Solveurs rapides pour des discrétisations robustes en mécanique des fluides numérique

Pierre Matalon

THÈSE POUR OBTENIR LE GRADE DE DOCTEUR DE L'UNIVERSITÉ DE MONTPELLIER

En Mathématiques et Modélisation

École doctorale : Information, Structures, Systèmes (I2S)

Unité de recherche : Institut Montpelliérain Alexander Grothendieck (IMAG)

En partenariat international avec Friedrich-Alexander Universität, Erlangen-Nürnberg, Allemagne

Fast solvers for robust discretizations in computational fluid dynamics

Solveurs rapides pour des discrétisations robustes en mécanique des fluides numérique

Présentée par Pierre MATALON

Le 15 10 2021

Sous la direction de Daniele DI PIETRO et Ulrich RUEDE

Devant le jury composé de

Daniele DI PIETRO, Professeur, Université de Montpellier	Directeur de thèse
Ulrich RUEDE, Professeur, Friedrich Alexander Universität, Erlangen-Nürnberg	Co-directeur de thèse
Paola ANTONIETTI, Professeur, Politecnico di Milano	Président du jury, rapporteur
Carmen RODRIGO, Professeur associé, Universidad de Zaragoza	Rapporteur
Lorenzo BOTTI, Professeur associé, Università degli studi di Bergamo	Examinateur
Axel KLAWONN, Professeur, Universität zu Köln	Examinateur
Harald KOESTLER, Professeur, Friedrich Alexander Universität, Erlangen-Nürnberg	Examinateur
Fabien MARCHE, Maître de Conférence, Université de Montpellier	Examinateur
Frank HUELSEMANN, Ingénieur docteur, EDF Lab Paris-Saclay	Invité
Paul MYCEK, Chargé de Recherche, CERFACS	Invité
Daniel RUIZ, Maître de Conférence, ENSEEIHT-IRIT	Invité

UNIVERSITÉ DE MONTPELLIER

Schnelle Löser für robuste Diskretisierungen in numerischer Strömungsdynamik

Der Technische Fakultät, Lehrstuhl Informatik 10 (Systemsimulation)
der Friedrich-Alexander-Universität
Erlangen-Nürnberg
zur
Erlangung des Doktorgrades

Doktor-Ingenieur (Dr.-Ing.)

vorgelegt von

Pierre MATALON

aus Toulouse

Als Dissertation genehmigt
von der Technische Fakultät, Lehrstuhl Informatik 10 (Systemsimulation)
der Friedrich-Alexander-Universität Erlangen-Nürnberg

Tag der mündlichen Prüfung:	der 15 Oktober 2021
Vorsitzender des Promotionsorgans:	Prof. Dr.-Ing. Knut Graichen
Gutachter:	Prof. Paola Antonietti
	Prof. Carmen Rodrigo
	Prof. Dr. Ulrich Rüde

The work of this Ph.D thesis has been conducted in collaboration with the following institutional partners:

Academic cotutelle:

UNIVERSITÉ DE MONTPELLIER

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

TECHNISCHE FAKULTÄT

Hosting and scientific collaboration:

CERFACS

CENTRE EUROPÉEN DE RECHERCHE ET DE FORMATION AVANCÉE EN CALCUL SCIENTIFIQUE

IRIT

Industrial partner:

EDF

Funding:

ANR AGENCE NATIONALE DE LA RECHERCHE

Aknowledgements

Getting accepted in Ph.D was the most difficult task. More difficult than actually doing it. Despite the anxiety, the moments of doubts, the premature overjoys followed by dreadful disappointments, the level of stress and the self-inflicted pressure that punctuated these last three years of research, I can say without a shred of doubt that convincing people to take me as a Ph.D student was a rougher and longer journey. I did hold the required degrees, and I did have good enough grades when I was an engineering student. Was I competent enough? Motivated enough? I hope the outcome of this thesis proves so. Then, why was it so hard for me to find a Ph.D? The answer is simple: I made a mistake. A mistake so big it would slam the gates of the academic world in my face and lock me out. And this unforgivable mistake was: I stepped out. Instead of starting a Ph.D right out of Master, as every well-behaved student should, I started an industrial career in the private sector. Clearly, I learnt at my expense how ill-inspired that was. This is not true in every country, but in France, we hate taking risks. The only acceptable candidates are the ones presenting linear academic and career paths. Ideally, you knew what you wanted to do at eighteen and never budged. Failures, attempts, hesitations and career changes are badly received and usually suffice to get your application instantly thrown out. For my part, the path that led me to this Ph.D was all but linear. At best, polynomial, at worst, strongly oscillatory.

Over the years, in my search for a Ph.D, the countless rejections and false hopes made me bitter. As bitter as you can be when nobody gives you the chance that has been given to people you feel less deserving. It quickly becomes you against an unfair world, and the world always wins. It beats you without violence, slowly, just by wearing you out. And it does so mechanically, unintentionally, without purpose or hate, without even acknowledging that you exist. But since I started the research work I close up with the present thesis, gratitude has replaced bitterness. And it is now my great pleasure to make use of this space to thank the people who pulled me out of that unfair world I was stuck in and let me into theirs.

I would like to start chronologically with Corinne Mailhes, the head of the TéSA laboratory, which I contacted during my search for a Ph.D. She gave me the wise yet simple advice to get in touch with my old professors from back when I was a student. Although this seems like the obvious thing to do, I must admit that this idea had never crossed my mind. After all, more than a decade had passed since I last saw a teacher. How could any one of them have any recollection of who I was? It was a long shot, but I gave it a try, and I have Corinne to thank for it. I sent an email to the math teachers I was the most fond of when I was a student: Joseph Noailles and Daniel Ruiz. And, by a profound mystery of the humain brain, Joseph Noailles remembered me! To this day, I still wonder how that is possible. Thank you, Joseph, for your great memory and your decisive recommendation. Turning you down when you offered me a

Ph.D fifteen years ago remains my deepest regret. At the time, I had an erroneous view on research, and an even more erroneous one on what I wanted to do with my life.

I met with Daniel in Toulouse shortly after. By a happy coincidence, an ANR proposal he was involved in was just accepted, which included the funding of a Ph.D. He could have easily found a more reassuring, freshly graduated candidate, but he decided to help me convince the other deciding parties to give me the position. Whenever they were in town, he would call me and I would show up, just to meet them informally. This little lobbying scheme worked perfectly, and no other applicant was ever interviewed. Daniel, I owe the best job I have ever had to your support, and for that I will be forever grateful. I'll be happy to become half the mathematician you are, and I'll be even happier to become half the *human being* you are. Of course, I also want to thank my Ph.D advisors Ulrich Rüde and Daniele Di Pietro, as well as Frank Hülsemann, for their approval. You could have stayed on the safe path by demanding a more classical candidate, but you agreed to take a chance on an atypical profile. Thank you!

During the course of this research work, I have been lucky to work with great people who have helped me navigate the ocean of unsolved problems so that, ultimately, a few solutions could be uncovered. So I would like to thank Uli for his crucial help when I started dipping my toes in scientific computing, his knowledge on multigrid methods and his guidance throughout the thesis; Daniele for his introduction to HHO, his answers to my many technical questions and his very thorough and indispensable paper reviews; Daniel for making me lay things down on the blackboard to make sure we understood everything from the very beginning, and for being the mathematical swiss army knife that got me out of trouble more than once; Frank for his introduction to AMG and his unfailing interest in my results; and Paul Mycek, for his great involvement and availability, who followed and participated to this work more closely than anyone else. Thank you all for trusting me and giving me complete autonomy to develop my ideas. More people, of course, deserve to be cited. In particular, Carola Kruse and M'Barek Fares at CERFACS, Jérôme Bonelle and Yvan Fournier at EDF, as well as Lorenzo Botti. I am looking forward to working with all of you again.

I have been happy to be part of the Algo team at CERFACS and the APO team at IRIT. Special thoughts to my fellow Ph.D students and postdocs: Philippe (I will always remember our trips to Erlangen and our nights at the Borriquito :-D), Benoît, Nicolas, Anthony, Martin, Vincent, Luce, Boris, Bastien, Antoine, and the permanent researchers: Anthony, Selime, Iain, Luc, Serge, Ehouarn, Joseph, Sandrine, Alfredo, Olivier, Ronan. Thank you also to Brigitte and Michèle at CERFACS, and SAM and Muriel at IRIT!

As the adventure of the Ph.D closes with a defense, I am grateful to the referees Paola Antonietti and Carmen Rodrigo for taking the time to carefully read this manuscript, as well as the other examiners, Lorenzo Botti, Axel Klawonn, Harald Köstler and Fabien Marche.

Research is an emotional roller coaster. Difficulties, small failures and temporary setbacks take disproportionate importance, which can quickly turn into psychological drama. On that matter, Lisa deserves my deepest acknowledgements, for her unfailing moral support and for constantly helping me put things in perspective. My little nephews, Milo and Octave, have also played a significant role. They also spend their lives trying, failing and finally succeeding, without giving up hope and while keeping big smiles on their faces (ok, not always, but most of the time). Thanks for being here!

This part would not be complete without a final word for the various doctoral offices I have been in contact with. It gives me great pleasure to finally acknowledge the efforts of those disembodied administrations to make my life a living hell. The many redundent, unclear at best, and sometimes completely illogical procedures came close to having me commited in a mental institution. The international cotutelle was an especially exquisite torture. Getting one administration to move is already known to be tough. Now, trying to get two administrations to communicate and work together resulted in the most perfect bureaucratic immobility I have ever seen. Administrative procedures complexify by the day, slowing down or blocking every initiative. They have the power to destroy entire countries, and I am conviced that one day they will. They certainly do have the power to suck the life out of people. (I wonder what they do with it... I'm guessing, consume it for eternal youth, as always.)

Contents

Chapter 1

Introduction

To explain all nature is too difficult a task
for any one man or even for any one age.

Isaac Newton (1643-1727)

1.1 Numerical simulation and scientific computing

Understanding the physical laws that govern the world we live in is a vast project and quite an ambitious task. Physicists over the centuries have endeavoured to describe physical phenomena through mathematical models, which play the role of an approximate reality we can understand, as opposed to the real world that we do not. A model's goal is to predict the evolution of a physical system, and is only as accurate as the predictions it provides. But even if understanding nature in all its complexity is far out of our intellectual reach, apprehending the main lines of its fabric through models has tremendous value. In a mathematical model, the system under study is represented by a list of variable values linked together by a system of equations determining their evolution in space and time. Once initial and boundary values are set, their progress can be computed by solving the equations, thus yielding future states of the system. An accurate model reproduces what nature does, and therefore shows that all the relevant mechanisms at play have been identified and understood.

Mathematical models usually involve a large number of unknowns. Solving their equations consequently requires a large computing power that only machines can provide. Making predictions of physical phenomena, that is, *simulating* nature, by the means of numerical operations performed on computers is then referred to as *numerical simulation*. Designing the models, solving the equations, and making the best use of the available computing resources places numerical simulation at the crossroad of physics, mathematics and computer science. Jointly, they give rise to a scientific discipline known as *scientific computing*. Additionally to understanding in what way physical laws explain our observations, making simulations on computers instead of resorting to actual physical experiments, which usually require heavy machinery or special instruments, as well as a tremendous amount of time, can drastically reduce the cost of the studies.

Numerical simulation can, in principle, be applied to any branch of physics, such as fluid dynamics, structural mechanics, electromagnetism, combustion, or wave propagation. The context of this dissertation is more specific, and focuses on applications relevant to Electricité de France (EDF) in Computational Fluid Dynamics (CFD), in particular Darcy flow in porous media and incompressible fluid mechanics.

1.2 On the way to more robust discretizations

Mathematical models of physical phenomena are usually expressed in the form of Partial Differential Equations (PDEs), whose resolution presents its fair share of mathematical challenges. One of them nourishes the active research field on *discretization* methods. Indeed, PDEs give rise to so-called *continuous* problems, whose unknown is a function belonging to an infinite-dimensional space. Unfortunately, except in very specific cases, the mathematical science in its current state does not allow to solve continuous problems in an exact manner, that is, exhibit an analytical formula describing the unknown function. However, it is possible to reduce the space of research to a finite-dimensional one, and find amongst its elements the closest one (according to a certain criterion) to the exact solution. This finite-dimensional space is called *discrete* space, as opposed to continuous, and is spanned by so-called *discrete*, or *approximate* solutions. The choice of the discrete space, as well as the properties enforced on the approximate solution, characterize the discretization method.

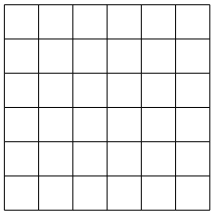
No one discretization method has been so far able to adequetely manage every PDE problem physics poses, and a broad spectrum of methods is therefore offered to engineers, according to the features of the problem they want to solve. In this thesis, we especially focus on the open issues of *complex geometries* and *non-smooth solutions* in CFD, which we tackle by considering the recently introduced Hybrid High-Order (HHO) methods [43].

1.2.1 Meshes

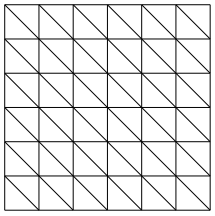
Solving a differential equation over a complex domain requires that the domain be accurately approximated by a mesh. Furthermore, that mesh must also enable an adequate representation of the solution. So, before discussing discretization methods, we shall enumerate the features a mesh can exhibit to help approximating a complex boundary and to improve the precision of the solution in the neighbourhood of a singularity. We preliminary introduce classical elements of terminology.

- **Mesh size.** Denoted by h, it corresponds to the largest element diameter and measures the "resolution" of a mesh.
- **Element shapes.** The elements partitioning the domain can have various shapes; see Figure 1.1. Whereas the simplest meshes are composed of Cartesian elements, less constrained meshes can be built using other shapes, such as triangles (resp. tetrahedra), quadrilaterals (resp. hexahedra), or general polygons (resp. polyhedra). It is also worth mentioning elements with curved edges (resp. faces), though we leave them out of our scope.
- **Mesh topology.** Adjacency relationships among elements, faces, edges, vertices.

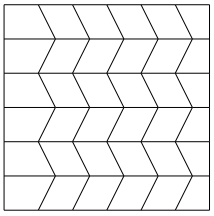
- **Mesh structure.** The mesh is called *structured* if the elements are arranged following a recognizable pattern (Figures 1.1a to 1.1c). The sole position of an element in that pattern then allows to deduce additional information, especially topological, that would otherwise have to be stored. If, moreover, all elements have the same shape, then the mesh is called *uniform*. Conversely, non-structured meshes are qualified as *unstructured* (Figures 1.1d and 1.1e).
- **Mesh quality.** Numerical methods are often sensitive to the presence of stretched, flattened or otherwise distorted elements. Each element T is then subject to a quality criterion in the form of an *aspect ratio* defined as $\varrho_T := d_T/h_T$, where h_T is the diameter of T and d_T the diameter of the largest ball embedded in T. This quality criterion is extended to the mesh through the maximum value of all ϱ_T, called *regularity parameter*. A mesh of good quality has a regularity parameter close to 1, while for a bad one it may be close to 0.
- **Local refinement.** In order to improve accuracy in specific areas, the mesh can be *locally refined*, meaning that a smaller mesh size is enforced in the corresponding part of the mesh. See Figure 1.2.
- **Hanging nodes and non-conformity.** Performing local refinement can lead to the onset of *hanging nodes*; see Figures 1.2a and 1.2b for a Cartesian mesh. A mesh node is qualified as such if it corresponds to one element's vertex while *hanging* on another's edge or face. As this particular configuration is often not natively managed by usual discretization methods, the presence of hanging nodes introduces the notion of *non-conforming* interfaces, and subsequently of *non-conforming* meshes. Notice that the flexibility of unstructured meshes allows mesh refinement without necessarily resorting to hanging nodes; see Figure 1.2c.



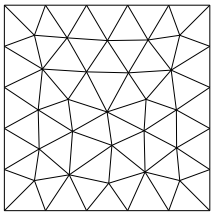
(a) Uniform, Cartesian



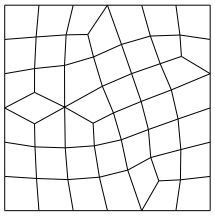
(b) Structured, triangular



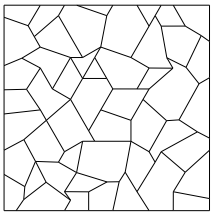
(c) Structured, quadrilateral



(d) Unstructured, triangular



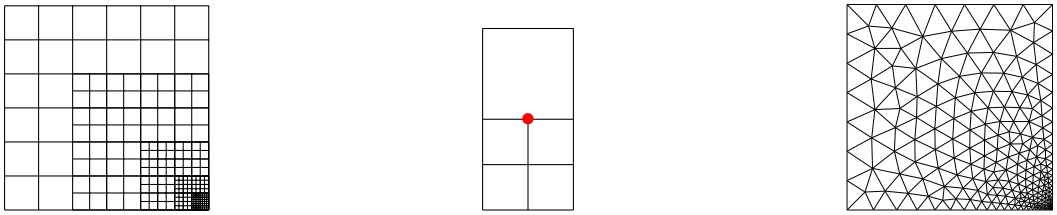
(e) Unstructured, quadrilateral



(f) Polygonal

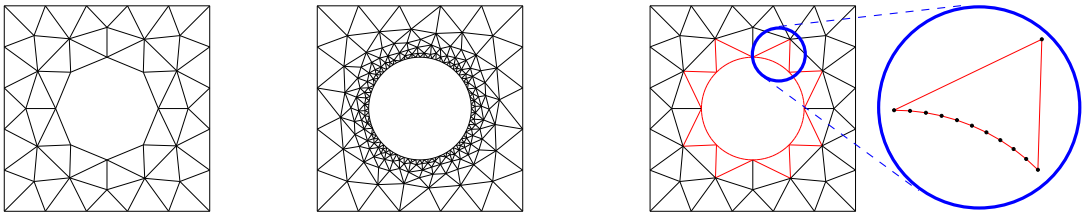
Figure 1.1 Mesh examples by element type and structure

Let us consider the example of a square domain embedding a circular hole. Taking advantage of their geometric flexibility, unstructured meshes constitute the most obvious and certainly the most popular approach to discretize such domain; see Figure 1.3a. In order to better approximate a curved boundary such as this one, the common approach is to locally refine the mesh around



(a) Local refinement with Cartesian elements (b) Hanging node (c) Local refinement with unstructured elements

Figure 1.2 Local refinement



(a) Unstructured mesh (b) ...with local refinement (c) ...with polygonal elements (in red)

Figure 1.3 Approximation of a complex boundary

the circular hole (Figure 1.3b). If more general elements are allowed, then another way to achieve the same geometric approximation is to use polygonal elements in the neighbourhood of the hole. The red polygons in Figure 1.3c present, at the boundary, the same number of small edges as generated by the local refinement of Figure 1.3b. Local refinement implies two drawbacks for a numerical approach: (i) an increase in the number of elements, which makes the computational burden heavier; (ii) the possible onset, especially in 3D, of streched or flattened elements, which may dramatically affect the convergence of the discretization or that of the linear solver. On the other hand, the use of polyhedral elements can achieve the same geometric accuracy without sensibly altering the mesh granularity, while offering more flexibility to preserve the aspect ratio of the elements, i.e. to build a high-quality mesh. Moreover, polyhedra allow the native management of hanging nodes and, more generally, non-conforming elements, by viewing usual element shapes, like tetrahedra or hexahedra, as identically shaped polyhedra with additional vertices (e.g., the top square of Figure 1.2b would actually be viewed as a square-shaped pentagon with two collinear edges).

Remark 1. *Although unstructured meshes offer more flexibility, structured ones are not to be discarded. Working with structured meshes may indeed offer compensations in terms of computation later in the process, such as matrix-free implementations due to constant stencils.*

1.2.2 Relevant features of discretization methods

Having stated (some) solutions to approximate complex domains and solutions with accuracy, the ability of discretization methods to handle such solutions is to be evaluated. We then briefly review the capabilities of classical and broadly used discretizations methods. As we identify their weaknesses, we introduce other, non-conforming methods exhibiting more suitable properties,

thus making our way to HHO discretizations. We emphasize that the goal is not to establish a comparison of discretization methods, but rather to introduce HHO to the reader through the obstacles met by well-known methods, which HHO and structurally similar methods can help overcome.

Given the previous discussion about approximating complex geometries, the desirable features we retain are the native support of non-conforming meshes, polyhedral elements, and higher-order of approximation. In the context of CFD applications, we also require continuity of the numerical fluxes.

The classical Finite Element and Finite Volume methods

The (continuous) Finite Element Method (FEM) [101] is certainly the most commonly employed method in the mechanical community. Its purely elemental point of view allowing unstructured meshes and local refinement, as well as the large collection of element shapes it can manage, have made it a tool of choice for a wide range of problems. Nonetheless, the global continuity properties of the approximation involve degrees of freedom (DoFs) shared between elements at interface nodes, implying a number of complications in the presence of hanging nodes or for mixing element shapes. The occurrence of a singularity also globally affects the convergence of the method. Another weakness of the standard FEM is its inability to enforce flux conservation at element interfaces, making it ill-suited to CFD applications. Note, however, that these remarks apply to the canonical H^1-conforming version of FEM. Mixed and non-conforming versions, less widespread, offer extended capabilities.

The Finite Volume method (FV) [58] on the other hand, introduces the computation of the fluxes at element interfaces and is therefore natively able to enforce their continuity across interfaces. While the classical Two-Point Flux Approximation scheme requires strict conditions on the mesh, Multi-Point Flux Approximation methods are more flexible. The DoFs, corresponding to the solution average within the cells, make the very notion of hanging nodes irrelevant, thus making non-conforming meshes admissible as well as polyhedral elements. However, these FV methods do not easily allow higher orders, especially on general unstructured meshes.

Discontinuous methods

Enforcing flux conservation demands careful control of the fluxes going through the interfaces. Additionally, the efficient approximation of non-smooth solutions requires to relax the regularity of the approximation near singularities. Both considerations plead in favor of the introduction of *non-conforming* methods in which element faces start playing a more significant role. We recall that a discretization method is referred to as *non-conforming* when the discrete space is not a subspace of the continuous one. As such, the Discontinuous Galerkin methods (DG) [46] reproduce the elemental structure of FEM while satisfying the equation in a way closer to FV. The method is based on spaces of piecewise polynomial functions that do not embed global continuity properties. Also allowing polynomial approximation at any arbitrary degree, the method can then be viewed either as a discontinuous FEM, or as a high-order FV method. The discontinuous setting hinges on the duplication of the DoFs located on the faces, so that the elements on each side can preserve independent control over them. However, that very duplication of DoFs, at the source of DG's robustness, is also considered as its main shortcoming,

insofar as it significantly increases the number of unknowns to solve.

Hybrid methods

The need to reduce the number of DoFs is addressed by modern hybrid methods. Indeed, in hybrid and hybridized methods, the cell unknowns are only locally coupled, i.e. coupled to the unknowns of the same cell or those of the associated faces. This allows for their local elimination from the global system, leaving the face unknowns as the only remaining ones in the resulting Schur complement. A first example of method whose DoFs verify this structural property is provided by the Hybridizable Discontinuous Galerkin (HDG) methods [39], directly originating from DG. A second one is HHO. See [36] and [43, Section 5.1.6] for the links between HHO and HDG. The basics of the mathematical construction of the HHO method are treated in Chapter 2.

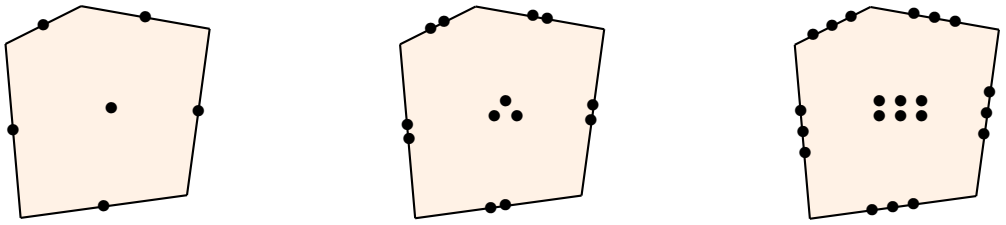
1.3 Fast linear solvers for HHO discretizations

1.3.1 Research subject

Hybrid discretizations have gained growing interest in recent years. In this thesis, we focus on HHO methods [43]. Amongst their key features, we can list the support of general polytopal meshes and of arbitrary approximation orders, as well as the optimal orders of convergence. Another built-in and defining feature of the HHO methods is the use, in the formulation of the bilinear form, of a higher-order potential reconstruction operator, which allows the gain of one additional order of approximation compared to, e.g., vanilla versions of HDG [36, 92]. Finally, the capability of HHO methods to adapt their design to the underlying physics, via problem-dependent local formulations, allows for more robust solutions with respect to the problem. To this day, HHO methods have been applied to a large variety of problems in fluid dynamics (heterogeneous anisotropic diffusion [48], incompressible Navier-Stokes [21], phase separation [34], creeping flows of non-Newtonian fluids [25], etc.) and structural mechanics (linear and nonlinear elasticity [23, 47] and poroelasticity [18, 24], etc.). Now that the method has gained sufficient maturity, its adoption for industrial applications depends on the availability of efficient linear solvers. The goal of this Ph.D thesis is to bridge this gap.

HHO methods hinge on DoFs located inside elements and on faces (see Figure 1.4), which can be globally viewed as broken polynomials respectively on the mesh and its skeleton (see Figure 1.5). We exclusively focus on cases where the element-defined DoFs are only locally coupled. As such, they can be expressed, element by element, in function of the DoFs on the faces, and subsequently eliminated from the global HHO linear system. This gives rise to a Schur complement of smaller size where only face unknowns remain. This process is known as *static condensation* in the mechanical literature, and the resulting system as a *statically condensed* system, or *trace* system, in reference to the mesh skeleton as the support for the set of globally coupled unknowns. The solution of the trace system, yielding the face unknowns, remains the costliest operation, after which the values of the element unknowns can be inexpensively recovered by solving small, independent linear systems.

As a consequence, the practical usefulness of HHO discretizations in an industrial context, where large problems have to be solved, depends on the existence of efficient linear solvers for the



(a) $k = 0$ (b) $k = 1$ (c) $k = 2$

Figure 1.4 HHO DoFs of a polygonal element for a polynomial degree k in the cell and on faces

Figure 1.5 Representation of the HHO DoFs as broken polynomials of degree $k = 1$ on the mesh (left) and its skeleton (right)

condensed system. In particular, the present research work is motivated by the aim to provide a solver for the open-source CFD software *code_saturne*[1] [7], developed and released by EDF. It is funded by the project Fast4HHO[2] of the French National Research Agency (ANR).

Our focus is on scalar elliptic equations, whose HHO discretizations give rise to trace matrices that are sparse, symmetric and positive-definite. Specifically, we aim at solving large systems of this type by means of a multigrid method [31, 118]. The main difficulty in the design of a geometric multigrid algorithm for trace systems resides in the location of the DoFs associated to the set of unknowns that remain after static condensation, namely, the face unknowns. Supported by the mesh skeleton, the broken polynomials defined by these DoFs are not suited for standard intergrid transfer operators, designed for *element*-defined functions. Hence the need for novel, *skeleton-based* multigrid methods.

1.3.2 Research objectives

We next state a list of criteria, relative to structure and performance, that a linear solver should exhibit to be considered as an adequate answer to the problem at hand. Besides the proper formalization of our research goals, this exercise will allow us to discuss existing solutions in light of these criteria, and therefore justify the need for new solvers and identify the gaps filled by this thesis.

We adopt the following objectives:

1. **HHO-compliancy.** More than the obvious criteria that the solver must be applicable to HHO systems, we want to emphasize that HHO should be its primary target, so that it can, if possible, take advantage of its defining features.
2. **Face-defined discrete spaces at all levels.** As it applies to a linear system with face

[1] www.code-saturne.org
[2] under contract ANR-17-CE23-0019

unknowns, the solver should comply with this specific setting and hinge on skeleton-based discrete spaces at every level.

3. **Applicability to general polyhedral meshes.** As the HHO discretization applies to such general meshes, this should also be the case for the solver.
4. **Algorithmic optimality.** Primary goal of every multigrid method, and most important property in order to achieve scalability and tackle large scale problems, the solver should exhibit a convergence rate that is independent of the number of unknowns.
5. **Fastness.** Whereas the previous property is purely qualitative and refers to the asymptotic behaviour of the solver, its practical usability also imposes two quantitative constraints in order to ensure an acceptable time to solution: (i) convergence speed: the solver should reach convergence after a "reasonable" number of iterations; (ii) iteration cost: the computational work of an iteration must also remain "under control".
6. **Robustness with respect to the polynomial degree.** Both of the above properties must also hold for higher polynomial degrees of approximation (if not all, at least for moderate values of practical interest).
7. **Robustness with respect to non-smooth solutions.** Reflecting the robustness of the discretization, the solver should display good performance when the solution is not smooth. Particularly, we consider in this work heterogeneous diffusion problems with highly discontinuous coefficient.

While the criteria relative to performance and robustness are classical for a multigrid method, the second one is arguable and deserves to be discussed, insofar as it may arbitrarily discard otherwise valid and possibly efficient methods for philosophical reasons. Indeed, we adopt the committed position that a solver should embrace the specificities of the problem to be better suited, i.e., frontally tackle the difficulty, rather than attempt to transform the problem into one for which solutions are already known. By taking this stand, we explicitely want to discard strategies that convert face-defined functions into continuous finite element functions in order to make use of a classical FEM solver.

Figure 1.6 illustrates the first issue underlying Criteria 2, i.e. the construction of a skeleton-based prolongation operator. On the left, a given skeletal coarse function on a 4-by-4 Cartesian grid. On the right, an illustration of an adequate representation of that coarse function on the fine skeleton of an 8-by-8 grid, after application of a suited prolongation operator. As all coarse edges are geometrically composed of two fine ones, the corresponding local coarse polynomials can be straightforwardly injected into the respective fine local spaces. However, there are fine edges absent from the coarse grid, namely, those that are geometrically embedded in coarse elements. The expected result on those fine edges is represented in dashed line. The issue can then be posed in these terms: how to reconstruct those dashed lines, given that no data exist at their locations on the coarse skeleton?

1.3.3 State of the art

Before reviewing existing solvers applicable to trace systems such as our own, we briefly report recent progress in multigrid algorithms for non-conforming methods.

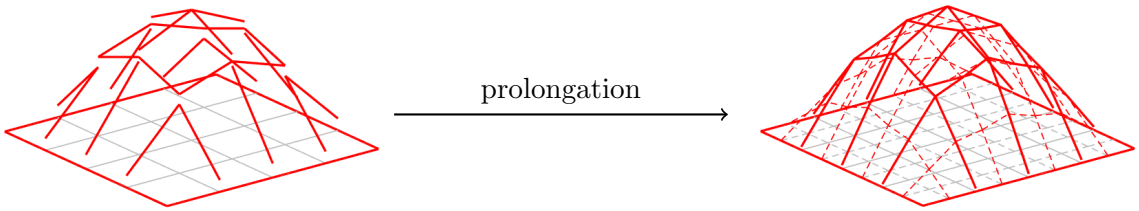


Figure 1.6 Illustration of the issue underlying the construction of a skeleton-based prolongation operator. How to reconstruct the dashed lines?

1.3.3.1 Multigrid for non-conforming methods

Let us begin with DG discretizations. As the lowest order case of DG can be interpreted as other discretization methods such as FV, multigrid algorithms have been designed for this case over the 1990s when working on those discretizations. Regarding the higher order cases, multigrid methods dedicated to DG have emerged in the early 2000s with [13, 68], which introduced the first h-multigrid algorithms for Interior Penaly (IP) formulations. At the same time, [70] performs a Fourier analysis on the matrix arising from the DG discretization of the 2D Poisson problem. Since then, p and hp-multigrid methods such as [4–6, 20, 60, 62, 91, 111] have especially retained attention. Regarding algebraic multigrid methods (AMGs), the Ph.D thesis [53] introduces the first algebraic multigrid algorithm for the solution of a DG system. Since then, other algebraic solvers have also been designed: [12, 17, 98, 100], or, more recently, [3], all based on smoothed aggregation.

We also mention recent advances on multigrid methods targeting other non-conforming discretizations and/or focusing on problems our research subject takes a special interest in, like diffusion in porous media or Darcy flows. As such, [9] proposes multigrid methods for Darcy–Forchheimer flow in fractured porous media. [10] presents a solver for Multipoint Flux Mixed Finite Element (MFMFE) schemes — based on the Brezzi-Douglas-Marini (BDM) framework — applied to the Darcy problem. [114] tackles the Isogeometric Analysis (IGA) discretizations, which are based on spline-type functions.

1.3.3.2 Trace system solvers

A literature review of trace system solvers shows the variety of paths one can follow to tackle this particular setting. Although no existing geometric h-multigrid method has specifically targeted HHO so far, a few trace system solvers — generally targeting HDG — have been designed over the last years. As HHO hinges on a comparable set of DoFs, these methods are supposedly applicable. With respect to Criterion 1, we, however, point out that no test report accounts for their performances on HHO, and a fortiori, no specific adaptation to HHO systems has been attempted.

In [37], the authors propose a geometric multigrid solver for general HDG discretizations of elliptic equations and low order approximation, where the face-defined functions are recast, via the adjoint of a trace operator, into globally continuous functions defined over the elements. This conversion then allows to make use of a known efficient solver, typically a standard piecewise linear continuous FEM multigrid solver. This special multigrid method actually takes its origin

from the one previously designed for hybridized versions of the Raviart–Thomas and Brezzi–Douglas–Marini methods in [69], from which the intergrid transfer operators are borrowed. A variation using an underlying algebraic multigrid method instead of a geometric one was experimented in [80]. These methods do not comply with Criterion 2.

A different approach is suggested in [123], where an *hp*-multigrid algorithm based on face unknowns at every level (therefore meeting Criterion 2) is proposed for HDG discretizations. It handles unstructured polyhedral meshes and is based on the use of Dirichlet-to-Neumann (DtN) maps to preserve energy from coarse to fine levels. DtN maps perform, between each level, the static condensation of the unknowns located on the fine faces interior to coarse elements. The management of high orders is carried out in the traditional way of putting a p-multigrid algorithm on top of a multigrid iteration in h. This fundamentally novel and otherwise simple, clever approach reports good performance and a scalable behaviour. However, the multigrid cycle is enhanced by the addition of a local subspace correction method at every level, and the number of smoothing steps performed increases as the levels coarsen, which tends to mask the performance of the method with a plain, standard multigrid cycle such as, e.g., V(1,1). Without going into details, our in-house implementation of this algorithm reports a convergence with a significant dependence on the mesh size with standard multigrid ingredients and on an HHO system. This violates Criterion 4.

Also working with face unknowns at every level, [103] works in the context of Discontinuous Petrov-Galerkin (DPG) discretizations. It relies on the natural idea of reversing the static condensation at the coarse level to recover cell unknowns, before taking the trace of the corresponding polynomials on the local interior fine faces. The recent convergence analysis [86] proves, for the Poisson problem, the optimal asymptotic behaviour of multigrid algorithms for HDG discretizations based on decondensation of the cell unknowns and trace on the fine faces. Reversing the static condensation is also the approach we have adopted in our first and main contribution. We however indicate that our work has been conducted independently of [86, 103] (with [86] still in preprint state to present date). Notice also that, despite this common starting point, our own contributions still differ in numerous ways, going from the HHO-dedicated enhancement to the enlarged spectrum of application and the improved robustness, as we tackle more complex problems and meshes.

Before closing this section, we also want to point out the efforts made to design p-multigrid preconditioners for non-elliptic equations: one can cite [65, 109], and especially [22], inasmuch as it directly targets HHO.

Finally, besides multigrid, other types of large scale solving methods devised for HDG deserve to be mentioned, such as domain decomposition [108, 119] and nested dissection [88].

1.3.4 Contributions and thesis outline

We next provide an executive summary of the manuscript highlighting the main contributions.

1.3.4.1 Model problem and HHO discretization

Chapter 2 is dedicated to the application of the HHO method on a scalar diffusion problem including a possibly anisotropic and heterogeneous permeability tensor (see formulation (2.1)). We introduce the *high-order potential reconstruction operator* (cf. (2.11)), which is the main ingredient in the definition of the discrete bilinear form. This operator, based only on an integration by parts formula, is locally defined. It allows, from a polynomial in the cell and polynomials of same degree on the faces, to reconstruct a polynomial *one degree higher* in the cell. This feature is advantageously employed in our geometric multigrid method to enhance its overall performance.

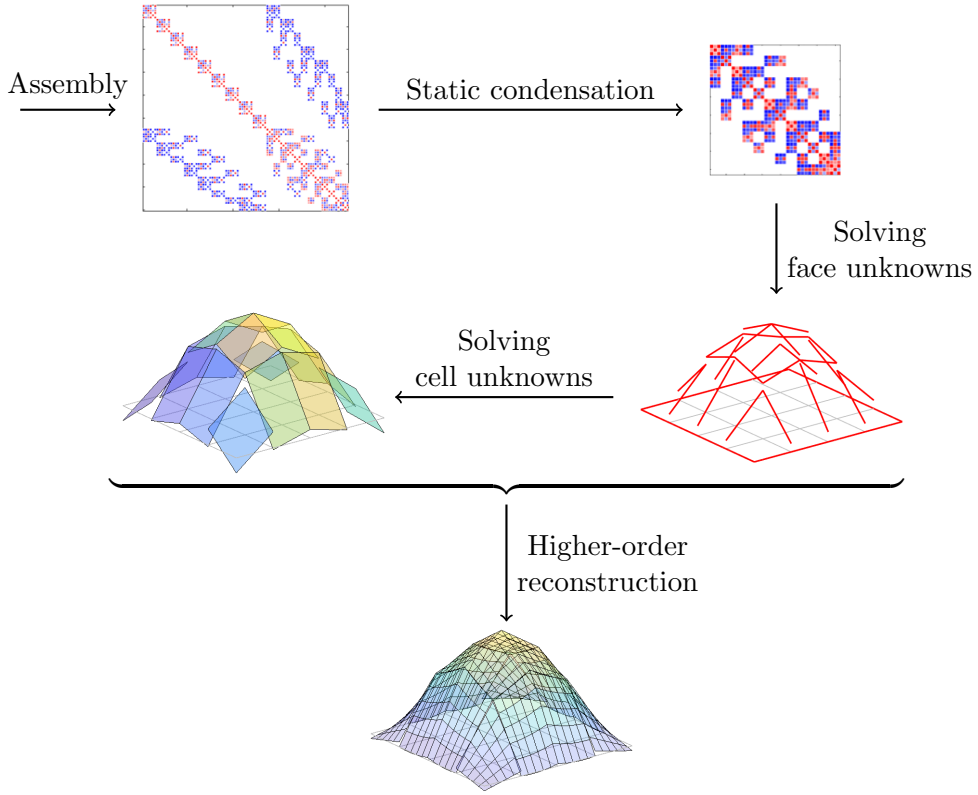


Figure 1.7 Summary of the HHO process

The general HHO process, from assembly to the reconstruction of the discrete approximation is summarized in Figure 1.7. Ordering the unknowns so that cell-defined ones come first and face-defined ones come last, the global *hybrid* matrix is a 2-by-2 block matrix, whose block (1, 1) is block-diagonal, owing to the local coupling of cell unknowns. Locally eliminating those unknowns, the static condensation gives rise to a smaller system relying on the face unknowns only. The following step, where this *condensed* system must be solved, marks the location of our contributions within the HHO process. The solution represents a broken polynomial on the mesh skeleton. The values of the face unknowns are then used to decondense the cell unknowns

and recover their values. Finally, in a post-processing step, by applying the high-order potential reconstruction to both cell and face unknowns, a broken, element-defined polynomial of one degree higher is reconstructed, thus yielding the final HHO approximation of the scalar potential.

1.3.4.2 A geometric h-multigrid method

Chapter 3 is devoted to the first original contribution of this Ph.D thesis, published in *SIAM Journal on Scientific Computing* [50]. In this contribution, we develop a novel, geometric h-multigrid algorithm

- based on approximation spaces supported by the mesh skeleton at every level,
- targeting HHO discretizations by making use of the underlying high-order potential reconstruction,
- natively handling higher orders (as opposed to, e.g., putting a p-multigrid on top of an h-one).

To handle the issue raised by Figure 1.6, the method relies on the design of a special prolongation operator that includes the construction of an intermediary state between the coarse skeletal function and its prolongation onto the fine skeleton. Precisely, a *cell-defined* potential is reconstructed on the coarse mesh, which allows, via a trace operator, a subsequent definition on the fine skeleton. Figure 1.8 illustrates these steps.

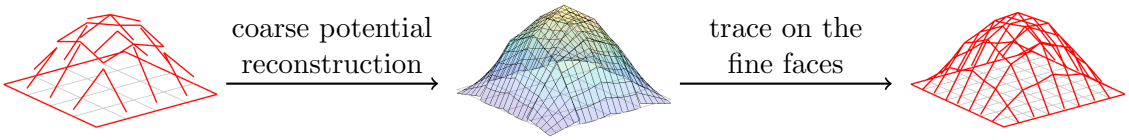


Figure 1.8 Prolongation from coarse to fine faces

The cell reconstruction is the core of our method and what makes it original. It works locally, and decomposes into two steps illustrated by Figure 1.9. First, a coarse cell-defined polynomial of degree k is recovered from the face-defined polynomials of degree k through the decondensation of the cell unknowns. Second, the higher-order reconstruction operator is applied to both cell and face unknowns in order to gain one degree of approximation in the cell.

Given that the reconstructed polynomial is of degree $k+1$, recovering the original polynomial degree k on the fine faces implies that the trace operation in the second step of Figure 1.8 must also lower the degree. To do so, the trace comes with a subsequent L^2-orthogonal projection onto the polynomial space of lower order k. Moreover, on the fine faces at the boundary of coarse elements, due to the discontinuous setting, the trace actually consists in taking the weighted

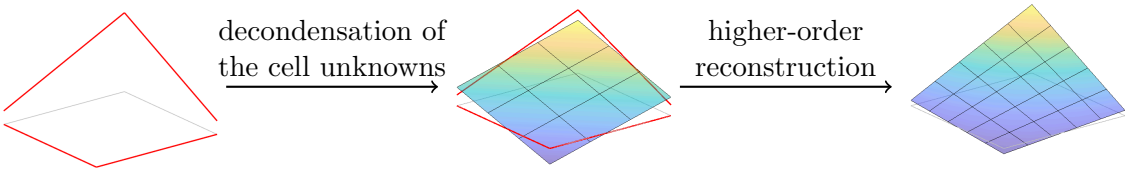


Figure 1.9 Reconstruction of a polynomial of degree $k+1$ from polynomials of degree k (here for $k=1$) on the four edges of a 2D square element

average of the traces computed on each side. The weights incorporate a dependence on the diffusion coefficient to smartly favor one or the other side according to the size of the possible jump. This allows to ensure the solver's robustness to the coefficient discontinuities. Notice that our solution does not leverage the natural injection from the coarse faces to the embedded fine ones, which our numerical experiments find less efficient, both in terms of convergence rate and of robustness to discontinuities.

Notice that the prolongation operator preserves the polynomial degree. Consequently, no p-multigrid method has to be involved. Instead, the same polynomial degree is preserved at every level, at the sole cost of using a blockwise smoother, where the block size is determined by the number of unknowns per face, so that all unknowns relative to the same local polynomial be relaxed together.

The numerical tests include homogeneous and heterogeneous isotropic problems in 2D and 3D domains, discretized by structured and unstructured meshes. With structured (Cartesian or simplicial) meshes on simple domains, the multigrid method, directly used as a solver, meets all the criteria listed in Section 1.3.2 but the third:

- convergence in a limited number of iterations, seemingly independently of the mesh size; see Figure 1.10;
- controlled computational cost through the rediscretization of the operator at the coarse levels and the use of standard smoothers (block Gauss-Seidel or Jacobi);
- robustness with respect to discontinuities of the diffusion coefficient, whose magnitude does not alter the convergence rate;
- robustness to higher orders, for which the solver exhibits the same properties.

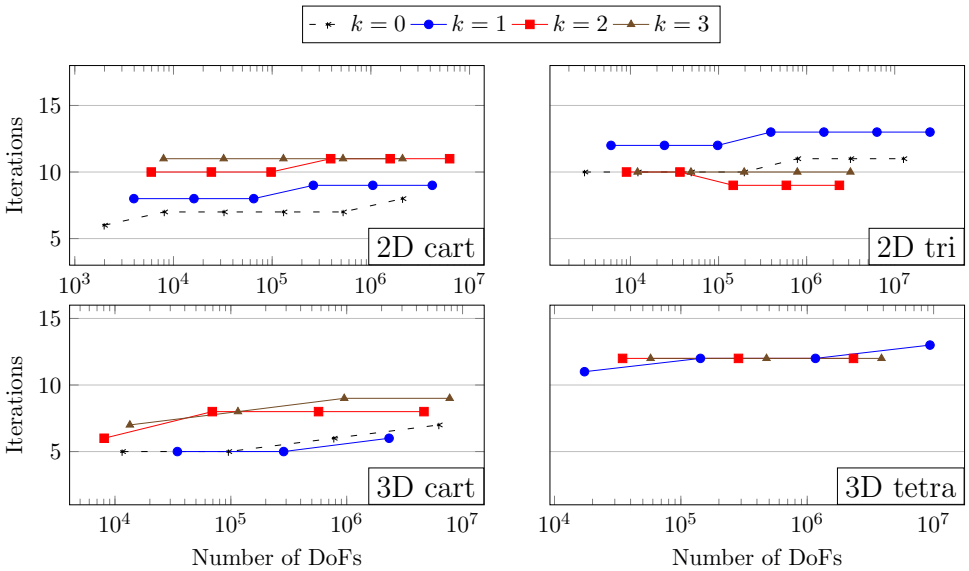


Figure 1.10 Number of iterations to reach a normalized residual of 10^{-8} for the homogeneous diffusion problem on structured meshes: 2D Cartesian (top left), 2D triangular (top right), 3D Cartesian (bottom left), 3D tetrahedral (bottom right). Refer to Section 3.2 for the details about the experimental setup.

However, on complex domains requiring highly unstructured meshes, optimal convergence is not achieved in general. The reason is twofold:

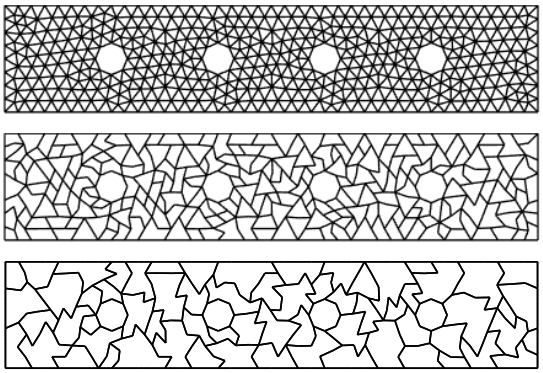


Figure 1.11 Nested coarsening strategy by agglomeration

- optimal convergence relies on the *faces* being coarsened between levels (not only the elements!);
- numerical experiments have shown the high sensitivity of the multigrid method to the mesh quality, i.e. to the presence of elements with bad aspect ratio. Optimality then also requires a hierarchy of high-quality meshes.

These demands may work against each other. Combined, they raise the issue of how to build the mesh hierarchy. Indeed, multigrid hierarchies are commonly constructed by successive refinements of an initial coarse mesh. If this ensures in a natural way the desired face coarsening between every level, it can also affect mesh quality, especially in 3D. Numerical experiments with Bey's tetrahedral refinement method [16] on a complex domain have shown that the obtained fine mesh was not of good enough quality for our multigrid solver to converge efficiently. Conversely, starting from a good quality fine mesh, there is no obvious method allowing to construct a nested coarse mesh while also enforcing face coarsening. For instance, Figure 1.11 illustrates the nested coarsening of an unstructured triangular mesh by an agglomeration method. Note, in particular, that the faces are not coarsened: fine faces remain at all levels. Such a hierarchy is not, then, considered admissible for our multigrid method. Conserving nested meshes while coarsening faces indeed requires coplanar fine faces (colinear edges in 2D) to be merged to form coarse ones. Unstructured meshes do not generally offer many of such opportunities. Nonetheless, we observe that if two fine faces are only *nearly* coplanar, they can still be merged to form one single, larger face for the coarse mesh, and that this approximation technique does not significantly affect the convergence of the solver. See Figure 1.12 for an admissble 2D example. This remark paves the way to non-nested coarsening strategies using face collapsing to coarsen the faces.

