6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11-15 June 2018, Glasgow, UK

A *P*-ADAPTIVE IMPLICIT DISCONTINUOUS GALERKIN METHOD FOR THE UNDER-RESOLVED SIMULATION OF COMPRESSIBLE TURBULENT FLOWS

A. Colombo[1], G. Manzinali[1], A. Ghidoni[2], G. Noventa[2], M. Franciolini[3], A. Crivellini[3] and F. Bassi[1]

[1] University of Bergamo, Department of Engineering and Applied Sciences, viale Marconi, 5 - 24044 Dalmine (BG), Italy, {alessandro.colombo, francesco.bassi}@unibg.it, gabriel.manzinali@mines-paristech.fr

[2] University of Brescia, Department of Mechanical and Industrial Engineering, via Branze, 38 - 25123, Brescia, Italy, {antonio.ghidoni, gianmaria.noventa}@unibs.it

[3] Marche Polytechnic University, Department of Industrial Engineering and Mathematical Science, via Brecce Bianche, 12 - 60131 Ancona, Italy, m.franciolini@pm.univpm.it, a.crivellini@univpm.it

Key words: Discontinuous Galerkin methods, p-adaptation, Compressible ILES

Abstract. In the last decades Computational Fluid Dynamics has become a widespread practice in several industrial fields, *e.g.*, aerodynamics, aeroacoustic. The growing need of high-fidelity flow simulations for the accurate determination of problem-specific quantities paved the way to higher-order methods such as the discontinuous Galerkin (DG) method. In this context, the industrial interest is strongly promoting the development of more and more efficient high-order CFD solvers. In this work we exploit some techniques, *i.e.* p-adaptation, quadrature reduction and load balancing, to enhance the computational efficiency of an existing DG code. The accuracy and efficiency of our approach will be assessed by computing the implicit Large Eddy Simulation of the flow past a circular cylinder at Reynolds number $\mathrm{Re} = 3900$, and around a NACA0018 airfoil at Reynolds number $\mathrm{Re} = 10000$ and angle of attack $\alpha = 15°$.

1 INTRODUCTION

In the last decades Computational Fluid Dynamics (CFD) has become a widespread practice in several industrial fields, *e.g.* aerodynamics, aeroacoustic, turbomachinery. The growing need of high-fidelity flow simulations for the accurate determination of problem-specific quantities paved the way to higher-order methods such as the Discontinuous Galerkin (DG) methods. DG methods, if compared to standard industrial code, show some deficiency in terms of computational efficiency. However, the industrial interest is strongly promoting research efforts to devise more efficient high-order CFD solvers. The

aim of this paper is to describe an approach to enhance the computational efficiency of an existing DG code [1], based on p-adaptation, quadrature reduction and load balancing.

A p-adaptation strategy that allows the variation of the polynomial degree of the solution between the elements is adopted, obtaining a sensible reduction of the simulation CPU time and memory, while not spoiling at all the high accuracy required by this class of simulations. In particular, being interested in unsteady flows computations, such as Direct Numerical Simulation (DNS) and Large Eddy Simulation (LES), we will investigate an adaptation procedure suitable for time-dependent problems. Adaptation is here driven by two simple indicators, based on interface pressure jumps and on the decay of the coefficients of the modal expansion. These sensors are coupled to guarantee a reasonable behaviour both for high- and low-degree polynomial approximations. Moreover, the degree of exactness of quadrature rules is adapted on the computational domain to avoid over integration of straight sided elements, and runtime load-balancing is applied, based on the Metis library capability to generate "weighted" graphs, to handle the degrees of freedom unbalance on each partition due to the adaptation algorithm.

The robustness and efficiency of the proposed approach is evaluated by computing the ILES of the flow past a circular cylinder at Reynolds number $\mathrm{Re} = 3900$, and around a NACA0018 airfoil at Reynolds number $\mathrm{Re} = 10000$ and angle of attack $\alpha = 15°$.

In Section 2 we briefly introduce the DG space and the time discretizations. Section 3 describes the definition of orthonormal hierarchical polynomial basis and the possible approaches to their evaluation, while Section 4 the error estimators used to drive the adaptaton. Sections 5 and 6 describe the techniques adopted to adapt the degree of exactness of the quadrature rules and to balance the load of each processor. Finally, in Section 7 a strategy for the p-adaptation is proposed and applied to different unsteady compressible test cases.

