INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2007; **55**:307–345
Published online 14 March 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1435

A modification of the artificial compressibility algorithm with improved convergence characteristics

Frank Muldoon[1] and Sumanta Acharya[2,*,†]

[1]*Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, U.S.A.*
[2]*Department of Mechanical Engineering, Louisiana State University, Baton Rouge, LA 70803, U.S.A.*

SUMMARY

The artificial compressibility algorithm has a significant drawback in the difficulty of choosing the artificial compressibility parameter, improper choice of which leads either to slow convergence or divergence. A simple modification of the equation for pressure in the artificial compressibility algorithm which removes the difficulty of choosing the artificial compressibility parameter is proposed. It is shown that the choice of the relaxation parameters for the new algorithm is relatively straightforward, and that the same values can be used to provide robust convergence for a range of application problems. This new algorithm is easily parallelized making it suitable for computations such as direct numerical simulation (DNS) which require the use of distributed memory machines. Two key benchmark problems are studied in evaluating the new algorithm: DNS of a fully developed turbulent channel flow, and DNS of a driven-cavity flow, using both explicit and implicit time integration schemes. The new algorithm is also validated for a more complex flow configuration of turbulent flow over a backward-facing step, and the computed results are shown to be in good agreement with experimental data and previous DNS work. Copyright © 2007 John Wiley & Sons, Ltd.

Received 17 April 2004; Revised 23 October 2006; Accepted 31 October 2006

KEY WORDS: artificial compressibility; SCGS; improved convergence

1. INTRODUCTION

The pressure–velocity coupling in the solution of the unsteady incompressible Navier–Stokes equations has long been a computationally expensive part of the solution process. The basic problem is that of determining an equation to solve for pressure. Within the framework of finite differences there have historically been four approaches. Each approach (briefly described below) has

*Correspondence to: Sumanta Acharya, Department of Mechanical Engineering, Louisiana State University, Baton Rouge, LA 70803, U.S.A.
†E-mail: acharya@me.lsu.edu

Contract/grant sponsor: United States Department of Energy
Contract/grant sponsor: The Office of Naval Research
Contract/grant sponsor: Louisiana State University
Contract/grant sponsor: Center for Computation and Technology at Louisiana State University

specific advantages and disadvantages. Their use has been dictated by the application at hand, and the preference of the user. In this paper we focus attention on the artificial compressibility and symmetric coupled Gauss Seidel (SCGS) algorithms. For acceptable performance, the artificial compressibility algorithm requires time consuming trial and error adjustment of the artificial compressibility parameter by experienced users. This aspect of the artificial compressibility algorithm is likely the reason why few, if any, commercial packages for solving the Navier–Stokes equations use the algorithm. This trial and error adjustment of the artificial compressibility parameter is particularly expensive for large-scale computations such as direct numerical simulations (DNS). A new algorithm which eliminates the problem of choosing the artificial compressibility parameter is proposed. The goal is to modify the artificial compressibility algorithm so that it does not require time consuming trial and error adjustment of the artificial compressibility parameter while retaining its ease of parallelization.

The first and most common approach for obtaining the pressure field have been methods that involve the solution of a Poisson equation for pressure. Such methods include the fractional step method [1] and the SIMPLE-type methods [2]. In these methods, an elliptic pressure-Poisson equation is derived from the momentum and continuity equations. The solution of the resulting Poisson equation is the greatest computational expense of this class of methods. While mention is made of these methods, the focus of the present work is on improvements to the artificial compressibility algorithm and no further mention is made of them.

The second popular approach for incompressible flow calculations is the method of artificial compressibility [3] which involves the addition of a pseudo time derivative to the equation set. It can be shown to be the result of low Mach number preconditioning of the compressible Navier–Stokes equations as the Mach number goes to zero. It is known that the explicit solution of the compressible Navier–Stokes equations at low Mach numbers is very inefficient [4, 5] due to the increasing stiffness of the equations as the Mach number is decreased. This is a result of the increasing ratio of the speed of sound to that of the velocity. To retain stability, increasingly small time steps are required in order to capture the acoustic waves (i.e. temporal pressure disturbances), the speed of which increases relative to the convection speed as the Mach number decreases. One approach to this problem would be to use an implicit method that eliminates the stability restriction on the time step. The solution of the equations arising from an implicit method would be obtained by an iterative method, direct methods being too expensive. If the time step needed to accurately capture the acoustic waves is of the order of that needed for stability for an explicit scheme, and one was interested in accurately capturing the acoustic waves, then there would be no reason to use an implicit method, which is more computationally expensive per time step. Therefore, in order to be computationally efficient, a time step would be chosen such that the convective terms (which have a substantially larger time scale than the acoustic waves at low Mach numbers) alone are accurately represented. This time step would be too large to accurately resolve the acoustic waves. As a result, the computational method of solving the equations cannot capture the physics concerning the acoustic waves. Each iteration of the iterative method can then be thought of as equivalent to an iteration in pseudo time. Low Mach number preconditioning introduces a pseudo time derivative and reduces the effective speed of sound so that the acoustic waves travel at a velocity close to the convective velocity. The preconditioning clearly must be chosen so that it does not affect the equations when the pseudo time derivative goes to zero. As in an implicit method, a time step will be chosen based on the convective terms. This time step will be much larger than that which would be chosen if the acoustic waves were of interest. Of crucial importance is that the acoustic waves are not in general resolved in an implicit method or in low Mach number preconditioning because

DOI: 10.1002/fld

the physical time step is far too large to resolve them. Another way of looking at this is that as the Mach number goes to zero, the governing equations become the incompressible Navier–Stokes equations which contain no time derivative of pressure. The method of artificial compressibility introduces a finite speed of sound into the incompressible Navier–Stokes equations which have an infinite speed of sound. The artificial compressibility algorithm requires the selection of a parameter which defines the artificially introduced speed of sound. The value of this parameter can vary by four orders of magnitude depending on the flow and the physical time step. Thus, the optimal value of this parameter has to be chosen by trial and error, and, given the large range of values that this parameter can take, several trial solutions must be attempted. Therefore, choosing this parameter can be a very time consuming task, particularly for large-scale computations using large grids.

The third approach is what is known as direct methods. These methods involve a global coupled solution of the entire discretized system in matrix form at one step. If an implicit time integration scheme is used for the convective terms, then iterations must be performed at each time step to solve the resulting non-linear equations. Direct methods are infeasible for large problems due to the computational expense of direct solutions of matrices, which scale as n^3 where n is the matrix dimension, i.e. the number of discrete variables.

The fourth approach is the direct application of an iterative method to the discretized system. There are a wide range of iterative methods suitable for solving the discretized system arising from the Navier–Stokes equations. The present work concerns the SCGS algorithm [6]. This algorithm, which is a local smoother for the Navier–Stokes equations on a staggered grid, has advantages and disadvantages compared to the artificial compressibility algorithm. One important disadvantage is its inherently serial nature. Most solutions to the unsteady incompressible Navier–Stokes equations are extremely computationally expensive in terms of spatial and temporal resolution requirements. For this reason, algorithms that work on vector, and, more importantly, on distributed memory parallel machines are required. This requirement eliminates the original SCGS algorithm due to its inherently serial nature. It should be noted that it is possible to implement a version of the SCGS algorithm which can be parallelized through the use of red–black grid colouring [7]. However, this places a severe restriction on the problem geometry, since arbitrary collections of multi-block grids cannot be coloured in such a fashion. A parallel version of SCGS has also been implemented by Degani and Fox [8], however, their algorithm causes the smoothing properties of the algorithm to be a function of the number of processors used which is undesirable from a standpoint of code verification.

This paper describes an algorithm suitable for parallel computations that combines elements of SCGS and the artificial compressibility algorithm. This work is motivated by the need to improve the artificial compressibility algorithm in order to find a more robust algorithm for the pressure–velocity coupling that is suitable for parallel computations. Comparisons are made between the new algorithm and the artificial compressibility algorithm for two problems using both explicit and implicit time integration schemes. The algorithms are particularly evaluated from the perspective of their suitability for the DNS of flows.

2. GOVERNING EQUATIONS AND DISCRETIZATION

2.1. Governing equations

The governing equations of interest in the present work are the non-conservative unsteady three-dimensional Navier–Stokes equations (Equations (1) and (2)). In the present work, the velocity

DOI: 10.1002/fld

vector is denoted by $\mathbf{u}$, the individual components of which are defined by $\mathbf{u} = (u, v, w)$. The term $\mathbf{B}$ is a body force term which, in the present work, is either zero or a constant.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u} + \mathbf{B} \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{2}$$

2.2. Temporal discretization

In the present work explicit and implicit time integration schemes are used for the convection and diffusion terms and comparisons are made between the results of both such schemes. The current time level (i.e. the time level for which the solution is sought) is denoted by the $n + 1$ superscript. The pressure gradient is always treated implicitly and appears only at the current time level. The explicit scheme used is the third-order accurate multi-level Adams–Bashford scheme, given by Equation (3). However, for one problem, flow over a backward-facing step, a second-order accurate multi-level Adams–Bashford explicit scheme is used, Equation (4)

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla p^{n+1} + \mathbf{B} + \frac{23}{12}\left(\frac{1}{Re}\nabla^2 \mathbf{u}^n - \mathbf{u}^n \cdot \nabla \mathbf{u}^n\right) - \frac{16}{12}\left(\frac{1}{Re}\nabla^2 \mathbf{u}^{n-1} - \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^{n-1}\right) + \frac{5}{12}\left(\frac{1}{Re}\nabla^2 \mathbf{u}^{n-2} - \mathbf{u}^{n-2} \cdot \nabla \mathbf{u}^{n-2}\right) \tag{3}$$

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla p^{n+1} + \mathbf{B} + \frac{3}{2}\left(\frac{1}{Re}\nabla^2 \mathbf{u}^n - \mathbf{u}^n \cdot \nabla \mathbf{u}^n\right) - \frac{1}{2}\left(\frac{1}{Re}\nabla^2 \mathbf{u}^{n-1} - \mathbf{u}^{n-1} \cdot \nabla \mathbf{u}^{n-1}\right) \tag{4}$$

The implicit scheme used is also third-order accurate in time and is given by

$$\frac{11\mathbf{u}^{n+1} - 18\mathbf{u}^n + 9\mathbf{u}^{n-1} - 2\mathbf{u}^{n-2}}{6\Delta t} = -\nabla p^{n+1} + \mathbf{B} + \frac{1}{Re}\nabla^2 \mathbf{u}^{n+1} - \mathbf{u}^{n+1} \cdot \nabla \mathbf{u}^{n+1} \tag{5}$$

2.3. Spatial discretization

Equations (1) and (2) are discretized using the finite difference method on a staggered grid on which each velocity component is defined on a grid shifted one-half cell from the main grid on which pressure is defined, as shown in Figure 1 in two dimensions for clarity. The staggered grid is used to avoid the appearance of spurious modes in the pressure field and is analogous to the use in the finite element method of one order lower shape functions for the pressure than the velocity to satisfy the Babuska–Brezzi inf–sup condition [9, 10]. Each pressure grid point is surrounded by six velocity grid points. For a staggered grid, pressure is not needed or defined on non-periodic boundaries [2]. The pressure gradient and continuity equation are represented by second-order centred schemes. The number of points in all schemes is retained as a non-periodic boundary is approached. This is done by keeping the number of points in the stencil constant while shifting towards the boundary the point at which the derivative is evaluated; this maintains the formal order of accuracy of the stencils near boundaries. For a centred scheme, this results in moving from a centred stencil to one that is biased away from the boundary. Note that for the second

DOI: 10.1002/fld

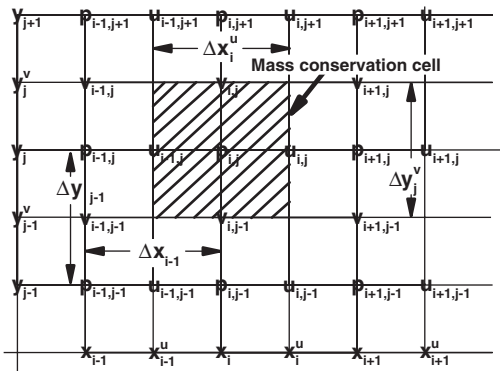

Figure 1. Staggered grid and mass conservation cell (shaded).

derivative centred stencils for the diffusive terms, this biasing reduces the formal order of accuracy by one on an uniform grid. The overall spatial accuracy of the discretization scheme used to solve Equations (1) and (2) is determined by the lowest order finite differences used and is therefore second order accurate. Further details of the finite difference schemes and the computer code used in the present work are given in [7]. All simulations are done in double precision. Different spatial discretization schemes for the convective and diffusive terms are used for the explicit and implicit time integration schemes. In the present work the subscript indices i, j, k refer to the grid points on a three-dimensional finite difference grid.

2.3.1. Spatial discretization of convective and diffusive terms, implicit time integration. The convective terms are discretized using the four point upwind-biased stencil given by Equation (6), where the c's are the finite difference weights which depend on the location of the grid points. On an uniform grid Equation (6) has third order accuracy

$$\left.\frac{\partial u}{\partial x}\right|_{i,j,k} = c^{+x}_{i,-2}u_{i-2,j,k} + c^{+x}_{i,-1}u_{i-1,j,k} + c^{+x}_{i,0}u_{i,j,k} + c^{+x}_{i,+1}u_{i+1,j,k} \quad \text{if } (u_{i,j,k}) \geqslant 0 \tag{6}$$

Due to the staggered grid, each velocity component is stored on a different grid. However, at each velocity grid point the other two velocities are needed in order to compute the terms ${}^{u}v\partial u/\partial y, {}^{u}w\partial u/\partial z, {}^{v}u\partial v/\partial x, {}^{v}w\partial v/\partial z, {}^{w}u\partial w/\partial x, {}^{w}v\partial w/\partial y$, where the prescripts indicate the grid on which the velocities are needed. These velocities (${}^{u}v, {}^{u}w, {}^{v}u, {}^{v}w, {}^{w}u, {}^{w}v$) are obtained by fourth-order accurate two-dimensional Lagrange interpolation (see [7] for details). A five point fourth-order accurate (on an uniform grid) central difference scheme is used for the diffusive terms.

For use in the explanation of the numerical method a detailed description of the spatial discretization is given. In the following, the discretization of the terms in the momentum equations which contain velocity are described by **E** and **G** which are defined over each of the three different velocity grids; i.e. ${}^{1}E, {}^{2}E, {}^{3}E$ are defined on the grids on which u, v, w are defined, respectively. For brevity, only the component of **E** and **G** corresponding to the x momentum equation is shown. All terms which contain velocity in the discretized x momentum equation away from non-periodic boundaries are given by Equation (7). In Equation (7), d^x, d^y, d^z are the finite difference weights of the diffusive scheme in the x, y, z directions, respectively, and $c^{\pm x}$, $c^{\pm y}$, $c^{\pm z}$ are the finite

DOI: 10.1002/fld

difference weights of the upwind (+) and downwind (−) convective schemes in the x, y, z directions, respectively.

2.3.2. Notation. In the present work, square brackets [ ] denote functional dependence, while parenthesis ( ) denote multiplication. That is, $a[b+c]$ means a is a function of $b+c$, while $a(b+c)$ means a times $b+c$

$$
\begin{aligned}
{}^1E_{i,j,k} = 0 = {} & \frac{11u^{n+1}_{i,j,k} - 18u^{n}_{i,j,k} + 9u^{n-1}_{i,j,k} - 2u^{n-2}_{i,j,k}}{6\Delta t} - {}^1B \\
& - d^x_{i,-2}u^{n+1}_{i-2,j,k} - d^x_{i,-1}u^{n+1}_{i-1,j,k} - d^x_{i,0}u^{n+1}_{i,j,k} - d^x_{i,+1}u^{n+1}_{i+1,j,k} - d^x_{i,+2}u^{n+1}_{i+2,j,k} \\
& - d^y_{j,-2}u^{n+1}_{i,j-2,k} - d^y_{j,-1}u^{n+1}_{i,j-1,k} - d^y_{j,0}u^{n+1}_{i,j,k} - d^y_{j,+1}u^{n+1}_{i,j+1,k} - d^y_{j,+2}u^{n+1}_{i,j+2,k} \\
& - d^z_{k,-2}u^{n+1}_{i,j,k-2} - d^z_{k,-1}u^{n+1}_{i,j,k-1} - d^z_{k,0}u^{n+1}_{i,j,k} - d^z_{k,+1}u^{n+1}_{i,j,k+1} - d^z_{k,+2}u^{n+1}_{i,j,k+2} \\
& + \delta^+[u^{n+1}_{i,j,k}](c^{+x}_{i,-2}u^{n+1}_{i-2,j,k} + c^{+x}_{i,-1}u^{n+1}_{i-1,j,k} + c^{+x}_{i,0}u^{n+1}_{i,j,k} + c^{+x}_{i,+1}u^{n+1}_{i+1,j,k}) \\
& + \delta^-[u^{n+1}_{i,j,k}](c^{-x}_{i,-1}u^{n+1}_{i-1,j,k} + c^{-x}_{i,0}u^{n+1}_{i,j,k} + c^{-x}_{i,+1}u^{n+1}_{i+1,j,k} + c^{-x}_{i,+2}u^{n+1}_{i+2,j,k}) \\
& + \delta^+[{}^uv^{n+1}_{i,j,k}](c^{+y}_{j,-2}u^{n+1}_{i,j-2,k} + c^{+y}_{j,-1}u^{n+1}_{i,j-1,k} + c^{+y}_{j,0}u^{n+1}_{i,j,k} + c^{+y}_{j,+1}u^{n+1}_{i,j+1,k}) \\
& + \delta^-[{}^uv^{n+1}_{i,j,k}](c^{-y}_{j,-1}u^{n+1}_{i,j-1,k} + c^{-y}_{j,0}u^{n+1}_{i,j,k} + c^{-y}_{j,+1}u^{n+1}_{i,j+1,k} + c^{-y}_{j,+2}u^{n+1}_{i,j+2,k}) \\
& + \delta^+[{}^uw^{n+1}_{i,j,k}](c^{+z}_{k,-2}u^{n+1}_{i,j,k-2} + c^{+z}_{k,-1}u^{n+1}_{i,j,k-1} + c^{+z}_{k,0}u^{n+1}_{i,j,k} + c^{+z}_{k,+1}u^{n+1}_{i,j,k+1}) \\
& + \delta^-[{}^uw^{n+1}_{i,j,k}](c^{-z}_{k,-1}u^{n+1}_{i,j,k-1} + c^{-z}_{k,0}u^{n+1}_{i,j,k} + c^{-z}_{k,+1}u^{n+1}_{i,j,k+1} + c^{-z}_{k,+2}u^{n+1}_{i,j,k+2}) \qquad (7)
\end{aligned}
$$

where if $(a>0)\delta^+[a]=a$; else; $\delta^+[a]=0$ and if $(a\leqslant 0)\delta^-[a]=a$; else; $\delta^-[a]=0$.

