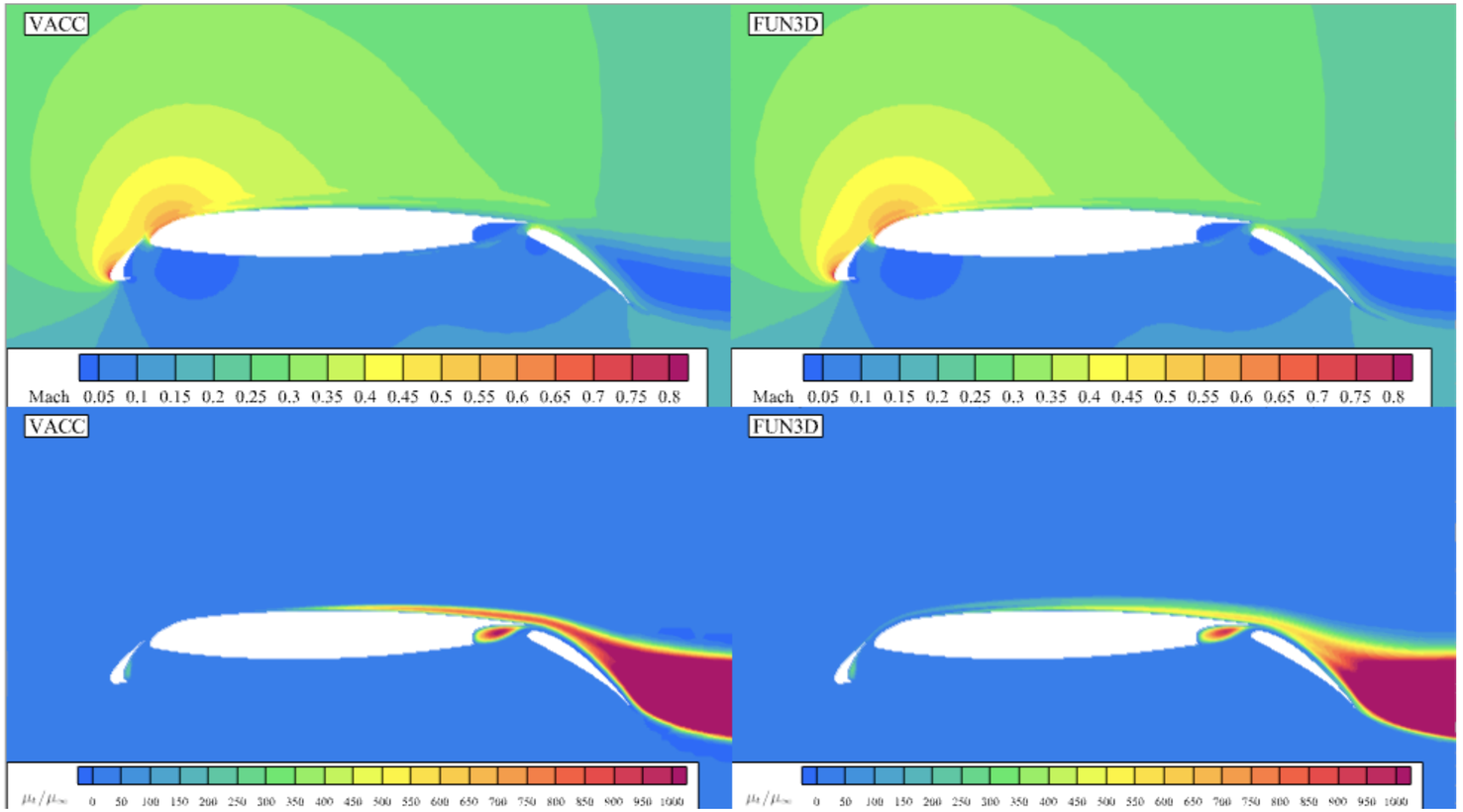
Cartesian Cut-cell Methods for High Reynolds Number Flows

Alex Kleb

Advanced Modeling & Simulation (AMS) Seminar Series

NASA Ames Research Center
February 5th, 2026



Copyright © by Alex Kleb

AEROSPACE ENGINEERING
UNIVERSITY of MICHIGAN ■ COLLEGE of ENGINEERING

Computational Fluid Dynamics Group

MDO Lab

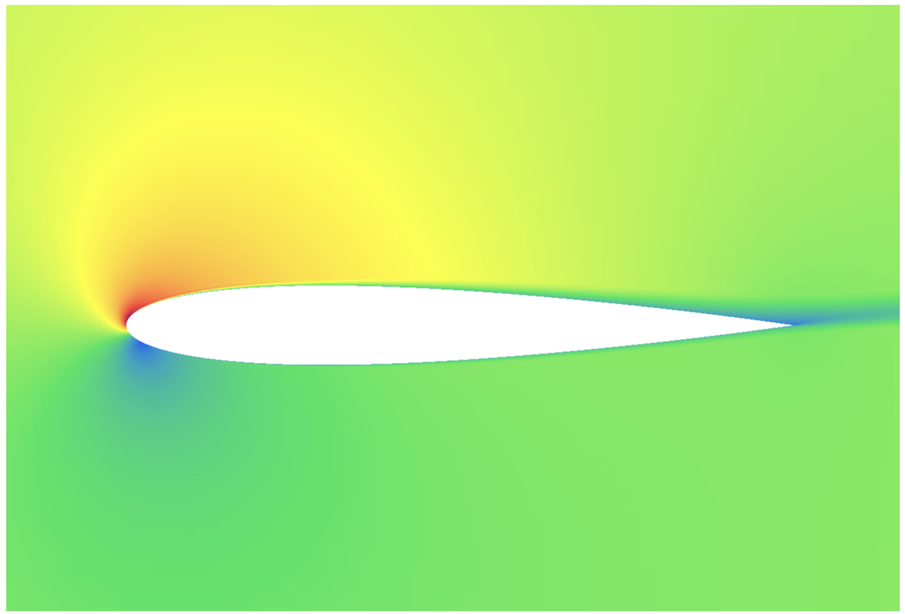
Outline

- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance function
- ODE wall model
- NASA TMR test cases
- Conclusions

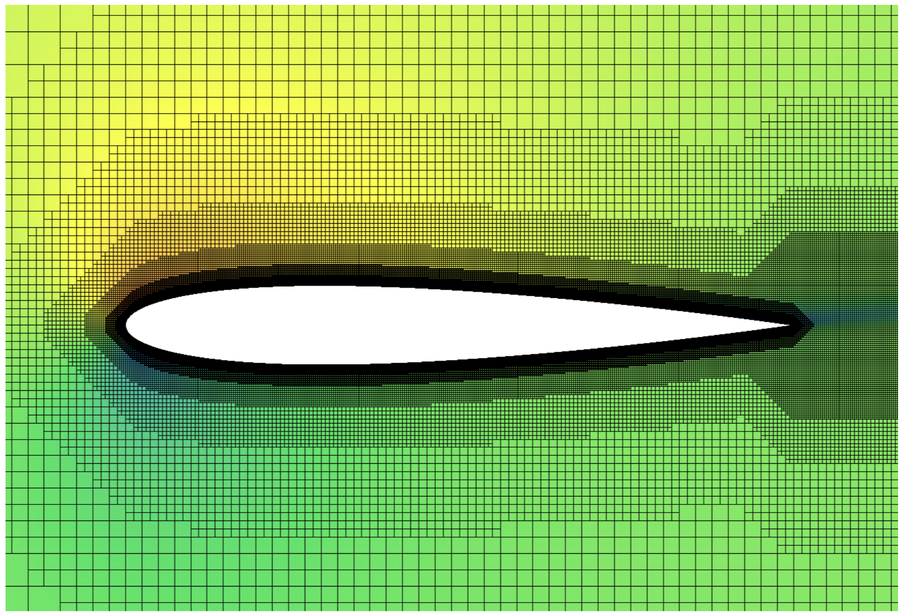
Outline

- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance dunction
- ODE wall model
- NASA TMR test cases
- Conclusions

CFD discretizes and solves PDEs governing fluid flow

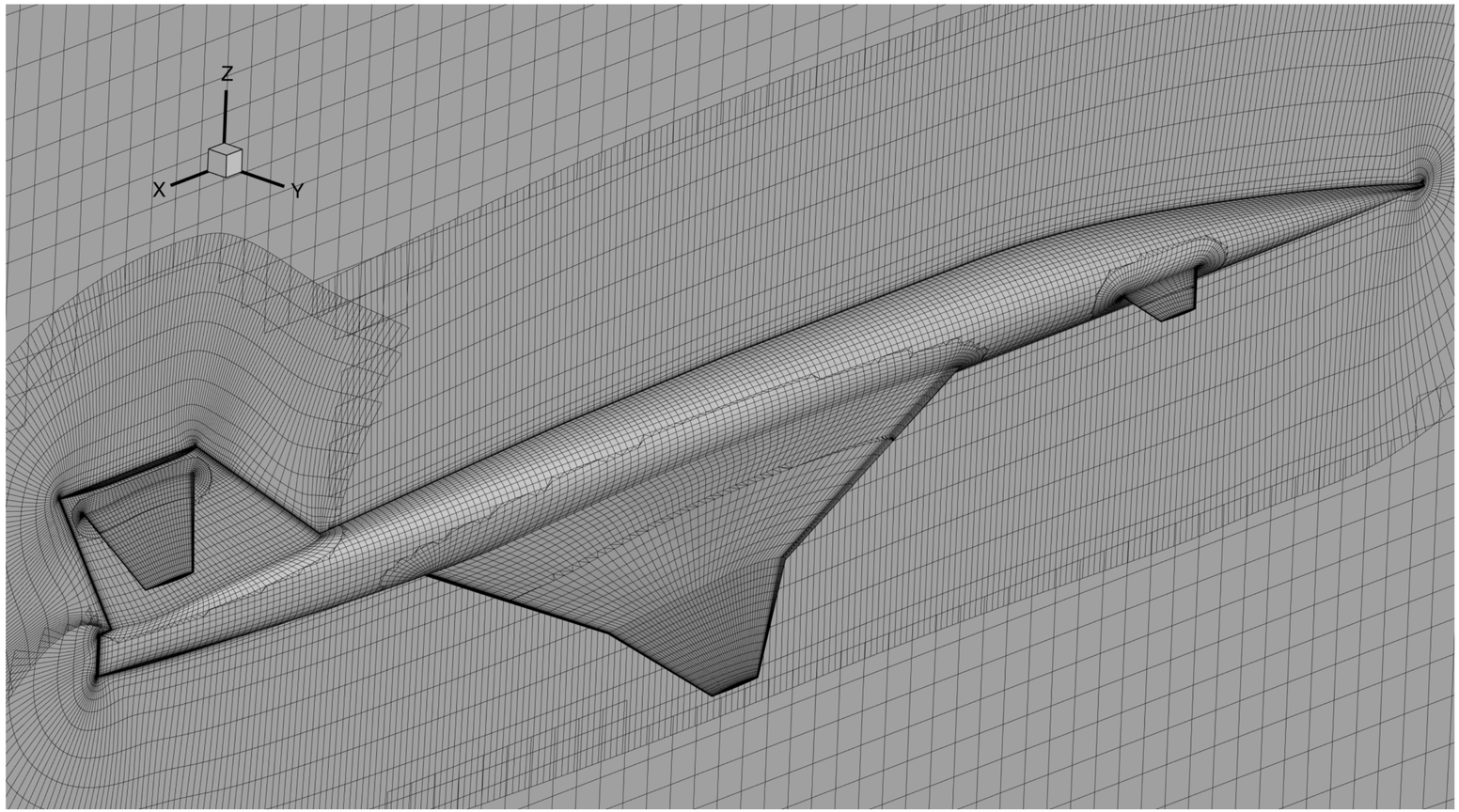


The continuous solution $\mathbf{u}(\vec{x}, t)$
is a function of time and space



The discrete solution $\mathbf{U} \in \mathbb{R}^N$
is a state vector

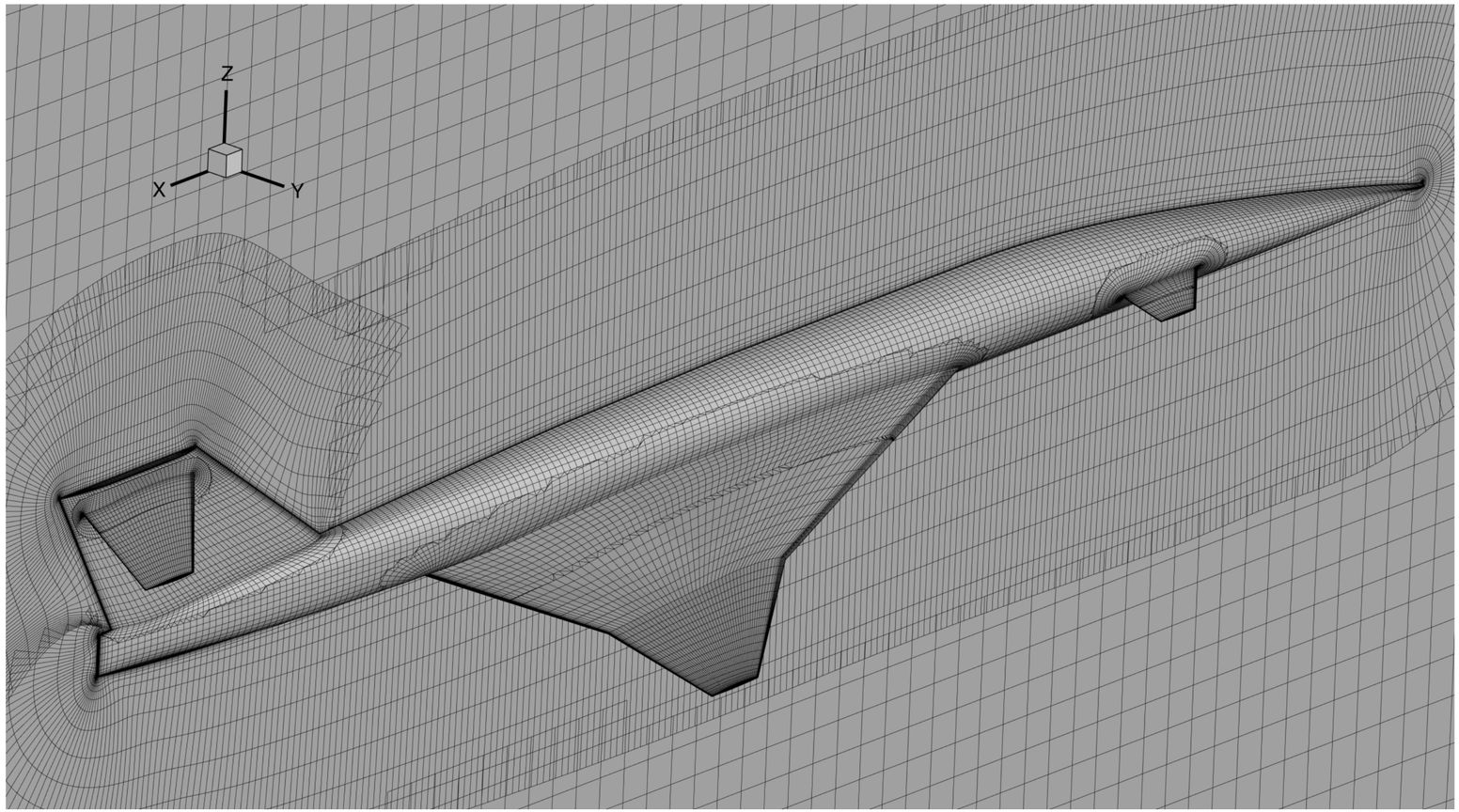
Generating accurate CFD meshes can be challenging



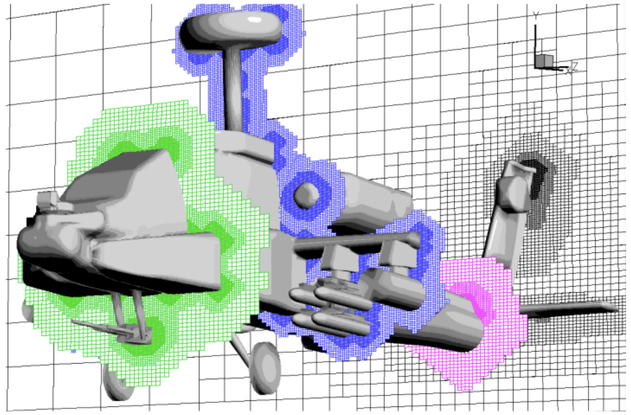
Generated in weeks by a meshing expert

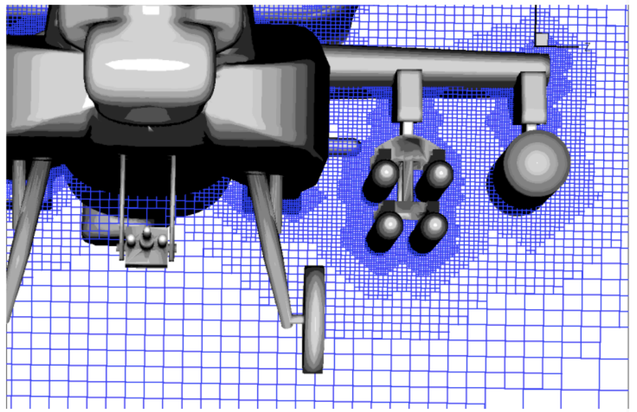
Seraj et al. “Aerodynamic shape optimization of a supersonic transport including a subsonic static margin constraint.” Aerospace Science and Technology, 2025.

Cartesian cut-cell meshes automate the meshing process



Generated in weeks by a meshing expert



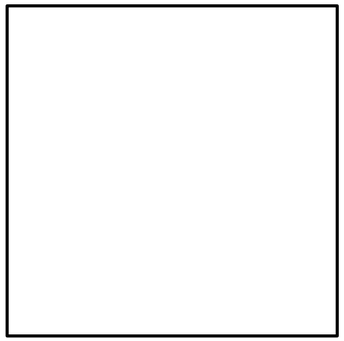


Generated in minutes
with no user intervention

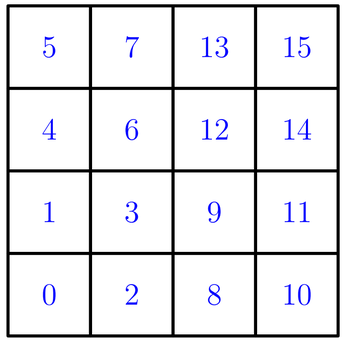
Seraj et al. “Aerodynamic shape optimization of a supersonic transport including a subsonic static margin constraint.” Aerospace Science and Technology, 2025.
Aftosmis “Solution adaptive Cartesian grid methods for aerodynamic flows with complex geometries.” von Karman Institute Lecture Series, 1997.

Cartesian cut-cell mesh generation is automatic

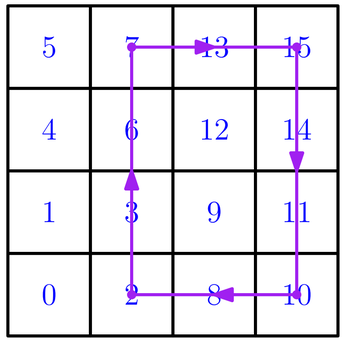
1.



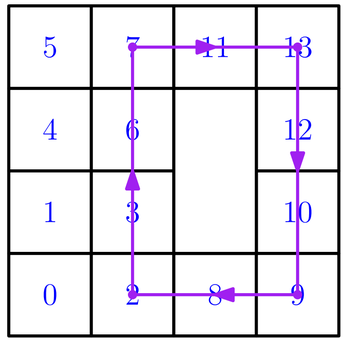
2.



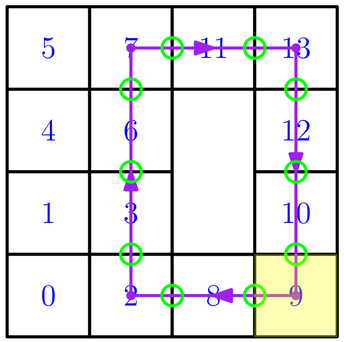
3.



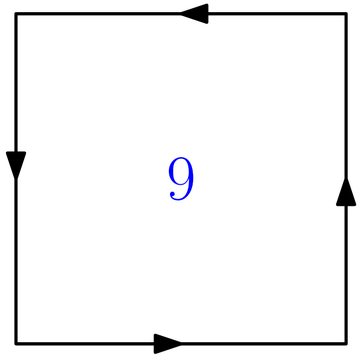
4.



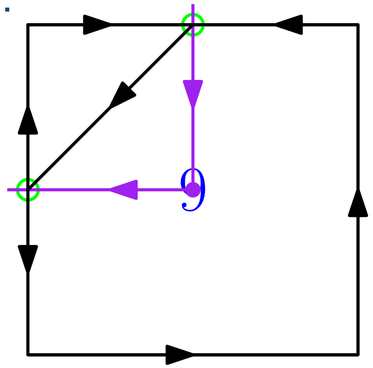
5.



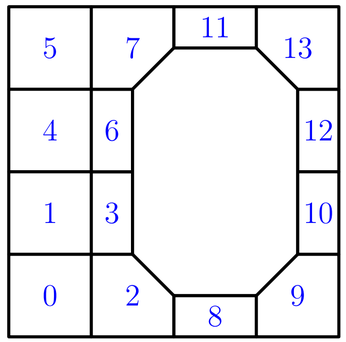
6.



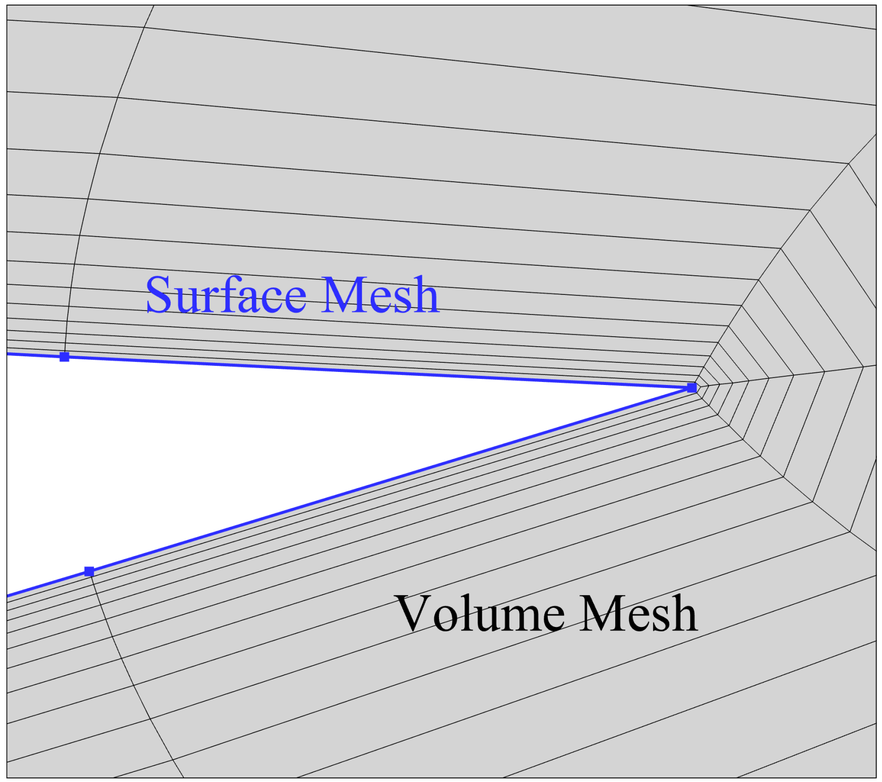
7.



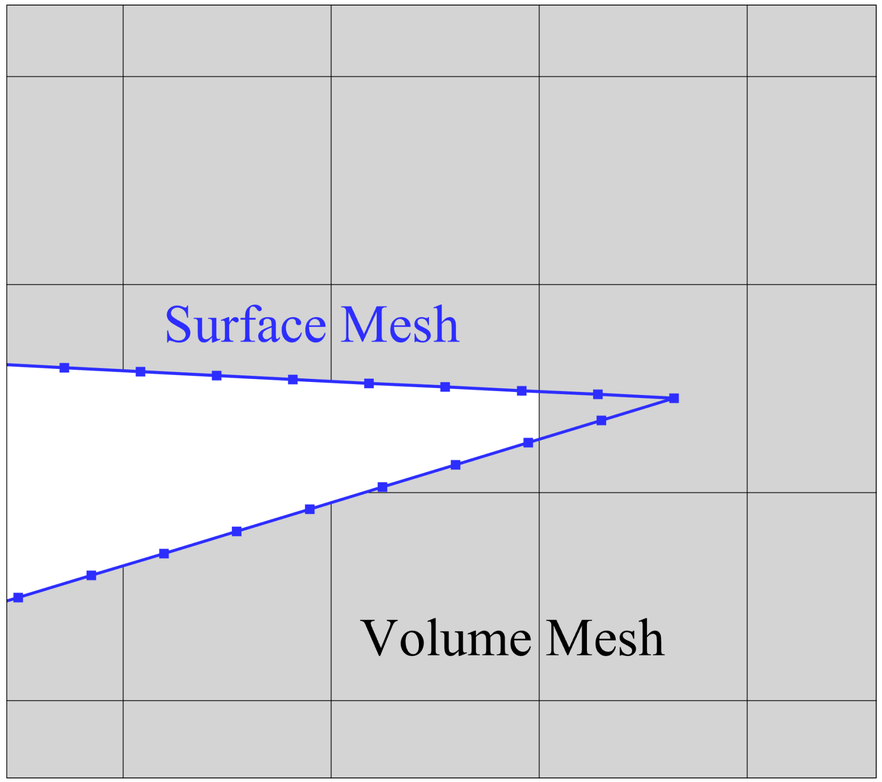
8.



Cut-cell methods decouple surface and volume meshes

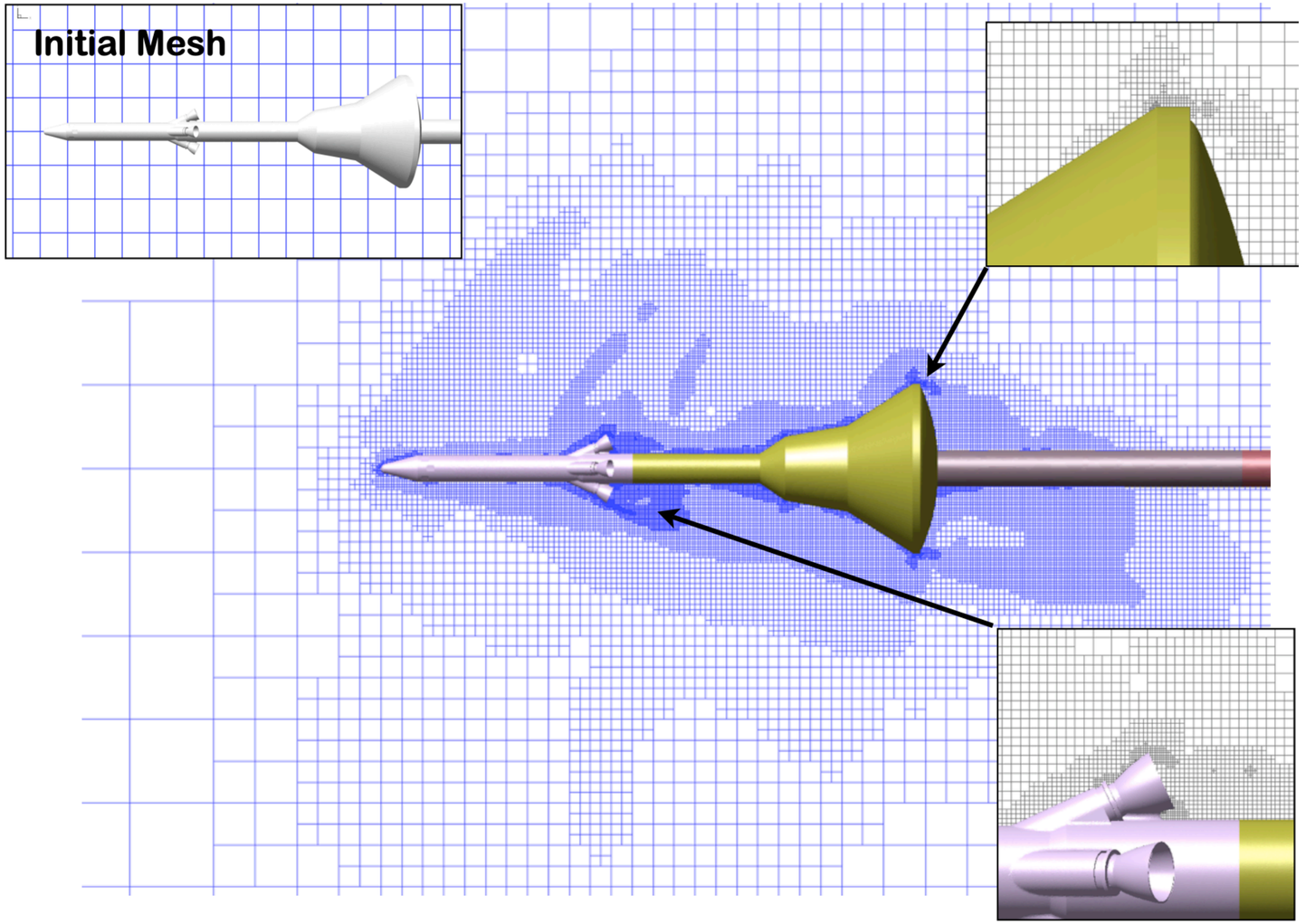


Coupled



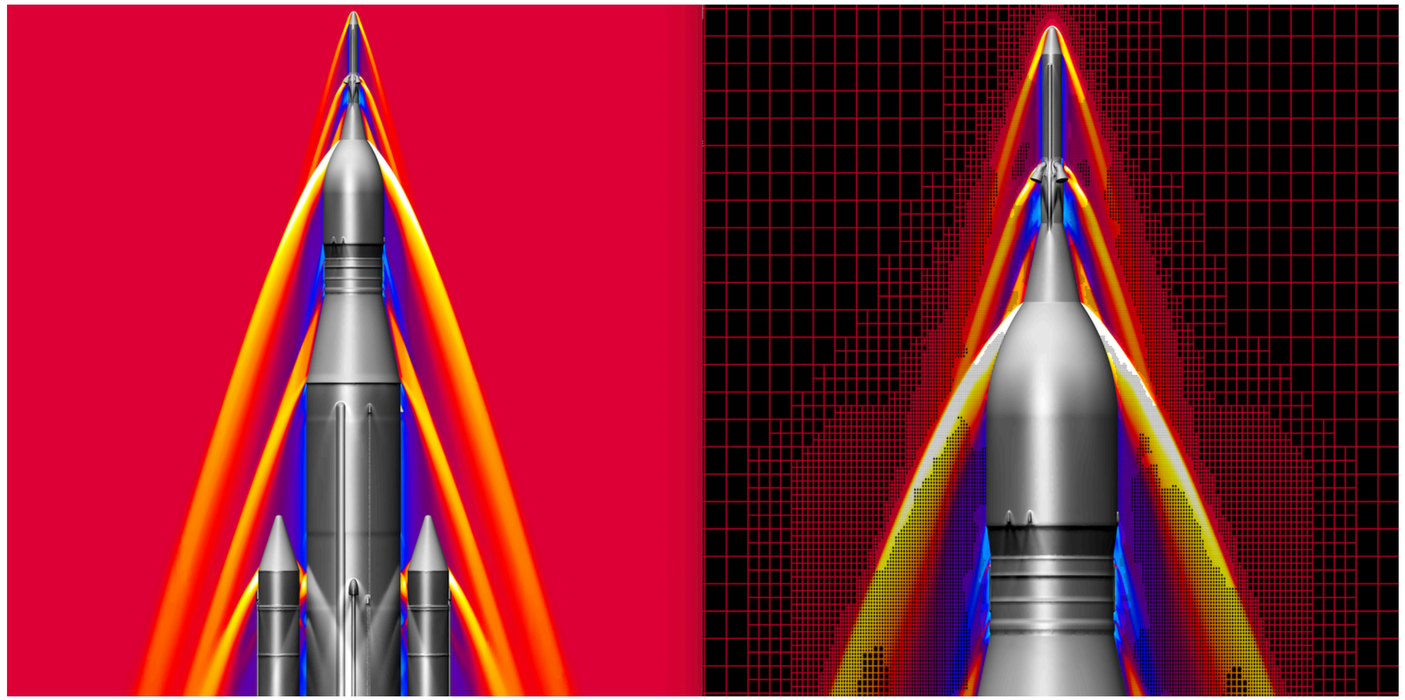
Decoupled

Adaptive mesh refinement ensures accuracy



Nemec et al. “Adjoint-based adaptive mesh refinement for complex geometries.” 46th AIAA Aerospace Sciences Meeting, 2008.

Cut-cell methods are great at solving inviscid flows

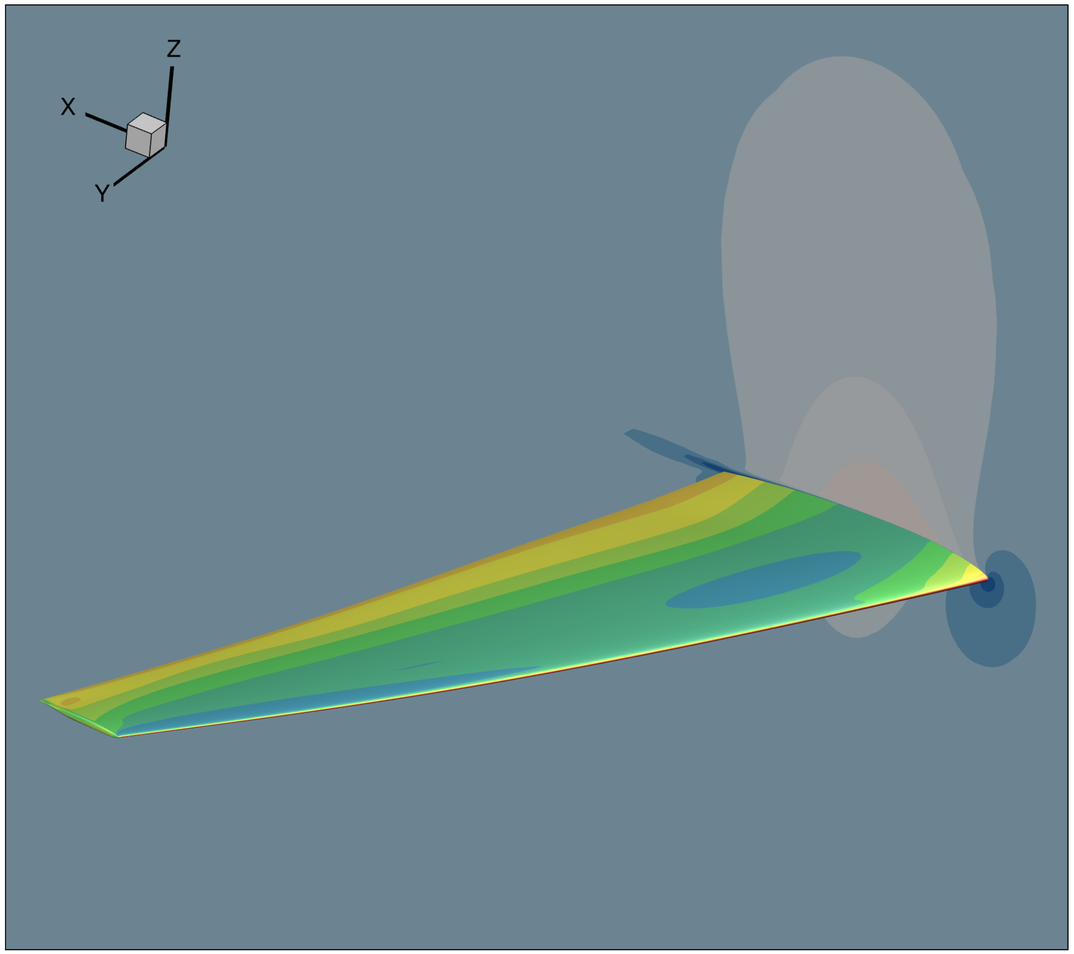






All images from the Cart3D website: https://www.nas.nasa.gov/publications/software/docs/cart3d/pages/cart3d_images.html

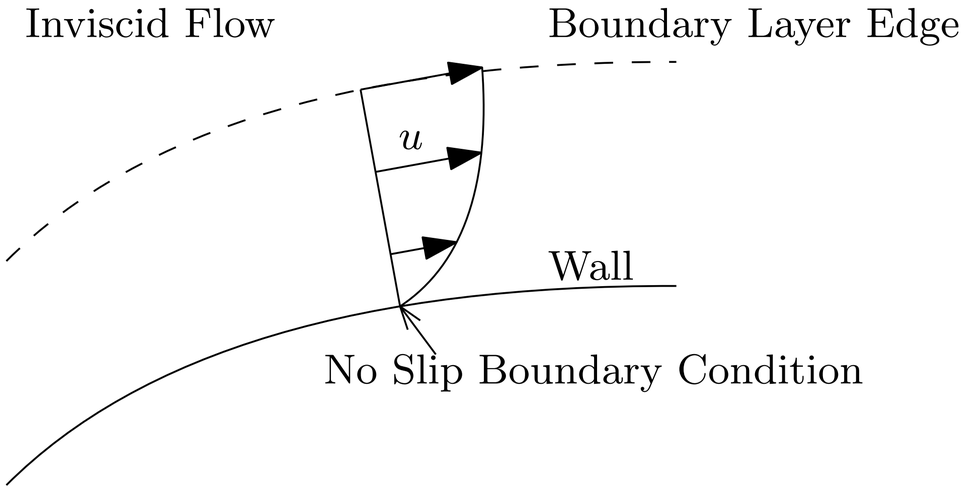
Viscous effects are important for commercial aircraft

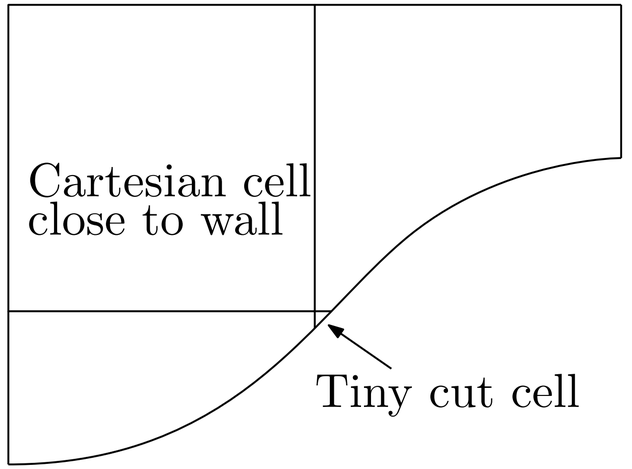


~50% of the total drag is skin friction

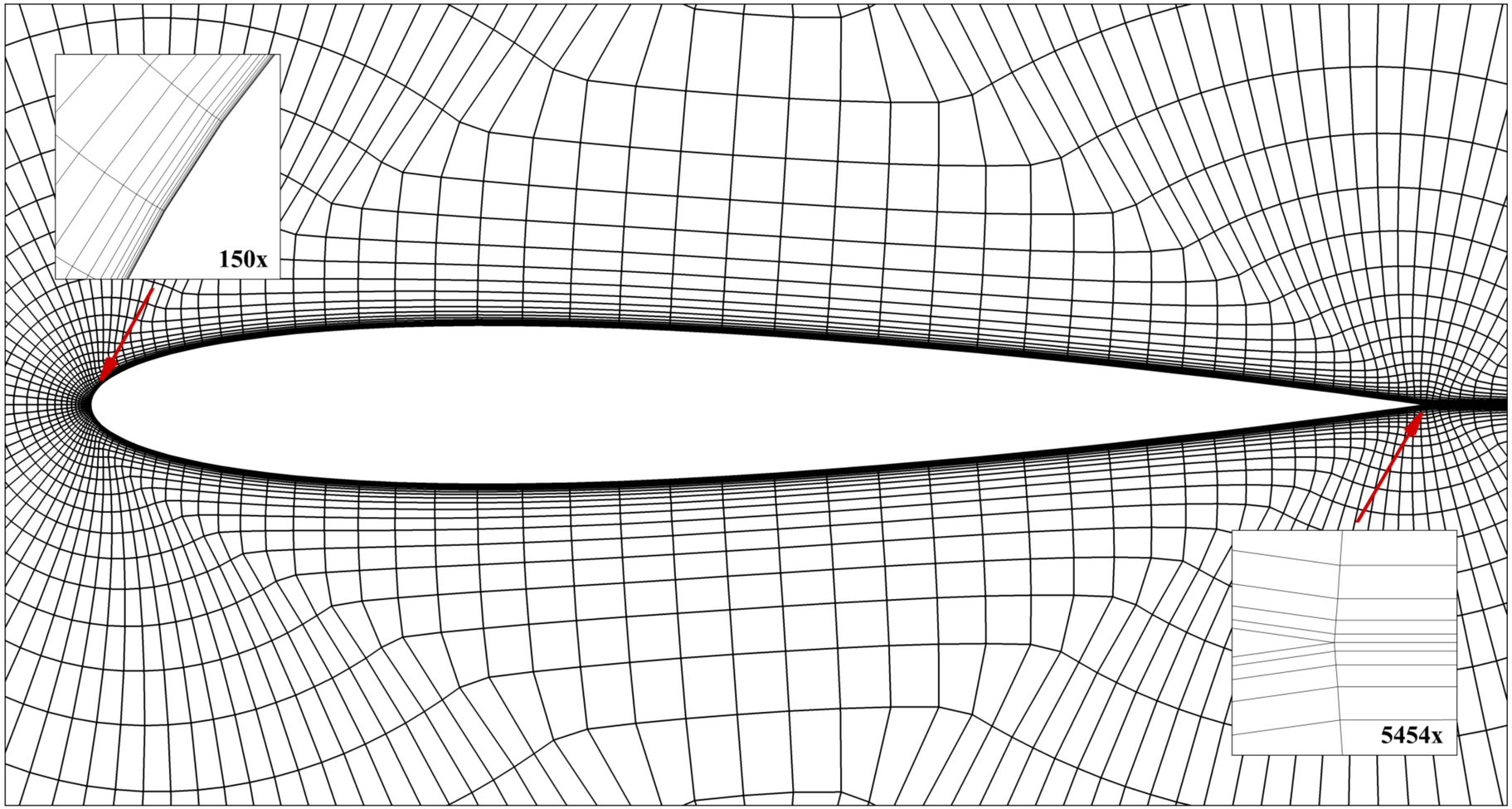
Cartesian cut-cell methods struggle with viscous flows

Irregular cells sizes near the wall introduce noise





Anisotropic cells cannot be generated without sacrificing robustness

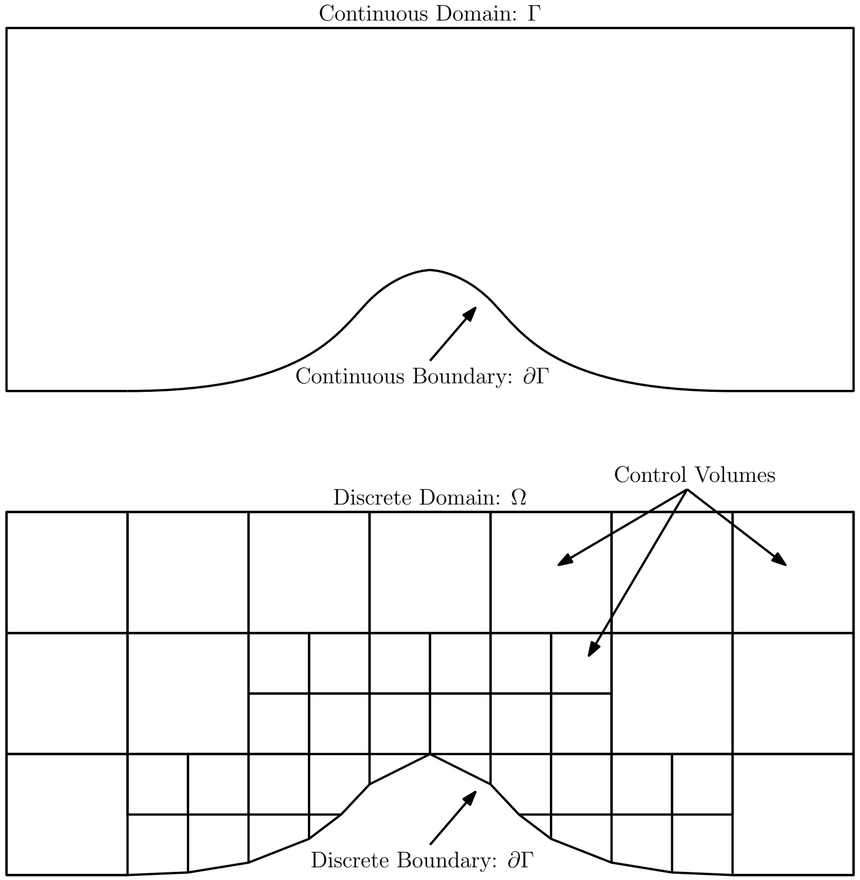


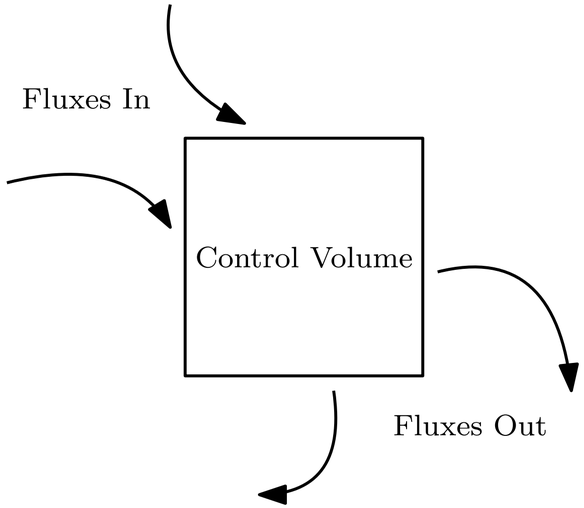
The goal of this work is to address these issues to enable viscous flow solutions with Cartesian cut-cell methods.

