INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:555–579 (DOI: 10.1002/fld.718)

The robustness issue on multigrid schemes applied to the Navier–Stokes equations for laminar and turbulent, incompressible and compressible flows

M. Vázquez[1,2,*,†], M. Ravachol[2], F. Chalot[2] and M. Mallet[2]

[1] *INRIA Sophia Antipolis, Projet Tropics-2004, Route des Lucioles, BP 93 06902 Sophia Antipolis Cedex, France*
[2] *Dassault Aviation, Pole Scientifique-78, Quai Marcel Dassault, BP 300 92152 St. Cloud Cedex, France*

SUMMARY

The paper's leitmotiv is condensed in one word: robustness. This is a real hindrance for the successful implementation of any multigrid scheme for solving the Navier–Stokes set of equations. In this paper, many hints are given to improve this issue. Instead of looking for the best possible speed-up rate for a particular set of problems, at a given regime and in a given condition, the authors propose some ideas pursuing reasonable speed-up rates in any situation. In a previous paper, the authors presented a multigrid method for solving the incompressible turbulent RANS equations, with particular care in the robustness and flexibility of the solution scheme. Here, these concepts are further developed and extended to compressible laminar and turbulent flows. This goal is achieved by introducing a non-linear multigrid scheme for compressible laminar (NS equations) and turbulent flow (RANS equations), taking benefit of a convenient master–slave implementation strategy. Copyright © 2004 John Wiley & Sons, Ltd.

KEY WORDS: multigrid; incompressible flows; compressible flows; RANS equations; turbulence; finite element method

1. INTRODUCTION

Although major issues of any non-linear multigrid implementation, the robustness, flexibility and reliability are rarely considered as a problem which deserves more than two lines. In general, multigrid papers start with an introduction of the kind of problems which are to be solved, followed by the description of the multigrid ideas implemented. Finally, some examples are shown to back the scheme introduced. Most of the time, the speed-up rates are indeed

*Correspondence to: M. Vázquez, INRIA Sophia Antipolis, Projet Tropics-2004, Route des Lucioles, BP 93 06902 Sophia Antipolis Cedex, France.
†E-mail: mariano.vazquez@sophia.inria.fr

Contract/grant sponsor: European Commission

Published online 8 April 2004

Received 14 August 2002
Revised 7 October 2003

impressive. The problems arise when, unaware of the difficulties inherent to each problem, a second group of researchers want to implement a similar approach but to a different set of physical problems, in a different context, using different solvers, discretizations, or numerical schemes, expecting the same efficiency. The multigrid basics are clear, its implementation is straight (although time demanding)... but many times not even a modest speed-up is attained, let alone of a plainly diverging, totally useless scheme. Why? We believe that there are many things that being done in a natural way in one case, are not at all evident in another case, possibly leading to a general failure. After experiencing the same situation, we focused on this point. We have studied a wide range of problems where multigrid is individually reported as successful by many authors. The Navier–Stokes set of equations provides the richness of the problems: compressible and incompressible, viscous and inviscid, turbulent and laminar flows are here under study. In all cases, the points in common are

- The geometric non-linear multigrid scheme, based in a FM (full multigrid)–FAS (full approximation scheme, introduced in Reference [1]).
- The numerical method: a finite element method in space and a fully implicit finite differences scheme in time.
- For the turbulent cases, the physical model is based on the Reynolds Averaged Navier–Stokes, RANS, equations. The problem is closed by means of the k–ε two-equation model (following Reference [2]). No special compressibility corrections are used in the compressible case.
- The general implementation strategy: a master–slave multigrid scheme. A *master* multigrid code, which controls the process' work flow is connected to some *slave* Navier–Stokes solvers. The master identifies the connectivity between the discretizations, constructs the interpolation matrixes and performs the interpolations themselves according to a given strategy. This strategy was proposed in Reference [3].
- The space discretization: unstructured meshes which form a typically non-nested multigrid hierarchy constructed independently (i.e. neither coarsening nor adaptive strategy are used).

Following these premises, both incompressible and compressible turbulent flow solvers are connected to the multigrid master according to two multigrid strategies, introduced in Reference [4] and here, respectively. In this way, we have studied a bunch of ideas for improving the robustness of the schemes. These ideas are independent of the numerical method chosen for discretize the flow equations and can be applied regardless of the individual flow solvers algorithm. Boundary conditions, operators, relaxation factors, source freezing are among the points considered. Also cycling strategies are analysed, particularly cascadic initial stages. These ideas can be combined with changes in the individual flow solvers numerical parameters depending on the order in the multigrid hierarchy. In this way, the whole process can be considered as a block.

The paper is ordered as follows. Firstly, the physical problems under study are briefly described, followed by a section on the numerical method chosen. Then, some basic multigrid concepts precede the description of the multigrid schemes for both of the great problem categories. Next, the robustness issue is faced, describing the different implementation ideas. The performance of what is proposed is tested through some numerical examples. A discussion and conclusion section closes the paper.

2. THE PHYSICAL PROBLEM

The Navier–Stokes set of equations is the object of this paper. Based on conservation principles and the continuum hypothesis, it is a set of transport differential equations that describes flow dynamics under very different ranges and conditions. It comprises two scalar equations for mass and energy conservation and a (spatial) vector one for the linear momentum.

Two great divisions can be established. On the one hand, laminar and turbulent flow. It is generally accepted (indeed some exceptions stand, but let us flow in the mainstream) that turbulent phenomena is covered by the Navier–Stokes equations, with all their founding hypotheses. Many controversial voices arise when turbulence is to be defined (*viz*. Reference [5] or Reference [6]), but basically it can be said that as the inertial forces grow stronger compared with viscous ones (i.e. when the Reynolds number increases), flow dynamics becomes more and more complex, involving a larger scale range. This happens gradually at first, in the laminar regime. All the significant scales can then be resolved by suitable numerical simulations of the Navier–Stokes set of equations, a set that we will call LaNS, for 'Laminar N–S'. But all of a sudden, for a so-called critical Reynolds number which depends on an undetermined number of flow conditions, the turbulent process is unleashed. In the turbulent regime (except for a few and very simplified cases where DNS, direct numerical simulation, is applied) something has to be done with the large amount of small scales that cannot be resolved. One brilliant idea (especially used in engineering problems) is to solve the so called *Reynolds averaged Navier–Stokes* (RANS) equations, where the unknown are the *mean flow* variables. Both the RANS and LaNS equations have basically the same form, except for an additional diffusion term, which accounts for the turbulence. In order to close the problem, the RANS new term is to be modelled. In this paper we follow the *two-equation* k–ε model of Reference [2], as cited in, for example, Reference [7]. In this case, to the RANS set, two more equations are added for the *turbulent kinetic energy* k and the *turbulent dissipation* ε. They are basically transport differential equations, like the RANS or LaNS, except for (very likely) strong non-linear source terms. Although the differences, the numerical method (including all the solving process technical aspects) used to solve the LaNS equations can be extended to the TuNS (we call here TuNS the equations set RANS $+ (k, \varepsilon)$) with relative ease.

The other great division is compressible and incompressible flow. The incompressibility constraint

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

(throughout the paper, Einstein's index summation convention is used unless the contrary is explicitly said) decouples the energy transport equation from the other two, which in turn can be solved independently. Therefore, while the incompressible flow dynamics is modelled exclusively by the continuity and linear momentum transport equations (*viz.* Reference [8]), compressible flows need also the energy transport one. The second main difference between compressible and incompressible flows is that only the former can develop *shock waves*, due to the convective term's non-linearities. Shock waves become a serious additional difficulty in all fronts: the physical analysis of the problem, the mathematical analysis of the differential equation, the numerical analysis of the discretized set and its solution process. For that reason, in contrast with the division laminar/turbulent, the solving numerical algorithm can greatly differ whether one considers compressible or incompressible flows. Typically (unless

when using the so-called *general* (or '*for all seasons*') algorithms) two separate codes are programmed, one for each kind of flow.

So to speak, both of the divisions are not parallel but orthogonal: laminar/turbulent flows can be in/compressible ones. As said above, the object of this paper are all of them.

2.1. The Navier–Stokes set

The continuum fluid mechanics governing set of equations comprises mass, linear momentum and energy transport ones. In its conservation form, it can be written as

$$\frac{\partial U_j}{\partial t} + \frac{\partial F_{ij}^{\mathrm{Adv}}}{\partial x_i} = \frac{\partial F_{ij}^{\mathrm{Diff}}}{\partial x_i} \tag{2}$$

where U_j is the conservative variables vector formed by density ρ, linear momentum ρu_i and total energy per unit volume $\rho e = \rho(C_v T + u_i u_i/2)$. Their corresponding advective and diffusive fluxes are F_{ij}^{Adv} and F_{ij}^{Diff}. As said before, the incompressibility constraint changes radically the form of the equations, the solving process and the kind of the solutions found. In this paper, we are focused in two different fully implicit schemes corresponding to each of these regimes.

Some points are shared by the two schemes. In both of them, the space is discretized by the finite element method. Stationary solutions are achieved by a finite differences scheme applied to the time derivatives which appear in (2), using local time steps to speed up the convergence. The non-linear advective fluxes are linearized and at each time step the resulting system is solved using a GMRES preconditioned method.

