Politecnico di Milano

SCHOOL OF INDUSTRIAL AND INFORMATION ENGINEERING
Master of Science in Mathematical Engineering

Algebraic multigrid methods for high order continuous and discontinuous Galerkin methods

Candidate:
Laura Melas
ID 836469

Thesis Advisor:
Paola F. Antonietti

Academic Year 2016–2017

Abstract

We present algebraic multigrid solvers for elliptic problems discretized by high order continuous and discontinuous Galerkin finite element methods. Algebraic multigrid is an effective technique for solving the linear system of equations stemming from the discretization of partial differential equations.

For continuous Galerkin finite element methods we consider classical and smoothed aggregation algebraic multigrid methods and show their effectiveness. For high order discontinuous Galerkin methods standard multigrid approaches cannot be employed because of redundancy of the degrees of freedom associated to the same grid point. In this case we present suitable modifications of standard agglomeration procedures and present new approaches, which are tested on extensive numerical experiments.

Sommario

In questa tesi verrá studiata una famiglia di solutori *algebraic multigrid* per problemi ellittici discretizzati attraverso i metodi agli elementi finiti di Galerkin continui e discontinui. Il metodo *algebraic multigrid* è una tecnica iterativa per risolvere i sistemi lineari risultanti dalla discretizzazione di equazioni alle derivate parziali. Per i metodi agli elementi finiti di Galerkin continui verranno studiati i metodi *algebraic multigrid* classico e di *smoothed aggregation* e verrá mostrata la loro efficienza computazionale. Per i metodi di alto ordine agli elementi finiti di Galerkin discontinui non è possibile utilizzare approcci multigrid standard a causa della ridondanza del numero di gradi di libertá associati allo stesso punto di griglia. In questo caso verranno proposte opportune modifiche delle procedure standard di aggregazione e saranno studiati nuovi approcci, testati su numerosi esperimenti numerici.

Contents

Ringraziamenti

Al termine di questa impegnativa ed interessante esperienza di ricerca, vorrei ringraziare le persone che hanno avuto un ruolo importante.

Ringrazio innanzitutto la Professoressa Paola Antonietti che mi ha trasmesso l'interesse e la passione per l'ambito di ricerca approfondito e che mi ha sostenuta durante la preparazione di questa tesi.

Ringrazio i miei genitori Graziella e Salvatore per il loro costante sostegno morale ed economico in tutta la mia carriera universitaria al Politecnico di Milano, soprattutto per avermi sempre motivata a dare il massimo e aver sempre creduto nelle mie potenzialitá.

Grazie a Federico, il mio ragazzo, che con amore, pazienza e fiducia mi ha sostenuta per tutti questi anni sia nelle esperienze di vita sia in quella universitaria.

Grazie a mio fratello Nicola e ai miei parenti che sono stati presenti col loro affetto e sostegno.

Infine grazie ai miei amici e compagni universitari con cui ho condiviso le mie gioie e le mie sventure. Un particolare grazie va a Sara, la mia piú cara amica.

Introduction

The numerical solution of partial differential equations is of fundamental importance in the description of phenomena in Engineering and Applied Science. An obvious demand that arises is the high accuracy of the numerical solution of the mathematical model, and therefore highly efficient arbitrarily accurate numerical techniques are needed.

This thesis focuses on multigrid methods for solving the linear system of equations stemming from high order conforming and discontinuous Galerkin discretizations of second order elliptic differential equations. Multigrid methods can be viewed as an acceleration of iterative schemes featuring a smoothing property. The smoothing property refers to an efficient reduction of oscillatory components of the error than the smooth ones. The main idea behind multigrid methods is to project the smooth error modes onto a coarser level where a relaxation will be more effective on all error components.
There are two main categories of multigrid methods: geometric and algebraic multigrid. The first depends on a hierarchy of geometric grids, the second is based on a purely algebraic approach to generalize the geometric multigrid whenever grid levels are not available.

The first multigrid methods have been proposed for linear conforming Galerkin discretizations. They were introduced with the geometric approach by [Fedorenko, 1961] but the methods have become frequently used only later on with the work of [Brandt, 1977a, Hackbusch, 1985, Bramble, 1993]. The first algebraic multigrid approach was proposed by [Brandt et al., 1982] with the idea of using the algebraic information provided by the stiffness matrix of the problem and later generalized by [Stüben, 1983, Ruge and Stüben, 1985]. The algebraic multigrid generalizations consists in three main techniques: classical, cf. [Ruge and Stüben, 1987], smoothed aggregation, cf. [Vaněk et al., 1996], and energy minimization-based algebraic multigrid, cf. [Mandel et al., 1999, Wan et al., 1999]. All these techniques have then been used as preconditioners for conjugate gradient, cf. [Braess, 1995]. In recent years, the algebraic multigrid approaches have been extended to high order conforming finite element methods, cf. [Heys et al., 2005], but it is still a research field.

Since the pioneer work of [Gopalakrishnan and Kanschat, 2003], multigrid methods have been extended to discontinuous Galerkin finite element discretizations of partial differential equations, especially for elliptic problems. The reason is that discontinuous Galerkin methods have become of great interest because of their flexibility in dealing with non-conforming grids and varying local approximation orders. The first developments of multigrid methods for linear discontinuous Galerkin methods can be found in the geometric framework, cf. [Gopalakrishnan and Kanschat, 2003, van Raalte and Hemker, 2005, Brenner and Zhao, 2005, Dobrev et al., 2006, Brenner and Owens, 2007, Brenner et al., 2009, Brenner et al., 2011]. Then, thanks to the flexibility of these methods, techniques of coupling geometric and p multigrid have been also proposed, cf. [Helenbrook and Atkins, 2006, Helenbrook and Atkins, 2008, Mascarenhas et al., 2010]. Moreover, more recent researches have proposed a new approach of hp-multigrid schemes for high order discontinuous Galerkin discretizations, cf. [Nastase and Mavriplis, 2006, Antonietti et al., 2015, Antonietti et al., 2017].

Geometric multigrid methods have been successfully applied to several problems, but they are often costly since the problem has to be rediscretized for each coarser level: this motivates an algebraic approach. Algebraic multigrid techniques for matrices stemming from low order discontinuous Galerkin finite element discretizations of elliptic equations can be found in [Prill et al., 2009, Bastian et al., 2012, Schroder, 2012], but it is unclear how to extend them to the high order framework. The first scalable algebraic multigrid for high order discontinuous Galerkin discretizations of the Poisson operator is developed by [Olson and Schroder, 2011], but it assumes the access to the mesh points in order to perform the first step of coarsening. This leads purely algebraic multigrid methods for high order discontinuous Galerkin discretizations to be still an open research topic.

The aim of this thesis is to develop multigrid methods for the efficient solution of the linear system of equations stemming from high order conforming and discontinuous Galerkin approximations of second order elliptic problems. The goal is to develop an algebraic solver that is both h- and p-independent, especially in the high order discontinuous Galerkin setting.

We describe in detail the organization of the thesis by summarizing the contents of the various chapters.

In Chapter 1 we introduce the Poisson problem, present its variational formulation and analyze its well-posedness. Then we introduce the Galerkin finite element method, first for conforming discretizations, and then for discontinuous ones.

In Chapter 2 we describe the multigrid principles: error smoothing and coarse grid correction. We then explain how they are combined to form a multigrid method. First we introduce the two-grid correction scheme in a geometric framework, then we extend it to the algebraic setting. Finally we discuss the multigrid cycles, e.g. V- and W-cycles, and show their employment as stand-alone solvers or as preconditioners for the conjugate gradient method.

In Chapter 3 we introduce the discussion on coarsening strategies and interpolation operators to build the coarser matrices in the conforming Galerkin discretizations framework based on classical and smoothed aggregation algebraic multigrid methods. Then, we focus on the application of algebraic multigrid methods to high order conforming discretizations.
In Chapter 4 we provide a numerical comparison of classical and smoothed aggregation algebraic multigrid methods for conforming Galerkin discretizations and we illustrate their performance and robustness.
In Chapter 5 we explain the coarsening strategies and interpolation operators to build the coarser matrices in the discontinuous Galerkin discretizations framework. We propose new algebraic multigrid methods based on the employment of classical and smoothed aggregation and extend the latter to high order discontinuous discretizations.
In Chapter 6 we present extensive numerical experiments to investigate the efficiency and robustness of our hybrid and smoothed block aggregation algebraic multigrid methods for discontinuous Galerkin discretizations.
In Chapter 7 we give a summary of the achieved results and we discuss open problems for future research.

Chapter 1

The Poisson Equation

In this chapter we present the Poisson problem and discretize it by using as a guideline the references [Quarteroni, 2014] for the continuous finite element method and [Rivière, 2008] for the discontinuous Galerkin one. Moreover throughout the chapter we will use the standard notation for Sobolev spaces, cf. [Adams, 1975].

1.1 Model Problem

Let $\Omega \subset \mathbb{R}^2$ be a bounded 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 following Poisson problem.

Problem 1.1.

$$\begin{cases} -\Delta u = f & \textit{in } \Omega \\ u = g & \textit{on } \partial\Omega \end{cases}$$

If we assume additional smoothness on f and g, Problem 1.1 has a strong solution $u \in C^2(\Omega) \cap C(\bar{\Omega})$.

1.1.1 Weak Formulation

To write the variational formulation of Problem 1.1 in the Hilbert space $H^1(\Omega)$ we introduce the following bilinear form $a_C : H^1(\Omega) \times H^1_0(\Omega) \to \mathbb{R}$

$$a_C(u, v) = \int_\Omega \nabla u \cdot \nabla v \, d\Omega,$$

and the following functional $F_C : H^1_0(\Omega) \to \mathbb{R}$

$$F_C(v) = \int_\Omega f v \, d\Omega.$$

The weak formulation reads as follows

Problem 1.2. *Find $u \in H^1(\Omega)$, $u = g$ on $\partial\Omega$, such that*

$$a_C(u, v) = F_C(v) \quad \forall v \in H^1_0(\Omega)$$

The solution $u \in H^1(\Omega)$ is the weak solution of the Problem 1.1. By employing the Lax-Milgram lemma we can prove existence and uniqueness of u.

Proposition 1.1. *The solution of Problem 1.2 exists and is unique.*

Proof. In order to prove existence and uniqueness of the solution we show that Lax-Milgram lemma is satisfied. We have to check that the bilinear form $a_C(\cdot,\cdot)$ is continuous and coercive and that the linear functional $F_C(\cdot)$ is continuous with respect to norm $\|\cdot\|_{H^1(\Omega)}$, i.e.,

$$a_C(u,u) \gtrsim \|u\|^2_{H^1(\Omega)} \ \forall u \in H^1(\Omega), \tag{1.1}$$

$$a_C(u,v) \lesssim \|u\|_{H^1(\Omega)}\|v\|_{H^1(\Omega)} \ \forall u,v \in H^1(\Omega), \tag{1.2}$$

$$F_C(v) \lesssim \|v\|_{H^1(\Omega)} \ \forall v \in H^1(\Omega). \tag{1.3}$$

Conditions (1.2) and (1.3) are a consequence of the Cauchy-Schwarz inequality, while condition (1.1) is a consequence of the Poincarè inequality with the use of the lifting operator.

Therefore, Lax-Milgram lemma is satisfied and there exists a unique $u \in H^1(\Omega)$ solution of Problem 1.2. □

1.2 Numerical Approximation

In this section we describe the numerical solution of the Poisson problem by using the continuous and discontinuous finite element methods.

We begin by constructing a quasi-uniform mesh $\mathcal{T}_h$ of the domain $\Omega \subset \mathbb{R}^2$ made of non-overlapping shape regular triangles. The partition $\mathcal{T}_h$ is such that for any element $T \in \mathcal{T}_h$ we have $\max_{x,y\in T} |x-y| \leq h$.

We present two methods: the continuous Galerkin (CG) finite element method and discontinuous Galerkin (DG) finite element one.

Before presenting these methods it is useful to provide some preliminary results. First of all we introduce the space of polynomials $\mathbb{P}_p$ of degree lower than or equal to $p \geq 1$ defined as

$$\mathbb{P}_p = \{g(\mathbf{x}) = \sum_{\substack{i,j\geq 0\\ i+j\leq p}} a_{ij}\, x^i\, y^j, \quad a_{ij} \in \mathbb{R}\}.$$

The dimension of $\mathbb{P}_p$ is given by

$$N_p = \dim \mathbb{P}_p = \frac{(p+1)(p+2)}{2}.$$

1.2.1 Continuous Galerkin Finite Element Methods

In this section we introduce the continuous Galerkin finite element method. Let $V^C_{hp} \subset H^1(\Omega)$ be a family of finite dimensional spaces parametrized by the mesh parameter h and defined as

$$V^C_{hp} = \{v_h \in C(\bar{\Omega}) : v_h|_T \in \mathbb{P}_p \quad \forall T \in \mathcal{T}_h\}, \quad p = 1,2,\ldots$$

and let $\mathring{V}_{hp}^C$ be a family of its finite dimensional subspaces

$$\mathring{V}_{hp}^C = \{v_h \in V_{hp}^C : v_h|_{\partial\Omega} = 0\}.$$

The approximate problem is called Galerkin problem and is of the form

Problem 1.3. *Find $u_h \in V_{hp}^C$, $u_h = g_h$ on $\partial\Omega \cap \mathcal{T}_h$, such that*

$$a_C(u_h, v_h) = F_C(v_h) \quad \forall v_h \in \mathring{V}_{hp}^C.$$

Here g_h is the approximation of the boundary data.

The spaces V_{hp}^C and $\mathring{V}_{hp}^C$ are well suited to approximate the Hilbert spaces $H^1(\Omega)$ and $H_0^1(\Omega)$, respectively. In fact if $v \in C(\bar{\Omega})$ and $v \in H^1(T)$ $\forall T \in \mathcal{T}_h$ then $v \in H^1(\Omega)$.

Proposition 1.2. *The solution of Problem 1.3 exists and is unique.*

Proof. Since $V_{hp}^C \subset H^1(\Omega)$, $a_C(\cdot,\cdot)$ and $F_C(\cdot)$ are the same of Problem 1.2, Proposition 1.2 follows from Proposition 1.1. □

We are also interested in the error estimates, cf. [Babuška and Suri, 1987, Babuška and Suri, 1994, Schwab, 1998].

Theorem 1.1. *Let $u \in H^1(\Omega)$ be the exact solution of the weak formulation defined in Problem 1.2 and let $u_h \in V_{hp}^C$ be its approximate solution obtained from Problem 1.3. If $u \in C(\bar{\Omega}) \cap H^{s+1}(\Omega)$, $s > 0$, then the following error estimates hold*

$$\|u - u_h\|_{H^1(\Omega)} \lesssim \frac{h^\eta}{p^s}\|u\|_{H^{s+1}(\Omega)}, \quad \eta = \min\{s, p\},$$

$$\|u - u_h\|_{L^2(\Omega)} \lesssim \frac{h^{\eta+1}}{p^s}\|u\|_{H^{s+1}(\Omega)}, \quad \eta = \min\{s, p\}.$$

The notations $\|\cdot\|_{L^2(\Omega)}$ and $\|\cdot\|_{H^k(\Omega)}$ denote the standard norms in $L^2(\Omega)$ and in $H^k(\Omega)$, $k \geq 1$, respectively.

Let $\{\phi_j\}_{j=1}^{N_h}$ be a basis of functions for the finite element space V_{hp}^C such that $V_{hp}^C = span\{\phi_j\}_{j=1}^{N_h}$, then Problem 1.3 is equivalent to the following linear system of equations

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

where

$$A \in \mathbb{R}^{N_h \times N_h}, \quad a_{ij} = a_C(\phi_j, \phi_i) \quad i, j = 1, \ldots, N_h, \tag{1.4}$$
$$\mathbf{f} \in \mathbb{R}^{N_h}, \quad f_i = F_C(\phi_i) \quad i = 1, \ldots, N_h$$

and $\mathbf{u} \in \mathbb{R}^{N_h}$, $\mathbf{u} = [u_1, \ldots, u_{N_h}]^T$ is the vector containing the unknown coefficients of the expansion of the discrete solution u_h in the chosen basis.
The stiffness matrix A associated to the continuous Galerkin finite element method has the following property.

Property 1.1. *The matrix A defined in* (1.4) *is symmetric and positive definite.*

Property 1.1 holds because the bilinear form $a_C(\cdot,\cdot)$ associated to the matrix A is coercive and symmetric.

1.2.2 Discontinuous Galerkin Finite Element Methods

In this section we introduce the discontinuous Galerkin finite element method. We define the broken Sobolev space on the grid $\mathcal{T}_h$ as

$$H^s(\mathcal{T}_h) = \{v \in L^2(\Omega) : v|_T \in H^s(T) \quad \forall T \in \mathcal{T}_h\},$$

$s > 3/2$. We denote by e the edges of elements of $\mathcal{T}_h$. Let $\Gamma_{\mathcal{E}}$ be the set of interior edges of the mesh $\mathcal{T}_h$. If $e \in \Gamma_{\mathcal{E}}$ then e has a unit normal vector $\mathbf{n}_e$. If $e \in \partial\Omega$ then $\mathbf{n}_e$ coincides with $\mathbf{n}$, the unit outward normal vector to $\partial\Omega$.
Let T_1^e and T_2^e be two adjacent elements that share one common side e. We know that if $v \in H^s(\mathcal{T}_h)$, $s \geq 1$, the trace of v along any side of one element T is well defined. It follows that T_1^e and T_2^e have the two traces of v well defined on e.
We can add those values and obtain an average $\{v\}$ for v along e or we can subtract and get a jump $[v]$ for v along e. We assume that the normal vector $\mathbf{n}_e$ is oriented from T_1^e to T_2^e.

$$\{v\} = \frac{1}{2}v|_{T_1^e} + \frac{1}{2}v|_{T_2^e} \quad [v] = v|_{T_1^e} - v|_{T_2^e} \quad \forall e = \partial T_1^e \cap \partial T_2^e.$$

If e belongs to the boundary $\partial\Omega$ we extend these definitions as follows

$$\{v\} = [v] = v|_{T_1^e} \quad \forall e = \partial T_1^e \cap \partial\Omega.$$

Now we introduce the following bilinear form $J : H^s(\mathcal{T}_h) \times H^s(\mathcal{T}_h) \to \mathbb{R}$

$$J(u,v) = \sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \sigma^e \frac{p^2}{|e|^\beta} \int_e [u][v]\, d\gamma,$$

which penalizes the jump of the function. Here σ^e is the penalty parameter and $\beta > 0$ depends on the dimension of the domain (in our case $\Omega \subset \mathbb{R}^2$ and $\beta = 1$). The notation $|e|$ means the length of the edge e.
Next we define the bilinear form $a_D : H^s(\mathcal{T}_h) \times H^s(\mathcal{T}_h) \to \mathbb{R}$ as

$$\begin{aligned} a_D(u,v) = &\sum_{T \in \mathcal{T}_h} \int_T \nabla u \cdot \nabla v \, d\Omega - \sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \int_e \{\nabla u \cdot \mathbf{n}_e\}[v]\, d\gamma \\ &- \sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \int_e \{\nabla v \cdot \mathbf{n}_e\}[u]\, d\gamma + J(u,v), \end{aligned}$$

and the functional $F_D : H^s(\mathcal{T}_h) \to \mathbb{R}$ as

$$F_D(v) = \int_\Omega f v \, d\Omega - \sum_{e \in \partial\Omega} \int_e \nabla v \cdot \mathbf{n}_e \, g \, d\gamma + \sum_{e \in \partial\Omega} \int_e \sigma^e \frac{p^2}{|e|^\beta} v g \, d\gamma.$$

The weak formulation of the Poisson Problem 1.1 in the broken Sobolev space $H^s(\mathcal{T}_h)$ is given by

Problem 1.4. *Find $u \in H^s(\mathcal{T}_h)$, $s > 3/2$, such that*

$$a_D(u, v) = F_D(v) \quad \forall v \in H^s(\mathcal{T}_h).$$

Note that Dirichlet boundary condition are weakly imposed.
Now we can proceed to the discretization of this alternative formulation of the Poisson Problem 1.1. Let $V_{hp}^D \subset H^s(\mathcal{T}_h)$, $s > 3/2$, be a family of finite dimensional spaces defined as

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

The approximate DG problem is of the form

Problem 1.5. *Find $u_h \in V_{hp}^D$ such that*

$$a_D(u_h, v_h) = F_D(v_h) \quad \forall v_h \in V_{hp}^D.$$

We note that the functions in V_{hp}^D are discontinuous along the edges of $\mathcal{T}_h$. Because of this, the basis functions of V_{hp}^D have a support contained in one element T. Formulation defined in Problem 1.5 is known as symmetric interior penalty (SIP) method [Wheeler, 1978, Arnold, 1982]. In order to prove the Proposition 1.3, we define the DG norm and we recall some useful properties.
The first one is the inverse inequality, while the second one gives the trace inequalities, cf. [Ciarlet, 1978, Warburton and Hesthaven, 2003].

Definition 1.1. *The space V_{hp}^D is equipped with the norm*

$$\|v\|_{DG}^2 = \sum_{T \in \mathcal{T}_h} \|\nabla v\|_{L^2(T)}^2 + \sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \|\sqrt{\sigma^e \frac{p^2}{|e|^\beta}}[v]\|_{L^2(e)}^2.$$

Property 1.2. *For any $v|_T \in \mathbb{P}_p$, $T \in \mathcal{T}_h$, we have the inverse inequality*

$$\|\nabla v\|_{L^2(T)}^2 \lesssim p^4 h^{-2} \|v\|_{L^2(T)}^2.$$

Property 1.3. *For any $v|_T \in \mathbb{P}_p$, $T \in \mathcal{T}_h$, we have the trace inequalities*

$$\|v\|_{L^2(e)}^2 \lesssim p^2 h^{-1} \|v\|_{L^2(T)}^2, \quad \|\nabla v \cdot \mathbf{n}_e\|_{L^2(e)}^2 \lesssim p^2 h^{-1} \|\nabla v\|_{L^2(T)}^2,$$

where $e \in \Gamma_{\mathcal{E}} \cup \partial\Omega$ is an edge of the element $T \in \mathcal{T}_h$.

The following result ensures the well-posedness of Problem 1.5, cf. [Houston et al., 2002, Epshteyn and Rivière, 2007, Antonietti and Houston, 2011].

Proposition 1.3. *If $\sigma^e \gg \sigma_{min}$, the solution of Problem 1.5 exists and is unique.*

Proof. In order to prove existence and uniqueness of the solution of Problem 1.5 we verify that Strang lemma is satisfied. We have to check that the bilinear form $a_D(\cdot,\cdot)$ is continuous and coercive and that the linear functional $F_D(\cdot)$ is continuous with respect to the DG norm.
We have to show that

$$a_D(u,u) \gtrsim \|u\|_{DG}^2 \ \forall u \in V_{hp}^D,$$

$$a_D(u,v) \lesssim \|u\|_{DG}\|v\|_{DG} \ \forall u,v \in V_{hp}^D,$$

$$F_D(v) \lesssim \|v\|_{DG} \ \forall v \in V_{hp}^D.$$

We begin by demonstrating the continuity of $a_D(\cdot,\cdot)$. Using Cauchy-Schwarz inequality, we get

$$\sum_{T\in\mathcal{T}_h} \int_T \nabla u \cdot \nabla v \, d\Omega \leq \sum_{T\in\mathcal{T}_h} \|\nabla u\|_{L^2(T)}\|\nabla v\|_{L^2(T)},$$

$$\sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega} \sigma^e \frac{p^2}{|e|^\beta}\int_e [u][v]\, d\gamma \lesssim \sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega} \|[u]\|_{L^2(e)}\|[v]\|_{L^2(e)},$$

$$\int_e \{\nabla u\cdot \mathbf{n}_e\}[v]\, d\gamma \leq \|\{\nabla u\cdot\mathbf{n}_e\}\|_{L^2(e)}\|[v]\|_{L^2(e)}.$$

Now from Properties 1.2 and 1.3, we find

$$\int_e \{\nabla u\cdot\mathbf{n}_e\}[v]\, d\gamma \lesssim \left(\|\nabla u\|^2_{L^2(T_1^e)} + \|\nabla u\|^2_{L^2(T_2^e)}\right)^{1/2} \left(\frac{1}{|e|^\beta}\right)^{1/2} \|[v]\|_{L^2(e)},$$

where $T_1^e, T_2^e \in \mathcal{T}_h$ are the elements that share edge e. Summing over all the edges, we have

$$\sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega}\int_e \{\nabla u\cdot\mathbf{n}_e\}[v]\, d\gamma \lesssim \left(\sum_{T\in\mathcal{T}_h}\|\nabla u\|^2_{L^2(T)}\right)^{1/2}\left(\sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega}\frac{1}{|e|^\beta}\|[v]\|^2_{L^2(e)}\right)^{1/2}. \tag{1.5}$$

We observe that the term

$$\sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega}\int_e \{\nabla v\cdot\mathbf{n}_e\}[u]\, d\gamma$$

can be bounded in a similar way.
Collecting all of the previous estimates, we deduce that the bilinear form $a_D(\cdot,\cdot)$ is continuous. We now proceed by proving the coercivity of $a_D(\cdot,\cdot)$. We have

$$a_D(u,u) = \sum_{T\in\mathcal{T}_h}\|\nabla u\|^2_{L^2(T)} - 2\sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega}\int_e \{\nabla u\cdot\mathbf{n}_e\}[u]\, d\gamma$$
$$+ \sum_{e\in\Gamma_{\mathcal{E}}\cup\partial\Omega}\|\sqrt{\sigma^e\frac{p^2}{|e|^\beta}}[u]\|^2_{L^2(e)}.$$

Recalling equation (1.5) and applying Young inequality, for any $\delta \in [0,1]$, we find

$$\sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \int_e \{\nabla u \cdot \mathbf{n}_e\}[v]\, d\gamma \leq \frac{\delta}{2} \sum_{T \in \mathcal{T}_h} \|\nabla u\|^2_{L^2(T)} + \frac{\alpha}{2\delta} \sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \frac{1}{|e|^\beta} \|[v]\|^2_{L^2(e)},$$

where $\alpha > 0$ is a constant dependent on h and p that derives from Properties 1.2 and 1.3.
Thus we obtain the following lower bound for $a_D(\cdot,\cdot)$

$$a_D(u,u) \geq (1-\delta) \sum_{T \in \mathcal{T}_h} \|\nabla u\|^2_{L^2(T)} + \sum_{e \in \Gamma_{\mathcal{E}} \cup \partial\Omega} \frac{\sigma^e p^2 - \frac{\alpha}{\delta}}{|e|^\beta} \|[v]\|^2_{L^2(e)}.$$

Choosing, for instance, $\delta = 1/2$ and σ^e large enough, then we have the coercivity result. Finally, using again Cauchy-Schwarz inequality and Properties 1.2 and 1.3, we get that $F(\cdot)$ is continuous. □

The error estimates are the following, cf. [Perugia and Schötzau, 2002,Houston et al., 2002,Stamm and Wihler, 2010,Antonietti et al., 2015].

