INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1819–1843
Published online 2 October 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1560

Large eddy simulations of incompressible turbulent flows using parallel computing techniques

A. Gokarn[1,*,†], F. Battaglia[1], R. O. Fox[2], J. C. Hill[2] and J. Reveillon[3]

[1]*Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, U.S.A.*
[2]*Department of Chemical and Biological Engineering, Iowa State University, Ames, IA 50011, U.S.A.*
[3]*University of Rouen-CORIA, INSA, Avenue de l'Universite, 76801 Saint Etienne du Rouvray, France*

SUMMARY

This paper presents a detailed procedure to solve incompressible high Reynolds number turbulent flows using large eddy simulations (LES) on distributed memory machines. The filtered Navier–Stokes equations are discretized using a partial-staggered variable arrangement and solved using a finite difference grid. A second-order central difference scheme and sixth-order compact scheme are employed for the spatial derivatives. A third-order low storage Runge–Kutta method is used for the temporal derivatives. Validation of the numerical scheme is performed first by simulating a driven cavity flow and flow over a backward-facing step. The dynamic Smagorinsky subgrid turbulence model is then validated for flow in a channel. Simulations are validated with relevant data available in literature. Since LES is computationally expensive, the solver is parallelized using message passing interface. An efficient parallel linear equation solver is utilized for solving the elliptical pressure Poisson equation. The parallel program is tested for solutions of flow in a complex flow configuration and preliminary results are compared with experimental data. Performance of the program for the same geometry is tested on a parallel cluster up to 256 processors. The novel approach in this work is the use of a partial-staggered variable arrangement for LES of turbulent flows, obviating the need for any form of artificial dissipation that might mask the subgrid effect on the solution. Copyright © 2007 John Wiley & Sons, Ltd.

Received 23 January 2007; Accepted 31 March 2007

KEY WORDS: finite difference; partial staggered; large eddy simulation; message passing interface

*Correspondence to: A. Gokarn, Department of Mechanical Engineering, Iowa State University, Ames, IA 50011, U.S.A.
†E-mail: anup.gokarn@gmail.com

Contract/grant sponsor: National Science Foundation; contract/grant number: CTS-0336435

1. INTRODUCTION

Numerical solutions to the incompressible Navier–Stokes equations have been documented since the early 20th century [1]. Mathematically, the incompressible flow governing equations present unique issues in the form of satisfying the incompressibility requirement. In an incompressible medium, pressure information travels at infinite speed, which is the root cause of all problems. In literature, two distinct approaches for satisfying this incompressibility constraint have been documented, and the solution differences using various numerical methods stem from the choice of the method. In 'pressure-based' methods, the incompressibility constraint is satisfied directly by decoupling pressure from the momentum equations and then solving the pressure Poisson equation [2, 3]. Occasionally, instead of pressure and velocity, variables such as the vorticity-stream function have been used. However, the use of the vorticity-stream function approach has been limited due to increased computational costs for three-dimensional calculations. The second method is referred to as a 'density-based' method and is coupled with the artificial compressibility formulation [4, 5]. The methodology closely mimics the compressible flow formulation, where momentum and continuity equations are coupled through density, and incompressibility is recovered in a limiting sense. In the pressure-based method, most of the computational time is spent in solving the elliptic pressure Poisson equation where a sparse matrix has to be inverted every time step. In the density-based method, a time derivative of the pressure term is added to the continuity equation. The governing equations are iterated in 'pseudo-time' until convergence is obtained, which is done for every time step. Efficient algorithms (e.g. preconditioning techniques) are often employed to obtain fast convergence. Both methods have been used by researchers and there is no evidence of the superiority of one method over another. Over the years, the computational fluid dynamics (CFD) community has seen improvements in numerical algorithms, in terms of both speed and accuracy, and the ability to obtain solutions for complex geometries and systems, all due to tremendous improvements in computing power. This is especially important for turbulent flow calculations which encompass a wide range of length and time scales. High grid resolution and small time steps lead to long calculation times and what takes weeks on a single computer can now be completed in a matter of hours on a supercomputer.

The aim of the present work is to outline a methodology for numerical simulations of complex turbulent flow problems. The novel approach in this work is the use of a partial-staggered variable arrangement for large eddy simulations (LES) of turbulent flows, obviating the need for any form of artificial dissipation (for stability) that might mask the subgrid effect on the solution. A pressure-based scheme is used for solving the governing equations. The application of an *in situ* turbulence generating methodology at the inflow boundary for LES is shown to work very well, thus preventing the need for the otherwise expensive numerical simulations required to provide suitable inflow conditions. The tools used in the description of the solution methodology have been addressed in a consistent and logical fashion so that readers attempting to solve similar problems can easily understand and adopt the methods used to suit their purpose. In the following sections, the governing equations, numerical scheme and solution methodology are described in detail. A set of validation studies are presented utilizing standard benchmark cases such as driven cavity flow, flow over a backward-facing step and turbulent channel flow. Parallel computing using message passing interface (MPI) is discussed and the parallel program is tested for simulations (using about 2×10^6 cells) of a confined jet flow for which particle image velocimetry (PIV) measurements are available. Performance of the parallel program is studied on an AMD Opteron cluster using up to 256 processors.

DOI: 10.1002/fld

2. GOVERNING EQUATIONS

The motion of a constant density fluid is governed by the incompressible Navier–Stokes equations. Direct numerical simulations (DNS) is the only means of solving these equations exactly. However, for complex flows, the present day computing facilities are not enough to obtain solutions in a reasonable time. LES is a methodology where solutions are obtained on a coarse grid (compared to DNS). Thus, the underlying philosophy of LES is to compute only the large-scale motions of the flow and model the effect of the small-scale motions on the evolution of the large scales. In order to separate the large scales from the small scales, a filtering operation is performed. Filtering is represented mathematically in physical space as a convolution product. The resolved (filtered) part denoted by $\overline{\phi}(x_i)$ is defined by the relation,

$$\overline{\phi}(x_i) = \int_D \phi(\xi_i) G(x_i - \xi_i; \overline{\Delta})\, \mathrm{d}\xi_i$$

where ϕ is a generic variable and convolution kernel G is a filter function associated with the cut-off scales in space $\overline{\Delta}$. The space vectors are represented by x_i and ξ_i, and D represents the entire domain. The non-dimensional filtered set of equations are given by,

$$\frac{\partial \overline{u}_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{\partial \overline{u}_i \overline{u}_j}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\left(\frac{\partial \overline{u}_i}{\partial x_j}\right) - \frac{\partial \tau_{ij}}{\partial x_j} \tag{2}$$

where the turbulent subgrid stress (SGS) tensor that requires modeling is

$$\tau_{ij} = \overline{u_i u_j} - \overline{u}_i \overline{u}_j \tag{3}$$

and t represents time, $\overline{u}_i$ is the filtered velocity field, $\overline{p}$ is the pressure and Re is the Reynolds number.

3. NUMERICAL SOLUTION: FINITE DIFFERENCE METHODS

3.1. Variable arrangement

A key issue during discretization is to select points in the domain at which the values of the unknown dependent primitive variables are to be computed. In literature, three such arrangements are found, and will be discussed next. For each arrangement, the grid points are displaced by δx, δy and δz in the x, y and z directions, respectively.

3.1.1. Non-staggered arrangement. The non-staggered grid arrangement [4] defines both pressure and velocities at the grid nodes (Figure 1). The non-staggered arrangement has advantages because of its simplicity and the fact that velocity is defined on the boundary where it is generally prescribed. This arrangement has significant advantages in complex geometries and situations where unstructured grids are often used. However, the main disadvantage of a non-staggered grid arrangement is that pressure is defined on the domain boundary. The resulting discretized system

DOI: 10.1002/fld

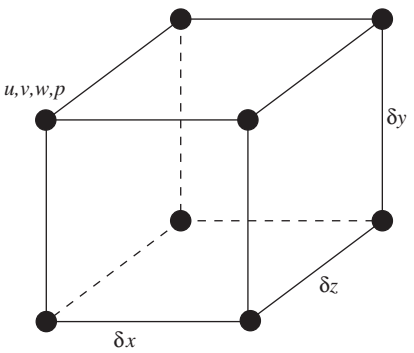

Figure 1. Non-staggered arrangement for a grid cell where velocities and pressure are defined at each node. δx, δy and δz are the grid point displacements along x, y and z directions, respectively.

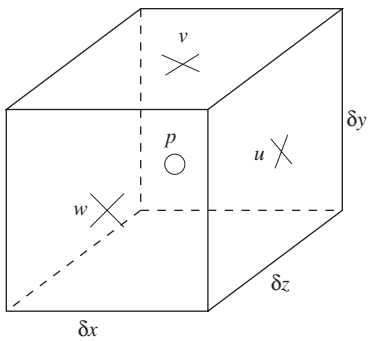

Figure 2. Complete staggered arrangement where velocities are defined at the cell face centers and pressure is defined at the cell volume center.

of equations for a node has no direct coupling between the pressure and velocities, resulting in what is known as odd–even splitting. Non-staggered central difference schemes are known to suffer because of pressure–velocity decoupling, which make the schemes unstable especially for high Reynolds number turbulent flows [6]. To prevent spurious pressure fluctuations, a popular remedy is to introduce 'artificial diffusion' which provides a stable solution [7]. This method requires the specification of *ad hoc* constants that determine the amount of extra diffusion introduced.