Incompressible flows. As said above, in this regime the density remains constant, decoupling pressure variations from thermal ones. We focus here in incompressible mechanical problems, with no heat transport equations. Then, the problem is modelled by the momentum equation plus the incompressibility constraint. The scheme we followed is widely described in References [9, 4]. The flow equations are solved with a monolithic scheme in pressure and velocity, with a traditional finite element method with SUPG stabilization, as introduced in Reference [10], with equal interpolation spaces for velocity and pressure.

Compressible flows. Now a certain state law (here the ideal gas state law) couples the pressure and temperature evolution. Following Reference [11], the compressible flow set of equations is solved on the so-called *entropy variables*. The full Navier–Stokes set, as written in (2) is firstly linearized, then the unknowns are changed to the entropy variables and finally, the resulting set is discretized by means of a GLS formulation (*viz*. References [12, 13]). When needed, the entropy variables can be easily transformed back to the physical ones.

2.2. Reynolds Averaged Navier–Stokes (RANS) equations

In order to model the turbulence effects, we have adopted the RANS approach (*viz*. References [7, 14]). In it, all the variables are written as a mean value plus a small oscillation. Set (2) is then projected using an *ad hoc mean*, that can be temporal or over the ensemble (Reynolds' or Favre's), typical of RANS, or spatial, using a certain space filter, typical of LES. The unkowns are then the *mean flow* variables, which are in turn coupled to the turbulence effects through additional equations (and hypotheses) and terms. The mean flow evolution is then modelled by the RANS equations: they have exactly the same form of Equations (2), except for the fact that now the unknowns are the mean ones and for additional diffusion terms

in momentum and heat transport equations, accounting for the turbulence effects. In both the compressible and incompressible regimes and for the more or less traditional turbulence modelling used here, these terms depends on the *Reynolds stress tensor*, which follows the Boussinesq approximation

$$R_{ij} = 2\mu_{\mathrm{T}} \left(s_{ij} - \frac{1}{3}\ \theta\ \delta_{ij} \right) - \frac{2}{3}\bar{\rho} k \delta_{ij} \tag{3}$$

where k is the *turbulent kinetic energy*, $s_{ij} = \frac{1}{2}(\partial \bar{u}_i/\partial x_j + \partial \bar{u}_j/\partial x_i)$, $\theta = \partial \bar{u}_k/\partial x_k$, the overlines label mean quantities. In order to close the mean flow equations, we have chosen the k–ε model. The k–ε model belongs to the kind known as *two-equation models*. It was introduced in early works like References [15, 2]. For a deeper description of the particular models used here see Reference [4] for the incompressible case and Reference [16] for the compressible one. In all of these models, the *turbulent viscosity* is defined as

$$\mu_{\mathrm{T}} = C_\mu f_\mu \rho \frac{k^2}{\varepsilon} \tag{4}$$

The quantities k and ε are the turbulent kinetic energy and *turbulent dissipation*, respectively. f_μ is either 1.0 or a damping function which depends on the model used. Other parameters that depends on the turbulence models are the following constants C_μ, σ_k, σ_ε, $C_{\varepsilon 1}$ and $C_{\varepsilon 2}$, some of them appearing in the equations below.

Two more transport equations describe their dynamics and, added to the RANS set, close the model:

$$\frac{\partial \bar{\rho} k}{\partial t} + \frac{\partial}{\partial x_i} \left(\bar{\rho} \bar{u}_i k - \left(\mu + \frac{\mu_{\mathrm{T}}}{\sigma_k} \right) \frac{\partial k}{\partial x_i} \right) = \omega_k(k, \varepsilon) \tag{5}$$

$$\frac{\partial \bar{\rho} \varepsilon}{\partial t} + \frac{\partial}{\partial x_i} \left(\bar{\rho} \bar{u}_i \varepsilon - \left(\mu + \frac{\mu_{\mathrm{T}}}{\sigma_\varepsilon} \right) \frac{\partial \varepsilon}{\partial x_i} \right) = \omega_\varepsilon(k, \varepsilon) \tag{6}$$

The turbulence sources $\omega_k(k, \varepsilon)$ and $\omega_\varepsilon(k, \varepsilon)$ depend on the k–ε model used. According to the standard Jones and Launder model (*viz*. Reference [2]),

$$\omega_k = P - \rho\varepsilon \tag{7}$$

$$\omega_\varepsilon = C_{\varepsilon 1} \frac{\varepsilon}{k} P - C_{\varepsilon 2} \frac{\rho \varepsilon^2}{k} \tag{8}$$

$$P = R_{ij} \frac{\partial \bar{u}_j}{\partial x_i} \tag{9}$$

The first and second term in (7) are, respectively, called the turbulent kinetic energy *production* P and *destruction* D.

Wall boundary conditions. In this work we use two different k–ε models: a *two-layer* one, introduced in Reference [17] and a *law-of-the-wall* model, *viz*. [7].

1. Two-layer model. In this case, turbulent transport equations are integrated down to the very physical wall. Due to the divergent value of ε there, turbulence near-wall behaviour should be considered carefully. On the one hand, an *ad hoc* damping function $f_\mu = f_\mu(y)$, multiplying the turbulent viscosity, eliminates its effect in the vicinity of the wall, where y is the distance to the wall. On the other hand, a special treatment is given to ε in the near-wall region, evaluating it from k instead of using its transport equation.
2. The law-of-the-wall model. In the second case, the *computational* wall is slightly off the *physical* one, leaving outside, i.e. unsolved, the conflictive region. This off-wall fictitious boundary becomes the real numerical one, where the tangential velocity is freed and a traction is imposed. To evaluate the traction, a hypothesis on the velocity dependency with the distance y in the excluded (normally very thin) region is assumed: the law of the wall. This law relates mean velocity with distance to the wall through friction velocity u_*, and is independent of inner and outer length scales. Assuming the hypotheses, a tangential traction $t^{\text{wall}} = n_i \sigma_{ij} g_j$ can be imposed, which depends on the friction velocity u_*. No additional corrections are considered in the compressible case.

3. DISCRETIZATION OF THE RANS + (k, ε) SET

The TuNS (or the LaNS) set is then discretized in space using the finite element method, as described in Reference [9] for the incompressible case or in Reference [16] (see also Reference [18] for a 1-equation model) for the compressible one. Briefly, on the one hand, the classical SUPG method [10] and the GLS one [12, 13], are used for the RANS (or the LaNS) set, for the incompressible and compressible cases respectively. The convective term is linearized using the Picard method. At each iteration, the linearized system is solved using a LDU incomplete preconditioned GMRES algorithm. On the other hand, the $(k$–$\varepsilon)$ are again discretized in space following the finite element method, but with the scheme introduced in Reference [19], linearizing the source terms in (7) and (8) according to

$$\omega_k = \left(P_{\text{Inc}} + \frac{2}{3}\mu_{\text{T}}\theta^2\right)^n - \left(\frac{2}{3}\rho\theta\right)^n k^{n+1} - \left(\frac{\rho\varepsilon}{k}\right)^n k^{n+1} \tag{10}$$

$$\omega_\varepsilon = C_{\varepsilon 1}\left(\frac{\varepsilon}{k}\left(P_{\text{Inc}} + \frac{2}{3}\mu_{\text{T}}\theta^2\right)\right)^n - C_{\varepsilon 1}\left(\frac{2}{3}\rho\theta\right)^n \varepsilon^{n+1} - C_{\varepsilon 2}\left(\frac{\rho\varepsilon}{k}\right)^n \varepsilon^{n+1} \tag{11}$$

and solving the linearized resulting system also with the GMRES preconditioned method. Here, the production has been decomposed in pure incompressible production and compressible effects:

$$P = P_{\text{Inc}} + \mu_{\text{T}}\frac{2}{3}\theta^2 - \frac{2}{3}\rho k\theta \tag{12}$$

$$P_{\text{Inc}} = -\mu_{\text{T}} 2 s_{ij}\frac{\partial \bar{u}_j}{\partial x_i} \tag{13}$$

Now, in both the in/compressible cases, the TuNS discretized problem can be written as follows: with the appropriate boundary conditions, find $\mathbf{x} = (\bar{U})$ ($\bar{U}$ note the mean variables

in both incompressible and compressible flow) and $\mathbf{y} = (k, \varepsilon)$, solution of

$$\begin{aligned} \mathbf{A}(\mathbf{x}, \boldsymbol{\mu}_{\mathrm{T}}(\mathbf{y})) \quad \mathbf{x} &= \mathbf{b} \\ \mathbf{C}(\mathbf{y}, \mathbf{x}) \quad \mathbf{y} &= \mathbf{f} \end{aligned} \tag{14}$$

where the coupling through $\boldsymbol{\mu}_{\mathrm{T}} = \boldsymbol{\mu}(y)$ is explicitly mentioned. The RANS unknown $\bar{U}$ represents mean velocity and pressure in the incompressible case and entropy variables in the compressible one. The solving strategy adopted is a 'staggered' smoother: at each Navier–Stokes iteration, one k–ε iteration is done.

