INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2006; **50**:579–596
Published online 7 September 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1063

Multiple semi-coarsened multigrid method with application to large eddy simulation

F. E. Ham[1,‡], F. S. Lien[2,*,†] and A. B. Strong[2,§]

[1]*Center for Integrated Turbulence Simulations, Stanford University, Stanford, CA 94305, U.S.A.*
[2]*Department of Mechanical Engineering, University of Waterloo, University Avenue West, Waterloo, Ontario, Canada N2L 3G1*

SUMMARY

The Multiple Semi-coarsened Grid (MSG) multigrid method of Mulder (*J. Comput. Phys.* 1989; **83**:303–323) is developed as a solver for fully implicit discretizations of the time-dependent incompressible Navier–Stokes equations. The method is combined with the Symmetric Coupled Gauss–Seidel (SCGS) smoother of Vanka (*Comput. Methods Appl. Mech. Eng.* 1986; **55**:321–338) and its robustness demonstrated by performing a number of large-eddy simulations, including bypass transition on a flat plate and the turbulent thermally-driven cavity flow. The method is consistently able to reduce the non-linear residual by 5 orders of magnitude in 40–80 work units for problems with significant and varying coefficient anisotropy. Some discussion of the parallel implementation of the method is also included. Copyright © 2005 John Wiley & Sons, Ltd.

KEY WORDS: MSG multigrid; symmetric coupled Gauss–Seidel; LES; incompressible Navier–Stokes

1. INTRODUCTION

Implicit discretizations of the Navier–Stokes equations are known to have favourable properties relative to their explicit counterparts. The most obvious is the relaxation or removal of the numerical stability constraint on the computational time step. With respect to large eddy simulation (LES), where a broad range of spatial and temporal scales must be accurately resolved, there are a number of other more subtle benefits. For example, the implicit discretization of the incompressible or variable-density form of the Navier–Stokes equations results in an operator symmetry in space and time that can have very favourable kinetic

*Correspondence to: F. S. Lien, Department of Mechanical Engineering, University of Waterloo, University Avenue West, Waterloo, Ontario, Canada N2L 3G1.
†E-mail: fslien@uwaterloo.ca
‡E-mail: fham@stanford.edu
§E-mail: astrong@sunwise.uwaterloo.ca

Contract/grant sponsor: Natural Sciences and Engineering Research Council of Canada (NSERC)

Received 19 January 2005
Revised 11 July 2005
Accepted 11 July 2005

energy conservation properties [1, 2]. The coupled implicit treatment of pressure also avoids the splitting errors associated with fractional step methods that can damp turbulence when larger time steps are used [1]. The implicit treatment of pressure in the fully compressible form of the Navier–Stokes equations results in a method capable of capturing long wavelength acoustic waves at low Mach number with zero artificial damping and no acoustic CFL limit [3]. For implicit methods to gain broad acceptance for LES, however, these benefits must be accompanied by fast, scalable solvers for the resulting global systems. At present, it is most common to solve these systems by wrapping a more standard projection–correction method in an outer loop and performing a number of outer iterations to converge each time step. Such a factorization can inadvertently result in very stiff systems, and the number of outer iterations required to converge the system in each time step can vary significantly.

In 1986, Vanka proposed a coupled multigrid method for the solution of the Navier–Stokes equations on structured, staggered grids [4]. The method used a coupled smoother on all grid levels that has come to be known as Symmetric Coupled Gauss–Seidel (SCGS) [5–7]. The method was shown to scale optimally (linearly) in terms of both work and storage for the problems investigated. For multigrid methods such as Vanka's, involving point-relaxation smoothers and isotropic coarsening, the convergence factor is known to degenerate dramatically in the presence of coefficient anisotropy. Coefficient anisotropy can result from large cell aspect ratios, anisotropic material properties, or asymmetric operators. In the case of LES, it is most commonly due to the highly stretched grids used to resolve turbulent boundary layers.

Since the time of Vanka's contribution, a number of solutions have been proposed to improve the robustness¶ of multigrid methods in the presence of large coefficient anisotropy. Most of these solutions involve some combination of the following two ideas: (1) the use of semi-coarsening, where the coarsening is not isotropic, and (2) improvements to the smoother so that both high and low frequency components of the residual distribution are effectively reduced in at least one or two directions (i.e. the so-called line and plane smoothers). When the semi-coarsening and/or smoother improvements are properly matched, multigrid methods recover their optimality, even in the presence of strong coefficient anisotropies.

In the present work, we restrict the discussion to multigrid methods that maintain a structured topology on all grid levels. Structured coarse grids significantly simplify implementation by allowing the same smoother (and the same discretization for the case of FAS multigrid) on all grid levels, and will generally result in more efficient codes by improving cache performance and avoiding the indirect memory addressing associated with unstructured coarse grids. Two methods have been widely applied that attempt to solve the anisotropic coefficient problem using structured coarse grids. The first is the combination of standard isotropic coarsening and alternating plane smoothing (e.g. isotropic coarsening followed by plane smoothing sweeps in the xy, xz, and then yz-directions) [8]. The second is a combination of semi-coarsening in a fixed direction with plane-smoothing in the perpendicular direction (e.g. z-semi-coarsening followed by xy-plane smoothing) [9]. This latter is even included as one of the 'grey box' multigrid solvers available in Lawrence Livermore National Lab's Hypre library [10]. The ability of these two techniques to solve the incompressible 3D Navier–Stokes equations on

¶The word 'robust' appears throughout the multigrid literature. Here we use it to describe a multigrid method for which the convergence rate per work unit is independent of grid size, stretching factor, and any coefficient anisotropies.

highly stretched grids was the subject of a recent investigation by Montero *et al*. [11]. Interestingly, they also used the same smoother as we do (Section 3.5), so their investigation is particularly relevant. Their conclusions can be summarized as follows: (1) For higher Reynolds numbers and highly stretched meshes in all three directions, the combination of isotropic coarsening and alternating-plane smoothing is not robust, and can even fail to converge. (2) The combination of semi-coarsening and plane-smoothing gave the best overall performance, but was described as 'fully robust' in the case of only one of the problems investigated (a laminar flat plate boundary layer), and in that case, for only one specific direction of semi-coarsening and plane smoothing (semi-coarsening in the wall-normal z-direction combined with xy-plane smoothing). Semi-coarsening in other directions was found to actually deteriorate the smoothing. In a sense, then, even the combination of semi-coarsening with plane smoothing cannot be considered robust because it requires some foreknowledge of the optimal semi-coarsening direction for the particular problem under investigation.

The reason why semi-coarsening multigrid solvers are not robust can be simply understood. In general, the optimal semi-coarsening direction will vary throughout the domain, and semi-coarsening or even isotropic coarsening that is appropriate in one part of the domain may be inappropriate or even detrimental in another. Thus it becomes impossible to accommodate all locally optimal coarsening directions with a single structured coarse grid (see Figure 1). As a solution to this problem, Mulder proposed a multigrid method based on multiple semi-coarsened grids (MSG) [12]. His MSG multigrid method forms the basis of the present contribution. In the following section, we present the incompressible Navier–Stokes equations as the system being solved, and describe their discretization on a structured staggered grid as a generalized algebraic system. The MSG multigrid method and its application to this system is described in Section 3. In Section 4, we present the results of the method applied to a number of problems, including flat-plate transition and buoyancy-driven cavity flows. Finally, some details of the parallel implementation are given in Section 5. To the authors' knowledge, this is the first time MSG multigrid has been used to solve the Navier–Stokes equations with application to LES.