Collecting all terms which multiply the velocity $u^{n+1}_{i,j,k}$ at the grid point where ${}^1E_{i,j,k}$ defined, the coefficient that multiplies $u^{n+1}_{i,j,k}$ can be defined as

$$
\begin{aligned}
{}^1G_{i,j,k} = {} & \frac{11}{6\Delta t} - d^x_{i,0} - d^y_{j,0} - d^z_{k,0} + (\delta^+[u^{n+1}_{i,j,k}]c^{+x}_{i,0} + \delta^-[u^{n+1}_{i,j,k}]c^{-x}_{i,0} \\
& + \delta^+[{}^uv^{n+1}_{i,j,k}]c^{+y}_{j,0} + \delta^-[{}^uv^{n+1}_{i,j,k}]c^{-y}_{j,0} + \delta^+[{}^uw^{n+1}_{i,j,k}]c^{+z}_{k,0} + \delta^-[{}^uw^{n+1}_{i,j,k}]c^{-z}_{k,0})
\end{aligned}
$$

The other two components of **E** and **G** can be obtained in a similar fashion by appropriate permutation of the velocity components and directions.

2.3.3. Spatial discretization of convective and diffusive terms, explicit time integration. A seven point central difference scheme is used for the derivatives in the convective terms; on an uniform grid the scheme has sixth-order accuracy. A monotonic limiter is applied to the convection scheme, for details see [7]. Sixth-order accurate two-dimensional Lagrange interpolation is used to interpolate the velocities $({}^uv, {}^uw, {}^vu, {}^vw, {}^wu, {}^wv)$ needed in computing the terms ${}^uv\partial u/\partial y$, ${}^uw\partial u/\partial z$, ${}^vu\partial v/\partial x$,

DOI: 10.1002/fld

$^{v}w\partial v/\partial z$, $^{w}u\partial w/\partial x$, $^{w}v\partial w/\partial y$. A seven point sixth-order accurate (on an uniform grid) central difference scheme is used for the diffusive terms. For the explicit time integration scheme $\mathbf{E} = \mathbf{u}^{\mathbf{n}+1}\Delta t^{-1} - \mathbf{B} + \Upsilon^{n}$, where Υ^{n} contains the convective and diffusive terms at the previous time steps, and $\mathbf{G} = \Delta t^{-1}$.

With the definitions of the arrays $\mathbf{E}$ and $\mathbf{G}$ another array $\mathbf{F} = \mathbf{E} - \mathbf{G}\mathbf{u}^{n+1}$ can be defined which depends weakly (for implicit time integration) or not at all (for explicit time integration) on the discrete velocity at the $n+1$ time step at the grid point where $\mathbf{F}$ is defined.

3. ARTIFICIAL COMPRESSIBILITY

This method was first proposed by Chorin [3], who used it to solve the steady incompressible Navier–Stokes equations. It has since been used by other researchers [11–19] to solve unsteady flows. It consists of the addition of an artificial time derivative to the momentum and continuity equations. The resulting system of equations is given by

$$\frac{\partial \mathbf{u}}{\partial \tau} + \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2\mathbf{u} + \mathbf{B} \tag{8}$$

$$\frac{\partial p}{\partial \tau} + \beta(\nabla\cdot\mathbf{u}) = 0, \quad \beta>0 \tag{9}$$

These equations are advanced in the artificial time dimension (pseudo time, τ) until the artificial time derivative goes to zero. When this happens the solution of Equations (1) and (2) is recovered. For an unsteady problem this must be done every time step. Because the continuity equation does not contain pressure, it is introduced by defining an artificial equation of state, $p = \rho\beta$ (where β is the artificial compressibility parameter), which results in the definition of an artificial speed of sound $\sqrt{\beta}$. At first sight, it might appear that β is relatively easy to choose. For instance, $\sqrt{\beta}$ could be chosen to be equal to some representative convective velocity as in [15]. It is also possible for $\sqrt{\beta}$ to not be a constant over the entire flow field, but instead to be chosen to be equal to a local velocity [20]. Computational experience has shown [12, 13, 17] that this is not the case, and that β can vary from 1 to 10 000 depending on the flow and the pseudo and physical time steps. It has been shown that for unsteady flows β is strongly a function of the physical time step [11, 17]. If all else is kept constant, as β is increased a smaller pseudo time step must be taken and the ratio of the momentum residuals to the residual of the continuity equation increases. Since time accuracy in pseudo time is of no concern, the time derivatives in Equations (8) and (9) should be discretized using methods chosen to maximize efficiency and robustness. Local time stepping, in which the equation is advanced at different rates in pseudo time depending on spatial position, can be used. This improves the convergence rate by not restricting the global pseudo time step to the minimum required locally.

3.1. Velocity

The momentum equations (Equation (1)) are represented in discretized form by

$$\mathbf{F} + \mathbf{G}\mathbf{u}^{n+1} + \nabla p^{n+1} = 0 \tag{10}$$

In Equation (10), $\mathbf{F}$ contains the convection and diffusion terms, source terms and all parts of the time derivative at previous levels. Note that if the explicit time integration schemes (Equations (3)

DOI: 10.1002/fld

or (4)) are used then $\mathbf{F}$ does not depend on $\mathbf{u}^{n+1}$. The pseudo time derivative term is now added to the momentum equations resulting in

$$\frac{\partial \mathbf{u}}{\partial \tau} + \mathbf{F} + \mathbf{G}\mathbf{u}^{n+1} + \nabla p^{n+1} = 0 \tag{11}$$

This pseudo time derivative term is represented by the first order accurate expression in Equation (12) where the index m indicates the pseudo time level.

$$\frac{\mathbf{u}^{n+1,m+1} - \mathbf{u}^{n+1,m}}{\Delta\tau} + \mathbf{F}^m + \mathbf{G}^m\mathbf{u}^{n+1,m} + \nabla p^{n+1,m} = 0 \tag{12}$$

3.1.1. Local time stepping. Define a local time step $\Delta\tau$ by Equation (13), where α is an under-relaxation factor

$$\Delta\tau = \frac{\alpha}{\mathbf{G}^m} \tag{13}$$

Substituting the above definitions in Equation (12) results in

$$\mathbf{u}^{n+1,m+1} = -(\mathbf{F}^m + \mathbf{G}^m\mathbf{u}^{n+1,m} + \nabla p^{n+1,m})\frac{\alpha}{\mathbf{G}^m} + \mathbf{u}^{n+1,m}$$

Rewriting

$$\mathbf{u}^{n+1,m+1} = -(\mathbf{F}^m + \nabla p^{n+1,m})\frac{\alpha}{\mathbf{G}^m} + (1-\alpha)\mathbf{u}^{n+1,m} \tag{14}$$

It can now be seen that this choice of time step is equivalent to Jacobi iteration with an under-relaxation factor of α. It is local time stepping because the pseudo time step is a function of the velocity and the spatial location. Other choices (such as a constant time step for the entire spatial domain) can be made for the time step. However, the use of a constant time step results in different under-relaxation factors across the spatial domain which is inefficient as some regions will have low under-relaxation factors. Note that while the local time stepping destroys time accuracy, there is no reason to be concerned with accuracy in pseudo time as it has no physical meaning; it is merely a means to obtain a solution. Note that if the convection and diffusion terms are integrated explicitly in physical time, then $\mathbf{G} = \Delta t^{-1}$ and the local time step is not local but is a constant over the entire spatial domain. If the time differencing is implicit then there is sometimes an advantage to freezing the evaluation of the implicit part of the computationally expensive convection and diffusion terms. Sub-cycling (not done in the present work) is then used to solve for pressure and velocity without recomputing the implicit part of the convection and diffusion terms. Note that for the implicit scheme, $\mathbf{G}^m$ in Equation (14) depends on the solution and it may be computationally efficient to perform the computationally expensive division operations required by $\alpha/\mathbf{G}^m$ only at some subset of time steps. In the present work, the under-relaxation parameters for the momentum equations are referred to as α_u, α_v, α_w.

3.2. Pressure

If an explicit first order method is used to discretize Equation (9) in pseudo time, the resulting equation for pressure is given by

$$p^{n+1,m+1} = -\beta(\nabla \cdot \mathbf{u}^{n+1,m+1})\Delta\tau + p^{n+1,m} \tag{15}$$

DOI: 10.1002/fld

As the grid is staggered and therefore the locations of the discrete pressure do not coincide with that of the discrete velocity locations on which the pseudo time step is defined, the pseudo time step in Equation (15) is chosen as the average of the six pseudo time steps in Equation (13) corresponding to the six velocity points surrounding each pressure location. This leaves β to be chosen. The choice of β greatly affects the convergence rate; a too large value of β will result in a non-convergent solution, while a too small value greatly slows convergence. In practice, the difficulty of choosing β is a severe drawback of the method.

4. SCGS

First proposed by Vanka [6], SCGS is an iterative method for solving Equations (1) and (2) on a staggered grid without needing to derive and solve a Poisson equation for the pressure. In the present work the SCGS algorithm is used to solve for the change in velocity and pressure at each iteration. To derive these equations, the discretized version of Equations (1) and (2) can be written in the following general non-linear matrix form:

$$H[\mathbf{q}]\mathbf{q} = s$$

where $\mathbf{q}$ is the vector of all discrete variables (u^{n+1}, v^{n+1}, w^{n+1}, p^{n+1}) over the entire computational domain, H represents the discretized differential operator for the Navier–Stokes equations which in general is a function of $\mathbf{q}$, and s is a source term. The residual R that results from an initial guess $\mathbf{q}^*$ can be written as

$$R = H[\mathbf{q}^*]\mathbf{q}^* - s = H[\mathbf{q}^*][\mathbf{q} - \Delta\mathbf{q}] - s = H[\mathbf{q}^*]\mathbf{q} - s - H[\mathbf{q}^*]\Delta\mathbf{q} \quad \text{where } \Delta\mathbf{q} + \mathbf{q}^* = \mathbf{q}$$

By definition $H[\mathbf{q}]\mathbf{q} - s = 0$. Since the intention is to use the procedure iteratively, the assumption that $H[\mathbf{q}^*] = H[\mathbf{q}]$ can be made. This results in the residual form given below

$$H[\mathbf{q}^*]\Delta\mathbf{q} = -R \tag{16}$$

It is permissible to change $H[\mathbf{q}^*]$ when used in Equation (16). For instance, if $H[\mathbf{q}^*]$ is replaced by $\widetilde{H}[\mathbf{q}^*]$, a zero residual would result in $\Delta\mathbf{q} = 0$ if $\widetilde{H}[\mathbf{q}^*]$ is a linearly independent matrix. Of course there is no guarantee that the sequence of iterates resulting from using $\widetilde{H}[\mathbf{q}^*]$ instead of $H[\mathbf{q}^*]$ will be a convergent sequence. This ability to modify the iteration matrix will be taken advantage of when certain off-diagonal terms are dropped from the matrices that describe the pressure–velocity coupling of some of the algorithms in the present work. Note that if an explicit time integration scheme is used for the convective terms in Equation (1) then H does not depend on $\mathbf{q}$. In this case, no approximation is involved in writing in residual form if all the terms in H are maintained.

The SCGS algorithm involves the coupling of the six momentum equations that surround a pressure location plus the continuity equation at that location. Referring to $p_{i,j}$ in Figure 1 which shows a two-dimensional schematic for simplicity, if an explicit formulation is used for the convective and diffusive terms, the equations for the six velocities surrounding $p_{i,j}$, along with the continuity equation at that same location, can be written in residual form as Equation (18). Note that because the convective and diffusive terms are integrated in time using an explicit scheme, no terms have been dropped or approximated in Equation (18). The residuals of the momentum and continuity

DOI: 10.1002/fld

equations are defined by Equation (17); $\Delta x_i^u, \Delta y_j^v, \Delta z_k^w, \Delta x_i, \Delta y_j, \Delta z_k$ are defined in Figure 1.

$$\begin{aligned}
U_{i,j,k} &= {}^1F_{i,j,k} + {}^1G_{i,j,k}u_{i,j,k}^{n+1} + \frac{p_{i+1,j,k}^{n+1} - p_{i,j,k}^{n+1}}{\Delta x_i} \\
V_{i,j,k} &= {}^2F_{i,j,k} + {}^2G_{i,j,k}u_{i,j,k}^{n+1} + \frac{p_{i,j+1,k}^{n+1} - p_{i,j,k}^{n+1}}{\Delta y_j} \\
W_{i,j,k} &= {}^3F_{i,j,k} + {}^3G_{i,j,k}u_{i,j,k}^{n+1} + \frac{p_{i,j,k+1}^{n+1} - p_{i,j,k}^{n+1}}{\Delta z_k} \\
D_{i,j,k} &= \frac{u_{i,j,k}^{n+1} - u_{i-1,j,k}^{n+1}}{\Delta x_i^u} + \frac{v_{i,j,k}^{n+1} - v_{i,j-1,k}^{n+1}}{\Delta y_j^v} + \frac{w_{i,j,k}^{n+1} - w_{i,j,k-1}^{n+1}}{\Delta z_k^w}
\end{aligned} \tag{17}$$

$$\begin{bmatrix}
{}^1G_{i-1,j,k} & 0 & 0 & 0 & 0 & 0 & \frac{1}{\Delta x_i} \\
0 & {}^1G_{i,j,k} & 0 & 0 & 0 & 0 & \frac{-1}{\Delta x_i} \\
0 & 0 & {}^2G_{i,j-1,k} & 0 & 0 & 0 & \frac{1}{\Delta y_j} \\
0 & 0 & 0 & {}^2G_{i,j,k} & 0 & 0 & \frac{-1}{\Delta y_j} \\
0 & 0 & 0 & 0 & {}^3G_{i,j,k-1} & 0 & \frac{1}{\Delta z_k} \\
0 & 0 & 0 & 0 & 0 & {}^3G_{i,j,k} & \frac{-1}{\Delta z_k} \\
\frac{-1}{\Delta x_i^u} & \frac{1}{\Delta x_i^u} & \frac{-1}{\Delta y_j} & \frac{1}{\Delta y_j} & \frac{-1}{\Delta z_k} & \frac{1}{\Delta z_k} & 0
\end{bmatrix}
\begin{bmatrix}
\Delta u_{i-1,j,k} \\ \Delta u_{i,j,k} \\ \Delta v_{i,j-1,k} \\ \Delta v_{i,j,k} \\ \Delta w_{i,j,k-1} \\ \Delta w_{i,j,k} \\ \Delta p_{i,j,k}
\end{bmatrix}$$

$$= -\begin{bmatrix}
U_{i-1,j,k} \\ U_{i,j,k} \\ V_{i,j-1,k} \\ V_{i,j,k} \\ W_{i,j,k-1} \\ W_{i,j,k} \\ D_{i,j,k}
\end{bmatrix} \tag{18}$$