Staggered smoother: $(\mathbf{x}^m, \mathbf{y}^m) = \Phi_{\mathrm{SS}}(m, b, f)$

Do m times

1. Solve $\mathbf{A}^n \ \mathbf{x}^{n+1} = \mathbf{b}$ where $\mathbf{A}^n = \mathbf{A}(\mathbf{x}^n, \mu_{\mathrm{T}}^n)$
2. Solve $\mathbf{C}^{n+1/2} \ \mathbf{y}^{n+1} = \mathbf{f}$ where $\mathbf{C}^{n+1/2} = \mathbf{C}(\mathbf{y}^n, \mathbf{x}^{n+1})$
3. Update $\mathbf{x}^n = \mathbf{x}^{n+1}$
4. Update $\mathbf{y}^n = \alpha_y \mathbf{y}^{n+1} + (1 - \alpha_y)\mathbf{y}^n$
5. Update $\mu_{\mathrm{T}}^n = \alpha_{\mu_{\mathrm{T}}} \mu_{\mathrm{T}}(\mathbf{y}^{n+1}) + (1 - \alpha_{\mu_{\mathrm{T}}})\mu_{\mathrm{T}}(\mathbf{y}^n)$

   and go back to 1.

We have observed that the use of a 'nested' smoother, where some k–ε iterations are done at each Navier–Stokes step does not lead to a faster marching process. In order to avoid some possible lack of robustness in the staggered smoother, the turbulent variables update can be done with two independent relaxation factors α_y and $\alpha_{\mu_{\mathrm{T}}}$. Regarding the iterative process to get stationary solutions and in order to attain good convergence rates for the individual flow solvers, we have chosen to solve the *transient* equations, meaning that the time derivative terms are not eliminated in the TuNS (or LaNS) set. In this way, the system matrices A and C are much better conditioned. The *fictitious* time interval (it is so because the iterative process does not correspond to a physical transient) is calculated from a CFL stability condition. Depending on the problem, $\mathrm{CFL} > 1$ can be used with remarkable convergence speed results (eventually, we have obtained good results for $\mathrm{CFL} > 10$). On the other hand, some unphysical initial conditions demand a $\mathrm{CFL} < 1$. We will return to this point later.

4. NON-LINEAR MULTIGRID APPLIED TO THE RANS + (k, ε) SET

4.1. Basic facts

Multigrid is a very popular and widespread technique for convergence speed-up. This idea was first applied to solve practical problems by A. Brandt as described in pioneering works [20, 1]. The concept behind these methods is based on two facts. First, different spatial frequencies errors are damped at different rates according to the following: the higher the frequency, the higher the rates. Second, higher frequencies are resolved only by finer grids. For that reason, alternative advance of the iterative procedure in grids of different element sizes, comprising a *hierarchy*, damps the errors acting selectively over the whole frequency spectrum. After some iterative steps in the finest grid, which smooths the error, the solution can be well

approximated in the next coarser mesh. The coarser mesh's right hand side is also modified by the addition of the current fine mesh residual, which is also transferred. It acts as a sort of injected source term, which drives the speed-up effect. There, some iterative steps are performed, the error is smoothed again and the solution and residuals are transferred to the following coarser mesh. This process continues until the coarsest mesh is reached and the coarse grid correction is transferred back to the finest grid. The whole process is repeated until some convergence criterium is accomplished. In references like [21, 22], complete overviews on the subject can be studied.

Loosely speaking, multigrid can be classified in *algebraic* and *geometrical multigrid*. In the first case, the hierarchy is constructed by means of 'stencils' which progressively reduce the rank of the original matrix (*viz*. Reference [23]). On the other hand, the geometrical approach reformulate the original continuum problem in different grain discretizations. In the present work, we follow this line, like in most of the finite element or volume algorithms.

The two basic ingredients of a multigrid scheme are:

- *The hierarchy of systems* ($\mathscr{H}$). The original continuum problem is discretized in a series of N grids $\Omega^{h_l}, 1<l<N$, having different (mean) cell sizes h, thus allowing an ordering according to the sizes. We call the *upper* problem the one solved in the finest grid Ω^{h_N}, i.e. the original one. The rest of the problems in the hierarchy are *lower* ones, $\Omega^{h_l}, 1<l<N-1$.
- *The transfer operators.* Data transfer between two given elements of $\mathscr{H}$ is done upwards through, say, F and downwards through, say, B. These generic operators act over discrete functions defined on the domain partitions. They are made of four basic ones. A first group does exclusively variables' transfers: a *restriction* r, which does it from fine to coarse grid and a *prolongation* p, which does the opposite. A second group, which is used for residuals, named here r^* and p^*, can be made in different ways, taking as a starting point the transposed of the first group. In multigrid schemes only p^* is relevant. Restriction and prolongation operators are constructed inheriting FEM properties. First, each node of a given spatial grid is located in the corresponding element of the neighbouring hierarchy grids. Then, its interpolation function is evaluated using the FEM shape functions.

As introduced in Reference [4], we consider several definitions of p^*, all of them functions of the transpose p^{t}. Firstly, it is classical to take directly

$$p^*_{\mathrm{tra}} = p^{\mathrm{t}} \tag{15}$$

After observing a bunch of cases, we can conclude that this operator provides the fastest speed-up convergence... when it converges. Anything transfered by this operator is increased proportionally as A_{l-1}/A_l, where the ratio of the areas A is locally evaluated. When p^*_{tra} is used to transfer the residuals to coarser grids the multigrid has proven to be less robust because the hierarchy needs to be constructed keeping $A_{l-1}/A_l<L$ *everywhere*, where the proper L can strongly depend on the problem, say a maximum value around 10. If this is not accomplished, the convergence and solution of the problem is not guaranteed.

In Reference [24] and in the context of Chimera-type or overset grids domain decomposition methods, the authors proposed an operator that was afterwards extended to multigrid to be used

as p^* [4]. It is formed by p^t, but column-wise normalized: p^*_{cwn}:

$$p^*_{cwn} = [p \ \mathrm{diag}(1/\beta_1, 1/\beta_2, \ldots, 1/\beta_{N_C})]^t \tag{16}$$

where N_F and N_C are the number of nodes of the fine and coarse grids, respectively, and β_i is the norm of each column i of p. According to the definition of p, each of its column norms is approximately (exactly in nested regular grids) the ratio between the areas of the elements connected by the operator. In this way, the use of p^*_{cwn} yields more 'coarsening independent' strategies than when p^*_{tra} is used, i.e. more robust. This gives a clear idea of what we are looking for: if for a given hierarchy, p^*_{tra} works well, then p^*_{cwn} will lead to a speed-up indeed, but probably lower than that of the former. And for that reason, it is expected that p^*_{cwn} could do it well even when p^*_{tra} is not working at all.

In general, multigrid processes follow different cycling strategies, V- or W-cycle. V-means that coarse grid correction and smoothing stages are done in a straight way, from lower to upper grids and *vice versa*. W-stands for the presence of 'u-turns' in intermediate grids. Additionally, the coarse grid correction stage can also be done with internal *post-smoothing* steps. We believe that all these possibilities should be allowed in the multigrid code because which of them is the fastest one depends on the problem.

A startup multigrid phase, what we call the *cascadic stage*, can be of great importance. Sometimes (but not always, indeed), a coarse mesh left alone can produce a good initial condition for a finer discretization when transferred up. In this way, the whole process starts in the coarsest discretization, where a certain number of steps are done. Then, the unknowns just obtained are prolonged to the second coarsest mesh. After a given number of cycles between both grids, the result is prolonged up to the third coarser, where more multigrid cycles can be done. This is done until reaching the finest mesh, where a V- or W-cycle multigrid begins until final convergence is reached. This strategy is usually known as *full multigrid* or *F-cycle* [22]. Sometimes, this is a very good idea: even when no multigrid is done after the cascadic cycle, a very important speed-up can be attained. However, depending on how coarse is the coarse discretization, and particularly for some problems, the solution in the coarser previous mesh can be a very bad initial condition for the next finer one.

4.2. *Application to the* RANS + (k, ε) *set*

As was said before, the discretization in space of the RANS set by the FEM produces a non-linear system of equations. Also, being the problem non-linear, its solution is achieved through linearization steps and at each step the system is solved implicitly. The solver's choice bias the multigrid strategy to follow. In this case, the FEM induces the use of a geometric approach. As described in the previous section, it allows an easier construction of the transfer operators between each element of the hierarchy, taking profit of its interpolation functions. On the other hand, the problem's non-linearity leads to a multigrid method that can cope with it, like a non-linear multigrid scheme considered here.

To apply multigrid to accelerate the convergence rate of solvers for non-linear systems is not a new idea. In Reference [21], the FAS (full approximation scheme) algorithm of Brandt [1] is cited as the first non-linear multigrid to appear. The present work follows the same line. Basically, the equations solver, i.e. the *smoother* Φ, is non-linear and the system is not solved on increments of the unknown, but on the unknown itself. The multigrid cycling is embedded in the whole solver, from *outside* the linearization cycle. Recent works using

non-linear multigrid techniques for solving these problems with the same kind of smoother are those of Reference [25] or Reference [26].

The non-linear multigrid process as described in the precedent section is applied to the problem (14). Now the smoother has two components: Φ_{RANS} and Φ_{TUR} and the unknowns, sources and matrix are, respectively $(\mathbf{x},\mathbf{y})$, $(\mathbf{b},\mathbf{f})$ and a matrix formed by two blocks with $(\mathbf{A},\mathbf{C})$. In this way, multigrid is acting on the *whole* problem. As observed by the authors, this *complete approach* gives the best results. Then, the proposed non-linear multigrid is summarized as follows.