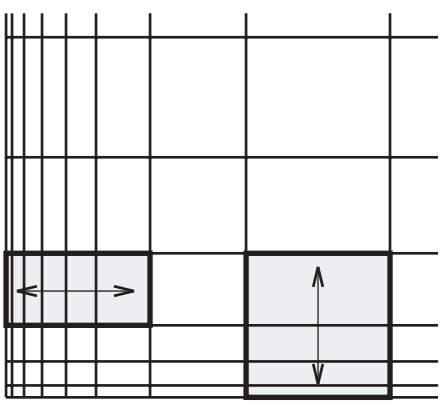

Figure 1. Example of how variations in the optimal direction of coarsening brought about by geometric anisotropy will, in general, result in unstructured coarse grids. Arrows indicate the preferred direction of semi-coarsening.

2. GOVERNING EQUATIONS

The equations of motion for the spatially filtered incompressible velocity and pressure fields (suitable for LES) are given by

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_j \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left((\nu + \nu_{\text{sgs}})\left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j}\right)\right) \tag{1}$$

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{2}$$

where the overbar represents spatial filtering on the scale of the grid. Equation (1) assumes Boussinesq dynamics to approximate the subgrid stresses, where the subgrid viscosity, ν_{sgs}, is given by the Smagorinsky closure

$$\nu_{\text{sgs}} = C\bar{\Delta}^2\sqrt{2\bar{S}_{ij}\bar{S}_{ij}} \tag{3}$$

In Equation (3), the resolved strain rate tensor is defined

$$\bar{S}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_j}{\partial x_i} + \frac{\partial \bar{u}_i}{\partial x_j}\right) \tag{4}$$

and the grid filter width is defined in terms of the local grid spacing

$$\bar{\Delta} = (\bar{\Delta}_x \bar{\Delta}_y \bar{\Delta}_z)^{1/3} \tag{5}$$

The Smagorinsky constant, C, is calculated using the dynamic procedure [13, 14] with averaging in the homogeneous direction(s) to avoid the numerical instability associated with large negative C values. In addition, the total viscosity $(\nu + \nu_{\text{sgs}})$ is not allowed to be negative.

In the present contribution, Equations (1) and (2) are discretized using the conservative second-order finite volume method of Ham *et al.* [1] on a structured Cartesian grid with staggered arrangement of velocity and pressure. Figure 2 illustrates the arrangement of variables on the grid in both space and time. Note that the subscripts now refer to the Cartesian indices of the variables (not to be confused with the Cartesian tensor notation used previously).

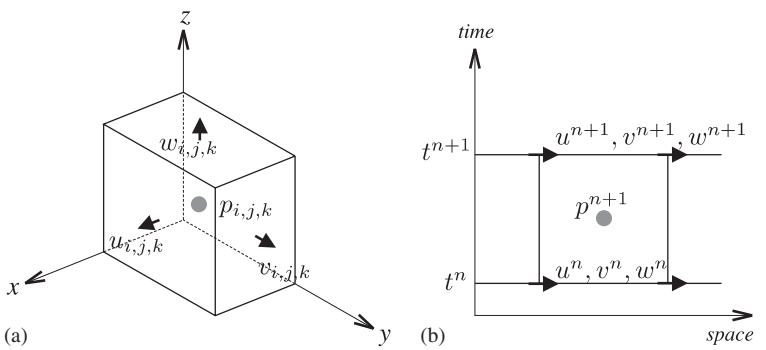

Figure 2. Staggered variable arrangement in: (a) space; and (b) space/time.

Using Newton linearization, the set of coupled algebraic equations for the velocity and pressure corrections can be written as follows. For the x-momentum equation

$$A^u_{ijk}u'_{ijk} + A^{u,u_W}_{ijk}u'_{i-1jk} + A^{u,u_E}_{ijk}u'_{i+1jk} + A^{u,u_S}_{ijk}u'_{ij-1k} + A^{u,u_N}_{ijk}u'_{ij+1k} + A^{u,u_B}_{ijk}u'_{ijk-1} + A^{u,u_T}_{ijk}u'_{ijk+1} + A^{u,p_P}_{ijk}p'_{ijk} + A^{u,p_E}_{ijk}p'_{i+1jk} + B^u_{ijk} = 0 \quad (6)$$

with similar equations for y and z-momentum. We note that the coefficients themselves depend on the solution, and thus must be recalculated throughout the iteration. This is performed at the start of every multigrid cycle. The continuity equation can be written as

$$A^{c,u_w}_{ijk}u'_{i-1jk} + A^{c,u_e}_{ijk}u'_{ijk} + A^{c,v_s}_{ijk}v'_{ij-1k} + A^{c,v_n}_{ijk}v'_{ijk} + A^{c,w_b}_{ijk}w'_{ijk-1} + A^{c,w_t}_{ijk}w'_{ijk} + B^c_{ijk} = 0 \quad (7)$$

A source term has been included in the continuity equation, although it will be zero on the fine grid. The efficient and robust solution of this coupled algebraic system using MSG multigrid is the focus of this contribution.

3. THE MSG MULTIGRID METHOD

The MSG multigrid method of Mulder [12] has two unique features that differentiate it from other multigrid methods: (1) all possible directions of semi-coarsening associated with a given fine grid are considered by using a specially-designed array of semi-coarsened grids, and (2) the optimal combination of coarse grid corrections to any given fine grid is achieved by using anisotropic prolongation operators that mirror the local anisotropy in the fine grid coefficients. In the following subsections, the various components of the method are developed for the algebraic system described previously.

3.1. The coarse grid array

All computations carried out as part of the present study are three-dimensional and involve an array of coarse grids that include all three directions of semi-coarsening associated with a given fine grid. For the purposes of illustration only, Figure 3 shows an array of coarse grids associated with a 2D 8×8 fine grid. The arrows indicate how the grids are linked by restriction (down arrows) and prolongation (up arrows). To prevent the amount of work and storage on the coarse grids from becoming prohibitive, grids of the same size at a given level are combined as a single grid. For example, the single 4×4 grid at multigrid level 2 is associated with both the y-semi-coarsening of the 4×8 grid and the x-semi-coarsening of the 8×4 grid at level 1. In this way, the linear scaling of storage requirement characteristic of multigrid methods can be maintained. Mulder showed that for large 2D grids, the total storage requirement approaches 4 times the fine grid requirement. When a similar array of semi-coarsened grids is constructed for a large 3D grid, the total storage requirement is found to approach 8 times the fine grid requirement.

