The hybridized DG methods for WS1, WS2, and CS2 test cases

P. Fernandez, N.C. Nguyen and J. Peraire

Aerospace Computational Design Laboratory
Department of Aeronautics and Astronautics, MIT

5th High-Order Workshop

Sunday, January 7th, 2018

WS1: Taylor-Green Vortex

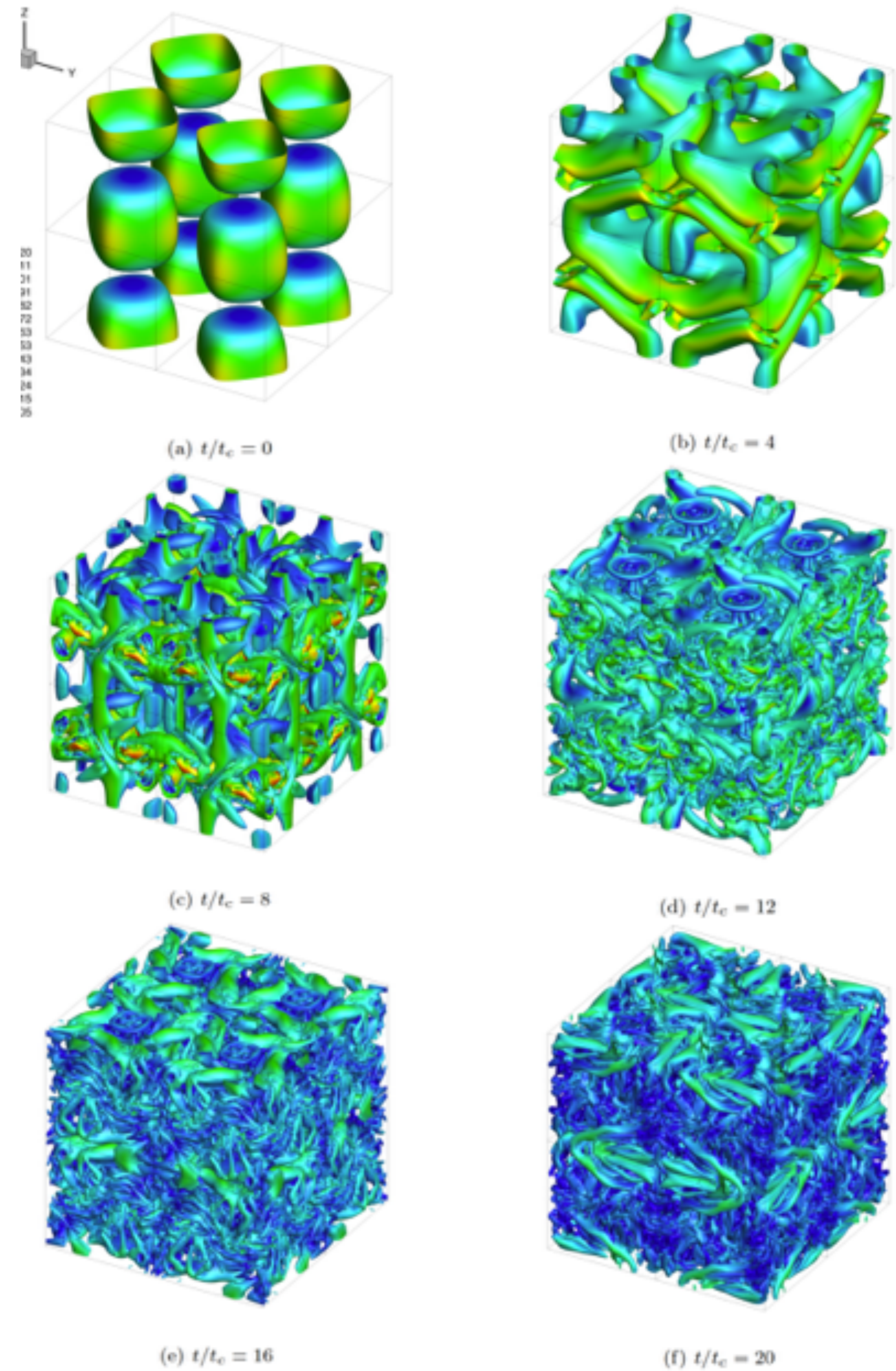
Case description

- **Flow conditions:**

  $$Re = U_0 \, L / \nu = 1{,}600 \text{ and } \infty$$

  $$M = U_0 / a_0 = 0.1$$

- **Solver:** DIGASO
- **Discretization scheme:**
  - Space: 3rd-order IEDG
  - Time: 3rd-order DIRK(3,3)
- **Quadrature rule:** Gauss-Legendre with exact integration of polynomials up to 5th-order
- **Number of elements:** 64 x 64 x 64 hexes
- **Baseline CFL number:** 0.375
- **Studies presented:**
  - Riemann solver
  - SGS model: ILES, Smagorinsky, WALE, Vreman
  - Time-step size



(a) $t/t_c = 0$ (b) $t/t_c = 4$ (c) $t/t_c = 8$ (d) $t/t_c = 12$ (e) $t/t_c = 16$ (f) $t/t_c = 20$

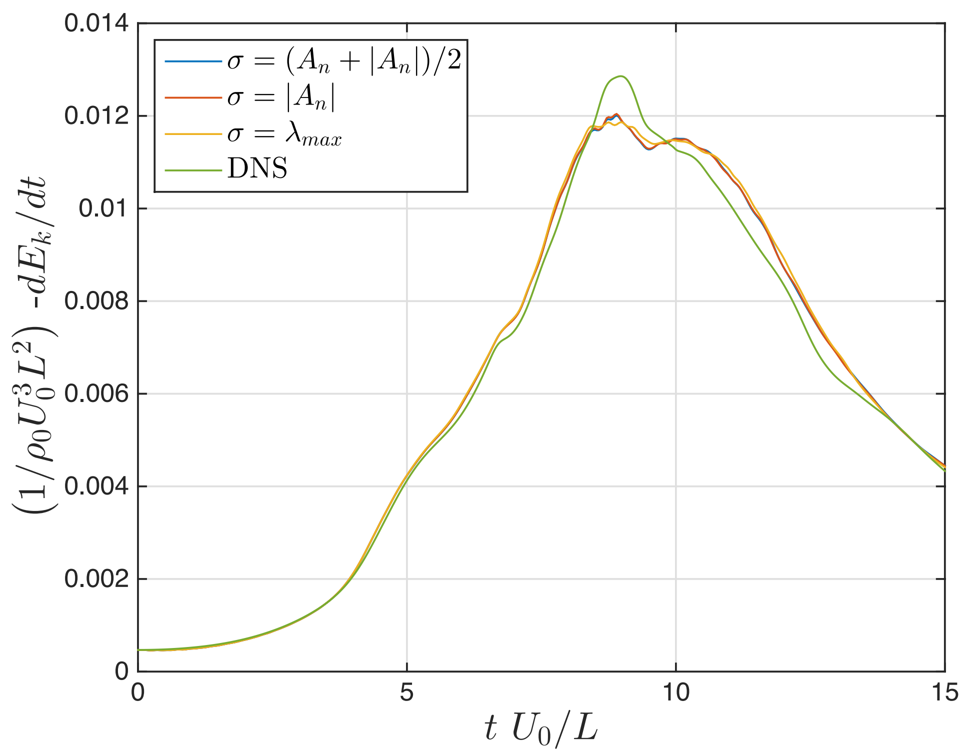
Source: G. Giangaspero et al., HiOCFD3, 2015

