
MOX-Report No. 42/2018

# Algebraic multigrid schemes for high-order discontinuous Galerkin methods

Antonietti, P.F.; Melas, L.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

# ALGEBRAIC MULTIGRID SCHEMES FOR HIGH-ORDER DISCONTINUOUS GALERKIN METHODS

Paola F. Antonietti* and Laura Melas*

* MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
Piazza L. da Vinci 32, 20133 Milano, Italy
paola.antonietti@polimi.it
laura.melas@polimi.it

**Keywords**: algebraic multigrid high-order discontinuous Galerkin.

**Abstract**

We present algebraic multigrid (AMG) methods for the efficient solution of the linear system of equations stemming from high-order discontinuous Galerkin discretizations of second-order elliptic problems. For discontinuous Galerkin methods standard multigrid approaches cannot be employed because of redundancy of the degrees of freedom associated to the same grid point. We present new aggregation procedures and test them on extensive two-dimensional numerical experiments that demonstrate that the proposed AMG method is uniformly convergent with respect to all the discretization parameters, namely the mesh-size and the polynomial approximation degree.

## 1 Introduction

High-order discontinuous Galerkin (DG) methods are widely employed for the numerical solution of partial differential equations because of their flexibility in dealing with non-conforming grids and elementwise varying approximation orders, see, e.g. [21, 27, 41] for an overview on DG methods.

*Paola F. Antonietti and Laura Melas have been partially supported by SIR (Scientific Independence of young Researchers) starting grant n. RBSI14VT0S PolyPDEs: Non-conforming polyhedral finite element methods for the approximation of partial differential equations funded by the Italian Ministry of Education, Universities and Research (MIUR).

In this work we focus on multigrid methods and present a new algebraic multigrid iterative scheme for the efficient solution of the linear system of equations stemming from high-order discontinuous Galerkin finite element approximations of second order elliptic differential equations. Since the pioneer work of Gopalakrishnan and Kanschat [24], multigrid methods for discontinuous Galerkin finite element discretizations of partial differential equations have been intensively studied.

The first developments of geometric multigrid methods for low-order, i.e. linear, discontinuous Galerkin methods can be found in [8, 9, 13–16, 22, 24, 47]. Multigrid techniques coupling geometric and $p$-multigrid approaches have also been studied, cf. [25, 26, 32]. Recently, new $hp$-multigrid schemes for high-order discontinuous Galerkin discretizations have been proposed and analyzed, cf. [3–5]. Algebraic multigrid techniques for matrices stemming from low-order discontinuous Galerkin finite element discretizations of elliptic equations can be found in [10, 40, 43]. The first scalable algebraic multigrid method for high-order discontinuous Galerkin discretizations of the Poisson operator is developed by Olson and Schroder [35]. It assumes the access to mesh points in order to perform the first step of coarsening, therefore employing a geometric information. To the best of our knowledge, purely algebraic multigrid methods for high-order discontinuous Galerkin discretizations have not been addressed so far. Indeed the work by Prill et. al. [40] requires the knowledge of the grid in order to build all the aggregates, Olson and Schroder [35] assume the access to the mesh information for the first coarsening step and the method of Bastian et al. [10] requires that the natural embedding operator is provided. More precisely, the AMG method proposed by Olson and Schroder [35] is a quasi-purely algebraic multigrid because it employs the geometric assumptions only for the first aggregation step.

In this paper we present a new algebraic multigrid method for the efficient solution of the linear systems of equations stemming from high-order discontinuous Galerkin approximations of second-order elliptic problems. We modify the first step of coarsening of the AMG method of Olson and Schroder [35] within an algebraic framework proposing a block-aggregation scheme applied to the finest level. For the coarse levels we employ the classical aggregation of Vaněk et al. [48] following the guideline given in [35]. With these steps our algorithm is fully algebraic because it employs only the entries of the matrix. We demonstrate that for the proposed AMG iterative scheme convergence is achieved independently of both the discretization parameters, namely the mesh-size and the polynomial approximation degree, making the method well suited for both low- and high-order DG approximations.

The remaining part of the paper is organized as follows. In Section 2 we introduce the model problem and its discontinuous Galerkin discretization. In Section 3 we propose our algebraic multigrid method based on smoothed

aggregation and extend it to high-order discontinuous discretizations. In Section 4 we present extensive numerical experiments to investigate the efficiency and robustness of our method. In Section 5 we give a summary of the achieved results and we draw some conclusions.

## 2 Model Problem and its DG Discretization

In this section we present the model problem and its discontinuous Galerkin discretization. Throughout the paper we use the standard notation for Sobolev spaces, cf. [1]. Let $\Omega \subset \mathbb{R}^2$ be a bounded polygonal domain and let $\mathbf{n}$ be the unit outward normal vector to the boundary $\partial\Omega$. For a given function $f \in L^2(\Omega)$ and a given $g \in H^{\frac{1}{2}}(\partial\Omega)$, we consider the weak formulation of the Poisson problem subject to essential boundary conditions: find $u \in V = \{v \in H^1(\Omega)\colon\ u = g \text{ on } \partial\Omega\}$ such that

$$\int_\Omega \nabla u \cdot \nabla v \, d\Omega = \int_\Omega f v \, d\Omega \quad \forall v \in H_0^1(\Omega). \tag{1}$$

Now we describe the numerical solution of (1) based on employing the discontinuous finite element method. We begin by constructing a mesh $\mathcal{T}_h$ of the domain $\Omega \subset \mathbb{R}^2$ made of non-overlapping shape-regular triangles of diameter $h_T$, and set $h = \max_T h_T$. We denote by $e$ the edges of elements of $\mathcal{T}_h$. Let $\mathcal{E}_I$ be the set of interior edges of the mesh $\mathcal{T}_h$, $\mathcal{E}_B$ the set of boundary edges and $\mathcal{E}{=}\mathcal{E}_I \cup \mathcal{E}_B$ the set of all edges. Let $e \in \mathcal{E}_I$ shared by two neighbouring elements $T^\pm$, for (regular enough) scalar and vector-valued functions $v$ and $\boldsymbol{\tau}$, respectively, we define the jumps and averages as

$$\{\!\{v\}\!\} = \frac{1}{2}(v^+ + v^-) \qquad [\![v]\!] = v^+\mathbf{n}^+ + v^-\mathbf{n}^-$$

$$\{\!\{\boldsymbol{\tau}\}\!\} = \frac{1}{2}(\boldsymbol{\tau}^+ + \boldsymbol{\tau}^-) \qquad [\![\boldsymbol{\tau}]\!] = \boldsymbol{\tau}^+ \cdot \mathbf{n}^+ + \boldsymbol{\tau}^- \cdot \mathbf{n}^-$$

where $\mathbf{n}^\pm$ is the unit normal vector to $e$ pointing outward to $T^\pm$, and $v^\pm$ and $\boldsymbol{\tau}^\pm$ are the traces of the functions $v$ and $\boldsymbol{\tau}$ on $T^\pm$, cf. [7]. If $e \in \mathcal{E}_B$ belongs to the boundary $\partial\Omega$ we extend these definition as follows $\{\!\{v\}\!\} = v$, $[\![v]\!] = v\mathbf{n}$, $\{\!\{\boldsymbol{\tau}\}\!\} = \boldsymbol{\tau}$ and $[\![\boldsymbol{\tau}]\!] = \boldsymbol{\tau} \cdot \mathbf{n}$, cf. [7]. Let $V_{hp}$ be a family of finite dimensional spaces defined as

$$V_{hp} = \{v \in L^2(\Omega) : v|_T \in \mathbb{P}_p(T) \quad \forall T \in \mathcal{T}_h\},$$

where $\mathbb{P}_p(T)$ is the space of polynomials of degree lower than or equal to $p \geq 1$. The space $V_{hp}$ is equipped with the norm

$$\|v\|_{DG}^2 = \sum_{T\in\mathcal{T}_h} \|\nabla v\|_{L^2(T)}^2 + \sum_{e\in\mathcal{E}} \|\sqrt{\gamma_e}[\![v]\!]\|_{L^2(e)}^2,$$

where, for a given penalty parameter $\sigma^e > 0$, $\gamma_e$ is defined edgewise as $\gamma_e = \sigma^e p^2/|e|$, $|e|$ being the length of the edge $e$.
Next we define the bilinear form $\mathcal{A} : V_{hp} \times V_{hp} \to \mathbb{R}$ as

$$\mathcal{A}(u,v) = \sum_{T\in\mathcal{T}_h} \int_T \nabla u \cdot \nabla v \, d\Omega - \sum_{e\in\mathcal{E}} \int_e \{\!\{\nabla u\}\!\} \cdot [\![v]\!] \, d\gamma$$
$$- \sum_{e\in\mathcal{E}} \int_e \{\!\{\nabla v\}\!\} \cdot [\![u]\!] \, d\gamma + \sum_{e\in\mathcal{E}} \gamma_e \int_e [\![u]\!] \cdot [\![v]\!] \, d\gamma,$$

and the functional $F : V_{hp} \to \mathbb{R}$ as

$$F(v) = \int_\Omega f v \, d\Omega - \sum_{e\in\mathcal{E}_B} \int_e \nabla v \cdot \mathbf{n}_e \, g \, d\gamma + \sum_{e\in\mathcal{E}_B} \gamma_e \int_e v g \, d\gamma,$$

The discontinuous Galerkin discrete problem reads: find $u_h \in V_{hp}$ such that

$$\mathcal{A}(u_h, v_h) = F(v_h) \quad \forall v_h \in V_{hp}, \tag{2}$$

which is known as symmetric interior penalty (SIP) method [6, 50]. The following result ensures the well-posedness of (2), cf. [6, 7, 50], see, e.g., [23, 28, 39, 44] for $hp-$version error estimates.

