Universidad Zaragoza
1542

Trabajo fin de máster
Máster en Mecánica aplicada
Programa oficial de postgrado en Mecánica computacional
Curso 2011-2012

Métodos multimalla geométricos en mallas semi-estructuradas de Voronoi

Pablo Salinas Cortés

5 de junio de 2012

Director: Francisco José Gaspar Lorenz

Departamento de matemática aplicada
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Métodos multimalla geométricos en mallas semi-estructuradas de Voronoi:

Resumen

Las ecuaciones en derivadas parciales son ampliamente utilizadas para modelizar gran cantidad de problemas físicos, debido a que son el tipo de ecuaciones que se usa para representar la naturaleza. Sin embargo, aunque están ampliamente extendidas en su uso, no siempre se pueden resolver analíticamente, y hay por tanto, que resolverlas numéricamente discretizándolas en una malla. Esta metodología, permite obtener los resultados deseados dentro de un dominio dadas unas condiciones de contorno. Por contra, los sistemas de ecuaciones obtenidos son muy grandes. Esto conlleva escoger muy cuidadosamente qué metodología se va a utilizar para resolverlo si no queremos malgastar recursos y tiempo. Dentro de los posibles algoritmos, el método multimalla destaca por su velocidad y por que el número de iteraciones necesarias para resolver es independiente del número de incógnitas.

Dentro de los métodos multimalla, nos encontramos dos, el geométrico y el algebraico. El primero, es muy rápido pero solo se puede aplicar en dominios simples, el algebraico por otra parte es más lento pero se puede usar en formas complejas. Nosotros, en este trabajo hemos utilizado un método semi-estructurado que consiste en realizar una triangulación no estructurada sobre el dominio, como en el caso algebraico, para captar bien la superficie, y posteriormente dentro de cada triángulo, subdividirlos geométricamente, conectando los puntos medios de los lados cuantas veces sea necesario para obtener la precisión deseada. Además, dentro de cada triángulo de la malla no estructurada se pueden seguir diferentes estrategias dependiendo de las características de este, por ejemplo si es equilátero o isósceles, permitiendo una mayor optimización.

Concretamente, hemos utilizado una malla de Delaunay como triangulación grosera y posteriormente hemos discretizado mediante volúmenes finitos. Como centros de los triángulos hemos usado los centros de Voronoi, toda malla Delaunay tiene asociada una de Voronoi, por sus buenas características; como por ejemplo que este centro es siempre perpendicular a los lados, lo que permite aproximar las derivadas con diferencias finitas. Sin embargo, el uso de estos puntos, a pesar de sus ventajas, introduce anisotropía en la discretización, exige el uso de triángulos acutángulos, así como al ir avanzando a lo largo de la jerarquía de mallas los puntos no están anidados, es decir, al restringir o prolongar los puntos no coinciden en coordenadas. Esto significa, que el método multimalla va a perder rendimiento si no hacemos nada para compensarlo. Para solucionar estos dos problemas hemos diseñado un conjunto de suavizadores que compensan los problemas de anisotropía y de no anidamiento, obteniendo unos resultados muy satisfactorios.

Fijándonos en casos más particulares, podemos tener materiales cuyas propiedades varíen de un punto a otro, por ejemplo, un material compuesto. Si la variación de estas propiedades es de varios órdenes de magnitud, el algoritmo multimalla convencional no va a funcionar correctamente. Sin embargo, este problema puede solventarse si en la malla más fina se utiliza la media harmónica en las zonas de contacto y en las mallas bastas del multimalla se tiene en cuenta que se debe mantener la continuidad en el flujo. Esto se consigue utilizando el operador de Galerkin en las mallas más bastas. A su vez, hemos propuesto un algoritmo que permite obtener este operador en el caso de mallas triangulares, utilizando solamente moléculas en su cálculo.

Los métodos propuestos, tanto los suavizadores, como el operador de Galerkin han sido puestos a prueba en varios ejemplos numéricos para comprobar su eficacia.

Contents

Chapter 1

Introduction

In this work, we are interested in the multigrid solution of the large sparse system of equations, arising from the cell-centered finite volume discretization of a two-dimensional partial differential equation on acute Delaunay triangulations. The use of Delaunay grids is motivated by the good properties of the discretizations obtained on this type of triangulations; by requiring them to be acute, we achieve monotonicity properties, see [4, 5, 16]. Every Delaunay grid has an associated dual mesh, known as Voronoi tesselation [28], constituted by convex polygons, each one composed of the points closer to a Delaunay vertex (its center) than to any other.

Voronoi polygons naturally appear in many situations in nature, such as the hottest parts of the sun, ice melting, and even on giraffe skin. Apart from these relations connected with natural science, not directly related to the numerical solution of PDEs, more relevant for us is the use of Voronoi grids for the discretization by finite volumes for oil reservoir simulations, oceanographic models, and other complex problems in science and engineering which are derived from conservation laws.

Finite volume methods are discretizations locally preserving some conservation properties. A first approach to this kind of methods was introduced for one-dimensional problems by Samarskii in 1960, see [24], as a finite difference method called balance method (or integro-interpolation method). For many years finite volume methods on rectangular and triangular grids were used in heat transfer and computational fluid mechanics [6, 20, 22]. From the theoretical point of view, we refer to the reader to Eymard et al. [9].

One of the most important aspects in the numerical solution of partial differential equations is the efficient solution of the corresponding large systems of equations arising from their discretization. Multigrid methods [3, 11, 27] are among the most powerful techniques for solving such type of algebraic systems, and they have become very popular among the scientific community. Basically, multigrid methods are based on the property of a strong smoothing effect on the error of many iterative methods, together with the fact that an smooth function can be well represented on coarser grids, where its approximation is less expensive. The multigrid performance strongly depends on the choice of the components of the algorithm, which are problem dependent, and it is crucial an harmonic interplay between the smoothing and the coarse-grid correction. There are two basic approaches to multigrid solvers: geometric and algebraic

multigrid. Whereas in geometric multigrid a hierarchy of grids must be proposed, in algebraic multigrid no information is used concerning the grid on which the governing PDE is discretized.
The focus on this work is to consider geometric multigrid methods. Not many authors have applied geometric multigrid methods on cell-centered discretizations. Most of these works have been done in rectangular grids. Historically, the pioneer work is due to Wesseling, [31], in which a multigrid method for interface problems was constructed to simulate oil reservoir problems. This work started a chain reaction of papers focused on this subject, see [12, 13, 14, 15, 29, 30, 32]. The W-cycle convergence of these multigrid methods, in the case of natural injection as prolongation, was theoretically analyzed by Bramble et al. [2], and in the case of V-cycle, the convergence was proved by Kwak et al. in [17] using certain weighted prolongation operators. By other hand, on triangular grids, Kwak et al., [18, 19], proposed a new multigrid method, extending their previous works.

For an irregular domain, it is very common to apply regular refinement to an unstructured input grid. In this way, a hierarchy of locally structured grids is generated. To perform this refinement, each triangle is divided into four congruent ones by connecting the midpoints of their edges. In this way, a hierarchy of grids is obtained, where transfer operators between two consecutive levels can be defined. These grids provide a suitable framework for the implementation of a geometric multigrid algorithm, permitting the use of stencil-based data structures, see [1], being necessary only a few stencils to represent the discrete operator, reducing drastically the memory required. In this work, very simple local inter-grid transfer operators have been chosen to make easier the communication between different input blocks. For this reason, powerful smoothers have to be designed. Different smoothers are used on each input block of the initial unstructured grid, depending on its geometry. Notice that these smoothers couldn't be implemented on a pure unstructured grid. We are speaking about, for example, the line-type smoothers, very necessary in the case of anisotropic problems.

Regarding particular applications, the diffusion equation with discontinuous coefficients is widely used in computational fluid dynamics. However, it is well-known that when standard multigrid is applied for solving equations with highly varying discontinuous coefficients, a deterioration in the convergence of the method can be obtained, and even divergence can be observed.

In the case of cell-centered approximations, the main idea to circumvent the problem is to use Galerkin approximation [32] on coarse grids. The disadvantage usually observed when using this approach, is that the stencils of the coarse-grid operators are often larger than the corresponding fine-grid stencil, what is problematic especially in three dimensions. However, note that the use of simple transfer operators preserves four-point stencils in the case of cell-centered schemes on triangular grids.

In the case of composite materials their components have nearly constant diffusivity, but vary by several orders of magnitude. In these cases, it is quite common to idealize the diffusivity by a piecewise constant function. This makes suitable the use of semi-structured grids to deal with these problems.

The rest of the work is organized as follows: Chapter 2 presents the basic principles of Multigrid. Chapter 3 is devoted to present the cell-centered discretization, deriving the corresponding equations which define the discrete operator for both unstructured and structured triangular grids. A multigrid method for this kind of scheme is proposed in Chapter 4. In particular, special smoothers are designed to overcome troubles arising from some difficult grid-geometries. In Chapter 5, the good performance of the proposed multigrid methods is demonstrated on structured triangular grids, comparing the behavior of the considered smoothers for different grid-geometries. In Chapter 6, some test examples are presented to show the applicability of the strategies followed throughout all the work. Finally, in Chapter 7, we focus the previous work in the resolution of diffusion equation with piece-wise discontinuous coefficients.

Chapter 2

Introduction to Multigrid

Multigrid methods, are a family of iterative methods based on the property of some classical iterative algorithms, which can annihilate the high frequency components of the error after few iterations, which leads to an smoothed error, see Figure 2.1.

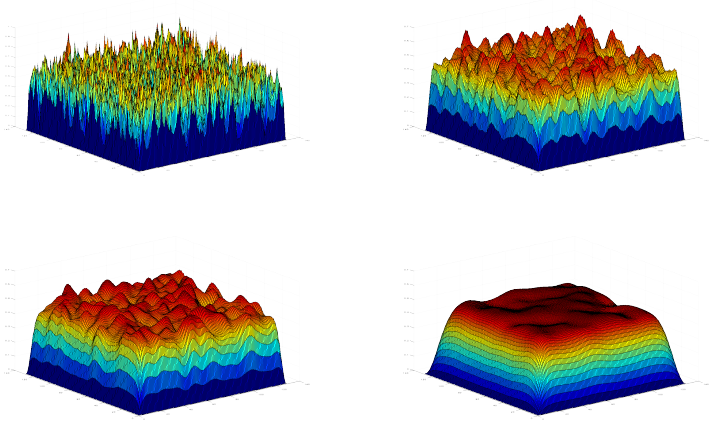


Figure 2.1: Representation of the error after: 0, 5, 10 and 50 iterations of a classical iterative algorithm.