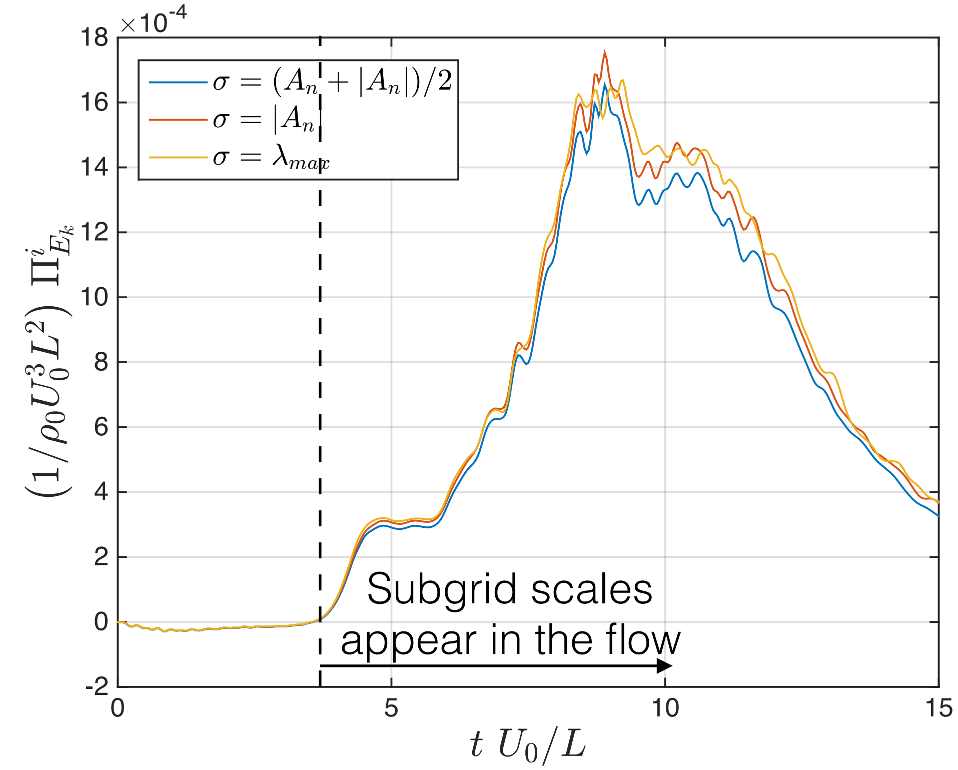
Effect of Riemann solver

Time evolution of kinetic energy dissipation rate

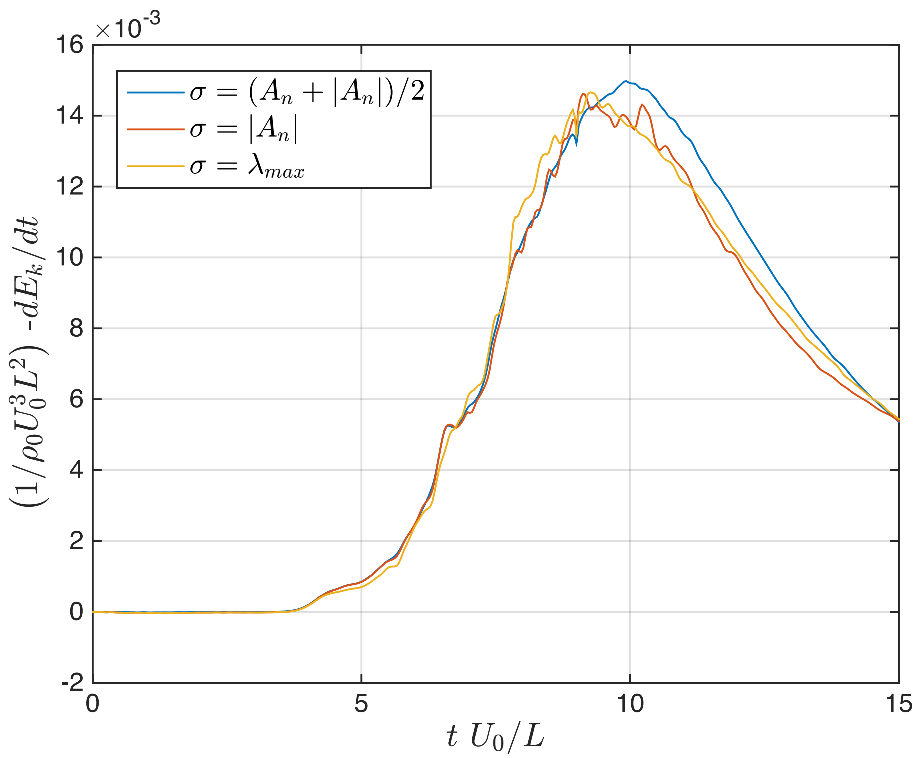
Numerical dissipation of kinetic energy

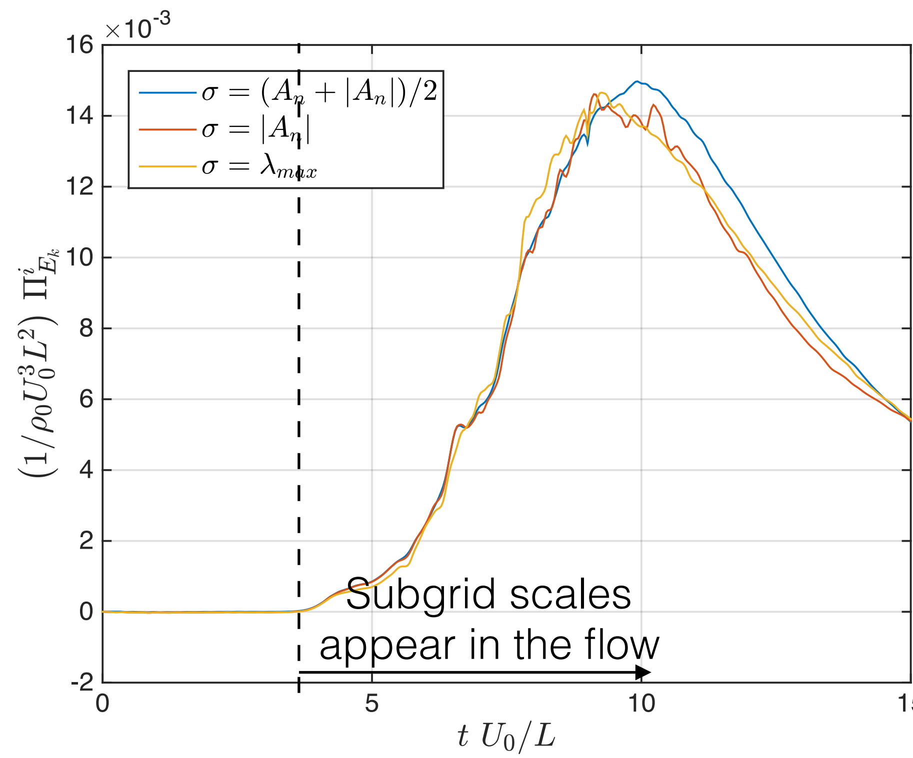
Re = 1600





Re = ∞





Laminar regime ⟷ Turbulent regime

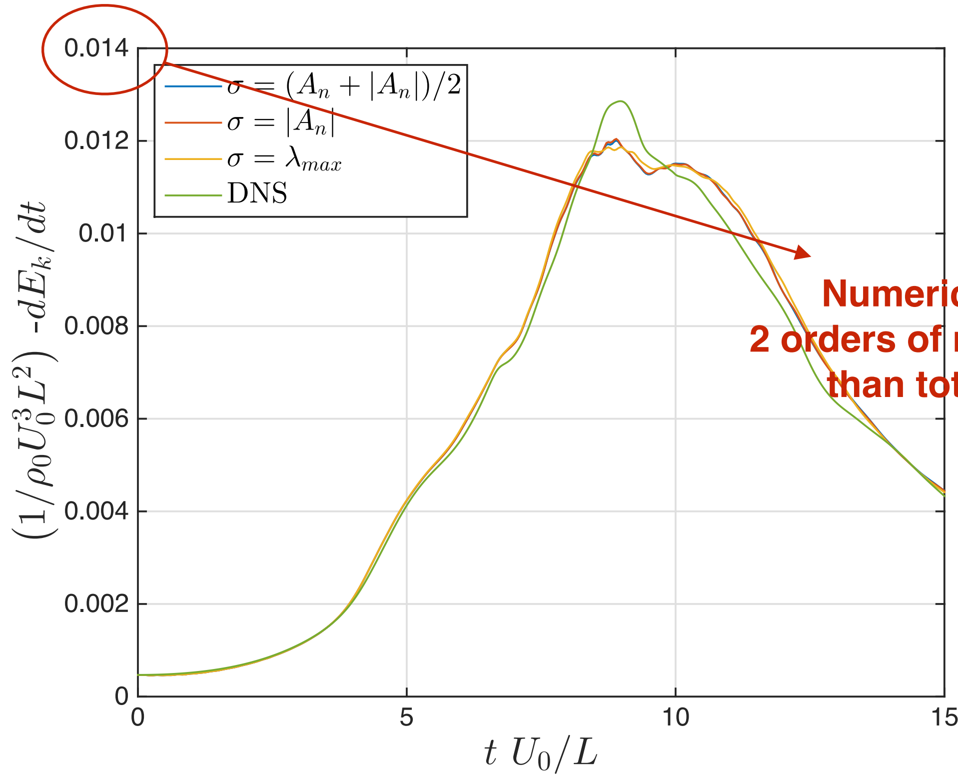
Laminar regime ⟷ Turbulent regime

Effect of Riemann solver

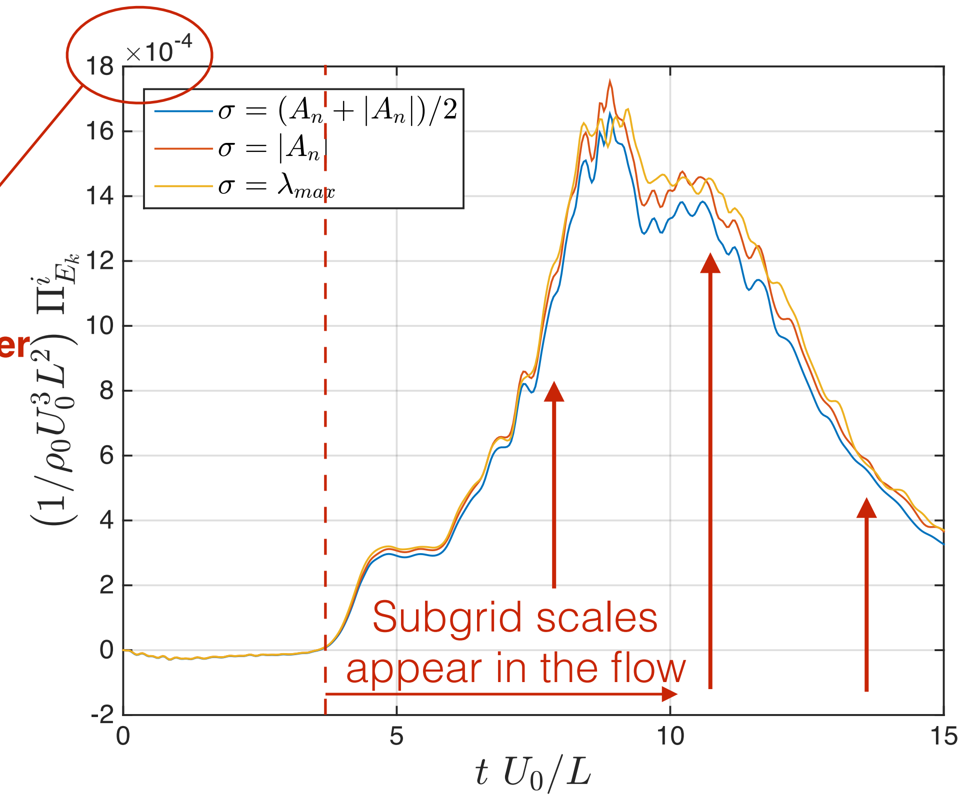
Time evolution of kinetic energy dissipation rate

Numerical dissipation of kinetic energy

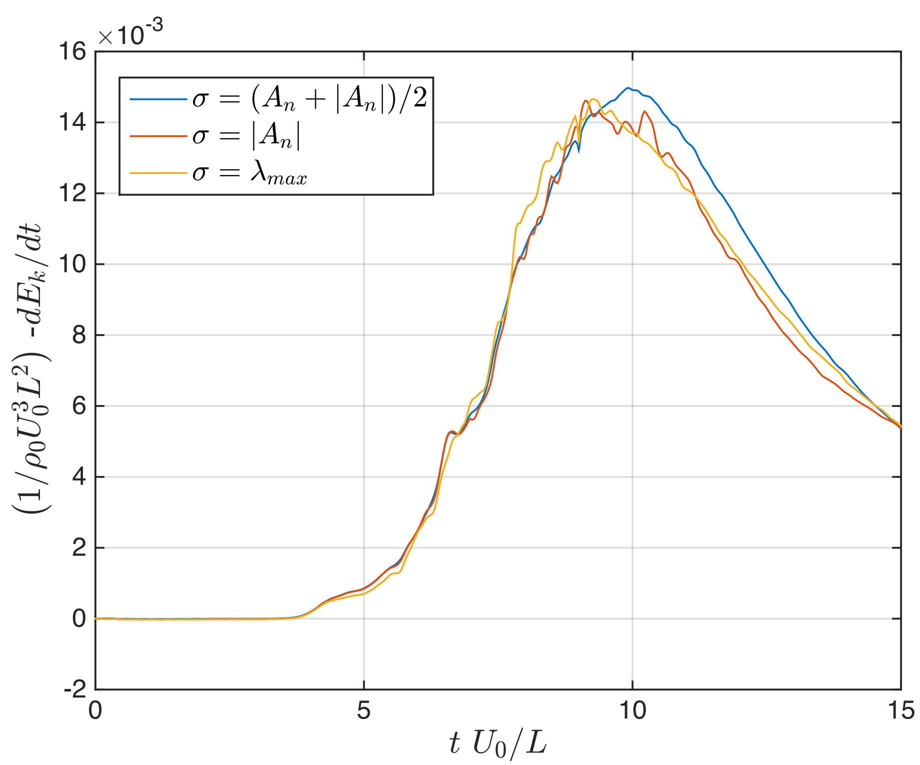
Re = 1600



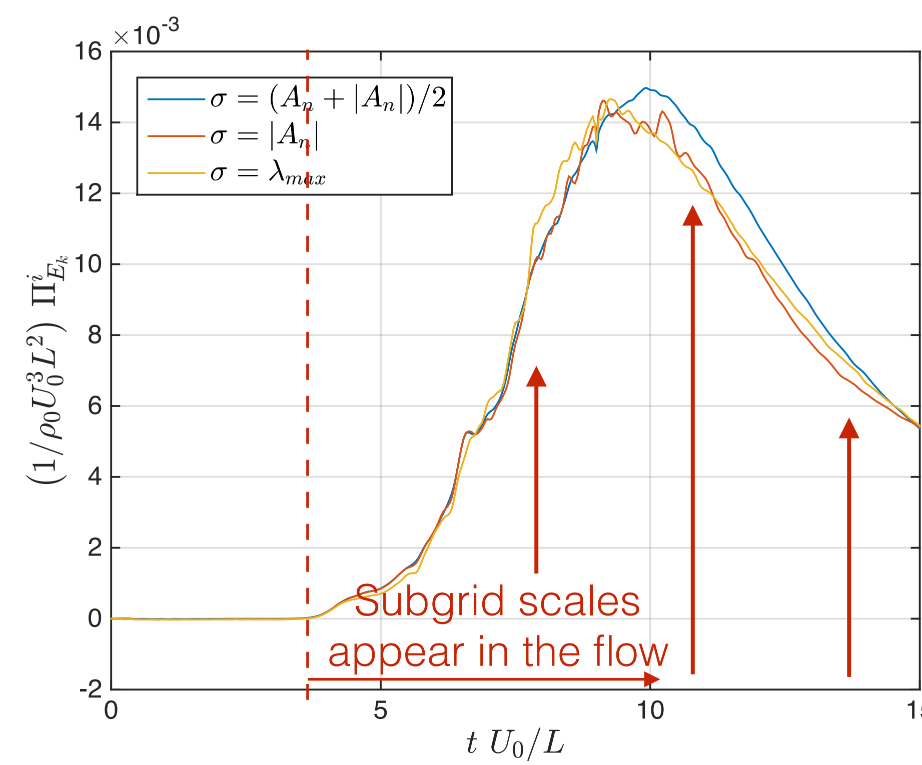
Numerical dissipation
2 orders of magnitude smaller than total dissipation