Outline

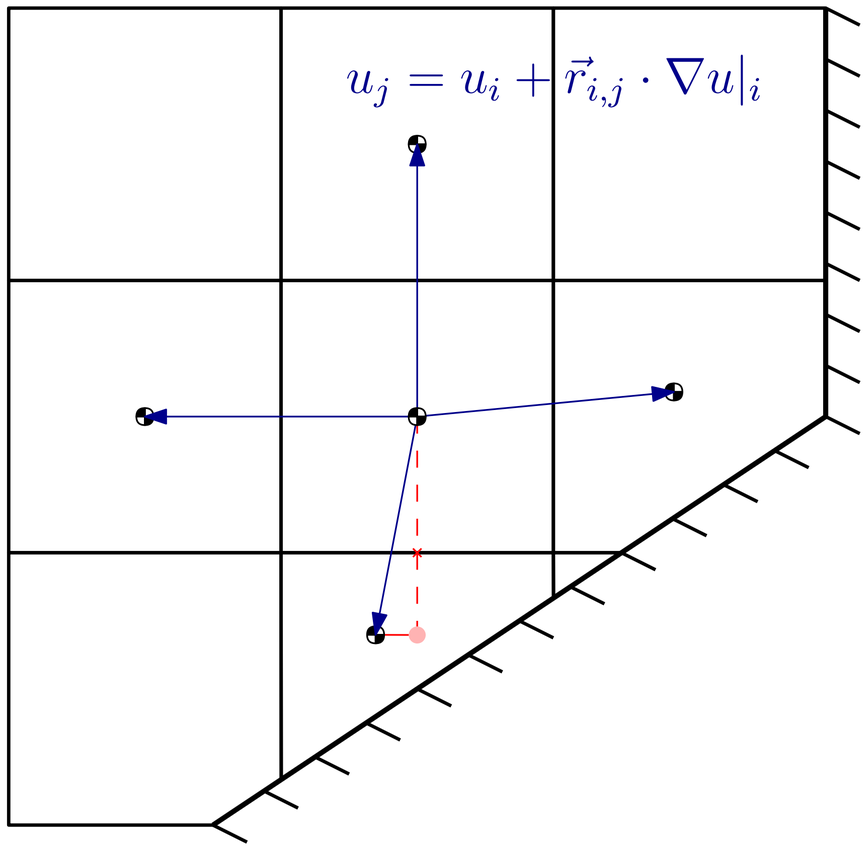
- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance function
- ODE wall model
- NASA TMR test cases
- Conclusions

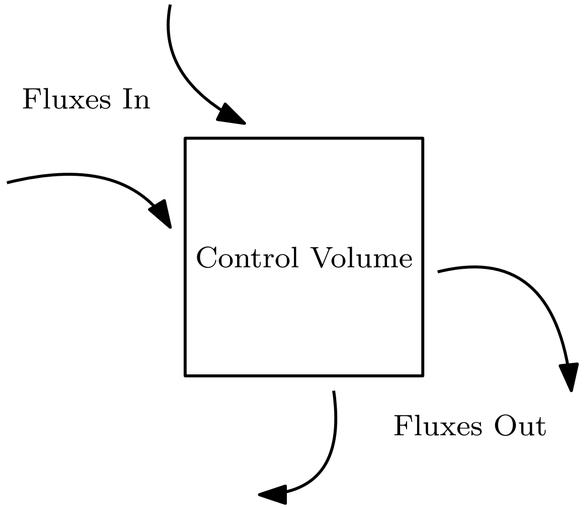
VACC is a second-order 2D finite volume flow solver



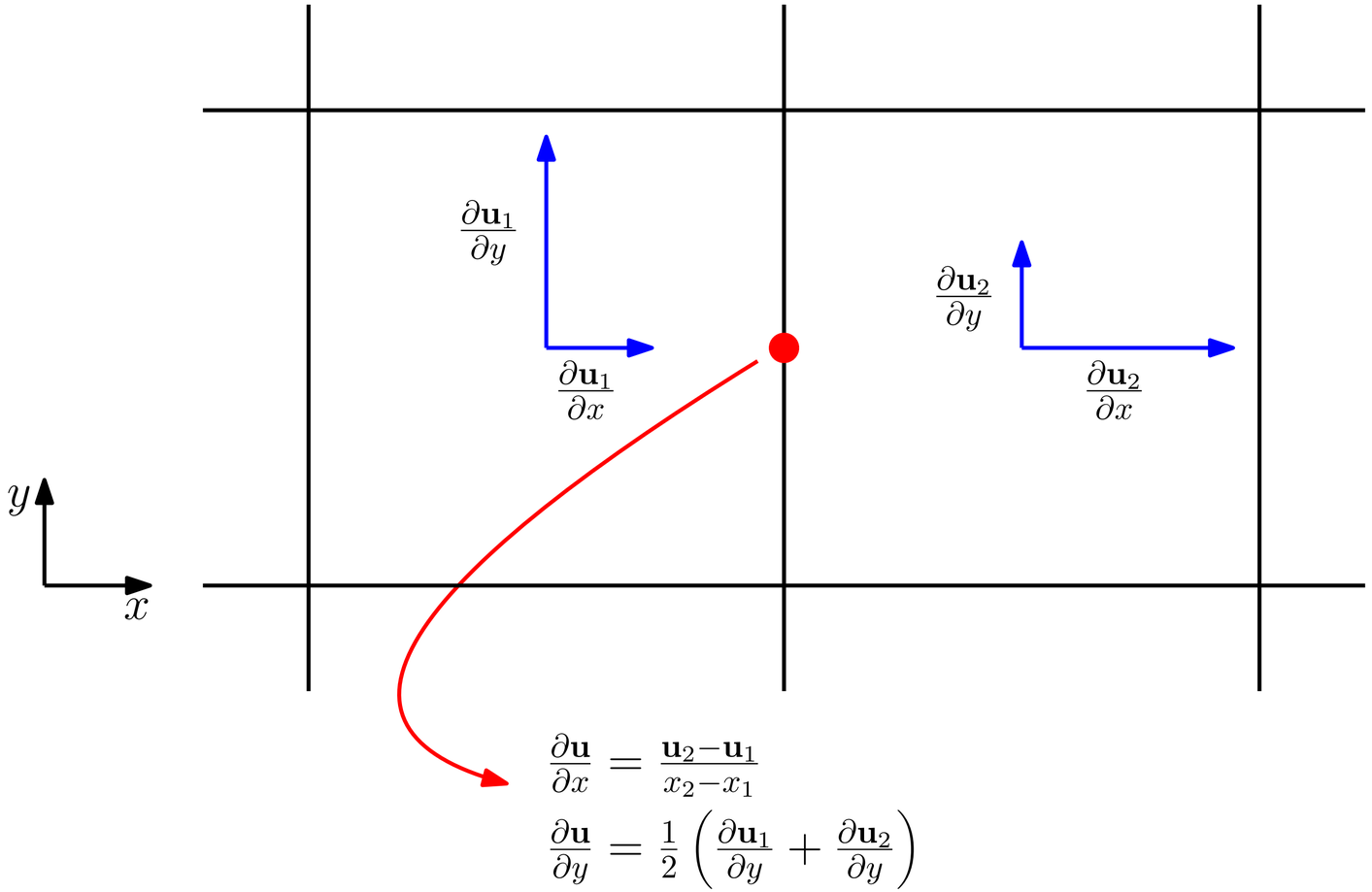


Inviscid fluxes require gradients at cell centers



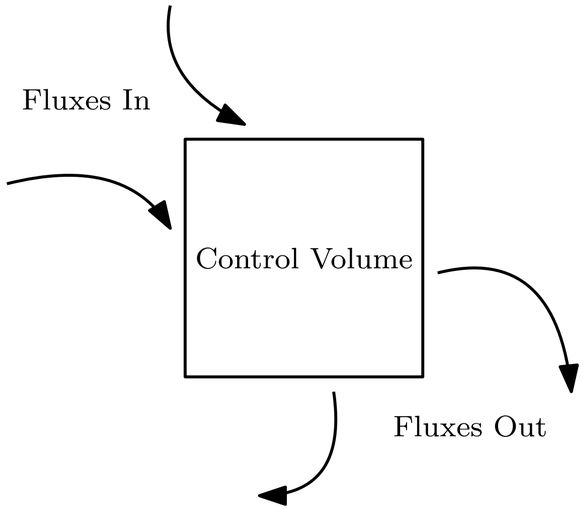


Viscous fluxes require gradients at face centroids



$$\frac{\partial \mathbf{u}}{\partial x} = \frac{\mathbf{u}_2 - \mathbf{u}_1}{x_2 - x_1}$$

$$\frac{\partial \mathbf{u}}{\partial y} = \frac{1}{2}\left(\frac{\partial \mathbf{u}_1}{\partial y} + \frac{\partial \mathbf{u}_2}{\partial y}\right)$$



Fluxes are defined by the Reynolds-Averaged Navier—Stokes Equations

$$\partial_j(\rho v_j) = 0$$ → Conservation of mass

$$\partial_j(\rho v_j v_i + \delta_{ij} p) - \partial_j(\tau_{ij}^{\text{RANS}}) = 0$$ → Conservation of momentum

$$\partial_j(\rho H v_j) - \partial_j(v_i \tau_{ij}^{\text{RANS}} - q_i^{\text{RANS}}) = 0$$ → Conservation of energy

Inviscid Fluxes

Viscous Fluxes

Spalart—Allmaras turbulence model for closure

$$\partial_j(\rho u_j \tilde{\nu}_n) - \partial_j\left(\frac{1}{\sigma}\rho(\nu+\nu')\partial_j\tilde{\nu}_n\right) = -\frac{1}{\sigma}(\nu+\nu')\partial_j\rho\partial_j\tilde{\nu}_n + \sqrt{\tilde{\nu}_\infty}\frac{c_{b2}\rho}{\sigma}\partial_j\tilde{\nu}_n\partial_j\tilde{\nu}_n + \frac{P-D}{\sqrt{\tilde{\nu}_\infty}}$$

Inviscid Flux

Viscous Flux

Source Terms

Spalart—Allmaras turbulence model requires a distance function

$$\partial_j(\rho u_j \tilde{\nu}_n) - \partial_j\left(\frac{1}{\sigma}\rho(\nu+\nu')\partial_j\tilde{\nu}_n\right) = -\frac{1}{\sigma}(\nu+\nu')\partial_j\rho\partial_j\tilde{\nu}_n + \sqrt{\tilde{\nu}_\infty}\frac{c_{b2}\rho}{\sigma}\partial_j\tilde{\nu}_n\partial_j\tilde{\nu}_n + \frac{P-D}{\sqrt{\tilde{\nu}_\infty}}$$

Inviscid Flux

Viscous Flux

Source Terms

Destruction Source Term $D = c_{w1}f_w\frac{\rho\tilde{\nu}^2}{\bar{d}^2}$

Spalart—Allmaras turbulence model requires a distance function

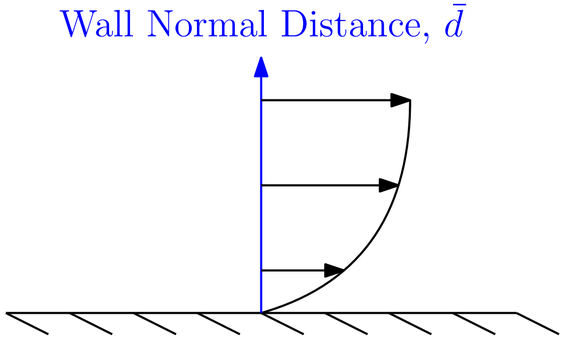
$$\partial_j(\rho u_j \tilde{\nu}_n) - \partial_j\left(\frac{1}{\sigma}\rho(\nu+\nu')\partial_j\tilde{\nu}_n\right) = -\frac{1}{\sigma}(\nu+\nu')\partial_j\rho\partial_j\tilde{\nu}_n + \sqrt{\tilde{\nu}_\infty}\frac{c_{b2}\rho}{\sigma}\partial_j\tilde{\nu}_n\partial_j\tilde{\nu}_n + \frac{P-D}{\sqrt{\tilde{\nu}_\infty}}$$

Inviscid Flux

Viscous Flux

Source Terms

Destruction Source Term $D = c_{w1}f_w\frac{\rho\tilde{\nu}^2}{\bar{d}^2}$



Spalart—Allmaras turbulence model requires a distance function

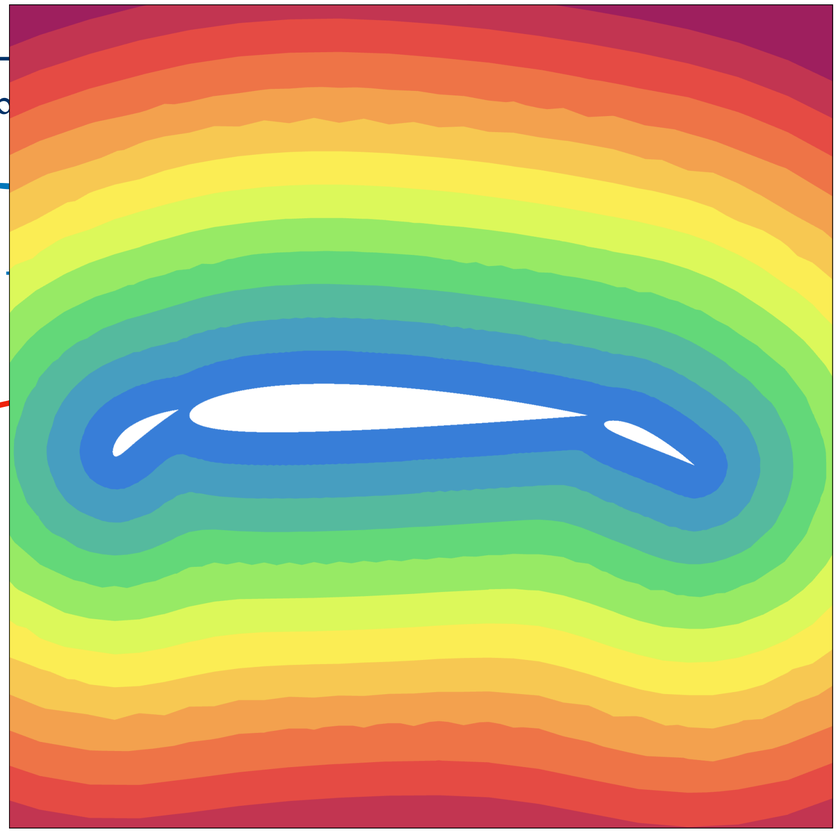
$$\partial_j(\rho u_j \tilde{\nu}_n) - \partial_j\left(\frac{1}{\sigma}\rho(\nu+\nu')\partial_j\tilde{\nu}_n\right) = -\frac{1}{\sigma}(\nu+\nu')\partial_j\rho\partial_j\tilde{\nu}_n + \sqrt{\tilde{\nu}_\infty}$$

Inviscid Flux

Viscous Flux

Source T

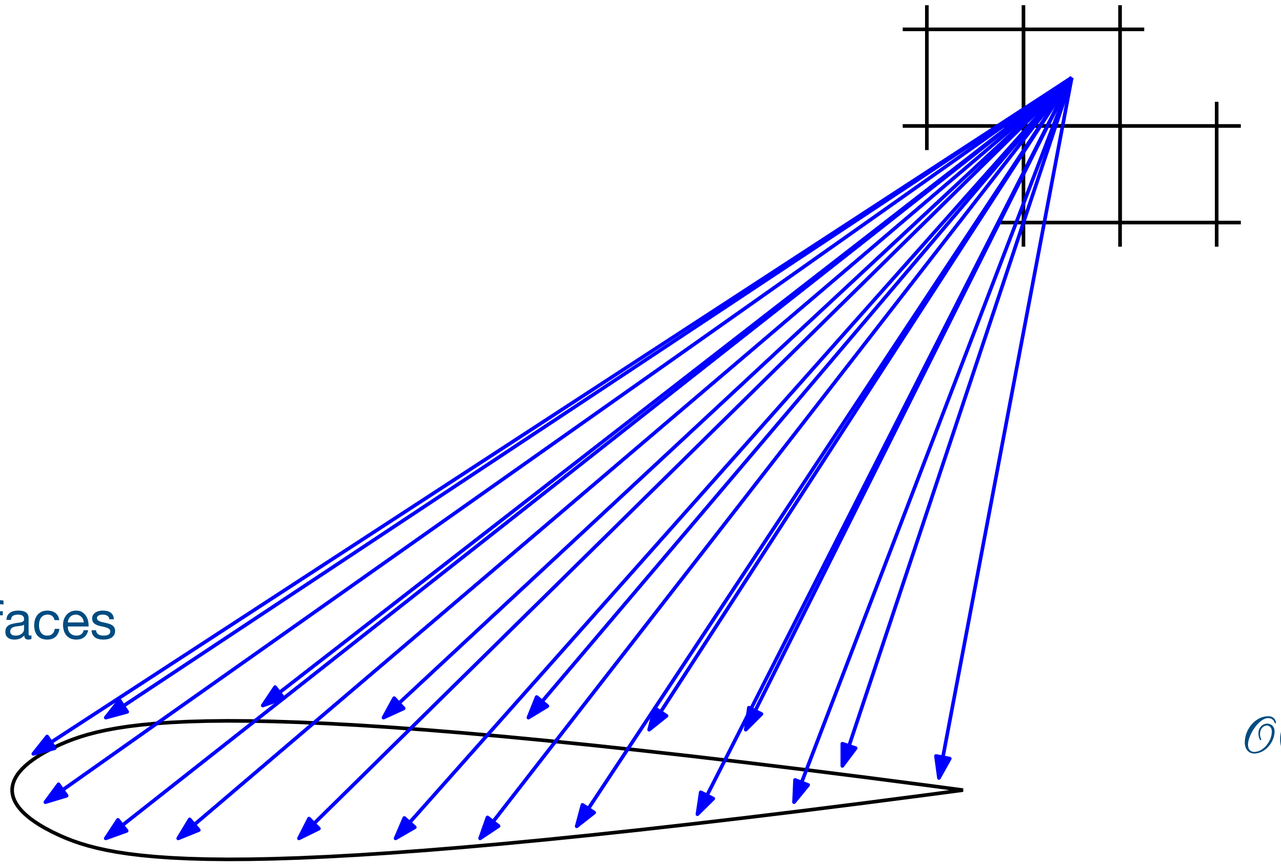
Destruction Source Term $D = c_{w1} f_w \frac{\rho\tilde{\nu}^2}{\bar{d}^2}$



Outline

- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance function
- ODE wall model
- NASA TMR test cases
- Conclusions

Naive distance function computes the distance in every cell exactly

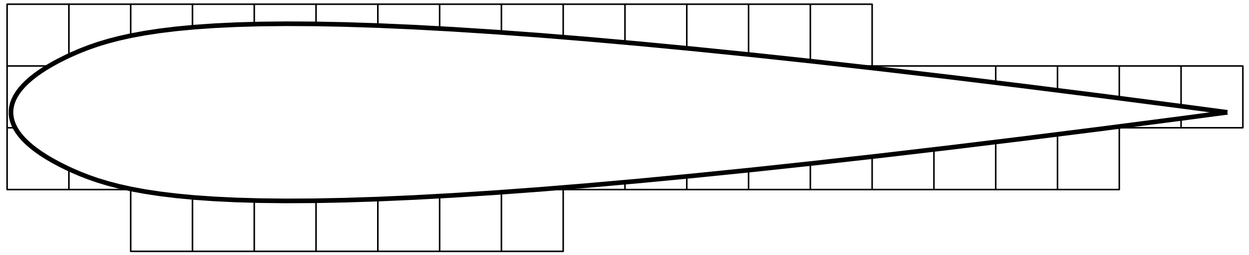


$\mathcal{O}(N)$ cells

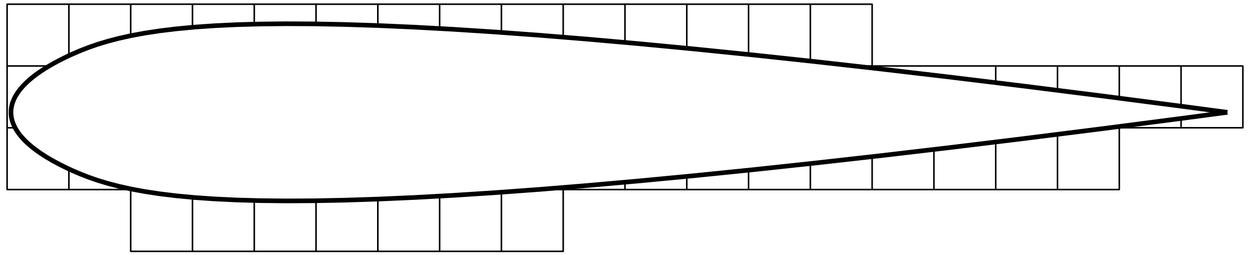
$\mathcal{O}(M)$ geometry surfaces

$\mathcal{O}(NM)$ time complexity

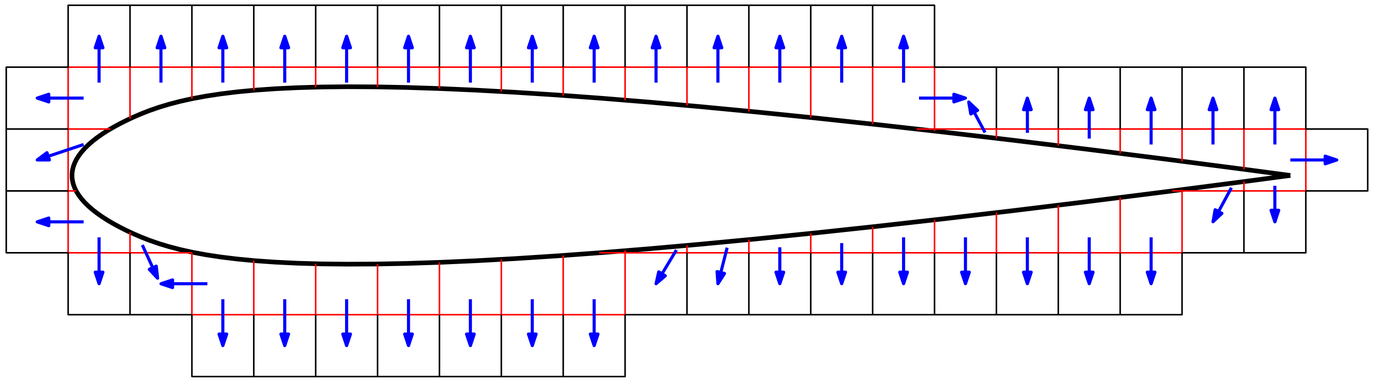
Traveling out from the surface is efficient



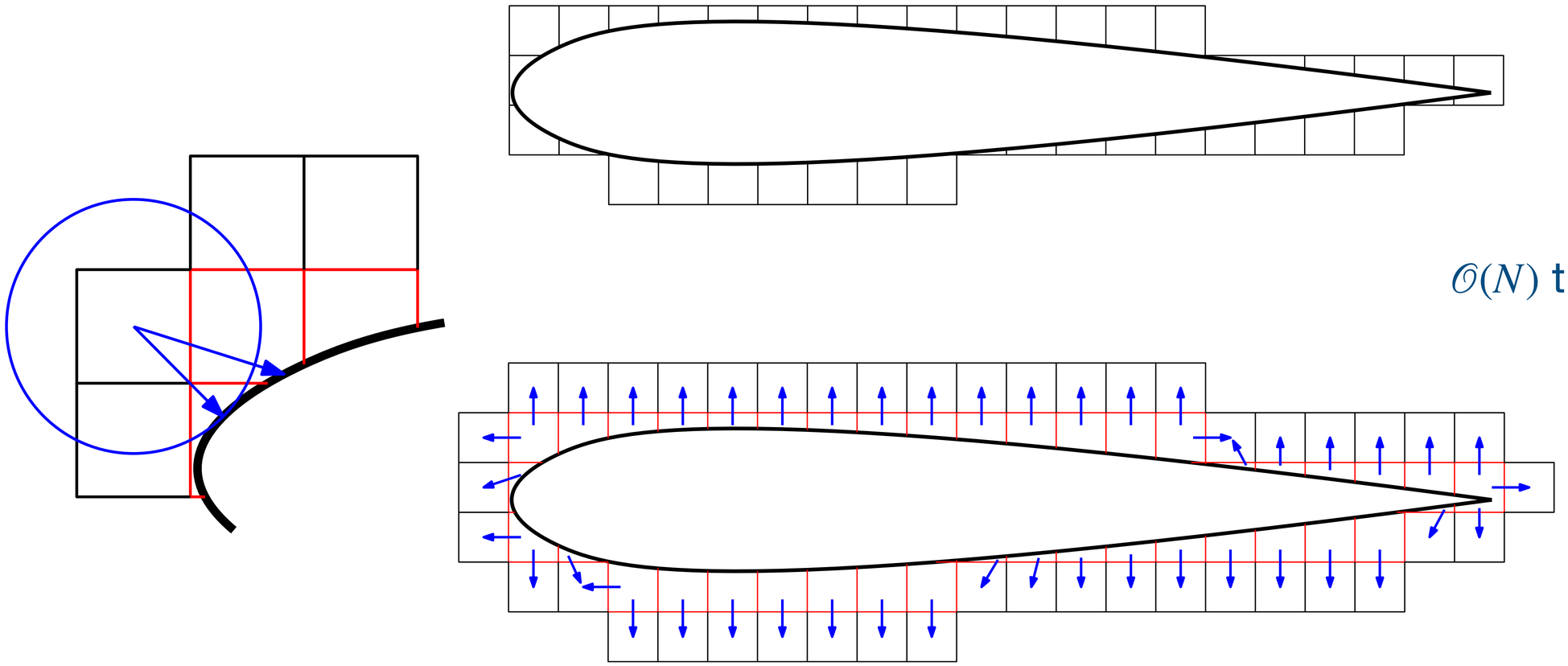
Traveling out from the surface is efficient



$\mathcal{O}(N)$ time complexity?



Traveling out from the surface is efficient



$\mathcal{O}(N)$ time complexity?

Fast methods for the Eikonal equation can take advantage of this type of search to compute distance functions in a single pass

$$|\nabla \underline{T}| = \frac{1}{\underline{f(\vec{x})}}$$

Arrival Time

Wave Speed



$$|\nabla \underline{T}| = 1$$

Distance

Fast methods for the Eikonal equation can take advantage of this type of search to compute distance functions in a single pass

$$|\nabla T| = \frac{1}{f(\vec{x})}$$

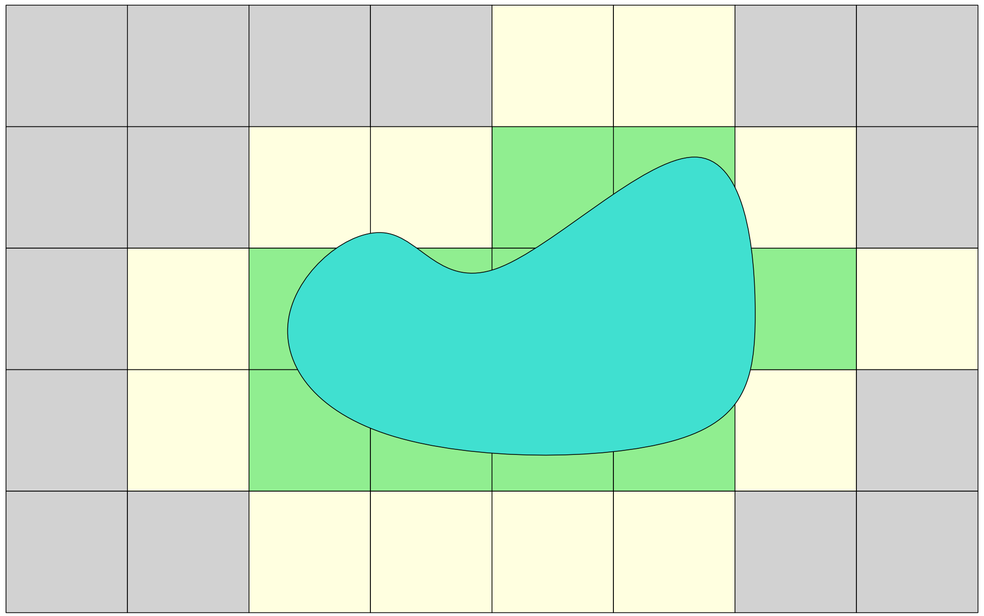
Arrival Time

Wave Speed

$$|\nabla T| = 1$$

Distance

Fast Marching Method



Fast methods for the Eikonal equation can take advantage of this type of search to compute distance functions in a single pass

$$|\nabla T| = \frac{1}{f(\vec{x})}$$

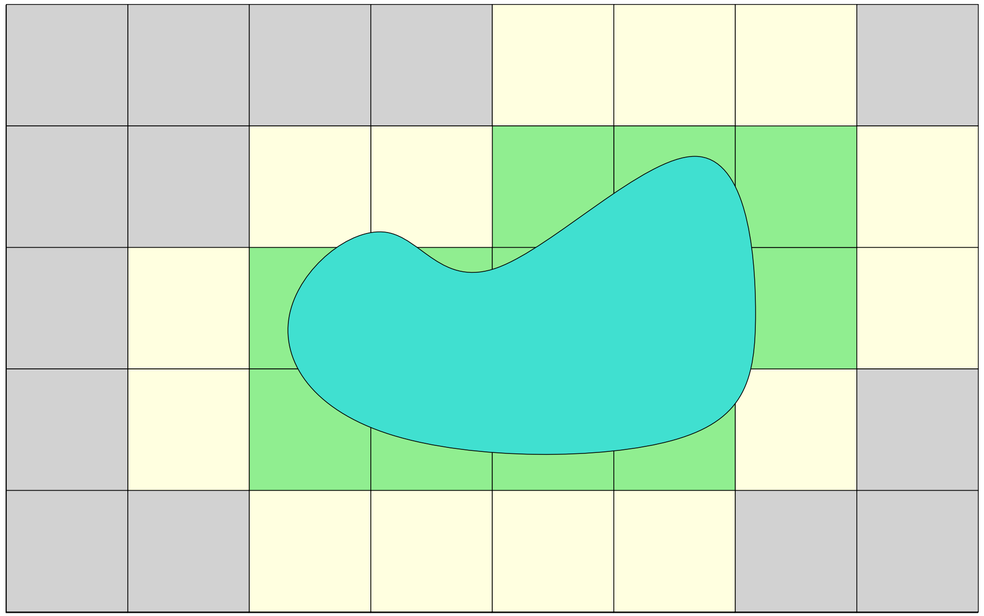
Arrival Time

Wave Speed

$$|\nabla T| = 1$$

Distance

Fast Marching Method



Fast methods for the Eikonal equation can take advantage of this type of search to compute distance functions in a single pass

$$|\nabla T| = \frac{1}{f(\vec{x})}$$

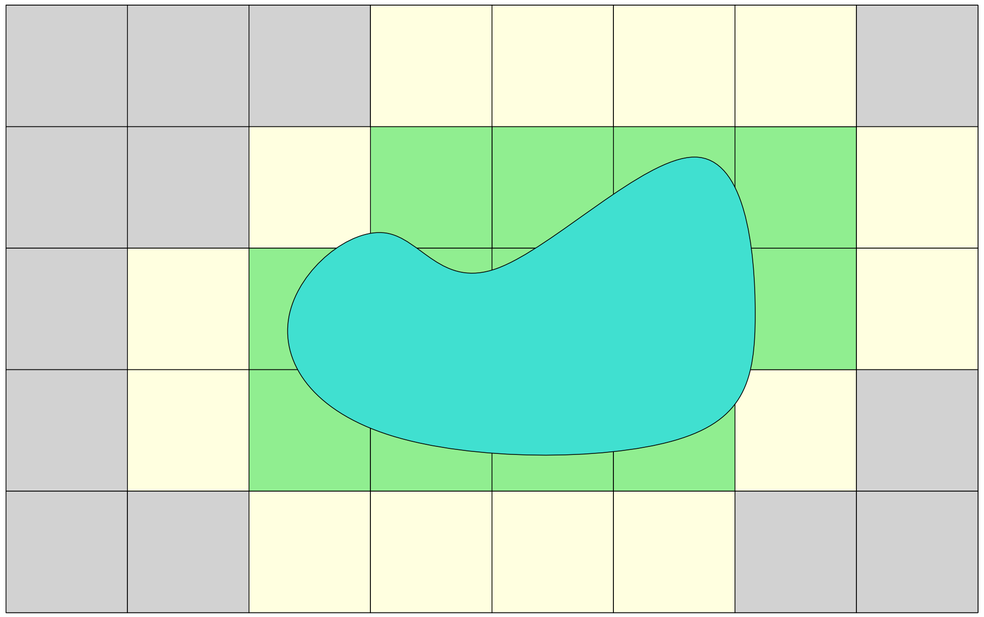
Arrival Time

Wave Speed

$$|\nabla T| = 1$$

Distance

Fast Marching Method



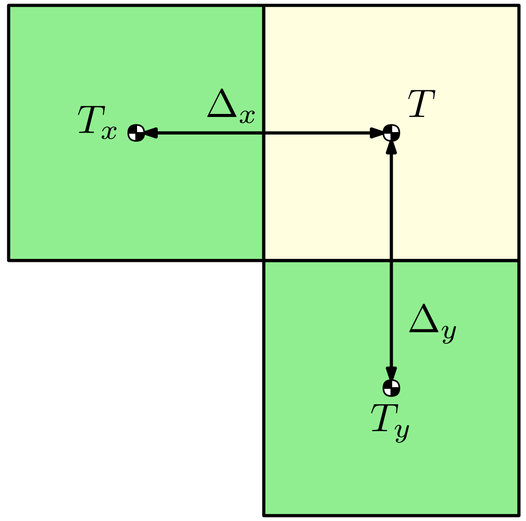
Potential distances are computed using finite differences

$$|\nabla T| = 1 \qquad ||\nabla T||^2 = \left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 = 1$$

Potential distances are computed using finite differences

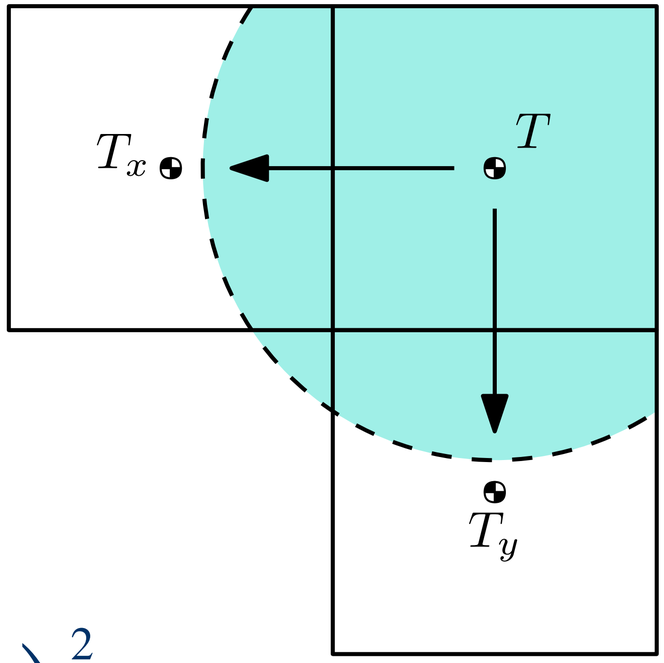
$$|\nabla T| = 1$$

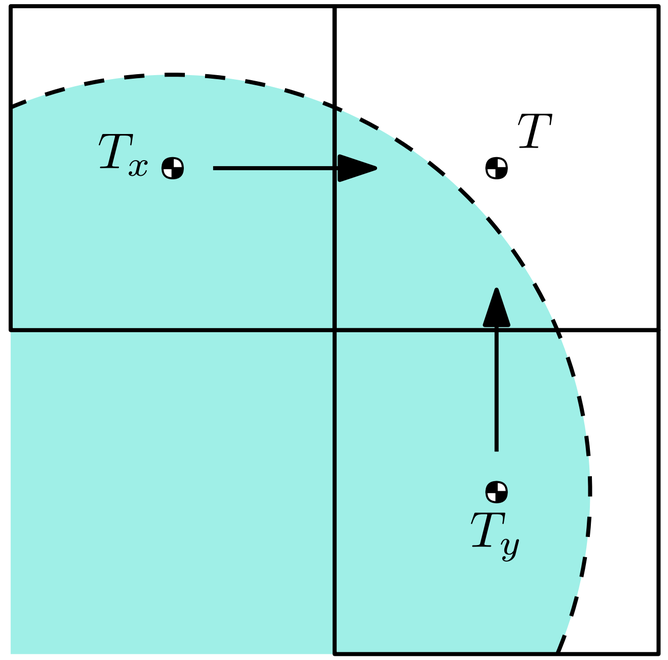
$$||\nabla T||^2 = \left(\frac{\partial T}{\partial x}\right)^2 + \left(\frac{\partial T}{\partial y}\right)^2 = 1$$



$$\left(\frac{T - T_x}{\Delta_x}\right)^2 + \left(\frac{T - T_y}{\Delta_y}\right)^2 = 1$$

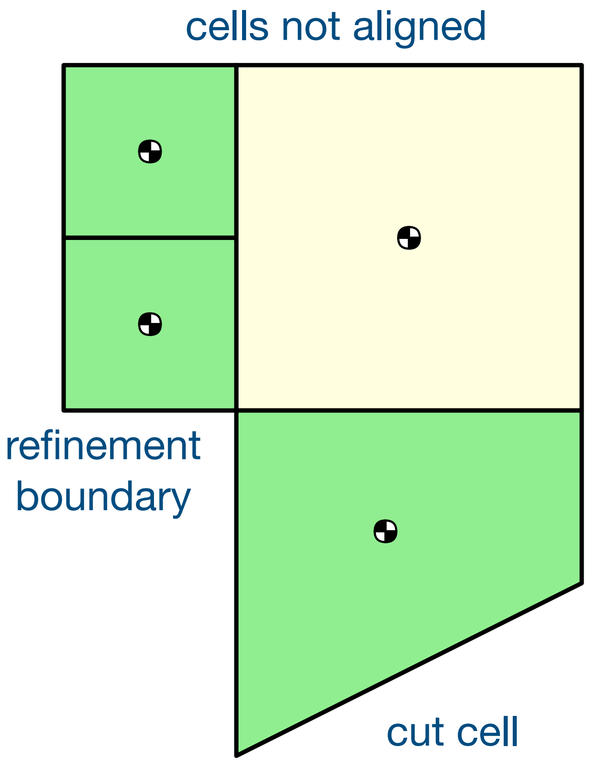
The two solutions correspond to a upwind and downwind solutions





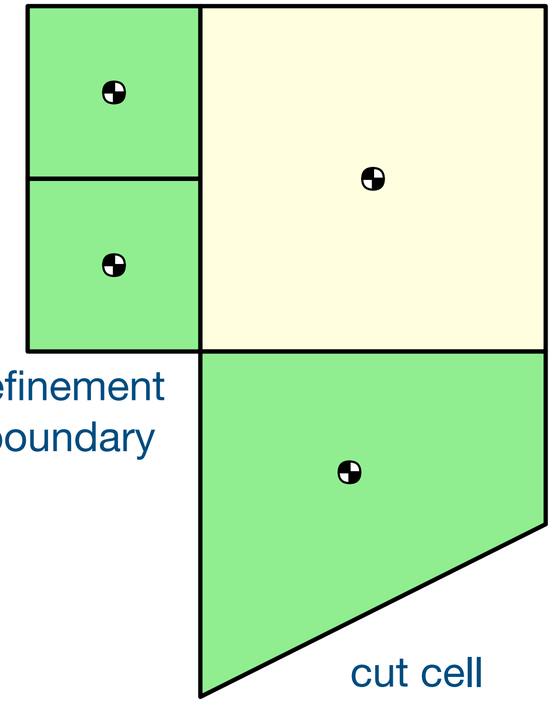
$$\left(\frac{T-T_x}{\Delta_x}\right)^2+\left(\frac{T-T_y}{\Delta_y}\right)^2=1$$

Non-Cartesian meshes can lead to problems

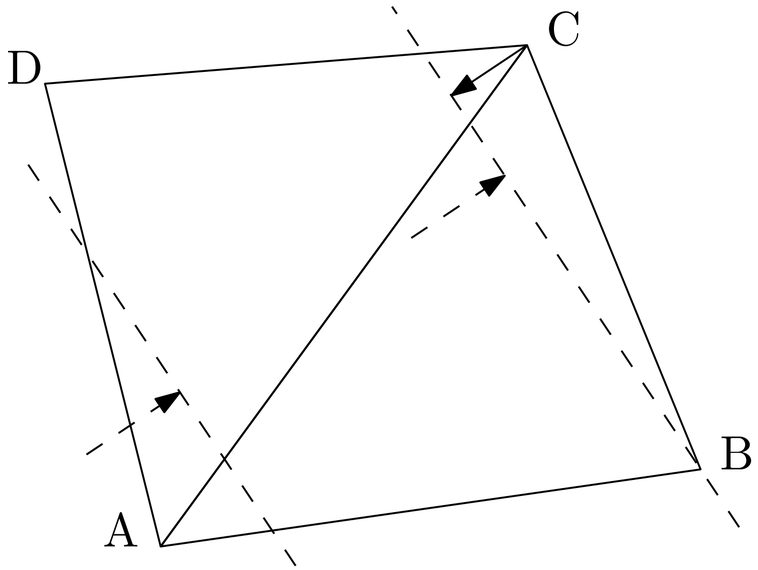


Non-Cartesian meshes can lead to problems

cells not aligned

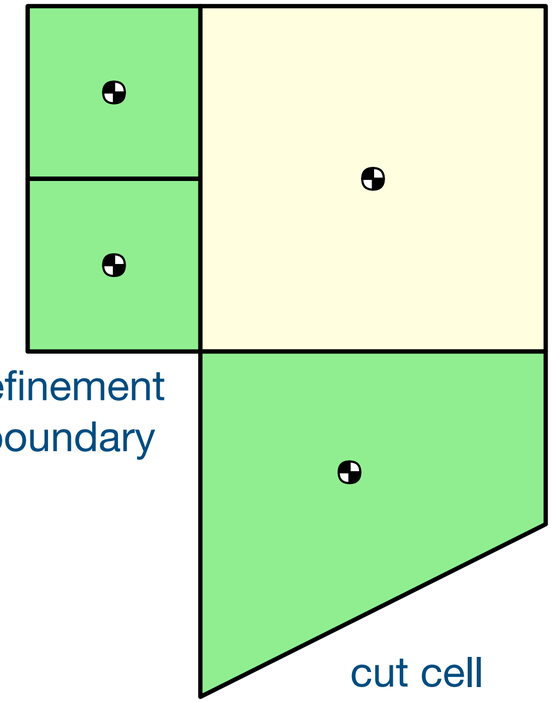


must identify proper stencil

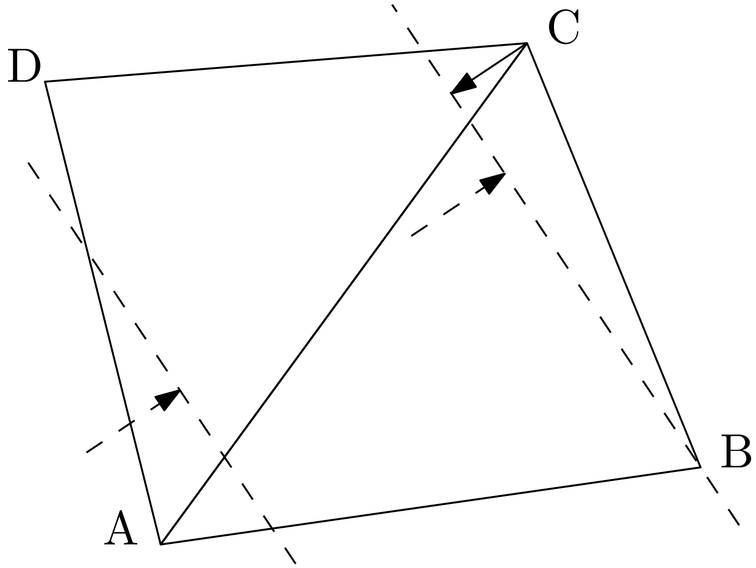


Non-Cartesian meshes can lead to problems

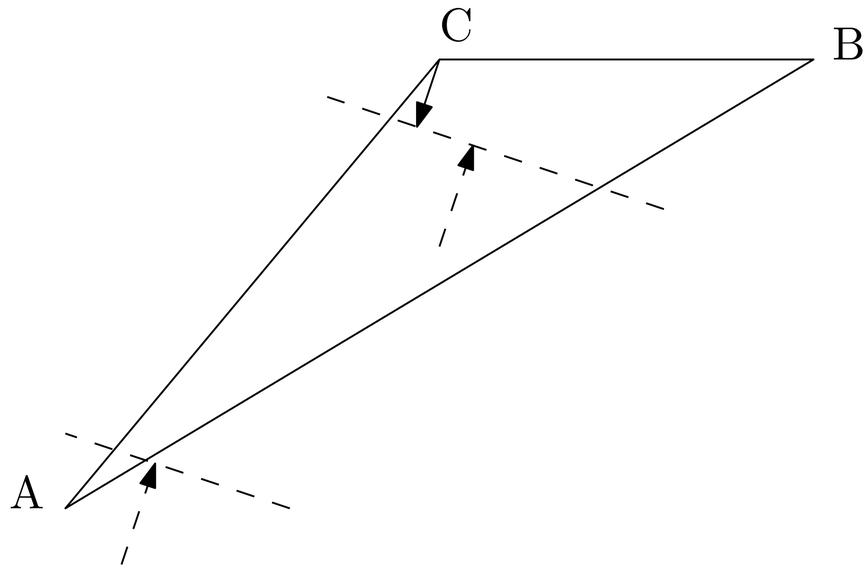
cells not aligned



must identify proper stencil



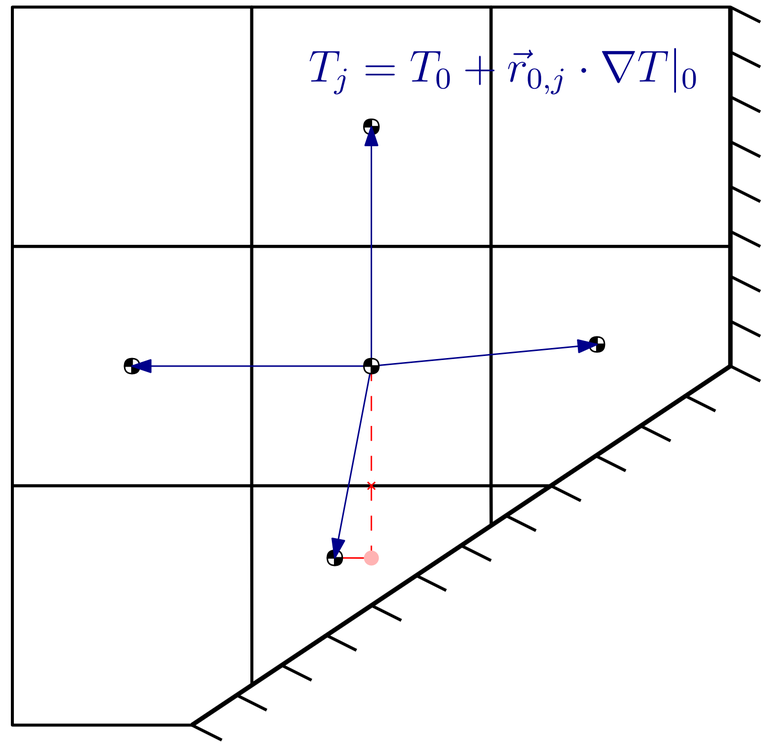
proper stencil can lack support



I developed a method for handling all of these situations through a unified and robust framework.