Considering a linear problem,

$$Au = b, \tag{2.1}$$

where A is an $n \times n$ matrix, u the unknowns vector and b the right hand side. Let us consider the following splitting notation:

$$A = M - N, \tag{2.2}$$

where M is an easily invertible matrix. By substituting in 2.1 we obtain $(M - N)u = b$ or equivalently, $Mu = Nu+b$. Now, by denoting u^m as the approximate solution after the m-iteration, the general scheme for an iterative algorithm is:

$$u^{m+1} = M^{-1}Nu^m + M^{-1}b \tag{2.3}$$

If we call u_e to the exact solution, we can define the error in the iteration m, as $e^m = u_e - u^m$, and the residual $r^m = b - Au^m$. We can combine these expressions with (2.1) in order to get a system of equations for the error:

$$r^m = b - Au_e + Ae^m = Ae^m, \tag{2.4}$$

therefore, we can solve a system for the error in order to get a better approximation of the solution. If we now combine the smoothing property, that allows to represent the error in a coarser grid without data loss, with the possibility to solve an equation for the error, we can define the following strategy:

1° Smooth.

2° Represent the error in a coarser grid.

3° Solve the equation exactly.

4° Use the error calculated to correct the approximation, u^m.

5° Iterate until obtaining the desired precision.

The above explained method is called two-grids method. Its efficiency is not a high improvement, however, its convergence rate is constant not matter the number of nodes, this is also known as h-independence rate. Nevertheless, two-grids method can be improved if, in the third step, we approximate the analytical solution by using another two-grids method and so on, leading to the so called multigrid method, which keeps the h-independence rate, and its numerical cost is about $O(n \times n)$.

Regarding the second and the forth step we can split the inter-grid transfer operators in:

- Restriction: Introduce, the data from a fine grid to a coarse one, it can be done by simple injection, just introduce in the coarse grid the value of the nodes that relies in the same coordinates, or by an average of nodes.
- Prolongation: Interpolate the solution of the coarse grid to the nodes of a finer one. The simplest, the adjoint of the simple injection, is to introduce the data from the coarse grid directly to a cluster of nodes.

One can realize that the performance of the multigrid method relies on the smoother, coarse grid transfer method and in the interpolation from a coarser grid to a finer one. The intercourse between the smoother and the inter-grid transfer operators are the key in multigrid. In fact there are two families of multigrid methods, depending on their intercourse:

- Geometric multigrid: The coarsening is fixed, usually the coarser grid is twice the finer, and the smoother must annihilate the half of the frequency errors. These methods are fast, but the geometry must be simple.
- Algebraic multigrid: The smoother is fixed, and the coarsening must adapt itself in order to compensate what the smoother cannot smooth. These family fits very easily complex geometries, but they are slower than the geometric multigrid.

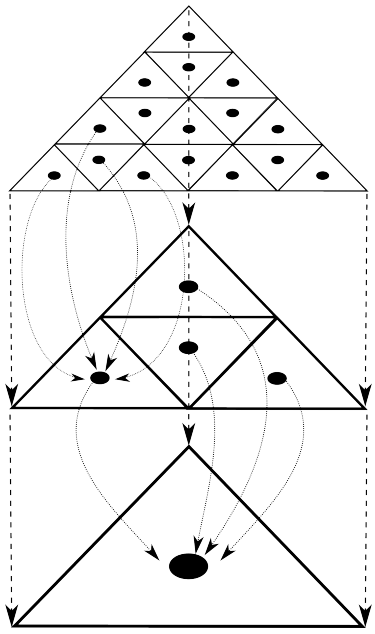


Figure 2.2: Relation, along the grids, between nodes for a simple injection and its adjoint.

Regarding the smoothers, the most common smoothers are the Jacobi iterative method with a relaxation parameter, Gauss-Seidel algorithm and a modification of the Gauss-Seidel, in which you apply a Gauss-Seidel step but in a chessboard manner, i.e: first the odd nodes and next the even ones, see Figure 2.3.

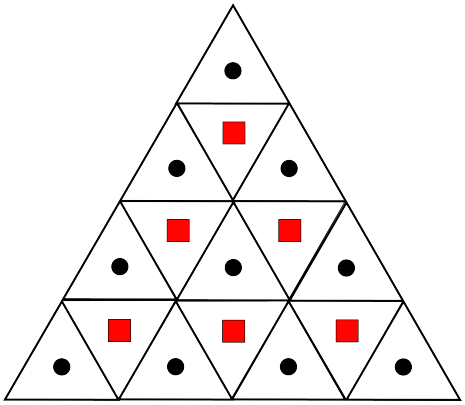


Figure 2.3: Red-Black smoother on a triangular grid.

There are three kind of multigrid methods depending on the effort to approximate the error on coarser grids, one get three different multigrid methods, V-cycle, W-cycle and F-cycle, see Figure 2.4. V-cycle is the fastest, but the weakest, F-cycle is usually preferred over W-cycle since it achieve similar re-

sults with just a 60% of numerical work [27].

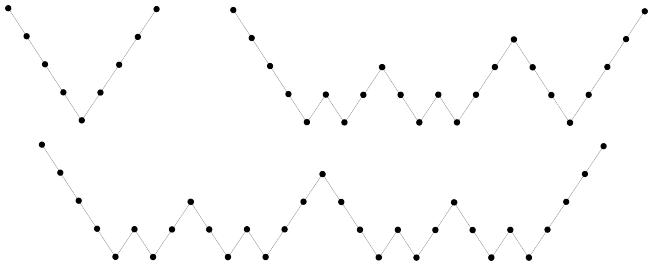


Figure 2.4: V-cycle, F-cycle and W cycle respectively.

As final appointment, multigrid methods perform very well for elliptic and parabolic equations, nevertheless for hyperbolic problems its efficiency is compromised, since the error is more difficult to smooth, however it is also possible [27].

Chapter 3

Discretization of a diffusion problem on Voronoi grids

3.1 Discretization on triangular unstructured grids

We are going to construct a finite volume discretization scheme on the Voronoi mesh associated with a Delaunay triangulation for the following boundary value problem:

$$-\Delta v = f, \text{ in } \Omega, \tag{3.1}$$

$$v = 0, \text{ on } \partial\Omega. \tag{3.2}$$

Firstly, we suppose to have a Delaunay triangulation $\mathcal{T}$ on the domain Ω, satisfying the usual admissibility assumption (see [5]), i.e. the intersection of two different triangles is either empty, a vertex, or a whole edge. Besides, as commented in the introduction, we restrict ourselves to an acute triangulation.

The grid points associated with the cell-centered scheme are the centers of the

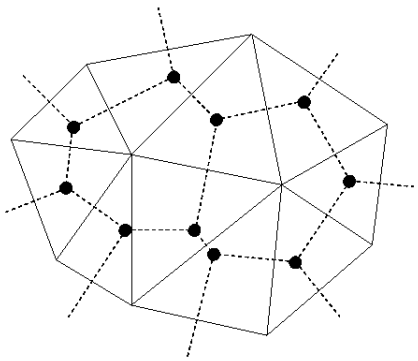


Figure 3.1: Unstructured mesh and its associated Voronoi grid.

circumscribed circle of each triangle, defining a Voronoi mesh, see Figure 3.1. Notice that from the previous restriction with regard to the angles of the triangulation, we are sure that each Voronoi point falls inside of a triangle. More-

over, the segment connecting two neighboring Voronoi points goes trough the midpoint of the common edge of their corresponding triangles, being also perpendicular to it. This latter property allows us to approximate the derivative in the normal direction, by the difference quotient on both Voronoi points. Using the divergence theorem on a triangle T of the triangulation $\mathcal{T}$, the following balance equation holds:

$$-\int_{l_1} \nabla v \cdot \mathbf{n}_1 \, \mathrm{d}l_1 - \int_{l_2} \nabla v \cdot \mathbf{n}_2 \, \mathrm{d}l_2 - \int_{l_3} \nabla v \cdot \mathbf{n}_3 \, \mathrm{d}l_3 = \int_T f(\mathbf{x}) \, \mathrm{dx}. \tag{3.3}$$

where $\mathbf{n}_i$ is the unit outward normal vector on the corresponding edge l_i of triangle T.
Now, each of these line integrals are approximated as the length of the corresponding edge multiplied by the flux evaluated in the midpoint of the edge. Afterwards, we approximate such fluxes using the Voronoi points. With regard to the integral in the right-hand side, we consider the following approximation:

$$\int_T f(\mathbf{x}) \, \mathrm{dx} \approx meas(T) \, f(\mathbf{x}_c), \tag{3.4}$$

being $\mathbf{x}_c$ the Voronoi point of triangle T, and where $meas(T)$ is the area of T.
Denoting $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ the Voronoi points of the triangles adjacent to T with

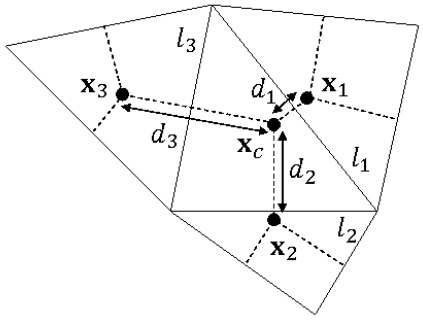


Figure 3.2: Notation for neighboring Voronoi points on an unstructured grid.

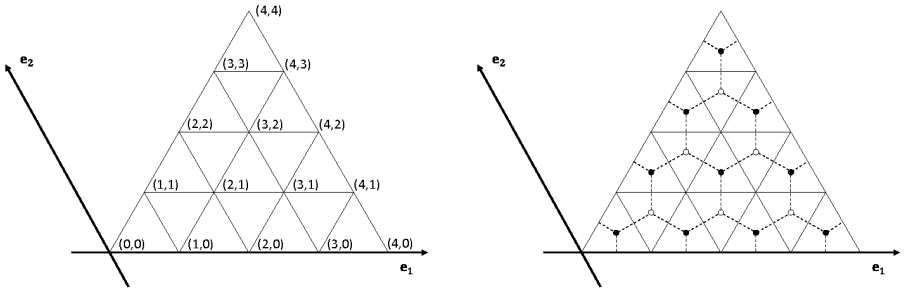
common edges l_1, l_2 and l_3, respectively, and d_i the distance between points $\mathbf{x}_c$ and $\mathbf{x}_i$, $d_i = \mathrm{dist}(\mathbf{x}_c, \mathbf{x}_i)$, with $i = 1, 2, 3$, (see Figure 3.2), we finally obtain the equation corresponding to node $\mathbf{x}_c$:

$$-\frac{1}{meas(T)} \sum_{i=1}^{3} \left(meas(l_i) \, \frac{v_h(\mathbf{x}_i) - v_h(\mathbf{x}_c)}{d_i} \right) = f(\mathbf{x}_c). \tag{3.5}$$

3.2 Discretization on triangular structured grids

Now, we are going to consider the particular case of the discretization of problem (3.1) on an structured triangular grid. In this kind of grids, it is very usual to work in stencil notation because it takes advantage of the structured ordering of the unknowns which contribute in the discretization on a fixed grid-point. In an

structured grid, any point is surrounded by the same grid-pattern, and using a suitable numbering of the grid-points it is easy to capture this pattern in a small matrix or "stencil" which stores the contributions of the neighboring unknowns. Then, first of all, a suitable numbering of the grid-points is needed. In triangular grids, a unitary basis of $\mathbb{R}^2$, $\{\mathbf{e}_1, \mathbf{e}_2\}$, where $\mathbf{e}_1$, and $\mathbf{e}_2$ are unit vectors defining the oblique coordinate system, is considered fitting the geometry of the triangle, as can be seen in Figure 3.3. Hence, a local numeration can be fixed according



a) b)

Figure 3.3: a) New basis in $\mathbb{R}^2$ fitting the geometry of a triangular grid, and local numeration for the regular Delaunay grid obtained on a triangular domain. b) Corresponding Voronoi mesh.

to the definition of the spatial basis. In this way, a manner of numbering nodes very convenient for identifying the neighboring nodes can be defined.
We consider a triangular grid arising on a triangular domain by applying a fixed number of regular refinement steps ℓ. This is done in the way that, on each refinement step every triangle is divided into four congruent ones by connecting the midpoints of their edges.
Then, we can define the corresponding grid in the following way:

$$G_\ell = \{\mathbf{x} = k_1 h_1\, \mathbf{e}_1 + k_2 h_2\, \mathbf{e}_2 \,|\, k_1 = 0, \ldots, 2^\ell,\ k_2 = 0, \ldots, k_1\}, \tag{3.6}$$

where $h = (h_1, h_2)$ is the grid spacing associated with the refinement level ℓ (h_1 is the grid spacing in the direction of $\mathbf{e}_1$, and h_2 in the direction of $\mathbf{e}_2$), so that the grid G_ℓ can also be denoted by G_h.
Thus, for a refinement level ℓ, a local numeration with double index (k_1, k_2), $k_1 = 0, \ldots, 2^\ell$, $k_2 = 0, \ldots, k_1$, is used in such a way that the indexes of the vertices of the triangle are $(0,0)$, $(2^\ell, 0)$, $(2^\ell, 2^\ell)$, as it can also be observed in Figure 3.3a) for $\ell = 2$.

