Available online at www.sciencedirect.com
ScienceDirect
Computers & Fluids 36 (2007) 445–454
computers & fluids
www.elsevier.com/locate/compfluid

A modified full multigrid algorithm for the Navier–Stokes equations

J. Yan [a,*], F. Thiele [a], L. Xue [b]

[a] Technische Universität Berlin, Hermann-Föttinger-Institute for Fluid Mechanics, Müller-Breslau-Strasse 8, D-10623 Berlin, Germany
[b] Department of Engineering Mechanics, Shanghai Jiaotong University, 200240 Shanghai, PR China

Received 29 March 2004; received in revised form 20 December 2005; accepted 20 January 2006
Available online 8 June 2006

Abstract

A modified full multigrid (FMG) method for the solution of the Navier–Stokes equations is presented. The method proposed is based on a V-cycle omitting the restriction procedure for dependent variables but retaining it for the residuals. This modification avoids possible mismatches between the mass fluxes and the restricted velocities as well as the turbulent viscosity and the turbulence quantities on the coarse grid. In addition, the pressure on the coarse grid can be constructed in the same way as the velocities. These features simplify the multigrid strategy and corresponding programming efforts. This algorithm is applied to accelerate the convergence of the solution of the Navier–Stokes equations for both laminar and high-Reynolds number turbulent flows. Numerical simulations of academic and practical engineering problems show that the modified algorithm is much more efficient than the FMG–FAS (Full Approximation Storage) method.

1. Introduction

The numerical solution of even laminar flows based on the Navier–Stokes equations requires a large number of grid points in order to capture complex flow phenomena and to obtain a grid-independent solution. For turbulent flows, Especially at high Reynolds number, even finer grid spacing is necessary in order to fully resolve turbulent shear layers. Among the iterative solution methods for the discretized Navier–Stokes equations, the SIMPLE algorithm [11] is probably the most widely used for the calculation of incompressible flows. Like most of the iterative methods used in existing solution algorithms, its convergence rate decreases rapidly with increasing grid density.

Among the acceleration techniques known, the multigrid method has been shown to be a very efficient and generally applicable procedure and has been applied to SIMPLE-based solution methods. Previous investigations have shown that it is efficient not only for laminar isothermal flows but also for the flows with strong heat transfer, e.g., buoyancy-driven flows in enclosures [3] and flows with radiative heat transfer [4]. In many cases, speed-up factors up to two orders were achieved, depending on the flow problem and the number of grid levels used. However, the application of the multigrid method to calculate turbulent flows [18,19,9] results in much lower speed-up factors.

Multigrid solution procedures for the Navier–Stokes equations mainly use FMG–FAS by Brandt [2], whereby the values of the variables and residuals from the fine grid are transferred to the next coarser level. The initial mass fluxes through the control volume faces of the coarse grid are restricted by the summation of the corresponding two or four fine-grid mass fluxes. However, this technique results in discrepancies between the mass fluxes and the restricted velocities, as well as the turbulent viscosity ν_t and the turbulence quantities (k and ϵ, etc.). This leads to lower stability and acceleration efficiency, or even divergence. To overcome this problem, a modified multigrid algorithm is proposed, in which only the residuals on the fine grid are transferred to the next coarser one. The initial quantities on the coarse grid are directly taken from the previous cycle. Therefore, no restriction is required for variables but only for residuals. This avoids the discrepancies and simplifies the multigrid strategy

* Corresponding author. Tel.: +49 30 314 23146; fax: +49 30 314 25405.
E-mail address: jianping.yan@cfd.tu-berlin.de (J. Yan).

0045-7930/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2006.01.017

and with it, the program structure. Additionally, the pressure-correction equation on the coarse grid can be constructed in the same way as those for the velocities.

The basic idea about the modified algorithm and its application to heat transfer has been published by Yan and Thiele [20]. In the present work, a general validation involving higher-order convection schemes, various geometries, different grid-types and turbulence models are given. Laminar flows (2D cavity, backward-facing step, 3D curved duct) as well as turbulent flows (2D model hill, 3D curved duct) are considered. The performance of this algorithm is compared with that of the FMG–FAS method.

It should be mentioned that the modified algorithm can be considered as an application in flow simulation of the nonlinear multigrid method (NLMG) of Hackbusch [6] which has since been discussed in the book by Trottenberg et al. [14].

2. Multigrid algorithm

Details of multigrid methods can be found in the book by Hackbusch [6]. Here, a brief summary of the features of the modified algorithm is given.

2.1. Two-grid algorithm

The basic multigrid features can be explained by the two-grid algorithm, which consists of pre-smoothing, coarse-grid correction, prolongation and post-smoothing.

During the *pre-smoothing*, v_1 relaxation sweeps of the discretized equation $L^h(\phi^h) = f^h$ on the fine grid h lead to an approximate solution $\hat{\phi}^h$. In general, this solution is not converged and contains a residual R^h:

$$L^h(\hat{\phi}^h) = f^h + R^h, \tag{1}$$

where L^h denotes the appropriate operator based on $\hat{\phi}^h$. In the FAS the approximate solution $\hat{\phi}$ and residual R^h are restricted to the coarse grid H by

$$\tilde{\phi}^H = [\widehat{I}_h^H]\hat{\phi}^h, \quad R^H = [I_h^H]R^h, \tag{2}$$

where $[\widehat{I}_h^H]$ and $[I_h^H]$ denote the appropriate restriction operators for the variable and the residual, respectively. Using $\tilde{\phi}$ as the starting solution, the approximate solution $\hat{\phi}$ on the coarse grid H is calculated by

$$L^H(\hat{\phi}^H) = f^H = L^H(\tilde{\phi}^H) - R^H, \tag{3}$$

from which the coarse-grid correction is evaluated:

$$\delta\phi^H = \hat{\phi}^H - \tilde{\phi}^H. \tag{4}$$

In the *modified algorithm*, the initial quantity $\tilde{\phi}^H$ is taken from the previous cycle instead of using the restriction from the fine grid. The restriction of the residual remains unaltered.

Once the approximate solution is obtained, the coarse-grid correction can be calculated by Eq. (4). Since $\hat{\phi}^H$ is initiated by $\tilde{\phi}^H$, when the residual on the fine grid vanishes, Eq. (3) is identically satisfied and no correction to the fine-grid solution is produced.

After the coarse-grid correction $\delta\phi^H$ is obtained, it will be prolongated to the fine grid and added to the previous approximate solution:

$$\delta\phi^h = [I_H^h]\delta\phi^H, \quad \tilde{\phi}^h = \hat{\phi}^h + \delta\phi^h. \tag{5}$$

Here, $[I_H^h]$ is an appropriate prolongation operator from the coarse grid to the fine one.

