



Multigrid correction-storage formulation applied to the numerical solution of incompressible laminar recirculating flows

José A. Rabi, Marcelo J.S. de Lemos *

*Departamento de Energia—IEME, Instituto Tecnológico de Aeronáutica—ITA-CTA,
12228-900, São José dos Campos SP, Brazil*

Received 8 September 1999; received in revised form 18 February 2003; accepted 28 March 2003

Abstract

Multigrid methods are known to reduce computational time of iterative solutions. In this paper, a multigrid technique is implemented following a correction storage (CS) formulation and a V-cycle strategy to numerically solve steady-state two-dimensional incompressible laminar recirculating flows. Structured, orthogonal and irregular meshes are employed to perform a finite volume discretization. Pressure–velocity is accomplished through the SIMPLE method and the TDMA and Gauss–Seidel algorithms are used to relax the resulting algebraic equations. The solution method is tested against the laminar flow between parallel plates and recirculating flow patterns are qualitatively presented. The advantages of using more than one grid level and the CS approach are discussed upon.

© 2003 Elsevier Inc. All rights reserved.

1. Introduction

Computational fluid dynamics has experienced improvements related not only to the availability of fast and high memory capacity computers but also to the development and application of efficient iterative methods. Numerical simulation is already incorporated to solving “real world” engineering problems like energy generation processes, environmental phenomena and flight engineering, to mention a few. In these problems, fluid flow (with or without heat transfer) is present and should be properly described.

* Corresponding author. Tel.: +55-12-3947-5860; fax: +55-12-3947-5842.

E-mail address: delemos@mec.ita.br (M.J.S. de Lemos).

Nomenclature

a_W, a_E, a_S, a_N, a_P	coefficients in discretized transport equation
b	source term in discretized transport equation
\mathbf{b}	matrix of source terms
\mathbf{A}	matrix of coefficients
C_w, C_e, C_s, C_n	convective (mass) fluxes
D	diffusive flux
$f_{x,W}, f_{x,P}, f$	linear interpolation factors
I_k^{k-1}, I_{k-1}^k	restriction and prolongation operators
L_x, L_y	solution domain lengths
M	total number of computational grids employed
NI, NJ	total number of control volumes (grid nodes)
P	pressure
P'	pressure correction (SIMPLE method)
r	residue (of a single control volume)
\mathbf{r}	matrix of residues
$R_U, R_V, R_{P'}$	normalized residues (finest grid)
Re	Reynolds number
s_U, s_V	viscous source terms
S_ϕ	source term in momentum equations (pressure gradient)
S_m	source term in pressure correction equation (mass imbalance)
t	CPU time
U, V	Cartesian velocity components
U_0	inlet velocity value (uniform profile)
x, y	Cartesian coordinates
$\delta x, \delta y$	control volume dimensions
$\Delta x_w, \Delta x_e, \Delta y_s, \Delta y_n$	internodal distances
ϕ	flow quantity
Φ	matrix of coarse grid corrections (for flow quantities)
Φ	matrix of flow quantities (finest mesh)
Γ_ϕ	diffusion coefficient of flow quantity
λ	flux blended deferred correction scheme combination factor
μ	molecular viscosity
ρ	density
$\xi_U, \xi_V, \xi_{P'}$	under-relaxation factors
<i>Subscripts</i>	
aux	auxiliary (intermediate) grid
k	computational grid level
M	finest grid level
nb	neighboring grid points (in summations)
P	central grid point

w, e, s, n west, east, south, north control volumes faces
 W, E, S, N west, east, south, north neighboring grid points
 1 coarsest grid level

Superscripts

CDS central differencing scheme
 DCS flux blended deferred correction scheme
i, j fine grid point identifiers
I, J coarse grid point identifiers
 new updated value
 UDS upwind differencing scheme
 * previous iterative values
 – approximated value

In order to obtain accurate results, well-refined meshes are often needed and their usage increases the computational effort. Accordingly, convergence rates of iterative numerical solutions are greatest in the beginning of calculations, slowing down sensibly as the iterative process goes on. Such numerical behavior can be explained by a spectral analysis, namely, the iterative method is capable of reducing efficiently only those Fourier error components whose wavelengths are smaller than or comparable to the grid spacing [1–3].

Multigrid methods achieve convergence acceleration by covering a broader wavelength spectrum through the iteration at a sequence of gradually less refined grids instead of iterating at a single grid. Long wavelengths in a fine mesh become smaller in a coarse one, where they can be better smoothed out. Therefore, in each grid level visited by the solution process, the corresponding error components are efficiently reduced, speeding up convergence.

Depending on how variables are handled in coarse meshes (i.e., in all meshes but the finest), a multigrid algorithm may be implemented following two distinct formulations. In the correction storage (CS) formulation, algebraic equations in coarse meshes are solved for the corrections of the dependable variables. On the other hand, in the full approximation storage (FAS) formulation, suitable for solving non-linear problems [1–3], the variables themselves are handled in all grid levels. Further, for the special case of a fixed [4] or calculated [5] flow field, studies can be conducted in order to estimate optimal multigrid parameters when using the CS scheme [4].

As mentioned, the literature recommends the application of the CS formulation for the solution of linear problems whereas the FAS formulation is more suited for non-linear situations [1–3]. Nevertheless, Jiang et al. [6] reported the numerical solution of the Navier–Stokes (non-linear) equations using the multigrid CS formulation. Motivated by this prior instance, the present work applies the multigrid method following the CS formulation to solve steady-state two-dimensional incompressible laminar recirculating flows, in geometries to be later described.

