NASA/CR-2001-210871
ICASE Report No. 2001-13

# Semicoarsening and Implicit Smoothers for the Simulation of a Flat Plate at Yaw

*Ruben S. Montero and Ignacio M. Llorente*
*Universidad Complutense, Madrid, Spain*

*Manuel D. Salas*
*ICASE, Hampton, Virginia*

*ICASE*
*NASA Langley Research Center*
*Hampton, Virginia*

*Operated by Universities Space Research Association*

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-2199

Prepared for Langley Research Center
under Contract NAS1-97046

May 2001

# SEMICOARSENING AND IMPLICIT SMOOTHERS FOR THE SIMULATION OF A FLAT PLATE AT YAW*

RUBEN S. MONTERO†, IGNACIO M. LLORENTE‡, AND MANUEL D. SALAS§

**Abstract.** This paper presents a full multigrid solver for the simulation of flow over a yawed flat plate. The two problems associated with this simulation; boundary layers and entering flows with non-aligned characteristics, have been successfully overcome through the combination of a plane-implicit solver and semicoarsening. In fact, this multigrid algorithm exhibits a textbook multigrid convergence rate, i.e., the solution of the discrete system of equations is obtained in a fixed amount of computational work, independently of the grid size, grid stretching factor and non-alignment parameter. Also, a parallel variant of the smoother based on a four-color ordering of planes is investigated.

**Key words.** plane implicit smoothers, semicoarsening, robust multigrid, flat plate

**Subject classification.** Applied and Numerical Mathematics

**1. Introduction.** The flow of a viscous fluid over a solid obstacle can be divided into two regions of interest. A very thin layer close to the surface of the body in which the velocity gradient normal to the surface is very large, and the remaining region where no such gradients occur and the influence of viscosity can be neglected. It can be shown from several exact solutions of the Navier-Stokes equations that the thickness of the boundary layer is proportional to the square root of the kinematic viscosity. Hence, in the simulation of high Reynolds number flows, a high density of nodes must be concentrated near the body surface to capture the viscous effects numerically.

It is well known that standard multigrid algorithms suffer from a slow-down in convergence in such an anisotropic situation (see for example [1, 15]). There are two main approaches to deal with these anisotropic operators. The first approach consists in improving the smoothing process by using an alternating direction block-implicit smoother [13]. This algorithm explores all the possible directions of coupling of the variables. On the other hand, the second approach relies on improving the coarse-grid operator. Algorithms like selective coarsening [7], flexible multiple semicoarsening [26] or block implicit relaxation combined with semicoarsening [6], among others, fall into this category. Although these methods have been successfully applied to fully elliptic equations [17] and the 2-D Navier-Stokes equations [18, 23] their application to the Navier-Stokes equations in 3-D has been limited.

The simulation analyzed in this work represents an entering flow type with the characteristics entering through one boundary. If the flow does not recirculate, downstream marching results in a very efficient solver for the convective operator. However, if the main stream velocities are not aligned with the grid lines the efficiency of the multigrid method degenerates dramatically. In this case, error components that are much smoother in the characteristic direction than in others, are not well approximated in coarser grids. The main reason is the increasing numerical viscosity induced on coarser levels [3, 9]. One way to prevent this

*This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-97046 while the authors were in residence at ICASE, NASA Langley Research Center, Hampton, VA 23681-2199. The first author was also supported in part by the US-Spain Joint Commission for Scientific and Technological Cooperation.

†Departamento de Arquitectura de Computadores y Automática, Universidad Complutense, 28040 Madrid, Spain

‡Departamento de Arquitectura de Computadores y Automática, Universidad Complutense, 28040 Madrid, Spain

§ICASE, Mail Stop 132C, NASA Langley Research Center, Hampton, VA 23681-2199

degradation in convergence is to use block implicit relaxation combined with semicoarsening [8]. This first approach has been applied to the 3-D constant coefficient convection equation in [14]. When solving the Navier-Stokes equations it is also possible to use distributive Gauss-Seidel relaxation (DGS) [3, 22]. The DGS decouples the elliptic component of the system (solved effectively by multigrid) from the non elliptic one which can be solved through marching.

The two problems involved in the simulation studied here, namely boundary layers and non-aligned grids with open characteristics, have been pointed out as one of the factors that prevent the achievement of optimal multigrid efficiencies in CFD codes [2]. Optimal convergence is defined as the resolution of the governing system of equations in a few (less than ten) work units. A work unit is usually defined as the time required to compute the residual of the system in the finest grid. This property is defined by Brandt as *Textbook Multigrid Efficiency* (TME) [1]. Another desirable quality of a multigrid solver is its robustness. The robustness of a smoother is defined as its ability to efficiently solve a wide range of problems. In this sense the definition of robustness is qualitative and has to be defined more precisely by setting up a set of suitable test problems. In the present context we will characterize the multigrid algorithms as robust if the solution of the governing system of equations can be attained in a fixed amount of work units independent of the grid size, grid stretching factor and the non-alignment parameter. We will refer to this property as *Textbook Multigrid Convergence* (TMC).

The purpose of this work is to present a multigrid algorithm which achieves textbook convergences for the simulation of the flow over a yawed flat plate. In order to solve the two problems outlined above we present in section 3 a FMG-FAS multigrid algorithm based on a plane implicit smoother combined with semicoarsening. The numerical results analyzed in section 4 show that the algorithm used in this work is fully robust for the model problem considered. In that section, we will also investigate a four color ordering of planes which enables the parallel implementation of the smoother. The paper ends with some conclusions.

**2. Finite Volume Discretization.** Let us consider the dimensionless steady state incompressible Navier-Stokes equations:

$$
\begin{aligned}
(\mathbf{u}\cdot\nabla)\mathbf{u} &= -\nabla p + \frac{1}{Re}\Delta\mathbf{u}, \\
\nabla\cdot\mathbf{u} &= 0,
\end{aligned}
\tag{2.1}
$$

where $\mathbf{u} \in \mathbb{R}e^3 = (u, v, w)$ is the non-dimensional velocity field and $p$ is the dimensionless pressure. $Re$ is the Reynolds number defined as $Re = \frac{U_\infty \cdot L}{\nu}$, where $U_\infty$ is a characteristic velocity, $L$ a characteristic length and $\nu$ the kinematic viscosity.

The system of non-linear equations 2.1 is discretized over an orthogonally structured grid with a staggered arrangement of unknowns, where the velocity field is defined on the control volume faces, and the pressure field at the center (see figure 2.1). The most important issues of the finite volume technique will be briefly repeated for the $u$ momentum equation. Integration of the convective terms of 2.1 over a control volume $CV_{ijk}$ gives:

$$
\int_{CV_{ijk}} \mathbf{u}\cdot\nabla u \, dV = \sum_k \int_{\partial CV_{ijk}} u(\mathbf{u}\cdot\mathbf{n})\, dS = \sum_k m_k u_k, \qquad \mathrm{k = e, w, s, n, t, b;}
\tag{2.2}
$$

where $\mathbf{n}$ is the outward normal vector to the $CV_{ijk}$ faces, and the indexes $e, w, ...$ stand for the usual cardinal notation (see [10]). As an example, let us consider the east face. For the approximation of the $u$ velocity an upwind biased definition is used, the low order or *driver* operator used to iterate the solution is a pure

