INT. J. NUM. METH. HEAT FLUID FLOW, VOL. 2, 127–137 (1992)

FAST MULTIGRID METHOD FOR 3-D TURBULENT INCOMPRESSIBLE FLOWS

XUE-SONG BAI AND LASZLO FUCHS

Department of Gasdynamics, Royal Institute of Technology, S-100 44 Stockholm, Sweden

ABSTRACT

The averaged Navier–Stokes and the k–ε turbulence model equations are used to simulate turbulent flows in some internal flow cases. The discrete equations are solved by different variations of Multigrid methods. These include both steady state as well as time dependent solvers. Locally refined grids can be added dynamically in all cases. The Multigrid schemes result in fast convergence rates, whereas local grid refinements allow improved accuracy with rational increase in problem size. The applications of the solver to a 3-D (cold) furnace model and to the simulation of the flow in a wind tunnel past an object prove the efficiency of the Multigrid scheme with local grid refinement.

KEY WORDS Incompressible turbulent flows Multigrid methods Local grid refinements

INTRODUCTION

Accurate solutions to the Navier–Stokes equation require large scale computations. The number of node points depends on the desired resolution. If only global features of the flow field are of interest or/and one has to limit the calculations then some modelling equations have to be used to complement the basic (averaged) Navier–Stokes equations. Models often used for small scale variations (in space and time) are known as 'turbulence' models. Here, we use a more or less standard such model, namely the two equation k–ε equations. Even when turbulence models are used one has to resolve the geometrical and larger physical scales of the problem. The wide variation of scales is the primary reason for the difficulties in simulating accurately real flow fields. Beside resolution problems one is faced with non-linearity which in many cases causes difficulties because of singular or near singular Jacobians. Thus, to solve the Navier–Stokes equations one must use means to allow rational distribution of node points (to limit the number of the discrete equations) and also use methods that can handle efficiently the system of non-linear discrete equations.

Multiplegrid (MG) methods have been applied in the past 15 years for non-linear problems in computational fluid dynamics[1–7]. The main emphasis in most MG publications is the efficiency of the method in question. The efficiency of the method is practically established for incompressible and compressible test cases. For problems of applied character this efficiency has been questioned in some cases. Here, we apply the MG method to more realistic cases and to where experimental data are also available. The implemented code uses either V- or W-cycles in the FAS (full approximation storage) mode. Steady flows can be solved by using the basic scheme directly, or by pseudo-time marching method. Genuinely time-dependent problems are integrated implicitly, by solving the implicit problem by a full Multigrid (FMG) cycle. The efficiency of the code is not as high as those in pure test cases, but it is still considerably higher than those achieved by other single grid methods.

0961–5539/92/020127–11$2.00

Received September 1991

Using local grid refinements turned out to be very effective as has been also reported previously[8–11]. Associated with local mesh refinement is the question of when and where to refine. These issues are not often addressed and therefore one cannot make a real assessment of the efficiency and accuracy of the grid refined results. These questions have been addressed to some extent for incompressible flows[8] and for transonic flows[10]. Other issues of locally refined grids are local conservation properties and imposing global constraints. These issues can be handled naturally if the information exchange among the grids is done appropriately (conservatively).

Here, we demonstrate that our implementation of local grid refinements really improves efficiency in the sense that without loss of accuracy one does not have to use a globally fine grid, but rather a mix of locally fine and globally coarse grids.

MATHEMATICAL MODEL OF INCOMPRESSIBLE TURBULENT FLOWS

Turbulent flows contain various scales of eddies. For instance, for the flow in furnaces, the ratio of largest eddy and smallest eddy can be of the order of at least several thousands. It means that, for three-dimensional problems, mesh points of order $O(10^{12})$ have to be used in order to capture all scale of eddies. Currently it is impossible to handle such large scale problems even on modern supercomputers. On the other hand, this is not necessary either. For most engineering problems, only the averaged properties of the flow field are of interests, hence lower resolution may be used.

An often used approximation for turbulence is the two-equation k–ε model. The model is based on Boussinesq assumption by introducing a turbulent eddy viscosity to approximate the turbulent Reynolds stresses introduced in the averaging process. The (Reynolds) averaged Navier–Stokes equations and k–ε equations are the system of governing partial differential equations that has to be solved.

Governing equations

For three-dimensional incompressible turbulent flows, the Reynolds averaged Navier–Stokes equations, in cartesian coordinates can be written in conservative form as follows:

$$\frac{\partial u}{\partial t}+\frac{\partial(uu)}{\partial x}+\frac{\partial(vu)}{\partial y}+\frac{\partial(wu)}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial x}=\frac{\partial}{\partial x}\left(\nu_{\text{eff}}\frac{\partial u}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_{\text{eff}}\frac{\partial u}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_{\text{eff}}\frac{\partial u}{\partial z}\right) \tag{1}$$

$$\frac{\partial v}{\partial t}+\frac{\partial(uv)}{\partial x}+\frac{\partial(vv)}{\partial y}+\frac{\partial(wv)}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial y}=\frac{\partial}{\partial x}\left(\nu_{\text{eff}}\frac{\partial v}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_{\text{eff}}\frac{\partial v}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_{\text{eff}}\frac{\partial v}{\partial z}\right) \tag{2}$$