Justification for applying CS methods to non-linear problems is here based on the following argument. When seeking stable and faster solutions for fluid flow problems, proper handling of the coupling among all velocity components, any scalars (such as temperature or mass fraction) and the pressure field, might turn out to be as important as the use of accelerating algorithms for individual variables, such as multigrid schemes. Also, for mid-size two-dimensional problems, the

gain in using multigrid is not as evident as in three-dimensional flows and, in these instances, adequate treatment of variable coupling involving thermal field [7], swirl [8], or buoyant swirling flows [9] might also offer advantages in the reduction of the overall computer effort required for convergence. Therefore, one can identify two independent and distinct factors that contribute to reducing the time to convergence: (a) adequate handling of the pressure–velocity–temperature–swirl coupling via a block-implicit solver [7–9], and (b) use of multigrid for relaxing individual variables in a segregated fashion [6]. In this sense, the use of simpler CS algorithms in conjunction with advanced implicit methods, instead of using more elaborate FAS algorithms, might benefit the overall solution process when numerically solving engineering flows.

As such, the aim of this contribution is to assess the use of a simpler CS method in absence of any additional numerical artifice to achieve fast convergence, such as the block-implicit treatment of the flow variables mentioned above. The novelty herein is therefore to evaluate the relative computational gain if linearization of the velocity–pressure coupling is handled via the CS artifice, without using any block-implicit scheme or a more complex FAS algorithm. In [7–9] implicit methods were used, but no multigrid scheme was applied. Here, on the contrary, a multigrid scheme in conjunction with segregated relaxation of individual variables is adopted. By investigating the two solution paths in separate, namely the block-implicit single grid methodology [7–9] and the segregated CS multigrid algorithm, this contribution aims at broadening the general view on available techniques for obtaining faster fluid flow solutions. Here, it is interesting to point out that in [9,11] a block-implicit solver was used in conjunction with the multigrid method so that advantages in using independent artifices (multigrid or block solver) could not be assessed.

The numerical method also includes finite volume discretization [12], the SIMPLE pressure–velocity coupling [13] and the TDMA and Gauss–Seidel iterative solution algorithms [12]. Numerical validation is accomplished by considering a laminar flow between parallel plates and comparing the resulting velocity profile against its analytical solution. Further flow patterns are also presented. The overall algorithm performance and the need for an alternative (e.g. FAS) multigrid approach are discussed, based on the computational effort required, measured in CPU time.

2. Mathematical model

As no heat transfer is considered, the continuity equation and the Navier–Stokes equations govern the fluid flow. They express mass and momentum conservation principles respectively and for a steady-state condition in a two-dimension Cartesian coordinate frame they are written as [14]

$$\frac{\partial}{\partial x}(\rho U) + \frac{\partial}{\partial y}(\rho V) = 0 \quad (1)$$

$$\frac{\partial}{\partial x}(\rho U^2) + \frac{\partial}{\partial y}(\rho UV) = \frac{\partial}{\partial x} \left(\mu \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial U}{\partial y} \right) - \frac{\partial P}{\partial x} + s_U \quad (2)$$

$$\frac{\partial}{\partial x}(\rho UV) + \frac{\partial}{\partial y}(\rho V^2) = \frac{\partial}{\partial x} \left(\mu \frac{\partial V}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial V}{\partial y} \right) - \frac{\partial P}{\partial y} + s_V \quad (3)$$

where

$$s_U = \frac{\partial}{\partial x} \left(\mu \frac{\partial U}{\partial x} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial V}{\partial x} \right) \quad \text{and} \quad s_V = \frac{\partial}{\partial x} \left(\mu \frac{\partial U}{\partial y} \right) + \frac{\partial}{\partial y} \left(\mu \frac{\partial V}{\partial y} \right)$$

are the viscous source terms. For the cases studied, ρ and μ are assumed to be constant. By the continuity equation (Eq. (1)) it can be shown that $s_U = s_V = 0$ under these conditions.

The physical phenomena described by the above transport equations contain similar terms. Accordingly, it is possible to express them under a general form, i.e.,

$$\frac{\partial}{\partial x} (\rho U \phi) + \frac{\partial}{\partial y} (\rho V \phi) = \frac{\partial}{\partial x} \left(\Gamma_\phi \frac{\partial \phi}{\partial x} \right) + \frac{\partial}{\partial y} \left(\Gamma_\phi \frac{\partial \phi}{\partial y} \right) + S_\phi \tag{4}$$

where ϕ stands for any the flow variable, Γ_ϕ denotes the corresponding diffusive coefficient and S_ϕ identifies the respective source term. Since ρ is known and constant, Eq. (4) is numerically solved only for the velocity components, i.e., $\phi = U, V$. In this case, $\Gamma_\phi = \mu$ and S_ϕ includes the corresponding pressure gradient.

3. Numerical method description

3.1. Finite volume discretization

The solution domain is divided into a number of rectangular different sized control volumes (CV), resulting in a structured orthogonal non-uniform mesh. Grid points are located according to a cell-centered scheme and variables are stored in a collocated arrangement [7]. A typical CV is sketched in Fig. 1. Differential equations are discretized by double integrating Eq. (4) over the CV.

Discretization of convective terms results in

$$C_e \phi_e - C_w \phi_w + C_n \phi_n - C_s \phi_s \tag{5}$$

where, for instance,

$$C_e = (\rho U)_e \delta y \tag{6}$$

is the convective mass flux through the CV east face. Analogous expressions hold for the remaining CV face fluxes [15–17].

The flux blended deferred correction scheme [18] (indicated here as CDS) is employed to perform nodal interpolation. In this scheme, interface quantities are approximated as linear combination of central differencing scheme (CDS) and upwind differencing scheme (UDS) values [7] according to

$$\phi_{\text{face}}^{\text{DCS}} = \lambda \phi_{\text{face}}^{\text{CDS}} + (1 - \lambda) \phi_{\text{face}}^{\text{UDS}} = \phi_{\text{face}}^{\text{UDS}} + \lambda (\phi_{\text{face}}^{\text{CDS}} - \phi_{\text{face}}^{\text{UDS}})^* \tag{7}$$

where the starred quantities in parenthesis (last equality) are numerical values from the previous iteration. The combination factor λ may vary from 0 (pure UDS) to 1 (pure CDS).