3.2. Cycle type

For MSG multigrid, the V-cycle is generally preferred because of the relatively large number of coarse grids. In any given V-cycle each grid is visited twice—once on the way down

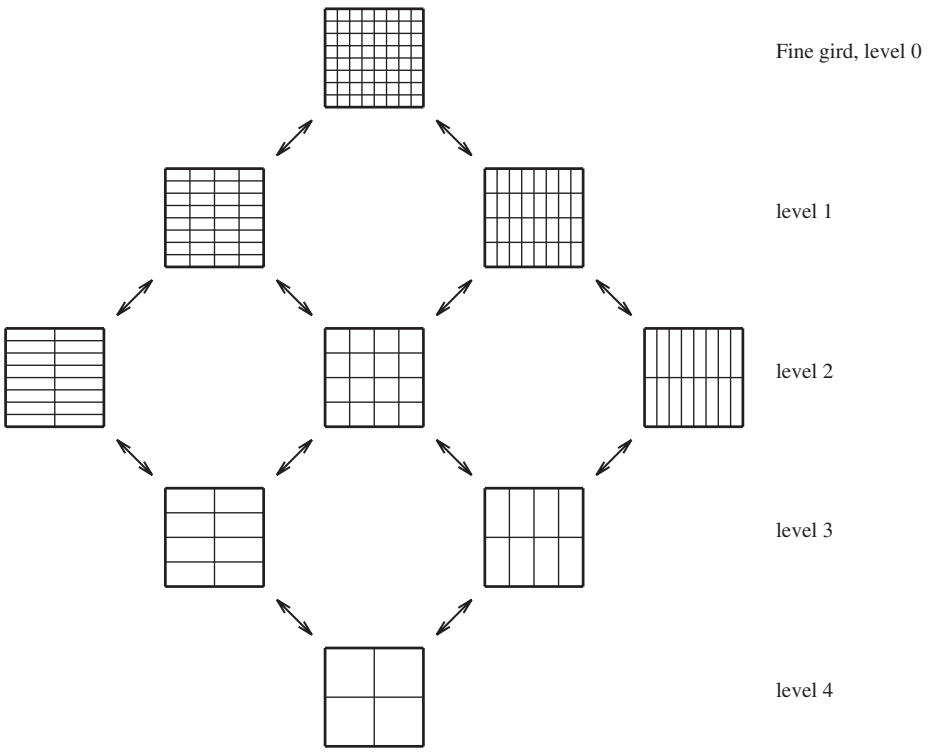

Figure 3. Schematic of the MSG coarse grid array for an 8×8 fine grid.

where smoothing is performed m times before restriction, and again on the way back up where smoothing is performed n times after prolongation. The cost of a 3D MSG V-cycle will then be approximately $8(m+n)$ workunits. For the typical values of $m=1$ and $n=3$, the computational cost becomes approximately 32 workunits per MSG V-cycle.

3.3. Some improvements

To those familiar with multigrid methods, these estimates of storage and work requirements will seem large and perhaps prohibitive, particularly when compared to the fractional overheads associated with standard multigrid methods (e.g. for isotropic factor-of-2 coarsening of a large 3D grid, the total storage scales as $1+1/8+1/64+\cdots \approx 1.14$ times the fine grid). One way to reduce these large storage and workunit-per-cycle (operator complexity) requirements is to increase the semi-coarsening factor. For an arbitrary semi-coarsening factor f, the total storage requirement M for 3D MSG multigrid is bounded as follows:

$$M \leqslant N \frac{f^3}{(f-1)^3} \tag{8}$$

where N is the size of the fine grid problem. For $f=2$, the result is $8N$ as mentioned previously. Increasing the semi-coarsening factor from 2 to 4 reduces the total storage requirement

to about 2.4 times the fine grid requirement. For V-cycle, the cycle complexity $C_{\text{V-cycle}}$ is bounded as follows:

$$C_{\text{V-cycle}} \leqslant (m+n)N\frac{f^3}{(f-1)^3} \tag{9}$$

where m and n are the number of smoothing sweeps before restriction and after prolongation, respectively. Using factor-of-4 coarsening, the cost is bounded by 9.6 workunits per cycle (assuming the same iteration counts m and n quoted above). The cycle complexities for the other two common cycle types are bounded as follows, although these were not tested in the present study:

$$C_{\text{F-cycle}} \leqslant (m+n)N\frac{f^3(f+2)}{(f-1)^4} \qquad \left.\frac{C_{\text{F-cycle}}}{C_{\text{V-cycle}}}\right|_{f=4} = 2 \tag{10}$$

$$C_{\text{W-cycle}} \leqslant (m+n)N\frac{f^3}{(f-2)^3} \qquad \left.\frac{C_{\text{W-cycle}}}{C_{\text{V-cycle}}}\right|_{f=4} \approx 3.4 \tag{11}$$

It remains to determine whether the reduction in workunits per cycle resulting from increasing semi-coarsening rates translates into a reduction in total solution time per time step, because the smoother will likely have a poorer convergence rate. By modelling the Helmholtz equation on 3D grids up to 32^3, Overman and Rosendale [15] showed that the reduction in computational cost per cycle for factor-of-4 semi-coarsening more than compensates for the reduced convergence rate. Our own results are in agreement with theirs. Figure 4 compares the convergence history of the present MSG multigrid method during the convergence of a single

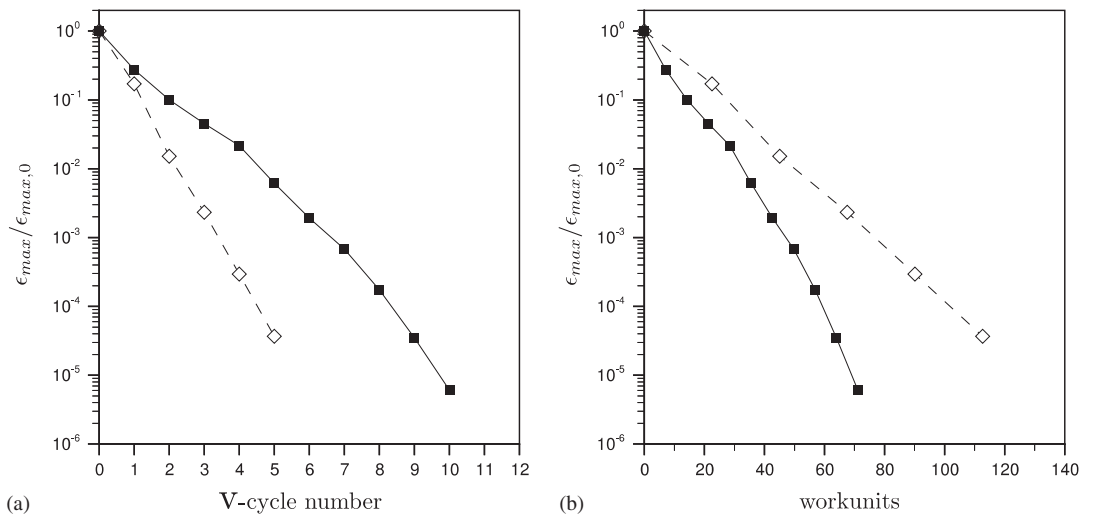

Figure 4. Convergence history for $64 \times 64 \times 64$ channel LES with MSG multigrid: ◇ factor-of-2 coarsening; ■ factor-of-4 coarsening.

time step for a $64 \times 64 \times 64$ LES of fully-developed turbulent channel flow. The two lines correspond to two different rates of semi-coarsening: factor-of-2 and factor-of-4. When plotted verses multigrid cycle number, the factor-of-2 semi-coarsening appears to give superior results. When plotted against workunits, however, the more costly V-cycle time associated with factor-of-2 semi-coarsening results in a convergence rate per work unit significantly worse than that for factor-of-4 semi-coarsening. Consequently, factor-of-4 coarsening was used exclusively in the present computations.

In addition to increasing the semi-coarsening factor, a second way to further reduce the storage and workunit-per-cycle for MSG multigrid is to skip certain grids when there is only weak coupling in that direction of coarsening. This idea was suggested by Mulder when he first proposed the MSG method [12]. For example, in a problem where the fine grid is highly compressed in the y-direction only, the lower branches of the grid hierarchy associated with x- and z-semi-coarsening can probably be eliminated without degrading the convergence rate per cycle. This technique was not implemented as part of the present work.

