Efficient implementation of an implicit DG method for CFD

code MIGALE state-of-the-art

F. Bassi[1], L. Botti[1], **A. Colombo**[1], A. Crivellini[3],
M. Franciolini[3], A. Ghidoni[2], G. Manzinali[1->8],
F. Massa[1], A. Nigro[5], G. Noventa[2], S. Rebay[2]

HPC methods for
Computational Fluid Dynamics and Astrophysics
CINECA, Bologna (Italy)
14th November 2017

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]
Stefano Rebay[2]

Andrea Crivellini[3]
Matteo Franciolini[3]

Carmine De Bartolo[4]

Alessandra Nigro[5]

Daniele Di Pietro[6]

Gabriel Manzinali[1->8]

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

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

2018 / 2015 — hi-o + HPC = 48h DNS/LES

IDIHOM

2014 / 2010 — industrialization of hi-o RANS

ADIGMA

2009 / 2006 — hi-o methods for comp. flows

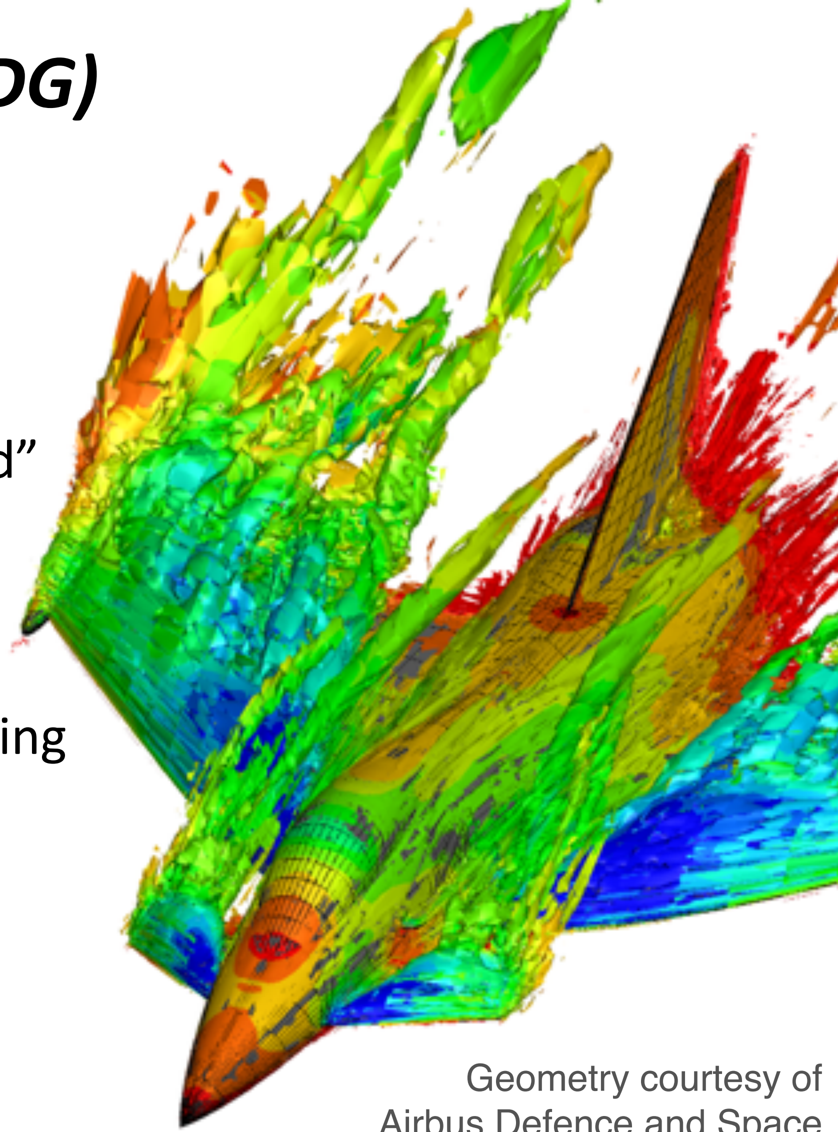
Why Discontinuous Galerkin (DG) methods for CFD?

Cons

High-order accuracy comes at an increased computational cost with respect to “standard” FD or FV

Pros

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



Geometry courtesy of Airbus Defence and Space

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 DG methods basics and opportunities with some practical hint on their implementation

MIGALE CODE

- 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

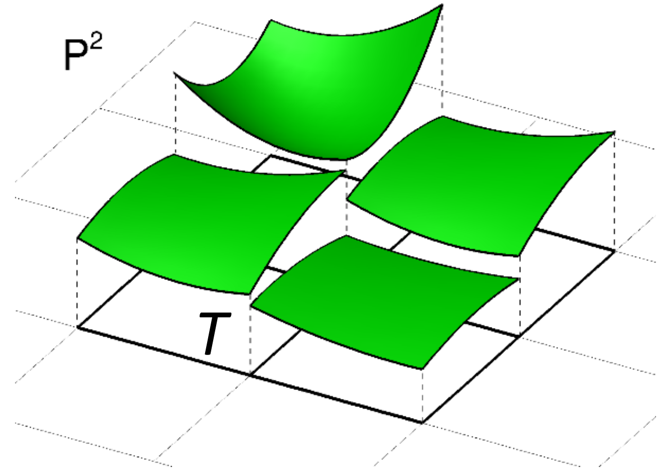
Our goal is to deal with different flow models using a unified numerical framework, e.g., time integrators, Riemann solvers

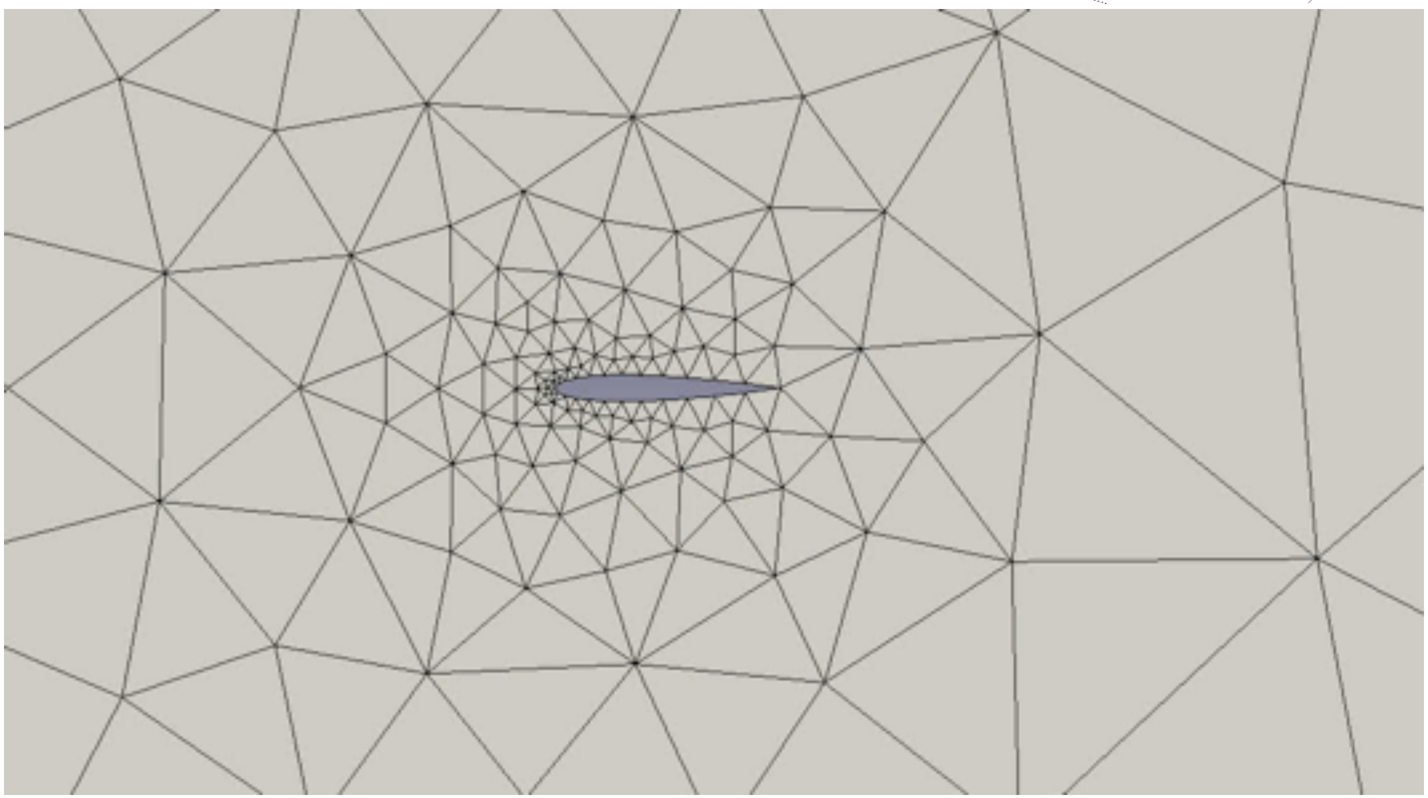
The DG method basic idea

the solution approximation

The numerical solution is approximated by high-order polynomial functions

Functions are not required to be continuous across the elements interfaces





For computing integrals any element $T \in \mathcal{T}_h$ can be mapped on a reference element T_{ref}, *e.g.* the unit quadrangle

The basis functions

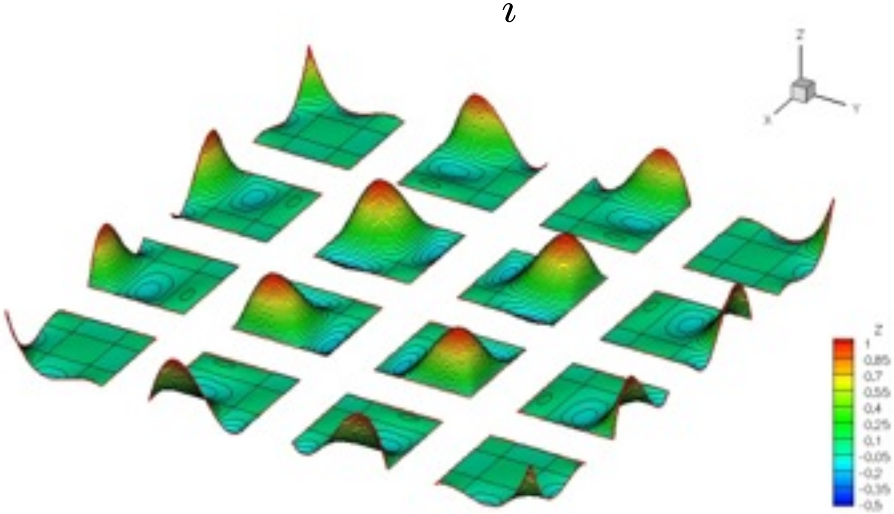
The basis functions

Basis functions can be defined on

a reference space

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

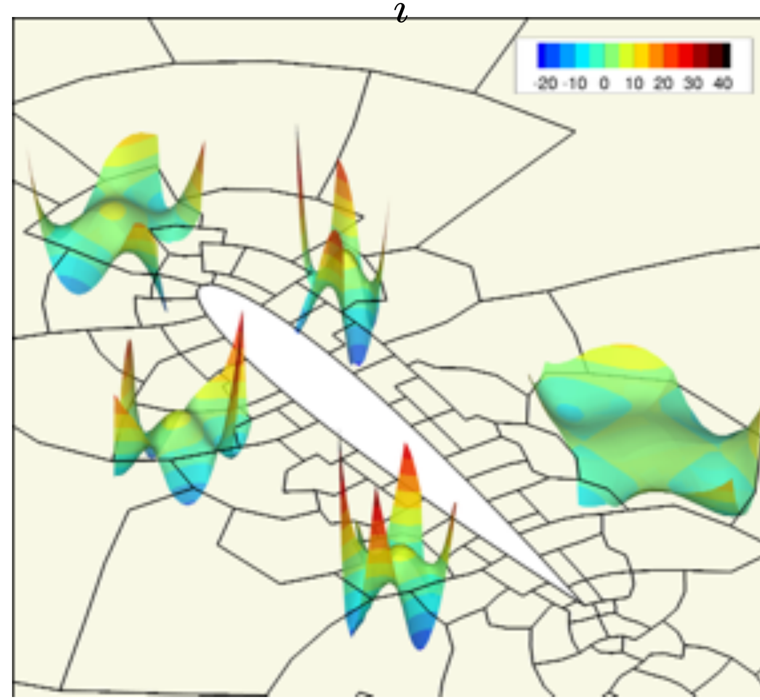
$$\mathbf{w}(\mathbf{x}(\boldsymbol{\xi}),t)|_{T_{ref}} = \sum_i \mathbf{W}_i(t)\phi_i(\boldsymbol{\xi})$$



the "physical" space

basis is built on the real (mesh) element of any shape

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

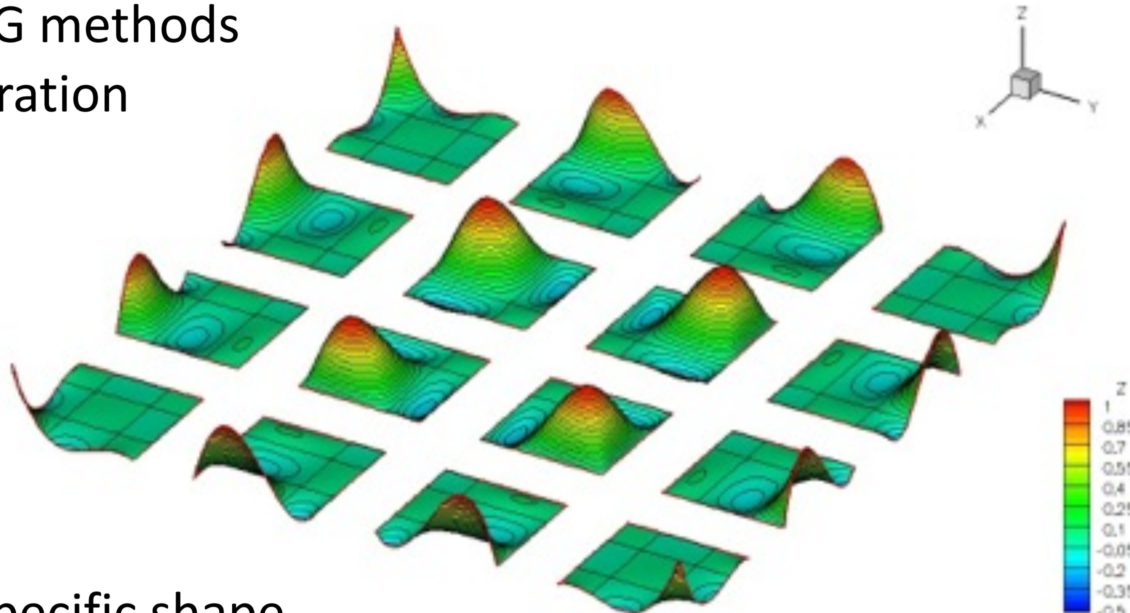


The basis functions

Basis functions on a reference frame

Pros

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



Cons

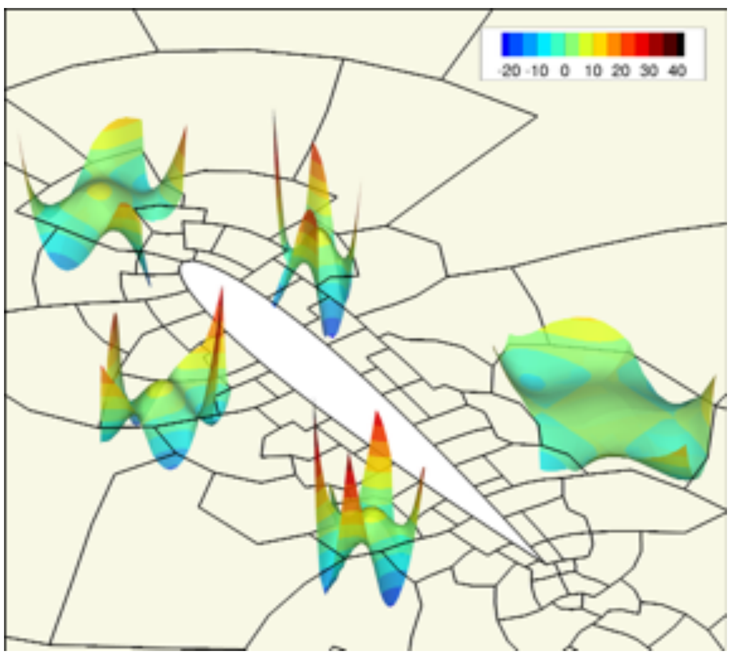
- defined for elements of specific shape
- extension to polytopal elements not straightforward
- stability issues for Legendere-Gauss-Lobatto nodes (aliasing → over-integration)
- polynomials on the reference element are no more polynomials on real elements with curved edges

The basis functions

Basis functions on the physical frame

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

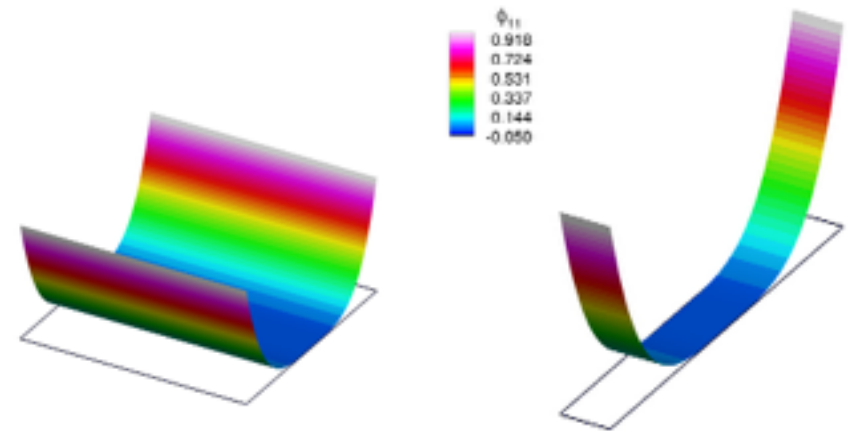
An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

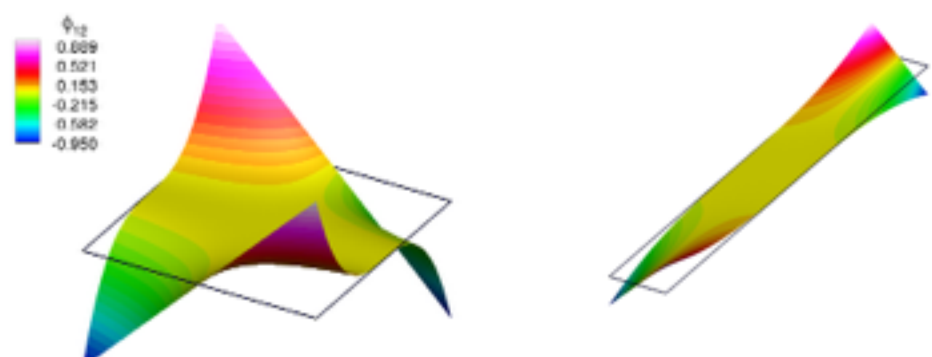
$$\mathbb{P}_d^k(\mathcal{T}_h) \stackrel{\text{def}}{=} \{\phi \in L^2(\Omega) \,|\, \phi_{|T} \in \mathbb{P}_d^k(T),\ \forall T \in \mathcal{T}_h\}$$

