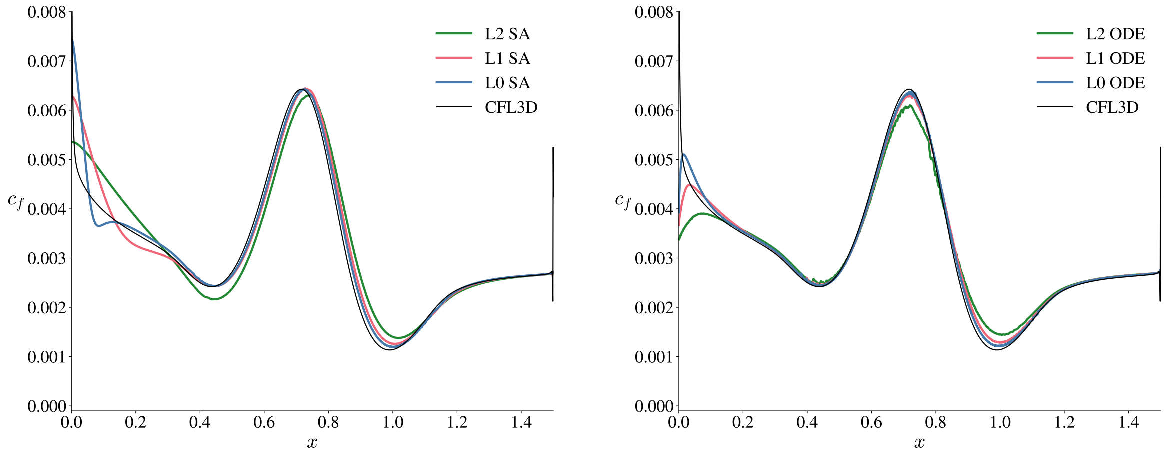
Figure 9: Velocity profiles for the 2D bump case. Both the SA and the ODE wall functions match the CFL3D solution data.




(a) SA Analytical Wall Function

(b) ODE Wall Function

Figure 10: The pressure coefficient distributions for the 2D bump case. The pressure matches the CFL3D solution well on all mesh levels.




(a) SA Analytical Wall Function

(b) ODE Wall Function

Figure 11: The skin friction distributions for the 2D bump case match the CFL3D solutions. Each wall function exhibits convergence towards the expected solution as the mesh is refined. The effect of the pressure oscillations in the solution is clear on the coarser ODE solution.

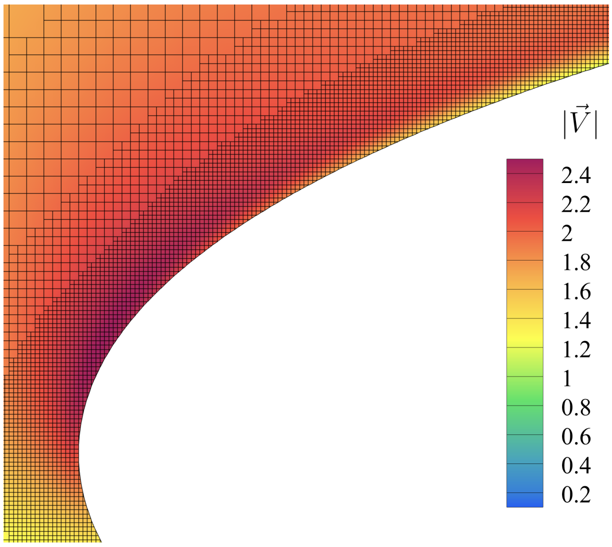



Figure 12: Velocity contours on the L1 mesh for the TMR NACA 0012 case near the leading edge and suction peak. The L1 mesh has a total of 258k cells. The freestream velocity is set to 1.

technique controls the background gradient noise effectively. The $\beta$ continuation parameter was not necessary for the bump case.

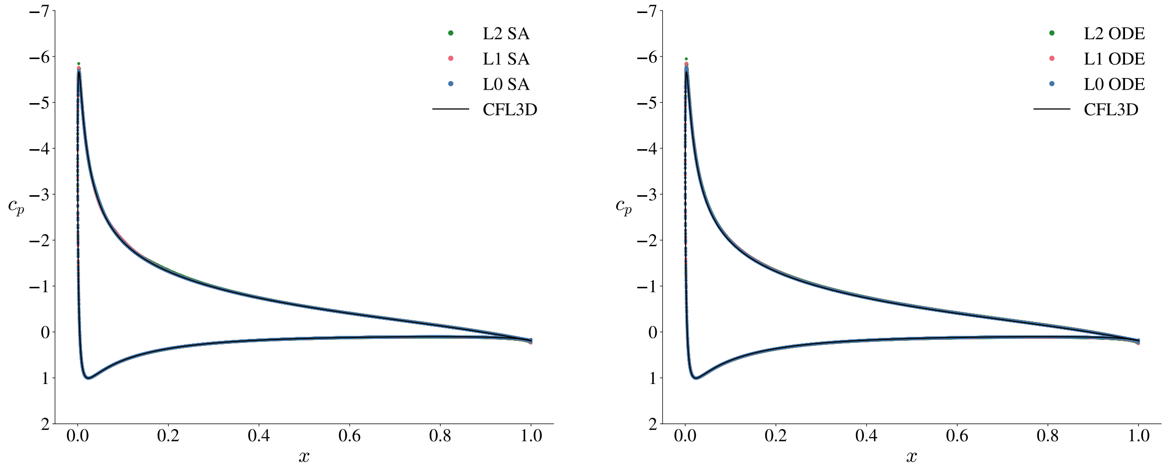
### 4.2 NACA 0012 Airfoil

The second case is the NACA 0012 airfoil. It is run at Mach 0.15 and Reynolds number $6 \times 10^6$ at $10°$ angle of attack [18]. This case introduces a stagnation point typical of external, aerodynamic flows. It also introduces a much wider range of $y^+$ values with forcing points on the pressure side farther outside the boundary layer than those on the suction side. The L0 mesh is the finest with about 430k cells. Figure 12 shows the L1 mesh, which has about 258k cells. The L2 mesh is the coarsest with around 126k cells.

