INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids **26**: 927–957 (1998)

A SEMI-COARSENING STRATEGY FOR UNSTRUCTURED MULTIGRID BASED ON AGGLOMERATION

JEROME FRANCESCATTO AND ALAIN DERVIEUX*
INRIA, B.P. 93, 06902 Sophia-Antipolis Cedex, France

SUMMARY

Extending multigrid concepts to the calculation of complex compressible flow is usually not straightforward. This is especially true when non-embedded grid hierarchies or volume agglomeration strategies are used to construct a gradation of unstructured grids. In this work, a multigrid method for solving second-order PDE's on stretched unstructured triangulations is studied. The finite volume agglomeration multigrid technique originally developed for solving the Euler equations is used (M.-H. Lallemand and A. Dervieux, in *Multigrid Methods, Theory, Applications and Supercomputing*, Marcel Dekker, 337–363 (1988)). First, a directional semi-coarsening strategy based on Poisson's equation is proposed. The second-order derivatives are approximated on each level by introducing a correction factor adapted to the semi-coarsening strategy. Then, this method is applied to solve the Poisson equation. It is extended to the 2D Reynolds-averaged Navier–Stokes equations with appropriate boundary treatment for low-Reynolds number turbulent flows. © 1998 John Wiley & Sons, Ltd.

KEY WORDS: multigrid methods; finite elements; finite volumes; semi-coarsening; numerical analysis; turbulent flows; compressible flows

1. INTRODUCTION

The prediction of complex compressible flows which relies on the Reynolds-averaged Navier–Stokes model computed on unstructured meshes has gained in maturity. This maturity will not be complete until low-Reynolds number statistical models (LRM), such as the so-called low-Reynolds k–ϵ model, are easily computed in 2D and 3D. A variety of methods for generating meshes with boundary layer elements are becoming available. Also, a number of improved numerical approximations have been derived recently [1]. The concern of this study is the improvement of iterative solution algorithms methods. Indeed, with model improvement, model error should also decrease. Even with accurate approximations, this requires the approximation error to be smaller, leading to the use of fine meshes. As a result, the asymptotic complexity of the algorithm of resolution becomes an important feature and motivates the use of a multigrid (MG) scheme.

A central issue in the application of MG to LRM is the efficiency for highly stretched meshes. Indeed, nodewise (explicit multistep Jacobi and Gauss–Seidel) smoothers are generally adopted for MG methods in flow problems. MG methods, relying on such smoothers, are not

* Correspondence to: INRIA, B.P. 93, 06902 Sophia-Antipolis Cedex, France. E-mail: Alain.Dervieux@sophia.inria.fr

CCC 0271–2091/98/080927–31$17.50

Received December 1996
Revised May 1997

so efficient when stretched meshes are used. High frequencies aligned with the mesh are neither attenuated by the fine grid nodewise smoothing, nor by the coarse grid correction if full coarsening (in both mesh directions) is applied. Two main approaches are possible in order to solve this problem: the smoother can be improved with a directionally implicit relaxation, such as a line relaxation or even ILU relaxation [2–4], or the coarse mesh construction can be the result of a directional semi-coarsening. In this paper, we adhere to the latter approach.

The decision of how to semi-coarsen must be made before stretched LRM calculations are treated with semi-coarsening. In numerous works dealing with structured meshes, it is interesting to keep the several coarse meshes at each level that all of the possible semi-coarsening offers (see for example Reference [5]). It is not natural to extend this multiple coarsening option to unstructured multigrid, since no *a priori* special direction can be identified in the mesh, except a stretching direction that can be identified, e.g. by a purely algebraic method (see References [6–8]). In this work, a single coarsening strategy is proposed, aligned with the stretching direction. This strategy has been studied for non-embedded meshes in Reference [9]. An approach in which volume-agglomeration is adopted for building coarser levels is presented in Reference [10]. The present study contributes to a deeper understanding of the mechanism of volume-agglomerated directional coarsening. In particular, the method by which the inconsistency introduced by the inaccurate transfers defined by agglomeration can be compensated by an anisotropic correction factor is discussed.

For this purpose, an analysis of directional or anisotropic coarsening is discussed in Section 1, in order to predict its efficiency compared with the application of an isotropic full coarsening algorithm, in the two cases of isotropic and stretched meshes. In the second section, the anisotropic agglomeration process and the way the anisotropic correction factor can be derived in a recursive way from one level to the coarser level is described. Some preliminary experiments related to the scalar Poisson equation are considered and analyzed in Section 3. Then laminar Navier–Stokes flows are computed with an implicit formulation involving an agglomerated anisotropic MG algorithm. Finally, two-layer LRM computations are presented and discussed, as the principal objective of this study.

2. SPATIAL APPROXIMATION

The approximation described in Reference [11], is a combination of finite volume and finite element methods. Convergence proofs of algorithms utilizing this combination are presented, e.g. by Baba and Tabata [12]. It is assumed that the computational domain Ω is bounded by a polygon. τ_h denotes a triangulation of Ω, n_h the total number of vertices in τ_h, and φ_i the basis function associated with each vertex or node a_i.

A dual finite volume partition of Ω, called the dual mesh of τ_h is derived and made of the control volumes C_i built from triangle medians around each vertex a_i (see Figure 1). Let $K(i)$ be the set of the indices of the neighboring nodes of a_i. $\partial C_{ij} = \partial C_i \cap \partial C_j = [G_{1,ij}, I_{ij}] \cup [I_{ij}, G_{2,ij}]$ is the interface between two cells and $\vec{n}_{ij} = \int_{\partial C_{ij}} \vec{n}\, \mathrm{d}\delta$ is the approximation of the normal vector to interface ∂C_{ij} (Figure 2).

The following scalar advection–diffusion equation on $\Omega \subset \mathbb{R}^2$ is discretized:

$$\begin{cases} -\Delta u + \vec{\nabla}.(\vec{V}u) = f \text{ on } \Omega \\ u|_\Gamma = 0 \end{cases} \tag{1}$$

where Ω is an open bounded domain with a regular boundary Γ, $\vec{V}$ a constant velocity vector, f a continuous function and $u(x, y)$ the unknown. The continuous problem on Ω, is approximated by the discrete problem defined on a triangulation Ω_h:

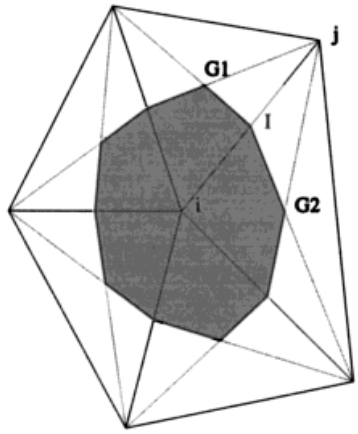

Figure 1. Cell C_i is shaded.

$$-\sum_{T,a_i\in T}\iint_T \vec{\nabla}u.\vec{\nabla}\varphi_i \,\mathrm{d}x\,\mathrm{d}y+\sum_{j\in K(i)}\int_{\partial C_{ij}} u\vec{\nabla}.\vec{n}\,\mathrm{d}\delta=\iint_{C_i} f_i\,\mathrm{d}x\,\mathrm{d}y. \tag{2}$$

This method results in two different kinds of spatial discretization (compatibility of these discretizations is the consequence of choosing median dual cells, see References [12,13] for details):

- the *diffusive term* is calculated on each triangle and a standard P1-Galerkin formulation ('linear elements') is used:

$$\iint_T \vec{\nabla}\, u.\vec{\nabla}\,\varphi_i\,\mathrm{d}x\,\mathrm{d}y=\sum_{j,a_j\in T}\mathrm{area}(T)\,u_j\,\vec{\nabla}\,\varphi_j.\vec{\nabla}\,\varphi_i;$$

  where u_j is the value of u at node a_j.
- the *advective term* corresponding to the flux between the two cells C_i and C_j, is calculated through the interface ∂C_{ij}. Flux integration for a_i is the sum of the contribution of all neighbors a_j. The integration on ∂C_{ij} of the advective term for $\vec{V}$ constant is written as:

$$\int_{\partial C_{ij}} u\vec{V}.\vec{n}\,\mathrm{d}\delta=\Phi(u_{ij},u_{ji},\vec{n}_{ij}), \tag{3}$$

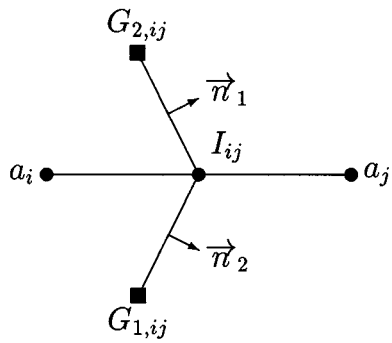

Figure 2. Interface ∂C_{ij}.

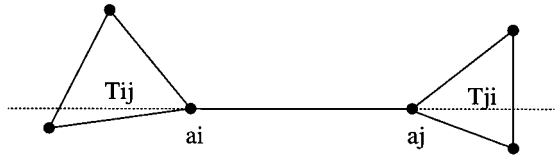

Figure 3. Downstream and upstream triangles T_{ij} and T_{ji}.

where

$$\begin{cases} \Phi(u_{ij}, u_{ji}, \vec{n}_{ij}) = \alpha_{ij}(\theta_{ij}u_{ij} + (1-\theta_{ij})u_{ij}) \\ \alpha_{ij} = \vec{V}.\vec{n}_{ij} \\ \theta_{ij} = \dfrac{1}{2}(\text{sign}\,(\alpha_{ij}) + 1) \end{cases}$$

The values u_{ij} and u_{ji} are interpolates of u into interface ∂C_{ij}. $u_{ij} = u_i$ and $u_{ji} = u_j$ results in a first-order scheme.

A second-order scheme is obtained using a MUSCL interpolation [14]:

$$\begin{aligned} u_{ij} &= u_i + \frac{1}{2}[(1-\beta)(\vec{\nabla}u)_{ij}^{\text{cent}} + \beta(\vec{\nabla}u)_{ij}^{upw}].\vec{a_i a_j} \\ u_{ji} &= u_j + \frac{1}{2}[(1-\beta)(\vec{\nabla}u)_{ij}^{\text{cent}} + \beta(\vec{\nabla}u)_{ji}^{upw}].\vec{a_i a_j} \end{aligned} \tag{4}$$

where the three following gradient approximations are given by:

$$(\vec{\nabla}u)_{ij}^{\text{cent}}.\vec{a_i a_j} = u_j - u_i, \qquad (\vec{\nabla}u)_{ij}^{upw} = \vec{\nabla}u(T_{ij}), \qquad (\vec{\nabla}u)_{ji}^{upw} = \vec{\nabla}u(T_{ji}), \tag{5}$$

and where T_{ij} and T_{ji} are the triangles intersected by the line (a_i, a_j) defined by the vector $\vec{a_i a_j}$ as shown in Figure 3. The scheme is second-order-accurate for $\beta = 1/2$ (Fromm scheme) and third-order-accurate on Cartesian meshes and linear systems for $\beta = 1/3$.

3. LINEAR MULTIGRID METHOD

The proposed multigrid method is an extension of the linear multigrid approach developed by Lallemand *et al.* [15,16] for solving linearized Euler systems, and extended to diffusion terms present in Poisson problems by Koobus *et al.* [17], and to high Reynolds turbulence models by Carré [13].