2 THE NUMERICAL FRAMEWORK

The Navier-Stokes equations for the m variables in d dimensions can be written in compact form as

$$\mathbf{P}(\mathbf{w})\frac{\partial \mathbf{w}}{\partial t} + \nabla \cdot \mathbf{F}_c(\mathbf{w}) + \nabla \cdot \mathbf{F}_v(\mathbf{w}, \nabla \mathbf{w}) = \mathbf{0}, \tag{1}$$

where $\mathbf{w} \in \mathbb{R}^m$ is the unknown solution vector, $\mathbf{F}_c, \mathbf{F}_v \in \mathbb{R}^m \otimes \mathbb{R}^d$ are the convective and viscous flux functions, and $\mathbf{P}(\mathbf{w}) \in \mathbb{R}^m \otimes \mathbb{R}^m$ is a transformation matrix. Employing the conservative variables $\mathbf{w}_c = [\rho, \rho u_i, \rho E]^T$ for compressible flows $\mathbf{P}$ reduces to the identity matrix ($\mathbf{P} = \mathbf{I}$). By multiplying Eq. (1) by an arbitrary smooth test function $\mathbf{v} = \{v_1, \ldots, v_m\}$, and integrating by parts, we obtain the weak formulation

$$\int_\Omega \mathbf{v} \cdot \left(\mathbf{P}(\mathbf{w}) \frac{\partial \mathbf{w}}{\partial t} \right) d\mathbf{x} - \int_\Omega \nabla \mathbf{v} : \mathbf{F}(\mathbf{w}, \nabla \mathbf{w})\, d\mathbf{x} + \int_{\partial\Omega} \mathbf{v} \otimes \mathbf{n} : \mathbf{F}(\mathbf{w}, \nabla \mathbf{w})\, d\sigma = \mathbf{0}, \tag{2}$$

where $\mathbf{F}$ is the sum of the convective and viscous flux functions and $\mathbf{n}$ is the unit vector normal to the boundary.

To discretize Eq. (2) we replace the solution $\mathbf{w}$ and the test function $\mathbf{v}$ with a finite element approximation $\mathbf{w}_h$ and a discrete test function $\mathbf{v}_h$, respectively, where $\mathbf{w}_h$ and $\mathbf{v}_h$

belong to the space $\mathbf{V}_h \overset{\text{def}}{=} [\mathbb{P}_d^k(\mathcal{T}_h)]^m$. $\mathbb{P}_d^k(T)$ denotes the restriction to non overlapping arbitrary shaped mesh elements $T \in \mathcal{T}_h$ of the polynomial functions of d variables and total degree $\leq k$. For each of the m equations of system (2), and without loss of generality, we choose the set of test and shape functions in any element T coincident with the set of orthogonal and hierachical basis functions in that element. The DG discretization of the viscous fluxes is based on the BR2 scheme, proposed in [2], while the convective numerical flux is computed from the solution of local Riemann problems in the normal direction at each quadrature point on elements faces.

By assembling together all the elemental contributions of the DG discretization, the system of Ordinary Differential Equations (ODEs) governing the evolution in time of the discrete solution can be written as

$$\frac{d\mathbf{W}}{dt} = \widetilde{\mathbf{R}}\left(\mathbf{W}\right), \tag{3}$$

with

$$\widetilde{\mathbf{R}}\left(\mathbf{W}\right) = \mathbf{M}^{-1}\left(\mathbf{W}\right)\mathbf{R}\left(\mathbf{W}\right), \tag{4}$$

where $\mathbf{W}$ is the global vector of unknown degrees of freedom, $\mathbf{M}$ is the global block diagonal mass matrix, and $\mathbf{R}\left(\mathbf{W}\right)$ is the vector of residuals. In this work we integrate in time by using the linearly implicit (Rosenbrock-type) Runge-Kutta schemes [3].

3 EFFICIENT EVALUATION OF THE BASIS FUNCTIONS

In this section we briefly describe the approach proposed in [4] to build a set of orthonormal and hierarchical basis functions. Let $\hat{\Phi}_T^k = \left\{\hat{\phi}_i^T\right\}_{i\in\left\{1,\dots,N_{dof}^T\right\}}$ denote an initial set of monomials defined in a reference frame relocated in the element barycentre and aligned with the principal axes of inertia of T. To obtain a new set of orthonormal basis functions, $\Phi_T^k = \left\{\phi_i^T\right\}_{i\in\left\{1,\dots,N_{dof}^T\right\}}$, we apply Algorithm 1 (modified Gram-Schmidt procedure, MGS) for all $T \in \mathcal{T}_h$, where $\left(\hat{\phi}_i^T, \phi_j^T\right)_T$ is the evaluation of the inner product, r_{ii}^T and r_{ij}^T the coefficients of the MGS algorithm.