Re = ∞



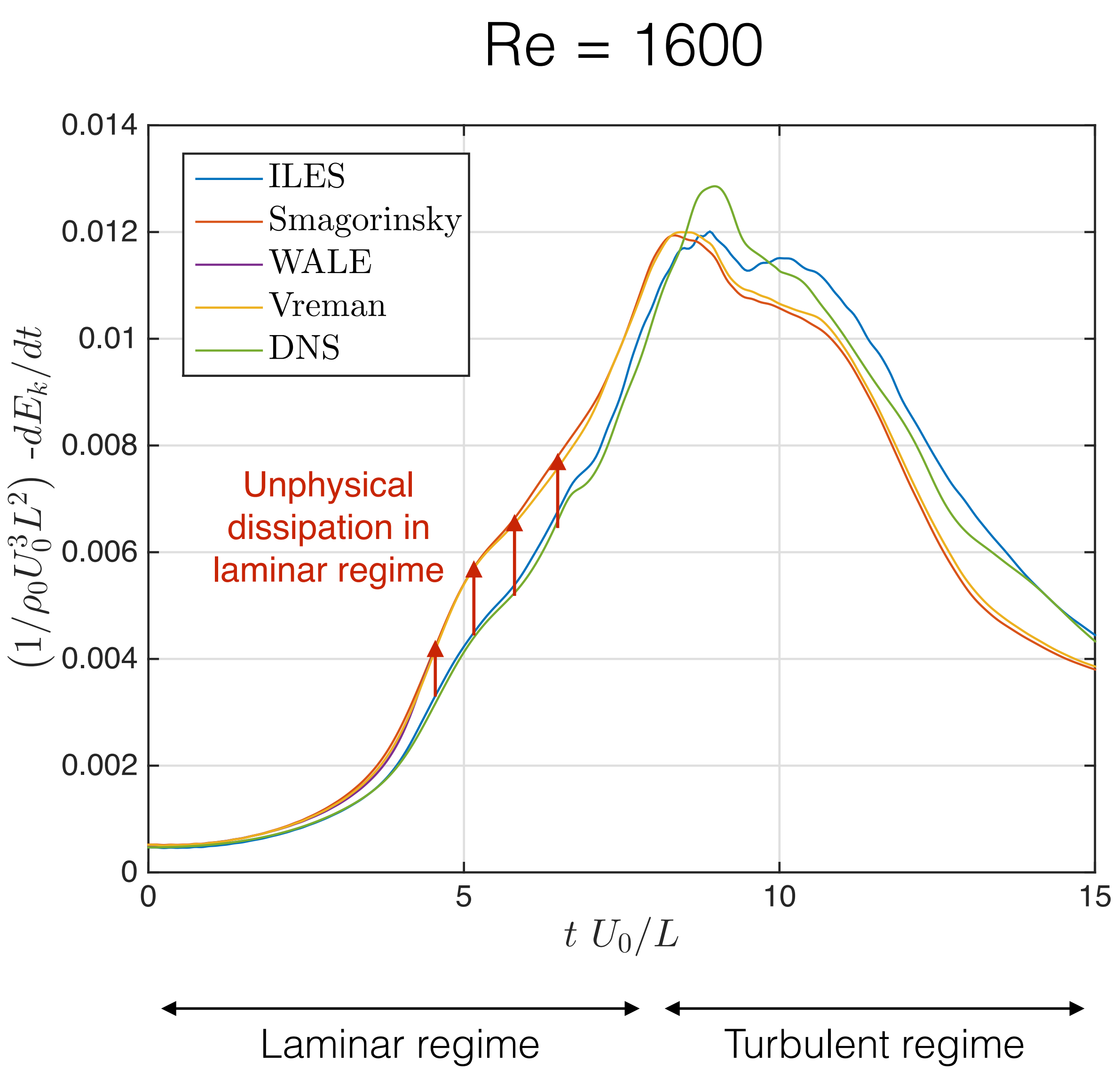
Laminar regime Turbulent regime

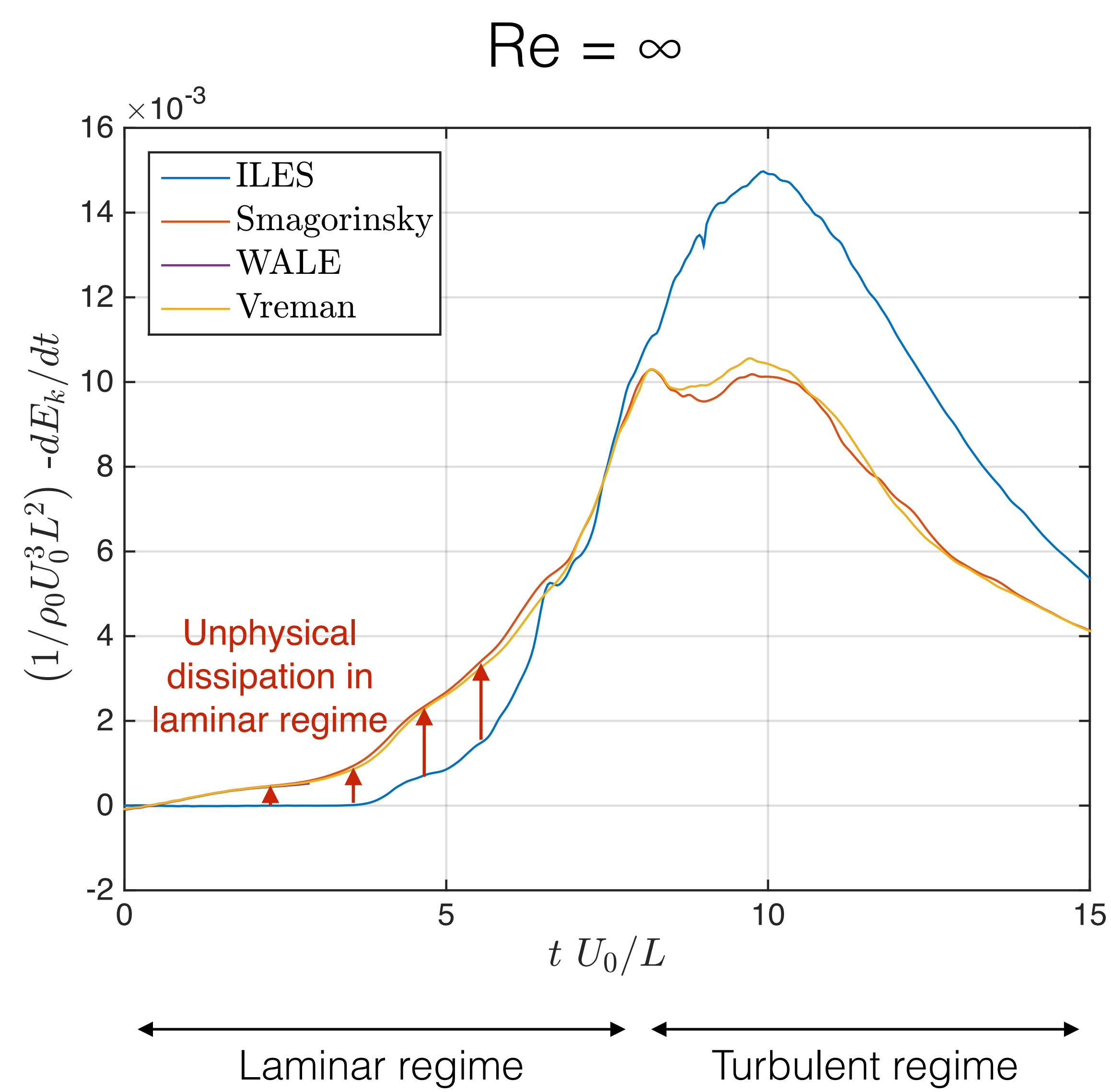


Laminar regime Turbulent regime

MIT AEROASTRO

Effect of SGS model



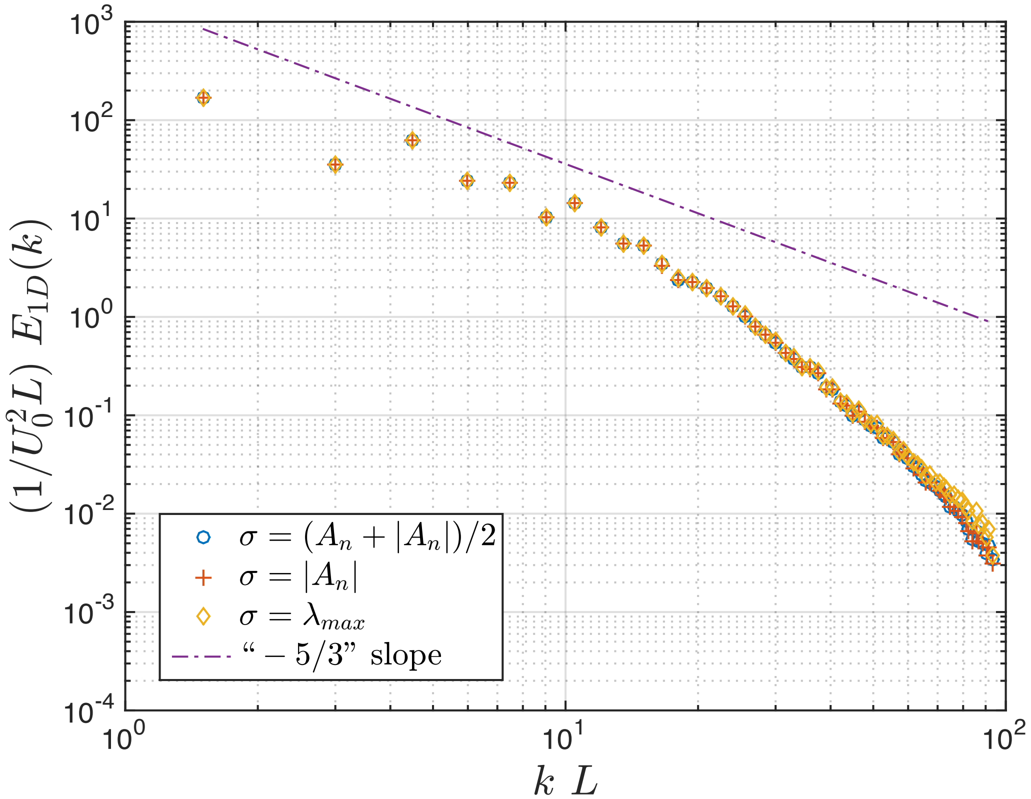


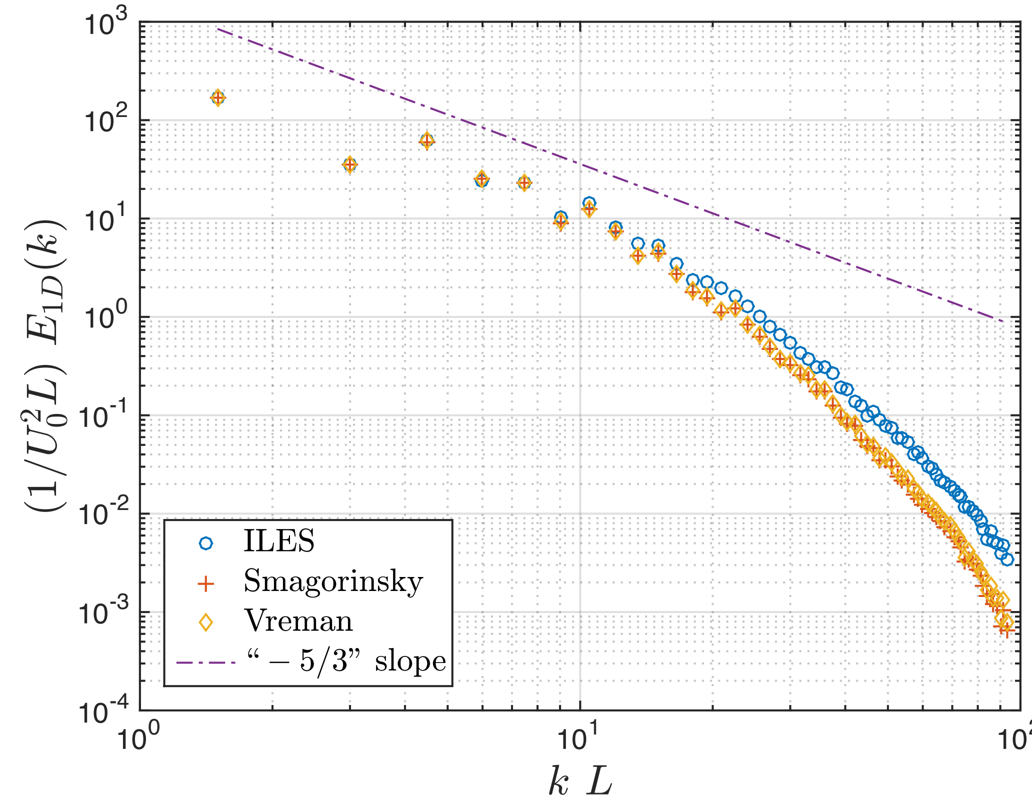
- All **SGS models produce unphysical dissipation** in the laminar regime
- Built-in ILES capability in DG more accurate than explicit SGS models to detect SGS scales and add numerical dissipation only under those conditions

Kinetic energy spectra at t = 9 L / U_0

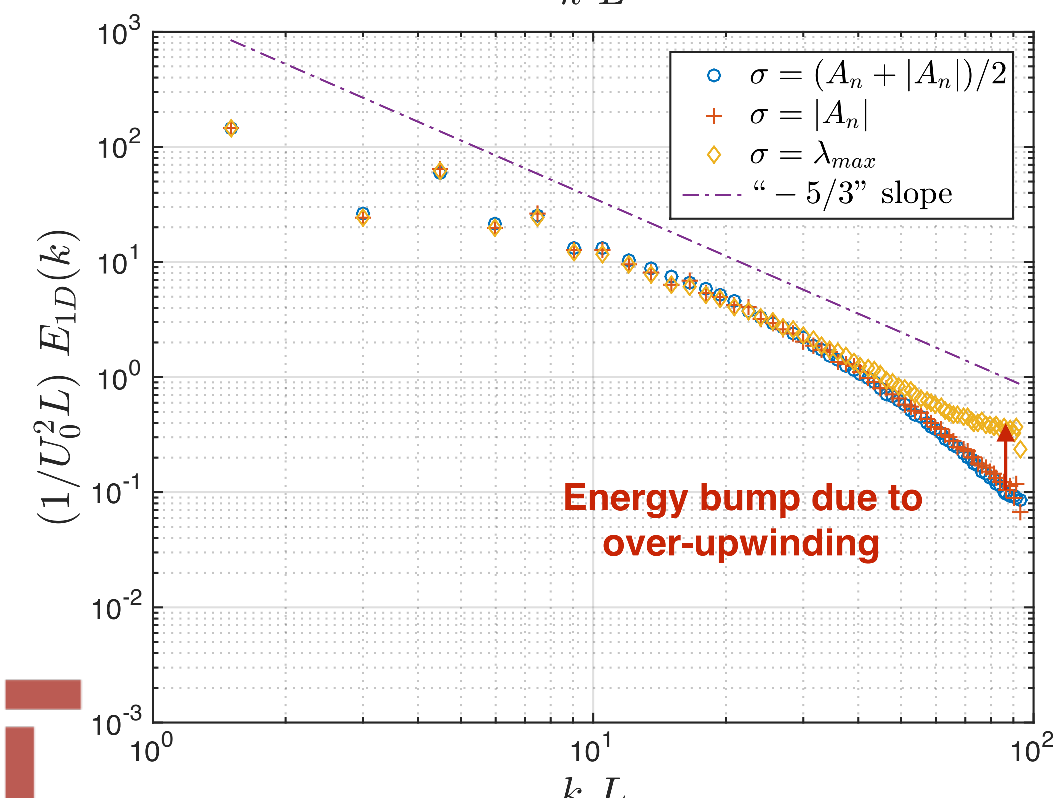
ILES with different Riemann solvers | Explicit vs. implicit LES

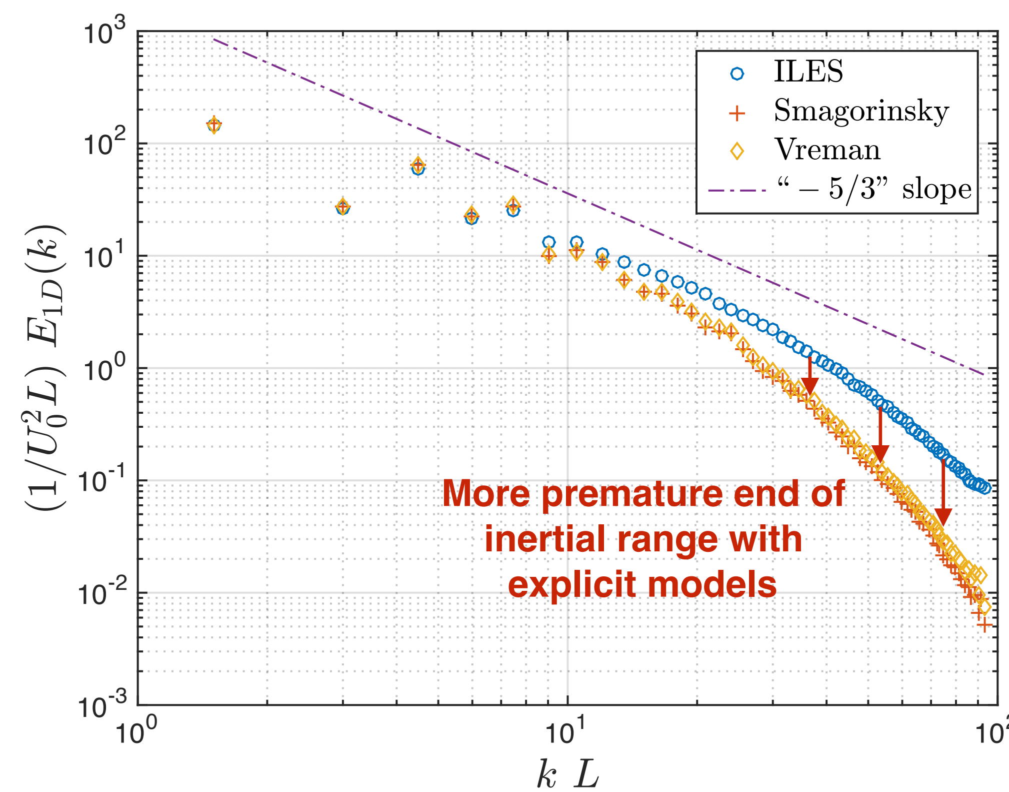
Re = 1600





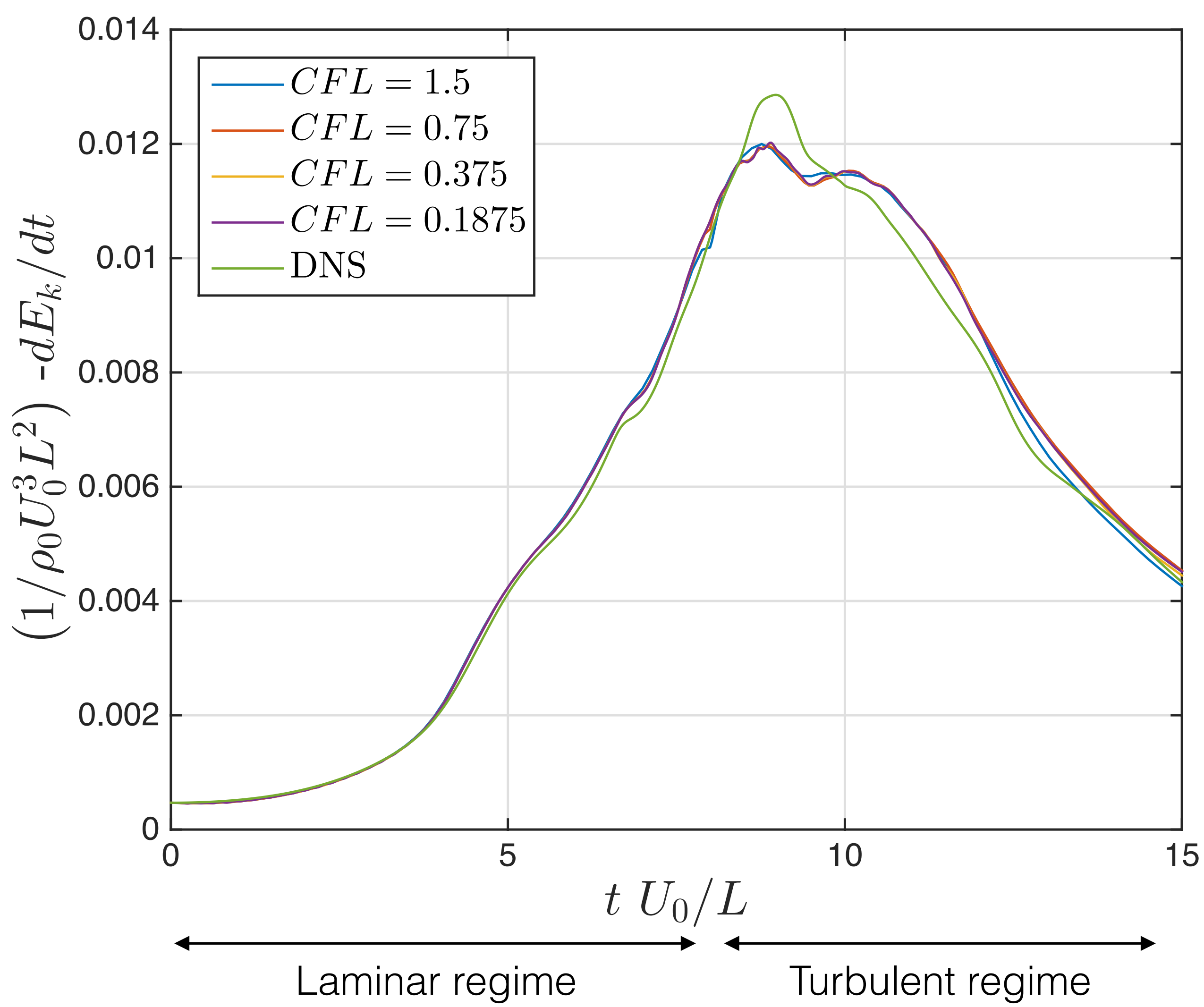
Re = ∞



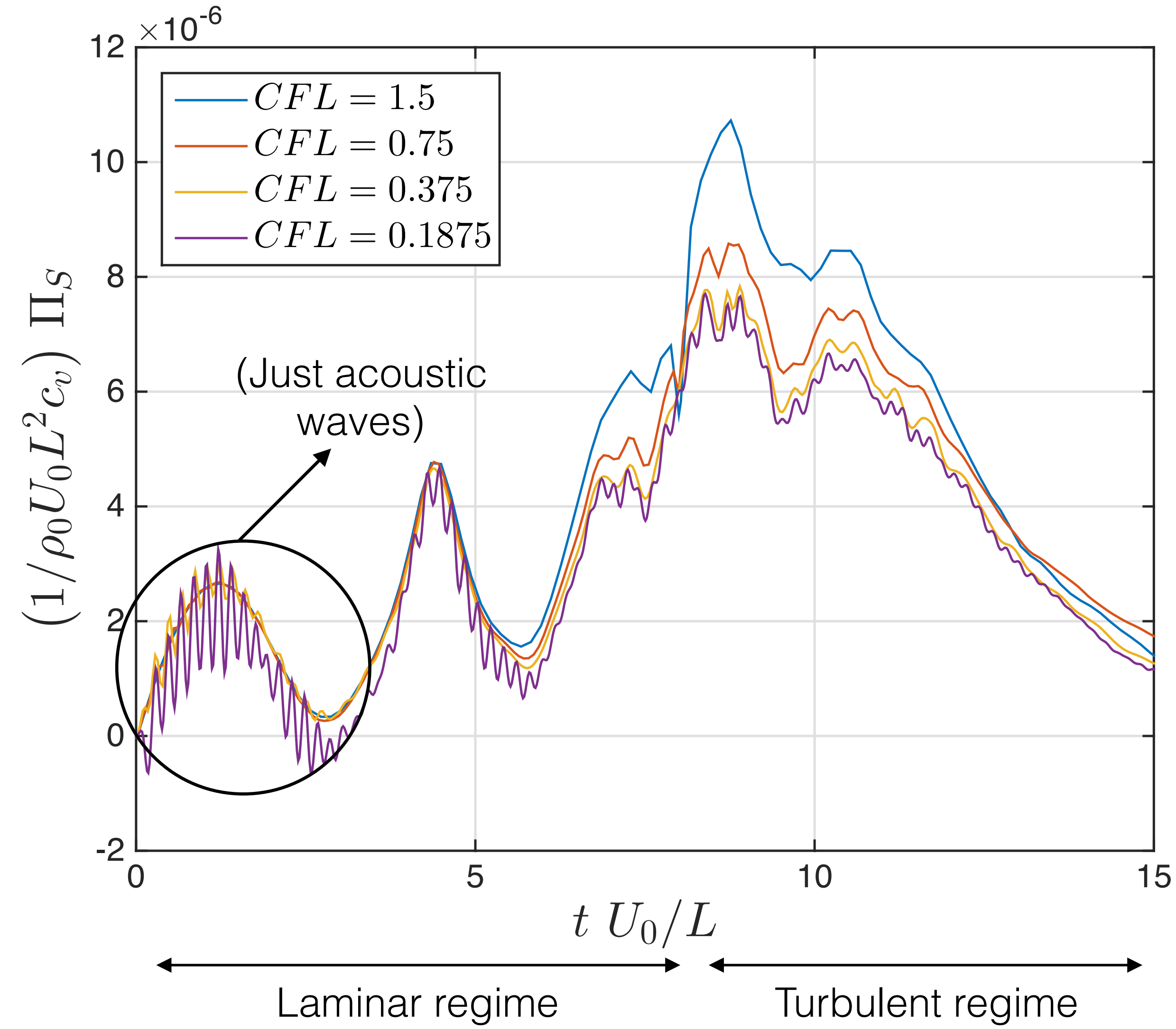


Effect of time-step size

Time evolution of kinetic energy dissipation rate



Numerical generation of entropy



- Vorticity and entropy modes captured with local CFL ~ 1. Acoustic modes require CFL << 1 (recall $M_0 = 0.1$)
- **Optimal LES implementations** likely with local **CFL > 1**: Global CFL >>> 1 for wall-bounded flows