By other hand, the considered discretization is based on the dual Voronoi mesh, represented in Figure 3.3b). In the particular case in which an structured grid as considered here is used, the obtained finite difference scheme results to be different depending on the grid point. More concretely, one-half of the grid points, those corresponding to an up oriented triangle, have the same equation and the other half, those corresponding to a down oriented triangle, have a "mirror image stencil", see Figure 3.4. In this sense, the Voronoi mesh, denoted by V_h could be split up into two sub-grids V_h^u (associated with the up-oriented triangles) and V_h^d, (corresponding to the down-oriented triangles), as seen in Figure 3.3b). These sub-grids can be defined from the grid G_h, in the following

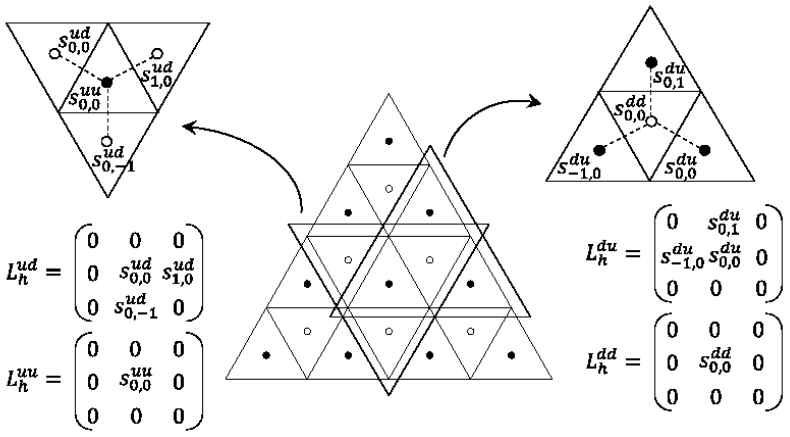


Figure 3.4: Stencils corresponding to two different grid-points: one associated with an up oriented triangle and the other with a down oriented triangle.

way:

$$V_h^u = \{\mathbf{x}^u_{k_1,k_2} = (k_1+\delta_1)h_1\,\mathbf{e}_1 + (k_2+\delta_2)h_2\,\mathbf{e}_2 \mid k_1h_1\,\mathbf{e}_1 + k_2h_2\,\mathbf{e}_2 \in G(\Omega)\} \quad (3.7)$$

$$V_h^d = \{\mathbf{x}^d_{k_1,k_2} = (k_1+\delta'_1)h_1\,\mathbf{e}_1 + (k_2+\delta'_2)h_2\,\mathbf{e}_2 \mid k_1h_1\,\mathbf{e}_1 + k_2h_2\,\mathbf{e}_2 \in G(\Omega)\} \quad (3.8)$$

where δ_i, δ'_i, with $i = 1, 2$, are suitable scalar values to reach Voronoi points from the primal ones following the considered local coordinate system, see Figure 3.5.

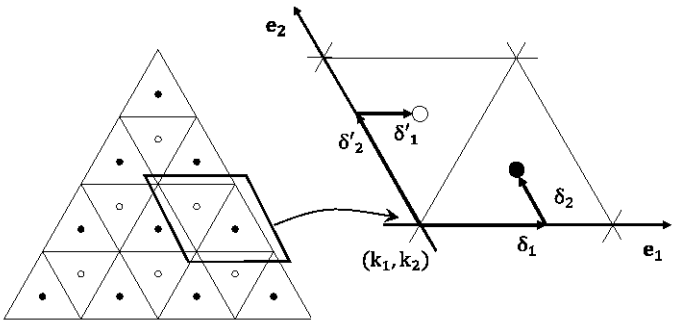


Figure 3.5: Voronoi mesh split into two sub-grids, and corresponding values of δ_i and δ'_i.

Then, a grid-function, v_h, defined on the Voronoi mesh V_h, could be split up into two different sub-grid functions, v_h^u and v_h^d, associated with sub-grids V_h^u and V_h^d, respectively.

In this way, given an arbitrary pair (k_1, k_2) associated with a node of G_h, the equations corresponding to the two Voronoi points $\mathbf{x}^u_{k_1,k_2}$ and $\mathbf{x}^v_{k_1,k_2}$, are

given by

$$L_h^{uu} v_h^u(\mathbf{x}^u_{k_1,k_2}) + L_h^{ud} v_h^d(\mathbf{x}^d_{k_1,k_2}) = f_h^u(\mathbf{x}^u_{k_1,k_2}), \tag{3.9}$$

$$L_h^{du} v_h^u(\mathbf{x}^u_{k_1,k_2}) + L_h^{dd} v_h^d(\mathbf{x}^d_{k_1,k_2}) = f_h^d(\mathbf{x}^d_{k_1,k_2}), \tag{3.10}$$

where these "scalar" operators are given in stencil form as:

$$L_h^{uu} = \frac{1}{meas(T)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{l_1}{d_1} + \frac{l_2}{d_2} + \frac{l_3}{d_3} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad L_h^{ud} = \frac{1}{meas(T)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\frac{l_1}{d_1} & -\frac{l_3}{d_3} \\ 0 & -\frac{l_2}{d_2} & 0 \end{bmatrix},$$

$$L_h^{du} = \frac{1}{meas(T)} \begin{bmatrix} 0 & -\frac{l_2}{d_2} & 0 \\ -\frac{l_3}{d_3} & -\frac{l_1}{d_1} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \qquad L_h^{dd} = \frac{1}{meas(T)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{l_1}{d_1} + \frac{l_2}{d_2} + \frac{l_3}{d_3} & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

3.2.1 Stencil dependence on two angles characterizing the triangular grid.

In order to know a priori the strong and weak connections between neighboring unknowns depending on the grid geometry, we are going to rewrite the stencils in function of some parameters characterizing the grid, that is, two angles, α and β, and the length l, of one edge of an arbitrary triangle of the grid, see Figure 3.6. As we will see, this is going to be very useful for the design of smoothers for different geometries, taking into account the strong connections appearing in the stencils.

Therefore, we are going to describe in detail the computation of the stencil for an arbitrary down-oriented Voronoi grid-point $\mathbf{x}^d_{k_1,k_2}$ in V_h^d. With this purpose, we can write the coordinates of the points involved in such stencil (see Figure 3.6) in terms of the previously explained geometric parameters in the following way:

$$\mathbf{x}^d_{k_1,k_2} = \frac{l}{2}\left(\frac{\sin\alpha\cos\beta}{3\sin(\alpha+\beta)} + 2, \frac{\cos(\alpha-\beta)}{\sin(\alpha+\beta)}\right), \tag{3.11}$$

$$\mathbf{x}^u_{k_1,k_2} = \frac{l}{2}\left(3, -\cot(\alpha+\beta)\right), \tag{3.12}$$

$$\mathbf{x}^u_{k_1-1,k_2} = \frac{l}{2}\left(1, -\cot(\alpha+\beta)\right), \tag{3.13}$$

$$\mathbf{x}^u_{k_1,k_2+1} = \frac{l}{2}\left(\frac{2\sin\alpha\cos\beta}{\sin(\alpha+\beta)} + 1, \frac{2\sin\alpha\sin\beta}{\sin(\alpha+\beta)} - \cot(\alpha+\beta)\right). \tag{3.14}$$

Since all the parameters involved in the stencil (area, distances between Voronoi points and lengths of the edges) are independent of the chosen coordinate system, for simplicity, these coordinates have been computed in the Cartesian coordinate system with respect to the origin, see Figure 3.6.

Due to the fact that the area of an arbitrary triangle T is given in terms of the geometric parameters as

$$meas(T) = \frac{l^2 \sin\alpha\sin\beta}{2\sin(\alpha+\beta)}, \tag{3.15}$$

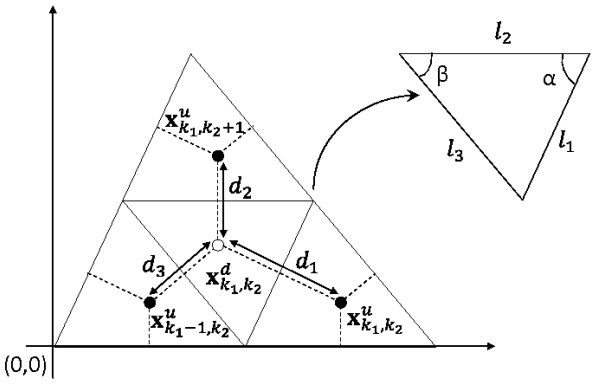


Figure 3.6: Notation for neighboring Voronoi points on an structured grid, characterized by angles α and β.

and the lengths of the sides of T are $l_2 = l$, $l_1 = \dfrac{l \sin\beta}{\sin(\alpha+\beta)}$, and $l_3 = \dfrac{l \sin\alpha}{\sin(\alpha+\beta)}$, after simple calculations, we finally obtain the stencils:

$$L_h^{du} = \frac{2\sin(\alpha+\beta)}{l^2 \sin\alpha \sin\beta} \begin{bmatrix} 0 & \tan(\alpha+\beta) & 0 \\ -\tan\alpha & -\tan\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.16}$$

$$L_h^{dd} = \frac{2\sin(\alpha+\beta)}{l^2 \sin\alpha \sin\beta} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \tan(\alpha+\beta) - \tan\alpha - \tan\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{3.17}$$

As previously commented, for an arbitrary up-oriented Voronoi grid-point $\mathbf{x}^u_{k_1,k_2}$ in V_h^u the stencil would be the "mirror image stencil" of (3.16)-(3.17), that is,

$$L_h^{uu} = \frac{2\sin(\alpha+\beta)}{l^2 \sin\alpha \sin\beta} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \tan(\alpha+\beta) - \tan\alpha - \tan\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{3.18}$$

$$L_h^{ud} = \frac{2\sin(\alpha+\beta)}{l^2 \sin\alpha \sin\beta} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\tan\beta & -\tan\alpha \\ 0 & \tan(\alpha+\beta) & 0 \end{bmatrix}. \tag{3.19}$$

Notice that depending on the angles characterizing the grid, some strong connections appear between unknowns and this fact makes that we will have to take care in the design of the smoothers in a geometric multigrid method, as will be discussed in next chapter.