3.1.2. Complete staggered arrangement. The staggered grid arrangement (Figure 2) introduced by Harlow and Welch [3] offers advantages over the non-staggered arrangement. For a staggered grid, velocities are defined at the cell faces, while pressure is defined at the cell center. The biggest advantage of the staggered arrangement is the strong coupling between the velocities and pressure, which alleviates convergence problems and oscillations in pressure and velocity fields [6]. On the other hand, a disadvantage of this arrangement is that only one of the velocity components is defined on each side of the domain boundary. Hence, it is necessary to employ ghost cells outside the domain to enforce boundary conditions.

DOI: 10.1002/fld

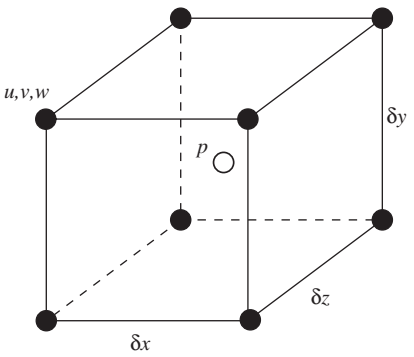

Figure 3. Partial-staggered arrangement where velocities are defined at the grid nodes and pressure is defined at the cell center.

3.1.3. Partial-staggered arrangement. A partial-staggered arrangement [8] defines velocities at the grid nodes (Figure 3) similar to the non-staggered arrangement while pressure is staggered to the cell center. The arrangement has the advantage that pressure is staggered with respect to velocity and the pressure–velocity decoupling is avoided (as in the completely staggered case). Moreover, the domain passes through all the velocity points and hence no ghost cells are required outside the domain.

3.1.4. Selection of grid variable arrangement. Numerical simulations were performed using the non-staggered and the partial-staggered arrangements to compare the stability of the two configurations. The divergence of the velocity field ($\nabla \cdot \mathbf{u}$) gives an indication if the incompressibility constraint is satisfied by a particular numerical scheme and this criterion was used to compare the two arrangements. Two-dimensional Poiseuille flow (200×48 cells) and three-dimensional duct flow simulations ($200 \times 48 \times 48$ cells), for which analytical solutions exist, were performed. The order of $\mathbf{\nabla \cdot u}$ on the non-staggered grid was 10^{-4}, whereas on the partial-staggered grid a value of 10^{-5} was obtained. Thus for a simple laminar flow, an order difference in magnitude was observed for the divergence of the velocity field. Moreover, the simulation on the non-staggered grid led to an instability in the solution due to the decoupled nature of the variable arrangement. The cause of decoupling is illustrated next [7].

On the non-staggered grid, the pressure gradient using a second-order central difference formulation has the form,

$$\frac{\partial \overline{p}_m^n}{\partial x} = \frac{\overline{p}_{m+1}^n - \overline{p}_{m-1}^n}{2\delta x}$$

where m represents the grid index and n is the time level. The Laplacian operator can be written as,

$$\frac{\partial}{\partial x}\frac{\partial \overline{p}_m^n}{\partial x} = \frac{\partial}{\partial x}\left(\frac{\overline{p}_{m+1}^n - \overline{p}_{m-1}^n}{2\delta x}\right)$$

$$= \frac{\partial \overline{p}_{m+1}^n/\partial x - \partial \overline{p}_{m-1}^n/\partial x}{2\delta x}$$

DOI: 10.1002/fld

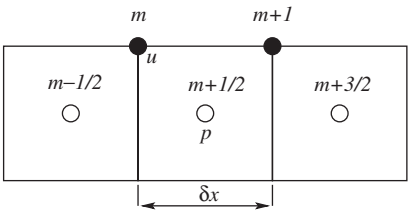

Figure 4. Index notation for the variables on a partial-staggered grid.

$$= \frac{(\overline{p}^n_{m+2} - \overline{p}^n_m)/2\delta x - (\overline{p}^n_m - \overline{p}^n_{m-2})/2\delta x}{2\delta x}$$

$$= \frac{\overline{p}^n_{m+2} - 2\overline{p}^n_m + \overline{p}^n_{m-2}}{(2\delta x)^2} \tag{4}$$

However, directly discretizing the Laplacian of pressure using a second-order central difference scheme has the form,

$$\frac{\partial^2 \overline{p}^n_m}{\partial x^2} = \frac{\overline{p}^n_{m+1} - 2\overline{p}^n_m + \overline{p}^n_{m-1}}{(\delta x)^2} \tag{5}$$

which is not consistent with Equation (4). If Equation (4) is used to represent the Laplacian of pressure, the formulation leads to pressure–velocity decoupling. The splitting is caused by the fact that the numerator of Equation (4) requires discrete pressure values at locations relative to $\overline{p}_m$ separated by $2\delta x$, causing the pressure solution to oscillate. In contrast, Equation (5) only uses adjacent pressure values at δx. To prevent odd–even splitting, some form of dissipation is added, e.g. [7],

$$\frac{\partial}{\partial x}\frac{\partial \overline{p}^n_m}{\partial x} = \frac{\overline{p}^n_{m+2} - 2\overline{p}^n_m + \overline{p}^n_{m-2}}{(2\delta x)^2} - \frac{\varepsilon}{4}(\delta x)^2 \frac{\partial^4 \overline{p}^n_m}{\partial x^4}$$

where ε is a constant whose value lies between 0 and 1. A value of 0 gives the discretized Equation (4), whereas a value of 1 gives back the form shown in Equation (5). The present study is intended for LES; therefore, artificially introducing numerical dissipation must be avoided. Thus, a partial-staggered grid is adopted in this study, and a detailed description of the arrangement is presented next.

A two-dimensional schematic is shown in Figure 4 representing index notation on a partial-staggered grid. The velocities are represented by filled circles and pressure by hollow circles. The subscripts for pressure indicate that they are displaced $\frac{1}{2}\delta x$ from the velocity nodes. Unlike the non-staggered approach, the divergence and gradient operators on u and p, respectively, for the partial-staggered scheme are defined differently. The divergence operator is defined at the cell center $(m + 1/2)$, while the gradient operator is defined at the cell face (m). The discrete forms of the divergence of velocity and the pressure gradient are shown for time level n,

$$\left.\frac{\partial \overline{u}^n}{\partial x}\right|_{m+1/2} = \frac{\overline{u}^n_{m+1} - \overline{u}^n_m}{2(\delta x/2)} \tag{6}$$

DOI: 10.1002/fld

$$\left.\frac{\partial \overline{p}^n}{\partial x}\right|_m = \frac{\overline{p}^n_{m+1/2} - \overline{p}^n_{m-1/2}}{2(\delta x/2)} \tag{7}$$

The Laplacian of pressure at $m + 1/2$ can then be expanded as follows:

$$\begin{aligned}\left.\frac{\partial}{\partial x}\frac{\partial \overline{p}^n}{\partial x}\right|_{m+1/2} &= \frac{\partial \overline{p}^n/\partial x|_{m+1} - \partial \overline{p}^n/\partial x|_m}{(2\delta x/2)} \\ &= \frac{(\overline{p}^n_{m+3/2} - \overline{p}^n_{m+1/2})/(2\delta x/2) - (\overline{p}^n_{m+1/2} - \overline{p}^n_{m-1/2})/(2\delta x/2)}{(2\delta x/2)} \\ &= \frac{\overline{p}^n_{m+3/2} - 2\overline{p}^n_{m+1/2} + \overline{p}^n_{m-1/2}}{(\delta x)^2}\end{aligned} \tag{8}$$

This discretization of the Laplacian of pressure (Equation (8)) is consistent with that of the discrete pressure gradient (Equation (7)) and divergence operator (Equation (6)) and thus eliminates the odd–even splitting of pressure solution. On the partial-staggered grid, the pressure gradients obtained on the cell faces have to be interpolated back on to the grid points, leading to an added computational effort. First-order interpolations are used for the purpose.

3.2. Spatial and temporal discretizations

LES models are dissipative in nature, hence, it is important to use non-dissipative discretization schemes. For the spatial discretizations shown in this section, only one direction will be shown for simplicity (subscript represents grid index m).