WS2: Turbulent Channel Flow

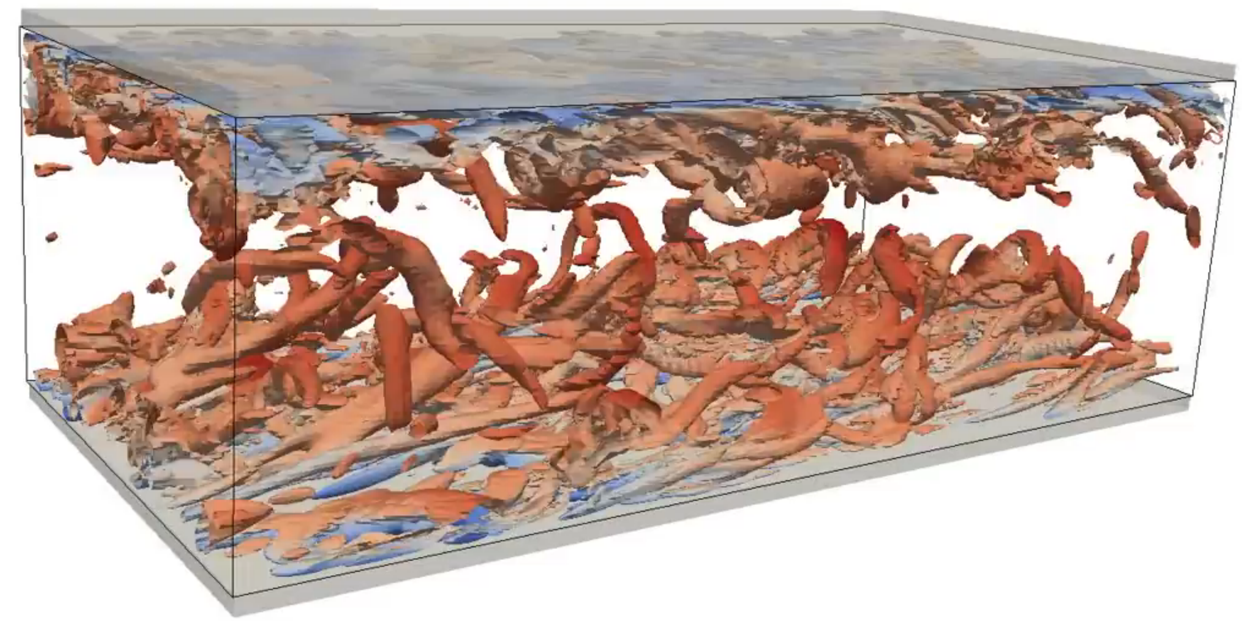
Case description

- **Flow conditions:**

  Re_τ =182 and 544 | M = 0.1

- **Channel size:** $4\pi\delta$ x 2δ x $2\pi\delta$
- **Solver:** DIGASO
- **Discretization scheme:**
  - Space: 3rd- and 5th- order IEDG
  - Time: 3rd-order DIRK(3,3)
- **Quadrature rule:** Gauss-Legendre with exact integration of polynomials up to 5th- and 9th-order, respectively.
- **Number of elements:** 48 x 32 x 40
- **Studies presented:**
  - SGS model

  Riemann solver study is presented in ***(Fernandez et al., 2017b)***

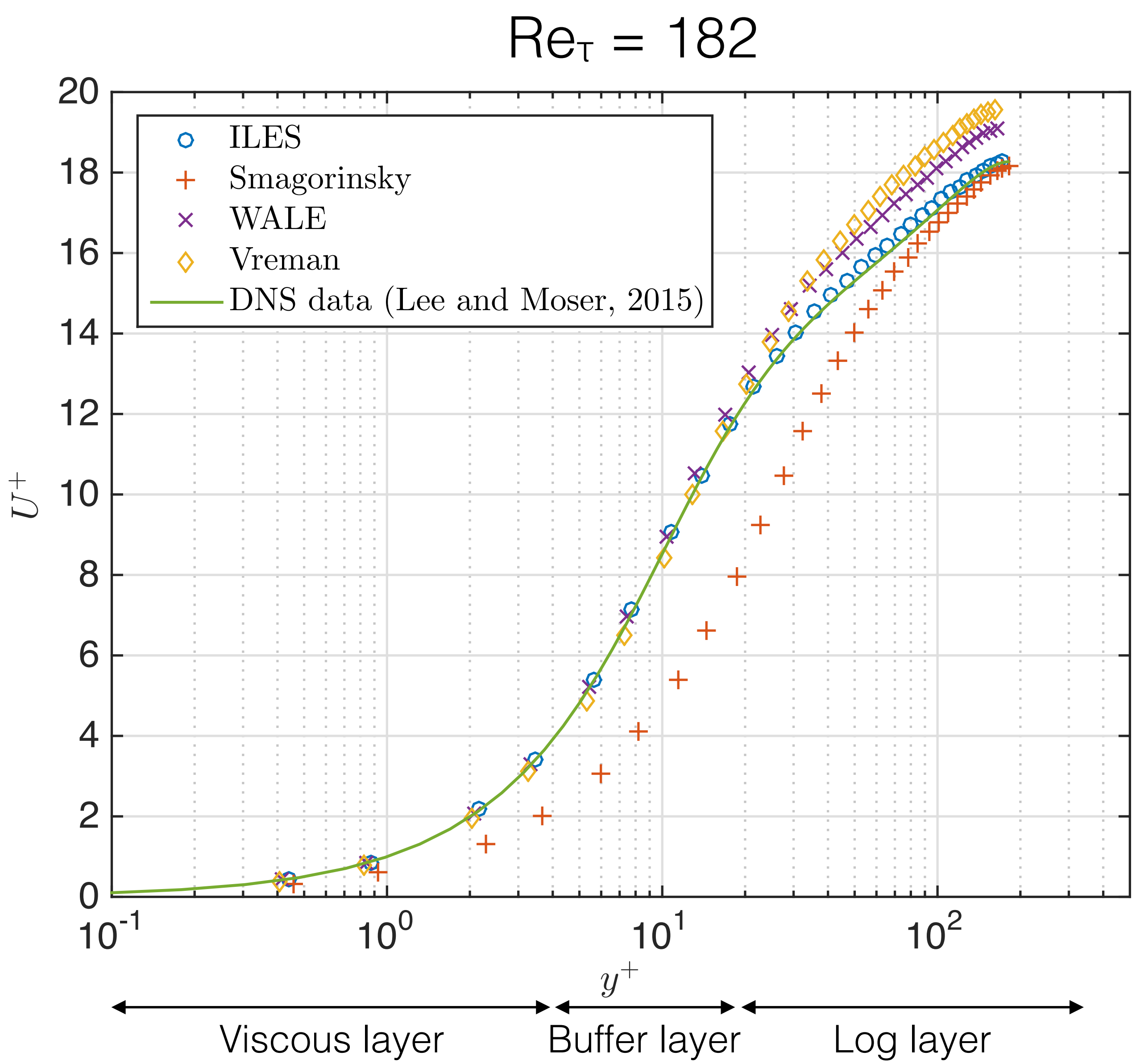


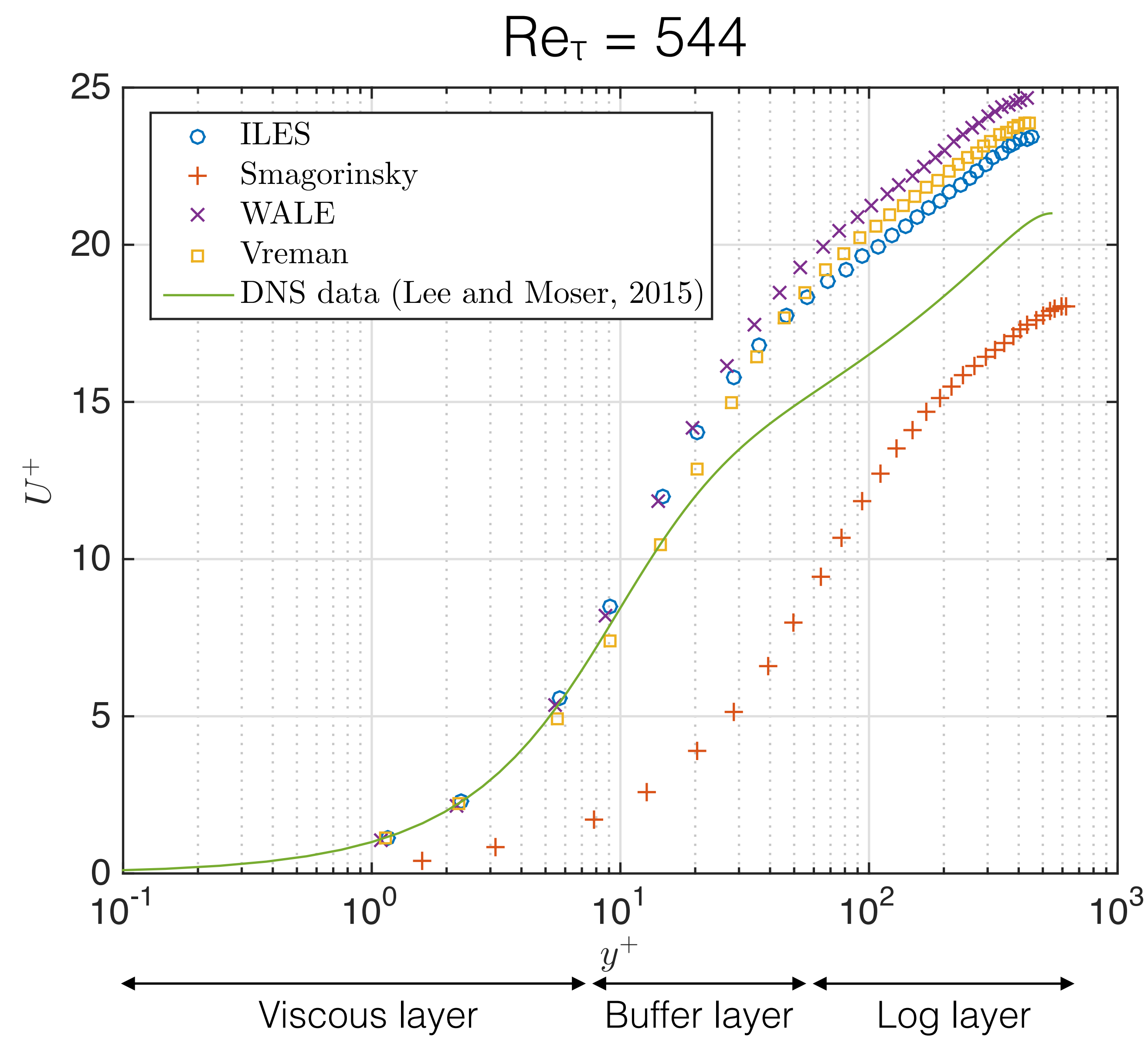
Source: P. Blonigan et al., APS DFD Meeting 2016

Mesh resolution in wall units

	Re_τ=182	Re_τ=544
Δx^+	23.8	71.2
Δy^+_{avg}	5.69	17.0
Δy^+_{wall}	0.438	1.310
Δz^+	14.3	42.7
Δt^+	0.290	0.740

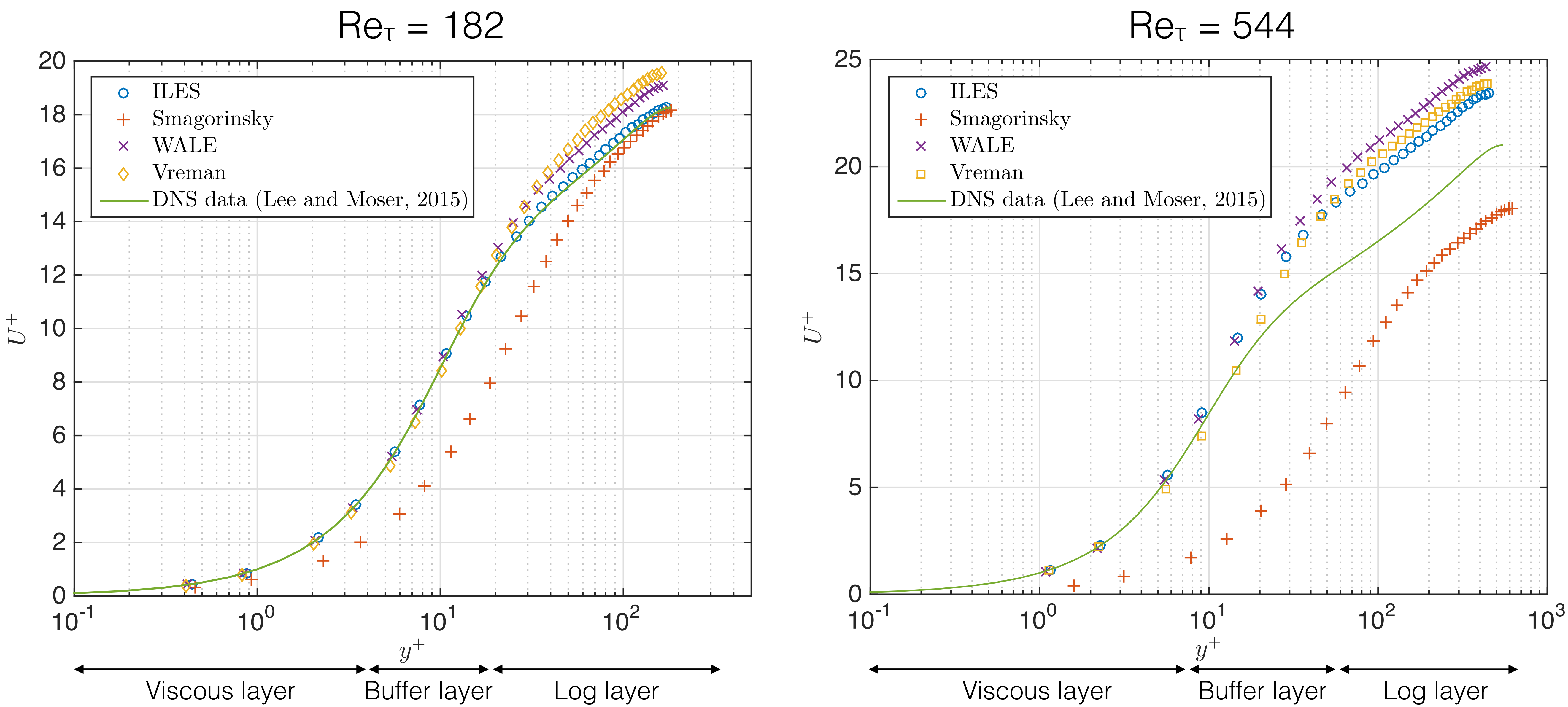
Effect of SGS model





- **ILES more accurate** than explicit LES at both Reynolds numbers
- Viscous sublayer: Accurately resolved in all cases (except Smagorinsky-LES)
- Log layer: Accurately resolved in all cases

