European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Computational Fluid Dynamics Conference 2001
Swansea, Wales, UK, 4-7 September 2001

MULTIGRID APPLIED TO A FULLY IMPLICIT FEM SOLVER FOR TURBULENT INCOMPRESSIBLE FLOWS

M. Vázquez †, M. Ravachol and M. Mallet

Dassault Aviation - Pole Scientifique 78 Quai Marcel Dassault BP 300 92152 St. Cloud CEDEX France.
† Email: mariano@cimne.upc.es

Key words: Multigrid, Incompressible Flows, Turbulence, Finite Element Method

Abstract. *A methodology for convergence speed-up of a fully implicit solver for the Random Averaged Navier-Stokes (RANS) equations for incompressible flows using multigrid (MG) techniques is here presented. The RANS set, comprising the mean flow Navier-Stokes equations and a 2-equation k-ε turbulence model, is discretized in space by applying the finite element method onto a hierarchy of meshes of different element sizes. To solve the system in the finest discretization, a non-linear multigrid scheme is applied to the hierarchy. A second objective of this work is to make more robust multigrids, trying to keep good speed-up rates. Considering this paper as a first part of a larger work, we leave aside momentarily the quest for the best possible speed-up rates, to develop some ideas that can make our multigrid more reliable, less dependent to the kind of problem we deal with or to the way the grid hierarchies are constructed: no particular grid-coarsening strategy is studied here. Among these ideas are a new residual transfer operator, boundary conditions treatment and different ways of implementing cascadic start-ups. This last point is very important because we have observed that somewhat contrary to intuition, the so called Full Multigrid scheme (a cascadic start-up stage followed by a traditional multigrid cycle) is not always the best option.*

1 INTRODUCTION

Convergence acceleration is a must when dealing with problems with high Reynolds numbers. This is particularly important in turbulent problems, when the number of degrees of freedom is dramatically increased. Finer meshes are needed to solve the broader scales range, and additionally, the flow equations set is enriched to include turbulence effects. We present here a methodology for convergence speedup of a fully implicit solver using multigrid techniques. The Random Averaged Navier-Stokes equations are discretized in space using the finite element method. This produces a non-linear system of equations, to be solved implicitly. The unknowns are the mean velocity and mean pressure and, in the cases studied, the turbulent kinetic energy k and the turbulent dissipation ϵ.

The flow equations are solved using the techniques presented in a M. Ravachol's previous work [15]. Stationary solutions are studied by linearizing the convective non-linear term and iterating until the steady state is reach. At each iteration, a preconditioned GMRES algorithm is used to solve both the mean and turbulent unknowns linearized systems. Pressure equation is stabilized by means of the GLS method [7], which when applied to the mean flow equations, allows equal interpolation spaces for both velocity and pressure. Besides, the same spaces are used for the turbulent variables k and ϵ. The numerical scheme for the turbulent part is based in that proposed by R. Strujis and co-workers in [16].

Due to the non-linearity of the target equations, non-linear multigrid (NLMG) ideas are used to tackle the convergence acceleration issue. One of the first schemes of this kind is introduced in 1977 by A. Brandt [2]: the Full Approximation Storage (FAS). Besides, the space discretization inherent to the FE method is particularly well-suited for constructing both the grids hierarchy and its transfer operators. These properties also allow deeper studies on convergence for some simple non-linear problems formulated in nested subspaces (viz. [22]). Based on these ideas, we build a NLMG scheme for solving the discretized RANS set.

A critical issue is the hierarchy construction. While in "Algebraic" MGs this is done using stencils, in "Geometric" MGs (like the one described here), a good *coarsening* procedure is needed. By coarsening it is usually understood a way of generating a coarse mesh from a given finer one keeping locally the relation coarse-to-fine. Agglomeration techniques (viz. [12]) are quite popular in the finite volume methods context. In FEM schemes, while structured grids offer no problem, the coarsening procedure can be rather complex for unstructured ones. We believe that this is a necessary point to be developed, but only when a robust multigrid is at hand. Therefore, in this paper we focus on ideas to make the NLMG proposed as robust as possible.

Following this line, among the ideas we propose, a conservative residual transfer operator is studied. It goes from fine to coarse grids, and allows the use of hierarchies constructed with lax criteria. In this way, meshes can be generated independently of each other, using the same mesh generator acting on the same geometry. Only different

generating parameters, like mean element size assigned to parts of the geometry, should be given. This operator was introduced in [19], applied to domain decomposition. For faster convergence, the operator construction is slightly modified, adapting it to multrigrid methods. In this way, this operator endows the scheme with both robustness and good speedup performance.

The paper is organized as follows. Section 2 introduces the physical problem. Section 3 described its discretization following the finite element method. The multigrid method is briefly introduced in Section 4. Section 5 presents the non-linear multigrid scheme applied to solve the system arised by the FEM discretization of the incompressible RANS set. An assessment of the proposed ideas is shown through some numerical examples in Section 6. Conclusions and future lines to follow close the article in Section 7.

2 THE PHYSICAL PROBLEM: THE RANS EQUATIONS FOR INCOMPRESSIBLE FLOWS

The physical problem under study in this work is the incompressible turbulent flow. As said before, turbulent flow can be modelled as a mean component and a fluctuation added. There are many models according to the kind of "mean" operator used and the way the fluctuation is both produced and destroyed. Models based on *Reynolds Averaged Navier-Stokes (RANS)* are among the most widespread, particularly in engineering applications. The basic concept in all of these models is that turbulent behavior is described also by the Navier-Stokes equations and that the continuum hypothesis holds. See for intance [21].

Let us picture a fluid contained in a given domain Ω and suppose density is constant in time and space. Its dynamics is then described through the vectorial field *velocity* $u_i = u_i(x_i, t)$ and the scalar field *pressure* $p = p(x_i, t)$, where $x_i \in \Omega$, and time $t \in [0, \infty)$. Here, vectors are noted as subindexed quantities, where the subindex conts the space dimension. If we suppose that these fields can be decomposed in mean component and fluctuation, i.e. $u_i = \bar{u}_i + u_i'$ and $p = \bar{p} + p'$, then if f_j is a given source term, the Reynolds Averaged Navier-Stokes equations are

$$\rho \frac{\partial \bar{u}_j}{\partial t} + \rho \bar{u}_i \frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial}{\partial x_i}(\delta_{ij}\bar{p} - \bar{\tau}_{ij} - R_{ij}) = f_j \quad (1)$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0, \quad (2)$$

where

$$\bar{\tau}_{ij} = \mu \left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i} \right), \quad (3)$$

$$R_{ij} = -\rho\overline{u_i' u_j'}, \tag{4}$$

being R_{ij} the Reynolds stress tensor. The solution of this set is $(\bar{u}_i, \bar{p})$, the mean flow field. In order to obtain it, the problem should be closed by modelling R_{ij}.