3.2.1. Second-order central difference scheme. The first- and second-order derivatives for a general function $f(x)$ are, respectively,

$$\frac{\partial f_m}{\partial x} = f'_m = \frac{f_{m+1} - f_{m-1}}{2\delta x}$$

$$\frac{\partial^2 f_m}{\partial x^2} = f''_m = \frac{f_{m+1} - 2f_m + f_{m-1}}{(\delta x)^2}$$

At the boundaries, one-sided approximations are used with second-order accuracy. For example at a left boundary,

$$f'_m = \frac{-3f_m + 4f_{m+1} - f_{m+2}}{2\delta x}$$

$$f''_m = \frac{2f_m - 5f_{m+1} + 4f_{m+2} - f_{m+3}}{(\delta x)^2}$$

3.2.2. Sixth-order compact scheme. A higher-order scheme can resolve a larger range of wave numbers as compared to a second-order scheme on the same grid. In other words, a higher-order

DOI: 10.1002/fld

scheme requires fewer grid points to resolve the same scales. A sixth-order low dispersive scheme proposed by Lele [9] is used here. The approximation for a first-order derivative is

$$f'_m + \alpha(f'_{m+1} + f'_{m-1}) = \frac{a}{2\delta x}(f_{m+1} - f_{m-1}) + \frac{b}{4\delta x}(f_{m+2} - f_{m-2})$$

with the constraint $a + 2^4 b = \frac{5!}{4!}(2\alpha)$ and $\alpha = \frac{1}{3}$, $a = \frac{1}{3}(4 + 2\alpha)$, $b = \frac{1}{3}(4\alpha - 1)$. The formulation is sixth-order accurate at the internal nodes. At the boundaries, however, the accuracy is reduced, where

$$f'_m + \alpha f'_{m+1} = \frac{1}{\delta x}(af_m + bf_{m+1} + cf_{m+2})$$

with $\alpha = 2$, $a = -\frac{5}{2}$, $b = 2$ and $c = \frac{1}{2}$. The scheme is third-order accurate at the boundary and fourth-order accurate at points adjacent to the boundary.

The approximation of a second-order derivative is

$$f''_m + \alpha(f''_{m+1} + f''_{m-1}) = \frac{a}{(\delta x)^2}(f_{m+1} - 2f_m + f_{m-1}) + \frac{b}{4(\delta x)^2}(f_{m+2} - 2f_m + f_{m-2})$$

with the constraint $a + 2^4 b = \frac{6!}{4!}\alpha$ and $\alpha = \frac{2}{11}$, $a = \frac{4}{3}(1 - \alpha)$, $b = \frac{1}{3}(10\alpha - 1)$ for sixth-order accuracy. At boundaries,

$$f''_m + \alpha f''_{m+1} = \frac{1}{(\delta x)^2}(af_m + bf_{m+1} + cf_{m+2} + df_{m+3})$$

with $\alpha = 11$, $a = 13$, $b = -27$ and $c = 15$ and $d = -1$. These schemes are computationally expensive and a tridiagonal matrix is inverted to calculate the derivatives using the efficient Thomas algorithm.

3.2.3. Time integration. A compact third-order Runge–Kutta method is used to march the solution in time [10]. The compact scheme requires minimum computational storage during a simulation. For an equation $\mathrm{d}x/\mathrm{d}t = f(x, t)$, the four stages are,

$$x^* = x^n + \tfrac{1}{4} f(x^n, t^n)\delta_t^n$$

$$x^{**} = x^n + \tfrac{8}{15} f(x^n, t^n)\delta_t^n$$

$$x^{***} = x^* + \tfrac{5}{12} f(x^{**}, t^{**})\delta_t^n$$

$$x^{n+1} = x^* + \tfrac{3}{4} f(x^{***}, t^{***})\delta_t^n$$

with

$$t^* = t^n + \tfrac{1}{4}\delta_t^n$$

$$t^{**} = t^n + \tfrac{8}{15}\delta_t^n$$

$$t^{***} = t^n + \tfrac{2}{3}\delta_t^n$$

$$t^{n+1} = t^n + \delta_t^n$$

DOI: 10.1002/fld

3.3. Numerical solution and the pressure Poisson equation

Beginning with the initial conditions, an elliptic pressure Poisson equation is obtained by decoupling pressure from the momentum equations [3]. Equation (2) can be rewritten as

$$\frac{\partial \overline{u}_i}{\partial t} = -\frac{\partial \overline{p}}{\partial x_i} + \mathscr{F}_i$$

where $\mathscr{F}_i$ represents the filtered convective, viscous and LES model terms. Discretizing the time term (one step of the Runge–Kutta scheme),

$$\frac{\overline{u}_i^{n+1} - \overline{u}_i^n}{\delta t} = -\frac{\partial \overline{p}^n}{\partial x_i} + \mathscr{F}_i^n \tag{9}$$

where n and $n+1$ are the current and next time levels, respectively, and δt is the time step. The continuity condition is enforced at $n+1$, whereby,

$$\frac{\partial \overline{u}_i^{n+1}}{\partial x_i} = 0 \tag{10}$$

and maintains incompressibility at the next time step. Taking the divergence of Equation (9) and using Equation (10), the pressure Poisson equation is obtained,

$$\frac{\partial}{\partial x_i}\frac{\partial \overline{p}^n}{\partial x_i} = \frac{\partial}{\partial x_i}\left[\frac{\overline{u}_i^n}{\delta t} + \mathscr{F}_i^n\right] \tag{11}$$

Once pressure is known, the velocity field at $n+1$ can be computed from Equation (9). Note that this formulation follows that of Harlow and Welch [3] and obviates the need for intermediate boundary conditions necessary in the fractional-step method [11]. The pressure solution is obtained using an iterative solver and is discussed later in the paper. Since explicit methods are used to discretize the equations, time-step restrictions based on the CFL (convective scale) and von Neumann (diffusive scale) criteria must be followed.

3.4. Comments on filter commutation

Near boundaries, a non-uniform mesh is used in order to reduce grid resolution requirements. A mapping function based on a Jacobian transformation is then used to obtain the spatial derivatives on the non-uniform mesh. It is known that on a non-uniform grid, the filter operation does not commute, i.e. $\overline{\partial u/\partial x} \neq \partial \overline{u}/\partial x$ and the error can be quantified as [12],

$$\overline{\frac{\partial u}{\partial x}} = \frac{\partial \overline{u}}{\partial x} - \alpha \overline{\Delta}^2 \left(\frac{\overline{\Delta}'}{\overline{\Delta}}\right) \frac{\partial^2 \overline{u}}{\partial x^2} + O(\overline{\Delta})^4$$

where $\alpha = \int \zeta^2 G(\zeta)\, \mathrm{d}\zeta$. The pressure Poisson discretization is second order, irrespective of the scheme used for convection and diffusion terms, and hence the overall accuracy of the filtered Navier–Stokes solution is second order. Therefore, the commutation error is on the same order as the finite difference error and is ignored.

DOI: 10.1002/fld

3.5. General boundary conditions

The velocity points are defined on the domain boundary, hence no-slip conditions can be directly enforced. For an inflow boundary, a known velocity field can be prescribed. For an outflow boundary, a convective outflow boundary condition is used [13],

$$\frac{\partial u}{\partial t} + \mathscr{U}\frac{\partial u}{\partial n} = 0$$

where $\mathscr{U}$ is a constant velocity calculated from the inflow, so as to conserve the mass flow rate. For the pressure Poisson equation, at the inflow boundary and along the no-slip walls, a zero normal pressure gradient is specified, $\partial\overline{p}/\partial n = 0$, and at the outflow boundary zero gage pressure is specified.

4. VERIFICATION AND VALIDATION

To ascertain the accuracy of the numerical scheme as well as the LES model, benchmark cases are simulated for laminar and turbulent flows using the partial-staggered grid. Solutions for lid driven cavity flow and flow over a backward-facing step ensure that the scheme is correctly implemented. No subgrid model was used for these two cases. For testing the LES model, a turbulent channel flow simulation is performed.

4.1. Lid-driven square cavity flow

A square cavity flow driven by a moving lid has been studied extensively in CFD research and is often used as a benchmark problem to validate solution algorithms. A two-dimensional schematic is shown in Figure 5. The geometry consists of four planes that form a square cavity and the top plane moves at a constant velocity U. Momentum is transferred by viscous forces and the flow develops downward forming a primary recirculation zone. With increasing Reynolds number (defined as $Re = LU/v$, where L is the height of the cavity), a secondary recirculation zone appears near the bottom corners of the cavity. Here the results are compared with those obtained by

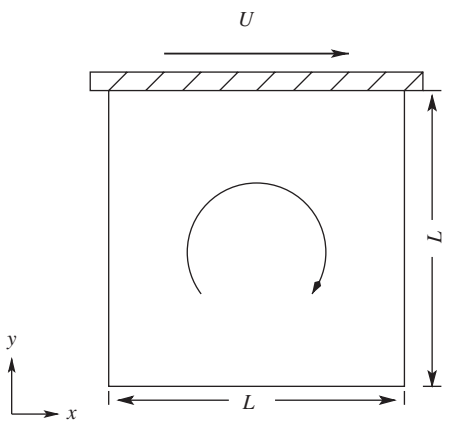

Figure 5. Two-dimensional schematic of a lid-driven square cavity flow.

