NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/nla.1864

Multigrid methods for cell-centered discretizations on triangular meshes

P. Salinas*,†, C. Rodrigo, F. J. Gaspar and F. J. Lisbona

Instituto Universitario de Matemáticas y Aplicaciones IUMA, Applied Mathematics Department, University of Zaragoza, Zaragoza, Spain

SUMMARY

This paper deals with the design of efficient multigrid methods for cell-centered finite volume schemes on semi-structured triangular grids. Appropriate novel smoothers are proposed for this type of discretizations, depending on the geometry of the grid. Because of the semi-structured character of the mesh, on each structured patch, different smoothers can be considered. In this way, the multigrid method is constructed in a block-wise form, and its global behavior will rely on the components on each block. Numerical experiments are presented to illustrate the good behavior of the proposed multigrid method. Copyright © 2012 John Wiley & Sons, Ltd.

Received 13 January 2012; Revised 6 July 2012; Accepted 15 October 2012

KEY WORDS: multigrid; Voronoi meshes; cell-centered finite difference schemes; semi-structured grids

1. INTRODUCTION

In this paper, 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 [1–3]. Every Delaunay grid has an associated dual mesh, known as Voronoi tessellation [4], composed of 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 fur. 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 [5] 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 [6–8]. From the theoretical point of view, we refer the reader to Eymard *et al.* [9].

*Correspondence to: P. Salinas, Instituto Universitario de Matemáticas y Aplicaciones IUMA, Applied Mathematics Department, University of Zaragoza, Zaragoza, Spain.

†E-mail: salinascortes86@gmail.com

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 [10–12] 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 a 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 to obtain a 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. Although algebraic multigrid can also handle very easily cell-centered discretizations on unstructured grids, 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 carried out in rectangular grids. Historically, the pioneer work is due to Wesseling [13] 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 [14–20]. The W-cycle convergence of these multigrid methods, in the case of natural injection as prolongation, was theoretically analyzed by Bramble *et al.* [21], and in the case of V-cycle, the convergence was proved by Kwak *et al.* in [22] using certain weighted prolongation operators. On the other hand, on triangular grids, Kwak *et al.* [23, 24] 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 [25], requiring only a few stencils to represent the discrete operator for piece-wise constant coefficient problems, 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, to overcome the weakness of the coarse-grid correction, the design of powerful smoothers is mandatory, because standard relaxation does not provide satisfactory results in most of the cases. Then, here different novel smoothers suitable for the considered cell-centered discretization are introduced. These smoothers are used on each input block of the initial unstructured grid, depending on its geometry, to overcome possible anisotropies of the arising Voronoi mesh. Notice that these smoothers could not be implemented on a pure unstructured grid, which involves an advantage of the geometric multigrid method presented here against the algebraic multigrid approach. We are speaking about, for example, the line-type smoothers, very necessary in the case of anisotropic problems.

The rest of the paper is organized as follows. Section 2 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 Section 3. In particular, special smoothers are designed to overcome troubles arising from some difficult grid-geometries. In Section 4, 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. Also in this section, the way of choosing the best strategy to follow on a particular triangular grid is explained. Finally, in Section 5, some test examples on semi-structured grids are presented to show the applicability of the strategies followed throughout all the work, and some summing up is made in Section 6.

2. DISCRETIZATION OF A DIFFUSION PROBLEM ON VORONOI GRIDS

In this section, the finite volume discretization on Voronoi meshes of a diffusion problem is presented. Such discretization is introduced for unstructured as well as for structured triangular meshes.

DOI: 10.1002/nla

However, we want to remark that in Section 5, a convection-diffusion problem will be considered, and its corresponding discretization will be obtained following the same steps as for the pure diffusion problem, as we will specify at that point.

2.1. 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:

$$-\Delta v = f, \text{ in } \Omega, \tag{1}$$

$$v = 0, \text{ on } \partial\Omega. \tag{2}$$

Firstly, we suppose to have a Delaunay triangulation $\mathcal{T}$ on the domain Ω, satisfying the usual admissibility assumption [2], that is, the intersection of two different triangles is empty, a vertex, or a whole edge. Besides, as we commented in Section 1, we restrict ourselves to an acute triangulation.

The grid points associated with the cell-centered scheme are the centers of the circumscribed circle of each triangle, defining a Voronoi mesh (Figure 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. Moreover, 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:

$$-\int_{l_1} \nabla v \cdot \mathbf{n}_1 \, \mathrm{d}l_1 - \int_{l_2} \nabla v \cdot \mathbf{n}_2 \, \mathrm{d}l_2 - \int_{l_3} \nabla v \cdot \mathbf{n}_3 \, \mathrm{d}l_3 = \int_T f(\mathbf{x}) \, \mathrm{d}x, \tag{3}$$

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 by using the Voronoi points. With regard to the integral in the right-hand side, we consider the following approximation:

$$\int_T f(\mathbf{x}) \, \mathrm{d}x \approx \text{meas}(T) \, f(\mathbf{x}_c), \tag{4}$$

where $\mathbf{x}_c$ denotes the Voronoi point of triangle T, and where meas(T) is the area of T.

Denoting $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$ as 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 $\mathbf{x}_i$, $d_i = \text{dist}(\mathbf{x}_c, \mathbf{x}_i)$, with $i = 1, 2, 3$ (Figure 2), we finally obtain the equation corresponding to node $\mathbf{x}_c$:

$$-\frac{1}{\text{meas}(T)} \sum_{i=1}^{3} \left(\text{meas}(l_i) \frac{v_h(\mathbf{x}_i) - v_h(\mathbf{x}_c)}{d_i} \right) = f(\mathbf{x}_c), \tag{5}$$

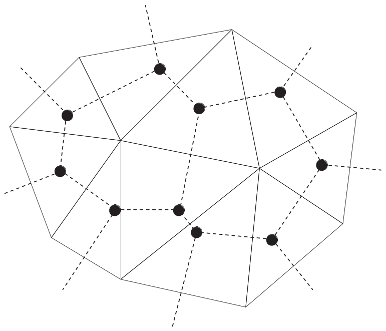


Figure 1. Unstructured mesh and its associated Voronoi grid.

DOI: 10.1002/nla

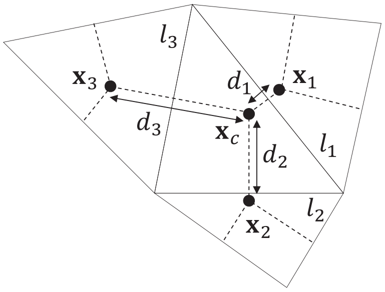


Figure 2. Notation for neighboring Voronoi points on an unstructured grid.