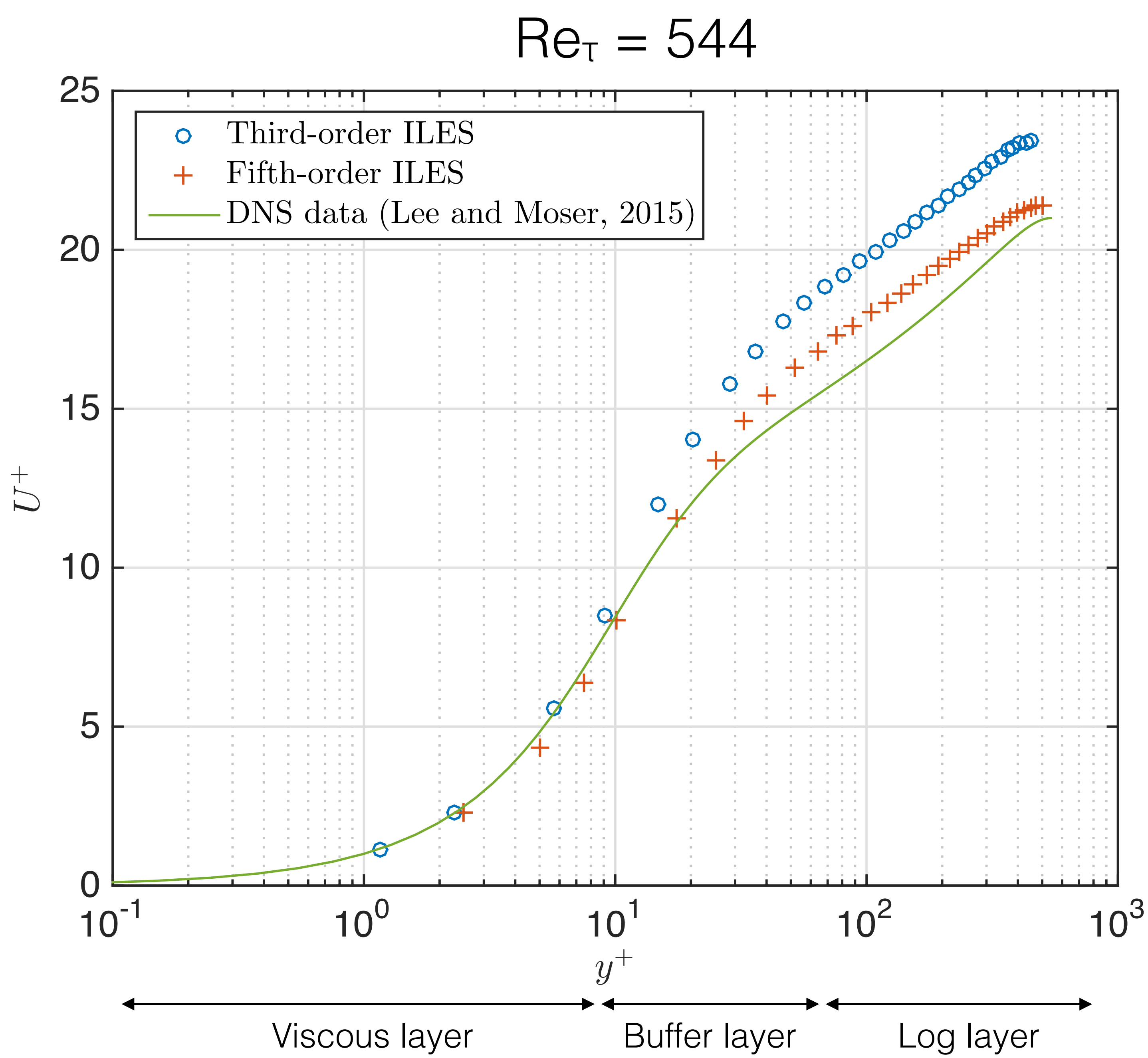
Effect of SGS model



- **Buffer layer: Not accurately resolved**. All simulations introduce too much "numerical transport of momentum" (not the same as "numerical dissipation"!)
- Ratio "momentum transport-to-dissipation" due to discretization errors and explicit SGS models is larger than the true SGS value

Effect of accuracy order

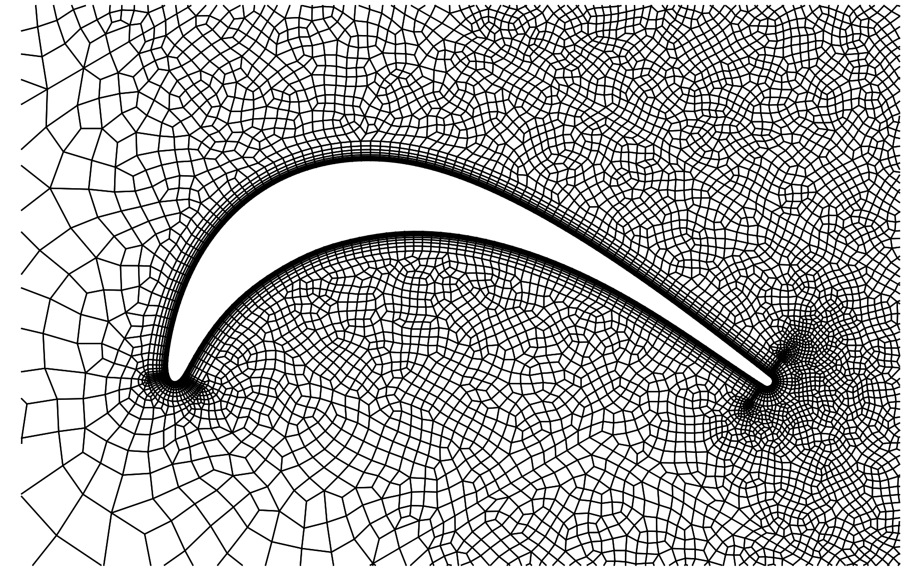
- 3rd- and 5th-order accurate LES have same number of DOFs and time-step size
- LES predictions improve by increasing accuracy order
- Accurately resolving the large turbulent scales by using a high-order method plays a more important role than that of the subgrid scales



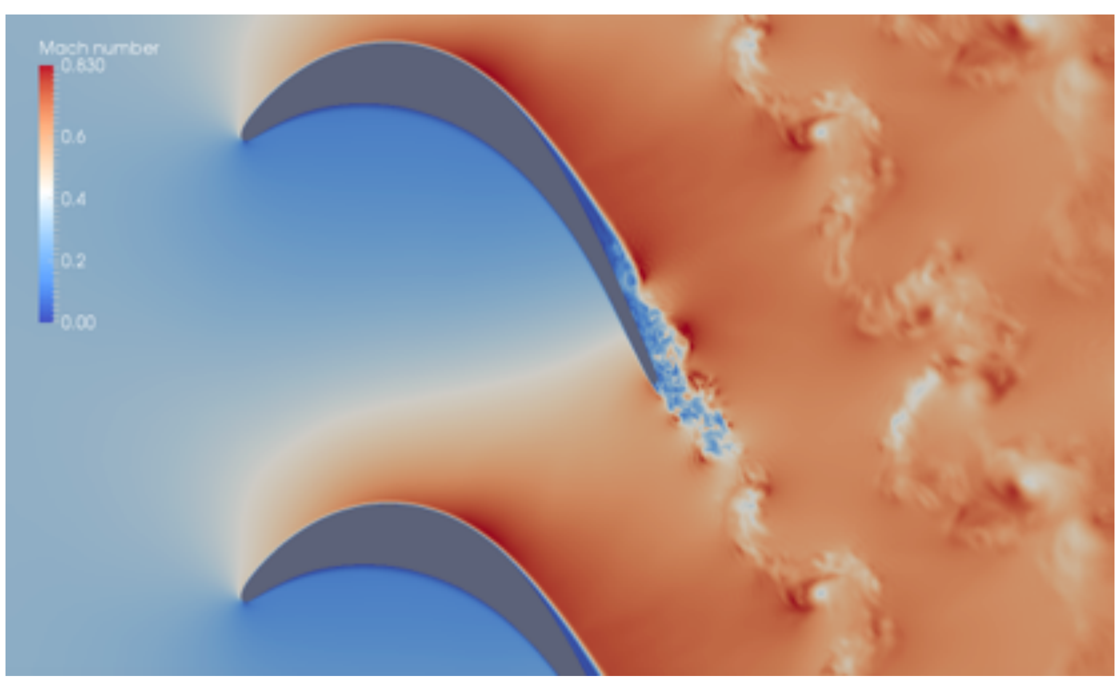
CS2: T106C LTP cascade

Case description

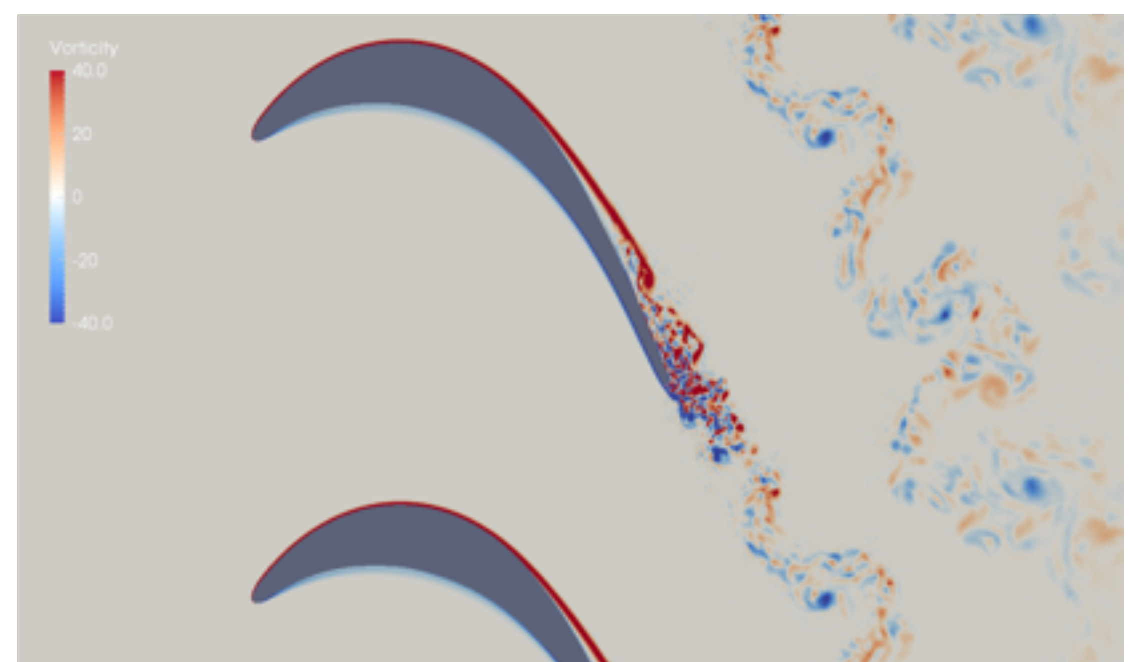
- **Solver:** DIGASO
- **Discretization scheme:**
  - Space: 3rd-order IEDG
  - Time: 3rd-order DIRK(3,3)
- **Number of elements:** 118,680 isoparametric hexes (baseline mesh)
- **Time-step size:** $\Delta t = 5 \cdot 10^{-3}\ c / v_\infty$ ($CFL_{global} = 6.6$)
- **LES model:** No model (ILES)
- **Riemann solver:** Lax-Friedrichs-type in ***(Fernandez et al., 2017b)***
- **Quadrature rule:** Gauss-Legendre with exact integration of polynomials up to 5th-order