DOI: 10.1002/fld

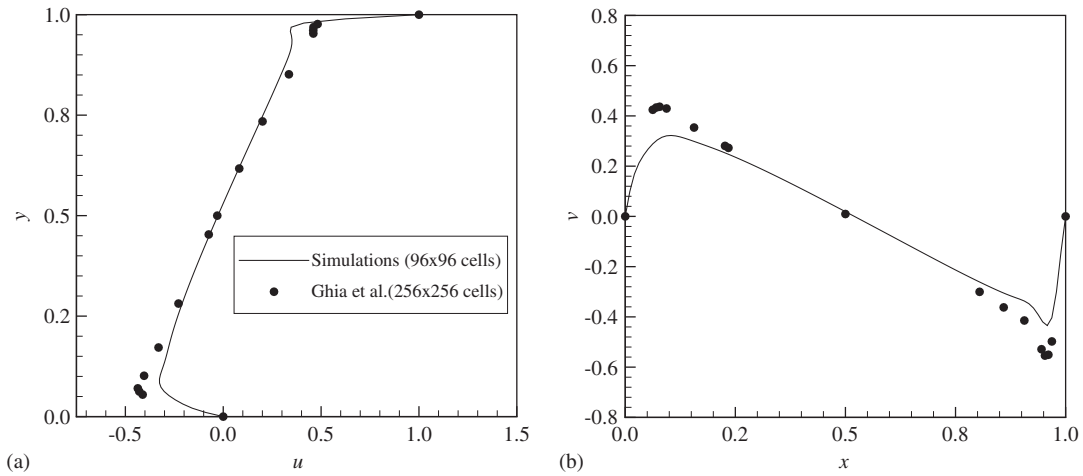

Figure 6. Profiles for the (a) u-velocity along the vertical centerline and (b) v-velocity along the horizontal centerline. Lines represent present work and symbols are results by Ghia *et al*. [14].

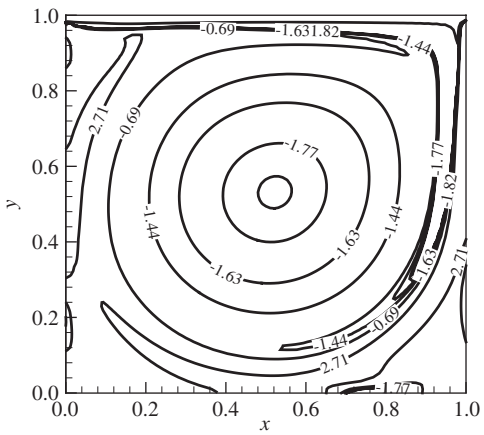

Figure 7. Two-dimensional solution of spanwise vorticity for driven cavity flow on a square domain with 96×96 cells at $Re = 5000$.

Ghia *et al*. [14] and Kim and Moin [11] at a Reynolds number of 5000. Ghia *et al*. used the vorticity-stream function approach to solve the Navier–Stokes equations using an implicit multigrid method on a mesh of size 256×256 cells. Kim *et al*. used a fractional time-stepping scheme in conjunction with an approximate factorization technique to solve the problem on a grid of size 96×96 cells. The grid size used in this work is also 96×96 cells. Figure 6 shows u- and v-velocity profiles along the vertical and horizontal centerlines of the cavity, respectively. The current simulations deviate from that of Ghia *et al*. only near the cavity walls and this is due to a lower resolution of the present simulation. The non-dimensional vorticity at the center of the primary vortex is -1.821 (refer to Figure 7) and is in close agreement with the values obtained by Ghia *et al*. with -1.860 and Kim and Moin with -1.812.

DOI: 10.1002/fld

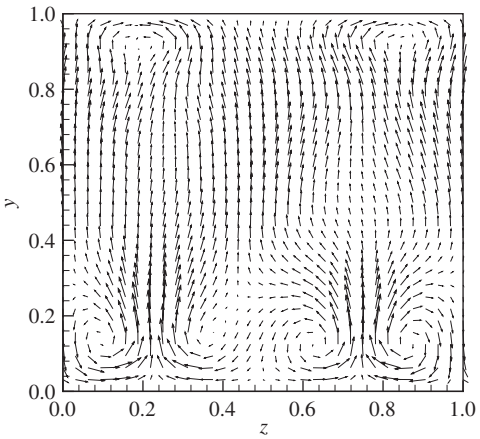

Figure 8. Velocity vectors in a (y–z) plane through the geometric center of the cubic cavity ($x = 0.5$) for three-dimensional-driven cavity flow at $Re = 1500$ to elucidate the Taylor–Görtler vortex structures.

Extensions of two-dimensional-driven cavity flows were conducted to study Taylor–Görtler vortex structures that are found in three-dimensional cavity flow experiments by Koseff *et al.* [15]. Koseff *et al.*, however, failed to reproduce this three-dimensional structure in their numerical simulations. Kim and Moin [11] were able to capture the longitudinal Taylor–Görtler vortex structures in their simulations (for Reynolds numbers greater than 900) and benchmarked their numerical scheme with this case. To check whether the partial-staggered scheme is able to capture these vortex structures, a three-dimensional-driven cavity simulation was performed on a coarse grid ($32 \times 32 \times 32$ cells) for a Reynolds number of 1500. To initialize the calculation, small random disturbances in the spanwise direction (z) were added. Periodic boundary conditions were used in the spanwise direction. Velocity vectors in a y–z plane through the geometric center ($x = 0.5$) of the cubic cavity are shown in Figure 8 for the present simulations. Although no quantitative measurements were reported in their paper, vortex structures similar to those reported by Kim and Moin are observed in the present simulations.

4.2. Flow over a backward-facing step

The flow over a backward-facing step in a channel is a good test case for the numerical method because a dissipative scheme will not predict the correct reattachment length of the recirculation zone downstream of the step. A two-dimensional schematic of the backward-facing step of height h is shown in Figure 9. The flow expands downstream of the step and reattaches at x_r forming a recirculation zone. A two-dimensional simulation of laminar flow is performed for a Reynolds number of 400. At the inflow boundary, located above the step, a parabolic velocity profile is prescribed. The downstream boundary was located at $x = 30h$ and 256×32 cells were used for the simulation. Figure 10 shows velocity vectors in the domain, colored by streamwise velocity contours. Streamlines are also shown and the reattachment length $x_r/h = 8.7$. Kim and Moin [11] report a reattachment length of 8.6 in their simulations, while experimental results at $Re = 400$ show a reattachment length of 8.4 [16].

DOI: 10.1002/fld

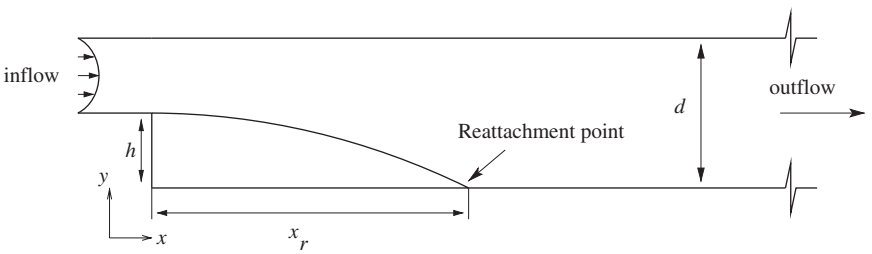

Figure 9. Flow over a backward facing step with expansion ratio $d/h = 2$.

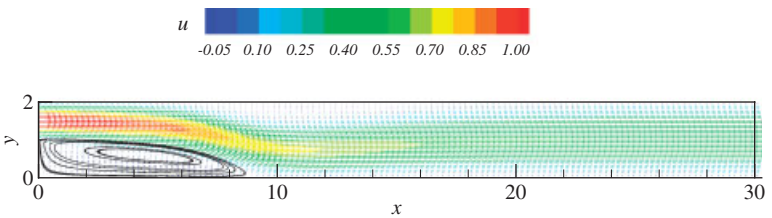

Figure 10. Velocity vectors superimposed with u-velocity contours and streamlines at $Re = 400$.

4.3. Turbulent channel flow

The driven cavity flow and flow over a backward-facing step proved the correctness of the finite difference scheme and the numerical solution procedure. To test the efficacy of the LES subgrid model on turbulent flow solutions, simulations of flow in a turbulent channel were performed. The turbulent channel flow problem is a standard benchmark case to test turbulence models. Here, the dynamic Smagorinsky model [17–19] is tested for the partial-staggered formulation and is described next.

The Smagorinsky model is an eddy-viscosity-type model, which represents the effect of the subgrid scales purely as an enhanced diffusivity for the large-scale flow [17]. The SGS stress tensor (Equation (3)) is modeled as,

$$\tau_{ij} - \tfrac{1}{3}\tau_{kk}\delta_{ij} = -2\nu_t \overline{S}_{ij}$$

where ν_t is the turbulent eddy-viscosity, δ_{ij} is the Kronecker delta function, and the filtered strain rate tensor is

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

The turbulent eddy-viscosity is defined by

$$\nu_t = (C_s\overline{\Delta})^2|\overline{S}|$$

where,

$$|\overline{S}| = \sqrt{2\overline{S}_{ij}\overline{S}_{ij}}$$

DOI: 10.1002/fld

and C_s is the Smagorinsky coefficient to be specified. The effective filter width $\overline{\Delta}$ is a function of the grid resolution and is defined as $\overline{\Delta} = (\Delta_1 \Delta_2 \Delta_3)^{1/3}$, where Δ_1, Δ_2 and Δ_3 represent the filter widths in x, y and z directions, respectively. In the dynamic model [18], the coefficient C_s is calculated 'dynamically' using information from two different length scales that are already part of the resolved solution. The procedure is generic and can be applied to any model. The coefficient C_s is a function of space and time and has been shown to be compatible with the local flow physics. Also, with the dynamic model, the eddy-viscosity asymptotically approaches zero near solid walls where viscous effects dominate and in regimes where the flow is laminar. The dynamic procedure requires a new filtering operation to be performed over a test-filter volume, with the test filter width $\widehat{\Delta}$ greater than the grid filter width $\overline{\Delta}$. An overbar, '$\overline{\ }$' denotes grid filtering (subgrid scales) and a caret, '$\widehat{\ }$' denotes test filtering (sub-test scales). For the test-filter operation, a Gaussian filter is used, represented by,

$$G(x_i - \xi_i) = \left[\frac{\sqrt{6/\pi}}{\overline{\Delta}}\right]^3 \exp\left(-\left[\frac{\sqrt{6}}{\overline{\Delta}}\right]^2 (x_i - \xi_i)^2\right)$$