DOI: 10.1002/fld

Equation (18) can be solved, at each pressure location, for the changes in the six velocities and the change in pressure. However, as one moves from one pressure location to the next pressure location solving the matrix equation, one will have two solutions for each interior velocity. If one solves the matrix equation at main grid point (i, j, k) one will get a value for $\Delta u_{i-1,j,k}$, $\Delta u_{i,j,k}$, $\Delta v_{i,j-1,k}$, $\Delta v_{i,j,k}$, $\Delta w_{i,j,k-1}$, $\Delta w_{i,j,k}$ and $\Delta p_{i,j,k}$. If one then solves the equation at main grid point $(i+1, j, k)$ one will get a value for $\Delta u_{i,j,k}$, $\Delta u_{i+1,j,k}$, $\Delta v_{i+1,j-1,k}$, $\Delta v_{i+1,j,k}$, $\Delta w_{i+1,j,k-1}$, $\Delta w_{i+1,j,k}$ and $\Delta p_{i+1,j,k}$. The result is that one will then have two different values for $\Delta u_{i,j,k}$. All attempts by the authors to combine these two values have been highly unstable. The original SCGS algorithm did not have this problem because, when solving the equation at main grid point $(i+1, j, k)$ in Figure 1, it used the previously computed value of $u_{i,j,k}$ to recompute the residuals. As a result, as one swept across the grid, one solved for each velocity twice but always used or kept the last one. Depending on the direction i, j or k that one sweeps across the grid while solving for the velocities and pressure, one gets a different answer. Unfortunately, SCGS is sensitive to the direction in which sweeps are made. Making a sweep in the direction opposite to the main velocity component can be unstable and presents a problem in recirculating flows. The algorithm does have the advantage of easy-to-choose under-relaxation parameters for the changes in velocity and pressure; the authors have used ~ 0.9 to solve very different flow problems, while others [21, 22] have used values in the range $(0.2 \leftrightarrow 0.75)$. A more serious defect is that it is an inherently serial algorithm. Note that one could use SCGS separately in parallel in each domain as assigned to each processor. One could then use some method of combining the two different solutions obtained for the velocity field at processor domain boundaries. This has not been explored in the present work. One reason is that an instability may occur at processor domain boundaries, similar to the one experienced when trying to combine the two different solutions obtained for the velocity field at all interior points. As the number of processors increases, the number of points at which some special treatment would be needed to combine the two solutions also increases. Indeed, in the limit as the number of processors approaches the number of main grid points, the highly unstable situation of having two different values for $\Delta u_{i,j,k}$ is approached. With this approach, the solution (since the iterative process is a function of the number of processors) and even worse the stability of the iterative process would depend on the number of processors used to obtain it. These are highly undesirable features of any numerical algorithm as it makes it essentially impossible to verify that the algorithm as implemented in software and hardware is correct. Using the code and algorithms in the present work, the authors have verified that it is possible to obtain a solution on one processor that is binary identical to one obtained from using multiple processors in parallel.

However, if we consider using Equation (18) to solve only for pressure we can see that we do not have these problems. The pressure at each main grid point can be uniquely determined in a completely parallel fashion independent of neighbouring pressures. Therefore, one can use SCGS to solve only for pressure and some other method (pseudo time stepping or relaxation) to solve for the velocity. This is the main idea behind the following new algorithm. While the derivation of the SCGS algorithm used an explicit time integration scheme for the convection and diffusion terms, the previous discussion also applies if an implicit time integration scheme is used for these terms.

5. NEW ALGORITHM, SCGS-PP

The idea for symmetric coupled Gauss Seidel parallel pressure (SCGS-PP) comes from combining elements of SCGS and artificial compressibility. The artificial compressibility algorithm is easily

DOI: 10.1002/fld

parallelized but, as noted earlier, contains the hard-to-choose artificial compressibility parameter β. The SCGS algorithm is inherently serial but has easy-to-choose relaxation parameters. The SCGS-PP algorithm is derived by replacing the equation for pressure in the artificial compressibility algorithm with the equation for pressure of the SCGS algorithm while keeping all other aspects of the artificial compressibility algorithm the same. In the resulting algorithm both the solution of the pressure and the velocity can be easily parallelized.

5.1. Explicit time integration of convective and diffusive terms

Solving Equation (18) for $\Delta p_{i,j,k}$ results in

$$\Delta p_{i,j,k} = -\chi_{i,j,k} D_{i,j,k} - \frac{\chi_{i,j,k}}{{}^1G_{i-1,j,k}\Delta x_i^u} U_{i-1,j,k} + \frac{\chi_{i,j,k}}{{}^1G_{i,j,k}\Delta x_i^u} U_{i,j,k} - \frac{\chi_{i,j,k}}{{}^2G_{i,j-1,k}\Delta y_j^v} V_{i,j-1,k}$$
$$+ \frac{\chi_{i,j,k}}{{}^2G_{i,j,k}\Delta y_j^v} V_{i,j,k} - \frac{\chi_{i,j,k}}{{}^3G_{i,j,k-1}\Delta z_k^w} W_{i,j,k-1} + \frac{\chi_{i,j,k}}{{}^3G_{i,j,k}\Delta z_k^w} W_{i,j,k} \tag{19}$$

where

$$\chi_{i,j,k} = \left(\begin{array}{l} \dfrac{1}{{}^1G_{i-1,j,k}\Delta x_i^u \Delta x_i} + \dfrac{1}{{}^1G_{i,j,k}\Delta x_i^u \Delta x_i} + \dfrac{1}{{}^2G_{i,j-1,k}\Delta y_j^v \Delta y_j} \\ \quad + \dfrac{1}{{}^2G_{i,j,k}\Delta y_j^v \Delta y_j} + \dfrac{1}{{}^3G_{i,j,k-1}\Delta z_k^w \Delta z_k} + \dfrac{1}{{}^3G_{i,j,k}\Delta z_k^w \Delta z_k} \end{array} \right)^{-1}$$

The equation for the change in pressure leads immediately to an equation for pressure, Equation (20). In practice it is frequently necessary to apply an under-relaxation parameter (α_p) to $\Delta p_{i,j,k}$ in order to achieve convergence; this is discussed in the Results section. The artificial compressibility algorithm is now modified by replacing just the equation for pressure (Equation (15)) with Equation (20).

$$p_{i,j,k}^{n+1,m+1} = \alpha_p \left(\begin{array}{l} -\chi_{i,j,k} D_{i,j,k} - \dfrac{\chi_{i,j,k}}{{}^1G_{i-1,j,k}\Delta x_i^u} U_{i-1,j,k} + \dfrac{\chi_{i,j,k}}{{}^1G_{i,j,k}\Delta x_i^u} U_{i,j,k} - \dfrac{\chi_{i,j,k}}{{}^2G_{i,j-1,k}\Delta y_j^v} V_{i,j-1,k} \\ \quad + \dfrac{\chi_{i,j,k}}{{}^2G_{i,j,k}\Delta y_j^v} V_{i,j,k} - \dfrac{\chi_{i,j,k}}{{}^3G_{i,j,k-1}\Delta z_k^w} W_{i,j,k-1} + \dfrac{\chi_{i,j,k}}{{}^3G_{i,j,k}\Delta z_k^w} W_{i,j,k} \end{array} \right)$$
$$+ p_{i,j,k}^{n+1,m} \tag{20}$$

5.2. Implicit time integration of convective and diffusive terms

As a result of the convective and diffusive terms, additional elements (indicated by $*$) on the off-diagonals arise in the matrix (Equation (21)) which describe the coupling between the six velocity points and pressure. Note that if implicit time integration of the convective terms is not used, then the additional elements depend only on the grid and time step. Otherwise, the additional elements

DOI: 10.1002/fld

will also depend on the solution, as a result of the non-linearity of the convective term.

$$\begin{bmatrix} {}^1G_{i-1,j,k} & * & 0 & 0 & 0 & 0 & \frac{1}{\Delta x_i} \\ * & {}^1G_{i,j,k} & 0 & 0 & 0 & 0 & \frac{-1}{\Delta x_i} \\ 0 & 0 & {}^2G_{i,j-1,k} & * & 0 & 0 & \frac{1}{\Delta y_j} \\ 0 & 0 & * & {}^2G_{i,j,k} & 0 & 0 & \frac{-1}{\Delta y_j} \\ 0 & 0 & 0 & 0 & {}^3G_{i,j,k-1} & * & \frac{1}{\Delta z_k} \\ 0 & 0 & 0 & 0 & * & {}^3G_{i,j,k} & \frac{-1}{\Delta z_k} \\ \frac{-1}{\Delta x_i^u} & \frac{1}{\Delta x_i^u} & \frac{-1}{\Delta y_j} & \frac{1}{\Delta y_j} & \frac{-1}{\Delta z_k} & \frac{1}{\Delta z_k} & 0 \end{bmatrix} \begin{bmatrix} \Delta u_{i-1,j,k} \\ \Delta u_{i,j,k} \\ \Delta v_{i,j-1,k} \\ \Delta v_{i,j,k} \\ \Delta w_{i,j,k-1} \\ \Delta w_{i,j,k} \\ \Delta p_{i,j,k} \end{bmatrix}$$

$$= - \begin{bmatrix} U_{i-1,j,k} \\ U_{i,j,k} \\ V_{i,j-1,k} \\ V_{i,j,k} \\ W_{i,j,k-1} \\ W_{i,j,k} \\ D_{i,j,k} \end{bmatrix} \tag{21}$$

These additional terms make it cumbersome to obtain an analytical expression for $\Delta p_{i,j,k}$ as was done for the case of the explicit convection and diffusion scheme. However, an expression for $\Delta p_{i,j,k}$ can be obtained by using the appropriate row of the inverted matrix in Equation (21) to form a linear combination of the residuals. Once this matrix inversion has been done, the amount of computational work needed to determine $\Delta p_{i,j,k}$ is the same as for the case of explicit convection and diffusion. Note that as the general procedure involves solving for the changes in the velocity and pressure, it may be possible to change the matrix in Equation (21) and still obtain a solution. In particular the inclusion of the additional elements on the off-diagonals arising from the convection and diffusion schemes is found to make little difference in the rate of convergence and hence these elements are not used in the present work. For the SCGS algorithm, this unimportance of the off-diagonals has been reported by others [21, 22]. When these off-diagonals terms are dropped the left-hand side of Equation (21) differs from Equation (18) only in the centre coefficients $({}^1G_{i-1,j,k}, {}^1G_{i,j,k}, {}^2G_{i,j-1,k}, {}^2G_{i,j,k}, {}^3G_{i,j,k-1}, {}^3G_{i,j,k})$, which in Equation (21) are a function of

DOI: 10.1002/fld

Table I. Side-by-side comparison of algorithms.

Artificial compressibility	SCGS-PP
Solve for velocity	
$\mathbf{u}^{n+1,m+1} = (\mathbf{F}^m - \nabla p^{n+1,m})\frac{\alpha}{\mathbf{G}^m} + (1-\alpha)\mathbf{u}^{n+1,m}$	
Solve for pressure	
$p_{i,j,k}^{n+1,m+1} = -\beta D_{i,j,k}^{n+1,m+1}\Delta\tau + p_{i,j,k}^{n+1,m}$	$p_{i,j,k}^{n+1,m+1} = \alpha_p \begin{pmatrix} \chi_{i,j,k} D_{i,j,k}^{n+1,m+1} \\ -\phi_{i-1,j,k} U_{i-1,j,k}(\mathbf{u}^{n+1,m+1}, p^{n+1,m}) \\ -\phi_{i,j,k} U_{i,j,k}(\mathbf{u}^{n+1,m+1}, p^{n+1,m}) \\ -\gamma_{i,j-1,k} V_{i,j-1,k}(\mathbf{u}^{n+1,m+1}, p^{n+1,m}) \\ -\gamma_{i,j,k} V_{i,j,k}(\mathbf{u}^{n+1,m+1}, p^{n+1,m}) \\ -\lambda_{i,j,k-1} W_{i,j,k-1}(\mathbf{u}^{n+1,m+1}, p^{n+1,m}) \\ -\lambda_{i,j,k} W_{i,j,k}(\mathbf{u}^{n+1,m+1}, p^{n+1,m}) \end{pmatrix} + p_{i,j,k}^{n+1,m}$
Repeat until convergence	

the solution, the grid and Δt, while in Equation (18) they equal Δt^{-1}. This ability to modify the matrix can be used for computational efficiency if an implicit convection scheme is used. In this case, even if the off-diagonal terms are dropped, the matrix will depend on the solution. However, it has been observed that it is not necessary to perform the matrix inversion at every iteration. It is more efficient computationally to perform the matrix inversion only at certain time steps and obtain a representative inverted matrix.

A comparison of the two algorithms is given in Table I. Note that the momentum equation residuals in the SCGS-PP algorithm depend on $p^{n+1,m}$ and not $p^{n+1,m+1}$. As a result the algorithm can be parallelized easily. The new algorithm has replaced the problem of choosing β with that of choosing α_p. It will be shown that it is much easier to choose α_p and that in addition the new algorithm has better convergence properties.

6. FOURIER MODE ANALYSIS

An important method of evaluating the properties of an iterative scheme is the Fourier mode analysis. The basic idea is to represent the error as a Fourier series and examine the effect of the iterative scheme in Fourier (frequency) space.

Any stationary iterative scheme to solve $Au = b$ can be written as

$$C(u^{m+1} - u^m) = b - Au^m \quad \text{or} \quad u^{m+1} = C^{-1}(b + (C - A)u^m) = C^{-1}b + (I - C^{-1}A)u^m$$

where m is the iteration count.

DOI: 10.1002/fld

Defining the current approximation to the solution u^m as $u^m = \varepsilon^m + u$, where ε^m is the error, u is the exact solution, and substituting results in

$$\varepsilon^{m+1} + u = C^{-1}b + (I - C^{-1}A)(\varepsilon^m + u) = C^{-1}b + I\varepsilon^m - C^{-1}A\varepsilon^m + Iu - C^{-1}Au$$
$$= C^{-1}(b - Au) + (I - C^{-1}A)\varepsilon^m + Iu$$

As $Au = b$ this reduces to

$$\varepsilon^{m+1} = (I - C^{-1}A)\varepsilon^m$$

Therefore, the behaviour of the error is a function only of the matrix $(I - C^{-1}A)$ and does not depend on the right-hand side vector b. Any grid function (e.g. the solution $u_{i,j,k}$ or the error $\varepsilon_{i,j,k}$) can be written as [23]

$$u_{i,j,k} = \sum_{k_z=1}^{N_z-1} \sum_{k_y=1}^{N_y-1} \sum_{k_x=1}^{N_x-1} a_{k_x,k_y,k_z} \sin\left(\frac{i\pi k_x}{N_x}\right) \sin\left(\frac{j\pi k_y}{N_y}\right) \sin\left(\frac{k\pi k_z}{N_z}\right) \tag{22}$$

where

$$a_{k_x,k_y,k_z} = \sum_{k=1}^{N_z-1} \sum_{j=1}^{N_y-1} \sum_{i=1}^{N_x-1} u_{i,j,k} \sin\left(\frac{i\pi k_x}{N_x}\right) \sin\left(\frac{j\pi k_y}{N_y}\right) \sin\left(\frac{k\pi k_z}{N_z}\right) \tag{23}$$

An analysis of how iterative schemes affect the Fourier spectrum can now be made by comparing the resulting Fourier coefficients of the error with those of the original error. Of particular interest is the ratio of the magnitude of the resulting Fourier coefficients to the original Fourier coefficients, at each frequency or wave number. This is known as the smoothing factor and is given by

$$\rho^m_{k_x,k_y,k_z} = \left| \frac{a^m_{k_x,k_y,k_z}}{a^{\text{original}}_{k_x,k_y,k_z}} \right| \quad \text{where } m \in [1, \infty) \text{ is the iteration index} \tag{24}$$

6.1. Model problem

A model problem is constructed with Dirichlet boundary conditions of $\mathbf{u} = 0$ on the boundary Γ. No boundary conditions are needed for pressure. The initial conditions at the previous time steps are $\mathbf{u} = 0$. Pressure is treated fully implicitly and therefore no initial conditions are needed for it. The solution to this problem is $\mathbf{u} = 0$ in the interior and pressure equal to a constant. As a result, the intermediate solution at any point in the iterative process is the error. Since there is no error on Γ, the error is defined only in the interior. As regards the discrete grid this means that the error is defined from the first to the last grid point in each coordinate direction excluding those points that lie on Γ. An error field is generated by the use of Equation (22) with $a_{k_x,k_y,k_z} = 1$ for the velocity and pressure. This error is imposed as an initial guess at the current time step and the iterative scheme is applied a number of times. The resulting velocity and pressure field (which is the error field) is then transformed to Fourier space by the use of Equation (23). As the pressure can only be determined up to a constant, it must be treated slightly differently from the velocity. After using Equation (22) with $a_{k_x,k_y,k_z} = 1$ to generate a pressure field, a constant is added to the field such that the pressure at one point is zero. The resulting field is then transformed to Fourier

