INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 18, 27–42 (1994)

UNSTRUCTURED VOLUME-AGGLOMERATION MG: SOLUTION OF THE POISSON EQUATION

BRUNO KOOBUS

INRIA, Sophia–Antipolis, 2004 route des Lucioles, B.P. 93, 06902 Valbonne Cedex, France

MARIE-HÉLÈNE LALLEMAND

INRIA, Rocquencourt, Domaine de Voluceau–Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France

AND

ALAIN DERVIEUX

INRIA, Sophia–Antipolis, 2004 route des Lucioles, B.P. 93, 06902 Valbonne Cedex, France

SUMMARY

We are interested in solving second-order PDEs with *multigrid* and unstructured meshes. The multigrid strategy we present here is adapted from the generalized finite volume agglomeration multigrid algorithm we have developed recently for the solution of the Euler equations. We now focus on Poisson's equation. A strategy is defined by introducing a correction factor for the diffusive terms, and some illustrating results are given.

0. INTRODUCTION: WHY UNSTRUCTURED VOLUME AGGLOMERATION?

Some industrial needs necessitate the simulation of complex flows. By *complex*, we mean that the geometry might be complex and/or the physical phenomena might show local high variations (high gradients, shocks, etc).

The first problem encountered in simulating this type of flow is to define a good mesh. By *good*, we mean to get a mesh which must be well adapted to both geometry and solution. One way to reach that goal is to define a regular mesh which is then locally refined where it is needed: we then end up with a mesh which is no longer regular. Another issue is to handle unstructured meshes and this is the option we are considering in this paper.

Many methods are available for solving PDEs on a finite element like mesh, but only a few of them can reach the maximal efficiency when dealing with very fine meshes. That means we are interested in low-complexity algorithms like Multi-Grid (MG) solvers, and we restrict our attention to this category.

Now is the problem of constructing a series of grids (levels) in the MG context. Whereas finite difference meshes induce easily defined coarser meshes, unstructured meshes set the important question on how to get coarser levels from a given arbitrary (fine) mesh. This is particularly the case for meshes obtained from mesh generators using, e.g. Voronoï or advancing-front techniques, or mesh adapter using, e.g. local refinements. At this point, we have to note that tree or

0271–2091/94/010027–16$13.00

Received January 1993
Revised August 1993

hierarchical, fully nested meshes are very useful for many simulations of physical phenomena, but are not yet very much applied to compressible CFD.

Then the common engineering question is how to define easily coarser meshes for solving efficiently some given flow problem on an arbitrary given (fine) mesh.

One answer is to build a sequence of unnested coarser meshes, e.g. as in References 1–4. This necessitates to handle a sequence of meshes instead of only one for each new computation.

Another answer is to apply some black-box linear solver such as Algebraic Multi-Grid (AMG)[5,6] where the different (coarse) levels are automatically built using only the initial (fine) matrix information.

Our answer is an agglomeration approach, i.e. coarser equations are derived on coarser meshes obtained by grouping the finite volumes of the fine grid. It needs only one mesh to generate automatically all the coarser meshes wanted, but compared to the above second answer, the coarsening is simple and is a low-storage process. It has also the advantage to define meshes which are all nested. This simplifies the way we want to define the transfer operators and also the programming.

The agglomeration approach has already been successfully applied to inviscid flow calculations.[7] It is, from our point of view, much more difficult to apply to viscous flow problems. The purpose of this paper is to show why it is difficult and to propose a device to overcome the main difficulty.

The paper is organized in the following way: we first give an idea of the relation existing between the volume agglomeration technique and the algebraic equation summing, in a simplified one-dimensional (1D) case; Section 2 is devoted to the definition of a volume agglomeration method related to a variational formulation of the two-dimensional (2D) Poisson equation; in Section 3 we give some information about some less-crucial options in the design of the MG scheme; at last, Section 4 illustrates our study by giving some significant numerical results and we then draw some conclusive remarks.

1. ALGEBRAIC AGGLOMERATION

1.1. A 1D trivial example

We consider the following periodic problem:

$$-\varepsilon u_{xx}+u_x=f \quad \text{on } [0, 1],\ u \text{ periodic of period } 1,$$

with the following finite difference/volume discretization

$$-\varepsilon\frac{u_{i+1}-2u_i+u_{i-1}}{h^2}+\frac{u_i-u_{i-1}}{h}=f_i \quad \text{for } i=1,\ldots,n-1, \tag{1}$$

where n denotes the total number of discretization nodes that we suppose to be odd, u_i stands for $u(x_i)$, $x_i=ih$ is the discretization node in the interval $[0, 1]$ which is regularly split with a space step $h=1/(n-1)$; $x_0\equiv 0$, $x_{n-1}\equiv 1$ and $x_n\equiv x_1$. Equation (1) leads to an algebraic system with a circulant $(n-1)\times(n-1)$ tridiagonal matrix, the row of which reads as

$$\left[-\frac{\varepsilon}{h^2}-\frac{1}{h} \quad 2\frac{\varepsilon}{h^2}+\frac{1}{h} \quad -\frac{\varepsilon}{h^2}\right]$$

and with a right-hand side (RHS) defined by the following $(n-1)$ component vector:

$$f=(f_1, f_2, \ldots, f_{n-1})^{\mathrm{T}}.$$

We now define the new coarse discretization derived from the original one by skipping only nodes with even indices. We denote by H the coarse space step which is twice larger than h. Now consider the new system obtained by

(1) summing line $i=2p-1$ and line $i+1=2p$ ($p=1,\ldots,(n-1)/2$), and
(2) identifying both fine unknowns u_{2p} and u_{2p-1} to the new coarse unknown value denoted by U_p.