3.1. Grid coarsening by isotropic agglomeration

Volume-agglomeration coarsening technique is based on a neighboring relation. First, from a fine unstructured triangulation, the dual finite volume partition is derived by building cells/volumes (around vertices) as shown above (Figure 1). Coarser finite volume partitions are then generated automatically. This is achieved by applying a technique of volume-agglomeration, described in Reference [16], that assembles neighboring control volumes on the finest grid (e.g. those having a common boundary) to build the cells of the coarser level, according to the following rule:

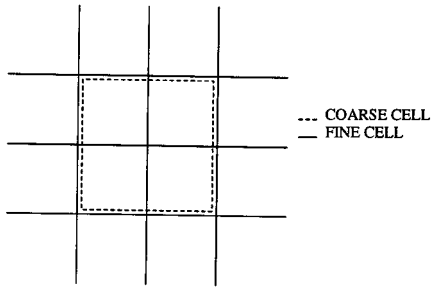

Figure 4. Isotropic coarsening structured mesh.

Consider successively every cell C_i of the domain.

1. *If C_i has already been included in a group (new coarse cell) then consider the next cell. Otherwise, create a new group containing C_i and put neighbors of C_i which do not already belong to another existing group into this group.*
2. *Go to the next cell.*

The main advantage of this method is to generate coarser grids automatically and rapidly, without rebuilding a genuine coarser triangulation.

3.2. Grid coarsening by anisotropic agglomeration

3.2.1. A short introduction to semi-coarsening. The essential principle of MG is to combine iterations that are good contractions for each subset of a partition of the whole set of frequencies of the considered system. In the case of what we call 'an isotropic algebraic system', which is e.g. provided by a discretization on a uniform mesh of an isotropic partial differential equation (i.e. relying on an isotropic operator), it is well-known that usual relaxation schemes are efficient contractions for all high frequencies. They can be successfully combined with a multigrid scheme relying on isotropic coarsening, as sketched in Figure 4.

If some cause of anisotropy arises, such as either an anisotropic physical property, or a stretched mesh, then the above property is lost.

It is then possible to think of a new class of relaxation that would smooth out all the high frequency modes; an example is line relaxation, but such a choice is rather cumbersome to apply to unstructured meshes.

In this work it is preferred to remark that usual relaxation is still a good smoother for a subclass of the high frequency modes, the *transverse* ones (to anisotropy direction), and to derive a coarser mesh that would still involve the non-smoothed *aligned* frequencies. This strategy is called semi-coarsening, and increases the mesh size in one direction only, as sketched in Figure 5.

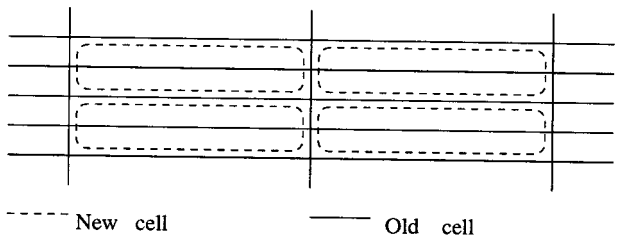

Figure 5. Anisotropic coarsening structured mesh.

Table I. Local mode Fourier analysis of usual multigrid algorithms for the Laplace operator

ϵ	Isotropic MG		Anisotropic MG	
	ω_{opt}	g_ϵ	ω_{opt}	g_ϵ
1.0	1.00	0.50	1.00	0.50
0.5	1.00	0.68	0 88	0.44
10^{-1}	1.00	0.98	0.83	0.41
10^{-3}	1.00	1.00	0.83	0.41
10^{-5}	1.00	1.00	0.83	0.41

Optimum relaxation parameter ω_{opt} and reduction factor g_ϵ of the Gauss–Seidel relaxation as functions of $\epsilon = \Delta x/\Delta y$; for the isotropic MG algorithm, convergence degrades with mesh stretching ($\epsilon \to 0$), while for the anisotropic semi-coarsened algorithm, convergence improves with stretching.

In the case of a periodic problem with the Laplace operator, a local mode analysis shows that semi-coarsening in the adequate (transverse) direction produces a fast converging multigrid algorithm when combined with a damped Gauss–Seidel relaxation, regardless of the strength of anisotropy. It is particularly faster than the standard isotropic multigrid, even when applied to an isotropic system. This fact is illustrated in Table I, which presents reduction factors for both strategies. It is observed that the best 2D case corresponds to the anisotropic MG algorithm applied to an infinitely anisotropic system, for which performances of the 1D MG algorithm are obtained.

3.2.2. Anisotropic agglomeration algorithm. Our algorithm relies on two mechanisms. Firstly, the 'local metrics' are identified, i.e. the stretching direction and strength. Secondly, agglomeration is adapted to local metrics.

In order to build the *local metrics*, inspired by an idea used in algebraic multigrid [6–18], it was decided to measure the coefficients of the finite element Laplace operator A, in order to evaluate the strong connections. The coefficients are denoted by a_{ij}, $(i, j) \in (1, \ldots, n_h)^2$, the entries of matrix A related to triangulation τ_h:

$$A = (a_{ij})_{ij}; \qquad a_{ij} = \iint \vec{\nabla}\varphi_i . \vec{\nabla}\varphi_j \, \mathrm{d}x \, \mathrm{d}y.$$

Definition 3.1. *For a given cell C_i, the neighborhood of C_i is defined by*

$$N_i = \{j \in I, \quad a_{ij} \neq 0\},$$

where $I = \{i, i = 1, \ldots, n_h\}$ denotes the set of indices i of all cells.

Strong connections are defined by:

Definition 3.2. *i is strongly connected to j if*

$$|a_{ij}| \geq \epsilon_1 \max_{p \in N_i} |a_{ip}|, \quad \epsilon_1 = \frac{1}{4}. \tag{6}$$

S_i denotes the set of indices j to which i is strongly connected.

The choice of 1/4 is rather standard (see Reference [18]); therefore, an actual anisotropic treatment is used only for an aspect ratio larger than two.

For each cell C_i, C_{j0} denotes the neighboring cell having the *strongest connection*:

$$a_{ij_0} = \max_{j \in S_i} |a_{ij}|,$$

and G_i denotes the *cell barycenter*.

This allows a *first guess* of the stretching direction to be made (Figure 6):

$$\vec{e}_\xi = \frac{\vec{G_i G}_{j_0}}{\|\vec{G_i G}_{j_0}\|} = (\cos(\theta_i), \ -\sin(\theta_i))^T \tag{7}$$

and the normal direction:

$$\vec{e}_\eta = (\sin(\theta_i), \cos(\theta_i))^T. \tag{8}$$

Then a more accurate stretching direction is defined as follows:

$$\vec{V}_i = \sum_{j \in S_i} |\vec{G_i G_j}.\vec{e}_\xi| \ \vec{e}_\xi + \sum_{j \in S_i} |\vec{G_i G_j}\vec{e}_\eta| \vec{e}_\eta. \tag{9}$$

The ratio L_i between components of $\vec{V}_i$:

$$L_i = \frac{\sum_{j \in S_i} |\vec{G_i G_j}\vec{e}_\xi|}{\sum_{j \in S_i} |\vec{G_i G_j}\vec{e}_\eta|} \tag{10}$$

determines the strength of stretching.

The *anisotropic agglomeration algorithm* is defined as follows:

Consider successively every cell C_i of the mesh:

1. *if C_i has already been included in a group (new coarse cell), then consider the next cell. Else create a new group containing C_i: identify the strongest connection j_0, and the stretching strength L_i. If $L_i \in [1/2, 2]$, then put into this group, the neighboring cells C_j such that $j \in N_i$ which do not*

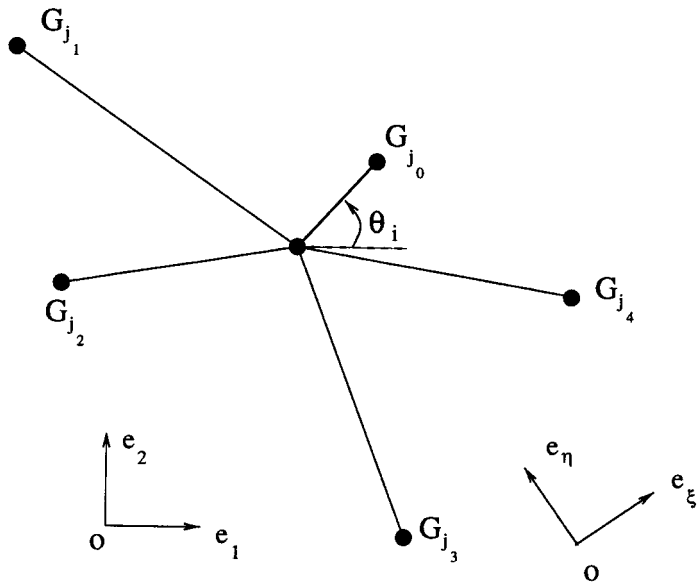

Figure 6. First guess for stretching; new orthogonal basis relying on the strongest connection.

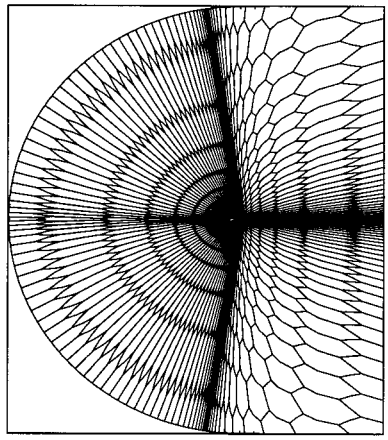

Figure 7. Fine grid: 3014 cells.

already belong to another existing group. If $L_i \leq 1/2$ *or* $L_i \geq 2$*, then put into this group, the neighboring cells* C_j *such that* $j \in S_i$ *(strong connections).*

2. *go to next cell.*

The choice of interval [1/2, 2] is made for coherence with $\epsilon_1 = (1/2)^2$ in (6).

A typical example of the application of this anisotropic coarsening is now presented. Starting from a *C*-type structured mesh around an airfoil we get the fine partition of median cells sketched in Figure 7. It is observed that some vertical and horizontal lines are also stretched. Conversely, the top and bottom right-hand-side region is more isotropic. In the semi-coarsening partition of Figure 8, directional coarsening is evident at the front-left region (two cells in one new cell); isotropic coarsening (fours cells in one new cell) can be observed in the bottom part of the right-hand-side.

3.3. Coarse grid equation for advective term

The convectives fluxes, integrated between two control volumes of the finest grid, are computed in the same way on the coarse grid between two macro-cells. Both conservative

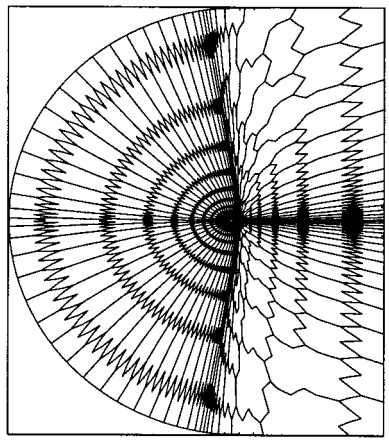

Figure 8. Coarse grid obtained by adaptative semi-coarsening: 1380 cells.

variables and normal vectors are derived from fine grids to coarser ones. The normal vectors, linked to each couple of neighboring coarse macro-cells, result from the summation of the finer grid normal vectors (for the fine cells which have a common boundary with the considered macro-cells). As a result, at most one flux is computed between two given macro-cells, i.e. the number of fluxes to compute *per cell* is comparable with the fine grid case.

3.4. Coarse grid equation for diffusive term

To evaluate diffusive terms on a coarse level, related basis functions are needed. The principle has been proposed by Koobus *et al.* [17] and consists of summing the fine grid basis functions in an algebraic equation point of view.

In the finite element formulation on the fine grid, the equations are integrated and assembled by edges and triangles. As no triangle exists on the coarser grids, it is necessary to define a new edge-based discretization.