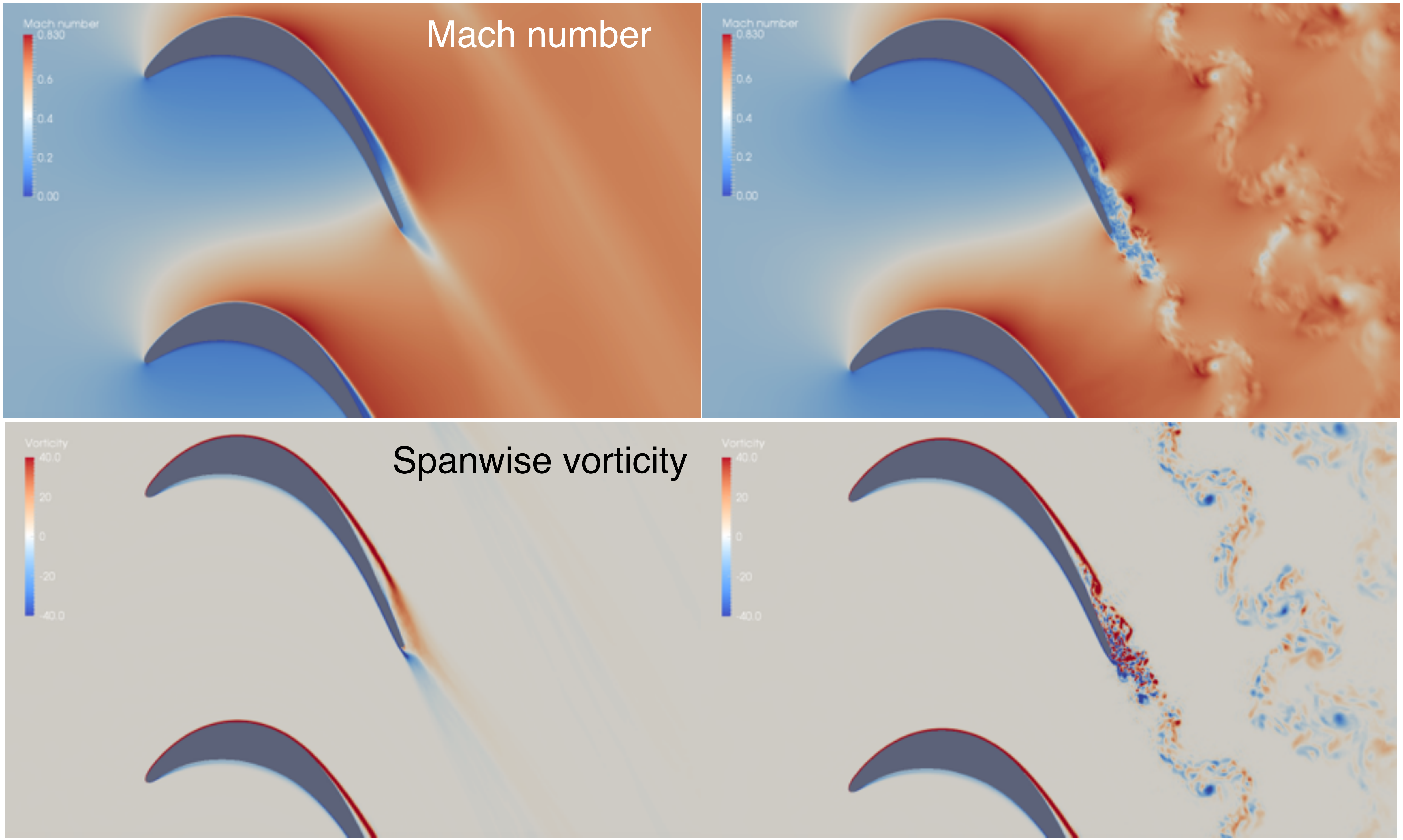
Mach number



Spanwise vorticity

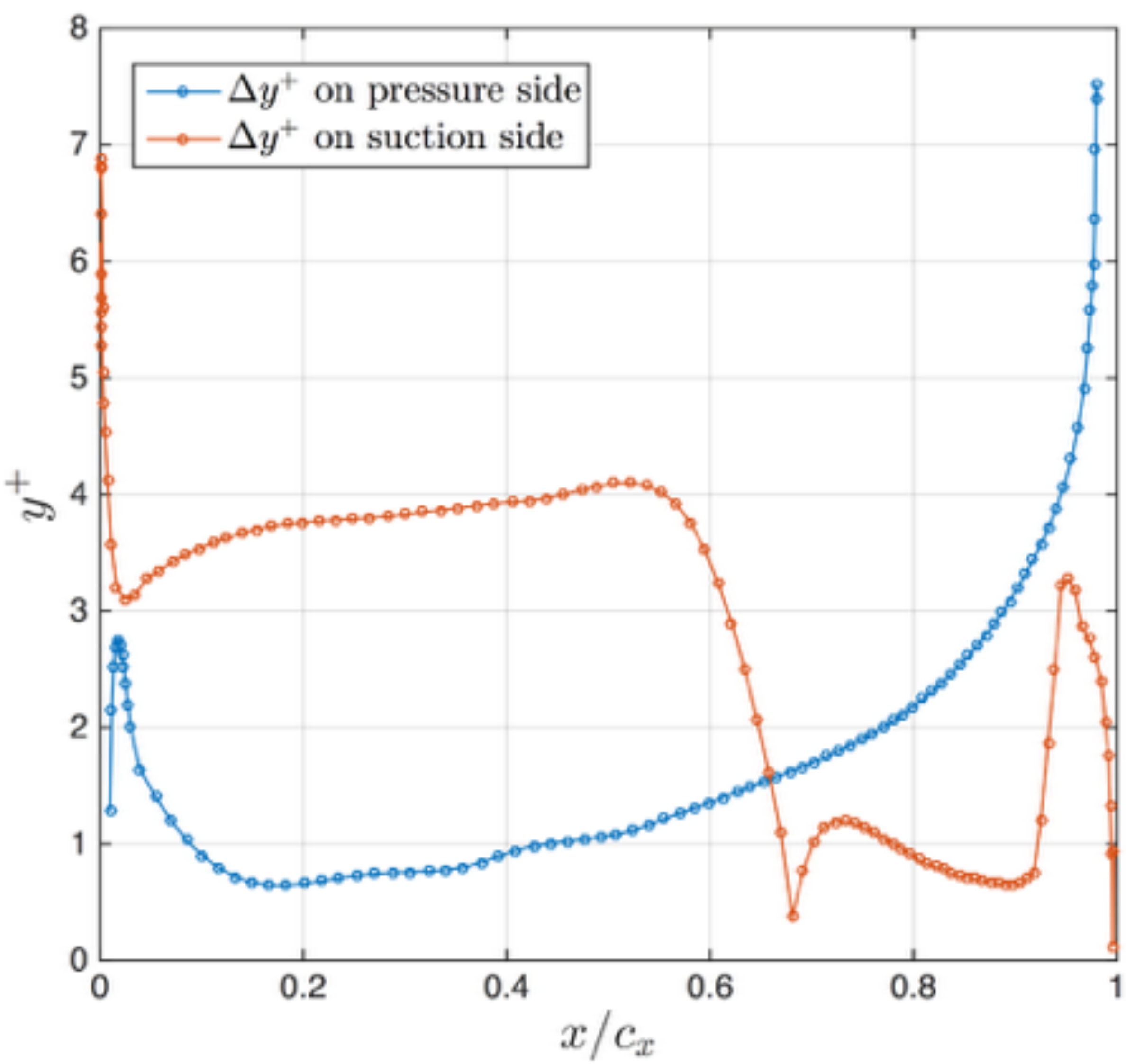


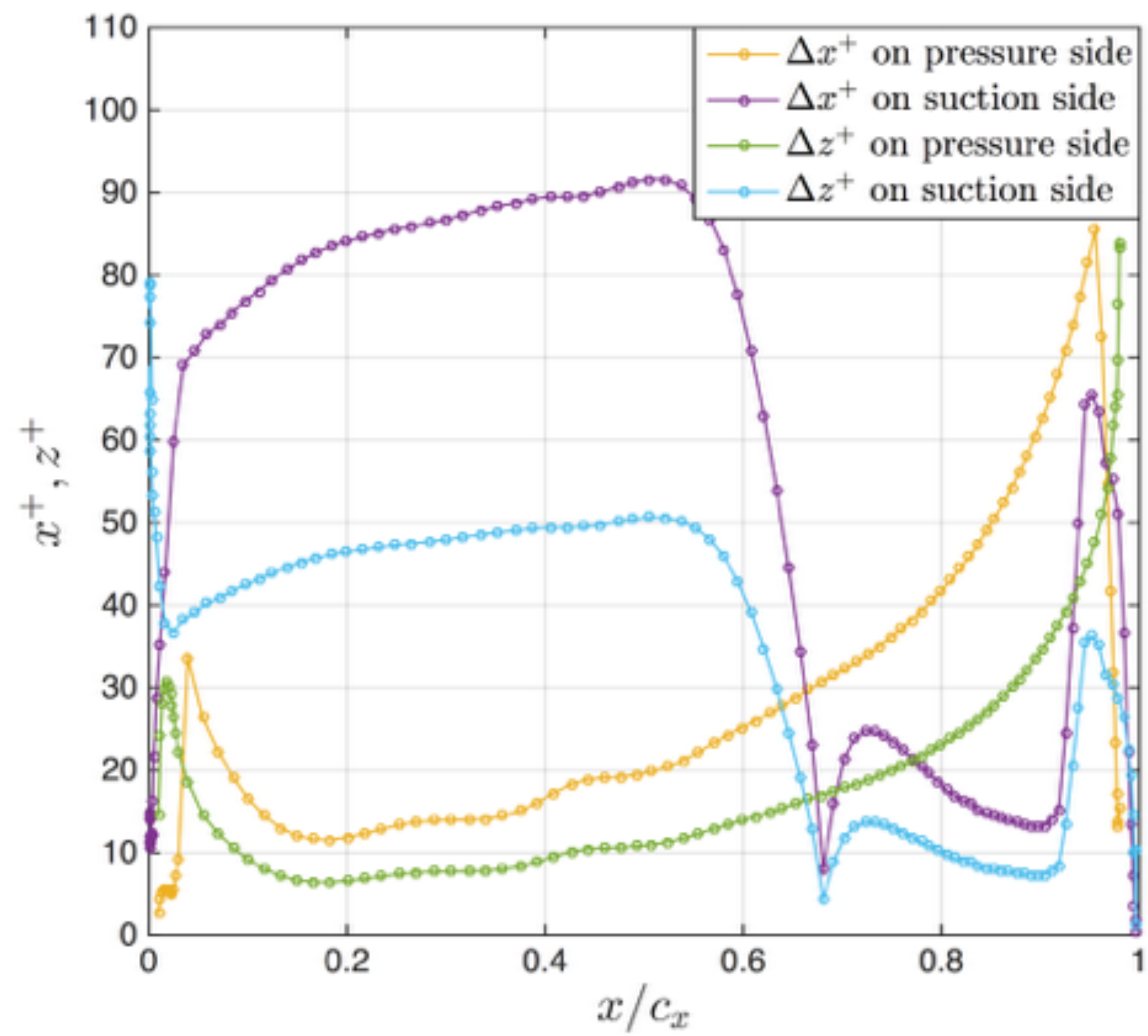
Flow fields on the periodic plane



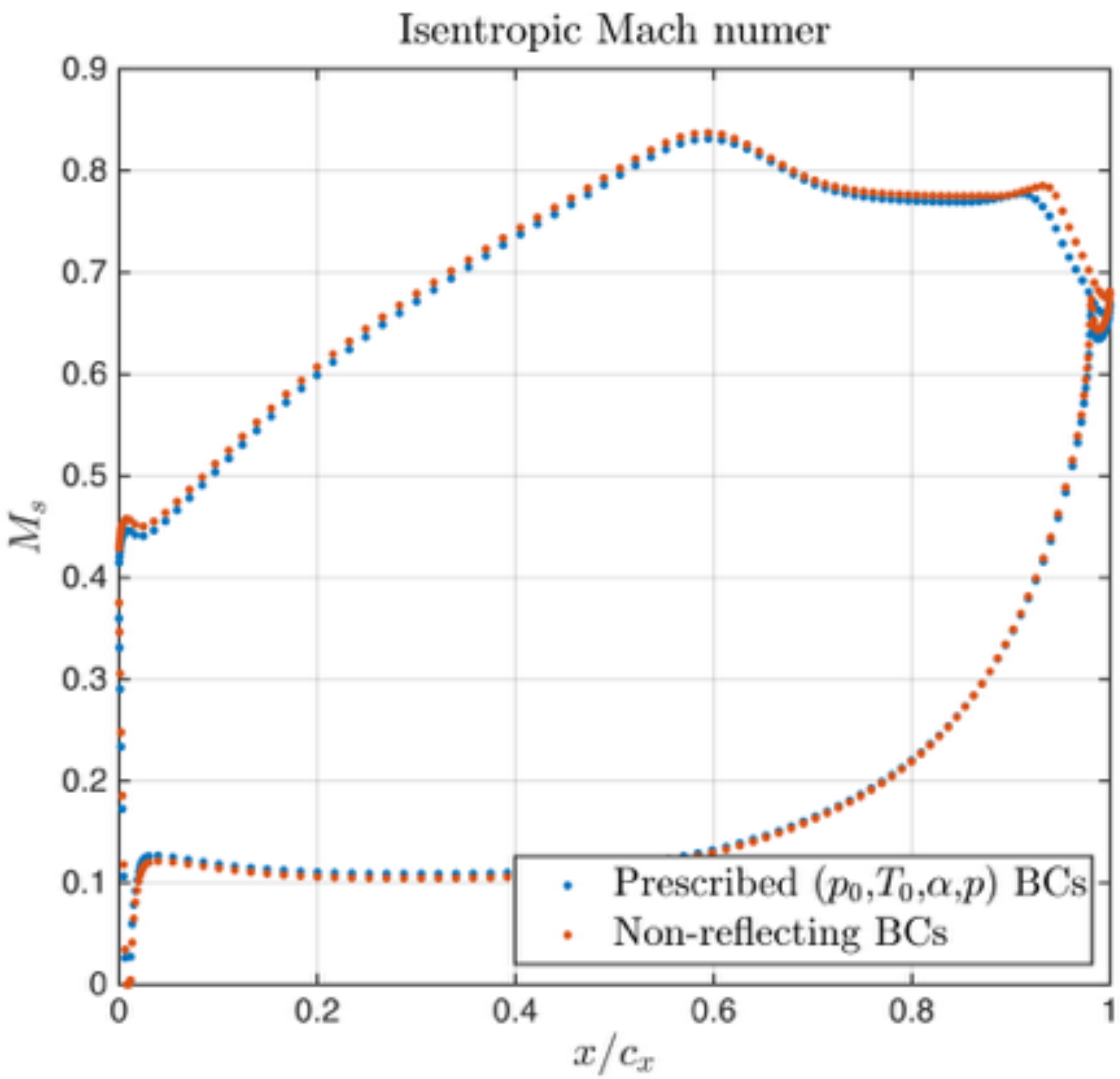
Time average Instantaneous

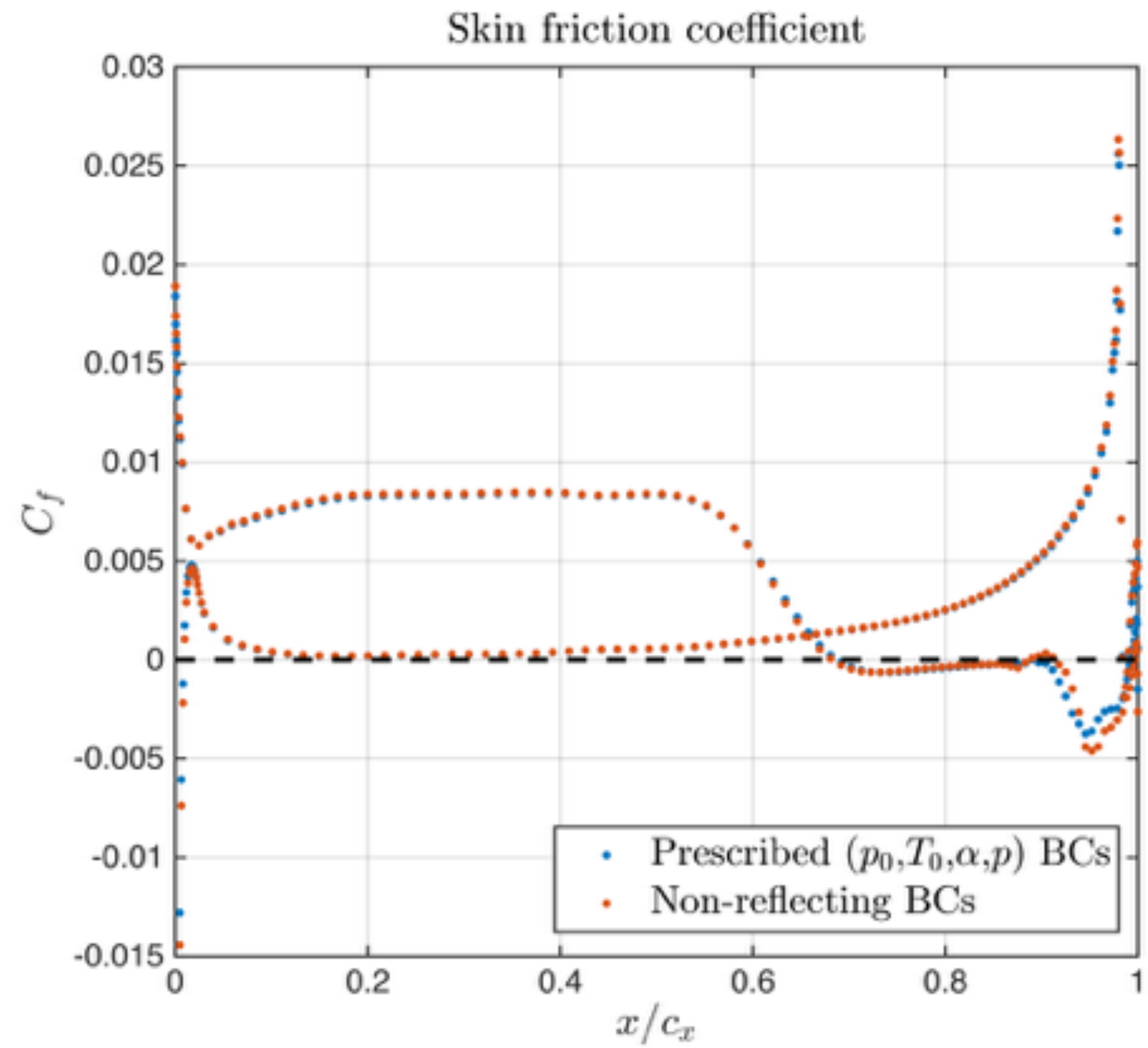
Non-dimensional grid size





Pressure and skin-friction coefficients

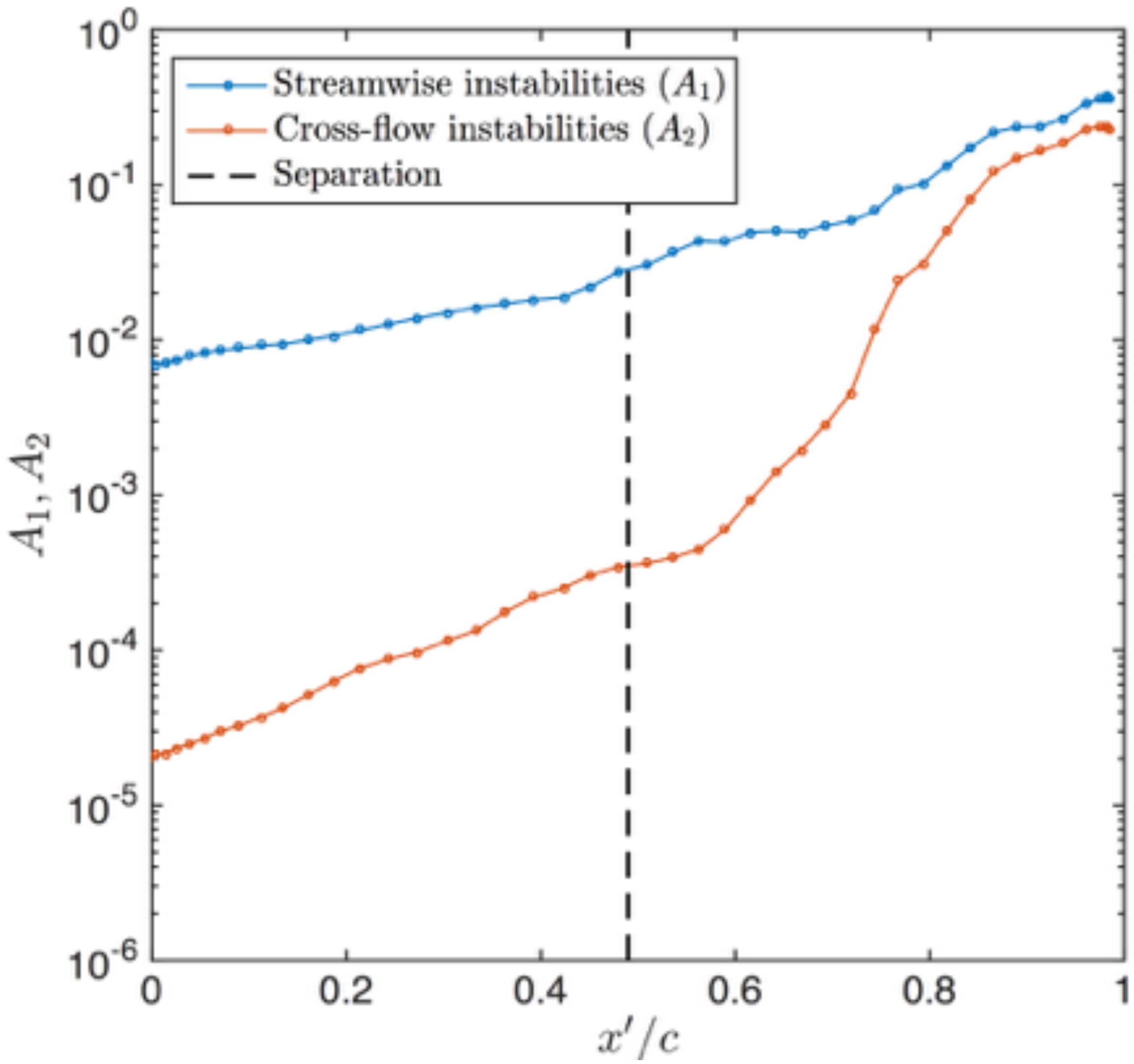




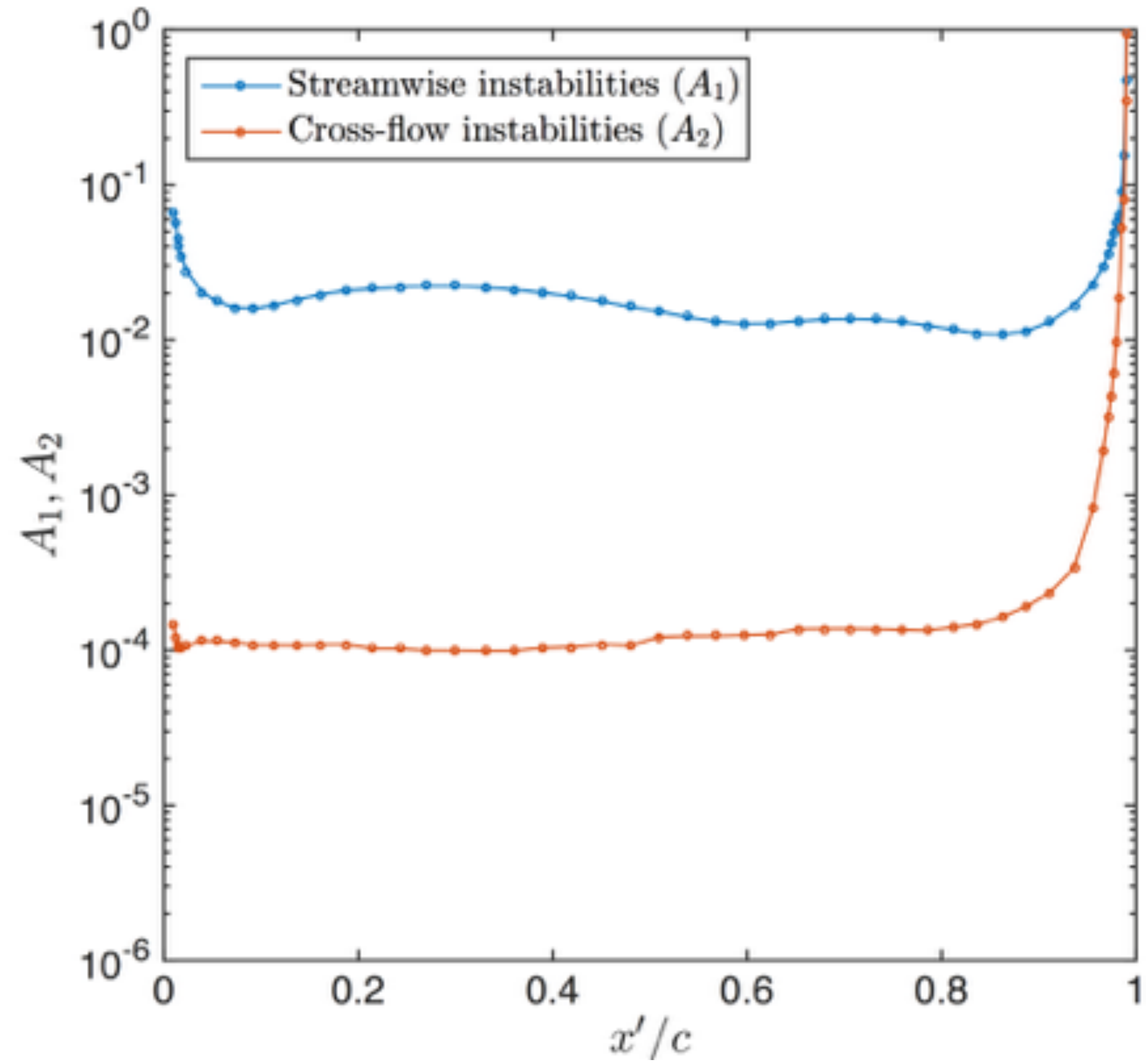
- Two types of inflow/outflow boundary conditions are considered
- Small differences in C_p and C_f observed

Analysis of boundary layer instabilities

Amplitude of instabilities - Suction side



Amplitude of instabilities - Pressure side

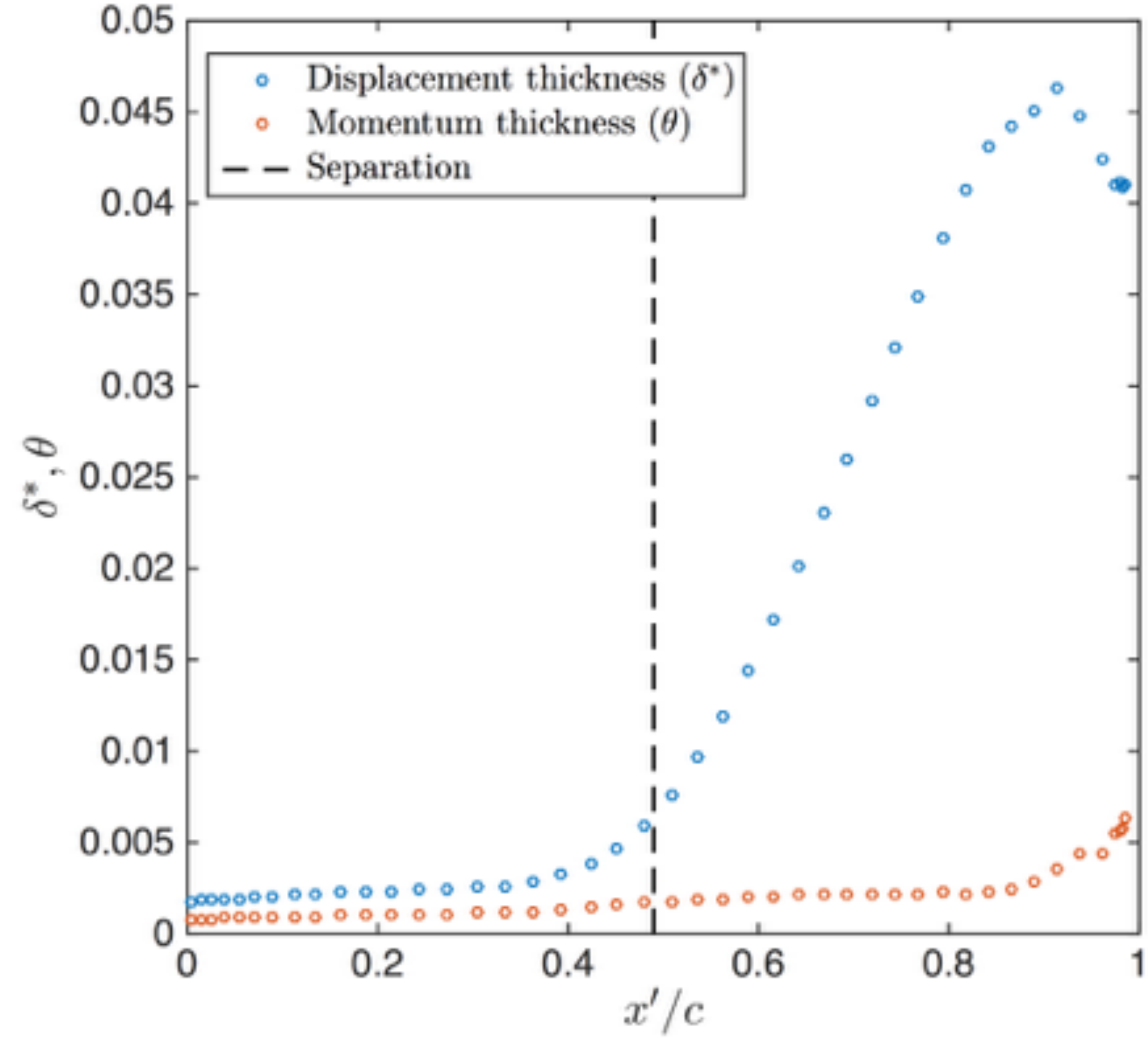


Natural transition through 2D unstable modes

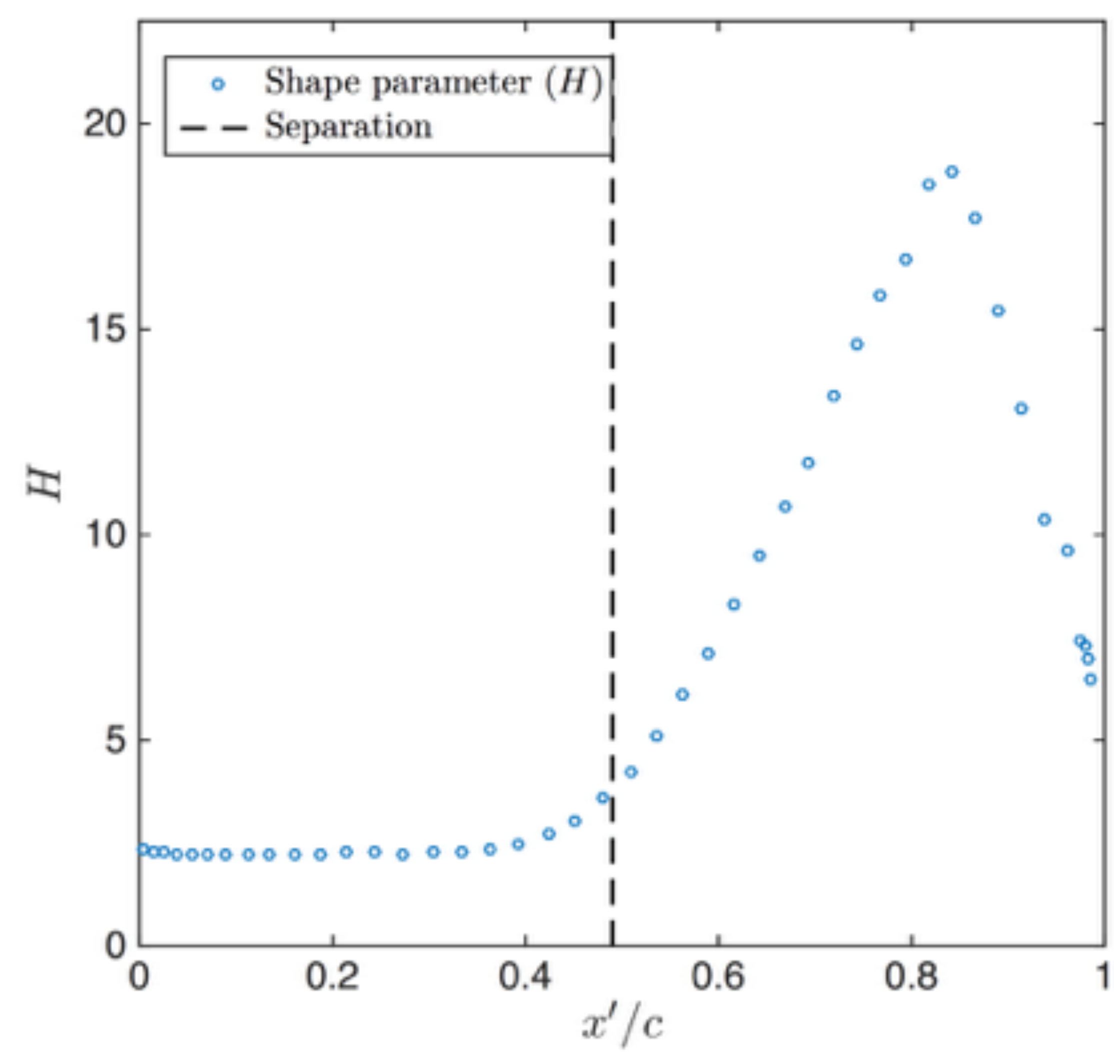
The nomenclature and details of the post-processing strategy are described in ***(Fernandez et al., 2017a)***

Suction side boundary layer

Displacement and momentum thickness



Shape parameter



The nomenclature and details of the post-processing strategy are described in ***(Fernandez et al., 2017a)***

Summary

- Implicit LES outperformed explicit SGS models for the transition prediction, wall-free turbulence and wall-bounded turbulence cases considered.
- Minor differences between Riemann solvers: DG methods have an auto-correction mechanism to compensate for overshoots in the Riemann solver.
- DG methods have a built-in implicit LES capability and add numerical dissipation in under-resolved turbulence simulations.
- Built-in implicit SGS model in DG methods is more accurate than explicit models since they add numerical dissipation only when SGS's are present in the flow
- Higher order are important to capture detailed physics in turbulent flows at higher Reynolds number.
- Optimal LES implementations are likely achieved with local $CFL_{local} > 1$ ($CFL_{global} >> 1$ for wall-bounded flows).

Summary

- IEDG is the hybridized DG method of choice: It inherits computational efficiency from EDG and BC robustness from HDG.
