Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 209 (2005) 504–540
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Design of a high-performance unsteady Naiver–Stokes solver using a flexible-cycle additive-correction multigrid technique

Hongyi Xu *, Weixing Yuan, Mahmood Khalid

Institute for Aerospace Research, National Research Council of Canada, Ottawa, Ont., Canada K1A 0R6

Received 14 September 2004; received in revised form 10 March 2005; accepted 11 March 2005
Available online 1 June 2005

Abstract

A multigrid technique, based on a flexible-cycle additive-correction multigrid (FCAC-MG) scheme, is utilized to design a high-performance solver for the unsteady incompressible Navier–Stokes (N–S) equations. The unsteady incompressible N–S solver discussed here incorporates the fractional step method and finite volume discretization over a staggered Cartesian grid. The current research indicates that the FCAC-MG acceleration technique is highly efficient, reliable and robust, which makes it feasible for CPU-intensive computations, such as large eddy simulation (LES) and direct numerical simulation (DNS). The high efficiency and the robustness of the solver are achieved through developing the FCAC-MG acceleration technique as well as flow-physics oriented solving strategies. The flow solver based on the FCAC-MG technique is applied to both temporal and spatial turbulence simulations using both LES and DNS. The residuals of the large-scale algebraic equation system are guaranteed to be continuously driven down to the level of the computer machine round-off error on each time marching step, which warrants strong conservations of mass and momentum satisfied over all the control volumes. In addition, the flow solver developed in the current research possesses the potential capability for handling complex geometry flows since the performance of the solver does not rely on the use of special type of operations on the large-scale algebraic equation system, unlike the cyclic reduction algorithm, but instead it is driven by a flow-physics oriented solving strategy. LES for flows in a square duct, a square annular duct and a confined square coaxial jet are performed to present the advantages of the FCAC-MG technique and the flow-physics oriented solving strategy. The analyses of the simulation results for these three flow configurations provide strong evidence that the flow solver based on FCAC-MG method is capable of capturing the major characteristics of turbulence physics and correctly predicting the relevant turbulent flow phenomena.

* Corresponding author.
E-mail address: Hongyi.xu@nrc-cnrc.gc.ca (H. Xu).

0021-9991/$ - see front matter.
doi:10.1016/j.jcp.2005.03.029

1. Introduction

Numerical simulation of the unsteady three-dimensional (3D) incompressible Navier–Stokes (N–S) equations is an area of primary research interest since these equations govern many industrial flows, such as low-speed aerodynamics and hydrodynamics. Research in this area has particularly intensified recently owing to a desire to simulate realistic turbulence using large eddy simulation (LES) or direct numerical simulation (DNS). It is well known that a huge number of grid points are required to accurately resolve turbulence structures and to obtain grid-independent solutions. Because of the critical demand of the grid resolution in turbulence simulations, procedures for solving large linear algebraic systems of equations are at the heart of large-scale computations. A flow solver developed for LES and DNS has to be highly efficient, reliable and robust to solve the large-scale linear algebraic equation system.

So far, one of the most popular schemes for solving the unsteady incompressible N–S equations is the fractional step method or projection method, first proposed by Chorin [1] and Temem [2] and later successfully applied to the simulation of unsteady flow problems by Kim and Moin [3]. As indicated in Kim and Moin's work, the application of the fractional-step method to the three-dimensional, incompressible, time-dependent N–S equations results in four sets of linear algebraic equations, namely, the three discretized momentum equations representing the momentum balance relations in three spatial directions and the pressure Poisson equation that projects the velocity field, satisfying the momentum relations, into a divergence-free field. Each of the momentum equations and the pressure Poisson equation can generally be written in a discrete form: $a_c\phi_c = \sum_{nb} a_{nb}\phi_{nb} + b_c$, where ϕ is the dependent variable of the three velocity components or the pressure, subscript c represents the centre point and nb stands for the nearby points around the center. As confirmed by a number of numerical experiments, such as [4,5], the properties of the discretized momentum equations are significantly different from those of the pressure Poisson equations. Owing to the stability requirement, the time step in the momentum equations has to be restricted to a small value so that the Courant–Friedrichs–Lewy (CFL) number criterion can be satisfied, typically requiring $CFL \leqslant 1$ if the scheme is not fully implicit. This time-step restriction makes the matrices of the discretized momentum equations highly diagonally dominant, i.e. $a_c \gg \sum_{nb} a_{nb}$. In addition to this, Dirichlet boundary conditions are generally used for the momentum equations, which tends to enhance the diagonal dominance in the momentum equation matrices. When compared to the discretized momentum equations, the pressure Poisson equations tend to be very stiff and ill-conditioned, i.e. $a_c \cong \sum_{nb} a_{nb}$. In addition, Neumann boundary conditions are usually applied to the pressure Poisson equation, which is equivalent to specifying the first derivative on the boundary, and tends not to enhance the diagonal dominance of the equations. Because of these reasons, solving the pressure Poisson equation is usually the CPU bottle-neck for the unsteady incompressible N–S equation system.

A variety of progresses and developments have been made in applying multigrid methods to efficiently solve the Navier–Stokes equations since Brandt [6] first introduced the multigrid method to solve large system of linear algebraic equations. Drikakis et al. [7] developed a nonlinear multigrid method for solving the three-dimensional incompressible N–S equations in conjunction with the artificial compressibility formulation. The method was based on the full multigrid (FMG)–full approximation storage (FAS) algorithm and V-cycles were used in the multigrid iterations. Their testing results demonstrated that the nonlinear multigrid algorithm offered a significant acceleration of the computations in comparison with single-grid and mesh-sequencing algorithms. Subsequently, Drikakis et al. [8] developed the adaptive-smoothing (AS) procedure to accelerate the multigrid computations based on the scheme presented in [7]. The principle of AS is to exploit the non-uniform convergence behavior of the numerical solution during the iterations to reduce the size of the computational domain and, subsequently, to reduce the total computing time. Numerical experiments indicated that the AS procedure performed better for external flows when it is applied in all grid levels of the multigrid method, while for internal flows the best performance is achieved when AS is applied in the finest grid only. Other recent contributions in the field of multigrid methods include the

research by Elman et al. [9] who developed a parallel block multilevel preconditioner for three-dimensional incompressible N–S equations. The parallel preconditioner was based on a block factorization of the coefficient matrix generated in an Oseen nonlinear iteration for the primitive variable formulation. This technique was applied to large-scale, parallel, three-dimensional transient and steady-state simulations employing algebraic multigrid (AMG) methods.

In the current paper, a high-performance unsteady incompressible N–S solver is designed using both conventional and modified tri-diagonal matrix algorithms (TDMAs) and a flexible-cycle [6] additive-correction multigrid [10] (FCAC-MG) technique. The unsteady incompressible N–S solver discussed here makes use of the fractional step method [3] and finite volume discretization over a staggered Cartesian grid. The major advantage of flexible cycle over the conventional V-cycle or W-cycle is that the computation on a given grid level always have the opportunity to move up or to go down one grid level, depending on whether the residual reduction on the current grid level is satisfied. This flexibility makes flexible-cycle scheme more efficient in the corrections between the fine and coarse grids. On the other hand, additive-correction scheme enables the integral conservation property being preserved on all multigrid levels. Therefore, as a combination of the two solving strategies, the FCAC-MG acceleration technique is highly efficient, reliable and robust, which makes it feasible for CPU-intensive computations, such as large eddy simulation (LES) and direct numerical simulation (DNS). The high efficiency and the robustness of the solver are achieved through developing the FCAC-MG acceleration technique as well as flow-physics oriented solving strategies. The flow solver based on the FCAC-MG technique is applied to both temporal and spatial turbulence simulations using LES. The residuals of the large-scale algebraic equation system, typically over a hundred thousand equations, are guaranteed to be continuously driven down to the level of the computer machine round-off error on each time marching step, which warrants strong conservations of the mass and momentum equations over all the control volumes. In addition, the flow solver developed in current research possesses the potential capability for handling complex geometry flows since the performance of the solver does not rely on the use of special type of operations on the large-scale linear algebraic equations system, unlike the cyclic reduction algorithm, but instead it is driven by a flow-physics oriented solving strategy. The solution of the unsteady incompressible N–S equations for flows in a square duct [5], a square annular duct [11] and a confined square coaxial jet [12] are used as examples to demonstrate the advantages of the FCAC-MG technique and the flow-physics oriented solving strategy. The analyses of the simulation results for these three flow configurations provide strong evidence that the flow solver based on FCAC-MG method is capable of capturing the major characteristics of turbulence physics and correctly predicting the relevant turbulent flow phenomena.

2. Mathematical formulation and discretization

The numerical simulations of flow phenomena, using either direct numerical simulation (DNS), large eddy simulation (LES) or a Reynolds-averaged Navier–Stokes (RANS) method, rely on the Navier–Stokes (N–S) equations as the mathematical basis. In the current investigation, incompressible flows are considered and the fluid properties of density and viscosity are assumed to be constants. Therefore, the non-dimensional form of the N–S equations can be written as

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

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_j \bar{u}_i}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{Re}\left\{\frac{\partial}{\partial x_j}(1+\nu_t)\left[\frac{\partial \bar{u}_i}{\partial x_j} + \frac{\partial \bar{u}_j}{\partial x_i}\right]\right\}, \tag{2}$$

where the indices $i, j = 1, 2, 3$ refer to the x, y and z directions, respectively; x is the streamwise direction, and y and z are the transverse directions; t is the time; $\bar{u}_i$ is the spatially filtered velocity component; $\bar{p}$ is the spatially filtered pressure; Re is the Reynolds number; and v_t is the subgrid scale eddy viscosity.

These governing equations are spatially discretized using a second-order finite volume formulation based on a staggered grid system, which proved to be effective and practical in the DNS computation of square duct flow by Gavrilakis [13]. As demonstrated by Kim and Moin [3], a common practice for the temporal discretization of the unsteady incompressible N–S equations is to apply the second-order Adams–Bashforth scheme for the convection terms and the second-order Adams–Moulton scheme for the diffusion terms. The explicit treatment of the nonlinear terms eliminates the need for linearization, and the implicit treatment of the diffusion terms eases the numerical stability restriction. The fractional step method in [3] is used to decouple the pressure and velocity and obtain the time-dependent pressure and the divergence-free velocity. With this approach, the discretized governing equations can then be expressed as

$$\frac{\tilde{u}_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2}\left(3C_i^n - C_i^{n-1}\right) + \frac{1}{2}\left(\tilde{D}_i^{n+1} + D_i^n\right), \tag{3}$$

$$\frac{u_i^{n+1} - \tilde{u}_i^{n+1}}{\Delta t} = -\frac{\partial \phi^{n+1}}{\partial x_i} \quad \text{and} \quad \frac{\partial u_i^{n+1}}{\partial x_i} = 0, \tag{4}$$

where $\tilde{u}_i$ is the intermediate velocity component and u_i^{n+1} is the final velocity field with superscript $n+1$ representing the current time step, and

$$C_i = -\frac{\partial}{\partial x_j}\left(u_i u_j\right) \quad \text{and} \quad D_i = \frac{1}{Re}\frac{\partial}{\partial x_j}\left((1+v_t)\frac{\partial u_i}{\partial x_j}\right).$$

The pressure, p, is linked with the pressure potential, ϕ, through the following formulation:

$$p = \phi - \frac{1}{2}\frac{\Delta t}{Re}\frac{\partial}{\partial x_j}\left[(1+v_t)\frac{\partial \phi}{\partial x_j}\right].$$

These discretization schemes lead to algebraic equations that can be expressed in a unified form of formulation for both the momentum and pressure Poisson equations:

$$a_{ijk}^p \phi_{ijk}^{n+1} = a_{ijk}^e \phi_{i+1jk}^{n+1} + a_{ijk}^w \phi_{i-1jk}^{n+1} + a_{ijk}^n \phi_{ij+1k}^{n+1} + a_{ijk}^s \phi_{ij-1k}^{n+1} + a_{ijk}^t \phi_{ijk+1}^{n+1} + a_{ijk}^b \phi_{ijk-1}^{n+1} + b_{ijk}, \tag{5}$$

where i, j, k denote the cell numbers in three spatial directions x, y, z, respectively.

3. Solver performance for the momentum equations

For the momentum equations, the central coefficient in the discretized equation (Eq. (5)) usually satisfies $a_{ijk}^p = 1/\Delta t + a_{ijk}^e + a_{ijk}^w + a_{ijk}^n + a_{ijk}^s + a_{ijk}^t + a_{ijk}^b$. Since Δt has to be kept at a small value so that the CFL number criterion can be satisfied for the stability requirement, the system of equations are highly diagonally dominate. As illustrated in the following cases, the residual levels of the momentum equations can be driven down to the level of the computer machine round-off error within a few iterations using a TDMA iterative scheme, and no special convergence acceleration technique is required.

3.1. Design of a modified TDMA solver for the momentum equations

It is well known that the conventional TDMA or Thomas algorithm [14] directly solves the discretized equations in one dimension and can be applied iteratively, in a line-by-line fashion, in the other two

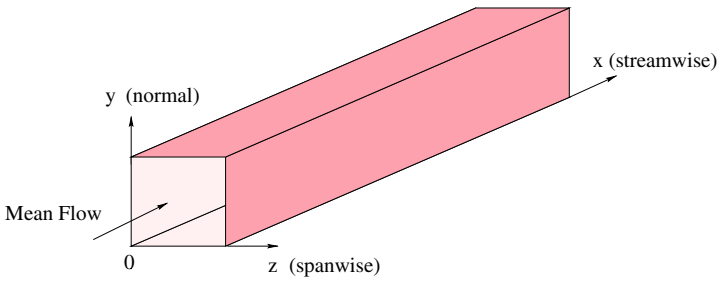


Fig. 1. Schematic and geometric notation of the square duct.

dimensions to solve multi-dimensional problems. When compared against direct solving methods, the conventional TDMA is computationally inexpensive and has the advantage of using the minimum amount of memory storage. So far, the conventional TDMA solver has been used for a wide variety of applications in spatial flow simulations in which the discretized equations (Eq. (5)) can be reduced to a tri-diagonal form in each of the three dimensions of space.

For temporal flow simulations, such as the flows inside a square duct and a square annular duct, the wall boundary conditions are applied in the two cross-streamwise spatial directions y and z, shown in Figs. 1 and 2. This results in the conventional tri-diagonal form of equations along these two directions, as illustrated by Eq. (6). These systems of equations can be solved by the procedures of forward elimination and backward substitution, the details of which can be found in [15].

$$\begin{Bmatrix} a^0_{c2} & a^0_{r2} & 0 & 0 & 0 & 0 & 0 & 0 \\ a^0_{l3} & a^0_{c3} & a^0_{r3} & 0 & 0 & 0 & 0 & 0 \\ 0 & a^0_{l4} & a^0_{c4} & a^0_{r4} & 0 & 0 & 0 & 0 \\ 0 & 0 & a^0_{l5} & a^0_{c5} & a^0_{r5} & 0 & 0 & 0 \\ 0 & 0 & 0 & a^0_{l6} & a^0_{c6} & a^0_{r6} & 0 & 0 \\ 0 & 0 & 0 & 0 & a^0_{l7} & a^0_{c7} & a^0_{r7} & 0 \\ 0 & 0 & 0 & 0 & 0 & a^0_{l8} & a^0_{c8} & a^0_{r8} \\ 0 & 0 & 0 & 0 & 0 & 0 & a^0_{l9} & a^0_{c9} \end{Bmatrix} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b^0_2 \\ b^0_3 \\ b^0_4 \\ b^0_5 \\ b^0_6 \\ b^0_7 \\ b^0_8 \\ b^0_9 \end{Bmatrix}, \tag{6}$$

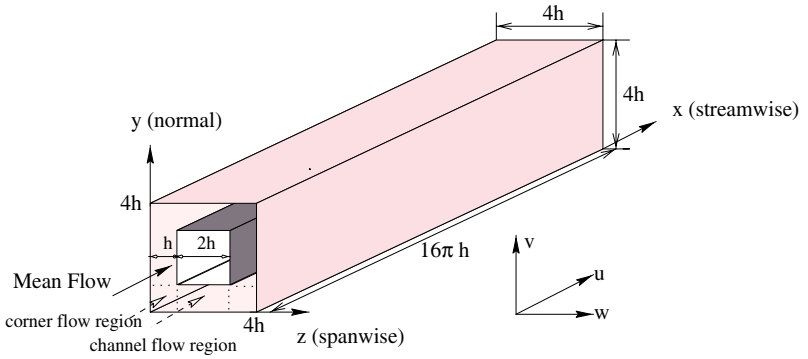


Fig. 2. Schematic and geometric notation of the square annular duct.

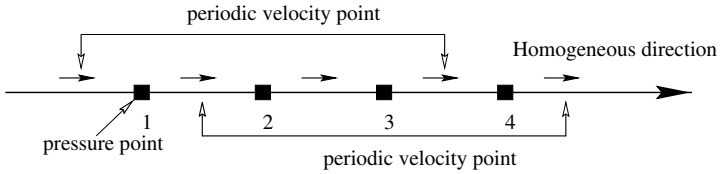


Fig. 3. Implementation of the velocity periodic boundary conditions on a one-dimensional staggered grid.

where the subscripts l, c, r represent the left, center and right coefficients, respectively, and the superscript denotes the operation number with 0 indicating the original coefficients without performing any operation.