**Proposition 2.1.** *If $\sigma^e > \sigma_{min}$, the solution of (2) exists and is unique.*

Let $\{\phi_j\}_{j=1}^{N_h}$ be a basis for the finite element space $V_{hp}$, i.e. $V_{hp} = span\{\phi_j\}_{j=1}^{N_h}$, then (2) is equivalent to the following linear system of equations

$$A\mathbf{u} = \mathbf{f}, \tag{3}$$

where $\mathbf{u} = [u_1, \ldots, u_{N_h}]^T \in \mathbb{R}^{N_h}$ is the vector containing the unknown coefficients of the expansion of the discrete solution $u_h$ in the chosen basis. The stiffness matrix $A$ in (3) is symmetric and positive definite, provided that $\sigma^e$ is large enough.
Next we describe the choice of the shape functions employed to span the discontinuous finite element space. We assume that $v_h \in V_{hp}$ is characterized by the values it takes at the points $P_i = (x_i, y_i)$, with $i = 1, \ldots, N_h$, and consequently the shape functions associated to the finite element space $V_{hp}$ are defined as the Lagrangian functions associated to the nodes with support on a single element.
To define the interpolation points, we consider the reference element $\hat{T} = \{(x, y) : x, y \geq 0, x + y \leq 1\}$ and define therein the Fekete points [18, 45]. Then for any $T \in \mathcal{T}_h$ those points are mapped, through a linear, invertible map $F_T : \hat{T} \to T$, $T \in \mathcal{T}_h$. In Figure 1 we show the Fekete points on the reference triangle for $p = 1, 2, 4, 7$.
*Remark 1. (Condition number of A).* It is of our interest considering the condition number $K_2(A)$ of the system matrix $A$ which results from the

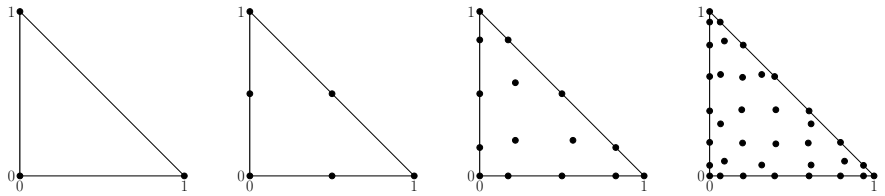



Figure 1: Fekete points on the reference triangle for different choices of $p$ (a) $p = 1$, (b) $p = 2$, (c) $p = 4$, (d) $p = 7$.

DG approximation of problem (2). We point out that, with this choice of interpolation points, the condition number $K_2(A)$ of $A$, defined as the ratio of the extreme eigenvalues of $A$, seems to behave as $K_2(A) = \mathcal{O}(p^3/h^2)$, as shown in Figure 2.
These results seem to indicate that, at least on triangular meshes in two-dimensions, a set of Lagrangian basis functions lead to an improvement of the condition number as a function of $p$. Indeed, in [2] it is proved that, whenever a modal basis based on Legendre polynomials is employed, the condition number of the resulting stiffness matrix behaves as $K_2(A) = \mathcal{O}(p^4/h^2)$.
We point out that, in the conforming setting, i.e. continuous spectral element methods, Pasquetti and Rapetti [37, 38] observed that the condition number is of order $p^4/h^2$, whenever Fekete points are employed, whereas Toselli and Widlund [46] and Bernardi and Maday [11] proved a behaviour of order $p^3/h^2$ if the interpolation points are obtained based on mapping, through the Dubiner map, the classical Gauss-Legendre points defined on the reference square onto the reference triangle.
The issue of proving sharp bounds on the condition number of $A$, whenever DG methods are employed and the discrete space is spanned based on employing Lagrangian functions associated to Fekete points, is under investigation and will be the subject of further research.

# 3 Smoothed-Block Aggregation AMG

In this section, we introduce the main ingredients for the AMG algorithm. We assume to have a sequence of successively coarser matrices $A_k \in \mathbb{R}^{N_k \times N_k}$, $k = 1, \dots K$, with the convention that $A_1 = A$, $N_k > N_{k+1}$, and

$$A_{k+1} = I_k^{k+1} A_k I_{k+1}^k \quad k = 1, \dots, K-1.$$

Here, $I_k^{k+1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k+1}}$ is a linear operator to be properly defined and $I_{k+1}^k = (I_k^{k+1})^T$, see [20, 31, 52], for example.

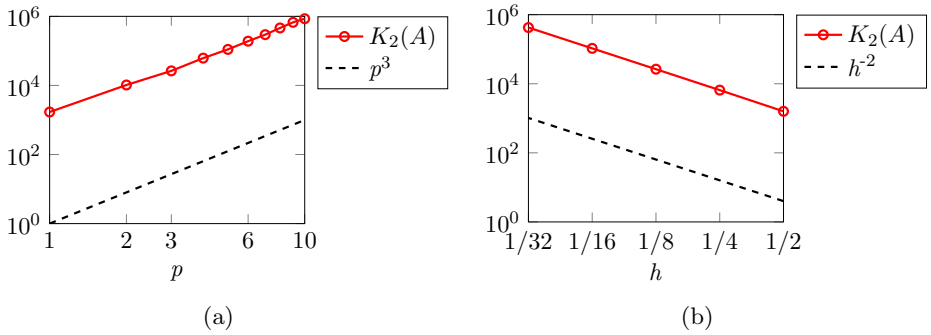



Figure 2: Condition number $K_2(A)$ as function of $p$ (left) and $h$ (right), unstructured triangular meshes, $\sigma^e = 10\ \forall e \in \mathcal{E}$.

By considering a suitable smoother (e.g. dumped Jacobi, symmetrized Gauss-Seidel,...), one iteration of the algebraic $\mu$-cycle scheme, $\mu = 1, 2$, is shown in Algorithm 3.1, cf. also [19, 52]. More precisely given $\mathbf{u}_k^l$, $\mathbf{u}_k^{l+1}$=AMG-$\mu$Cycle$(\ldots, \mathbf{u}_k^l, \ldots)$ returns the $(l+1)-$th iteration to solve $A_k\mathbf{u}_k = \mathbf{f}_k$. If we have a $\mu$-cycle scheme with $\mu = 1$ we refer to it as a V-cycle, whereas for $\mu = 2$ we call the method W-cycle. In particular we denote with V($\nu_1$,$\nu_2$)-cycle and W($\nu_1$,$\nu_2$)-cycle the two methods above with $\nu_1$ pre-smoothing and $\nu_2$ post-smoothing iterations [20]. Algebraic multigrid can be used as a stand-alone solver or as a preconditioner to accelerate the convergence of Krylov-based iterative schemes, such as conjugate gradient method.

**Algorithm 3.1** One Iteration of AMG-$\mu$Cycle to solve $A_k\mathbf{u}_k = \mathbf{f}_k$

**function** $\mathbf{u}_k^{l+1}$=AMG-$\mu$Cycle($\nu_1$,$\nu_2$,$A_k$,$\mathbf{u}_k^l$,$\mathbf{f}_k$,$I_k^{k+1}$,$I_{k+1}^k$)
  **if** $k = K$ **then**
    $\mathbf{u}_K = A_K^{-1}\mathbf{f}_K$ ▷ Coarsest level
    **return** $\mathbf{u}_K$
  **else**
    Relax $\nu_1$ times on $A_k\mathbf{u}_k = \mathbf{f}_k$ with initial guess $\mathbf{u}_k^0$ ▷ Pre-smoothing
    $\mathbf{f}_{k+1} = I_k^{k+1}\,(\mathbf{f}_k - A_k\mathbf{u}_k^{\nu_1})$ ▷ Restriction of the residual
    $\mathbf{e}_{k+1}^0 = \mathbf{0}_{k+1}$
    **for** $\lambda = 1 : \mu$ **do**
      $\mathbf{e}_{k+1}^\lambda$ = AMG-$\mu$Cycle($\nu_1$,$\nu_2$,$A_{k+1}$,$\mathbf{e}_{k+1}^{\lambda-1}$,$\mathbf{f}_{k+1}$,$I_{k+1}^{k+2}$,$I_{k+2}^{k+1}$)
    **end for**
    $\mathbf{u}_k^{\nu_1+1} = \mathbf{u}_k^{\nu_1} + I_{k+1}^k\mathbf{e}_{k+1}^\mu$ ▷ Interpolation and Correction
    Relax $\nu_2$ times on $A_k\mathbf{u}_k = \mathbf{f}_k$ with initial guess $\mathbf{u}_k^{\nu_1+1}$ ▷ Post-smoothing
    **return** $\mathbf{u}_k^{\nu_1+\nu_2+1}$
  **end if**
**end function**

Our method is based on smoothed aggregation [48], and extends the results

of Olson and Schroder [35] to high-order DG methods. The main challenge in the DG setting is the redundancy of the degrees of freedom associated to the same grid points. A similar issue occurs, for example, for systems of partial differential equations where there exists multiple unknowns at the same grid point. This difficulty can be solved with strategies known as "point" or "block" approaches [42, 48]: these techniques are based on local aggregation of variables associated to the same grid point. Here we extend this idea to deal with the multiple unknowns associated to the same grid point typical of DG methods. It is based on the use of the local aggregation for the first level coarsening, then for all the other levels different aggregation schemes can be used.
In the next sections we detail the main steps at the basis of our AMG solvers.