2.1 The k-ε model

The $k-\varepsilon$ model belongs to the kind known as *2-equation models*. It was introduced in early works like [6, 9]. In all of these models, R_{ij} is modelled following the *Boussinesq approximation*:

$$R_{ij} = 2\mu_{\mathrm{T}}\left(\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right) - \frac{2}{3}\rho k \delta_{ij}, \tag{5}$$

where, according to the $k-\varepsilon$ model, the *turbulent viscosity* is defined as

$$\mu_{\mathrm{T}} = C_\mu f_\mu \rho \frac{k^2}{\varepsilon}. \tag{6}$$

The quantities k and ε are the *turbulent kinetic energy* and *turbulent dissipation* respectively. Two more transport equations describe their dynamics and, added to the RANS set, close the model.

Following the usual approximations and hypotheses assumed, the RANS / k-ε set is:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{7}$$

$$\rho\frac{\partial \bar{u}_j}{\partial t} + \rho\bar{u}_i\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial}{\partial x_i}(\delta_{ij}\bar{p} - \bar{\tau}_{ij} - R_{ij}) = f_j \tag{8}$$

$$\rho\frac{\partial k}{\partial t} + \rho\frac{\partial}{\partial x_i}\left(\bar{u}_i k - (\mu + \frac{\mu_{\mathrm{T}}}{\sigma_k})\frac{\partial k}{\partial x_i}\right) = \omega_k(k,\varepsilon) \tag{9}$$

$$\rho\frac{\partial \varepsilon}{\partial t} + \rho\frac{\partial}{\partial x_i}\left(\bar{u}_i \varepsilon - (\mu + \frac{\mu_{\mathrm{T}}}{\sigma_\varepsilon})\frac{\partial \varepsilon}{\partial x_i}\right) = \omega_\varepsilon(k,\varepsilon) \tag{10}$$

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 (e.g. see [9]),

$$\omega_k = P_k - \rho\varepsilon, \tag{11}$$

$$\omega_\varepsilon = C_{\varepsilon 1}\frac{\varepsilon}{k}P_k - C_{\varepsilon 2}\frac{\rho\varepsilon^2}{k}, \tag{12}$$

$$P_k = R_{ij}\frac{\partial \tilde{u}_i}{\partial x_i}, \tag{13}$$

where f_μ is either 1.0 or a damping function which depends on the model used, and

$$C_\mu = 0.09, \qquad \sigma_k = 1.0, \qquad \sigma_\varepsilon = 1.3, \qquad C_{\varepsilon 1} = 1.44, \qquad C_{\varepsilon 2} = 1.92.$$

The first and second term in (11) are respectively called the turbulent kinetic energy *Production* and *Destruction*.

Boundary conditions

In this work we test two different $k-\varepsilon$ models: a *two-layer* one, introduced in [13] and a *law-of-the-wall* model, viz. [21].

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 behavior should be considered carefully. On 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. The *turbulent Reynolds Number* Re_{y} defines the interface between these regions. Being

$$Re_{\mathrm{y}} = \frac{\sqrt{k}y}{\mu}, \tag{14}$$

it is considered that this interface occurs at $Re_{\mathrm{y}} = 150$. Beyond this value, $f_\mu = 1.0$, and the complete RANS set is used. Below it, the equation for ε is replaced by a direct algebraic relation and μ_{T} depends only on k:

$$\varepsilon = \frac{k^{3/2}}{f_\varepsilon} \tag{15}$$

$$\mu_{\mathrm{T}} = C_\mu f_\mu \sqrt{k}. \tag{16}$$

where the damping functions are

$$f_\mu = Cy(1 - \exp(-\frac{Re_{\mathrm{y}}}{A})) \tag{17}$$

$$f_\varepsilon = Cy(1 - \exp(-\frac{Re_{\mathrm{y}}}{A_2})), \tag{18}$$

and $C = \chi C_\mu^{-3/4}$, $A = 70$, $A_2 = 2C$. The von Karman's constant is $\chi = 0.41$.

In this model, wall boundary conditions for k, ε and velocity are (n_i and g_i are the normal and tangent versors respectively)

$$\begin{aligned} k^{\mathrm{wall}} &= 0, \\ \varepsilon^{\mathrm{wall}} &= 0, \\ u_i g_i &= 0, \\ u_i n_i &= 0. \end{aligned}$$

In inflows, k, ε and $\bar{u}_i$ are prescribed to given variables, while left free in outflows.

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. There, in this off-wall boundary, 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. Its expression is

$$\frac{\bar{u}}{u_*} = \frac{1}{\chi} \log\left(y^+\right) + C, \tag{19}$$

where Δ is the approximate length of the viscous sub layer, or the distance from the "numerical" wall to the physical one, $\chi = 0.41$ is again the von Karman's constant and C is around 5.5. Also

$$y^+ = \frac{u_* \Delta}{\nu}, \tag{20}$$

where Δ is the turbulent length scale, which depends on the problem. This law is valid for $30 < y^+ < 100$. An alternative is an extended range law of the wall, like the *Reichard law*, which includes also the buffer zone and the log layer, cited in [21]. This is used specially in problems with low Re zones (for intance, near stagnation points)

$$\frac{\bar{u}}{u_*} = \frac{1}{\chi} \ln(1 + 0.4y^+) + 7.8 \left[1 - \exp\left(\frac{-y^+}{11} \right) - \frac{-y^+}{11} \exp(-0.33y^+) \right]. \tag{21}$$

Assuming the hypothesis above, a tangential traction $t^{\text{wall}} = n_i \sigma_{ij} g_j$ can be imposed. It depends on the friction velocity u_* evaluated from (19).

Then, boundary conditions for k, ε and velocity in the "wall" (now slightly off the real solid wall) are

$$\begin{aligned} k^{\text{wall}} &= \frac{u_*^2}{\sqrt{C_\mu}} \\ \varepsilon^{\text{wall}} &= \frac{u_*^3}{\chi \Delta} \\ n_i \sigma_{ij} g_j &= -\bar{\rho} u_*^2 \\ u_i n_i &= 0 \end{aligned}$$

In inflows and outflows, again turbulent variables and velocities are given.

3 FINITE ELEMENT METHOD DISCRETIZATION OF THE RANS SET

The RANS set is discretized in space using the finite element method, as carefully described in [15]. In brief, to allow equal interpolation spaces, the classical SUPG method, introduced in [3], is used for the Navier-Stokes set. The convective term is linearized using the Picard method. At each iteration, the linearized system is solved using a

preconditioned GMRES algorithm. On the other hand, the turbulence equations are again discretized in space following the finite element method, but with the scheme introduced in [16], linearizing the source terms in (11) and (12) according to

$$\omega_k = P_k^n - \rho \frac{\varepsilon^n}{k^n} k^{n+1}, \tag{22}$$

$$\omega_\varepsilon = C_{\varepsilon 1} \frac{\varepsilon^n}{k^n} P_k^n - C_{\varepsilon 2} \frac{\rho \varepsilon^n}{k} \varepsilon^{n+1} \tag{23}$$

and solving the linearzided resulting system also with the GMRES method.

