JOURNAL OF COMPUTATIONAL PHYSICS **146,** 301–321 (1998)
ARTICLE NO. CP986067

A Nonlinear Multigrid Method for the Three-Dimensional Incompressible Navier–Stokes Equations

D. Drikakis,* O. P. Iliev,† and D. P. Vassileva†

**UMIST, Department of Mechanical Engineering, P.O. Box 88, Manchester M60 1QD, United Kingdom;*
†*Institute of Mathematics and Informatics, Bulgarian Academy of Science,*
Acad. G. Bonchev St., bl. 8, BG-1113 Sofia, Bulgaria
E-mail: drikakis@umist.ac.uk, oleg@math.acad.bg, vasileva@math.acad.bg

Received December 18, 1997; revised July 12, 1998

A nonlinear multigrid method is developed for solving the three-dimensional Navier–Stokes equations in conjunction with the artificial compressibility formulation. The method is based on the full multigrid (FMG)—full approximation storage (FAS)—algorithm and is realized via an "unsteady-type" procedure, according to which the equations are not solved exactly on the coarsest grid, but some pseudo-time iterations are performed on the finer grids and some on the coarsest grid. The multigrid method is implemented in conjunction with a third-order upwind characteristics-based scheme for the discretization of the convection terms, and the fourth-order Runge–Kutta scheme for time integration. The performance of the method is investigated for three-dimensional flows in straight and curved channels as well as flow in a cubic cavity. The multigrid acceleration is assessed in contrast to the single-grid and mesh-sequencing algorithms. The effects of various multigrid components on the convergence acceleration, such as prolongation operators, as well as pre- and postrelaxation iterations, are also investigated.

Key Words: multigrid; Navier–Stokes equations; upwind schemes; artificial compressibility.

1. INTRODUCTION

During the last decade, rapid advances in computer hardware have provided new avenues in the numerical simulation of three-dimensional fluid flows. However, even with the use of the most powerful super-computers the CPU requirements for three-dimensional steady and unsteady computations are still very high. Beside the use of parallel computers which allow us to reduce the computing time by increasing the number of processors, multigrid methods have been established as a powerful tool for accelerating the numerical convergence and, thus reducing the computing time.

0021-9991/98 $25.00

The origin of the multigrid method is found in the papers of Fedorenko [1] and Bakhvalov [2], and later on in the work of Brandt [3]. To make reference to all past works in connection with multigrid methods would be a task for the introduction of a book (e.g. [4, 5]) rather than of the present paper. However, it is worth mentioning that most of the developments and applications of multigrid method for incompressible flows are related to elliptic systems of equations and mainly to SIMPLE-type approaches (e.g. Cope *et al*. [6]). On the other hand, very few studies have dealt with the development of multigrid methods in conjunction with the artificial compressibility method (ACM) [7]. The idea of ACM is to introduce a pseudo-temporal equation for the pressure through the continuity equation and, subsequently to couple the continuity with the momentum equations. Although there exist different implicit and semi-implicit methods for the incompressible Navier–Stokes equations and the present authors have experience with such methods, including semi-implicit projection methods of SIMPLE-type and algorithms for vorticity–vector potential formulation of the incompressible Navier–Stokes equations, there are several arguments which justify the use of ACM for computing incompressible flows. The most important arguments are: (i) there is recently an increasing interest to explicit methods since they offer high efficiencies in parallel computations, especially in the case of massively parallel computations; (ii) the discretisation schemes and solvers developed in conjunction with ACM for incompressible flows have many similarities with the methods developed for compressible flows. Therefore, the computational experience and developments gained from incompressible flows can easily be transferred to compressible flows and vice versa. Since, in many industrial applications both the computation of compressible and incompressible flows is sometimes required, e.g. automotive industry, the usage of ACM-based approaches offers a lot of advantages.

Concerning the development of multigrid methods in conjunction with ACM, there are very few studies in the literature. Farmer *et al*. [8] have developed a multigrid scheme for the solution of the Euler equations in conjunction with the ACM and applied it to free surface flows. They reported that 400 W-multigrid cycles were required to achieve convergence for the inviscid flow around a ship hull including free surface effects. Sheng *et al*. [9] have developed a multigrid algorithm for 3D incompressible turbulent flows in conjunction with the ACM and Newton relaxation methods. They investigated two different approaches for building coarse grid equations as well as the influence of implicit correction smoothing on increasing the stability of the scheme. They reported fast convergence rates for the case of external flows, but the multigrid efficiency appeared to deteriorate in the case of complex internal flows. The above multigrid methods were similar to the Jameson's multigrid procedures originally developed for the solution of the compressible Euler equations [10–12] and later on applied to the compressible Navier–Stokes equations [13, 14]. Lin and Sotiropoulos [15] have also recently developed a three-level V-cycle multigrid algorithm in conjunction with the ACM using a first-order upwind differencing for the discretization of the convection terms during the coarse grid iterations, while various schemes were implemented and tested for the discretization of these terms on the fine grid. They employed the FAS scheme proposed by Brandt, starting with an estimate of the solution on the finest grid and performing a fixed number of iterations on the coarser grids.

Other recent contributions in the field of multigrid methods, however, less relevant to the approaches used in this study, include the work by Dailey and Pletcher [16] who presented the implementation of multigrid method in conjunction with the preconditioned Navier–Stokes equations for low-Mach number two-dimensional, steady and unsteady flows. Lotstedt [17] also investigated three different relaxation procedures, Runge–Kutta

time-stepping, GMRES, and modified GMRES, in conjunction with a multigrid algorithm for solving the steady state Euler equations, while recently Steelant *et al.* [18] presented a theoretical analysis of different implicit methods in multigrid form for the case of low Mach number flows.

The main differences between the present multigrid method and the aforementioned contributions are:

- the combination of the FMG and FAS approaches for solving the artificial compressibility formulation of the Navier–Stokes equations. The FMG procedure is used to provide a good initial approximation before the execution of V-cycles on the fine-grid, but also to calculate the basic *coarse-grid functions* used in the FAS procedure (see discussion about the FAS algorithm in Section 3);
- the implementation of the FMG-FAS in conjunction with a third-order upwind characteristics-based scheme [19, 20], the latter being employed for the discretization of the convection terms at all grid levels, and
- the implementation and testing of various prolongation operators in conjunction with the FAS-FMG procedure, including a new operator, henceforth labelled *mixed-prolongation*, which is based on an upwind prolongation in the streamwise direction and bilinear prolongation in the cross-stream direction.

Subsequently, the objectives of the present study are (i) to develop the FMG-FAS nonlinear multigrid method and demonstrate its efficiency in various 3D incompressible flows, (ii) to investigate the effects of various multigrid components on the convergence acceleration, such as prolongation operators, coarsest-grid iterations, as well as pre- and postrelaxation iterations, and (iii) to assess the performance of the method against the mesh-sequencing and single-grid algorithms.

The remainder of the paper is organised as follows. In Section 2 the solution method is presented. The multigrid algorithm is described in Section 3 and the results are presented in Section 4. Finally, in Section 5 conclusions from the present study are drawn.

2. GOVERNING EQUATIONS AND SOLUTION METHOD

The governing equations and discretization scheme are described in detail in [19, 20]. In this paper the method is briefly presented in order to understand the implementation of the multigrid algorithm. The governing equations are the Navier–Stokes equations in curvilinear coordinates (ξ, η, ζ):

$$(JU)_t + (E_I)_\xi + (F_I)_\eta + (G_I)_\zeta = (E_V)_\xi + (F_V)_\eta + (G_V)_\zeta.$$

The unknown solution vector U is

$$U = (p/\beta, u, v, w)^{\mathrm{T}},$$

where p is the pressure; u, v, and w are the velocity components; and β is the artificial compressibility parameter.