### 3.1 Algebraic Block-Aggregation Algorithm

In the literature we can find different algorithms of the aggregation techniques for AMG methods applied to problem with DG discretizations that exploit the idea of the local aggregation, but all are based on the knowledge of geometric or topological information, cf. [29, 35, 40].
Here we propose a new purely algebraic block-aggregation coarsening strategy based on block-aggregation. The algorithm that we present is built through to the analysis of the matrix entries associated with each degree of freedom, as described in the following.
Given the matrix $A_k \in \mathbb{R}^{N_k \times N_k}$, its entries $a_{ij}$, $i, j = 1, \ldots, N_k$, and its set of unknowns $\mathcal{V} = \{1, \ldots, N_k\}$, namely the degrees of freedom of the problem, we split the set of points in a disjoint covering such that $\mathcal{V} = \bigcup_{j=1}^{N_{k+1}} \mathcal{V}_j$, $N_{k+1} \leq N_k$, and $\mathcal{V}_l \cap \mathcal{V}_j = \emptyset$ for $l \neq j$. In particular, the algorithm aims at providing suitable disjoint sets such that each one of them contains the multiple variables associated to the same physical grid point, cf. Algorithm 3.2.
Algorithm 3.2 is made of three steps: startup singleton or aggregation, enlargement of the decomposition sets, and cancellation of the empty sets. First, for each $i \in \mathcal{V}$, the function find_strongest_connection($i$) chooses the node $I \in \mathcal{V}$ to which the unknown $i$ has the strongest connection, cf. Section 3.3 below. If the strongest connection between $i$ and $I$ is negative, i.e. $a_{iI} < 0$, then the nodes $i$ and $I$ are grouped together (startup aggregation), otherwise the node $i$ is processed alone (startup singleton). Once the startup phase is concluded, the algorithm proceeds with the enlargement of the decomposition sets, based on joining sets with at least one node in common. Finally, empty sets are deleted from the disjoint covering. Algorithm 3.2 is based on the function find_strongest_connection, that is detailed in the following. For a given strength function $s(i, j)$ such that smaller values indicate a stronger connection, e.g. the evolution strength function defined in Section 3.3 below, we assume that the strongest connected points to $i$ are

**Algorithm 3.2** Algebraic Block-Aggregation Algorithm

$h = 0$
**for all** $i \in \mathcal{V}$ **do**
  $I$=find_strongest_connection($i$)
  **if** $a_{iI} \geq 0$ **then**
    **if** $\forall h : \mathcal{V}_h \cap \{i\} = \emptyset$ **then** ▷ Startup Singleton
      $h = h + 1$, $\mathcal{V}_h = \{i\}$
    **end if**
  **else**
    **if** $\forall h : \mathcal{V}_h \cap \{i, I\} = \emptyset$ **then** ▷ Startup Aggregation
      $h = h + 1$, $\mathcal{V}_h = \{i, I\}$
    **else** ▷ Enlarging the Decomposition Sets
      **if** $\exists \tilde{h} : \mathcal{V}_{\tilde{h}} \cap \{i\} \neq \emptyset$ & $\forall h : \mathcal{V}_h \cap \{I\} = \emptyset$ **then**
        $\mathcal{V}_{\tilde{h}} = \mathcal{V}_{\tilde{h}} \cup \{I\}$
      **else if** $\exists \tilde{h} : \mathcal{V}_{\tilde{h}} \cap \{I\} \neq \emptyset$ & $\forall h : \mathcal{V}_h \cap \{i\} = \emptyset$ **then**
        $\mathcal{V}_{\tilde{h}} = \mathcal{V}_{\tilde{h}} \cup \{i\}$
      **else if** $\exists \tilde{h}_1 : \mathcal{V}_{\tilde{h}_1} \cap \{i\} \neq \emptyset$ & $\exists \tilde{h}_2 : \mathcal{V}_{\tilde{h}_2} \cap \{I\} \neq \emptyset$ & $\tilde{h}_1 \neq \tilde{h}_2$ **then**
        $\mathcal{V}_{\tilde{h}_1} = \mathcal{V}_{\tilde{h}_1} \cup \mathcal{V}_{\tilde{h}_2}$, $\mathcal{V}_{\tilde{h}_2} = \emptyset$
      **end if**
    **end if**
  **end if**
**end for**
$j = 0$ ▷ Deleting the Empty Sets
**for all** $h$ **do**
  **if** $\mathcal{V}_h \neq \emptyset$ **then**
    $j = j + 1$, $\mathcal{V}_j = \mathcal{V}_h$
  **end if**
**end for**

given by

$$\mathcal{S}_i = \{j : s(i, j) \leq \theta\}, \tag{4}$$

where $\theta \geq 1$ is a given threshold. Next we fix $\theta$ and then the function find_strongest_connection($i$) returns an arbitrary point in the set $\mathcal{S}_i$.

## 3.2 Interpolation Operator $I_k^{k+1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k+1}}$

Given the disjoint partition $\mathcal{V} = \bigcup_{j=1}^{N_{k+1}} \mathcal{V}_j$, $N_{k+1} \leq N_k$ given by Algorithm 3.2, it is natural to construct the interpolation operator in a similar manner as done for the smoothed aggregation algebraic multigrid by Vaněk et al. [48].

In particular, we modify the algorithm of Vaněk et al. [48] in an energy-minimization framework as follows. We define algebraically smooth error modes to be grid functions with a small Rayleigh quotient, cf. [33] and therefore equivalent to the near null-space or low energy modes. Hence, a tentative interpolation operator is constructed in such a way that it preserves the near null-space mode vector $\mathbf{w}_k \in \mathbb{R}^{N_k}$, cf. [12, 17, 30, 36, 49, 51]. More

precisely, the vector $\mathbf{w}_k$ is the numerical solution of $A_k\mathbf{w}_k = \mathbf{0}_k$ obtained after $\eta$ smoothing steps with initial guess $\mathbf{w}_k^0 = \mathbf{1}_k$.
We first set

$$[\tilde{I}_{k+1}^k]_{ij} = \begin{cases} w_i & i \in \mathcal{V}_j \\ 0 & otherwise \end{cases}, \quad i = 1, \dots, N_k,\ j = 1, \dots, N_{k+1}$$

and apply the Gram-Schmidt orthonormalization algorithm to each column of $\tilde{I}_{k+1}^k$ to improve conditioning. Then, the interpolation operator is defined by a classical dumped-Jacobi smoothing step, i.e.,

$$I_{k+1}^k = (I_k - \omega D_k^{-1} A_k)\tilde{I}_{k+1}^k,$$

where $\omega = 2/3$, $D_k$ is the diagonal of $A_k$ and $I_k$ is the identity matrix.
*Remark 2.* Other approaches can be employed to construct the interpolation matrix. For example, since our problem is symmetric and positive definite, we can employ the Krylov-based framework, cf. [35,36], where we substitute the simple damped Jacobi smoothing step with a fixed number of iterations of the conjugate gradient method.

### 3.3 Evolution Measure

In this section we recall the evolution measure proposed by Olson et. al. [34] which combines the local knowledge of both algebraic smooth error and the behaviour of the interpolation. In the DG framework this measure is necessary to define the strongest connections in Algorithm 3.2 and in the classical aggregation scheme, cf. [35, 48].
In order to take account for algebraic smooth error, we define $\mathbf{z}_k \in \mathbb{R}^{N_k}$ as

$$\mathbf{z}_k = (I_k - \omega_k D_k^{-1} A_k)^m \mathbf{e}_k(i),$$

where $\mathbf{e}_k(i) \in \mathbb{R}^{N_k}$ is the unit vector centered at $i \in \mathcal{V}$, $\omega_k = 1/\rho(D_k^{-1}A_k)$ and $m$ is an integer that has to be properly chosen. In our computation $m = 4$, cf. [35].
Then we have to consider the local knowledge of the interpolation. Assume that the interpolation operator is defined as in Section 3.2. Given a point $i \in \mathcal{V}$ we would like to be able to measure the ability of each column of the tentative interpolation operator $\tilde{I}_{k+1}^k$ to interpolate $\mathbf{z}_k$ for all points $j$ in the algebraic neighborhood of $i$, i.e. $j \in N_i$, where $N_i = \{j : a_{ij} \neq 0\}$. Therefore this quantity is measured only for points $j \in N_i$, in particular with exact interpolation enforced at point $i$.
We define the evolution measure as

$$e(i,j) = \left| 1 - \frac{w_j z_i}{w_i z_j} \right| \quad i, j = 1, \dots, N_k,$$

where $w_j$ and $z_j$ are the $j-$th components of vectors $\mathbf{w}_k$, defined in Section 3.2, and $\mathbf{z}_k$, respectively. Since our problem is symmetric, we define the symmetrized version of the evolution measure as

$$e_S(i,j) = e(i,j) + e(j,i).$$

Finally the symmetric evolution strength function is defined as

$$s(i,j) = \frac{e_S(i,j)}{\min_{k \neq i} e_S(i,k)}. \tag{5}$$

The symmetric evolution measure defined above is employed to identify the connections in our algorithm, cf. Section 4 below. Our algorithm makes use of the following two steps that are the block- and classical aggregations, respectively. On the finest level, we employ Algorithm 3.2 with the choice of evolution strength function (5) and $\theta = 1$ in (4). On the coarsest levels we use the aggregation scheme of Vaněk et al. [48] with still evolution strength function (5) but with $\theta \in [2,4]$, cf. (4).
This choice is guided by the following properties that hold in the DG framework: we employ our block-aggregation for the finest level because it is suited to aggregate the multiple degrees of freedom associated to each grid point, on the other way we use the classical aggregation for the coarsest levels because it builds larger agglomerates and this is better to have less unknowns associated to these levels.
In Figure 3 we show some examples of block-aggregation for matrices stemming from linear DG discretizations on structured and unstructured simplicial meshes and with penalty parameter $\sigma^e = 10, 20, 30$, cf. Section 2. For $\sigma^e = 30$, we obtain the same aggregations as for $\sigma^e = 20$. For simplicity, these results have been omitted. Moreover, when we compute the evolution measure, we fix $\mathbf{w}_k = \mathbf{1}_k$. Each aggregate set is represented with a distinct number as mark.
We notice that, as expected, our block-aggregation algorithm seems to be fairly insensitive on the value of the penalty parameter.

