INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **58**:355–380
Published online 18 February 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1713

A collocated, iterative fractional-step method for incompressible large eddy simulation

Giridhar Jothiprasad[1,*,†] and David A. Caughey[2]

[1]*General Electric Global Research, ES-500, 1 Research Circle, Niskayuna, NY 12309, U.S.A.*
[2]*Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14853, U.S.A.*

SUMMARY

Fractional-step methods are commonly used for the time-accurate solution of incompressible Navier–Stokes (NS) equations. In this paper, a popular fractional-step method that uses pressure corrections in the projection step and its iterative variants are investigated using block-matrix analysis and an improved algorithm with reduced computational cost is developed. Since the governing equations for large eddy simulation (LES) using linear eddy-viscosity-based sub-grid models are similar in form to the incompressible NS equations, the improved algorithm is implemented in a parallel LES solver. A collocated grid layout is preferred for ease of extension to curvilinear grids. The analyzed fractional-step methods are viewed as an iterative approximation to a temporally second-order discretization. At each iteration, a linear system that has an easier block-LU decomposition compared with the original system is inverted. In order to improve the numerical efficiency and parallel performance, modified ADI sub-iterations are used in the velocity step of each iteration. Block-matrix analysis is first used to determine the number of iterations required to reduce the iterative error to the discretization error of $O\left(\Delta t^2\right)$. Next, the computational cost is reduced through the use of a reduced stencil for the pressure Poisson equation (PPE). Energy-conserving, spatially fourth-order discretizations result in a 7-point stencil in each direction for the PPE. A smaller 5-point stencil is achieved by using a second-order spatial discretization for the pressure gradient operator correcting the volume fluxes. This is shown not to reduce the spatial accuracy of the scheme, and a fourth-order continuity equation is still satisfied to machine precision. The above results are verified in three flow problems including LES of a temporal mixing layer. Copyright © 2008 John Wiley & Sons, Ltd.

Received 5 June 2007; Revised 1 November 2007; Accepted 6 November 2007

KEY WORDS: LES; fractional step; incompressible flow; parallel computing; pressure Poisson; block matrix

*Correspondence to: Giridhar Jothiprasad, General Electric Global Research, ES-500, 1 Research Circle, Niskayuna, NY 12309, U.S.A.
†E-mail: jothipra@research.ge.com

1. INTRODUCTION

Incompressible viscous flows arise in many scientific and engineering applications. The time-accurate solution of incompressible Navier–Stokes (NS) equations is important both for the simulation of high Reynolds number turbulent flow, often with turbulence modeling, as well as for 'moderate' Reynolds number flows occurring in biological systems. Hence, many numerical methods have been developed for this purpose. The mathematical difficulty in solving these equations is the elliptic behavior of the pressure waves, the speed of which is infinite. This gives rise to a mixed system of elliptic–parabolic equations for the unknown velocity components W_i and pressure p. Finite-difference and finite-volume approaches that solve directly for these unknowns are called primitive-variable approaches and can be grouped into two broad categories: artificial compressibility methods [1–4] and fractional-step methods [5–8].

Many authors have compared the relative efficiency and advantages of the different approaches, and the general conclusion is that for time-accurate solutions the pressure projection approach using the fractional-step method [7, 9, 10] is around 3 times faster than the artificial compressibility approach [11]. The fractional-step method consists of a sequence of steps that solves for the velocity and pressure at each time level. In the first step, the momentum equations are solved for a provisional velocity field using the best available estimate for the pressure distribution. This step, in general, will not yield a divergence-free velocity field unless the correct pressure distribution is employed. Hence, in the second step, the pressure distribution is improved, usually by solving a pressure Poisson equation (PPE), such that the computed pressure field will cause the velocity field to become divergence free.

The fractional-step method commonly used in unsteady, incompressible flow simulations becomes first-order accurate in time when boundaries are present. Attempts to achieve higher order accuracy in time for fractional-step methods have involved either modifying the boundary conditions (BCs) [10, 12], modifying the discretized equations being solved [13], or developing higher-order splitting schemes [14, 15]. Runge–Kutta-based fractional-step methods [16] that solve the PPE only at the last sub-step have also been developed. Most analyses of fractional-step schemes have focused on studying the loss in time accuracy near boundaries. They are based on a normal mode analysis of the unsteady Stokes equations [12, 17, 18] that are discretized only in time. However, such analyses provide limited information on the effect of spatial discretizations on the temporal and spatial accuracy.

In this paper, the block-matrix-based analysis procedure by Perot [13] is extended to analyze the accuracy of the fractional-step method originally developed by Bell *et al*. [19] and its iterative variants [20]. Many implementations [21] also treat the viscous terms implicitly using alternating direction implicit (ADI) sub-iterations when solving for the provisional velocity in the first step of the fractional step. The temporal order of the error introduced by such sub-iterations is discussed and efficient temporally second-order accurate algorithms are identified.

Finally, the computational cost of the fractional-step method is reduced through the use of a reduced stencil for the PPE that does not affect the spatial order of accuracy. Since the fractional-step method is being developed to carry out large eddy simulation (LES) of incompressible flows, the convective terms in the momentum equation and the divergence operator appearing in the continuity equation are discretized to fourth-order spatial accuracy. The use of such high-order schemes results in a very large stencil for the PPE. A method to reduce the stencil size while satisfying the discretized continuity equation to machine precision is presented. Since a significant

DOI: 10.1002/fld

fraction of the computational time in a fractional-step method is spent in solving the PPE, the reduced stencil size is important in reducing the computational cost.

As mentioned earlier, the intent is to develop a solver for the LES of incompressible flows. The governing equations for LES using linear eddy-viscosity-based sub-grid models are similar in form to the incompressible NS equations and, since the focus here is on the basic numerical algorithm rather than sub-grid models, the method is also applicable to the NS equations, themselves. The remainder of the paper is organized as follows. In Section 2, the governing equations and the numerical algorithm used are described. In Section 3, the block-matrix-based analysis procedure for the iterative fractional-step method is introduced. In Section 4, the block-matrix-based analysis procedure is used to study the order of accuracy of the fractional-step method. In Section 5, the novel scheme to reduce the stencil size of the PPE is developed. In Section 6, results are provided to substantiate the analysis. Finally, in Section 7, the contributions of the present work are summarized.

2. NUMERICAL ALGORITHM

The governing equations for LES of incompressible flows using linear eddy-viscosity-based sub-grid models are

$$\frac{\partial W_i}{\partial t}+\underbrace{\frac{\partial (W_i W_j)}{\partial x_j}}_{\text{Convective}}=-\underbrace{\frac{\partial p}{\partial x_i}}_{\text{Pressure}}+\underbrace{\frac{\partial}{\partial x_j}(2\nu S_{ij}^W)}_{\text{Viscous}}-\underbrace{\frac{\partial}{\partial x_j}(2\nu_r S_{ij}^W)}_{\text{Sub-grid}} \tag{1}$$

$$\underbrace{\frac{\partial W_i}{\partial x_i}}_{\text{Continuity}}=0 \tag{2}$$

where W_i is the resolved, or computed, LES field; p is the pressure field; $2\nu_r S_{ij}^W$ is the sub-grid scale stress tensor representing the dynamic effect of the unresolved small eddies on the evolution of the large resolved eddies and has been modeled using a linear eddy-viscosity based sub-grid model; and $S_{ij}^W=\frac{1}{2}(\partial W_i/\partial x_j+\partial W_j/\partial x_i)$ is the strain rate of the computed LES field.

A collocated grid scheme, in which one solves for both the Cartesian velocities W_i and the volume fluxes F_i, is employed here. In the collocated grid layout, the Cartesian velocity components W_i are defined at the centers, while the volume fluxes F_i are defined at the mid-points of the corresponding faces of the control volumes (cells) in computational space, see Figure 1. All scalars, such as the pressure p, viscosity ν, and eddy-viscosity ν_r, are also stored at the cell centers. This layout is advantageous as it avoids spurious oscillations in the pressure field due to pressure–velocity decoupling [22] and allows the governing equations to be relatively easily cast in strong conservation form on curvilinear grids [23].

The numerical algorithm is implemented here only for a special case of curvilinear grids, namely stretched Cartesian grids, since the primary motivation is to solve representative problems such as the temporal mixing layer. For such grids, a staggered grid layout also works well [24], but the collocated grid layout was chosen to make the code easily extensible to other complicated curvilinear grids.

DOI: 10.1002/fld

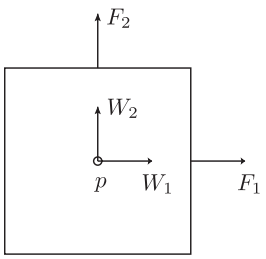

Figure 1. Dependent variables and their locations in a collocated grid layout.

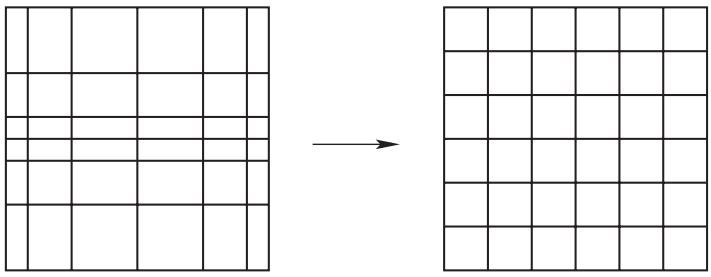

Figure 2. Mapping of stretched physical grid to uniform computational grid.