The result of this process will be called in the following the *coarsened system.*

Remark 1.1. Note that this process is a Galerkin process and can be interpreted algebraically as follows. Let $A=(a_{ij})_{i,j=1,\ldots,n}$ be an $n\times n$ square matrix and denote the set of fine indices by I (the cardinality of I is n). Now suppose that we have a certain partition of I: $I=I_1\cup I_2\cup\cdots\cup I_N$, $N\ll n$ and that A is reordered following this partition. Then A can be written as

$$A=\begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1N} \\ A_{21} & A_{22} & \cdots & A_{2N} \\ \vdots & \cdots & \ddots & \vdots \\ A_{N1} & \cdots & \cdots & A_{NN} \end{pmatrix},$$

where $A_{LP}=(a_{i_L j_P})_{i_L\in I_L, j_P\in I_P}$, with $L, P=1,\ldots,N$, is an $N_L\times N_P$ block matrix, where N_L denotes the cardinality of the subset I_L, $L=1,\ldots,N$. Now if we define the new $N\times N$ coarse matrix $A^C=(\alpha^C_{PL})_{P,L=1,\ldots,N}$ derived from A by summing all entries belonging to a same block matrix A_{LP} and if we denote the resulting quantity by α^C_{LP}, we end up with the following coarse matrix

$$A^C=\mathcal{I}^T A\mathcal{I},$$

where the restriction operator $\mathcal{I}^T$ is the transpose of $\mathcal{I}$, and $\mathcal{I}$ is an $n\times N$ interpolation operator with coefficients ω_{j_P}, $j=1,\ldots,n$, $P=1,\ldots,N$ defined by

$$\omega_{j_P}=\begin{cases} 1 & \text{if } j\in I_P, \\ 0 & \text{otherwise.} \end{cases}$$

Now if u denotes the n-component fine unknown vector, then the coarse matrix A^C is associated with the N-component coarse unknown vector which we can denote by U^C. This means that for a given P, $P=1,\ldots,N$, we identify all fine variables u_j, $j\in I_P$ with the single coarse variable U_P. As it is algebraically written (with the transfer operators $\mathcal{I}$ and $\mathcal{I}^T$), this coarsening process is of Galerkin type. Therefore, properties such as symmetry and positiveness of the original matrix A still hold for the coarse grid matrices.

1.2. The advection case

We first consider the case $\varepsilon=0$ (advection equation); then we get the resulting coarsened equation

$$\frac{U_p-U_{p-1}}{h}=(f_{2p-1}+f_{2p}),\quad p=1,\ldots,N-1 \text{ with } N=(n+1)/2.$$

Noting that the new spatial step H is twice as large as h, one can verify that the coarsened system is also a *consistent* approximation of the advection continuous equation; further, we get a finite volume formulation that relies on the agglomeration of two neighbouring cells ($2m$ and $2m+1$, $m=0,\ldots,(n-2)/2$); this approach has been developed as the volume-agglomeration approach and has been applied to 2D and 3D Euler flow calculations.[7]

1.3. The diffusion case: inconsistency

We now set $\varepsilon = 1$ and drop the advection term; applying the coarsening process as previously described (summing line $2p$ to $2p-1$ and applying the identification: $U_{p-1} \equiv u_{2(p-1)}$, $U_p \equiv u_{2p-1} \equiv u_{2p}$, $U_{p+1} \equiv u_{2(p+1)} \equiv u_{2p+1}$), we now get the following system:

$$-\frac{U_{p+1}-2U_p+U_{p-1}}{h^2} = (f_{2p-1}+f_{2p}) \quad \text{for } p=1,\ldots,N-1.$$

We note that this system is *no more consistent* with the continuous system, a factor 2 is missing on the RHS.

Since we wish to combine diffusion and convection, any correction of this problem must be applied only to the diffusion term (and not to the RHS for example).

One evident remedy is to multiply the diffusion term by the factor 1/2; more generally, in 1D, we suggest to apply a factor 2^{-k} for the kth coarser mesh.

2. THE VARIATIONAL POINT OF VIEW

The above construction can be extended to the multidimensional case with unstructured meshes. One way is to consider a Galerkin variational formulation; we restrict our investigations to P1 approximations (continuous piecewise linear basis functions, with triangles (2D) or tetrahedra (3D), where the discretization nodes coincides with the vertices); for simplicity, we will consider only the 2D context.

Let V_h be the space generated by the standard continuous piecewise linear basis functions, denoted by ϕ_i, which are defined by

$$\phi_i = 1 \quad \text{at vertex } i, \ 0 \text{ on any other vertex,}$$

where i is any vertex of the triangulation not belonging to the boundary. Any function u in V_h can thus be expressed by

$$u = \sum_i u_i \phi_i.$$

Since the Galerkin principle is used, the basis functions are also the test functions. From Poisson's problem (with the Dirichlet boundary conditions),

$$-\Delta u = f, \quad \text{on } \Omega \subset \mathbb{R}^2,$$

we derive the following system:

$$\int_\Omega \left[\sum_i u_i \nabla \phi_i \cdot \nabla \phi_j - f\phi_j \right] dv = 0 \quad \text{for any } j.$$

In order to keep the Galerkin standpoint, we decided to compress our system by replacing the above basis functions by a smaller set of functions which are linear combinations of the original ones; moreover, in order to be able to apply a finite volume scheme in the volume-agglomeration framework, we define these new basis functions as follows.

Let I^f be the set of fine indices i and let n be its cardinality number. We

(i) define the following partition I^f:

$$I^f = \{1,\ldots,i,\ldots,n\} = I_1 \cup I_2 \cup \cdots \cup I_N,$$

where $N \ll n$;

(ii) define the new (coarse) basis functions, Φ_J, for any subset I_J, $J=1,\ldots,N$ by

$$\Phi_J=\sum_{j\in I_J}\phi_j.$$

Defining the N-component coarse unknown vector U by the following identification:

$$U_J\equiv u_j,\quad \forall j\in I_J,\ J=1,\ldots,N,$$

the new (coarsened) system to be solved is then written as

$$\int_\Omega\left[\sum_{I=1,\ldots,N}U_I\nabla\Phi_I\cdot\nabla\Phi_J-f\Phi_J\right]dv=0\quad\text{for all } J=1,\ldots,N.$$

In practice, the subsets I_J are built from neighbouring relations; a *coarsening algorithm* straightly deduced from an efficient algorithm which has been extensively used in Reference 7 and is defined in the following process.