On the fine mesh any function f_h can be expressed by:

$$f_h(x, y) = \sum_i f_i \varphi_i(x, y),$$

where f_i is the value of f_h value at a triangulation vertex (x_i, y_i).

The diffusive fluxes can all be written as:

$$\text{Flux}_i = \sum_{j \in K(i) \cup i} f_j \int_\Omega C(x, y) \frac{\partial \varphi_i}{\partial x_l} \frac{\partial \varphi_j}{\partial x_m} \, \mathrm{d}x \, \mathrm{d}y,$$

where $C(x, y)$ contains certain characteristic values of the flow, which are averaged on edges for $i \neq j$.

Let us consider the *fine mesh integrals*:

$$L_{lm}^{ij} = \int_\Omega \frac{\partial \varphi_i}{\partial x_l} \frac{\partial \varphi_j}{\partial x_m} \, \mathrm{d}x \, \mathrm{d}y, \quad l = 1, 2; \quad m = 1, 2; \quad x_1 = x \quad \text{and} \quad x_2 = y. \tag{11}$$

Integrals L_{lm}^{ij} contain the basis functions gradients of the FEM formulation. They are solved on the fine grid and assembled by cells and edges.

A summation of these by neighboring relations on virtual coarse points, for edges connecting two macro-cells I and J, allows us to define the *coarse integrals* (L_{lm}^{IJ}):

$$L_{lm}^{IJ} = \int_\Omega \frac{\partial \varphi_I}{\partial x_l} \frac{\partial \varphi_J}{\partial x_m} \, \mathrm{d}x \, \mathrm{d}y = \sum_{i \in I, j \in J} \int_\Omega \frac{\partial \varphi_i}{\partial x_l} \frac{\partial \varphi_j}{\partial x_m} \, \mathrm{d}x \, \mathrm{d}y, \tag{12}$$

where $\varphi_I = \Sigma_{i \in I} \varphi_i$ and $\varphi_J = \Sigma_{j \in J} \varphi_j$ are *coarse basis functions*.

This is a purely algebraic construction. From the point of view of analysis, the related transfers are not regular enough to ensure that the standard 'approximation property' of multigrid theory [19] is satisfied. In fact, the coarse grid system is not a consistent approximation of the differential equation. For example, for a Cartesian mesh with isotropic coarsening as in Figure 4, a multiplicative factor 1/2 should be introduced for each second-derivative in order to recover consistency. At this point, the agglomeration formulation as proposed in [17] and [20] involves the introduction of a *correction factor*, designed for approximately recovering the consistency of the coarse grid system.

The new feature in the present work is that directional coarsening must be locally accounted for. For example, in the orthogonal mesh case of Figure 5, the correction factor should apply only to vertical second-derivatives. In the general case, a correction factor should be applied

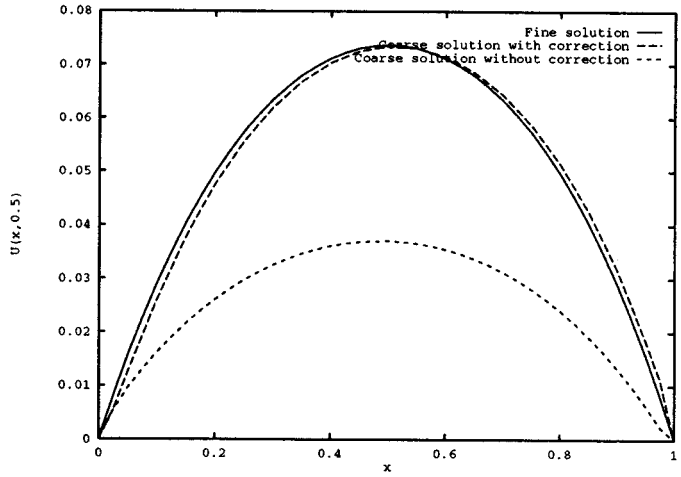

Figure 9. Anisotropic agglomeration for the Poisson problem in a $[0, 1]^2$ square with a uniform structured mesh (Figure 11): fine- and coarse-level solutions produced by the anisotropic coarsening with and without correction term, distribution for $y = 0.5$.

for the derivative $\partial^2/\partial\xi^2$ in the direction ξ of coarsening, and no correction factor for the derivative $\partial^2/\partial\eta^2$ in the orthogonal direction η.

G^{k+1} denotes the coarse grid built from a fine grid G^k. Each basis function gradient of the $k+1$ level is multiplied by a matrix $\mathscr{C}^I_{k,k+1}$ defined by:

$$\mathscr{C}^I_{k,k+1} = (R^k_i)^{-1} \begin{pmatrix} \mathscr{C}^\xi_k & 0 \\ 0 & \mathscr{C}^\eta_k \end{pmatrix} (R^k_i), \tag{13}$$

where $k+1$ denotes the index of the coarse level (coarse grid G^{k+1} built from the fine grid G^k), R^k_i is a rotation matrix for cell C_i from co-ordinates (x, y) to co-ordinates (ξ, η) defined later. Coefficients $\mathscr{C}^\xi_k, \mathscr{C}^\eta_k$ are, respectively, directional corrective factors of the basis function gradient $\vec{\nabla}\varphi_I$ following ξ and η, applied as follows:

$$L^{IJ}_{lm} = \int_\Omega \left(\sum_{n=1}^{2} (\mathscr{C}^I_{k,k+1})_{l,n} \frac{\partial \varphi_I}{\partial x_n} \right) \left(\sum_{n=1}^{2} (\mathscr{C}^J_{k,k+1})_{m,n} \frac{\partial \varphi_J}{\partial x_n} \right) \mathrm{d}x\ \mathrm{d}y. \tag{14}$$

For an isotropic coarsening (see Figures 9 and 11), these correction factors are equal, $\mathscr{C}^\xi_k = \mathscr{C}^\eta_k = \mathscr{C}_k$, and are defined together with the rotation matrix by the following ad-hoc formulas:

$$\mathscr{C}_k = \sqrt{2}\,\frac{(N_k - 1)}{(2N_k - 1)}, \qquad R^k_i = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \text{in (13), (14),} \tag{15}$$

where N_k is an approximation of the number of nodes in one direction (square root of the number of nodes n_k of fine level G^k, see Reference [17] for details).

For an anisotropic coarsening (see Figure 13), these correction factors are put equal to unity in the stretching direction, and to $\mathscr{C}_k$ in the orthogonal direction:

$$\mathscr{C}^\xi_k = \mathscr{C}_k, \qquad \mathscr{C}^\eta_k = 1, \qquad R^k_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i \\ \sin\theta_i & \cos\theta_i \end{pmatrix} \quad \text{in (13), (14),} \tag{16}$$

where θ_i is defined in Equations (7) and (8).

Table II. Poisson equation on a uniform mesh

Cycles	Isotropic MG				Anisotropic MG (with)				Anisotropic MG (without)			
	nivx	μ_{moy}	α_6	*cplx*	*nivx*	μ_{moy}	α_6	*cplx*	*nivx*	μ_{moy}	α_6	*cplx*
Two-G (2, 2)	2	0.202	10	∞	2	0.166	9	∞	2	0.510	22	∞
V-cycle (2, 2)	5	0.227	11	67	5	0.177	9	56	5	0.800	64	392
F-cycle (2, 2)	5	0.175	9	68	5	0.159	9	70	5	0.669	36	273

Comparison between isotropic and anisotropic MG. Two-G is a two-grid variant with complete convergence on coarse level, *V*-cycle and *F*-cycle (2, 2) are multigrid cycles with two Gauss–Seidel sweeps per level and influence of the correction factor (with or without) for the anisotropic MG; (α_6 denotes the number of cycles needed for a six-decade residual reduction, *nivx* denotes the number of grid, μ_{moy} is the mean reduction factor of the residual and *cplx* denotes the arithmetic complexity by node, defined as the number of fluxes for a solution; convergence is as good or better with the anisotropic MG, with a complexity slightly larger for the same number of cycles.

Table III. Poisson equation solution on a stretched mesh

Cycles	Isotropic MG				Anisotropic MG (with)				Anisotropic MG (without)			
	nivx	μ_{moy}	α_6	*cplx*	*nivx*	μ_{moy}	α_6	*cplx*	*nivx*	μ_{moy}	α_6	*cplx*
Two-G (2, 2)	2	0.796	62	∞	2	0.201	10	∞	2	0.454	19	∞
V-cycle (2, 2)	5	0.801	64	391	6	0.224	11	87	6	0.752	50	401
F-cycle (2, 2)	5	0.798	63	476	6	0.227	11	130	6	0.615	30	367

Comparison between isotropic and anisotropic MG and influence of the correction factor (with or without) for anisotropic MG (same symbols as in Table 2).

This device does not ensure that the inter-grid consistency is completely recovered. In fact, this is not strictly true, but a significant part of the error is removed (see Figure 10) and the effect of this correction on convergence will be verified later (Tables II and III). On a uniform mesh, both isotropic agglomeration and anisotropic agglomeration will produce about the same coarse level as shown by Figures 11 and 12.

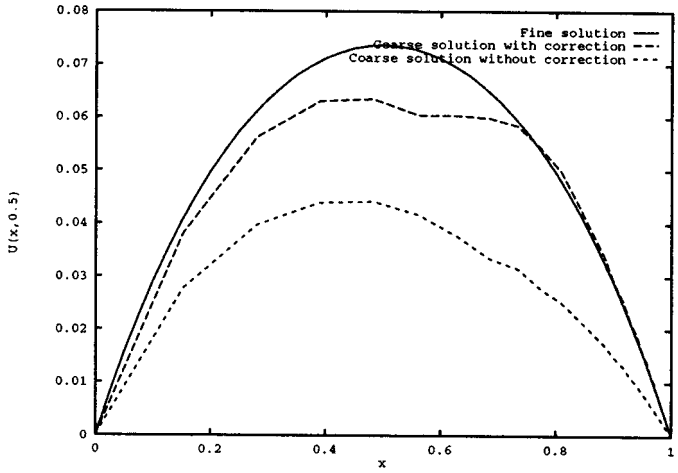

Figure 10. Anisotropic agglomeration for the Poisson problem in a $[0, 1]^2$ square with a stretched structured mesh (Figure 13): fine- and coarse-level solutions by the anisotropic coarsening with and without correction term, distribution for $y = 0.5$.

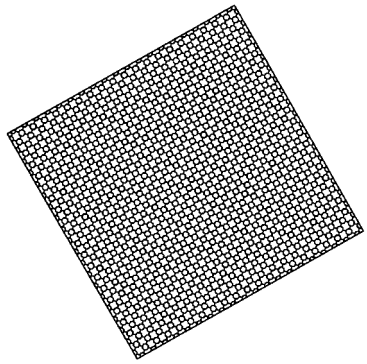

Figure 11. Fine mesh: 1681 cells.

3.5. Multigrid features

The essential features of the linear multigrid solver are standard: correction scheme formulation and V-cycle or F-cycle. We use a nodewise (block 4×4) Gauss–Seidel iteration as a smoother. The details can be found in References [13,20]. The transfer operators are specific to the agglomeration techniques.