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



AR=1 *AR=4*

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



AR=1 *AR=4*

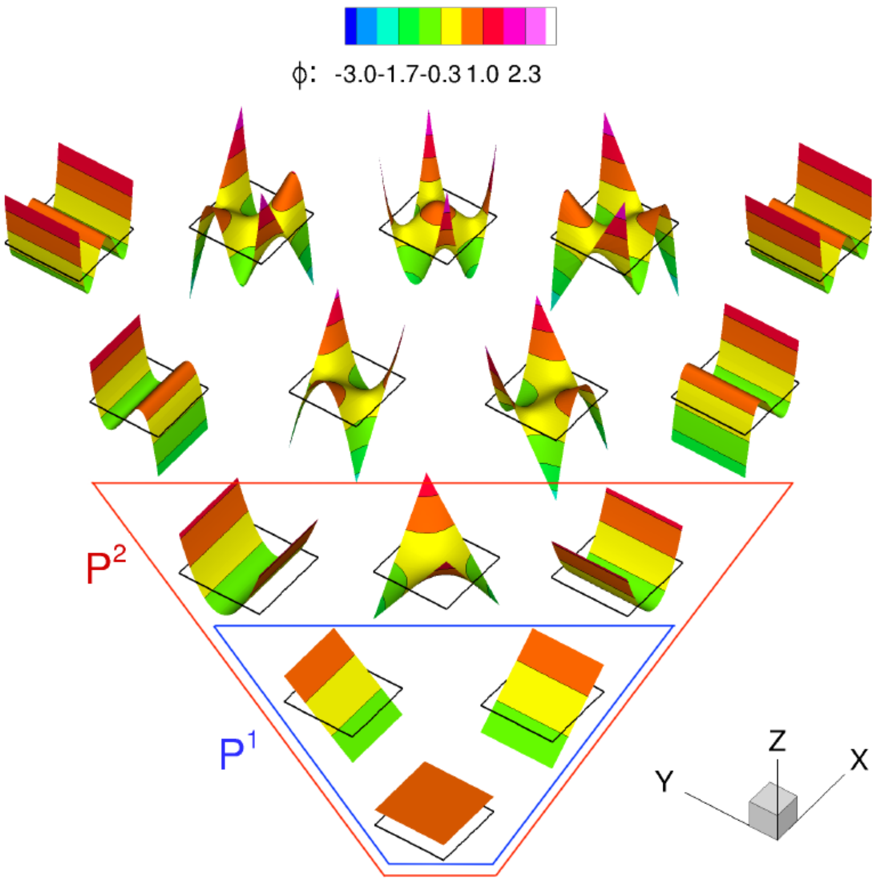
The basis functions - physical frame

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

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

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

    $\phi_i^T \leftarrow b_i^T$

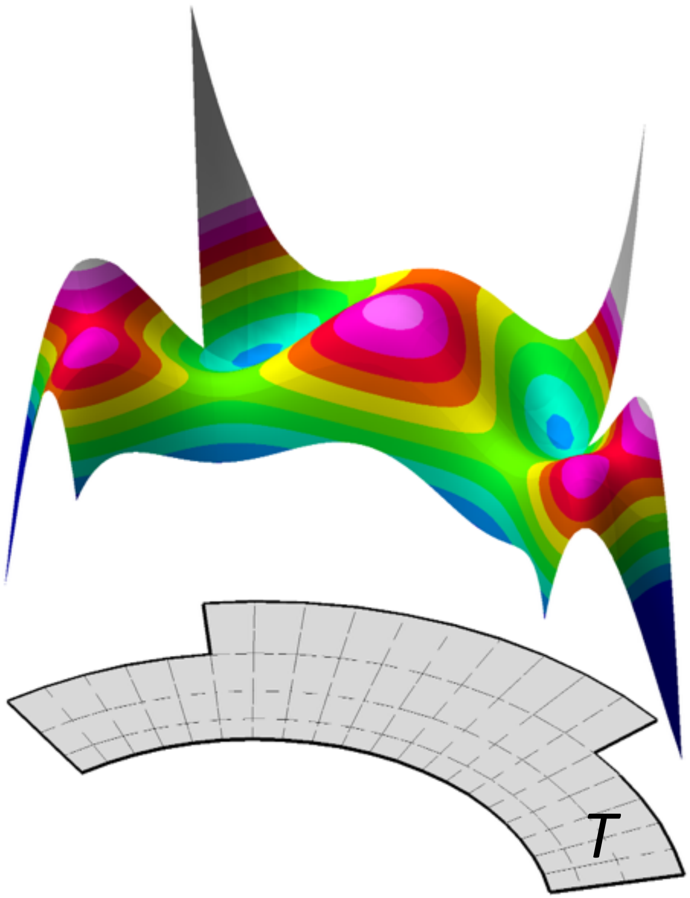
end for

The basis functions - physical frame

An orthonormal and hierarchical set

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

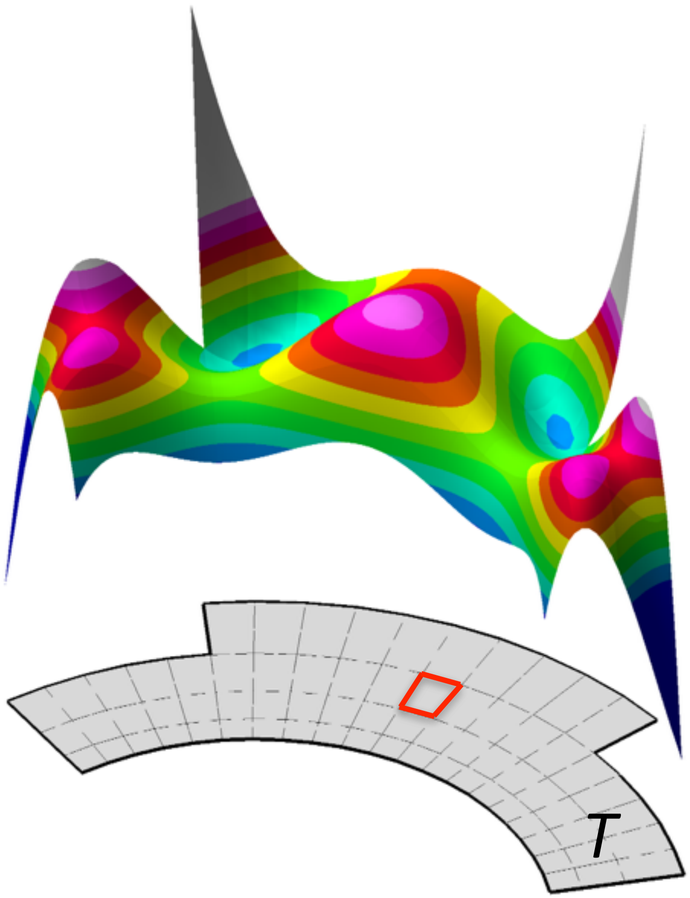
The basis functions - physical frame

An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

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



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

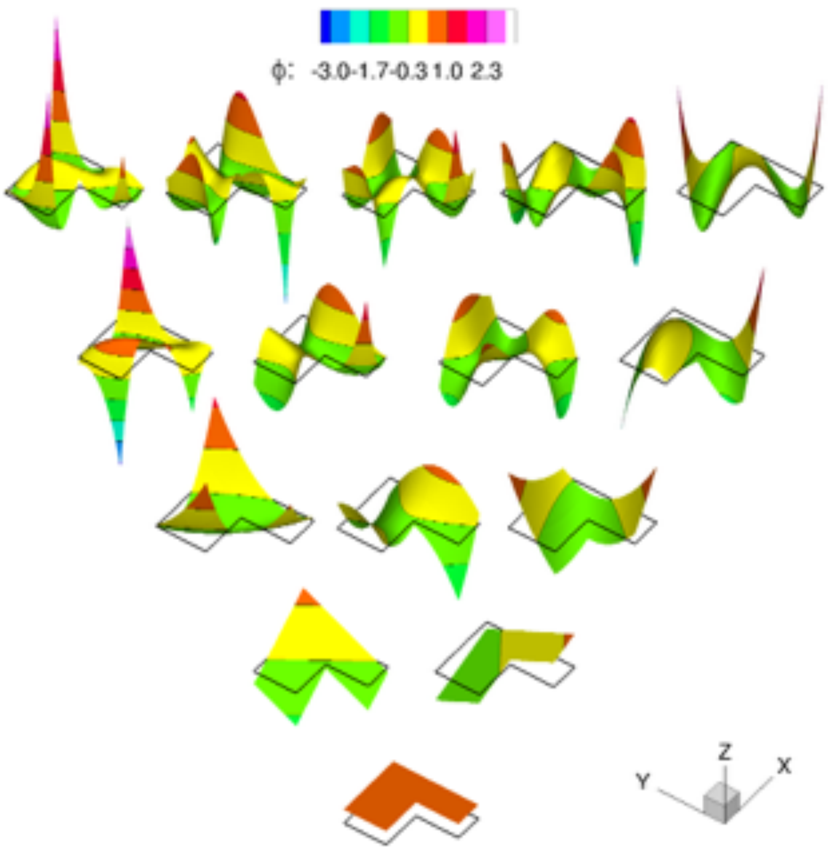
The basis functions - physical frame

An orthonormal and hierarchical set

UNIVERSITÀ DEGLI STUDI DI BERGAMO

We define discrete polynomial spaces in physical coordinates

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



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

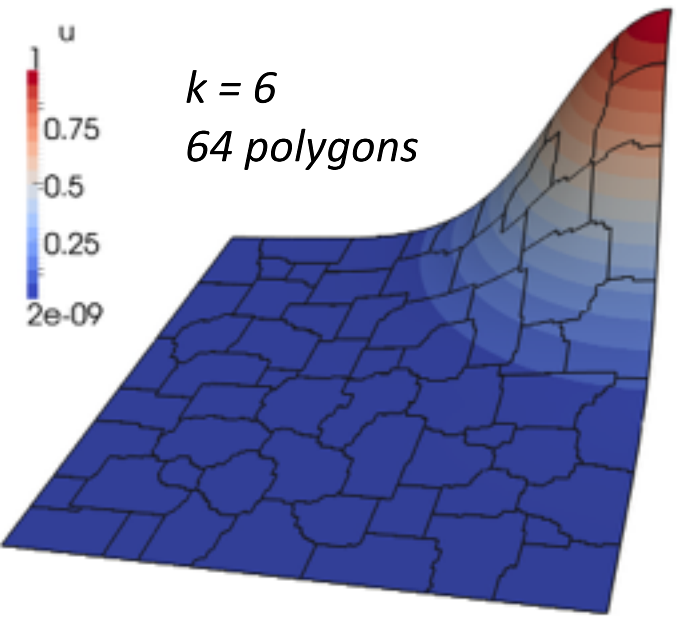
L2-Projection and DG solution tests: quadrilateral vs. polygonal elements

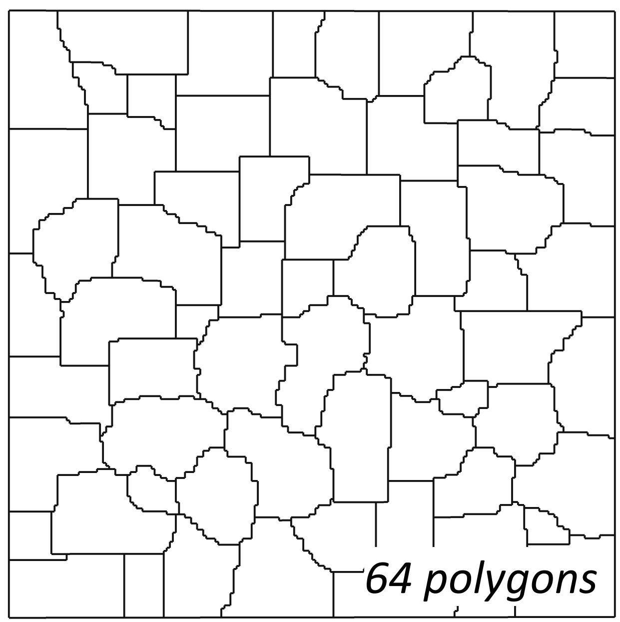
Test on the exact solution of a Poisson problem proposed in [Karniadakis and Sherwin, 2005]

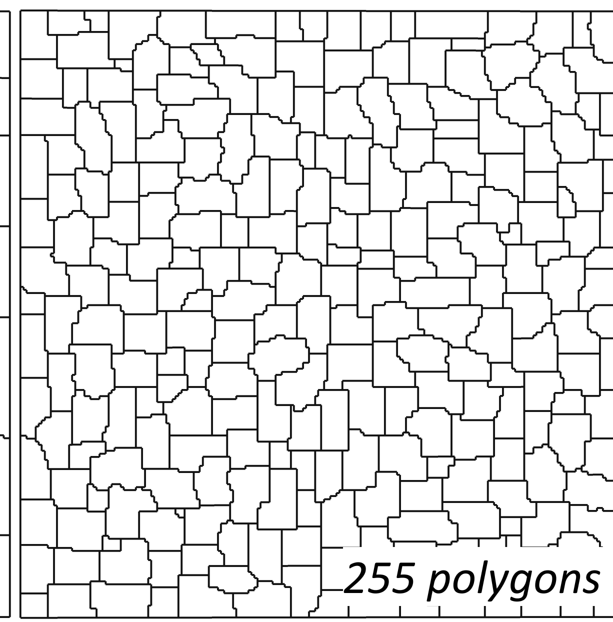
$$u = e^{-2.5[(x-1)^2+(y-1)^2]} \qquad \Omega = [-1,1]^2$$

<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]





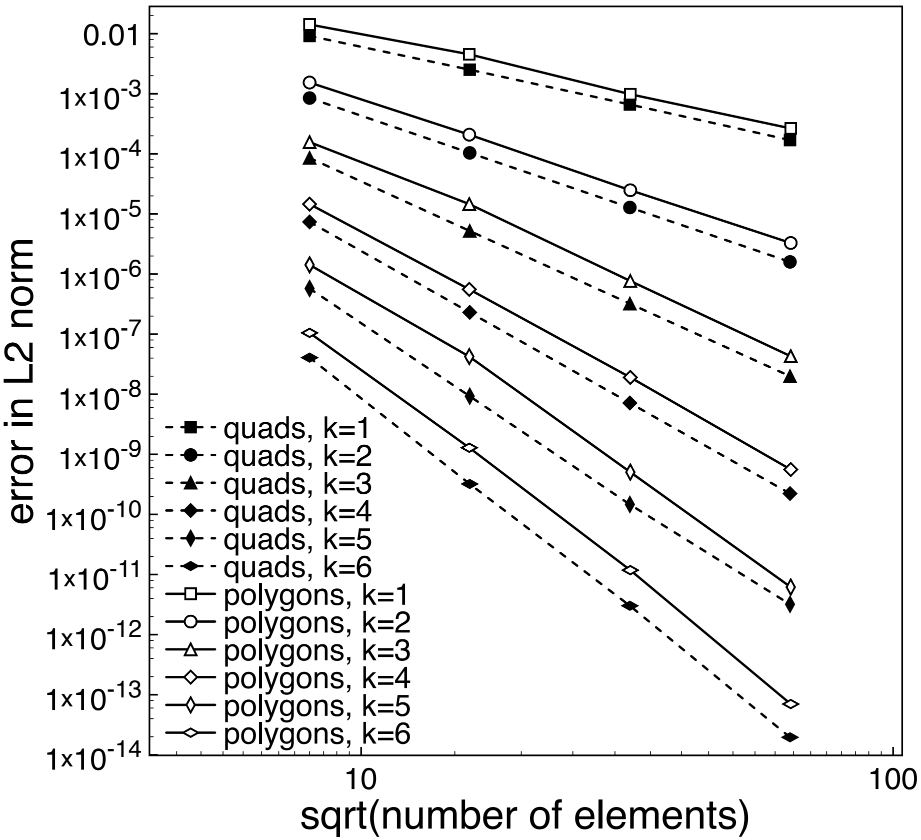


1) http://www-users.cs.umn.edu/~moulitsa/software.html

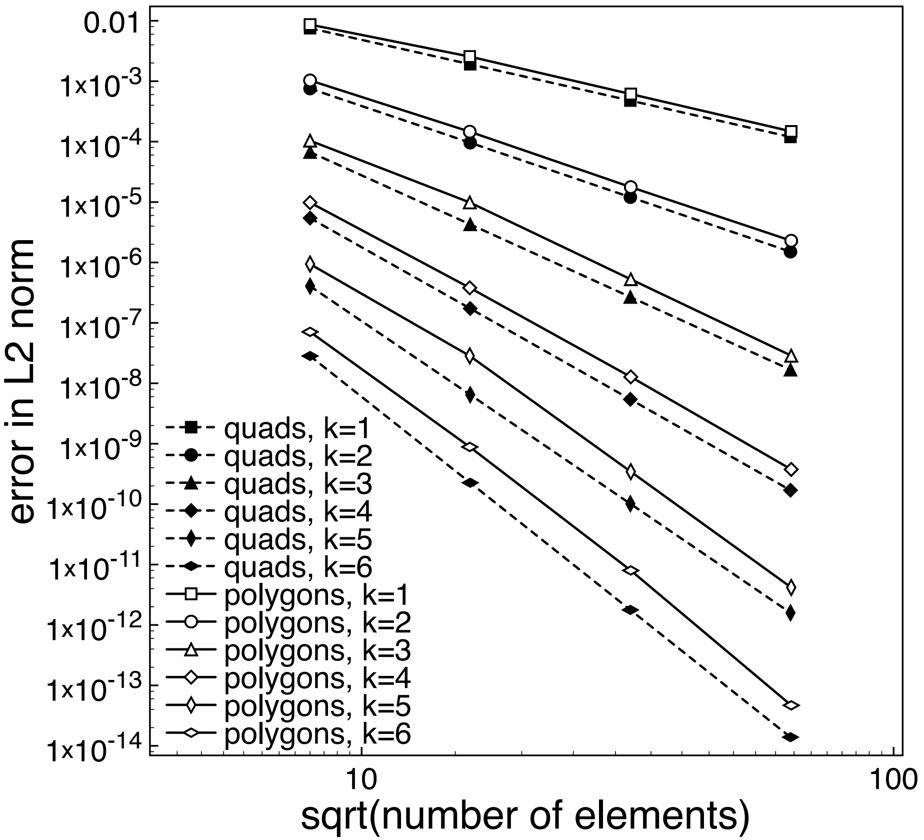
The basis functions - physical frame

L2-Projection and DG solution tests: quadrilateral vs. polygonal elements

Projection test on *u*



Poisson problem DG solution

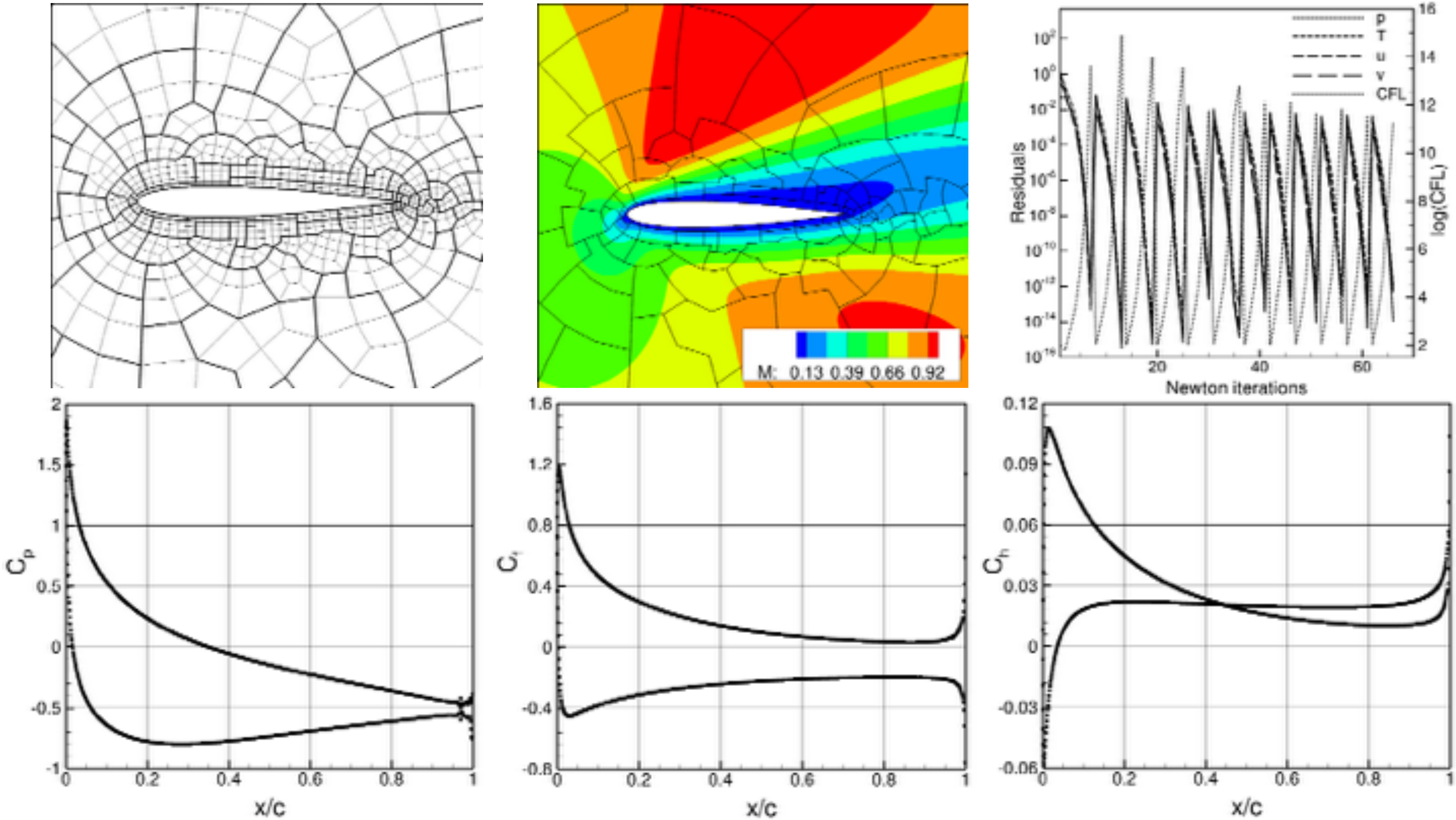


Test on the exact solution of a Poisson problem proposed in [Karniadakis and Sherwin, 2005]

