Numerical Algorithms **33:** 319–330, 2003.

A flux correction multigrid for compressible flow

Aleš Janka

INRIA Sophia Antipolis, 2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis, France
E-mail: Ales.Janka@sophia.inria.fr

Received 4 December 2001; accepted 1 October 2002

A finite-volume based linear multigrid algorithm is proposed and used within an implicit linearized scheme to solve Navier–Stokes equations for compressible laminar flows. Coarse level problems are constructed algebraically based on convective and diffusive fluxes, without the knowledge of coarse geometry. Numerical results for complex 2D geometries such as airfoils, including stretched meshes, show mesh size independent convergence and efficiency of the method compared to other finite-volume-based multigrid method.

Keywords: multigrid method, Navier–Stokes, compressible viscous laminar flow

AMS subject classification: 65M55, 76M12

1. Problem formulation and discretization

Consider we are solving for $u(x,t) : (\Omega \times [0,\infty]) \to \mathbb{R}$ which satisfies

$$\frac{\partial u}{\partial t} - \operatorname{div}(\kappa \operatorname{grad} \tilde{u}) + \operatorname{div} \mathbf{F}(u) = f \quad \text{in } \Omega, \tag{1}$$

with suitable boundary conditions on $\partial\Omega$, and initial condition for $t = 0$, on a bounded domain $\Omega \in \mathbb{R}^2$, where $\tilde{u} = \tilde{u}(u)$, $\mathbf{F}(u) \in \mathbb{R}^2$, $f \in L^2$ and κ is a (2×2) matrix generally depending on u and spatial coordinates. Let us search for a stationary solution $u(x,\infty)$, if this exists.

We assume that the domain Ω is covered by a system of P1 finite elements τ_h, regular and quasi-uniform. In case of dominant convection, it is a well-known fact that the centered approximation of (1) suffers from stability problems. To stabilize, we are using a simple first-order upwind scheme for convective terms, based on finite volume techniques. For each node i of the triangulation we define an associated control volume C_i as a polygonal cell whose vertices are the centers of edges connecting node i to its neighbours and the centers of gravity of the triangles having i as a node.

Integrating (1) over C_i, using Gauss formula and splitting the cell boundary ∂C_i into cell interfaces ∂C_{ij} between two cells C_i and C_j, $\partial C_i = \bigcup_j \partial C_{ij}$, $\partial C_{ij} = \partial C_i \cap \partial C_j$

gives

$$\int_{C_i} \frac{\partial u}{\partial t}\,\mathrm{d}\Omega - \sum_{j\in\mathcal{N}(i)} \int_{\partial C_{ij}} (\kappa\,\mathrm{grad}\,\tilde{u})\cdot\mathbf{n}\,\mathrm{d}\Gamma + \sum_{j\in\mathcal{N}(i)} \int_{\partial C_{ij}} \mathbf{F}(u)\cdot\mathbf{n}\,\mathrm{d}\Gamma = \int_{C_i} f\,\mathrm{d}\Omega, \quad (2)$$

where $\mathbf{n}$ is the exterior unit normal vector to C_i.

At this stage, we introduce flux functions Φ_h^C and Φ_h^D which approximate the convective and diffusive flux integrals, i.e.

$$\int_{\partial C_{ij}} \mathbf{F}(u)\cdot\mathbf{n}\,\mathrm{d}\Gamma \approx \Phi_h^C(u_i, u_j, \mathbf{n}_{ij}), \qquad \int_{\partial C_{ij}} (\kappa\,\mathrm{grad}\,\tilde{u})\cdot\mathbf{n}\,\mathrm{d}\Gamma \approx \Phi_h^D(\tilde{u}_i, \tilde{u}_j),$$

where $\mathbf{n}_{ij} = \int_{\partial C_{ij}} \mathbf{n}\,\mathrm{d}\Gamma$ and $\mathcal{N}(i)$ is the set of nodes neighbouring to node i. Like in [2], we take a simple first-order approximation of convective flux,

$$\Phi_h^C(u_i, u_j, \mathbf{n}_{ij}) = \frac{1}{2}\big[\mathbf{F}(u_i)\cdot\mathbf{n}_{ij} + \mathbf{F}(u_j)\cdot\mathbf{n}_{ij} - \big|\mathbf{J}(u_{ij})\cdot\mathbf{n}_{ij}\big|(u_j - u_i)\big], \quad (3)$$

where $\mathbf{J}(u_{ij})$ is the Roe's generalization of Jacobian of $\mathbf{F}(\cdot)$ at an interface state u_{ij}.

The diffusive term is approximated by a finite volume–finite element technique [3] by taking

$$\Phi_h^D(\tilde{u}_i, \tilde{u}_j) = (\tilde{u}_j - \tilde{u}_i) \sum_{T\in\tau_h: i,j\in T} \int_T \big(\bar{\kappa}(T)\,\mathrm{grad}\,\varphi_j\big)\cdot\mathrm{grad}\,\varphi_i\,\mathrm{d}\Omega,$$

where φ_i, φ_j are the P1 finite element basis functions on nodes i and j and $\bar{\kappa}(T)$ is the average of κ on a triangle T.

In this presentation we use the numerical fluxes from [2]. Nevertheless, the multigrid concept introduced further is based on a universal finite volume reasoning and can be thus applied on a wider range of finite volume schemes.