Discretization of diffusive terms results in

$$\left(\Gamma_\phi \frac{\partial \phi}{\partial x} \right)_e \delta y - \left(\Gamma_\phi \frac{\partial \phi}{\partial x} \right)_w \delta y + \left(\Gamma_\phi \frac{\partial \phi}{\partial y} \right)_n \delta x - \left(\Gamma_\phi \frac{\partial \phi}{\partial y} \right)_s \delta x \tag{8}$$

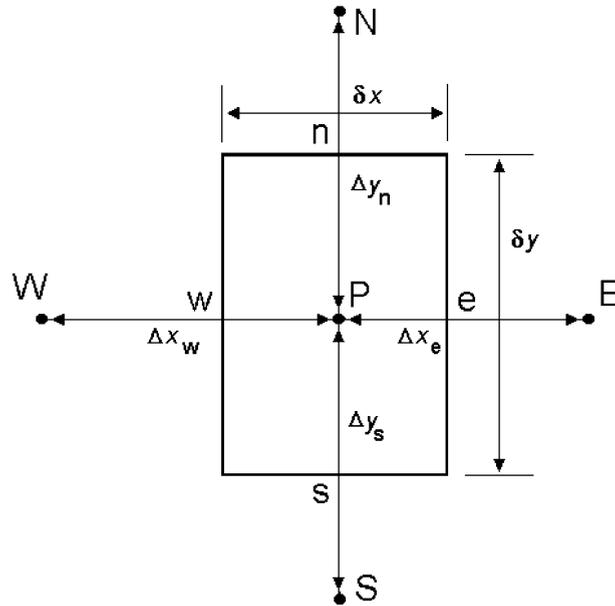


Fig. 1. Sketch of control volume to perform discretization.

Interface gradients are evaluate according to a CDS as

$$\left(\frac{\partial \phi}{\partial x}\right)_e = \frac{\phi_E - \phi_P}{\Delta x_e} \quad (9)$$

with similar expressions holding for the remaining CV interfaces [15–17].

The pressure gradient is employed to discretize the source term as, for example,

$$\int_{\delta V} S_\phi dv = - \int_{\delta V} \frac{\partial P}{\partial x} dv = -(P_e - P_w)\delta y \quad (10)$$

Neighboring grid point values are interpolated linearly to give interface pressure values,

$$P_e = (1 - f_{x,P})P_P + f_{x,P}P_E, \quad P_w = (1 - f_{x,W})P_W + f_{x,W}P_P$$

or extrapolated linearly at boundary faces,

$$P_e = P_P + (P_P - P_W)(1 - f_{x,W}), \quad P_w = P_P - (P_E - P_P)f_{x,P}$$

where the f s are interpolation factors. For the x -direction between nodes P and E , one has for example (see Fig. 1 for clarity),

$$f_{x,P} = \frac{x_e - x_P}{x_E - x_P} \quad (11)$$

Pressure gradients along the y -direction are discretized similarly [15–17].

Substitution of all approximate expressions for interface quantities and gradients, Eqs. (5)–(10), gives the final algebraic equation for a general grid node P

$$a_P \phi_P = a_E \phi_E + a_W \phi_W + a_N \phi_N + a_S \phi_S + b \quad (12)$$

For instance, the east interface coefficient results in,

$$a_E = \max[-C_e^*, 0] + D_e \tag{13}$$

The operator $\max[a - b]$ returns the greater of a and b . It is worth remembering that $\phi = U$ or V and $\Gamma_\phi = \mu$, so that

$$D_e = \frac{\mu_e^* \delta y}{\Delta x_e} \tag{14}$$

is the diffusive flux at the CV east interface. The remaining coefficients are defined similarly and they can be found in [4].

The discretized source term b contains contributions from the pressure gradient and from the CDS previous iteration values as

$$b = S_\phi \delta v + \lambda \left(\sum_{nb} a_{nb}^{DCS} \phi_{nb}^* - a_P^{DCS} \phi_P^* \right) \tag{15}$$

where for the east face, for example, the CDS coefficient reads,

$$a_E^{DCS} = -\max[-C_e^*, 0] - C_e^* f_{x,P} \tag{16}$$

Here, $S_\phi \delta v = (P_e - P_w) \delta y$ when $\phi = U$ and $S_\phi \delta v = (P_n - P_s) \delta x$ for $\phi = V$. The subscript “nb” indicates that summation is over the neighboring CVs.

3.2. Pressure–velocity coupling

Pressure–velocity coupling follows the SIMPLE—Semi-implicit method for pressure-linked equations algorithm [13]. The basic idea is to solve a pressure correction equation derived from the momentum and continuity equations. It can be shown that [13] the resulting pressure correction equation is

$$a_P P'_P = a_W P'_P + a_E P'_P + a_S P'_P + a_N P'_P - S_m \tag{17}$$

For example, the east coefficient is evaluated as

$$a_E = \frac{\rho_e \delta y^2}{(a_P)_e} \tag{18}$$

where $(a_P)_e$ is related to the momentum equation central coefficients from the grid nodes P and E . The remaining coefficients are defined in a similar fashion [15–17].

Oscillatory solutions caused by the collocated variable arrangement [7] are avoided by adopting a special interpolation scheme for the interface velocity values, which are needed for the evaluation of the mass imbalance source term S_m . For instance, the east face velocity is then calculated according to

$$U_e^* = (1 - f_{x,P}) \left[U_P^* + \frac{\delta y (P_e^* - P_w^*)}{(a_P)_P} \right] + f_{x,P} \left[U_E^* + \frac{\delta y (P_{ee}^* - P_e^*)}{(a_P)_E} \right] - \frac{\delta y (P_E^* - P_P^*)}{(1 - f_{x,P})(a_P)_P + f_{x,P}(a_P)_E}$$

After the pressure correction equation has been relaxed, the method suggests corrections on the pressure itself, for the velocity components and for the mass fluxes [13].