- A solution restriction from fine G^k to coarse G^{k+1} is used to rebuild the coarse matrix, it is done by *averaging* fine solutions W^k belonging to the same coarse cell:

$$[R^s_{k,k+1}(W^k)](C^{k+1}_j) = \frac{\sum_{l \in I^k_j} \text{area}(C^k_l) W^k(C^k_l)}{\text{area}(C^{k+1}_j)}$$

  for each coarse cell C^{k+1}_j of G^{k+1}, where I^k_j is the list of subcells of cell C^{k+1}_j, i.e. the set of level k indices I, such that $C^{k+1}_j = \bigcup_{l \in I^k_j} C^k_l$ and where area (C^k_l) denotes the area of cell C^k_l of G^k.
- The residual restriction from fine G^k to coarse G^{k+1} is done by *summing* fine level residuals res^k belonging to the same coarse cell:

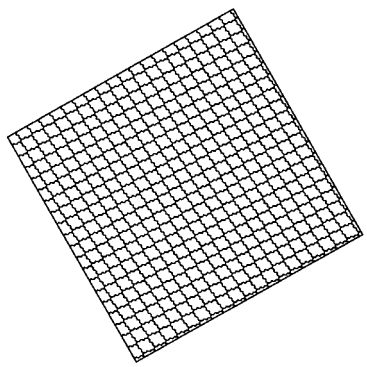

Figure 12. Coarse grid: 441 cells.

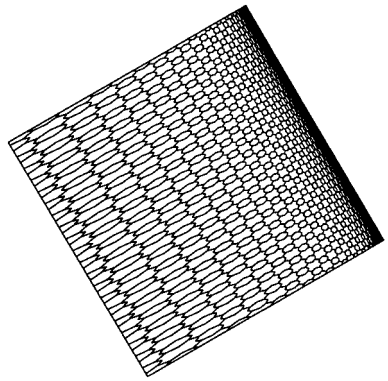

Figure 13. Stretched fine mesh: 1681 cells.

$$[R^{rhs}_{k,k+1}(\mathrm{res}^k)](C_j^{k+1}) = \sum_{l \in I_j^k} \mathrm{res}^k(C_l^k)$$

for each coarse cell C_j^{k+1} of G^{k+1}.

- The correction prolongation from coarse (G^{k+1}) to fine (G^k) is composed of a *trivial injection* of correction e^{k+1}:

$$[P_{k+1,k}(e^{k+1})](C_j^k) = e^{k+1}(C_l^{k+1})$$

for each fine cell C_j^k of G^k, where l is the index of the coarse cell C_l^{k+1} which contains fine cell C_j^k.

4. APPLICATION TO A LINEAR SCALAR PROBLEM

We first consider the application of a linear MG cycle built using the above options for solving the Poisson problem from Equation (1) with $\vec{V} = \vec{0}$ and $f = 1$ on several types of meshes with Dirichlet condition on the boundary. The approximation is a purely P1-Galerkin one (linear

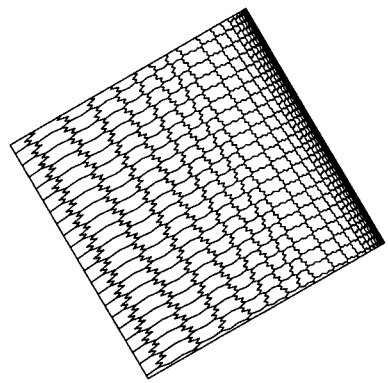

Figure 14. Coarse grid obtained by adaptive semi-coarsening: 761 cells.

Table IV. Poisson equation solution on a flat plate mesh

Cycles	Isotropic MG				Anisotropic MG (with)				Anisotropic MG (without)			
	nivx	μ_{moy}	α_6	*cplx*	*nivx*	μ_{moy}	α_6	*cplx*	*nivx*	μ_{moy}	α_6	*cplx*
Two-G (2, 2)	2	0.902	135	∞	2	0.191	10	∞	2	0.472	20	∞
V-cycle (2, 2)	5	0.903	137	827	6	0.384	16	132	8	0.817	70	588
F-cycle (2, 2)	5	0.906	141	1041	6	0.239	11	141	8	0.683	38	522

Comparison between isotropic and anisotropic MG and influence of the correction factor (with or without) for anisotropic MG (same symbols as in Table 2).

elements). The number of levels is taken to be large enough to have no influence on convergence rate.

4.1. Structured uniform mesh

The first mesh considered is a square 1×1 and not stretched. The mesh contains 41×41 points (see Figure 11). A rotation was applied to avoid alignment of mesh with Cartesian co-ordinates.

It is shown in Table II that convergence is equally good with both isotropic and anisotropic agglomeration algorithms. Since the problem is isotropic, the corresponding coarse levels are not very different; main differences arise near the boundary.

4.2. Stretched structured mesh

The mesh is topologically equivalent to the mesh of Section 4.1, but with a geometrical progression starting from one side (see Figure 13). The maximal cell aspect ratio is 100. The semi-coarsening algorithm shows (Figure 14) three types of behaviour: the left part of mesh is directionally coarsened in one direction, the right part is directionally coarsened in the orthogonal direction, and the median part is fully coarsened.

From Table III, it can be seen that the isotropic algorithm has difficulties. The anisotropic one does not perform much worse than in the previous isotropic test case.

4.3. 'Flat plate' mesh

A mesh usually used for calculations of a turbulent flow on a flat plate (113×81 nodes) is considered. The maximum aspect ratio is 5000. The iterative solution of the Poisson equation is examined. The effect of such a stretching on the standard MG approach with the regular coarsening is obvious: a loss in convergence speed is observed and the reduction factor is 0.90 (see Table IV). Contrarily, the anisotropic semi-coarsening strategy proves again to be much less sensitive to stretching (reduction factor is 0.24; Table IV). Again the correction factor shows its dramatic influence (Table IV).

4.4. Some comments

Performances with the diffusion–advection model are generally better than for the Poisson case (see References [20,17] for details). Although more costly in the above experiments, the *F*-cycle appears to have a convergence less sensitive to test cases. However, the *V*-cycle is generally more efficient and will be used in the rest of the paper.

5. EXTENSION TO NAVIER–STOKES EQUATION

5.1. Physical model

The governing equations are obtained by applying Reynolds averaging to the compressible Navier–Stokes equations, and by modelling the Reynolds stress by the Boussinesq assumption; this results in the well-known (k-ε) model of Launder–Spalding [21]. The equations can be written in a conservative form as:

$$\frac{\partial W}{\partial t}+\frac{\partial F(W)}{\partial x}+\frac{\partial G(W)}{\partial y}=\frac{1}{Re}\left(\frac{\partial R(W)}{\partial x}+\frac{\partial S(W)}{\partial y}\right)+\frac{\partial \tilde{R}(W)}{\partial x}+\frac{\partial \tilde{S}(W)}{\partial y}+\Omega(W),$$

where

- $W(x,\ y, t)$ is vector function of $\mathbb{R}^6$, the components of which are the non-dimensional conservative variables $(\rho, \rho u, \rho v, E', \rho k, \rho\varepsilon)^T$
- $F(W)$ and $G(W)$ are the functions of the convective fluxes
- $R(W)$, and $S(W)$ are the functions of laminar viscous fluxes and Re is the laminar Reynolds number
- $\tilde{R}(W)$, and $\tilde{S}(W)$ are the functions of turbulent viscous fluxes
- $\Omega(W)$ is the source term of the turbulence

In order to treat the hyperbolic terms in the same way as in the laminar case, an appropriate change of variables is used. We introduce:

$$\begin{cases} p' = p+\dfrac{2}{3}\rho k \\ E' = E+\beta\rho k \quad \text{and} \quad \beta = -1+\dfrac{2}{3(\gamma-1)}, \end{cases}$$

where

$$\begin{cases} E = \rho C_v T+\dfrac{1}{2}\rho(u^2+v^2)+\rho k \\ p = (\gamma-1)\rho C_v T, \quad \gamma = 1.4 \end{cases},$$

where p is the pressure, E the total energy per unit volume, ρ the fluid density, k the turbulent kinetic energy, C_v denotes the specific heat at constant volume, T the temperature and u, v are the components of fluid velocity.

The relation between E' and p' is written the classical way as:

$$p' = (\gamma-1)\left(E'-\frac{1}{2}\rho(u^2+v^2)\right).$$

Thus, the change of variables introduced in the initial system allows the classical conservative form of the physical variables to be retained.

μ_t, ω_k, ω_ε respectively, denote the turbulent viscosity and the components of source term for k, ε:

$$\begin{cases} \mu_t = c_\mu\dfrac{\rho k^2}{\varepsilon} \\ \omega_k = -\rho\varepsilon+\mathscr{P} \\ \omega_\varepsilon = c_{\varepsilon 1}\dfrac{\varepsilon}{k}\mathscr{P}-c_{\varepsilon 2}\dfrac{\rho\varepsilon^2}{k} \end{cases}$$

where

$$\mathscr{P} = -\left(\frac{2}{3}\rho k\delta_{ij} - \mu_t\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_k}\delta_{ij}\right)\right)\frac{\partial u_i}{\partial x_j},$$

where $c_\mu = 0.09$, $c_{\varepsilon 1} = 1.44$ and $c_{\varepsilon 2} = 1.92$ are the empirical constants of the modeling, ε denotes the turbulent dissipation rate and $\mathscr{P}$ the production term of the turbulent. The laminar Prandtl number is $Pr = 0.725$ and the turbulent Prandtl number is $Pt = 0.86$.

5.1.1. Near-wall treatment. The above transport equations on k and ε are obtained by assuming that the turbulent effects are dominant in the flow domain [22]. Thus, the standard k–ε model is not valid in regions where the viscous effects are large compared with the turbulent effects (near-wall zones).

In order to account for low-Reynolds number effects (near-wall effects), Chen and Patel [23] proposed to solve the k–ε equations only in the high-Reynolds number regions and to use a low-Reynolds number one-equation model in the near-wall region. In many cases, one-equation models show a better agreement with experiments near the wall than standard two-equation models [24]. But the main advantage is that one-equation models require less mesh nodes in the viscous sublayer than low-Reynolds two-equation models and this increases the computational efficiency and convergence properties of the numerical method. Following the approach of Chen and Patel [23], the one-equation low-Reynolds number model of Wolfshtein [25] has been included in the present method. In regions adjacent to the surface, where $R_y < 200$, the mean-flow equations and the equation for the turbulent kinetic energy are solved, and the characteristic length scales are determined via algebraic relations. In these regions, the eddy viscosity employed is defined by

$$\mu_t = c_\mu \rho \sqrt{k} f_\mu, \qquad f_\mu = \sqrt{\frac{\rho_w}{\rho}}\, l_\mu,$$

where ρ_w is the fluid density of the wall, and the dissipation rate of turbulent kinetic energy employed for the modelled source terms in the equation for k is given by

$$\varepsilon = \frac{k^{3/2}}{l_\varepsilon},$$

with l_μ and l_ε are two algebraic length scales, defined in the following way:

$$l_\mu = C_1 y\left(1 - \exp\left(\frac{-R_y}{A_\mu}\right)\right) \qquad l_\varepsilon = C_1 y\left(1 - \exp\left(\frac{-R_y}{A_\varepsilon}\right)\right).$$

The damping functions, which contain the turbulent Reynolds number R_y, are introduced to mimic the correct behaviour near the wall and for matching with the region where the flow is totally dominated by effects due to the molecular viscosity. The model requires two constants, which are

$$C_1 = \kappa c_\mu^{-3/4}, \qquad A_\mu = 70, \qquad A_\varepsilon = 2C_1,$$

where κ and c_μ take the usual values of 0.45 and 0.09.

A detailed discussion of the merits of this model compared with genuine low-Reynolds two-equations models, showing that it is a good compromise, is presented in [24].