In the streamwise direction (the x direction indicated in Figs. 1 and 2), however, periodic velocity boundary conditions are applied because of the homogeneous turbulence assumption. This implementation is schematically illustrated in Fig. 3 for a staggered grid arrangement and can be mathematically expressed as $\phi_{i_b} = \phi_{i_e-1}$ and $\phi_{i_e} = \phi_{i_b+1}$. The nodal points are in the range of $i = i_b \cdots i_e$ with i_b and i_e being the periodic boundary points. The number of equations is $n = i_e - i_b - 1$. Here, for instance, $i_b = 1$ and $i_e = 10$ are used in Eq. (7). The implementation of these velocity boundary conditions results in a modified tri-diagonal form of equations.

$$\begin{Bmatrix} a^0_{c2} & a^0_{r2} & 0 & 0 & 0 & 0 & 0 & a^0_{l2} \\ a^0_{l3} & a^0_{c3} & a^0_{r3} & 0 & 0 & 0 & 0 & 0 \\ 0 & a^0_{l4} & a^0_{c4} & a^0_{r4} & 0 & 0 & 0 & 0 \\ 0 & 0 & a^0_{l5} & a^0_{c5} & a^0_{r5} & 0 & 0 & 0 \\ 0 & 0 & 0 & a^0_{l6} & a^0_{c6} & a^0_{r6} & 0 & 0 \\ 0 & 0 & 0 & 0 & a^0_{l7} & a^0_{c7} & a^0_{r7} & 0 \\ 0 & 0 & 0 & 0 & 0 & a^0_{l8} & a^0_{c8} & a^0_{r8} \\ a^0_{l9} & 0 & 0 & 0 & 0 & 0 & a^0_{l9} & a^0_{c9} \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b^0_2 \\ b^0_3 \\ b^0_4 \\ b^0_5 \\ b^0_6 \\ b^0_7 \\ b^0_8 \\ b^0_9 \end{Bmatrix}. \tag{7}$$

The key issue here is to find a solution procedure that can inverse the modified tri-diagonal matrix in Eq. (7) at a cost of o(N) number of operations with N being the number of equations. With the sparseness structure given by Eq. (7), the solution procedure for the system of equations is conceptually designed based on the Gaussian elimination method, which consists of forward elimination and backward substitution similar to a conventional tri-diagonal equation system. The detailed solving procedure of Eq. (7) is provided in Appendix I of this paper.

3.2. Convergence performance for the momentum equations

3.2.1. Temporal LES of a square duct flow

The turbulent flow inside a square duct, as illustrated in Fig. 1, is a classic flow problem that has received a great deal of coverage both experimentally and numerically. The LES results obtained using the current code were validated in [5,11]. The focus in this section is directed at the solution schemes and the convergence performance of the solver and some detailed computational results will be provided in Section 5. The number of control volumes used in the LES was set at $N_x \times N_y \times N_z = 128 \times 32 \times 32$ in the three spatial directions. Fig. 4 presents the typical convergence history for the momentum equation of U component. The convergence behaviors for the other two components (V and W components) are similar to that of U component presented in Fig. 4. The residual, or more precisely the total residual, hereafter is defined as the sum of the residuals on individual control volumes, i.e. $r = \sum_{n=1}^{N} |a_c\phi_c - \sum_{nb} a_{nb}\phi_{nb} + b_c|$ with N

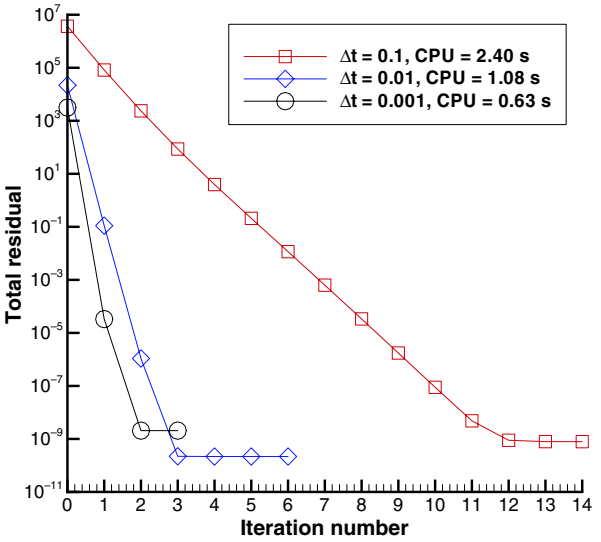


Fig. 4. Residual convergence histories of the momentum u-equation for the temporal LES of a square duct flow. The CPU time were measured on a Dell Precision 530 system with dual 2.2 GHz processors.

being the total number of control volumes. Since double digital precision is used in the current code, the effective digital number that can be trusted in the simulation is about sixteen. Since the total number of the control volumes is on the order of 10^5 (131,072), the solution reaches the level of the computer machine round-off error when the total residual of the discretized equations is driven down by about 11 orders of magnitude, i.e. o(10^{-11}). The residual convergence history is highly dependent on the value of non-dimensional time step, Δt, which directly affects the diagonal dominance of the matrices of the momentum equations. Three typical time steps, namely $\Delta t = 0.1, 0.01, 0.001$, were chosen to demonstrate the convergence performance. These three time steps correspond to *CFL* numbers of 155, 5.91 and 0.47. Because of the stability requirement described in [4], the time step satisfying $\Delta t \leqslant 0.001$ was used for the temporal LES of square duct flow, which guaranteed a *CFL* number less than unity. The residual drop rate was fairly high when $CFL \leqslant 1$, usually o(10^{6-7}) for the first TDMA sweeping and o(10^{4-5}) for the second TDMA iteration, and no special acceleration technique was required to obtain such a large residual drop rate. The total CPU time required to solve the three momentum equations on one time step is at 2.11 s for $\Delta t = 0.001$. This performance was measured on a Dell Precision 530 system with dual CPUs, which was equipped with Pentium 4 Xeon Chips with a 2.2-GHz clock speed and a cache size of 512 KB. The machine had 2.0 GB of RAM memory and a 400 MHz front side bus (FSB).

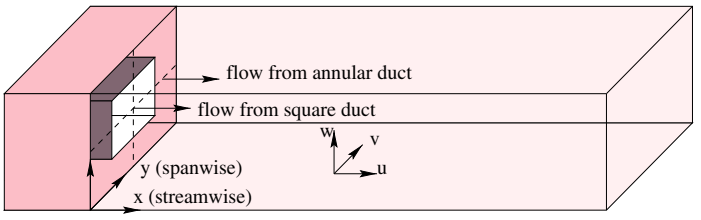


Fig. 5. Schematic and geometric notation of the confined square coaxial jet.

3.2.2. Temporal LES of a square annular duct flow

In [11], the LES method was first applied to the turbulent flow in a square annular duct, as depicted in Fig. 2, which is an extension to the simple square duct problem depicted in Fig. 1. The significance of the investigation lies not only in scientific curiosity about the turbulent structure evolution for the new flow configuration, but also in establishing a turbulent database to extract the fully developed turbulent information that can be used for the inlet boundary conditions to perform a spatial LES of a confined square coaxial jet system [12], shown in Fig. 5. The detailed flow physics analyses for square annular duct can be found in [11]. The attention here is paid to the convergence performance of the solver and some important flow simulation results will be provided in Section 5. The grid resolution used in [11] was set at $N_x \times N_y \times N_z = 128 \times 128 \times 128$ in the three spatial directions. Based on the argument used in the above section, the solution reaches the level of the computer machine round-off error when the total residual of the discretized equations are driven down by about ten orders of magnitude, i.e. $o(10^{-10})$, since the computation was performed using double digital precision. A time step of $\Delta t = 0.0005$, which gives a maximum *CFL* number of 0.821, was chosen to demonstrate the convergence performance. The residual convergence histories for the three momentum equations are similar to the convergence behaviors presented in Fig. 4. The total amount of CPU time required to obtain the solutions to the three momentum equations was 27.60 s for one time step, and the total RAM memory requirement was 0.415 GB, which was benchmarked on the Dell Precision 530 system described above. The modified TDMA solver can efficiently drive the residuals of the momentum equations down to the level of the computer machine round-off error and no special acceleration technique is required.

3.2.3. Spatial LES of a confined square coaxial jet flow

The temporal LES computations of turbulent flows in a square duct and a square annular duct were inspired, to a large extent, by a motivation to understand the spatial evolution of turbulent flow in a confined square coaxial jet, shown in Fig. 5. The objective of the spatial simulation of confined square coaxial jet flow was to gain insight into the turbulent mixing phenomena, particularly when streamwise turbulent shear flow and fully developed turbulence-driven secondary shear flow coexist in one system. Details of the flow physics analyses can be found in [12]. The attention here is focused on the solver convergence performance and some characteristic flow phenomena in confined square coaxial jet will be presented in Section 5. For a spatial simulation, the discretized momentum equations can be reduced to the conventional tri-diagonal form of the equations in each of the three spatial directions. Therefore, the conventional TDMA solver can be applied to solve the discretized momentum equations. The LES computation was conducted at a grid resolution of $N_x \times N_y \times N_z = 128 \times 64 \times 64$, where x was the streamwise direction, and y and z were the two cross-streamwise directions. The non-dimensional time step was set to $\Delta t = 0.0001$ which gave a maximum *CFL* number of 0.527. Since the number of control volumes was on the order of 10^5 and the computation was performed using double digital precision, the solution reached the level of computer machine round-off error when the total residual was driven down by about eleven orders of magnitude, i.e. $o(10^{-11})$. The residual convergence performance for the three momentum equations is similar to the convergence behavior presented in Fig. 4. The conventional TDMA solver was capable of driving the residual down to the level of the computer

Table 1
Conventional and modified TDMA solver performance for the three benchmark cases

	Grid size	RAM memory (GB)	CFL number	Solver	CPU time (s)
Square duct	$128 \times 32 \times 32$	0.033	0.470	Modified TDMA	2.11
Annular duct	$128 \times 128 \times 128$	0.415	0.821	Modified TDMA	27.60
Coaxial jet	$128 \times 64 \times 64$	0.186	0.527	Conventional TDMA	8.34

machine round-off error within a couple of iterations. The total amount of CPU time required to obtain a solution to the momentum equations was about 5.05 s for one time step advancing, which was measured on the Dell Precision 530 system described above.

In summary, the computational parameters and performances of both the conventional and modified TDMA solvers are listed in Table 1 for the three benchmark cases, in which the CPU time represents the computing time per time step for unsteady calculations. The residual convergence, shown in Fig. 4, indicates that the conventional and the modified TDMA solvers described in Section 3.1 are capable of efficiently driving the residuals of the momentum equations down to the level of the computer machine round-off error and that no special acceleration technique is required to achieve this objective.

4. Solver performance for the pressure Poisson equations

For the pressure Poisson equations, the central coefficient in Eq. (5) satisfies $a^p_{ijk} = a^e_{ijk} + a^w_{ijk} + a^n_{ijk} + a^s_{ijk} + a^t_{ijk} + a^b_{ijk}$. Usually, Neumann boundary conditions are applied at the inlet/outlet and wall boundary surfaces. Such a system of equations is very stiff and ill-conditioned. Usually the system involves at least one and, in some cases, two singularity points, as demonstrated in the following section. Therefore, solving the pressure Poisson equation system is the most difficult and CPU time intensive task. Some robust accelerating techniques are necessary to drive the residuals down to a satisfactory level. In the current research, the flexible-cycle algorithm in [6] is combined with the additive-correction multigrid technique in [10], FCAC-MG, to accelerate the rate at which residuals of the pressure Poisson equations are driven down to the level of the computer machine round-off error.

4.1. Design of a modified TDMA solver for the pressure Poisson equations

The periodic boundary conditions for three velocity components are imposed at the inlet and outlet surfaces of the square duct or square annular duct. The schematic implementation of the velocity periodic boundary conditions is demonstrated in Fig. 3 using a one-dimensional staggered grid system. The derivation of the discretized pressure Poisson equation follows the same procedure illustrated in [3]. It should be emphasized here that the sole purpose of the pressure Poisson equation is to project the intermediate velocity field onto a divergence-free velocity field and the derivation of the boundary conditions for the pressure Poisson equation ought to strictly obey this principle. Assuming that the periodic velocity conditions are imposed in the x direction and considering the points adjacent to one surface of the periodic velocity boundary, the discretization of the continuity equation using a central difference scheme gives

$$\frac{1}{\delta x}\left(u^{n+1}_{2,j,k} - u^{n+1}_{1,j,k}\right) + \frac{1}{\delta y}\left(v^{n+1}_{2,j+1,k} - v^{n+1}_{2,j,k}\right) + \frac{1}{\delta z}\left(w^{n+1}_{2,j,k+1} - w^{n+1}_{2,j,k}\right) = 0, \tag{8}$$

where δx, δy, δz are the grid intervals of the corresponding grid points and superscript $n+1$ represents the current time layer.

Using the fractional step method and the periodic velocity conditions in the x direction, it is apparent that

$$u^{n+1}_{2,j,k} = \tilde{u}^{n+1}_{2,j,k} - \Delta t\left(\frac{\partial \phi}{\partial x}\right)_{2,j,k}, \quad u^{n+1}_{1,j,k} = u^{n+1}_{m-1,j,k} = \tilde{u}^{n+1}_{m-1,j,k} - \Delta t\left(\frac{\partial \phi}{\partial x}\right)_{m-1,j,k}, \tag{9}$$

$$v^{n+1}_{2,j+1,k} = \tilde{v}_{2,j+1,k} - \Delta t\left(\frac{\partial \phi}{\partial y}\right)_{2,j+1,k}, \quad v^{n+1}_{2,j,k} = \tilde{v}_{2,j,k} - \Delta t\left(\frac{\partial \phi}{\partial y}\right)_{2,j,k}, \tag{10}$$

$$w_{2,j,k+1}^{n+1} = \tilde{w}_{2,j,k+1} - \Delta t\left(\frac{\partial \phi}{\partial z}\right)_{2,j,k+1}, \quad w_{2,j,k}^{n+1} = \tilde{w}_{2,j,k} - \Delta t\left(\frac{\partial \phi}{\partial z}\right)_{2,j,k}, \tag{11}$$

where $i = 1$ and $i = m$ stand for the grid points of streamwise velocity on the two boundary surfaces in streamwise direction, respectively.

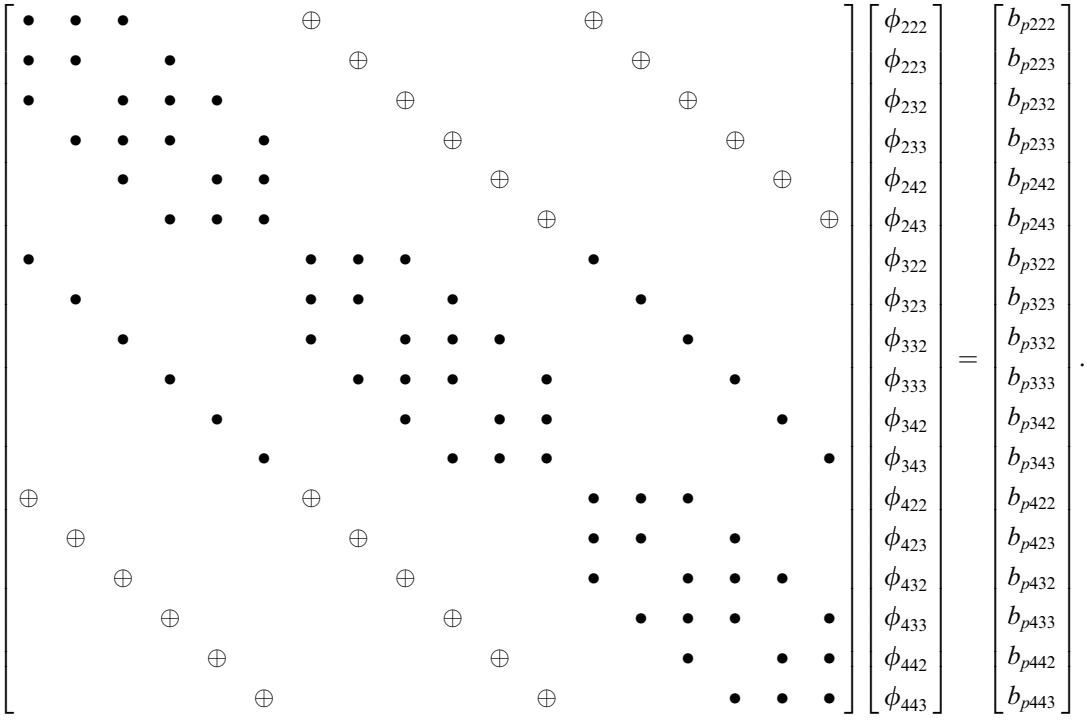
Substituting Eqs. (9)–(11) into Eq. (8) and using central finite differences to approximate the derivatives in Eqs. (9)–(11), it is not difficult to get the discretized pressure Poisson equations at points adjacent to the $i = 1$ surface on which the periodic velocity boundaries are assigned:

$$\begin{aligned} a_{p2,j,k}\phi_{2,j,k} = {} & a_{w2,j,k}\left(\phi_{m-2,j,k} - \phi_{m-1,j,k}\right) + a_{s2,j,k}\phi_{2,j-1,k} + a_{b2,j,k}\phi_{2,j,k-1} a_{e2,j,k}\phi_{3,j,k} + a_{n2,j,k}\phi_{2,j+1,k} \\ & + a_{t2,j,k}\phi_{2,j,k+1} + b_{p2,j,k}, \end{aligned} \tag{12}$$

where p is the center point and w, e, s, n, t, b represent the nodes surrounding point p. Following the same procedure, the discretized pressure Poisson equations for the points adjacent to the $i = m$ surface with the periodic velocity boundaries can be derived as

$$\begin{aligned} a_{pm-1,j,k}\phi_{m-1,j,k} = {} & a_{wm-1,j,k}\phi_{m-2,j,k} + a_{sm-1,j,k}\phi_{m-1,j-1,k} + a_{bm-1,j,k}\phi_{m-1,j,k-1} + a_{em-1,j,k}\left(\phi_{2,j,k} - \phi_{3,j,k}\right) \\ & + a_{nm-1,j,k}\phi_{m-1,j+1,k} + a_{tm-1,j,k}\phi_{m-1,j,k+1} + b_{pm-1,j,k}. \end{aligned} \tag{13}$$

The discretized pressure Poisson equations take a banded matrix form, as illustrated in Eq. (14), in three-dimensional space with the periodic velocity boundary conditions in x or i direction and the wall boundary conditions in the other two directions. Here, a $N_x \times N_y \times N_z = 5 \times 5 \times 4$ (including boundary points) banded sparse matrix is chosen as an example to show the structure of the matrix, with '•' representing the regular non-zero entries and '⊕' represents the non-zero entries caused by the periodic velocity boundary conditions in x direction.