In the *post-smoothing*, with $\tilde{\phi}^h$ as starting solution v_2 relaxation sweeps of the discretized equation $L^h(\phi^h) = f^h$ on the fine grid are carried out to obtain an improved approximate solution.

2.2. Modified FMG with V-cycle

The two-grid algorithm procedure described above can be further generalized to a sequence of grids using a recursion procedure, which leads to the modified FMG with V-cycle as shown in Fig. 1. After a convergent solution on the coarsest grid is obtained, it is then extrapolated to the next finer grid and serves there as the initialization. When the solution on this grid is convergent, it is transferred to the next finer grid and so on, until the finest grid is reached. With FAS, the starting solution on the coarse

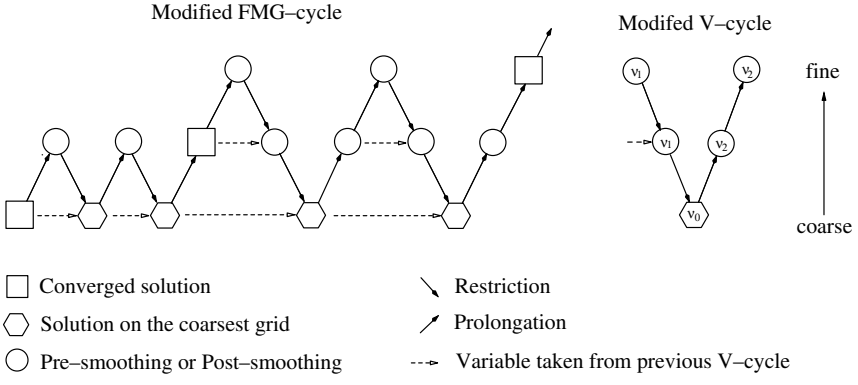

Fig. 1. Modified FMG-algorithm and V-cycle.

grid is restricted from the finer grid, but in the modified algorithm it is taken from the previous cycle.

In Fig. 1, the arrow $\searrow$ symbolizes the residual transfer from the fine grid to the next coarser one. The arrow $\nearrow$ holds for the prolongation of the coarse-grid correction. The arrow $\rightarrow$ symbolizes the processes in which the variable is taken from the previous cycle. v_0 is the number of relaxation sweeps on the coarsest grid. Usually the solution on the coarsest grid requires more relaxation sweeps than the fine grid, as the coarsest grid correction should be as exact as possible.

2.3. Multigrid algorithm for the Navier–Stokes equations

The numerical simulation of turbulent flows is based on the Reynolds-averaged Navier–Stokes equations. A description of the SIMPLE algorithm, finite-volume discretization and turbulence closures can be found in the book by Ferziger and Perić [5]. For the purpose of conveying the essential features of the multigrid algorithm, the coarse grid control volume is obtained by merging four fine grid cells together in 2D cases, while it is constructed from eight fine grid cells in 3D cases. As a consequence, the residuals are restricted to the coarse grid by the summation of the corresponding fine grid residuals. In the following text the method description focuses only on the two-dimensional case.

Applying the Gaussian divergence theorem, the integration of the governing equations over a finite volume such as that shown in Fig. 2 results in a balance of convective and diffusive cell-face fluxes and a volume-integrated net source. Appropriate approximations for the convection fluxes through the cell faces using the QUICK [8] or MUSCL [7] as well as central differences for the diffusive fluxes leads to a weighted-average formula in the following form:

$$-a_P\phi_P + \sum_{\mathrm{nb}} a_{\mathrm{nb}}\phi_{\mathrm{nb}} + S_\phi = f_\phi. \tag{6}$$

Here, nb denotes neighbours of the node P involved in the polynomials, and ϕ stands for either the velocity components u_i, pressure correction p' or turbulence quantities e.g., k, ϵ. The definitions for the coefficients a_P and a_{nb} can be also found in [5]. For clarity, Eq. (6) for the linearized momentum equations is rewritten as follows:

$$L\{u_i\} + D_i(p) + S\{u_i\} = f\{u_i\}, \tag{7}$$

where $L\{u_i\}$ represents the quantity $-a_P\phi_P + \sum a_{\mathrm{nb}}\phi_{\mathrm{nb}}$. $D_i(p)$ and $S\{u_i\}$ denote the term for the pressure gradient in the i-direction and the source term, respectively. $f\{u_i\}$ is the right-hand side term, which becomes zero on the finest grid while it may be non-zero on any coarse grid.

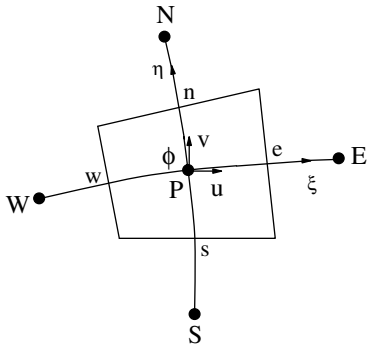

Fig. 2. Finite volume and storage arrangement.

After performing v_1 outer iteration sweeps with the SIMPLE algorithm on the fine grid h, the intermediate approximations of velocities $\hat{u}_i^h$ and pressure $\hat{p}^h$ are obtained. In general, the intermediate solution is not yet converged and the residuals $R^h\{\hat{u}_i^h\}$ do not vanish

$$L^h\{\hat{u}_i^h\} + D_i^h(\hat{p}^h) + S^h\{\hat{u}_i^h\} = f^h\{u_i\} + R^h\{\hat{u}_i^h\}. \tag{8}$$

In order to obtain the converged solution, the corrections for velocity components and pressure are sought, namely,

$$u_i^h = \hat{u}_i^h + \delta u_i^h, \quad p^h = \hat{p}^h + \delta p^h, \tag{9}$$

so that $\overline{R}^h\{u_i\}$ vanishes:

$$L^h\{u_i^h\} + D_i^h(p) + S^h\{u_i^h\} = f^h\{u_i\}. \tag{10}$$

Subtracting Eq. (8) from Eq. (10) yields a relationship which serves as a basis for multigrid coupling:

$$L^h\{u_i^h\} + D_i^h(p^h) + S^h\{u_i^h\} = \underbrace{L^h\{\hat{u}_i^h\} + S^h\{\hat{u}_i^h\} + D_i^h\{\hat{p}^h\} - R^h\{\hat{u}_i^h\}}_{\text{constant}}. \tag{11}$$