Least-squares gradient reconstruction replaces finite difference approximation of the derivatives

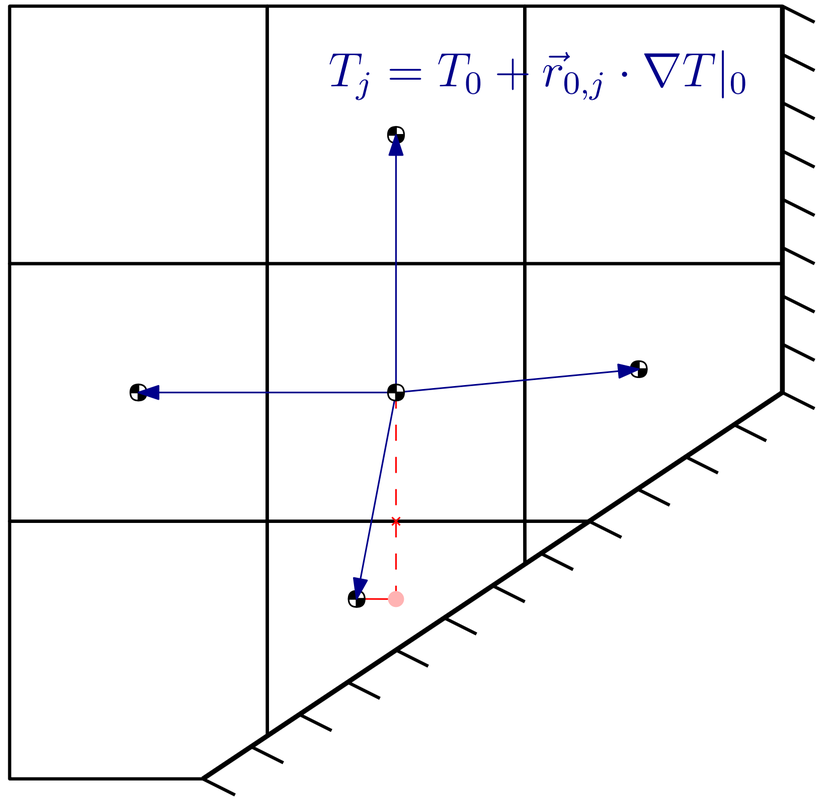
$$\frac{\partial T}{\partial x} \approx \frac{T - T_x}{\Delta_x} \qquad \frac{\partial T}{\partial y} \approx \frac{T - T_y}{\Delta_x}$$



The Eikonal equation is used as a constraint on the least-squares optimization

minimize $\left(\underline{\underline{\mathbf{R}}}\nabla T + \underline{T} - \underline{T_j}\right)^T \left(\underline{\underline{\mathbf{R}}}\nabla T + \underline{T} - \underline{T_j}\right)$

subject to $||\nabla T||^2 = 1$



A change of variables removes the constraint

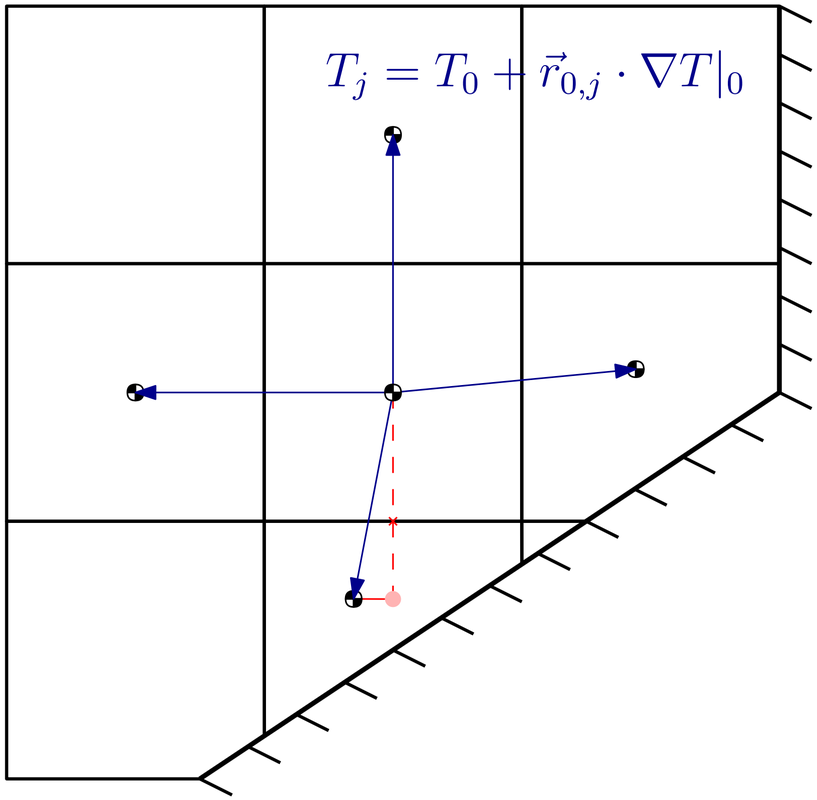
minimize $\left(\underline{\underline{\mathbf{R}}}\nabla T + \underline{T} - \underline{T_j}\right)^T \left(\underline{\underline{\mathbf{R}}}\nabla T + \underline{T} - \underline{T_j}\right)$

subject to $||\nabla T||^2 = 1$



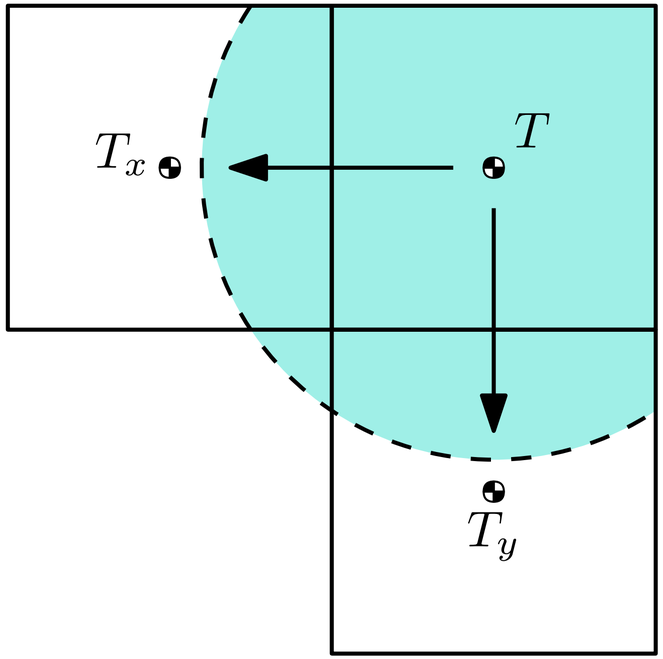
minimize $\left(\underline{\underline{\mathbf{R}}}\nabla T + \underline{T} - \underline{T_j}\right)^T \left(\underline{\underline{\mathbf{R}}}\nabla T + \underline{T} - \underline{T_j}\right)$

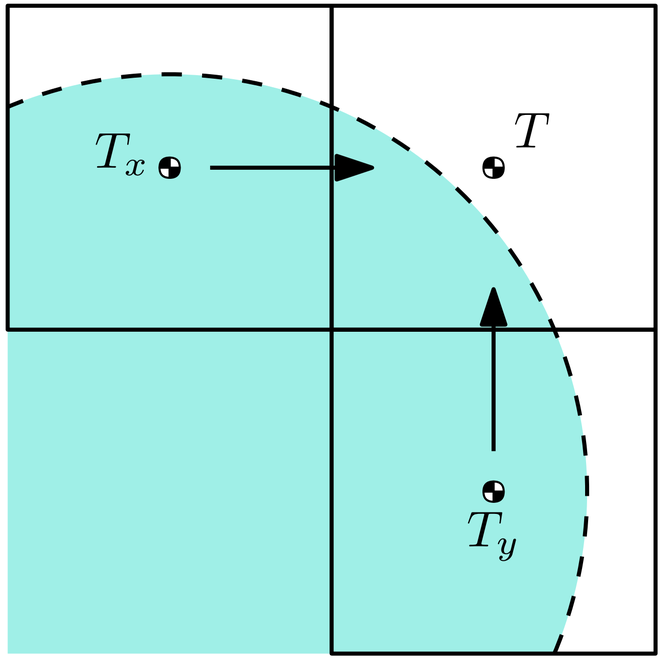
with $\frac{\partial T}{\partial x} = \cos(\theta)$ and $\frac{\partial T}{\partial y} = \sin(\theta)$



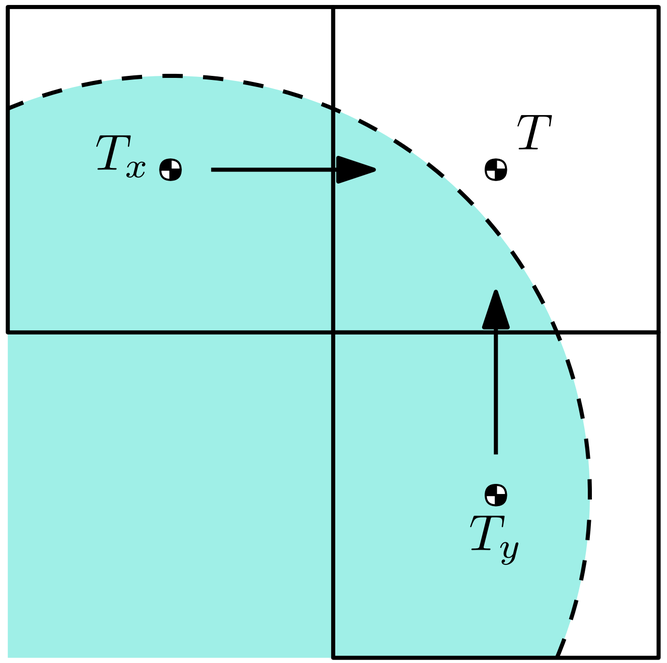
Variable bounds on T are used to enforce the upwind condition

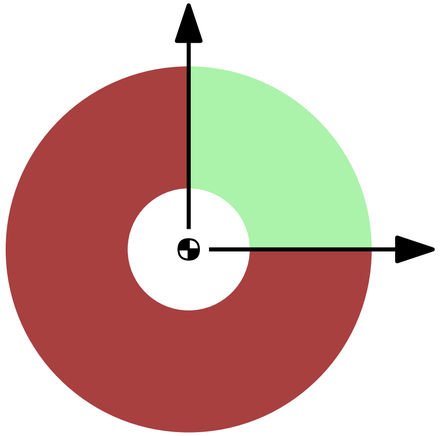
$T > T_i \quad \forall\, T_i \in$ stencil



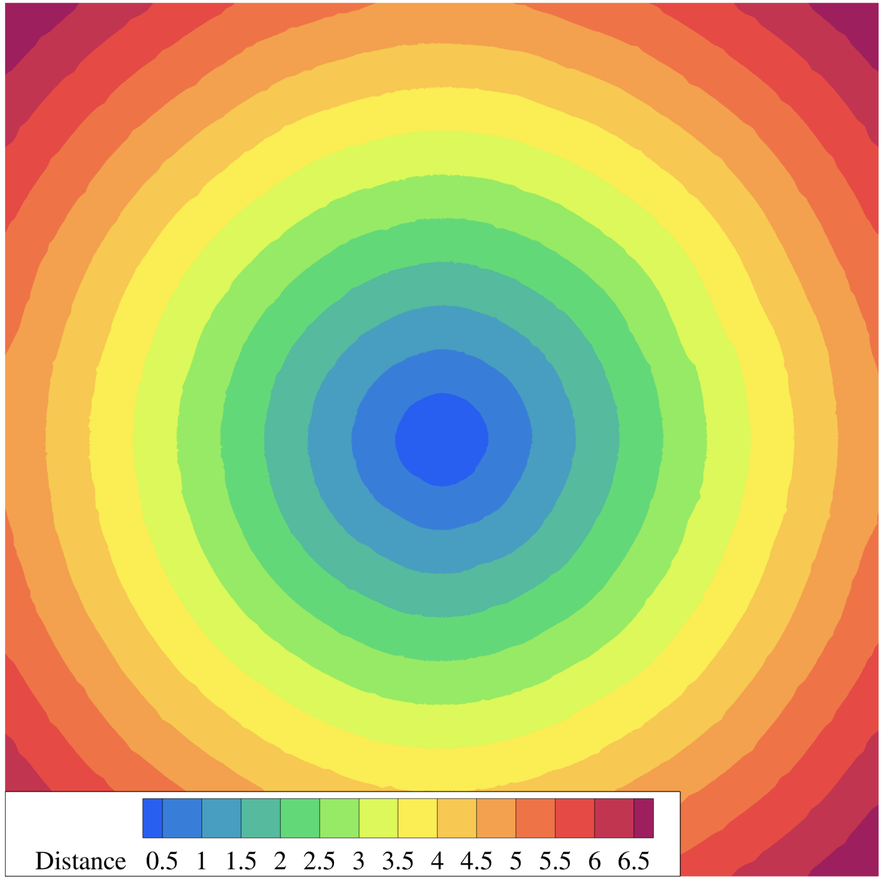


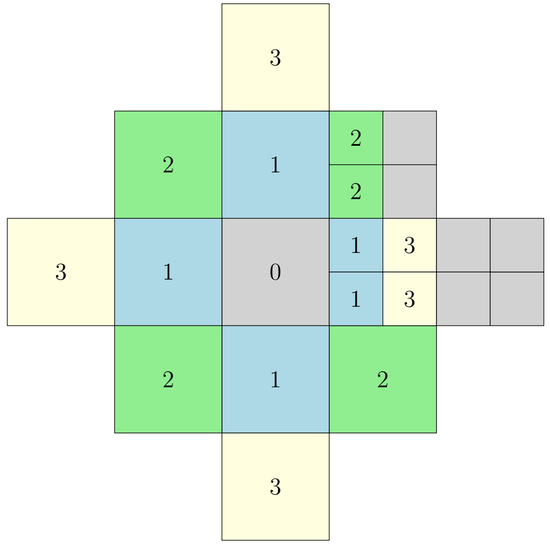
Variable bounds on θ are used to enforce the causality condition



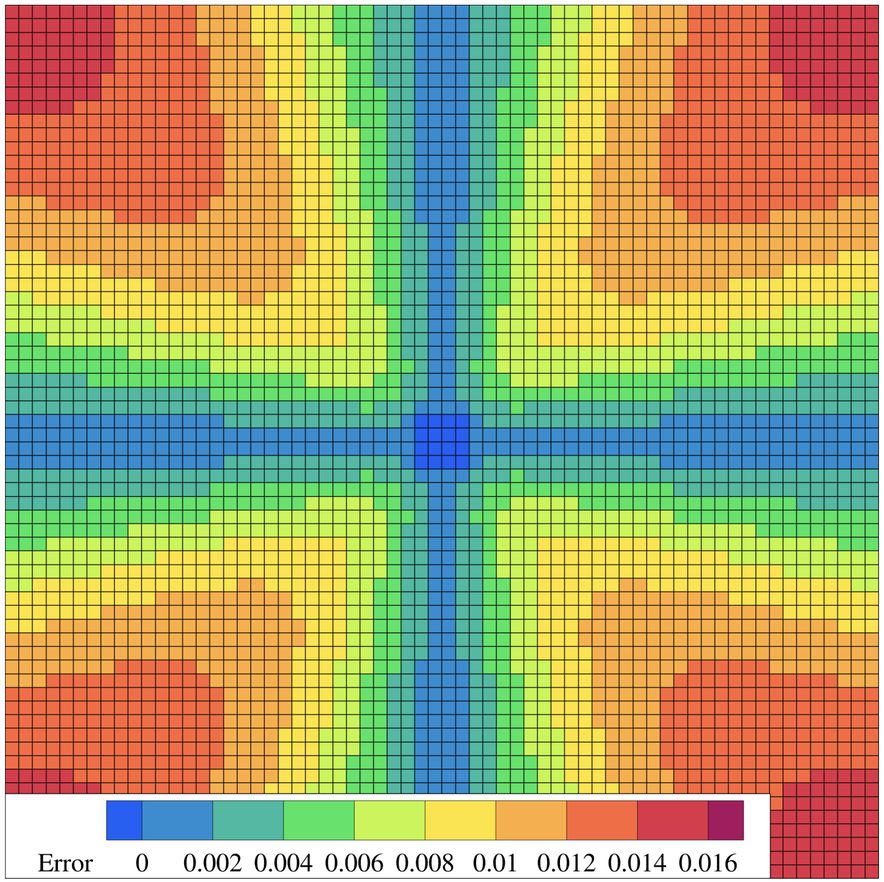


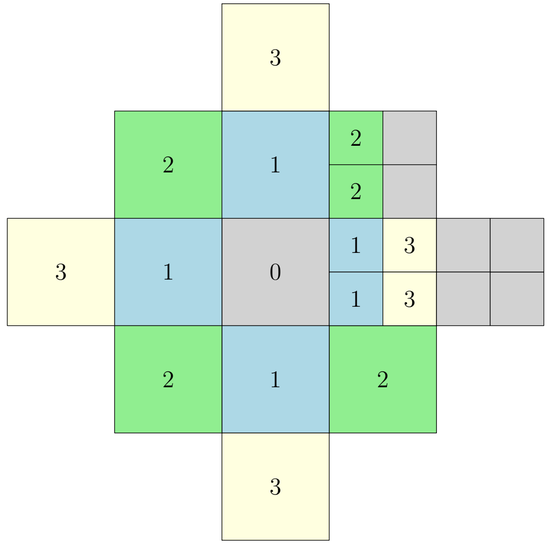
Disc distance function example



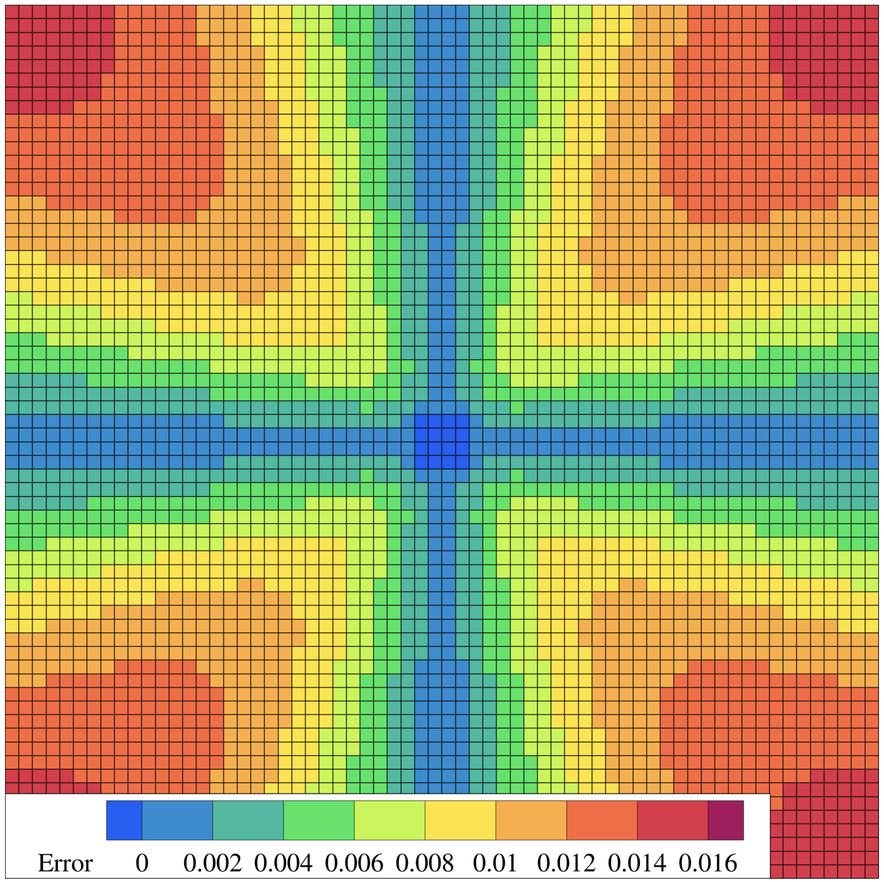


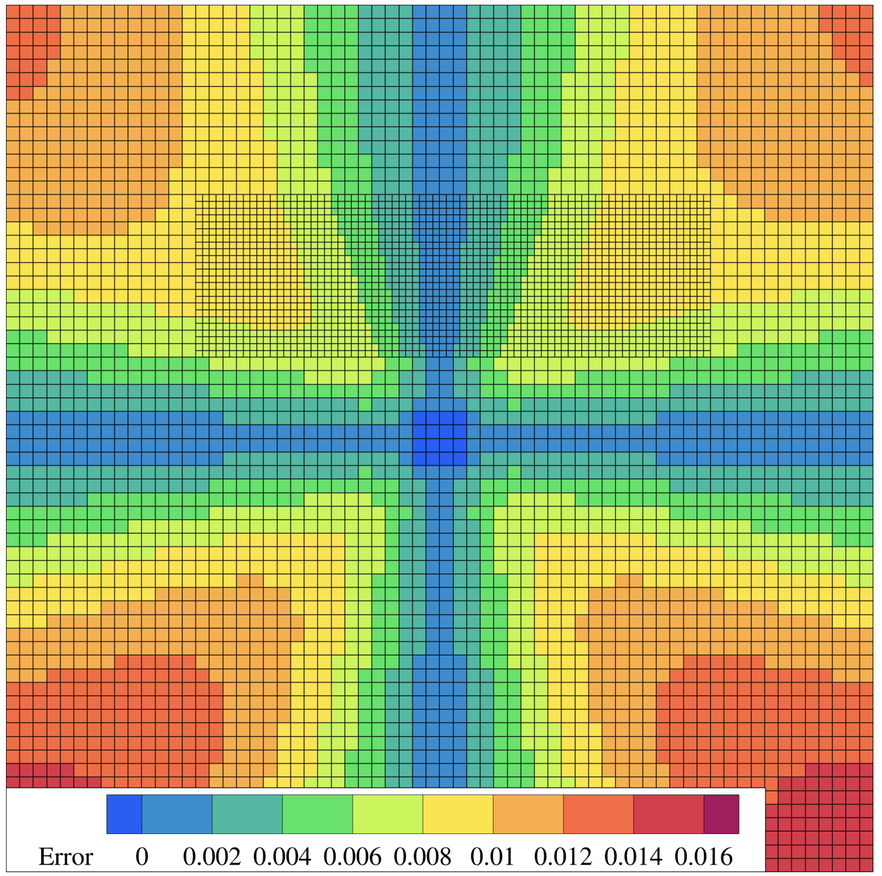
Error accumulates along the diagonals

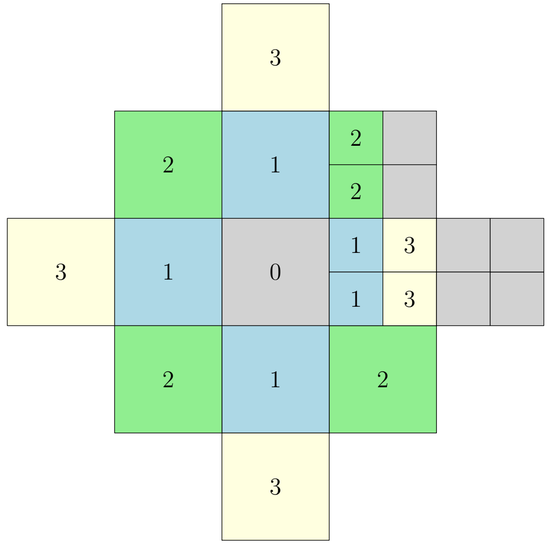




Error accumulates along the diagonals

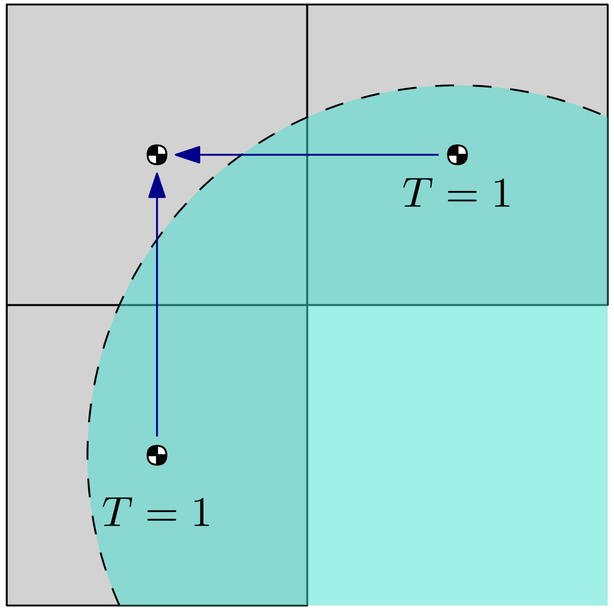




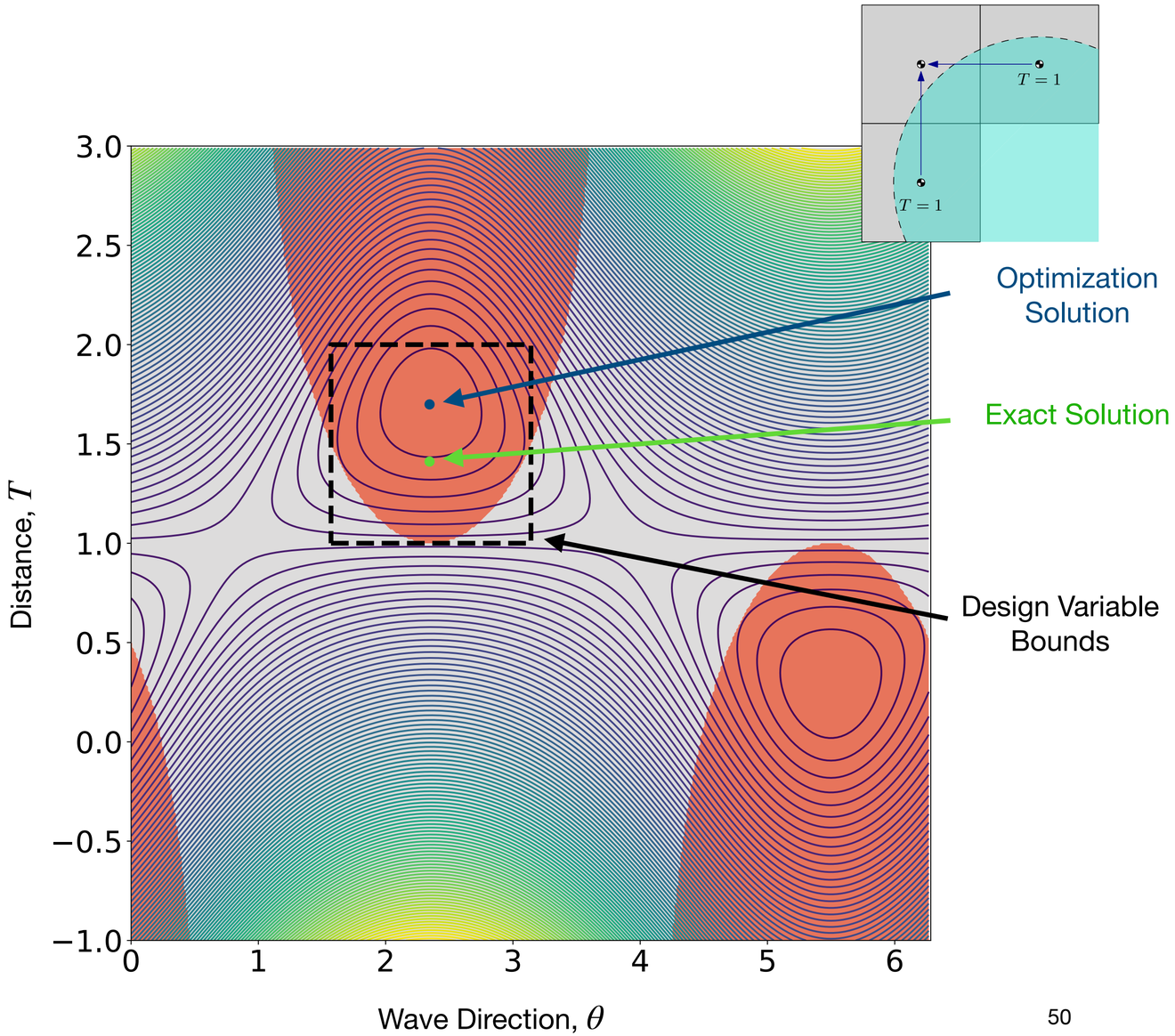


An upper bound on T ensures exact straight line distances do not include discretization errors

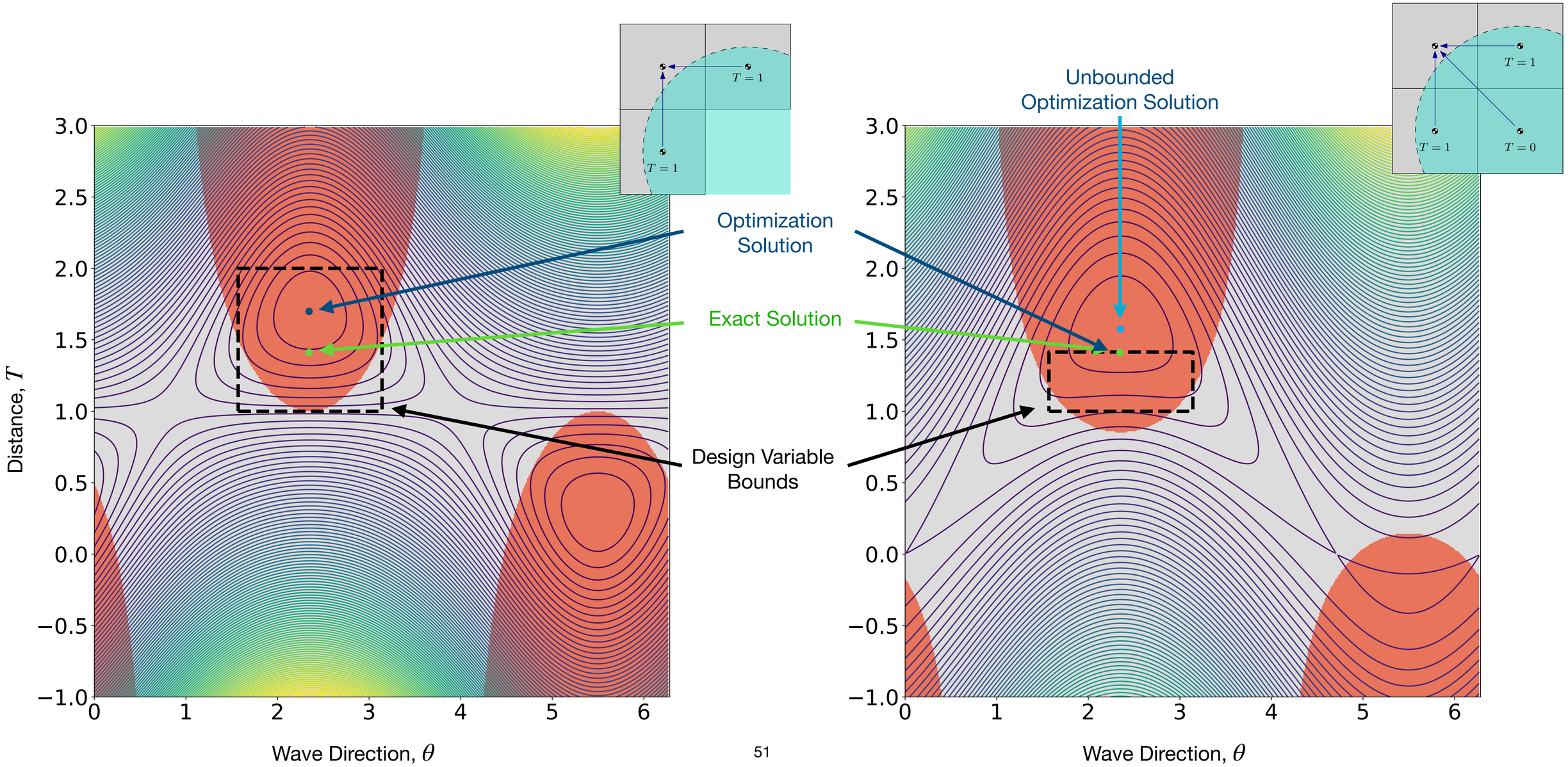
$$T < \min_{T_i \in stencil} T_i + ||\vec{x}_o - \vec{x}_i||$$



Design Spaces

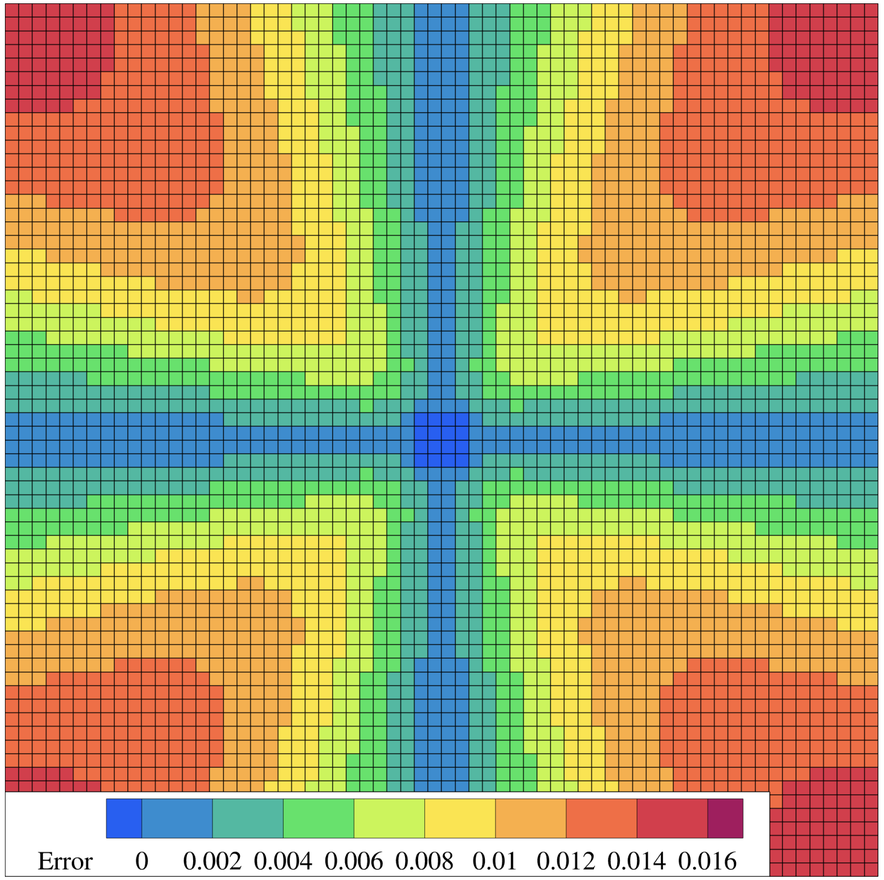


The upper bound allows the optimizer to find the exact solution

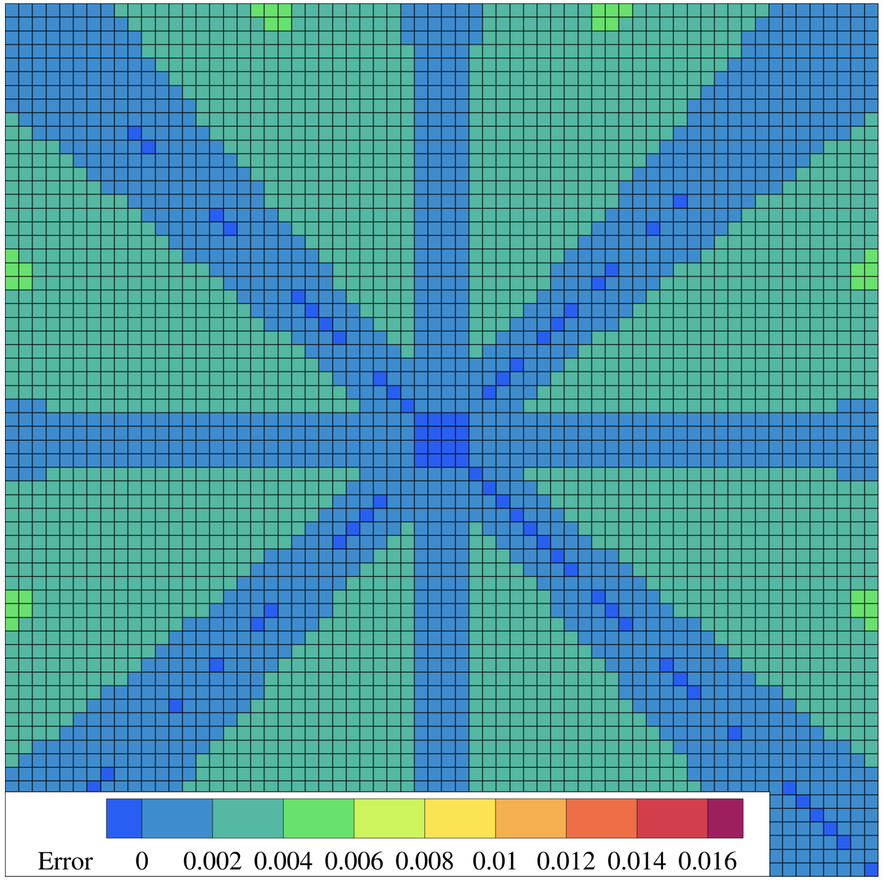


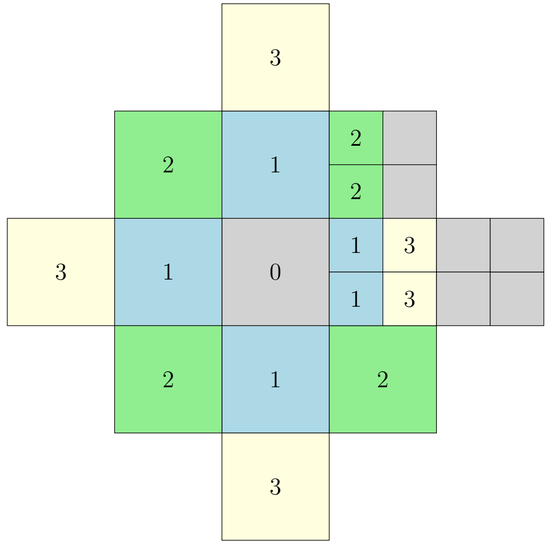
Diagonals no longer accumulate error

Level 1 Stencil

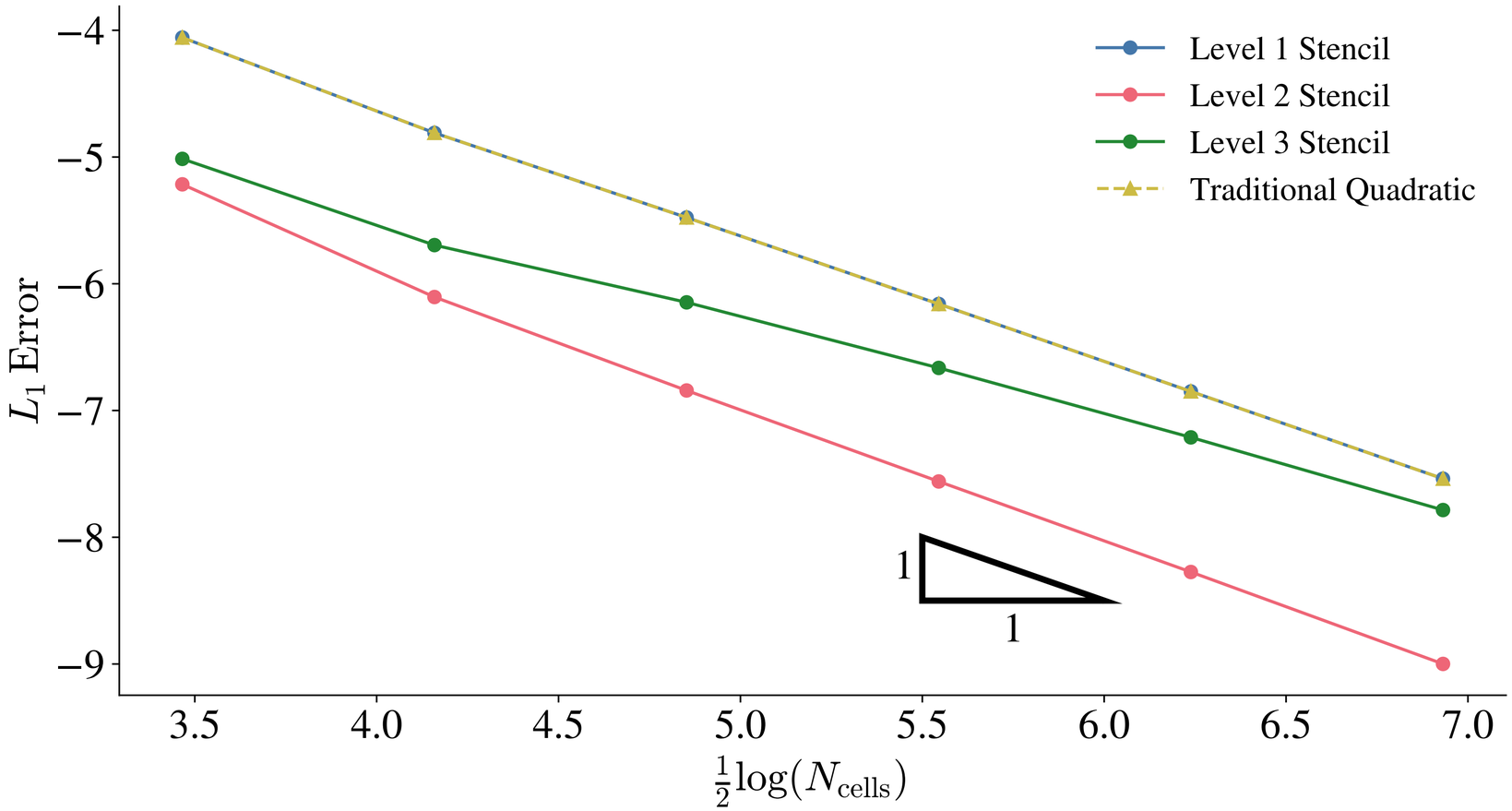


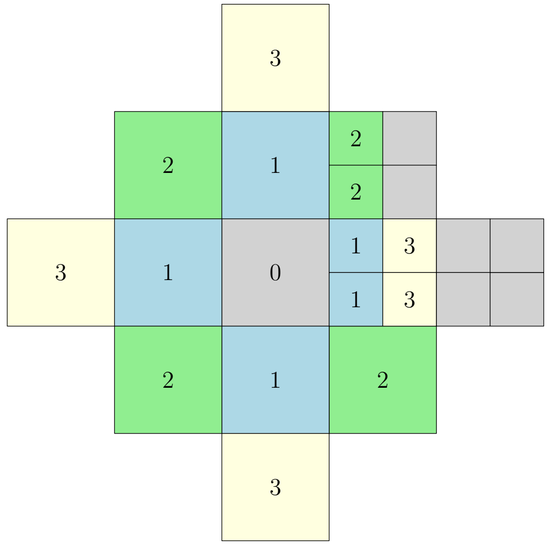
Level 2 Stencil





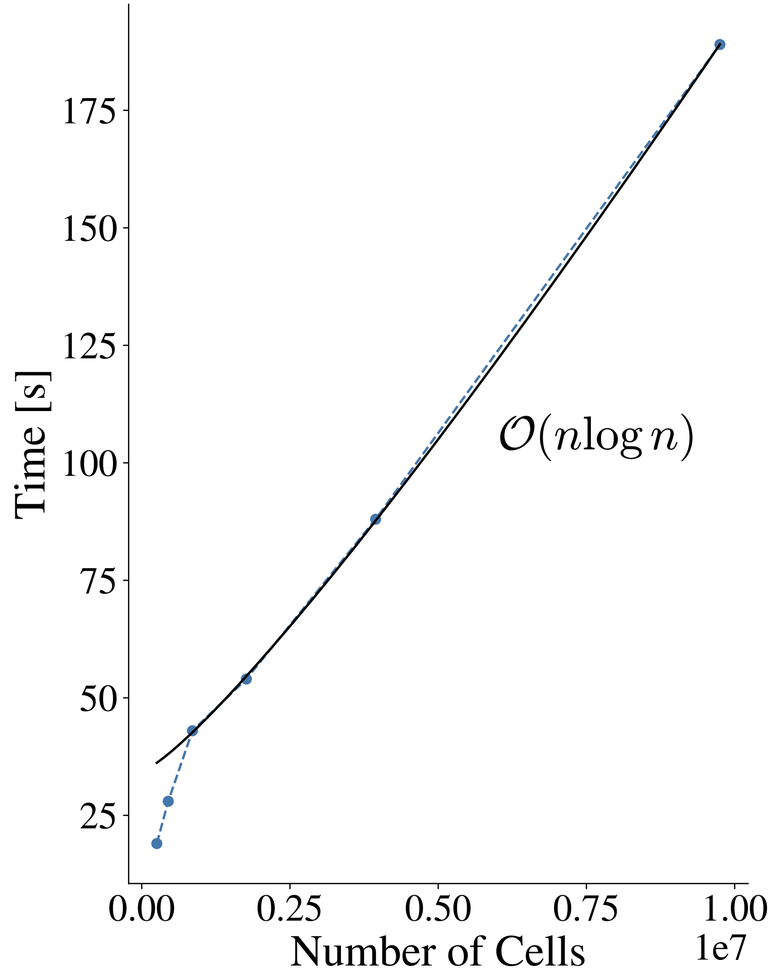
Eikonal Order of Accuracy





Eikonal solver is robust and efficient

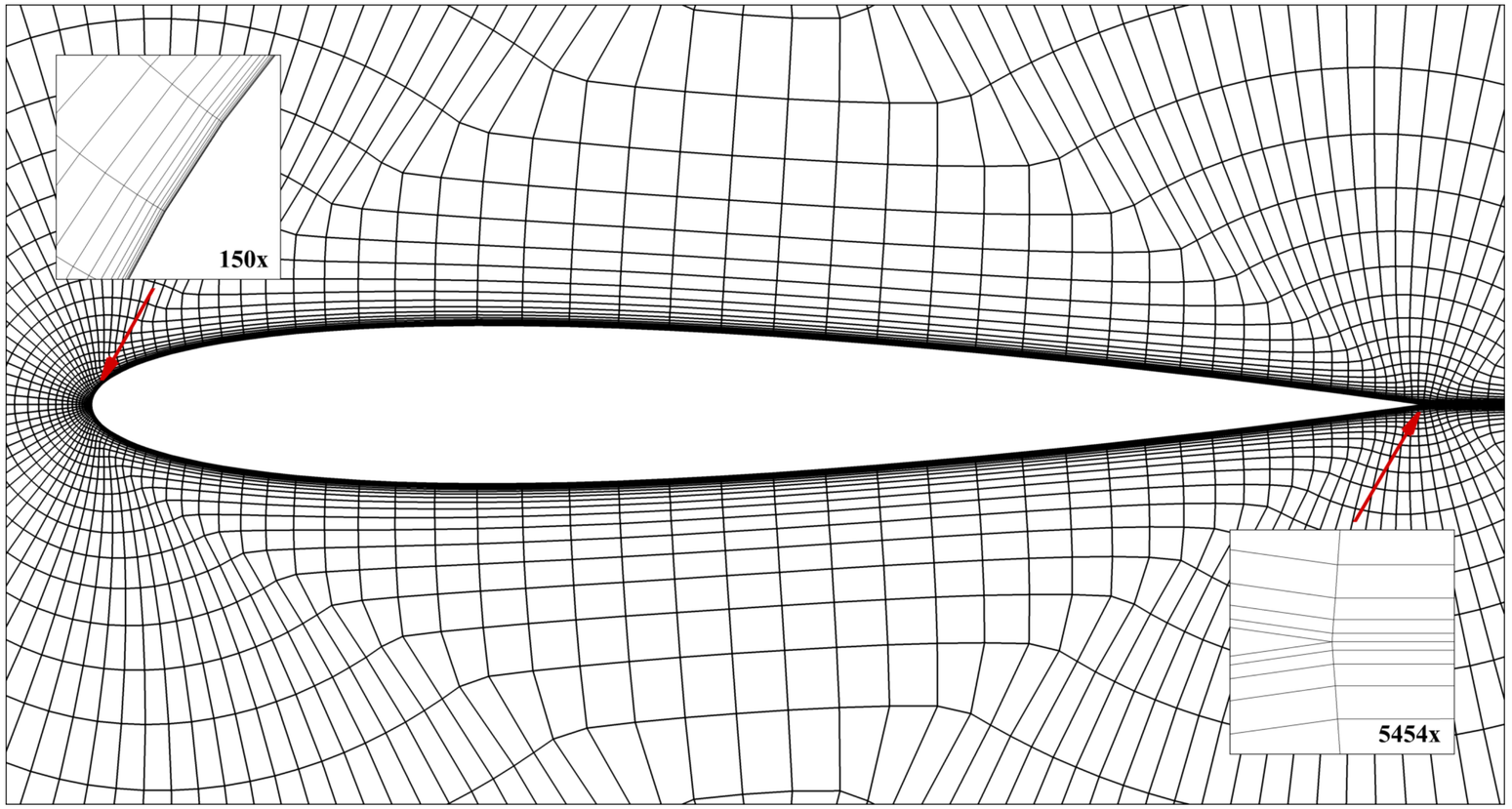
- Demonstrates efficient time complexity scaling
- Works on arbitrary point clouds
- Extends to higher order accuracy and dimensions

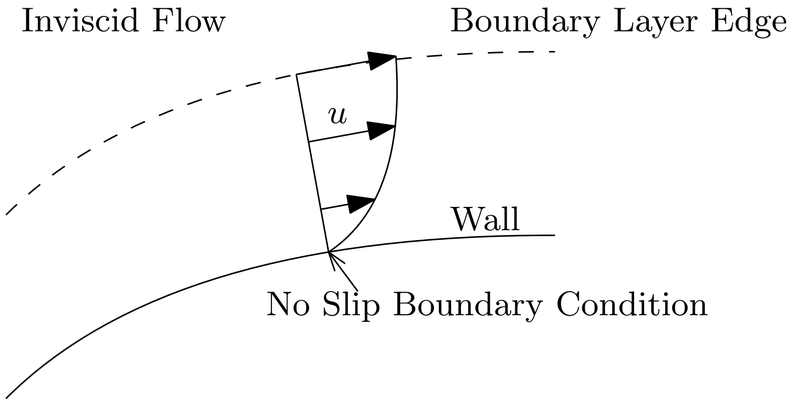


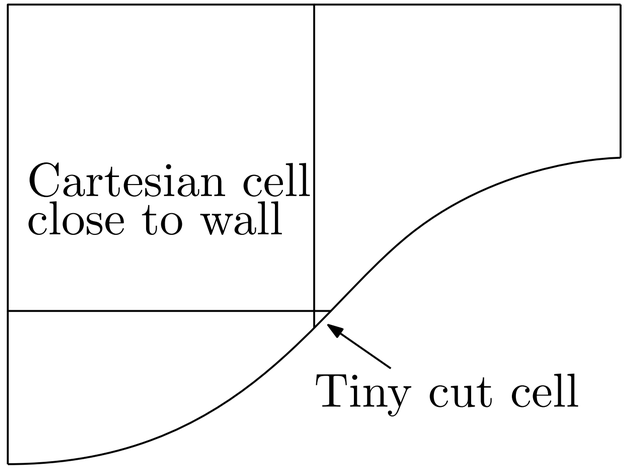
Outline

- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance function
- ODE wall model
- NASA TMR test cases
- Conclusions

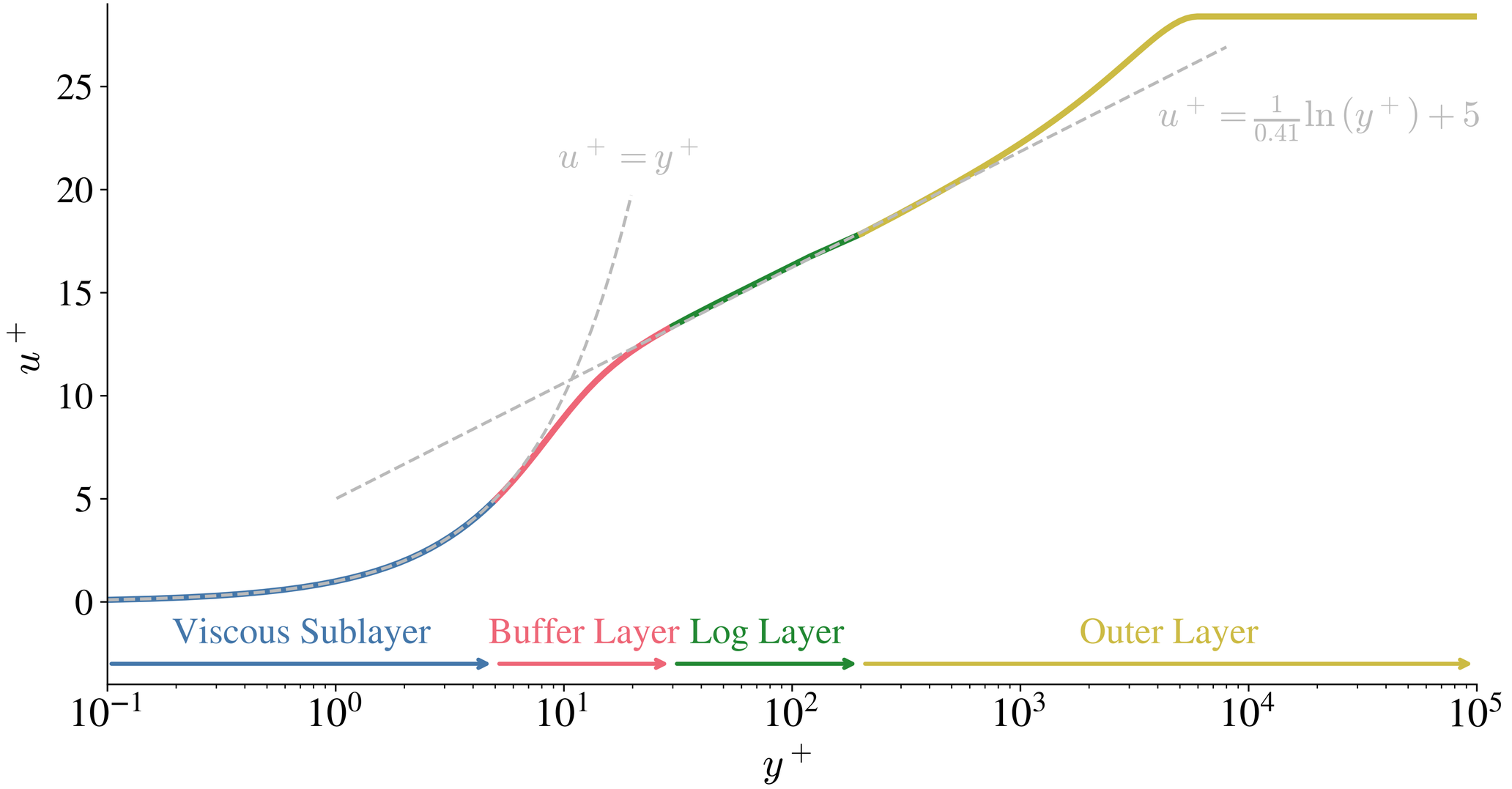
Solving the RANS-SA equations on Cartesian cut-cell meshes