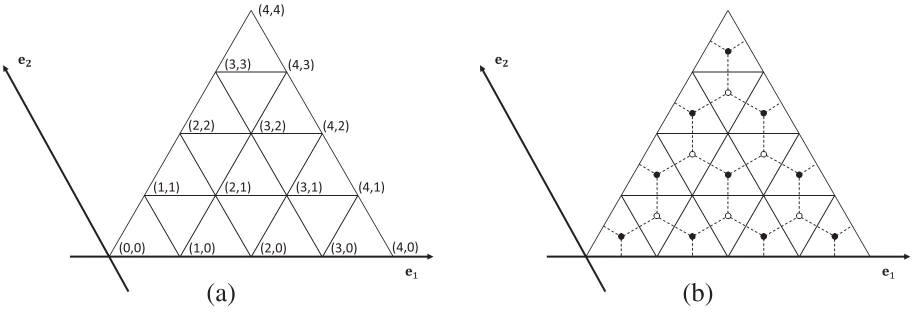
where v_h is the grid function approximating the solution of the continuous problem on the Voronoi points. This scheme is first-order accurate in a discrete H^1-norm, whereas it is computationally observed that second order is achieved in L^2-norm. The theoretical analysis of this approach can be seen in [26–28].

2.2. *Discretization on triangular structured grids*

Now, we are going to consider the particular case of the discretization of problem (1) on a 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 a 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. Hence, a local numeration can be fixed according 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 performed 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:

$$G_\ell = \left\{ \mathbf{x} = k_1 h_1 \,\mathbf{e}_1 + k_2 h_2 \,\mathbf{e}_2 \,|\, k_1 = 0, \ldots, 2^\ell,\; k_2 = 0, \ldots, k_1 \right\}, \tag{6}$$

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), $k_1 = 0, \ldots, 2^\ell$,



(a)

(b)

Figure 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.

DOI: 10.1002/nla

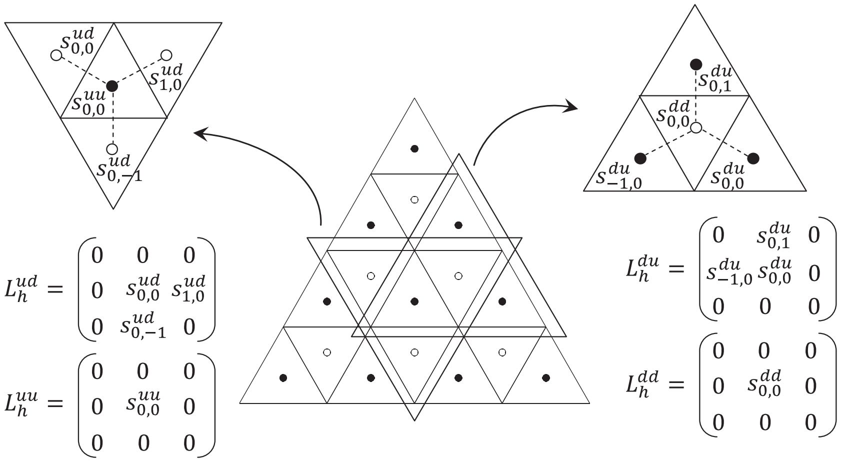
$k_2 = 0, \dots, k_1$, is used in such a way that the indexes of the vertices of the triangle are $(0,0)$, $(2^\ell, 0)$, $(2^\ell, 2^\ell)$, as it can also be observed in Figure 3(a) for $\ell = 2$.

On the other hand, the considered discretization is based on the dual Voronoi mesh, represented in Figure 3(b). In the particular case in which a structured grid as considered here is used, the obtained finite difference scheme becomes 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' (Figure 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(b). These sub-grids can be defined from the grid G_h, in the following way:

$$V_h^u = \left\{ \mathbf{x}^u_{k_1,k_2} = (k_1 + \delta_1) h_1\, \mathbf{e}_1 + (k_2 + \delta_2) h_2\, \mathbf{e}_2 \mid k_1 h_1\, \mathbf{e}_1 + k_2 h_2\, \mathbf{e}_2 \in G_h \right\}, \tag{7}$$

$$V_h^d = \left\{ \mathbf{x}^d_{k_1,k_2} = \left(k_1 + \delta_1'\right) h_1\, \mathbf{e}_1 + \left(k_2 + \delta_2'\right) h_2\, \mathbf{e}_2 \mid k_1 h_1\, \mathbf{e}_1 + k_2 h_2\, \mathbf{e}_2 \in G_h \right\}, \tag{8}$$

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 (Figure 5). 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



$$L_h^{ud} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & s^{ud}_{0,0} & s^{ud}_{1,0} \\ 0 & s^{ud}_{0,-1} & 0 \end{pmatrix} \qquad L_h^{du} = \begin{pmatrix} 0 & s^{du}_{0,1} & 0 \\ s^{du}_{-1,0} & s^{du}_{0,0} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$L_h^{uu} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & s^{uu}_{0,0} & 0 \\ 0 & 0 & 0 \end{pmatrix} \qquad L_h^{dd} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & s^{dd}_{0,0} & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Figure 4. Stencils corresponding to two different grid points: one associated with an up-oriented triangle and the other with a down-oriented triangle.

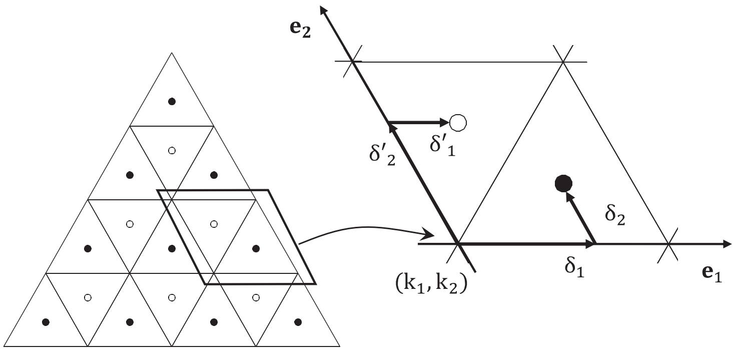


Figure 5. Voronoi mesh split into two sub-grids, and corresponding values of δ_i and δ_i'.