For the scalar quantities ϕ such as the turbulent kinetic energy k and its dissipation rate ϵ, the analogous equation can be written as

$$L^h\{\phi^h\} + S\{\phi^h\} = \underbrace{L\{\hat{\phi}^h\} + S\{\hat{\phi}^h\} - R^h\{\hat{\phi}^h\}}_{\text{constant}}. \tag{12}$$

Applying the multigrid method, the corrections can be calculated on the coarse grid H. The equations now read for the velocity component u_i

$$L^H\{\hat{u}_i^H\} + D_i^H(\hat{p}^H) + S^H\{\hat{u}_i^H\} = f^H\{u_i\} = \underbrace{L^H\{\tilde{u}_i^H\} + D_i^H\{\tilde{p}^H\} + S^H\{\tilde{u}_i^H\} - [I_h^H]R^h\{\hat{u}_i^h\}}_{\text{constant}} \tag{13}$$

and for the scalar ϕ ($\phi = k$, ϵ, etc.)

$$L^H\{\hat{\phi}^H\} + S^H\{\hat{\phi}^H\} = f^H\{\phi\} = \underbrace{L^H\{\tilde{\phi}^H\} + S^H\{\tilde{\phi}^H\} - [I_h^H]R^h\{\hat{\phi}^h\}}_{\text{constant}}. \tag{14}$$

In the above equations, the variables $\tilde{u}_i^H$, $\tilde{p}^H$ and $\tilde{\phi}^H$ are the initial quantities for velocity components, pressure and the scalar quantities while $\hat{u}_i^H$, $\hat{p}^H$ and $\hat{\phi}^H$ denote their intermediate approximations on the coarse grid, respectively. In the FAS algorithm $\tilde{u}_i^H$, $\tilde{p}^H$ and $\tilde{\phi}^H$ are restricted from the finer grid, but in the modified algorithm they are taken directly from the previous cycle. On the right-hand side, the terms $[I_h^H]R^h\{\hat{u}_i^h\}$ and $[I_h^H]R^h\{\hat{\phi}^h\}$ are the restricted residuals

obtained from the fine grid, whereas the other terms are calculated on the coarse grid based on the starting quantities. The right-hand side terms remain unchanged within the iterative process on the coarse grid.

The handling of the pressure-correction equation within the multigrid concept deserves more attention. The pressure-correction equation in the SIMPLE algorithm can be written in a general form as

$$L\{p'\} + Q_{\mathrm{m}} = f\{p'\}, \tag{15}$$

where the right-hand side term $f\{p'\}$ is zero on the finest grid, but may be non-zero on the coarse grid. Q_m is the mass imbalances as follows:

$$Q_{\mathrm{m}} = \dot{m}_{\mathrm{e}} + \dot{m}_{\mathrm{w}} + \dot{m}_{\mathrm{n}} + \dot{m}_{\mathrm{s}}. \tag{16}$$

Here, the suffixes w, s, e and n refer to the west, south, east, and north faces of the 2D control volume, respectively. $\dot{m}$ denotes the mass flux through the volume faces. Once the pressure corrections are obtained, they are used to correct the velocity components and the mass fluxes. This leads to a new mass imbalance Q^h_{m}, which is defined as the residual $R^h\{p'\}$:

$$R^h\{p'\} = Q^h_{\mathrm{m}}. \tag{17}$$

The pressure-correction equation on the coarse grid can then be expressed as

$$L\{p'^H\} + Q^H_{\mathrm{m}} = \underbrace{\widetilde{Q}^H_{\mathrm{m}} - [I^H_h]R^h\{p'\}}_{\text{constant}} = f^H\{p'\}, \tag{18}$$

where $\widetilde{Q}^H_{\mathrm{m}}$ is the mass imbalance taken from the previous cycle.

As Eq. (18) is solved with the right-hand side term $f^H\{p'\}$ on the coarse grid H, the global mass flux should be corrected with the sum of $f^H\{p'\}$ through the entire domain, i.e., $\sum f^H\{p'\}$. In cases of open through-flow geometries, we assume that the sum of mass fluxes at the inlet boundary is equal to the sum of mass fluxes at the outlet on the finest grid. On the coarse grid, $\sum f^H\{p'\}$ should be added to the sum of inlet mass fluxes to satisfy the global conservation condition.

After performing ν_1 SIMPLE-relaxation sweeps with the initial quantities $\tilde{u}^H_i$, $\tilde{p}^H$ and $\tilde{\phi}^H$, the coarse grid solution including the coarse-grid pressure is obtained. Therefore, the coarse-grid corrections can be calculated by

$$\delta u^H_i = \hat{u}^H_i - \tilde{u}^H_i, \quad \delta p^H = \hat{p}^H - \tilde{p}^H, \quad \delta\phi^H = \hat{\phi}^H - \tilde{\phi}^H, \tag{19}$$

which are then prolongated to the fine grid using

$$\delta u^h_i = [I^h_H]\delta u^H_i, \quad \delta p^h_i = [I^h_H]\delta p^H, \quad \delta\phi^h = [I^h_H]\delta\phi^H \tag{20}$$

and added to the previous intermediate approximations:

$$\tilde{u}^h_i = \hat{u}^h_i + \lambda\delta u^h_i, \quad \tilde{p}^h = \hat{p}^h + \lambda\delta p^h, \quad \tilde{\phi}^h = \hat{\phi}^h + \lambda\delta\phi^h. \tag{21}$$

Here, λ is an under-relaxation factor, $0 < \lambda \leqslant 1$, which has been found useful in some cases of complex three-dimensional simulations.

It should be mentioned that the coarse-grid correction δp^H for pressure is calculated directly by Eq. (19), whereby the $\hat{p}$ and $\tilde{p}$ denote the approximate pressure and its initial value, respectively. This procedure differs from those described in [9,12], where the so-called correction of pressure correction ($\delta p'$) is solved. Moreover, the corrections to the mass fluxes on the fine grid are evaluated from the velocity corrections δu^h_i and added to the mass fluxes obtained previously.

The boundary conditions on all grid levels are treated in the same manner as the in any single grid algorithm.

