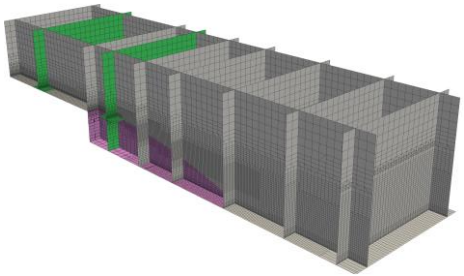
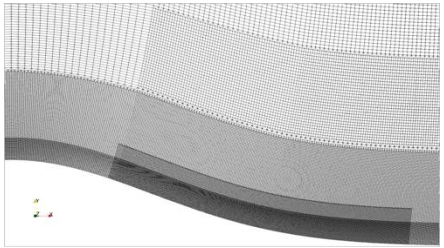
Grid-adaptation for large eddy simulations: a sequence of ideas over 10+ years

Johan Larsson
Department of Mechanical Engineering
University of Maryland

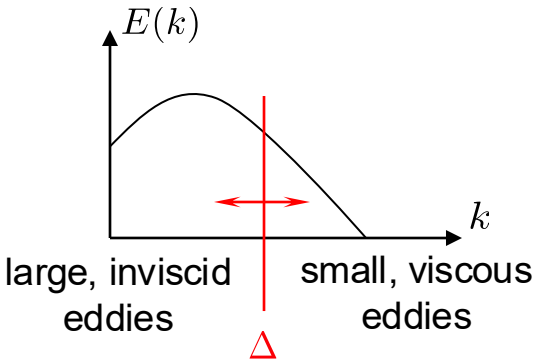
Advanced Modeling & Simulation (AMS) Seminar Series
NASA Ames Research Center
July 1, 2025

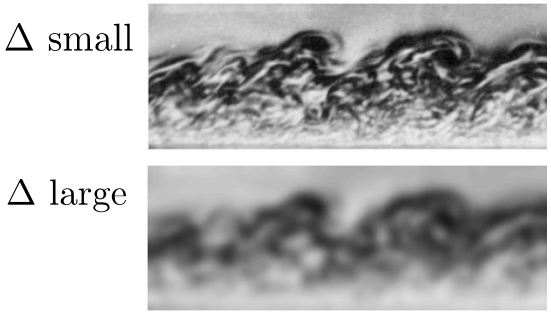


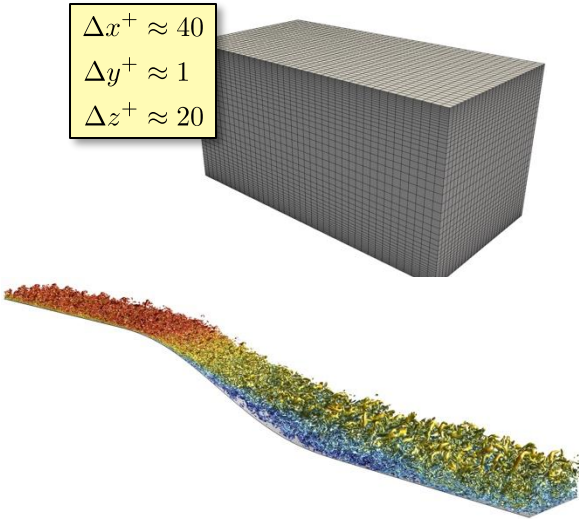


Supported by the NASA Transformational Tools and Technologies (TTT) project, the NSF Fluid Dynamics program, the Naval Air Warfare Center Aircraft Division, the ONR Aerodynamics program, and the NNSA Predictive Science Academic Alliance program (PSAAP-3).
Supercomputing resources provided by the University of Maryland HPC center and the NNSA PSAAP-3 program.

The grid is the most important factor in ~~LES~~ numerical PDEs







Current practice in grid-generation: manual, based on user experience

- requires a lot of human time
- grids may be far from "optimal" – unnecessarily many cells in some regions
- not systematic – different users end up with different grids and possibly different results

What it should be like: automatic, a fully integrated part of the solution process/algorithm

- user should specify the desired accuracy for quantities-of-interest, or the available computational budget

Grid-adaptation in LES is different – the desired solution is “rough”

Numerical analysis: can we speak of the “leading order error term” in LES?

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} = \frac{d^2u}{dx^2} + \frac{h^2}{12}\frac{d^4u}{dx^4} + \frac{h^4}{360}\frac{d^6u}{dx^6} + \dots$$

u fixed, $h \to 0$ $\quad h \to 0,\ (ik)^n \hat{u} \sim h^{-n}\hat{u}$

The residual is the source term of error => grid-adaptation should try to reduce the residual

$$\underbrace{\mathcal{R} \equiv \mathcal{N}_{\text{num}}(u_{\text{exact}})}_{\text{residual}} \approx \underbrace{\mathcal{N}_{\text{num}}(u_{\text{num}})}_{=0} + \underbrace{\frac{\partial \mathcal{N}_{\text{num}}}{\partial u}}_{\text{propagation}} \underbrace{(u_{\text{exact}} - u_{\text{num}})}_{\text{error}}$$

Thought-experiment: what happens if you refine the grid of a typical LES?

resolved + modeled stress = $\overline{u_i' u_j'} - C\Delta^2 |S| S_{ij}$

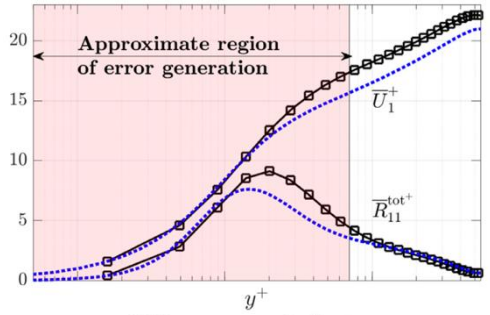
actual LES ↑ ↓ ↑

residual estimate after interpolating onto a refined grid – ↓ –

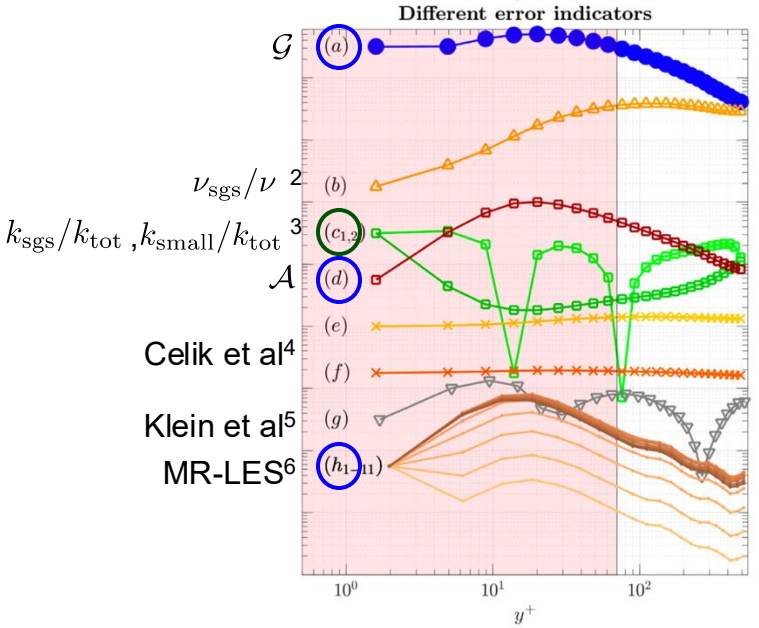
LES residual estimators

A numerical experiment to assess LES residual estimators[1]:
Channel flow with $\Delta/h = 1.6$ to minimize numerical errors

$(\Delta x^+, y_w^+, \Delta z^+) = (70, 1.3, 30)$



The conventional wisdom that LES is accurate
if some fraction (usually assumed to be 80%)
of the energy is resolved is simply wrong



[1] *Toosi & Larsson (CaF 2020)*
[2] *Geurts & Frohlich (PoF 2002)*
[3] *Pope (NJPhys 2004)*
[4] *Celik et al (JoT 2009, JFE 2009)*
[5] *Klein et al (FTC 2005, JoT 2006)*
[6] *Legrand et al (AIAA conf 2018)*

General grid-adaptation process

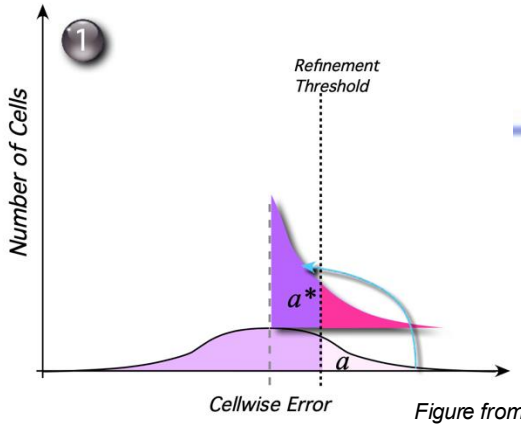


Figure from Nemec et al (AIAA conf 2008)

1. Estimate the residual = source of error = $\mathcal{R}$

   Path A: split cells

   Path B: re-generate grid

B2. Model how the residual would change with a different grid-spacing Δ

$$\mathcal{R}_\Delta \approx c\,\Delta^\alpha \qquad c = \frac{\mathcal{R}_{\Delta_{\text{old}}}}{\Delta_{\text{old}}^\alpha}$$

residual "density" computed from old solution/grid

B3. Find the optimal grid-spacing that minimizes the total residual at a fixed computational cost

$$\mathcal{J} = \iiint c\,\Delta^\alpha dV + \lambda \iiint \frac{1}{\Delta^3} dV$$

choose Lagrange multiplier λ to get desired number of elements

$$\Rightarrow \quad c\,\Delta_{\text{opt}}^\alpha\,\Delta_{\text{opt}}^3 = \frac{3\lambda}{\alpha} = \text{constant}$$

element-integrated residual is equi-distributed in space

B4. Create a new grid with a resolution as close to Δ_{opt} as possible

Attempt #1:
directional small-scale energy

Small-scale energy in a directional sense

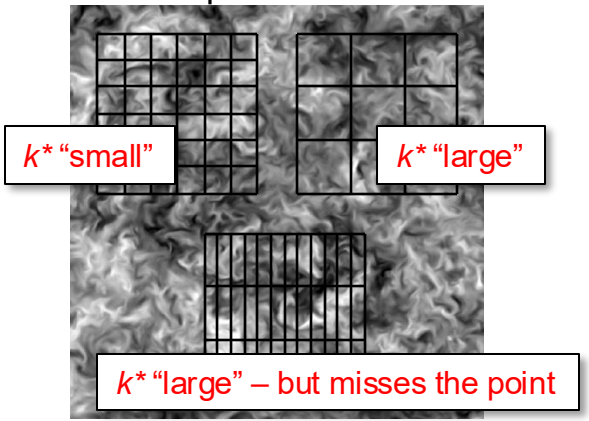
Some approaches (Pope[1], Bose et al[2]) refine based on the energy in the smallest scales

- E.g. (Bose et al): compute energy in high-pass test-filtered (small scale) velocity field

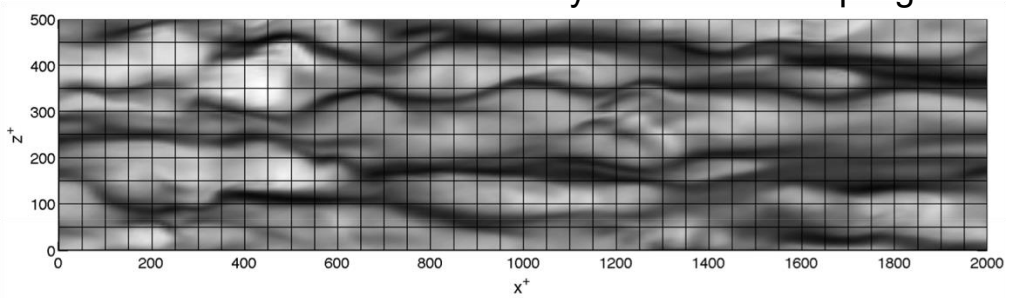
$$\overline{u}_i^* = \overline{u}_i - \widehat{\overline{u}}_i \qquad k^* = \frac{\overline{u}_i^* \, \overline{u}_i^*}{2}$$

and then refine based on *k** – this answers *where* to refine, but not *how to*

Isotropic turbulence[3]