# 4 Numerical Experiments

In this section we test the robustness and the efficiency of our algebraic multigrid method in solving the linear system of equations stemming from high-order discontinuous finite element discretizations of problem (1). We consider a sequence of structured and unstructured simplicial meshes with granularity $h = 1/2, 1/4, 1/8, 1/16, 1/32$, and let the polynomial approximation degree $p$ vary from 1 to 10. For each $h$ and $p$, we obtain a linear system of equations that we solve with our smoothed block-aggregation AMG, cf. Section 3. At the first step of coarsening, i.e. $k = 1$, we use the block-aggregation algorithm, cf. Algorithm 3.2 with the strongest evolution

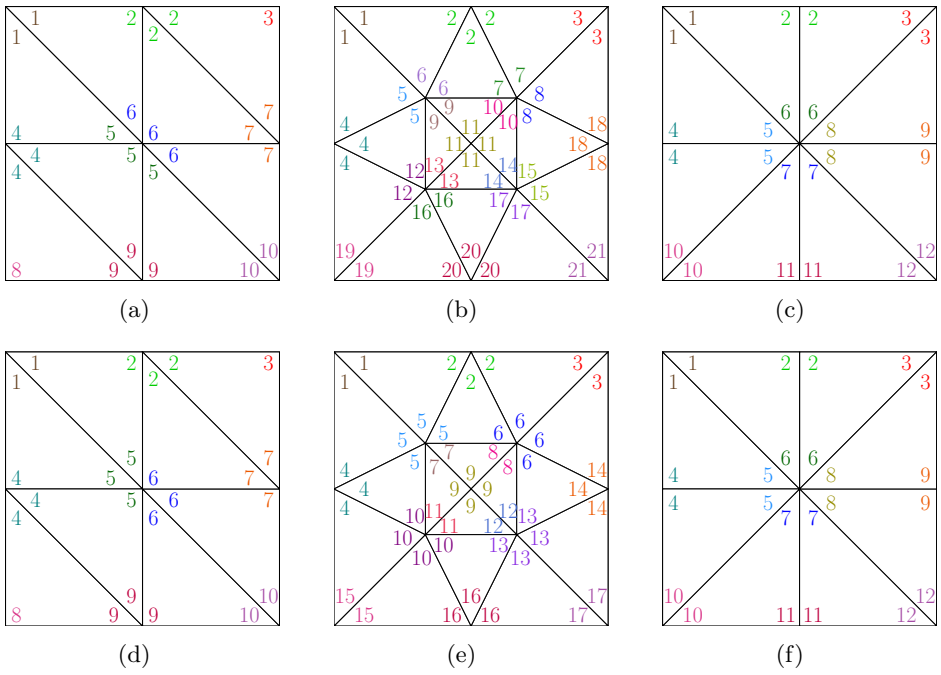



Figure 3: Examples of block-aggregation for different meshes with $p = 1$, $h \cong 1/2$ and $\sigma^e = 10$ (top) and $\sigma^e = 20$ (bottom).

connection defined in (5) and with $\theta = 1$, cf. (4). For the coarser levels, i.e. $k = 2, \ldots, K$, we use the classical aggregation of Vaněk et al. [48] with still the evolution strength function in (5) and $\theta = 2$ in (4), cf. also [35].
For any multigrid level $k$, the associated interpolation operator is the one proposed in Section 3.2 with $\eta = p$ smoothing iterations of classical Gauss-Seidel, where $p$ is the polynomial degree. For the smoothing interpolation we compare both the Jacobi iteration and the Krilov-based framework with 2 iterations of conjugate gradient method, cf. Remark 2. We remark that for moderate values of $p$, one iteration of CG is enough. In our numerical results we denote by J-smoother/CG-smoother the smoothed block-aggregation algebraic multigrid with Jacobi/CG smoothing interpolation step, cf. Remark 2.
In our numerical tests we test the W($\nu_1$,$\nu_2$)-cycle with the classical Gauss-Seidel relaxation as a stand-alone AMG solver. Moreover, we also consider a PCG method with a preconditioner given by the the W($\nu_1$,$\nu_2$) iteration with a symmetric Gauss-Seidel smoother. We refer to the preconditioned conjugate gradient with W($\nu_1$,$\nu_2$)-cycle preconditioner as PCG W($\nu_1$,$\nu_2$)-cycle.
Let $N$ be the iteration counts needed to reduce the initial relative residual below a tolerance $tol = 10^{-8}$, we compute the convergence factor $\rho$ defined

by

$$\rho = \exp\left(\frac{1}{N}\log\frac{||\mathbf{r}_N||}{||\mathbf{r}_0||}\right),$$

where $\mathbf{r}_N$ and $\mathbf{r}_0$ are the final and initial residuals, respectively.
In Section 4.1 and 4.2 we report the results when employing the W-cycle algorithm as iterative scheme and as preconditioner for the conjugate gradient method, respectively. All the proposed solver components are summarized in Figure 4.

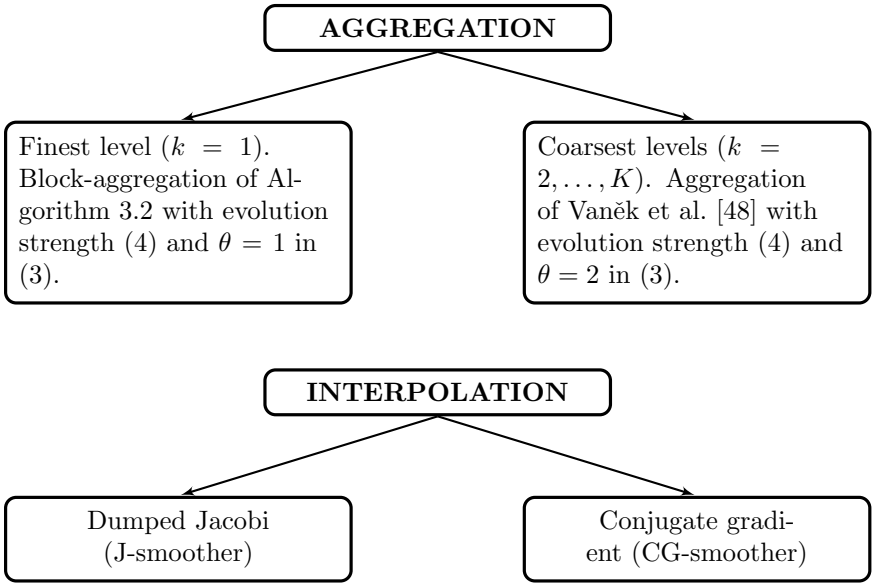



Figure 4: Aggregation and interpolation steps in our AMG algorithm.

### 4.1 W-cycle Algorithm as Iterative Scheme

In this section we present some numerical results to investigate the performance of the W-cycle AMG algorithm as iterative scheme. In Figures 5 and 6 we compare the W-cycle AMG with J- and CG-smoother in terms of $p$- and $h$-scalability, respectively, when employing $\nu_1 = \nu_2 = \nu = 1, 3$ pre- and post-smoothing iterations. We note that if we employ the AMG method with the CG- rather than the J-smoother we obtain better results both in terms of convergence factor and scalability.
We remark that in our tests the W-cycle AMG with J-smoother do not converge for $p = 10$, therefore in Figure 6 we show only the W-cycle AMG with CG-smoother when $p = 10$.
Concerning the $h$- and $p$-scalability we observe that the J-smoother AMG method seems to be scalable only if the number of smoothing steps is sufficiently large, even if we can observe $hp$-weak-scalability for $p = 1, \ldots, 7$

and all tested $h$ for smaller values of smoothing iterations $\nu$. In order to have $h$- and $p$-scalability for the J-smoother AMG method, for all the considered $h$ and $p$, we should increase the number of pre- and post-smoothing iterations, cf. [13, 14, 16]. On the other hand we have that the CG-smoother AMG method is $hp$-quasi-scalable for $\nu = 1, 2$ and $hp$-scalable for $\nu = 3$ for all considered $h$ and $p$. The difference in $h$- and $p$-scalability when varying $\nu = 1, 2, 3$ for AMG method with CG-smoother is small, so it is worth considering the method with $\nu = 1$ because it has a lower computational costs.
In Table 1 we report the computed convergence factors for the J- and CG-smoother AMG methods on both structured and unstructured grids when varying the number of smoothing iterations $\nu_1 = \nu_2 = \nu = 1, 2, 3$. From the results reported in Table 1 we can conclude that, as expected, the AMG algorithms performs better for larger number of smoothing iterations.
In Table 2 we report the results obtained for the J- and CG-smoother AMG methods when varying the number of coarsening levels $K = 2, \ldots, 5$ and solving the problem discretized on structured and unstructured grids, respectively. From the results reported in Table 2, it seems that all the proposed methods converge uniformly with respect to the number of levels $K$.
As already observed we can summarize the following considerations:

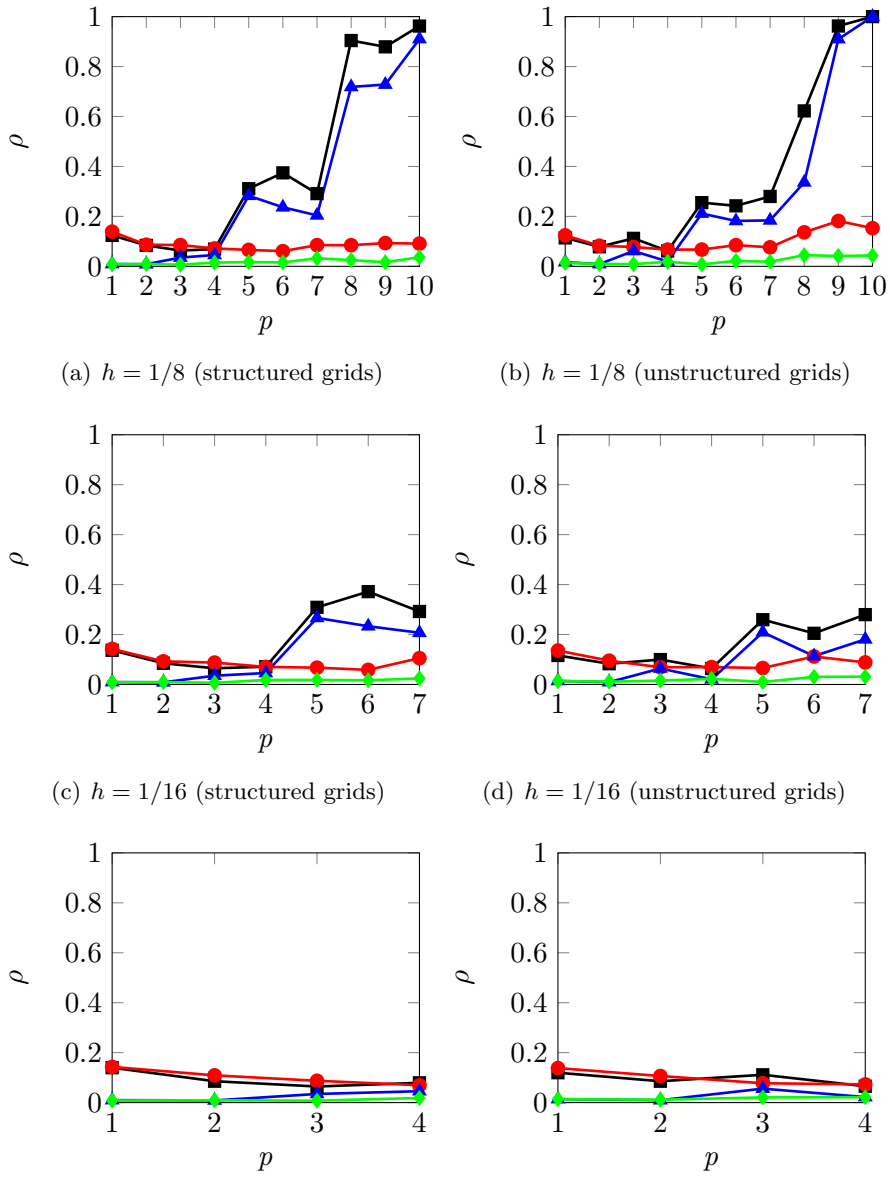
- the J-smoother AMG method seems to be scalable w.r.t. both the discretization parameters $h$ and $p$, and the number of multigrid levels provided that the number $\nu$ of smoothing steps is chosen large enough ($\nu \gg 3$). It seems to be $hp$-weak-scalable for $p = 1, \ldots, 7$ and all tested $h$ for smaller values of $\nu$;

- the CG-smoother AMG method seems to be scalable w.r.t. both the discretization parameters $h$ and $p$, and the number of levels provided that the number of smoothing steps is large enough (in our computations $\nu = 3$);

- the J-smoother AMG method, even if it seems to be only $hp$-weak-scalable, features lower computational costs compared to the CG-smoother AMG one.

### 4.2 W-cycle Algorithm as Preconditioner for PCG Method

In this section we repeat the numerical tests presented in Section 4.1, and we present some numerical results to test the efficiency of the W-cycle algorithm as preconditioner for the PCG method.
In Figures 7 and 8 we report the computed convergence factors based on employing the J- and CG-smoother AMG as preconditioners for PCG method in terms of $p$- and $h$-scalability, respectively, when employing $\nu_1 = \nu_2 = \nu = 1, 3$ pre- and post-smoothing iterations. We remark that in our tests the




(a) $h = 1/8$ (structured grids)

(b) $h = 1/8$ (unstructured grids)

(c) $h = 1/16$ (structured grids)

(d) $h = 1/16$ (unstructured grids)

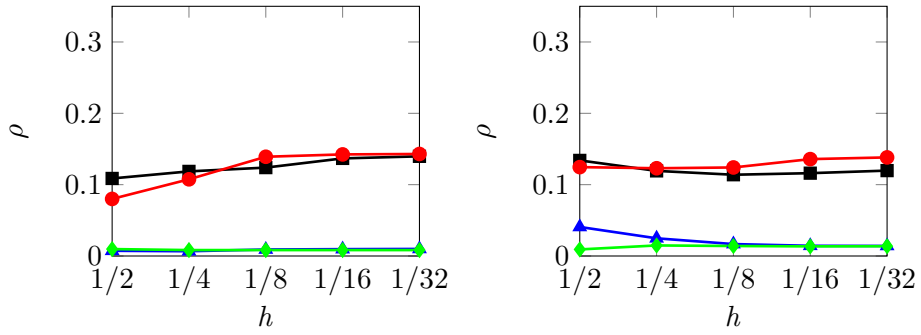
(e) $h = 1/32$ (structured grids)

(f) $h = 1/32$ (unstructured grids)

Figure 5: Convergence factor of the W-cycle algorithm as a function of $p$ for different values of $h = 1/8, 1/16, 1/32$ on structured (left) and unstructured (right) meshes: J-smoother, $\nu = 1$ (—■—); J-smoother, $\nu = 3$ (—▲—); CG-smoother, $\nu = 1$ (—●—); CG-smoother, $\nu = 3$ (—◆—).

J-smoother PCG method do not converge for $p = 10$, therefore in Figure 8 we show only the CG-smoother PCG method when $p = 10$.
In Table 3 we report the convergence factor for the J- and CG-smoother

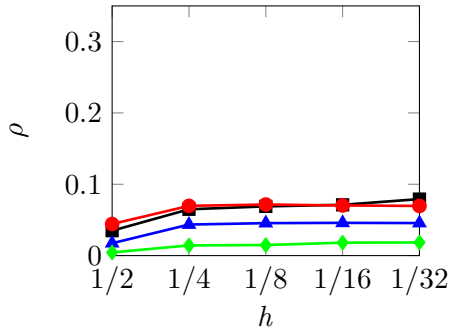
PCG methods both on structured and unstructured grids, respectively, when varying the number of smoothing iterations $\nu_1 = \nu_2 = \nu = 1, 2, 3$.
In Table 4 we report the results obtained for the J- and CG-smoother PCG methods when varying the number of coarsening levels $K = 2, \ldots, 5$ and solving the problem discretized on structured and unstructured grids, respectively.
For the sake of comparison, in Tables 3 and 4 we also report the values of the converge factor when we employ the CG method.
Considering the results presented, we can obtain the same conclusions as in



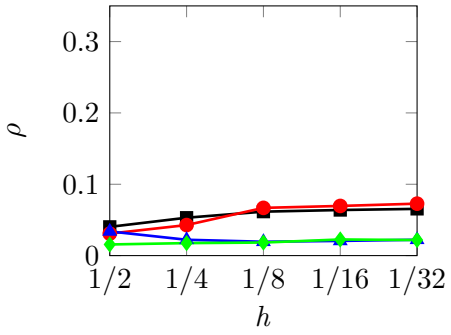

(a) $p = 1$ (structured grids)



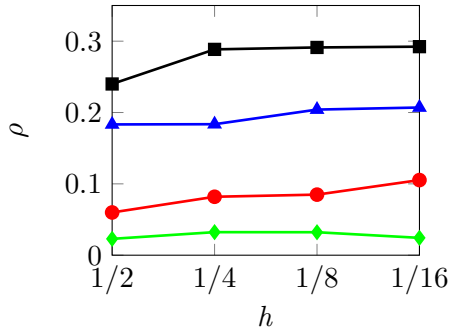

(b) $p = 1$ (unstructured grids)



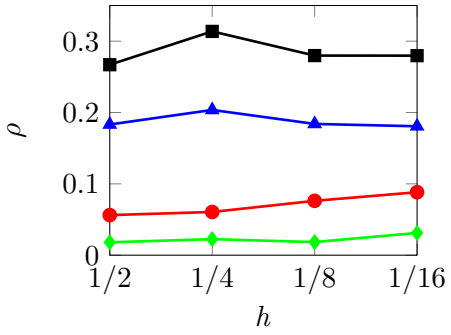

(c) $p = 4$ (structured grids)