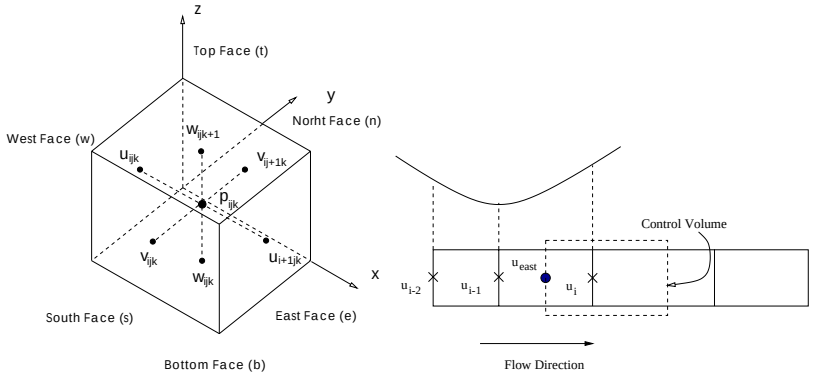


FIG. 2.1. *Placement of the unknowns in the control volume (left-hand chart). Approximation of the velocity at the surface of the control with the QUICK scheme (right-hand chart).*

upwind:

$$u_e = \begin{cases} u_{ijk} + S^+ & (\mathbf{u}\cdot\mathbf{n})_e > 0; \\ u_{i+1jk} + S^- & (\mathbf{u}\cdot\mathbf{n})_e < 0. \end{cases} \tag{2.3}$$

where the $S^{\pm}$ terms corresponds to the contribution of higher order operator corrections. The $S^{\pm}$ source terms are calculated, within the multigrid cycle as explained in section 3, using a second order QUICK [12] scheme. With QUICK, the velocity at the surface of the control volume is interpolated by fitting a parabola to the values of the velocity at three consecutive nodes: the two nodes located on either side of the surface of interest, plus the adjacent node in the upstream direction (see figure 2.1). The mass fluxes $m_k$ can be easily evaluated using linear weighted interpolation. The rest of the details about the discretization of 2.1 have been discussed in detail in [16].

**3. The Multigrid Algorithm.** The base solver that we have employed in this work is a *full multigrid* FMG algorithm [1]. Let us define a set of grids $G = \{\Omega_k : k = 0, 1, 2, ..., N\}$ where $\Omega_0$ is the finest target grid and the rest of the grids are obtained by applying some coarsening procedure. In the FMG algorithm the calculations start on the coarsest grid $\Omega_N$. Once the problem is solved, the solution is interpolated to the next finer level $\Omega_{N-1}$ to provide a good initial approximation to the discrete problem on that level. This procedure is repeated until the finest grid $\Omega_0$ is reached. The main goal of a FMG algorithm is to provide an approximation $\hat{u}^0$ of the discrete solution $u^0$ up to an algebraic error $||\hat{u}^0 - u^0||$ which is smaller than the discretization error $||u - u^0||$.

Because of the non-linearity of the Navier-Stokes equations, each level in the FMG process is solved with some *full approximation scheme* (FAS) [1] multigrid cycles. The FAS cycle for a given grid $\Omega_n$ can be recursively defined as follows; let us consider the non-linear discrete problem on $\Omega_n$:

$$L_n u^n = f^n, \tag{3.1}$$

After applying $\nu_1$ iterations of a non-linear smoother to the system 3.1 a new approximation $\hat{u}^n$ is obtained. Now, the approximation $\hat{u}^n$ and the residual $r^n = f^n - L_n\hat{u}^n$ are transfered to the next coarser grid $\Omega_{n+1}$:

$$\begin{aligned} u^{n+1} &= I_n^{n+1}\hat{u}^n; \\ r^{n+1} &= I_n^{n+1} r^n, \end{aligned} \tag{3.2}$$

the restriction operators $I_n^{n+1}$ are discussed below. On the grid $\Omega_{n+1}$ the defect equation is solved:

$$L_{n+1}\hat{u}^{n+1} = r^{n+1} + L_{n+1}u^{n+1};$$

$$\hat{u}^{n+1} = u^{n+1} + \Delta u^{n+1}, \tag{3.3}$$

where $L_{n+1}$ is some coarse-grid approximation to $L_n$. Note that in the FAS algorithm we solve for the *full approximation* $\hat{u}^{n+1}$ rather than for the correction $\Delta u^{n+1}$. The approximate solution to the coarse-grid problem 3.3 is obtained in $\gamma$ multigrid cycles for the grid $\Omega_{n+1}$. When the coarsest level $\Omega_N$ is reached, the solution to 3.2 is obtained by several relaxation sweeps ($\nu_0$) of the smoothing process. Once the system 3.3 is solved, the correction $\Delta u^{n+1}$ is transferred back to the finer grid $\Omega_n$ and added to the fine-grid approximation:

$$\hat{u}^n \leftarrow \hat{u}^n + I^n_{n+1}(\hat{u}^{n+1} - u^{n+1}), \tag{3.4}$$

so $\nu_2$ sweeps of the non-linear smoothing process are applied to the system 3.1, using the new solution $\hat{u}^n$ as the initial guess. In algorithm 1 a recursive implementation of the FAS cycle is shown.

**Algorithm 1** FAS($\nu_1$,$\nu_2$,$\gamma$,n) multigrid cycle for a given grid $\Omega_n$, where $\nu_1$ and $\nu_2$ denote the number of pre and post-smoothing iterations. The cycle type is fixed with $\gamma$.

```
if n=N then
  û^N = Smooth(L_N, u^N, f^N, ν_0)
else
  û^n = Smooth(L_n, u^n, f^n, ν_1)
  r^n ← f^n − L_n û^n
  r^{n+1} ← I_n^{n+1} r^n
  u^{n+1} ← I_n^{n+1} û^n
  f^{n+1} ← r^{n+1} + L_{n+1} u^{n+1}
  for i = 0 to γ do
    FAS(ν_1, ν_2, γ, n+1)
  end for
  û^n ← û^n + I_{n+1}^n (u^{n+1} − û^{n+1})
  û^n = Smooth(L_n, û^n, f^n, ν_2)
end if
```

The multigrid cycle is characterized by the number of pre and post-smoothing iterations $(\nu_1, \nu_2)$, and $\gamma$ which sets the order in which the grids are visited. Depending on $\gamma$, the cycle is denoted by V$(\nu_1, \nu_2)$ if $\gamma = 1$ and by W$(\nu_1, \nu_2)$ if $\gamma = 2$. We will also consider an F-cycle, which corresponds to an index between the V and W-cycles, i.e. $1 < \gamma < 2$. In figure 3.1 the flowchart of the cycles used in this work are shown. In general, a growing $\gamma$ implies an increasing complexity [27, 24] and more smoothing sweeps on coarser levels which harms the parallel properties of the cycle. However, low $\gamma$ cycles (i.e., V-cycles) are known to be less robust than W-cycles, specially in convection dominated problems [18, 3]. This is one of the reasons why in practice, F-cycles are often used as a trade-off between V and W-cycles. Note also that using semicoarsening as defined subsequently, the storage requirement of the multigrid algorithm is twice that of the single grid algorithm.

**3.1. Restriction and Prolongation.** Solution and residuals transfers are dictated by the staggered arrangement of unknowns and the coarsening procedure used. In the following we will consider a fine grid $\Omega_h$ defined by the nodes:

$$\Omega_h = \{x \in \mathbb{R}^3 : x = \mathbf{kh}, \quad \mathbf{k} = (i, j, k), \quad \mathbf{h} = (h_x, h_y, h_z), \quad i = 0, ..., n_x,$$

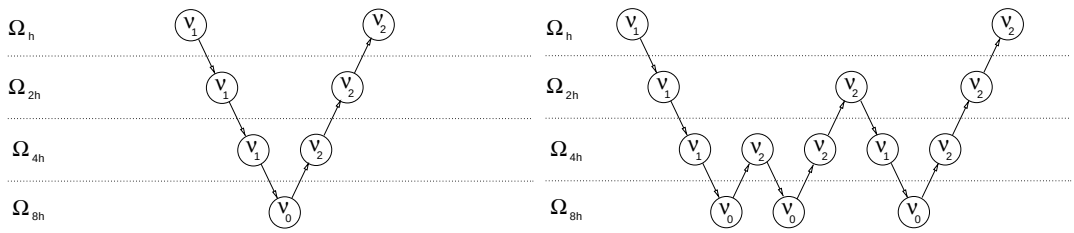


FIG. 3.1. *Scheme of a V-cycle $V(\nu_1, \nu_2)$ (left-hand chart) and an F-cycle $F(\nu_1, \nu_2)$ where $\nu_0$ represents the number of iterations of the smoother performed to solve the coarsest level*

$$j = 0, ..., n_y, \quad k = 0, ..., n_z, \quad h_x = 1/n_x, \quad h_y = 1/n_y, \quad h_z = 1/n_z, \},$$

and a coarse grid $\Omega_{h+1}$ obtained by semicoarsening from $\Omega_h$. With semicoarsening there is only one direction in which the mesh space is doubled, let's say for example $H_x = 2h_x, H_y = h_y, H_z = h_z$ or equivalently $N_x = n_x/2, N_y = n_y, N_z = n_z$. Hence all Fourier modes in the $y$ and $z$ directions can be exactly represented on $\Omega_{h+1}$ and the smoother has only to damp components of the error oscillating highly in the $x$ direction.