Chapter 4

Multigrid method

The performance of geometric multigrid methods is strongly dependent on the choice of adequate components to the considered problem. The main components are the smoother $\mathbf{S}_h$, inter-grid transfer operators: restriction $\mathbf{I}_h^{2h}$ and prolongation $\mathbf{I}_{2h}^h$, and the coarse-grid operator $\mathbf{L}_{2h}$. These components have to be chosen so that they efficiently interplay with each other in order to obtain a good connection between the relaxation and the coarse-grid correction. In this chapter the proposed cell-centered multigrid algorithm is described. All the attention is focused in the detailed explanation of the considered smoothers and the special features appearing due to the cell-centered character of the discretization.

Although the presentation of such components is done on a regular structured grid, our purpose is to apply the proposed multigrid method in the framework of semi-structured grids. Therefore, the choice of the corresponding components is done also with a view to this application. In this case, we will use a block-wise multigrid algorithm, where each triangle of the coarsest grid is treated as a different block with regard to the smoothing process. This block-wise strategy is suitable thanks to the possibility of choosing different smoothers for triangles having different geometries, thus resulting in an improvement of the characteristics of our algorithm. Besides, we will have to take care in the communication among the triangles of the coarsest triangulation. Next, we are going to describe the components of the algorithm that we are going to consider throughout all this work.

4.1 Coarse-grid correction

In the application of geometric multigrid, a hierarchy of grids is needed in order to accelerate the convergence of the smoother, by using solutions obtained on the coarser meshes as corrections. As previously commented, in order to obtain such hierarchy of grids, we divide the initial triangles into four congruent ones by connecting the midpoints of the edges, and so forth until the mesh has the desired fine scale to approximate the solution of the problem.
When vertex-centered discretizations are considered on triangular grids, grid points laying on coarser grids also belong to the finer grids, giving rise to a

so-called nested hierarchy of grids. However, when the considered cell-centered discretizations are used, it is worth to note that except in the case of equilateral triangles, the grid hierarchy results to be non-nested, see Figure 4.1.

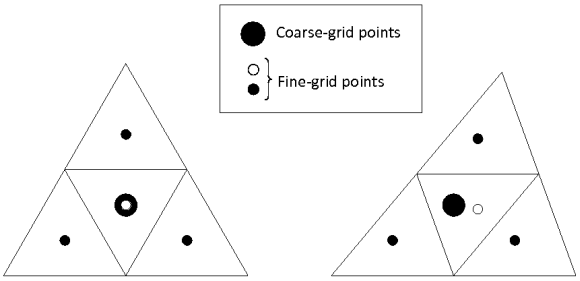


Figure 4.1: Nested (left) and non-nested (right) hierarchies.

This makes the interplay between smoothing and coarse-grid correction specially difficult, requiring the design of new smoothers or new inter-grid transfer operators. With a view to the application of the proposed multigrid method on semi-structured grids, our proposal is to consider very simple inter-grid operators, since this will facilitate the communication between the blocks composing these grids. In particular, injection operator is considered as the prolongation, and its adjoint is chosen as the restriction, resulting in the fact that only the four fine-grid points surrounding a coarse-grid point contribute to its restriction. More concretely, the considered restriction operator, $\mathbf{I}_h^{2h}$, is given in the following way

$$\mathbf{I}_h^{2h} = \begin{bmatrix} (I_h^{2h})^{uu} & (I_h^{2h})^{ud} \\ (I_h^{2h})^{du} & (I_h^{2h})^{dd} \end{bmatrix}, \text{ with } \begin{cases} (I_h^{2h})^{uu} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & (I_h^{2h})^{ud} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/4 & 1/4 \\ 0 & 1/4 & 0 \end{bmatrix}, \\ (I_h^{2h})^{du} = \begin{bmatrix} 0 & 1/4 & 0 \\ 1/4 & 1/4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, & (I_h^{2h})^{dd} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1/4 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \end{cases}$$

as shown in Figure 7.9, and the corresponding prolongation fulfils $\mathbf{I}_{2h}^{h} = 4\,\mathbf{I}_h^{2h}$. The choice of these inter-grid transfer operators leads us to make an effort in the

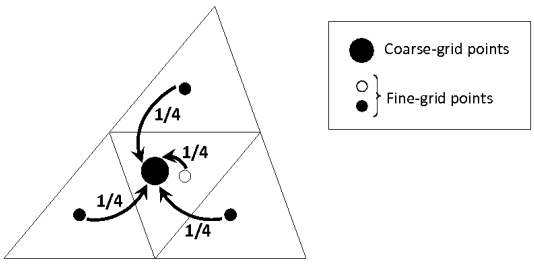


Figure 4.2: Restriction operator.

smoothing process. Then, we must design efficient smoothers capable of taking charge of the remaining components of the error, which cannot be eliminated by the coarse-grid correction part of the algorithm. Actually, the design of suitable smoothers for cell-centered grids is a challenge in this context.

4.2 Smoothers

The smoother usually plays an important role in multigrid algorithms, mainly in the geometric approach, resulting the choice of a suitable smoother an important feature for the efficiency of these methods. Moreover, as previously commented, in the framework we are working with, this choice is even more relevant.
Due to the general observation that errors become smooth if strongly connected unknowns are collectively updated, appropriate smoothers have been designed depending on the magnitude of the coefficients of the stencils, given in (3.16)-(3.19). The following smoothers have been considered and tested in order to fulfil the previous requirement.

4.2.1 Jacobi smoother.

For almost equilateral triangles, the magnitude of all the entries of the stencils is similar, and therefore a point-wise smoother is enough to satisfactorily reduce the components of the error. The easiest smoother to perform is a Jacobi type smoother, which consists of computing the approximation of each unknown, by using non-updated values of the rest of the unknowns. Notice that this implies that it makes not difference the order in which the grid points are visited, what makes Jacobi scheme well-suited for parallel processing. However, for difficult problems, usually this smoother does not give enough satisfactory results, and some variants have to be considered.
As is the case for Jacobi smoother presented here, some standard smoothers are based on a decomposition on the positive and negative parts of the operator, which correspond to the updated an non-updated unknowns before the current step. Here we are going to present the corresponding decomposition for Jacobi smoother. In order to do this, only the positive part of the operator is displayed. Taking into account that only the diagonal blocks of the operator contribute in this positive part, it holds

$$(L_h^{uu})^+ = L_h^{uu}, \quad (L_h^{dd})^+ = L_h^{dd}. \tag{4.1}$$

4.2.2 Red-Black smoother.

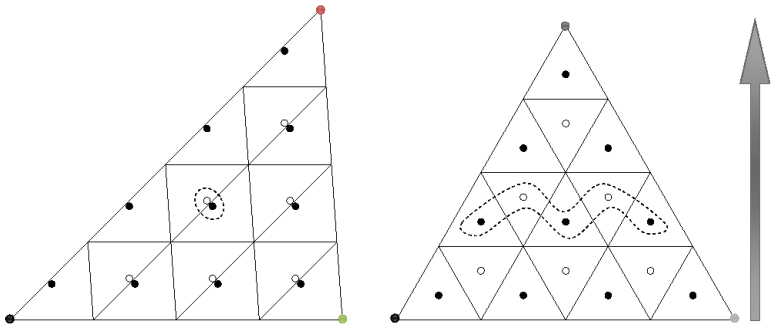
Due to the fact that unknowns related to up or down-oriented triangles have no direct connection with each other, it seems natural to simultaneously update all unknowns associated with equally oriented triangles, giving rise to a pattern relaxation scheme. Since two different types of grid-points are distinguished, a two-color relaxation process, called here Red-Black smoother, is considered.
More concretely, one iteration of this relaxation scheme consists of two partial steps. In the first one, unknowns corresponding to up-oriented triangles are updated, and in the second step those associated with the down-oriented triangles are relaxed by using the updated values. Thus, the complete smoothing operator $\mathbf{S}_h$ is given by the composition of two partial step operators, $\mathbf{S}_h^u$ and $\mathbf{S}_h^d$, which correspond to apply a Jacobi step on each type of grid-points, that is, $\mathbf{S}_h = \mathbf{S}_h^d \, \mathbf{S}_h^u$. These partial step operators are characterized by a different decomposition than the previous Jacobi over all the grid-points, in the way that for $\mathbf{S}_h^u$, for example, the positive parts of the scalar operators are:

$$(L_h^{uu})^+ = L_h^{uu}, \quad (L_h^{dd})^+ = I_h, \tag{4.2}$$

and if $\mathbf{S}_h^d$ is considered, the identity operator will correspond to $(L_h^{uu})^+$, and $(L_h^{dd})^+ = L_h^{dd}$.

4.2.3 Diamond smoothers.

For almost rectangular triangles, a strong connection between only two nodes involved in the stencil appears, due to the anisotropy of the Voronoi mesh. Therefore, since the common lore claims that smoothing must be done in the direction of the strong connection, in this case, both unknowns will have to be simultaneously relaxed. These unknowns are associated with the closest Voronoi points corresponding to different-oriented triangles, as seen in Figure 4.3a). Therefore, a small (2×2)-system must be solved for each of these pairs of unknowns. Different orderings can be chosen to visit these blocks. We have chosen the lexicographic one, but of course, many orderings are possible.
In triangular grids, three different diamond smoothers, associated with the three edges of a triangle, can be defined. If a triangle characterized by angles α and β is considered, we can assign a different color to each of its vertex, in the way that, for example, black color is associated with the vertex corresponding to angle α, green color with that vertex associated with β, and red color corresponds to vertex of angle $180^{\text{O}} - (\alpha + \beta)$. In this manner, each diamond smoother can be named with the color corresponding to the vertex opposite to its associated edge. Following this rule, diamond smoother appearing in Figure 4.3a) is called green-diamond smoother.



a) b)

Figure 4.3: a) Green-diamond smoother and b) Red-wormy smoother.