The inviscid fluxes E_I, F_I, G_I and the viscous fluxes E_V, F_V, G_V are written as

$$\begin{aligned} E_I &= J(\tilde{E}_I \xi_x + \tilde{F}_I \xi_y + \tilde{G}_I \xi_z) \\ F_I &= J(\tilde{E}_I \eta_x + \tilde{F}_I \eta_y + \tilde{G}_I \eta_z) \\ G_I &= J(\tilde{E}_I \zeta_x + \tilde{F}_I \zeta_y + \tilde{G}_I \zeta_z) \end{aligned}$$

$$E_V = J(\tilde{E}_V \xi_x + \tilde{F}_V \xi_y + \tilde{G}_V \xi_z)$$
$$F_V = J(\tilde{E}_V \eta_x + \tilde{F}_V \eta_y + \tilde{G}_V \eta_z)$$
$$G_V = J(\tilde{E}_V \zeta_x + \tilde{F}_V \zeta_y + \tilde{G}_V \zeta_z),$$

where the fluxes with "tildes" denote the corresponding Cartesian fluxes:

$$\tilde{E}_I = \begin{pmatrix} u \\ u^2 + p \\ uv \\ uw \end{pmatrix}, \quad \tilde{F}_I = \begin{pmatrix} v \\ uv \\ v^2 + p \\ vw \end{pmatrix}, \quad \tilde{G}_I = \begin{pmatrix} w \\ uw \\ vw \\ w^2 + p \end{pmatrix},$$

$$\tilde{E}_V = \begin{pmatrix} 0 \\ \sigma_{xx} \\ \sigma_{xy} \\ \sigma_{xz} \end{pmatrix}, \quad \tilde{F}_V = \begin{pmatrix} 0 \\ \sigma_{yx} \\ \sigma_{yy} \\ \sigma_{yz} \end{pmatrix}, \quad \tilde{G}_V = \begin{pmatrix} 0 \\ \sigma_{zx} \\ \sigma_{zy} \\ \sigma_{zz} \end{pmatrix}.$$

The terms σ_{ij} $(i, j = x, y, z)$ are the viscous stresses and J is the Jacobian of the transformation from Cartesian to generalised coordinates:

$$J = x_\xi(y_\eta z_\zeta - y_\zeta z_\eta) + x_\eta(y_\zeta z_\xi - y_\xi z_\zeta) + x_\zeta(y_\xi z_\eta - y_\eta z_\xi).$$

A characteristics-based method [19, 20] is used for the discretization of the inviscid terms. A Riemann solution in each flow direction can be constructed by splitting the inviscid equations into three one-dimensional equations. The primitive variables (p, u, v, w) at the cell faces of the computational volume can then be defined as functions of their values $(p_\kappa, u_\kappa, v_\kappa, w_\kappa)$ on the characteristics denoted by the subscript κ $(\kappa = 0, 1, 2)$. For example, the primitive variables at the cell faces $\xi = \text{const}$ are calculated as

$$u = R\tilde{x} + u_0(\tilde{y}^2 + \tilde{z}^2) - v_0\tilde{x}\tilde{y} - w_0\tilde{x}\tilde{z}$$
$$v = R\tilde{y} + v_0(\tilde{x}^2 + \tilde{z}^2) - w_0\tilde{z}\tilde{y} - u_0\tilde{x}\tilde{y}$$
$$w = R\tilde{z} + w_0(\tilde{y}^2 + \tilde{x}^2) - v_0\tilde{z}\tilde{y} - u_0\tilde{x}\tilde{z}$$
$$p = p_1 - \lambda_1(\tilde{x}(u - u_1) + \tilde{y}(v - v_1) + \tilde{z}(w - w_1)),$$

where

$$R = \frac{1}{2s}(p_1 - p_2 + \tilde{x}(\lambda_1 u_1 - \lambda_2 u_2) + \tilde{y}(\lambda_1 v_1 - \lambda_2 v_2) + \tilde{z}(\lambda_1 w_1 - \lambda_2 w_2))$$

$$s = \sqrt{\lambda_0^2 + \beta}, \quad \tilde{\phi} = \frac{\xi_\phi}{\sqrt{\xi_x^2 + \xi_y^2 + \xi_z^2}}, \quad \phi = x, y, z$$

In the above formulas λ_0, λ_1, and λ_2 are the eigenvalues defined by

$$\lambda_0 = u\tilde{x} + v\tilde{y} + w\tilde{z},$$
$$\lambda_1 = \lambda_0 + s, \quad \lambda_2 = \lambda_0 - s. \tag{1}$$

The characteristic variables are calculated by a third-order upwind scheme,

$$U^{(l,r)}_{i+1/2,\kappa} = \frac{1}{2}\big((1 + \text{sign}(\lambda_\kappa))U^{(l)}_{i+1/2} + (1 - \text{sign}(\lambda_\kappa))U^{(r)}_{i+1/2}\big),$$

where

$$U^{(l,r)}_{i+1/2,\kappa} = (p_\kappa, u_\kappa, v_\kappa, w_\kappa)^{\mathrm{T}}$$
$$U^{(l)}_{i+1/2} = \frac{1}{6}(5U_i - U_{i-1} + 2U_{i+1})$$
$$U^{(r)}_{i+1/2} = \frac{1}{6}(5U_{i+1} - U_{i+2} + 2U_i).$$

The viscous terms are discretized by central differences and the time integration is obtained by an explicit Runge–Kutta method,

$$\begin{aligned}
U^{(1)} &= U^n \\
U^{(2)} &= U^n - \frac{\Delta t}{2} N\left(U^{(1)}\right) \\
U^{(3)} &= U^n - \frac{\Delta t}{2} N\left(U^{(2)}\right) \\
U^{(4)} &= U^n - \Delta t\, N\left(U^{(3)}\right) \\
U^{n+1} &= U^n - \frac{\Delta t}{6}\left[N\left(U^{(1)}\right) + 2N\left(U^{(2)}\right) + 2N\left(U^{(3)}\right) + N\left(U^{(4)}\right)\right],
\end{aligned}$$

where n is the previous time level and N comes from rewriting the Navier–Stokes equations in the form:

$$(JU)_t + N(JU) = 0.$$

The local time step Δt varies between the four Runge–Kutta stages and is defined by

$$\Delta t = \frac{CFL}{\max\limits_m\{\mu_m\}}, \tag{2}$$

where $\mu_m = \max\{(|\lambda_1|, |\lambda_2|)\sqrt{\chi_x^2 + \chi_y^2 + \chi_z^2}\}$, m ($m = 1, 2, \ldots, 6$) is the volume cell-face pointer, λ_1 and λ_2 are the eigenvalues (see Eq. (1)) at the cell faces, and χ stands either for ξ, η, or ζ. Thus, the local time step Δt depends on the grid and flow velocities. A CFL number of 0.5 was used in all calculations.

3. MULTIGRID ALGORITHM

To accelerate the convergence of the single-grid Navier–Stokes method presented in the preceding section, a *full multigrid-full approximation storage* (FMG-FAS) algorithm has been developed. As discussed in the introduction, the multigrid method is amongst the most popular approaches for accelerating fluid flow computations. Although the method was originally suggested for solving elliptic flow problems, it has also been successfully implemented in other cases, such as hypersonic flows [21]. However, little experience has been acquired so far from its implementation in conjunction with the artificial compressibility formulation [8, 9, 15].

In the present work various prolongation operators have been implemented and tested. Some of the prolongation operators used here, e.g. upwind prolongation, cannot be used in conjunction with elliptic systems of equations. In the present work a three-level multigrid

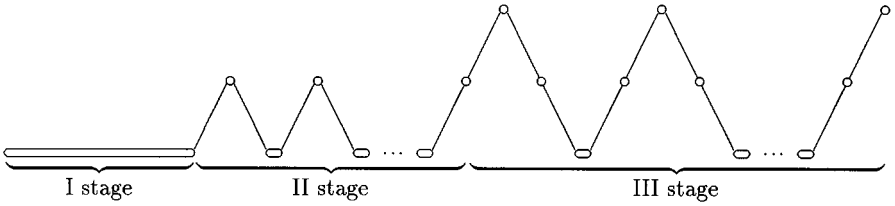

FIG. 1. Schematic of the full multigrid (FMG) for three grids. I. Single grid computation on the coarsest grid; II. Two-level multigrid computation on the intermediate grid; III. Three-level multigrid computation on the finest grid.

has been developed. There are several reasons for the choice of the so-called *short-multigrid* and among them are:

- the *coarsest-grid level* should have a sufficient number of grid points to provide a good correction onto the fine grid. Numerical experiments have shown that in the case of very coarse grids the efficiency of the multigrid is significantly reduced. This has also been observed in theoretical investigations of multilevel algorithms for nonsymmetric (see, e.g. [22, 23] and references therein) and nonlinear problems (see, e.g. [24] and references therein);
- *short-multigrid* algorithms are more efficient in parallel computations, as demonstrated in previous studies by Ålund *et al.* [25] and Axelsson and Neytcheva [26, 27];
- finally, it should be pointed out that the use of several grid levels increases the complexity of the computer code and memory requirements. If calculations on very fine grids are required, an alternative strategy could be to solve the equations sequentially on sets of three grids. In this case the initial condition, as well as the *coarse-grid* functions (see also the discussion about the FAS algorithm in this section), can be obtained by the solutions on the previous set of three grids.