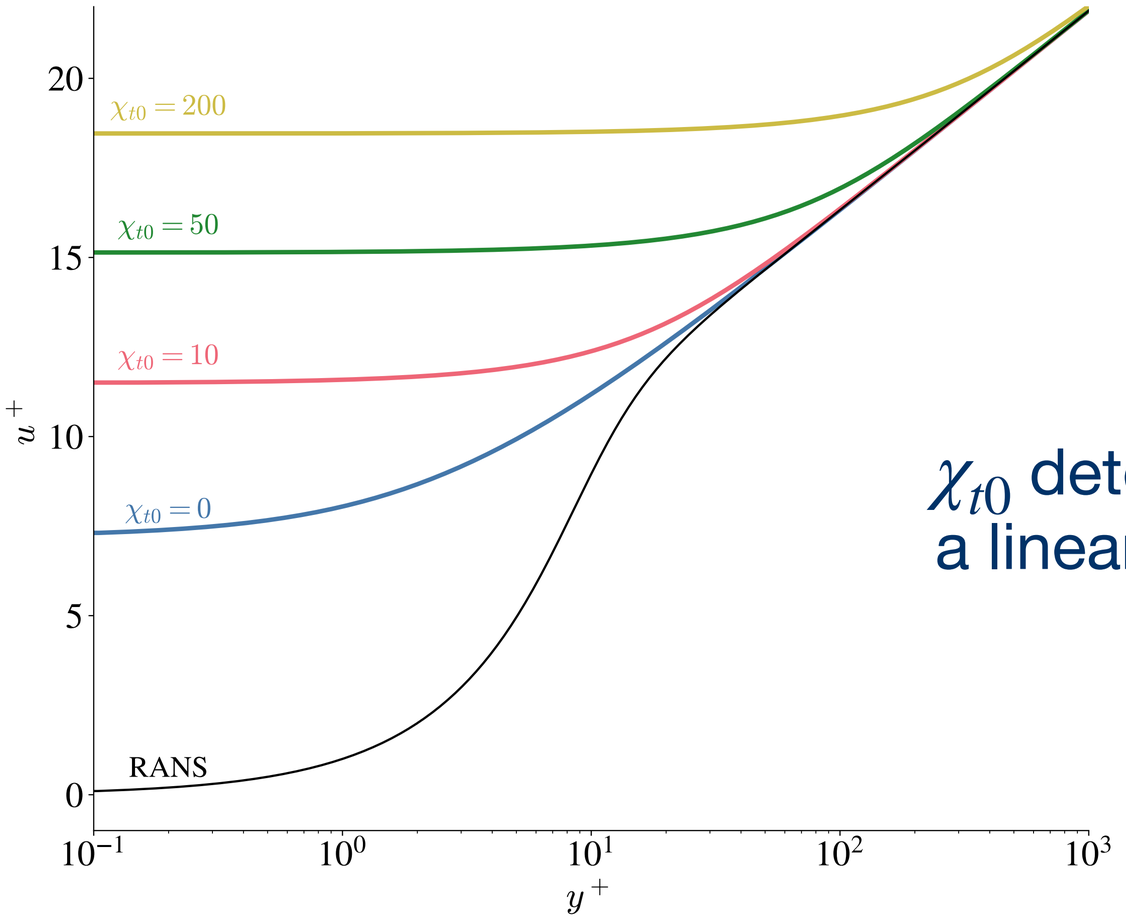




Wall resolved simulations need points in the viscous sublayer



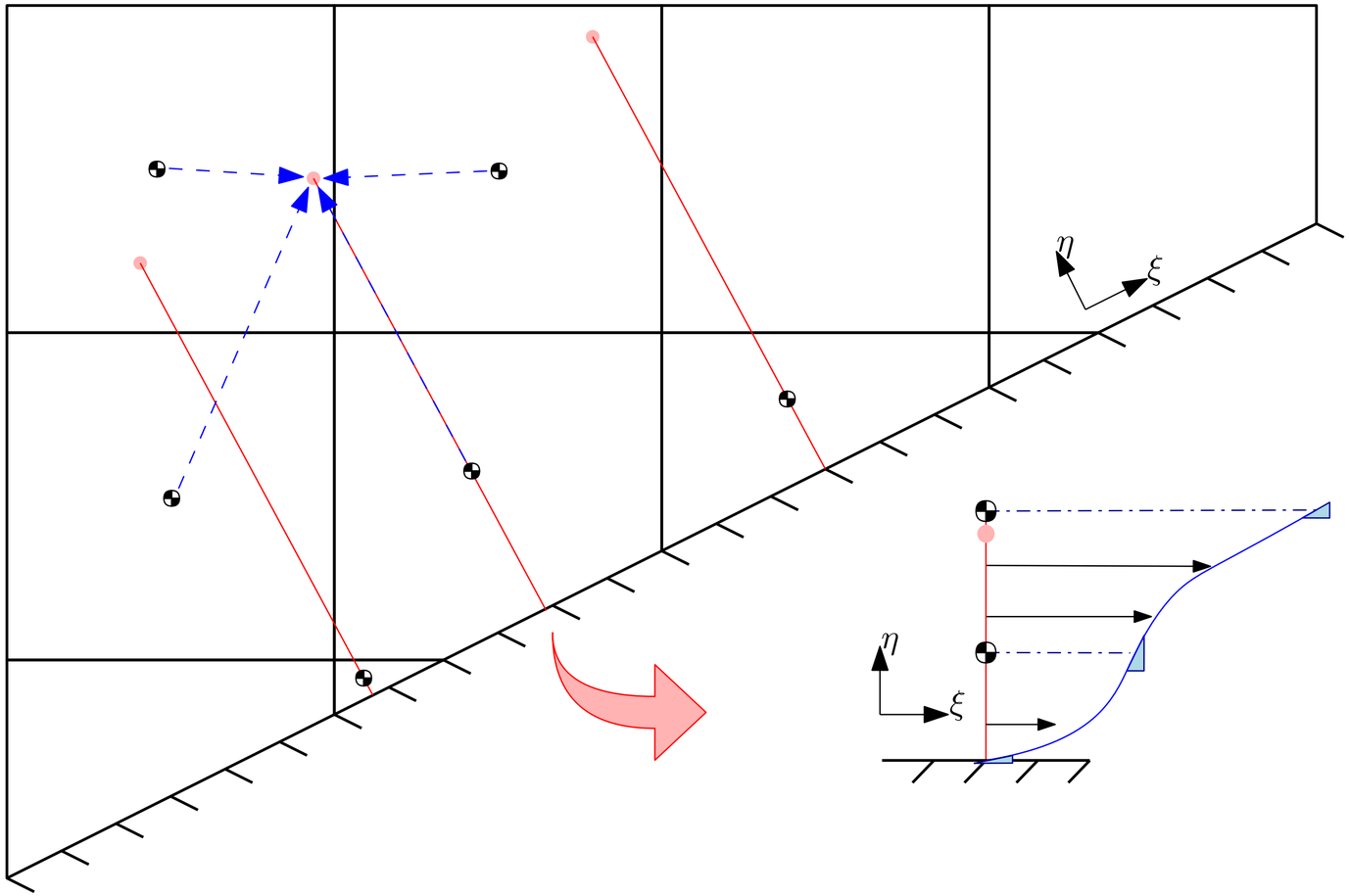
Slip wall approach modifies the governing equations [1]



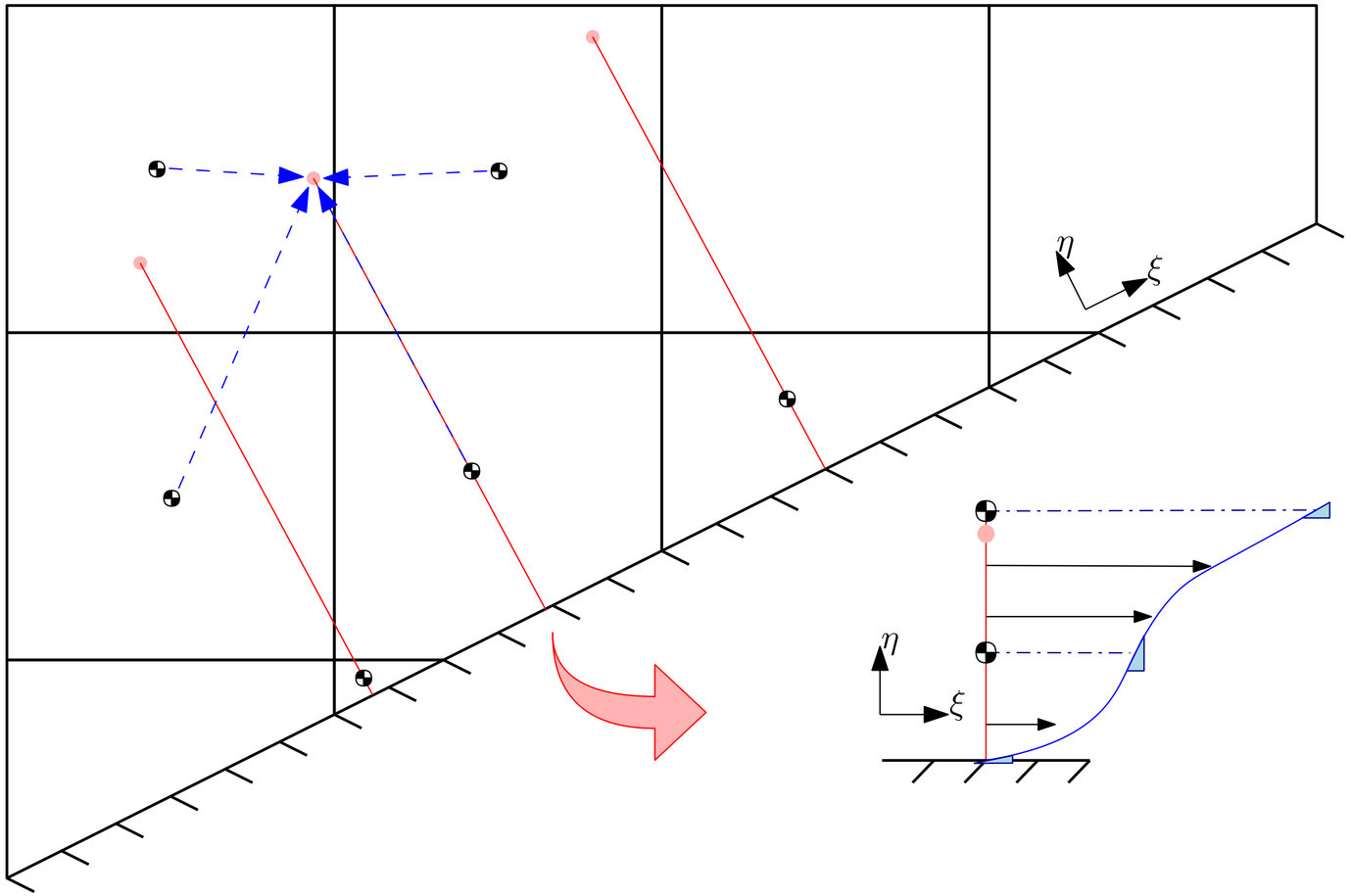
χ_{t0} determines how quickly a linear profile appears

[1] 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.

Interior forcing points inform boundary layer approximations



Interior forcing points inform boundary layer approximations

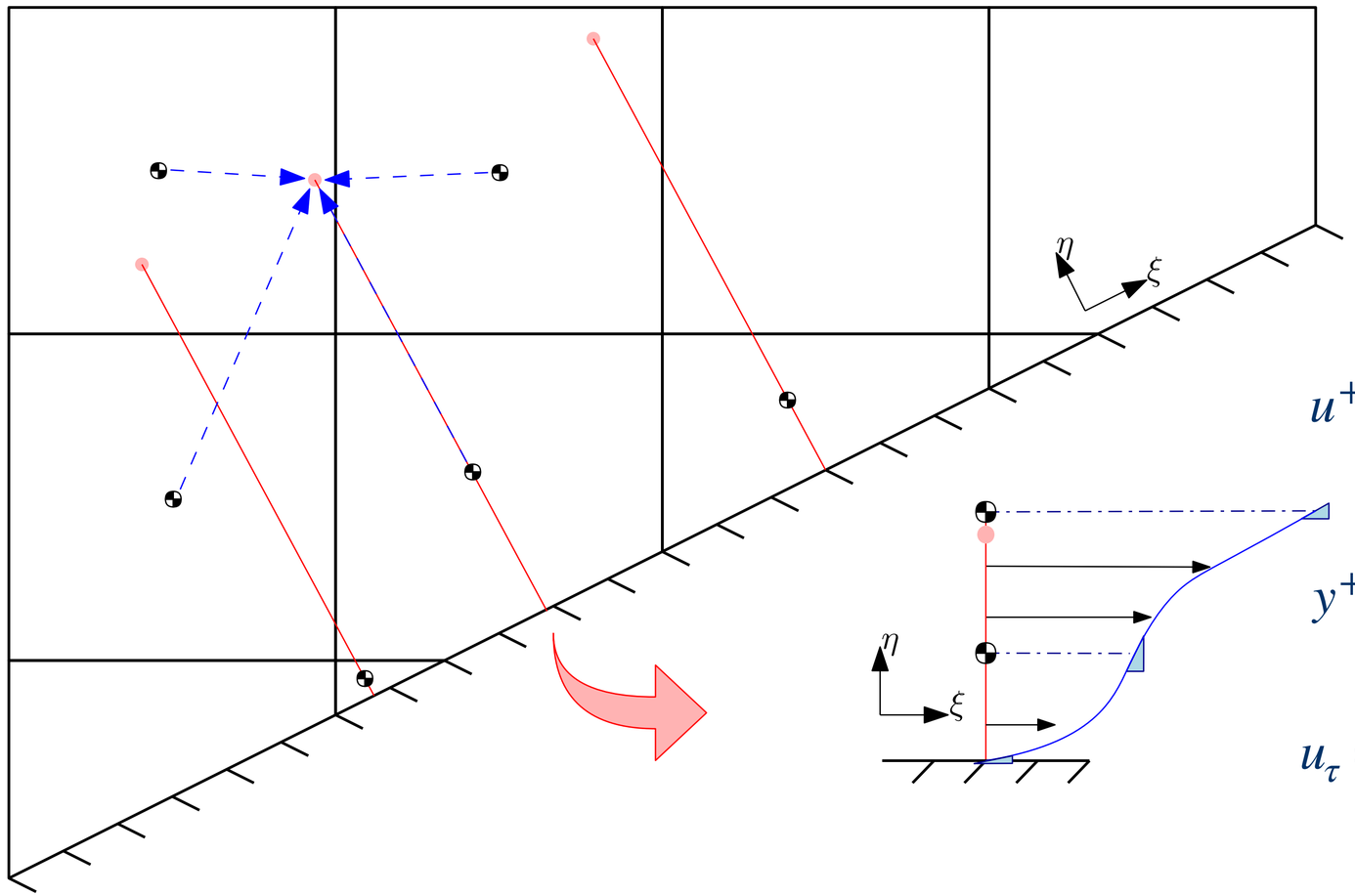


η wall-normal distance

ξ wall-tangential distance

q_t wall-tangential velocity

Interior forcing points inform boundary layer approximations



η wall-normal distance

ξ wall-tangential distance

q_t wall-tangential velocity

$u^+ = \frac{q_t}{u_\tau}$ non dimensional wall-tangential velocity

$y^+ = \frac{\eta u_\tau}{\nu}$ non dimensional wall-normal distance

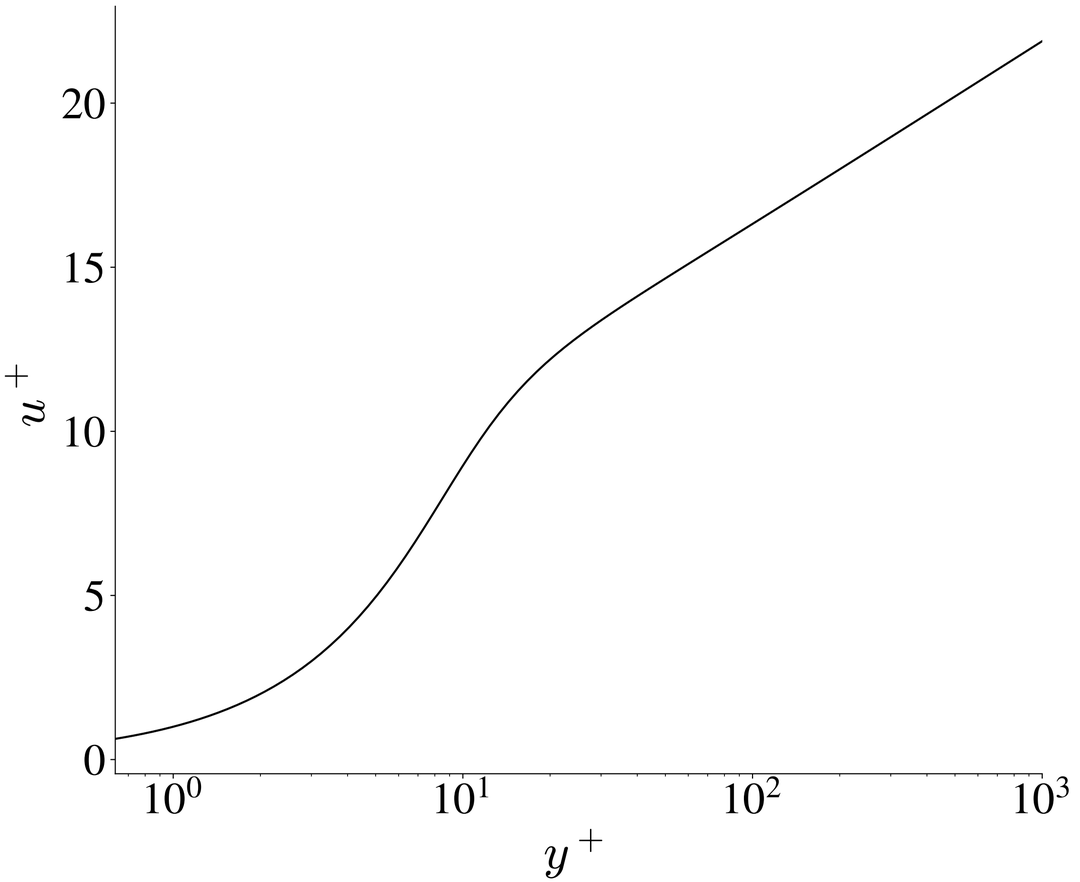
$u_\tau = \sqrt{\frac{\tau_w}{\rho}}$ friction velocity

SA analytical wall function derived from equilibrium model [2]

$$(\nu + \nu_t)\frac{\partial q_t}{\partial \eta} = u_\tau^2$$ Wall tangential momentum

$$\tilde{\nu} = \kappa u_\tau \eta$$ Turbulence variable

$$u^+(y^+) = \bar{B} + c_1 \log\left(\left(y^+ + a_1\right)^2 + b_1^2\right) - c_2 \log\left(\left(y^+ + a_2\right)^2 + b_2^2\right)$$
$$- c_3 \,\text{atan2}\left(b_1, y^+ + a_1\right) - c_4 \,\text{atan2}\left(b_2, y^+ + a_2\right),$$



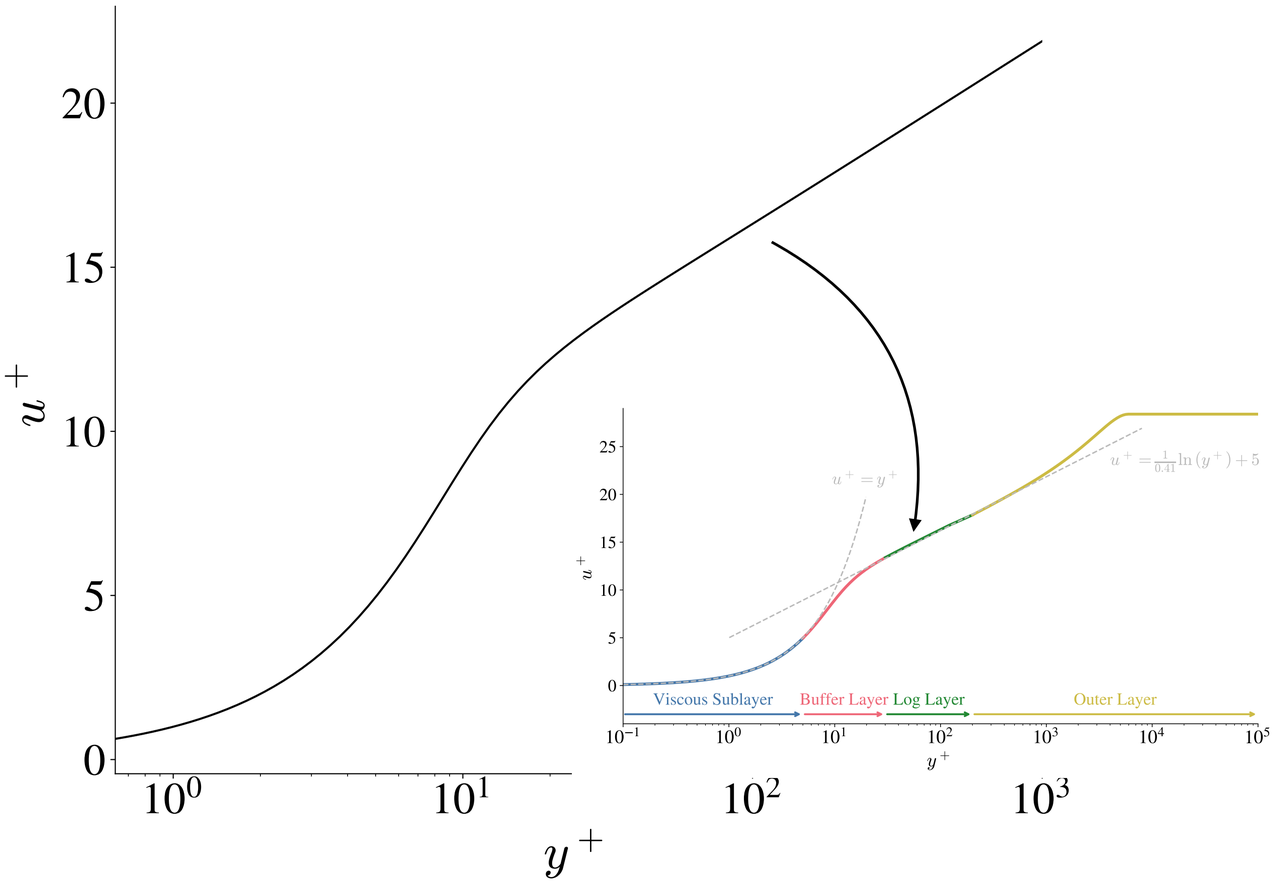
[2] Marsha Berger and Michael Aftosmis. Progress towards a Cartesian cut-cell method for viscous compressible flow. In 50th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, page 1301, 2012.

SA analytical wall function derived from equilibrium model [2]

$$(\nu + \nu_t)\frac{\partial q_t}{\partial \eta} = u_\tau^2$$ Wall tangential momentum

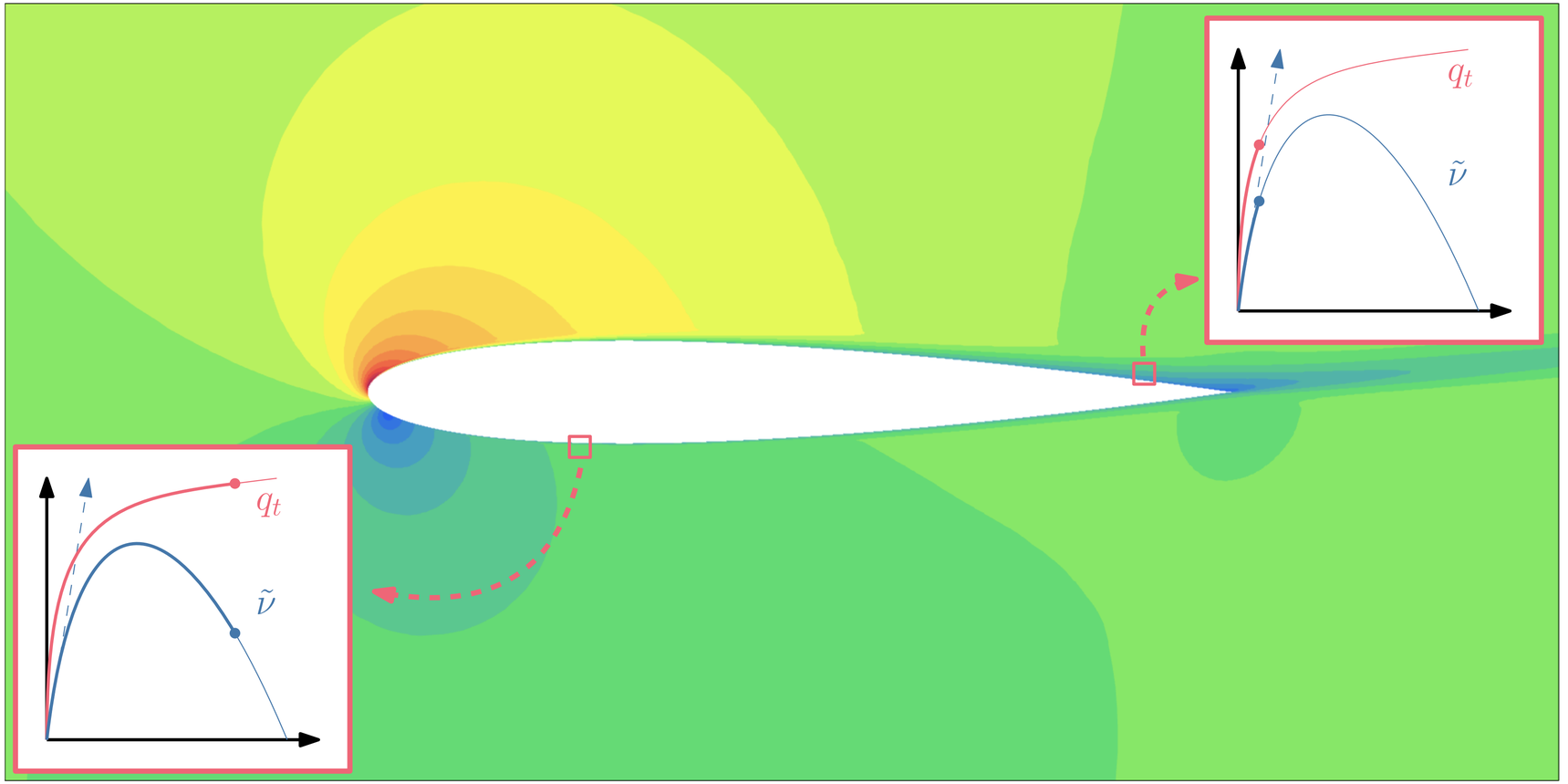
$$\tilde{\nu} = \kappa u_\tau \eta$$ Turbulence variable

$$u^+(y^+) = \bar{B} + c_1 \log\left(\left(y^+ + a_1\right)^2 + b_1^2\right) - c_2 \log\left(\left(y^+ + a_2\right)^2 + b_2^2\right)$$
$$- c_3 \,\text{atan2}\left(b_1, y^+ + a_1\right) - c_4 \,\text{atan2}\left(b_2, y^+ + a_2\right),$$



[2] 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.

SA wall function is only valid within the log region



ODE wall model [3] reintroduces pressure momentum balance

$$\frac{\partial}{\partial \eta}\left[(\nu+\nu_t)\frac{\partial q_t}{\partial \eta}\right]=0$$

$$\tilde{\nu}=\kappa u_\tau \eta$$



$$\frac{\partial}{\partial \eta}\left[(\mu+\mu_t)\frac{\partial u}{\partial \eta}\right]=\left.\frac{\partial p}{\partial \xi}\right|_F+\psi(\eta)\rho\Big|_F\left[u_F\left.\frac{\partial u}{\partial \xi}\right|_F+v_F\left.\frac{\partial u}{\partial \eta}\right|_F\right]$$

$$\frac{\partial}{\partial \eta}\left[\frac{1}{\sigma}(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial \eta}\right]=-\frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial \eta}\right)^2-(P-D),$$

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

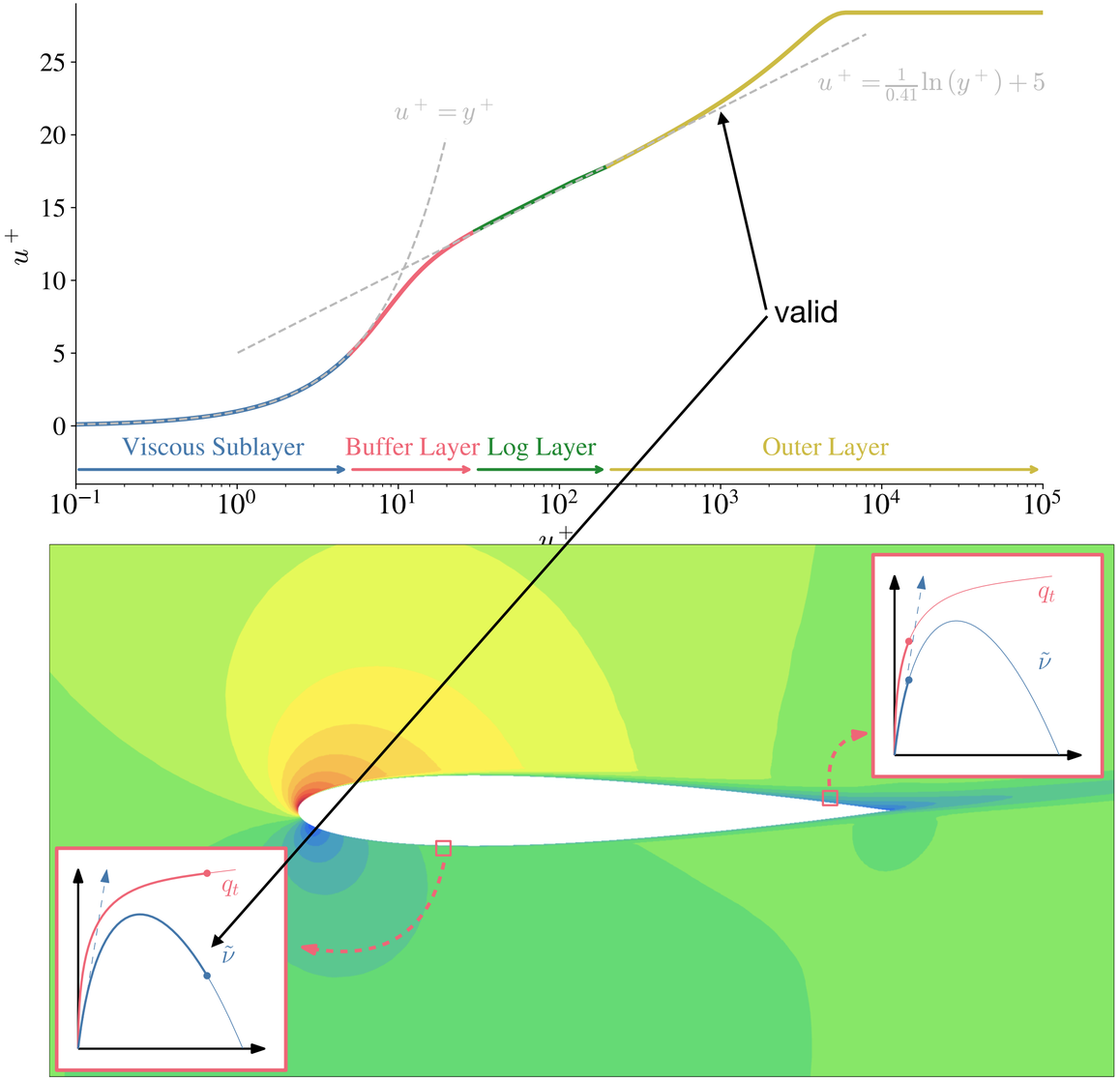
ODE wall model [3] reintroduces pressure momentum balance

$$\frac{\partial}{\partial \eta}\left[(\nu+\nu_t)\frac{\partial q_t}{\partial \eta}\right]=0$$

$$\tilde{\nu}=\kappa u_\tau \eta$$

$$\frac{\partial}{\partial \eta}\left[(\mu+\mu_t)\frac{\partial u}{\partial \eta}\right]=\left.\frac{\partial p}{\partial \xi}\right|_F+\psi(\eta)\rho\Big|_F\left[u_F\left.\frac{\partial u}{\partial \xi}\right|_F+v_F\left.\frac{\partial u}{\partial \eta}\right|_F\right]$$

$$\frac{\partial}{\partial \eta}\left[\frac{1}{\sigma}(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial \eta}\right]=-\frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial \eta}\right)^2-(P-D),$$

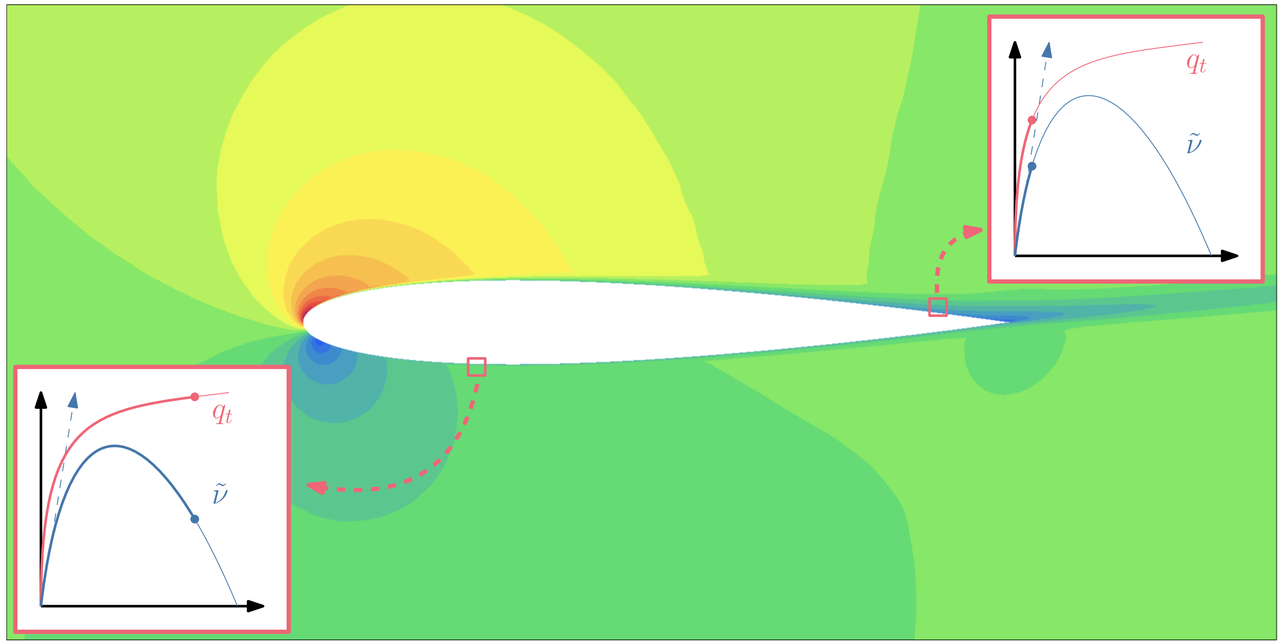


[3] Marsha J Berger and Michael J Aftosmis. An ODE-based wall model for turbulent flow simulations. AIAA Journal, 56(2):700–714, 2018

ODE solver must be accurate, efficient, and robust

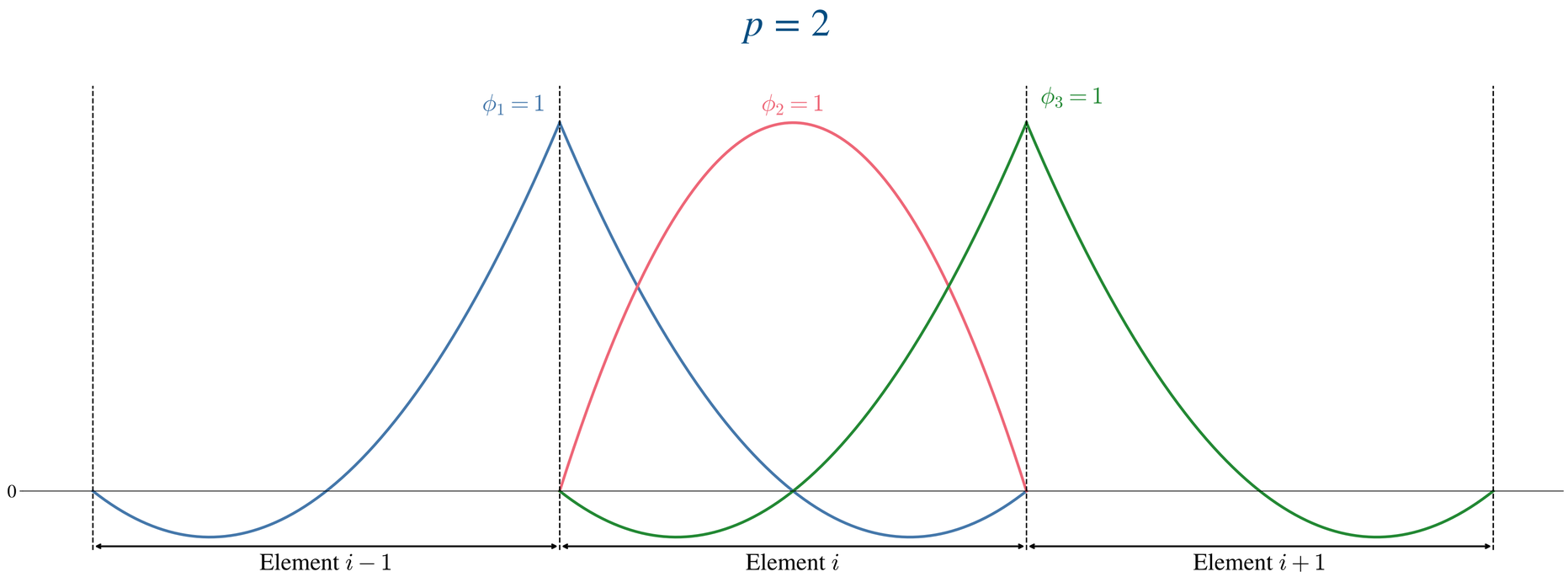
$$\frac{\partial}{\partial \eta}\left[(\mu+\mu_t)\frac{\partial u}{\partial \eta}\right] = \frac{\partial p}{\partial \xi}\bigg|_F + \psi(\eta)\rho\Big|_F\left[u_F\frac{\partial u}{\partial \xi}\bigg|_F + v_F\frac{\partial u}{\partial \eta}\bigg|_F\right]$$

$$\frac{\partial}{\partial \eta}\left[\frac{1}{\sigma}(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial \eta}\right] = -\frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial \eta}\right)^2 - (P-D),$$



I developed a robust 1D continuous Galerkin ODE solver with mesh optimization to efficiently solve wall models.

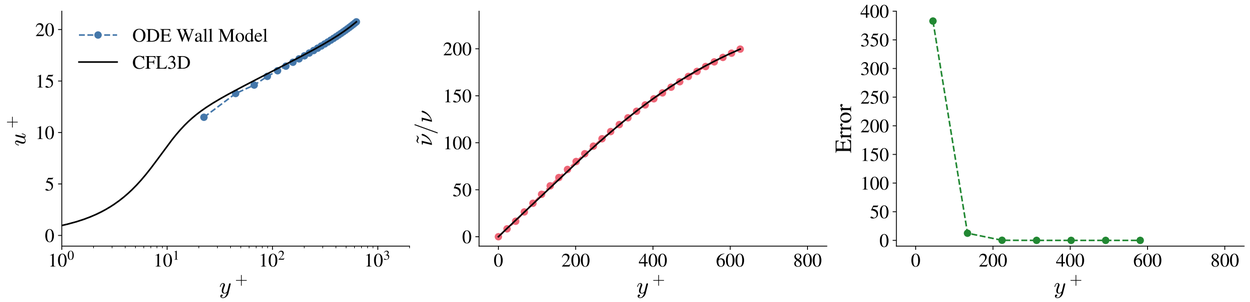
ODEs are solved using a high-order continuous Galerkin discretization with a Newton—Krylov solver



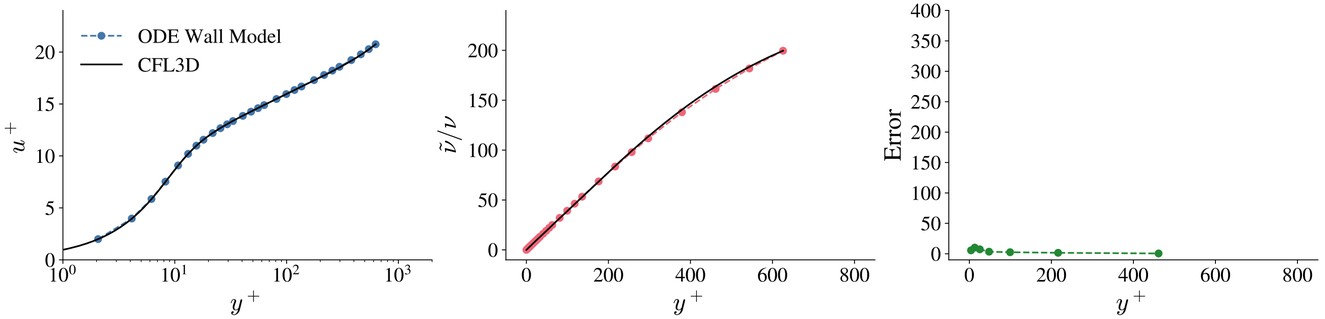
ODE wall model mesh are optimized using an adjoint weighted residual

Wall-Tangential Velocity | Turbulence Variable | Error Metric

Initial Mesh



Final Optimized Mesh



ODEs scale the source term strength to ensure solutions are found

$$\beta \in [0,1]$$

$$\frac{\partial}{\partial \eta}\left[(\mu+\mu_t)\frac{\partial u}{\partial \eta}\right] = \beta\left(\left.\frac{\partial p}{\partial \xi}\right|_F + \psi(\eta)\rho\Big|_F\left[u_F\left.\frac{\partial u}{\partial \xi}\right|_F + v_F\left.\frac{\partial u}{\partial \eta}\right|_F\right]\right)$$

$$\frac{\partial}{\partial \eta}\left[\frac{1}{\sigma}(\nu+\tilde{\nu})\frac{\partial \tilde{\nu}}{\partial \eta}\right] = -\frac{c_{b2}}{\sigma}\left(\frac{\partial \tilde{\nu}}{\partial \eta}\right)^2 - (P-D),$$

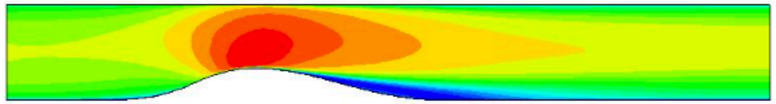
Outline

- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance function
- ODE wall model
- NASA TMR test cases
- Conclusions

The NASA turbulence modeling resource is the database for turbulence model verification

Langley Research Center

Turbulence Modeling Resource



Turbulence Modeling Resource

Purpose

The purpose of this site is to provide a central location where Reynolds-averaged Navier-Stokes (RANS) turbulence models are documented. This effort is guided by the Turbulence Model Benchmarking Working Group (TMBWG), a working group of the Fluid Dynamics Technical Committee of the American Institute of Aeronautics and Astronautics (AIAA).

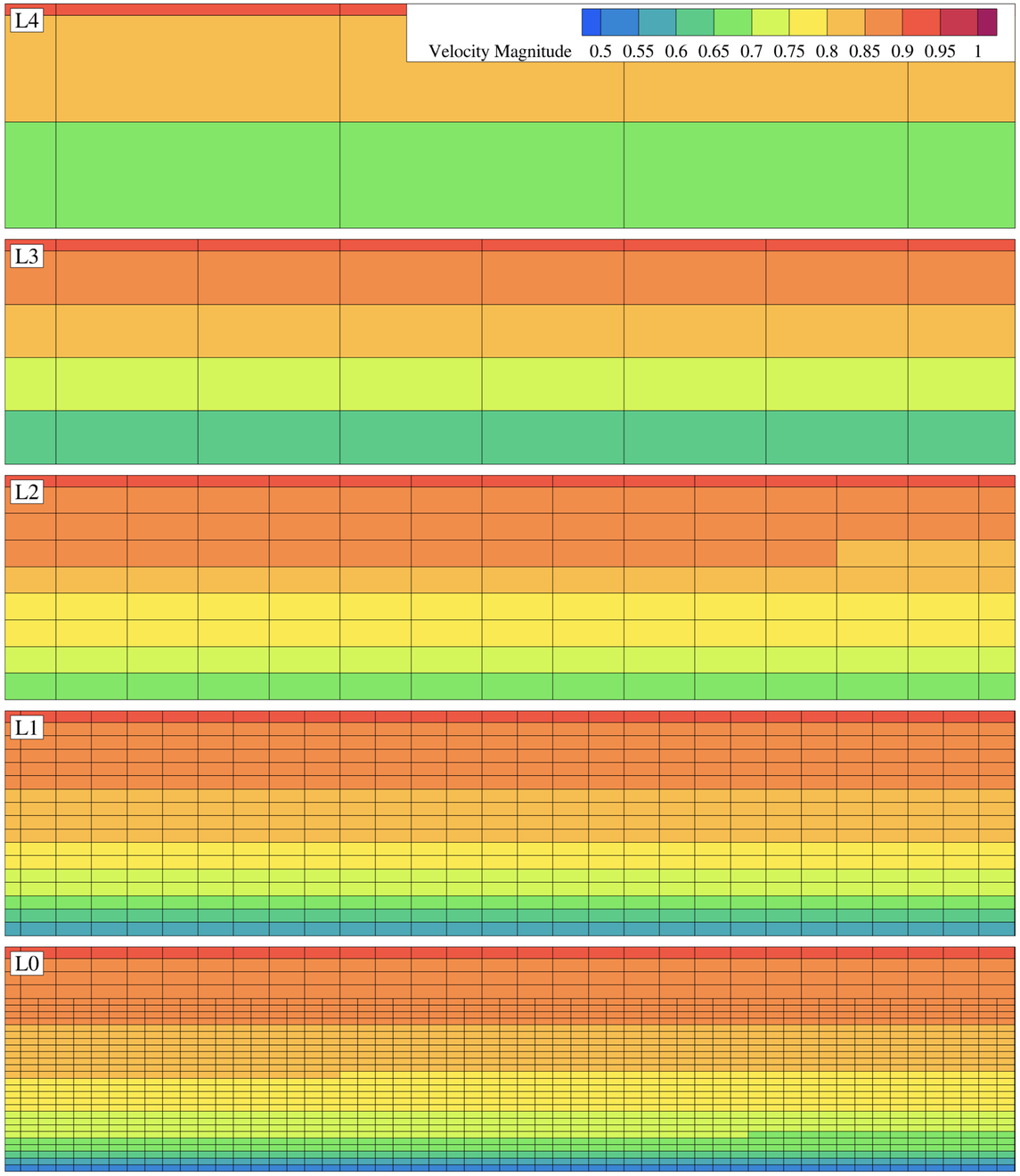
The objective is to provide a resource for CFD developers to:

- obtain accurate and up-to-date information on widely-used RANS turbulence models, and
- verify that models are implemented correctly.