This smoother is based on a decomposition of positive and negative parts of the operator. Although three different diamond smoothers can be considered, the corresponding decomposition can be done analogously. Then, in order to do this description, we consider the green-diamond smoother, which can be seen in Figure 4.3a). The positive parts of the involved scalar operators are given as

follows:

$$(L_h^{uu})^+ = L_h^{uu},\ (L_h^{ud})^+ = \begin{bmatrix} 0 & 0 & 0 \\ 0 & s_{0,0}^{ud} & 0 \\ 0 & s_{0,-1}^{ud} & 0 \end{bmatrix},$$
$$(L_h^{du})^+ = \begin{bmatrix} 0 & 0 & 0 \\ s_{-1,0}^{du} & s_{0,0}^{du} & 0 \\ 0 & 0 & 0 \end{bmatrix},\ (L_h^{dd})^+ = L_h^{dd}. \tag{4.3}$$

4.2.4 Wormy smoothers.

When a very small angle characterizes the triangulation, the strong coupling appears between the Voronoi points associated with the up and down-oriented triangles connected in the direction of the edge opposite to the vertex corresponding to this small angle. Therefore, all those points are simultaneously updated, see Figure 4.3b), and a tridiagonal system must be solved for each of these "wormy-lines". For this reason, this smoother will be called wormy-smoother.
Similarly to the previous case of the diamond smoother, in triangular grids, three different wormy-smoothers can be defined associated with the three edges of a triangle. Again, each wormy-smoother can be named with the color corresponding to the vertex opposite to its associated edge. Following this criterion, wormy-smoother appearing in Figure 4.3b) is called red-wormy smoother. Notice that these smoothers can be performed by visiting the "lines" from vertex to edge or, on the contrary, from edge to vertex, and this latest is the chosen option. From the description of this smoother, we can obtain the decomposition of the discrete operator which gives rise to wormy-smoother. Analogously to the diamond smoother, three different wormy smoothers can be defined, whose decomposition can be obtained in a similar way. In order to present the corresponding decomposition of the operator, the red-wormy smoother, displayed in Figure 4.3b), is considered. In this way, the positive parts of the scalar operators result as follows:

$$(L_h^{uu})^+ = L_h^{uu},\ (L_h^{ud})^+ = \begin{bmatrix} 0 & 0 & 0 \\ 0 & s_{0,0}^{ud} & s_{1,0}^{ud} \\ 0 & s_{0,-1}^{ud} & 0 \end{bmatrix},$$
$$(L_h^{du})^+ = \begin{bmatrix} 0 & 0 & 0 \\ s_{-1,0}^{du} & s_{0,0}^{du} & 0 \\ 0 & 0 & 0 \end{bmatrix},\ (L_h^{dd})^+ = L_h^{dd}. \tag{4.4}$$

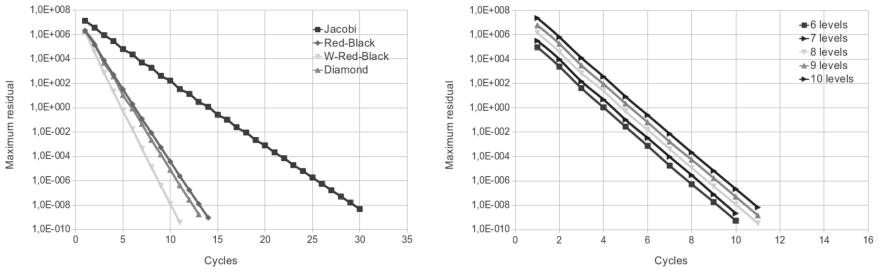
Concluding, we can say that each of these wormy-smoothers will be suitable when the angle corresponding to the vertex of its color is small, and in this way, any possible triangulation will have associated a wormy-smoother giving a satisfactory convergence factor.

Chapter 5

Results of the proposed multigrid method on structured grids

In order to see the suitability of the introduced smoothers in the design of an efficient multigrid algorithm, we present some experiments comparing their behavior in regular structured grids. In particular, model problem (3.1)-(3.2) is solved in the structured grid arising from the regular refinement of a triangular domain, characterized by two of their angles. In all the presented experiments, an $F(2,2)-$cycle is considered. $F-$cycle is preferred to $V-$cycle due to the poor chosen inter-grid transfer operators.

We begin by considering an equilateral triangular domain. As any anisotropy arises from the grid resulting of a regular refinement, it could be seen natural to think in applying a simple point-wise smoother, like Jacobi or lexicographic Gauss-Seidel, to this kind of triangulations. However, as previously commented, for this situation a pattern relaxation scheme could be more appropriately. Thus, we are going to present some convergence results on a regular equilateral grid, comparing the behavior of multigrid by considering: undamped Jacobi smoother, Red-Black smoother, $\omega-$Red-Black smoother (with $\omega = 1.15$) and diamond smoother. In Figure 5.1a), we show the history of the convergence on a grid obtained after eight refinement levels, and by considering as stopping criterion to reduce the maximum residual until 10^{-8}. First of all, a rather surprising observation could be concluded from this figure: the performance of undamped Jacobi appears to be a satisfactory choice as smoother for cell-centered discretizations on triangular grids (as also seen for other type of discretizations on triangular grids, [10], and in the context of full-multigrid on rectangular grids, [23]), despite the well-known lack of smoothing property of this iterative scheme. Notwithstanding this unusual behavior, the obtained Jacobi results are largely improved by Red-Black smoother and diamond smoother. At the same time, the convergence factors provided by both smoothers are enhanced by the Red-Black smoother with relaxation parameter $\omega = 1.15$, which has been obtained by experimental tests. This improving effect was pointed out in [21] for cell-centered discretizations on rectangular grids, where it was validated by a local Fourier analysis. Moreover, the good behavior of the multigrid based

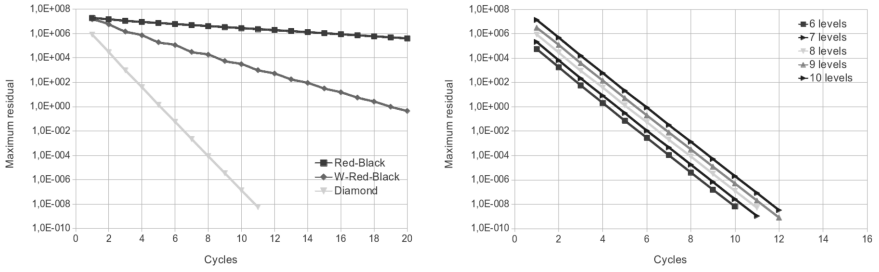


a) b)

Figure 5.1: a) Comparison of smoothers on an equilateral triangular grid. b) History of the convergence for different numbers of refinement levels by using $\omega-$Red-Black smoother.

on the $\omega-$Red-Black smoother is confirmed in Figure 5.1b), where its robustness with regard to the discretization parameter is shown. In this figure, the history of the convergence of the method is displayed for different numbers of refinement levels, resulting to be independent. Therefore, we conclude from this experiment that $\omega-$Red-Black smoother seems to be a good choice for almost equilateral triangulations.
However, this good behavior deteriorates very quickly when the shape of the triangle tends to be rectangular or is characterized by a very small angle.

In the case of almost rectangular triangles, point-wise smoothers are not suitable anymore due to the anisotropy of the Voronoi mesh. By other hand, diamond smoother results in a very efficient smoother when this kind of grids are considered. As an example, a triangular domain characterized by angles $\alpha = 45^{\mathrm{O}}$ and $\beta = 85^{\mathrm{O}}$ is fixed. In Figure 5.2a), the history of the multigrid convergence by using different smoothers is displayed. More concretely, Red-Black, $\omega-$Red-Black and diamond smoothers are used in this comparison. In all cases, the finest grid results by applying eight refinement levels to the initial triangular domain.



a) b)

Figure 5.2: a) Comparison of smoothers on an almost rectangular triangular grid. b) History of the convergence for different numbers of refinement levels by using diamond smoother.

As we can observe, very poor rates are obtained when both Red-Black smoothers are considered, whereas the convergence factors provided by the new diamond smoother are very satisfactory, achieving the convergence in only eleven cycles. Besides, in Figure 5.2b), where the history of the convergence is shown for different numbers of refinement levels, the robustness of this smoother with respect to the space discretization parameter is demonstrated.
Although convergence factors provided by diamond smoother are very satisfactory for many grid configurations, when a triangulation characterized by a very small angle is used, this smoother gives rise to poor rates. This behavior can be seen in Figure 5.3, where asymptotic convergence factors of the diamond smoother based multigrid are shown for a wide range of pairs of angles characterizing the grid.

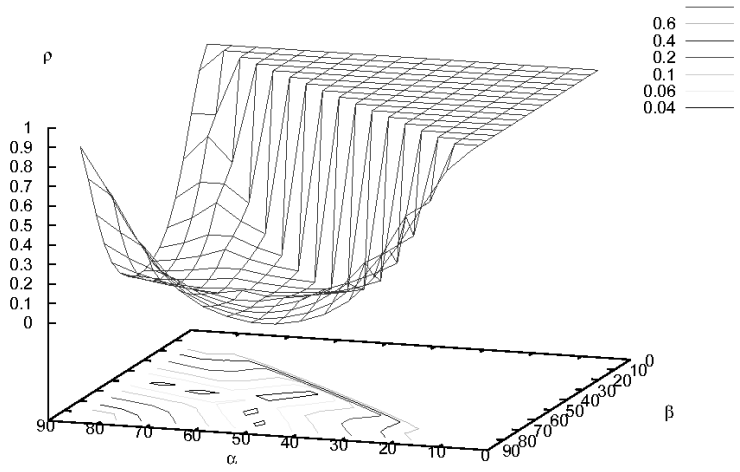
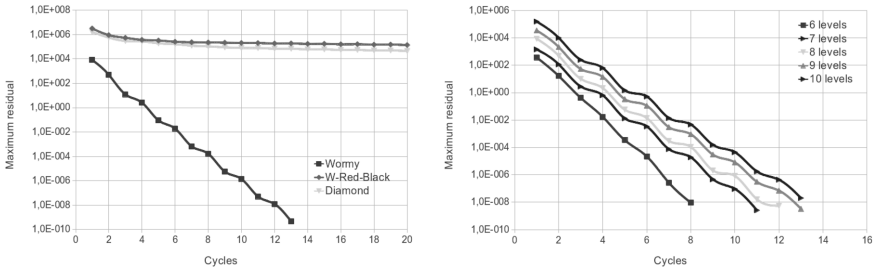


Figure 5.3: Experimentally computed convergence factors for the diamond smoother based multigrid and four smoothing steps, for different triangles in function of two of their angles.

To overcome these troubles appearing when the primal mesh is anisotropic, wormy-smoother in the direction of the anisotropy is a suitable smoother, largely improving the convergence factors provided by the rest of point-wise or block-wise smoothers. To validate this statement, we are going to compare the multigrid convergence by using each one of the smoothers proposed in this work, when an isosceles triangle with a small angle of 10^{o} is considered as domain of our problem. With this purpose, in Figure 5.4a), the multigrid convergence provided by using $\omega-$Red-Black, diamond and wormy smoothers is depicted for eight refinement levels. From this picture, it is clear that wormy-smoother is the best choice for this type of triangulations. Moreover, an $h-$independent convergence is also shown in Figure 5.4b).

Concluding, from the results presented in this chapter, it seems that a reasonable strategy to follow would be to apply the point-wise $\omega-$Red-Black smoother for almost equilateral triangles, the collective diamond smoother for almost rectangular triangles, and finally the appropriate block collective wormy smoother



a) b)

Figure 5.4: a) Comparison of smoothers on an isosceles triangular grid with smallest angle 10^{o}. b) History of the convergence for different numbers of refinement levels by using wormy-smoother.

when triangulations with a small angle appear.

