Basics of DG methods with applications to compressible and incompressible flows

code MIGALE state-of-the-art

F. Bassi[1], L. Botti[1], A. Colombo[1], A. Crivellini[3],
A. Ghidoni[2], G. Manzinali[1], F. Massa[1],
A. Nigro[4], G. Noventa[2], S. Rebay[2]

HPC methods for
Computational Fluid Dynamics and Astrophysics
CINECA, Bologna (Italy)
4th November 2016

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

...with the contribution of

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Francesco Bassi[1]
Alessandro Colombo[1]
Lorenzo Botti[1]
Francesco Carlo Massa[1]
Marco Savini[1]
Nicoletta Franchina[1]

Antonio Ghidoni[2]
Gianmaria Noventa[2]
Marco Lorini[2]
Stefano Rebay[2]

Andrea Crivellini[3]

Carmine De Bartolo[4]
Alessandra Nigro[4]

Daniele Di Pietro[5]

Pietro Tesini[6]

[1] Università degli Studi di Bergamo
[2] Università degli Studi di Brescia
[3] Università Politecnica delle Marche
[4] Università della Calabria
[5] University of Montpellier, France
[6] SKF, Sweden

TiLDA $\frac{2018}{2015}$

Towards Industrial LES/DNS in Aeronautics
Paving the Way for Future Accurate CFD
grant agreement No.635962

…with the contribution of

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Francesco Bassi[1]
Alessandro Colombo[1]
Lorenzo Botti[1]
Francesco Carlo Massa[1]
Marco Savini[1]
Nicoletta Franchina[1]

Antonio Ghidoni[2]
Gianmaria Noventa[2]
Marco Lorini[2]
Stefano Rebay[2]

Andrea Crivellini[3]

Carmine De Bartolo[4]
Alessandra Nigro[4]

Daniele Di Pietro[5]

Pietro Tesini[6]

[1] Università degli Studi di Bergamo
[2] Università degli Studi di Brescia
[3] Università Politecnica delle Marche
[4] Università della Calabria
[5] University of Montpellier, France
[6] SKF, Sweden

TILDA 2015–2018

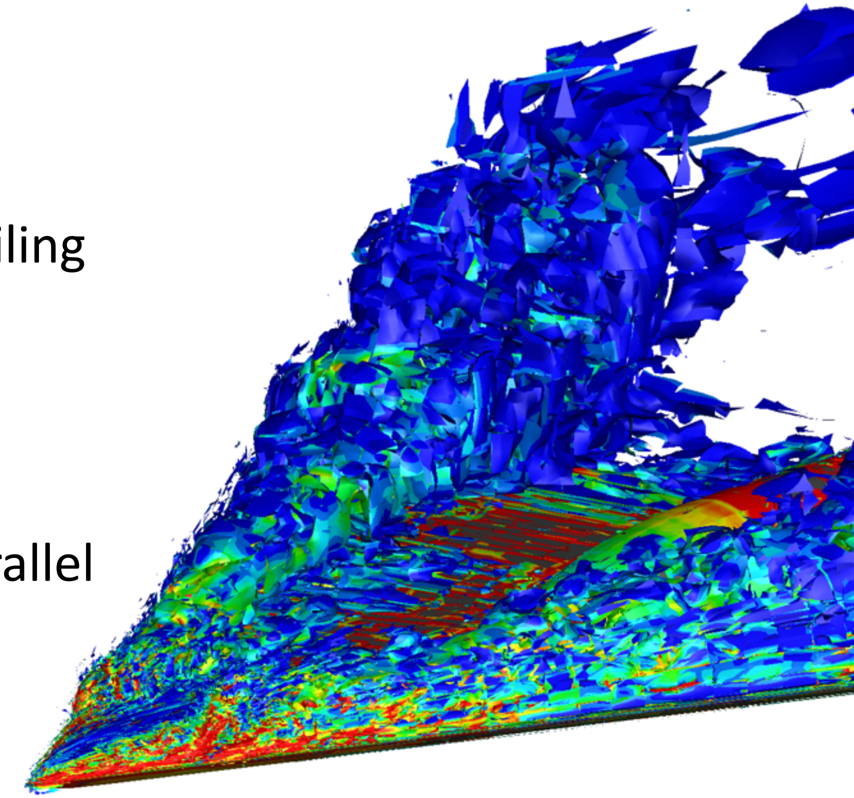
Towards Industrial LES/DNS in Aeronautics
Paving the Way for Future Accurate CFD
grant agreement No.635962

IDIHOM 2010–2014

ADIGMA 2006–2009

The Discontinuous Galerkin (DG) methods for CFD?

UNIVERSITÀ DEGLI STUDI DI BERGAMO



Pros

- Great geometrical flexibility without spoiling at all the accuracy
- Straightforward implementation of h/p - adaptive techniques
- Compact stencil, useful for massively parallel computer platforms

Cons

The higher accuracy comes at an increased computational cost with respect to “standard” FD or FV

Growing interest in their application to unsteady problems to address complex and computationally demanding simulations of turbulent flows

Purpose of this presentation

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

To give an overview of the building blocks we chose for a DG implementation the most general and flexible as possible

Our goal is to deal with different flow models using a unified numerical framework, e.g., time integrators, Riemann solvers

- Discontinuous Galerkin (DG) method on hybrid grids
- Physical frame orthonormal basis functions
- 2D/3D steady and unsteady compressible and incompressible flows
- Explicit and implicit time accurate integration
- Fixed or rotating frame of reference

- Euler
- Navier–Stokes
- RANS + k-ω (EARSM)
- Hybrid RANS/LES (X-LES)

MPI parallelism
Fortran language

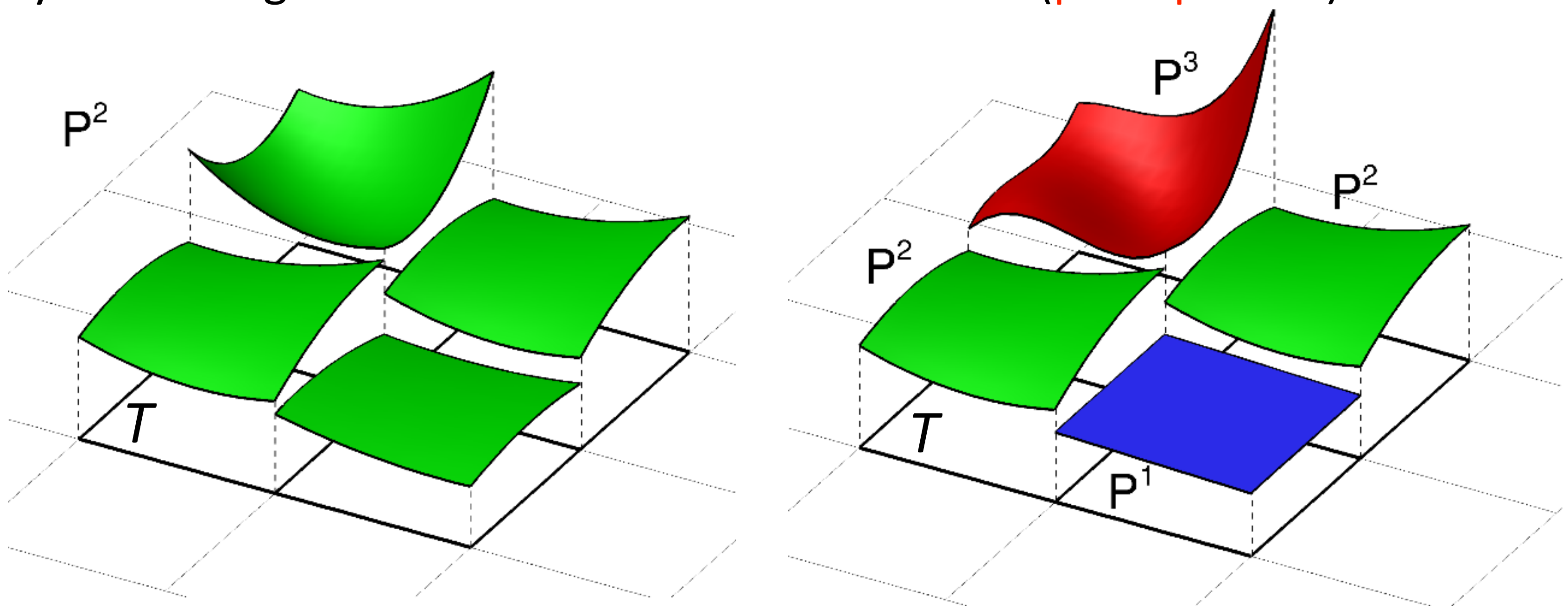
The DG method in a nutshell

the solution approximation

The numerical solution is approximated by high-order polynomial functions

Functions are not required to be continuous across the elements interfaces

DG methods allow in a natural way to locally vary the polynomial degree of the solution in each element (p-adaptation)



For quadrature computations any element $T \in \mathcal{T}_h$ (or parts…) can be mapped on a reference element T_{ref}, *e.g.* the unit quadrangle

The DG method in a nutshell

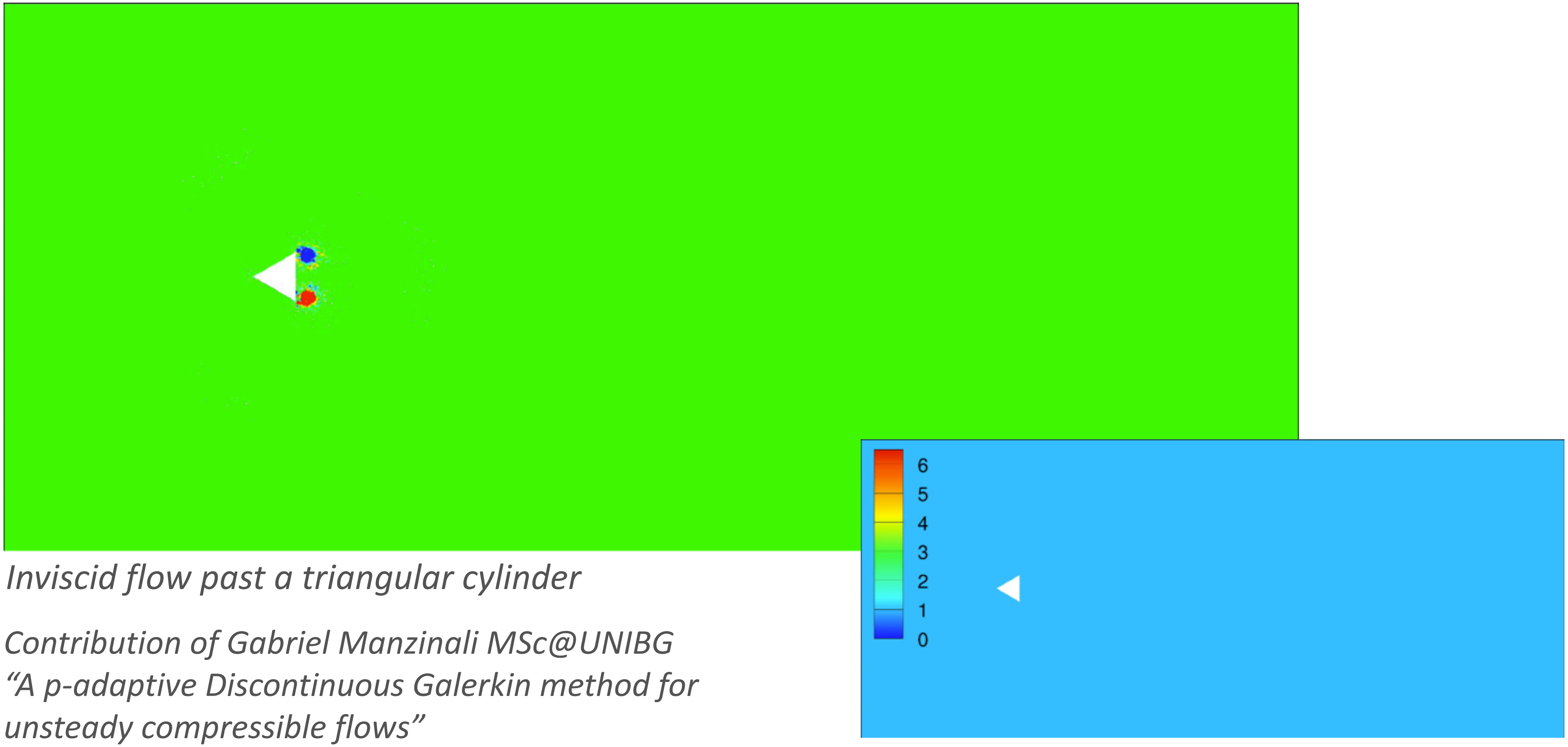
the solution approximation

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

The numerical solution is approximated by high-order polynomial functions

Functions are not required to be continuous across the elements interfaces

DG methods allow in a natural way to locally vary the polynomial degree of the solution in each element (p-adaptation)



Inviscid flow past a triangular cylinder

Contribution of Gabriel Manzinali MSc@UNIBG
"A p-adaptive Discontinuous Galerkin method for unsteady compressible flows"

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

The basis functions

The basis functions

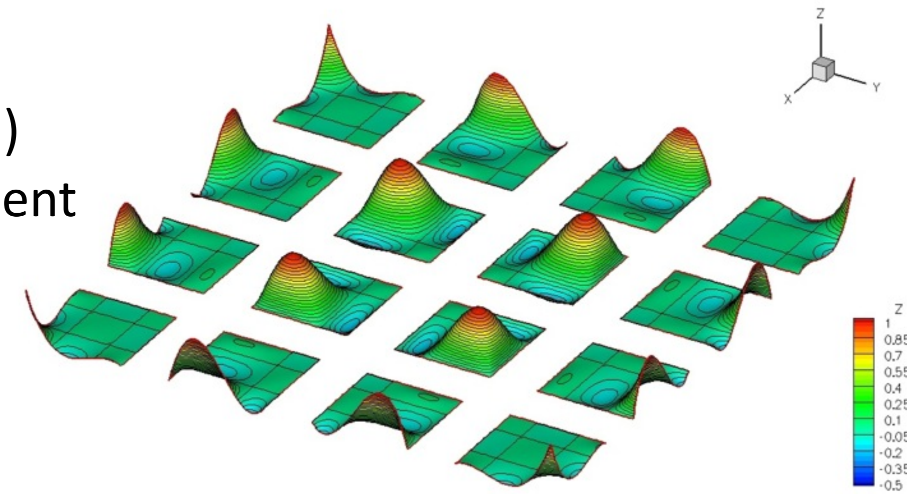
opportunities…

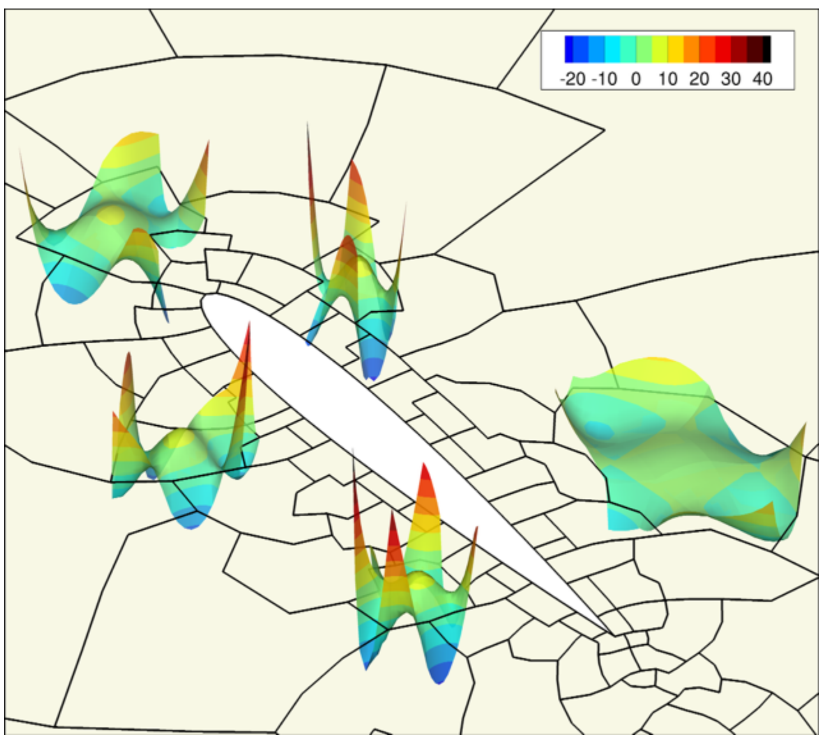
UNIVERSITÀ DEGLI STUDI DI BERGAMO

Basis functions can be defined

- on reference elements (quad, tria, …) and then *mapped* on the mesh element

$$\mathbf{w}(\boldsymbol{\xi}, t)|_{T_{ref}} = \sum_i \mathbf{W}_i(t)\phi_i(\boldsymbol{\xi})$$





- on real (mesh) element of any shape

$$\mathbf{w}(\mathbf{x}, t)|_{T} = \sum_i \mathbf{W}_i(t)\phi_i(\mathbf{x})$$

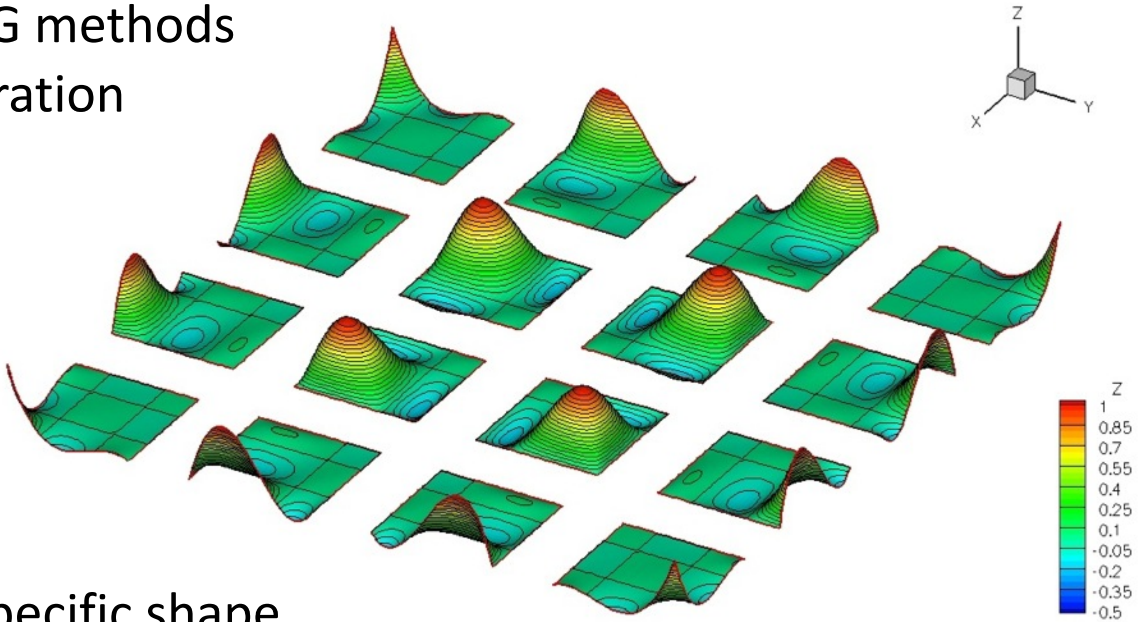
…both have pros and cons…

The basis functions

Basis functions on a reference frame

Pros

efficiency proper of nodal DG methods with interpolation and integration nodes coincident



Cons

- defined for elements of specific shape
- stability issues for Legendere-Gauss-Lobatto nodes (aliasing → over-integration)
- polynomials on the reference element are no more polynomials on real elements with curved edges

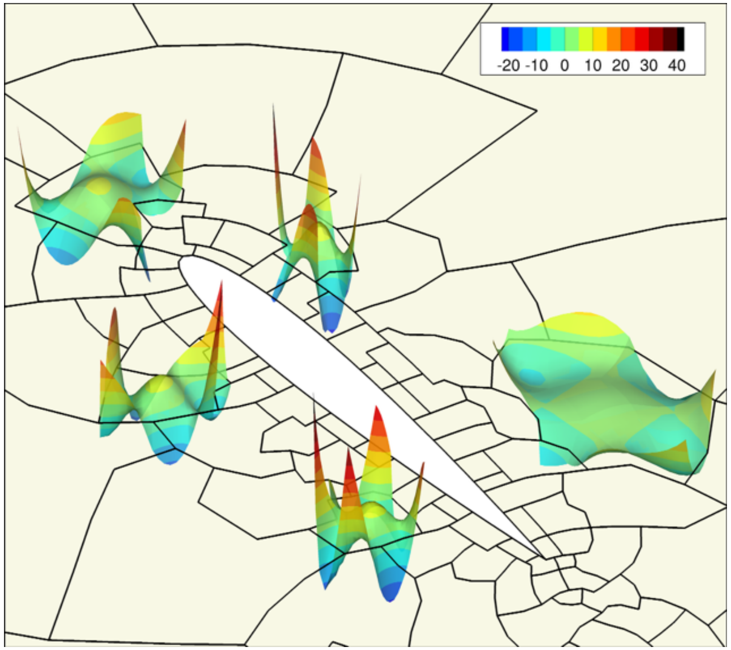
The basis functions

Basis functions on the physical frame

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Pros

- defined for arbitrary shape possibly curved elements
- well-conditioned orthogonal and hierarchical shape functions
- polynomials are exactly represented and integrated
- provide the basic framework for appealing *h*-multigrid techniques



Cons

- cost of integration
- inefficiency due to modal representation

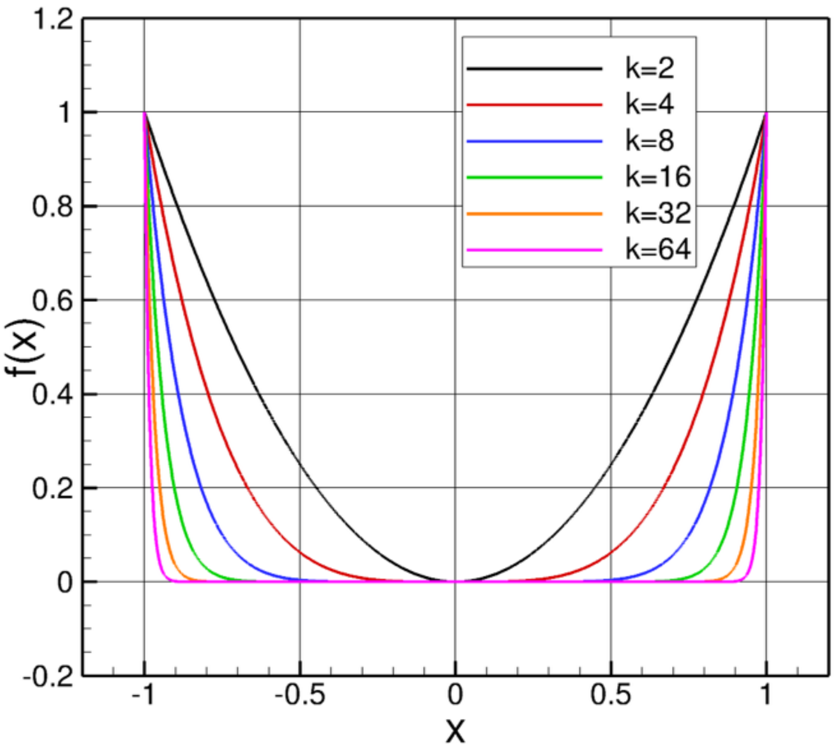
The basis functions - physical frame

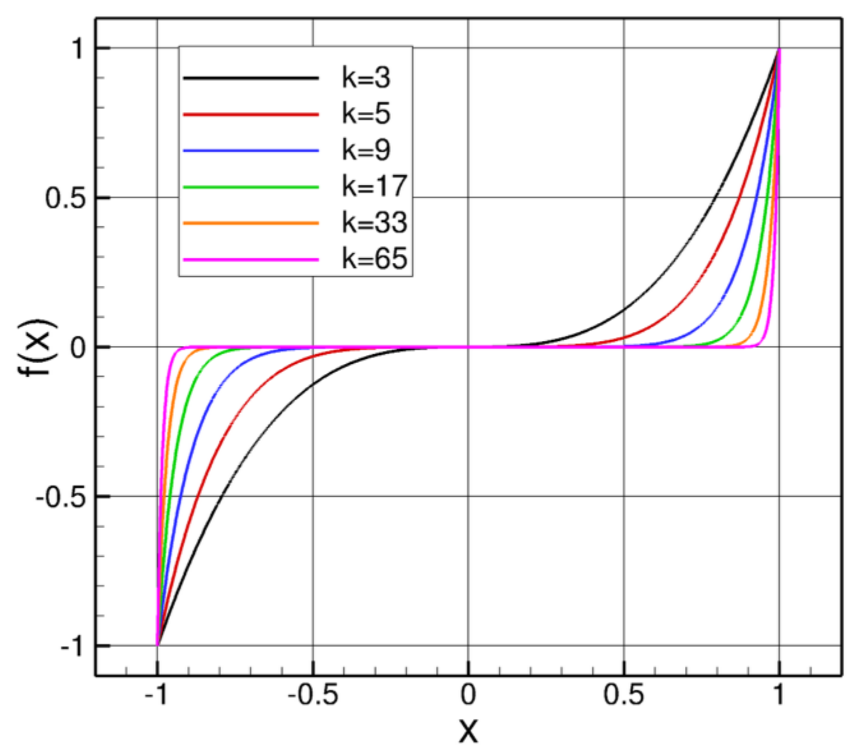
An orthonormal and hierarchical set

We define discrete polynomial spaces in physical coordinates

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\mathrm{def}}{=} \left\{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}_d^k(T),\, \forall T \in \mathcal{T}_h\right\}$$

A trivial choice as the monomial basis leads to ill-conditioned linear systems particularly when dealing with highly stretched elements





The basis functions - physical frame

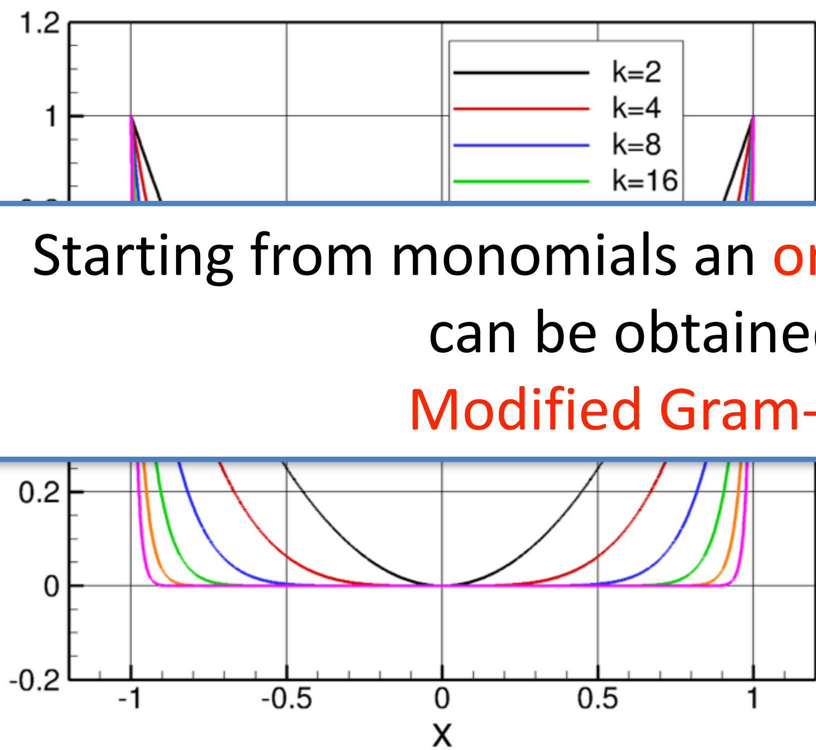
An orthonormal and hierarchical set

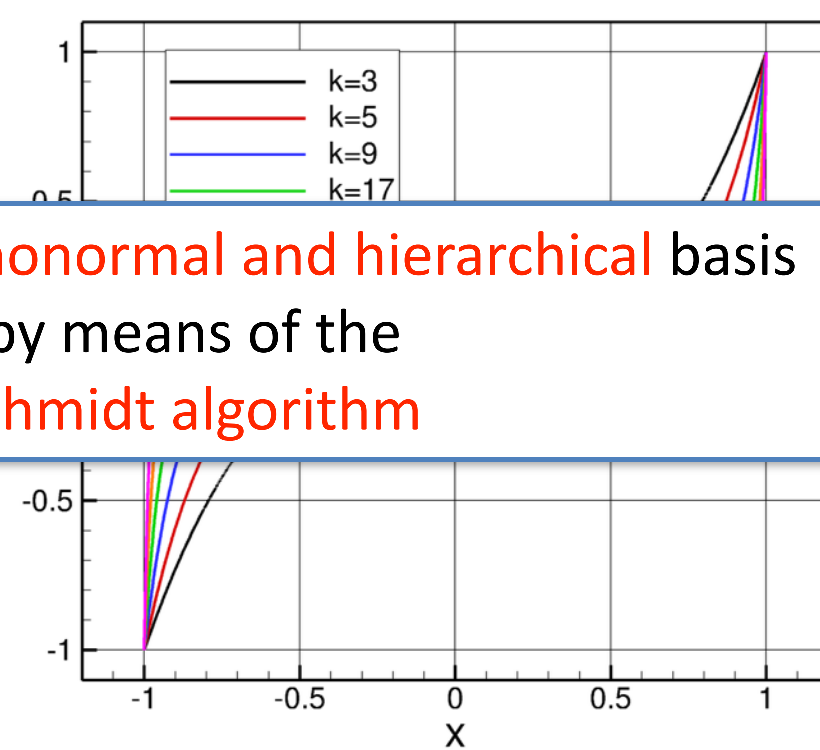
UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \left\{ \phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}_d^k(T),\ \forall T \in \mathcal{T}_h \right\}$$

A trivial choice as the monomial basis leads to ill-conditioned linear systems particularly when dealing with highly stretched elements





Starting from monomials an orthonormal and hierarchical basis can be obtained by means of the Modified Gram-Schmidt algorithm

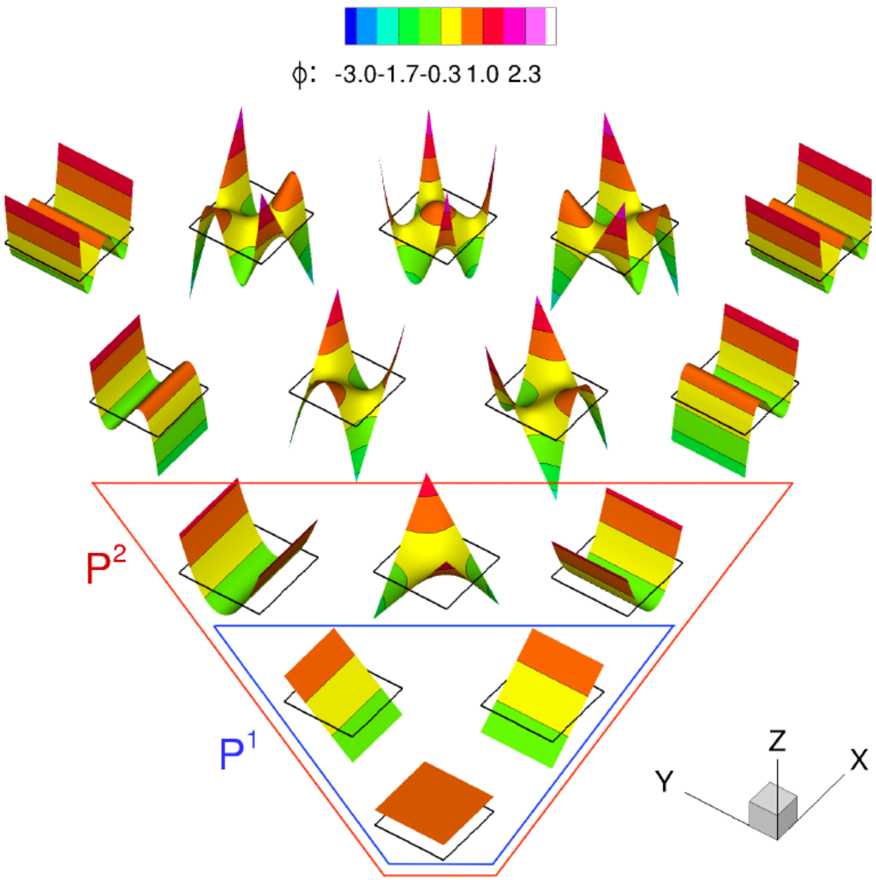
The basis functions - physical frame

An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}_d^k(T),\, \forall T \in \mathcal{T}_h\}$$



for *i=1 to* N_{dof}^T ***do***

  for *j=1 to i-1* ***do***

    $r_{ij}^T \leftarrow (b_i^T, \phi_j^T)_T$

    $b_i^T \leftarrow b_i^T - r_{ij}^T \phi_j^T$

  end for

  $r_{ii}^T \leftarrow [(b^T, \phi^T)_T]^{1/2}$

  $b_i^T \leftarrow b_i^T / r_{ii}^T$

  $\phi_i^T \leftarrow b_i^T$

end for

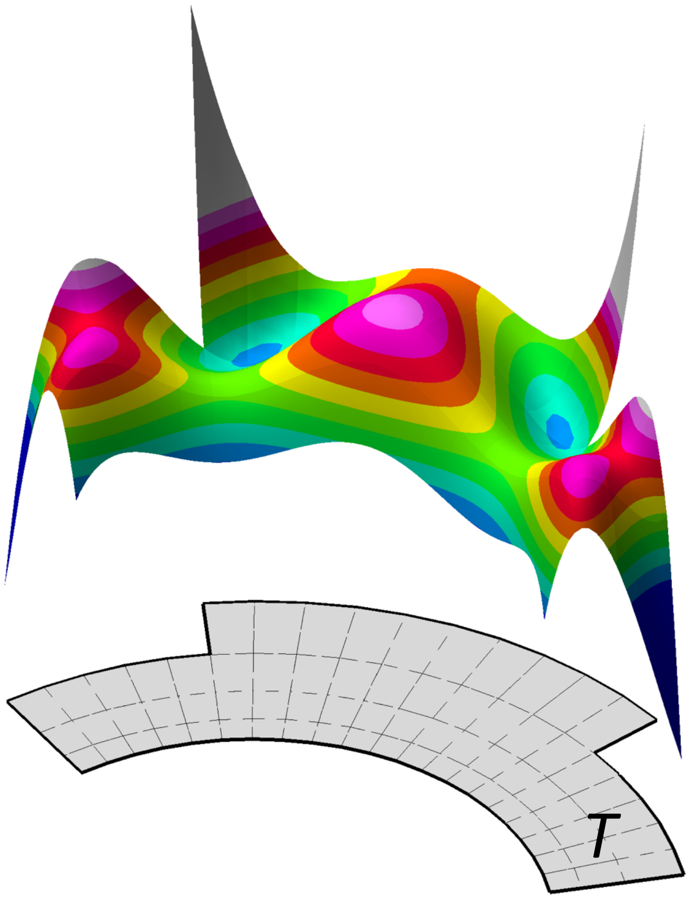
The basis functions - physical frame

An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}_d^k(T),\, \forall T \in \mathcal{T}_h\}$$



The only requirement to build such basis is to be able to perform integration

We can deal with elements of any shape, possibly curve

In the context of mesh elements built via agglomeration on top of a finer grid made of canonical elements we perform integration on the sub-elements

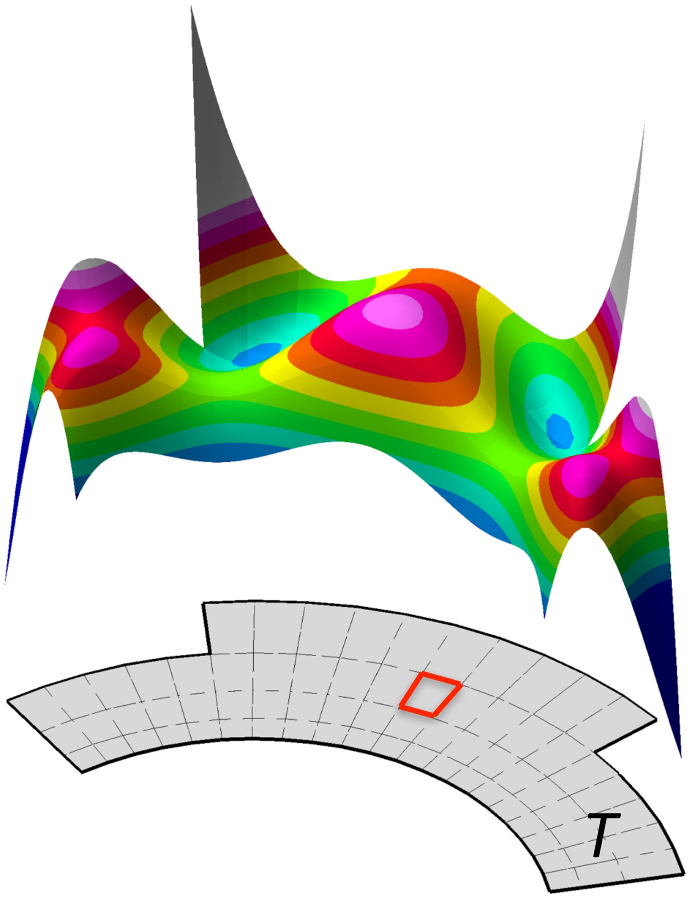
The basis functions - physical frame

An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

We define discrete polynomial spaces in physical coordinates

$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \left\{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}_d^k(T),\ \forall T \in \mathcal{T}_h\right\}$$



The only requirement to build such basis is to be able to perform integration

We can deal with elements of any shape, possibly curve

In the context of mesh elements built via agglomeration on top of a finer grid made of canonical elements we perform integration on the sub-elements

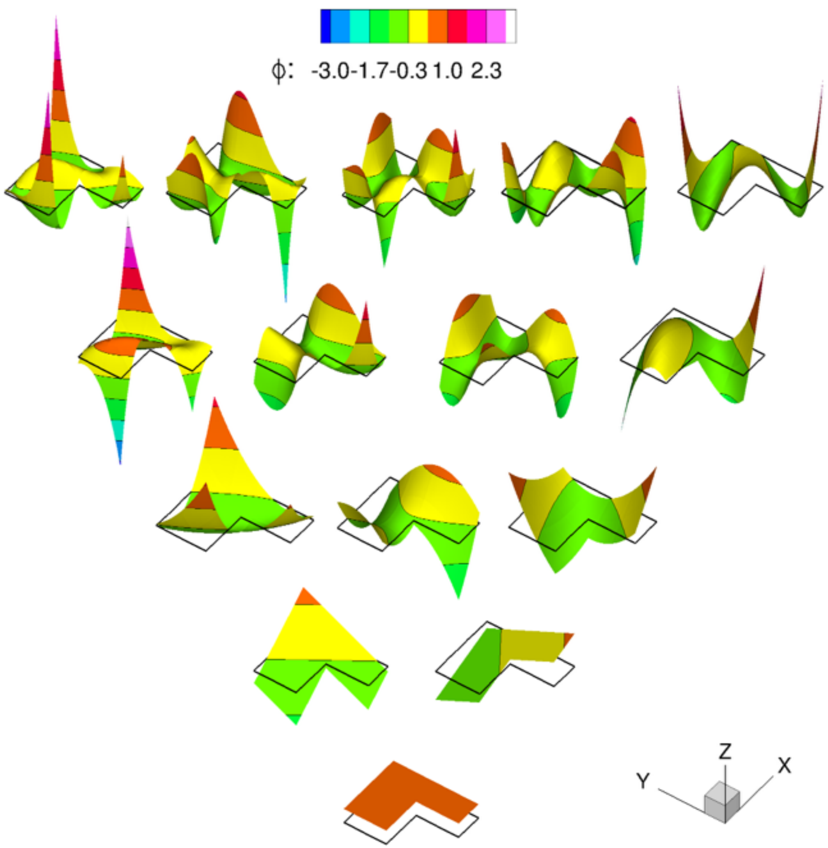
The basis functions - physical frame

An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

$$\mathbb{P}^k_d(\mathcal{T}_h) \stackrel{\text{def}}{=} \left\{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}^k_d(T),\ \forall T \in \mathcal{T}_h\right\}$$



The only requirement to build such basis is to be able to perform integration

We can deal with elements of any shape, possibly curve

In the context of mesh elements built via agglomeration on top of a finer grid made of canonical elements we perform integration on the sub-elements

The basis functions - physical frame

L2-Projection test: quadrilateral vs. polygonal elements

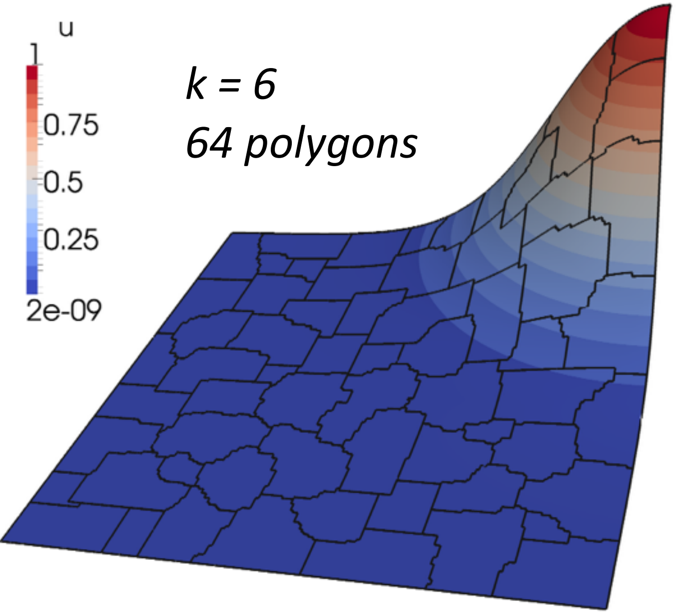
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