The three-level FMG-FAS algorithm is schematically shown in Figs. 1 and 2, while the basic steps are listed below:

Auxiliary stage I—single grid solution

$U_{cg} := N_{cg}^{-1} 0_{cg}$ compute coarsest grid solution
$U_{ig}^{0} := P U_{cg}$ prolongation—initial guess on the intermediate grid

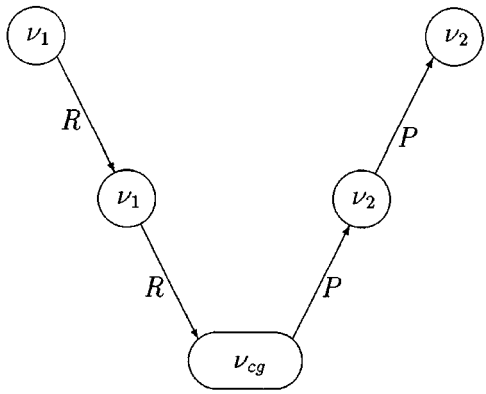

FIG. 2. Schematic of the V-cycle.

Auxiliary stage II—multigrid sweeps on two grids

repeat

$U_{ig} := S_{ig}(U_{ig}, 0_{ig}, \nu_1)$	ν_1 prerelaxation iterations
$d_{ig} := N_{ig}U_{ig}$	compute intermediate grid defect
$d_{cg} := Rd_{ig}$	restriction of the defect to the coarsest grid
$f_{cg} := -d_{cg} + N_{cg}\bar{V}_{cg}$	compute right-hand side on the coarsest grid
$V_{cg} := N_{cg}^{-1} f_{cg}$	compute coarsest grid approximate solution
$c_{cg} := V_{cg} - \bar{V}_{cg}$	compute correction on the coarsest grid
$c_{ig} := Pc_{cg}$	prolongation of the correction to the intermediate grid
$U_{ig} := U_{ig} + c_{ig}$	correct solution on the intermediate grid
$U_{ig} := S_{ig}(U_{ig}, 0_{ig}, \nu_2)$	ν_2 postrelaxation iterations

until the steady state solution on the intermediate grid is achieved

$U_{fg}^0 := PU_{ig}$	prolongation—initial guess on the finest grid

Stage III—multigrid sweeps on three grids (V-cycles)

repeat

$U_{fg} := S_{fg}(U_{fg}, 0_{fg}, \nu_1)$	ν_1 prerelaxation iterations
$d_{fg} := N_{fg}U_{fg}$	compute finest grid defect
$d_{ig} := Rd_{fg}$	restriction of the defect to the intermediate grid
$f_{ig} := -d_{ig} + N_{ig}\bar{V}_{ig}$	compute right hand side on the intermediate grid
$V_{ig} := S_{ig}(V_{ig}, f_{ig}, \nu_1)$	ν_1 prerelaxation iterations
$d_{ig} := -f_{ig} + N_{ig}V_{ig}$	compute intermediate grid defect
$d_{cg} := Rd_{ig}$	restriction of the defect to the coarsest grid
$f_{cg} := -d_{cg} + N_{cg}\bar{V}_{cg}$	compute right hand side on the coarsest grid
$V_{cg} := N_{cg}^{-1} f_{cg}$	compute coarsest grid approximate solution
$c_{cg} := V_{cg} - \bar{V}_{cg}$	compute correction on the coarsest grid
$c_{ig} := Pc_{cg}$	prolongation of the correction to the intermediate grid
$V_{ig} := V_{ig} + c_{ig}$	correct solution on the intermediate grid
$V_{ig} := S_{ig}(V_{ig}, f_{ig}, \nu_2)$	ν_2 postrelaxation iterations
$c_{ig} := V_{ig} - V_{ig}$	compute correction on the intermediate grid
$c_{fg} := Pc_{ig}$	prolongation of the correction to the finest grid
$U_{fg} := U_{fg} + c_{fg}$	correct solution on the finest grid
$U_{fg} := S_{fg}(U_{fg}, 0_{fg}, \nu_2)$	ν_2 postrelaxation iterations

until the steady state solution on the finest grid is achieved

The components of the above full multigrid–full approximation storage (FMG-FAS) algorithm are discussed below.

1. *Full Multigrid* (*FMG*)

According to the FMG approach, computations are initially performed on the coarsest grid in order to provide a good initial guess for the intermediate grid. The same procedure is repeated on the intermediate grid in order to provide a good initial guess for the finest grid. Thus, FMG for three grids can be divided into three stages: two auxiliary stages, where the steady state coarsest and intermediate grid solutions are computed, and the main stage where multigrid sweeps on three grids are performed.

2. *Full Approximation Storage* (*FAS*)

This algorithm was first proposed by Brandt (see [3] and discussion in [4, 5]) due to the nonlinearity of the problem. In the present work the FMG is combined with FAS. As known, for linear problems a correction of the solution on the fine grid can be directly computed on coarser grids using the same solution matrix with the right-hand sides of the equations being the *restricted defect*. However, this is not the case when nonlinear problems are solved. For nonlinear problems the multigrid corrections are formed as differences between some basic, reference solution and the currently computed approximation of this solution. That is why the three-grid FAS algorithm requires the calculation of the so-called *coarse-grid* functions. In the case of the three-level multigrid these functions need to be defined for the coarsest, $\bar{V}_{cg}$, and intermediate grids, $\bar{V}_{ig}$, respectively. In the original Brandt's algorithm (henceforth labelled FAS-1) these functions are computed as projections of the current intermediate and finest grid solutions onto the coarsest and intermediate grids, respectively,

$$\bar{V}_{cg} = RV_{ig}, \quad \bar{V}_{ig} = RU_{fg}$$

where R is the restriction operator. Another approach (henceforth labelled FAS-2) is realized here; the computed, via the FMG, steady state coarsest and intermediate grid solutions, U_{cg} and U_{ig}, are used as coarsest and intermediate grid functions in Brandt's FAS algorithm: $\bar{V}_{cg} = U_{cg}$, $\bar{V}_{ig} = U_{ig}$.

Several numerical experiments were performed during the development of the present method and showed that the above implementation improves the performance of the multigrid algorithm in the case of fine grids and, additionally, this performance was retained for all flow cases and grids used. The effects of the coarse grid function on the multigrid acceleration are demonstrated in the results section for cavity flow calculations.

3. *Relaxation Procedure*

The single-grid algorithm described in Section 2 is used as relaxation procedure, (S_{fg}, S_{ig}) and coarsest-grid solver. It should be noted that the Navier–Stokes solver used on the coarsest and intermediate grids is slightly different than the original single-grid solver. This is due to the fact that the right-hand side of the Navier–Stokes equations is identically zero inside the domain only in the case of the single-grid algorithm. In the case of the multigrid method the right-hand side of the equations on the coarsest and intermediate grids is not zero, due to the additional terms arising from the FAS linearization procedure.