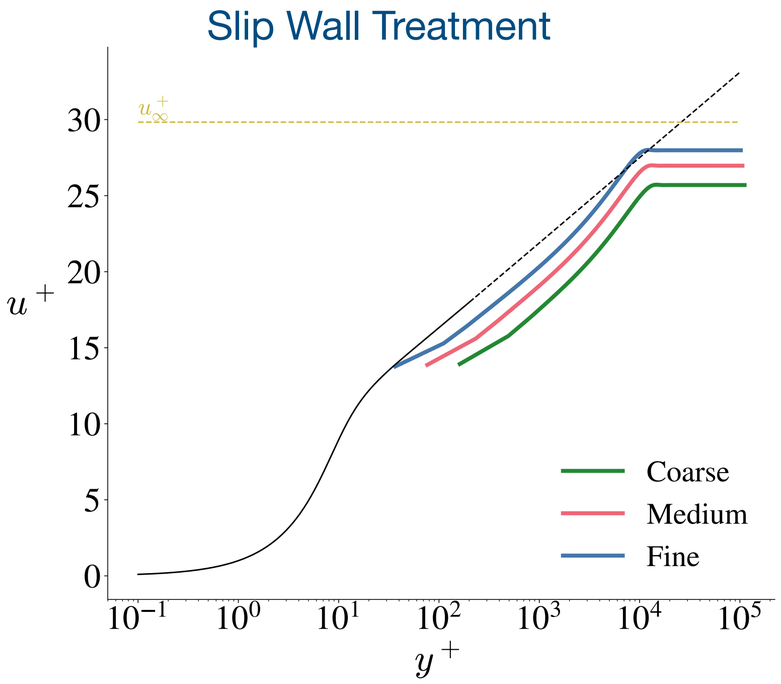
https://turbmodels.larc.nasa.gov/index.html

Grid Aligned Finite Flat Plate

- Mach 0.2
- Reynolds number 10^7
- Finest mesh has approximately 640k cells, off wall spacing of 10^{-4}



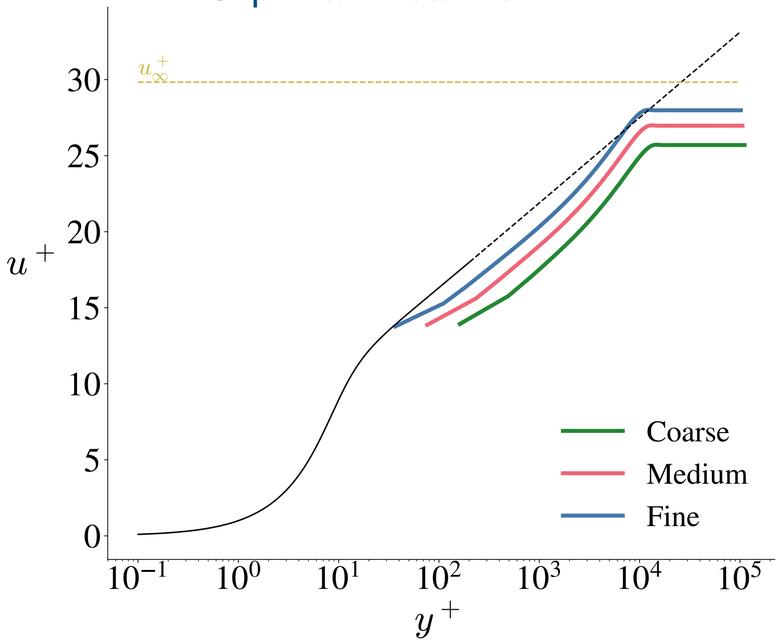
Grid Aligned Plate Velocity Profiles



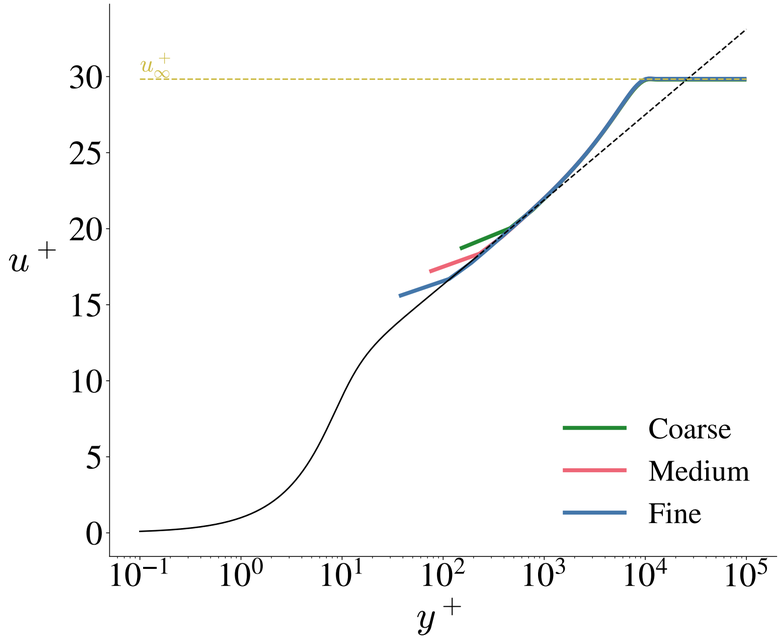
$$Re_x = 2 \times 10^7$$

Grid Aligned Plate Velocity Profiles

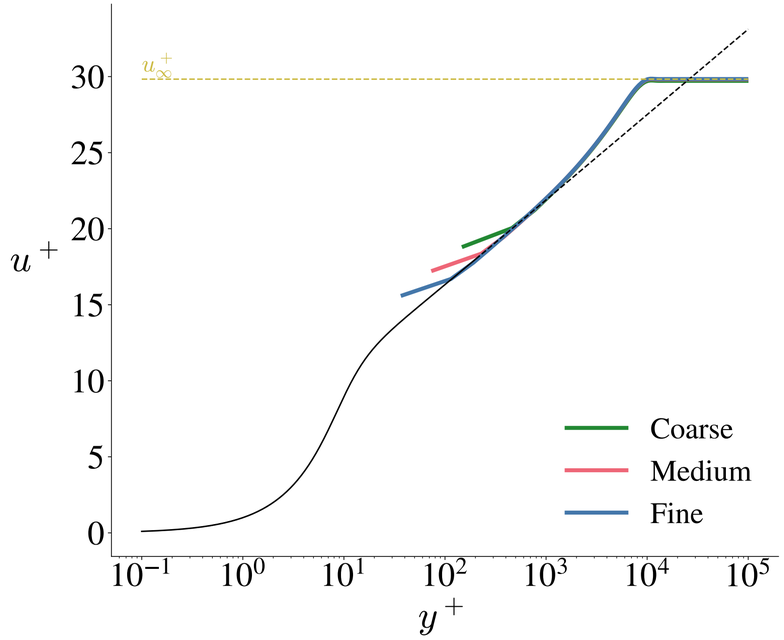
Slip Wall Treatment



SA Wall Function



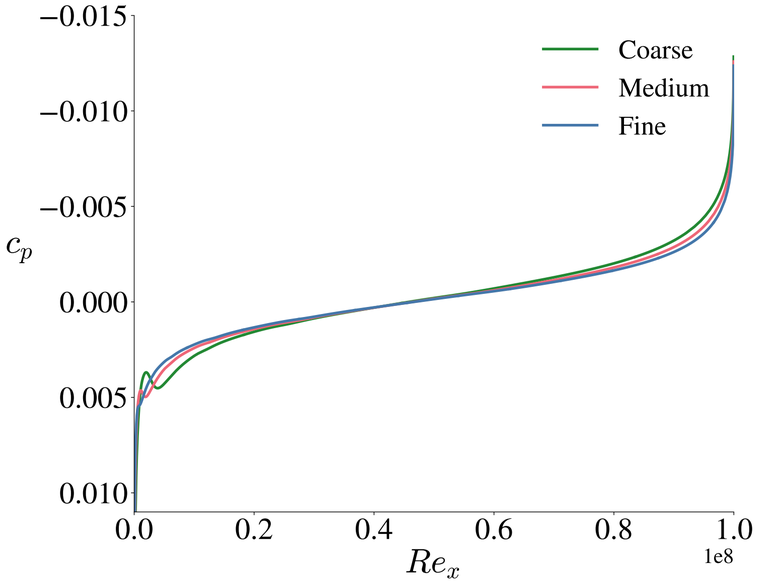
ODE Wall Model



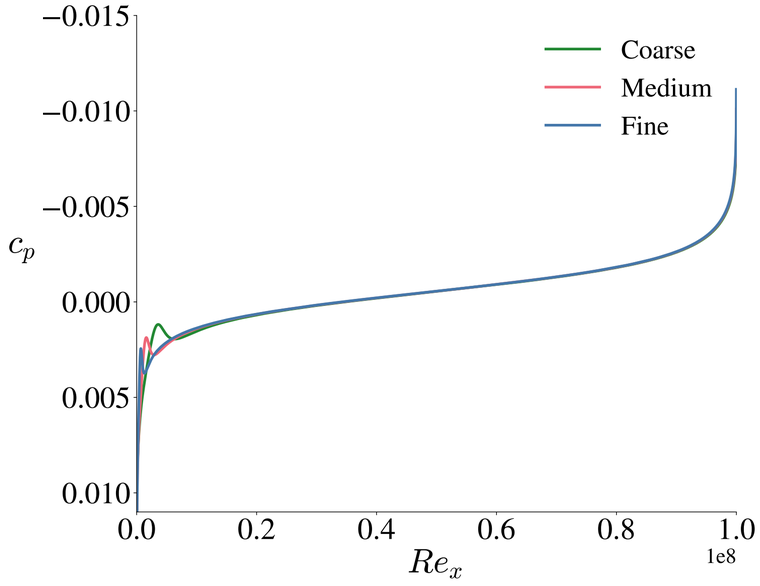
$$Re_x = 2 \times 10^7$$

Grid Aligned Flat Plate Pressure Coefficients

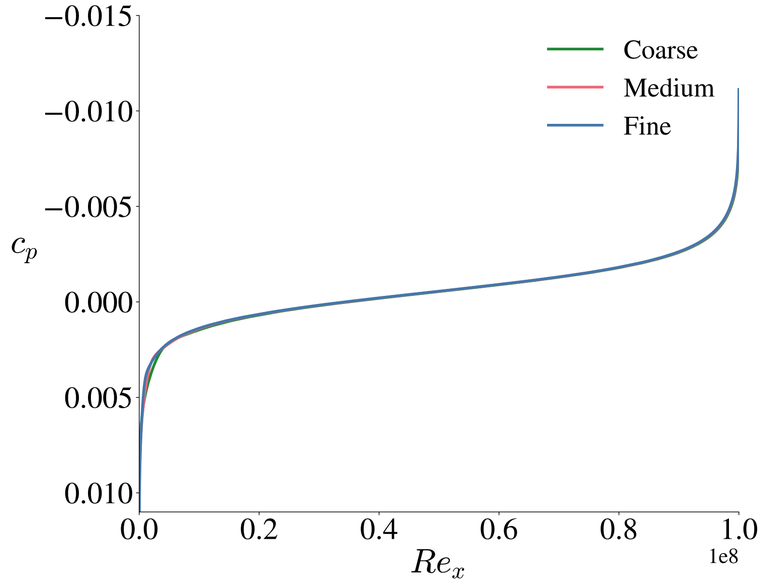
Slip Wall Treatment



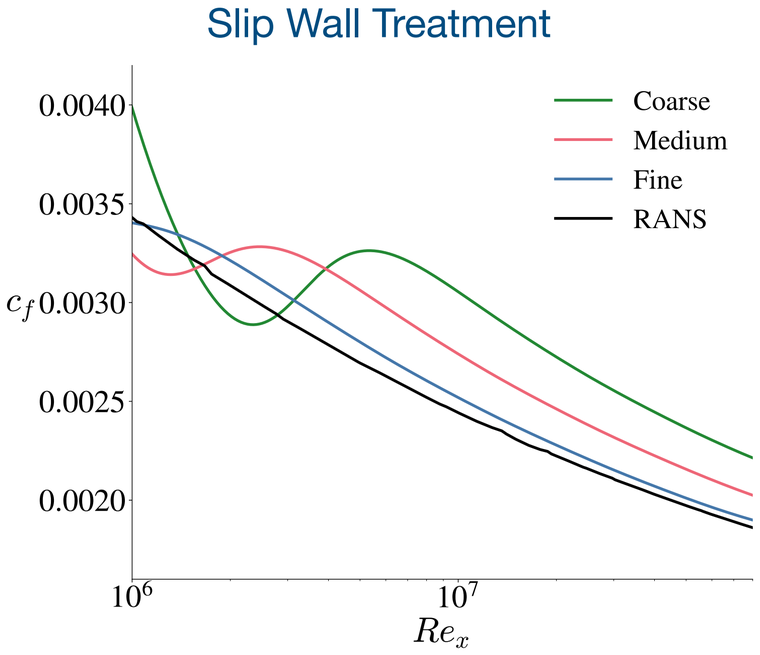
SA Wall Function



ODE Wall Model

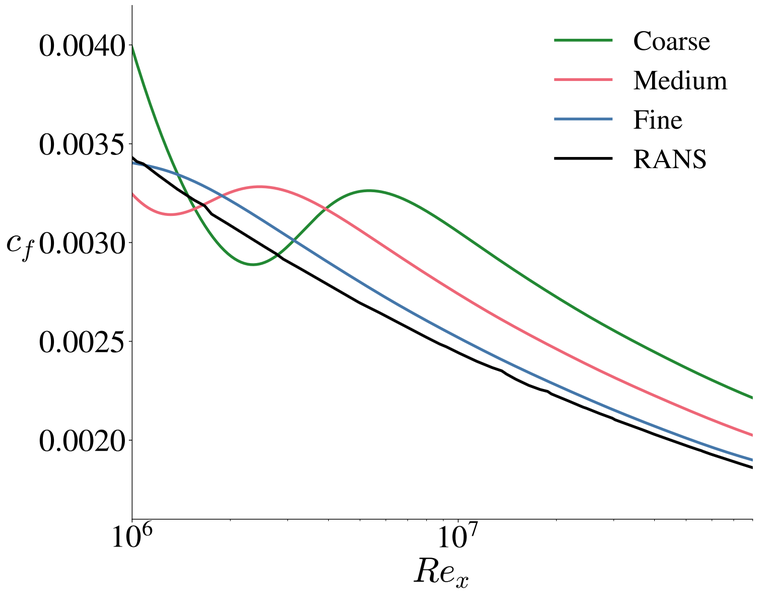


Slip wall treatment demonstrates mesh converging skin frictions

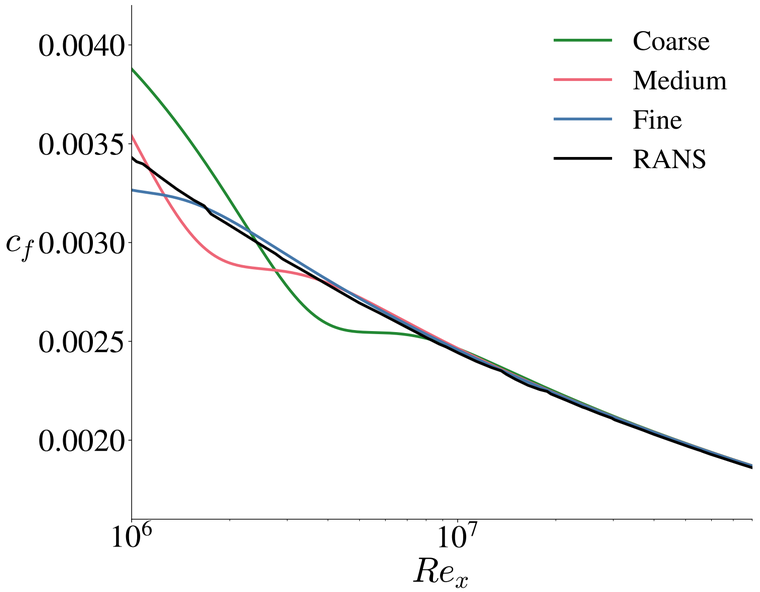


Forcing point methods demonstrate asymptotically converged skin friction regardless of mesh level

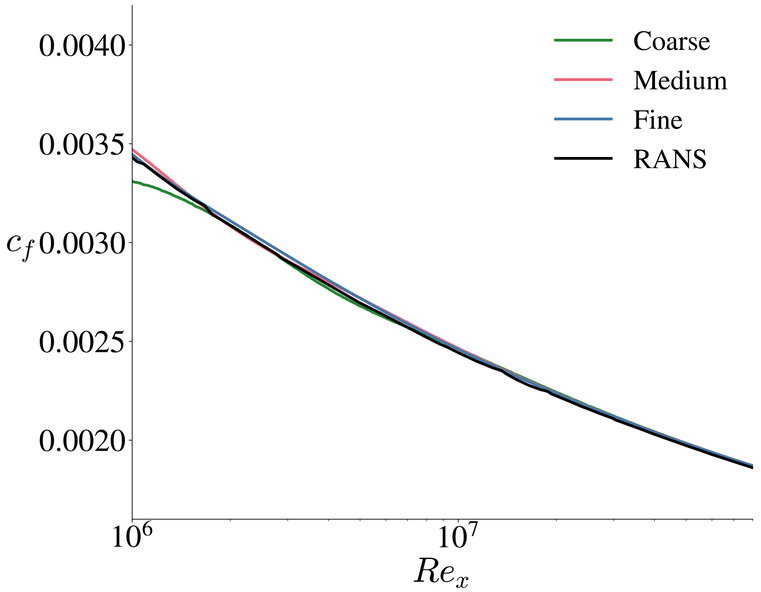
Slip Wall Treatment



SA Wall Function

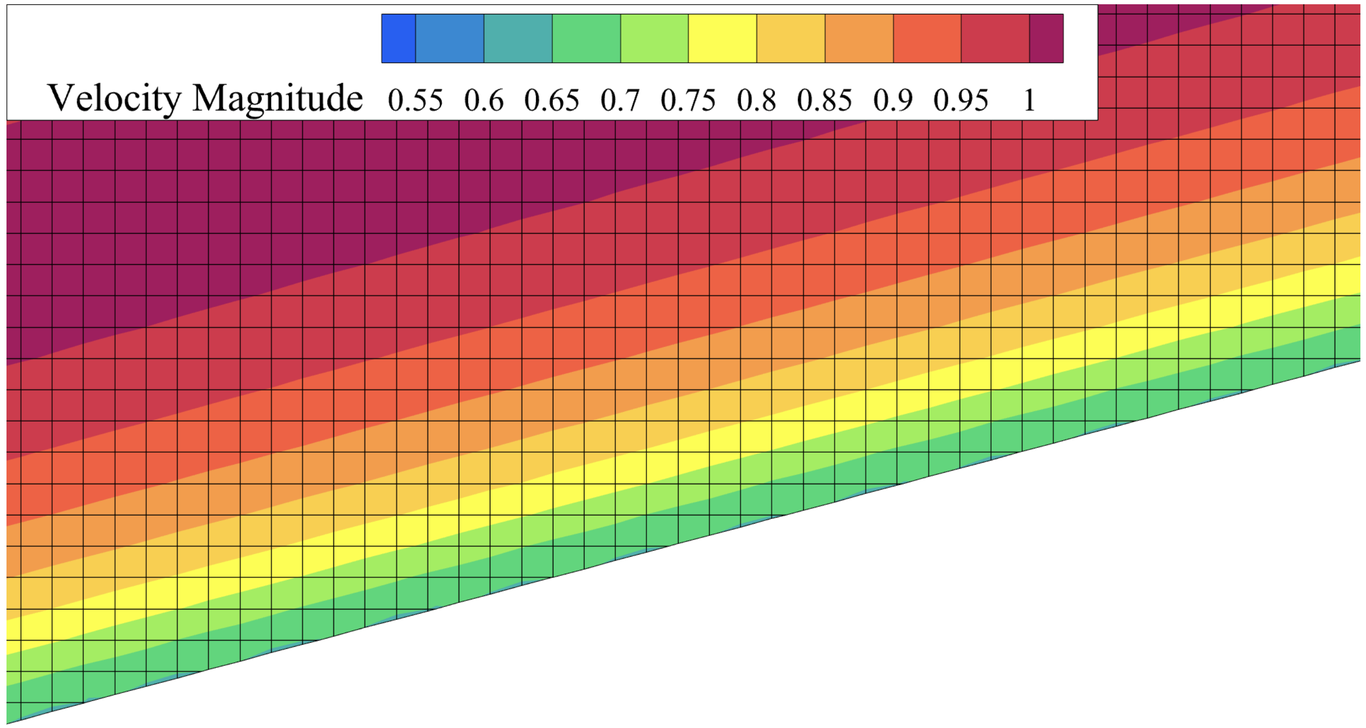


ODE Wall Model

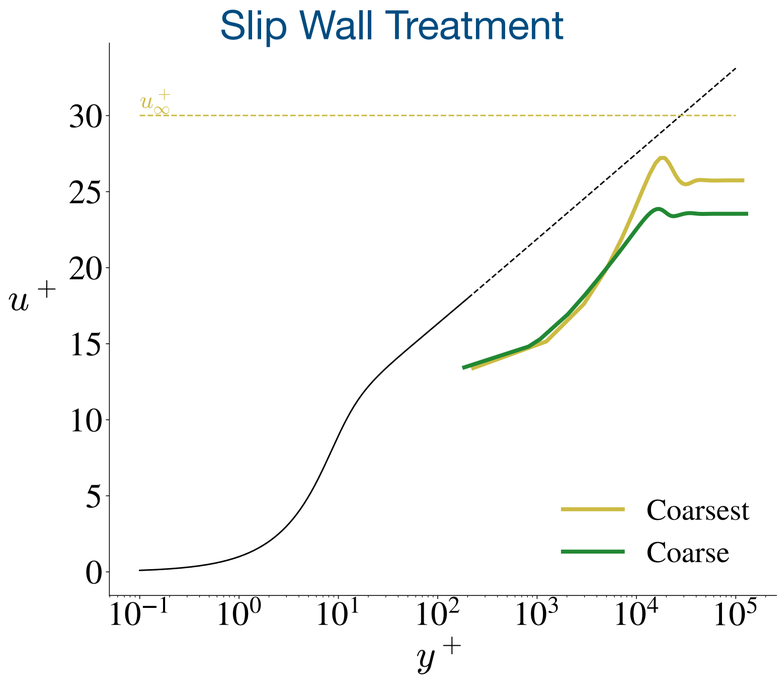


Rotated Finite Flat Plate

- Mach 0.2
- Reynolds number 10^7
- L0 Mesh has approximately 640k cells, off wall spacing of 10^{-4}



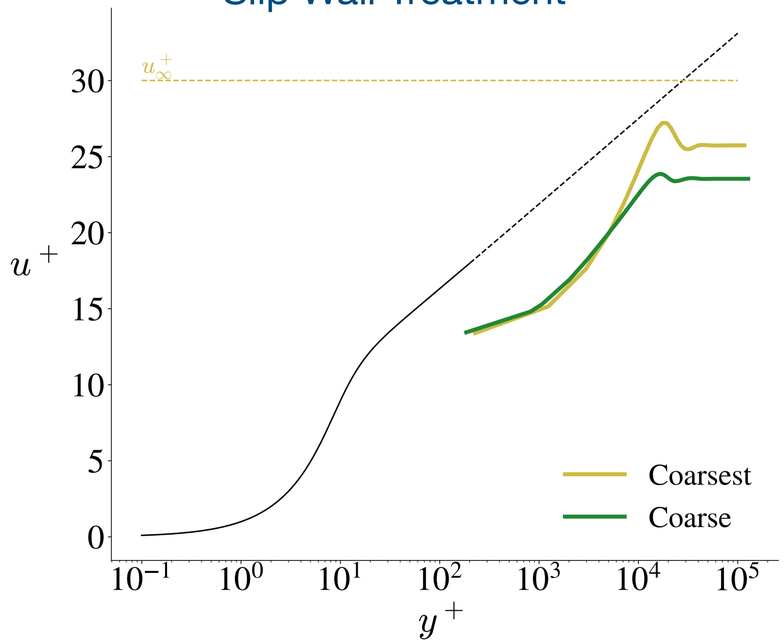
Slip wall treatment lacks mesh convergence



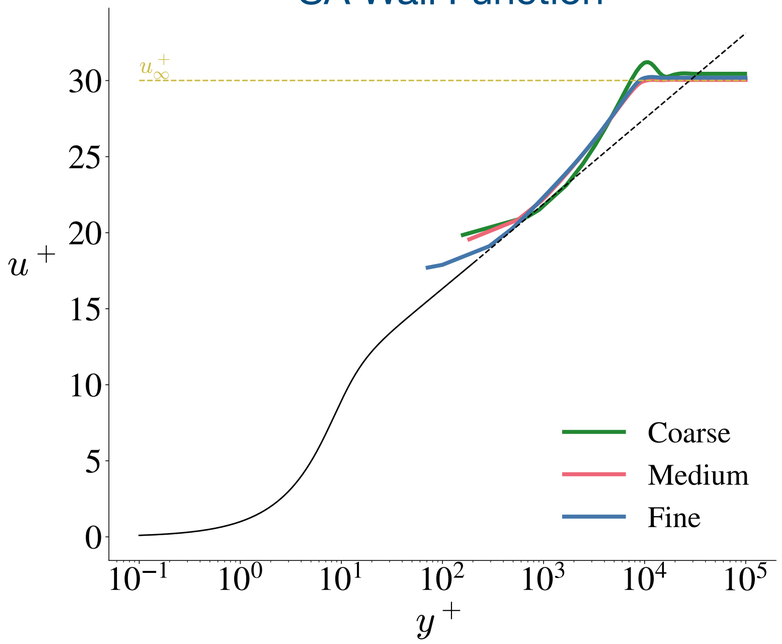
$$Re_x = 2 \times 10^7$$

Forcing point methods exhibit viscous overshoot on coarsest mesh

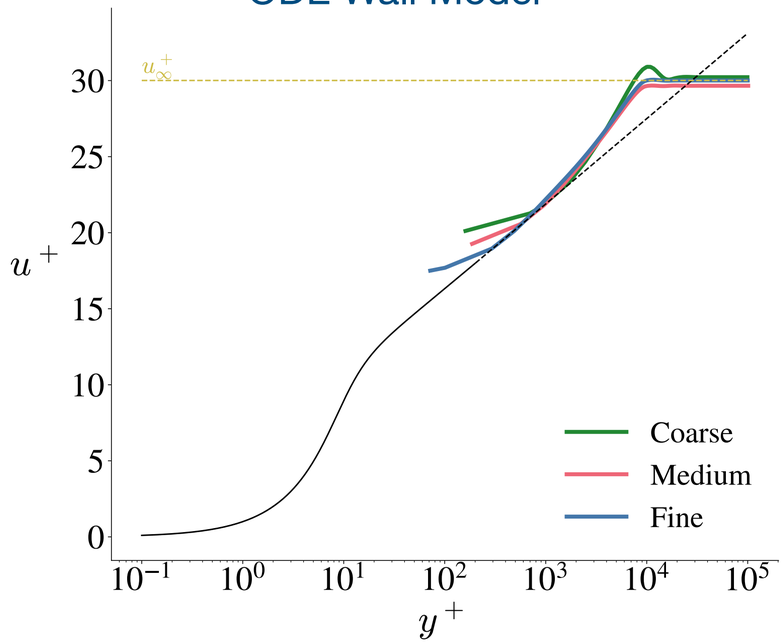
Slip Wall Treatment



SA Wall Function

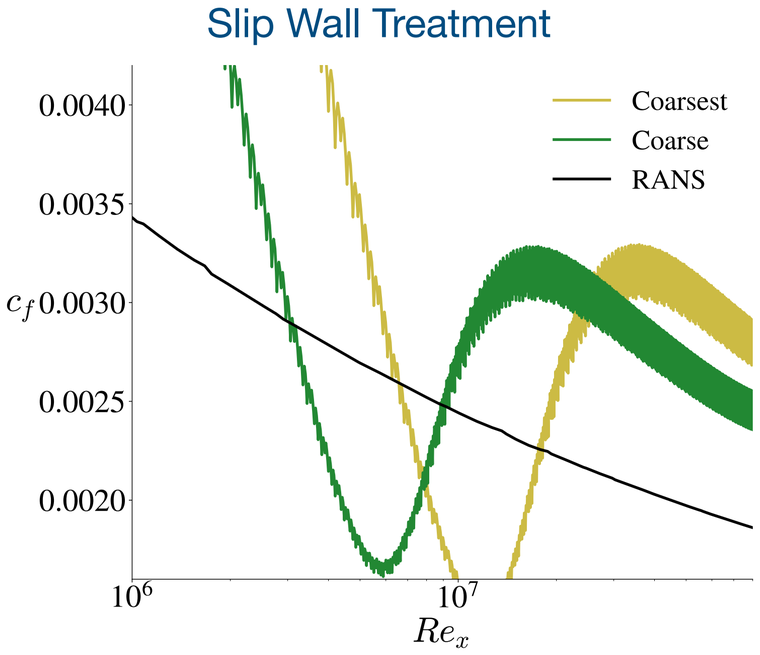


ODE Wall Model



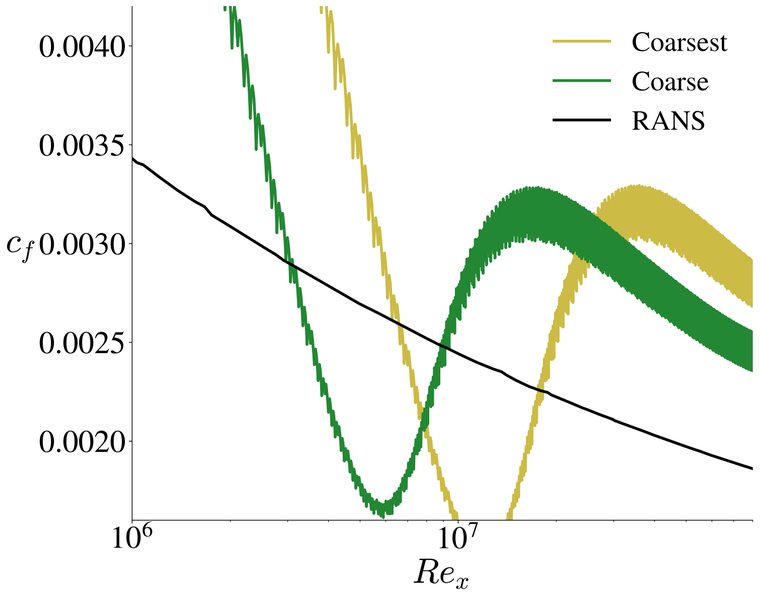
$$Re_x = 2 \times 10^7$$

Slip wall treatment generates noisy skin frictions

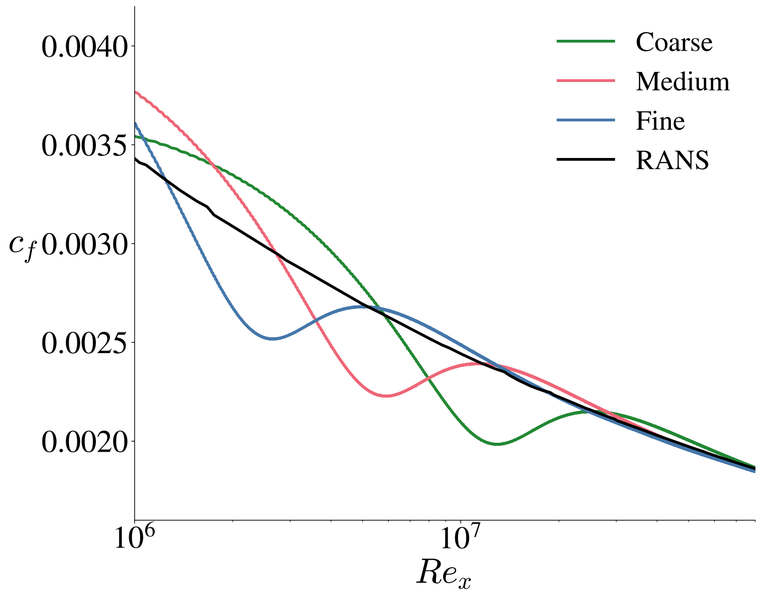


SA wall function produces smooth skin frictions

Slip Wall Treatment

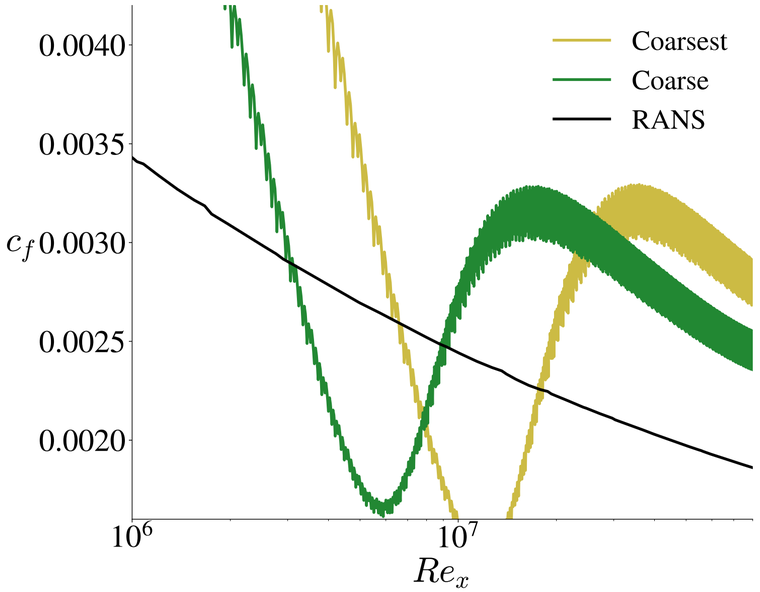


SA Wall Function

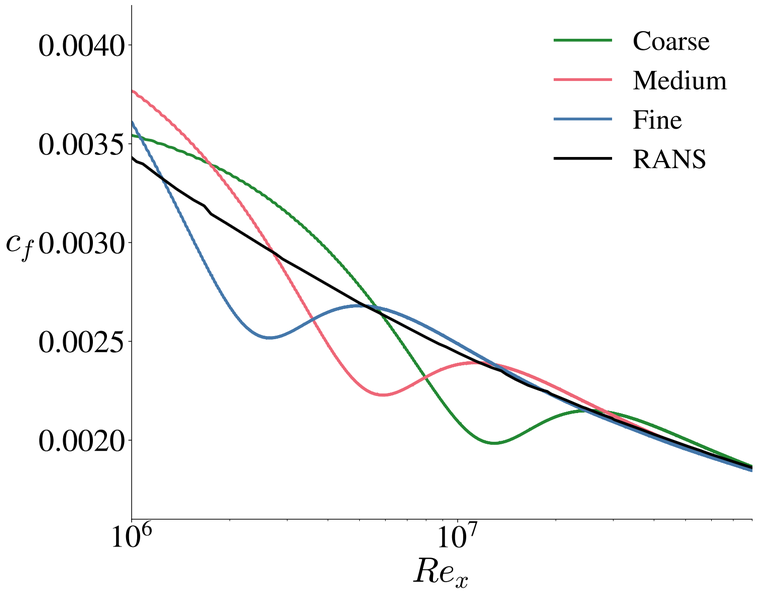


ODE wall model introduces some noise

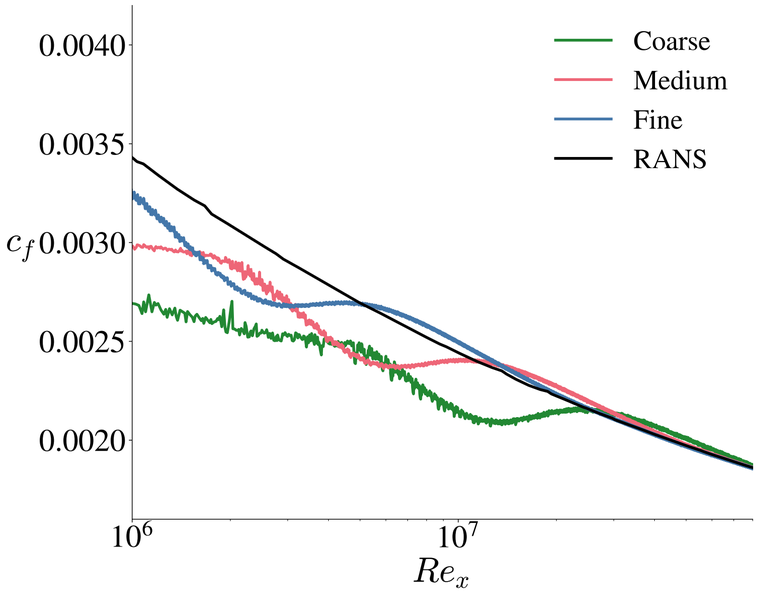
Slip Wall Treatment



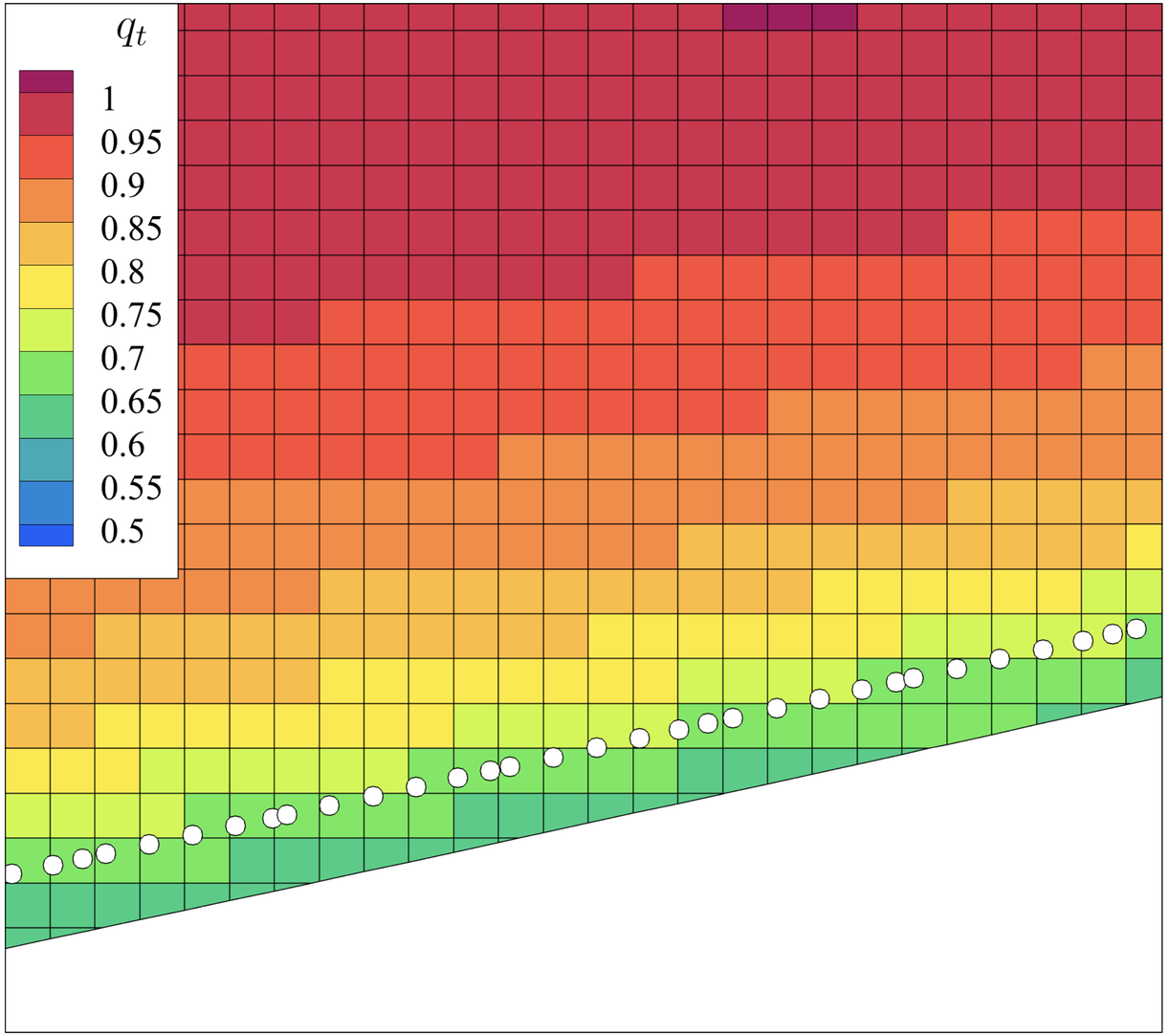
SA Wall Function

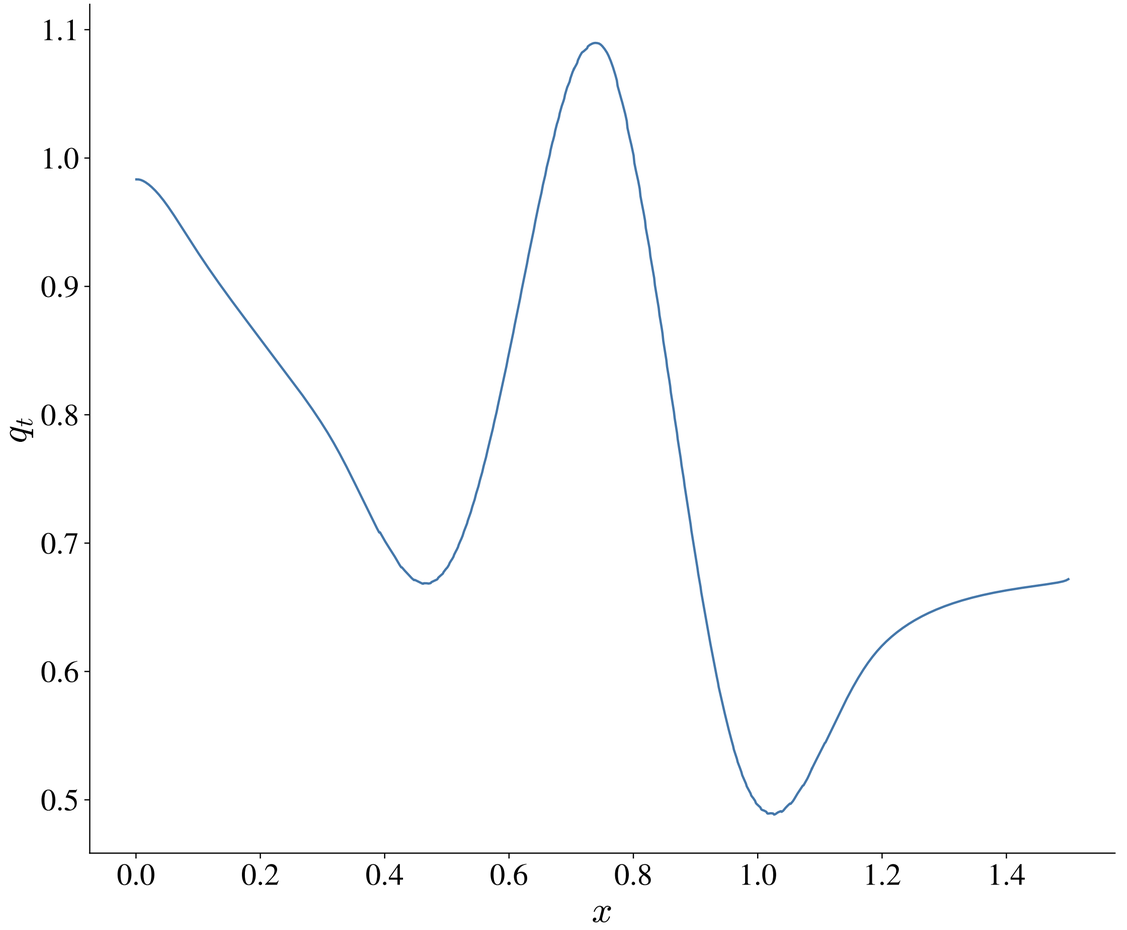


ODE Wall Model

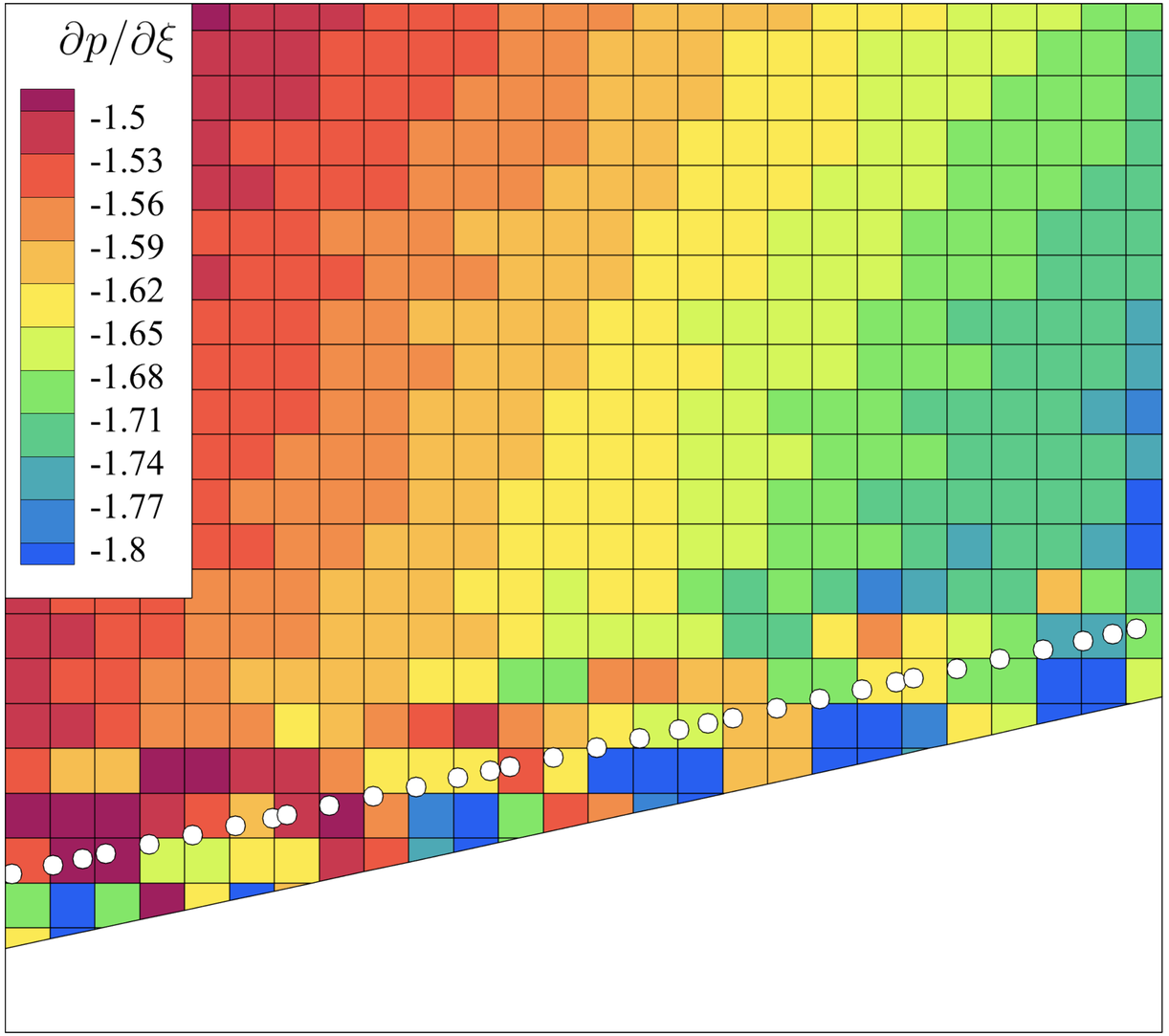


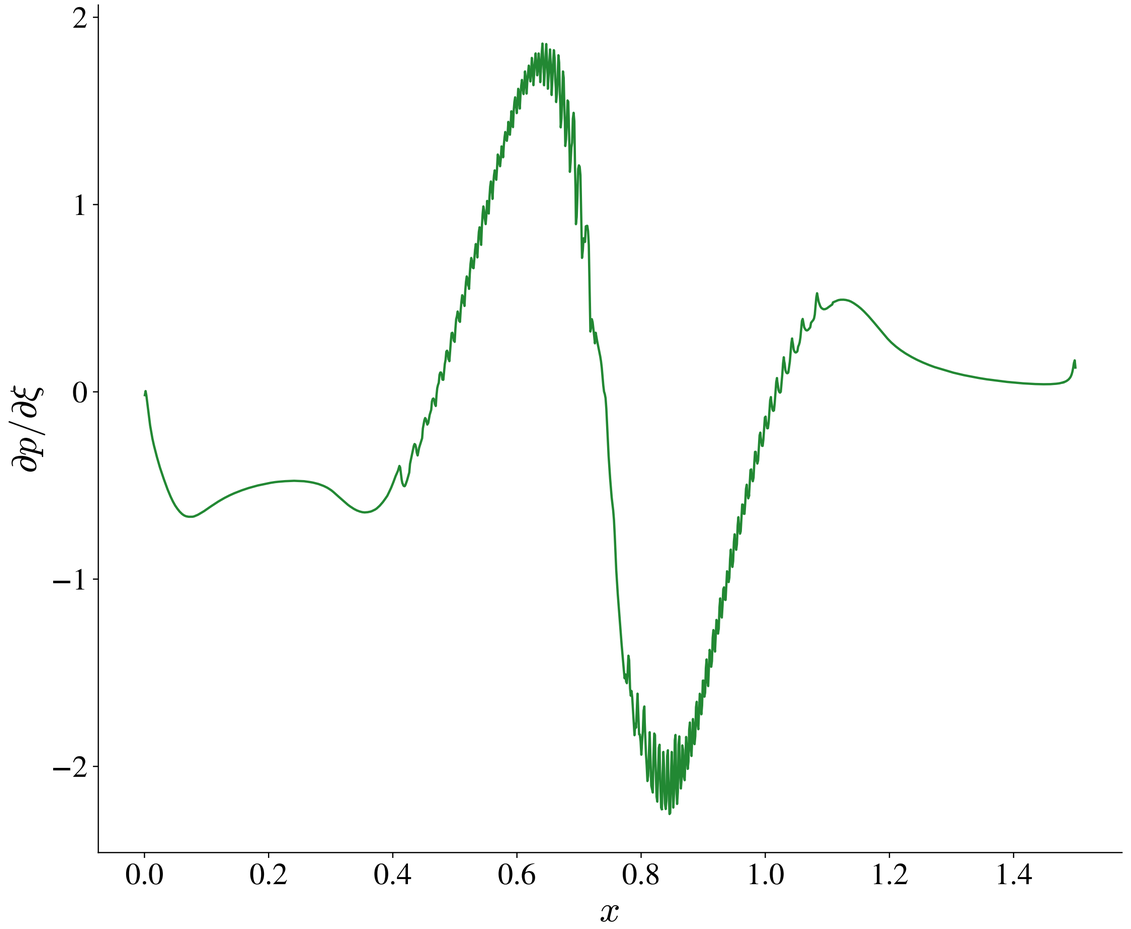
Tangential Velocity is Interpolated Smoothly





Pressure Oscillations affect Skin Frictions





Gradient Quantities use a Different Interpolator

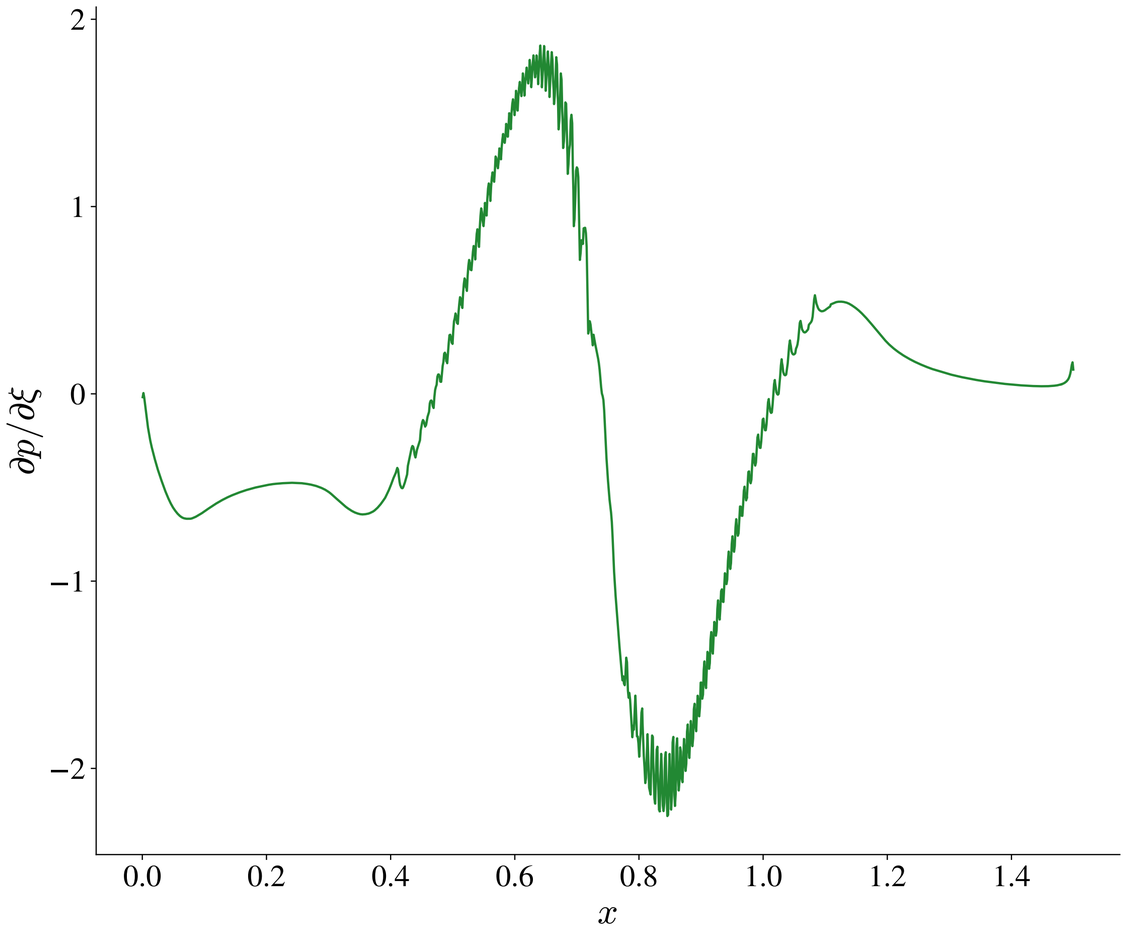
Primal Quantities

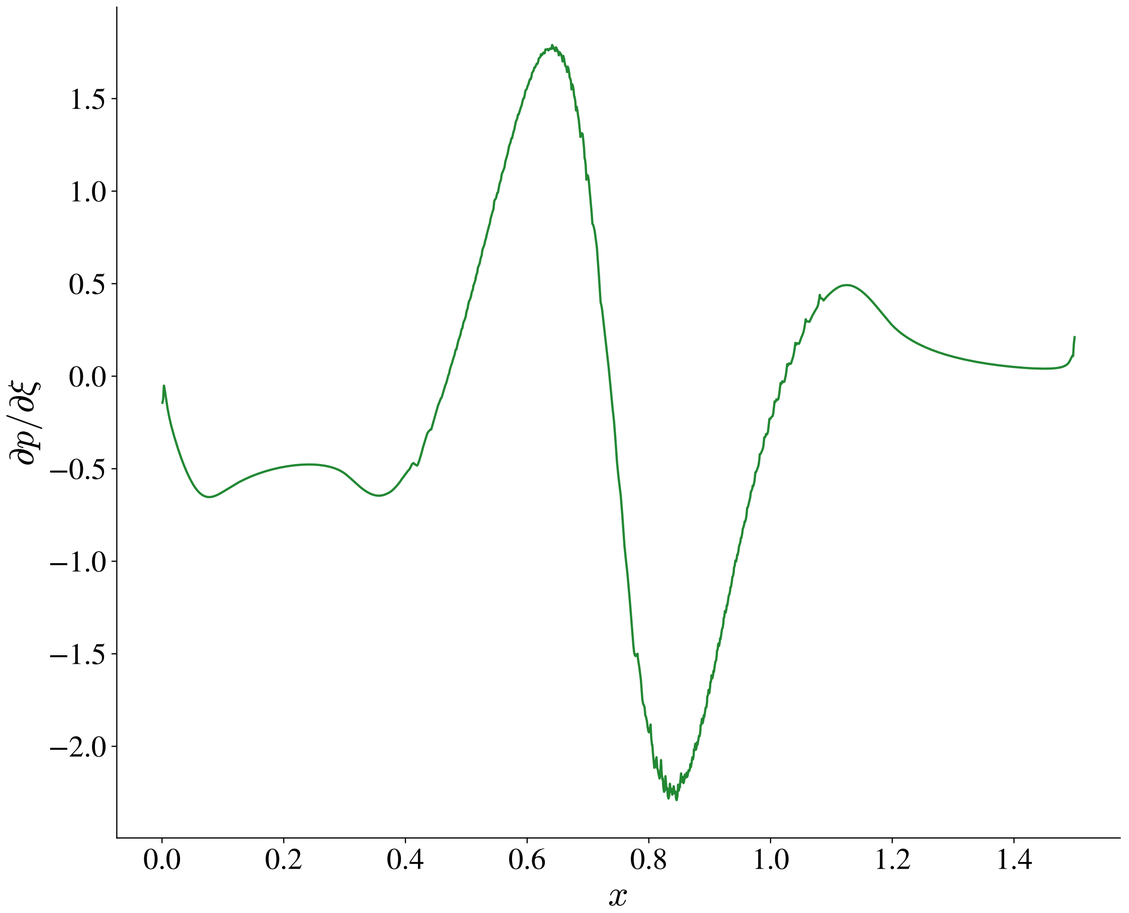
- Quadratic fit in the wall normal direction
- Linear fit in the wall tangential direction
- Uses all cells within stencil