The discretized problem can be written as: with the appropriate boundary conditions, find $\boldsymbol{x} = (\bar{u}_i, \bar{p})$ and $\boldsymbol{y} = (k, \varepsilon)$, solution of

$$\left\{ \begin{array}{lll} \boldsymbol{A}(\boldsymbol{x}, \boldsymbol{\mu_{\mathrm{T}}}(\boldsymbol{y})) \quad \boldsymbol{x} & = & \boldsymbol{b} \\ \boldsymbol{C}(\boldsymbol{y}, \boldsymbol{x}) \quad \boldsymbol{y} & = & \boldsymbol{f} \end{array} \right. \tag{24}$$

where the coupling through $\boldsymbol{\mu_{\mathrm{T}}} = \boldsymbol{\mu_{\mathrm{T}}}(\boldsymbol{y})$ is explicitly mentioned. The solving strategy adopted is a "staggered" smoother: at each Navier - Stokes iteration, one $k-\varepsilon$ iteration is done.

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

Do m times
1. Solve $\boldsymbol{A}^n\ \boldsymbol{x}^{n+1} = \boldsymbol{b}$, where $\boldsymbol{A}^n = \boldsymbol{A}(\boldsymbol{x}^n, \mu_{\mathrm{T}}^n)$
2. Solve $\boldsymbol{C}^{n+1/2}\ \boldsymbol{y}^{i+1} = \boldsymbol{f}$, where $\boldsymbol{C}^{n+1/2} = \boldsymbol{C}(\boldsymbol{y}^n, \boldsymbol{x}^{n+1})$
3. Update $\boldsymbol{x}^n = \boldsymbol{x}^{n+1}$
4. Update $\boldsymbol{y}^n = \boldsymbol{y}^p$
5. Update $\mu_{\mathrm{T}}^n = \mu_{\mathrm{T}}(\boldsymbol{y}^p)$ and go back to 1.

We have observed that the use of a "nested" smoother, where some $k-\varepsilon$ 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 a relaxation factor of 0.5.

4 MULTIGRID. BASIC CONCEPTS

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 [1, 2]. 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 smooth the error, the solution can be well approximated in the next coarser mesh. There, some more iterative steps are performed, the error is smoothed again and the solution is 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 [20] and [12], complete overviews on the subject can be studied.

Loosely speaking, MG 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. [11]). 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 discretization of the RANS set lead us to the following problem: find $\boldsymbol{x}$, which is the solution of

$$\boldsymbol{A}(\boldsymbol{x})\ \boldsymbol{x} = \boldsymbol{b} \tag{25}$$

being $\boldsymbol{A}(\boldsymbol{x})$ the non-linear system matrix and $\boldsymbol{b}$ a source term. To solve it, a given iterative procedure is proposed, consisting basically of a series of linearization steps and at each step i a certain $\boldsymbol{x}^i$ is obtained. At a given step n, the outcome of the iterations is

$$\boldsymbol{x}^n = \Phi(n, \boldsymbol{x}^0, \boldsymbol{b}) = \Phi^n(\boldsymbol{x}^0, \boldsymbol{b}), \tag{26}$$

where Φ is a non-linear smoother (i.e. iterative solution procedure), which does n steps starting from $\boldsymbol{x}^0$.

Re-writing (25),

$$\begin{aligned} \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^0\boldsymbol{x}^0 &= \boldsymbol{b} - \boldsymbol{A}^0\boldsymbol{x}^0 \\ &:= \boldsymbol{r}^0, \end{aligned} \tag{27}$$

where $\boldsymbol{A}^0 = \boldsymbol{A}(\boldsymbol{x}^0)$, gives a good insight of the the NLMG principle. To solve the system on $\boldsymbol{x}$, instead of on $\boldsymbol{\Delta x}$ is central in the non - linear case. The idea is to apply Φ to get $\boldsymbol{x}^n$ in the *finest* element of the hierarchy and then to tranfer it, together with $\boldsymbol{r}^n$, to the next *coarser* element, applying there Φ m times to get $\boldsymbol{x}^{n+m}$. Finally, this is transfered back as a correction $\Delta\boldsymbol{x}$ to the first, finest mesh. The scheme in Fig. 1 shows a 2-grid NLMG cycle, easily extensible to n-grid ones. We call *smoothing stage* the "downwards" stage, namely, when variables and residuals are transferred from finer to coarser grids. The *coarse grid correction stage* is the "upwards" stage: corrections are transferred back to finer grids from coarser ones. Downwards and upwards transfer operators are noted as F and B, we come back to them below.

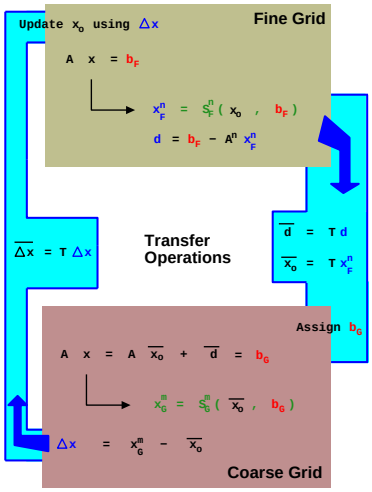

Figure 1: Multigrid's basic scheme.

To build both the spaces and the operators relating them, two points are to be considered:

- *The hierarchy of systems.* The original continuum problem is discretized in a series of grids, having different (mean) cell sizes, thus allowing an ordering according to the sizes.

- *The transfer operators.* (F and B in Fig. 1) Data transfer between two given elements of the hierarchy $\mathcal{H}$ is done through two kinds of basic operators: a *restriction* r, which does it from fine to coarse grid and a *prolongation* p , which does the opposite. Transferring between the elements is done both ways, through the preceding operators and their adjoints r* and p*, all of them operating over discrete functions defined on the partitions. For that reason, F and B are not necessarily r and p respectively, but derived from them. Besides, its construction also depends on what is transferred: residuals, variables or coarse grid corrections.

Let us briefly explain how the operators work. Through restriction, "information" is lost in the sense that not all the fine grid elements contain one node of the coarse grid. When unknowns are transferred from fine to coarse, this is not relevant: we only need a starting value which follows as close as possible the fine grid. On the other hand, residuals are transferred in a different way, taking profit of the information carried in the prolongation operator [20, 14, 11] using its adjoint p^*.

We consider several definitions of the adjoint prolongation operator, all of them functions of the transpose p^{t}. The first one, and more used, is to take directly

$$\mathrm{p}^*_{\mathrm{tra}} = \mathrm{p}^{\mathrm{t}}. \tag{28}$$

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 $\mathrm{p}^*_{\mathrm{tra}}$ is used to transfer the residuals to coarser grids the MG has proven to be less robust because the hierarchy needs to be constructed keeping $A_{l-1}/A_l < L$ *everywhere* (L given, depends on the problem. A reasonable value is 10.0). If this is not accomplished, the convergence and solution of the problem is not guaranteed.