Theorem 1.2. *Let $u \in H^{s+1}(\mathcal{T}_h)$, $s \geq 1$, be the exact solution of the weak formulation defined in Problem 1.4 and let $u_h \in V_{hp}^D$ be its approximate solution obtained from Problem 1.5. Then the following error estimates hold*

$$\|u - u_h\|_{DG} \lesssim \frac{h^\eta}{p^{s-1/2}} \|u\|_{H^{s+1}(\mathcal{T}_h)}, \quad \eta = \min\{s,p\},$$

$$\|u - u_h\|_{L^2(\Omega)} \lesssim \frac{h^{\eta+1}}{p^s} \|u\|_{H^{s+1}(\mathcal{T}_h)}, \quad \eta = \min\{s,p\}.$$

The notations $\|\cdot\|_{L^2(\Omega)}$ and $\|\cdot\|_{H^k(\mathcal{T}_h)}$ denote the standard norms in $L^2(\Omega)$ and in the broken Sobolev space $H^k(\mathcal{T}_h)$, respectively.

As we did in the previous section, we can state the finite element method for the approximation of the Poisson Problem 1.5. Let $\{\phi_j\}_{j=1}^{N_h}$ be a basis of functions for the finite element space V_{hp}^D such that $V_{hp}^D = span\{\phi_j\}_{j=1}^{N_h}$, then Problem 1.5 is equivalent to the following linear system of equations

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

where

$$A \in \mathbb{R}^{N_h \times N_h}, \quad a_{ij} = a_D(\phi_j, \phi_i) \quad i,j = 1,\dots,N_h, \tag{1.6}$$
$$\mathbf{f} \in \mathbb{R}^{N_h}, \quad f_i = F_D(\phi_i) \quad i = 1,\dots,N_h$$

and $\mathbf{u} \in \mathbb{R}^{N_h}$, $\mathbf{u} = [u_1,\dots,u_{N_h}]^T$ is the vector containing the unknown coefficients of the expansion of the discrete solution u_h in the chosen basis.
The stiffness matrix A associated to the discontinuous Galerkin finite element method has the following property.

Property 1.4. *The matrix A defined in* (1.6) *is symmetric and positive definite.*

Property 1.4 holds because the bilinear form $a_D(\cdot,\cdot)$ associated to the matrix A is coercive (see Proposition 1.3) and symmetric.

1.3 Basis Functions for the Finite Element Spaces

In this section we describe the choice of the shape functions employed to span the conforming and the discontinuous finite element spaces.
We assume that $v_h \in V_{hp}^C$ is characterized by the values it takes at the points $N_i = (x_i, y_i)$, with $i = 1, \ldots, N_h$, and consequently the shape functions associated to the finite element space V_{hp}^C are defined as the Lagrangian functions associated to the nodes.
We define the nodes on the reference element $\hat{T} = \{(x,y) : x, y \geq 0, x + y \leq 1\}$ as Fekete points [Taylor et al., 2000, Briani et al., 2012]. In Figure 1.1 we show the Fekete points on the reference triangle for $p = 1, 2, 4, 10$.

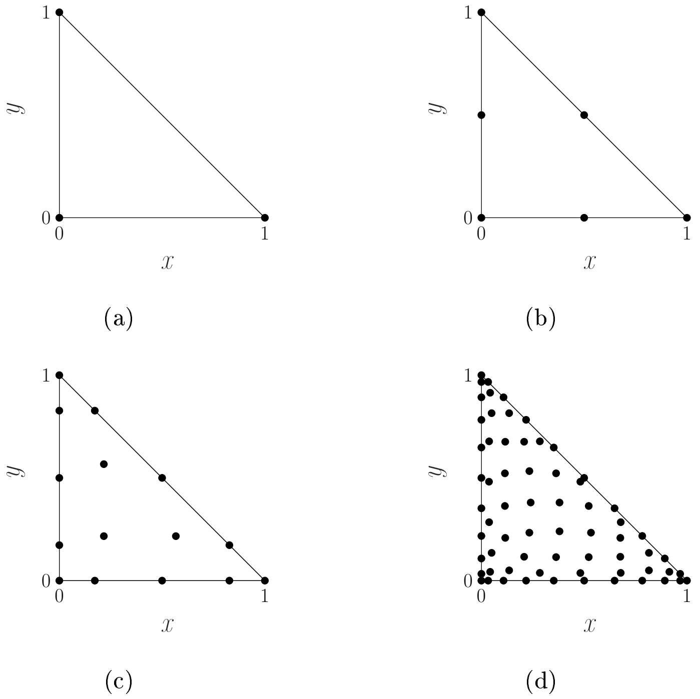


Figure 1.1: Fekete points on the reference triangle (a) $p = 1$, (b) $p = 2$, (c) $p = 4$, (d) $p = 10$.

The discrete space V_{hp}^C is spanned by $V_{hp}^C = span\{\phi_j\}_{j=1}^{N_h}$, where ϕ_i is the characteristic Lagrangian function of degree p associated to the Fekete point N_i, i.e.,

$$\phi_j(N_i) = \delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases} \quad i, j = 1, \ldots, N_h,$$

cf. [Ciarlet and Raviart, 1972].
We employ the same set of shape functions to span $V_{hp}^D = span\{\phi_j\}_{j=1}^{N_h}$, but now without any inter-element continuity constraint. We point out that other choices can be made for the basis functions, for example we can consider the ones based on modal expansion [Koornwinder, 1975, Dubiner, 1991, Hesthaven and Warburton, 2008].

Chapter 2

Multigrid Principles

In this chapter, we discuss how to solve a generic linear system of equations

$$A\mathbf{u} = \mathbf{f} \tag{2.1}$$

stemming from the discretization of differential equations, where A is a symmetric and positive definite matrix and sparse. A possibility is to use classical iterative methods such as Jacobi, dumped Jacobi, Gauss-Seidel, conjugate gradient and so on. The big defect that most of these methods have is that they suffer from stagnation after a few iterations. Indeed such methods possess the so-called smoothing property [Chan and Elman, 1989]. The error can be represented as a linear combination of the eigenvectors of the iteration matrix associated with the selected iterative method. Eigenvectors associated with high wavenumbers represent oscillatory modes, whereas eigenvectors associated with low wavenumbers represent smooth modes. The smoothing property refers to more efficient reduction of oscillatory components of the error than the smooth ones. For this reason multigrid methods have become one of the most powerful method to solve the large linear system of the form (2.1) [Brandt, 1982, Hackbusch, 1982, McCormick, 1987, Briggs et al., 2000, Trottenberg et al., 2001].

There are two main categories of multigrid methods: geometric and algebraic multigrid. Both are based on the representation of smooth error modes as oscillatory ones so the relaxation of the chosen iterative method (e.g. Jacobi or Gauss-Seidel method) will be more effective on all error components. In the following, we are going to detail how this representation occurs.

In this work we are interested in the algebraic multigrid method (AMG), but since it is based on some elementary concepts that arise from geometric multigrid it is natural to present both of them in a geometric "setting".

2.1 Geometric Multigrid

We follow the presentation of [Stüben and Trottenberg, 1982].

Let $\mathcal{T}_h$ be a fine grid associated with a discretized problem with numerical solution u_h. Thanks to the local Fourier analysis introduced by [Brandt, 1977b], we know that smooth error components on $\mathcal{T}_h$ can be represented as oscillatory modes if they are projected on $\mathcal{T}_H$, where $\mathcal{T}_H$ is a coarser grid respect to $\mathcal{T}_h$. The transition of the error from the fine grid $\mathcal{T}_h$ to the coarser one $\mathcal{T}_H$ is possible thanks to suitable intergrid operators. With the coarse grid idea in mind we can distinguish two basic elements of geometric multigrid: error smoothing by relaxation method and recursive correction on coarser grids, cf. Sections 2.1.2 and 2.1.3 below.

2.1.1 Intergrid Operators

In the geometric multigrid method there are two type of information transfers, therefore we need to define two intergrid operators based on the geometric information of the problem: prolongation and restriction operators, that are defined in the following.

Given a pair of coarser and finer meshes $\mathcal{T}_H$ and $\mathcal{T}_h$, respectively, we denote by V_H and V_h two finite dimensional element spaces associated to $\mathcal{T}_H$ and $\mathcal{T}_h$, respectively, such that $\dim(V_H) < \dim(V_h) < +\infty$.

Definition 2.1. *Let $\mathbf{v}_h \in V_h$ and $\mathbf{v}_H \in V_H$ be discrete functions defined on V_h and V_H, respectively. The interpolation or prolongation operator $I_H^h : V_H \to V_h$ is such that*

$$I_H^h \mathbf{v}_H = \mathbf{v}_h.$$

Definition 2.2. *Let $\mathbf{v}_h \in V_h$ and $\mathbf{v}_H \in V_H$ be discrete functions defined on V_h and V_H, respectively. The restriction operator $I_h^H : V_h \to V_H$ is such that*

$$I_h^H \mathbf{v}_h = \mathbf{v}_H.$$

If $V_H \subseteq V_h$, a simple example of these operators is given by the interpolation as I_H^h and the injection as I_h^H, cf. [Brandt, 1982, Gaspar et al., 2009, Gaspar et al., 2010] for example.

2.1.2 Smoothing

The geometric multigrid method involves the use of classical iterative methods like those of Jacobi or Gauss-Seidel featuring the smoothing property: after few iterations there will be an efficient reduction of oscillatory components of the error but the smooth ones will remain [Chan and Elman, 1989].

Denoting by $\mathbf{u}_*$ the approximate solution of (2.1) obtained via an iterative relaxation scheme, the error is defined as

$$\mathbf{e} = \mathbf{u} - \mathbf{u}_*, \tag{2.2}$$

where $\mathbf{u}$ denotes the exact solution of (2.1) and the residual is given by

$$\mathbf{r} = \mathbf{f} - A\mathbf{u}_*. \tag{2.3}$$

There exists a important relationship between the error and the residual.

Problem 2.1. *The residual equation read as follows*

$$A\mathbf{e} = \mathbf{r},$$

where $\mathbf{e}$ *and* $\mathbf{r}$ *are the error and the residual defined in* (2.2) *and* (2.3).

Having defined the two equations that guide the multigrid method we can now understand the role they have.

2.1.3 Coarse-Grid Correction

The geometric multigrid method is based on the concepts of smoothing and coarse-grid correction which are designed to complement each other. There are two reasons why we want to solve the problem on a coarser grid: first of all we can solve a smaller numerical problem, secondly with the use of a coarser grid we can improve the speed of convergence.
Since we use iterative methods with smoothing property to solve the problem on the fine grid $\mathcal{T}_h$ we know that after few iterations the convergence deteriorates. So we are interested to move on a coarser grid $\mathcal{T}_H$ in order to convert the smooth error components into oscillatory ones and then solve the residual equation on $\mathcal{T}_H$. The solution of the residual equation is an approximation of the error and it is then interpolated back on $\mathcal{T}_h$ and used to improve convergence of the iterative methods. Now we give a rigorous and formal structure to the newly introduced concepts. They can be represented by the procedure in Algorithm 2.1 which is known as two-grid correction scheme [Briggs et al., 2000].

Algorithm 2.1 One Iteration of Two-Grid Correction Scheme

```
function MG($\nu_1$,$\nu_2$,$A_h$,$\mathbf{u}_h^0$,$\mathbf{f}_h$,$A_H$,$I_h^H$,$I_H^h$)
    Relax $\nu_1$ times on $A_h\mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^0$
    Compute the fine-grid residual $\mathbf{r}_h^{\nu_1} = \mathbf{f}_h - A_h\mathbf{u}_h^{\nu_1}$
    Restrict $\mathbf{r}_h^{\nu_1}$ to the coarse grid by $\mathbf{r}_H = I_h^H\mathbf{r}_h^{\nu_1}$
    Solve $A_H\mathbf{e}_H = \mathbf{r}_H$
    Interpolate the coarse-grid error $\mathbf{e}_H$ to the fine grid by $\mathbf{e}_h = I_H^h\mathbf{e}_H$
    Correct the fine-grid approximation by $\mathbf{u}_h^{\nu_1+1} = \mathbf{u}_h^{\nu_1} + \mathbf{e}_h$
    Relax $\nu_2$ times on $A_h\mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^{\nu_1+1}$
    return $\mathbf{u}_h^{\nu_1+\nu_2+1}$
end function
```

As input we must provide two integer values ν_1 and ν_2 that are the number of pre-smoothing and post-smoothing iterations. We have also to give some geometric information about the problem: $\mathcal{T}_h$ and $\mathcal{T}_H$ are the two grids, A_h and A_H are the stiffness matrices defined on the fine and coarse grid, respectively. The relationship

between the discrete spaces associated with $\mathcal{T}_h$ and $\mathcal{T}_H$ is given by the intergrid operators I_h^H and I_H^h. Finally we have to provide the datum $\mathbf{f}_h$ and an initial guess for the solution $\mathbf{u}_h$.
To solve the residual equation on the coarser grid $\mathcal{T}_H$ we will employ a direct method.

2.1.4 Variational Properties

The coarse-grid problem operator, as presented by [Mandel et al., 1987,Briggs et al., 2000], can be defined so that it respects the following Galerkin condition

$$A_H = I_h^H A_h I_H^h \tag{2.4}$$

and

$$I_h^H = c\, {I_H^h}^T \quad c \in \mathbb{R}.$$

Equation (2.4) defines the coarse-grid operator. The second property is the relationship between the intergrid operators: it has to preserve constant vectors. These properties are desirable for most applications.

2.2 Algebraic Multigrid

In this section we define the algebraic multigrid method (AMG) from the multigrid concepts presented in Section 2.1. In [Briggs et al., 2000] we find an explanation of the AMG method with frequent analogies to the geometric multigrid one. More technical and detailed explanations on algebraic multigrid methods can be found in [Stüben, 1999, Xu and Zikatanov, 2017]. The first developments of the AMG method date back to [Brandt et al., 1985, Brandt, 1986].
In order to define the AMG method we need to build definitions, strategies, problems and properties without the use of geometric structures.
In the geometric multigrid the unknown variables are defined at grid points on a fine grid and we then select a subset of these points as a coarse grid. In the algebraic context, we identify the fine grid points as the indices of the unknowns, so the coarse grid is a subset of variables indices.
At this point starting from a linear system we must build a structure similar to the geometric one, based on graph theory.

2.2.1 Graph Theory

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ we want represent its geometric information using the graph theory [Gibbons, 1985, Diestel, 2010]. Before that we give some preliminary definitions on graphs useful for the sequel.

Definition 2.3. *An undirected graph $\mathcal{G}$ is a pair ($\mathcal{V}$,$\mathcal{E}$) where $\mathcal{V}$ is a set of vertices (points) and $\mathcal{E}$ is a set of edges. An edge $e \in \mathcal{E}$ is an unordered pair (j,k), where $j, k \in \mathcal{V}$, $j \neq k$.*

Definition 2.4. *The neighborhood N_i of a vertex i are the vertices adjacent to the vertex i, that is*

$$N_i = \{j : (i, j) \in \mathcal{E}\}.$$

Definition 2.5. *The degree of a vertex is the number of edges connected to it defined as*

$$d_i = |\{j : (i, j) \in \mathcal{E}\}|.$$

Definition 2.6. *A graph $\mathcal{G}_0 = (\mathcal{V}_0, \mathcal{E}_0)$ is called subgraph of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ if $\mathcal{V}_0 \subset \mathcal{V}$ and $\mathcal{E}_0 \subset \mathcal{E}$.*

Definition 2.7. *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, an independent set $\mathcal{V}_S \subseteq \mathcal{V}$ is a subset of vertices in which for each pair of nodes $i, j \in \mathcal{V}_S$ it follows $(i, j) \notin \mathcal{E}$. A maximal independent set $\mathcal{V}_M$ is a set such that $\mathcal{V}_M \cup \{i\}$ is not independent for every $i \in \mathcal{V} \setminus \mathcal{V}_M$.*

Definition 2.8. *The adjacency graph of a symmetric matrix $A \in \mathbb{R}^{n \times n}$ is an undirected graph denoted by $\mathcal{G}(A) = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V}$ and $\mathcal{E}$ defined as*

$$\mathcal{V} = \{1, \ldots, n\}, \quad \mathcal{E} = \{(j, k) : a_{jk} \neq 0\}.$$

An example of adjacency graph of a symmetric matrix is given in Figure 2.1.

$$A = \begin{pmatrix} * & * & * & * & * & 0 \\ * & * & * & 0 & 0 & 0 \\ * & * & * & * & 0 & 0 \\ * & 0 & * & * & * & * \\ * & 0 & 0 & * & * & 0 \\ 0 & 0 & 0 & * & 0 & * \end{pmatrix}$$

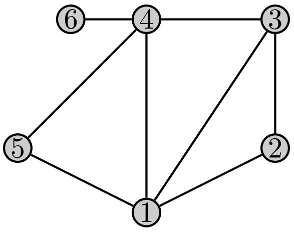


Figure 2.1: Symmetric matrix (left) and its associated adjacency graph (right).

The adjacency graph of a matrix gives us all the geometric information that we need: the grid is entirely defined by that graph. Now that we can represent the fine grid we want to understand how to build the coarse grid. Our goal is to select a subgraph of the adjacency graph associated to a given symmetric matrix $A \in \mathbb{R}^{n \times n}$.

2.2.2 Algebraic Smoothing

In the geometric context we can define the smoothing property of an iterative method with the local Fourier analysis of the error modes. Instead in the algebraic one we must proceed by analogy. Thus, we define algebraic smooth error to be any error that is not reduced effectively by the chosen iterative method [Brandt, 1986].
Given a smoothing operator $R \in \mathbb{R}^{n \times n}$ associated to the selected iterative method, formally we write:

Definition 2.9. *We define an error $\mathbf{e} \in \mathbb{R}^n$ to be algebraically smooth if it converges slowly with respect to a given smoothing operator R, that is, if $R\mathbf{e} \approx \mathbf{e}$.*

If the error is algebraically smooth, from the error equation $A\mathbf{e} = \mathbf{r}$ we get

$$A\mathbf{e} \approx \mathbf{0}, \tag{2.5}$$

which means that smooth error has relatively small residual. One immediate implication of (2.5) is that $r_i \approx 0$, $i = 1, \ldots, n$, so

$$a_{ii} e_i \approx -\sum_{j \neq i} a_{ij} e_j \quad i = 1, \ldots, n. \tag{2.6}$$

Equation (2.6) says that if $\mathbf{e}$ is a smooth error then e_i, $i = 1, \ldots, n$, can be approximate by a weighted average of its neighbors. This condition is important to construct the transfer operators.
A key step is to build the coarse-grid subgraph of $\mathcal{G}(A)$. This subgraph has to well represent the fine-grid smooth error as oscillatory on the coarse grid.
In the following we present some basic principles to select the coarse-grid subgraph.

2.2.3 Influence and Dependence

The fine grid can be represented by the adjacency graph of matrix $A \in \mathbb{R}^{n \times n}$. We need to identify vertices to be deleted from the graph to provide an appropriate subgraph of $\mathcal{G}(A)$. To do this we must provide a measure of the importance of the vertex through its connections. A strength function associated to a matrix is such that

$$s : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+.$$

Definition 2.10. *Given a threshold $0 < \theta \leq 1$, we say that the vertex i strongly depends on j if*

$$s(i, j) \geq \theta.$$

Definition 2.11. *If the vertex i strongly depends on j, then j strongly influences i.*

There are several definitions of strength functions, some examples are reported below.

$$s_1(i,j) = -\frac{a_{ij}}{\max_{k\neq i}(-a_{ik})}, \tag{2.7}$$

$$s_2(i,j) = \frac{|a_{ij}|}{\frac{1}{|N(i)|}\sum_{k\neq i}|a_{ik}|}, \tag{2.8}$$

$$s_3(i,j) = -\frac{a_{ij}}{\min\left(\max_{k\neq i}(-a_{ik}), \max_{k\neq j}(-a_{jk})\right)}, \tag{2.9}$$

$$s_4(i,j) = \frac{|a_{ij}|}{\min\left(\frac{1}{|N(i)|}\sum_{k\neq i}|a_{ik}|, \frac{1}{|N(j)|}\sum_{k\neq j}|a_{jk}|\right)}, \tag{2.10}$$

$$s_5(i,j) = \frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}. \tag{2.11}$$

The strength functions defined in (2.9), (2.10) and (2.11) are symmetric. In particular (2.9) and (2.10) are the symmetrized version of strength functions defined in (2.7) and (2.8), respectively. If the strength function is symmetric, i.e. $s(i,j) = s(j,i)$, Definitions 2.10 and 2.11 can be written as a unique definition as follows

Definition 2.12. *Given a threshold* $0 < \theta \leq 1$, *we say that the vertices* i *and* j *are strongly connected if*

$$s(i,j) \geq \theta.$$

We then define the strength matrix $S \in \mathbb{R}^{n\times n}$ with entries

$$s_{ij} = \begin{cases} 1 & s(i,j) \geq \theta \\ 0 & otherwise \end{cases}.$$

2.2.4 Coarsening Strategies

Coarsening strategies are all based on graph partitioning of grid domain [Saad, 2003]. The most relevant techniques are based on maximal independent set and aggregation algorithms. In this section we give a general explanation that will be detailed in later chapters. As in the geometric problem, the coarse grid must be selected so that the smooth error can be well represented and that has fewer points than the fine grid, so that the residual problem can be solved with little expense.
Given the strength matrix $S \in \mathbb{R}^{n\times n}$ we want to select the subgraph of $\mathcal{G}(A)$ that best represents it. To do this we use the information in matrix S because it represents the strong connections. Therefore we apply the coarsening algorithms on $\mathcal{G}(S)$ which does not have the weak connections of $\mathcal{G}(A)$.
Without getting into the details we have that these algorithms have the following inputs/outputs.

```
Algorithm 2.2 Maximal Independent Set (MIS)
  Input: Graph $\mathcal{G}(S)$ with $n$ vertices
  Output: $\mathcal{V} = C \cup F$, where $C$ is a set of disconnected vertices and $F = \mathcal{V} \setminus C$
```

```
Algorithm 2.3 Aggregation
  Input: Graph $\mathcal{G}(S)$ with $n$ vertices
  Output: $\mathcal{V} = \bigcup_{k=1}^{m} \mathcal{V}_k$, $m \le n$, and $\mathcal{V}_k \cap \mathcal{V}_j = \emptyset$ for $k \neq j$
```

Note that we refer to the graph $\mathcal{G}(A)$ and to its subgraph with the terminology of the geometric multigrid, that is, $\mathcal{G}(A)$ is the fine grid and the subgraph is the coarse one.

2.2.5 Transfer and Coarse-Grid Operators

With the strategies defined in the previous section we understand how to build the coarse grid. At this point it is good to introduce a suitable notation to refer to the fine or coarse grid: as in the geometric case we refer to fine grid with the subindex h and to coarse one with the subindex H. More precisely for a matrix M and a vector $\mathbf{v}$ we have: on "fine" grid $M_h \in \mathbb{R}^{N_h \times N_h}$, $\mathbf{v}_h \in \mathbb{R}^{N_h}$ and on "coarse" grid $M_H \in \mathbb{R}^{N_H \times N_H}$, $\mathbf{v}_H \in \mathbb{R}^{N_H}$, with $N_H \le N_h$.
We define the algebraic transfer operators as follows.

Definition 2.13. *Let $\mathbf{v}_h \in \mathbb{R}^{N_h}$ and $\mathbf{v}_H \in \mathbb{R}^{N_H}$ be approximate functions. The interpolation or prolongation operator $I_H^h : \mathbb{R}^{N_H} \to \mathbb{R}^{N_h}$ is such that*

$$I_H^h \mathbf{v}_H = \mathbf{v}_h.$$

Definition 2.14. *Let $\mathbf{v}_h \in \mathbb{R}^{N_h}$ and $\mathbf{v}_H \in \mathbb{R}^{N_H}$ be approximate functions. The restriction operator $I_h^H : \mathbb{R}^{N_h} \to \mathbb{R}^{N_H}$ is such that*

$$I_h^H \mathbf{v}_h = \mathbf{v}_H.$$

We want that the properties defined in Section 2.1.4 hold also in the algebraic context [Mandel et al., 1987,Briggs et al., 2000]. For transfer operators we have

$$I_h^H = {I_H^h}^T, \tag{2.12}$$

instead the coarse-grid operator is constructed using the Galerkin condition

$$A_H = I_h^H A_h I_H^h. \tag{2.13}$$

In the next chapters we will define the interpolation operator and then, thanks to (2.12) and (2.13), the restriction and coarse-grid operators will be automatically defined.