Gradient Quantities

- Linear fit in the wall normal direction
- Linear fit in the wall tangential direction
- Does not use cut cells in stencil

Gradient Quantities use a Different Interpolator

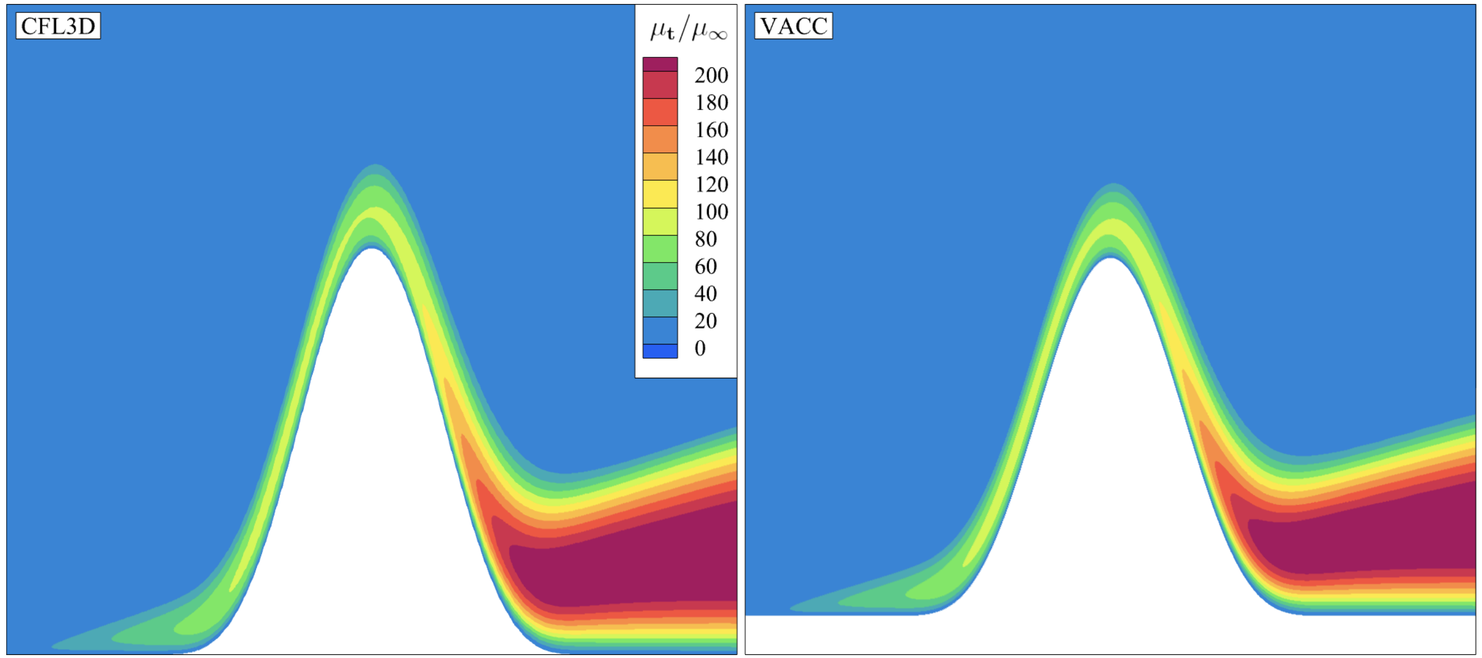




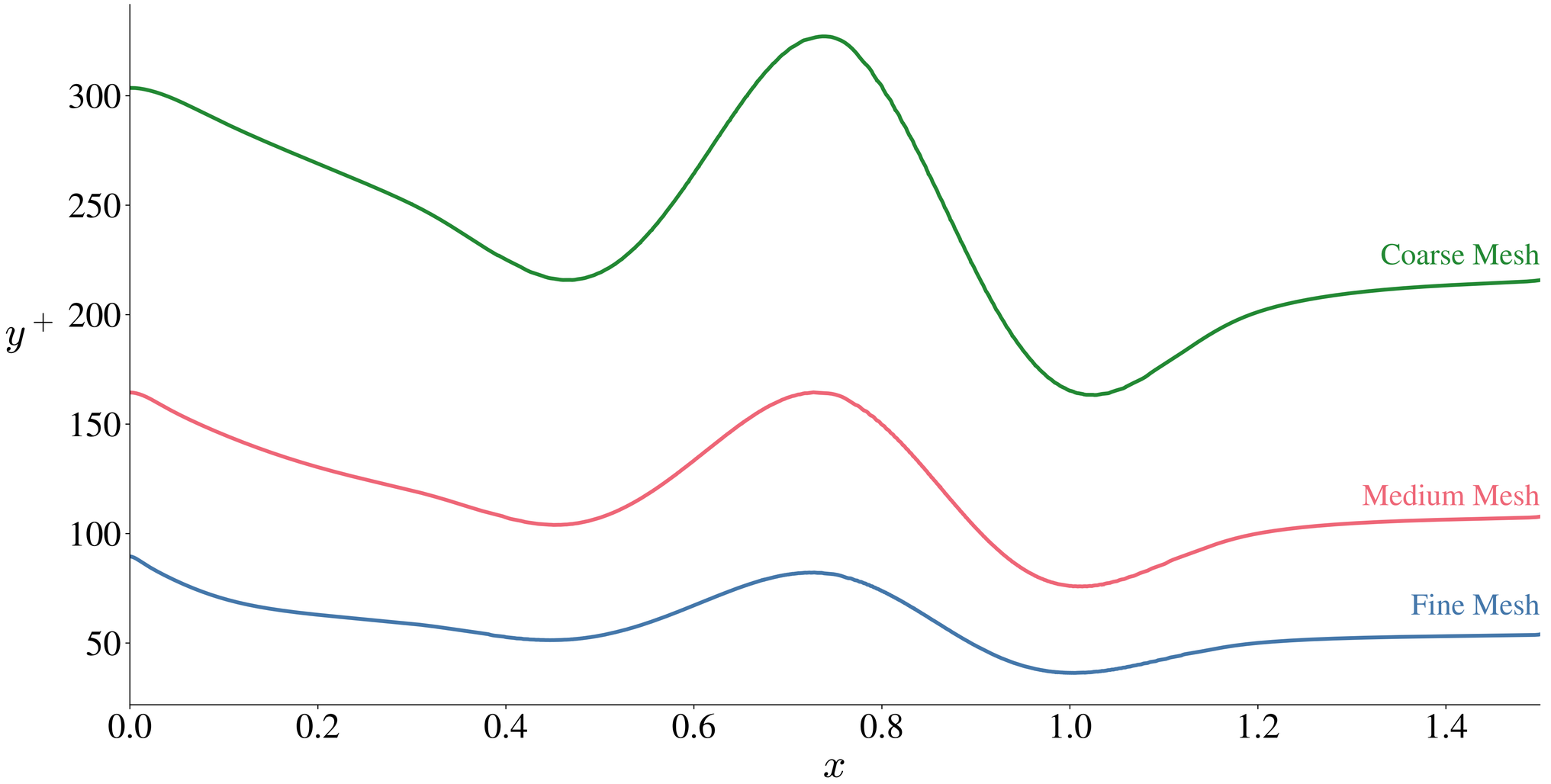
The slip wall treatment has no way to handle the irregular cut cells and is left out for the remainder of the test cases.

2D Turbulent Bump

- Introduces pressure gradient
- Mach 0.2
- Reynolds number 3×10^6

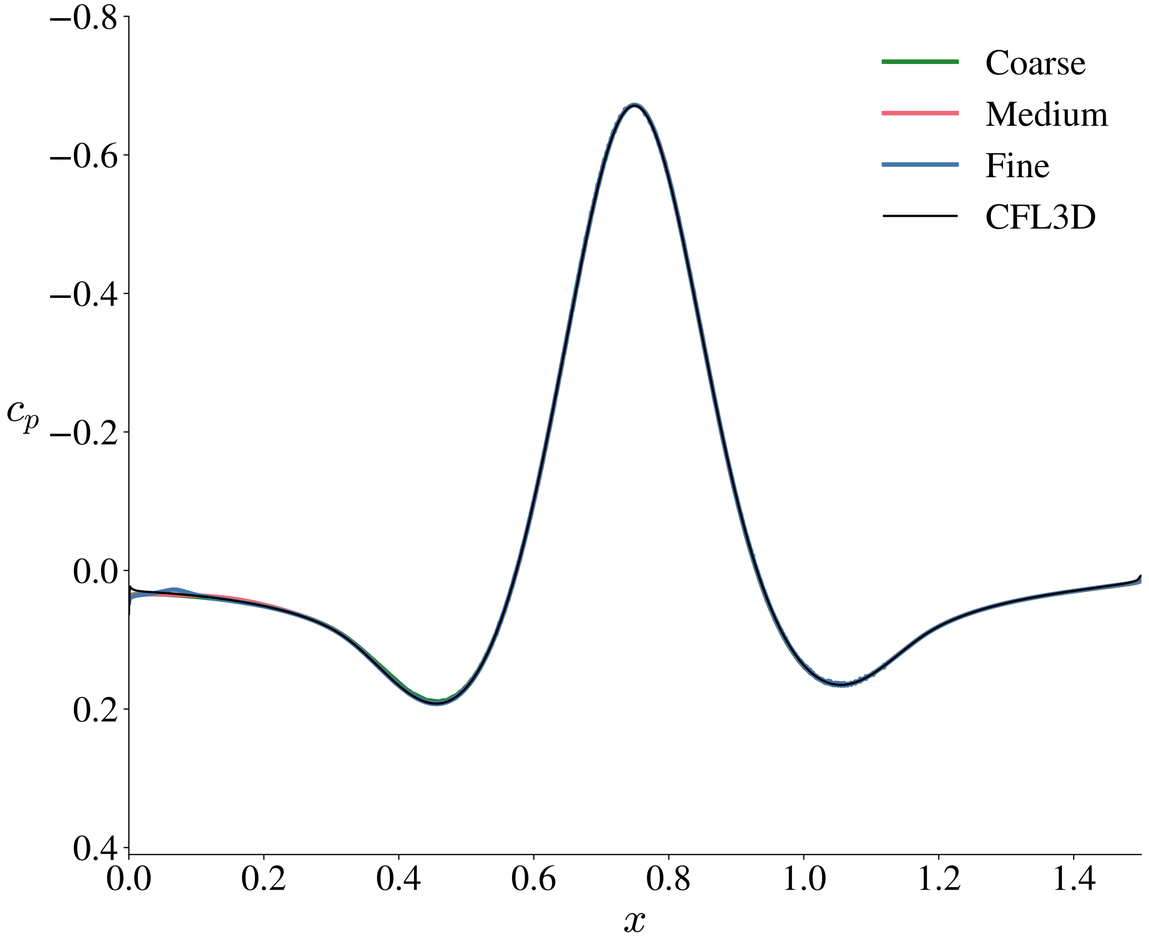


Bump Forcing Point y+

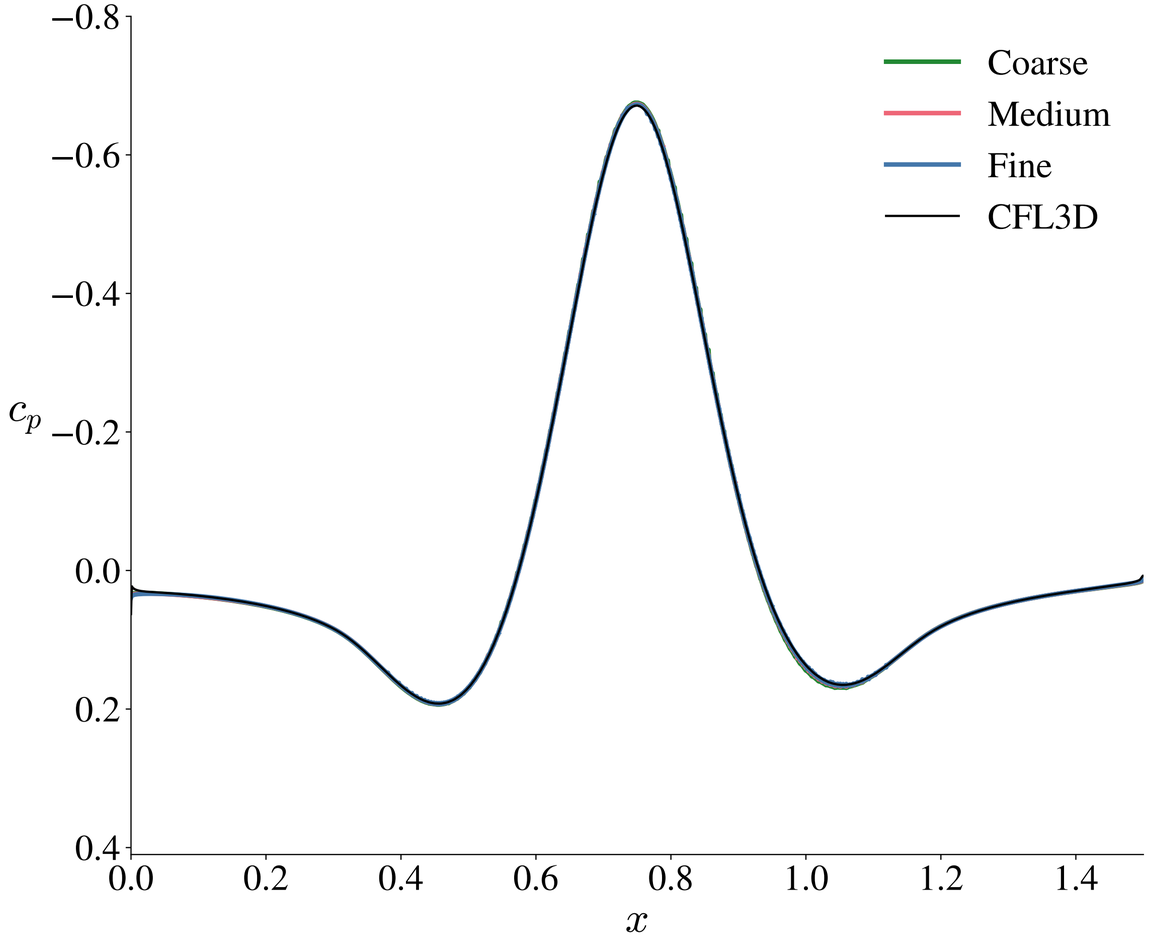


Both forcing point approaches match surface pressures on all meshes

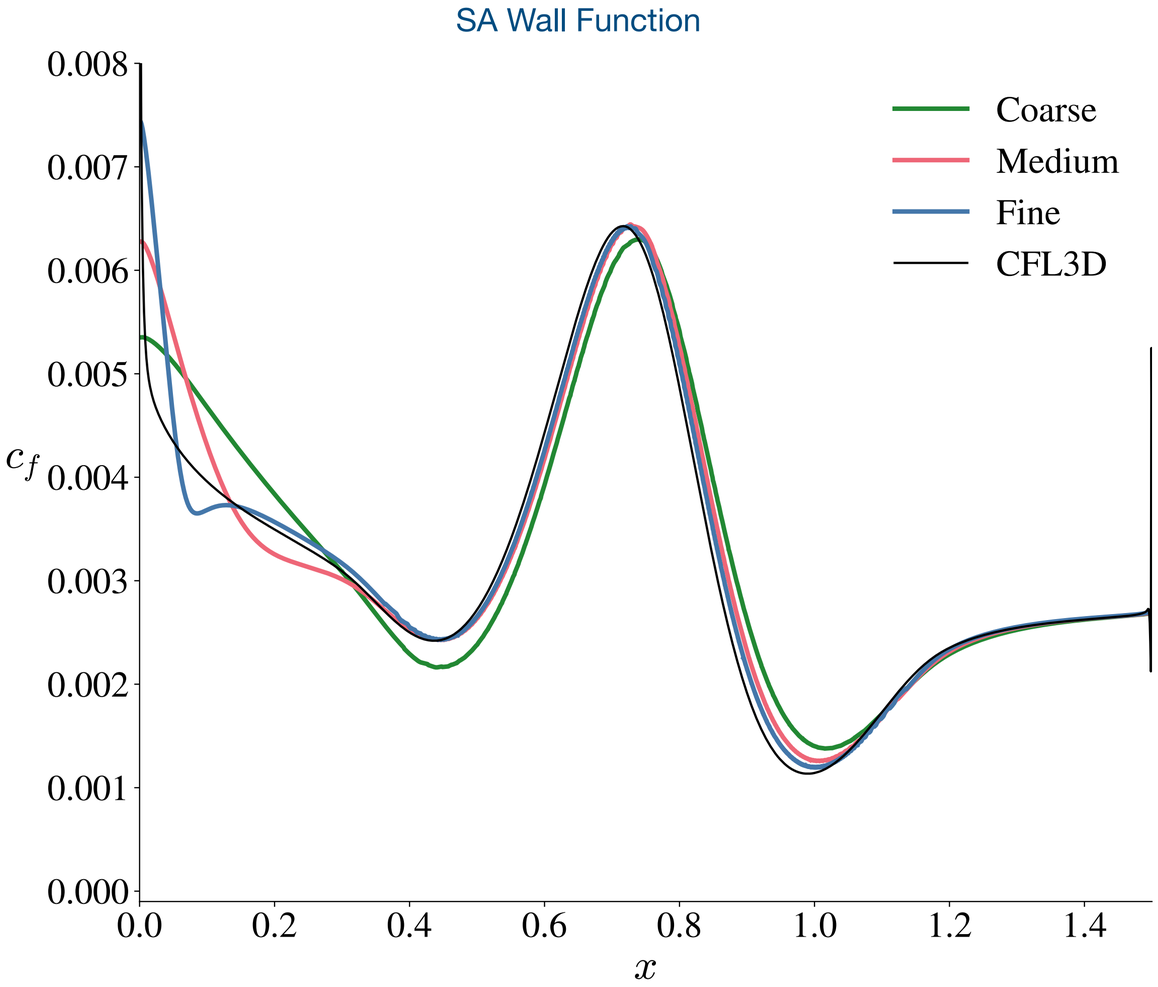
SA Wall Function



ODE Wall Model

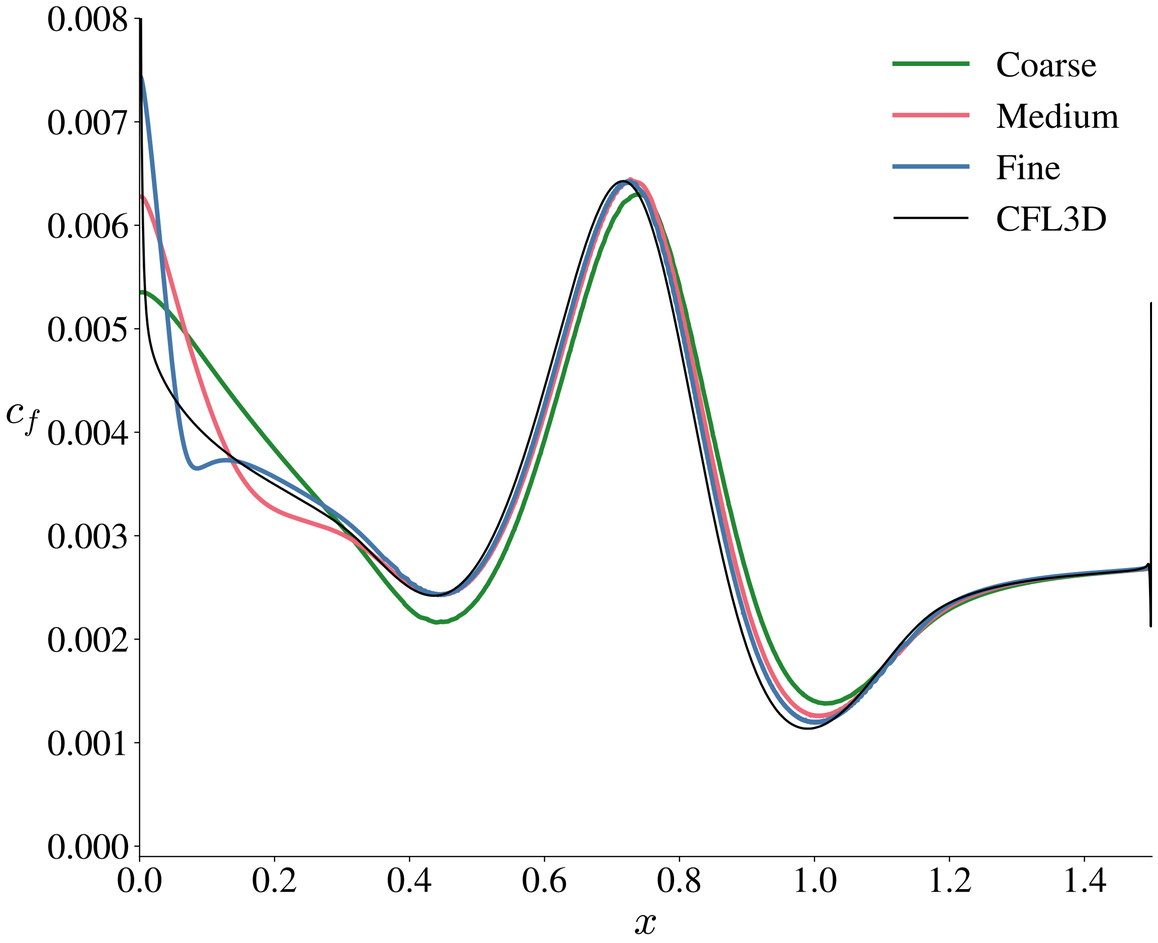


SA Wall function skin frictions exhibits trouble with boundary layer start up

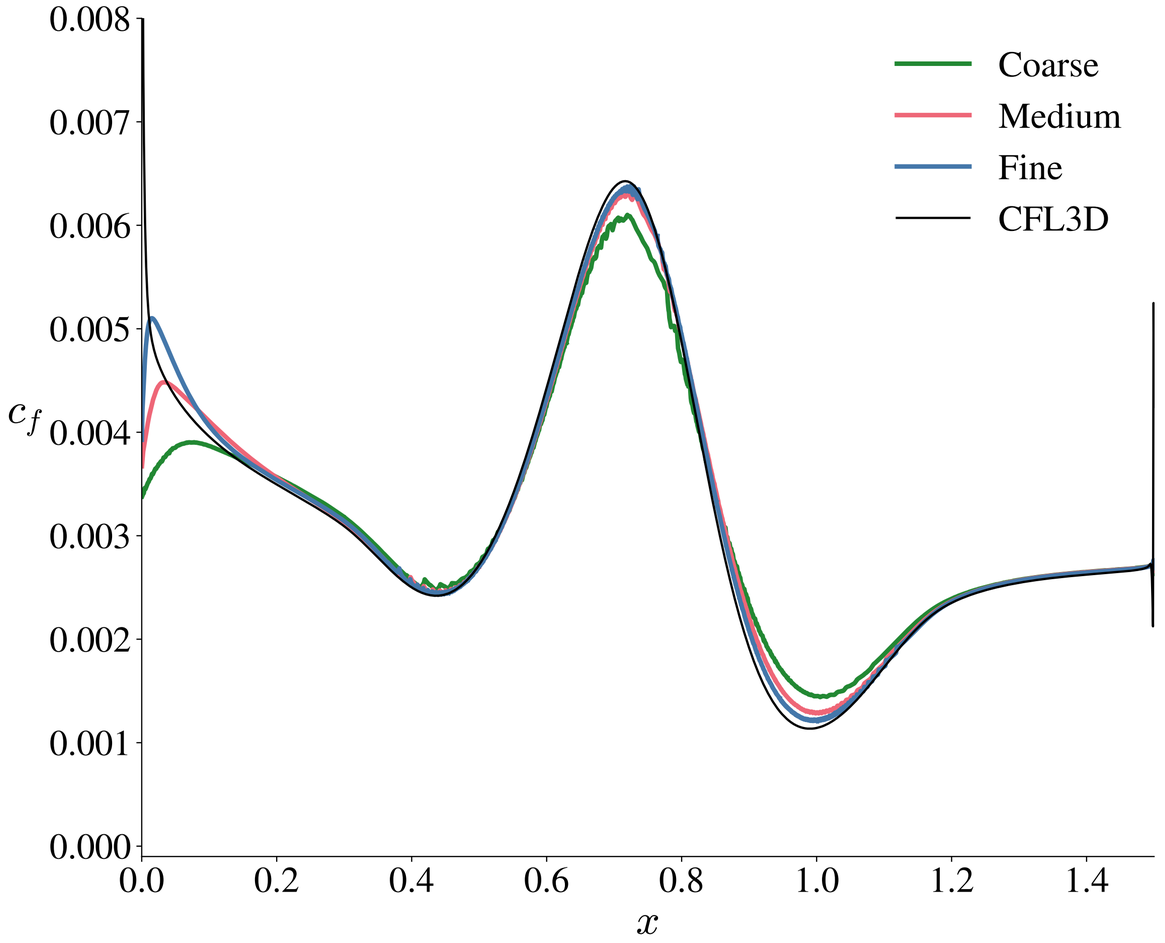


ODE wall model provides more accurate skin frictions on coarser meshes

SA Wall Function

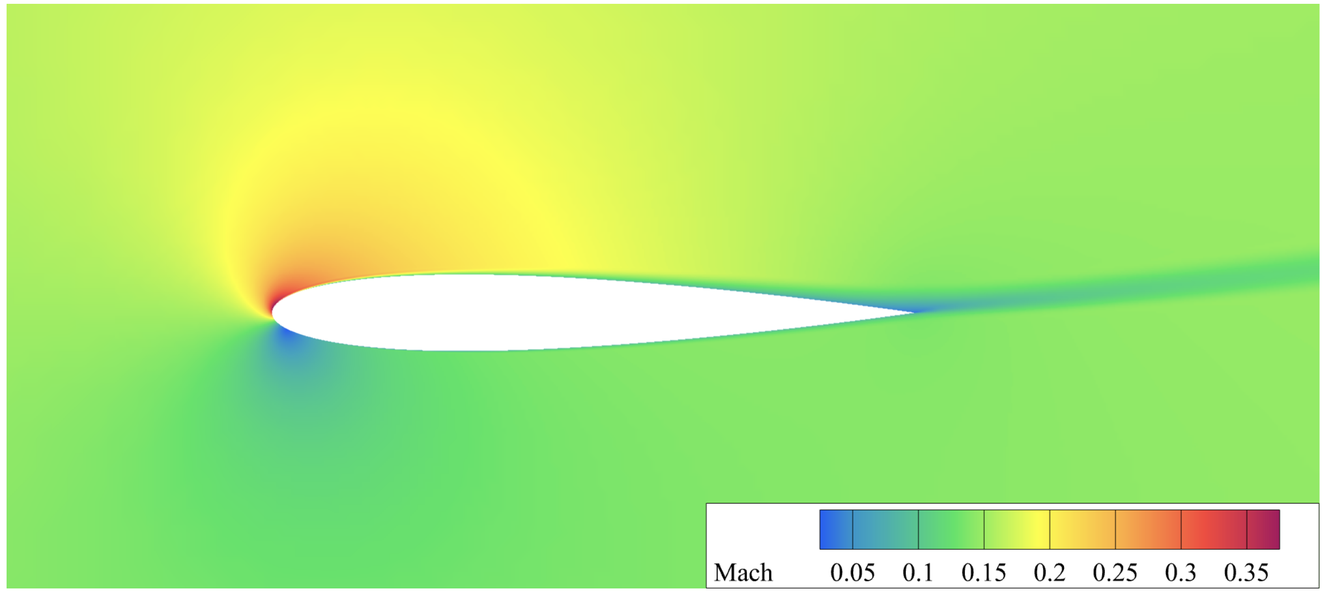


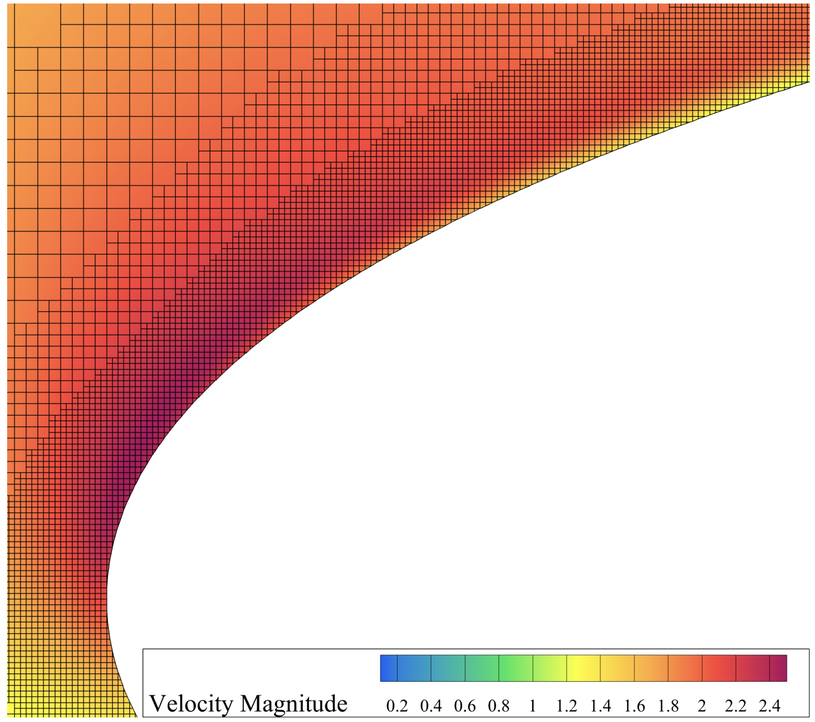
ODE Wall Model



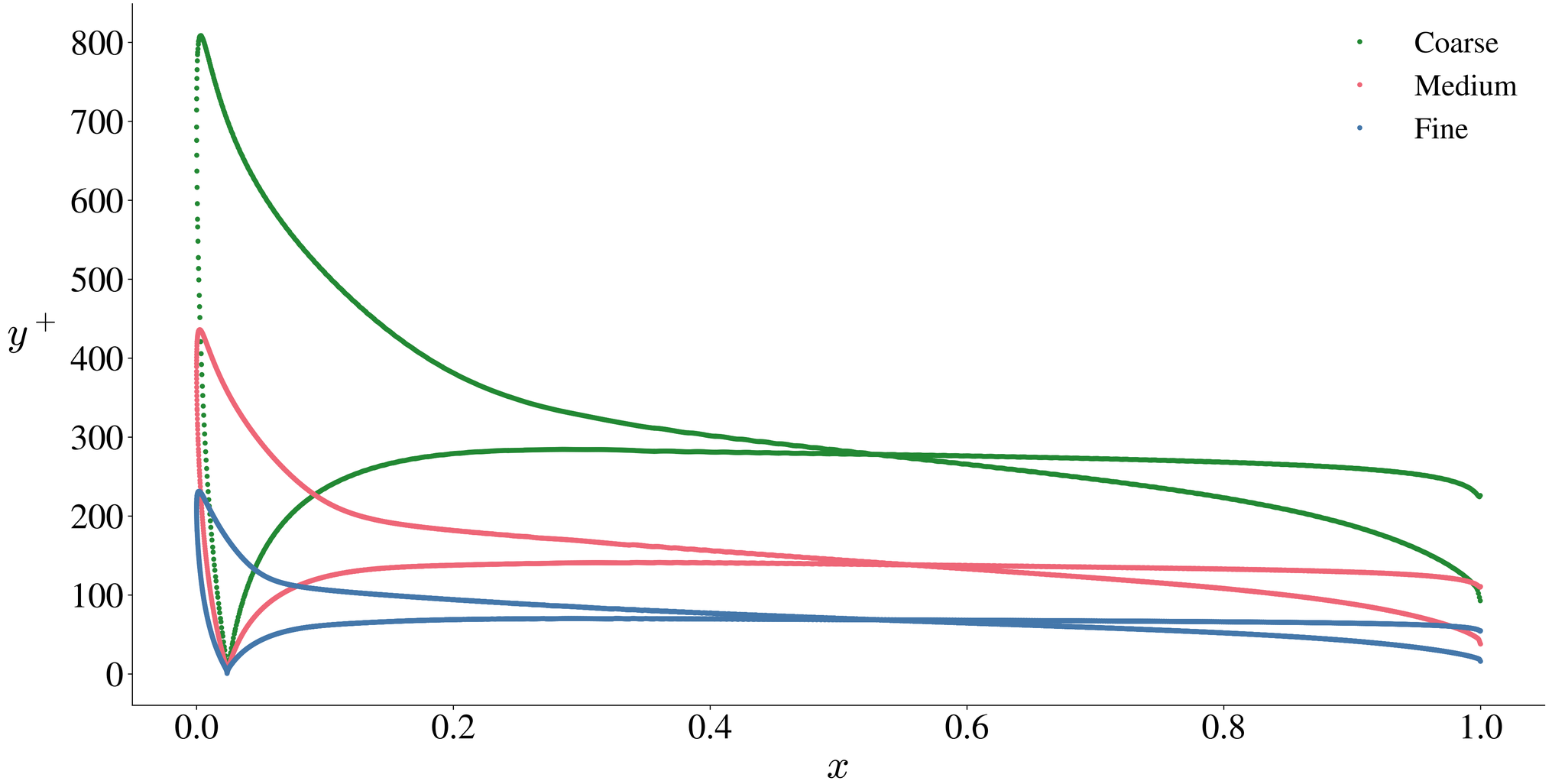
NACA 0012 Airfoil

- Realistic external flow example
- Mach 0.15
- Angle of attack 10 degrees
- Reynolds number 6×10^6