Each cell i is considered successively (alternately in the increasing or decreasing numeral order according to the coarse level number to be built):

(i) if the current cell i has already been included in a coarse zone then go to (ii); else create a new coarse zone I containing the cell i, and neighbouring cells j which do not already belong to another previously defined coarse zone;
(ii) if all the cells have been coloured stop, else consider the next cell i and go to (i).

The resulting compressed/agglomerated system is inconsistent, as shown in the 1D case; the missing property of the coarse approximation space is the density of the basis set in the Sobolev space H^1 (while density is true in L_2).

The consequence of the inconsistency is not that a coarse grid correction will not help to reach the solution, but that low frequencies will not be damped with a reduction factor much less than one; instead, we may have a damping factor close to one, leading to a very poor convergence speed.

In order to overcome the inconsistency problem of the above agglomerated system, we introduce the following correction, which is empirically derived from simplified cases:

Correction rule: *In the corrected system, viscous terms are multiplied by the factor*

$$K_{\mathcal{N}}=2(\mathcal{N}-1)^2/(2\mathcal{N}-1)^2,$$

where $\mathcal{N}$ is an approximation of the number of nodes in one direction. That is estimated as follows:

$$\mathcal{N}=\sqrt[d]{\mathrm{NS}},$$

where d is the space dimension and NS *the total number of coarse zones.*

We can derive the following lemma.

Lemma 2.1. *The above correction yields a consistent coarsened approximation in the case of a Cartesian mesh (orthogonal, regular) discretizing a rectangle.*

Proof. We consider the following problem:

$$\begin{aligned}&-\Delta u=1,\quad \text{on } \Omega=]0,1[\times]0,1[,\\ &u|_\Gamma=0,\quad \text{with } \Gamma=\partial\Omega.\end{aligned}\tag{2}$$

Let m_i^h be the vertices of the triangulation discretizing Ω, defined by

$$m_i^h=(x_k^h, y_l^h)=((k-1)\,h, (l-1)\,h), \tag{3}$$

where $(k, l)\in[1, \ldots, n]\times[1, \ldots, n]$ and $i=(k-1)\,n+l\in[1, \ldots, n^2]$, $h=1/(n-1)$ is the space step for both x and y directions.

We discretize problem (2) by a Galerkin variational formulation with the standard continuous piecewise linear basis functions ϕ_i as test functions. Then we get the following linear system:

$$A^h u^h=s^h,$$

where

$$s_i^h=h^2,$$

$$A_{ii}^h=4, \quad A_{ii-1}^h=A_{ii+1}^h=A_{ii-n}^h=A_{ii+n}^h=-1, \quad \forall i/m_i^h\in\Omega.$$

We now consider the coarse grid made of zones I, $I=1, \ldots, N^2$, $N=n/2$ (n is supposed to be even), deduced from the fine cells i, $i=1, \ldots, n^2$ by the previous agglomerating/coarsening algorithm. We now write the variational formulation of problem (2), relying on the previously defined coarse basis functions, on this coarse grid discretization of Ω. This leads to the following linear system:

$$A^C u^C=s^C,$$

where

$$s_I^C=4h^2,$$

$$A_{II}^C=8, \qquad A_{II-1}^C=A_{II+1}^C=A_{II-N}^C=A_{II+N}^C=-2.$$

The exact solution of this system is given by

$$u_I^C=2(h/H)^2\,u_I^H,$$

with $H=1/(N-1)$, and where u^H is the exact discrete solution of the classical P1 Galerkin formulation of problem (2), relying on the triangulation in which nodes m_I^H are defined similar to (3), but where h and n are replaced by H and N, respectively. The correction factor K_N is, therefore given by $K_N=2(N-1)^2/(2N-1)$. □

3. MG ALGORITHM

Owing to the algebraic way of constructing the coarse grid equations (see Remark 1.1), it is clear that some of the good properties of the fine grid matrix will also hold for all the successive coarse grid matrices; in particular, we can derive the following lemma.

Lemma 3.1. *If the initial matrix is an M-matrix, then the resulting coarse matrices are also M-matrices.*

In particular, this is true for the discretized diffusion operator when the triangulation contains no obtuse angles (otherwise, the positivity may be violated), and for the discretized advection operator, when using a first-order upwind discretization scheme.

We now define the transfer operators of the MG algorithm. Let $N_{\max}$ be the total number of grid levels built from an initial fine grid G_1:

$$G_{N_{\max}}\subset G_{N_{\max}-1}\subset\cdots\subset G_2\subset G_1$$

and we denote by n_k the total number of control volumes of grid G_k by V^k the space of grid functions defined on G_k, for $k=1,\ldots,N_{\max}$; for two successive levels k and $k+1$, we denote by $\tilde{I}$ the set of level-k indices i, $i\in[1,\ldots,n_k]$, defining the level-$(k+1)$ control volume of index I, $I\in[1,\ldots,n_{k+1}]$. We define in the following the restriction operator $T_{k,k+1}: V^k\to V^{k+1}$, and the prolongation operator $T_{k+1,k}: V^{k+1}\to V^k$, for $k=1,\ldots,N_{\max}-1$:

(a) The *restriction operator* $T_{k,k+1}$, which transfers the (current fine) residual $r_k\in V^k$ from the grid G_k made of control volumes $\mathcal{V}_i^k$, $i=1,\ldots,n_k$, to the (immediate coarser) grid G_{k+1} made of control volumes $\mathcal{V}_I^{k+1}$, $I=1,\ldots,n_{k+1}$, defines the new transferred residual $r_{k+1}\in V^{k+1}$ by

$$r_{k+1,I}\overset{\text{def}}{=}[T_{k,k+1}(r_k)]_I=\sum_{i\in\tilde{I}} r_{k,i},$$

where $r_{k,i}$ is the value of the fine residual r_k on cell $\mathcal{V}_i^k$, $i=1,\ldots,n_k$, and $[T_{k,k+1}(r_k)]_I=r_{k+1,I}$ is the corresponding transferred value on zone $\mathcal{V}_I^{k+1}$, $\tilde{I}\ni i$, $I=1,\ldots,n_{k+1}$.