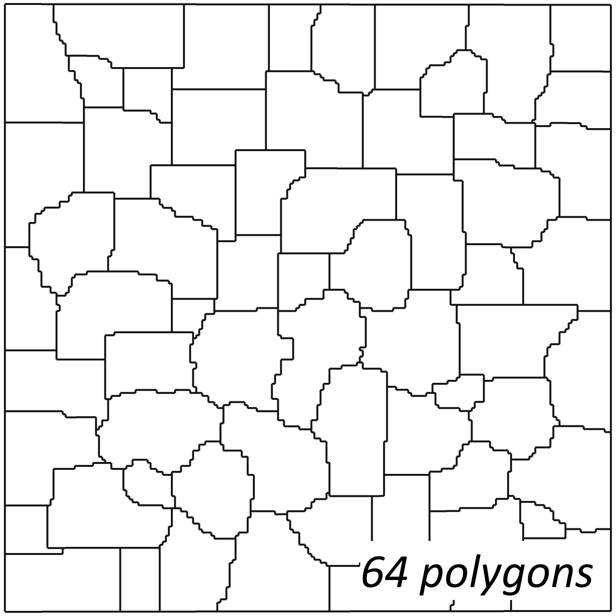
Test on the exact solution of a Poisson problem proposed in [Karniadakis and Sherwin, 2005]

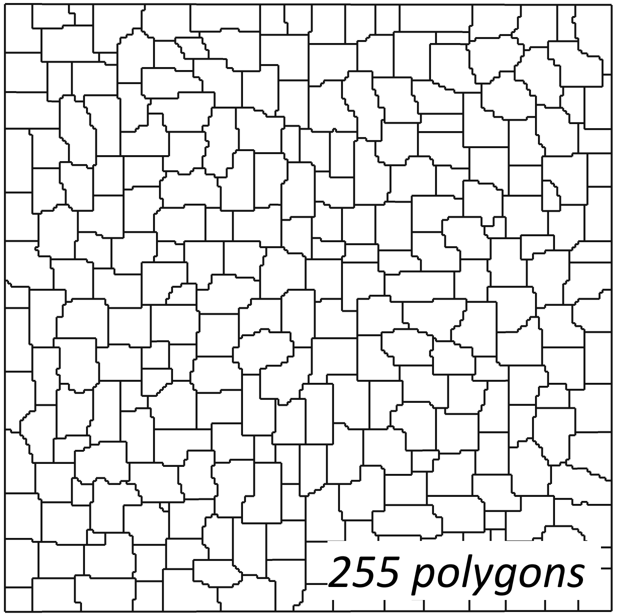
$$u = e^{-2.5[(x-1)^2+(y-1)^2]} \qquad \Omega = [-1,1]^2$$

<u>mesh sequences</u>

- 64, 256, 1028, 4096 uniform quadrilaterals grids
- 64, 255, 1028, 4122 polygonal elements grids built on top of a 200x200 quadrilaterals grid using MGridGen[1]





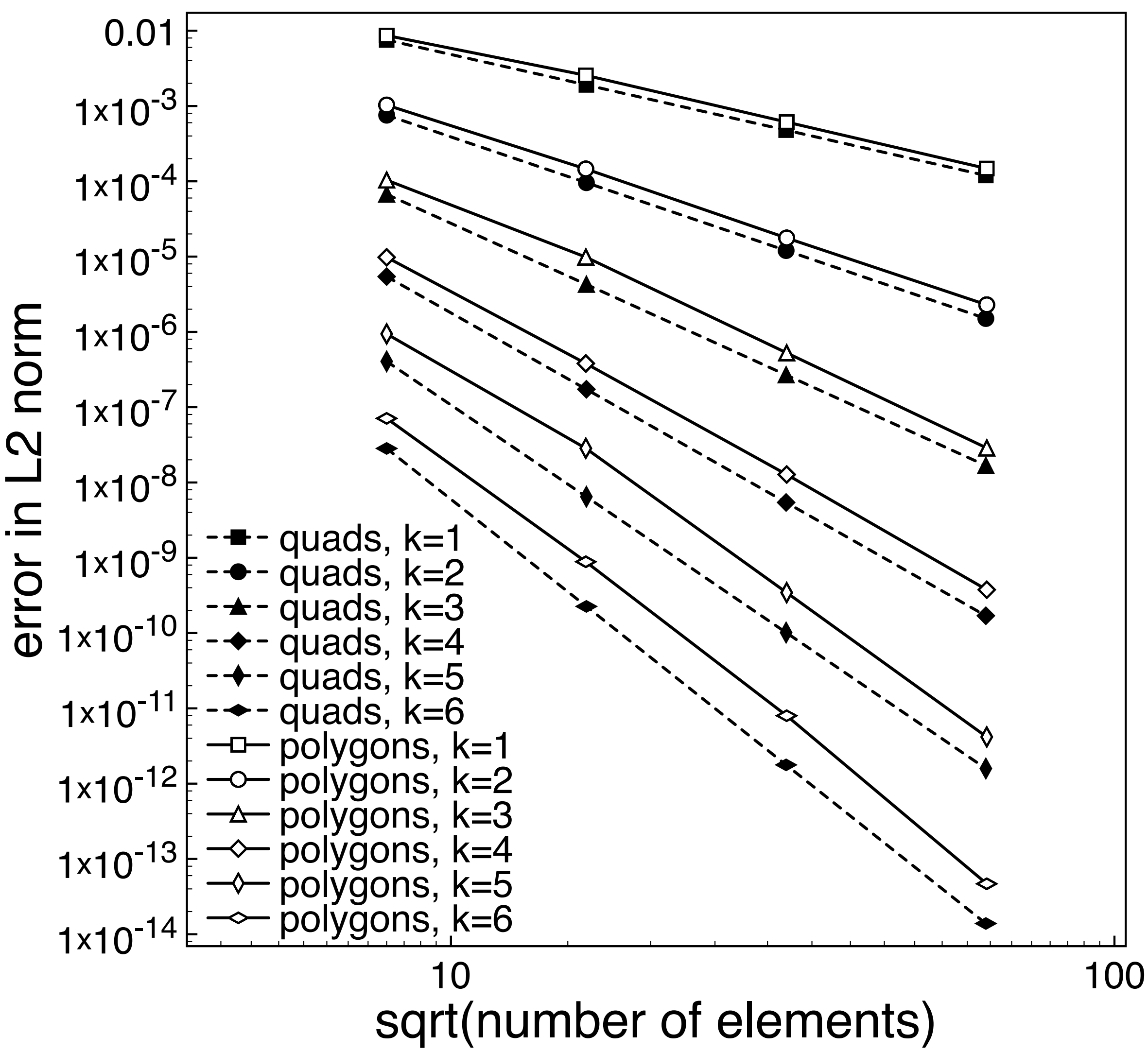


1) *http://www-users.cs.umn.edu/~moulitsa/software.html*

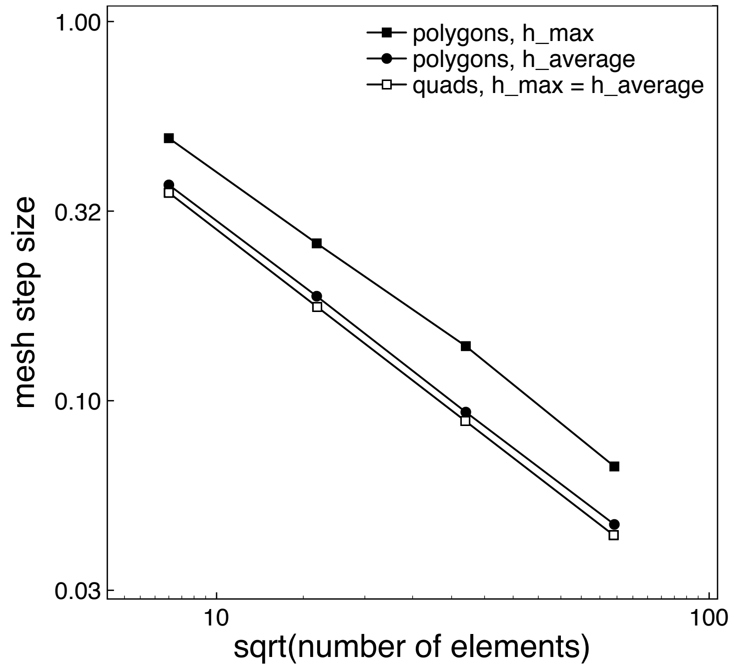
The basis functions - physical frame

L2-Projection test: quadrilateral vs. polygonal elements

UNIVERSITÀ DEGLI STUDI DI BERGAMO



The DG convergence rate (h^{k+1}) on both uniform quadrilateral and polygonal elements meshes is attained

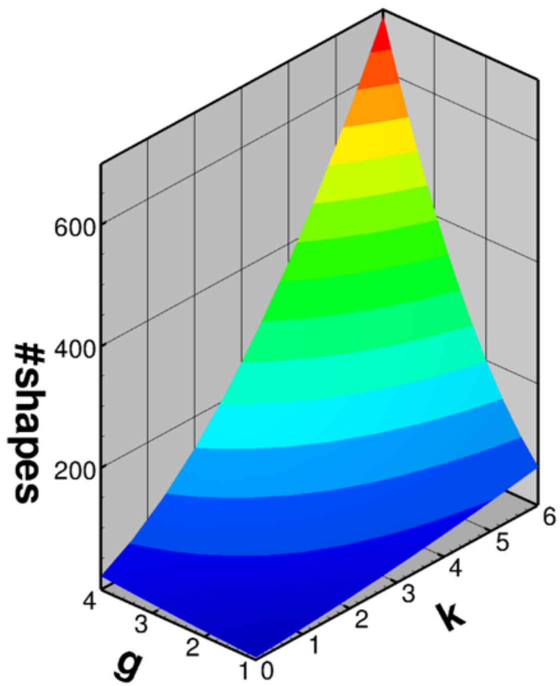


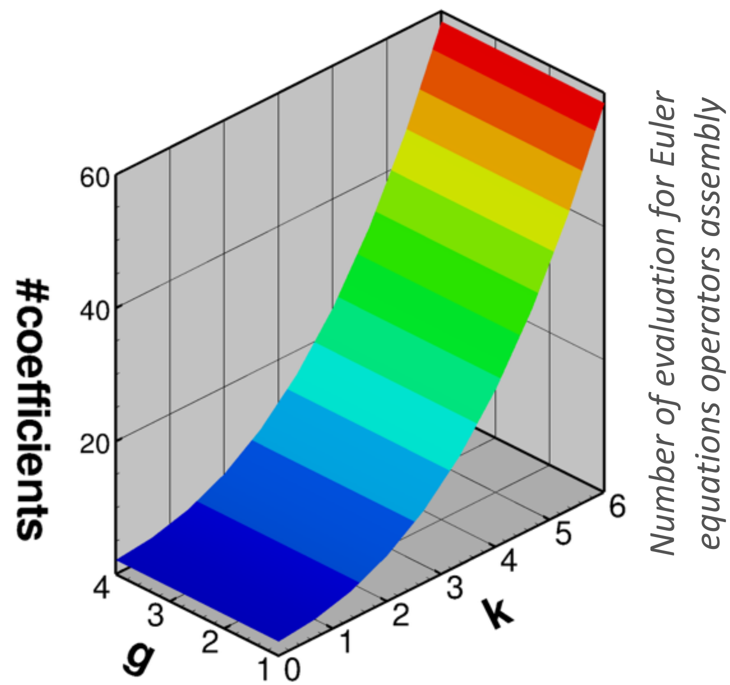
Hints on an efficient implementation of physical frame basis functions

UNIVERSITÀ DEGLI STUDI DI BERGAMO

To assembly the DG operators we will integrate over mesh elements $T \in \mathcal{T}_h$

The evaluation of basis functions (and their derivatives) at each quadrature point (QP) can strongly affect the solver performance





In particular when dealing with high-order discretizations on curved meshes (***g***>*1*)!

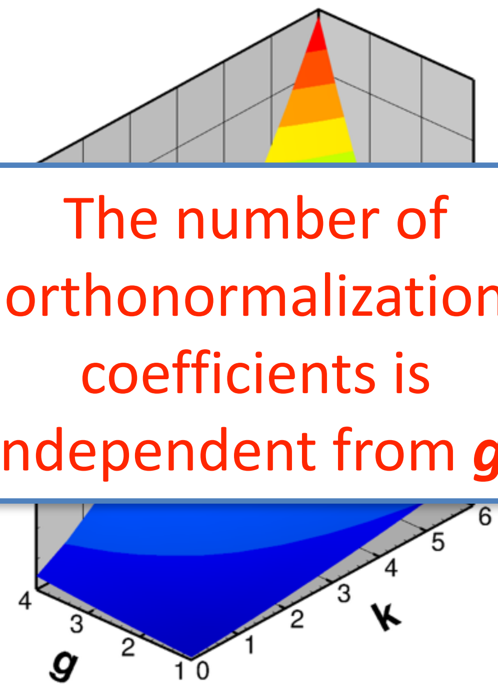
g *is the polynomial degree of the reference-to-physical-frame mapping* $\mathbf{x}(\boldsymbol{\xi})$

Hints on an efficient implementation of physical frame basis functions

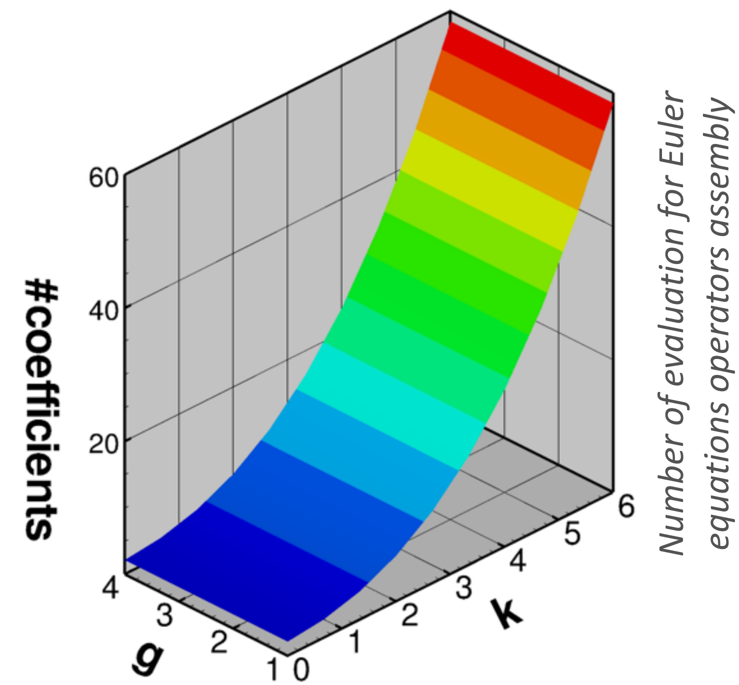
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

To assembly the DG operators we will integrate over mesh elements $T \in \mathcal{T}_h$

The evaluation of basis functions (and their derivatives) at each quadrature point (QP) can strongly affect the solver performance



The number of orthonormalization coefficients is independent from ***g***!



In particular when dealing with high-order discretizations on curved meshes (***g***>*1*)!

g *is the polynomial degree of the reference-to-physical-frame mapping* $\mathbf{x}(\boldsymbol{\xi})$

Hints on an efficient implementation of physical frame basis functions

UNIVERSITÀ DEGLI STUDI DI BERGAMO

To assembly the DG operators we will integrate over mesh elements $T \in \mathcal{T}_h$

The evaluation of basis functions (and their derivatives) at each quadrature point (QP) can strongly affect the solver performance

for $i=1$ to N_{dof}^T **do**

  for $j=1$ to $i-1$ **do**

    $r_{ij}^T \leftarrow (b_i^T, \phi_j^T)_T$

    $b_i^T \leftarrow b_i^T - r_{ij}^T \phi_j^T$

  end for

  $r_{ii}^T \leftarrow [(b^T, \phi^T)_T]^{1/2}$

  $b_i^T \leftarrow b_i^T / r_{ii}^T$

  $\phi_i^T \leftarrow b_i^T$

end for

$$\text{Monomial basis} \xrightarrow[\text{coefficients } r_{ii},\ r_{ij}]{MGS} \text{Orthonormal basis}$$

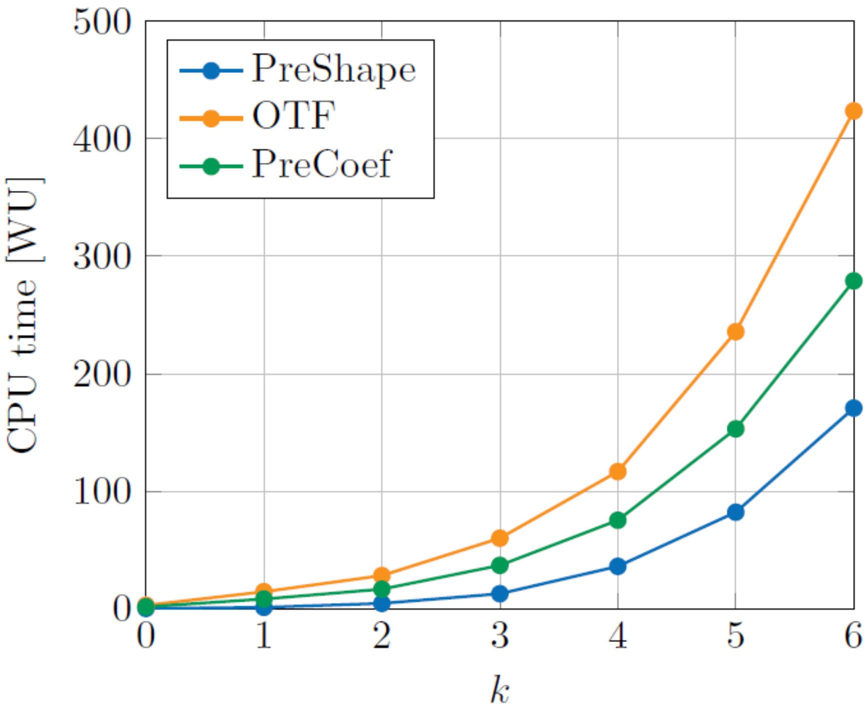
Strategy	Coefficients evaluation	Shapes evaluation	CPU usage	Memory footprint
OTF	During assembly	During assembly	High	Low
PreCoef	During pre-proc.	During assembly	Medium	Medium
PreShape	During pre-proc.	During pre-proc.	Low	High

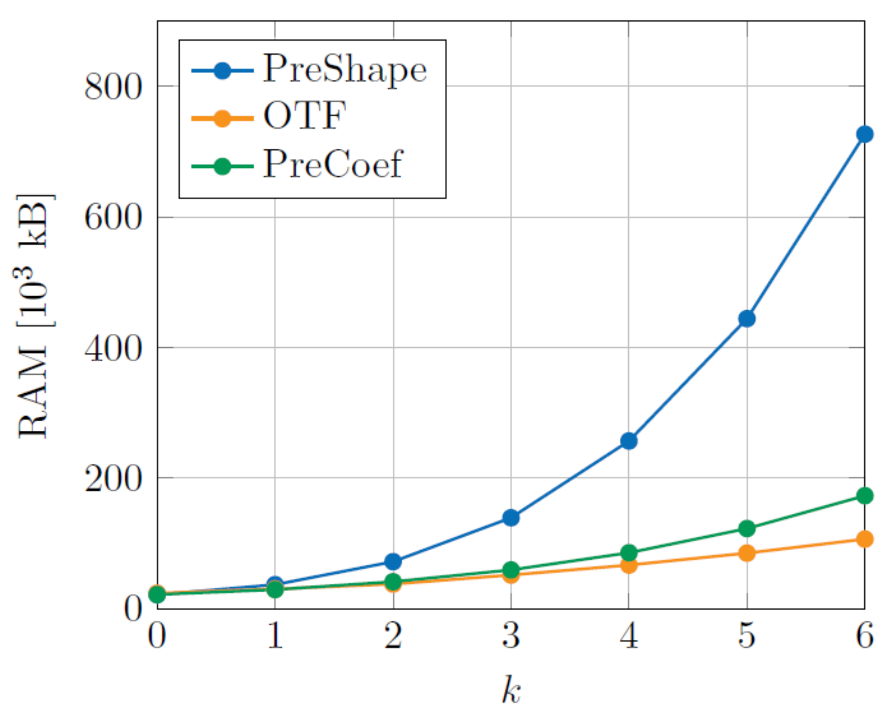
Hints on an efficient implementation of physical frame basis functions

UNIVERSITÀ DEGLI STUDI DI BERGAMO

To assembly the DG operators we will integrate over mesh elements $T \in \mathcal{T}_h$

The evaluation of basis functions (and their derivatives) at each quadrature point (QP) can strongly affect the solver performance





Performance test on the inviscid isentropic vortex transported by a uniform flow
100x100 straight-sided quadrilateral elements

Hints on an efficient implementation of physical frame basis functions

UNIVERSITÀ DEGLI STUDI DI BERGAMO

To assembly the DG operators we will integrate over mesh elements $T \in \mathcal{T}_h$

The evaluation of basis functions (and their derivatives) at each quadrature point (QP) can strongly affect the solver performance

$\mathrm{CPU}_*/\mathrm{CPU}_{\mathrm{PreShape}}$

k	PreShape	OTF	PreCoef
0	$3.18e{-1}$	9.46	5.54
1	$1.26e{+0}$	11.61	6.66
2	$4.64e{+0}$	6.10	3.61
3	$1.28e{+1}$	4.68	2.88
4	$3.62e{+1}$	3.23	2.08
5	$8.22e{+1}$	2.87	1.86
6	$1.71e{+2}$	2.48	1.63

$\%\Delta\mathrm{RAM}$

k	PreShape	OTF	PreCoef
0	$2.17e{+1}$	$+10.3\%$	$+1.2\%$
1	$3.70e{+1}$	-18.0%	-20.5%
2	$7.22e{+1}$	-47.4%	-42.5%
3	$1.39e{+2}$	-62.9%	-57.3%
4	$2.57e{+2}$	-73.9%	-66.6%
5	$4.44e{+2}$	-80.8%	-72.3%
6	$7.27e{+2}$	-85.3%	-76.2%

CPU ratio and ΔRAM percentage for PreCoef and OTF with respect to the PreShape performance ([WU] and [10^3 kB])

Hints on an efficient implementation of physical frame basis functions

UNIVERSITÀ DEGLI STUDI DI BERGAMO

To assembly the DG operators we will integrate over mesh elements $T \in \mathcal{T}_h$

The evaluation of basis functions (and their derivatives) at each quadrature point (QP) can strongly affect the solver performance

An overall best strategy for physical frame shapes evaluation can not be defined a priori!

…our guidelines…

1) The best choice depends on the simulation at hand, *e.g.* RANS, DNS

2) As numerical methods are more and more related to the hardware also basis evaluation has to deal with the available hardware

3) High-order meshes need a lot of QPs, the full storage of shapes and their derivatives can become comparable with the size of the implicit operator!

4) To pre-compute orthonormalization coefficients and runtime compute the basis at QPs is an appealing compromise for p-adpatation strategies

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

DG for diffusive problems

DG for purely diffusive problems

the BRx schemes [Bassi and Rebay, 1997]

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Let us consider the heat equation

$$\frac{\partial u}{\partial t} = \nabla^2 u$$

equipped with suitable initial and boundary (Dirichlet, Neumann) data

$$u(\mathbf{x},0) = u_0(\mathbf{x}) \quad u(\mathbf{x},t) = \alpha(\mathbf{x},t)|_{\partial\Omega_D} \quad \mathbf{n}\cdot u|_{\partial\Omega_N} = \beta(\mathbf{x},t)$$

to DG discretize this equation we re-write the equation as a first order system

$$\begin{cases} \boldsymbol{\theta} = \boldsymbol{\nabla} u, & \qquad u(\mathbf{x},t)|_{\partial\Omega_{h,D}} = \alpha(\mathbf{x},t) \\ \dfrac{\partial u}{\partial t} = \boldsymbol{\nabla}\cdot\boldsymbol{\theta} & \qquad \boldsymbol{\theta}\cdot\boldsymbol{n}|_{\partial\Omega_{h,N}} = \beta(\mathbf{x},t) \end{cases}$$

DG for purely diffusive problems

the BRx schemes [Bassi and Rebay, 1997]

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We consider the weak formulation of the system (integration by parts)

$$\int_{\Omega_h} \boldsymbol{\phi} \cdot \boldsymbol{\theta} \, \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \cdot \boldsymbol{\phi}) u \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega_h} (\boldsymbol{\phi} \cdot \boldsymbol{n}) \widehat{u} \, \mathrm{d}\sigma$$

$$\int_{\Omega_h} \phi \frac{\partial u}{\partial t} \, \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \phi) \cdot \boldsymbol{\theta} \, \mathrm{d}\mathbf{x} + \int_{\partial\Omega_h} (\phi \boldsymbol{n}) \cdot \widehat{\boldsymbol{\theta}} \, \mathrm{d}\sigma$$

with weakly imposed BCs

$$\widehat{u} \equiv \begin{cases} \alpha & \text{if } \ \partial\Omega_h \in \partial\Omega_{h,D} \\ u & \text{if } \ \partial\Omega_h \in \partial\Omega_{h,N} \end{cases} \qquad \widehat{\boldsymbol{\theta}} \equiv \begin{cases} \boldsymbol{\theta} & \text{if } \ \partial\Omega_h \in \partial\Omega_{h,D} \\ \beta \boldsymbol{n} & \text{if } \ \partial\Omega_h \in \partial\Omega_{h,N} \end{cases}$$

DG for purely diffusive problems

the BRx schemes *[Bassi and Rebay, 1997]*

UNIVERSITÀ DEGLI STUDI DI BERGAMO

we approximate solution and test functions (scalar and vector) with

$$u_h, \phi_h \in \mathbb{P}^k \qquad\qquad \boldsymbol{\theta}_h, \boldsymbol{\phi}_h \in \mathbb{P}^k_d$$

$$\mathbb{P}^k_d(\mathcal{T}_h) \stackrel{\text{def}}{=} \left\{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}^k_d(T),\, \forall T \in \mathcal{T}_h\right\}$$

we obtain the weak formulation for an element $T \in \mathcal{T}_h$

$$\int_T \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = -\int_T (\boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h) u_h \, \mathrm{d}\mathbf{x} + \int_{\partial T} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) \widehat{u_h} \, \mathrm{d}\sigma$$

$$\int_T \phi_h \frac{\partial u_h}{\partial t} \, \mathrm{d}\mathbf{x} = -\int_T (\boldsymbol{\nabla} \phi_h) \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} + \int_{\partial T} (\phi_h \boldsymbol{n}) \cdot \widehat{\boldsymbol{\theta}_h} \, \mathrm{d}\sigma$$

where $\widehat{u_h}, \widehat{\boldsymbol{\theta}_h}$ are functions of the left and right state at elements interfaces

DG for purely diffusive problems

the BRx schemes [Bassi and Rebay, 1997]

Summation of the elmental contribution

$$
\begin{aligned}
\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = & - \int_{\Omega_h} (\boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h) u_h \, \mathrm{d}\mathbf{x} \\
& + \int_{\mathcal{F}_h^i} \left[(\boldsymbol{\phi}_h \cdot \boldsymbol{n})^- + (\boldsymbol{\phi}_h \cdot \boldsymbol{n})^+ \right] \widehat{u_h} \, \mathrm{d}\sigma \\
& + \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) \widehat{u} \, \mathrm{d}\sigma
\end{aligned}
$$

$$
\begin{aligned}
\int_{\Omega_h} \phi_h \frac{\partial u_h}{\partial t} \, \mathrm{d}\mathbf{x} = & - \int_{\Omega_h} (\boldsymbol{\nabla} \phi_h) \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} \\
& + \int_{\mathcal{F}_h^i} \left[(\phi_h \boldsymbol{n})^- + (\phi_h \boldsymbol{n})^+ \right] \cdot \widehat{\boldsymbol{\theta}_h} \, \mathrm{d}\sigma \\
& + \int_{\mathcal{F}_h^b} (\phi_h \boldsymbol{n}) \cdot \widehat{\boldsymbol{\theta}} \, \mathrm{d}\sigma
\end{aligned}
$$

where $\mathcal{F}_h \equiv \mathcal{F}_h^i \cup \mathcal{F}_h^b$ is the set of the mesh faces F partitioned in internal faces $\mathcal{F}_h^i$ and boundary faces $\mathcal{F}_h^b$

DG for purely diffusive problems

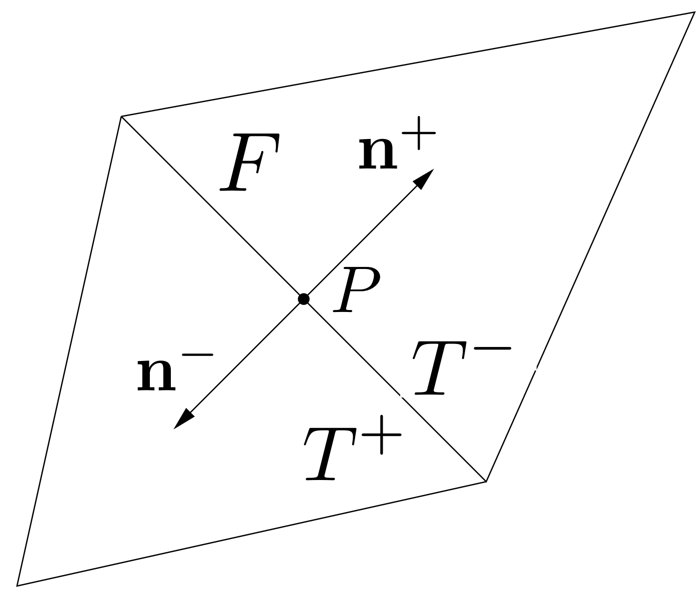
some useful trace operators

We introduce the following trace operators

The jump

$$[\![\phi_h]\!] \equiv (\phi_h \boldsymbol{n})^- + (\phi_h \boldsymbol{n})^+$$

$$[\![\boldsymbol{\phi}_h]\!] \equiv (\boldsymbol{\phi}_h \cdot \boldsymbol{n})^- + (\boldsymbol{\phi}_h \cdot \boldsymbol{n})^+$$



The average

$$\{\phi_h\} \equiv \frac{1}{2}\left(\phi_h^- + \phi_h^+\right)$$

$$\{\boldsymbol{\phi}_h\} \equiv \frac{1}{2}\left(\boldsymbol{\phi}_h^- + \boldsymbol{\phi}_h^+\right)$$

These definitions can be extended to boundary faces by replacing the value on the exterior element by the boundary data

DG for purely diffusive problems

the BRx schemes [Bassi and Rebay, 1997]

We introduce the jump operator

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h) u_h \, \mathrm{d}\mathbf{x}$$

$$+ \int_{\mathcal{F}_h^i} [\![\boldsymbol{\phi}_h]\!] \, \widehat{u_h} \, \mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) \widehat{u} \, \mathrm{d}\sigma$$

$$\int_{\Omega_h} \phi_h \frac{\partial u_h}{\partial t} \, \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \phi_h) \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x}$$

$$+ \int_{\mathcal{F}_h^i} [\![\phi_h]\!] \cdot \widehat{\boldsymbol{\theta}_h} \, \mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\phi_h \boldsymbol{n}) \cdot \widehat{\boldsymbol{\theta}} \, \mathrm{d}\sigma$$

Bassi F, Rebay S, Mariotti G, Pedinotti S, Savini M. Ahigh-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows. In: Decuypere R, Dibelius G, editors. 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, Technologisch Instituut, Antwerpen, Belgium, 1997. pp. 99–108

DG for purely diffusive problems

the BR1 scheme

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Flux functions $\widehat{u_h}, \widehat{\boldsymbol{\theta}_h}$ on boundary integrals are not uniquely defined

we thus introduce the centered numerical fluxes

$$\widehat{u_h} = \{u_h\} \equiv \frac{1}{2}\left(u_h^- + u_h^+\right) \qquad \widehat{\boldsymbol{\theta}_h} = \{\boldsymbol{\theta}_h\} \equiv \frac{1}{2}\left(\boldsymbol{\theta}_h^- + \boldsymbol{\theta}_h^+\right)$$

and obtain the BR1 scheme

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h) u_h \, \mathrm{d}\mathbf{x} + \int_{\mathcal{F}_h^i} [\![\boldsymbol{\phi}_h]\!] \{u_h\} \, \mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) \widehat{u} \, \mathrm{d}\sigma$$

1st equation

$$\int_{\Omega_h} \phi_h \frac{\partial u_h}{\partial t} \, \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \phi_h) \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} + \int_{\mathcal{F}_h^i} [\![\phi_h]\!] \cdot \{\boldsymbol{\theta}_h\} \, \mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\phi_h \boldsymbol{n}) \cdot \widehat{\boldsymbol{\theta}} \, \mathrm{d}\sigma$$

2nd equation

DG for purely diffusive problems

towards the lifting operators definition

UNIVERSITÀ DEGLI STUDI DI BERGAMO

By considering the following algebraic trace relation

$$(u_h \boldsymbol{\phi}_h \cdot \boldsymbol{n})^- + (u_h \boldsymbol{\phi}_h \cdot \boldsymbol{n})^+ = [\![\boldsymbol{\phi}_h]\!] \{u_h\} + \{\boldsymbol{\phi}_h\} \cdot [\![u_h]\!]$$

we obtain the identity

$$\boxed{\begin{aligned}\int_{\Omega_h} \boldsymbol{\nabla} \cdot (u_h \boldsymbol{\phi}_h) \ \mathrm{d}\mathbf{x} = \int_{\Omega_h} (u_h \boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h + \boldsymbol{\phi}_h \cdot \boldsymbol{\nabla} u_h) \ \mathrm{d}\mathbf{x} \equiv \\ \int_{\mathcal{F}_h^i} ([\![\boldsymbol{\phi}_h]\!] \{u_h\} + \{\boldsymbol{\phi}_h\} \cdot [\![u_h]\!]) \ \mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) u_h \ \mathrm{d}\sigma\end{aligned}}$$

consequence of the divergence theorem

$$- \int_{\Omega_h} u_h \boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h \ \mathrm{d}\mathbf{x} = \int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\nabla} u_h \ \mathrm{d}\mathbf{x}$$

$$- \int_{\mathcal{F}_h^i} (u_h \boldsymbol{\phi}_h \cdot \boldsymbol{n})^+ + (u_h \boldsymbol{\phi}_h \cdot \boldsymbol{n})^- \ \mathrm{d}\sigma - \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) u_h \ \mathrm{d}\sigma =$$

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\nabla} u_h \ \mathrm{d}\mathbf{x} - \int_{\mathcal{F}_h^i} ([\![\boldsymbol{\phi}_h]\!] \{u_h\} + \{\boldsymbol{\phi}_h\} \cdot [\![u_h]\!]) \ \mathrm{d}\sigma - \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) u_h \ \mathrm{d}\sigma$$

DG for purely diffusive problems

towards the lifting operators definition

We reformulate the 1st equation of the BR1 scheme

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \,\mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla} \cdot \boldsymbol{\phi}_h) u_h \,\mathrm{d}\mathbf{x} + \int_{\mathcal{F}_h^i} [\![\boldsymbol{\phi}_h]\!] \{u_h\} \,\mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) \widehat{u} \,\mathrm{d}\sigma$$

as

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \,\mathrm{d}\mathbf{x} = \int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\nabla} u_h \,\mathrm{d}\mathbf{x}$$

$$-\int_{\mathcal{F}_h^i} \left([\![\boldsymbol{\phi}_h]\!] \{u_h\} + \{\boldsymbol{\phi}_h\} \cdot [\![u_h]\!]\right) \mathrm{d}\sigma - \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) u_h \,\mathrm{d}\sigma$$

$$+ \int_{\mathcal{F}_h^i} [\![\boldsymbol{\phi}_h]\!] \{u_h\} \,\mathrm{d}\sigma + \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n}) \widehat{u} \,\mathrm{d}\sigma$$

DG for purely diffusive problems

towards the lifting operators definition

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = \int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\nabla} u_h \, \mathrm{d}\mathbf{x} - \int_{\mathcal{F}_h^i} \{\boldsymbol{\phi}_h\} \cdot [\![u_h]\!] \, \mathrm{d}\sigma - \int_{\mathcal{F}_h^b} (\boldsymbol{\phi}_h \cdot \boldsymbol{n})(u_h - u_b) \, \mathrm{d}\sigma$$

The jump operator opportunely take into account of BC datum

$$[\![u_h]\!] = \begin{cases} [\![u_h]\!] & \text{on} \quad \mathcal{F}_h^i \\ (u_h - u_b)\boldsymbol{n} & \text{on} \quad \mathcal{F}_h^b \end{cases}$$

as first equation of the BR1 we obtain

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = \int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\nabla} u_h \, \mathrm{d}\mathbf{x} - \int_{\mathcal{F}_h} \{\boldsymbol{\phi}_h\} \cdot [\![u_h]\!] \, \mathrm{d}\sigma$$

DG for purely diffusive problems

the global lifting operator

We define the lifting operator as

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{r}\left(\boldsymbol{v}_h\right) \, \mathrm{d}\mathbf{x} \equiv - \int_{\mathcal{F}_h} \{\boldsymbol{\phi}_h\} \cdot \boldsymbol{v}_h \, \mathrm{d}\sigma$$

it can be applied to any function $\boldsymbol{v}_h \in [L^2(F)]^d$ with $F \in \mathcal{F}_h \equiv \mathcal{F}_h^i \cup \mathcal{F}_h^b$ including jumps

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{r}\left(\llbracket u_h \rrbracket\right) \, \mathrm{d}\mathbf{x} = - \int_{\mathcal{F}_h} \{\boldsymbol{\phi}_h\} \cdot \llbracket u_h \rrbracket \, \mathrm{d}\sigma$$

thus obtaining as 1st equation of the BR1

$$\int_{\Omega_h} \boldsymbol{\phi}_h \cdot \boldsymbol{\theta}_h \, \mathrm{d}\mathbf{x} = \int_{\Omega_h} \boldsymbol{\phi}_h \cdot \left(\boldsymbol{\nabla} u_h + \boldsymbol{r}\left(\llbracket u_h \rrbracket\right)\right) \, \mathrm{d}\mathbf{x}$$

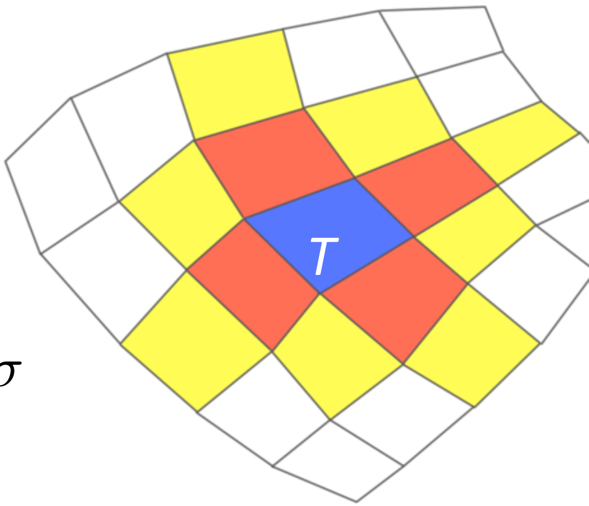
we obtained a “modified gradient” $\boldsymbol{\theta}_h = \boldsymbol{\nabla} u_h + \boldsymbol{r}\left(\llbracket u_h \rrbracket\right)$ that properly take into account for the effect of interface jumps on $\boldsymbol{\nabla} u_h$

DG for purely diffusive problems

the BR1 scheme

Second equation of BR1

$$\int_{\Omega_h} \phi_h \frac{\partial u_h}{\partial t}\ \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\nabla \phi_h)\cdot(\nabla u_h + \boldsymbol{r}(\llbracket u_h \rrbracket))\ \mathrm{d}\mathbf{x}$$
$$+\int_{\mathcal{F}_h^i} \llbracket \phi_h \rrbracket \cdot \{\nabla u_h + \boldsymbol{r}(\llbracket u_h \rrbracket)\}\ \mathrm{d}\sigma$$
$$+\int_{\mathcal{F}_h^b} (\phi_h \boldsymbol{n})\cdot\left(\widehat{\nabla u_h} + \boldsymbol{r}(\llbracket u_h \rrbracket)\right)\ \mathrm{d}\sigma$$



where $\widehat{\nabla u_h}$ is a proper treatment of gradient at BC

BR1

the scheme issues

1) Large spatial support, the neighbors of *T* and neighbors of the neighbors

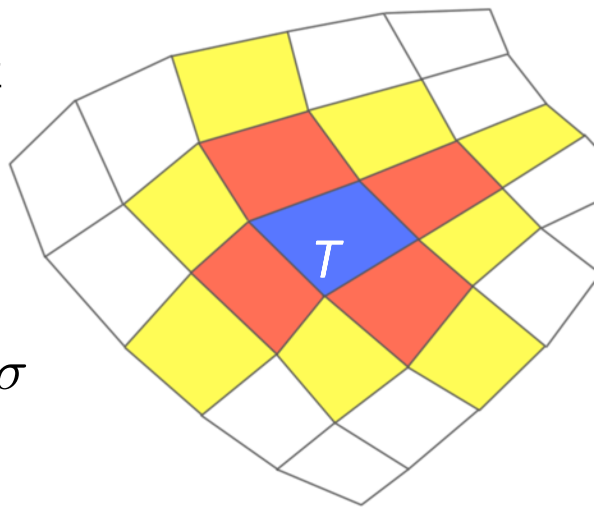
2) Suboptimal order of accuracy for odd degree polynomial approximations

DG for purely diffusive problems

the BR1 scheme

Second equation of BR1

$$\int_{\Omega_h} \phi_h \frac{\partial u_h}{\partial t}\ \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla}\phi_h)\cdot(\boldsymbol{\nabla}u_h + \boldsymbol{r}([\![u_h]\!]))\ \mathrm{d}\mathbf{x}$$
$$+\int_{\mathcal{F}_h^i} [\![\phi_h]\!]\cdot\{\boldsymbol{\nabla}u_h + \boldsymbol{r}([\![u_h]\!])\}\ \mathrm{d}\sigma$$
$$+\int_{\mathcal{F}_h^b} (\phi_h \boldsymbol{n})\cdot\left(\widehat{\boldsymbol{\nabla}u_h} + \boldsymbol{r}([\![u_h]\!])\right)\ \mathrm{d}\sigma$$



where $\widehat{\boldsymbol{\nabla}u_h}$ is a proper treatment of gradient at BC

BR1

the scheme issues

1) Large spatial support, the neighbors of *T* and neighbors of the neighbors

2) Suboptimal order of accuracy for odd degree polynomial approximations

...from BR1 to BR2...