(14)

It is important to note that Eq. (14) can be solved only if the pressure potentials ϕ are fixed at two points. The periodic velocity boundary conditions and the continuity equations implicitly require that the following two relations must hold:

$$\sum_{i=2}^{L-2}\sum_{j=2}^{M-1}\sum_{k=2}^{N-1} b_{p\cdot i\cdot j\cdot k} = 0 \quad \text{and} \quad \sum_{i=3}^{L-1}\sum_{j=2}^{M-1}\sum_{k=2}^{N-1} b_{p\cdot i\cdot j\cdot k} = 0. \tag{15-1}$$

These can be equivalently expressed in integral form

$$\oint\!\!\oint_{\Omega_1} \rho\vec{V}\cdot \mathrm{d}\vec{s} = 0 \quad \text{and} \quad \oint\!\!\oint_{\Omega_2} \rho\vec{V}\cdot \mathrm{d}\vec{s} = 0, \tag{15-2}$$

where $i = 1, 2, \ldots, L$ and $j = 1, 2, \ldots, M$, $k = 1, 2, \ldots, N$ are streamwise and cross-streamwise pressure points, respectively, and $i = 1, L$, $j = 1, M$ and $k = 1, N$ are the boundary points in the three spatial directions; Ω_1 and Ω_2 are the boundary surfaces of the domains defined by the index limits in the two formulations of Eq. (15-1), respectively; and $\rho\vec{V}\cdot \mathrm{d}\vec{s}$ is the mass flux across the boundary surfaces of the domains. Eqs. (15-1) or (15-2) then provide the compatibility for the two solvability conditions for Eq. (14).

To demonstrate the correctness of the above statement, consider the one-dimensional case of a staggered four-point system with the periodic velocity boundary conditions, as presented in Fig. 3. Eq. (14) can be reduced to the following linear equation system, assuming that the grid points are equally spaced:

$$\begin{bmatrix} 1 & -1 & -1 & 1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \end{bmatrix} = \begin{bmatrix} b_2 \\ b_3 \\ b_4 \\ b_5 \end{bmatrix}. \tag{16}$$

It is not difficult, by using Cramer's rule, to see that Eq. (16) becomes positive-definite and solvable only after the values of ϕ_2 and ϕ_5 are fixed. The physical interpretation for this "solvability condition" is that the pressure fluctuation field needs two reference values in order to make the N–S equations with periodic velocity boundary conditions determined. The numerical investigation indicates that the solvability conditions guarantees that the discretized continuity equations are solvable and satisfied to computer round-off precision; that is, it is divergence-free. It is conjectured that the two reference values are associated with the longest wavelength of the pressure fluctuation field in the periodic direction. Therefore, the two reference points ought to be chosen adjacent to the two end-surfaces where the periodic velocity boundary conditions are imposed.

To solve Eq. (14) using the TDMA scheme, the equations in the y and z directions can be reduced to the standard tri-diagonal form and the conventional TDMA approach can be applied. In the streamwise direction (the x direction in Figs. 1 and 2), however, the periodic velocity boundary conditions result in a modified tri-diagonal form of the equation as illustrated in Eq. (17). The nodal points are in the range of $i = i_b \cdots i_e$ and the number of equations is $n = i_e - i_b - 1$. Here, for instance, $i_b = 1$ and $i_e = 10$ are used.

$$\begin{Bmatrix} a^0_{c2} & a^0_{r2} & 0 & 0 & 0 & 0 & a^0_{w22} & a^0_{w12} \\ a^0_{l3} & a^0_{c3} & a^0_{r3} & 0 & 0 & 0 & 0 & 0 \\ 0 & a^0_{l4} & a^0_{c4} & a^0_{r4} & 0 & 0 & 0 & 0 \\ 0 & 0 & a^0_{l5} & a^0_{c5} & a^0_{r5} & 0 & 0 & 0 \\ 0 & 0 & 0 & a^0_{l6} & a^0_{c6} & a^0_{r6} & 0 & 0 \\ 0 & 0 & 0 & 0 & a^0_{l7} & a^0_{c7} & a^0_{r7} & 0 \\ 0 & 0 & 0 & 0 & 0 & a^0_{l8} & a^0_{c8} & a^0_{r8} \\ a^0_{r9} & -a^0_{r9} & 0 & 0 & 0 & 0 & a^0_{l9} & a^0_{c9} \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b^0_2 \\ b^0_3 \\ b^0_4 \\ b^0_5 \\ b^0_6 \\ b^0_7 \\ b^0_8 \\ b^0_9 \end{Bmatrix}, \tag{17}$$

where $a^0_{w12} = a^0_{l2}$, $a^0_{w22} = -a^0_{l2}$, and the superscript of the coefficients represents the operation index, as demonstrated in the matrices in Appendix II. Eq. (17) can be solved using forward elimination and backward substitution following a similar procedure to that described in Section 3.1. The detailed solving procedure of Eq. (17) is presented in Appendix II of this paper.

4.2. Development of a high-performance solver for the pressure Poisson equations

Most flow problems, such as the three flow cases considered in this paper, involve one dominate flow direction, which in most cases is the streamwise direction. The authors' research experience has shown that highly anisotropic coefficients are created for the dominating flow direction. Therefore, the pressure Poisson equations have to be solved directly in the streamwise direction using the solving procedure demonstrated in Appendix II. The FCAC multigrid acceleration technique is applied in the other two directions to solve the equations iteratively. The coupling between the streamwise direction and the cross-streamwise direction is basically handled by using TDMA iterative scheme. Using this solving strategy, the maximum residual convergence rate can be achieved and the residual of the pressure Poisson equation (three-dimensional residual) is guaranteed to be driven down to the level of the computer machine round-off error. This solving method is known as a flow-physics oriented solving strategy.

The additive-correction multigrid scheme described in [10] is used for the two cross-streamwise directions (y and z). The cell-centered two-level multigrid configuration is sketched in Fig. 6, in which the grid point $(i \quad j \quad k)$ on the coarse level is surrounded by four grid points on the fine level in the y and z directions, namely: $(i \quad 2j-2 \quad 2k-2)$, $(i \quad 2j-2 \quad 2k-1)$, $(i \quad 2j-1 \quad 2k-2)$ and $(i \quad 2j-1 \quad 2k-1)$. The variables on the coarse and fine levels are denoted by superscripts c and f, respectively, in the following equations. Based on the idea introduced in [10], the following algebraic equation system can be used to determine the correction for the fine grid level:

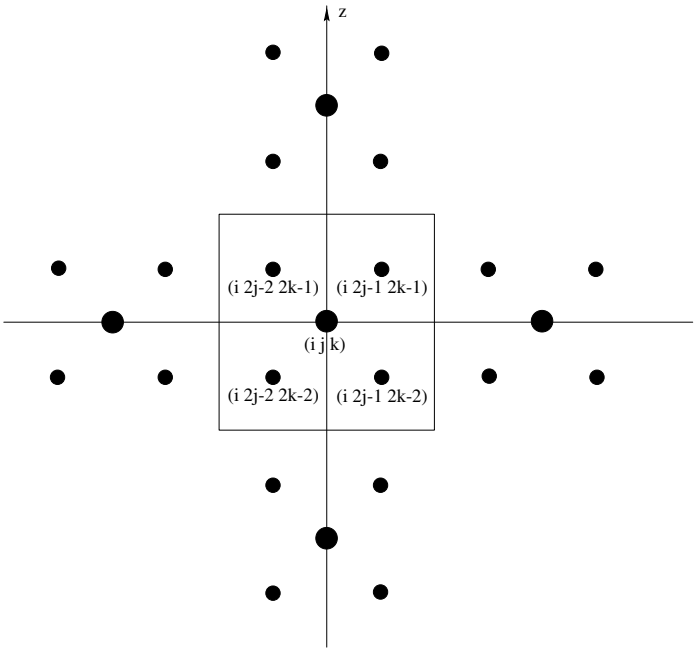


Fig. 6. Schematic of a cell-centered two-level multigrid configuration.

$$a^c_{p\cdot i\cdot j\cdot k}\phi^c_{i\cdot j\cdot k} = a^c_{e\cdot i\cdot j\cdot k}\phi^c_{i+1\cdot j\cdot k} + a^c_{n\cdot i\cdot j\cdot k}\phi^c_{i\cdot j+1\cdot k} + a^c_{t\cdot i\cdot j\cdot k}\phi^c_{i\cdot j\cdot k+1} + a^c_{w\cdot i\cdot j\cdot k}\phi^c_{i-1\cdot j\cdot k} + a^c_{s\cdot i\cdot j\cdot k}\phi^c_{i\cdot j-1\cdot k} + a^c_{b\cdot i\cdot j\cdot k}\phi^c_{i\cdot j\cdot k-1} + b^c_{i\cdot j\cdot k}, \tag{18}$$

where the coefficients on the coarse grid are calculated from the following restriction formulae:

$$\begin{aligned}
a^c_{e\cdot i\cdot j\cdot k} &= a^f_{e\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{e\cdot i\cdot 2j-2\cdot 2k-1} + a^f_{e\cdot i\cdot 2j-1\cdot 2k-2} + a^f_{e\cdot i\cdot 2j-1\cdot 2k-1},\\
a^c_{w\cdot i\cdot j\cdot k} &= a^f_{w\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{w\cdot i\cdot 2j-2\cdot 2k-1} + a^f_{w\cdot i\cdot 2j-1\cdot 2k-2} + a^f_{w\cdot i\cdot 2j-1\cdot 2k-1},\\
a^c_{n\cdot i\cdot j\cdot k} &= a^f_{n\cdot i\cdot 2j-1\cdot 2k-2} + a^f_{n\cdot i\cdot 2j-1\cdot 2k-1},\\
a^c_{s\cdot i\cdot j\cdot k} &= a^f_{s\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{s\cdot i\cdot 2j-2\cdot 2k-1},\\
a^c_{t\cdot i\cdot j\cdot k} &= a^f_{t\cdot i\cdot 2j-2\cdot 2k-1} + a^f_{t\cdot i\cdot 2j-1\cdot 2k-1},\\
a^c_{b\cdot i\cdot j\cdot k} &= a^f_{b\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{b\cdot i\cdot 2j-1\cdot 2k-2},\\
a^c_{p\cdot i\cdot j\cdot k} &= a^f_{p\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{p\cdot i\cdot 2j-1\cdot 2k-2} + a^f_{p\cdot i\cdot 2j-2\cdot 2k-1} + a^f_{p\cdot i\cdot 2j-1\cdot 2k-1} - \left(a^f_{s\cdot i\cdot 2j-1\cdot 2k-2} + a^f_{s\cdot i\cdot 2j-1\cdot 2k-1}\right)\\
&\quad - \left(a^f_{n\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{n\cdot i\cdot 2j-2\cdot 2k-1}\right) - \left(a^f_{b\cdot i\cdot 2j-2\cdot 2k-1} + a^f_{b\cdot i\cdot 2j-1\cdot 2k-1}\right) - \left(a^f_{t\cdot i\cdot 2j-2\cdot 2k-2} + a^f_{t\cdot i\cdot 2j-1\cdot 2k-2)}\right),\\
b^c_{p\cdot i\cdot j\cdot k} &= r^f_{i\cdot 2j-2\cdot 2k-2} + r^f_{i\cdot 2j-2\cdot 2k-1} + r^f_{i\cdot 2j-1\cdot 2k-2} + r^f_{i\cdot 2j-1\cdot 2k-1}.
\end{aligned} \tag{19}$$

Eq. (19) provides the restriction operations that are used in the multigrid cycles. It is not difficult to see, from Eq. (19), that for additive-correction method, there are no further decision to be made related to the treatment of boundary conditions, the transfer of residuals, or the interpolation of the dependent variable. The summation of the equations for all fine-grid control volumes, which lie within the same block of a coarser grid, makes it equivalent to demanding the integral conservation over each coarse-grid control volume.

The residuals on the fine grid level $r^f_{i\cdot j\cdot k}$ are calculated from the current iterative values of $\hat{\phi}_{i\cdot j\cdot k}$ using the following relation:

$$r^f_{i\cdot j\cdot k} = -a^f_{p\cdot i\cdot j\cdot k}\hat{\phi}^f_{i\cdot j\cdot k} + a^f_{e\cdot i\cdot j\cdot k}\hat{\phi}^f_{i+1\cdot j\cdot k} + a^f_{n\cdot i\cdot j\cdot k}\hat{\phi}^f_{i\cdot j+1\cdot k} + a^f_{t\cdot i\cdot j\cdot k}\hat{\phi}^f_{i\cdot j\cdot k+1} + a^f_{w\cdot i\cdot j\cdot k}\hat{\phi}^f_{i-1\cdot j\cdot k} + a^f_{s\cdot i\cdot j\cdot k}\hat{\phi}^f_{i\cdot j-1\cdot k} + a^f_{b\cdot i\cdot j\cdot k}\hat{\phi}^f_{i\cdot j\cdot k-1} + b^f_{i\cdot j\cdot k}. \tag{20}$$

A typical two-level multigrid iterative algorithm consists of restriction, relaxation on the coarse grid and prolongation. After a number of relaxation sweeps, such as TDMA sweeps, on the fine grid level, the residuals are calculated using Eq. (20) and are restricted to the coarse grid using Eq. (19). The restricted residuals are then used as the source terms in Eq. (18) and relaxation sweeps are used to solve Eq. (18) on the coarse level. The solutions to Eq. (18) are then utilized as the corrections to be prolongated back to the fine grid using the following relations to update the current iterative solution of $\hat{\phi}^f_{i\cdot j\cdot k}$:

$$\begin{aligned}
\hat{\phi}^f_{i\cdot 2j-2\cdot 2k-2} &= \hat{\phi}^f_{i\cdot 2j-2\cdot 2k-2} + \phi^c_{i\cdot j\cdot k}, \quad \hat{\phi}^f_{i\cdot 2j-2\cdot 2k-1} = \hat{\phi}^f_{i\cdot 2j-2\cdot 2k-1} + \phi^c_{i\cdot j\cdot k},\\
\hat{\phi}^f_{i\cdot 2j-1\cdot 2k-2} &= \hat{\phi}^f_{i\cdot 2j-1\cdot 2k-2} + \phi^c_{i\cdot j\cdot k}, \quad \hat{\phi}^f_{i\cdot 2j-1\cdot 2k-1} = \hat{\phi}^f_{i\cdot 2j-1\cdot 2k-1} + \phi^c_{i\cdot j\cdot k}.
\end{aligned} \tag{21}$$

Eq. (21) provides the prolongation formulation in the FCAC multigrid calculation. Obviously, the restriction (Eq. (19)) and prolongation (Eq. (21)) are second-order accuracy in space and no extra interpolations are needed.