5.2. Time-implicit upwind scheme

It is well known that a global Newton iteration starting from an arbitrary initialization cannot be applied to a compressible flow. In order to progressively approach the convergence domain of a (modified) Newton iteration, a backward-Euler linearized implicit time advancing scheme is constructed, with the following features:

- A Roe approximate Riemann solver is used for the approximation of the fluid convective terms; the positivity-preserving multi-component Riemann flux proposed in Reference [26] is considered for the turbulence convective terms.
- Linearization: except for the production part in the source term, the linearization is obtained by freezing the Jacobian in Roe's flux difference splitting for convective terms, and by exact differentiation of diffusion and source terms. Also, the turbulent viscosity μ_t is frozen so that the k and ε variables are implicitly coupled to each other but uncoupled from the other four flow variables.
- Implicit first-order preconditioning: it is performed using a first-order Godunov scheme because it is tridiagonal in 1D and better conditioned than the second-order one, which produces larger matrices (pentadiagonal in 1D).
- Local time stepping and time step incrementation: a local time step is computed on each cell so that the Courant number (denoted by 'CFL' in the sequel) is somewhat uniform on the mesh. The CFL number is specified as an increasing function of time and of the non-linear residual L_2 norm, in order to ensure the progressive switch from the unsteady phase to the asymptotic convergence.
- Each time step involves the solution of the two linear systems (mean flow and closure variables) by an agglomeration multigrid method described in next subsection.
- Each computation is started from a uniform flow.

5.3. Multigrid solution of the linear system

The linearized system to be solved is essentially of the same type as the advection–diffusion model. Its isotropic MG treatment is proposed in Reference [13] and some main features of that approach and some differences with Reference [13] are now recalled.

Main features: coarse grid systems rely on finite volume for the (first-order-accurate) advective terms and on integrals L_{lm}^{IJ} (see (12)) for other terms. Variable coefficients arising from the freezing of flow field are derived from restriction of the flow field variables, and edgewise constant values are taken for simplifying the building of edge-based diffusion terms.

Novelties with respect to Reference [13]: several important quantities are transferred to coarse grids: friction velocity, normal distance to wall, turbulent local Reynolds number R_y, wall laminar viscosity v_t. The reduction of unknowns in the layer is obtained on each level by putting non-diagonal terms for ε to zero, and fixing the diffusion of ε in (block) diagonal terms.

MG-cycling options: the V-cycle with two sweeps per level is used. Only one cycle per time step is performed.

6. APPLICATION TO FLOW PROBLEMS

6.1. Laminar flow around an airfoil

The introduction of the semi-coarsening method in the laminar Navier–Stokes solver is now considered. The farfield Mach number is 0.8, the angle of attack is 10°, the Reynolds number

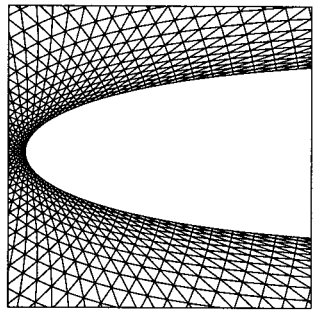

Figure 15. Zoom of the airfoil mesh near leading edge.

is 73 and the wall is adiabatic. The mesh used (involving about 14 000 nodes) is stretched near the airfoil and in other places according to a C-type structured topology (see Figures 15 and 16). The parameter β for the MUSCL interpolation (see (4)) is equal to 1/2 (without limitation procedure) which results in From's half-upwing biasing. The CFL is a linear function of the time iteration number (kt): $\mathrm{CFL} = 1000 + 500 \times kt$. Figure 17 depicts the pseudo-time non-linear convergence of the implicit scheme when it is equipped by the present MG scheme. When the isotropic version is used, the convergence and efficiency are better than with an equivalent single grid approach, but its convergence in 878 iterations (for ten decades) makes it rather disappointing compared with the computation performed with a non-stretched mesh. The anisotropic semi-coarsened version is much better with a convergence in 153 iterations and is four times more efficient in terms of computer time (see Table V). In Figure 18, the effect of the correction factor is analyzed.

6.2. Turbulent supersonic flow over a flat plate

A second test case is the computation of a turbulent flat plate flow. The case was introduced and experimented on by Mabey (see Reference [27]; farfield Mach number is 4.52 and Reynolds number is 28.2×10^6). This flow was chosen recently as a test case for the ETMA Workshop [28] for which a common mesh involving 113×81 nodes was provided by Vrije University of Brussels. This mesh, for which the first node row corresponds to a Y^+ less than 1, was used in this study. The mesh is highly stretched with an aspect ratio near wall of 5000. The turbulence model chosen is a $k-\epsilon$ one with a two-layer Chen-Patel [23] treatment. The parameter β for the MUSCL interpolation (see (4)) is equal to 1/2 for all the variables. We

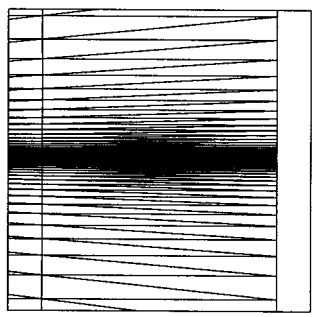

Figure 16. Zoom of the mesh on a region of the wake.

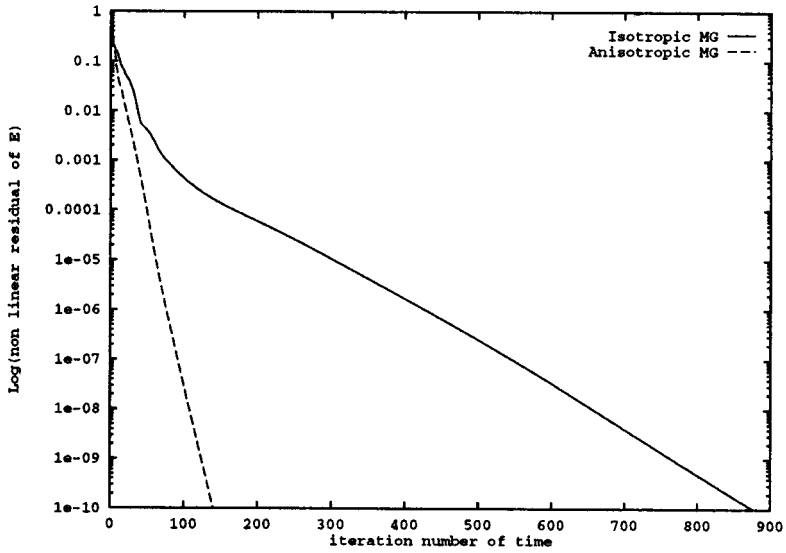

Figure 17. Laminar flow past a NACA airfoil: convergence history for the isotropic MG (878 iterations) and the anisotropic one (153 iterations).

use a limiter derived Van Albada and Van Leer [14] for variables k and ε for which positive values are compulsory. The CFL is a linear function of the time iteration number (kt): $\mathrm{CFL} = 10 + 10 \times kt$. In Figures 19 and 20, an idea of the weak level of model and numerical errors can be seen (mesh convergence is observed as good, except at proximity of the uniform flow and a fairly good agreement with measurements of Mabey is verified, see also Reference [20]). In Figure 21, three convergence histories are presented, obtained with an implicit time advancing in which the linearized system is solved by three different algorithms, but with about the same CPU time per time iteration (see also Figure 22). In Figure 23, the effect of the correction factor is analyzed (see also Table VI).

The slower convergence is obtained with approximately ten linear Gauss–Seidel (single grid) sweeps for each time step; after a rather good initial phase, convergence slows to a disappointing rate. The second convergence results from the application of a MG cycling relying on isotropic coarsening. According to theory, asymptotic convergence of MG should not be much better than single grid on this very stretched mesh; confirmation is given by convergence history: after a phase of better convergence than single grid, convergence degrades to an analogous asymptotic rate, proving that, for example, in a four-decade converged solution, a component of the error was not damped by a factor of ten.

Table V. Laminar flow past a NACA airfoil

Algorithm	Isotropic MG	Anisotropic MG
CPU time (min)	63	12

CPU time using the two MG methods on Dec Alpha 600/266 MHz workstation.

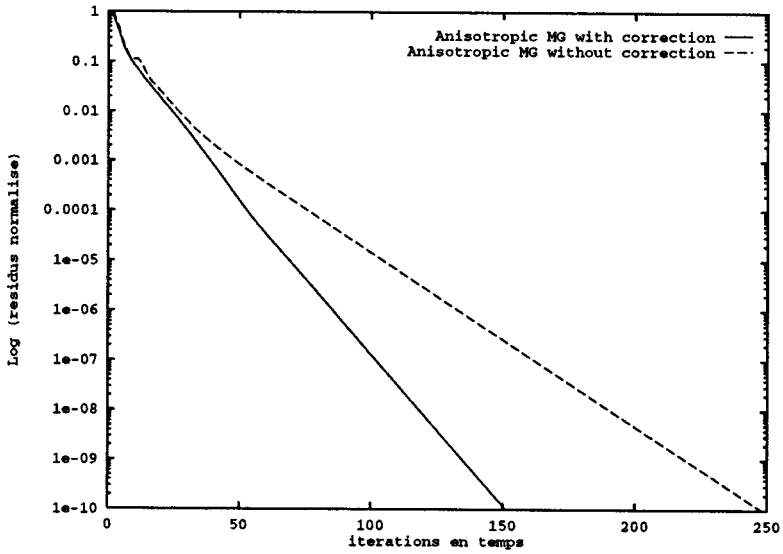

Figure 18. Laminar flow past a NACA airfoil: convergence history for the anisotropic MG with correction factor (153 iterations) and without correction factor (250 iterations).

Conversely, a quasi-constant convergence rate is obtained with the new anisotropic algorithm. Furthermore, the convergence is essentially not sensitive to mesh size (Figure 22).

6.3. *Turbulent subsonic flow around an airfoil*

We consider now the numerical simulation of the high Reynolds number past a RAE2822 airfoil. The RAE2822 aerofoil flow has been tested by Cook *et al.* [29] over a range of transonic flow conditions in the wind tunnel at RAE Farnborough. Boundary layer transition

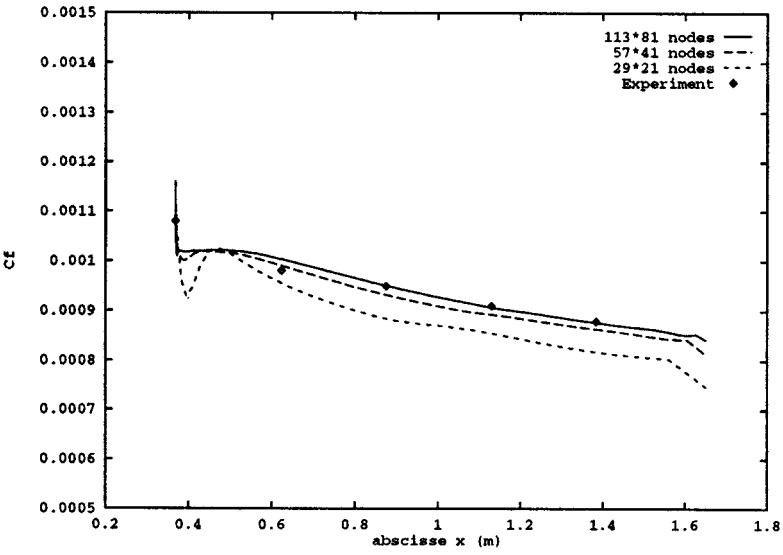

Figure 19. Mabey flat plate test case for 29×21, 57×41, 113×81 meshes: friction coefficient.