Wall-turbulence in the buffer layer with an isotropic grid



Potential saving of anisotropic grid-adaptation for wall-turbulence starting from isotropic grid:

$$(\Delta x^+, \Delta y^+, \Delta z^+) \approx (1,1,1) \rightarrow (40,1,20)$$

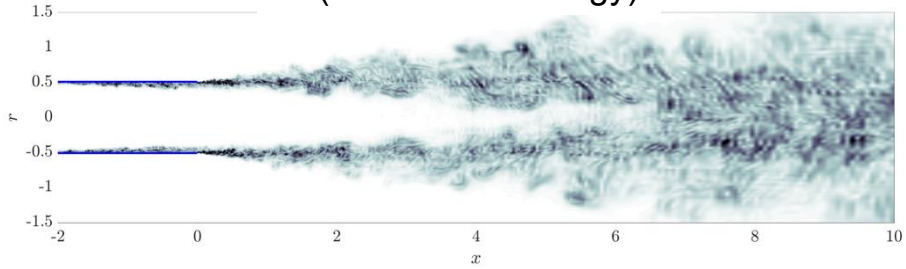
[1] *Pope (NJoP 2004)*
[2] *Bose et al (CTR 2011)*
[3] *Hebert and de Bruyn Kops (PoF 2006)*

In LES, the small-scale energy is not isotropic

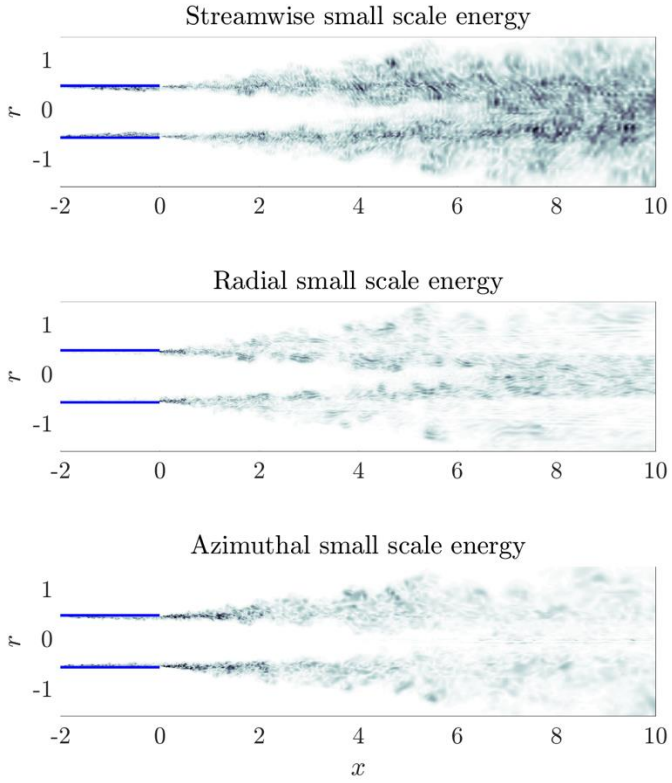
The basic hypothesis of this work:
The directional nature of the smallest resolved scales says something about the required grid-spacing

Example: Mach 0.9 jet by Bres et al[1] on a user-generated unstructured grid

*k** (small-scale energy)



Directional small-scale energy



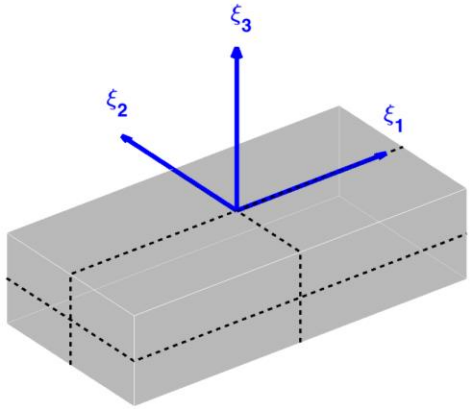
[1] *Bres et al (AIAA 2016-3050)*

Proposed anisotropic error indicator – structured grids

For structured grid, trivial to implement 1D directional filters:

$$\widehat{\overline{u}}_i^{(\xi_1)}, \widehat{\overline{u}}_i^{(\xi_2)}, \widehat{\overline{u}}_i^{(\xi_3)} \qquad \text{(low-pass test-filtered)}$$

$$\overline{u}_i^{(*,\xi_1)} = \overline{u}_i - \widehat{\overline{u}}_i^{(\xi_1)} \qquad \text{(high-pass test-filtered)}$$



For example:

$$\widehat{\overline{u}}_i^{(\xi_1)} = \frac{\overline{u}_i(\xi-\Delta\xi)}{4} + \frac{\overline{u}_i(\xi)}{2} + \frac{\overline{u}_i(\xi+\Delta\xi)}{4}$$

Refine in each direction only if

$$\mathcal{A}(\xi_1) = \sqrt{\left\langle \overline{u}_i^{(*,\xi_1)} \overline{u}_i^{(*,\xi_1)} \right\rangle}$$

is larger than some threshold (and similarly in the other directions)

[1] *Toosi and Larsson (AIAA 2017)*

Proposed anisotropic error indicator – general geometries[1]

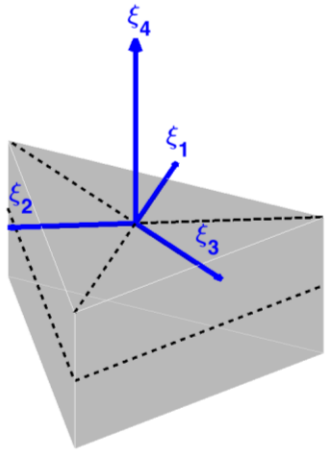
Key difficulty – how to directionally test-filter in arbitrary geometries?

A differential, directional test-filter:

$$\overline{u}_i = \widehat{\overline{u}}_i^{(\xi)} - \frac{\Delta_\xi^2}{4}\,\xi_j\xi_k\,\frac{\partial^2\,\widehat{\overline{u}}_i^{(\xi)}}{\partial x_j \partial x_k}$$

Approximate inversion[2]:

$$\widehat{\overline{u}}_i^{(\xi)} \approx \overline{u}_i + \frac{\Delta_\xi^2}{4}\,\xi_j\xi_k\,\frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_k}$$



High-pass filtered (i.e., small-scale component):

$$\overline{u}_i^{(*,\xi)} = \overline{u}_i - \widehat{\overline{u}}_i^{(\xi)} \approx -\frac{\Delta_\xi^2}{4}\,\xi_j\xi_k\,\frac{\partial^2 \overline{u}_i}{\partial x_j \partial x_k}$$

Error indicator in any direction:

$$\mathcal{A}(\xi) = \sqrt{\left\langle \overline{u}_i^{(*,\xi)}\,\overline{u}_i^{(*,\xi)} \right\rangle}$$

Entirely a post-processing operation!

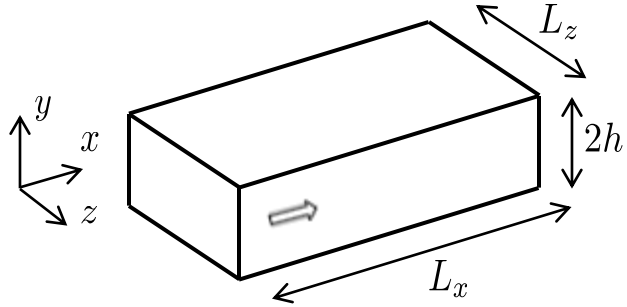
[1] *Toosi and Larsson (AIAA 2017)*
[2] *van Cittert (1931)*

Test for channel flows – where the “optimal” grid is known

$$(L_x, L_y, L_z) = (10h, 2h, 3h)$$

$$Re_b = 30000\ ,\ \ Re_\tau \approx 830$$

$$M = \frac{U_b}{c_w} = 0.8$$



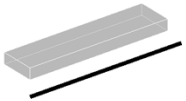
Solution-adaptive finite-difference code,
6th-order central / 5th-order WENO, minimal numerical dissipation

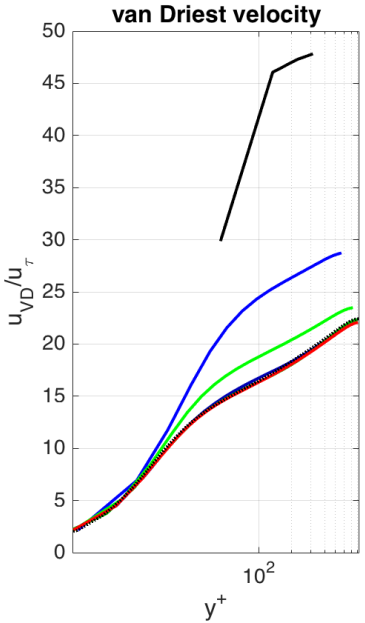
Dynamic Smagorinsky model

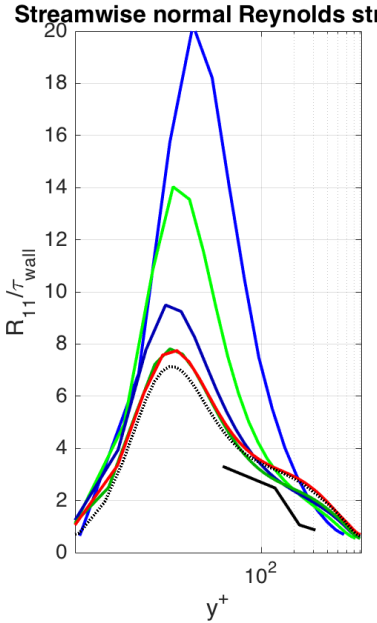
Constant mass flux

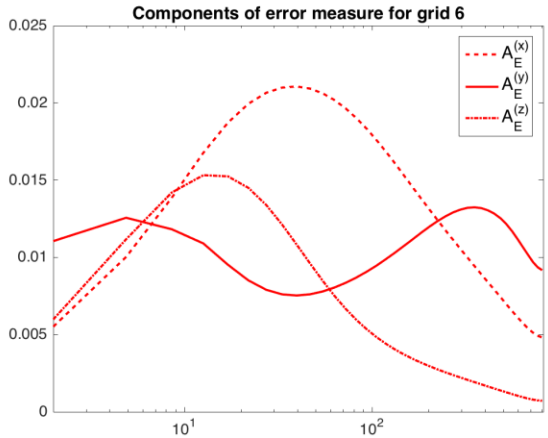
Convergence metric: $$\varepsilon_{prev} = \frac{1}{5}\frac{\int_a^b \left|U_1^+ - U_{1,prev}^+\right| d(\ln y^+)}{\int_a^b U_{1,prev}^+ d(\ln y^+)} + \sum_{R_{ij}} \frac{1}{5}\frac{\int_a^b \left|R_{ij}^+ - R_{ij,prev}^+\right| d(\ln y^+)}{\int_a^b R_{kk,prev}^+/2\, d(\ln y^+)}$$

Grid-adaptation for a channel flow







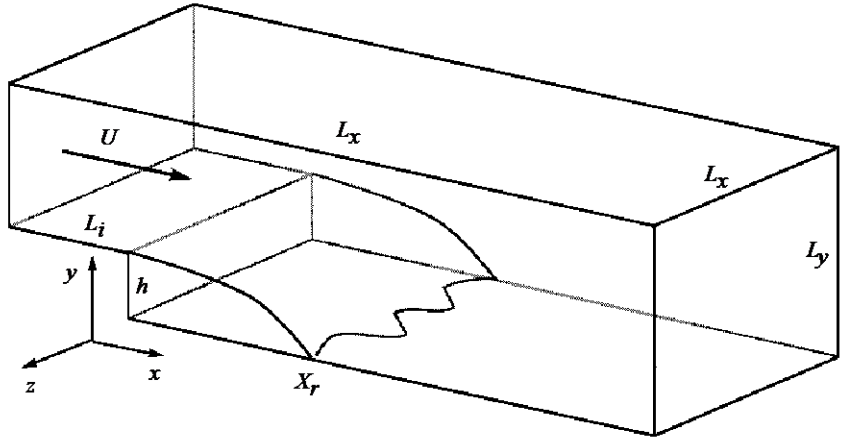


Grid	(N_x, N_y, N_z)	$(\Delta x, \Delta y_c, \Delta z)/\delta$	$(\Delta x^+, y_1^+, \Delta z^+)$	Re_τ	ε_{prev} (%)
LES-1	(40, 8, 12)	(0.25, 0.25, 0.25)	(89, 45, 89)	356	-
LES-2	(35, 32, 15)	(0.29, 0.14, 0.20)	(170, 2.2, 120)	611	200
LES-3	(60, 50, 40)	(0.17, 0.090, 0.075)	(130, 1.7, 57)	758	17
LES-4	(96, 60, 96)	(0.10, 0.075, 0.031)	(85, 1.6, 26)	820	13
LES-5	(152, 72, 180)	(0.066, 0.063, 0.017)	(54, 1.3, 14)	822	5.8
LES-6	(220, 80, 270)	(0.046, 0.052, 0.011)	(38, 1.5, 9.2)	827	2.1

Application to a backward-facing step using *OpenFoam*

Experiment by Jovic & Driver[1]

$Re_h \approx 5000$



Taken from Le et al (JFM 1997)

DNS by Le et al[2]:

- domain [-10h, 20h] x [0, 6h] x [0, 4h]
- about 9.4M grid points, deemed borderline in their paper

Current LES:

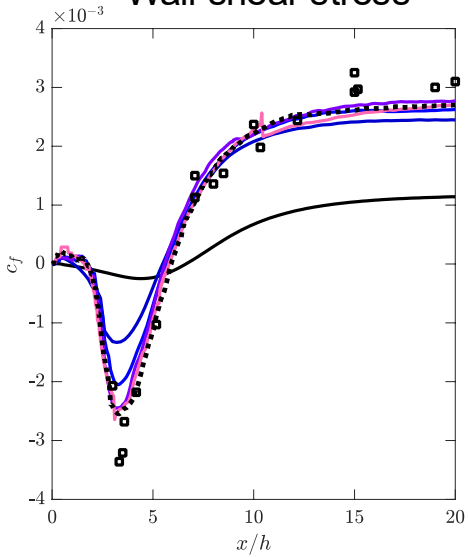
- OpenFOAM with dynamic k-equation subgrid model, central 2nd-order numerics
- domain [-20h, 25h] x [0, 6h] x [0, 4h] – 50% larger than Le et al.
- initial grid ("LES-1"): cubic cells with size 0.2h (125K cells)

[1] *Jovic & Driver (NASA report 1994)*
[2] *Le et al (JFM 1997)*

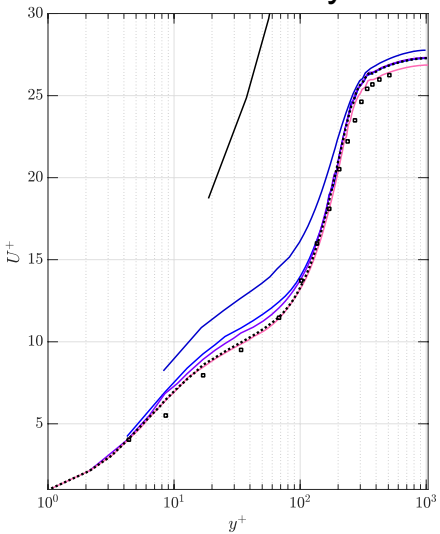
Application to a backward-facing step using *OpenFoam*

Grid	N_{tot}	N_{struc}	at x/h = -3 (upstream of step) $(\Delta x^+, y_1^+, \Delta z^+)$	middle of detached shear layer $(\Delta x, \Delta y, \Delta z)/\delta_{shear}$	ε_{prev} (%)	ε_{ref} (%)
LES-1	125k	125k	(28, 14, 28)	(0.5, 0.5, 0.5)	-	29
LES-2	273k	517k	(40, 2.5, 20)	(0.4, 0.1, 0.2)	23	12
LES-3	694k	2.3M	(46, 2.9, 12)	(0.14, 0.036, 0.071)	8.0	5.2
LES-4	1.7M	11.2M	(24, 1.5, 6.0)	(0.071, 0.018, 0.036)	3.2	4.9
LES-5	3.2M	16.2M	(24, 1.5, 6.1)	(0.036, 0.018, 0.036)	3.1	3.3
DNS	9.1M	70M	(12, 0.76, 6.1)	(0.036, 0.0089, 0.036)	-	-

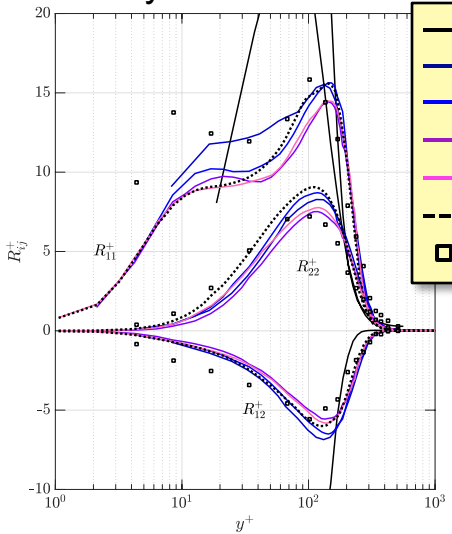
Wall shear stress

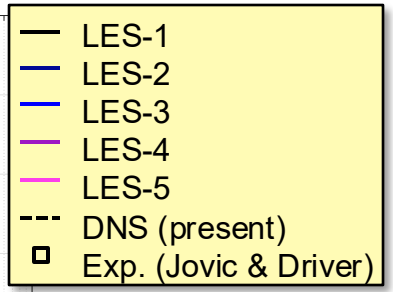


Mean velocity



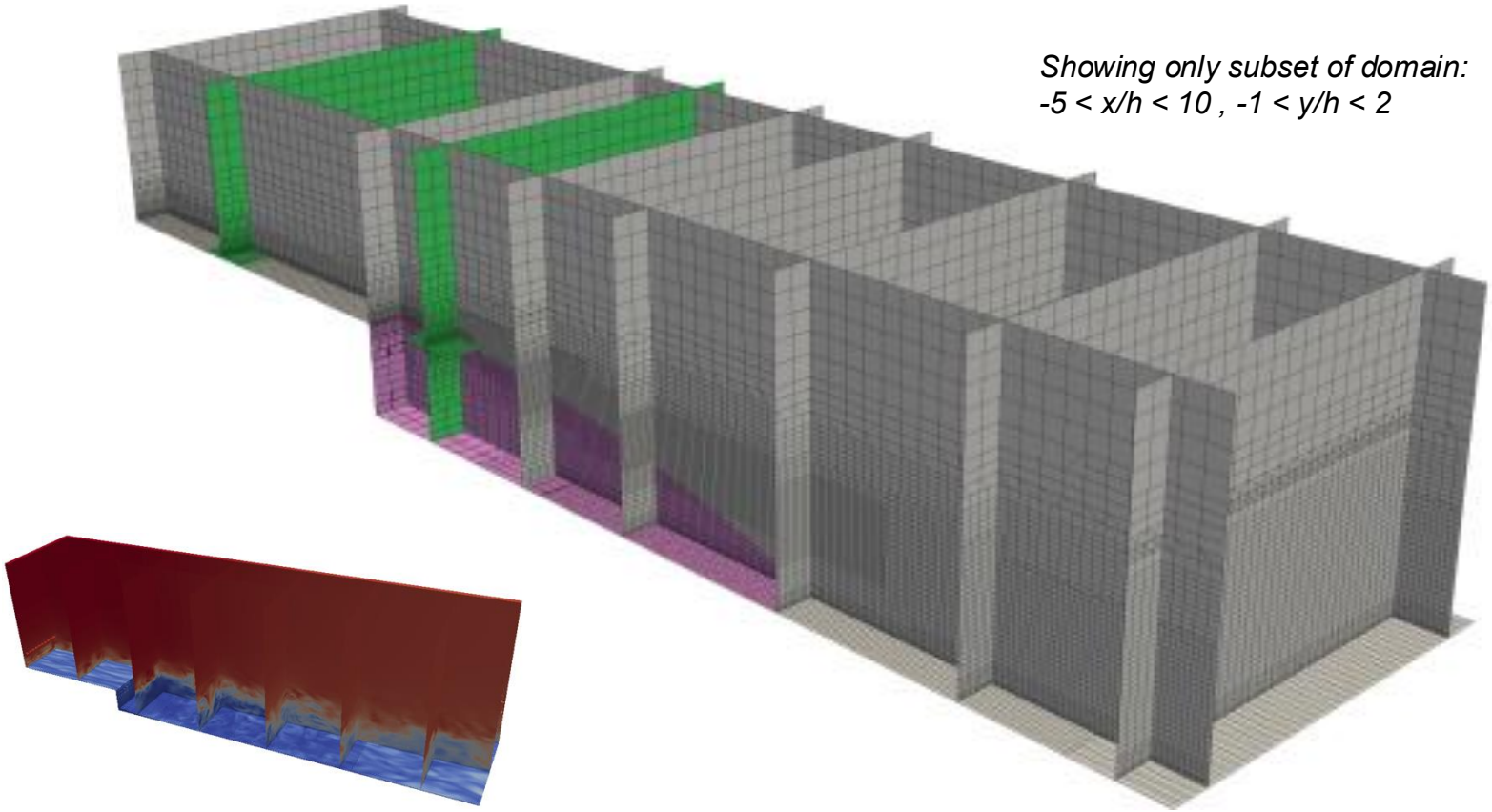
Reynolds stresses





Application to a backward-facing step using *OpenFoam*

Third adaptation (fourth grid), 5% error compared to DNS
1.7M cells, generated “automatically” from the isotropic initial grid



Showing only subset of domain:
-5 < x/h < 10 , -1 < y/h < 2

Attempt #4:
estimated LES modeling residual

Estimating the LES modeling residual from the governing equation[1]

Equation: $\mathcal{N}_{\text{num}}(u_{\text{num}}) = 0$

$$\frac{\partial u}{\partial t} = \mathcal{F}(u)$$

$$\mathcal{N}(u) = \mathcal{F}(u) - \frac{\partial u}{\partial t}$$

Residual:
$$\underbrace{\mathcal{R} \equiv \mathcal{N}_{\text{num}}(u_{\text{exact}})}_{\text{residual}} \approx \underbrace{\mathcal{N}_{\text{num}}(u_{\text{num}})}_{=0} + \underbrace{\frac{\partial \mathcal{N}_{\text{num}}}{\partial u}}_{\text{propagation}} \underbrace{(u_{\text{exact}} - u_{\text{num}})}_{\text{error}}$$

LES residual:
$$\mathcal{R}_{\widehat{\overline{\Delta}}} \equiv \mathcal{N}_{\text{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\text{exact}}\right) \approx \mathcal{N}_{\text{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\text{num}}\right)$$
$$= \mathcal{F}_{\text{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\text{num}}\right) - \frac{\partial \widehat{\overline{u}}_{\text{num}}}{\partial t}$$
$$= \mathcal{F}_{\text{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\text{num}}\right) - \widehat{\mathcal{F}_{\text{num},\overline{\Delta}}(\overline{u}_{\text{num}})}$$
$$= \frac{\partial}{\partial x_j}\left(\widehat{\overline{u}_i \overline{u}_j} - \widehat{\overline{u}}_i \widehat{\overline{u}}_j + \widehat{\tau_{ij,\overline{\Delta}}(\overline{u}_k)} - \tau_{ij,\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_k\right)\right)$$

How to implement in a code with explicit time-stepping:

1. Call $\mathcal{F}$ with LES solution, low-pass-filter the result $\widehat{\mathcal{F}_{\text{num},\overline{\Delta}}(\overline{u}_{\text{num}})}$
2. Low-pass-filter the LES solution, call $\mathcal{F}$ with coarsened subgrid-model length scale $\mathcal{F}_{\text{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\text{num}}\right)$

$\overline{u}$ = LES solution

$\widehat{\overline{u}}$ = low-pass-filtered LES solution

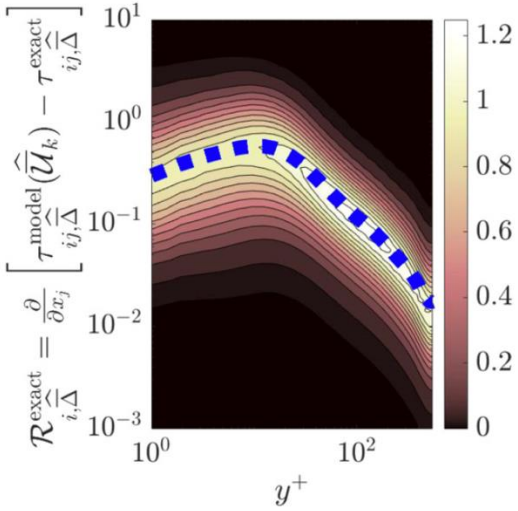
[1] *Toosi & Larsson (CaF 2020, JCP 2021)*

A posteriori assessment of the estimated LES modeling residual[1]

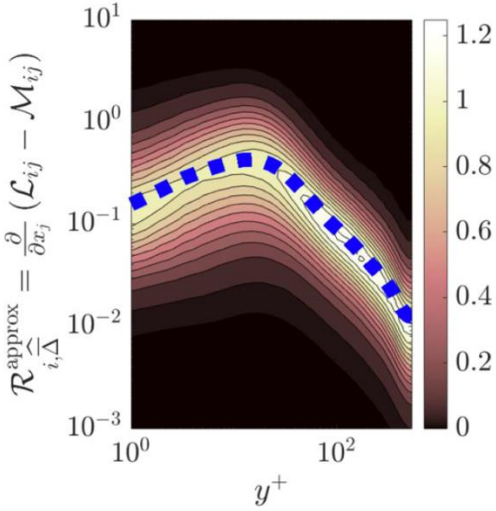
PDFs of the LES modeling residuals in channel flow

$$\mathcal{R}_{\widehat{\overline{\Delta}}} \equiv \mathcal{N}_{\mathrm{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\mathrm{exact}}\right) \approx \mathcal{N}_{\mathrm{num},\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_{\mathrm{num}}\right)$$