(b) The *prolongation operator* $T_{k+1,k}$, which transfers the level-$(k+1)$ error function $e_{k+1}\in V^{k+1}$ (i.e. the solution of the residual equation on the coarse level $k+1$) from grid G_{k+1} to the fine grid G_k, defines the correction to be added to the solution of the level-k problem; this operator is the composition of a projection operator $P_{k+1,k}: V^{k+1}\to V^k$ and an averaging operator $Q_k: V^k\to V^k$ defined by

(1) *Projection operator* $P_{k+1,k}: V^{k+1}\to V^k$ corresponds to the natural injection operator and is given by

$$P_{k+1,k}(e_{k+1})\overset{\text{def}}{=}e_k,$$

where e_k is the level-k grid function defined on the fine grid G_k by

$$e_{k,i}=e_{k+1,I},\quad \forall i=1,\ldots,n_k,\quad \forall I=1,\ldots,n_{k+1},\ \text{s.t. } \tilde{I}\ni i,$$

where $e_{k,i}$ and $e_{k+1,I}$ denote the value of e_k on control volume $\mathcal{V}_i^k$ and the value of e_{k+1} on control volume $\mathcal{V}_I^{k+1}$, respectively.

(2) *Averaging operator* $Q_k: V^k\to V^k$ is given by

$$Q_k(e_k)=\beta\,\text{ave}(e_k)+(1-\beta)e_k,$$

with

$$[\text{ave}(e_k)]_i\overset{\text{def}}{=}\frac{1}{\text{AREA}}\sum_{j\in\mathcal{N}^k(i)\cup\{i\}}\text{area}\,(\mathcal{V}_j^k)\,e_{k,j},$$

where

$$\text{AREA}=\sum_{j\in\mathcal{N}^k(i)\cup\{i\}}\text{area}\,(\mathcal{V}_j^k),$$

$\mathcal{N}^k(i)$ is the set of level-k indices j, corresponding to the control volumes $\mathcal{V}_j^k$ neighbouring $\mathcal{V}_i^k$, and β is a positive weighting coefficient between $e_{k,i}$ and $[\text{ave}(e_k)]_i$ that may be larger than one.

Therefore, the prolongation operator $T_{k+1,k}$ is written as

$$T_{k+1,k}(e_{k+1})=Q_k\circ P_{k+1,k}(e_{k+1}).$$

It is important to note that this prolongation operator, due to the averaging operator Q_k, smoothens the resulting interpolated error e_k on the (current) fine grid G_k.

4. IMPLEMENTATION

4.1. The coarse grid equations storage, memory requirements

Instead of handling the matrix coefficients, we perform the gathering of the equations from the following integrals:

$$a_{lm}^{ij} = \int_{\Omega} \frac{\partial \phi_i}{\partial x_l} \frac{\partial \phi_j}{\partial x_m} \, dv,$$

with $l, m = 1, \ldots, d$, d being the space dimension, and $i, j = 1, \ldots, n$, n being the total number of control volumes of the grid to be considered.

We say that to any couple of coarse zones (I, J) such that at least one a_{lm}^{IJ} is not zero, corresponds a 'segment'. On the fine grid, segments are edges.

On the finest grid G_1, there is no need of storing those integrals a_{lm}^{ij} and the assembling of the finest grid equation is done by segments and by elements (triangles).

On the coarse grids G_k, $k = 2, \ldots, N_{\max}$, the assembling of the coarse grid equations is done in two steps: we are storing all the integrals a_{lm}^{IJ}, $I, J = 1, \ldots, n_k$. The computations of these quantities are done by a loop over the triangles for the first coarse level $k = 2$, and by a loop over the level-k segments for the other coarse levels $k+1$, $k = 2, \ldots, N_{\max} - 1$. The coarse grid equations are then gathered by considering the coarse grid segments.

Therefore, the memory requirement of the multigrid process for storing the 2D diffusive terms (storage a_{lm}^{JJ} on the coarse grids) is given by

$$(\text{ns} + \tfrac{4}{3}\,\text{nseg}) \times \text{real numbers} \simeq 5\,\text{ns} \times \text{real numbers},$$

since $\text{nseg} \simeq 3\,\text{ns}$, where ns and nseg are, respectively, the number of vertices and segments of the initial (finest) mesh.

5. NUMERICAL EXPERIMENTS

For each experiment, a solution is considered to be converged if the iterative residual $\text{Res}(\alpha)$ reaches the tolerance $\varepsilon_r = 10^{-6}$; $\text{Res}(\alpha)$ is the iterative residual obtained at the iteration number α and is defined by

$$\text{Res}(\alpha) = \frac{\| u^{\alpha} - u^{\alpha-1} \|_{l^2}}{\| u^{1} \|_{l^2}},$$

where u^{α} is the αth multigrid solution iterate, with $u^0 = 0$.

As a smoother, we use either the Gauss–Seidel iteration, or the Jacobi iteration with under-relaxation.

The value of the weighting coefficient β of the prolongation operator $T_{k+1,k}$ is set at 1·5. This value has been chosen after a series of experiments on diffusion and advection for the three meshes which are considered in the sequel. For a V-cycle scheme with one pre- and post-smoothing on each grid, it appears that performing no smoothing in the transfer ($\beta = 0$) results in very bad performance for the Jacobi smoother with the uniform mesh, and for both smoothers if the mesh is stretched. Giving priority to the worse performances (stretched mesh), a good compromise is to set $\beta = 1{\cdot}5$. This value is set for all the results we present in the following section.

5.1. Impact of the agglomeration strategy on the MG convergence

We consider the following 1D problem:

$$-u_{xx}=1, \quad \text{on }]0,1[,$$
$$u(0)=u(1)=0.$$

We define a uniform discretization of the unit interval, resuting in a mesh with $n=101$ nodes. We are interested in the solution of this problem by an ideal two-grid scheme (i.e. the coarse grid equation is solved up to its full convergence) in the two following cases of agglomeration procedure:

(i) the agglomeration procedure is the one previously defined in Section 2;
(ii) the agglomeration procedure is modified by adding the following restriction in constructing the coarse zones: as soon as a new coarse zone I contains only a single fine cell i, this new coarse zone is destroyed and the cell i is put into another existing coarse zone containing at least one neighbour of the cell i.

