INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **38**:789–810 (DOI: 10.1002/fld.253)

Numerical solution of incompressible flows by discrete singular convolution

D. C. Wan, Y. C. Zhou and G. W. Wei*,†

Department of Computational Science, National University of Singapore, Singapore 117543.

SUMMARY

A discrete singular convolution (DSC) solver is developed for treating incompressible flows. Three different two-dimensional benchmark problems, the Taylor problem, the driven cavity flow, and a periodic shear layer flow, are utilized to test the accuracy, to explore the reliability and to demonstrate the efficiency of the present approach. Solution of extremely high accuracy is attained in the analytically solvable Taylor problem. The results of treating the other problems are in excellent agreement with those in the literature. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: discrete singular convolution; Taylor problem; driven cavity flow; periodic shear layer flow

1. INTRODUCTION

The prediction of complex unsteady fluid flow requires numerical methods which provide high accuracy at an affordable cost. Two major aspects are involved in the construction of an accurate, efficient and robust flow solver. One of the aspects is iterative schemes or, more precisely, algorithms for the time discretization and treatment of the pressure field in an incompressible flow. In general, an iterative algorithm must be stable, convergent and consistent with the boundary conditions and the spatial discretization. The other aspect concerns spatial discretization, in which most effort has been focused on developing either global methods [1–4] or local methods [5–8]. Global methods, such as Chebyshev spectral approximation, pseudospectral methods and fast Fourier transform, are highly localized in their spectral representations, but unlocalized in their coordinate space. By contrast, local methods, such as finite difference, finite volume, finite strip and finite element methods, have high spatial localization, but are poorly localized in their spectral representations. As a consequence, global methods are usually much more accurate than local methods, while the major advantage of local methods is their flexibility in handling complex geometries and boundary conditions,

*Correspondence to: G. W. Wei, Department of Computational Science, National University of Singapore, Faculty of Science, Block S17, Level 7, 10 Kent Ridge Crescent, Singapore 117543.
†E-mail: cscweigw@nus.edu.sg

Received 3 November 1999
Revised 18 June 2001

which are crucial in many practical applications. Both global and local methods have achieved great success in solving various fluid dynamical problems. However, for complex flow phenomena, both global and local methods encounter enormous difficulties in maintaining a high numerical accuracy. Global methods become cumbersome to implement over complex geometries. Spectral element methods can be used to overcome the geometric difficulty but they lose accuracy at element boundaries. Standard local methods are known to converge very slowly when the grid is refined. In many situations, local methods have difficulties to capture small-scale structures occurring in a complex flow field, such as those in turbulence, due to their limited numerical resolution. Although some special schemes, such as quadratic upwind interpolation for convective kinematics (QUICK), and high-order-accurate compact difference schemes, are very useful for treating complex flows, they are often tedious to implement. Moreover, their associated numerical dissipation and non-physical oscillations may hinder their applications in dealing with mixed fluid flow phenomena, such as turbulent flow, multiphase flow and combustion in an irregular domain. Therefore, it is desirable to have methods that combine global methods' accuracy with local methods' flexibility. Wavelets and related multiresolution techniques are expected to fill this gap. With features of time–frequency (or position–momentum) localization and multiresolution analysis, wavelet theory and related techniques have had great impact on signal processing, image compression and telecommunication. Thus, enormous effort has been made recently for developing wavelet methods to solve ordinary differential equations (ODEs) and partial differential equations (PDEs) [9], including the most recently reported adaptive wavelet algorithm [10]. However, this effort is hindered, to a certain degree, by the difficulty of implementing boundary conditions in the multiresolution analysis and by the lack of sufficiently accurate and efficient wavelet algorithms.

In order to combine the merits of both global methods and local methods, a discrete singular convolution (DSC) algorithm [11–13] was developed. The DSC algorithm provides a robust and highly accurate approach for computer realization of singular convolutions, which involve singular kernels of either delta type, Hilbert type, or Abel type. It has the wavelet feature of time–frequency localization, but avoids the difficulty and redundancy of multiresolution analysis. In the DSC algorithm, numerical approximations of a function and its derivatives can be treated as convolutions with kernels. When the delta distribution is appropriately approximated by a sequence, it can be discretized on a grid for interpolation. Extremely high computational accuracy can be achieved if the DSC is properly implemented in a numerical algorithm. Historically, the delta distribution was informally used by physicists and engineers, and was later formalized in rigorous mathematical framework by Schwarz [14] and others. One of approximations to the delta distribution, which is often used in the DSC algorithm, is Shannon's delta kernel. The most important property of Shannon's delta kernel is that, it actually forms an orthogonal sampling basis for the Paley–Wiener reproducing kernel Hilbert space. Therefore, a bandlimited L^2 function on the real line can be exactly reconstructed from a set of discrete function values. In order to improve the local and asymptotic behaviour of Shannon's delta kernel, a regularization procedure proposed in an earlier work [15] was used to increase the regularity of the kernel [11]. As a result, discrete approximations of tempered distributions can be constructed by using the regularized Shannon's delta kernel in the framework of the DSC algorithm [11]. It turns out that, while treating the delta distribution, the DSC approach provides a unified description [12] to a number of commonly used computational methods. The DSC algorithm has been successfully applied to the treatment of the (non-linear) sine-Gordon homoclinic orbit singularity [16], eigenvalue problems of the

Fokker–Planck equation [11] and the Schrödinger equation [17], structural analysis [18, 19], the formation of nanoscale morphological patterns [20] and the solution of the Navier–Stokes equations with periodic boundary conditions [21, 22]. It was shown that the DSC algorithm provides an excellent scheme for image edge detection [23].

The objective of the present work is to explore the utility of the DSC algorithm in the framework of the successive over-relaxation (SOR) for the solution of the Navier–Stokes equation and the Euler equation for incompressible flows. A previous computation demonstrated that the DSC approach is capable of providing extremely high accuracy for treating the Taylor problem and the periodic shear layer flow with relatively fewer grid points in the computational domain [21]. However, the linear system of coupled algebraic equations was solved by using the standard lower–upper (LU) triangular decomposition scheme. As such, it requires large computer memory and is very difficult to handle large-scale computations of practical origins. The present development makes it possible for the highly accurate DSC algorithm to be used in fluid flow simulations with less computer memory requirement. Moreover, it is relatively easy to adopt the present method for flows involving complex boundaries.

This paper is organized as follows. A brief review of the DSC algorithm is given in Section 2 to improve readability of the present work. Section 3 is devoted to a detailed description of spatial–temporal discretizations of the governing Navier–Stokes equations. A fractional time step and potential function method is employed to overcome the difficulty occurring in treating the pressure field in incompressible flows. The standard SOR iteration technique is utilized for solving the resulting Neumann–Poisson equation. In Section 4, three numerical problems are chosen to test the present DSC–SOR approach. The numerical accuracy is confirmed by using an analytically solvable Taylor problem—two-dimensional Navier–Stokes equations with periodic boundary conditions. The utility of the present approach for incompressible flows is demonstrated further by solving the driven cavity problem and the Euler equation for periodic shear layers. This paper ends with a conclusion.

2. DISCRETE SINGULAR CONVOLUTION

In this section, the discrete singular convolution (DSC) algorithm is briefly reviewed both for the integrity of the presentation and for the convenience of the reader. More details of the algorithm can be found in Reference [11].