Figure 13 presents the pressure coefficient distributions for the SA and ODE wall functions on the NACA 0012 airfoil. The pressure coefficient distributions match the CFL3D solutions for both wall functions at all mesh resolutions. Figure 14 presents the skin friction coefficient distributions on the NACA 0012 airfoil. These skin friction plots begin to show the benefits of the momentum balance source term in the ODE wall model. The SA wall function skin friction distributions on the upper surface struggle to converge to the expected skin friction values. The L2 mesh skin friction is not able to recover from the suction peak until around 50% of the chord. Even on the finest mesh, the skin friction does not recover until around 15% of the chord. On the other hand, the ODE wall model is able to match the upper surface skin friction before 15% of the chord on even the coarsest mesh. Both wall functions exhibit mesh convergence, trending towards the CFL3D solution as the mesh is refined.

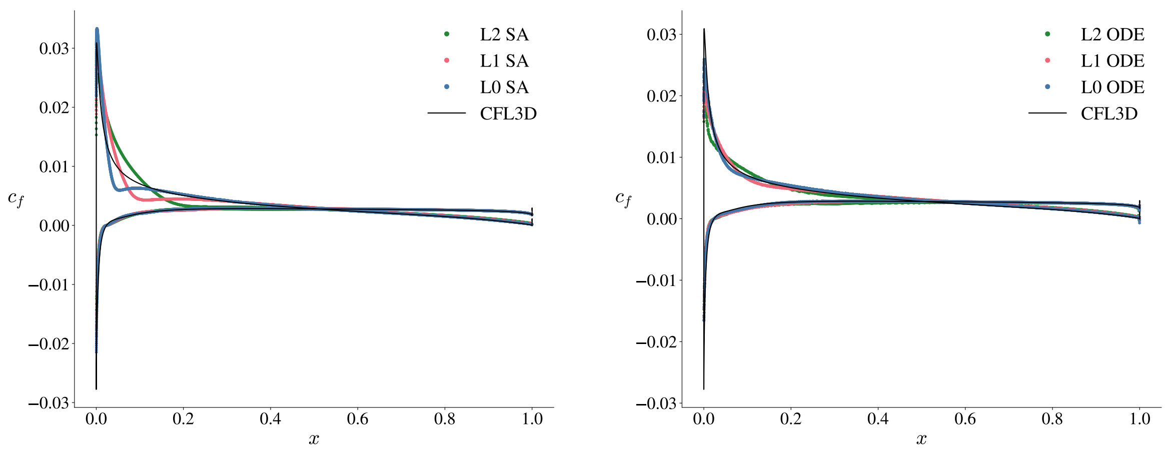
Skin friction oscillations are smaller for this case when compared to the turbulent bump. As the strength of the global pressure gradient across the geometry increases, the noise visible in the local pressure gradient is reduced. The nonuniform discretization error in the cut cells remains of constant magnitude. With the stronger pressure gradients present in airfoil problems, the signal-to-noise ratio increases dramatically.

Figure 15 compares $x$-velocity and eddy viscosity in the wake immediately downstream of the airfoil ($x/c = 1.001$) to reference data from the TMR. The tangential velocities match the CFL3D solution well. Both the SA and ODE wall functions converge towards the CFL3D solution as the mesh is refined. The wall functions were not used to extract these lines; the velocity profiles come from the background mesh. The SA wall function eddy viscosity matches the CFL3D eddy viscosity well on the finest mesh. The ODE wall model is better able to capture the eddy viscosity going to zero at $y = 0$ than the SA wall function on coarser meshes. However, it fails to reach the peak on the upper surface. This behavior is



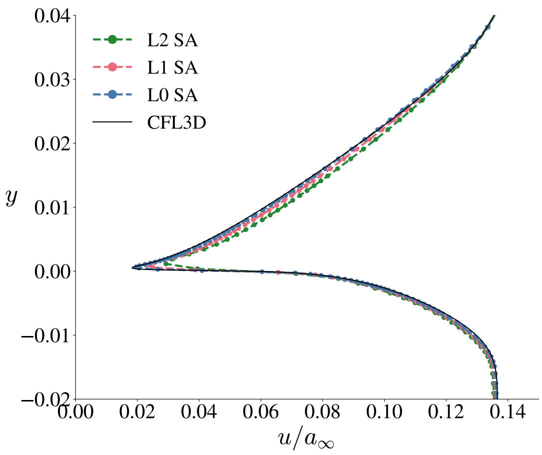

(a) SA wall function (b) ODE wall model

Figure 13: The pressure coefficient distributions for the NACA 0012 airfoil case. The pressures match the CFL3D solution well on all levels of meshes for both wall functions.

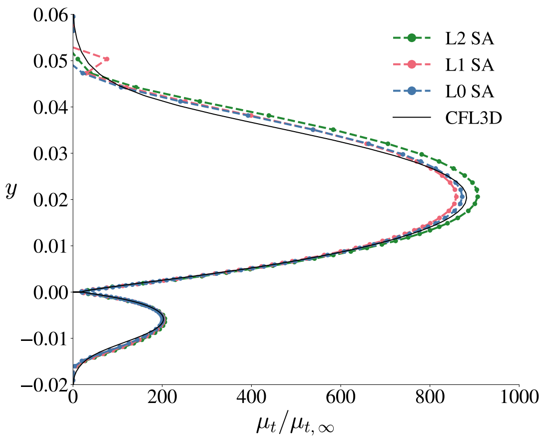



(a) SA wall function (b) ODE wall model

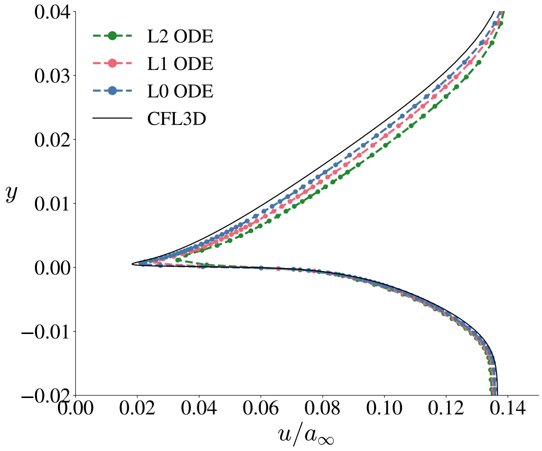
Figure 14: The skin friction coefficient distributions for the NACA 0012 airfoil case. The SA wall function skin friction struggles to match the skin frictions in the presence of the strong pressure gradient near the suction peak. The ODE wall model is able to capture the suction peak pressure gradient better.