On application of the test filter to the Navier–Stokes equations (Equation (2)), the sub-test scale stress tensor, T_{ij}, is obtained as,

$$T_{ij} = \widehat{\overline{u_i u_j}} - \widehat{\overline{u}}_i \widehat{\overline{u}}_j$$

The quantities τ_{ij} and T_{ij} are related by the following identity given by Germano *et al.* [18],

$$L_{ij} = T_{ij} - \widehat{\tau}_{ij} = \widehat{\overline{u_i u_j}} - \widehat{\overline{u}}_i \widehat{\overline{u}}_j$$

where L_{ij} is the resolved turbulent stress that can be explicitly calculated. The term T_{ij} is also modeled similar to the subgrid scale turbulent stress τ_{ij} using the Smagorinsky model with the same model constant C_s,

$$\tau_{ij} - \tfrac{1}{3}\tau_{kk}\delta_{ij} = -2C_s^2 \overline{\Delta}^2 |\overline{S}| \overline{S}_{ij}$$

$$T_{ij} - \tfrac{1}{3}T_{kk}\delta_{ij} = -2C_s^2 \widehat{\Delta}^2 |\widehat{\overline{S}}| \widehat{\overline{S}}_{ij}$$

$$L_{ij} - \tfrac{1}{3}\text{Ł}_{kk}\delta_{ij} = C_s^2 M_{ij}$$

where

$$M_{ij} = 2\overline{\Delta}^2 \left(\widehat{|\overline{S}|\overline{S}_{ij}} - \frac{\widehat{\Delta}^2}{\overline{\Delta}^2} |\widehat{\overline{S}}| \widehat{\overline{S}}_{ij}\right)$$

Using a least-squares approach [19], the coefficient C_s^2 is evaluated as

$$C_s^2 = \frac{\langle L_{ij} M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle}$$

where angled brackets denote averaging in the homogeneous directions.

The flow geometry is shown in Figure 11. The channel is bounded only in the normal (y) direction and extends to infinity in x and z. The streamwise (x) and spanwise (z) directions are

DOI: 10.1002/fld

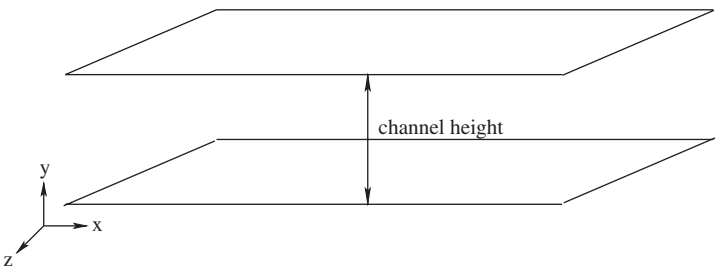

Figure 11. Schematic of turbulent channel flow. Walls at the top and bottom extend to infinity in the x and z directions.

homogeneous with periodic boundary conditions and the normal direction is bounded by no-slip walls. For the initial condition, analytical expressions for the three velocity components are used for $Re_\tau = 180$ (based on wall stress τ_w) [20],

$$u(x, y, z) = C(1 - y^8) + \varepsilon 2\pi \sin(\pi y)\cos(x)\sin(z)$$

$$v(x, y, z) = -\varepsilon(1 + \cos(\pi y))\sin(x)\sin(z)$$

$$w(x, y, z) = -\varepsilon\pi \sin(x)\sin(\pi y)\cos(z)$$

where $C = 7.764 Re_\tau^{1/7}$ is the center line velocity and $\varepsilon = 0.1C$. This initial condition satisfies boundary conditions and is divergence free. Essentially, it represents a three-dimensional disturbance superimposed on a two-dimensional mean flow. In addition, a mean streamwise pressure gradient term is added to the streamwise momentum equation ($F\delta_{i1}$). This is required to maintain the mass flow rate in the system to its initial value. The streamwise pressure gradient is adjusted at each time step [21] to provide the desired mass flow rate as,

$$F^{n+1} = F^n - \frac{1}{\Delta t}\left[\left(\frac{\dot{m}}{A_c}\right)^0 - 2\left(\frac{\dot{m}}{A_c}\right)^n + \left(\frac{\dot{m}}{A_c}\right)^{n-1}\right]$$

where $\dot{m}^0$ is the initial mass flow rate, and $\dot{m}^n$ is the average mass flow rate at time level n, Δt is the physical time step, and A_c is the cross-flow area of the channel. The average mass flow rate is given by

$$\frac{\dot{m}}{A_c} = \frac{1}{L_y L_z}\int_{-1}^{1} \langle \rho u \rangle_{x,z}\, dy$$

where $\langle\rangle_{x,z}$ denotes an ensemble average in the x and z directions. The desired mass flow rate is $(\dot{m}/A_c)^0$ for accurate conservation of mass.

The non-dimensional channel dimensions (with respect to channel half-width) are $4\pi \times 2 \times 2\pi$. Uniform grid cells are used in x and z, while stretched grid cells are used in the y-direction [22]. The domain is divided into $96 \times 64 \times 80$ cells in x, y and z directions, respectively. For the subgrid model, the dynamic Smagorinsky model is used with a test filter to grid filter ratio of 2.0 (refer to [19] for details).

DOI: 10.1002/fld

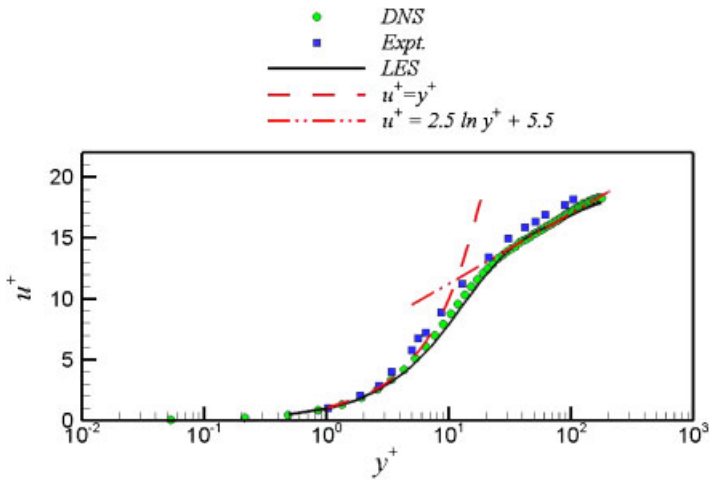

Figure 12. Mean streamwise velocity profile in wall coordinates. DNS by Kim *et al*. [23] and experiments by Niederschulte *et al*. [24].

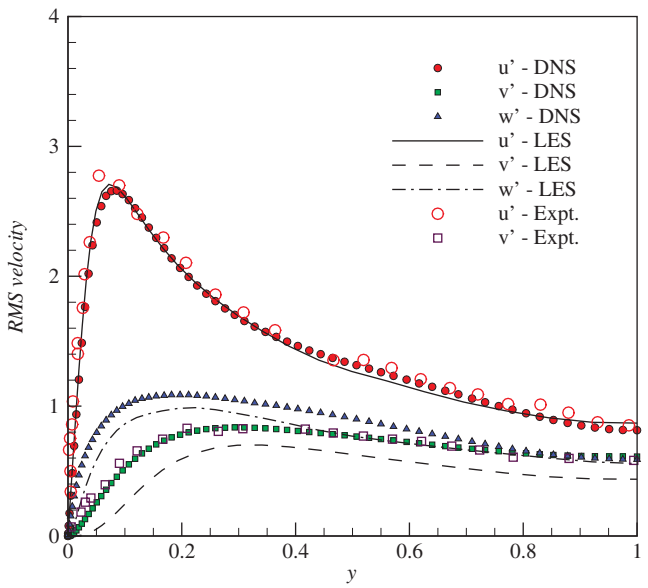

Figure 13. Root-mean-square of velocity fluctuations normalized by wall friction velocity. DNS by Kim *et al*. [23] and experiments by Niederschulte *et al*. [24].