In the context of distribution theory, a singular convolution can be defined by

$$F(t) = (T * \eta)(t) = \int_{-\infty}^{\infty} T(t-x)\eta(x)\,\mathrm{d}x \tag{1}$$

where T is a singular kernel and $\eta(t)$ is an element of the space of test functions. A singular kernel can be regarded as an element of the dual space of test functions, and thus is a distribution. Interesting examples include singular kernels of Hilbert type and of delta type. The former plays an important role in the theory of analytical functions, processing of analytical signals, theory of linear responses and Radon transform, whereas the latter is a key element in the theory of approximations and the numerical solution of differential equations. Since the present work deals exclusively with the numerical solution of partial differential equations,

our attention is focused on the singular kernels of delta type

$$T(x) = \delta^{(q)}(x), \quad (q = 0, 1, 2, \ldots) \tag{2}$$

where superscript (q) denotes the qth-order 'derivative' of the delta distribution with respect to x, which should be understood as generalized derivatives of distributions. When $q = 0$, the kernel, $T(x) = \delta(x)$, is important for the interpolation of surfaces and curves, including applications to the design of engineering structures. The kernels of other q values can be used for solving differential equations. It is easy to see that the singular kernels of delta type are singular—they cannot be directly digitized in a computer. Therefore, the singular convolution of Equation (1) cannot be directly used for numerical computations. Moreover, the restriction to the test function is too strict for most practical applications. To avoid the difficulty of using singular expressions directly in a computer, one can construct a sequence of approximations, δ_α, to approximately represent the delta distribution, $\delta(x)$.

$$\lim_{\alpha \to \infty} \delta_\alpha(x) \to \delta(x) \tag{3}$$

A simple example is Shannon's delta sequence

$$\delta_\alpha(x) = \frac{\sin(\alpha x)}{\pi x} \tag{4}$$

By definition, it recovers the delta distribution at the limit,

$$\lim_{\alpha \to \infty} \int \frac{\sin \alpha(x - y)}{\pi(x - y)} \eta(y)\, \mathrm{d}y = \eta(x) \tag{5}$$

In fact, Equation (5) is valid in many other conditions where the function of interest is no longer restricted to test functions. For example, for a given $\alpha > 0$, Shannon's delta kernel provides a basis for the Paley–Wiener reproducing kernel Hilbert space, B_α^2, which is a subspace of the Hilbert space $L^2(R)$. As such, an L^2 function bandlimited to α (i.e. those functions that have compact support in the interval $[-\alpha, \alpha]$ in their Fourier representation) can be exactly reproduced by a convolution.

For numerical computations, what is important is *discrete singular convolution* which provides appropriate approximations to the original convolution of distributions

$$f(x) \approx \sum_x \delta_\Delta(x - x_k) f(x_k) \tag{6}$$

where

$$\delta_\Delta(x - x_k) = \Delta \delta_\alpha(x - x_k) \tag{7}$$

is designed for being used in discrete summations. Here $\{x_k\}$ is an appropriate set of discrete points centred around the point x and Δ is the grid spacing. Depending on the mathematical properties of the kernel, δ_α, the restriction on the $f(x_k)$ can be relaxed to include many common-occurring functions. The discrete convolution can be realized by a sampling series. In the case of Shannon's delta kernel, it is most efficient to sample at the Nyquist frequency $\alpha = \pi/\Delta$. Because the Nyquist frequency provides highest computational efficiency both on

and off a grid. In fact, it provides exact results when the sampling points are extended to an infinite series

$$f(x)=\sum_{k=-\infty}^{\infty} f(x_k)\,\frac{\sin((\pi/\Delta)(x-x_k))}{(\pi/\Delta)(x-x_k)} \quad \forall f\in B^2_{\pi/\Delta} \tag{8}$$

This is recognized as Shannon's sampling theorem which reconstructs $B^2_{\pi/\Delta}$ function from a set of discrete values. Shannon's delta kernel forms a *sampling basis* for the Paley–Wiener reproducing kernel Hilbert space. Obviously, the requirement of infinitely many sampling points is impractical for computer realization. In signal processing language, Shannon's sampling kernel is an ideal low-pass filter which is necessarily an infinite impulse response (IIR) function. A truncated Shannon's delta kernel is practical to be used but the truncation error is substantial. In order to improve the localization and asymptotic behaviour of Shannon's delta kernel, a regularization procedure was employed to efficiently reduce the truncation error [11]

$$\delta_\alpha(x)\to\delta_{\alpha,\sigma}(x)=\delta_\alpha(x)R_\sigma(x) \tag{9}$$

where R_σ is a regularizer which satisfies

$$\lim_{\sigma\to\infty} R_\sigma(x)=1 \tag{10}$$

and

$$\int_{-\infty}^{\infty}\lim_{\alpha\to\infty}\delta_\alpha(x)R_\sigma(x)\,\mathrm{d}x=R_\sigma(0)=1 \tag{11}$$

A commonly used regularizer is the Gaussian

$$R_\sigma(x)=\exp\left(-\frac{x^2}{2\sigma^2}\right),\quad \sigma>0 \tag{12}$$

where σ determines the width of the Gaussian window and is often varied in association with the central frequency $\alpha=\pi/\Delta$ (i.e. $\sigma=r\Delta$, here, r is a parameter chosen in computations). It is easy to see that the Gaussian regularizer in Equation (12) satisfies the requirements of Equations (10) and (11). By combining Equations (7), (9) and (12), the regularized Shannon's delta kernel in its discrete form can be given as

$$\delta_{\Delta,\sigma}(x-x_k)=\frac{\sin[(\pi/\Delta)(x-x_k)]}{[(\pi/\Delta)(x-x_k)]}\,\exp\left(-\frac{(x-x_k)^2}{2\sigma^2}\right) \tag{13}$$

The difference between the present discrete form and the continuous form in Equation (4) is to be noted. Expression (13) can be used to provide discrete approximations to the singular convolution kernels of the delta type

$$f^{(q)}(x)\approx\sum_{k=-W}^{W}\delta^{(q)}_{\Delta,\sigma}(x-x_k)f(x_k) \tag{14}$$

where the superscript (q) denotes the qth-order derivative with respect to x, and $2W+1$ is the computational bandwidth which is centred around x and is usually smaller than the whole computational domain. Therefore, the resulting approximation matrix has a banded structure,

which is particularly important to large-scale computations. The nature of approximation in Equation (14) is due to the truncation and the regularization.

Recently, a mathematical estimation for the choice of W, σ, Δ has been given [24]. For example, if the L^2 error for approximating an L^2 function f is set to $10^{-\eta}$, the following relations are to be satisfied:

$$r(\pi - B\Delta) > \sqrt{4.16\eta} \quad \text{and} \quad \frac{W}{r} > \sqrt{4.61\eta} \tag{15}$$

where $r = \sigma/\Delta$ and B is the frequency bounded for the function of interest, f. The first inequality states that for a given grid size Δ, a large r is required for approximating high-frequency component of an L^2 function. The second inequality indicates that if one chooses the ratio $r = 3$, then the half bandwidth $W \sim 30$ can be used to ensure the highest accuracy in a double precision computation ($\eta = 15$).

When the regularized Shannon's delta kernel is used, the detailed expressions for $\delta_{\Delta,\sigma}(x)$, $\delta^{(1)}_{\Delta,\sigma}(x)$ and $\delta^{(2)}_{\Delta,\sigma}(x)$ can be easily given as:

$$\delta_{\Delta,\sigma}(x) = \begin{cases} \dfrac{\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\pi x/\Delta} & (x \neq 0) \\ 1 & (x = 0) \end{cases} \tag{16}$$

$$\delta^{(1)}_{\Delta,\sigma}(x) = \begin{cases} \dfrac{\cos(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{x} - \dfrac{\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\pi x^2/\Delta} \\ \quad - \dfrac{\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\pi\sigma^2/\Delta} & (x \neq 0) \\ 0 & (x = 0) \end{cases} \tag{17}$$

and

$$\delta^{(2)}_{\Delta,\sigma}(x) = \begin{cases} \dfrac{-(\pi/\Delta)\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{x} - 2\dfrac{\cos(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{x^2} \\ \quad - 2\dfrac{\cos(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\sigma^2} + 2\dfrac{\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\pi x^3/\Delta} \\ \quad + \dfrac{\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\pi\sigma^2 x/\Delta} + \dfrac{x\sin(\pi x/\Delta)\exp(-x^2/2\sigma^2)}{\pi\sigma^4/\Delta} & (x \neq 0) \\ -\dfrac{3 + \pi^2\sigma^2/\Delta^2}{3\sigma^2} & (x = 0) \end{cases} \tag{18}$$

It should be pointed out that although only the regularized Shannon's delta kernel and its derivatives are used in this work, there are many other excellent discrete approximations described elsewhere [11, 18].