The basis functions - physical frame

A CFD tests: polygonal elements

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

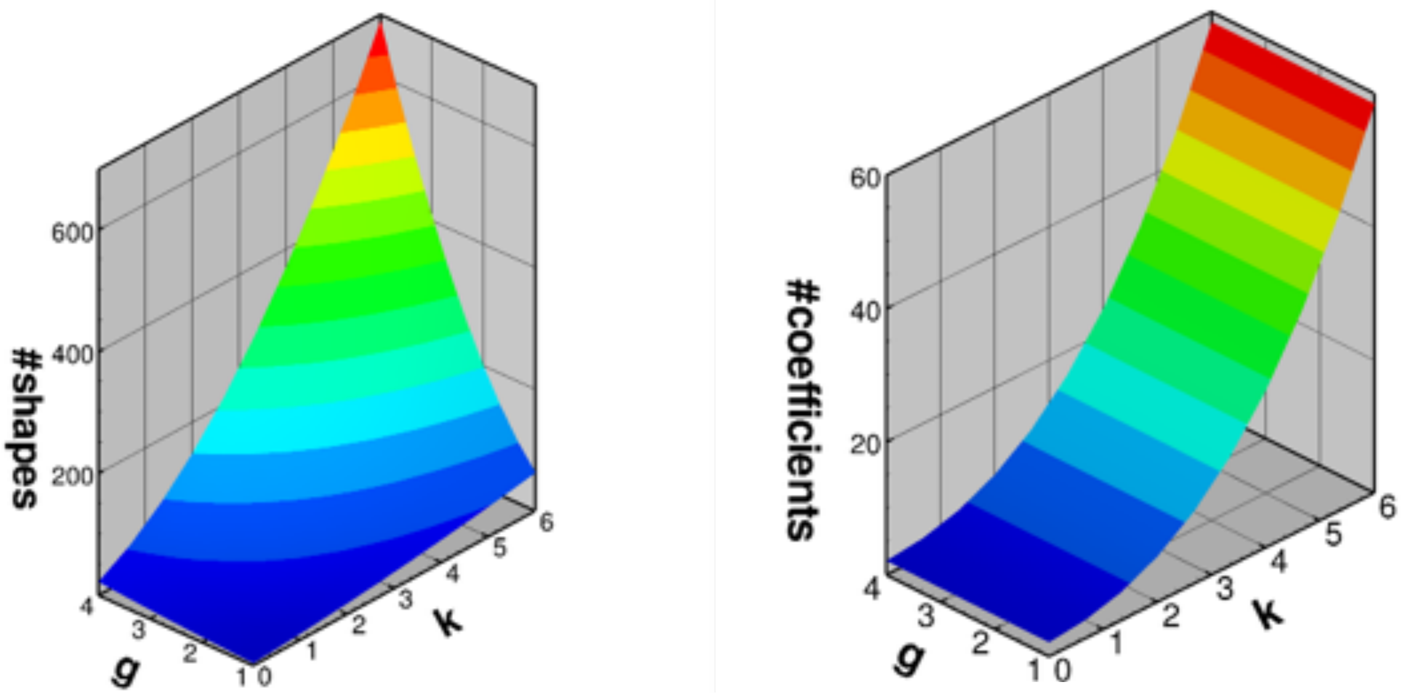


Hints on an efficient implementation of physical frame basis functions

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



Number of evaluation for Euler equations operators assembly

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

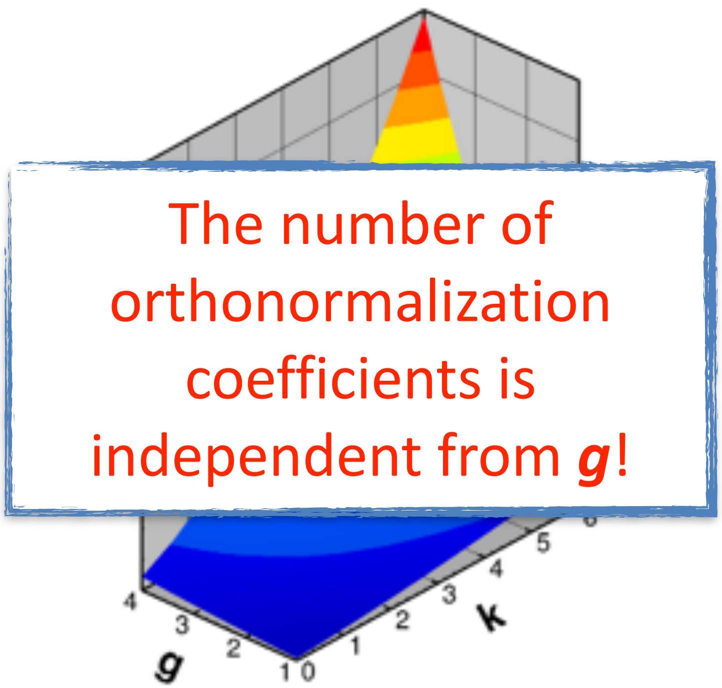
g *is the polynomial degree of the reference-to-physical-frame mapping* $\mathbf{x}(\boldsymbol{\xi})$

Hints on an efficient implementation of physical frame basis functions

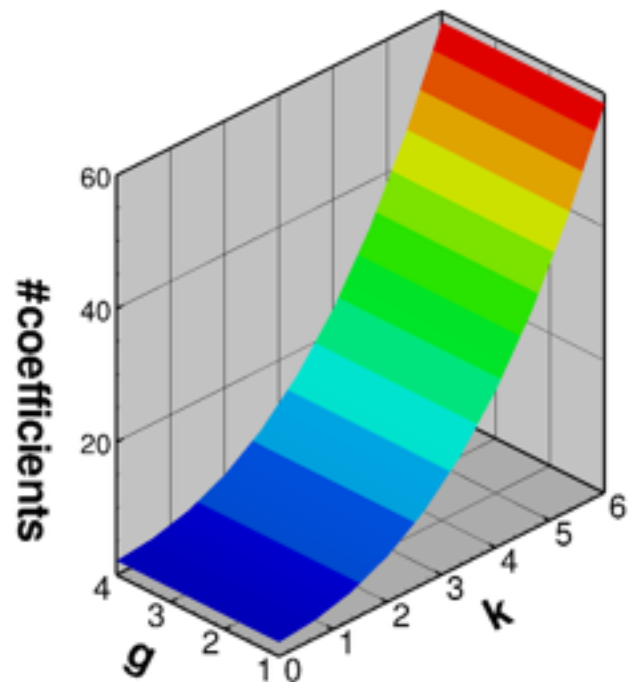
UNIVERSITÀ DEGLI STUDI
DI BERGAMO

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***!



Number of evaluation for Euler equations operators assembly

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

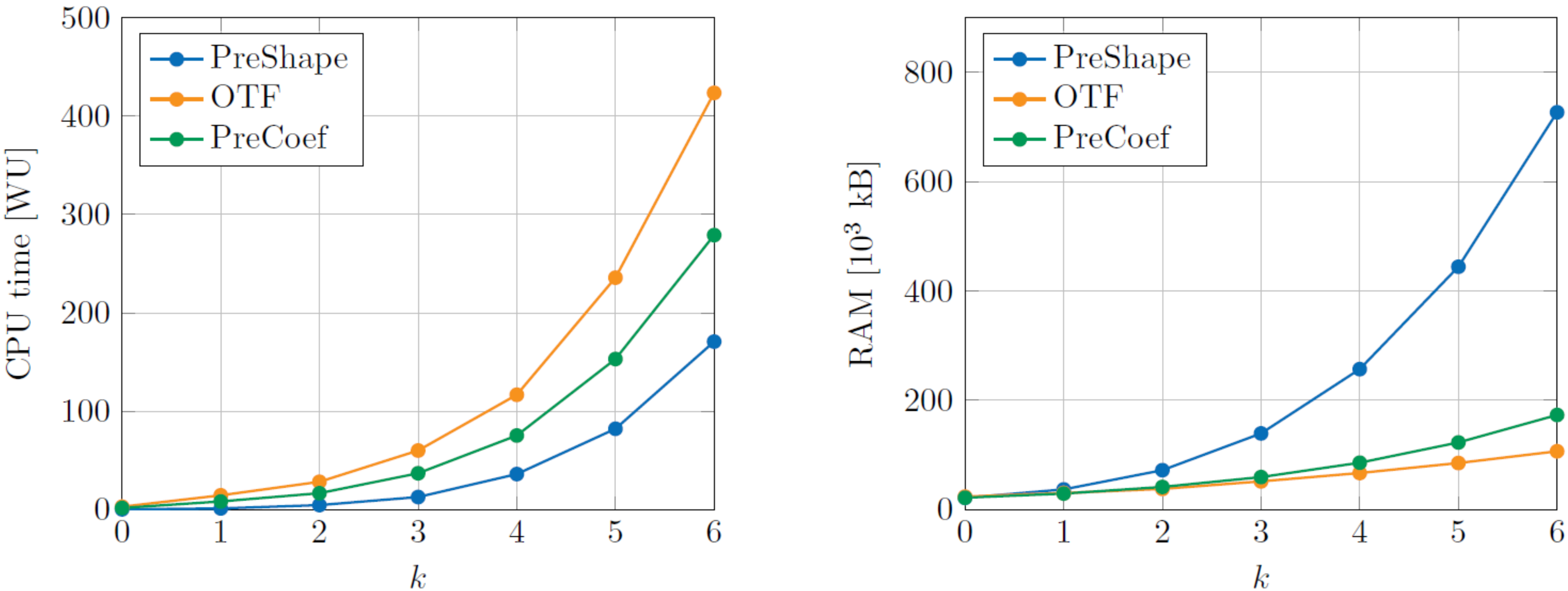
Strategy	Coefficients evaluation	Shapes evaluation	CPU usage	Memory footprint
OTF	During assembly	During assembly	High	Low
PreCoef	During pre-proc.	During assembly	Medium	Medium
PreShape	During pre-proc.	During pre-proc.	Low	High

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

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

…our guidelines…

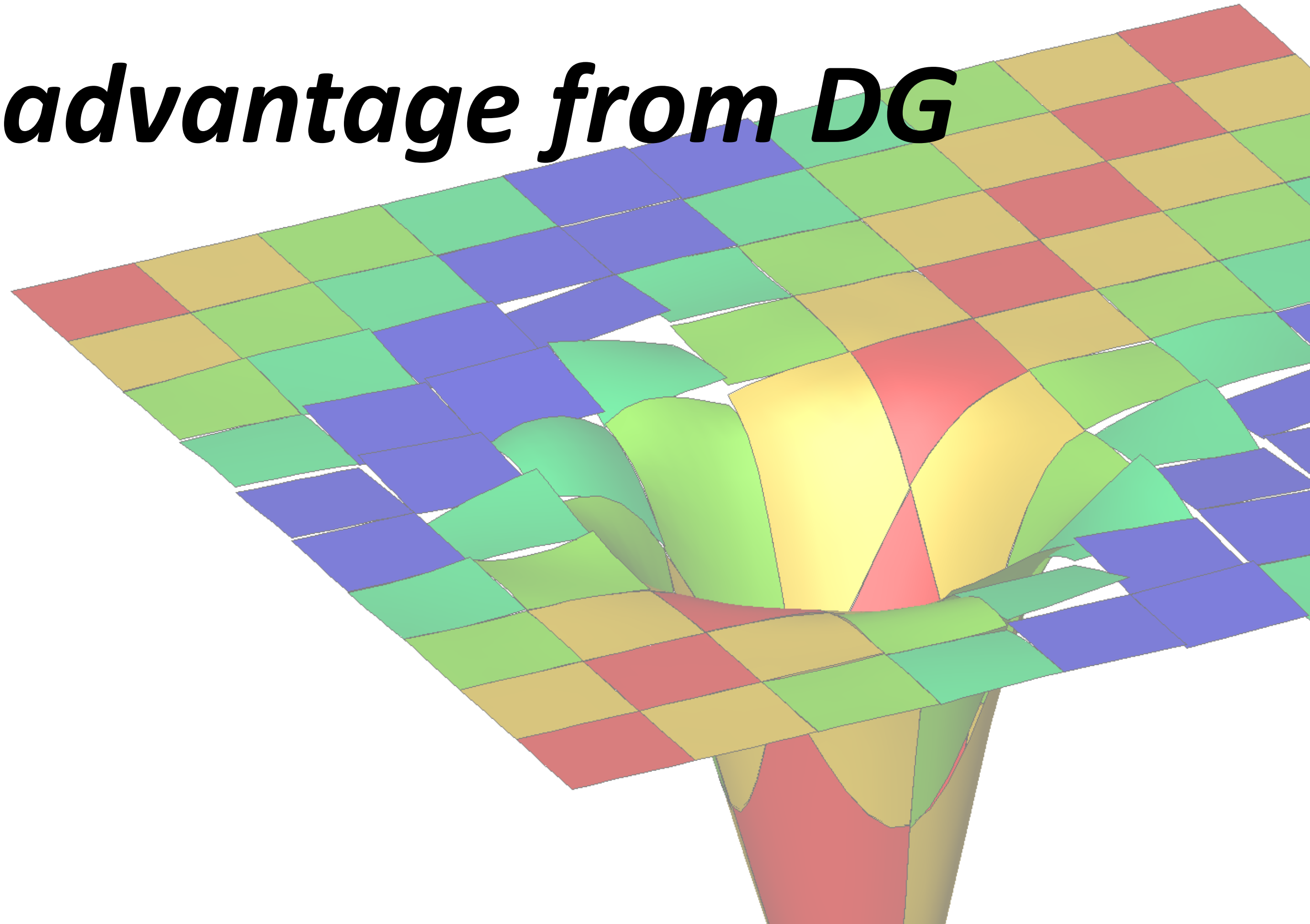
1) The best choice depends on the simulation at hand, *e.g.* RANS, DNS

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

Take advantage from DG



Take advantage from DG peculiarities

UNIVERSITÀ DEGLI STUDI DI BERGAMO

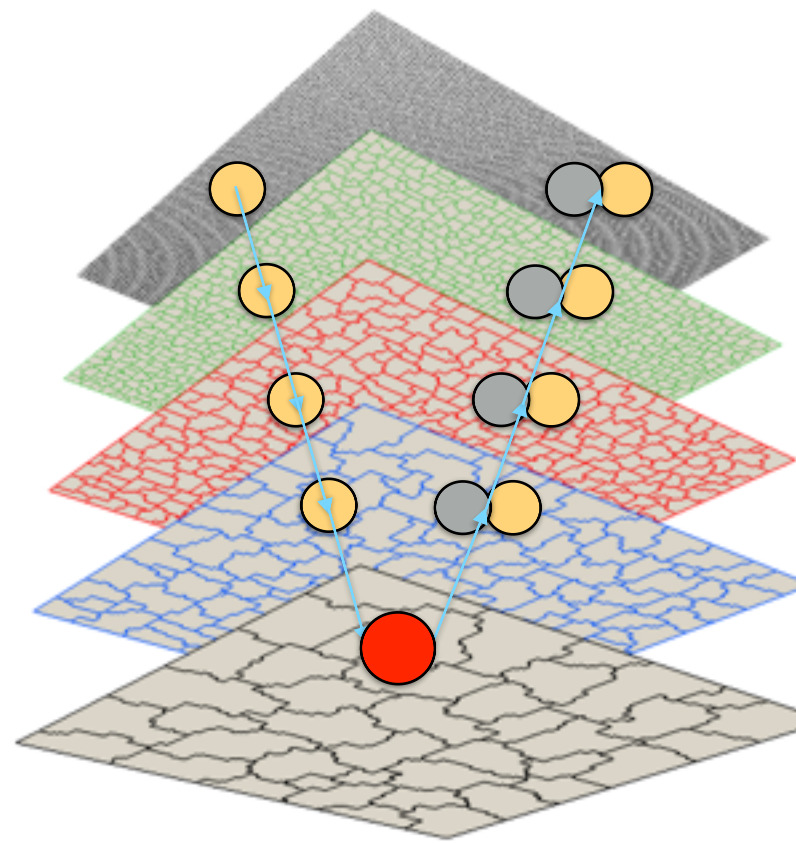
Being able to deal with agglomerated elements and relying on nested polynomial spaces we can boost our solution with multigrid (MG)

Linear MG is an iterative solution strategy for linear (or linearized) systems

MG efficiently solves $Au = f$ by exploiting the solution of several coarse problems $A_l \Delta u_l = r_l$

The coarse problems can be explicitly built on

- a sequence of *h*-coarsened grids *h*-MG
  agglomeration yields nested grids of arbitrarily shaped elements
- a sequence of *k*-coarsened problems *p*-MG
  different levels are discretized with different order of accuracy



In both cases the use of orthonormal and hierarchical basis in physical space greatly simplify the implementation!