The velocity statistics are compared with the DNS results of Kim *et al.* [23] and the experimental results of Niederschulte *et al*. [24]. The DNS study employed about 4×10^6 grid points ($192 \times 129 \times 160$ in x, y, z) and used pseudospectral methods. The experiments of Niederschulte

DOI: 10.1002/fld

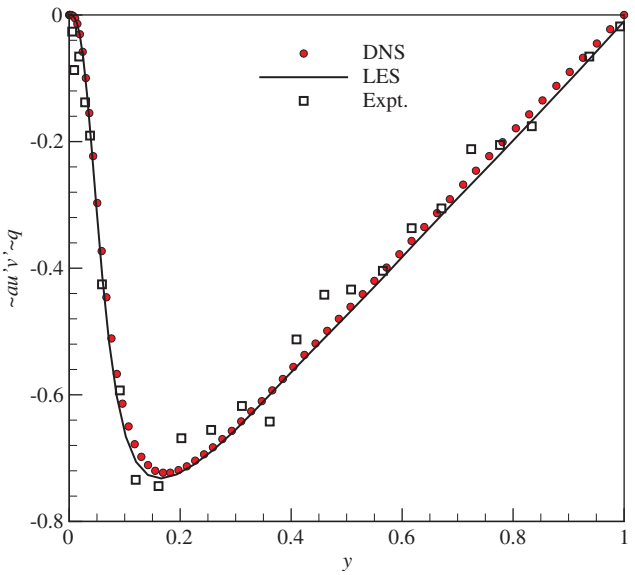

Figure 14. Normalized Reynolds shear stress. DNS by Kim *et al.* [23] and experiments by Niederschulte *et al.* [24].

were performed specifically to examine the accuracy of Kim *et al.*'s DNS results. Good, but not exact agreement was found between the experimental and DNS results. The mean streamwise profile is shown in wall coordinates in Figure 12. The mean profile was obtained by averaging in the homogeneous directions (x and z) and in time. The velocity is normalized by the friction velocity as $u^+ = u/u_\tau$, where $u_\tau = \sqrt{\tau_w/\rho}$. The dashed line and the dash-dotted line represent the law of the wall and the log law, respectively. The LES agree well with the DNS and experimental data. A value of u_τ of 0.98 was obtained in the LES calculations making the effective $Re_\tau = 176.4$. The normalized rms velocity fluctuations are shown in Figure 13. The rms velocities are underpredicted, and this is not completely unexpected, since, the LES data represent resolved scale intensities in which the small-scale contribution in not included. The Reynolds shear stress is shown in Figure 14 and compares well with the data. In general, the LES and DNS data match well, thus providing confidence in the application of the model to other general flows.

5. PARALLEL COMPUTING

As the dimensionality of system and complexity of the problem increase, the need for obtaining quick solutions becomes a priority. Computing time becomes a bottleneck for simulations of turbulent flows that require high grid resolutions. In the present case, the computation of pressure is an expensive part of the numerical solution procedure since pressure in the entire domain has to be solved simultaneously due to the nature of the elliptic Poisson equation. The incompressibility condition must be satisfied at every time step, i.e. each stage of the Runge–Kutta time integration scheme, hence, the Poisson equation is solved four times in each time step. These factors lead to the necessity of parallel computing.

DOI: 10.1002/fld

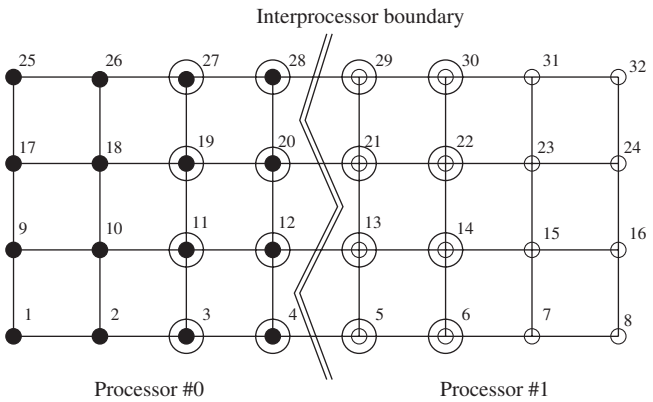

Figure 15. Partitioning of a finite difference grid for parallel computing.

MPI is a programming standard used on parallel computers with distributed memory [25]. The main idea is to use domain decomposition to break up the geometry into parts such that every processor gets its own share of calculations. A schematic of a simple two-dimensional domain is shown in Figure 15. The geometry is divided such that two processors share the computational domain, each having 16 grid points for its own calculations (small filled circles for processor 0 and small unfilled circles for processor 1). The interprocessor boundary is shown in the figure with a double line. Two layers of grid points (represented by large circles concentric to the small circles) adjacent to the interprocessor boundary represent the points over which communication is required. The MPI library provides routines for communication between processors.

Solutions to the first- and second-order derivatives for the sixth-order compact finite difference scheme are obtained by solving the Thomas algorithm. It is a direct method and the solution is obtained in two steps, forward elimination and backward substitution. Due to the substitution procedures of the algorithm, it is not scalable on a parallel machine. However, since many such forward and backward steps are required along a particular direction (e.g. x-direction) during a calculation, an overlap can be achieved in each stage to reduce the overall computational time. This technique is known as *pipelining*. Details about the procedure can be found in Stoessel *et al*. [26]. For the second-order central difference schemes, only one layer of cells is required for communication.

To solve the linear system of equations for the Poisson equation on a three-dimensional domain, a hepta-diagonal sparse matrix must be inverted. For this purpose, an efficient parallel program, *Aztec*, developed at the Sandia National Laboratory is coupled with the finite difference solver [27]. Aztec is an iterative library for solving a system of linear equations, and is designed especially for distributed memory parallel machines and it also offers a variety of preconditioned Krylov solvers. The bi-conjugate gradient method with stabilization and least-squares preconditioning is used for the present calculations. Aztec is coupled with the LES solver such that solutions can be obtained on multiple processors. The input/output (I/O) is performed individually by each processor.

5.1. Simulations of high Reynolds number flows

The parallel solver developed in the preceding sections is utilized for simulations of a confined planar jet. The schematic of the geometry is shown in Figure 16. There are three inlet streams at the

DOI: 10.1002/fld

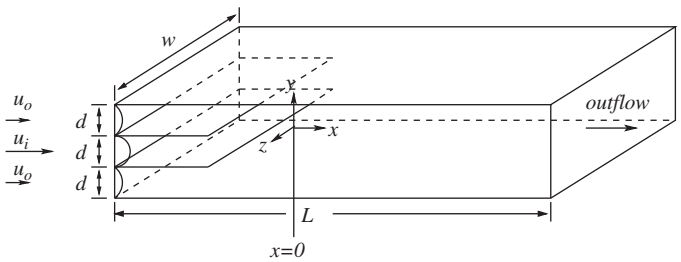

Figure 16. Schematic of a confined planar jet.

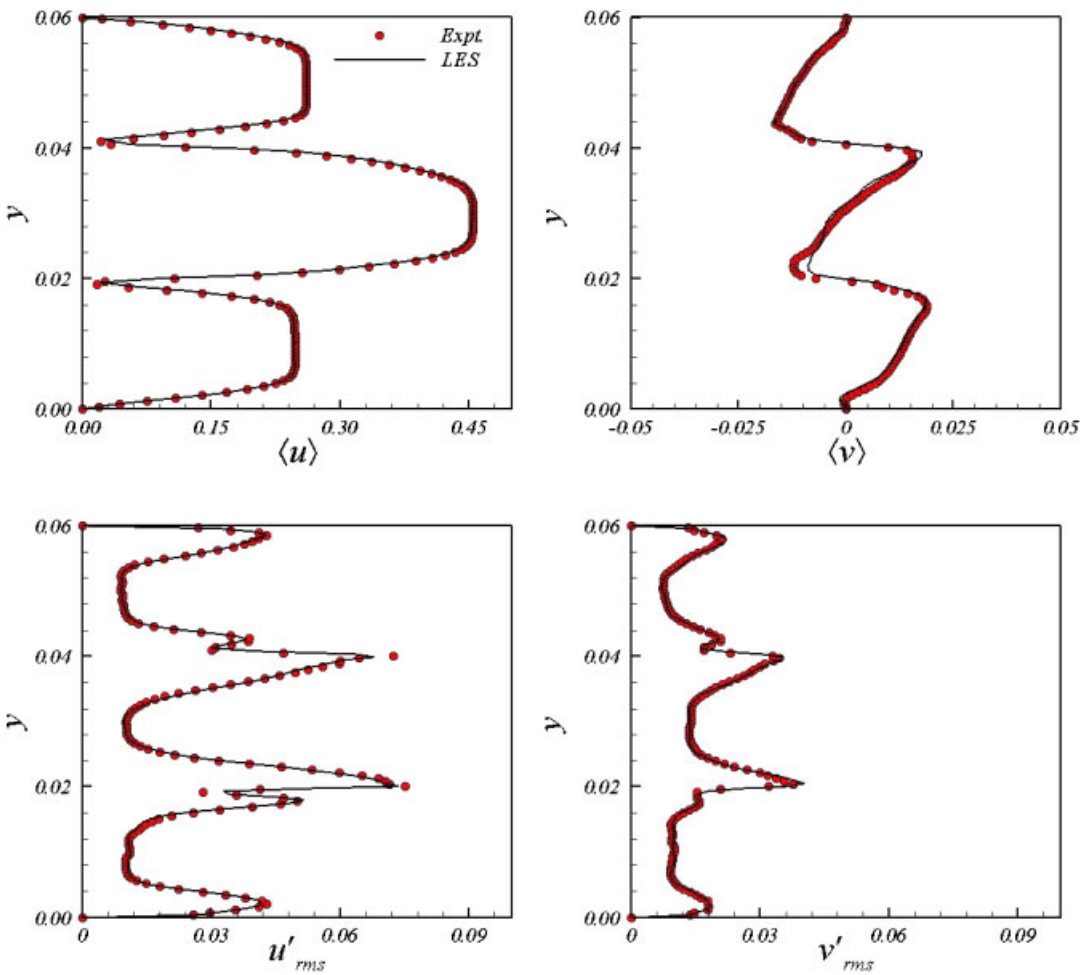

Figure 17. Inflow statistics comparing mean and rms velocities in the streamwise and cross-stream directions at $x = 0$ cm for experiments [29] and LES.