The restriction operator $I^h_{h+1}$ in equation 3.2 is used to restrict values from the fine grid $\Omega_h$ to the coarse grid $\Omega_{h+1}$. The component of the velocity parallel to the direction in which the coarsening is performed is transfered using injection, while the other two components together with the scalar field are restricted using linear weighted interpolation:

$$\begin{aligned}
&(I^{h+1}_h u)_{2ijk} = u_{ijk}, \\
&(I^{h+1}_h v)_{2ijk} = (\delta x^- v_{2ijk} + \delta x^+ v_{2i+1jk})/\Delta x, \\
&(I^{h+1}_h w)_{2ijk} = (\delta x^- w_{2ijk} + \delta x^+ w_{2i+1jk})/\Delta x, \\
&(I^{h+1}_h p)_{2ijk} = (\delta x^- p_{2ijk} + \delta x^+ p_{2i+1jk})/\Delta x, \\
&\Delta x = x_{2i+2jk} - x_{2ijk} \quad \delta x^+ = x_{2i+2jk} - x_{2i+1jk} \quad \delta x^- = x_{2i+1jk} - x_{2ijk}, \\
&\forall i \in [0, N_x - 1] \quad \forall j \in [0, N_y - 1] \quad \forall k \in [0, N_z - 1].
\end{aligned}$$

Note that because of the staggered arrangement of unknowns, the velocity component parallel to the coarsened direction is treated in a vertex-centered way, while the rest of the variables are transferred as cell-centered.

The operator $I^{h+1}_h$ in equation 3.4 is used to transfer data from a coarser grid $\Omega_{h+1}$ to the finer $\Omega_h$. In this case the prolongation is a weighted linear interpolation for the *vertex centered* variable:

$$\begin{aligned}
&(I^{h+1}_h u)_{2ijk} = u_{ijk}, \\
&(I^{h+1}_h u)_{2i+1jk} = (\delta x^- u_{ijk} + \delta x^+ u_{i+1jk})\Delta x, \\
&\Delta x = x_{2i+1jk} - x_{2i-1jk} \quad \delta x^+ = x_{2i+1jk} - x_{2ijk} \quad \delta x^- = x_{2ijk} - x_{2i-1jk}, \\
&\forall i \in [0, N_x - 1] \quad \forall j \in [0, N_y - 1] \quad \forall k \in [0, N_z - 1].
\end{aligned}$$

The *cell centered* variables are treated using weighted linear interpolation, for example, for the $v$ component of the velocity field we have:

$$\begin{aligned}
&(I^{h+1}_h v)_{2ijk} = (0.5\delta x^+ v_{ijk} + (\delta x^- + 0.5\delta x^+) v_{i-1jk})/\Delta x \\
&(I^{h+1}_h v)_{2i-1jk} = ((\delta x^+ + 0.5\delta x^-) v_{ijk} + 0.5\delta x^- v_{i-1jk})/\Delta x \\
&\Delta x = x_{2i+2jk} - x_{2ijk} \quad \delta x^+ = x_{2i+2jk} - x_{2i+1jk} \quad \delta x^- = x_{2i+1jk} - x_{2ijk}, \\
&\forall i \in [0, N_x - 1] \quad \forall j \in [0, N_y - 1] \quad \forall k \in [0, N_z - 1].
\end{aligned}$$

When values outside the computational domain are needed in the above set of formulas they are extrapolated from the inner nodes with the help of the boundary condition.

**3.2. Smoothing Operator.** The present approach uses a coupled smoother where the momentum and continuity equation are satisfied simultaneously. In particular, we have chosen a cell-implicit Symmetric Coupled Gauss Seidel (SCGS) method as the base of the smoothing process. This smoother was introduced by Vanka [25] and subsequently considered in [23, 16]. In the SCGS the CV are scanned in some prescribed order, then for each CV the continuity and momentum equations are relaxed as follows:

1. The mass fluxes of the momentum equations for the six cell faces of the CV are calculated. Also in this stage the corrections made by the QUICK scheme are updated based on the current approximation. This is equivalent to a local Picard linearization. Considering implicitly only the diagonals of the momentum equations and those terms multiplying the pressure inside the CV, we can build the following system:

$$\begin{pmatrix} (L_d^u)_{ijk} & 0 & 0 & 0 & 0 & 0 & 1/\delta x \\ 0 & (L_d^u)_{i+1jk} & 0 & 0 & 0 & 0 & -1/\delta x \\ 0 & 0 & (L_d^v)_{ijk} & 0 & 0 & 0 & 1/\delta y \\ 0 & 0 & 0 & (L_d^v)_{ij+1k} & 0 & 0 & -1/\delta y \\ 0 & 0 & 0 & 0 & (L_d^w)_{ijk} & 0 & 1/\delta z \\ 0 & 0 & 0 & 0 & 0 & (L_d^w)_{ijk+1} & -1/\delta z \\ -1/\delta x & 1/\delta x & -1/\delta y & 1/\delta y & -1/\delta z & 1/\delta z & 0 \end{pmatrix} \begin{pmatrix} \widetilde{u}_{ijk} \\ \widetilde{u}_{i+1jk} \\ \widetilde{v}_{ijk} \\ \widetilde{v}_{ij+1k} \\ \widetilde{w}_{ijk} \\ \widetilde{w}_{ijk+1} \\ \widetilde{p}_{ijk} \end{pmatrix} = \begin{pmatrix} R^u_{ijk} \\ R^u_{i+1jk} \\ R^v_{ijk} \\ R^v_{ij+1k} \\ R^w_{ijk} \\ R^w_{ijk+1} \\ 0 \end{pmatrix}, \tag{3.5}$$

   where the $R^{u,v,w}$ terms represents the contribution of the explicit variables.
2. The system 3.5 is easily solved using Gaussian elimination. A more implicit version of the system (3.5) that includes off-diagonal elements in the first six rows is also possible, which corresponds to considering implicitly all the references to unknowns inside the CV. However, the convergence factor is similar and the system is more expensive to solve than 3.5 [23, 16].
3. The velocity components and the pressure of the CV are updated using under-relaxation:

$$u^{n+1} = u^n + \omega_u(\widetilde{u} - u^n)$$
$$p^{n+1} = p^n + \omega_p(\widetilde{p} - p^n).$$

   In the following simulations the under-relaxation factors have been fixed as $\omega_p = 1.0$ and $\omega_u = 0.8$. However, in general the optimum values of $\omega_u$ are strongly problem dependent and have to be set empirically.

**3.3. Plane Implicit Smoothers.** The use of highly anisotropic grids is common practice in the field of CFD. Grid nodes are usually concentrated in certain regions of the computational domain for accuracy reasons or to capture small scale physical phenomena such as the boundary layers mentioned before. In some situations, when the direction of the anisotropies is known beforehand, the multigrid convergence can be improved using an implicit smoother in the direction of strong coupling of the unknowns [1]. These implicit solvers have been widely studied in previous work in the simulation of the incompressible Navier-Stokes equations, see for example [22, 19, 4, 5].

However, if the stretched grid generates aspect ratios whose relative magnitudes vary for different parts of the computational domain, the multigrid techniques based on block-wise smoothers combined with full coarsening fail to smooth error components [1, 27]. In these situations, the problem can be effectively solved with a block implicit smoother combined with semicoarsening. In particular, in the following simulations we

will use an $z$-semicoarsening (i.e., doubling mesh space only in the $z$ direction as explained in section 3.1) combined with a $xy$-plane implicit solver.