2.2.6 Two-Level Correction

In this section we revise the concepts introduced in Section 2.1.3 within an algebraic setting. The idea at the basis of the algorithm is the same, but the geometrical notions are replaced by algebraic ones.

As input we must provide the number of smoothing iterations ν_1 and ν_2, $A_h \in \mathbb{R}^{N_h \times N_h}$ and the relationship between the two grids is given by the algebraic transfer operators I_h^H and I_H^h. Finally, we have to provide the datum $\mathbf{f}_h \in \mathbb{R}^{N_h}$ and an initial guess for the solution $\mathbf{u}_h^0 \in \mathbb{R}^{N_h}$. One iteration of the AMG algorithm is shown in Algorithm 2.4.

Algorithm 2.4 One Iteration of the Algebraic Two-Grid Correction Scheme

```
function AMG($\nu_1$,$\nu_2$,$A_h$,$\mathbf{u}_h^0$,$\mathbf{f}_h$,$A_H$,$I_h^H$,$I_H^h$)
    Relax $\nu_1$ times on $A_h\mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^0$        ▷ Pre-smoothing
    $\mathbf{r}_h^{\nu_1} = \mathbf{f}_h - A_h\mathbf{u}_h^{\nu_1}$                                          ▷ Residual
    $\mathbf{r}_H = I_h^H\mathbf{r}_h^{\nu_1}$                                                ▷ Restriction
    Solve $A_H\mathbf{e}_H = \mathbf{r}_H$
    $\mathbf{e}_h = I_H^h\mathbf{e}_H$                                                ▷ Interpolation
    $\mathbf{u}_h^{\nu_1+1} = \mathbf{u}_h^{\nu_1} + \mathbf{e}_h$                                           ▷ Correction
    Relax $\nu_2$ times on $A_h\mathbf{u}_h = \mathbf{f}_h$ with initial guess $\mathbf{u}_h^{\nu_1+1}$   ▷ Post-smoothing
    return $\mathbf{u}_h^{\nu_1+\nu_2+1}$
end function
```

2.3 Multigrid Cycles

Till now we have seen multigrid concepts on two grids, in the following we will extend them to more levels [Briggs, 1987, Briggs et al., 2000]. Since we are particularly interested in AMG we present the extension of the two-level multigrid in algebraic context [Briggs and McCormick, 1987]. We need a new notation associated with levels: the current level is denoted by k and the number of unknowns on the level by N_k.

Denoting by K the coarsest level, the following components are needed for AMG:

1. **Grid Operators:** $A = A_1, \ldots, A_K$;
2. **Intergrid Operators:** $I_{k+1}^k : \mathbb{R}^{N_{k+1}} \to \mathbb{R}^{N_k}$ and $I_k^{k+1} : \mathbb{R}^{N_k} \to \mathbb{R}^{N_{k+1}}$,
   $k = 1, \ldots, K-1$.

With the use of the grid and intergrid operators introduced above, we solve the linear system of equations defined in (2.1): the stopping criterion is given by a maximum number of iterations ($N_{iter} \leq N_{max}$) and the residual norm normalized respect to the datum must be under a certain given tolerance ($||\mathbf{r}|| \leq tol\,||\mathbf{f}||$).

As shown in Algorithm 2.5, until the stopping criterion is satisfied, we repeat an iteration of a generic algebraic μ-cycle scheme (AMG-μCycle). Precisely, the family of algebraic μ-cycle schemes combine recursively the concept of coarse-grid correction.

Algorithm 2.5 AMG Solve Phase

```
  while $N_{iter} \leq N_{max}$ & $||\mathbf{r}|| \leq tol\,||\mathbf{f}||$ do
      $\mathbf{u}$ =AMG-$\mu$Cycle($A$,$\mathbf{f}$)
      $\mathbf{r} = \mathbf{f} - A\mathbf{u}$
  end while
```

2.3.1 Algebraic μ-cycle Scheme

In this section we present the algebraic μ-cycle schemes: they combines the coarse-grid correction idea on more levels [Briggs and McCormick, 1987], as shown in Algorithm 2.6.

Algorithm 2.6 One Iteration of Algebraic μ-cycle Scheme (AMG-μCycle)

```
  function AMG-$\mu$Cycle($\nu_1$,$\nu_2$,$A_k$,$\mathbf{u}_k^0$,$\mathbf{f}_k$,$I_k^{k+1}$,$I_{k+1}^k$)
      if $k = K$ then
          Solve $A_K\mathbf{u}_K = \mathbf{f}_K$                                    ▷ 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
          $\mathbf{r}_k^{\nu_1} = \mathbf{f}_k - A_k\mathbf{u}_k^{\nu_1}$                                    ▷ Residual
          $\mathbf{f}_{k+1} = I_k^{k+1}\mathbf{r}_k^{\nu_1}$                                     ▷ Restriction
          $\mathbf{u}_{k+1}^0 = \mathbf{0}$
          $\mathbf{u}_{k+1}^{\nu_1+\nu_2+1}$ = AMG-$\mu$Cycle($\nu_1$,$\nu_2$,$A_{k+1}$,$\mathbf{u}_{k+1}^0$,$\mathbf{f}_{k+1}$,$I_{k+1}^{k+2}$,$I_{k+2}^{k+1}$)    $\mu$ volte
          $\mathbf{u}_k^{\nu_1+1} = \mathbf{u}_k^{\nu_1} + I_{k+1}^k\mathbf{u}_{k+1}^{\nu_1+\nu_2+1}$               ▷ 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
```

In practice, only $\mu = 1$ and $\mu = 2$ are used. If we have a μ-cycle scheme with $\mu = 1$ we refer to it as a V-cycle, instead if we have $\mu = 2$ we call the method W-cycle. In particular we denote with V(ν_1,ν_2)-cycle and W(ν_1,ν_2)-cycle the two methods above with ν_1 pre-smoothing and ν_2 post-smoothing iterations [Briggs et al., 2000].

In Figures 2.2 and 2.3 we report the pictures describing the different approaches detailed in this section.

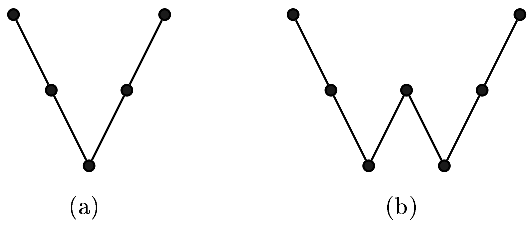


Figure 2.2: Illustration of multigrid cycles on three levels: (a) V-cycle and (b) W-cycle.

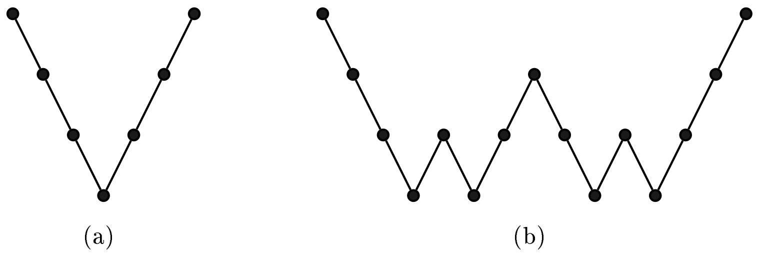


Figure 2.3: Illustration of multigrid cycles on four levels: (a) V-cycle and (b) W-cycle.

2.4 Algebraic Multigrid as Preconditioner

Algebraic multigrid was initially construct as a stand-alone solver, but in order to increase the robustness it is useful to combine it with other methods. In the simplest case AMG can be used as a preconditioner to accelerate the convergence of Krylov based iterative schemes, such as conjugate gradient [Kettler, 1982].

The need to use AMG with acceleration methods such as conjugate gradient comes from the fact that in many practical cases the selection of coarse grid and transfer operators is not optimal. For this reason AMG is used as preconditioner to accelerate the convergence.

In the AMG preconditioned conjugate gradient method we have to compute the preconditioned residual $\mathbf{z}$, by AMG as a preconditioner, i.e.

$$\mathbf{z} = \text{AMG-}\mu\text{Cycle}(A, \mathbf{r}).$$

For more details on the preconditioned conjugate gradient method with AMG see Algorithm 2.7, cf. [Quarteroni et al., 2007].

Algorithm 2.7 Preconditioned Conjugate Gradient with AMG

```
function PCG-$\mu$Cycle($A$,$\mathbf{u}_0$,$\mathbf{f}$,$N_{max}$,$tol$)
    $N_{iter} = 0$
    $\mathbf{r}_0 = \mathbf{f} - A\mathbf{u}_0$
    $\mathbf{z}_0 =$ AMG-$\mu$Cycle($A, \mathbf{r}_0$)                                ▷ PCG-step
    $\mathbf{p}_0 = \mathbf{z}_0$
    while $N_{iter} \leq N_{max}$ & $||\mathbf{r}_0|| \leq tol\,||\mathbf{f}||$ do              ▷ PCG-iteration
        $\alpha = \frac{\mathbf{r}_0^T \mathbf{z}_0}{\mathbf{p}_0^T A \mathbf{p}_0}$
        $\mathbf{u} = \mathbf{u}_0 + \alpha \mathbf{p}_0$
        $\mathbf{r} = \mathbf{r}_0 - \alpha A \mathbf{p}_0$
        $\mathbf{z} =$ AMG-$\mu$Cycle($A, \mathbf{r}$)                             ▷ PCG-step
        $\beta = \frac{\mathbf{r}^T \mathbf{z}}{\mathbf{r}_0^T \mathbf{z}_0}$
        $\mathbf{p} = \mathbf{z} + \beta \mathbf{p}_0$
        $\mathbf{u}_0 = \mathbf{u}$, $\mathbf{r}_0 = \mathbf{r}$, $\mathbf{p}_0 = \mathbf{p}$, $\mathbf{z}_0 = \mathbf{z}$                   ▷ Save old variables
        $N_{iter} = N_{iter} + 1$
    end while
end function
```

Chapter 3

Algebraic Multigrid for Conforming Finite Element Method

In this chapter we discuss algebraic multigrid methods for the solution of the linear system stemming from continuous Galerkin finite element approximations of the Poisson problem. We have to construct suitable coarsening strategies and define how to construct the associated transfer operators. There are three techniques in the literature: classical, aggregation and energy minimization algebraic multigrid. Classical AMG [Stüben, 1983, Ruge and Stüben, 1987] is based on maximal independent set as coarsening strategy and its transfer operators are defined with the definition of smooth error. Aggregation and energy minimization AMG are both based on aggregation strategies, but they differ for the definition of transfer operators. Aggregation algebraic multigrid [Vakhutinsky et al., 1979, Blaheta, 1986] defines the transfer operators in such a way that constant vectors are preserved, whereas energy minimization AMG operators have to satisfy certain conditions on the energy of the problem [Mandel et al., 1999, Olson et al., 2011]. Here we focus on classical and smoothed aggregation algebraic multigrid methods. The smoothed aggregation technique [Vaněk, 1992, Vaněk et al., 1996] is an improvement of method based on aggregation strategy.

The development of these methods is based on the hypothesis that the matrix associated with the linear system is an M-matrix, that is,

Definition 3.1. *A symmetric definite positive matrix $A \in \mathbb{R}^{n\times n}$ is called M-matrix if it satisfy the following properties:*

$$a_{ii} > 0 \quad i = 1, \dots, n,$$

$$a_{ij} \leq 0 \quad i \neq j,\ i, j = 1, \dots, n.$$

The matrices stemming from low order (i.e. linear) continuous Galerkin finite element discretizations are M-matrices, whereas if $p \geq 2$ the resulting stiffness matrices belong to the class of essentially positive matrices [Brandt, 1986]. For this reason

we first present the two techniques for the M-matrices because there is a direct correspondence between grid points and unknown variables and it allows us to build the algebraic multigrid method based on the geometric one. Then we will extend these concepts to essentially positive matrices for which the unknown variables are associated to grid points and high order degrees of freedom.
Without loss of generality in this chapter we will consider only two-level methods, except for convergence estimates.
Finally, we introduce the following inner products, that will be useful for convergence estimates.

Definition 3.2. *Given a symmetric positive definite matrix $A \in \mathbb{R}^{n\times n}$, let $D \in \mathbb{R}^{n\times n}$ be the diagonal matrix of A, we define*

$$(\mathbf{u},\mathbf{v})_0 = (D\mathbf{u},\mathbf{v}), \quad (\mathbf{u},\mathbf{v})_1 = (A\mathbf{u},\mathbf{v}), \quad \forall \mathbf{u},\mathbf{v} \in \mathbb{R}^n,$$

where $(\cdot,\cdot)$ is the Euclidian product. With the notation $||\cdot||_0$, $||\cdot||_1$ we denote their induced norms.

3.1 Classical Algebraic Multigrid for M-matrices

In this section we present the classical algebraic multigrid developed by [Stüben, 1983,Ruge and Stüben, 1987]. Later developments can be found in [Cleary et al., 1998,Cleary et al., 2000]. The topic is described using as a guideline [Stüben, 1999]. Since this method is based on the definition of algebraic smooth error we first give a new interpretation of it and then we detail how to construct the coarsening strategy and the interpolation operator. After that we show the convergence results.

3.1.1 Interpretation of Algebraic Smooth Error

We have seen in Section 2.2.2 that, for a given smoothing operator $R_h \in \mathbb{R}^{N_h\times N_h}$, the algebraic smooth error is characterized by the relation $R_h\mathbf{e}_h \approx \mathbf{e}_h$. Recalling that if the error is smooth then $A_h\mathbf{e}_h \approx \mathbf{0}_h$, it follows that

$$a_{ii}e_i \approx -\sum_{j\neq i} a_{ij}e_j \quad i = 1,\dots,N_h. \tag{3.1}$$

If A is an M-matrix equation (3.1) can be written with a more intuitive interpretation. An algebraic smooth error satisfies the following property [Brandt, 1986,Ruge and Stüben, 1987]

$$\frac{1}{2}\sum_{i,j=1}^{N_h}(-a_{ij})(e_i-e_j)^2 + \sum_{i=1}^{N_h}\left(\sum_{j=1}^{N_h} a_{ij}\right)e_i^2 \ll \sum_{i=1}^{N_h} a_{ii}e_i. \tag{3.2}$$

If $\sum_{i \neq j} |a_{ij}| \approx a_{ii}$, equation (3.2) means that, on the average for each i

$$\sum_{\substack{j=1\\j\neq i}}^{N_h} \frac{|a_{ij}|}{a_{ii}} \frac{(e_i - e_j)^2}{e_i^2} \ll 1 \quad i = 1, \dots, N_h.$$

That is, algebraic smooth error varies slowly in the direction of strong connections, i.e., if $|a_{ij}| \gg a_{ii}$, with $a_{ij} < 0$, then $e_i \approx e_j$. This observation is crucial to choose the best definition of strong connections and then to define the transfer operators.

3.1.2 Coarsening: Colouring Scheme

Given an M-matrix $A_h \in \mathbb{R}^{N_h \times N_h}$ and its adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we want to define a coarsening strategy based on maximal independent set. We construct a C/F-splitting based on strong couplings: C is the set of points of coarse grid and in F there are the points which belong only to the fine grid. Thanks to the interpretation of algebraic smooth error given in the previous section we assume the strength function to be

$$s(i,j) = -\frac{a_{ij}}{\max_{k \neq i}(-a_{ik})}. \tag{3.3}$$

Recalling the Definitions 2.10 and 2.11 from Section 2.2.3, we construct the strength matrix $S \in \mathbb{R}^{N_h \times N_h}$ with entries

$$s_{ij} = \begin{cases} 1 & -a_{ij} \geq \theta \max_{k \neq i}(-a_{ik}) \\ 0 & \textit{otherwise} \end{cases}$$

and its adjacency graph $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, $\mathcal{G}_S \subset \mathcal{G}$. Recalling the general concepts of coarsening strategies given in Section 2.2.4, we describe in detail the algorithm known as colouring scheme [Stüben, 1983, Ruge and Stüben, 1987]. Before describing the coarsening approach in detail, we need to give some more definitions and to introduce certain heuristic rules useful for the construction of the interpolation operator.

Definition 3.3. *We define the set S_i of all strong connections of the index i as*

$$S_i = \{j : (i,j) \in \mathcal{E}_S\}.$$

Definition 3.4. *We define the set S_i^T of variables strongly connected to the vertex i as*

$$S_i^T = \{j \in \mathcal{V} : i \in S_j\}.$$

Since the relation of variables strongly connected is generally non-symmetric it is useful to have defined both S_i and S_i^T.

Definition 3.5. *Let N_i be the neighborhood of the vertex i as defined in Definition 2.4. For a given C/F-splitting its points can be divided into three categories:*

1. *the coarse interpolatory set for i: $C_i = C \cap S_i$,*
2. *the strong non-interpolatory set for i: $D_i^s = F \cap S_i$,*
3. *the weak non-interpolatory set for i: $D_i^w = N_i \setminus S_i$.*

Note that the set D_i^w may contain both coarse- and fine-grid points. Armed with these definitions, the heuristic criteria that guide the initial selection of the C-points are the following:

Heuristic 3.1. *For each $i \in F$, every point $j \in S_i$ should either be in C, or should be strongly connected to at least one point in C_i.*

Heuristic 3.2. *C should be a maximal subset of all points with the property that no C-point strongly depends on another C-point.*

Heuristic 3.1 is used to construct the interpolation operator (see Section 3.1.3 below) whereas Heuristic 3.2 is designed to control the size of the coarse grid. Since it is not always possible to enforce both Heuristics 3.1 and 3.2, we choose to enforce Heuristic 3.1 because the interpolation formula depends on it, while we use Heuristic 3.2 as a guide.
We now have all the ingredients to define the colouring scheme [Stüben, 1983, Ruge and Stüben, 1987]: the algorithm proceeds in two steps. We first make an initial partition into C- and F-points. The goal in the first step is to create a set of C-points that satisfies Heuristic 3.2 (see Algorithm 3.1). Then we make a second step, modifying initial F-points to C-points as necessary to enforce Heuristic 3.1 (see Algorithm 3.2). Note that in Algorithm 3.1 the notation $|\cdot|$ denotes the cardinality of a set.

Algorithm 3.1 Preliminary C-point choice

```
$C = \emptyset$, $F = \emptyset$, $U = \mathcal{V}$
$\lambda_i = |S_i^T|$, $i \in U$
while $U \neq \emptyset$ do
    Select $i$ with maximum $\lambda_i$, $i \in U$
    $C = C \cup \{i\}$, $U = U \setminus \{i\}$
    For all $j \in S_i^T \cap U$: $F = F \cup \{j\}$, $U = U \setminus \{j\}$
    $\lambda_i = |S_i^T \cap U| + 2|S_i^T \cap F|$, $i \in U$
end while
```

In Algorithm 3.1 we start by picking a first point to be a C-point. Then all points strongly connected to it are selected as F-points. Next we select another point from the remaining unselected points in set U to become a C-point and all points which are strongly connected to it become again F-points. This process is repeated until all points in U have been assigned to C or F. In order to avoid random C/F-splitting, we introduce λ_i which measure the value of the point $i \in U$ as a C-point: the point that has maximum measure is the best candidate to become a C-point.

Algorithm 3.2 Final C-point choice

```
$T = \emptyset$
**while** $T \subset F$ **do**
    Pick $i \in F \setminus T$, $T = T \cup \{i\}$
    $C_i = C \cap S_i$, $D_i^s = F \cap S_i$, $D_i^w = N_i \setminus S_i$, $\tilde{C}_i = \emptyset$
    **for all** $j \in D_i^s$ **do**
        **if** $S_j \cap C_i = \emptyset$ **then**
            **if** $\tilde{C}_i \neq \emptyset$ **then**
                $C = C \cup \{i\}$, $F = F \setminus \{i\}$
                $\tilde{C}_i = \emptyset$
                **exit for**
            **else**
                $\tilde{C}_i = \{j\}$, $C_i = C_i \cup \{j\}$, $D_i^s = D_i^s \setminus \{j\}$
            **end if**
        **end if**
    **end for**
    $C = C \cup \tilde{C}_i$, $F = F \setminus \tilde{C}_i$
**end while**
```

Algorithm 3.2 verifies that, for each F-point i, all points in D_i^s have a strong connection to at least one point in C_i. If there exists only one point $j \in D_i^s$ such that $i \in F$ does not satisfy Heuristic 3.1, then j becomes a C-point, otherwise, if the algorithm finds more than one point in D_i^s such that $i \in F$ does not satisfy Heuristic 3.1, then the F-point i is put into C. This is done in order to minimize the number of new C-points introduced. The algorithm terminates when all F-points have been tested, i.e. when the set T, which denotes the tested F-points, is such that $T \supseteq F$.

We next discuss a variant of Algorithm 3.2 where we introduce a simplification: when the F-point i is such that at least one point $j \in D_i^s$ does not strongly depend on C_i, then i becomes a C-point. In this way there are fewer checks to be performed, cf. Algorithm 3.3.

Algorithm 3.3 Final C-point choice (Variant)

```
$T = \emptyset$
while $T \subset F$ do
    Pick $i \in F \setminus T$, $T = T \cup \{i\}$
    $C_i = C \cap S_i$, $D_i^s = F \cap S_i$, $D_i^w = N_i \setminus S_i$
    for all $j \in D_i^s$ do
        if $S_j \cap C_i = \emptyset$ then
            $C = C \cup \{i\}$, $F = F \setminus \{i\}$
        end if
    end for
end while
```

Now we present a classic example in the literature of the colouring scheme for the Poisson problem with linear conforming discretizations on simplicial (Figure 3.1) and on cartesian (Figure 3.2) meshes [Stüben, 1999].

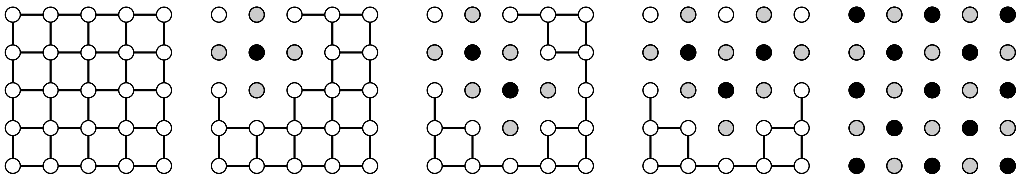


Figure 3.1: First steps and final result of the colouring scheme in case of simplicial meshes (black bullets are points in the set C, light grey bullets are points in the set F).

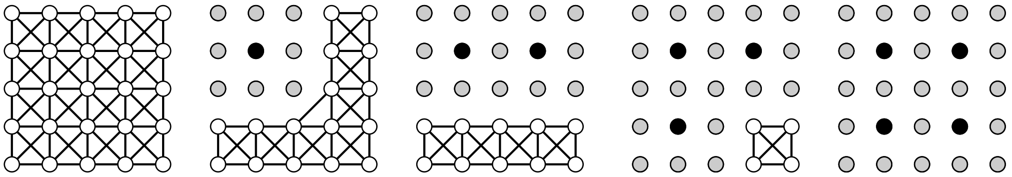


Figure 3.2: First steps and final result of the colouring scheme in case of cartesian meshes (black bullets are points in the set C, light grey bullets are points in the set F).

In Figures 3.1 and 3.2, the C-points are shown in black, the F-points are shown in grey and the undecided points are white. Edges of the graph are removed as the algorithm accounts for the dependencies of the new C- and F-points.

3.1.3 Interpolation Operator

Given a C/F-splitting we can define the interpolation operator. Two strategies are possible: standard [Ruge and Stüben, 1987] and direct interpolation [Stüben, 1983]. The difference between them is that the first technique takes into account all the connections, both weak and strong, while the second accounts only the strong ones. Here, we focus on the technique of [Ruge and Stüben, 1987].

We assume that, given $\mathbf{e}_H \in \mathbb{R}^{N_H}$, the interpolation operator is constructed such that

$$[\mathbf{e}_h]_i = [I_H^h \mathbf{e}_H]_i = \begin{cases} e_i^H & i \in C \\ \sum_{k \in C_i} w_{ik} e_k^H & i \in F \end{cases}, \quad i = 1, \dots, N_h,$$

where the interpolation weights w_{ik} have to be determined. As anticipated in Section 3.1.1, we are guided by the definition of smooth error to find good coefficients. Given the operator $A_h \in \mathbb{R}^{N_h \times N_h}$, we write equation (3.1) with the information of its adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in the following way

$$a_{ii} e_i^h \approx - \sum_{j \in N_i} a_{ij} e_j^h \quad i = 1, \dots, N_h. \tag{3.4}$$

Taking into account the splitting of the neighborhood N_i as in Definition 3.5, equation (3.4) can be rewritten as

$$a_{ii} e_i^h \approx - \sum_{j \in C_i} a_{ij} e_j^h - \sum_{j \in D_i^s} a_{ij} e_j^h - \sum_{j \in D_i^w} a_{ij} e_j^h \quad i = 1, \dots, N_h.$$

To determine w_{ik}, we need to replace e_j^h with approximations in terms of e_i^h or e_k^H, where $k \in C_i$.

The substitution in the summation on C_i is trivial: we have only to relabel e_j^h in e_k^H. Instead, for summations on D_i^s and D_i^w we have to make some considerations on the representation of the smooth error.

For points $j \in D_i^w$, we can replace e_j^h by e_i^h, obtaining

$$\left(a_{ii} + \sum_{j \in D_i^w} a_{ij} \right) e_i^h \approx - \sum_{k \in C_i} a_{ik} e_k^H - \sum_{j \in D_i^s} a_{ij} e_j^h \quad i = 1, \dots, N_h. \tag{3.5}$$

We can justify this substitution in the following way: in D_i^w there are the points j with weak connections respect to point i which have small a_{ij}, so any error committed in making this assignment will be relatively insignificant.