Take advantage from DG peculiarities

UNIVERSITÀ DEGLI STUDI DI BERGAMO

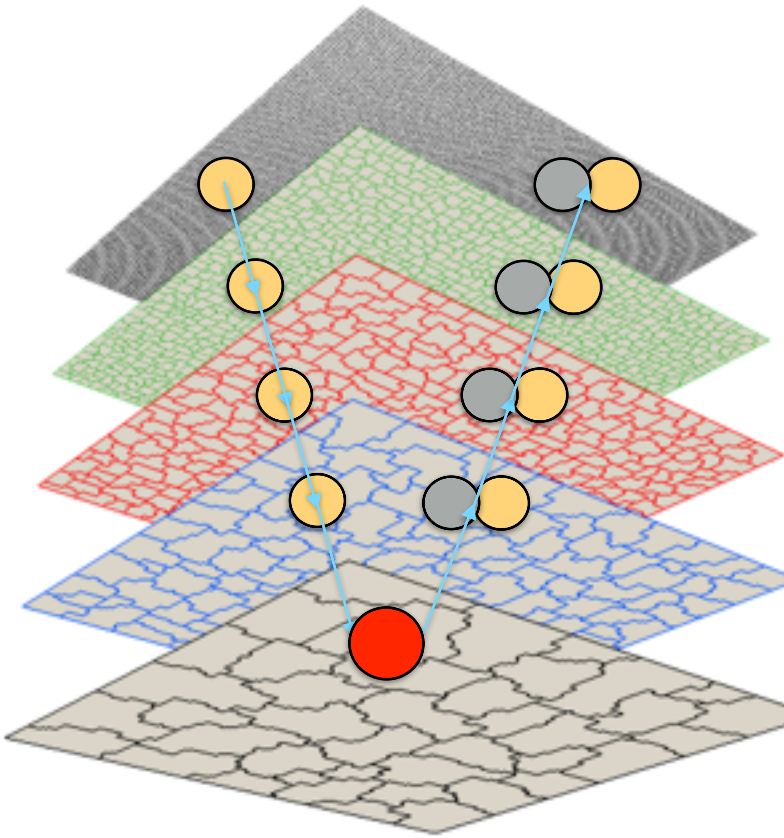
Being able to deal with agglomerated elements and relying on nested polynomial spaces we can boost our solution with multigrid (MG)

Linear MG is an iterative solution strategy for linear (or linearized) systems

MG efficiently solves $Au = f$ by exploiting the solution of several coarse problems $A_l \Delta u_l = r_l$

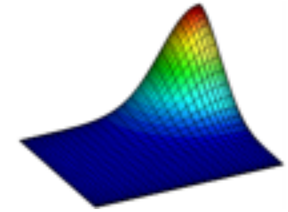
Key ideas:

1. Iterative solvers can efficiently smooth the high-frequency modes of the error
2. Low-frequency modes of the error appear more oscillatory on coarser spaces
3. MG exploits smoothers acting on coarser spaces to accelerate the convergence
4. The error on the finest level can be reduced trough the coarser levels corrections



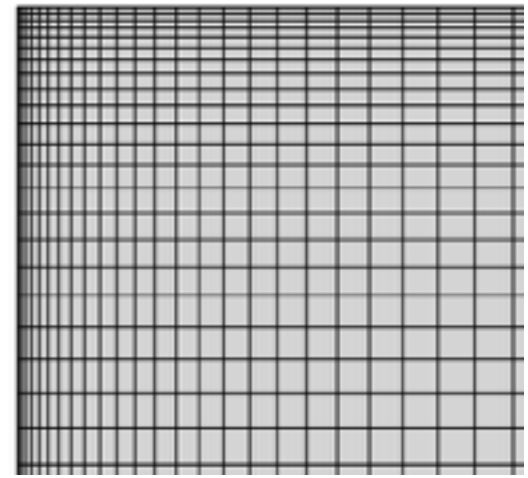
Multigrid as a preconditioner for GMRES

Boost your solution! A metter of ***ACCURACY…***



Test on the exact solution of a Poisson problem on a graded 256^2 el. grid

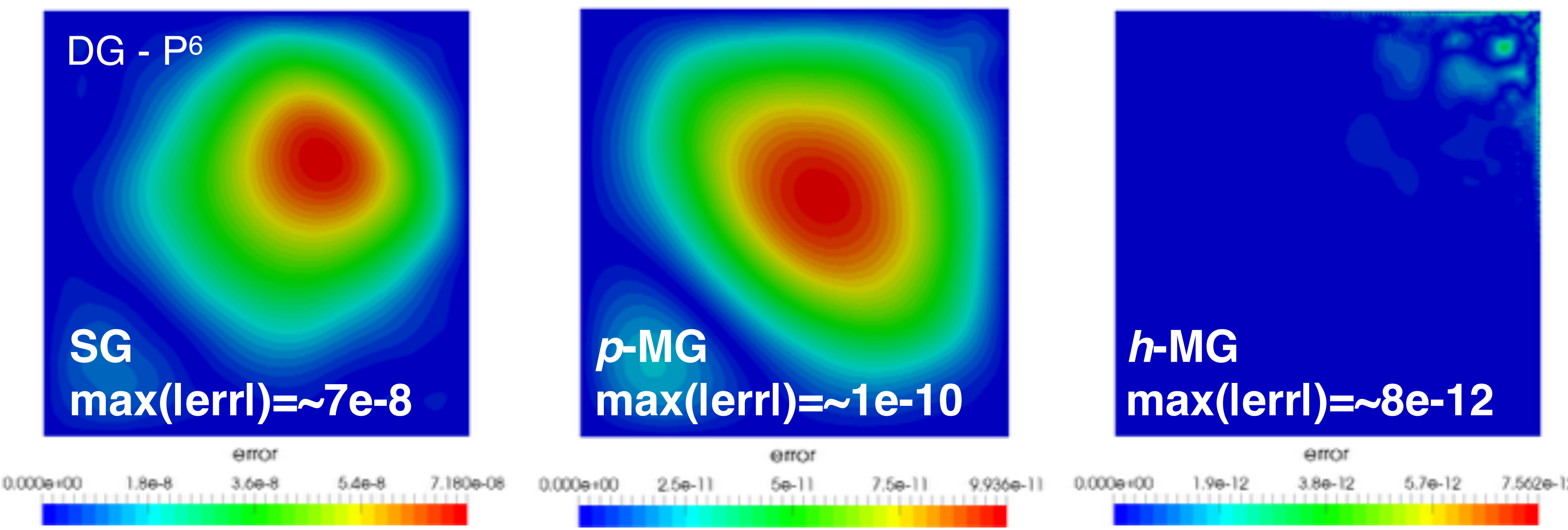
$$u = e^{-2.5[(x-1)^2+(y-1)^2]} \qquad \Omega = [-1,1]^2$$

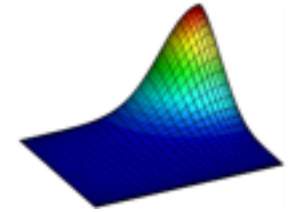


GMRES(200) parameters: *rtol=1e-14*, n_{its}=*2000*

x-MG

- *1 GMRES iteration on the intermediate levels (if any)*
- *coarse solver: GMRES(200) rtol=1e-3,* n_{its} *= 400*





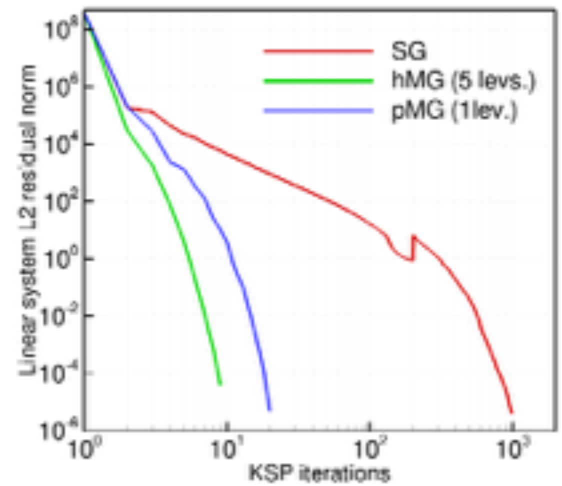
Multigrid as a preconditioner for GMRES

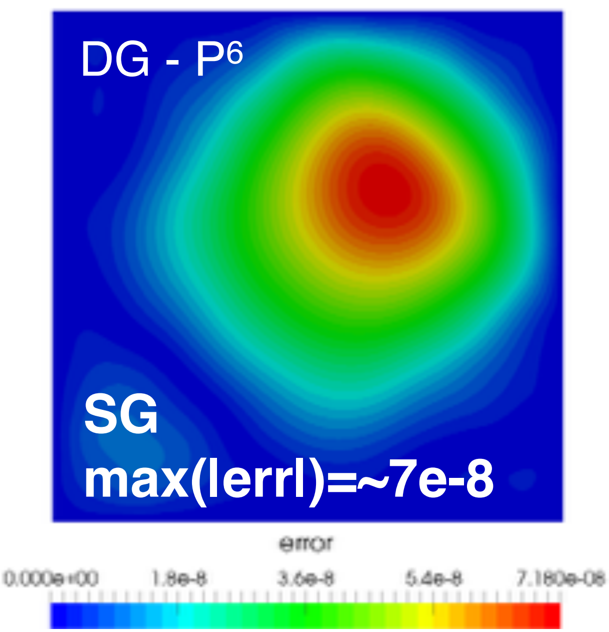
*Boost your solution! A metter of **ACCURACY**…*

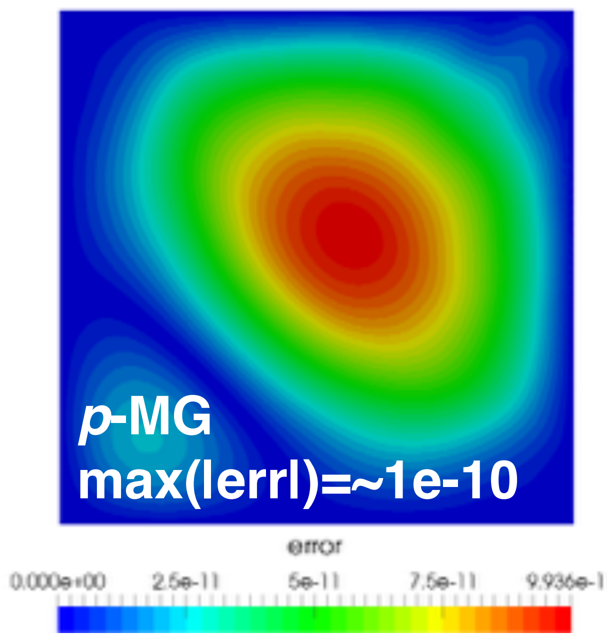
Test on the exact solution of a Poisson problem on a graded 256^2 el. grid

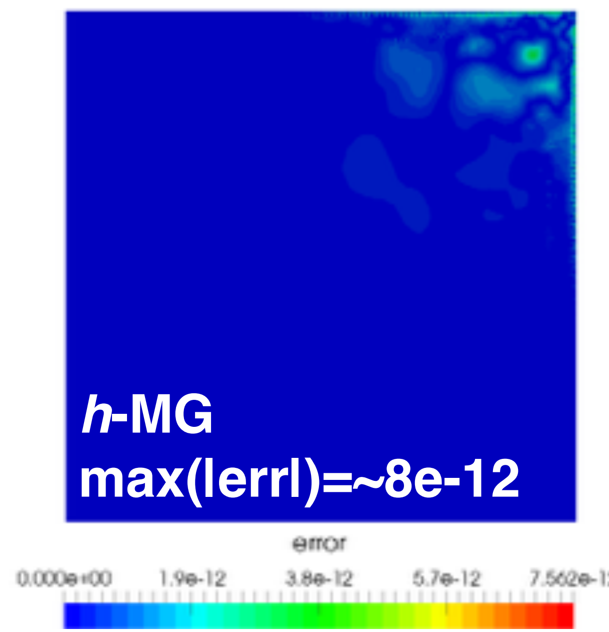
$$u = e^{-2.5[(x-1)^2+(y-1)^2]} \qquad \Omega = [-1,1]^2$$

Although both SG and *x*-MG reach the tight value *rtol=1e-14* according to the L2 residual norm convergence test, we observe very different results in terms of solution error due to the different smoothing properties of the iterative linear solvers

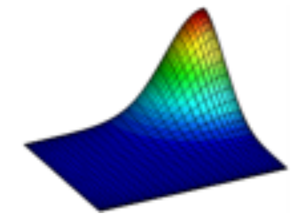








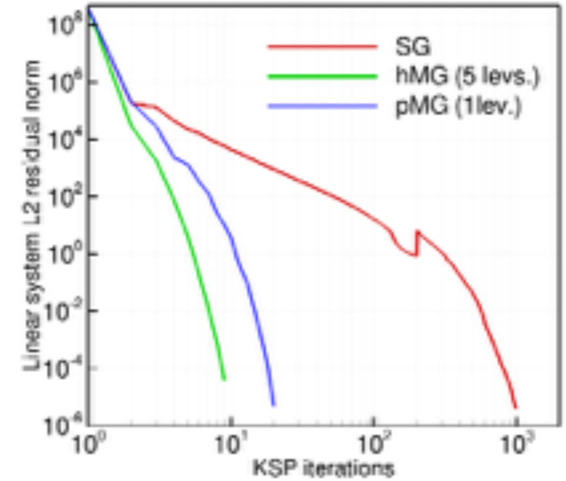
Multigrid as a preconditioner for GMRES



*Boost your solution! A metter of **ACCURACY**…*

Test on the exact solution of a Poisson problem on a graded 256^2 el. grid

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



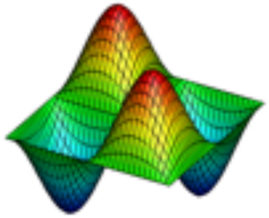
Multigrid strategies exhibit far better performances with respect to the SG solver, especially on graded grids

For high-fidelity simulations, e.g. DNS, a convergence test alternative to the L2 norm of the system residual must be considered

	SG	*h*	*p*
GMRES its.	983	9	20
L2 solution error	5.25e-8	5.5e-13	8.8e-11
solution time (*x*	—	4.6%	14%
assembly time (*x*	—	225%	101%
total time (	—	26%	11%

Multigrid as a preconditioner for GMRES

Boost your solution! A metter of ***EFFICIENCY…***

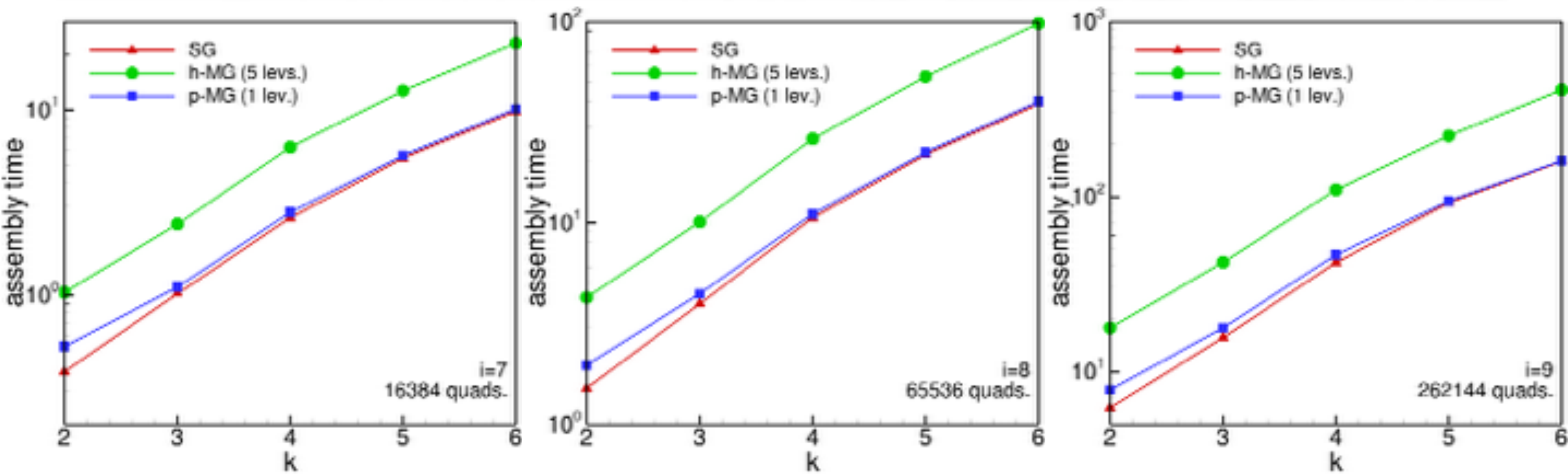


Test on the exact solution of a Poisson problem on a set of 2^{2i} (*i=6,…,9*) el. unif. grids varying the polynomial degree *k* of the DG solution (*rtol = 1e-12*)

$$u = \sin(\pi x)\sin(\pi y) \qquad \Omega = [-1,1]^2$$

A brief note on x-MG implementation in our DG framework…

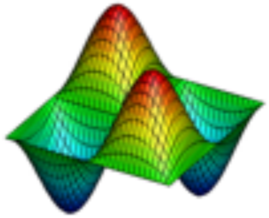
- *p*-MG algorithm is much easier to implement than *h*-MG
- *p*-MG restriction and prolongation operators are trivial and their use is very efficient in terms of number of operations



The cost of operators assembly is always in favor of p-MG

Multigrid as a preconditioner for GMRES

Boost your solution! A metter of ***EFFICIENCY…***

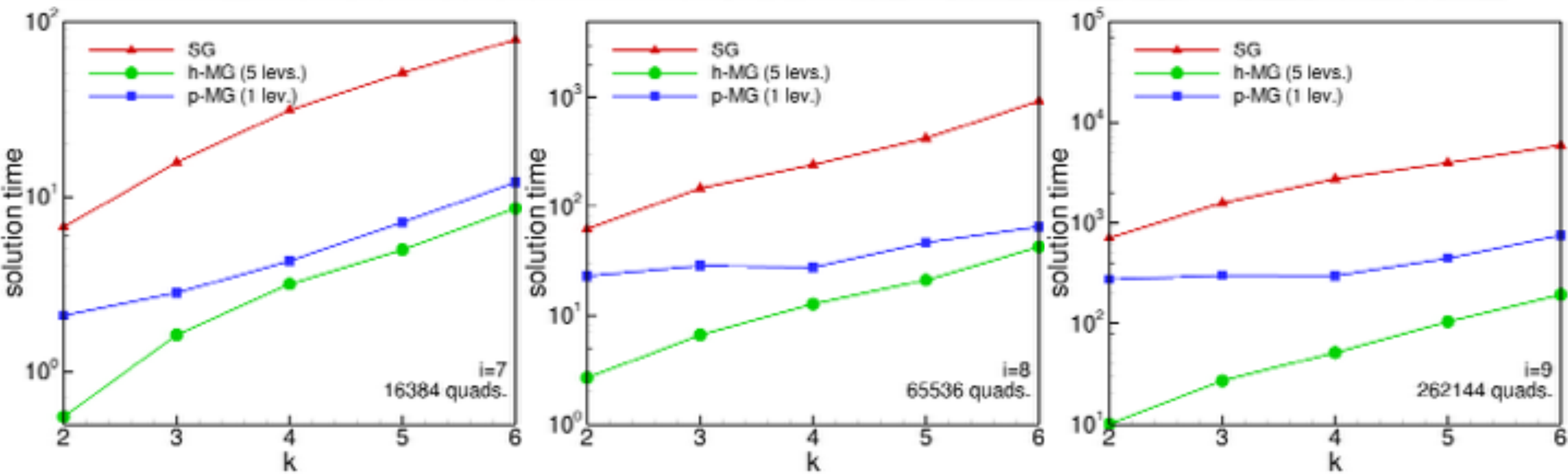


Test on the exact solution of a Poisson problem on a set of 2^{2i} (*i=6,…,9*) el. unif. grids varying the polynomial degree *k* of the DG solution (*rtol = 1e-12*)

$$u = \sin(\pi x)\sin(\pi y) \qquad \Omega = [-1,1]^2$$

A brief note on x-MG implementation in our DG framework…

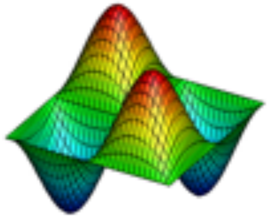
- *p*-MG algorithm is much easier to implement than *h*-MG
- *p*-MG restriction and prolongation operators are trivial and their use is very efficient in terms of number of operations