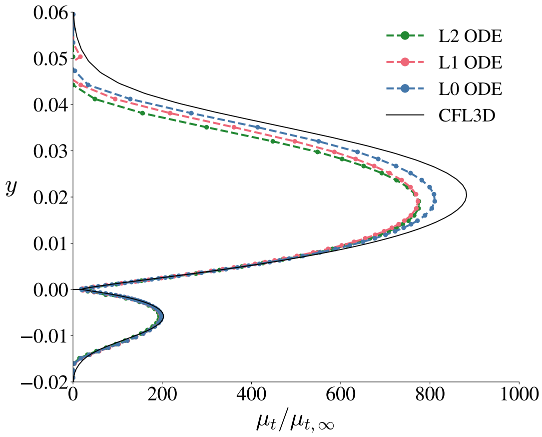

(a) SA wall function tangential velocity




(b) SA wall function eddy viscosity

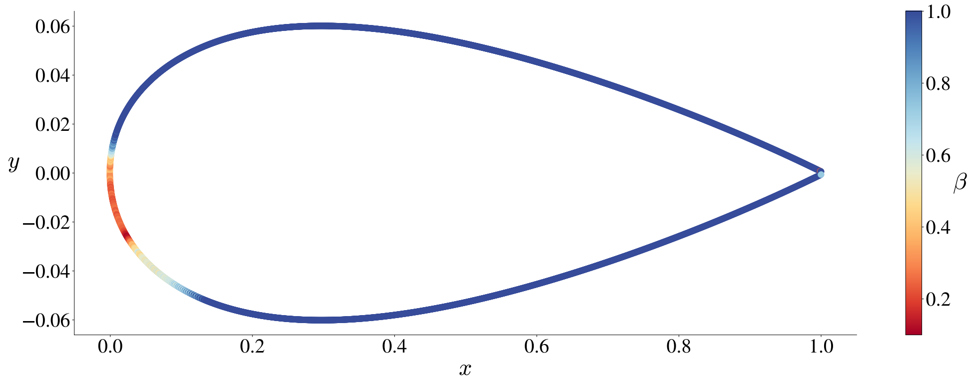



(c) ODE wall model tangential velocity

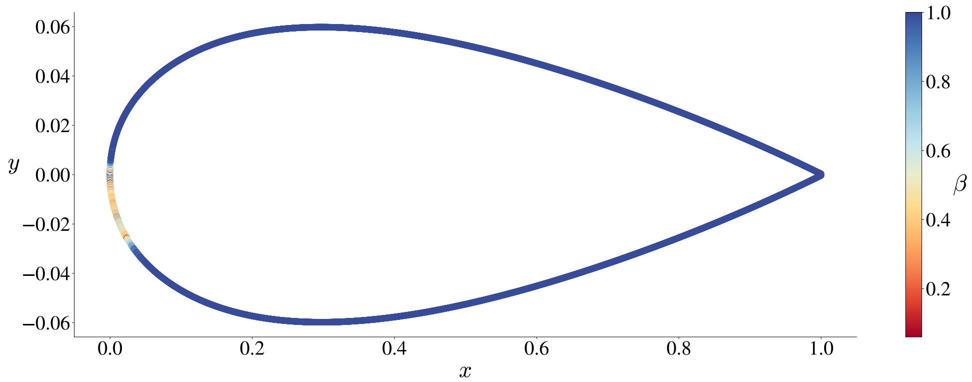



(d) ODE wall model eddy viscosity

Figure 15: The tangential velocity and eddy viscosity profiles at the trailing edge of the airfoil. Both methods are able to closely follow the troughs in both quantities near the wall.




(a) Medium mesh




(b) Fine mesh

Figure 16: The final $\beta$ parameter values for the NACA 0012 case from equation 37. The x-axis is stretched about two times larger than the y-axis. $\beta$ is reduced near the stagnation point on the airfoil. Refining the mesh reduces the percentage of wall models where $\beta$ is reduced.

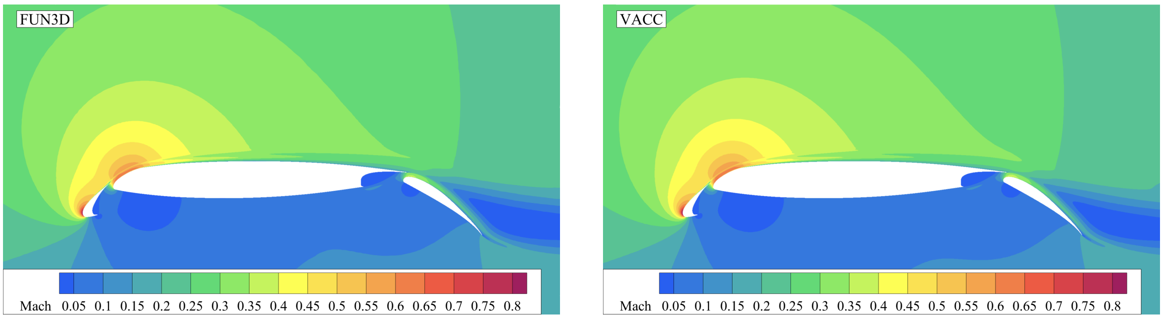
Table 1: Mesh refinement leads to a smaller percentage of ODE wall models reducing source term strength. The spatial area where the $\beta$ reduction is needed also shrinks.

| **Mesh** | **#** $\beta < 1$ | % $\beta < 1$ |
|---|---|---|
| Coarse | 403 | 13.75% |
| Medium | 537 | 9.16% |
| Fine | 399 | 3.40% |

dependent upon turbulence variable profile startup strategies near the leading edge. Choosing different coupling methods can have a profound impact on the startup of the boundary layer where the wall function forcing points are too far outside of the boundary layer to have correct physics. This is true of both the SA and ODE wall functions, but because the ODE wall model depends on more states in the background mesh, it often displays these differences more readily.