$$\frac{\partial w}{\partial t}+\frac{\partial(uw)}{\partial x}+\frac{\partial(vw)}{\partial y}+\frac{\partial(ww)}{\partial z}+\frac{1}{\rho}\frac{\partial p}{\partial z}=\frac{\partial}{\partial x}\left(\nu_{\text{eff}}\frac{\partial w}{\partial x}\right)+\frac{\partial}{\partial y}\left(\nu_{\text{eff}}\frac{\partial w}{\partial y}\right)+\frac{\partial}{\partial z}\left(\nu_{\text{eff}}\frac{\partial w}{\partial z}\right) \tag{3}$$

where u, v, w are the velocity components in x, y, z directions respectively and p is the pressure. The continuity equation is:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{4}$$

For turbulence closure, the two-equation k–ε model is:

$$\frac{\partial k}{\partial t}+\frac{\partial(uk)}{\partial x}+\frac{\partial(vk)}{\partial y}+\frac{\partial(wk)}{\partial z}=\frac{\partial}{\partial x}\left(\frac{\nu_{\text{eff}}}{\Gamma_k}\frac{\partial k}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_{\text{eff}}}{\Gamma_k}\frac{\partial k}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\nu_{\text{eff}}}{\Gamma_k}\frac{\partial k}{\partial z}\right)+S_k \tag{5}$$

$$\frac{\partial \varepsilon}{\partial t}+\frac{\partial(u\varepsilon)}{\partial x}+\frac{\partial(v\varepsilon)}{\partial y}+\frac{\partial(w\varepsilon)}{\partial z}=\frac{\partial}{\partial x}\left(\frac{\nu_{\text{eff}}}{\Gamma_\varepsilon}\frac{\partial \varepsilon}{\partial x}\right)+\frac{\partial}{\partial y}\left(\frac{\nu_{\text{eff}}}{\Gamma_\varepsilon}\frac{\partial \varepsilon}{\partial y}\right)+\frac{\partial}{\partial z}\left(\frac{\nu_{\text{eff}}}{\Gamma_\varepsilon}\frac{\partial \varepsilon}{\partial z}\right)+S_\varepsilon \tag{6}$$

k is turbulent kinetic energy, ε is dissipation rate of turbulent kinetic energy. Γ_k, Γ_ε are model constants, S_k, S_ε are source terms, such that:

$$S_k = \nu_t G - \varepsilon \tag{7}$$

$$S_\varepsilon = C_1 \frac{\varepsilon}{k} \nu_t G - C_2 \frac{\varepsilon^2}{k} \tag{8}$$

$$G = 2\left[\left(\frac{\partial u}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial y}\right)^2 + \left(\frac{\partial w}{\partial z}\right)^2\right] + \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right)^2 + \left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)^2 + \left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)^2 \tag{9}$$

In the above formulations, ν_{eff} is the so called 'effective viscosity coefficient', ν_t is turbulent eddy viscosity:

$$\nu_{\text{eff}} = \nu_L + C_\mu \frac{k^2}{\varepsilon}, \quad \nu_t = C_\mu \frac{k^2}{\varepsilon} \tag{10}$$

ν_L is kinematic laminar viscosity and C_μ is a model constant.

Boundary conditions

For u, v, w at wall and inflow boundries, Dirichlet boundary conditions are used ($u = v = w = 0$ at wall, given at inflow). At outflow boundary the reduced N–S equations are used[4]. k, ε boundary conditions at inflow are given, at outflow boundary zero second derivatives of k, ε are assumed, at wall boundaries the wall function model[12] is adopted.

$$\begin{aligned}
k: &\quad \frac{\partial k}{\partial n} = 0; \qquad \int_0^{y_p} \varepsilon_p \, dy = C_\mu k_p^{3/2} \ln(E y^+)/\kappa \\
\varepsilon: &\quad \varepsilon_p = C_\mu^{3/4} k_p^{3/2}/(\kappa y_p) \\
\nu_{\text{eff}}: &\quad \nu_{\text{eff}} = \frac{C_\mu^{1/4} k_p^{1/2} \kappa y_p}{\ln(E y^+)} \qquad \text{for} \quad y^+ > 11.63 \\
&\quad \nu_{\text{eff}} = \nu_L \qquad\qquad\qquad\quad \text{for} \quad y^+ \leqslant 11.63 \\
y^+, u^*: &\quad y^+ = \frac{u^* y_p}{\nu_L}, \qquad u^* = V_p \kappa / \ln\left(\frac{E u^* y_p}{\nu_L}\right)
\end{aligned} \tag{11}$$

Subscript p denotes the first grid point near wall. V_p is the velocity component at point p parallel to wall, y_p is distance between the first grid point and wall, n is the direction normal to wall. E and κ are constants. u^* is the wall shear velocity, y^+ is the dimensionless distance from the solid wall.

In the above formulations, model constants are determined from experiments. Here we use[12]: $C_1 = 1.44$, $C_2 = 1.92$, $\Gamma_k = 1.0$, $\Gamma_\varepsilon = 1.3$, $C_\mu = 0.09$, $E = 9.0$, $\kappa = 0.435$.