1.3.4.3 Extension to non-nested meshes

Non-nestedness is the path we follow in our second contribution to overcome the limitations of our multigrid method and successfully manage untructured 3D cases. Chapter 4 is then dedicated to the adaptation of the nested version of our algorithm to non-nested mesh hierarchies and its efficient implementation for practical use. Its scientific content was published in the *International Journal for Numerical Methods in Engineering* [51].

Compliance to non-nested settings is performed by inserting an additional step in the

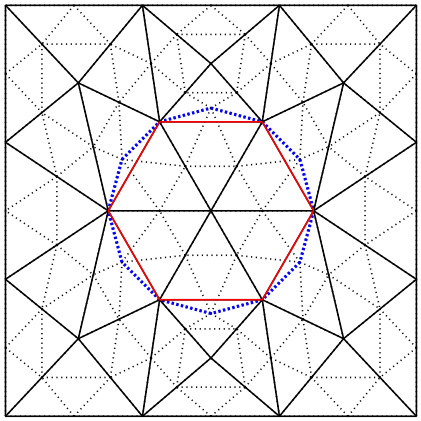


Figure 1.12 Coarsening of nearly colinear edges. The fine mesh is represented in dotted lines, the coarse mesh in solid lines. The non-nestedness is highlighted by colors: the blue fine edges are coarsened into the red ones.

definition of the prolongation operator. Starting with polynomials lying on the coarse faces, the nested version begins with the reconstruction of a broken *element*-defined polynomial on the coarse mesh. This step is unchanged. We then propose to orthogonally project in L^2-norm this coarse broken polynomial onto the non-nested fine mesh. Finally, the end of the process also follows the nested version: the trace of the result is computed on the fine faces.

An approximate L^2-orthogonal projection operator

The numerical evaluation of this L^2-orthogonal projection operator hinges on the projection of the local coarse basis functions onto the fine bases, i.e., on the computation of the L^2-inner products of the coarse and fine basis functions over the fine elements. As a direct consequence, the local definition of the functional bases makes the intersections of coarse and fine elements the respective integration supports to these inner products. However, computing the geometric intersections between coarse and fine elements can be prohibitive. So, instead of this exact computation, we propose the implementation of an approximate operator that does not require the explicit computation of intersections. It is based on the subdivision of the fine elements, by adopting the simplifying hypothesis that each sub-element is fully included in the coarse element that contains its barycenter. Formally, it translates to the following: assuming, for any fine element T_f, a given subdivision $\texttt{Sub}(T_f)$, we then define, for any pair of coarse and fine elements (T_c, T_f), the approximate intersection

$$T_c \cap T_f \approx \bigcup_{\substack{t \in \texttt{Sub}(T_f) \\ \texttt{barycenter}(t) \in T_c}} t \,.$$

Figure 1.13 illustrates, on coarse and fine Delaunay meshes, the accuracy of the method according to the subdivisions of the fine triangles: in (a), no subdivision is performed, i.e. $\texttt{Sub}(T_f) = T_f$ for all T_f; in (b), sub-triangles are obtained by connecting the middle-edges of the fine elements. The fine elements can be refined multiple times for even better accuracy, though at the cost of an increasing number of integrals to compute. In practice, the approximate operator derived from

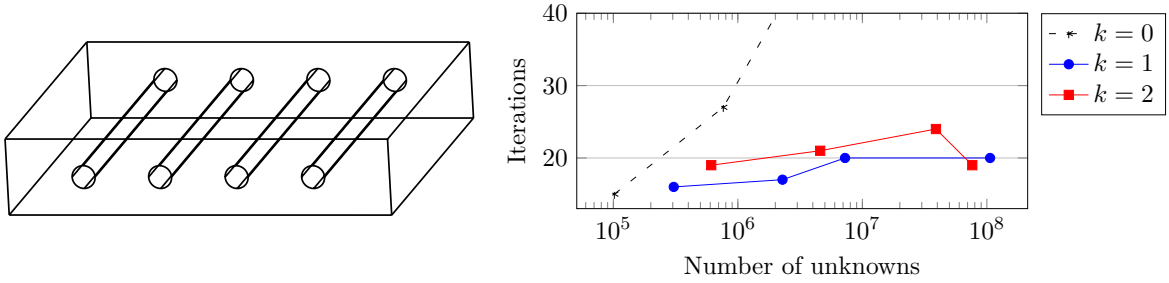
(a) No subdivision

(b) Sub-triangulation obtained by connecting the middle-edges of the fine elements

Figure 1.13 Distribution of the fine elements' sub-triangles to their "closest" coarse element, i.e. the one containing their barycenter. Without actual subdivision, (a) then shows the superposition of the coarse and fine triangular meshes. The fine triangles are colored according to the coarse one that contains their barycenter. (b) shows the same colored partioning, this time for a standard triangulation of the fine elements.

the fine elements being subdivided only once is found to be sufficient to achieve good multigrid results in 3D and for low polynomial orders.

We evaluate the accuracy of this approximation by comparing with the exact operator, and we assess the convergence of our multigrid method using non-nested meshes obtained by independent retriangulation of the domain at each level. These tests demonstrate the sufficient accuracy of the approximation for moderate polynomial degrees in 3D, as well as the substential gain in setup time that the technique offers by avoiding the computation of geometric intersections. In particular, we numerically demonstrate the optimal convergence of our non-nested multigrid algorithm on an unstructured 3D test case that the nested version failed to solve; see Figure 1.14.



(a) Plate with four cylindric holes

(b) Algorithmic scalability plot

Figure 1.14 3D test case using the geometry (a). The algorithmic scalability plot (b) of the solver is obtained with the coarse meshes independently retriangulated at each level, and the L^2-orthogonal projection computed approximately using Bey's method to subdivide the fine tetrahedra. (Note that $k = 0$ is a special case, for which optimality is not necessarily achieved. It was already so with the nested multigrid method on structured meshes.)

A polyhedral coarsening strategy

In practice, building a high-quality mesh for a real industrial case study can be a challenging task, which may occupy a meshing engineer for several months. Requiring multiple high-quality

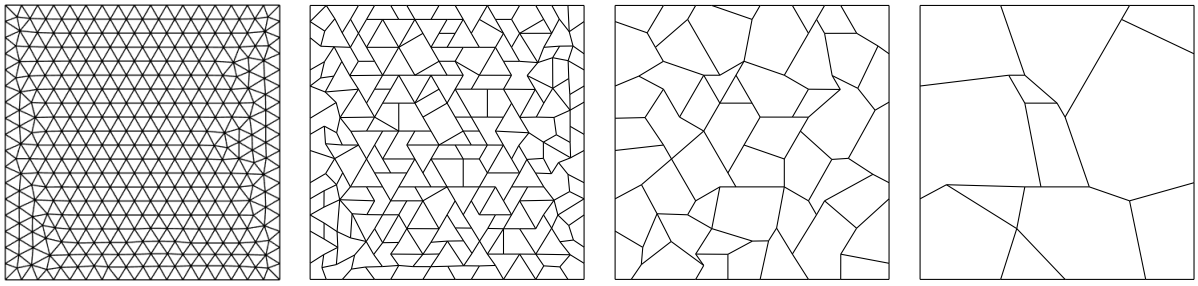


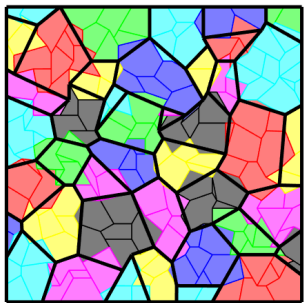
Figure 1.15 Successive coarsenings of a triangulated square by agglomeration and interface collapsing

meshes of the same geometry at different granularities in order to feed a multigrid solver is then not always conceivable. From the user's standpoint, providing the solver with the sole fine mesh is a preferable option. To this end, we also want to pave the way for the construction of suitable coarsening strategies for this type of multigrid method. Therefore, this chapter also includes the abstract definition of a coarsening strategy in order to build, from a given fine mesh, a hierarchy of non-nested coarse meshes in which faces are coarsened. In particular, the method is based on element agglomeration, to which we add a step of face collapsing at the interfaces between agglomerated elements. Figure 1.15 shows the result of such a strategy on a triangulated square. Given that coarse operators come from rediscretization, this strategy based on agglomeration is made possible by HHO being a polytopal method. We provide explicit details about our implementation in 2D, explaining how the the approximate L^2-orthogonal projection can be made exact through a clever way of subtriangulating the fine elements; refer to Figure 1.16 for more details. The non-nested multigrid method resulting from this coarsening strategy is finally evaluated on the simplified geometry of a real industrial test case provided by EDF, which also results in an asymptotically optimal behaviour.

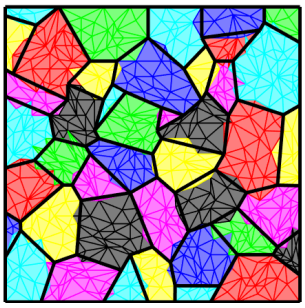
1.3.4.4 Algebraic multigrid

The geometric multigrid algorithm and its non-nested extension that we have devised provide a first option for the solution of HHO systems. In Chapter 5, we develop another approach, in the form of an Algebraic Multigrid method (AMG). This contribution gave rise to the submitted preprint [49].

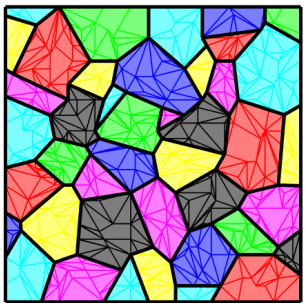
AMG solvers [59, 112] are very popular for the solution of large scale linear systems arising from the discretization of elliptic equations on unstructured meshes. Unlike geometric multigrid methods, which require a hierarchy of meshes of different granularity, algebraic algorithms classically do not need more information than the linear system to solve. Discarding all geometric information as input parameter results in the most appreciated feature of these methods, that is, their usability in a black-box fashion. Adopting a new discretization in an industrial context requires heavy preliminary testing, that can be facilitated if the software for the appropriate solver is already available on the market or if its development can easily be externalized. Being isolated from the mesh, which can be generated, stored, and transferred in numerous ways, AMG solvers ally interoperability and performance. Additionally, they are generic solvers that can be used in various ways and for many advanced problems. They are



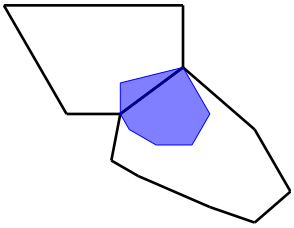
(a) Distribution of the fine elements to their "closest" coarse element.



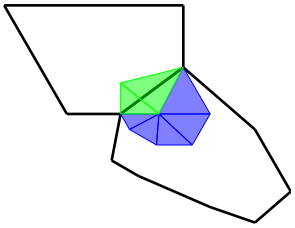
(b) The fine elements are subtriangulated by a barycentric method.



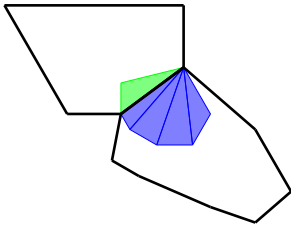
(c) The fine elements are subtriangulated such that they do not cross any coarse element's edge.



(d) Zoom-in on a fine element overlapping two coarse ones.



(e) Barycentric triangulation.



(f) Adapted triangulation preventing subtriangles from overlapping two coarse elements.

Figure 1.16 The top figures show how the fine polygons (in (a)) or their subtriangulations (in (b),(c)) are clustered in the process of approximating the coarse/fine intersection involved in the computation of the L^2-orthogonal projection. The coarse edges are represented by thick black segments. In (b), the fine elements are triangulated by a barycentric method. In (c), the triangulation is adapted to prevent subtriangles to overlap multiple coarse elements. The bottom figures zoom in on a fine element overlapping two coarse ones. In (d), the whole fine element is affected to one of them for the computation of the approximate L^2-orthogonal projection. In (e) and (f), the subtriangles are dispatched on one or the other coarse element according to the location of their barycenters.

particularly successful in the context of parallel computing [11], and can be employed as coarse solver in another multigrid method [113] or in connection with other techniques such as domain decomposition [76, 77].

Usual AMG solvers designed for low-order finite element or finite difference methods deduce mesh information under the assumption that each row in the matrix corresponds to a DoF located at a *mesh node* or *element*. Thus, the mesh connectivity graph can be reconstructed algebraically, and coarsening strategies mimicking geometric algorithms can then be performed in order to build the coarse levels. Especially focusing on aggregation-based methods, nodes/elements are being aggregated in order to give rise to coarse DoFs. However, in our hybrid setting at the lowest order, the unknowns of the system are actually linked to faces, i.e. neither nodes nor elements. Consequently, at first glance it might seem peculiar, from a geometrical point of view, to apply the above approach in this context. Indeed, aggregation-based coarsening can then be interpreted as aggregating *faces*. Although it may give natural results for adjacent faces, especially if they are close to being coplanar, it sometimes aggregates faces that do not even touch. In this case, it is difficult to perceive a geometrical sense in this aggregation. Nonetheless, numerical tests with a standard aggregation-based AMG method show that the approach still works well, which can be geometrically justified by forgetting about the DoFs being actually face-defined and considering them as mere nodal values located at the center of the faces. That being said, one can legitimately wonder if a coarsening strategy making geometrical sense in light of the actual meaning of the DoFs as face-defined values could not yield even better results.

Restricting our scope to *lowest-order* hybrid methods (not only HHO), the idea at the origin of this work is the algebraic reconstruction of the mesh topology based no longer on the condensed matrix, but on the uncondensed one. Indeed, like traditional AMG methods, we retrieve geometric information on the coupling of the DoFs from algebraic data. However, as the condensed matrix only provides information on the faces, we use the *uncondensed* version to reconstruct the connectivity graph between elements and faces. Note that it means that this method requires more information than the sole system to solve. Parts of the uncondensed matrix must indeed be brought to the algorithm as additional information. This makes the method less "black-box", but still purely algebraic. Once the so-called algebraic mesh is retrieved, especially the neighbouring information between elements, an *element*-based aggregation method can be set up in order to mimic the behaviour of a geometric coarsening or semi-coarsening strategy. Keeping in mind that, in our hybrid setting, *faces* must be coarsened between levels, we complement the element aggregation with the face collapsing technique devised in Chapter 4. The construction of the intergrid transfer operators follows plain aggregation principles and leverages the decondensation of the cell unknowns already at the source of the special prolongation operator devised in our geometric multigrid method.

AMG methods directly used as solvers may lack efficiency; see, e.g., [122, p. 663] or [89]. Using them as preconditioners for a Krylov method is generally favored. To build an efficient solver, we adopt the choices made by AGMG [94]. Namely, we use the so-called K-cycle, which introduces Krylov acceleration into the multigrid recursive cycle. Secondly, one such cycle is used to precondition an outer Krylov method. More generally, the technical choices made in this work are borrowed from AGMG (pairwise aggregation, strong negative coupling criterion, K-cycle...) in order to establish a proper comparison with a standard AMG solver that works

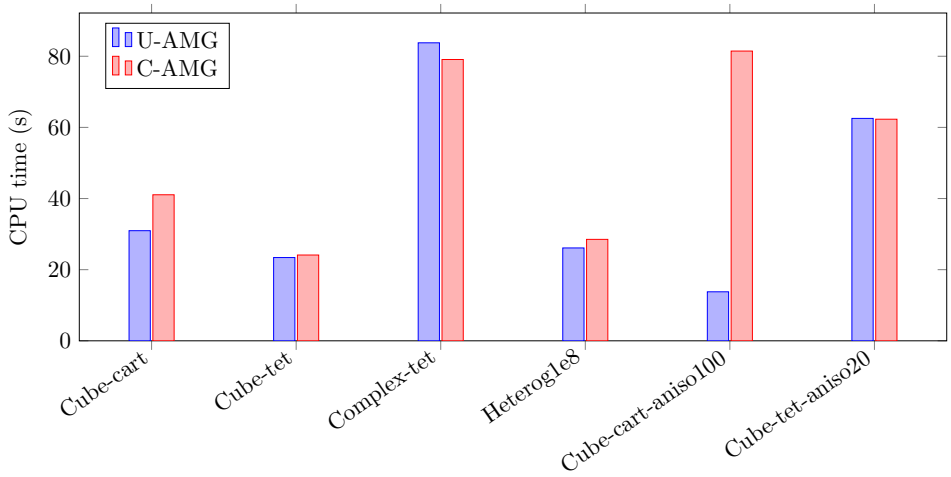


Figure 1.17 AMG solver comparison in CPU time. Refer to Table 5.1 for the details of the test cases, all comprising several million unknowns.

only on the condensed system.

Our method is applied to the lowest order HHO discretizations of 2D and 3D diffusion problems. The tests include homogeneous, heterogeneous, isotropic and anisotropic problems on structured Cartesian and unstructured simplicial meshes. The methodology adopted compares our novel method, denoted by U-AMG (standing for *uncondensed* AMG), to a standard aggregation-based AMG, denoted by C-AMG (standing for *condensed* AMG). C-AMG views DoFs as nodes and implements a node-defined coarsening strategy from the condensed system, while U-AMG reconstructs the elements from the uncondensed matrix and implements an *element*-defined coarsening strategy. The agglomeration criteria, cycle, smoothers, as well as every other technical choices are identical for both solvers. The results of this comparison are provided in Figure 1.17. We report equivalent performances in isotropic and in unstructured cases. The added value of the new algorithm actually appears in anisotropic problems with Cartesian meshes, where the solver exhibits an enhanced robustness (test case `Cube-cart-aniso100`). The element-based aggregation strategy enables one to take advantage of the Cartesian structure of the mesh to follow the direction of anisotropy, what a *node*-based strategy only succeeds to a lesser extent. Although this very specific, trivial test case might seem restrictive, this feature can actually be exploited in a larger range of applications. Namely, U-AMG can offer substantial added value for solving problems comprising both isotropic and anisotropic regions, providing that the anisotropic ones are discretized by Cartesian elements oriented in the direction of anisotropy. This includes orthotropic diffusion problems. The solver, in this case, can ally the flexibility of AMG to handle unstructured meshes on isotropic regions while exploiting the special element shapes on anisotropic ones.

1.3.4.5 Scientific communications

The contributions of this Ph.D thesis have been made available to the scientific community through journal papers, open-access preprints, and talks at international conferences. References and download links are gathered at the end of the dissertation, page 103.

Chapter 2

The Hybrid High-Order method

Nature laughs at the difficulties of integration.

Pierre-Simon de Laplace (1749-1827)

Originally introduced in [44] (see also [45] and the monograph [43]), HHO methods hinge on discrete unknowns that are broken polynomials on the mesh and its skeleton. The adjective *hybrid* refers to their union, in spite of their different natures, to form one set of unknowns. Moreover, they are designed so that element-based unknowns are not directly coupled with each other. As a result, the corresponding DoFs can be efficiently eliminated from the linear system by computing a Schur complement element by element, a procedure known in the mechanical literature as *static condensation*. The discrete solution can then be obtained in two steps: first, the Schur complement system is solved, yielding the values of the face unknowns; second, cell unknowns are recovered element-wise by solving a small local system. This second step is inexpensive inasmuch as it can be parallelized, leaving the first step as the costliest operation. Consequently, the problem matrix in the context of hybridized methods is usually the Schur complement matrix obtained after static condensation, also called *trace, statically condensed*, or sometimes *Lagrange multiplier system* (referring to the interpretation of face unknowns as Lagrange multipliers enforcing a discrete flux continuity constraint, see [43, Section 5.4.6]). For a more detailed introduction to hybridization, we refer the reader to the first pages of [39] and also [43, Appendix B.3.2].

The defining feature of HHO methods is the embedding of a higher-order potential reconstruction into the definition of the discrete bilinear form. As a result, up to one order of convergence is gained with respect to other hybrid methods [38, 40]; see, e.g., the discussion in [36] and also [43, Section 5.1.6]. Once the values of the discrete unknowns, defining local polynomials of some fixed maximum degree in cells and on faces, have been found, the application of the potential reconstruction in a post-processing step reconstructs a discrete approximate that is one degree higher.

In this chapter, we adopt a constructive approach to describe the HHO discretization of scalar elliptic equations. Thus, after introducing general notations in Section 2.1 and the variable diffusion equation as model problem in Section 2.2, we introduce the fundamental components of

the HHO methods in Section 2.3. In particular, a local approximation of the continuous solution can be defined through the projection of its gradient onto a polynomial space and the choice of the missing constant by a closure condition. This so-called *elliptic projection* can, in fact, be computed from the simple L^2-orthogonal projections of the solution in the cell and on its faces, through the application of a special *higher-order reconstruction operator*, cornerstone of the method. We close this chapter in Section 2.4 with a list of theoretical results regarding the convergence, robustness and computational properties of the HHO methods, as well as some *take-out* ideas for the comprehension of the work described in the next chapters.

2.1 Notation

2.1.1 Domain and mesh

Let $d \in \{2,3\}$ be the space dimension and Ω a bounded polyhedral domain of $\mathbb{R}^d$. We consider a mesh $(\mathcal{T}_h, \mathcal{F}_h)$ of Ω in the sense of [43, Definition 1.4], with $\mathcal{T}_h$ denoting the set of open polyhedral elements, $\mathcal{F}_h$ the set of faces, and $h := \max_{T\in\mathcal{T}_h} \mathrm{diameter}(T)$ the mesh size. Meshes of practical relevance included in this definition correspond to decompositions of the domain into polyhedra not necessarily convex or star-shaped, and possibly including hanging nodes. The set $\mathcal{F}_h$ is partitioned as $\mathcal{F}_h^{\mathrm{I}} \cup \mathcal{F}_h^{\mathrm{B}}$, where $\mathcal{F}_h^{\mathrm{I}}$ denotes the set of internal faces and $\mathcal{F}_h^{\mathrm{B}}$ the set of boundary faces. For all $T \in \mathcal{T}_h$, $\mathcal{F}_T$ collects the mesh faces lying on the boundary of T. Reciprocally, given a face $F \in \mathcal{F}_h$, $\mathcal{T}_F$ collects the elements which F is a face of. Notice that $\mathrm{card}(\mathcal{T}_F) = 2$ for internal faces and $\mathrm{card}(\mathcal{T}_F) = 1$ for boundary faces. For all $T \in \mathcal{T}_h$ and $F \in \mathcal{F}_T$, $\mathbf{n}_{TF}$ denotes the unit vector normal to F pointing out of T.

2.1.2 Lebesgue and Sobolev spaces

Let $X \in \mathcal{T}_h \cup \mathcal{F}_h \cup \{\Omega\}$. $L^2(X)$ denotes the Hilbert space of square-integrable functions over X, equipped with its usual inner product

$$(u,v)_X := \int_X vw \qquad \forall v,w \in L^2(X).$$

The same notation is also used for the inner product of $[L^2(X)]^d$, i.e.

$$(\mathbf{v},\mathbf{w})_X := \int_X \mathbf{v}\cdot\mathbf{w} \qquad \forall \mathbf{u},\mathbf{v} \in [L^2(X)]^d.$$

Additionally, we denote by $H^1(X)$ the Sobolev space of order 1, that is, the space spanned by functions of $L^2(X)$ whose partial derivatives are also square-integrable, and by $H^1_0(X)$ its subspace with vanishing trace on the boundary ∂X of X:

$$H^1_0(X) := \{v \in H^1(X) \mid v_{|\partial X} = 0\}.$$

2.1.3 Polynomial spaces

Let $\ell \in \mathbb{N}$ be a polynomial degree, and $X \in \mathcal{T}_h \cup \mathcal{F}_h \cup \{\Omega\}$. $\mathbb{P}^\ell(X)$ denotes the space spanned by the restriction to X of d-variate polynomials of degree at most ℓ. When $X \in \mathcal{F}_h$, $\mathbb{P}^\ell(X)$ is

isomorphic to the space of $(d-1)$-variate polynomials of total degree $\leq \ell$. Given the set of mesh elements $\mathcal{T}_h$, we also introduce the broken polynomial space $\mathbb{P}^\ell(\mathcal{T}_h) := \bigtimes_{T\in\mathcal{T}_h} \mathbb{P}^\ell(T)$.

2.2 Model problem

Given a source function $f \in L^2(\Omega)$, we consider the following diffusion problem with homogeneous Dirichlet boundary conditions:

$$\begin{cases} -\nabla\cdot(\mathbf{K}\nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where the diffusion tensor $\mathbf{K}\colon \Omega \to \mathbb{R}^{d\times d}_{\text{sym}}$ (with $\mathbb{R}^{d\times d}_{\text{sym}}$ collecting symmetric $d\times d$ real matrices) is assumed uniformly elliptic and piecewise constant over a fixed partition of Ω into polyhedra. The variational formulation of problem (2.1) reads

$$\begin{gathered} \text{Find } u \in H^1_0(\Omega) \text{ such that} \\ a(u,v) = \int_\Omega fv \quad \forall v \in H^1_0(\Omega), \end{gathered} \tag{2.2}$$

where the bilinear form $a\colon H^1(\Omega)\times H^1(\Omega) \to \mathbb{R}$ is such that, for all $v, w \in H^1(\Omega)$,

$$a(v,w) := (\mathbf{K}\nabla v, \nabla w)_\Omega = \int_\Omega \mathbf{K}\nabla u\cdot\nabla v.$$

We now consider a polyhedral mesh $\mathcal{T}_h$ which partitions Ω in such a way that the diffusion tensor is constant inside each element, and we denote $\mathbf{K}_T := \mathbf{K}|_T$. Decomposing both global integrals in (2.2) as sums of local integrals over the elements of the mesh $\mathcal{T}_h$, problem (2.2) becomes

$$\sum_{T\in\mathcal{T}_h} (\mathbf{K}_T\nabla u, \nabla v)_T = \sum_{T\in\mathcal{T}_h} (f,v)_T \quad \forall v \in H^1_0(\Omega). \tag{2.3}$$

2.3 Fundamentals of the HHO method

Following [43, Section 3.1], we introduce the spaces and operators used in HHO to discretize (2.3).

2.3.1 The elliptic projection

Let $\ell \in \mathbb{N}$ denote a polynomial degree. Let X denote either a mesh element or face.
$\pi^\ell_X\colon L^2(X) \to \mathbb{P}^\ell(X)$ denotes the L^2-orthogonal projector on $\mathbb{P}^\ell(X)$. For all $v \in L^2(X)$, $\pi^\ell_X v$ is characterized by the orthogonality condition

$$\forall w \in \mathbb{P}^\ell(X), \quad (\pi^\ell_X v - v, w)_X = 0. \tag{2.4}$$

For all $T \in \mathcal{T}_h$, the *oblique elliptic projector* $\widetilde{\pi}^\ell_T\colon H^1(T) \to \mathbb{P}^\ell(T)$ also enforces an orthogonality condition on the error, this time applied to the gradients of the operands, and taking the diffusion tensor into account for the future purpose of mirroring the left-hand side of (2.3). Specifically,

given $v \in H^1(T)$, we enforce the following condition:

$$\forall w \in \mathbb{P}^\ell(T), \quad (\mathbf{K}_T \nabla(\widetilde{\pi}_T^\ell v - v), \nabla w)_T = 0. \tag{2.5a}$$

Note that the adjective *oblique* refers to the introduction of the tensor into the formula. We can show that given $v \in H^1(T)$, (2.5a) defines a unique gradient $\nabla\widetilde{\pi}_T^\ell v$. $\widetilde{\pi}_T^\ell v$ is then determined up to an additive constant, which we fix by imposing that $\widetilde{\pi}_T^\ell v$ and v have the same average value over T, i.e.

$$(\widetilde{\pi}_T^\ell v, 1)_T = (v, 1)_T. \tag{2.5b}$$

Condition (2.5a) characterizes a projector in the sense that $\nabla\widetilde{\pi}_T^\ell v$ minimizes the distance of ∇v from the space $\nabla\mathbb{P}^\ell(T)$ w.r.t. the norm induced by the oblique inner product $(\mathbf{K}_T \cdot, \cdot)_T$. Adding the constraint (2.5b) then yields an equivalent characterization of the oblique elliptic projector $\widetilde{\pi}_T^\ell$ in terms of norm minimization:

$$\widetilde{\pi}_T^\ell v = \operatorname*{arg\,min}_{\substack{w \in \mathbb{P}^\ell(T) \\ (w,1)_T = (v,1)_T}} \|\mathbf{K}_T^{1/2}(\nabla v - \nabla w)\|_T^2.$$

(Note that in the formulation above, $\|\cdot\|_T$ denotes the norm on the d-dimensional vector space $[L^2(T)]^d$.) Additionally, we observe that the elliptic projector preserves polynomials of degree at most ℓ, i.e.,

$$\widetilde{\pi}_T^\ell v = v \qquad \forall v \in \mathbb{P}^\ell(T). \tag{2.6}$$

Indeed, letting $v \in \mathbb{P}^\ell(T)$, we deduce from (2.5a) that $\nabla(\widetilde{\pi}_T^\ell v - v) = 0$, making $\widetilde{\pi}_T^\ell v - v$ a constant polynomial, whose closure condition (2.5b) imposes to be zero, hence the result.

2.3.2 Computation of the elliptic projection from the L^2-projections

Let $T \in \mathcal{T}_h$. Considering $v \in H^1(T)$, we show that, for any polynomial degree $k \in \mathbb{N}$ arbitrarily fixed, the elliptic projection of v of degree $k+1$ can be fully determined knowing only its L^2-projections of degree k in the interior and on the faces of T.

First of all, let us recall the following integration by parts formula: for all $w \in \mathcal{C}^\infty(\overline{T})$,

$$(\mathbf{K}_T \nabla v, \nabla w)_T = -(v, \nabla \cdot (\mathbf{K}_T \nabla w))_T + \sum_{F \in \mathcal{F}_T} (v, \mathbf{K}_T \nabla w \cdot \mathbf{n}_{TF})_F. \tag{2.7}$$

According to (2.5a) with $\ell = k+1$, we have, for all $w \in \mathbb{P}^{k+1}(T)$,

$$(\mathbf{K}_T \nabla \widetilde{\pi}_T^{k+1} v, \nabla w)_T = (\mathbf{K}_T \nabla v, \nabla w)_T.$$

Transforming the right-hand side of this expression by means of the integration by parts formula (2.7), we have

$$(\mathbf{K}_T \nabla \widetilde{\pi}_T^{k+1} v, \nabla w)_T = -(v, \underbrace{\nabla \cdot (\mathbf{K}_T \nabla w)}_{\in \mathbb{P}^{k-1}(T)})_T + \sum_{F \in \mathcal{F}_T} (v, \underbrace{\mathbf{K}_T \nabla w \cdot \mathbf{n}_{TF}}_{\in \mathbb{P}^k(F)})_F. \tag{2.8}$$

Noticing that $w \in \mathbb{P}^{k+1}(T)$ implies $\nabla \cdot (\mathbf{K}_T \nabla w) \in \mathbb{P}^{k-1}(T) \subset \mathbb{P}^k(T)$, we can replace the term $(v, \nabla \cdot (\mathbf{K}_T \nabla w))_T$ with $(\pi_T^k v, \nabla \cdot (\mathbf{K}_T \nabla w))_T$ thanks to (2.4). Similarly, $\mathbf{K}_T \nabla w \cdot \mathbf{n}_{TF}$ being of degree k, we can replace v in the last inner product with $\pi_T^k v$. (2.8) then becomes

$$(\mathbf{K}_T \nabla \widetilde{\pi}_T^{k+1} v, \nabla w)_T = -(\pi_T^k v, \nabla \cdot (\mathbf{K}_T \nabla w))_T + \sum_{F \in \mathcal{F}_T} (\pi_F^k v, \mathbf{K}_T \nabla w \cdot \mathbf{n}_{TF})_F. \tag{2.9a}$$

This equation points out how $\nabla \widetilde{\pi}_T^{k+1} v$ can be fully determined only knowing $\pi_T^k v$ and $(\pi_F^k v)_{F \in \mathcal{F}_T}$. The constant needed to recover $\widetilde{\pi}_T^{k+1} v$ from $\nabla \widetilde{\pi}_T^{k+1} v$ can be obtained through the constraint (2.5b), enforcing that $\widetilde{\pi}_T^{k+1} v$ and v have the same average value, i.e.,

$$(\widetilde{\pi}_T^{k+1} v, 1)_T = (v, 1)_T.$$

Now, seeing the constant function 1 as a polynomial in $\mathbb{P}^0(T) \subset \mathbb{P}^k(T)$, v can once again be replaced with $\pi_T^k v$ in the right-hand side, hence

$$(\widetilde{\pi}_T^{k+1} v, 1)_T = (\pi_T^k v, 1)_T. \tag{2.9b}$$

2.3.3 Discrete spaces and operators

The preceding remarks lead us to introduce the following *hybrid space* of local DoFs, for all $T \in \mathcal{T}_h$:

$$\underline{U}_T^k := \left\{ \underline{v}_T := (v_T, (v_F)_{F \in \mathcal{F}_T}) \mid v_T \in \mathbb{P}^k(T),\ v_F \in \mathbb{P}^k(F)\ \ \forall F \in \mathcal{F}_T \right\}.$$

The *hybrid* adjective, associated with the underlined notation, is reminiscent of the fact that vectors of $\underline{U}_T^k$ are defined as a collection of distinct objects in their natures, namely cell- and face-defined functions.

The *local interpolation operator* $\underline{I}_T^k \colon H^1(T) \to \underline{U}_T^k$ is defined such that, for all $v \in H^1(T)$,

$$\underline{I}_T^k v := \left(\pi_T^k v, (\pi_F^k v)_{F \in \mathcal{F}_T} \right). \tag{2.10}$$

Inspired by (2.9), we define the *local higher-order reconstruction operator* $p_T^{k+1} \colon \underline{U}_T^k \to \mathbb{P}^{k+1}(T)$ such that, for all $\underline{v}_T := (v_T, (v_F)_{F \in \mathcal{F}_T}) \in \underline{U}_T^k$, $p_T^{k+1} \underline{v}_T$ is the only polynomial of degree at most $k+1$ verifying

$$\begin{cases} (\mathbf{K}_T \nabla p_T^{k+1} \underline{v}_T, \nabla w)_T = -(v_T, \nabla \cdot (\mathbf{K}_T \nabla w))_T + \displaystyle\sum_{F \in \mathcal{F}_T} (v_F, \mathbf{K}_T \nabla w \cdot \mathbf{n}_{TF})_F & \text{(2.11a)} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall w \in \mathbb{P}^{k+1}(T), \\ (p_T^{k+1} \underline{v}_T, 1)_T = (v_T, 1)_T. & \text{(2.11b)} \end{cases}$$

By definition, we have the identity

$$(p_T^{k+1} \circ \underline{I}_T^k) = \widetilde{\pi}_T^{k+1}. \tag{2.12}$$

Moreover, given the polynomial invariance of the elliptic projector (2.6), the following polynomial consistency property directly follows from (2.12):

$$(p_T^{k+1} \circ \underline{I}_T^k) w = w \quad \forall w \in \mathbb{P}^{k+1}(T). \tag{2.13}$$

The global hybrid discrete space is defined as

$$\underline{U}_h^k := \{\underline{v}_h := ((v_T)_{T\in\mathcal{T}_h}, (v_F)_{F\in\mathcal{F}_h}) \,|\, v_T \in \mathbb{P}^k(T) \;\; \forall T \in \mathcal{T}_h, \; v_F \in \mathbb{P}^k(F) \;\; \forall F \in \mathcal{F}_h\}.$$

For a generic vector of discrete unknowns $\underline{v}_h \in \underline{U}_h^k$ expressed as $\underline{v}_h := ((v_T)_{T\in\mathcal{T}_h}, (v_F)_{F\in\mathcal{F}_h})$, we denote its restriction to T by $\underline{v}_T := (v_T, (v_F)_{F\in\mathcal{F}_T}) \in \underline{U}_T^k$. We also define $\underline{U}_{h,0}^k$ as the subset of $\underline{U}_h^k$ whose face-based functions vanish on $\partial\Omega$, i.e.

$$\underline{U}_{h,0}^k := \{\underline{v}_h \in \underline{U}_h^k \mid v_F = 0 \quad \forall F \in \mathcal{F}_h^{\mathrm{B}}\}.$$

The global potential reconstruction operator $p_h^{k+1} : \underline{U}_h^k \to U_{\mathcal{T}_h}^{k+1}$ is obtained patching the corresponding local counterparts: for all $\underline{v}_h \in \underline{U}_h^k$, we set

$$(p_h^{k+1}\underline{v}_h)_{|T} := p_T^{k+1}\underline{v}_T \qquad \forall T \in \mathcal{T}_h.$$

2.3.4 Discretization of the model problem

2.3.4.1 HHO formulation

The global bilinear form $a_h : \underline{U}_h^k \times \underline{U}_h^k \to \mathbb{R}$ is assembled from elementary contributions as follows:

$$a_h(\underline{u}_h, \underline{v}_h) := \sum_{T\in\mathcal{T}_h} a_T(\underline{u}_T, \underline{v}_T),$$

where for all $T \in \mathcal{T}_h$, the local bilinear form $a_T : \underline{U}_T^k \times \underline{U}_T^k \to \mathbb{R}$ is defined as

$$a_T(\underline{u}_T, \underline{v}_T) := (\mathbf{K}_T \nabla p_T^{k+1}\underline{u}_T, \nabla p_T^{k+1}\underline{v}_T)_T + s_T(\underline{u}_T, \underline{v}_T). \tag{2.14}$$

In this expression, the first term is responsible for consistency while the second, involving the bilinear form $s_T : \underline{U}_T^k \times \underline{U}_T^k \to \mathbb{R}$, is required to ensure stability of the scheme. The global discrete problem then reads

$$\begin{aligned} &\text{Find } \underline{u}_h \in \underline{U}_{h,0}^k \text{ such that} \\ &a_h(\underline{u}_h, \underline{v}_h) = \sum_{T\in\mathcal{T}_h} (f, v_T)_T \qquad \forall \underline{v}_h \in \underline{U}_{h,0}^k. \end{aligned} \tag{2.15}$$

2.3.4.2 Stabilization

Design conditions for the stabilization bilinear form s_T are provided in [43, Assumption 3.9]. These conditions imply, in particular, that s_T must depend on its argument only through the *difference operators* $\delta_T^k : \underline{U}_T^k \to \mathbb{P}^k(T)$ and, for all $F \in \mathcal{F}_T$, $\delta_{TF}^k : \underline{U}_T^k \to \mathbb{P}^k(F)$ such that, for all

$\underline{v}_T \in \underline{U}_T^k$,

$$\delta_T^k \underline{v}_T := \pi_T^k(p_T^{k+1}\underline{v}_T - v_T) \tag{2.16a}$$

$$\forall F \in \mathcal{F}_T, \quad \delta_{TF}^k \underline{v}_T := \pi_F^k(p_T^{k+1}\underline{v}_T - v_F). \tag{2.16b}$$

These operators capture the higher-order correction that the reconstruction p_T^{k+1} adds to the cell and face unknowns, respectively. Remark that they vanish whenever their argument is the interpolate of a polynomial in $\mathbb{P}^{k+1}(T)$. Indeed, letting $w \in \mathbb{P}^{k+1}(T)$, we have

$$\delta_T^k \underline{I}_T^k w \overset{(2.10)}{=} \pi_T^k(p_T^{k+1}\underline{I}_T^k w - \pi_T^k w) \overset{(2.13)}{=} \pi_T^k(w - \pi_T^k w) = \pi_T^k w - \pi_T^k w = 0.$$

Likewise with δ_{TF}^k.

A classical expression for s_T is the following:

$$s_T(\underline{v}_T, \underline{w}_T) := \sum_{F\in\mathcal{F}_T} \frac{K_{TF}}{h_F}((\delta_{TF}^k - \delta_T^k)\underline{v}_T, (\delta_{TF}^k - \delta_T^k)\underline{w}_T)_F,$$

where $K_{TF} := \mathbf{K}_T\mathbf{n}_{TF}\cdot\mathbf{n}_{TF}$ for all $F \in \mathcal{F}_T$. This stabilization penalizes the difference between the higher-order correction inside the element and on its faces. By construction, it vanishes when applied to polynomial functions of total degree $\leq k+1$, and it is symmetric positive semi-definite. Consequently, it can safely be added to the otherwise *unstable* bilinear form defined by the consistency term of (2.14) without loss of any important property, namely polynomial consistency and symmetry.

Remark 2. *(Links with Hybridizable Discontinuous Galerkin methods) In the present formulation, cell and face unknowns represent local polynomials of equal degree k. A variant consists in taking cell unknowns of degree $k+1$ instead of k, in which case the method is linked to a special formulation of Hybridizable Discontinuous Galerkin methods [83, 96]. As shown in [36], this formulation admits a reduced stabilization enabling improved convergence properties comparable to those of HHO methods. For a broad discussion on the links and differences between HHO and Hybridizable Discontinuous Galerkin methods, see [43, Section 5.1.6].*

2.3.5 Assembly and static condensation

2.3.5.1 Global system

The local contributions corresponding to the representations, in the selected basis for $\underline{U}_{h,0}^k$, of the bilinear form a_T (cf. (2.14)) and of the linear form $\underline{U}_T^k \ni \underline{v}_T \mapsto (f, v_T)_T \in \mathbb{R}$ are, respectively, the matrix $\mathbf{A}_T$ and the vector $\mathbf{B}_T$ such that

$$\mathbf{A}_T := \begin{pmatrix} \mathbf{A}_{TT} & \mathbf{A}_{T\mathcal{F}_T} \\ \mathbf{A}_{\mathcal{F}_T T} & \mathbf{A}_{\mathcal{F}_T\mathcal{F}_T} \end{pmatrix}, \qquad \mathbf{B}_T := \begin{pmatrix} \mathbf{b}_T \\ \mathbf{0} \end{pmatrix}, \tag{2.17}$$

in which the unknowns have been numbered so that cell unknowns come first and face unknowns come last; see [43, Appendix B] for further details. After assembling the local contributions and eliminating the boundary unknowns by a strong enforcement of the Dirichlet boundary

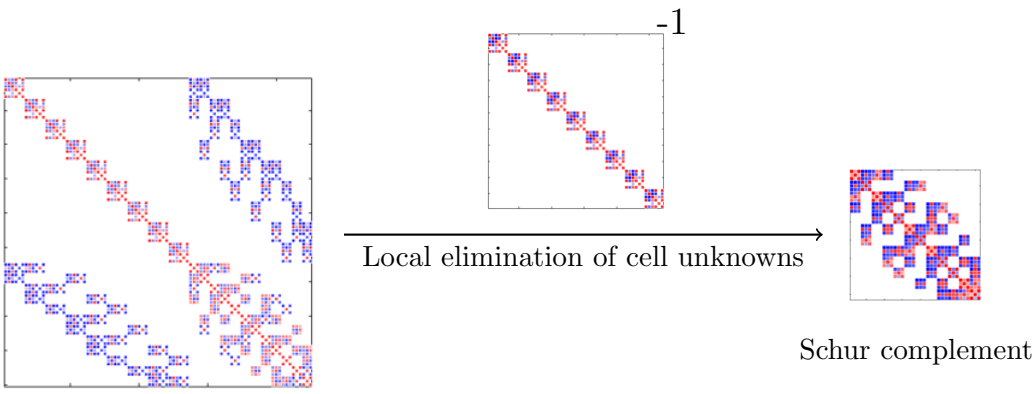


Figure 2.1 Static condensation

condition, we end up with a global linear system of the form

$$\begin{pmatrix} \mathbf{A}_{\mathcal{T}_h\mathcal{T}_h} & \mathbf{A}_{\mathcal{T}_h\mathcal{F}_h^{\mathrm{I}}} \\ \mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{T}_h} & \mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{F}_h^{\mathrm{I}}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_{\mathcal{T}_h} \\ \mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{\mathcal{T}_h} \\ \mathbf{0} \end{pmatrix}. \tag{2.18}$$

2.3.5.2 Static condensation

Since cell-DoFs are coupled with each other only through face-DoFs, $\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}$ is block-diagonal, therefore inexpensive to invert. The static condensation process takes advantage of this property to locally eliminate the cell-DoFs: it goes by expressing $\mathbf{v}_{\mathcal{T}_h}$ in terms of $\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}}$ in the first equation of (2.18):

$$\mathbf{v}_{\mathcal{T}_h} = -\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}\mathbf{A}_{\mathcal{T}_h\mathcal{F}_h^{\mathrm{I}}}\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}} + \mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}\mathbf{b}_{\mathcal{T}_h}, \tag{2.19}$$

and then replacing $\mathbf{v}_{\mathcal{T}_h}$ with its expression (2.19) in the second equation:

$$\left(\mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{F}_h^{\mathrm{I}}} - \mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{T}_h}\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}\mathbf{A}_{\mathcal{T}_h\mathcal{F}_h^{\mathrm{I}}}\right)\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}} = -\mathbf{A}_{\mathcal{F}_h^{\mathrm{I}}\mathcal{T}_h}\mathbf{A}_{\mathcal{T}_h\mathcal{T}_h}^{-1}\mathbf{b}_{\mathcal{T}_h}, \tag{2.20}$$

thus yielding a smaller system, involving only face unknowns. See Figure 2.1 for a matrix representation of the static condensation. The main advantage of this technique is the reduction of the problem size, especially for high polynomial degrees k.

2.3.5.3 Post-processing: higher-order reconstruction

After solving (2.20) for the face unknowns $\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}}$, the cell unknowns are recovered element by element by the local counterpart of (2.19):

$$\mathbf{v}_T = -\mathbf{A}_{TT}^{-1}\mathbf{A}_{T\mathcal{F}_T}\mathbf{v}_{\mathcal{F}_T} + \mathbf{A}_{TT}^{-1}\mathbf{b}_T, \tag{2.21}$$

where $\mathbf{v}_T$ denotes the restriction to T of $\mathbf{v}_{\mathcal{T}_h}$, and $\mathbf{v}_{\mathcal{F}_T}$ the restriction of $\mathbf{v}_{\mathcal{F}_h^{\mathrm{I}}}$ to the faces of T interior to the domain, completed by zeros for boundary faces to obtain $\mathbf{v}_{\mathcal{F}_h}$.

Now that $\underline{v}_h := (v_{\mathcal{T}_h}, v_{\mathcal{F}_h}) \in \underline{U}_h^k$ is available via its algebraic counterpart, we can apply the potential reconstruction operator p_T^{k+1} to finally construct the discrete HHO solution.

2.4 Conclusion and summary

This chapter presented the construction of the HHO method. With respect to the desirable features mentioned in the introductory Chapter 1, we would also like to mention the following important properties and cite the theoretical results they come from.

- **Optimal convergence.** For the Poisson problem, under standard elliptic regularity hypotheses, and assuming a regular enough solution (namely, $H^{k+2}(\mathcal{T}_h)$), the error estimate in L^2-norm shows a convergence in h^{k+2} (refer to [43, Theorem 2.32]).
- **Robustness.** For the more general variable diffusion problem (2.1), [43, Theorem 3.19] proves the full robustness of the scheme with respect to heterogeneity, in the sense that the error estimate in energy norm does not depend on discontinuities occurring in the diffusion tensor across mesh elements. The scheme is also *partially* robust to anisotropy, in that the upper bound, although depending on each local anisotropy ratio, is not globally conditioned by the maximum value of those local ratios. Note that in L^2-norm, however, the convergence in h^{k+2} cited above is no longer valid in the case of a piecewise constant diffusion tensor, even if the solution is $H^{k+2}(\mathcal{T}_h)$. Indeed, the elliptic regularity hypothesis is not verified. For this model, elliptic regularity is only known if Ω is convex and $\mathbf{K}$ is Lipschitz continuous (refer to [43, Remark 3.21]).
- **Flux conservation.** The method satisfies local balances with continuous normal trace of the fluxes at element interfaces. See [43, Lemma 2.25] for the Poisson problem and [43, Lemma 3.17] for the variable diffusion problem.
- **Computational gain of hybridization.** The possibility for hybridized methods of statically condensing their cell unknowns allows for a drastic reduction of the globally coupled DoFs. Compared to the plain non-hybridized DG method of degree k, the downsizing can be substantial, and grows larger with k. Indeed, denoting by $N_{k,d}^{\mathrm{HHO}}$ and $N_{k,d}^{\mathrm{DG}}$ the respective numbers of globally coupled DoFs for DG and HHO, we have

$$N_{k,d}^{\mathrm{DG}} = \sum_{T\in\mathcal{T}_h} \dim(\mathbb{P}^k(T)) = \binom{k+d}{k} \operatorname{card}(\mathcal{T}_h)$$
$$N_{k,d}^{\mathrm{HHO}} = \sum_{F\in\mathcal{F}_h} \dim(\mathbb{P}^k(F)) = \binom{k+(d-1)}{k} \operatorname{card}(\mathcal{F}_h),$$

  which gives, for $d = 3$,

$$N_{k,3}^{\mathrm{DG}} = \frac{1}{6}(k+3)(k+2)(k+1)\operatorname{card}(\mathcal{T}_h)$$
$$N_{k,3}^{\mathrm{HHO}} = \frac{1}{2}(k+2)(k+1)\operatorname{card}(\mathcal{F}_h).$$

  Especially, the number of DoFs grows in k^3 for DG and in k^2 for HHO.