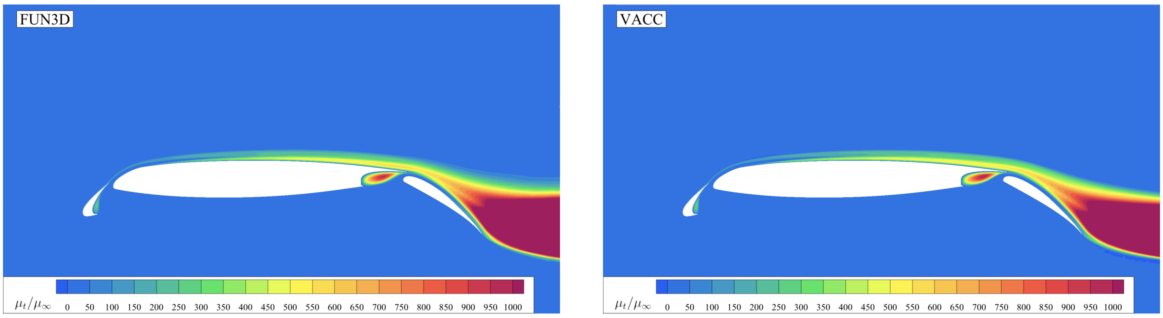
Figure 16 shows the distribution of the $\beta$ parameter from equation 37. A small fraction of wall models have to reduce the strength of the source terms to converge to solutions. The wall models that struggle the most are those located at and around the stagnation point. Around the stagnation point, the assumptions used in the wall model are already weak. That $\beta < 1$ only where the wall model's assumptions are already dubious indicates that reducing the source-term magnitude for convergence does not affect the validity of the solution.

Refining the mesh decreases the percentage of wall models that have $\beta$ reduced. Table 1 reports the number of wall models where $\beta < 1$ as a percentage of the total number of wall models across the three meshes. The number of wall models with $\beta < 1$ stays roughly the same even though the total number




(a) FUN3D (b) VACC

Figure 17: Multielement airfoil Mach contours. VACC shows close agreement with FUN3D.




(a) FUN3D (b) VACC

Figure 18: Multielement airfoil eddy viscosity contours. VACC shows close agreement with FUN3D. It captures the shear layer passing from the slat to the main element.

of wall models roughly doubles with each mesh refinement. As a result, the percentage of wall models where $\beta < 1$ shrinks as the mesh is refined. Additionally, the total area on the airfoil affected by the reduced source terms also shrinks. As the mesh is refined, forcing points move closer to the wall and the wall models become easier to solve. This behavior is visible in Figure 16.

The NACA 0012 case demonstrates the value of the methods proposed in this work. The ODE wall model delivers comparable results to body-fitted meshes on affordable meshes. It provides more accurate results on coarser meshes than the SA wall function. Noise in the skin friction is not an issue due to the proposed background gradient interpolator. The global Newton–Krylov solution strategy and wall model continuation strategy converged without issue. The MOESS algorithm provided robust, accurate adjoint-driven mesh adaptation for the ODE wall model.

### 4.3 Multielement Airfoil

The final case is the 2D multielement airfoil case from the NASA TMR [18]. This is the most complex 2D case available in the TMR for turbulence model verification. It is a high-lift configuration run at Mach 0.2, 16° angle of attack, and Reynolds number $5 \times 10^6$. The airfoil is a cross-section taken from the 3D NASA CRM-HL and includes a main element, slat, and flap with small gaps and coves. The geometry definition for the cut-cell mesh generator comes from the finest surface mesh provided by the TMR. The reference data provided by the TMR for this case is FUN3D.

Figure 17 presents Mach contours from VACC and FUN3D. The primary challenge with this case is the thin boundary layer on the slat and the shear layer propagation from each portion of the airfoil to the next. The shear layer on the slat and the flow between the slat and main element generate an inviscid jet that affects the main airfoil boundary layer (Figure 18). The size of mesh elements near the wall is observed through the $y^+$ values in Figure 19. The turbulent variable profiles for the ODE wall model

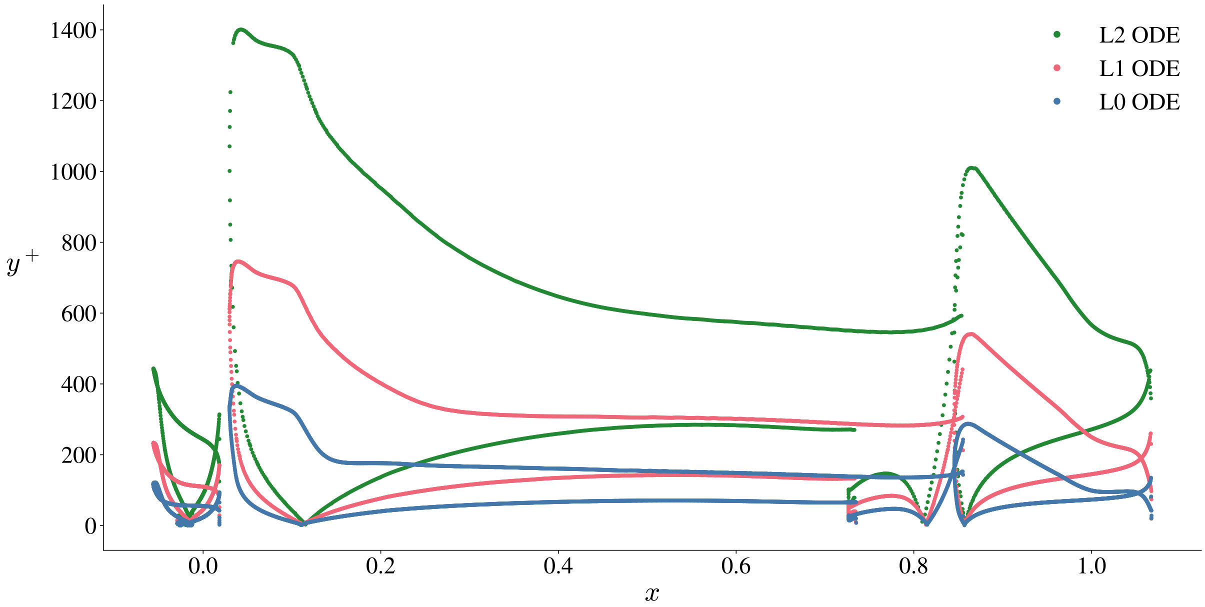
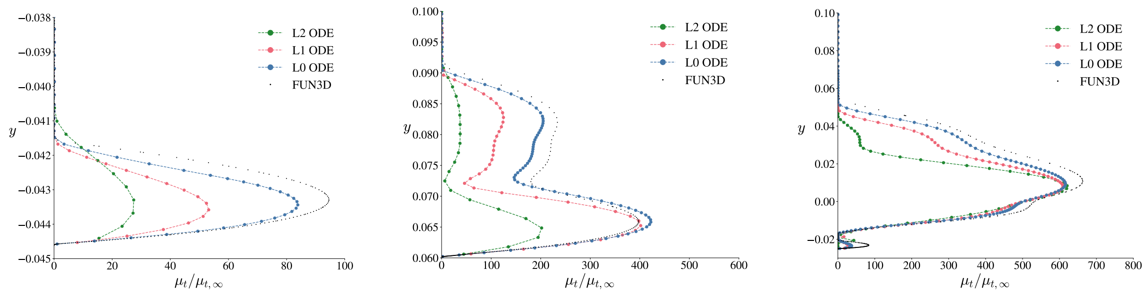