3.4. Restriction and prolongation operators

The key to making MSG multigrid robust is through the proper design of restriction and prolongation operators. The present implementation is developed from the ideas of Naik and Rosendale [16], where residuals are restricted isotropically to all semi-coarsened grids, but corrections are prolongated anisotropically using a locally calculated weighting that depends on the relative strengths of the equation coefficients in each coordinate direction.

3.4.1. Restriction. Let $\Omega^{l,m,n}$ represent a coarse grid somewhere in the grid hierarchy, and $b^{l,m,n}$ a source coefficient on that grid that must be determined by restriction. For 3D MSG multigrid, the typical coarse grid will be linked to three different fine grids, each being finer by the coarsening factor in one of the three coordinate directions. Using the superscript notation, we denote these grids as $\Omega^{l-1,m,n}$, $\Omega^{l,m-1,n}$, and $\Omega^{l,m,n-1}$, and their associated residuals $r^{l-1,m,n}$, $r^{l,m-1,n}$, and $r^{l,m,n-1}$, respectively. Defining the restriction operators R_x, R_y, and R_z corresponding to each direction of semi-coarsening, the coarse grid source coefficient is determined as follows:

$$b^{l,m,n} = w_{x,R} R_x r^{l-1,m,n} + w_{y,R} R_y r^{l,m-1,n} + w_{z,R} R_z r^{l,m,n-1} \tag{12}$$

where $w_{x,R}$, $w_{y,R}$, and $w_{z,R}$ are weights. Proper scaling of the coarse grid equations requires that $w_{x,R} + w_{y,R} + w_{z,R} = 1$. In the present implementation, we used isotropic restriction

$$w_{x,R} = w_{y,R} = w_{z,R} = \tfrac{1}{3} \tag{13}$$

In cases where the number of fine grids associated with a given coarse grid was only 1 or 2 (as can occur near the edges of the coarse grid array), the relevant weights were set to $w = 1$ or $w = \frac{1}{2}$, respectively.

The actual restriction operator used in the present work was insertion for cell-centred quantities such as pressure and temperature and for velocities not aligned with the semi-coarsening direction. This choice is straightforward to implement, and has a strong flux-based interpretation as described by Hutchinson and Raithby [17]. For the velocity components aligned with the semi-coarsening direction, 1-dimensional linear restriction operators were used with the weights calculated in the computational domain (i.e. assuming the grid spacing was constant).

3.4.2. Prolongation. Now let $\Omega^{l,m,n}$ represent a fine grid somewhere in the grid hierarchy, and $v^{l,m,n}$ a correction on that grid that must be determined by prolongation. The typical fine grid is linked to three different coarse grids, each semi-coarsened in one of the three coordinate directions. Using the superscript notation, we denote these grids as $\Omega^{l+1,m,n}$, $\Omega^{l,m+1,n}$, and $\Omega^{l,m,n+1}$, and their associated corrections $v^{l+1,m,n}$, $v^{l,m+1,n}$, and $v^{l,m,n+1}$, respectively. Defining the restriction operators P_x, P_y, and P_z corresponding to each direction of semi-coarsening, the fine grid correction is determined as follows:

$$v^{l,m,n} = w_{x,P}P_x v^{l+1,m,n} + w_{y,P}P_y v^{l,m+1,n} + w_{z,P}P_z v^{l,m,n-1} \tag{14}$$

where once again $w_{x,P}$, $w_{y,P}$, and $w_{z,P}$ are weights that must sum to unity for proper scaling of the coarse grid correction. In this case, however, the weights are allowed to vary throughout the domain. The idea is to set them locally to mirror the underlying anisotropy in the fine grid discrete operator(s). For the present implementation we use the 2D weightings suggested by [16] extended to 3D, which can be stated generally as

$$w_{x,P} = \frac{A_x^2}{A_x^2 + A_y^2 + A_z^2} \tag{15}$$

$$w_{y,P} = \frac{A_y^2}{A_x^2 + A_y^2 + A_z^2} \tag{16}$$

$$w_{z,P} = 1 - w_{x,P} - w_{y,P} = \frac{A_z^2}{A_x^2 + A_y^2 + A_z^2} \tag{17}$$

where A_x, A_y, and A_z are measures of the local coefficient strength in each of the three coordinate directions. In the present implementation, the continuity coefficients were found to give acceptable results for the prolongation of all corrections considered (pressure, velocity, and temperature)

$$A_x^2 = (A^{c,u_w})^2 + (A^{c,u_e})^2 \tag{18}$$

$$A_y^2 = (A^{c,v_s})^2 + (A^{c,v_n})^2 \tag{19}$$

$$A_z^2 = (A^{c,w_b})^2 + (A^{c,w_t})^2 \tag{20}$$

Similar to the case of restriction, the actual prolongation operators used were insertion (block correction [17]) for cell-centred quantities such as pressure and temperature, insertion for face-based velocities not aligned with the semi-coarsening direction, and linear interpolation of the corrections in the computational domain for aligned velocity components.

3.5. The SCGS smoother

The SCGS method of Vanka [4] was chosen as the smoother for the present MSG multigrid method. SCGS has the advantage of not requiring a special pressure or pressure correction equation. The coupled treatment of variables allows the continuity equation to remain in its standard conservation form.

One iteration of SCGS consists of cycling through all the cells of the domain in Gauss–Seidel fashion. As each cell is visited, a coupled set of 7 correction equations is assembled and solved to correct the cell-centred pressure and 6 surrounding face-centred velocities. In the course of a complete pass through the grid, each pressure will be corrected once, and each velocity corrected twice.

The set of 7 correction equations solved for cell $[i,j,k]$ are normally written in the following matrix form:

$$\begin{bmatrix} \frac{A^u_{i-1jk}}{\alpha} & 0 & 0 & 0 & 0 & 0 & A^{u,p_E}_{i-1jk} \\ 0 & \frac{A^u_{ijk}}{\alpha} & 0 & 0 & 0 & 0 & A^{u,p_P}_{ijk} \\ 0 & 0 & \frac{A^v_{ij-1k}}{\alpha} & 0 & 0 & 0 & A^{v,p_N}_{ij-1k} \\ 0 & 0 & 0 & \frac{A^v_{ijk}}{\alpha} & 0 & 0 & A^{v,p_P}_{ijk} \\ 0 & 0 & 0 & 0 & \frac{A^w_{ijk-1}}{\alpha} & 0 & A^{w,p_T}_{ijk-1} \\ 0 & 0 & 0 & 0 & 0 & \frac{A^w_{ijk}}{\alpha} & A^{w,p_P}_{ijk} \\ A^{c,u_w}_{ijk} & A^{c,u_e}_{ijk} & A^{c,v_s}_{ijk} & A^{c,v_n}_{ijk} & A^{c,w_b}_{ijk} & A^{c,w_t}_{ijk} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_{i-1jk} \\ \Delta u_{ijk} \\ \Delta v_{ij-1k} \\ \Delta v_{ijk} \\ \Delta w_{ijk-1} \\ \Delta w_{ijk} \\ \Delta p_{ijk} \end{bmatrix} = \begin{bmatrix} -R^u_{i-1jk} \\ -R^u_{ijk} \\ -R^v_{ij-1k} \\ -R^v_{ijk} \\ -R^w_{ijk-1} \\ -R^w_{ijk} \\ -R^c_{ijk} \end{bmatrix} \tag{21}$$

where the A's are the current coefficients from the linearized momentum and continuity equations (Section 2), α is a relaxation factor ($0 < \alpha \leqslant 1$), Δu, Δv, Δw and Δp are the corrections to the velocity and pressure values (or velocity and pressure correction values in the case of the first non-linear discretization), and the R's are the momentum and continuity residuals. As shown by Vanka, this system is easily solved for the corrections by Gaussian elimination.