At last, we would like to conclude this chapter by briefly enumerating *take-out* ideas about HHO, which we find the most relevant in view of the contributions contained in the next chapters.

- **Interpretation of the DoFs.** The element of $\underline{U}_h^k$ associated to a function $v \in H^1(\Omega)$

through the interpolator is $\big((\pi_T^k v)_{T\in\mathcal{T}_h}, (\pi_F^k v)_{F\in\mathcal{F}_h}\big)$, showing that the element and face unknowns of the HHO method (2.15) can be interpreted as local L^2-orthogonal projections of the exact solution.

- **Higher-order reconstruction.** The application of p_T^{k+1} to a local set of hybrid unknowns of degree k allows the gain of one degree of approximation. Precisely, For any $v \in H^1(T)$, applying p_T^{k+1} to the interpolate of v yields the local oblique elliptic projection of v on $\mathbb{P}^{k+1}(T)$ (cf. (2.12)).
- **Discrete approximate.** As a result, given DoFs of degree k, the approximation provided by the HHO method is a broken polynomial of degree $k+1$. Particularly, it is obtained in a post-processing step by the computation of $p_h^{k+1}\underline{u}_h$, where $\underline{u}_h$ is the solution of the discrete formulation (2.15).
- **The global HHO process**, from assembly to the reconstruction of the discrete approximate is summarized in Figure 2.2.

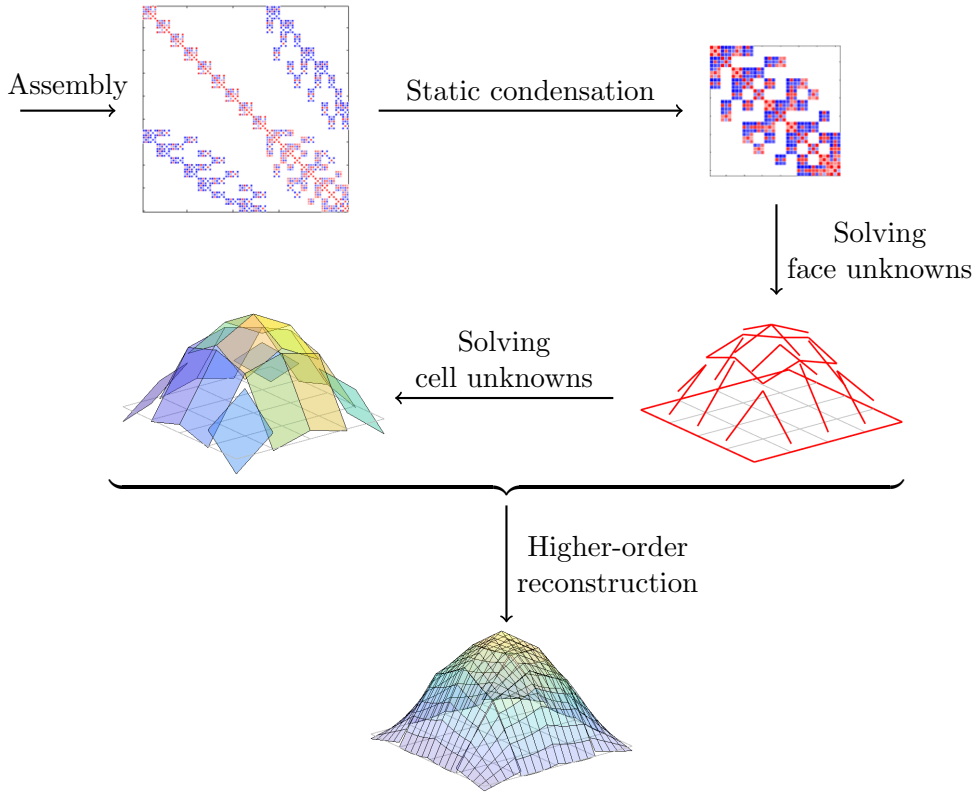


Figure 2.2 Summary of the HHO process

Chapter 3

A geometric multigrid method for HHO discretizations

About a now-famous iterative method for the solution of linear systems:
"*I recommend this method to you for imitation. You will hardly ever again eliminate directly, at least not when you have more than 2. The indirect procedure can be done while half asleep, or while thinking about other things.*"

Carl Friedrich Gauß — in a letter to Gerling on Dec. 26th 1823

The content of this chapter was published in *SIAM Journal on Scientific Computing* (Copper Mountain Special Section 2020) [50].

Considering the HHO formulation (2.15) of the diffusion problem (2.1), we address in this chapter the efficient solution of the trace system (2.20) by means of a novel, geometric, h-multigrid algorithm. The method we propose hinges on face-defined functions at every grid level, and works in synergy with the discretization through intergrid transfer operators leveraging the higher-order potential reconstruction (2.11). It also applies to any polynomial degree of approximation without resorting to an additional p-multigrid, which, in practice, can be seen as a valuable reduction of the implementation cost.

Our algorithm development is based on the systematic approach proposed in the seminal guide to multigrid development [28]. The method consists in identifying the individual difficulties and obstacles that may inhibit the optimal performance of a multigrid algorithm. For each of the difficulties, appropriate multigrid components are developed. Here, in particular, we start from the Laplace problem discretized on the skeleton of a simple Cartesian mesh to first develop a multigrid method that is scalable in the number of unknowns and robust with respect to the polynomial degree. With this algorithm, we then proceed to work on more general problems and meshes. One consequence of this approach is that we focus on multigrid as a solver, not as

a preconditioner. When used only as preconditioner, this tends to obscure misconceptions in the design of the multigrid components. The multigrid algorithms developed here can serve as efficient stand-alone solvers, but they can also serve as preconditioners, as we will explore in future research.

In this work, the polynomial order of approximation is preserved at every level at the sole cost of using a blockwise smoother instead of a pointwise one. This approach originates from the remark that a high-order finite element discretization yields a block matrix, whose diagonal blocks are formed by the degrees of freedom connected to the same cell. This configuration usually destroys the desirable M- or H-matrix structure and, along with it, the convergence of pointwise smoothers; on the other hand, the block structure paves the way to using block versions of similar smoothers. In a more functional way of thinking, relaxing together the DoFs related to the same polynomial comes as intuitive. The robustness of the multigrid algorithms using block smoothers for high-order methods has been experimentally illustrated in [74] and later used in practical solvers such as [98].

This chapter is organized as follows. Section 3.1 is devoted to the construction of the multigrid algorithm and illustrates how it takes advantage of the HHO potential reconstruction operator. Numerical results for various polynomial degrees are presented in Section 3.2, considering both homogeneous and heterogeneous diffusion problems in two and three space dimensions. The numerical experiments show that the number of iterations is nearly independent of the mesh size and of the presence of jumps in the diffusion coefficient.

3.1 Multigrid algorithm

3.1.1 Coarsening strategy

The levels of the multigrid method are numbered from 1 to L, L being the finest and 1 the coarsest. In what follows, we denote by ℓ the generic level and by h_ℓ the corresponding mesh size. To simplify the notation, from this point on h_ℓ is replaced by ℓ in subscripts so we write, e.g., $\mathcal{T}_\ell$ instead of $\mathcal{T}_{h_\ell}$, $\mathcal{F}_\ell$ instead of $\mathcal{F}_{h_\ell}$, and so on.

Relative to those levels, we consider a hierarchy of nested polyhedral meshes $(\mathcal{T}_\ell, \mathcal{F}_\ell)_{\ell=1\dots L}$. We assume the hierarchy to successively coarsen not only elements, but also faces. This means that, for all $\ell = 1 \dots L$, letting $h_{\mathcal{T}_\ell} := \max_{T\in\mathcal{T}_\ell} h_T$ and $h_{\mathcal{F}_\ell} := \max_{F\in\mathcal{F}_\ell} h_F$, it holds

$$h_{\mathcal{T}_{\ell-1}} > h_{\mathcal{T}_\ell}, \qquad h_{\mathcal{F}_{\ell-1}} > h_{\mathcal{F}_\ell}.$$

Standard coarsening of structured Cartesian and triangular meshes, as well as unstructured meshes obtained from successive structured refinements of an initial coarse mesh fall under the scope of these assumptions; examples of admissible coarsening strategies are illustrated in Figure 3.1. Requiring that the faces be coarsened is justified by our algorithm being face-defined at every level. Indeed, the smoother applies to faces the same way it applies to elements in a classical element-defined multigrid method: once the high frequencies of the error have been annihilated on the fine mesh, the smoother requires coarser elements to reach the low frequencies on the coarse mesh. For the same reason, a multigrid working on the mesh skeleton needs the faces to be coarsened: the consequence of a face not being coarsened between a fine and a coarse

mesh would be to keep the smoother working on the same range of frequencies, leaving it unable to efficiently reduce the lowest ones; see Figure 3.13 below.

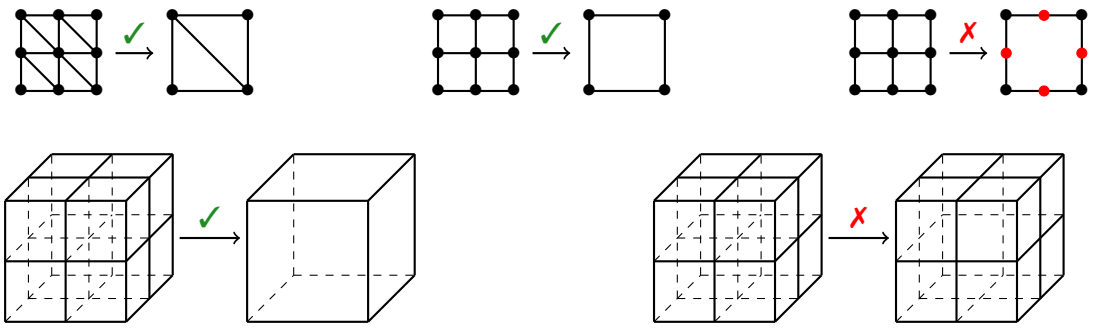


Figure 3.1 Coarsening examples. In the top row (2D), the first two coarsenings are admissible, whereas the third one is not: edges have been removed, but none of the remaining ones has been coarsened. Similar 3D examples in the bottom row. In the non-admissible case, the coarsened cube has 6 sets of 4 coplanar faces, which have not been coarsened.

We will also assume that, for every $\ell = 1 \dots L$, the diffusion coefficient is piecewise constant on $\mathcal{T}_\ell$, so that jumps can occur at faces but not inside elements.

3.1.2 Discrete spaces

Let $k \in \mathbb{N}$ be a fixed polynomial degree. For all level $\ell = 1 \dots L$, we introduce the following broken polynomial spaces, respectively supported by the mesh and its skeleton:

$$U_{\mathcal{T}_\ell}^{k+\delta} := \left\{ v_{\mathcal{T}_\ell} := (v_T)_{T\in\mathcal{T}_\ell} \mid v_T \in \mathbb{P}^{k+\delta}(T) \;\; \forall T \in \mathcal{T}_\ell \right\} \qquad \text{for } \delta \in \{0,1\},$$
$$U_{\mathcal{F}_\ell}^{k} := \left\{ v_{\mathcal{F}_\ell} := (v_F)_{F\in\mathcal{F}_\ell} \mid v_F \in \mathbb{P}^{k}(F) \;\; \forall F \in \mathcal{F}_\ell \right\}.$$

The homogeneous Dirichlet boundary condition is strongly enforced in the following subspace of $U_{\mathcal{F}_\ell}^k$:

$$U_{\mathcal{F}_\ell,0}^{k} := \left\{ v_{\mathcal{F}_\ell} \in U_{\mathcal{F}_\ell}^{k} \mid v_F = 0 \;\; \forall F \in \mathcal{F}_\ell^{\mathrm{B}} \right\}.$$

3.1.3 Prolongation

We consider two successive levels ℓ (fine) and $\ell-1$ (coarse). In this algorithm, *faces* support the functions at every level. To prolongate a coarse function onto the fine mesh skeleton, which includes some faces that are not present in the coarse mesh, we propose an intermediary step that passes through the cells (Figure 3.2). Following this idea, the prolongation operator $P\colon U_{\mathcal{F}_{\ell-1},0}^k \to U_{\mathcal{F}_\ell,0}^k$ is defined as the composition

$$P = \Pi_{\ell-1}^{\ell} \circ \Theta_{\ell-1}, \tag{3.1}$$

where the coarse level potential reconstruction operator $\Theta_{\ell-1}\colon U_{\mathcal{F}_{\ell-1},0}^k \to U_{\mathcal{T}_{\ell-1}}^{k+1}$ reconstructs a broken polynomial of degree $k+1$ on $\mathcal{T}_{\ell-1}$ from face unknowns; then, the trace prolongation operator $\Pi_{\ell-1}^{\ell}\colon U_{\mathcal{T}_{\ell-1}}^{k+1} \to U_{\mathcal{F}_\ell,0}^k$ maps the polynomials of degree $k+1$ defined on the coarse cells to a broken polynomial function of degree k on the fine skeleton.

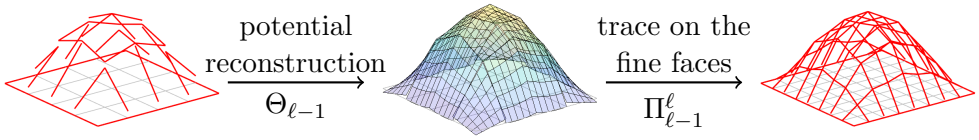


Figure 3.2 Prolongation from coarse to fine edges.

3.1.3.1 Θ_ℓ: from faces to cells

This operator is at the core of the algorithm and is what makes it original. Given a trace error function $e_{\mathcal{F}_\ell} \in U^k_{\mathcal{F}_\ell,0}$ as the operand of Θ_ℓ, we recover a cell-defined error function $e_{\mathcal{T}_\ell} \in U^{k+1}_{\mathcal{T}_\ell}$ by locally reversing the static condensation process, then take advantage of the potential reconstruction operator introduced in Section 2.3.3 to gain one order of approximation inside the cells.

As these operations are local, the process will be outlined for a generic mesh element $T \in \mathcal{T}_\ell$. Defining $e_{\mathcal{F}_T} := (e_F)_{F\in\mathcal{F}_T}$, we let $\mathbf{e}_{\mathcal{F}_T} := (\mathbf{e}_F)_{F\in\mathcal{F}_T}$ denote its algebraic representation as vectors of coefficients in the selected polynomial bases. If we denote by $(\mathbf{x}^\top_{\mathcal{T}_\ell}, \mathbf{x}^\top_{\mathcal{F}_\ell})^\top$ the vector obtained completing the solution vector of the global system (2.18) with boundary unknowns equal to zero, then its restriction to T, namely $(\mathbf{x}^\top_T, \mathbf{x}^\top_{\mathcal{F}_T})^\top$, is the solution of the local system defined by (2.17), i.e.

$$\begin{pmatrix} \mathbf{A}_{TT} & \mathbf{A}_{T\mathcal{F}_T} \\ \mathbf{A}_{\mathcal{F}_T T} & \mathbf{A}_{\mathcal{F}_T\mathcal{F}_T} \end{pmatrix} \begin{pmatrix} \mathbf{x}_T \\ \mathbf{x}_{\mathcal{F}_T} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_T \\ \mathbf{0} \end{pmatrix},$$

from which the static condensation process expresses $\mathbf{x}_T$ in terms of $\mathbf{x}_{\mathcal{F}_T}$ as

$$\mathbf{x}_T = -\mathbf{A}^{-1}_{TT}\mathbf{A}_{T\mathcal{F}_T}\mathbf{x}_{\mathcal{F}_T} + \mathbf{A}^{-1}_{TT}\mathbf{b}_T. \tag{3.2}$$

We now introduce the local face-defined approximate solution vector $\tilde{\mathbf{x}}_{\mathcal{F}_T}$ such that $\mathbf{e}_{\mathcal{F}_T} = \mathbf{x}_{\mathcal{F}_T} - \tilde{\mathbf{x}}_{\mathcal{F}_T}$, and, inspired by (3.2), we define the associated cell-based approximate vector $\tilde{\mathbf{x}}_T$ by

$$\tilde{\mathbf{x}}_T := -\mathbf{A}^{-1}_{TT}\mathbf{A}_{T\mathcal{F}_T}\tilde{\mathbf{x}}_{\mathcal{F}_T} + \mathbf{A}^{-1}_{TT}\mathbf{b}_T. \tag{3.3}$$

Definition (3.3) ensures consistency in the sense that for $\tilde{\mathbf{x}}_{\mathcal{F}_T} = \mathbf{x}_{\mathcal{F}_T}$, it yields $\tilde{\mathbf{x}}_T = \mathbf{x}_T$ by (3.2). We can finally define the error on the cell by setting $\mathbf{e}_T := \mathbf{x}_T - \tilde{\mathbf{x}}_T$, and replace $\mathbf{x}_T$ and $\tilde{\mathbf{x}}_T$ with their respective expressions (3.2) and (3.3), thus cancelling the terms involving $\mathbf{b}_T$ and giving

$$\mathbf{e}_T = -\mathbf{A}^{-1}_{TT}\mathbf{A}_{T\mathcal{F}_T}(\mathbf{x}_{\mathcal{F}_T} - \tilde{\mathbf{x}}_{\mathcal{F}_T}) = -\mathbf{A}^{-1}_{TT}\mathbf{A}_{T\mathcal{F}_T}\mathbf{e}_{\mathcal{F}_T}. \tag{3.4}$$

Once e_T is retrieved from its algebraic representation given by (3.4), the local potential reconstruction p^{k+1}_T defined in (2.11) is applied to the hybrid vector $(e_T, e_{\mathcal{F}_T})$ to obtain an approximate error of degree $k+1$ on the cell:

$$(\Theta_\ell e_{\mathcal{F}_\ell})_{|T} := p^{k+1}_T(e_T, e_{\mathcal{F}_T}) \qquad \forall T \in \mathcal{T}_h. \tag{3.5}$$

Figure 3.3 summarizes the process.

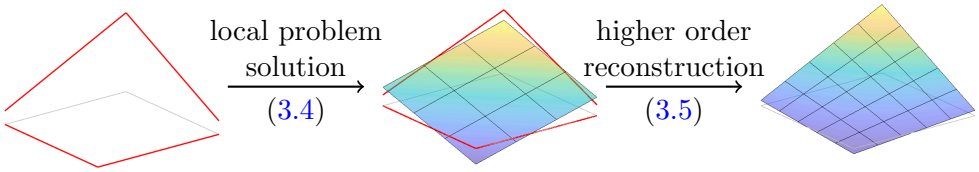


Figure 3.3 Reconstruction of a polynomial of degree $k+1$ from polynomials of degree k (here for $k=1$) on the four edges of a 2D square element.

3.1.3.2 $\Pi_{\ell-1}^{\ell}$: from cells to faces

For any $v \in U_{\mathcal{T}_{\ell-1}}^{k+1}$ and any $F \in \mathcal{F}_\ell$, $(\Pi_{\ell-1}^{\ell} v)_{|F}$ is built as the L^2-orthogonal projection on $\mathbb{P}^k(F)$ of the weighted average of the traces of v on both sides of F if F is an internal face, while $(\Pi_{\ell-1}^{\ell} v)_{|F}$ is set equal to zero if F is a boundary face, i.e.,

$$(\Pi_{\ell-1}^{\ell} v)_{|F} := \begin{cases} w_{T_1F}\, \pi_F^k(v_{|T_1})_{|F} + w_{T_2F}\, \pi_F^k(v_{|T_2})_{|F} & \text{if } F \in \mathcal{F}_\ell^{\mathrm{I}}, \\ 0 & \text{otherwise,} \end{cases} \tag{3.6}$$

where T_1, T_2 denote the distinct elements in $\mathcal{T}_F \subset \mathcal{T}_\ell$, π_F^k is the L^2-projector on $\mathbb{P}^k(F)$, and the weights satisfy

$$w_{T_1F} + w_{T_2F} = 1 \tag{3.7a}$$

$$\frac{w_{T_1F}}{w_{T_2F}} = \frac{K_{T_1F}}{K_{T_2F}}, \tag{3.7b}$$

where we remind the reader that, for $i \in \{1,2\}$, $K_{T_iF} := \mathbf{K}_{T_i}\mathbf{n}_{T_iF} \cdot \mathbf{n}_{T_iF}$. Enforcing both constraints (3.7) yields, for $i \in \{1,2\}$,

$$w_{T_iF} := \frac{K_{T_iF}}{K_{T_1F} + K_{T_2F}}. \tag{3.8}$$

3.1.4 Multigrid components

The prolongation operator P is defined by (3.1). The restriction operator R is defined as the adjoint of P in the usual way. Interpreted algebraically as matrices (using the notations $\mathbf{R}$ and $\mathbf{P}$), it means $\mathbf{R} = \mathbf{P}^\top$. Note that $\Pi_{\ell-1}^{\ell}$ does not make a distinction between the fine faces contained in the skeleton of the coarse grid and those that are not; consequently, the polynomials on coarse faces are not transferred identically to the fine grid, but instead take on new values coming from the (weighted) average of the reconstructed cell-polynomials on each side. The alternative way of prolongating coarse functions from coarse faces to their respective identical fine faces, namely keeping them unchanged, has also been tested (cf. Section 3.2.4) and yields a less efficient algorithm. This observation is consistent with the fact that solving the local problems produces additional information that the coarse polynomials do not possess. In addition, the reconstruction using higher degree polynomials also results in higher accuracy in the case where two fine faces are agglomerated into a single coarse one: the polynomial of degree $k+1$ on the coarse cell can induce two different polynomials of degree k on the two corresponding fine faces, which would not be the case if the reconstruction were only of degree k.

The coarse grid operator at level $\ell - 1$ can be chosen either as the discretized operator on the respective coarse mesh, or as the Galerkin construction: $A_{\ell-1} := RA_\ell P$. The numerical tests in the next section show equivalent performances.

In order to relax the DoFs related to the same polynomial function together, block versions of standard fixed-point smoothers are chosen, whose block size corresponds to the number of DoFs on each face.

3.2 Numerical results

3.2.1 Experimental setup

The numerical tests have been performed on the diffusion problem (2.1) in various d-dimensional domains $\Omega \subset \mathbb{R}^d$, $d \in \{2, 3\}$. The unit square/cube $\Omega := (0, 1)^d$ is used to study the algorithm on structured meshes, whereas more complicated geometries shall be used for unstructured ones. The source function f is chosen so that the analytical solution of the homogeneous problem corresponds to $(x, y) \mapsto \sin(4\pi x)\sin(4\pi y)$ in 2D and $(x, y, z) \mapsto \sin(4\pi x)\sin(4\pi y)\sin(4\pi z)$ in 3D. For structured cases, given an integer $N > 0$, the domain is discretized by a Cartesian grid composed of N^d square/cubic elements of side length $1/N$. Each of them is respectively decomposed into 2 triangles or 6 tetrahedra if the mesh is simplicial. In what follows, k denotes the polynomial degree on the faces (meaning that the HHO method ultimately yields an approximation of degree $k+1$). Our multigrid algorithm is used to solve the statically condensed linear system (2.20). The mesh is successively coarsened until the coarse system reaches a size with less than 1000 unknowns. On the coarsest level, the system is solved by a direct solver. The operators on the coarser levels are constructed directly as the discretization of the equation on the respective coarse meshes. The prolongation operator is defined according to (3.1) and the restriction operator is taken equal to its transpose in the usual sense. The smoother is a block Gauss–Seidel method, in which the block size corresponds to the number of face-DoFs. In pre-smoothing, the iteration is performed in lexicographic order, while in post-smoothing, it is performed in anti-lexicographic order to ensure the symmetry of the overall iteration. Note that experiments have also been performed with block-Jacobi with damping factor $2/3$, showing qualitatively equivalent results. This could be the basis for a parallel implementation. Here we will only report detailed results for the block Gauss–Seidel smoother. The multigrid cycles will vary depending on the test. An L^2-orthogonal Legendre basis is chosen to represent the local polynomials on cells and faces. The stopping criterion is set to $\|\mathbf{r}\|_2/\|\mathbf{b}\|_2 < 10^{-8}$, where $\mathbf{r}$ denotes the residual vector, $\mathbf{b}$ the right-hand side of the linear system, and $\|\cdot\|_2$ the Euclidean norm on the vector space of coordinates.

3.2.2 Homogeneous diffusion on structured meshes

The diffusion tensor field is constant across the domain and equals the identity matrix. The model problem is discretized using four structured meshes: Cartesian and triangular in 2D, Cartesian and tetrahedral in 3D. The mesh hierarchies are constructed from the fine mesh by standard coarsening. This strategy ensures the hierarchical nestedness as well as geometrically

similar elements at every level. Note that the tetrahedral meshes are built from the Cartesian ones, where each cube is divided into six geometrically similar tetrahedra (cf. [16, Figure 9]).

3.2.2.1 Preliminary tests using classical multigrid cycles

Figure 3.4 presents performance results of the multigrid algorithm as a solver, using the cheapest symmetric V-cycle that ensures convergence in a reasonable number of iterations as well as good scalability, namely V(1,1) for 2D meshes and the 3D Cartesian one, V(2,2) for the tetrahedral mesh. Leaving the lowest order case aside for the moment, these results are consistent with the desired multigrid property of a convergence rate that is independent of the mesh size and the number of levels, provided that a sufficient (yet reasonable) number of smoothing steps are performed. Moreover, although the number of iterations may increase moderately with the polynomial degree, the algorithm still exhibits the same desirable properties for high approximation orders.

For the lowest order case $k = 0$, the results are plotted throughout the tests in dashed lines. Here the results are less clear. Although in case of $k = 0$ we still observe good scalability on Cartesian meshes, the convergence on the triangular meshes deteriorates with growing mesh size. For the tetrahedral mesh in 3D and $k = 0$ no data is shown in Figure 3.4 since this version does not converge with the V(2,2) cycle. Here, more smoothing steps would be needed to ensure convergence. Our hypothesis is that the difference lies in the approximation properties of the HHO method, especially in the L^2-error estimate for cell unknowns, where the case $k = 0$ is discriminated (cf. [43, Lemma 2.33]). As a matter of fact, it is well known that the convergence of geometric multigrid methods built on coarse rediscretizations depends on the convergence properties of the underlying discretization through the coarse grid correction step.

3.2.2.2 Multigrid cycle optimization

While the above results demonstrate the asymptotic optimality of our new multigrid algorithm, we now proceed to studying how the inherent design options of multigrid can be used to further improve the real-life efficiency. In particular, we identify the most efficient cycle structure and how much pre- and post-smoothing should be performed. To assess the performance impact of these choices, it is necessary to define a criterion modeling the trade-off between convergence rate and iteration cost. We emphasize that the sole number of iterations is not sufficient to assess the solver's overall efficiency, because the cost of each iteration must be taken into account. Hence, Figure 3.5 compares the performance, measured in total computational work to reach convergence, of different multigrid cycles on a 2D test problem (triangular mesh, $N = 512$, $k = 1$). In the left plot, the numerical values have been obtained by taking the theoretical computational work (in flops) of the multigrid algorithm, using the following simplifying rules: (i) the asymptotic value of the work count is used, meaning that only the dominant term (in the matrix size or non-zero entries) is kept; (ii) the work of the direct solver on the coarsest grid is neglected. The total number of iterations required to achieve convergence is displayed for information in the right plot. Recalling that all tests stop upon reaching the same convergence criterion, we consider all the solutions produced to be equivalent: for instance, V(1,1) is about 50% more computationally expensive than V(0,2) for the same quality result. Note that V- and W-cycles

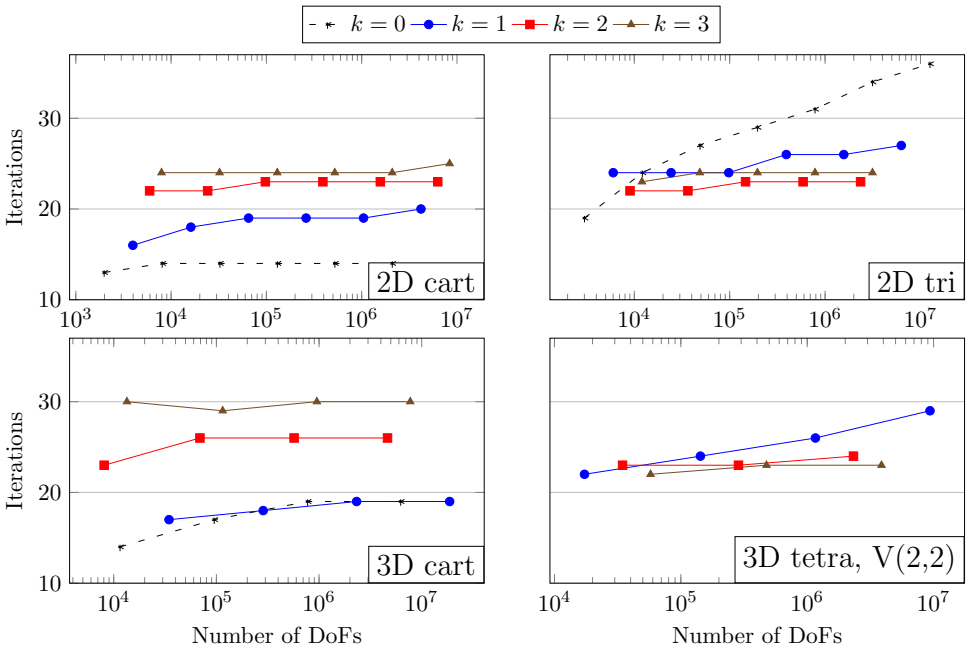


Figure 3.4 Number of iterations to achieve convergence for the homogeneous problem on structured meshes: 2D Cartesian (top left), 2D triangular (top right), 3D Cartesian (bottom left), 3D tetrahedral (bottom right). The first three are solved using the V(1,1) cycle, while the last one (tetrahedral mesh) is solved using the V(2,2) cycle.

have been tested, and exhibit, for the same numbers of smoothing steps, the same convergence rate. Since W-cycles are more computationally expensive by definition, the corresponding results are not presented in further detail. The comparisons have also been made in terms of CPU time in order to compare the estimates of computational work with respect to a hard practical criterion. Again, these results show a similar ranking and allow to draw the same conclusions; so they are not displayed in detail either.

Now, we can comment on the importance of post-smoothing: for example, amongst V(1,2), V(2,1) and V(0,3), although they all have the same total number of smoothing steps, and consequently the same cost per cycle iteration, V(0,3) is found to be the most efficient. More generally, among all cycles V(ν_1,ν_2) with $\nu_1 + \nu_2 = \nu$, the option $\nu_1 = 0$ and $\nu_2 = \nu$ appears to be the most efficient. Moreover, we find that the extra cost of more post-smoothing is compensated to a great extent by a better convergence rate. Particularly, moving away from the sweet spot V(0,ν), e.g. by taking V(0,$\nu + 1$) instead, only induces a minor overhead, which grants a pragmatic flexibility in the actual choice of the number of post-smoothing steps.

Figure 3.6 presents the same tests in 3D on the tetrahedral mesh. They also clearly show the superiority of cycles with post-smoothing only. Since both the lexicographic and the antilexicographic Gauss-Seidel smoothers depend on the numbering of the DoFs, we have checked whether these observations depend on a particular numbering of the unknowns. To this end, we have additionally performed experiments with the damped block Jacobi smoother with $\omega = 2/3$ as the under-relaxation parameter (see Figure 3.7). This test leads to the same qualitative conclusions, but with a milder quantitative effect. Based on these measures, we settle for V(0,3) in 2D and V(0,6) in 3D as the most efficient cycles when using block Gauss-Seidel. In Figure 3.8

we present the results of the same scalability tests as in Figure 3.4 using these "optimized" cycles. Besides the expected improved convergence rates, we point out that the iteration count for different polynomial degrees are now almost the same: in all cases, the number of iterations lies in a narrow interval regardless of the polynomial degree. Again the lowest order on the tetrahedral mesh constitutes an exception, since the method still diverges in that case.

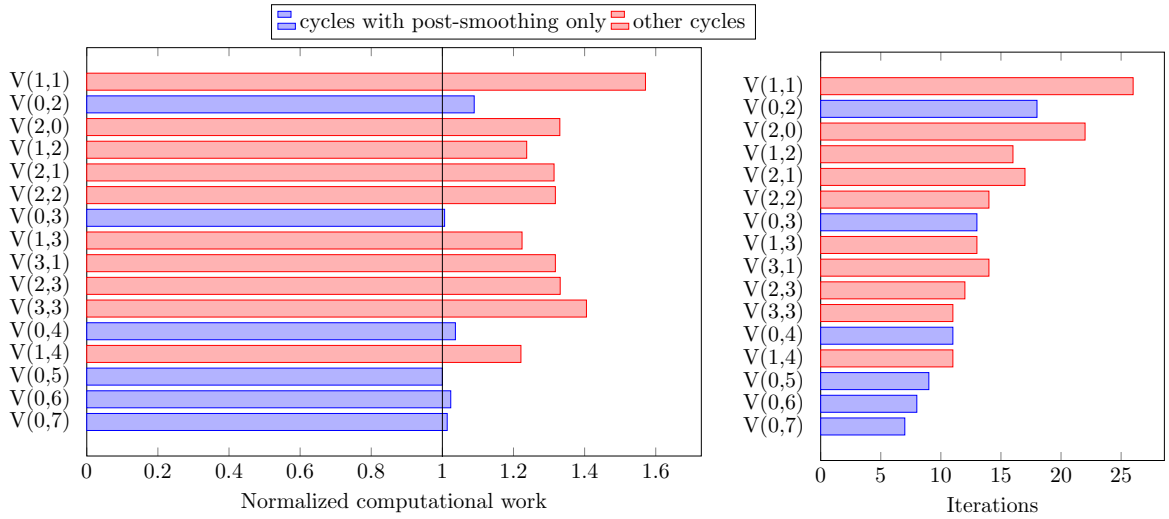


Figure 3.5 Cycle comparison on the 2D test problem, triangular mesh, $N = 512$, $k = 1$ ($\approx 1.6 \times 10^6$ DoFs). The pre-smoother is the lexicographic block Gauss-Seidel, the post-smoother is the antilexicographic block Gauss-Seidel. In the first plot, the numerical values in flops of the computational work are normalized by the lowest one. V(0,1) and V(1,0), being very inefficient in comparison to the others, are not presented here.

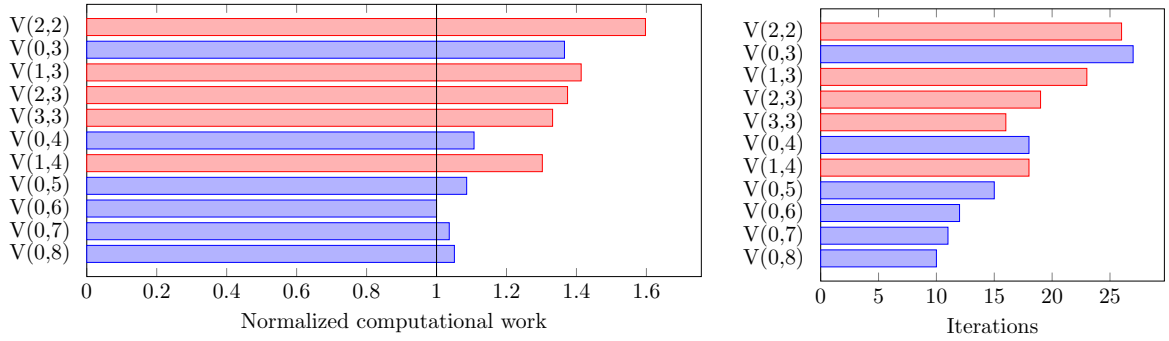


Figure 3.6 Cycle comparison on the 3D test problem, tetrahedral mesh, $N = 32$, $k = 1$ ($\approx 1.2 \times 10^6$ DoFs). The pre-smoother is the lexicographic Block Gauss-Seidel, the post-smoother is the antilexicographic Gauss-Seidel. In the first plot, the numerical values in flops of the computational work are normalized by the lowest one. The first cycles, with less than 3 or 4 total iterations are not efficient and therefore not presented here.

The multigrid algorithm can be used as a preconditioner for Krylov-space methods and in particular for the Conjugate Gradient (CG) method. In the latter case, the algorithm requires formally a symmetric positive definite preconditioner to ensure convergence. Consequently, the choice of a symmetric and therefore suboptimal multigrid cycle seems to be necessary, unless the conditions of [72] are met. A thorough investigation of the present multigrid algorithm as

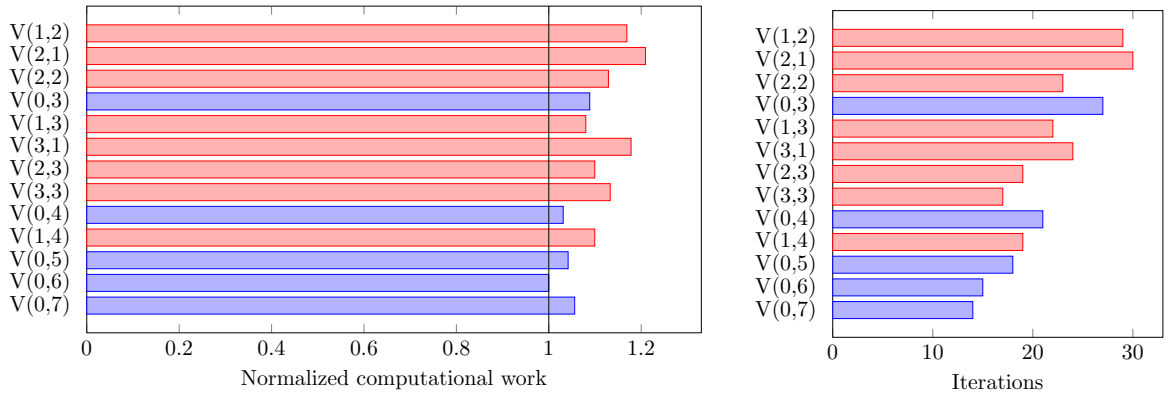


Figure 3.7 Cycle comparison on the 2D test problem, triangular mesh, $N = 512$, $k = 1$ ($\approx 1.6 \times 10^6$ DoFs). The pre- and post-smoothers are the damped block Jacobi smoother with the relaxation parameter 2/3. In the first plot, the numerical values in flops of the computational work are normalized by the lowest one. The first cycles, with less than 3 or 4 total smoothing steps are not efficient and therefore not presented here.

preconditioner is outside the scope of this chapter.

3.2.3 Heterogeneous diffusion

The domain is split into four quadrants as illustrated in Figure 3.9a. The heterogeneity pattern is such that each pair of opposite quadrants have the same, homogeneous, diffusion coefficient. On each homogeneous part Ω_i, $i = 1, 2$, the diffusion tensor is defined as $\mathbf{K}|_{\Omega_i} := \kappa_i \mathbf{I}_d$, where κ_i is a positive scalar constant and $\mathbf{I}_d$ denotes the identity matrix of size d.

Our first test evaluates the convergence rate for varying values of the coefficient ratio $\rho_{\mathbf{K}} := \kappa_1/\kappa_2$ in the range $1 \leq \rho_{\mathbf{K}} \leq 10^8$. The results demonstrate robustness of the algorithm with respect to the heterogeneity. Regardless of the magnitude of the coefficient ratio, the convergence rate remains unchanged and matches that of the homogeneous case; see Figure 3.12a.

In [75], Kellogg studied the analytical solution of a specific case of such a configuration. The source function is set $f \equiv 0$ and non-homogeneous Dirichlet boundary conditions are imposed. The particular solution u exhibits a singularity at the center of the square and has reduced regularity $u \in H^{1+\epsilon}$, $0 < \epsilon \leq 1$. Since the strength of the singularity and thus the regularity $1+\epsilon$ can be adjusted via the coefficient ratio, this problem is often used to benchmark discretizations and solvers. Here we set the parameters of the Kellogg problem such that we have a strong singularity of $\epsilon = 0.1$, corresponding to $\rho_{\mathbf{K}} \approx 161$. The analytical solution u is illustrated in Figure 3.9b, and Figure 3.10 shows the scalability of the multigrid solver and its robustness with respect to the polynomial degree. Here, the V(1,1) cycle is used, but other cycle types exhibit the same properties.

3.2.4 Impact of different choices in the algorithm

3.2.4.1 Alternative prolongation operators

Here we discuss alternatives in the coarse reconstruction of the cell-defined polynomial and in the trace prolongation on the fine faces. Especially, in the definition of $\Theta_{\ell-1}$, reconstructing a

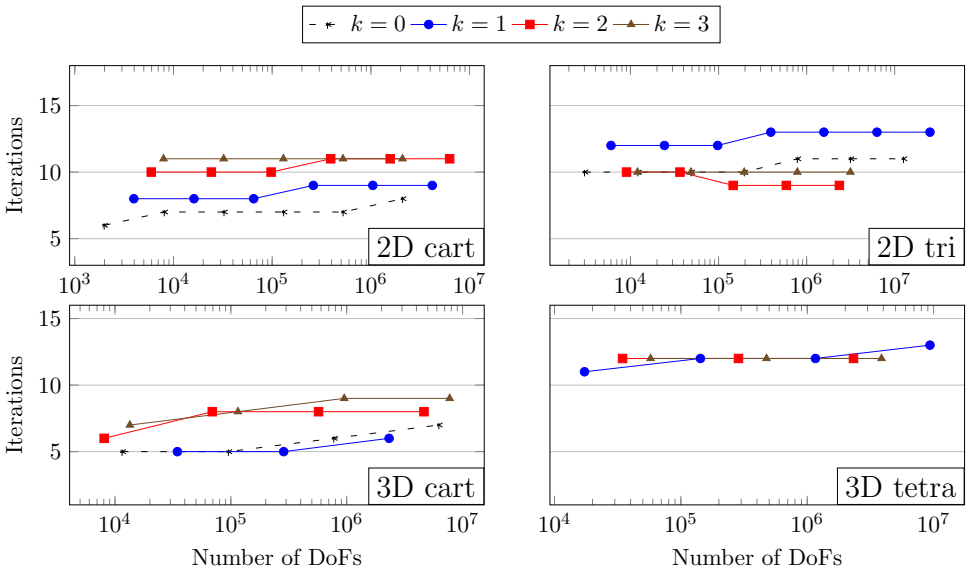
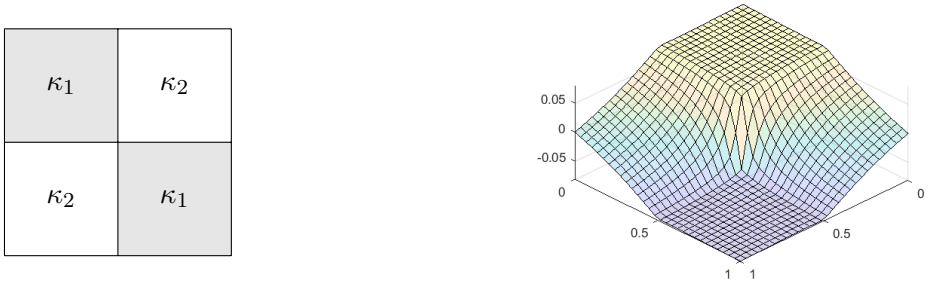


Figure 3.8 Number of iterations to achieve convergence for the homogeneous problem on structured meshes: 2D Cartesian (top left), 2D triangular (top right), 3D Cartesian (bottom left), 3D tetrahedral (bottom right). The V(0,3) cycle is used for 2D problems, the V(0,6) for 3D. The absence of the lowest order case on the tetrahedral mesh is due to the divergence of the multigrid method.



(a) Chiasmus heterogeneity pattern.

(b) Kellogg's solution.

Figure 3.9 (a) Square domain partitioned into four quadrants defining an heterogeneity pattern in the shape of a chiasmus. (b) Analytical solution of the Kellogg problem.

polynomial of higher degree may be optional. As a matter of fact, only solving the cell unknowns by (3.4) and skipping the higher order reconstruction (3.5) could be enough to construct a suitable cell-based polynomial from which to take the trace on the fine faces. In that second step of the prolongation, namely $\Pi_{\ell-1}^{\ell}$, we could also rely on the nestedness of the meshes to identically transfer the polynomials on the coarse faces to the fine grid using the canonical injection[1], instead of taking the average of the traces of the cell-based polynomials on both sides (see (3.6)). Table 3.1 summarizes these options. In order to quantify the impact of the choices made, Figure 3.11 compares the performance in term of scalability of the four option combinations applied to a homogeneous test problem. With the optimal V(0,3) cycle (left plot), the results show the good scalability of all options, with a better convergence rate for the final

[1]Here, the canonical injection refers to the linear operator that identically transfers the elements from one space to a larger one. It is not to be mistaken with the *straight injection* designating, in multigrid terminology, a special type of restriction operator.

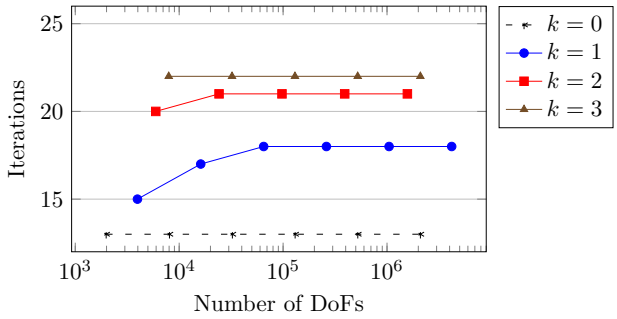


Figure 3.10 Scalability results on the Kellogg problem. Number of V(1,1) iterations to achieve convergence for a growing number of DoFs.

algorithm (about 15% better than with any other option combination). However, the results with the V(1,2) cycle (right plot) indicate that the differences between options may amplify when using non-optimal cycles. Indeed, in this case, it seems that taking the average on both sides instead of using the canonical injection on the faces geometrically shared by the coarse and fine meshes becomes an important criterion to achieve the most scalable behaviour. On the other hand, the reconstruction of higher degree seems to simply improve the convergence rate, with no visible impact on scalability.

	Option label	Description
$\Theta_{\ell-1}$	cell $k+1$	Formula (3.5).
	cell k	Formula (3.4), the higher-order reconstruction (using p_T^{k+1}) is skipped.
$\Pi_{\ell-1}^{\ell}$	average	Formula (3.6).
	injection	The polynomials on the coarse faces are identically transferred to the fine grid using the canonical injection.

Table 3.1 Summary of the 4 options defined for the algorithm.