The flexible cycle in [6] is used to determine when the coarse-grid correction should be employed. The FCAC-MG solution procedure contains two parts. The first part consists of a subroutine of TDMA sweeps, which is controlled by the residual convergence rate. A flowchart of the TDMA solver is depicted

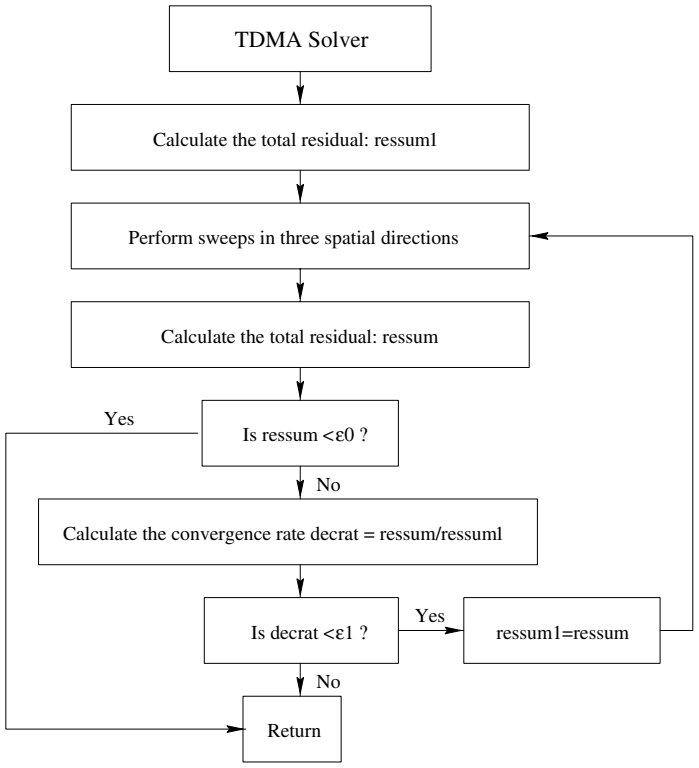


Fig. 7. Flowchart of the TDMA solver controlled by the residual convergence rate.

in Fig. 7. If the total residual before iteration n on the current mesh is $R^n = \sum_{i,j,k} r^n_{i,j,k}$ and the residual after iteration n is R^{n+1}, then another TDMA sweeping iteration is performed if the residual convergence rate satisfies $R^{n+1} \leqslant f \cdot R^n$, where the value for f is usually set to 0.5 [10]. If the convergence rate is lower than 0.5, i.e. $R^{n+1}/R^n > f$, a correction on the coarser grid is required, which invokes the second part of the FCAC-MG solution procedure.

A flowchart for the second part of the FCAC-MG solution procedure on grid level n is provided in Fig. 8. The execution of the solving process is basically controlled by the residue reduction level criterion. The initial total residual on the grid level n is calculated after starting the FCAC-MG algorithm. Following that, the TDMA solver is invoked and the total residual after a number of TDMA iterations is obtained, as shown in Fig. 7. Then, the program determines whether the residual reduction level has been satisfied. If the residual reduction is less than a given criterion $\varepsilon 1$, the FCAC-MG algorithm on grid level n is stopped and the calculation is moved to grid level $n-1$. If the residual reduction level is greater than the given criterion $\varepsilon 1$, the flow information, including the equation coefficients and residuals, will be restricted to grid level $n+1$ using Eq. (19) and the FCAC-MG process is started on grid level $n+1$. The process continues until the residual reduction level on this grid is reduced to the given criterion $\varepsilon 1$. Following that, the solution on grid level $n+1$ is prolongated back to grid level n by Eq. (21) and the TDMA solver on grid level n is invoked until the residual reduction level is satisfied. The flowchart shown in Fig. 8 can be used recursively from the coarsest grid to the finest grid. The residual reduction criterion, $\varepsilon 1$, is usually set to 0.1 to give an adequate convergence that produces an effective correction for the TDMA solver on the current grid level. In order to guarantee that the residual on the finest grid will be driven down to the level of machine round-off error, the restriction of the grid has to reach the

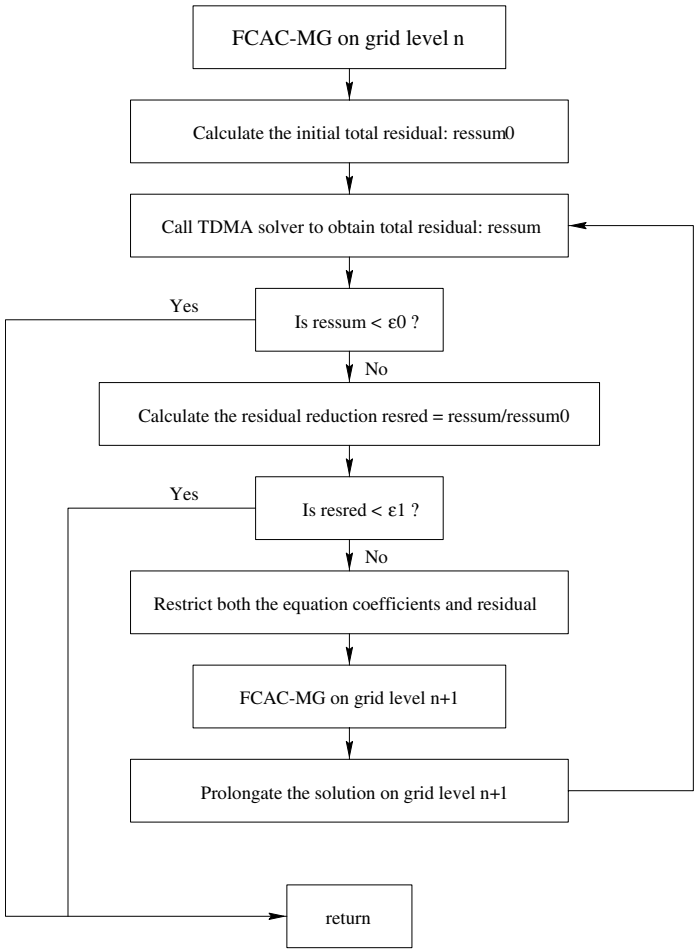


Fig. 8. Flowchart of the FCAC multigrid solver controlled by the residual reduction level.

coarsest grid level and the correction equations on the coarsest grid have to be solved directly. This finding is consistent with the statement in [6] that, on the coarsest grid, an indefinite problem should be solved directly, i.e., not by relaxation of any kind.

The advantage of flexible-cycle scheme is that the calculations on a given grid level always have the opportunity to move up or to go down one grid level, depending on whether the residual reduction on the current grid level is satisfied. In contrast, the conventional V-cycle or W-cycle regulates the multigrid cycle in a prescribed manner. Therefore, the flexible-cycle scheme can provide more efficient corrections between the fine and coarse grids. On the other hand, Eqs. (18)–(21) indicate that the additive-correction scheme establishes the equations on the coarser grid through agglomerating the corresponding coefficients and source terms on the fine grid. As indicated in [10], this procedure reflects the physically desirable property that the integral form of the conservation equation be satisfied over the computational domain on all grid levels. Therefore, as a combination of these two schemes, the FCAC-MG exhibits a reliable, efficient and robust performance when solving the large-scale system of pressure Poisson equations.

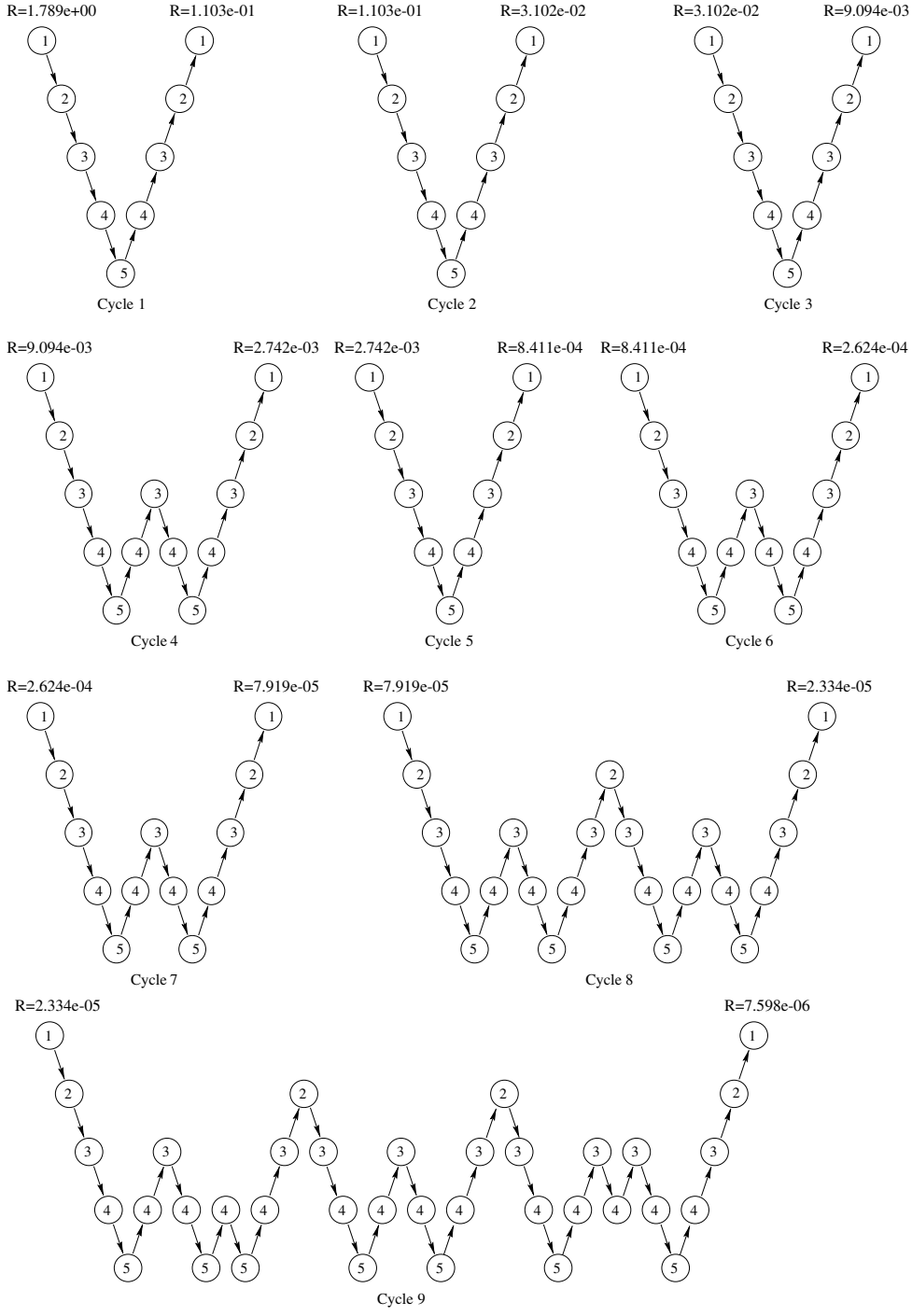


Fig. 9. FCAC-MG solver performance in the LES of a square duct flow.

4.3. Convergence performance for the pressure Poisson equations

For the temporal LES of a square duct flow, illustrated in Fig. 1, the number of control volumes was set to $N_x \times N_y \times N_z = 128 \times 32 \times 32$ in the three spatial directions. The multigrid scheme, as discussed in the previous section, was implemented in the two cross-streamwise directions, which consisted of five grid levels from the finest grid ($128 \times 32 \times 32$) to the coarsest grid ($128 \times 2 \times 2$). As depicted in the FCAC-MG flowchart in Fig. 8, one multigrid cycle started from the point where the program began to invoke the TDMA solver. The cycle ended at the same point after the program went through the multigrid steps of restriction, relaxation and prolongation and then returned to the TDMA solver. Fig. 9 shows the typical first nine multigrid cycles to demonstrate the performance of the FCAC-MG technique when solving the pressure Poisson equation. The multigrid behavior followed the typical V-cycle pattern for the first three cycles, which brought the residual down from $R = 1.789\text{e+}00$ to $R = 9.094\text{e}-03$. From Cycle 4 to Cycle 6, the solver exhibited mixed behavior between V-cycles and W-cycles that drove the residual down from $R = 9.094\text{e}-03$ to $R = 2.624\text{e}-04$. As the multigrid solver progressed further from Cycle 7 to Cycle 9, the coarser grid levels (*grid* – 3, 4, 5) were visited more and more frequently, indicating that more and more low-frequency residuals were smoothed out and the solver was automatically adjusting itself to the smoothing procedure on the coarser grid levels more frequently. It is interesting to note that Cycle 8 followed a double W-cycle performance and Cycle 9 exhibited the so-called multi-W-cycle behavior. The first nine multigrid cycles drove the total residual down from $R = 1.789\text{e+}00$ to $R = 7.598\text{e}-06$. The entire residual convergence history with the FCAC-MG algorithm is depicted in Fig. 10, and the results are compared with the solution procedure that used only the TDMA solver without a multigrid correction. The TDMA solver with the FCAC-MG algorithm was capable of efficiently driving the residual down to the level of the computer machine round-off error within 35 FCAC-MG cycles. The residual was driven down by about twelve orders of magnitude from an initial value of 1.789e+00 to a converged value of 2.46e−12. The CPU time was 24.56 s measured on the Dell Precision 530 system.

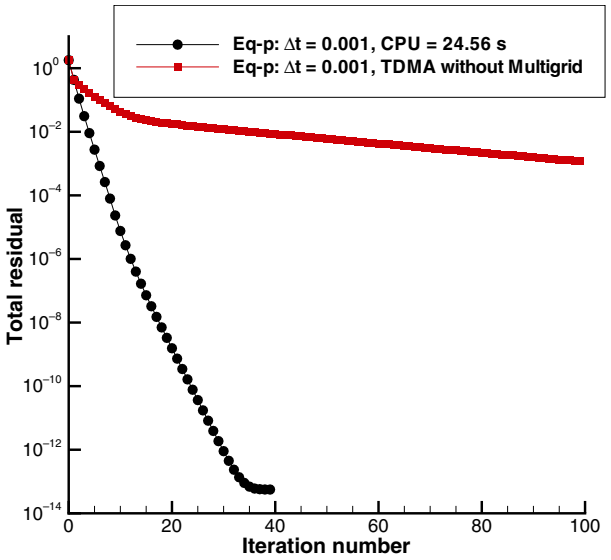


Fig. 10. Residual convergence history of the pressure Poisson equation for the LES in a square duct flow. The CPU time were measured on a Dell Precision 530 system with dual 2.2 GHz processors.

In the temporal LES of a square annular duct flow, illustrated in Fig. 2, the number of control volumes was set to $N_x \times N_y \times N_z = 128 \times 128 \times 128$ in the three spatial directions. The multigrid scheme was implemented in the two cross-streamwise directions and consisted of six grid levels from the finest grid ($128 \times 128 \times 128$) to the coarsest grid ($128 \times 4 \times 4$). Also, a grid dependence study was performed by using a coarser grid with the number of control volumes setting to $N_x \times N_y \times N_z = 128 \times 64 \times 64$. The computation using the coarser grid consisted of five grid levels from the finest grid ($128 \times 64 \times 64$) to the coarsest

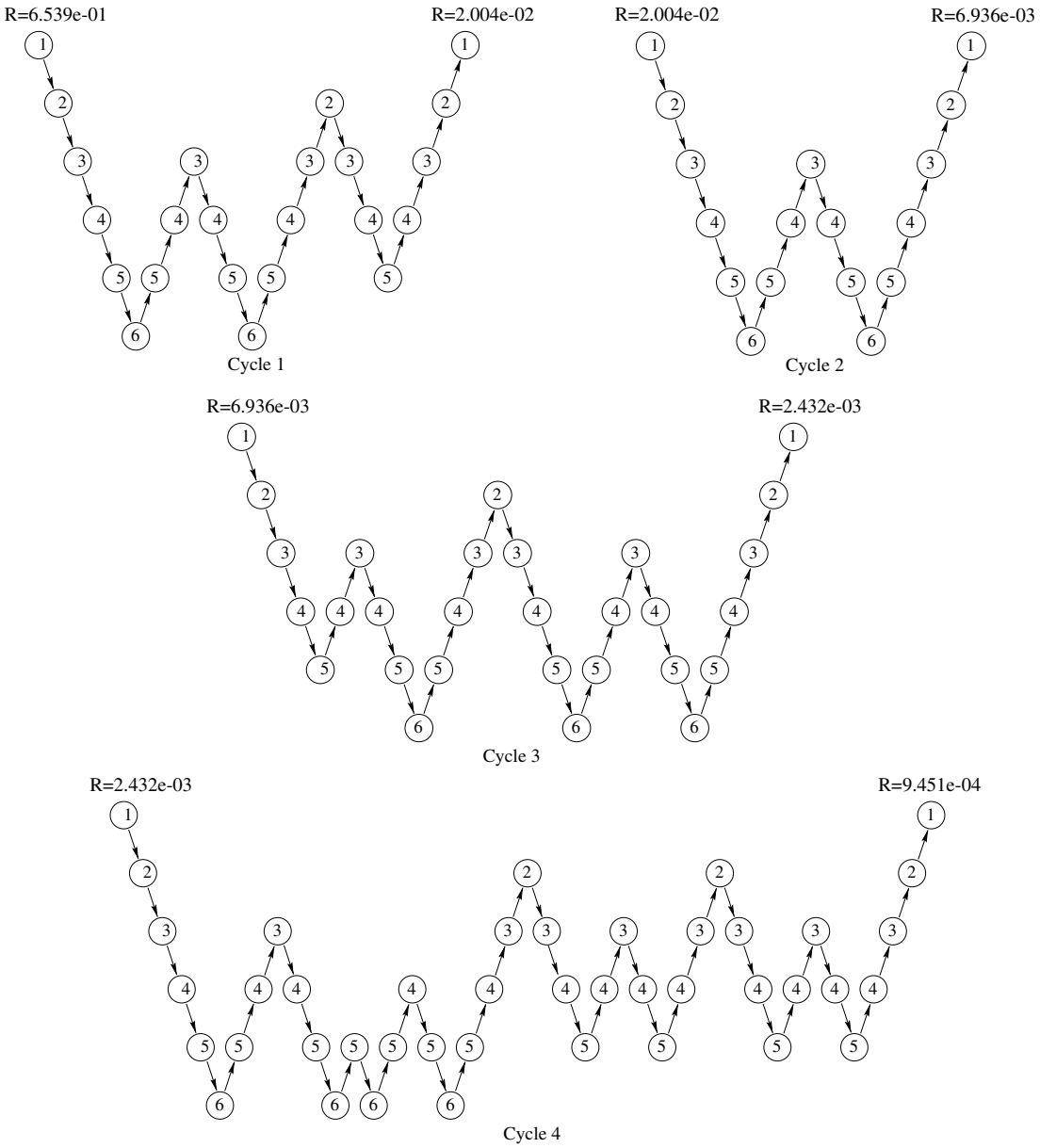


Fig. 11. FCAC-MG solver performance in the LES of a square annular duct flow.