For points $j \in D_i^s$ it is a bit more complicated. Since those points are strongly connected to ones in the interpolatory set C_i, we make the following approximation

$$e_j^h \approx \frac{\sum_{k \in C_i} a_{jk} e_k^H}{\sum_{k \in C_i} a_{jk}}. \tag{3.6}$$

Notice that this approximation requires the Heuristic 3.1 to be verified and it is made such that it preserves constant vectors.

If we now substitute equation (3.6) into (3.5), which is then solved for e_i^h, we find that the interpolation weights are given by

$$w_{ik} = -\frac{a_{ik} + \sum_{j \in D_i^s} \frac{a_{ij}a_{jk}}{\sum_{l \in C_i} a_{jl}}}{a_{ii} + \sum_{j \in D_i^w} a_{ij}}.$$

3.1.4 Convergence of the V-cycle

In this section we consider convergence analysis for the V-cycle AMG classical methods.

The following result is due to [McCormick, 1985,Ruge and Stüben, 1987].

Theorem 3.1. *Let $A_k > 0$, $k = 1, \dots, K$. Suppose that the interpolation operators I_{k+1}^k, $k = 1, \dots, K-1$, have full rank and that the restriction and coarse-grid operators are constructed with the variational properties defined in Section 2.2.5. Let $R_k > 0$, $k = 1, \dots, K$, satisfy the smoothing property, i.e.,*

$$||R_k \mathbf{e}_k||_1^2 \leq ||\mathbf{e}_k||_1^2 - \alpha ||T_k \mathbf{e}_k||_1^2,$$

with $\alpha > 0$ independent of k and $\mathbf{e}_k$, and T_k, $k = 1, \dots, K-1$, defined as

$$T_k = I_k - I_{k+1}^k A_{k+1}^{-1} I_k^{k+1} A_k,$$

I_k being the identity matrix of dimension N_k. Then $\alpha \leq 1$ and the V-cycle convergence factor is bounded by $\sqrt{1-\alpha}$ with respect to the energy norm.

3.2 Smoothed Aggregation Algebraic Multigrid for M-matrices

In this section it is presented the smoothed aggregation algebraic multigrid [Vaněk, 1995,Vaněk et al., 1996]. Its first developments are in [Vaněk, 1992,Míka and Vaněk, 1992]. Since the smoothed aggregation is an improvement of the "unsmoothed" aggregation method [Blaheta, 1986] we will be point out the main differences.

3.2.1 Coarsening: Aggregation Algorithm

In this section we construct a disjoint decomposition of the set of points based on aggregation strategy. Formally, given the matrix $A_h \in \mathbb{R}^{N_h \times N_h}$ and its adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we want to split the set of points in a disjoint covering such that

$\mathcal{V} = \bigcup_{j=1}^{N_H} \mathcal{V}_j$, $N_H \leq N_h$, and $\mathcal{V}_l \cap \mathcal{V}_j = \emptyset$ for $l \neq j$. We assume the strength function to be

$$s(i,j) = \frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}. \tag{3.7}$$

Given the symmetric strength function in equation (3.7) we recall Definition 2.12 from Section 2.2.3.

Definition 3.6. *Given a threshold $0 < \theta \leq 1$, we say the nodes i and j are strongly connected if*

$$|a_{ij}| \geq \theta\sqrt{a_{ii}a_{jj}}.$$

With these definitions, as described in Section 2.2.3, we construct the strength matrix $S \in \mathbb{R}^{N_h \times N_h}$ with entries

$$s_{ij} = \begin{cases} 1 & |a_{ij}| \geq \theta\sqrt{a_{ii}a_{jj}} \\ 0 & otherwise \end{cases}$$

and its adjacency graph $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, $\mathcal{G}_S \subset \mathcal{G}$. Before describing the aggregation scheme of [Vaněk, 1995,Vaněk et al., 1996], we give two useful definitions.

Definition 3.7. *Let $\mathcal{S}_i$ be the set of the strongly neighborhood of point i defined as*

$$\mathcal{S}_i = \{j : (i,j) \in \mathcal{E}_S\} \cup \{i\}.$$

Definition 3.8. *Let $\mathcal{I}$ be the set of isolated points defined as*

$$\mathcal{I} = \{i : N_i = \emptyset\},$$

where N_i is defined in Definition 2.4.

With these definitions we can proceed to define the aggregation scheme as presented in [Vaněk et al., 1996], cf. Algorithm 3.4.

The aggregation scheme is made of four steps: initialization, startup aggregation, enlargement of the decomposition sets and handling the remnants. The first one define the set $\mathcal{U}$ of points to be aggregate. Then the second step creates an initial covering of disjoint set which does not necessary include all the points. In the third step we add, if possible, the remaining points $i \in \mathcal{U}$ to one of the set $\mathcal{V}_k$ to which node i is strongly connected. If more than one set exists the function find_strongest_connection(k) choose the one to which the point has the strongest connection. Finally, the last step creates aggregates with the remaining nodes $i \in \mathcal{U}$ formed by subsets of strongly connected neighborhoods.

Algorithm 3.4 Aggregation Scheme

```
$\mathcal{U} = \mathcal{V}_S \setminus \mathcal{I}$, $j = 0$                                ▷ Initialization
for all $i \in \mathcal{U}$ do                                          ▷ Startup Aggregation
    if $\mathcal{S}_i \subset \mathcal{U}$ then
        $j = j + 1$, $\mathcal{V}_j = \mathcal{S}_i$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_i$
    end if
end for
for $k = 1 : j$ do                              ▷ Enlarging the Decomposition Sets
    $\tilde{\mathcal{V}}_k = \mathcal{V}_k$
end for
for all $i \in \mathcal{U}$ do
    if $\exists k : \mathcal{S}_i \cap \tilde{\mathcal{V}}_k \neq \emptyset$ then
        $K = \text{find\_strongest\_connection}(k)$;
        $\mathcal{V}_K = \mathcal{V}_K \cup \{i\}$, $\mathcal{U} = \mathcal{U} \setminus \{i\}$
    end if
end for
for all $i \in \mathcal{U}$ do                                     ▷ Handling the Remnants
    $j = j + 1$, $\mathcal{V}_j = \mathcal{U} \cap \mathcal{S}_i$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{V}_j$
end for
```

In Figures 3.3 and 3.4 we report an example of the aggregation scheme for the Poisson problem with linear conforming discretizations on simplicial and on cartesian meshes, cf. [Yang, 2006]. Black points denote root-points, the ones coloured in grey are points associated to the second step and the lines make the aggregates.

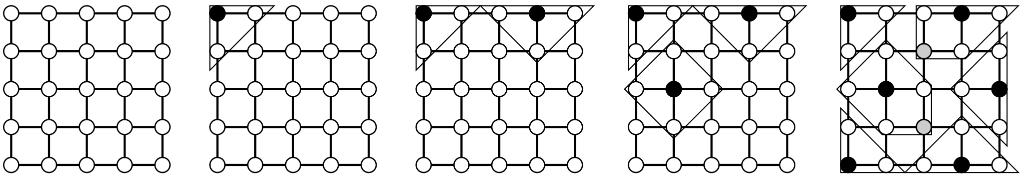


Figure 3.3: First steps and final result of the aggregation scheme in case of simplicial meshes.

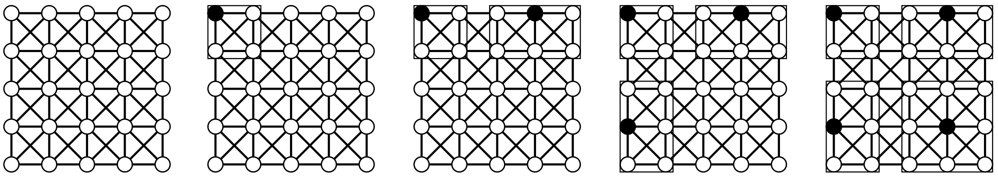


Figure 3.4: First steps and final result of the aggregation scheme in case of cartesian meshes.

3.2.2 Interpolation Operator

Now we describe the interpolation operator. First we construct a tentative piecewise constant interpolation such that it preserves constant vectors. Then it will improved by smoothing. The tentative operator is defined as

$$[\tilde{I}_H^h]_{ij} = \begin{cases} 1 & i \in \mathcal{V}_j \\ 0 & otherwise \end{cases}, \quad i = 1, \ldots, N_h, \; j = 1, \ldots, N_H.$$

Then we apply the Gram-Schmidt orthonormalization algorithm to each column of $\tilde{I}_H^h$ to improve conditioning.
Until now we have used the concepts of unsmoothed aggregation [Blaheta, 1986], now we proceed in the smoothing step as proposed in [Vaněk et al., 1996]. The final interpolation will now be improved by a simple damped Jacobi smoothing to get

$$I_H^h = (I_h - \omega D_h^{-1} A_h^F) \tilde{I}_H^h, \tag{3.8}$$

where $\omega = 2/3$ and A_h^F is the filtered matrix given by

$$a_{ij}^F = \begin{cases} a_{ij} & j \in \mathcal{S}_i, \, i \neq j \\ 0 & otherwise \end{cases}, \quad a_{ii}^F = a_{ii} - \sum_{\substack{j=1 \\ j \neq i}} (a_{ij} - a_{ij}^F), \tag{3.9}$$

where D_h denotes the diagonal of A_h and I_h is the identity matrix. The matrix A_h^F is built in order to take into account of the weak connections so that the interpolation will be more effective on strong ones.

3.2.3 Convergence of the V-cycle

In this section we present the convergence analysis of the smoothed aggregation algebraic multigrid method. Unlike the classical one, we have less restrictive assumptions to be respected and therefore we get a different convergence estimate. This analysis was introduced in [Bramble et al., 1991] and then extended for the proposed method by [Vaněk et al., 2001, Brezina et al., 2012]. The estimate result reads as follows.

Theorem 3.2. *Let the interpolation operators I_{k+1}^k, $k = 1, \ldots, K-1$, be defined as described in Section 3.2.2 and let the restriction and coarse-grid operators be constructed as described in Section 2.2.5. Let the smoothers $R_k > 0$, $k = 1, \ldots, K$, be symmetric positive definite matrices satisfying*

$$\lambda_{min}(I_k - R_k A_k) \geq 0 \quad \text{and} \quad \lambda_{min}(R_k) \gtrsim \frac{1}{\max_i \lambda_i(A_k)},$$

where I_k is the identity matrix of dimension N_k. Finally, let $\{\mathcal{V}_j^k\}_{j=1}^{N_k}$, $k = 1, \ldots,$ $K-1$, be the aggregates defined in Section 3.2.1 and assume that for every $\mathbf{v}_k \in \mathbb{R}^{N_k}$, $k = 1, \ldots, K$, it holds

$$\sum_{j=1}^{N_k} \min_{w \in \mathbb{R}} ||\mathbf{v}_k - w\mathbf{1}||^2_{l^2(\mathcal{V}_j^k)} \lesssim C^{k-1} ||\mathbf{v}_k||_1^2,$$

where $\mathbf{1} \in \mathbb{R}^{N_k}$ is the unitary vector and

$$||\mathbf{v}_k||^2_{l^2(\mathcal{V}_j^k)} = \sum_{m \in \mathcal{V}_j^k} v_m^2.$$

Then the V-cycle convergence factor is bounded by $1 - 1/g(K)$ in the energy norm, with $g(K) \sim (K-1)^3$.

3.3 Algebraic Multigrid Methods for Essentially Positive Matrices

High order continuous finite element discretizations of the Poisson problem does not lead to M-matrices. Nevertheless the resulting matrices are essentially positive type matrices, according to the following definition.

Definition 3.9. *Given a matrix $A \in \mathbb{R}^{n \times n}$ we define*

$$a_{ij}^- = \begin{cases} a_{ij} & a_{ij} < 0 \\ 0 & a_{ij} \geq 0 \end{cases} \quad \text{and} \quad a_{ij}^- = \begin{cases} 0 & a_{ij} \leq 0 \\ a_{ij} & a_{ij} > 0 \end{cases}, \quad i, j = 1, \ldots, n,$$

that denote negative and positive entries, respectively.

Definition 3.10. *A positive definite matrix $A \in \mathbb{R}^{n \times n}$ is called essentially positive type matrix if there exists a constant $\beta > 0$ such that, for all $\mathbf{e} \in \mathbb{R}^n$,*

$$\sum_{i,j=1}^{n} (-a_{ij})(e_i - e_j)^2 \geq \beta \sum_{i,j=1}^{n} (-a_{ij}^-)(e_i - e_j)^2.$$

According to [Brandt, 1986, Stüben, 1999] we have the following result.

Theorem 3.3. *Let A be the stiffness matrix stemming from high order continuous Galerkin finite element discretizations of the Poisson problem, then A is essentially positive.*

3.3.1 Classical Algebraic Multigrid

Let $A_h \in \mathbb{R}^{N_h \times N_h}$ be an essentially positive type matrix. In this case, we have that equation (3.2) becomes

$$\frac{\beta}{2} \sum_{i,j=1}^{N_h} (-a_{ij}^{-})(e_i - e_j)^2 + \sum_{i=1}^{N_h} \left(\sum_{j=1}^{N_h} a_{ij} \right) e_i^2 \ll \sum_{i=1}^{N_h} a_{ii} e_i, \tag{3.10}$$

which leads to the conclusion that the algebraic smooth error for essentially positive matrices varies slowly in the direction of large negative connections. Therefore, we need to split the connectivity structure only with negative off-diagonal entries. We then extend the classical algebraic multigrid to essentially positive type matrices. As proposed in [Ruge and Stüben, 1987,Stüben, 1999,Xu and Zikatanov, 2017], we proceed by constructing the filtered M-matrix A_h^M as

$$a_{ij}^M = a_{ij}^{-},\ i \neq j, \quad a_{ii}^M = a_{ii} + \sum_{\substack{j=1 \\ j \neq i}}^{N_h} a_{ij}^{+}, \tag{3.11}$$

and employ A_h^M to construct the C/F-splitting and the interpolation operator as done in Section 3.1.

We can then generalize the convergence result of Theorem 3.1 as follows, cf. [Ruge and Stüben, 1987,Stüben, 1999].

Theorem 3.4. *Let $A_k > 0$, $k = 1, \ldots, K$, be essentially positive type matrix with $\sum_{j=1}^{N_k} a_{ij} \geq 0$ that satisfies equation (3.10). Suppose that the interpolation operators I_{k+1}^k, $k = 1, \ldots, K-1$, have full rank and that the restriction and coarse-grid operators are constructed with the variational properties defined in Section 2.2.5. Let $R_k > 0$, $k = 1, \ldots, K$, satisfy the smoothing property, i.e.,*

$$||R_k \mathbf{e}_k||_1^2 \leq ||\mathbf{e}_k||_1^2 - \alpha ||T_k \mathbf{e}_k||_1^2,$$

with $\alpha > 0$ independent of k and $\mathbf{e}_k$, and T_k, $k = 1, \ldots, K-1$, defined as

$$T_k = I_k - I_{k+1}^k A_{k+1}^{-1} I_k^{k+1} A_k,$$

I_k being the identity matrix of dimension N_k. Then $\alpha\beta \leq 1$ and the V-cycle convergence factor is bounded by $\sqrt{1 - \alpha\beta}$ with respect to the energy norm.

3.3.2 Smoothed Aggregation Algebraic Multigrid

In this section we extend the smoothed algebraic multigrid to essentially positive type matrices.

We can use the method explained in Section 3.2 using either the matrix $A_h \in \mathbb{R}^{N_h \times N_h}$, cf. [Vaněk et al., 1996] or the filtered M-matrix $A_h^M \in \mathbb{R}^{N_h \times N_h}$ as defined

in equation (3.11), cf. [Xu and Zikatanov, 2017], but the smoothing step of the interpolation formula must be changed.
Indeed for essentially positive type matrices, if we filter the matrix as in equation (3.9), it means to distribute badly the matrix entries values associated to weak connections because they may be of any sign.
If we use the matrix $A_h \in \mathbb{R}^{N_h \times N_h}$, we have to substitute equation (3.8) with

$$I_H^h = (I_h - \omega D_h^{-1} A_h) \tilde{I}_H^h, \tag{3.12}$$

otherwise if we use the filtered M-matrix $A_h^M \in \mathbb{R}^{N_h \times N_h}$ we set

$$I_H^h = (I_h - \omega D_h^{-1} A_h^M) \tilde{I}_H^h. \tag{3.13}$$

It can be shown that the convergence result is the same of the Theorem 3.2, cf. [Vaněk et al., 2001, Brezina et al., 2012].

Chapter 4

Numerical Experiments: Algebraic Multigrid for Conforming Finite Element Method

In this chapter we demonstrate the robustness and the efficiency of algebraic multigrid in solving the linear system of equations stemming from conforming finite element discretizations of the Poisson problem. The test problem is given by

$$\begin{cases} -\Delta u = -(x^2+y^2)e^{xy} & \text{in } \Omega \\ \quad u = e^{xy} & \text{on } \partial\Omega \end{cases},$$

where $u(x,y) = e^{xy}$ is the exact solution and $\Omega = (0,1)^2$. We consider a sequence of structured and unstructured simplicial meshes, see Figures 4.1 and 4.2, respectively, of mesh size $h = 1/2, 1/4, 1/8, 1/16, 1/32$. Moreover we let the polynomial approximation degree p vary from 1 to 10.

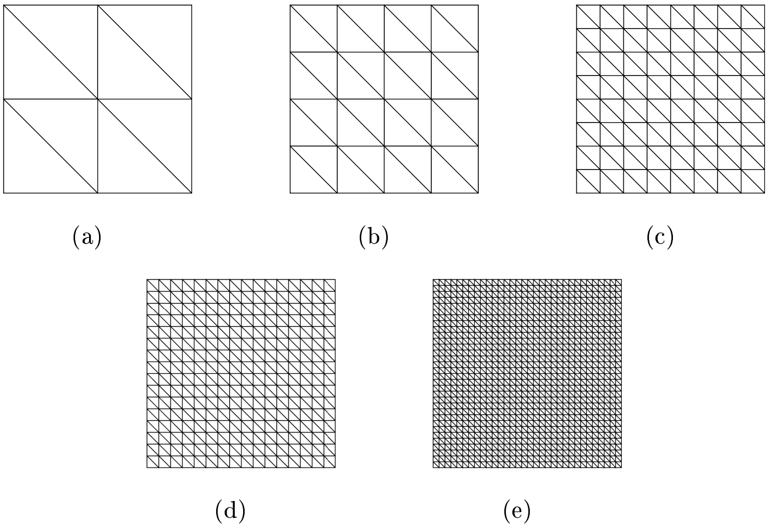


Figure 4.1: Sequence of structured simplicial meshes (a) $h = 1/2$, (b) $h = 1/4$, (c) $h = 1/8$, (d) $h = 1/16$, (e) $h = 1/32$.

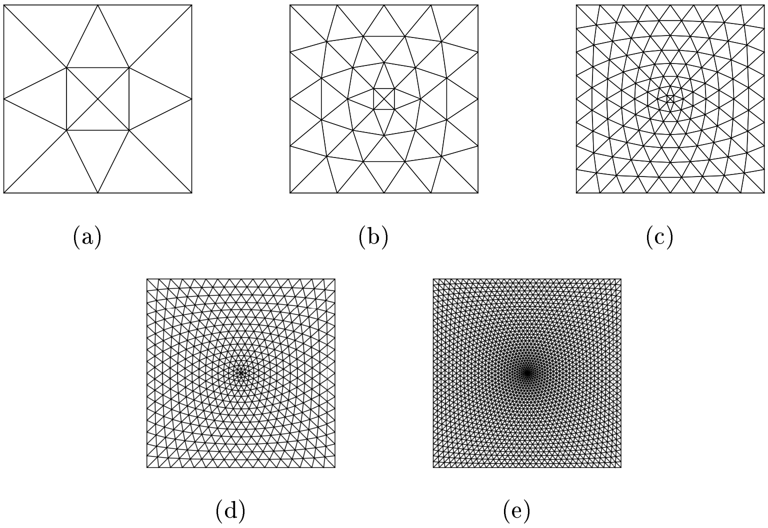


Figure 4.2: Sequence of unstructured simplicial meshes (a) $h = 1/2$, (b) $h = 1/4$, (c) $h = 1/8$, (d) $h = 1/16$, (e) $h = 1/32$.

For each h and p, we obtain a linear system of equations that we solve with algebraic multigrid methods presented in Chapters 2 and 3. Therefore, we have to choose its parameters, distinguishing the ones associated with the setup phase, i.e. the construction of coarser matrices and interpolation operators, and those associated with the solution phase.

4.1 Setup Phase

Given the matrix $A_h \in \mathbb{R}^{N_h \times N_h}$, (also denoted A_1 in multilevel setting), we fix the threshold of strong connection $\theta = 0.25$. Let K be the number of desired coarser "grids", then we build the coarser matrices and the interpolation operators with classical or smoothed aggregation setup. In general in our tests we have $K = 4$, but in order to show numerical convergence results we also let K vary from 2 to 5.
For classical AMG method we construct the coarser matrices as detailed in Chapter 3, whereas for smoothed aggregation AMG one we make an additional assumption about the smoothing step of interpolation operator, cf. Sections 3.2.2 and 3.3.2. Given the tentative interpolation operator $\tilde{I}_H^h$, we assume the final one to be

$$I_H^h = \begin{cases} (I_h - \omega D_h^{-1} A_h)\tilde{I}_H^h & p \leq 5 \\ (I_h - \omega D_h^{-1} A_h^M)\tilde{I}_H^h & p > 5 \end{cases}. \tag{4.1}$$

This choice is made on the basis of some numerical evidence: if $p > 5$ and we use (3.12) as smoothing step then the solution phase has a worse convergence behaviour than to use (3.13), on the other hand if we use for all p formula (3.13) we build unnecessarily the filtered M-matrix A_h^M for lower p. Therefore (4.1) is a good compromise.

In our numerical results we denote with C-AMG the classical algebraic multigrid, whereas we use SA-AMG to refer to the smoothed aggregation one.

4.2 Solution Phase

After the setup phase is completed we have to decide how to set the AMG-μCycle (Section 2.3), then we have to choose if we use it in a stand-alone algebraic multigrid solver or if we employ it as preconditioner to accelerate the convergence of the conjugate gradient method.

Relaxation is performed by Gauss-Seidel method with ν_1 and ν_2 as pre-smoothing and post-smoothing iterations, therefore we get the V(ν_1,ν_2)- and W(ν_1,ν_2)-cycles. In our numerical tests, for all h and p associated with finite element discretizations, we focus on V(1,1)- and W(1,1)-cycles in a stand-alone AMG solver, whereas we employ the V(2,2)-cycle with PCG method because if we use the V(1,1)-cycle the convergence of SA-AMG degenerates for higher values of p. We denote the preconditioned conjugate gradient with V(2,2)-cycle as PCG V(2,2)-cycle. In order to show numerical convergence results varying the number of smoothing iterations, we also study the V(ν_1,ν_2)- and W(ν_1,ν_2)-cycles with $\nu = \nu_1 = \nu_2 = 1, 2, 3$ with some fixed h and p.

All algorithms have as stopping criteria a tolerance of 10^{-8} on the relative residual norm and a maximum number of iterations of 150.

4.3 Numerical Results

Given the iteration counts N needed to reduce the initial relative residual below a tolerance $tol = 10^{-8}$,we compute the convergence factor ρ 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.

Before showing the numerical results we specify that the "-" notation indicates that the method does not satisfied the stopping criteria within 150 iterations.

In Tables 4.1, 4.2 and 4.3 we report the iteration counts and the convergence factor for the C-AMG and SA-AMG methods when varying both the mesh size $h = 1/2, \ldots, 1/32$ and the polynomial approximation degree $p = 1, \ldots, 10$.

Results shown in Table 4.1 have been obtained based on employing the V(1,1)-cycle algorithm on both structured and unstructured grids. We have repeated the same set of experiments based on employing the W(1,1)-cycle and PCG V(2,2)-cycle algorithms; the results are reported in Tables 4.2 and 4.3, respectively.

From the results reported in Tables 4.1, 4.2 and 4.3 we can conclude that C-AMG and SA-AMG methods have the same behaviour of p-independence for $p = 1, \dots, 6$ on both structured and unstructured meshes, whereas for $p > 6$ the two methods show p-dependence. This can be expected because the constant defining the essentially positive matrices (Section 3.3) depends on p.

In particular we note that V(1,1)- and W(1,1)-cycles have the same convergence behaviour and that the best performance is obtained with the PCG V(2,2)-cycle because of both a bigger number of smoothing iterations and the acceleration by conjugate gradient method. Indeed if we use PCG V(2,2)-cycle we can have a better p-independence for $p = 1, \dots, 8$. These results are also shown in Figure 4.3, and are in agreement with some references in the literature.

[Cleary et al., 2000] show that the convergence factor for C-AMG applied to the Poisson problem discretization is much better on structured mesh than unstructured one when the number of grid points of the mesh becomes very large. Since the matrices used in our tests have a relatively small size in terms of the number of grid points, we agree with [Cleary et al., 2000] that the behaviour on the two meshes is similar. Other details on multigrid convergence for unstructured grids can be found in [Chan et al., 1998].

A comparison between C-AMG and SA-AMG is also reported in [Stüben, 2001]. In general C-AMG performs better than SA-AMG when the complexity of problem is high. Indeed when the problem complexity becomes higher SA-AMG may show h-dependence, in particular for the V-cycle. In our tests we can appreciate this behaviour in few cases of SA-AMG because of relatively small matrices complexity: one of these, where it is particularly evident, is the case of unstructured mesh and $p = 7$, as shown in Figure 4.4.

[Braess, 1995] shows that AMG is an uniform preconditioner, as expected from theoretical results: this occurred in all our tests.

Finally in Tables 4.4, 4.5 and 4.6 we report the iteration counts needed to achieve convergence and the convergence factor for the C-AMG and SA-AMG methods when varying the number of smoothing iterations $\nu = 1, 2, 3$. In Tables 4.7, 4.8 and 4.9 we report the same results varying the number of coarsening levels $K = 2, \dots, 5$.