The cost of solution is always in favor of h-MG

Multigrid as a preconditioner for GMRES

Boost your solution! A metter of ***EFFICIENCY…***

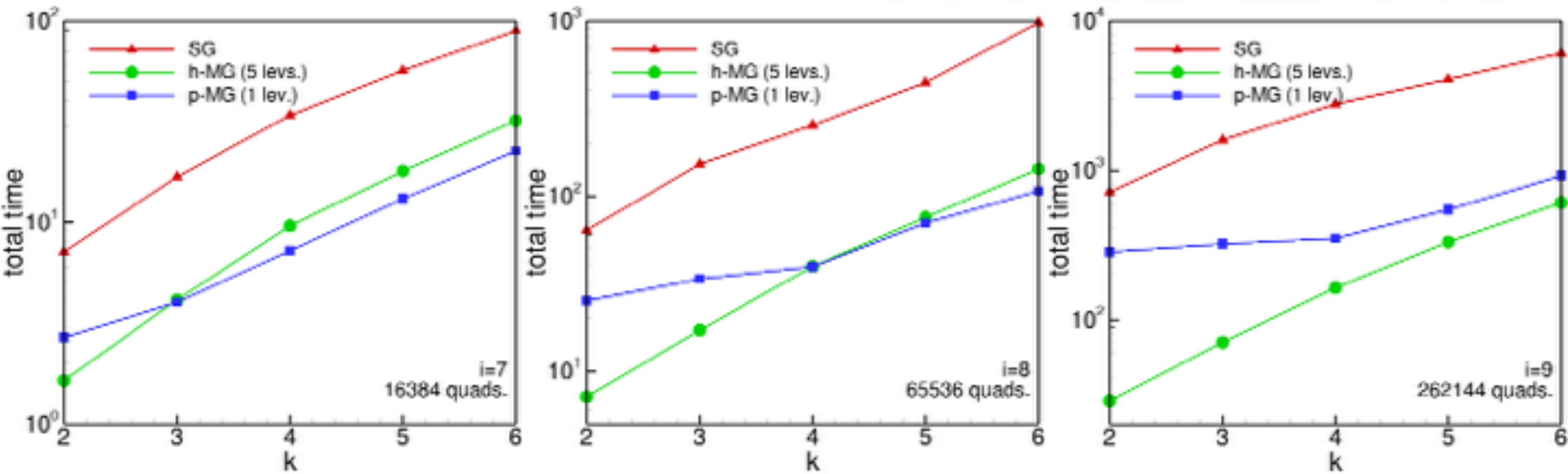


Test on the exact solution of a Poisson problem on a set of 2^{2i} (*i=6,…,9*) el. unif. grids varying the polynomial degree *k* of the DG solution (*rtol = 1e-12*)

$$u = \sin(\pi x)\sin(\pi y) \qquad \Omega = [-1,1]^2$$

A brief note on x-MG implementation in our DG framework…

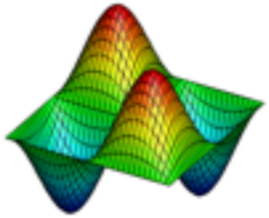
- *p*-MG algorithm is much easier to implement than *h*-MG
- *p*-MG restriction and prolongation operators are trivial and their use is very efficient in terms of number of operations



x-MG strategies are always more efficient than SG

Multigrid as a preconditioner for GMRES

Boost your solution! A metter of ***EFFICIENCY…***

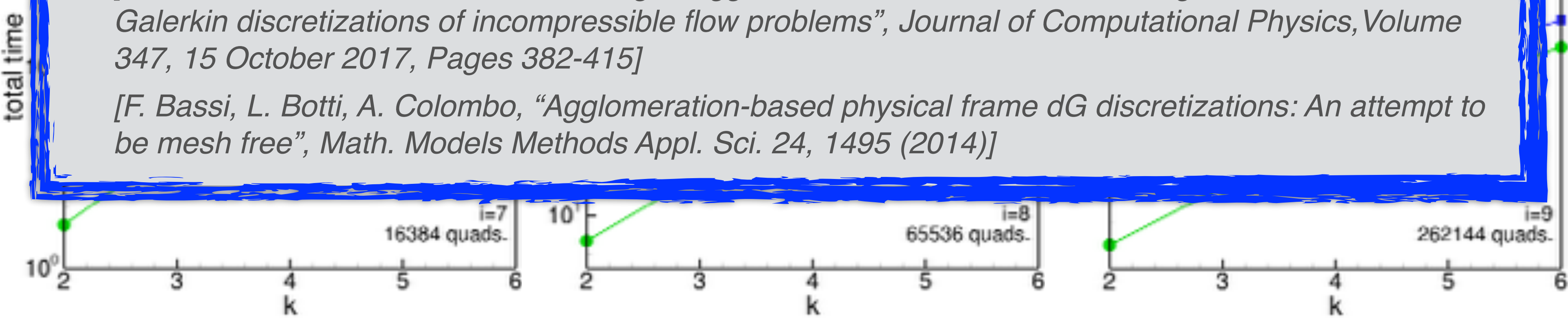


...on x-MG...

- *x-MG always pay off!*
- *p-MG is easy to implement and can be considered as a valid alternative to h-MG for coarse meshes and high-order of accuracy* *[Franciolini, M., Crivellini, A., Nigro, A. "An implicit discontinuous Galerkin method with reduced memory footprint for the simulation of turbulent flows", accepted at: DLES11 Proceedings, ERCOFTAC series, Springer]*
- *h-MG becomes very attractive for very fine meshes*

  [L. Botti, A. Colombo, F. Bassi, "h-multigrid agglomeration based solution strategies for discontinuous Galerkin discretizations of incompressible flow problems", Journal of Computational Physics, Volume 347, 15 October 2017, Pages 382-415]

  [F. Bassi, L. Botti, A. Colombo, "Agglomeration-based physical frame dG discretizations: An attempt to be mesh free", Math. Models Methods Appl. Sci. 24, 1495 (2014)]

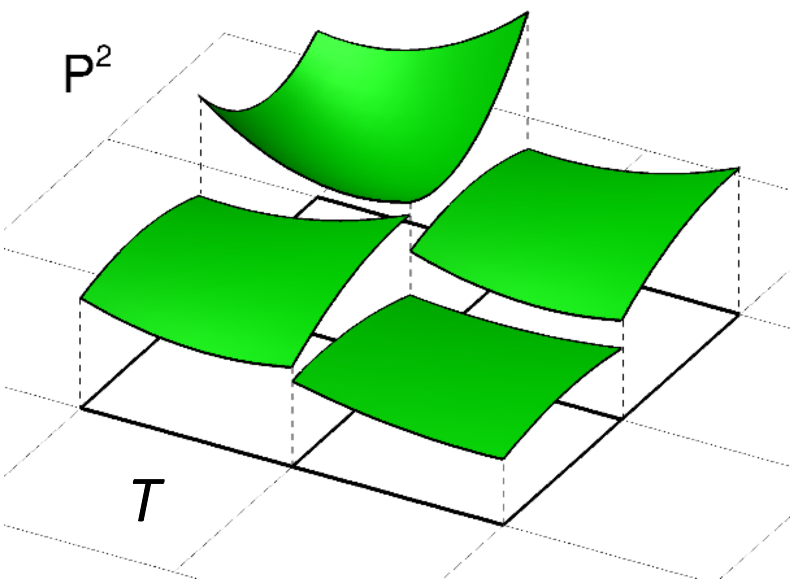


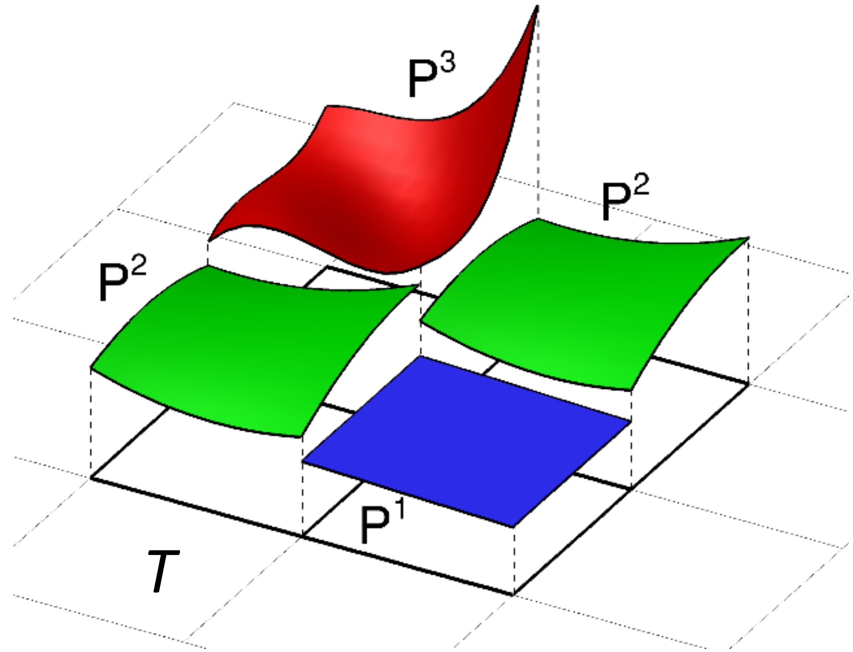
x-MG strategies are always more efficient than SG

Take advantage from DG peculiarities

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

Locally adapt the accuracy of your discretization with DG!





Aside of locally refine/coarsen elements according to some errors estimator, DG methods also allow in a natural way to locally vary the solution accuracy by varying the polynomial degree of the solution in each cell (*p*-adaptation)

Adapt your discretization accuracy within DG - a simple test

The output of interest of simulation are often time averaged quantities (C_L, C_D,…)

Efficient runtime averaging procedure are available within code for statistics purpose

Solution adaptation is driven by error estimators
applead to the **runtime** computed **time-averaged** solution

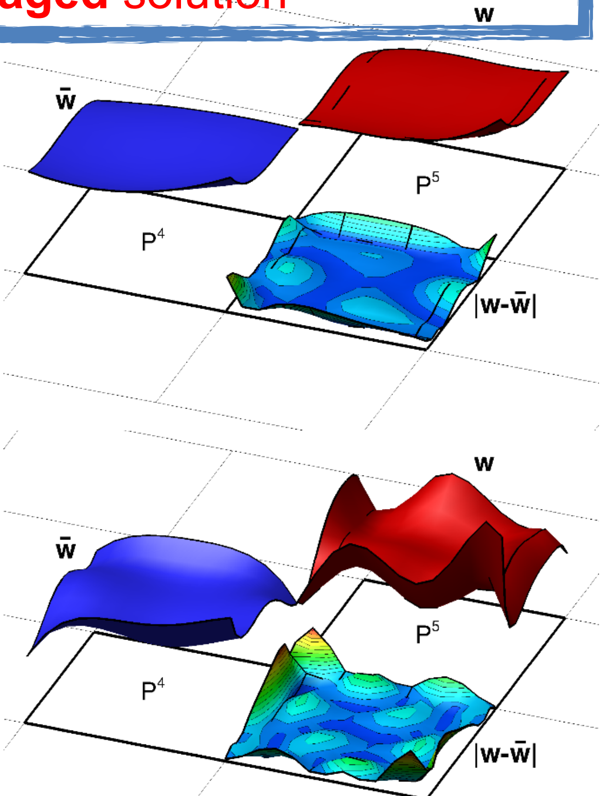
To obtain an efficient estimator both for low- and high- order approximations we combine:

1. based on pressure jumps at interfaces

$$\eta_T^{JMP}(w) = \max_{sides} \max_{j} \left| \frac{w(x_j,t) - w(x_j,t)^+}{w(x_j,t) + w(x_j,t)^+} \right|$$

2. based on the spectral decay of the soution (SDI)

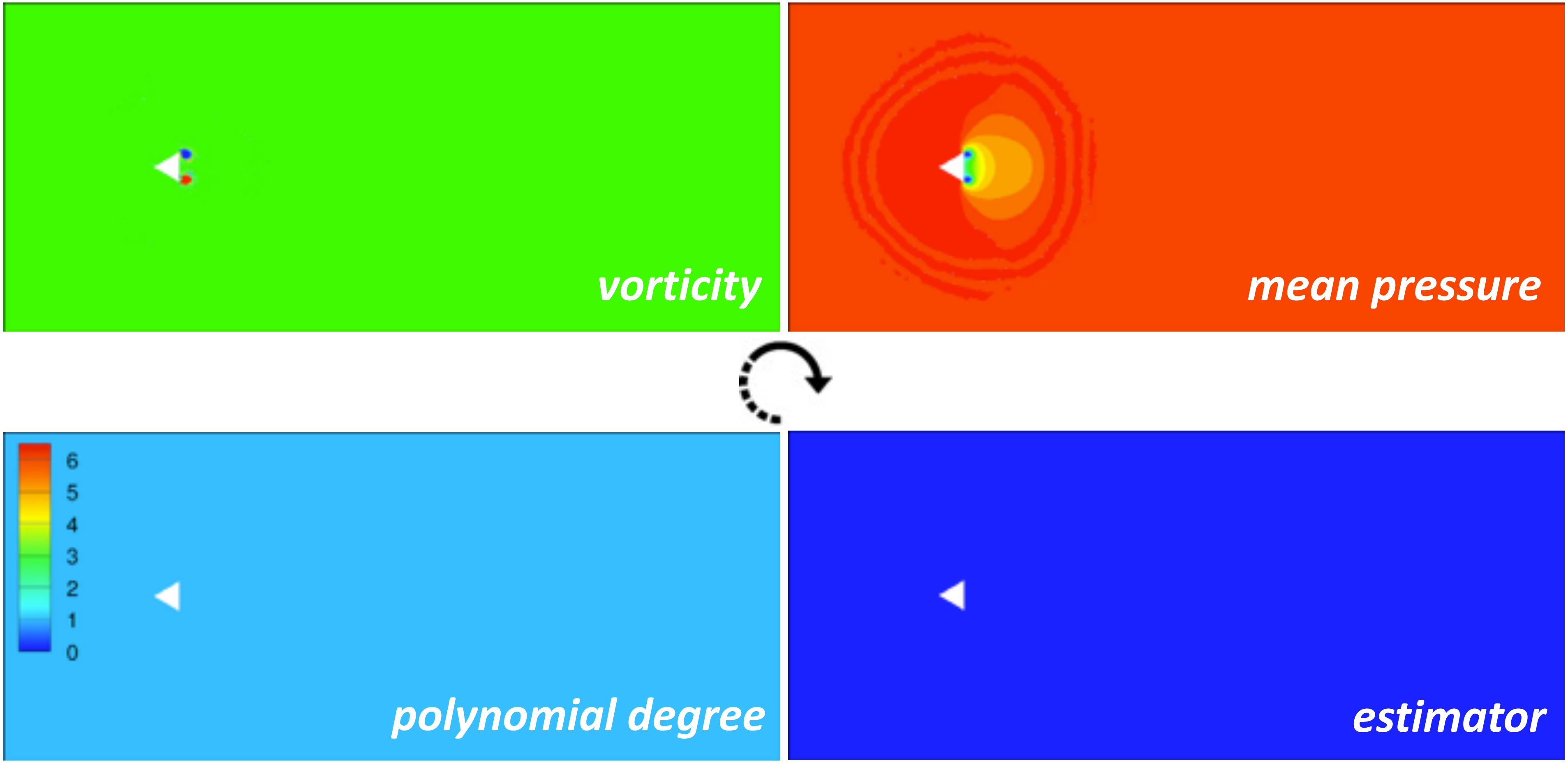
$$\eta_T^{SDI} = \frac{\int_T (w - \bar{w})^2 d\mathbf{x}}{\int_T w^2 d\mathbf{x}}$$



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

Adapt your discretization accuracy within DG - a simple test

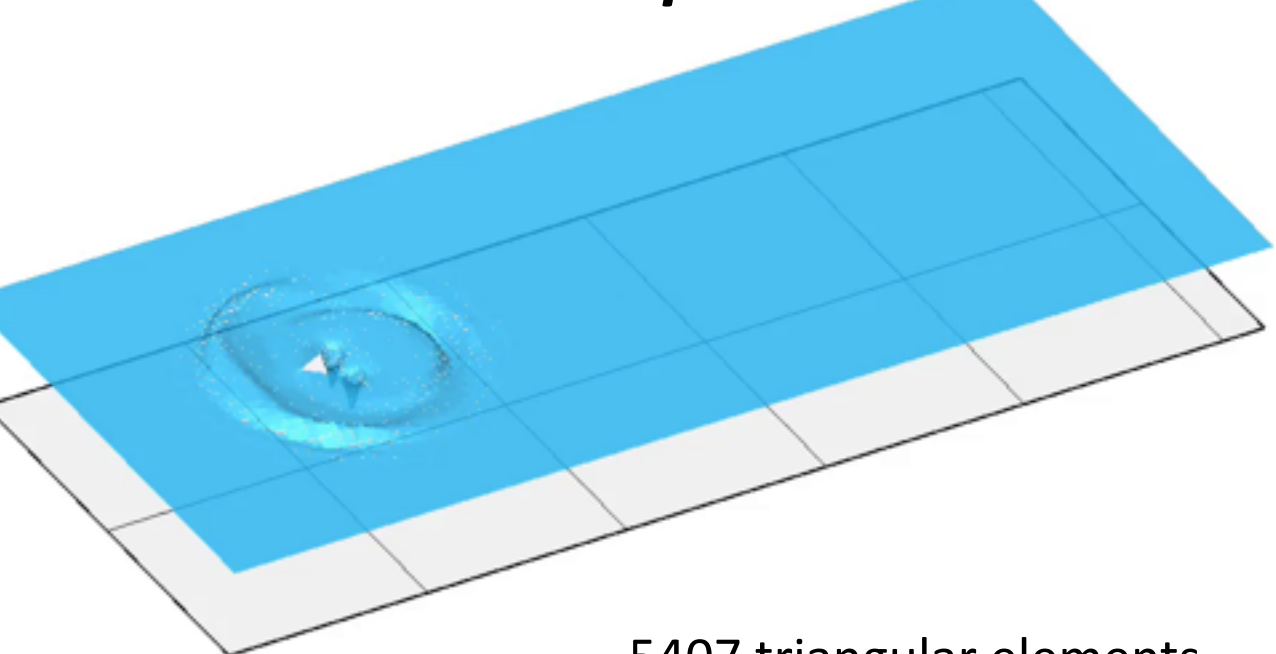
A simple test case but representative of the intended applications, *i.e.* separated flows behind bodies, the problem of an inviscid flow past a triangular cylinder has been considered



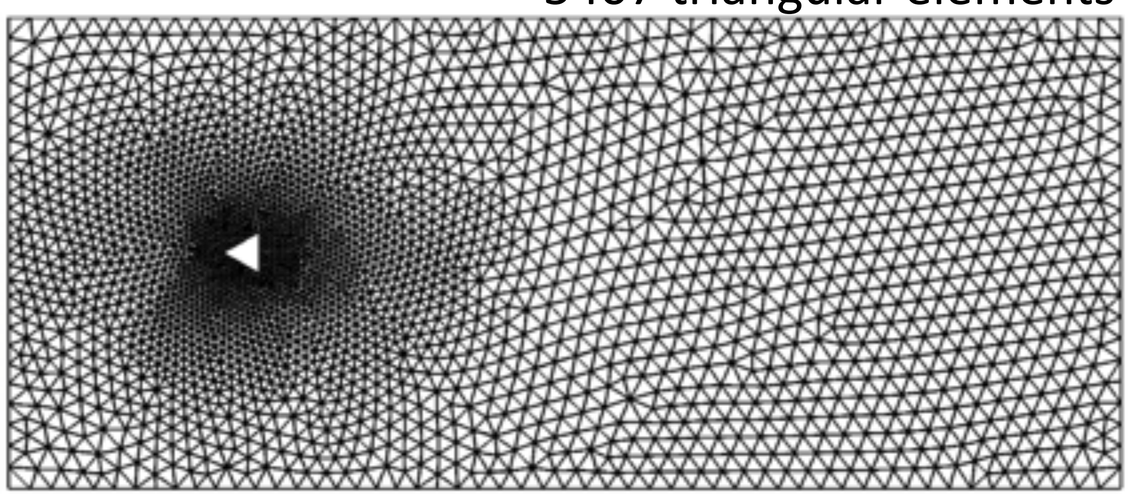
Contribution of Gabriel Manzinali MSc@UNIBG "A p-adaptive Discontinuous Galerkin method for unsteady compressible flows" now @MINES_ParisTech, France