grid ($128 \times 4 \times 4$). The results of the grid dependence study will be presented in Section 5.2. Fig. 11 gives the first four multigrid cycles, which followed irregular W-cycle patterns. The first four multigrid cycles drove the residue down from $R = 6.539e{-}01$ to $R = 9.451e{-}04$. As the multigrid iteration progressed and the residues of the pressure Poisson equation were driven down, the multigrid cycle visited the coarser grid more and more frequently. This observation indicates that, as the residuals were driven down, the solution procedure had to smooth out more and more low- frequency components in the residuals and therefore the solver had to adjust itself to more and more coarser grid relaxations. The flexible-cycle iteration scheme, in this regard, can meet the needs of the solver better compared to the conventional V-cycle and W-cycle. Fig. 12 shows the residual convergence history for the TDMA solver with the FCAC-MG algorithm and the TDMA solver without a multigrid acceleration. The TDMA solver with the FCAC-MG acceleration could efficiently drive the residual down to the level of the computer machine round-off error within 25 FCAC-MG cycles. The residual was driven down about ten orders of magnitude from an initial value of 6.539e−01 to a converged value of 1.54e−11. The CPU time cost for these multigrid cycles was 242.94 s on the Dell Precision 530 system.

In the spatial LES of a confined square coaxial jet, the number of control volumes was set to $N_x \times N_y \times N_z = 128 \times 64 \times 64$ in the three spatial directions. The conventional TDMA solver was used in three spatial directions and the FCAC-MG acceleration technique was applied in the two cross-streamwise directions. The number of multigrid levels was set to 5, from the finest grid of $N_x \times N_y \times N_z = 128 \times 64 \times 64$ to the coarsest grid of $N_x \times N_y \times N_z = 128 \times 4 \times 4$. The performance of the FCAC-MG algorithm in the first three cycles is shown in Fig. 13. The residual of the pressure Poisson equation was driven down from an initial value of 19.47 to an end value of 0.2689. Similar to the performance in the square annular duct flow, the multigrid path exhibited a double W-cycle pattern for the first two multigrid cycles and then changed to an irregular multi-W-cycle pattern in the third multigrid cycle. As the multigrid iteration progressed and the residual became smaller and smaller, the multigrid cycles visited the coarser grid more frequently to smooth out the low frequency components in the resid-

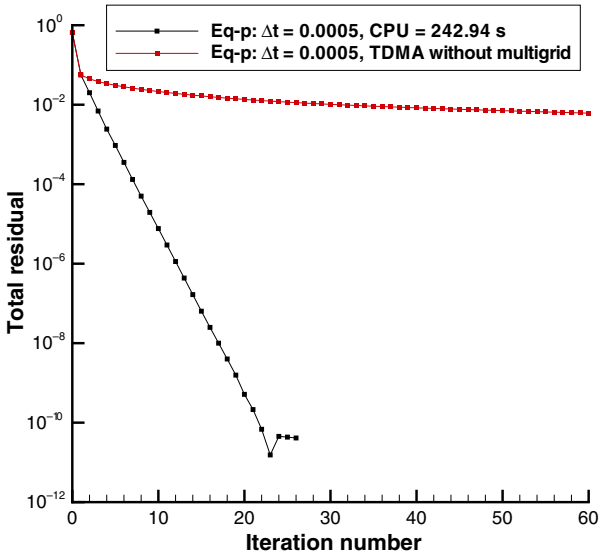


Fig. 12. Residual convergence history of the pressure Poisson equation for the LES in a square annular duct flow. The CPU time were measured on a Dell Precision 530 system with dual 2.2 GHz processors.

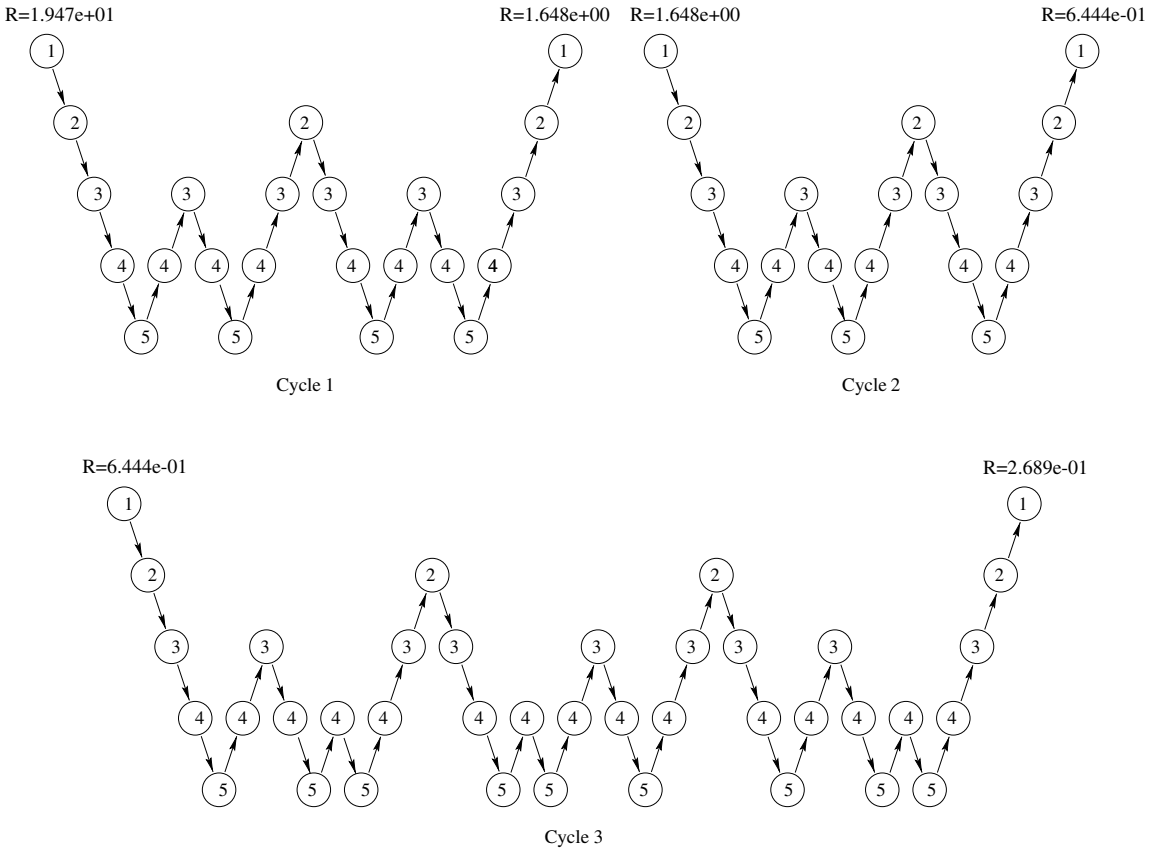


Fig. 13. FCAC-MG solver performance in the LES of a confined square coaxial jet flow.

ual, indicating that the flexible-cycle scheme could automatically adjust itself to meet the needs of the solver. Fig. 14 shows the residual convergence history, which indicated that the residual could be driven down by about twelve orders of magnitude from an initial value of 1.947e+01 to a converged value of 1.93e−11. The CPU time cost for obtaining the converged solution at this residual level was 147.60 s on the Dell Precision 530 system.

Table 2 summarizes the computational parameters and performance of the FCAC-MG solver for the three benchmark cases, consisting of the square duct, square annular duct and confined square coaxial jet flows. In Figs. 10, 12 and 14, the convergence performance of the residuals with the FCAC-MG acceleration were compared with the performance of the solver that used only the TDMA without multigrid. The comparisons indicated that the conventional and modified TDMA solvers coupled with the FCAC-MG acceleration are capable of efficiently driving the residuals of the pressure Poisson equations down to the level of the computer machine round-off error. Compared with the CPU time cost for solving the momentum equations, the CPU time spent solving pressure Poisson equation was usually 10–20 times greater. Therefore, obtaining the solution to pressure Poisson equation was the bottle-neck when solving the unsteady incompressible N–S equations. Here, the CPU time represents the computing time per time step for unsteady calculations.

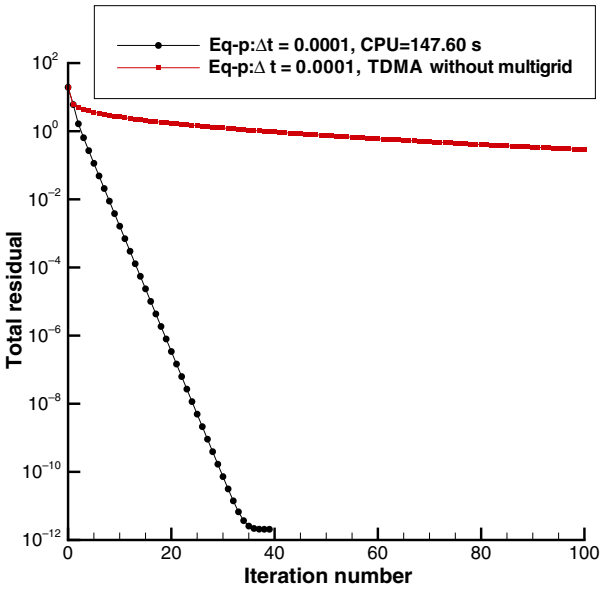


Fig. 14. Residual convergence history of the pressure Poisson equation for the LES in a confined square coaxial flow. The CPU time were measured on a Dell Precision 530 system with dual 2.2 GHz processors.

Table 2
FCAC-MG Solver performance for the three benchmark cases

	Grid size	RAM memory (GB)	CFL number	CPU time (s)
Square duct	$128 \times 32 \times 32$	0.033	0.470	24.56
Annular duct	$128 \times 128 \times 128$	0.415	0.821	242.94
Coaxial jet	$128 \times 64 \times 64$	0.186	0.527	147.60

5. Presentation of results and discussion

The previous sections were focused on the solving strategies and the convergence performance of the FCAC-MG solver when applied to LES of turbulent flows in the square duct, square annular duct and square confined coaxial jet. In the current section, the simulation results of these three cases will be presented to demonstrate the accuracy and correctness of the developed flow solver with regard to revealing the relevant turbulence physics.

5.1. LES of turbulent flow in square duct

The turbulent flow inside a square duct, as illustrated in Fig. 1, is a classic flow problem that has received a great deal of investigations both experimentally and numerically. The solver developed in current study is validated by the DNS data from [13,16] as well as the experimental measurements from [17,18]. Fig. 15 presents the mean streamwise velocity along the wall-bisector of the square duct. The LES results from the current investigation agree well with the DNS results at high Reynolds number and there is about 10% discrepancy when compared with the DNS results at low Reynolds number. This discrepancy might be attributed to the use of the Smagorinsky Subgrid-Scale model in the current LES computation. To further evaluate the quality of the current LES results, the 'law of the wall' relation is used to check the mean

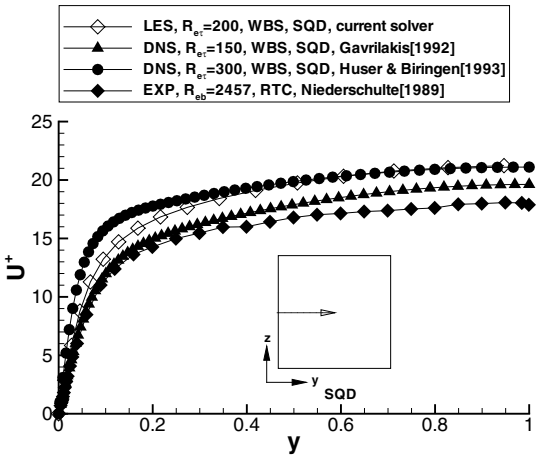


Fig. 15. Mean streamwise velocity along wall-bisector of square duct.

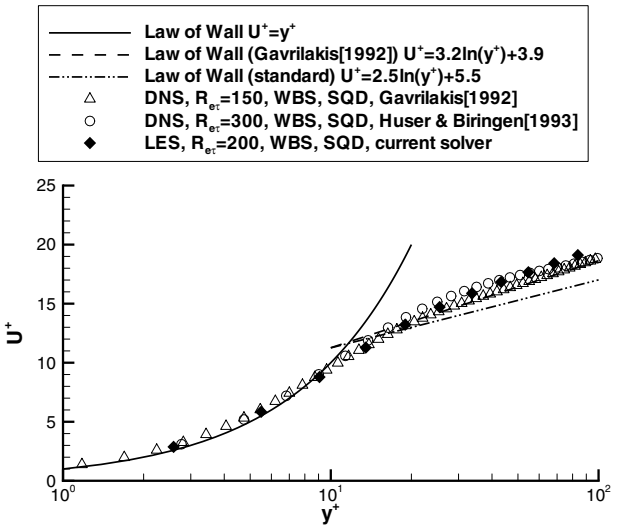


Fig. 16. Logarithm plot of the mean streamwise velocity near wall in square duct.

streamwise velocity behavior in the near wall region. The present LES results, as shown in Fig. 16, are in close agreement with those from [13].

In addition to the mean streamwise velocity, the mean turbulence-driven secondary flow and the turbulence statistics, particularly the Reynolds stresses, are important quantities to judge the quality of the simulation. Fig. 17 presents the mean turbulence-driven secondary flow overlapped by the streamwise velocity isovels in the square duct. As is well recognized by Nikuradse [19], the secondary flow consists of streamwise counter-rotating vortex pair around the corners of the square duct. The secondary flow distorts the isovels towards the corners and increases the corner-bisector momentum transfer. The streamwise turbulence intensity u'_{rms} along wall-bisector of the square duct is presented in Fig. 18. The computed data are compared with the experimental measurements from [17] in rectangular duct and from [18] in plane channel. The DNS data from [13,16] are included. The LES results based on current solver tend to slightly

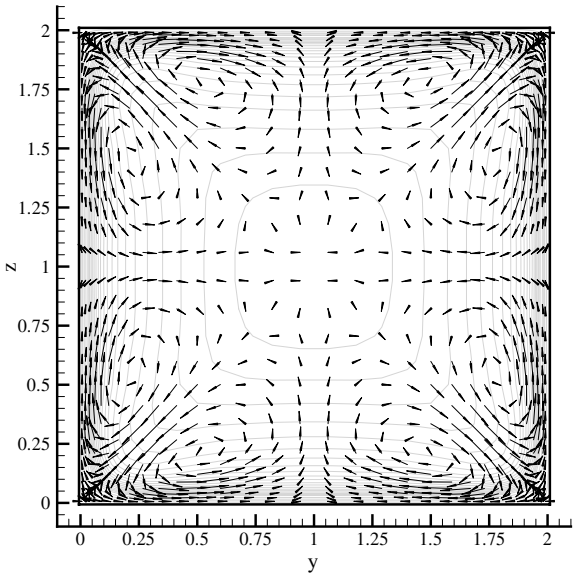


Fig. 17. Mean turbulence-driven secondary flow pattern in square duct.