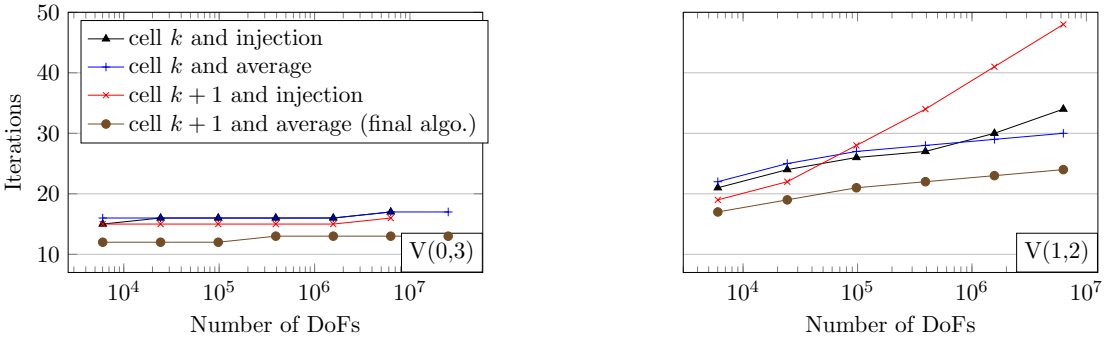
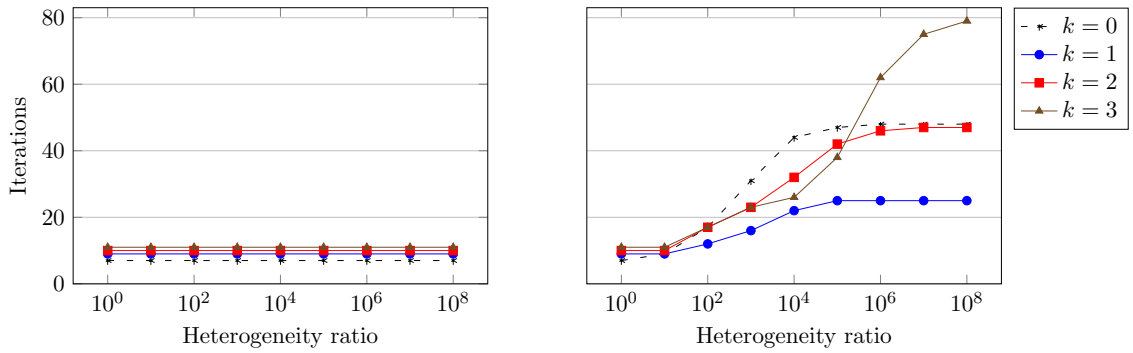


Figure 3.11 Scalability comparison of different versions of the algorithm, applied on the 2D homogeneous problem discretized with the structured triangular mesh for $k = 1$. On the left, the V(0,3) cycle is used, on the right, V(1,2). The various option combinations are labeled according to Table 3.1.

3.2.4.2 Weighting strategy for heterogeneous problems

In the same way, in the case of a heterogeneous problem, we quantify the impact of weighting the cell contribution in $\Pi_{\ell-1}^{\ell}$ proportionally to its diffusion coefficient via (3.8). This strategy is hereafter called *heterogeneous weighting*, as opposed to the alternative that is, given a non-boundary face, to take from each cell of which it is the interface an equally weighted contribution (i.e. to use weighting factors of 1/2). The first thing to be noted is that, if we do not use the heterogeneous weighting (3.8), our algorithm diverges when the heterogeneity ratio $\rho_{\mathbf{K}} \geq 50$. Now, if we use the Galerkin operator instead of the discretized operator on the coarse grids, the algorithm becomes much more robust to high heterogeneity ratios and allows for a quantitative comparison. Figure 3.12 illustrates the differences in the weighting strategies for an increasing heterogeneity ratio, using the Galerkin operator. The heterogeneous weighting ensures perfect robustness with respect to $\rho_{\mathbf{K}}$ regardless of the polynomial degree. But without it, the convergence rate of the algorithm clearly becomes sensitive to the strength of the discontinuity. Moreover, this sensitivity intensifies with the increase of the polynomial degree. We want to clearly state that the sole purpose of this experiment without the diffusion-dependent weights is to highlight their importance. To this end, the Galerkin operator is only used as a means to establish a comparison, and therefore, in the context of this multigrid with all its features, must not be considered more robust to heterogeneous test cases than the rediscretization.



(a) Heterogeneous weighting (3.8)

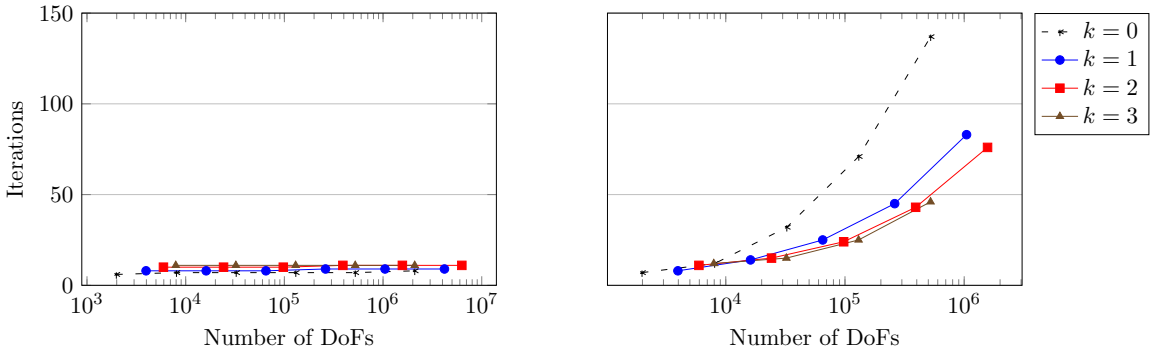
(b) Homogeneous weighting ($w_{T_1F} = w_{T_2F} = \frac{1}{2}$)

Figure 3.12 Robustness of our multigrid algorithm for the heterogeneous 2D test problem with (a) and without (b) the heterogeneous weighting strategy (3.8), in terms of number of iterations to achieve convergence for various orders of magnitude of the heterogeneity ratio $\rho_{\mathbf{K}}$. The square domain is discretized by a Cartesian mesh with $N = 64$, partitioned in four quadrants as described in 3.2.3. The multigrid algorithm uses the Galerkin operator and the V(0,3) cycle.

3.2.4.3 Role of the face coarsening

We now investigate the need for coarsening the faces in the coarsening strategy and show that without doing so, the solver's performance degrades rapidly. In our standard coarsening of uniform Cartesian meshes, two edges in 2D (or four faces in 3D) are ideally combined to become a single one. Consequently, each mesh cell has four edges (or six faces in 3D), and this on all levels. Alternatively, we also have the option that each coarse cell is represented with eight edges, colinear by pairs (see the last (inadmissible) coarsening strategy described by Figure 3.1). One of the effects of this alternative coarsening is that the number of unknowns per level is reduced less

aggressively. Indeed, asymptotically, the number of unknowns is only decreasing by a factor of two per level (whether in 2D or 3D), whereas the standard coarsening rate is four in 2D and eight in 3D. For this nonstandard coarsening, the coarse grid spaces are then enlarged. In a Galerkin setting, the usual coarse grid spaces are subspaces of these enlarged coarse grid spaces, and as a consequence, we expect the two-grid convergence rates to improve. This is indeed verified, under the condition that the Galerkin operator is invertible (which is not necessarily the case with this coarsening strategy). However, this improvement cannot be observed in V-cycles with more levels, and not at all if we step out of the Galerkin setting to use the rediscretized operator. This observation is caused by the neglect of one important condition in multigrid convergence: as the smoother only efficiently reduces the high-frequency components of the error, it is crucial that the remaining low frequencies be seen as higher frequencies in the coarser spaces. And this can happen only if the geometric entities on which the DoFs lie are coarsened in between levels. As we work on the condensed system, whose unknowns rest on the mesh skeleton, this condition means that the *faces* should be coarsened. Only in this appropriate setting can the smoother at each level successfully target its own range of frequencies. If the faces are not coarsened, the smoother on the coarser grids spends most of its effort to compute irrelevant solution modes, which causes the convergence rate to deteriorate. Figure 3.13 illustrates this convergence degradation as the number of DoFs (and therefore the number of levels) grows. Note that the overall performance of the solver also reduces due to the increased amount of work and the slower coarsening.



(a) With face coarsening

(b) Without face coarsening

Figure 3.13 On the right, scalability test of the algorithm using a coarsening strategy which does not coarsen faces. The test problem is the homogeneous problem on the 2D Cartesian mesh, solved by the V(0,3) cycle of our multigrid algorithm, using the coarsening strategy described on the right-hand side of Figure 3.1. In the left plot, standard coarsening is used in comparison.

3.2.5 Unstructured meshes

The convergence of a geometric multigrid method relies on the approximation properties of the underlying discretization scheme through its coarse grid correction step. Furthermore, most discretization schemes are sensitive to the quality of the mesh, degrading in presence of flattened or stretched elements. As a direct consequence, the convergence of a multigrid method is often also sensitive to the mesh quality. The reader may refer to [2] for further details about the sensitivity of the HHO method to the element shapes. Moreover, even when starting from a

good-quality coarse (resp. fine) mesh, the construction of a suitable mesh hierarchy may be difficult. In such an unstructured mesh hierarchy, the distortions of the elements must be kept under control when refining (resp. coarsening) the mesh. This is less a problem in 2D, but it remains difficult in 3D [32, 110]. Tetrahedral mesh refinement preserving mesh quality is still a topic of recent research [99, 117]. These difficulties help explain why more costly cycles may be required for highly unstructured 3D meshes.

In all the following tests, the mesh hierarchy is built by successive refinement of a coarse mesh. In 2D, we find that the convergence on unstructured meshes is qualitatively and quantitatively comparable to the convergence on structured meshes. Here the V(0,3) cycle is found to be sufficient. In 3D, the lower quality of tetrahedral meshes forces us to use costlier cycles. First of all, since the meshing method used to discretize the different refinements of the cubic domain (described in 3.2.2) is not applicable on general geometries, we investigate the impact of another, more generally applicable, tetrahedral refinement method. The method used is inspired by Bey's refinement algorithm [16]. Figure 3.14 shows that trading the refinement strategy for one that does not conserve the topology of the tetrahedra causes a serious performance degradation, which can be mitigated at the cost of substantially more smoothing steps. In order to quantify the loss of performance with respect to the loss of mesh quality upon refinement, we use the regularity indicator ϱ_h defined as

$$\varrho_h := \max_{T\in\mathcal{T}_h} \varrho_T \qquad \text{with} \qquad \varrho_T := \frac{d_T}{h_T} \quad \forall T \in \mathcal{T}_h, \tag{3.9}$$

where d_T denotes the diameter of the largest ball included in T. A good regularity parameter is close to 1 while a bad one may be close to 0. As a reference, a cube has a regularity parameter of 0.64. Now, the original coarse mesh, namely the cubic domain divided into 6 geometrically identical tetrahedra, has a ϱ_h of 0.21. The so-called Cartesian tetrahedral refinement method described in 3.2.2 does not change the geometry of the refined tetrahedra, so the regularity parameter is conserved on the refined meshes, and we have seen that the V(0,3) cycle exhibits scalable behaviour and fast convergence. On the other hand, our custom Bey's method degrades the mesh quality during the first refinement (but not during the next ones), yielding in this case a regularity parameter of 0.14, which corresponds to a loss of 1/3. Figure 3.14 shows that over three times more smoothing steps are required to compensate for the poorer mesh quality and to recover comparable performance.

Using this time the custom Bey's refinement method, a cycle comparison in the model of Figure 3.6 finds V(0,10) as the most efficient cycle in this context. V(0,10) is therefore used for the test of the highly unstructured 3D mesh presented in Figure 3.15. In this test case, the initial coarse mesh has $\varrho_h = 0.10$, which degrades to 0.06 after refinement. The poor initial mesh quality and the further degradation of 40% result in sub-optimal performance: in spite of the large number of smoothing steps, the convergence degrades for larger meshes. Thus, the desired h-independent convergence cannot be confirmed. Table 3.2 summarizes the impact of the mesh quality and the refinement method on the performance of the multigrid algorithm.

Having stated the sensitivity of the algorithm to the mesh quality, along with the known problem of refining (resp. coarsening) unstructured tetrahedral meshes without (too much) degradation, it is important to recall HHO as a polyhedral method. In this context, taking

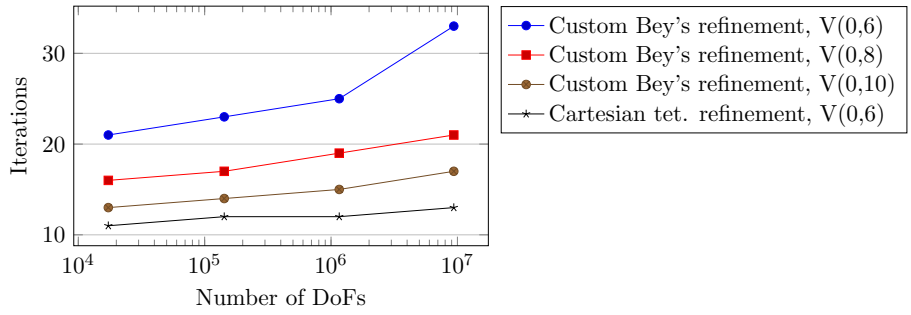


Figure 3.14 Comparison on the cubic domain of the Cartesian tetrahedral refinement method described in 3.2.2 and the custom Bey's refinement method, with different cycles, in terms of scalable performance.

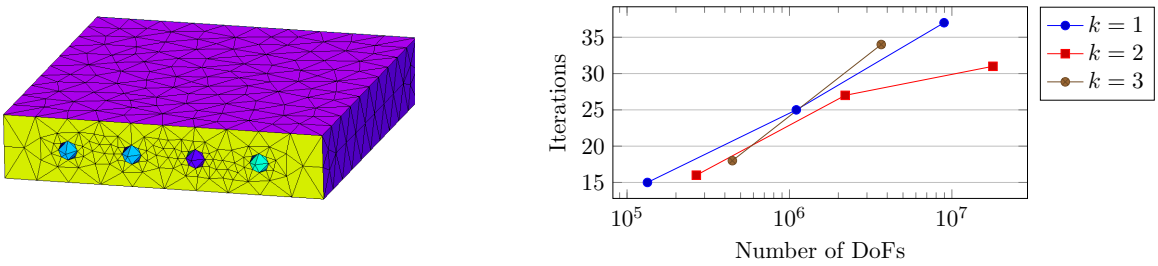


Figure 3.15 On the left: tetrahedral mesh of a plate with four polyhedral holes going through the object from one side to the other. On the right: Scalability results of the multigrid algorithm, using the V(0,10) cycle and Bey's tetrahedral refinement. The case $k = 0$ is divergent.

Domain	Mesh	Refinement method	ϱ_1	ϱ_L	Quality loss	Required cycle
Cube	Cartesian	Standard	0.64	0.64	0%	V(0,3)
	Tetrahedral	Cartesian tet.	0.21	0.21	0%	V(0,6)
	Tetrahedral	Custom Bey	0.21	0.14	33%	V(0,10)
Plate w/ holes	Tetrahedral	Custom Bey	0.10	0.06	40%	> V(0,10)

Table 3.2 Numerical values of the mesh quality and the quality loss caused by the refinement method, along with their consequences on the minimum cycle required to retrieve a close-to-optimal convergence rate. ϱ_1 (resp. ϱ_L) corresponds to the quality indicator (3.9) of the initial coarse mesh (resp. of the fine mesh obtained by the application of the refinement method). The domain "Plate w/ holes" refers to Figure 3.15.

advantage of the flexibility of general polyhedral meshes is one way of overcoming these difficulties and keep the mesh quality under control. For the same purpose, the use of non-nested meshes, discussed in the next paragraph, can constitute an additional tool.

3.2.6 Non-nested meshes

In this section we relax the requirement that the meshes must be nested. Although no rigorous redefinition of the algorithm is made here, we will explain the changes that are necessary for non-nested meshes.

- Coarsening strategy: The hierarchy $(\mathcal{T}_\ell)_{\ell=1\dots L}$ is now non-nested, in which we consider

two successive levels ℓ (fine) and $\ell-1$ (coarse). Figure 3.16 presents an example of two such levels. For a fine element $T \in \mathcal{T}_\ell$, we define its associated coarse element $T^{\ell-1} \in \mathcal{T}_{\ell-1}$ as the one coarse element that contains "most of" T, and we make the assumption that the definition used for "most of" ensures existence and uniqueness of $T^{\ell-1}$ for all $T \in \mathcal{T}_\ell$. In our implementation, we have considered that a coarse element contains "most of" the fine element T if it contains its centroid. Note that when the meshes are nested, $T^{\ell-1}$ simply reduces to the coarse element that geometrically embeds T. Now, given a fine face $F \in \mathcal{F}_\ell$, F is either absent from the coarse mesh (black dotted edges in Figure 3.16), or still geometrically present in a coarsened form, which we denote by $F^{\ell-1}$ (solid edges). Non-nestedness implies that F may no longer be geometrically embedded in $F^{\ell-1}$ (like blue dotted edges are not embedded in the coarse red ones). If so, we talk of *non-nested faces*. The assumption that discontinuities in the diffusion coefficient do not happen inside elements consequently implies that the faces describing such discontinuities must still be nested.

- Trace on the fine faces: Although the reconstruction from the coarse faces to the coarse cells remains the same, inasmuch as the fine mesh is not involved, taking the trace of a coarse cell-defined polynomial on a fine face which is not fully included in the closure of the coarse cell is no longer possible. As example, consider in Figure 3.16 one coarse triangle with a red edge and pointing to the center of the figure: the blue edges corresponding to the refinement of the red edge are fully outside the coarse triangle, while two interior fine edges (which are simply not present on the coarse mesh in a coarsened form) are only partially included in the coarse triangle. In this case, we consider that the cell-defined polynomial is extended outside of the cell boundaries to overlap the targeted fine cells, which allows to take the trace on those faces. Considering $v \in U_{\mathcal{T}_{\ell-1}}^{k+1}$ and a face $F \in \mathcal{F}_\ell^{\mathrm{I}}$, formula (3.6) then becomes

$$(\Pi_{\ell-1}^{\ell} v)_{|F} := w_{T_1F}\, \pi_F^k(\mathfrak{v}_{|T_1})_{|F} + w_{T_2F}\, \pi_F^k(\mathfrak{v}_{|T_2})_{|F},$$

  where $\mathfrak{v}$ is the extension of $v_{|T^{\ell-1}}$ to $T_1 \cup T_2$.

These remarks are equivalent to stating that non-nested faces correspond to slight perturbations of nested ones.

Our algorithm is tested on a hierarchy of non-nested 2D meshes obtained by successive refinements for a domain containing a disk. The curved boundary of this disk is approximated more accurately with each refinement, see Figure 3.17. The results of Figure 3.18 show that the algorithm does not suffer from the non-nestedness in this form.

This observation is important regarding the design of coarsening strategies for unstructured meshes. In this case, agglomerating faces that are close to being coplanar (close to colinear in 2D) to form coarser ones seems possible. However, non-nested meshes must be employed with great care in heterogeneous domains. The approximation of curved boundaries depends on the granularity of the mesh (see the circle approximations in Figure 3.17). For the method to preserve its performance and be oblivious of the coefficient jump at the interface, it is crucial that no fine element of one physical region be associated to a coarse element of another, so that no element-defined data can be transferred from one region to the other upon prolongation.

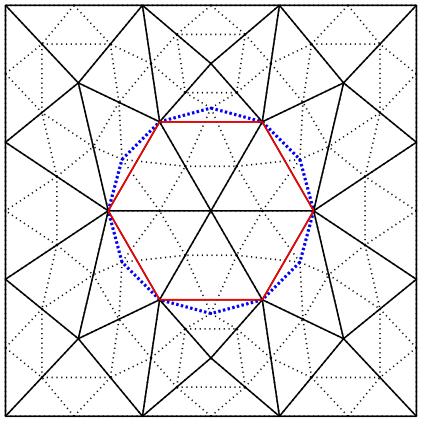


Figure 3.16 Example of admissible non-nested coarsening. The fine mesh is represented in dotted lines, the coarse mesh in solid lines. The non-nestedness is highlighted by colors: the blue fine edges are coarsened into the red ones.

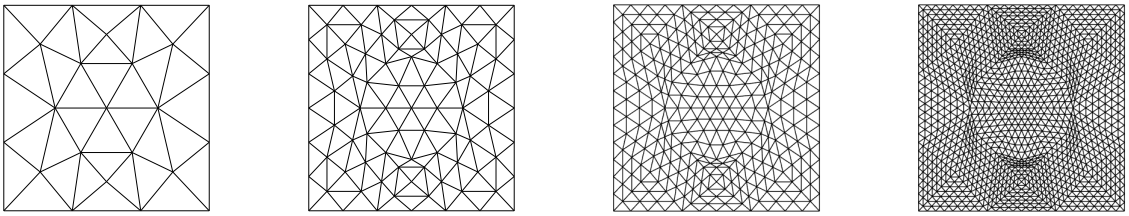


Figure 3.17 Geometry of a disk embedded in a square, coarsely meshed and successively refined by a splitting method. Each refinement approximates more accurately the disk's shape, consequently yielding a mesh hierarchy that is non-nested at the disk's boundary. The non-nestedness of the first two meshes is highlighted in Figure 3.16.

With this constraint, the problem where different constant isotropic coefficients are set in the elements located inside and outside of the circle approximation at every level of Figure 3.17 can be solved efficiently, with results comparable to Figure 3.18.

3.3 Conclusion

The multigrid solver proposed and developed in this chapter is fast, scalable with respect to the mesh size, and robust to heterogeneity, provided a good enough mesh is used. Moreover, these desirable properties hold also for higher polynomial order. Adding a p-multigrid method to lower the degree down to $k = 1$ before using the present h-algorithm is another way to handle high orders. A comparison of both approaches in terms of overall cost deserves a dedicated study which will be part of future work.

The algorithm proposed works for general meshes. However, the need to coarsen the faces can make the design of admissible coarsening strategies more difficult. While excellent convergence rates are observed for canonical nested mesh hierarchies, additional complexity must be expected when the faces are not co-planar. We have shown that fine faces that are sufficiently close to being co-planar may be approximated on coarse meshes by straight coarse faces without loss of performance. This experiment points out non-nested meshes as a possible path in the search for coarsening strategies in which the faces are also coarsened. Moreover, freeing ourselves from

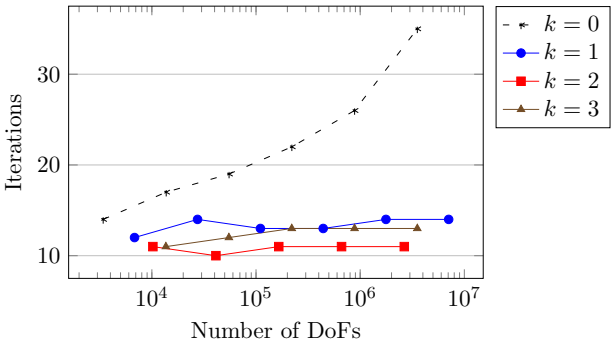


Figure 3.18 Number of iterations to achieve convergence for the homogeneous problem on the non-nested mesh hierarchy described by Figure 3.17, solved by the V(0,3) cycle of our multigrid algorithm.

the nestedness constraint offers more flexibility to ensure the conservation of the mesh quality across levels. This path is therefore the one we choose to explore in Chapter 4.

Chapter 4

Extension of the multigrid method to non-nested meshes

Computers are useless. They can only give you answers.

Pablo Picasso (1881-1973)

The content of this chapter was published in the *International Journal for Numerical Methods in Engineering* [51].

Used as a solver, the multigrid method developed in the previous chapter exhibits a near-perfect h-independent convergence rate on structured 2D and 3D meshes, as well as robustness with respect to the polynomial degree. However, on complex 3D geometries requiring highly unstructured meshes, the average performance may degrade rapidly with the problem size. Indeed, the algorithm requires a hierarchy of nested meshes (although it is experimentally shown that slightly non-nested meshes do not affect the global performance), and such hierarchies are usually obtained from successive refinements of an initial coarse mesh. Working with a hierarchy of nested meshes obtained from successive refinements of an initial coarse mesh has, however, some drawback. The first one is the loss of the flexibility offered by unstructured meshes to accurately approximate complex regions, as the initial coarse mesh must already be a good approximation of the geometry. Having stated this, provided an efficient coarse solver, starting from an initial coarse grid that is nonetheless fine enough to adequately capture the geometry can still offer possibilities of very efficient multigrid implementations, such as Hierarchical Hybrid Grids [15, 78]. And to help in this approach, recent techniques [14] based on polynomial mappings of the elements allow to better approximate curved boundaries upon each refinement. The second drawback of the nested approach is linked to the fact that, in 3D, most methods of tetrahedral subdivision generate elements of degraded shape quality [32, 109, 110, 117], which most often affects the accuracy of the approximate solution as well as the performance of the linear solvers. The nested multigrid method presented in Chapter 3 indeed shows high sensitivity to the mesh regularity and therefore poor convergence on unstructured meshes obtained by tetrahedral refinement. Exploring non-nested grids as an alternative to hierarchies produced

by mesh refinement is the purpose of the present work. To that end, and before describing our adaptation of the nested algorithm to non-nested grids, we state that existing non-nested multigrid methods have been devised for the same motives as ours, although none of them applicable to trace systems of hybrid discretizations: continuous Finite Element Methods (FEM) are mostly targeted [1, 27, 61, 63, 73] and a recent one has been designed for Discontinuous Galerkin (DG) methods [4], closer to our setting.

Compliance to non-nested settings is performed by inserting an additional step in the definition of the prolongation operator: starting with polynomials lying on the coarse faces, the nested version begins with the reconstruction of a broken *element*-defined polynomial on the coarse mesh, which we propose (as already performed in the DG context by [4]) to orthogonally project in L^2-norm onto the non-nested fine mesh. In a next step, the trace of the result can be computed on the fine faces. The numerical evaluation of this L^2-orthogonal projection operator hinges on the projection of the local coarse basis functions onto the fine bases, i.e. on the computation of the L^2-inner products of the coarse and fine basis functions over the fine elements. As a direct consequence, the local definition of the functional bases makes the intersections of coarse and fine elements the respective integration supports to these inner products. The prerequisite of computing the geometric intersections between coarse and fine elements can occur as computationally prohibitive. So instead of this exact computation, we propose the implementation of an approximate operator that does not require intersections. We also refer to a previous work [52] that tackles the practical computation of L^2-orthogonal projections as well.

Our non-nested multigrid method is validated by numerical tests using independent retriangulations of the domain at every level. However, in practice, building a high-quality mesh for a real, industrial case study can already be an arduous task, which may occupy a meshing engineer for several months. Requiring multiple high-quality meshes of the same geometry at different granularities in order to feed a multigrid solver is then not always conceivable. From the user's standpoint, providing the solver with the sole fine mesh is a preferable option. To this end, we also want to pave the way for the construction of suitable coarsening strategies for this type of multigrid method. We will provide guidelines by designing a full example of this strategy implementing the relevant features. In particular, in a face-defined multigrid method, as the smoother operates on the mesh skeleton, the efficient reduction of the low-frequency components of the error relies on accessing coarse representations of the face-defined polynomials. This implies that *faces* must be coarsened between levels (cf. Section 3.2.4.3), which is a new constraint imposed to any suited coarsening strategy. Unfortunately, usual agglomeration methods [79], typical candidates for the coarsening of unstructured meshes, do not meet this requirement. More generally, methods that conserve embedded meshes do so by conserving fine faces on the coarse mesh, which goes against this new constraint. This observation points out non-nested methods as suitable alternatives. As such, strategies which build coarse tetrahedra from the fine tetrahedra's vertices [87] seem well-adapted to our problem, as long as the shape quality of the coarse elements is sufficient. This condition drives us towards *polytopal* coarse elements in order to make use of their shape flexibility to construct high-quality coarse meshes. Methods based on Voronoi diagrams [85, 115] fulfill that purpose, as well as our face coarsening constraint. Nonetheless, we choose to propose another option based on element agglomeration. Indeed, additionally to being easy to understand and implement, these methods also benefit from simple

derivations to anisotropic grids [64]. Thus, we suggest to add to any chosen agglomeration process a step of *face collapsing*, already used in different fashions [97, 106]. Moreover, we will show that agglomeration-based methods ease the construction of our approximate L^2-orthogonal projection.

Having established the reasons driving us to non-nested settings, this chapter is organized around the adaptation of the nested solver of Chapter 3 and its efficient implementation for the purpose of its practical use for the solution of condensed linear systems arising from the HHO discretization of scalar elliptic equations on complex geometries. Thus, we begin, in Section 4.1, we recall the ingredients in the construction of the nested multigrid method and extend its range of application to non-nested meshes through the use of the L^2-orthogonal projection. This approach is validated by numerical tests using independent retriangulations at every levels. Insofar as the newly used L^2-orthogonal projection depends, in its exact evaluation, on the expensive computation of the intersections between coarse and fine elements, we devote Section 4.2 to the definition of a cheaper approximate operator. We also propose in Section 4.3 a suited non-nested and polytopal coarsening strategy based on element agglomeration and face collapsing. Finally, numerical tests are presented in Section 4.4, which demonstrate the optimality of our multigrid method, used as a solver, on 2D and 3D diffusion problems.

4.1 Multigrid method on non-nested meshes

Placing ourselves in the same context as the previous chapter, we consider the diffusion problem (2.1), its HHO discretization of degree k (2.15), and aim at solving the statically condensed system (2.20). Extending the range of application of the multigrid method described in Chapter 3, we now assume a *non-nested* hierarchy of polytopal meshes.

Consistently with the multigrid literature, we use the subscript $\ell \in \{1, \dots, L\}$ to index the levels in the mesh hierarchy, and we denote by h_ℓ the meshsize at level ℓ. We sort the levels so that L refers to the finest mesh, whereas 1 corresponds to the coarsest one, i.e. $h_L < h_{L-1} < \cdots < h_1$. Importantly, we further assume that, from a level $\ell > 1$ to the immediately coarser one $(\ell - 1)$, not only are the elements coarsened, but so are the faces. To ease the notation, the subscripts h_ℓ may be replaced with ℓ, so that the mesh hierarchy may be denoted by $(\mathcal{T}_\ell, \mathcal{F}_\ell)_{\ell=L\dots1}$.

4.1.1 Prolongation operator

Considering two successive levels $\ell > 1$ (fine) and $\ell - 1$ (coarse), we define the prolongation operator $P\colon U^k_{\mathcal{F}_{\ell-1},0} \to U^k_{\mathcal{F}_\ell,0}$ as the composition of three operators,

$$P := \Pi_\ell \circ J^\ell_{\ell-1} \circ \Theta_{\ell-1}. \tag{4.1}$$

In this formula,

- the coarse level potential reconstruction operator $\Theta_{\ell-1}\colon U^k_{\mathcal{F}_{\ell-1},0} \to U^{k+1}_{\mathcal{T}_{\ell-1}}$ reconstructs, from the face unknowns, a broken polynomial of degree $k+1$ on $\mathcal{T}_{\ell-1}$. As its definition does not change w.r.t. the nested algorithm, we refer the reader to Section 3.1.3.1.
- $J^\ell_{\ell-1}\colon U^{k+1}_{\mathcal{T}_{\ell-1}} \to U^{k+1}_{\mathcal{T}_\ell}$ is now taken equal to the L^2-orthogonal projection to include the non-nested case.

- The trace operator $\Pi_\ell \colon U_{\mathcal{T}_\ell}^{k+1} \to U_{\mathcal{F}_\ell,0}^{k}$ maps the polynomials of degree $k+1$ defined on the fine elements to a broken polynomial of degree k on the fine skeleton. Although in the nested version, the trace on the fine faces could directly be computed from the *coarse* cell-defined functions because of the embedding of the meshes, it is now computed from the *fine* cell-defined functions after performing the L^2-projection. However, the semantics of this step remains unchanged. Consequently, a formula similar to (3.6) defines Π_ℓ. Namely, for all $v \in U_{\mathcal{T}_\ell}^{k+1}$, $F \in \mathcal{F}_\ell$, and the two disctict elements T_1, T_2 of $\mathcal{T}_F \subset \mathcal{T}_\ell$ in the case where $F \in \mathcal{F}_\ell^{\mathrm{I}}$,

$$(\Pi_\ell v)_{|F} := \begin{cases} w_{T_1F}\, \pi_F^k(v_{|T_1})_{|F} + w_{T_2F}\, \pi_F^k(v_{|T_2})_{|F} & \text{if } F \in \mathcal{F}_\ell^{\mathrm{I}}, \\ 0 & \text{otherwise,} \end{cases}$$

with the scalar weights defined, for $i \in \{1,2\}$, as

$$w_{T_iF} := \frac{K_{T_iF}}{K_{T_1F} + K_{T_2F}}.$$

4.1.2 Multigrid components

Having defined the prolongation operator, the restriction is set to its adjoint in the usual fashion, i.e. with respect to the coarse and fine face-defined global inner products. As such, the matrix of one transfer operator in the chosen polynomial basis is the transpose of the other. Coarse grid operators come from the rediscretization of the problem on the coarse meshes. In order to relax together all the unknowns related to the same local polynomial, block versions of standard fixed point iterations (like damped Jacobi or SOR) should be used, with a block size corresponding to the number of DoFs per face. For instance, for a 3D problem with the polynomial degree $k = 1$ chosen for the face-defined polynomials, the blocks will be of size 3×3. For our numerical tests, the relaxation method is set to the block Gauss-Seidel iteration. In order to keep the computational cost as low as possible, we use the post-smoothing-only cycles that were found to be the most efficient in terms of total theoretical work or CPU time (cf. Section 3.2.2.2). In particular, we use V(0,3) in 2D and V(0,6) in 3D. Finally, on the coarsest level, the system is exactly solved by a direct method.

4.2 Approximation of the L^2-orthogonal projection

In the second step of the prolongation, the global broken polynomial, reconstructed on the coarse mesh from the DoFs on the coarse faces, is projected onto the fine mesh via the L^2-orthogonal projection operator denoted $J_{\ell-1}^{\ell} \colon \mathcal{T}_{\ell-1} \to \mathcal{T}_\ell$. The construction of this projection requires the computation of the L^2-inner products of all pairs of coarse/fine basis functions over the fine elements. Locally, one such integral can only be non-zero if the supports of the fine and coarse basis functions intersect, i.e. if the respective fine and coarse elements overlap. In this case, the non-zero part of the integral is restricted to the intersection of the fine and coarse elements.

For any element T, we denote by $\mathcal{B}_T$ the selected basis for $\mathbb{P}^k(T)$. For all $T_c \in \mathcal{T}_{\ell-1}$ and $T_f \in \mathcal{T}_\ell$, we then have

$$(\varphi_c, \varphi_f)_{T_f} = (\varphi_c, \varphi_f)_{T_c \cap T_f} \qquad \forall (\varphi_c, \varphi_f) \in \mathcal{B}_{T_c} \times \mathcal{B}_{T_f}. \tag{4.2}$$

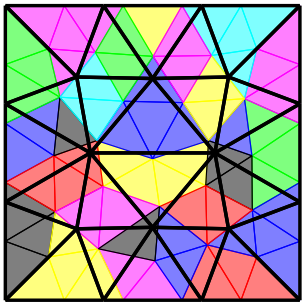
Consequently, the exact construction of $J_{\ell-1}^{\ell}$ preliminary requires the computation of the geometric intersection of every pair of overlapping fine/coarse elements. We can see two drawbacks to this method: firstly, the computation of the intersections adds a costly load to the computational burden. Although the actual overhead depends on the efficiency of the algorithm and its implementation, our numerical tests based on the state-of-the-art geometric library CGAL [66] find it too heavy for practical use. Secondly, the intersection of usual element shapes, even plain simplices, generally yields a polytope. In practice, if only simplicial meshes are used, one might want to avoid introducing other shapes, over which integral computation may be more expensive.

Given these practical limitations, we introduce a mechanism to compute the operator $J_{\ell-1}^{\ell}$ in a cheaper and simpler way, though approximately, by avoiding computing element intersections altogether. It is based on the following approximation (exact when the meshes are nested): For any pair of coarse and fine elements (T_c, T_f),

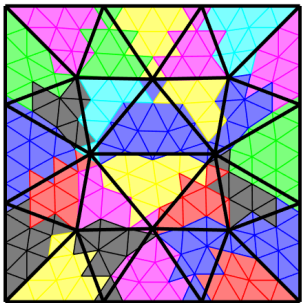
$$T_c \cap T_f \approx \begin{cases} T_f & \text{if } T_c \text{ is the coarse element which "embeds } T_f \text{ the most"}, \\ \varnothing & \text{otherwise.} \end{cases} \tag{4.3}$$

According to the element shapes, determining which coarse element embeds the largest part of a given fine one may have multiple interpretations, which may also be implemented in different fashions. For simplicial elements, it suffices to choose the coarse element containing the barycenter of the fine one.

Following this model, each integral over an intersection is either discarded or computed over the whole fine element, in which case the support of the coarse basis function is implicitly extended to include the entire fine element. The number of integrals to compute is then reduced to the number of fine elements.



(a) Distribution of the fine elements to their "closest" coarse element.



(b) Distribution of the fine elements' sub-triangles to their "closest" coarse element.

Figure 4.1 (a) shows the superposition of coarse and fine triangular meshes. The coarse edges are represented by thick black segments. The fine triangles are colored according the coarse triangle they are "closest" to, i.e. the one containing their barycenters. (b) shows the same colored partioning, this time for the fine elements' sub-triangles.

The validity of this method hinges on the assumption that, if the solution is *smooth enough* and if the fine elements *do not stick too much out* of the coarse element they are associated to, global approximation properties are preserved. Figure 4.1a shows on a triangulated 2D

example how the fine elements are distributed among the coarse ones and how much they can stick out. Graphically, the more the colored clusters of fine elements resemble their respective coarse elements, the less the coarse basis functions must be extended outside of their supports, thus losing accuracy, and therefore the better the approximation of the integrals. Especially, the higher the polynomial degree, the greater the negative impact of the support extension. Outside the element, a high degree polynomial may be an especially poor approximation of the solution. For the same reason, local smoothness of the solution is required, which emphasizes the restriction not to have elements crossing the boundaries of physical subdomains, i.e., where coefficient discontinuities can occur.

Our numerical tests show that in 2D, on the triangulated square, the approximate operator constructed by this method exhibits good enough accuracy to reproduce the results obtained with the exact computation only for $k \leq 1$, and fails to be a good approximate for higher orders. In 3D, the method is unsuccessful for all orders. The method is improved the following way: instead of approximating intersections by an all-or-nothing result (either the whole fine element or the empty set) as in (4.3), we can increase the granularity by subdividing the fine elements, and distribute the subshapes among the coarse elements the same way as before. Assuming, for any fine element $T_f \in \mathcal{T}_\ell$ a given subdivision $\mathrm{Sub}(T_f)$, and denoting by $\mathrm{closest}(\cdot)$ the function associating to any element or subelement its "closest" coarse element, we then define the new approximate intersection as

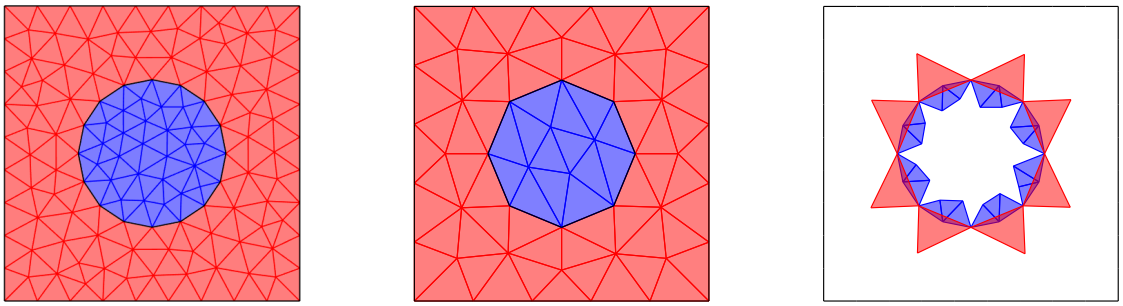
$$T_c \cap T_f \approx \bigcup_{\substack{t \in \mathrm{Sub}(T_f) \\ \mathrm{closest}(t) = T_c}} t \,. \tag{4.4}$$

Figure 4.1b shows how sub-triangles obtained by connecting the middle-edges of the fine elements now approach the coarse ones.

It is clear that the quality of the approximation depends on the granularity of the subdivision. A fortiori, the higher the polynomial degree, the finer the subdivision must be. For that purpose, the fine elements can then be refined multiple times for even better accuracy, though at the cost of an increasing number of integrals to compute. However, the approximate operator derived from the fine elements being subdivided *only once* is found to be sufficient to achieve good multigrid results in 3D and for low polynomial orders. We refer to Section 4.4.2 for the details of the numerical tests.

Remark 3. *(Heterogeneous case with curved region boundary) If the geometry is partitioned into multiple physical regions with curved boundary, the different levels of grid may approximate this boundary in a non-nested way. Then, imposing that jumps in the coefficient do not occur inside elements inevitably leads to coarse elements overlapping fine ones of different regions. Figure 4.2 illustrates a two-region domain separated by a circular interface. The approximation of the circle and therefore the discrete delimitation of the regions depend on the granularity of the mesh. Figures 4.2a and 4.2b, respectively, show a fine and coarse mesh obtained by Delaunay triangulation, while Figure 4.2c shows how coarse elements belonging to the red region overlap fine triangles of the blue region. These cases must be handled with great care. In such a configuration, it is crucial that values of DoFs belonging to one region should never be transferred to another one. The L^2-orthogonal projection must then keep this separation. Indeed, the solution cannot be locally represented with accuracy unless it is kept smooth inside elements. In other words, the*

intersection between a coarse element and a fine one that belong to two different regions must never be computed, and be assimilated to the empty set, even though, in fact, they geometrically overlap. Infringing this guideline results in a quick deterioration of multigrid convergence, and divergence can even occur for small-size problem provided a large jump in the coefficient.



(a) Fine mesh (b) Coarse mesh (c) Superposition at the interface

Figure 4.2 (a) (resp. (b)) presents a fine (resp. coarse) mesh obtained by Delaunay triangulation of a square domain embedding a round physical region. In (c), the red coarse elements overlapping fine blue ones are plotted.

4.3 Agglomeration-based coarsening strategy with face collapsing

In this section, we lay the foundations of an abstract coarsening strategy for polytopal meshes that also coarsens faces. The steps are descibed by Algorithm 4.1. We recall that in this abstract setting, the generic term 'face' refers to the interface between elements (it being an actual face in 3D or an edge in 2D), and the term 'neighbours' refers to a pair of elements sharing a face. Step 1 is standard and defines any agglomeration method. The face coarsening comes from Step 2, where multiple fine faces are collapsed into a single coarse one. Figure 4.3 illustates the result of successive coarsenings in 2D. Recall that the created polygons are not necessarily convex. Also, note that this algorithm does not ensure that every fine face is either removed (being interior to an agglomerate) or coarsened (by face collapsing): some fine faces may find themselves unaltered by the process. However, numerical experiments show that the number of unaltered faces between two successive levels decreases with the number of times the coarsening strategy is applied, i.e. the number of levels built.

Algorithm 4.1 Abstract coarsening strategy with face collapsing

Step 1. Agglomerate each fine element with its non-already agglomerated neighbours to form one polytopal coarse element.

Step 2. Collapse into one single coarse face the interfaces between two neighbouring coarse elements that are composed of multiple fine faces.

Remark 4. *(Preventing domain erosion) The face collapsing step removes vertices from the mesh. In order to keep the domain from "eroding" at its corners (whether at domain boundaries or at interfaces between inner regions), one must make sure that the vertices that describe the*

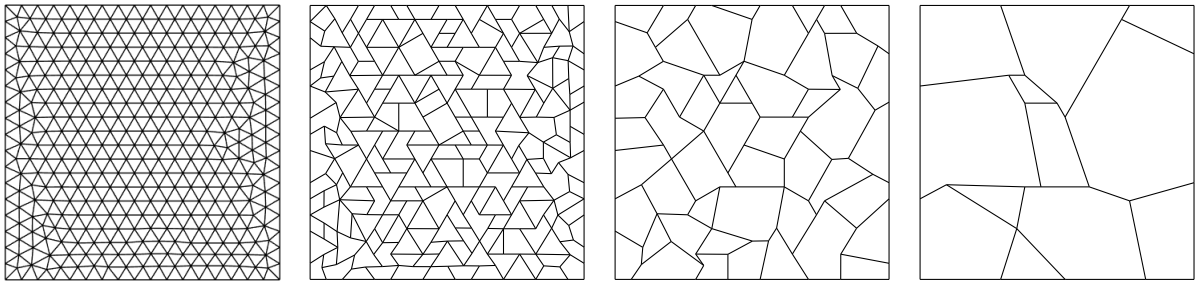


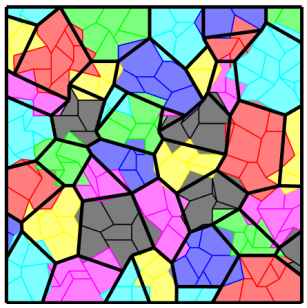
Figure 4.3 Successive coarsenings of a triangulated square.

domain geometry do not vanish in a face collapsing operation. To do so, it suffices to prevent the faces sharing such vertices from collapsing into one coarse face. Applied to a square domain or inner region, it means that at each of the four corners, the two orthogonal corner edges should never collapse together, so that the global squared shape would be preserved.

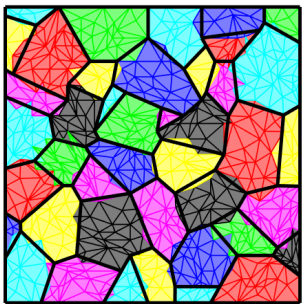
4.3.1 Coarsening in 2D

In 2D, the abstract Step 2 can be detailed as such: for all interfaces composed of multiple fine edges, remove all vertices interior to that interface and connect the boundary vertices to form a new coarse edge. This way, we ensure *node-nestedness*, i.e. that the set of vertices of the coarse mesh is a subset of the fine vertices. Although having a node-nested hierarchy is not mandatory to achieve good multigrid performance, it may help: assuming that the fine nodes adequately capture the geometry, using subsets of those nodes at coarse levels should present some desirable properties with respect to the geometric approximation.

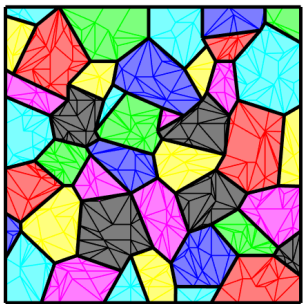
Regarding the computation of the L^2-orthogonal projection developed in the preceding section, numerical experiments show that the rough approximation given by (4.3) is not viable; see Figure 4.4a for a graphical illustration. The polygons must be subdivided to improve precision and, clearly, the method of subdivision influences the ultimate accuracy of the approximation. While a barycentric triangulation (Figures 4.4b and 4.4e) only offers practical usability for $k \leq 1$, an optimal triangulation (i.e. yielding an exact approximation; see Figures 4.4c and 4.4f) can be derived in the context of a coarsening strategy, inasmuch as the intergrid relationships are explicitly formed and can therefore be exploited. In the present coarsening strategy, the agglomeration step does not cause non-nestedness, which can only occur during the edge collapsing phase. Firstly, only fine elements possessing an edge that has been collapsed and that crosses the coarse collapsed edge must be subject to a careful subdivision; the other fine elements are fully embedded in a coarse one, and therefore do not need to be subdivided at all. Then, for the relevant fine elements, it suffices to generate triangles keeping on one side of the coarse edge. Algorithm 4.2 presents the details of the optimal subdivision we have used. Note that it applies to convex polygons; the non-convex ones require a preliminary step of convex partitioning before Algorithm 4.2 can be applied. This algorithm starts by triangulating the polygon independently of the coarse edges. Each triangle is then subtriangulated in order not to cross the coarse edges, following Algorithm 4.3. Note that this algorithm relies on evaluations of intersections between coarse and fine edges, which node-nestedness can certainly ease, insofar as a large part of the crossings between coarse and fine edges will then occur at mesh vertices.



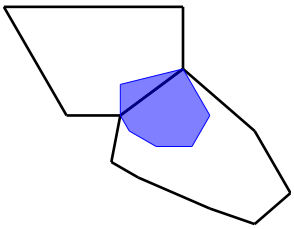
(a) Distribution of the fine elements to their "closest" coarse element.



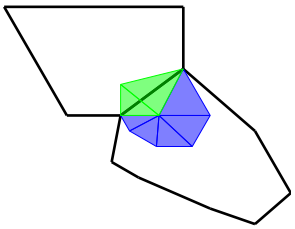
(b) The fine elements are subtriangulated by a barycentric method.



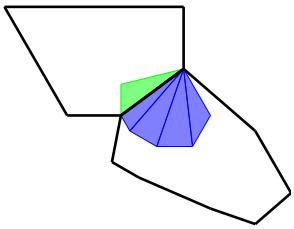
(c) The fine elements are subtriangulated such that they do not cross any coarse element's edge.