In the present implementation, the residuals were calculated at the same time as the matrix was assembled in each cell, thus including the latest solution information in the true spirit of Gauss–Seidel methods. Since the residual calculations make up a significant part of the computational expense of the method, an alternative would be to calculate the residuals once at the start of each smoothing cycle. The benefit, if any, of this approach was not investigated as part of the present work.

In an effort to improve the smoother, some authors have included the off-diagonal neighbour coefficients in the first 6 rows of the matrix [18, 19]. Only minor improvements to the convergence rate were observed, and the system becomes significantly more expensive to solve. Our own tests came to the same conclusions. The best alternative appears to be to only include the diagonal coefficients and to relax the system slightly. In our experience, relaxation was best applied to the diagonal as shown in Equation (21). This differs, for example, from the relaxation technique used by Montero *et al.* for the same smoother [11]. With the present approach, the relaxation has the effect of increasing the diagonal dominance of the matrix, but still ensures that the resulting corrections exactly satisfy the continuity equation for the given cell. For the problems considered in this work, $\alpha = 0.75$ was found to yield consistent and monotonic rates of residual reduction.

4. RESULTS

Figure 5 compares typical convergence histories from a single time step for a variety of problems investigated as part of this work. The problems involved varying degrees of anisotropy in one, two, or all three directions, including

(1) homogeneous isotropic turbulence on uniform grids,
(2) flat-plate transition with stretching in the wall-normal direction,
(3) buoyancy-driven square cavity flow at $Ra = 10^8$ and 1.58×10^9 with 2 directions of wall-normal stretching,
(4) buoyancy-driven cubical cavity flow at $Ra = 10^8$ with 3 directions of wall-normal stretching.

A brief description of cases (2) and (3)—the two largest LES problems investigated here—including some sample results is given in the next subsections.

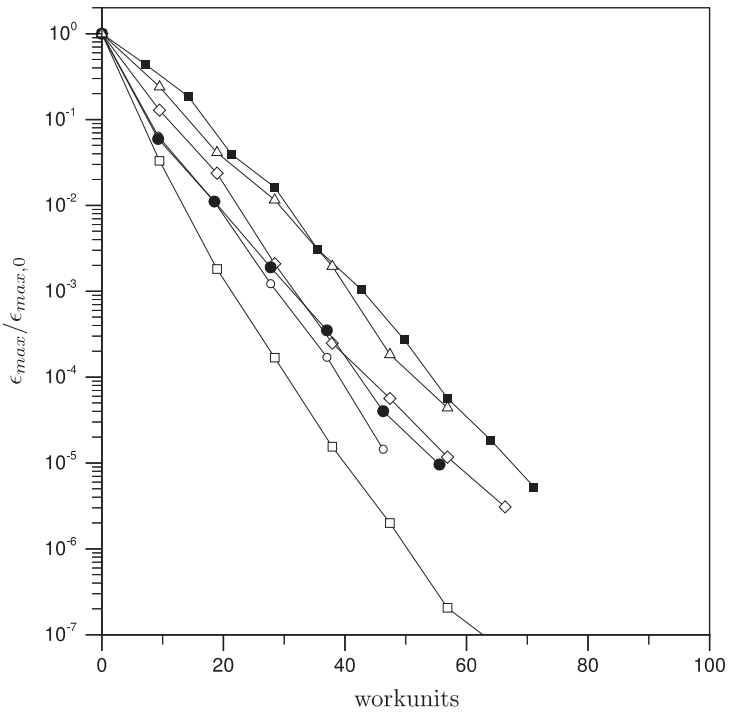

Figure 5. Typical convergence history per time step for a variety of different incompressible 3D Navier–Stokes problems using MSG multigrid with factor-of-4 coarsening: ■ $256 \times 64 \times 48$ flat plate transition; □ 64^3 homogeneous isotropic turbulence on a uniform grid; ◇ $96 \times 96 \times 64$ turbulent buoyancy driven cavity at $Ra = 1.58 \times 10^9$; ○ $64 \times 64 \times 32$ square cavity at $Ra = 10^8$ with maximum aspect ratio ≈ 35; • $64 \times 64 \times 32$ square cavity with maximum aspect ratio ≈ 100; △ $64 \times 64 \times 64$ cubical cavity $Ra = 10^8$.

4.1. Flat-plate transition

The MSG multigrid method was applied to the LES of wake-induced bypass transition on a flat plate to investigate the ability of fully conservative methods to properly capture the bypass transition process. This work was carried out as part of the Stanford Summer Program 2000, and is available in a more comprehensive form on the web at ctr.stanford.edu.

Figure 6 motivates the present LES, following the experiment of Liu and Rodi [20]. The computational domain used in the present simulation is also illustrated schematically.

The application of boundary conditions followed the procedure described for the DNS of Wu *et al.* [21]. At the domain inlet, a Dirichlet boundary condition was applied consisting of a laminar Blasius profile superposed with the periodically passing wake profile. The wake profile was interpolated from a filtered precomputation of a self-similar plane wake generated by Wu. The inlet velocities in the streamwise direction were scaled at the start of each time step to ensure a constant mass flow into the domain. At the outlet, the convective boundary condition was applied. The no-slip condition was imposed at the wall and along the freestream, the Blasius laminar velocity was imposed in the vertical direction, and a zero-vorticity condition applied to the other 2 components. The periodic boundary condition was applied in the spanwise direction.

Figure 7 shows some results from these calculations, where the fluctuating velocities in the wall normal direction are used at 5 equally spaced times to visualize the transition. The interaction between the passing wake and the laminar boundary layer appears as elongated puffs at $t/T = 0$, but breakdown to a turbulent spot does not occur until some time closer to $t/T = 0.4$. At $t/T = 0.4$, the isolated spot is clearly discernable with its characteristic arrowhead pointing upstream. As the turbulent spot is convected downstream, it grows and eventually merges with the fully turbulent portion of the boundary layer.

4.2. Buoyancy-driven cavity flow

To test the performance of the MSG multigrid method when the anisotropy varies throughout the domain in 2 or all 3 directions, the LES of the turbulent buoyancy-driven cavity was

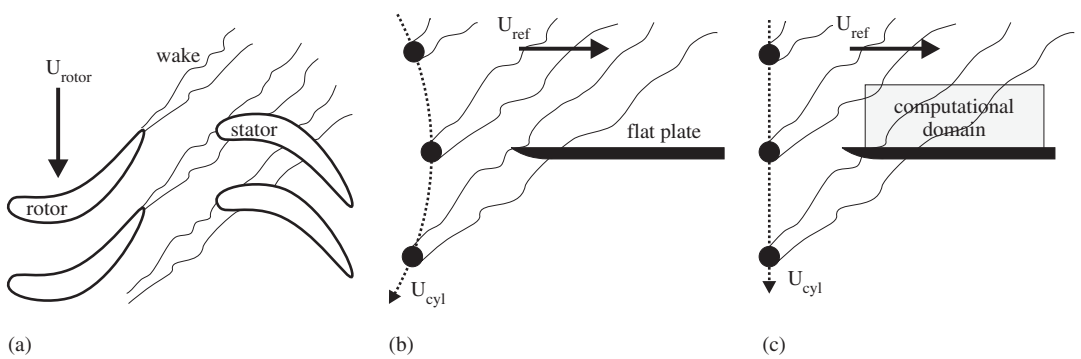

Figure 6. (a) Schematic of rotor-stator wake interaction; (b) layout in the experiments of Liu and Rodi [20]; and (c) layout in the DNS of Wu *et al.* [21], and the present LES.