It is noted that Equation (14) is very efficient since just one kernel is required for the whole computational domain $[a, b]$ for given Δ and σ. We refer this kernel as translationally invariant. However, there is a technical aspect which concerns Equation (14) at a computational boundary. It is obvious that functions, $f(x_k)$, need to be located outside the computational

domain $[a,b]$, where their values are usually undefined. Therefore, it is necessary to make assumptions about those function values that are beyond the computational domain. In the DSC algorithm, such functions of $f(x_k)$ are obtained according to their corresponding boundary conditions. For example, in Dirichlet boundary conditions, such $f(x_k)$ can be taken to be $f(a)$ or $f(b)$; in periodic boundary conditions, such $f(x_k)$ can be obtained by periodic extensions from their corresponding values inside the computational domain $[a,b]$; in Neumann boundary conditions, such $f(x_k)$ can be determined by $f'(a)$ (or $f'(b)$).

3. METHOD OF SOLUTION

In this section, solution algorithm for incompressible flows is described in detail. The governing equations for the problem are reviewed before an iterative scheme for treating the pressure field is presented. Finally, the successive over-relaxation approach is adopted in association with the DSC algorithm to elucidate a new DSC-SOR solver.

3.1. Governing equations and spatial discretization

The two-dimensional Navier–Stokes equations in the Cartesian coordinates for an incompressible viscous flows can be written in the dimensionless form as

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) \tag{19}$$

$$\frac{\partial v}{\partial t} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) \tag{20}$$

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{21}$$

where u and v are the velocity components in the x- and y-directions, respectively, p is the pressure field, t is time and Re is the Reynolds number. For the convenience of discussion, the Navier–Stokes equations are written in the form of

$$D(U) = 0 \tag{22}$$

$$\frac{\partial U}{\partial t} = F(U) - \nabla p \tag{23}$$

where $U = (u,v)^{\mathrm{T}}$, D denotes the divergence operator and $F(U) = [G(U), H(U)]^{\mathrm{T}}$ is the shorthand notation for the combination of viscous term and the non-linear term

$$G(U) = \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) - \left(u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}\right) \tag{24}$$

$$H(U) = \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) - \left(u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}\right) \tag{25}$$

representing its components in x- and y-directions.

In this work, a uniform grid in both the x- and y-directions is employed (i.e. $\Delta x = \Delta y = \Delta$). As such, the resulting DSC kernel is translationally invariant and only a single kernel is required for all computations. We denote a grid point (x_i, y_j) by $x_i = i\Delta x$ and $y_j = j\Delta y$ and the differences of two grid points by $x_i - x_{i+k} = -k\Delta x$ and $y_j - y_{j+k} = -k\Delta y$. The values of u, v and p at a grid point (x_i, y_j) are denoted by $u_{i,j}$, $v_{i,j}$ and $p_{i,j}$, respectively. Therefore, the spatial discretizations of $G(U)$, $H(U)$, $D(U)$ and ∇p at point (i, j) are given by

$$D_h(U) = \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta x) u_{i+k,j} + \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta y) v_{i,j+k} \tag{26}$$

$$G_h(U) = \frac{1}{Re}\left[\sum_{k=-W}^{W} \delta^{(2)}_{\Delta,\sigma}(k\Delta x) u_{i+k,j} + \sum_{k=-W}^{W} \delta^{(2)}_{\Delta,\sigma}(k\Delta y) u_{i,j+k}\right] - \left[u_{i,j} \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta x) u_{i+k,j} + v_{i,j} \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta y) u_{i,j+k}\right] \tag{27}$$

$$H_h(U) = \frac{1}{Re}\left[\sum_{k=-W}^{W} \delta^{(2)}_{\Delta,\sigma}(k\Delta x) v_{i+k,j} + \sum_{k=-W}^{W} \delta^{(2)}_{\Delta,\sigma}(k\Delta y) v_{i,j+k}\right] - \left[u_{i,j} \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta x) v_{i,j+k} + v_{i,j} \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta y) v_{i+k,j}\right] \tag{28}$$

$$\nabla_h p = \left[\sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta x)\, p_{i+k,j}, \sum_{k=-W}^{W} \delta^{(1)}_{\Delta,\sigma}(k\Delta y)\, p_{i,j+k}\right]^{\mathrm{T}} \tag{29}$$

These expressions are used in the temporal discretization of the Navier–Stokes equations.

3.2. Temporal discretization and the treatment of pressure

A third-order Runge–Kutta (RK) scheme is used in the present study. In this scheme, Equation (23) can be solved by

$$U^{(1)} = \alpha_1 U^n + \beta_1[\Delta t(F_h(U^n) - \nabla_h p^{(1)})] \tag{30}$$

$$U^{(2)} = \alpha_2 U^n + \beta_2[U^{(1)} + \Delta t(F_h(U^{(1)}) - \nabla_h p^{(2)})] \tag{31}$$

$$U^{(n+1)} = \alpha_3 U^n + \beta_3[U^{(2)} + \Delta t(F_h(U^{(2)}) - \nabla_h p^{n+1})] \tag{32}$$

where $(\alpha_1, \alpha_2, \alpha_3) = (1, 3/4, 1/3)$, $(\beta_1, \beta_2, \beta_3) = (1, 1/4, 2/3)$ and Δt is the time increment. The superscripts $n, n+1$ denote time t_n and t_{n+1}, and (1), (2) represent the first and second step values, respectively.

Considering the first step of the RK scheme, an intermediate velocity field $U^{*(1)}$ can be obtained through

$$U^{*(1)} = \alpha_1 U^n + \beta_1[\Delta t(F_h(U^n) - \nabla_h p^n)] \tag{33}$$

where $U^{*(1)}$ satisfies the same boundary conditions as those for the physical velocity field U^{n+1} at time t^{n+1}. The time increment of this equation is restricted by the CFL condition

$$\max(\Delta t) \leqslant \min\left(\frac{4}{Re(|u|+|v|)^2}, \frac{Re((\Delta x)^2+(\Delta y)^2)}{4}\right) \tag{34}$$

By subtracting Equation (30) from Equation (33), one has

$$U^{(1)} - U^{*(1)} = -\beta_1 \Delta t \nabla_h (p^{(1)} - p^n) \tag{35}$$

Updating the pressure field at each step of the RK scheme requires special care, since there is no obvious way to arrive at a governing equation for the pressure field. A common approach is to 'derive' a pressure equation from the incompressible condition, Equation (21). Many specially designed schemes have been developed for this derivation. For example, there is a family of semi-implicit pressure-correction methods known as semi-implicit method for pressure linked equation (SIMPLE), consistent SIMPLE algorithm (SIMPLEC) and SIMPLER [25]. Artificial compressibility method [26], Marker and Cell (MAC) method [27], fractional-step projection method [28] and its many variations are also commonly used in the field [29]. In this work, we adopt a fractional-time-step and potential-function method (FTSPFM), also known as the pressure correction method [30, 31], which is a variant of the MAC method for solving the governing Equations (19), (20) and (21). In the FTSPFM method, an intermediate velocity field and a potential function are employed to correct the values of the velocity and pressure fields at a new time step.