The capability of solving on stretched Cartesian grids is achieved by mapping the stretched physical grid onto a uniform computational grid (see Figure 2). Such a method is preferred because it is relatively easy to transform the governing equations to a new coordinate system. In stretched Cartesian grids, the physical grid can be constructed as a product of uni-directional mappings

$$\xi_i = \xi_i(x_i) \tag{3}$$

where ξ_i are the coordinates in computational space, and x_i are the coordinates in physical space. Hence, the transformation tensor $\partial\xi_i/\partial x_j$ that defines the properties of the transformation at any point becomes a diagonal matrix

$$\frac{\partial \xi_i}{\partial x_j} = \begin{bmatrix} \dfrac{\partial \xi_1}{\partial x_1} & 0 & 0 \\ 0 & \dfrac{\partial \xi_2}{\partial x_2} & 0 \\ 0 & 0 & \dfrac{\partial \xi_3}{\partial x_3} \end{bmatrix} \tag{4}$$

DOI: 10.1002/fld

The determinant of the transformation tensor

$$J \equiv \det\left(\frac{\partial \xi_i}{\partial x_j}\right) \tag{5}$$

is proportional to the ratio of the volumes in the computational and physical space.

Vinokur [25] and Viviand [26] have shown that the governing equations in the transformed space can be cast in strong conservation form. This form is particularly well suited for developing numerical discretizations that are momentum and mass conserving, because all the terms in this form appear as divergences of some flux. The conservation equations to be solved in the transformed computational space are given in the strong conservation form

$$\frac{\partial (J^{-1} W_i)}{\partial t} + \frac{\partial (F_m W_i)}{\partial \xi_m} = -\frac{\partial}{\partial \xi_m}\left(J^{-1}\frac{\partial \xi_m}{\partial x_i} p\right) + \frac{\partial}{\partial \xi_m}\left[J^{-1}\frac{\partial \xi_m}{\partial x_i}(v+v_r)\left\{\frac{\partial W_i}{\partial \xi_l}\frac{\partial \xi_l}{\partial x_j} + \frac{\partial W_j}{\partial \xi_n}\frac{\partial \xi_n}{\partial x_i}\right\}\right] \tag{6}$$

$$\frac{\partial F_m}{\partial \xi_m} = 0 \tag{7}$$

where $F_m = J^{-1}(\partial \xi_m / \partial x_j) W_j$ is the volume flux per unit area crossing the surface of constant ξ_m in the transformed space. Detailed derivation of the transformed governing equations can be found in computational fluid dynamics textbooks [27, 28].

A temporally second-order accurate discretization of the governing LES equations, Equations (1)–(2), on a collocated grid layout, is given by

$$\frac{W_i^{n+1} - W_i^n}{\Delta t} = -\underbrace{\left[\frac{3}{2} H_i\left\{\mathbf{W^n}, \mathbf{F^n}\right\} - \frac{1}{2} H_i\{\mathbf{W^{n-1}}, \mathbf{F^{n-1}}\}\right]}_{\text{Convective, Adams–Bashforth}} - \underbrace{G_i^W\{p^{n+1}\}}_{\text{Pressure gradient}} + \underbrace{\frac{1}{2}[D2\{W_i^{n+1}; v, v_r^{n+1}\} + D2\{W_i^n; v, v_r^n\}]}_{\text{Sub-grid+Viscous dissipation, Crank–Nicolson}} \tag{8}$$

$$B_i\{\mathbf{W^{n+1}}\} = F_i^{n+1} \tag{9}$$

$$\underbrace{C^{\mathrm{F}}\{\mathbf{F^{n+1}}\} = 0}_{\text{Continuity}} \tag{10}$$

The difference from a staggered grid implementation is that the discretization includes an interpolation equation for the volume fluxes, Equation (9), and the continuity equation is discretized based on these volume fluxes defined at the cell faces instead of on the velocities defined at the

DOI: 10.1002/fld

cell centers. Also, in these equations:

- superscripts $()^n$ denote the time level;
- $H_i\{\mathbf{W}, \mathbf{F}\}$ is the spatial discretization of the convective term $\partial(W_i W_j)/\partial x_i$ computed using the given velocity field $\mathbf{W}$ and volume flux $\mathbf{F}$;
- $D2\{W_i; \nu, \nu_r\}$ is the spatial discretization of the viscous and the linear eddy-viscosity sub-grid model terms $\partial/\partial x_j[2(\nu+\nu_r)S_{ij}^W]$;
- $G_i^W\{p\}$ is the spatial discretization of the pressure gradient, $\partial p/\partial x_i$, computed at the cell centers, and the superscript W denotes that this discretization is used to advance the velocities W_i and not the volume fluxes F_i;
- $B_i\{\mathbf{W}\}$ is the operator used to compute the volume fluxes F_i by interpolating velocity $\mathbf{W}$ to the cell faces and multiplying by the area of the appropriate cell face.
- $C^{\mathrm{F}}\{\mathbf{F}\}$ is the spatially fourth-order accurate discrete continuity operator based on the volume fluxes F_i instead of on velocity W_i.

The convective terms are discretized explicitly in time using the Adams–Bashforth scheme. The explicit discretization of the convective terms imposes a Courant–Friedrichs–Lewy (CFL) stability condition ($(W_i\Delta t/\Delta x) \sim O(1)$) requiring the time step Δt to be proportional to the grid spacing Δx. However, such a stability condition is not overly restrictive because accuracy considerations for LES also require the CFL number to be of order one. On the other hand, the viscous terms and sub-grid stresses are treated implicitly using the Crank–Nicolson scheme. This removes the more restrictive Fourier stability condition ($(\nu\Delta t/\Delta x^2) \sim O(1)$), which requires Δt to be proportional to Δx^2. Particularly in regions of grid refinement, the elimination of the Fourier stability constraint is advantageous, making the above choices for the time discretization quite popular [10, 23].

In the differential (not discretized) equations, the convective and pressure terms simply rearrange kinetic energy within the domain but do not dissipate energy. It has been observed that the stability and accuracy of numerical schemes improves [24, 29] if the discrete version satisfies these energy-conservation properties as well. Hence, the convective term is approximated by the energy-conserving, spatially fourth-order accurate approximation

$$\frac{\partial(F_m W_i)}{\partial \xi_m} \overset{C}{=} H_i\{\mathbf{W}^{\mathbf{n}}, \mathbf{F}^{\mathbf{n}}\} = \frac{9}{8}\frac{\delta_1[F_m \overline{W_i}^{1\xi_i}]}{\delta_1 \xi_m} - \frac{1}{8}\frac{\delta_3[F_m \overline{W_i}^{3\xi_i}]}{\delta_3 \xi_m} \tag{11}$$

where

$$\left.\frac{\delta_n \phi}{\delta_n \xi_1}\right|_{\xi_1,\xi_2,\xi_3} \equiv \frac{\phi(\xi_1+(n/2)\Delta\xi_1, \xi_2, \xi_3) - \phi(\xi_1-(n/2)\Delta\xi_1, \xi_2, \xi_3)}{n\Delta\xi_1} \tag{12}$$

$$\overline{\phi}^{n\xi_1}|_{\xi_1,\xi_2,\xi_3} \equiv \frac{\phi(\xi_1+(n/2)\Delta\xi_1, \xi_2, \xi_3) + \phi(\xi_1-(n/2)\Delta\xi_1, \xi_2, \xi_3)}{2} \tag{13}$$

It can be proved [29] that the above approximation for the convective term conserves energy if the discrete continuity operator is chosen to be

$$\frac{\partial(F_m)}{\partial \xi_m} \overset{C}{=} C^{\mathrm{F}}\{\mathbf{F}\} = \frac{9}{8}\frac{\delta_1 F_m}{\delta_1 \xi_m} - \frac{1}{8}\frac{\delta_3 F_m}{\delta_3 \xi_m} \tag{14}$$

The above is clearly a fourth-order accurate approximation to the continuity equation.

DOI: 10.1002/fld

Lastly, the volume fluxes at the cell faces are determined from the velocities at the cell centers using the fourth-order interpolation

$$F_i \equiv [\tfrac{9}{8}\,\overline{W_i}^{1\xi_i} - \tfrac{1}{8}\,\overline{W_i}^{3\xi_i}] * A_i \tag{15}$$

where A_i is the surface area of the cell face perpendicular to ξ_i.

3. BLOCK-MATRIX ANALYSIS OF ITERATIVE FRACTIONAL-STEP METHOD

Block-matrix analysis of the fractional-step method was first introduced by Perot [13]. Perot used this analysis to describe the analogy between a temporally first-order accurate fractional-step method and the block-LU factorization of the discretized equations on staggered grids. He further extended the analysis to develop a temporally second-order accurate fractional-step method. Here, the block-matrix analysis is used to study the temporally second-order accurate fractional-step methods introduced by Bell *et al.* [19] and its iterative variants [20] on a collocated grid layout. The analysis presented in this section is extended in later sections to study the order of accuracy and reduce the stencil of the PPE.

The discretized equations, Equations (8)–(10), are rearranged so that all the unknowns, $\mathbf{W^{n+1}}$, $\mathbf{F^{n+1}}$, and p^{n+1}, appear on the left-hand side:

$$\begin{aligned}
&\left\{\frac{I}{\Delta t} - \frac{1}{2}D2\{;v^{n+1}, v_r^{n+1}\}\right\} W_i^{n+1} + G_i^W\{p^{n+1}\} \\
&\quad = -\left[\frac{3}{2}H_i\{\mathbf{W^n}, \mathbf{F^n}\} - \frac{1}{2}H_i\{\mathbf{W^{n-1}}, \mathbf{F^{n-1}}\}\right] \\
&\qquad + \left\{\frac{I}{\Delta t} + \frac{1}{2}D2\{;v^{n}, v_r^{n}\}\right\} W_i^{n} + \mathrm{BCs}
\end{aligned} \tag{16}$$

$$B_i\{\mathbf{W^{n+1}}\} - F_i^{n+1} = 0 \tag{17}$$

$$C^{\mathrm{F}}\{\mathbf{F^{n+1}}\} = 0 \tag{18}$$

Note that at each time step, a (nearly) linear system of equations subject to linear constraints needs to be inverted (the dependence of the eddy viscosity on the velocity field introduces a weak non-linearity). However, the constraints imposed by the discrete continuity equation make it difficult to solve this system of equations while maintaining second-order temporal accuracy. Also, the BCs are applied on $\mathbf{W^{n+1}}$ and not on the provisional velocities that appear in the middle of the fractional step. Hence, the BCs are an implicit part of the spatial discretization operators and appear partly as source terms on the right-hand side (RHS). As mentioned earlier, other researchers have developed schemes that modify the velocity BCs so as to apply them on the provisional velocities and achieve second-order temporal accuracy; but such schemes are not analyzed in this paper.