SOLUTION PROCEDURE

The numerical solution procedure includes three different Multigrid strategies for different flow situations. Each Multigrid strategy is related to its own discretization, which is performed on a grid system with local refinements. This section also includes the numerical treatment of boundary conditions.

Grid systems

In order to resolve the small length scales in some flow regions (e.g. boundary layer), or details of the geometry itself, one must use fine grid distribution, while in other region (e.g. mainflow) one may use less fine grids. In order to solve the problem satisfactorily, one may either use very fine grid distribution everywhere, which would be very costly both in terms of computer resources,

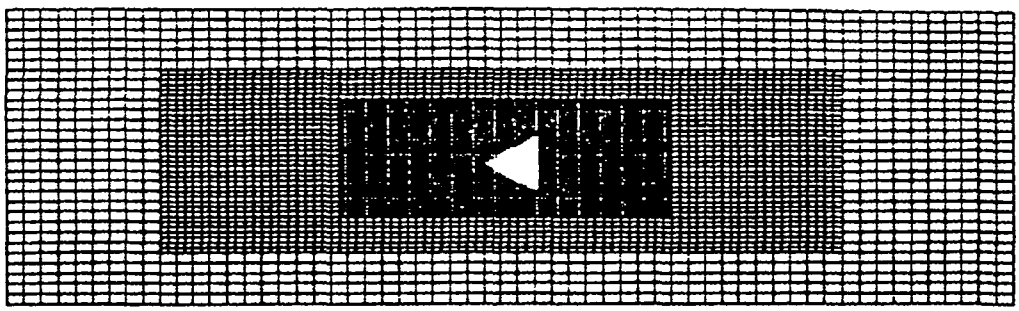

Figure 1 A local refined mesh past a triangular obstacle

or use stretch grids which is non-optimal in reducing the number of unknowns. This method may also decrease the accuracy of the numerical discrete approximation, and often also reduces the convergence rate. Local grid refinements can treat this problem more adequately. For regions with small length scales, refined grids can be added, while in other regions the global coarse grid can be used. Such a grid system is shown in *Figure 1*. The local grid refinement may be isotropic (i.e. in all directions), or non-isotropic (i.e. in one or two directions only) as in case of boundary layers.

Based on the discussion above, we use rectangular grid system in our code. The nodes in each direction are evenly spaced. Such a grid system only requires the storage of three integers (number of node points in three directions) and three real variables (mesh sizes in three directions). In regions with small length scale locally refined grids are added. To define this local grid one needs seven integers and three reals. There are no theoretical restrictions in repeatedly introducing further levels of locally refined grids. In our code we have used up to 6 levels of locally refined grids.

Time dependent flows

The time–space discretization of (1)–(6) can be performed by using various methods, which may be classified into three main types: implicit methods, explicit methods, and semi-implicit splitting methods. von Neumann analysis shows that explicit methods limit the time step to be proportional to the square of spatial mesh size (as long as viscous terms are of any importance). This is a very severe limitation, imposing very small time steps when the grid is refined. Implicit methods do not impose such time step limitation. However, in such situations one has to solve an implicit problem. In cases when for physical reasons one may use long time steps, a fully implicit method may be most appropriate. Time integration can be done by using a FMG (with a single MG cycle) step to solve the implicit problem. In cases when the time step is limited for physical reasons, the dimensionless time scale is often of the order of the dimensionless spatial scale. In such cases we prefer a splitting scheme: implicit for the diffusive terms and explicit for the convective terms. Take (1) as an example, at point (i,j,l) the discretized equation is:

$$\frac{u^{n+1}-u^n}{\Delta t}+\Delta_x^+(uu)^n+\Delta_y^+(vu)^n+\Delta_z^+(wu)^n+\frac{1}{\rho}\Delta_x p^{n+1}$$
$$=\Delta_x(\nu_{\text{eff}}\Delta_x u)^{n+1}+\Delta_y(\nu_{\text{eff}}\Delta_y u)^{n+1}+\Delta_z(\nu_{\text{eff}}\Delta_z u)^{n+1} \quad (12)$$

where $\Delta_x, \Delta_y, \Delta_z$ are central difference operators, $\Delta_x^+, \Delta_y^+, \Delta_z^+$ are upwind difference operators for the first spatial derivatives. Δt is the time step. Put nth level terms (mainly convective terms) into right hand side F_u, let superscript m denote the grid level, (12) can be represented by:

$$L_u^m u^m-\frac{1}{\rho}\Delta_x p^m=F_u^m \quad (13)$$

where L_u is an elliptic difference operator. Similar formulae are used to discretize the other two momentum equations and k–ε equations. The continuity equation is discretized using implicit central difference. The effective viscosity v_{eff} is updated using (10) (at grid level m):

$$L_v^m v_{\text{eff}}^m \equiv v_{\text{eff}}^m - \left(C_\mu \frac{k^2}{\varepsilon}\right)^m - v_L = F_v^m \tag{14}$$