Figure 19: The $y^+$ values along the multielement airfoil. The slat has two additional levels of refinement to capture the rapidly accelerating flow around the first suction peak. This is visible in the low $y^+$ values on the first element.

demonstrate an ability to track the shear layer transfer through each component of the airfoil (Figure 20). Mesh refinement increases the amount of turbulent variable production on the slat, where the boundary layer is thin. This drives the eddy viscosity towards the appropriate solution under mesh refinement, and this translates to more accurate predictions of the eddy viscosity in the transferred shear layer on both the main airfoil and the flap. The NASA TMR demonstrates that the shear layer transfer between elements is extremely sensitive to changes in the mesh between elements. The tangential velocity profiles of the ODE wall model demonstrate similar mesh convergence across all three components (Figure 21).

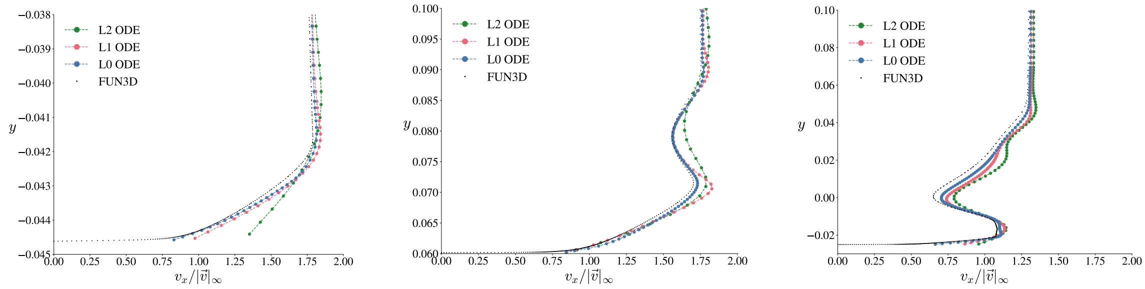
Figure 22 presents the surface pressures on the multielement airfoil. The ODE wall model matches the FUN3D results well and exhibits mesh convergence under refinement. The SA wall function exhibits departure from the FUN3D results, particularly around the suction peaks. The results are not yet mesh converged, demonstrating that the SA wall function requires significantly more resolution than the ODE wall model to remain accurate.

Figure 23 presents the skin friction distributions for the multielement airfoil. The SA wall function exhibits



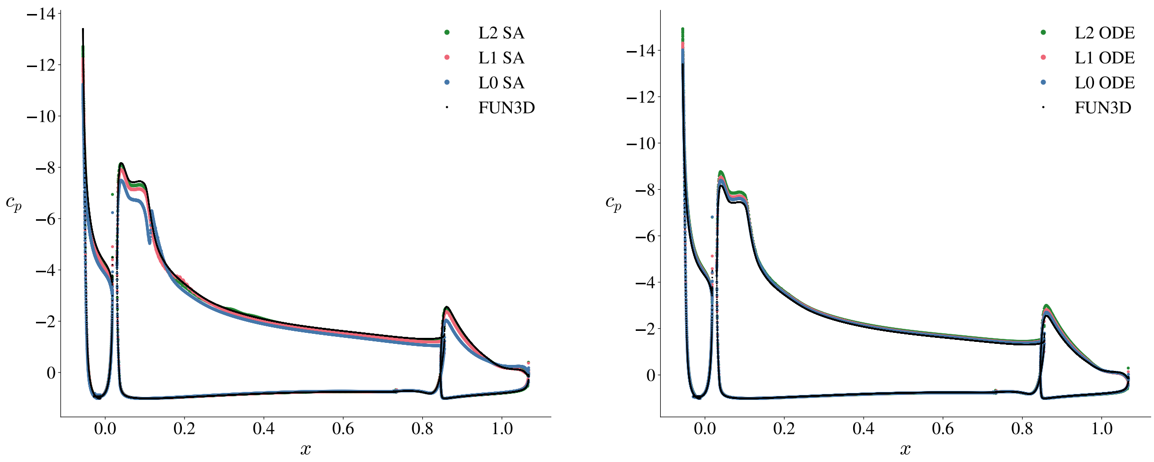

(a) Slat at $x = -0.03$ (b) Main at $x = 0.4$ (c) Flap at $x = 0.95$

Figure 20: ODE wall model eddy viscosity profiles along the airfoil. The ODE wall model solution is converging towards the finest FUN3D solution. Additional mesh resolution increases the turbulent variable production on the slat, which can then be seen carrying through the other two components of the airfoil.