From an architectural point of view, it is interesting to note that multigrid methods often reach a disappointingly small percentage of their theoretically available CPU performance due to a limited cache reuse. Some authors have successfully improved locality using different techniques such as data access transformations and data layout transformations [21]. Although we have not introduced any of these techniques, we should remark that plane smoothers exploit blocking in an implicit way, whereas point smoothers have to perform global sweeps through data sets that are too large to fit in the cache. Despite not having discussed cache memory exploitation in this paper, in a previous paper [20] we have included a data reuse analysis for a similar robust multigrid algorithm applied to an anisotropic diffusion equation.

Block implicit smoothers are usually based on a direct solver. These implementations take advantage of the relatively small size of the corresponding implicitly solved 1-D problem. The 3-D counterpart does not present this possibility since the size of the 2-D system is no longer small enough to consider a direct solver. However, an exact direct solver for the planes is not needed, as has been shown in [13] for the 3-D Poisson equation and in [19] for the incompressible Navier-Stokes equations. This consideration drastically reduces the computational cost of the overall algorithm compared to that of a direct plane solver.

In the present work, the planes will be approximately solved with a 2-D multigrid algorithm consisting of one FAS V(1,1) cycle (see figure 3.1). The same kind of anisotropies found in the 3-D problem may appear in the 2-D system. Thus a robust multigrid algorithm is, again, necessary. In particular a cell-wise SCGS smoother described in section 3.2 combined with semicoarsening has been found fully robust for the simulation of the yawed flat plate. However, for greater 2-D problem sizes a point-wise smoother may not be fully robust, and an implicit line smoother or a greater $\gamma$ cycle should be used [3, 16]. The other components of the 2-D multigrid cycle, such as the restriction and prolongation operators or the smoothing process can be easily deduced from those derived in sections 3.1 and 3.2.

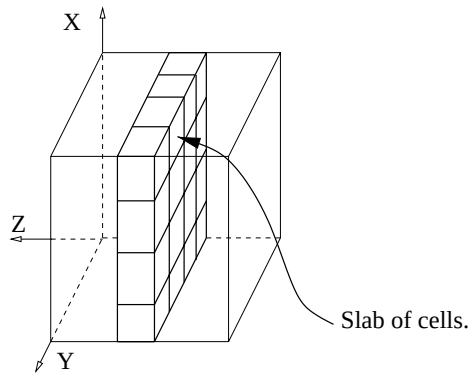


FIG. 3.2. *Slab of cells updated simultaneously when using the xy-plane implicit smoother.*

The coupled philosophy of the SCGS will also be applied in the plane solver. The plane smoother simultaneously relaxes the momentum and continuity equations of the cells included in the plane. Note that the plane is understood as a slab of cells as shown in figure 3.2. Hence, all velocity components and pressures contained within the plane will be updated at the same time. Let us consider for example a $yz$-plane, for which the procedure to solve the 2-D problem on the plane is as follows:

1. The mass fluxes and second-order corrections of the 2-D system are computed based on the current solution in the plane. This step corresponds to a global linearization of the 2-D problem. Let us

define the vector $X_i$ that accommodates the variables for the whole plane of cells as:

$$
\begin{aligned}
&X_i^T = (\mathbf{u}, \mathbf{u}^+, \mathbf{v}, \mathbf{w}, \mathbf{p}) \quad \text{with} \\
&\mathbf{u} = \{u_{ijk} \,:\, i = I,\ \forall k, j \in [0, n]\}, \quad \mathbf{u}^+ = \{u_{ijk} \,:\, i = I+1,\ \forall k, j \in [0, n]\}, \\
&\mathbf{v} = \{v_{ijk} \,:\, i = I,\ \forall k, j \in [0, n]\}, \quad \mathbf{w} = \{w_{ijk} \,:\, i = I,\ \forall k, j \in [0, n]\}, \\
&\mathbf{p} = \{p_{ijk} \,:\, i = I,\ \forall k, j \in [0, n]\}.
\end{aligned}
$$

The system of equations for the plane can be written in terms of residuals and corrections as:

$$L_i \ \Delta X_i = r_i, \tag{3.6}$$

where $r_i = f_i - L_i X_i$ is the residual of the $i^{th}$ $yz$-plane and $\Delta X_i = X_i^{n+1} - X_i^n$ is the increment of the solution.

2. System 3.6 is solved by applying one FAS V(1,1) multigrid cycle as explained before. In the following simulations the coarsest level is solved with 10 iterations of the smoothing process ($\nu_0 = 10$).
3. The solution in the plane is updated via underrelaxation:

$$
\begin{aligned}
X_i^{n+1} &= X_i^n + \omega \Delta X_i, \\
\omega &= (\omega_u, \omega_u, \omega_u, \omega_u, \omega_p),
\end{aligned}
$$

where $\omega_u$ and $\omega_p$ are the underrelaxation factors for the velocity and pressure field defined in section 3.2.

Many types of plane smoothers can be easily constructed by defining a specific ordering of the planes. It is important to note that the second order operator extends the stencil to the planes $i$, $i \pm 1$ and $i \pm 2$ depending on the direction of the flow. In order to avoid these dependencies, the construction of the smoother can be based on a four color ordering of planes (see figure 3.3). Because a four color smoother should be used as the basis of a parallel implementation, the behavior of this smoother will be analyzed and compared to the sequential lexicographic order in section 4.

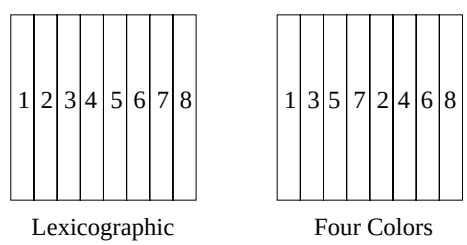


FIG. 3.3. *Planes orderings for the implicit plane smoother.*

**4. Numerical Experiments.** We consider a cascade of square plates of side $L$ as shown in figure 4.1. In particular, we assume that the leading edge of the plate is placed at a distance $L$ from the inflow plane of the computational domain. The outflow boundary is situated at $2L$ after the trailing edge. The boundary conditions are defined as follows: on the west face of the computational domain we prescribe a inflow conditions consistent with some angle of yaw ($\theta$). The north and south faces satisfy a periodic boundary condition. In order to assure periodicity in the $y$ direction, we will only consider angles of yaw such that there is no interaction between the wakes of the plates. The largest angle of yaw studied in this work will be $\theta = 45°$. For this situation the wake of the plates ($||u|| < 1$) at $z = 0$ has been shaded in figure

4.1. Periodicity in the $y$ direction is clearly shown for this case, and also therefore for the angles $0° \leq \theta \leq 45°$ used in the following simulations. Note that for $\theta = 0°$ the periodic boundary condition reduces to symmetry on the north and south boundary. On the plate we impose a no-slip condition while the top and bottom faces will hold symmetric boundary conditions.