DG for purely diffusive problems

from the BR1 to the BR2: the local lifting operator

UNIVERSITÀ DEGLI STUDI DI BERGAMO

The large stencil is due to the $\boldsymbol{r}\left(\llbracket u_h \rrbracket\right)$ integral on internal faces

We introduce the local lifting operator

$$\int_{\Omega_h} \phi_h \cdot \boldsymbol{r}_f\left(\boldsymbol{v}_h\right) \, \mathrm{d}\mathbf{x} \equiv -\int_F \{\phi_h\} \cdot \boldsymbol{v}_h \, \mathrm{d}\sigma$$

the local lifting operator is nonzero at the elements that share F only and is related to the global lifting operator as

$$\boldsymbol{r}\left(\boldsymbol{v}_h\right) = \sum_{F \in \mathcal{F}} \boldsymbol{r}_f\left(\boldsymbol{v}_h\right)$$

BR2

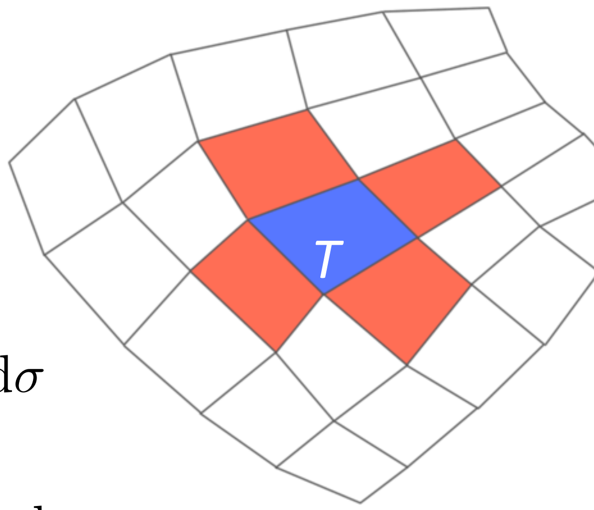
To recover the correct order of accuracy proper of DG and to obtain a stencil as compact as possible we replaced $\boldsymbol{r}\left(\llbracket u_h \rrbracket\right)$ with $\eta_f \boldsymbol{r}_f\left(\llbracket u_h \rrbracket\right)$ in surface integrals

DG for purely diffusive problems

the BR2 scheme

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

Second equation of BR2



$$\int_{\Omega_h} \phi_h \frac{\partial u_h}{\partial t}\ \mathrm{d}\mathbf{x} = -\int_{\Omega_h} (\boldsymbol{\nabla}\phi_h)\cdot\left(\boldsymbol{\nabla}u_h + \boldsymbol{r}\left([\![u_h]\!]\right)\right)\ \mathrm{d}\mathbf{x}$$
$$+\int_{\mathcal{F}_h^i} [\![\phi_h]\!]\cdot\left\{\boldsymbol{\nabla}u_h + \eta_f \boldsymbol{r}_f\left([\![u_h]\!]\right)\right\}\ \mathrm{d}\sigma$$
$$+\int_{\mathcal{F}_h^b} (\phi_h \boldsymbol{n})\cdot\left(\widehat{\boldsymbol{\nabla}u_h} + \eta_f \boldsymbol{r}_f\left([\![u_h]\!]\right)\right)\ \mathrm{d}\sigma$$

where, according to [Arnold et al., 2002], the penalty factor η_f must be greater than the number of faces of the elements

DG for purely diffusive problems

the BR2 scheme for any shape of element

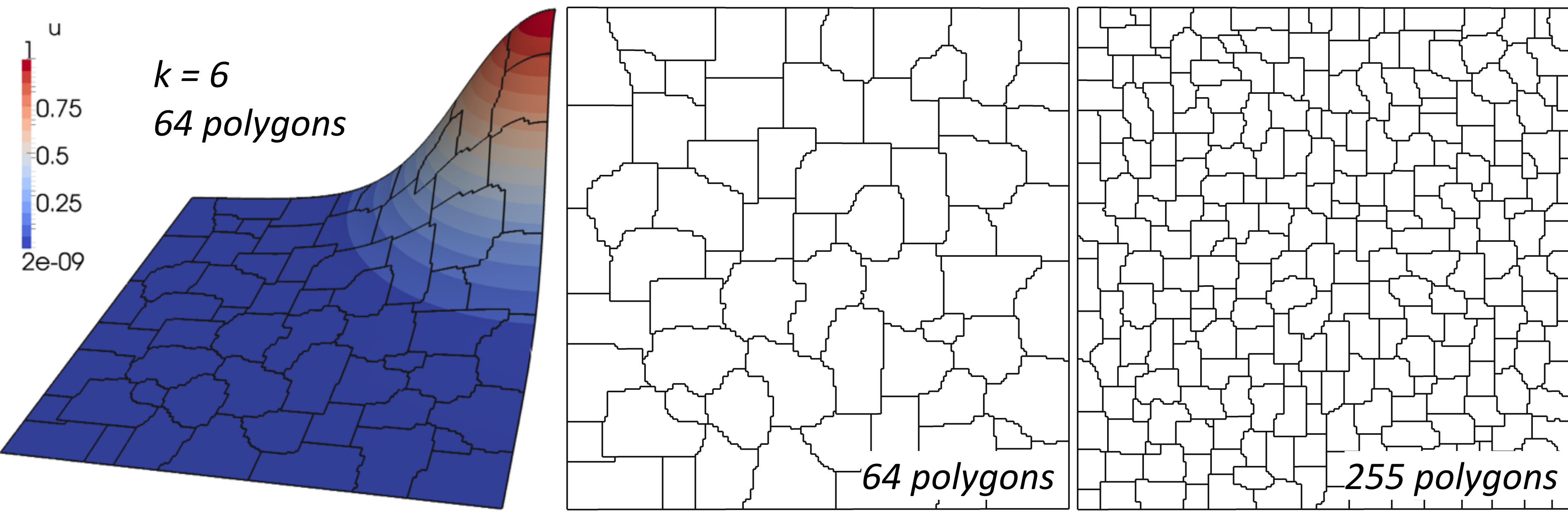
UNIVERSITÀ DEGLI STUDI DI BERGAMO

Test on the exact solution of a Poisson problem proposed in [Karniadakis and Sherwin, 2005]

$$u = e^{-2.5[(x-1)^2+(y-1)^2]} \qquad \Omega = [-1,1]^2$$

<u>mesh sequences</u>

- 64, 256, 1028, 4096 uniform quadrilaterals grids
- 64, 255, 1028, 4122 polygonal elements grids built on top of a 200x200 quadrilaterals grid using MGridGen[1]

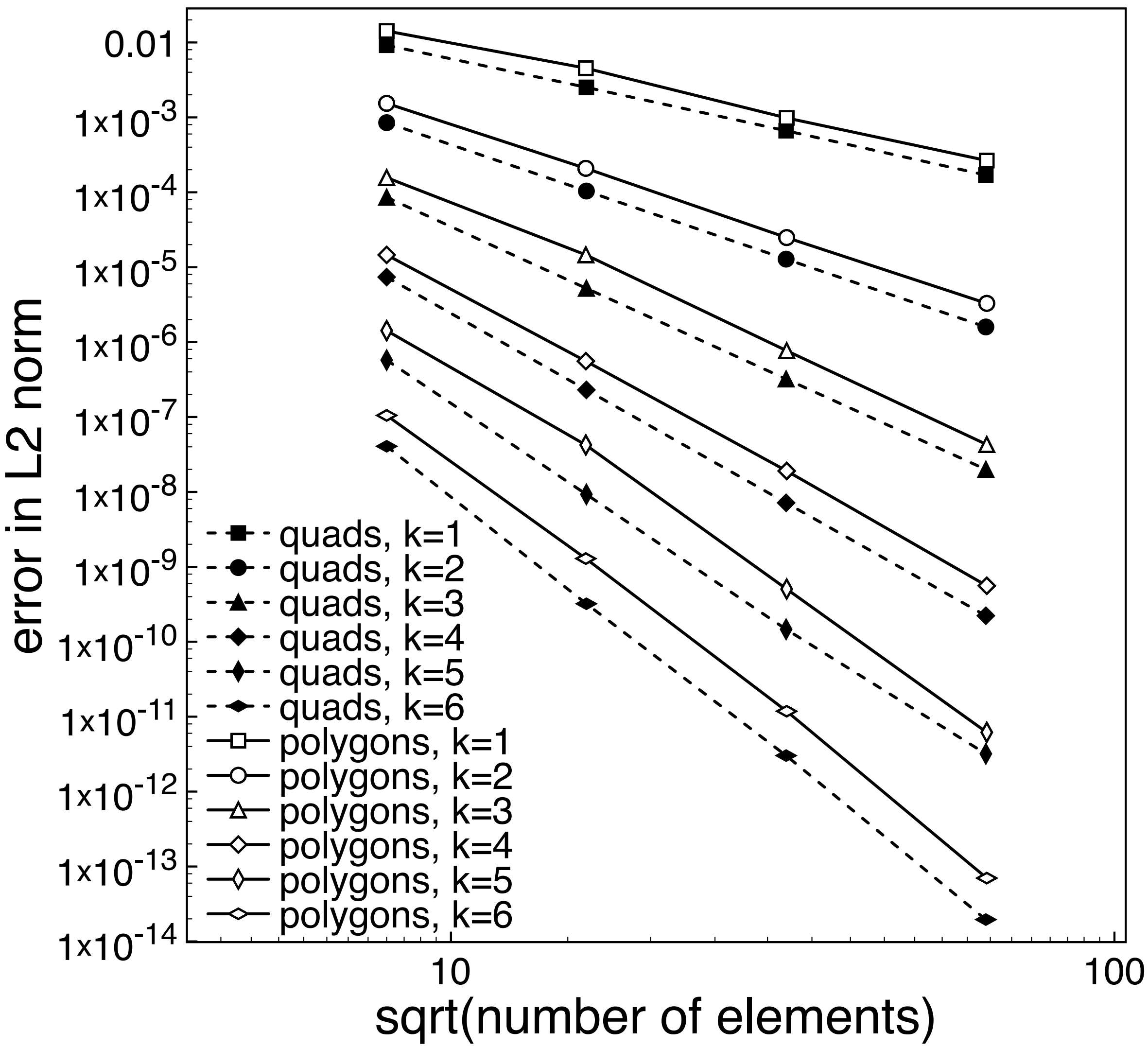


1) *http://www-users.cs.umn.edu/~moulitsa/software.html*

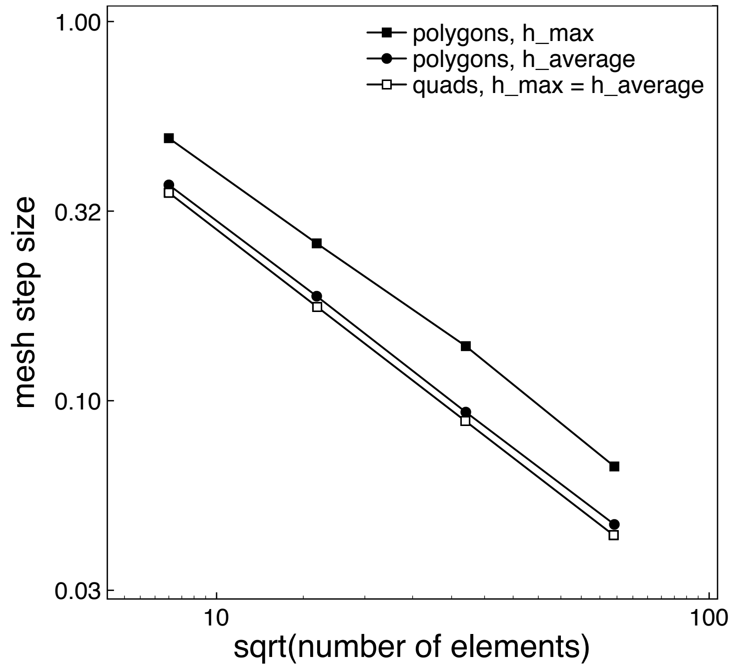
DG for purely diffusive problems

the BR2 scheme for any shape of element

UNIVERSITÀ DEGLI STUDI
DI BERGAMO



The DG convergence rate (h^{k+1}) on both uniform quadrilateral and polygonal elements meshes is attained



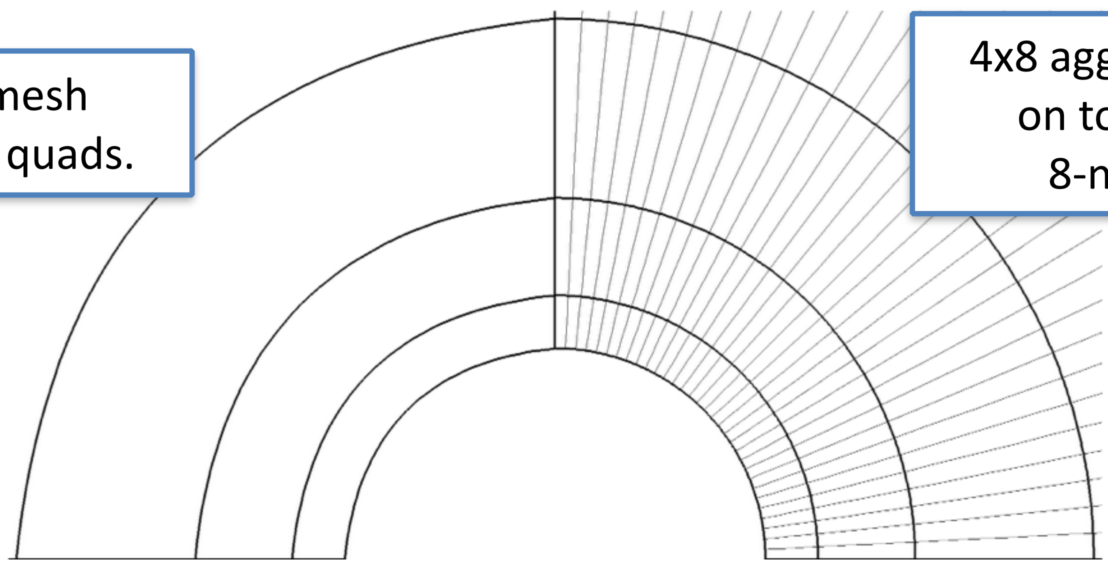
Why so interested in agglomerated elements?

An alternative approach to accurate boundary approximation

UNIVERSITÀ DEGLI STUDI DI BERGAMO

When the error of the geometrical representation overwhelms the approximation error the convergence proprieties are compromised

Once a underlying fine grid is available, the agglomerated mesh density can be tuned to fit the hardware while keeping the fine grid boundary resolution



4x8 mesh
8-node quads.

4x8 agglomerated grid
on top of a 128x8
8-node quads.

Pros

accurate boundary representation

same number of DOFs

relies on standard meshes

Cons

lack of efficient quadrature rules

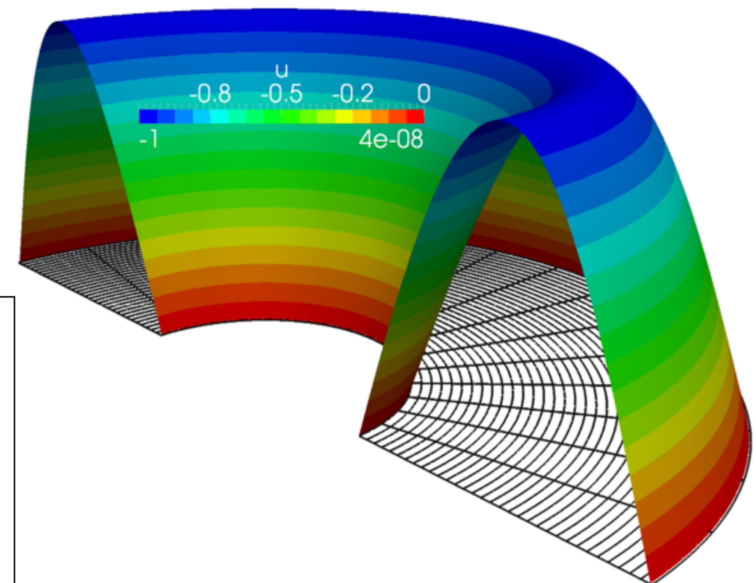
assembly overcomes solution cost

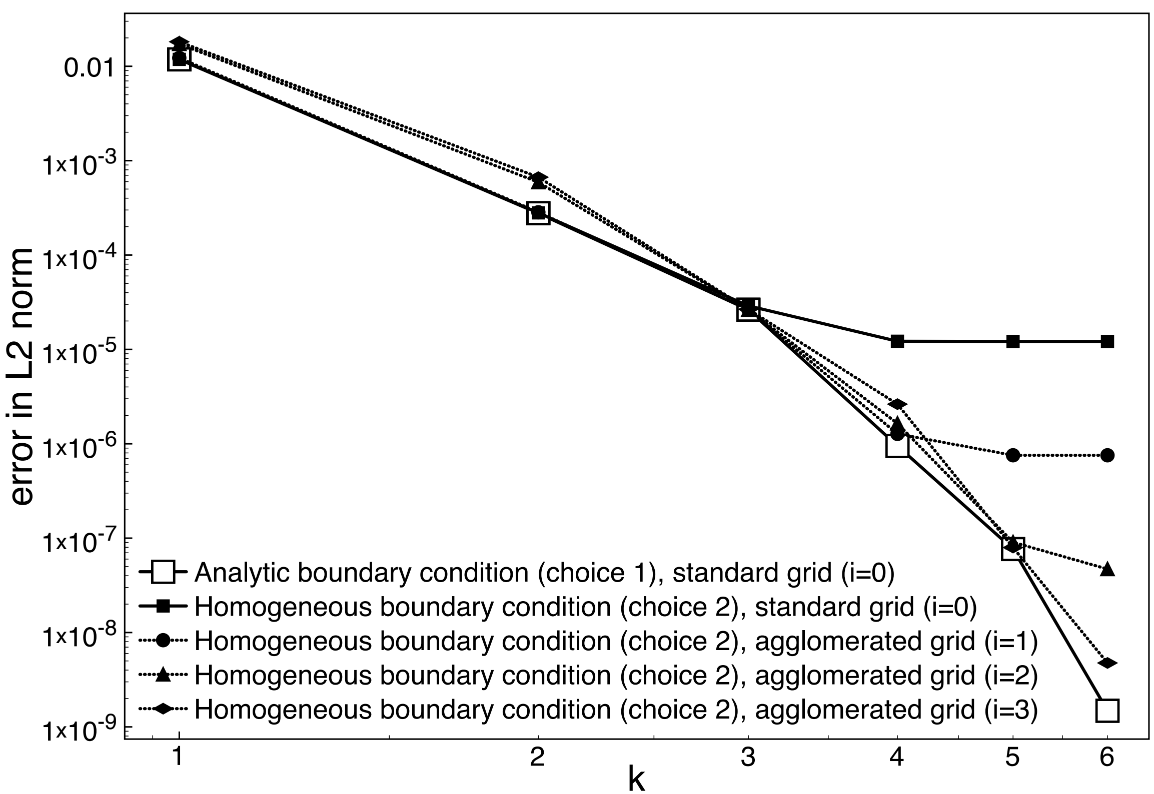
Why so interested in agglomerated elements?

An alternative approach to accurate boundary approximation

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

We test against the exact solution of Poisson problem proposed in [Gobbert and Yang, 2008]





<u>meshes sequences</u>

- Standard 32x32 8-node quads. grid
- Agglomerated 32x32 grids built on top of $(32 \cdot 2^i)$x32, i={1, 2, 3} 8-node quads. grids

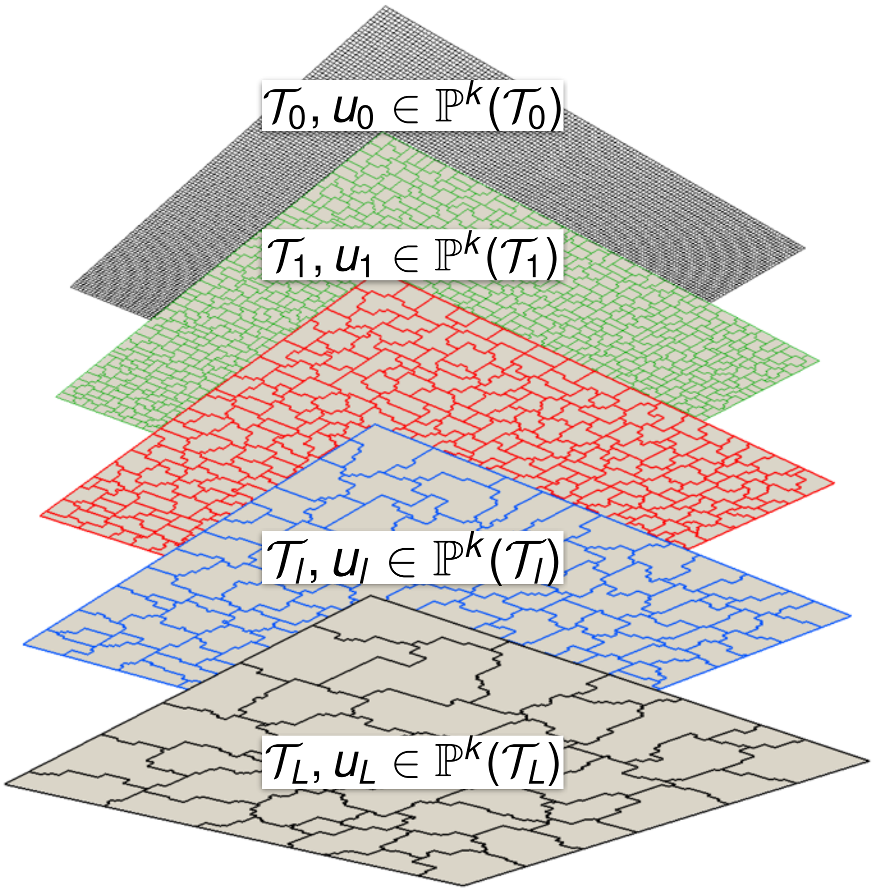
Why so interested in agglomerated elements?

Improve the solver efficiency: the h-multigrid algorithm

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

h-multigrid is an iterative solver strategy

we want to solve linear (or linearized) equation systems $Au = f$ arising from DG discretizations



***h-MG* in a nutshell (*I/II*)**

to efficiently solve Au = f several coarse problems $A_l \,\Delta u_l = r_l$ are solved

coarse problems are explicitly built on a sequence of *h*-coarsened grids

h-coarsened grids are computed by recursive agglomeration of the fine grid *[MGridGen, Moulitsas Karypis, 2001]*

agglomeration yields nested grids of arbitrarily shaped elements

to use h-MG we need discrete DG spaces over agglomerated elements meshes

$$\mathbb{P}^k(\mathcal{T}_l) = \{v \in L^2(\Omega) : v|_\kappa \in \mathbb{P}^k(\kappa), \forall \kappa \in \mathcal{T}_l\}$$

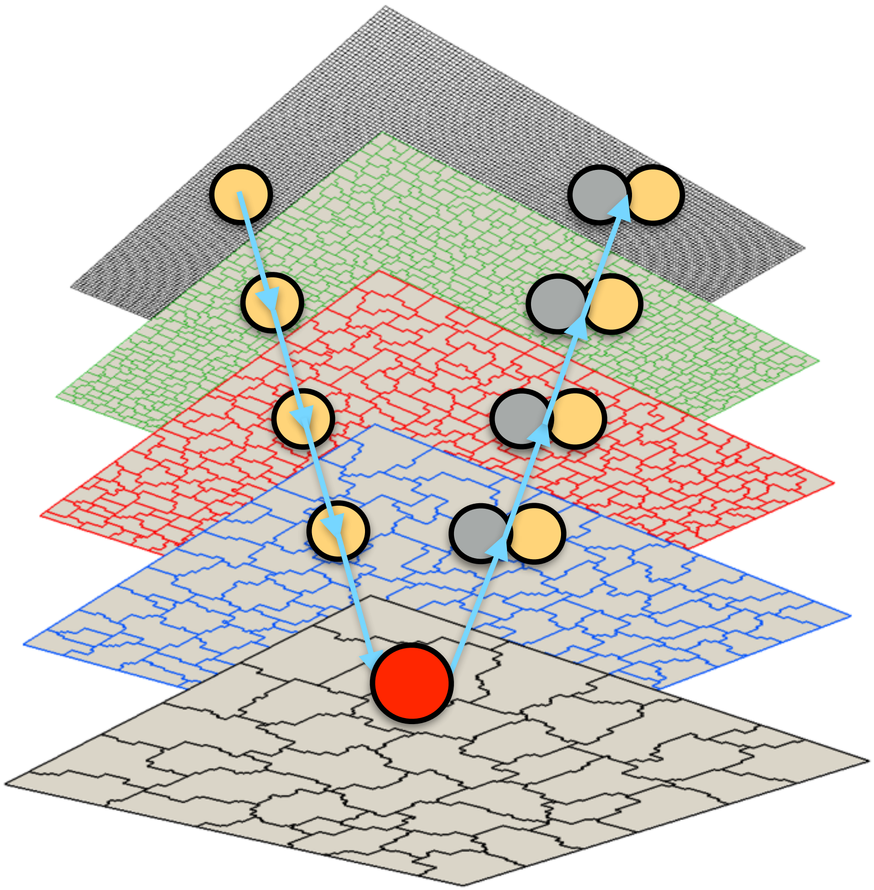
Why so interested in agglomerated elements?

Improve the solver efficiency: the h-multigrid algorithm

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

how the solver walks across
the grid sequence?

the V-cycle



h-MG in a nutshell (*II/II*)

Smoothers damp high-frequency modes of the solution error

Iterative solvers are effective smoothers

Low-frequency modes of the error appears more oscillatory on coarser grids

To accelerate convergence we exploit smoothers on a coarser grid

Intergrid transfer operators map informations between grids

prolongation $l \rightarrow l\text{-}1$

restriction $l \rightarrow l\text{+}1$

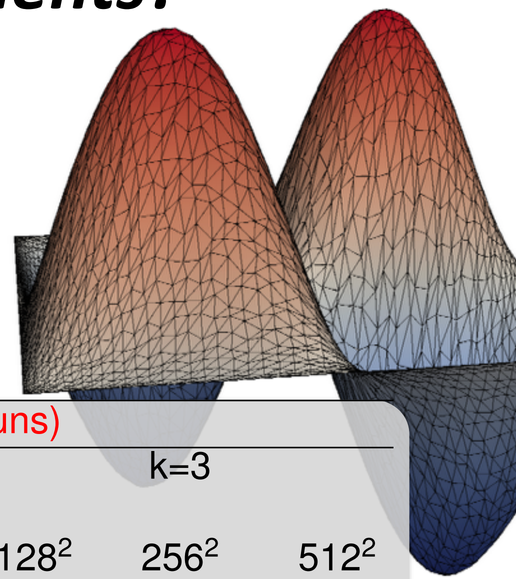
Why so interested in agglomerated elements?

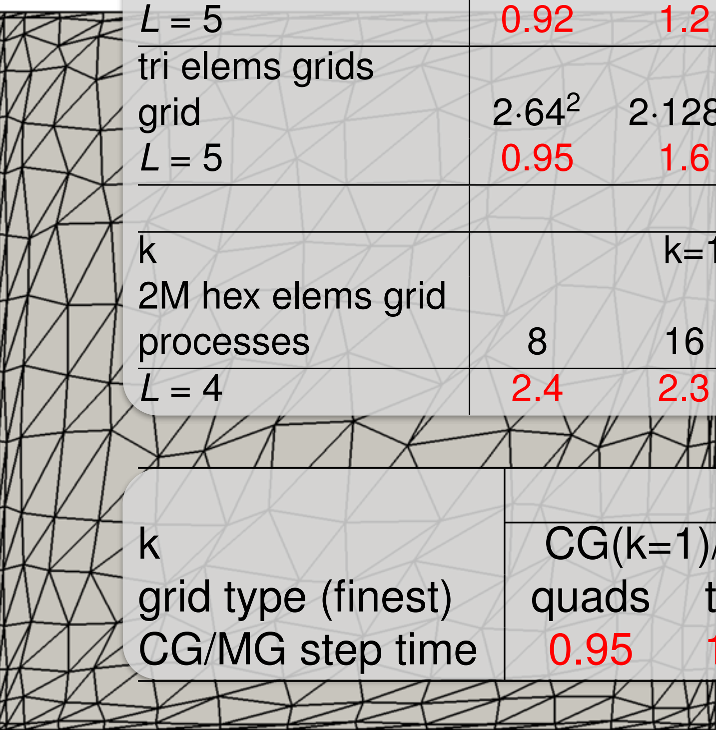
Improve the solver efficiency: the h-multigrid algorithm

$$\begin{cases} -\Delta u = f & \text{in} \quad \Omega \\ u = 0 & \text{on} \quad \partial\Omega \end{cases}$$

$$u = \prod_{i=1}^{d} sin(\pi x_i)$$

$$\Omega = [-1, 1]^d$$



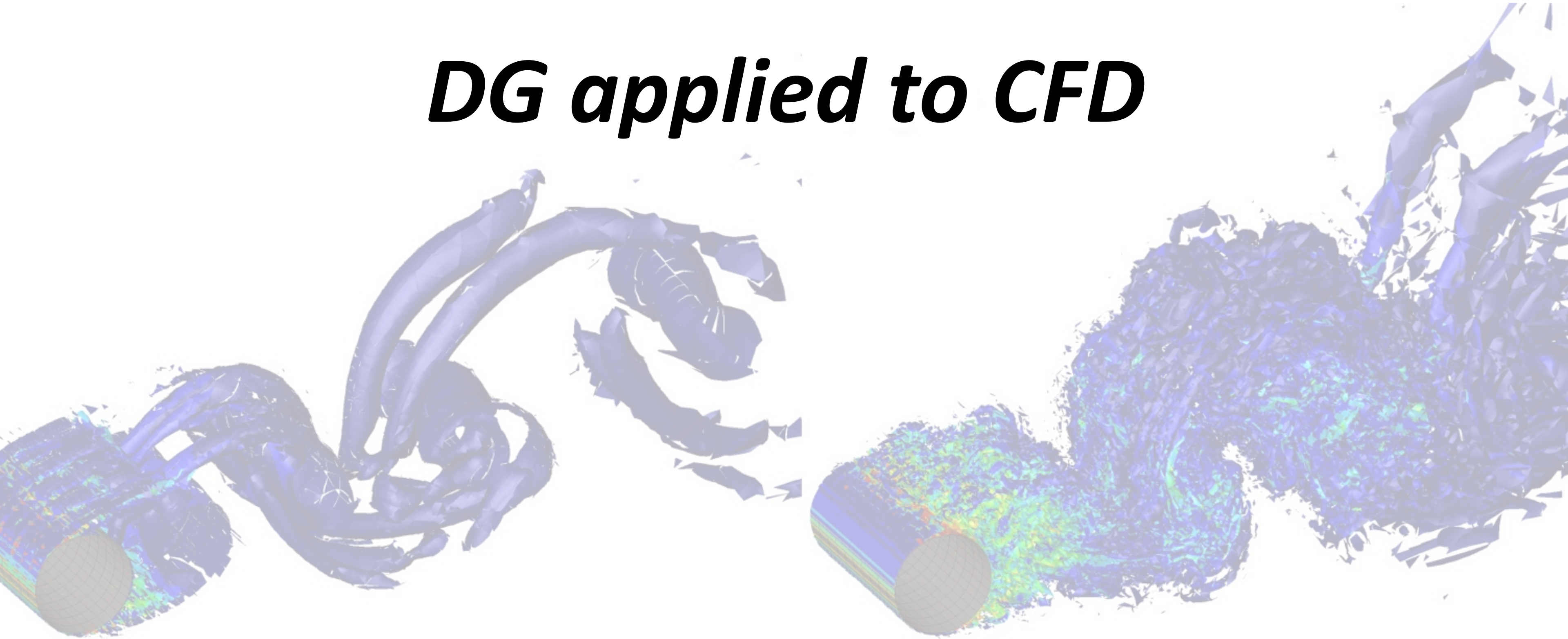


cpu time (CG/MG)	total step time speedup (2D serial runs)								
k	k=1			k=2			k=3		
quad elems grids									
grid	128^2	256^2	512^2	128^2	256^2	512^2	128^2	256^2	512^2
L = 5	0.92	1.2	1.8	1.5	2.3	3.4	2.0	3.1	5.4
tri elems grids									
grid	$2{\cdot}64^2$	$2{\cdot}128^2$	$2{\cdot}256^2$	$2{\cdot}64^2$	$2{\cdot}128^2$	$2{\cdot}256^2$	$2{\cdot}64^2$	$2{\cdot}128^2$	$2{\cdot}256^2$
L = 5	0.95	1.6	2.4	1.4	2.3	3.5	1.7	3.1	5.0
	total step time speedup (3D parallel runs)								
k	k=1			k=2			k=3		
2M hex elems grid									
processes	8	16	32	16	32	64	32	64	128
L = 4	2.4	2.3	2.3	3.6	3.4	3.2	3.6	4.3	4.7

	step time ratio								
k	CG(k=1)/MG(k=2)			CG(k=2)/MG(k=3)			CG(k=1)/MG(k=3)		
grid type (finest)	quads	tris	hexs	quads	tris	hexs	quads	tris	hexs
CG/MG step time	0.95	1.0	1.1	1.6	1.3	1.6	0.55	0.37	0.53

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

DG applied to CFD



DG applied to CFD

The inviscid flows governing equations - Euler

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} + \frac{\partial \widehat{\tau}_{ji}}{\partial x_j}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j H) = \frac{\partial}{\partial x_j}\left(u_i \tau_{ij} - q_j + u_i \widehat{\tau_{ij}} - \widehat{q}_j\right) - P_k + D_k$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \overline{\mu}_t)\frac{\partial k}{\partial x_j}\right] + P_k - D_k$$

$$\frac{\partial}{\partial t}(\rho \widetilde{\omega}) + \frac{\partial}{\partial x_j}(\rho u_j \widetilde{\omega}) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_j}\right] + (\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_k}\frac{\partial \widetilde{\omega}}{\partial x_k} + P_\omega - D_\omega + C_D$$

DG applied to CFD

The viscous flows governing equations - Navier—Stokes

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} + \frac{\partial \widehat{\tau}_{ji}}{\partial x_j}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j H) = \frac{\partial}{\partial x_j}\left(u_i \tau_{ij} - q_j + u_i \widehat{\tau_{ij}} - \widehat{q}_j\right) - P_k + D_k$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \overline{\mu}_t)\frac{\partial k}{\partial x_j}\right] + P_k - D_k$$

$$\frac{\partial}{\partial t}(\rho \widetilde{\omega}) + \frac{\partial}{\partial x_j}(\rho u_j \widetilde{\omega}) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_j}\right] + (\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_k}\frac{\partial \widetilde{\omega}}{\partial x_k} + P_\omega - D_\omega + C_D$$

DG applied to CFD

Turbulent flows governing equations (models)

Reynolds-averaged Navier—Stokes+ k-$\widetilde{\omega}$ (EARSM), X-LES

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ji}}{\partial x_j} + \frac{\partial \widehat{\tau}_{ji}}{\partial x_j}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j H) = \frac{\partial}{\partial x_j}\left(u_i \tau_{ij} - q_j + u_i \widehat{\tau_{ij}} - \widehat{q}_j\right) - P_k + D_k$$

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma^* \overline{\mu}_t)\frac{\partial k}{\partial x_j}\right] + P_k - D_k$$

$$\frac{\partial}{\partial t}(\rho \widetilde{\omega}) + \frac{\partial}{\partial x_j}(\rho u_j \widetilde{\omega}) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_j}\right] + (\mu + \sigma \overline{\mu}_t)\frac{\partial \widetilde{\omega}}{\partial x_k}\frac{\partial \widetilde{\omega}}{\partial x_k} + P_\omega - D_\omega + C_D$$

Reynolds averaged Navier-Stokes equations closed with the Wilcox k-ω model

Non standard implementation using $\widetilde{\omega} = \log(\omega)$

Why X-LES?