Option (ii) was used in Reference 7 to avoid too small coarse zones. In the more recent versions, alternating the numbering order while grouping cells is sufficient to avoid very small coarse zones (i.e. zones reduced to a single fine zone). The smoother used in our ideal 2-G algorithm is the classical point Gauss–Seidel relaxation.

The mean reduction factor μ of the residual and the number of ideal 2-G cycles needed to reach the convergence level of 10^{-6} are given in Table I for the previous agglomeration procedures (i) and (ii). If α_{conv} denotes the total number of ideal 2-G cycles leading to convergence, then μ is defined by

$$\mu=\sqrt[\alpha_{conv}]{[\mathrm{Res}(\alpha_{conv})]}.$$

These notations are kept in all the following numerical tests.

One can note that the results are much better in the case of agglomeration (i). This is the option we keep in the sequel.

5.2. 2D advection–diffusion equation: some numerical experiments

We consider now the problem $-\varepsilon\,\Delta u+\mathrm{div}(\mathbf{V}u)=1$, with the homogeneous Dirichlet boundary conditions. We propose to solve this problem with a V-cycle MG scheme, where two iterations of the basic iterative method (one pre- and post-smoothing) are done on each level at each cycle. For most of the following numerical experiments, this number of smoothing sweeps per level corresponds with the optimal CPU time.

The convergence of the MG scheme is studied for three types of meshes:

(a) a structured uniform fine mesh,
(b) an unstructured fine mesh,
(c) a stretched structured fine mesh.

Table I. Ideal 2-grid convergence factors

	Agglomeration (i)	Agglomeration (ii)
μ	0·275	0·389
α_{conv}	12	16

Structured uniform fine mesh. The calculations are done on a uniform square mesh with 41×41 nodes (see Figure 1). The number of grids used is five and the finite volume partitions of the different grids are sketched in Figure 2.

The multigrid convergence factors for several values of ε and V are given in Tables II and III. In Table II, the under-relaxation parameter ω of Jacobi smoother is set at 0·8. Indeed, with this value of ω, Jacobi has the best smoothing properties for the Laplacian problem.

For the same convergence level, a single-grid Gauss–Seidel solver leads to $\alpha_{conv} = 2213$, and a single-grid Jacobi solver (with under-relaxation parameter $\omega = 0{\cdot}8$) leads to $\alpha_{conv} = 5521$.

Note that for the Poisson equation (Table II), the mean reduction factor is better with Gauss–Seidel as a smoother; indeed, it is well known that Gauss–Seidel has better smoothing

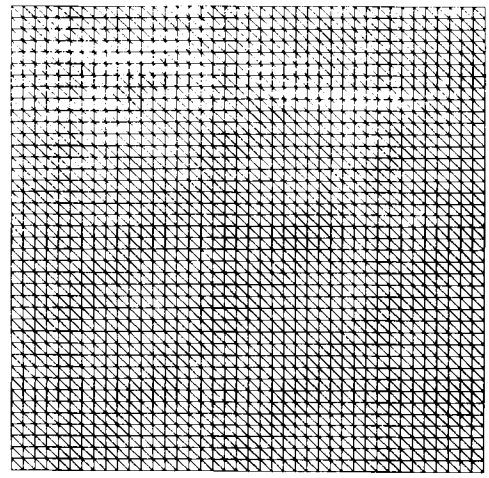

Figure 1. The 41×41 triangulation of the 2D unit square

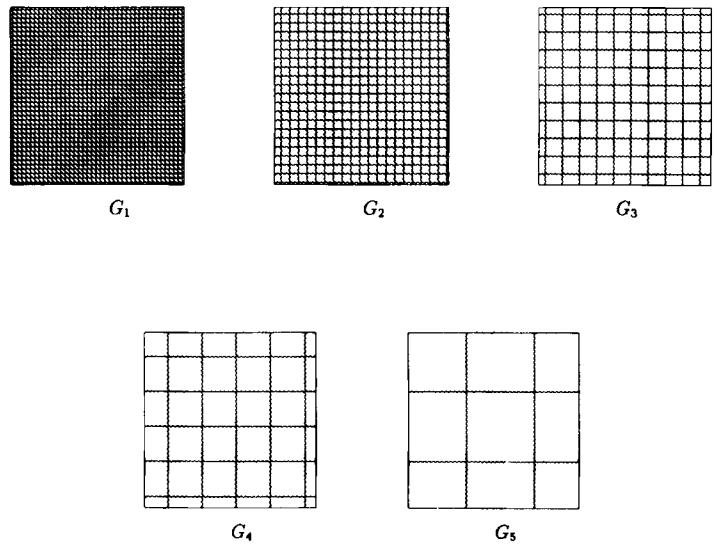

Figure 2. The finite volume partition grids G_k, $k = 1, \ldots, 5$

properties than Jacobi (even used with an optimal relaxation parameter), when applied to Poisson's equation, but the difference is not so big with this regular (and not very fine) mesh. Table III shows that the mean reduction factor of the residual becomes much smaller when the convection part dominates the diffusion part. This can be explained by the fact that, with these specific mesh and V, Gauss–Seidel tends to become a direct solver when ε is small compared to the mesh size.

If the cost of the transfers is neglected, the arithmetic complexity of one V-cycle, where two smoothing sweeps of Jacobi (one pre- and post-smoothing) are done on each level, can be evaluated as about four sweeps of Jacobi on the finest grid.[11] If the matrix coefficients are not stored, the computation for assembling a flux with Gauss–Seidel is twice as important as with Jacobi. Therefore, the calculation cost of the RHS of the linear system to be solved on a coarse grid G_k, $k=2, \ldots, N_{\max}$, is about half a sweep of Gauss–Seidel on the grid G_{k-1}. The arithmetic complexity of one V-cycle with two Gauss–Seidel iterations on each level (one pre- and post-smoothing) can then be evaluated as about 10/3 sweeps of Gauss–Seidel on the finest grid. Consequently, the relative efficiency (in terms of arithmetic complexity) between the single-grid scheme (1G) and the multigrid scheme (MG) for the Poisson equation (Table II) is given by

$$\frac{1G_{(GS)}}{MG_{(GS)}} = 51{\cdot}1, \qquad \frac{1G_{(J)}}{MG_{(J)}} = 92{\cdot}0,$$

where GS (resp. J) stands for Gauss–Seidel (resp. Jacobi). Note the significant gain in arithmetic complexity obtained by the MG scheme.