From the results reported in Tables 4.4, 4.5 and 4.6 we can conclude that the performance of AMG methods improves for larger values of smoothing iterations.

From the results in Tables 4.7, 4.8 and 4.9 we observe that, as predicted from Theorems 3.1 and 3.4, C-AMG has a constant convergence factor independently of K, whereas SA-AMG seems to exhibit a mild pre-asymptotic dependence on K probably because of the small complexity of the problem and the reduced number of grid points of the meshes.

In conclusion our numerical results show that both C-AMG and SA-AMG are effective solvers for matrices stemming from the conforming discretizations of the Poisson problem at least for $p = 1, \dots, 6$. For higher values of p new interpolation formulas are mandatory to achieve full scalability.

Table 4.1: Iteration counts and convergence factor: V(1,1)-cycle, $K = 4$.

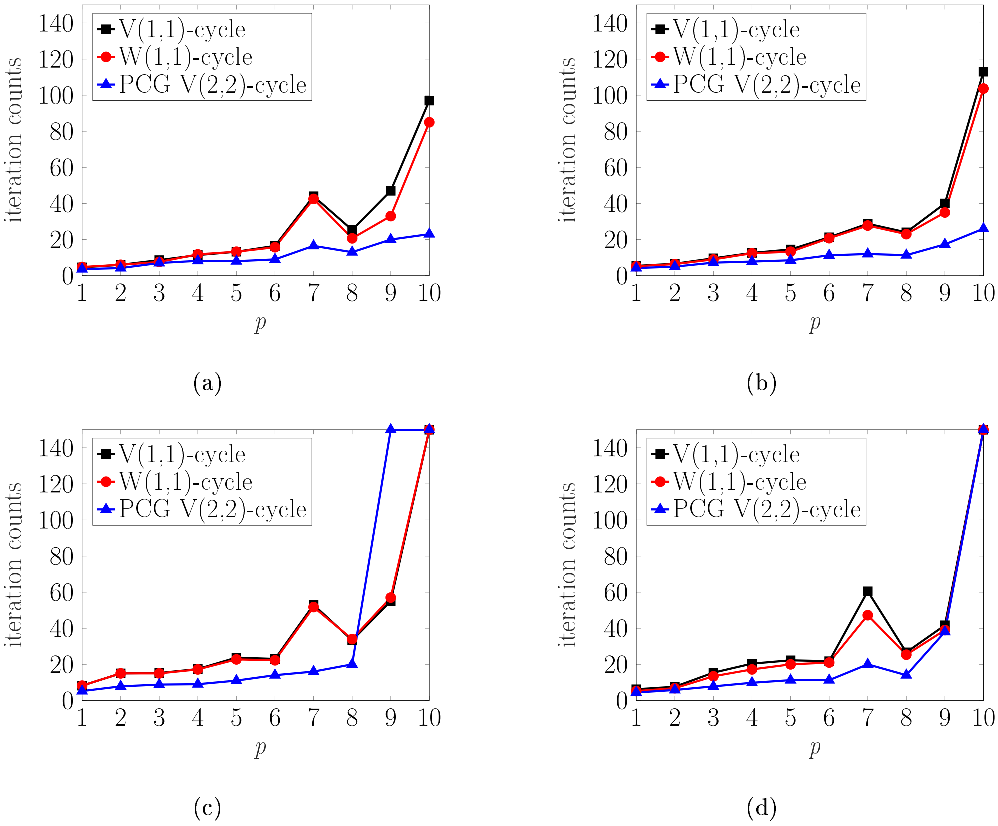
p	h	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
1	1/2	1	0	1	0	5	0.0236	6	0.0395
	1/4	4	0.0099	6	0.0325	5	0.0184	5	0.0228
	1/8	6	0.0371	10	0.1456	5	0.0243	6	0.0298
	1/16	6	0.0436	11	0.1871	6	0.0311	6	0.0331
	1/32	6	0.0439	13	0.2384	6	0.0330	8	0.0775
2	1/2	6	0.0250	6	0.0266	7	0.0538	6	0.0456
	1/4	6	0.0329	12	0.2041	6	0.0381	7	0.0479
	1/8	6	0.0347	19	0.3786	6	0.0425	6	0.0342
	1/16	6	0.0340	19	0.3770	7	0.0707	8	0.0851
	1/32	6	0.0334	19	0.3694	7	0.0538	11	0.1806
3	1/2	6	0.0341	10	0.1557	7	0.0609	13	0.2291
	1/4	10	0.1321	16	0.3007	9	0.1137	11	0.1728
	1/8	9	0.1214	17	0.3257	9	0.1083	10	0.1471
	1/16	9	0.1211	17	0.3232	10	0.1539	15	0.2928
	1/32	9	0.1207	16	0.3154	13	0.2375	28	0.5125
4	1/2	12	0.2049	16	0.3162	13	0.2241	16	0.3050
	1/4	12	0.2106	18	0.3474	10	0.1583	15	0.2851
	1/8	11	0.1793	18	0.3480	13	0.2225	16	0.2981
	1/16	11	0.1845	17	0.3344	13	0.2365	20	0.3921
	1/32	11	0.1820	18	0.3519	14	0.2634	35	0.5864
5	1/2	14	0.2579	20	0.3837	13	0.2207	22	0.4224
	1/4	13	0.2365	24	0.4546	14	0.2667	21	0.4158
	1/8	13	0.2393	25	0.4681	16	0.3125	22	0.4217
	1/16	13	0.2414	26	0.4905	14	0.2667	24	0.4589
6	1/2	16	0.3136	21	0.4119	21	0.4091	20	0.3907
	1/4	16	0.3152	22	0.4237	22	0.4282	18	0.3586
	1/8	17	0.3231	23	0.4385	21	0.4055	21	0.4134
	1/16	17	0.3224	26	0.4888	21	0.4055	28	0.5085
7	1/2	43	0.6511	48	0.6773	25	0.4702	52	0.7014
	1/4	44	0.6567	54	0.7080	27	0.5038	45	0.6610
	1/8	45	0.6626	54	0.7088	31	0.5459	49	0.6836
	1/16	44	0.6579	56	0.7192	32	0.5610	96	0.8252
8	1/2	26	0.4854	28	0.5170	27	0.5027	26	0.4894
	1/4	25	0.4721	36	0.5944	24	0.4634	24	0.4594
	1/8	25	0.4713	36	0.5969	21	0.4058	30	0.5358
9	1/2	48	0.6803	40	0.6296	36	0.5956	45	0.6614
	1/4	47	0.6734	61	0.7379	39	0.6225	42	0.6433
	1/8	46	0.6667	64	0.7489	45	0.6608	38	0.6155
10	1/2	71	0.7698	-	-	86	0.8064	-	-
	1/4	112	0.8480	-	-	117	0.8543	-	-
	1/8	108	0.8431	-	-	136	0.8732	-	-

Table 4.2: Iteration counts and convergence factor: W(1,1)-cycle, $K = 4$.

p	h	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
1	1/2	1	0	1	0	5	0.0236	6	0.0395
	1/4	5	0.0095	6	0.0325	5	0.0173	5	0.0135
	1/8	6	0.0375	10	0.1456	5	0.0230	5	0.0153
	1/16	6	0.0461	11	0.1872	5	0.0247	5	0.0200
	1/32	6	0.0461	13	0.2317	6	0.0304	5	0.0222
2	1/2	6	0.0304	6	0.0266	7	0.0562	6	0.0413
	1/4	6	0.0253	12	0.2044	6	0.0366	6	0.0376
	1/8	6	0.0269	19	0.3782	6	0.0447	6	0.0265
	1/16	6	0.0264	19	0.3757	7	0.0715	7	0.0529
	1/32	6	0.0258	19	0.3676	6	0.0456	9	0.1212
3	1/2	6	0.0344	10	0.1557	6	0.0452	13	0.2254
	1/4	8	0.0756	16	0.3004	8	0.0758	11	0.1735
	1/8	8	0.0806	17	0.3230	8	0.0782	10	0.1431
	1/16	8	0.0827	16	0.3162	10	0.1476	14	0.2469
	1/32	8	0.0838	16	0.3125	13	0.2299	19	0.3688
4	1/2	11	0.1854	16	0.3162	13	0.2223	16	0.3053
	1/4	12	0.2143	18	0.3492	10	0.1585	15	0.2835
	1/8	12	0.2097	18	0.3473	12	0.2116	15	0.2917
	1/16	12	0.2054	17	0.3375	13	0.2233	19	0.3758
	1/32	12	0.2006	17	0.3349	14	0.2524	21	0.4141
5	1/2	14	0.2598	20	0.3876	12	0.2029	20	0.3958
	1/4	13	0.2332	23	0.4463	13	0.2367	19	0.3758
	1/8	13	0.2372	24	0.4557	15	0.2785	19	0.3701
	1/16	13	0.2391	24	0.4600	13	0.2407	22	0.4261
6	1/2	15	0.2853	21	0.4127	21	0.4119	20	0.3903
	1/4	16	0.3045	22	0.4277	22	0.4229	18	0.3579
	1/8	16	0.3085	23	0.4365	20	0.3888	21	0.4101
	1/16	16	0.3090	23	0.4386	20	0.3870	25	0.4764
7	1/2	40	0.6292	47	0.6756	23	0.4403	53	0.7034
	1/4	44	0.6542	54	0.7082	27	0.4986	44	0.6564
	1/8	43	0.6516	53	0.7044	30	0.5397	41	0.6342
	1/16	43	0.6515	53	0.7051	31	0.5503	51	0.6956
8	1/2	21	0.4127	29	0.5214	26	0.4891	27	0.4976
	1/4	20	0.3966	36	0.5959	23	0.4431	24	0.4587
	1/8	21	0.4020	37	0.6038	20	0.3903	25	0.4683
9	1/2	36	0.5939	46	0.6668	32	0.5577	43	0.6503
	1/4	32	0.5592	61	0.7380	36	0.5966	40	0.6279
	1/8	31	0.5506	64	0.7481	37	0.6066	34	0.5813
10	1/2	57	0.7231	-	-	73	0.7759	-	-
	1/4	100	0.8315	-	-	108	0.8426	-	-
	1/8	98	0.8282	-	-	130	0.8677	-	-

Table 4.3: Iteration counts and convergence factor: PCG V(2,2)-cycle, $K = 4$.

p	h	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
1	1/2	1	0	1	0	3	0.0007	3	0.0005
	1/4	4	0.0046	4	0.0069	4	0.0029	4	0.0034
	1/8	4	0.0077	6	0.0320	4	0.0072	4	0.0047
	1/16	4	0.0096	7	0.0562	5	0.0120	5	0.0088
	1/32	5	0.0121	8	0.0716	5	0.0103	6	0.0230
2	1/2	4	0.0089	4	0.0118	5	0.0129	5	0.0146
	1/4	4	0.0083	8	0.0895	5	0.0143	5	0.0153
	1/8	4	0.0080	9	0.1055	5	0.0102	5	0.0105
	1/16	4	0.0087	9	0.1141	5	0.0172	6	0.0295
	1/32	5	0.0091	9	0.1149	5	0.0139	8	0.0620
3	1/2	6	0.0329	6	0.0447	6	0.0334	6	0.0370
	1/4	7	0.0594	8	0.0859	7	0.0505	6	0.0386
	1/8	7	0.0604	9	0.1210	7	0.0523	6	0.0406
	1/16	7	0.0605	10	0.1396	7	0.0594	9	0.0891
	1/32	8	0.0652	11	0.1492	9	0.0974	12	0.1608
4	1/2	9	0.1129	8	0.0962	6	0.0445	11	0.1843
	1/4	8	0.0901	9	0.1196	7	0.0551	8	0.0927
	1/8	8	0.0896	9	0.1215	9	0.0978	8	0.0904
	1/16	8	0.0880	9	0.1208	8	0.0917	10	0.1241
	1/32	8	0.0863	10	0.1355	9	0.1007	12	0.1871
5	1/2	8	0.0881	9	0.1332	8	0.0837	11	0.1767
	1/4	8	0.0831	10	0.1706	8	0.1011	11	0.1631
	1/8	8	0.0796	12	0.2090	9	0.1171	11	0.1675
	1/16	8	0.0788	13	0.2365	9	0.1164	12	0.1868
6	1/2	9	0.1367	16	0.3318	11	0.1812	11	0.1812
	1/4	9	0.1334	13	0.2486	11	0.1799	11	0.1751
	1/8	9	0.1309	13	0.2367	11	0.1828	10	0.1459
	1/16	9	0.1305	14	0.2529	12	0.1900	13	0.2094
7	1/2	19	0.3795	12	0.2407	11	0.1884	15	0.2993
	1/4	15	0.3029	15	0.3138	12	0.2084	16	0.3185
	1/8	16	0.3192	17	0.3481	12	0.2048	17	0.3325
	1/16	16	0.3273	20	0.3939	13	0.2295	32	0.5413
8	1/2	13	0.2449	19	0.3884	11	0.1848	13	0.2268
	1/4	13	0.2488	19	0.3977	12	0.1976	13	0.2408
	1/8	13	0.2522	22	0.4414	11	0.1721	16	0.2883
9	1/2	20	0.4143	-	-	14	0.2783	45	0.6792
	1/4	20	0.4012	-	-	19	0.3838	27	0.5180
	1/8	20	0.3984	-	-	19	0.3741	42	0.6547
10	1/2	17	0.3424	-	-	26	0.4927	-	-
	1/4	26	0.5017	-	-	25	0.4827	-	-
	1/8	26	0.4973	-	-	27	0.4940	-	-



(a)

(b)

(c)

(d)

Figure 4.3: Iteration counts as a function of p for (a) C-AMG: structured mesh, (b) C-AMG: unstructured mesh, (c) SA-AMG: structured mesh, (d) SA-AMG: unstructured mesh.

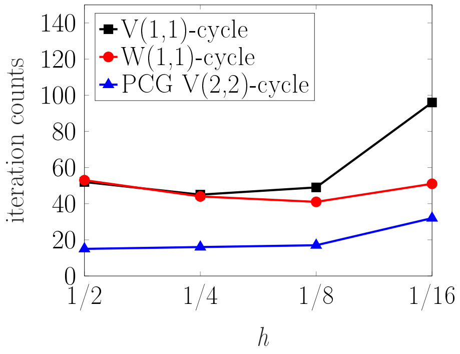


Figure 4.4: Iteration counts as a function of h for SA-AMG: unstructured mesh, $p = 7$.

Table 4.4: Iteration counts and convergence factor: V(ν,ν)-cycle, $K = 4$.

	ν	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	1	6	0.0439	13	0.2384	6	0.0330	8	0.0775
$h = 1/32$	2	5	0.0174	10	0.1488	5	0.0151	7	0.0519
	3	5	0.0122	9	0.1194	4	0.0093	6	0.0351
$p = 4$	1	11	0.1793	18	0.3480	13	0.2225	16	0.2981
$h = 1/8$	2	11	0.1786	12	0.2150	11	0.1789	12	0.2113
	3	10	0.1538	11	0.1782	10	0.1542	11	0.1848
$p = 10$	1	71	0.7698	-	-	86	0.8064	-	-
$h = 1/2$	2	42	0.6416	107	0.8414	59	0.7301	114	0.8502
	3	32	0.5616	73	0.7754	45	0.6610	78	0.7890

Table 4.5: Iteration counts and convergence factor: W(ν,ν)-cycle, $K = 4$.

	ν	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	1	6	0.0461	13	0.2317	6	0.0304	5	0.0222
$h = 1/32$	2	5	0.0106	10	0.1452	4	0.0097	4	0.0055
	3	4	0.0050	9	0.1272	4	0.0062	4	0.0037
$p = 4$	1	12	0.2097	18	0.3473	12	0.2116	15	0.2917
$h = 1/8$	2	10	0.1482	13	0.2264	11	0.1717	12	0.2007
	3	9	0.1263	11	0.1832	10	0.1457	11	0.1727
$p = 10$	1	57	0.7231	-	-	73	0.7759	-	-
$h = 1/2$	2	35	0.5897	107	0.8414	50	0.6917	114	0.8502
	3	31	0.5452	73	0.7754	39	0.6223	78	0.7890

Table 4.6: Iteration counts and convergence factor: PCG V(ν,ν)-cycle, $K = 4$.

	ν	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	1	8	0.1009	15	0.2737	8	0.0853	7	0.0586
$h = 1/32$	2	5	0.0121	8	0.0716	5	0.0103	6	0.0230
	3	4	0.0054	7	0.0534	4	0.0049	6	0.0148
$p = 4$	1	13	0.2563	14	0.2754	12	0.2107	11	0.1882
$h = 1/8$	2	8	0.0896	9	0.1215	9	0.0978	8	0.0904
	3	7	0.0576	8	0.0865	7	0.0635	8	0.0701
$p = 10$	1	51	0.7116	-	-	-	-	-	-
$h = 1/2$	2	17	0.3424	-	-	26	0.4927	-	-
	3	14	0.2596	28	0.5299	16	0.3084	30	0.5458

Table 4.7: Iteration counts and convergence factor: V(1,1)-cycle.

	K	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	2	6	0.0461	13	0.2317	6	0.0304	5	0.0130
$h = 1/32$	3	6	0.0444	13	0.2317	6	0.0271	6	0.0422
	4	6	0.0439	13	0.2384	6	0.0330	8	0.0775
	5	6	0.0434	13	0.2376	6	0.0373	7	0.0719
$p = 4$	2	12	0.2090	18	0.3473	12	0.2117	15	0.2916
$h = 1/8$	3	12	0.2127	17	0.3361	12	0.2115	15	0.2914
	4	11	0.1793	18	0.3480	13	0.2225	16	0.2981
	5	12	0.2104	18	0.3484	13	0.2272	16	0.2981
$p = 10$	2	54	0.7092	-	-	70	0.7683	-	-
$h = 1/2$	3	65	0.7516	-	-	88	0.8108	-	-
	4	71	0.7698	-	-	86	0.8064	-	-
	5	70	0.7678	-	-	85	0.8046	-	-

Table 4.8: Iteration counts and convergence factor: W(1,1)-cycle.

	K	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	2	6	0.0461	13	0.2317	6	0.0304	5	0.0130
$h = 1/32$	3	6	0.0461	13	0.2317	6	0.0304	5	0.0224
	4	6	0.0461	13	0.2317	6	0.0304	5	0.0222
	5	6	0.0461	13	0.2317	6	0.0304	5	0.0222
$p = 4$	2	12	0.2090	18	0.3473	12	0.2117	15	0.2916
$h = 1/8$	3	12	0.2097	18	0.3473	12	0.2116	15	0.2916
	4	12	0.2097	18	0.3473	12	0.2116	15	0.2917
	5	12	0.2097	18	0.3473	12	0.2116	15	0.2917
$p = 10$	2	54	0.7092	-	-	70	0.7683	-	-
$h = 1/2$	3	57	0.7229	-	-	73	0.7761	-	-
	4	57	0.7231	-	-	73	0.7759	-	-
	5	57	0.7231	-	-	73	0.7759	-	-

Table 4.9: Iteration counts and convergence factor: PCG V(2,2)-cycle.

	K	Structured Mesh				Unstructured Mesh			
		C-AMG		SA-AMG		C-AMG		SA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	2	4	0.0086	8	0.0697	4	0.0075	3	0.0009
$h = 1/32$	3	5	0.0111	8	0.0697	4	0.0088	6	0.0167
	4	5	0.0121	8	0.0716	5	0.0103	6	0.0230
	5	5	0.0121	8	0.0716	5	0.0109	6	0.0229
$p = 4$	2	8	0.0819	9	0.1224	9	0.0988	8	0.0897
$h = 1/8$	3	8	0.0880	9	0.1207	9	0.0968	8	0.0896
	4	8	0.0896	9	0.1215	9	0.0978	8	0.0904
	5	8	0.0909	9	0.1214	9	0.0980	8	0.0904
$p = 10$	2	15	0.3006	-	-	19	0.3861	-	-
$h = 1/2$	3	17	0.3365	-	-	25	0.4829	-	-
	4	17	0.3424	-	-	26	0.4927	-	-
	5	17	0.3426	-	-	27	0.5018	-	-

Chapter 5

Algebraic Multigrid for Discontinuous Finite Element Method

In this chapter we present algebraic multigrid methods for the efficient solution of the linear system of equations stemming from discontinuous Galerkin finite element approximations of the Poisson problem. In particular we discuss algorithms that extend those presented in Chapter 3. For discontinuous Galerkin methods standard multigrid techniques cannot be employed because of redundancy of the degrees of freedom associated to the same mesh nodes, therefore some multigrid principles described in Chapter 2 must be revised, in particular the coarsening strategies and the definition of the transfer operators.

Works by [Hemker et al., 2003, Hemker et al., 2004] show that Fourier error analysis on two-level grids is valid also for discontinuous Galerkin methods, consequently there are the premises for constructing multigrid methods. Guided by the geometric multigrid for discontinuous finite element setting [Brenner et al., 2009, Brenner et al., 2011, Antonietti et al., 2015, Antonietti et al., 2017], we propose algebraic multigrid methods based on classical and smoothed aggregation ones. Hybrid algebraic multigrid extend the method by [Chang et al., 1996], whereas the smoothed block aggregation improves the ones by [Vaněk et al., 1996, Olson and Schroder, 2011].

The main problem we have to deal with is the multiplicity of degrees of freedom associated to each grid points. A similar issue occurs for partial differential equations systems where there exists multiple unknowns at the same nodal location. This difficulty can be solved with strategies known as "point" or "block" approaches [Ruge and Stüben, 1987, Vaněk et al., 1996]: these techniques are based on local aggregation of variables associated to the same grid point. Moreover we have to build appropriately the interpolation operator taking into account that the matrices arising from discontinuous approximations cannot be approximated by filtered M-matrices.

The remaining part of this chapter is organized as follows: first we describe the local aggregation with the associated interpolation operator, then we discuss the hybrid and smoothed block aggregation algebraic multigrid algorithms.

5.1 Local Aggregation

Block aggregation is of crucial importance for algebraic multigrid for discontinuous Galerkin finite element methods. This is based on the access to mesh points, cf. [Ruge and Stüben, 1987,Vaněk et al., 1996,Olson and Schroder, 2011]
In this work we propose a new coarsening strategy purely algebraic 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 on the mesh, as described in the following.

5.1.1 Coarsening: Algebraic Block Aggregation

Given the matrix $A_h \in \mathbb{R}^{N_h \times N_h}$ and its adjacency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, we split the set of points in a disjoint covering such that $\mathcal{V} = \bigcup_{j=1}^{N_H} \mathcal{V}_j$, $N_H \leq N_h$, and $\mathcal{V}_l \cap \mathcal{V}_j = \emptyset$ for $l \neq j$. In particular, given the adjacency graph $\mathcal{G}_{S_+} = (\mathcal{V}_{S_+}, \mathcal{E}_{S_+})$ associated to a generic strongest connection matrix, the algorithm aims at providing suitable sets such that each one of them contains the multiple variables associated to a grid point, cf. Algorithm 5.1.
Algorithm 5.1 is made of three steps: startup singleton or aggregation, enlargement of the decomposition sets, and deleting 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 point i has the strongest connection. If the strongest connection between i and I is negative, 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.

5.1.2 Interpolation Operator

Given the disjoint partition $\mathcal{V} = \bigcup_{j=1}^{N_H} \mathcal{V}_j$, $N_H \leq N_h$ given by Algorithm 5.1, it is natural to construct the interpolation operator in a similar manner as done for the smoothed aggregation algebraic multigrid by [Vaněk et al., 1996].
In particular, we modify the algorithm of [Vaněk et al., 1996] as follows. We define algebraically smooth error modes to be grid functions with a small Rayleigh quotient, cf. [McCormick and Ruge, 1982] and therefore equivalent to the near null-space or low energy modes. Hence the tentative interpolation operator is constructed such that it preserves the near null-space mode vector $\mathbf{w} \in \mathbb{R}^{N_h}$, cf. [Brezina et al.,

Algorithm 5.1 Block Aggregation Scheme

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

2005, Brannick et al., 2006, Olson et al., 2011]. The vector $\mathbf{w}$ is the numerical solution of $A\mathbf{w} = \mathbf{0}$ obtained after η smoothing steps with initial guess $\mathbf{w}^0 = \mathbf{1}$. Therefore the operator is defined as

$$[\tilde{I}_H^h]_{ij} = \begin{cases} w_i & i \in \mathcal{V}_j \\ 0 & otherwise \end{cases}, \quad i = 1, .., N_h,\ j = 1, .., N_H.$$

and finally the interpolation is improved by a simple damped Jacobi smoothing to get

$$I_H^h = (I_h - \omega D_h^{-1} A_h)\tilde{I}_H^h, \tag{5.1}$$

where $\omega = 2/3$, D_h is the diagonal of A_h and I_h is the identity matrix. As it was done for smoothed aggregation algebraic multigrid in Section 3.2.2, we apply the Gram-Schmidt orthonormalization algorithm to each column of $\tilde{I}_H^h$ to improve conditioning.

5.2 Hybrid Algebraic Multigrid

In this section we present our new method: we join together the algebraic block aggregation with the extension to general matrices of classical algebraic multigrid presented in [Chang et al., 1992, Chang et al., 1996]. This last algorithm modifies the strength function and the interpolation operator of classical algebraic multigrid methods for M-matrices. Indeed if we employ the method presented in Section 3.1 for general matrices, the setup phase may fail. Therefore instead of using the strength function defined in (3.3), we assume it to be

$$s(i,j) = \frac{|a_{ij}|}{\max_{k\neq i}|a_{ik}|}. \tag{5.2}$$

The first step of the hybrid algebraic multigrid algorithm is to apply the algebraic block aggregation, cf. Algorithm 5.1, therefore we detail how the function find_strongest_connection(i) works.
Guided by the strength function definition in (5.2), we assume that the strongest connected points to i are given by

$$\mathcal{S}_i^+ = \{j : |a_{ij}| = \max_{k\neq i}|a_{ik}|\}. \tag{5.3}$$

Therefore the function find_strongest_connection(i) returns one of the point in the set $\mathcal{S}_i^+$.