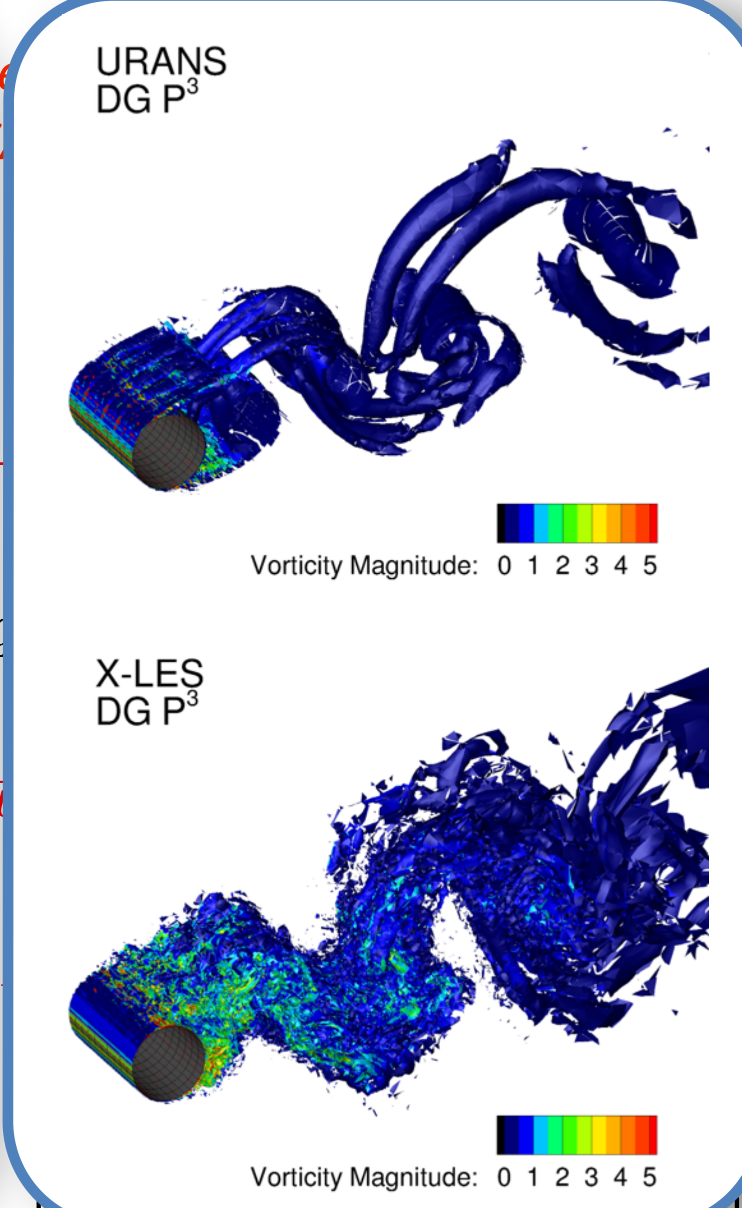
For those high Reynolds number flows where the RANS formulation suffers from prediction limitations, e.g. massively separated flows, but LES is too demanding

Pros

- hybrid RANS\LES formulation independent from the wall distance
- use in LES mode of a clearly defined SGS based on the k-equation
- use of a k-ω turbulence model integrated to the wall

Cons

the *filter width* parameter is often related to the local element size



Non standard implementation using $\widetilde{\omega} = \log(\omega)$

DG applied to CFD

Turbulent flows governing equations (models)
RANS+k-$\widetilde{\omega}$ (EARSM), X-LES

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Heat flux and stress tensor

$$q_j = -\frac{\mu}{\mathrm{Pr}}\frac{\partial h}{\partial x_j} \qquad \widehat{q}_j = -\frac{\overline{\mu}_t}{\mathrm{Pr}_t}\frac{\partial h}{\partial x_j}$$

$$\tau_{ij} = 2\mu\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] \qquad \widehat{\tau}_{ij} = 2\overline{\mu}_t\left[S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right] - \frac{2}{3}\rho\overline{k}\delta_{ij}$$

source terms

$$P_k = \widehat{\tau}_{ij}\frac{\partial u_i}{\partial x_j} \qquad P_\omega = \alpha\left[\alpha^*\frac{\rho}{e^{\widetilde{\omega}_r}}\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho\delta_{ij}\right]\frac{\partial u_i}{\partial x_j}$$

$$D_k = \beta^*\rho\overline{k}\hat{\omega} \qquad D_\omega = \beta\rho\overline{k}e^{\widetilde{\omega}_r} \qquad C_D = \sigma_d\frac{\rho}{e^{\widetilde{\omega}_r}}\max\left(\frac{\partial k}{\partial x_k}\frac{\partial\widetilde{\omega}}{\partial x_k}, 0\right)$$

where

$$\overline{\mu}_t = \alpha^*\frac{\rho\overline{k}}{\hat{\omega}} \qquad \hat{\omega} = \max\left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1\mathit{\Delta}}\right) \qquad \overline{k} = \max(0, k)$$

DG applied to CFD

Impact of X-LES on source terms and turbulent quantities

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$P_k = \widehat{\tau_{ij}} \frac{\partial u_i}{\partial x_j} \qquad P_\omega = \alpha \left[\alpha^* \frac{\rho}{e^{\widetilde{\omega}_r}} \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho \delta_{ij} \right] \frac{\partial u_i}{\partial x_j}$$

$$D_k = \beta^* \rho \overline{k} \hat{\omega} \qquad D_\omega = \beta \rho \overline{k} e^{\widetilde{\omega}_r} \qquad C_D = \sigma_d \frac{\rho}{e^{\widetilde{\omega}_r}} \max \left(\frac{\partial k}{\partial x_k} \frac{\partial \widetilde{\omega}}{\partial x_k}, 0 \right)$$

where

$$\overline{\mu}_t = \alpha^* \frac{\rho \overline{k}}{\hat{\omega}} \qquad \hat{\omega} = \max \left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1 \Delta} \right) \qquad \overline{k} = \max(0, k)$$

our implementation actually includes three models

	RANS	LES	ILES
$\overline{\mu}_t$	$\alpha^* \frac{\rho \overline{k}}{e^{\widetilde{\omega}_r}}$	$\alpha^* \rho \sqrt{\overline{k}} C_1 \Delta$	0
D_k	$\beta^* \rho \overline{k} e^{\widetilde{\omega}_r}$	$\beta^* \rho \frac{\overline{k}^{\frac{3}{2}}}{C_1 \Delta}$	0

DG applied to CFD

Impact of X-LES on source terms and turbulent quantities

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$P_k = \widehat{\tau_{ij}} \frac{\partial u_i}{\partial x_j} \qquad P_\omega = \alpha \left[\alpha^* \frac{\rho}{e^{\widetilde{\omega}_r}} \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho \delta_{ij} \right] \frac{\partial u_i}{\partial x_j}$$

$$D_k = \beta^* \rho \overline{k} \hat{\omega} \qquad D_\omega = \beta \rho \overline{k} e^{\widetilde{\omega}_r} \qquad C_D = \sigma_d \frac{\rho}{e^{\widetilde{\omega}_r}} \max\left(\frac{\partial k}{\partial x_k} \frac{\partial \widetilde{\omega}}{\partial x_k}, 0 \right)$$

where

$$\overline{\mu}_t = \alpha^* \frac{\rho \overline{k}}{\hat{\omega}} \qquad \hat{\omega} = \max\left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1 \Delta} \right) \qquad \overline{k} = \max(0, k)$$

our implementation actually includes three models

	RANS	LES	ILES
$\overline{\mu}_t$	$\alpha^* \frac{\rho \overline{k}}{e^{\widetilde{\omega}_r}}$	$\alpha^* \rho \sqrt{\overline{k}} C_1 \Delta$	0
D_k	$\beta^* \rho \overline{k} e^{\widetilde{\omega}_r}$	$\beta^* \rho \frac{\overline{k}^{\frac{3}{2}}}{C_1 \Delta}$	0

DG applied to CFD

Impact of X-LES on source terms and turbulent quantities

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$P_k = \widehat{\tau_{ij}} \frac{\partial u_i}{\partial x_j} \qquad P_\omega = \alpha \left[\alpha^* \frac{\rho}{e^{\widetilde{\omega}_r}} \left(S_{ij} - \frac{1}{3} \frac{\partial u_k}{\partial x_k} \delta_{ij} \right) - \frac{2}{3} \rho \delta_{ij} \right] \frac{\partial u_i}{\partial x_j}$$

$$D_k = \beta^* \rho \overline{k} \hat{\omega} \qquad D_\omega = \beta \rho \overline{k} e^{\widetilde{\omega}_r} \qquad C_D = \sigma_d \frac{\rho}{e^{\widetilde{\omega}_r}} \max \left(\frac{\partial k}{\partial x_k} \frac{\partial \widetilde{\omega}}{\partial x_k}, 0 \right)$$

where

$$\overline{\mu}_t = \alpha^* \frac{\rho \overline{k}}{\hat{\omega}} \qquad \hat{\omega} = \max \left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1 \Delta} \right) \qquad \overline{k} = \max(0, k)$$

our implementation actually includes three models

	RANS	LES	ILES
$\overline{\mu}_t$	$\alpha^* \frac{\rho \overline{k}}{e^{\widetilde{\omega}_r}}$	$\alpha^* \rho \sqrt{\overline{k}} C_1 \Delta$	0
D_k	$\beta^* \rho \overline{k} e^{\widetilde{\omega}_r}$	$\beta^* \rho \frac{\overline{k}^{\frac{3}{2}}}{C_1 \Delta}$	0

DG applied to CFD

Impact of X-LES on source terms and turbulent quantities

UNIVERSITÀ DEGLI STUDI DI BERGAMO

$$P_k = \tau_{ij}\frac{\partial u_i}{\partial x_j} \qquad P_\omega = \alpha\left[\alpha^*\frac{\rho}{e^{\widetilde{\omega}_r}}\left(S_{ij} - \frac{1}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right) - \frac{2}{3}\rho\delta_{ij}\right]\frac{\partial u_i}{\partial x_j}$$

$$D_k = \beta^*\rho\overline{k}\hat{\omega} \qquad D_\omega = \beta\rho\overline{k}e^{\widetilde{\omega}_r} \qquad C_D = \sigma_d\frac{\rho}{e^{\widetilde{\omega}_r}}\max\left(\frac{\partial k}{\partial x_k}\frac{\partial\widetilde{\omega}}{\partial x_k}, 0\right)$$

where

$$\overline{\mu}_t = \alpha^*\frac{\rho\overline{k}}{\hat{\omega}} \qquad \hat{\omega} = \max\left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1\Delta}\right) \qquad \overline{k} = \max(0, k)$$

our implementation actually includes three models

	RANS	LES	ILES
$\overline{\mu}_t$	$\alpha^*\frac{\rho\overline{k}}{e^{\widetilde{\omega}_r}}$	$\alpha^*\rho\sqrt{\overline{k}}C_1\Delta$	0
D_k	$\beta^*\rho\overline{k}e^{\widetilde{\omega}_r}$	$\beta^*\rho\frac{\overline{k}^{\frac{3}{2}}}{C_1\Delta}$	0

DG discretization of the fluid dynamics equations

The governing equations can be written in compact form as

$$\mathbf{P}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{w}) + \nabla \cdot \mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w}) + \mathbf{s}(\mathbf{w}, \nabla \mathbf{w}) = \mathbf{0}$$

for compressible flows a common choice for **w** is

$$\mathbf{w}_c = [\rho, \rho u_i, \rho E, \rho k, \rho\widetilde{\omega}]^T \rightarrow \mathbf{P}(\mathbf{w}) = \mathbf{I}$$

Alternatives to $\mathbf{w}_c$ have been investigated by several authors in order

- to obtain a well defined behavior of variables in the incompressible limit of compressible flows
- to deal with low Mach number flows (*p, **u**, T*) [Bassi et al., 2009]
- to design schemes suited for both compressible and incompressible flows
- to simplify the implicit implementation of a method
- ...

DG discretization of the fluid dynamics equations

The governing equations can be written in compact form as

$$\mathbf{P}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{w}) + \nabla \cdot \mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w}) + \mathbf{s}(\mathbf{w}, \nabla \mathbf{w}) = \mathbf{0}$$

for compressible flows a common choice for **w** is

$$\mathbf{w}_c = [\rho, \rho u_i, \rho E, \rho k, \rho \widetilde{\omega}]^T \rightarrow \mathbf{P}\left(\mathbf{w}\right) = \mathbf{I}$$

Alternatives to $\mathbf{w}_c$ have been investigated by several authors in order

- to obtain a well defined behavior of variables in the incompressible limit of compressible flows
- to deal with low Mach number flows (*p, **u**, T*) [Bassi et al., 2009]
- to design schemes suited for both compressible and incompressible flows
- to simplify the implicit implementation of a method
- to ensure the positivity of thermodynamic variables at discrete level

DG discretization of the fluid dynamics equations

The working variables

The governing equations can be written in compact form as

$$\mathbf{P}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{w}) + \nabla \cdot \mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w}) + \mathbf{s}(\mathbf{w}, \nabla \mathbf{w}) = \mathbf{0}$$

we adopt a set of variables based on $\widetilde{p} = \log(p)$ and $\widetilde{T} = \log(T)$ to ensure the positivity of all thermodynamic variables at discrete level

$$\mathbf{w} = \left[\widetilde{p}, u_i, \widetilde{T}, k, \widetilde{\omega}\right]^T \qquad \mathbf{P}\left(\mathbf{w}\right) = \frac{\partial \mathbf{w}_c}{\partial \mathbf{w}}$$

- unlike $\widetilde{\omega}$ equation, we do not transform the equations, we substitute p, T with $e^{\widetilde{p}}$, $e^{\widetilde{T}}$ and use a polynomial approximation for $\widetilde{p}$ and $\widetilde{T}$
- this approach certainly improved the robustness of high-order simulations of transonic flows

DG discretization of the fluid dynamics equations

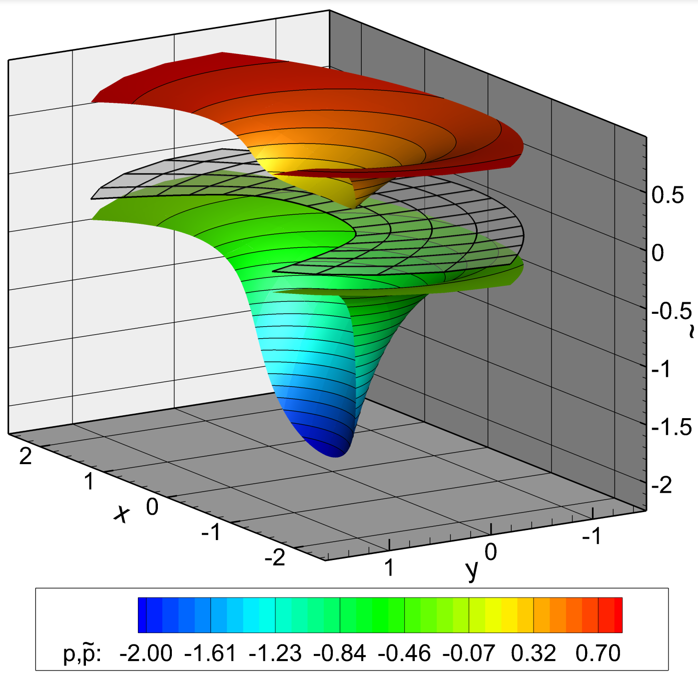
The logarithmic working variables - a numerical experiment

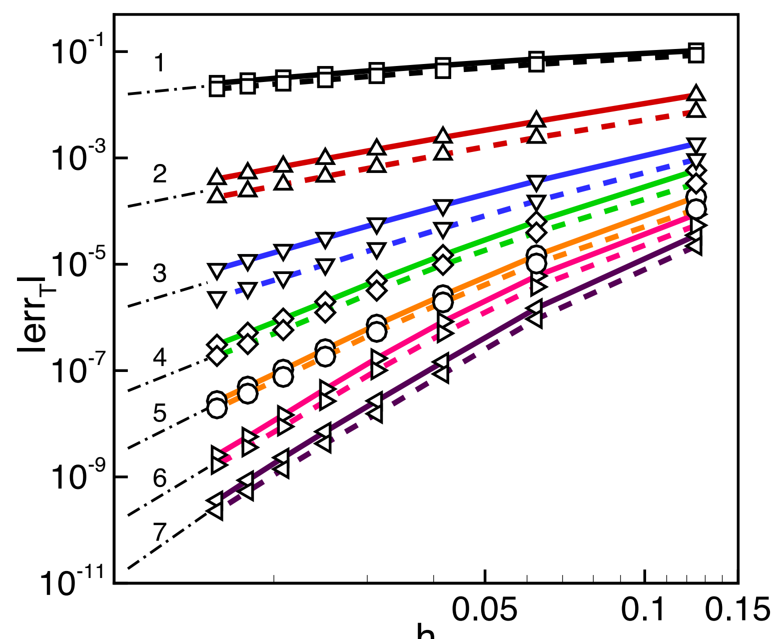
We work with polynomial approximations not directly for p and T but for their logarithms $\widetilde{p} = log(p)$ and $\widetilde{T} = log(T)$

$$\widetilde{p}(\mathbf{x}) = \phi_j(\mathbf{x}) W_j^{\widetilde{p}}, \quad \widetilde{T}(\mathbf{x}) = \phi_j(\mathbf{x}) W_j^{\widetilde{T}}, \quad j = 1, \cdots, N_{dof}^k$$

In this way the computed values $p = e^{\widetilde{p}}$ and $T = e^{\widetilde{T}}$ are always positive

Improved code robustness with a low implementation effort $(\mathbf{M_P})$!





DG discretization of the fluid dynamics equations

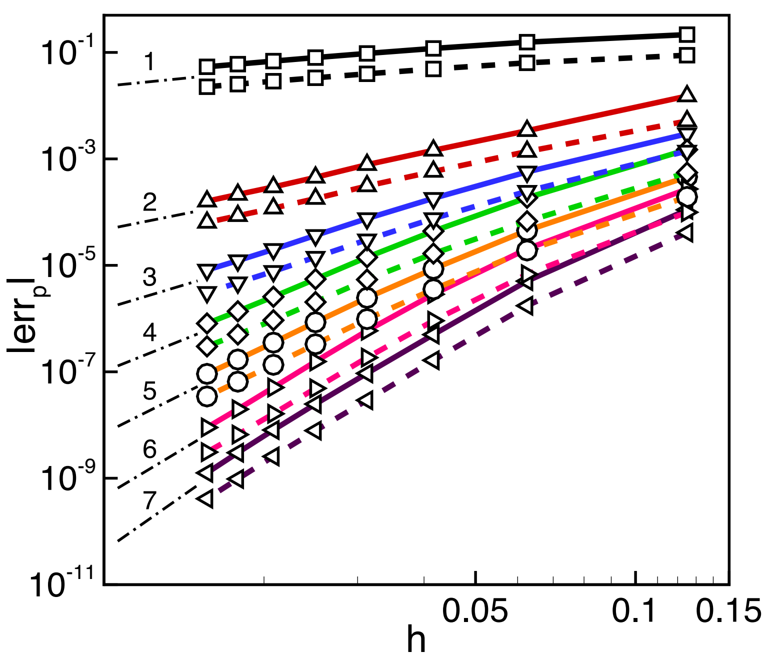
The logarithmic working variables - a numerical experiment

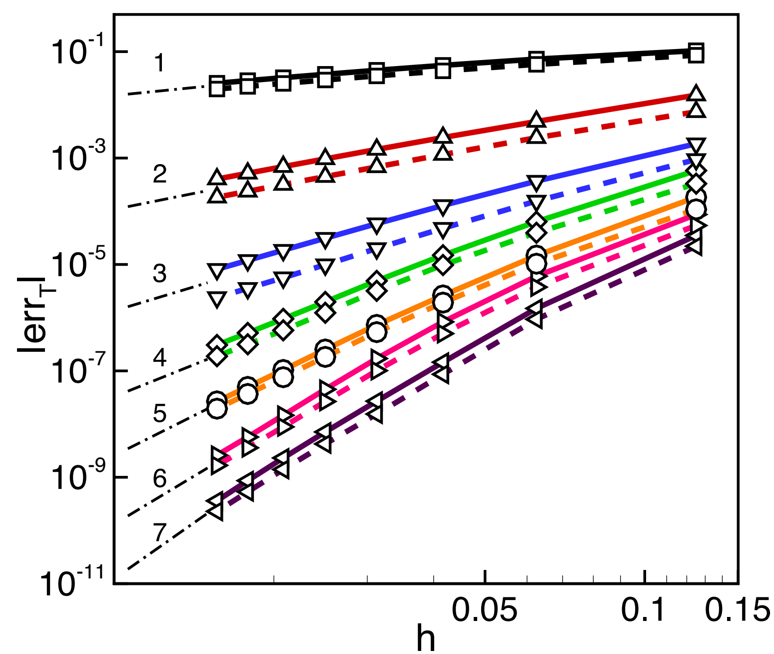
We work with polynomial approximations not directly for p and T but for their logarithms $\widetilde{p} = log(p)$ and $\widetilde{T} = log(T)$

$$\widetilde{p}(\mathbf{x}) = \phi_j(\mathbf{x}) W_j^{\widetilde{p}}, \quad \widetilde{T}(\mathbf{x}) = \phi_j(\mathbf{x}) W_j^{\widetilde{T}}, \quad j = 1, \cdots, N_{dof}^k$$

In this way the computed values $p = e^{\widetilde{p}}$ and $T = e^{\widetilde{T}}$ are always positive

Improved code robustness with a low implementation effort $(\mathbf{M_P})$!





DG discretization of the fluid dynamics equations

The DG discretization equations consists in seeking, for *j = 1, . . . , m*, the elements of the global vector **W** of unknown dof s.t.

$$\sum_{T\in\mathcal{T}_h}\int_T \phi_i P_{j,k}\left(\mathbf{w}_h\right)\phi_l \frac{dW_{k,l}}{dt}d\mathbf{x} - \sum_{T\in\mathcal{T}_h}\int_T \frac{\partial \phi_i}{\partial x_n} F_{j,n}\left(\mathbf{w}_h, \nabla\mathbf{w}_h + \mathbf{r}\left(\llbracket\mathbf{w}_h\rrbracket\right)\right)d\mathbf{x}$$

$$+\sum_{F\in\mathcal{F}_h}\int_F \llbracket\phi_i\rrbracket_n \widehat{F}_{j,n}\left(\mathbf{w}_h^{\pm}, \left(\nabla\mathbf{w}_h + \eta_F\mathbf{r}_F\left(\llbracket\mathbf{w}_h\rrbracket\right)\right)^{\pm}\right)d\sigma$$

$$+\sum_{T\in\mathcal{T}_h}\int_T \phi_i s_j\left(\mathbf{w}_h, \nabla\mathbf{w}_h + \mathbf{r}\left(\llbracket\mathbf{w}_h\rrbracket\right)\right)d\mathbf{x} = 0 \qquad i = 1,\ldots,N_{dof}^T$$

repeated indices imply summation $k = 1,\ldots,m,\ l = 1,\ldots,N_{dof}^T,\ n = 1,\ldots,d$

For compressible flows interface convective fluxes treated with the exact Riemann solver of [Gottlieb and Groth, 1988] or the van Leer flux vector splitting method as modified by [H¨anel et al., 1987]

BR2 scheme for the viscous term [Bassi and Rebay, 1997, Arnold et al., 2002]

DG discretization of the fluid dynamics equations

The DG discretization equations consists in seeking, for *j = 1, . . . , m*, the elements of the global vector **W** of unknown dof s.t.

$$\sum_{T\in\mathcal{T}_h}\int_T \phi_i P_{j,k}\left(\mathbf{w}_h\right)\phi_l \frac{dW_{k,l}}{dt}d\mathbf{x} - \sum_{T\in\mathcal{T}_h}\int_T \frac{\partial\phi_i}{\partial x_n}F_{j,n}\left(\mathbf{w}_h,\nabla\mathbf{w}_h+\mathbf{r}\left(\llbracket\mathbf{w}_h\rrbracket\right)\right)d\mathbf{x}$$

$$+\sum_{F\in\mathcal{F}_h}\int_F \llbracket\phi_i\rrbracket_n \widehat{F}_{j,n}\left(\mathbf{w}_h^{\pm},\left(\nabla\mathbf{w}_h+\eta_F\mathbf{r}_F\left(\llbracket\mathbf{w}_h\rrbracket\right)\right)^{\pm}\right)d\sigma$$

$$+\sum_{T\in\mathcal{T}_h}\int_T \phi_i s_j\left(\mathbf{w}_h,\nabla\mathbf{w}_h+\mathbf{r}\left(\llbracket\mathbf{w}_h\rrbracket\right)\right)d\mathbf{x}=0 \qquad i=1,\ldots,N_{dof}^T$$

repeated indices imply summation $k=1,\ldots,m,\ l=1,\ldots,N_{dof}^T,\ n=1,\ldots,d$

For compressible flows interface convective fluxes treated with the exact Riemann solver of [Gottlieb and Groth, 1988] or the van Leer flux vector splitting method as modified by [H¨anel et al., 1987]

BR2 scheme for the viscous term [Bassi and Rebay, 1997, Arnold et al., 2002]

DG discretization of the fluid dynamics equations

The DG discretization equations consists in seeking, for *j = 1, . . . , m*, the elements of the global vector **W** of unknown dof s.t.

$$\sum_{T\in\mathcal{T}_h}\int_T \phi_i P_{j,k}\left(\mathbf{w}_h\right)\phi_l \frac{dW_{k,l}}{dt}d\mathbf{x} - \sum_{T\in\mathcal{T}_h}\int_T \frac{\partial \phi_i}{\partial x_n} F_{j,n}\left(\mathbf{w}_h, \nabla\mathbf{w}_h + \mathbf{r}\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)d\mathbf{x}$$

$$+ \sum_{F\in\mathcal{F}_h}\int_F \llbracket \phi_i \rrbracket_n \widehat{F}_{j,n}\left(\mathbf{w}_h^{\pm}, \left(\nabla\mathbf{w}_h + \eta_F \mathbf{r}_F\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)^{\pm}\right)d\sigma$$

$$+ \sum_{T\in\mathcal{T}_h}\int_T \phi_i s_j\left(\mathbf{w}_h, \nabla\mathbf{w}_h + \mathbf{r}\left(\llbracket \mathbf{w}_h \rrbracket\right)\right)d\mathbf{x} = 0 \qquad i = 1,\ldots,N_{dof}^T$$

repeated indices imply summation $k = 1,\ldots,m$, $l = 1,\ldots,N_{dof}^T$, $n = 1,\ldots,d$

For compressible flows interface convective fluxes treated with the exact Riemann solver of [Gottlieb and Groth, 1988] or the van Leer flux vector splitting method as modified by [H¨anel et al., 1987]

BR2 scheme for the viscous term [Bassi and Rebay, 1997, Arnold et al., 2002]

DG applied to CFD

A fully coupled approach to the incompressible flows simulation

UNIVERSITÀ DEGLI STUDI DI BERGAMO

For the incompressible case we lack of a time derivative in the mass equation

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_j) = 0$$

The continuity equation is a constraint!

<u>*Main features of the method*</u>

- fully implicit
- the inviscid interface flux rely on a local Riemann problem associated with an artificial compressibility perturbation of the Euler equation
- the formulation of a suitable convective flux naturally fits within the DG framework
- this tight coupling between pressure and velocity stabilizes the method and allows equal-order approximation for both pressure and velocity
- a pretty natural way to extent an existing compressible solver

DG applied to CFD

A fully coupled approach to the incompressible flows simulation

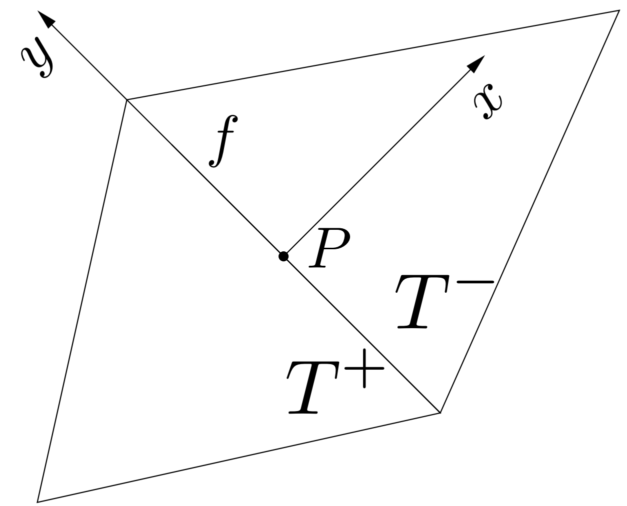
UNIVERSITÀ DEGLI STUDI DI BERGAMO

For notation brevity we set

$$u \stackrel{\text{def}}{=} \mathbf{u}\cdot\mathbf{n}, \quad \mathbf{v} \stackrel{\text{def}}{=} \mathbf{u} - u\mathbf{n}$$

restricting the problem to the normal direction

$$\begin{aligned} \frac{\partial u}{\partial x} &= 0, \\ \frac{\partial u}{\partial t} + \frac{\partial (u^2 + p)}{\partial x} &= 0, \\ \frac{\partial \mathbf{v}}{\partial t} + \frac{\partial (u\mathbf{v})}{\partial x} &= 0 \end{aligned}$$



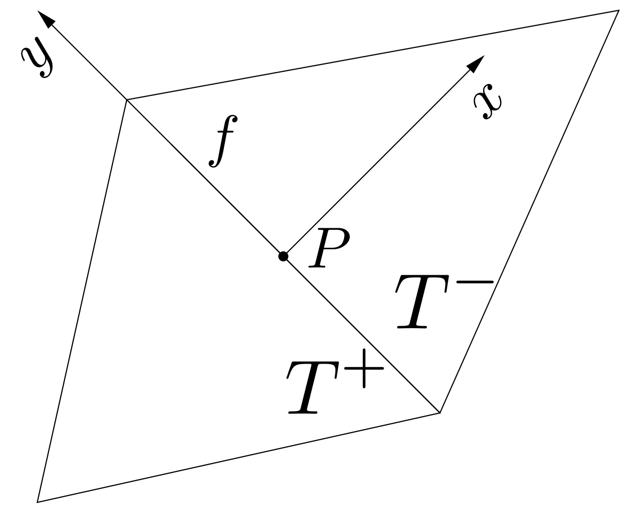
DG applied to CFD

A fully coupled approach to the incompressible flows simulation

UNIVERSITÀ DEGLI STUDI DI BERGAMO

For notation brevity we set

$$u \stackrel{\mathrm{def}}{=} \mathbf{u}\cdot\mathbf{n}, \quad \mathbf{v} \stackrel{\mathrm{def}}{=} \mathbf{u} - u\mathbf{n}$$



restricting the problem to the normal direction

$$\frac{1}{c^2}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} = 0,$$
$$\frac{\partial u}{\partial t} + \frac{\partial (u^2 + p)}{\partial x} = 0,$$
$$\frac{\partial \mathbf{v}}{\partial t} + \frac{\partial (u\mathbf{v})}{\partial x} = 0$$

The hyperbolic nature of the local problem can be recovered by adding an artificial compressibility term to the mass equation

DG applied to CFD

A fully coupled approach to the incompressible flows simulation

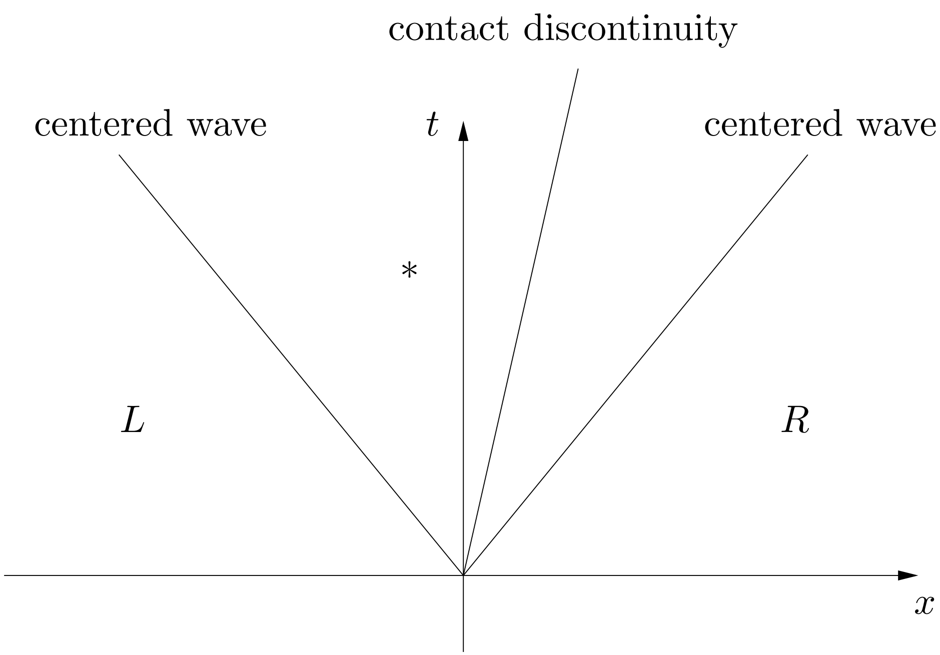
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

The solution of the Riemann problem entails four states separated by two centered waves a contact discontinuity

Riemann invariants (rarefactions) and the Rankine-Hugoniot jump conditions (shocks) allows obtain the problem solution **u***, *p**

We define the inviscid fluxes as

$$\hat{\mathbf{F}}(\mathbf{u}_h^+, p_h^+; \mathbf{u}_h^-, p_h^-) \stackrel{\text{def}}{=} \mathbf{F}(\mathbf{u}^*, p^*)$$



The normal velocity component is given by a non-linear equation
the pressure is easily determined once **u*** and the wave pattern are known

Time integration - unsteady problems

UNIVERSITÀ DEGLI STUDI DI BERGAMO

DG space discretized equations can be written as a system of ODEs\DAEs

$$\mathbf{M_P}\left(\mathbf{W}\right)\frac{d\mathbf{W}}{dt}+\mathbf{R}\left(\mathbf{W}\right)=\mathbf{0}$$

$\mathbf{R}$ is the vector of residuals and $\mathbf{M_P}$ is the global block diagonal matrix

if $\mathbf{w}=\mathbf{w}_c \rightarrow \mathbf{M_P}\left(\mathbf{W}\right)=\mathbf{I}$; if $\mathbf{w}=\mathbf{w}_p \rightarrow \mathbf{M_P}\left(\mathbf{W}\right)=\mathbf{I}-\mathbf{J}^{1,1}$

Implicit accurate time integration by means of linearly implicit Rosenbrock-type Runge-Kutta schemes [Bassi et al., 2007, Bassi et al., 2014b]

$$\mathbf{W}^{n+1}=\mathbf{W}^n+\sum_{j=1}^{s} b_j\mathbf{K}_j$$

$$\left(\frac{\mathbf{I}}{\Delta t}+\gamma\widetilde{\mathbf{J}}\right)^n\mathbf{K}_i=-\widetilde{\mathbf{R}}\left(\mathbf{W}^n+\sum_{j=1}^{i-1}\alpha_{ij}\mathbf{K}_j\right)-\widetilde{\mathbf{J}}^n\sum_{j=1}^{i-1}\gamma_{ij}\mathbf{K}_j \quad i=1,\ldots,s$$

where

$$\mathbf{J}=\frac{\partial\mathbf{R}}{\partial\mathbf{W}} \qquad \widetilde{\mathbf{R}}=\mathbf{M_P^{-1}}\mathbf{R} \qquad \widetilde{\mathbf{J}}=\frac{\partial\widetilde{\mathbf{R}}}{\partial\mathbf{W}}=\mathbf{M_P^{-1}}\left(\mathbf{J}-\frac{\partial\mathbf{M_P}}{\partial\mathbf{W}}\widetilde{\mathbf{R}}\right)$$

and b_i, α_{ij}, γ_{ij} are real coefficients

Time integration - unsteady problems

Rosenbrock schemes

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Equivalent formulation to avoid the matrix-vector product $\mathbf{J}^n \sum_{j=1}^{i-1} \gamma_{ij} \mathbf{K}_j$ and more suited for implementation when dealing with change of variables

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \sum_{j=1}^{s} m_j \mathbf{Y}_j$$

$$\left(\frac{\mathbf{M_P}}{\gamma \Delta t} + \mathbf{J} - \frac{\partial \mathbf{M_P}}{\partial \mathbf{W}} \widetilde{\mathbf{R}} \right)^n \mathbf{Y}_i = -\mathbf{M}_\mathbf{P}^n \left[\widetilde{\mathbf{R}} \left(\mathbf{W}^n + \sum_{j=1}^{i-1} a_{ij} \mathbf{Y}_j \right) - \sum_{j=1}^{i-1} \frac{c_{ij}}{\Delta t} \mathbf{Y}_j \right]$$

$$i = 1, \ldots, s$$

the coefficients of the transformed scheme are given by

$$(m_1, \ldots, m_s) = (b_1, \ldots, b_s)\, \mathbf{\Gamma}^{-1} \qquad (a_{ij}) = (\alpha_{ij})\, \mathbf{\Gamma}^{-1} \qquad (c_{ij}) = \gamma^{-1} \mathbf{I}_s - \mathbf{\Gamma}^{-1}$$

where $\mathbf{\Gamma}^{-1} \overset{\text{def}}{=} (\gamma_{ij})^{-1}$ is the inverse of the matrix of coefficients (γ_{ij})

Only a linear system need to be solved for each stage *i.e.* the Jacobian $\mathbf{J} = \partial \mathbf{R} / \partial \mathbf{W}$ is assembled and factored only once per time step!

Why high-order Rosenbrock schemes?

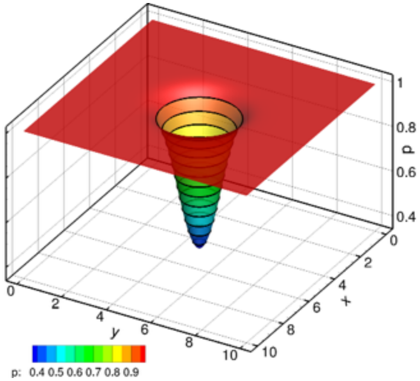
UNIVERSITÀ DEGLI STUDI DI BERGAMO

Several high-order temporal schemes are implemented

- Modified Extended BDF
- Two Implicit Advanced Step-point (TIAS)
- Explicit Singly Diagonally Implicit R-K (ESDIRK)
- linearly implicit Rosenbrock method

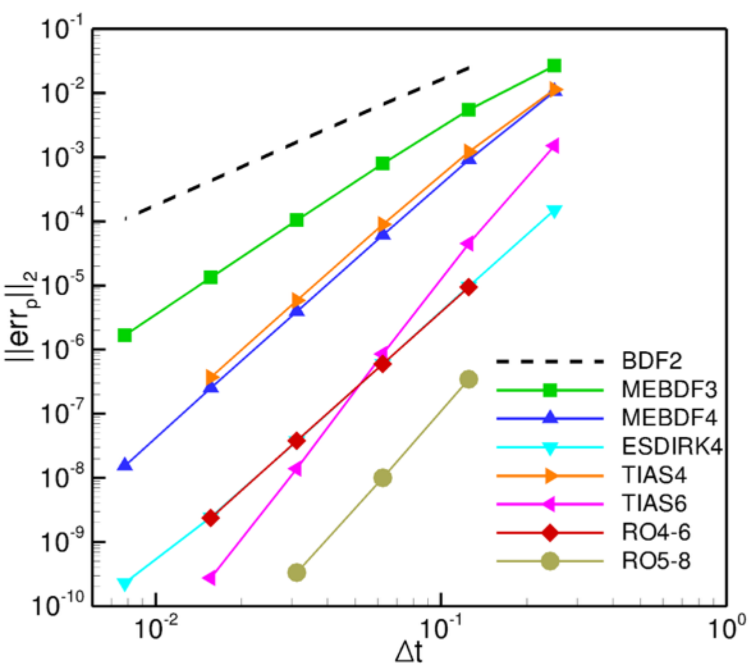
non-linear systems solution

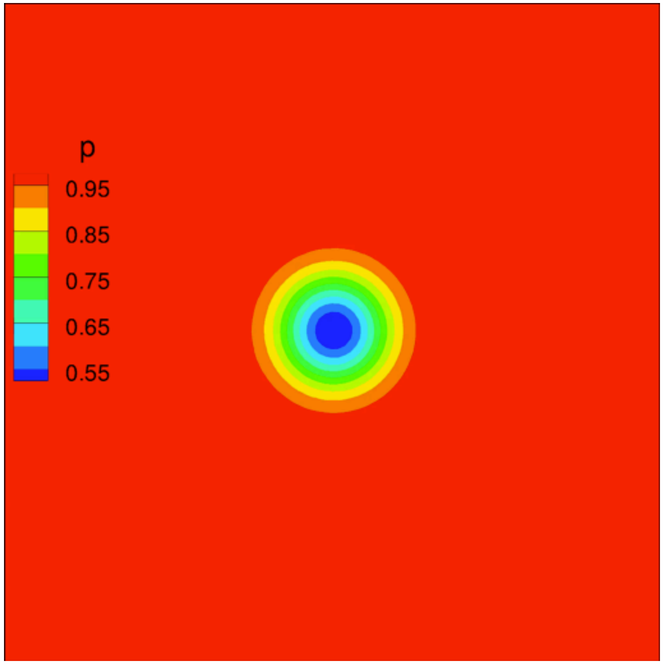
linear systems solution (here via GMRES)



i) Hi-O schemes are more efficient than Lo-O ones for high required accuracy

ii) Rosenbrock-type schemes are appealing both for accuracy and efficiency





Convection of an isentropic vortex

P^6 solution on 50X50 el.

Why high-order Rosenbrock schemes?

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

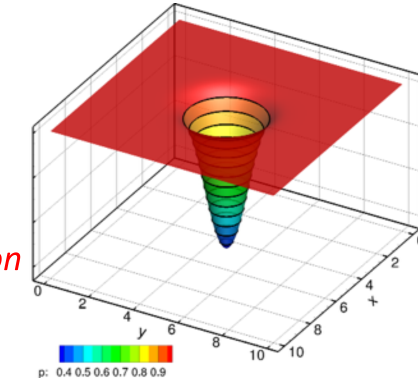
Several high-order temporal schemes are implemented

- Modified Extended BDF
- Two Implicit Advanced Step-point (TIAS)
- Explicit Singly Diagonally Implicit R-K (ESDIRK)

} *non-linear systems solution*

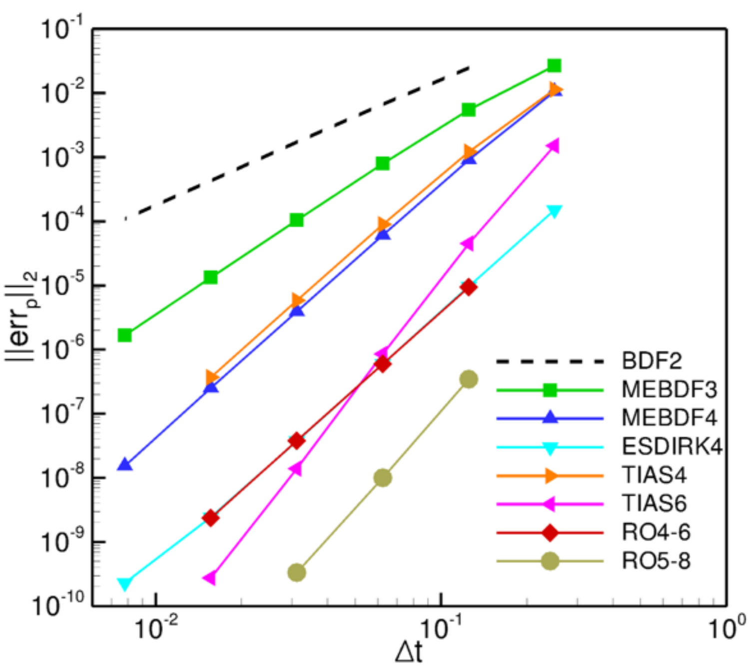
- linearly implicit Rosenbrock method

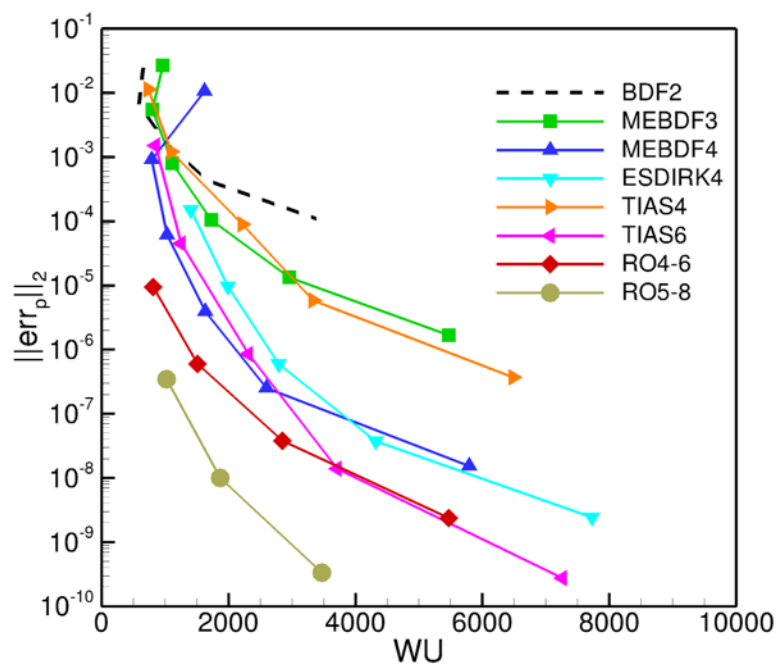
linear systems solution (here via GMRES)



i) Hi-O schemes are more efficient than Lo-O ones for high required accuracy

ii) Rosenbrock-type schemes are appealing both for accuracy and efficiency





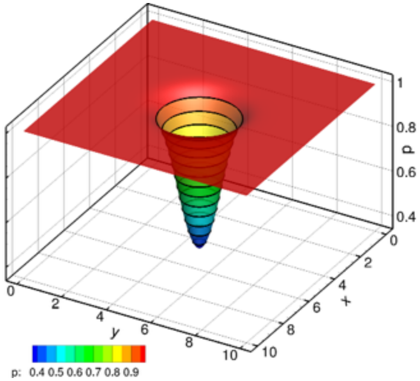
Convection of an isentropic vortex

P^6 solution on 50X50 el.