(d) Zoom-in on a fine element overlapping two coarse ones.



(e) Barycentric triangulation.



(f) Adapted triangulation preventing subtriangles from overlapping two coarse elements.

Figure 4.4 The top figures show how the fine polygons (in (a)) or their subtriangulations (in (b),(c)) are clustered in the process of approximating the coarse/fine intersection involved in the computation of the L^2-orthogonal projection. The coarse edges are represented by thick black segments. In (b), the fine elements are triangulated by a barycentric method. In (c), the triangulation is adapted to prevent subtriangles to overlap multiple coarse elements. The bottom figures zoom in on a fine element overlapping two coarse ones. In (d), the whole fine element is affected to one of them for the computation of the approximate L^2-orthogonal projection. In (e) and (f), the subtriangles are dispatched on one or the other coarse element according to the location of their barycenters.

4.3.2 Coarsening in 3D

We have not generalized the preceding coarsening strategy to 3D. Instead, we briefly state the issue and propose directions.

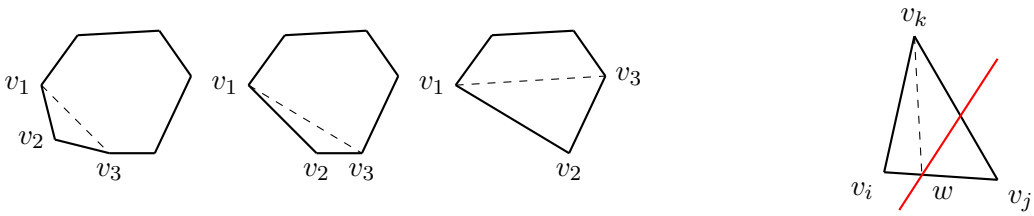
In the 3D setting, it is generally not possible to collapse neighbouring faces into a single one as straightforwardly as in 2D, because the edges framing the fine faces do not usually describe a planar region that could define a new face. However, allowing the addition of new vertices, one can derive a variety of face collapsing methods. A simple one consists in choosing three non-colinear vertices amongst those of the fine faces, thus defining a 2D plane, and then orthogonally projecting the frame's vertices onto that plane to obtain the coarse face. However, in an attempt to preserve, on the coarse mesh, the approximation of the geometry given by the fine one, a more suitable choice for the plane defining the collapsed face would be one minimizing the distance to the vertices lying on the edge frame of the fine faces.

Algorithm 4.2 SubdivideConvexPolygon(V, E)

Input: V: set of vertices sorted in direct order, representing a convex polygon. E: set of coarse edges not to cross.

Output: T: set of triangles partitioning the polygon and not crossing the coarse edges.

1: Let (v_1, v_2, v_3) be the first 3 vertices in V
2: $T :=$ SubdivideTriangle(v_1, v_2, v_3, E)
3: **if** card$(V) > 3$ **then**
4: $\quad T := T \cup$ SubdivideConvexPolygon$(V \setminus \{v_2\}, E)$ // see Figure 4.5a
5: **end if**



(a) First three steps of the triangulation of a convex polygon.

(b) Division of a triangle according to the intersection between the red coarse edge and the triangle edge $[v_i, v_j]$.

Figure 4.5 Respective illustrations for Algorithm 4.2 and Algorithm 4.3.

4.4 Numerical results

4.4.1 Experimental setup

The numerical tests presented in this section have been performed on the diffusion problem (2.1), on 2D and 3D domains, where the source $f \in L^2(\Omega)$ is a discontinuous piecewise constant function. The unit square and cube shall be used as preliminary tests before moving on to more complicated domains. The problems are discretized by the HHO method described in Chapter 2, in which the polynomial degree k of the element- and face-defined polynomials is taken between 0 and 3. We recall that the discrete solution ultimately provided by the method after higher-order reconstruction is a broken polynomial of degree $k+1$. The local polynomial bases in elements and on faces are L^2-orthogonal Legendre bases.

The multigrid method defined in Section 4.1.2 is used as a solver for the solution of the statically condensed system (2.20), and our main goal is to study its asymptotic behaviour so that it can be used for large scale problems. The fine mesh is obtained by Delaunay triangulation of the domain, and coarse meshes are built until the system reaches a maximum size of 1000 unknowns or if the mesh cannot be coarsened anymore. Two different strategies are employed to build the coarse meshes: (i) independent remeshing: letting h be the meshsize of the fine simplicial mesh, the coarse mesh is obtained independently by retriangulation of the domain, enforcing a meshsize $H \approx 2h$; (ii) for 2D problems, the agglomeration-based coarsening strategy with face collapsing, described in Section 4.3. Table 4.1 summarizes, for each strategy, the various methods evaluated in the numerical tests to compute the L^2-orthogonal projection. The stopping criterion is based on the backward error $\|\mathbf{r}\|_2/\|\mathbf{b}\|_2$, where $\mathbf{r}$ denotes the residual of the algebraic system, $\mathbf{b}$ the right-hand side, and $\|\cdot\|_2$ the standard Euclidean norm on the

Algorithm 4.3 SubdivideTriangle(v_1, v_2, v_3, E)

Input: v_1, v_2, v_3: the vertices of the triangle sorted in direct order. E: set of coarse edges not to cross.

Output: T: set of subtriangles partitioning the triangle and not crossing the coarse edges.

if $E = \emptyset$ **then** $T := \{(v_1, v_2, v_3)\}$ // no need to subtriangulate
else
  Let $e \in E$ // take one coarse edge in the list, the others will be handled by recursion
  $noTriangleEdgeCrossesTheCoarseEdge :=$ **true**
  for $i = 1, 2, 3$ **do**
    Given v_i, let v_j, v_k be the other two s.t. (v_i, v_j, v_k) are in direct order.
    $W := e \cap [v_i, v_j]$
    if $W = \emptyset$ **or** $W = \{v_i\}$ **or** $W = \{v_j\}$ **or** $W = [v_i, v_j]$ **then continue for loop**
    else
      $noTriangleEdgeCrossesTheCoarseEdge :=$ **false**
      Define w s.t. $W = \{w\}$.
      // division into two triangles and recursion; see Figure 4.5b
      $T :=$ SubdivideTriangle(v_i, w, v_k, E)
      $T := T \cup$ SubdivideTriangle(w, v_j, v_k, E)
      break for loop
    end if
  end for
  if $noTriangleEdgeCrossesTheCoarseEdge$ **then**
    $T :=$ SubdivideTriangle$(v_1, v_2, v_3, E \setminus \{e\})$
  end if
end if

vector space of coordinates. In all tests, we say that convergence is achieved when the criterion $\|\mathbf{r}\|_2/\|\mathbf{b}\|_2 < 10^{-8}$ is reached.

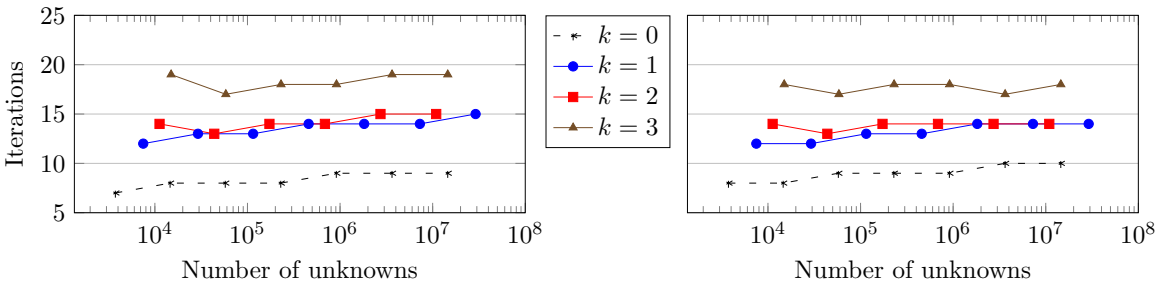
Coarsening strategy	L^2-orthogonal projection
Independent simplicial remeshing	Exact, by computing intersections (cf. (4.2))
	Approximate, w/o subtriangulation (cf. (4.3))
	Approximate, w/ subtriangulation (cf. (4.4)) by middle-edge connection in 2D, Bey's method in 3D
Agglomeration w/ face collapsing	Exact, by computing intersections (cf. (4.2))
	Approximate, w/ barycentric subtriangulation (cf. (4.4) and Figure 4.4e)
	Exact = Approximate w/ optimal subtriangulation (cf. (4.4) and Figure 4.4f)

Table 4.1 Summary of the testing combinations.

4.4.2 Assessment of the approximate L^2-projection

We want to assess how the loss of accuracy implied by the approximate L^2-orthogonal projection (see Section 4.2) affects the convergence of the multigrid solver. In order not to add other difficulties that would interfere with the results, we use the unit square as domain of study. The coarse meshes are built independently from each other by retriangulation of the domain, ensuring good quality at every level. As a reference to the best achievable result, a first test is made with

the L^2-orthogonal projection operator computed exactly, meaning that the intersections of fine and coarse elements are actually computed, over which the required inner products are evaluated (as per (4.2)). Our implementation uses the CGAL library [66] to compute intersections. In this first setting, Figure 4.6a shows, for all polynomial degrees k, the scalable behaviour of the solver, whose convergence rate appears to be independent of the number of unknowns. The number of V(0,3)-cycles required to achieve convergence remains moderate (below 20), although higher than with nested meshes. This first experiment shows the validity of our non-nested approach to multigrid.



(a) Exact L^2-orthogonal projection

(b) Approximate L^2-orthogonal projection with sub-triangulation

Figure 4.6 Number of V(0,3)-cycle iterations to achieve convergence according to the number of unknowns in the system. The geometry is the unit square, and the hierarchy of meshes is obtained by independent remeshing. Each caption states the method of computation of the L^2-orthogonal projection.

It is important to mention that — in our implementation — the computation of the intersections consumes for $k = 1$ over 80% of the CPU time used during the setup phase. The approximate L^2-orthogonal projection we propose allows to reduce the setup cost by a factor of 10 to 40, depending on the granularity of the fine element subdivisions. As an example, Figure 4.7 compares, for a test problem with $k = 1$, the CPU time consumed to compute the L^2-orthogonal projections, according to the method used: exact evaluation, approximate without subdivision (given by (4.3)), approximate with subdivision (given by (4.4)). One can clearly see that the largest share is spent in the exact evaluation of the intersection, which makes approximate methods remarkably more effective.

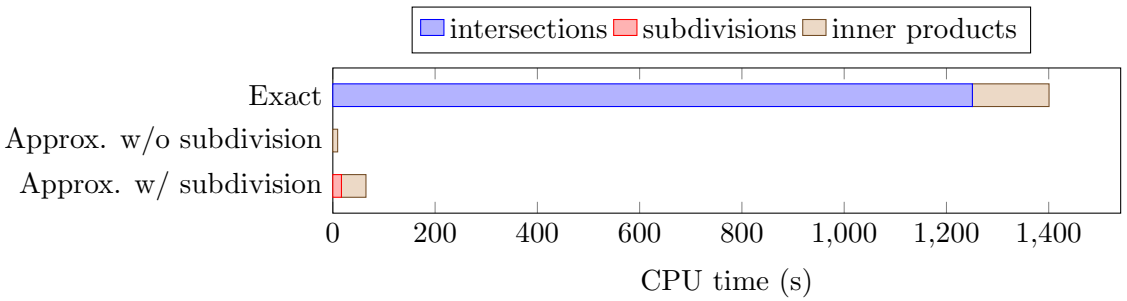


Figure 4.7 Comparison in CPU time of different methods for the computation of the L^2-orthogonal projection during the setup phase of the multigrid. Each bar is divided into sections corresponding to subtasks. The test problem is the unit square meshed by 10^5 triangles, $k = 1$.

In order to assess the usability of our approximate methods, we now compare to the first scalability results the performance of the solver delivered by our approximations. The approximate L^2-orthogonal projection without subdivision (i.e. (4.3)) shows, in 2D, a lack of robustness w.r.t. the polynomial degree. Although the approximation does not seem to degrade the convergence rate for $k \leq 1$, it worsens with $k = 2$, and the solver finally diverges for $k = 3$. Moreover, the same test performed in 3D on the unit cube causes the solver to diverge for all values of k, and so does the use of the 2D polygonal coarsening strategy defined in Section 4.3. These limitations make us discard this simple method.

We next focus on the finer approximation (4.4), where the fine elements are subdivided into subshapes to increase the granularity of the fine-coarse associations. In our tests, we use standard refinement methods to subdivide simplicial elements: connection of the middle-edges in 2D, and Bey's tetrahedral refinement [16] in 3D. Only one step of refinement is performed. Figure 4.6b presents the performance of the multigrid solver on the unit square using this refined L^2-orthogonal projection. We can see that the results given by the exact method are now reproduced. On the unit cube (Figure 4.8), the solver exhibits good performance and scalable behaviour for $k \leq 2$, but diverges for $k = 3$. Note that these 3D results cannot be compared to those of the exact method, as the latter has not been implemented. Referring to the discussion in Section 4.2, we stress that higher orders can be managed with additional steps of refinement in order to improve the accuracy of the approximate L^2-orthogonal projection. Given the CPU times of Figure 4.7, multiple refinements would still be beneficial compared to an exact computation of the intersections.

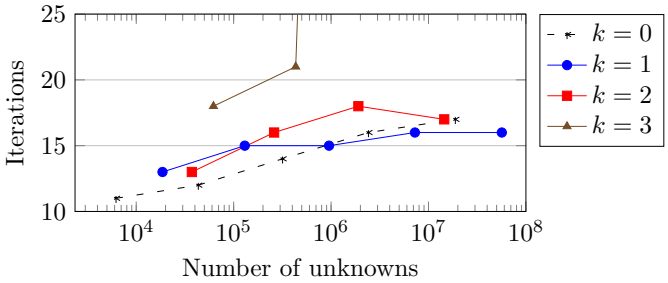


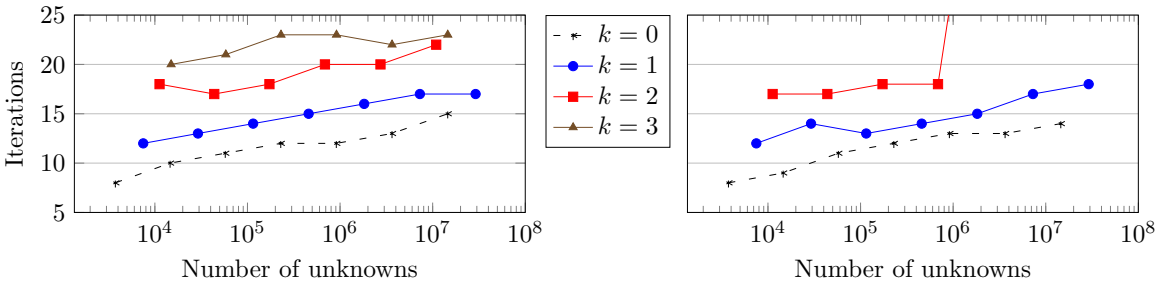
Figure 4.8 Number of V(0,6)-cycle iterations to achieve convergence according to the number of unknowns in the system. The geometry is the unit cube, which is independently retriangulated at each level. The L^2-orthogonal projection is computed approximately with subtriangulation of the fine elements via Bey's method.

4.4.3 Assessment of the agglomeration coarsening strategy with face collapsing

We now evaluate how the multigrid method responds to the coarsening strategy described in Section 4.3, especially when coupled to the approximate L^2-orthogonal projection. Figure 4.9 then presents the same scalability tests as the previous section, this time using the agglomeration coarsening with face collapsing to successively build the coarse meshes from an initial fine triangulation. As a reference, Figure 4.9a shows the results when the L^2-orthogonal projection is computed exactly. We can already remark that the convergence rate is slightly worse and the

trend slightly less flat than with the independent remeshing (cf. Figure 4.6a). Various reasons can explain these differences, the main one probably being the simplicity of implementation of our coarsening strategy, where we do not control and subsequently improve the element shapes and sizes.

Algorithmic scalability tests with the L^2-orthogonal projection approximately computed without subdivision of the fine elements are not presented here: except for $k = 0$, the solver quickly diverges. In Figure 4.9b, we use the barycentric triangulation to subdivide the fine elements (cf. Figure 4.4e) and implement the approximation (4.4). We can see that the good perfomance of $k = 1$ is now recovered, while for $k = 2$, the solver still diverges at moderate problem sizes. Finally, we stress that in the context of the coarsening strategy, the determination of an optimal fine subtriangulation (i.e. whose subelements do not overlap the limits of the coarse elements) is facilitated, which yields an exact implementation of the L^2-orthogonal projection, and therefore leads to results equivalent to Figure 4.9a.



(a) Exact L^2-proj., or approximate w/ optimal subtriangulation

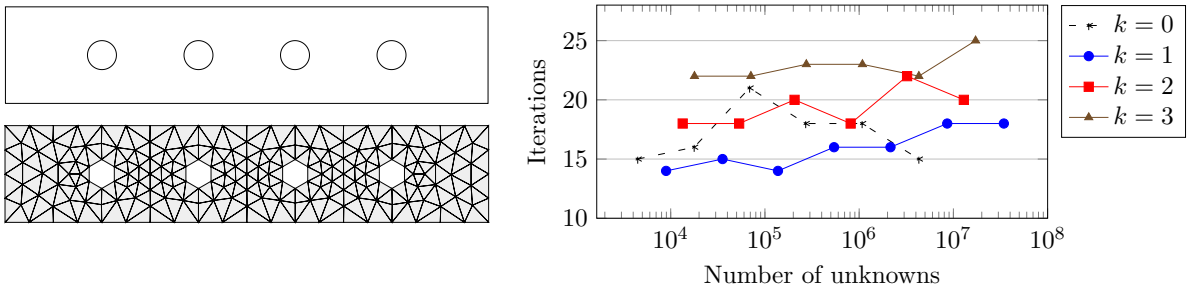
(b) Approximate L^2-proj. w/ barycentric subtriangulation

Figure 4.9 Algorithmic scalability plots: number of V(0,3)-cycle iterations to achieve convergence according to the problem size. The geometry is the unit square. The coarse meshes are built using the agglomeration coarsening strategy with face collapsing. The captions state the method of computation for the L^2-orthogonal projection. The absence of the curve $k = 3$ in (b) means a very large number of iterations or divergence of the solver.

4.4.4 Complex geometry test cases

We now extend the experiments to geometries requiring unstructured meshes in order to validate the method on a wider class of problems. Figures 4.10a and 4.11a respectively draw similar geometries in 2D and 3D containing circular (resp. cylindric) holes, thus requiring unstructured meshes in their discrete settings. Still focusing on the capablility of the multigrid solver to handle large scale problems, Figures 4.10b and 4.11b present the respective algorithmic scalability plots of the solver. Starting from a fine simplicial mesh obtained by Delaunay triangulation, the coarse meshes are built using the coarsening strategy with face collapsing for the 2D tests, and by independent remeshing for the 3D tests. Note that in 2D, the displayed mesh then corresponds to a coarse triangulation that is actually not used (only the fine mesh is triangular); but it shows how the holes are approximated by linear edges. In particular, whichever the method, the holes are approximated, at each level, with respect to the mesh granularity, i.e. by polygons/polyhedra with less and less edges/faces as the mesh grows coarser. The L^2-orthogonal projection is

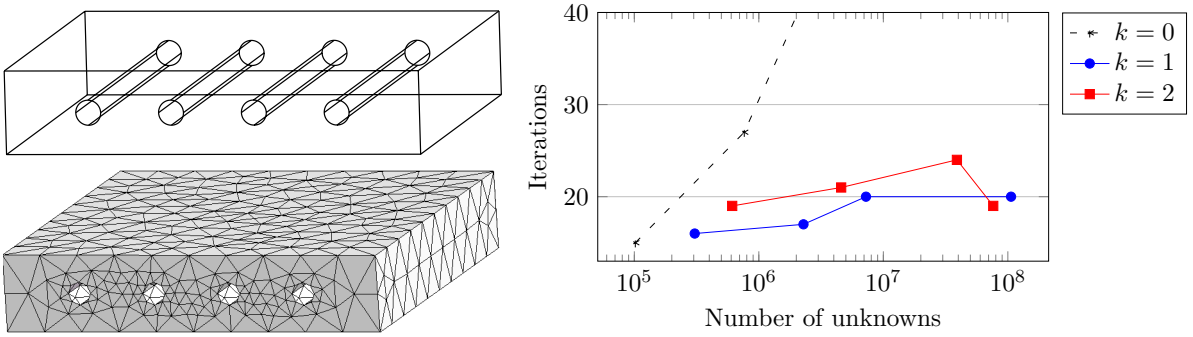
computed exactly in 2D through the construction of an optimal subtriangulation of the elements. In 3D, the L^2-orthogonal projection is computed approximately with tetrahedral subdivision by Bey's method. The 2D results presented in Figure 4.10b show that our multigrid method still exhibits the desired scalable behaviour for all $k \leq 3$. In 3D, the results of Figure 4.11b consistently show the same scalable behaviour for expected polynomial degrees, namely $k = 1$ and 2. The case $k = 0$ is a special case (cf. Section 3.2.2.1) which is not expected to necessarily work on unstructured problems, while the approximation of the L^2-orthogonal projection explains the divergence of the solver for $k = 3$ (cf. Figure 4.8 for the test on the unit cube). Although the displayed datapoints have been obtained with Dirichlet boundary conditions, the results with Neumann conditions on the holes, not reported here for the sake of brevity, indicate the same behaviour.



(a) Bar with four circular holes.

(b) Algorithmic scalability plot with the V(0,3)-cycle.

Figure 4.10 2D complex geometry and associated algorithmic scalability plot for the multigrid solver. The coarse meshes are built using the agglomeration coarsening strategy with face collapsing. The L^2-orthogonal projection is computed exactly.



(a) Plate with four cylindric holes.

(b) Algorithmic scalability plot with the V(0,6)-cycle.

Elements	Hybrid DoFs	Face unknowns (system size)	Multigrid levels	Iterations	Asymptotic convergence rate
17,848,483	177,511,492	106,117,560	5	20	0.45

(c) Details on the last datapoint of $k = 1$.

Figure 4.11 3D test case using the geometry (a). The algorithmic scalability plot (b) of the solver is obtained with the coarse meshes independently retriangulated at each level, and the L^2-orthogonal projection computed approximately using Bey's subdivision.

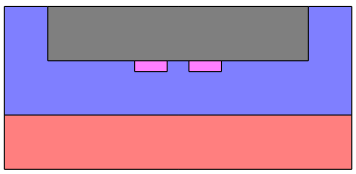
4.4.5 Heterogeneous test case

The algorithm is finally tested on a heterogeneous domain, plotted in Figure 4.12a. This test case, provided by EDF, is a proxy application for an industrial setting, which is characterized by jumps in the diffusion coefficients of several orders of magnitude in an otherwise relatively straightforward geometry. The proxy differs from the industrial setting in its geometry and its diffusion coefficients. However, the most salient feature, the ratio between the largest and the smallest diffusion coefficient, is close to the relevant regime. The domain is composed of four homogeneous and isotropic subdomains. Having assigned a color to each of those regions, their respective diffusion tensors $\kappa_{\text{color}} I$, where I denotes the identity matrix of dimension 2, define a global discontinuous tensor. The values of the coefficients κ_{color} are given in the caption of Figure 4.12a and lead to a maximum jump of 10^8 located at the interface between the gray and blue regions. All meshes in the multigrid hierarchy align with the jumps, i.e. jumps only occur at interfaces and not inside elements. When the coarse meshes come from independent remeshing of the domain, the scalability plot of Figure 4.12b demonstrates a convergence rate that is near-independent to the mesh size for all degrees except the lowest order. With the coarsening strategy, on the other hand, (cf. Figure 4.12c), algorithmic scalability is achieved for all degrees. Moreover, consistently with the results of Section 3.2.3 on nested meshes, the convergence rate of the multigrid method is found to be independent of the size of the discontinuities in the coefficient.

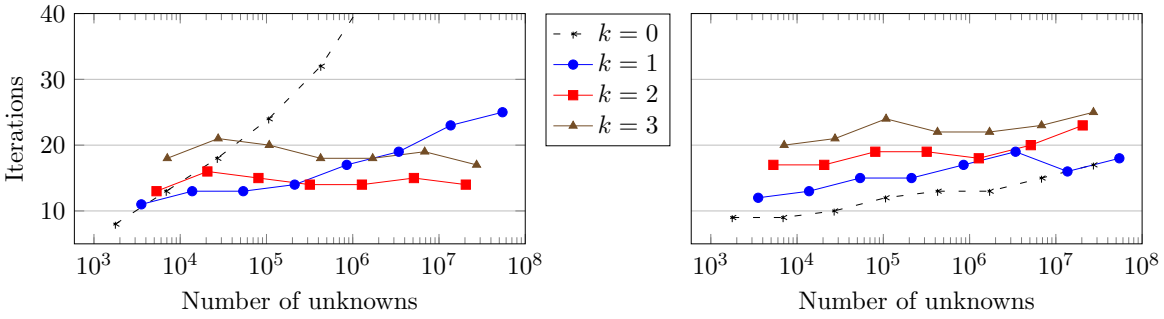
Remark 5. *(Corner singularities) Heterogeneity in such a domain creates corner singularities which make the discretization lose its optimal convergence rate in the general case. Consequently, although the linear solver converges fast, it converges towards a solution that is not necessarily accurate. It is therefore not imperative to impose the linear solver to reach such a low algebraic error. In order to recover the optimal convergence rate of the discretization, techniques of local mesh refinement or energy correction [56] must be used.*

4.4.6 Computational insight

Having focused our numerical tests on the asymptotic convergence of the solver, we now also comment on the computational cost of the iterations. We emphasize that the operation of prolongation remains local, therefore, the corresponding matrix is sparse. Indeed, given a coarse face interfacing two coarse neighbours, its prolongation stencil is limited to the faces linked to the fine elements that overlap those two coarse neighbours. During the setup phase, the prolongation operators are computed at every level and stored in memory. For the smoothers (namely, block Gauss-Seidel methods), the factorization of the diagonal blocks is also computed once and stored in memory to be reused at every iteration. With that setup, and using the cycles described in the numerical tests, intergrid transfers compose about 30% of the total computational work (in flops) of the multigrid iteration, regardless of the space dimension and the polynomial degree. In our implementation, that corresponds to less than 10% of the CPU time. Assuming negligible cost for coarse solving, the rest of the work is distributed among smoothing and residual computation.



(a) Heterogeneous domain with the coefficients $\kappa_{\text{red}} = 30$, $\kappa_{\text{blue}} = 1$, $\kappa_{\text{gray}} = 10^8$, $\kappa_{\text{pink}} = 100$.



(b) Algorithmic scalability plot using independent remeshing.

(c) Algorithmic scalability plot using the coarsening strategy with face collapsing.

Elements	Hybrid DoFs	Face unknowns (system size)	Multigrid levels	Iterations	Asymptotic convergence rate
18,170,130	109,008,748	54,498,358	9	18	0.40

(d) Details on the last datapoint of (c) $k = 1$.

Figure 4.12 The heterogeneous domain described by (a) is composed of homogeneous, isotropic regions. Their respective diffusion coefficients are given in the caption. In (b), the coarse meshes are built by retriangulation and the L^2-orthogonal projection is computed approximately with subtriangulation of the fine elements. In (c), the coarsening strategy is used.

4.5 Conclusion

In this work, we have successfully extended to non-nested mesh hierarchies an efficient nested multigrid solver. Indeed, without requiring stronger smoothing, our adaptation allows to preserve the optimality of the convergence rate on a wider class of problems. The extra cost implied by the numerical evaluation of the L^2-orthogonal projection is also kept to a minimum thanks to an efficient approximation enabling us to discard the expensive computation of intersections between elements. The agglomeration coarsening strategy with face collapsing that we have developed, although naively implemented here, gives promising results, thus offering leads to more advanced methods fulfilling two purposes: coarsening the faces, and preparing the subtriangulation of the fine elements for the purpose of the construction of an accurate approximation of the L^2-orthogonal projection operator.

Chapter 5

An algebraic multigrid method for condensed systems arising from hybrid discretizations

It is my experience that proofs involving matrices can be shortened by 50% if one throws the matrices out.

Emil Artin (1898-1962)

The content of this chapter is submitted to an international, peer-reviewed journal paper [49].

Hybrid discretizations have been part of the landscape of numerical methods to solve Partial Differential Equations (PDEs) since the seventies. In his 1978 book [35, p. 421], P. G. Ciarlet states the following definition: "*we may define (...) as a* hybrid method *any finite element method based on a formulation where one unknown is a function, or some of its derivatives, on the set* Ω*, and the other unknown is the trace of some of its derivatives of the same function, or the trace of the function itself, along the boundaries of the set* K" (Ω representing the domain of study and K a mesh element). Although hybridization of finite element methods first appeared as an implementation trick [124], it was later proven [8] that the new unknowns at faces, introduced as Lagrange multipliers, held additional information on the exact solution, which could be exploited to improve the accuracy of the numerical approximation. A large number of finite element schemes have given rise to hybrid counterparts, starting with the mixed formulations of Raviart-Thomas (RT) [102] and Brezzi-Douglas-Marini (BDM) [30]. More recently, in the context of Discontinuous Galerkin (DG) methods, hybridization was also used to overcome its main drawback, namely, the large number of unknowns resulting from the lack of continuity at element interfaces. Indeed, hybridization allows for the local elimination of cell-based unknowns from the global system, leaving the face unknowns as the only remaining ones in the resulting Schur complement, also called *statically condensed* or *trace* system. Examples of methods whose DoFs verify this structural property include, in particular, Hybridizable Discontinuous Galerkin (HDG), Compatible Discrete Operators (CDO) [19], Hybrid High-Order (HHO) methods [44, 45],

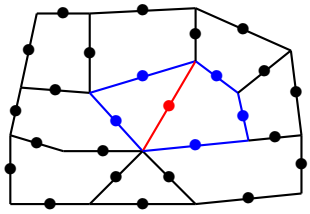
Mimetic Finite Differences (MFD) [42], Mixed and Hybrid Finite Volumes (MHFV) [54, 55, 57]. For a more extensive introduction to hybrid methods and hybridization, we refer to the preface of [43] and the first pages of [39].

Algebraic multigrid (AMG) solvers [59, 112] are very popular for the solution of large linear systems arising from the discretization of elliptic equations on unstructured meshes. Unlike geometric multigrid methods, which require a hierarchy of meshes of different granularity, algebraic algorithms classically do not need more information than the linear system to solve. Discarding all geometric information as input parameter results in the most appreciated feature of these methods, that is, their usability in a black-box fashion.

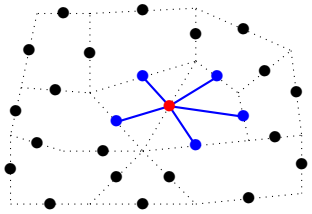
The availability of an easy-to-use, scalable linear solver is essential to help popularize novel discretization methods with the industrial actors, to whom it is crucial to efficiently solve problems of large size. Adopting a new discretization in an industrial context requires heavy preliminary testing, that can be facilitated if the software for the appropriate solver is already available on the market or if its development can easily be externalized. Being isolated from the mesh, which can be generated, stored, and transferred in numerous ways, AMG solvers ally interoperability and performance. Although novel hybrid methods like HHO have gained growing interest in recent years, thus pushing the development of ad-hoc geometric multigrid algorithms [37, 50, 80, 123] or other iterative techniques [88, 119], we are not aware of any AMG specifically targeting condensed systems arising from such discretizations at this time.

Usual AMG solvers designed for low-order finite element or finite difference methods infer mesh information under the assumption that each row in the matrix corresponds to a DoF located at a *mesh node* or *element*. Thus, the connectivity graph of the mesh can be reconstructed algebraically, and coarsening strategies mimicking geometric algorithms can then be performed in order to build the coarse levels. Algebraic algorithms are commonly separated in two families according to how their coarsening strategies can be geometrically interpreted. In the first one, one defines the coarse unknowns as a subset of the fine ones. Geometrically, in an isotropic setting, it consists in selecting fine nodes to keep on the coarse mesh, in such a way that the domain is still uniformly covered while the number of nodes is significantly reduced. This approach have given rise to the so-called Classical AMG (also referred to as C/F AMG) [104, 105], of which BoomerAMG [71] can be mentioned as a popular implementation. The other family regroups aggregation-based methods [26, 33, 90, 94]. In such methods, unknowns are now respectively assimilated to node-defined DoFs (or DoFs within distinct elements), which can then be agglomerated to define a coarse mesh. Among the well-known representatives of aggregation-based AMG software packages, one can cite AGMG [95]. We refer to [116] for a numerical comparison of both approaches applied on a specific application of the Navier–Stokes equations. In the present work, we especially focus on aggregation-based methods. In our hybrid setting at the lowest order, the unknowns of the system are actually linked to faces, i.e. neither nodes nor elements. Consequently, at first glance it might seem peculiar, from a geometrical point of view, to apply the above approaches in this context. Indeed, looking at the example stencil illustrated by Figure 5.1a, aggregation-based coarsening might (and sometimes actually does) aggregate the red DoF with the blue one located the further on its right. As their respective edges do not touch, it is difficult to perceive a geometrical sense in this aggregation. Nonetheless, numerical tests with AGMG show that the approach still works well, which can be geometrically

justified by forgetting about the DoFs being actually face-defined and considering them as mere nodal values. See Figure 5.1b for an illustration of the algebraic stencil as perceived by standard AMG methods. That being said, one can legitimately wonder if a coarsening strategy making geometrical sense in light of the actual meaning of the DoFs as face-defined values could not yield even better results.



(a) Geometric, face-aware view of the stencil



(b) Algebraic view of the stencil (unaware of the faces), as perceived by a standard AMG method

Figure 5.1 Stencil given by the statically condensed matrix at the lowest order. Solid points represent DoFs, which, in this context, are located at faces (edges here). The stencil of the red DoF corresponds to the set of blue ones.

The idea at the origin of the present work is the algebraic reconstruction of the mesh information based, not on the condensed matrix, but on the uncondensed one, which contains the connectivity graph between elements and faces. Note that it implies that this method requires more information than the sole system to solve. Parts of the uncondensed matrix must indeed be brought to the algorithm as additional information, which makes the method less "black-box", but still purely algebraic. Among similar approaches, one can cite AMGe [29]. Once the so-called algebraic mesh is retrieved, especially the neighbouring information between elements, an *element*-based aggregation method can be set up in order to mimic the behaviour of a geometric coarsening or semi-coarsening strategy. Although the construction of the coarse levels is mainly based on *plain* aggregation principles, the prolongation operator also uses techniques borrowed from *smoothed aggregation* methods [98, 120, 121].

AMG methods directly used as solvers may lack efficiency [122, p. 663][89]. Using them as preconditioners for a Krylov method is generally favored. Moreover, plain aggregation methods also suffer from slower convergence than Classical AMG in V-cycle. To handle these issues, we adopt the choices made by AGMG [94]. Namely, we use the so-called K-cycle, which introduces Krylov acceleration into the multigrid recursive cycle. Secondly, one such cycle is used to precondition an outer Krylov method. As the K-cycle does not yield a constant preconditioner, the outer iteration is required to be *flexible*. More generally, the technical choices made in this work are borrowed from AGMG (pairwise aggregation, strong negative coupling criterion, K-cycle...) in order to establish a proper comparison with a standard AMG solver that relies only on the condensed system.

The rest of this work is organized as follows. Section 5.1 lists the features we assume for the underlying discretization to fit our method. Section 5.2 describes the construction of our algebraic multigrid algorithm. In Section 5.3, we apply our method to the lowest order HHO discretization of homogeneous and heterogeneous diffusion problems in 2D and 3D. The outer solver is a Flexible Conjugate Gradient, preconditioned with our algebraic multigrid in

conjunction with the K-cycle: compared to a standard aggregation-based AMG, we report equivalent performances in CPU time, an enhanced robustness to anisotropy on Cartesian meshes, and a similar quasi-optimal asymptotic behaviour. Finally, we discuss limitations and future work in the concluding Section 5.4.

5.1 Assumptions

We consider a scalar elliptic PDE over a domain discretized by a polytopal mesh. For simplicity, we suppose Dirichlet boundary conditions. We assume that the PDE is discretized by a lowest-order hybrid discretization method DoFs corresponding to one scalar value per cell and per face. Throughout this work, the subscript T (resp. F) will consistently refer to the cell-based (resp. face-based) quantities. We also assume that the global *uncondensed* linear system arising from the hybrid discretization at hand is symmetric positive definite, of the form

$$\begin{pmatrix} A_{TT} & A_{TF} \\ A_{TF}^\top & A_{FF} \end{pmatrix} \begin{pmatrix} x_T \\ x_F \end{pmatrix} = \begin{pmatrix} b_T \\ b_F \end{pmatrix}, \tag{5.1}$$

from which Dirichlet boundary unknowns have been eliminated, and where A_{TT} represents the coupling among cell-DoFs, A_{TF} between cell- and face-DoFs, and A_{FF} among face-DoFs. Assuming the discretization is such that the cell unknowns are only locally coupled, A_{TT} is diagonal, and thus inexpensive to invert. The statically condensed system resulting from the local elimination of the cell unknowns is

$$\widetilde{A} x_F = \widetilde{b}, \qquad \widetilde{A} := A_{FF} - A_{TF}^\top A_{TT}^{-1} A_{TF}, \quad \widetilde{b} := b_F - A_{TF}^\top A_{TT}^{-1} b_T. \tag{5.2}$$

As a Schur complement, $\widetilde{A}$ is also symmetric positive definite.

5.2 Algebraic multigrid

We propose to construct an algebraic multigrid method to solve the condensed system (5.2) by using the coupling information given in the uncondensed matrix (5.1). We base our multigrid algorithm on ingredients classically used in aggregation-based AMG. AGMG [94] will serve as a reference for specific technical choices such as the pairwise aggregation, the strong negative coupling criterion, the Krylov acceleration in the multigrid cycle. We also take inspiration from the good results of the geometric multigrid algorithm [51] for the adaptation of the coarsening strategy to the hybrid setting, as well as for the multigrid prolongation operator.

5.2.1 Construction of the algebraic mesh

It is straightforward to algebraically reconstruct the geometric relationships using the connectivity graph given by A_{TF}. Rows of A_{TF} correspond to elements, while columns correspond to faces. Adopting the notation $[1, n] := \{1, \dots, n\}$ for all $n \in \mathbb{N}_+^*$, we then define the set of element indices $T := [1, n_T]$ (resp. the set of face indices $F := [1, n_F]$) where n_T (resp. n_F) is the number of rows (resp. columns) of A_{TF}. For each $i \in T$, the locations of the non-zero coefficients

in the i-th row of A_{TF} correspond to the associated face indices, which we collect in the set $F_i \subset F$. Reciprocally, for all $k \in F$, we collect in $T_k \subset T$ the element indices that contain in their boundary the face of index k. In A_{TF}, two different rows having a non-zero entry in the same column correspond to neighbouring elements. Their interface is given by the faces algebraically defined by the indices of such columns. Formally, for all $(i,j) \in T^2$, i and j are neighbours if $F_i \cap F_j \neq \emptyset$. Moreover, we define the function $\sigma_i \colon F_i \to T$ such that $\sigma_i(k) =: \sigma_{ik}$ is the neighbour of the element of index i sharing the face of index k. Algorithm 5.1 summarizes the process.

Algorithm 5.1 `BuildMesh`

Input: A_{TF}

Output: Mesh defined as the dataset $M := (T, F, (F_i)_{i\in T}, (T_k)_{k\in F}, (\sigma_{ik})_{(i,k)\in T\times F})$

1: $n_T := \text{rows}(A_{TF})$; $T := [1, n_T]$
2: $n_F := \text{cols}(A_{TF})$; $F := [1, n_F]$
3: **for** $i \in T$ **do** $F_i := \{k \in F \mid (A_{TF})_{ik} \neq 0\}$ **end for**
4: **for** $k \in F$ **do** $T_k := \{i \in T \mid (A_{TF})_{ik} \neq 0\}$ **end for**
5: **for** $i \neq j \in T$ **do**
6: **if** $F_i \cap F_j \neq \emptyset$ **then**
7: $\forall k \in F_i \cap F_j$, set $\sigma_{ik} := j$ and $\sigma_{jk} := i$
8: **end if**
9: **end for**

5.2.2 Mesh coarsening by element aggregation and face collapsing

Now that we have built the algebraic mesh, that is, a list of elements, a list of faces, as well as the links between them and subsequently the neighbouring relationships, we are able to algebraically reproduce a geometric *element-based* aggregation strategy. The framework of the present contribution does not restrict the aggregation method, as long as the required information for choosing the aggregates can be retrieved from the uncondensed system. That is why the way the elements are agglomerated will remain abstract in the general algorithm. As such, Algorithm 5.2, which describes the global process of element aggregation, refers to the abstract function `BuildAggregate` (at step 6). `BuildAggregate` takes an element $i \in T$ as an argument and returns a list of elements (including i) chosen to form an aggregate. The simplest aggregation method, corresponding to clustering i with all its unaggregated neighbours, would be enough to put our algorithm to the test. However, it would only rely on the element connectivity graph, i.e. on the location of the non-zero coefficients in the block A_{TF}, regardless of their values. In order to manage anisotropic problems and give an example of how semi-coarsening can be performed in our hybrid setting, we give in Section 5.2.3 a hybrid counterpart of the node-defined pairwise aggregation based on the *strong negative coupling* criterion, as it is formulated in the early version of AGMG described by [94]. We denote by $(G_{T,i})_{i\in[1,n_{T,c}]}$ the produced aggregates, with $n_{T,c}$ defining the number of aggregates.

In a multigrid method that applies to trace systems, as the smoother operates on the face unknowns, the efficient reduction of the low-frequency components of the error relies on accessing coarse representations of the face-defined functions. This implies that *faces* must be coarsened between levels (see Section 3.2.4.3), which is a new constraint imposed to any suited coarsening strategy. Consequently, we combine the element aggregation with an additional step of *face*

Algorithm 5.2 `ElementAggregation`

Input: Mesh, output of Algorithm 5.1

Output: Aggregation information $\mathcal{G}_T$, defined as the collection of data:
$(G_{T,i})_{i\in[1,n_{T,c}]}$: element aggregates
$(g_i)_{i\in[1,n_T]}$: association of the element i to the aggregate g_i it belongs to
$(\mathring{F}_i)_{i\in[1,n_{T,c}]}$: fine faces interior to the aggregates
$(\widehat{F}_i)_{i\in[1,n_{T,c}]}$: fine faces at the boundary of the aggregates

1: `Todo` $:= T$ // remaining non-aggregated elements
2: $n := 0$ // aggregate index
3: **while** `Todo` $\neq \emptyset$ **do**
4: Select $i \in$ `Todo`
5: $n := n + 1$
6: $G_{T,n} :=$ `BuildAggregate`$(i,$ `Todo`$)$ // see Algorithm 5.5 for a possible algo.
7: **for** $j \in G_{T,n}$ **do** // save for each fine element the aggregate it is in
8: $g_j := n$
9: **end for**
10: $\mathring{F}_n := \{k \in \bigcup_{i\in G_{T,n}} F_i \mid \exists i \neq j \in G_{T,n} \text{ s.t. } k \in F_i \cap F_j\}$ // interior faces
11: $\widehat{F}_n := \left(\bigcup_{i\in G_{T,n}} F_i\right) \setminus \mathring{F}_n$ // boundary faces
12: `Todo` $:=$ `Todo` $\setminus G_{T,n}$
13: **end while**
14: $n_{T,c} := n$

aggregation, also called *face collapsing*. In particular, we reproduce the technique devised in [51], which consists in merging into single faces the interfaces between aggregates.

During the element aggregation process, the fine faces are split into two disjoint subsets $\mathring{F} \cup \widehat{F} = F$ according to their situation w.r.t. the aggregates. $\mathring{F}$ regroups the faces interior to an aggregate, i.e. the faces shared by two elements aggregated together. Geometrically speaking, those faces are "removed" to give rise to the aggregates. The remaining faces, which compose the aggregates' boundaries, are collected in $\widehat{F}$. We also denote their local counterparts, with respect to each aggregate, by $(\mathring{F}_i)_{i\in[1,n_{T,c}]}$ and $(\widehat{F}_i)_{i\in[1,n_{T,c}]}$. See Figure 5.2a for a geometric illustration.

(a) Element aggregation (b) Face collapsing

Figure 5.2 Aggregation process with face collapsing. In (a), elements are aggregated, yielding two aggregates. "Removed" edges, represented in dashed red lines, are collected in $\mathring{F}$, while the remaining ones are collected in $\widehat{F}$. Then, in (b), the interface between the two neighbouring aggregates, here made of two edges (in dashed red lines), is collapsed into a single one (solid blue line). The other edges yield *singleton* face aggregates.

Neighbouring relationships between element aggregates can be directly deduced from $\widehat{F}$. We can then collapse into one single face the interfaces between aggregates without altering the coarse adjacency graph. Note that each interface, whether it is made of multiple faces or only

one, gives rise to one face aggregate, so singleton aggregates are produced. Figure 5.2b gives a geometric interpretation of the face collapsing, and Algorithm 5.3 formalizes the process.

Algorithm 5.3 `FaceCollapsing`

Input: Fine mesh, output of Algorithm 5.1
Aggregation information, output of Algorithm 5.2
Output: Face collapsing information $\mathcal{G}_F$, defined as the collection of data:
$(G_{F,k})_{k\in[1,n_{F,c}]}$: face aggregates
$(H_i)_{i\in[1,n_{T,c}]}$: collapsed faces defining the new boundaries of the aggregates

1: `Todo` $:= \widehat{F}$ // remaining non-collapsed faces
2: $m := 0$ // face aggregate index
3: **while** `Todo` $\neq \emptyset$ **do**
4: Select $k \in$ `Todo`
5: $m := m + 1$
6: Let $G := \bigcup_{i\in T_k} g_i$ // element aggregates the face k is at the interface of
7: $G_{F,m} := \bigcap_{n\in G} \widehat{F}_n$ // fine faces (including k) composing that interface
8: **for** $n \in G$ **do**
9: $H_n := H_n \cup \{m\}$ // in the coarse mesh, m is now a face of the aggregate n
10: **end for**
11: `Todo` $:=$ `Todo` $\setminus G_{F,m}$
12: **end while**
13: $n_{F,c} := m$

Algorithm 5.4 `MeshCoarsening`

Input: M: mesh, output of Algorithm 5.1

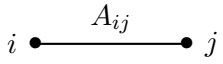
1: $\mathcal{G}_T :=$ `ElementAggregation`(M) // Algorithm 5.2
2: $\mathcal{G}_F :=$ `FaceCollapsing`$(M, \mathcal{G}_T)$ // Algorithm 5.3
3: $T_c := [1, n_{T,c}]$ // coarse elements := aggregates
4: $F_c := [1, n_{F,c}]$ // coarse faces := collapsed faces

Now that aggregates have been made for elements and faces by the global Algorithm 5.4, they can be numbered and become the coarse elements and faces, thus defining a coarse mesh.

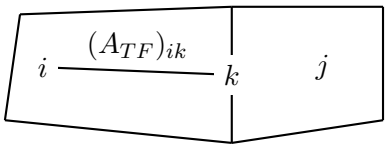
5.2.3 Pairwise aggregation by strong negative coupling

This strategy allows to aggregate pairs of neighbouring elements in the direction of strong anisotropy, and gives an implementation of the abstract method `BuildAggregate` at step 6 of Algorithm 5.2. The choice of the neighbours for the constitution of the aggregates follows an adaptation to hybrid unknowns of the usual rule of *negative coupling* employed in standard AMG. For each element, this rule allows to evaluate, for all of its neighbours, a numerical criterion indicating their strength of connection. Only those which have a strong enough connection and are not already aggregated are considered for aggregation. Among them, the strongest one is chosen, and leads to a pair aggregate. However, if none of the strong neighbours are available, i.e. they have all already been previously aggregated, then the element stays alone in a so-called *singleton* aggregate.

Before introducing our hybrid criterion for the strong negative relationship, let us recall the *node-defined* criterion used by standard AMG methods. As multiple variations of this criterion



(a) Coupling of nodes i and j in standard AMG.



(b) Coupling of elements i and j via their common face k in the hybrid setting.