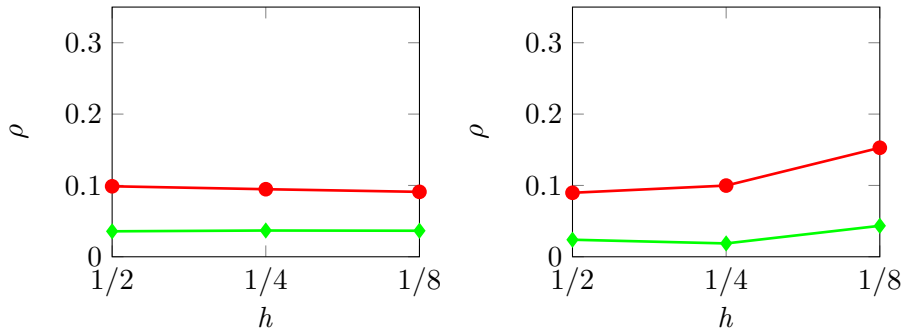

(d) $p = 4$ (unstructured grids)




(e) $p = 7$ (structured grids)




(f) $p = 7$ (unstructured grids)




(g) $p = 10$ (structured grids) (h) $p = 10$ (unstructured grids)

Figure 6: Convergence factor of the W-cycle algorithm as a function of $h$ for different values of $p = 1, 4, 7, 10$ on structured (left) and unstructured (right) meshes: J-smoother, $\nu = 1$ (—■—); J-smoother, $\nu = 3$ (—▲—); CG-smoother, $\nu = 1$ (—●—); CG-smoother, $\nu = 3$ (—◆—).

Table 1: Convergence factor of the W-cycle algorithm as a function of $\nu$, $K = 4$.

| Structured Grids | | | | |
|---|---|---|---|---|
| | $\nu$ | $p = 1$ $h = 1/32$ | $p = 4$ $h = 1/8$ | $p = 7$ $h = 1/2$ |
| J-smoother | 1 | 0.1397 | 0.0690 | 0.2400 |
| | 2 | 0.0276 | 0.0573 | 0.2013 |
| | 3 | 0.0099 | 0.0455 | 0.1833 |
| CG-smoother | 1 | 0.1430 | 0.0716 | 0.0599 |
| | 2 | 0.0226 | 0.0162 | 0.0315 |
| | 3 | 0.0083 | 0.0148 | 0.0229 |
| Unstructured Grids | | | | |
| | $\nu$ | $p = 1$ $h = 1/32$ | $p = 4$ $h = 1/8$ | $p = 7$ $h = 1/2$ |
| J-smoother | 1 | 0.1197 | 0.0616 | 0.2670 |
| | 2 | 0.0251 | 0.0212 | 0.2069 |
| | 3 | 0.0141 | 0.0194 | 0.1832 |
| CG-smoother | 1 | 0.1382 | 0.0669 | 0.0562 |
| | 2 | 0.0277 | 0.0200 | 0.0212 |
| | 3 | 0.0137 | 0.0183 | 0.0181 |

Section 4.1. We observe that the J-smoother seems to be scalable w.r.t. all the discretization parameters and the number of multigrid levels provided that the number of smoothing iterations is sufficiently large and the CG-smoother AMG one seems to be scalable for smaller values of $\nu$ (in our

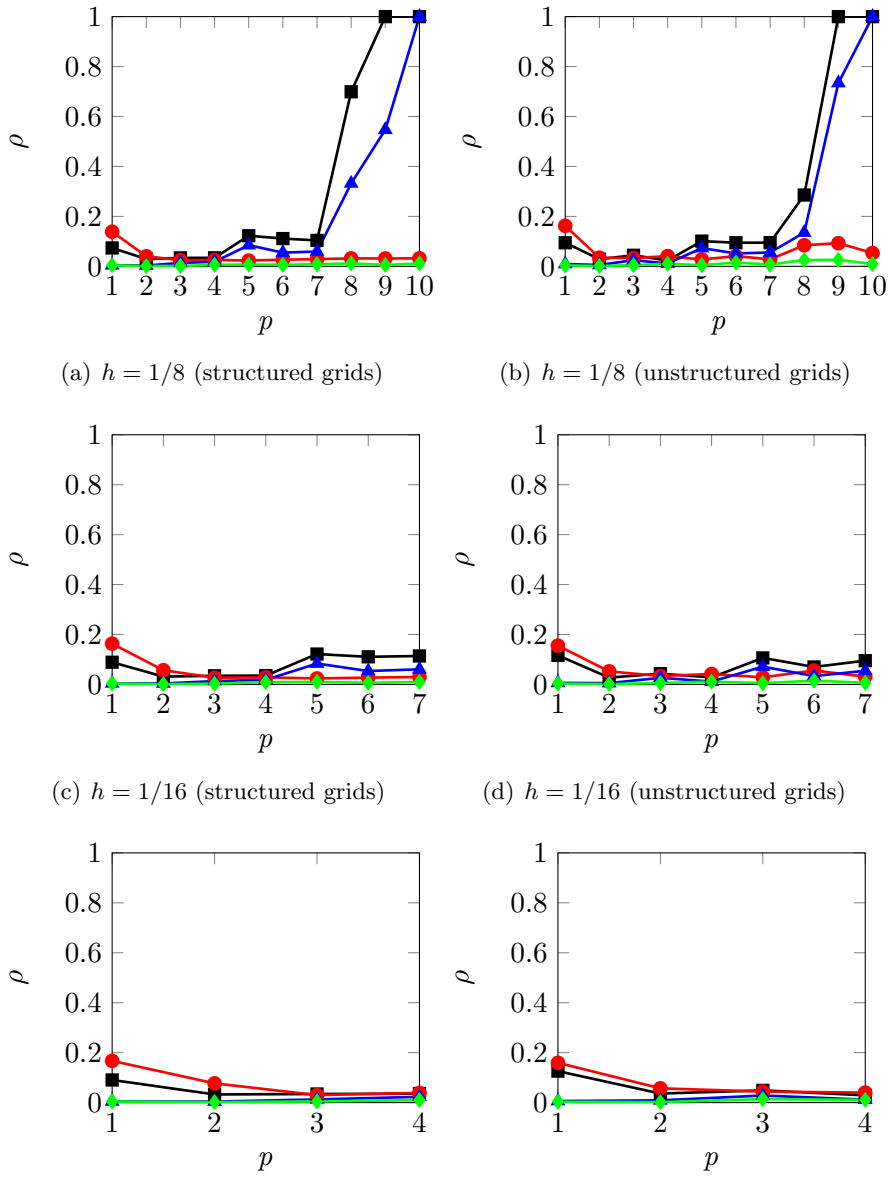
Table 2: Convergence factor of the W(1,1)-cycle algorithm as a function of the number of levels $K$.

| Structured Grids | | | | |
|---|---|---|---|---|
| | $K$ | $p=1$ $h=1/32$ | $p=4$ $h=1/8$ | $p=7$ $h=1/2$ |
| J-smoother | 2 | 0.1397 | 0.0686 | 0.2386 |
| | 3 | 0.1397 | 0.0690 | 0.2399 |
| | 4 | 0.1397 | 0.0690 | 0.2400 |
| | 5 | 0.1397 | 0.0690 | 0.2400 |
| CG-smoother | 2 | 0.1430 | 0.0717 | 0.0601 |
| | 3 | 0.1430 | 0.0716 | 0.0599 |
| | 4 | 0.1430 | 0.0716 | 0.0599 |
| | 5 | 0.1430 | 0.0716 | 0.0599 |
| Unstructured Grids | | | | |
| | $K$ | $p=1$ $h=1/32$ | $p=4$ $h=1/8$ | $p=7$ $h=1/2$ |
| J-smoother | 2 | 0.1196 | 0.0615 | 0.2633 |
| | 3 | 0.1197 | 0.0616 | 0.2670 |
| | 4 | 0.1197 | 0.0616 | 0.2670 |
| | 5 | 0.1197 | 0.0616 | 0.2670 |
| CG-smoother | 2 | 0.1382 | 0.0669 | 0.0571 |
| | 3 | 0.1382 | 0.0669 | 0.0562 |
| | 4 | 0.1382 | 0.0669 | 0.0562 |
| | 5 | 0.1382 | 0.0669 | 0.0562 |

computations $\nu = 3$). In addition we notice that when we employ the two algorithms as preconditioner for the conjugate gradient method we obtain better values of the convergence factor.

# 5 Conclusions

We have presented a new algebraic multigrid method for solving the linear system of equations stemming from high-order discontinuous Galerkin finite element discretizations of second order elliptic problems.
We have extended the standard algebraic multigrid approach, by proposing a new algebraic block-aggregation scheme that suitably handles the redundancy of the degrees of freedom associated to the same grid point. In addiction we have employed a different strength function of connection, cf. [34], and an adaptive smoothed aggregation method, cf. [17], following the guideline of Olson and Schroder [35]. In particular we modified the first step of geometric coarsening within an algebraic framework leading our schemes to be purely AMG methods for high-order DG discretizations. The obtained




(a) $h = 1/8$ (structured grids)

(b) $h = 1/8$ (unstructured grids)

(c) $h = 1/16$ (structured grids)

(d) $h = 1/16$ (unstructured grids)

(e) $h = 1/32$ (structured grids)

(f) $h = 1/32$ (unstructured grids)

Figure 7: Convergence factor of the PCG W-cycle algorithm as a function of $p$ for different values of $h = 1/8, 1/16, 1/32$ on structured (left) and unstructured (right) meshes: J-smoother, $\nu = 1$ (—■—); J-smoother, $\nu = 3$ (—▲—); CG-smoother, $\nu = 1$ (—●—); CG-smoother, $\nu = 3$ (—◆—).