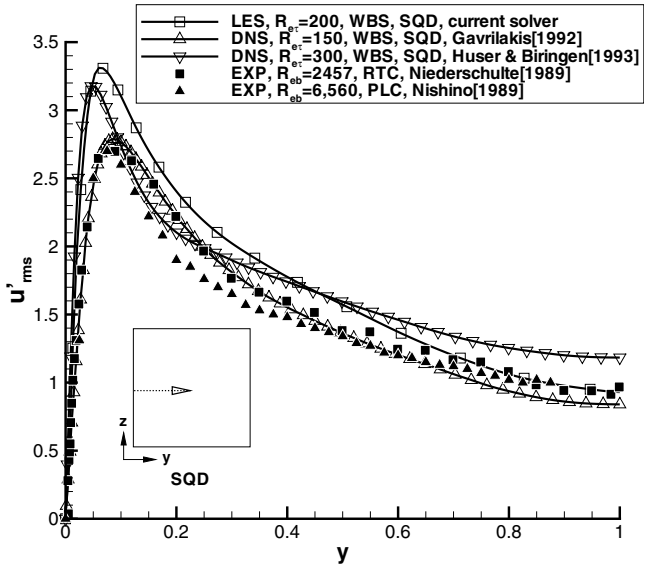


Fig. 18. Streamwise turbulence intensity u'_{rms} along the wall-bisector of square duct.

over-predict u'_{rms} (about 15%) in the near wall region. The peak value location from the current solver is predicted at $y^+ = 13.5$, whereas this location is at $y^+ = 13.0$ in the DNS by Gavrilakis [13] and at $y^+ = 15.0$ in the DNS by Huser and Biringen [16]. Figs. 19 and 20 present, respectively, comparisons of the turbulence intensities, v'_{rms} and w'_{rms}, along wall-bisector line. The present LES results, while under-predicting the magnitudes of v'_{rms} and w'_{rms} by about 20% compared with DNS results from Gavrilakis [13], well duplicate the DNS results in the near wall region and agree well with the lower Reynolds number

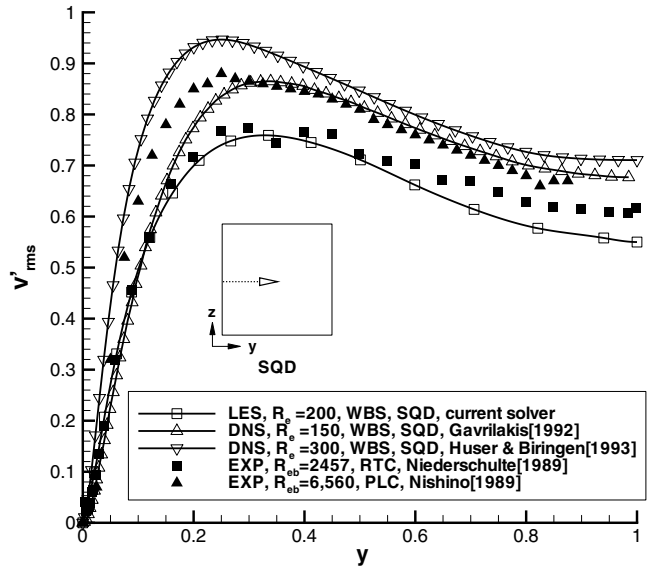


Fig. 19. Cross-streamwise turbulence intensity v'_{rms} along the wall-bisector of square duct.

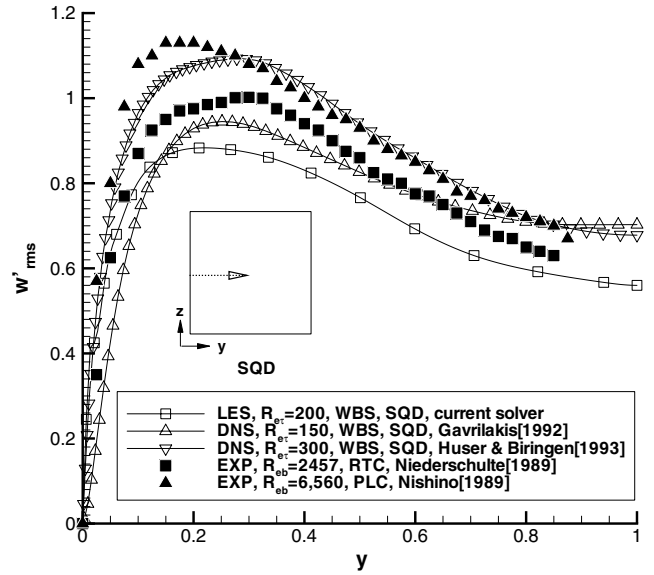


Fig. 20. Cross-streamwise turbulence intensity w'_{rms} along the wall-bisector of square duct.

experimental data. The distribution of turbulent shear stress $-\overline{u'v'}$ along the wall-bisector is presented in Fig. 21. The prediction from the current LES solver is in good agreement with the DNS results from Gavrilakis [13]. Some noticeable difference can be found, by comparing the DNS prediction of Huser and Biringen [16] with the DNS results from [13], which is assumed to be attributed to the difference in Reynolds number. The verification and validation on the case of square duct flow confirms that current LES solver captures most of the turbulence physics in the flow, including turbulence-driven secondary flow and the relevant turbulence statistics.

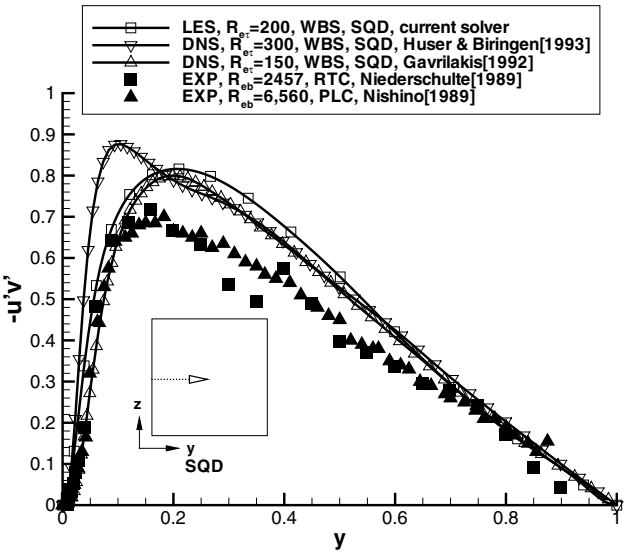


Fig. 21. Turbulence shear stress $-\overline{u'v'}$ along the wall-bisector of square duct.

5.2. LES of turbulent flow in square annular duct

The current LES solver is applied to a square annular duct flow as shown in Fig. 2, which is an extension to a simple square duct in Fig. 1. To the best of our knowledge, there exists no previous LES or DNS investigation for flow in such configuration. This is a complex flow, compared with a square duct, because the square annular duct contains both the concave 90° corners around the outer square duct and convex 90° corners near the inner square duct. The turbulence structures, particularly the turbulence-driven secondary flow structures, are expected to be more complex than those in square duct. Figs. 22(a) and (b) present the mean streamwise velocity contours and the mean turbulence-driven secondary flow, respectively. The contours of the mean streamwise velocity in Fig. 22(a) clearly identify a bulge away from the convex corner of the inner square duct and a bulge towards the concave corner of the outer square duct. Following Nikuradse [19], the bulge away from the convex corner and the bulge towards the concave corner imply the existence of secondary flows pointing away from the convex corner and directed towards the concave corner, respectively. The vector plot of the mean secondary flow in Fig. 22(b) confirms this conjecture, which exhibits the turbulence-driven secondary flow in the square annular duct as a chain of counter-rotating vortex pairs symmetrically placed around the bisector of both concave and convex 90° corners. The formation of the turbulence-driven secondary flow in a square annular duct can be linked to the same mechanisms as those in a square duct, in both cases, the anisotropy of turbulence stresses is the driving force that generates the secondary flows. The streamwise mean vorticity transport equation, Eq. (22), for statistically stationary flow was used by Bradshaw [20] to identify the vorticity generation contributions from the anisotropy distributions of Reynolds shear and normal stresses.

$$U\frac{\partial\bar{\Omega}_x}{\partial x}+V\frac{\partial\bar{\Omega}_x}{\partial y}+W\frac{\partial\bar{\Omega}_x}{\partial z}=\nu\left(\frac{\partial^2\bar{\Omega}_x}{\partial x^2}+\frac{\partial^2\bar{\Omega}_x}{\partial y^2}+\frac{\partial^2\bar{\Omega}_x}{\partial z^2}\right)+\bar{\Omega}_x\frac{\partial U}{\partial x}+\bar{\Omega}_y\frac{\partial V}{\partial y}+\bar{\Omega}_z\frac{\partial W}{\partial z}$$
$$+\underbrace{\left(\frac{\partial^2}{\partial y^2}-\frac{\partial^2}{\partial z^2}\right)\left(-\overline{v'w'}\right)}_{\text{shear-stress-contribution}}+\underbrace{\frac{\partial^2}{\partial y\partial z}\left(\overline{v'^2}-\overline{w'^2}\right)}_{\text{normal-stress-contribution}}. \tag{22}$$

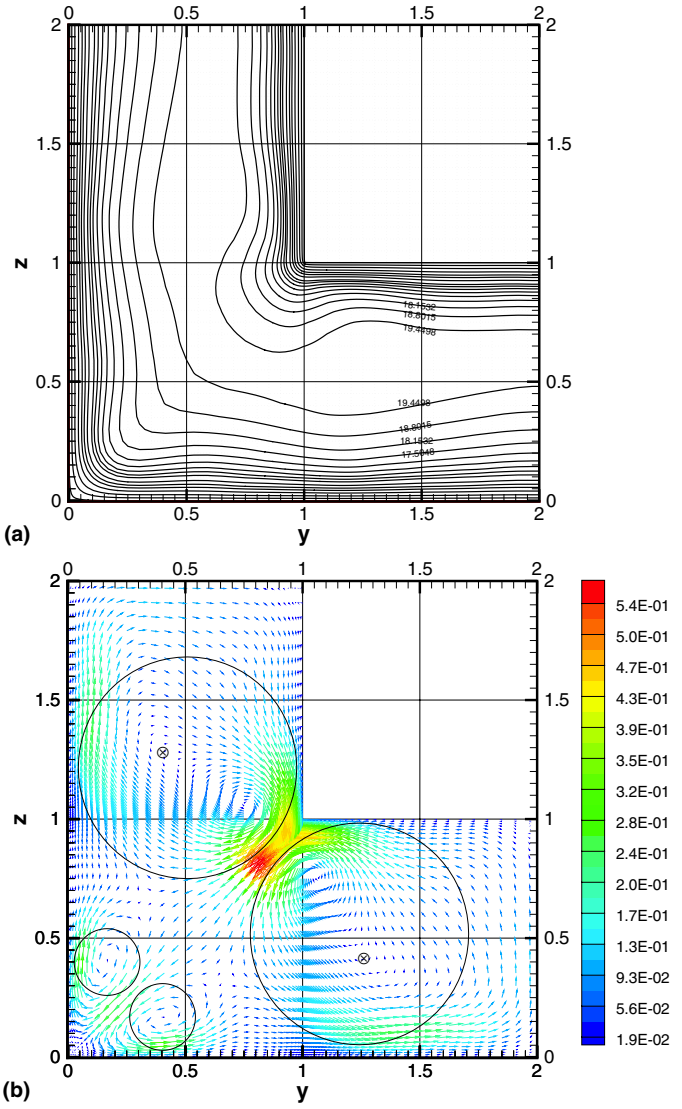


Fig. 22. (a) Mean streamwise velocity contours; (b) mean turbulence-driven secondary flow in square annular duct.

Obviously, Eq. (22) indicates that the Reynolds stress distributions, particularly $-\overline{v'w'}$, $\overline{v'^2}$ and $\overline{w'^2}$, have to be accurately captured so that the patterns of turbulence-driven secondary flows can be predicted correctly. Comparing Fig. 17 with Fig. 22(b), it can be evidently seen that the pattern of the secondary flow around the concave corner of the outer duct resembles what exhibits near the concave corner region of square duct flow. The resemblance of the vorticity structures suggests that the turbulent flow structures, including both mean flow and Reynolds stresses, are strongly flow configuration dependent.

As mentioned in the Section 4.3, a grid dependence study was performed for the LES in the square annular duct flow using two grids resolutions, namely, grid I with $N_x \times N_y \times N_z = 128 \times 64 \times 64$ and grid II with $N_x \times N_y \times N_z = 128 \times 128 \times 128$. The streamwise mean velocity U^+ in the square annular duct is compared to DNS results from a square duct in Figs. 23(a) and (b), which provide comparisons along both wall-bisector and corner-bisector, with horizontal axes normalized by the wall-bisector distance and diagonal

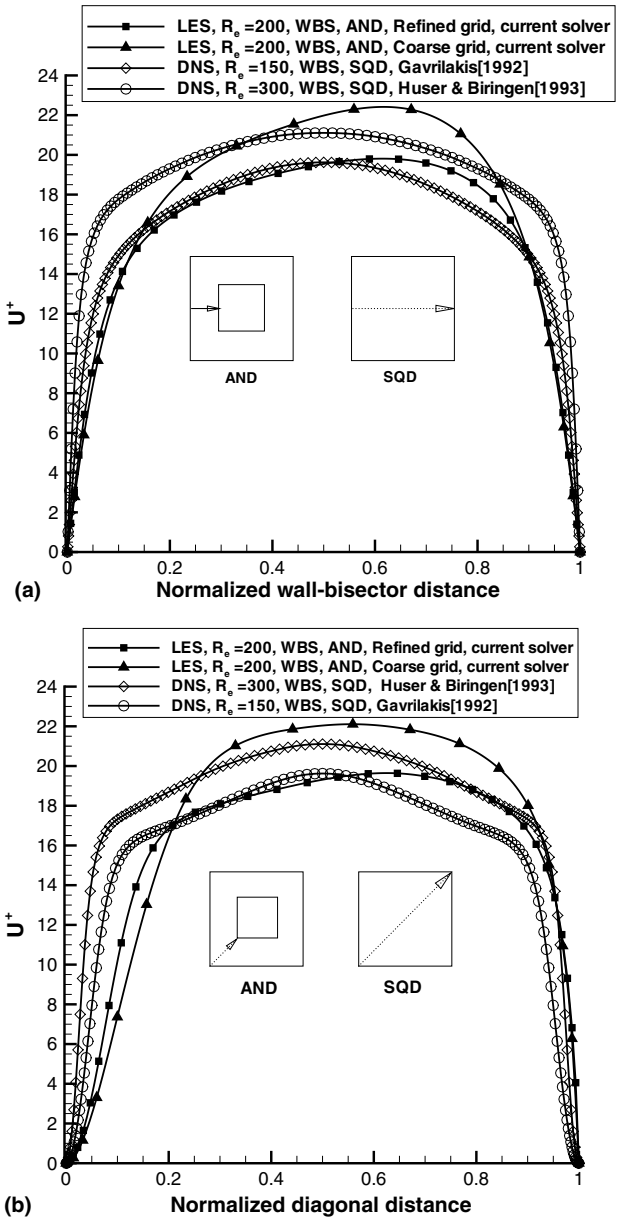


Fig. 23. Mean streamwise velocity along (a) the wall-bisector and (b) the corner-bisector.

distance, respectively. Also, these figures provide the comparisons of the results based on coarser and refined grids. The streamwise mean velocity using coarser grid exhibits an evident coarse-grid diffusion effect, which tends to over-predict U^+ and causes the velocity profile to protrude further in the region away from the wall. The current LES results in square annular duct is performed at $R_{e\tau} = 200$ based on h in Fig. 2, which is closer to $R_{e\tau} = 150$ in the DNS from [13] than $R_{e\tau} = 300$ in the DNS from [16]. Therefore, the LES results using refined grid are more comparable to the U^+ distributions from [13] in terms of both

velocity magnitudes and the near-wall velocity gradients. The current grid-dependence study clearly indicates that the results obtained using the FCAC-MG flow solver consistently converge to more accurate solution as the computational grid becomes more and more refined.

To investigate the near-wall behavior of the streamwise mean velocity, logarithmic-law is applied to check the U^+ distribution in the near-wall region. Fig. 24 presents the U^+ distributions near both inner and outer walls of square annular duct, and they are compared to the DNS results from [13,16] as well as the logarithmic law relation. In the region of $y^+ \leqslant 10$, the U^+ distributions near both inner and outer walls agree well with the DNS results and the linear diffusion relation. In the logarithmic region ($20 \leqslant y^+ \leqslant 100$), the U^+ distribution near the outer wall is closer to the square duct DNS results and the U^+ near the inner wall is higher than the logarithmic law, which is conjectured to be attributed to the higher turbulence production near the inner wall.