Implicit time integration and linearization. The choice of numerical flux functions Φ_h^C and Φ_h^D results in a system of nonlinear ODEs in time: find $u_i(t)$ on C_i such that

$$\begin{aligned}\int_{C_i} \frac{\partial u_i}{\partial t} &+ \sum_{T\in\tau_h: i\in T} \sum_{k\in T} \tilde{u}_k \int_T \big(\bar{\kappa}(T)\,\mathrm{grad}\,\varphi_k\big)\cdot\mathrm{grad}\,\varphi_i\,\mathrm{d}\Omega \\ &+ \sum_{j\in\mathcal{N}(i)} \frac{1}{2}\big[\mathbf{F}(u_i)\cdot\mathbf{n}_{ij} + \mathbf{F}(u_j)\cdot\mathbf{n}_{ij} - \big|\mathbf{J}(u_{ij})\cdot\mathbf{n}_{ij}\big|(u_j - u_i)\big] = \int_{C_i} f\,\mathrm{d}\Omega. \quad (4)\end{aligned}$$

Let us choose an implicit one-step strategy for solving this nonlinear system and let us linearize to get an approximated Jacobian of the problem. Having a known guess u^n, let us solve for an update δu^n to get the next guess $u^{n+1} = u^n + \delta u^n$. The temporal term is approximated by a forward difference formula

$$\int_{C_i} \frac{\partial u_i}{\partial t} \approx \mu(C_i)\frac{u_i^{n+1} - u_i^n}{(\delta t^n)_i}.$$

The task of linearization consists in approximating the Jacobians of both convective and diffusive fluxes, $(\partial/\partial u_m)\Phi_h^C(\tilde{u}_i, \tilde{u}_j, \mathbf{n}_{ij})$ and $(\partial/\partial u_m)\Phi_h^D(\tilde{u}_i, \tilde{u}_j)$, on each edge. The edgewise flux function $\Phi_h^D(\tilde{u}_i, \tilde{u}_j)$ depends, however, not only on the values of the respective neighbouring cells, but also their common neighbours (through the trianglewise average $\bar{\kappa}(T)$). To avoid difficulties with differentiating, we take, for the sake of linearization, a slightly different diffusive flux function which uses an edgewise average $\bar{\kappa}_{ij}$ of κ,

$$\bar{\Phi}_h^D(\tilde{u}_i, \tilde{u}_j) = (\tilde{u}_j - \tilde{u}_i) \sum_{T \in \tau_h : i, j \in T} \int_T (\bar{\kappa}_{ij} \operatorname{grad} \varphi_j) \cdot \operatorname{grad} \varphi_i \, \mathrm{d}\Omega. \tag{5}$$

Assume that we have an initial estimate u^n of u, respectively $\tilde{u}^n$ of $\tilde{u}$ and we linearize $\Phi_h^D(\tilde{u}_i, \tilde{u}_j)$ and $\Phi_h^C(\tilde{u}_i, \tilde{u}_j, \mathbf{n}_{ij})$ around this state. By a first-order Taylor expansion we have

$$\begin{aligned}
\Phi_h^D\big(\tilde{u}_i^{n+1}, \tilde{u}_j^{n+1}\big) &\approx \Phi_h^D\big(\tilde{u}_i^n, \tilde{u}_j^n\big) + \frac{\partial \bar{\Phi}_h^D(\tilde{u}_i^n, \tilde{u}_j^n)}{\partial u_i^n}\big(u_i^{n+1} - u_i^n\big) \\
&\quad + \frac{\partial \bar{\Phi}_h^D(\tilde{u}_i^n, \tilde{u}_j^n)}{\partial u_j^n}\big(u_j^{n+1} - u_j^n\big), \\
\Phi_h^C\big(u_i^{n+1}, u_j^{n+1}, \mathbf{n}_{ij}\big) &\approx \Phi_h^C\big(u_i^n, u_j^n, \mathbf{n}_{ij}\big) + \frac{\partial \Phi_h^C(u_i^n, u_j^n, \mathbf{n}_{ij})}{\partial u_i^n}\big(u_i^{n+1} - u_i^n\big) \\
&\quad + \frac{\partial \Phi_h^C(u_i^n, u_j^n, \mathbf{n}_{ij})}{\partial u_j^n}\big(u_j^{n+1} - u_j^n\big).
\end{aligned}$$

The implicit linearized strategy reduces the problem into a sequence of linear algebraic problems: for a given u^n, compute $u^{n+1} = u^n + \delta u^n$ by solving for δu^n,

$$(T_h + D_h + C_h)\delta u^n = b_h, \tag{6}$$

where T_h, D_h and C_h denote the temporal, convective and diffusive components of the problem matrix, generally functions of u^n,

$$(D_h)_{ii} = \sum_{j \in \mathcal{N}(i)} \frac{\partial}{\partial u_i^n}\bigg[\big(\tilde{u}_j^n - \tilde{u}_i^n\big) \sum_{T \in \tau_h : i, j \in T} \int_T (\bar{\kappa}_{ij} \operatorname{grad} \varphi_j) \cdot \operatorname{grad} \varphi_i \, \mathrm{d}\Omega\bigg], \tag{7}$$

$$(D_h)_{ij} = \frac{\partial}{\partial u_j^n}\bigg[\big(\tilde{u}_j^n - \tilde{u}_i^n\big) \sum_{T \in \tau_h : i, j \in T} \int_T (\bar{\kappa}_{ij} \operatorname{grad} \varphi_j) \cdot \operatorname{grad} \varphi_i \, \mathrm{d}\Omega\bigg], \quad i \neq j, \tag{8}$$