Algorithm 1 Modified Gram-Schmidt orthogonalization algorithm

for $i = 1$ **to** N_{dof}^T **do**
  for $i = 1$ **to** N_{dof}^T **do**
    $r_{ij}^T \leftarrow \left(\hat{\phi}_i^T, \phi_j^T\right)_T$
    $\hat{\phi}_i^T \leftarrow \hat{\phi}_i^T - r_{ij}^T \phi_j^T$ {remove the projection of $\hat{\phi}_i^T$ onto ϕ_j^T }
  end for
  $r_{ii}^T \leftarrow \sqrt{\left(\hat{\phi}_i^T, \phi_j^T\right)_T}$
  $\hat{\phi}_i^T \leftarrow \hat{\phi}_i^T / r_{ii}^T$ {normalize}
  $\phi_i^T \leftarrow \hat{\phi}_i^T$
end for

The basis functions can be evaluated following three different strategies, characterized by different computational cost and memory:

- Full storing (PreShape). For each element and face the basis functions and, if needed, their derivatives are evaluated at each Gauss point, during pre-processing and stored in memory.
- Orthonormalization coefficients storing (PreCoef). The monomials evaluation, their orthonormalization and the computation of the orthonormalization coefficients are performed separately. The coefficients are evaluated and stored during pre-processing for each element, according to the corresponding polynomial degree. Then, the monomials are evaluated on-the-fly and orthonormalized, using the pre-computed coefficients.
- On-the-fly evaluation (OTF). There is no storage related to the basis evaluation, and both the monomials and the orthonormalization coefficients are re-computed on-the-fly.

The test cases employed to compare the performance of the three proposed implementation was the transport of a vortex in uniform flow, both with Euler and Navier–Stokes equations. In the latter case, the Reynolds number was $Re = 100$. In this section we focus only on the efficiency of the solver rather than considering the temporal accuracy. Three different meshes, respectively made of 25^2, 50^2, and 100^2 quadrilateral elements with linear edges, were considered.

Fig. 1 shows the performance comparison, in terms of CPU time and memory, regarding the computation of the inviscid test case for the three strategies as a function of the total number of degrees of freedom. The computational time is given in terms of Work Units (WU) defined as $\mathrm{WU} = t_s n_c / \tau_b$, where t_s is the measured CPU time of a simulation on n_c cores and τ_b is the reference TauBench time of the hardware[1]. According to this study, the PreCoef strategy seems to be an a appealing compromise between CPU time and the memory footprint, and, therefore, it will be used for the development of our p-adaptation strategy. In particular we pre-compute the orthonormalization coefficients corresponding to the maximum polynomial degree allowed by the user defined adaptation parameters, here fixed at $k = 6$, and use them to orthonormalize monomials according to the local (elemental) polynomial degree of the solution on-the-fly. Same trends have been obtained for the viscous vortex.

4 A p-ADAPTATION STRATEGY

The adaptation is driven by error estimators that control the solution accuracy within the domain, identifying the regions lacking/exceeding the requested resolution. In this work these regions will be refined/coarsened by increasing/decreasing the degree of the polynomial approximation of the solution. Among the several alternatives proposed in literature for DG schemes, we rely on the combination of two indicators. The first one is

[1] -n 250000 -s 10 define the reference TauBench workload for the hardware benchmark

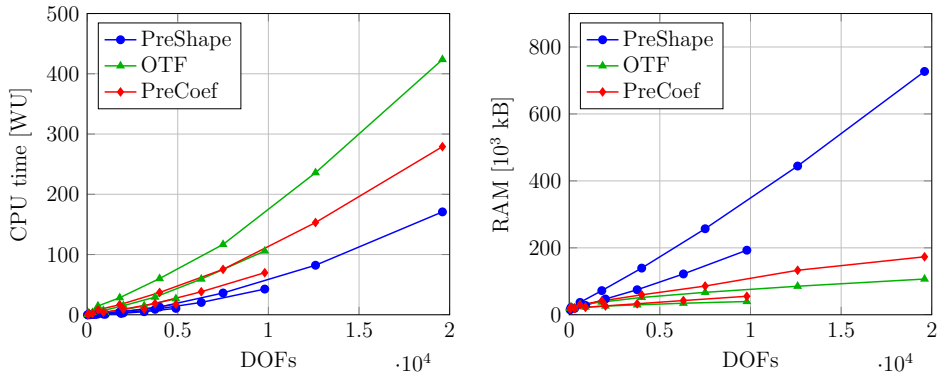


Figure 1: Overall performance of the three different implementations as a function of the degree of freedom for the isoentropic vortex

based on the magnitude of the solution jumps at grid cell interfaces [6], while the second involves the decay rate of the modal coefficients [7]. It is worth pointing out that the latter indicator is local to each element and does not require information of the solution on the neighbouring elements.