DOI: 10.1002/nla

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

$$L_h^{uu} v_h^u \left(\mathbf{x}^u_{k_1,k_2}\right) + L_h^{ud} v_h^d \left(\mathbf{x}^d_{k_1,k_2}\right) = f_h^u \left(\mathbf{x}^u_{k_1,k_2}\right), \tag{9}$$

$$L_h^{du} v_h^u \left(\mathbf{x}^u_{k_1,k_2}\right) + L_h^{dd} v_h^d \left(\mathbf{x}^d_{k_1,k_2}\right) = f_h^d \left(\mathbf{x}^d_{k_1,k_2}\right), \tag{10}$$

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

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

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

2.2.1. Stencil dependence on two angles characterizing the triangular grid. An important feature to take into account in a stencil is the strength of the connections between the involved unknowns. Each entry of the stencil defines the coefficient of the corresponding unknown in the equation associated with the central point. Then, if some coefficient is large relative to the other coefficients in the same equation, then a small change in the value of the associated unknown has more effect on the value of the unknown at the central point, and therefore, we will say that these both unknowns are strongly connected [29].

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 as a 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 (Figure 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 a Voronoi grid point, associated with an arbitrary down-oriented triangle $\mathbf{x}^d_{k_1,k_2}$ in V_h^d. As 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, the coordinates of the points involved in

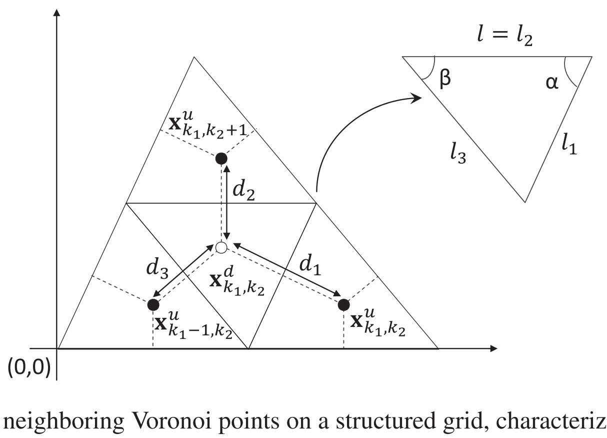


Figure 6. Notation for neighboring Voronoi points on a structured grid, characterized by angles α and β.

such stencil can be computed in the Cartesian coordinate system with respect to the origin (Figure 6). And, in terms of the previously explained geometric parameters, they result in the following:

$$\mathbf{x}^d_{k_1,k_2} = \frac{l}{2}\left(\frac{2\cos\alpha\sin\beta}{\sin(\alpha+\beta)}+1, \frac{2\sin\alpha\sin\beta}{\sin(\alpha+\beta)}+\cot(\alpha+\beta)\right), \tag{11}$$

$$\mathbf{x}^u_{k_1,k_2} = \frac{l}{2}\,(3, -\cot(\alpha+\beta)), \tag{12}$$

$$\mathbf{x}^u_{k_1-1,k_2} = \frac{l}{2}\,(1, -\cot(\alpha+\beta)), \tag{13}$$

$$\mathbf{x}^u_{k_1,k_2+1} = \frac{l}{2}\left(\frac{2\cos\alpha\sin\beta}{\sin(\alpha+\beta)}+1, \frac{2\sin\alpha\sin\beta}{\sin(\alpha+\beta)}-\cot(\alpha+\beta)\right). \tag{14}$$

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

$$\text{meas}(T) = \frac{l^2\sin\alpha\sin\beta}{2\sin(\alpha+\beta)}, \tag{15}$$

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 computing d_i $i = 1,2,3$ from (11)–(14), we can finally obtain the stencils:

$$L_h^{du} = \frac{2\sin(\alpha+\beta)}{l^2\sin\alpha\sin\beta}\begin{bmatrix} 0 & \tan(\alpha+\beta) & 0 \\ -\tan\alpha & -\tan\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{16}$$

$$L_h^{dd} = \frac{2\sin(\alpha+\beta)}{l^2\sin\alpha\sin\beta}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -\tan(\alpha+\beta)+\tan\alpha+\tan\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{17}$$

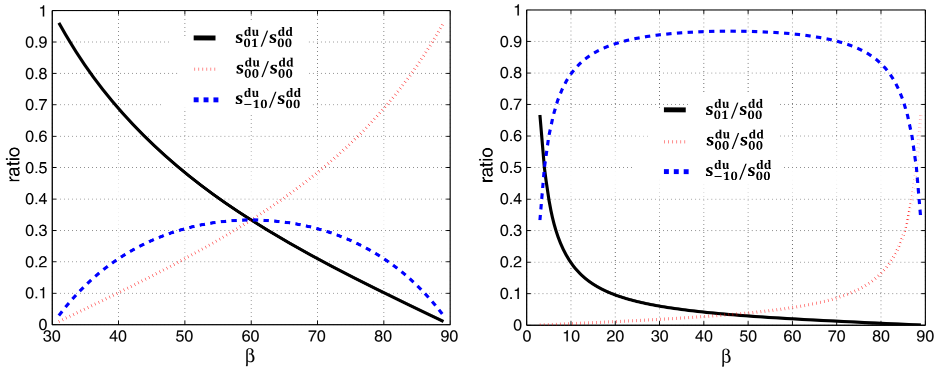
As previously commented, for a Voronoi grid point, associated with an arbitrary up-oriented triangle, $\mathbf{x}^u_{k_1,k_2}$ in V_h^u the stencil would be the 'mirror image stencil' of (16) and (17), that is,

$$L_h^{uu} = \frac{2\sin(\alpha+\beta)}{l^2\sin\alpha\sin\beta}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -\tan(\alpha+\beta)+\tan\alpha+\tan\beta & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{18}$$

$$L_h^{ud} = \frac{2\sin(\alpha+\beta)}{l^2\sin\alpha\sin\beta}\begin{bmatrix} 0 & 0 & 0 \\ 0 & -\tan\beta & -\tan\alpha \\ 0 & \tan(\alpha+\beta) & 0 \end{bmatrix}. \tag{19}$$

Notice that depending on the angles characterizing the grid, some strong connections appear between some unknowns. In particular, in Figure 7, we show the variability of the stencil weights, for a Voronoi point $\mathbf{x}^d_{k_1,k_2}$ associated with a down-oriented triangle, depending on one of the angles of the triangulation. More concretely, we display the ratios between the three extra-diagonal coefficients in the equation and the central term, that is, s^{du}_{00}/s^{dd}_{00}, s^{du}_{01}/s^{dd}_{00}, and s^{du}_{-10}/s^{dd}_{00}, for two different situations in which angle α is fixed and angle β varies among all its possible values satisfying the requirement of an acute triangulation. In Figure 7(a), α is 60°, and we can see that when $\beta = 60°$, that is for an equilateral triangle, the strength of the three connections in the stencil is the same. When angle β is small or large enough, we observe that two of the connections are weak, whereas the other one is very strong. On the other hand, in Figure 7(b), α is fixed as 88°. We can see that for almost the whole range of values of β, one of the connections in the stencil is strong and the other ones are much weaker, whereas when β is either very small or very large, then two of the connections in the stencil are of equal strength. These strong connections between the unknowns involved in a stencil will have to be taken into account in the design of the smoothers in a geometric multigrid method, as will be discussed in next section.