- *Smoothing stage.* After n steps of the chosen smoother, the laminar and turbulent residuals $\mathbf{d}_{\mathrm{lam,F}}$ and $\mathbf{d}_{\mathrm{tur,F}}$, the variables $\mathbf{x}_{\mathrm{F}}^n$ and $\mathbf{y}_{\mathrm{F}}^n$ and the turbulent viscosity μ_{TF}^n are transferred from the *fine* grid to the *coarse* one according to

$$\begin{aligned}
\mathbf{d}_{\mathrm{lam,C}} &= p^*\mathbf{d}_{\mathrm{lam,F}} \quad \text{where } \mathbf{d}_{\mathrm{lam,F}} = \mathbf{b}_{\mathrm{F}} - \mathbf{A}_{\mathrm{F}}^n\mathbf{x}_{\mathrm{F}}^n \\
\mathbf{d}_{\mathrm{tur,C}} &= p^*\mathbf{d}_{\mathrm{tur,F}} \quad \text{where } \mathbf{d}_{\mathrm{tur,F}} = \mathbf{f}_{\mathrm{F}} - \mathbf{C}_{\mathrm{F}}^n\mathbf{y}_{\mathrm{F}}^n \\
\mathbf{x}_{\mathrm{C}}^0 &= \mathrm{r}\ \mathbf{x}_{\mathrm{F}}^n \\
\mathbf{y}_{\mathrm{C}}^0 &= \mathrm{r}\ \mathbf{y}_{\mathrm{F}}^n \\
\mu_{\mathrm{TC}} &= \mathrm{r}\ \mu_{\mathrm{TF}}^n
\end{aligned} \tag{17}$$

  The last transfer from fine to coarse, for the turbulent viscosity μ_{T}, will eventually lead to the fact that (4) is not verified in the coarse mesh. However, we have observed that this favours the overall robustness of the scheme, particularly from oscillations around shocks and boundary layers.
- *Coarse grid correction stage.* After m steps applied with the chosen smoother, the laminar and turbulent coarse grid corrections $\Delta\mathbf{x}_{\mathrm{C}}$ and $\Delta\mathbf{y}_{\mathrm{C}}$ are transferred back from the *coarse* grid to the *fine* one according to

$$\begin{aligned}
\Delta\mathbf{x}_{\mathrm{F}} &= p\ \Delta\mathbf{x}_{\mathrm{C}} \quad \text{where } \Delta x_{\mathrm{C}} = \mathbf{x}_{\mathrm{C}}^m - \mathbf{x}_{\mathrm{C}}^0 \\
\Delta\mathbf{y}_{\mathrm{F}} &= p\ \Delta\mathbf{y}_{\mathrm{C}} \quad \text{where } \Delta\mathbf{y}_{\mathrm{C}} = \mathbf{y}_{\mathrm{C}}^m - \mathbf{y}_{\mathrm{C}}^0
\end{aligned} \tag{18}$$

This is our default scheme. It is programmed in a master–slave strategy: one multigrid code connected (via PVM, MPI or directly by UNIX sockets) to different slave solver codes. In our case, the slaves can be in/compressible fractional step solvers (previously studied in References [3, 27]), incompressible monolithic ones [4] or compressible monolithic ones, as presented here.

4.3. The robustness issue

Several implementation problems can arise when multigrid is plainly applied, caused by the non-linearity of the original solver, the use of unstructured meshes, hierarchy construction, three-dimensionality, the turbulent coupling and so on. Some points are critical to improve the convergence, some others the robustness and reliability of the scheme and some others to

keep both under control. We sum up below some of the difficulties we have faced and the implementation solutions we adopted, including those discussed above.

- *Boundary conditions updates.* Based on our own experience, boundary conditions should *not* be updated in the lower grids of the hierarchy, but only in the upper one. In this way, the velocity values in the lower grids will be fixed to those that come from the upper one. This is particularly important in the case of curved boundaries. For instance, when a non-slipping boundary is present, the velocity interpolation from the upper grid to a lower one will (very likely) result in non-zero velocity in the corresponding wall nodes. This interpolated value is the proper velocity prescription for wall nodes in the coarse grid embedded in a multigrid process, as opposed to the zero velocity prescription for the single grid process. The same rule applies to turbulence unknowns, but in this case, checking the *positivity* of the interpolated values, because wall values for k and ε cannot be negative numbers. This can happen for nodes falling inside the numerical boundary defined by the upper grid (that is to say *outside* the numerical domain). All this becomes crucial in turbulent boundary layers due to the very strong gradients.
- *Nodes elimination.* Connected with the boundary conditions for the lower grids, another possibility that we have studied is to eliminate nodes from the coarse grids. Taking into account the distance to the 'numerical wall' as defined in the upper grid, all the first stripe of nodes (and maybe the second too) can be 'eliminated' from the flow solving process in the coarsest grids by simply fixing the unknowns that has came from the next upper one. This distance to the wall is the same that is used for the two-layers model, so no additional evaluation is needed. In this way, the coarse grids can be constructed (recall we consider here independent mesh generation, without any agglomeration or coarsening) with their wall nodes slightly off the real numerical wall. Therefore, the problems caused by the coarse grids' nodes falling inside the body will be avoided. This idea can be pushed further to what the authors name a *patch multigrid*, that will be faced in future works. We will come back to the point in the concluding section.
- *Transfer operators construction.* As said before, we are looking for a robust multigrid strategy, which keeps positive speed-up rates even for very lax hierarchy construction. On these grounds, the problem of designing an efficient and reasonably automatized hierarchy construction algorithm can then be faced. We see through the examples the importance of the transfer operator construction. The CWN and SEL (which is the CWN operator with *a cut off value for the ratio between areas*, see Reference [4]) operators studied are very easy to build, and all together with classical TRA range from a 'conservative' strategy (CWN) to an 'aggressive' one (TRA), with SEL operator between them, which in turn can be tuned by the choice of the cut off value L_c. This operators allows the use of a loosely constructed hierarchy, where the surface ratio between the elements of neighbouring grids can reach high values. That is to say, when out of two grids, the coarse grid is 'too coarse'.
- *Transfer relaxation.* Residual or coarse grid correction relaxations are also interesting solutions for improve robustness. Due to the fact that operators like CWN and SEL act on residual transference, we have observed that any kind of residual relaxation is unnecessary combined with these operators, for they provide a sort of locally defined relaxation. On the other hand, coarse grid relaxation is indeed very useful, specially when high gradients are present in turbulent boundary layers and shocks. While for the RANS

and LaNS equations it is implemented as a coarse grid correction reduction by a factor $0.0 \leqslant \alpha_{\mathrm{CGC}} \leqslant 1.0$:

$$\mathbf{x}_{\mathrm{F}}^{n+1} = \mathbf{x}_{\mathrm{F}}^{n} + \alpha_{\mathrm{CGC}} \Delta \mathbf{x}_{\mathrm{F}} \tag{19}$$

for the turbulent variables, it is preferred the relaxation proposed in Reference [28]:

$$\mathbf{y}_{\mathrm{F}}^{n+1} = \mathbf{y}_{\mathrm{F}}^{n} \left(\frac{\mathbf{y}_{\mathrm{F}}^{n} + \boldsymbol{\Delta}^{+}}{\mathbf{y}_{\mathrm{F}}^{n} - \boldsymbol{\Delta}^{-}} \right) \tag{20}$$

where

$$\boldsymbol{\Delta}^{+} = \alpha_{\mathrm{CGC}} \Delta \mathbf{y}_{\mathrm{F}} \quad \text{and} \quad \boldsymbol{\Delta}^{-} = 0 \quad \text{if } \Delta \mathbf{y}_{\mathrm{F}} > 0$$

$$\boldsymbol{\Delta}^{-} = -\alpha_{\mathrm{CGC}} \Delta \mathbf{y}_{\mathrm{F}} \quad \text{and} \quad \boldsymbol{\Delta}^{+} = 0 \quad \text{if } \Delta \mathbf{y}_{\mathrm{F}} < 0$$

This kind of relaxation gives a much better control over the positivity of the turbulent variables in the transient iterative process. We have used it in all the compressible cases, we found that a reasonable value for α_{CGC} is 0.5.

- *Adaptive and fixed V/W-cyclings, full multigrid and cascadic multigrid.* As said above, all along this work, by V- or W-cycle multigrid we understand the classical cycling (*viz*. Reference [21]), which starts the whole solution process in the upper grid. We have seen that instead of a fixed strategy, which keeps constant the iterations in each of the grids, an *adaptive* cycling should not be discarded. It can be easily implemented by checking the residual evolution in the coarse grids. On the other hand, by F-cycle, or full multigrid (*viz*. Reference [22]), it is understood a process that begins in the lower grid. In this sense the F-cycle comprises two major stages: the first goes upwards, transferring only variables. The initial condition at each of the upper grids is *eventually* closer to the solution there, yielding a faster convergence rate. When this grid sequencing stage is the sole solution process, it is known as *cascadic multigrid* (like in Reference [29] or Reference [30]). But once the upper grid is reached, a second cycling stage can be carried out, either using a V- or a W-cycle, resulting in an F-cycle. We will call these major stages as the *cascadic stage* and the *cyclic stage* respectively. As a refinement of the F-cycle strategy, the lowest grids can be discarded after the cascadic stage, keeping the upper ones for the cyclic stage. Another possibility is the use of a different set of boundary conditions in the coarse grid to be discarded: a wall law condition instead of a no-slip one. Across the whole range of problems studied, we have seen the importance of the cascadic stage. While for some problems it is completely useful, bearing most of the weight of the speed up process, for some other problems it must be absolutely discarded.
- *Freezing turbulent sources.* Turbulent source terms, namely production and destruction terms ($P+D$), can be wrongly computed in the coarse grids, producing strong instabilities that can slow down the scheme's speed-up properties or even spoil the convergence. We have seen that this can be avoided by 'freezing' the source terms during the smoothing stage. By freezing we understand that source terms are calculated only in the upper grid and transferred naturally as part of the residual to the rest of the hierarchy. This point is very important in compressible cases.