5.2.1 Coarsening

Given the strength function in (5.2), we construct the strength matrix $S \in \mathbb{R}^{N_h\times N_h}$ with entries

$$s_{ij} = \begin{cases} 1 & |a_{ij}| \geq \theta \max_{k\neq i}|a_{ik}| \\ 0 & otherwise \end{cases}$$

and its adjacency graph $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, $\mathcal{G}_S \subset \mathcal{G}$. Given the strongest connection definition in (5.3), we build the matrix $S_+ \in \mathbb{R}^{N_h\times N_h}$ with entries

$$s_{ij}^+ = \begin{cases} 1 & |a_{ij}| = \max_{k\neq i}|a_{ik}| \\ 0 & otherwise \end{cases}$$

and its adjacency graph $\mathcal{G}_{S_+} = (\mathcal{V}_{S_+}, \mathcal{E}_{S_+})$, $\mathcal{G}_{S_+} \subset \mathcal{G}_S$.

Then we apply the coarsening strategy: block aggregation on $\mathcal{G}_{S_+}$, cf. Algorithm 5.1 and colouring scheme on $\mathcal{G}_S$, cf. Algorithms 3.1, 3.2 and 3.3.

In Figures 5.1 and 5.2 we show some examples of block aggregation with the definition of strongest connection given in (5.3) for matrices stemming from discontinuous Galerkin discretizations on structured and unstructured simplicial meshes. Each aggregate set is represented with distinct colour and mark.

In order to do this we have to fix the penalty parameter σ^e, e.g. $\sigma^e = 5, 10, 20, 30$, cf. Section 1.2.2.

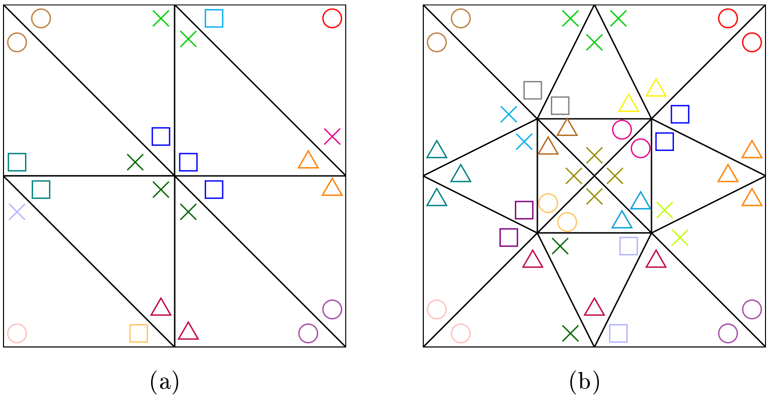


Figure 5.1: Examples of block aggregation for (a) structured and (b) unstructured meshes with $p = 1$, $h = 1/2$ and $\sigma^e = 5$.

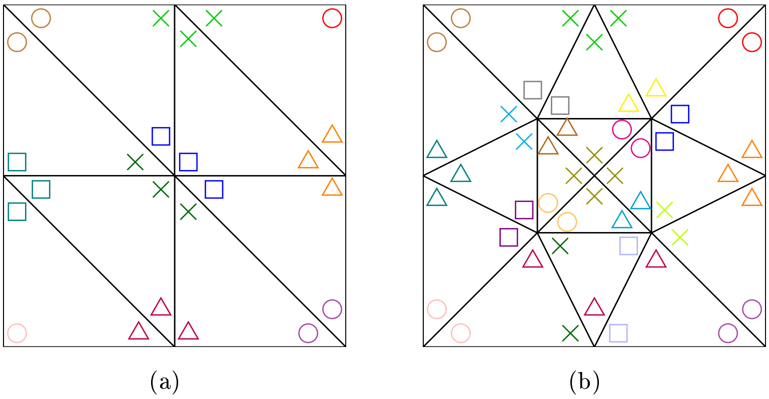


Figure 5.2: Examples of block aggregation for (a) structured and (b) unstructured meshes with $p = 1$, $h = 1/2$ and $\sigma^e = 10, 20, 30$.

We note that for σ^e big enough the block aggregation scheme produces the same disjoint covering of degrees of freedom.

5.2.2 Interpretation of Algebraic Smooth Error

Recalling from Sections 2.2.2 and 3.1.2, given a C/F-splitting we divide the neighborhood into three categories as proposed in Definition 3.5 and from the definition of an algebraically smooth error, i.e. $A_h \mathbf{e}_h \approx \mathbf{0}_h$, we get

$$a_{ii} e_i^h \approx -\sum_{j \in C_i} a_{ij} e_j^h - \sum_{j \in D_i^s} a_{ij} e_j^h - \sum_{j \in D_i^w} a_{ij} e_j^h \quad i = 1, .., N_h. \tag{5.4}$$

Now we introduce a geometric assumption that will guide us in defining the interpolation operator coefficients, cf. [Chang et al., 1992, Chang et al., 1996]. We assume that the algebraic error is smooth between points i and j if $a_{ij} < 0$ or $|a_{ij|}$ is small, and it is oscillating if $a_{ij} > 0$ is large. With this idea, we have to define measures to understand the nature of the error in order to decide how to interpolate it.
Let the following quantities be measures of nature of the algebraic error between points i and j:

$$\xi_{ij} = -\frac{\sum_{k \in C_i} a_{jk}}{\sum_{k \in C_i} |a_{jk}|}$$

and

$$\eta_{ij} = \frac{|a_{ji}| l_{ij}}{\sum_{k \in C_i} |a_{jk}|},$$

where l_{ij} is the cardinality of the set

$$L_{ij} = \{k : k \in C_i, a_{jk} \neq 0\}.$$

The quantity ξ_{ij} indicates whether there exists a large positive connection a_{jk} between j and $k \in L_{ij}$ and by the geometric assumption it follows that the error between points i and j is smooth if $\xi_{ij} \geq 0.5$ and $a_{ij} < 0$. The quantity η_{ij} approximately gives the ratio of the strength between points j and i to the average strength between j and elements in the set L_{ij}, therefore it describes how much these points are strongly connected.

5.2.3 Interpolation Operator

In this section we define the interpolation operator taking into account the considerations about the algebraic smooth or oscillating error made in the previous one.
As in Section 3.1.3 we assume that, given $\mathbf{e}_H \in \mathbb{R}^{N_H}$, the interpolation operator is constructed as follows

$$[\mathbf{e}_h]_i = [I_H^h \mathbf{e}_H]_i = \begin{cases} e_i^H & i \in C \\ \sum_{k \in C_i} w_{ik} e_k^H & i \in F \end{cases}, \quad i = 1, .., N_h,$$

where the interpolation weights w_{ik} have to be determined.

Reasoning as in Section 5.2.2, we replace the e_j^h in equation (5.4) by its approximation in terms of e_i^h or e_k^H, with $k \in C_i$. The substitution in the summation on C_i is made relabelling e_j^h in e_k^H. Instead, for summations on D_i^s and D_i^w we define

$$g_{jk} = \frac{|a_{jk}|}{\sum_{k \in C_i} |a_{jk}|}, \quad j \in D_i^s \cup D_i^w,\ k \in C_i$$

in order to preserve constant vectors and we make the following approximations. For points $j \in D_i^w$ we have

$$e_j^h = \begin{cases} e_i^h & l_{ij} = 0,\ a_{ij} < 0 \\ -e_i^h & l_{ij} = 0,\ a_{ij} > 0 \\ 2\sum_{k \in C_i} g_{jk} e_k^H - e_i^h & l_{ij} > 0,\ \xi_{ij} \geq 0.5,\ a_{ij} < 0 \\ \sum_{k \in C_i} g_{jk} e_k^H & \textit{otherwise} \end{cases} \tag{5.5}$$

and for points $j \in D_i^s$ we propose

$$e_j^h = \begin{cases} 2\sum_{k \in C_i} g_{jk} e_k^H - e_i^h & \eta_{ij} < 0.75,\ \xi_{ij} \geq 0.5,\ a_{ij} < 0 \\ \frac{1}{2}\left(\sum_{k \in C_i} g_{jk} e_k^H + e_i^h\right) & \eta_{ij} > 2,\ \xi_{ij} \geq 0.5,\ a_{ij} < 0 \\ \sum_{k \in C_i} g_{jk} e_k^H & \textit{otherwise} \end{cases} . \tag{5.6}$$

Substituting the equations (5.5) and (5.6) into (5.4) and defining the following sets

$$G_i^1 = \{j \in D_i^w : l_{ij} = 0\},$$

$$\begin{aligned} G_i^2 =& \{j \in D_i^w : l_{ij} > 0,\ \xi_{ij} \geq 0.5,\ a_{ij} < 0\} \\ & \cup\ \{j \in D_i^s : \eta_{ij} < 0.75,\ \xi_{ij} \geq 0.5,\ a_{ij} < 0\}, \\ G_i^3 =& \{j \in D_i^s : \eta_{ij} > 2,\ \xi_{ij} \geq 0.5,\ a_{ij} < 0\}, \\ G_i^4 =& \{j \in D_i^w : j \notin G_i^1, j \notin G_i^2, j \notin G_i^3\}, \end{aligned}$$

we get the interpolations weights defined as

$$w_{ik} = -\frac{\bar{a}_{ik}}{\bar{a}_{ii}}.$$

with

$$\bar{a}_{ik} = a_{ik} + 2\sum_{j \in G_i^2} a_{ij} g_{jk} + \frac{1}{2}\sum_{j \in G_i^3} a_{ij} g_{jk} + \sum_{j \in G_i^4} a_{ij} g_{jk},$$

$$\bar{a}_{ii} = a_{ii} - \sum_{j \in G_i^1} |a_{ij}| - \sum_{j \in G_i^2} a_{ij} + \frac{1}{2}\sum_{j \in G_i^3} a_{ij}.$$

Note that the construction of the interpolation operator makes sense for matrices stemming from discontinuous Galerkin approximations of order $p = 1, 2$ because for $p \geq 3$ the block aggregation scheme, cf. Algorithm 5.1 tends to aggregate the "bubbles" and this is not desirable for the given interpolation formula. Moreover this interpolation operator is constructed assuming a linear conforming finite element discretization and a certain regularity of the mesh, cf. [Chang et al., 1992, Chang et al., 1996] so we expect that it can fail in the case of higher order discretizations or non-regular meshes.

5.3 Smoothed Block Aggregation Algebraic Multigrid

In this section we present a new algebraic multigrid method for matrices arising from high order discontinuous Galerkin finite element methods. The method is smoothed aggregation-based [Vaněk et al., 1996] and it takes account of its improvements for DG problems made by [Olson and Schroder, 2011].

5.3.1 Evolution Measure

The strength function of standard smoothed aggregation multigrid, cf. equation (3.7) is motivated by the assumption that the matrix is stemming from conforming finite element methods. Therefore it does not hold in this framework and we need to define a new strength measure.

We present the evolution measure proposed by [Olson et al., 2010] which combines the local knowledge of both algebraic smooth error and the behaviour of the interpolation.

In order to take account for algebraic smooth error, we define the z function, according to weighted Jacobi, as

$$z = (I_h - \omega D_h^{-1} A_h)^m \delta_i$$

where δ_i is the δ-function centered at i. Typical values are $\omega = 1/\rho(D_h^{-1} A_h)$ and $m = 2$.

Then we have to consider the local knowledge of the interpolation. Assume that the interpolation operator is defined as in Section 5.1.2. Given a point $i \in \mathcal{V}$ we would like to be able to measure the ability of each column of $\tilde{I}_H^h$ to interpolate z for all points j in the algebraic neighborhood of i, i.e. $j \in N_i$. Therefore this quantity is measured only for points $j \in N_i$, in particular with exact interpolation enforced at point i.

Thanks to all these considerations, the evolution measure reads as follows

$$e(i,j) = \left| 1 - \frac{b_j z_i}{b_i z_j} \right|$$

and its symmetrized version is

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

Finally the symmetric evolution strength function is defined as

$$s(i,j) = \frac{e_S(i,j)}{\min_{k\neq i} e_S(i,k)}. \tag{5.7}$$

According to this new strength function, since smaller values indicate a stronger connection, the Definition 2.12 must be changed as follows.

Definition 5.1. *Given a threshold* $2 \leq \theta \leq 4$, *we say that the vertices* i *and* j *are strongly connected if*

$$s(i,j) \leq \theta.$$

In order to apply the algebraic block aggregation, cf. Algorithm 5.1, we have to explain how the function find_strongest_connection(i) works.

Guided by the strength function definition in (5.7), we assume that the strongest connected points to i are given by

$$\mathcal{S}_i^+ = \{j : e_S(i,j) = \min_{k\neq i} e_S(i,k)\}. \tag{5.8}$$

Therefore the function find_strongest_connection(i) returns one of the point in the set $\mathcal{S}_i^+$.

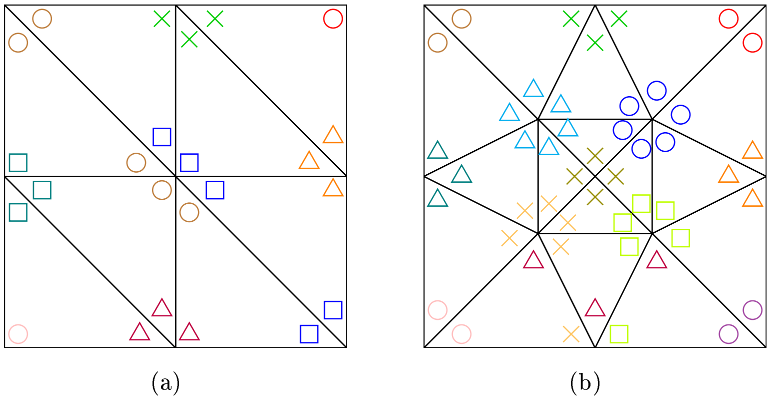
5.3.2 Coarsening and Interpolation Operator

In this section we discuss the coarsening strategy and the definition of the interpolation operator for our smoothed block aggregation algebraic multigrid taking into account all the tools introduced so far.

As first level coarsening we use the local aggregation algorithm with strongest evolution connection, cf. Algorithm 5.1, then we have to treat all other levels.

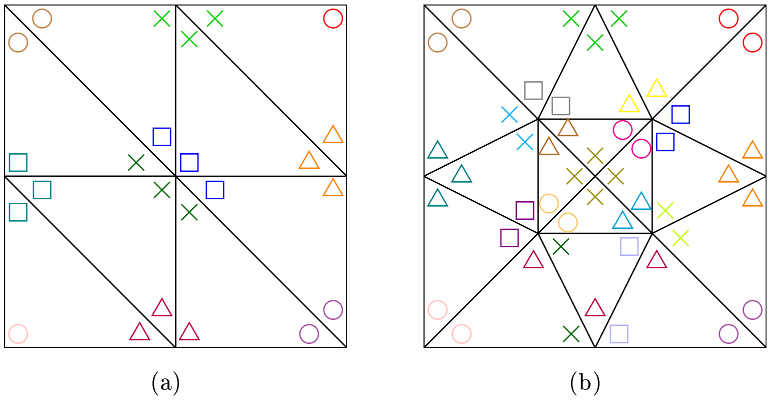
In Figures 5.3, 5.4 and 5.5 we show some examples of block aggregation with the definition of strongest connection in (5.8) for matrices stemming from discontinuous Galerkin discretizations on structured and unstructured simplicial meshes. Each aggregate set is represented with distinct colour and mark. In order to do this we have to fix the penalty parameter σ^e, e.g. $\sigma^e = 5, 10, 20, 30$, cf. Section 1.2.2.

We note that for σ^e big enough the block aggregation scheme produces the same disjoint covering of degrees of freedom.



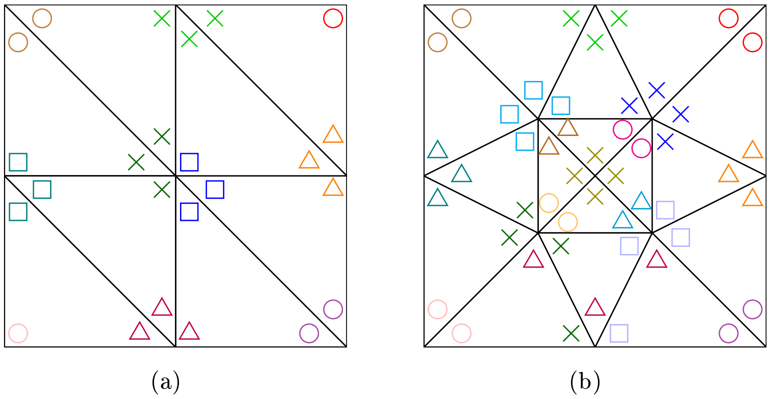
(a) (b)

Figure 5.3: Examples of block aggregation for (a) structured and (b) unstructured meshes with $p = 1$, $h = 1/2$ and $\sigma^e = 5$.



(a) (b)

Figure 5.4: Examples of block aggregation for (a) structured and (b) unstructured meshes with $p = 1$, $h = 1/2$ and $\sigma^e = 10$.



(a) (b)

Figure 5.5: Examples of block aggregation for (a) structured and (b) unstructured meshes with $p = 1$, $h = 1/2$ and $\sigma^e = 20, 30$.

Returning to our method, since it is aggregation based, we have two choice of aggregation algorithms for coarser levels: we can use the standard smoothing aggregation one with evolution measure, cf. Algorithm 3.4, or we can extend the use of local aggregation algorithm with strongest evolution connection, cf. Algorithm 5.1.

In order to perform these proposed methods, we have to build the strength and the strongest connection matrices and their associated adjacency graphs. Given the strength function in (5.7) and guided by Definition 5.1, we construct the strength matrix $S \in \mathbb{R}^{N_h \times N_h}$ with entries

$$s_{ij} = \begin{cases} 1 & e_S(i,j) \leq \theta \min_{k \neq i} e_S(i,k) \\ 0 & otherwise \end{cases}$$

and its adjacency graph $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, $\mathcal{G}_S \subset \mathcal{G}$. Given the strongest connection definition in (5.8), we build the matrix $S_+ \in \mathbb{R}^{N_h \times N_h}$ with entries

$$s^+_{ij} = \begin{cases} 1 & e_S(i,j) = \min_{k \neq i} e_S(i,k) \\ 0 & otherwise \end{cases}$$

and its adjacency graph $\mathcal{G}_{S_+} = (\mathcal{V}_{S_+}, \mathcal{E}_{S_+})$, $\mathcal{G}_{S_+} \subset \mathcal{G}_S$.

The main difference between the two proposed coarsening strategies is the type of aggregation: aggressive or non-aggressive. In some cases it may be necessary to require a non-aggressive aggregation because otherwise the aggregation fails in capturing all grid information and therefore the method does not reduce the residual.

In both cases the associated interpolation operator is the one proposed in Section 5.1.2 to all multigrid levels with $\eta = p$, where p is the polynomial degree.

Numerical results reported in Chapter 6 have been obtained with the aggressive strategy: block aggregation for finest matrix, cf. Algorithm 5.1 and standard aggregation for coarser matrices, cf. Algorithm 3.4.

Chapter 6

Numerical Experiments: Algebraic Multigrid for Discontinuous Finite Element Method

In this chapter we present some numerical results to investigate the efficiency and robustness of our algebraic multigrid methods in solving the linear system of equations stemming from discontinuous finite element discretizations of the Poisson problem. The test problem and the meshes are as those considered in Chapter 4. In order to discretize the Poisson problem with discontinuous finite element methods we set $\sigma^e = 10$, cf. Section 1.2.2.

6.1 Setup Phase

Given the matrix $A_h \in \mathbb{R}^{N_h \times N_h}$, (also denoted A_1 in multilevel setting), we fix the number K of desired coarser "grids", then we build the coarser matrices and the interpolation operators with hybrid or smoothed block aggregation setup. In our tests we have $K = 4$, but in order to show numerical results varying the number of coarser matrices we also study K varies from 2 to 5 for some fixed h and p associated with the finite element discretizations.

Since the hybrid and the smoothed block aggregation AMG have different setup phases we describe them in two sections. In our numerical results we denote with H-AMG the hybrid algebraic multigrid, whereas we use SBA-AMG to refer to smoothed block aggregation one.

6.1.1 Setup Phase of Hybrid Algebraic Multigrid

We remark that the hybrid AMG method cannot be employed for the polynomial approximation degree $p \geq 3$. For this method we construct the coarser matrices and interpolation operators as detailed in Section 5.2. If we denote by k the current "grid" level, the hybrid AMG method can be summarized as in Table 6.1.

Table 6.1: Setup Phase of Hybrid Algebraic Multigrid

level k	**Coarsening Strategy**	**Interpolation Operator** I_{k+1}^{k}
1	Block Aggregation	Smoothing Step Based
$2,\dots,K$	C/F-splitting	Smooth Error Based

If $k = 1$ we apply the block aggregation scheme, cf. Algorithm 5.1, to the matrix $A_1 = A_h$ with the strongest connection function defined in Section 5.2 and the interpolation formula in Section 5.1.2 with $\eta = p$, otherwise if $k = 2,\dots,K$ we build the coarser matrices with the colouring scheme with $\theta = 0.25$ as described in Section 3.1.2 and the interpolation operator is constructed with instructions given in Section 5.2.3.

6.1.2 Setup Phase of Smoothed Block Aggregation Algebraic Multigrid

For smoothed block aggregation AMG method we construct the coarser matrices and interpolation operators as detailed in Section 5.3. The SBA-AMG method can be summarized as in Table 6.2.

Table 6.2: Setup Phase of Smoothed Block Aggregation Algebraic Multigrid

level k	**Coarsening Strategy**	**Interpolation Operator** I_{k+1}^{k}
1	Block Aggregation	Smoothing Step Based
$2,\dots,K$	Standard Aggregation	Smoothing Step Based

If $k = 1$ we apply the block aggregation scheme, cf. Algorithm 5.1, to the matrix $A_1 = A_h$ with the evolution measure as strongest connection function defined in Section 5.3.1 with parameter $m = 4$, cf. [Olson and Schroder, 2011] and the interpolation formula in Section 5.1.2 with $\eta = p$, otherwise if $k = 2,\dots,K$ we build the coarser matrices with the aggregation scheme with $\theta = 2$ as described in Section 5.3. For smoothed block aggregation AMG we make an additional assumption about the smoothing step of interpolation operator, cf. Section 5.3. Given the tentative interpolation operator $\tilde{I}_H^h$, we assume the final one to be

$$I_H^h = \begin{cases} (I_h - \omega D_h^{-1} A_h)\tilde{I}_H^h & p \le 7 \\ (I_h - \omega D_h^{-1} A_h^M)\tilde{I}_H^h & p > 7 \end{cases}, \tag{6.1}$$

where A_h^M is the M-filtered matrix of A_h. This choice is done under the hypothesis that the most important connections have negative entries in the matrix. The choice of the new smoothing step of the interpolation operator is made on the basis of some numerical experiments: if $p > 7$ and we use (5.1) as smoothing step then the solution phase could be not convergent at all because the matrix conditioning deteriorates as p increase, therefore we try to improve it using (6.1).

6.2 Solution Phase

The solution phase associated with algebraic multigrid methods for matrices stemming from discontinuous discretizations of the Poisson problem is the same of Section 4.2, except for the PCG V(2,2)-cycle which is substitute by the PCG W(2,2)-cycle because in discontinuous finite element framework the W-cycle seems to perform better, cf. [Brenner et al., 2009, Brenner et al., 2011, Antonietti et al., 2015].

6.3 Numerical Results

In this section we compute the convergence factor ρ as done in Chapter 4. In the following "*" means that the method cannot be employed.
In order to underline the importance of aggregating the multiple degrees of freedom associated with a grid point in discontinuous Galerkin discretization framework we show an example of the use of H-AMG and SBA-AMG without the block aggregation scheme, cf. Algorithm 5.1 applied to matrices stemming from discontinuous discretizations with $p = 1, 2$ and $h = 1/32$ with the use of W(1,1)-cycle on both structured and unstructured meshes, cf. Table 6.3.
We specify that SBA-AMG seems to have a good performance also without the block aggregation scheme, but we point out that the Algorithm 5.1 is important for higher values of p.

Table 6.3: Iteration counts and convergence factor: W(1,1)-cycle, $K = 4$, no Block Aggregation Scheme

	Structured Mesh				Unstructured Mesh			
	H-AMG		SBA-AMG		H-AMG		SBA-AMG	
	N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$, $h = 1/32$	56	0.7183	11	0.1707	45	0.6626	12	0.2062
$p = 2$, $h = 1/32$	78	0.7880	8	0.0822	-	-	8	0.0852

In Tables 6.4, 6.5 and 6.6 we report the iteration counts needed to achieve convergence and the convergence factor for the H-AMG and SBA-AMG methods when varying both the mesh size $h = 1/2, \dots, 1/32$ and the polynomial approximation degree $p = 1, \dots, 10$.
Results shown in Table 6.4 have been obtained based on employing the V(1,1)-cycle algorithm on both structured and unstructured grids. We have repeated the same

set of experiments based on employing the W(1,1)-cycle and PCG W(2,2)-cycle algorithms; the results are reported in Tables 6.5 and 6.6, respectively.

From the results reported in Tables 6.4, 6.5 and 6.6 we can conclude that H-AMG is a method which does not have good results except for the W(1,1)- and PCG W(2,2)-cycles for structured mesh. This is because the block aggregation scheme does not provide a perfect description of the geometric aggregation of degrees of freedom which is required by the interpolation formula in Section 5.2.3. Despite the results obtained are not satisfactory, we note that the block aggregation scheme greatly improves the convergence of the method that does not take it into account, cf. Table 6.3. We report this information in Figure 6.1 and show the h-dependence of the H-AMG method.