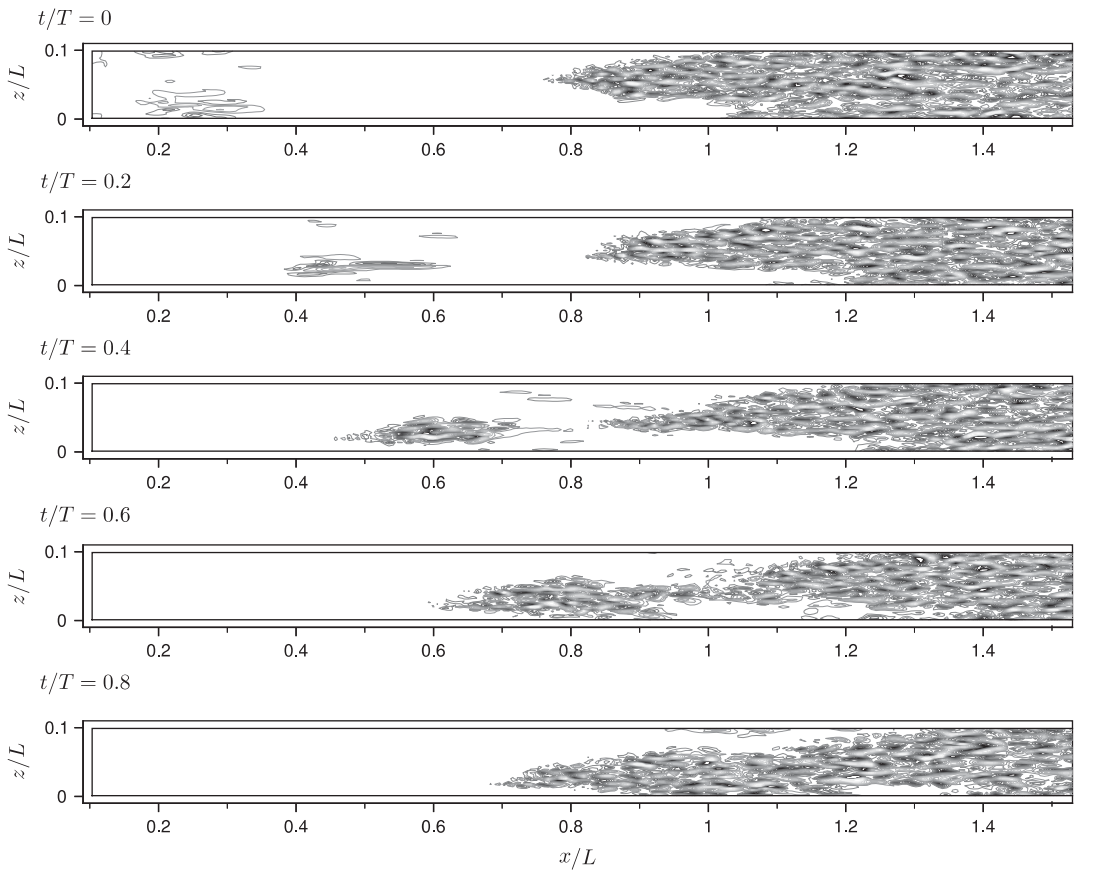

Figure 7. Visualization of turbulent spot formation and growth using v-component of fluctuating velocity in the x–z plane near the wall ($y/\delta_{99} = 0.4$ at $x/L = 0.8$). Contours represent $-0.1 < v/U_{\text{ref}} < 0.1$ in 0.01 increments.

performed. Using the Boussinesq assumption, these simulations required the solution of one additional transport equation for temperature. This was done in a fully-coupled fashion, and its solution was added to the coupled SCGS smoother as described by Wesseling [5]. In this subsection we present results from the turbulent square cavity flow—Figure 8(b). The domain size was $L \times L \times L/2$ with the hot wall at $x = 0$, the cold wall at $x = L$, and the sidewalls at $y = 0$ and $y = L$. A grid of $96 \times 96 \times 64$ was used, with grid stretching towards the boundaries in both the x- and y-directions using the distribution of the following equation:

$$x(i) = \frac{1}{2}\left(\frac{\tanh(\alpha(2i/N_i - 1))}{\tanh(\alpha)} + 1\right) \quad i = 0 \ldots N_i \tag{22}$$

where $\alpha = 2.8$. This resulted in a maximum aspect ratio of about 64, and about 10 grid points inside the peak in mean v-velocity along the vertical hot and cold walls. This is similar to the distribution used in the recent simulations of the same problem by Peng and Davidson [22], and other authors have found this degree of stretching to be approximately optimal [23].

Boundary conditions were $u=v=w=0$ for all walls ($x=0,L,\ y=0,L$). The temperature $T=T_{\mathrm{h}}=1$ was applied to the hot wall at $x=0$, and $T=T_{\mathrm{c}}=0$ to the cold wall at $x=L$. Along the horizontal side walls, a Dirichlet boundary condition was applied to match the measured profiles reported in the recent experimental work of Tian and Karianis [24, 25]. A periodic boundary condition was applied in the z-direction.

Figure 9 compares two three-dimensional views of the square cavity simulation using the instantaneous temperature isosurface $T=(T_{\mathrm{h}}+T_{\mathrm{c}})/2$. The simulation on the right (Figure 9(b))

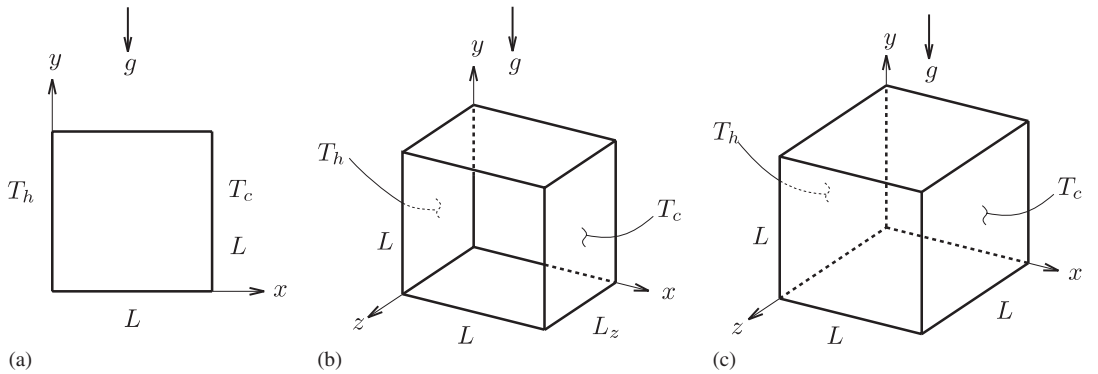

Figure 8. Geometry and boundary conditions for three configurations of the buoyancy-driven cavity flow problem: (a) 2D square cavity; (b) 3D square cavity with spanwise periodic boundary conditions; and (c) 3D cubical cavity.