At finest grid (denoted by superscript M), $F_v^M = 0$. At grid level m, let

$$U = (u\ v\ w\ p\ k\ \varepsilon\ v_{\text{eff}})^{\mathrm{T}}, \qquad F = (F_u\ F_v\ F_w\ F_m\ F_k\ F_\varepsilon\ F_v)^{\mathrm{T}} \tag{15}$$

the whole discretized system can be represented as:

$$L^m U^m = F^m \tag{16}$$

The diagonal form of L^m that has been used in this study, has the form:

$$L = \begin{pmatrix} L_u & 0 & 0 & -\Delta x/\rho & 0 & 0 & 0 \\ 0 & L_v & 0 & -\Delta_y/\rho & 0 & 0 & 0 \\ 0 & 0 & L_w & -\Delta_z/\rho & 0 & 0 & 0 \\ \Delta_x & \Delta_y & \Delta_z & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_\varepsilon & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & L_v \end{pmatrix}$$

At finest grid ($m = M$), F^M is known from nth time level solution. The difference operator L is elliptic, hence can be solved by the FAS (full approximation storage scheme) very efficiently.

The Multigrid method consists of a so-called 'smoother' and data transfer procedures among the different grid levels. The transfer of residuals from grid level m to level $m-1$, and the transfer of the dependent variables from level m to level $m-1$ is done by volume averaging. The transfer of the corrections from level m to level $m+1$ is done by trilinear interpolation. For more details see References 4, 8 and 10.

The relaxation method for system (16) at each grid level is known as the 'smoother'. In the present smoother the pressure field is coupled to velocity field in the following way: First, consider the correction problem of (16), at grid level m:

$$L\Delta U = R \tag{17}$$

where $\Delta U = U^{q+1} - U^q$ is the correction of dependent variables between qth iteration and $(q+1)$th iteration. $R = F - LU^q$ is residual of equation at qth iteration. ΔU is split as:

$$\Delta U = \begin{pmatrix} 1 & 0 & 0 & \Delta_x/\rho & 0 & 0 & 0 \\ 0 & 1 & 0 & \Delta_y/\rho & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta_z/\rho & 0 & 0 & 0 \\ 0 & 0 & 0 & L_u & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \Delta U^* \tag{18}$$

with $\Delta U^* = (\Delta u^*\ \Delta v^*\ \Delta w^*\ \phi\ \Delta k^*\ \Delta\varepsilon^*\ \Delta v_{\text{eff}}^*)^{\mathrm{T}}$. Then from (17) and (18) we have (assuming that

L_u, L_v, L_w are identical):

$$\begin{pmatrix} L_u & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_v & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & L_w & 0 & 0 & 0 & 0 \\ \Delta_x & \Delta_y & \Delta_z & Q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & L_k & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & L_\varepsilon & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & L_\gamma \end{pmatrix} \Delta U^* = R \tag{19}$$

where $Q=(\Delta_{xx}+\Delta_{yy}+\Delta_{zz})/\rho$, $\Delta_{xx}, \Delta_{yy}, \Delta_{zz}$ are central difference operators for the second spatial derivatives. To summarize, the relaxation step at level m consists of two stages: from (19) to get ΔU^*, from (18) to get ΔU. This procedure is continued until the errors are smoothed out.

V- (or *W*-) cycles can be used in the MG process. A *V*-cycle is, from grid level M to $M-1$ to ... to 2 to 1, then back to 2 to ... to $M-1$ to M. After several (usually one or two) cycles a converged solution at time step $n+1$ is obtained.

Acceleration of convergence to steady state (time–space MG)

Using the time marching method proposed above, steady state solution can be achieved for steady problems after marching enough time steps. In order to accelerate this process further, we have used another MG cycle. The procedure is, after several time steps, to compute the residuals of the steady state equations and put the residuals and dependent variables into a coarser grid level. From this coarse grid level several time steps being taken (applying the MG method (12)–(19) to the implicit part). During this time marching, the maximum time step can be doubled since the grid size itself has been doubled. This process is continued until the coarsest grid level is reached. Then the correction informations are transferred from coarse grid to finer grid until the finest grid is reached. This procedure is repeated until a steady state solution is found.

Steady Multigrid method

Consider (1) as an example. The discretized equation is:

$$\alpha\frac{\Delta u}{\Delta t}+L_*(\Delta u)=F\equiv L_*(u)+\frac{1}{\rho}\Delta_x p \tag{20}$$

where Δu is the correction of u, α is a relaxation parameter. The 'time dependent' term acts only as an under relaxation factor. L_* is the difference operator:

$$L_*(\phi)=\Delta_x^+(u\phi)+\Delta_y^+(v\phi)+\Delta_z^+(w\phi)-\Delta_x(\nu_{\text{eff}}\Delta_x\phi)-\Delta_y(\nu_{\text{eff}}\Delta_y\phi)-\Delta_z(\nu_{\text{eff}}\Delta_z\phi)$$

similar treatment is applied to the k–ε equations. The discretized equations are coupled in a similar way as (17)–(19) and then solved by *V*- (or *W*-) cycle FAS Multigrid cycles.

Symmetric successive point relaxation (SSPR) has been used for relaxation (e.g. in solving (19) for ΔU^*) in the above-mentioned three procedures. The total work unit count may be larger than for methods like line relaxation or plane relaxations, but total CPU time has often been found shorter in most general cases.