Since pressure is a scalar quantity, one can define a first-step potential function $\phi^{(1)}$

$$\phi^{(1)} = -\beta_1 \Delta t (p^{(1)} - p^n) \tag{36}$$

Hence, the pressure difference can be rewritten in terms of the first step potential function

$$U^{(1)} - U^{*(1)} = \nabla_h \phi^{(1)} \tag{37}$$

By taking divergence over Equation (37), one attains the following Poisson equation for the first-step potential function

$$\nabla_h^2 \phi^{(1)} = D_h(U^{(1)}) - D_h(U^{*(1)}) \tag{38}$$

Here, $\nabla_h^2 \phi^{(1)}$ is given by

$$\nabla_h^2 \phi_{i,j}^{(1)} = \sum_{k=-W}^{W} \delta_{\Delta,\sigma}^{(2)}(k\Delta x)\phi_{i+k,j}^{(1)} + \sum_{k=-W}^{W} \delta_{\Delta,\sigma}^{(2)}(k\Delta y)\phi_{i,j+k}^{(1)} \tag{39}$$

From Equation (22), the first-step velocity $U^{(1)}$ should satisfy the continuity equation, i.e.

$$D_h(U^{(1)}) = 0 \tag{40}$$

It follows that the Poisson equation for the first-step potential function can be simplified

$$\nabla_h^2 \phi^{(1)} = -D_h(U^{*(1)}) \tag{41}$$

Since the first-step intermediate velocity field $U^{*(1)}$ and first-step velocity $U^{(1)}$ satisfy the same boundary condition as that for the physical velocity U^{n+1} at time t^{n+1}, it follows from Equation (37) that, the first-step potential function $\phi^{(1)}$ satisfies the following Neumann boundary condition

$$\nabla_h \phi^{(1)} = 0 \tag{42}$$

at most computational boundary except for one point. Equation (41), together with Equation (42), constitutes a Neumann–Poisson problem for the first order potential function $\phi^{(1)}$. To attain a unique solution for pressure, the Neumann boundary condition is modified by assigning a fixed constant value for the potential function ϕ (or pressure) at one grid point along the outer physical boundary. The right-hand side of Equation (41) involves only quantities determined from the first-step intermediate velocity field $U^{*(1)}$, which in turn, can be obtained by explicit expressions of Equation (33) from the known velocity U^n and pressure p^n.

In the present DSC algorithm, the most time-consuming part of computations is the solution of the Poisson equation (41). Any attempt to improve the computational efficiency should first focus on the Poisson solver. In an earlier work [22], both the Poisson equation for pressure and the Helmholtz equation for the velocity field were solved by the direct LU decomposition approach. As such, it was very difficult to handle large-scale computations required for complex flow problems. To reduce the memory requirement, iterative approaches are used. To this end, the successive over relaxation (SOR) method is utilized for solving the Poisson equation (41). The SOR expression for Equation (41) is (here we have dropped the superscript (1))

$$\begin{aligned}\phi_{i,j}^{l+1} = \frac{\omega}{2\delta_{\Delta,\sigma}^{(2)}(0)} \Bigg(& -D_h(U^*) - \sum_{k=-W}^{-1} \delta_{\Delta,\sigma}^{(2)}(k\Delta x)\phi_{i+k,j}^{l+1} - \sum_{k=1}^{W} \delta_{\Delta,\sigma}^{(2)}(k\Delta x)\phi_{i+k,j}^{l} \\ & - \sum_{k=-W}^{-1} \delta_{\Delta,\sigma}^{(2)}(k\Delta y)\phi_{i,j+k}^{l+1} - \sum_{k=1}^{W} \delta_{\Delta,\sigma}^{(2)}(k\Delta y)\phi_{i,j+k}^{l} \Bigg) + (1-\omega)\phi_{i,j}^{l}\end{aligned} \tag{43}$$

where the superscript l is the number of iterations, ω is the over relaxation factor, which can be empirically chosen in the range from 1 to 2 in practical computations. By making use of the boundary condition for ϕ in Equation (42), the values of ϕ located outside of computational domain can be obtained by symmetric extensions of ϕ inside computational domain. Therefore, we solve the Poisson equation both at the interior and boundary points.

The iterative convergence criterion for Equation (43) is set to

$$\max |\phi_{i,j}^{l+1} - \phi_{i,j}^{l}| \leqslant \varepsilon \tag{44}$$

where $\varepsilon > 0$ is a given small parameter.

After obtaining the $\phi^{(1)}$, the updated velocity and the pressure at the first step can be evaluated from the following correction equations

$$u_{i,j}^{(1)} = u_{i,j}^{*(1)} + \sum_{k=-W}^{W} \delta_{\Delta,\sigma}^{(1)}(k\Delta x)\phi_{i+k,j}^{(1)} \tag{45}$$

$$v_{i,j}^{(1)} = v_{i,j}^{*(1)} + \sum_{k=-W}^{W} \delta_{\Delta,\sigma}^{(1)}(k\Delta y)\phi_{i,j+k}^{(1)} \tag{46}$$

and

$$p_{i,j}^{(1)} = p_{i,j}^{n} - \frac{\phi_{i,j}^{(1)}}{\beta_1 \Delta t} \tag{47}$$

Similarly, the velocity and the pressure at the second step and at the next time step t_{n+1} are computed. The overall numerical procedure is summarized as follows:

1. From the known velocity and pressure fields at time t_n or from the initial guessed values for the fields, the intermediate velocity fields (u^*, v^*) are computed according to Equation (33).
2. By making use of u^* and v^*, the potential function ϕ is evaluated by using the SOR method according to Equation (43). The convergence of this inner iteration is controlled by the tolerance ε.
3. By using u^*, v^* and ϕ, the new velocity is evaluated from Equations (45) and (46).
4. By using ϕ, the new pressure is updated from Equation (47).
5. Note that there are three such steps within one time-step because of the adoption of the third-order RK scheme. The third step produces the velocity and pressure fields at the time step t_{n+1}.
6. The whole procedure is repeated (outer iterations) and the solution is updated until an overall convergence is achieved or an initially designed stop-time is reached.

4. RESULTS AND DISCUSSION

In this section, three benchmark numerical examples are chosen to demonstrate the performance of the present DSC–SOR approach for the solution of incompressible flows. The first test example is the Taylor problem, a two-dimensional Navier–Stokes equation with periodic boundary conditions for incompressible flows. This problem has an exact solution which enables us to validate the computational accuracy and efficiency of the present approach. The second example is a driven cavity flow, a standard problem for validating new numerical algorithms. The third test example is a double periodic shear layer flow in a square box. The latter two examples demonstrate the stability, reliability and robustness of the present method

for solving incompressible flow problems. All computations were carried out on DEC/Compaq workstations.

4.1. The Taylor problem

This example is used to check the accuracy and efficiency of the present DSC–SOR method for the Navier–Stokes equations. Here we consider the two-dimensional Navier–Stokes equations presented in Equations (19), (20) and (21) in a unit square $[0, 2\pi] \times [0, 2\pi]$ and the periodic boundary conditions. The initial velocity fields are given by

$$u(x, y, 0) = -\cos(x)\sin(y) \tag{48}$$

$$v(x, y, 0) = \sin(x)\cos(y) \tag{49}$$

The exact solution for this problem is known as

$$u(x, y, t) = -\cos(x)\sin(y)\exp\left(-\frac{2t}{Re}\right) \tag{50}$$

$$v(x, y, t) = \sin(x)\cos(y)\exp\left(-\frac{2t}{Re}\right) \tag{51}$$

$$p(x, y, t) = -\frac{1}{4}[\cos(2x) + \cos(2y)]\exp\left(-\frac{4t}{Re}\right) \tag{52}$$

In the present computation, we set $\Delta x = \Delta y = \Delta = 1/(N-1)$ with $N = 33$ and 65. The Reynolds numbers ranging from $Re = 20$ to ∞ are calculated. A fixed time increment $\Delta t = 0.001$ is used. In order to examine the influence of bandwidth W on the accuracy, three values of $W = 8, 16$ and 32, with corresponding regularization parameters being $\sigma/\Delta = 1.73$, 2.46 and 3.2, respectively, are considered. It is quite efficient to set the convergence constant as $\varepsilon = 10^{-12}$ with the over relaxation factor ranging between 1.2 and 1.8. To implement a single convolution kernel, the values of u, v, u^*, v^* and p outside the computational domain are obtained from those inside the computational domain by using the periodic condition.