DOI: 10.1002/nla



(a) (b)

Figure 7. Variability of the ratios between the coefficients of a stencil associated with triangles with (a) $\alpha = 60^\circ$ or (b) $\alpha = 88^\circ$, and β varying among all its possible values satisfying the requirement of an acute triangulation.

3. 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 section, 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 carried out 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 performed 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 paper.

3.1. *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. Regarding the coarse-grid approximation, a direct PDE discretization on the coarse grids has been used.

When vertex-centered discretizations are considered on triangular grids, grid points lying 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 becomes non-nested (Figure 8). 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

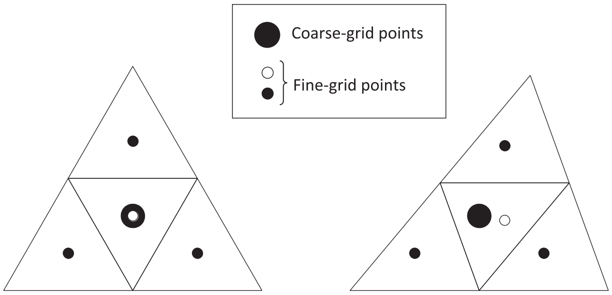


Figure 8. Nested (left) and non-nested (right) hierarchies.

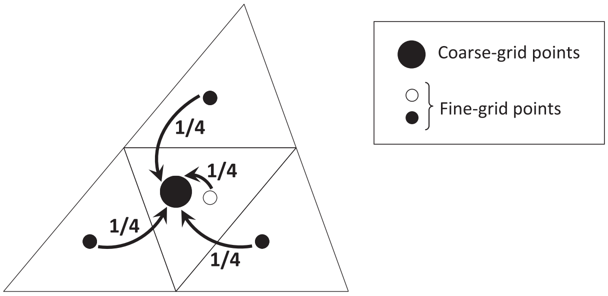


Figure 9. Restriction operator.

inter-grid operators, because 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:

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

as shown in Figure 9, and the corresponding prolongation fulfills $\mathbf{I}_{2h}^h = 4\,\mathbf{I}_h^{2h}$.

The choice of these inter-grid transfer operators leads us to make an effort in the 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.

3.2. Smoothers

The smoother usually plays an important role in multigrid algorithms, mainly in the geometric approach. The choice of a suitable smoother is an important feature to guarantee the efficiency of these methods. Moreover, as previously commented, in the framework we are working with, this choice is even more relevant. Because of the general observation that errors become smooth if strongly connected unknowns are collectively updated [12, 30], appropriate smoothers have been designed depending on the magnitude of the coefficients of the stencils, given in (16)–(19). The following smoothers have been considered and tested in order to fulfill the previous requirement.

DOI: 10.1002/nla

3.2.1. 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 the order in which the grid points are visited makes no difference, which makes the 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 and non-updated unknowns before the current step [12]. 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

$$\left(L_h^{uu}\right)^+ = L_h^{uu}, \quad \left(L_h^{dd}\right)^+ = L_h^{dd}. \tag{20}$$

3.2.2. Red-black smoother. Because of the fact that unknowns related to up-oriented 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. As 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 applying 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

$$\left(L_h^{uu}\right)^+ = L_h^{uu}, \quad \left(L_h^{dd}\right)^+ = I_h, \tag{21}$$

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

3.2.3. Diamond smoothers. For almost-right triangles, a strong connection between only two nodes involved in the stencil appears, because of the anisotropy of the Voronoi mesh. Therefore, because the common lore claims that smoothing must be performed 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 10(a). 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 is associated with that vertex associated with β, and red color corresponds to vertex of angle $180^\circ - (\alpha + \beta)$. In this manner, each diamond smoother can be named with the color corresponding to the vertex opposite to the edge across which the strong coupling appears. Following this rule, diamond smoother appearing in Figure 10(a) is called green-diamond 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 performed analogously. Then, in order to do this description, we consider the green-diamond

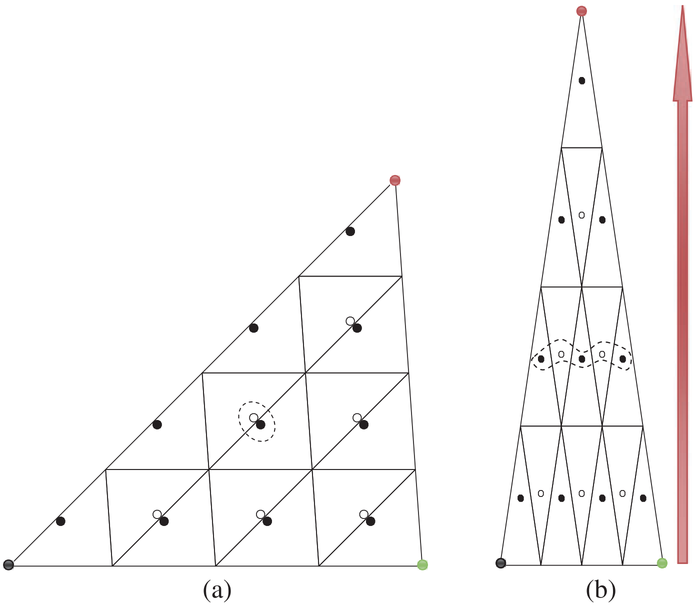


Figure 10. (a) Green-diamond smoother and (b) red-wormy smoother.

smoother, which can be seen in Figure 10(a). The positive parts of the involved scalar operators are given as follows:

$$\left(L_h^{uu}\right)^+ = L_h^{uu},\ \left(L_h^{ud}\right)^+ = \begin{bmatrix} 0 & 0 & 0 \\ 0 & s_{0,0}^{ud} & 0 \\ 0 & s_{0,-1}^{ud} & 0 \end{bmatrix},\ \left(L_h^{du}\right)^+ = \begin{bmatrix} 0 & 0 & 0 \\ s_{-1,0}^{du} & s_{0,0}^{du} & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \left(L_h^{dd}\right)^+ = L_h^{dd}. \tag{22}$$