DOI: 10.1002/fld

The block-matrix form of the discretized equations, Equations (8)–(10), are given by

$$\begin{bmatrix} A & 0 & G^W \\ -B & I & 0 \\ 0 & C^{\mathrm{F}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{W^{n+1}} \\ \mathbf{F^{n+1}} \\ p^{n+1} \end{bmatrix} = \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix} + \mathrm{BCs} \tag{19}$$

where

$$\mathbf{W} \equiv \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}, \quad \mathbf{F} \equiv \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}, \quad G^W \equiv \begin{bmatrix} G_1^W \\ G_2^W \\ G_3^W \end{bmatrix}$$

$$A \equiv \left(\frac{I}{\Delta t} - \frac{1}{2} L\{; v^{n+1}, v_r^{n+1}\} \right)$$

$$L\{; v, v_r\} \equiv \begin{bmatrix} D2\{; v, v_r\} & 0 & 0 \\ 0 & D2\{; v, v_r\} & 0 \\ 0 & 0 & D2\{; v, v_r\} \end{bmatrix}$$

$$s \equiv \text{Source terms}$$

$$B \equiv \text{Interpolation operator to obtain } \mathbf{F} \text{ from } \mathbf{W}$$

Equation (19) is easily seen to be of the form, $Px = b$. Since P is hard to invert directly, an iterative procedure is used in which P is replaced by a matrix M which is easier to invert. At the mth iteration

$$M\delta x^{[m]} = b - Px^{[m-1]} \tag{20}$$

is solved to obtain a better estimate for the solution $x^{[m]} = x^{[m-1]} + \delta x^{[m]}$, by using the residual of the earlier estimate, $b - Px^{[m-1]}$. More details on the convergence and stability of such iterative schemes can found in [30].

An efficient iterative scheme requires that M be *easy to invert* and be *close to* P so that the iterative procedure converges in a few (2 or 3) iterations. In the iterative fractional-step method, M is obtained by approximating P with

$$P = \begin{bmatrix} A & 0 & G^W \\ -B & I & 0 \\ 0 & C^{\mathrm{F}} & 0 \end{bmatrix} \xrightarrow[\text{approx.}]{} M \equiv \begin{bmatrix} \hat{A} & 0 & \Delta t \hat{A} G^W \\ -B & I & -\Delta t B G^W + \Delta t G^{\mathrm{F}} \\ 0 & C^{\mathrm{F}} & 0 \end{bmatrix} \tag{21}$$

where

$$G^{\mathrm{F}} \equiv \begin{bmatrix} G_1^{\mathrm{F}} \\ G_2^{\mathrm{F}} \\ G_3^{\mathrm{F}} \end{bmatrix}$$

DOI: 10.1002/fld

represents the spatial discretization of the pressure gradient multiplied by the surface area of cell face normal to x_i, $\partial p/\partial x_i * A_i$, that is used to correct the volume fluxes F_i. Here, the superscript F denotes the fact that this discretization is used to correct the volume fluxes F_i and not the velocities W_i. The following approximations are made:

- *A is replaced by $\hat{A}$*: The inversion of M requires the inversion of A as will be seen later. Since it is hard to invert A directly, A is often inverted using ADI factorizations or iterative schemes based on ADI factorizations. Hence, sub-iterations to invert A within each iteration of the iterative fractional-step method are present. Since only a finite number of sub-iterations is performed, this introduces an iterative error which can be interpreted as exactly inverting a nearby matrix, $\hat{A}$. The exact form of $\hat{A}$ depends on the sub-iterations used and the number of sub-iterations. Details on the sub-iteration to invert A are given later. Note that A is dominated by $I/\Delta t$ for small Δt. Hence A is of order $O(1/\Delta t)$, and the sub-iteration is chosen such that $A-\hat{A}$ is either of order $O(1)$ or $O(\Delta t)$.
- *G is replaced by $\Delta t\hat{A}G^W$ and 0 is replaced by $-\Delta t BG+\Delta t G^{\mathrm{F}}$*: This approximation makes it easier to invert M compared with P, as there is a block-LU decomposition for M in closed form

$$M \equiv \begin{bmatrix} \hat{A} & 0 & \Delta t\hat{A}G^W \\ -B & I & -\Delta t BG^W+\Delta t G^{\mathrm{F}} \\ 0 & C^{\mathrm{F}} & 0 \end{bmatrix} = \begin{bmatrix} \hat{A} & 0 & 0 \\ -B & I & 0 \\ 0 & C^{\mathrm{F}} & -\Delta t C^{\mathrm{F}}G^{\mathrm{F}} \end{bmatrix} \begin{bmatrix} I & 0 & \Delta t G^W \\ 0 & I & \Delta t G^{\mathrm{F}} \\ 0 & 0 & I \end{bmatrix} \tag{22}$$

  Again note that $\hat{A}$ is dominated by $I/\Delta t$ for small Δt; hence, $G^W-\Delta t\hat{A}G^W$ is of order $O(\Delta t)$. The other approximation introduces an $O(\Delta t)$ error as well.

Next, that M is close to P is demonstrated by comparing order estimates of the various matrices. Because P is dominated by the $I/\Delta t$ term for small Δt, it is $O(1/\Delta t)$:

$$P = \begin{bmatrix} \dfrac{I}{\Delta t}+\cdots & \cdots & \cdots \\ \cdots & \cdots & 0 \\ 0 & \cdots & 0 \end{bmatrix} = O\left(\frac{1}{\Delta t}\right) \tag{23}$$

The changes introduced to approximate P with M introduce at most only an $O(1)$ difference between the two matrices:

$$M-P = \begin{bmatrix} \hat{A}-A & 0 & (\Delta t\hat{A}-I)G^W \\ 0 & 0 & \Delta t(BG^W-G^{\mathrm{F}}) \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} O(1) \text{ or } O(\Delta t) & 0 & O(\Delta t) \\ 0 & 0 & O(\Delta t) \\ 0 & 0 & 0 \end{bmatrix} \tag{24}$$

Hence, M is not only close to P but also much easier to invert than P. In practice, only two or three iterations are usually sufficient to adequately converge the process.

The similarity to the conventional description of the fractional-step method can be seen by expanding the steps involved in the mth iteration of the iterative fractional-step procedure. The

DOI: 10.1002/fld

equations solved in the mth iteration are

$$\begin{bmatrix} \hat{A} & 0 & \Delta t\hat{A}G^W \\ -B & I & -\Delta t BG^W + \Delta t G^{\mathrm{F}} \\ 0 & C^{\mathrm{F}} & 0 \end{bmatrix} \begin{bmatrix} \delta\mathbf{W}^{[\mathbf{m}]} \\ \delta\mathbf{F}^{[\mathbf{m}]} \\ \delta p^{[m]} \end{bmatrix} = r^{[m-1]} = \begin{bmatrix} s \\ 0 \\ 0 \end{bmatrix}$$

$$+\mathrm{BCs} - \begin{bmatrix} A & 0 & G^W \\ -B & I & 0 \\ 0 & C^{\mathrm{F}} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{W}^{[\mathbf{m-1}]} \\ \mathbf{F}^{[\mathbf{m-1}]} \\ p^{[m-1]} \end{bmatrix} \quad (25)$$

The steps involved in this block-LU decomposition are detailed below:

1. Solve for a provisional velocity $\hat{\mathbf{W}}$ using the latest approximation to the pressure $p^{[m-1]}$ according to

$$\hat{A}(\delta\hat{\mathbf{W}}) = s + \mathrm{BCs} - AW^{[m-1]} - G^W p^{[m-1]} \quad (26)$$

   where the provisional velocity, $\hat{\mathbf{W}} = \mathbf{W}^{[\mathbf{m-1}]} + \delta\hat{\mathbf{W}}$.
2. Compute the provisional volume flux $\hat{\mathbf{F}}$ by interpolating the velocity $\hat{\mathbf{W}}$ to the cell faces using

$$\hat{\mathbf{F}} = B\hat{\mathbf{W}} \quad (27)$$

3. Solve the PPE

$$C^{\mathrm{F}}G^{\mathrm{F}}\{\delta p^{[m]}\} = \frac{C^{\mathrm{F}}\{\hat{\mathbf{F}}\}}{\Delta t} \quad (28)$$

   for the pressure correction $\delta p^{[m]}$ that projects the *provisional volume flux* onto a divergence-free space.
4. Correct the provisional volume flux using the pressure correction $\delta p^{[m]}$ according to

$$\mathbf{F}^{[\mathbf{m}]} = \hat{\mathbf{F}} - (\Delta t)G^{\mathrm{F}}\{\delta p^{[m]}\} \quad (29)$$

   so that it becomes divergence free.
5. Correct the provisional velocity using the pressure correction $\delta p^{[m]}$ according to

$$\mathbf{W}^{[\mathbf{m}]} = \hat{\mathbf{W}} - (\Delta t)G^W\{\delta p^{[m]}\} \quad (30)$$

6. Update the pressure according to

$$p^{[m]} = p^{[m-1]} + \delta p^{[m]}$$

Hence, performing a single iteration is equivalent to the fractional-step method introduced by Bell *et al.* [19], while performing more than one iteration is equivalent to its iterative variants [20].