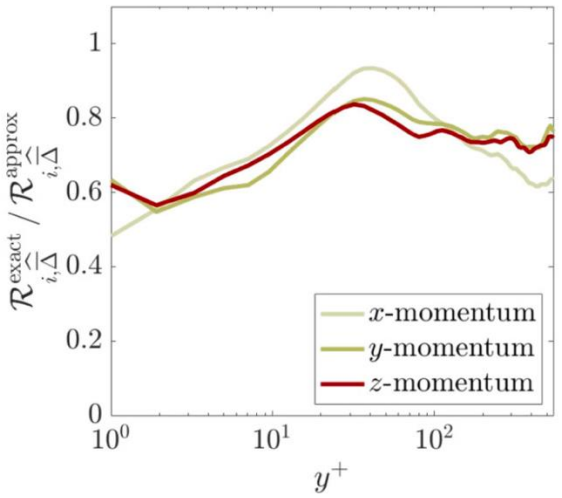
LES residual computed from DNS



LES residual estimated from LES



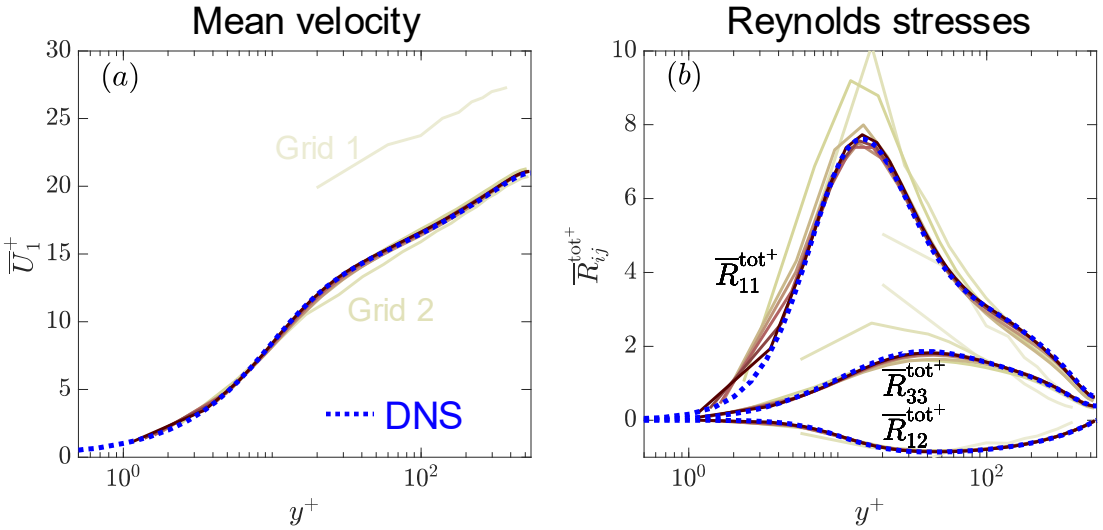
Ratio DNS/LES

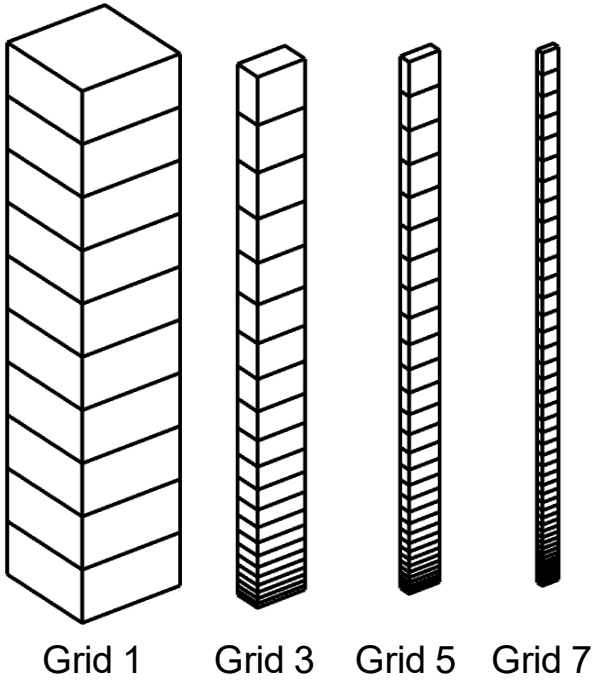


[1] *Toosi & Larsson (JCP 2021)*

Application to a channel flow[1]

Grid	N_{tot}	N_y	$(\overline{\Delta}_x^+, \overline{y}_1^+, \overline{\Delta}_z^+)$	$(\overline{\Delta}_x, \overline{\Delta}_{y,c}, \overline{\Delta}_z)/H$	Re_τ	$e_{\mathrm{QoI}}^{\mathrm{DNS}}$ (%)	$\mathcal{A}(\mathbf{x}, \mathbf{n})$
DSM-1	$15k$	20	$(80, 20, 80)$	$(0.20, 0.10, 0.20)$	398	32	32
DSM-2	$74k$	34	$(77, 5.6, 55)$	$(0.14, 0.099, 0.010)$	553	12	9.2
DSM-3	$251k$	44	$(53, 2.3, 29)$	$(0.098, 0.091, 0.054)$	536	6.5	7.9
DSM-4	$514k$	50	$(45, 1.7, 19)$	$(0.082, 0.080, 0.035)$	544	3.3	5.2
DSM-5	$1.18M$	60	$(34, 1.4, 13)$	$(0.063, 0.065, 0.024)$	544	1.8	4.3
DSM-6	$2.53M$	72	$(25, 1.6, 10)$	$(0.046, 0.052, 0.018)$	542	1.5	2.7
DSM-7	$5.80M$	90	$(18, 1.4, 7.6)$	$(0.033, 0.041, 0.014)$	540	< 1	< 1
DSM-8	$11.1M$	108	$(14, 1.2, 6.3)$	$(0.025, 0.033, 0.012)$	541	< 1	

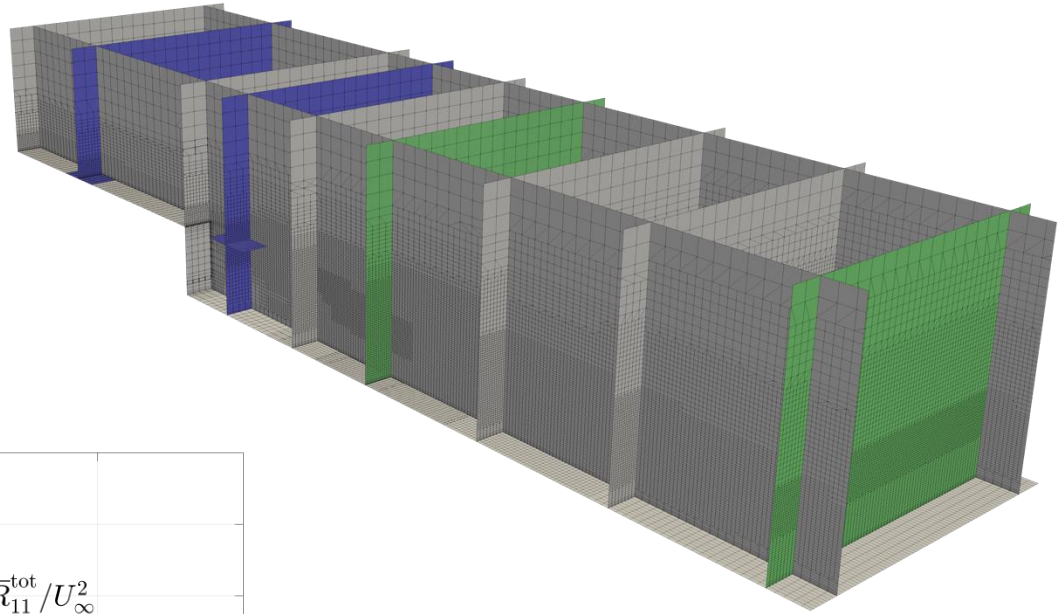


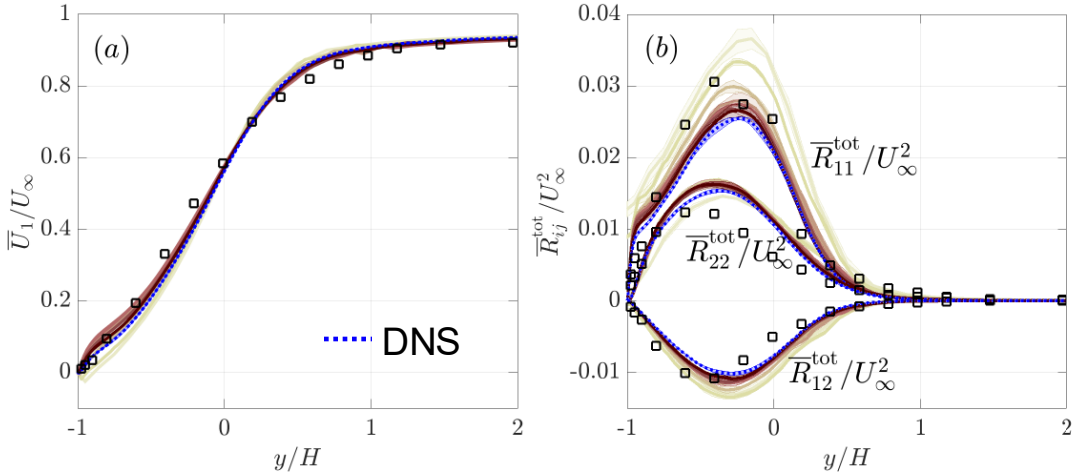


[1] *Toosi & Larsson (CaF 2020, JCP 2021)*

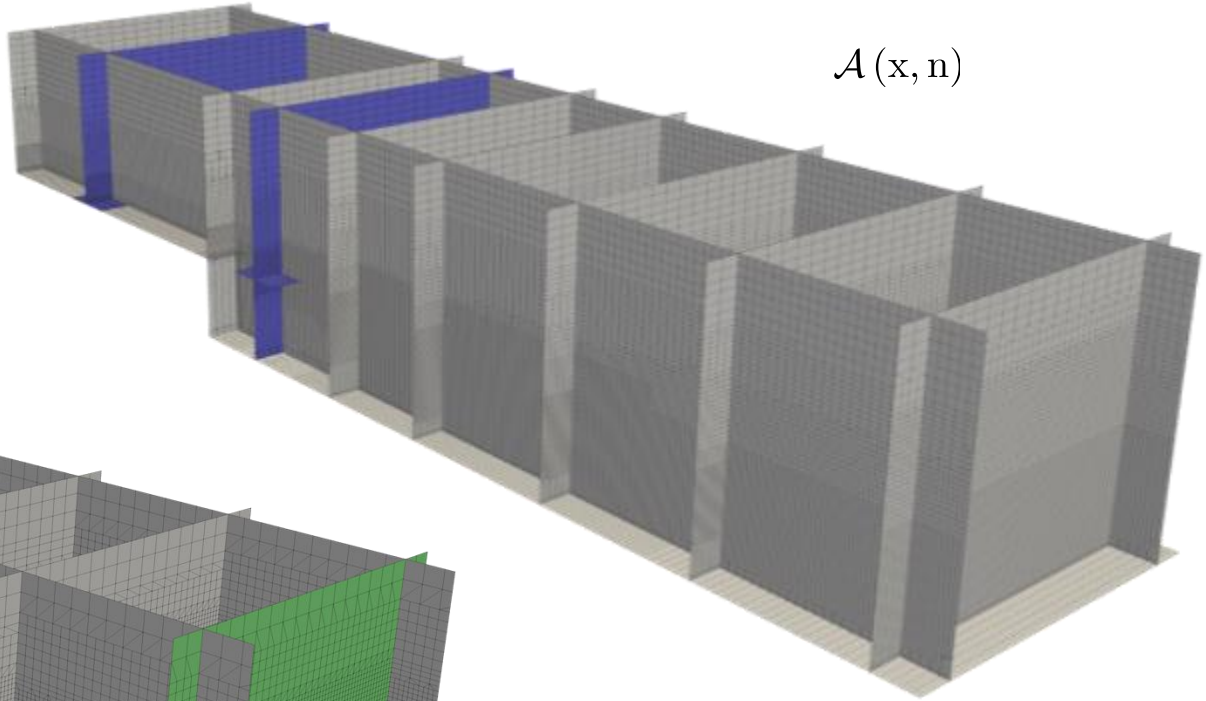
Application to a backward-facing step using *OpenFoam*