3.2.4. Wormy smoothers. When a very small angle characterizes the triangulation, the strong coupling appears between the Voronoi points associated with the triangles situated in the direction of the edge opposite to the vertex corresponding to this small angle (Figure 10(b)). Therefore, all those points are simultaneously updated, and a tridiagonal system must be solved for each of these ‘wormy-lines’. For this reason, this smoother will be called wormy smoother. Similar 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, and they can be named with the color corresponding to the vertex opposite to such edge. Following this criterion, wormy smoother appearing in Figure 10(b) is called red-wormy smoother. Notice that these smoothers can be performed by visiting the ‘lines’ in a lexicographic order, from vertex to edge or, on the contrary, from edge to vertex, and this latest is the chosen option, as is represented by the arrow in Figure 10(b). 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 10(b), is considered. In this way, the positive parts of the scalar operators result as follows:

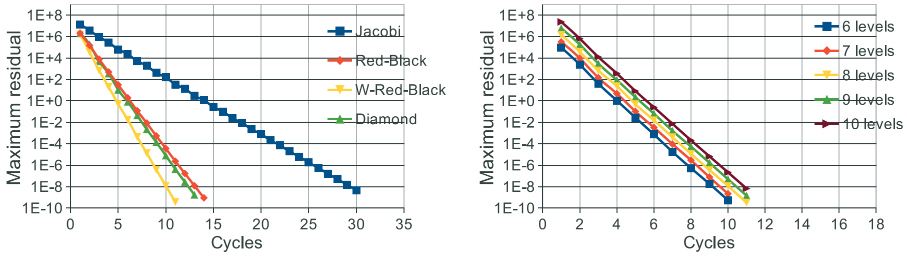
$$\left(L_h^{uu}\right)^+ = L_h^{uu},\ \left(L_h^{ud}\right)^+ = \begin{bmatrix} 0 & 0 & 0 \\ 0 & s_{0,0}^{ud} & s_{1,0}^{ud} \\ 0 & s_{0,-1}^{ud} & 0 \end{bmatrix},\ \left(L_h^{du}\right)^+ = \begin{bmatrix} 0 & 0 & 0 \\ s_{-1,0}^{du} & s_{0,0}^{du} & 0 \\ 0 & 0 & 0 \end{bmatrix},\ \left(L_h^{dd}\right)^+ = L_h^{dd}. \tag{23}$$

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 as we will see in the next section; any possible triangulation will have associated a wormy smoother giving a satisfactory convergence factor.

4. 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 (1)–(2) is solved in the structured grid arising from the regular refinement of a triangular domain, characterized by two of their angles. Unless stated otherwise, homogeneous Dirichlet boundary conditions, a random initial guess, and a zero right-hand side are considered in the numerical experiments to avoid round-off errors. Initially, an F-cycle is considered because it is cheaper than a W-cycle, and more robust than a simple V-cycle [12]. However, in Section 5, its behavior will be compared with that of a V-cycle. Regarding the number of smoothing steps, we have fixed two pre-smoothing and two post-smoothing steps. All the numerical experiments have been performed in an *Intel Nehalem* (Intel, Santa Clara, CA) 1.66 GHz.

We begin by considering an equilateral triangular domain. As any anisotropy arises from the grid resulting from a regular refinement, it could be seen natural to think in applying a simple point-wise smoother, such as Jacobi or lexicographic Gauss–Seidel, to this kind of triangulations. However, as we previously commented, for this situation, a pattern relaxation scheme could be more appropriate. Thus, we are going to present some convergence results on a regular equilateral grid, comparing the behavior of multigrid by considering the following: undamped Jacobi smoother, red-black smoother, ω-red-black smoother (with $\omega = 1.15$), and diamond smoother. In Figure 11(a), 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 [31] and in the context of full-multigrid on rectangular grids [32]), 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 [33] for cell-centered discretizations on rectangular grids, where it was validated by a local Fourier analysis. Moreover, the good behavior of the multigrid based on the ω-red-black smoother is confirmed in Figure 11(b), 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 ω-red-black smoother seems to be a good choice for almost-equilateral triangulations. However, this good behavior deteriorates quickly when the shape of the triangle tends to be rectangular or is characterized by a very small angle.



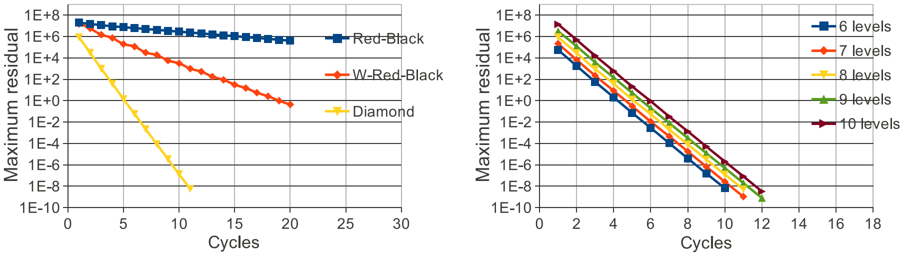
(a)

(b)

Figure 11. (a) Comparison of smoothers on an equilateral triangular grid. (b) History of the convergence for different numbers of refinement levels by using ω-red-black smoother.

In the case of almost-right triangles, point-wise smoothers are not suitable anymore because of the anisotropy of the Voronoi mesh. On the other hand, diamond smoother results in a very efficient smoother when this kind of grids is considered. As an example, a triangular domain characterized by angles $\alpha = 45°$ and $\beta = 85°$ is fixed. In Figure 12(a), the history of the multigrid convergence by using different smoothers is displayed. More concretely, red-black, ω-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. 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 11 cycles. Besides, in Figure 12(b), 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 13, where asymptotic convergence factors of the diamond smoother-based multigrid are shown for a wide range of pairs of angles (α, β) characterizing the grid.

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° is considered as domain of our problem. With this purpose, in Figure 14(a), the multigrid convergence provided by using ω-red-black, diamond,



(a) (b)

Figure 12. (a) Comparison of smoothers on an almost-right triangular grid. (b) History of the convergence for different numbers of refinement levels by using diamond smoother.

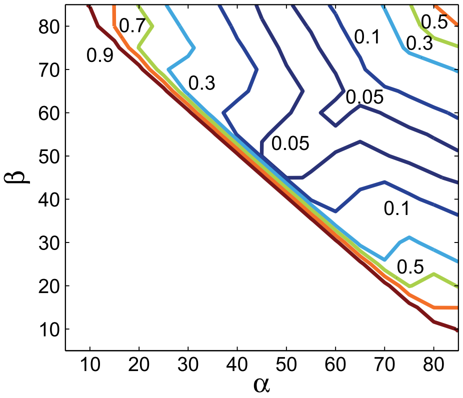


Figure 13. Experimentally computed convergence factors for the diamond smoother based multigrid and four smoothing steps, for different triangles in function of two of their angles.