3.3. Under-relaxation

The equation set of the physical model is non-linear and coupled. Because of this and due to the fact that terms are neglected when the pressure correction equation is obtained, the SIMPLE algorithm tends to diverge if no under-relaxation is employed. Therefore, the pressure correction is given by

$$P_P = P_P^* + \xi_{P'} P_P' \quad (19)$$

while the velocity components are under-relaxed according to

$$\frac{a_P}{\xi_\phi} \phi_P = \sum_{nb} a_{nb} \phi_{nb} + b + (1 - \xi_\phi) \frac{a_P}{\xi_\phi} \phi_P^* \quad (20)$$

where $\xi_{P'}$, ξ_U and ξ_V are the relaxation factors. It is worth mentioning that these last two factors need not be equal.

3.4. Multigrid correction storage method

The convergence rate of the numerical solution is greatest at the beginning of calculations, slowing down sensibly as the iterative process goes on. This is because the smoothing algorithm is capable of reducing efficiently only those Fourier error components whose wavelengths are smaller than or comparable to the grid spacing. In other words, errors having different wavelength are supposed to be smoothed in a grid with adequate mesh spacing. The idea of the multigrid method is to cover a broader error spectrum by employing more than one grid level for the iterative process, which tends to accelerate convergence.

Assembling Eq. (12) for each CV results in an algebraic equation system of the form

$$\mathbf{A}_k \mathbf{\Phi}_k = \mathbf{b}_k \quad (21)$$

where \mathbf{A}_k is the matrix of coefficients, $\mathbf{\Phi}_k$ is the matrix of unknowns and \mathbf{b}_k are the source terms of Eq. (12). Subscript k refers to a given grid level, with $k = 1$ corresponding to the coarsest grid and $k = M$ to the finest. When only one mesh level is employed by the solution method, this subscript is dropped down.

Motivated by Jiang et al. [6], this work implements a multigrid correction storage method, although in general this formulation is not recommended to solve non-linear problems. In this formulation, coarse grid approximations are obtained for the correction of the flow quantity being numerically solved. In other words, $\mathbf{\Phi}_k$ stores the flow quantity itself when $k = M$, whereas $\mathbf{\Phi}_k$ stores its corrections for $k < M$. Moreover, restriction operations in the CS formulation are much simpler to implement than in its counterpart FAS. Before continuing, a word about problem linearization seems timely.

When the convective term are expanded in terms of correction and previous iteration values on has,

$$\mathbf{A}_k \mathbf{\Phi}_k = \mathbf{b}_k$$

After the equation system, Eq. (21), has been relaxed by a small number of (smoothing) iterations, an intermediate value $\bar{\Phi}_k$ is obtained along with its correction $\phi_k = \Phi_k - \bar{\Phi}_k$.

Defining the residue as

$$\mathbf{r}_k = \mathbf{b}_k - \mathbf{A}_k \bar{\Phi}_k \tag{22}$$

one shows [1–3] that the correction ϕ_k is the solution of

$$\mathbf{A}_k \phi_k = \mathbf{r}_k \tag{23}$$

which has the same form of Eq. (21). However, Eq. (23) is smoother [3] than Eq. (21) and it is approximated by a coarser grid equation

$$\mathbf{A}_{k-1} \phi_{k-1} = \mathbf{r}_{k-1}, \quad \mathbf{r}_{k-1} = I_k^{k-1} \mathbf{r}_k \tag{24}$$

The operator I_k^{k-1} , named restriction, transfers values from grid k to grid $k - 1$.

After being obtained, the coarse grid approximation for the correction $\bar{\phi}_{k-1}$ is taken back to the fine grid by the prolongation operator I_{k-1}^k

$$\bar{\phi}_k = I_{k-1}^k \bar{\phi}_{k-1} \tag{25}$$

in order to refine the intermediate value $\bar{\Phi}_k$

$$\Phi_k^{\text{new}} = \bar{\Phi}_k + \bar{\phi}_k \tag{26}$$

These numerical operations, Eqs. (21)–(26), are concatenated through all available k values (i.e. grid levels). The sequence as how the iteration process migrates from one grid level to another is what distinguishes the so-called V-cycle from the W-cycle [3]. In the present paper, only V-cycling strategies are considered.

The residue restriction needed in Eq. (24) is accomplished by summing up the residues corresponding to the equations of the four fine grid CVs that compose the coarse grid one, as sketched in Fig. 2. This operation can be mathematically expressed as

$$r_{k-1}^{IJ} = r_k^{ij} + r_k^{ij+1} + r_k^{i+1j} + r_k^{i+1j+1} \tag{27}$$

where indexes ij and IJ locate the CV in the fine grid and in the coarse grid respectively.

Matrix \mathbf{A}_k also undergoes restriction. Its coefficients contain convective and diffusive terms (Eq. (13)) and they need special treatment when changing grid level. Diffusive terms are recalculated after each grid level change since they depend on the grid geometry (Eq. (14)) namely, distances between nodes and CV dimensions. When restriction occurs, fine grid mass fluxes (convective terms) (Eq. (6)) are summed up at control volume faces in order to compose the corresponding coarse grid mass flux, as shown in Fig. 2 (west and south face are not pictured for clarity). Such restriction procedures are commonly used in the literature [4,19,21,23–27].

The prolongation operator I_{k-1}^k is numerically accomplished via bilinear interpolation [4,20,22–26]. In the present paper, this operator is implemented over a non-uniform grid. The basic idea is to use an auxiliary mesh between the fine and the coarse grids. This intermediate grid defines a new set of nodes which store values ϕ_{aux}^{ij} resulting from the application of the operator along one coordinate (say y) in the coarse grid. Then, the operator is again applied along the remaining coordinate (x) in order to obtain the fine grid values.