- *Increasing solver robustness.* We have observed very strange things happening in compressible cases. Almost converged multigrid solutions can experiment sudden and completely unattained blows. By carefully following the residual evolution, we have seen that this happens in the vicinity of shocks. A very effective way of avoiding this is by increasing the solver's own robustness through minor modifications. Relaxing turbulent variables updates, and especially the turbulent viscosity is a must (by means of two different relaxation factors α_y and α_{μ_T}). This is independent of the multigrid process.
- *Modifying solver numerical strategy.* As said in precedent sections, on each individual flow solver we follow a transient-like iterative strategy. Initial non-physical conditions (like constant velocities or densities) can lead to very strong transient instabilities that are supposed to be damped by the solution process. Indeed it happens, but at the price that the CFL condition number used to calculate Δt (see Section 3) has to be lower, particularly in high Mach number viscous problems. On the other hand, when a cascadic stage precedes the multigrid, this low CFL number is confined to the first, very coarse mesh, which runs very fast. The initial condition that now passes to the next finer discretization is smoother, allowing much larger CFL numbers, which greatly improves overall convergence.

We look for solutions that could always be applied, in the quest for a multigrid process as automatic as possible. They should be applied in the same form and with the same parameters in all problems: laminar/turbulent, in/compressible, viscous/inviscid, 2/3D, always leading to improved convergence rates. Therefore, cyclings, pre/post smoothing steps, cascadic steps, freezing, relaxation factors were kept as fixed as allowed by the problems. Also, we constructed the hierarchies in a completely independent way, i.e. no coarsening or adaptivity strategies were used, keeping in mind that the use of these strategies in a next stage will lead to even better multigrids, once the robustness issue is analysed. Hierarchies are then constructed by simply changing the mesh generator parameters to produce grids of different sizes. However, some mesh generators allows to construct a coarser grid by de-refining a fine mesh following a lax coarsening process. In particular we have tested C.O. Gooch's code GRUMMP (http://tetra.mech.ubc.ca/GRUMMP) to produce the coarse grids, with very interesting results. Additionally, we used the same numerical strategy in all the solvers involved: the same preconditioners, GMRES parameters, etc. This is also a very important point, that we left out of the scope of this paper. In Reference [4] we have shown that, rather unexpectedly, the GMRES precision (Krylov number, tolerances) can be much lower in the upper grid than in the single grid problem, further improving the multigrid speed-up.

5. NUMERICAL EXAMPLES

In this paper, we concentrate on compressible flow examples. Incompressible flow problems have been considered in the referenced previous papers by the authors. Therefore, the former is analysed through a supersonic (at Mach 4) double wedge configuration at three different regimes: laminar inviscid, laminar viscous and turbulent. The latter is here analysed only for the turbulent regime through two examples: what we call a step-function-inflow tube and a three element airfoil at different incidence angles. All of the contour level pictures are shown

here just to give a qualitative idea of the complexity of the flow solved. Therefore, the level values are never shown.

5.1. Compressible flows

A supersonic double wedge at Mach 4, is considered as the test case. Both of the wedges lengths are 1.0 (this length 1.0 is used to calculate the Reynolds number) and their steepness are 15 and 35°, respectively. We choose it as benchmark because it can develop a wide variety of physical features depending on the flow regime, using basically the same domain (we follow Reference [31]). A shock produced by the first 15° wedge interacts with a second one, produced in turn by the 35° wedge. In the case of laminar, viscous flows, the second shock interacts with the boundary layer and a long vortex is formed. In the turbulent case, this vortex disappears.

The three considered regimes are:

(i) Mach 4, laminar, inviscid flow.
(ii) Mach 4, laminar, $Re = 10^6$.
(iii) Mach 4, turbulent, $Re = 10^6$, 2-layer model.

Multigrid strategy. In all of the regimes, we have followed the strategy outlined below:

1. Three-grid non-nested hierarchy. The coarsest grid is unstructured, both the medium and finest ones are structured. For the laminar problems ((i) and (ii)), the first node off the wall is at about 0.008 units, in order to resolve the boundary layer and the strong vortex. The medium grid is obtained by approximately doubling the spacings. However, in the direction normal to the wedges, due to the boundary layer refinement the elements surface ratio is more than four, rising up to 12. On the other hand, for the turbulent problem (iii), the finest mesh is finer, with the first node off at 0.0005. The medium and coarse grids are the same as before. The turbulent problem is a perfect test to evaluate the robustness of the CWN operator: the loosely constructed hierarchy plainly fails when used in combination with a traditional operator due to the element surface ratio in the boundary layer, which in (iii) can rise up to 50.
2. Cascadic 3-grid stage, followed by a multigrid 2-grid stage. It starts in the coarsest grid, which is afterwards discarded and leaving only the medium and finest meshes until convergence is reached thanks to a V-cycle multigrid strategy. This strategy is labelled as '1+2 CasMG+MG' in the convergence plots.
3. The cascadic stage allows to increase the CFL number once finished. The CFL number in the first (coarse) mesh is less than one (0.3 or 0.5). In the other grids it is about 10 times larger (3, 5, or even 10).

(i) *Mach 4*, *laminar*, *inviscid flow*. The Mach number contours are shown in Figure 1, top. A shock is formed before the first wedge, which interacts in turn with a second one formed after the next wedge. The finest and coarsest grids of the hierarchy are shown in Figure 2. The first nodes off the wall are at about 0.008 unit lengths. The convergence plot (Figure 3, top) shows an important speed-up. Between the '1+2 CasMG+MG' and 'Single' plots, it is shown also the '3 CasMG', that labels the pure cascadic multigrid. In this Euler problem, it can be a good strategy, although not as good as '1+2 CasMG+MG'. The classical TRA residual restriction operator is used.

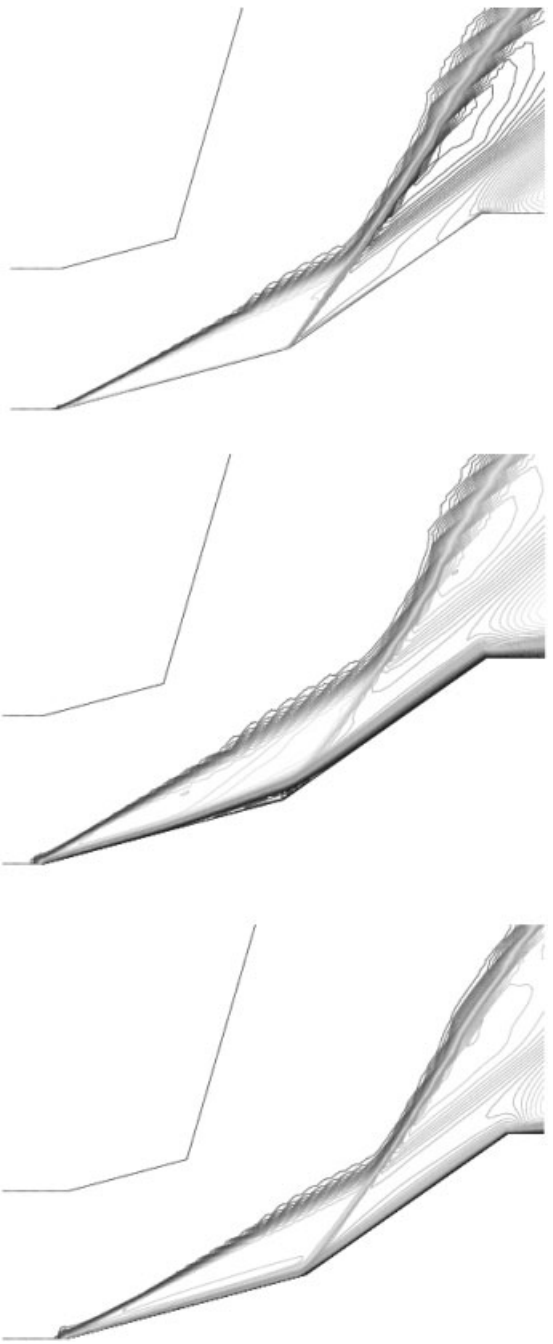

Figure 1. Double wedge, Mach 4. Mach number contours. Top, (i) laminar, inviscid flow. Medium, (ii) laminar, $Re = 10^6$. Bottom, (iii) turbulent, $Re = 10^6$.