In [19] and in the context of Chymera-type domain decomposition methods, the authors propose an operator that can be used in MG as p^*. It is formed by p^{t}, but column-wise normalized:

$$\mathrm{p}^*_{\mathrm{cwn}} = [\quad \mathrm{p} \quad \mathrm{diag}\,(1/\alpha_1\ ,\ 1/\alpha_2\ ,\ \ldots\ ,\ 1/\alpha_{N_{\mathrm{C}}}) \quad]^t, \tag{29}$$

where N_{F} and N_{C} are the number of nodes of the fine and coarse grids respectively and

$$\alpha_i = \sum_{i=1}^{N_{\mathrm{F}}} \mathrm{p}_{i,1} \tag{30}$$

is the norm of each column i of p. According to the definition of p, each of these 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 $\mathrm{p}^*_{\mathrm{cwn}}$ yields more "coarsening indepent" strategies than when $\mathrm{p}^*_{\mathrm{tra}}$ is used, i.e. more robust.

However, if for a given hierarchy the algorithm using $\mathrm{p}^*_{\mathrm{tra}}$ converges, then the use of $\mathrm{p}^*_{\mathrm{cwn}}$ in the same case will yield worse convergence rates. In order to attain the same speed-up with both operators, we propose a third one, $\mathrm{p}^*_{\mathrm{sel}}$:

$$\mathrm{p}^*_{\mathrm{sel}} = [\quad \mathrm{p} \quad \mathrm{diag}\,(1/\beta_1\ ,\ 1/\beta_2\ ,\ \ldots\ ,\ 1/\beta_{N_\mathrm{C}}) \quad]^t\,. \tag{31}$$

where

$$\beta_i = \begin{cases} \alpha_i/L_\mathrm{c} & \text{if } \alpha_i > L_\mathrm{c} \\ 1 & \text{otherwise.} \end{cases}$$

In the case of $\mathrm{p}^*_{\mathrm{sel}}$, normalization is not done for all the columns, but selectively. L_c is a threshold (say, 10): where the ratio is smaller than that, don't normalize, otherwise, keep normalization factor just under the threshold.

Briefly, at each stage, variables $\boldsymbol{x}$, residuals $\boldsymbol{d}$ and coarse grid corrections $\boldsymbol{\Delta x}$ are transferred according to the following:

- Smoothing Stage, after n steps:

$$\begin{array}{rcl} \boldsymbol{d}_\mathrm{C} & = & \mathrm{p}^*\ \boldsymbol{d}_\mathrm{F}, \quad \text{where} \quad \boldsymbol{d}_\mathrm{F} = \boldsymbol{b}_\mathrm{F} - \boldsymbol{A}^n_\mathrm{F}\boldsymbol{x}^n_\mathrm{F} \\ \boldsymbol{x}^0_\mathrm{C} & = & \mathrm{r}\ \boldsymbol{x}^n_\mathrm{F} \end{array} \tag{32}$$

- Coarse Grid Correction Stage, after m steps:

$$\boldsymbol{\Delta x}_\mathrm{F} = \mathrm{p}\ \boldsymbol{\Delta x}_\mathrm{C}, \quad \text{where} \quad \boldsymbol{\Delta x}_\mathrm{C} = \boldsymbol{x}^m_\mathrm{C} - \boldsymbol{x}^0_\mathrm{C} \tag{33}$$

where p^* is chosen out of $\mathrm{p}^*_{\mathrm{tra}}$, $\mathrm{p}^*_{\mathrm{cwn}}$ and $\mathrm{p}^*_{\mathrm{sel}}$.

In general, MG processes follow different cycling strategies, *V-cycle* or *W-cycle* (Fig. 2). *V-* means that coarse grid correction and smoothing stages are done in a straight way, from l to 1 and from 1 to l meshes respectively. *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.

A startup MG phase can be of great importance. Sometimes, a coarse mesh left alone can produce a good initial condition for a finer one, when transferred up. In this way, the whole MG process starts in the coarsest l discretization, where a certain number of steps are done. Then, the unknowns just obtained are prolonged to the second coarsest $l-1$ mesh. After a given number of MG cycles between l and $l-1$, the result is prolonged up to $l-2$, where MG cycles between l, $l-1$ and $l-2$ are done. This is process is done until reaching the finest mesh, where a V- or W-cycle MG begins until final convergence is reached. This strategy is usually known as *Full MG* or *F-cycle* [12].

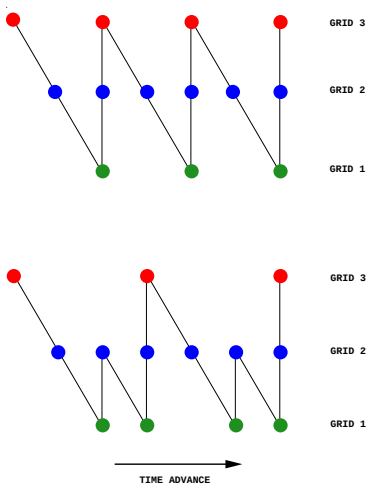

Figure 2: V and W-cycling.

5 NON-LINEAR MULTIGRID APPLIED TO THE DISCRETIZED RANS SET

As was said before, the discretization in space of the Reynolds Averaged Navier - Stokes set by the Finite Element Method 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 the its interpolation functions. On the other hand, the problem's non-linearity leads to a MG method that can cope with it: a NLMG scheme.

To apply multigrid to accelerate the convergence rate of solvers for non-linear systems is not a new idea. In [20], the FAS (full approximation storage) algorithm of Brandt [2] is cited as the first NLMG to appear. The present work follows the same line. Basically, the smoother Φ is non-linear and the system is not solved on increments of the unknown, but on the unknown itself. The MG cycling is embedded in the whole solver, from *outside* the linearization cycle. Recent works using NLMG techniques for solving these problems using the same kind of smoother are those of [4] or [10]. Of course, it can be different, like in [18], where a fractional step procedure, the CBS method (previously proposed in [24], [25] and so on) leads to explicit or semi-implicit iterative smoothers Φ. A completely different alternative strategy would be to use linear multigrids to speed-up only the solution process of the interior linear system.

The NLMG process described in Section 4 and sketched in Fig. 1 is applied to the problem (24). Now the smoother is either Φ_{NS} or Φ_{SS} and $\boldsymbol{x}$, $\boldsymbol{b}$, and matrix $\boldsymbol{A}$ are replaced by $(\boldsymbol{x}, \boldsymbol{y})$, $(\boldsymbol{b}, \boldsymbol{f})$ and a matrix formed by two blocks with $(\boldsymbol{A}, \boldsymbol{C})$ respectively. In this way,

multigrid is acting on the *whole* problem. We call this the the *complete approach*. On the other hand, the NLMG ideas can be applied *only* to the laminar problem, leaving aside the turbulent part. This will be the *laminar approach*. In this case, the complete problem is solved only in the finest discretization: while the laminar part is solved in the whole hierarchy following the multigrid scheme, the turbulent part is only solved in the finest grid and the laminar unknowns, their residuals and the turbulent viscosity transferred to the rest of the hierarchy. In doing this work we have compared both strategies, observing the superior performance of the *complete approach*, namely when multigrid's tranferring of unknowns, residuals and corrections is applied to the whole problem, laminar and turbulent. Therefore, following this line, the NLMG under the complete approach can be summarized as follows.