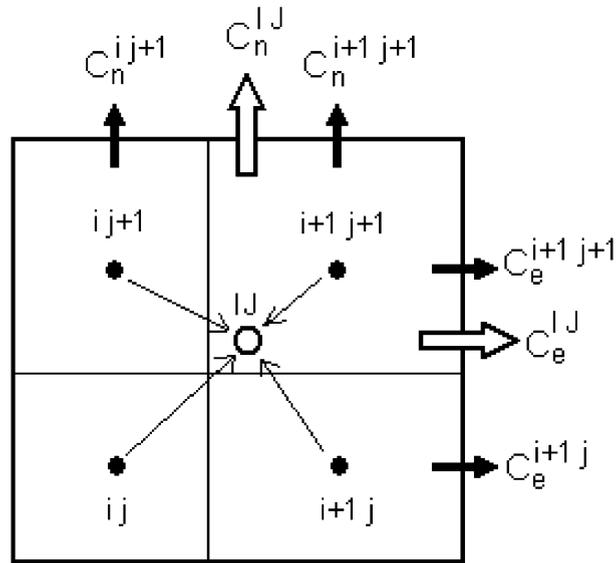


Fig. 2. Restriction procedures: mass flux and residue summation.

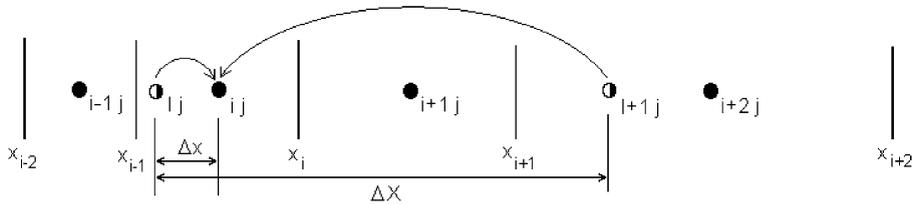


Fig. 3. Prolongation: recovering the fine grid from the auxiliary grid.

Considering this auxiliary mesh, fine grid values are obtained according to

$$\phi_k^{ij} = (1 - f)\phi_{aux}^{Ij} + f\phi_{aux}^{I+1j} \tag{28}$$

where the linear interpolation factor f , for the situation sketched in Fig. 3 (whose nomenclature follow that of Fig. 2), is given by

$$f = \frac{\Delta x}{\Delta X} = \frac{x_{i-1} - x_{i-2}}{x_{i+2} - x_{i-2}} \tag{29}$$

4. Results

The flow geometries and boundary conditions considered in this paper are sketched in Fig. 4a–d. Although there is no recirculation in the flow corresponding to Fig. 4a, it is helpful to numerically validate the solution method because its developed profile can be analytically described.

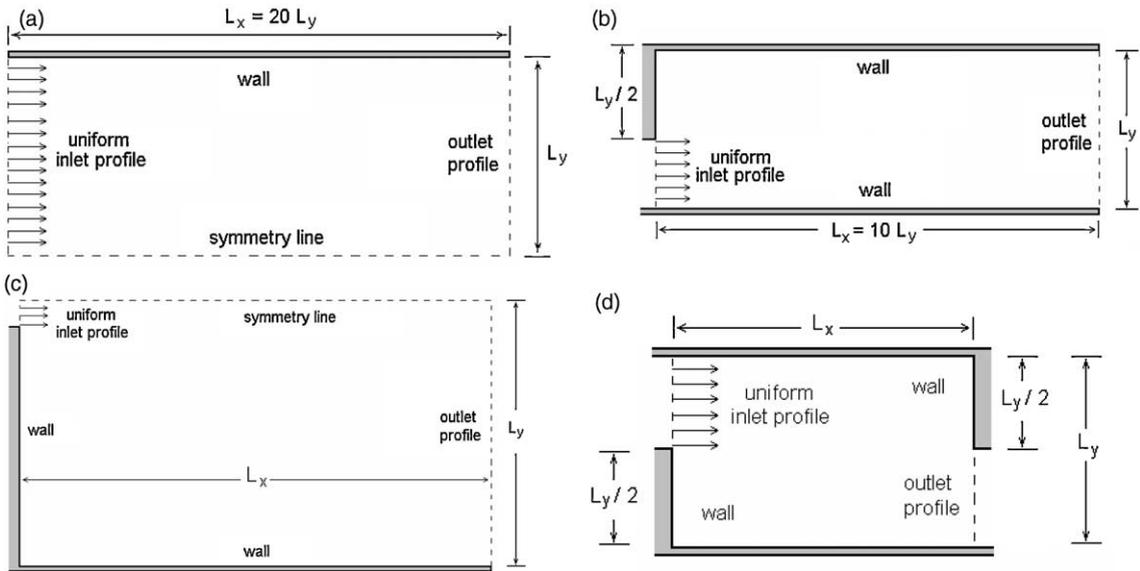


Fig. 4. Laminar flow geometries and boundary conditions considered: (a) parallel plates; (b) backward facing step; (c) confined jet; (d) rectangular tank.

Accordingly, the solution method was initially tested against the laminar flow between parallel plates (Fig. 4a) for which the U -developed profile is analytically expressed by [28]

$$U(y) = \frac{3}{2} U_0 \left[1 - \left(\frac{y}{L_y} \right)^2 \right] \tag{30}$$

The adopted values were $L_y = 0.05$ m, $L_x = 1.0$ m, $U_0 = 0.1$ m/s, $\rho = 1.0$ kg/m³, $\mu = 10^{-4}$ kg/m s, $\xi_U = 0.8$, $\xi_V = 0.6$ and $\xi_{p'} = 0.05$. The V-cycling strategy was fixed at one pre-smoothing and one post-smoothing iteration ($v^{pre} = v^{post} = 1$) and five coarsest grid iterations ($v^{cg} = 5$). Pure UDS ($\lambda = 0$) was employed as interpolation scheme and the number of CVs along the x and y directions were $NI = 160$ and $NJ = 32$, respectively. Although the pure upwind formulation was used for simplicity, the more general formulation (7) could have been used without difficulty. The reason for using $\lambda = 0$ (UDS) relied solely on the grounds of simplicity. For this reason, effects of using $\lambda \neq 0$ on convergence rates are here not reported. In Fig. 5, a good agreement between the four-grid numerical solution and Eq. (30) can be verified.