4. *Intergrid Transfer Operators*

The restriction operator for the residuals is obtained by the volumes' weighted summation of the residuals over the fine-grid control volumes (CVs) which subsequently form the current coarse-grid CVs. The present implementation of the multigrid algorithm is based on the assumption that any coarse-grid CV consists of eight fine-grid CVs. In the case of simple geometries the original domain is covered by a coarse grid and this grid is further refined in such a way that any coarse-grid volume is split into eight fine-grid volumes. For complex geometries it is, however, suggested to first generate the finest grid, and then to construct the coarser grids by eliminating lines of the fine grid.

Various prolongation operators have been employed. For the sake of simplicity these operators are described below for the case of uniform grids. However, in the case of the

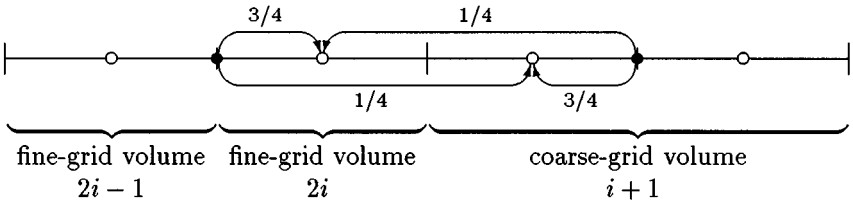

FIG. 3. Schematic of the linear prolongation in 1D case: •, coarse-grid CV; ○, fine-grid CV.

bilinear and trilinear prolongation on nonuniform grids, geometrical factors (grid-weighted averages) such as the distances between grid nodes, have been taken into account. Let us denote by U^f and U^c the values of the variable U on the fine and coarse grids, respectively. For one-dimensional problems, the fine-grid cells with indices $(2i-1)$ and $(2i)$ will form a coarse-grid cell denoted by the index (i).

The simplest definition of the prolongation operator is the linear interpolation (see also Fig. 3):

$$U^f_{2i} = \frac{1}{4}U^c_{i+1} + \frac{3}{4}U^c_i$$

$$U^f_{2i+1} = \frac{3}{4}U^c_{i+1} + \frac{1}{4}U^c_i.$$

For two- and three-dimensional cases, bilinear or trilinear prolongation formulas can be obtained by combining 1D linear interpolation. A schematic of the bilinear interpolation for the two-dimensional case is shown in Fig. 4, while for the three-dimensional case the prolongated value onto a fine-grid cell with indices $(2i, 2j, 2k)$ is given by:

$$U^f_{2i,2j,2k} = \frac{1}{64}U^c_{i+1,j+1,k+1} + \frac{3}{64}U^c_{i+1,j+1,k} + \frac{3}{64}U^c_{i+1,j,k+1} + \frac{3}{64}U^c_{i,j+1,k+1}$$
$$+ \frac{9}{64}U^c_{i+1,j,k} + \frac{9}{64}U^c_{i,j+1,k} + \frac{9}{64}U^c_{i,j,k+1} + \frac{27}{64}U^c_{i,j,k}.$$

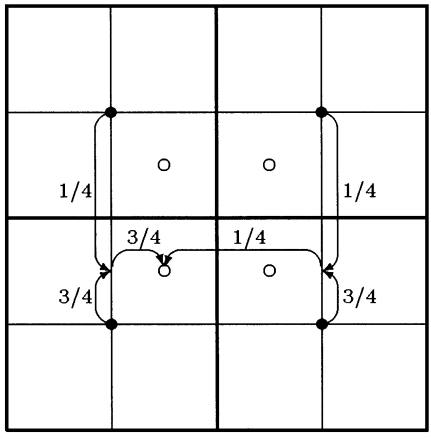

FIG. 4. Schematic of the bilinear prolongation in 2D case: •, coarse-grid node; ○, fine-grid node.

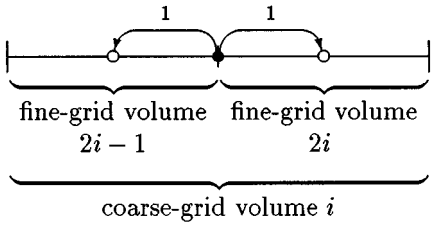

FIG. 5. Schematic of the piece-wise constant prolongation in 1D case.

Another prolongation operator tested in this study is the piece-wise constant prolongation. For the 1D case this operator is defined as (see also Fig. 5)

$$U^f_{2i-1} = U^c_i$$
$$U^f_{2i} = U^c_i$$

while for a three-dimensional problem the operator is written as

$$U^f_{2i-1,2j-1,2k-1} = U^f_{2i-1,2j-1,2k} = U^f_{2i-1,2j,2k-1} = U^f_{2i-1,2j,2k}$$
$$= U^f_{2i,2j-1,2k-1} = U^f_{2i,2j-1,2k} = U^f_{2i,2j,2k-1} = U^f_{2i,2j,2k} = U^c_{i,j,k}$$

It should also be pointed out that in contrast to the trilinear prolongation the present piece-wise constant prolongation is not based on grid-weighted averaging. An upwind piece-wise constant prolongation has also been implemented. For the one-dimensional case this is written as

$$U^f_{2i} = U^f_{2i+1} = U^c_i \quad \text{for } u_i > 0$$

and

$$U^f_{2i} = U^f_{2i+1} = U^c_{i+1} \quad \text{for } u_i < 0,$$

where u_i is the velocity of the fluid. The above is schematically shown in Fig. 6 for the 2D case when $u_i > 0$ (the extension in 3D is straightforward).

In addition to the implementation of the trilinear, piece-wise constant and upwind piece-wise constant prolongation, a combination of upwind prolongation in the streamwise direction and bilinear in the cross-stream plane (henceforth labelled *mixed-prolongation*) was also implemented and tested. This prolongation is schematically shown in Fig. 7 for the 2D case when $u_i > 0$.

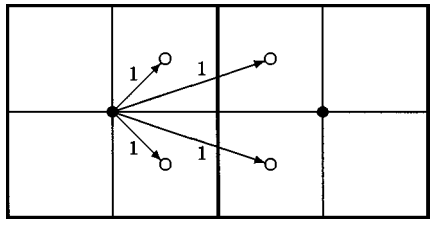

FIG. 6. Schematic of the upwind piece-wise constant prolongation in 2D case.

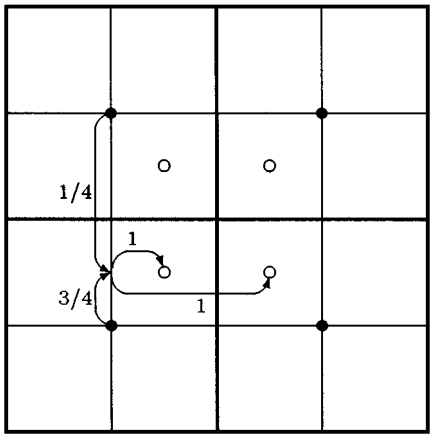

FIG. 7. Schematic of the mixed-prolongation in 2D case.

5. *Coarsest Grid Solution*

A multigrid algorithm can be realized by either a "steady-type" or "unsteady-type" procedure. The "steady-type" procedure is similar to that used in multigrid algorithms for elliptic problems. In this case the solution on the coarsest grid is computed almost exactly. In the "unsteady-type" multigrid the equations on the coarsest grid are not solved until the convergence is achieved. Similarly to the single-grid solution a pseudo-unsteady problem is solved according to which some of the time steps are performed on the finer grids and others on the coarsest grid. In the present work an "unsteady-type" multigrid has been developed.

4. RESULTS

The performance of the multigrid algorithm was investigated for the following three-dimensional incompressible flows: (i) the flow development in a straight 3D channel of square cross-section, (ii) flow in a cubic cavity, and (iii) flow in a 3D channel with strong curvature.

For the channel flow a Reynolds number of 100, based on the centre-line velocity and channel width, is used. The calculations were performed on a single quadrant of the channel due to the symmetry. In order to investigate the multigrid performance on different grids, calculations were performed for two cases corresponding to fine grids with size $58 \times 39 \times 39$ (case 1) and $42 \times 17 \times 17$ (case 2), respectively. The grid points were slightly clustered in the x-direction near the channel entrance, while the grid was uniform in the other directions. The computed pressure coefficient along the channel centre-line was compared with the experimental data of Beavers *et al.* [28] (Fig. 8). The axial development of the streamwise velocity at the channel centre-line as well as the velocity profile at $X/(D * \mathrm{Re}) = 0.02$ (D is the channel width) were compared with the corresponding laser Doppler velocimetry measurements of Goldstein and Kreid [29] (see also Fig. 8).