Why high-order Rosenbrock schemes?

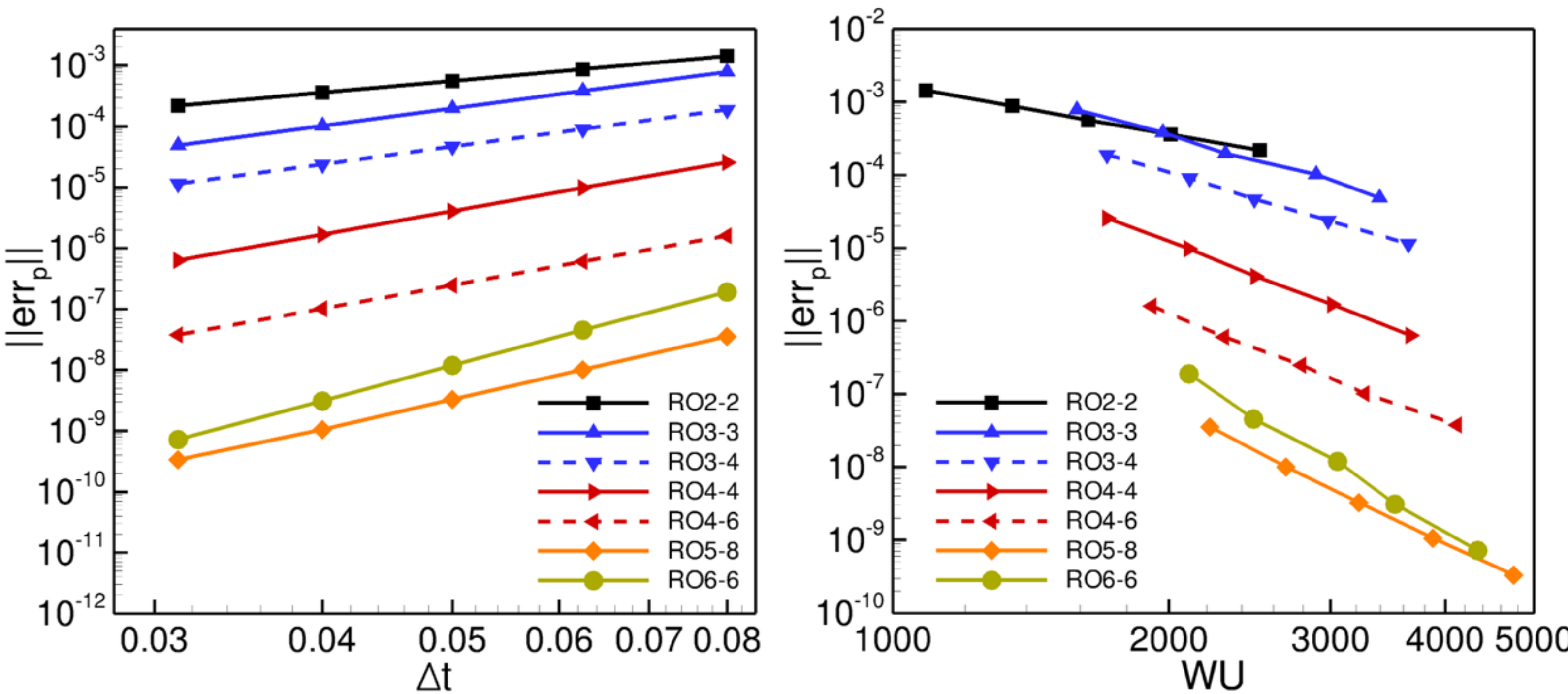
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

Several Rosenbrock schemes, from order two to order six, have been compared



No need to "exactly" solve systems: GMRES tolerance can be increased with confidence with a significant reduction of WU

For a given order of accuracy, among the schemes considered, those with more stages are more accurate and efficient, *e.g.* RO5-8 vs. RO6-6



Convection of an isentropic vortex

P^6 solution on 50X50 el.

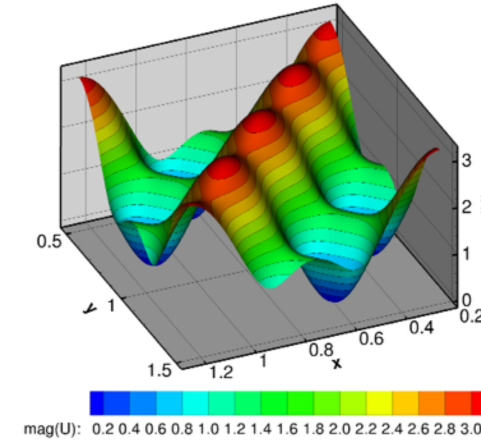
Why high-order Rosenbrock schemes?

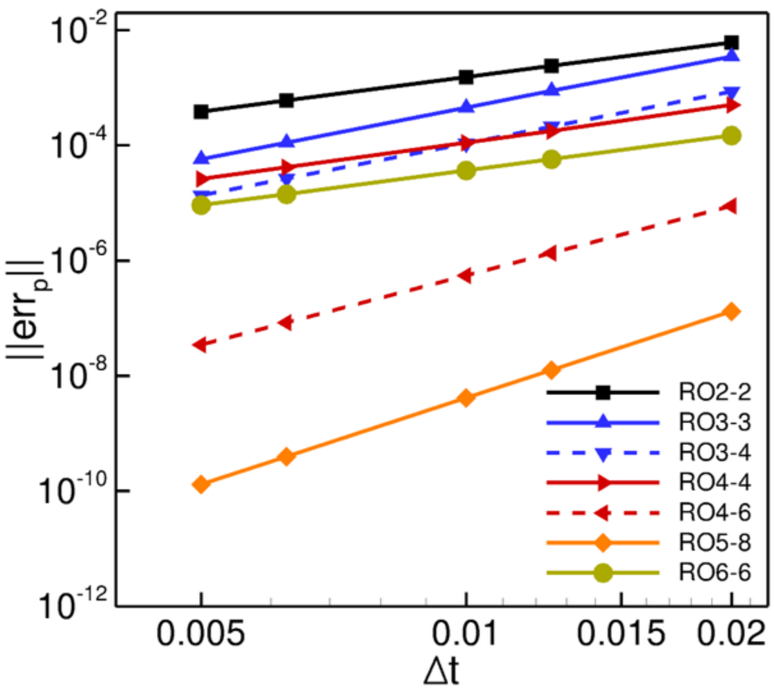
UNIVERSITÀ DEGLI STUDI DI BERGAMO

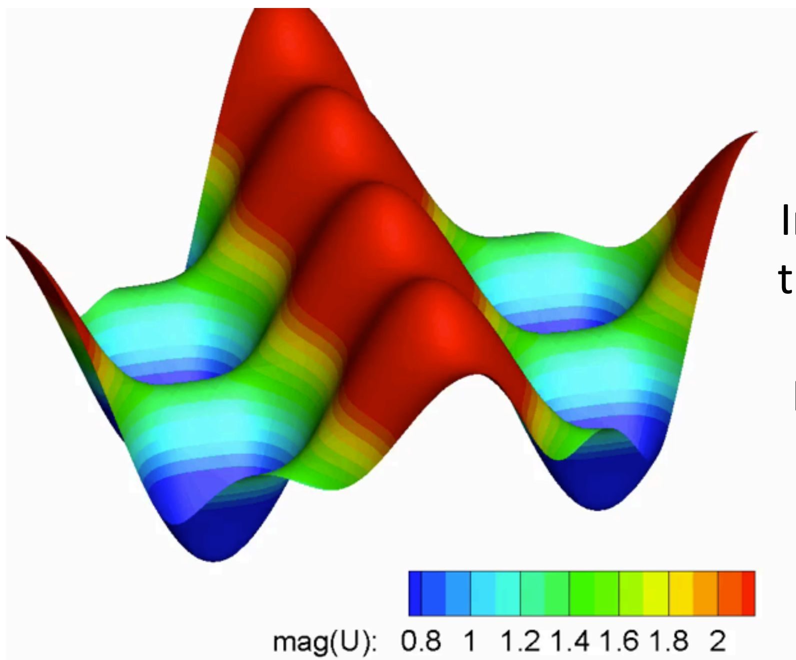
Some schemes are suited for DAEs systems, as those arising from our discretization for incompressible flows

Those schemes verify the formal order of convergence for all the variables, while schemes designed for ODEs only, exhibit order reduction for pressure error

For the incompressible case we observed a far more difficult convergence of the iterative solver than the compressible case







Incompressible traveling waves

P^6 solution on 50X50 el.

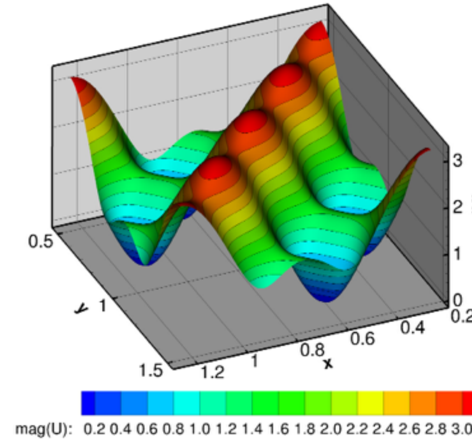
Why high-order Rosenbrock schemes?

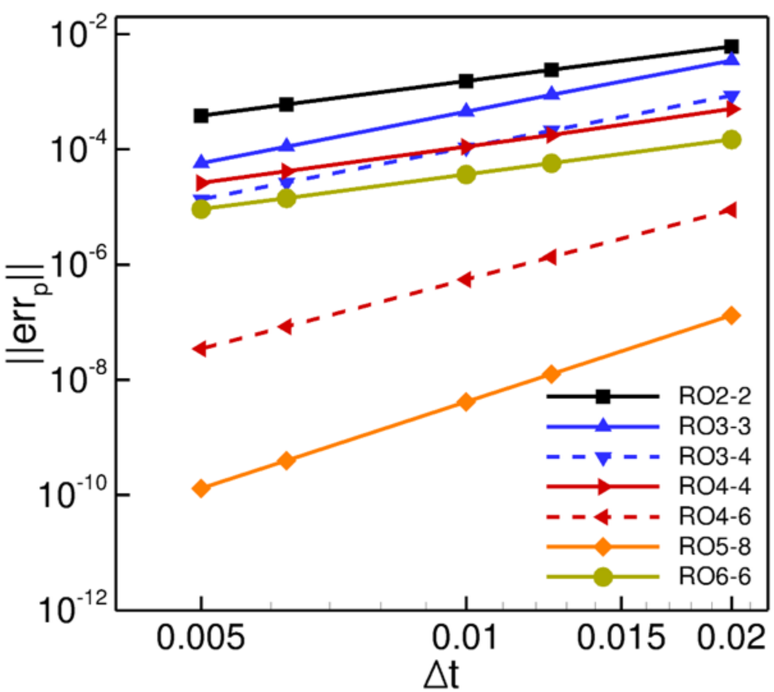
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

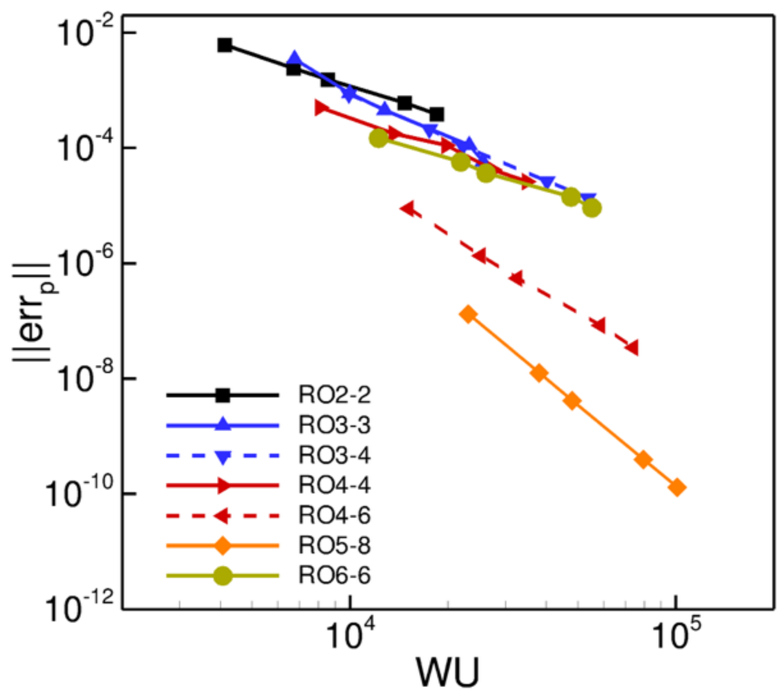
Some schemes are suited for DAEs systems, as those arising from our discretization for incompressible flows

Those schemes verify the formal order of convergence for all the variables, while schemes designed for ODEs only, exhibit order reduction for pressure error

For the incompressible case we observed a far more difficult convergence of the iterative solver than the compressible case







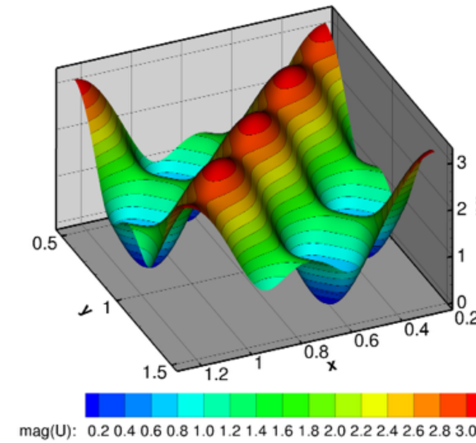
Incompressible traveling waves

P^6 solution on 50X50 el.

Why high-order Rosenbrock schemes?

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

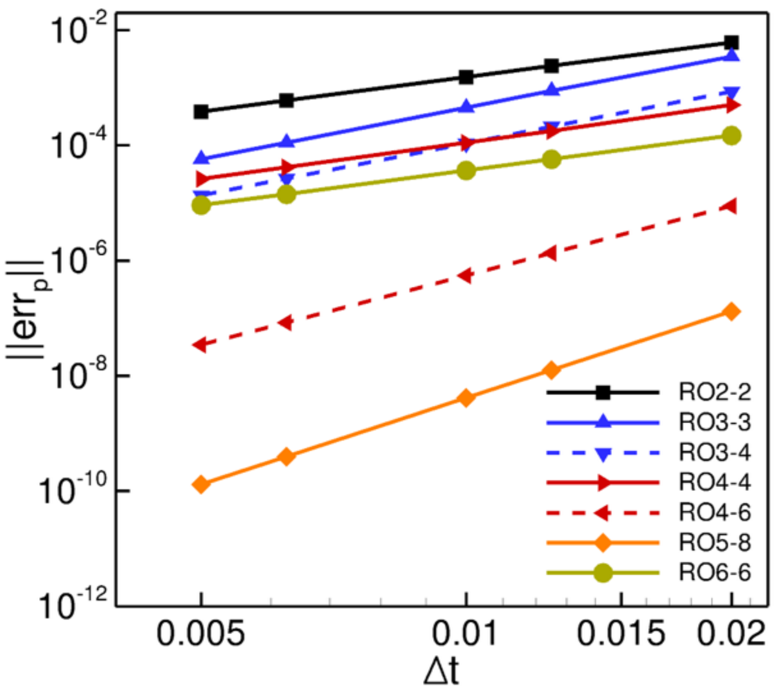
Some schemes are suited for DAEs systems, as those arising from our discretization for incompressible flows

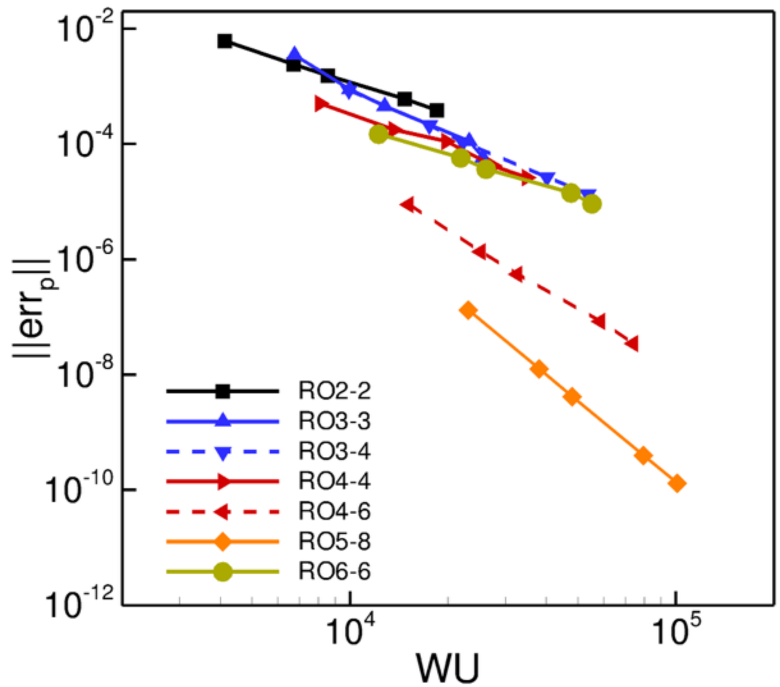


Those schemes verify the formal order of convergence for all the variables, while schemes designed for ODEs only, exhibit order reduction for pressure error

For the i ... difficult converg ... sible case

Initialization, time-step adaptation, h-MG, …





Incompressible traveling waves

P^6 solution on 50X50 el.

Why high-order Rosenbrock schemes?

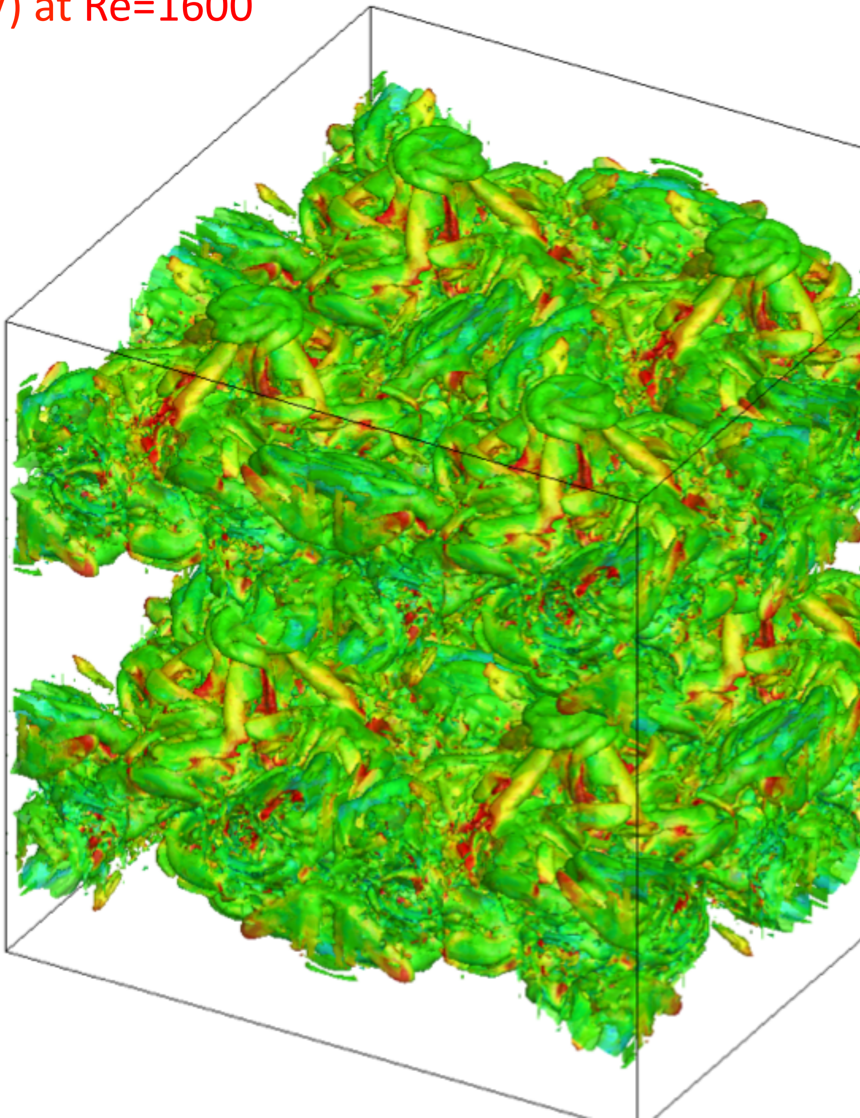
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

Comparison between the explicit RK4-5 and the RO5-8 on a DNS
Taylor Green Vortex (TGV) at Re=1600

A periodic and transitional flow defined by a simple initial condition that evolves in the creation of small scales followed by decay

	256^3 dofs		128^3 dofs		64^3 dofs	
k	ne_x	ne	ne_x	ne	ne_x	ne
1	161	4,194,304	–	–	–	–
2	119	1,677,722	59	209,715	–	–
3	94	838,861	47	104,858	24	13,107
4	–	–	39	59,919	20	7,490
5	–	–	–	–	17	4,681

Three given number of DOFs obtained by varying the polynomial degree together with mesh density are considered



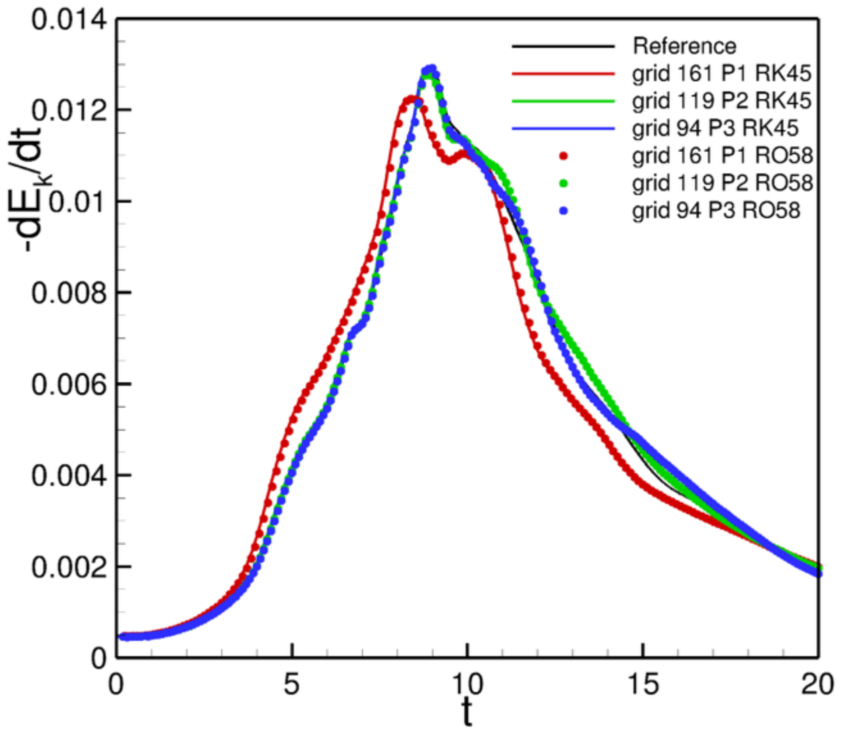
Why high-order Rosenbrock schemes?

UNIVERSITÀ DEGLI STUDI DI BERGAMO

Comparison between the explicit RK4-5 and the RO5-8 on a DNS
Taylor Green Vortex (TGV) at Re=1600

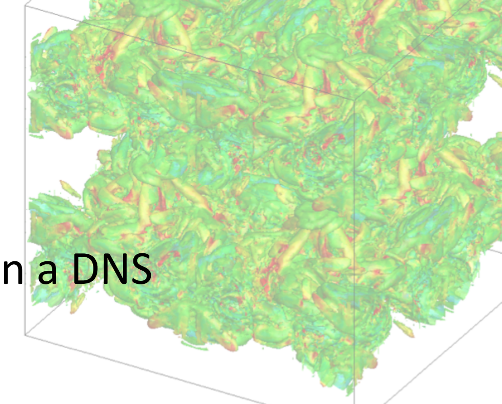
A periodic and transitional flow defined by a simple initial condition that evolves in the creation of small scales followed by decay

	256^3 dofs		128^3 dofs		64^3 dofs	
k	ne_x	ne	ne_x	ne	ne_x	ne
1	161	4,194,304	–	–	–	–
2	119	1,677,722	59	209,715	–	–
3	94	838,861	47	104,858	24	13,107
4	–	–	39	59,919	20	7,490
5	–	–	–	–	17	4,681



Three given number of DOFs obtained by varying the polynomial degree together with mesh density are considered

Why high-order Rosenbrock schemes?



Comparison between the explicit RK4-5 and the RO5-8 on a DNS
Taylor Green Vortex (TGV) at Re=1600

		256^3 DOFs				128^3 DOFs				64^3 DOFs			
k	CFL	ne_x	n_{it}	CPU(s)	RAM(GB)	ne_x	n_{it}	CPU(s)	RAM(GB)	ne_x	n_{it}	CPU(s)	RAM(GB)
1	0.35	161	16,443	$23.0M$	28.12	–	–	–	–	–	–	–	–
2	0.2	119	21,149	$50.1M$	87.89	59	10,310	$2.8M$	11.76	–	–	–	–
3	0.15	94	22,099	$109.2M$	131.25	47	10,907	$6.6M$	17.97	24	5,443	$0.45M$	4.69
4	0.1	–	–	–	–	39	13,527	$19.7M$	26.95	20	6,770	$1.3M$	6.25
5	0.1	–	–	–	–	–	–	–	–	17	5,783	$2.7M$	7.81

$$CPU_r = CPU_{RK4-5}/CPU_{RO5-8}$$

	256^3 DOFs					128^3 DOFs					64^3 DOFs				
k	ne_x	f^*	n_{it}	CPU_r	RAM(GB)	ne_x	f^*	n_{it}	CPU_r	RAM(GB)	ne_x	f^*	n_{it}	CPU_r	RAM(GB)
1	161	0.12	167	1.174	272	–	–	–	–	–	–	–	–	–	–
2	119	0.1	200	2.934	608	59	0.17	119	2.476	76	–	–	–	–	–
3	94	0.1	200	2.057	1,140	47	0.12	167	2.447	144	24	0.17	119	1.890	23
4	–	–	–	–	–	39	0.12	167	3.222	248	20	0.17	119	2.372	36
5	–	–	–	–	–	–	–	–	–	–	17	0.17	119	2.292	56

The RO5-8 implicit scheme is roughly 2–3 times faster than the RK4-5 scheme
with a memory usage 6–10 times larger

– We are not using conservative variables! –

Time integration - steady problems (in brief)

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

For steady problems we rely on the Linearized Backward Euler (LBE) $(s = 1, \gamma = 1,\, m_1 = 1)$

$$\mathbf{W}^{n+1} = \mathbf{W}^{n} + \sum_{j=1}^{s} m_j \mathbf{Y}_j$$

$$\left(\frac{\mathbf{I}}{\gamma \Delta t} + \widetilde{\mathbf{J}}\right)^{n} \mathbf{Y}_i = -\widetilde{\mathbf{R}}\left(\mathbf{W}^{n} + \sum_{j=1}^{i-1} a_{ij}\mathbf{Y}_j\right) + \sum_{j=1}^{i-1} \frac{c_{ij}}{\Delta t}\mathbf{Y}_j \qquad i = 1, \dots, s$$

A pseudo-transient continuation strategy for the CFL number evolution is implemented, with local time step given by

$$\Delta t_T = \mathrm{CFL}\frac{h_T}{c + d}$$

$$c = |\mathbf{u}| + a, \qquad d = 2\frac{\mu_e + \lambda_e}{h_T}, \qquad h_T = d\frac{\Omega_T}{S_T}$$

Time integration - steady problems (in brief)

UNIVERSITÀ DEGLI STUDI DI BERGAMO

For steady problems we rely on the Linearized Backward Euler (LBE) $(s = 1, \gamma = 1, m_1 = 1)$

$$\mathbf{W}^{n+1} = \mathbf{W}^n + \Delta \mathbf{W}$$

$$\left(\frac{\mathbf{I}}{\Delta t} + \widetilde{\mathbf{J}} \right)^n \Delta \mathbf{W} = -\widetilde{\mathbf{R}} \left(\mathbf{W}^n \right)$$

A pseudo-transient continuation strategy for the CFL number evolution is implemented, with local time step given by

$$\Delta t_T = \mathrm{CFL} \frac{h_T}{c + d}$$

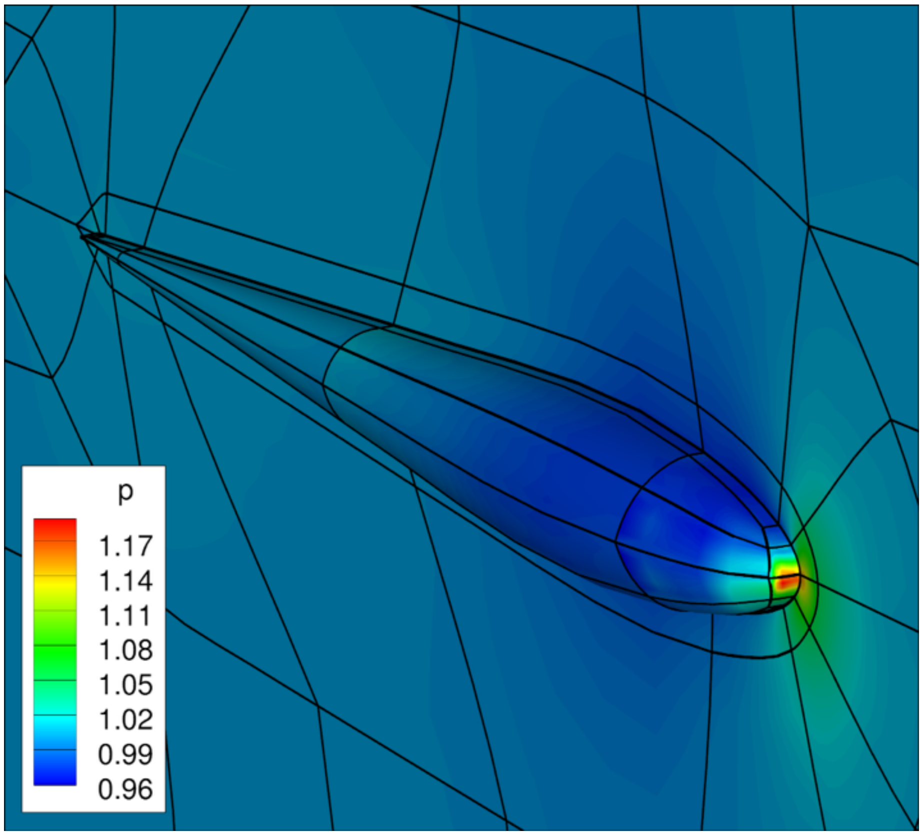
$$c = |\mathbf{u}| + a, \qquad d = 2 \frac{\mu_e + \lambda_e}{h_T}, \qquad h_T = d \frac{\Omega_T}{S_T}$$

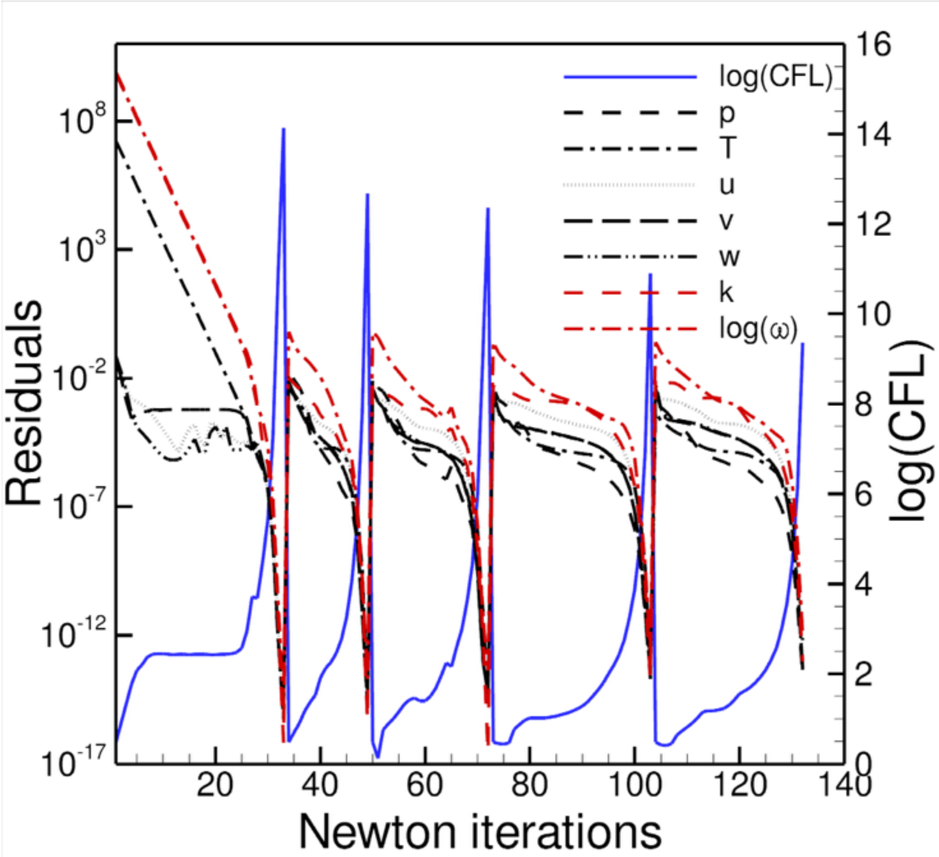
Time integration - steady problems *(in brief)*

The Newton quadratic convergence - a numerical experiment

UNIVERSITÀ DEGLI STUDI DI BERGAMO

For simple test cases implicit time integration combined with CFL evolution rule provides quadratic Newton convergence





BTC0, analytical 3D body of revolution, 832 hexahedral el. with quartic edges

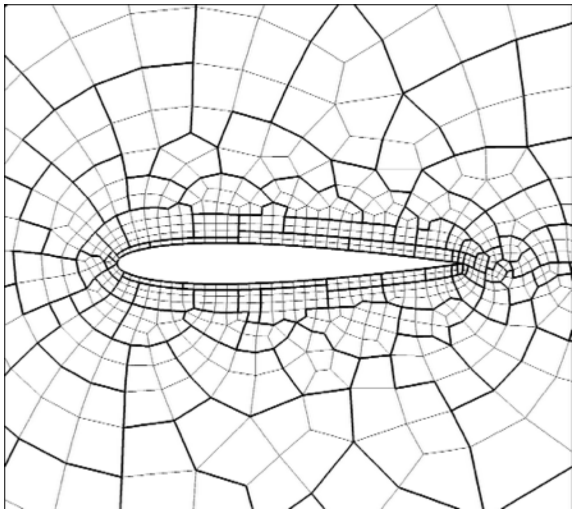
$M_\infty = 0.5, Re = 10^7, \alpha = 5^\circ$ (EARSM3)

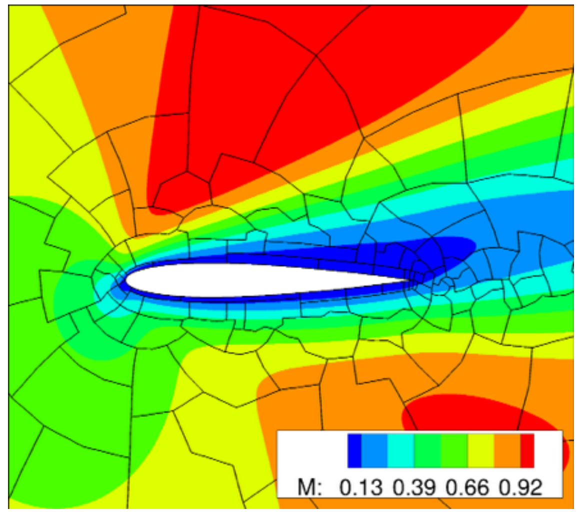
Time integration - steady problems *(in brief)*

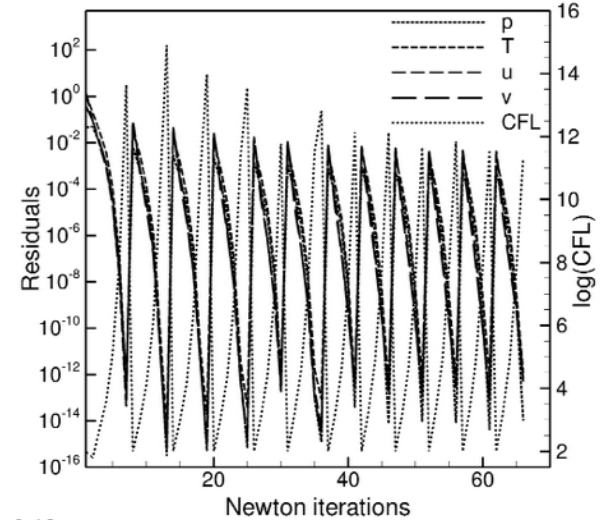
The Newton quadratic convergence - on aggl. meshes too

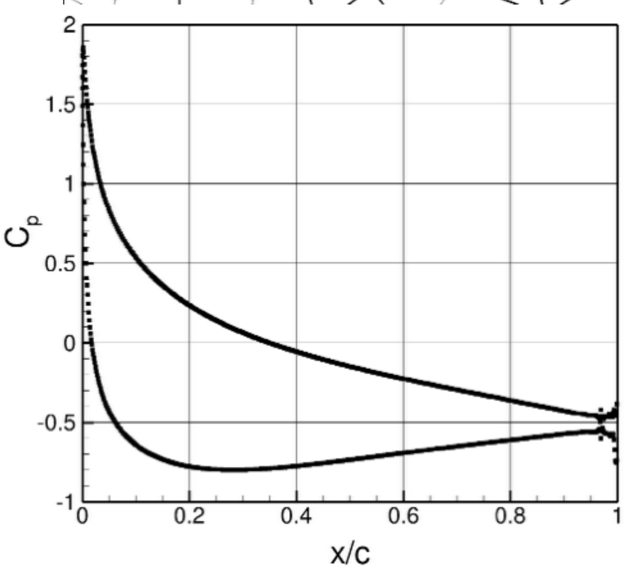
UNIVERSITÀ DEGLI STUDI DI BERGAMO

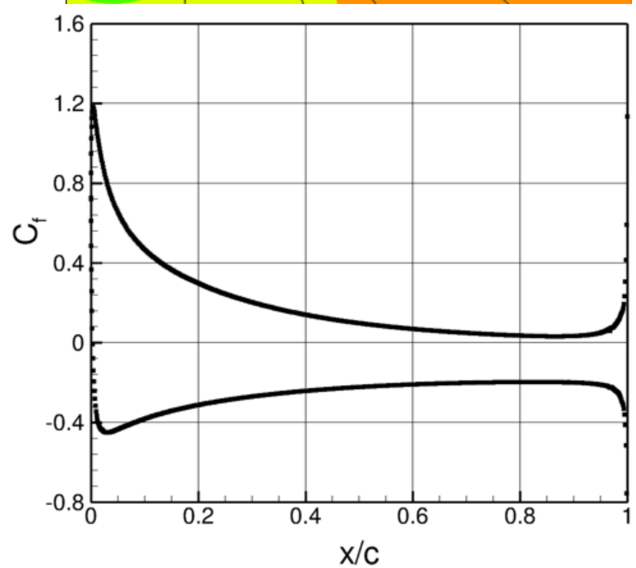
NACA0012 $M_\infty = 0.8, Re = 73, \alpha = 10°$, 178 agglomerated elements grid built on a 1197 hybrid mesh with cubic edges