DOI: 10.1002/fld

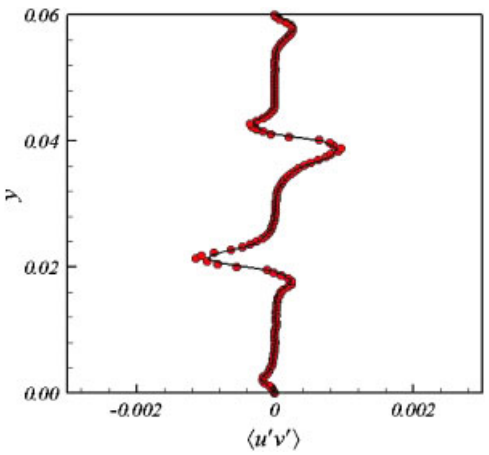

Figure 18. Inflow statistics comparing measured and LES resolved shear stress at $x = 0$ cm for experiments [29] and LES.

inflow boundary, each separated by splitter plates. The channel dimensions are $0.3 \times 0.06 \times 0.1\,\mathrm{m}^3$ (x, y, z, respectively). The volumetric flow rates in the outer streams are 0.4 L/s and in the inner stream is 0.8 L/s. The Reynolds number based on average velocity and hydraulic diameter is 20 000. The flow is bounded by no-slip walls in y- and z, directions. The simulation utilized about 2×10^6 cells ($224 \times 96 \times 96$ cells in x, y and z, respectively). Non-uniform cells were used with clustering in the shear layers and near walls. The simulations were performed on the Opteron cluster using 32 processors (8 nodes). Data for statistics were collected after three residence times so as to flush out the initial condition. Nearly 6000 realizations in time were used to compute the LES statistics. The entire simulation took about 3000 CPU (central processing unit) hours on the cluster with a CFL of 0.2.

The LES are compared with experimental data, hence, inflow conditions that are similar to those in the experiments are required for the simulation. For this purpose, the inflow conditioning technique of Klein *et al.* [28] is used. The method generates inflow velocity signals that have statistical properties similar to those in the experiments. The inflow velocities are given by

$$\overline{u}_i = \langle u_i \rangle + a_{ij} \mathscr{U}_j$$

where $\langle u_i \rangle$ is the mean inflow velocity from experiments, $\mathscr{U}_i$ is a provisional signal with a prescribed two-point statistic (from experiments), and

$$(a_{ij}) = \begin{pmatrix} (R_{11})^{1/2} & 0 & 0 \\ R_{21}/a_{11} & (R_{22} - a_{21}^2)^{1/2} & 0 \\ R_{31}/a_{11} & (R_{32} - a_{21}a_{31})/a_{22} & (R_{33} - a_{31}^2 - a_{32}^2)^{1/2} \end{pmatrix}$$

where R_{ij} is the correlation tensor known from experiments. The advantage of this method is that inflow conditions are obtained as the simulation progresses, unlike methods where prior computations are required specifically to generate and store inflow velocities. Also, the method works even though the velocity field in the inlet plane is not divergence free.

DOI: 10.1002/fld

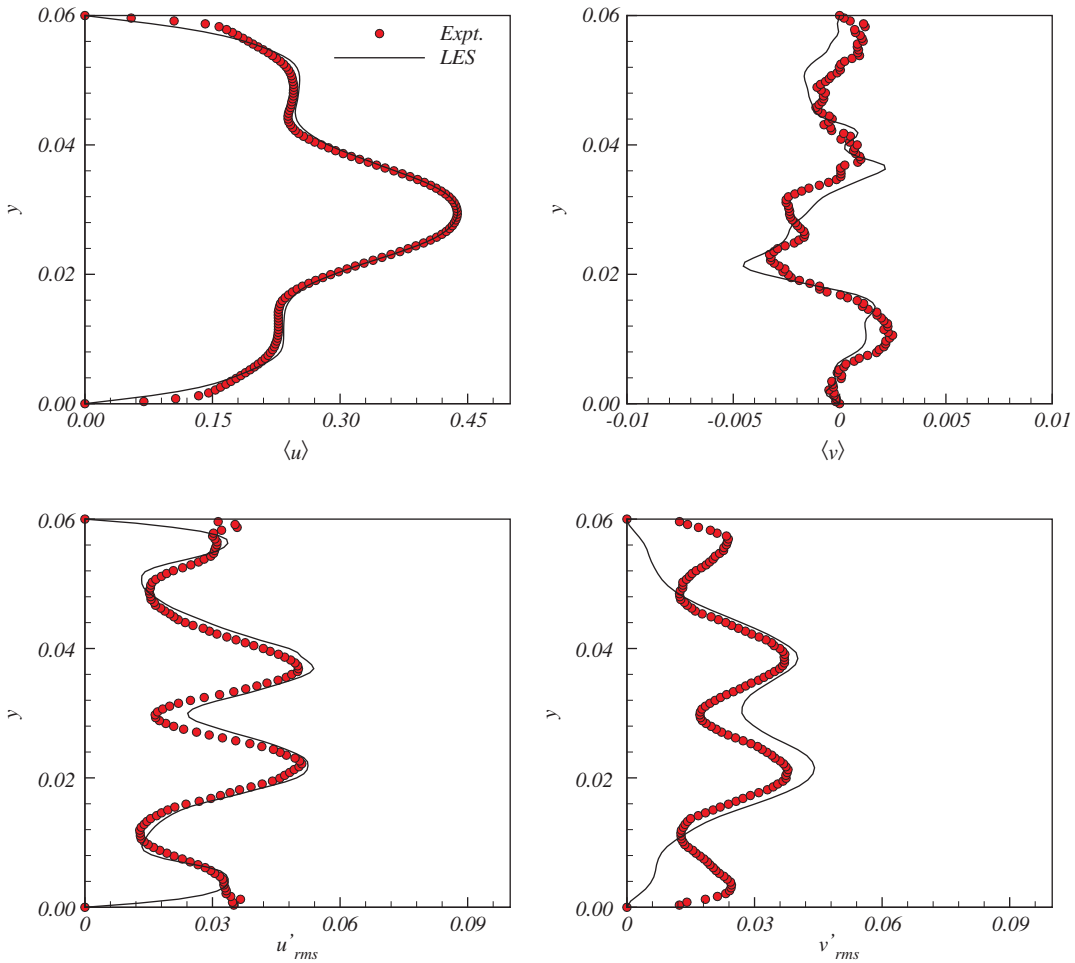

Figure 19. Comparison of mean and rms velocities in the streamwise and cross-stream directions at $x = 7$ cm for experiments [29] and LES.

PIV measurements were made for the same configurations in the x–y plane at $z = 0.05$ m. The comparison of LES and PIV data is made at two locations. The first measurement station is located 0.1 cm downstream of the splitter plate tip (due to difficulty in measurements upstream) and the PIV data at this location are used as inflow condition for the LES. This location are referred to as $x = 0$ and all locations downstream are relative to this initial position. Figures 17 and 18 compare the first- and second-order one-point statistics at $x = 0$ cm. PIV data are represented by circles and LES by lines. The inflow turbulence generator is able to reproduce velocity fields that very closely represent the PIV statistics. Statistics are also compared at one more downstream location ($x = 7$ cm) in Figures 19 and 20. The mean profile compares well with experiments but there is a slight underprediction near the upper and lower walls which is attributed to insufficient resolution. This is also noticeable in the v-rms profile. In the shear layers, good agreement is obtained between

DOI: 10.1002/fld

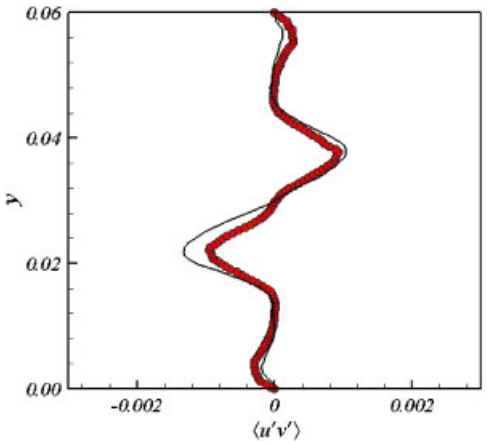

Figure 20. Comparison of resolved shear stress at $x = 7$ cm for experiments [29] and LES.