The efficiency of the multigrid method was assessed against the mesh-sequencing and single-grid algorithms. According to the mesh-sequencing technique the solution is first obtained on a sequence of coarser grids in order to provide an initial guess for the solution on the fine grid. One can easily understand that the first auxiliary stage of the FAS-FMG method is identical with the mesh-sequencing procedure used for solving the equations on the coarsest grid.

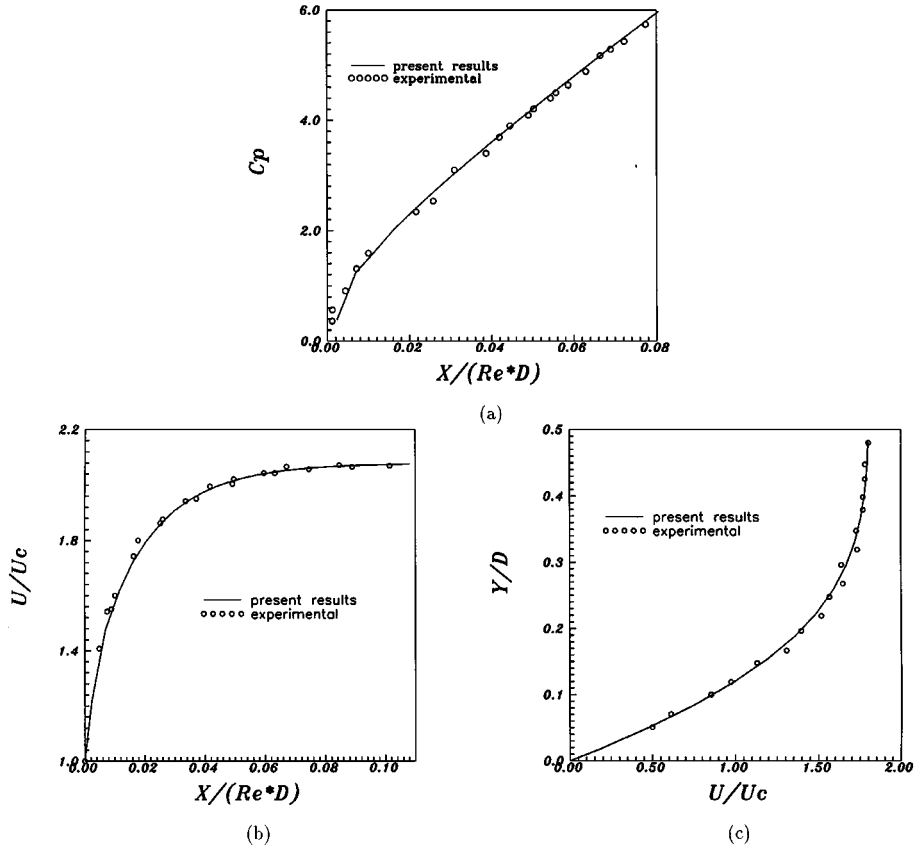

FIG. 8. Comparison of the computations with experimental results (Re = 100) for the three-dimensional entry flow in a rectangular channel: (a) pressure distribution along the centreline; (b) development of the velocity along the centreline; and (c) streamwise velocity profile at $X/(\mathrm{Re} * D) = 0.02$ (grid $58 \times 39 \times 39$).

The pressure residual against the corresponding work units is plotted in Fig. 9 for the multigrid (MG), mesh-sequencing (MS) and single-grid (SG) solutions for the case (1), while the corresponding work units for the cases (1) and (2) are shown in Table I. The work units are calculated by taking into account that one relaxation step at the grid level l is equivalent to $1/8^{l-1}$ work units of the finest grid ($l = 1$). The pre- and postrelaxation

TABLE I
MG Sweeps and Total Work Units for the 3D Channel Flow (Three-Level MG)

Grid	Method	Work units	MG sweeps
$58 \times 39 \times 39$	MG	2429	105
	MS	32400	
	SG	>67000	
$42 \times 17 \times 17$	MG	1000	47
	MS	9500	
	SG	>19500	

Note. Re = 100.

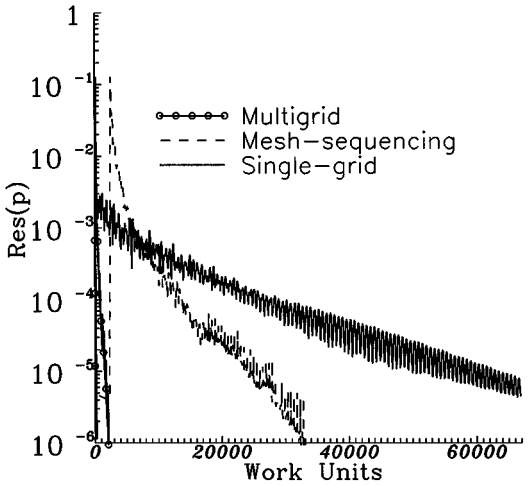

FIG. 9. Convergence history for the 3D entry flow in channel (grid $58 \times 39 \times 39$).

iterations have also been included in the total work units. As can be seen from Table I by comparing the SG and MG work units, the MG algorithm offers a significant acceleration of the computations by a factor of over 30 and 19.5 for the cases (1) and (2), respectively. Comparing to the MS solution the acceleration factor is 13.3 and 9.5 for the cases (1) and (2), respectively. In the case of the SG solution the exact acceleration cannot be estimated since the computation was stopped before the prescribed convergence level is reached. The fact that an acceleration over 30 times had already been achieved, provided sufficient evidence for the efficiency of the MG algorithm. One can also notice from Table I that the MG sweeps are increasing from case (2) to case (1). This is due to the "unsteady-type" of multigrid implementation. According to the present implementation, the equations are not solved on the coarse grids of the MG cycle up to the final convergence and, therefore, the MG sweeps required will not be independent of the grid size. We have found that the above increases the performance of the MG solver by reducing the total work units, but as a result the MG sweeps will vary between different grids.

The above computations were performed using the following combination of multigrid parameters: $\nu_1 = 0$ pre-relaxations, $\nu_{cg} = 100$ steps on the coarsest grid, and $\nu_2 = 15$ post-relaxations on each multigrid sweep. In all computations presented in this paper the artificial compressibility parameter was kept constant, equal to 1. The *mixed-prolongation* was used for the velocities after the auxiliary stage of the FMG, while the trilinear interpolation was used to prolongate the pressure and corrections from the coarse to the fine grid within each multigrid sweep.

Most of the multigrid studies in literature employ the trilinear interpolation as a prolongation operator. In some papers the choice of the piece-wise constant prolongation is also used. In general, the trilinear interpolation should be used for second-order derivatives (e.g. viscous terms) and the piece-wise interpolation for first-order derivatives (e.g. convective terms). However, the best choice for the prolongation operator is not obvious, especially in viscous flows where large gradients occur. In the present study four different variants of the prolongation operator were investigated:

1. *mixed-prolongation* for u, v, and w after the auxiliary stage, trilinear prolongation for p and corrections;

TABLE II
MG Sweeps (Three-Level MG) and Total Work Units Using Different Prolongation Operators

Type of prolongation	MG sweeps	Work units
(1)	105	2429
(2)	137	2875
(3)	231	4988
(4)	226	4886

2. trilinear prolongation at all stages of the FMG-FAS procedure;
3. piece-wise constant prolongation at all stages of the FMG-FAS procedure;
4. upwind piece-wise constant prolongation for u, v, and w after the auxiliary stage, and trilinear prolongation for p and corrections.