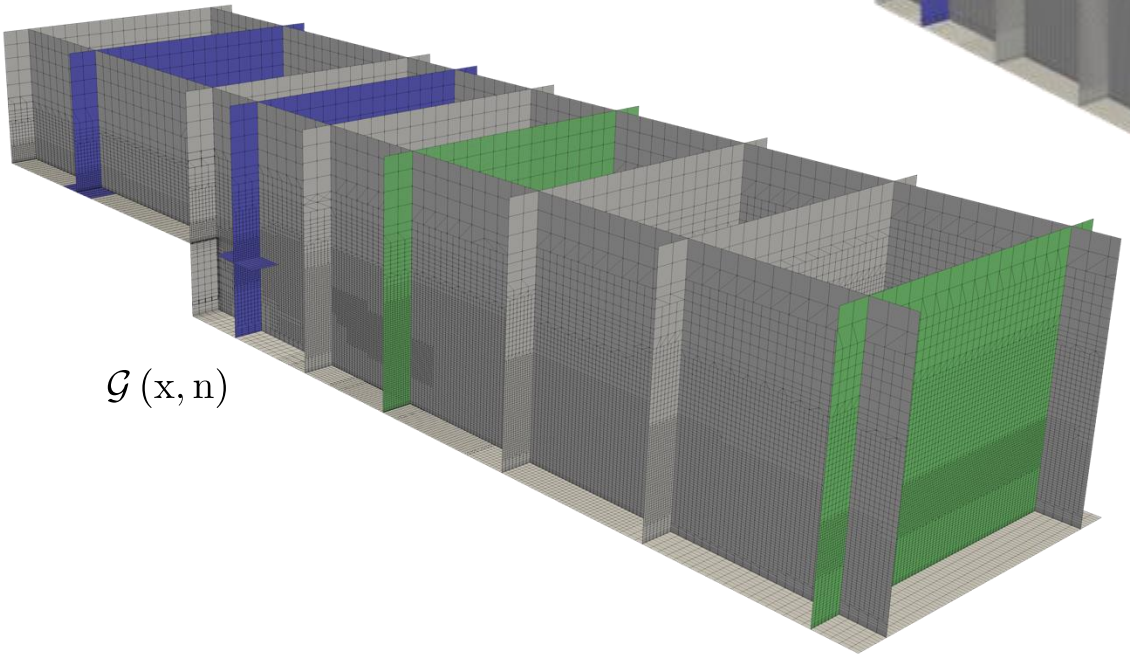
Grid	N_{tot}	$(\overline{\Delta}_x^+, \overline{y}_1^+, \overline{\Delta}_z^+)$	$(\overline{\Delta}_x, \overline{\Delta}_y, \overline{\Delta}_z)/\delta_{\text{shear}}$	$e_{\text{QoI}}^{\text{DNS}}$ (%)
G-1	149k	$(42, 10, 42)$	$(0.21, 0.17, 0.33)$	11.1
G-2	297k	$(42, 2.6, 21)$	$(0.16, 0.078, 0.16)$	10.5
G-3	611k	$(45, 1.4, 11)$	$(0.16, 0.049, 0.078)$	5.6
G-4	1.32M	$(47, 1.5, 12)$	$(0.076, 0.038, 0.076)$	4.9
G-5	2.13M	$(25, 0.77, 6.2)$	$(0.070, 0.035, 0.035)$	5.4
G-6	3.41M	$(25, 0.77, 6.1)$	$(0.068, 0.034, 0.034)$	3.5
G-7	6.72M	$(12, 0.76.6.0)$	$(0.034, 0.017, 0.034)$	2.5
DNS	54M	$(6.0, 0.38.3.0)$	$(0.017, 0.0086, 0.017)$	0





Comparison of the final grids

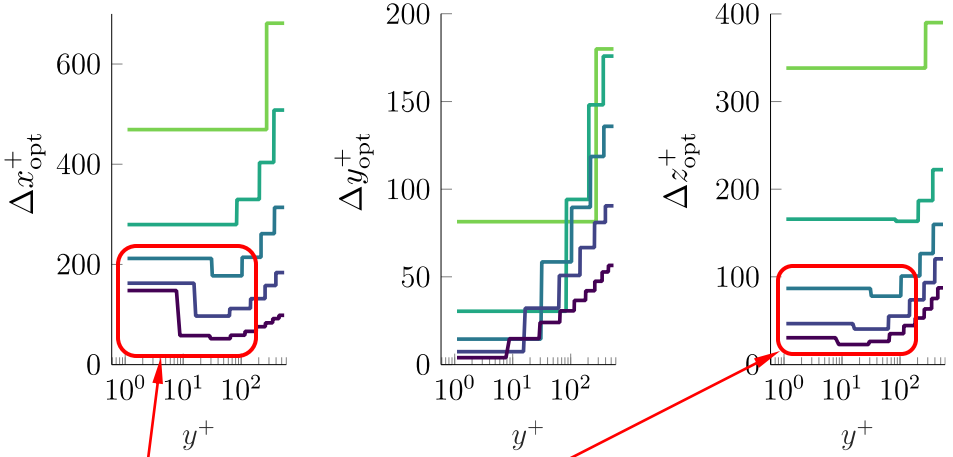




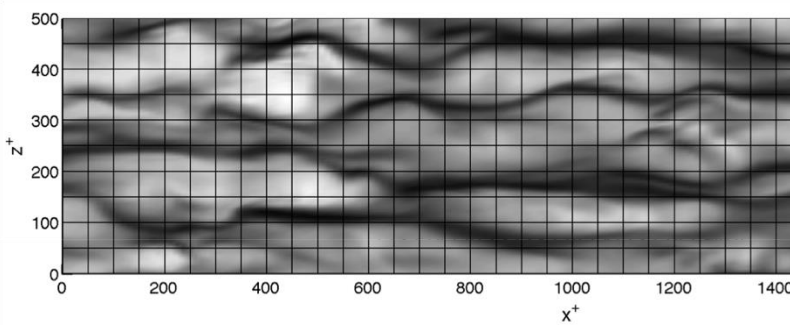
The grid-adaptation picks up subtle near-wall behavior[1]

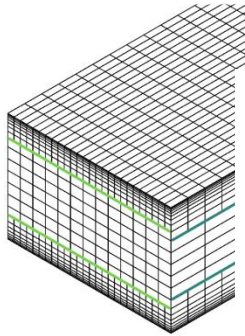
Channel at $Re_\tau \approx 550$
DGSEM code *Flexi*[2] with 5th order polynomials

Algorithm-suggested element sizes for a sequence of 5 grids

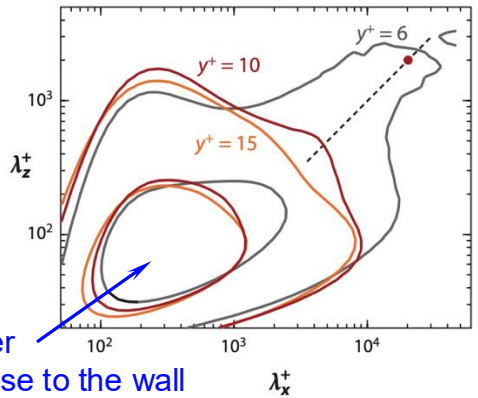


Algorithm suggests larger elements in the wall-parallel directions very close to the wall





Spectral density of near-wall vorticity[3]



Shift towards larger structures very close to the wall

[1] *Blind et al (CaF 2023)*
[2] *https://www.flexi-project.org*
[3] *Jimenez (Annu Rev 2012)*

Some applications

Separated flow over a smooth ramp

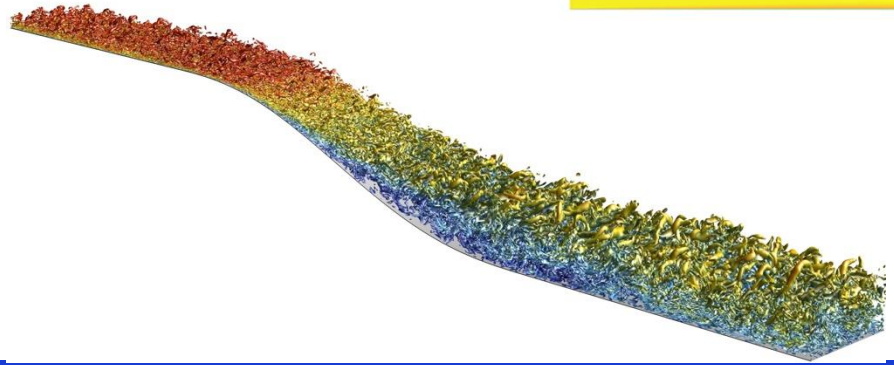
Flow used in the High-Fidelity CFD Verification workshop 2022 and 2024

Ali Uzun DNS (blinded results until the January 2024 workshop):

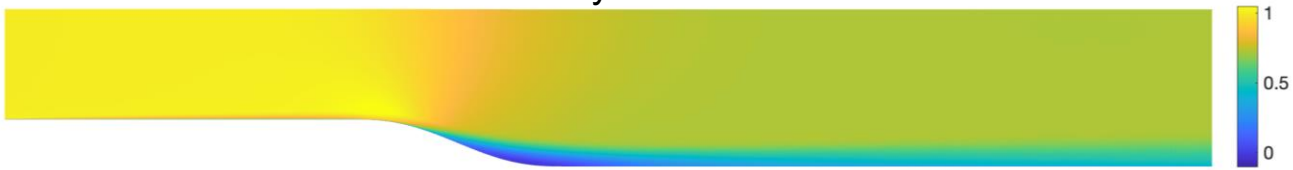
$\mathrm{M} = 0.2, Re_L = 667000, Re_\theta \approx 2000$

$L_z = 12\delta_{\mathrm{in}} = 9\delta_{\mathrm{ref}} = 0.288L$

Initial grid with 40M cells created based on user expertise
Simulation run with WMLES



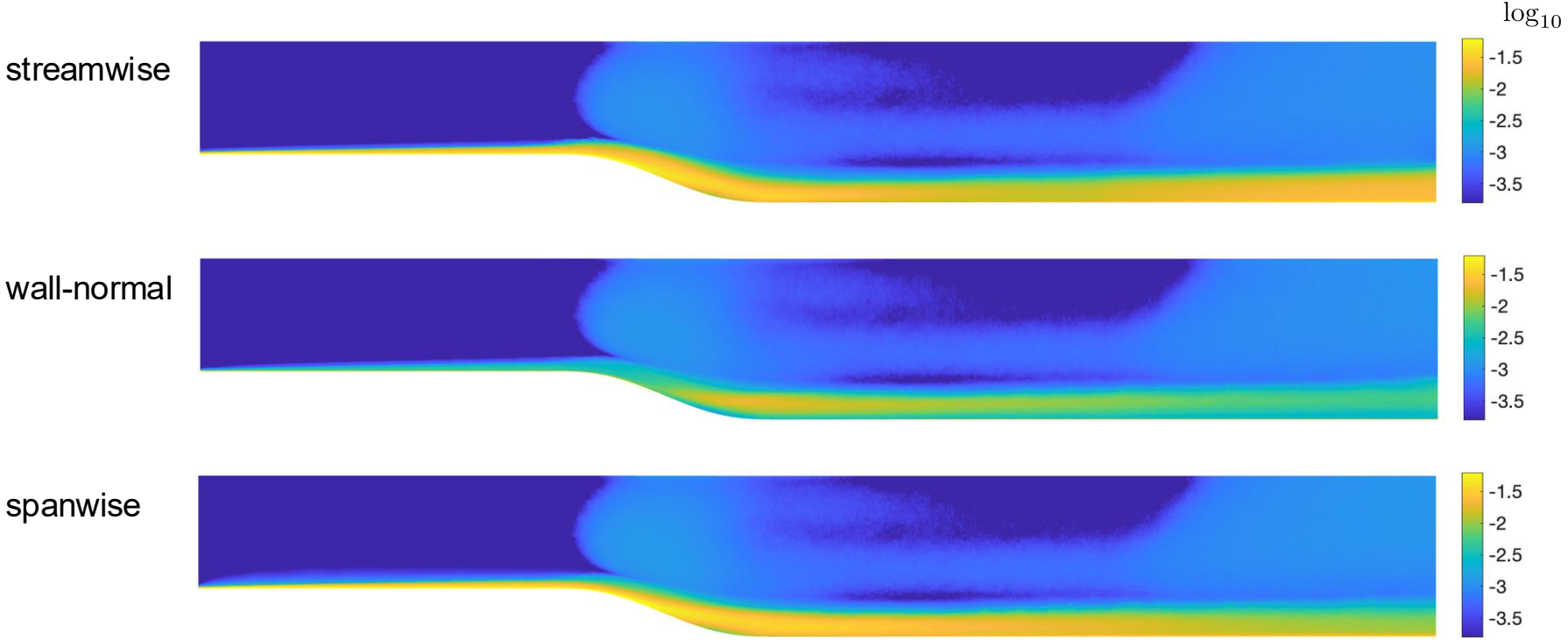
Mean velocity in x-direction



Separated flow over a smooth ramp[1]

Estimated residuals caused by insufficient resolution in each direction[2]

$$\mathcal{R}_{\widehat{\overline{\Delta}}} \approx \frac{\partial}{\partial x_j}\left(\widehat{\overline{u_i}\overline{u_j}} - \widehat{\overline{u}}_i\widehat{\overline{u}}_j + \widehat{\tau_{ij,\overline{\Delta}}(\overline{u}_k)} - \tau_{ij,\widehat{\overline{\Delta}}}\left(\widehat{\overline{u}}_k\right)\right)$$