DOI: 10.1002/fld

4. TEMPORAL ACCURACY OF THE ITERATION ERROR

Although the original discretized equations are second-order accurate in time, the iteration error introduced by performing only a finite number of iterations can reduce the temporal order of accuracy. In this section, the minimum number of iterations needed to reduce the temporal accuracy of the iteration error to that of the numerical discretization error of $O(\Delta t^2)$ is determined. Details on the choice of $\hat{A}$ and its influence on the temporal order of accuracy are also discussed.

The iteration error in the mth iteration is defined to be the residual

$$r^{[m]} \equiv b - Px^{[m]} \tag{31}$$

computed using the mth estimate for the solution $x^{[m]}$. Equation (20) can be easily reduced to show that

$$r^{[m]} = (NM^{-1})r^{[m-1]} \tag{32}$$

where $N \equiv M - P$. Applying the above equation recursively gives

$$r^{[m]} = (NM^{-1})^m r^{[0]} \tag{33}$$

Next, order estimates of N, M and $r^{[0]}$ are used to show that $r^{[m]} = O(\Delta t^m)$ or $O(\Delta t^{2m})$. It is seen from Equation (23) that M is of order of $O(1/\Delta t)$. In order to make an estimate of $r^{[0]}$, it is necessary to know $x^{[0]}$. The present implementation takes $x^{[0]}$ to be given by the values at the previous time level

$$x^{[0]} = \begin{bmatrix} \mathbf{W^n} \\ \mathbf{F^n} \\ p^n \end{bmatrix} \tag{34}$$

hence, $r^{[0]}$ is $O(1)$. Equation (24) shows that there are two cases for the temporal order of N, which accordingly determines the order of the iteration error:

1. *N is $O(1)$*: In this case, (NM^{-1}) is of $O(\Delta t)$. Using these order estimates and Equation (32), it can be seen that $r^{[m]}$ is $O(\Delta t^m)$. Hence, if two iterations are performed the iteration error becomes the same order as the numerical discretization error, which is $O(\Delta t^2)$. If three iterations are performed, the iterative error becomes $O(\Delta t^3)$, but, because the discretization error is still $O(\Delta t^2)$, the overall scheme continues to be second-order accurate in time.
2. *N is $O(\Delta t)$*: In this case, (NM^{-1}) is of $O(\Delta t^2)$. Again, using these order estimates and Equation (32), it can be seen that $r^{[m]}$ is $O(\Delta t^{2m})$. Hence, after a single iteration the iteration error becomes the same order as the numerical discretization error which is $O(\Delta t^2)$.

Hence, depending on the order of N, one or two iterations are sufficient to achieve the desired second-order temporal accuracy. In Section 6, numerical results are shown to confirm this result, and the computational efficiency in typical flow problems is used to comment on the relative merits of the two cases.

Equation (24) also shows that the choice of $\hat{A}$ largely determines the temporal order of N. Hence, the choice of $\hat{A}$ in the current parallel implementation is briefly discussed. As mentioned

DOI: 10.1002/fld

earlier, a finite number of sub-iterations of an iterative scheme is used to invert A. Starting with the matrix A, successive approximations are introduced to obtain the matrix A' that is inverted at each sub-iteration of the iterative scheme. The definition of $\hat{A}$ depends both on the matrix A' and the number of sub-iterations performed.

The matrix A, defined in Equation (19), arises from the spatial discretization of viscous and residual stress terms and consists of discretizations of second-order normal derivatives such as $\partial/\partial x_1((\,)\partial(\,)/\partial x_1)$ and cross derivatives such as $\partial/\partial x_1((\,)\partial(\,)/\partial x_2)$,

$$\begin{aligned} A &\equiv \frac{I}{\Delta t} - \frac{1}{2} L\{; v^{n+1}, v_r^{n+1}\} \\ &= \frac{I}{\Delta t} - \frac{1}{2}(L_{x_1}\{;\} + L_{x_2}\{;\} + L_{x_3}\{;\} + L_{\text{cross}}\{;\}) \end{aligned} \tag{35}$$

where $L_{x_1}\{;\}$ is the spatially fourth-order accurate discretization of the corresponding normal derivative term, $\frac{\partial}{\partial x_1}((\,)\partial(\,)/\partial x_1)$; and $L_{\text{cross}}\{;\}$ represents the spatially second-order accurate discretization of all the cross-derivative terms. The cross-derivative terms are the only terms discretized to second-order spatial accuracy; all other terms are discretized to fourth-order spatial accuracy. In incompressible flow simulations with constant viscosity, there are no cross derivatives and so the overall scheme is spatially fourth-order accurate.

The first approximation to A is based on the ADI factorization

$$A''' = \frac{1}{\Delta t}\left(I - \frac{\Delta t}{2} L_{x_1}\right)\left(I - \frac{\Delta t}{2} L_{x_2}\right)\left(I - \frac{\Delta t}{2} L_{x_3}\right) \tag{36}$$

In the above, at each sub-iteration, the cross-derivative terms are treated explicitly and the normal derivative terms in the different directions have been factored out separately. Each factor requires the inversion of linear systems for unknowns along lines in a single direction, say x_1, thereby reducing the bandwidth of the linear system being inverted. Expanding A''', it can be seen that the leading order temporal error introduced by the above approximation is $O(1)$

$$\begin{aligned} A - A''' &= \left[\frac{I}{\Delta t} - \frac{1}{2}(L_{x_1} + L_{x_2} + L_{x_3} + L_{\text{cross}})\right] \\ &\quad - \left[\frac{I}{\Delta t} - \frac{1}{2}(L_{x_1} + L_{x_2} + L_{x_3}) + O(\Delta t)\right] \\ &= O(1) \end{aligned} \tag{37}$$

The next approximation is motivated by the need to reduce the communication time between processes in a parallel implementation. On a distributed memory parallel machine, data need to be divided or *domain decomposed* among the various processes. In the current implementation, data in the x_1x_2 planes are stored in each process but are divided along the x_3 direction. Hence, the inversion along lines in the x_3 direction (due to the factor $(I - (\Delta t/2)L_{x_3})$) requires swapping from x_1x_2 planes in a process to x_2x_3 planes. Such swapping requires communication of $O(N_1N_2N_3)$ variables making the computation and communication requirements to be of the same order. The fact that the planes must be swapped in each sub-iteration further exacerbates the problem.

DOI: 10.1002/fld

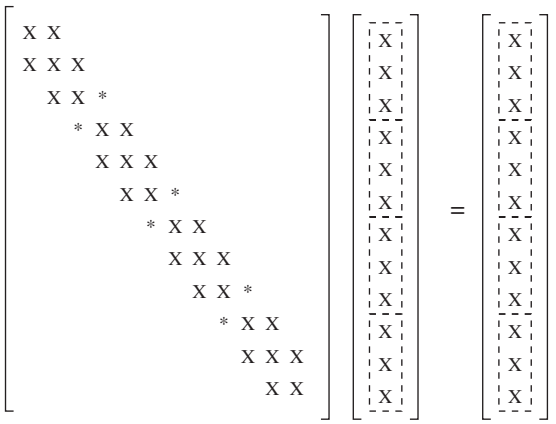

Figure 3. Schematic representation of a tridiagonal matrix in which non-zero elements are denoted by crosses and asterisks. Dashed box denotes the division of data among four processes. Elements denoted by asterisks necessitate communication between process for solving the linear system.

In order to address this problem, the linear system arising in the x_3 direction is modified so that no communication is required between the processes. Typically, it is necessary to solve a banded linear system along lines in the x_3 direction. Figure 3 is a schematic representation of a tridiagonal matrix in which the non-zero elements are denoted by either crosses or asterisks. The division of data among, say, four processes is denoted by dashed boxes. If the elements denoted by asterisks were zero, the system would correspond to four smaller linear systems which could be solved without the exchange of data between the processes.

Thus, in order to reduce the communication requirements the original matrix system arising along lines in the x_3 direction is approximated by taking the elements denoted by asterisks to be zero. This is here referred to as using *broken lines in the x_3 direction* and denoted by the operator $(I-(\Delta t/2)L^B_{x_3})$. In the definition for the matrix A''', Equation (36), $(I+(\Delta t/2)L_{x_3})$ has been approximated by broken lines in the x_3 direction

$$A''=\frac{1}{\Delta t}\left(I-\frac{\Delta t}{2}L_{x_1}\right)\left(I-\frac{\Delta t}{2}L_{x_2}\right)\left(I-\frac{\Delta t}{2}L^B_{x_3}\right) \tag{38}$$

The inversion of A'' now requires *no communication* between the processes. An algorithm very similar to the above can be found in [21].

Figure 4 compares the parallel speedup using broken lines in the x_3 direction against the original ADI factorization, Equation (36). The SPEC number is the number of times the given subroutine can be executed in a unit of time (say, a day) on a given machine. Here, the SPEC number for the subroutine that performs the matrix inversion of A'' in each sub-iteration is plotted as a function of the number of parallel processes. It is clear that the broken lines modification in the x_3 direction results in nearly linear speedup with additional processes.