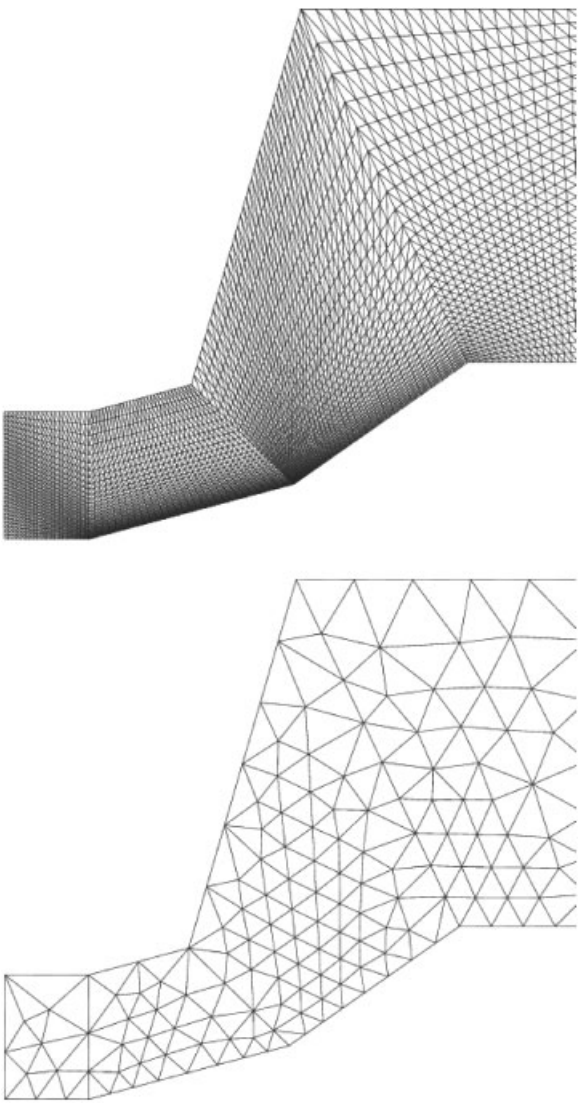

Figure 2. Double wedge, Mach 4. Top, fine grid (for (i) and (ii)). Bottom, coarse grid.

(ii) *Mach 4*, *laminar*, $Re = 10^6$. The Mach number contours are shown in Figure 1, medium. The interaction between the second shock and the boundary layer produces a separation zone, with a large vortex. The hierarchy and the strategy are the same as in the previous case. The shock–boundary layer interaction presents usually a very slow convergence rate, due to the progressive vortex formation in the separation zone. In this case, multigrid is particularly effective (Figure 3, medium). Again, the classical TRA residual restriction operator is used.

(iii) *Mach 4*, *turbulent*, $Re = 10^6$, 2-*layer model.* The Mach number contours are shown in Figure 1, bottom. The vortex has disappeared, dissipated by the turbulent viscosity effects.

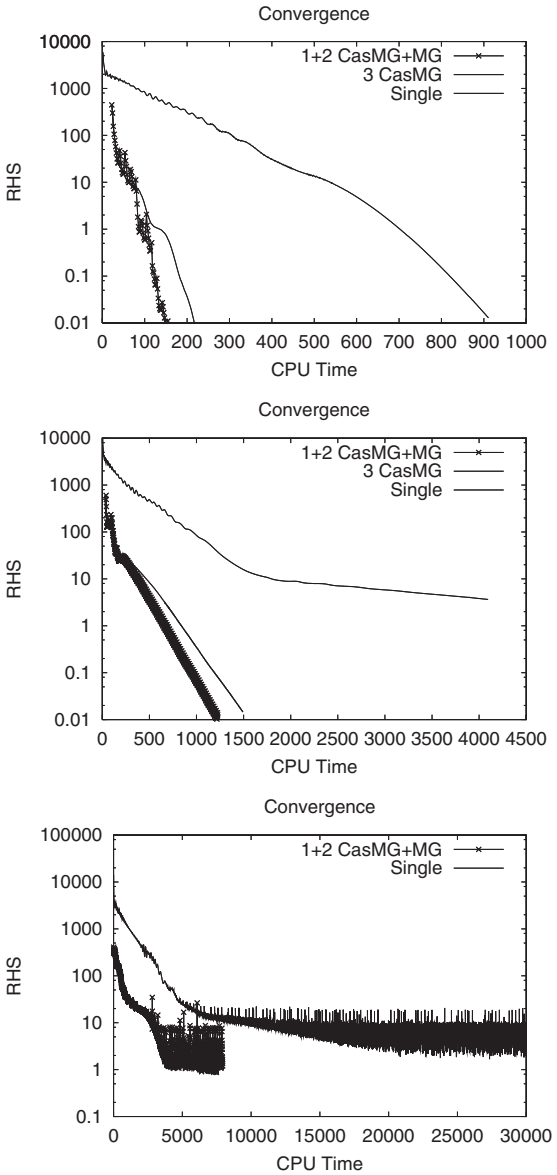

Figure 3. Double wedge, Mach 4. Convergence plots. Top, case (i), medium, case (ii) bottom, case (iii).

The turbulence model is the 2-layer one, with no compressibility corrections at all. While the cycling strategy remains the same, the fine grid is here finer, as described above, in order to capture the turbulent boundary layer effects. In this case, we have evaluated the '1+2 CasMG+MG' strategy (Figure 3, bottom), which again has proven to be very effective. In this

case, the CWN operator is used. Turbulent source freezing and coarse grid correction relaxation ($\alpha_{CGC} = 0.5$ as described in the preceding section) are definitely needed. The oscillations in the queue of the convergence graphs are due to the shock capturing numerical diffusion implemented in the turbulent compressible flow code, being spatially confined around the shocks. This numerical effect has been seen before and, although negligible in most of the cases, it can be eliminated by implementing a more sophisticated residual dependent shock capturing diffusion. In this case, it represents an oscillation in the convergence between the 4th and 6th degree of magnitude, appearing also in the single mesh results.

5.2. *Incompressible flows*

In Reference [4], we have focused in incompressible flows. There, some of the concepts presented here to improve robustness were introduced, the analysis going from laminar to turbulent problems. So here we restrict to turbulent problems. Two examples are shown: a step-function-inflow tube and a triple element airfoil.

5.2.1. Step-function-inflow tube. The computational domain is a rectangle with height:length ratio of 2:30 (Figure 4). We have called this example 'step-function-inflow tube' because of the step distribution of the inflow velocity prescription. The left side upper half (length L) is the inflow, where a constant horizontal velocity is prescribed. The left side lower half (length L too) and the bottom are non-slipping boundaries. This produces a large vortex just downstream of the inflow. Finally, the right side is the outflow and the top is a slipping boundary. The Reynolds number computed using L is $Re = 10^5$. The inflow prescriptions are

$$\begin{aligned} \mathbf{u}_{\text{inf}} &= (1,0) \\ k_{\text{inf}} &= \beta u_{\text{inf}}^2 \\ \mu_{T\,\text{inf}} &= \alpha\mu \\ \varepsilon_{\text{inf}} &= C_\mu k_{\text{inf}}^2/\mu_{T\,\text{inf}} \end{aligned} \tag{21}$$

where $\Delta = 10^{-4}$, $\beta = 0.001$ and the input turbulent viscosity factor $\alpha = 100$, obtained now by fixing k and μ_T, and deriving ε from them. The two-layer model described above [17] is used in this example. Therefore, u_i, k and ε are set to zero at the non-slipping boundaries.

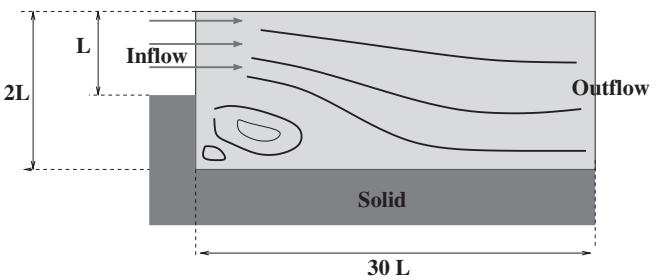

Figure 4. Step-function-inflow tube. Problem and boundary conditions.

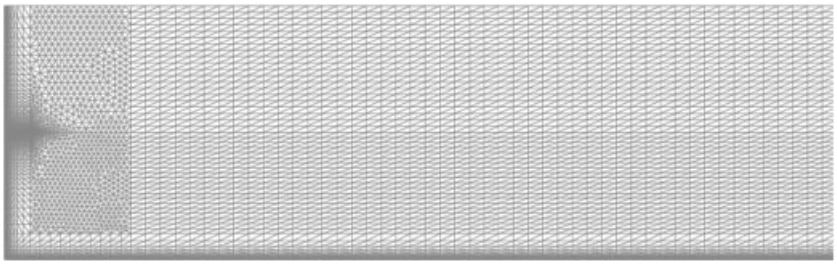

Figure 5. Step-function-inflow tube. Upper grid (close-up near the inflow).

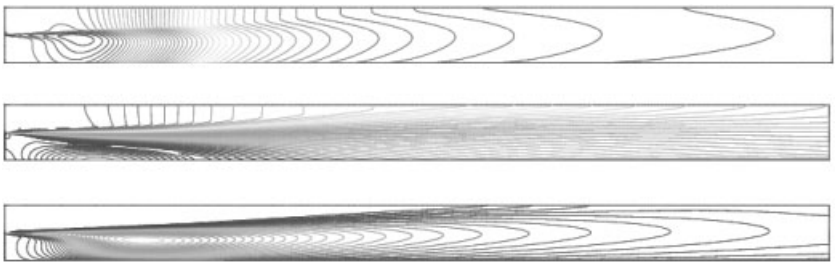

Figure 6. Step-function-inflow tube. Pressure, velocity and turbulent kinetic energy contours.

The domain is discretized with a mixed structured and non-structured grid (Figure 5). The upper grid is made of 13221 nodes and 25894 P1 elements. Pressure, velocity and turbulent kinetic energy are shown in Figure 6.