Adapt your discretization accuracy within DG - a simple test

UNIVERSITÀ DEGLI STUDI
DI BERGAMO



5407 triangular elements



1st ADAPTATION +20% DOFs

2nd ADAPTATION +21% DOFs

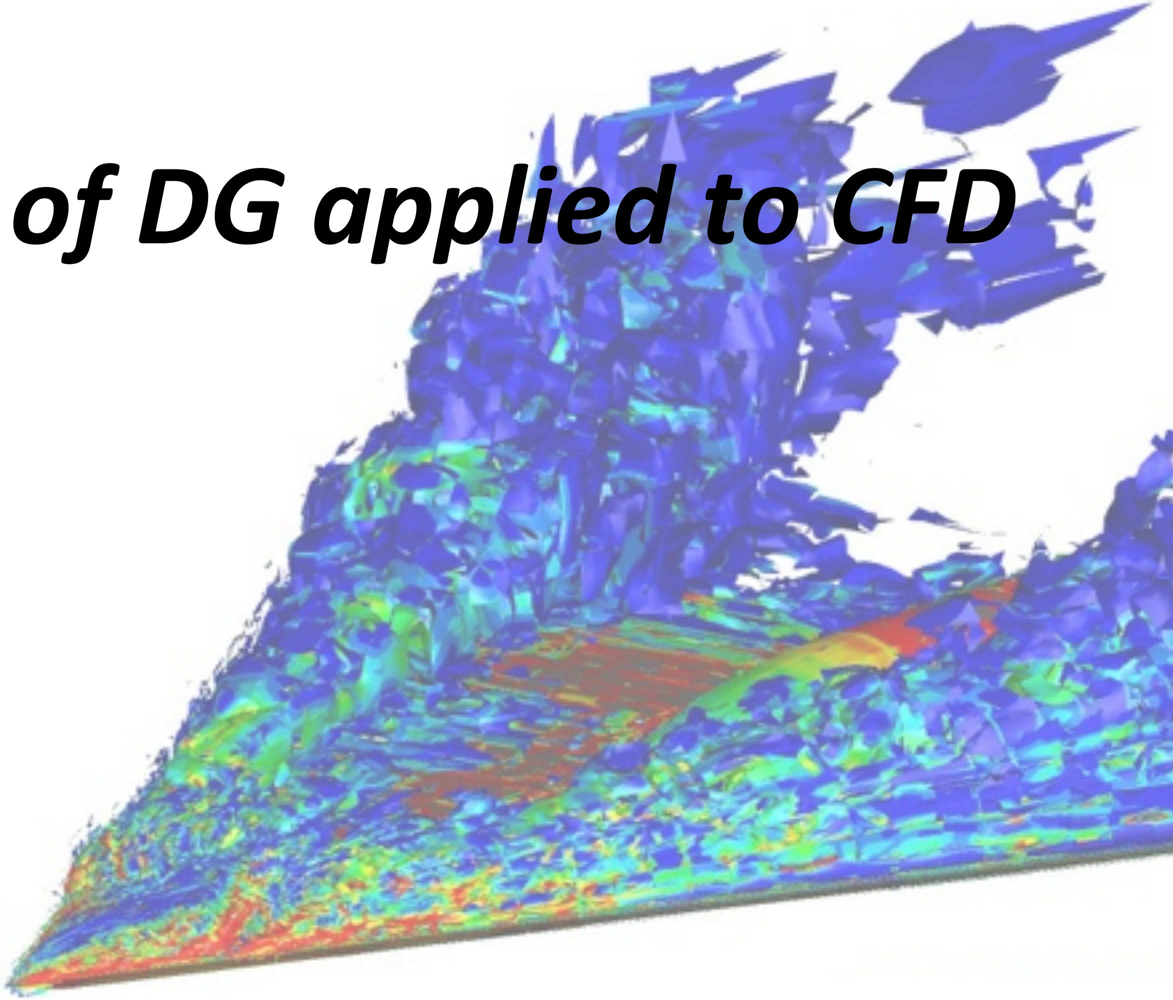
3rd ADAPTATION +18% DOFs

4th ADAPTATION +17% DOFs

5th ADAPTATION +14%DOFs

UNIVERSITÀ DEGLI STUDI
DI BERGAMO

An example of DG applied to CFD



Details of a DG method for the CFD

Modelled turbulent flows governing equations

RANS+k-$\widetilde{\omega}$ (EARSM), X-LES

UNIVERSITÀ DEGLI STUDI DI BERGAMO

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

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

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

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

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

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

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

Details of a DG method for the CFD

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

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

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

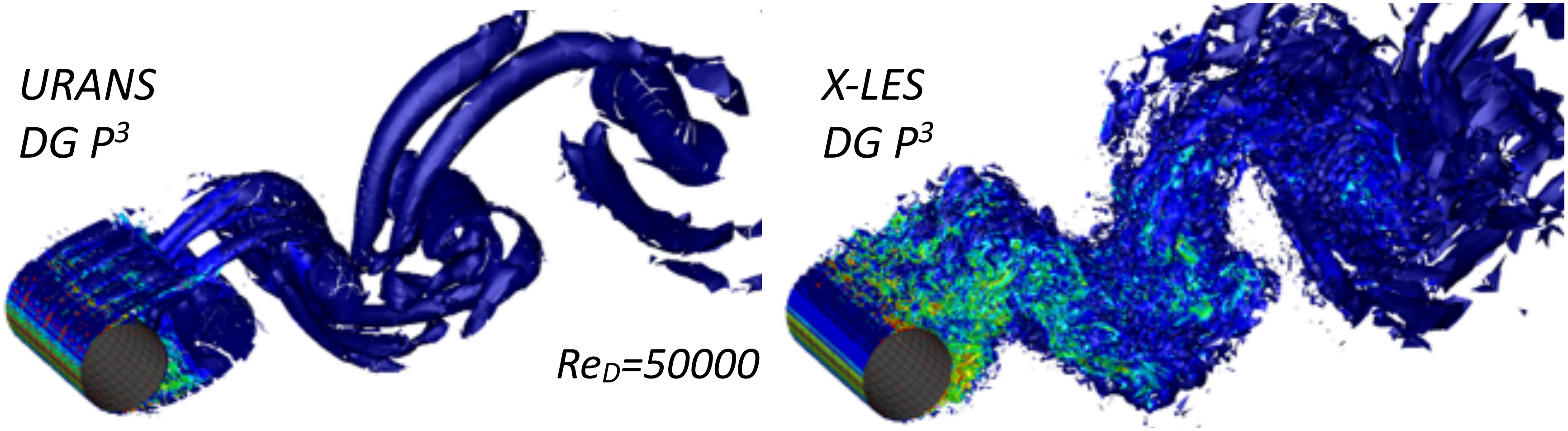
$$D_k = \beta^*\rho\overline{k}\hat{\omega} \qquad D_\omega = \beta\rho\overline{k}e^{\widetilde{\omega}_r} \qquad C_D = \sigma_d\frac{\rho}{e^{\widetilde{\omega}_r}}\max\left(\frac{\partial k}{\partial x_k}\frac{\partial\widetilde{\omega}}{\partial x_k}, 0\right)$$

where

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

Why a hybrid RANS-LES model?

For those high Reynolds number flows where the RANS formulation suffers from prediction limitations, *e.g.* massively separated flows, but LES seems (to date) too demanding



Why the eXtra Large Eddy Simulation (X-LES)? **[Kok et al., 2004]**

- is a hybrid RANS-LES formulation relying on Boussinesq hypothesis for both the prediction of SGS or Reynolds stresses **[Yoshizawa, 1986]**
- LES mode uses a clearly defined dynamic SGS based on *k*-equation
- use of a *k*-*ω* turbulence model integrated to the wall, no wall functions
- a formulation independent from the wall distance
- same high-order (subcell) representation for both LES and RANS zones

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 applied to CFD

Impact of X-LES on source terms and turbulent quantities

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

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

where

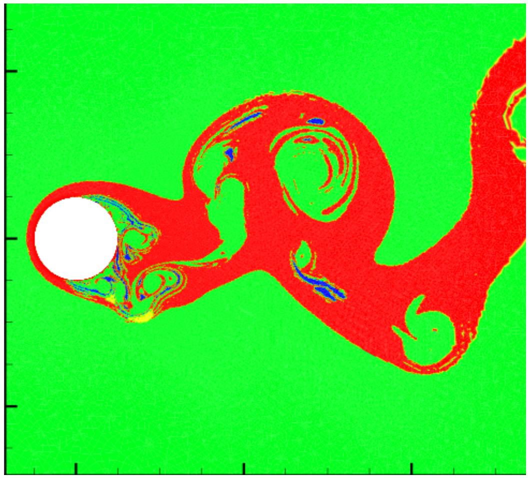
$$\overline{\mu}_t = \alpha^*\frac{\rho\overline{k}}{\hat{\omega}} \qquad \hat{\omega} = \max\left(e^{\widetilde{\omega}_r}, \frac{\sqrt{\overline{k}}}{C_1\Delta}\right) \qquad \overline{k} = \max(0, k)$$

our im

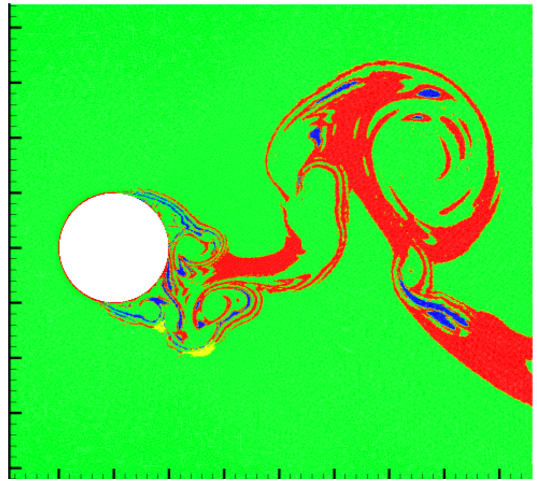
Flexible - acting on the filter width parameter Δ the amount of RANS modeling can be minimized and reduced at the boundary

	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

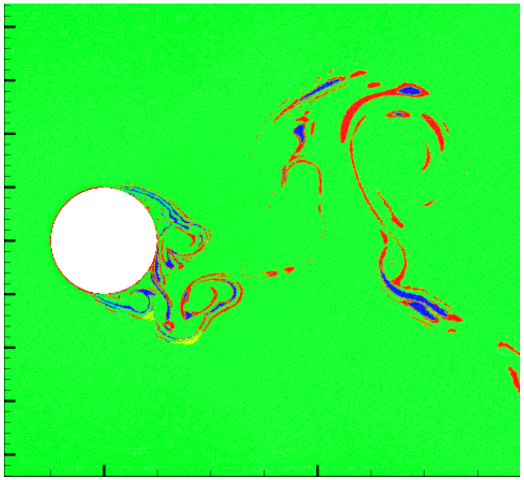
Models distribution: RANS, LES, ILES



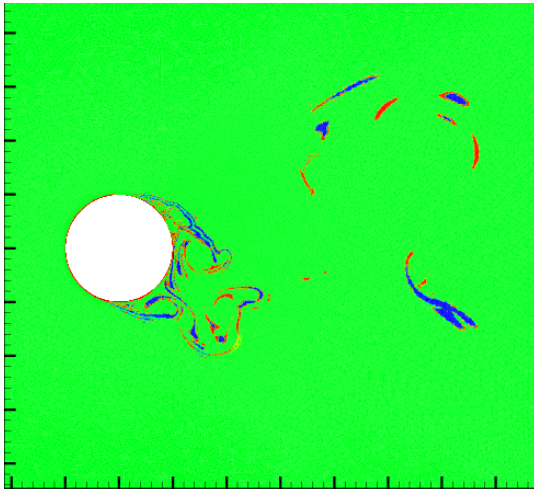
$\Delta = 3.5\times10^3$



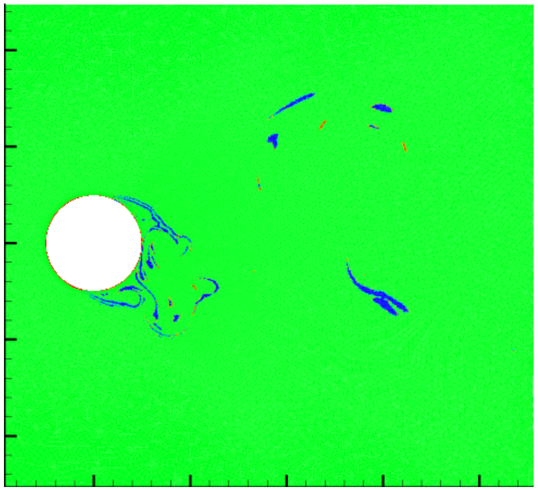
$\Delta = 3.3\times10^3$



$\Delta = 3\times10^3$

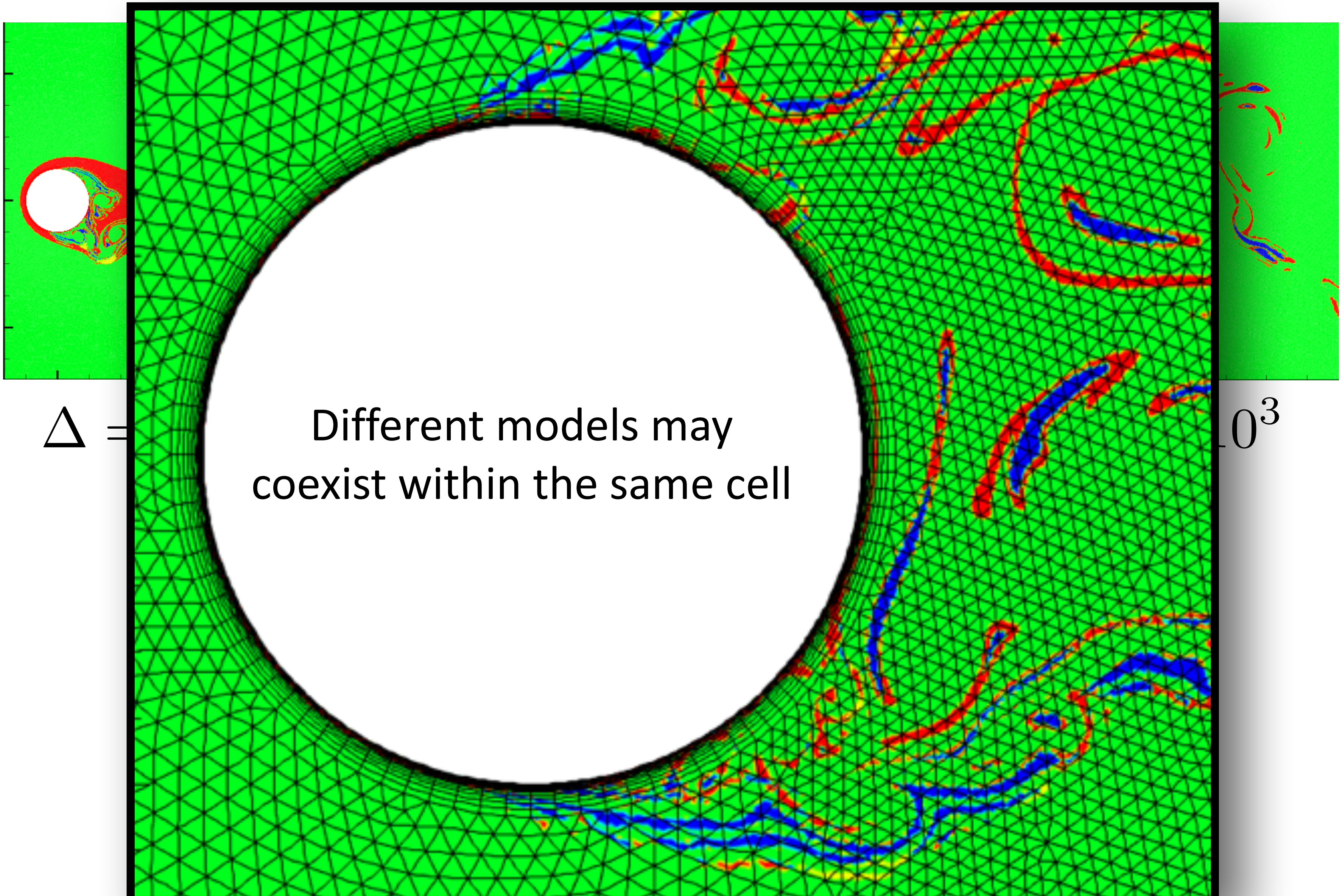


$\Delta = 2\times10^3$



$\Delta = 1\times10^3$

Models distribution: RANS, LES, ILES



$\Delta = 2\times10^3$ $\Delta = 1\times10^3$

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

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

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

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

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

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

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

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

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

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 viscous term

The BR2 scheme in a nutshell [Bassi and Rebay, 1997]

Some definitions…

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

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

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

We introduce the local lifting operator

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

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

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

DG discretization of the viscous term

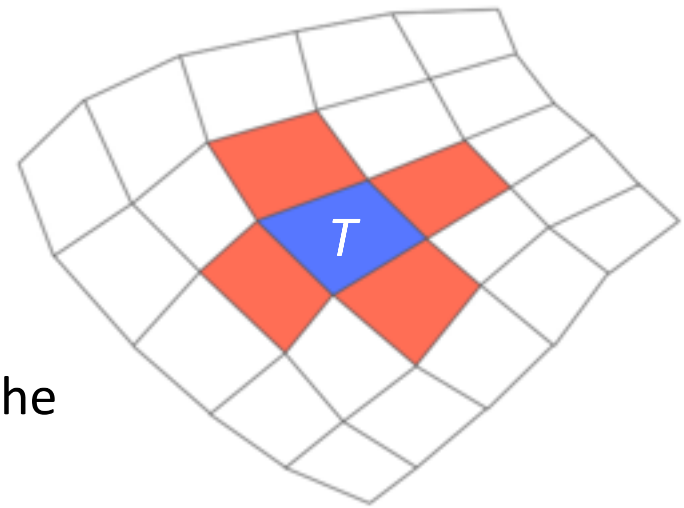
The BR2 scheme in a nutshell [Bassi and Rebay, 1997]

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

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



gradients are “corrected” with the local and and global lifting operators of the solution’s jump in the volume and surface terms respectively

$$\mathbf{z}_h = \nabla \mathbf{w}_h + \mathbf{r}(\llbracket u_h \rrbracket) \qquad \mathbf{z}_{h_F} = \nabla \mathbf{w}_h + \eta_F \mathbf{r}_F(\llbracket u_h \rrbracket)$$

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

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

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

where $\mathbf{\Gamma}^{-1} \stackrel{\text{def}}{=} (\gamma_{ij})^{-1}$ 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?