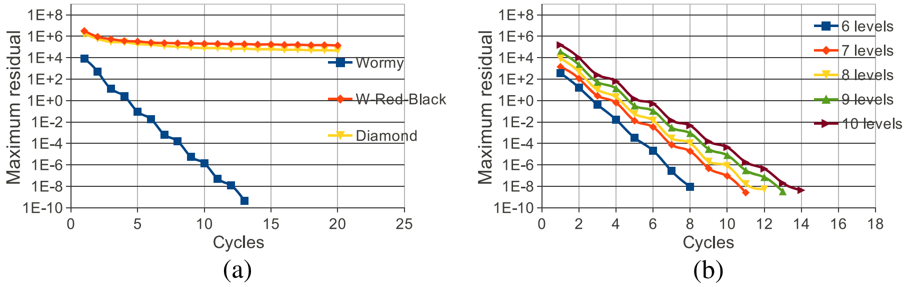


Figure 14. (a) Comparison of smoothers on an isosceles triangular grid with smallest angle 10°. (b) History of the convergence for different numbers of refinement levels by using wormy smoother.

Table I. Number of cycles necessary to reduce the initial residual in a factor of 10^{-10}, and the corresponding CPU times, by using an F-cycle.

		Equilateral (ω-red-black)		Right triangle (diamond smoother)		Sharp triangle (wormy smoother)	
Levels	Unknowns	Cycles	Time (s)	Cycles	Time (s)	Cycles	Time (s)
6	4096	7	0.03	8	0.03	6	0.04
7	16,384	7	0.09	9	0.10	6	0.13
8	65,536	7	0.35	9	0.39	7	0.58
9	262,144	7	1.35	9	1.55	7	2.28
10	1,048,576	7	5.76	9	6.81	7	9.66

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 14(b).

From the results presented in this section, it seems that a reasonable strategy to follow would be to apply the point-wise ω-red-black smoother for almost-equilateral triangles, the collective diamond smoother for almost-right triangles, and finally the appropriate block collective wormy smoother when triangulations with a small angle appear. To summarize, in Table I, we show for the three considered representative triangles the results corresponding to the best smoother for each geometry. In particular, for different numbers of refinement levels, the number of cycles necessary to reduce the initial residual in a factor of 10^{-10}, and the CPU time are shown. As we can observe, a convergence independent on the number of unknowns is obtained in all the three cases.

From a practical point of view, for any given triangular geometry, it 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 be used for each triangle of the input grid to achieve a local desired convergence factor on this triangle, and in this way, to reach globally such convergence factor, because the quality of the general algorithm will depend on the local results obtained for each coarse triangle. 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 angles of the triangle. That is, taking into account that wormy smoother is about twice expensive than ω-red-black smoother, and diamond smoother is about 10% more expensive than the latter, we chose for each triangle the cheapest smoother that gives a convergence factor below a desired one. Following this strategy, the corresponding guideline to reach a global convergence factor about 0.1 is shown in Figure 15. In the future, the possibility of validating these results by performing a special local Fourier analysis for cell-centered discretizations as those considered here, is very appealing because it would make possible a deeper analysis of the influence of the geometry on the multigrid convergence. However, this will be the purpose of a forthcoming paper, which is out of the scope of this work.

DOI: 10.1002/nla

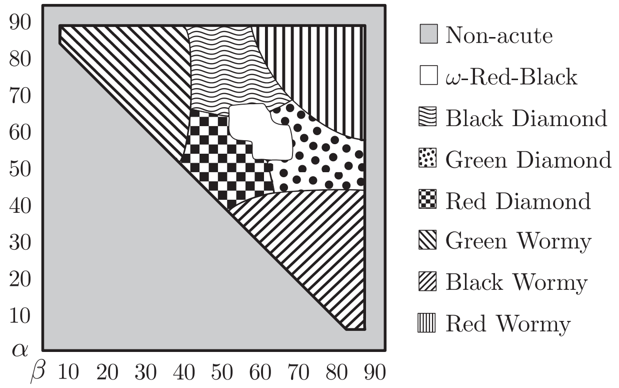


Figure 15. Guideline to choose suitable smoothers to reach an asymptotic convergence factor about 0.1 on different triangles.

5. NUMERICAL EXPERIMENTS ON SEMI-STRUCTURED GRIDS

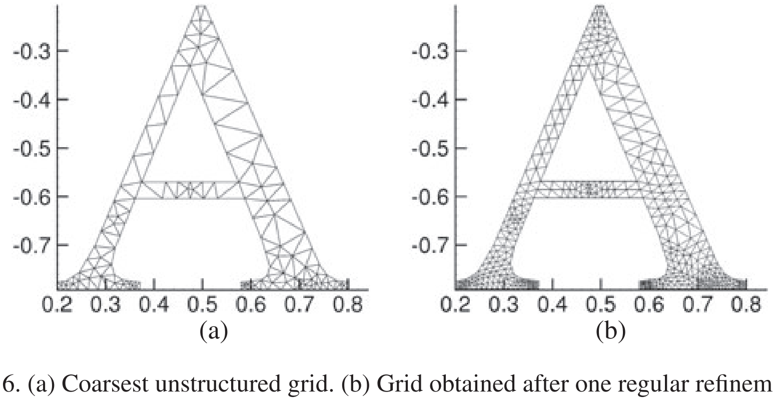
In this section, we consider two model problems: a Laplace problem in an A-shaped domain and a convection-diffusion problem in a square domain. The strategy shown in Figure 15 has been applied to perform the numerical experiments. In both problems, *aCute* software [34, 35], which is based on *Triangle* [36, 37], has been used to generate the initial unstructured acute Delaunay triangulation.

5.1. Laplace problem in an A-shaped domain

In the first numerical experiment, model problem (1) is solved in an A-shaped domain, as shown in Figure 16. To this purpose, an initial unstructured grid composed of 201 triangles is considered, as depicted in Figure 16(a). 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 16(b), as an example.

Following the guideline displayed in Figure 15, we have chosen the most efficient smoother for each triangle of the input unstructured triangulation. Selected smoothers can be seen in Figure 17. 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 $F(2,2)$-cycle, the number of cycles necessary to reduce the initial residual in a factor of 10^{-10} and the CPU time, together with the asymptotic convergence factor, are shown in Table II for different numbers of refinement levels. 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 0.12 is obtained. In order to know the behavior of the V-cycle, in the same table, we have also added the



(a)
(b)

Figure 16. (a) Coarsest unstructured grid. (b) Grid obtained after one regular refinement level.