Unstructured fine mesh. A NACA0012, aerofoil mesh involving 3114 nodes (see Figure 3) is now used for the same model equation. Six grids (depicted in Figure 4) are used to define the V-cycle scheme. Convergence results are given in Tables IV and V.

Table II. $\varepsilon=1$, $\mathbf{V}=(0, 0)^{\mathrm{T}}$

	MG-Jacobi ($\omega=0{\cdot}8$)	MG-Gauss–Seidel
μ	0·364	0·300
α_{conv}	15	13

Table III. $\mathbf{V}=(1, 0)^{\mathrm{T}}$, MG-Gauss–Seidel

	$\varepsilon=1$	$\varepsilon=10^{-1}$	$\varepsilon=10^{-3}$
μ	0·306	0·234	0·117
α_{conv}	13	11	18

Table IV. $\varepsilon=1$, $\mathbf{V}=(0, 0)^{\mathrm{T}}$

	MG-Jacobi ($\omega=0{\cdot}8$)	MG-Gauss–Seidel
μ	0·690	0·496
α_{conv}	39	21

Table V. $\mathbf{V}=(1, 0)^T$, MG-Gauss–Seidel

	$\varepsilon=1$	$\varepsilon=10^{-1}$	$\varepsilon=10^{-3}$
μ	0·422	0·402	0·399
α_{conv}	17	17	17

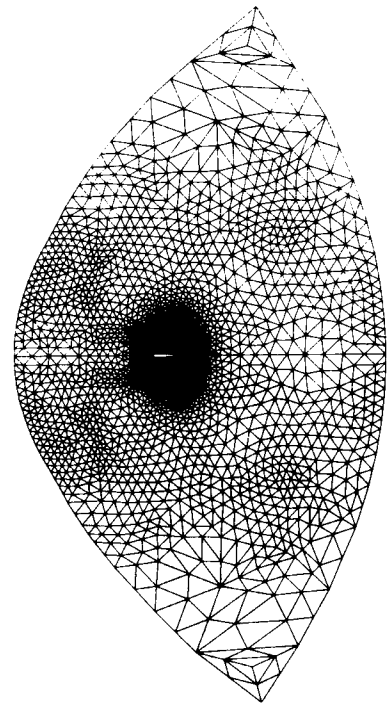

Figure 3. The NACA0012 aerofoil triangulation with 3114 nodes

For the same convergence level, a single-grid Gauss–Seidel solver leads to $\alpha_{conv}=3007$, whereas a single-grid Jacobi solver (with a relaxation parameter $\omega=0{\cdot}8$) leads to $\alpha_{conv}=7395$.

One can note that the gain in the convergence obtained by Gauss–Seidel versus Jacobi (Table IV) is more important in this example than in the previous unit square example. We can also observe that the mean reduction factor μ improves only slightly when the diffusion is decreasing (Table V). The relative efficiency between the single-grid scheme and the multigrid one for the Poisson equation (Table IV) is given by

$$\frac{1G_{(GS)}}{MG_{(GS)}}=43{\cdot}0, \qquad \frac{1G_{(J)}}{MG_{(J)}}=47{\cdot}4.$$

Note again the improvement of the MG complexity.

Stretched structured fine mesh. We present now calculations on a 41×41 square stretched mesh in which the maximal ratio between the length of two orthogonal sides of the flattest elements is one hundred (see Figure 5). Five grids are used for the experiments (see Figure 6).

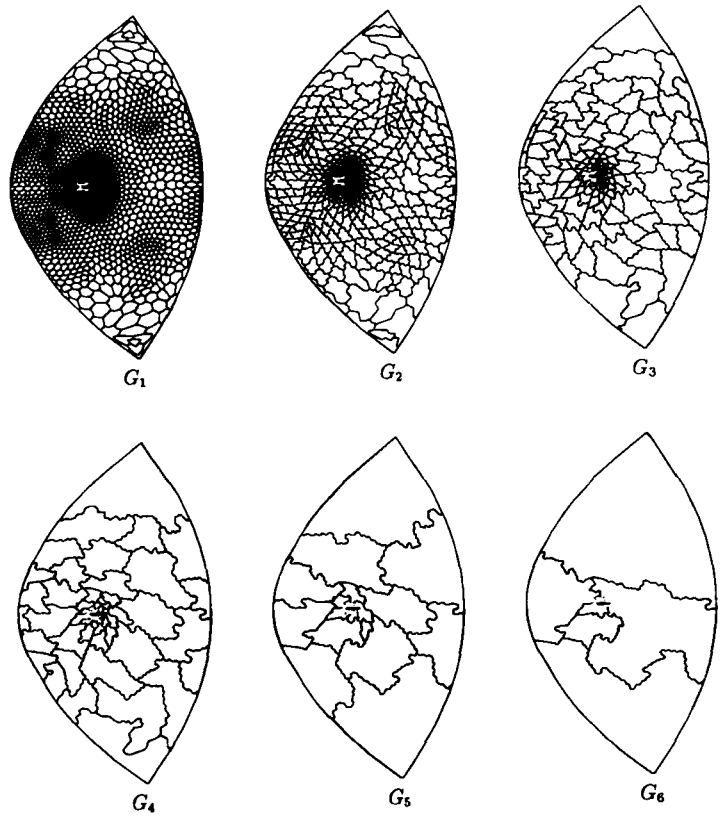

Figure 4. The NACA0012 aerofoil finite volume partition grids G_k, $k=1,\ldots,6$