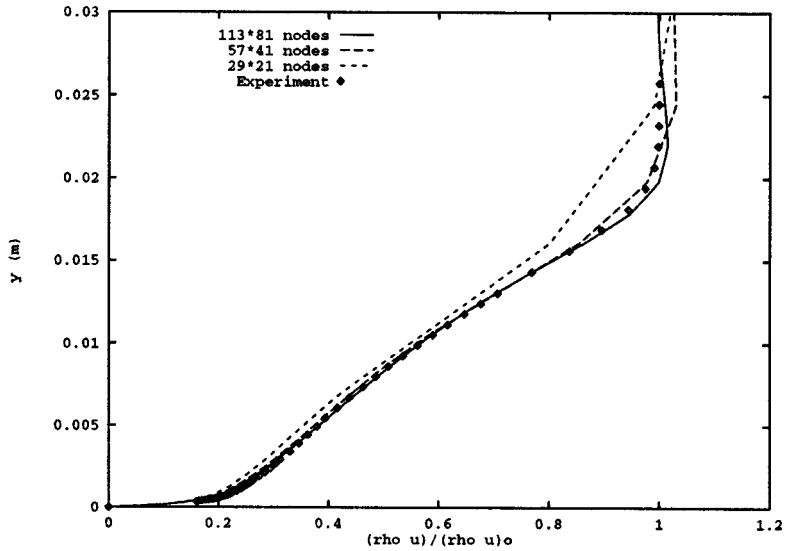

Figure 20. Mabey flat plate test case for 29×21, 57×41, 113×81 meshes: x-moment on $x = 1.384$.

was fixed near the leading edge of the aerofoil ($x/c = 0.03$) by a rough transition strip. Two typical cases for the validation of the turbulence model and of computational methods are chosen (Case 6 and Case 9).

The grid obtained from a direct triangulation of a *C-type* 267×65 structured grid is shown in Figure 24 with 193 nodes on the aerofoil surface, 32 in the wake and 65 nodes in the normal direction of the profile. The mesh is highly stretched with a near wall aspect ratio of 2400. The normal distance of the first node was taken as 7.9×10^{-6}. The parameter β for the MUSCL

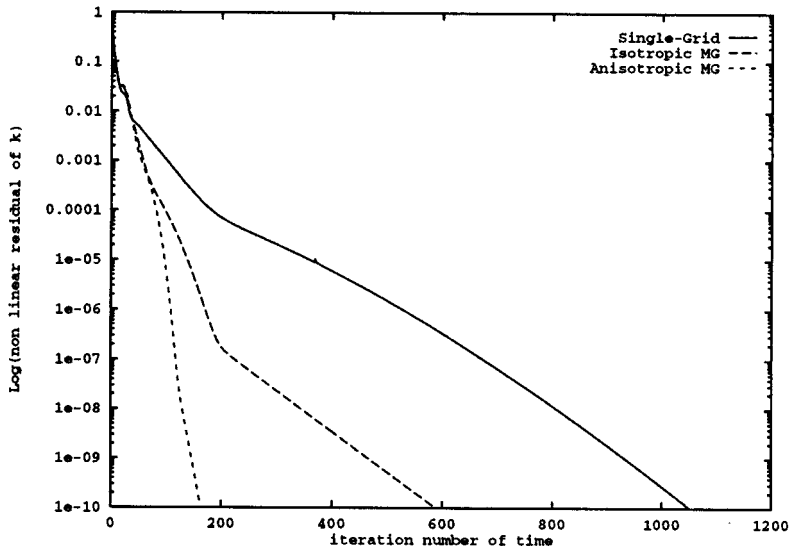

Figure 21. Turbulent flow past a flat plate for 113×81 mesh: convergence history for the single grid implicit algorithm (1050 iterations), isotropic MG (580 iterations) and the anisotropic MG (165 iterations).

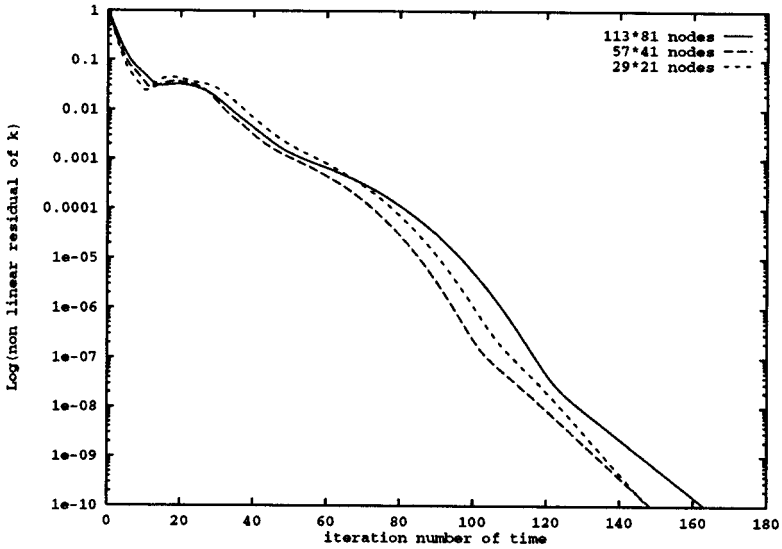

Figure 22. Turbulent flow past a flat plate for 29×21, 57×41, 113×81 meshes: convergence histories for the anisotropic MG algorithm.

interpolation is equal to 1/5 for the fluid variables and is equal to 1/3 for the turbulent variables. The limiter proposed by Spekreijse [30] for variables k and ε is used here. The CFL is a linear function of the time iteration (kt) number:

$$\mathrm{CFL} = 10 + 10 \times kt.$$

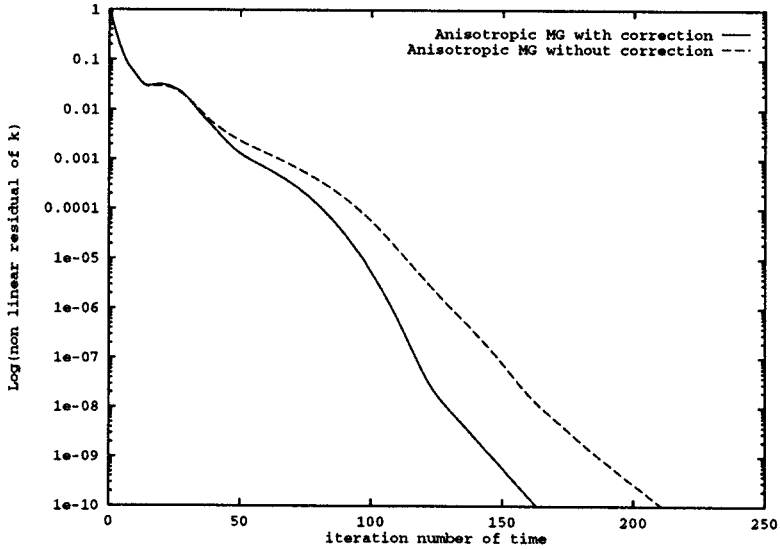

Figure 23. Turbulent flow past a flat plate for 113×81 mesh: convergence history for the anisotropic MG with correction factor (165 iterations) and without correction factor (215 iterations).

Table VI. Turbulent flow past a flat plate

Algorithm	Single grid	Isotropic MG	Anisotropic MG
CPU time (min)	65	42	13

CPU time using the two MG methods on Dec Alpha 600/266 MHz workstation.

6.3.1. Case 6. Case 6 is a supercritical case with a moderate shock-wave on the airfoil upper surface. The farfield Mach number is 0.725, the angle of attack is 2.54°, and the Reynolds number is 6.5×10^6.

For this last case, a convergence of five decades is obtained with the single grid algorithm, but not very easily; indeed, a quite slow asymptotic convergence, with a total of 1949 time iterations is observed (Figure 25, see also Figures 26 and 27). When the MG algorithm is

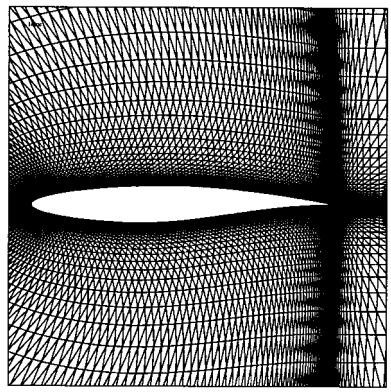

Figure 24. Zoom of the RAE2822 airfoil mesh.

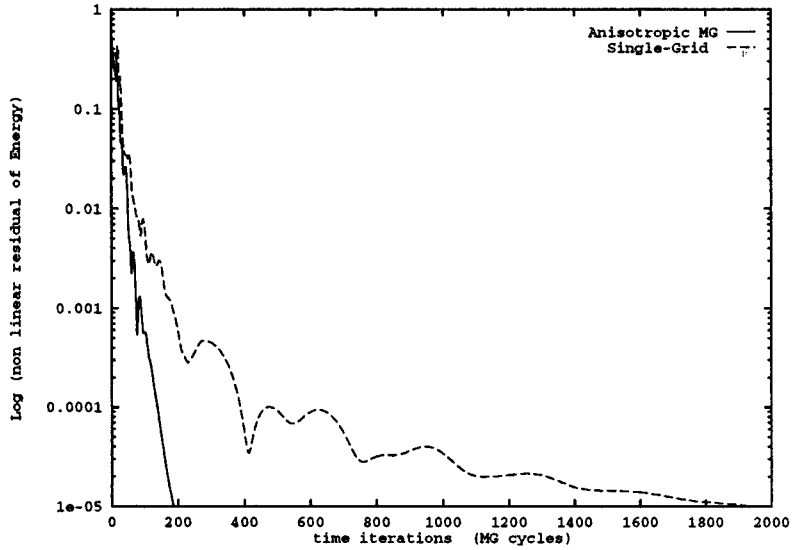

Figure 25. Turbulent flow past a RAE2822 airfoil (Case 6): convergence history for the single grid implicit algorithm (1949 iterations) and the anisotropic MG (188 iterations).

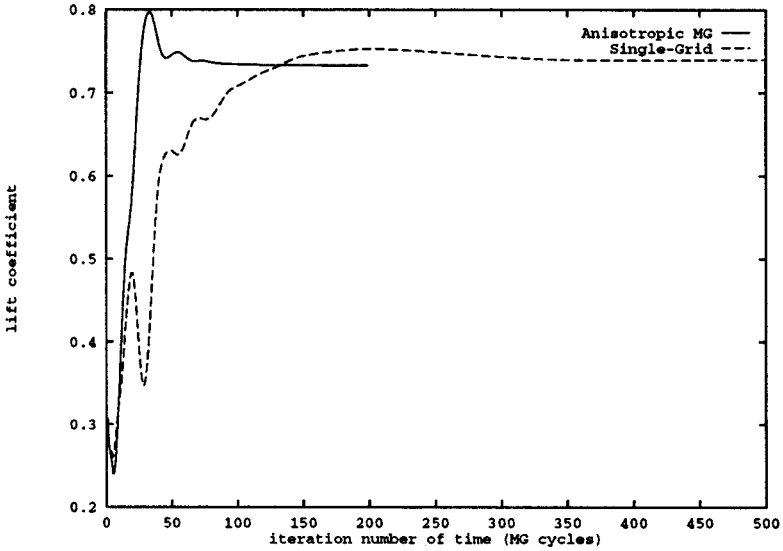

Figure 26. Turbulent flow past a RAE2822 airfoil (Case 6): comparison of the lift coefficient evolution as a function of time-iteration.

applied, the improvement is evident, with a convergence in 188 iterations (Figure 25). In Figure 28, the effect of the correction factor is analyzed. In Figure 26 it is verified that convergence of the lift coefficient is slow with the single grid algorithm and fast with the MG algorithm (Table VII). In fact, a sufficient convergence of mean values is obtained with 122 iterations (Table IX).