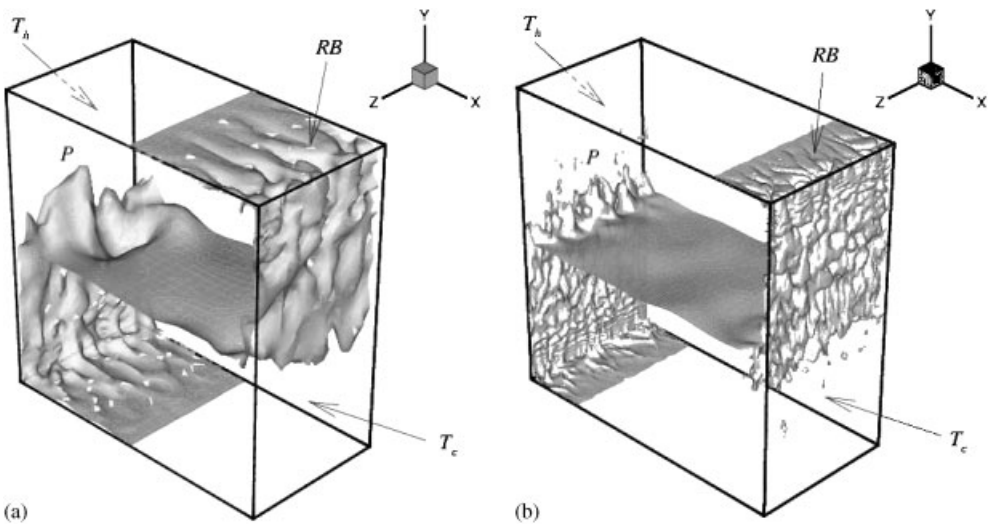

Figure 9. Use of $T=(T_{\mathrm{h}}+T_{\mathrm{c}})/2$ isosurface to illustrate flow structure at two different Rayleigh numbers. $P=$Plume, $RB=$Rayleigh–Bénard: (a) $Ra=10^8$, $Pr=0.71$; and (b) $Ra=1.58\times10^9$, $Pr=0.71$.

is at $Ra = 1.58 \times 10^9$. The simulation on the left is the same flow at the lower Rayleigh number $Ra = 10^8$. The difference in the scale of flow structures at the two Rayleigh numbers is immediately apparent. In the centre of the flow, the isosurface is seen to be flat and quite smooth, indicative of the quiescent, stable stratification of the core. On the far side near the hot wall the isosurface is lifted and deformed in the rising turbulent boundary layer, where thermal plumes might be expected (marked P). One of the most interesting flow features evident with this isosurface visualization technique is the Rayleigh–Bénard instability along the top sidewall. As the relatively hot fluid of the top horizontal boundary layer approaches the cold wall, the near wall portion of the boundary layer is cooled by the sidewall, producing an unstable thermal stratification. As this instability breaks down and the cold fluid falls away from the wall, it pulls the isosurface with it, leaving fissure-like tracks (marked RB). The spacing of these tracks and thus the spacing of the breakdown in unstable stratification appears to be regular and Rayleigh-number related.

5. PARALLEL IMPLEMENTATION

An attractive feature of multigrid solvers is their relatively straightforward parallelization by the method of domain decomposition. While an optimal parallel implementation of the present MSG multigrid method was not the focus of this research, this section briefly describes the approach taken and gives some speedup results. For more detailed investigations of parallel implementations of MSG multigrid, the reader is directed to [26, 27].

All computations were carried out on a parallel PC cluster of up to 40 nodes[‖] connected by fast ethernet [28]. The LES methodology described in this paper, including the MSG multigrid solver, was implemented in parallel for this computational resource using Cartesian domain decomposition, where each processor iterates on a Cartesian sub-domain of the global domain and exchanges halo data with neighbouring processes on a regular basis. MPI was used for inter-processor communication [29]. In the present implementation, semi-coarsening of the local sub-domain was performed up to a coarseness of two control volumes on the local Cartesian sub-domain. Further coarsening would have required the redistribution of the coarsest grids to a smaller number of processors, so in the present implementation there will be a number of coarse grids omitted from the grid hierarchy that would have been present in an identical single-processor calculation. Because the grid aspect ratios are the principle source of anisotropy in the present simulations, any adverse effect of this omission of coarse grids can be minimized by choosing a global partitioning that is geometrically isotropic.

A second difference between a single-processor simulation and a multi-processor simulation is in the smoother. For ease of implementation, the Gauss–Seidel smoother was implemented on a processor basis. Although a partition-independent red–black smoother would be possible for the structured Cartesian domains involved, this was not investigated. For a more thorough discussion of these parallelization issues, the reader is directed to References [30, 31].

Figure 10 presents some typical results for the parallel speedup and parallel efficiency for the solution of the buoyancy-driven cubical cavity problem at $Ra = 4 \times 10^4$ with experiment conducted by Leong *et al.* [32] (Figure 8(c)). The parallel efficiency is seen to improve

[‖]Computers were single processor PIII 450 and 533 MHz machines with 256 Mb RAM each.

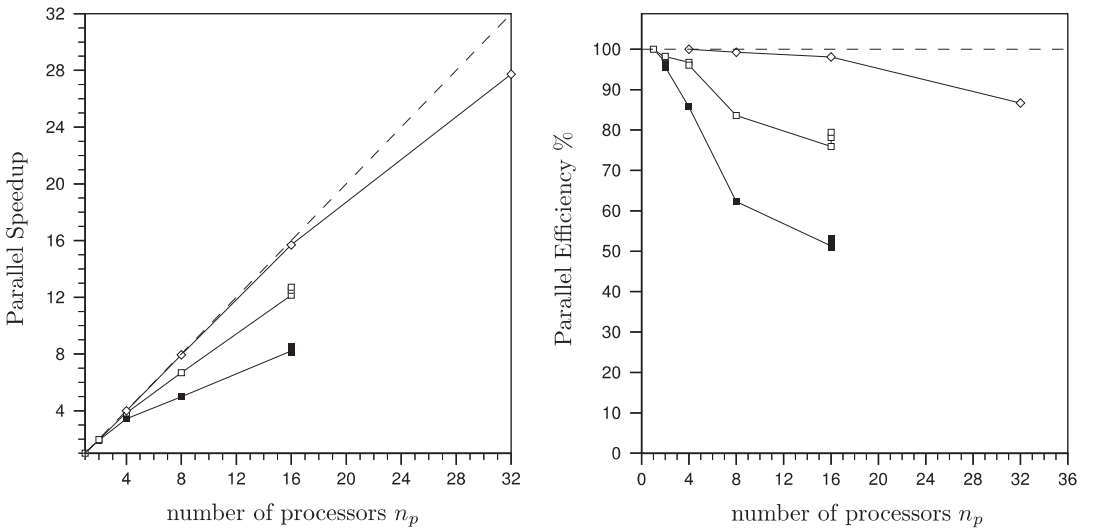

Figure 10. Parallel speedup and efficiency for buoyancy-driven cubical cavity problem for different grid sizes: ■ 32^3 □ 64^3; ◇ 128^3; – – – – ideal.

substantially for larger problem sizes, as is well known for parallel implementations using the method of domain decomposition [33].

The largest problem investigated as part of the present research was the $256 \times 64 \times 48 = 786\,432$ LES of wake-induced bypass transition (Section 6.1). This calculation solved the fully-conservative discretization of the Navier–Stokes equations using 32 nodes of the cluster, and required 3 days to compute 5500 time steps (approximately 47 s per time step). The next largest problem was the $96 \times 96 \times 64 = 589\,824$ LES of the turbulent square cavity at $Ra = 1.58 \times 10^9$ (Section 6.2). This problem solved the coupled Navier–Stokes and energy equations using 18 nodes of the cluster, and required 3.5 days to compute 5000 time steps (approximately 60 s per time step).