Table I shows the L_2 errors between present computed velocity field and the exact velocity field of Equations (50) and (51) at two Reynolds numbers corresponding to four different grid spacings. The solution is computed up to $t = 2$. The results of E and Shu [32] are also listed in Table I for a comparison. Their results are obtained by using a high-order essentially non-oscillatory (ENO) scheme. It is evident that the accuracy of present results is much higher than that of E and Shu. Both results improve as the mesh is refined. The accuracy of present results is getting better as the Reynolds number increases, because the velocity and pressure fields vary slowly at high Reynolds numbers. In contrast, the accuracy of E and Shu deteriorates at high Reynolds numbers. It is also obvious that, the accuracy of the DSC algorithm increases with the bandwidth. This property has important implication in practical computations: one could select an appropriate bandwidth to achieve a desired level of accuracy in a numerical computation without changing one's computer code. Even with a small bandwidth and a relatively coarse mesh (8 and 33×33), the DSC has already given a result compatible with

Table I. L_2 errors for the Taylor problem ($t = 2$).

	DSC			ENO
N	$W = 8$	$W = 16$	$W = 32$	
$Re = 20$				
33	2.23(−4)	1.12(−9)	8.36(−10)	
65	8.19(−5)	6.49(−10)	9.37(−11)	2.93(−3)
129				1.80(−4)
257				1.10(−5)
$Re = \infty$				
33	7.68(−13)	4.68(−13)	4.47(−13)	
65	2.34(−13)	2.22(−13)	2.10(−13)	
129				1.46(−3)
257				1.11(−4)

that of the ENO at a much finer grid. All these results indicate that the present DSC–SOR method is extremely accurate and reliable for solving the Navier–Stokes equation with periodic boundary conditions. In fact, in an earlier direct approach [21], the DSC algorithm attained an even smaller error ($\sim 10^{-15}$) for the case of $Re = \infty$ by using 33 grid points in each dimension. However, the present iterative algorithm requires much smaller computer memory for similar calculations and thus has the potential for being used in large-scale computations.

4.2. Driven cavity flow

The driven cavity flow is another standard test problem for evaluating the reliability and efficiency of potential Navier–Stokes equation solvers [33–38]. The computational domain is a unit box $[0, 1] \times [0, 1]$. This simple geometry makes it possible to compare different numerical methods in details. The incompressible condition is adopted in the computational domain and the no-slip velocity boundary condition is applied to all boundaries with the tangential velocity value at the top boundary (the velocity of the moving lid) being set to 1.

In fact, the problem is non-trivial, particularly at high Reynolds numbers, due to the occurrence of complex flow structure and small-scale vortices. A large primary eddy is formed in the cavity and it gradually moves towards the centre, as the Reynolds number increases. Two secondary eddies are lodged in the bottom dowstream and upstream corners of the cavity. At sufficiently high Reynolds numbers, a third secondary eddy occurs close to top upstream corner. Various tertiary vortices in both bottom left and right corners appear when the Reynolds number is close to 10 000.

In the present DSC–SOR approach, the velocity at the corner nodes on the top boundary is fixed to zero to avoid inflow and outflow through the corners. In order to have a unique solution, the potential function ϕ value at the midpoint of the bottom side is fixed to zero. The velocity and pressure initial conditions are set to zero inside the domain. The computational bandwidth is chosen as $W = 32$. The regularization parameter is set to $\sigma = 3.2\Delta$. The over relaxation factor is taken as $\omega = 1.8$, and the convergent parameter of $\varepsilon = 10^{-6}$ is used. The values of u, v, u^*, v^* outside the computational domain are set to their corresponding values on the box boundary, while p values outside the computational domain are obtained by symmetric extensions from their values inside the computational domain. In the present

Table II. Results for u-velocity along the vertical centerline at $Re = 1000$ with different meshes.

	DSC				Ghia *et al.*
y	33×33	65×65	129×129	201×201	129×129
1.0000	1.00000	1.00000	1.00000	1.00000	1.00000
0.9766	0.63025	0.67498	0.66990	0.66706	0.65928
0.9688	0.55464	0.58358	0.58151	0.58065	0.57492
0.9609	0.49769	0.51348	0.51365	0.51401	0.51117
0.9531	0.45744	0.46305	0.46534	0.46705	0.46604
0.8516	0.34253	0.32052	0.32743	0.33010	0.33304
0.7344	0.19718	0.18255	0.18467	0.18542	0.18719
0.6172	0.06368	0.05686	0.05635	0.05630	0.05702
0.5000	−0.05594	−0.05646	−0.05953	−0.06023	−0.06080
0.4531	−0.10207	−0.10031	−0.10440	−0.10538	−0.10648
0.2813	−0.27793	−0.26568	−0.27281	−0.27495	−0.27805
0.1719	−0.37878	−0.36364	−0.37407	−0.37863	−0.38289
0.1016	−0.28356	−0.27728	−0.28531	−0.28996	−0.29730
0.0703	−0.20935	−0.20577	−0.21110	−0.21468	−0.22220
0.0625	−0.18884	−0.18665	−0.19143	−0.19474	−0.20196
0.0547	−0.16653	−0.16663	−0.17125	−0.17430	−0.18109
0.0000	0.00000	0.00000	0.00000	0.00000	0.00000

work, simulations are performed over a range of Reynolds numbers ($Re = 100$, 400, 1000, 3200, 5000, 7500 and 10 000). In all calculations, a small time increment ($\Delta t = 0.0001$) is initially given, then the time increment is automatically optimized by using the CFL stability condition of Equation (34) for the rest of the computations. Solutions for small Reynolds numbers ($Re = 100$, 400) are obtained by using $N = 65$ in each dimension, while $N = 129$ in each dimension is utilized for higher Reynolds numbers ($Re = 1000$, 3200). A fine mesh of 201×201 was used to simulate flows at Reynolds numbers of $Re = 5000, 7500$ and 10 000.

In order to investigate the reliability and the grid-independence nature of the DSC simulations, three different meshes are employed for the case of $Re = 1000$ to show that the present solution is indeed grid independent. Tables II and III list the velocity values along the geometrical centre for the case of $Re = 1000$ at different meshes together with the solution of Ghia *et al.* [34] for a comparison. Even at a mesh of 33×33, the agreement is very good and the numerical solution is essentially grid independent as the mesh reaches 129×129. The streamline contour of this case is plotted in Figure 1, which shows the same characteristic as that of Ghia *et al.* [34]. The convergence history of the case $Re = 1000$ is given in Figure 2 for a mesh of 129×129. The residual of velocity is defined as $\max(|u_{i,j}^{n+1} - u_{i,j}^{n}|, |v_{i,j}^{n+1} - v_{i,j}^{n}|)$. We assume that a converged state is reached if the residual is smaller than a given positive constant. It is seen from Figure 2 that the present simulation converges well with respect to time integration.

To further validate the present algorithm, velocity profiles at Reynolds numbers of 100, 400 and 3200 are depicted in Figure 3. It is found that for low Reynolds numbers of 100 and 400, a mesh of 65×65 gives results which compare well with those of Ghia *et al.* [34] obtained by using a larger mesh (129×129). The present simulation provides an excellent comparison with that of Ghia *et al.* [34] at $Re = 3200$, see Figures 3(c) and 3(d).

Table III. Results for v-velocity along the horizontal centerline at $Re = 1000$ with different meshes.