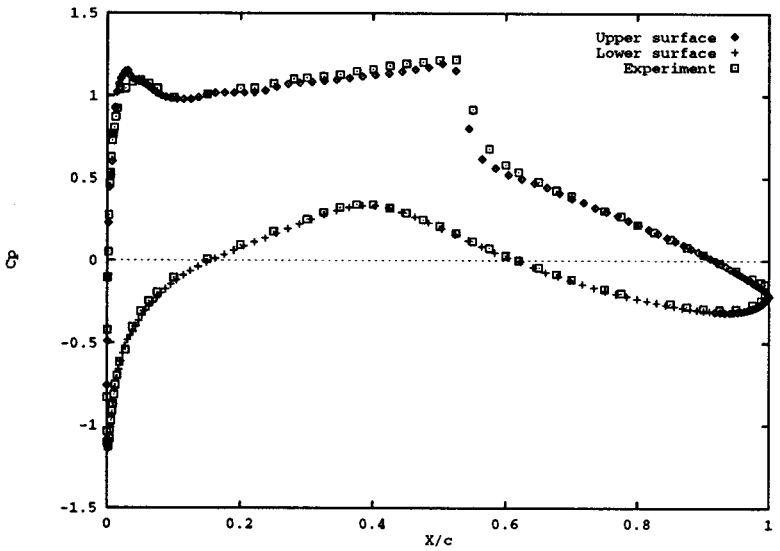

Figure 27. Turbulent flow past a RAE2822 airfoil (Case 6): comparison of computed pressure coefficient with experimental measurements of Cook *et al.* [29].

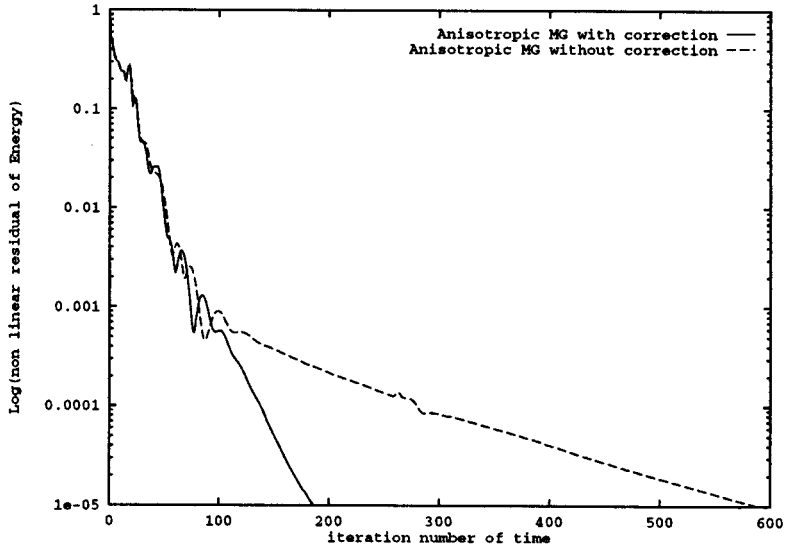

Figure 28. Turbulent flow past a RAE2822 airfoil (Case 6): convergence history for the anisotropic MG with correction factor (188 iterations) and without correction factor (591 iterations).

The results can be compared with experimental results in Table VIII and Figure 27; the improvement in CPU time is between a factor of 7.8 (see Table VII) and 11.9 (considering 122 iterations for MG (Table IX).

6.3.2. Case 9. Case 9 is a popular supercritical one. The farfield Mach number is 0.73, the angle of attack is 2.79°, and the Reynolds number is 6.5×10^6.

For this case, the single grid implicit scheme needs 2534 time iterations to reach a steady state; this is to be compared with the MG scheme which needs 174 time iterations (see Table X, Figures 29 and 30).

The results can be compared with experimental results in Table XI and Figure 31; the improvement in CPU time is between a factor of 10.9 (see Table X) and 17.3 (considering 114 iterations, see Table XII). This illustrates the fact that MG is not an algorithm for easy problems, on the contrary, it shows a better behavior for difficult problems. In Figure 32, the effect of the correction factor is analyzed.

6.4. Single versus multi-coarsening: a discussion

In the case of structured meshes, several authors (Mulder [5], Radespiel and Swanson [31]) recommend the application of a multi-coarsening strategy. This consists of considering the two directions of the mesh for semi-coarsening, then two coarse meshes are obtained $(\Delta x, 2\Delta y)$ and $(2\Delta x, \Delta y)$. At the next coarser level, four coarse meshes are obtained, etc.

The advantage is increased robustness; e.g. viscous terms integrated on a stretched region may require a semi-coarsening for $(\Delta x, 2\Delta y)$, while advective mesh alignment may require a $(2\Delta x, \Delta y)$ semi-coarsening in order to account for the 1D character of advection (along streamline).

The main disadvantage is the extra cost, since in 2D each level may be as costly as the fine level; then complexity for FMG will not be $O(N)$ but $O(N \log(N))$. The problem is still harder in 3D!

In this study, it can be observed that the dual-mesh triangular approximation involves a large enough amount of transverse numerical viscosity so that advective mesh alignment will be very rarely encountered. Thus, single coarsening is sufficiently robust and much less expensive in terms of computational cost. The above experiments show that our assumption is reasonable, at least for the test cases computed.

7. CONCLUSION

The main question addressed in this paper was whether MG unstructured methods could perform as well on stretched meshes as they do on isotropic ones. Indeed, it has been stressed that the theory seems to indicate that convergence should be at least as good for an anisotropic algorithm applied to a stretched mesh as for an isotropic algorithm applied to an isotropic (non-stretched) mesh. In other words, the only source of additional cost should be the fact that semi-coarsened meshes contain more nodes than fully coarsened ones.

In this study, it was demonstrated by numerical experiments that the isotropic algorithm (which performs well for isotropic meshes) has a much slower convergence when applied to

Table VII. Case 6: CPU time for the two methods on a Dec Alpha 600/266 MHz workstation for a non-linear residual smaller than 10^{-5} (see Figure 25)

Algorithm	Single grid	Anisotropic MG
CPU time (min)	227	29

Table VIII. Case 6: lift coefficient C_L, pressure drag coefficient $C_D(P)$, friction drag coefficient $C_D(V)$, total drag coefficient $C_D(T)$ and pitching moment coefficients $C_m(1/4)$ for experiment and computation fully iteratively converged

Case 6	C_L	$C_D(P)$	$C_D(V)$	$C_D(T)$	$C_m(1/4)$
Experiment [29]	0.7430	—	—	0.01270	−0.0950
$k-\varepsilon$ converged	0.7336	0.01342	0.00638	0.01980	−0.0948

Table IX. Case 6: lift coefficient C_L, pressure drag coefficient $C_D(P)$, friction drag coefficient $C_D(V)$, total drag coefficient $C_D(T)$ and pitching moment coefficients $C_m(1/4)$ for experiment and computation converged with 122 time iterations

Case 6	C_L	$C_D(P)$	$C_D(V)$	$C_D(T)$	$C_m(1/4)$
Anisotropic MG	0.7339	0.01345	0.00637	0.01982	−0.0948
Single grid	0.7208	0.01526	0.00635	0.02161	−0.0967

Table X. Case 9: CPU time for the two methods on a Dec Alpha 600/266 MHz workstation for a non-linear residual smaller than 10^{-5} (see Figure 29)

Algorithm	Single grid	Anisotropic MG
CPU time (min)	295	27

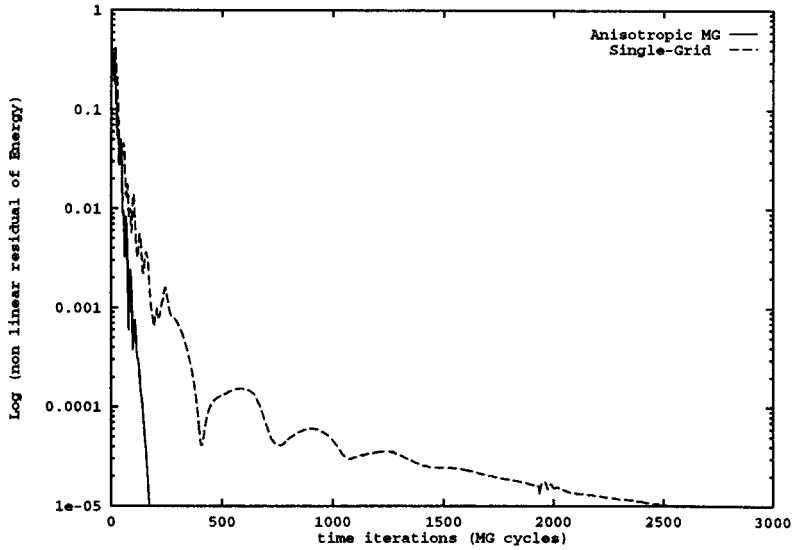

Figure 29. Turbulent flow past a RAE2822 airfoil (Case 9): convergence history for the single grid implicit algorithm (2534 iterations) and the anisotropic MG (174 iterations).

stretched meshes in the case of a model problem (Poisson equation), as well as for complex compressible flows. In some cases, asymptotic convergence is not significantly better than for the analogous single grid scheme.

We have proposed an anisotropic algorithm for which experiments show the expected improvement:

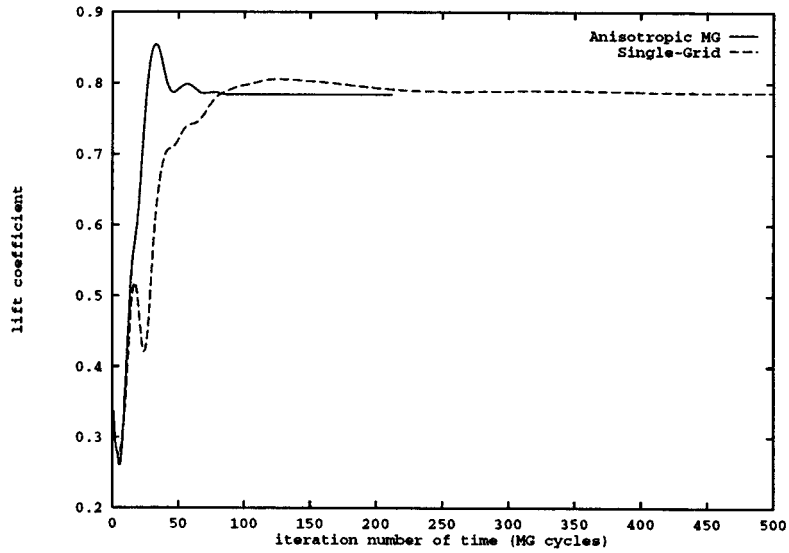

Figure 30. Turbulent flow past a RAE2822 airfoil (Case 9): comparison of the lift coefficient evolution as a function of time-iteration.

Table XI. Case 9: lift coefficient C_L, pressure drag coefficient $C_D(P)$, friction drag coefficient $C_D(V)$, total drag coefficient $C_D(T)$ and pitching moment coefficients $C_m(1/4)$ for experimental and computation fully iteratively converged

Case 9	C_L	$C_D(P)$	$C_D(V)$	$C_D(T)$	$C_m(1/4)$
Experiment [29]	0.8030	—	—	0.01680	−0.099
$k-\varepsilon$ converged	0.7877	0.01699	0.00632	0.02331	−0.0986

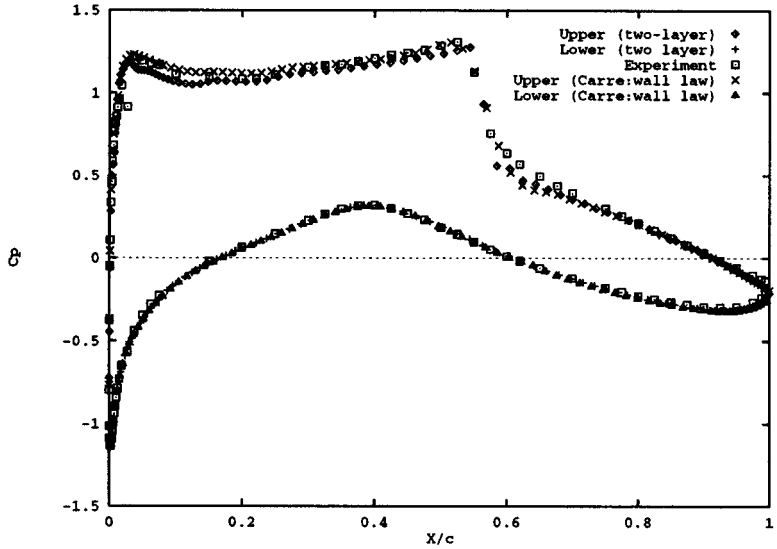

Figure 31. Turbulent flow past a RAE2822 airfoil (Case 9): comparison of computed pressure coefficient, obtained with the two-layer model and the wall law model (Carré [13]), with experimental measurements of Cook *et al.* [29].