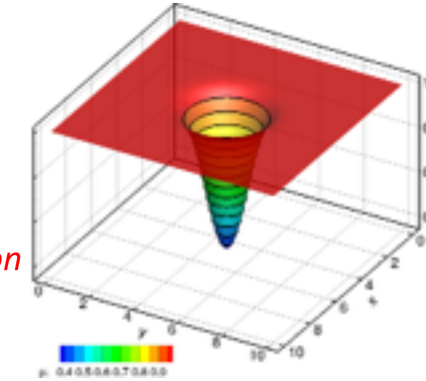
Several high-order temporal schemes are implemented

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

} *non-linear systems solution*

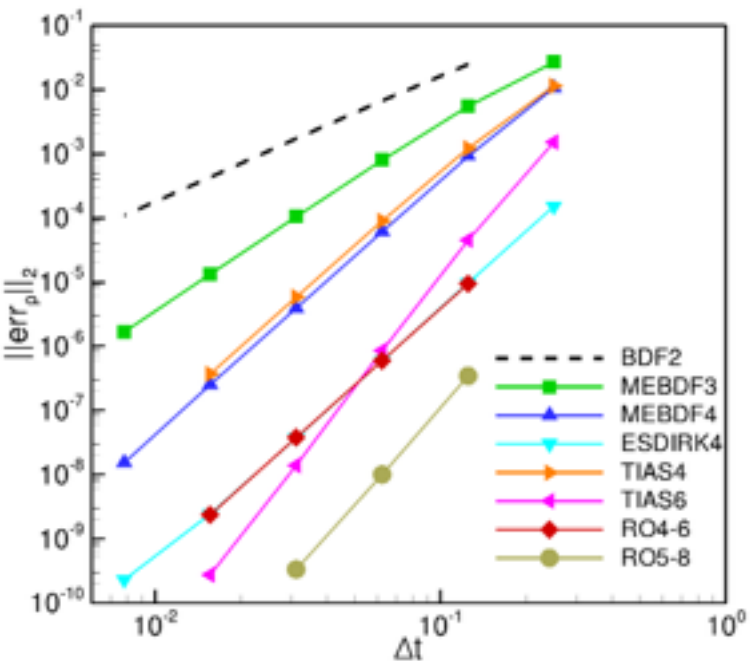
- linearly implicit Rosenbrock method

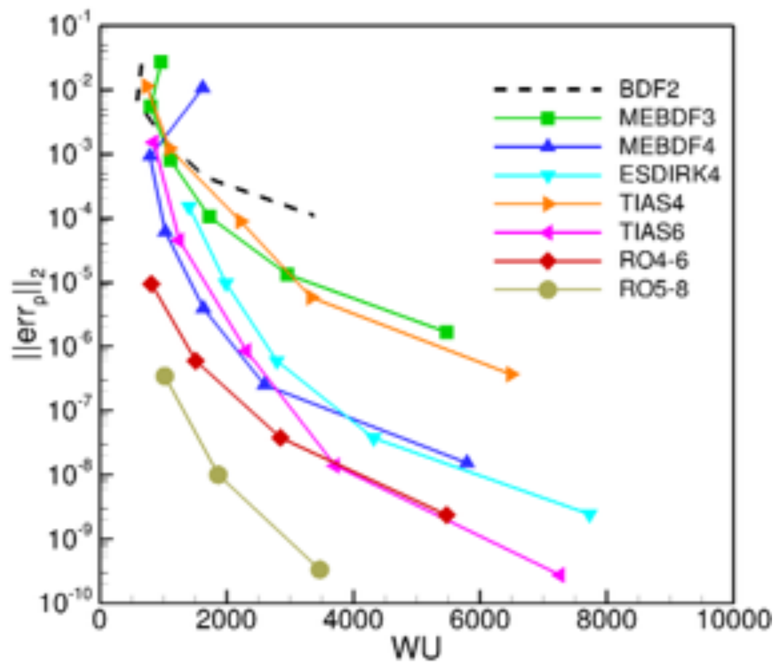
linear systems solution (here via GMRES)



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



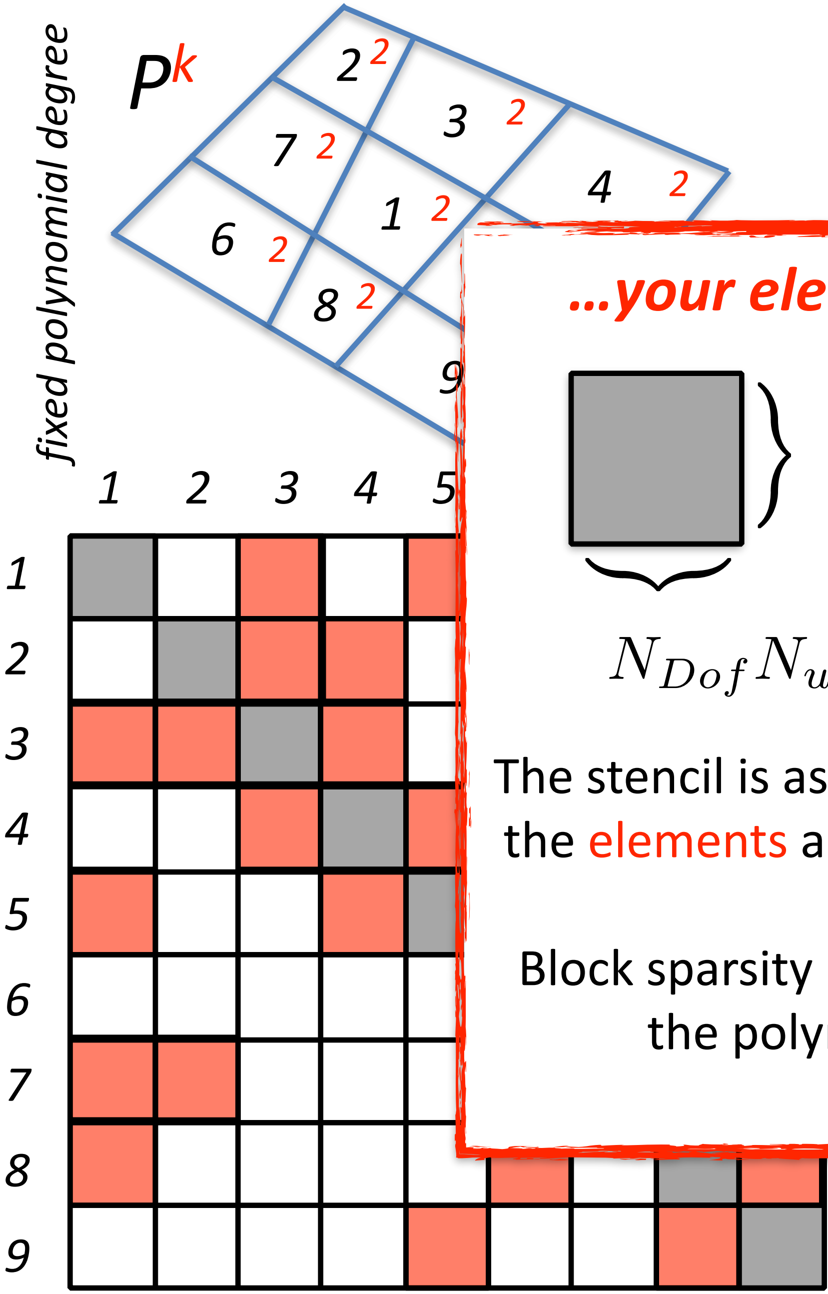


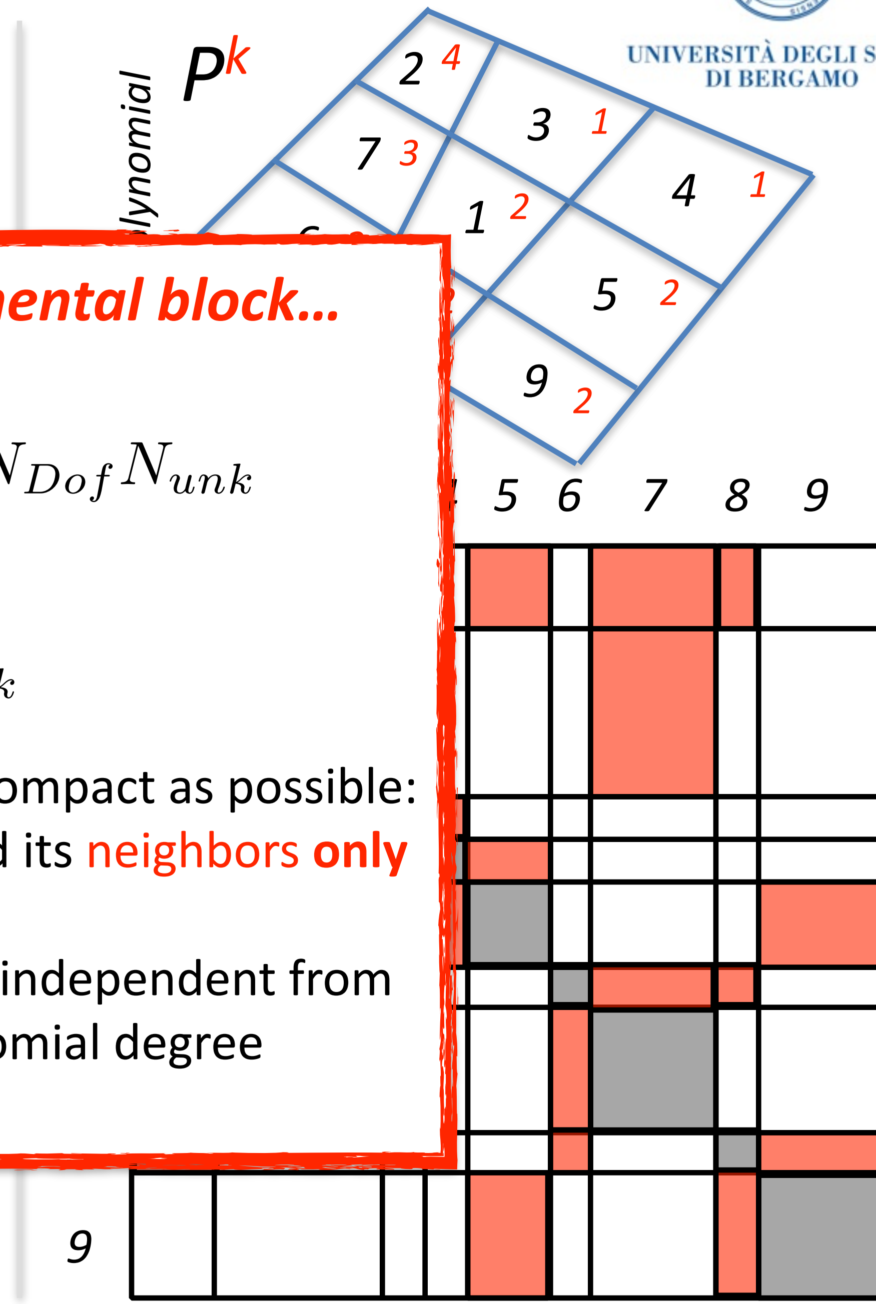
Convection of an isentropic vortex

P^6 solution on 50X50 el.

How the implicit operator looks like?

UNIVERSITÀ DEGLI STUDI
DI BERGAMO





...your elemental block...

$N_{Dof} N_{unk}$

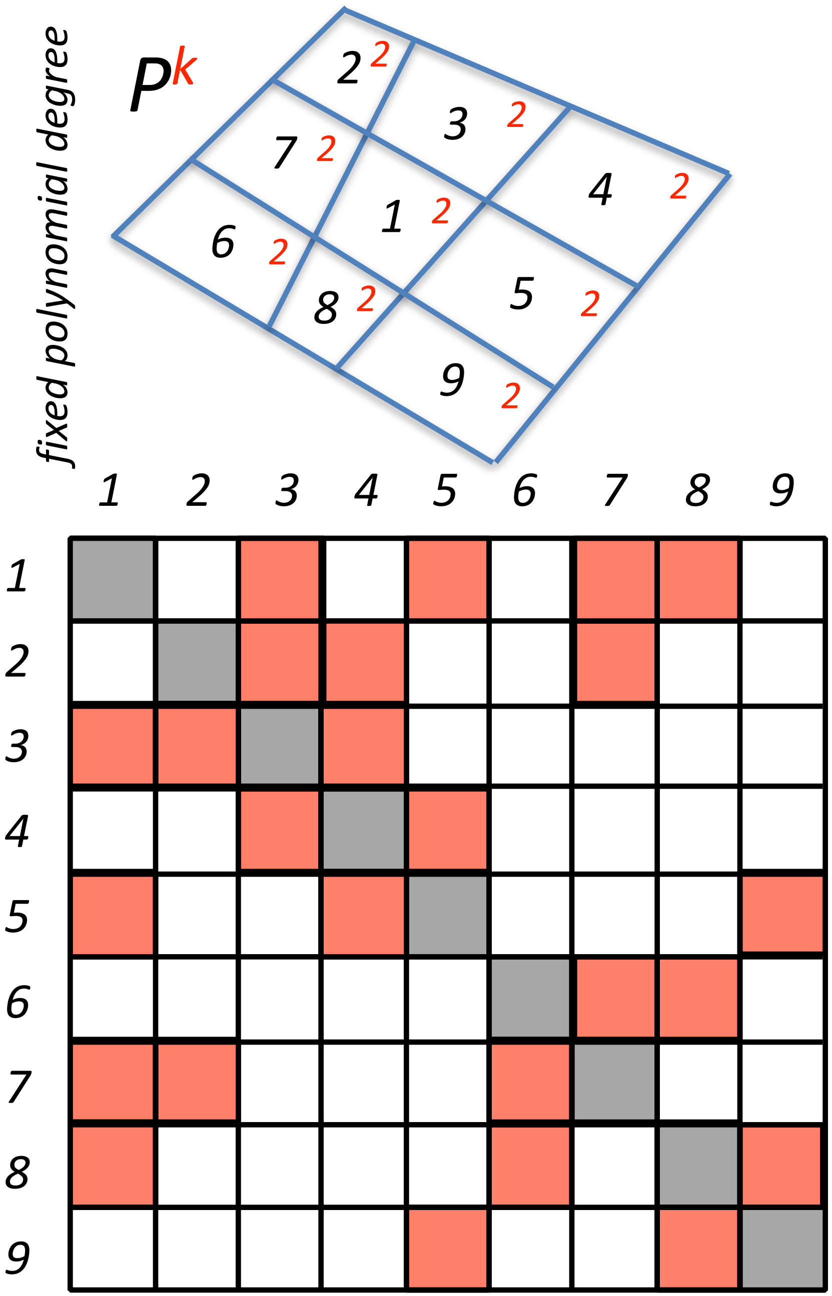
$N_{Dof} N_{unk}$

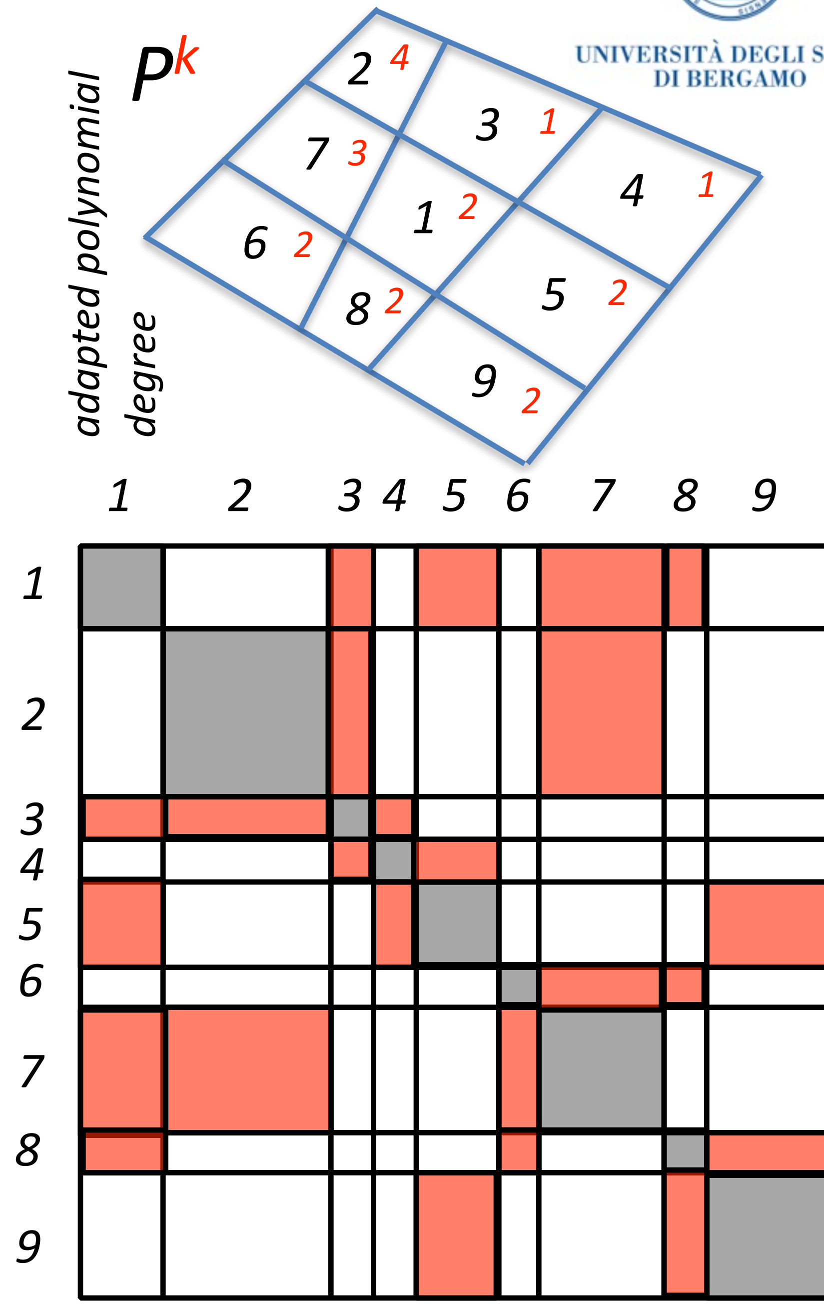
The stencil is as compact as possible: the elements and its neighbors **only**

Block sparsity is independent from the polynomial degree

How the implicit operator looks like?