Chapter 6

Numerical experiments

From the practical point of view, for any given triangular geometry would be nice to be capable of choosing a suitable smoother in order to reach a desired convergence factor. Moreover, for semi-structured grids, it is imperative to know the smoother to use for each triangle of the input grid, in order to achieve globally a desired convergence factor. In order to reach this, a set-up phase has been implemented in the multigrid code; it consists of reading an already calculated database containing the most efficient strategy depending on the triangle angles. The corresponding guideline to reach a global convergence factor about 0.1 is shown in Figure 6.1, and it has been numerically calculated by doing extensive computations in regular triangular grids.

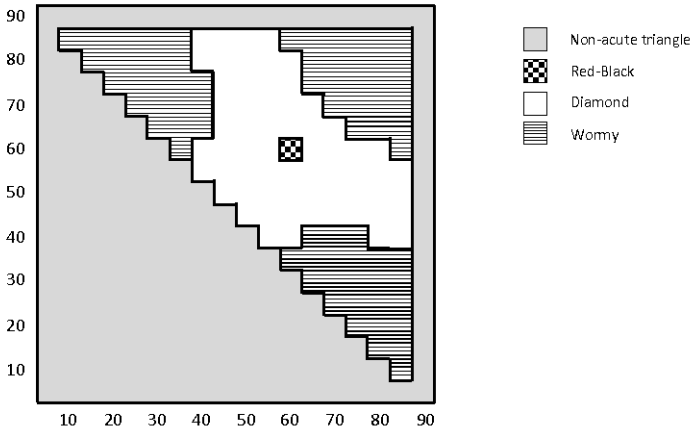


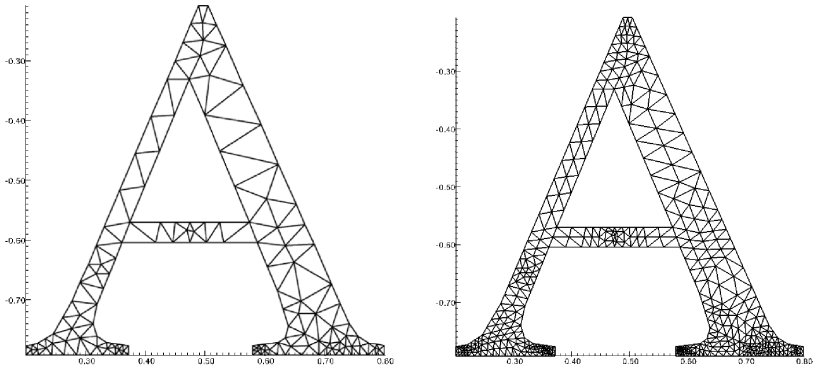
Figure 6.1: Guideline to choose suitable smoothers to reach an asymptotic convergence factor about 0.1 on different triangles.

Strategy shown in Figure 6.1 has been followed in the two model problems considered here: a Laplace problem in an A-shaped domain, and a convection-diffusion problem in an square domain. In both problems, *aCute* software, [7, 8], which is based on *Triangle*, [25, 26], has been used to generate the initial

unstructured acute Delaunay triangulation.

6.1 Laplace problem in an A-shaped domain

In the first numerical experiment, model problem (3.1) is solved in an A-shaped domain, as shown in Figure 6.2. To this purpose, an initial unstructured grid composed of 201 triangles is considered, as depicted in Figure 6.2a). From this mesh, a hierarchy of grids is constructed by applying regular refinement, and the grid resulting after one refinement step is shown in Figure 6.2b), as an example.



a) b)

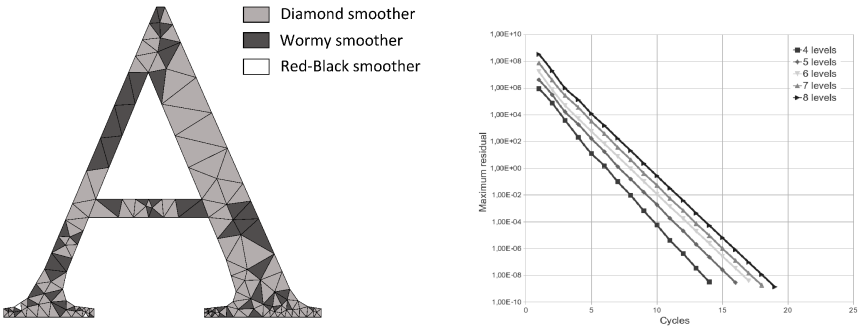
Figure 6.2: a) Coarsest unstructured grid. b) Grid obtained after one regular refinement level.

Following the guideline displayed in Figure 6.1, we have chosen the most efficient smoother for each triangle of the input unstructured triangulation. Selected smoothers can be seen in Figure 6.3a). It is important to remark that to achieve the desired global convergence factor, an extra relaxation on Voronoi nodes close to the internal boundaries of the initial coarsest grid, has been necessary.

After applying the proposed strategy by using an F-cycle, the history of the convergence for different numbers of refinement levels is shown in Figure 6.3b). The stopping criterion has been chosen as the maximum residual to be less than 10^{-8}. It is observed that the convergence is independent of the space discretization parameter h, and that in few iterations the residual is reduced as desired. Moreover, an asymptotic convergence factor about 0.1 is obtained by taking a random initial guess and zero right-hand side.

6.2 Convection-diffusion problem on a square domain

The strategy proposed in this work can be applied to more complex problems as, for example, a convection-diffusion model, which can be written in divergent



a) b)

Figure 6.3: a) Different smoothers for the triangles composing the initial triangulation of the A-shaped domain. b) Multigrid convergence for Poisson problem on the A-shaped domain.

form as:

$$-\nabla \cdot (\nabla v + \mathbf{b}\, v) = f, \text{ in } \Omega, \tag{6.1}$$

where $\mathbf{b}(\mathbf{x})$ is a given velocity field, whose divergence is assumed to be zero. In order to obtain a difference scheme by the cell-centered finite volume method, we follow the same approach that we have explained in detail in Chapter 3, by using a central difference scheme to approximate the convective term. In this numerical experiment an square domain of unit length and Dirichlet boundary conditions are considered, and a constant vector $\mathbf{b} = (1, 0)$ is fixed in the whole domain. Thus, the following equation on each of the grid-nodes $\mathbf{x}_c$ results:

$$-\frac{1}{meas(T)} \sum_{i=1}^{3} \left(meas(l_i) \left(\frac{v_h(\mathbf{x}_i) - v_h(\mathbf{x}_c)}{d_i} + \mathbf{b} \cdot \mathbf{n}_i \frac{v_h(\mathbf{x}_i) + v_h(\mathbf{x}_c)}{2} \right) \right) = f(\mathbf{x}_c). \tag{6.2}$$

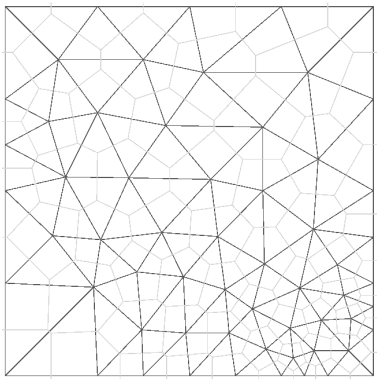


Figure 6.4: Coarsest unstructured grid together with the associated Voronoi mesh.

We consider an initial unstructured grid, composed of 96 triangles, as seen in Figure 6.4, in which, for illustration, the dual Voronoi mesh has been displayed. The hierarchy of grids is obtained by regular refinement. As the convective part of the problem is not dominant, and its derivatives are of lower order, the behavior of the multigrid will be similar to that obtained for a pure diffusive problem, and therefore we will follow the guideline given in Figure 6.1 to choose the suitable local smoother on each input triangle, and this selection is displayed in Figure 6.5. The proposed geometric multigrid method is applied to solve the corresponding large sparse linear system of equations. An F-cycle is used to test the independency of the multigrid convergence with regard to the discretization parameters.

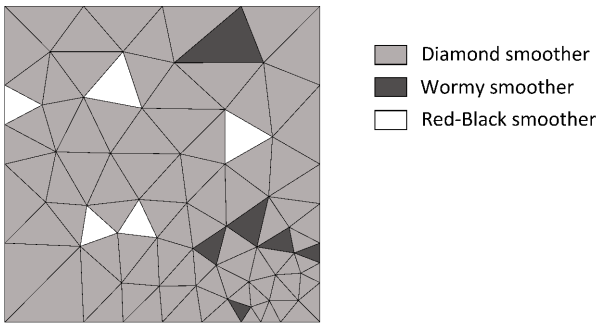


Figure 6.5: Different smoothers considered on each triangular block of the input grid.

N. of levels	N. of unknowns	N. of cycles	ρ_h
4	24576	9	0.09
5	98304	9	0.09
6	393216	9	0.09
7	1572864	9	0.10
8	6291456	9	0.10

Table 6.1: Number of iterations to reduce the initial residual in a factor of 10^{-10}, and corresponding asymptotic convergence rates for different numbers of refinement levels, by using an F-cycle

In Table 6.1, for different numbers of refinement levels, the asymptotic convergence rate, ρ_h, and the number of iterations necessary to reduce the initial residual in a factor of 10^{-10}, are displayed.

Chapter 7

Piecewise discontinuous diffusion coefficient

In this chapter we will consider the diffusion problem:

$$-\nabla \cdot (\kappa(x,y)\nabla u) = f, \text{ in } \Omega, \tag{7.1}$$

$$u = g, \text{ on } \Gamma \tag{7.2}$$

where $\kappa(x,y)$ is a discontinuous the diffusion coefficient. In particular, here we are interested in problems were κ is piecewise constant. Considering again the cell-centered discretization scheme, equation 3.5 reads now as:

$$-\frac{1}{meas(T)} \sum_{i=1}^{3} \left(\kappa_i^H meas(l_i) \frac{v_h(\mathbf{x}_i) - v_h(\mathbf{x}_c)}{d_i} \right) = f(\mathbf{x}_c). \tag{7.3}$$

Where $\mathbf{x}_1$, $\mathbf{x}_2$, $\mathbf{x}_3$ are the Voronoi points of the triangles adjacent to T with common edges l_1, l_2 and l_3, respectively, and d_i the distance between points $\mathbf{x}_c$ and $\mathbf{x}_i$, with $i = 1, 2, 3$, (see Figure 3.2). The coefficient κ_i^H appearing in (7.3) is the harmonic average given by:

$$\kappa_i^H = \frac{2\kappa_c \kappa_i}{\kappa_c + \kappa_i}, \tag{7.4}$$

which is the most accurate method of known techniques of averaging, [32, 24].