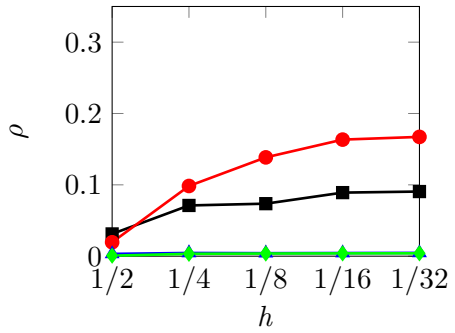
results show that the proposed AMG methods are scalable with respect to the mesh-size $h$, the polynomial degree $p$ and the number of multigrid levels provided that the number $\nu$ of smoothing steps is large enough.

## References

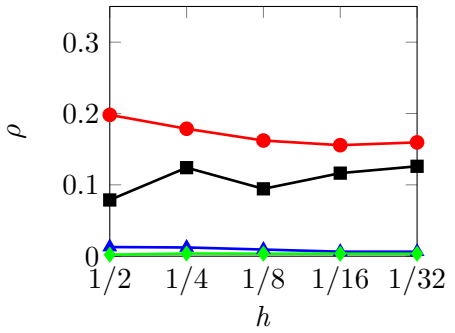

[1] R. A. Adams. *Sobolev spaces*. Academic Press, New York-London, 1975. Pure and Applied Mathematics, Vol. 65.

[2] P. F. Antonietti and P. Houston. A class of domain decomposition preconditioners for *hp*-discontinuous Galerkin finite element methods. *J. Sci. Comput.*, 46(1):124–149, 2011.

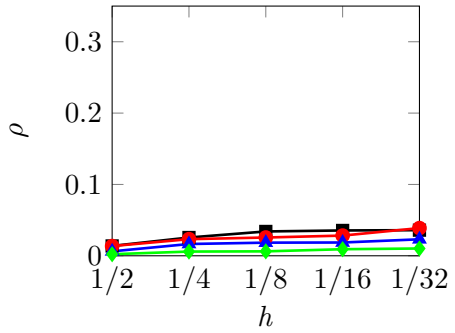
[3] P. F. Antonietti, P. Houston, X. Hu, M. Sarti, and M. Verani. Multi-

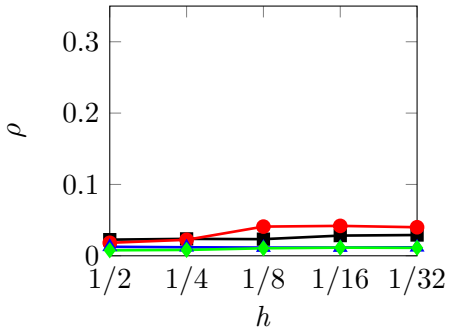

(a) $p = 1$ (structured grids)



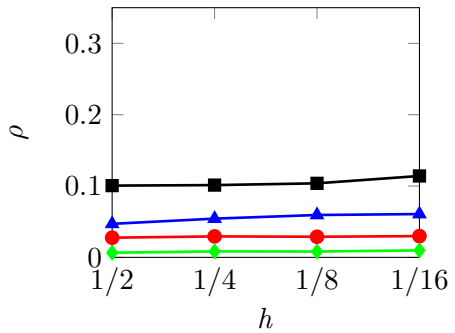

(b) $p = 1$ (unstructured grids)



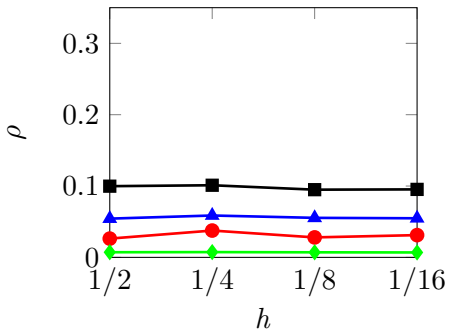

(c) $p = 4$ (structured grids)



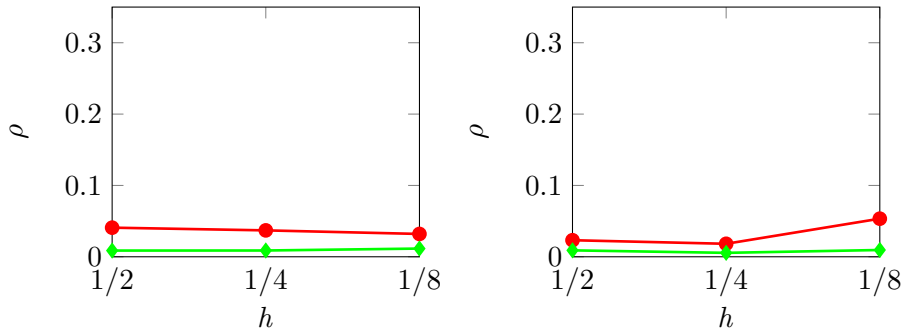

(d) $p = 4$ (unstructured grids)




(e) $p = 7$ (structured grids)




(f) $p = 7$ (unstructured grids)




(g) $p = 10$ (structured grids) (h) $p = 10$ (unstructured grids)

Figure 8: Convergence factor of the PCG W-cycle algorithm as a function of $h$ for different values of $p = 1, 4, 7, 10$ on structured (left) and unstructured (right) meshes: J-smoother, $\nu = 1$ (—■—); J-smoother, $\nu = 3$ (—▲—); CG-smoother, $\nu = 1$ (—●—); CG-smoother, $\nu = 3$ (—◆—).

Table 3: Convergence factor of the PCG W-cycle algorithm as a function of $\nu$, $K = 4$ and comparison with CG.

| Structured Grids | | | | |
|---|---|---|---|---|
| | $\nu$ | $p = 1$ $h = 1/32$ | $p = 4$ $h = 1/8$ | $p = 7$ $h = 1/2$ |
| J-smoother | 1 | 0.0906 | 0.0341 | 0.1006 |
| | 2 | 0.0096 | 0.0205 | 0.0609 |
| | 3 | 0.0046 | 0.0184 | 0.0470 |
| CG-smoother | 1 | 0.1672 | 0.0256 | 0.0276 |
| | 2 | 0.0128 | 0.0076 | 0.0085 |
| | 3 | 0.0040 | 0.0062 | 0.0065 |
| CG | | 0.9483 | 0.9546 | 0.9056 |
| Unstructured Grids | | | | |
| | $\nu$ | $p = 1$ $h = 1/32$ | $p = 4$ $h = 1/8$ | $p = 7$ $h = 1/2$ |
| J-smoother | 1 | 0.1260 | 0.0233 | 0.0998 |
| | 2 | 0.0139 | 0.0136 | 0.0610 |
| | 3 | 0.0063 | 0.0115 | 0.0543 |
| CG-smoother | 1 | 0.1595 | 0.0409 | 0.0263 |
| | 2 | 0.0090 | 0.0128 | 0.0093 |
| | 3 | 0.0034 | 0.0105 | 0.0072 |
| CG | | 0.9641 | 0.9677 | 0.9401 |

grid algorithms for *hp*-version interior penalty discontinuous Galerkin methods on polygonal and polyhedral meshes. *Calcolo*, 54(4):1169–

Table 4: Convergence factor of the PCG W(1,1)-cycle algorithm as a function of the number of levels $K$ and comparison with CG.

| Structured Grids | | | | |
|---|---|---|---|---|
| | $K$ | $p=1$ $h=1/32$ | $p=4$ $h=1/8$ | $p=7$ $h=1/2$ |
| J-smoother | 2 | 0.0905 | 0.0339 | 0.0989 |
| | 3 | 0.0906 | 0.0341 | 0.1006 |
| | 4 | 0.0906 | 0.0341 | 0.1006 |
| | 5 | 0.0906 | 0.0341 | 0.1006 |
| CG-smoother | 2 | 0.1671 | 0.0257 | 0.0274 |
| | 3 | 0.1672 | 0.0256 | 0.0276 |
| | 4 | 0.1672 | 0.0256 | 0.0276 |
| | 5 | 0.1672 | 0.0256 | 0.0276 |
| CG | | 0.9483 | 0.9546 | 0.9056 |
| Unstructured Grids | | | | |
| | $K$ | $p=1$ $h=1/32$ | $p=4$ $h=1/8$ | $p=7$ $h=1/2$ |
| J-smoother | 2 | 0.1261 | 0.0201 | 0.0941 |
| | 3 | 0.1260 | 0.0233 | 0.0998 |
| | 4 | 0.1260 | 0.0233 | 0.0998 |
| | 5 | 0.1260 | 0.0233 | 0.0998 |
| CG-smoother | 2 | 0.1596 | 0.0409 | 0.0264 |
| | 3 | 0.1595 | 0.0409 | 0.0263 |
| | 4 | 0.1595 | 0.0409 | 0.0263 |
| | 5 | 0.1595 | 0.0409 | 0.0263 |
| CG | | 0.9641 | 0.9677 | 0.9401 |

1198, 2017.

[4] P. F. Antonietti and G. Pennesi. V-cycle multigrid algorithms for discontinuous Galerkin methods on non-nested polytopic meshes. *J. Sci. Comput.*, 2018.

[5] P. F. Antonietti, M. Sarti, and M. Verani. Multigrid algorithms for *hp*-discontinuous Galerkin discretizations of elliptic problems. *SIAM J. Numer. Anal.*, 53(1):598–618, 2015.