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

$$\begin{aligned}
\boldsymbol{d}_{\text{lam,C}} &= \text{p}^* \, \boldsymbol{d}_{\text{lam,F}}, \quad \text{where} \quad \boldsymbol{d}_{\text{lam,F}} = \boldsymbol{b}_{\text{F}} - \boldsymbol{A}_{\text{F}}^n \boldsymbol{x}_{\text{F}}^n \\
\boldsymbol{d}_{\text{tur,C}} &= \text{p}^* \, \boldsymbol{d}_{\text{tur,F}}, \quad \text{where} \quad \boldsymbol{d}_{\text{tur,F}} = \boldsymbol{f}_{\text{F}} - \boldsymbol{C}_{\text{F}}^n \boldsymbol{y}_{\text{F}}^n \\
\boldsymbol{x}_{\text{C}}^0 &= \text{r} \, \boldsymbol{x}_{\text{F}}^n \\
\boldsymbol{y}_{\text{C}}^0 &= \text{r} \, \boldsymbol{y}_{\text{F}}^n \\
\boldsymbol{\mu}_{\mathbf{T}\text{C}} &= \text{r} \, \boldsymbol{\mu}_{\mathbf{T}_{\text{F}}}^n
\end{aligned} \tag{34}$$

- **Coarse Grid Correction Stage.** After m steps applied with the chosen smoother, the laminar and turbulent coarse grid corrections $\boldsymbol{\Delta x}_{\text{C}}$ and $\boldsymbol{\Delta y}_{\text{C}}$ are tranferred back from the *coarse* grid to the *fine* one according to

$$\begin{aligned}
\boldsymbol{\Delta x}_{\text{F}} &= \text{p} \, \boldsymbol{\Delta x}_{\text{C}}, \quad \text{where} \quad \boldsymbol{\Delta x}_{\text{C}} = \boldsymbol{x}_{\text{C}}^m - \boldsymbol{x}_{\text{C}}^0 \\
\boldsymbol{\Delta y}_{\text{F}} &= \text{p} \, \boldsymbol{\Delta y}_{\text{C}}, \quad \text{where} \quad \boldsymbol{\Delta y}_{\text{C}} = \boldsymbol{y}_{\text{C}}^m - \boldsymbol{y}_{\text{C}}^0
\end{aligned} \tag{35}$$

This is our default scheme. However, some problems with large turbulent production and dissipation (like in the third numerical example below) can present difficulties. This terms are highly non-linear and can produce large numerical oscillations in the coarser grids. To avoid this kind of problems, some solutions where studied, which are proposed below as implementation issues.

5.1 Implementation issues

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. This is particularly important in the case of curved boundaries. For instance, when a no 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.

- **Transfer operators construction.** As said before, we are looking for a robust MG 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 operators here studied are very easy to build, and all together with TRA they range from a "conservative" strategy (CWN) to an "aggressive" one (TRA), with SEL operator between them, which in turns can be tunned by the choice of L_{c}.

- **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 any of these operators. On the other hand, coarse grid relaxation is indeed very useful, specially when high gradients are present. It is implemented as a coarse grid correction reduction by a factor $0.0 \leq \theta_{\mathrm{CGC}} \leq 1.0$:

$$\boldsymbol{x}_{\mathrm{F}}^{n+1} = \boldsymbol{x}_{\mathrm{F}}^{n} + \theta_{\mathrm{CGC}}^{n} \boldsymbol{\Delta x}_{\mathrm{F}}. \tag{36}$$

  A reasonable value is $\theta_{\mathrm{CGC}} = 0.5$.

- **V/W-cyclings, Full MG and Cascadic MG.** As said at the end of Section 4, all along this work, by V- or W-cycle multigrid we understand the classical cycling (viz. [20]), which starts the whole solution process in the upper grid. And by F-cycle, full multigrid (viz. [12]), which on the other hand 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 [17] or [23]). 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 (Fig. 3). 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. Moreover, the upwards stage can be done with some sort of cycling (V- or W-) (Fig. 4).

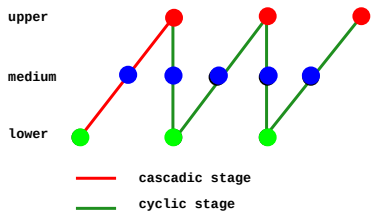

Figure 3: Cascadic and cyclic stages.

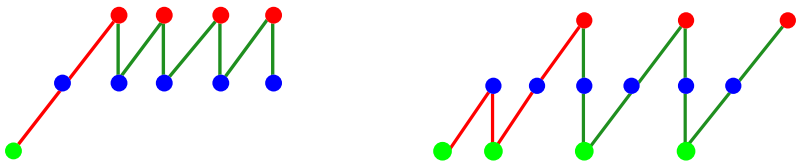

Figure 4: Discarding lower grid after cascadic stage (left) and cascadic stage with cycling.

All these possibilities open a wide range of strategies. In order to clearly set the basis of the algorithm, we concentrate only in those most representative. Firstly, pure V-cycle. Then, pure F-cycle (as in Fig. 3). And finally, discarding lower grid after the cascadic stage (left of Fig. 4). We have observed that opposing to what can be expected, F-cycle is *eventually*, not always, the best option. The cascadic sequencing aims to provide a better initial condition for the upper grids. But being the problem (highly) non-linear, the coarse grids could be too coarse to provide a

good one. This is what happen to the "Step-function-inflow tube" and to the "Three element airfoil" examples shown here. In the former case, no cascadic stage is the best option. In the latter one, a slight change in the boundary condition (slipping condition instead of law-of-the-wall one) of the lower grid provides its convergence, guaranteeing a good performance of the F-cycle.

- **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.

6 NUMERICAL EXAMPLES

6.1 Boundary layer

A very simple example, a turbulent boundary layer, is the first one shown. Although its simplicity, it provides a good insight of the method proposed here. The numerical domain is shown in Fig. 5. The inflows (top boundary included) prescriptions are:

$$\begin{aligned} \boldsymbol{u}_{inf} &= (1,0) \\ k_{inf} &= \beta u_{inf}^2 \\ \varepsilon_{inf} &= k_{inf}^{3/2}/\Delta. \end{aligned} \tag{37}$$

In this problem, the turbulent length scale is $\Delta = 0.05$ and the input turbulent kinetic energy factor is $\beta = 0.0001$. The law of the wall is applied in the bottom boundary to impose a tangent traction. It is calculated from to the Reichard extended law (21), yielding to the boundary conditions defined in (22). Both velocity and turbulent variables are left free in the outflow. The Re per unit length is 10^3.

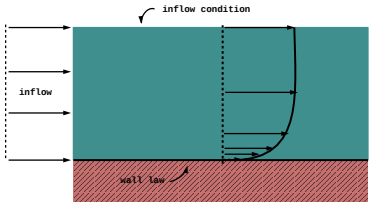

Figure 5: Turbulent boundary layer. Problem and boundary conditions.