DOI: 10.1002/nla

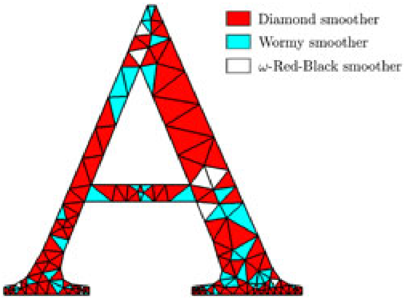


Figure 17. Different smoothers for the triangles composing the initial triangulation of the A-shaped domain.

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

		$V(2,2)$			$F(2,2)$		
Levels	Unknowns	Cycles	ρ_h	Time (s)	Cycles	ρ_h	Time (s)
4	51,456	9	0.16	1.78	8	0.09	3.27
5	205,824	9	0.19	3.62	8	0.11	6.32
6	823,296	10	0.21	10.16	9	0.11	15.96
7	3,293,184	11	0.23	34.69	9	0.12	43.66
8	13,172,736	11	0.26	123.65	9	0.12	142.52

results corresponding to a $V(2,2)$-cycle, following the same strategy than for the F-cycle, that is, the guideline shown in Figure 15. From these results, we can observe a slight deterioration in the asymptotic convergence factors with respect to the number of unknowns. This degradation of the convergence may be due to the fact that the accuracy of the transfer operators is insufficient to fulfill the well-known rule of thumb [30, 38],

$$m_{\mathrm{P}} + m_{\mathrm{R}} > m,$$

where m_{P} and m_{R} are defined as the highest degree plus one of the polynomials that are interpolated exactly by the prolongation and the restriction, respectively; and m is the order of the partial differential equation to be solved. Here, injection is considered, and then, m_{P} and m_{R} are both equal to one. As the order of the equation is 2, the strict inequality is not satisfied. Thus, it is not possible to guarantee efficient convergence rates because of the weakness of the coarse-grid correction. Anyway, from the practical point of view, it could be interesting the use of V-cycles for solving this type of problems, because in very few iterations, the desired convergence is achieved.

5.2. *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 form as follows:

$$-\nabla \cdot (\nabla v + \mathbf{b}\, v) = f, \text{ in } \Omega, \tag{24}$$

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 Section 2, by using a central difference scheme to approximate the convective term [27]. In this numerical experiment, a square domain of unit length and Dirichlet

DOI: 10.1002/nla

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:

$$-\frac{1}{\text{meas}(T)} \sum_{i=1}^{3} \left(\text{meas}(l_i) \left(\frac{v_h(\mathbf{x}_i) - v_h(\mathbf{x}_c)}{d_i} + \mathbf{b} \cdot \mathbf{n}_i \frac{v_h(\mathbf{x}_i) + v_h(\mathbf{x}_c)}{2} \right) \right) = f(\mathbf{x}_c). \qquad (25)$$

We consider an initial unstructured grid, composed of 96 triangles, as seen in Figure 18, 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 15 to choose the suitable local smoother on each input triangle, and this selection is displayed in Figure 19. The proposed geometric multigrid method is applied to solve the corresponding large sparse linear system of equations. First, an $F(2, 2)$-cycle is used to test the independency of the multigrid convergence with regard to the discretization parameters. In Table III, 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 together with the CPU time. Similar to the previous numerical experiment, the results corresponding to apply a $V(2, 2)$-cycle are also shown in this table. Again, a deterioration of the asymptotic convergence factor is observed, but the application of this cycle can be interesting because of the small number of iterations necessary to reach the convergence.

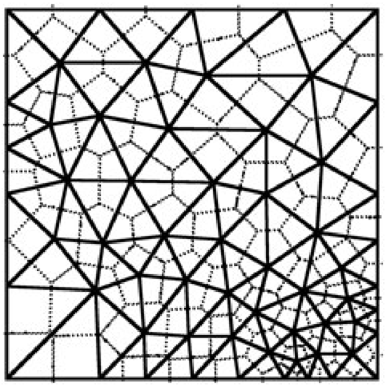


Figure 18. Coarsest unstructured grid together with the associated Voronoi mesh.

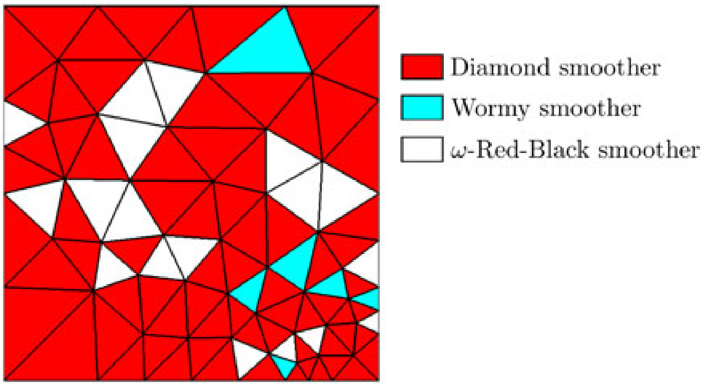


Figure 19. Different smoothers considered on each triangular block of the input grid.

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

		$V(2,2)$			$F(2,2)$		
Levels	Unknowns	Cycles	ρ_h	Time (s)	Cycles	ρ_h	Time (s)
4	24,576	9	0.19	0.74	8	0.09	1.19
5	98,304	10	0.20	1.76	8	0.09	2.54
6	393,216	10	0.21	4.54	8	0.09	6.07
7	1,572,864	11	0.22	15.53	8	0.09	17.11
8	6,291,456	11	0.22	55.16	8	0.09	56.33

6. 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. Because of 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, because of 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.

ACKNOWLEDGEMENTS

The authors thank the referees for their valuable comments, which permitted us to improve the original paper. This research has been partially supported by FEDER/MCYT Projects MTM 2010-16917 and the DGA (Grupo consolidado PDIE).