The velocity components and the pressure correction normalized residues reduction histories of the four-grid (4 g) and the one-grid (1 g) solution of this laminar flow are pictured in Fig. 6. Residues are calculated and normalized in the finest grid level according to

$$R_\phi = \sqrt{\sum_{ij} (r_M^{ij})^2 / (NI - 2)(NJ - 2)}, \quad r_M^{ij} = a_P \phi_P - \left(\sum_{nb} a_{nb} \phi_{nb} \right) \tag{31}$$

It is observed that the four-grid algorithm has better performance when compared to the single grid one, in terms of CPU time spent to run the program until convergence is achieved.

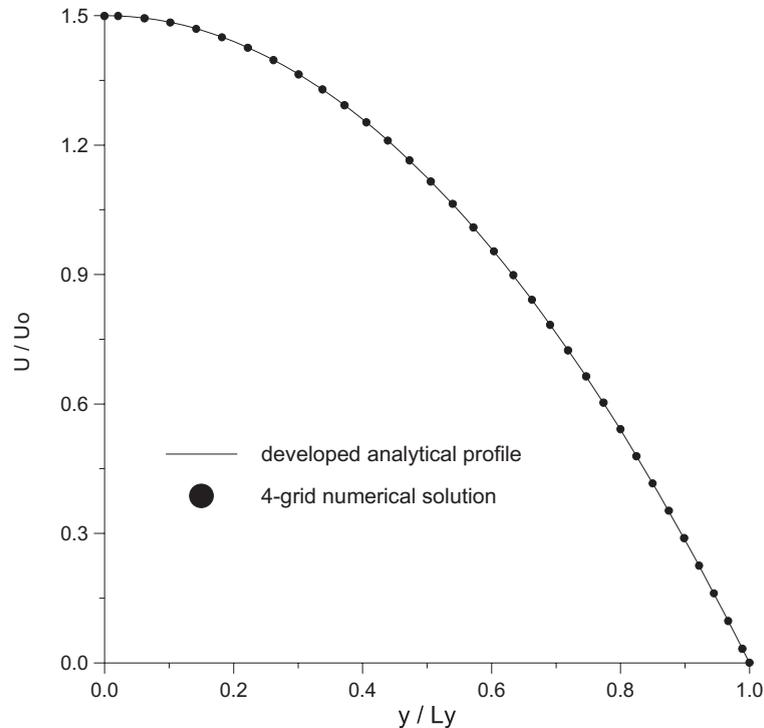


Fig. 5. Laminar flow between parallel plates: numerical validation of four-grid solution.

Table 1 helps to verify the grid refining (i.e. grid size) influence on the multigrid algorithm performance. The number of grid levels employed is M and the ratio of the one-grid computing time to the M -grid one is presented in the last column. One can think of this time ratio as a measure of the relative computational effort saving. It is worth noting that savings increase as grids become finer and such feature is what makes multigrid methods interesting.

Recirculating flows (Fig. 4b–d) are considered next. Table 2 summarizes all geometric and physical values concerned to the cases studied. Pure UDS ($\lambda = 0$) was again employed. Fig. 7a–c help to qualitatively visualize the flow field pattern obtained from multigrid numerical solutions. The recirculating regions can be seen clearly.

Residue levels and computational efforts (in terms of CPU time spent) from solutions using different numbers of grid levels (M) are displayed and compared in Table 3. The last column shows the ratio of the single-grid solution time (t_1) to the multigrid one (t_M). Again, this can be seen as a measure of the relative computational effort economy. Here, performance in using multigrid is measured in terms of time savings related to the use of a single grid. Other methods using the equivalence of fine grid iterations (work units) are also available but, at the end, this approach does not necessarily gives an evaluation of the actual time saved because equivalent iteration cycles, using one or multiple grids, consume different amounts of time. Also, other effects such as influence of Reynolds number is here not considered and an investigation on the effect of Peclet number when solving thermal problems can be found in Ref. [15]. It is verified that in general this economy increases as the number of grids used enlarges.

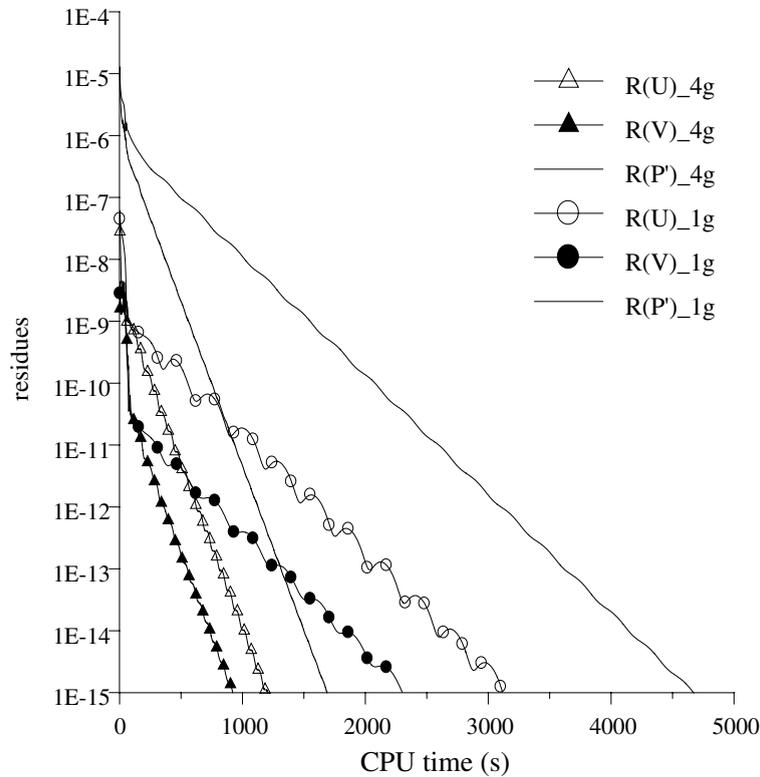


Fig. 6. Flow between parallel plates: U , V and P' residue reduction histories.