Figure 5.3 Coupling values in standard and hybrid settings

exist, we follow the example of AGMG in the version of [94]. Given the stiffness matrix A and an algebraic node i associated to the i-th row of A, the coupling coefficient modelling the connection of j to i is provided by the matrix entry A_{ij} (see Figure 5.3a). We say that j is negatively coupled (or connected) to i if $A_{ij} < 0$, and the strength of connection is defined by the modulus of that coefficient. The strongest connection then corresponds to $\bar{c}_i := \max_{j|A_{ij}<0} |A_{ij}|$. Given a weak/strong connection threshold $0 < \beta \leq 1$ (typically set to 0.25), the set of nodes strongly connected to i is $\{j \mid A_{ij} < 0 \text{ and } |A_{ij}| \geq \beta\, \bar{c}_i\}$.

In our case, in hybrid form, elements are coupled through A_{TF}. Specifically, given an element of index $i \in T$ and its neighbour of index j, the coupling coefficient is provided through their common face of index k by the matrix coefficient $(A_{TF})_{ik}$; see Figure 5.3b. We then introduce the following definition for the negative coupling criterion: j is negatively coupled to i via k if $(A_{TF})_{ik} < 0$. Now, for the purpose of managing heterogeneous problems by preventing aggregation across large jumps in the diffusion coefficient, we remark that this sole value is not enough to detect a discontinuity between i and j. Indeed, $(A_{TF})_{ik}$ only bears information local to i. We also notice that, in the heterogeneous isotropic diffusion case, the coefficient $(A_{TF})_{ik}$ is scaled by the actual diffusion coefficient of the element of index i. We then introduce the heterogeneity ratio between the elements of indices i and j connected by the face of index k as

$$\rho_{ij} := \max\left(\frac{(A_{TF})_{ik}}{(A_{TF})_{jk}}, \frac{(A_{TF})_{jk}}{(A_{TF})_{ik}}\right) > 1. \tag{5.3}$$

Meaning to penalize aggregation across jumps, instead of simply defining the coupling strength by $|(A_{TF})_{ik}|$, we define it as

$$c_{ik} := |(A_{TF})_{ik}|/\rho_{i\sigma_{ik}},$$

where we recall that σ_{ik} refers to the element index j that shares the face index k with i. According to this criterion, the remaining definitions are straightforward. The strongest connection to i is given by

$$\bar{c}_i := \max_{k\in F_i, (A_{TF})_{ik}<0} c_{ik},$$

and the set of faces strongly connected to i by

$$\underline{F}_i := \{k \in F_i \mid (A_{TF})_{ik} < 0 \text{ and } c_{ik} \geq \beta\, \bar{c}_i\}. \tag{5.4}$$

Finally, the strong neighbours of i may be retrieved in the set $\{\sigma_{ik}, k \in \underline{F}_i\}$. The corresponding implementation of the abstract function `BuildAggregate` is given by Algorithm 5.5.

Notice that the number of singleton aggregates can significantly vary depending on the

Algorithm 5.5 `BuildAggregate`

Input: $i \in T$: element to aggregate
$\widetilde{T} \subset T$: non-aggregated elements
Output: G: aggregate

1: // Collect faces strongly connected to i through which neighbours are still available
2: $\widetilde{F}_i := \{k \in \underline{F}_i \mid \sigma_{ik} \in \widetilde{T}\}$ // cf. (5.4) for the definition of $\underline{F}_i$
3: **if** $\widetilde{F}_i \neq \emptyset$ **then**
4: $\quad k := \arg\max_{\ell \in \widetilde{F}_i} c_{i\ell}$ // face with the strongest coupling
5: $\quad G := \{i, \sigma_{ik}\}$ // aggregation of i and its neighbour relative to k
6: **else**
7: $\quad G := \{i\}$ // i forms a singleton aggregate
8: **end if**

order following which the elements are aggregated. So, to minimize the number of singleton aggregates, the elements are beforehand parsed and attributed a priority value in order to favor those that have the fewest strong neighbours. Especially, we follow the priority numbering algorithm described in [41] and process elements by order of priority at step 4 of Algorithm 5.2.

5.2.4 Cell- and face-defined auxiliary prolongation operators

Given $T := [1, n_T]$ (resp. $F := [1, n_F]$) the fine elements (resp. faces) indices in the algebraic mesh, we denote by $T_c := [1, n_{T,c}]$ (resp. $F_c := [1, n_{F,c}]$) the coarse elements (resp. faces) constructed by the aggregation process of Section 5.2.2 (Algorithm 5.4). We start by defining an auxiliary cell-defined prolongation matrix Q_T (of size $n_T \times n_{T,c}$) in the manner of plain aggregation:

$$\forall i \in T,\ \forall j \in T_c, \quad (Q_T)_{ij} := \begin{cases} 1 & \text{if } i \in G_{T,j} \\ 0 & \text{otherwise.} \end{cases} \tag{5.5a}$$

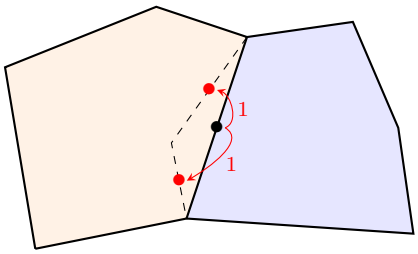
This highly sparse prolongation operator (exactly 1 non-zero per row) transfers the unknown values respectively assigned to the coarse elements onto the fine elements they aggregate. Without smoothed aggregation techniques, all fine elements of the same aggregate receive the same value. Regarding the faces, we define the auxiliary prolongation matrix Q_F (of size $n_F \times n_{F,c}$) such that for $k \in F$,

(i) if $k \in \widehat{F}$, i.e. k belongs to a face aggregate,

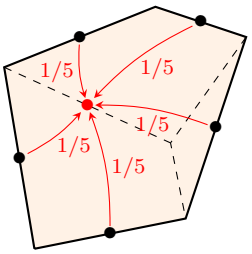
$$\forall \ell \in F_c, \quad (Q_F)_{k\ell} := \begin{cases} 1 & \text{if } k \in G_{F,\ell} \\ 0 & \text{otherwise;} \end{cases} \tag{5.5b}$$

(ii) if $k \in \mathring{F}$, let m be the coarse element embedding k (i.e. $k \in \mathring{F}_m$)) and H_m its set of (potentially) collapsed faces; then,

$$\forall \ell \in F_c, \quad (Q_F)_{k\ell} := \begin{cases} 1/\operatorname{card}(H_m) & \text{if } \ell \in H_m \\ 0 & \text{otherwise.} \end{cases} \tag{5.5c}$$



(a) Aggregated faces ($\in \widehat{F}$)



(b) Interior/removed face ($\in \mathring{F}$)

Figure 5.4 Operator Q_F. The fine red DoFs are set by the coarse black ones.

To summarize, an aggregated face takes the value of the corresponding coarse aggregate (just like the elements), and a "removed" face, embedded in a coarse element, takes the average value of that coarse element's faces; see Figure 5.4.

5.2.5 Multilevel hierarchy

As the method described in Section 5.2.2 does not necessarily yield an aggressive enough coarsening [94], and also in order to build more levels for the multigrid hierarchy, we want to repeat the coarsening process, thus defining the so-called *multiple* coarsening. To do so, one has to define a coarse version of the uncondensed matrix (5.1) to allow recursive execution.

Given the initial blocks A_{TT}, A_{TF} and A_{FF} of the fine uncondensed matrix, we use the auxiliary prolongation operators introduced in Section 5.2.4 to define coarse counterparts in a Galerkin fashion:

$$\begin{pmatrix} A_{TT,c} & A_{TF,c} \\ A_{TF,c}^\top & A_{FF,c} \end{pmatrix} := \begin{pmatrix} Q_T & \\ & Q_F \end{pmatrix}^\top \begin{pmatrix} A_{TT} & A_{TF} \\ A_{TF}^\top & A_{FF} \end{pmatrix} \begin{pmatrix} Q_T & \\ & Q_F \end{pmatrix}. \tag{5.6}$$

Note that in practice, only the blocks used in the algorithm must be assembled. In this work, we only need A_{TF} (for the coarsening strategy) and A_{TT} (used in the multigrid prolongation operator P_F further described in Section 5.2.6).

Algorithm 5.6 describes one step of coarsening. In addition to building the coarse blocks (step 4), the coarsening process also constructs the operator P_F that will be used as prolongation operator in the multigrid algorithm (step 5). Indeed, although Q_F could be employed for that purpose, we choose to explore another, more efficient approach (the construction of the operator P_F is described in Section 5.2.6 below). Furthermore, the coarse operator for the lower level of the multigrid algorithm is defined as the Galerkin operator, constructed from P_F and the condensed matrix $\widetilde{A}$, initialized at the finest level by the Schur complement (5.2) (step 6). Finally, to be more consistent with this coarse operator, we recompute the coarse blocks following formula (5.6) in which Q_F is replaced with P_F, i.e.

$$\begin{pmatrix} A_{TT,c} & A_{TF,c} \\ A_{TF,c}^\top & A_{FF,c} \end{pmatrix} := \begin{pmatrix} Q_T & \\ & P_F \end{pmatrix}^\top \begin{pmatrix} A_{TT} & A_{TF} \\ A_{TF}^\top & A_{FF} \end{pmatrix} \begin{pmatrix} Q_T & \\ & P_F \end{pmatrix}. \tag{5.7}$$

Again, only the blocks actually needed, here $A_{TF,c}$ only, are recomputed; see step 7.

While Algorithm 5.6 performs one step of coarsening, Algorithm 5.7 handles the recursion

until a targeted coarsening factor is reached. The end of the multiple coarsening process defines one new multigrid level, and the two-level prolongation operator is defined by successively chaining the prolongation operators coming out of each coarsening (step 7).

Algorithm 5.6 `Coarsening`

Input: A_{TT}, A_{TF}, $\widetilde{A}$
Output: $A_{TT,c}$, $A_{TF,c}$, $\widetilde{A}_c$, P_F

1: $M :=$ `BuildMesh`(A_{TF}) // Algorithm 5.1
2: $[G_T, G_F] :=$ `MeshCoarsening`(M) // Algorithm 5.4
3: Compute Q_T and Q_F by (5.5)
4: $A_{TT,c} := Q_T^\top A_{TT} Q_T$; $A_{TF,c} := Q_T^\top A_{TF} Q_F$ // cf. (5.6)
5: Compute P_F by (5.10)
6: $\widetilde{A}_c := P_F^\top \widetilde{A} P_F$
7: $A_{TF,c} := Q_T^\top A_{TF} P_F$ // cf. (5.7)

Algorithm 5.7 `MultipleCoarsening`

Input: A_{TT}, A_{TF}, $\widetilde{A}$, `targetCF`
Output: $A_{TT,c}$, $A_{TF,c}$, $\widetilde{A}_c$, P_F

1: $A_{TT,\text{aux}} := A_{TT}$; $A_{TF,\text{aux}} := A_{TF}$
2: $\widetilde{A}_{\text{aux}} := \widetilde{A}$
3: $P_F := I$
4: `cf` $:= 0$ // coarsening factor
5: **while** `cf` $<$ `targetCF` **do**
6: $[A_{TT,c}, A_{TF,c}, \widetilde{A}_c, P_{F,\text{aux}}] :=$ `Coarsening`$(A_{TT,\text{aux}}, A_{TF,\text{aux}}, \widetilde{A}_{\text{aux}})$ // Algorithm 5.6
7: $P_F := P_F P_{F,\text{aux}}$
8: $A_{TT,\text{aux}} := A_{TT,c}$; $A_{TF,\text{aux}} := A_{TF,c}$
9: $\widetilde{A}_{\text{aux}} := \widetilde{A}_c$
10: `cf` $:= \text{cols}(A_{TF}) / \text{cols}(A_{TF,c})$
11: **end while**

5.2.6 Multigrid prolongation operator

Although Q_F could also be used as prolongation operator for the multigrid algorithm, we choose to explore another approach, which happens to give better results. Thus, we would like to emphasize that Q_F is only employed to build the coarse blocks during the setup phase, while P_F, described in this section, defines the prolongation operator used in the multigrid iterations. It is meant to be an algebraic counterpart of the geometric prolongation operator defined in Chapter 3, which relies on the decondensation of the cell unknowns.

First, we introduce a preliminary prolongation operator denoted by $P_F^{(0)}$. For all $k \in F$, its k-th row $(P_F^{(0)})_k$ is defined as

$$(P_F^{(0)})_k := \begin{cases} (Q_F)_k & \text{if } k \in \widehat{F} \\ (\Pi_c^f \Theta_c)_k & \text{if } k \in \mathring{F}. \end{cases} \tag{5.8}$$

In this definition, $\Theta_c \in \mathbb{R}^{n_{T,c} \times n_{F,c}}$ locally computes the value on the coarse cells from their respective coarse faces, while $\Pi_c^f \in \mathbb{R}^{n_F \times n_{T,c}}$ transfers the value associated the coarse cells to

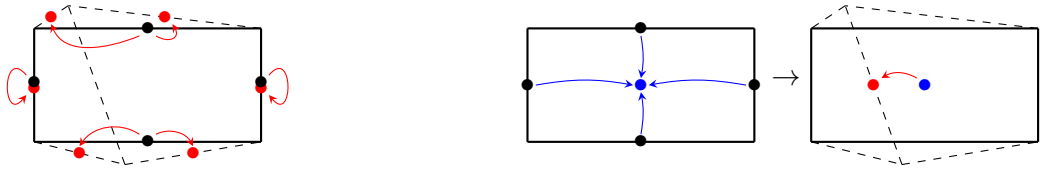
their respective interior fine faces. We define

$$\Theta_c := -A_{TT,c}^{-1} A_{TF,c}, \tag{5.9}$$

which reverses the static condensation by solving for the cell unknowns local problems on the coarse cells given values on the faces. Next, for any face $k \in F$,

$$\forall n \in T_c, \qquad (\Pi_c^f)_{kn} := \begin{cases} 1 & \text{if } k \in \mathring{F}_n \\ 0 & \text{otherwise.} \end{cases}$$

Figure 5.5 illustrates $P_F^{(0)}$.



(a) Q_F for $\widehat{F}$ (b) $\Pi_c^f \Theta_c$ for $\mathring{F}$

Figure 5.5 Preliminary prolongation operator $P_F^{(0)}$. In these figures, we consider two fine elements (dashed lines) aggregated into one (solid lines). The DoFs on the coarse faces are represented by black dots, on the fine faces by red dots, on the coarse element by a blue dot.

Second, we remark that the stencil, in $P_F^{(0)}$, of the DoFs associated to removed fine faces (i.e. $k \in \mathring{F}$) is local to coarse elements. Given that the stencil in $\widetilde{A}$ is also local to coarse elements for those unknowns, one sweep of Jacobi smoothing can be applied to them without enlarging the prolongation stencil. This allows to boost the convergence with virtually no additional computational cost. Note that a second smoothing iteration would enlarge the stencil outside of coarse elements, which we do not want. Setting the damping factor to $\omega := 2/3$, the smoothing matrix is defined by $J := I - \omega \widetilde{D}^{-1} \widetilde{A}$, where I is the identity matrix and $\widetilde{D}$ the diagonal part of $\widetilde{A}$. The final multigrid prolongation operator P is then defined row-wise for all row $k \in F$ as

$$(P_F)_k := \begin{cases} (Q_F)_k & \text{if } k \in \widehat{F} \\ (J P_F^{(0)})_k & \text{if } k \in \mathring{F}. \end{cases} \tag{5.10}$$

To conclude about the formulation of P_F, we want to point out its mixed construction with respect to aggregation-based methods: plain aggregation is used for $\widehat{F}$, while $\mathring{F}$ benefits from smoothed prolongation.

5.2.7 Multigrid method

A hierarchy of L levels is built by multiple coarsening following Section 5.2.5, and numbered from 1 (the coarsest) to L (the finest). At each level ℓ, the prolongation operator is given by (5.10), which we simply denote by P_ℓ instead of $P_{F,\ell}$. The other multigrid ingredients are chosen as per the variational framework: namely, the restriction is set to $P_\ell^\top$, and the coarse operator $\widetilde{A}_{\ell-1}$ to the Galerkin construction, initialized by the condensed matrix (5.2) as the finest operator

$\widetilde{A}_L$ (i.e. $\widetilde{A}_{\ell-1} := P_\ell^\top \widetilde{A}_\ell P_\ell, \quad \forall \ell = 2, \ldots, L$). The other parameters of the method (smoothers, cycle, coarsening factor, weak/strong coupling threshold, coarse grid solver) are left to the user's discretion; our choices are detailed in Section 5.3.1.

5.2.8 Usage as a preconditioner

Prolongation operators arising from plain aggregation are known to yield poor approximation properties of the coarse grid correction. However, it is known [89] that this loss of approximation, leading to bad convergence of the V-cycle, can be compensated by the use of the K-cycle ([94, Algorithm 3.2]), and by preconditioning a Krylov method. The detail of the K-cycle at a generic level ℓ is recalled in Algorithm 5.8, and is applied, in our context, on the hierarchy of condensed matrices $(\widetilde{A}_\ell)_{\ell=1\ldots L}$ and prolongation operators $(P_\ell)_{\ell=2\ldots L}$.

Algorithm 5.8 KCyclePrec$_\ell$

Data: Hierarchies of matrices $(A_k)_{k=1\ldots\ell}$ and prolongation operators $(P_k)_{k=2\ldots\ell}$
Input: Residual r
Output: The approximate solution e of the linear system $A_\ell\, e = r$

1: **if** $\ell = 1$ **then**
2: Direct solving: $e := A_\ell^{-1} r$
3: **else**
4: Relaxation using smoother M_ℓ: $e := M_\ell^{-1}\, r$
5: Residual computation: $r := r - A_\ell\, e$
6: Restriction of the residual: $r_{\ell-1} := P_\ell^\top\, r$
7: The residual equation is solved at level $\ell - 1$ by 1 or 2 iterations of a Krylov method preconditioned by KCyclePrec$_{\ell-1}$ with 0 initial guess:
$e_{\ell-1} :=$ **InnerKrylov**$(A_{\ell-1}, r_{\ell-1}, \text{KCyclePrec}_{\ell-1}, 0)$
8: Coarse grid correction: $e := e + P_\ell\, e_{\ell-1}$
9: Residual computation: $r := r - A_\ell\, e$
10: Relaxation using smoother M_ℓ: $e := M_\ell^{-1}\, r$
11: **end if**

In this cycle, only the way the residual equation is solved on the coarse level (step 7) differs from the standard V- and W-cycles. Instead of performing 1 (in V-cycle) or 2 (in W-cycle) iterations of the same cycle at the lower level through a direct recursion, the K-cycle performs 1 or 2 iterations of an inner Krylov method, itself preconditioned by the current K-cycle algorithm at the lower level. Inspired from [94], the number of iterations executed (1 or 2) is decided dynamically, according to the effective reduction of the residual: if the residual norm is not reduced by a factor of at least 4 after the first iteration, then a second one is performed.

The final solver is then built by using Algorithm 5.8 to precondition the same Krylov method as for the (1 or 2) inner iterations performed within the the K-cycle. Note that the variable number of Krylov iterations in the K-cycle makes the latter a *variable* preconditioner, which implies that a *flexible* version of the Krylov method has to be used. Flexible versions of Krylov methods are usually obtained by the use of a truncature-restart strategy regarding the orthogonalization of the Krylov vectors. Algorithm 5.9 presents such a Flexible Conjugate Gradient [93], the so-called FCG(1) (also referred to as IPCG), obtained by orthogonalization of the research direction against only one previous Krylov vector, without restart.

Algorithm 5.9 Preconditioned Flexible Conjugate Gradient FCG(1)

Input: Matrix A, right-hand side b, preconditioner Prec, initial guess x_0
Output: The approximate solution x of the linear system $Ax = b$

$x := x_0$
$r := b - A\,x$
for $i = 1, 2, \ldots$ **do**
  $w := \mathrm{Prec}(r)$
  if $i = 1$ **then** $d := w$
  else $d := w - \frac{w^\top A\, d_{\mathrm{old}}}{d_{\mathrm{old}}^\top A d_{\mathrm{old}}} d_{\mathrm{old}}$
  end if
  $\alpha := d^\top r / d^\top A\, d$
  $x := x + \alpha d$
  $r := r - \alpha A d$
  $d_{\mathrm{old}} := d$
end for

5.3 Numerical tests

5.3.1 Experimental setup

Letting Ω be a bounded polytopal domain of $\mathbb{R}^d$, $d \in \{2, 3\}$, we consider the diffusion problem

$$\begin{cases} -\nabla \cdot (\mathbf{K} \nabla u) = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $f \in L^2(\Omega)$ is a given source term and $\mathbf{K} \colon \Omega \to \mathbb{R}^{d \times d}$ is the diffusion tensor field, which is assumed to be real, symmetric, uniformly elliptic. This problem is discretized by the HHO method [43] at the lowest order, which matches the structural requirements of Section 5.1. The homogeneous Dirichlet boundary condition is handled by elimination. The multigrid preconditioner given by Algorithm 5.8 performs one sweep of Gauss–Seidel in lexicographic order as pre-smoothing and one sweep of Gauss–Seidel in anti-lexicographic order as post-smoothing. We refer to this cycle as the K(1,1)-cycle. As the arising system is symmetric positive definite, we choose the Flexible Conjugate Gradient FCG(1) (cf. Algorithm 5.9) for the outer iteration as well as for the inner iteration of the K-cycle, meaning that the FCG, as outer solver, is preconditioned by the K(1,1)-cycle of our multigrid method. The preconditioner being symmetric positive definite, convergence of the outer FCG is ensured. To build each coarse level, multiple pairwise aggregations with the weak/strong coupling threshold $\beta = 0.25$ are performed (Section 5.2.3), enforcing a coarsening factor ≥ 3.8. Coarse levels are built until the operator matrix has less than 1000 rows, where the system is solved by a direct solver. Iterations stop when the backward error, defined by the residual normalized by the right-hand-side, reaches a value lower than 10^{-8}. In the following results, note that the number of iterations refers to the outer solver, i.e. FCG.

5.3.2 Methodology

The main goal of the following numerical experiments is to establish a comparison between the solver developed in this work and the equivalent one made in the way of standard AMG. The

former uses the uncondensed matrix to devise an *element-based* coarsening strategy, while the latter is directly working on the condensed system by implementing an *node-defined* coarsening strategy. Consequently, additionally to our novel algorithm, which we will refer to as Uncondensed AMG (U-AMG), we introduce the so-called Condensed AMG (C-AMG), which uses the nodewise-equivalent coarsening strategy directly on the condensed system. The pairwise aggregation is then performed according to the *nodewise* strong coupling relationship described in the introductory paragraphs of Section 5.2.3, and we note that the multiple pairwise aggregation reduces in this case to the double pairwise aggregation. The prolongation operator follows plain aggregation, i.e. is built similarly to the operator Q_T in (5.5a). The rest of the method shall be parametrized identically to U-AMG (same Krylov method, smoothers, cycle, etc.).

Comparing overall performances of two iterative methods is a difficult exercise. The convergence rate or number of iterations, alone, is not sufficient to establish a fair comparison, because the actual time to solution also depends on the iteration cost. Combining both criteria is usually made in terms of computational work or CPU time. The plain aggregation prolongation matrices, which contain only ones, is therefore applied to vectors without any theoretical flop, although its practical application still consumes non-negligible CPU time. As a consequence, we find the computational work not to be a good indicator in that case. As our U-AMG and C-AMG implementations both benefit from identical software components and optimizations, we adopt the CPU time (in sequential execution) as overall performance criterion. Additionally, classical data used to assess convergence and cost of multigrid methods shall also be given. Especially, we respectively introduce the operator and grid complexity values as

$$\mathcal{C}_{op} := \sum_{\ell=1}^{L} \frac{\mathrm{nnz}(\widetilde{A}_\ell)}{\mathrm{nnz}(\widetilde{A}_L)}, \qquad \mathcal{C}_{gd} := \sum_{\ell=1}^{L} \frac{\mathrm{rows}(\widetilde{A}_\ell)}{\mathrm{rows}(\widetilde{A}_L)}.$$

These indicators give insight into the memory requirement and the computational cost of multigrid solvers.

Given the chosen parameters, namely FCG Krylov method, Gauss-Seidel smoothers, K(1,1)-cycle, etc., C-AMG corresponds almost exactly to the algorithm implemented by AGMG in the version of [94]. One minor difference is that our algorithm omits the special treatment of strongly diagonal-dominant rows, made to manage Dirichlet boundary conditions enforced by penalization. Furthermore, the current release of the software AGMG implements a quality control over the aggregates described in [90], which may significantly improve its overall performance, especially in anisotropic cases, where the "shape" of the coarse elements plays an essential role in the convergence rate. Such a quality control preventing the formation of "bad" aggregates is omitted in our C-AMG and U-AMG algorithms. Additionally, differences in the implementation prevents a fair comparison, in terms of execution time, with the fully optimized AGMG, for which better results can reasonably be expected. The term *implementation* here refers to any factor, besides the algorithm itself, that can influence the CPU time. Typically, it includes the software technologies employed (programming language, third-party libraries, compiling options, etc.) as well as the efficiency of the coding itself. For those reasons, results obtained with the current release of AGMG shall be included for information, more as a reference to a state-of-the-art solver than as direct comparative data.

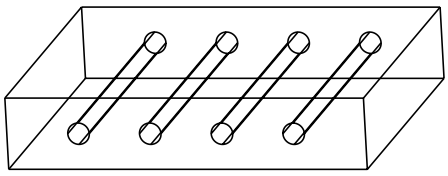
5.3.3 Numerical results

5.3.3.1 Speed and robustness

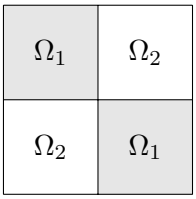
Table 5.1 describes the test cases studied. Simple and complex geometries are used, discretized by Cartesian or unstructured simplicial meshes. Tests with anisotropic and heterogeneous tensors are performed. They all gather between 3 and 6 million face unknowns, ensuring at least 6 multigrid levels. Although all but the heterogeneous one are 3D problems, we point out that the results are consistent in 2D. The test results are displayed in Table 5.2. They include the following data: operator complexity ($\mathcal{C}_{op}$); grid complexity ($\mathcal{C}_{gd}$); number of multigrid levels (L); number of iterations to reach the convergence criterion (it); asymptotic convergence rate (ϱ), defined as the geometric mean of the residual convergence ratios for the last five iterations; solve CPU time in seconds, excluding setup (t). Figure 5.7 summarizes in a comparative chart the solve CPU times of the solvers. As explained in Section 5.3.2, this figure shall concentrate most of the comments in this section.

Test case	Geometry	Mesh	Tensor	Elements	Unknowns
`Cube-cart`	Cube	Cartesian	Isotropic, homogeneous	2,097,152	6,242,304
`Cube-tet`	Cube	Unstruct. tetrahedral	Isotropic, homogeneous	1,224,179	2,418,910
`Complex-tet`	Figure 5.6a	Unstruct. tetrahedral	Isotropic, homogeneous	3,319,309	6,532,291
`Heterog1e8`	Square	Unstruct. triangular	Isotropic, heterogeneous according to Figure 5.6b	2,431,032	3,644,496
`Cube-cart-aniso100`	Cube	Cartesian	Anisotropic in the x direction, coefficient 100	2,097,152	6,242,304
`Cube-tet-aniso20`	Cube	Unstruct. tetrahedral	Anisotropic in the x direction, coefficient 20	1,224,179	2,418,910

Table 5.1 Description of the test cases



(a) Geometry of test case `Complex-tet`: 3D plate with cylindrical holes



(b) Heterogeneity pattern of test case `Heterog1e8`: for $i = 1, 2$, $\mathbf{K}_{|\Omega_i} := \kappa_i I$, with $\kappa_1/\kappa_2 = 10^8$

Figure 5.6 Supplementary figures for test cases `Complex-tet` (a) and `Heterog1e8` (b)

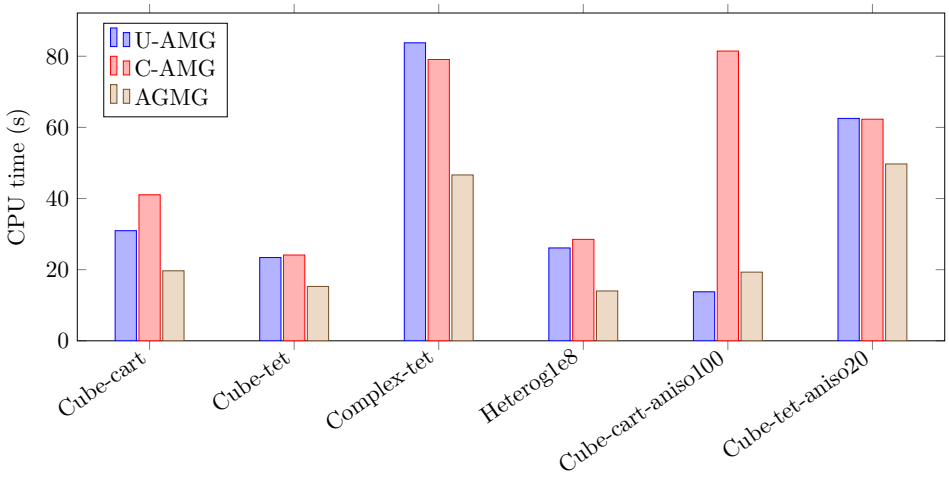


Figure 5.7 Solver comparison in CPU time

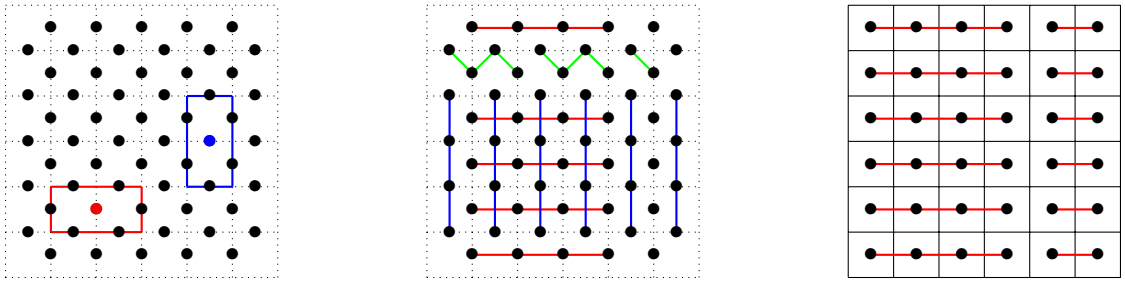
Let us first examine the dependency on the mesh. On a structured Cartesian mesh (`Cube-cart`), we remark that U-AMG is significantly faster than C-AMG (-25%). However, on the same geometry, this time with an unstructured tetrahedral mesh (`Cube-tet`), we get equivalent solve time. Finally, on a tetrahedral mesh describing a complex geometry (`Complex-tet`), the advantage of U-AMG fades out: U-AMG becomes slightly slower than C-AMG ($+6\%$). Next, on a heterogeneous problem with large coefficient jump (`Heterog1e8`), we see that both methods perform equivalently. Finally, tackling anisotropic problems, U-AMG is considerably faster than C-AMG on a Cartesian mesh (`Cube-cart-aniso100`), whereas they show comparable performance on an unstructured one (`Cube-tet-aniso20`). This set of tests demonstrates that U-AMG is favored by Cartesian meshes. To justify this result, we begin by recalling that the remaining unknowns of the condensed system are located on the faces. Indeed, viewed as nodes located at the center of the faces, these DoFs are *not* displayed, relative to each other, in a Cartesian way. See the node locations in Figure 5.8a: geometrically speaking, compared to the usual 2D Cartesian grid of element width h, the nodes form a set of rows evenly spaced by $h/2$, and where every other row has been shifted by $h/2$, giving the impression that the nodes are diagonally aligned. A fortiori, the Cartesian structure is partially lost in the sense that only one Cartesian direction is present in the stencil of each node (see red and blue stencils in Figure 5.8a). The problem for C-AMG becomes visible on an anisotropic setting, where the anisotropy follows —for instance— the x-axis. Although one wants the aggregation process to produce horizontal aggregates, the shapes actually formed are more diverse, and can even be vertical. Figure 5.8b illustrates the aggregates obtained by the double pairwise aggregation in this case: while desired horizontal aggregates are represented in red, one can also see vertical aggregates in blue, as well as "waves" in green. Referring to the red stencil of Figure 5.8a, we notice that nodes located on vertical grid lines have horizontal stencils, which allows them to be aggregated horizontally and form red aggregates. Similarly, nodes located on horizontal grid lines have inherently *vertical* stencils (in blue). Specifically, their stencils do not contain any node to aggregate with in the horizontal direction in order to comply with the anisotropy. Nodes on the same grid line are indeed not part of the stencil. Consequently, due to the values of coefficients and the game of aggregation priorities, other shapes are formed instead: vertical aggregates in blue or, better

Cube-cart	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG	1.33	1.30	7	19	0.38	31.0
C-AMG	1.51	1.34	8	15	0.25	41.1
AGMG	2.03	1.64	9	24		19.7
Cube-tet	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG	1.78	1.22	6	31	0.51	23.4
C-AMG	1.51	1.34	7	27	0.49	24.1
AGMG	1.98	1.79	7	28		15.3
Complex-tet	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG	1.76	1.22	7	31	0.51	83.8
C-AMG	1.51	1.34	8	27	0.46	79.1
AGMG	1.96	1.78	7	27		46.6
Heterog1e8	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG	1.55	1.27	7	27	0.42	26.1
C-AMG	1.42	1.34	7	23	0.38	28.5
AGMG	1.52	1.40	7	20		18.8
Cube-cart-aniso100	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG	1.32	1.33	7	10	0.15	13.8
C-AMG	1.95	1.33	8	30	0.54	81.5
AGMG	1.70	1.51	7	23		19.3
Cube-tet-aniso20	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG	1.82	1.23	6	77	0.80	62.5
C-AMG	1.60	1.43	8	75	0.78	62.3
AGMG	2.97	2.78	6	55		49.7

Table 5.2 Test results

(because closer to horizontal), waves in green. On the other hand, the reconstruction of the actual elements performed by U-AMG yields entities with fully Cartesian stencils, allowing the desired semi-coarsening; see Figure 5.8c. This explains why U-AMG performs so much better than C-AMG on the `Cube-cart-aniso100` test case. Note that this advantage is not limited to anisotropy directions that follow one of the axes; this profitable behaviour is also observed for orthotropic diffusion, namely, when the elements line up in the anisotropy direction. They can be rectangles in 2D and hexahedra in 3D, but also, more loosely, polytopes having two opposite faces orthogonal to the direction of anisotropy. However, if the mesh is fully unstructured, aggregating nodes probably offers more, or at least equivalent flexibility to follow the direction of anisotropy than aggregating elements. Hence the results obtained on the `Cube-tet-aniso20` test case, where U-AMG loses its superiority.

These remarks on the shapes of the aggregates allows us to interpret more closely the results of AGMG. As stated in Section 5.3.2, AGMG implements a complex quality control preventing bad aggregates to be formed, which we have not carried out in C-AMG. In particular, we think that aggregates such as the blue ones in Figure 5.8b (namely, those orthogonal to the direction of anisotropy) do not occur in AGMG thanks to that quality control, thus explaining the large



(a) Face DoFs and stencils. (b) Node aggregation. (c) Element aggregation.

Figure 5.8 (a) Location of the face DoFs on a Cartesian grid. (b) and (c): result of the nodewise and elementwise double pairwise aggregations, according to an anisotropic problem following the x-axis.

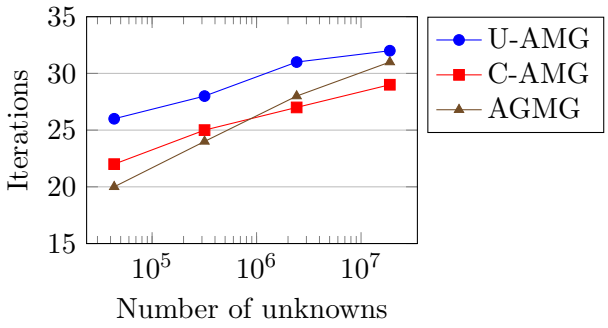


Figure 5.9 Asymptotic behaviour

performance gap between C-AMG and AGMG on the `Cube-cart-aniso100` test case. We can also suppose that, when the problem is isotropic and the mesh unstructured, there are not many bad aggregates to prevent. In that case, we can then admit that the difference in CPU time between C-AMG and AGMG results from other aspects of the implementation. Looking at the results of the test cases `Cube-tet`, `Complex-tet` and `Heterog1e8`, we can attribute 35 to 50% of the CPU time consumed by C-AMG to an implementation overhead. As U-AMG benefits from the same implementation, this proportion gives a hint on how to compare U-AMG to AGMG. Specifically, we remark that even in spite of this overhead, U-AMG still performs better than AGMG on the test case `Cube-cart-aniso100`. This indicates that the new algorithm can lead to an improved efficiency for such cases.

5.3.3.2 Asymptotic behaviour

Figure 5.9 presents, for the test case `Cube-tet` and for each solver, the number of iterations required to achieve convergence according to the number of unknowns in the system. We remark that U-AMG scales the same way as C-AMG, and slightly better than AGMG. This means that the new algorithm offers equivalent robustness to the meshsize as the existing method, and shares its algorithmic quasi-optimality.

5.3.3.3 Convergence/cost trade-off

We remark from Table 5.2 that the number of iterations required by U-AMG to reach convergence is generally higher than for C-AMG. We would like to discuss in this section the link between convergence rate and aggressiveness of coarsening.

The so-called multiple coarsening performed by U-AMG and C-AMG recursively coarsens until a desired coarsening factor (relative to the number of unknowns, i.e. the number of faces) is achieved. Note that for C-AMG, the number of required steps of coarsening is always 2, whereas for U-AMG, it needs to be higher to build the first levels, and decreases as the levels grow coarser. For actual values, refer to the number of coarsening steps performed between each level, indicated in Table 5.3 for the test case `Cube-tet`. The fact that unknowns are face unknowns, again, explains this phenomenon. Indeed, one step of coarsening corresponds to aggregating elements pairwise and collapsing faces between aggregates. Consequently, the efficient reduction of unknowns heavily relies on opportunities to collapse faces. Now, starting from a simplicial mesh, i.e. polytopes with minimal number of faces, the possibilities of collapsing faces is limited, and so is the size of the subsequent face aggregates. The situation starts to improve as the levels grow coarser because the elements then have a larger number of faces, which benefits the face collapsing process.

Level ℓ	Coarsening steps	Coarsening factor	rows($\widetilde{A}_\ell$)	nnz($\widetilde{A}_\ell$)
fine 6	-	-	2,418,910	16,757,242
5	4	5.5	440,204	9,848,798
4	3	5.4	81,081	2,702,515
3	3	7.2	11,243	416,655
2	2	4.1	2725	97,405
coarse 1	2	4.3	626	19,258

Table 5.3 Details of the adaptive multiple coarsening strategy of U-AMG for the test case `Cube-tet`

The downside of enforcing a coarsening factor, thus triggering multiple steps of coarsening, is that element aggregates can be large between two levels, which deteriorates the accuracy of the prolongation operator, and therefore that of the coarse grid correction. On the other hand, by fixing the number of coarsening steps performed between each levels, we expect a better accuracy, but costlier iterations. In order to compare both strategies, Table 5.4 presents the coarsening details when a constant number of two coarsening steps is performed. Besides the larger number of levels built due to the less aggressive coarsening, we emphasize that between the highest levels, where the coarsening factor is low, the sparsity of the operator is barely improved, which implies similar smoothing costs at those levels. Finally, we compare their respective multigrid results in Table 5.5. As expected, the fixed double coarsening strategy induces a better convergence rate than the multiple coarsening, with a number of iterations that is now lower than both C-AMG and AGMG. However, the operator and grid complexities have increased. While the grid complexity is still reasonable, in the sense that it is equivalent to that of AGMG, the operator complexity is significantly larger than with the adaptive multiple coarsening strategy, which reflects the high cost of smoothing and memory storage. All in all, the solver converges in

more CPU time, hence our choice of the multiple coarsening method. Nonetheless, the double coarsening is yet not to be discarded. Finding ways to sparsen the coarse operators in order to optimize the trade-off between convergence rate and operator complexity is another research path.

Level ℓ	Coarsening steps	Coarsening factor	rows($\widetilde{A}_\ell$)	nnz($\widetilde{A}_\ell$)
fine 8	-	-	2,418,910	16,757,242
7	2	2.2	1,101,412	15,003,704
6	2	2.5	444,030	10,002,480
5	2	3.0	148,892	4,530,558
4	2	3.5	43,078	1,515,066
3	2	3.8	11,357	416,155
2	2	4.0	2846	101,960
coarse 1	2	4.2	681	20,925

Table 5.4 Details of the fixed double coarsening strategy of U-AMG for the test case `Cube-tet`

`Cube-tet`	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
U-AMG (multiple coarsening)	1.78	1.22	6	31	0.51	23.4
U-AMG (double coarsening)	2.88	1.72	8	25	0.46	25.5
C-AMG	1.51	1.34	7	27	0.49	24.1
AGMG	1.98	1.79	7	28		15.3

Table 5.5 Comparative solver results

5.3.4 Alternative algorithms

In order to justify our algorithmic choices, we present supplementary numerical results using alternative prolongation operators. In particular, we want to compare the results of our method with those obtained using Q_F as prolongation operator (cf. Section 5.2.6). Indeed, since Q_F is used to build coarse levels in the setup phase, re-using it as the prolongation operator in the multigrid iterations comes as a more straightforward solution than constructing a new operator. Second, in order to evaluate the effect of the partial smoothing (cf. J in (5.10)), we also consider the multigrid method without this enhancement. Namely, it corresponds to using $P_F^{(0)}$ (cf. (5.8)) as prolongation operator instead of P_F, and to introduce the operator Q_F^{smooth} as the counterpart of Q_F, enhanced with the same partial smoothing. Let us first consider the results obtained on the `Cube-tet` test case, given in the top half of Table 5.6. While plain Q_F provides a faster solver than $P_F^{(0)}$, the addition of the partial smoothing makes the final P_F and Q_F^{smooth} give equivalent results. In particular, the addition of one Jacobi sweep significantly improves the convergence rate of $P_F^{(0)}$, resulting in a non-negligible reduction of the CPU time, whereas no notable improvement is observed with Q_F. Although the results given by P_F and Q_F^{smooth} on isotropic test cases do not present much difference, the better robustness of P_F manifests itself on the anisotropic test case `Cube-cart-aniso100`. Indeed, with or without additional smoothing, the method based on P_F gives significantly better results than that based on Q_F. This difference can be explained by the simplicity of Q_F. Clearly, assigning the mere average value of the local

boundary faces to the DoFs on the local interior faces does not take the anisotropic coefficient into account. On the other hand, the decondensation of the cell unknowns performed by P_F through formula (5.9) successfully does so.

`Cube-tet`	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
P_F	1.78	1.22	6	31	0.51	23.4
Q_F^{smooth}	1.79	1.22	6	30	0.51	23.4
$P_F^{(0)}$	1.80	1.22	6	32	0.56	28.0
Q_F	1.80	1.22	6	31	0.51	24.1
`Cube-cart-aniso100`	$\mathcal{C}_{op}$	$\mathcal{C}_{gd}$	L	it	ϱ	t
P_F	1.32	1.33	7	10	0.15	13.8
Q_F^{smooth}	1.32	1.32	7	15	0.36	22.1
$P_F^{(0)}$	1.32	1.32	7	19	0.47	28.8
Q_F	1.31	1.30	7	27	0.56	39.3

Table 5.6 Results with alternative prolongation operators

5.4 Conclusion

The solver developed in this work proposes an alternative AMG approach for the solution of linear systems arising from lowest-order hybrid discretizations. Although not entirely "black-box" (because it requires parts of the uncondensed system), it remains purely algebraic. Compared to the equivalent aggregation-based AMG constructed in the standard way (i.e. by viewing system unknowns as nodes), it shows similar performance in most cases, while being more robust with respect to orthotropic anisotropy. Consequently, it can offer substantial added value for solving problems comprising both isotropic and anisotropic regions, like, e.g., Darcy flows. The solver, in this case, allies the flexibility of AMG to handle unstructured meshes on isotropic regions while exploiting the special element shapes on anisotropic ones. The cost of this improvement is payed during the setup phase: (i) more memory storage may be required because of the use of the uncondensed matrix; but the blocks needed by the setup may be kept in storage anyway, because they are also needed to recover the cell unknowns after solving the condensed system. (ii) As it requires to reconstruct the elements unknowns, the coarsening strategy is less direct than other AMG methods, which evidently implies a costlier setup.

Hybrid discretizations achieve their full potential in high order of approximation. Yet, this solver only applies to the lowest order. Even for more classical, non-hybrid discretizations, purely algebraic solvers for higher orders are still an open problem. In aggregation-based methods, the difficulty lies in the transfer of high order components from the coarse unknowns to the fine ones they aggregate. In this context, the elementwise view of the aggregation process is certainly easier to work with and geometrically interpret than a face aggregation.

Chapter 6

Conclusion and perspectives

For every problem, there is one solution which is simple, neat, and wrong.

H. L. Mencken (1880-1956)

This Ph.D thesis adds to the broad spectrum of multigrid methods two novel skeleton-based algorithms for statically condensed systems: one geometric, the other algebraic. To HHO, it brings efficient options for the solution of large systems arising from elliptic equations. As such, these solvers contribute to extend the HHO ecosystem and favor its practical usability for industrial applications. Equipped with efficient solvers, hybrid discretizations may also have a role to play in the pursuit of exascale computing. Indeed, their compact stencils and, therefore, local assemblies and local reconstructions of the solution, make them well-suited to parallel implementation. Furthermore, leveraging high orders of approximation can be done at lower cost thanks to the static condensation. This could allow to reach the same accuracy as other discretization methods for less unknowns to solve. For instance, what would be acknowledged as an exascale problem in FEM may not be considered as such in HHO, provided a high degree of approximation.

More generally, this thesis focuses on the definition of the solvers from a mathematical standpoint, leaving aside their implementation for parallel architectures. Applying state-of-the-art computer science techniques is the next step to prove their practical scalability and pave the way towards efficient industrial software applications. Novel code generation techniques [82, 84, 107], matrix-free implementations thanks to hierarchical hybrid grids [67, 81], hardware optimization through the generation of block structured grids [125], especially retain attention.

Although our geometric multigrid method, by conserving the same polynomial degree at every level, exhibits asymptotic optimality for high orders, one may want to compare its performance, in terms of overall cost, against the usual alternative, namely, the p-multigrid approach. This comparison deserves a dedicated study, which will be part of future work.

Scalar elliptic equations constituted the natural starting point of the research for efficient solvers. Symmetry, positive-definiteness and ellipticity are indeed aggreeable properties to multigrid methods. More difficult settings lie ahead with the Stokes and Navier-Stokes equations, logical sequels of this thesis in the context of CFD applications.

Bibliography

[1] Mark F. Adams. Parallel multigrid solvers for 3D unstructured finite element problems in large deformation elasticity and plasticity. *International Journal for Numerical Methods in Engineering*, 48(8):1241–1262, 2000. doi:10.1002/(SICI)1097-0207(20000720)48:8<1241::AID-NME946>3.0.CO;2-R.

[2] Joubine Aghili, Daniele A. Di Pietro, and Berardo Ruffini. An hp-Hybrid High-Order Method for Variable Diffusion on General Meshes. *Computational Methods in Applied Mathematics*, 17(3):359–376, 2017. doi:10.1515/cmam-2017-0009.

[3] P. F. Antonietti, P. Houston, X. Hu, M. Sarti, and M. Verani. Multigrid algorithms for *hp*-version interior penalty discontinuous Galerkin methods on polygonal and polyhedral meshes. *Calcolo*, 54(4):1169–1198, 2017. doi:10.1007/s10092-017-0223-6.

[4] P. F. Antonietti and G. Pennesi. V-cycle Multigrid Algorithms for Discontinuous Galerkin Methods on Non-nested Polytopic Meshes. *Journal of Scientific Computing*, 78(1):625–652, 2019. doi:10.1007/s10915-018-0783-x.

[5] Paola F. Antonietti, Marco Sarti, and Marco Verani. Multigrid Algorithms for hp-Discontinuous Galerkin Discretizations of Elliptic Problems. *SIAM Journal on Numerical Analysis*, 53(1):598–618, 2015. doi:10.1137/130947015.