	DSC				Ghia *et al.*
x	33×33	65×65	129×129	201×201	129×129
1.0000	0.00000	0.00000	0.00000	0.00000	0.00000
0.9688	−0.19627	−0.20509	−0.21209	−0.21680	−0.21388
0.9609	−0.26065	−0.26615	−0.27516	−0.28072	−0.27669
0.9531	−0.32182	−0.32408	−0.33486	−0.34100	−0.33714
0.9453	−0.37737	−0.37714	−0.38871	−0.39524	−0.39188
0.9063	−0.51362	−0.49748	−0.50918	−0.51464	−0.51550
0.8594	−0.43031	−0.40692	−0.41615	−0.41943	−0.42665
0.8047	−0.31860	−0.30178	−0.31024	−0.31313	−0.31966
0.5000	0.02962	0.02719	0.02626	0.02599	0.02526
0.2344	0.32570	0.30996	0.31648	0.31904	0.32235
0.2266	0.33347	0.31756	0.32454	0.32730	0.33075
0.1563	0.36281	0.34975	0.36143	0.36628	0.37095
0.0938	0.30735	0.30284	0.31593	0.32139	0.32627
0.0781	0.28505	0.28009	0.29315	0.29854	0.30353
0.0703	0.27096	0.26688	0.27972	0.28505	0.29012
0.0625	0.25387	0.25159	0.26436	0.26965	0.27485
0.0000	0.00000	0.00000	0.00000	0.00000	0.00000

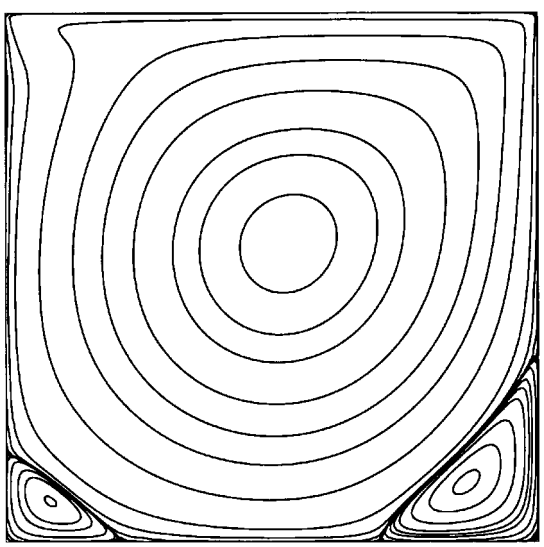

Figure 1. Streamline contours at $Re = 1000$, with a uniform mesh of 129×129.

Figure 4 presents converged streamline contour plots at four different Reynolds numbers. In general, the centre of the primary eddy is initially located at the top right corner and soon moves towards the geometric centre of the cavity as the time increases at low Reynolds numbers. As the Reynolds number increases, the primary eddy becomes relatively smaller

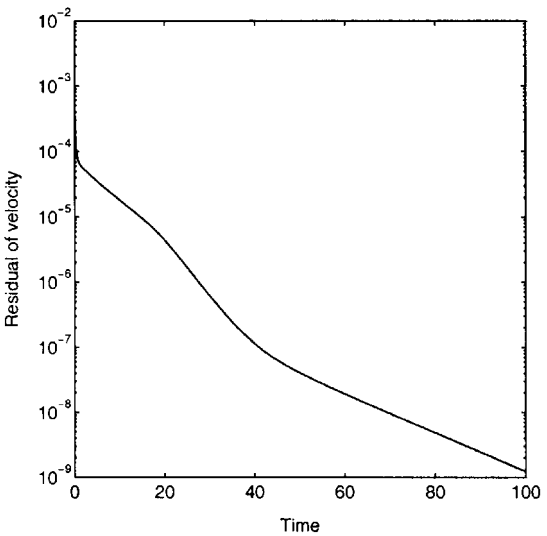

Figure 2. Convergence history of the residual of velocity at $Re = 1000$ with a uniform mesh of 129×129.

and moves towards the steady-state position slowly, whereas the relative size of secondary eddies increases. For the cases of $Re = 3200$, 5000 and 7500, the third secondary eddy appears near the upstream top corner. Initially, all secondary eddies appear very small at corners (or near the wall for the case of the third secondary eddy), and they grow up to steady-state sizes as time increases. These results are in excellent agreement with those in the literature [34].

Although Ghia *et al.* [34] presented steady-state solution for driven cavity flow up to $Re = 10\,000$, it is well-known that the system becomes unstable around $Re = 7763$ [39]. Recently, Liffman [40] confirmed this unsteady character at $Re = 10\,000$ by using a highly accurate collocation spectral solver. Here we test our approach by considering the case of $Re = 10\,000$. To remove the corner singularities, we adopt a new initial velocity for the upper lid of the cavity

$$u(x_i, 1.0, 0) = 1.0 - \exp[-20.0(1.0 - x_i^2)] \tag{53}$$

$$v(x_i, 1.0, 0) = 0 \tag{54}$$

The mesh size adopted in this case is still 201×201, and other parameters used are the same as those stated earlier. Apart from the usual central primary vortex, the secondary vortices at the two bottom corners and the right top corner, there are tertiary vortices in left and right corners. The flow is also found to be unsteady and gives a quasi-periodic pattern. These results are in excellent agreement with those of Liffman [40].

4.3. Periodic shear layer flow

The last problem is a double shear layer flow [41], which is also a benchmark model for testing the reliability and numerical resolution of potential numerical algorithms for incompressible

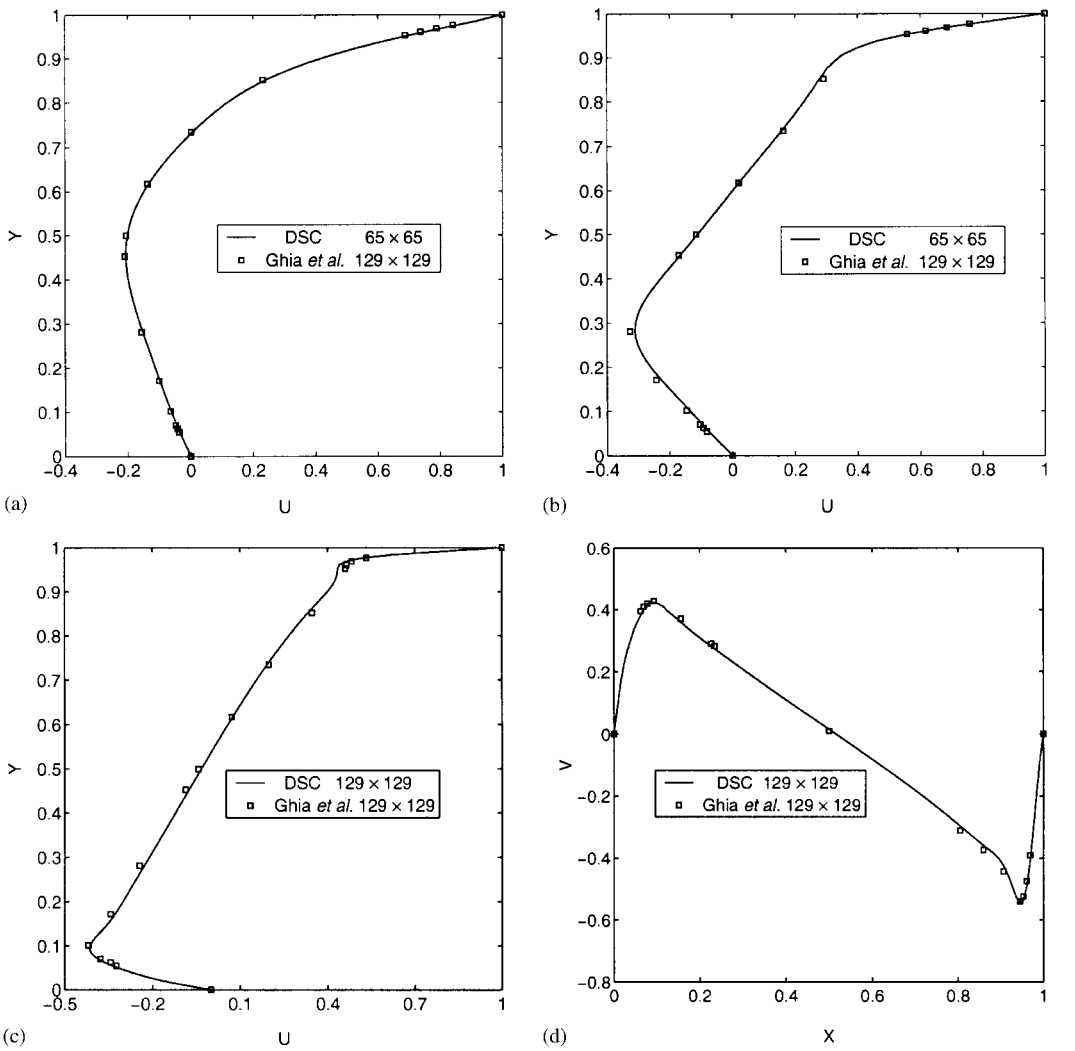

Figure 3. Comparison of velocity profiles, (a) $Re = 100$, horizontal velocity along the vertical centreline; (b) $Re = 400$, horizontal velocity along the vertical centreline; (c) $Re = 3200$, horizontal velocity along the vertical centreline; (d) $Re, = 3200$, vertical velocity along the horizontal centreline.