UNIVERSITÀ DEGLI STUDI DI BERGAMO





Some results: transonic compressor rotor

RANS+k-ω (initialization and comparison)

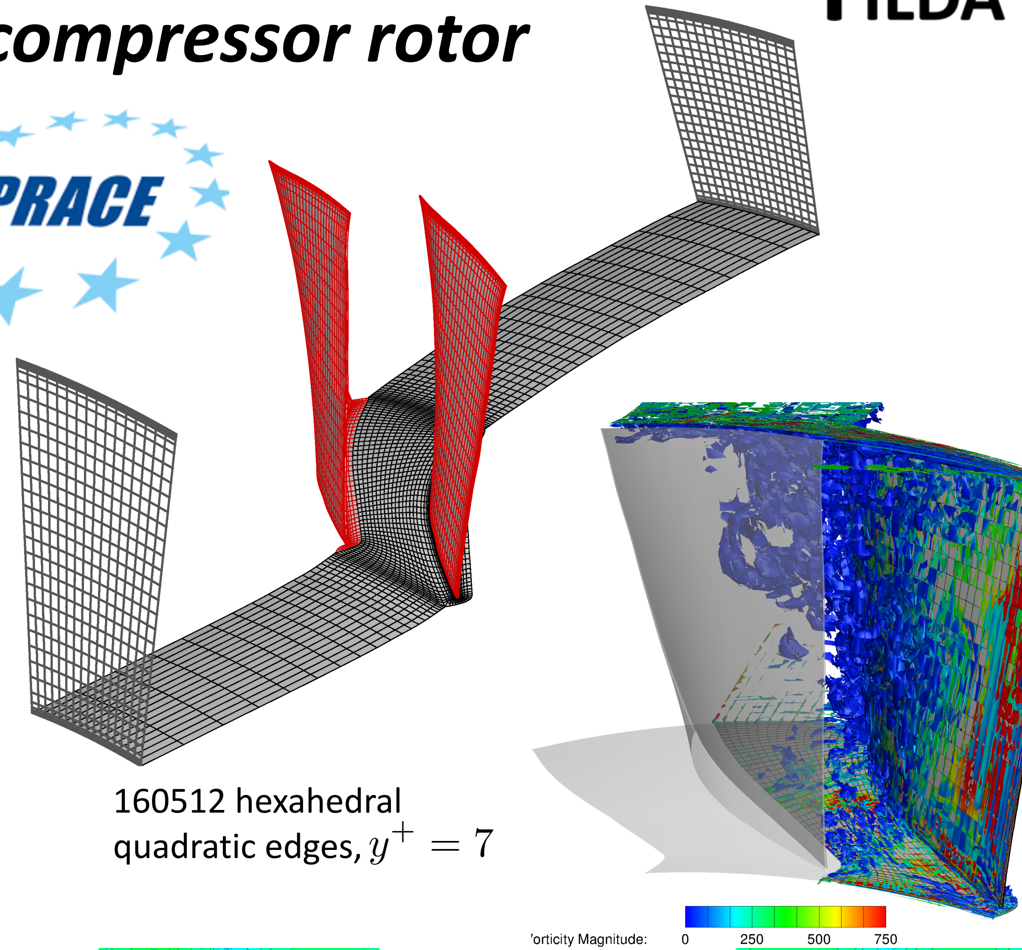
- DG P² (98% of the chocked mass flow)
- DG P³ Performance map

X-LES

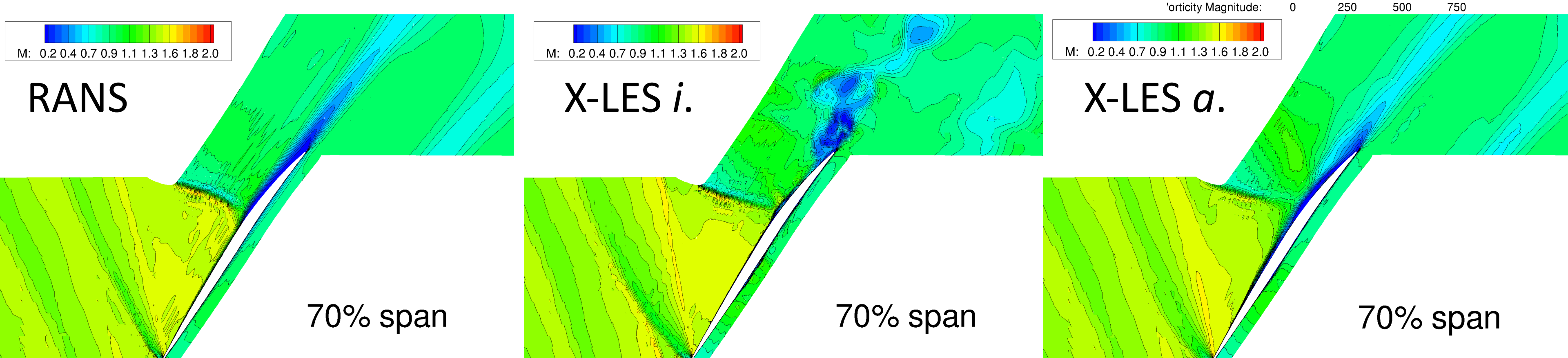
- DG P² X-LES computation using the 3rd order 3 stages Rosenbrok (ROS3P)
- DG P³ (ROS3P) on going (~98%)
- Filter width Δ=1°-3

Boundary conditions

- p_{01} = 101325Pa, T_{01} = 288K, Tu_1 = 3%
- ω= 1800rad/s
- α_1 = 0°



160512 hexahedral quadratic edges, $y^+ = 7$



Some results: transonic compressor rotor

RANS+k-ω (initialization and comparison)

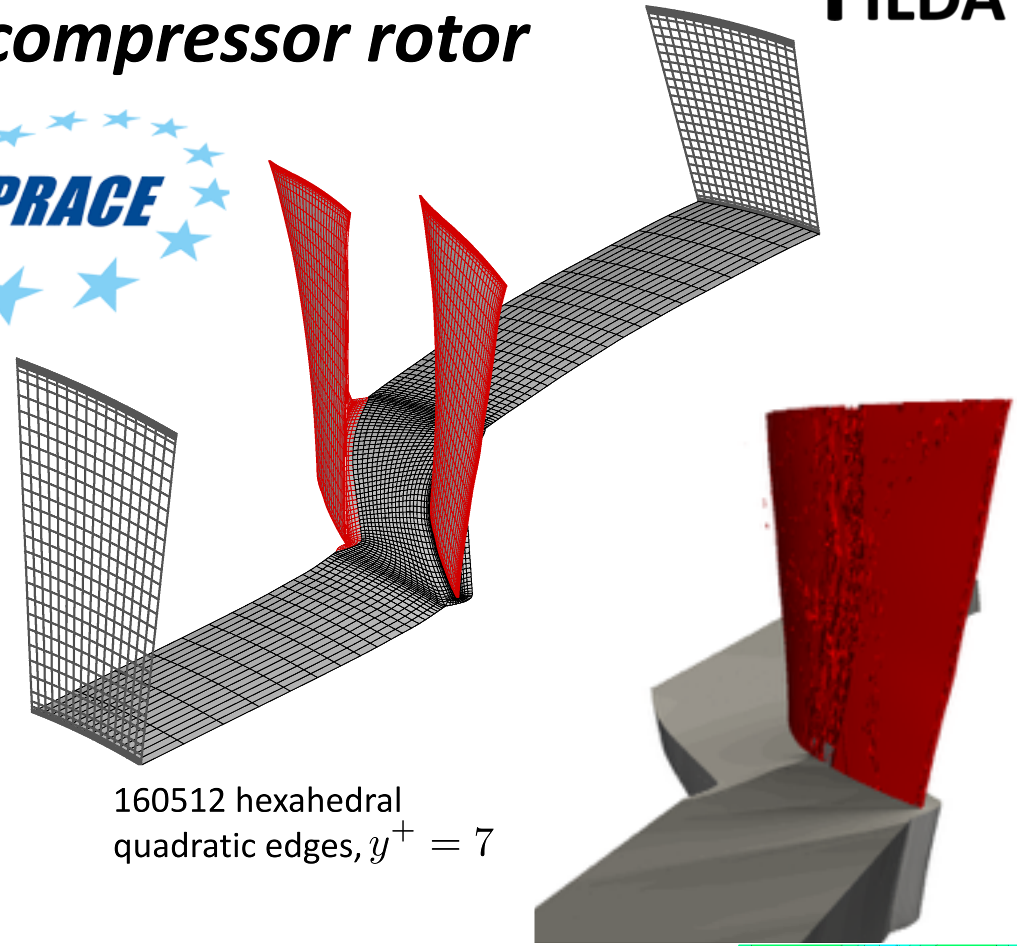
- DG P^2 (98% of the chocked mass flow)
- DG P^3 Performance map

X-LES

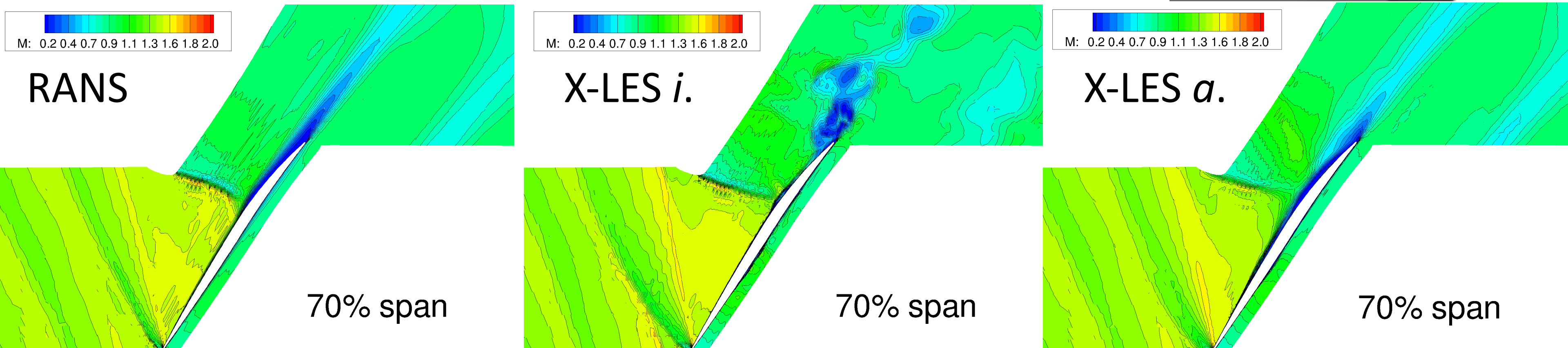
- DG P^2 X-LES computation using the 3rd order 3 stages Rosenbrok (ROS3P)
- DG P^3 (ROS3P) on going (~98%)
- Filter width Δ=1°-3

Boundary conditions

- p_{01} = 101325Pa, T_{01} = 288K, Tu_1 = 3%
- ω= 1800rad/s
- α_1 = 0°



160512 hexahedral quadratic edges, $y^+ = 7$



Some results: transonic compressor rotor

RANS+k-ω (initialization and comparison)

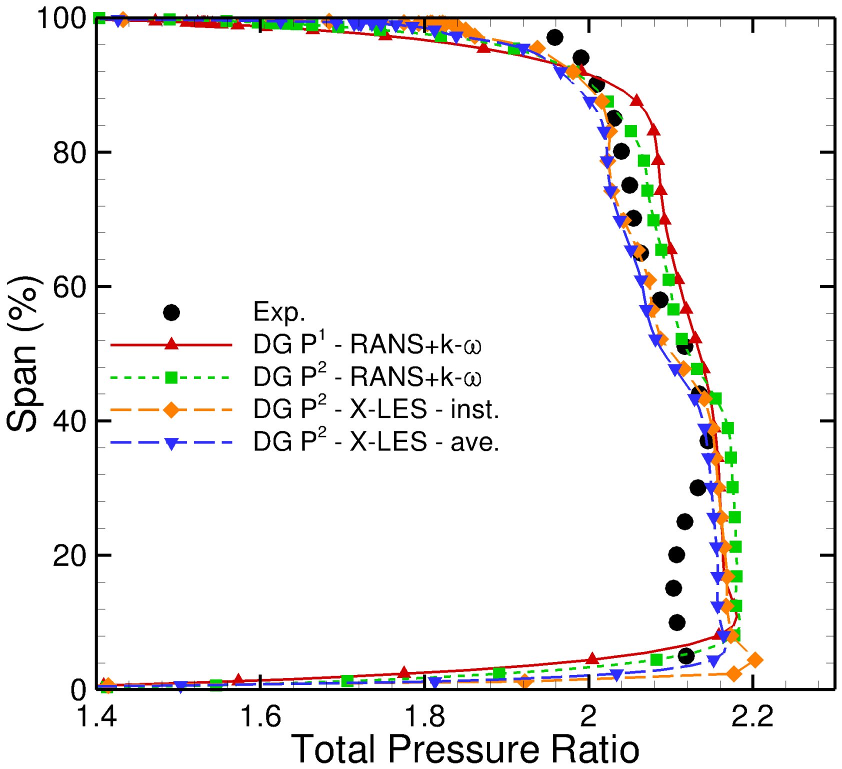
- DG P^2 (98% of the chocked mass flow)
- DG P^3 Performance map

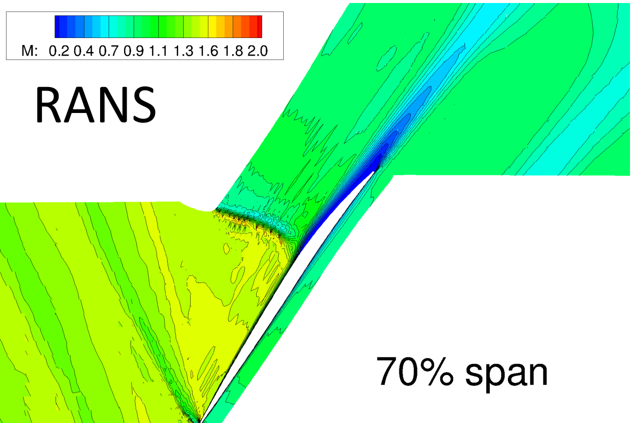
X-LES

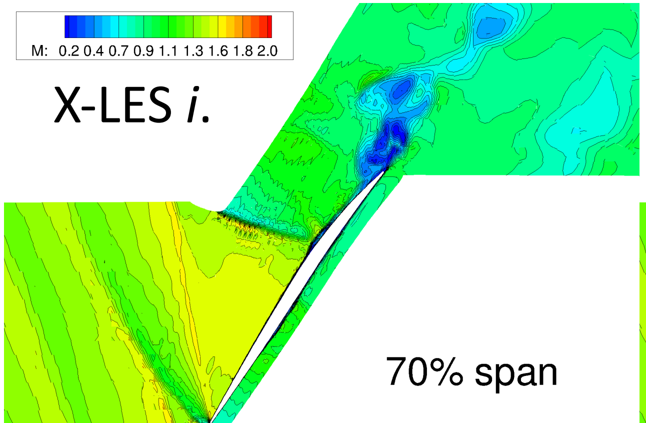
- DG P^2 X-LES computation using the 3rd order 3 stages Rosenbrok (ROS3P)
- DG P^3 (ROS3P) on going (~98%)
- Filter width Δ=1°-3

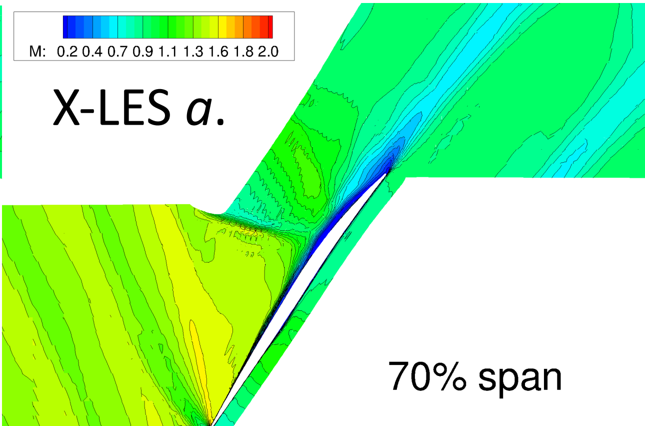
Boundary conditions

- p_{01} = 101325Pa, T_{01} = 288K, Tu_1 = 3%
- ω= 1800rad/s
- α_1 = 0°





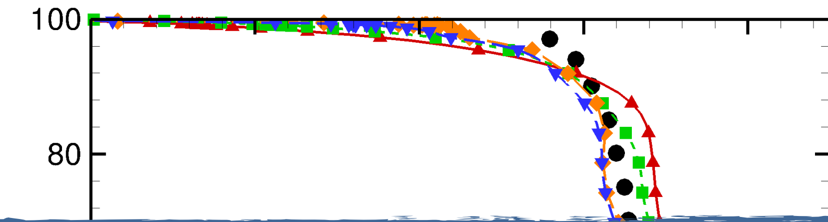




Some results: transonic compressor rotor

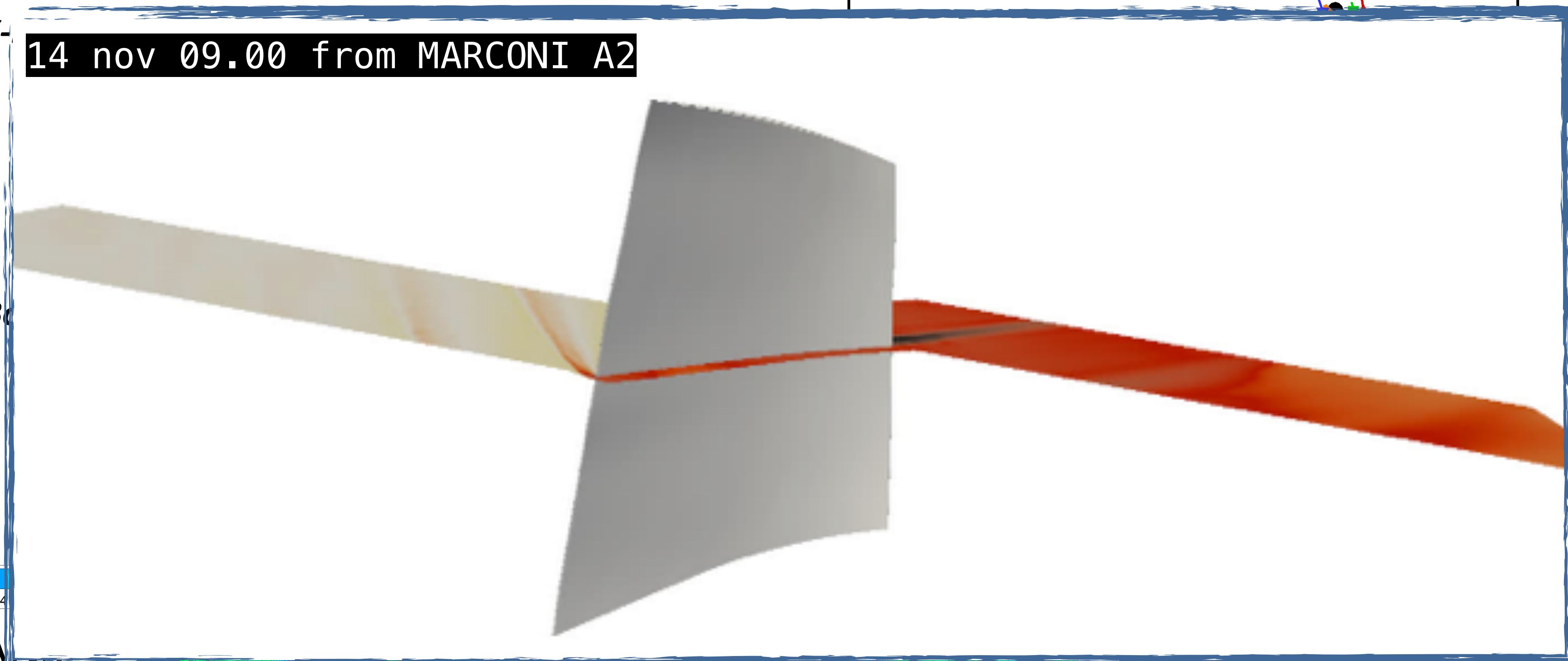
RANS+k-ω (initialization and comparison)

- DG P^2 (98% of the chocked mass flow)
- DG P^3 Performance map

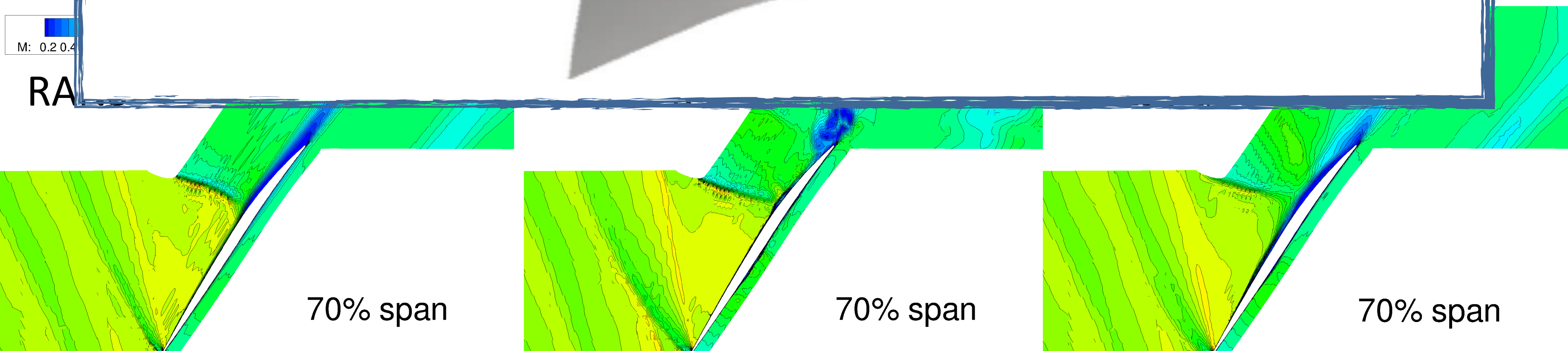


X-

Bo



RA



Grazie dell’attenzione

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.

[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.