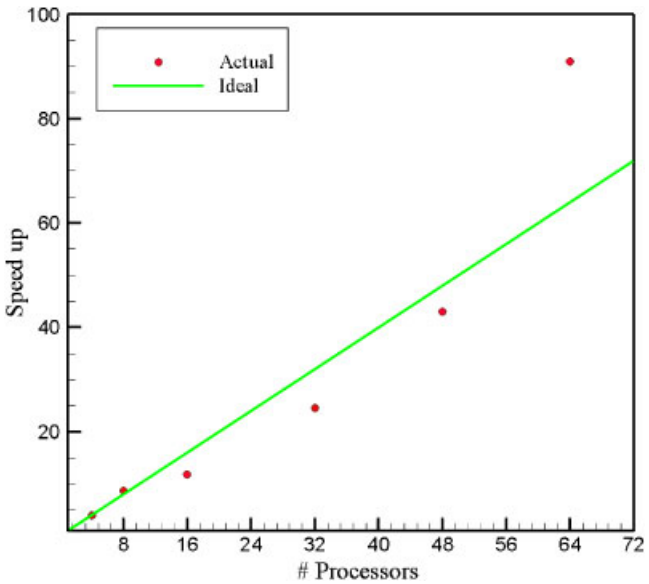

Figure 21. Performance on AMD Opteron cluster from 4 to 64 processors.

experimental data and LES. Further tests are required to determine the optimum grid size for the LES in future studies.

5.2. Parallel performance

The parallel solver is tested on an AMD Opteron cluster which has 376 processor cores (94 nodes) with 8 GB of memory per node. Each node is a dual processor, dual core 2.4 GHz AMD 280 Opteron with 1 MB on-chip cache. The nodes are interconnected with a high-performance

DOI: 10.1002/fld

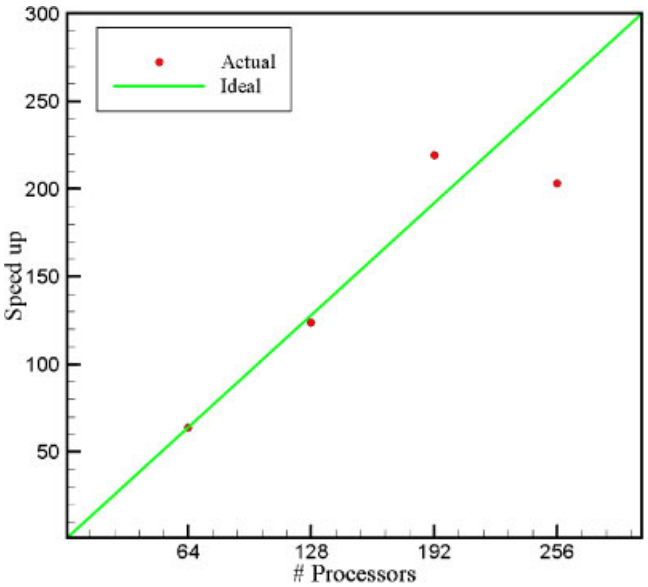

Figure 22. Performance on AMD Opteron cluster from 64 to 256 processors.

InfiniPath HTX communication network for MPI communication and a Gigabit Ethernet switch for I/O. Each test utilized at least 1 h of CPU time. Due to time limitations per processor on the cluster, the test was divided into two parts: one using 4–64 processors and another using 64–256 processors. Speed-up on n processors is generally defined as the ratio of time taken by a single processor calculation to the time taken by the n-processor calculation. For the tests using 4–64 processors, speed-up is based on four processors as a reference, and tests using 64–256 processors use a reference of 64 processors. Figures 21 and 22 show the performance curves, where symbols represent the simulations while the solid line represents the ideal case. Overall, the solver performed well on multiple processors, and a few cases show better than ideal or super-linear speed-up, which is attributed to an efficient cache utilization for those calculations.

6. SUMMARY

In this paper, an incompressible flow solver for turbulent flow simulations using large eddy simulations was described. A partial-staggered pressure grid was used to prevent oscillations in the solution while conserving mass flow rate. The solver was tested for simple laminar and turbulent flows for validation of the numerical scheme and subgrid model. MPI was utilized to reduce time and memory requirements. The sixth-order scheme was based on the pipelining method of calculation and the pressure solution was obtained using an efficient parallel iterative method. A wall-bounded, two-shear layer flow was simulated using two million cells and preliminary results were compared with PIV data obtained for the same geometry. LES compared well with data and solutions were obtained in a reasonable amount of time. Performance tests showed good speed-up on an Opteron cluster, highlighting the importance of parallel computing in LES.

DOI: 10.1002/fld

ACKNOWLEDGEMENTS

The authors wish to thank the National Science Foundation (CTS-0336435) for their financial support of this work. Additional appreciation is extended to the High Performance Computing center at Iowa State University for their computer and technical support.

REFERENCES

1. Thom A. The flow past circular cylinder at low speeds. *Procedures*, *Royal Society of London* 1933; **141**:651–666.
2. Chorin AJ. Numerical solution of Navier–Stokes equations. *Mathematics of Computation* 1968; **22**(104):745–762.
3. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible Navier–Stokes equations. *The Physics of Fluids* 1965; **8**(12):2182–2189.
4. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
5. Merkle CL, Feng JZ. Unified time-marching procedure for compressible and incompressible flows. *Journal of Hydrodynamics* 1994; **7**(4):13–21.
6. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. McGraw-Hill: London, 1980.
7. Sotiropoulos F, Abdallah S. The discrete continuity equation in primitive variable solutions of incompressible flow. *Journal of Computational Physics* 1991; **95**:212–227.
8. Fortin M, Peyret R, Temam R. *Lecture Notes in Physics*, vol. 8. Springer: New York, 1971; 337–342.
9. Lele SK. Compact finite difference schemes with spectral like resolution. *Journal of Computational Physics* 1992; **103**:16–42.
10. Wray AA. Minimal storage time-advancement schemes for spectral methods. *Technical Report*, CTR, Stanford University, 1990.
11. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
12. Ghosal S, Moin P. The basic equations for the large eddy simulation of turbulent flows in complex geometry. *Journal of Computational Physics* 1995; **118**:24–37.
13. Ferziger JH, Perić M. *Computational Methods for Fluid Dynamics*. Springer: New York, 2002.
14. Ghia U, Ghia KN, Shin CT. High-resolutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
15. Koseff JR, Street RL, Gresho PM, Upson CD, Humphrey JAC, To W-M. A three-dimensional lid-driven cavity flow: experiment and simulation. *Proceedings of the 3rd International Conference on Numerical Methods in Laminar and Turbulent Flow*, Seattle, WA, U.S.A., 1983; 564–591.
16. Armaly BF, Durst F, Pereira JCF, Schonung B. Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics* 1983; **127**:473–496.
17. Smagorinsky J. General circulation experiments with the primitive equations. Part I. The basic experiment. *Monthly Weather Review* 1963; **91**:99–164.
18. Germano M, Piomelli U, Moin P, Cabot WH. A dynamic subgrid-scale eddy viscosity model. *The Physics of Fluids A* 1991; **3**(7):1760–1765.
19. Lilly DK. A proposed modification of the Germano subgrid scale method. *The Physics of Fluids A* 1992; **4**(3):633–635.
20. Moin P, Kim J. On the numerical solution of time dependent viscous incompressible fluid flows involving solid boundaries. *Journal of Computational Physics* 1980; **35**:381–392.
21. Benocci C, Pinelli A. The role of the forcing term in the large eddy simulation of equilibrium channel flow. In *Engineering Turbulence Modeling and Experiments*, Rodi W, Ganic EN (eds). Elsevier: New York, 1990; 287–296.
22. Moin P, Kim J. Numerical investigation of turbulent channel flow. *Journal of Fluid Mechanics* 1982; **118**:341–377.
23. Kim J, Moin P, Moser R. Turbulence statistics in fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics* 1987; **177**:133–166.
24. Niederschulte MA, Adrian RJ, Hanratty TJ. Measurements of turbulent flow in a channel at low Reynolds numbers. *Experiments in Fluids* 1990; **9**:222–230.
25. Snir M, Otto S, Huss-Lederman S, Walker D, Dongarra J. *MPI–The Complete Reference*, vol. 1 (The MPI Core second edn). The MIT Press: Cambridge, MA, 2001.

DOI: 10.1002/fld

26. Stoessel A, Hilka M, Baum M. Parallel direct numerical simulation of turbulent reactive flow. *Direct Numerical Simulation for Turbulent Reacting Flows*. Éditions Technip, 1996; 33–47.
27. Tuminaro RS, Heroux M, Hutchinson SA, Shadid JN. *Official Aztec User's Guide*: *Version 2.1*. http://www.cs.sandia.gov/CRF/aztec1.html, 1999.
28. Klein M, Sadiki A, Janicka A. A digital filter based generation of inflow data for spatially developing direct numerical or large eddy simulations. *Journal of Computational Physics* 2003; **186**:652–665.
29. Feng H, Olsen MG, Liu Y, Fox RO, Hill JC. Investigation of turbulent mixing in a confined planar-jet reactor. *AIChE Journal* 2005; **51**(10):2649–2664.

DOI: 10.1002/fld