The domain is discretized with a structured grid, refined towards the boundary layer and uniformly spaced horizontally. It has 2145 nodes and 4096 P1 elements (64x32). The constant velocity inflow prescription produces a strong velocity gradient at the left bottom corner, which propagates downstream.

The hierarchy is nested, constructed by multiplying 2 or 4 times either the horizontal spacing or the vertical spacing or both. The test is done with 2-grid sets: always the same upper (fine) grid plus a lower (coarse) one, selected out of several options. We use the following notation: "$i\mathbf{x}j\mathbf{y}$" means a 2-grid multigrid scheme using as lower grid a coarse one formed by multiplying i times horizontal spacing and j times vertical spacing of the finest one. TRA, CWN, SEL and SG mean transpose, column-wise normalized transpose and selective column-wise normalized transpose projection adjoint transfer operators used, and single-grid, respectively. No $P + D$ freezing is needed in this example.

Convergence results are shown in Fig. 6 and Fig. 7. In Fig. 6 (left) convergence curves range from the SG's to "TRA 2x2y"'s, with a 3.5 global speed-up, i.e. the ratio of total CPU time the residual needs to reach say 10^{-6} in both cases. The curves between represent other different combinations. Out of them, it is remarkable the "CWN 1x4y" performance. It gives a reasonable 2.5 speed-up allowing the use of a very coarse lower grid, which blows when used in the plain "TRA 1x4y" (of course, not shown in the graph: it blows).

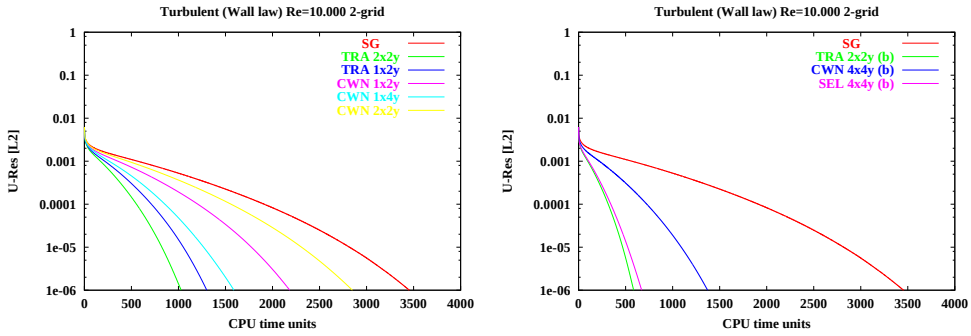

Figure 6: Turbulent boundary layer. Convergence graphs (1) and (2).

To check the real performance of the methods proposed, we ran both upper and lower problems using exactly the same set of numerical parameters, like CFL number, Krylov GMRES space dimension, GMRES iterations and toleration, precondition strategy, and so on. However, several studies on convergence of multilevel methods applied to mildly or semilinear methods (viz. [17] or [8]) suggests that providing the solution of the linear problem for a given linearization step exists, the transferred coarse grid correction (in the case of multigrid) or the extrapolated solution (in simple multilevel methods) is smooth enough to allow the use of very lax numerical parameters sets in the fine grid. For instance, low Krylov GMRES dimension in the upper grid can reduce considerably the

total computing time. And indeed it could be so low than it cannot be used to compute the SG problem for it diverges. For this case we could go down to dimension 3. This fact is shown in Fig. 6 (right) and Fig. 7. By "(b)" we note low Krilov dimension (only in the upper grid) problems: we consider dimension 5, while in the lower grid it is kept dimension 15, like in the SG problem. In Fig. 7 the SG performance, "TRA 2x2y" and "CWN 1x4y" are shown again to be compared with "TRA 2x2y(b)" and "CWN 1x4y(b)".

Selective column-wise transpose (SEL) performance is compared to CWN and TRA in Fig. 6 (right). In this graph SG convergence is again plotted as a reference. While "CWN 4x4y(b)" speeds up the convergence by 2.5 (like, "CWN 1x4y"), "SEL 4x4y(b)" gives almost the same convergence improvement than "TRA 2x2y(b)", around 7.

From the first example, we can conclude the following. When compared to TRA, the use of CWN operator produces a more robust scheme, less dependent on coarsening strategies: in this particular case, 4x4y means that the lower grid is locally 16 times coarser. However, its performance is not so impressive as TRA. But SEL scheme improves the CWN speed-up rates up to TRA level, keeping its robustness properties. In this case, the threshold in (4) is $L_c = 3$.

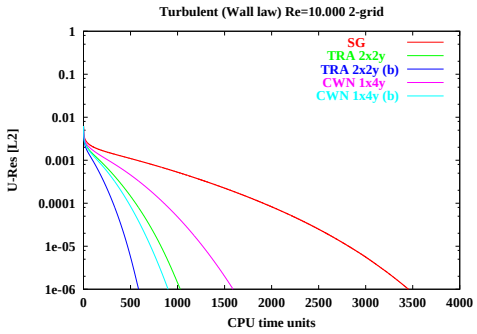

Figure 7: Turbulent boundary layer. Convergence graphs (3).

6.2 Step-function-inflow tube

In this example, the flow structure is more complex. The computational domain is a rectangle with height:length ratio of 2 : 30 (Fig. 8). 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

$$\begin{aligned} \boldsymbol{u}_{inf} &= (1,0) \\ k_{inf} &= \beta u_{inf}^2 \\ \mu_{T\,inf} &= \alpha\mu \\ \varepsilon_{inf} &= C_\mu k_{inf}^2/\mu_{T\,inf}. \end{aligned} \tag{38}$$

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 [13] is used in this example. Therefore, u_i, k and ε are set to zero at the non-slipping boundaries. The domain is discretized with a mixed structured and non-structured grid. The upper grid is made of 13221 nodes and 25894 P1 elements. Pressure, velocity and turbulent kinetic energy are shown in Fig. 9.

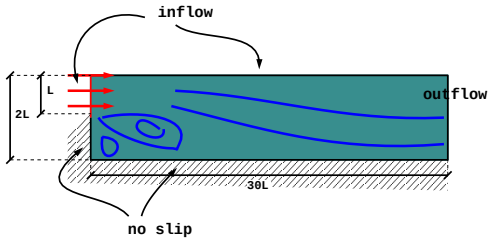

Figure 8: Step-function-inflow tube. Problem and boundary conditions.

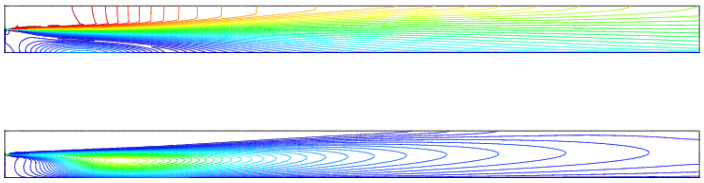

Figure 9: Step-function-inflow tube. Velocity and turbulent kinetic energy contours.