The 'law of the wall' is originally used to characterize boundary layer types of flows. It would be of interest to investigate the validity of this law in the vicinity of the two corners in the square annular duct, namely, the convex 90° corner and the concave 90° corner. Fig. 24 provides the distributions of the streamwise mean velocity along the corner-bisectors of both convex and concave corners. These distributions are compared with the traditional 'law of the wall' and the DNS results along the corner-bisector of the concave corner in the square duct. The consistency between the results from LES and DNS in Figs. 24, 25(a) and (b) indicates that the streamwise velocity distribution in the present calculations is not affected by either the implementation of the wall damping function or the subgrid scale (SGS) model, as long as the grid resolution is sufficient. It is evident that the near-wall behavior of the mean streamwise velocity does possess some common characteristics, since the distributions near the concave corner-bisector are almost identical for both square duct and square annular duct. The distributions are also linear in the logarithmic outer region. Therefore, it is conjectured that some universal relations can be constructed, namely, U^+ as a function of d^+, where d^+ is the distance from the corners as indicated in Fig. 25. The concave corner case, by curve-fitting the LES results in the region of $30 \leqslant d^+ \leqslant 100$ in Fig. 25(a), gives a linear logarithmic relation $U^+(d^+) = 2.5 \ln d^+ + 6.5$ in the outer layer. Similarly, use of the Van Driest wall-damping form gives $U^+(d^+) = d^+(1 - e^{-d^+/25})$ in the linear stress inner region, which fits the distribution of U^+ in the region

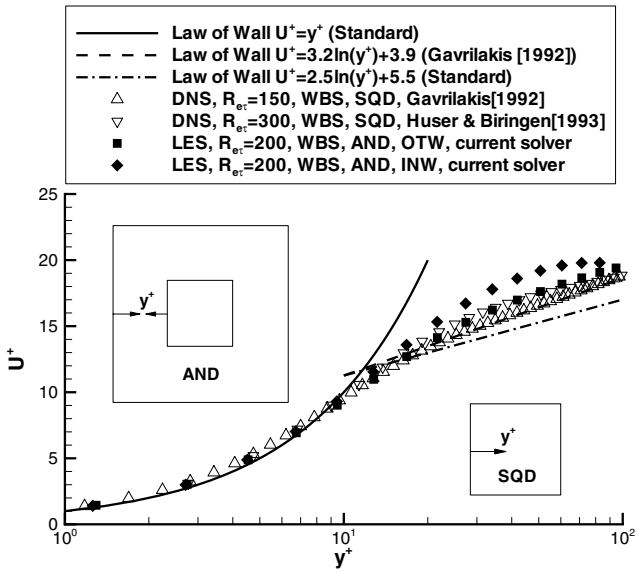


Fig. 24. Mean streamwise velocity along the wall-bisector compared with law of the wall.

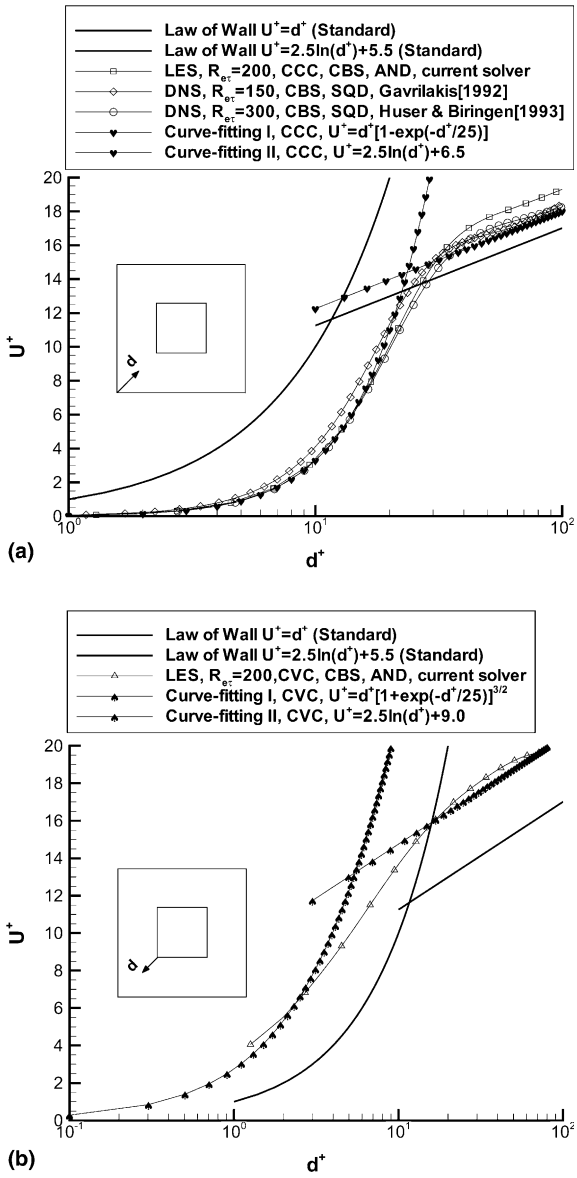


Fig. 25. Logarithm plot of the mean streamwise velocity near (a) the concave 90° and (b) the convex 90° corner-bisectors in square annular duct.

of $0 \leqslant d^+ \leqslant 20$ quite well for both DNS and LES results. For the convex corner, Fig. 25(b) indicates that the current LES grid is somewhat sparse near the convex 90° corner. The grid points need to be distributed well into the region of $d^+ \leqslant 1.0$ to resolve the flow near convex corner. Based on the current LES data, the $U^+ - d^+$ dependence near the convex 90° corner can be curve-fit as $U^+ = d^+(1 + e^{-d^+/25})^{3/2}$ and an outer layer relation $U^+ = 2.5\ln d^+ + 9.0$ is roughly in line with the computed results. The above analysis leads to a conclusion respectively that, the concave 90° corner and the convex 90° corner impose a damping factor of $1 - e^{-d^+/25}$ and an enhancement factor of $(1 + e^{-d^+/25})^{3/2}$ onto the turbulent sub-layer diffusion of the streamwise mean velocity in near corner regions.

The above analyses provide clear evidences that the flow solver based on the FCAC-MG method, applied to temporal simulation of turbulent flow in square annular duct, is capable of accurately capturing the major characteristics of turbulence physics and correctly predicting the relevant turbulent flow phenomena.

5.3. LES of turbulent flow in confined square coaxial jet

In the current investigation, the LES flow solver based on the FCAC-MG is applied to simulate the turbulent flow in a confined square coaxial jet as presented in Fig. 5. The simulation involves two flow

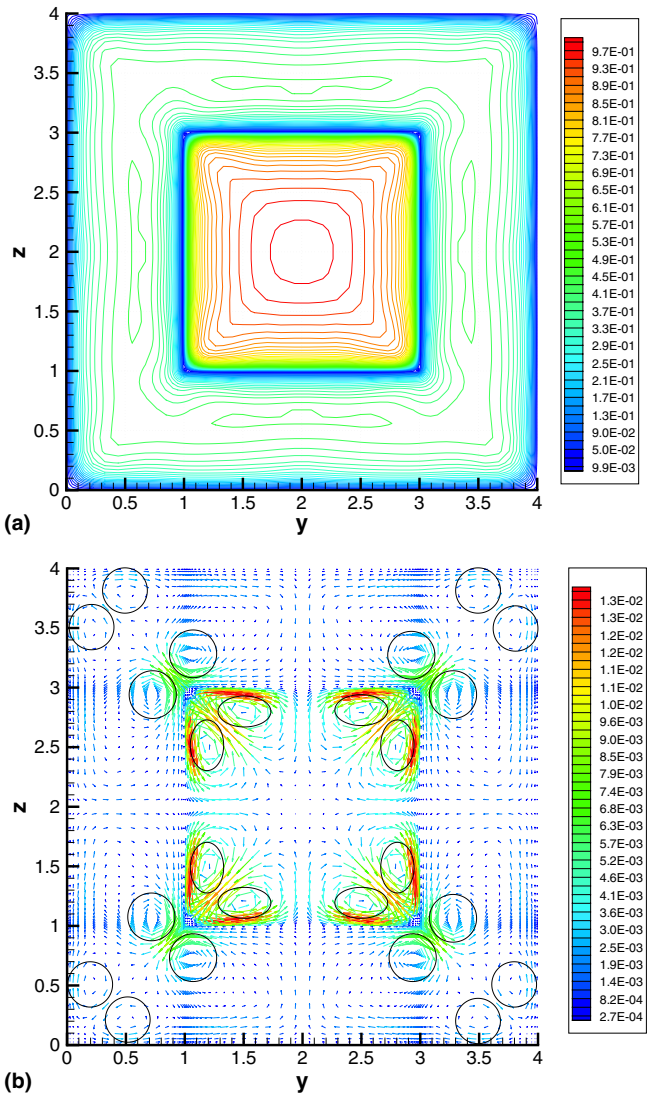


Fig. 26. (a) Mean streamwise velocity contours at inlet colour-mapped by its magnitude U/U_{cl}; (b) mean secondary flow vector-plot at inlet colour-mapped by secondary velocity magnitude $\sqrt{\bar{v}^2 + \bar{w}^2}/U_{cl}$ in confined square coaxial jet.

streams issuing from an inner square duct and a square annular duct into a confined square chamber. The spatial evolution of the flow mixing between the two streams is predicted. One of the most challenging issues in Computational Fluid Dynamics (CFD) is the prescription of turbulent inflow and outflow conditions in the non-homogeneous directions, as commented by Rogallo and Moin [21] and Ciofalo [22]. The inflow condition appears to be more troublesome since the influence of the upstream conditions persists well downstream in most convection-dominated flow cases. A common practice to prescribe the turbulent inflow is to make use of simulation results from a relevant temporal turbulent flow simulation, as seen in [23,24]. The investigation of the flows inside both square and annular ducts, as presented in Sections 5.1 and 5.2, provides an important insight that demonstrates the need for precise specification of the turbulent inflow conditions for the confined square coaxial jet. Figs. 26(a) and (b) present the realistic mean streamwise shear, caused by the streamwise velocity difference, and the secondary shear, induced by the turbulence-driven secondary flows. The Reynolds stresses on the inlet, as demonstrated in Fig. 27 for streamwise turbulent kinetic energy, present a highly anisotropic and coherent type of distributions which could not be obtained if the time-dependent inflow conditions were not copied from the temporal simulations in both square duct and annular duct.

The distribution of streamwise vorticity on the wall-bisector plane is presented in Fig. 28. As seen in the Figure, the instantaneous streamwise vorticity are characterized by the streaky structures both in the regions of near-wall and near-shear layer. These streaky structures have a tendency to tilt with the leading front in the higher velocity inner flow and trailing front in the lower velocity flow that emanates from the annulus or from the wall.

The decay of the mean streamwise velocity along the centerline of the jet is presented in Fig. 29. Here, $U_{\max}$ is the maximum value of the mean streamwise velocity on the jet centerline at the inlet. Two linear decay regions can be found, namely linear decay region I in $0.0 < x/D_e < 3.6$ and linearly decay region II in $5.8 < x/D_e$. Here, D_e is the equivalent diameter defined as the diameter of a round slot with the same exit area as the square slot. By linear regression, see, for example, [25,26], the two linear decay regions can be expressed as $U_{\max}/U_{\mathrm{cl}} = K_u(x/D_e + C_u)$, where U_{cl} is the mean streamwise velocity along the jet centerline, K_u is the mean streamwise velocity decay rate on the jet centerline and C_u is the kinematic virtual origin of the jet. Linear regression gives $K_u = 0.008$, $C_u = 125$ for linear decay region I and $K_u = 0.045$, $C_u = 18.55$ for linear decay region II. The decay rate in region I is much lower than the one in region II, indicating that the major turbulent mixing takes place in the region $5.8 < x/D_e$. Fig. 29 clearly indicates that confining the jets causes the mean centreline velocity to decay much slower than a free jet. The current prediction gives a centerline velocity decay rate of 0.045 for the confined square jet, which is in good agreement with the experimental decay rate at 0.046 from [26] for a confined plane jet. The parallel shift of the two curves, (the difference in the regression constants), is quite probably attributable to the different flow configurations used in the computation and the experiment. That is, Chua and Lua [26] used a 6:1 aspect ratio jet issuing into a 20:1 aspect ratio receiving chamber, whereas here we consider a square configuration.

In summary, the FCAC-MG solver is applied to the spatial simulation of turbulent flow inside a confined square coaxial jet. In the current simulation, the instantaneous temporal simulation results on one of the cross-sections of the square and annular ducts are imposed onto the inlet plane of the confined square coaxial jet. Such prescription of inflow conditions faithfully represents the turbulent inlet conditions and makes it possible to realistically investigate two types of turbulent mixing mechanisms that originate from the streamwise shear and turbulence-driven secondary shear. The flow analyses, the details of which can be found in [12], indicate that the FCAC-MG solver is capable of capturing the turbulent flow physics and accurately predict the turbulent flow phenomena in the confined square coaxial jet.

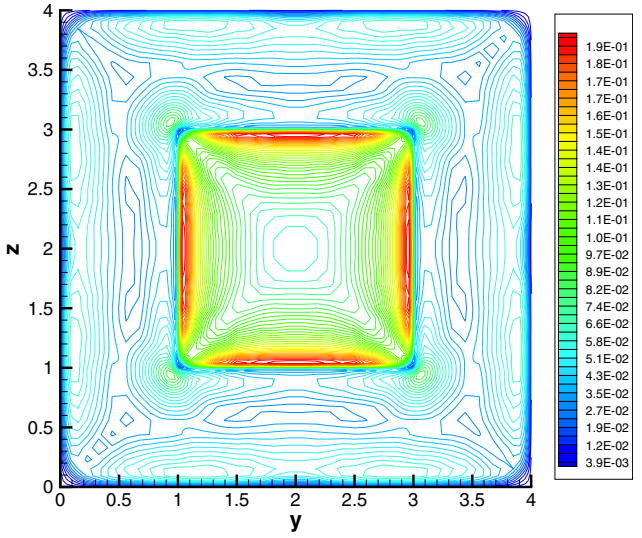


Fig. 27. Streamwise turbulence intensity $\sqrt{\overline{u'^2}}/U_{\mathrm{cl}}$ at inlet of confined square coaxial jet.

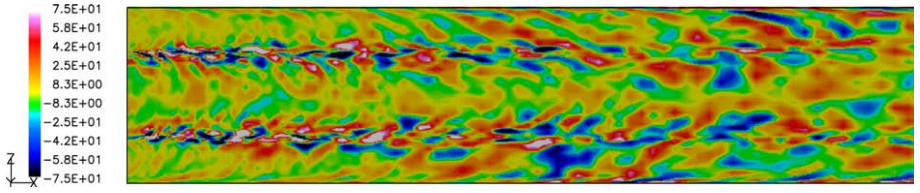


Fig. 28. Instantaneous distribution of streamwise vorticity on wall-bisector plane of confined square coaxial jet.

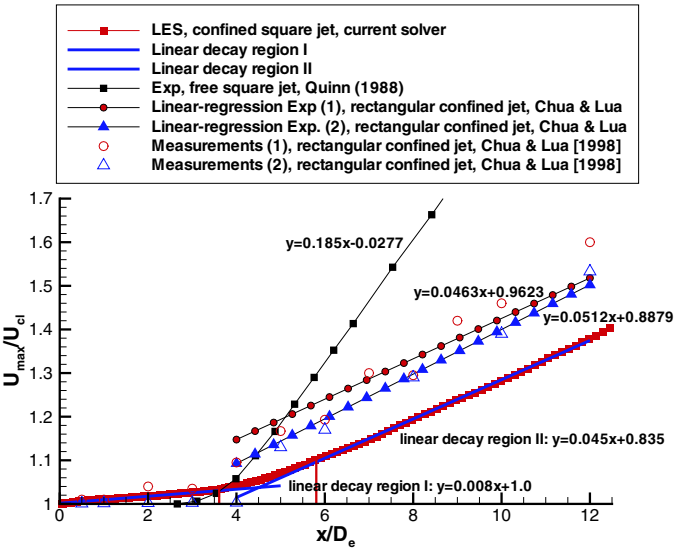


Fig. 29. Mean streamwise velocity decay along the centerline of the confined coaxial jet.

6. Conclusions

Modified TDMA solvers, for both the momentum equations and pressure Poisson equations, were developed to perform temporal turbulence simulations of the flows in both a square duct and a square annular duct. The solutions for the momentum equations are relatively easy to converge and both the conventional and modified TDMA solvers are sufficient to drive the residual down to the level of the computer machine round-off error within a few iterations.

When solving the incompressible unsteady N–S equations, it is well understood that the majority of the CPU time is spent solving the pressure Poisson equation. Towards this end, a robust convergence acceleration technique, such as the FCAC-MG method used here, is required to drive the residual down to a satisfactory level.

A flexible-cycle additive-correction multigrid (FCAC-MG) scheme was utilized to design a high-performance solver for the pressure Poisson equations. The FCAC-MG technique was applied to the three-benchmark cases using a flow-physics oriented solving strategy, which in the present study involved the application of the FCAC-MG technique to the two cross-streamwise directions and the use of a direct solver in the streamwise direction. By making use of this solution strategy, the maximum residual convergence rate could be achieved and the residual of the pressure Poisson equations were guaranteed to be driven down to the level of the computer machine round-off errors.

The multigrid performance analysis indicated that the flexible-cycle scheme could provide more efficient corrections between the fine and coarse grids. This scheme enabled the solver to adjust itself to provide more relaxation on coarser grids as the residuals were driven down and more low-frequency components of the residuals were smoothed out.

The analyses of the simulation results for the three flow configurations, namely, square duct, square annular duct and confined square coaxial jet, indicate that the flow solver based on the FCAC-MG method is capable of capturing the major characteristics of turbulence physics and correctly predicting the relevant turbulent flow phenomena.

Acknowledgment

The authors wish to thank Dr. McIlwain for a number of constructive suggestions and helpful discussions in the writing of the current paper.