- for the elliptic model problem, convergence becomes quasi-insensitive to mesh stretching: while $\approx 9-10$ cycles are used for the isotropic MG on non-stretched meshes, the anisotropic MG requires 10–12 cycles for the same computation, but also $\approx 10-12$ cycles for a stretched mesh computation;
- for the compressible flows considered, the poorer asymptotic convergence rate of the isotropic version is not observed for the new anisotropic one; inversely, convergence rate is rather constant throughout the convergence path.

Table XII. Case 9: lift coefficient C_L, pressure drag coefficient $C_D(P)$, friction drag coefficient $C_D(V)$, total drag coefficient $C_D(T)$ and pitching moment coefficients $C_m(1/4)$ for experiment and computation converged with 114 time iterations

Case 9	C_L	$C_D(P)$	$C_D(V)$	$C_D(T)$	$C_m(1/4)$
Anisotropic MG	0.7879	0.01705	0.00632	0.02337	−0.0987
Single grid	0.7745	0.01913	0.00630	0.02543	−0.1018

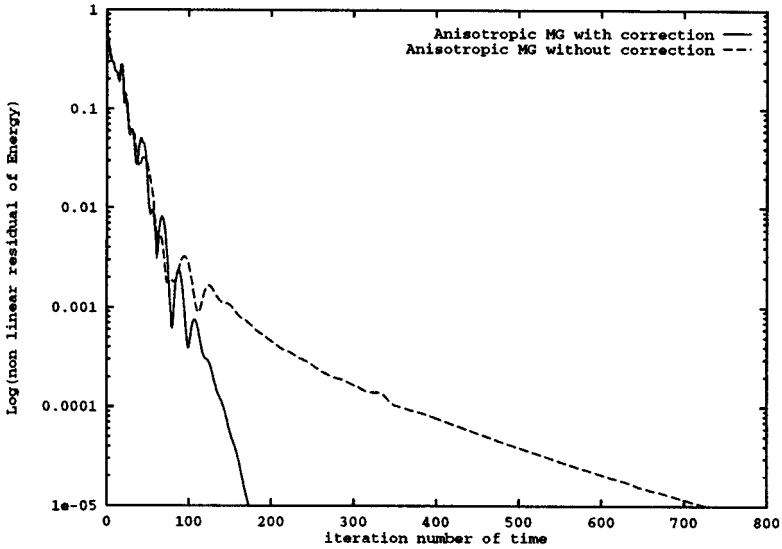

Figure 32. Turbulent flow past a RAE2822 airfoil (Case 9): convergence history of the anisotropic MG with correction factor (174 iterations) and without correction factor (728 iterations).

This last point is essential. The slowest error mode is generally not solved with a strategy in which convergence is stopped after some cycles of quasi-stagnation. The fact that, for the isotropic version, this happens when the residual has already achieved a low level can be misleading, especially when shocks and boundary layers can interact, and may yield erroneous predictions. It may happen in other cases that the slower error mode is large at initial conditions and that the poor asymptotic rate appears early.

A weak dependence of convergence on the mesh size has also been observed.

However, the presented algorithm may not benefit from these ideal properties in all cases, and further experiments and improvements are necessary to qualify it as a kind of ultimate anisotropic MG method in 2D.

Also, most calculations are to be done with locally structured stretched meshes because both good unstructured and highly stretched meshes are still difficult to build.

For the extension to 3D, although no explicit use of the two-dimensionality is made in building the present method, some—hopefully resolvable—additional difficulties are anticipated in the derivation of a semi-coarsening algorithm ensuring efficiency for most types of meshes.

ACKNOWLEDGMENTS

The authors would like to thank Gilles Carré, Bruno Koobus, Jean Antoine Désidéri and Hervé Guillard from INRIA Sophia-Antipolis for their valuable advice and suggestions. The authors would also like to acknowledge Franck Cantariti from University of Glasgow, Michel Ravachol from Dassault-Aviation and Erbing Shang from University of Brussels for providing meshes and for the fruitful discussions.

APPENDIX A. NOMENCLATURE

$c_f = 2\tau_w/\rho_\infty u_\infty^2$	friction coefficient
C_p	specific heat at constant pressure
C_v	specific heat at constant volume
E	total energy per unit volume
k	turbulence kinetic energy
P	pressure
Pr	Prandtl number
Pt	turbulent Prandtl number
$\mathscr{P}$	eddy production term
$R_y = \sqrt{\rho/\rho_w}\sqrt{k}y/v_w$	local Reynolds number
T	temperature
u	horizontal velocity
$u_f = \sqrt{\tau_w/\rho}$	friction velocity
v	vertical velocity
x	horizontal co-ordinate
y	vertical co-ordinate
$y^+ = u_f y/v_w$	non-dimensional y
$\gamma = C_p/C_v$	specific heat ratio
ε	turbulent dissipation ratio
μ	molecular viscosity
μ_t	eddy viscosity
$v = \mu/\rho$	kinetic molecular viscosity
ρ	fluid density
τ	laminar stress tensor
τ_t	turbulent stress tensor
τ_w	wall shear stress

REFERENCES

1. H. Deconinck, R. Struijs, K. G. Powell and P. Roe, *AIAA Paper 91-1532*, AIAA 10th Computational Fluid Dynamics Conference, Honolulu, Hawaii, 1991.
2. R. Kettler, 'Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods', *Multigrid Methods*, *Lecture Notes in Mathematics*, **960**, 502–534 (1982).
3. G. Wittum, 'Linear iterations as smoothers in multigrid methods: Theory with applications to incomplete decompositions', *Impact Comput. Sci. Eng.*, **1**, 180–215 (1989).
4. P.M. de Zeeuw, 'Matrix-dependent prolongations and restrictions in a black-box multigrid solver', *J. Comp. Appl. Math.*, **33**, 1–27 (1990).
5. A.W. Mulder, 'A new multigrid approach to convection problems', *J. Comput. Phys.*, **83**, 303–323 (1989).
6. J. Ruge and K. Stüben, 'Efficient solution of finite difference and finite element equation by algebraic multigrid (AMG)', *Arbeitspapiere der GMD*, **89** (1984).
7. R.D. Lonsdale, 'An algebraic multigrid solver for the Navier–Stokes equations on unstructured meshes', *Int. J. Numer. Methods Heat Fluid Flow*, **3**, 3–14, (1993).
8. R. Webster, 'An algebraic multigrid solver for Navier–Stokes problems', *Int. J. Numer. Methods Fluids*, **18**, 761–780 (1994).
9. E. Morano, D.J. Mavriplis and V. Venkakrishnan 'Coarsening strategies for unstructured multigrid techniques with application to anisotropic problems', *ICASE Report No. 95-34*, 1995.
10. D.J. Mavriplis and V. Venkatakrishnan, 'A unified multigrid solver for the Navier–Stokes equations on mixed element meshes', *ICASE Report No 95-53*, 1995.

11. A. Dervieux, 'Steady Euler simulations using unstructured meshes', Von Karman Institute Lecture Series 85-04, in G. Geymonat (ed.), *Partial Differential Equations of Hyperbolic Type and Applications*, World Scientific, Singapore, 1985, pp. 34–105.
12. K. Baba and M. Tabata, 'On a conservative upwind finite element scheme for cinvective diffusion equations', *R.A.I.R.O. Numer. Anal.*, **15**, 3–25 (1981).
13. G. Carré, 'An implicit multigrid method by agglomeration applied to turbulent flows', *Comput. Fluids* (1996) **26**, 299–320 (1997).
14. B. Van Leer, 'Flux vector splitting for the Euler equations', *Lecture Notes in Physics*, Springer, Berlin, **170**, 405–512 (1982).
15. M.-H. Lallemand and A. Dervieux, 'A multigrid finite element method for solving the two-dimensional Euler equations, in S.F. McCormick (ed.), *Multigrid Methods*, *Theory*, *Applications and Supercomputing*, Marcel Dekker Inc., New York and Basel, 1988, pp. 337–363.
16. M.-H. Lallemand, H. Steve and A. Dervieux, 'Unstructured multigridding by volume agglomeration: current status', *Comput. Fluids*, **21**, 397–433 (1992).
17. B. Koobus, M.-H. Lallemand and A. Dervieux, 'Unstructured volume-agglomeration MG: solution of the Poisson equation', *Int. J. Numer. Methods Fluids*, **18**, 27–42 (1994).
18. A. Brandt, S.F. 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, pp. 257–284.
19. W. Hackbusch, 'Multi-grid methods and application', *Springer Series in Computational Mathematics*, Springer, Berlin. Vol. 4, 1985.
20. J. Francescatto, 'Résolution de l'équation de Poisson par une méthode multigrille agglomérée sur maillages étirés', *INRIA Research Report (in French)*, *No 2712*, 1995.
21. B.E. Launder and D.B. Spalding, 'The numerical computation of turbulent flows', *Comput. Methods Appl. Mech. Eng.*, **3**, 269–289 (1974).
22. W.P. Jones and B.E. Launder, 'The prediction of laminarization with a two-equation model of turbulence', *Int. J. Heat Mass Transfer*, **15**, 301–314 (1972).
23. H.C. Chen and V.C. Patel, 'Near-wall turbulence models for complex flows including separation', *AIAA J.*, **26**, 641–648 (1988).
24. J. Francescatto, 'Modèles bas-Reynolds appliqués a une couche limite compressible', *INRIA Research Report (in French)*, *No 2837*, 1996.
25. M. Wolfshtein 'The velocity and temperature distribution in one-dimensional flow with turbulence augmentation and pressure gradient', *Int. J. Heat Mass Transfer*, **12**, 301–318 (1969).
26. B. Larrouturou, 'How to preserve the mass fraction positivity when computing compressible multi-component flows', *J. Comput. Phys.*, **1**, 59–84 (1991).
27. P.J. Finley and H.H. Fernholz, 'A critical compilation of compressible turbulent boundary data', *AGARDograph*, 1977, No. 223.
28. A. Dervieux, J.-P. Dussauge and M. Braza (eds.), 'Computation and comparison of efficient turbulence models for aeronautics', *Proc. ETMA Workshop*, to be published by Vieweg.
29. P.H. Cook, M.A. McDonald and M.C.P. Firmin, 'Pressure distributions, and boundary layer and wake measurements', *AGARD AR 138*, A6-1–A6-77 (1979)
30. S.P. Spekreijse, 'Multigrid solution of the steady Euler equations', *Ph.D. Thesis*, University of Amsterdam, 1987.
31. R. Radespiel and R.C. Swanson 'Progress with multigrid schemes for hypersonic flow problems', *ICASE Report No 91-89*, 1991.