Instead the SBA-AMG method has an optimal performance of h- and p-independence for $p = 1, \ldots, 7$ on both structured and unstructured meshes for all tests that have been done, whereas for $p > 7$ the method show h- and p-dependence: we note that our additional assumption in (6.1) is not sufficient to have h- and p-independence for V(1,1)- and W(1,1)-cycle, but it is shown that the PCG W(2,2)-cycle performs well.

In particular we note if we use PCG W(2,2)-cycle we can have a quasi optimal h- and p-independence for all $h = 1/2, \ldots, 1/32$ and $p = 1, \ldots, 10$ thanks to the acceleration by conjugate gradient method and a bigger number of smoothing iterations. In Figures 6.2 and 6.3 we report the same results obtained with SBA-AMG.

We point out that these results are in agreement with analogous ones reported in the literature, cf. [Olson and Schroder, 2011].

In Tables 6.7, 6.8 and 6.9 we report the iteration counts and the convergence factor for the H-AMG and SBA-AMG methods when varying the number of smoothing iterations $\nu = 1, 2, 3$. In Tables 6.10, 6.11 and 6.12 we report the results obtained when varying the number of coarsening levels $K = 2, \ldots, 5$.

From the results reported in Tables 6.7, 6.8 and 6.9 we can conclude that, as expected, for discontinuous Galerkin finite element methods AMG performs better for larger number of smoothing iterations.

From the results of Tables 6.10, 6.11 and 6.12 we can conclude that H-AMG seems not to be uniformly convergent independently of the number of levels K, whereas SBA-AMG shows K-independence

Finally, we observe that SBA-AMG seems to be a good solver for matrices stemming from the discontinuous discretization of the Poisson problem for $p = 1, \ldots, 7$. For higher values of p is better to investigate the energy minimization algebraic multigrid as done by [Olson and Schroder, 2011, Olson et al., 2011], where algebraic multigrid is coupled with geometric block aggregation on finest matrix and a standard algebraic aggregation with the evolution measure on coarser ones.

Table 6.4: Iteration counts and convergence factor: V(1,1)-cycle, $K = 4$.

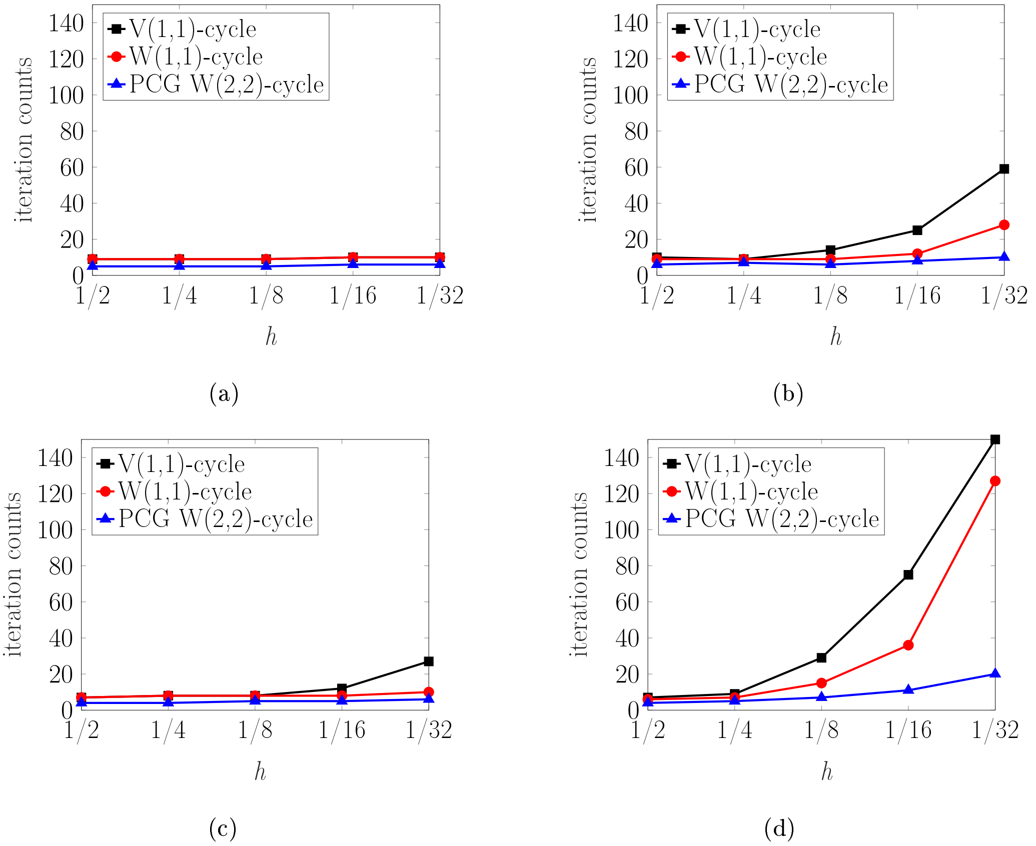
p	h	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
1	1/2	9	0.1088	8	0.0988	10	0.1342	9	0.1173
	1/4	9	0.1198	9	0.1190	9	0.1167	9	0.1187
	1/8	9	0.1290	9	0.1203	14	0.2551	10	0.1560
	1/16	10	0.1354	10	0.1523	25	0.4736	9	0.1209
	1/32	10	0.1355	10	0.1457	59	0.7295	10	0.1502
2	1/2	7	0.0690	7	0.0642	7	0.0559	6	0.0396
	1/4	8	0.0871	8	0.0849	9	0.1199	8	0.0969
	1/8	8	0.0949	8	0.0925	29	0.5245	9	0.1249
	1/16	12	0.2026	9	0.1116	75	0.7812	10	0.1472
	1/32	27	0.5006	9	0.1144	-	-	11	0.1717
3	1/2	*	*	6	0.0461	*	*	7	0.0544
	1/4	*	*	7	0.0671	*	*	8	0.0818
	1/8	*	*	7	0.0680	*	*	10	0.1554
	1/16	*	*	7	0.0701	*	*	10	0.1530
	1/32	*	*	7	0.0708	*	*	10	0.1528
4	1/2	*	*	6	0.0392	*	*	6	0.0448
	1/4	*	*	8	0.0912	*	*	10	0.1475
	1/8	*	*	8	0.0969	*	*	10	0.1512
	1/16	*	*	8	0.0970	*	*	11	0.1863
	1/32	*	*	8	0.0966	*	*	11	0.1796
5	1/2	*	*	12	0.2040	*	*	15	0.2901
	1/4	*	*	16	0.3043	*	*	18	0.3515
	1/8	*	*	16	0.3105	*	*	19	0.3701
	1/16	*	*	17	0.3284	*	*	19	0.3712
6	1/2	*	*	20	0.3933	*	*	14	0.2619
	1/4	*	*	20	0.3902	*	*	13	0.2367
	1/8	*	*	19	0.3736	*	*	16	0.3080
	1/16	*	*	19	0.3750	*	*	12	0.2080
7	1/2	*	*	16	0.3020	*	*	18	0.3478
	1/4	*	*	20	0.3965	*	*	21	0.4085
	1/8	*	*	24	0.4571	*	*	18	0.3571
	1/16	*	*	27	0.5044	*	*	19	0.3604
8	1/2	*	*	18	0.3548	*	*	37	0.6060
	1/4	*	*	32	0.5580	*	*	38	0.6157
	1/8	*	*	43	0.6499	*	*	44	0.6558
9	1/2	*	*	33	0.5694	*	*	30	0.5385
	1/4	*	*	35	0.5896	*	*	40	0.6291
	1/8	*	*	51	0.6964	*	*	53	0.6983
10	1/2	*	*	35	0.5900	*	*	40	0.6301
	1/4	*	*	41	0.6366	*	*	43	0.6501
	1/8	*	*	57	0.7238	*	*	61	0.7429

Table 6.5: Iteration counts and convergence factor: W(1,1)-cycle, $K = 4$.

p	h	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
1	1/2	9	0.1084	9	0.1087	9	0.1041	10	0.1340
	1/4	9	0.1184	9	0.1186	9	0.1073	9	0.1193
	1/8	9	0.1226	9	0.1239	9	0.1121	9	0.1139
	1/16	10	0.1357	10	0.1368	12	0.2003	9	0.1161
	1/32	10	0.1389	10	0.1397	28	0.5131	9	0.1197
2	1/2	7	0.0608	7	0.0656	6	0.0394	6	0.0393
	1/4	8	0.0831	8	0.0830	7	0.0695	7	0.0627
	1/8	8	0.0837	8	0.0837	15	0.2754	8	0.0783
	1/16	8	0.0852	8	0.0849	36	0.5975	8	0.0830
	1/32	10	0.1492	8	0.0856	127	0.8648	8	0.0855
3	1/2	*	*	6	0.0345	*	*	6	0.0452
	1/4	*	*	7	0.0609	*	*	7	0.0680
	1/8	*	*	7	0.0633	*	*	9	0.1116
	1/16	*	*	7	0.0648	*	*	8	0.0995
	1/32	*	*	7	0.0645	*	*	9	0.1111
4	1/2	*	*	6	0.0348	*	*	6	0.0394
	1/4	*	*	7	0.0647	*	*	7	0.0532
	1/8	*	*	7	0.0693	*	*	7	0.0617
	1/16	*	*	7	0.0715	*	*	7	0.0639
	1/32	*	*	8	0.0795	*	*	7	0.0641
5	1/2	*	*	12	0.2089	*	*	15	0.2841
	1/4	*	*	15	0.2920	*	*	14	0.2534
	1/8	*	*	16	0.3122	*	*	14	0.2567
	1/16	*	*	16	0.3096	*	*	14	0.2622
6	1/2	*	*	18	0.3591	*	*	13	0.2372
	1/4	*	*	19	0.3758	*	*	12	0.1965
	1/8	*	*	19	0.3752	*	*	14	0.2577
	1/16	*	*	19	0.3718	*	*	12	0.2046
7	1/2	*	*	13	0.2366	*	*	14	0.2669
	1/4	*	*	15	0.2884	*	*	17	0.3242
	1/8	*	*	15	0.2911	*	*	15	0.2803
	1/16	*	*	15	0.2922	*	*	16	0.2987
8	1/2	*	*	18	0.3505	*	*	30	0.5389
	1/4	*	*	26	0.4854	*	*	32	0.5607
	1/8	*	*	32	0.5620	*	*	31	0.5513
9	1/2	*	*	20	0.3877	*	*	23	0.4472
	1/4	*	*	24	0.4641	*	*	23	0.4479
	1/8	*	*	32	0.5602	*	*	27	0.5102
10	1/2	*	*	24	0.4553	*	*	25	0.4749
	1/4	*	*	31	0.5460	*	*	27	0.5039
	1/8	*	*	39	0.6209	*	*	31	0.5411

Table 6.6: Iteration counts and convergence factor: PCG W(2,2)-cycle, $K = 4$.

p	h	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
1	1/2	5	0.0172	5	0.0172	6	0.0519	8	0.0935
	1/4	5	0.0245	5	0.0250	7	0.0596	7	0.0711
	1/8	5	0.0282	5	0.0284	6	0.0497	7	0.0517
	1/16	6	0.0324	6	0.0328	8	0.0754	7	0.0524
	1/32	6	0.0331	6	0.0335	10	0.1290	7	0.0556
2	1/2	4	0.0102	4	0.0102	4	0.0087	4	0.0087
	1/4	4	0.0115	4	0.0115	5	0.0158	4	0.0115
	1/8	5	0.0164	4	0.0130	7	0.0590	5	0.0177
	1/16	5	0.0186	5	0.0172	11	0.1420	5	0.0191
	1/32	6	0.0361	5	0.0178	20	0.3554	5	0.0207
3	1/2	*	*	4	0.0059	*	*	5	0.0179
	1/4	*	*	5	0.0212	*	*	5	0.0269
	1/8	*	*	5	0.0231	*	*	6	0.0381
	1/16	*	*	6	0.0271	*	*	6	0.0385
	1/32	*	*	6	0.0270	*	*	7	0.0438
4	1/2	*	*	4	0.0119	*	*	5	0.0189
	1/4	*	*	5	0.0233	*	*	5	0.0179
	1/8	*	*	6	0.0306	*	*	5	0.0178
	1/16	*	*	6	0.0318	*	*	5	0.0181
	1/32	*	*	6	0.0319	*	*	5	0.0183
5	1/2	*	*	6	0.0489	*	*	8	0.0901
	1/4	*	*	8	0.0919	*	*	8	0.0870
	1/8	*	*	9	0.1142	*	*	8	0.0875
	1/16	*	*	9	0.1139	*	*	8	0.0865
6	1/2	*	*	7	0.0775	*	*	6	0.0519
	1/4	*	*	8	0.0825	*	*	7	0.0572
	1/8	*	*	8	0.0836	*	*	7	0.0699
	1/16	*	*	8	0.0832	*	*	7	0.0542
7	1/2	*	*	7	0.0685	*	*	7	0.0752
	1/4	*	*	7	0.0738	*	*	7	0.0741
	1/8	*	*	8	0.0842	*	*	7	0.0720
	1/16	*	*	8	0.0860	*	*	7	0.0733
8	1/2	*	*	9	0.1218	*	*	12	0.2170
	1/4	*	*	11	0.1760	*	*	13	0.2377
	1/8	*	*	13	0.2292	*	*	13	0.2392
9	1/2	*	*	8	0.1020	*	*	10	0.1440
	1/4	*	*	11	0.1725	*	*	11	0.1783
	1/8	*	*	13	0.2229	*	*	13	0.2347
10	1/2	*	*	9	0.1298	*	*	10	0.1507
	1/4	*	*	11	0.1716	*	*	11	0.1635
	1/8	*	*	13	0.2276	*	*	13	0.2358



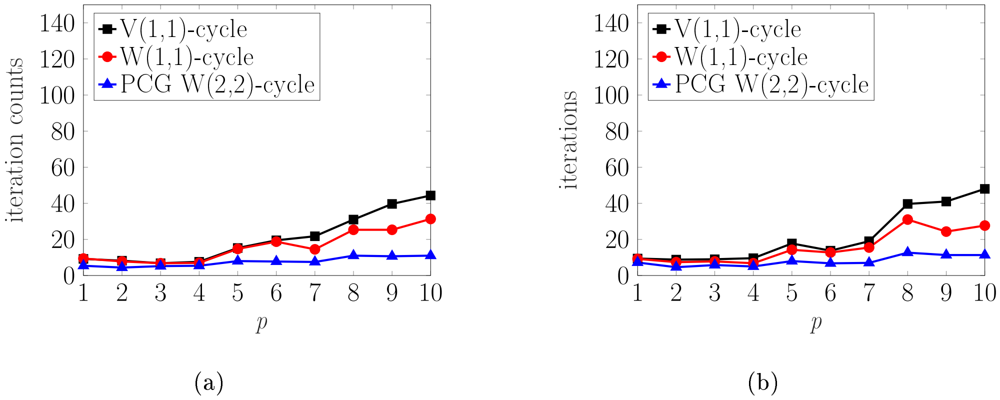
(a)

(b)

(c)

(d)

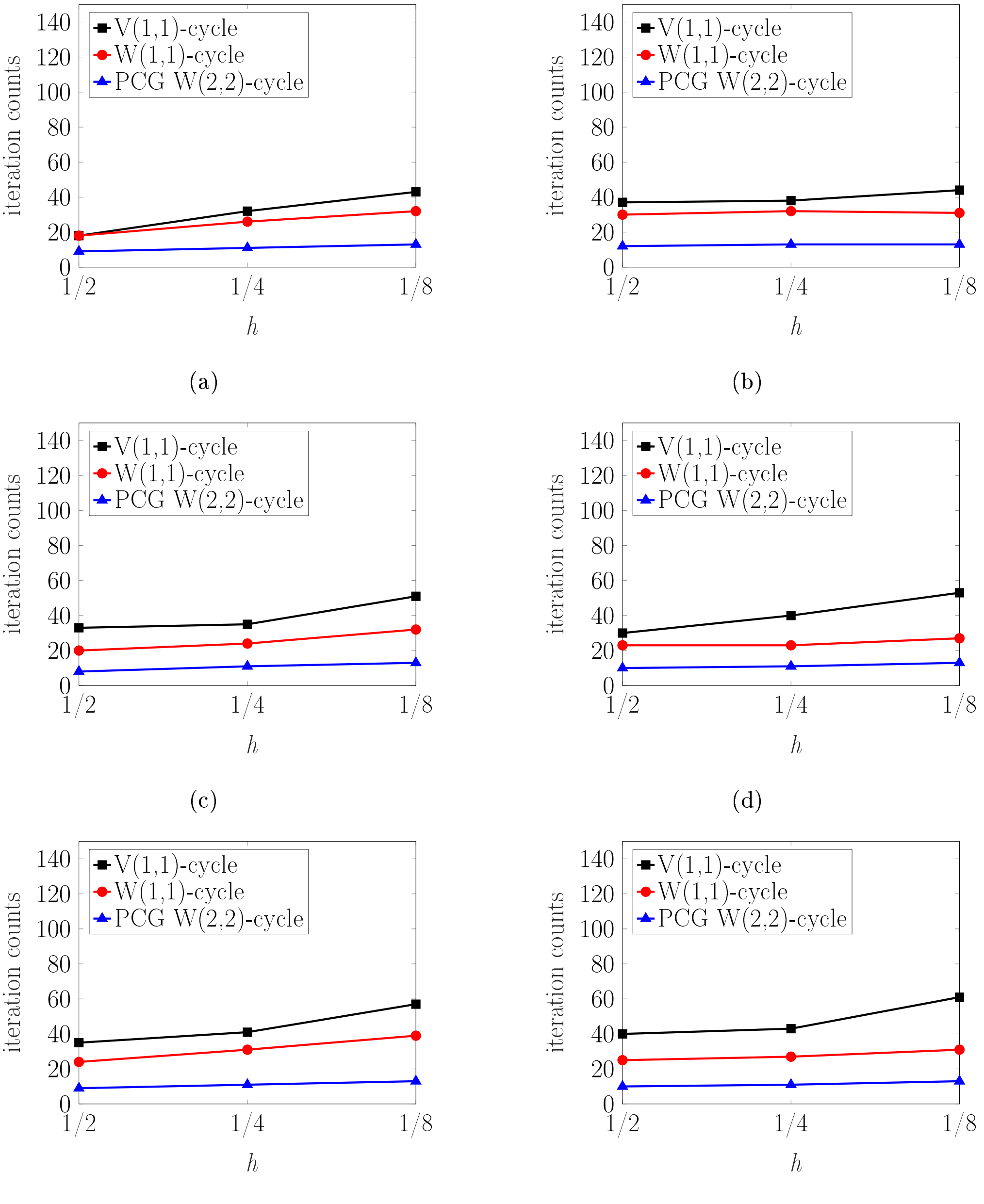
Figure 6.1: Iteration counts as a function of h for H-AMG: (a) structured mesh, $p = 1$, (b) unstructured mesh, $p = 1$, (c) structured mesh, $p = 2$, (d) unstructured mesh, $p = 2$.



(a)

(b)

Figure 6.2: Iteration counts as a function of p for SBA-AMG: (a) structured mesh, (b) unstructured mesh.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 6.3: Iteration counts as a function of h for SBA-AMG: (a) structured mesh, $p = 8$, (b) unstructured mesh, $p = 8$, (c) structured mesh, $p = 9$, (d) unstructured mesh, $p = 9$, (e) structured mesh, $p = 10$, (f) unstructured mesh, $p = 10$.

Table 6.7: Iteration counts and convergence factor: V(ν,ν)-cycle, $K = 4$.

	ν	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	1	10	0.1355	10	0.1457	59	0.7295	10	0.1502
$h = 1/32$	2	7	0.0556	6	0.0419	54	0.7085	7	0.0713
	3	6	0.0391	6	0.0378	51	0.6958	7	0.0561
$p = 4$	1	*	*	8	0.0969	*	*	10	0.1512
$h = 1/8$	2	*	*	7	0.0682	*	*	8	0.0891
	3	*	*	7	0.0579	*	*	7	0.0671
$p = 10$	1	*	*	35	0.5900	*	*	40	0.6301
$h = 1/2$	2	*	*	27	0.4983	*	*	29	0.5275
	3	*	*	24	0.4581	*	*	25	0.4743

Table 6.8: Iteration counts and convergence factor: W(ν,ν)-cycle, $K = 4$.

	ν	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	1	10	0.1389	10	0.1397	28	0.5131	9	0.1197
$h = 1/32$	2	6	0.0279	6	0.0276	26	0.4855	5	0.0251
	3	5	0.0112	4	0.0099	25	0.4708	5	0.0141
$p = 4$	1	*	*	7	0.0693	*	*	7	0.0617
$h = 1/8$	2	*	*	7	0.0573	*	*	5	0.0218
	3	*	*	6	0.0455	*	*	5	0.0196
$p = 10$	1	*	*	24	0.4553	*	*	25	0.4749
$h = 1/2$	2	*	*	21	0.4042	*	*	20	0.3980
	3	*	*	20	0.3863	*	*	19	0.3709

Table 6.9: Iteration counts and convergence factor: PCG W(ν,ν)-cycle, $K = 4$.

	ν	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	1	18	0.3840	17	0.3655	-	-	-	-
$h = 1/32$	2	6	0.0331	6	0.0335	10	0.1290	7	0.0556
	3	5	0.0122	5	0.0121	9	0.0765	5	0.0148
$p = 4$	1	*	*	6	0.0482	*	*	6	0.0509
$h = 1/8$	2	*	*	6	0.0306	*	*	5	0.0178
	3	*	*	5	0.0228	*	*	5	0.0150
$p = 10$	1	*	*	11	0.1913	*	*	12	0.2064
$h = 1/2$	2	*	*	9	0.1298	*	*	10	0.1507
	3	*	*	9	0.1165	*	*	10	0.1389

Table 6.10: Iteration counts and convergence factor: V(1,1)-cycle.

	K	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	2	10	0.1397	10	0.1397	9	0.1190	9	0.1196
$h = 1/32$	3	9	0.1289	10	0.1458	55	0.7133	10	0.1388
	4	10	0.1355	10	0.1457	59	0.7295	10	0.1502
	5	10	0.1360	10	0.1468	61	0.7383	10	0.1550
$p = 4$	2	*	*	7	0.0686	*	*	7	0.0615
$h = 1/8$	3	*	*	8	0.0921	*	*	9	0.1275
	4	*	*	8	0.0969	*	*	10	0.1512
	5	*	*	8	0.0976	*	*	10	0.1549
$p = 10$	2	*	*	22	0.4220	*	*	20	0.3923
$h = 1/2$	3	*	*	34	0.5800	*	*	40	0.6282
	4	*	*	35	0.5900	*	*	40	0.6301
	5	*	*	35	0.5895	*	*	40	0.6300

Table 6.11: Iteration counts and convergence factor: W(1,1)-cycle.

	K	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	2	10	0.1397	10	0.1397	9	0.1190	9	0.1196
$h = 1/32$	3	10	0.1389	10	0.1397	28	0.5120	9	0.1197
	4	10	0.1389	10	0.1397	28	0.5131	9	0.1197
	5	10	0.1389	10	0.1397	28	0.5132	9	0.1197
$p = 4$	2	*	*	7	0.0686	*	*	7	0.0615
$h = 1/8$	3	*	*	7	0.0693	*	*	7	0.0617
	4	*	*	7	0.0693	*	*	7	0.0617
	5	*	*	7	0.0693	*	*	7	0.0617
$p = 10$	2	*	*	22	0.4220	*	*	20	0.3923
$h = 1/2$	3	*	*	24	0.4553	*	*	25	0.4748
	4	*	*	24	0.4553	*	*	25	0.4749
	5	*	*	24	0.4553	*	*	25	0.4749

Table 6.12: Iteration counts and convergence factor: PCG W(2,2)-cycle.

	K	Structured Mesh				Unstructured Mesh			
		H-AMG		SBA-AMG		H-AMG		SBA-AMG	
		N	ρ	N	ρ	N	ρ	N	ρ
$p = 1$	2	6	0.0335	6	0.0335	7	0.0549	7	0.0557
$h = 1/32$	3	6	0.0331	6	0.0335	10	0.1287	7	0.0556
	4	6	0.0331	6	0.0335	10	0.1290	7	0.0556
	5	6	0.0331	6	0.0335	10	0.1290	7	0.0556
$p = 4$	2	*	*	6	0.0306	*	*	5	0.0160
$h = 1/8$	3	*	*	6	0.0306	*	*	5	0.0178
	4	*	*	6	0.0306	*	*	5	0.0178
	5	*	*	6	0.0306	*	*	5	0.0178
$p = 10$	2	*	*	9	0.1239	*	*	9	0.1280
$h = 1/2$	3	*	*	9	0.1298	*	*	10	0.1507
	4	*	*	9	0.1298	*	*	10	0.1507
	5	*	*	9	0.1298	*	*	10	0.1507

Chapter 7
Conclusions and Future Work

In this thesis we have presented new algebraic multigrid methods for solving the linear system of equations stemming from high order conforming and discontinuous Galerkin finite element discretizations of second order elliptic problems.

For each method, we have shown their effectiveness highlighting the advantages and disadvantages. Classical and smoothed aggregation algebraic multigrid methods designed for M-matrices, cf. [Ruge and Stüben, 1987, Vaněk et al., 1996] are well extended to essentially positive matrices stemming from conforming discretizations when $p = 1, \dots, 6$. On the other hand, in order to achieve good scalability for problems discretized with discontinuous Galerkin finite element methods we have extended the standard algebraic multigrid methods, by proposing a new algebraic block aggregation scheme to address the difficulty of the redundancy of the degrees of freedom associated to the same grid point. In addiction, as proposed by [Olson and Schroder, 2011], we have employed a new strength of connection well suited for general matrices, cf. [Olson et al., 2010] and an adaptive smoothed aggregation method, cf. [Brezina et al., 2005]. The obtained results show that the smoothed block aggregation scheme is the first purely scalable algebraic multigrid method for discontinuous Galerkin discretizations of the Poisson problem with a good performance at least for moderate values of p.