DOI: 10.1002/fld

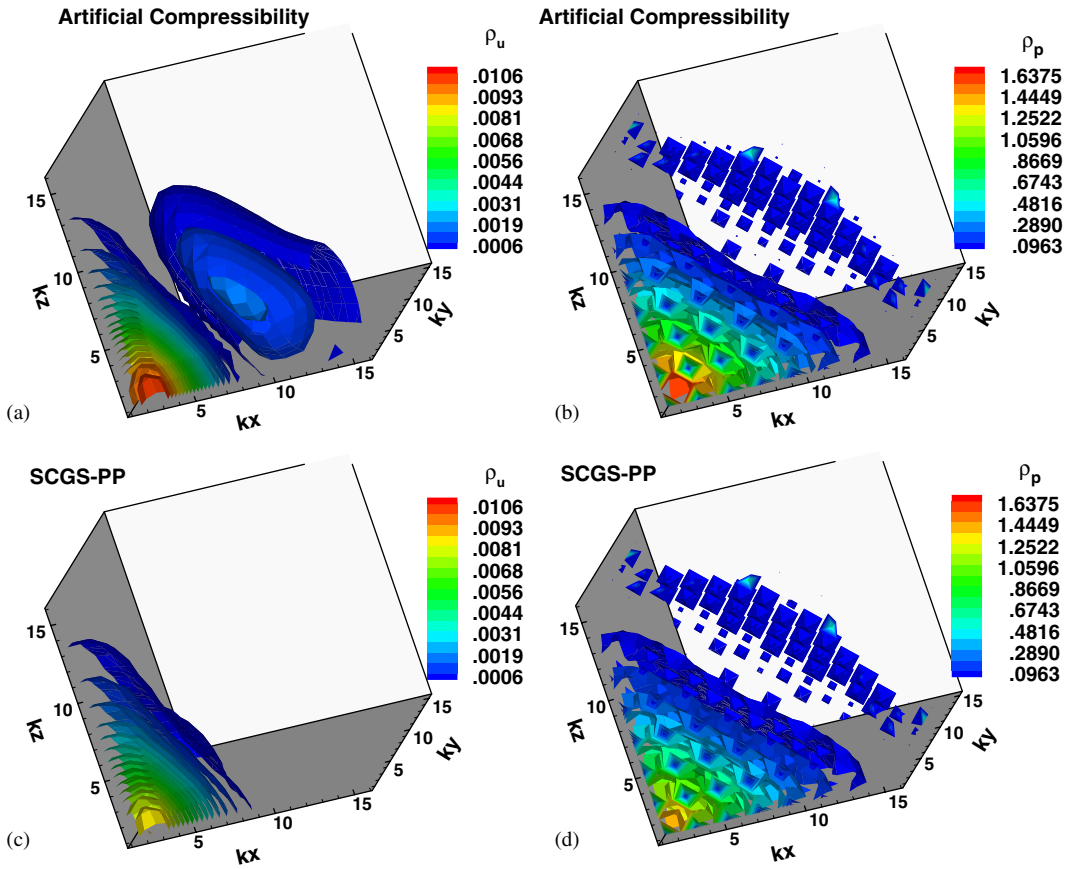

Figure 2. Isosurfaces of ρ_u and ρ_p of model problem on a $16 \times 16 \times 16$ grid.

space by the use of Equation (23) and stored. After applying the iterative algorithm, the resulting pressure field is adjusted by a constant so that the pressure at the same point is zero. This pressure field is then transformed to Fourier space at which point Equation (24) can be used.

6.2. *Model problem results*

Figure 2 shows isosurfaces of the smoothing factor for u and p as a function of wave number space for the algorithms. As expected for methods that involve local smoothing, both algorithms show the highest value of the smoothing factor (Equation (24)) at low wave numbers (near the origin in wave number space). The artificial compressibility algorithm has higher smoothing factors for both u and p compared to the SCGS-PP algorithm. Figure 3 shows the maximum smoothing factors as a function of iteration number. The maximum smoothing factors of both u and p are smaller with SCGS-PP than artificial compressibility. Figure 4 shows the residual as a function of iteration number, with both the average and the maximum residuals decaying faster with SCGS-PP than with artificial compressibility.

DOI: 10.1002/fld

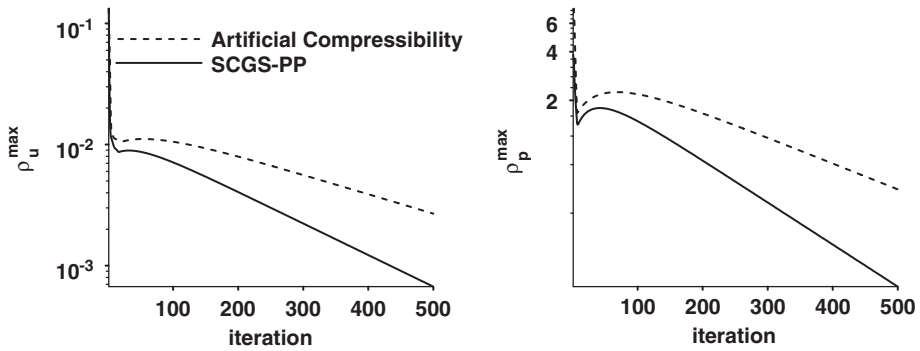

Figure 3. Maximum smoothing factors of model problem on a $16 \times 16 \times 16$ grid.

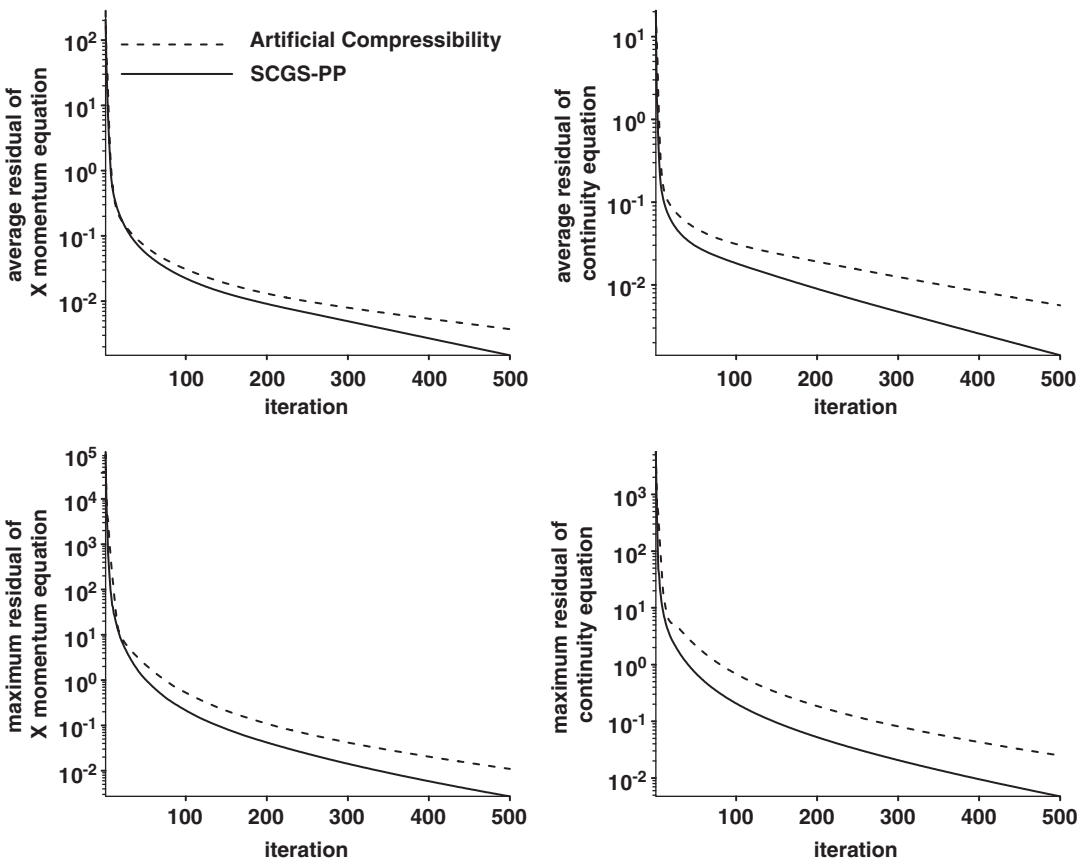

Figure 4. Residual history of model problem on a $16 \times 16 \times 16$ grid.

DOI: 10.1002/fld

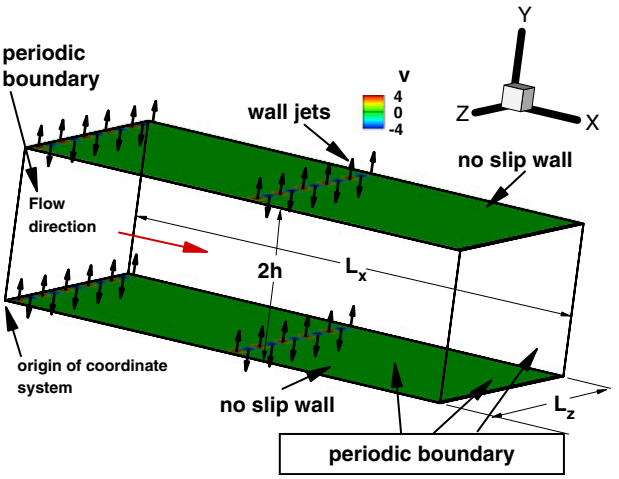

Figure 5. Channel flow schematic showing wall jets used to trigger turbulence at the start of the simulation. Wall jets are shut off after flow has travelled 12.4h based on the bulk velocity and the channel half height h.

7. RESULTS

In this section, the artificial compressibility and SCGS-PP algorithms are evaluated by applying them to the DNS of a channel and a driven-cavity flow. In addition, the SCGS-PP algorithm is used for the DNS of flow over a backward-facing step. Note that the residual of the continuity equation is defined by the last equation of Equation (17).

7.1. DNS of channel flow

7.1.1. Problem description. A channel flow at a Reynolds number of 180 is simulated. The flow is driven by a constant body force term which is added to the x momentum equation. All quantities are non-dimensionalized by the channel half height h and the friction velocity $u_\tau = \sqrt{\nu \partial \overline{u}/\partial y|_{y=0}}$. A schematic of the flow domain along with boundary conditions and dimensions is given in Figure 5. The computational domain is 6.4 units in the streamwise direction (x), 2 units in the wall normal direction and 3.2 units in the cross-stream direction (z). The grid dimensions are $128 \times 128 \times 128$. Evenly spaced grids are used in the x and z directions, with grid spacings in wall units of 9 and 4.5, respectively. A stretched grid that concentrates points near the two walls is used in the y direction. The first three main grid points in wall units are at 0.716, 2.158 and 3.671 from the wall in the y direction. The maximum grid spacing in wall units in y is 5.04 (at the channel centre line). It has been shown [24] that this resolution is sufficient for this Reynolds number. It is necessary to use a method to trigger turbulence for this particular flow. The method used in the present work is to place two rows of wall jets on the top and bottom walls, see Figure 5. These wall jets either injected or removed fluid from the domain depending on their spatial location. The net sum of mass injected into the domain was approximately zero. These wall jets were turned on until the flow (based on the bulk velocity) had travelled 12.4h (2232 wall units), after which they were turned

DOI: 10.1002/fld

Table II. Channel flow time step and numerical parameters, explicit time integration scheme.

Case	Time step and numerical parameters	Notes
Case 1	Artificial compressibility, $\Delta t = 0.00005,\ \beta = 4000,\ \alpha_u = \alpha_v = \alpha_w = 0.915$	Optimal value of β
Case 2	Artificial compressibility, $\Delta t = 0.00005,\ \beta = 400,\ \alpha_u = \alpha_v = \alpha_w = 0.915$	Non-optimal value of β
Case 3	Artificial compressibility, $\Delta t = 0.00015,\ \beta = 400,\ \alpha_u = \alpha_v = \alpha_w = 0.915$	Optimal value of β
Case 4	SCGS-PP, $\Delta t = 0.00005,\ \alpha_p = \alpha_u = \alpha_v = \alpha_w = 0.925$	
Case 5	SCGS-PP, $\Delta t = 0.00015,\ \alpha_p = \alpha_u = \alpha_v = \alpha_w = 0.925$	

off. The calculations then proceeded until the flow (based on the bulk velocity) travelled a further $14.6h$ (4860 wall units) to allow the transients caused by the jets to decay and the turbulence to become fully developed. After this the simulation was advanced in time until the flow (based on the bulk velocity) has travelled a further $108h$ (19 440 wall units). In addition, as an initial condition, the u velocity is set to the bulk velocity, the v velocity at each interior point is set to a random number between -2 and 2 (to aid in triggering turbulence) and the w velocity is set to zero.

7.1.2. Explicit time integration scheme results. Two different time steps ($\Delta t = 0.00015$ and 0.00005) are used to demonstrate the effect of the time step on the various numerical parameters that are required for the artificial compressibility and SCGS-PP algorithms. Both of these time steps are well within the range needed to accurately resolve the turbulent fluctuations of this particular flow [25, 26]. For this flow the time step of 0.00015 corresponds to a maximum CFL number of approximately 0.0645 based on the u velocity and grid spacing in the x direction; the CFL numbers based on the other two directions being smaller. Table II shows the numerical parameters used for both algorithms. In Cases 1 and 3 the value of β required by the artificial compressibility algorithm was chosen using a trial and error method in which β was continually increased until an instability sets in, causing the solution to diverge. At that point β was decreased somewhat and this value, which is considered to be optimal, was used for the simulation. The consequences of not choosing an optimal value of β are shown later; one of the consequences is that the solution diverges. In general, the simulation had to be integrated for a few hundred physical time steps in order to determine whether the solution was diverging. The value of β found is considered optimal, considering the constraints of the author's time involved in performing the trial and error simulations, approximately 10 of which were done in order to find each optimal value of β. The choice of the parameters $\alpha_p, \alpha_u, \alpha_v, \alpha_w$ required by the SCGS-PP algorithm was made simply by using values that have been used by the authors in solving other flows, such as jets in cross-flow and flow over spheres and cylinders. Case 2 is included to demonstrate the effect of using a value of β that is optimal for a time step of $\Delta t = 0.00015$ for a simulation for which a smaller time step of $\Delta t = 0.00005$ is used. A different value of $\beta = 4000$, is optimal for this lower time step of $\Delta t = 0.00005$. Using $\beta = 4000$ for a simulation with a larger time step of $\Delta t = 0.00015$, results in a diverging solution. Ten iterations were used for a time step of $\Delta t = 0.00005$ and 20 for a time step of $\Delta t = 0.00015$.

Residual level: Figure 6 shows the history as a function of time of the average residual of the continuity equation at a time step of $\Delta t = 0.00005$. This is defined by the arithmetic mean of the absolute value of the residual of the continuity equation at every pressure grid point. It is observed that for the artificial compressibility algorithm, using the value of β that is optimal for a time step of $\Delta t = 0.00015$ for a simulation using a smaller time step of $\Delta t = 0.00005$ results in a residual

DOI: 10.1002/fld

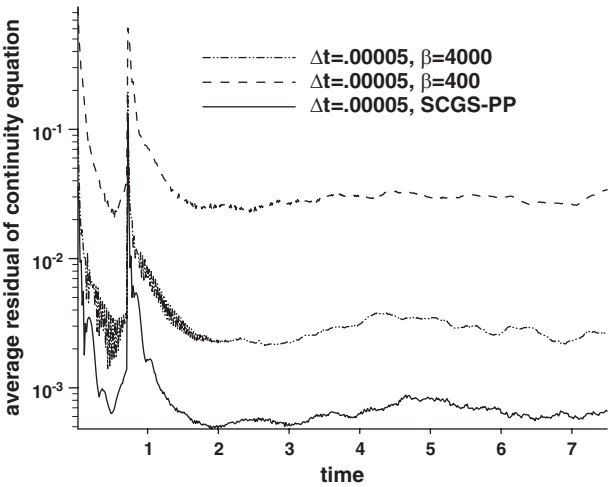

Figure 6. Average residual of the continuity equation, channel flow, explicit time integration scheme (Equation (3)), $\Delta t = 0.00005$.

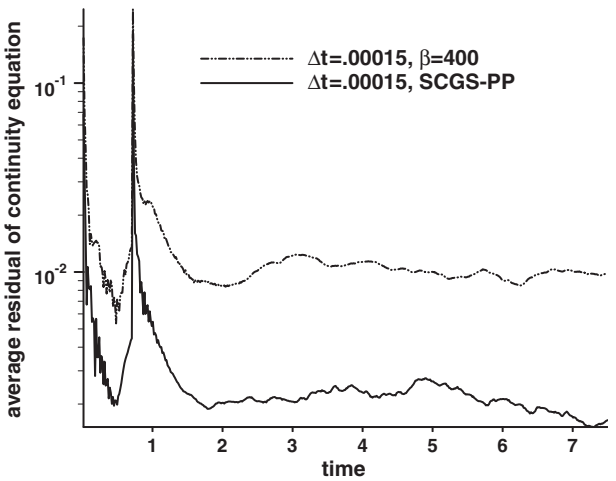

Figure 7. Average residual of the continuity equation, channel flow, explicit time integration scheme (Equation (3)), $\Delta t = 0.00015$.

level that is an order of magnitude higher than that obtained using the optimal value of β for the smaller time step. In addition, even if using the optimum value of β, the value of the residual for the artificial compressibility algorithm is approximately six times larger than that resulting from the SCGS-PP algorithm. The spike in the graph is the result of turning off the wall jets used to trigger the transition to turbulence. The same trends exist for the time step of $\Delta t = 0.00015$, as can be seen in Figure 7.