Table 1
Flow between parallel plates: grid-refining influence on the computational effort

$NI \times NJ$	M	$R_U (\times 10^{-18})$	$R_V (\times 10^{-20})$	$R_{P'} (\times 10^{16})$	t_M (s)	t_1/t_M
160 × 32	4	2.76	9.22	9.95	1688.4	2.768
	3	2.01	6.59	10.0	1874.3	2.493
	2	1.29	4.74	9.99	1912.1	2.444
	1	1.70	3.56	9.87	4672.8	1
240 × 32	4	3.53	21.0	9.99	2615.4	4.066
	3	2.71	11.6	9.93	3218.2	3.304
	2	1.67	8.11	10.0	3457.1	3.076
	1	2.34	5.85	10.0	10634.5	1
320 × 32	4	6.56	53.0	9.98	2919.0	4.474
	3	3.07	17.5	9.95	3251.6	4.017
	2	3.44	13.0	9.93	3626.1	3.602
	1	3.85	12.7	9.95	13060.8	1

However, there may be a limiting number of grid levels to be employed. For instance, the confined jet multigrid solution employed three grid levels at most because only four finest grid CVs are used to perform the inlet region and using a fourth grid level would make the left upper

Table 2
Summary of values adopted for the recirculating flows

Flow type	Backward facing step	Confined jet	Rectangular tank
L_x (m)	0.5	2.0	0.8
L_y (m)	0.05	0.5	0.6
U_0 (m/s)	0.2	0.01	0.01
ρ (kg/m ³)	1.0	1.0	1.0
μ (kg/m s)	10^{-4}	10^{-4}	10^{-4}
$NI \times NJ$	144×48	160×64	128×96
$\xi_U, \xi_V, \xi_{P'}$	0.8, 0.6, 0.03	0.8, 0.6, 0.01	0.8, 0.6, 0.01
$\nu^{pre}, \nu^{post}, \nu^{cg}$	1, 1, 5	1, 1, 1	1, 1, 1

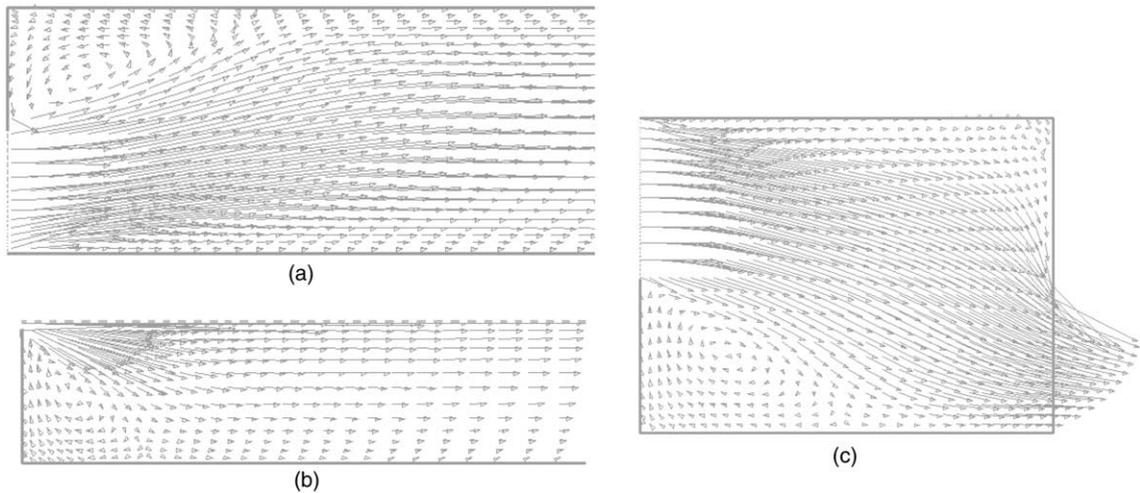


Fig. 7. Flow field visualization (multigrid solutions): (a) backward facing step; (b) confined jet; (c) rectangular tank.

Table 3
Recirculating flows: computational effort results

Flow type	M	$R_U (\times 10^{-18})$	$R_V (\times 10^{-19})$	$R_{P'} (\times 10^{-16})$	t_M (s)	t_1/t_M
Backward facing step	4	2.99	3.96	9.92	1608.3	2.962
	3	2.54	3.62	9.91	1797.0	2.562
	2	2.52	4.78	9.95	1765.0	2.608
	1	1.43	1.71	9.96	4603.0	1
Confined jet	3	0.194	0.389	9.13	1558.6	4.833
	2	0.997	1.37	9.60	3756.6	2.005
	1	1.85	2.20	9.05	7533.1	1
Rectangular tank	3	3.27	2.12	9.96	3887.8	2.098
	2	2.55	2.07	9.96	4703.8	1.733
	1	2.60	1.62	9.92	8158.0	1

corner CV to be requested by two distinct boundary conditions (namely, inlet flow and wall). Divergence was observed when a fourth grid level was used to solve the flow through rectangular tank problem. This may suggest that the corresponding coarsest grid (16×12) was not fine enough to handle properly the recirculating regions.

Results show that the application of the multigrid technique can speed up the iterative algorithm by values varying from 1.7 up to 4.8 times, depending on the flow geometry and the number of grids employed. Considering multigrid standards, these poor results suggest that a multigrid FAS formulation may be more adequate for circulating flow problems, unless block-implicit solutions are simultaneously applied [7–11]. As far as flow pattern is concerned, the convergence rate acceleration did not jeopardize qualitatively the expected results as pictured in Fig. 7.