In the broken lines approach, the number of off-diagonal elements neglected in the banded linear system along lines in the x_3 direction increases with the number of processes. Hence, the iteration error after a finite number of sub-iterations is expected to increase with the number of processes. However, irrespective of the number of processes, the temporal order of the error introduced by

DOI: 10.1002/fld

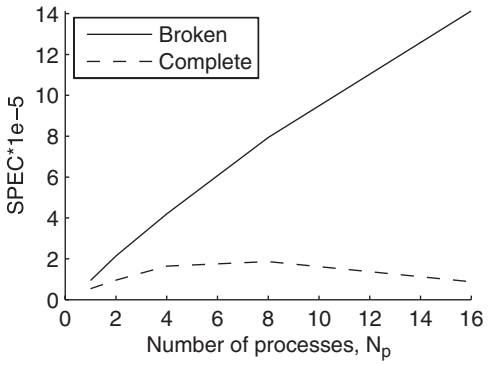

Figure 4. Comparison of the parallel performance of broken and complete lines in the x_3 direction for the ADI sub-iterations. Comparison is for the 2D periodic vortex problem on a 96^3 cell grid, described in detail in Section 6.1.

the broken lines approximation is always $O(1)$ because

$$\begin{aligned} A''' - A'' &= \left[\frac{1}{\Delta t}\left(I-\frac{\Delta t}{2}L_{x_1}\right)\left(I-\frac{\Delta t}{2}L_{x_2}\right)\left(I-\frac{\Delta t}{2}L_{x_3}\right)\right] \\ &\quad -\left[\frac{1}{\Delta t}\left(I-\frac{\Delta t}{2}L_{x_1}\right)\left(I-\frac{\Delta t}{2}L_{x_2}\right)\left(I-\frac{\Delta t}{2}L^{B}_{x_3}\right)\right] \\ &= \left[\frac{1}{\Delta t}\left(I-\frac{\Delta t}{2}L_{x_1}\right)\left(I-\frac{\Delta t}{2}L_{x_2}\right)\left(-\frac{\Delta t}{2}L_{x_3}+\frac{\Delta t}{2}L^{B}_{x_3}\right)\right] \\ &= O(1) \end{aligned} \tag{39}$$

The final approximation to A'' is related to the computation of eddy viscosity and viscosity at each ADI sub-iteration. In LES or in simulations where either the eddy viscosity or the viscosity changes between time levels, Equation (35) requires that the viscosity values at time level $n+1$ be used in evaluating the various operators. However, since the flow field at time level $n+1$ is unknown, the last known approximation to the flow field at time level $n+1$ is used in the computation of the eddy viscosity. In order to assess the temporal order of the error introduced by this approximation, the worst-case scenario of using the viscosity and eddy viscosity values at time level n in the ADI sub-iterations is analyzed. This implies the following definition of A':

$$A' = \frac{1}{\Delta t}\left(I-\frac{\Delta t}{2}L_{x_1}\{;\nu^n,\nu_r^n\}\right)\left(I-\frac{\Delta t}{2}L_{x_2}\{;\nu^n,\nu_r^n\}\right)\left(I-\frac{\Delta t}{2}L^{B}_{x_3}\{;\nu^n,\nu_r^n\}\right) \tag{40}$$

Since the operators such as L_{x_1} that appear in Equation (38) are linear in the viscosity values used, the temporal order of the error of using the viscosity values at time level n in these operators is

$$\begin{aligned} L_{x_1}\{;\nu^{n+1},\nu_r^{n+1}\} - L_{x_1}\{;\nu^n,\nu_r^n\} &= L_{x_1}\{;\nu^n+O(\Delta t),\nu_r^n+O(\Delta t)\} - L_{x_1}\{;\nu^n,\nu_r^n\} \\ &= L_{x_1}\{;\nu^n,\nu_r^n\} + L_{x_1}\{;O(\Delta t),O(\Delta t)\} - L_{x_1}\{;\nu^n,\nu_r^n\} \\ &= O(\Delta t) \end{aligned} \tag{41}$$

DOI: 10.1002/fld

Hence, the temporal order of the error introduced in A' is

$$A'' - A' = O(\Delta t) \tag{42}$$

It should be noted that the viscosity values are also required for the computation of the residual in Equation (26). It can be easily seen that using the last known approximation to the flow at time level $n+1$ for computing the viscosity values introduces only an $O(\Delta t^2)$ error. Since this is of the same order as the desired temporal accuracy, the above approximation to the residual computation is acceptable.

Thus, the temporal order of the error introduced by the above successive approximations is

$$A - A' = O(1) \tag{43}$$

Finally, the number of ADI sub-iterations required to achieve desired second-order temporal accuracy is determined. If a single ADI sub-iteration is performed, then $A - \hat{A} = O(1)$, and two fractional-step iterations are required to reduce the iterative error to $O(\Delta t^2)$. However, if two ADI sub-iterations are performed, then $A - \hat{A}$ becomes $O(\Delta t)$, and only a single fractional-step iteration is required to reduce the iterative error to $O(\Delta t^2)$. All numerical simulations presented here are typically carried out using four processes with two fractional-step iterations and two ADI sub-iterations.

5. PRESSURE POISSON EQUATION

Since the code is being developed for LES of incompressible flows, fourth-order spatial accuracy is desired. But such high-order discretizations result in a very large stencil size for the PPE. In this section, the discretization used for the pressure gradient operator that reduces the stencil size of the PPE without loss in fourth-order spatial accuracy is described.

There are two pressure gradient operators that need to be determined. First is the pressure gradient correcting the velocity, $G_i^W\{p\}$, which appears in the discretized momentum equation, Equation (8). Since fourth-order spatial accuracy is desired, the pressure gradient G_i^W needs to be fourth-order accurate and is chosen as

$$G_i^W\{p\} \equiv \left[\frac{4}{3}\frac{\delta_2 p}{\delta_2 \xi_i} - \frac{1}{3}\frac{\delta_4 p}{\delta_4 \xi_i}\right]\left(\frac{\partial \xi_i}{\partial x_i}\right) \tag{44}$$

Second is the pressure gradient correcting the volume flux, $G_i^{\mathrm{F}}\{p\}$, so that it becomes divergence free, Equation (29). In this sense, the pressure correction $\delta p^{[m]}$ can be thought of as a Lagrange multiplier which constrains the volume flux $\mathbf{F}^{[\mathbf{m}]}$ to be divergence free. G_i^{F} is one of the $O(\Delta t)$ approximations introduced in Equation (21) to invert M easier than P; hence, there is no spatial accuracy requirement on the choice of G_i^{F}. But the choice of G_i^{F} affects the PPE because the Poisson operator is given by $C^{\mathrm{F}}\{\mathbf{G}^{\mathrm{F}}\{\bullet\}\}$ (see Equation (28)). Hence, G_i^{F} is chosen to minimize the stencil size of the Poisson operator.

Since the continuity operator, $C^{\mathrm{F}}\{\}$, must be spatially fourth-order accurate, a 4-point stencil is needed in each direction. If a fourth-order accurate discretization for G_i^{F} were used as well, this would also have a 4-point stencil in each direction. The combination of these two operators would result in a 7-point stencil in each direction for the PPE, which would be costly to invert. The

DOI: 10.1002/fld

size of the stencil can be made smaller, however, since there is no accuracy requirement on G_i^{F}. Nevertheless, the iterations in the iterative fractional-step scheme will converge to the solution of Equations (8)–(10), which is discretized to be spatially fourth-order accurate. Thus, G_i^{F} is chosen to be

$$G_i^{\mathrm{F}}\{p\} \equiv \left[\frac{\delta_1 p}{\delta_1 \xi_i}\right]\left(\frac{\partial \xi_i}{\partial x_i}\right) A_i \tag{45}$$

a spatially second-order accurate discretization of the pressure gradient multiplied by the surface area of the cell face normal to ξ_i. This approximation has a 2-point stencil in each direction, and the resulting PPE has only a 5-point stencil in each direction.

Inversion of the PPE is the computationally expensive step in the fractional-step method. Hence, reducing its stencil size plays an important role in reducing the computational cost. Experiments were performed with several different choices for the two pressure gradient operators; the combination described above was found to work well in terms of maintaining spatial accuracy yet remaining computationally efficient.

6. RESULTS

In this section, results to substantiate the temporal and spatial order of accuracy estimates derived in the previous sections are presented. Results are shown for the following three flows: 2D periodic vortex, Taylor–Green vortex, and LES of temporally evolving mixing layers.

6.1. 2D periodic vortex

The initial conditions for a periodic vortical flow in the x_1x_3 plane are given by

$$\begin{aligned} W_1(0) &= -\cos x_1 \sin x_3 \\ W_2(0) &= 0 \\ W_3(0) &= \cos x_3 \sin x_1 \end{aligned} \tag{46}$$

Since the sub-grid stress terms are of the same form as the viscous terms, the sub-grid viscosity ν_r was set to 0 for code validation. The evolution of this viscous flow is such that the amplitude of the vortex decays exponentially in time. This is one of the few flow problems for which there is a known analytical solution:

$$\begin{aligned} W_1(t) &= \mathrm{e}^{-2\nu t}(-\cos x_1 \sin x_3) \\ W_2(t) &= 0 \\ W_3(t) &= \mathrm{e}^{-2\nu t}(\cos x_3 \sin x_1) \end{aligned} \tag{47}$$

In order to completely validate the difference coding for all three coordinate directions, solutions for a periodic vortex in the x_2x_3 plane were also computed and seen to be identical to those having the vortex in the x_1x_3 plane. Although this problem has certain symmetries, any serious error in the code would have been detected using this test case.