Other measures of error: The instantaneous value of v integrated over the entire xz plane at any location in the y direction is zero as a result of conservation of mass. Figure 8 shows the time

DOI: 10.1002/fld

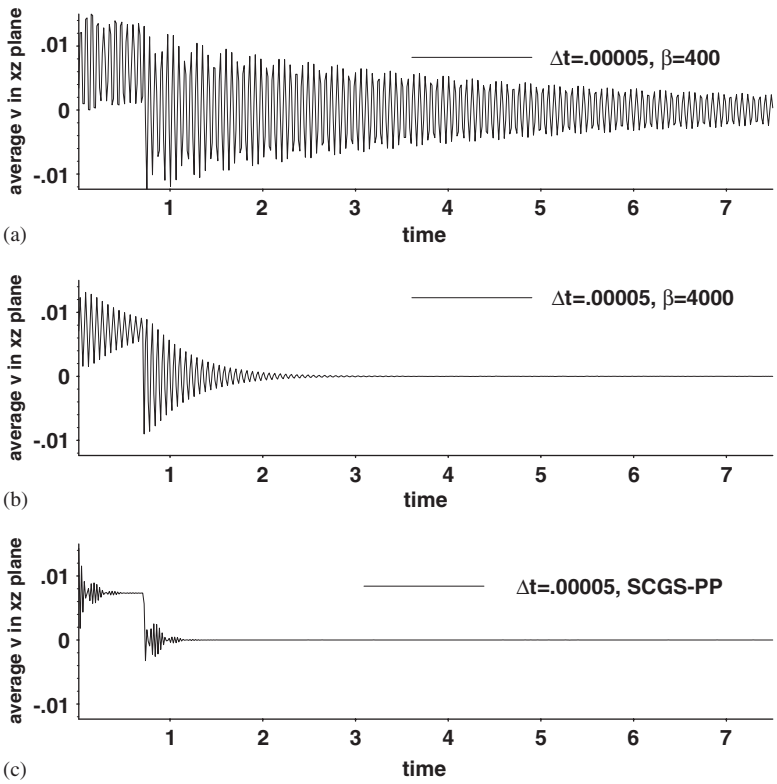

Figure 8. Spatially averaged v in xz plane, channel flow, explicit time integration scheme (Equation (3)), $\Delta t = 0.00005$.

history of this quantity (from an xz plane at $y = h$). The value of v integrated over the entire xz plane is positive in the beginning of the simulations because the net mass flow into the domain, due to the wall jets, is not exactly zero. The sudden changes in Figure 9 are the result of turning off the wall jets used to trigger turbulence, after which the value quickly begins to oscillate around the correct value of zero. It can be seen that the errors in this quantity take a significantly longer time to decay for the artificial compressibility algorithm, particularly if a non-optimized value of β is used. Note that the artificial compressibility algorithm, even when using the optimal value of β, performs significantly worse than the SCGS-PP algorithm.

Statistics: In order to characterize turbulent flows it is necessary to collect statistics that are the result of time averaging various quantities. As the flow field is homogenous in the x and z directions, spatial averaging is carried out in both these directions. Statistics are collected at each time step after the flow (based on the bulk velocity) has travelled $27h$ (4860 wall units), at which point the transients caused by the jets have decayed and the turbulence has become fully developed. After this point, statistics are collected over a length of time in which the flow (based on the bulk velocity) travels $108h$ (19 440 wall units). Figure 10 compares the only (for this flow) non-zero non-normal turbulent Reynold's stress of the present work with that of [27, 28]. The results of the present work compare quite well. Due to symmetries, there are certain statistical quantities which are zero for this flow. For instance, the mean value of the velocity v in the wall normal direction

DOI: 10.1002/fld

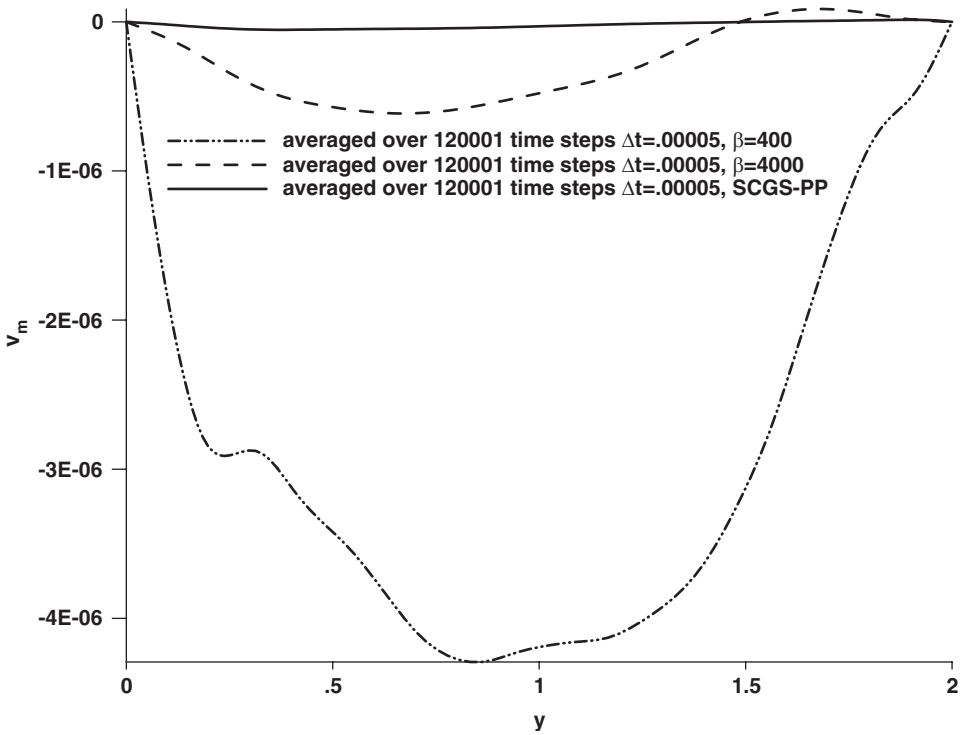

Figure 9. Temporally and spatially averaged v, channel flow, explicit time integration scheme (Equation (3)), $\Delta t = 0.00005$.

should be zero. Figure 9 shows the mean v at a time step of $\Delta t = 0.00005$. It can be seen that the artificial compressibility algorithm in general results in larger errors. The use of a non-optimized value of β results in significantly larger errors.

7.1.3. Implicit time integration scheme results. Two different time steps, $\Delta t = 0.0015$ and 0.0005, are used to demonstrate the effect of the time step on the numerical parameters that are required for the algorithms. Both of these time steps are within the range needed to accurately resolve the turbulent fluctuations of this particular flow [25, 26]. For this flow the time step of 0.0015 corresponds to a maximum CFL number of approximately 0.64 based on the u velocity and grid spacing in the x direction; the CFL numbers based on the other two directions being smaller. Table III shows the numerical parameters used for the algorithms. It is necessary to use smaller under-relaxation factors for the implicit time integration scheme than for the explicit time integration scheme. This is because a non-linear system of equations must be solved at each physical time step if an implicit time integration scheme is used for the convective terms as in Equation (5). In Cases 10 and 12 the value of β required by the artificial compressibility algorithm was chosen by the same trial and error method as used for the explicit time integration scheme. As for the explicit time integration scheme, a few hundred physical time steps were needed to determine whether the solution was diverging. The choice of the parameters α_p, α_u, α_v, α_w required by the SCGS-PP algorithm was made simply by using values that the authors have used in the past for other flows,

DOI: 10.1002/fld

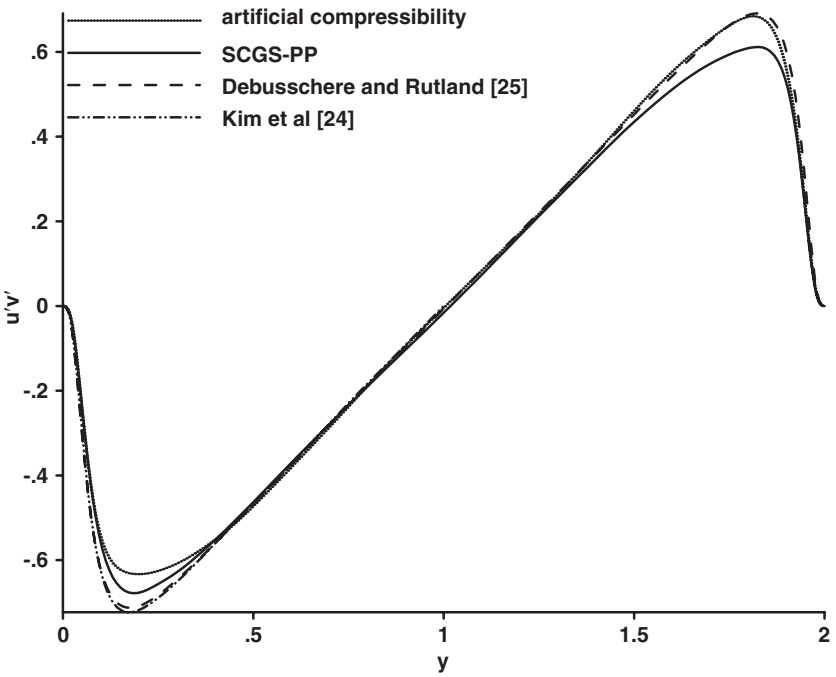

Figure 10. Comparison of statistics with results of others, channel flow, explicit time integration scheme (Equation (3)), $\Delta t = 0.00005$.

Table III. Channel flow time step and numerical parameters, implicit time integration scheme.

Case	Time step and numerical parameters	Notes
Case 10	Artificial compressibility, $\Delta t = 0.0005,\ \beta = 575,\ \alpha_u = \alpha_v = \alpha_w = 0.4$	Optimal value of β
Case 11	Artificial compressibility, $\Delta t = 0.0005,\ \beta = 125,\ \alpha_u = \alpha_v = \alpha_w = 0.4$	Non-optimal value of β
Case 12	Artificial compressibility, $\Delta t = 0.0015,\ \beta = 125,\ \alpha_u = \alpha_v = \alpha_w = 0.4$	Optimal value of β
Case 13	SCGS-PP, $\Delta t = 0.0005,\ \alpha_p = 0.225,\ \alpha_u = \alpha_v = \alpha_w = 0.8$	
Case 14	SCGS-PP, $\Delta t = 0.0015,\ \alpha_p = 0.225,\ \alpha_u = \alpha_v = \alpha_w = 0.8$	

such as jets in cross-flow and flow over spheres and cylinders. Case 11 is included to demonstrate the effect of using the same value of β that is optimal for a time step of $\Delta t = 0.0015$ for a simulation for which a smaller time step of $\Delta t = 0.0005$ is chosen. Using $\beta = 575$ (which is optimal for a time step of $\Delta t = 0.0005$) for a simulation using a larger time step of $\Delta t = 0.0015$, results in a diverging solution. Sixty iterations were used for both time steps. For computational efficiency terms, such as $\alpha/\mathbf{G}^m$, $\chi_{i,j,k}$, $\chi_{i,j,k}/^1 G_{i,j,k}\Delta x_i^u$ in Equations (14) and (19) which depend on the solution and involve computationally expensive division operations, are recomputed at the first iteration every 150 time steps.

Residual level: Figure 11 shows the history as a function of time of the average residual of the continuity equation at a time step of $\Delta t = 0.0005$. For the artificial compressibility algorithm, using the value of β that is optimal for a time step of $\Delta t = 0.0015$ results in a residual level that

DOI: 10.1002/fld

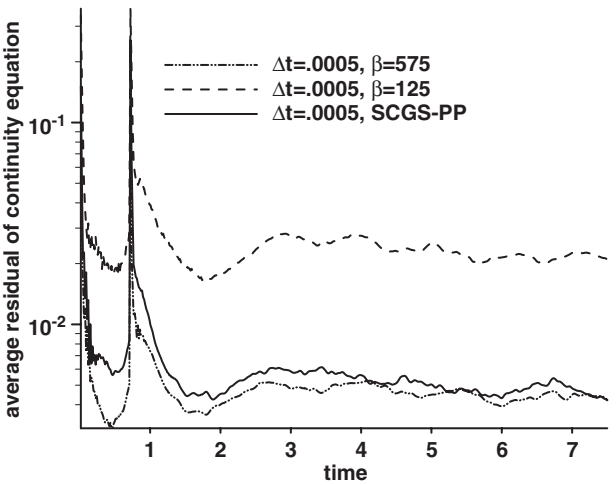

Figure 11. Average residual of the continuity equation, channel flow, implicit time integration scheme (Equation (5)), $\Delta t = 0.0005$.

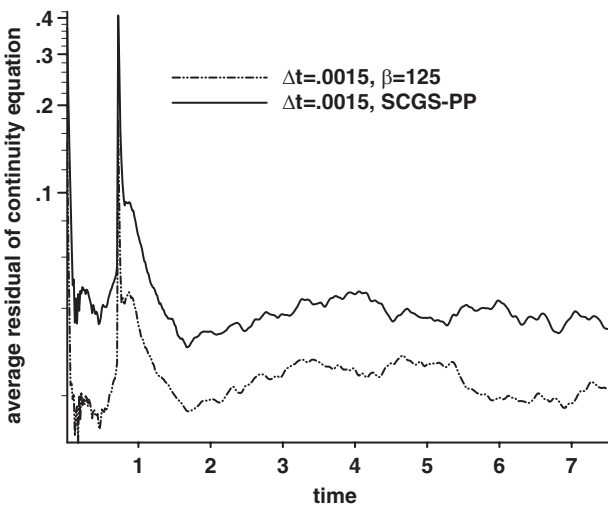

Figure 12. Average residual of the continuity equation, channel flow, implicit time integration scheme (Equation (5)), $\Delta t = 0.0015$.

is somewhat higher than that obtained using an optimal value of β for this time step $\Delta t = 0.0005$. The difference, however, is not nearly as great as in the explicit time integration cases. The spike in the graph is the result of turning off the wall jets used to trigger the transition to turbulence. For both time steps, a slightly lower residual is obtained using the artificial compressibility algorithm (Figure 12). However, note that this occurs only after using an optimal value of β which was obtained by a painstaking trial and error process. The reason for the small differences in the residual history between the algorithms may be due to the fact that at these higher CFL numbers more of the error

DOI: 10.1002/fld

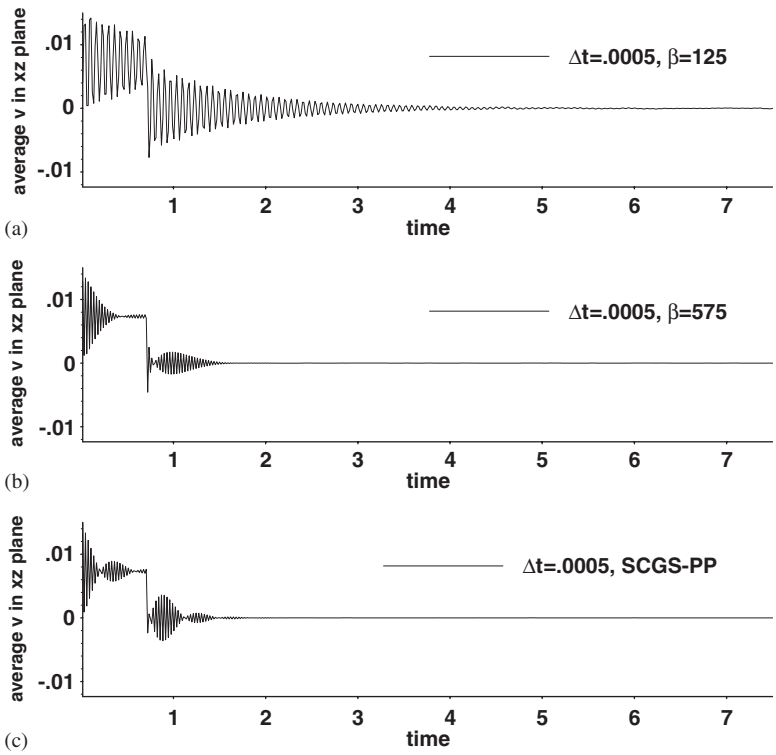

Figure 13. Spatially averaged v in xz plane, channel flow, implicit time integration scheme (Equation (5)), $\Delta t = 0.0005$.

exists in the low wave number range. In this range, the smoothing properties of the algorithms approach each other for the linear system resulting from explicit time integration as was shown by the Fourier analysis. It is to be expected that this behaviour also exists for the non-linear system resulting from implicit time integration. As for the explicit case, using a value of β that is optimal for a certain time step for the same simulation using a larger time step results in a diverging solution for the artificial compressibility algorithm.