- Below ~5th order accuracy, IEDG allows for more efficient implementations than other DG methods.
- Beyond ~5th order accuracy, memory requirements and flop count become prohibitive, and IEDG does not provide any advantages.
- ILU(0) + RAS(1) type preconditioners are extremely efficient for most LES cases considered.
- Accuracy and stability of ILU deteriorates for low cell Peclet numbers (time-step size needs to be reduced more than linearly w.r.t. mesh size): Issues for wall-resolved LES at Re > 500,000.
- Efficiency and scalable nonlinear solvers are required for these methods to be adopted in industrial applications.

Questions?

For additional details:

P. Fernandez, N.C. Nguyen, J. Peraire, The hybridized Discontinuous Galerkin method for Implicit Large-Eddy Simulation of transitional turbulent flows, J. Comput. Phys. 336 (1) (2017) 308–329.

P. Fernandez, N.C. Nguyen, J. Peraire, Subgrid-scale modeling and implicit numerical dissipation in DG-based Large-Eddy Simulation, In: 23rd AIAA Computational Fluid Dynamics Conference, Denver, USA, 2017.

P. Fernandez, N.C. Nguyen, J. Peraire, A physics-based shock capturing method for unsteady laminar and turbulent flows, In: 56th AIAA Aerospace Sciences Meeting, Gaylord Palms, USA, 2018.

SciTech talk:

Title: A physics-based shock capturing method for unsteady laminar and turbulent flows

Session: FD-03, CFD for Capturing Flow Discontinuities

Setting: Monday, January 8, 9:30 AM, Room Sun 5

pablof@mit.edu cuongng@mit.edu peraire@mit.edu

Hybridized DG methods

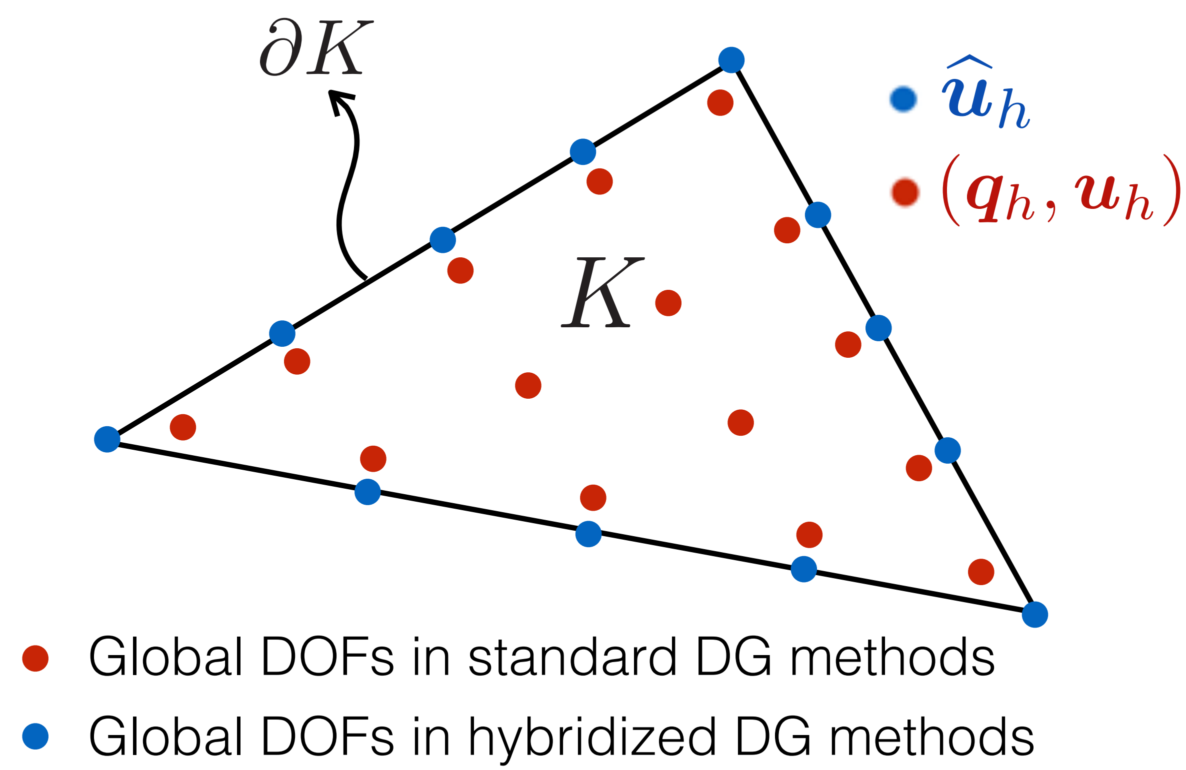
Hybridized DG schemes

- Consider the unsteady compressible Navier-Stokes equations:

$$\begin{aligned} \boldsymbol{q} - \nabla \boldsymbol{u} &= \boldsymbol{0}, \quad \text{in } \Omega \times (0,T] \ , \\ \frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}_{NS}(\boldsymbol{u}, \boldsymbol{q}) &= \boldsymbol{0}, \quad \text{in } \Omega \times (0,T]. \end{aligned} \qquad \boldsymbol{u} = \begin{pmatrix} \rho \\ \rho v_j \\ \rho E \end{pmatrix}$$