Treatment of boundary conditions

The wall boundary conditions for k, ε (wall function), are implemented in a way different from that of single grid methods. Instead of adding wall function explicitly, which is used in most single grid codes, the method used here updates k, ε at wall point in the way such that k, ε satisfy wall function (11) at point next to the wall. This step is only performed on the finest grid level. k, ε at wall points on coarse grid levels are updated by the MG data transfer procedures.

The boundary conditions on the local grid are handled depending on the location of the local grid point. The variable values at 'boundary points' that do not belong to a physical boundary are assigned by interpolation from a coarser grid that contains the local grid. In order to get a converged local grid solution, it is mandatory to maintain the mass conservation on all grids. The interpolation order is supposed[9] to be $m+p$, where m is the order of partial differential equation, p is the order of numerical approximation. In our study, the same trilinear interpolation scheme as the one used in the MG procedure turned out to be adequate also for local grid 'boundary points' value updates. The physical boundary conditions are applied on the finest possible (local or global) available grid.

NUMERICAL RESULTS

In order to evaluate the efficiency of the present schemes, their convergence characteristics are compared. Consider the flow on a (cold) furnace model. A single inlet is placed at some distance from the floor and the outlet is placed in the ceiling. *Figure 2* shows a typical velocity vector field of this case. Uniform rectangular grids are used. In order to examine the grid influence two grids are used, i.e. (1) $22 \times 14 \times 14$, using 3 levels; (2) $42 \times 26 \times 26$, with 4 levels. The Reynolds number based on inlet width is 7000. Results from various schemes are shown in *Figures 3–6*.

Figure 3 shows the time dependent flow convergence behaviour based on grid $42 \times 26 \times 26$. The results are calculated by using the time splitting scheme (12): the broken line depicts the convergence history using single grid relaxation to solve the implicit part, the solid line is for the time-dependent MG procedure of the previous section. The horizontal axis shows work units. One work unit is defined to be equal to the computational effort on the finest grid level. The vertical axis is the averaged corrections of the dependent variables on the finest grid. In *Figure 3* the convergence criterion in each time step is that the averaged correction less than 1.0^{-5}. The Figure demonstrates clearly that the MG method is superior to the single grid method

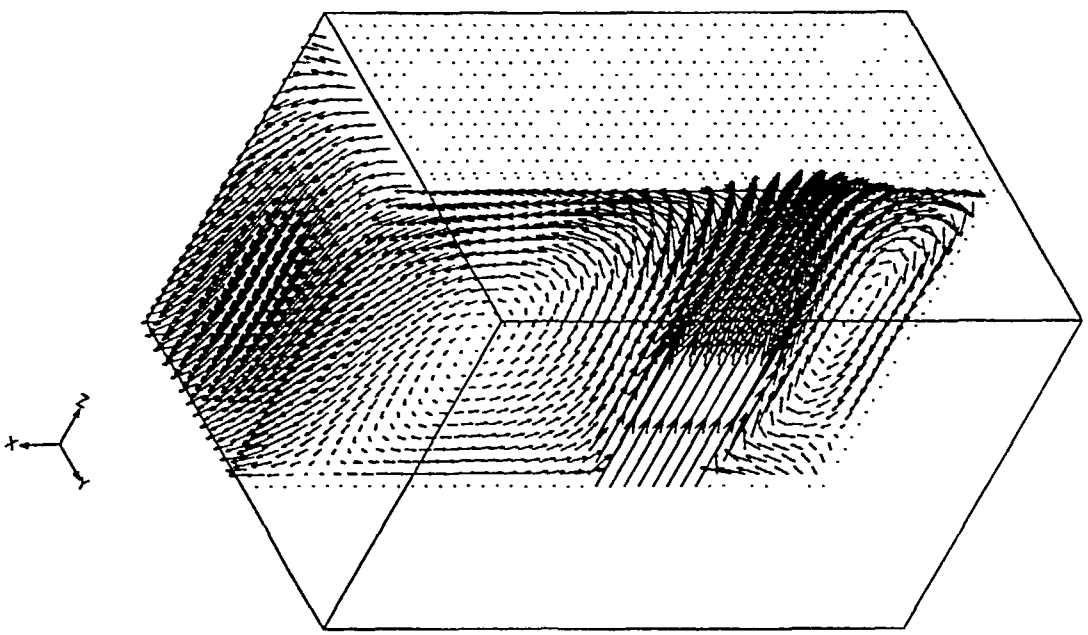

Figure 2 Velocity vector field in a furnace

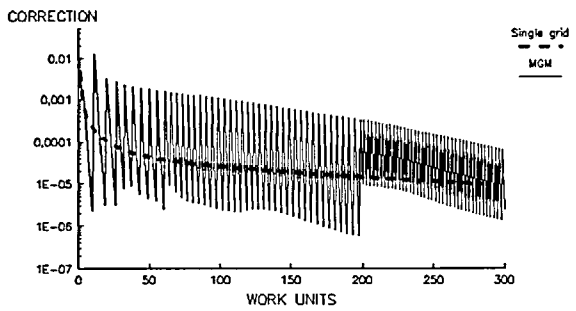

Figure 3 Convergence history of time dependent flow. ---, single grid; ——, MG method