REFERENCES

1. Brandts J, Korotov S, Křížek M. The discrete maximum principle for linear simplicial finite element approximations of a reaction–diffusion problem. *Linear Algebra and its Applications* 2008; **429**:2344–2357.
2. Ciarlet PG. *The Finite Element Method for Elliptic Problems*, Series "Studies in Mathematics and its Applications". North-Holland: Amsterdam, 1978.
3. Karátson J, Korotov S. Discrete maximum principles for finite element solutions of nonlinear elliptic problems with mixed boundary conditions. *Numerische Mathematik* 2005; **99**:669–698.
4. Voronoi C. Nouvelles applications des paramètres continus à la théorie des forms quadratures. *Journal für die Reine und Angewandte Mathematik* 1908; **134**:198–287.
5. Samarskii AA. Parabolic equations and difference methods for their solution. In *Proceedings of All-Union Conference on Differential Equations, 1958*. Publishers of Armenian Ac. Sci., 1960; 148–160. (in Russian).
6. Colella P, Puckett EG. *Modern numerical methods for fluid flow*, Lecture Notes. Department of Mechanical Engineering, University of California: Berkeley, CA, 1994.
7. Leveque RJ. *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Vol. xix. Cambridge University Press: Cambridge, 2002.
8. Patankar SV, Spalding DB. *Heat and Mass Transfer in Boundary Layers*. Morgan–Grampian: London, 1967.
9. Eymard R, Gallouët T, Herbin R. Finite volume methods. In *Handbook of Numerical Analysis*, Vol. 7. North Holland: Amsterdam, 2000; 713–1020.
10. Brandt A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation* 1977; **31**:333–390.
11. Hackbusch W. *Multi-Grid Methods and Applications*. Springer: Berlin, 1985.
12. Trottenberg U, Oosterlee CW, Schüller A. *Multigrid*. Academic Press: New York, 2001.

DOI: 10.1002/nla

13. Wesseling P. A multigrid method for elliptic equations with a discontinuous coefficient. In *Proceedings of the First International Conference on Industrial and Applied Mathematics (ICIAM 87)*, van der Burgh AHP, Mattheij RMM (eds). SIAM: Philadelphia, 1988; 173–183.
14. Khalil M. Analysis of linear multigrid methods for elliptic differential equations with discontinuous and anisotropic coefficients. *PhD Thesis*, Technical University Delft, The Netherlands, 1989.
15. Khalil M, Wesseling P. A cell-centered multigrid method for three-dimensional anisotropic-diffusion and interface problems. In *Preliminary Proceedings of the 4th Copper Mountain Conference on Multigrid Methods*, Vol. 3, Mandel J, McCormick SF (eds). Computational Mathematics Group, Univ. of Colorado: Denver, 1989; 99–117.
16. Khalil M, Wesseling P. Vertex-centered and cell-centered multigrid for interface problems. In *Preliminary Proceedings of the 4th Copper Mountain Conference on Multigrid Methods*, Vol. 3, Mandel J, McCormick SF (eds). Computational Mathematics Group, Univ. of Colorado: Denver, 1989; 61–97.
17. Khalil M, Wesseling P. Vertex-centered and cell-centered multigrid for interface problems. *Journal of Computational Physics* 1992; **98**:1–20.
18. Wesseling P. Cell-centered multigrid for interface problems. *Journal of Computational Physics* 1988; **79**:85–91.
19. Wesseling P. Cell-centered multigrid for interface problems. In *Multigrid Methods: Theory, Applications, and Supercomputing*, Vol. 110 of Lecture Notes in Pure and Applied Mathematics, Mc-Cormick SF (ed.). Marcel Dekker: New York, 1988; 631–641.
20. Wesseling P. *An Introduction to Multigrid Methods*. John Wiley: Chichester, UK, 1992.
21. Bramble J, Ewing R, Pasciak J, Shen J. The analysis of multigrid algorithms for cell centered finite difference methods. *Advances in Computational Mathematics* 1996; **5**:15–29.
22. Kwak DY. V-cycle multigrid for cell centered finite differences. *SIAM Journal on Scientific Computing* 1999; **21**:552–564.
23. Kwak DY, Lee JS. Multigrid algorithm for the cell centered finite difference method II: discontinuous coefficient case. *Numerical Methods for Partial Differential Equations* 2004; **20**(5):723–741.
24. Kwak DY, Lee JS. Comparison of V-cycle multigrid method for cell-centered finite difference on triangular meshes. *Numerical Methods for Partial Differential Equations* 2006; **22**:1080–1089.
25. Bergen B, Gradl T, Hülsemann F, Ruede U. A massively parallel multigrid method for finite elements. *Computational Science & Engineering* 2006; **8**:56–62.
26. Herbin R. An error estimate for a finite volume scheme for a diffusion-convection problem on a triangular mesh. *Numerical Methods for Partial Differential Equations* 1995; **11**:165–174.
27. Mishev ID. Finite volume methods on Voronoi meshes. *Numerical Methods for Partial Differential Equations* 1998; **14**:193–212.
28. Vassilevski PS, Petrova SI, Lazarov RD. Finite difference schemes on triangular cell-centered grids with local refinement. *SIAM Journal on Scientific and Statistical Computing* 1992; **13**:1287–1313.
29. Briggs WL, Henson VE, McCormick SF. *A Multigrid Tutorial*, (2nd edn). SIAM Publications: Philadelphia, 2000.
30. Brandt A. Guide to multigrid development. In *Multigrid Methods*, Vol. 960/1982, Lecture Notes in Mathematics. Springer: New York, 1982; 220–312.
31. Gaspar FJ, Gracia JL, Lisbona FJ. Fourier analysis for multigrid methods on triangular grids. *SIAM Journal on Scientific Computing* 2009; **31**:2081–2102.
32. Rodrigo C, Gaspar FJ, Oosterlee CW, Yavneh I. Accuracy measures and Fourier analysis for the full multigrid algorithm. *SIAM Journal on Scientific Computing* 2010; **32**:3108–3129.
33. Mohr M, Wienands R. Cell-centered multigrid revisited. *Computing and Visualization in Science* 2004; **7**:129–140.
34. Erten H, Üngör A. Quality triangulations with locally optimal Steiner points. *SIAM Journal on Scientific Computing* 2009; **31**:2103–2130.
35. Erten H, Üngör A. Computing triangulations without small and large angles. *International Symposium on Voronoi Diagrams (ISVD)*, Copenhagen, Denmark, 2009; 192–201.
36. Shewchuk JR. Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, Vol. 1148 of Lecture Notes in Computer Science, Lin MC, Manocha D (eds). Springer-Verlag: Berlin, May 1996; 203–222. (From the First ACM Workshop on Applied Computational Geometry.)
37. Shewchuk JR. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications* 2002; **22**:21–74.
38. Hemker PW. On the order of prolongations and restrictions in multigrid procedures. *Journal of Computational and Applied Mathematics* 1990; **32**:423–429.

DOI: 10.1002/nla