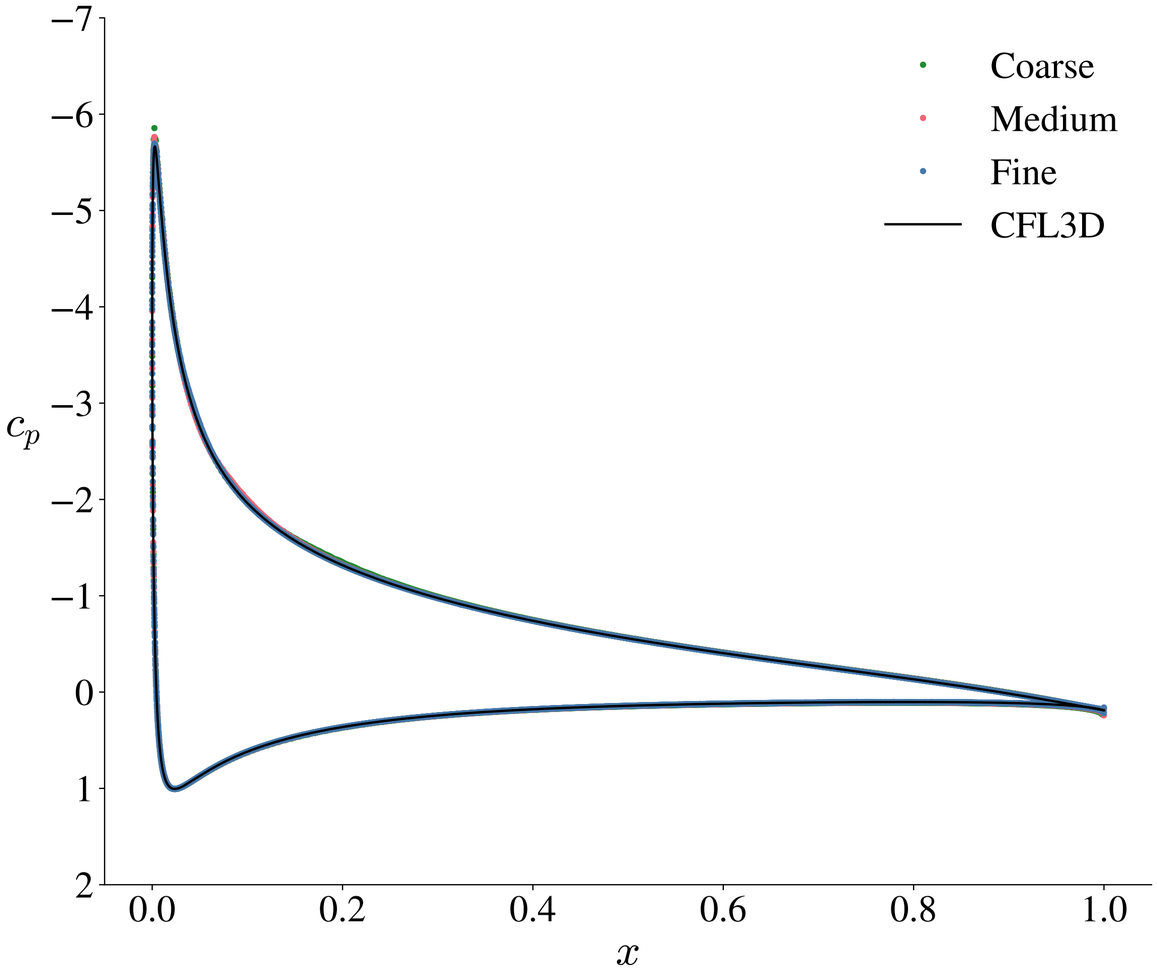


NACA 0012 Forcing Point y+

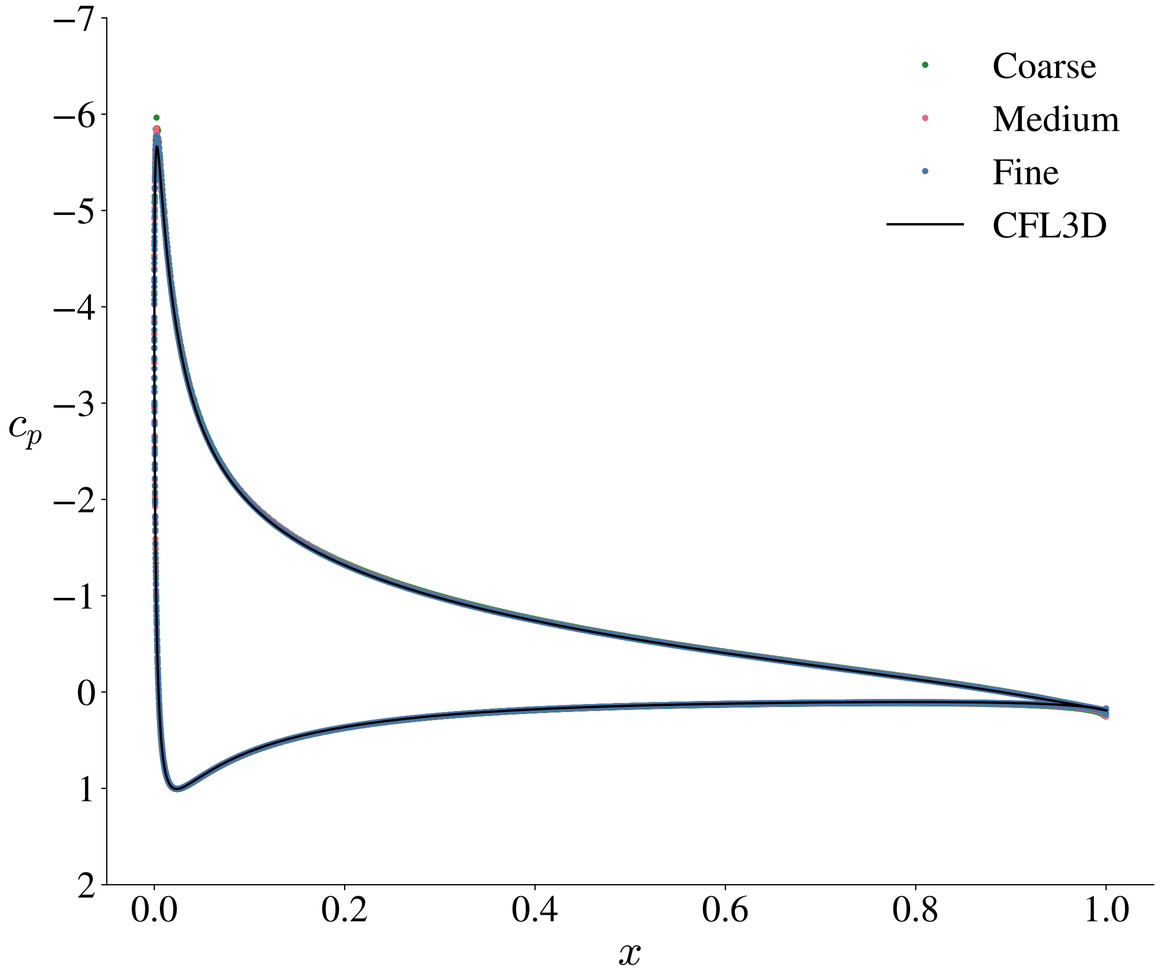


Both forcing point methods match CFL3D surface pressures

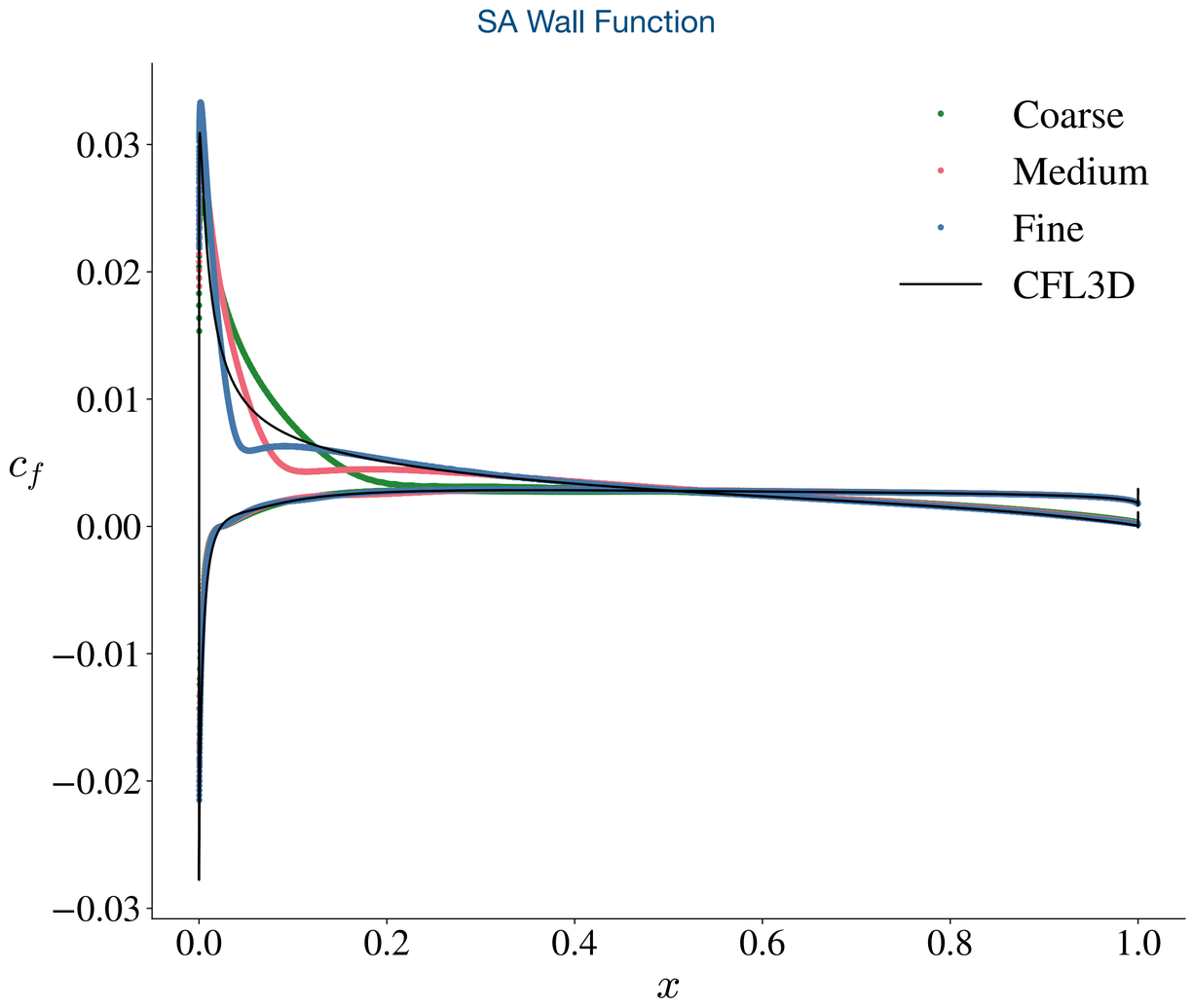
SA Wall Function



ODE Wall Model

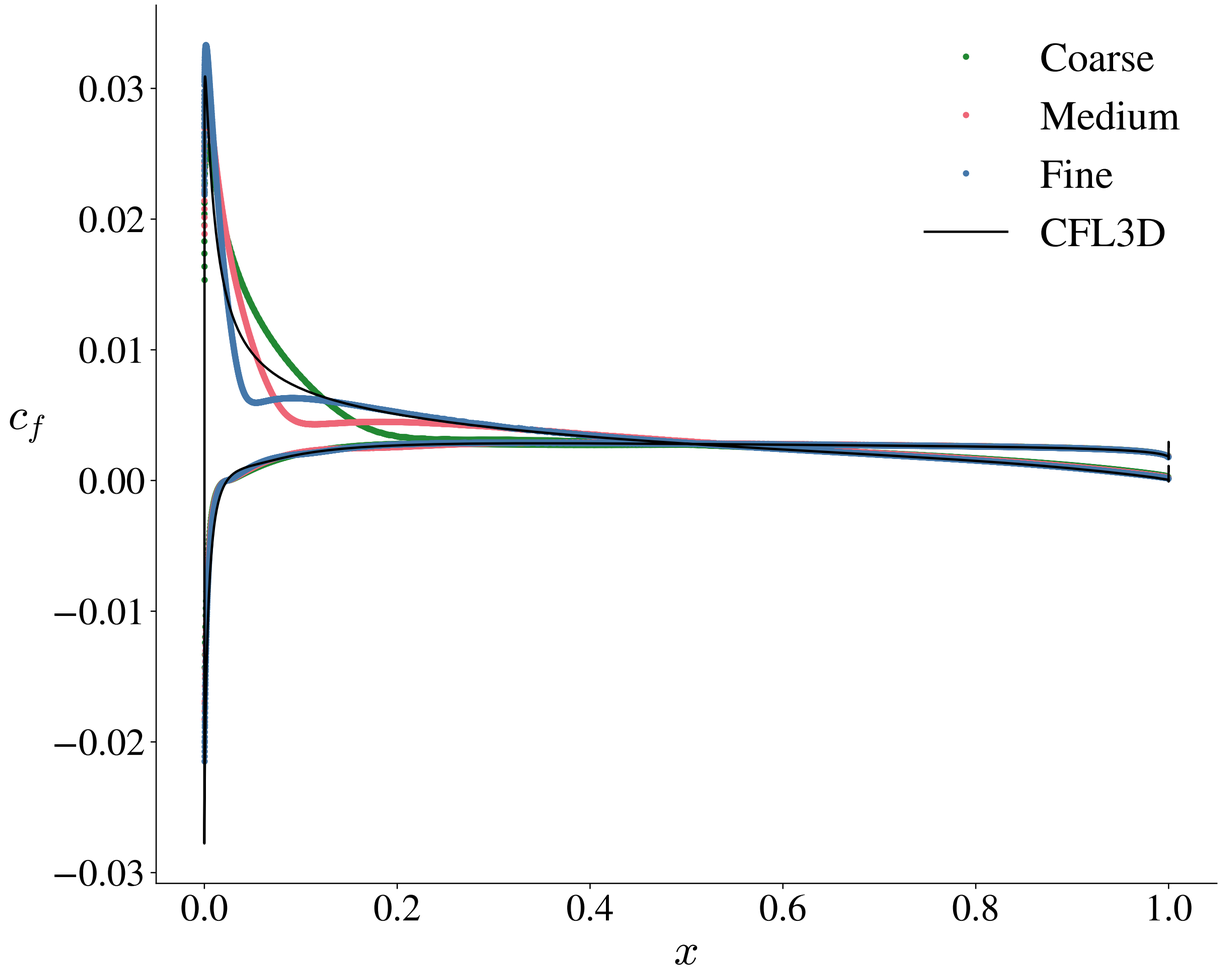


SA wall function skin friction still struggles with boundary layer startup

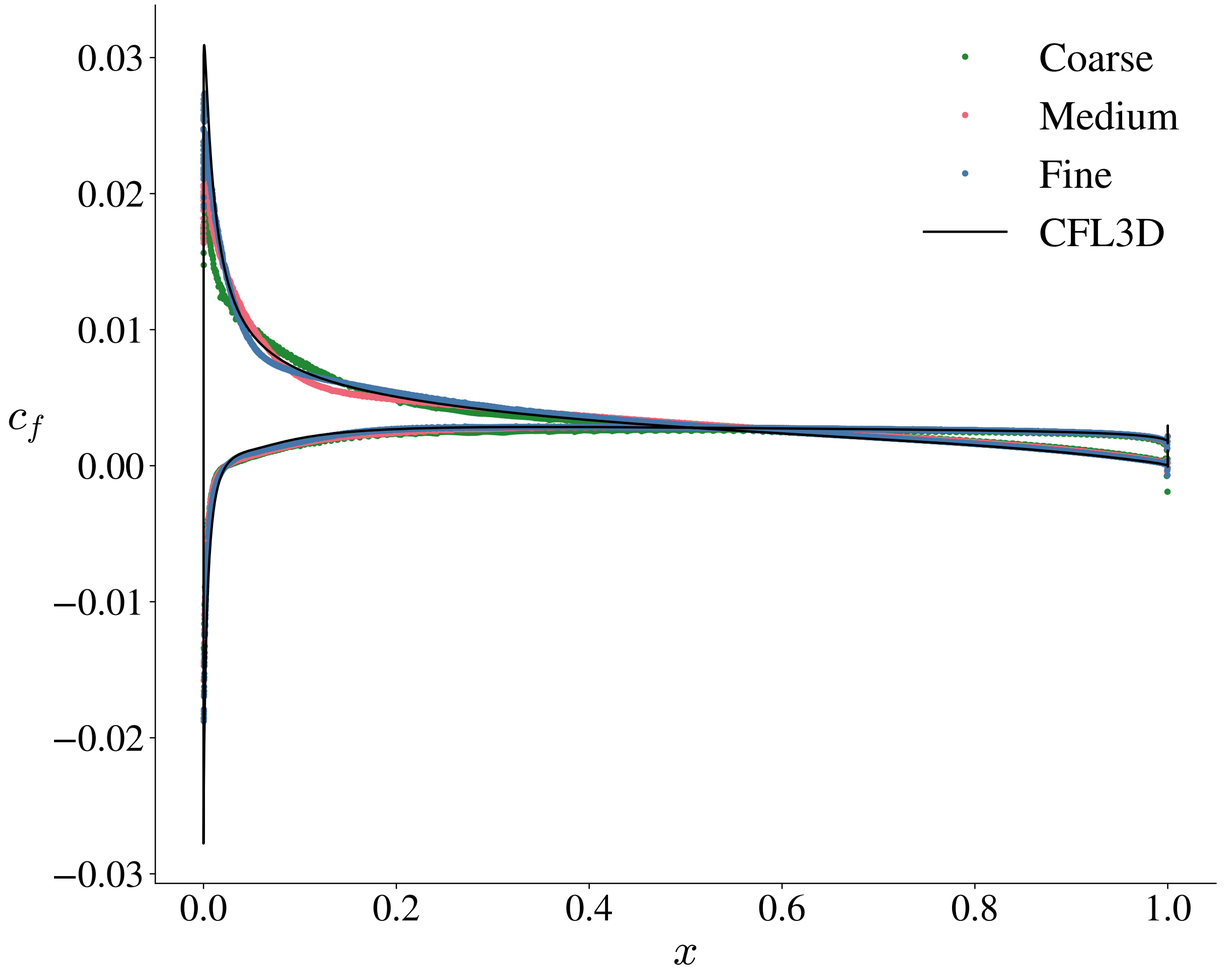


ODE wall model predicts accurate skin frictions across mesh levels

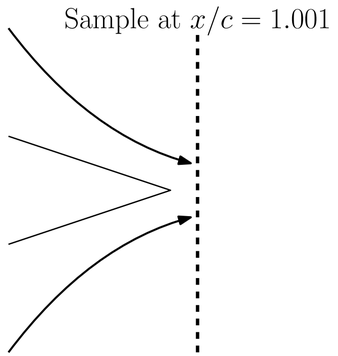
SA Wall Function



ODE Wall Model

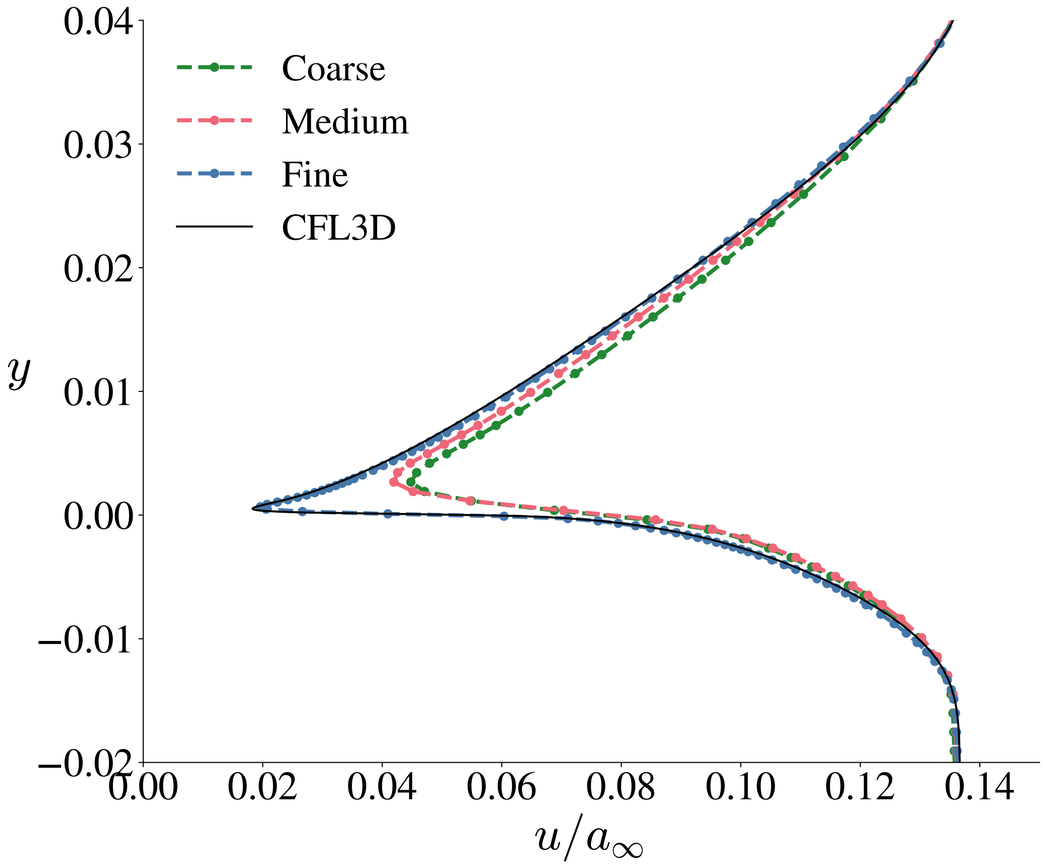


Trailing edge profile samples

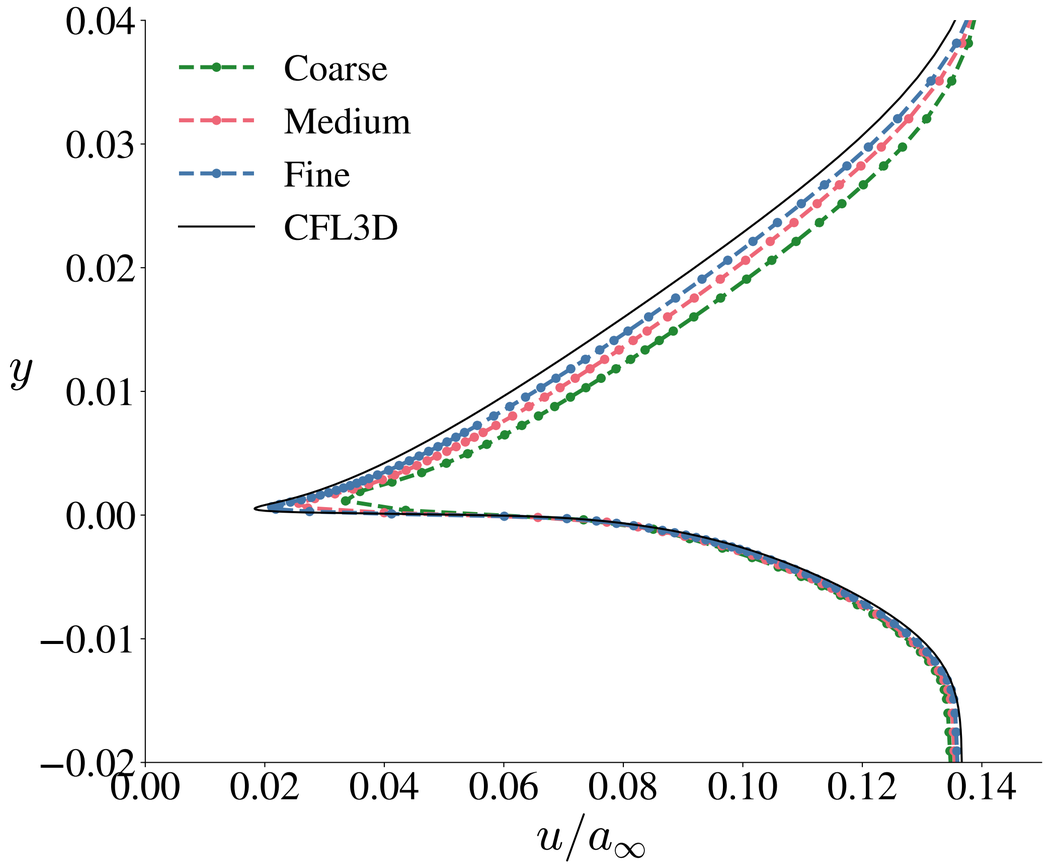


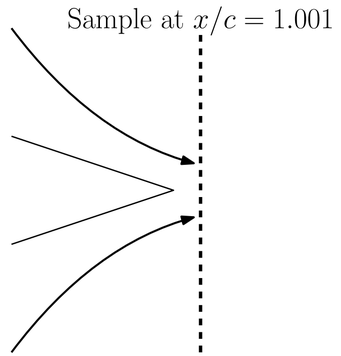
ODE wall model demonstrates increased exit velocity accuracy across mesh levels

SA Wall Function



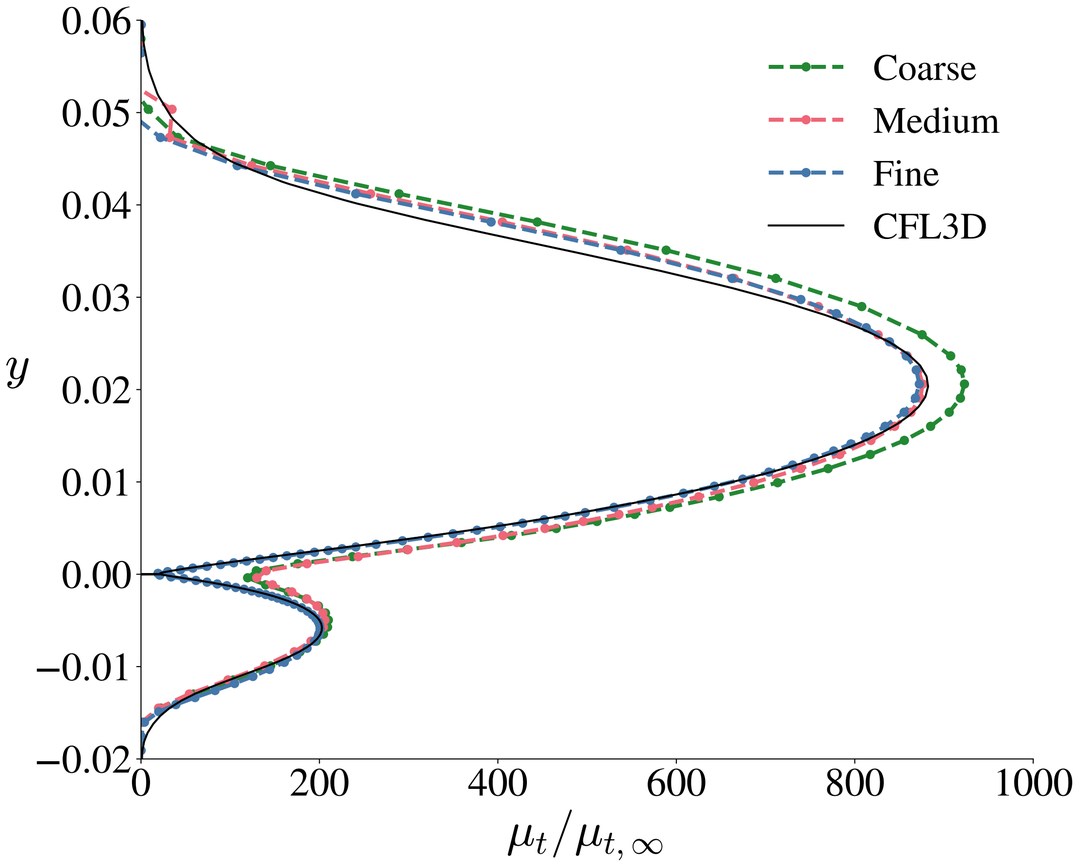
ODE Wall Model



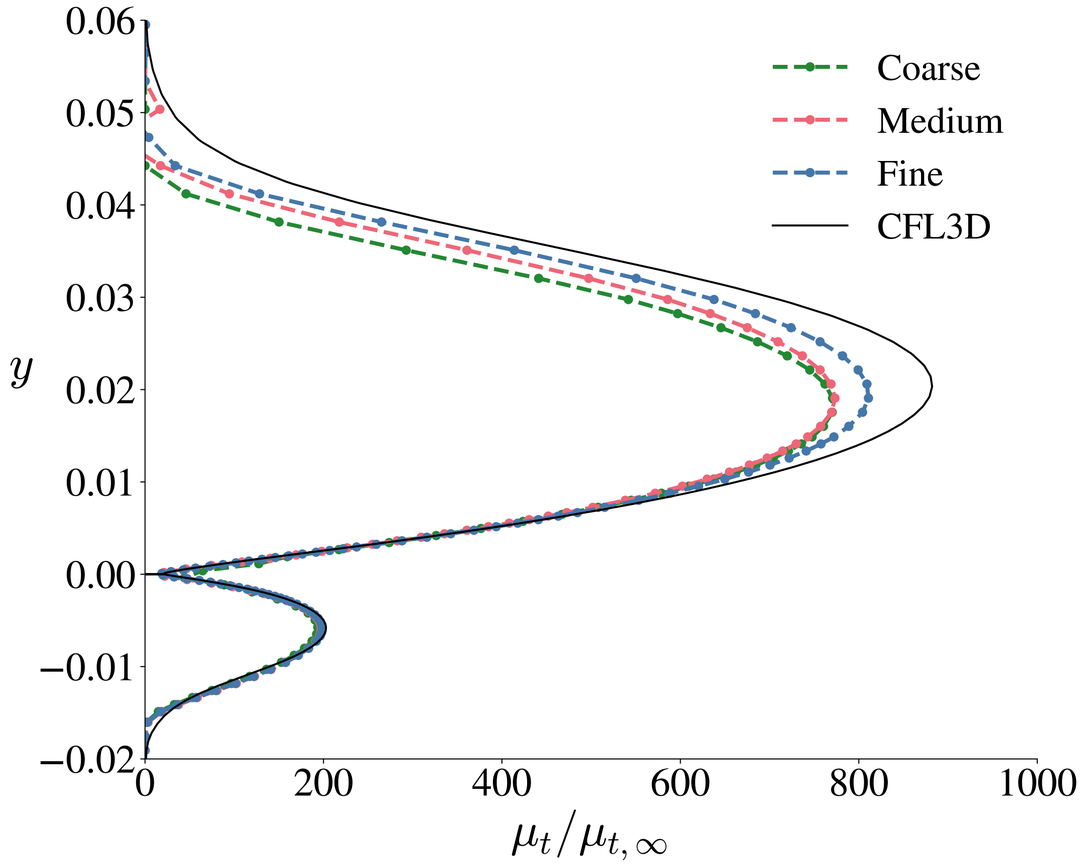


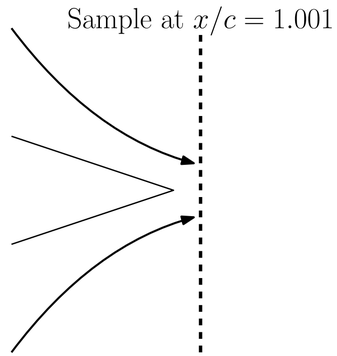
ODE wall model demonstrates increased turbulence variable accuracy near the wall

SA Wall Function



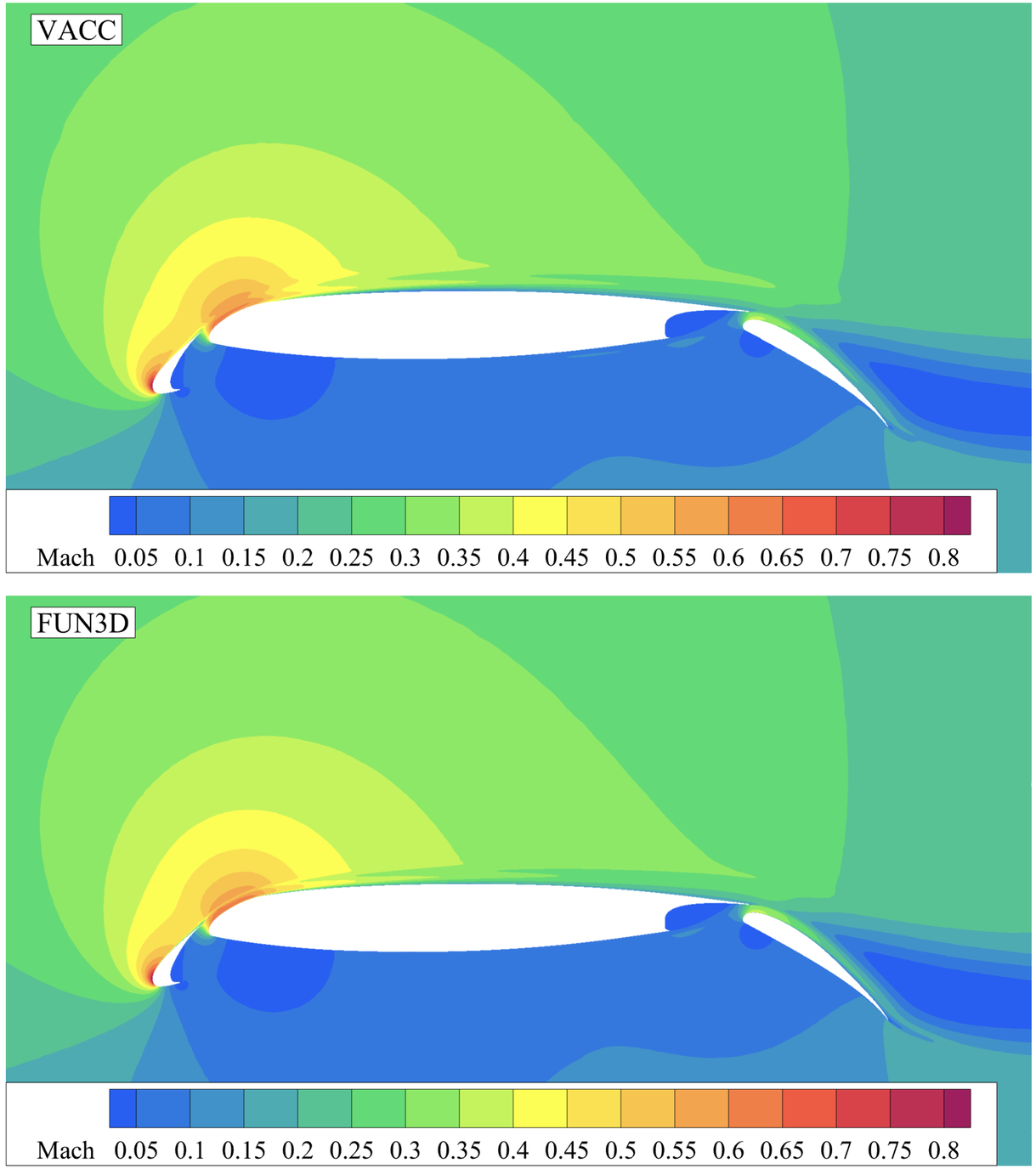
ODE Wall Model



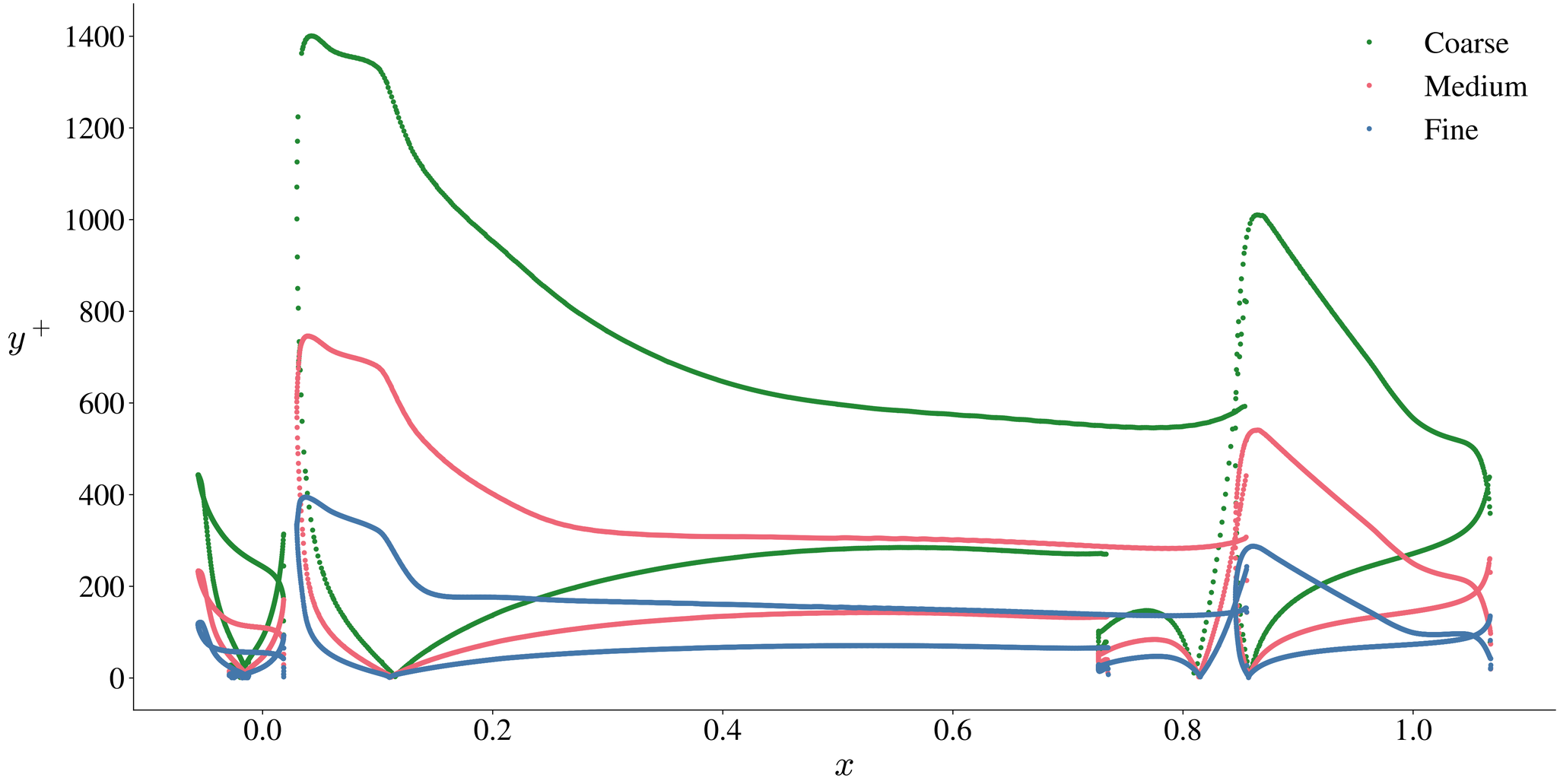


Multielement Airfoil

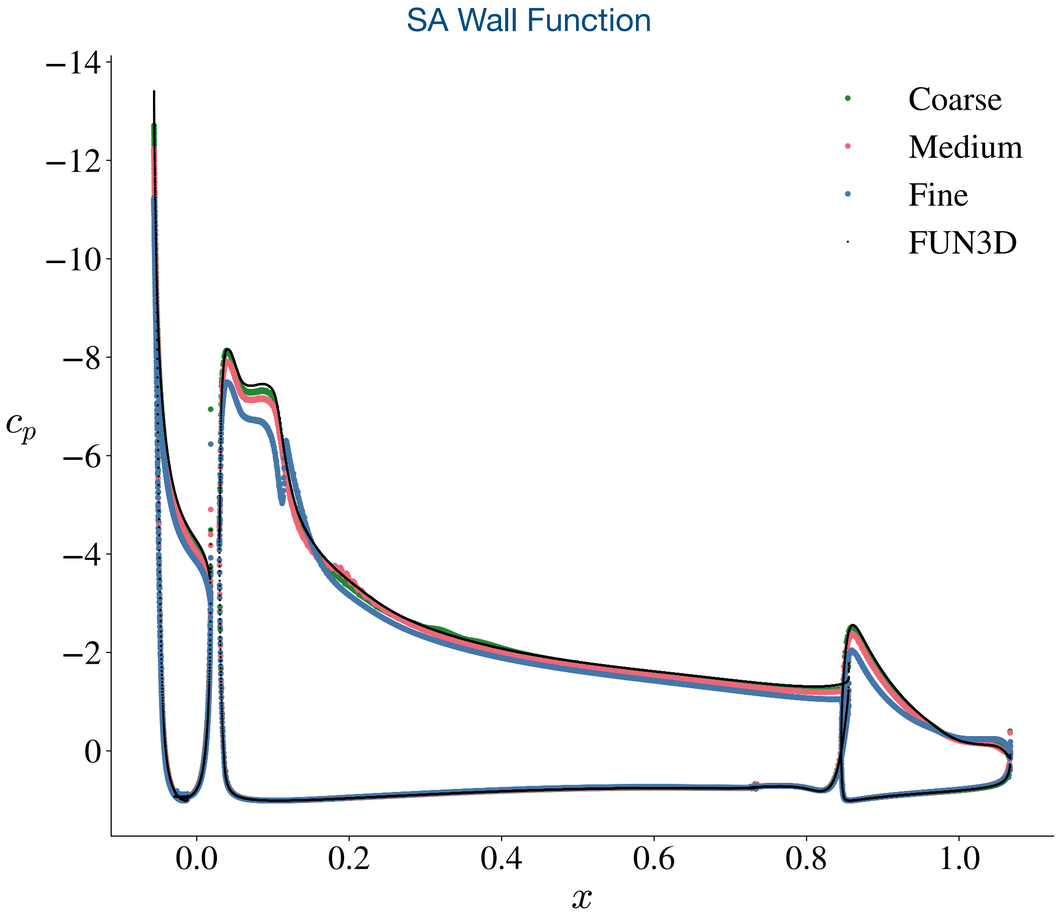
- Complex high-lift external flow example
- Mach 0.2
- Angle of attack 16 degrees
- Reynolds number 5×10^6



Multielement Airfoil Forcing Point y+

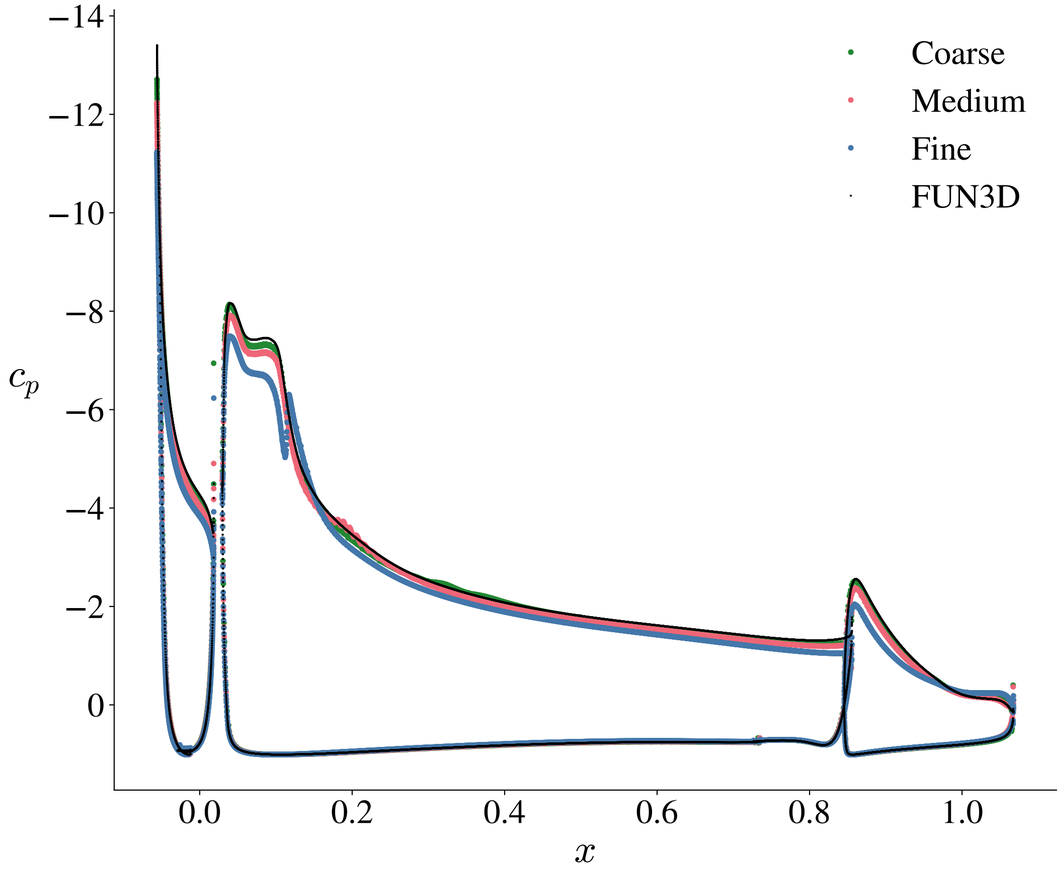


SA wall function surface pressures struggle with main element suction peak

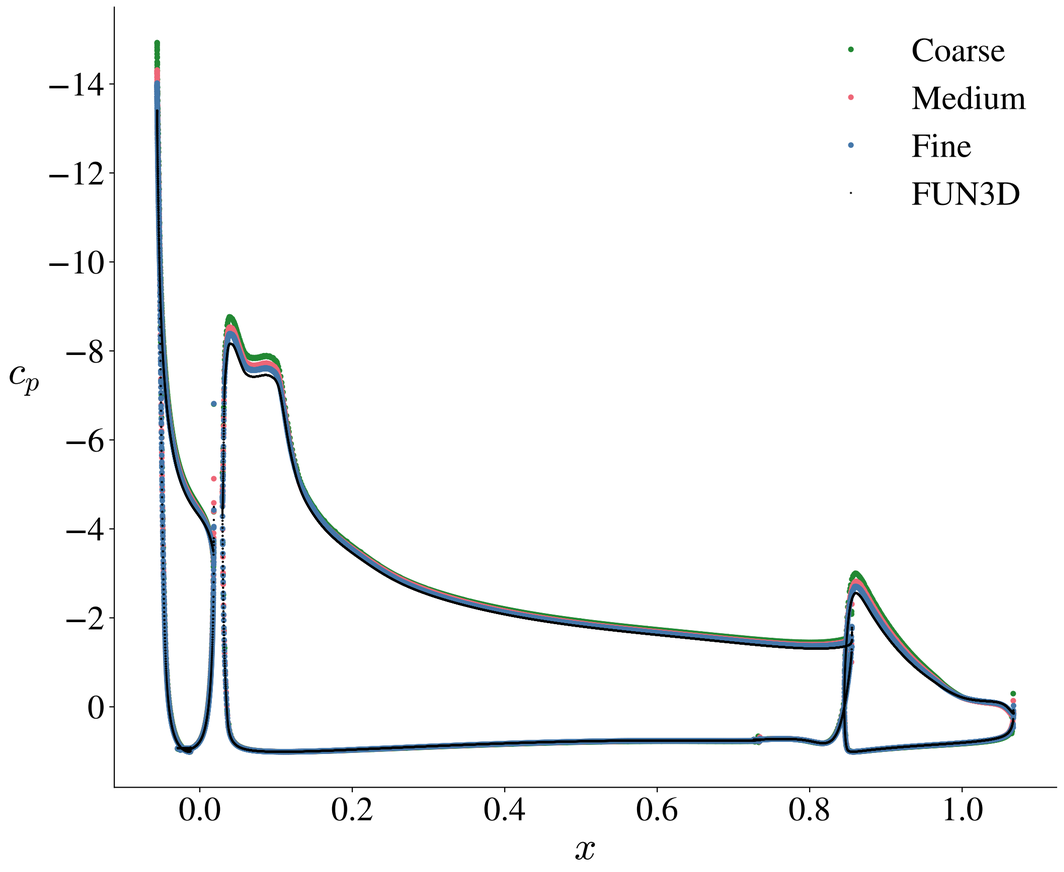


ODE wall model demonstrates excellent mesh convergence

SA Wall Function

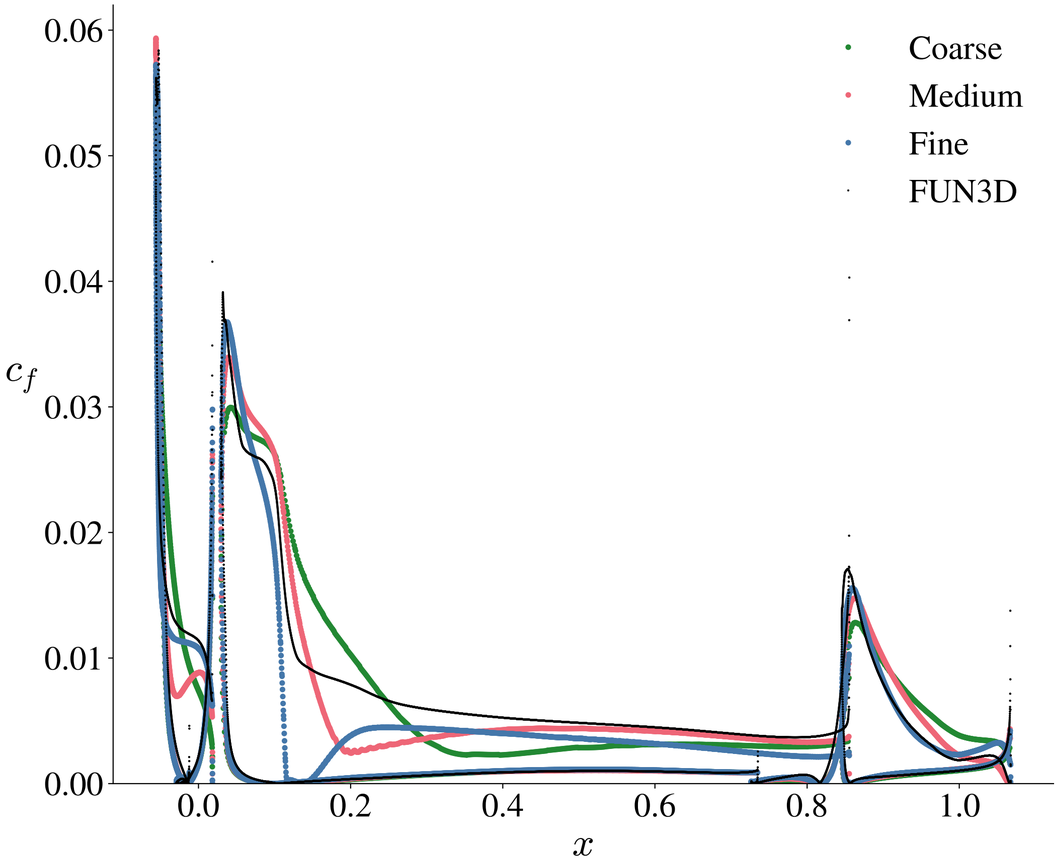


ODE Wall Model

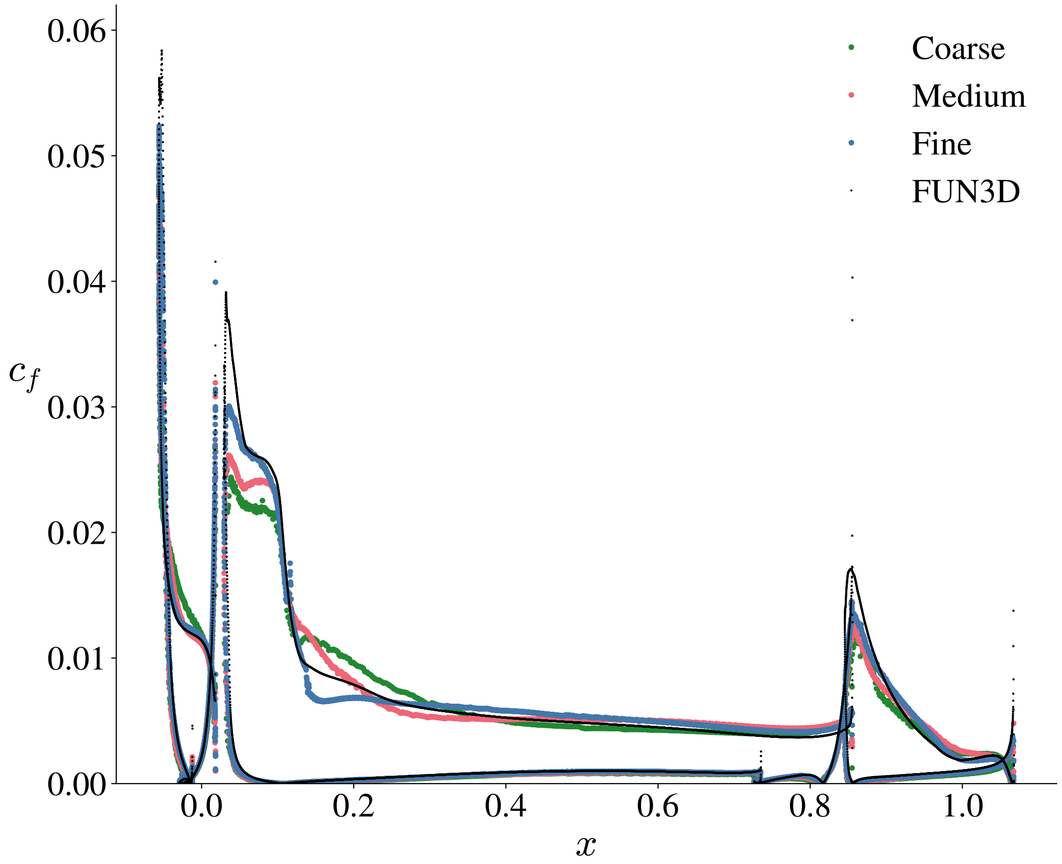


Both methods demonstrate mesh convergent skin friction

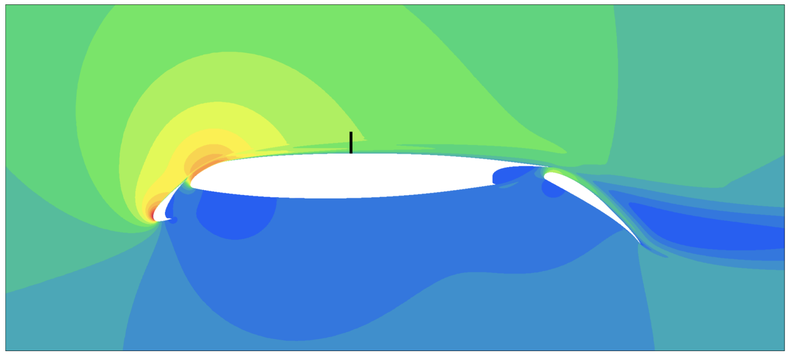
SA Wall Function



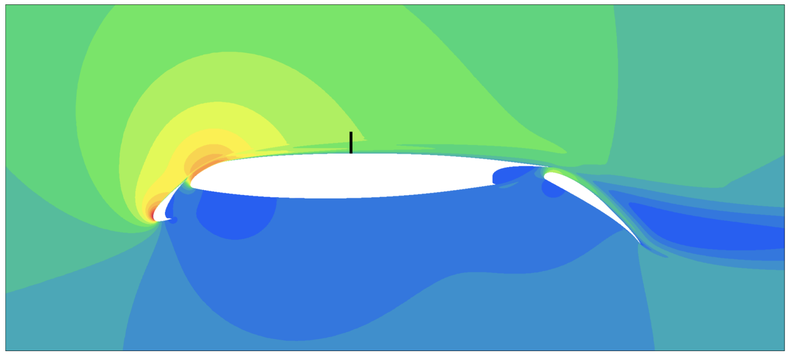
ODE Wall Model

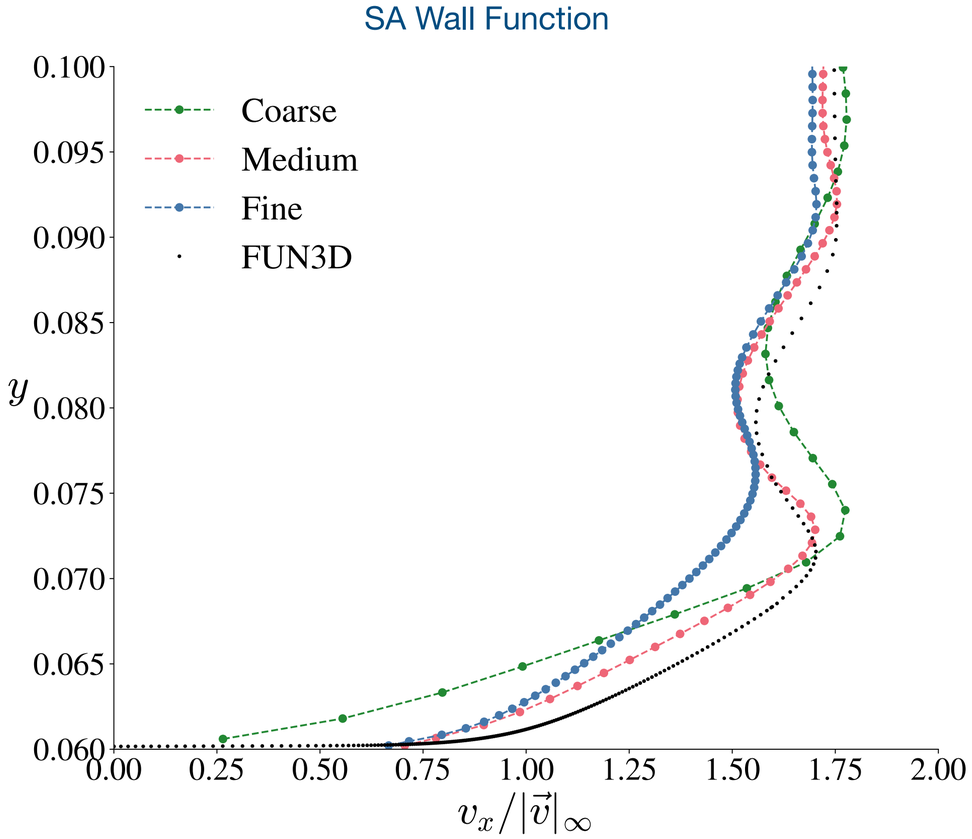


Looking at data from one line cut

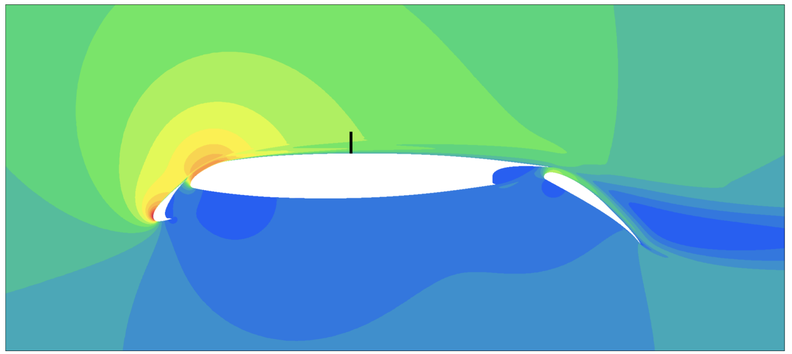


The SA wall function velocity is not mesh convergent

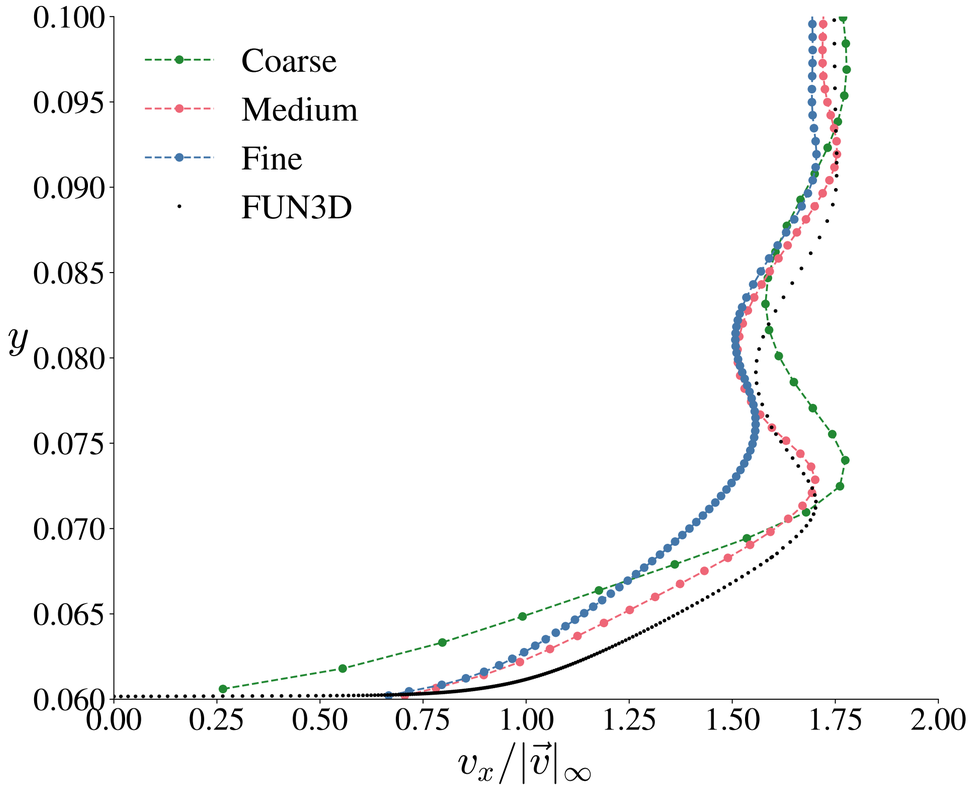




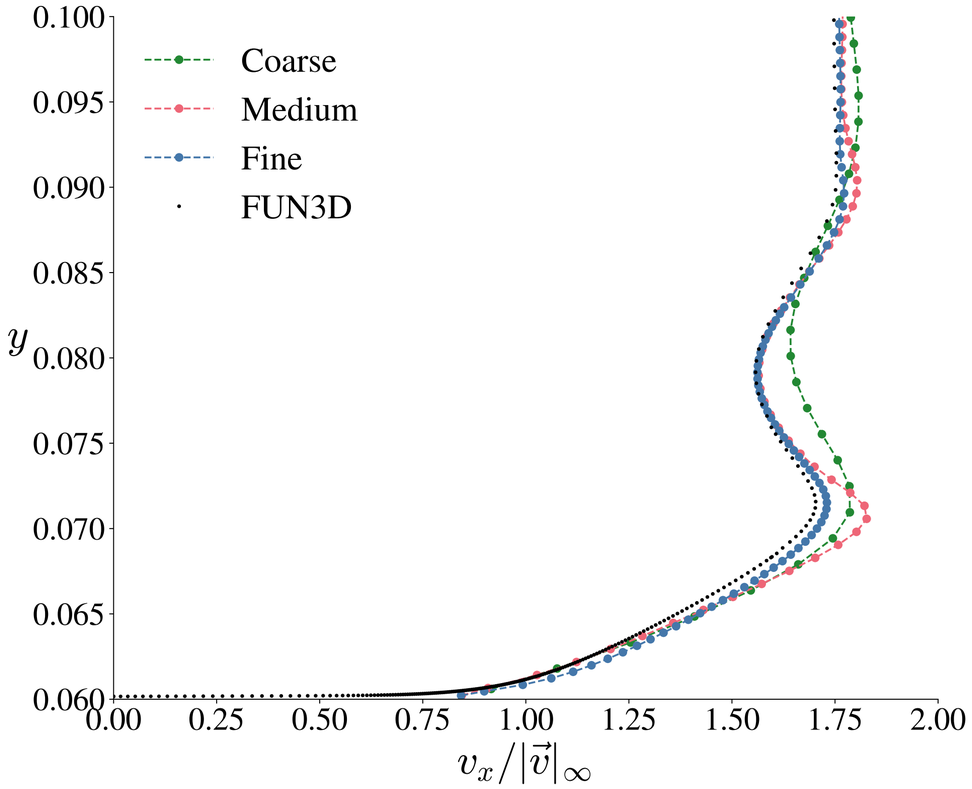
The ODE wall model shows good agreement with FUN3D