[6] Paola F. Antonietti, Marco Sarti, and Marco Verani. Multigrid Algorithms for High Order Discontinuous Galerkin Methods. In Thomas Dickopf, Martin J. Gander, Laurence Halpern, Rolf Krause, and Luca F. Pavarino, editors, *Domain Decomposition Methods in Science and Engineering XXII*, volume 104, pages 3–13. Springer International Publishing, Cham, 2016. doi:10.1007/978-3-319-18827-0_1.

[7] Frédéric Archambeau, Namane Méchitoua, and Marc Sakiz. Code Saturne: A Finite Volume Code for the computation of turbulent incompressible flows - Industrial Applications. *International Journal on Finite Volumes*, 1(1), 2004. URL: http://ijfv.math.cnrs.fr/spip.php?article3.

[8] Arnold, D. N. and Brezzi, F. Mixed and nonconforming finite element methods : implementation, postprocessing and error estimates. *ESAIM: M2AN*, 19(1):7–32, 1985. doi:10.1051/m2an/1985190100071.

[9] A. Arrarás, F. J. Gaspar, L. Portero, and C. Rodrigo. Geometric multigrid methods for Darcy–Forchheimer flow in fractured porous media. *Computers & Mathematics with Applications*, 78(9):3139–3151, 2019. doi:10.1016/j.camwa.2019.04.031.

[10] Andrés Arrarás, Francisco J. Gaspar, Laura Portero, and Carmen Rodrigo. Multigrid solvers for multipoint flux approximations of the Darcy problem on rough quadrilateral grids. *Computational Geosciences*, 25(2):715–730, 2021. doi:10.1007/s10596-020-09979-w.

[11] Allison H. Baker, Axel Klawonn, Tzanio Kolev, Martin Lanser, Oliver Rheinbach, and Ulrike Meier Yang. Scalability of Classical Algebraic Multigrid for Elasticity to Half a Million Parallel Tasks. In Hans-Joachim Bungartz, Philipp Neumann, and Wolfgang E. Nagel, editors, *Software for Exascale Computing - SPPEXA 2013-2015*, Lecture Notes in Computational Science and Engineering, pages 113–140, Cham, 2016. Springer International Publishing. doi:10.1007/978-3-319-40528-5_6.

[12] Peter Bastian, Markus Blatt, and Robert Scheichl. Algebraic multigrid for discontinuous Galerkin discretizations of heterogeneous elliptic problems. *Numerical Linear Algebra with Applications*, 19(2):367–388, 2012. doi:10.1002/nla.1816.

[13] Peter Bastian and Volker Reichenberger. Multigrid for Higher Order Discontinuous Galerkin Finite Elements Applied to Groundwater Flow. Technical Report, 2000.

[14] S. Bauer, M. Mohr, U. Rüde, J. Weismüller, M. Wittmann, and B. Wohlmuth. A two-scale approach for efficient on-the-fly operator assembly in massively parallel high performance multigrid codes. *Applied Numerical Mathematics*, 122:14–38, 2017. doi:10.1016/j.apnum.2017.07.006.

[15] Benjamin Karl Bergen and Frank Hülsemann. Hierarchical hybrid grids: data structures and core algorithms for multigrid. *Numerical Linear Algebra with Applications*, 11(2-3):279–291, 2004. doi:10.1002/nla.382.

[16] J. Bey. Tetrahedral grid refinement. *Computing*, 55(4):355–378, 1995. doi:10.1007/BF02238487.

[17] Markus Blatt. *A Parallel Algebraic Multigrid Method for Elliptic Problems with Highly Discontinuous Coefficients*. PhD thesis, Ruprecht-Karls-Universität, Heidelberg, 2010. URL: https://www.researchgate.net/publication/45251415_A_Parallel_Algebraic_Multigrid_Method_for_Elliptic_Problems_with_Highly_Discontinuous_Coefficients.

[18] Daniele Boffi, Michele Botti, and Daniele A. Di Pietro. A nonconforming high-order method for the biot problem on general meshes. *SIAM Journal on Scientific Computing*, 38(3):A1508–A1537, 2016. doi:10.1137/15M1025505.

[19] Bonelle, Jérôme and Ern, Alexandre. Analysis of compatible discrete operator schemes for elliptic problems on polyhedral meshes. *ESAIM: M2AN*, 48(2):553–581, 2014. doi:10.1051/m2an/2013104.

[20] L. Botti, A. Colombo, A. Crivellini, and M. Franciolini. h-p-hp-Multilevel discontinuous Galerkin solution strategies for elliptic operators. *International Journal of Computational Fluid Dynamics*, 33(9):362–370, 2019. doi:10.1080/10618562.2019.1688306.

[21] Lorenzo Botti, Daniele A. Di Pietro, and Jérôme Droniou. A Hybrid High-Order method for the incompressible Navier–Stokes equations based on Temam's device. *Journal of Computational Physics*, 376:786 – 816, 2019. doi:10.1016/j.jcp.2018.10.014.

[22] Lorenzo Botti and Daniele Antonio Di Pietro. p-Multilevel preconditioners for HHO discretizations of the Stokes equations with static condensation. *Communications on Applied Mathematics and Computation*, 2021. To appear. doi:10.1007/s42967-021-00142-5.

[23] Michele Botti, Daniele A. Di Pietro, and Pierre Sochala. A Hybrid High-Order method for nonlinear elasticity. *SIAM Journal on Numerical Analysis*, 55(6):2687–2717, 2017. doi:10.1137/16M1105943.

[24] Michele Botti, Daniele A. Di Pietro, and Pierre Sochala. A Hybrid High-Order discretization method for nonlinear poroelasticity. *Computational Methods in Applied Mathematics*, 20(2):227 – 249, 01 Apr. 2020. doi:10.1515/cmam-2018-0142.

[25] Michele Botti, Daniel Castanon Quiroz, Daniele A. Di Pietro, and André Harnist. A Hybrid High-Order method for creeping flows of non-Newtonian fluids. *Calcolo*, 58(2):19, 2021. doi:10.1007/s10092-021-00410-z.

[26] D. Braess. Towards algebraic multigrid for elliptic problems of second order. *Computing*, 55(4):379–393, 1995. doi:10.1007/BF02238488.

[27] D. Braess, M. Dryja, and W. Hackbush. A Multigrid Method for Nonconforming FE-Discretisations with Application to Non-Matching Grids. *Computing*, 63(1):1–25, 1999. doi:10.1007/s006070050048.

[28] Achi Brandt and Oren E Livne. *Multigrid techniques: 1984 guide with applications to fluid dynamics*. SIAM, 2011.

[29] M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge. Algebraic Multigrid Based on Element Interpolation (AMGe). *SIAM Journal on Scientific Computing*, 22(5):1570–1592, 2001. doi:10.1137/S1064827598344303.

[30] Franco Brezzi, Jim Douglas, and L. D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2):217–235, 1985. doi:10.1007/BF01389710.

[31] William L. Briggs, Van Emden Henson, and Steve F. McCormick. *A multigrid tutorial*. Society for Industrial and Applied Mathematics, 2000.

[32] Peter R. Brune, Matthew G. Knepley, and L. Ridgway Scott. Unstructured Geometric Multigrid in Two and Three Dimensions on Complex and Graded Meshes. *SIAM Journal on Scientific Computing*, 35(1):A173–A191, 2013. doi:10.1137/110827077.

[33] V. E. Bulgakov. Multi-level iterative technique and aggregation concept with semi-analytical preconditioning for solving boundary-value problems. *Communications in Numerical Methods in Engineering*, 9(8):649–657, 1993. doi:10.1002/cnm.1640090804.

[34] Florent Chave, Daniele A. Di Pietro, Fabien Marche, and Franck Pigeonneau. A Hybrid High-Order Method for the Cahn–Hilliard problem in Mixed Form. *SIAM Journal on Numerical Analysis*, 54(3):1873–1898, 2016. Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/15M1041055.

[35] P. G Ciarlet. *The Finite Element Method for Elliptic Problems*. Elsevier, Burlington, 1978. URL: http://www.123library.org/book_details/?id=41072.

[36] B. Cockburn, D. A. Di Pietro, and A. Ern. Bridging the Hybrid High-Order and Hybridizable Discontinuous Galerkin methods. *ESAIM: Math. Model Numer. Anal.*, 50(3):635–650, 2016. doi:10.1051/m2an/2015051.

[37] B. Cockburn, O. Dubois, J. Gopalakrishnan, and S. Tan. Multigrid for an HDG method. *IMA Journal of Numerical Analysis*, 34(4):1386–1425, 2014.

[38] Bernardo Cockburn, Bo Dong, and Johnny Guzmán. A superconvergent LDG-hybridizable Galerkin method for second-order elliptic problems. *Mathematics of Computation*, 77(264):1887–1916, 2008. doi:10.1090/S0025-5718-08-02123-6.

[39] Bernardo Cockburn, Jayadeep Gopalakrishnan, and Raytcho Lazarov. Unified Hybridization of Discontinuous Galerkin, Mixed, and Continuous Galerkin Methods for Second Order Elliptic Problems. *SIAM Journal on Numerical Analysis*, 47(2):1319–1365, 2009. doi:10.1137/070706616.

[40] Bernardo Cockburn, Johnny Guzmán, and Haiying Wang. Superconvergent discontinuous Galerkin methods for second-order elliptic problems. *Mathematics of Computation*, 78(265):1–1, 2009. doi:10.1090/S0025-5718-08-02146-7.

[41] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. In *Proceedings of the 1969 24th national conference*, ACM '69, pages 157–172, New York, NY, USA, 1969. Association for Computing Machinery. doi:10.1145/800195.805928.

[42] Lourenço Beirão da Veiga, Konstantin Lipnikov, and Gianmarco Manzini. *The Mimetic Finite Difference Method for Elliptic Problems*. Springer International Publishing, Cham, 2014. doi:10.1007/978-3-319-02663-3.

[43] D. A. Di Pietro and J. Droniou. *The Hybrid High-Order method for polytopal meshes*. Number 19 in Modeling, Simulation and Application. Springer International Publishing, 2020. doi:10.1007/978-3-030-37203-3.

[44] D. A. Di Pietro and A. Ern. A hybrid high-order locking-free method for linear elasticity on general meshes. *Comput. Meth. Appl. Mech. Engrg.*, 283:1–21, 2015. doi:10.1016/j.cma.2014.09.009.

[45] D. A. Di Pietro, A. Ern, and S. Lemaire. An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators. *Comput. Meth. Appl. Math.*, 14(4):461–472, 2014. Open access (editor's choice). doi:10.1515/cmam-2014-0018.

[46] Daniele A. Di Pietro and Alexandre Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Mathématiques et Applications. Springer-Verlag, Berlin Heidelberg, 2012. URL: www.springer.com/us/book/9783642229794.

[47] Daniele A. Di Pietro and Alexandre Ern. Equilibrated tractions for the hybrid high-order method. *Comptes Rendus Mathematique*, 353(3):279 – 282, 2015. doi:10.1016/j.crma.2014.12.009.

[48] Daniele A. Di Pietro and Alexandre Ern. Arbitrary-order mixed methods for heterogeneous anisotropic diffusion on general meshes. *IMA Journal of Numerical Analysis*, 37(1):40–63, 2016. doi:10.1093/imanum/drw003.

[49] Daniele A. Di Pietro, Frank Hülsemann, Pierre Matalon, Paul Mycek, and Ulrich Rüde. Algebraic multigrid preconditioner for statically condensed systems arising from lowest-order hybrid discretizations. Submitted, 2021. URL: https://hal.archives-ouvertes.fr/hal-03272468.

[50] Daniele A. Di Pietro, Frank Hülsemann, Pierre Matalon, Paul Mycek, Ulrich Rüde, and Daniel Ruiz. An h-multigrid method for Hybrid High-Order discretizations. *SIAM Journal on Scientific Computing*, pages S839–S861, 2021. Open access: hal-02434411. doi:10.1137/20M1342471.

[51] Daniele A. Di Pietro, Frank Hülsemann, Pierre Matalon, Paul Mycek, Ulrich Rüde, and Daniel Ruiz. Towards robust, fast solutions of elliptic equations on complex domains through HHO discretizations and non-nested multigrid methods. *International Journal for Numerical Methods in Engineering*, 122(22):6576–6595, 2021. Open access: hal-03163476. doi:10.1002/nme.6803.

[52] Thomas Dickopf and Rolf Krause. Evaluating local approximations of the l2-orthogonal projection between non-nested finite element spaces. *Numerical Mathematics: Theory, Methods and Applications*, 7(3):288–316, 2014. doi:10.1017/S100489790000012X.

[53] Veselin Asenov Dobrev. *Preconditioning of Discontinuous Galerkin Methods for Second Order Elliptic Problems*. PhD thesis, Texas A&M University, 2007. URL: https://pdfs.semanticscholar.org/2c28/4f402744a43e046fd357b24413b0305ecf7d.pdf.

[54] Jérôme Droniou and Robert Eymard. A mixed finite volume scheme for anisotropic diffusion problems on any grid. *Numerische Mathematik*, 105(1):35–71, 2006. doi:10.1007/s00211-006-0034-1.

[55] Jérôme Droniou, Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. A unified approach to mimetic finite difference, hybrid finite volume and mixed finite volume methods. *Mathematical Models and Methods in Applied Sciences*, 20(02):265–295, 2010. doi:10.1142/S0218202510004222.

[56] H. Egger, U. Rüde, and B. Wohlmuth. Energy-Corrected Finite Element Methods for Corner Singularities. *SIAM Journal on Numerical Analysis*, 52(1):171–193, 2014. doi:10.1137/120871377.

[57] R. Eymard, T. Gallouët, and R. Herbin. Discretization of heterogeneous and anisotropic diffusion problems on general nonconforming meshes SUSHI: a scheme using stabilization and hybrid interfaces. *IMA Journal of Numerical Analysis*, 30(4):1009–1043, 2010. doi:10.1093/imanum/drn084.

[58] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite Volume Methods. In J. L. Lions and Philippe Ciarlet, editors, *Solution of Equation in Rn (Part 3), Techniques of Scientific Computing (Part 3)*, volume 7 of *Handbook of Numerical Analysis*, pages 713–1020. Elsevier, 2000. doi:10.1016/S1570-8659(00)07005-8.

[59] Robert D Falgout. An Introduction to Algebraic Multigrid. *Computing in Science & Engineering*, 8(6):24–33, 2006. doi:10.1109/MCSE.2006.105.

[60] Niklas Fehn, Peter Munch, Wolfgang A. Wall, and Martin Kronbichler. Hybrid multigrid methods for high-order discontinuous Galerkin discretizations. *Journal of Computational Physics*, 415:109538, 2020. doi:10.1016/j.jcp.2020.109538.

[61] Y. T. Feng, D. Perić, and D. R. J. Owen. A non-nested Galerkin multi-grid method for solving linear and nonlinear solid mechanics problems. *Computer Methods in Applied Mechanics and Engineering*, 144(3):307–325, 1997. doi:10.1016/S0045-7825(96)01183-8.

[62] Krzysztof J Fidkowski. A High-Order Discontinuous Galerkin Multigrid Solver for Aerodynamic Applications. Master's thesis, Massachusetts Institute of Technology, 2004.

[63] J. Fish, M. Pandheeradi, and V. Belsky. An efficient multilevel solution scheme for large scale non-linear systems. *International Journal for Numerical Methods in Engineering*, 38(10):1597–1610, 1995. doi:10.1002/nme.1620381002.

[64] Jerome Francescatto and Alain Dervieux. A semi-coarsening strategy for unstructured multigrid based on agglomeration. *International Journal for Numerical Methods in Fluids*, 26(8):927–957, 1998. doi:10.1002/(SICI)1097-0363(19980430)26:8<927::AID-FLD679>3.0.CO;2-0.

[65] Matteo Franciolini, Krzysztof J. Fidkowski, and Andrea Crivellini. Efficient discontinuous Galerkin implementations and preconditioners for implicit unsteady compressible flow simulations. *Computers & Fluids*, 203:104542, 2020. doi:10.1016/j.compfluid.2020.104542.

[66] Geert-Jan Giezeman and Wieger Wesselink. 2D polygons. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.1 edition, 2020. URL: https://doc.cgal.org/5.1/Manual/packages.html#PkgPolygon2.

[67] Björn Gmeiner, Harald Köstler, Markus Stürmer, and Ulrich Rüde. Parallel multigrid on hierarchical hybrid grids: a performance study on current high performance computing clusters. *Concurrency and Computation: Practice and Experience*, 26(1):217–240, 2014. doi:10.1002/cpe.2968.

[68] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numerische Mathematik*, 95(3):527–550, 2003. doi:10.1007/s002110200392.

[69] Jayadeep Gopalakrishnan and Shuguang Tan. A convergent multigrid cycle for the hybridized mixed method. *Numerical Linear Algebra with Applications*, 16(9):689–714, 2009. doi:10.1002/nla.636.

[70] P.W. Hemker and M.H. van Raalte. Fourier two-level analysis for higher dimensional discontinuous Galerkin discretisation. *Computing and Visualization in Science*, 7(3):159–172, 2004. doi:10.1007/s00791-004-0136-1.

[71] Van Emden Henson and Ulrike Meier Yang. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41(1):155–177, 2002. doi:10.1016/S0168-9274(01)00115-5.

[72] Michael Holst and Stefan Vandewalle. Schwarz methods: to symmetrize or not to symmetrize. *SIAM Journal on Numerical Analysis*, 34(2):699–722, 1997.

[73] V. John, P. Knobloch, G. Matthies, and L. Tobiska. Non-Nested Multi-Level Solvers for Finite Element Discretisations of Mixed Problems. *Computing*, 68(4):313–341, 2002. doi:10.1007/s00607-002-1444-2.

[74] Guido Kanschat. Robust smoothers for high-order discontinuous Galerkin discretizations of advection–diffusion problems. *Journal of Computational and Applied Mathematics*, 218(1):53–60, 2008.

[75] R. B. Kellogg. Singularities in interface problems. In BERT Hubbard, editor, *Numerical Solution of Partial Differential Equations–II*, pages 351–400. Academic Press, 1971.

[76] A. Klawonn, M. Lanser, O. Rheinbach, and J. Weber. Preconditioning the coarse problem of BDDC methods - three-level, algebraic multigrid, and vertex-based preconditioners. 2019.

[77] Axel Klawonn, Martin Lanser, and Oliver Rheinbach. Using Algebraic Multigrid in Inexact BDDC Domain Decomposition Methods. In Petter E. Bjørstad, Susanne C. Brenner, Lawrence Halpern, Hyea Hyun Kim, Ralf Kornhuber, Talal Rahman, and Olof B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering XXIV*, Lecture Notes in Computational Science and Engineering, pages 425–433, Cham, 2018. Springer International Publishing. doi:10.1007/978-3-319-93873-8_40.

[78] Nils Kohl, Dominik Thönnes, Daniel Drzisga, Dominik Bartuschat, and Ulrich Rüde. The HyTeG finite-element software framework for scalable multigrid solvers. *International Journal of Parallel, Emergent and Distributed Systems*, 34(5):477–496, 2019. doi:10.1080/17445760.2018.1506453.

[79] Bruno Koobus, Marie-Hélène Lallemand, and Alain Dervieux. Unstructured volume-agglomeration MG: Solution of the Poisson equation. *International Journal for Numerical Methods in Fluids*, 18(1):27–42, 1994. doi:10.1002/fld.1650180103.

[80] M. Kronbichler and W. Wall. A Performance Comparison of Continuous and Discontinuous Galerkin Methods with Fast Multigrid Solvers. *SIAM Journal on Scientific Computing*, 40(5):A3423–A3448, 2018. doi:10.1137/16M110455X.

[81] Sebastian Kuckuk, Björn Gmeiner, Harald Köstler, and Ulrich Rüde. A Generic Prototype to Benchmark Algorithms and Data Structures for Hierarchical Hybrid Grids. *Parallel Computing: Accelerating Computational Science and Engineering (CSE)*, pages 813–822, 2014. Publisher: IOS Press. doi:10.3233/978-1-61499-381-0-813.

[82] Sebastian Kuckuk and Harald Köstler. Generation of Highly Parallel Multigrid Solvers for CFD Applications. In *SIAM Conference on Parallel Processing for Scientific Computing 2018*, 2018.

[83] C. Lehrenfeld. *Hybrid Discontinuous Galerkin methods for solving incompressible flow problems*. PhD thesis, Rheinisch-Westfälischen Technischen Hochschule, Aachen, 2010.

[84] Christian Lengauer, Sven Apel, Mathias Bolten, Shigeru Chiba, Ulrich Rüde, Jürgen Teich, Armin Größlinger, Frank Hannig, Harald Köstler, Lisa Claus, Alexander Grebhahn, Stefan Groth, Stefan Kronawitter, Sebastian Kuckuk, Hannah Rittich, Christian Schmitt, and Jonas Schmitt. ExaStencils – Advanced Multigrid Solver Generation. In *Software for Exascale Computing – SPPEXA 2016-2019*, volume 136 of *Lecture Notes in Computer Science and Engineering*, pages 405–452. Springer, 2020. doi:10.1007/978-3-030-47956-5_14.

[85] Bruno Lévy and Yang Liu. Lp centroidal voronoi tessellation and its applications. *ACM Trans. Graph.*, 29(4), 2010. doi:10.1145/1778765.1778856.

[86] Peipei Lu, Andreas Rupp, and Guido Kanschat. HMG – Homogeneous multigrid for HDG. 2020. URL: http://arxiv.org/abs/2011.14018.

[87] E. Morano, D. J. Mavriplis, and V. Venkatakrishnan. Coarsening Strategies for Unstructured Multigrid Techniques with Application to Anisotropic Problems. *SIAM Journal on Scientific Computing*, 20(2):393–415, 1998. doi:10.1137/S1064827595287638.

[88] Sriramkrishnan Muralikrishnan, Tan Bui-Thanh, and John N. Shadid. A multilevel approach for trace system in HDG discretizations. *Journal of Computational Physics*, 407:109240, 2020. doi:10.1016/j.jcp.2020.109240.

[89] Adrian C. Muresan and Yvan Notay. Analysis of Aggregation-Based Multigrid. *SIAM Journal on Scientific Computing*, 30(2):1082–1103, 2008. doi:10.1137/060678397.

[90] Artem Napov and Yvan Notay. An Algebraic Multigrid Method with Guaranteed Convergence Rate. *SIAM Journal on Scientific Computing*, 34(2):A1079–A1109, 2012. doi:10.1137/100818509.

[91] Cristian R. Nastase and Dimitri J. Mavriplis. High-order discontinuous Galerkin methods using an hp-multigrid approach. *Journal of Computational Physics*, 213(1):330–357, 2006. doi:10.1016/j.jcp.2005.08.022.

[92] N. C. Nguyen, J. Peraire, and B. Cockburn. An implicit high-order hybridizable discontinuous galerkin method for linear convection-diffusion equations. *Journal of Computational Physics*, 228(9):3232–3254, 2009. doi:10.1016/j.jcp.2009.01.030.

[93] Yvan Notay. Flexible Conjugate Gradients. *SIAM Journal on Scientific Computing*, 22(4):1444–1460, 2000. doi:10.1137/S1064827599362314.

[94] Yvan Notay. An aggregation-based algebraic multigrid method. *Electronic Transactions on Numerical Analysis*, 37:123–146, 2010.

[95] Yvan Notay and Artem Napov. A massively parallel solver for discrete Poisson-like problems. *Journal of Computational Physics*, 281:237–250, 2015. doi:10.1016/j.jcp.2014.10.043.

[96] Issei Oikawa. A Hybridized Discontinuous Galerkin Method with Reduced Stabilization. *Journal of Scientific Computing*, 65(1):327–340, 2015. doi:10.1007/s10915-014-9962-6.

[97] Carl Ollivier-Gooch. Coarsening unstructured meshes by edge contraction. *International Journal for Numerical Methods in Engineering*, 57(3):391–414, 2003. doi:10.1002/nme.682.

[98] Luke N. Olson and Jacob B. Schroder. Smoothed aggregation multigrid solvers for high-order discontinuous Galerkin methods for elliptic problems. *Journal of Computational Physics*, 230(18):6959–6976, 2011.

[99] Miroslav S. Petrov and Todor D. Todorov. Refinement strategies related to cubic tetrahedral meshes. *Applied Numerical Mathematics*, 137:169 – 183, 2019. doi:10.1016/j.apnum.2018.11.006.

[100] F. Prill, Mária Lukácová-Medvid'ová, and Ralf Hartmann. Smoothed Aggregation Multigrid for the Discontinuous Galerkin Method. *SIAM Journal on Scientific Computing*, 31:3503–3528, 2009. doi:10.1137/080728457.

[101] Alfio Quarteroni. *Numerical models for differential problems*. Number Volume 16 in MS&A - Modeling, simulation and applications. Springer, Cham, third edition, corrected publication 2018 edition, 2018.

[102] P. A. Raviart and J. M. Thomas. A mixed finite element method for 2-nd order elliptic problems. In Ilio Galligani and Enrico Magenes, editors, *Mathematical Aspects of Finite Element Methods*, pages 292–315, Berlin, Heidelberg, 1977. Springer Berlin Heidelberg.

[103] Nathan V. Roberts and Jesse Chan. A geometric multigrid preconditioning strategy for dpg system matrices. *Computers & Mathematics with Applications*, 74(8):2018—-2043, 2017.

[104] J. W. Ruge and K. Stüben. 4. Algebraic Multigrid. In *Multigrid Methods*, Frontiers in Applied Mathematics, pages 73–130. Society for Industrial and Applied Mathematics, 1987. doi:10.1137/1.9781611971057.ch4.

[105] John Ruge and Klaus Stüben. Efficient solution of finite difference and finite element equations by algebraic multigrid (AMG). GMD, 1984.

[106] David Salinas, Florent Lafarge, and Pierre Alliez. Structure-Aware Mesh Decimation. *Computer Graphics Forum*, page 20, 2015. URL: https://hal.inria.fr/hal-01111203.

[107] Jonas Schmitt, Sebastian Kuckuk, and Harald Köstler. Constructing Efficient Multigrid Solvers with Genetic Programming. In Association for Computing Machinery, editor, *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*, pages 1012 – 1020, New York, NY, USA, 2020. doi:10.1145/3377930.3389811.

[108] Joachim Schoeberl and Christoph Lehrenfeld. Domain Decomposition Preconditioning for High Order Hybrid Discontinuous Galerkin Methods on Tetrahedral Meshes. In T. Apel and O. Steinbach, editors, *Advanced Finite Element Methods and Applications*, volume 66 of *Lecture Notes in Applied and Computational Mechanics*, pages 27–56. Springer, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-30316-6_2.

[109] Jochen Schütz and Vadym Aizinger. A hierarchical scale separation approach for the hybridized discontinuous Galerkin method. *Journal of Computational and Applied Mathematics*, 317:500–509, 2017. doi:10.1016/j.cam.2016.12.018.

[110] L. Ridgway Scott and Shangyou Zhang. Higher-Dimensional Nonnested Multigrid Methods. *Mathematics of Computation*, 58(198):457–466, 1992. doi:10.2307/2153196.

[111] Benjamin Stamm and Thomas P. Wihler. hp-Optimal discontinuous Galerkin methods for linear elliptic problems. *Mathematics of Computation*, 79(272):2117–2117, 2010. doi:10.1090/S0025-5718-10-02335-5.

[112] K. Stüben. A review of algebraic multigrid. In C. Brezinski and L. Wuytack, editors, *Numerical Analysis: Historical Developments in the 20th Century*, pages 331–359. Elsevier, Amsterdam, 2001. doi:10.1016/B978-0-444-50617-7.50015-X.

[113] Hari Sundar, George Biros, Carsten Burstedde, Johann Rudi, Omar Ghattas, and Georg Stadler. Parallel geometric-algebraic multigrid on unstructured forests of octrees. In *SC '12: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, pages 1–11, 2012. doi:10.1109/SC.2012.91.

[114] Stefan Takacs. Fast multigrid solvers for conforming and non-conforming multi-patch Isogeometric Analysis. *Computer Methods in Applied Mechanics and Engineering*, 371:113301, 2020. doi:10.1016/j.cma.2020.113301.

[115] Cameron Talischi, Glaucio H. Paulino, Anderson Pereira, and Ivan F. M. Menezes. Polygonal finite elements for topology optimization: A unifying paradigm. *International Journal for Numerical Methods in Engineering*, 82(6):671–698, 2010. doi:10.1002/nme.2763.

[116] S. J. Thomas, S. Ananthan, S. Yellapantula, J. J. Hu, M. Lawson, and M. A. Sprague. A Comparison of Classical and Aggregation-Based Algebraic Multigrid Preconditioners for High-Fidelity Simulation of Wind Turbine Incompressible Flows. *SIAM Journal on Scientific Computing*, 41(5):S196–S219, 2019. Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/18M1179018.

[117] Todor D. Todorov. The optimal refinement strategy for 3-D simplicial meshes. *Computers & Mathematics with Applications*, 66(7):1272–1283, 2013. doi:10.1016/j.camwa.2013.07.026.

[118] Ulrich Trottenberg, Cornelius W. Oosterlee, and Anton Schuller. *Multigrid*. Academic Press, 2001.

[119] Xuemin Tu and Bin Wang. A BDDC algorithm for second-order elliptic problems with hybridizable discontinuous Galerkin discretizations. *Electronic Transactions on Numerical Analysis*, 45, 2016.

[120] Petr Vaněk. Acceleration of convergence of a two-level algorithm by smoothing transfer operators. *Applications of Mathematics*, 37(4):265–274, 1992. doi:10.21136/AM.1992.104509.

[121] Petr Vaněk. Fast multigrid solver. *Applications of Mathematics*, 40(1):1–20, 1995. doi:10.21136/AM.1995.134274.

[122] Roman Wienands and Cornelis W. Oosterlee. On Three-Grid Fourier Analysis for Multigrid. *SIAM Journal on Scientific Computing*, 23(2):651–671, 2001. doi:10.1137/S106482750037367X.

[123] Tim. Wildey, Sriramkrishnan. Muralikrishnan, and Tan. Bui-Thanh. Unified Geometric Multigrid Algorithm for Hybridized High-Order Finite Element Methods. *SIAM Journal on Scientific Computing*, 41(5):S172–S195, 2019. doi:10.1137/18M1193505.

[124] O. C. Zienkiewicz. Displacement and equilibrium models in the finite element method by B. Fraeijs de Veubeke, Chapter 9, Pages 145–197 of Stress Analysis, Edited by O. C. Zienkiewicz and G. S. Holister, Published by John Wiley & Sons, 1965. *International Journal for Numerical Methods in Engineering*, 52(3):287–342, 2001. doi:10.1002/nme.339.

[125] D. Zint, R. Grosso, V. Aizinger, and H. Köstler. Generation of Block Structured Grids on Complex Domains for High Performance Simulation. *Computational Mathematics and Mathematical Physics*, 59(12):2108–2123, 2019. doi:10.1134/S0965542519120182.

Contributions

International, peer-reviewed journal papers

- D. A. Di Pietro, F. Hülsemann, P. Matalon, P. Mycek, U. Rüde.
  Algebraic multigrid preconditioner for statically condensed systems arising from lowest-order hybrid discretizations.
  Submitted. Preprint: https://hal.archives-ouvertes.fr/hal-03272468
- D. A. Di Pietro, F. Hülsemann, P. Matalon, P. Mycek, U. Rüde, D. Ruiz.
  Towards robust, fast solutions of elliptic equations on complex domains through HHO discretizations and non-nested multigrid methods.
  International Journal for Numerical Methods in Engineering, 122(22):6576-6595, 2021.
  DOI: 10.1002/nme.6803
  Open access: https://hal.archives-ouvertes.fr/hal-03163476
- D. A. Di Pietro, F. Hülsemann, P. Matalon, P. Mycek, U. Rüde, D. Ruiz.
  An h-multigrid method for Hybrid High-Order discretizations.
  SIAM Journal on Scientific Computing, 2021 (Copper Mountain Special Section 2020).
  DOI: 10.1137/20M1342471
  Open access: https://hal.archives-ouvertes.fr/hal-02434411

Talks at international conferences

- *Algebraic multigrid preconditioner for statically condensed systems arising from lowest-order hybrid discretizations.* (+ paper for the student paper competition)
  Copper Mountain Conference on Multigrid Methods 2021 (online)
- *Toward robust, fast solutions of elliptic PDEs through HHO discretizations and multigrid solvers.*
  Sparse Days 2020 (online)
- *An h-multigrid method for Hybrid High-Order discretizations.*
  Copper Mountain Conference on Iterative Methods 2020 (cancelled due to Covid-19, participation to the student paper competition)

Poster

- *Fast solvers for Hybrid High-Order discretizations.*
  CERFACS PhD day 2020
  Open access: https://hal.archives-ouvertes.fr/hal-03228427

Symbols

d	Space dimension: 1, 2 or 3
$\mathbb{R}^{d\times d}_{\text{sym}}$	Space of symmetric real matrices of size $d \times d$
Ω	Domain of study $\subset \mathbb{R}^d$
$\partial\Omega$	Domain boundary
∇	Gradient operator $\Omega \to \mathbb{R}^d$
$\nabla\cdot$	Divergence operator $\mathbb{R}^d \to \mathbb{R}$
$\mathbf{v}\cdot\mathbf{w}$	Euclidean inner product of $\mathbf{v}$ and $\mathbf{w} \in \mathbb{R}^d$
f	Source function $\Omega \to \mathbb{R}$
$\mathbf{K}$	Diffusion tensor $\Omega \to \mathbb{R}^{d\times d}_{\text{sym}}$
h	Mesh size
$\mathcal{T}_h$	Set of mesh elements
$\mathcal{F}_h$	Set of mesh faces
$\mathcal{F}_h^{\text{I}}$	Set of internal faces
$\mathcal{F}_h^{\text{B}}$	Set of boundary faces
T	Generic element $\in \mathcal{T}_h$
F	Generic face $\in \mathcal{F}_h$
$\mathcal{F}_T$	Set of faces of element T
$\mathcal{T}_F$	Set of elements that F is a face of
$\mathbf{n}_{TF}$	Unit vector normal to F pointing out of T
$\mathbf{K}_T$	Restriction of $\mathbf{K}$ to the element T
K_{TF}	$\mathbf{K}_T\mathbf{n}_{TF}\cdot\mathbf{n}_{TF}$
$\mathcal{C}^\infty(X)$	Functional space of inifinitely differentiable functions defined on $X \subset \mathbb{R}^d$
$L^2(X)$	Functional space of square-integrable functions defined on $X \subset \mathbb{R}^d$
$(v,w)_X$	$L^2(X)$-inner product: $\int_X vw$ if $v,w \in L^2(X)$, or $\int_X v\cdot w$ if $v,w \in [L^2(X)]^d$
$H^1(X)$	Sobolev space of order 1
$H^1_0(\Omega)$	Subspace of $H^1(\Omega)$ with vanishing trace on the boundary $\partial\Omega$
k	Parameter of the HHO method defining the polynomial degree on the faces
$\mathbb{P}^k(T)$	Space of d-variate polynomials local to element T of total degree at most k
$\mathbb{P}^k(F)$	Space of $(d-1)$-variate polynomials local to face F of total degree at most k
π_X^k	L^2-orthogonal projector onto the polynomial space $\mathbb{P}^k(X)$, X being a cell or a face
$\widetilde{\pi}_T^{k+1}$	Elliptic projector onto the polynomial space $\mathbb{P}^{k+1}(T)$

$\underline{U}_T^k$	Local hybrid space of DoFs: $\mathbb{P}^k(T) \times \left(\bigtimes_{F\in\mathcal{F}_T} \mathbb{P}^k(F)\right)$
$\underline{v}_T$	Local hybrid DoFs: $\underline{v}_T := (v_T, (v_F)_{F\in\mathcal{F}_T})$
$\underline{U}_h^k$	Global hybrid space of DoFs: $\left(\bigtimes_{T\in\mathcal{T}_h} \mathbb{P}^k(T)\right) \times \left(\bigtimes_{F\in\mathcal{F}_h} \mathbb{P}^k(F)\right)$
$\underline{v}_h$	Global hybrid DoFs: $\underline{v}_h := ((v_T)_{T\in\mathcal{T}_h}, (v_F)_{F\in\mathcal{F}_h})$
$U_{\mathcal{T}_h}^k$	Global space of cell-DoFs: $\bigtimes_{T\in\mathcal{T}_h} \mathbb{P}^k(T)$
$v_{\mathcal{T}_h}$	Global cell-DoFs: $v_{\mathcal{T}_h} := (v_T)_{T\in\mathcal{T}_h}$
$U_{\mathcal{F}_h}^k$	Global space of face-DoFs: $\bigtimes_{F\in\mathcal{F}_h} \mathbb{P}^k(F)$
$v_{\mathcal{F}_h}$	Global face-DoFs: $v_{\mathcal{F}_h} := (v_F)_{F\in\mathcal{F}_h}$

Acronyms

AGMG	Name of a specific AMG software: AGgregation-based AMG
AMG	Algebraic Multigrid
ANR	Agence Nationale de la Recherche (French National Research Agency)
BDM	Brezzi-Douglas-Marini
C-AMG	Condensed Algebraic Multigrid (as opposed to U-AMG)
CFD	Computational Fluid Dynamics
CG	Conjugate Gradient
CGAL	Name of geometric software library
DG	Discontinuous Galerkin
DoF	Degree of Freedom
DtN	Dirichlet-to-Neumann
EDF	Electricité de France
FCG	Flexible Conjugate Gradient
FEM	Finite Element Method
FV	Finite Volumes
HDG	Hybridizable Discontinuous Galerkin
HHO	Hybrid High-Order
IGA	Isogeometric Analysis
MFMFE	Multipoint Flux Mixed Finite Elements
PDE	Partial Differential Equation
RT	Raviart-Thomas
SOR	Successive Over-Relaxation
U-AMG	Uncondensed Algebraic Multigrid

List of Figures

List of Algorithms

Summary

The present thesis focuses on fast numerical solutions of partial differential equations discretized by the recent Hybrid High-Order (HHO) method. The arising linear systems are solved by the means of novel, efficient multigrid methods. This research work is funded by the project Fast4HHO[1] of the French National Research Agency, granted to Electricité de France (EDF).

Context and motivation

HHO discretizations [43] have gained growing interest in recent years. Amongst their key features, we can list the support of general polytopal meshes and of arbitrary approximation orders, as well as their optimal orders of convergence. Another built-in and defining feature of the HHO methods is the use, in the formulation of the bilinear form, of a higher-order potential reconstruction operator, which allows the gain of one additional order of approximation compared to similar hybrid methods, like Hybridized Discontinuous Galerkin (HDG) [36, 92]. Finally, the capability of HHO methods to adapt their design to the underlying physics, via problem-dependent local formulations, allows for more robust solutions with respect to the problem. Up to this day, HHO methods have been successfully derived for a large variety of problems in fluid dynamics (heterogeneous anisotropic diffusion [48], incompressible Navier-Stokes [21], creeping flows of non-Newtonian fluids [25]) and structural mechanics (linear and nonlinear elasticity [23, 47] and poroelasticity [18, 24]). Now that the method has gained sufficient maturity, its adoption for industrial applications hangs on the availability of efficient linear solvers. The goal of this Ph.D thesis is to bridge this gap. More precisely, this dissertation focuses on relevant applications to EDF in Computational Fluid Dynamics (CFD), in particular Darcy flow in porous media and incompressible fluid mechanics.

HHO methods hinge on degrees of freedom (DoFs) located inside elements and on faces, which can be globally viewed as broken polynomials respectively on the mesh and its skeleton. We exclusively focus on cases where the element-defined DoFs are only locally coupled. As such, they can be expressed, element by element, in function of the DoFs on the faces, and subsequently eliminated from the global HHO linear system. This gives rise to a Schur complement of smaller size where only face unknowns remain. This process is known as *static condensation* in the mechanical literature, and the resulting system as a *statically condensed* system, or *trace* system, in reference to the mesh skeleton as the support for the set of globally coupled unknowns. The solution of the trace system, yielding the face unknowns, remains the costliest operation, after which the values of the element unknowns can be inexpensively recovered by solving small, independent linear systems.

[1] under contract ANR-17-CE23-0019

As a consequence, the practical usefulness of HHO discretizations in an industrial context, where large problems have to be solved, depends on the existence of efficient linear solvers for the *condensed* system. Especially, the present research work is motivated by the aim to provide a solver for the free, open-source CFD software *code_saturne*[1] [7], developed and released by EDF.

In this Ph.D thesis, we focus on scalar, second order, elliptic equations, whose HHO discretizations give rise to trace matrices that are sparse, symmetric and positive-definite. Specifically, we aim at solving large systems of this type by means of a multigrid method [31, 118]. The main difficulty in the design of a geometric multigrid algorithm for a trace system resides in the location of the DoFs associated to the set of unknowns that remain after static condensation, namely, the face unknowns. Supported by the mesh skeleton, the broken polynomials defined by these DoFs are not suited for standard intergrid transfer operators, applicable to *element*-defined functions. Multigrid algorithms designed for Discontinuous Galerkin (DG) discretizations are therefore excluded, hence the need for novel, *skeleton-based* multigrid methods.

Thesis outline and contributions

Introduction

In Chapter 1, we first justify the need for novel discretizations such as HHO to tackle the open issues of *complex geometries* and *non-smooth* solutions in CFD. We next state a list of criteria, relative to structure and performance, that a linear solver should exhibit to be considered as an adequate answer to the problem at hand. Besides the proper formalization of our research goals, this exercise allows us to discuss existing solutions in light of these criteria, and therefore justify the need for new solvers and identify the gaps filled by the contributions of this thesis. Then follows a thorough state-of-the-art of existing solvers, especially multigrid, targeting trace systems. Finally, a detailed summary of our contributions is presented.

Model problem and HHO discretization

Chapter 2 is dedicated to the application of the HHO method on a model problem for scalar second-order elliptic equations, namely, the diffusion problem including a uniformly elliptic permeability tensor. We especially introduce the *high-order potential reconstruction operator*, which is the main ingredient in the definition of the discrete bilinear form. This operator, based only on an integration by parts formula, is locally defined. It allows, from a polynomial in the cell and polynomials of same degree on the faces, to reconstruct a polynomial *one degree higher* in the cell. This feature is advantageously employed in our geometric multigrid method to enhance its overall performance.

A geometric h-multigrid method

Chapter 3 is devoted to the first original contribution of this Ph.D thesis, namely the development of a novel, geometric h-multigrid algorithm (i) based on approximation spaces supported by the

[1] www.code-saturne.org

mesh skeleton at every level, (ii) targeting HHO discretizations by making use of the underlying high-order potential reconstruction, (iii) natively handling higher orders (as opposed to, e.g., putting a p-multigrid on top of an h- one). The method relies on the design of a special prolongation operator that includes the construction of an intermediary state between the coarse skeletal function and its prolongation onto the fine skeleton. Precisely, a *cell-defined* potential is reconstructed on the coarse mesh, which allows, via a trace operator, a subsequent definition on the fine skeleton.

The cell reconstruction is the core of our method and what makes it original. It works locally, and decomposes into two steps. Firstly, a coarse cell-defined polynomial of degree k is recovered from the face-defined polynomials of degree k through the decondensation of the cell unknowns. Secondly, the higher-order reconstruction operator is applied to both cell and face unknowns in order to gain one degree of approximation in the cell. Given that the reconstructed polynomial is of degree $k+1$, recovering the original polynomial degree k on the fine faces implies that the trace operation must also lower the degree. To do so, the trace comes with a subsequent L^2-orthogonal projection onto the polynomial space of lower order k. Moreover, on the fine faces at the boundary of coarse elements, due to the discontinuous setting, the trace actually consists in taking the weighted average of the traces computed on each side. The weights take the diffusion coefficient into account to ensure robustness to discontinuities.

The numerical tests include homogeneous and heterogeneous isotropic problems in 2D and 3D domains, discretized by structured and unstructured meshes. With structured meshes on simple domains, whether with Cartesian or simplicial elements, the multigrid method, directly used as a solver, exhibits the following properties: (i) convergence in a limited number of iterations, seemingly independently of the mesh size; (ii) controlled computational cost through the rediscretization of the operator at the coarse levels and the use of standard smoothers (namely, block Gauss-Seidel or Jacobi); (iii) robustness to discontinuities of the diffusion coefficient, whose magnitude does not alter the convergence rate; (iv) robustness to higher orders, for which the solver exhibits the same properties.

However, on complex domains requiring highly unstructured meshes, optimal convergence is not achieved in general. The reason is twofold: (i) optimal convergence relies on the *faces* being coarsened between levels (not only the elements!); (ii) numerical experiments have shown the high sensitivity of the multigrid method to the mesh quality, i.e. to the presence of elements with bad aspect ratio. Optimality then also requires a hierarchy of high-quality meshes. Combined, these demands raise the issue of how to build the mesh hierarchy. Indeed, multigrid hierarchies are commonly constructed by successive refinements of an initial coarse mesh. If refinement ensures face coarsening between every level from the fine mesh to the coarse one, it also often has the nasty habit to affect mesh quality, especially in 3D. Conversely, starting from a good quality fine mesh, there is no obvious method allowing to construct a nested coarse mesh while also enforcing face coarsening. Non-nestedness is indeed the path we choose to follow in our second contribution to overcome the limitations of our multigrid method and successfully manage untructured 3D cases.

Extension to non-nested meshes

Chapter 4 is dedicated to the adaptation of the nested version of our algorithm to non-nested mesh hierarchies and its efficient implementation for practical use. Compliance to non-nested settings is performed by inserting an additional step in the definition of the prolongation operator: starting with polynomials lying on the coarse faces, the nested version begins with the reconstruction of a broken *element*-defined polynomial on the coarse mesh. This step is unchanged. We then propose to orthogonally project in L^2-norm this coarse broken polynomial onto the non-nested fine mesh. Finally, the end of the process also follows the nested version: the trace of the result is computed on the fine faces.

The numerical evaluation of this L^2-orthogonal projection operator hinges on the projection of the local coarse basis functions onto the fine bases, i.e. on the computation of the L^2-inner products of the coarse and fine basis functions over the fine elements. As a direct consequence, the local definition of the functional bases makes the intersections of coarse and fine elements the respective integration supports to these inner products. However, computing the geometric intersections between coarse and fine elements can be computationally prohibitive. So, instead of this exact computation, we propose the implementation of an approximate operator that does not require the explicit computation of intersections. It is based on the subdivision of the fine elements, by adopting the simplifying hypothesis that each sub-element is fully included in the coarse element that contains its barycenter. We evaluate the accuracy of this approximation through comparative experiments with the exact operator, in which we assess the convergence of our multigrid method where the non-nested meshes are obtained by independent retriangulation of the domain at each level. These tests demonstrate the sufficient accuracy of the approximation for moderate polynomial degrees in 3D, as well as the substential gain in setup time that the technique offers by avoiding the computation of geometric intersections. In particular, we demonstrate the optimal convergence of our non-nested multigrid algorithm on an unstructured 3D test case that the nested version failed to solve.

In practice, building a high-quality mesh for a real, industrial case study can be an arduous task, which may occupy a meshing engineer for several months. Requiring multiple high-quality meshes of the same geometry at different granularities in order to feed a multigrid solver is then not always conceivable. From the user's standpoint, providing the solver with the sole fine mesh is a preferable option. This is why this chapter also includes the abstract definition of a coarsening strategy in order to build, from a given fine mesh, a hierarchy of non-nested coarse meshes in which faces are coarsened. In particular, the method is based on element agglomeration, to which we add a step of face collapsing at the interfaces between agglomerates. We provide explicit details about our implementation in 2D, among which we especially explain how the the approximate L^2-orthogonal projection can be made exact through a clever way of subtriangulating the fine elements. The non-nested multigrid method yielded by this coarsening strategy is finally evaluated on the simplified geometry of a real, industrial test case provided by EDF, which also results in an asymptotically optimal behaviour.

Algebraic multigrid

The geometric multigrid algorithm and its non-nested extension that we have devised provide a first option for the solution of HHO systems. In Chapter 5, we develop another approach, in the form of an Algebraic Multigrid method (AMG).