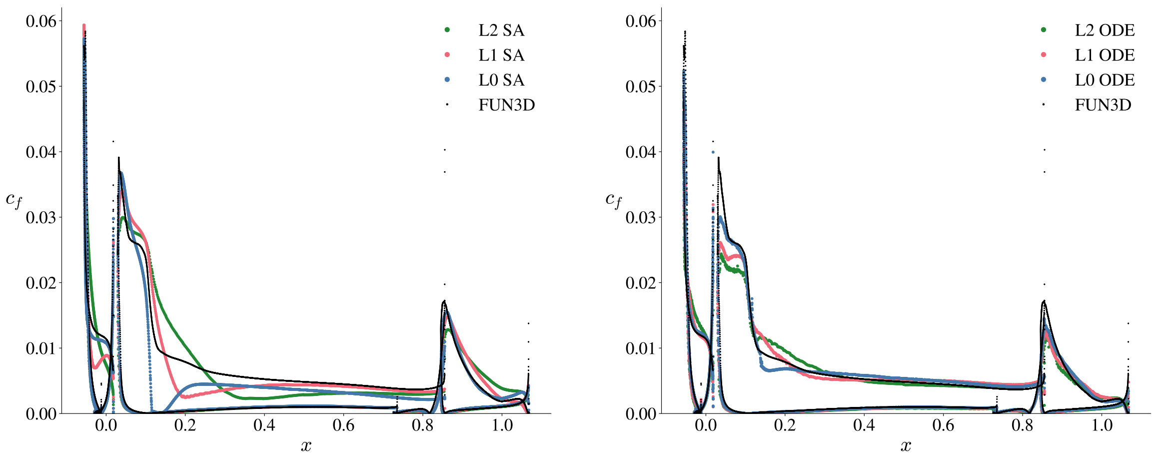

(a) Slat at $x = -0.03$ (b) Main at $x = 0.4$ (c) Flap at $x = 0.95$

Figure 21: ODE wall model tangential velocity profiles along the airfoil. The ODE wall model solution is converging towards the finest FUN3D solution.




(a) SA wall function (b) ODE wall model

Figure 22: Multielement airfoil surface pressure coefficient distributions. The ODE wall model surface pressures match the FUN3D results well and converge towards the solution. The SA wall function exhibits a good match in areas away from the boundary layer startup, particularly on the main element.




(a) SA wall function (b) ODE wall model

Figure 23: Multielement airfoil skin friction coefficients. The SA wall function has a hard time dealing with the coarse mesh over the top of the main element. The startup behavior is similar to that observed on the NACA 0012 case. The ODE exhibits good mesh convergence and matches the fine FUN3D skin frictions.

Table 2: Lift coefficient errors are computed from the finest FUN3D solution ($C_L = 3.785$). The mesh sizes between FUN3D and VACC are similar, but not identical. The order of accuracy of the methods is tabulated in the final row. The SA wall function does not exhibit mesh convergence because the mesh is too coarse.

| **Mesh** | **FUN3D** | **SA** | **ODE** |
|---|---|---|---|
| Coarse | $1.17 \times 10^{-1}$ | $7.86 \times 10^{-2}$ | $1.96 \times 10^{-1}$ |
| Medium | $9.21 \times 10^{-2}$ | $1.44 \times 10^{-1}$ | $1.19 \times 10^{-1}$ |
| Fine | $5.11 \times 10^{-2}$ | $3.05 \times 10^{-1}$ | $7.82 \times 10^{-2}$ |
| **Order** | **1.45** | **N/A** | **1.50** |

Table 3: Drag coefficient errors are computed from the finest FUN3D solution ($C_D = 0.06215$). The mesh sizes between FUN3D and VACC are similar, but not identical. The order of accuracy of the methods is tabulated in the final row. The SA wall function does not exhibit mesh convergence because the mesh is too coarse.

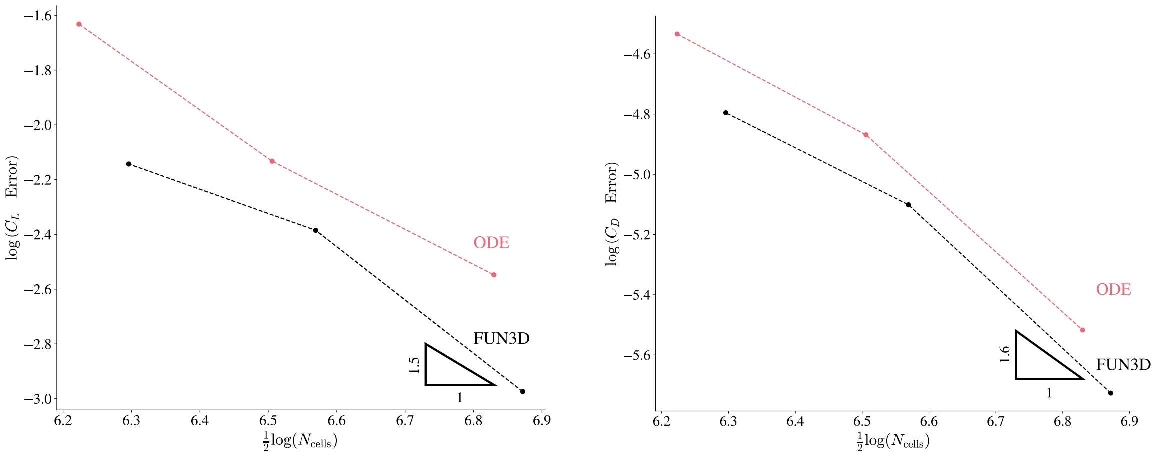
| **Mesh** | **FUN3D** | **SA** | **ODE** |
|---|---|---|---|
| Coarse | $8.27 \times 10^{-3}$ | $7.11 \times 10^{-3}$ | $1.07 \times 10^{-2}$ |
| Medium | $6.09 \times 10^{-3}$ | $7.72 \times 10^{-3}$ | $7.68 \times 10^{-3}$ |
| Fine | $3.26 \times 10^{-3}$ | $2.62 \times 10^{-2}$ | $4.02 \times 10^{-3}$ |
| **Order** | **1.62** | **N/A** | **1.63** |

similar issues as in the NACA 0012 case, particularly in the startup of the boundary layer compared to the ODE wall model. It is again clear that the SA wall function has not yet reached mesh convergence. Additional wall resolution is needed, particularly on the main and flap portions of the airfoil. Conversely, the ODE wall model is mesh converged on these coarse meshes ($y^+ > 1000$) is further corroborated by comparing FUN3D lift and drag convergence to the ODE wall model. All of the errors for both the SA and ODE wall functions along with the FUN3D results are tabulated in Tables 2 and 3. The ODE wall model exhibits similar mesh convergence to FUN3D for both lift and drag. The errors are slightly higher for a given number of cells (Figure 24). The FUN3D meshes are a carefully crafted family of body-fitted meshes curated to ensure low discretization errors. Conversely, the distribution of mesh points for the VACC solutions is not ideal. Output-based adaptive mesh refinement could generate an improved distribution of cells for VACC, but this was outside the scope of this work. Regardless, the ODE wall model exhibits good error convergence. The SA wall function is clearly not yet mesh converged. Once again, the ODE wall model exhibits superior accuracy on coarse meshes.