5. Conclusions

The multigrid technique was applied to solve 2-D laminar recirculating flow problems. The numerical method included finite volume discretization and SIMPLE pressure–velocity coupling. Structured, orthogonal and non-uniform meshes were employed and the algebraic equation system was relaxed by TDMA. Multigrid was implemented in a correction storage formulation and only V-cycle strategy was considered. Solution method was numerically validated by an existing analytical profile. All results showed slightly better performance of multigrid solutions when compared to their single-grid counterpart, without jeopardizing qualitatively the flow field pattern. Convergence accelerations up to 4.8 times were observed, which suggest that a full approximation storage formulation may be more adequate when no additional convergence acceleration artifact is used.

Acknowledgements

Authors would like to thank CNPq (Brazil) for their financial support throughout this research, undertaken at the Computational Transport Phenomena Laboratory, Department of Energy, ITA, Brazil.

References

- [1] A. Brandt, Multi-level adaptive solutions to boundary-value problems, *Math. Comput.* 31 (138) (1977) 333–390.
- [2] K. Stüben, U. Trottenberg, *Multigrid methods*, in: *Lecture Notes in Mathematics*, vol. 960, Springer-Verlag, Berlin, 1982, pp. 1–76.
- [3] W. Hackbusch, *Multi-Grid Methods and Applications*, Springer-Verlag, Berlin, 1985.
- [4] J.A. Rabi, M.J.S. de Lemos, Optimization of convergence acceleration in multigrid numerical solutions of conductive–convective problems, *Appl. Math. Comput.* 124 (2) (2001) 215–226.
- [5] M.S. Mesquita, M.J.S. de Lemos, Optimal multigrid solutions of two-dimensional convection–conduction problems, *Appl. Math. Comput.* (in press).
- [6] Y. Jiang, C.P. Chen, P.K. Tucker, Multigrid solutions of unsteady Navier–Stokes equations using a pressure method, *Numer. Heat Transfer—Part A* 20 (1991) 81–93.

- [7] M.J.S. de Lemos, Flow and heat transfer in rectangular enclosures using a new block implicit numerical method, *Numer. Heat Transfer—Part B* 37 (4) (2000) 489–508.
- [8] M.J.S. de Lemos, A block-implicit method for numerical simulation of swirling flows in a model combustor, *Int. Commun. Heat Mass Transfer*, submitted for publication.
- [9] M.J.S. de Lemos, A block-implicit numerical procedure for simulation of buoyant swirling flows in a model furnace, *Int. J. Numer. Meth. Fluids*, submitted for publication.
- [10] S.P. Vanka, A calculation procedure for three-dimensional steady recirculating flows using multigrid methods, *Comput. Meth. Appl. Mech. Eng.* 55 (1986) 321–338.
- [11] S.P. Vanka, Block-implicit multigrid solution of Navier–Stokes equations in primitive variables, *J. Comput. Phys.* 65 (1986) 138–158.
- [12] S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corporation, New York, 1980.
- [13] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, *Int. J. Heat Mass Transfer* 15 (1972) 1787–1806.
- [14] R.B. Bird, W.E. Stewart, E.N. Lightfoot, *Transport Phenomena*, John Wiley & Sons, New York, 1960.
- [15] J.A. Rabi, M.J.S. de Lemos, The effects of Peclet number and cycling strategy on multigrid numerical solutions of convective–conductive problems, in: *Proc. 7th AIAA/ASME Joint Thermophysics & Heat Transfer Conference*, Paper AIAA-98-2584, Albuquerque, NM, 1998.
- [16] J.A. Rabi, M.J.S. de Lemos, Multigrid numerical solution of incompressible laminar recirculating flows, in: *Proc. ENCIT-98—7th Braz. Cong. Eng. Th. Sci.*, Rio de Janeiro, RJ, Brazil, vol. 2, 1998, pp. 915–920.
- [17] J.A. Rabi, M.J.S. de Lemos, Influence of Peclet number on numerical solution of two-dimensional convection–diffusion.
- [18] P.K. Khosla, S.G. Rubin, A diagonally dominant second-order accurate implicit scheme, *Comput. Fluids* 2 (12) (1974) 207.
- [19] B.R. Hutchinson, G.D. Raithby, A multigrid method based on the additive correction strategy, *Numer. Heat Transfer* 9 (5) (1996) 511–537.
- [20] S.P. Vanka, Block-implicit multigrid calculation of two-dimensional recirculating flows, *Comput. Meth. Appl. Mech. Eng.* 86 (1986) 29–48.
- [21] B.R. Hutchinson, P.F. Galpin, G.D. Raithby, Application of additive correction multigrid to the coupled fluid flow equations, *Numer. Heat Transfer* 13 (2) (1988) 133–147.
- [22] M.C. Thompson, J.H. Ferziger, An adaptive multigrid technique for the incompressible Navier–Stokes equations, *J. Comput. Phys.* 82 (1999) 94–121.
- [23] M. Peric, M. Rüger, G. Scheuerer, A finite volume multigrid method for calculating turbulent flows, in: *Seventh Symposium on Turbulent Shear Flows*, 7.3.1–7.3.6, Stanford University, 1989.
- [24] M. Hortmann, M. Peric, G. Scheuerer, Finite volume multigrid prediction of laminar convection: benchmark solutions, *Int. J. Numer. Meth. Fluids* 11 (1990) 189–207.
- [25] D.S. Joshi, S.P. Vanka, Multigrid calculation procedure for internal flows in complex geometries, *Numer. Heat Transfer—Part B* 20 (1991) 61–80.
- [26] L. Bai, N.K. Mitra, M.C. Fiebig, A. Kost, A multigrid method for predicting periodically fully developed flow, *Int. J. Numer. Meth. Fluids* 18 (1994) 843–852.
- [27] P.S. Sathyamurthy, S.V. Patankar, Block-correction-based multigrid method for fluid flow problems, *Numer. Heat Transfer—Part B* 25 (1994) 375–394.
- [28] R.K. Shah, A.L. London, *Laminar flow forced convection in ducts*, in: *Advances in Heat Transfer*, Academic Press, New York, 1978.