$$(C_h)_{ii} = \frac{1}{2} \sum_{j \in \mathcal{N}(i)} \big(\mathbf{J}\big(u_i^n\big) \cdot \mathbf{n}_{ij} + \big|\mathbf{J}\big(u_{ij}^n\big) \cdot \mathbf{n}_{ij}\big|\big),$$

$$(C_h)_{ij} = \frac{1}{2}\big(\mathbf{J}\big(u_j^n\big) \cdot \mathbf{n}_{ij} - \big|\mathbf{J}\big(u_{ij}^n\big) \cdot \mathbf{n}_{ij}\big|\big),$$

terms with $(\partial/\partial u_i)(|\mathbf{J}(u_{ij})\cdot\mathbf{n}_{ij}|)(u_j-u_i)$ were neglected [2]. The right-hand side vector b_h is

$$(b_h)_i = \int_{C_i} f\,\mathrm{d}\Omega - \sum_{T\in\tau_h: i\in T}\sum_{k\in T}\tilde{u}_k^n\int_T\big(\bar{\kappa}(T)\,\mathrm{grad}\,\varphi_k\big)\cdot\mathrm{grad}\,\varphi_i\,\mathrm{d}\Omega$$
$$-\frac{1}{2}\sum_{j\in\mathcal{N}(i)}\big[\mathbf{F}\big(u_i^n\big)\cdot\mathbf{n}_{ij}+\mathbf{F}\big(u_j^n\big)\cdot\mathbf{n}_{ij}-\big|\mathbf{J}\big(u_{ij}^n\big)\cdot\mathbf{n}_{ij}\big|\big(u_j^n-u_i^n\big)\big]. \tag{9}$$

As we are interested only in the "converged" stationary state $u(x,\infty)$, t might not have the conventional role of time which passes globally by the same pace on each cell C_i of the domain Ω, it might advance locally. For the same reason, we are not obliged to solve a nonlinear problem on each timestep, it can be replaced by a linearized approximation.

The temporal matrix is diagonal, $(T_h)_{ii}=\mu(C_i)/(\delta t^n)_i$. The local timestep $(\delta t^n)_i$ on a control cell C_i is calculated by

$$\big(\delta t^n\big)_i = C_T\big(\delta t^n_{\mathrm{stab}}\big)_i,\quad \big(\delta t^n_{\mathrm{stab}}\big)_i \leqslant \frac{h_i^2}{\|\mathbf{J}(u_i^n)\mathbf{n}\|h_i+2\kappa}, \tag{10}$$

where $(\delta t^n_{\mathrm{stab}})_i$ is the maximal timestep allowed by the stability condition of a simple explicit scheme for 1D advection–diffusion, h_i is the characteristic mesh-size of the cell C_i and C_T is a given parameter (in [2] called the "CFL number") usually ranging from 10^3 to 10^{10} which can vary from timestep to timestep.

2. Multigrid solver for the linearized problem

Recent comparisons [5] of multigrid performances show that linearized multigrid schemes can compete, when used as preconditioners of a Newton–Krylov method, with a nonlinear FAS multigrid solving the nonlinear stationary problem. At present, we concentrate on a linear multigrid as a solver for the linearized system (6). This multigrid might be then used as a preconditioner.

The main idea of any multigrid approach is to decompose the solution of the linear (finest) problem into a treatment of related problems of geometrically decreasing size, called coarse problems and to combine their solutions with the finest one to speed-up convergence on the finest level. The task of coarse-grid corrections is to propagate information rapidly, in terms of number of iterations of a simple iterative method, across the computational domain, while the role of pre- and post-smoothing on finer levels is to cope with local imprecisions of the coarse-grid correction.

Let us set $A_J = T_h + D_h + C_h$ and $b_J = b_h$ on each timestep n and apply a multigrid V-cycle on J multigrid levels (level J is the finest one) to solve for x in

$$A_J x = b_J.$$

Algorithm 1. Set $x_k = 0$ and perform one multigrid V-cycle on each level k:

1. Presmoothing: $x_k \leftarrow (I - R_k A_k)x_k + R_k b_k$, where R_k is a preconditioner of some simple iterative method (e.g., Gauss–Seidel or Jacobi iterations).

1. Coarse-grid correction: if $k > 1$,

   (a) Restrict the residual $r_k = b_k - A_k x_k$ to get a coarse right-hand side $b_{k-1} = I_{k-1}^{k} r_k$.

   (b) Solve for x_{k-1} of a coarse-level problem $A_{k-1}x_{k-1} = b_{k-1}$ by a recursive application of this algorithm on the coarse level $(k - 1)$.

   (c) Prolongation: correct x_k by $x_k \leftarrow x_k + I_k^{k-1} x_{k-1}$.

3. Post-smoothing: $x_k \leftarrow (I - R_k A_k)x_k + R_k b_k$.

The multigrid algorithm 1 is characterized by the interlevel transfer operators, restriction $I_{k-1}^{k} : M_k \to M_{k-1}$ and prolongation $I_k^{k-1} : M_{k-1} \to M_k$, and a series of coarse-level operators $A_k : M_k \to M_k$, where M_J is the finest level and M_1 is the coarsest level spaces.