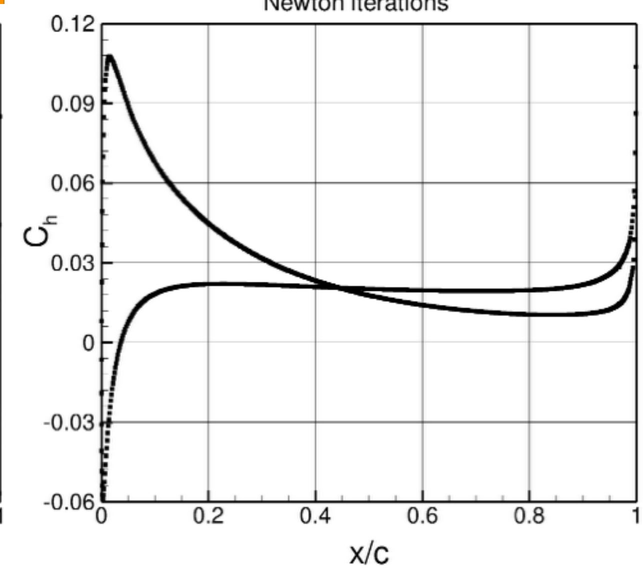












Parallel implementation

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

Parallel implementation features (to date)

- SPMD (Single Process, Multiple Data) paradigm
- pure MPI-based parallelism via the PETSc library
- thanks to the compactness of DG only the data of neighbors at partition boundaries need to be shared
- Data on remote processors are accessed through the PETSc interface to MPI
- Code inputs are the computational grid partitions (pre-processing)

Parallel implementation performance (PRACE preparatory)

FERMI	***Tier-0@CINECA (I)***	***BlueGene/Q***	***Strong****
GALILEO	***Tier-0@CINECA (I)***	***IBM NeXtScale***	***Strong***
CURIE	***Tier-0@GENCI (F)***	***bullx series***	***Strong****
HORNET	***Tier-1@HLRS (D)***	***Cray XC30***	***Strong and Weak****

*Comparison between explicit (ERK3-3) and implicit (RO3-3) time integrators

Parallel performance

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

CURIE

# core	ROS3P – P^3		ROS3P – P^4		ERK33 – P^3		ERK33 – P^4	
	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]
152	584.00	100.00	2810.08	100.00	42.44	100.00	155.59	100.00
264	336.61	99.89	1626.39	99.48	24.51	99.69	89.76	99.80
456	196.11	99.26	944.60	99.16	14.43	98.06	52.27	99.22
912	100.52	96.83	477.00	98.19	7.37	95.99	26.98	96.12
1824	52.98	91.86	243.66	96.11	3.80	93.01	13.69	94.69
3344	31.46	84.37	138.67	92.11	2.15	89.70	7.65	92.47
5016	24.35	72.68	99.05	85.97	1.51	84.96	5.16	91.41

FERMI

#core	ROS3P – P^3		ERK33 – P^3	
	time[s]	eff. [%]	time[s]	eff. [%]
1024	10808.8	100.00	741.33	100.00
2048	5478.91	98.64	375.75	98.65
4096	2820.37	95.81	192.63	96.21
8192	1472.11	91.78	97.67	94.87
16384	771.18	87.60	49.27	94.04
32768	414.49	81.49	25.19	91.97

GALILEO

#core	ERK33 – P^3	
	time[s]	eff. [%]
16	518.16	100.00
32	261.85	98.94
64	133.22	97.23
128	67.65	95.75
256	34.72	93.26
512	18.17	89.26

Preconditioned (Block Jacobi) GMRES parameters:
tolerance 1.e-14, maximum number of Krylov-subspaces vectors 20, maximum number of restarts 1

Parallel performance

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

HORNET Strong

# core	ROS3P – P^3		ROS3P – P^4		ERK33 – P^3		ERK33 – P^4	
	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]
256	1451.00	100.00	8122.48	100.00	149.68	100.00	579.48	100.00
512	729.20	99.49	4089.02	99.32	75.41	99.24	290.91	99.60
1024	372.28	97.44	2070.54	98.07	38.06	98.31	146.84	98.66
2048	195.08	92.97	1084.39	93.63	19.96	93.70	76.49	94.70
4096	107.64	84.25	567.38	89.47	10.30	90.76	39.52	91.64
8192	63.20	71.75	332.87	76.25	5.45	85.74	20.47	88.44

HORNET Weak

# core	ROS3P – P^3		ROS3P – P^4		ERK33 – P^3		ERK33 – P^4	
	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]
256	190.19	100.00	1059.08	100.00	18.89	100.00	74.09	100.00
512	190.87	99.64	1060.05	99.91	19.10	98.89	74.11	99.97
1024	191.23	99.46	1060.72	99.85	19.57	96.53	75.41	98.25
2048	195.08	97.49	1084.39	97.67	19.96	94.61	76.49	96.86
4096	198.48	95.82	1099.51	96.32	20.05	94.22	78.22	94.72

- the better strong scalability can be obtained with explicit schemes
- strong scalability improves with the polynomial degree
- strong scalability results are similar for all platforms

Parallel performance

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

HORNET Strong

# core	ROS3P – P^3		ROS3P – P^4		ERK33 – P^3		ERK33 – P^4	
	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]	time[s]	eff. [%]
256	1451.00	100.00	8122.48	100.00	149.68	100.00	579.48	100.00
512	729.20	99.49	4089.02	99.32	75.41	99.24	290.91	99.60
1024	372.28	97.44	2070.54	98.07	38.06	98.31	146.84	98.66
2048	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	94.70
409[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	91.64
819[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	88.44

HORNET Weak

# co[illegible]	[illegible]		[illegible]		[illegible]		[illegible]33 – P^4	
	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	eff. [%]
256	190.19	100.00	1059.08	100.00	18.89	100.00	74.09	100.00
512	190.87	99.64	1060.05	99.91	19.10	98.89	74.11	99.97
1024	191.23	99.46	1060.72	99.85	19.57	96.53	75.41	98.25
2048	195.08	97.49	1084.39	97.67	19.96	94.61	76.49	96.86
4096	198.48	95.82	1099.51	96.32	20.05	94.22	78.22	94.72

DGXTRA

Discontinuous **G**alerkin method for the

X-LES of **TRA**nsonic flows

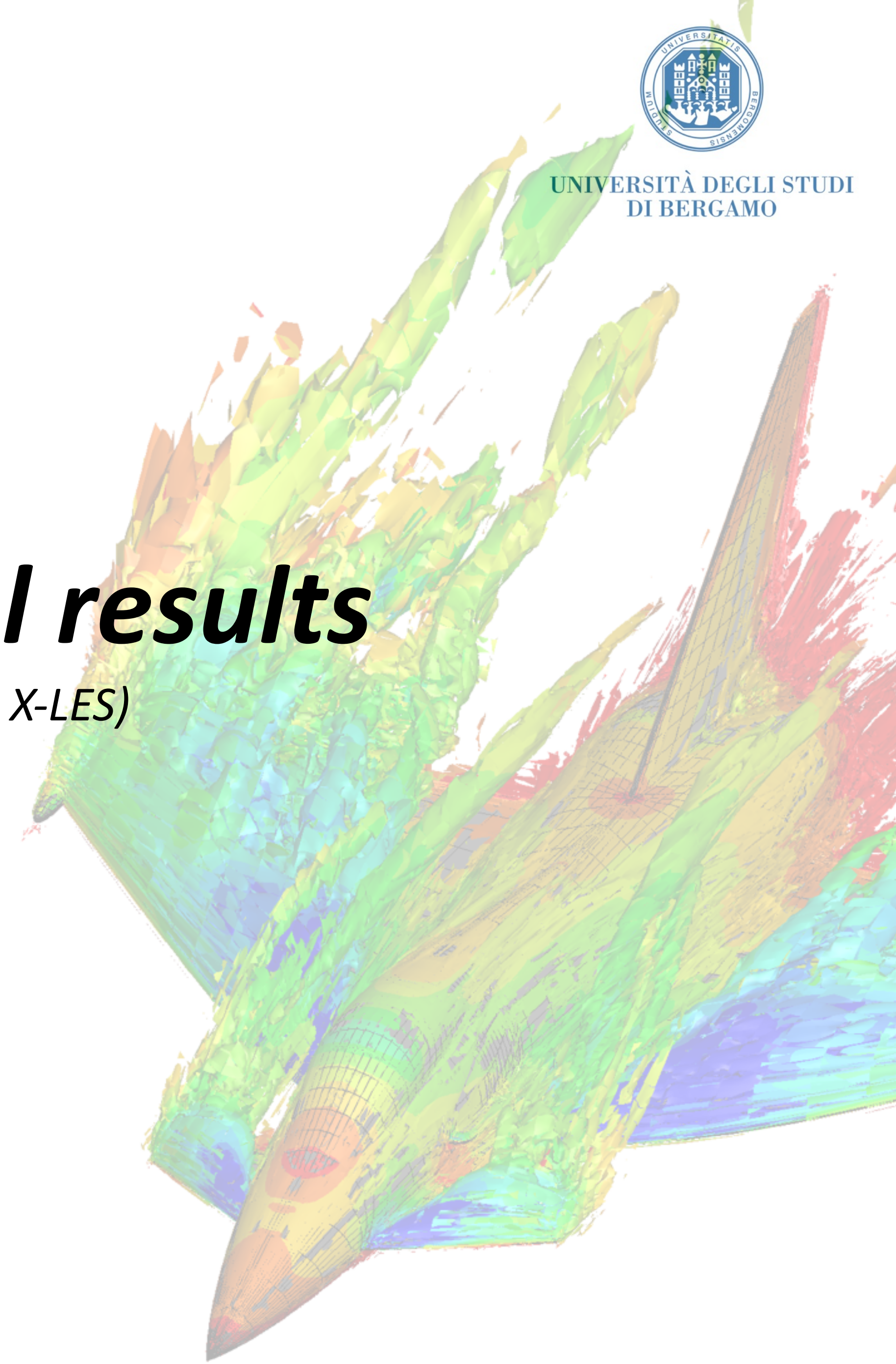
Funded by the 13th PRACE Call

PRACE

- the better strong scalability can be obtained with explicit schemes
- strong scalability improves with the polynomial degree
- strong scalability results are similar for all platforms

Numerical results

(DNS, ILES, X-LES)



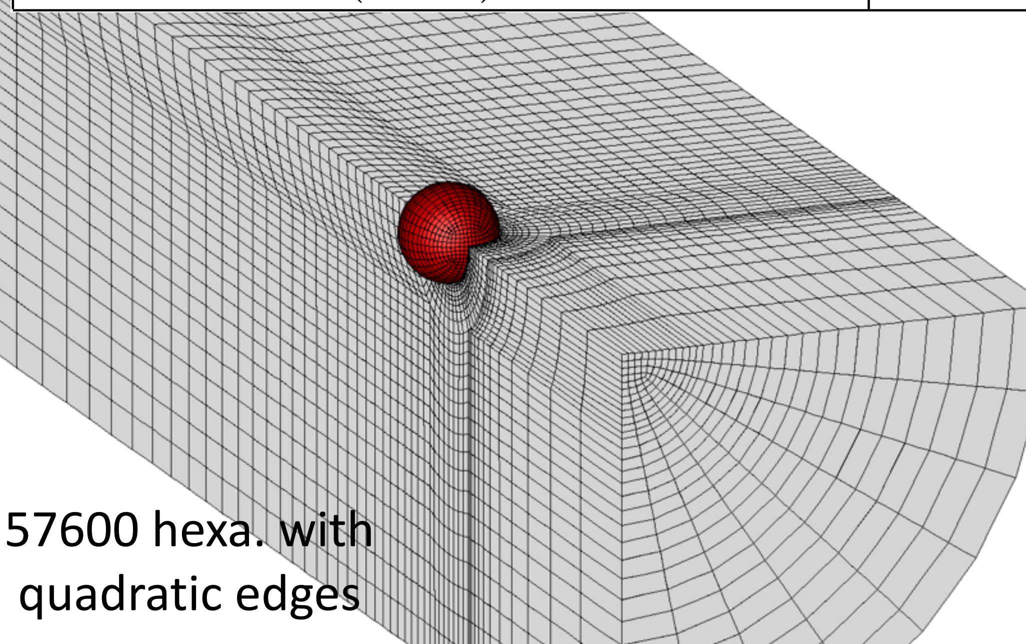
Geometry courtesy of Airbus Defence and Space

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

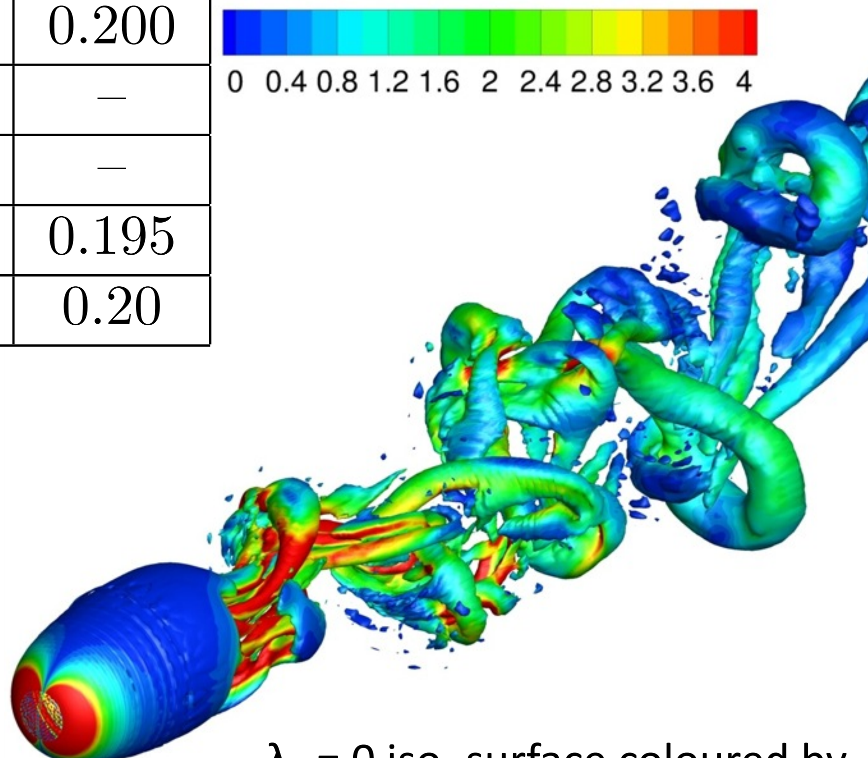
DNS of the incompressible flow past a sphere, Re=1000

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

	C_d	St
$\mathbb{P}^3$ (Iannelli-Baker $\Delta t = 4 \cdot 10^{-3}$)	0.471	0.201
$\mathbb{P}^4$ (ROS3P $\Delta t = 4 \cdot 10^{-3}$)	0.475	0.200
Hindenlang et al. (2012)	0.48	–
Ploumhans (2002)	0.48	–
Tomboulides and Orzsag (2000)	–	0.195
Poon et al. (2009)	0.46	0.20



57600 hexa. with quadratic edges



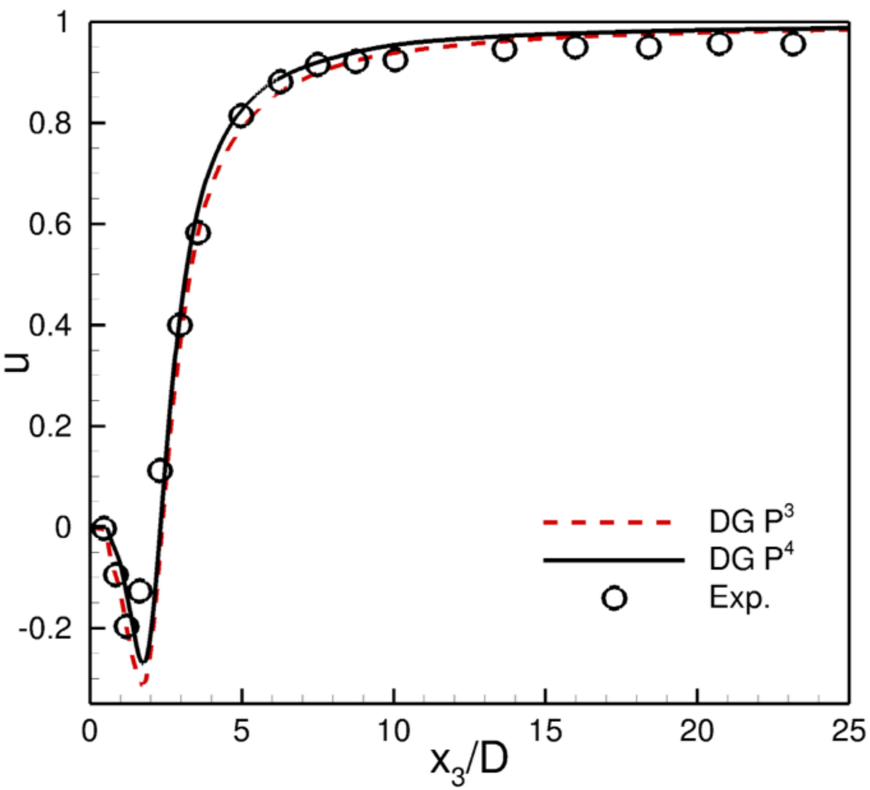
$\lambda_2 = 0$ iso–surface coloured by vorticity magnitude

(a ISCRA funded project)

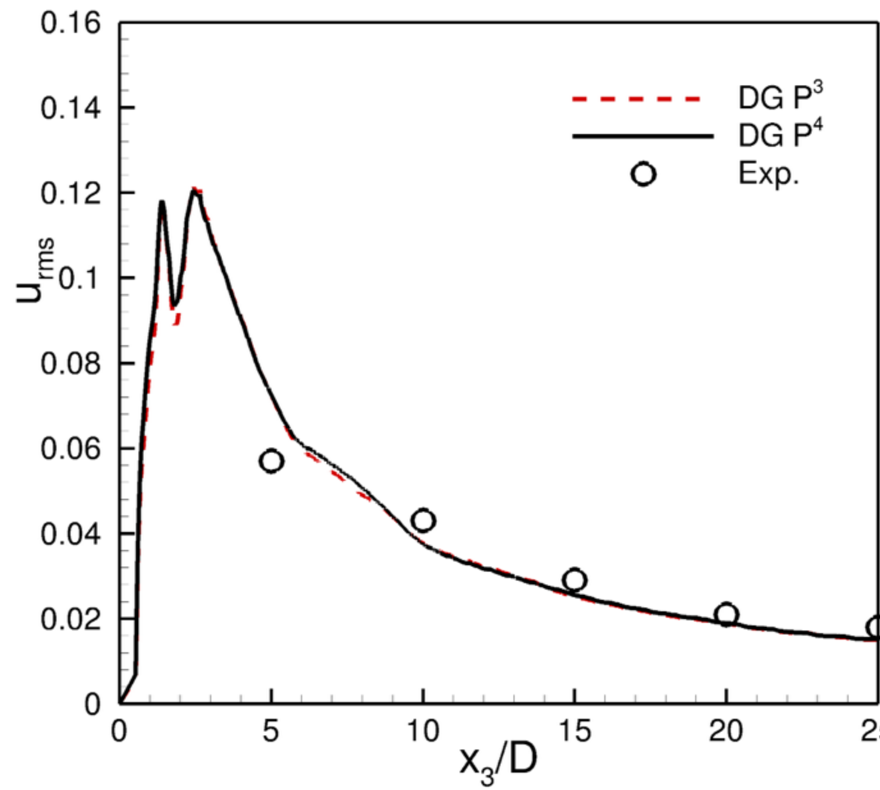
DNS of the incompressible flow past a sphere, Re=1000

UNIVERSITÀ DEGLI STUDI DI BERGAMO

DG $P^{3,4}$ solutions with second-order, two-stage [Iannelli and Baker, 1988] and third-order, three-stage ROS3P [Lang and Verwer, 2001]



averaged streamwise velocity



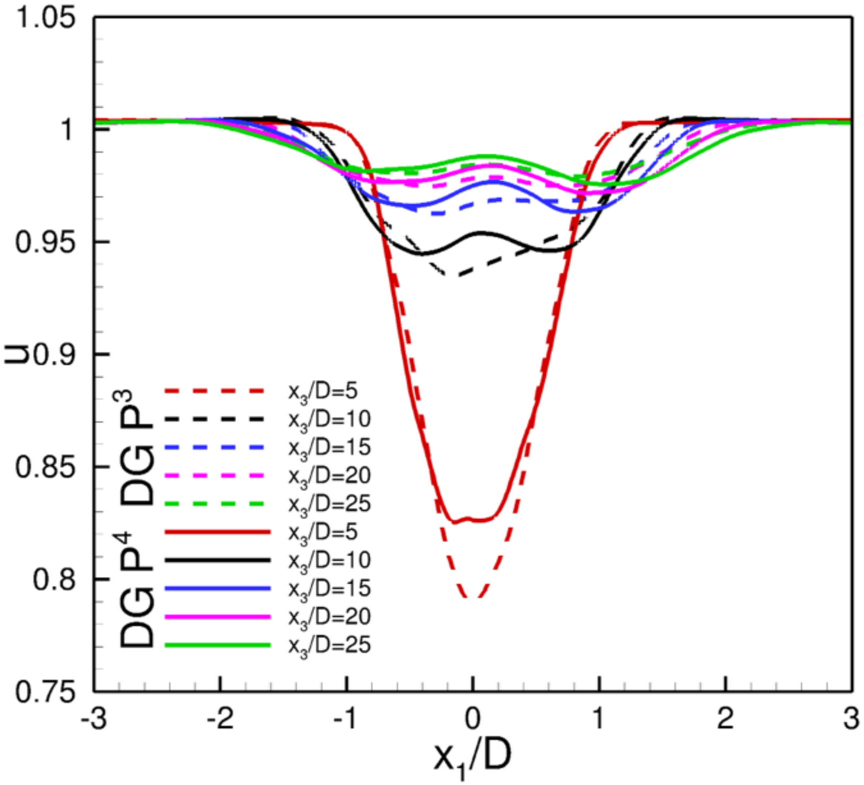
RMS streamwise velocity

DG $P^{3,4}$ DNS solutions along the wake (r/D=0)

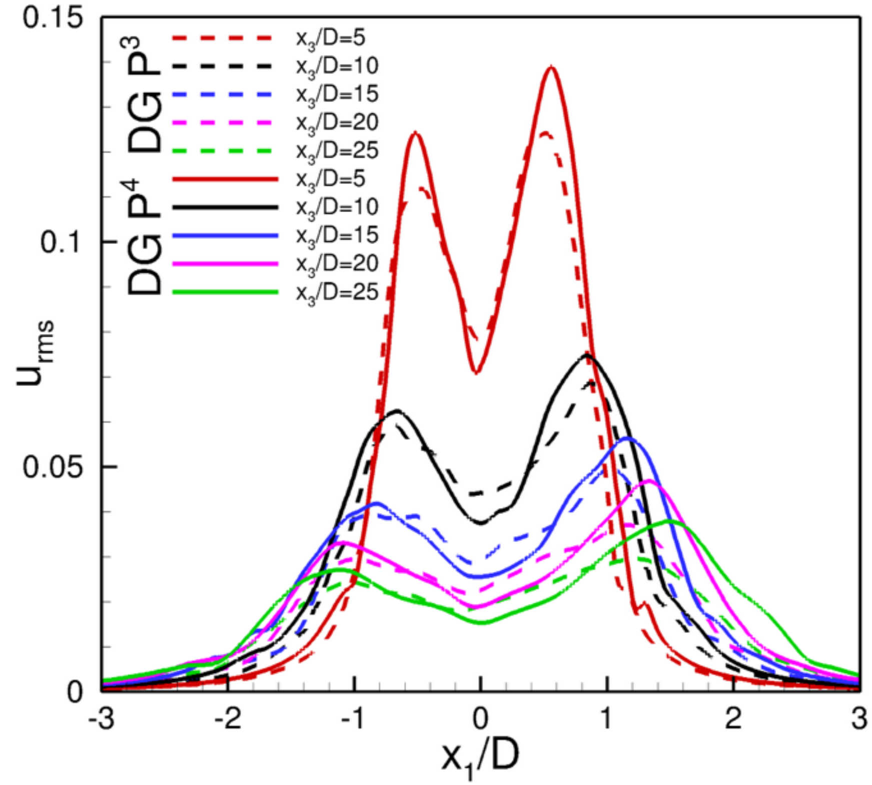
Experimental data for Re= 960 [Wu and Faeth, 1993]

DNS of the incompressible flow past a sphere, Re=1000

Reliable turbulent statistics should require longer simulations.
Here T = 130 and T = 100, for P^3 and P^4, respectively



averaged streamwise velocity



RMS streamwise velocity

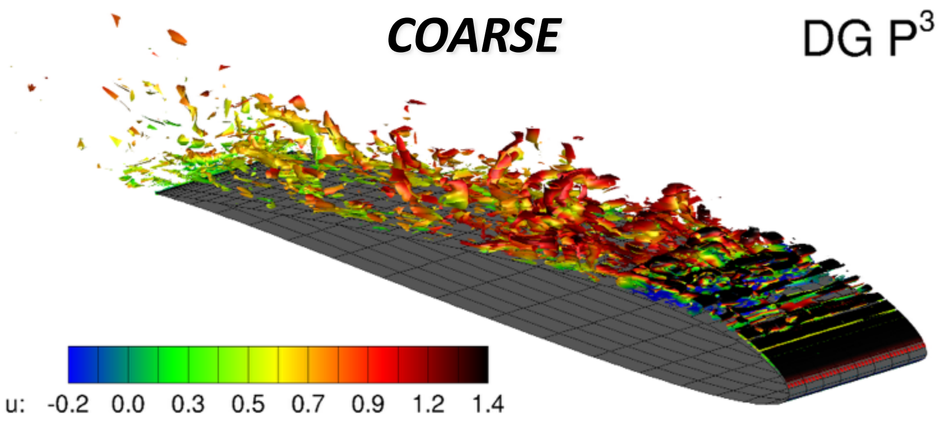
Although the averaged flow-field is expected to be axisymmetric, the P4 solution is in good agreement with [Tomboulides and Orszag, 2000]

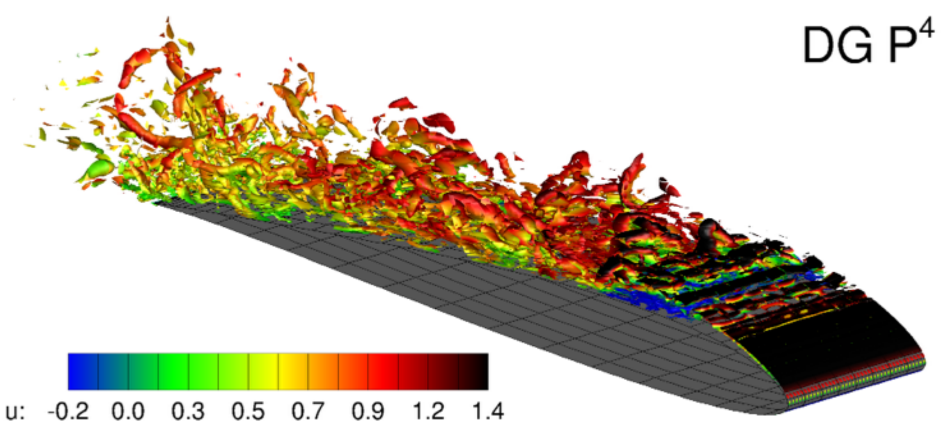
Implicit LES of the transitional flow around the SD7003 wing

UNIVERSITÀ DEGLI STUDI DI BERGAMO

ILES of a transitional compressible flow around a wing

$$M_\infty = 0.1, Re = 60000, \alpha = 8^\circ$$





Fifth-order, eight-stage RO5-8 Rosenbrock-type scheme [Di Marzo, 1993]

P^3 and P^4 solutions on a coarse mesh of 20054 hexa. with quartic edges

P2 and P3 solutions on a fine mesh of 160512 hexa. with quartic edges

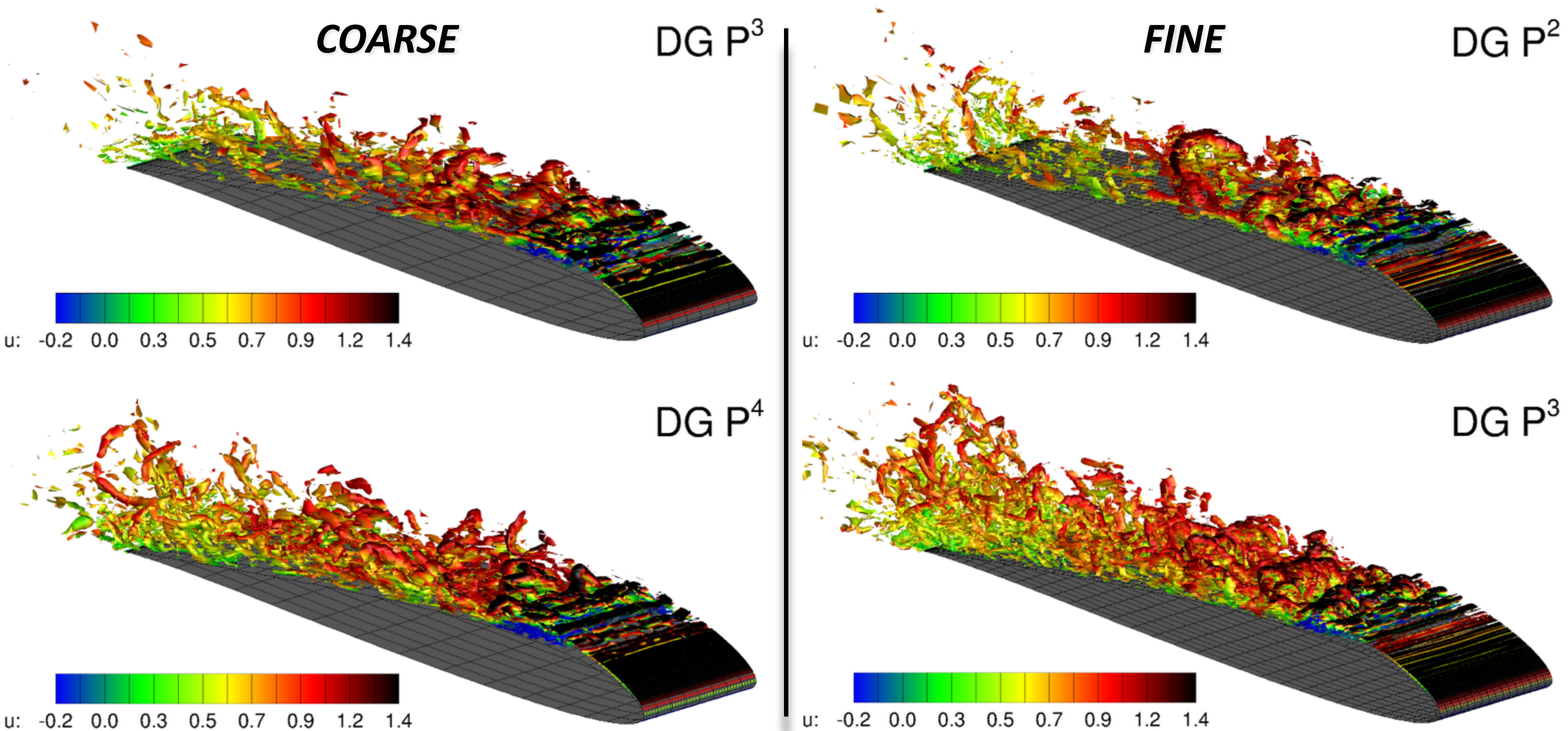
(a LISA funded project)

Implicit LES of the transitional flow around the SD7003 wing

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

ILES of a transitional compressible flow around a wing

$$M_\infty = 0.1, Re = 60000, \alpha = 8^\circ$$

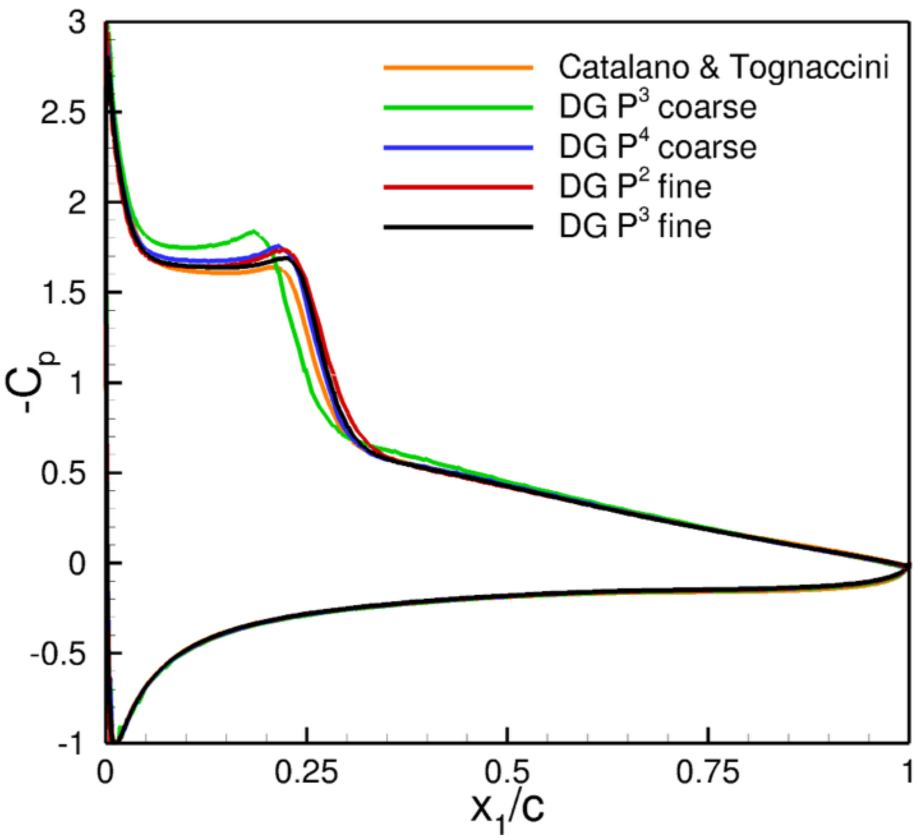


Implicit LES of the transitional flow around the SD7003 wing

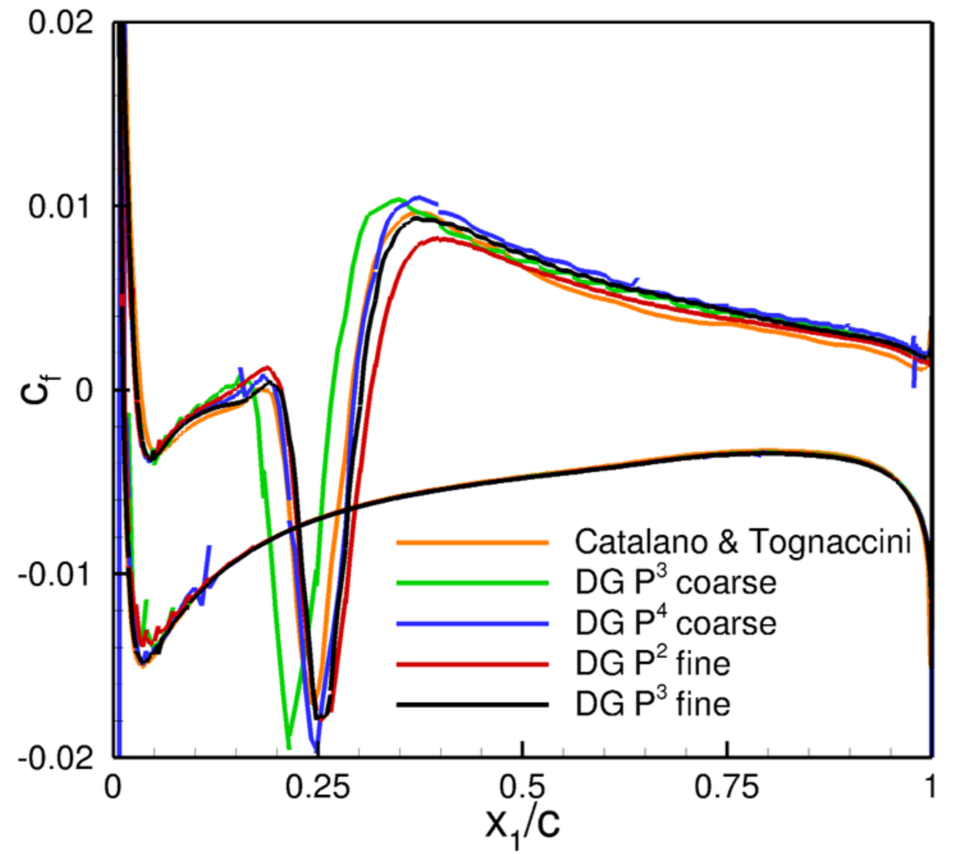
UNIVERSITÀ DEGLI STUDI DI BERGAMO

ILES of a transitional compressible flow around a wing

$$M_\infty = 0.1, Re = 60000, \alpha = 8^\circ$$



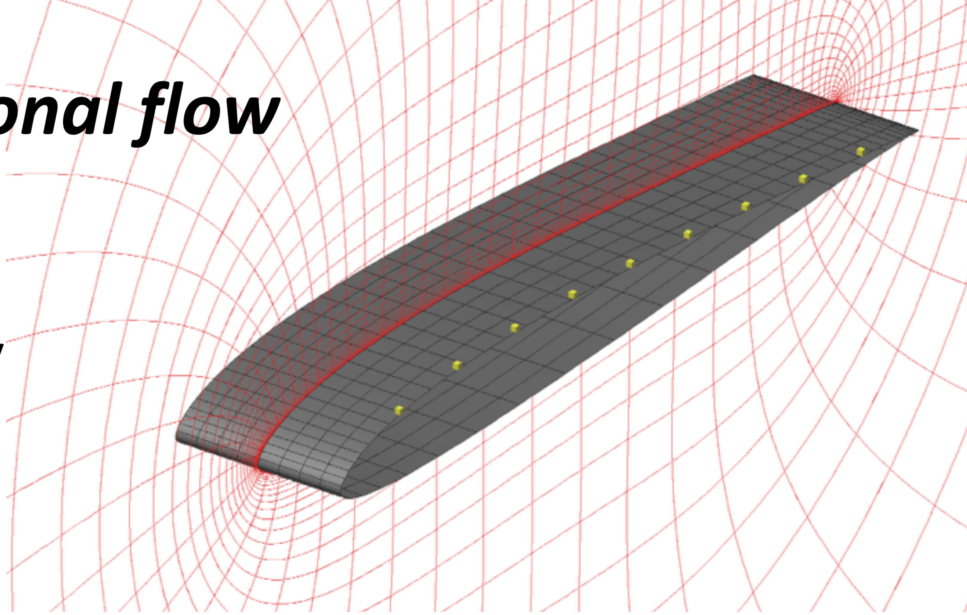
mean pressure coefficient



mean skin friction coefficient

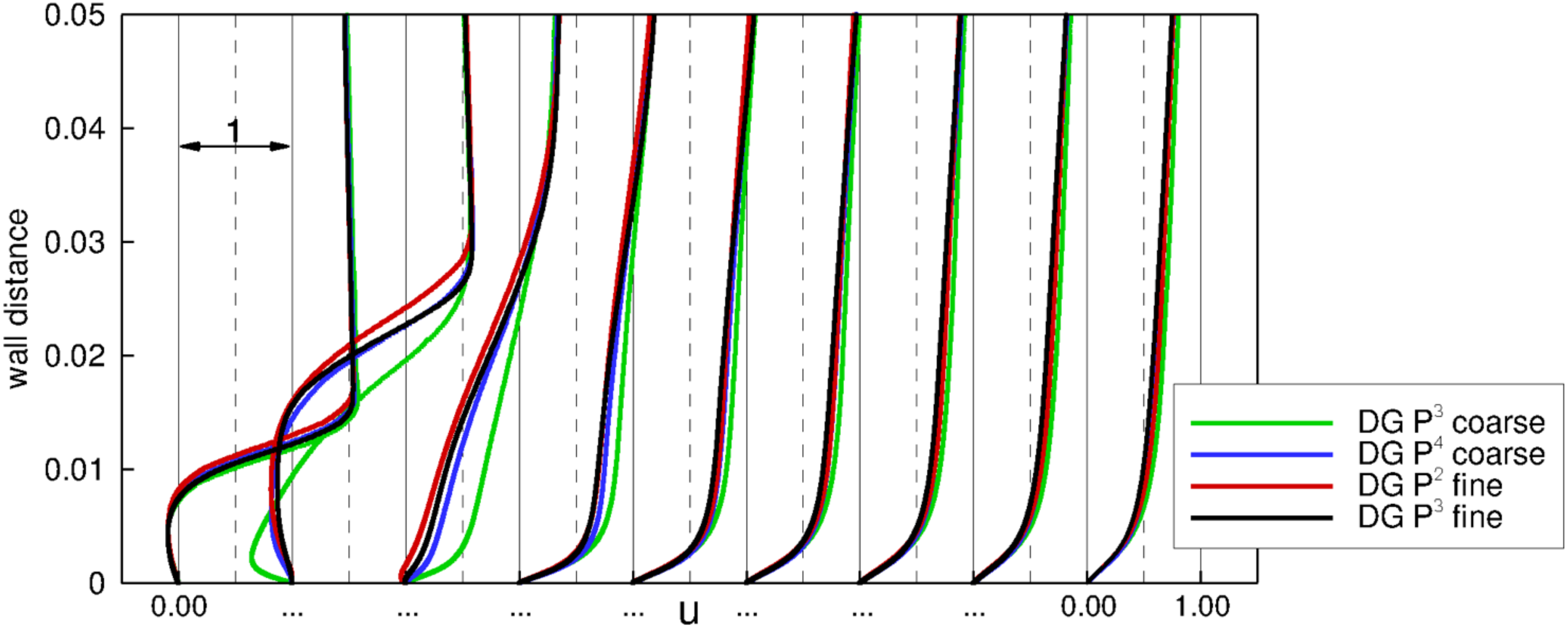
P. Catalano, R. Tognaccini, RANS analysis of the low-Reynolds number flow around the SD7003 airfoil, Aerosp. Sci. Technol. 15 (8) (2011) 615–626