[1] *Oberoi et al (IJCFD 2024)*
[2] *Toosi & Larsson (CaF 2020, JCP 2021)*

Separated flow over a smooth ramp[1]

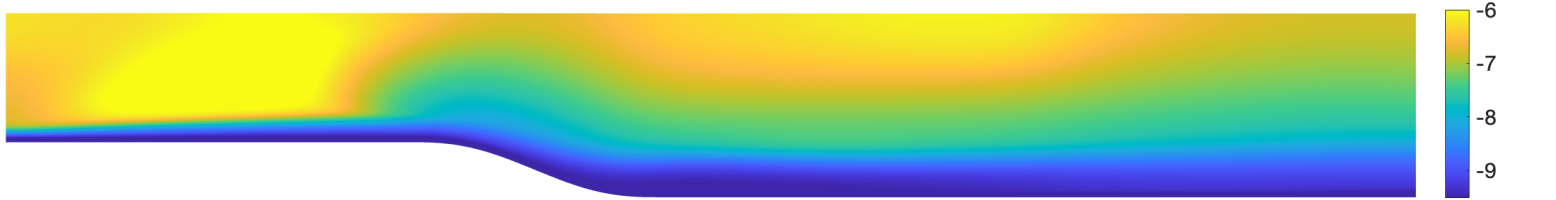
Estimated optimal grid-spacings along the old grid lines

$$\mathcal{J} = \iiint c\,\Delta^{\alpha} dV + \lambda \iiint \frac{1}{\Delta^3} dV \quad \Rightarrow \quad c\,\Delta_{\text{opt}}^{\alpha}\,\Delta_{\text{opt}}^{3} = \frac{3\lambda}{\alpha} = \text{constant}$$

streamwise



wall-normal



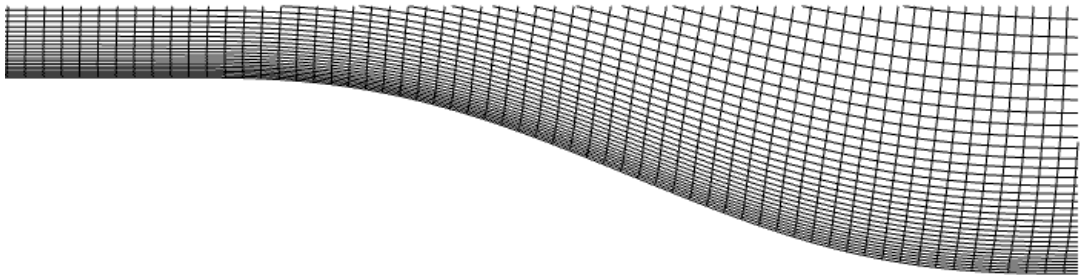
spanwise



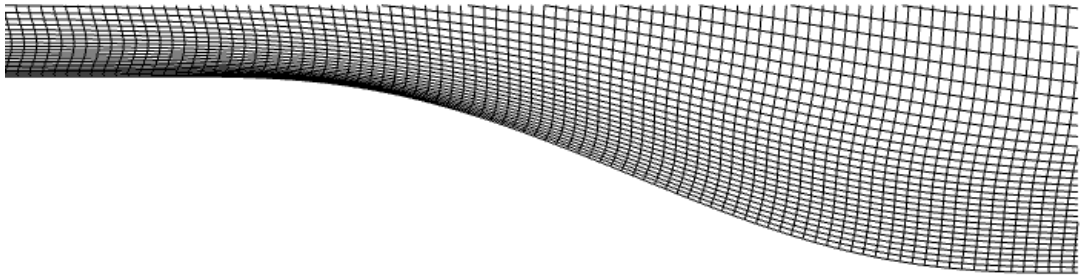
[1] *Oberoi et al (IJCFD 2024)*

Separated flow over a smooth ramp[1]

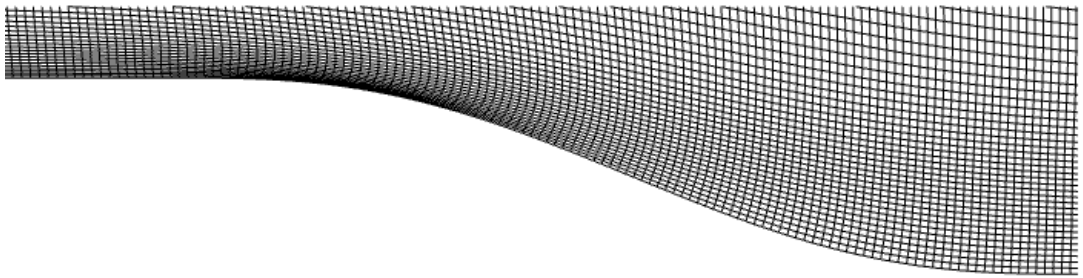
gridA
$1380 \times 223 \times 130$



gridB
$1907 \times 226 \times 202$



gridC
$2627 \times 332 \times 242$



[1] *Oberoi et al (IJCFD 2024)*

WMLES of airfoil with unstructured grid in *x-y* and mortars in *z*[1]

NACA 64A-110 airfoil at Mach 0.72 and Re_c=930,000

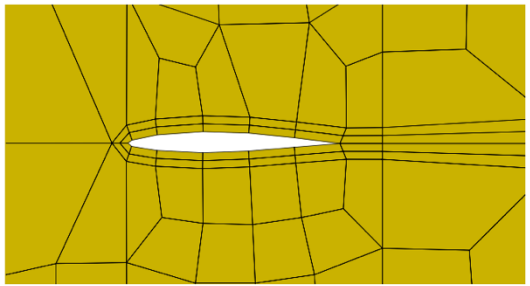
DGSEM code *Flexi*[2] with 5th order polynomials

2D unstructured quad mesh generated in ANSA following the suggested spacings as closely as possible

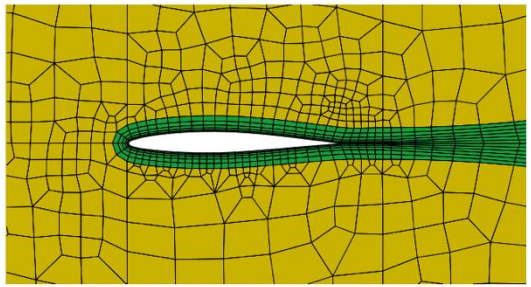
Wall-model exchange location held fixed at $h_{\mathrm{wm}} = 0.00134c$

Sequence of meshes

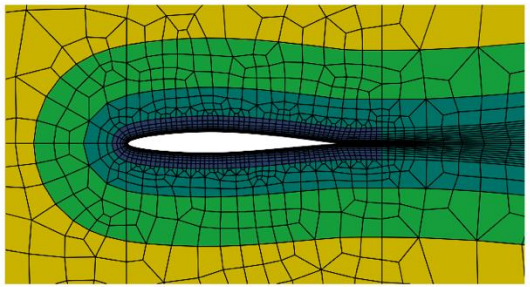
The color shows the spanwise grid-spacing



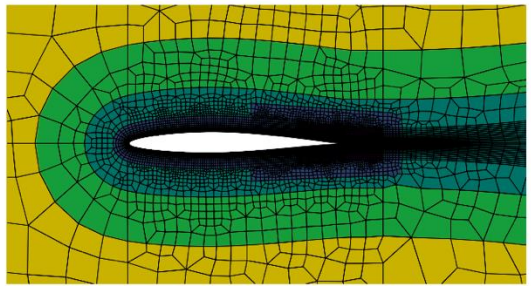
(a) Initial mesh $l = 0$ around the airfoil.



(b) First refinement iteration $l = 1$.



(c) Second refinement iteration $l = 2$.



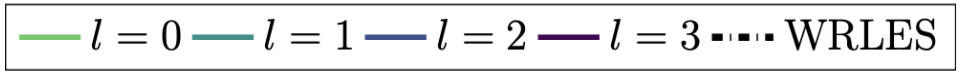
(d) Third refinement iteration $l = 3$.

[1] *Blind et al (CaF 2023)*

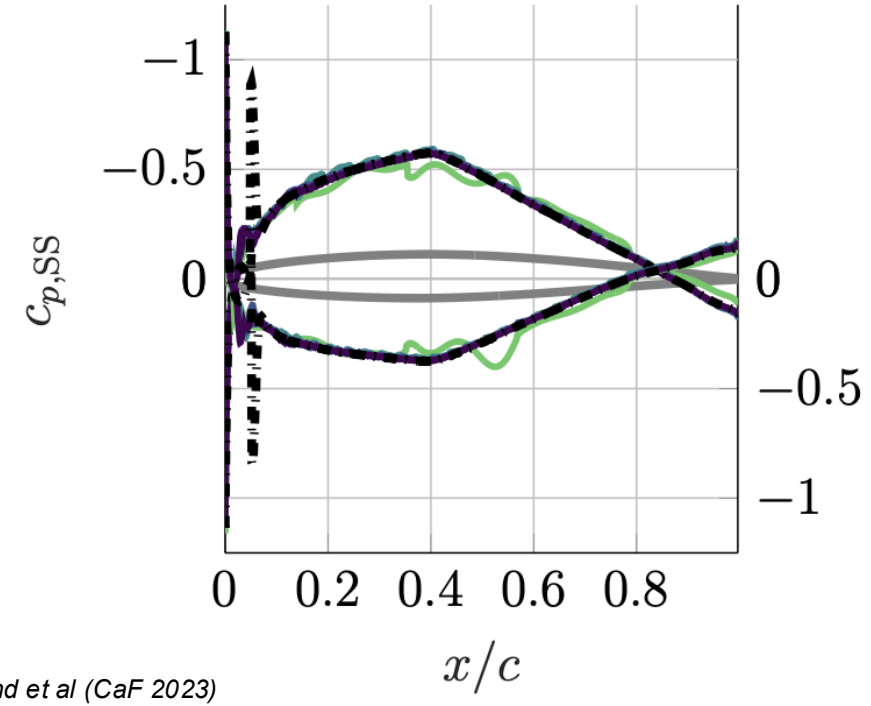
[2] *https://www.flexi-project.org*

WMLES of airfoil with unstructured grid in *x*-*y* and mortars in *z*[1]

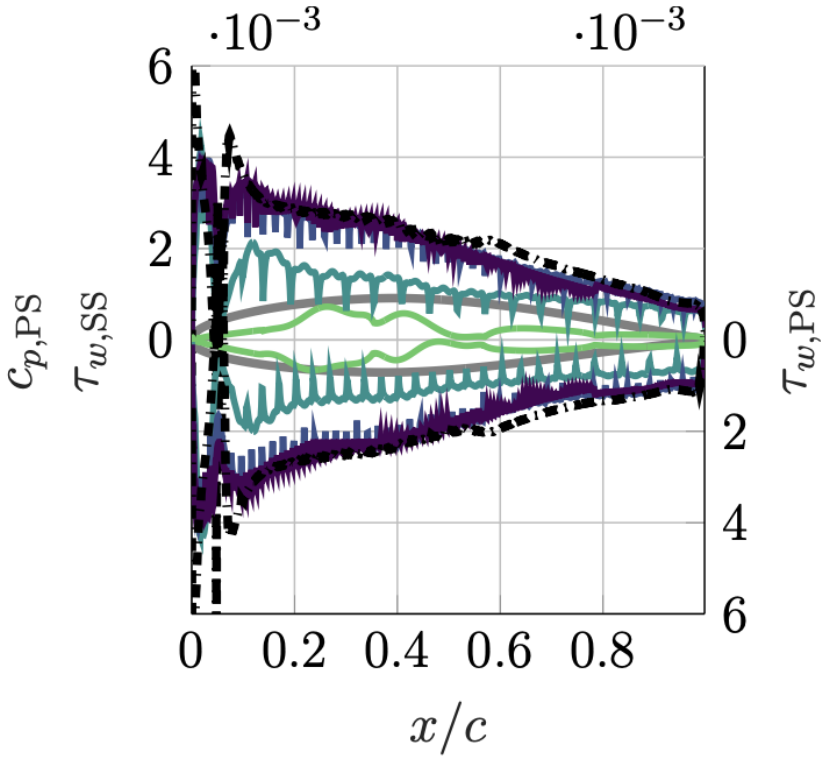
SS = suction side
PS = pressure side



Pressure coefficient



Friction coefficient

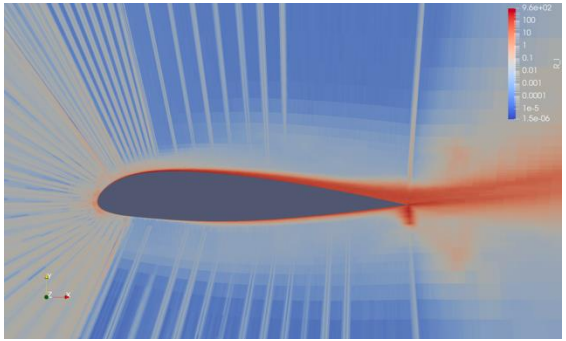
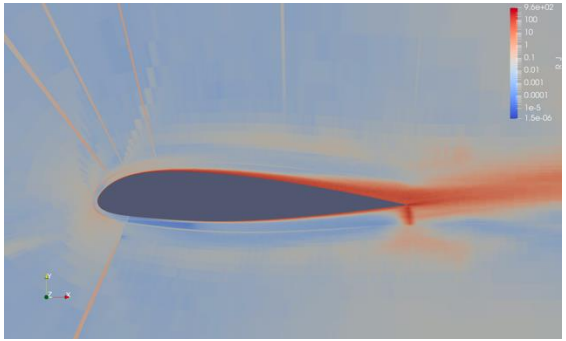
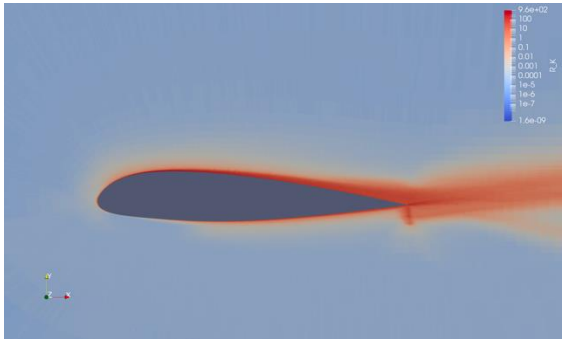