In this way, given an arbitrary pair (k_1, k_2) associated with a node of G_h, the equations corresponding to the two Voronoi points $\mathbf{x}^u_{k_1,k_2}$ and $\mathbf{x}^v_{k_1,k_2}$, are given by

$$L_h^{uu} v_h^u(\mathbf{x}^u_{k_1,k_2}) + L_h^{ud} v_h^d(\mathbf{x}^d_{k_1,k_2}) = f_h^u(\mathbf{x}^u_{k_1,k_2}), \tag{7.5}$$

$$L_h^{du} v_h^u(\mathbf{x}^u_{k_1,k_2}) + L_h^{dd} v_h^d(\mathbf{x}^d_{k_1,k_2}) = f_h^d(\mathbf{x}^d_{k_1,k_2}), \tag{7.6}$$

where "scalar" operators L_h^{uu}, L_h^{ud}, L_h^{du} and L_h^{dd} can be obtained from equation (7.3), and are given in stencil form as:

$$L_h^{uu} = \frac{1}{meas(T)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa_1^H \frac{l_1}{d_1} + \kappa_2^H \frac{l_2}{d_2} + \kappa_3^H \frac{l_3}{d_3} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad L_h^{ud} = \frac{1}{meas(T)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\kappa_1^H \frac{l_1}{d_1} & -\kappa_3^H \frac{l_3}{d_3} \\ 0 & -\kappa_2^H \frac{l_2}{d_2} & 0 \end{bmatrix},$$

$$L_h^{du} = \frac{1}{meas(T)} \begin{bmatrix} 0 & -\kappa_2^H \frac{l_2}{d_2} & 0 \\ -\kappa_3^H \frac{l_3}{d_3} & -\kappa_1^H \frac{l_1}{d_1} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad L_h^{dd} = \frac{1}{meas(T)} \begin{bmatrix} 0 & 0 & 0 \\ 0 & \kappa_1^H \frac{l_1}{d_1} + \kappa_2^H \frac{l_2}{d_2} + \kappa_3^H \frac{l_3}{d_3} & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where the distances d_1, d_2, d_3 and the lengths l_1, l_2, l_3 are defined depending on the orientation of the triangle, as seen in Figure 7.1. For example, for an up-oriented triangle d_2 is defined as the distance between $\mathbf{x}^u_{k_1,k_2}$ and $\mathbf{x}^d_{k_1,k_2-1}$, and l_2 as the length of the edge between those Voronoi points.

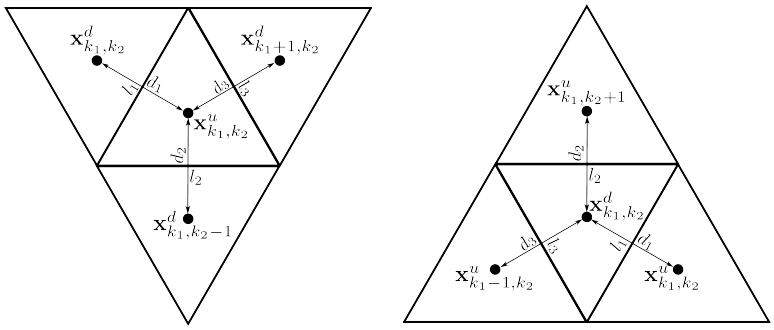


Figure 7.1: Notation used to construct the stencil on a Voronoi point at (a) an up-oriented triangle or at (b) a down-oriented triangle.

When the semi-structured grid is considered, different stencils are necessary to describe the discrete operator on the structured grid arising on a triangle T of this mesh. Notice that all the down-oriented triangles comprising this grid are "interior", that is, they only connect with up-oriented triangles of T. In this way, the stencils corresponding to their Voronoi nodes will be identical. However, for those grid-points associated with up-oriented triangles, different stencils appear depending on their location, since it is necessary to take into account the connections with other triangles belonging to other block of the coarsest grid.

7.1 Galerkin operator

When large jumps in the diffusion coefficient κ occur in the domain, a direct discretization on coarse grids may not work properly. However, in most of the multigrid methods for discontinuous coefficients problems, the Galerkin approach have been used satisfactorily. This method consists of defining the coarse-grid operator in terms of the fine-grid operator, $\mathbf{L}_k$, the restriction,$\mathbf{I}_k^{k-1}$ and the prolongation, $\mathbf{I}_{k-1}^k$, in the following way:

$$\mathbf{L}_{k-1} = \mathbf{I}_k^{k-1} \mathbf{L}_k \mathbf{I}_{k-1}^k \tag{7.7}$$

With this method, it is not straight to work directly with stencil notation as the operations are matrix multiplications.

We propose an easy algorithm to calculate $\mathbf{L}_{k-1}$ using only stencils. This method must be applied once per stencil operator. Firstly, the notation for this algorithm needs to be modified, nevertheless using the following conversion of the previously explained stencils, its application is straightforward:

$$Stencil \quad = \quad \begin{bmatrix} 0 & 0 & 0 \\ L_h^{ud}(0,0) & L_h^{uu}(0,0) & L_h^{ud}(1,0) \\ 0 & L_h^{ud}(0,1) & 0 \end{bmatrix}. \tag{7.8}$$

Secondly, we have to define an stencil of stencils:

$$S_t(j,i) \quad = \quad \begin{bmatrix} 0 & 0 & 0 \\ S_E(jj,ii) & S_C(jj,ii) & S_W(jj,ii) \\ 0 & S_N(jj,ii)|S_S(jj,ii) & 0 \end{bmatrix}, \tag{7.9}$$

Where the subfix is their position in cardinal coordinates, being S_C the center, see Figure 7.2.

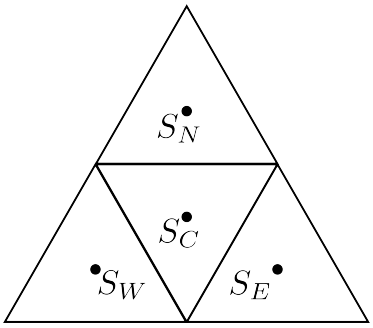


Figure 7.2: Stencils notation to be used in the Stencil of Stencils

All the stencils coordinates must be defined from -1 to 1, being the center $(0,0)$, and the κ must be included. The stencil of stencil, S_t must be a 4 dimensional array in which the two first coordinates are the position of the stencil, and the last two the coordinates the position entries of that stencil. The algorithm is as follows:

```
Program Galerkin_Operator
StencilC = 0
n = 0
center = 1
!Compute A_coarse
do ii = 0 to 1
    do jj = 1 to -1 add -1
        do i = 0 to 1
           do j = 1 to -1 add -1
           n = n + 1
               if ((ii = 0 and jj = 0) or (i = 0 and j = 0) or
                (n = center)) then
                   StencilC(0,0) = StencilC(0,0) +
                     + S_t(j,i,-jj,ii) * R(j,i)
```

```
                else
                    StencilC(jj,ii) = StencilC(jj,ii) +
                      + S_t(-j,i,jj,ii) * R(-j,i)
            end do
        end do
    n = 0
    center = center + 1
    end do
end do
!Store stencil
StencilC = StencilC * Cf
```

Where Cf, is an scale constant to compensate the fact that the local prolongation and restriction do not satisfy the following inequation [32]:

$$m_p + m_r > m. \tag{7.10}$$

In this equation, m_p and m_r are the order of the prolongation and the restriction, respectively, that is, the highest degree plus one of polynomials that are interpolated exactly, and m is the order of the differential equation.

When the simple injection is selected, the Galerkin operator is not consistent and it must substituted by the following expression [33]

$$\mathbf{L}_{k-1} = \frac{1}{2}\mathbf{I}_k^{k-1}\mathbf{L}_k\mathbf{I}_{k-1}^k, \quad k = 1, \ldots, f \tag{7.11}$$

this modified version provides better multigrid convergence rate results over the original Galerkin operator.

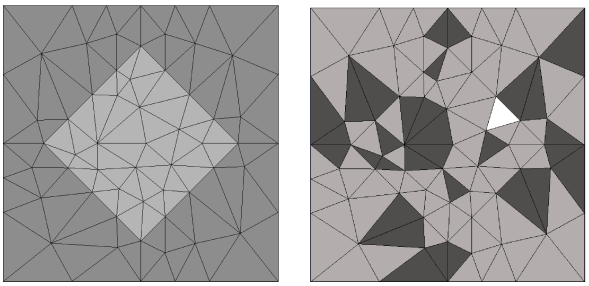
7.2 Numerical experiments

In this section we are going to present two numerical experiments to demonstrate the efficiency of the proposed block-wise multigrid algorithm based on Galerkin approach. In the first experiment, problem (7.1) is solved on the unit square with two different distributions of diffusion coefficients. This is a typical benchmark problem on this study area. In the second one, the same problem is solved on a more complex domain, simulating a more real situation. In both experiments the proposed multigrid algorithm is applied by using an F-cycle with two pre- and two post-smoothing steps.

7.2.1 Diffusion problem on the unit square with various diffusion coefficients

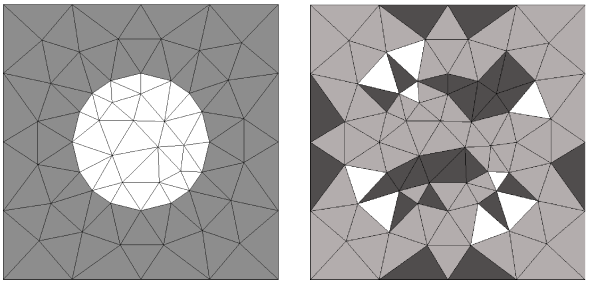
In the first experiment, we are going to solve problem (7.1) on the unit square with different diffusion coefficients defined piecewise on two subdomains of different shapes, as shown in Figures 7.3 and 7.4. This consists of a benchmark problem taken from the literature [32]. More concretely, in the first test case, the inner subdomain has a rhombus shape with a 0.5 side length, see Figure 7.3, whereas the second one consists of an hexadecagon with diameter 0.5 simulating a circle, see Figure 7.4. The right-hand side is defined as $f = xy$, the Dirichlet boundary conditions are given by $g = x^2+y^2$, and the diffusion coefficient values

are 0.333×10^5 for the internal subdomains and 2 for the rest of the square, see Figures 7.3 and 7.4. In the same figures the corresponding coarsest grids are also represented.



a) b)

Figure 7.3: a) Coarsest unstructured mesh for the first test case, and distribution of diffusion coefficients: 0.333×10^5 at the white region and 2 at the shaded part. b) Different smoothers for the triangles of the coarsest grid: white is Red-Black smoother, diamond smoother is represented by light-grey, and wormy smoother by dark-grey.



a) b)

Figure 7.4: a) Coarsest unstructured mesh for the second test case, and distribution of diffusion coefficients: 0.333×10^5 at the white region and 2 at the shaded part. b) Different smoothers for the triangles of the coarsest grid: white is Red-Black smoother, diamond smoother is represented by light-grey, and wormy smoother by dark-grey.

For these examples, Red-Black, Wormy and Diamond smoothers have been considered, along with some extra-relaxation process on Voronoi nodes close to the internal boundaries of the initial coarsest grid.