The flow was simulated in a box with edge length 2π and the viscosity, ν, was set to 0.01. This 2D problem was simulated as a 3D problem with periodic BCs in the x_2 direction. Periodic

DOI: 10.1002/fld

BCs were also applied in the x_1 direction. Since the code had the capability to apply non-periodic BCs in the x_3 direction, the implementation of several different types of BCs in this direction was also verified using this test problem. The BC applied in the x_3 direction was one of the following:

1. For the case, bc = per, periodic BCs are enforced in the x_3 direction for all three velocity components.
2. For the case bc = nper1, W_3 and its third partial derivative in the x_3 direction are set to zero:

$$\begin{aligned} \frac{\partial W_1}{\partial x_3} = \frac{\partial^3 W_1}{\partial x_3^3} = 0 \quad \text{at } x_3 = \frac{\pi}{2}, \frac{5\pi}{2} \\ \frac{\partial W_2}{\partial x_3} = \frac{\partial^3 W_2}{\partial x_3^3} = 0 \quad \text{at } x_3 = \frac{\pi}{2}, \frac{5\pi}{2} \\ W_3 = \frac{\partial^3 W_3}{\partial x_3^3} = 0 \quad \text{at } x_3 = \frac{\pi}{2}, \frac{5\pi}{2} \end{aligned} \tag{48}$$

3. For the case bc = nper3, W_3 and its second partial derivative in the x_3 direction are set to zero:

$$\begin{aligned} \frac{\partial W_1}{\partial x_3} = \frac{\partial^3 W_1}{\partial x_3^3} = 0 \quad \text{at } x_3 = \frac{\pi}{2}, \frac{5\pi}{2} \\ \frac{\partial W_2}{\partial x_3} = \frac{\partial^3 W_2}{\partial x_3^3} = 0 \quad \text{at } x_3 = \frac{\pi}{2}, \frac{5\pi}{2} \\ W_3 = \frac{\partial^2 W_3}{\partial x_3^2} = 0 \quad \text{at } x_3 = \frac{\pi}{2}, \frac{5\pi}{2} \end{aligned} \tag{49}$$

First, the dependence of the temporal order of accuracy on the numerical parameters, such as the number of fractional-step iterations nts and number of ADI sub-iterations nadi was studied. The flow was simulated on a 64^3 uniform grid. Non-periodic BCs (bc = nper1) were used in the x_3 direction. A sequence of time steps, $\Delta t = 0.05$, 0.025, 0.0125, and 0.00625 was used to study the temporal accuracy.

Figures 5 and 6 show the variation of volume-averaged kinetic energy at time $t = 10$ with Δt^2. The best-fit lines are also shown to determine the value at $\Delta t = 0$. First, it can be seen that the value at $\Delta t = 0$ is the same for all combinations of numerical parameter values. When the number of ADI sub-iterations nadi is 1, $A - \hat{A} = O(1)$ and two fractional-step iterations are required to achieve second-order temporal accuracy. When nadi = 2, $A - \hat{A} = O(\Delta t)$ and only a single fractional-step iteration is required to achieve second-order temporal accuracy. Further, for this test problem, the errors incurred by using two fractional-step iterations and a single ADI sub-iteration and a single fractional-step iteration and two ADI sub-iterations are comparable. However, the latter strategy requires one fewer pressure Poisson solves, making it the preferred method in terms of computational efficiency. Similar results were observed when periodic BCs were enforced in the x_3 direction as well. Finally, Figure 7 shows the effect of the broken-lines approximation on the temporal convergence when the number of processes is varied between

DOI: 10.1002/fld

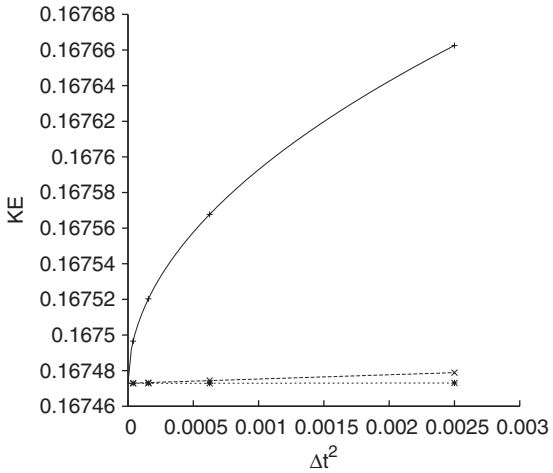

Figure 5. Effect of varying fractional-step iterations nts on the convergence of volume-averaged kinetic energy (KE) with Δt^2 at time $t=10$ while ADI sub-iterations nadi is set to 1. The 2D periodic vortex at $Re=100$ was computed using four processes and non-periodic boundary conditions (bc=nper1) in the x_3 direction. Best-fit curves are also included. +, nts=1; solid line, $0.1675+0.0038(\Delta t)^1$; ×, nts=2; dashed line, $0.1675+0.0024(\Delta t)^2$; ∗, nts=3; dotted line, $0.1675+0.0000(\Delta t)^2$.

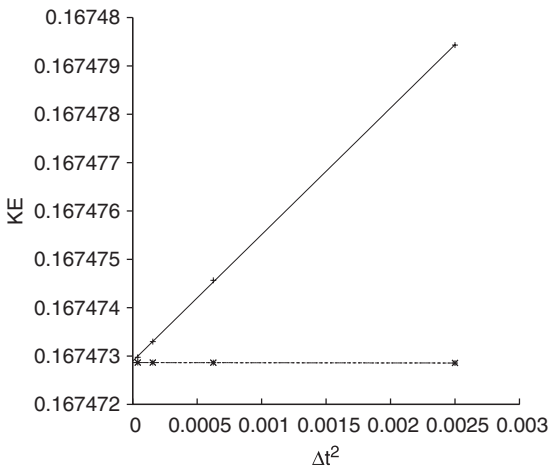

Figure 6. Effect of varying fractional-step iterations nts on the convergence of volume-averaged kinetic energy (KE) with Δt^2 at time $t=10$ while ADI sub-iterations nadi is set to 2. The 2D periodic vortex at $Re=100$ was computed using four processes and non-periodic boundary conditions (bc=nper1) in the x_3 direction. Best-fit curves are also included. +, nts=1; solid line, $0.1675+0.0026(\Delta t)^2$; ×, nts=2; dashed line, $0.1675+0.0000(\Delta t)^2$; ∗, nts=3; dotted line, $0.1675+0.0000(\Delta t)^2$.

DOI: 10.1002/fld

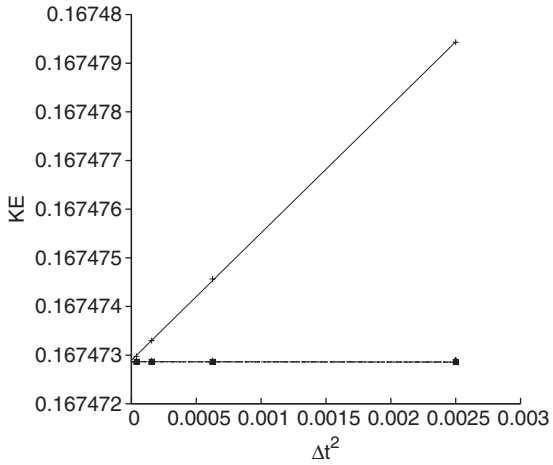

Figure 7. Effect of varying the number of processes np on the convergence of volume-averaged kinetic energy (KE) with Δt^2 at time $t=10$. The 2D periodic vortex at $Re=100$ was computed using non-periodic boundary conditions (bc = nper1) in the x_3 direction and with ADI sub-iterations nadi and fractional-step iterations nts set to 2. For comparison, the difference in convergence observed when the fractional-step iterations nts is reduced to 1 is also included. Best-fit curves are also shown. +, nts = 1; solid line, $0.1675+0.0026(\Delta t)^2$; ×, np = 2; dashed line, $0.1675+0.0000(\Delta t)^2$; ■, np = 4; dash-dot line, $0.1675+0.0000(\Delta t)^2$; •, np = 8; double dash line, $0.1675+0.0000(\Delta t)^2$; △, np = 16; triple dash line, $0.1675+0.0000(\Delta t)^2$.

2 and 16. It can be seen that the error introduced by the broken-line approximation is negligible as compared when the error introduced by reducing the number of fractional-step iterations from 2 to 1. Also, the scheme continues to be temporally second-order accurate irrespective of the number of processes.

Next, a mesh refinement study was carried out to verify the spatial order of accuracy. In these simulations, the cross-derivative viscous terms are neglected and fourth-order spatial accuracy is expected. The neglect of the cross-derivative terms is justified in this case because the physical viscosity is constant in space; therefore, the continuity equation forces their combined contribution to the evolution of momentum to be zero. Figure 8 shows the variation of volume-averaged kinetic energy at time $t=10$ with Δx^2. The flow was simulated on a sequence of uniform grids having 8^3, 16^3, 32^3, and 64^3 grid cells. The time step was set to 0.0025. Figure 8 verifies that the scheme is spatially fourth-order accurate confirming the earlier analysis that the use of spatially second-order discretization for the pressure gradient correcting the volume flux, $G_i^{\mathrm{F}}\{p\}$, does not reduce the spatial order of accuracy of the scheme.

6.2. *Taylor–Green vortex*

Next, the code was used to carry out a discrete numerical simulation (DNS) of the Taylor–Green vortex flow [31], and the results are compared with those from a Fluent simulation of the same flow. This flow is one of the simplest systems in which one can study the generation of small scales

DOI: 10.1002/fld

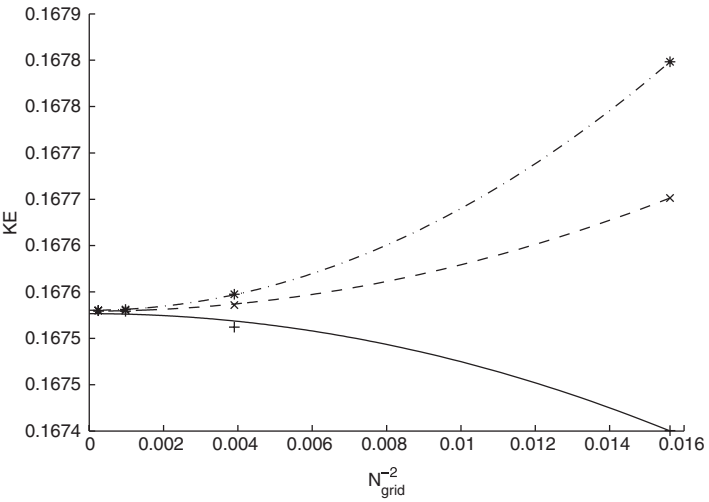

Figure 8. Variation of volume-averaged kinetic energy with Δx^2 at time $t=10$. The 2D periodic vortex problem at $Re=100$ was computed using four processes and various BCs in the x_3 direction. Best-fit curves are also included. +, bc=nper1; solid line, $0.1676-0.5183N_{\text{grid}}^{-4}$; ×, bc=nper3; dashed line, $0.1676+0.4998N_{\text{grid}}^{-4}$; ∗, bc=per; dash-dot line, $0.1676+1.0981N_{\text{grid}}^{-4}$.

and the turbulence resulting from 3D vortex stretching. The initial conditions for the Taylor–Green vortex flow are

$$\begin{aligned} u_1 &= \frac{2}{\sqrt{3}}\sin\left(\frac{2\pi}{3}\right)\sin x_1\cos x_2\cos x_3 \\ u_2 &= \frac{2}{\sqrt{3}}\sin\left(-\frac{2\pi}{3}\right)\cos x_1\sin x_2\cos x_3 \\ u_3 &= 0 \end{aligned} \tag{50}$$

Although the streamlines lie in the x_1x_2 plane at time $t=0$, the flow soon becomes 3D due to the variation in the x_3 direction introduced by the $\cos x_3$ term in the initial conditions.