Possible further extensions include the following improvements. For the smoothed aggregation and the smoothed block aggregation algebraic multrigrid methods, we can substitute the classic interpolation smoothing step of weighted Jacobi with the use of a different interpolation operator which improves the accuracy, cf. [Mandel et al., 1999, Wan et al., 1999, Olson et al., 2011], in order to have a better p-independence of the AMG methods. In particular the algebraic multigrid methods for high order conforming and discontinuous Galerkin discretizations are studied in [Heys et al., 2005, Olson, 2007, Olson and Schroder, 2011, Sundar et al., 2015], therefore these works can be a starting point to improve the proposed methods.

Possible future developments also include the testing for higher values of p. In addition we will expand the proposed methods to convection-diffusion and anisotropic diffusion for both conforming, cf. [Morano et al., 1998] and discontinuous discretizations, cf. [Schroder, 2012].

Moreover, instead of considering a quasi-uniform mesh of the domain made of non-overlapping shape regular triangles, it is possible to consider polygonal and polyhedral meshes, cf. [Antonietti et al., 2017].

Concerning the computational aspects we point out that for higher values of p the construction of the coarser matrices and interpolation operators becomes more expensive, therefore we should develop new methods based on parallel strategies, cf. [Cleary et al., 1998, Siefert et al., 2014].

Finally we could deepen new algebraic multigrid methods built without geometric assumptions, cf. [Falgout and Vassilevski, 2004] which are therefore more suitable for general matrices.

Bibliography

[Adams, 1975] Adams, R. A. (1975). *Sobolev spaces*. Academic Press [A subsidiary of Harcourt Brace Jovanovich, Publishers], New York-London. Pure and Applied Mathematics, Vol. 65. (Cited on page 5.)

[Antonietti and Houston, 2011] Antonietti, P. F. and Houston, P. (2011). A class of domain decomposition preconditioners for *hp*-discontinuous Galerkin finite element methods. *J. Sci. Comput.*, 46(1):124–149. (Cited on page 9.)

[Antonietti et al., 2017] Antonietti, P. F., Houston, P., Hu, X., Sarti, M., and Verani, M. (2017). Multigrid algorithms for *hp*-version interior penalty discontinuous Galerkin methods on polygonal and polyhedral meshes. *Calcolo*. (Cited on pages 2, 53 and 78.)

[Antonietti et al., 2015] Antonietti, P. F., Sarti, M., and Verani, M. (2015). Multigrid algorithms for *hp*-discontinuous Galerkin discretizations of elliptic problems. *SIAM J. Numer. Anal.*, 53(1):598–618. (Cited on pages 2, 11, 53 and 67.)

[Arnold, 1982] Arnold, D. N. (1982). An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760. (Cited on page 9.)

[Babuška and Suri, 1987] Babuška, I. and Suri, M. (1987). The *h-p* version of the finite element method with quasi-uniform meshes. *RAIRO Modél. Math. Anal. Numér.*, 21(2):199–238. (Cited on page 7.)

[Babuška and Suri, 1994] Babuška, I. and Suri, M. (1994). The *p* and *h-p* versions of the finite element method, basic principles and properties. *SIAM Rev.*, 36(4): 578–632. (Cited on page 7.)

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

[Blaheta, 1986] Blaheta, R. (1986). A multilevel method with correction by aggregation for solving discrete elliptic problems. *Apl. Mat.*, 31(5):365–378. (Cited on pages 27, 34 and 37.)

[Braess, 1995] Braess, D. (1995). Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55(4):379–393. (Cited on pages 1 and 44.)

[Bramble, 1993] Bramble, J. H. (1993). *Multigrid methods*, volume 294 of *Pitman Research Notes in Mathematics Series*. Longman Scientific & Technical, Harlow; copublished in the United States with John Wiley & Sons, Inc., New York. (Cited on page 1.)

[Bramble et al., 1991] Bramble, J. H., Pasciak, J. E., Wang, J. P., and Xu, J. (1991). Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.*, 57(195):23–45. (Cited on page 37.)

[Brandt, 1977a] Brandt, A. (1977a). Multi-level adaptive solutions to boundary-value problems. *Math. Comp.*, 31(138):333–390. (Cited on page 1.)

[Brandt, 1977b] Brandt, A. (1977b). Multi-level adaptive techniques (MLAT) for partial differential equations: ideas and software. In *Mathematical software, III (Proc. Sympos., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1977)*, pages 277–318. Publ. Math. Res. Center, No. 39. Academic Press, New York. (Cited on page 15.)

[Brandt, 1982] Brandt, A. (1982). Guide to multigrid development. In *Multigrid methods (Cologne, 1981)*, volume 960 of *Lecture Notes in Math.*, pages 220–312. Springer, Berlin-New York. (Cited on pages 15 and 16.)

[Brandt, 1986] Brandt, A. (1986). Algebraic multigrid theory: the symmetric case. *Appl. Math. Comput.*, 19(1-4):23–56. Second Copper Mountain conference on multigrid methods (Copper Mountain, Colo., 1985). (Cited on pages 18, 20, 27, 28 and 38.)

[Brandt et al., 1982] Brandt, A., McCormick, S., and Ruge, J. (1982). Algebraic multigrid (AMG) for automatic multigrid solution with application to geodetic computations. *Technical Report, Inst. Comp. Studies, Colo. State Univ.* (Cited on page 1.)

[Brandt et al., 1985] Brandt, A., McCormick, S., and Ruge, J. (1985). Algebraic multigrid (AMG) for sparse matrix equations. In *Sparsity and its applications (Loughborough, 1983)*, pages 257–284. Cambridge Univ. Press, Cambridge. (Cited on page 18.)

[Brannick et al., 2006] Brannick, J., Brezina, M., MacLachlan, S., Manteuffel, T., McCormick, S., and Ruge, J. (2006). An energy-based AMG coarsening strategy. *Numer. Linear Algebra Appl.*, 13(2-3):133–148. (Cited on page 54.)

[Brenner et al., 2011] Brenner, S. C., Cui, J., Gudi, T., and Sung, L.-Y. (2011). Multigrid algorithms for symmetric discontinuous Galerkin methods on graded meshes. *Numer. Math.*, 119(1):21–47. (Cited on pages 2, 53 and 67.)

[Brenner et al., 2009] Brenner, S. C., Cui, J., and Sung, L.-Y. (2009). Multigrid methods for the symmetric interior penalty method on graded meshes. *Numer. Linear Algebra Appl.*, 16(6):481–501. (Cited on pages 2, 53 and 67.)

[Brenner and Owens, 2007] Brenner, S. C. and Owens, L. (2007). A W-cycle algorithm for a weakly over-penalized interior penalty method. *Comput. Methods Appl. Mech. Engrg.*, 196(37-40):3823–3832. (Cited on page 2.)

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

[Brezina et al., 2005] Brezina, M., Falgout, R., MacLachlan, S., Manteuffel, T., McCormick, S., and Ruge, J. (2005). Adaptive smoothed aggregation (αSA) multigrid. *SIAM Rev.*, 47(2):317–346. (Cited on pages 54 and 77.)

[Brezina et al., 2012] Brezina, M., Vaněk, P., and Vassilevski, P. S. (2012). An improved convergence analysis of smoothed aggregation algebraic multigrid. *Numer. Linear Algebra Appl.*, 19(3):441–469. (Cited on pages 37 and 40.)

[Briani et al., 2012] Briani, M., Sommariva, A., and Vianello, M. (2012). Computing Fekete and Lebesgue points: simplex, square, disk. *J. Comput. Appl. Math.*, 236(9):2477–2486. (Cited on page 12.)

[Briggs and McCormick, 1987] Briggs, W. and McCormick, S. (1987). Introduction. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 1–30. SIAM, Philadelphia, PA. (Cited on pages 23 and 24.)

[Briggs, 1987] Briggs, W. L. (1987). *A multigrid tutorial.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. (Cited on page 23.)

[Briggs et al., 2000] Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A multigrid tutorial.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition. (Cited on pages 15, 17, 18, 22, 23 and 24.)

[Chan and Elman, 1989] Chan, T. F. and Elman, H. C. (1989). Fourier analysis of iterative methods for elliptic problems. *SIAM Rev.*, 31(1):20–49. (Cited on pages 15 and 16.)

[Chan et al., 1998] Chan, T. F., Xu, J., and Zikatanov, L. (1998). An agglomeration multigrid method for unstructured grids. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, volume 218 of *Contemp. Math.*, pages 67–81. Amer. Math. Soc., Providence, RI. (Cited on page 44.)

[Chang et al., 1996] Chang, Q., Wong, Y. S., and Fu, H. (1996). On the algebraic multigrid method. *J. Comput. Phys.*, 125(2):279–292. (Cited on pages 53, 56, 58 and 60.)

[Chang et al., 1992] Chang, Q. S., Wong, Y. S., and Li, Z. F. (1992). New interpolation formulas of using geometric assumptions in the algebraic multigrid method. *Appl. Math. Comput.*, 50(2-3):223–254. (Cited on pages 56, 58 and 60.)

[Ciarlet, 1978] Ciarlet, P. G. (1978). *The finite element method for elliptic problems*. North-Holland Publishing Co., Amsterdam-New York-Oxford. Studies in Mathematics and its Applications, Vol. 4. (Cited on page 9.)

[Ciarlet and Raviart, 1972] Ciarlet, P. G. and Raviart, P.-A. (1972). General Lagrange and Hermite interpolation in $\mathbf{R}^n$ with applications to finite element methods. *Arch. Rational Mech. Anal.*, 46:177–199. (Cited on page 13.)

[Cleary et al., 1998] Cleary, A. J., Falgout, R. D., Henson, V. E., and Jones, J. E. (1998). Coarse-grid selection for parallel algebraic multigrid. In *Solving irregularly structured problems in parallel (Berkeley, CA, 1998)*, volume 1457 of *Lecture Notes in Comput. Sci.*, pages 104–115. Springer, Berlin. (Cited on pages 28 and 78.)

[Cleary et al., 2000] Cleary, A. J., Falgout, R. D., Henson, V. E., Jones, J. E., Manteuffel, T. A., McCormick, S. F., Miranda, G. N., and Ruge, J. W. (2000). Robustness and scalability of algebraic multigrid. *SIAM J. Sci. Comput.*, 21(5): 1886–1908. Iterative methods for solving systems of algebraic equations (Copper Mountain, CO, 1998). (Cited on pages 28 and 44.)

[Diestel, 2010] Diestel, R. (2010). *Graph theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, Heidelberg, fourth edition. (Cited on page 18.)

[Dobrev et al., 2006] Dobrev, V. A., Lazarov, R. D., Vassilevski, P. S., and Zikatanov, L. T. (2006). Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. *Numer. Linear Algebra Appl.*, 13(9):753–770. (Cited on page 2.)

[Dubiner, 1991] Dubiner, M. (1991). Spectral methods on triangles and other domains. *J. Sci. Comput.*, 6(4):345–390. (Cited on page 13.)

[Epshteyn and Rivière, 2007] Epshteyn, Y. and Rivière, B. (2007). Estimation of penalty parameters for symmetric interior penalty Galerkin methods. *J. Comput. Appl. Math.*, 206(2):843–872. (Cited on page 9.)

[Falgout and Vassilevski, 2004] Falgout, R. D. and Vassilevski, P. S. (2004). On generalizing the algebraic multigrid framework. *SIAM J. Numer. Anal.*, 42(4): 1669–1693. (Cited on page 78.)

[Fedorenko, 1961] Fedorenko, R. P. (1961). A relaxation method of solution of elliptic difference equations. *Ž. Vyčisl. Mat. i Mat. Fiz.*, 1:922–927. (Cited on page 1.)

[Gaspar et al., 2010] Gaspar, F., Gracia, J. L., Lisbona, F. J., and Rodrigo, C. (2010). Efficient geometric multigrid implementation for triangular grids. *J. Comput. Appl. Math.*, 234(4):1027–1035. (Cited on page 16.)

[Gaspar et al., 2009] Gaspar, F. J., Gracia, J. L., Lisbona, F. J., and Rodrigo, C. (2009). On geometric multigrid methods for triangular grids using three-coarsening strategy. *Appl. Numer. Math.*, 59(7):1693–1708. (Cited on page 16.)

[Gibbons, 1985] Gibbons, A. (1985). *Algorithmic graph theory.* Cambridge University Press, Cambridge. (Cited on page 18.)

[Gopalakrishnan and Kanschat, 2003] Gopalakrishnan, J. and Kanschat, G. (2003). A multilevel discontinuous Galerkin method. *Numer. Math.*, 95(3):527–550. (Cited on pages 1 and 2.)

[Hackbusch, 1982] Hackbusch, W. (1982). Multigrid convergence theory. In *Multigrid methods (Cologne, 1981)*, volume 960 of *Lecture Notes in Math.*, pages 177–219. Springer, Berlin-New York. (Cited on page 15.)

[Hackbusch, 1985] Hackbusch, W. (1985). *Multigrid methods and applications*, volume 4 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin. (Cited on page 1.)

[Helenbrook and Atkins, 2006] Helenbrook, B. T. and Atkins, H. L. (2006). Application of p-multigrid to discontinuous Galerkin formulations of the Poisson equation. *AIAA Journal*, 44. (Cited on page 2.)

[Helenbrook and Atkins, 2008] Helenbrook, B. T. and Atkins, H. L. (2008). Solving discontinuous Galerkin formulations of Poisson's equation using geometric and p multigrid. *AIAA Journal*, 46. (Cited on page 2.)

[Hemker et al., 2003] Hemker, P. W., Hoffmann, W., and van Raalte, M. H. (2003). Two-level Fourier analysis of a multigrid approach for discontinuous Galerkin discretization. *SIAM J. Sci. Comput.*, 25(3):1018–1041. (Cited on page 53.)

[Hemker et al., 2004] Hemker, P. W., Hoffmann, W., and van Raalte, M. H. (2004). Fourier two-level analysis for discontinuous Galerkin discretization with linear elements. *Numer. Linear Algebra Appl.*, 11(5-6):473–491. (Cited on page 53.)

[Hesthaven and Warburton, 2008] Hesthaven, J. S. and Warburton, T. (2008). *Nodal discontinuous Galerkin methods*, volume 54 of *Texts in Applied Mathematics*. Springer, New York. Algorithms, analysis, and applications. (Cited on page 13.)

[Heys et al., 2005] Heys, J. J., Manteuffel, T. A., McCormick, S. F., and Olson, L. N. (2005). Algebraic multigrid for higher-order finite elements. *J. Comput. Phys.*, 204(2):520–532. (Cited on pages 1 and 77.)

[Houston et al., 2002] Houston, P., Schwab, C., and Süli, E. (2002). Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39(6):2133–2163. (Cited on pages 9 and 11.)

[Kettler, 1982] Kettler, R. (1982). Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In *Multigrid methods (Cologne, 1981)*, volume 960 of *Lecture Notes in Math.*, pages 502–534. Springer, Berlin-New York. (Cited on page 25.)

[Koornwinder, 1975] Koornwinder, T. (1975). Two-variable analogues of the classical orthogonal polynomials. In *Theory and application of special functions (Proc. Advanced Sem., Math. Res. Center, Univ. Wisconsin, Madison, Wis., 1975)*, pages 435–495. Math. Res. Center, Univ. Wisconsin, Publ. No. 35. Academic Press, New York. (Cited on page 13.)

[Mandel et al., 1999] Mandel, J., Brezina, M., and Vaněk, P. (1999). Energy optimization of algebraic multigrid bases. *Computing*, 62(3):205–228. (Cited on pages 1, 27 and 77.)

[Mandel et al., 1987] Mandel, J., McCormick, S., and Bank, R. (1987). Variational multigrid theory. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 131–177. SIAM, Philadelphia, PA. (Cited on pages 18 and 22.)

[Mascarenhas et al., 2010] Mascarenhas, B. S., Helenbrook, B. T., and Atkins, H. L. (2010). Coupling p-multigrid to geometric multigrid for discontinuous Galerkin formulations of the convection-diffusion equation. *J. Comput. Phys.*, 229(10):3664–3674. (Cited on page 2.)

[McCormick, 1985] McCormick, S. F. (1985). Multigrid methods for variational problems: general theory for the V-cycle. *SIAM J. Numer. Anal.*, 22(4):634–643. (Cited on page 34.)

[McCormick, 1987] McCormick, S. F., editor (1987). *Multigrid methods*, volume 3 of *Frontiers in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA. (Cited on page 15.)

[McCormick and Ruge, 1982] McCormick, S. F. and Ruge, J. W. (1982). Multigrid methods for variational problems. *SIAM J. Numer. Anal.*, 19(5):924–929. (Cited on page 54.)

[Míka and Vaněk, 1992] Míka, S. and Vaněk, P. (1992). Acceleration of convergence of a two-level algebraic algorithm by aggregation in smoothing process. *Appl. Math.*, 37(5):343–356. (Cited on page 34.)

[Morano et al., 1998] Morano, E., Mavriplis, D. J., and Venkatakrishnan, V. (1998). Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems. *SIAM J. Sci. Comput.*, 20(2):393–415. (Cited on page 78.)

[Nastase and Mavriplis, 2006] Nastase, C. R. and Mavriplis, D. J. (2006). High-order discontinuous Galerkin methods using an *hp*-multigrid approach. *J. Comput. Phys.*, 213. (Cited on page 2.)

[Olson, 2007] Olson, L. (2007). Algebraic multigrid preconditioning of high-order spectral elements for elliptic problems on a simplicial mesh. *SIAM J. Sci. Comput.*, 29(5):2189–2209. (Cited on page 77.)

[Olson et al., 2010] Olson, L. N., Schroder, J., and Tuminaro, R. S. (2010). A new perspective on strength measures in algebraic multigrid. *Numer. Linear Algebra Appl.*, 17(4):713–733. (Cited on pages 60 and 77.)

[Olson and Schroder, 2011] Olson, L. N. and Schroder, J. B. (2011). Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems. *J. Comput. Phys.*, 230(18):6959–6976. (Cited on pages 2, 53, 54, 60, 66, 68 and 77.)

[Olson et al., 2011] Olson, L. N., Schroder, J. B., and Tuminaro, R. S. (2011). A general interpolation strategy for algebraic multigrid using energy minimization. *SIAM J. Sci. Comput.*, 33(2):966–991. (Cited on pages 27, 54, 68 and 77.)

[Perugia and Schötzau, 2002] Perugia, I. and Schötzau, D. (2002). An *hp*-analysis of the local discontinuous Galerkin method for diffusion problems. *J. Sci. Comput.*, 17(1-4):561–571. (Cited on page 11.)

[Prill et al., 2009] Prill, F., Lukáčová-Medviďová, M., and Hartmann, R. (2009). Smoothed aggregation multigrid for the discontinuous Galerkin method. *SIAM J. Sci. Comput.*, 31(5):3503–3528. (Cited on page 2.)

[Quarteroni, 2014] Quarteroni, A. (2014). *Numerical models for differential problems*, volume 8 of *MS&A. Modeling, Simulation and Applications*. Springer, Milan, second edition. Translated from the fifth (2012) Italian edition by Silvia Quarteroni. (Cited on page 5.)

[Quarteroni et al., 2007] Quarteroni, A., Sacco, R., and Saleri, F. (2007). *Numerical mathematics*, volume 37 of *Texts in Applied Mathematics*. Springer-Verlag, Berlin, second edition. (Cited on page 25.)

[Rivière, 2008] Rivière, B. (2008). *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. Theory and implementation. (Cited on page 5.)

[Ruge and Stüben, 1985] Ruge, J. and Stüben, K. (1985). Efficient solution of finite difference and finite element equations. In *Multigrid methods for integral and differential equations (Bristol, 1983)*, volume 3 of *Inst. Math. Appl. Conf. Ser. New Ser.*, pages 169–212. Oxford Univ. Press, New York. (Cited on page 1.)

[Ruge and Stüben, 1987] Ruge, J. W. and Stüben, K. (1987). Algebraic multigrid. In *Multigrid methods*, volume 3 of *Frontiers Appl. Math.*, pages 73–130. SIAM, Philadelphia, PA. (Cited on pages 1, 27, 28, 29, 30, 33, 34, 39, 53, 54 and 77.)

[Saad, 2003] Saad, Y. (2003). *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition. (Cited on page 21.)

[Schroder, 2012] Schroder, J. B. (2012). Smoothed aggregation solvers for anisotropic diffusion. *Numer. Linear Algebra Appl.*, 19(2):296–312. (Cited on pages 2 and 78.)

[Schwab, 1998] Schwab, C. (1998). *p- and hp-finite element methods*. Numerical Mathematics and Scientific Computation. The Clarendon Press, Oxford University Press, New York. Theory and applications in solid and fluid mechanics. (Cited on page 7.)

[Siefert et al., 2014] Siefert, C., Tuminaro, R., Gerstenberger, A., Scovazzi, G., and Collis, S. S. (2014). Algebraic multigrid techniques for discontinuous Galerkin methods with varying polynomial order. *Comput. Geosci.*, 18(5):597–612. (Cited on page 78.)

[Stamm and Wihler, 2010] Stamm, B. and Wihler, T. P. (2010). *hp*-optimal discontinuous Galerkin methods for linear elliptic problems. *Math. Comp.*, 79(272):2117–2133. (Cited on page 11.)

[Stüben, 1983] Stüben, K. (1983). Algebraic multigrid (AMG): experiences and comparisons. *Appl. Math. Comput.*, 13(3-4):419–451. (Cited on pages 1, 27, 28, 29, 30 and 33.)

[Stüben, 1999] Stüben, K. (1999). Algebraic multigrid (AMG): An introduction with applications. *GMD-Report.* (Cited on pages 18, 28, 32, 38 and 39.)

[Stüben, 2001] Stüben, K. (2001). A review of algebraic multigrid. *J. Comput. Appl. Math.*, 128(1-2):281–309. Numerical analysis 2000, Vol. VII, Partial differential equations. (Cited on page 44.)

[Stüben and Trottenberg, 1982] Stüben, K. and Trottenberg, U. (1982). Multigrid methods: fundamental algorithms, model problem analysis and applications. In *Multigrid methods (Cologne, 1981)*, volume 960 of *Lecture Notes in Math.*, pages 1–176. Springer, Berlin-New York. (Cited on page 15.)

[Sundar et al., 2015] Sundar, H., Stadler, G., and Biros, G. (2015). Comparison of multigrid algorithms for high-order continuous finite element discretizations. *Numer. Linear Algebra Appl.*, 22(4):664–680. (Cited on page 77.)

[Taylor et al., 2000] Taylor, M. A., Wingate, B. A., and Vincent, R. E. (2000). An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal.*, 38(5):1707–1720. (Cited on page 12.)

[Trottenberg et al., 2001] Trottenberg, U., Oosterlee, C. W., and Schüller, A. (2001). *Multigrid.* Academic Press, Inc., San Diego, CA. With contributions by A. Brandt, P. Oswald and K. Stüben. (Cited on page 15.)

[Vakhutinsky et al., 1979] Vakhutinsky, I. Y., Dudkin, L. M., and Ryvkin, A. A. (1979). Iterative aggregation—a new approach to the solution of large-scale problems. *Econometrica*, 47(4):821–841. (Cited on page 27.)

[van Raalte and Hemker, 2005] van Raalte, M. H. and Hemker, P. W. (2005). Two-level multigrid analysis for the convection-diffusion equation discretized by a discontinuous Galerkin method. *Numer. Linear Algebra Appl.*, 12(5-6):563–584. (Cited on page 2.)

[Vaněk, 1992] Vaněk, P. (1992). Acceleration of convergence of a two-level algorithm by smoothing transfer operators. *Appl. Math.*, 37(4):265–274. (Cited on pages 27 and 34.)

[Vaněk, 1995] Vaněk, P. (1995). Fast multigrid solver. *Appl. Math.*, 40(1):1–20. (Cited on pages 34 and 35.)

[Vaněk et al., 2001] Vaněk, P., Brezina, M., and Mandel, J. (2001). Convergence of algebraic multigrid based on smoothed aggregation. *Numer. Math.*, 88(3): 559–579. (Cited on pages 37 and 40.)

[Vaněk et al., 1996] Vaněk, P., Mandel, J., and Brezina, M. (1996). Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems. *Computing*, 56(3):179–196. International GAMM-Workshop on Multi-level Methods (Meisdorf, 1994). (Cited on pages 1, 27, 34, 35, 37, 39, 53, 54, 60 and 77.)

[Wan et al., 1999] Wan, W. L., Chan, T. F., and Smith, B. (1999). An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comput.*, 21(4):1632–1649. (Cited on pages 1 and 77.)

[Warburton and Hesthaven, 2003] Warburton, T. and Hesthaven, J. S. (2003). On the constants in *hp*-finite element trace inverse inequalities. *Comput. Methods Appl. Mech. Engrg.*, 192(25):2765–2773. (Cited on page 9.)

[Wheeler, 1978] Wheeler, M. F. (1978). An elliptic collocation-finite element method with interior penalties. *SIAM J. Numer. Anal.*, 15(1):152–161. (Cited on page 9.)

[Xu and Zikatanov, 2017] Xu, J. and Zikatanov, L. (2017). Algebraic multigrid methods. *Acta Numer.*, 26:591–721. (Cited on pages 18, 39 and 40.)

[Yang, 2006] Yang, U. M. (2006). Parallel algebraic multigrid methods—high performance preconditioners. In *Numerical solution of partial differential equations on parallel computers*, volume 51 of *Lect. Notes Comput. Sci. Eng.*, pages 209–236. Springer, Berlin. (Cited on page 36.)