Multigrid strategy. In this example, a 3-grid hierarchy produces a speed-up of around 5 (shown in Figure 7). This is a typical case where a cascadic initial stage is a very bad choice, because the coarse meshes are 'too coarse' to produce a good initial state for the upper grid. The complete multigrid strategy for this problem is:

1. Three-grid non-nested hierarchy. The hierarchy comprises 3 grids constructed by doubling the local element lengths of the precedent grid. This doubling is done approximately, except in the structured and homogeneous zones. Again, this procedure leads to a very lax constructed hierarchy. The first node's distance to the wall for the finest grid is approximately 0.0001.
2. No cascadic stage.
3. V-cycle multigrid, with source freezing for the turbulent variables, SEL operator with $L_c = 10$ and CGC relaxation with $\alpha_{CGC} = 0.5$. The same GMRES parameters are used in all the cases. One smoothing iteration is done in each grid, with no post-smoothing.

5.2.2. Triple element airfoil with different incidence angles. This high lift device is solved using the k–ε model as described in previous sections. The boundary condition imposed in the profiles comes from the extended law of the wall. The Reynolds number per unit length is $Re_L = 3.6 \times 10^6$. We show here the convergence improvement for two different incidence

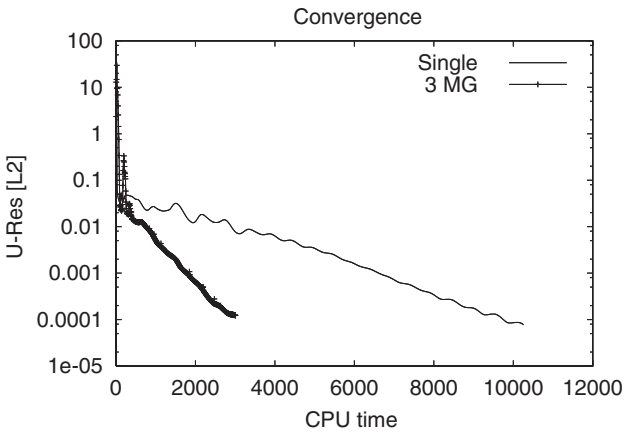

Figure 7. Step-function-inflow tube. Convergence graph.

angles: 12° and 16°. Inflow conditions are those of (21), except for the inflow velocity, that varies according to the incidence angles (Figures 8 and 9).

Multigrid strategy. A 3-grid hierarchy produces in this case speed-ups up to almost 10. In Figure 10 the iteration evolution of the pressure lift coefficient is shown for two different incidence angles. These angles were chosen in order to prevent flow separation because in this particular case a wall law model is studied, leaving for future works the scheme performance assessment at larger angles, using a model with low-Reynolds' number correction.

The strategy adopted follows these lines:

1. Three-grid non-nested hierarchy. In this case the first coarse grid was constructed by de-refinement of the upper one. This was done using C.O. Gooch's GRUMMP code, which takes as input a given mesh and construct a coarser one, isotropic and unstructured by approximately doubling the local element size. This can be done very loosely, just by giving the factor by which the coarsening is done. As reported in its web page (http://tetra.mech.ubc.ca/GRUMMP) it works for unstructured 2D and 3D meshes, without high aspect ratio elements (we have only tested the 2D option). The coarsest grid was constructed independently.
2. As in the compressible cases, we used a cascadic 3-grid stage, followed by a multigrid 2-grid stage, labelled '1+2 CasMG+MG'. The lower grid solution, that is passed to the next upper grid to be used as an initial condition there is obtained also with a wall law model, but 'placing' the numerical wall at a large y^+, that is to say, the flow is there almost free to tangentially slip, with a low tangential traction t^{wall} prescribed.
3. Due to the curvature of the profiles and the hierarchy construction procedure, the profiles boundary condition for the lower grid once the cascadic stage is over is fixed from the values that come from the upper grid after each transferring, checking also the positivity of the turbulent variables.
4. V-cycle multigrid, no source freezing for the turbulent variables, CWN operator, no relaxation.

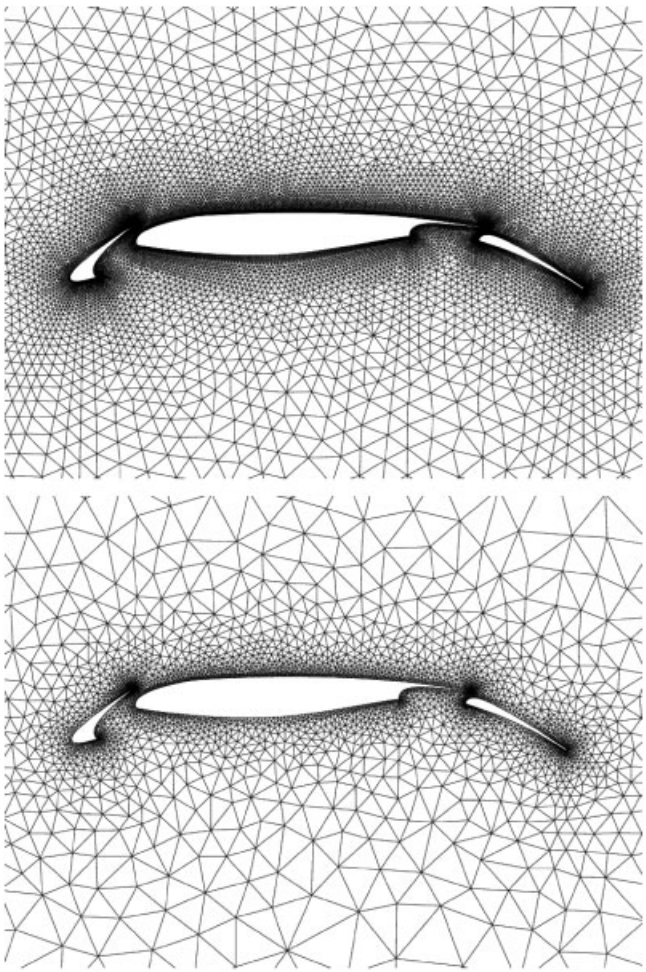

Figure 8. Triple element airfoil. Pressure contours. Top, fine grid. Bottom, next coarse grid, produce by coarsening using GRUMMP.

5. The cascadic stage allows again to increase the CFL number once finished. The maximum speed-ups are reached if after the cascadic stage, the CFL number for the finest grid rise up to 100 (curves '(b)' in Figure 10), i.e. ten times larger than for the single grid case or the multigrid with equal CFLs (resp., curves labelled 'Single' and '(a)' in Figure 10). We recall that for the single grid, this large CFL is completely banned from the very beginning of the iterative process, allowed only after many iterations with smaller CFL number have been done and when the forming boundary layer gradients are greatly smoothed. The CPU time offset seen in curves '(a)' and '(b)' in Figure 10 accounts for time spent by the cascadic stage.

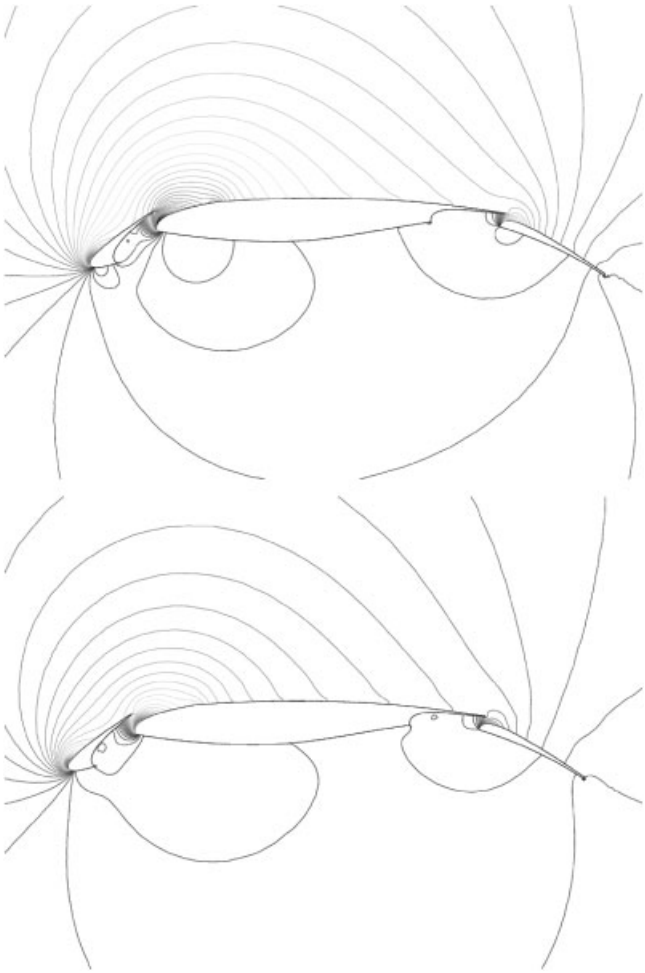

Figure 9. Triple element airfoil. Pressure contours. Top, $\alpha = 12^\circ$. Bottom, $\alpha = 16^\circ$.