In this work, we rest within the class of algorithms with nested-coarse spaces, and introduce a multigrid which heavily uses the idea of agglomeration of finite volume control cells (i.e. aggregation of discrete unknowns) of the finest level problem A_J to design the coarse level problems A_k. Thus, for our purposes, the spaces M_k are piecewise constant on the control cells C_i^k, $C_i^J = C_i$. Let the coarse level cells $\{C_i^{k-1}\}$ be obtained by agglomeration of fine level cells $\{C_i^k\}$ so that the discrete diameter of C_i^{k-1}, i.e. $\max_{C_j^k \in C_i^{k-1}}(\mathrm{diam}(C_i^{k-1})/\mathrm{diam}(C_j^k))$, is roughly 2. The discrete diameter is also referred to as the coarsening ratio. Dually to the set of agglomerated cells $\{C_i^k\}$ we can construct a set of zero–one prolongations (by natural injection from M_{k-1} to M_k):

$$\left(I_k^{k-1}\right)_{ip} = \begin{cases} 1 & \text{if } C_i^k \in C_p^{k-1}, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Let us choose the restriction I_{k-1}^{k} to be the transpose of I_k^{k-1}. We would like to formulate our coarse level problems variationally in a Galerkin manner by

$$A_{k-1} = \left(I_k^{k-1}\right)^{\mathrm{T}} A_k I_k^{k-1} \quad \text{and} \quad b_{k-1} = \left(I_k^{k-1}\right)^{\mathrm{T}} b_k, \tag{12}$$

which corresponds to sums, in terms of matrix and vector elements,

$$(A_{k-1})_{pq} = \sum_{C_i^k \in C_p^{k-1}} \sum_{C_j^k \in C_q^{k-1}} (A_k)_{ij} \quad \text{and} \quad (b_{k-1})_p = \sum_{C_i^k \in C_p^{k-1}} (b_k)_i. \tag{13}$$

Formulating A_{k-1} in this way has some practical advantages: it gives a natural treatment for the convective part of the problem: the formulation of C_{k-1} by the variational way, $C_{k-1} = (I_k^{k-1})^{\mathrm{T}} C_k I_k^{k-1}$, with the right-hand side contributions as in (13) is identical to rediscretization by the technique of section 1 with the convective flux Φ_h^C formally replaced by a coarse-level convective flux function Φ_{k-1}^C,

$$\begin{aligned}\Phi_{k-1}^C(u_p, u_q, \mathbf{n}_{pq}) &= \sum_{\partial C_{ij}^J \subset \partial C_{pq}^{k-1}} \Phi_J^C\big(\big(I_J^{k-1}u\big)_i, \big(I_J^{k-1}u\big)_j, \mathbf{n}_{ij}\big)\\ &= \frac{1}{2}\Bigg[\mathbf{F}(u_p)\cdot\mathbf{n}_{pq} + \mathbf{F}(u_q)\cdot\mathbf{n}_{pq} - \sum_{\partial C_{ij}^J \subset \partial C_{pq}^{k-1}} \big|\mathbf{J}(u_{pq})\cdot\mathbf{n}_{ij}\big|(u_q - u_p)\Bigg].\end{aligned}$$

Here, Φ_J^C is the convective flux Φ_h^C of (3) and the composite prolongation $I_J^{k-1}: M_{k-1} \to M_J$ is the natural injection, $\mathbf{n}_{pq} = \sum_{ij} \mathbf{n}_{ij}$, $\partial C_{ij}^J \subset \partial C_{pq}^{k-1}$. Although the flux function Φ_{k-1}^C introduces more artificial diffusion than a standard rediscretization with an analog of Φ_h^C on coarse level $(k-1)$,

$$\sum_{\partial C_{ij}^J \subset \partial C_{pq}^{k-1}} \big|\mathbf{J}(u_{pq})\cdot\mathbf{n}_{ij}\big|(u_q - u_p) \quad \text{vs.} \quad \Big|\mathbf{J}(u_{pq})\cdot\sum_{ij}\mathbf{n}_{ij}\Big|(u_q - u_p),$$

it gives a reasonable way how to discretize the convective terms on coarse levels.

This is not the case for the diffusive term. Still, the formulation of $\widetilde{D}_{k-1} = (I_k^{k-1})^{\mathrm{T}} D_k I_k^{k-1}$ is equivalent to rediscretization by the technique of section 1, with the diffusive flux Φ_h^D formally replaced by $\widetilde{\Phi}_{k-1}^D$,

$$\widetilde{\Phi}_{k-1}^D(u_p, u_q) = \sum_{\partial C_{ij}^J \subset \partial C_{pq}^{k-1}} \Phi_h^D\big(\big(I_J^{k-1}u\big)_i, \big(I_J^{k-1}u\big)_j\big). \tag{14}$$

The flux $\widetilde{\Phi}_{k-1}^D$, however, is no longer consistent in the finite volume sense [1], despite the fact that the consistency of the fine level flux Φ_h^D follows from the soundness of the discretization on the finest level. For Φ_h^D there should exist a constant-preserving operator $Q_h : H^2(\Omega) \to M_h$ such that the sum of approximation errors of diffusive flux across the interface ∂C_{ij},

$$R_{ij}(u) = \frac{1}{\mu(\partial C_{ij})}\bigg(\int_{\partial C_{ij}} \kappa \operatorname{grad} u \cdot \mathbf{n}_{ij}\,\mathrm{d}\Gamma - \Phi_h^D\big((Q_h u)_i, (Q_h u)_j\big)\bigg),$$