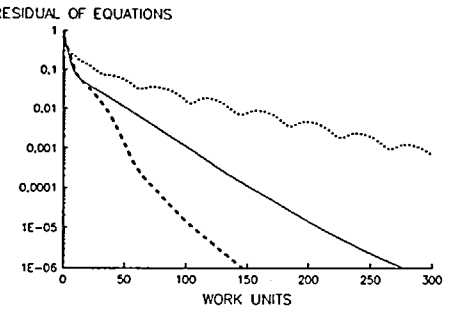

Figure 4 Convergence history on coarse grid (22 × 14 × 14). ——, MG method; ---, time-space MG method; ·····, single grid method

(in each time step MG method has convergence rate about 0.2–0.3, while the single grid method about 0.997). After 280 work units, the single grid scheme completed one time step, while MG method marched 75 time steps. In actual calculations one would impose less severe stopping criterion on the MG solver, to a single FMG cycle (or at most 2 cycles). This implies that the MG method is at least 75 times faster than single grid method in this case.

Figures 4–6 show the convergence histories of the present methods to steady state. The convergence criterion in each case is that the steady state residual norm is reduced by six orders of magnitude, i.e.

$$\bar{R}_{\text{final}} \leqslant 10^{-6} \bar{R}_{\text{initial}}$$

where $\bar{R}$ is defined as:

$$\bar{R} = \sqrt{\frac{R_u^2 + R_v^2 + R_w^2 + R_m^2 + R_k^2 + R_\varepsilon^2}{6N_P}}$$

where $R_u, R_v, R_w, R_m, R_k, R_\varepsilon$ are the sums of residuals of the steady state part of x-momentum, y-momentum, z-momentum equations, continuity equation, and k–ε equations. N_P is the total number of unknowns.

Figure 4 shows the results based on a grid $22 \times 14 \times 14$. The solid line is the result from time-dependent MG procedure described earlier. That is, the implicit part is solved by a MG acceleration. The dotted line is the single grid result, which uses the splitting scheme, but no MG cycles are used to solve the implicit part (solved by single grid iterations). The broken line corresponds to the time-space MG scheme of the previous section. *Figure 5* shows the results based on the fine grid ($42 \times 26 \times 26$). The three lines mark similar cases as in *Figure 4*.

These Figures indicate that the usage of MG procedure does improve the convergence to a steady state solution. The usage of MG procedure when solving the implicit part of (12) leads to the faster convergence rate than with single grid method, while the result of the time–space MG is the best among the above-mentioned three schemes. However, it is also noted that the convergence rate is dependent on the grid size (*Figures 4* and *5*). This is due to the time marching step limitation properties. The grid dependent convergence behaviour can be improved by using the steady Multigrid method as seen in *Figure 6*. The solid line corresponds to the coarser grid ($22 \times 14 \times 14$) and the broken line to the finer grid ($42 \times 26 \times 26$). As seen, the convergence rates of the two grids are nearly equal. The residuals are reduced by 6 orders of magnitudes within 70 work units. Single grid results are also depicted in *Figure 6* for comparison. The dotted line corresponds to coarser grid and the long broken line to the finer grid.

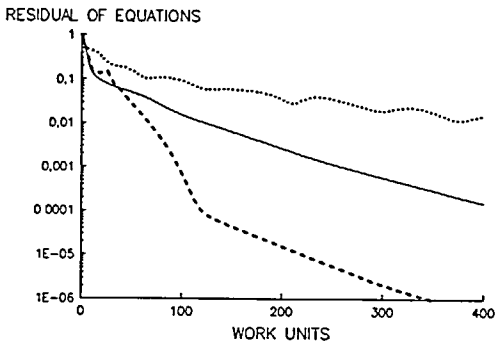

Figure 5 Convergence history on uniformly fine grid (42 × 26 × 26). —, MG method; – – –, time-space MG method; ·····, single grid method

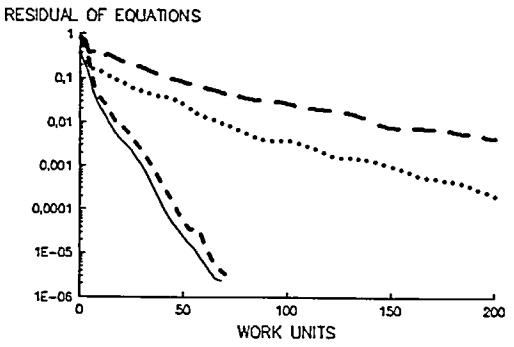

Figure 6 Convergence history based on steady MG method. —, MG method (coarser grid); – – –, MG method (finer grid); ·····, single coarser grid; — — —, single finer grid