In general, the calculations of turbulent flows require a higher grid density to capture the large variations associated with strong turbulent shear. They require special attention in the treatment of the turbulent viscosity ν_{t} and the source terms:

- The turbulent viscosity ν_{t} is fixed on the coarse grid. This means, once the converged solution on the coarse grid is obtained, ν_{t} remains unchanged in the next multigrid cycle.
- The right-hand side term $f^H\{\phi\}$ in Eq. (14) is evaluated as follows: if $f^H\{\phi\} > 0$, it is included in the right-hand-side term $\widehat{S}^H\{\hat{\phi}^H\}$; otherwise it is included in the diagonal coefficient via a division by ϕ^H. This commonly-used implementation method enhances the diagonal dominance of the solution matrix, thereby increasing the robustness.
- As the coarse-grid corrections could be negative, it is possible that the turbulence quantities can assume unphysical negative values. To prevent this, a realizability constraint is applied, whereby a positive value is enforced using $\tilde{\phi}^h = |\hat{\phi}^h + \lambda[I^h_H]\delta\phi^H|$.

Experience has demonstrated that such treatment results in a high stability of the overall algorithm.

3. Application to laminar and turbulent flows

The multigrid algorithms described above have been implemented in a general-purpose computer code to solve fluid flow and heat transfer in complex geometries. The program is based on a non-orthogonal, co-located grid arrangement and the SIMPLE algorithm. The momentum interpolation proposed by Rhie and Chow [13] is employed for calculating the cell-face mass fluxes to avoid pressure oscillations.

For reliable validation of the multigrid method, this work focuses on higher-order convection schemes (QUICK, MUSCL) as well as on various grids (non-uniform, Cartesian, highly skewed, non-orthogonal). Their influences are evaluated for 2D flow configurations such as the driven cavity, backward-facing step and model hill as well as 3D cases. For comparison and to demonstrate the convergence behaviour of the present multigrid procedure in the simulation of turbulent flows, the k–ϵ turbulence model is applied in both the high-*Re* formulation with wall function and the

low-*Re* formulation of Lien and Leschziner [10]. The later is referred to LL k–ϵ model in the following.

In all examples, a solution is considered to be convergent when the mass, momentum, and turbulence quantity residuals on the finest grid fall below 10^{-3} corresponding to their scales. In the case of open through-flow geometries, these scales refer to the inlet fluxes, whereas for closed cavities the lid velocities and cavity depth are used as reference scales. The under-relaxation factors are set to 0.7 for the velocity components, 0.3 for pressure and 0.7 for turbulence quantities, respectively. The smoothing numbers v_1, v_2 and v_0 are fixed to 3, 2 and 8, respectively. All 2D calculations are run on a Silicon Graphics workstation, and a Cray-T3E has been used for the 3D flow problems. The multigrid performance is demonstrated by CPU-time in (s) and speed-up factor. The CPU-time required for the single grid (SG) calculation serves as reference for the speed-up factors. To validate the performance of the modified multigrid algorithm, all 2D cases are additionally calculated with the FMG–FAS for comparison.

3.1. Lid-driven flow in a 2D square cavity

Due to its established role as a test of computational performance, the lid-driven flow in a 2D square cavity was selected to evaluate the multigrid methods. A sequence of five non-uniform grids ranging from 16×16 to 256×256 cells is adopted to resolve flow features.

Computations have been carried out for $Re = 100$, 400 and 1000 with the MUSCL and QUICK schemes. Table 1 shows the CPU-time for SG calculation. The results indicate that the QUICK scheme needs less CPU-time than the MUSCL scheme to obtain a convergent solution. At $Re = 100$ the QUICK scheme needs about 10% less CPU-time than the MUSCL scheme. At $Re = 400$ and 1000 on the fine grids, the difference in CPU-time reduces to less than 5%. The SG procedure requires more CPU-time with increasing Reynolds number on the coarse grids, while on the finer grid the opposite trend is observed. This is in agreement with the observations in [4,9,12].

With application of the multigrid technique, the results in Table 2 show a remarkable gain in the speed-up factor when the grid density increases. In general, the speed-up factor declines slightly as Reynolds number grows, which is consistent with the findings in [16,9]. The modified FMG procedure underlines its potential for the finer grids with 128×128 and 256×256 cells where the speed-up factors are about 40% higher compared to the FMG–FAS in this work. For comparison, the speed-up factors from [9] are listed in this table. On the coarse grid the present

Table 1
CPU-time for SG on various grid level for laminar flow in 2D square cavity

Grid	MUSCL			QUICK		
	$Re = 100$	$Re = 400$	$Re = 1000$	$Re = 100$	$Re = 400$	$Re = 1000$
16^2	0.12	0.17	0.20	0.11	0.16	0.180
32^2	1.24	1.40	1.44	1.08	1.20	1.35
64^2	21.99	20.71	19.72	19.86	18.72	18.26
128^2	598.29	520.07	468.73	567.45	492.23	444.68
256^2	10994.89	8454.69	7581.09	9689.09	8204.58	7573.64

Table 2
Comparison of speed-up factors in terms of multigrid algorithm, grid level with QUICK scheme for laminar flow in 2D square cavity

Grid	Mod. FMG			FMG–FAS			[9]		
	$Re = 100$	400	1000	100	400	1000	100	400	1000
16^2	1.0	1.0	1.0	1.0	1.0	1.0	1.0	0.9	0.8
32^2	2.9	2.2	1.8	2.8	2.1	1.6	3.9	2.5	1.8
64^2	12.2	11.0	9.3	11.1	8.9	6.6	19.9	10.6	6.2
128^2	60.8	65.8	59.9	49.1	49.6	40.9	43.7	26.5	16.8
256^2	228.6	288.8	283.5	180.0	187.4	169.0	–	–	–

Table 3
Comparison of the maximum negative velocity on the vertical cavity centerline by the modified FMG, QUICK and MUSCL for laminar flow in 2D square cavity

Grid	MUSCL		QUICK		HYBRID [16]		
	$Re = 10^2$	$Re = 10^3$	$Re = 10^2$	$Re = 10^3$	Grid	$Re = 10^2$	$Re = 10^3$
16^2	−0.193	−0.250	−0.191	−0.260	–	–	–
32^2	−0.206	−0.322	−0.206	−0.321	40^2	−0.202	−0.258
64^2	−0.211	−0.363	−0.211	−0.367	80^2	−0.209	−0.338
128^2	−0.212	−0.381	−0.212	−0.379	160^2	−0.212	−0.381
256^2	−0.213	−0.382	−0.213	−0.380	320^2	−0.213	−0.387

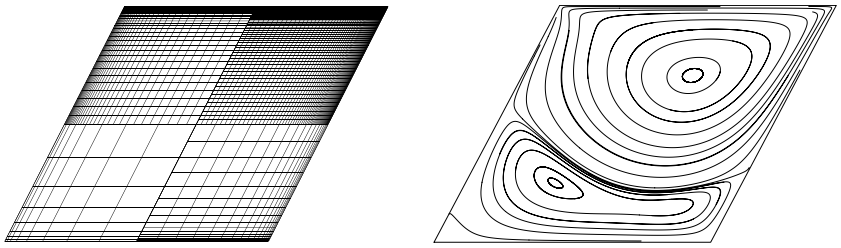

Fig. 3. (a) Geometry and four grids; (b) streamlines for laminar flow at $Re = 400$ in a 2D skewed cavity.

speed-up factors of FMG–FAS is a little lower, but with four grid levels they are much higher than those in [9].