[1] *Blind et al (CaF 2023)*

WMLES of an airfoil with a block-structured grid

streamwise | wall-normal | spanwise

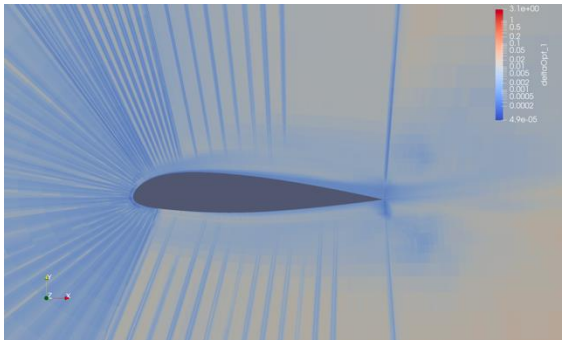
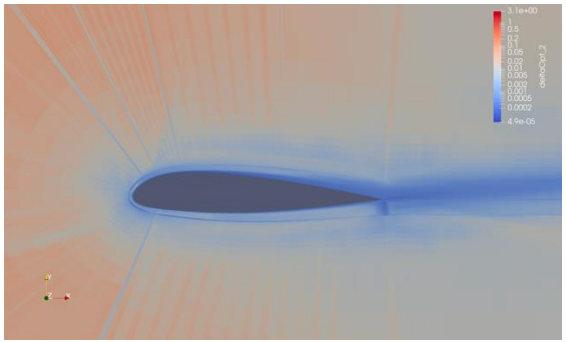
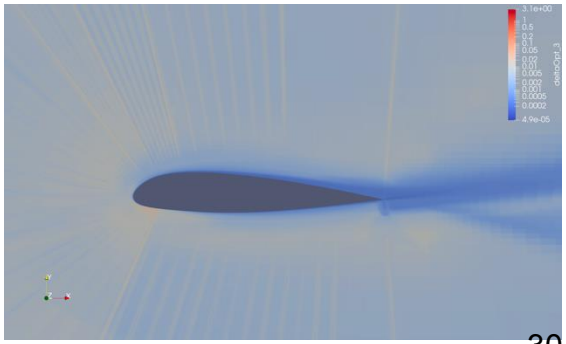
residuals

Minimize the domain-integrated residuals:

$$\mathcal{J} = \iiint c\,\Delta^{\alpha} dV + \lambda \iiint \frac{1}{\Delta^3} dV \quad \Rightarrow \quad c\,\Delta_{\text{opt}}^{\alpha}\,\Delta_{\text{opt}}^{3} = \frac{3\lambda}{\alpha} = \text{constant}$$

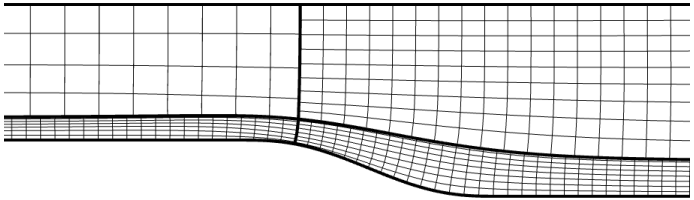
optimal grid-spacings

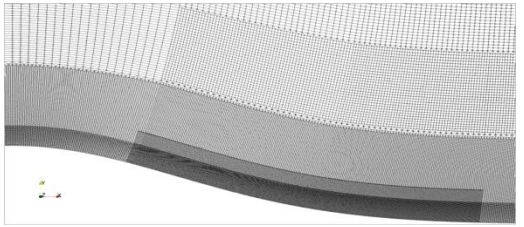




Future/ongoing directions

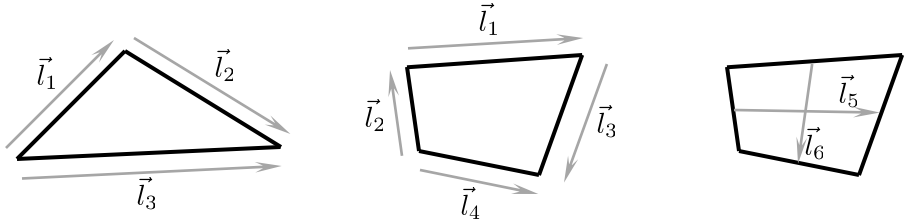
Generation of metric-conforming structured grids

Adapted *structured* grids????





Definition[1-4]: a metric-conforming grid has $\vec{l}^T M \vec{l} = 1$ for every edge in every element



The Riemannian metric *M* encodes the desired size and orientation:

$$M = X\Lambda X^T \qquad \Lambda = \mathrm{diag}\left(\Delta_{1,\mathrm{opt}}^{-2}, \Delta_{2,\mathrm{opt}}^{-2}, \Delta_{3,\mathrm{opt}}^{-2}\right) \qquad X = \left(\vec{e}_{1,\mathrm{opt}}, \vec{e}_{2,\mathrm{opt}}, \vec{e}_{3,\mathrm{opt}}\right)$$

The metric links the residual and the new grid:

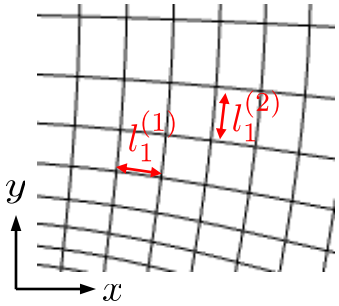
residual estimate → balance error(s) and cost → M → create grid → new grid

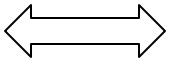
[1] *George et al (Adv Eng Software 1991)*
[2] *Castro-Diaz et al (Int J Num Meth Flu 1997)*
[3] *Frey & Alauzet (Comp Meth Appl Mech Eng 2005)*
[4] *Fidkowski & Darmofal (AIAA J 2011)*

Generation of metric-conforming structured grids[1]

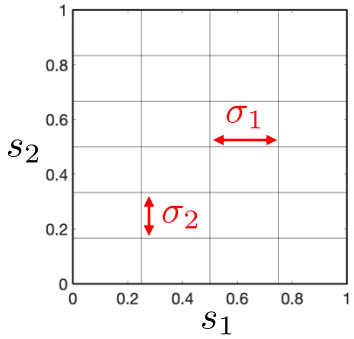
Structured grid coordinate mapping:

Physical space





Computational space



Grid-spacing in α direction: $$l_i^{(\alpha)} = \frac{\partial x_i}{\partial s_\alpha}\sigma_\alpha, \ \alpha = 1,2,3, \ (\text{no sum on } \alpha)$$

Criterion on desired grid-spacing: $$l_i^{(\alpha)} M_{ij} l_j^{(\alpha)} = 1, \ \alpha = 1,2,3, \ (\text{no sum on } \alpha)$$

Misfit in grid-spacing: $$m_\alpha = \sigma_\alpha^2 M_{ij} \frac{\partial x_i}{\partial s_\alpha}\frac{\partial x_j}{\partial s_\alpha} - 1, \ \alpha = 1,2,3, \ (\text{no sum on } \alpha)$$

[1] *Oberoi, Hasan & Larsson (Int J CFD 2024)*

Generation of metric-conforming structured grids[1]

Minimize the misfit in grid-spacing – produces a cost functional

$$\mathcal{L} = \iiint \sum_{\alpha=1}^{3} m_\alpha^2 \, ds_1 ds_2 ds_3$$

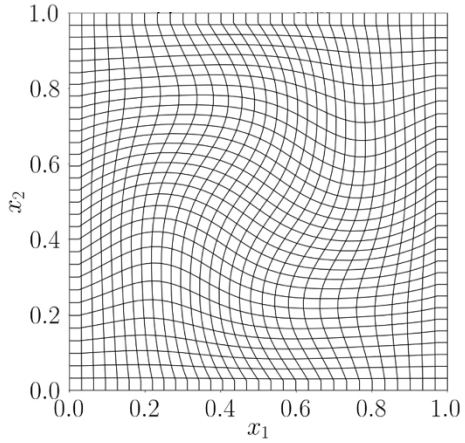
Calculus of variations – the solution is the Euler-Lagrange equation

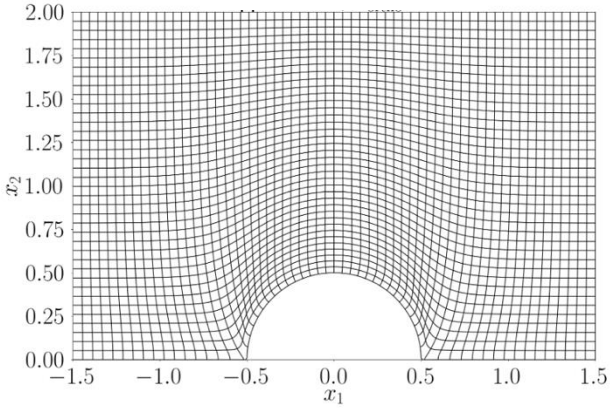
$$-\sum_\alpha 8\sigma_\alpha^4 M_{kl} M_{ij} \frac{\partial x_l}{\partial s_\alpha} \frac{\partial x_i}{\partial s_\alpha} \frac{\partial^2 x_j}{\partial s_\alpha^2} - \sum_\alpha 4\sigma_\alpha^4 M_{kl} \frac{\partial M_{ij}}{\partial x_p} \frac{\partial x_l}{\partial s_\alpha} \frac{\partial x_p}{\partial s_\alpha} \frac{\partial x_i}{\partial s_\alpha} \frac{\partial x_j}{\partial s_\alpha}$$
$$- m_\alpha \sigma_\alpha^2 \left(4M_{kj} \frac{\partial^2 x_j}{\partial s_\alpha^2} + 4 \frac{\partial M_{kj}}{\partial x_p} \frac{\partial x_p}{\partial s_\alpha} \frac{\partial x_j}{\partial s_\alpha} - 2 \frac{\partial M_{ij}}{\partial x_k} \frac{\partial x_i}{\partial s_\alpha} \frac{\partial x_j}{\partial s_\alpha} \right) = 0, \;\; k = 1, 2, 3.$$

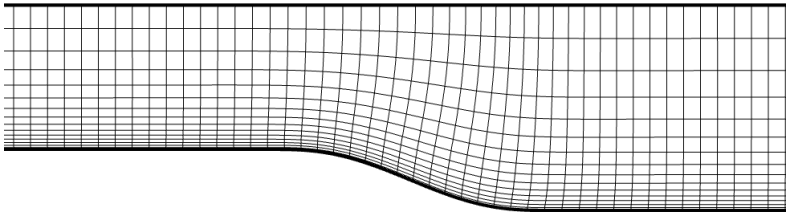
Can be solved for given metric field, producing a coordinate mapping $x_i(s_j)$ and σ_α

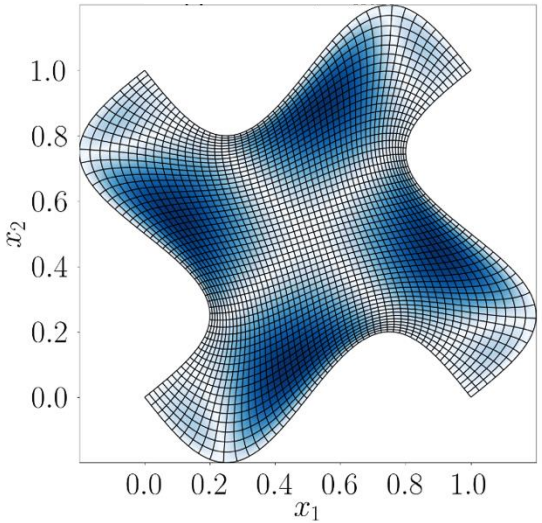
[1] *Oberoi, Hasan & Larsson (Int J CFD 2024)*