- The hybridized DG approach **(Nguyen et al., 2015)** introduces additional variables $\widehat{\boldsymbol{u}}_h$ on the element faces ∂K. Then:

$$(\boldsymbol{q}_h, \boldsymbol{u}_h)|_K = f(\widehat{\boldsymbol{u}}_h|_{\partial K}), \quad \forall K \in \mathcal{T}_h$$



- This yields a global problem:

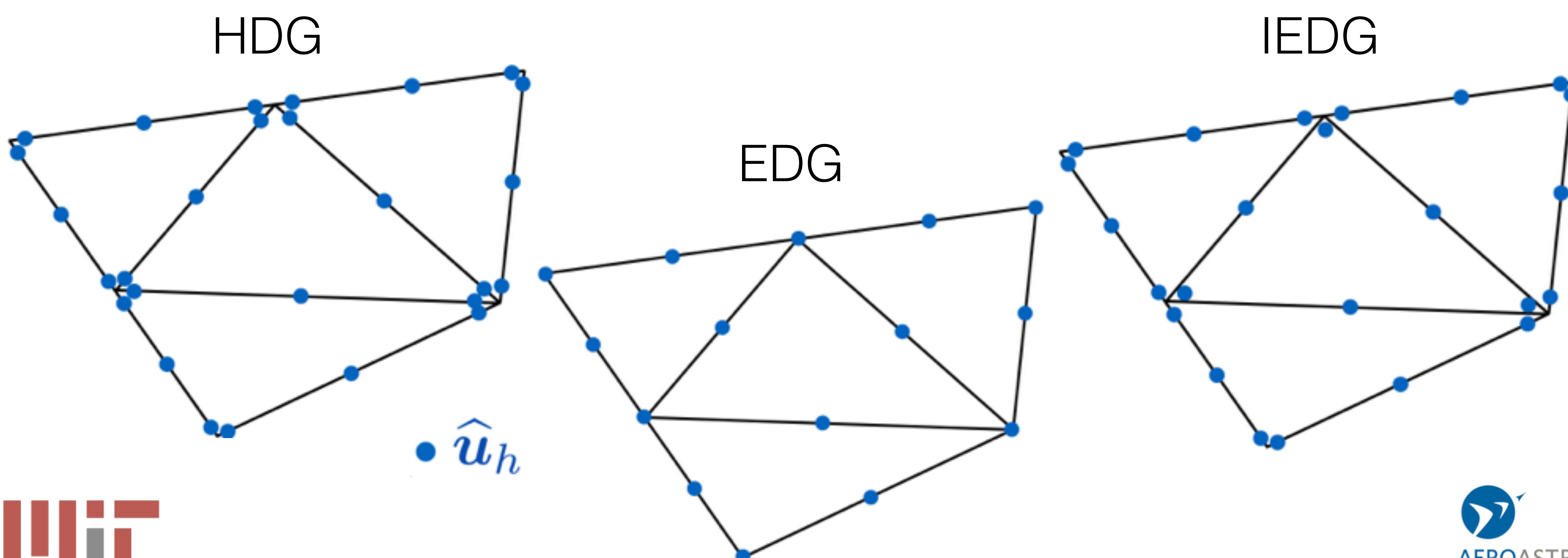
$$\boldsymbol{r}_h(\boldsymbol{u}_h) = \boldsymbol{0} \longrightarrow \boldsymbol{r}_h(\widehat{\boldsymbol{u}}_h) = \boldsymbol{0}$$

Standard DG — Hybridized DG

Hybridized DG schemes

- "Hybridized DG schemes" (Nguyen et al., 2015) are a family of numerical schemes
- Different choices of the space for $\widehat{\boldsymbol{u}}_h$ lead to different schemes. Three examples:

Method	Nature of $\widehat{\boldsymbol{u}}_h$ space
Hybridizable DG (HDG) (Peraire et al., 2010)	Discontinuous across faces
Embedded DG (EDG) (Peraire et al., 2011)	Continuous across faces
Interior Embedded DG (IEDG) (Fernandez et al., 2016)	Interior faces: Continuous Boundary faces: Discontinuous



MIT AEROASTRO

Hybridized DG. Efficiency and accuracy

Efficiency

- Non-zero entries in Jacobian of global system:

$$\mathrm{NNZ} = N_p N_c^2 \alpha_{\mathrm{NNZ}}$$

$\mathrm{NNZ} \equiv$ Number of non-zeros
$N_p \equiv$ Number of mesh vertices
$N_c \equiv$ Number of components of the PDE

Values of α_{NNZ} (tetrahedra)

	Accuracy order			
	2nd	3rd	4th	5th
Standard DG	480	3,000	12,000	36,750
HDG	756	3,024	8,400	18,900
EDG	15	230	1,311	4,410
IEDG	<15	<230	<1,311	<4,410

Accuracy

- Optimal accuracy ($p+1$) in $\boldsymbol{u}_h$.
- Superconvergence ($p+2$) in $\boldsymbol{u}_h$ can be inexpensively achieved (HDG only)
- Optimal accuracy ($p+1$) in $\boldsymbol{q}_h$ (HDG only)

	Accuracy order	
	$\boldsymbol{u}_h$	$\boldsymbol{q}_h$
Standard DG	$p+1$	p
HDG	$p+1$	$p+1$
EDG	$p+1$	p
IEDG	$p+1$	p

Parallel implementation and iterative solvers

Parallel iterative solver

- The hybridized DG discretization yields a nonlinear system of equations at every time-step

$$\boldsymbol{r}_h(\widehat{\boldsymbol{u}}_h) = \mathbf{0}$$

- An efficient and scalable solution procedure is required for these methods to be adopted in industrial applications

- We discuss next on parallel iterative solvers

Parallel iterative solver

$$\boldsymbol{r}_h(\widehat{\boldsymbol{u}}_h) = \boldsymbol{0}$$

$$\Big\downarrow \text{Linearization}$$

$$\boldsymbol{K}_h \; \delta\widehat{\boldsymbol{u}}_h = -\boldsymbol{r}_h$$

Nonlinear system: Newton or quasi-Newton method

- Initial guess $\widehat{\boldsymbol{u}}_h^{n,0}$ at time-step n computed with a reduced-basis minimum-residual algorithm:

$$\widehat{\boldsymbol{u}}_h^{n,0} = \sum_{j=1}^{s} \alpha_j \; \widehat{\boldsymbol{u}}_h^{n-j}$$

$$(\alpha_1, \ldots, \alpha_s) = \arg \min_{(\beta_1,\ldots,\beta_s)\in\mathbb{R}^s} \left\| \boldsymbol{r}_h\Big(\sum_{j=1}^{s} \beta_j \; \widehat{\boldsymbol{u}}_h^{n-j}\Big) \right\|^2$$

Parallel iterative solver

$$\boldsymbol{r}_h(\widehat{\boldsymbol{u}}_h) = \boldsymbol{0}$$

$$\Big\downarrow \text{Linearization}$$

$$\boldsymbol{K}_h \; \delta\widehat{\boldsymbol{u}}_h = -\boldsymbol{r}_h$$

Nonlinear system: Newton or quasi-Newton method

- Initial guess $\widehat{\boldsymbol{u}}_h^{n,0}$ at time-step n computed with a reduced-basis minimum-residual algorithm

Linear system: Restarted GMRES method

- Parallel preconditioner $\boldsymbol{M}^{-1}$: "Traced node"-based $\delta-$ overlapping restrictive additive Schwarz **(Cai et al., 1999)**:

$$\boldsymbol{M}^{-1} := \sum_{i=1}^{N} \boldsymbol{R}_i^0 \; (\boldsymbol{K}_h)_i^{-1} \; \boldsymbol{R}_i^\delta$$

$$(\boldsymbol{K}_h)_i = \boldsymbol{R}_i^\delta \; \boldsymbol{K}_h \; \boldsymbol{R}_i^\delta$$

$\boldsymbol{R}_i^\beta \equiv$ Restriction operator onto the $\beta-$ overlap subdomain

$N \equiv$ Number of subdomains (i.e. processors)

- Subdomain preconditioner: Block incomplete LU factorization with zero fill-in (BILU0) and MDF reorder **(Persson et al., 2008)**

$$(\boldsymbol{K}_h)_i^{-1} \approx \widetilde{\boldsymbol{U}}_i^{-1} \; \widetilde{\boldsymbol{L}}_i^{-1}$$

Parallel iterative solver

$$\boldsymbol{r}_h(\widehat{\boldsymbol{u}}_h) = \boldsymbol{0}$$

$$\Big\downarrow \text{Linearization}$$

$$\boldsymbol{K}_h \; \delta\widehat{\boldsymbol{u}}_h = -\boldsymbol{r}_h$$

Nonlinear system: Newton or quasi-Newton method

- Initial guess $\widehat{\boldsymbol{u}}_h^{n,0}$ at time-step n computed with a reduced-basis minimum-residual algorithm

Linear system: Restarted GMRES method

- Parallel preconditioner $\boldsymbol{M}^{-1}$: "Traced node"-based $\delta-$ overlapping restrictive additive Schwarz **(Cai et al., 1999)**
- Subdomain preconditioner: Block incomplete LU factorization with zero fill-in (BILU0) and MDF reorder **(Persson et al., 2008)**

Parallel iterative solver

$$\boldsymbol{r}_h(\widehat{\boldsymbol{u}}_h) = \boldsymbol{0}$$

$$\Big\downarrow \text{Linearization}$$

$$\boldsymbol{K}_h \; \delta\widehat{\boldsymbol{u}}_h = -\boldsymbol{r}_h$$

Nonlinear system: Newton or quasi-Newton method

- Initial guess $\widehat{\boldsymbol{u}}_h^{n,0}$ at time-step n computed with a reduced-basis minimum-residual algorithm

Linear system: Restarted GMRES method

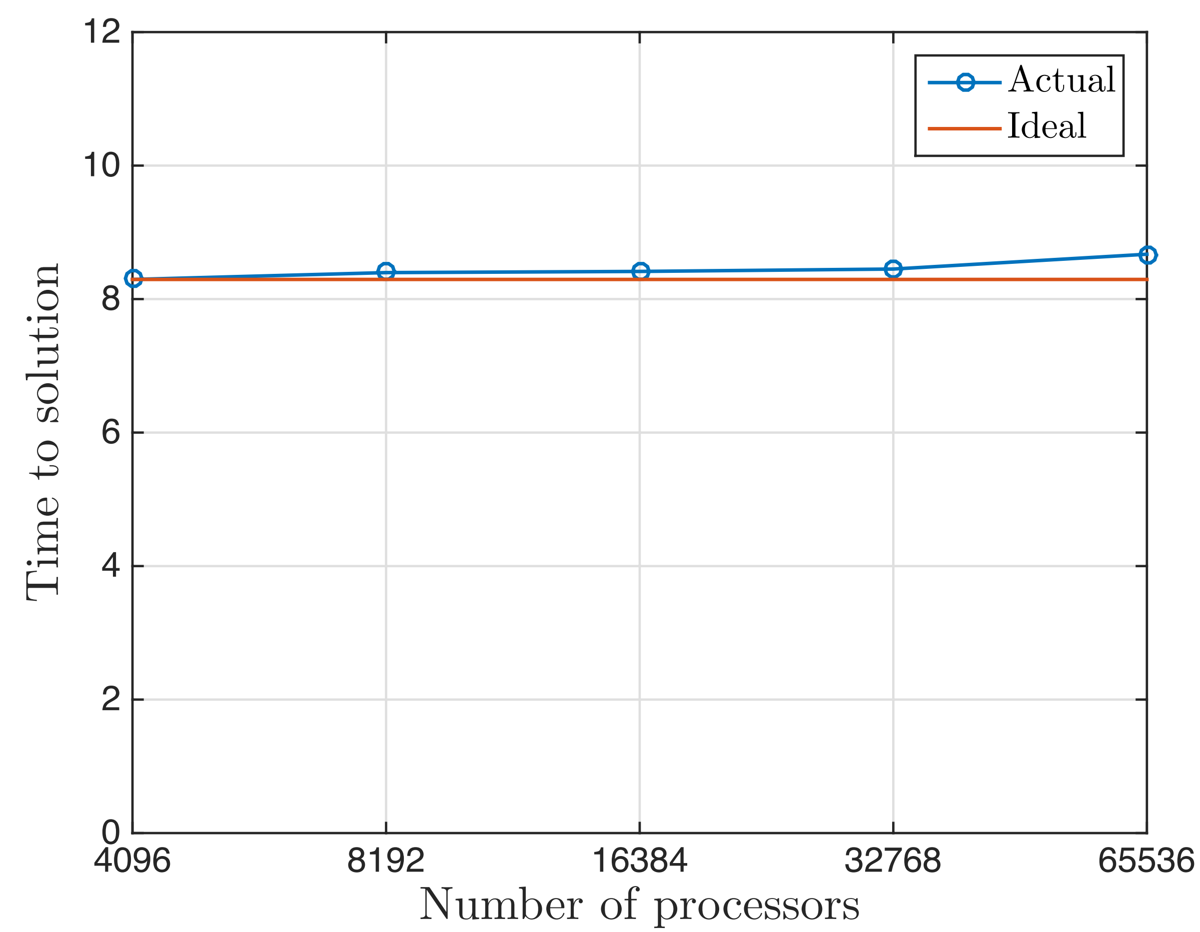
- Parallel preconditioner $\boldsymbol{M}^{-1}$: "Traced node"-based $\delta-$ overlapping restrictive additive Schwarz (Cai et al., 1999)
- Subdomain preconditioner: Block incomplete LU factorization with zero fill-in (BILU0) and MDF reorder (Persson et al., 2008)

Others:

- Mixed-precision approach for Newton-GMRES algorithm
- Adaptive quadrature rules
- Stabilized ILU factorization (sILU)
- Minimum Interaction Domain Decomposition (MIDD)

Parallel scalability

- Weak scaling for LES of Ecole Centrale de Lyon compressor cascade
- Low-speed compressor cascade with Reynolds numbers ranging from 200,000 to 400,000
- Numerical discretization:
  - Space: 3rd-order IEDG
  - Time: 3rd-order DIRK(3,3)
- Additional details:
  - Time-step size is kept content in all runs
  - Computing platform: Titan (OLCF)
  - One MPI rank per physical core
  - GPU computing and hybrid OpenMP/MPI parallelization disabled



References

1. X.C. Cai, M. Sarkis, A restricted additive Schwarz preconditioner for general sparse linear systems, SIAM J. Sci. Comput. 21 (2) (1999) 792–797.
2. P. Fernandez, N.C. Nguyen, X. Roca, J. Peraire, Implicit large-eddy simulation of compressible flows using the Interior Embedded Discontinuous Galerkin method, In: 54th AIAA Aerospace Sciences Meeting, San Diego, USA, 2016.
3. P. Fernandez, N.C. Nguyen, J. Peraire, The hybridized Discontinuous Galerkin method for Implicit Large-Eddy Simulation of transitional turbulent flows, J. Comput. Phys. 336 (1) (2017) 308–329.
4. P. Fernandez, N.C. Nguyen, J. Peraire, Subgrid-scale modeling and implicit numerical dissipation in DG-based Large-Eddy Simulation, In: 23rd AIAA Computational Fluid Dynamics Conference, Denver, USA, 2017.
5. P. Fernandez, N.C. Nguyen, J. Peraire, A physics-based shock capturing method for unsteady laminar and turbulent flows, In: 56th AIAA Aerospace Sciences Meeting, Gaylord Palms, USA, 2018.
6. N.C. Nguyen, J. Peraire, B. Cockburn, A class of embedded discontinuous Galerkin methods for computational fluid dynamics, J. Comput. Phys. 302 (1) (2015) 674–692.
7. J. Peraire, N.C. Nguyen, B. Cockburn, A Hybridizable Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations, In: 48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition, Orlando, USA, 2010.
8. J. Peraire, N.C. Nguyen, B. Cockburn, An Embedded Discontinuous Galerkin Method for the Compressible Euler and Navier-Stokes Equations, In: 20th AIAA Computational Fluid Dynamics Conference, Honolulu, USA, 2011.
9. P.-O. Persson, J. Peraire, Newton-GMRES preconditioning for Discontinuous Galerkin Discretizations of the Navier-Stokes Equations, SIAM J. Sci. Comput. 30 (6) (2008) 2709–2733.