The jump indicator is calculated for a solution component w (the pressure for this work), and its value η is defined as the maximum jump at the element T interfaces:

$$\eta_T^{JMP}(w) = \max_{if} \max_{j} \left| \frac{\left(w(x_j,t) - w(x_j,t)^+\right)_{is}}{\left(w(x_j,t) + w(x_j,t)^+\right)_{is}} \right|, \tag{5}$$

where $w(x_j,t)$ is the solution at the j-th surface quadrature points x_j at time t on the if-th interface. An alternative indicator is the Spectral Decay Indicator (SDI), which is here based on the hierarchical modal set of basis functions Φ_T^k. Being the approximation of the solution given by

$$w^T(\mathbf{x}) = \sum_{j=1}^{N_{dof}^T} W_j \phi_j(\mathbf{x}), \tag{6}$$

where the number of degrees of freedom N_{dof}^T depends on the local (to the element) polynomial degree k_T. The truncated expansion for a lower polynomial degree can be written as

$$\bar{w}^T(\mathbf{x}) = \sum_{j=1}^{L_{dof}^T} W_j \phi_j(\mathbf{x}), \tag{7}$$

where L_{dof}^T is the number of degrees of freedom related to the polynomial degree $k_T - 1$. Finally the estimator can be written as

$$\eta_T^{SDI} = \frac{\int_T (w - \bar{w})^2 d\mathbf{x}}{\int_T w^2 d\mathbf{x}}, \tag{8}$$

which relates the amplitudes of highest modes to the amplitude of the total modes.

Numerical experiments on the vortex test case show that the former indicator is reliable for any polynomial degree, whereas the latter gives reasonable indications only for $k \geq 2$, assuming unit valure over the whole computation domain for lower degrees. Moreover η^{JMP} is more diffusive, identifying a larger region for adaptation. According to these observations, we implemented a combination of the two indicators

$$\eta_T^{TOT} = \eta_T^{SDI} \mathrm{INT}\left(\frac{k_T}{\max(2,k_T)}\right) + \frac{1}{\max\left(1,k_T\right)}\eta_T^{JMP} \quad \forall T \in \mathcal{T}_h, \tag{9}$$

where INT represent an integer division (η_T^{SDI} is set to 0 for $k_T = \{0,1\}$). Before the coupling both indicators are normalized over the domain according to their maximum and minimum values.

Being interested on unsteady flow problems, the error estimators used to drive the adaptation of the order of accuracy are computed on the time-averaged solution. The pseudo code of the adaptation procedure is reported in Algorithm 2, where $\hat{k}$ is the initial polynomial degree (fixed or variable over the computational domain), N_{cyc} the total number of time steps required by the simulation, k_{max} the maximum allowable polynomial degree defined by the user ($k_{max} = 6$), $\mathcal{N}$ the number of time-steps elapsing between two adaptation cycles or between the simulation beginning and the first adaptation cycle, $\mathcal{G}_r$ the percentage of the total number of elements that will be marked for refinement, $\mathcal{G}_c$ the percentage of the total number of elements that will be marked for coarsening, n_{adp} the number of adaptation cycles to be performed, POS_T the position of the element sorted in increasing order according to η_T^{TOT}.

Algorithm 2 Adaptation algorithm

$\mathcal{L} = 0$
$k_T = \hat{k} \forall T \in \mathcal{T}_h$
for $i_{cyc} = 1$ **to** N_{cyc} **do**
  integrate the governing equation in time
  evaluate runtime the time-average of the solution, $\bar{\mathbf{W}}$
  if $mod(i_{cyc}, \mathcal{N}) = 0$ **and** $\mathcal{L} \neq n_{adp}$ **then**
    $\mathcal{L} \leftarrow \mathcal{L} + 1$
    compute and normalize η_T^{JMP} and η_T^{SDI} $\forall T \in \mathcal{T}_h$
    compute and normalize η_T^{TOT} $\forall T \in \mathcal{T}_h$
    for $T \in \mathcal{T}_h$ **do**
      if $POS_T \geq (1 - \mathcal{G}_r) card(\mathcal{T}_h)$ **then**
        $k_T \leftarrow \min(k_T + 1, k_{max})$
      else if $POS_T \leq (\mathcal{G}_c) card(\mathcal{T}_h)$ **then**
        $k_T \leftarrow \max(k_T - 1, 1)$
      end if
    end for
    L_2 projection of the solution
  end if
end for

Orthonormal and herarchical modal bases simplify the L^2 projection operators. In practice, the DOFs of the restricted solution are equal to the low-order subset of their high-order representations, while the DOFs of the prolongated solution are the same as the low-order solution with null high-order components.

5 REDUCED QUADRATURES

The number of quadrature points rapidly increases when considering high order polynomials and curved elements. However, as demonstrated in [4], not always the exact quadrature is necessary to maintain the theoretical order of accuracy. To identify only the regions of the mesh that require full quadrature points, we introduce the following measure for the integration error on the element T:

$$\epsilon_{i,T} = |m^*_{ii} - m^{ex}_{ii}| \quad \forall T \in \mathcal{T}_h, \tag{10}$$

where m^*_{ii} denotes the value of the i-th diagonal entry of the local mass matrix computed with the reduced quadrature rule, whereas m^{ex}_{ii} is the expected value as a result of exact integration. Then we use on the elements of $\mathcal{T}_h$ an integration rule with the minimum degree of exactness needed to satisfy the condition:

$$\max_{i \in \left\{1,\dots,N^T_{dof}\right\}} \epsilon_{i,T} \leq tol, \quad \forall T \in \mathcal{T}_h, \tag{11}$$

where tol is a user-defined tolerance in the diagonal entries of the mass matrix.