The results using the above four variants are shown in Table II. In this table the number of multigrid sweeps (MG sweeps) on the fine-grid, required for steady state solution with accuracy of 10^{-6} for the L_2-norm of the u-residual, is shown. The number of work units is also given to allow comparison of the efficiency between different multigrid variants. As can be seen, the best results are obtained using the mixed-prolongation for velocities at the end of each auxiliary stage, combined with the trilinear interpolation for prolongating the pressure and corrections from the coarse to the fine grid during the multigrid sweeps.

As mentioned in Section 3, in the "unsteady-type" multigrid the equations on the coarsest grid are not solved up to the convergence, but a number of relaxation steps are performed. The dependence of the multigrid convergence on the relaxation steps is shown in Table III. The best performance is obtained for 100 relaxations, but the optimum number of relaxation iterations is not possible to be defined in advance. Various numerical experiments performed in this study showed that 80 to 100 relaxation steps on the coarsest grid during the MG cycles are sufficient to provide satisfactory convergence rates.

The effect of the pre- and postrelaxations has also been investigated and results are shown in Table IV. It is seen that multigrid is more efficient when only postrelaxations are performed. This conclusion is also in agreement with previous investigations by de Zeeuw [30] for the case of linear problems.

The second case is the flow in a cubic cavity at $\mathrm{Re} = 1000$. No-slip boundary conditions for all velocity components on the wall were employed. In addition, no-slip conditions were used for the v and w velocity components on the upper lid of the cavity. Calculations were performed for two cases with the finest grids, $31 \times 31 \times 31$ and $47 \times 47 \times 47$, respectively.

TABLE III
Effects of the Coarse-Grid Iterations on the MG Sweeps (Three-Level MG)

ν_{cg}	MG sweeps	Work units
100	105	2429
400	86	2447
30	210	4332

TABLE IV
Effects of the Pre- and Postrelaxation Iterations on the MG Sweeps and Total Work Units (Three-Level MG; Grid $58 \times 39 \times 39$)

ν_1	ν_2	MG sweeps	Work units
0	15	105	2429
10	50	80	5521
5	10	188	2900

The streamlines in the $x-y$ ($z=0.5$) and $y-z$ ($x=0.5$) planes are shown in Fig. 10. The convergence histories for the fine-grid case are shown in Fig. 11 and the work units are summarized in Table V. For the fine-grid case, 86 MG sweeps are sufficient to provide the steady state solution. In this case acceleration of the convergence by a factor of about 32, compared to the SG solution, was achieved. An investigation of the pre- and postrelaxation iterations on the multigrid performance was carried out using the $31 \times 31 \times 31$ grid, and the results are summarized in Table VI. It was found that 10 to 15 relaxation iterations are sufficient for an efficient multigrid solution. Results for the effects of the coarse-grid function (see Section 3.2) on the MG performance are also presented in Table VII. The results indicate that the FAS-2 implementation offers a 15% to 20% reduction of the total work units.

The last case is the three-dimensional flow at $\mathrm{Re}=790$ in a 90° bend of 40×40 mm cross section. This flow case was experimentally studied by Humphrey *et al*. [31]. A schematic of the flow geometry is shown in Fig. 12. The mean radius of the bend is 92 mm attached to the end of rectangular channel. A straight extension section is attached upstream of the bend entrance. The parameters of the experiment are such that the bend has a large enough turning angle and a small enough mean radius to generate severe distortion and a significant secondary flow. The multigrid performance was investigated on two different grids. The first grid has 80 nodes in the streamwise direction, and 80×40 in the transverse plane, i.e. a total 256,000 grid points. The second grid has 40 nodes in the streamwise and 40×20 in

FIG. 10. Streamlines on the planes $x = 0.5$ (left) and $z = 0.5$ (right), for the grid $47 \times 47 \times 47$.

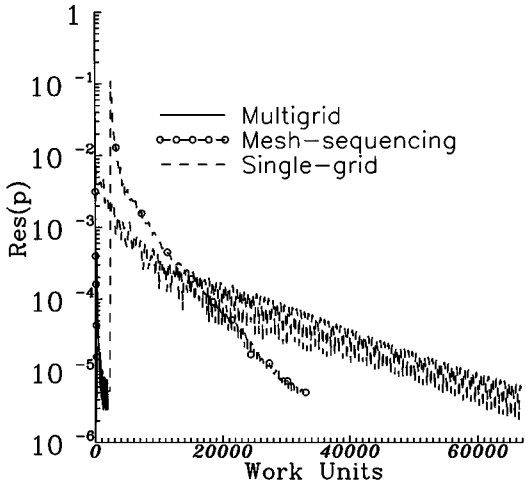

FIG. 11. Convergence history for the cubic cavity flow (grid $47 \times 47 \times 47$).

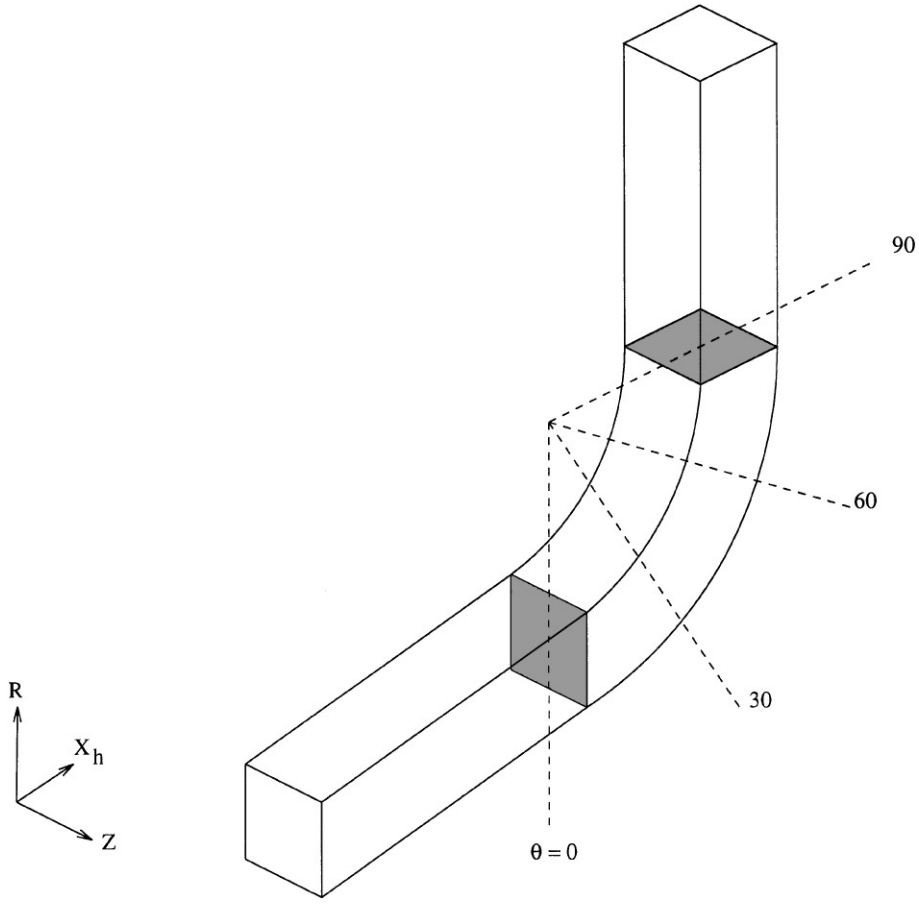

FIG. 12. Schematic of the 90° curved channel.

TABLE V
MG Sweeps and Total Work Units for the Cubic Cavity Flow (Three-Level MG)

Grid	Method	Work units	MG sweeps
$47 \times 47 \times 47$	MG	1935	86
	MS	33600	
	SG	~62000	
$31 \times 31 \times 31$	MG	1250	53
	MS	4200	
	SG	8500	

Note. Re = 1000.