Other measures of error: Figure 13 shows the time history of the instantaneous value of v integrated over the entire xz plane at $y = h$. It can be seen that the errors in this quantity take significantly longer to decay for the artificial compressibility algorithm, if a non-optimized value of β is used. If an optimal value of β is used the artificial compressibility algorithm performs as well as SCGS-PP, however, finding this optimal value of β requires a painstaking trial and error process even if only the physical time step is changed. Note that no attempt in the present work is made to tune the values of the under-relaxation factors for the SCGS-PP algorithm for different values of the time step.

Statistics: All implicit cases were run for the same length of time and had statistics collected over the same period of time as the explicit cases. Figure 14 shows the mean v for the algorithms at a time step of $\Delta t = 0.0005$. As for the explicit cases, using the value of β that is optimal for a time step of $\Delta t = 0.0015$ results in significantly greater error than that obtained using the optimal value of β for this time step of $\Delta t = 0.0005$.

DOI: 10.1002/fld

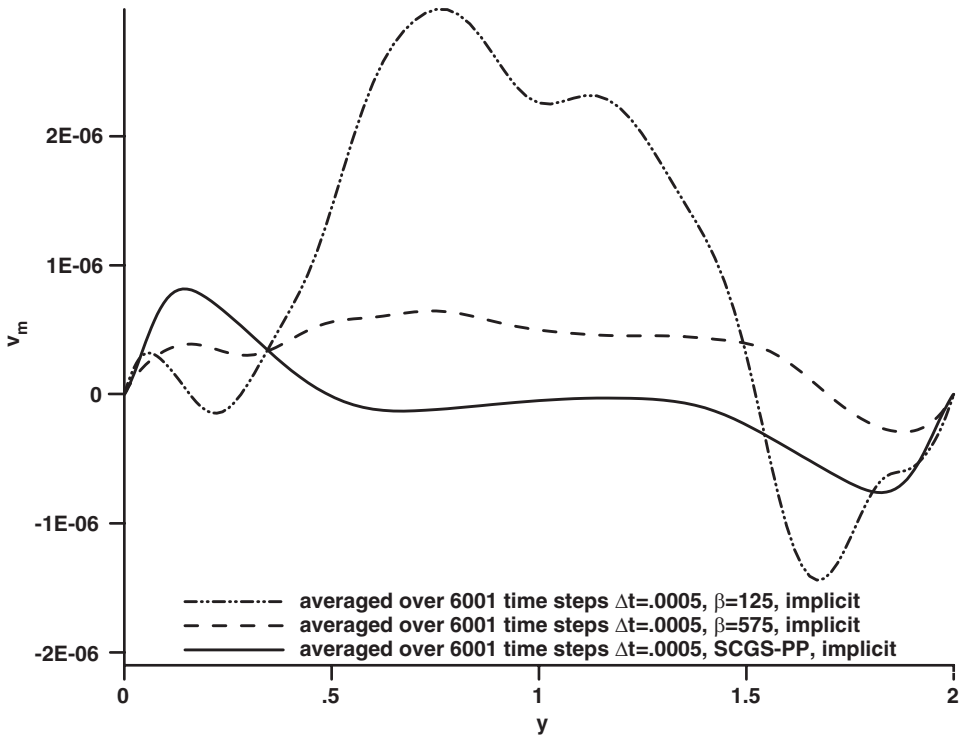

Figure 14. Temporally and spatially averaged v, channel flow, implicit time integration scheme (Equation (5)), $\Delta t = 0.0005$.

7.2. *DNS of driven cavity*

A driven-cavity flow at a Reynolds number of 10 000 based on the moving wall velocity and the cavity height h is simulated. At this Reynolds number the flow is unsteady. The flow is driven by the movement of the top wall in the positive x direction. All quantities are non-dimensionalized by h and the velocity of the top wall. A schematic of the flow domain along with boundary conditions and dimensions is given in Figure 15. The grid dimensions are $64 \times 64 \times 64$. A stretched grid, in which the ratio of the maximum grid spacing to the minimum is 5.54, is used to concentrate points near the walls in all three directions. For the velocity field, the initial conditions for the interior velocity are that all three components of velocity are set to a random number between $-\frac{1}{2}$ and $\frac{1}{2}$. This is done to observe the transient behaviour of the flow as it reaches a physically realistic state from the unphysical initial state. The initial condition for pressure is to set it to zero everywhere.

7.2.1. Explicit time integration scheme results. Two different time steps, $\Delta t = 0.0012$ and 0.0003, are used to demonstrate the effect of the time step on the numerical parameters of the algorithms. For this flow, the time step of 0.0012 corresponds to a maximum CFL number of approximately 0.16 based on the u velocity and grid spacing in the x direction. Table IV shows the numerical parameters used for the algorithms. In Cases 1 and 3 the value of β required by the artificial compressibility algorithm was chosen using the same trial and error method used for the channel

DOI: 10.1002/fld

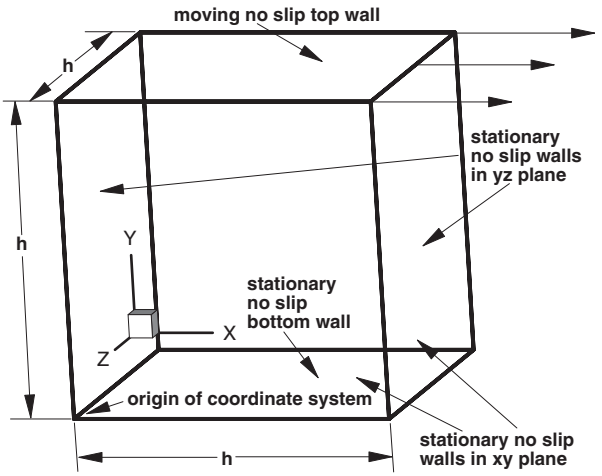

Figure 15. Driven cavity geometry and boundary conditions.

Table IV. Driven cavity time step and numerical parameters, explicit time integration scheme.

Case	Time step and numerical parameters	Notes
Case 1	Artificial compressibility, $\Delta t = 0.0003,\ \beta = 60,\ \alpha_u = \alpha_v = \alpha_w = 0.915$	Optimal value of β
Case 2	Artificial compressibility, $\Delta t = 0.0003,\ \beta = 4.25,\ \alpha_u = \alpha_v = \alpha_w = 0.915$	Non-optimal value of β
Case 3	Artificial compressibility, $\Delta t = 0.0012,\ \beta = 4.25,\ \alpha_u = \alpha_v = \alpha_w = 0.915$	Optimal value of β
Case 4	SCGS-PP, $\Delta t = 0.0003,\ \alpha_p = \alpha_u = \alpha_v = \alpha_w = 0.925$	
Case 5	SCGS-PP, $\Delta t = 0.0012,\ \alpha_p = \alpha_u = \alpha_v = \alpha_w = 0.925$	

flow. For the SCGS-PP algorithm, the same parameters α_p, α_u, α_v, α_w that were used for the channel flow are used for this flow. Case 2 is included to demonstrate the effect of using the value of β that is optimal for a time step of $\Delta t = 0.0012$ for a simulation using a different time step of $\Delta t = 0.0003$. Using the value of $\beta = 60$, which is optimal for a time step of $\Delta t = 0.0003$, for a simulation using a time step of $\Delta t = 0.0012$, results in a diverging solution for which no results could be obtained. Eight iterations were used for a time step of $\Delta t = 0.0003$ and 10 for a time step of $\Delta t = 0.0012$.

Residual level: Figure 16 shows the history as a function of time of the average residual of the continuity equation at a time step of $\Delta t = 0.0003$. It can be seen that for the artificial compressibility algorithm using the value of β that is optimal for a time step of $\Delta t = 0.0012$ results in a residual level that is an order of magnitude higher than that obtained using an optimal value of β for this particular time step. Note that while the residual level decays differently than the channel flow it also does not go to zero since the flow is unsteady.

Other measures of error: A measure of error that can be used for this flow is the mass flow through a plane that cuts through the entire cavity. Due to the boundary conditions for this flow, the mass flow through such a plane should be zero. For convenience, the plane chosen is the yz plane in the centre of the cavity (at $x = 0.5h$). Figure 17 shows the behaviour of this integrated quantity as a function of time. A dramatic increase in the error for the artificial compressibility algorithm

DOI: 10.1002/fld

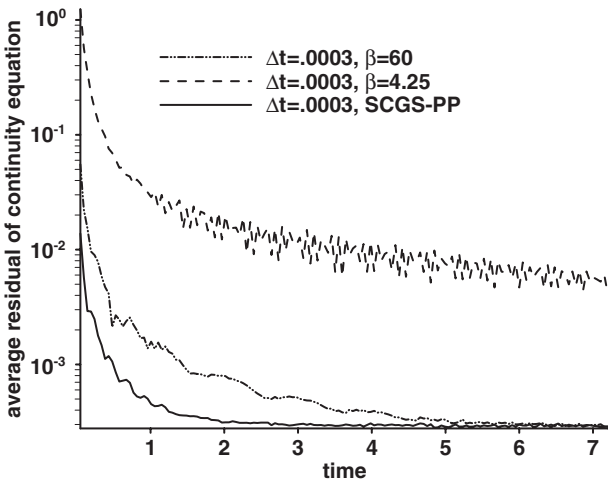

Figure 16. Average residual of the continuity equation, driven cavity, explicit time integration scheme (Equation (3)), $\Delta t = 0.0003$.

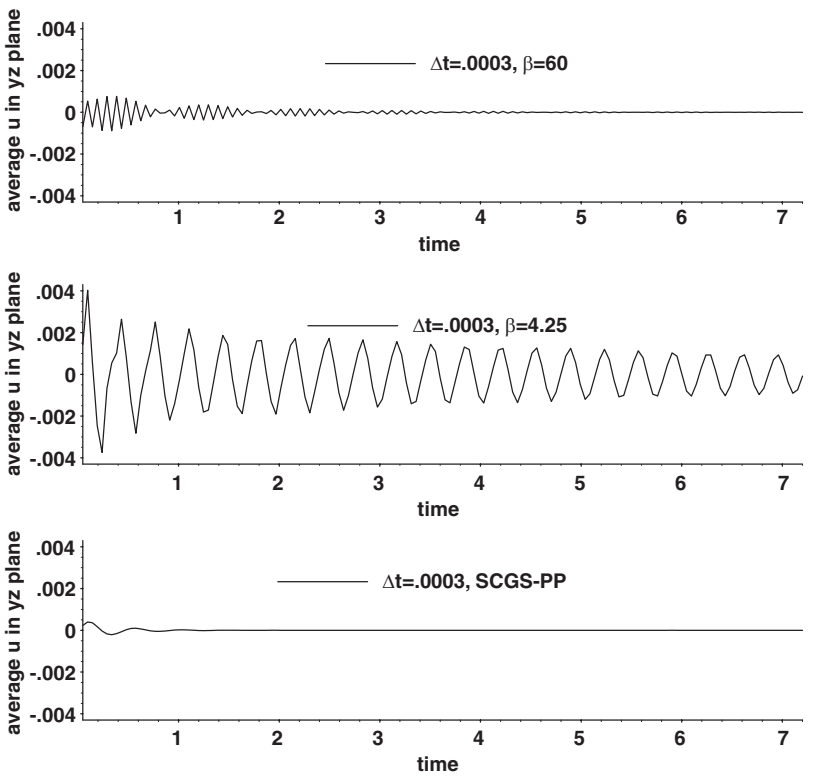

Figure 17. Mass flow through yz plane at $x = 0.5h$, driven cavity, explicit time integration scheme (Equation (3)), $\Delta t = 0.0003$.

DOI: 10.1002/fld

can be seen if a non-optimized value of β is used. Even when using the optimal value of β the artificial compressibility algorithm has a significantly larger error than the SCGS-PP algorithm.

7.2.2. Implicit time integration scheme results. Time steps of $\Delta t = 0.003, 0.006$ are used, with the time step of $\Delta t = 0.006$ corresponding to a maximum CFL number of approximately 0.85 based on the v velocity and grid spacing in the y direction. Table V shows the numerical parameters used for the algorithms. As for the channel flow, it is necessary to use smaller under-relaxation factors for the implicit time integration scheme than for the explicit time integration scheme. As demonstrated previously, using a value of β that is optimal for a smaller time step of $\Delta t = 0.003$ in a simulation with a larger time step of $\Delta t = 0.006$, results in a solution that quickly diverges. Twenty-four iterations were used for a time step of $\Delta t = 0.003$ and 30 for a time step of $\Delta t = 0.006$. For computational efficiency, terms such as $\alpha/\mathbf{G}^m$, $\chi_{i,j,k}$, $\chi_{i,j,k}/^1 G_{i,j,k}\Delta x_i^u$ in Equations (14) and (19) which depend on the solution and involve computationally expensive division operations, are recomputed at the first iteration every 60 time steps.

Residual level: Figure 18 shows the history as a function of time of the average residual of the continuity equation at a time step of $\Delta t = 0.003$. It can be seen that for the artificial compressibility

Table V. Driven cavity time step and numerical parameters, implicit time integration scheme.

Case	Time step and numerical parameters	Notes
Case 10	Artificial compressibility, $\Delta t = 0.003,\ \beta = 2.6,\ \alpha_u = \alpha_v = \alpha_w = 0.85$	Optimal value of β
Case 11	Artificial compressibility, $\Delta t = 0.003,\ \beta = 0.75,\ \alpha_u = \alpha_v = \alpha_w = 0.85$	Non-optimal value of β
Case 12	Artificial compressibility, $\Delta t = 0.006,\ \beta = 0.75,\ \alpha_u = \alpha_v = \alpha_w = 0.85$	Optimal value of β
Case 13	SCGS-PP, $\Delta t = 0.003,\ \alpha_p = \alpha_u = \alpha_v = \alpha_w = 0.7$	
Case 14	SCGS-PP, $\Delta t = 0.006,\ \alpha_p = \alpha_u = \alpha_v = \alpha_w = 0.7$	

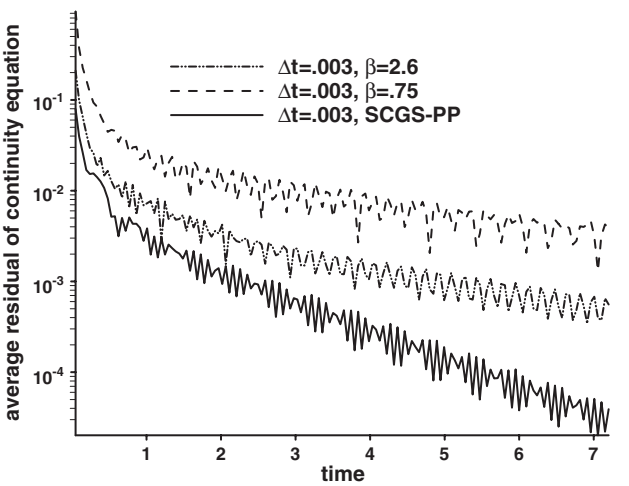

Figure 18. Average residual of the continuity equation, driven cavity, implicit time integration scheme (Equation (5)), $\Delta t = 0.003$.

DOI: 10.1002/fld

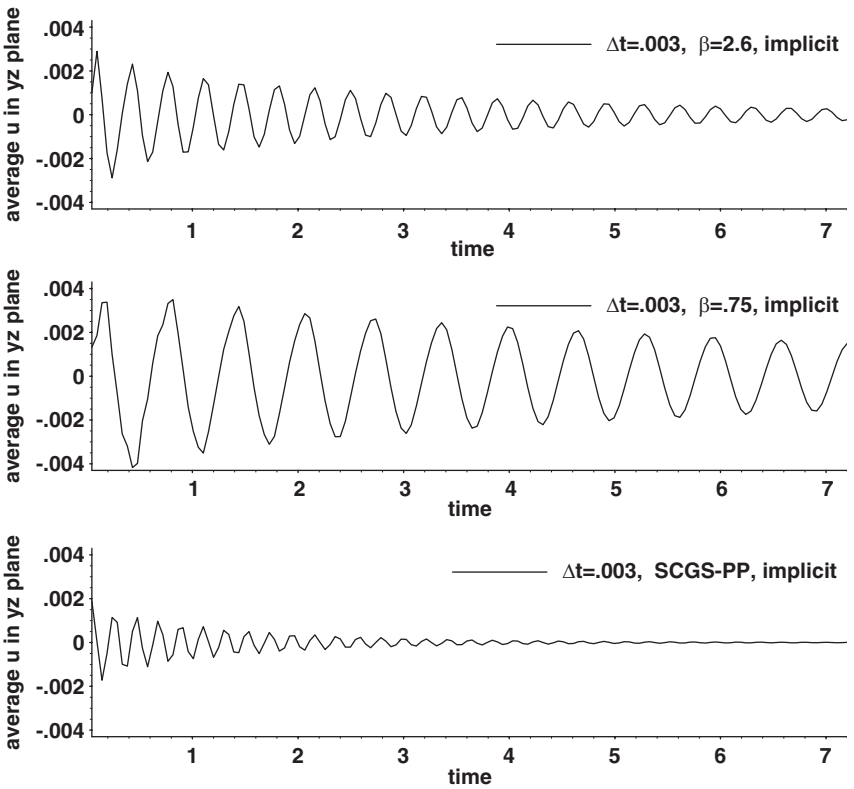

Figure 19. Mass flow through yz plane at $x = 0.5h$, driven cavity, implicit time integration scheme (Equation (5)), $\Delta t = 0.003$.

algorithm using the value of β that is optimal for a time step of $\Delta t = 0.006$ results in a residual level that is an order of magnitude higher than that obtained using an optimal value of β for this particular time step of $\Delta t = 0.003$. Note that the artificial compressibility algorithm, even when using the optimal value of β, performs significantly worse than the SCGS-PP algorithm.