Table 3 shows that the velocities along the vertical cavity centerline are in agreement with Vanka's results [16] calculated with a HYBRID scheme. This indicates that higher-order convection schemes yield results closer to the "grid-independent" values with fewer grid cells.

3.2. Lid-driven laminar flow in 2D skewed cavity

Many engineering applications require highly skewed grids which may influence the solution efficiency. Therefore, the skewness effect is investigated for a cavity flow with 27° side wall angle. The non-uniform grids, ranging from 16^2 to 128^2 cells, and the stream lines at $Re = 400$ are displayed in Fig. 3.

Table 4 shows the speed-up dependence on the Reynolds number, convection schemes and multigrid algorithms. The results indicate that the modified FMG is about 40% more efficient than the FMG–FAS. This behaviour is in agreement with the results for square cavity flow with orthogonal grid. Hence, the grid skewness has no clear influence on the efficiency of the modified FMG. Compared to the investigation by Lien and Leschziner [9], the speed-up factor is up to about two times higher for the FMG–FAS.

Table 4
Comparison of speed-up factors with respect to multigrid algorithm, grid level with QUICK for laminar flow in a 2D skewed cavity

Grid	Mod. FMG		FMG–FAS		[9]	
	$Re = 100$	$Re = 400$	$Re = 100$	$Re = 400$	$Re = 100$	$Re = 400$
16^2	1.00	1.00	1.00	1.00	–	–
32^2	3.28	2.19	2.98	2.05	3.3	2.1
64^2	13.53	11.27	11.76	8.75	15.5	11.7
128^2	61.32	70.53	49.36	50.37	27.2	31.3

Table 5
Comparison of CPU-time and speed-up factor for backward-facing step flow at $Re = 100$ for different multigrid procedures and grid levels with QUICK scheme

Grid	CPU-time			Speed-up		
	SG	Mod. FMG	FMG–FAS	Mod. FMG	FMG–FAS	[12]
40×20	0.90	0.90	0.90	1.00	1.00	–
80×40	12.86	2.54	4.05	5.06	3.17	–
160×80	360.27	9.20	19.71	38.92	18.27	9.85
320×160	6840.36	40.79	–	167.68	–	–

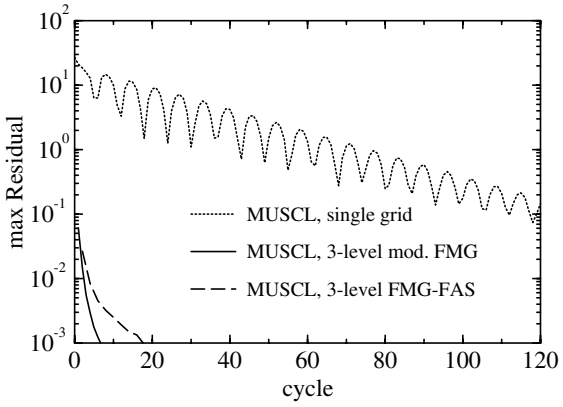

Fig. 5. Convergence histories in terms of cycle number for backward-facing step flow, 160×80 cells on the finest grid.

Table 6
CPU-time for the SG and multigrid algorithms and turbulence models on various grids for turbulent flow over a model hill

Grid	k–ϵ			LL k–ϵ		
	SG	Mod. FMG	FMG–FAS	SG	Mod. FMG	FMG–FAS
50×20	3.2	3.2	3.2	4.5	4.5	4.5
100×40	30.6	15.1	23.6	47.5	14.9	19.6
200×80	772.8	78.3	164.4	923.5	85.9	110.5
400×160	12468.9	324.6	1029.5	14361.0	396.1	774.6

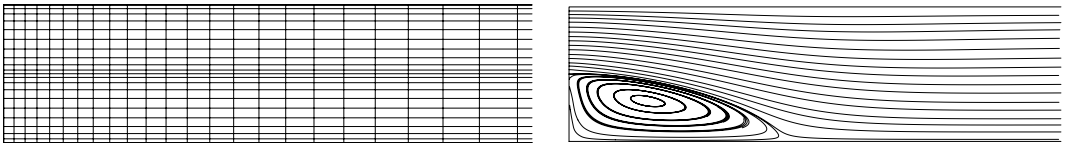

Fig. 4. The coarsest grid and streamlines for backward-facing step flow at $Re = 100$.

Table 7
Speed-up factors in terms of multigrid algorithm, turbulence model and grid level, MUSCL scheme for turbulent flow over a model hill

Grid	k–ϵ		LL k–ϵ	
	Mod. FMG	FMG–FAS	Mod. FMG	FMG–FAS
50×20	1.00	1.00	1.00	1.00
100×40	1.93	1.29	3.18	2.41
200×80	9.87	4.70	10.75	8.36
400×160	38.41	12.11	36.25	18.54

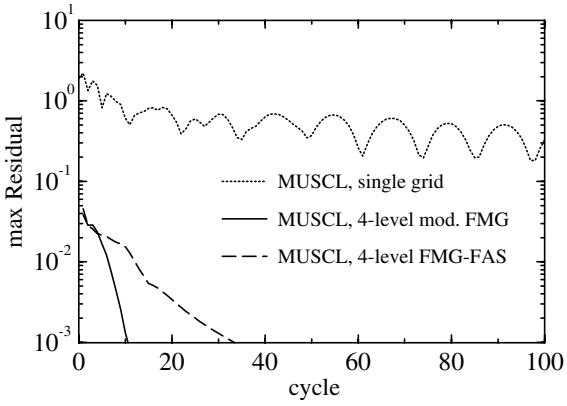

Fig. 6. Convergence histories in terms of cycle number for turbulent flow over a model hill, FMG with the LL k–ϵ model.