6. CONCLUSIONS AND FUTURE LINES

In this paper we have presented some ideas intended to render more robust non-linear multigrid schemes applied to speed-up the convergence of the Navier–Stokes equations solution process. These ideas can be applied to the widest range of problems: compressible and incompressible, viscous and inviscid, laminar and turbulent, and were consequently tested here under all these flow regimes. We have observed that many times the robustness of a scheme is a somewhat forgotten issue. It is not a matter of how efficient to accelerate convergence rates can be a multigrid scheme proposed, but if it will give *any* convergence at all for a different problem. This lack of robustness is basically derived from the high non-linearity that can present the Navier–Stokes' equations, related to high Reynolds numbers, compressibility effects and turbulence modelling. We have shown that using rather simple ideas on boundary

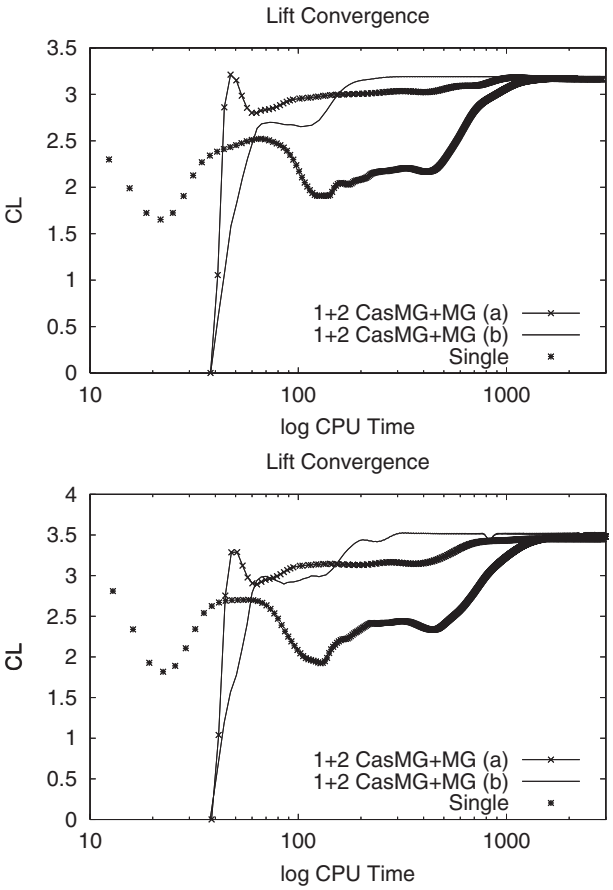

Figure 10. Triple element airfoil. Lift convergence. Top, $\alpha = 12^{\circ}$. Bottom, $\alpha = 16^{\circ}$.

conditions, relaxation, cycling strategies, operators construction, etc., a multigrid scheme can really improve this point. Moreover, all the examples shown here *need* some of these ideas to run.

On the pure implementation side, we remark the modular *master MG—slaves flow solvers* strategy. This has allowed the authors to work in a very flexible fashion. We started with the laminar incompressible, then we pass to the turbulent incompressible and finally to the turbulent compressible directly. The modular implementation has made that each of the steps includes the improvements of the previous one and that the time spent becomes smaller at each successive step. As a bonus, techniques like domain decomposition can also be studied with very little additional programming effort and even combined with multigrid. Particularly, this line can be explored in the future. The first results on what we called *patch multigrid* are indeed encouraging. The domain is decomposed in Chimera type or overset sub-domains, a 'background' and one or more 'patches'. In this way more than one multigrid thread can be constructed, at almost no additional cost, for the operators are constructed once. To the advantages of the decomposition it is added the speed-up of the multigrid.

The examples presented here, the supersonic double wedge and both the incompressible turbulent step-function-inflow tube and the triple element airfoil, are 2D examples. However, due to the way these ideas have been developed, we believe that it can be almost directly tested in 3D problems. Extensive tests are to be done in the future in this line.

ACKNOWLEDGEMENTS

One of the authors (M. Vázquez) thanks the European Commission for the Marie Curie Fellowship held by him while he was in Dassault Aviation-UPMC's Pôle Scientifique doing the research project summarized in this paper.

REFERENCES

1. Brandt A. Multi-level adaptive solutions to boundary value problems. *Mathematics of Computation* 1977; **31**:333–390.
2. Launder BE, Spalding DB. The numerical computation of turbulent flows. *Computer Methods in Applied Mechanics and Engineering* 1974; **3**:269–289.
3. Vázquez M. Numerical modelling of compressible laminar and turbulent flow. The Characteristic Based Split (CBS) Finite Element General Algorithm. *Doctoral Thesis.* Escola Tècnica Superior d'Enginyers de Camins, Canals i Ports. Universitat Politècnica de Catalunya, Barcelona, 1999.
4. Vázquez M, Ravachol M, Mallet M. Multigrid applied to a fully implicit FEM solver for turbulent incompressible flows. *Proceedings of the ECCOMAS 2001 Computational Fluid Dynamics Conference*, Swansea, UK. IMA, (edited in CD ROM) 2001.
5. Lesieur M, Métais O. New trends in large-eddy simulations of turbulence. *Annual Review of Fluid Mechanics* 1996; **28**:45–82.
6. Holmes P, Lumley JL, Berkooz G. *Turbulence*, *Coherent Structures*, *Dynamical Systems and Symmetry*. Cambridge University Press: Cambridge, 1996.
7. Wilcox DC. *Turbulence Modeling for CFD*. DCW Industries, 1993.
8. Batchelor GK. *An Introduction to Fluid Dynamics*. Cambridge University Press: Cambridge, 1967.
9. Ravachol M. Unstructured Finite Elements for Incompressible Flows. *AIAA* **97-1864**, 67–75, 1997.
10. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes Equation. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
11. Hughes TJR, Franca LP, Mallet M. A new finite element method for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier–Stokes equations and the Second Law of the Thermodynamics. *Computer Methods in Applied Mechanics and Engineering* 1986; **54**:223–234.
12. Hughes TJR, Franca LP, Hulbert GM. A new finite element method for computational fluid dynamics: VIII. The Galerkin/Least-Squares method for multidimensional advective–diffusive equations. *Computer Methods in Applied Mechanics and Engineering* 1989; **73**:173–189.
13. Shakib F, Hughes TJR, Johan Z. A new finite element method for computational fluid dynamics: X. The compressible Euler and Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1991; **89**:141–219.
14. McComb WD. *The Physics of Fluid Turbulence*. Oxford University Press: Oxford, 1990.
15. Harlow FH, Nakayama P. Transport of turbulence energy decay rate. Los Alamos Science Lab., *University California Report* LA-3854, 1968.
16. Hauke G. *A Unified Approach to Compressible and Incompressible Flows and a New Entropy-consistent Formulation of the k–ε Model*. Department of Mechanical Engineering, Stanford University: Stanford, 1995.
17. Chen HC, Patel VC. Near-wall turbulence models for complex flows including separation. *AIAA Journal* 1988; **26**:641–648.
18. Jansen K, Johan Z, Hughes TJR. Implementation of a one-equation turbulence model within a stabilized finite element formulation of a symmetric advective–diffusive system. *Computer Methods in Applied Mechanics and Engineering* 1993; **105**:405–433.
19. Strujis R, Roe PL, Deconinck H. *Fluctuations Splitting Schemes for the 2D Euler Equations*. Von Karman Institute. 1991-11/AR. 1991.
20. Brandt A. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. *Proceedings of the 3rd International Conference on Numerical Methods in Fluid Mechanics*, vol. 1. Springer: Berlin 1973; **1**:82–89.

21. Wesseling P. Introduction to Multi-Grid Methods. *NASA, ICASE Report*. CR-195045. 95-11. 1995.
22. Mavriplis DJ. Multigrid techniques for unstructured meshes. *Notes* prepared for 26th Computational Fluid Dynamics, Lecture Series Program, Von Karman Institute for Fluid Dynamics, Rhode-Saint-Genèse, Belgium. 1995.
23. Mavriplis DJ. Multigrid solution of the two-dimensional Euler equations on unstructured triangular meshes. *AIAA Journal* 1988; **26**(7):824–831.
24. Vázquez M, Houzeaux G, Codina R. Chimera type domain decomposition methods applied to fractional step finite element schemes for incompressible flows. *Proceedings of the ECCOMAS 2000 Computational Fluid Dynamics Conference*, Barcelona, Spain (FIB-CIMNE, UPC, edited in CD ROM) 2000.
25. Cornelius C, Volgmann W, Stoff H. Calculation of three-dimensional turbulent flow with a finite volume multigrid method. *International Journal for Numerical Methods in Fluids* 1999; **31**:703–720.
26. Lavery N, Taylor C. Iterative and multigrid methods in the finite element solution of incompressible and turbulent fluid flow. *International Journal for Numerical Methods in Fluids* 1999; **30**:609–634.
27. Vázquez M, Codina R. Numerical Solution of the Navier–Stokes Equations using a splitting technique with multigrid acceleration. *Proceedings of the 4th World Congress on Computational Mechanics*, Buenos Aires, Argentina, 1998.
28. Dick E, Steelant J. Coupled solution of the steady compressible Navier–Stokes equations and the k–ε turbulence equations with a multigrid method. *Applied Numerical Mathematics* 1997; **23**:49–61.
29. Timmermann G. A cascadic multigrid algorithm for semilinear elliptic problems. *Numerische Mathematik* 2000; **86**:717–731.
30. Xu X, Desideri JA, Janka A. Cascadic multigrid for convection–diffusion equation *INRIA Report*, 2000.
31. Olejniczak J, Wright MJ, Candler GV. Numerical study of inviscid shock interactions on double-wedge geometries. *Journal of Fluid Mechanics* 1997; **352**:1–25.