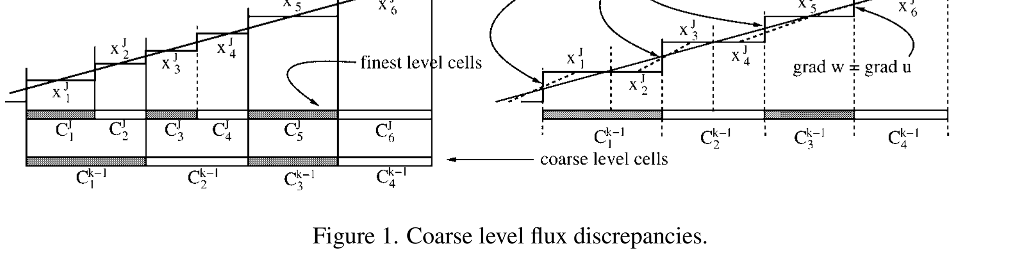

Figure 1. Coarse level flux discrepancies.

is bounded [1] by

$$\sum_{ij} R_{ij}(u)^2 \mu(\partial C_{ij}) h_{ij} \leqslant ch^2 |u|^2_{2,\Omega}, \quad \text{i.e. } R(u) = 0 \text{ for a linear } u, \tag{15}$$

where $\mu(\partial C_{ij})$ is the area of the interface ∂C_{ij}, h_{ij} and h are the local and global mesh-sizes, and $|\cdot|_{2,\Omega}$ is the H^2 semi-norm. In particular, equation (15) imposes that for all (locally) linear functions u, $R_{ij}(u)$ must be zero, i.e. the numerical flux Φ^D_h must be actually exact. Also, for such Q_h there exist unique (flux-evaluation) points $\{x^J_i\}$ such that $(Q_h u)_i = u(x^J_i)$ for every u linear on C^J_i.

The property (15), however, does not hold for $\widetilde{\Phi}^D_{k-1}$. As illustrated on the right of figure 1 for a 1D example, the fact that I^{k-1}_J is not a linear interpolation causes that for a linear function u the coarse fluxes $\widetilde{\Phi}^D_{k-1}$ (here proportional to grad w, w piecewise linear, $w(x^J_i) = (I^{k-1}_J Q_{k-1} u)_i$) do not in general coincide on ∂C^{k-1}_{ij} with the fluxes of the finest level (grad u). They might however be the same for some interfaces (cf. $\partial C^{k-1}_{3,4}$ in figure 1) of cells which have not been agglomerated. Moreover, in 2D things might get complicated in zones where coarsening just in one direction is applied to compensate for high aspect ratio of the finest level meshes.

To assure consistency on coarse levels we propose to correct the diffusive flux in (14). Expressing the discrepancies in terms of fluxes accross every cell interface, as hinted in (15), enables local treatment of this problem, not only on structured grids, but also on generally unstructured and stretched ones. Instead of flux in (14) we take

$$\Phi^D_{k-1}(u_p, u_q) = c_{pq} \widetilde{\Phi}^D_{k-1}(u_p, u_q), \tag{16}$$

where the corrective factor $c_{pq} = c_{qp}$ is designed so that at least for one class of linear functions u, $\nabla u \parallel \mathbf{n}_{pq}$, the key property $R(u) = 0$ of (15) holds. We set

$$c_{pq} = \frac{\sum_{\partial C^J_{ij} \subset \partial C^{k-1}_{pq}} \Phi^D_h((Q_h u)_i, (Q_h u)_j)}{\sum_{\partial C^J_{ij} \subset \partial C^{k-1}_{pq}} \Phi^D_h((I^{k-1}_J Q_{k-1} u)_i, (I^{k-1}_J Q_{k-1} u)_j)},$$

where $Q_{k-1} : H^2 \to M_{k-1}$ is the L^2 projection. In practice, the flux-correction by (16) amounts to correcting

$$(D_{k-1})_{pq} = c_{pq} \cdot \bigl(\widetilde{D}_{k-1}\bigr)_{pq}, \qquad (D_{k-1})_{pp} = -\sum_{q \neq p} (D_{k-q})_{pq}.$$

The scheme can still be expressed in terms of (corrected) fluxes and as such it is conservative, if the finest level scheme is conservative.

3. Numerical experiments for compressible viscous fluids

Let us have a system of partial differential equations modelling laminar viscous compressible flow. The most common mathematical model of the physics is condensed to Navier–Stokes equations. Let us search for a function $W = (\rho, \rho v_x, \rho v_y, E)^{\mathrm{T}}$ satisfying

$$\int_{C_i} \frac{\partial W}{\partial t} + \int_{C_i} \left(\frac{\partial F(W)}{\partial x} + \frac{\partial G(W)}{\partial y} \right) = \frac{1}{\mathrm{Re}} \int_{C_i} \left(\frac{\partial R(W)}{\partial x} + \frac{\partial S(W)}{\partial y} \right) \quad \text{in } \Omega, \tag{17}$$

with

$$\begin{aligned}
F(W) &= \left(\rho v_x, \rho v_x^2 + p, \rho v_x v_y, (E+p) v_x\right)^{\mathrm{T}}, \\
G(W) &= \left(\rho v_y, \rho v_x v_y, \rho v_y^2 + p, (E+p) v_y\right)^{\mathrm{T}}, \\
R(W) &= \left(0, \tau_{xx}, \tau_{xy}, v_x \tau_{xx} + v_y \tau_{xy} + \frac{\gamma \mu}{\mathrm{Pr}} \frac{\partial e}{\partial x}\right)^{\mathrm{T}}, \quad \text{and} \\
S(W) &= \left(0, \tau_{xy}, \tau_{yy}, v_x \tau_{xy} + v_y \tau_{yy} + \frac{\gamma \mu}{\mathrm{Pr}} \frac{\partial e}{\partial y}\right)^{\mathrm{T}}.
\end{aligned}$$