TABLE VI
Effects of the Pre- and Postrelaxation Iterations on the MG Sweeps and Total Work Units (Three-Level MG; Cubic Cavity Flow, Grid $31 \times 31 \times 31$)

ν_1	ν_2	MG sweeps	Work units
0	15	53	1250
10	15	75	2376
0	10	73	1510
0	20	46	1327
0	5	123	1352

TABLE VII
Effects of the FAS-1 and FAS-2 Implementation on the Total Work Units (Cubic Cavity Flow, Grid $47 \times 47 \times 47$)

Grid	FAS-version	Work units
$47 \times 47 \times 47$	FAS-1	2322
	FAS-2	1935
$31 \times 31 \times 31$	FAS-1	1438
	FAS-2	1250

TABLE VIII
MG Sweeps and Total Work Units for the 3D Flow in a Curved Channel (Three-Level MG)

Grid	Method	Work units	MG sweeps
$80 \times 80 \times 40$	MG	2290	100
	MS	33200	
	SG	>70000	
$40 \times 40 \times 20$	MG	1294	62
	MS	11000	
	SG	23000	

Note. Re = 790.

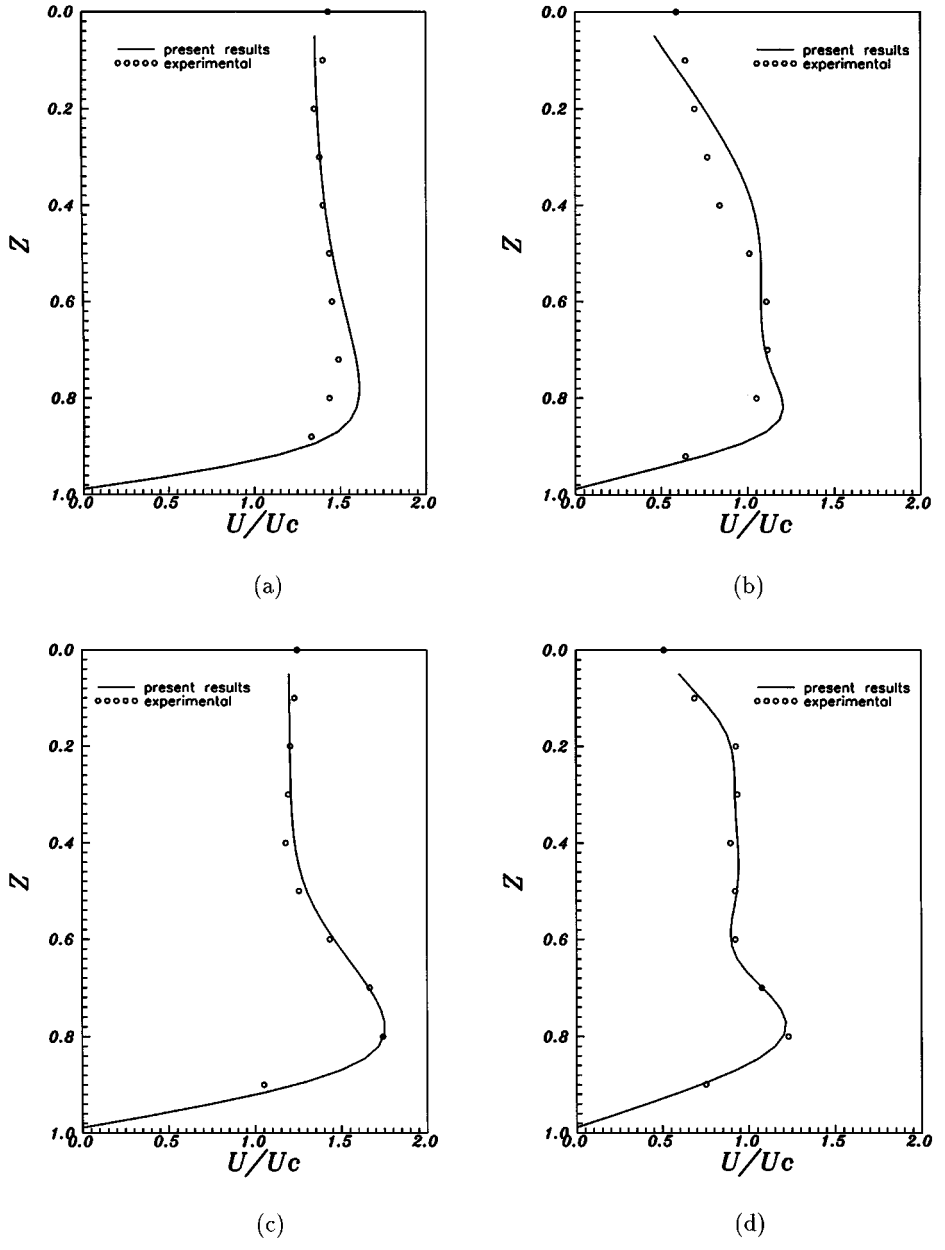

FIG. 13. Comparison of the numerical (grid $80 \times 80 \times 40$) with experimental results [31]: (a) $\theta = 60^\circ$ and $\tilde{R} = 0.3$; (b) $\theta = 60^\circ$ and $\tilde{R} = 0.7$; (c) $\theta = 90^\circ$ and $\tilde{R} = 0.3$; (d) $\theta = 90^\circ$ and $\tilde{R} = 0.7$.

the transverse plane. Computations were carried out using the half section of the duct in the z-direction because of symmetry. As inflow conditions the corresponding developed flow in a straight duct at $\mathrm{Re} = 790$ was imposed at the inlet.

In Fig. 13 comparisons of the present computations with the experimental results of [31] are shown. The comparisons are presented at two different radial locations, $\tilde{R} = 0.3$ and $\tilde{R} = 0.7$, for angles $\theta = 60^\circ$ and $\theta = 90^\circ$. The $\tilde{R}$ is defined by $\tilde{R} = (R - R_o)/(R_i - R_o)$,

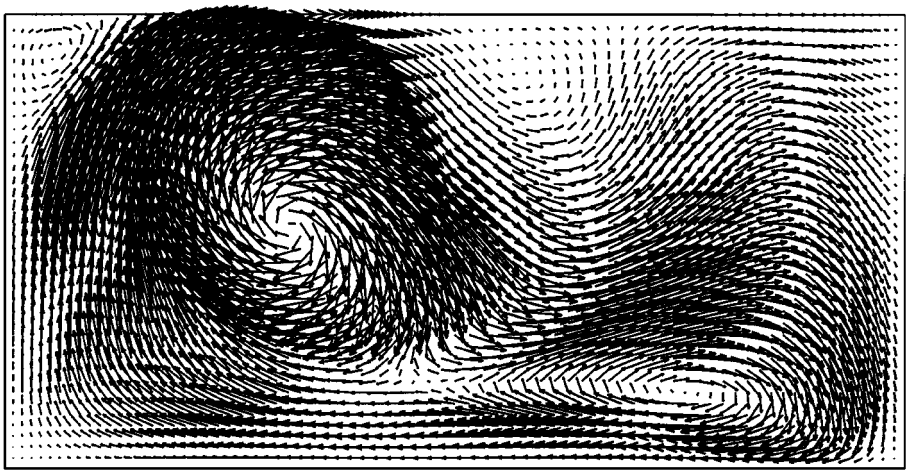

FIG. 14. Formation of the secondary flow at $\theta = 90^\circ$ (grid $80 \times 80 \times 40$).

where R_i and R_o are the inner and outer radius, respectively. The comparisons show that the present predictions are in satisfactory agreement with the experimental results. In Fig. 14 the formation of secondary flow at $\theta = 90^\circ$ is shown.

Similar to the previous cases calculations were also carried out using the mesh-sequencing and single-grid algorithms. The work units and MG sweeps are shown in Table VIII, and the convergence histories are shown in Fig. 15. The MG algorithm offers a significant acceleration of the convergence, especially in the case of the fine grid. Compared to the corresponding SG work units the acceleration factor is over 31 times. The SG computation was stopped when the convergence level had reached the value of 7×10^{-6} compared to the corresponding value of 10^{-6} for the MG solution. An acceleration factor of 14.5 is achieved when the MG work units are compared with the corresponding ones for the MS technique.