Other measures of error: Figure 19 shows the behaviour of the mass flow through the same yz plane used with the explicit scheme as a function of time. A significantly larger error for the artificial compressibility algorithm can be seen if a non-optimized value of β is used. Note that even when using the optimal value of β the artificial compressibility algorithm has a larger error than the SCGS-PP algorithm.

7.3. DNS of flow over a backward-facing step

The results are concluded by applying the SCGS-PP algorithm to DNS of flow over a backward-facing step. This flow is a good test of a numerical method as it has a strong separation and recirculation region. A schematic of the flow domain along with boundary conditions and dimensions is given in Figure 20. Referring to Figure 20, $L_f = 5h$, $L_p = 4h$, $L_{up} = 10h$ and $L_{down} = 20$. All quantities are non-dimensionalized by the step height h and the freestream velocity U_0. The

DOI: 10.1002/fld

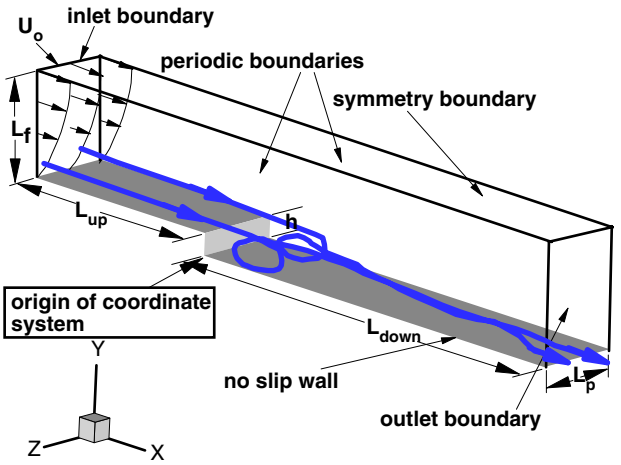

Figure 20. Backward-facing step schematic.

Reynolds number based on freestream velocity and step height is 5000. A total of 8 465 792 grid points are used in discretizing the domain using a Cartesian grid. Grid points are concentrated near all three of the no-slip walls. Due to the large time and memory constraints the domain is decomposed into 128 Cartesian zones and each zone is assigned to a processor on a parallel computer.

7.3.1. Boundary conditions. Providing realistic inflow boundary conditions for a turbulent flow is a challenging part of a simulation. The most physically realistic method is to trip the boundary layer while extending the upstream domain far enough that the correct spatially developed boundary layer results. In the present study the mean boundary layer is tripped by means of a row of hypercubes just downstream of the inlet. In the study with which comparisons are made [29] (referred to as LMK in figures) the inflow condition was provided by adding a fluctuating component with a prescribed energy spectrum onto the mean profile. The hypercubes are represented by the immersed boundary method [30, 31]. Due to the relatively coarse grid near the inlet, only the u velocity points were defined as immersed boundary points. This coarse grid precludes any meaningful resolution of the hypercubes, which simply function as generic obstacles in the flow field. At the inlet a spline fit of the experimental data of [32–34] (referred to as JD in figures) was imposed on u and v while w was set to zero. All velocity components were set to zero on all walls. At the freestream boundary $y = L_f + h$ the conditions given by Equation (25) are imposed

$$\frac{\partial u}{\partial y} = \frac{\partial w}{\partial y} = v = 0 \tag{25}$$

At the outlet a convective boundary condition given by Equation (26) is used for all three velocity components

$$\frac{\partial \mathbf{u}}{\partial t} + U_c \frac{\partial \mathbf{u}}{\partial x} = 0 \tag{26}$$

DOI: 10.1002/fld

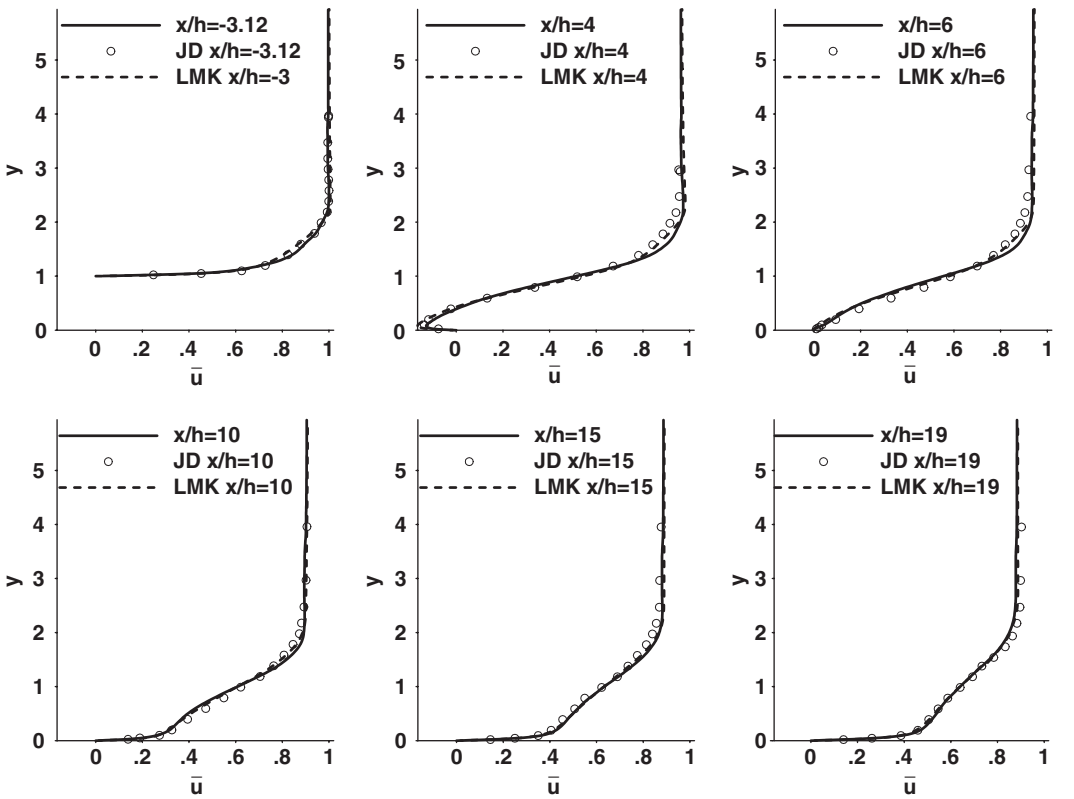

Figure 21. Mean u velocity at different x/h locations, flow past backward-facing step. JD corresponds to References [32–34] while LMK corresponds to Reference [29].

In Equation (26) U_c is a constant chosen such that when multiplied by the area of the outflow plane the result equals the total volume flow into the domain. This boundary condition allows structures to pass out of the domain without generating significant reflections or gradients that could affect events upstream.

7.3.2. Numerical issues. Five iterations of the SCGS-PP algorithm are done at each physical time step. The physical time step is 0.0005, resulting in a maximum CFL number of ≈ 0.15 at $x = 1 \times 10^{-3}$, $y = 2.3$. The first three u velocity grid spacings in the x direction from the step face are at 3.732×10^{-3}, 3.924×10^{-3} and 4.125×10^{-3}. The first three v velocity grid spacings in the y direction from the upstream wall are at 5.256×10^{-3}, 3.777×10^{-3} and 6.342×10^{-3}. The first three v velocity grid spacings in the x direction from the downstream wall are at 5.920×10^{-3}, 6.142×10^{-3} and 6.371×10^{-3}. The step size in the z direction is 5.26×10^{-3}. The initial conditions are u set equal to the freestream value upstream of the step and to U_c downstream of the step. At all interior points v and w and p are set to zero. The explicit second-order accurate Adams–Bashford time integration scheme given by Equation (4) is used for the convective and diffusive terms instead of the explicit third-order accurate Adams–Bashford time integration scheme used elsewhere in the present work.

DOI: 10.1002/fld

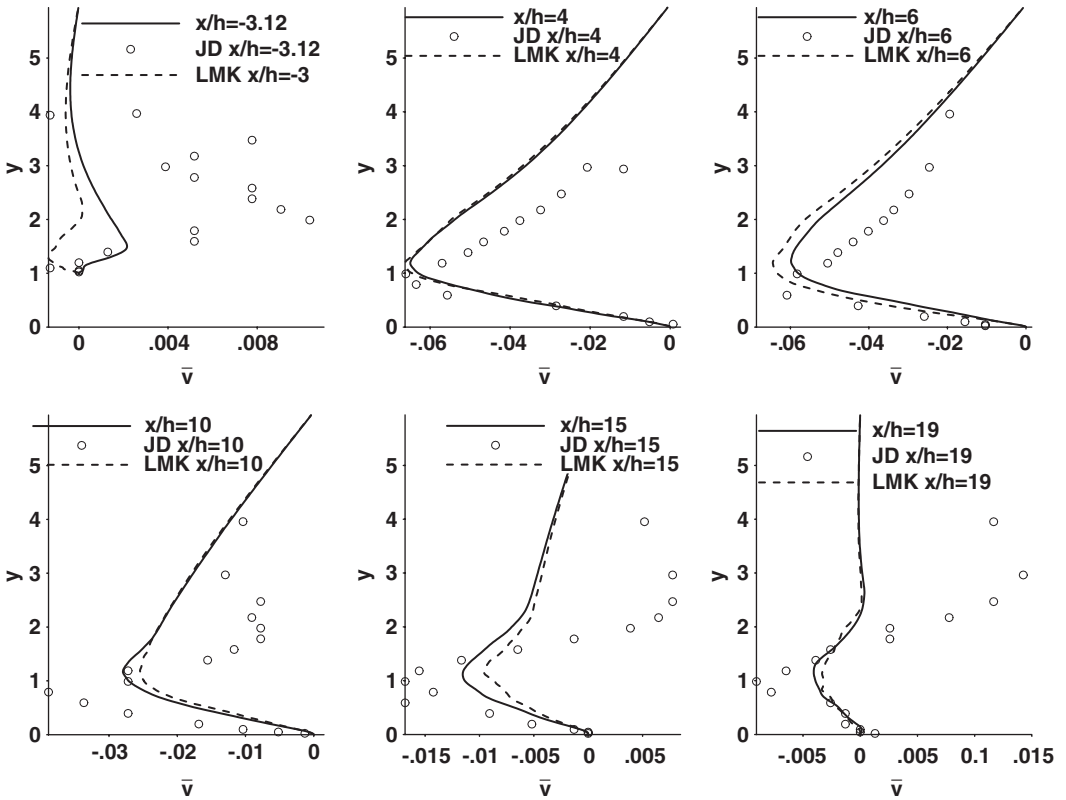

Figure 22. Mean v velocity at different x/h locations, flow past backward-facing step. JD corresponds to References [32–34] while LMK corresponds to Reference [29].

7.3.3. Statistics. Because the flow field is homogenous in the z direction, spatial averaging is carried out in this direction. The statistical averages of the various quantities are computed as a running average. The solution is advanced for 180 000 time steps before statistics are collected. This corresponds to travelling $90h$ at the freestream velocity. The statistics are averaged over 460 700 time steps. This corresponds to travelling $230.35h$ at the freestream velocity.

Excellent agreement with DNS data [29] and experimental data [32–34] is seen for the mean u profile in Figure 21. In particular, the inlet boundary condition results in a correct u profile upstream of the backward-facing step. The streamwise Reynolds stress component is shown in Figure 23. The $\overline{u'v'}$ component of the Reynolds stress tensor is shown in Figure 24. Results agree well with [29, 32–34]. For v mean in Figure 22, a significant discrepancy exists between the experimental data and the computational results of the present work and the computational results of [29]. The DNS of both the present work and the DNS of [29] show good agreement with each other but only qualitative agreement with the experimental data. Note that the values of v mean are relatively small and it is possible that the uncertainties at these low magnitudes contribute to the observed differences.

The mean reattachment length is found to be $6.278h$. It is determined by locating the first pair (in the x direction) of grid points closest to the wall where the sign of the mean streamwise velocity

DOI: 10.1002/fld

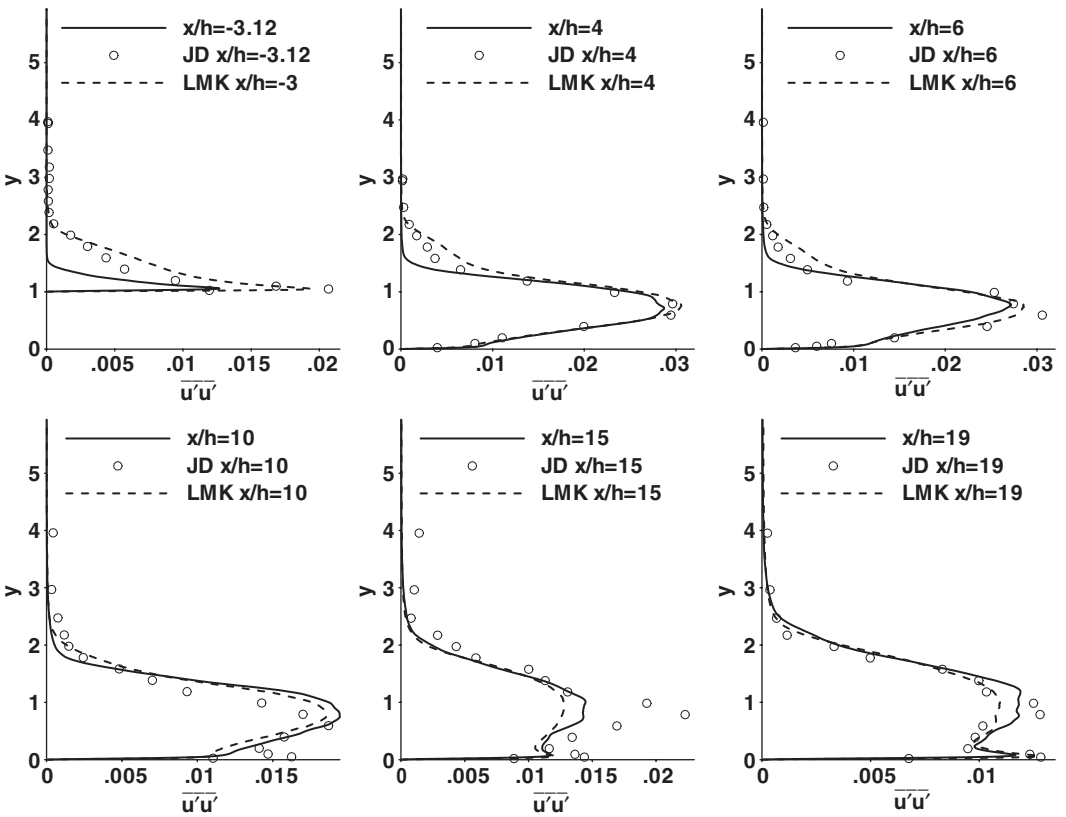

Figure 23. $\overline{u'u'}$ at different x/h locations, flow past backward-facing step. JD corresponds to References [32–34] while LMK corresponds to Reference [29].

changes. The mean reattachment length as measured by Jovic and Driver [32–34] is $6h$. The value computed by Le *et al.* [29] is $6.28h$.