The Reynolds number for this flow is defined as $Re=1/v$. The DNS of the Taylor–Green flow was carried out for $Re=50$ and 200. The simulations were carried out in a 3D box of edge length, 2π, and a sequence of uniform grids containing 24^3, 32^3, 48^3, 64^3, and 96^3 cells was used to study the variation of the flow evolution with grid spacing. In these simulations, the cross-derivative viscous terms are again neglected because the physical viscosity is constant in space. All simulations were carried out with a constant time step $\Delta t=0.00625$. Periodic BCs were used in all three directions. The time histories of the volume-averaged dissipation are well predicted by the current code. It is noted that the peak of the dissipation rate is not determined correctly when the grid spacing is not small enough to resolve the small-scale flow structures. The Fluent simulations were carried out with the same parameter values and on Cartesian cubic grids with

DOI: 10.1002/fld

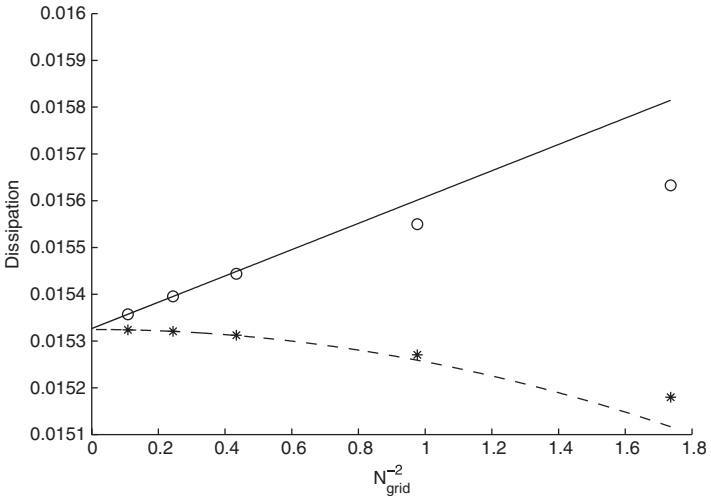

Figure 9. Variation of volume-averaged dissipation rate at time $t=2$ with N_{grid}^{-2} for $Re=50$. N_{grid} is the number of grid points in each coordinate direction. Best-fit curves are also included. $*$, iterative fractional-step scheme; dashed line, best fit for iterative fractional-step with y-intercept$=0.01532461$; $\circ$, fluent simulations; solid line, best fit for Fluent simulations with y-intercept$=0.01532663$.

the same number of grid points even though Fluent has the capability to use unstructured grids. Fluent uses a spatially second-order accurate scheme.

Figure 9 shows the variation of volume-averaged dissipation rate at time $t=2$ with N_{grid}^{-2} for $Re=50$. The time chosen for comparison is close to that of the peak dissipation rate so that the flow has sufficient small-scale structures. Since a uniform grid was used, the number of grid points in each direction N_{grid} is inversely proportional to the grid spacing. The power of 2 was chosen because a spatially second-order scheme (such as the Fluent code) should produce a linear variation of the dissipation, while the current spatially fourth-order scheme should produce a quadratic variation at sufficiently small grid spacings. The curve fits in Figure 9 are drawn using the data points at the smallest two grid spacings. Since the other data points, at least those at smaller grid spacings, lie very close to these curves, this is another verification that the spatial order of accuracy of the current code is 4 and that of Fluent is 2. This figure further allows us to compare the asymptotic value of the volume-averaged dissipation obtained at zero grid spacing by determining the y-intercept. It is noted that the asymptotic values for the two codes are virtually identical, further increasing confidence in the current code. Similar results were observed at $Re=200$.

6.3. LES of temporal mixing layer

In this sub-section, the performance of the current code is studied for the LES of the temporally evolving mixing layer occurring between two fluid streams moving with equal and opposite free-stream speeds U_0. The flow was simulated in a 3D cubic domain $[0, L_1]\times[0, L_2]\times[-L_3/2, L_3/2]$. Periodic BCs are enforced in the x_1 and x_2 directions, while free-slip BCs are enforced in the x_3

DOI: 10.1002/fld

direction:

$$\begin{aligned}
\frac{\partial W_1}{\partial x_3} = \frac{\partial^3 W_1}{\partial x_3^3} = 0 \quad \text{at } x_3 = -\frac{L_3}{2}, \frac{L_3}{2} \\
\frac{\partial W_2}{\partial x_3} = \frac{\partial^3 W_2}{\partial x_3^3} = 0 \quad \text{at } x_3 = -\frac{L_3}{2}, \frac{L_3}{2} \\
W_3 = \frac{\partial^3 W_3}{\partial x_3^3} = 0 \quad \text{at } x_3 = -\frac{L_3}{2}, \frac{L_3}{2}
\end{aligned} \tag{51}$$

Uniform grid spacing was used in the periodic x_1 and x_2 directions, while a stretched Cartesian grid was used in the x_3 direction (since the largest flow gradients occur in the region between the two fluid streams). An algebraic transformation [32, 33] was used to map the non-uniform grid in physical space x_i to the uniform grid in computational space ξ_i:

$$\xi_3 = \frac{x_3}{|x_3| + \dfrac{L_3}{2}} \tag{52}$$

The above maps $x_3 : [-L_3/2, L_3/2] \rightarrow \xi_3 : [-\frac{1}{2}, \frac{1}{2}]$.

The flow is started from the hyperbolic tangent mean velocity profile given by $W_1 = U^0 \tanh(x_3/\delta_w^0)$, where U^0 is the free-stream speed and δ_w^0 is half the initial vorticity thickness defined as

$$\delta_w \equiv \frac{U^0}{(\partial W_3/\partial x_3)\big|_{\max}} \tag{53}$$

The hyperbolic tangent velocity profile has an inflection point, making it susceptible to inviscid instabilities. To initiate turbulence, 2D and 3D eigenfunction perturbations were superimposed on the mean velocity profile [34–36]. To allow two vortex pairings, the length of the computational domain was chosen to be four times the stream-wise wavelength λ_{x_1} of the most unstable eigenfunction from linear stability analysis. The values selected for the various numerical parameters defining the initial conditions are summarized as follows:

$$\begin{aligned}
U^0 &= 1.0 \\
\delta_w^0 &= 1.0 \\
\nu &= 0.02 \\
Re^0 = \frac{U^0 \delta_w^0}{\nu} &= 50 \\
\lambda_{x_1} &= 2.35(2\pi) \\
L_1 = L_2 = L_3 &= 59 \approx 4\lambda_{x_1}
\end{aligned} \tag{54}$$

DOI: 10.1002/fld

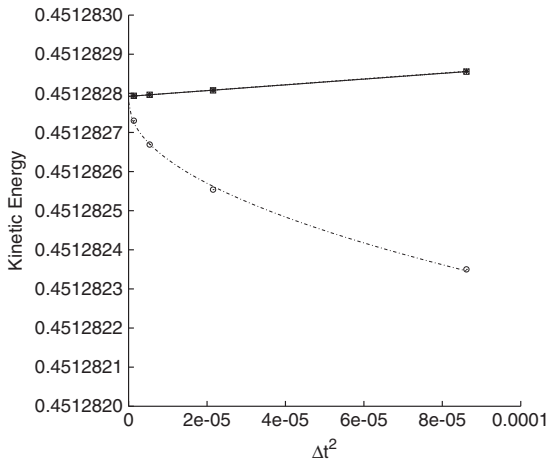

Figure 10. Effect of varying the ADI sub-iterations nadi and fractional-step iterations nts on the temporal accuracy of the LES of mixing layer. Simulations were carried out using a 64^3 non-uniform grid with $\Delta = 1.84375$ and a sequence of time steps: $\Delta t = 0.02$, 0.01, 0.005, and 0.0025. Best-fit curves are also shown. +, nts = 1, nadi = 2; solid line, $0.4513 + 0.0007(\Delta t)^2$; ∘, nts = 1, nadi = 2 without updating the eddy viscosity at each ADI sub-iteration; dash-dot line, $0.4513 - 4.7\text{e}{-5}(\Delta t)^1$; ∗, nts = 2, nadi = 1; dashed line, $0.4513 + 0.0007(\Delta t)^2$; ∗, nts = 2, nadi = 2; dotted line, $0.4513 + 0.0007(\Delta t)^2$.

The LES was carried out using the Smagorinsky sub-grid model [37], in which

$$v_r = (C_S \Delta)^2 (2 S_{ij}^W S_{ij}^W)^{1/2} \tag{55}$$

where C_S is the Smagorinsky constant (here taken as $C_S = 0.17$), and Δ is the filter width.