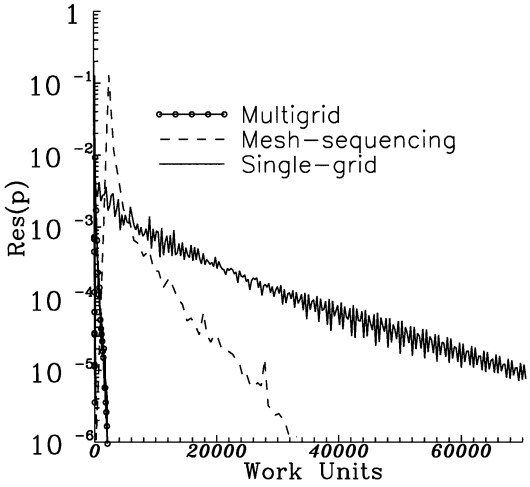

FIG. 15. Convergence history for the 90° curved channel case (grid $80 \times 80 \times 40$).

5. CONCLUSIONS

A nonlinear full multigrid—full approximation storage algorithm was developed in conjunction with the artificial compressibility formulation and a third-order upwind characteristics-based method. The time integration was obtained by the fourth-stage Runge–Kutta scheme. The multigrid performance was investigated for various three-dimensional incompressible flows and validation of the method was performed on different grid sizes up to 256,000 grid points.

The results showed that the present multigrid algorithm offers a significant acceleration of the computations in comparison with the single-grid and mesh-sequencing algorithms. The effects of different prolongation operators on the multigrid performance were also investigated. It was found that the prolongation operators may have significant effects on the multigrid acceleration. The best results were obtained by a combination of the *mixed-prolongation* for the velocities at the end of each auxiliary stage, and trilinear prolongation for the pressure and corrections during the multigrid cycles. An investigation of the effects of pre- and postrelaxation iterations on the multigrid performance was also carried out. The best results were obtained when pre-relaxation iterations were not performed. The optimum number of postrelaxation iterations is difficult to be estimated in advance. An increase of the postrelaxation iterations leads usually to a reduction of the MG sweeps, but also in an increase of the total work units since the computational work per MG-sweep increases. On the other hand, several numerical experiments were performed during the development of the present method and showed that 15 postrelaxation iterations are sufficient to provide good convergence rates.

The coarse-grid iterations may also have a significant effect on the multigrid accelaration. Similar to the postrelaxation iterations, the optimum number of coarse-grid iterations cannot be estimated in advance. On the other hand, an increase of this number by a factor of 3 or 4 will not lead to a significant increase of the total work units because these iterations are performed on the coarsest grid. The numerical experiments revealed that the definition of the coarse-grid function in the FAS procedure can also affect the multigrid acceleration. It was shown that the definition of this function on the basis of coarse-grid solutions obtained through the FMG procedure, improves the multigrid performance. Finally, the same discretization scheme was used at all grid levels and no problems were encountered concerning the numerical stability of the third-order upwind differencing.

ACKNOWLEDGMENT

The financial support from CEC under Grant CP 1239 PEGAS is greatly acknowledged.

REFERENCES

1. R. P. Fedorenko, A relaxation method for solving elliptic difference equations, *USSR Comput. Math. Phys.* **1**, 1092 (1961).
2. N. S. Bakhvalov, On the convergence of a relaxation method with natural constraints on the elliptic operator, *USSR Comput. Math. Phys.* **6**, 101 (1966).
3. A. Brandt, A multilevel adaptive solutions of boundary value problems, *Math. Comput.* **31**, 333 (1977).
4. W. Hackbusch, *Multi-Grid Methods and Applications* (Springer-Verlag, Berlin/Heidelberg/New York/Tokyo, 1985).
5. P. Wesseling, *An Introduction to Multigrid Methods* (Wiley, New York, 1991).

6. W. K. Cope, G. Wang, and S. P. Vanka, A staggered grid multilevel method for the simulation of fluid flow in 3-D complex geometries, *AIAA Paper 94-0778* (1994).
7. A. J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2**, 12 (1967).
8. J. Farmer, L. Martinelli, and A. Jameson, A fast multigrid method for solving incompressible hydrodynamic problems with free surfaces, *AIAA-93-0767* (1993).
9. Ch. Sheng, L. K. Taylor, and D. L. Whitfield, An efficient multigrid acceleration for solving the 3D incompressible Navier–Stokes equations in generalized curvilinear coordinates, *AIAA-93-2335* (1994). [*AIAA J.* **33**(11) (1995)]
10. A. Jameson, Solution of the Euler equations for two dimensional transonic flow by a multigrid method, *Appl. Math. Comput.* **13**, 327 (1983).
11. A. Jameson, Computational transonics, *Comm. Pure Appl. Math.* **41**, 507 (1988).
12. A. Jameson, A vertex based multigrid algorithm for three-dimensional compressible flow calculations, in *ASME Symposium on Numerical Methods for Compressible Flows, Annaheim, December, 1986.*
13. F. Liu and A. Jameson, Multigrid Navier–Stokes calculations for three-dimensional cascades, *AIAA J.* **31**, 1785 (1993).
14. H. Kuerten and B. Geurts, Compressible turbulent flow simulation with a multigrid multiblock method, in *Proc. Copper Mountain Multigrid Conference, 1993*, p. 305.
15. F. B. Lin and F. Sotiropoulos, Strongly-coupled multigrid method for 3D incompressible flows using near-wall turbulence closures, *J. Fluids Engnrg.* **119**, 314 (1997).
16. L. D. Dailey and R. H. Pletcher, Evaluation of multigrid acceleration for preconditioned time-accurate Navier–Stokes algorithms, *Comput. Fluids* **25**, 791 (1996).
17. P. Lotstedt, Improved convergence to the steady state of the Euler equations by enhanced wave propagation, *J. Comput. Phys.* **114**, 34 (1994).
18. J. Steelant, E. Dick, and S. Pattijn, Analysis of robust multigrid methods for steady viscous low Mach number flows, *J. Comput. Phys.* **136**, 603 (1997).
19. D. Drikakis, P. Govatsos, and D. Papantonis, A characteristic based method for incompressible flows, *Int. J. Num. Meth. Fluids* **19**, 667 (1994).
20. D. Drikakis, A parallel multiblock characteristics-based method for three-dimensional incompressible flows, *Adv. Eng. Software* **26**, 111 (1996).
21. R. Radespiel and R. C. Swanson, Progress with multigrid schemes for hypersonic flow problems, *J. Comput. Phys.* **116**, 103 (1995).
22. P. S. Vassilevski, Preconditioning nonsymmetric and indefinite finite element matrices, *J. Numer. Lin. Alg. Appl.* **1**, 59 (1992).
23. J. Xu, New class of iterative methods for non-self adjoint or indefinite problems, *SIAM J. Numer. Anal.* **29**, 303 (1992).
24. J. Xu, A novel two-grid method for semilinear elliptic equations, *SIAM J. Sci. Comput.* **15**, 231 (1994).
25. A. Ålund, P. Lotstedt, and M. Silen, Parallel single and multiple grid solution of industrial compressible flow problems, *SAAB Military Aircraft Report* L-0-1 R166, 15.11.1996.
26. O. Axelsson and M. Neytcheva, Scalable parallel algorithms in CFD computations, in *Computational Fluid Dynamics Review, 1995*, edited by M. Hafez and K. Oshima, p. 837.
27. O. Axelsson and M. Neytcheva, Some basic facts for efficient massively parallel computation, *CWI Quart.* **9**, 9 (1996).
28. G. S. Beavers, E. M. Sparrow, and R. A. Magnusson, Experiments on hydrodynamically developing flow in rectangular ducts, *Int. J. Heat Mass Transfer* **13**, 689 (1970).
29. R. J. Goldstein and D. K. Kreid, Measurement of laminar flow development in a square duct using a laser-doppler flowmeter, *J. Appl. Mech. Ser. E* **89**, 813 (1967).
30. P. M. de Zeeuw, *Acceleration of Iterative Methods by Coarse Grid Corrections*, Ph.D. thesis, CWI Amsterdam, 1996.
31. A. C. Humphrey, A. M. K. Taylor, and J .H. Whitelaw, Laminar flow in a square duct of strong curvature, *J. Fluid Mech.* **83**, 509 (1977).