6 LOAD BALANCING

When the p-adaptation algorithm is applied in parallel, an imbalance of the number of DOF per partition is a side effect. To avoid a drastic reduction of the parallel efficiency, a load-balancing procedure is considered. The approach is here based on repartitioning via the Metis [8] library, exploiting the ability to generate weighted graphs. The weights on the vertex (elements) have been set to take into account the variable polynomial order over the mesh. Figure 2 shows the load of the processors without (left) and with (right) the use of the load balancing approach, after six p-refinements, during the simulation of the vortex test case (see Sec. 3).

7 NUMERICAL RESULTS

The robustness and efficiency of the proposed approach is evaluated by computing the ILES of two problems of growing complexity. The first one involves the two-dimensional flow past a circular cylinder at Reynolds number Re = 3900 (based on the diameter D), while the second one deals with the turbulent flow around a NACA0018 airfoil at Reynolds number Re = 10000 (based on the chord c) and angle of attack $\alpha = 15°$. For both testcases, a two-dimensional mesh was generated with an in-house 2D high-order mesh generation code [9]. For the NACA testcase, the 2D mesh was extruded in the span-wise direction (8 elements along the span-wise-direction), assuming the flow periodic with a period $0.2c$.

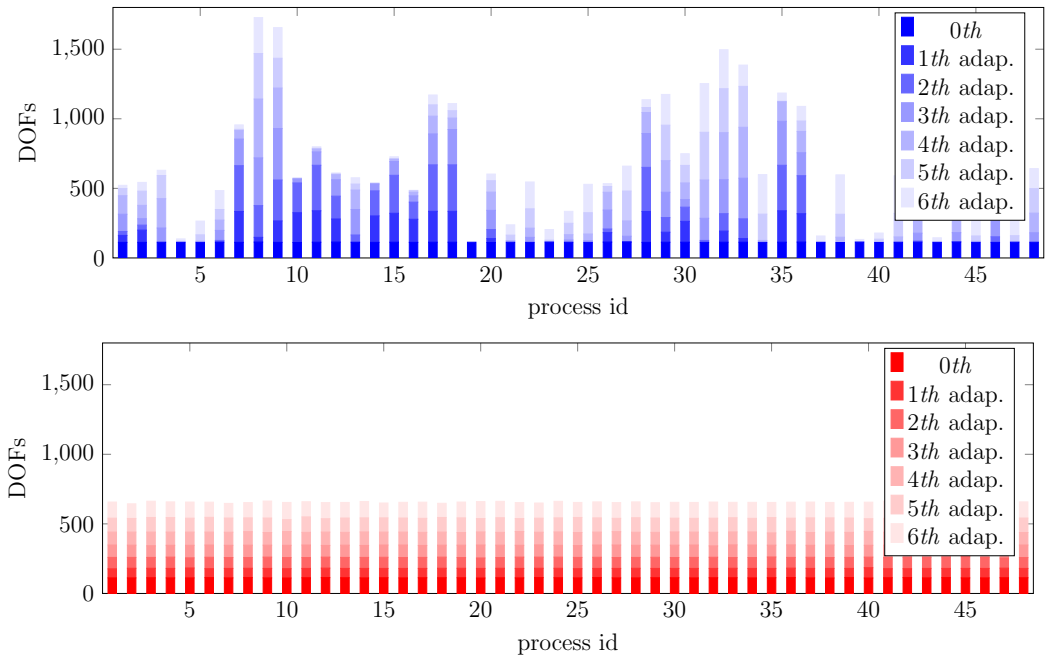


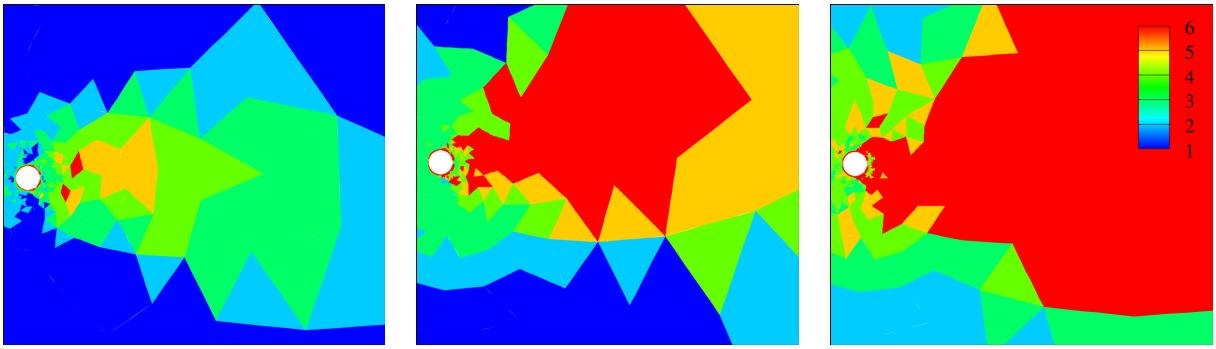
Figure 2: Load of the processors without (top) and with (bottom) the use of the load balancing approach after six p-refinements, invicid isoentropic vortex