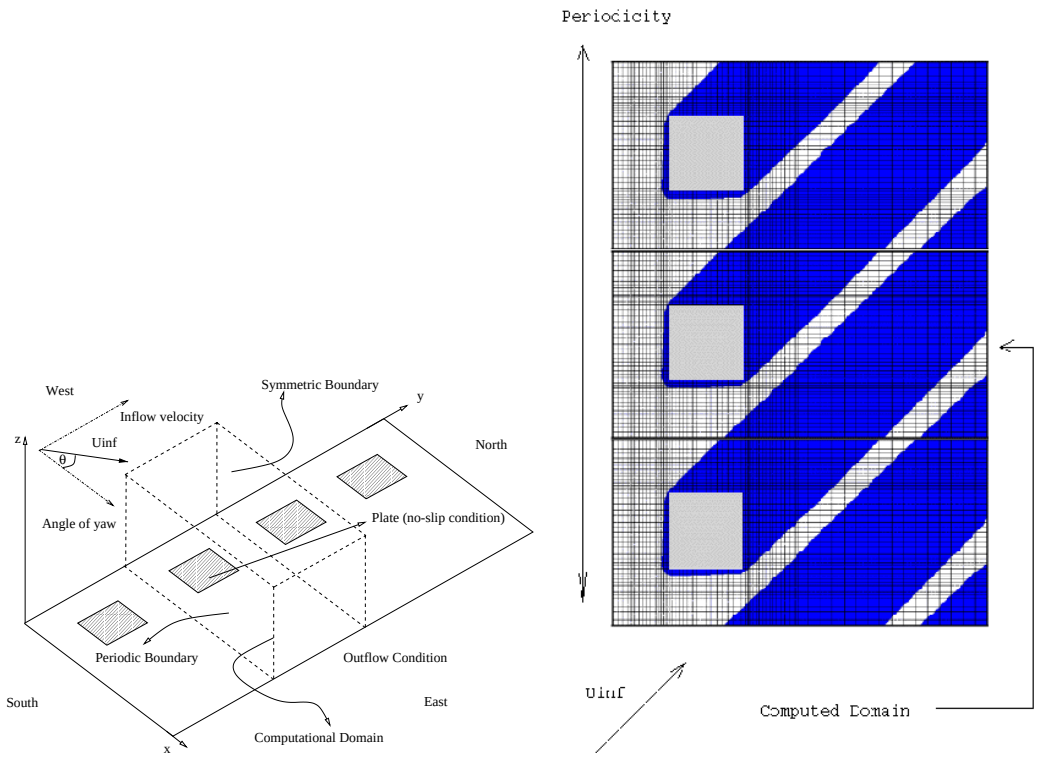


FIG. 4.1. *Schematic configuration of the numerical experiments (left-hand chart). Periodicity in y-direction for the simulation of a flat-plate at $\theta = 45°$, the shaded area corresponds to the wake of each plate.*

The treatment of the outflow needs a closer a look since we need to apply a proper boundary condition in the wake region. For the main stream flow we can impose a zero gradient of the velocity field in the $X$ direction:

$$\frac{\partial U}{\partial X} = \frac{\partial V}{\partial X} = \frac{\partial W}{\partial X} = 0. \tag{4.1}$$

However these equations are not valid in the wake of the plate and need to be improved. In most cases, the flow in the wake of a plate is turbulent. For $Re < 10^6$, under carefully controlled conditions, the boundary layer can remain laminar past the trailing edge. However because the velocity profiles in the wake have a point of inflexion, the wake usually becomes turbulent away from the trailing edge. In this work we will confine $Re < 10^5$ and assume that the wake remains laminar at a distance of $2L$ behind the trailing edge. For the wake region, the outflow boundary condition will be derived using Goldstein's calculations of the velocity distribution in the wake of a finite 2-D flat plate [11]. Hence, in order to apply the proper boundary conditions, it is important to set the wake bounds. We will assume that the wake region has a negligible thickness, beyond the width of the plate, $L/cos(\theta)$, in the $y$ direction ($\delta_{xy} \approx 0$), see figure 4.2. The estimation of the thickness in the $z$ direction is based on the numerical values [11] of the velocity in the wake at a given distance behind the plate. In particular, with the outflow placed at $2L$ after the trailing edge of the plate, and fixing the wake limit where $||u|| = 0.999$, for a Reynolds number of $10^4$, it is found that $\delta_{xz} \approx 0.1L$.

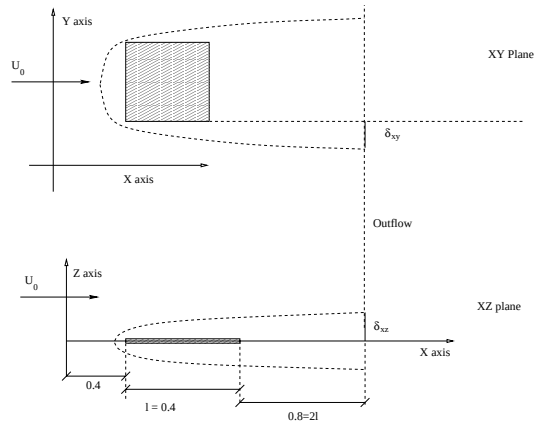


FIG. 4.2. *Wake dimensions of the plate at zero angle of yaw*

We will first consider the plate at zero angle of yaw. The gradient of the velocity component parallel to the inflow velocity, and that perpendicular to the plate, will be improved using Goldstein's calculation for the 2-D plate. The notation introduced in [11] will be reproduced here in order to clarify the subsequent expressions:

| | |
|---|---|
| $Re_L = U_\infty L/\nu$ | : Reynolds number based on the plate length |
| $D = 1.328\rho U_\infty^2 L/\sqrt{Re_L}$ | : Viscous drag coefficient from Blasius theory |
| $\kappa_D = D/\rho U_\infty^2 d$ | : dimensionless drag coefficient |
| $A = 1.328/4\sqrt{\pi}$ | : Constant of integration |
| $d = 4L$ | : representative length of the plate |
| $X$,$Z$ | : 2-D Cartesian coordinates |
| $U$,$W$ | : Dimensional 2-D velocity field |
| $x = X/d$, $z = (U_\infty d/\nu)^{1/2} Z/d$ | : dimensionless coordinates |
| $u = U/U_\infty$, $w = (U_\infty d/\nu)^{1/2} W/U_\infty$ | : dimensionless components of the velocity field |

In what follows we will only need the component of the velocity perpendicular to the plate which can be written with a second order approximation as:

$$w = -\frac{A}{\sqrt{2x}} g_1(\eta) + \frac{A}{x^{3/2}} \left[ \frac{A}{2^{3/2}} \left( g_2(\eta) + g_2(\eta) + g_2(\eta) \right) + \frac{\eta}{\sqrt{2\pi Re_L}} \right], \tag{4.2}$$

where:

$$\begin{aligned} \eta &= \frac{z}{\sqrt{2x}}; \\ g_1(\eta) &= \eta e^{-\frac{1}{2}\eta^2}; \\ g_2(\eta) &= \sqrt{\frac{\pi}{2}} (1-\eta^2) e^{-\frac{1}{2}\eta^2} \operatorname{erf}\left(\frac{\eta}{\sqrt{2}}\right); \\ g_3(\eta) &= -\eta e^{-\eta^2}; \\ g_4(\eta) &= -\sqrt{(}\pi) \operatorname{erf}(\eta). \end{aligned} \tag{4.3}$$

The gradient of $U$ in the $X$ direction, will be calculated in terms of the derivative of $w$ with respect to $z$. With the help of the mass conservation equation we can express the $U$ derivative as:

$$\frac{\partial U}{\partial X} = -\frac{U_\infty}{d} \frac{\partial w}{\partial \eta} \frac{\partial \eta}{\partial z};$$

$$\frac{\partial w}{\partial \eta}\frac{\partial \eta}{\partial z} = -\frac{A}{2x}g_{1\eta} + \frac{A}{(2x)^2}\left[A\left(g_{2\eta}+g_{2\eta}+g_{2\eta}\right) + \frac{2}{\sqrt{\pi Re_L}}\right], \tag{4.4}$$

where the functions $g_{i\eta}$ are:

$$\begin{aligned}
g_{1\eta} &= (1-\eta^2)e^{-\frac{1}{2}\eta^2};\\
g_{2\eta} &= \sqrt{\frac{\pi}{2}}e^{-\frac{1}{2}\eta^2}\left[\eta(\eta^2-3)\mathrm{erf}\left(\frac{\eta}{\sqrt{2}}\right) + (1-\eta^2)\sqrt{\frac{2}{\pi}}e^{-\frac{1}{2}\eta^2}\right];\\
g_{3\eta} &= (2\eta^2-1)e^{-\eta^2};\\
g_{4\eta} &= -2e^{-\eta^2}.
\end{aligned} \tag{4.5}$$

Although expression 4.1 is probably a good outflow condition for the $W$-component we can get a better representation of the gradient using the derivative of expression 4.2 with respect to $X$, that can be written as:

$$\frac{\partial W}{\partial X} = \frac{U_\infty}{2d\sqrt{Re_L}}\left[\frac{\partial w}{\partial x} + \frac{\partial w}{\partial \eta}\frac{\partial \eta}{\partial x}\right], \tag{4.6}$$

where,

$$\begin{aligned}
\frac{\partial \eta}{\partial x} &= -\frac{\eta}{2x};\\
\frac{\partial w}{\partial x} &= \frac{A}{(2x)^{3/2}}g_1\eta - \frac{3A}{(2x)^{5/2}}\left[A(g_2(\eta)+g_3(\eta)+g_4(\eta)) + \frac{2\eta}{\sqrt{\pi Re_L}}\right].
\end{aligned} \tag{4.7}$$

The benefits of the boundary conditions that we have just derived are clearly shown in figure 4.3. In this figure, the pressure contour lines at $z = 0$ have been plotted for two different simulations with $\theta = 0°$ and a Reynolds number of 10000. In the right-hand chart the boundary condition 4.1 has been used for the whole outflow boundary. At the outflow, in the wake of the plate, there is a region of a favorable pressure gradient that accelerates the velocity at the outlet in order to match the wrong outflow condition. In the left-hand chart the boundary conditions 4.4 and 4.6 have been used in the wake region combined with 4.1 outside the wake. As a result of this boundary condition the low pressure zone in the wake of the plate disappears.