[6] D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

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

[8] F. Bassi, L. Botti, and A. Colombo. Agglomeration-based physical frame dG discretizations: an attempt to be mesh free. *Math. Models Methods Appl. Sci.*, 24(8):1495–1539, 2014.

[9] F. Bassi, L. Botti, A. Colombo, D. A. Di Pietro, and P. Tesini. On the flexibility of agglomeration based physical space discontinuous Galerkin discretizations. *J. Comput. Phys.*, 231(1):45–65, 2012.

[10] P. Bastian, M. Blatt, and R. Scheichl. Algebraic multigrid for discontinuous Galerkin discretizations of heterogeneous elliptic problems. *Numer. Linear Algebra Appl.*, 19(2):367–388, 2012.

[11] C. Bernardi and Y. Maday. *Approximations spectrales de problèmes aux limites elliptiques*, volume 10 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer-Verlag, Paris, 1992.

[12] J. Brannick, M. Brezina, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. An energy-based AMG coarsening strategy. *Numer. Linear Algebra Appl.*, 13(2-3):133–148, 2006.

[13] S. C. Brenner, J. Cui, T. Gudi, and L.-Y. Sung. Multigrid algorithms for symmetric discontinuous Galerkin methods on graded meshes. *Numer. Math.*, 119(1):21–47, 2011.

[14] S. C. Brenner, J. Cui, and L.-Y. Sung. Multigrid methods for the symmetric interior penalty method on graded meshes. *Numer. Linear Algebra Appl.*, 16(6):481–501, 2009.

[15] S. C. Brenner and L. Owens. A $W$-cycle algorithm for a weakly over-penalized interior penalty method. *Comput. Methods Appl. Mech. Engrg.*, 196(37-40):3823–3832, 2007.

[16] S. C. Brenner and J. Zhao. Convergence of multigrid algorithms for interior penalty methods. *Appl. Numer. Anal. Comput. Math.*, 2(1):3–18, 2005.

[17] M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge. Adaptive smoothed aggregation ($\alpha$SA) multigrid. *SIAM Rev.*, 47(2):317–346, 2005.

[18] M. Briani, A. Sommariva, and M. Vianello. Computing Fekete and Lebesgue points: simplex, square, disk. *J. Comput. Appl. Math.*, 236(9):2477–2486, 2012.

[19] W. Briggs and S. McCormick. Introduction. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 1–30. SIAM, Philadelphia, PA, 1987.

[20] W. L. Briggs, V. E. Henson, and S. F. McCormick. *A multigrid tutorial*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2000.

[21] D. A. Di Pietro and A. Ern. *Mathematical aspects of discontinuous Galerkin methods*, volume 69 of *Mathématiques & Applications (Berlin) [Mathematics & Applications]*. Springer, Heidelberg, 2012.

[22] V. A. Dobrev, R. D. Lazarov, P. S. Vassilevski, and L. T. Zikatanov. Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. *Numer. Linear Algebra Appl.*, 13(9):753–770, 2006.

[23] E. H. Georgoulis and E. Süli. Optimal error estimates for the *hp*-version interior penalty discontinuous Galerkin finite element method. *IMA J. Numer. Anal.*, 25(1):205–220, 2005.

[24] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numer. Math.*, 95(3):527–550, 2003.

[25] B. T. Helenbrook and H. L. Atkins. Application of *p*-multigrid to discontinuous Galerkin formulations of the Poisson equation. *AIAA Journal*, 44, 3 2006.

[26] B. T. Helenbrook and H. L. Atkins. Solving discontinuous Galerkin formulations of Poisson's equation using geometric and *p* multigrid. *AIAA Journal*, 46, 4 2008.

[27] J. S. Hesthaven and T. Warburton. *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York, 2008. Algorithms, analysis, and applications.

[28] P. Houston, C. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39(6):2133–2163, 2002.

[29] J. E. Jones and P. S. Vassilevski. AMGe based on element agglomeration. *SIAM J. Sci. Comput.*, 23(1):109–133, 2001.

[30] J. Mandel, M. Brezina, and P. Vaněk. Energy optimization of algebraic multigrid bases. *Computing*, 62(3):205–228, 1999.

[31] J. Mandel, S. McCormick, and R. Bank. Variational multigrid theory. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 131–177. SIAM, Philadelphia, PA, 1987.

[32] B. S. Mascarenhas, B. T. Helenbrook, and H. L. Atkins. Coupling *p*-multigrid to geometric multigrid for discontinuous Galerkin

formulations of the convection-diffusion equation. *J. Comput. Phys.*, 229(10):3664–3674, 2010.

[33] S. F. McCormick and J. W. Ruge. Multigrid methods for variational problems. *SIAM J. Numer. Anal.*, 19(5):924–929, 1982.

[34] L. N. Olson, J. Schroder, and R. S. Tuminaro. A new perspective on strength measures in algebraic multigrid. *Numer. Linear Algebra Appl.*, 17(4):713–733, 2010.

[35] L. N. Olson and J. B. Schroder. Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems. *J. Comput. Phys.*, 230(18):6959–6976, 2011.

[36] L. N. Olson, J. B. Schroder, and R. S. Tuminaro. A general interpolation strategy for algebraic multigrid using energy minimization. *SIAM J. Sci. Comput.*, 33(2):966–991, 2011.

[37] R. Pasquetti and F. Rapetti. Spectral element methods on triangles and quadrilaterals: comparisons and applications. *J. Comput. Phys.*, 198(1):349–362, 2004.

[38] R. Pasquetti and F. Rapetti. Spectral element methods on unstructured meshes: comparisons and recent advances. *J. Sci. Comput.*, 27(1-3):377–387, 2006.

[39] I. Perugia and D. Schötzau. An *hp*-analysis of the local discontinuous Galerkin method for diffusion problems. *J. Sci. Comput.*, 17(1-4):561–571, 2002.

[40] F. Prill, M. Lukáčová-Medvidová, and R. Hartmann. Smoothed aggregation multigrid for the discontinuous Galerkin method. *SIAM J. Sci. Comput.*, 31(5):3503–3528, 2009.

[41] B. Rivière. *Discontinuous Galerkin methods for solving elliptic and parabolic equations*, volume 35 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. Theory and implementation.

[42] J. W. Ruge and K. Stüben. Algebraic multigrid. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 73–130. SIAM, Philadelphia, PA, 1987.

[43] J. B. Schroder. Smoothed aggregation solvers for anisotropic diffusion. *Numer. Linear Algebra Appl.*, 19(2):296–312, 2012.

[44] B. Stamm and T. P. Wihler. *hp*-optimal discontinuous Galerkin methods for linear elliptic problems. *Math. Comp.*, 79(272):2117–2133, 2010.

[45] M. A. Taylor, B. A. Wingate, and R. E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal.*, 38(5):1707–1720, 2000.

[46] A. Toselli and O. Widlund. *Domain decomposition methods—algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.

[47] M. H. van Raalte and P. W. Hemker. Two-level multigrid analysis for the convection-diffusion equation discretized by a discontinuous Galerkin method. *Numer. Linear Algebra Appl.*, 12(5-6):563–584, 2005.

[48] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196, 1996. International GAMM-Workshop on Multi-level Methods (Meisdorf, 1994).

[49] W. L. Wan, T. F. Chan, and B. Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comput.*, 21(4):1632–1649, 1999.

[50] M. F. Wheeler. An elliptic collocation-finite element method with interior penalties. *SIAM J. Numer. Anal.*, 15(1):152–161, 1978.

[51] J. Xu and L. Zikatanov. On an energy minimizing basis for algebraic multigrid methods. *Comput. Vis. Sci.*, 7(3-4):121–127, 2004.

[52] J. Xu and L. Zikatanov. Algebraic multigrid methods. *Acta Numer.*, 26:591–721, 2017.

# MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

**41/2018** Mazzieri, I.; Melas, L.; Smerzini, C.; Stupazzini, M.
*The role of near-field ground motion on seismic risk assessment in large urban areas*

**39/2018** Ferro, N.; Micheletti, S.; Perotto, S.
*Density-based inverse homogenization with anisotropically adapted elements*

**40/2018** Chiappa, A.S.; Micheletti, S.; Peli, R.; Perotto, S.
*Mesh adaptation-aided image segmentation*

**38/2018** Domanin, M.; Gallo, D.; Vergara, C.; Biondetti, P.; Forzenigo, L.V.; Morbiducci, U.
*Prediction of long term restenosis risk after surgery in the carotid bifurcation by hemodynamic and geometric analysis*

**37/2018** Bonaventura, L.; Della Rocca A.;
*Convergence analysis of a cell centered finite volume diffusion operator on non-orthogonal polyhedral meshes*

**32/2018** Dal Santo, N.; Deparis, S.; Manzoni, A.; Quarteroni, A.
*An algebraic least squares reduced basis method for the solution of nonaffinely parametrized Stokes equations*

**34/2018** Laurino, F.; Coclite, A.; Tiozzo, A.; Decuzzi, P.; Zunino, P.;
*A muliscale computational approach for the interaction of functionalized nanoparticles with the miscrovasculature*

**35/2018** Possenti, L.; Casagrande, G.; Di Gregorio, S.; Zunino, P.; Costantino, M.L.
*Numerical simulations of the microvascular fluid balance with a non-linear model of the lymphatic system*

**36/2018** Agosti, A.; Ambrosi, D.; Turzi, S.
*Strain energy storage and dissipation rate in active cell mechanics*

**31/2018** Quarteroni, A.
*The role of statistics in the era of big data: A computational scientist' perspective*