Finally we note that the grids used throughout the present work were all evenly divisible by integer powers of 2. The method is not limited to the use of such grids in general. This restriction does, however, considerably simplify the implementation of the prolongation and restriction routines required by the MSG multigrid solver.

6. SUMMARY

The MSG multigrid method combined with the SCGS smoother of Vanka has been investigated as a global solver for fully-implicit discretizations of the time-dependent incompressible Navier–Stokes equations. The coarse grids required by MSG multigrid are developed using factor-of-4 semi-coarsening, resulting in a total memory requirement of approximately 2.4 times the fine grid alone, and a computation cost per V-cycle of about 9.5 workunits. When applied to a number of different large eddy simulations involving varying degrees of coefficient anisotropy, the method was consistently able to reduce the residual by 5 orders of

magnitude in 50–80 workunits (i.e. approximately 5–8 V-cycles). A parallel implementation of the method was shown to scale reasonably on a parallel cluster up to 32 processes. As part of ongoing work, the applicability of the method to FAS multigrid will be investigated, where the coarse-grid operators are generated by discretization of the equations on the coarse grids.

ACKNOWLEDGEMENTS

The authors are grateful to the Natural Sciences and Engineering Research Council of Canada (NSERC) for the financial support of this work.

REFERENCES

1. Ham FE, Lien FS, Strong AB. A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids. *Journal of Computational Physics* 2002; **177**:117–133.
2. Pierce CD. *Ph.D. Thesis*, Department of Mechanical Engineering, Stanford University, 2002.
3. Wall C, Pierce CD, Moin P. A semi-implicit method for resolution of acoustic waves in low Mach number flows. *Journal of Computational Physics* 2002; **181**(2):545–563.
4. Vanka SP. A calculation procedure for three-dimensional steady recirculating flows using multigrid methods. *Computer Methods in Applied Mechanics and Engineering* 1986; **55**:321–338.
5. Wesseling P. *An Introduction to Multigrid Methods*. Wiley: New York, 1991.
6. Trottenberg U, Oosterlee CW, Schueller A. *Multigrid*. Academic Press: London, 2000.
7. Wesseling P, Oosterlee CW. Geometric multigrid with application to computational fluid dynamics. *Journal of Computational and Applied Mathematics* 2001; **128**(2):311–334.
8. McBryan OA, Frederickson PO, Linden J, Schuller A, Solchenbach K, Stuben K, Thole C. Multigrid methods on parallel computers—a survey of recent developments. *Impact Computer Science and Engineering* 1991; **3**:1–75.
9. Schaffer S. A semicoarsening multigrid method for elliptic partial differential equations with highly discontinuous and anisotropic coefficients. *SIAM Journal on Scientific Computing* 1998; **20**(1):228–242.
10. Falgout RD, Yang UM. Hypre: a library of high performance preconditioners. In *Computational Science—ICCS 2002*, *Part III*, Sloot PMA, Tan CJK, Dongarra JJ, Hoekstra AG (eds). Lecture Notes in Computer Science, vol. 2331. Springer: Berlin, 2002; 632–641. Also available as Lawrence Livermore National Laboratory, *Technical Report UCRL-JC-146175*.
11. Montero RS, Llorente IM, Salas MD. Robust multigrid algorithms for the Navier–Stokes equations. *Journal of Computational Physics* 2001; **173**:412–432.
12. Mulder WA. A new multigrid approach to convection problems. *Journal of Computational Physics* 1989; **83**:303–323.
13. Germano M, Piomelli U, Moin P, Cabot WH. A dynamic subgrid-scale eddy viscosity model. *Physics of Fluids A* 1991; **3**(7):1760–1765.
14. Lilly DK. A proposed modification of the Germano subgrid scale closure method. *Physics of Fluids A* 1992; **4**:633–635.
15. Overman A, Van Rosendale J. Mapping robust parallel multigrid algorithms to scalable memory architectures. *Proceedings of the 6th Copper Mountain Conference on Multigrid Methods*, 1993.
16. Naik NH, Van Rosendale J. The improved robustness of multigrid elliptic solvers based on multiple semicoarsened grids. *SIAM Journal on Numerical Analysis* 1993; **30**(1):215–229.
17. Hutchinson BR, Raithby GD. A multigrid method based on the additive correction strategy. *Numerical Heat Transfer* 1986; **9**:511–537.
18. Thompson MC, Ferziger JH. An adaptive multigrid technique for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **82**:94–121.
19. Wittum G. The use of fast solvers in computational fluid dynamics. In *Notes on Numerical Fluid Dynamics*, *Proceedings of the 8th GAMM Conference on Numerical Methods in Fluid Mechanics*, Wesseling P (ed.). 1990; 574–581.
20. Liu X, Rodi W. Experiments on transitional boundary layers with wake-induced unsteadiness. *Journal of Fluid Mechanics* 1991; **231**:229–256.
21. Wu X, Jacobs RG, Hunt JCR, Durbin PA. Simulation of boundary layer transition induced by periodically passing wakes. *Journal of Fluid Mechanics* 1999; **398**:109–154.
22. Peng S-H, Davidson L. Large eddy simulation for turbulent buoyant flow in a confined cavity. *International Journal of Heat and Fluid Flow* 2001; **22**:323–331.

23. Manole DM, Lage JL. Nonuniform grid accuracy test applied to the natural-convection flow within a porous medium cavity. *Numerical Heat Transfer*, *Part B* 1993; **23**:351–368.
24. Tian YS, Karayiannis TG. Low turbulence natural convection in an air filled square cavity. Part I: the thermal and fluid flow fields. *International Journal of Heat and Mass Transfer* 2000; **43**:849–866.
25. Tian YS, Karayiannis TG. Low turbulence natural convection in an air filled square cavity. Part II: the turbulence quantities. *International Journal of Heat and Mass Transfer* 2000; **43**:867–884.
26. Oosterlee CW. The convergence of parallel multiblock multigrid methods. *Applied Numerical Mathematics* 1995; **19**:115–128.
27. Washio T, Oosterlee CW. Flexible multiple semicoarsening for three dimensional singularly perturbed problem. *SIAM Journal on Scientific Computing* 1998; **19**:1646–1666.
28. Becker D, Merkey P. *The Beowulf Project*. World Wide Web, http://www.beowulf.org
29. Argonne National Laboratory. *MPICH—A Portable Implementation of MPI*. World Wide Web, http://www-unix.mcs.anl.gov/mpi/mpich
30. Lien FS. High performance computing of turbulent flows. *ERCOFTAC Course on High Performance Computing in Fluid Dynamics*, Delft, The Netherlands. Kluwer Academic Publishers: Dordrecht, 1996; 201–236.
31. Douglas CC. Multigrid methods in science and engineering. *IEEE Computational Science and Engineering* 1996; 55–68.
32. Leong WH, Hollands KGT, Brunger AP. On a physically-realizable benchmark problem in internal natural convection. *International Journal of Heat and Mass Transfer* 1998; **41**:3817–3828.
33. Brown PN, Falgout RD, Jones JE. Semicoarsening multigrid on distributed memory machines. *SIAM Journal on Scientific Computing* 2000; **21**(5):1823–1834.