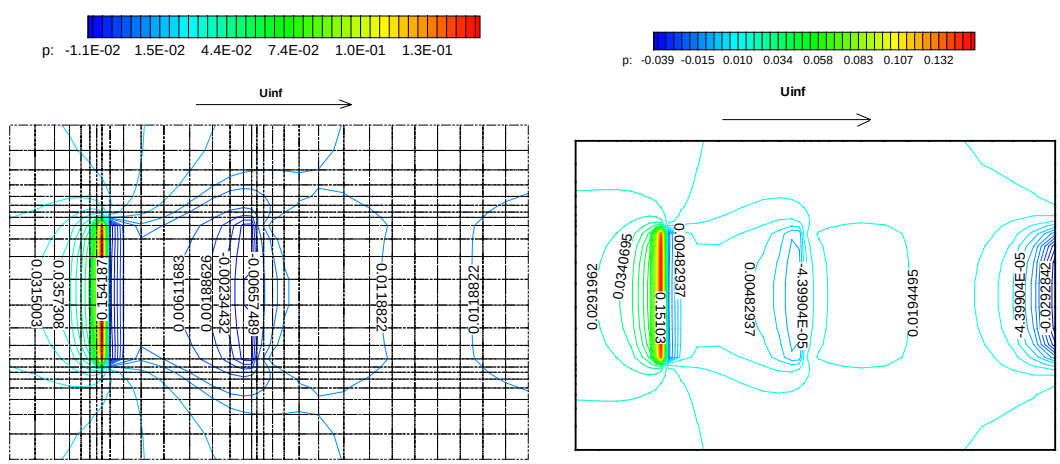


FIG. 4.3. *Pressure contour lines at z=0 for a Reynolds number of 10000 using a zero gradient wake condition (right-hand chart) and the improved boundary condition (left-hand chart) for* $\theta = 0°$.

The solution is verified by comparing the $u$-velocity in the middle of the plate with the Blasius analytical solution for a 2-D plate (figure 4.4). The low discrepancy near the layer edge is due to the highly stretched grid used in this simulation. The spatial convergence of the viscous drag coefficient $D_f$ versus the nominal grid spacing for a set of grids with an increasing number of points in the $z$ direction (regular spacing) is shown in figure 4.4, where:

$$D_f = \int_{plate} \mu \frac{\partial u}{\partial z}_{z=0} dS \tag{4.8}$$

Second order accuracy is clear; the coefficient converges to a value approximately 3% lower ($D_f = 0.020455$) than the one predicted by the Blasius theory ($D_f = \frac{1.328}{\sqrt{Re_l}} = 0.02099$). It is interesting to note that the accuracy obtained with the 128×24×32 grid is the same as that obtained with the 32×24×32 stretched grid, with the consequent saving in computing time.

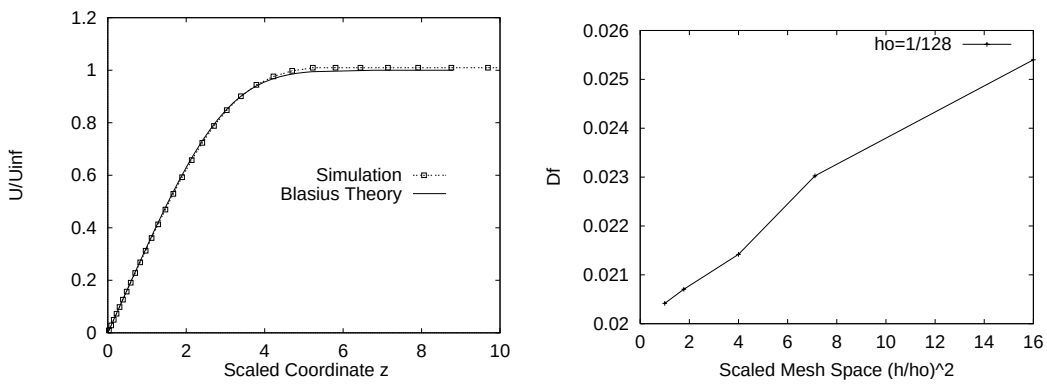


FIG. 4.4. *Simulation comparison with Blasius theory at the middle of the plate with $Re = 10^4$ (left-hand chart). Convergence of the viscous drag versus nominal grid spacing (right-hand chart).*

When the inflow is at some angle of yaw, the outflow-wake boundary conditions are approximated by rotating the zero yaw angle solution. This approximation is only valid for *moderate* angles of yaw, given that the characteristics of the problem do not diverge too much from the main stream. However, in a general situation for larger angles of yaw, such as those considered below, this assumption is no longer valid. The effect of the rotated outflow conditions on the plate has been checked by running a case with the outflow plane further out. The flow field over the plate is not perceptively affected by the rotated boundary conditions. In fact, when comparing both results the difference in the flow over the plate has been found to be below the 2%. If we define $(x', y', z')$ as a frame rotated by the angle of yaw $\theta$, and, if $(u', v', w')$ are the velocity components in the $(x', y', z')$ frame we have:

$$\begin{aligned} u &= u' \cos\alpha + v' \sin\alpha \\ v &= -u' \sin\alpha + v' \cos\alpha \\ w &= w' \end{aligned} \tag{4.9}$$

Now, the gradients $\partial U/\partial X$ and $\partial W/\partial X$ can be easily obtained from equation 4.9. Note also that at yaw, the effective length of the plate changes and so equation 4.2 has to be modified to take this into account.

In figure 4.5 the drag coefficient versus the angle of yaw is shown for the 64×48×64 grid with a Reynolds number of $10^4$. In this case, the contribution from the $u$ and $v$ components of the velocity field have to be

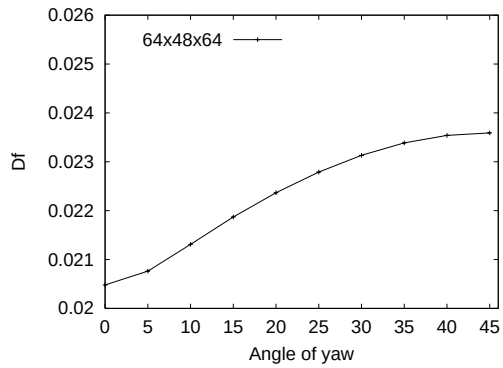


FIG. 4.5. *Viscous Drag coefficient versus the angle of yaw for a* $64\times48\times64$ *stretched grid at* $Re = 10^4$.

taken into account in the shear stress vector:

$$D_f = \int_{plate} \mu \left( \frac{\partial u}{\partial z} \cos\alpha + \frac{\partial v}{\partial z} \sin\alpha \right)_{z=0} dS. \tag{4.10}$$

This integral is computed with second order accuracy using the midpoint rule, where the derivatives are approximated with a three point formula. The pressure contours at $z = 0$ for two different angles of yaw ($\theta = 20°$ and $\theta = 45°$) are shown in figure 4.6. For these cases, the pressure gradients in the wake of the plates at the outflow can also be seen. However, the magnitude of these gradients are small compared to those that appear at $\theta = 0°$ and the zero gradient boundary condition for the wake of the plate.