7.1 FLOW PAST A CIRCULAR CYLINDER

The ILES of the flow past a 2D circular cylinder is computed for a Reynolds number Re = 3900 based on the cylinder diameter D and free-stream conditions. The hybrid mesh consists of 4343 elements (quadrilateral and triangles) with quadratic edges, and the farfield is set at $100D$. Starting from a $\mathbb{P}^1$ uniform flow solution, different simulations have been performed in order to investigate the influence of the p-adaptation parameters on the numerical results, *i.e.* the number of adaptive cycles $n_{adp} = (5, 10, 20)$, the number of time-steps between two adaptation cycles $\mathcal{N} = (200, 100, 50)$, and the percentage of the total number of elements that will be marked for refinement $\mathcal{G}_r = (10\%, 20\%, 30\%)$. The solution was adapted only for the first 1000 iterations to obtain the polynomial distribution used to computed the unsteady solution. The results on the adapted mesh are then compared with those obtained using a fixed polynomial order ($\mathbb{P}^6$). Figures 3 and 4 show the effect of the adaptation algorithm parameters on the domain polynomial distribution. Figure 5 compares the C_p and C_f distributions obtained with different pair $(n_{adp}, \mathcal{N})$ (keeping fixed $\mathcal{G}_r = 30\%$) and with fixed high-order polynomial approximation($\mathbb{P}^6$). Distribution are in good agreement only for the following pair: $(n_{adp} = 10, \mathcal{N} = 100)$ and $(n_{adp} = 20, \mathcal{N} = 50)$.

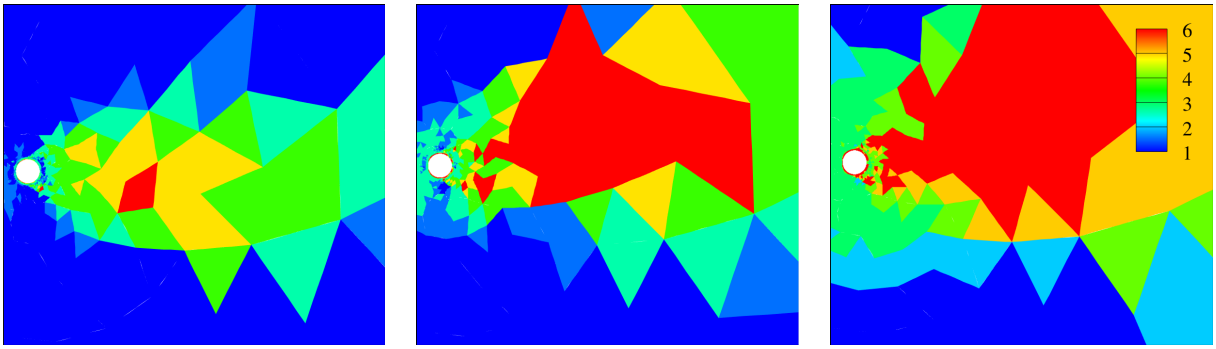
7.2 FLOW AROUND A NACA0018

The ILES of the flow around a NACA0018 is computed for a Reynolds number Re = 10000 based on the airfoil chord c and free-stream conditions, with an angle of attack $\alpha = 15°$. The hybrid 3D mesh consists of 17056 elements (prism and hexahedra) with



(a) $n_{adp} = 5$ and $\mathcal{N} = 200$ (b) $n_{adp} = 10$ and $\mathcal{N} = 100$ (c) $n_{adp} = 20$ and $\mathcal{N} = 50$

Figure 3: Polynomial approximation $\mathbb{P}^n$ distribution around the cylinder, $n_{adp} = 5$ and $\mathcal{N} = 200$ (left), $n_{adp} = 10$ and $\mathcal{N} = 100$ (middle), $n_{adp} = 20$ and $\mathcal{N} = 50$ (right), $\mathcal{G}_r = 30\%$



(a) $\mathcal{G}_r = 10\%$ (b) $\mathcal{G}_r = 20\%$ (c) $\mathcal{G}_r = 30\%$

Figure 4: Polynomial approximation $\mathbb{P}^n$ distribution around the cylinder, $n_{adp} = 10$ and $\mathcal{N} = 100$