3.3. 2D backward-facing step flow

In this configuration, the inlet is half of the total channel height H while the length of the computational domain is

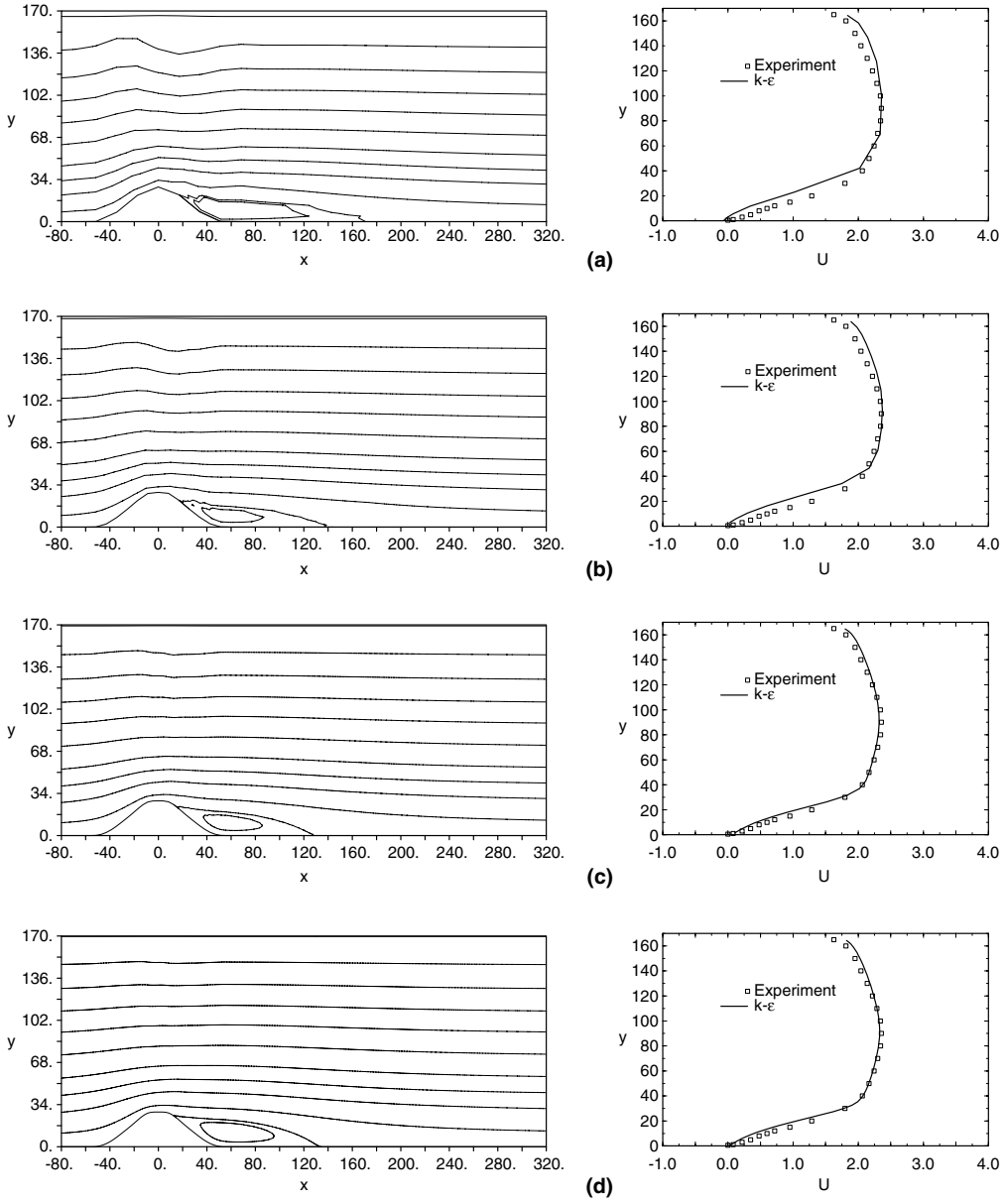

Fig. 7. Streamlines and U-profile comparison at $x = 0$ for turbulent flow over a model hill on various grids: (a) 25×10; (b) 50×20; (c) 100×40; (d) 200×80.

$10H$, to assure the Neumann condition at the outlet. A parabolic velocity profile is prescribed at the inlet. Non-uniform grids ranging from 40×20 to 160×80 cells are adopted, as seen in Fig. 4, where the coarsest grid and streamlines are displayed.

The computation for $Re = 100$, based on the mean velocity at the inlet and the channel height, has been carried out with the QUICK scheme. For comparison, the same under-relaxation factors used by [12] are applied, namely, 0.7 for velocities and 0.2 for pressure correction.

The results support those obtained from the cavity case. Here, the speed-up factor for the modified FMG is up to 80% higher than that in the work of Orth [12], where the FMG–FAS was used. With four grid levels, the modified FMG procedure displays its efficiency by the speed-up factor of 167 and less than 41 s CPU-time as shown in Table 5.

It should be mentioned that with the FMG–FAS the solution is not convergent using four grid levels. This is probably due to the discrepancies between the restricted velocities and initial mass fluxes on the coarse grid, for instance, at the inlet where the parabolic velocity profile is prescribed.

The convergence histories shown in Fig. 5 support this conclusion, as the modified FMG procedure without velocity restriction clearly demonstrates a stable and efficient convergence. Behind the step a large recirculation area is produced. The reattachment length is $3.20H$, which is in good agreement with [12].

3.4. Turbulent flow over a model hill

The turbulent flow over a model hill [1] at $Re = 6000$ based on the hill height is calculated using the standard k–ϵ model as well as the LL k–ϵ model. For validation, four non-uniform grids ranging from 50×20 to 400×160 cells are used. Two separate grid systems are adopted, with respect to the high-Re and low-Re turbulence model formulations.

Table 6 shows the CPU-time required for a convergent solution for the SG technique, the different multigrid algorithms and various grids. When using the multigrids, the coarsest grid level is 50×20 cells. Table 7 shows the corresponding speed-up factors. With four grid levels, the speed-up factors of 38.4 and 12.1 can be obtained for the modified FMG and the FMG–FAS algorithms, respectively.

Fig. 6 shows the convergence histories. As seen here, using the modified FMG, the maximal residual decreases faster than that of the FMG–FAS algorithm. The reason is most probably the non-matching between the flow field and the restricted turbulence quantities on the coarse grid for the FMG–FAS. In particular, the turbulent viscosity does not correspond to the turbulent kinetic energy and its dissipation rate. As seen from these results, the modified algorithm has better performance than the standard one, but the convergence rate is not as monotonic as that observed in the laminar cases. The stream function plots arising from the calculation with the k–ϵ model and the modified multigrid method, and the comparison of the numerical U-velocity profiles with experimental data at $x = 0$ on the various grid levels are given in Fig. 7. On the fine grid, the numerical results show reasonable agreement for the very challenging flow case.