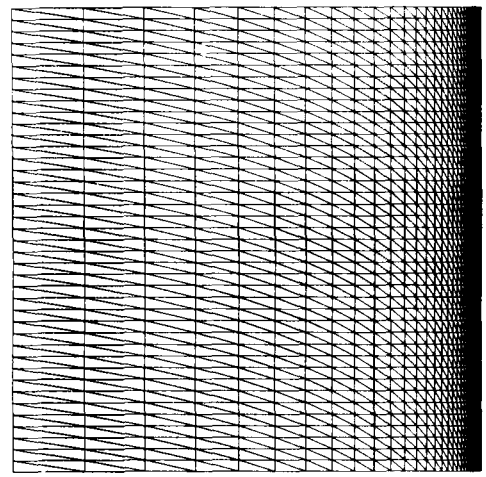

Figure 5. The stretched 41×41 triangulation of the 2D unit square

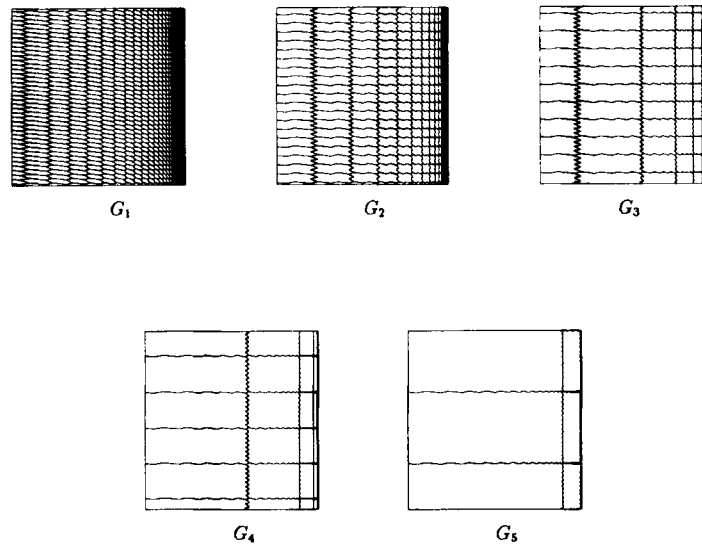

Figure 6. The finite volume partition grids G_k, $k=1,\ldots,5$

Table VI. $\varepsilon=1$, $\mathbf{V}=(0,0)^T$

	MG-Jacobi ($\omega=0{\cdot}8$)	MG-Gauss–Seidel
μ	0·925	0·842
α_{conv}	178	82

The results about multigrid convergence are given in Table VI for the Poisson equation $-\Delta u=1$ with the Dirichlet boundary conditions.

For the same convergence level, a single-grid Gauss–Seidel solver leads to $\alpha_{conv}=1440$, whereas a single-grid Jacobi solver ($\omega=0{\cdot}8$) leads to $\alpha_{conv}=3597$.

We point out that the convergence rate is much slower than the one gotten in the case of uniform meshes. This is essentially due to the fact that standard solvers such as Gauss–Seidel and especially Jacobi are not very efficient smoothers in the case of meshes having 'very flat' cells. As an illustration, we show in Figure 7 the dependency of the mean reduction factor versus the number of smoothing sweeps (Gauss–Seidel relaxations) done on the fine level at each MG-cycle for an ideal two-grid scheme. From the behaviour of the convergence history, we deduce that a better smoother would improve a lot our existing MG-method in the case of stretched meshes; indeed, increasing the number of Gauss–Seidel relaxations on the fine level up to a moderate number of relaxation sweeps, results in a corresponding acceleration of convergence to be compared with the 1400 Gauss–Seidel iterations required on the fine grid for the same convergence level. The same behaviour is observed with a five-grid V-cycle scheme (see Figure 8).

Moreover, we obtain the following relative efficiency between the single-grid scheme and the multigrid scheme for each of the two smoothers Gauss–Seidel and Jacobi:

$$\frac{1G_{(GS)}}{MG_{(GS)}}=5{\cdot}3, \qquad \frac{1G_{(J)}}{MG_{(J)}}=5{\cdot}1.$$

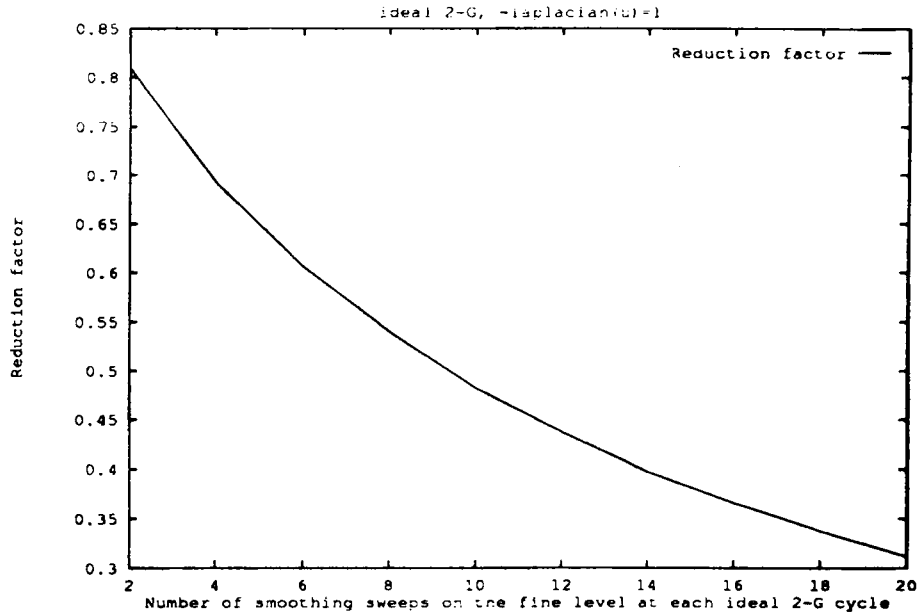

Figure 7. Ideal 2-G convergence factor as a function of number of fine smoothing sweeps (pre- and post-smoothings)