Appendix I. Solution procedure of modified tri-diagonal equations (Eq. (7)) from momentum equation

For the forward elimination process, the following operations must be performed so that the element entries become zero in the lower triangle of the matrix and the diagonal entries become unity. For $[row \quad i = 1 \ldots n-2]$, the following three-step operation is required:

Step I: $\frac{1}{a_{ci}} \times [row \quad i]$,
Step II: $[row \quad i+1] - a_{l(i+1)} \times [row \quad i]$,
Step III: $[row \quad n] - a_{rn} \times [row \quad i]$,

which transforms Eq. (7) in the following form:

$$\begin{Bmatrix} 1 & a_{r2}^{n-2} & 0 & 0 & 0 & 0 & 0 & a_{l2}^{n-2} \\ 0 & 1 & a_{r3}^{n-2} & 0 & 0 & 0 & 0 & a_{l3}^{n-2} \\ 0 & 0 & 1 & a_{r4}^{n-2} & 0 & 0 & 0 & a_{l4}^{n-2} \\ 0 & 0 & 0 & 1 & a_{r5}^{n-2} & 0 & 0 & a_{l5}^{n-2} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n-2} & 0 & a_{l6}^{n-2} \\ 0 & 0 & 0 & 0 & 0 & 1 & a_{r7}^{n-2} & a_{l7}^{n-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{c8}^{n-2} & a_{r8}^{n-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{l9}^{n-2} & a_{c9}^{n-2} \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n-2} \\ b_3^{n-2} \\ b_4^{n-2} \\ b_5^{n-2} \\ b_6^{n-2} \\ b_7^{n-2} \\ b_8^{n-2} \\ b_9^{n-2} \end{Bmatrix}, \tag{A1-1}$$

where the superscript $n-2$ denotes the operation number to transform the equations.

Then, for $[row \quad i = n-1]$, the following two-step operation is required:

Step I: $\frac{1}{a_{ci}} \times [row \quad i]$,
Step II: $[row \quad i+1] - a_{l(i+1)} \times [row \quad i]$,

which transforms Eq. (A1-1) in the following form:

$$\begin{Bmatrix} 1 & a_{r2}^{n-1} & 0 & 0 & 0 & 0 & 0 & a_{l2}^{n-1} \\ 0 & 1 & a_{r3}^{n-1} & 0 & 0 & 0 & 0 & a_{l3}^{n-1} \\ 0 & 0 & 1 & a_{r4}^{n-1} & 0 & 0 & 0 & a_{l4}^{n-1} \\ 0 & 0 & 0 & 1 & a_{r5}^{n-1} & 0 & 0 & a_{l5}^{n-1} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n-1} & 0 & a_{l6}^{n-1} \\ 0 & 0 & 0 & 0 & 0 & 1 & a_{r7}^{n-1} & a_{l7}^{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & a_{r8}^{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{c9}^{n-1} \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n-1} \\ b_3^{n-1} \\ b_4^{n-1} \\ b_5^{n-1} \\ b_6^{n-1} \\ b_7^{n-1} \\ b_8^{n-1} \\ b_9^{n-1} \end{Bmatrix}. \tag{A1-2}$$

Finally, for $[row \quad i = n]$, the following one-step operation is required:

Step I: $\frac{1}{a_{ci}} \times [row \quad i]$,

which transforms Eq. (A1-2) into Eq. (A1-3):

$$\begin{Bmatrix} 1 & a_{r2}^{n} & 0 & 0 & 0 & 0 & 0 & a_{l2}^{n} \\ 0 & 1 & a_{r3}^{n} & 0 & 0 & 0 & 0 & a_{l3}^{n} \\ 0 & 0 & 1 & a_{r4}^{n} & 0 & 0 & 0 & a_{l4}^{n} \\ 0 & 0 & 0 & 1 & a_{r5}^{n} & 0 & 0 & a_{l5}^{n} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n} & 0 & a_{l6}^{n} \\ 0 & 0 & 0 & 0 & 0 & 1 & a_{r7}^{n} & a_{l7}^{n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & a_{r8}^{n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n} \\ b_3^{n} \\ b_4^{n} \\ b_5^{n} \\ b_6^{n} \\ b_7^{n} \\ b_8^{n} \\ b_9^{n} \end{Bmatrix}. \tag{A1-3}$$

For the backward substitution process, the following three-step operations must be performed so that a solution to the modified tri-diagonal equations can be derived:

Step I: for $[row \quad i = n]$: $\phi_i = b_i^n$;
Step II: for $[row \quad i = n-1]$: $\phi_i = b_i^n - a_{ri}^n \phi_n$;
Step III: for $[row \quad i = n-2 \ldots 1]$: $\phi_i = b_i^n - a_{li}^n \phi_n - a_{ri}^n \phi_{i+1}$.

The above procedure clearly shows that a solution to the modified tri-diagonal equations can be obtained at a cost of o(N) number of operations. Compared with the solution procedure for the conventional tri-diagonal system, an extra column of storage ($a_{li}^n \quad i = 1 \ldots n-1$) and an extra set of operations are required to solve the modified tri-diagonal system.

Appendix II. Solution procedure of modified tri-diagonal equations (Eq. (17)) from pressure Poisson equation

For the forward elimination process, the following operations must be performed so that the element entries in the lower triangle of the matrix become zero and the diagonal entries become unity. For $[row \quad i = 1 \ldots n-2]$, the following three-step operation is required:

Step I: $\frac{1}{a_{ci}} \times [row \quad i]$,
Step II: $[row \quad i+1] - a_{l(i+1)} \times [row \quad i]$,
Step III: $[row \quad n] - a_{rn} \times [row \quad i]$,

which transforms Eq. (17) as shown below.

For the operations using $[row \quad i = 1]$:

$$\begin{Bmatrix} 1 & a_{r2}^1 & 0 & 0 & 0 & 0 & a_{w22}^1 & a_{w12}^1 \\ 0 & a_{c3}^1 & a_{r3}^1 & 0 & 0 & 0 & a_{w23}^1 & a_{w13}^1 \\ 0 & a_{l4}^1 & a_{c4}^1 & a_{r4}^1 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{l5}^1 & a_{c5}^1 & a_{r5}^1 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{l6}^1 & a_{c6}^1 & a_{r6}^1 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{l7}^1 & a_{c7}^1 & a_{r7}^1 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{l8}^1 & a_{c8}^1 & a_{r8}^1 \\ 0 & a_{r9}^1 & 0 & 0 & 0 & 0 & a_{l9}^1 & a_{c9}^1 \end{Bmatrix}_{n \times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^1 \\ b_3^1 \\ b_4^1 \\ b_5^1 \\ b_6^1 \\ b_7^1 \\ b_8^1 \\ b_9^1 \end{Bmatrix} \tag{A2-1}$$

for the operations using $[row \quad i = 2 \ldots n-3]$:

$$\begin{Bmatrix} 1 & a_{r2}^{n-3} & 0 & 0 & 0 & 0 & a_{w22}^{n-3} & a_{w12}^{n-3} \\ 0 & 1 & a_{r3}^{n-3} & 0 & 0 & 0 & a_{w23}^{n-3} & a_{w13}^{n-3} \\ 0 & 0 & 1 & a_{r4}^{n-3} & 0 & 0 & a_{w24}^{n-3} & a_{w14}^{n-3} \\ 0 & 0 & 0 & 1 & a_{r5}^{n-3} & 0 & a_{w25}^{n-3} & a_{w15}^{n-3} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n-3} & a_{w26}^{n-3} & a_{w16}^{n-3} \\ 0 & 0 & 0 & 0 & 0 & a_{c7}^{n-3} & a_{r7}^{n-3} & a_{w17}^{n-3} \\ 0 & 0 & 0 & 0 & 0 & a_{l8}^{n-3} & a_{c8}^{n-3} & a_{r8}^{n-3} \\ 0 & 0 & 0 & 0 & 0 & a_{r9}^{n-3} & a_{l9}^{n-3} & a_{c9}^{n-3} \end{Bmatrix}_{n \times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n-3} \\ b_3^{n-3} \\ b_4^{n-3} \\ b_5^{n-3} \\ b_6^{n-3} \\ b_7^{n-3} \\ b_8^{n-3} \\ b_9^{n-3} \end{Bmatrix} \tag{A2-2}$$

for the operations using $[row \quad i = n-2]$:

$$\begin{Bmatrix} 1 & a_{r2}^{n-2} & 0 & 0 & 0 & 0 & a_{w22}^{n-2} & a_{w12}^{n-2} \\ 0 & 1 & a_{r3}^{n-2} & 0 & 0 & 0 & a_{w23}^{n-2} & a_{w13}^{n-2} \\ 0 & 0 & 1 & a_{r4}^{n-2} & 0 & 0 & a_{w24}^{n-2} & a_{w14}^{n-2} \\ 0 & 0 & 0 & 1 & a_{r5}^{n-2} & 0 & a_{w25}^{n-2} & a_{w15}^{n-2} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n-2} & a_{w26}^{n-2} & a_{w16}^{n-2} \\ 0 & 0 & 0 & 0 & 0 & 1 & a_{r7}^{n-2} & a_{w17}^{n-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{c8}^{n-2} & a_{r8}^{n-2} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{l9}^{n-2} & a_{c9}^{n-2} \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n-2} \\ b_3^{n-2} \\ b_4^{n-2} \\ b_5^{n-2} \\ b_6^{n-2} \\ b_7^{n-2} \\ b_8^{n-2} \\ b_9^{n-2} \end{Bmatrix}. \tag{A2-3}$$

For $[row \quad i = n-1]$, the following two-step operation is required:

Step I: $\frac{1}{a_{ci}} \times [row \quad i]$,
Step II: $[row \quad i+1] - a_{l(i+1)} \times [row \quad i]$,

which transforms Eq. (A2-3) into the following form:

$$\begin{Bmatrix} 1 & a_{r2}^{n-1} & 0 & 0 & 0 & 0 & a_{w22}^{n-1} & a_{w12}^{n-1} \\ 0 & 1 & a_{r3}^{n-1} & 0 & 0 & 0 & a_{w23}^{n-1} & a_{w13}^{n-1} \\ 0 & 0 & 1 & a_{r4}^{n-1} & 0 & 0 & a_{w24}^{n-1} & a_{w14}^{n-1} \\ 0 & 0 & 0 & 1 & a_{r5}^{n-1} & 0 & a_{w25}^{n-1} & a_{w15}^{n-1} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n-1} & a_{w26}^{n-1} & a_{w16}^{n-1} \\ 0 & 0 & 0 & 0 & 0 & 1 & a_{r7}^{n-1} & a_{w17}^{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & a_{r8}^{n-1} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_{c9}^{n-1} \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n-1} \\ b_3^{n-1} \\ b_4^{n-1} \\ b_5^{n-1} \\ b_6^{n-1} \\ b_7^{n-1} \\ b_8^{n-1} \\ b_9^{n-1} \end{Bmatrix}. \tag{A2-4}$$

For $[row \quad i = n]$, a one-step operation is required:

Step I: $\frac{1}{a_{ci}} \times [row \quad i]$,

which results in

$$\begin{Bmatrix} 1 & a_{r2}^{n} & 0 & 0 & 0 & 0 & a_{w22}^{n} & a_{w12}^{n} \\ 0 & 1 & a_{r3}^{n} & 0 & 0 & 0 & a_{w23}^{n} & a_{w13}^{n} \\ 0 & 0 & 1 & a_{r4}^{n} & 0 & 0 & a_{w24}^{n} & a_{w14}^{n} \\ 0 & 0 & 0 & 1 & a_{r5}^{n} & 0 & a_{w25}^{n} & a_{w15}^{n} \\ 0 & 0 & 0 & 0 & 1 & a_{r6}^{n} & a_{w26}^{n} & a_{w16}^{n} \\ 0 & 0 & 0 & 0 & 0 & 1 & a_{r7}^{n} & a_{w17}^{n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & a_{r8}^{n} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{Bmatrix}_{n\times n} \cdot \begin{Bmatrix} \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \\ \phi_9 \end{Bmatrix} = \begin{Bmatrix} b_2^{n} \\ b_3^{n} \\ b_4^{n} \\ b_5^{n} \\ b_6^{n} \\ b_7^{n} \\ b_8^{n} \\ b_9^{n} \end{Bmatrix}. \tag{A2-5}$$

For the backward substitution process, the following four-step operations are required to obtain a solution to the modified tri-diagonal equations:

Step I: for $[row \quad i = n]$: $\phi_i = b_i^n$;
Step II: for $[row \quad i = n-1]$: $\phi_i = b_i^n - a_{ri}^n \phi_n$;

Step III: for $[row \quad i = n-2]$: $\phi_i = b_i^n - a_{ri}^n \phi_{i+1} - a_{w1i}^n \phi_n$;
Step IV: for $[row \quad i = n-3 \ldots 1]$: $\phi_i = b_i^n - a_{ri}^n \phi_{i+1} - a_{w1i}^n \phi_n - a_{w2i}^n \phi_{n-1}$.

By adopting the above solving procedure, the solution to the modified tri-diagonal equations can be obtained at a cost of o(N) number of operations. When compared against the solution procedure for the conventional tri-diagonal system, two extra columns of storage ($a_{w1i}^n \quad i = 1 \ldots n-2$), ($a_{w2i}^n \quad i = 1 \ldots n-3$) and two extra sets of operations are required to solve the modified tri-diagonal system.

References

[1] A.J. Chorin, The numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.
[2] R. Temam, Sur l'approximation de la solution des equations de Navier–Stokes par la methods des pas fractionnaires (I), Arch. Rat. Mech. Anal. 32 (1969) 135–153.
[3] J. Kim, P. Moin, Application of a fractional-step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308–323.
[4] R.K. Madabhushi, S.P. Vanka, Large eddy simulation of turbulence-driven secondary flow in a square duct, Phys. Fluids A 3 (11) (1991) 2734–2745.
[5] H. Xu, Large eddy simulation of turbulent flows in square and annular ducts and confined square coaxial jet, Ph.D. Thesis, Department of Mechanical Engineering, Queen's University, 1997.
[6] A. Brandt, Multi-level adaptive solution to boundary-value problems, Math. Comput. 31 (1977) 333–390.
[7] D. Drikakis, O.P. Iliev, D.P. Vassileva, A nonlinear multigrid method for three-dimensional incompressible Navier–Stokes equations, J. Comput. Phys. 146 (1998) 301–321.
[8] D. Drikakis, O.P. Iliev, D.P. Vassileva, Acceleration of multigrid flow computations through dynamic adaptation of the smoothing procedure, J. Comput. Phys. 165 (2000) 566–591.
[9] H.C. Elman, V.E. Howle, J.N. Shadid, R.S. Tuminaro, A parallel block multi-level preconditioner for the 3D incompressible Navier–Stokes equations, J. Comput. Phys. 187 (2003) 504–523.
[10] B.R. Hutchinson, G.D. Raithby, A Multigrid method based on the additive correction strategy, Numer. Heat Transfer 9 (1986) 511–537.
[11] H. Xu, A. Pollard, Large eddy simulation of turbulent flow in a square annular duct, Phys. Fluids 13 (11) (2001) 3321–3337.
[12] H. Xu, M. Khalid, A. Pollard, Large eddy simulation of turbulent flow in a confined square coaxial jet, Int. J. Comput. Fluid Dynamics 17 (5) (2003) 339–356.
[13] S. Gavrilakis, Numerical simulation of low-Reynolds-number turbulent flow through a straight square duct, J. Fluid Mech. 244 (1992) 101–129.
[14] L.H. Thomas, Elliptic problems in linear difference equations over a network, Watson Sci. Comput. Lab Report, Columbia University, New York, 1949.
[15] H.K. Versteeg, W. Malalasekera, Introduction to Computational Fluid Dynamics: The Finite Volume Method, Prentice-Hall, 1995.
[16] A. Huser, S. Biringen, Direct numerical simulation of turbulent flow in a square duct, J. Fluid Mech. 257 (1993) 65–95.
[17] M.A. Niederschulte, Turbulent flow through a rectangular channel, Ph.D thesis, University of Illinois at Urbana-Champaign, 1989.
[18] K. Nishino, N. Kasagi, Turbulence statistics measurement in a two-dimensional channel flow using a three-dimensional particle tracking velocimeter, in: Seventh Symposium on Turbulent Shear Flows, Stanford University, August 21–23, 1989.
[19] J. Nikuradse, Turbulente stromung in nicht-kreisformigen Rohren, Ing. Arch. 1 (1930) 306–332.
[20] P. Bradshaw, Turbulent secondary flows, Ann. Rev. Fluid Mech. 19 (1987) 53–74.
[21] R.S. Rogallo, P. Moin, Numerical simulation of turbulent flows, Ann. Rev. Fluid Mech. 16 (1984) 99–137.
[22] M. Ciofalo, Large eddy simulation: a critical survey of models and applications, Adv. Heat Transfer 25 (1994) 321–419.
[23] K. Akselvoll, P. Moin, Large eddy simulation of turbulent confined co-annular jets and turbulent flow over a backward facing step, Report TF-63, Thermo-Sciences Division, Department of Mechanical Engineering, Stanford University, 1995.
[24] M. Breuer, W. Rodi, Large eddy simulation of turbulent flow through straight and curved ducts, ERCOFTAC Bull. 22 (1994) 26–29.
[25] W.R. Quinn, Experimental and numerical study of a turbulent free square jet, Phys. Fluids 31 (5) (1988).
[26] L.P. Chua, A.C. Lua, Measurements of a confined jet, Phys. Fluids 10 (12) (1998) 3137–3144.