Here μ is the viscosity parameter, Re and Pr are the Reynolds and Prantl's numbers, τ_{ij} the stress tensor, γ is the gas constant, and C_{V} is the specific heat at constant volume. The physical unknowns are denoted by ρ density, $(v_x, v_y)^{\mathrm{T}}$ velocity, and e specific internal energy. We are using adimensioned system in conservative form with variables: ρ, momentum $(\rho v_x, \rho v_y)^{\mathrm{T}}$ and E total energy per unit volume. Suppose the fluid is a perfect gas, then

$$E = \rho e + \frac{1}{2}\rho\left(v_x^2 + v_y^2\right) = \rho C_{\mathrm{V}} T + \frac{1}{2}\rho\left(v_x^2 + v_y^2\right), \qquad p = (\gamma - 1)\rho C_V T,$$

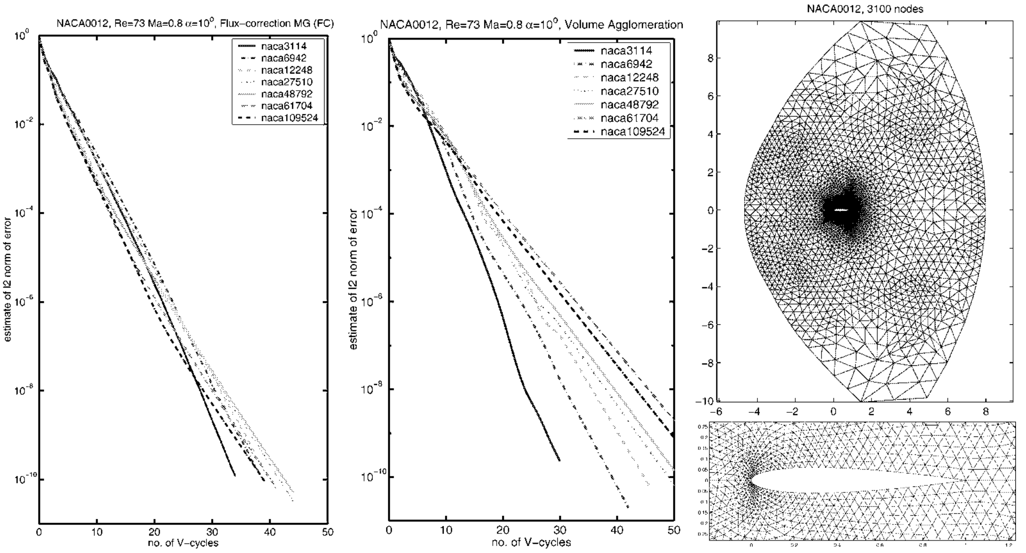

Figure 2. NACA viscous transsonic: linear convergence (left), mesh (right).

$$\tau_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial v_k}{\partial x_k}\delta_{ij}.$$

In this study, we restrict ourselves to external flows around an airfoil. The condition on the far-field mesh boundary $\Gamma_\infty \subset \partial\Gamma$ is given by a free-stream state

$$\rho_\infty = 1, \qquad \mathbf{v}_\infty = (v_{x,\infty}, v_{y,\infty})^{\mathrm{T}}, \quad \text{with } \|\mathbf{v}_\infty\| = 1, \qquad p_\infty = \left(\gamma M_\infty^2\right)^{-1},$$

where M_∞ is the Mach number of a uniform flow. As to the conditions on the obstacle, we pose an isothermic no-slip condition, i.e. we set $\mathbf{v}_0 = \mathbf{0}$, and $T_0 = T_\infty(1 + ((\gamma - 1)/2)M_\infty^2)$, which constrains directly to zero the ρv_x and ρv_y components of W and the temperature condition appears in the E energy component through the law of state, $E = \rho C_{\mathrm{V}} T_0$, $p = (\gamma - 1)E$, on Γ_0.