Usual AMG solvers designed for low-order finite element or finite difference methods deduce mesh information under the assumption that each row in the matrix corresponds to an unknown related to a DoF located at a *mesh node* or *elements*. Thus, the mesh connectivity graph can be reconstructed algebraically, and coarsening strategies mimicking geometric algorithms can then be performed in order to build the coarse levels. Especially focusing on aggregation-based methods, nodes are being aggregated in order to give rise to coarse DoFs. However, in our hybrid setting at the lowest order, the unknowns of the system are actually linked to faces, i.e. neither nodes nor elements. Consequently, at first glance it might seem peculiar, from a geometrical point of view, to apply the above approach in this context. Indeed, aggregation-based coarsening can then be interpreted as aggregating *faces*. Although it may give natural results for neighbouring faces, especially if they are close to being colinear, it sometimes aggregate faces that do not even touch. In this case, it is difficult to perceive a geometrical sense in this aggregation. Nonetheless, numerical tests with a standard aggregation-based AMG method show that the approach still works well, which can be geometrically justified by forgetting about the DoFs being actually face-defined and considering them as mere node values located at the center of the faces. That being said, one can legitimately wonder if a coarsening strategy making geometrical sense in light of the actual significance of the DoFs as face-defined values could not yield even better results.

Restricting our scope to *lowest-order* hybrid methods (not only HHO), the idea at the origin of this work is the algebraic reconstruction of the mesh information based, no longer on the condensed matrix, but on the uncondensed one. Indeed, like traditional AMG methods, we retrieve geometric information on the coupling of the DoFs from algebraic data. However, as the condensed matrix only gives information on the faces, we use the *uncondensed* version to reconstruct the connectivity graph between elements and faces. Once the so-called algebraic mesh is retrieved, especially the neighbouring information between elements, an *element*-based aggregation method can be set up in order to mimic the behaviour of a geometric coarsening or semi-coarsening strategy. Keeping in mind that, in our hybrid setting, *faces* must be coarsened between levels, we complement the element aggregation with the face collapsing technique devised in Chapter 4. The method is used in conjonction with the so-called K-cycle to precondition an outer Krylov method. The technical choices made in this work are borrowed from AGMG [95] (pairwise aggregation, strong negative coupling criterion, K-cycle...) in order to establish a proper comparison with a standard AMG solver that works only on the condensed system.

Our method is applied to the lowest order HHO discretizations of 2D and 3D diffusion problems. The test spectrum includes homogeneous, heterogeneous, isotropic and anisotropic problems on structured Cartesian and unstructured simplicial meshes. The methodology adopted compares our novel method to a standard aggregation-based AMG that views DoFs as nodes and implements a node-defined coarsening strategy from the condensed system. The aggregation criteria, cycle, smoothers, as well as every other technical choices are identical for both solvers

to establish a comparison. We report equivalent performances in isotropic and in unstructured cases. The added value of the new algorithm actually appears in anisotropic problems with Cartesian meshes, where the solver exhibits an enhanced robustness. Although this very specific, trivial test case might seem restrictive, this feature can actually be exploited in a larger range of cases. Namely, the method can offer substantial added value for solving problems comprising both isotropic and anisotropic regions, providing that the anisotropic ones are discretized by Cartesian elements oriented in the direction of anisotropy. The solver, in this case, uses the flexibility of AMG to handle unstructured meshes on isotropic regions while exploiting the special element shapes on anisotropic ones.

Scientific communications

The contributions of this Ph.D thesis have been made available to the scientific community through journal papers, open-access preprints, and talks at international conferences. References and download links are gathered page 103.

Résumé

Cette thèse a pour objet la résolution rapide d'équations aux dérivées partielles discrétisées avec la méthode Hybrid High-Order (HHO), ou méthode hybride d'ordre élevé. Les systèmes linéaires obtenus sont résolus au moyen d'efficaces nouvelles méthodes multigrilles. Ce travail de recherche est financé par le projet ANR Fast4HHO[1], sous la gestion d'EDF.

Contexte et motivation

Les discrétisations HHO [43] suscitent un intérêt croissant depuis quelques années. Parmi leurs caractéristiques principales, on peut citer le support des maillages polyédriques généraux et des ordres polynomiaux arbitrairement élevés, ainsi que l'optimalité de leur convergence. Un autre ingrédient, sur lequel repose la construction de ces méthodes, est l'utilisation, dans la formulation de la forme bilinéaire, d'un opérateur de reconstruction à l'ordre élevé, ce qui permet le gain d'un degré d'approximation par rapport à des méthodes hybrides similaires, telles que les méthodes de Garlerkin discontinues hybridisées (HDG) [36, 92]. Pour finir, la capacité des méthodes HHO à adapter leur formulation à la physique du problème rendent leurs approximations plus robustes. Jusqu'à aujourd'hui, les méthodes HHO ont été explicitées pour une grande variété de problèmes de la mécanique des fluides (diffusion hétérogène et anisotropique [48], équations de Navier-Stokes incompressibles [21], écoulements visqueux de fluides non newtoniens [25]) et de la mécanique des structures (élasticité et poroélsticité linéaire et non linéaire [18, 23, 24, 47]). Maintenant que la méthode est suffisamment mature, son adoption par l'industrie dépend de l'existence de solveurs linéaires efficaces. L'objectif de cette thèse est de remplir ce vide. Plus précisément, cette dissertation se concentre sur les problèmes de mécanique des fluides numérique d'intérêt pour EDF, à savoir les écoulements darcéens en milieu poreux et la mécanique des fluides incompressibles.

Les degrés de liberté (DDLs) des méthodes HHO sont situés dans les éléments sur les faces. Ils peuvent être interprétés globalement comme des polynômes brisés sur le maillage et son squelette. Nous nous concentrons exclusivement sur les problèmes où les DDLs d'éléments sont uniquement couplés localement. Dans ce cas, ils peuvent être exprimés, élément par élément, en fonction des DDLs de faces, et peuvent donc être éliminés du système linéaire HHO global. Cela engendre un complément de Schur de taille réduite au sein duquel ne restent que les inconnues de faces. Ce processus est connu dans la littérature mécanique sous le nom de *condensation statique*, et le système résultat sous le nom de système *condensé statiquement*, ou système *aux traces*, en référence au fait que le squelette du maillage correspond au support des inconnues couplées globalement. Résoudre ce système aux traces, donnant les valeurs des inconnues de

[1] contrat ANR-17-CE23-0019

faces, reste l'opération la plus coûteuse, après quoi les valeurs des inconnues de cellules peuvent être retrouvées en résolvant à faible coût des petits systèmes linéaires indépendants.

Conséquemment, l'utilité pratique des discrétisations HHO dans un contexte industriel, dans lequel il faut résoudre des problèmes de grande taille, dépend de l'existence de solveurs linéaires efficaces pour le système *condensé*. En outre, ce travail de recherche trouve sa motivation dans l'objectif de fournir un solveur au logiciel de mécanique des fluides numérique open-source et gratuit *code_saturne*[1] [7], développé et distribué par EDF.

Dans cette thèse, nous nous concentrons sur les équations scalaires elliptiques du second ordre, dont la discrétisation HHO engendre des matrices aux traces creuses, symétriques et définies positives. Notre objectif est de résoudre de tels systèmes de grande taille au moyen de méthodes multigrilles [31, 118]. La difficulté principale dans l'élaboration d'un algorithme multigrille géométrique pour un système aux traces tient à la localisation des DDLs associés aux inconnues qui restent après la condensation statique, c'est-à-dire les inconnues de faces. Supportés par le squelette du maillage, les polynômes brisés définis par ces DDLs ne sont pas adaptés aux opérateurs de transfert intergrilles standards, qui s'appliquent sur des fonctions définies sur les *éléments*. Les algorithmes multigrilles construits pour les discrétisations de Galerkin discontinues (DG) sont donc exclus, d'où le besoin de nouvelles méthodes multigrilles construites *sur le squelette*.

Plan de la thèse et contributions

Introduction

Dans le chapitre 1, nous commençons par justifier le besoin de nouvelles discrétisations telles qu'HHO pour traiter les problèmes encore ouverts des *géométries complexes* et des solutions *non régulières* en mécanique des fluides numérique. Ensuite, nous définissons une liste de critères structurels et de performance qu'un solveur linéaire doit remplir afin d'être considéré comme une réponse adéquate au problème à résoudre. A part la formalisation de nos objectifs de recherche, cet exercice nous permet de discuter des solutions existantes au regard de ces critères, et par conséquent de justifier le besoin de nouveaux solveurs et d'identifier les lacunes comblées par les contributions de cette thèse. Il s'ensuit un état de l'art minutieux des solveurs existants, en particulier multigrilles, qui ciblent les systèmes aux traces. Enfin, un résumé détaillé de nos contributions est présenté.

Problème modèle et discrétisation HHO

Le chapitre 2 est dédié à l'application de la méthode HHO à un problème modèle pour les équations scalaires elliptiques du second ordre. En l'occurrence, le problème de la diffusion incluant un tenseur de perméabilité. On introduit en particulier l'*opérateur de reconstruction de potentiel d'ordre supérieur*, l'ingrédient principal dans la définition de la forme bilinéaire. Cet opérateur, uniquement construit à partir d'une formule d'intégration par parties, se définit localement. Il permet, à partir d'un polynôme défini dans la cellule et de polynômes de même degré sur les faces, de reconstruire un polynôme *d'un degré supérieur* dans la cellule. Cette

[1] www.code-saturne.org

reconstruction est exploitée dans notre multigrille géométrique pour améliorer ses performances générales.

Une méthode h-multigrille géométrique

Le chapitre 3 est dédié à la première contribution originale de cette thèse de doctorat : le développement d'un nouvel algorithm h-multigrille géométrique (i) basé sur des espaces d'approximation supportés par le squelette du maillage à tous les niveaux, (ii) ciblant les discrétisations HHO grâce à l'utilisation de la reconstruction de potentiel d'ordre supérieur, (iii) prenant en compte de façon native les ordres élevés (contrairement à, par exemple, l'ajout d'un multigrille en p au-dessus de celui en h). La méthode est basée sur le design d'un opérateur de prolongation particulier, qui inclut la construction d'un état intermédiaire entre celui de fonction sur le squelette grossier et celui de fonction sur le squelette fin. Plus précisément, un potentiel est reconstruit *sur les cellules* du maillage coarse permettant, grâce à un opérateur de trace, d'en définir un sur le squelette fin.

La reconstruction sur la cellule est le cœur de la méthode et ce qui la rend originale. Elle fonctionne localement, et se décompose en deux temps. Premièrement, un polynôme de degré k, défini sur les cellules grossières, est retrouvé grâce aux polynômes de degré k sur les faces. Deuxièmement, l'opérateur de reconstruction d'ordre supérieur est appliqué conjointement au polynôme de cellule et aux polynômes de faces afin de gagner un degré d'approximation dans la cellule. Etant donné que le polynôme ainsi reconstruit est de degré $k+1$, retrouver le degré polynomial k de départ implique que l'opération de trace fasse également baisser le degré. Pour ce faire, la trace s'accompagne d'une projection L^2-orthogonale sur l'espace polynomial de degré inférieur k. Par ailleurs, sur les faces fines à la frontière des éléments grossiers, à cause des discontinuités, la trace est choisie comme la moyenne pondérée des traces calculées de chaque côté. Les poids prennent en compte le coefficient de diffusion afin d'assurer un comportement robuste aux discontinuités.

Les tests numériques contiennent des problèmes isotropiques homogènes et hétérogènes sur des domaines 2D et 3D, discrétisés avec des maillages structurés et non structurés. Avec des maillages structurés sur domaines simples, qu'ils soient faits d'éléments cartésiens ou de simplexes, la méthode multigrille, directement utilisée comme solveur, présente les propriétés suivantes: (i) convergence en un nombre raisonnable d'itérations; (ii) coût de calcul maîtrisé grâce à la rediscrétisation de l'opérateur aux niveaux grossiers et à l'utilisation de lisseurs standards (Gauss-Seidel ou Jacobi par blocs); (iii) robustesse aux discontinuités du coefficient de diffusion, dont l'ordre de grandeur n'affecte pas le taux de convergence; (iv) robustesse aux ordres élevés, pour lesquels le solveur présente les mêmes propriétés.

Cependant, sur les domaines complexes qui requièrent des maillages fortement non structurés, la convergence optimale n'est généralement pas atteinte. Cela tient à deux observations: (i) l'optimalité requiert que les *faces* (et non uniquement les éléments) aient également une représentation grossière; (ii) les expériences numériques ont montré la sensibilité élevée de la méthode multigrille à la qualité du maillage, c'est-à-dire à la présence d'éléments présentant un mauvais rapport de forme. L'optimalité requiert alors une hiérarchie de maillage de haute qualité. Combinées, ces demandes pose le problème de la façon dont la hiérarchie de maillages

est obtenue. En effet, pour le multigrille, elles sont généralement construites par raffinements successifs d'un maillage grossier initial. Si le raffinement garantit une représentation grossière des faces entre chaque niveau, il a souvent la mauvaise habitude de dégrader la qualité du maillage de départ, particulièrement en 3D. A l'inverse, en démarrant d'un maillage fin de bonne qualité, il n'y a pas de méthode évidente de construction d'un maillage grossier imbriqué tout en assurant une représentation grossière des faces. Le passage aux maillages non imbriqués est en effet le chemin que nous suivons dans notre seconde contribution afin de dépasser les limitations de notre multigrille et de réussir à gérer les cas 3D non structurés.

Extension aux maillages non emboîtés

Le chapitre 4 est dévolu à l'adaptation de la version imbriquée de notre algorithme aux maillages non imbriqués, ainsi qu'à son implémentation efficace en vue d'une utilisation pratique.

La gestion de la non imbrication se fait par l'insertion d'une étape de plus dans la définition de l'opérateur de prolongation : démarrant des polynômes sur les faces grossières, la version emboîtée commence par la reconstruction d'un polynôme brisé sur les *éléments* du maillage grossier. Cette étape reste inchangée. Nous proposons ensuite de projeter orthogonalement, en norme L^2, ce polynôme brisé sur le maillage fin non emboîté. Pour finir, la fin du processus suit également la version emboîtée : la trace du résultat est calculée sur les faces fines.

L'évaluation numérique de cette projection L^2-orthogonale repose sur la projection des fonctions de base locales grossières sur les bases fines, c'est-à-dire sur le calcul des produits scalaires L^2 des fonctions de base grossières et fines sur les éléments fins. Conséquemment, la définition locale des fonctions de base définit les intersections des éléments grossiers et fins comme les supports d'intégration respectifs de ces produits scalaires. Toutefois, calculer les intersections géométriques entre les éléments grossiers et fins peut présenter un coût de calcul prohibitif. C'est pourquoi, à la place de ce calcul exact, nous proposons l'implémentation d'un opérateur approximatif qui ne requiert pas le calcul explicite des intersections. Celui-ci est basé sur la subdivision des éléments fins, en adoptant l'hypothèse simplificatrice selon laquelle chaque sous-élément est entièrement inclus dans l'élément grossier contenant son barycentre. Nous estimons la précision de cette approximation par des expériences comparatives avec l'opérateur exact, dans lesquelles est évaluée la convergence de notre méthode multigrille où les maillages non imbriqués sont obtenus par une retriangulation indépendante à tous les niveaux. Ces tests démontrent la précision suffisante de l'approximation pour les ordres polynomiaux modérés en 3D, ainsi que le gain substantiel en temps de préparation que la technique offre en s'affranchissant du calcul des intersections. En outre, on y démontre la convergence optimale de notre algorithme non imbriqué sur un cas de test 3D non structuré que la version imbriquée ne parvenait pas à résoudre.

En pratique, construire un maillage de haute qualité pour une étude industrielle réelle peut s'avérer une tâche ardue, capable d'occuper un ingénieur de maillages pour plusieurs mois. Demander plusieurs maillages de haute qualité de la même géométrie avec des granularités différentes afin d'alimenter un solveur multigrille n'est donc pas toujours envisageable. Du point de vue de l'utilisateur, fournir au solveur le seul maillage fin est une meilleure option. C'est pourquoi ce chapitre inclut également la définition abstraite d'une stratégie de coarsening afin de contruire,

pour un maillage fin donné, une hiérarchie de maillages non emboîtés avec représentation grossière des faces. En outre, la méthode est basée sur l'agglomération d'éléments, à laquelle vient s'ajouter une étape de simplification de faces aux interfaces entre les agglomérats. Notre implémentation en 2D est explicitée, où nous expliquons en particulier comment l'approximation de la projection L^2 peut être rendue exacte grâce à une façon intelligente de sous-trianguler les éléments fins. Pour finir, la méthode multigrille non imbriquée qui en résulte est évaluée sur la simplification d'une géométrie industrielle réelle utilisée par EDF. Elle manifeste également un comportement asymptotique optimal.

Multigrille algébrique

Le multigrille géométrique et son extension aux maillages non imbriqués fournissent une première option pour résoudre les systèmes HHO. Dans le chapitre 5, nous développons une autre approche, qui prend la forme d'un multigrille algébrique (AMG).

Les solveurs AMG habituels, faits pour les éléments finis ou différences finies d'ordre bas, déduisent des informations sur le maillage sous l'hypothèse que chaque ligne de la matrice correspond à une inconnue liée à un DDL localisé sur un *nœud de maillage* ou un *élément*. Ainsi, le graphe de connectivité du maillage peut être reconstruit algébriquement, et des stratégies de coarsening copiant des algorithmes géométriques peuvent donc être exécutés pour construire les niveaux grossiers. En se focalisant sur les méthodes d'agrégation, les nœuds sont agrégés entre eux afin de définir les DDLs grossiers. Cependant, dans notre configuration hybride d'ordre bas, les inconnues du système sont liées aux faces, donc ni aux nœuds ni aux éléments. En conséquence, il peut sembler étrange à première vue, d'un point de vue géométrique, d'appliquer l'approche précédente dans ce contexte. En effet, l'agrégation peut alors être interprétée comme une agrégation de *faces*. Bien que cela puisse donner des résultats naturels pour des faces voisines, surtout si elles sont presque coplanaires, il arrive aussi parfois que le processus agrège des faces qui ne sont même pas en contact. Dans ce cas, il est difficile de percevoir un sens géométrique à cette agrégation. Néanmoins, des tests numériques avec une méthode standard d'AMG basée sur l'agrégation démontre que l'approche fonctionne tout de même correctement, ce qui peut être justifié de façon géométrique en oubliant que les DDLs sont localisés aux faces pour les considérer comme de simple valeurs de nœuds (qui s'avèrent être situés aux centres des faces). Ceci étant dit, on peut légitimement se demander si une stratégie de coarsening ayant un sens géométrique au regard de la véritable nature des DDLs en tant que valeurs aux faces ne pourrait pas donner de meilleurs résultats.

En restreignant notre périmètre aux méthodes hybrides *d'ordre le plus bas* (pas uniquement HHO), l'idée à l'origine de ce travail est la reconstruction algébrique des informations de maillage en se basant, non plus sur la matrice condensée, mais sur la matrice non condensée. En effet, comme les méthodes AMG traditionnelles, on récupère les informations géométriques sur le couplage des DDLs à partir de données algébriques. Néanmoins, comme la matrice condensée ne possède d'informations que sur les faces, on utilise la version non condensée pour reconstruire le graphe de connectivité entre les éléments et les faces. Une fois ce qu'on appelle le *maillage algébrique* retrouvé, en particulier les informations de voisinage entre les éléments, une méthode d'agrégation *des éléments* peut être mise en place pour reproduire le comportement d'une méthode

de coarsening ou semi-coarsening géométrique. En se rappelant que dans notre configuration hybride, les *faces* doivent être représentées de façon plus grossière entre les niveaux, on adjoint à l'agrégation des éléments la technique de simplification de faces définie dans le chapitre 4. La méthode est utilisée en conjonction avec le K-cycle pour préconditionner une méthode de Krylov. Les choix techniques qui ont été faits sont empruntés d'AGMG [95] (agrégation par paires, critère de couplage fortement négatif, etc.) dans le but d'établir une comparaison convenable avec un solveur AMG standard qui travaille uniquement sur le système condensé.

Notre méthode est appliquée aux discrétisations HHO d'ordre $k = 0$ de problèmes de diffusion 2D et 3D. Les tests incluent des problèmes homogènes, hétérogènes, isotropiques et anisotropiques sur des maillages cartésiens structurés et des maillages de simplexes non structurés. La méthodologie adoptée compare notre nouvelle méthode à un AMG standard basé sur l'agrégation qui regarde les DDLs comme des nœuds et implémente une stratégie de coarsening nodale à partir du système condensé. Le critère d'agrégation, le cycle, les lisseurs, ainsi que tout autre choix technique sont identiques pour les deux solveurs afin de pouvoir établir une comparaison. Les tests rendent compte de performances équivalentes dans les cas isotropiques et dans les cas non structurés. La valeur ajoutée du nouvel algorithme est mise en évidence sur les problèmes anisotropiques avec maillages cartésiens, pour lesquels le solveur offre une meilleure robustesse. Bien que ce test soit trivial et très spécifique, ce qui peut sembler restrictif, ce résultat peut en réalité être exploité dans un panel de cas de test plus large. En l'occurrence, la méthode offre une valeur ajoutée substantielle pour la résolution de problèmes comprenant à la fois des régions isotropiques et anisotropiques, sous réserve que les régions anisotropiques soit discrétisées par des éléments cartésiens orientés dans le sens de l'anisotropie. Le solveur, dans ce cas, utilise la flexibilité des AMGs pour gérer les maillages non structurés des régions isotropiques tout en exploitant la forme particulière des éléments dans les régions anisotropiques.

Communications scientifiques

Les contributions de cette thèse de doctorat ont été mises à disposition de la communauté scientifique par le biais d'articles de journaux, de preprints en accès libre, et de présentations dans des conférences internationales. Les références correspondantes et liens de téléchargement sont regroupés page 103.

Zusammenfassung

Kontext und Motivation

HHO-Diskretisierungen [43] haben in den letzten Jahren zunehmend ein breiteres Interesse gefunden. Zu ihren Hauptmerkmalen zählen die Unterstützung allgemeiner Polytopgitter und beliebiger Approximationsordnungen sowie ihre optimale Konvergenzordnung. Ein weiteres inhärentes und definierendes Merkmal der HHO-Methoden ist die Verwendung eines Rekonstruktionsoperators höherer Ordnung für das Potential in der Formulierung der bilinearen Form, so dass die Approximationsordnung im Vergleich zu ähnlichen hybriden Methoden, wie der Hybridized Discontinuous Galerkin Methode (HDG) [36, 92], um eins erhöht wird. Schließlich können HHO-Methoden über problemabhängige lokale Formulierungen an die zugrunde liegende Physik angepasst werden, so dass robustere Lösungsverfahren möglich werden. Bis heute wurden HHO-Methoden erfolgreich für eine Vielzahl von Problemen in der Strömungsdynamik (heterogene anisotrope Diffusion [48], inkompressible Navier-Stokes-Gleichhungen [21], kriechende Strömungen nicht-newtonscher Flüssigkeiten [25]) und Strukturmechanik (lineare und nichtlineare Elastizität [23, 47] und Poroelastizität [18, 24]) entwickelt. Damit hat die Methode inzwischen eine ausreichende Reife erlangt, sodass industrielle Anwendungen von der Verfügbarkeit effizienter linearer Löser abhängt. Das Ziel dieser Dissertation ist es, diese Lücke zu schließen. Insbesondere ist diese Dissertation auf relevante Anwendungen von EDF in der numerischen Strömungsmechanik (CFD) fokussiert und dabei speziell auf Darcy-Strömungen in porösen Medien und die inkompressible Strömungsmechanik.

HHO-Methoden basieren auf Freiheitsgraden (DoFs) innerhalb von Elementen und auf den Interfaceflächen, die global als gebrochene Polynome bzw. auf dem Gitter und seinem Skelett betrachtet werden können. Wir konzentrieren uns ausschließlich auf Fälle, in denen die elementdefinierten Freiheitsgrade nur lokal gekoppelt sind. Als solche können sie, Element für Element, in Funktion der Freiheitsgrade auf den Interfaces ausgedrückt und anschließend aus dem globalen linearen HHO-System eliminiert werden. Dies führt zu einem Schur-Komplement verringerter Größe, bei dem nur noch Interface-Unbekannte übrig bleiben. Dieser Prozess ist in der Mechanikliteratur als *statische Kondensation* bekannt und das resultierende System als *statisch kondensiertes* System oder *Trace*-System, wobei das Gitterskelett der Träger für die Menge der global gekoppelten Unbekannten ist. Die Lösung des Spursystems, die die Unbekannten auf den Interfaces liefert, bleibt die kostspieligste Operation. Die Werte der Unbekannten in den Elementen selbst können durch die Lösung kleiner, unabhängiger linearer Systeme effizient rekonstruiert werden.

Damit hängt die praktische Nutzbarkeit von HHO-Diskretisierungen in einem industriellen Kontext, in dem große Probleme gelöst werden müssen, von der Existenz effizienter linearer

Löser für das *kondensierte* System ab. Die vorliegende Forschungsarbeit ist insbesondere durch das Ziel motiviert, einen Löser für das freie, quelloffenen CFD-Programm *code_saturne*[1] [7] zu realisieren, das von EDF entwickelt und freigegeben wurde.

In dieser Dissertation konzentrieren wir uns auf skalare, elliptische Gleichungen zweiter Ordnung, deren HHO-Diskretisierungen zu dünn besetzten symmetrischen und positiv- definiten Spurmatrizen führen. Konkret geht es darum, große Systeme dieses Typs mit Hilfe eines Mehrgitterverfahrens [31, 118] zu lösen. Die Hauptschwierigkeit bei der Entwicklung eines geometrischen Mehrgitteralgorithmus für ein Trace-System liegt in der geometrischen Lage der Freiheitsgrade und der Unbekannten, die nach der statischen Kondensation übrig bleiben, nämlich der Interface-Unbekannten. Da sie auf dem Gitterskelett liegen, sind die gebrochenen Polynome, die durch diese Freiheitsgrade definiert sind, nicht geeignet um Standard-Mehrgittertransferoperatoren anzuwenden, da diese nur für *elementdefinierte* Funktionen anwendbar sind. Mehrgitteralgorithmen, die für Discontinuous Galerkin (DG)-Diskretisierungen entwickelt wurden, können daher nicht verwendet werden, so dass neuartige, *skelettbasierte* Mehrgitterverfahren benötigt werden.

Gliederung der Dissertation und Beiträge

Einleitung

In Kapitel 1 begründen wir zunächst den Bedarf an neuartigen Diskretisierungen wie HHO, um die ungelösten Probleme *komplexer Geometrien* und *nicht-glatter* Lösungen in CFD zu behandeln. Als nächstes nennen wir eine Liste von Kriterien, die ein linearer Löser aufweisen sollte, damit er als adäquate Antwort auf das vorliegende Problem betrachtet werden kann. Neben der korrekten Formalisierung unserer Forschungsziele können wir damit bestehende Lösungen im Hinblick auf diese Kriterien zu diskutieren und somit den Bedarf an neuen Lösern begründen, sowie die Lücken zu identifizieren, die durch die Beiträge dieser Arbeit geschlossen werden. Es folgt ein gründlicher Überblick über den Stand der Technik bestehender Löser, insbesondere von Mehrgitterverfahren für Trace-Systeme. Schließlich wird eine detaillierte Zusammenfassung unserer Beiträge präsentiert.

Modellproblem und HHO-Diskretisierung

Kapitel 2 widmet sich der Anwendung der HHO-Methode auf ein Modellproblem für skalare elliptischen Gleichungen zweiter Ordnung, nämlich dem Diffusionsproblem mit einem uniform elliptischen Permeabilitätstensor. Wir führen insbesondere den *Potentialrekonstruktionsoperator hoher Ordnung* ein, der der Hauptbestandteil der Definition der diskreten bilinearen Form ist. Dieser Operator, der nur auf einer Formel für partielle Integration basiert, ist lokal definiert. Er erlaubt es, aus einem Polynom in der Zelle und Polynomen gleichen Grades auf den Interfaces ein Polynom höheren Grades in der Zelle zu rekonstruieren. Diese Eigenschaft wird in unserem geometrischen Mehrgitterverfahren vorteilhaft eingesetzt um seine Effizienz zu verbessern.

[1] www.code-saturne.org

Geometrisch h-Mehrgitter-Algorithmus

Kapitel 3 ist dem ersten originären Beitrag dieser Doktorarbeit gewidmet, nämlich der Entwicklung eines neuartigen, geometrischen h-Mehrgitter-Algorithmus, der (i) auf Approximationsräumen basiert, die auf jeder Ebene durch das Gitterskelett unterstützt werden, (ii) auf HHO-Diskretisierungen abzielt, indem er die zugrundeliegende Potentialrekonstruktion hoher Ordnung nutzt, (iii) höhere Ordnungen nativ handhabt (im Gegensatz z.B. zum Aufsetzen eines p-Mehrgitterverfahrens auf ein h-Mehrgitterverfahren). Die Methode beruht auf dem Entwurf eines speziellen Prolongationsoperators, der die Konstruktion eines Zwischenzustandes zwischen der groben Skelettfunktion und ihrer Prolongation auf das Feinskelett beinhaltet. Konkret wird auf dem Grobgitter ein *zelldefiniertes* Potential rekonstruiert, das über einen Trace-Operator eine anschließende Definition auf dem Feinskelett ermöglicht.

Die Zellrekonstruktion ist die zentrale Komponente unserer Methode und das, was sie originell macht. Sie arbeitet lokal und gliedert sich in zwei Schritte. Erstens wird ein grobes zelldefiniertes Polynom vom Grad k aus den interface-definierten Polynomen vom Grad k durch die Dekondensation der Zellunbekannten zurückgewonnen. Zweitens wird der Rekonstruktionsoperator höherer Ordnung sowohl auf die Zell- als auch auf die Interfaceunbekannten angewendet, um einen Grad der Approximation in der Zelle zu gewinnen. Da das rekonstruierte Polynom vom Grad $k+1$ ist, impliziert die Rückgewinnung des ursprünglichen Polynoms vom Grad k auf den feinen Flächen, dass die Trace-Operation auch den Grad verringern muss. Um dies zu tun, wird die Spur mit einer anschließenden L^2-orthogonalen Projektion auf den Polynomraum niedrigerer Ordnung k versehen. Darüber hinaus besteht die Spur auf den feinen Flächen am Rand der groben Elemente aufgrund der diskontinuierlichen Einstellung eigentlich darin, den gewichteten Durchschnitt der auf jeder Seite berechneten Spuren zu berechnen. Die Gewichte berücksichtigen den Diffusionskoeffizienten, um Robustheit gegenüber Unstetigkeiten zu gewährleisten.

Die numerischen Tests umfassen homogene und heterogene isotrope Probleme in 2D- und 3D-Gebieten, die durch strukturierte und unstrukturierte Gitter diskretisiert werden. Bei strukturierten Gittern auf einfachen Gebieten, ob mit kartesischen oder simpliziellen Elementen, zeigt das Mehrgitterverfahren wenn es direkt als Löser verwendet wird, folgende Eigenschaften: (i) Konvergenz in einer beschränkten Anzahl von Iterationen, annähernd unabhängig von der Gittergröße; (ii) kontrollierte Rechenkosten durch die Rediskretisierung des Operators auf den groben Ebenen und die Verwendung von Standard-Glättungen (nämlich Block-Gauß-Seidel oder Jacobi); (iii) Robustheit gegenüber Unstetigkeiten des Diffusionskoeffizienten, wobei die Sprunggröße die Konvergenzrate nicht verändert; (iv) Robustheit gegenüber höheren Ordnungen, für die der Löser die gleichen Eigenschaften aufweist.

Auf komplexen Gebieten, die stark unstrukturierte Gitter erfordern, wird jedoch im Allgemeinen keine optimale Konvergenz erreicht. Der Grund dafür ist ein zweifacher: (i) optimale Konvergenz beruht darauf, dass die Interfaces zwischen den Ebenen vergröbert werden (nicht nur die Elemente!); (ii) numerische Experimente haben die hohe Empfindlichkeit des Mehrgitterverfahrens gegenüber der Gitterqualität gezeigt, d. h. gegenüber dem Vorhandensein von Elementen mit ungünstigem Seitenverhältnis. Optimalität erfordert dann auch eine Hierarchie von qualitativ hochwertigen Gittern. Kombiniert werfen diese Anforderungen die Frage auf, wie man die Gitterhierarchie aufbaut. In der Tat werden Mehrgitterhierarchien üblicherweise durch

sukzessive Verfeinerungen eines anfänglichen groben Gitters aufgebaut. Wenn die Verfeinerung eine Interfacevergröberung zwischen jeder Ebene vom feinen Gitter zum groben Gitter sicherstellt, hat sie auch oft die unangenehme Eigenschaft dass die Qualität des Gitters beeinträchtigt wird. Dies gilt besonders in 3D. Umgekehrt gibt es keine offensichtliche Methode, die es erlaubt, ausgehend von einem feinen Gitter guter Qualität ein grobes Teilgitter zu konstruieren und gleichzeitig eine Interfacevergröberung zu erzwingen. Deshalb betrachten wir in der Tat in unserem zweiten Beitrag nichtverschachtelte Gitter, um die Limitierung unserer bisherigen Mehrgittermethode zu überwinden und unstrukturierte 3D-Probleme erfolgreich zu lösen.

Erweiterung auf nicht-verschachtelte Gitter

Kapitel 4 widmet sich der Anpassung der verschachtelten Version unseres Algorithmus an nicht-verschachtelte Gitterhierarchien und dessen effizienten Implementierung für den praktischen Einsatz. Die Anpassung an das nicht-geschachtelte Szenario erfolgt durch das Einfügen eines zusätzlichen Schrittes in die Definition des Prolongationsoperators: Ausgehend von Polynomen, die auf den groben Interfaces liegen, beginnt die geschachtelte Version mit der Rekonstruktion eines gebrochenen elementdefinierten Polynoms auf dem groben Gitter. Dieser Schritt bleibt unverändert. Wir schlagen dann vor, dieses grobe gebrochene Polynom orthogonal in der L^2-Norm auf das nicht-geschachtelte feine Gitter zu projizieren. Der Abschluss des Prozesses folgt schließlich ebenfalls der verschachtelten Variante: Die Spur des Ergebnisses wird auf den feinen Interfaces berechnet.

Die numerische Auswertung dieses L^2-orthogonalen Projektionsoperators hängt von der Projektion der lokalen Grobbasisfunktionen auf die Feinbasen ab, d. h. von der Berechnung der L^2-Skalarprodukte der Grob- und Feinbasisfunktionen über die Feinelemente. In direkter Folge macht die lokale Definition der Funktionsbasen die Schnittpunkte von Grob- und Feinelementen zu den jeweiligen Integrationsstützen dieser inneren Produkte. Die Berechnung der geometrischen Schnittpunkte zwischen Grob- und Feinelementen kann jedoch sehr rechenaufwändig sein. Daher schlagen wir anstelle dieser exakten Berechnung die Implementierung eines approximativen Operators vor, der keine explizite Berechnung der Schnittpunkte erfordert. Er basiert auf der Unterteilung der feinen Elemente, indem die vereinfachende Hypothese angenommen wird, dass jedes Unterelement vollständig in dem groben Element enthalten ist, das sein Baryzentrum enthält. Wir bewerten die Genauigkeit dieser Näherung durch Vergleichsexperimente mit dem exakten Operator, in denen wir die Konvergenz unserer Mehrgittermethode bewerten, bei der die nicht verschachtelten Netze durch unabhängige Retriangulation des Gebiets auf jeder Ebene erhalten werden. Diese Tests zeigen die ausreichende Genauigkeit der Approximation für moderate Polynomgrade in 3D, sowie die substanziellen Verkürzung der Rechenzeit, die das Verfahren durch die Vermeidung der Berechnung geometrischer Schnittpunkte bietet. Insbesondere demonstrieren wir die optimale Konvergenz unseres nicht-verschachtelten Mehrgitter- Algorithmus an einem unstrukturierten 3D-Testfall, den die Version mit geschachtelten Gittern nicht lösen konnte.

In der Praxis kann die Erstellung eines hochwertigen Gitters für eine reale, industrielle Fallstudie eine mühsame Aufgabe sein, so dass ein Ingenieur mehrere Monate lang mit der Gittergenerierung beschäftigt sein kann. Es ist dann nicht realistisch, dass mehrere hochwertige Gitter für die gleiche Geometrie aber mit unterschiedlicher Feinheit verfügbar sind, wie es ein

Mehrgitterlöser benötigen würde. Aus Sicht des Anwenders sollte besser nur ein einziges feines Gitter für den Löser benötigt werden. Deshalb enthält dieses Kapitel auch die abstrakte Definition einer Vergröberungsstrategie, um aus einem gegebenen feinen Gitter eine Hierarchie von nicht verschachtelten Grobgittern aufzubauen, in denen auch die Interfaces vergröbert werden. Die Methode basiert insbesondere auf der Agglomeration von Elementen, zu der wir einen Schritt des Zusammenfassens von Interfaces an den Schnittstellen zwischen Agglomeraten hinzufügen. Wir liefern explizite Details zu unserer Implementierung in 2D, wobei wir insbesondere erklären, wie die approximative L^2-Orthogonalprojektion durch eine geschickte Art der Subtriangulierung der feinen Elemente exakt gemacht werden kann. Das durch diese Vergröberungsstrategie gewonnene nicht verschachtelte Mehrgitterverfahren wird schließlich auf der vereinfachten Geometrie eines realen, industriellen Testfalls von EDF evaluiert, was ebenfalls zu einem asymptotisch optimalen Verhalten führt.

Algebraisches Mehrgitterverfahren

Der von uns entwickelte geometrische Mehrgitteralgorithmus und seine nicht verschachtelte Erweiterung stellen eine erste Möglichkeit zur Lösung von HHO-Systemen dar. In Kapitel 5 entwickeln wir einen weiteren Ansatz in Form eines algebraischen Mehrgitterverfahrens (AMG). Übliche AMG-Löser, die für Finite-Elemente- oder Finite-Differenzen-Methoden niedriger Ordnung entwickelt wurden, leiten die Gitterinformationen unter der Annahme ab, dass jede Zeile in der Matrix einer Unbekannten entspricht, die sich auf einen Freiheitsgrad bezieht, der sich an einem Netzknoten oder Elementen befindet. Auf diese Weise kann der Konnektivitätsgraph des Gitters algebraisch rekonstruiert werden. Vergröberungsstrategien, die geometrische Algorithmen nachahmen, können dann durchgeführt werden um die Grobgitterhierarchie aufzubauen. Insbesondere bei den aggregationsbasierten Methoden werden Knoten aggregiert, um grobe Freiheitsgrade zu erzeugen. In unserem hybriden Ansatz niedrigster Ordnung liegen die Unbekannten des Systems jedoch auf den Interfaces d. h. weder auf den Knoten noch in den Elementen. Daher mag es auf den ersten Blick aus geometrischer Sicht merkwürdig erscheinen, den obigen Ansatz in diesem Kontext anzuwenden. In der Tat kann die aggregationsbasierte Vergröberung dann als Aggregation von Interfaces interpretiert werden. Obwohl sie natürliche Ergebnisse für benachbarte Interfaces liefern kann, insbesondere wenn sie nahezu kollinear sind, werden manchmal Flächen aggregiert, die sich nicht einmal berühren. In diesem Fall ist es dann schwierig, in dieser Aggregation eine geometrische Interpretation zu erkennen. Nichtsdestotrotz zeigen numerische Tests mit einer standardmäßigen aggregationsbasierten AMG-Methode, dass der Ansatz immer noch gut funktioniert, was geometrisch gerechtfertigt werden kann, wenn man vergisst, dass die Freiheitsgrade tatsächlich auf den Interfaces definiert sind und wenn man sie als Knotenwerte interpretiert, die sich in der Mitte der Interfaces befinden. Dennoch kann man sich berechtigterweise fragen, ob eine Vergröberungsstrategie nicht noch bessere Ergebnisse liefern könnte, wenn man die tatsächlichen Bedeutung der Freiheitsgrade als interface-definierte Werte nutzen würde.

Wir beschränken uns auf hybride Methoden niedrigster Ordnung (nicht nur HHO). Die Idee, die dieser Arbeit zugrunde liegt, ist die algebraische Rekonstruktion der Gitterinformation, die nicht mehr auf der kondensierten, sondern auf der nicht kondensierten Matrix basiert. Wie

bei traditionellen AMG-Methoden gewinnen wir die geometrische Information über die Kopplung der Freiheitsgrade aus den algebraischen Daten. Da die kondensierte Matrix jedoch nur Informationen über die Interfaces liefert, verwenden wir die unkondensierte Version, um den Konnektivitätsgraphen zwischen Elementen und Flächen zu rekonstruieren. Sobald das sogenannte algebraische Gitter rekonstruiert wurde und insbesondere die Nachbarschaftsinformationen zwischen den Elementen verfügbar ist, kann eine elementbasierte Aggregationsmethode genutzt werden, um das Verhalten einer geometrischen Vergröberungs- oder Semi-Vergröberungsstrategie zu imitieren. Da in unserer hybriden Umgebung die Interfaces zwischen den Stufen vergröbert werden müssen, ergänzen wir die Element-Aggregation durch die in Kapitel 4 entwickelte Technik der Agglomeration der Interfaces. Die Methode wird in Verbindung mit dem sogenannten K-Zyklus zur Vorkonditionierung eines äußeren Krylov-Verfahrens verwendet. Die in dieser Arbeit getroffenen technischen Entscheidungen sind dem AGMG-Verfahren [95] entlehnt (paarweise Aggregation, starkes negatives Kopplungskriterium, K-Zyklus...), so dass ein angemessener Vergleich mit einem Standard-AMG-Löser möglich wird, der nur auf dem kondensierten System arbeitet.

Unsere Methode wird auf die HHO-Diskretisierungen niedrigster Ordnung von 2D- und 3D-Diffusionsproblemen angewendet. Das Testspektrum umfasst homogene, heterogene, isotrope und anisotrope Probleme auf strukturierten kartesischen Gittern und unstrukturierten Simplexgittern. Die angewandte Methodik vergleicht unsere neue Methode mit einer standardmäßigen aggregationsbasierten AMG, die Freiheitsgrade als Knoten betrachtet und eine knotendefinierte Vergröberungsstrategie aus dem kondensierten System nutzt. Die Aggregationskriterien, der Zyklus, die Glätter sowie alle anderen technischen Entscheidungen sind für beide Löser identisch, um einen Vergleich zu ermöglichen. Wir können über eine gleichwertige Leistung in isotropen und in unstrukturierten Fällen berichten. Der Mehrwert des neuen Algorithmus zeigt sich tatsächlich bei anisotropen Problemen mit kartesischen Netzen, wo der Löser eine verbesserte Robustheit aufweist. Obwohl dieser sehr spezifische, triviale Testfall restriktiv erscheinen mag, kann diese Eigenschaft tatsächlich in einem größeren Bereich von Fällen ausgenutzt werden. Die Methode kann nämlich einen erheblichen Mehrwert für die Lösung von Problemen bieten, die sowohl isotrope als auch anisotrope Regionen umfassen, vorausgesetzt, die anisotropen Regionen werden durch kartesische Elemente diskretisiert, die in Richtung der Anisotropie orientiert sind. Der Löser nutzt in diesem Fall die Flexibilität von AMG, um unstrukturierte Netze in isotropen Regionen zu behandeln, während er die speziellen Elementformen in anisotropen Regionen ausnutzt.

Wissenschaftliche Kommunikation

Die Beiträge dieser Dissertation wurden der wissenschaftlichen Gemeinschaft in Form von Zeitschriftenartikeln, Open-Access-Preprints und Vorträgen auf internationalen Konferenzen zur Verfügung gestellt. Referenzen und Download-Links befinden sich auf Seite 103.

Abstract: We consider a second-order, elliptic partial differential equation (PDE) discretized by the Hybrid High-Order (HHO) method. HHO is a polyhedral method that handles arbitrary polynomial orders, and for which globally coupled unknowns are located at faces. To efficiently solve the linear system arising after static condensation, this work proposes novel, skeleton-based multigrid methods. One is geometric, the other is algebraic. The geometric algorithm is an h-multigrid method that conserves the polynomial degree at every level. It handles non-nested, unstructured, polyhedral meshes. Numerical tests on homogeneous and heterogeneous diffusion problems show fast convergence, scalability in the mesh size and polynomial order, and robustness with respect to heterogeneity of the diffusion coefficient. The algebraic multigrid method (AMG) applies to the lowest order hybrid methods. It leverages the *uncondensed* matrix to extract the connectivity graph between elements and faces, and subsequently implements an element-defined aggregation-based coarsening strategy. Used as a preconditioner, this AMG conserves the performance and scalability of standard plain aggregation AMGs that directly work on the condensed system, while exhibiting notable improvement on anisotropic problems with Cartesian meshes.

Keywords: Partial differential equations, Hybrid High-Order, multigrid, static condensation, non-nested meshes, L^2-orthogonal projection.

*

Abrégé: On considère une équation aux dérivées partielles (EDP) elliptique du second ordre discrétisée par la méthode Hybrid High-Order (HHO). HHO est une méthode polyédrique qui supporte les ordres polynomiaux arbitraires, et pour laquelle les inconnues globalement couplées sont situées aux faces. Afin de résoudre efficacement le système linéaire obtenu après condensation statique, ce travail propose de nouvelles méthodes multigrilles basées sur le squelette du maillage. L'une est géométrique, l'autre algébrique. L'algorithme géométrique est un h-multigrille qui conserve le degré polynomial à tous les niveaux. Il gère les maillages polyédriques non structurés et non imbriqués. Les tests numériques sur des problèmes de diffusion homogènes et hétérogènes montrent une convergence rapide, un comportement asymptotique optimal par rapport à la taille du problème et les ordres polynomiaux, ainsi qu'une grande robustesse aux discontinuités du coefficient de diffusion. Le multigrille algébrique (AMG) s'applique aux méthodes hybrides d'ordre bas. Le graphe de connectivité entre éléments et faces est extrait de la matrice *non condensée*, ce qui permet l'implémentation du méthode de coarsening basée sur l'agrégation des éléments. Utilisé comme préconditionneur, cet AMG conserve les performance et scalabilité des AMGs basés sur l'agrégation simple qui travaillent directement sur le système condensé, tout en affichant une amélioration notable sur les problèmes anisotropiques avec maillage cartésien.

Mots clés : Equations aux dérivées partielles, méthodes hybrides d'ordre élevé, multigrille, condensation statique, maillages non imbriqués, projection L^2-orthogonale.

*

Kurzfassung: Wir betrachten eine elliptische partielle Differentialgleichung (PDE) zweiter Ordnung, die mit der Hybrid High-Order (HHO)-Methode diskretisiert wird. HHO ist eine polyedrische Methode, die beliebige Polynomordnungen erlaubt und bei der global gekoppelte Unbekannte auf den Interfaces zwischen den Elementen liegen. Um das nach der statischen Kondensation entstehende lineare System effizient zu lösen, werden in dieser Arbeit neuartige, skelettbasierte Mehrgitterverfahren vorgeschlagen. Das eine ist geometrisch, das andere algebraisch. Der geometrische Algorithmus ist ein h-Mehrgitterverfahren, das den Polynomgrad auf jeder Hierarchieebene gleich lässt. Es verwendet nicht ineinander geschachtelte, unstrukturierte, polyedrische Gitter. Numerische Tests an homogenen und heterogenen Diffusionsproblemen zeigen schnelle Konvergenz, Skalierbarkeit in der Gittergröße und Polynomordnung, sowie Robustheit gegenüber der Heterogenität des Diffusionskoeffizienten. Das algebraische Mehrgitterverfahren (AMG) gehört zu den Hybridverfahren niedrigster Ordnung. Sie nutzt die *unkondensierte* Matrix, um den Konnektivitätsgraphen zwischen Elementen und Interfaces zu extrahieren, und implementiert anschließend eine elementdefinierte aggregationsbasierte Vergröberungsstrategie. Als Vorkonditionierer verwendet, erhält dieses AMG-Verfahren die Leistung und Skalierbarkeit des aggegrationsbasierten Standard-AMG-Verfahrens, das direkt auf dem kondensierten System arbeiten. Es zeigt darüber hinaus eine bemerkenswerte Verbesserung bei anisotropen Problemen mit kartesischen Netzen.

Schlüsselwörter: Partielle Differentialgleichungen, Hybrid High-Order, Mehrgitter, statische Kondensation, nicht-verschachtelte Netze, L^2-orthogonale Projektion.