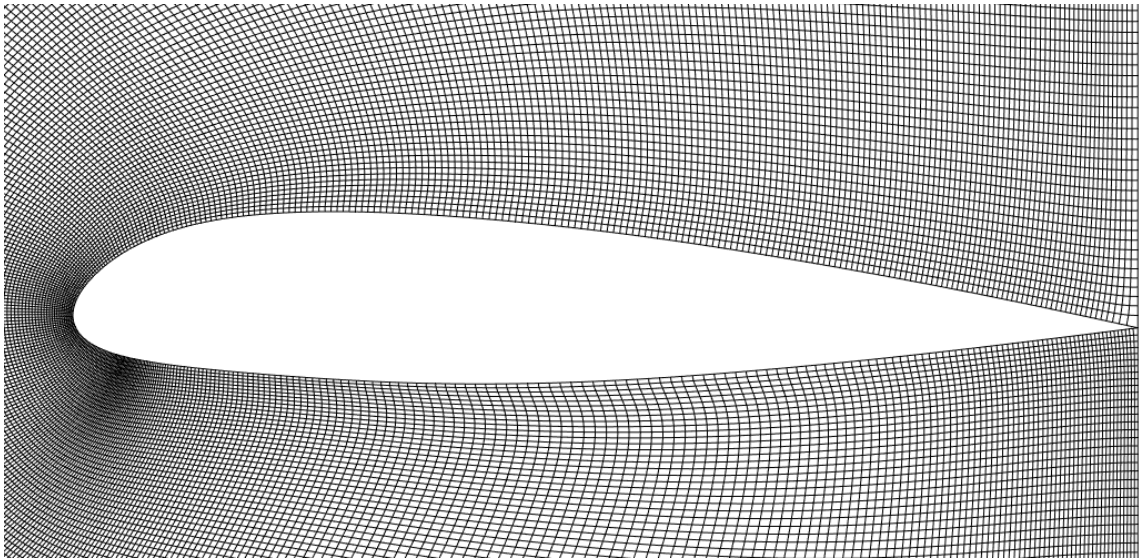
Generation of metric-conforming structured grids





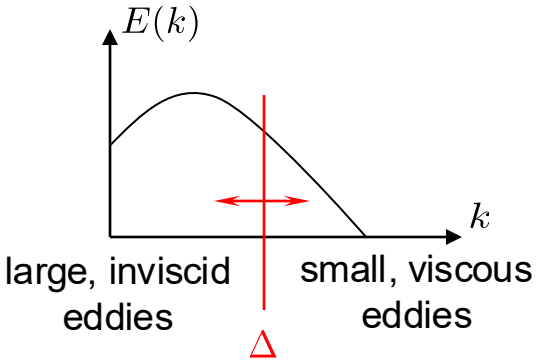




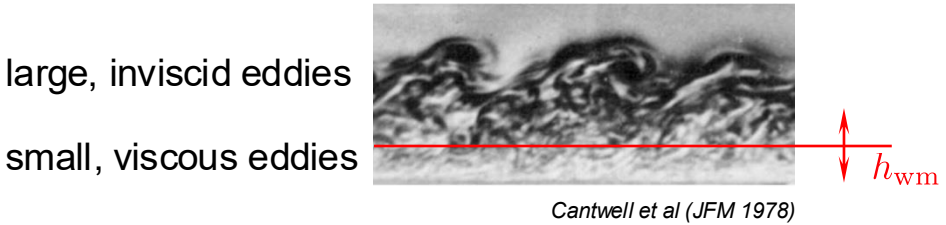


Finding the modeling interface in wall-modeled LES

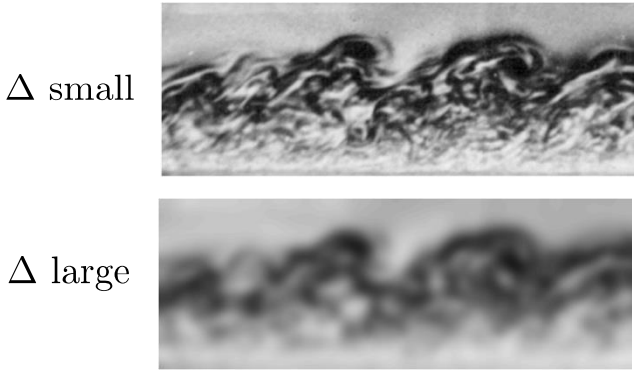
Subgrid modeling

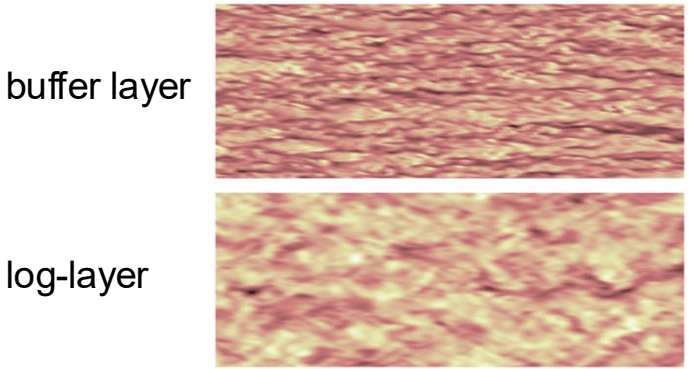


Wall-modeling



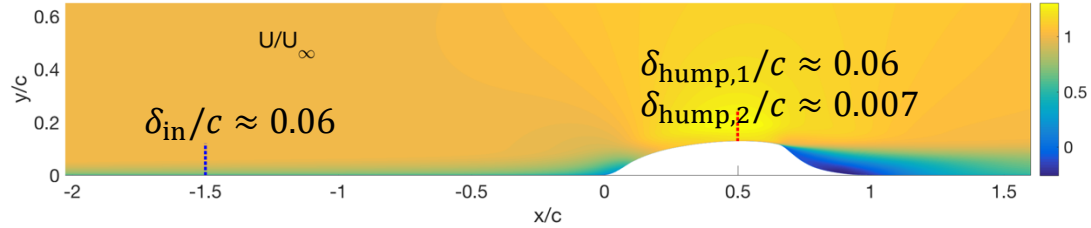
Cantwell et al (JFM 1978)



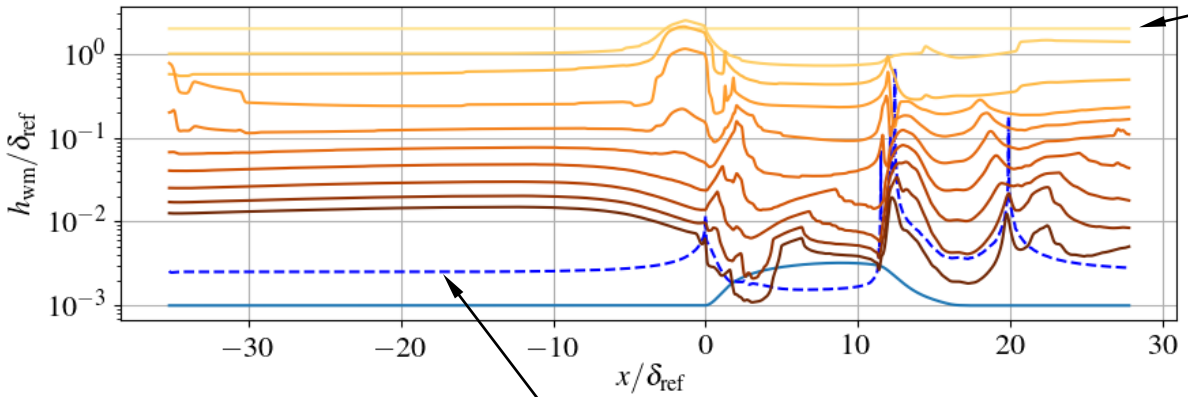


Finding the modeling interface in wall-modeled LES[1]

A priori test for the NASA wall-mounted hump[2]



Wall-model interface location h_{wm}



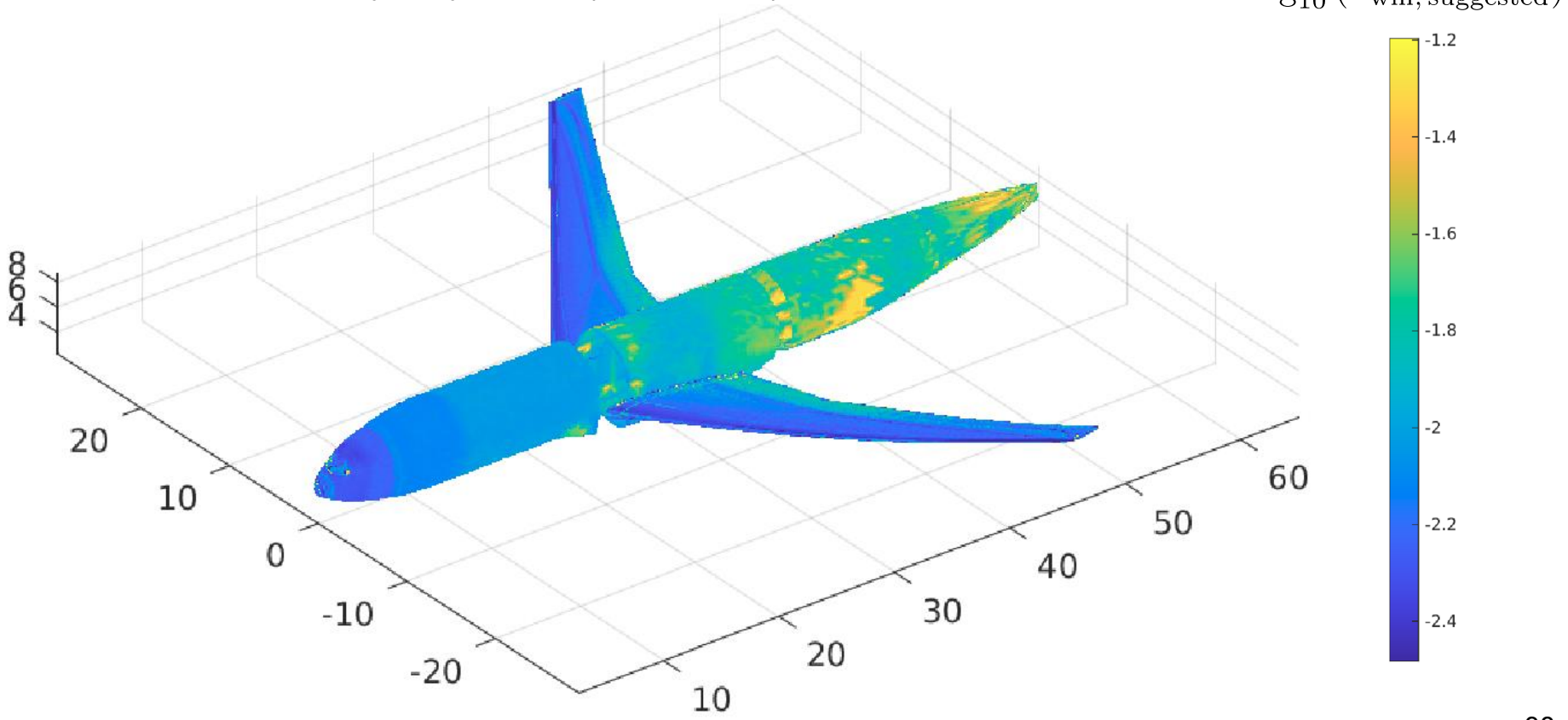
initial h_{wm} well outside the boundary layer

sequence of adapted h_{wm} profiles, for increasing desired computational cost

[1] *Kahraman & Larsson (AIAA J 2022)*
[2] *Uzun & Malik (AIAA J 2018)*

Finding the modeling interface in wall-modeled LES

NASA High-Speed Common Research Model
Mean flow field provided by Aditya Ghate (NASA Ames)



Concluding thoughts

- Happy with our ability to identify where/how to modify the mesh (=estimate directional residuals) – entirely a post-processing operation, has been implemented in multiple different CFD codes

- Required future work:
  - How to actually create the adapted meshes, ensuring smoothness, accuracy, geometrical flexibility, etc?
  - How to combine grid- and h_{wm}-adaptation for WMLES?
  - Need to create a more automated workflow to make this useful in practical engineering
  - Need to try this for more complex flows, including with shocks, multi-phase, etc

- Long-term vision: CFD should be like Matlab's ode45 function

- Common question: "How expensive is it to run on a long sequence of grids?" Actually, it's less than you'd think: if the cost increases by a factor of 4 in each step, the total cost is $1 + (1/4) + (1/4)^2 + (1/4)^3 + \cdots = 4/3$ times the cost of the finest grid alone