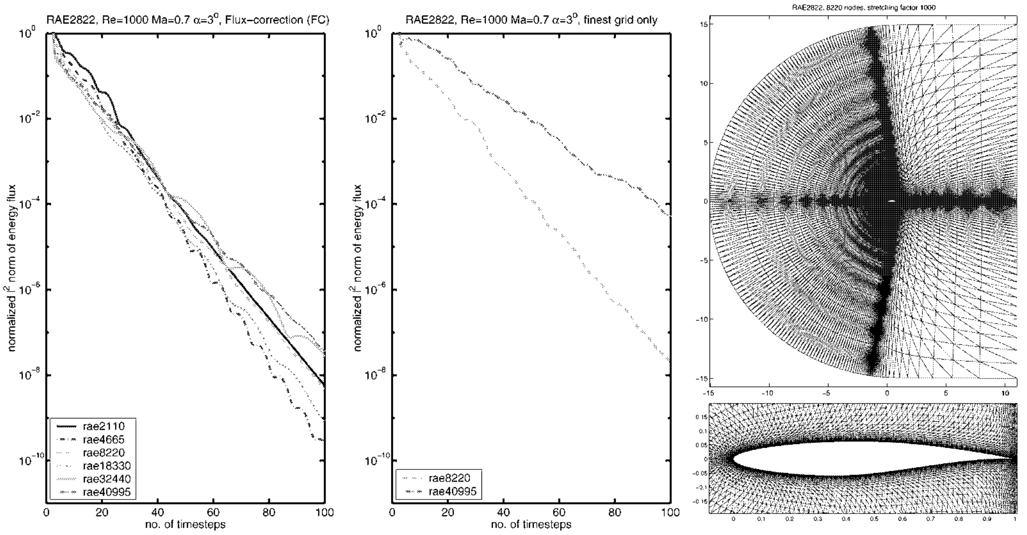

Figure 3. RAE transsonic: nonlinear convergence (left) and mesh (right).

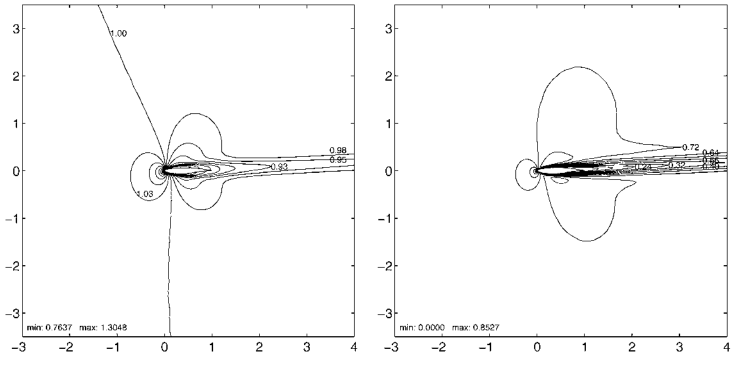

Figure 4. RAE transsonic: converged density and Mach number.

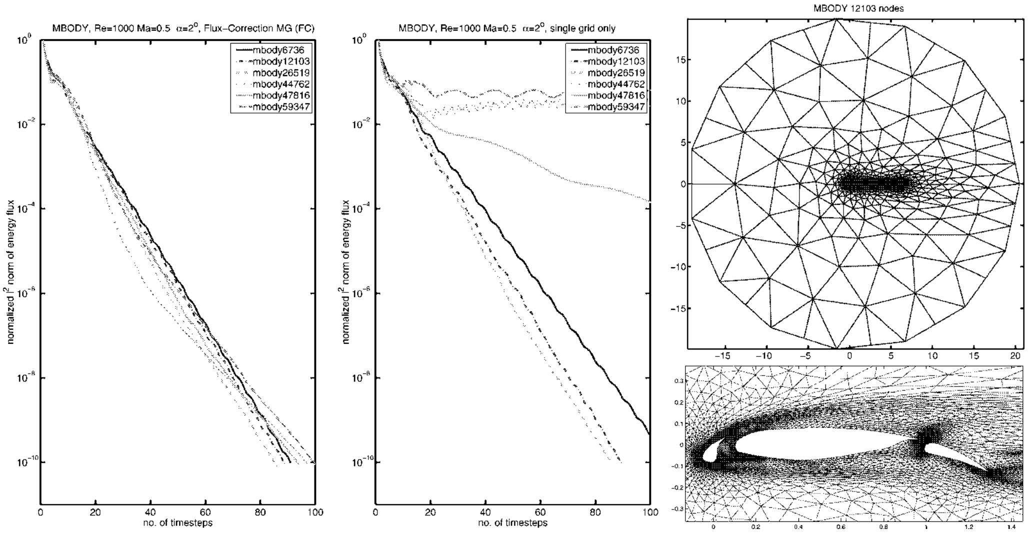

Figure 5. MBODY subsonic: nonlinear convergence (left) and mesh (right).

The Navier–Stokes equations can be rewriten in the form (1), with $\tilde{u}(u)$: $(\rho, \rho v_x, \rho v_y, E)^{\mathrm{T}} \to (\rho, v_x, v_y, e)^{\mathrm{T}}$. For explicitly written formulas for Jacobian matrices, flux contributions, and calculation of the local timestep $(\delta t^n_{\mathrm{stab}})_i$, please refer to [2].

Transsonic flow at high incidence and small Reynolds number. We perform a test on the NACA0012 geometry with $M_\infty = 0.8$, angle of attack $\alpha = 10°$, C_T of (10) $C_T = 1000$, $\mathrm{Re} = 73$. First, we apply multigrid V-cycles with 3 and 3 Jacobi iterations on the first time-step only and see if the convergence rate depends on refinements of the mesh. Convergence histories for meshes of different mesh-sizes are plotted in figure 2. We clearly see, that while the proposed algorithm is roughly mesh-size independent, the performance of the Volume-Agglomeration multigrid [4] deteriorates with the mesh-size.