In Figure 10, the temporal order of accuracy was verified by performing the LES on a sequence of time steps. The eddy viscosity v_r is a function of both space and time and needs to be updated at each ADI sub-iteration to achieve desired second-order temporal accuracy. Again, the errors incurred by using two fractional-step iterations and a single ADI sub-iteration and a single fractional-step iteration and two ADI sub-iterations are comparable, but the latter strategy is preferred in terms of computational efficiency.

7. CONCLUSION

The block-matrix analysis originally developed by Perot [13] is extended to study the order of accuracy of the fractional-step method developed by Bell *et al.* [19] and its iterative variants [20] on a collocated grid layout. The starting point of the analysis is a temporally second-order discretization with explicit treatment of the convective terms using Adams–Bashforth and implicit treatment of the viscous and sub-grid scale stresses using Crank–Nicolson. A collocated grid layout is preferred for ease of extension to curvilinear grids. The capability to solve on stretched Cartesian grids is implemented by mapping the physical grid onto a uniform computational grid.

DOI: 10.1002/fld

The discretization results in a nearly linear system of equations that is hard to solve due to the constraints imposed by the discrete continuity equation. Hence, an iterative procedure is used, in which a linear system with an easier block-LU decomposition as compared with the original system is inverted at each iteration. The analogy between the block-LU decomposition carried out at each iteration and the conventional fractional step is also detailed.

The number of iterations required to reduce the iterative error to the discretion error of $O(\Delta t^2)$ is discussed. In order to improve the numerical efficiency, ADI sub-iterations are used in the velocity step of each iteration. The ADI sub-iteration is further modified to improve parallel performance. The modification approximates the linear system arising in the x_3 direction by neglecting elements that necessitate communication between processes. The iteration error introduced by performing a finite number of fractional-step iterations and ADI sub-iterations is investigated using the block-matrix analysis. The most efficient fractional-step method that achieves second-order temporal accuracy is identified to require two ADI sub-iterations and a single fractional-step iteration.

Energy-conserving discretization is used for the convective terms to improve the stability and accuracy of the numerical scheme. The requirement of energy conservation constrains the continuity equation to a specific spatially fourth-order accurate discretization. This discrete continuity equation combined with a fourth-order discretization for the pressure gradient to correct the volume flux results in a large 7-point stencil in each direction for the PPE. The stencil is reduced to a 5-point stencil in each direction by discretizing the pressure gradient operator using a second-order discretization. It is shown that this change does not affect the spatial order of accuracy of the scheme because the pressure gradient operator used to advance the momentum equations and the discrete continuity equation are still spatially fourth-order accurate. Lastly, for simplicity, the cross-derivative terms in the sub-grid scale stresses are discretized to second-order spatial accuracy.

The order of accuracy estimates presented in this paper are verified in the three flow problems: DNS of 2D periodic vortex, DNS of Taylor–Green vortex, and LES of temporally evolving mixing layer. The 2D periodic vortex calculations confirmed that second-order time accuracy can be achieved by using two fractional-step iterations with a single ADI sub-iteration or a single fractional-step iteration with two ADI sub-iterations. Since the errors in the two cases are comparable, the latter strategy is preferred since it requires one fewer pressure Poisson solves. Fourth-order spatial accuracy is expected in these simulations, as the cross-derivative viscous terms are neglected. A mesh refinement study verified that the scheme is spatially fourth-order accurate and confirmed the earlier analysis that the use of spatially second-order discretization for the pressure gradient correcting the volume flux, $G_i^{\mathrm{F}}\{p\}$, does not reduce the spatial order of accuracy of the scheme. The DNS of the Taylor–Green vortex was also carried out neglecting the cross-derivative terms. Fourth-order spatial accuracy is again confirmed. Both these solutions demonstrated that the approximations introduced in the PPE did not reduce the spatial order of accuracy. Finally, the LES of a temporally evolving mixing layer was simulated using the Smagorinsky model. The desired second-order time accuracy is maintained even though the eddy viscosity is a function of both space and time for this case.

ACKNOWLEDGEMENTS

Initial phases of this research were performed when both authors participated in the 2002 Summer program of the Center of Turbulence Research at NASA Ames Research Center and Stanford University. The authors would also like to thank Prof. Stephen B. Pope for his helpful suggestions and stimulating discussions.

DOI: 10.1002/fld

REFERENCES

1. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
2. Steger JL, Kutler P. Implicit finite-difference procedures for the computation of vortex wakes. *AIAA Paper*, *76-385*, 1976.
3. Choi D, Merkle CL. Application of time-iterative schemes to incompressible flow. *AIAA Journal* 1985; **23**: 1518–1524.
4. Kwak D, Chang JLC, Shanks SP, Chakravarthy SK. A three-dimensional incompressible Navier–Stokes solver using primitive variables. *AIAA Journal* 1986; **24**(3):390–396.
5. Harlow FH, Welch JE. Numerical calculation of time-dependent viscous incompressible flow with free surface. *Physics of Fluids* 1965; **8**(12):2182–2189.
6. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, 1980.
7. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
8. Issa RI, Gossman AD, Watkins AP. The computation of compressible and incompressible recirculating flows by an non-iterative implicit scheme. *Journal of Computational Physics* 1986; **62**:66–82.
9. Yanenko NN. *The Method of Fractional Steps. The Solution of Problems of Mathematical Physics in Several Variables*. Springer: Berlin, 1971.
10. Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *Journal of Computational Physics* 1985; **59**:308–323.
11. Kwak D, Kris C, Dacles-Mariani J, Rogers S, Yoon S. Incompressible Navier–stokes computations in aerospace applications and beyond. In *Frontiers of Computational Fluid Dynamics 1998*, Caughey DA, Hafez MM (eds). World Scientific: Singapore, 1998.
12. Orszag SA, Israeli M, Deville MO. Boundary conditions for incompressible flows. *Journal of Scientific Computing* 1986; **1**(1):75–110.
13. Perot JB. An analysis of the fractional step method. *Journal of Computational Physics* 1993; **108**:51–58.
14. Dukowicz JK, Dvinsky AS. Approximate factorization as a high order splitting for the implicit incompressible flow equations. *Journal of Computational Physics* 1992; **102**:336–347.
15. Karniadakis GE. High-order splitting methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1991; **97**:414–443.
16. Le H, Moin P. An improvement of fractional step methods for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1991; **92**:369–379.
17. Weinan E, Liu J-G. Projection method. II: Godunov–Ryabenki analysis. *SIAM Journal on Numerical Analysis* 1996; **33**(4):1597–1621.
18. Strikwerda JC, Lee YS. The accuracy of the fractional step method. *SIAM Journal on Numerical Analysis* 1999; **37**(1):37–47.
19. Bell JB, Colella P, Glaz HM. A second order projection method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **85**:257.
20. Armfield S, Street R. Note: the fractional-step method for the Navier–Stokes equations on staggered grids: the accuracy of three variations. *Journal of Computational Physics* 1999; **153**:660–665.
21. Pierce CD. Progress-variable approach for large eddy simulation of turbulent combustion. *Ph.D. Thesis*, Stanford University, July 2001.
22. Rhie CM, Chow WL. Numerical study of turbulent flow past an airfoil with trailing edge separation. *AIAA Journal* 1983; **21**:1525–1532.
23. Zang Y, Street RL, Koseff JR. A non-staggered grid fractional step method for time-dependent incompressible Navier–Stokes equations in curvilinear coordinates. *Journal of Computational Physics* 1994; **114**:18–33.
24. Vasilyev OV. High order finite difference schemes on non-uniform meshes with good conservation properties. *Journal of Computational Physics* 2000; **157**:746–761.
25. Vinokur M. Conservation equations of gas-dynamics in curvilinear coordinate systems. *Journal of Computational Physics* 1974; **14**:105–125.
26. Viviand H. Conservative forms of gas dynamic equations. *La Recherche Aerospatiale* 1974; (1974-1):65–68.
27. Tannehill JC, Anderson DA, Pletcher RH. *Computational Fluid Mechanics and Heat Transfer* (2nd edn). Taylor & Francis: London, 1997; 338–341.
28. Fletcher CAJ. *Computational Techniques for Fluid Dynamics*, vol. II, Chapter 12 (2nd edn). Springer-Verlag, 1991; 47–80.

DOI: 10.1002/fld

29. Morinishi Y, Lund TS, Vasilyev OV, Moin P. Fully conservative higher order finite difference schemes for incompressible flow. *Journal of Computational Physics* 1998; **143**:90–124.
30. Golub GH, Van Loan CF. *Matrix Computations* (3rd edn). The Johns Hopkins University Press: Baltimore, MD, 1996; 508–517.
31. Brachet ME, Meiron DI, Orszag SA, Nickel BG, Morf RH, Frisch U. Small-scale structure of the Taylor–Green vortex. *Journal of Fluid Mechanics* 1983; **130**:411–452.
32. Grosch CE, Orszag SA. Numerical solution of problems in unbounded regions: coordinate transforms. *Journal of Computational Physics* 1977; **25**:273–296.
33. Boyd JP. The optimization of convergence for Chebyshev polynomial methods in an unbounded domain. *Journal of Computational Physics* 1981; **45**:43–79.
34. Moser RD, Rogers MM. The three-dimensional evolution of a plane mixing layer: pairing and transition to turbulence. *Journal of Fluid Mechanics* 1993; **247**:275–320.
35. Vreman B, Geurts B, Kuerten H. Large-eddy simulation of the turbulent mixing layer. *Journal of Fluid Mechanics* 1997; **339**:357–390.
36. Gicquel LYM, Givi P, Jaberi FA, Pope SB. Velocity filtered density function for large eddy simulation of turbulent flows. *Physics of Fluids* 2002; **14**(3):1196–1212.
37. Smagorinsky J. General circulation experiments with the primitive equations. *Monthly Weather Review* 1963; **91**:99–164.

DOI: 10.1002/fld