flows [22, 32, 38, 41]. In particular, an earlier direct DSC algorithm [22] was found to provide excellent results for this problem with a very small mesh size. Furthermore, this is an inviscid flow problem, governed by the Euler equation (i.e., the case of $Re = \infty$ for the Navier–Stokes equation). The computational domain is defined on the square box $[0, 2\pi] \times [0, 2\pi]$ with periodic boundary conditions in both horizontal and vertical directions. From the Taylor problem, we know that the present algorithm is extremely accurate for the Euler equation.

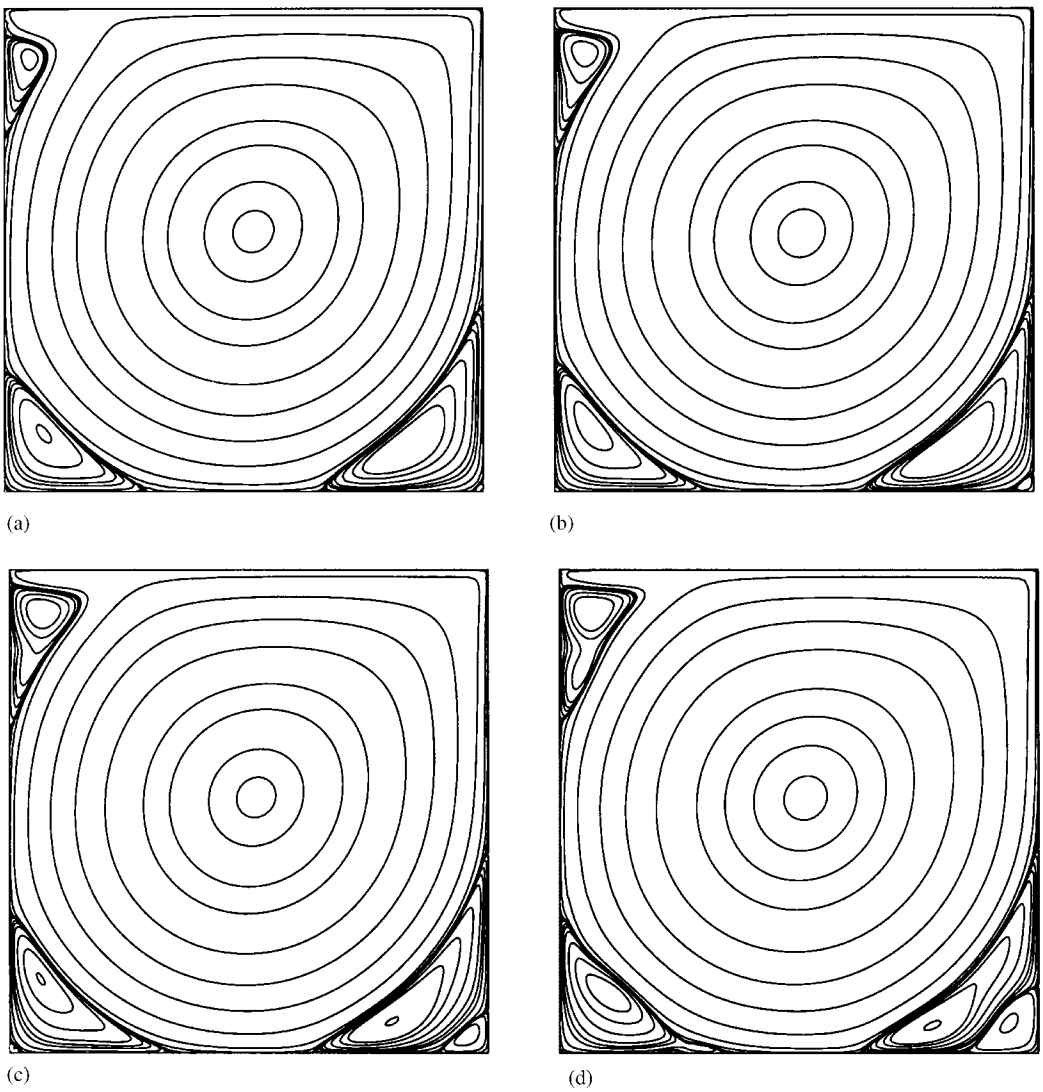

Figure 4. Contours of streamlines at different Reynolds numbers: (a) $Re = 3200,\ 129 \times 129$; (b) $Re = 5000;\ 201 \times 201$; (c) $Re = 7500,\ 201 \times 201$; (d) $Re = 10\,000,\ 201 \times 201$.

However, the initial velocity field is different in the present case

$$u(x, y, 0) = \begin{cases} \tanh\left(\dfrac{y - \pi/2}{\rho}\right) & y \leqslant \pi \\ \tanh\left(\dfrac{3\pi/2 - y}{\rho}\right) & y > \pi \end{cases} \tag{55}$$

$$v(x, y, 0) = \eta \sin(x) \tag{56}$$

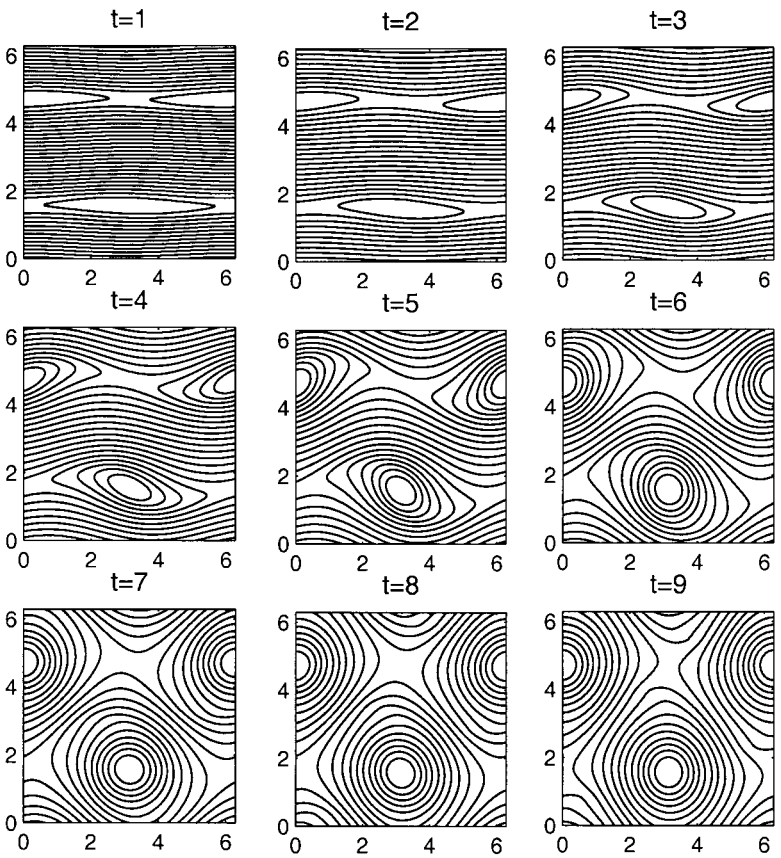

Figure 5. Select streamline contours for the periodic shear layer flow with $A = 15$ and a mesh of 129×129.

where $\rho = \pi/A$ (A is a positive constant) and $\eta = 0.05$. It is well known that this initial flow field describes horizontal shear layers of finite thickness that are perturbed by a small amplitude vertical velocity. The evolution under the Euler equation leads to a periodic array of large vortices, with the shear layer between the rolls being thinned by the large straining field. This problem becomes very challenging as the parameter A increases. Because the initial flow field becomes discontinuous at the limit of $A \to \infty$. The solution quickly develops into multiple roll-ups with smaller and smaller scales. The resolution is lost eventually with a fixed grid size.