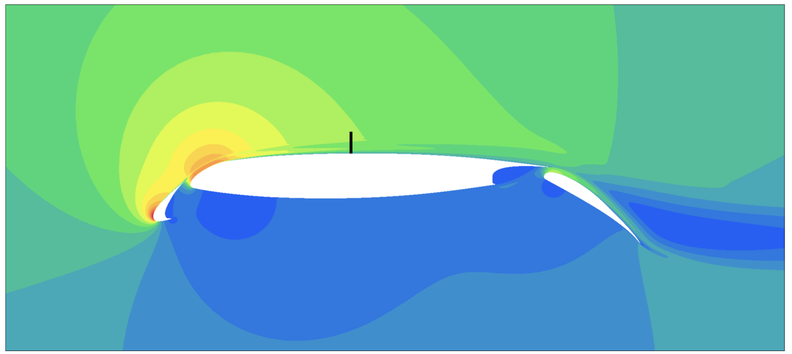
SA Wall Function

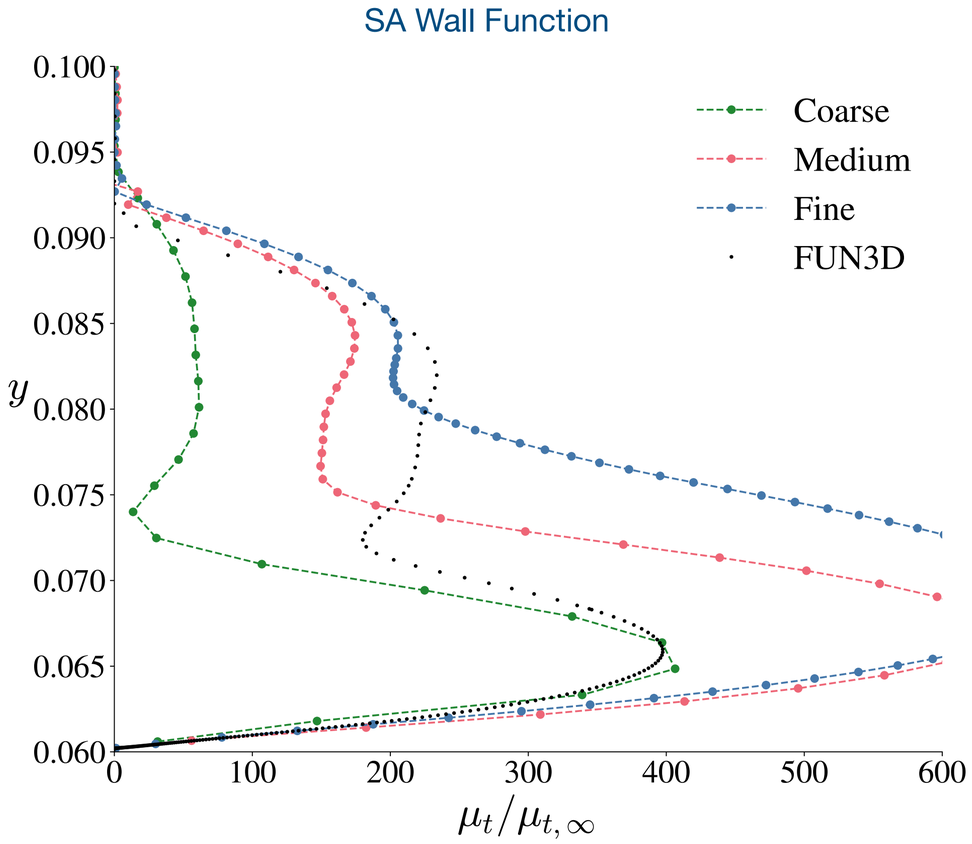


ODE Wall Model

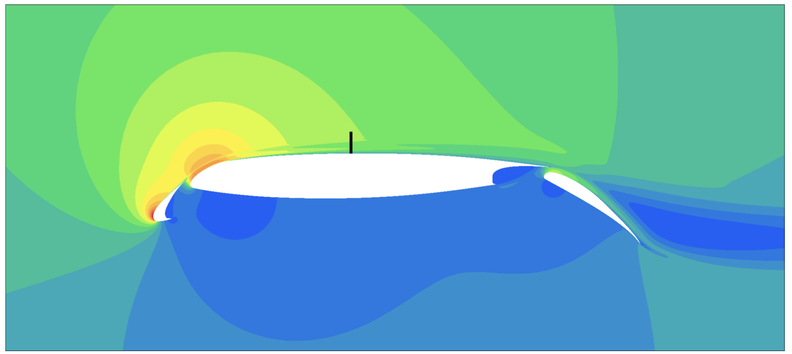


SA wall function is not mesh convergent

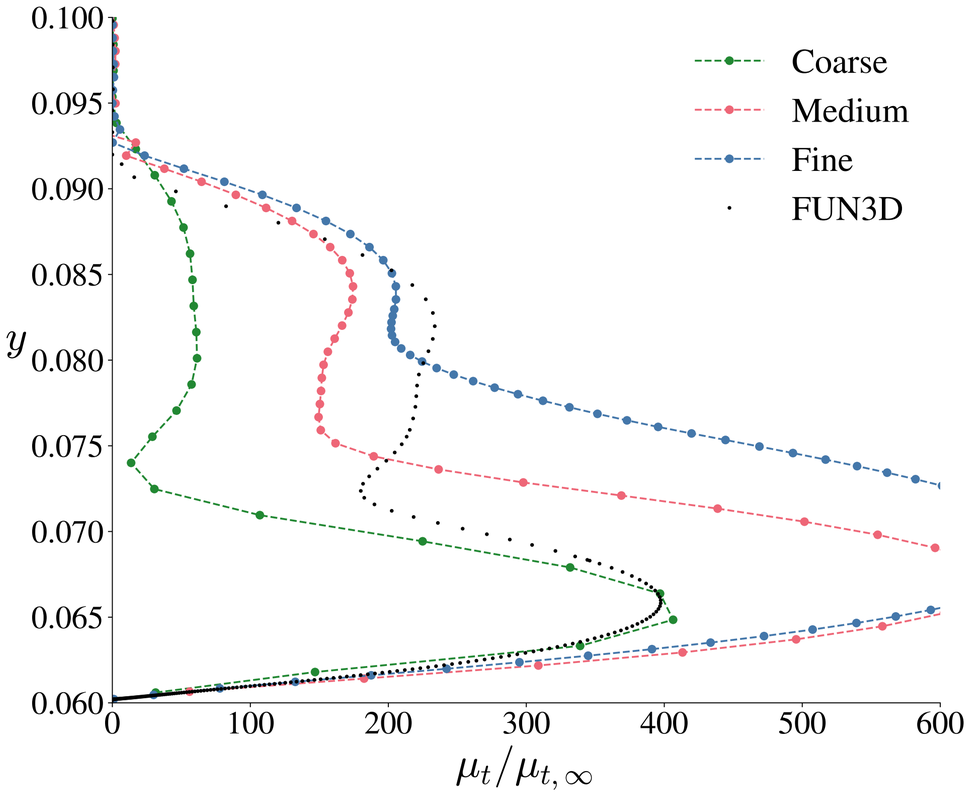




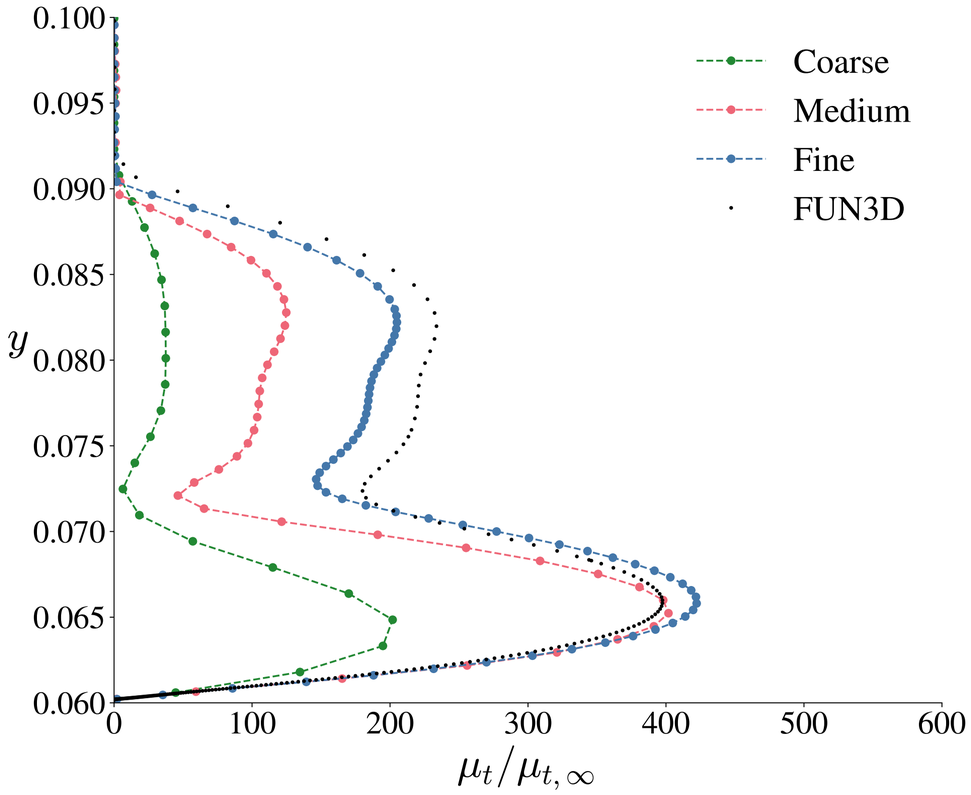
ODE wall model shows mesh convergence



SA Wall Function

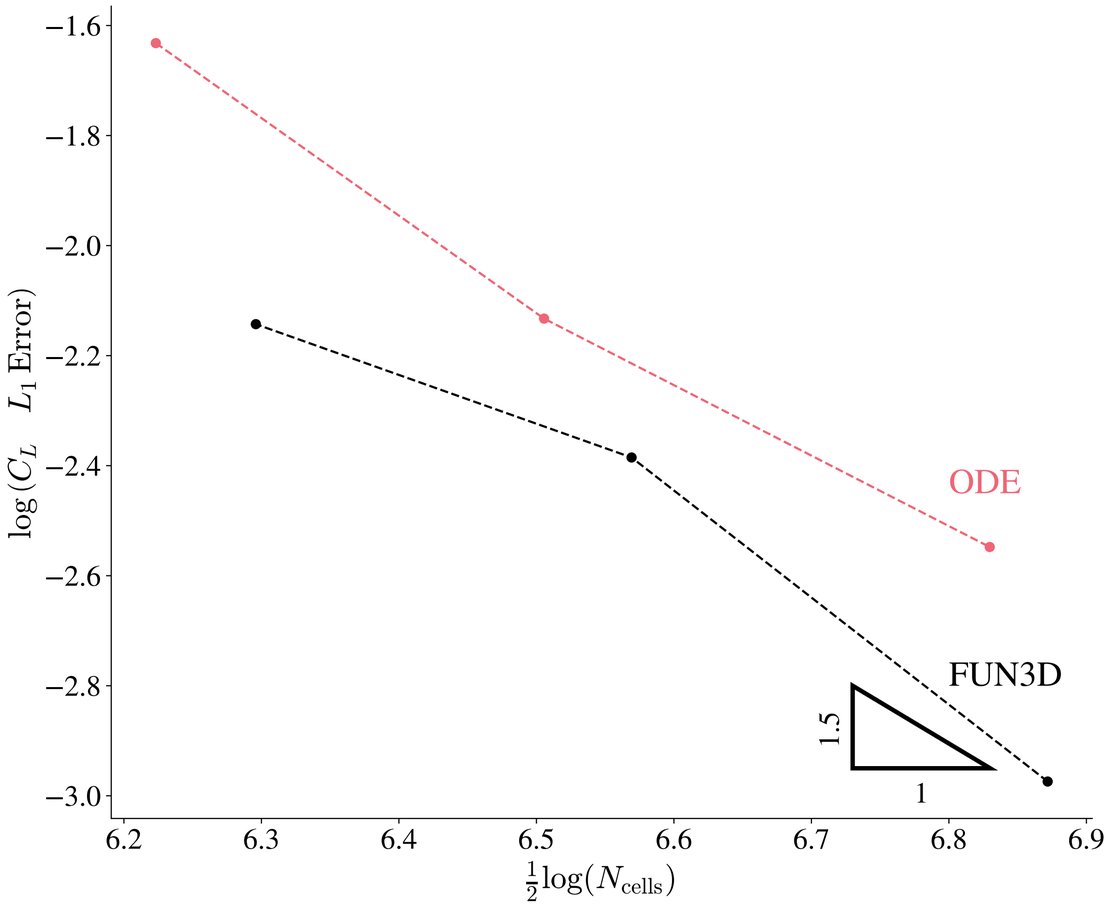


ODE Wall Model

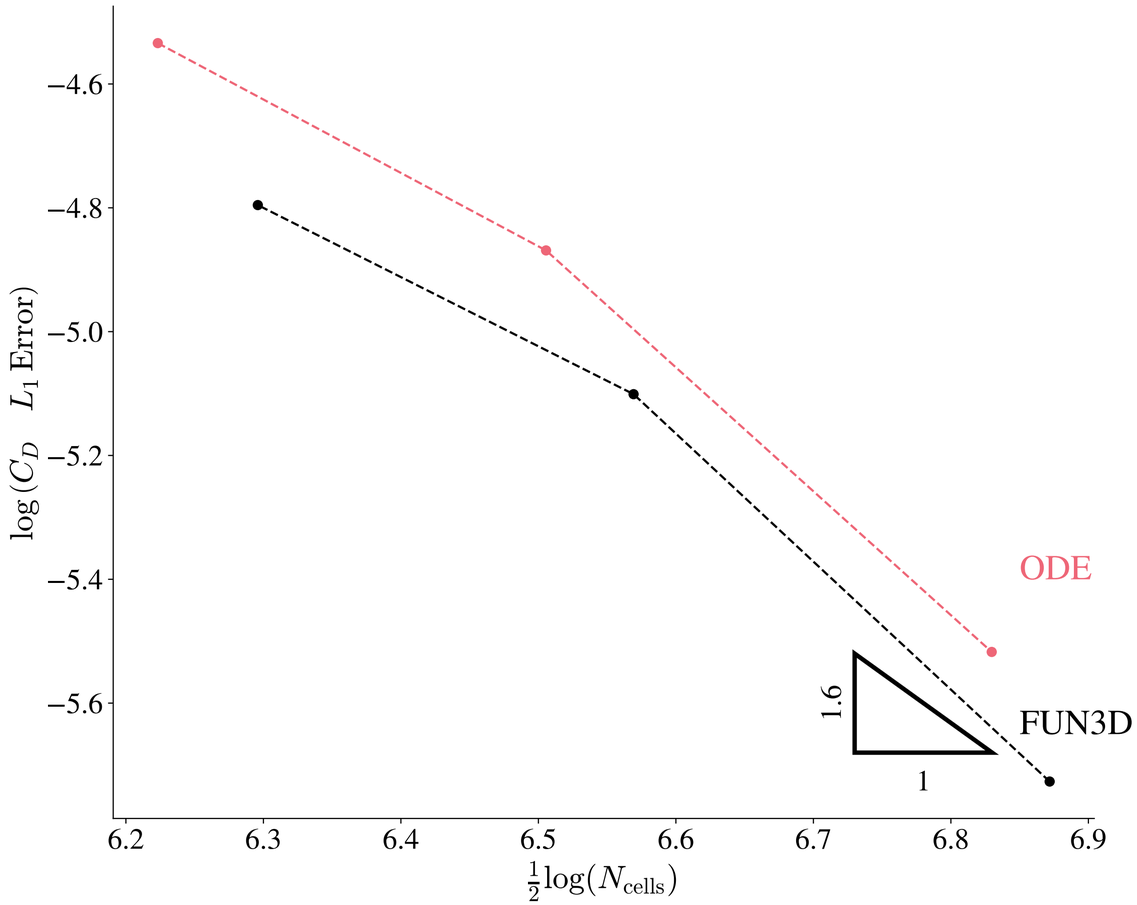


ODE wall model exhibits excellent lift and drag error convergence

Lift Error



Drag Error

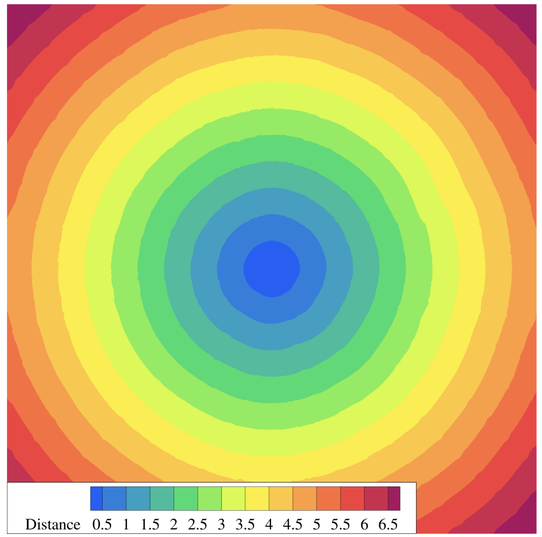


Outline

- Introduction
- Viscous Aerodynamic Cartesian Cut-cell (VACC)
- Eikonal distance function
- ODE wall model
- NASA TMR test cases
- Conclusions

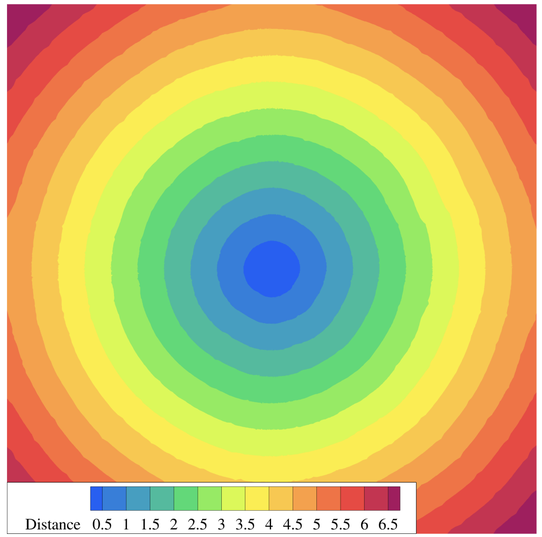
Conclusions

Eikonal Solver

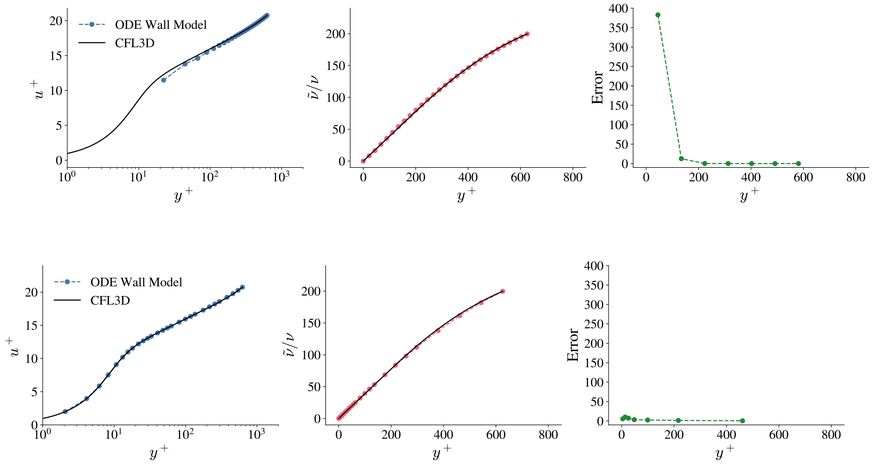


Conclusions

Eikonal Solver

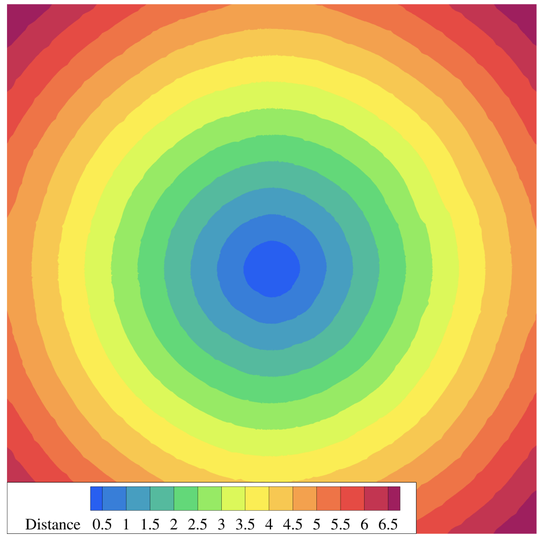


ODE Wall Model Solver and Integration

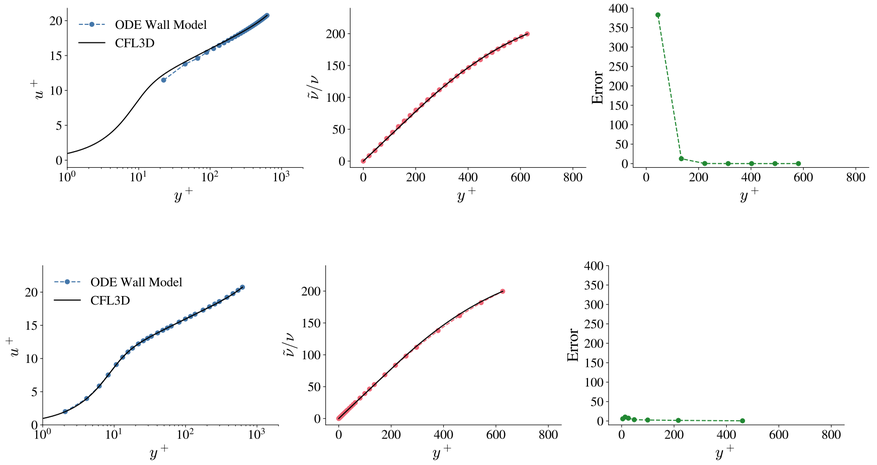


Conclusions

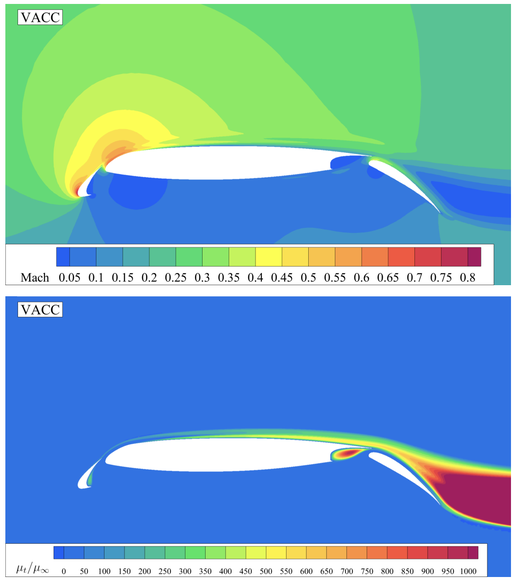
Eikonal Solver



ODE Wall Model Solver and Integration



Compared Cartesian Cut-cell Approaches on Increasingly Complex Test Cases



Thank you





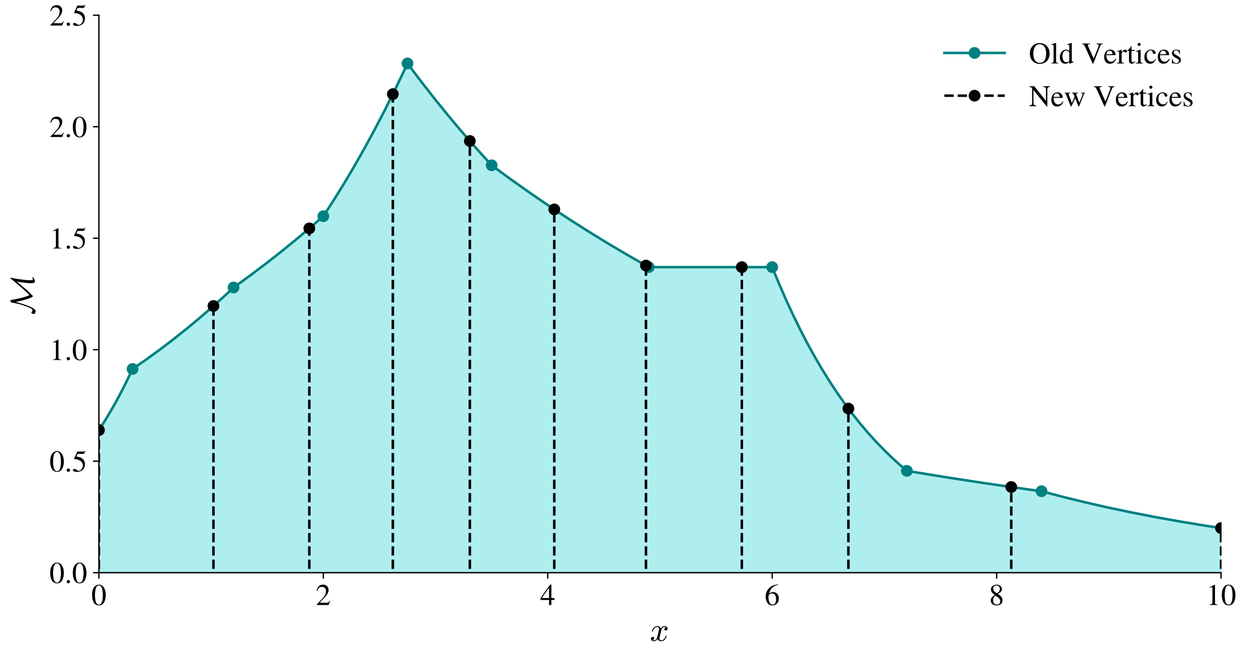


National
Science
Foundation

GOOD THROUGH HERE

1D MOESS

- Computes an update to the Riemman metric field that minimizes error for a given cost
- Uses the adjoint weighted residual for the wall shear stress as the error metric
- Generates a metric conforming mesh from the updated metric field



Jacobian-Free Newton—Krylov

- Fully coupled turbulence model scaled by freestream turbulence variable
- Pseudo-transient continuation globalization strategy
- Fréchet derivatives for matrix vector products
- Block diagonal point Jacobi preconditioner for GMRES

$$\left(\frac{\mathbf{A}}{\Delta \mathbf{t}} \mathbf{P}_t + \left. \frac{\partial \mathbf{R}}{\partial \mathbf{u}} \right|_{\mathbf{u}^n} \right) \Delta \mathbf{u} = - \mathbf{R} \left(\mathbf{u}^n \right)$$

$$\mathbf{P}_{t,\text{local}} = \frac{\partial \mathbf{u}_{\text{conservative}}}{\partial \mathbf{u}_{\text{primitive}}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 \\ (u^2 + v^2)/2 & \rho u & \rho v & 1/(\gamma - 1) & 0 \\ \tilde{\nu} & 0 & 0 & 0 & \rho \end{bmatrix}$$

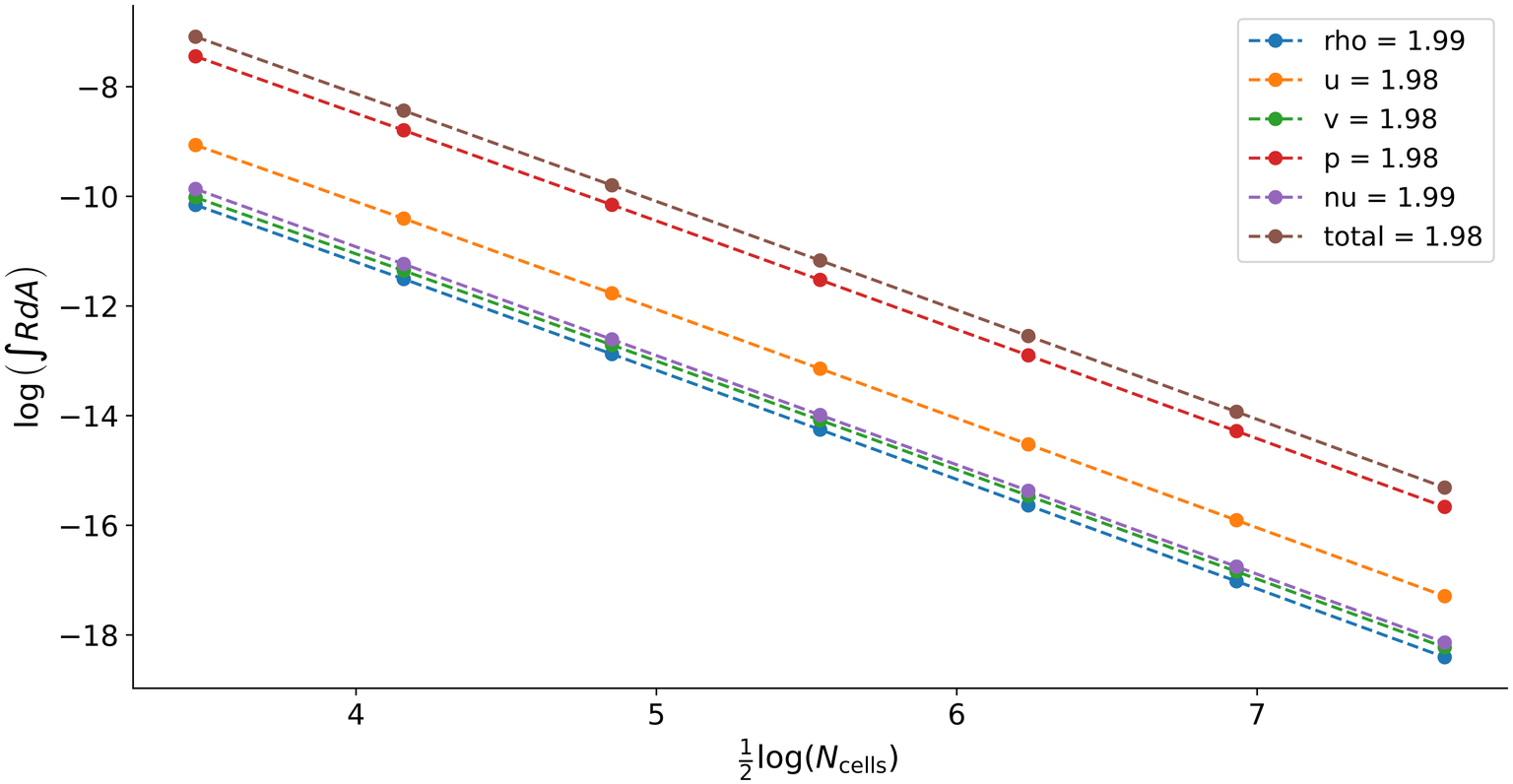
Complex Eikonal Geometry Eyeball Test

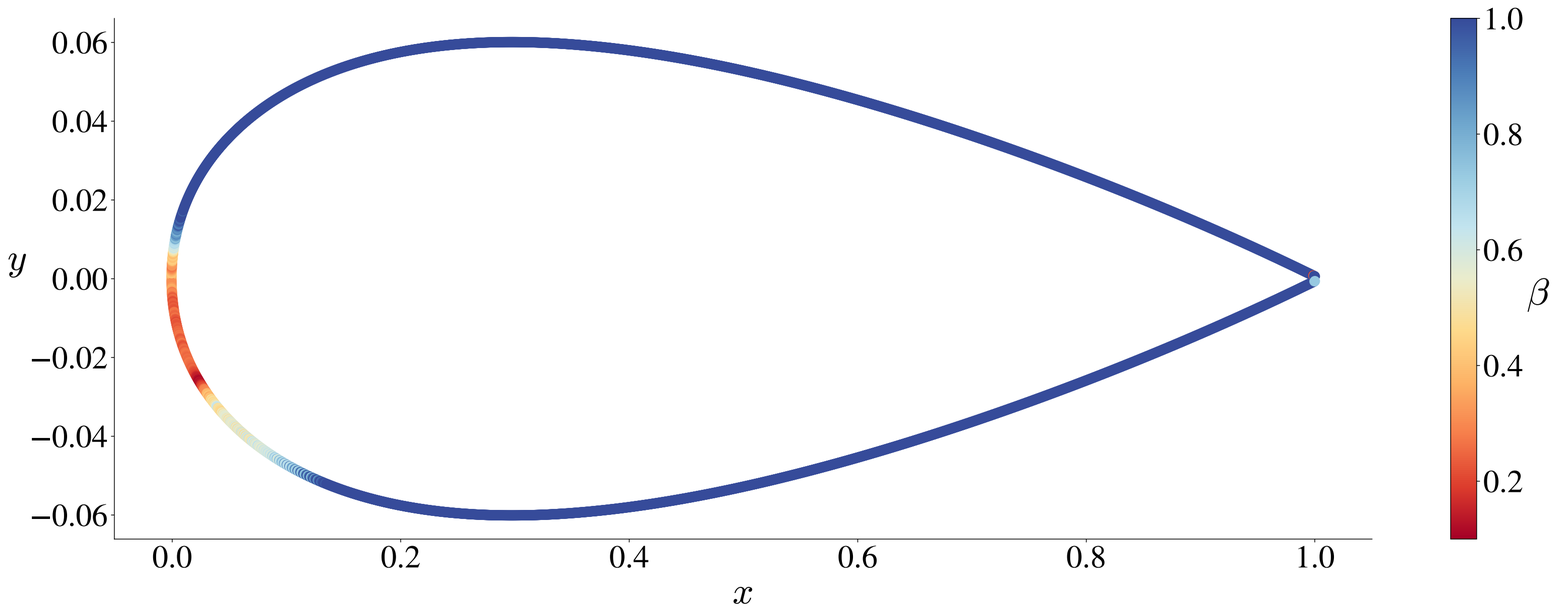


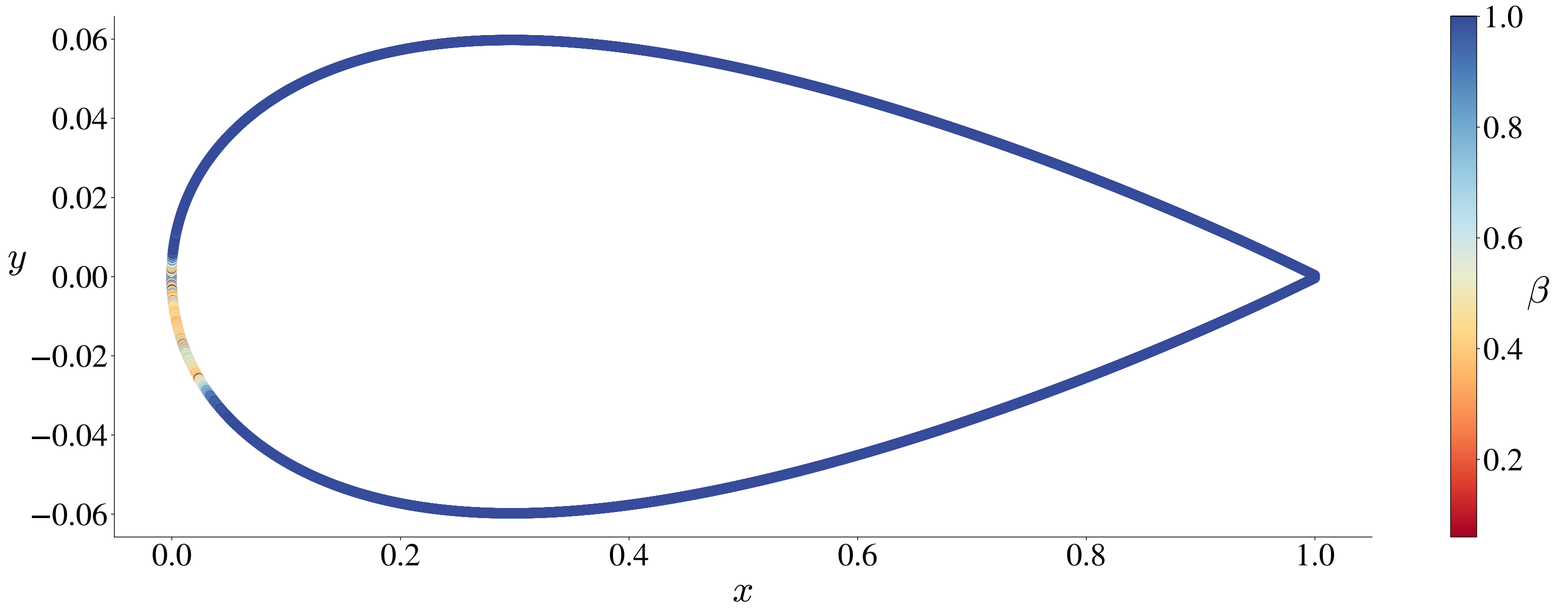


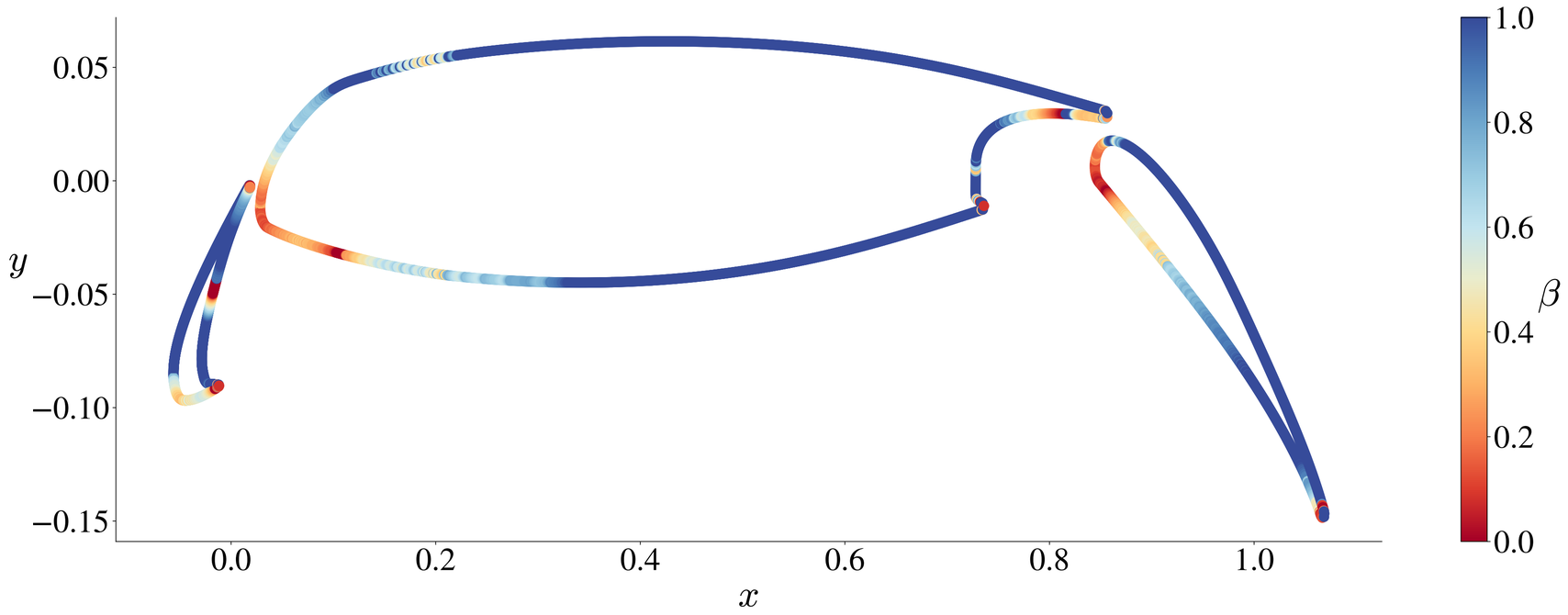
RANS SA-neg Manufactured Solution

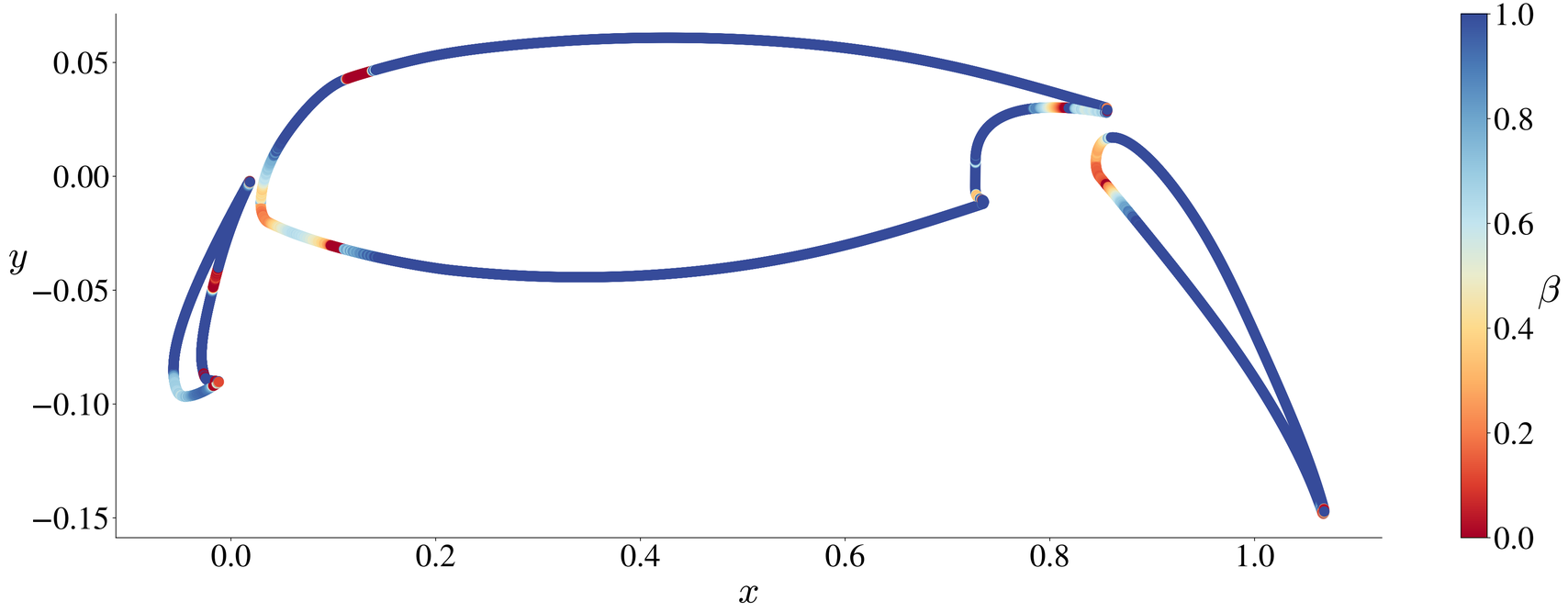
- Sinusoidal functions for each state component
- Initialized solution to exact value at cell centroids
- Called residual evaluation
- Integrate residual over domain
- Verified interior fluxes and source terms



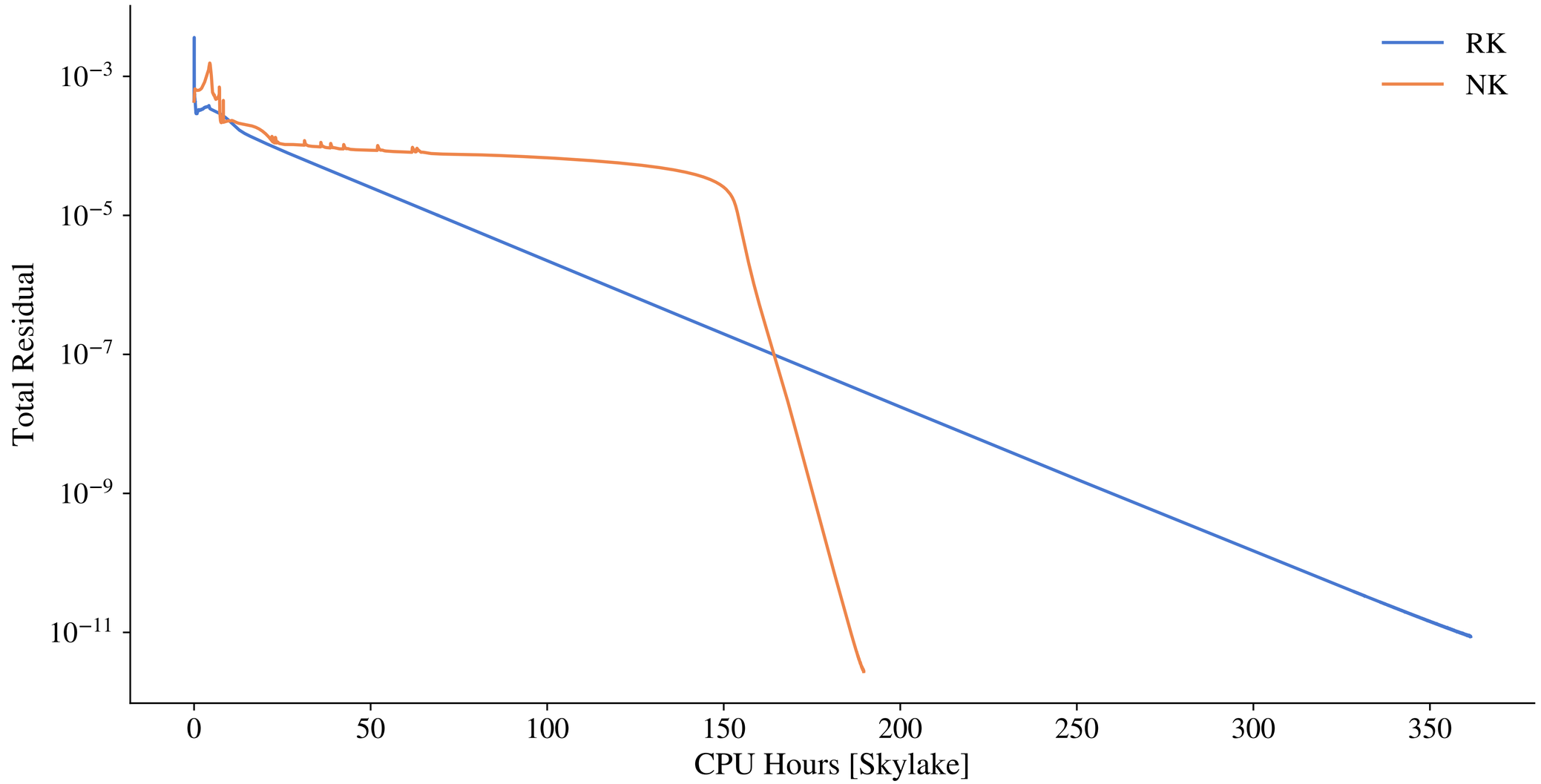

0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
y
0.0
0.2
0.4
0.6
0.8
1.0
x
1.0
0.8
0.6
0.4
0.2
β


0.06
0.04
0.02
0.00
−0.02
−0.04
−0.06
y
0.0
0.2
0.4
0.6
0.8
1.0
x
1.0
0.8
0.6
0.4
0.2
β

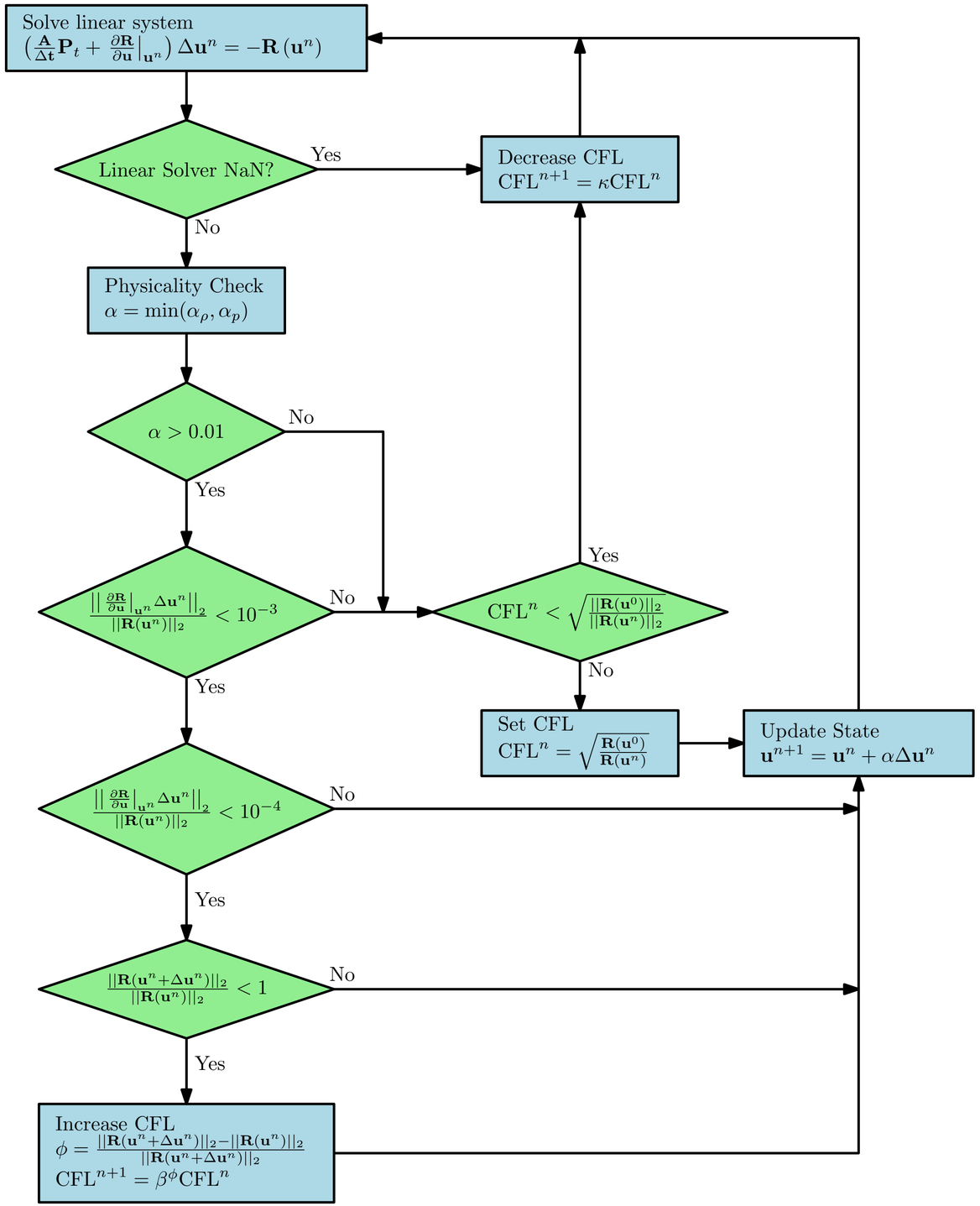

0.05
0.00
−0.05
−0.10
−0.15
y
0.0
0.2
0.4
0.6
0.8
1.0
x
1.0
0.8
0.6
0.4
0.2
0.0
β


0.05
0.00
−0.05
−0.10
−0.15
y
0.0
0.2
0.4
0.6
0.8
1.0
x
1.0
0.8
0.6
0.4
0.2
0.0
β

Grid Aligned Flat Plate Convergence



CFL Controller



ODE wall model exhibits excellent lift and drag mesh convergence

Lift Error

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

Drag Error

Mesh	FUN3D	SA	ODE
Coarse	8.27×10^{-3}	7.11×10^{-3}	1.07×10^{-2}
Medium	6.09×10^{-3}	7.72×10^{-3}	7.68×10^{-3}
Fine	3.26×10^{-3}	2.62×10^{-2}	4.01×10^{-3}
Accuracy	**1.62**	**N/A**	**1.63**