Implicit LES of the transitional flow around the SD7003 wing

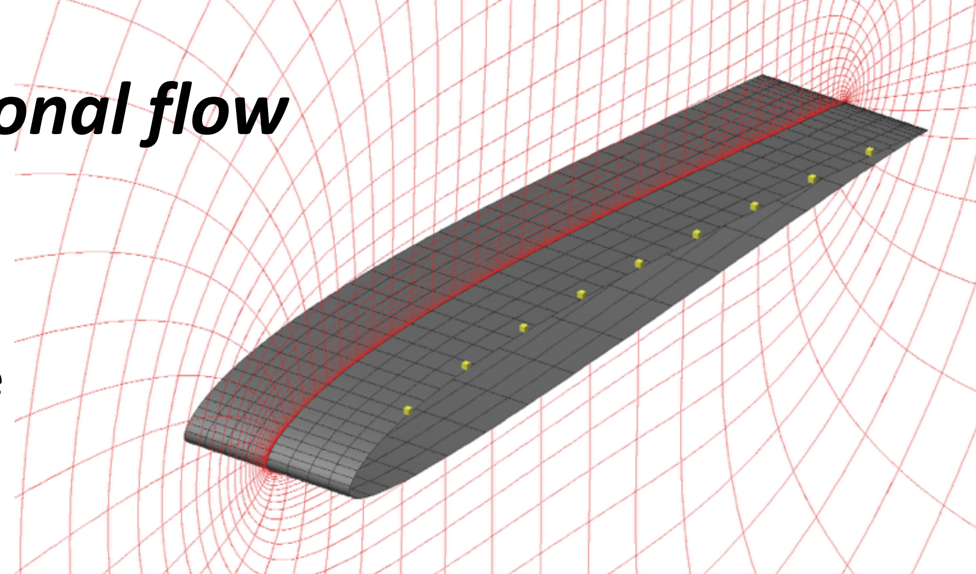


Profiles of mean x-component of velocity at chordwise locations

x/c ={0.1, ..., 0.9}



Implicit LES of the transitional flow around the SD7003 wing



Details of the laminar separation bubble and mean aerodynamic loads

	DOFs	LSB details				Aerodynamic loads		
		x_s/c	x_r/c	L/c	H/c	C_D	C_L	C_m
RK5-8–DG $\mathbb{P}^3$ coarse	401280	0.027	0.268	0.241	0.017	0.0423	0.9615	-0.0233
RK5-8–DG $\mathbb{P}^4$ coarse	702240	0.027	0.294	0.267	0.021	0.0454	0.9534	-0.0224
RK5-8–DG $\mathbb{P}^2$ fine	1605120	0.027	0.314	0.287	0.022	0.0471	0.9548	-0.0235
RK5-8–DG $\mathbb{P}^3$ fine	3210240	0.028	0.303	0.275	0.021	0.0457	0.9441	-0.0223
DGSEM $\mathbb{P}^3$ [BB]	4.26M	0.027	0.310	-	-	0.045	0.923	-
DGSEM $\mathbb{P}^7$ [BB]	4.55M	0.030	0.336	-	-	0.050	0.932	-
Comp. FD $\mathcal{O}(6)$ [GV]	53.4M	0.031	0.303	0.272	0.020	0.0447	0.917	-0.0187
SBP-SAT $\mathcal{O}(4)$ [BZ]	4.48M	0.037	0.200	-	-	0.034	0.968	-

x_s and x_r are the separation and reattachment points coordinates, L and H the separation bubble length and height

[BZ] P.D. Boom, D.W. Zingg, Time-accurate flow simulations using an efficient Newton–Krylov–Schur approach with high-order temporal and spatial discretization, AIAA Paper 2012-0383, 2013

[BB] T. Bolemann, A. Beck, D. Flad, H. Frank, V. Mayer, C.-D. Munz, High-order discontinuous Galerkin schemes for large-eddy simulations of moderate Reynolds number flows, in: N. Kroll, C. Hirsch, F. Bassi, C. Johnston, K. Hillewaert (Eds.), IDIHOM: Industrialization of High-Order Methods— A Top-Down Approach, in: Notes on Numerical Fluid Mechanics and Multidisciplinary Design, vol. 128, Springer International Publishing, 2015, pp. 435–456

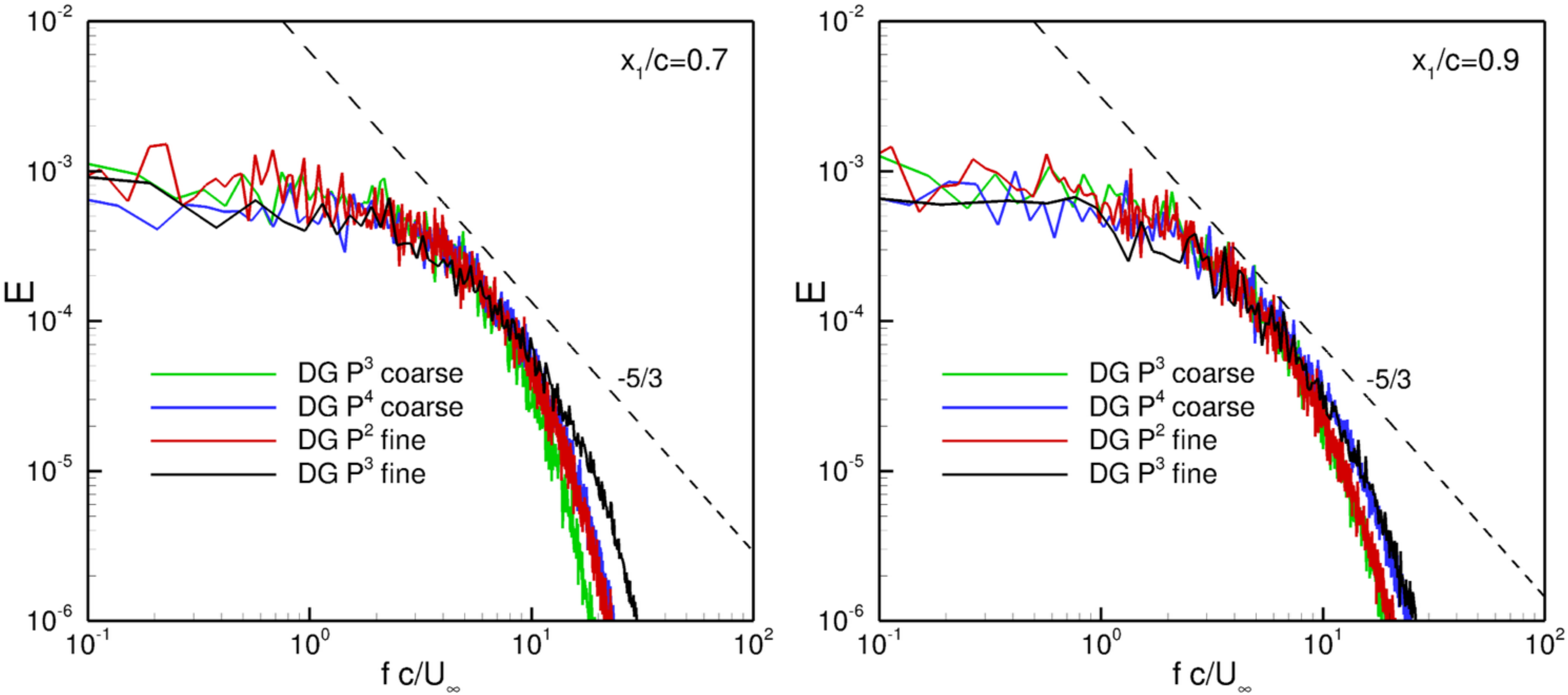
[GV] D.J. Garmann, M.R. Visbal, C3.3: Implicit large eddy-simulations of transitional flow over the SD7003 airfoil using compact finite-differencing and filtering, in: 2nd International Workshop on High-Order CFD Methods, Cologne, Germany, 2013

Implicit LES of the transitional flow around the SD7003 wing

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

ILES of a transitional compressible flow around a wing

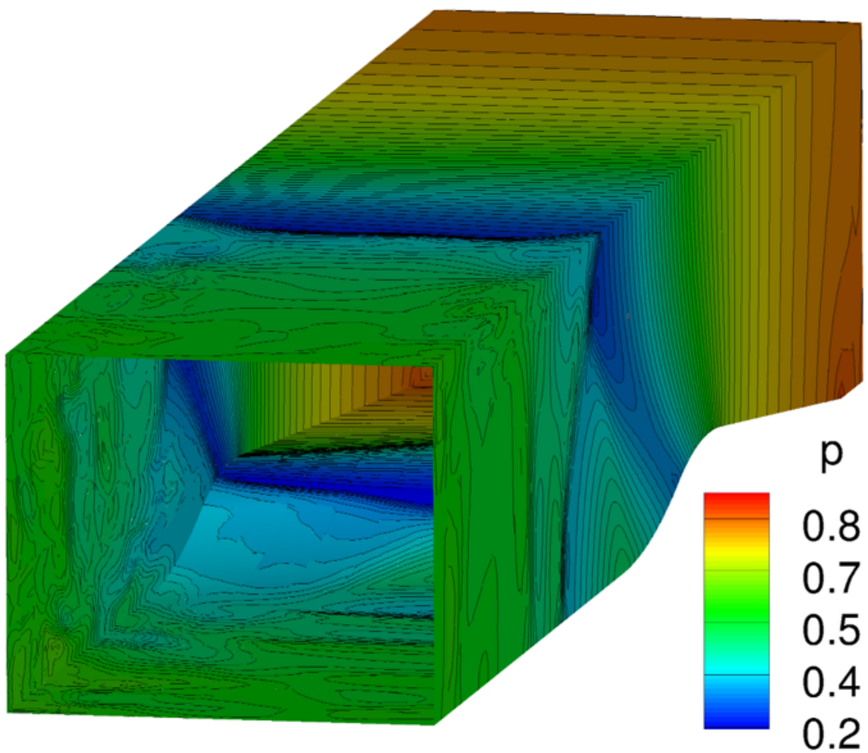
$M_\infty = 0.1, Re = 60000, \alpha = 8^\circ$



Energy spectra at chordwise locations *x/c={0.7, 0.9}*, $E = \frac{1}{n_i}\sum_{i=1}^{n_i}\left(\frac{1}{2}\left(PSD(u)+PSD(v)+PSD(w)\right)\right)$

X-LES of a shock BL interaction on a swept bump

P^2 converged computations with RANS+k-ω (also in its low-Re version) and EARSM1 have been performed and used as initialization for X-LES



72960 hexahedral elements with quadratic edges

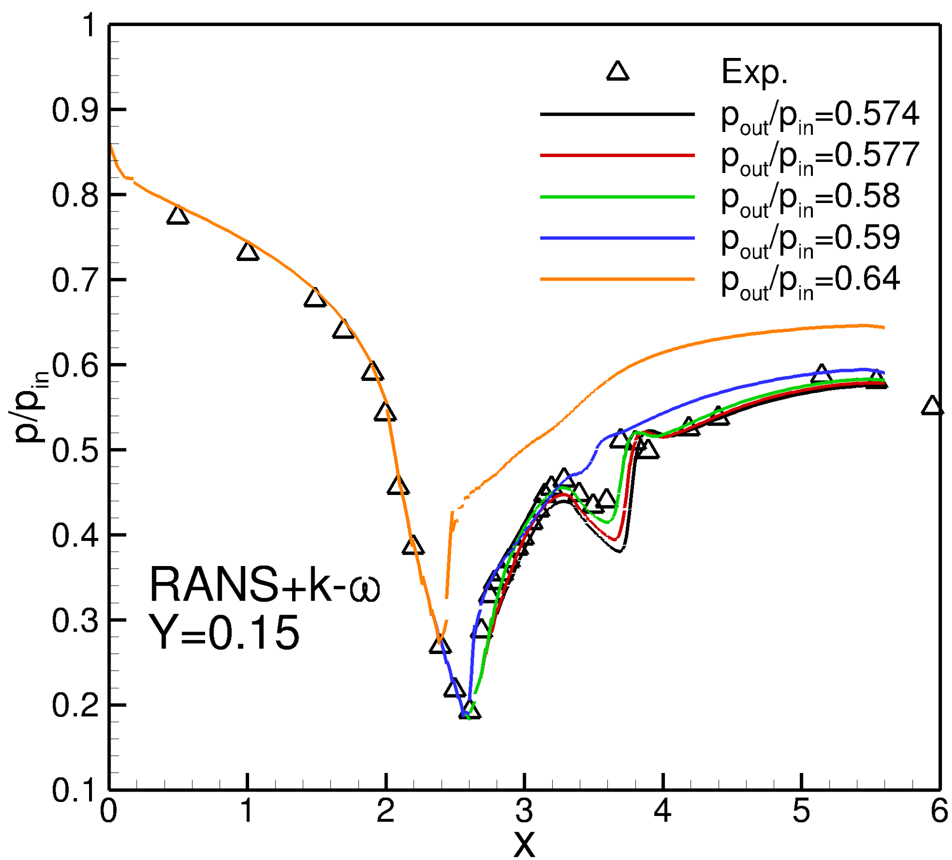
- Inlet boundary conditions
  - p_{0i} = 92000Pa
  - T_{0i} = 300K
  - $Re_H = 1.69 \times 10^6$
- Outlet static pressure used to impose the shock position *(model dependent!)*
- LBE to quickly find the "right" pressure ratio
- RO3-3 for the time-accurate solution
- Filter width Δ = 5e-2 *(strong influence!)*
- P^3 computations underway on *MARCONI@CINECA (a LISA funded project)*

X-LES of a shock BL interaction on a swept bump

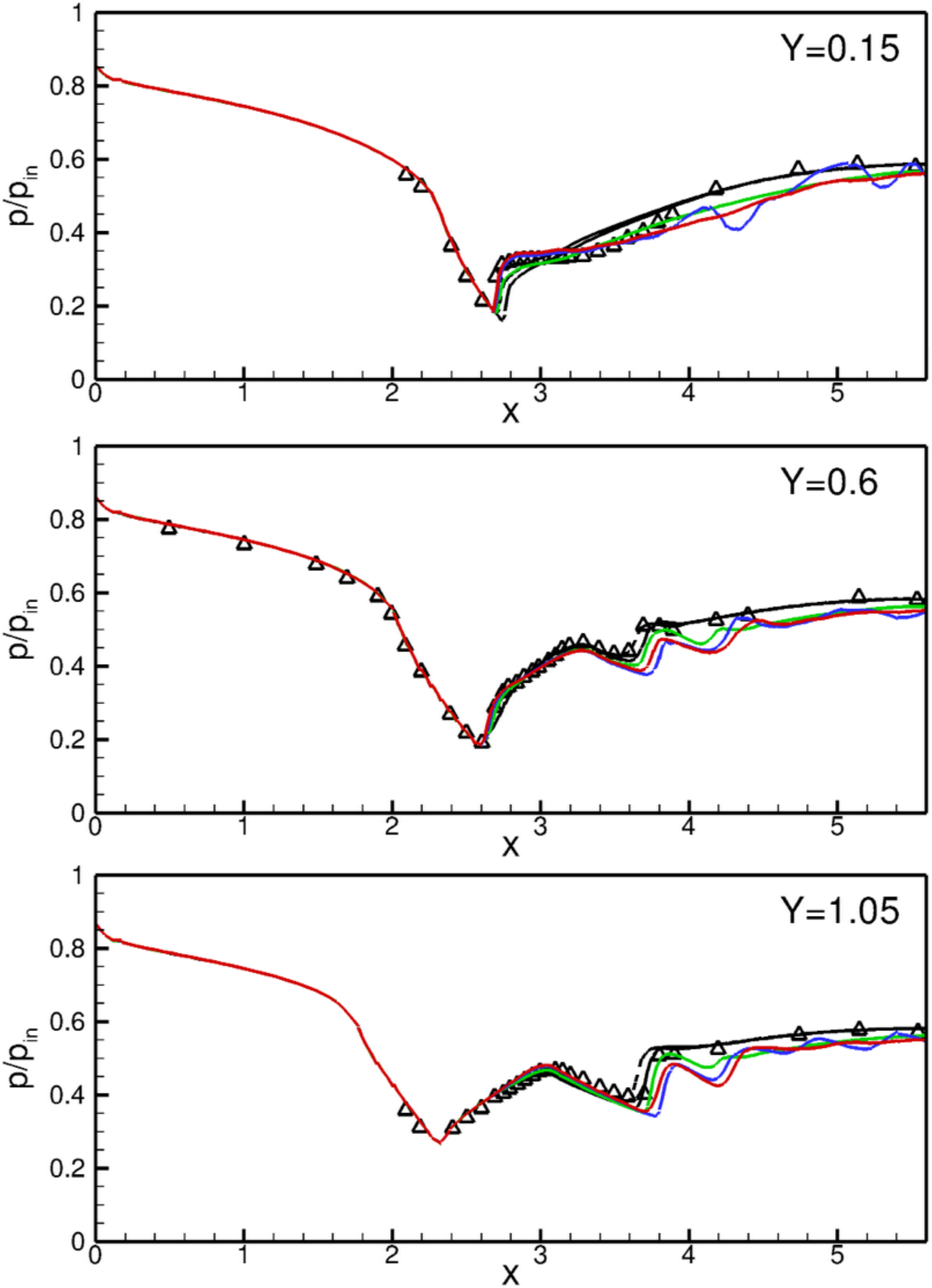
UNIVERSITÀ DEGLI STUDI DI BERGAMO

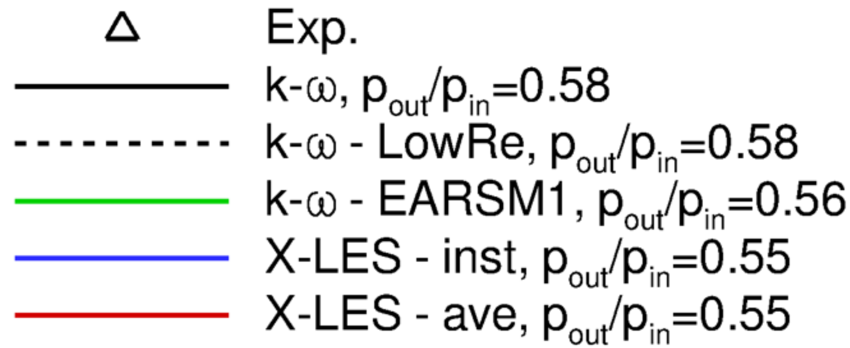
TiLDA

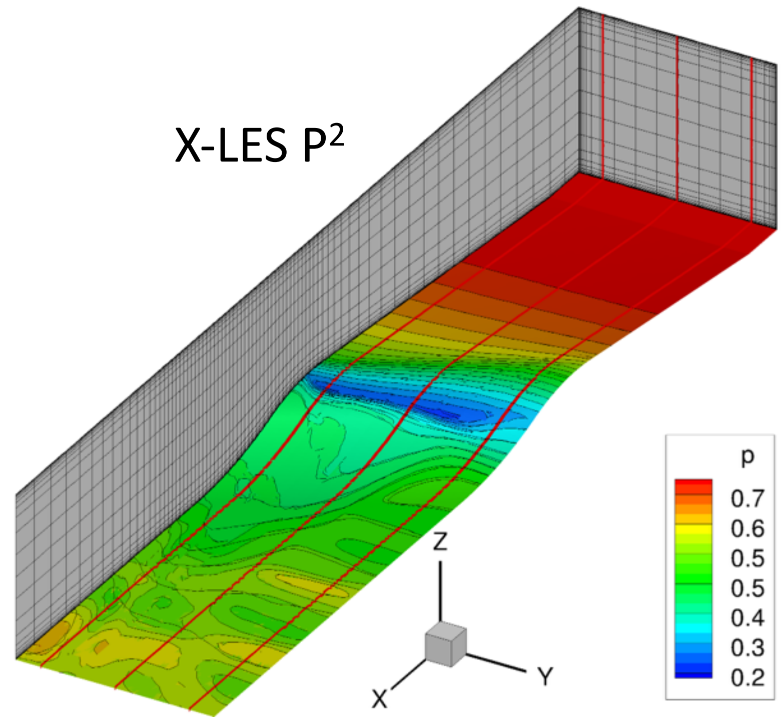
P^2 converged computations with RANS+k-ω (also in its low-Re version) and EARSM1 have been performed and used as initialization for X-LES



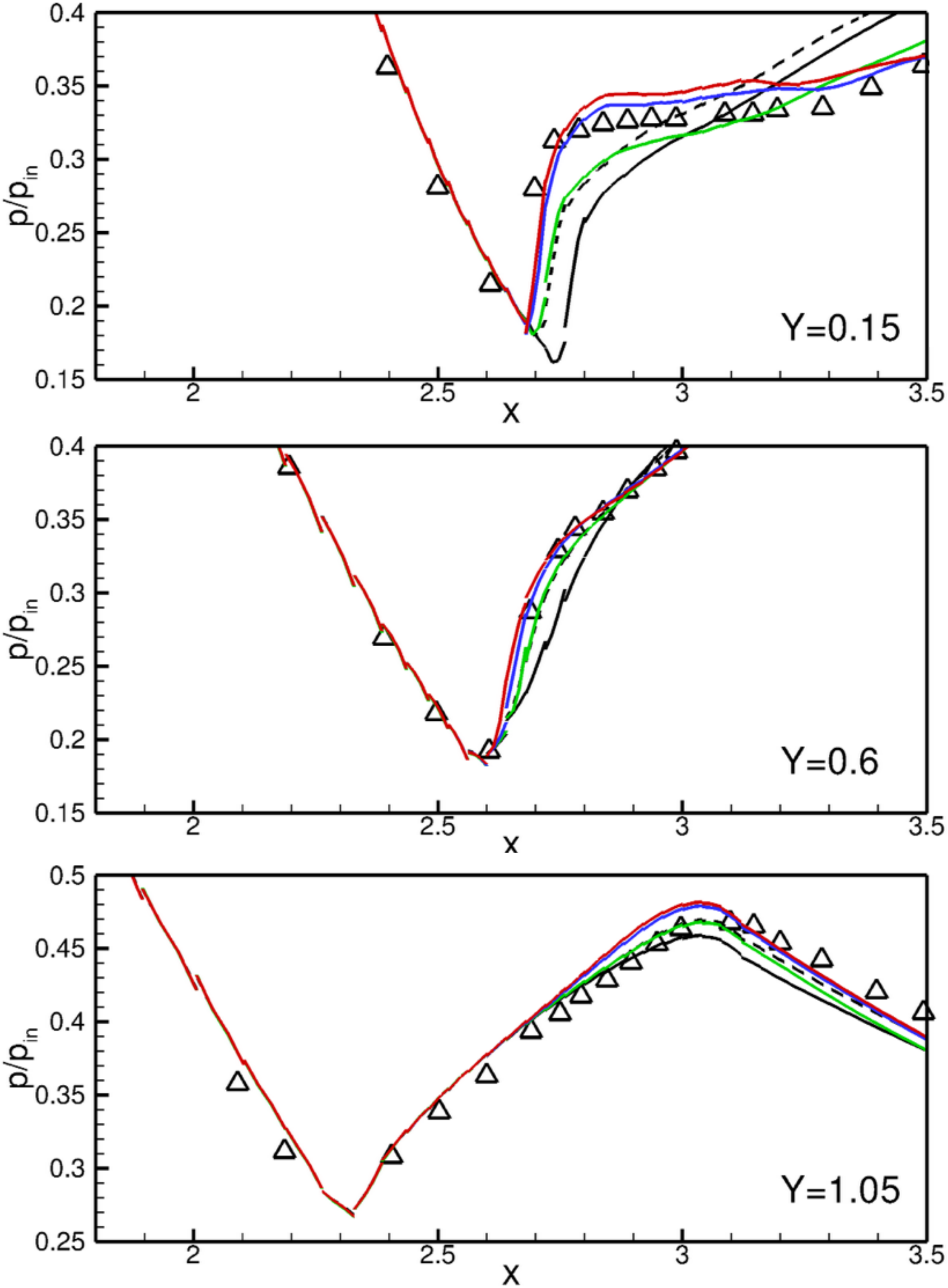
- Inlet boundary conditions
  - p_{0i} = 92000Pa
  - T_{0i} = 300K
  - Re_H = 1.69 x 10^6
- Outlet static pressure used to impose the shock position *(model dependent!)*
- LBE to quickly find the "right" pressure ratic
- RO3-3 for the time-accurate solution
- Filter width Δ = 5e-2 *(strong influence!)*
- P^3 computations underway on *MARCONI@CINECA (a LISA funded project)*

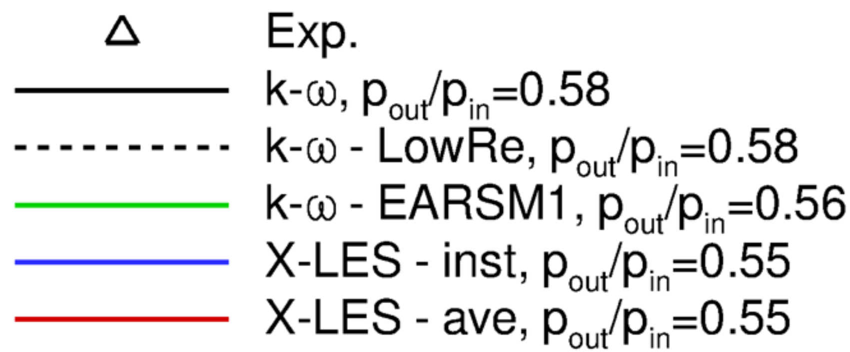


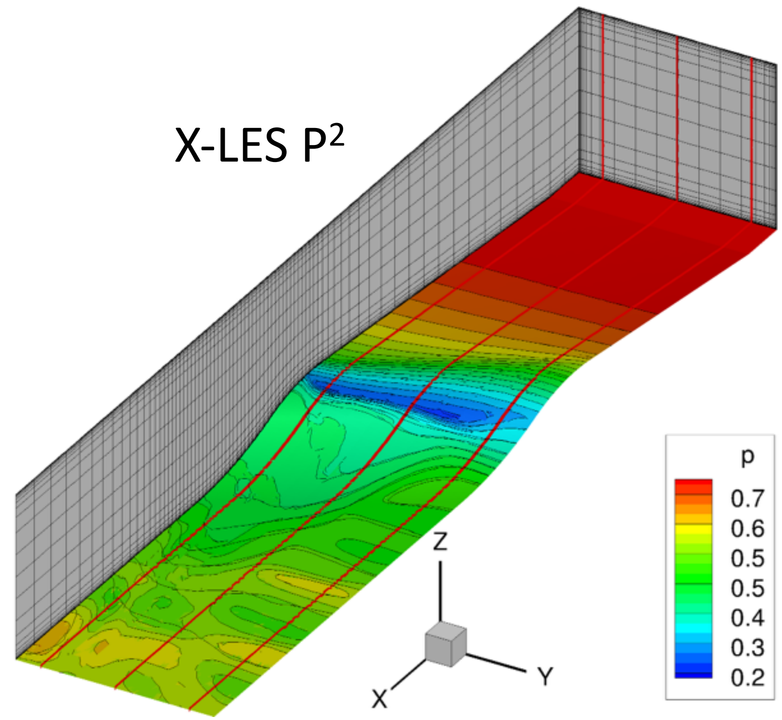




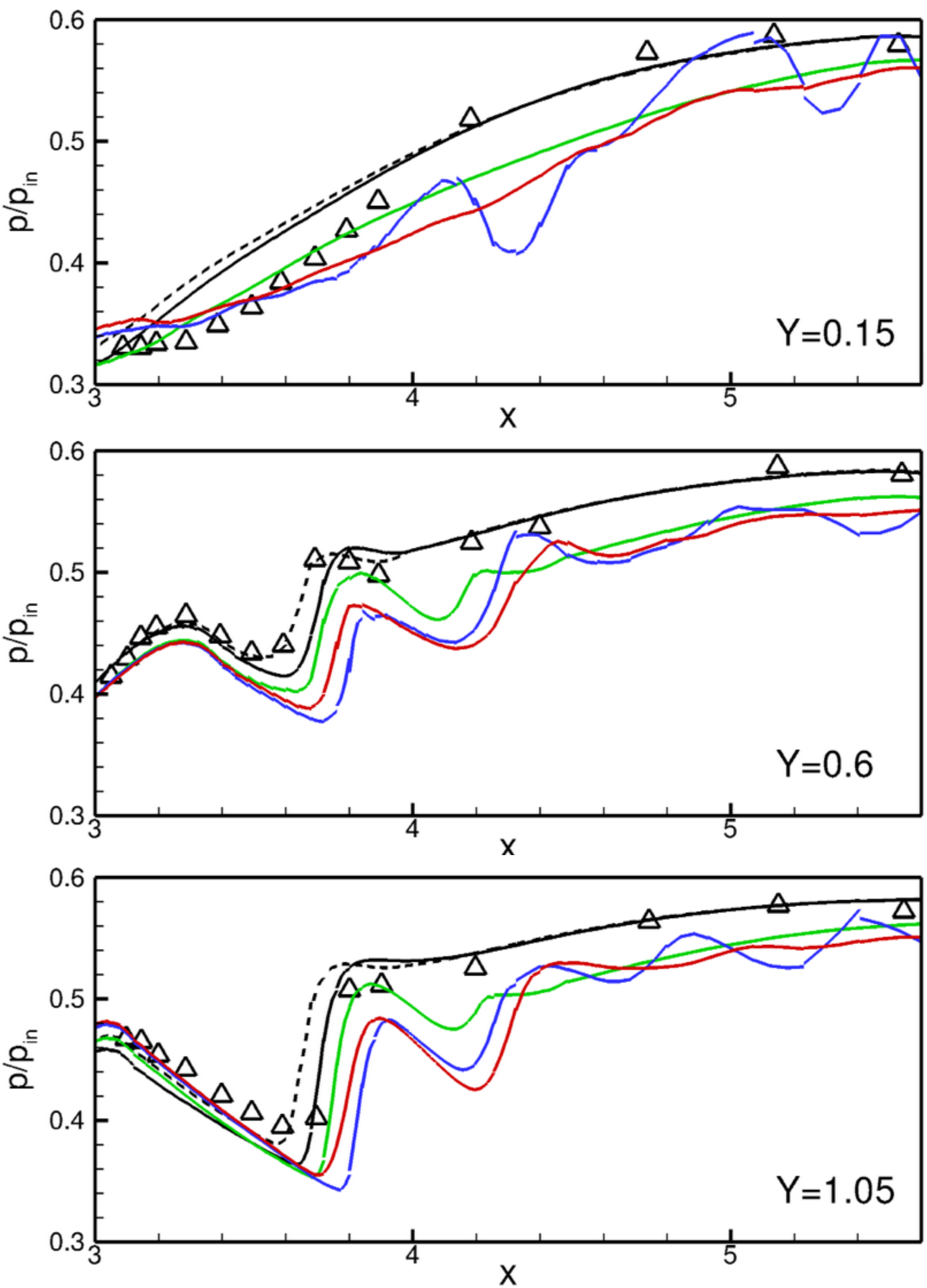
The profiles are in reasonable agreement with the experiments and the numerical results of Cahen et al.

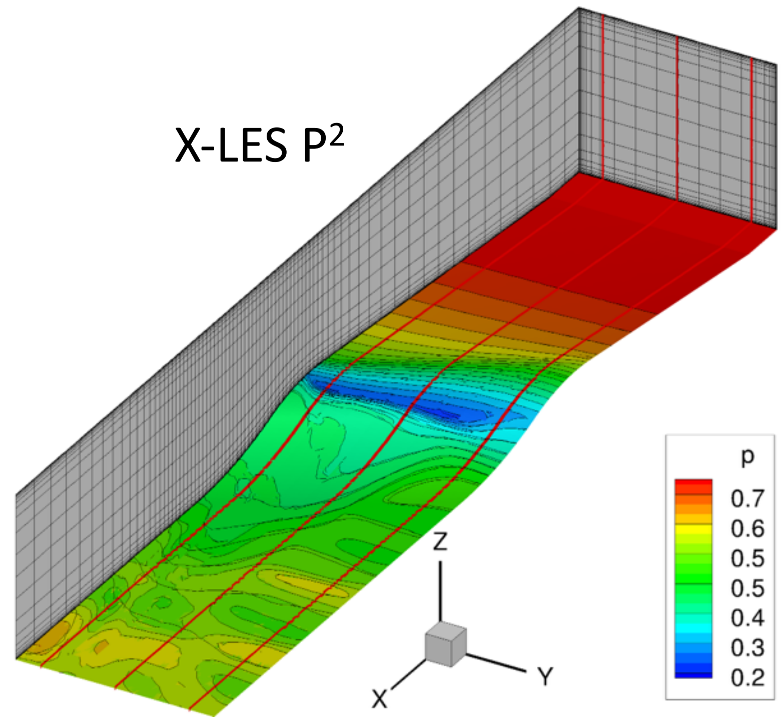




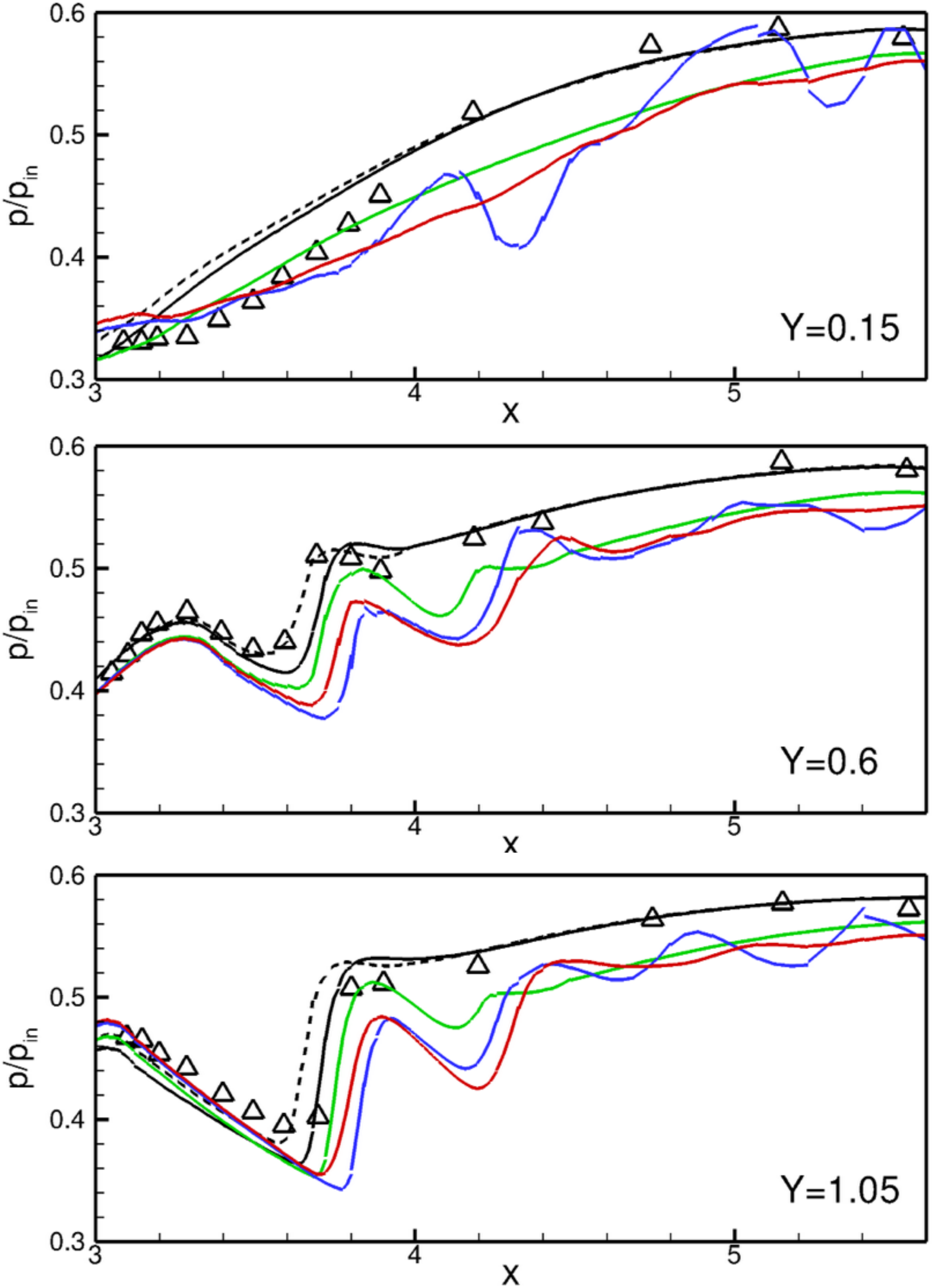


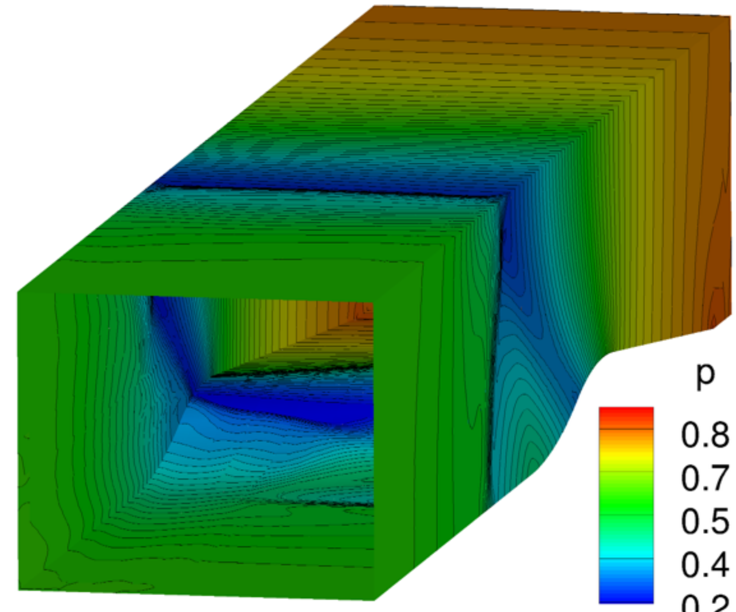
The profiles are in reasonable agreement with the experiments and the numerical results of Cahen et al.



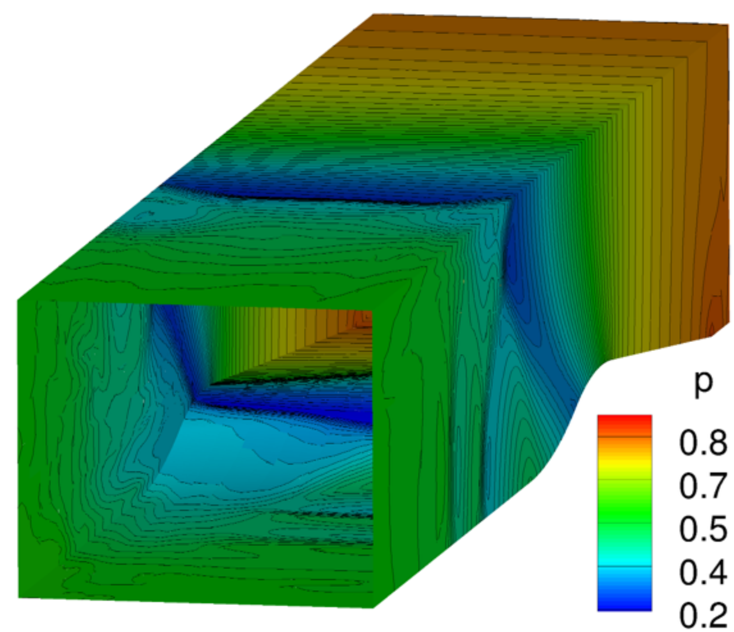


The profiles are in reasonable agreement with the experiments and the numerical results of Cahen et al.