Transsonic flow at incidence and moderate Reynolds number. Let us now try the multigrid Navier–Stokes solver on a stretched geometry of RAE2822 in the transsonic regime with $\mathrm{Re} = 1000$, $M_\infty = 0.8$ and $\alpha = 3°$. We use 3 5-level multigrid V-cycles per timestep, with 3 and 3 Gauss–Seidel iterations which reduces the l^2 linear residual norm approximately by a factor of $5 \cdot 10^{-2}$. Set C_T from (10) to 1000 and iterate in time to obtain a stationary solution (cf. figure 4). Convergence histories for the algorithms are presented in figure 3. The last plot in figure 3 belongs to a case when 16 × 3 sweeps of a simple single-grid Gauss–Seidel method are used to solve the linearized problem, which is superior in complexity to the 3 multigrid V-cycles used in the test. We observe, that while the nonlinear convergence of the single-grid method deteriorates with refinement, it remains approximately the same for all mesh-sizes when multigrid is used.

Subsonic flow of multi body airfoil at moderate Reynolds number. Let us test the multigrid Navier–Stokes solver on a multi-body airfoil MBODY. Set $\mathrm{Re} = 1000$, $M_\infty = 0.5$

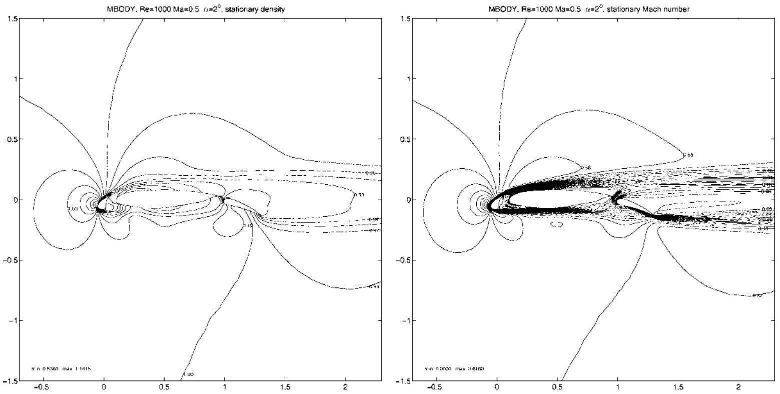

Figure 6. MBODY subsonic: converged density and Mach number.

and the angle of attack $\alpha = 2°$. The mesh is displayed in figure 5. The difficulty of the computation lies not only in the anisotropies of the mesh (the maximal aspect ratio attains 100), but also in the fact, that the flow contains regions with recirculation. Use a timestep variable coefficient C_T of (10), $C_T = \min(10^3 + (1.3)^n, 10^{10})$ and repeat the nonlinear convergence experiment. Figure 6 shows the distribution of density and Mach number of the converged stationary solution. Figure 5 displays the nonlinear convergence histories for both the Flux-Correction multigrid and a single grid Gauss–Seidel method. On each timestep we have used one or two V-cycles of a multigrid method which has reduced the linear residual at least by a factor 10^{-1}. As pre and post-smoothers we have used 6 Gauss–Seidel iterations. For the single grid calculation, we have used 240 iterations of Gauss–Seidel per timestep, which is largely superior in complexity to the employed V-cycles.

4. Conclusion

A similar technique has been used already in [4], introducing the Volume Agglomeration multigrid method, with the correction $c_{pq} = c_{\text{glob}}$ chosen constant for the whole domain. However, as we have seen here, the corrective factor seems to depend on local geometry of each agglomerate. This is why the technique presented in [4] was difficult to apply to stretched meshes.

The proposed scheme has more virtues than just being conservative: in cases when the system of coarse level agglomerated cells forms an admissible (possibly unstructured) finite volume mesh in the sense of [1], the corrected scheme becomes a finite volume scheme. In those cases we obtain consistent discretizations on coarse levels with nested solution spaces, which is quite difficult to achieve in a finite element framework.

From the point of view of algorithmic complexity, even when directional coarsening is used, sparse matrices representing discrete operators on coarse levels do not show an excessive fill-in, as it might be the case with multigrid methods by smoothed prolongation [6].

From the theoretical point of view it rests to provide a rigorous convergence proof in cases when coarse level schemes are not completely consistent, i.e. when the requirement (15) holds just for one class of linear functions u, $\operatorname{grad} u \parallel \mathbf{n}_{ij}$. We believe that in this case the resulting errors of the coarse level schemes are compensated by pre- and post-smoothing.

References

[1] R. Eymard, T. Gallouët, and R. Herbin, *Finite Volume Methods*, Handbook of Numerical Analysis, Vol. VII (North-Holland, Amsterdam, 2000) pp. 713–1020.

[2] L. Fezoui, S. Lanteri, B. Larouturrou and C. Olivier, Résolution numérique des équations de Navier–Stokes pour un fluide compressible en maillage triangulaire, INRIA Research Report No. 1033 (1989).

[3] H. Guillard, *Mixed Element Volume Approach in Computational Fluid Dynamics*, von Karman Institute Lecture Series (1995).

[4] B. Koobus, M.-H. Lallemand and A. Dervieux, Unstructured volume agglomeration MG: Solution of the Poisson equation, Internat. J. Numer. Methods Fluids 18 (1994) 27–42.

[5] D. Mavriplis, An assesment of linear versus nonlinear multigrid methods for unstructured mesh solvers, ICASE Report No. 2001-12 (May 2001), and in: *Proc. of the 10th Copper Mountain Conf. on Multigrid Methods* (2001).

[6] P. Vaněk, A. Janka and H. Guillard, Convergence of the algebraic multigrid based on smoothed aggregation II: Extension to a Petrov–Galerkin method, INRIA Research Report No. 3683.