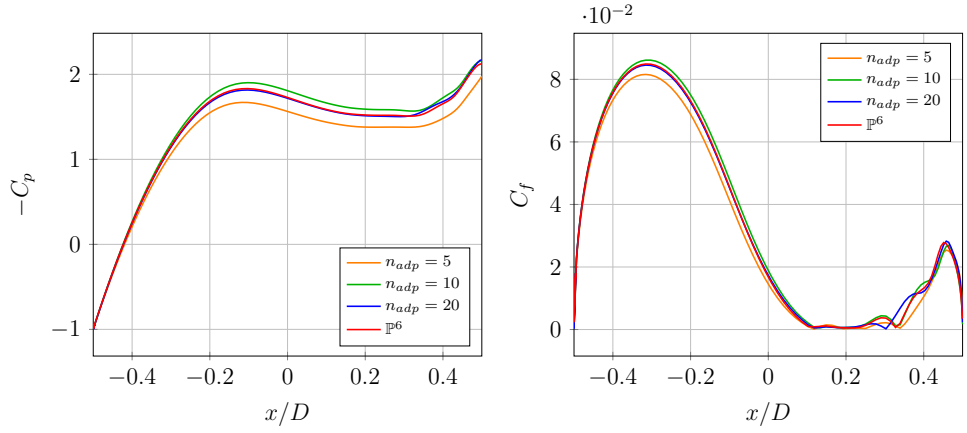


Figure 5: Pressure (left) and skin friction (right) coefficient with different n_{adp} and N ($n_{adp} = 5$ and $\mathcal{N} = 200$, $n_{adp} = 10$ and $\mathcal{N} = 100$, $n_{adp} = 20$ and $\mathcal{N} = 50$), $\mathcal{G}_r = 30\%$

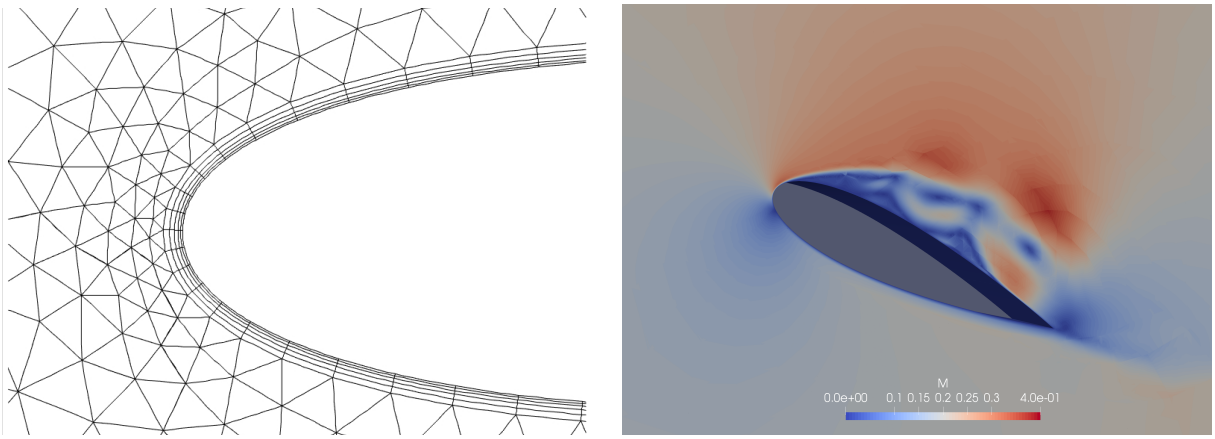


Figure 6: Detail of the computational mesh, 17056 elements (prism and hexahedra) with quadratic edges and Mach magnitude contour for the p-adapted ($\mathbb{P}^{1\to4}$) 3D solution

quadratic edges, and farfield located at $100c$. A detail of the computational mesh is depicted in Fig. 6 (left). Preliminary results show in Fig. 6 (right) the Mach number contours after three adaptation cycles. The adapted 3D solution contains elements ranging from $\mathbb{P}^1$ to $\mathbb{P}^4$. 2D computations have been also performed to investigate the influence of the adaptations parameters. Figure 7 shows the polynomial distribution (left) and the Mach number contours (right) after ten adaptation cycles. The adapted 2D solution contains elements ranging from $\mathbb{P}^1$ to $\mathbb{P}^6$.

8 CONCLUSIONS

In this work we implemented strategy to adapt the order of polynomial approximation on the domain to increase the computational efficiency of a high-order DG solver, based on a p-adaptation algorithm, quadrature reduction technique, and load balancing suitable for unsteady compressible flows simulation in a high-order DG code. We investigated three different approaches for the evaluation of the orthonormal and hierarchical set of basis functions, showing that an appealing compromise between CPU time and the memory footprint is achieved storing only the orthonormalization coefficients. The p-adaptation is driven by a simple yet reliable algorithm to efficiently perform massively parallel computations of DNS, LES of turbulent flows. In the first test case, *i.e.* the ILES simulation of the flow past a 2D circular cylinder, the comparison of C_p and C_f distributions obtained with the proposed strategy or fixed solution polynomial approximation ($\mathbb{P}^6$) shows almost overlying curves. Obviously the results of the former simulation rely on a lower number of DOFs.