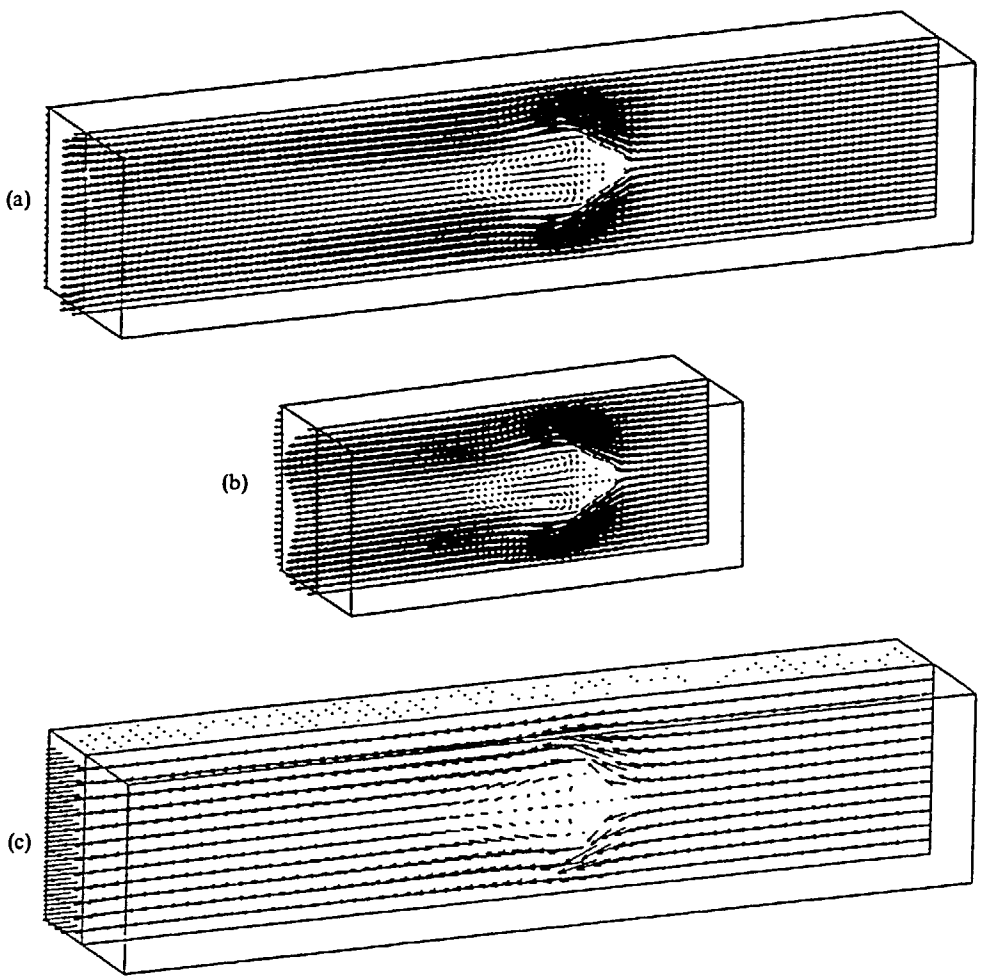

Figure 7 Velocity vector field in various grids. (a) Global fine grid; (b) local fine grid; (c) global coarse grid