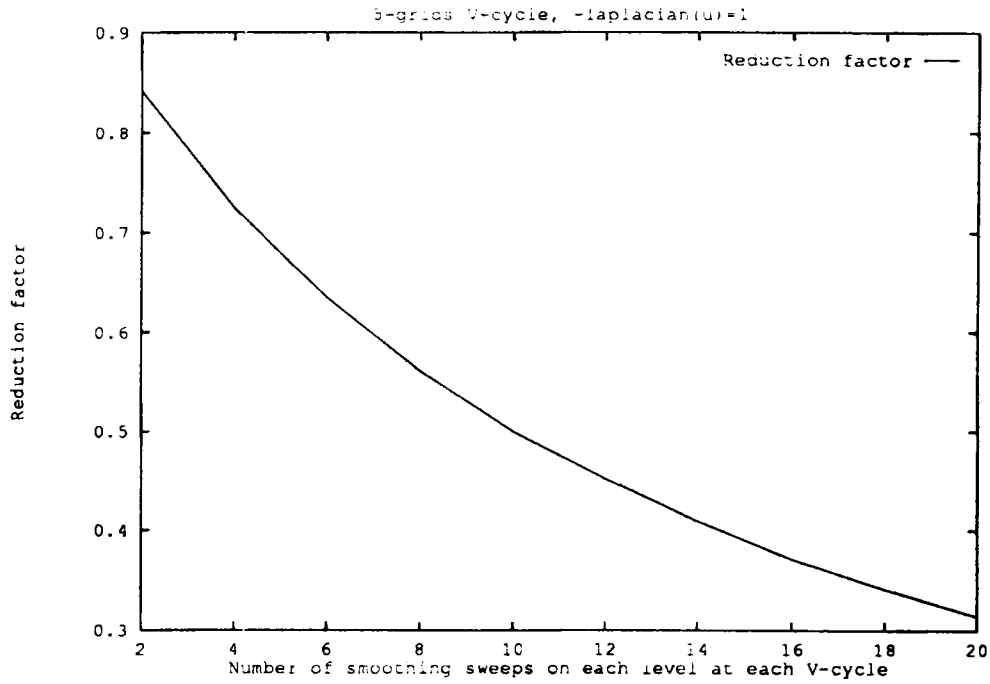

Figure 8. 5-G V-cycle convergence factor as a function of number of smoothing sweeps (pre- and post-smoothings) done on each level

Note that the efficiency of the multigrid scheme compared to the single grid one is much less important than in the previous cases (a) and (b).

Another point to emphasize is that in all the above experiments the convergence factors were observed to be approximately level independent.

Finally note (see Table VI) that the number of V-cycles is divided by a factor greater than two when Gauss–Seidel is used as a smoother instead of Jacobi.

6. CONCLUSION

Solving the Poisson equation in 2D is neither a big challenge nor our ultimate goal; our motivation was to find a method extending to viscous flows a MG algorithm which was already successfully applied to inviscid flows[7] on unstructured meshes.

We found an important theoretical difficulty related to the inter-level inconsistency, that remains unsolved; however, the trick that we applied seems to be efficient in the different practical examples that we have considered. Systems on very irregular but non-stretched meshes are solved within 20 cycles; convergence results obtained in the case of stretched meshes appeared to depend more on the efficiency of the smoother, rather than on the quality of the resulting coarse systems. The efficient treatment of stretched meshes is a classical challenge, in particular for the definition of good and robust MG schemes. Another remedy that comes to mind is to define a more sophisticated agglomerating process which would respect, at least geometrically, the scaling of the mesh size.

Extensions to other types of elements and 3D problems seem to be reasonably possible.

The global evaluation we can make from these preliminary studies is, from our opinion, positive and promising with respect to further extensions to the calculations of viscous compressible flows. For that program, several methods can be exploited, such as implicit solvers for Navier–Stokes,[9,10] and MG relaxation methods for Euler,[11] although those methods are already developed in the context of multiple triangulations.

REFERENCES

1. J. Peraire, J. Peiro and K. Morgan, 'A 3-D finite element multi-grid solver for the Euler equations', *AIAA Paper 92-0449*, 1992.
2. D. Mavriplis and A. Jameson, 'Multigrid solution of the two-dimensional Euler equations on unstructured meshes', *AIAA Paper 87-0350*, 1987.
3. M.-P. Leclercq and B. Stoufflet, 'Characteristic multigrid method, application to solve the Euler equations with unstructured and unnested grids', *J. Comp. Phys.*, **104** (2), 329–346 (1993).
4. H. Guillard, 'Note-nested multi-grid method with DeLaunay coarsening', *INRIA Research Report 1898*, 1993.
5. A. Brandt, S. McCormick and J. Ruge, 'Algebraic multigrid (AMG) for sparse matrix equations', in D. J. Evans (ed.), *Sparsity and its Applications*, Cambridge University Press, Cambridge, 1984.
6. J. Ruge, S. McCormick and M.-H. Lallemand, 'AMG for the Euler equations', *DRET Report*, 1989.
7. M.-H. Lallemand, H. Steve and A. Dervieux, 'Unstructured multigridding by volume agglomerating: current status', *Comput. Fluids*, **21**, 397–433 (1992).
8. M.-H. Lallemand, 'Schémas décentrés multigrilles pour la résolution des équations d'Euler en éléments finis', *Thesis*, University of Marseille, 1988.
9. L. Fezoui, S. Lanteri, B. Larrouturou and C. Olivier, 'Résolution numérique des équations de Navier–Stokes pour un fluide compressible en maillage triangulaire', *Rapport de Recherche INRIA 1033*, 1989.
10. C. Olivier and B. Larrouturou, 'On the numerical approximation of the k–ε turbulence model for two dimensional compressible flows', *INRIA Research Report 1526*, 1991.
11. E. Morano, H. Guillard, A. Dervieux, M.-P. Leclercq and B. Stoufflet, 'Faster relaxations for non-structured MG with Voronoï coarsening', in Ch. Hirsch (ed.), *Computational Fluid Dynamics '92, Proc. 1st European Computational Fluid Dynamics Conf.*, Brussels, 7–11 September 1992.