The proposed block-wise multigrid method has been applied to solve both test cases. Red-black, wormy and diamond smoothers have been used for different triangles of the coarsest grid, as shown in Figures 7.3 7.4. Regarding the obtained multigrid convergence, in Table 7.2.1, the number of iterations necessary to reduce the initial residual in a factor of 10^{-10} are shown for both test cases, together with the asymptotic convergence rates and the number of unknowns for each number of refinement levels. We observe an h-independent convergence for both problems, and although these results are slightly worse than those obtained in the case of constant diffusion coefficients, as expected,

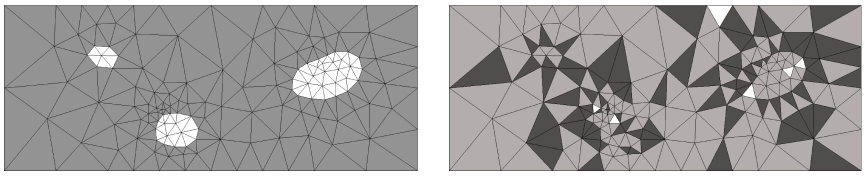
		Diamond		Circle	
Levels	Unknowns	Cycles	ρ_h	Cycles	ρ_h
4	6912	8	0.15	8	0.13
5	27648	8	0.18	9	0.17
6	110592	8	0.22	9	0.19
7	442368	8	0.24	9	0.20
8	1769472	9	0.25	9	0.21

Table 7.1: Number of iterations to reduce the residual in a factor of 10^{-10}, and asymptotic convergence rates for both test cases.

the method shows a very satisfactory convergence. On the other hand, when direct discretization is used on coarse grids, a very poor convergence rate is obtained.

7.2.2 Diffusion problem on a composite material domain

In the second experiment, problem (7.1) is solved on a rectangular domain composed of two different materials with different diffusion coefficients: 1 and 0.001, as we can see in Figure 7.5a).
The considered coarsest grid is also shown in the same figure, and also we can observe that it is composed of triangles with very disparate geometries. For this reason different smoothers are considered for the triangles of the coarsest triangulation. In particular the smoothers chosen for these triangles are shown in Figure 7.5b). In this way, the block-wise multigrid previously proposed is used for solving the problem. We want to comment that an extra-relaxation on Voronoi nodes close to the internal boundaries of the initial coarsest grid has been necessary.



a) b)

Figure 7.5: a) Diffusion coefficients for the second experiment, white color represents $\kappa = 0,001$ and grey $\kappa = 1$. b) Different used smoothers: white is red-black, diamond smoother is represented by light-grey and wormy smoother by dark-grey.

Now we are going to compare the behavior of the multigrid algorithm by considering both, direct discretization on coarse grids and the Galerkin approach. For this purpose, in Figure 7.6, the history of the convergence of the method, for two different refinement levels, to reach a final maximum residual below 10^{-7}, is displayed. In this case, the method based on direct discretization leads to divergence, while that based on Galerkin approach yields very satisfactory and robust results. Moreover, we observe that the convergence is independent of the discretization parameter, and with only twelve/thirteen cycles the residual

reaches the desired value. Note, that in this experiment the use of Galerkin coarse-grid operator is mandatory.

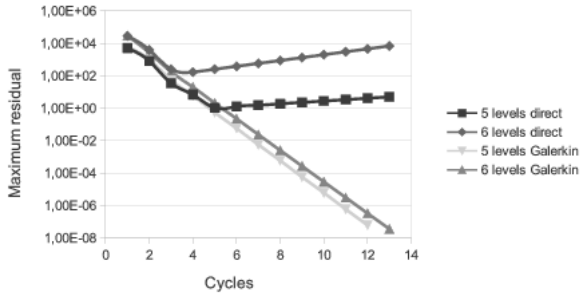


Figure 7.6: Comparison between direct discretization and Galerkin for two different refinement levels.

Chapter 8

Conclusions

In this work, the design of efficient multigrid methods for cell-centered finite volume schemes on semi-structured triangular grids has been the main focus. Due to the cell-centered character of the discretization and because of the application of the proposed strategy on semi-structured grids, very simple inter-grid transfer operators have been used in the design of these methods, leading to the requirement of stronger smoothers. Thus, appropriate novel smoothers are proposed depending on the geometry of the grid. Moreover, due to the semi-structured nature of the grid, different smoothers can be used on each structured region, giving rise to a block-wise multigrid method. The global behavior relies on its components on each block. Furthermore, the good behavior of the proposed multigrid method has been illustrated by some numerical experiments. A fast and h-independent convergence has been obtained, concluding that the adopted strategy yields very efficient solvers on relatively complex domains for cell-centered discretizations. Also, the proposed Galerkin operator gives good convergence rates even for high diffusion coefficient jumps, concluding that this methodology is efficient also for heterogeneous materials, which, in first instance are prohibitive. Moreover, the algorithm proposed is fast an efficient as it only use the collide stencils to get the Galerkin operator, making the user of this operator more plausible than having to multiply three matrices.

Bibliography

[1] B. Bergen, T. Gradl, F. Hülsemann, U. Ruede, *A massively parallel multigrid method for finite elements*, Comput. Sci. Eng., 8: 56–62, 2006.

[2] J. Bramble, R. Ewing, J. Pasciak, J. Shen, *The analysis of multigrid algorithms for cell centered finite difference methods*, Adv. Comput. Math., 5: 15-29, 1996.

[3] A. Brandt, *Multi–level adaptive solutions to boundary-value problems*, Math. Comput., 31: 333–390, 1977.

[4] J. Brandts, S. Korotov, M. Křížek, *The discrete maximum principle for linear simplicial finite element approximations of a reaction–diffusion problem*, Linear Algebra Appl., 429: 2344–2357, 2008.

[5] P.G. Ciarlet, *The finite element method for elliptic problems*, Series "Studies in Mathematics and its Applications", North-Holland, Amsterdam, 1978.

[6] P. Colella, E.G. Puckett, *Modern Numerical Methods for Fluid Flow*, Lecture Notes, Department of Mechanical Engineering, University of California, Berkeley, CA, 1994.

[7] H. Erten, A. Üngör, *Quality Triangulations with Locally Optimal Steiner Points*, SIAM J. Sci. Comput., 31: 2103–2130, 2009.

[8] H. Erten, A. Üngör, *Computing Triangulations without Small and Large Angles*, International Symposium on Voronoi Diagrams (ISVD) 2009.

[9] R. Eymard, T. Gallouët, R. Herbin, *Finite volume methods*, in: Handbook of Numerical Analysis, Vol.7, North Holland, Amsterdam, 2000, pp. 713–1020.

[10] F.J. Gaspar, J.L. Gracia, F.J. Lisbona, *Fourier analysis for multigrid methods on triangular grids*, SIAM J. Sci. Comput., 31: 2081–2102, 2009.

[11] W. Hackbusch, *Multi–grid methods and applications*, Springer, Berlin, 1985.

[12] M. Khalil, *Analysis of linear multigrid methods for elliptic differential equations with discontinuous and anisotropic coefficients.* PhD thesis, Technical University Delft, The Netherlands, 1989.

[13] M. Khalil, P. Wesseling, *A cell-centered multigrid method for three-dimensional anisotropic-diffusion and interface problems*, in: Mandel, J., McCormick, S.F. (eds.), Preliminary Proc. of the 4th Copper Mountain Conference on Multigrid Methods, Vol. 3, pp. 99-117, Denver, 1989. Computational Mathematics Group, Univ. of Colorado.

[14] M. Khalil, P. Wesseling, *Vertex-centered and cell-centered multigrid for interface problems*, in: Mandel, J., McCormick, S.F. (eds.), Preliminary Proc. of the 4th Copper Mountain Conference on Multigrid Methods, Vol. 3, pp. 61-97, Denver, 1989. Computational Mathematics Group, Univ. of Colorado.

[15] M. Khalil, P. Wesseling, *Vertex–centered and cell–centered multigrid for interface problems*, J. Comput. Phys., 98: 1-20, 1992.

[16] J. Karátson, S. Korotov, *Discrete maximum principles for finite element solutions of nonlinear elliptic problems with mixed boundary conditions*, Numer. Math. 99: 669-698, 2005.

[17] D.Y. Kwak, *V–cycle multigrid for cell centered finite differences*, SIAM J. Sci. Comput., 21: 552-564, 1999.

[18] D.Y. Kwak, J.S. Lee, *Multigrid algorithm for the cell centered finite difference method II: Discontinuous coefficient case*, Numer. Meth. Partial Differential Eqs., 20(5): 723-741, 2004.

[19] D.Y. Kwak, J.S. Lee, *Comparison of V–cycle Multigrid Method for Cell–Centered Finite Difference on Triangular Meshes*, Numer. Meth. Partial Differential Eqs., 22: 1080–1089, 2006.

[20] R.J. Leveque, *Finite volume methods for hyperbolic problems*, Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press. xix, 558, 2002.

[21] M. Mohr, R. Wienands, *Cell–centered multigrid revisited*, Comput. Visual. Sci., 7: 129–140, 2004.

[22] S.V. Patankar, D.B. Spalding, *Heat and mass transfer in boundary layers*, Morgan–Grampian, London, 1967.

[23] C. Rodrigo, F.J. Gaspar, C.W. Oosterlee, I. Yavneh, *Accuracy Measures and Fourier Analysis for the Full Multigrid Algorithm*, SIAM J. Sci. Comput., 32: 3108–3129, 2010.

[24] A.A. Samarskii, *Parabolic equations and difference methods for their solution*, in: Proc. of All–Union Conference on Differential Equations, 1958, Publishers of Armenian Ac.Sci., 1960, pp 148–160.

[25] J.R. Shewchuk, *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*, in "Applied Computational Geometry: Towards Geometric Engineering" (Ming C. Lin and Dinesh Manocha, editors), volume 1148 of Lecture Notes in Computer Science, pages 203–222, Springer-Verlag, Berlin, May 1996. (From the First ACM Workshop on Applied Computational Geometry.)

[26] J.R. Shewchuk, *Delaunay Refinement Algorithms for Triangular Mesh Generation*, Comput. Geom., 22: 21–74, 2002.

[27] U. Trottenberg, C.W. Oosterlee, A. Schüller, *Multigrid*, Academic Press, New York, 2001.

[28] C. Voronoi, *Nouvelles applications des paramètres continus à la théorie des forms quadratures*, J. Reine Angew Math., 134: 198–287, 1908.

[29] P. Wesseling, *Cell–centered multigrid for interface problems*, J. Comput. Phys. 79: 85-91, 1988.

[30] P. Wesseling, *Cell–centered multigrid for interface problems*, in: Mc–Cormick, S.F. (ed.), Multigrid Methods: Theory, Applications, and Supercomputing, Vol. 110 of Lecture Notes in Pure and Applied Mathematics. New York: Marcel Dekker 1988, pp. 631-641.

[31] P. Wesseling, *A multigrid method for elliptic equations with a discontinuous coefficient*, in: van der Burgh, A.H.P., Mattheij, R.M.M. (eds.), Proceedings of the First International Conference on Industrial and Applied Mathematics (ICIAM 87). Philadelphia: SIAM 1988, pp. 173–183.

[32] P. Wesseling, *An introduction to multigrid methods*, John Wiley, Chichester, UK, 1992.

[33] D.J. Mavriplis, V.Ventakatrishnan, *Agglomeration multigrid for viscous turbulent flows*, ICASE, 1994