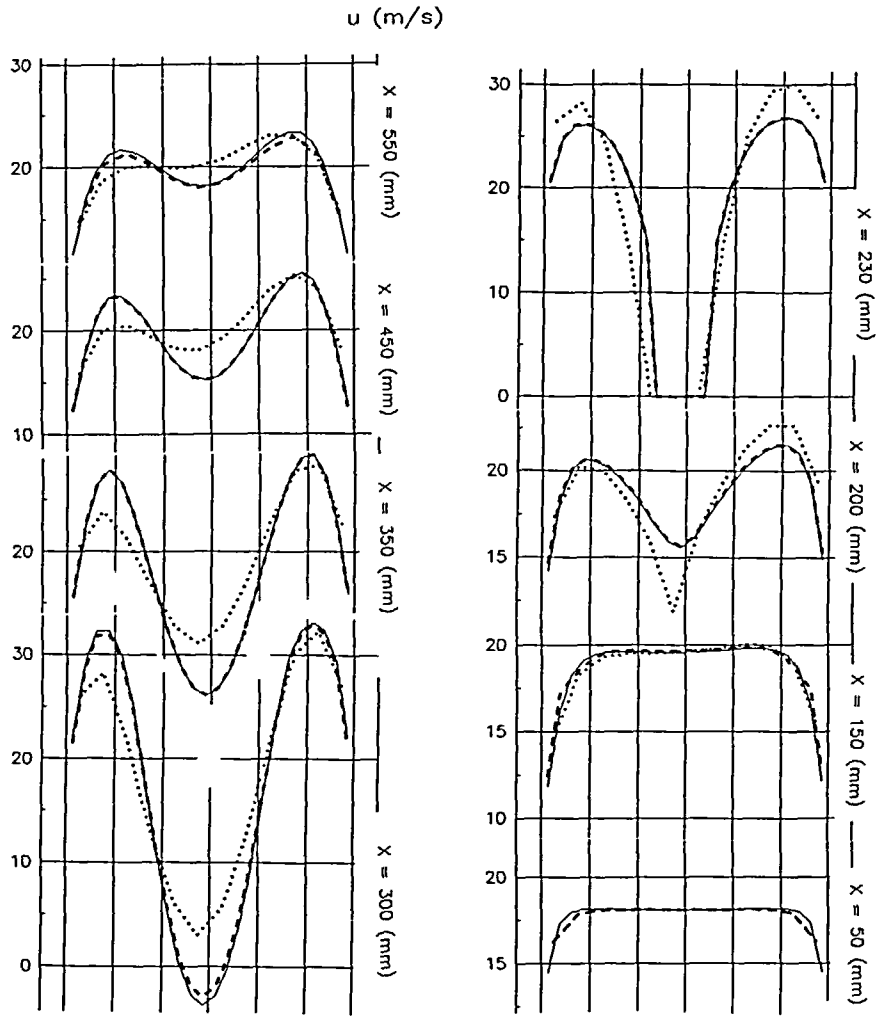

Figure 8 *u*-velocity component profiles at different cross-section. ·····, Coarse grid; – – –, coarse grid with local fine grid; ——, fine grid

The above results show the improvement in convergence rate due to MG procedures. The efficiency can be further improved by using local grid refinements. *Figure 7* shows a case of low speed wind tunnel flow with a triangular obstacle placed in the middle of the tunnel. The Reynolds number in this case is 0.12×10^6 based on the height of the tunnel. Three grids have been used: (1) a global coarse grid $62 \times 14 \times 14$, 3 levels; (2) a global fine grid $122 \times 26 \times 26$, 4 levels and (3) a global coarse grid $62 \times 14 \times 14$, 3 levels with one level local grid refinement $62 \times 26 \times 26$.

Figure 7 shows the velocity vector field in symmetry plane. *Figure 8* shows the *u*-velocity component distribution in various sections along main flow direction. The solid lines in *Figure 8* are the results of the global fine grid, the broken lines corresponds to the global coarse grid with the local grid refinement, the dotted lines are the results of the global coarse grid. *Figure 8* shows that the result on the combined coarse grid and locally refined grid has the same accuracy as that on the global fine grid, both are better than result on the global coarse grid.

In the case with local grid refinement, the required computer memory and CPU times are halved compared to the case with the global fine grid.

CONCLUSIONS

It has been shown that considerable improvements in efficiency can be achieved for turbulent flow calculations in three space dimensions by employing the MG procedures and local grid refinements. From the results shown in this paper we may conclude that:

(1) for time accurate flows, the splitting scheme is very efficient after implementing MG procedures. In each timestep, using a single grid method the convergence rate is about 0.997, with MG acceleration it can be 0.2–0.3;
(2) when steady state or time-periodic solutions are sought, the time accurate MG method can be accelerated by using the time–space MG procedure;
(3) the steady MG procedure seems to be faster than the time marching methods when applied to steady state problems;
(4) without loss of accuracy, further improvement in efficiency can be achieved by employing local grid refinements.

ACKNOWLEDGEMENTS

We appreciate the financial support of the Swedish National Board for Technical Development (STU) as part of grants 87-05306 and 89-2269.

REFERENCES

1 Brandt, A. Multi-level adaptive solutions to boundary-value problem, *Math. Comput. Phys.*, **31**, 330–390 (1977)
2 Hackbusch, W. *Multi-Grid Methods and Applications*, Springer Verlag, Berlin (1985)
3 Fuchs, L. Finite difference methods for plane steady inviscid transonic flows, *TRITA-GAD-2* (1977)
4 Fuchs, L. and Zhao, H. S. Solution of three-dimensional viscous incompressible flows by a Multi-grid method, *Int. J. Num. Meth.Fluids*, **4**, 539–555 (1984)
5 Demuren, A. O. Application of Multi-grid methods for solving the Navier–Stokes equations, *NASA TM 102359* (Oct. 1989)
6 Lien, F.-S. and Leschziner, M. A. Multigrid convergence acceleration for complex flow including turbulence, *Multigrid Methods*, Vol. III (Ed. W. Hackbusch and U. Trottenberg), Birkhauser Verlag, pp. 277–288 (1991)
7 Fuchs, L. A Newton Multi-grid method for the solution of nonlinear partial differential equations, *Proc. BAIL-I Conf.* (Ed. J. J. Miller), Boole Press, pp. 291–296 (1980)
8 Fuchs, L. A local mesh-refinement technique for incompressible flows, *Comp. Fluids*, **14**, 69–81 (1986)
9 Bai, D. and Brandt, A. Local mesh refinement multilevel techniques, *SIAM J. Sci. Stat. Comput.*, **8**, 109–134 (1987)
10 Gu, C. Y. and Fuchs, L. Accurate numerical computation of airfoil flows using cartesian coordinates. *Proc. 4th Int. Conf. Numerical Methods in Laminar and Turbulent Flows*, pp. 468–479, Pineridge Press, Swansea (1985)
11 Fuchs, L. Solution of 3-D problem using overlapping grids and multi-grid methods, *Multigrid Methods*, Vol. III (Ed. W. Hackbusch and U. Trottenberg), Birkhauser Verlag, pp. 167–177 (1991)
12 Launder, B. E. and Spalding, D. B. The numerical computation of turbulent flow, *Comp. Meth. Appl. Mech. Eng.*, **3**, 269–289 (1974)