In Fig. 10 the convergence for single and multigrid is shown. The hierarchy comprises 3 grids constructed by doubling the local element lengths of the prececent grid. This doubling is done approximately, except in the structured and homogeneous zones. The strategy used in this case is a V-cycle, with source-freezing for the turbulent variables, SEL operator with $L_{\mathrm{c}} = 10$ and CGC relaxation with $\theta_{\mathrm{CGC}} = 0.5$. The same GMRES parameters are used in all the cases. One smoothing iteration is done in each grid, with no post-smoothing. FMG was discarded here because the lower grid is too coarse to yield a steady solution.

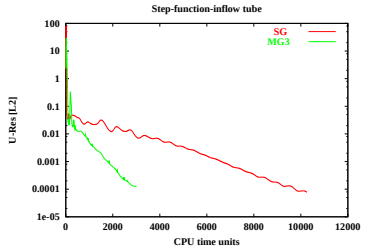

Figure 10: Step-function-inflow tube. Convergence graph.

In this example, a 3-grid hierarchy produces a speed-up of around 5. As we focus in the robustness of the algorithm, our objective was not to push the multigrid performance to the limit, but to construct the hierarchy as fast and simple as possible. As said before, we do not apply any coarsening algorithm but use the same input geometry in the mesh generator, only changing the generating local parameters.

6.3 Three-element airfoil

This is a classical example of complex 2D flow. This high lift device consists of an airfoil with a frontal slat and a flap (Fig. 11). A large vortex is produced under the slat, wich interacts with the boundary layer in the airfoil. A smaller elongated vortex is also in the camber of the airfoil. In the gaps between the elements, the flow accelerates and a long wake appears downstream the device.

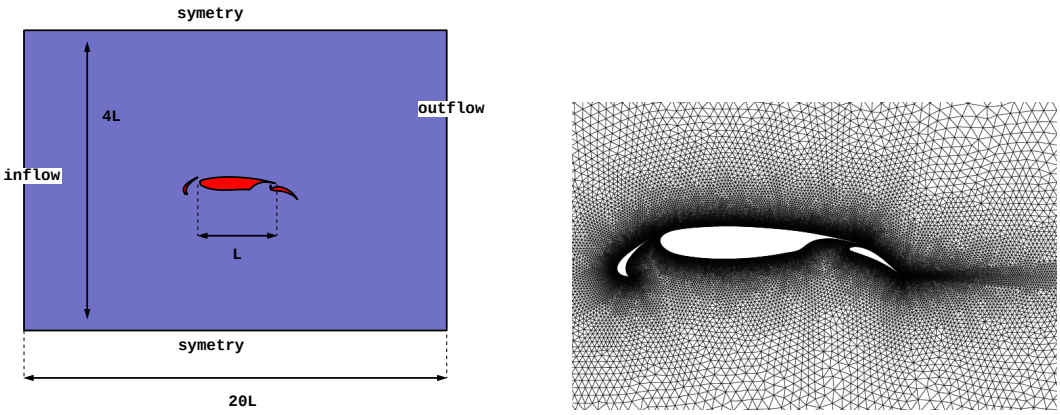

Figure 11: Three element airfoil. Problem set and upper grid detail.

In this case, the attack angle is 0°, resulting in a take-off configuration. The Reynolds number is computed using L, which is the length of the main component (Fig. 11). It is $Re = 3.6 \times 10^6$. The upper hierarchy grid consists of 38856 nodes and 75887 P1 elements.

The Reichard's law of the wall (21) is used on the foils, being (22) the boundary conditions there. Therefore, no elongated elements are present, as opposed to the previous example (Fig. 11). A slipping boundary condition is in the top and bottom boundaries. The inflow conditions are the same as in (38), with $\Delta = 3.6 \times 10^{-4}$, $\beta = 0.001$ and the input turbulent viscosity factor $\alpha = 360$. Pressure and turbulent viscosity are shown in Fig. 12.

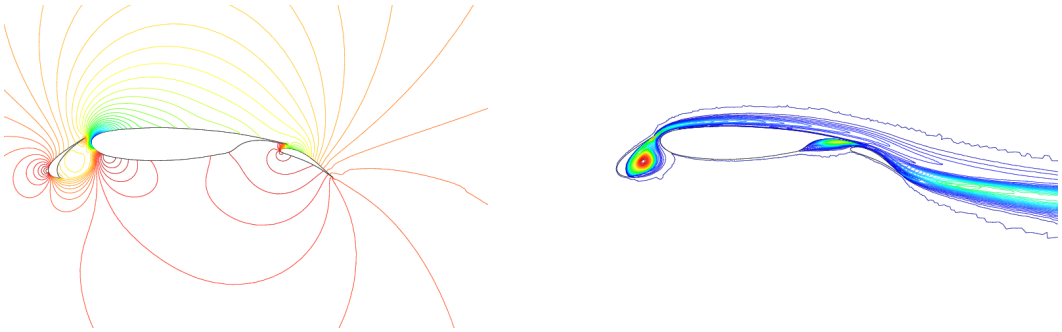

Figure 12: Three element airfoil. Pressure (left) and turbulent viscosity contours.

Following the ideas described above, the multigrid hierarchy is again constructed somewhat loosely. By simply doubling the local element size and increasing steepness factor of the mesh generator, two coarser meshes are constructed independently. Therefore while the resulting speed-up is not so large (around 2), the hierarchy construction is very simple. Assuming that the most significative part of the flow is around the device, to assess the multigrid performance, its C_{L} convergence is used, which we define as

$$DCL^n = \frac{\sqrt{\left(C_{\text{L}}^{n+1} - C_{\text{L}}^{n}\right)^2}}{C_{\text{L}}^{n+1}}, \tag{39}$$

it is plotted in Fig. 13. The pressure lift coefficient is defined as

$$C_{\text{L}} = \frac{F_y}{\frac{1}{2}\rho U_\infty^2} \tag{40}$$

where F_y is the vertical component of the pressure force exerted on the complete device, computed as the boundary integral of the pressure on the airfoils. The converged value thus obtained is $C_{\text{L}} = 2.44$.

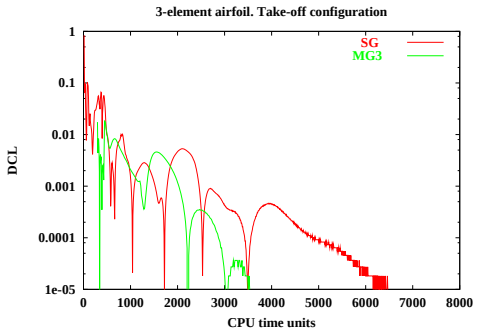

Figure 13: Three element airfoil. DCL convergence graphs.

In this case, we observe that the F-cycle is the best option. Due to the fact that the coarse grid is too coarse, to achieve a good initial condition to be transferred to the medium grid the wall boundary condition must be modified. Instead of imposing a tangent traction obtained from the law of the wall, a slipping condition, (i.e. freeing tangent traction) does it very well. The strategy used is that of Fig. 4, left: after doing some iterations in the lower and medium grids being the upper one reached, a multigrid V-cycle starts discarding the lower one. In the V-cycle, one smoothing iteration is done in each grid, with no post-smoothing. The convergence rate curve is labelled as "MG3" in Fig. 13. No freezing of the turbulent sources was needed and the SEL operator is used with $L_c = 5$. The same GMRES parameters are used in all the cases.