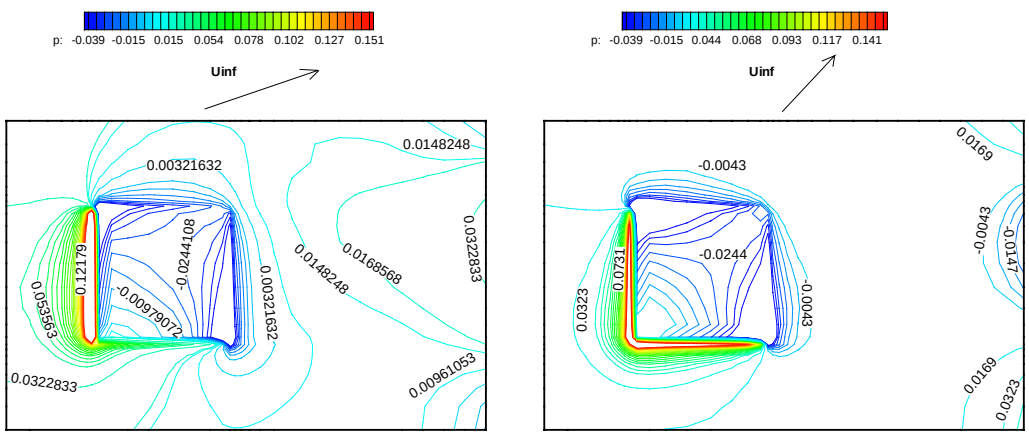


FIG. 4.6. *Pressure contour lines at z=0 for a Reynolds number of 10000 for* $\theta = 20°$ *(left-hand chart) and* $\theta = 45°$ *(right-hand chart).*

In order to capture the viscous effects, the grids used in this work are highly stretched near the plate. Moreover, the grids are refined (geometrically stretched, see below) near the plate edges to reduce the large discretization errors in those zones as advocated by Thomas et al. [22] (see for example figure 4.3). The simulations have been performed over two different grids of dimensions: $32\times24\times32$ and $64\times48\times64$. These grids use a stretching of the form $h_{k+1} = \beta h_k$, the stretching factor $\beta$ being equal to 1.3 and 1.2 respectively. It is interesting to note that, in order to keep the $h^2$ solution accuracy the stretching factor should be $\beta = 1 + O(h)$. In the following simulations the number of multigrid levels has been fixed so that the coarsest level has 4 planes.

Figure 4.7 shows the $L_2$-norm of the residual versus F(2,1)-cycles with an $xy$-plane implicit smoother combined with $Z$ semicoarsening for several angles of yaw and two different smoothers. The residual norm is reduced by four orders of magnitude in the first five cycles in the finest grid in all cases (corresponding to a convergence rate of roughly 0.1 per fine grid iteration). This average convergence rate is close to that obtained for the 3-D Poisson equation using a plane implicit smoother combined with semicoarsening [17]. Moreover, the convergence rate is also independent of the angle of the non-alignment parameter for angles $0^\circ \leq \theta \leq 45^\circ$.

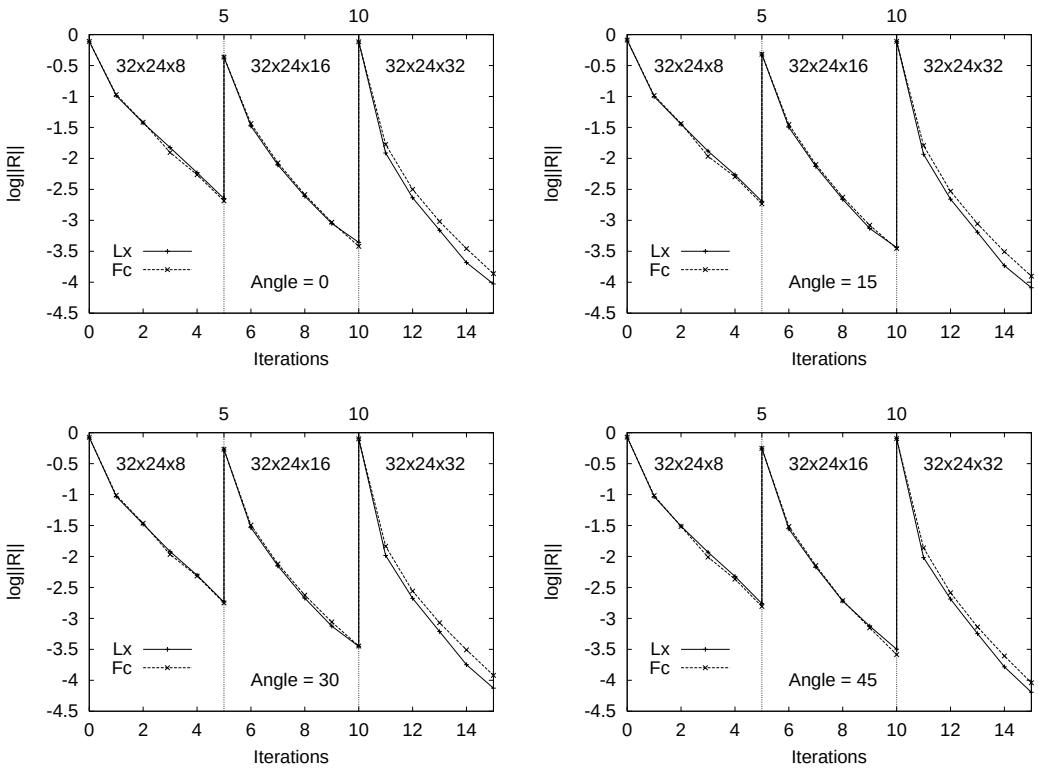


FIG. 4.7. *$L_2$-norm of the average residual versus F(2,1)-cycles with an $xy$-plane implicit smoother combined with $z$ semicoarsening with a lexicographic (Lx) and a four-color (Fc) ordering of planes, and several angles of yaw. The Simulation was performed on a $32 \times 24 \times 32$ grid and a Reynolds number of $10^4$.*

Figure 4.8 shows the $L_2$-norm of the residual versus F(2,1)-cycles for a $64 \times 48 \times 64$ grid. The residual norm in this case is also reduced by four orders of magnitude in the first five cycles in the finest grid in all cases. Thus, the combination of plane-implicit smoothers with semicoarsening considered here exhibits a textbook convergence rate, i.e., independent of the problem size, grid stretching factor and angle of yaw. Although this problem is a convection-dominated non-aligned flow for which multigrid schemes might have convergence difficulties, we do not experience these difficulties for the grids and Reynolds numbers used. However in a general situation an explicit coarse-grid correction of cross-characteristic components may be needed (see for example [8]). Moreover, when using a four-color smoother the efficiency of the algorithm is not reduced. This result enables a viable parallel implementation of the multigrid algorithm studied.

We have studied the case of $Re = 10^5$ and have obtained a convergence rate of 0.2 and a drag coefficient within 5% of that predicted by the Blasius solution. The results indicate that a finer mesh is needed to improve the accuracy of the results. This will be the subject of further studies that would exploit the