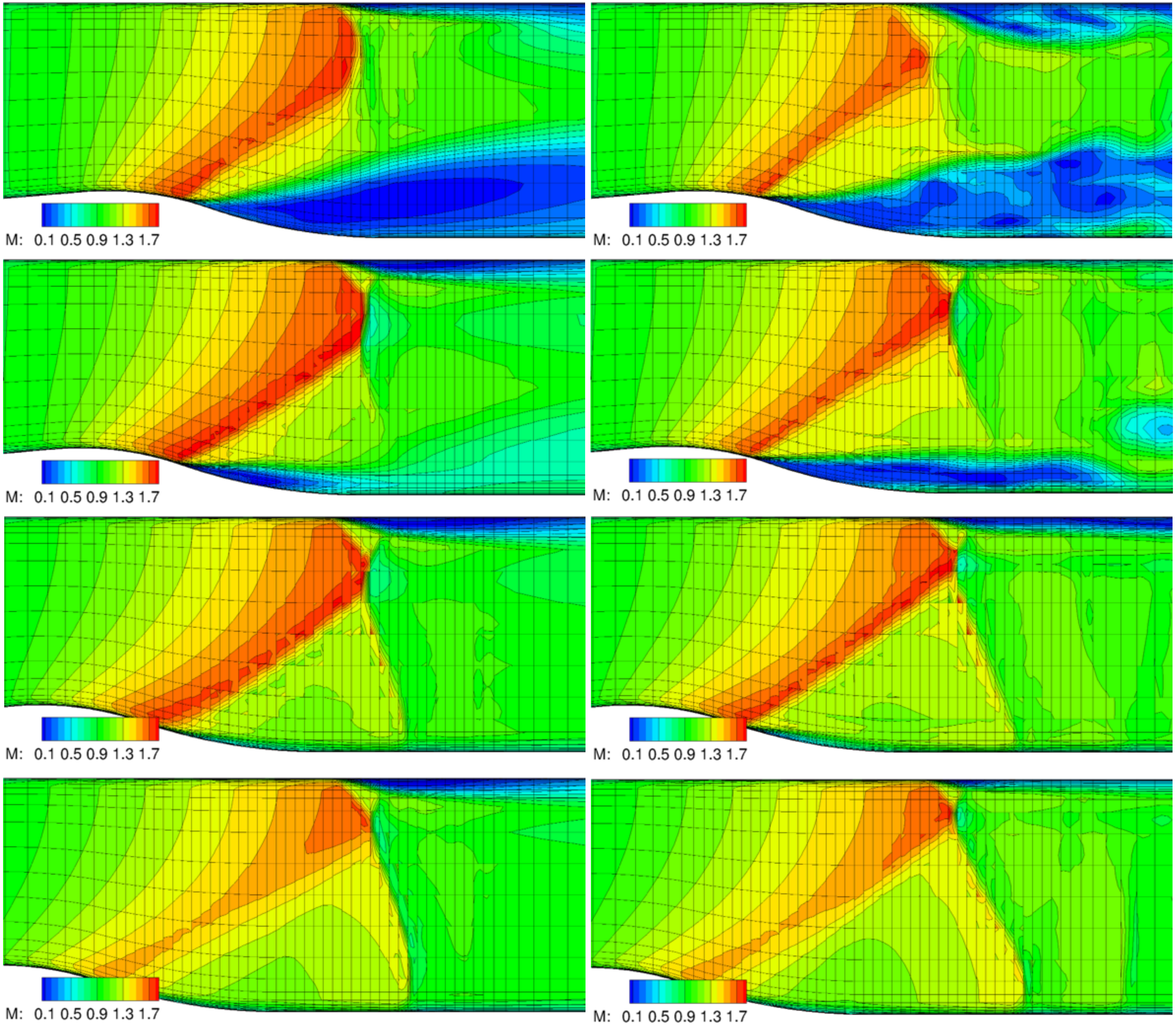
$k-\omega$



X – LES ave., RO3–3

TILDA

UNIVERSITÀ DEGLI STUDI
DI BERGAMO



y=0.1

y=0.3

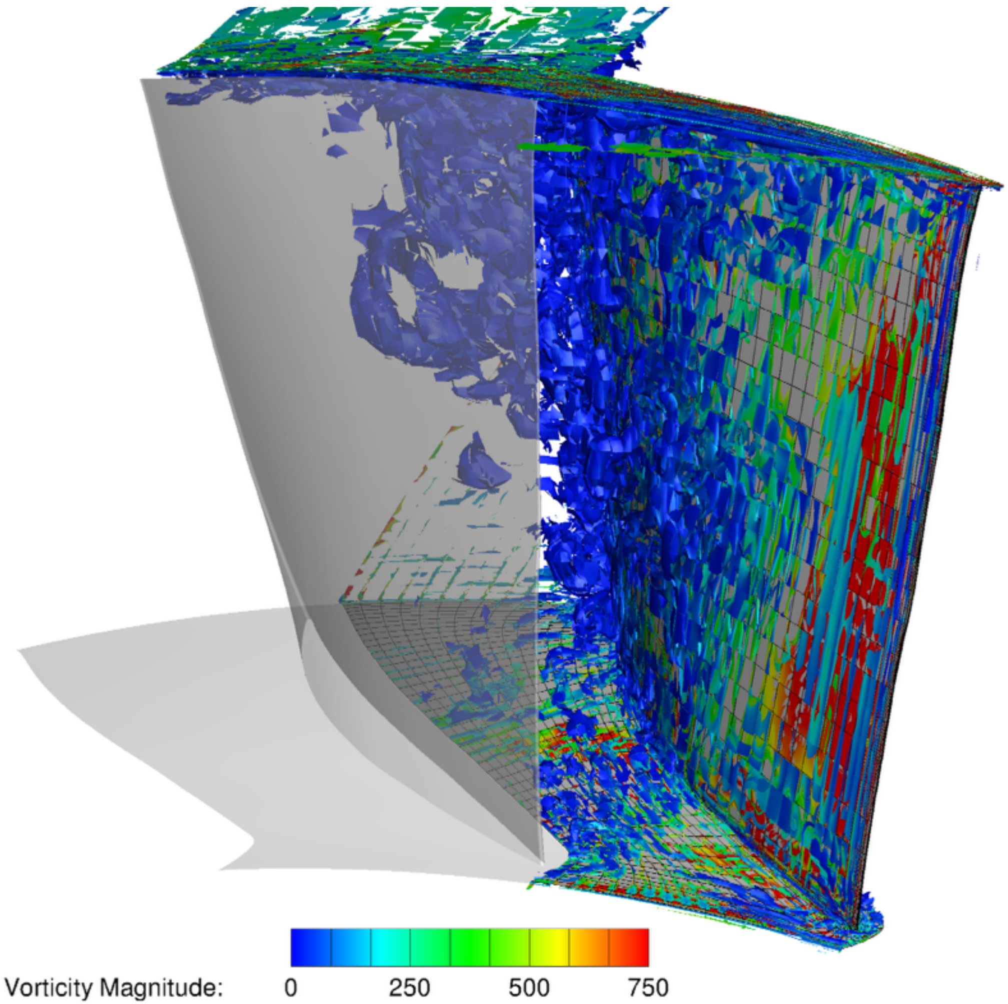
y=0.6

y=0.9

$k\!-\!\omega$ $\quad$ $\mathrm{X-LES\ inst.,\ RO3\!-\!3}$

X-LES of the transonic flow field in the NASA Rotor 37

TILDA
UNIVERSITÀ DEGLI STUDI DI BERGAMO



iso-surface of Q-criterion
160512 hexahedral elements

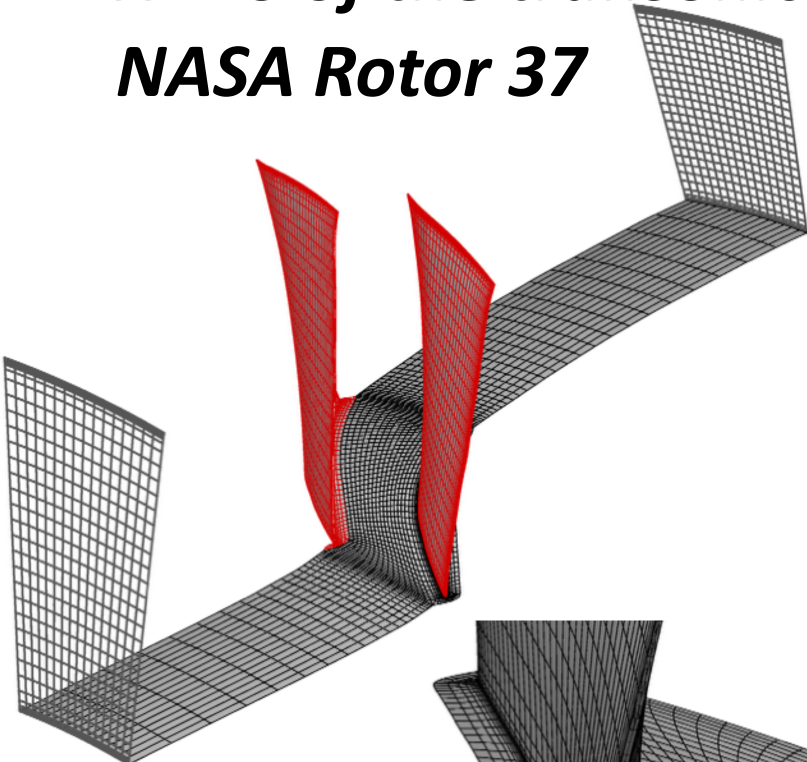
- P^2 computation using RO3-3
- Filter width Δ=5e-5
- Boundary conditions
  - p_{01} = 101325Pa
  - T_{01} = 288K
  - ω= 1800rad/s
  - Tu_1 = 3%
  - α_1 = 0°
- P^3 computations underway on *MARCONI@CINECA*

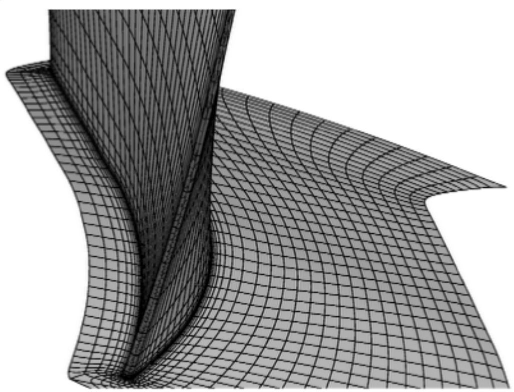
According to our first experiences, for a practical usage of X-LES, initializing with RANS seems mandatory

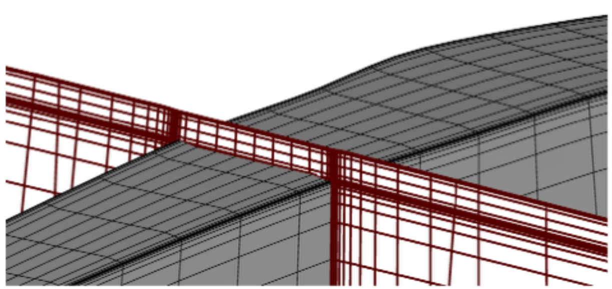
X-LES of the transonic flow field in the NASA Rotor 37

UNIVERSITÀ DEGLI STUDI DI BERGAMO

TILDA







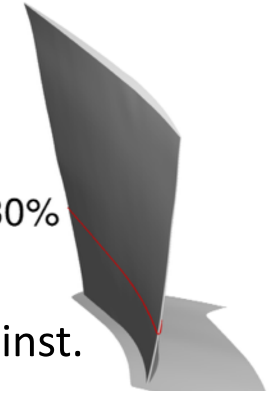
160512 hexahedral elements with parabolic edges

- P^2 computation using RO3-3
- Filter width Δ=5e-5
- Boundary conditions
  - p_{01} = 101325Pa
  - T_{01} = 288K
  - ω= 1800rad/s
  - Tu_1 = 3%
  - α_1 = 0°

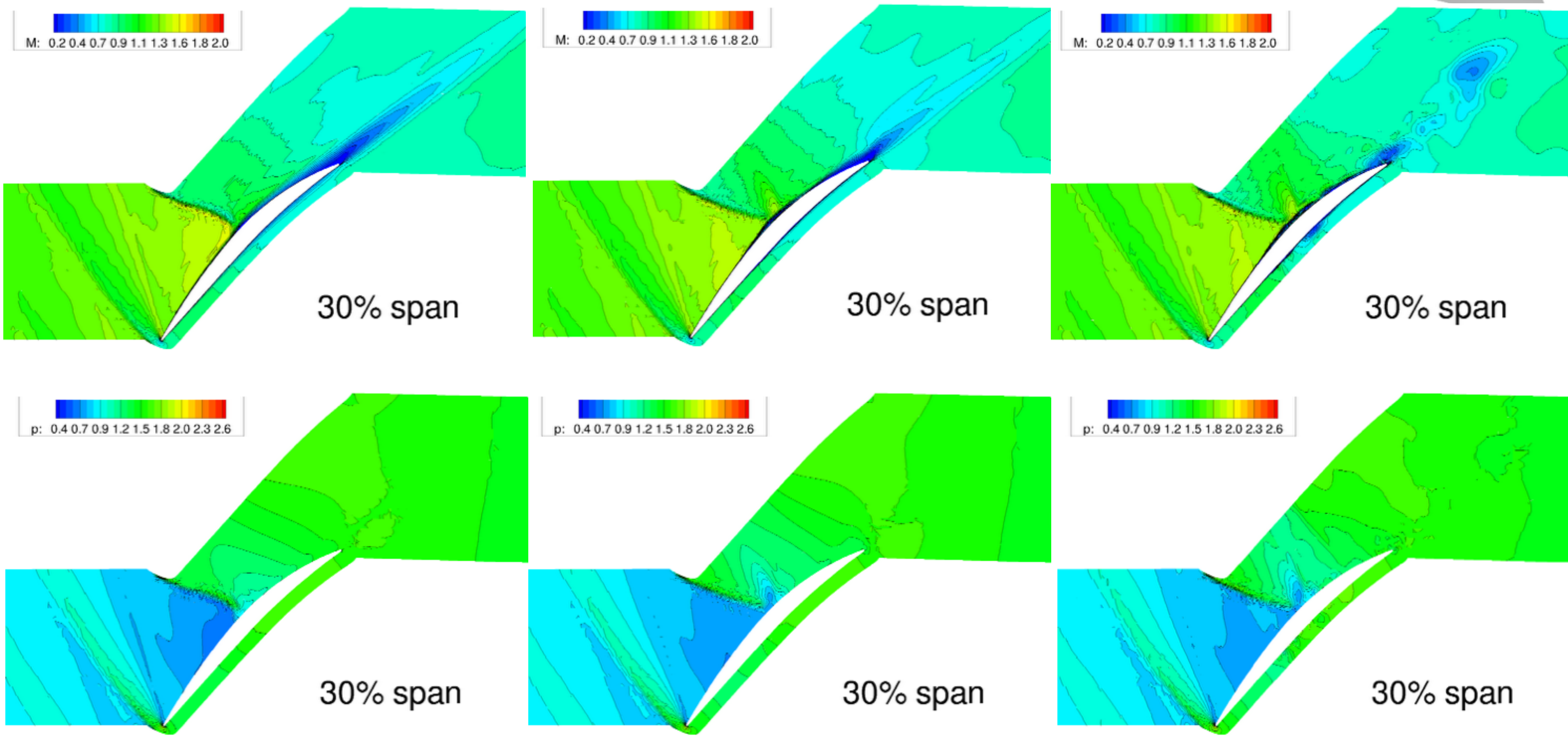
P^3 computations underway on *MARCONI@CINECA*

According to our first experiences, for a practical usage of X-LES, initializing with RANS seems mandatory

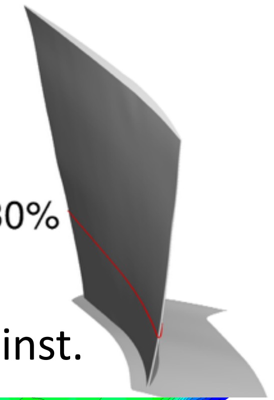
X-LES of the transonic flow field in the NASA Rotor 37



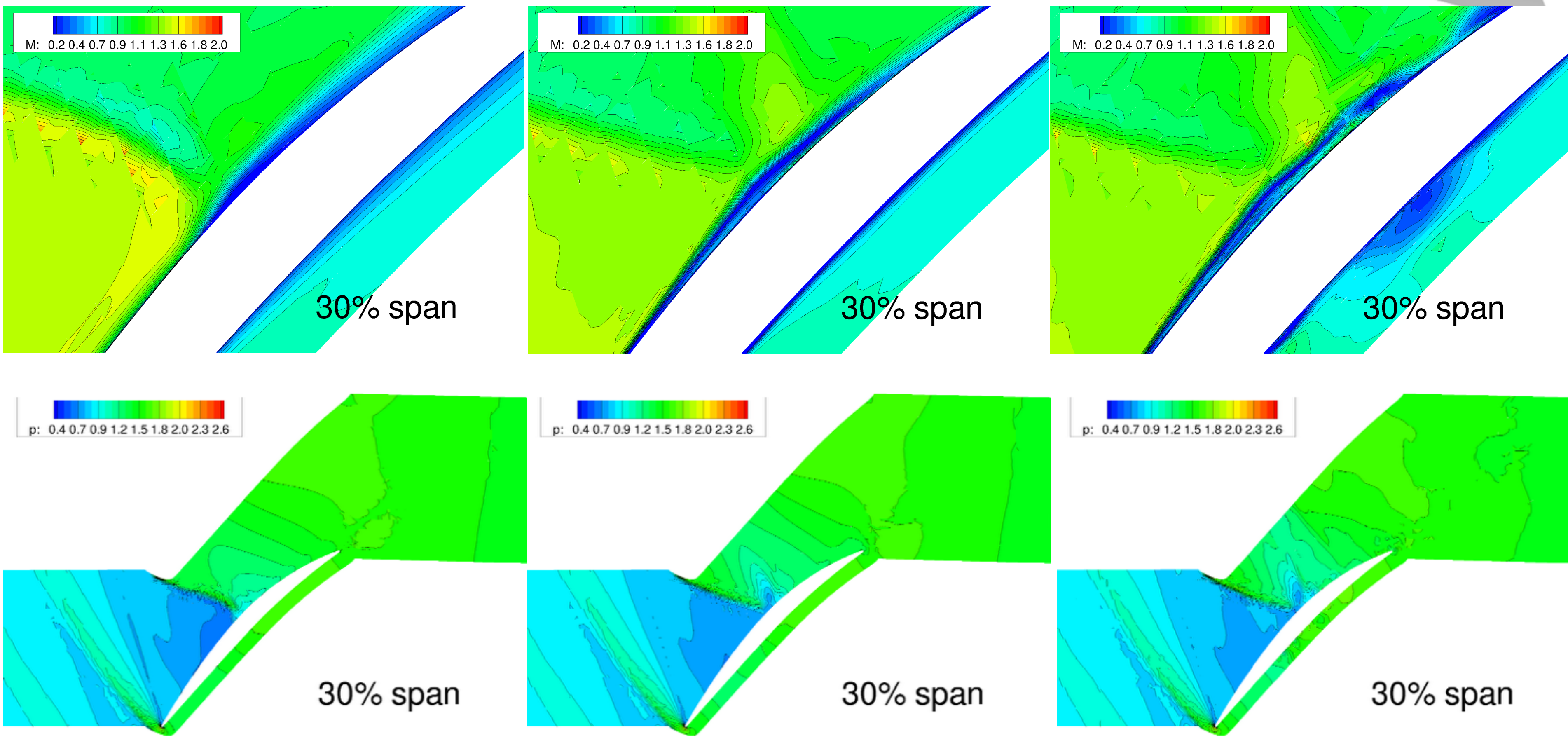
RANS X-LES ave. X-LES inst.



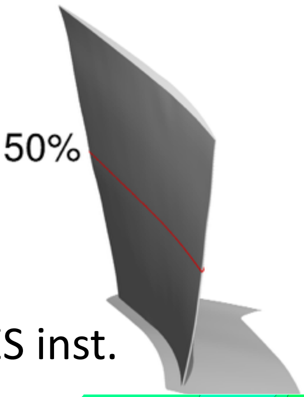
X-LES of the transonic flow field in the NASA Rotor 37



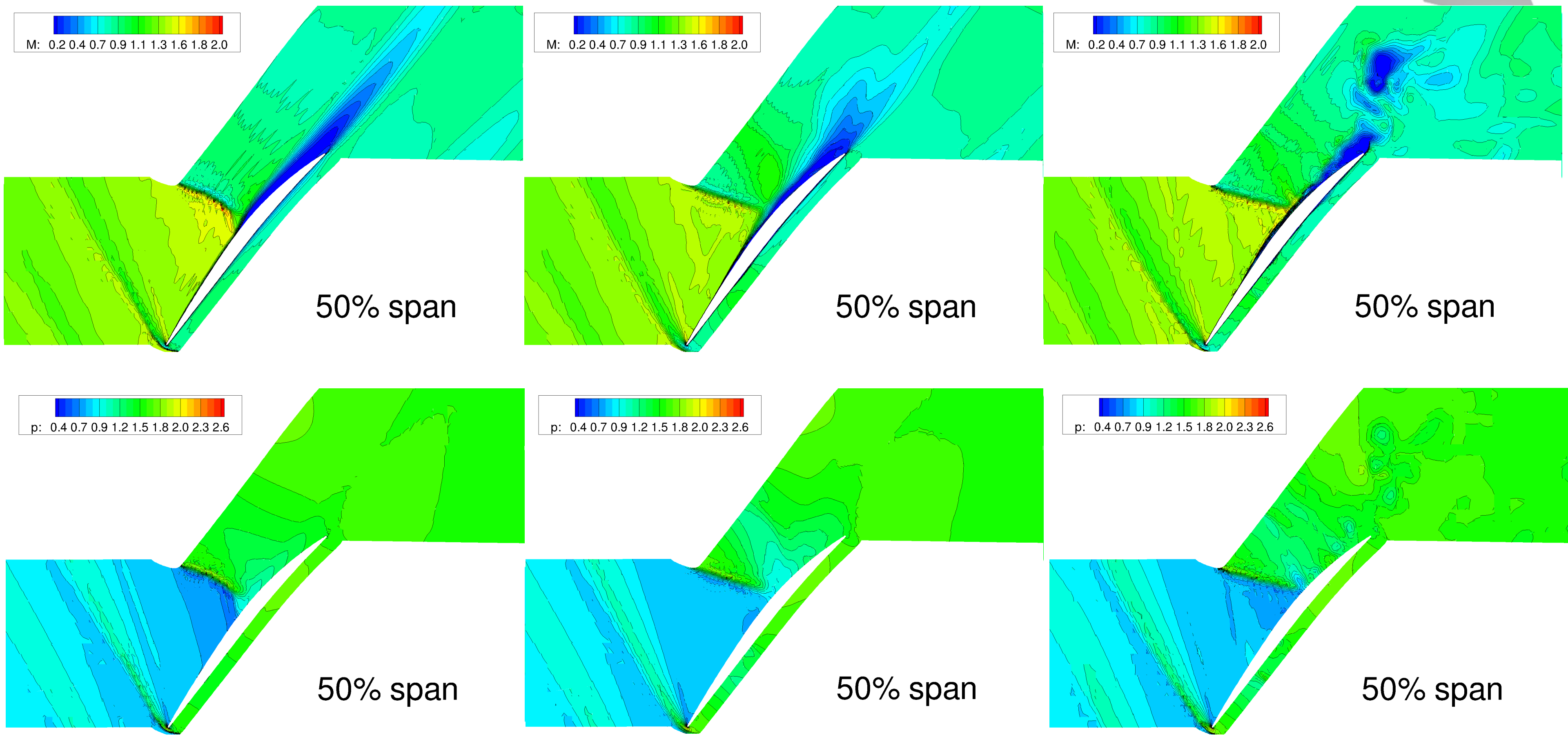
RANS | X-LES ave. | X-LES inst.



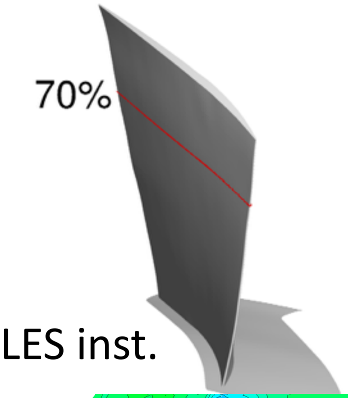
X-LES of the transonic flow field in the NASA Rotor 37



RANS | X-LES ave. | X-LES inst.



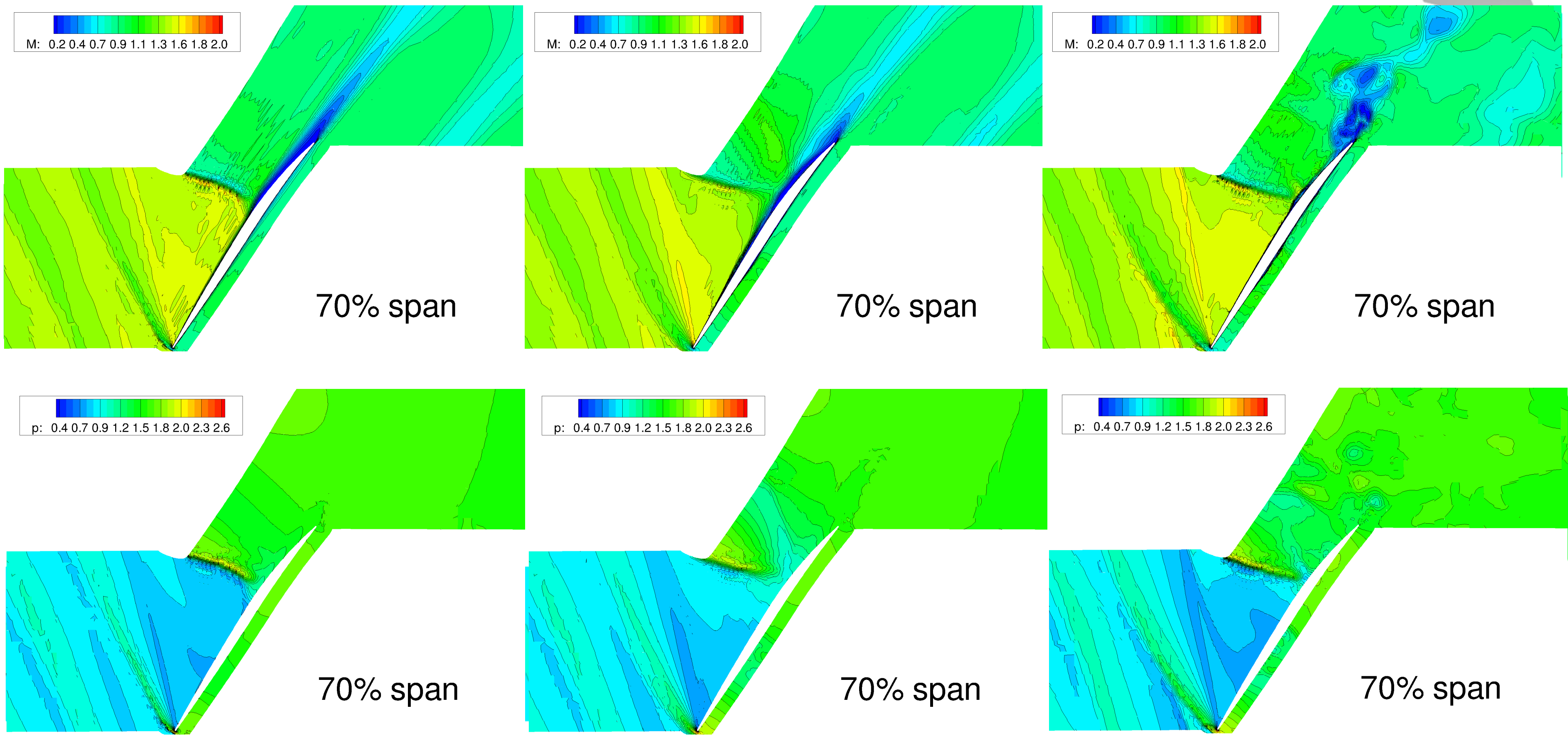
X-LES of the transonic flow field in the NASA Rotor 37



RANS

X-LES ave.

X-LES inst.

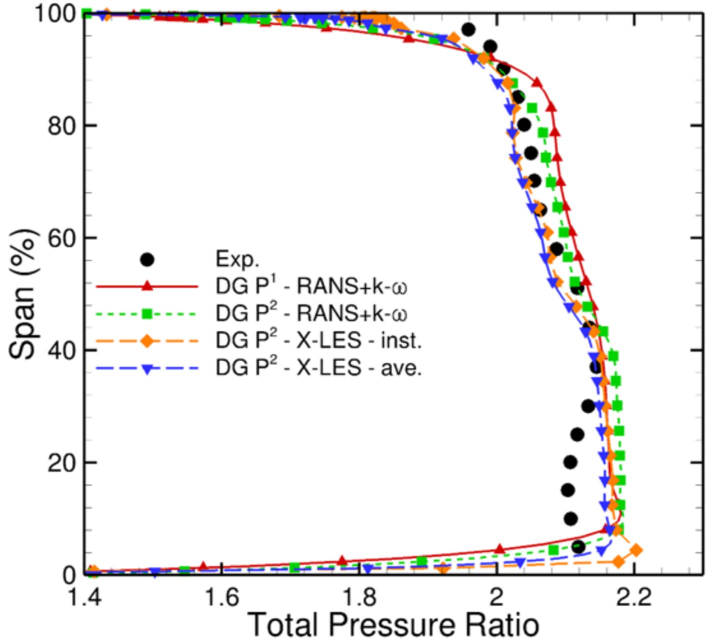


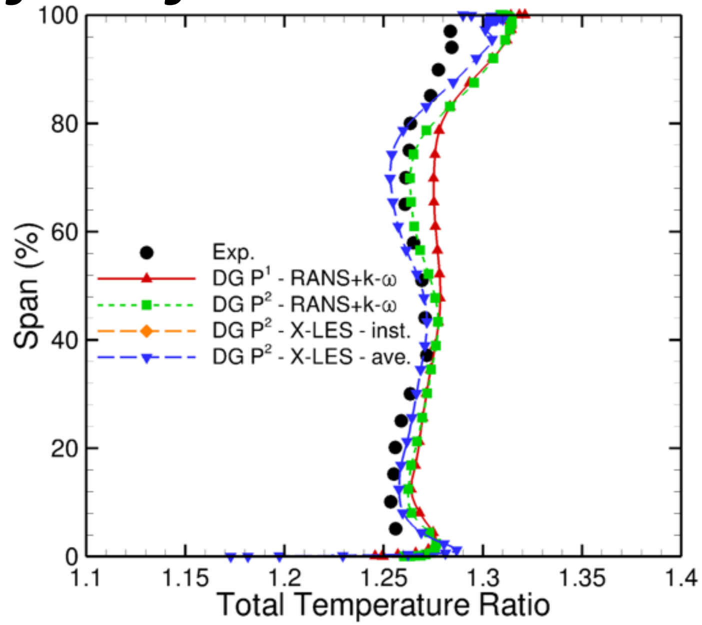
X-LES of the transonic flow field in the NASA Rotor 37

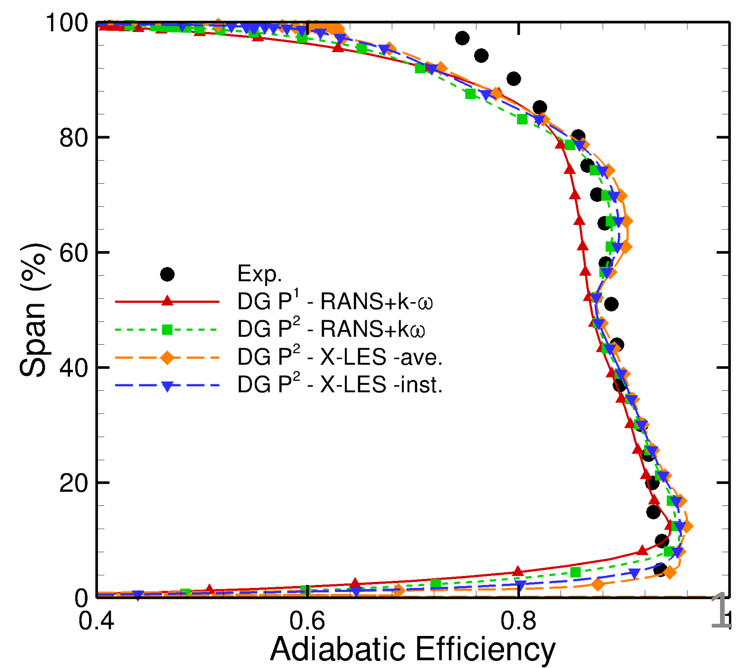
UNIVERSITÀ DEGLI STUDI DI BERGAMO

TILDA

Spanwise distributions







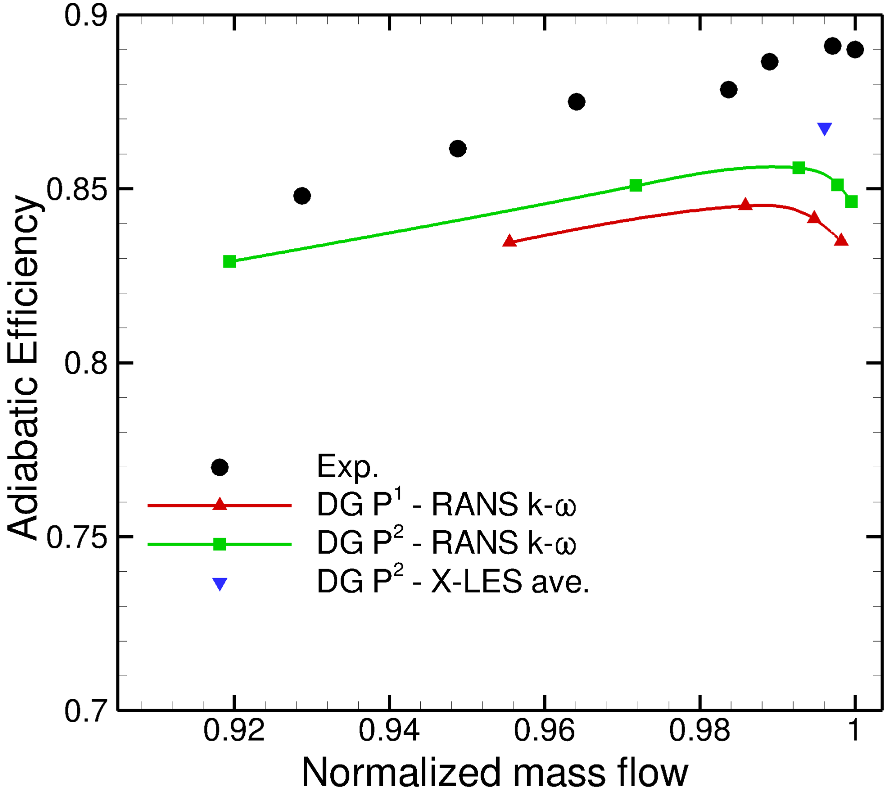
Initializing the solution from a flow field corresponding
to a normalized mass flow,
resulting from a RANS computation, of ≈0.98
X-LES moved towards ≈0.996

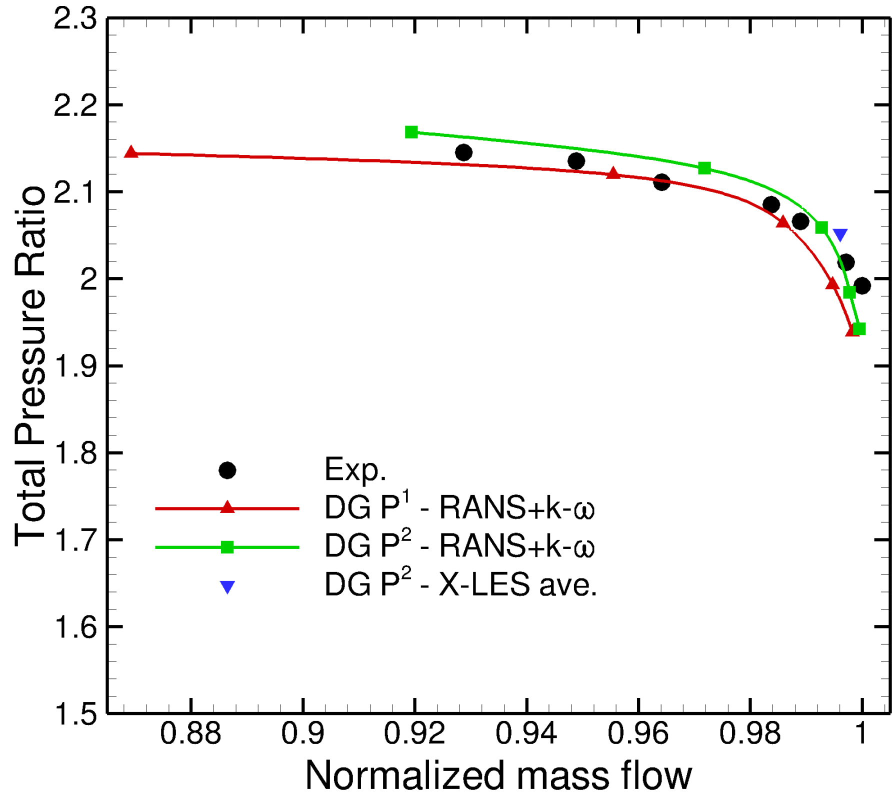
X-LES of the transonic flow field in the NASA Rotor 37

UNIVERSITÀ DEGLI STUDI DI BERGAMO

TiLDA

Spanwise distributions





Initializing the solution from a flow field corresponding to a normalized mass flow, resulting from a RANS computation, of ≈0.98

X-LES moved towards ≈0.996

References I/III

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

[Arnold et al., 2002] Arnold, D. N., Brezzi, F., Cockburn, B., and Marini, D. (2002). Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal., 39(5):1749–1779.

[Karniadakis and Sherwin, 2005] Karniadakis, G. E. and Sherwin, S. (2005). Spectral/hp Element Methods for Computational Fluid Dynamics. Numerical Mathematics and Scietific Computation. Oxford University Press, USA.

[Gobbert and Yang, 2008] Gobbert, K. and Yang, S. (2008). Numerical demonstration of finite element convergence for Lagrange elements in COMSOL Multiphysics. In Proceedings of the COMSOL Conference 2008 Boston.

[Bassi, F. and Rebay, S., 1997] Bassi, F. and Rebay, S. (1997). A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier–Stokes equations. J. Comput. Phys., 131:267–279.

[Gottlieb and Groth, 1988] J.J.Gottlieb, C.P.T.Groth (1988), Assessment of Riemann solvers for unsteady one-dimensional inviscid flows of perfect gases, J. Comput. Phys. 78 437–458.

[Hänel et al.,1987] D. Hänel, R. Schwane, G. Seider (1987), On the accuracy of upwind schemes for the solution of the Navier–Stokes equations, AIAA Paper 87-1105.

References II/III

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

[Bassi et al., 2007] F. Bassi, A. Crivellini, D.A. DiPietro, S. Rebay (2007), An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows, Comput. Fluids 36 1529–1546.

[Bassi et al., 2014b] F. Bassi, L. Botti, A. Colombo, A. Ghidoni, F. Massa (2015), Linearly implicit Rosenbrock-type Runge-Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows, Comput. Fluids, 118, pp. 305-320, http://dx.doi.org/10.1016/j.compfluid.2015.06.007.

[Iannelli and Baker, 1988] G. S. Iannelli, A. J. Baker (1988), A stiffly-stable implicit Runge–Kutta algorithm for CFD applications, AIAA Paper 88-0416, AIAA.

[Lang and Verwer, 2001] J. Lang, J. Verwer, ROS3P—An accurate third-order Rosenbrock solver designed for parabolic problems, BIT 41 (4) (2001) 731–738.

[MGridGen, Moulitsas Karypis, 2001] Moulitsas I, Karypis G (2001). MGridGen/ ParmGridGen, serial/parallel library for generating coarse meshes for multigrid methods. Technical Report Version 1.0. University of Minnesota, Department of Computer Science/Army HPC Research Center. <http://www-users.cs.umn.edu/~moulitsa/software.html>

References III/III

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

[Bassi et al., 2009] F. Bassi, C. De Bartolo, R. Hartmann, A. Nigro (2009), A discontinuous Galerkin method for inviscid low Mach number flows, J. Comput. Phys. 228 (11) 3996 – 4011. doi:http://dx.doi.org/10.1016/j.jcp.2009.02.021.

[Wu and Faeth, 1993] Wu, J. and Faeth, G. (1993), Sphere wakes in still surroundings at intermediate reynolds numbers. AIAA J., 31:1448.

[Tomboulides and Orszag, 2000] Tomboulides, A. and Orszag, S. (2000), Numerical investigation of transitional and weak turbulent flow past a sphere. J. Fluid Mech., 416:45–73.

[Di Marzo, 1993] Di Marzo, G. (1993). RODAS5(4) - M´ethodes de Rosenbrock d'ordre 5(4) adaptées aux problemes différentiels-algébriques. MSc Mathematics Thesis; Faculty of Science, University of Geneva, Switzerland.

Grazie dell'attenzione