7 Conclusions. Future Lines

In this paper we have presented a non-linear multigrid scheme for solving a FEM algorithm for the Reynolds Averaged Navier-Stokes set of equations, using a k-ε turbulence model. Also, several ideas for improving the robustness of the multigrid scheme are proposed. This work has been conducted having in mind one central idea: a robust multigrid. From the beginning we have noticed the importance of this point, which has not a unique pair cause and cure. The lack of reliability is due to several factors and many solutions can be proposed to improve the performance. Additionally, these factors and solutions are not exclusive to a given algorithm, in particular this one, but to *all* kind of multigrids. On these grounds, almost all of the ideas proposed here, concerning this weakness (like operator construction, cascadic start-up stages, boundary condition treatment, corrections relaxation and so on), can be applied to many multigrid schemes.

The NLMG proposed, together with the other ideas, is tested through three 2-D examples. The first one is very simple and introduces the scheme in its many different flavors. The second and third ones show more complex flow behaviour. In the step-function-inflow tube, a down-to-the-wall integration model is used, resulting in an upper grid with highly elongated elements. In this kind of problems, multigrid should perform really well, taking

into account that a cascadic start-up stage is a very bad option here. On the other hand, in the last example, the use of a cascadic start-up leads to a better speed-up, providing a minor, although decisive, change in the wall boundary conditions.

We believe that the speed-up rates for these kind of problems can be surely improved. We need to remark that the problems are not too large (around 25.000 and 75.000 elements respectively) and the hierarchies are small (3 grids at most, being 2 for the multigrid cycling in the airfoils). On the other hand, the grids are constructed independently of each other, simply by changing the parameters of the mesh generator program (which, in turn, is not a good one). Hitherto our work was to set the NLMG scheme for the target problem, to identify sources of convergence problems and to propose solutions to tackle them.

From now on, once that what we propose is tested, we can follow different lines. First, we move to compressible RANS problems. Second, to attain larger speed-ups, the hierarchies should be constructed by coarsening. Additionally, this will lead to a more automatic process. And last but not least, at the moment we are writting this paper, we are carrying out some tests on an original multigrid construction which performs the cycling in disjoint domains, placing the lower grids in the regions of the original domain where the largest error sources are, like *patches*. The first results are encouraging us to continue this line.

REFERENCES

[1] A. Brandt. Multi-level adaptive technique (MLAT) for fast numerical solution to boundary value problems. In H. Cabannes and R. Temam, editors, *Proc. III Int. Conf. on Numerical Methods in Fluid Mechanics*, volume 1, pages 82–89. Springer - Verlag, 1973.

[2] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comput.*, 31:333–390, 1977.

[3] A.N. Brooks and T.J.R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equation. *Comp. Meth. Appl. Mech. Eng.*, 32:199–259, 1982.

[4] C. Cornelius, W. Volgmann, and H. Stoff. Calculation of three-dimensional turbulent flow with a finite volume multigrid method. 31:703–720, 1999.

[5] W. Hackbusch. *Iterative Solution of Large Sparse System of Equations*. Springer - Verlag, 1994.

[6] F.H. Harlow and P. Nakayama. Transport of turbulence energy decay rate. University California Report LA-3854, Los Alamos Science Lab., 1968.

[7] T.J.R. Hughes, L.P. Franca, and G.M. Hulbert. A New Finite Element Method for Computational Fluid Dynamics: VIII. The Galerkin/Least-Squares Method for Multidimensional Advective-Diffusive Equations. *Comp. Meth. Appl. Mech. Eng.*, 73:173–189, 1989.

[8] M.G. Larson and A.J. Niklasson. Adaptive multilevel finite element approximations of semilinear elliptic boundary value problems. *Numerische Mathematik*, 84:249–274, 1999.

[9] B.E. Launder and D.B. Spalding. The numerical computation of turbulent flows. *Comp. Meth. Appl. Mech. Eng.*, 3:269–289, 1974.

[10] N. Lavery and C. Taylor. Iterative and multigrid methods in the finite element solution of incompressible and turbulent fluid flow. *Int. J. Num. Meth. Fluids.*, 30:609–634, 1999.

[11] D.J. Mavriplis. Multigrid solution of the two - dimensional Euler equations on unstructured triangular meshes. *AIAA Journal*, 26(7):824–831, 1988.

[12] D.J. Mavriplis. Multigrid techniques for unstructured meshes, March 1995.

[13] V.C. Patel and H.C. Chen. Near-wall turbulence models for complex flows including separation. *AIAA Journal*, 29(6), 1990.

[14] J. Peraire, J. Peiro, and K. Morgan. Multigrid solution of the 3-D compressible Euler equations on unstructured tetrahedral grids. *Int. J. Num. Meth. Eng.*, 36:1029–1044, 1993.

[15] M. Ravachol. Unstructured finite elements for incompressible flows. Number AIAA 97-1864, pages 67–75, Snowmass Village, CO, USA, June 1997.

[16] R. Strujis, P.L. Roe, and H. Deconinck. Fluctuations splitting schemes for the 2D Euler equations. Technical Report 1991-11/AR, Von Karman Institute, 1991.

[17] G. Timmermann. A cascadic multigrid algorithm for semilinear elliptic problems. *Numerische Mathematik*, 86:717–731, 2000.

[18] M. Vázquez and R. Codina. Numerical solution of the Navier - Stokes equations using a splitting technique with Multigrid acceleration. In *Proc. 4th World Congress on Computational Mechanics, Buenos Aires, Argentina*, volume Part 2, page 663. International Association for Computational Mechanics, 1998.

[19] M. Vázquez, G. Houzeaux, and R. Codina. Chimera type domain decomposition methods applied to fractional step finite element schemes for incompressible flows. In *Proceedings of the ECCOMAS 2000 Computational Fluid Dynamics Conference*, Barcelona, Spain, 2000. FIB-CIMNE, UPC (edited in CD ROM).

[20] P. Wesseling. Introduction to Multi - Grid methods. CR - 195045 ICASE 95 - 11, NASA, 1995.

[21] D.C. Wilcox. *Turbulence Modeling for CFD*. DCW Industries, 1993.

[22] Jinchao Xu. Two-grid discretization techniques for linear and nonlinear PDEs. *SIAM J. Numer. Anal.*, 33(5):1759–1777, 1996.

[23] X. Xu, J.A. Desideri, and A. Janka. Cascadic multigrid for convection-diffusion equation. INRIA Report , INRIA, October 2000.

[24] O.C. Zienkiewicz and R. Codina. A general algorithm for compressible and incompressible flow - Part I. The Split, Characteristic-Based Scheme. *Int. J. Num. Meth. Fluids.*, 20:869–885, 1995.

[25] O.C. Zienkiewicz, K. Morgan, B.V.K. Satya Sai, R. Codina, and M. Vázquez. A general algorithm for compressible and incompressible flow - Part II. Tests on the Explicit Form. *Int. J. Num. Meth. Fluids.*, 20:887–913, 1995.