The complexity of this case requires a larger percentage of ODE source terms to be reduced compared to the NACA 0012. However, Table 4 indicates that the percentage of wall models with reduced source terms decreases as the mesh is refined. Figure 25 shows the locations of the $\beta$ reduction on the medium and fine meshes. Similar to the NACA 0012 case, the source terms are reduced around stagnation points. In the multielement airfoil case, the source terms are also reduced near sharp edges and in areas where

Table 4: Mesh refinement leads to a smaller percentage of ODE wall models reducing source term strength. The spatial area where the $\beta$ reduction is needed also shrinks. The complexity of this case requires more wall models to reduce their source strength compared to the NACA 0012.

| **Mesh** | # $\beta < 1$ | % $\beta < 1$ |
|---|---|---|
| Coarse | 1397 | 51.45% |
| Medium | 1945 | 35.85% |
| Fine | 1961 | 18.06% |




(a) Lift error

(b) Drag error

Figure 24: Lift and drag coefficient error convergences. The order of accuracy of each method was computed by performing a linear fit on the visualized data. The triangles represent the slope of the linear fit. The FUN3D lift convergence rate was 1.45, while VACC was 1.50. The FUN3D drag convergence rate was 1.62, while VACC was 1.63.

the flow is separated. The areas where $\beta < 1$ shrink as the mesh is refined.

This case demonstrates the proposed methodologies on the most complex 2D case available on the TMR. The flow contains multiple near-incompressible stagnation points and shear layer transfer between elements. The results are as accurate as those presented on the TMR and show variations with similar magnitude to those seen when using different body-fitted meshes. The solver and ODE wall model techniques worked straight out of the box without special considerations or meshing expertise. Smooth pressure and skin friction distributions were obtained despite irregular cut cells. The ODE demonstrates convergence in both the volume and on the body on meshes with $y^+$ values exceeding 1000. The mesh convergence rate is comparable to state-of-the-art body-fitted approaches.

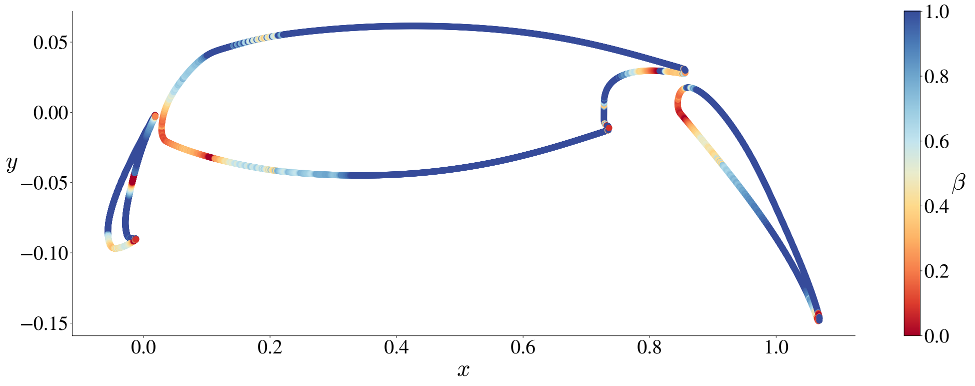
## 5. Conclusion

In this work, we improved our viscous Cartesian cut-cell solver from [14]. We made modifications to improve the interior forcing point approach with an ODE wall model developed by Berger and Aftosmis [20].

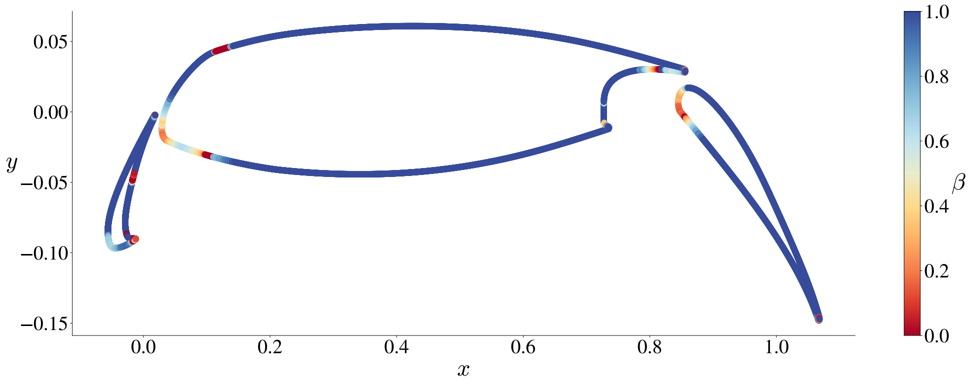
Firstly, we introduced a forcing-point interpolator for gradient quantities that favors lower-order averaging over exact reconstruction. Even in inviscid flows, gradient reconstructions have discontinuous truncation errors from the irregular stencils in cut cells. Smoothing these errors at interior forcing points is critical for viscous simulations, where gradient quantities are used directly in flux evaluations.

Secondly, we introduced a continuation strategy for the ODE wall models. We identified their stiff momentum source term as the main cause of stalled global convergence. To address this issue, we scale the stiff source terms in the ODE wall model with a continuation parameter, $\beta \in [0, 1]$, whenever a wall model fails to solve. Because the wall models continue to supply boundary layer approximations even when they cannot fully converge, they no longer stall the global solver.

We demonstrated these modifications on three increasingly complex two-dimensional cases from the NASA turbulence modeling resource [18]. The gradient interpolation controlled the cut-cell noise. The noise was worst on the turbulent bump, which has the largest noise-to-signal ratio. The two airfoils required continuation near stagnation points, but the fraction of wall models affected shrank under refinement in both cases. On the multielement airfoil, the most complex case, the ODE wall model




(a) Medium mesh




(b) Fine mesh

Figure 25: The final $\beta$ parameter values for the multielement airfoil case from equation 37. The x-axis is stretched about 5 times relative to the y-axis. $\beta$ is reduced around the stagnation point and other sharp edges where separation and reattachment are expected. Refining the mesh reduces the number of wall models where $\beta$ is reduced.

matched body-fitted FUN3D lift and drag convergence rates (1.50 versus 1.45 for lift, 1.63 versus 1.62 for drag) on meshes with $y^+ > 1000$, while the SA wall function did not reach mesh convergence.