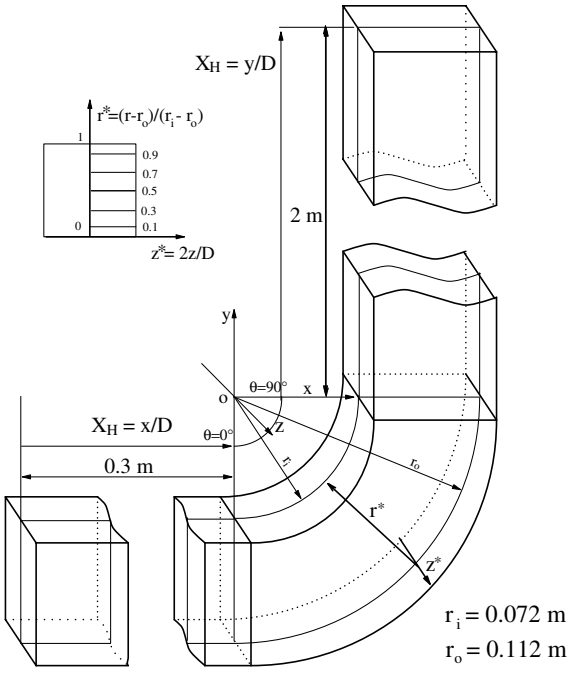

Fig. 8. Dimensions of bend and coordinate definition sketch.

3.5. Curved duct flow with strong secondary motion

This subsection reports on the performance of applying the modified FMG to a three-dimensional curved duct flow which has been investigated experimentally by Taylor et al. [15]. The flow enters a duct of square cross section with a dimension of 40 mm × 40 mm, followed by a 90° bend of mean radius 92 mm, with upstream and downstream tangent lengths of 0.3 m and 2.0 m, respectively, as shown in Fig. 8.

For numerical simulation, a block profile of the mean velocity is applied at the inlet. It is set to 1.98 cm/s and 1.00 m/s for the laminar and turbulent flows, respectively, corresponding to the Reynolds numbers of 790 and 40,000. The non-uniform grids with $128 \times 64 \times 64$ cells on the finest level are divided into eight blocks in the i direction and three grid levels are considered. The standard k–ϵ model and the LL k–ϵ model are applied in conjunction with the MUSCL scheme. The computations have been

Table 8
Speed-up factor for laminar and turbulent duct flow, with $128 \times 64 \times 64$ grid cells

$Re = 790$	$Re = 40{,}000$	
Laminar	k–ϵ	LL k–ϵ
7.61	7.90	7.96

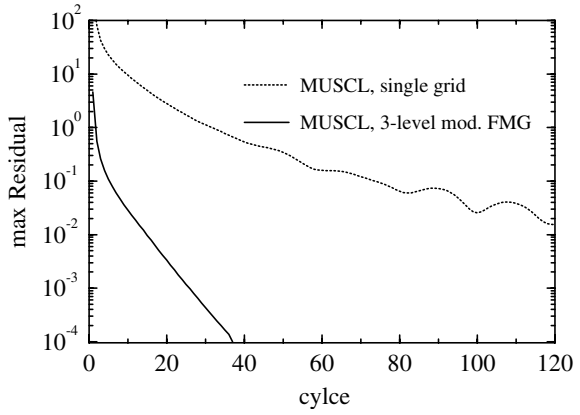

Fig. 9. Convergence histories in terms of cycle number for laminar flow in a curved duct.

carried out on a CRAY-T3E. The parallel algorithm adopted is available in [17]. In this case, the under-relaxation factor $\lambda = 0.7$ in Eq. (21) is used.

The speed-up factors are shown in Table 8. For the laminar flow, a speed-up factor of 7.7 has been obtained. Fig. 9 displays the convergence histories. The development of the streamwise velocity is depicted in Fig. 10, where the agreement between the calculation and experiment is very good.

A speed-up factor of up to 7.96 has been obtained for the turbulent flow simulation. The profiles of the streamwise velocity are shown in Fig. 11, which are plotted from the results for the LL k–ϵ model. The tangential growth of the turbulent boundary layers is slower than that for the laminar flow and thus the boundary layers at $X_H = -0.25$ are thinner, where the core fluid is displaced further toward

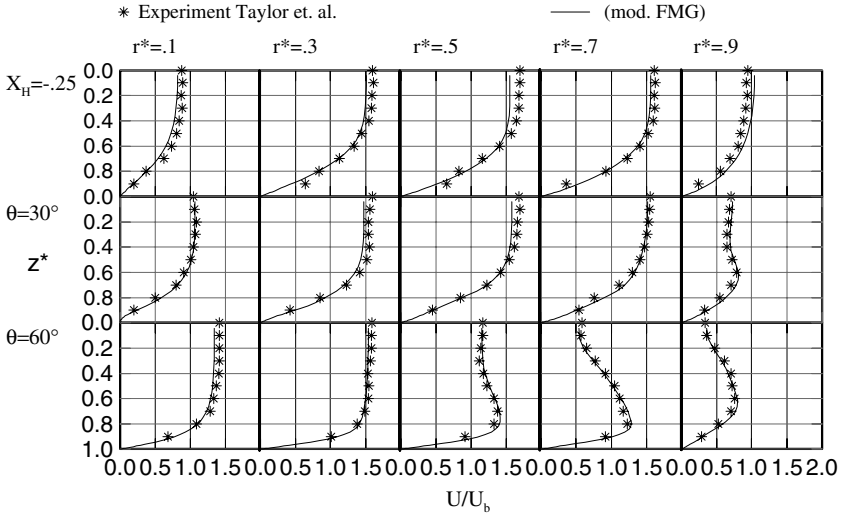

Fig. 10. Profiles of U/U_b at successive streamwise stations $X_H = -0.25$, $\theta = 30°$ and $\theta = 60°$ for laminar flow in a curved duct.