8. COMPUTATIONAL WORK AND EFFICIENCY

8.1. Computational work

The computational work involved in the application of an algorithm is an important factor in weighing its usefulness. It is difficult to obtain an accurate measure of the computational work involved in an algorithm which will be used on different types of computers, since this is a function of the computer architecture, the problem dimensions, the programming methodology and the optimizations performed by the compiler. However, a useful comparison can be made by comparing the number of floating point operations required by each algorithm. Table VI lists the ratio of floating point operations of SCGS-PP to artificial compressibility for different numbers of iterations. The ratio differs depending on the number of iterations as at each physical time step there are certain operations common to both algorithms which are performed once each physical time

DOI: 10.1002/fld

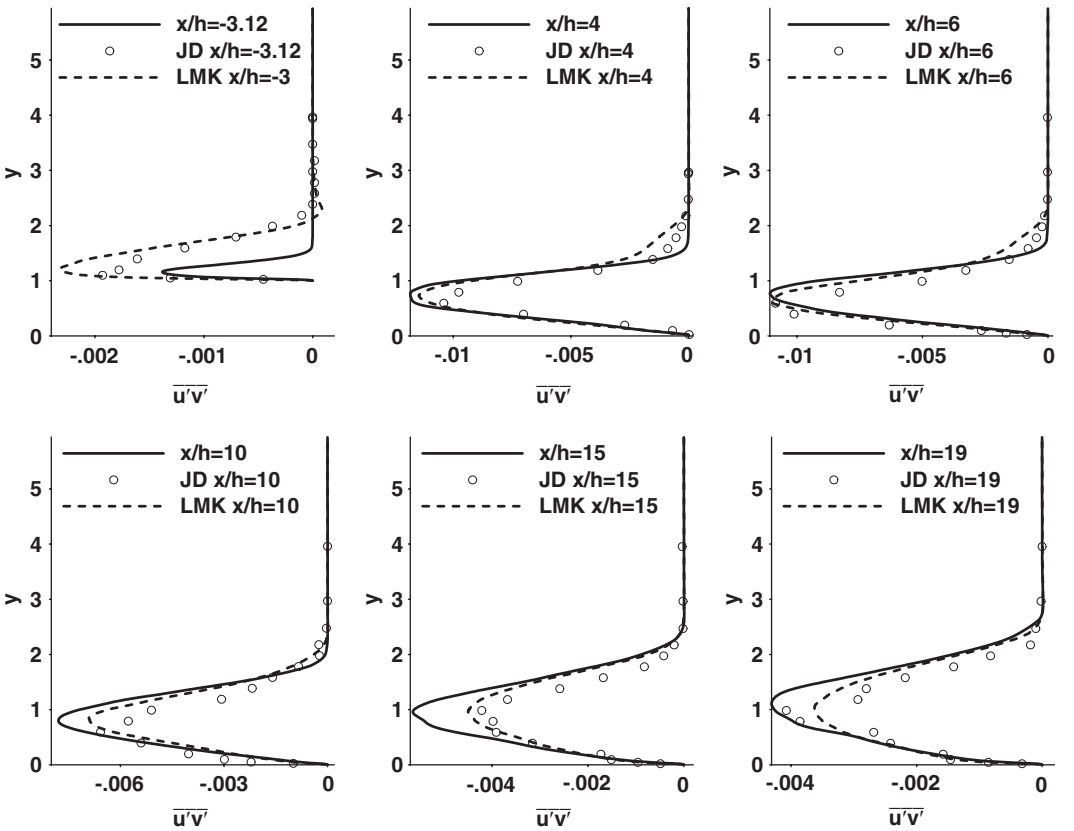

Figure 24. $\overline{u'v'}$ at different x/h locations, flow past backward-facing step. JD corresponds to References [32–34] while LMK corresponds to Reference [29].

step. For a fair comparison, the values in Table VI are obtained using the same spatial convection diffusion scheme for the explicit scheme as described for the implicit scheme. For the explicit scheme the ratio asymptotes to 1.774 while for commonly used numbers of iterations it is less. The ratio for the implicit scheme is smaller, asymptoting to 1.081, because the amount of additional computational work consumed by the SCGS-PP algorithm is small relative to the computational work consumed by the convection and diffusion terms. Table VII shows that the results of tests of the actual wall clock time of the algorithms on different computers are close to that predicted by comparing the number of floating point operations. The results in Table VII are obtained by using Equation (3) and the spatial discretization scheme described in Section 2.3.3 for the explicit time integration scheme and Equation (5) and the spatial discretization scheme described in Section 2.3.1 for the implicit time integration scheme.

8.2. Computational efficiency

The above measure compares the computational work required to solve a time step with both algorithms using the same number of iterations. However, the SCGS-PP algorithm has greater

DOI: 10.1002/fld

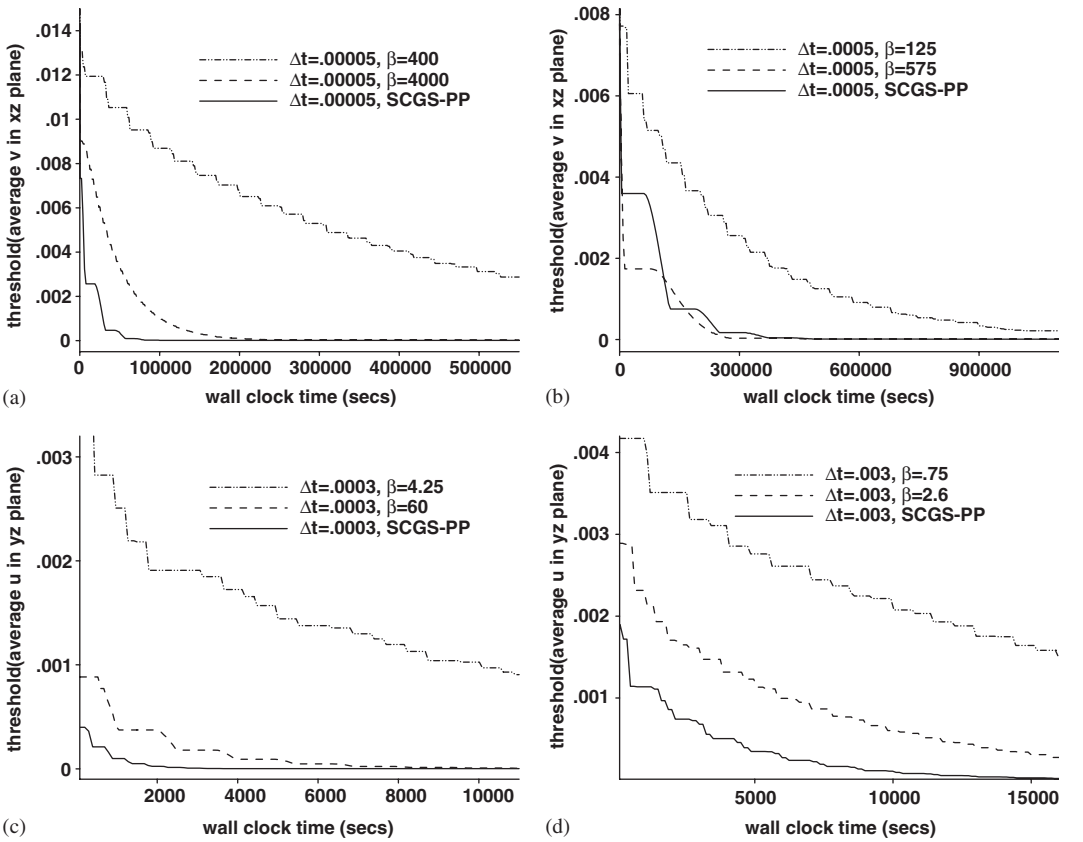

Figure 25. Computational efficiency: (a) channel flow explicit scheme; (b) channel flow implicit scheme; (c) driven cavity explicit scheme; and (d) driven cavity implicit scheme.

computational efficiency since it reduces the residual more per iteration. Figure 25 compares the computational efficiency of the algorithms in reducing the error in mass flow through the planes described above for both the channel and driven-cavity flows. The threshold function in Figure 25 is defined as the maximum of the absolute value of a function extending from x to $x_{\max}$, i.e. $\text{threshold}[x] = \max|f[x : x_{\max}]|$ and is used to show when a function reaches and remains below a value. The x axis in Figure 25 is the wall clock time on a single 2.4 GHz Opteron 250 processor. For the explicit time integration scheme, the SCGS-PP algorithm is significantly more computationally efficient than the artificial compressibility algorithm even when the artificial compressibility algorithm uses the optimal value of β. Results are mixed for the implicit time integration scheme if the artificial compressibility algorithm uses the optimal value of β, if, however, it does not, then the SCGS-PP algorithm is much more computationally efficient.

8.2.1. Note on multi-grid. Multi-grid algorithms have seen much use in solvers for the Navier–Stokes equations and arguably are the best general purpose solver for these equations. Therefore, it is appropriate to address the use of our method within a multi-grid algorithm. Assuming appropriate

DOI: 10.1002/fld

Table VI. Ratio of floating point operations per real time step of SCGS-PP to artificial compressibility.

# iterations	Explicit	Implicit
1	1.063	1.366
2	1.117	1.226
4	1.203	1.155
8	1.322	1.118
16	1.455	1.100
∞	1.774	1.081

Table VII. Ratio of actual wall clock time per real time step of SCGS-PP to artificial compressibility for a $65 \times 65 \times 65$ grid on different computers.

	IBM POWER4 1.45 GHz		2.4 GHz Opteron 250		Pentium(R) IV 43.2 GHz	
# iterations	Explicit	Implicit	Explicit	Implicit	Explicit	Implicit
1	1.077	1.953	1.055	1.926	1.070	1.758
2	1.142	1.618	1.084	1.540	1.128	1.492
4	1.263	1.415	1.195	1.285	1.226	1.289
8	1.456	1.302	1.357	1.218	1.385	1.221
16	1.731	1.249	1.436	1.138	1.589	1.167
32	2.049	1.219	1.719	1.101	1.744	1.150

prolongation and restriction operators, the efficiency of a multi-grid algorithm depends on the error reduction (or smoothing) properties of the underlying iterative method. We have demonstrated the improved smoothing properties of our methods on a number of problems including one where we analysed the smoothing properties in wave number space. This analysis of the methods in wave number space shows that they reduce the appropriate error components for a multi-grid iterative method (i.e. the high wave number component). However, there may be concern that the values of the relaxation parameters may change on the coarser grids used in multi-grid methods. We have used the same values for the relaxation parameters for three different problems (one set of relaxation parameters for the explicit time integration scheme is used on all problems and another set is used with the implicit time integration scheme on all problems) and have shown improved error reduction properties over the artificial compressibility method. This shows the insensitivity of the relaxation parameters used by our methods for a variety of problems and we believe this is a much stronger test of the sensitivity of the relaxation parameters than that given by a multi-grid method where the same problem is solved but on coarser grids. While multi-grid algorithms are an excellent general purpose solver for the Navier–Stokes equations, their benefits are limited for DNS problems such as we are concerned with in the present work. The reason lies in the form of the error at each time step of an unsteady flow, where small CFL numbers are needed for temporal accuracy in DNS. Since the previous time step is available as a starting guess in the iterative process and as its difference from the solution (i.e. the error) lies almost entirely in the spatial high wave number region (this being more so the smaller the CFL number) there is very little of the error that can be represented on coarser grids. Therefore, the payoff from the computational cost

DOI: 10.1002/fld

associated with transferring the solution to coarser grids and iterating on these grids is very small and it is frequently more efficient to perform all iterations on the fine grid where the error is well represented.

9. CONCLUSION

A new algorithm (SCGS-PP) has been developed which is a simple modification of the artificial compressibility algorithm. This new algorithm eliminates the difficulty of choosing the artificial compressibility parameter. This is a significant improvement as the artificial compressibility parameter is difficult to choose as it can vary four orders of magnitude depending on the flow and the physical time step of the simulation. This artificial compressibility parameter is strictly a numerical artefact and in no way affects the physics of the flow. While the artificial compressibility algorithm can perform well if a proper choice of the artificial compressibility parameter is made, improper choices lead either to slow convergence or divergence of the solution. For the new algorithm the choice of the relaxation parameters is straightforward, and the same values can be used to provide robust convergence for a range of application problems. For two very different flows, using both explicit and implicit time integration schemes, the same values of the relaxation parameters are used with the SCGS-PP algorithm while the artificial compressibility algorithm requires values of the artificial compressibility parameter that vary by almost two orders of magnitude. In addition the convergence rate in various metrics of the SCGS-PP algorithm is much better than the artificial compressibility algorithm. Unlike the SCGS algorithm, the new algorithm is easily parallelized making it suitable for large-scale calculations such as DNS which require use of parallel computers.

ACKNOWLEDGEMENTS

This work was supported by the United States Department of Energy, The Office of Naval Research, Louisiana State University and The Center for Computation and Technology at Louisiana State University. The simulations were run on an IBM SP2, a Linux Xeon cluster and workstations at Louisiana State University, IBM SP3 at the Aeronautical Systems Center at Wright–Patterson Air Force Base and an IBM SP4 at the Arctic Region Supercomputing Center.

REFERENCES

1. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
2. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, 1980.
3. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
4. Pletcher RH, Chen K. On solving the compressible Navier–Stokes equations for unsteady flows at very low Mach numbers. *AIAA Paper 93-3368*.
5. Merkle CL, Choi Y-H. Computation of low-speed compressible flows with time-marching procedures. *International Journal for Numerical Methods in Engineering* 1988; **25**:293–311.
6. Vanka SP. A calculation procedure for three-dimensional steady recirculating flows using multigrid methods. *Computer Methods in Applied Mechanics and Engineering* 1986; **55**:321–338.
7. Muldoon F. Numerical methods for the unsteady incompressible Navier–Stokes equations and their application to the direct numerical simulation of turbulent flows. *Ph.D. Thesis*, Louisiana State University, 2004.
8. Degani AT, Fox GC. Parallel multigrid computation of the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1996; **128**:223–236.
9. Babuska I. The finite element method with Lagrangian multipliers. *Numerische Mathematik* 1973; **279**:179–192.

DOI: 10.1002/fld

10. Brezzi F. On the existence, uniqueness and approximation of saddle point problems arising from Lagrangian multipliers. *RAIRO—Analyse Numerical Analysis* 1974; **279**:129–154.
11. McHugh P, Ramshaw J. Damped artificial compressibility iteration scheme for implicit calculations of unsteady incompressible flow. *International Journal for Numerical Methods in Fluids* 1995; **21**:141–153.
12. Soh WY, Goodrich JW. Unsteady solution of incompressible Navier–Stokes equations. *Journal of Computational Physics* 1988; **79**:113–134.
13. Rogers SE, Kwak D. Steady and unsteady solutions of the incompressible Navier–Stokes equations. *AIAA Journal* 1991; **29**:603–610.
14. Beddhu M, Taylor LK, Whitfield DL. A time accurate calculation procedure for flows with a free surface using a modified artificial compressibility formulation. *Applied Mathematics and Computation* 1994; **65**:33–48.
15. Marx YP. Time integration schemes for the unsteady incompressible Navier–Stokes equations. *Journal of Computational Physics* 1994; **112**:182–209.
16. Muldoon F, Acharya S. Dynamics of large-scale structures for jets in a crossflow. *ASME Journal of Turbomachinery* 1998; **121**:557–587.
17. Kim W-W, Menon S. An unsteady incompressible Navier–Stokes solver for large eddy simulation of turbulent flows. *International Journal for Numerical Methods in Fluids* 1999; **31**:983–1017.
18. Corvellec C, Bruel P, Sabelnikv VA. A time accurate scheme for the calculations of unsteady reactive flows at low Mach number. *International Journal for Numerical Methods in Fluids* 1999; **29**:207–227.
19. Gilmanov A, Sotiropoulos F. A hybrid Cartesian/Immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies. *Journal for Computational Physics* 2005; **207**:457–492.
20. Turkel E. Preconditioned methods for solving the incompressible and low speed compressible equations. *Journal of Computational Physics* 1987; **72**:277–298.
21. Ham FE, Lien FS, Strong AB. Multiple semi-coarsened multigrid method with application to large eddy simulation. *International Journal for Numerical Methods in Fluids* 2006; **50**:579–596.
22. Thompson MC, Ferziger JH. An adaptive multigrid technique for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **82**:94–121.
23. Wesseling P. *An Introduction to Multigrid Methods*. Wiley: Chichester, 1992.
24. Kawamura H, Ohsaka K, Abe H, Yamamoto K. DNS of turbulent heat transfer in channel flow with low to medium-high Prandtl number. *International Journal of Heat and Fluid Flow* 1998; **19**:482–491.
25. Choi H, Moin P. Effects of the computational time step on numerical solutions of turbulent flow. *Journal of Computational Physics* 1994; **113**:1–4.
26. Ham FE, Lien FS, Strong AB. A fully conservative second-order finite difference scheme for incompressible flow on nonuniform grids. *Journal of Computational Physics* 2002; **117**:117–133.
27. Kim J, Moin P, Moser R. Turbulence statistics in a fully developed channel flow at low Reynolds number. *Journal of Fluid Mechanics* 1987; **177**:133–166.
28. Debusschere B, Rutland CJ. Turbulent scalar transport mechanisms in plane channel and Couette flow. *International Journal of Heat and Mass Transfer* 2003; **47**:1771–1781.
29. Le H, Moin P, Kim J. Direct numerical simulation of turbulent flow over a backward facing step. *Journal of Fluid Mechanics* 1997; **330**:349–374.
30. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics* 2000; **161**:35–60.
31. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220–252.
32. Jovic S, Driver D. Backward-facing step measurements at low Reynolds number $Re = 5000$. *NASA TM 108807*.
33. Jovic S, Driver D. Reynolds number effect on the skin friction in separated flows behind a backward-facing step. *Experiments in Fluids* 1995; **18**:464–467.
34. Driver D, Jovic S. CMP31: backward facing step-experiment. *AGARD Advisory Report*, *AGARD-AR-345*. (A selection of test cases for the validation of large-eddy simulations of turbulent flows 1998; 195–196).

DOI: 10.1002/fld