Future work will be focused in two areas. The cell distribution for the cut-cell solver is not yet ideal, and output-based adaptive refinement could close the remaining error gap with the curated body-fitted meshes. More significantly, every case here is two-dimensional. Extending the interior forcing point method to three dimensions is the natural next step toward automated RANS simulations on the arbitrarily-complex geometries that motivate Cartesian cut-cell methods.

## References

[1] Antony Jameson, Luigi Martinelli, and Niles A. Pierce. Optimum aerodynamic design using the Navier–Stokes equations. *Theoretical and Computational Fluid Dynamics*, 10(1–4):213–237, 1998.

[2] Eric J. Nielsen and W. Kyle Anderson. Aerodynamic design optimization on unstructured meshes using the Navier–Stokes equations. *AIAA Journal*, 37(11):1411–1419, 1999.

[3] Joaquim R. R. A. Martins. Perspectives on aerodynamic design optimization. In *AIAA SciTech Forum*, Orlando, FL, January 2020.

[4] Michael J. Aftosmis. Lecture notes in solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries. *VKI Lecture Series*, 1997-02, 1997.

[5] Hiroaki Nishikawa. Improved wall-normal derivative formulae for anisotropic adaptive simplex-element grids. *arXiv preprint arXiv:2101.11475*, 2021.

[6] Z. J. Wang. A quadtree-based adaptive cartesian/quad grid flow solver for navier-stokes equations. *Computers & Fluids*, 27(4):529–549, 1998.

[7] Michel Delanaye, Michael Aftosmis, Marsha Berger, Yen Liu, and Thomas Pulliam. Automatic hybrid-Cartesian grid generation for high-reynolds number flows around complex geometries. In *37th Aerospace Sciences Meeting and Exhibit*, page 777, 1999.

[8] Z. J. Wang and R. F. Chen. Anisotropic solution-adaptive viscous Cartesian grid method for turbulent flow simulation. *AIAA Journal*, 40(10):1969–1978, 2002.

[9] William Dawes, Simon Harvey, S. Fellows, C. Favaretto, and Aditya Velivelli. Viscous layer meshes from level sets on Cartesian meshes. In *45th AIAA Aerospace Sciences Meeting and Exhibit*, page 555, 2007.

[10] Yoshiharu Tamaki, Motoshi Harada, and Taro Imamura. Near-wall modification of Spalart–Allmaras turbulence model for immersed boundary method. *AIAA journal*, 55(9):3027–3039, 2017.

[11] Carmen-Ioana Ursachi, Steven R Allmaras, David L Darmofal, and Marshall C Galbraith. Stress-equivalent Spalart–Allmaras wall model with local boundary conditions for Reynolds-averaged Navier–Stokes. *AIAA Journal*, pages 1–17, 2024.

[12] Francesco Capizzano. Turbulent wall model for immersed boundary methods. *AIAA Journal*, 49(11):2367–2381, 2011.

[13] Marsha Berger and Michael Aftosmis. Progress towards a Cartesian cut-cell method for viscous compressible flow. In *50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, page 1301, 2012.

[14] Alex Kleb, Krzysztof J. Fidkowski, and Joaquim R. R. A. Martins. Solving high Reynolds number flows on Cartesian cut-cell meshes using a Jacobian-free Newton–Krylov method. In *Proceedings of the 12th International Conference on Computational Fluid Dynamics (ICCFD12)*, July 2024.

[15] Alexandre J. Favre. The equations of compressible turbulent gases. Technical report, Universite D'Aix Marseille, 1965.

[16] Philippe Spalart and Steven Allmaras. A One-Equation Turbulence Model for Aerodynamic Flows. *La Recherche Aerospatiale*, 1:5–21, 1994.

[17] Steven R. Allmaras, Forrester T. Johnson, and Philippe R. Spalart. Modifications and Clarifications for the Implementation of the Spalart-Allmaras Turbulence Model. Big Island, Hawaii, July 2012.

[18] Christopher Rumsey. NASA turbulence modeling resource, 2019.

[19] Marco Ceze and Krzysztof J Fidkowski. Drag prediction using adaptive discontinuous finite elements. *Journal of Aircraft*, 51(4):1284–1294, 2014.

[20] Marsha J Berger and Michael J Aftosmis. An ODE-based wall model for turbulent flow simulations. *AIAA Journal*, 56(2):700–714, 2018.

[21] Alex Kleb, Krzysztof J. Fidkowski, and Joaquim R. R. A. Martins. Solving high Reynolds number flows on Cartesian cut-cell meshes using an ODE wall function with momentum balance. *Computers & Fluids*, 304:106882, 2026.

[22] Marian Nemec and Michael J. Aftosmis. Toward automatic verification of goal-oriented flow simulations. NASA/TM 2014-218386, National Aeronautics and Space Administration, Center for AeroSpace Information, 7115 Standard Drive, Hanover, MD 21076, 2014.

[23] Marian Nemec, Michael J Aftosmis, and Liam Smith. Uncertainty estimates for sonic-boom pressure signatures and loudness carpets. In *AIAA Aviation 2021 Forum*, page 2753, 2021.

[24] Yingbin Zhang, Honglei Bai, Chunguang Xu, Yang Yang, Aihua Wang, and Huaibao Zhang. A Cartesian grid based non-equilibrium wall model for RANS simulation of high reynolds number turbulent flows. *Aerospace Science and Technology*, 176:112116, 2026.

[25] Masayuki Yano and David L Darmofal. An optimization-based framework for anisotropic simplex mesh adaptation. *Journal of Computational Physics*, 231(22):7626–7649, 2012.

[26] S. L. Krist, R. T. Biedron, and C. L Rumsey. CFL3D user's manual (version 5.0). NASA/TM 1998-208444, 1998.

[27] Robert T. Biedron, Jan-Reneé Carlson, Joseph M. Derlaga, Peter A. Gnoffo, Dana P. Hammond, Kevin E. Jacobson, William T. Jones, Bil Kleb, Elizabeth M. Lee-Rausch, Eric J. Nielsen, Michael A.

Park, Christopher L. Rumsey, James L. Thomas, Kyle B. Thompson, Aaron C. Walden, Li Wang, and William A. Wood. FUN3D manual: 13.7. Technical Report NASA/TM–20205010139, NASA Langley Research Center, Hampton, Virginia, November 2020.

[28] Mike Park, Beth Lee-Rausch, and Christopher Rumsey. FUN3D and CFL3D computations for the first high lift prediction workshop. In *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, page 936, 2011.