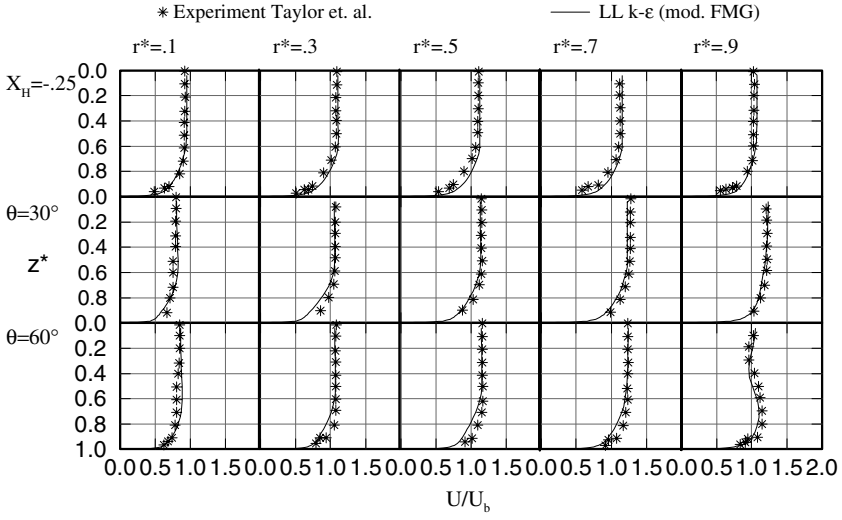

Fig. 11. Profiles of U/U_b at successive streamwise stations $X_H = -0.25$, $\theta = 30°$ and $\theta = 60°$ for turbulent flow in a curved duct.

the inner wall surface than for laminar results. The core fluid remains at the inner wall, with no evidence of low momentum fluid being accumulated at this surface, until $\theta = 60°$. Thereafter the rapid creation of a region of low streamwise momentum at the inner wall is evident and continues to the exit of the bend with corresponding migration of core fluid toward the outer wall.

4. Conclusion

A modified multigrid algorithm has been developed in this work. The performance of the algorithm is assessed based on tests with five different flow problems. Based on the results of the computations, the following conclusions can be drawn:

- For calculations with a low number of grid levels (less than three), the speed-up achieved with the modified FMG algorithm is comparable with that of the standard method. However, when three or more levels are used the modified method achieves considerably faster speed-up factors (roughly 40% more than the standard method).
- The speed-up factors obtainable with the standard FMG–FAS deteriorate strongly in turbulent flow simulations. The modified FMG algorithm however does not exhibit this disadvantage; comparable speed-up factors are observed both for laminar and turbulent flow simulations. This has considerable significance for industrial applications.
- Due to the manner in which the mass flux, turbulence and pressure-correction variables are handled, the implementation is significantly more simple than the standard multigrid method.

The reason for these beneficial properties lies in the modified formulation. In the standard method, the initial mass fluxes through the control volume faces of the coarse grid are usually restricted by the summation of the corresponding two or four fine-grid mass fluxes. This always leads to a mismatch with the restricted velocities. By keeping the coarse grid mass fluxes and velocities from the previous cycle, this problem does not occur with the modified method. The same technique is applied to the turbulence variables, which avoids a further problem with the original multigrid algorithm. In the standard approach, incorrect values of the eddy viscosity (which is defined in terms of these variables) contributed to the poor performance for turbulence simulations.

References

[1] Almeida GP, Durao DFG, Heitor MV. Wake flow behind two-dimensional model hills. Exp Therm Fluid Sci 1993;7:87–101.
[2] Brandt A. Multi-level adaptive solutions to boundary-value problems. Math Comput 1977;31:333–90.
[3] Demirdžić I, Lilek Ž, Perić M. Fluid flow and heat transfer test problems for non-orthogonal grids: bench-mark solutions. Int J Numer Meth Fluids 1992;15:329–54.
[4] Demirdžić I, Perić M. Numerical study of grey-body surface radiation coupled with fluid flow for general geometries using a finite volume multigrid solver. Int J Heat Fluid Flow 1996;6:3–18.
[5] Ferziger JH, Perić M. Computational methods for fluid dynamics. Berlin, Heidelberg, New York: Springer-Verlag; 1996.
[6] Hackbusch W. Multigrid methods and applications. New York, Heidelberg, Berlin: Springer-Verlag; 1985.
[7] van Leer B. Toward the ultimate conservative difference scheme, V. A second-order sequel to Godunov's method. J Comput Phys 1977;23:101–36.
[8] Leonard BP. A stable and accurate convective modeling procedure based on quadratic upstream interpolation. Comput Meth Appl Eng 1979;19:59–98.
[9] Lien FS, Leschziner MA. Multigrid acceleration for recirculating laminar and turbulent flows computed with a non-orthogonal colocated finite-volume scheme. Comput Meth Appl Mech Eng 1994;118:351–71.
[10] Lien FS, Leschziner MA. Assessment of turbulence-transport models including non-linear RNG eddy-viscosity formulation and second-moment closure for flow over a backward-facing step. Comput Fluids 1994;23:983–1004.
[11] Patankar SV, Spalding DB. A calculation procedure for heat, mass and momentum transfer in three-dimensions parabolic flows. Int J Heat Mass Transfer 1972;15:1787–806.
[12] Orth A. Mehrgittermethode zur Berechnung inkompressibler, stationärer Strömungen mit krummlinigen Berandungen. Dissertation, Karlsruhe, Germany, 1991.
[13] Rhie CM, Chow WL. Numerical study of the turbulent flow past an airfoil with trailing edge separation. AIAA J 1983;21:1325–32.
[14] Trottenberg U, Oosterlee CW, Schüller A. Multigrid. San Diego, Dan Franciso, New York, Boston, London, Sydney, Tokyo: Academic Press; 2001.
[15] Taylor AM, Whitelaw JH, Yianneskis M. Curved ducts with strong secondary motion: Velocity measurements of developing laminar and turbulent Flow. J Fluids Eng 1982;104:350–9.
[16] Vanka SP. Block-implicit multigrid solution of Navier–Stokes equations in primitive variables. J Comput Phys 1986;65:138–58.
[17] Xue L. Entwicklung eines effizienten parallelen Lösungsalgorithmus zur dreidimensionalen Simulation komplexer turbulenter Strömungen. Dissertation, TU-Berlin, Germany, 1998.
[18] Zheng X, Liao C, Liu C, Sung CH, Huang TT. Multigrid computation of incompressible flows using two-equation turbulence models: Part I – numerical method. J Fluids Eng 1997;119:893–9.
[19] Zheng X, Liao C, Liu C, Sung CH, Huang TT. Multigrid computation of incompressible flows using two-equation turbulence models: Part II – Applications. J Fluids Eng 1997;119:900–5.
[20] Yan J, Thiele F. Performance and accuracy of a modified full multigrid algorithm for fluid flow and heat transfer. Numer Heat Transfer Part B 1998;34:323–38.