Future work will address the possible implementation of more advanced adaptation strategies, error estimators, and the matrix-free approach.

9 ACKNOWLEDGMENT

The results reported in this paper have been achieved using the Italian Super-Computing Resource Allocation (ISCRA) resources (MARCONI-KNL based at CINECA), within the

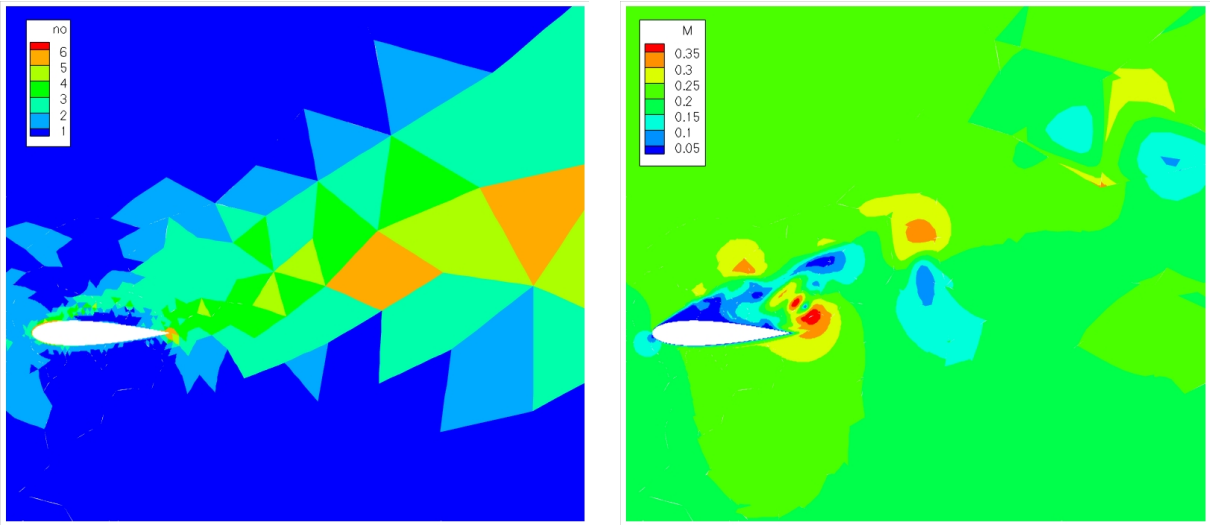


Figure 7: Polynomial distribution (left) and the Mach number contours (right) after ten adaption cycles (2D)

project "High performance implicit matrix-free time integration schemes for Discontinuous Galerkin solutions of incompressible turbulent flows" (HP10BMA1AP).

REFERENCES

[1] Bassi, F. and Botti, L. and Colombo, A. and Crivellini, A. and Ghidoni, A. and Nigro, A. and Rebay, S., *Time integration in the discontinuous galerkin code MIGALE - Unsteady problems*, Notes on Numerical Fluid Mechanics and Multidisciplinary Design 128, 2015.

[2] Bassi, F. and Rebay, S. and Mariotti, G. and Pedinotti, S. and Savini, M. *AA high-order accurate discontinuous finite element method for inviscid and viscous turbomachinery flows*. Proceedings of the 2nd European Conference on Turbomachinery Fluid Dynamics and Thermodynamics, 1997.

[3] Bassi, F. and Botti, L. and Colombo, A. and Ghidoni, A. and Massa, F., *Linearly implicit Rosenbrock-type Runge-Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows*, Computers & Fluids 118, 2015.

[4] Bassi, F. and Botti, L. and Colombo, A. and Di Pietro, D.A. and Tesini, P., *On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations*. J. Comput. Phys. 231, 2012.

[5] Nigro, A. and De Bartolo, C. and Bassi, F. and Ghidoni, A., *Up to sixth-order accurate A-stable implicit schemes applied to the Discontinuous Galerkin discretized Navier-Stokes equations*. J. Comput. Phys. 276, 2014.

[6] Krivodonova, L. and Xin, J. and Remacle, J.-F. and Chevaugeon, N. and Flaherty, J.E., *Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws*. Applied Numerical Mathematics 48, 2004.

[7] Persson, P.-O. and Peraire, J., *Sub-cell shock capturing for discontinuous Galerkin methods*. Collection of Technical Papers - 44th AIAA Aerospace Sciences Meeting 2, 2006.

[8] Karypis, G. and Kumar, V., *MeTis: Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 5.0*. `http://www.cs.umn.edu/~metis`, 2009.

[9] Ghidoni, A. and Pelizzari, E. and Rebay, S. and Selmin, V., *3D anisotropic unstructured grid generation*. International Journal for Numerical Methods in Fluids 51, 2006.