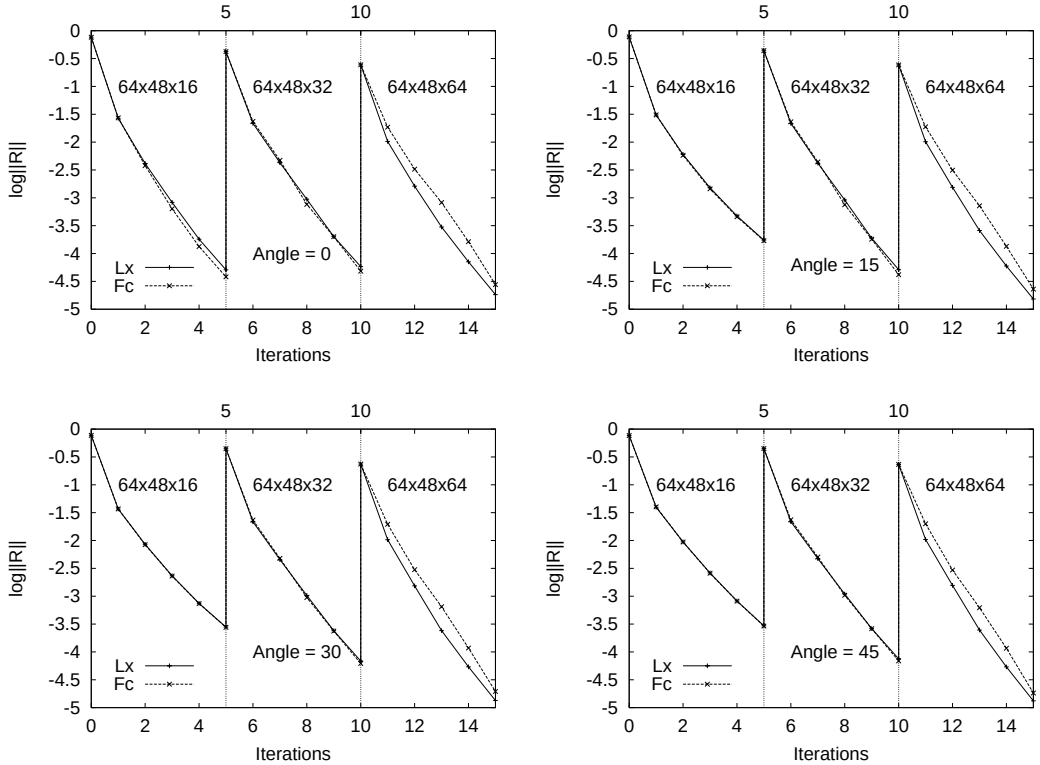


FIG. 4.8. *$L_2$-norm of the average residual versus F(2,1)-cycles with an xy-plane implicit smoother combined with z semicoarsening with a lexicographic (Lx) and a four-color (Fc) ordering of planes and several angles of yaw. The Simulation was performed on a $64 \times 48 \times 64$ grid and a Reynolds number of $10^4$.*

parallel properties of the algorithm.

**5. Conclusions.** A multigrid algorithm to solve the incompressible 3-D Navier-Stokes equations with severely stretched grids and non-aligned flows has been presented. The core of the multigrid algorithm is a coupled plane-implicit solver combined with semicoarsening. The plane solver used is also a robust 2-D multigrid algorithm based on a cell-implicit smoother combined with semicoarsening. This plane smoother has been found fully robust for the problem sizes covered in this work. Textbook multigrid convergence has been demonstrated for the simulation of a yawed flat plate boundary layer. That is, the convergence rate is independent of the grid size, grid stretching factor and the angle of yaw. Moreover, this convergence factor has been found to be close to the value expected for elliptic equations. A parallel variant of the smoothing process based in a four-color ordering of planes has also been analyzed. The convergence rate does not deteriorate in this situation, achieving TMC for the parallel smoother. The outflow boundary treatment has been discussed in depth in this report. The code has been validated with Blasius theory and an accurate measurement of the viscous drag coefficient has been obtained by exploiting the robustness of the solver.

## REFERENCES

[1] A. BRANDT, *Multigrid techniques: 1984 guide with applications to fluid dynamics*, Tech. Rep. GMD-Studien 85, GMD, May 1984.

[2] ——, *Barriers to Achieving Textbook Multigrid Efficiency (TME) in CFD*, ICASE Interim Report No.

32, (1998).

[3] A. Brandt and I. Yavneh, *On multigrid solution of high-Reynolds incompressible entering flows*, J. Comput. Phys., 101 (1992), pp. 151–164.

[4] D. J. Mavriplis, *Multigrid strategies for viscous flow solvers on anisotropic unstructured meshes*, Tech. Rep. 98-6, ICASE, 1998.

[5] ——, *Large scale parallel viscous flow computations using and unstructured multigrid algorithm*, Tech. Rep. 99-44, ICASE, 1999.

[6] J. E. Dendy, S. F. McCormick, J. Ruge, T. Russell, and S. Schaffer, *Multigrid methods for three-dimensional petroleum reservoir simulation*, in Tenth SPE Symposium on Reservoir Simulation, February 1989.

[7] B. Diskin, *Multigrid algorithm with conditional coarsening for the nonaligned sonic flow*, Electronic Trans. Num. An., 6, pp. 106–119. Proceedings of *the Eighth Copper Mountain Conference on Multigrid Methods*, 1997.

[8] B. Diskin, *Solving upwind-biased discretizations II: Solver using semicoarsening*, Tech. Rep. 99-25, ICASE, 1999.

[9] B. Diskin and J. L. Thomas, *Solving upwind-biased discretizations: Defect-correction iterations*, Tech. Rep. 99-14, ICASE, 1999.

[10] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*, Springer-Verlag, 1996.

[11] S. Goldstein, *On the two-dimensional steady flow of a viscous fluid behind a solid body*, Proc. Royal Society of London Series A, (1933), pp. 545–573.

[12] T. Hayase, J. Humphrey, and R. Greif, *A Consistently Formulated QUICK scheme for fast and stable convergence using finite-volume iterative calculation procedures*, Computational Physics, 98 (1992), pp. 108–118.

[13] I. M. Llorente and N. D. Melson, *Behavior of plane relaxation methods as multigrid smoothers*, Electronic Transactions on Numerical Analysis, 10 (2000), pp. 92–114.

[14] I. M. Llorente, M. Prieto-Matias, and B. Diskin, *An efficient parallel multigrid solver for 3-d convection-dominated problems*, Tech. Rep. 2000-29, ICASE, 2000.

[15] O. A. McBryan, P. O. Frederickson, J. Linden, A. Schuller, K. Solchenbach, K. Stuben, C. Thole, and U.Trottenberg, *Multigrid methods on parallel computers-a survey of recent developments*, Impact of Computing in Science and Engineering, 3 (1991), pp. 1–75.

[16] R. S. Montero and I. M. Llorente, *Robust multigrid algorithms for the incompresible navier-stokes equations*, Tech. Rep. 2000-27, ICASE, 2000.

[17] R. S. Montero, M. Prieto, I. M. Llorente, and F. Tirado, *Robust multigrid algorithms for 3-D elliptic equations on structured grids*, in Multigrid Methods VI, E. Dick, K. Riemslagh, and J. Vierendeels, eds., 1999, pp. 193–199.

[18] C. Oosterlee, F. Gaspar, T. Washio, and R. Wienands, *Multigrid line smoothers for higher order upwind discretizations of convection-dominated problems*, J. Comput. Phys., 1 (1998), pp. 274–307.

[19] C. W. Oosterlee, *A GMRES-based plane smoother in multigrid to solve 3-D anisotropic fluid flow problems*, J. Comput. Phys., 130 (1997), pp. 41–53.

[20] M. Prieto, R. Montero, D. Espadas, I. Llorente, and F. Tirado, *Parallel multigrid for anisotropic elliptic equations*, Journal Parallel and Distributed Computing, (2000).

[21] U. Rüde, *Iterative algorithms on high performance architectures*, in Proceedings of the Europar'97 Conference, 1997, pp. 57–71.

[22] J. Thomas, B. Diskin, and A. Brandt, *Textbook multigrid efficiency for the incompressible Navier-Stokes equations: High reynolds number wakes and boundary layers*, Tech. Rep. 99-51, ICASE, 1999.

[23] M. Thompson and J. Ferziger, *An adaptative multigrid technique for the incompressible Navier-Stokes equations*, J. Comput. Phys., 82 (1989), pp. 94–121.

[24] U. Trottenbrg and K. Oosterlee, *Parallel adaptative multigrid*, Tech. Rep. 1026, GMD, 1996.

[25] S. P. Vanka, *Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables*, Computational Physics, 65 (1986), pp. 138–158.

[26] T. Washio and K. Oosterlee, *Flexible multiple semicoarsening for three-dimensional singularly perturbed problems*, SIAM J. Sci. Comput., 19 (1998), pp. 1646–1666.

[27] P. Wesseling, *An Introduction to Multigrid Methods*, John Wiley & Sons, New York, 1992.