In the present computation, we choose $W = 32$, $\sigma = 3.2\Delta$, $\omega = 1.8$ and $\varepsilon = 10^{-8}$. Periodic boundary extensions are used to treat the values of u, v, u^*, v^* and p outside the computational domain. We consider $A = 15$ in this work which was calculated on a uniform mesh of $\Delta x = \Delta y = \Delta = 2\pi/(N-1)$ with $N = 129$. Time increment is set to $\Delta t = 0.001$ initially and later it is automatically optimized according to the CFL stability condition for the rest of the computation.

Streamline plots in Figure 5 illustrate vividly how double shear layers evolve into a periodic array of vortices. Figure 6 depicts the vorticity contours. The shear layers quickly develop

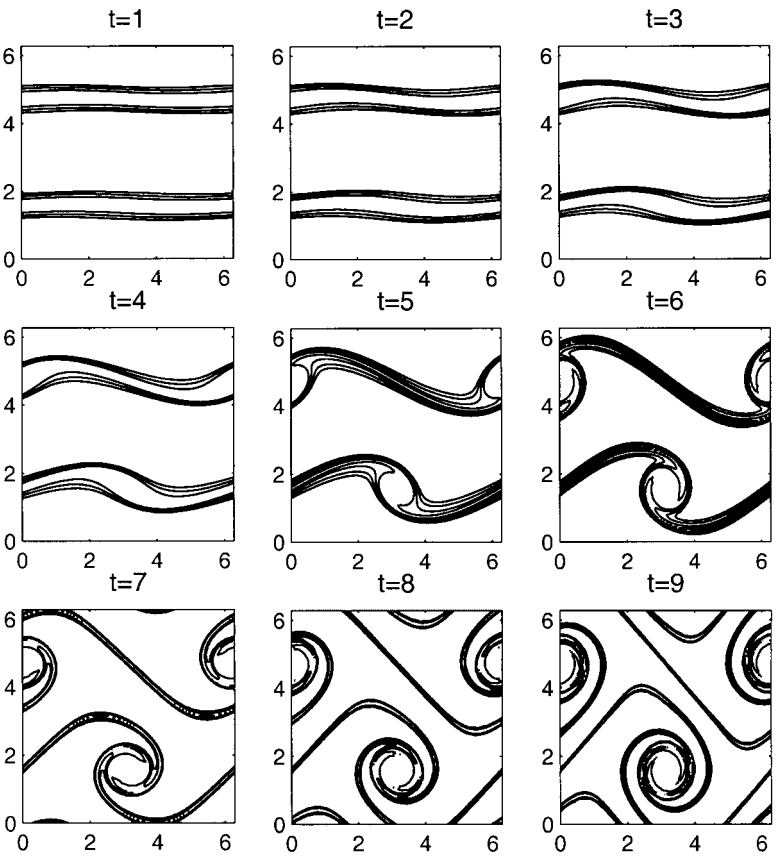

Figure 6. Select vorticity contours for the periodic shear layer flow with $A = 15$ and a mesh of 129×129.

into roll-ups that are stretched and thinned by the large straining field. Similar behaviour was observed by previous researchers [32, 38]. These results are in excellent agreement with those in earlier references [32, 38, 41]. It should be pointed out that our calculation starts to lose resolution at $t = 10$, which is indicated by the occurrence of wiggles in the vorticity contour plots. Similar phenomenon can be noted in previous results [41], which was computed by using a spectral collocation method with a much larger (512×512) mesh size. We have checked that the kinetic energy is well-preserved in our computations. With the increase in the value of A, the problem becomes more and more challenging. Although we could refine the mesh to achieve results with a satisfactory resolution, the solution of this problem may have to follow the trail of a recently proposed algorithm with the capability of shock capture [42], so that vortex structures with high resolution can be attained.

5. CONCLUSION

In this work, a discrete singular convolution (DSC) algorithm is developed for simulating incompressible flows. The method of fractional time steps in association with a potential

function is employed for the time stepping and for the treatment of the pressure field. The resulting Neumann–Poisson equation for the potential function is solved by the successive over-relaxation (SOR) algorithm. Three benchmark incompressible flow problems are employed to explore the reliability, to test the accuracy and to demonstrate the efficiency of the present DSC–SOR method. The accuracy obtained in our calculation of the analytically solvable Taylor problem is extremely high. The driven cavity flow problem is utilized to test the reliability of the present method for handling no-slip boundary conditions. The validity of the present method is checked by a comparison of velocity data at $Re = 1000$ with benchmark ones in the literature [34]. The convergence history is examined and velocity profiles are compared with those of Ghia *et al.* [34] at a few different Reynolds numbers. Both primary and secondary eddies attained in the present computations are in excellent agreement with those of other established methods [34–36] in the field. Our results of $Re = 10\,000$ are in excellent agreement with those of Liffman [40] obtained by using a spectral collocation solver. A periodic shear layer flow describing the vortices formation of jet motion is employed to test present algorithm for resolving small-scale singularities. This is a very challenging problem, although its geometry and boundary conditions are very simple. Both spectral methods and local methods encounter difficulties in handling this problem. It is found that the present approach provides competitive results for this example. Our results indicate that the present DSC based algorithm provides an alternative approach for solving incompressible flow problems.

ACKNOWLEDGEMENTS

This work was supported by the National University of Singapore.

REFERENCES

1. Cooley JW, Tukey JW. An algorithm for the machine calculation of complex Fourier series. *Mathematics of Computation* 1965; **19**:297–301.
2. Orszag SA. Comparison of pseudospectral and spectral approximations. *Studies in Applied Mathematics* 1972; **51**:253–259.
3. Fornberg B. On Fourier method for the integration of hyperbolic equations. *SIAM Journal on Numerical Analysis* 1975; **12**:509–528.
4. Canuto C, Hussaini MY, Quarteroni A, Zang TA. *Spectral Methods in Fluid Dynamics.* Springer: Berlin, 1988.
5. Forsythe GE, Wasow WR. *Finite-difference Methods for Partial Differential Equations.* Wiley: New York, 1960.
6. Isaacson E, Keller HB. *Analysis of Numerical Methods.* Wiley: New York, 1966.
7. Zienkiewicz OC. *The Finite Element Method in Engineering Science.* McGraw-Hill: London, 1971.
8. Fenner RT. *Finite Element Methods for Engineers.* Imperial College Press: London, 1996.
9. Liu WK. Wavelet and multiple scale reproducing kernel methods. *International Journal for Numerical Methods in Fluids* 1995; **21**:901–931.
10. Beylkin G, Keiser J. On the adaptive numerical solution of nonlinear partial differential equations in wavelet bases. *Journal of Computational Physics* 1997; **132**:233–259.
11. Wei GW. Discrete singular convolution for the solution of the Fokker–Planck equations. *Journal of Chemical Physics* 1999; **110**:8930–8942.
12. Wei GW. A unified approach for solving the Fokker–Planck equation. *Journal of Physics A: Mathematics and General* 2000; **33**:4935–4953.
13. Wei GW. Wavelet generated by using discrete singular convolution kernels. *Journal of Physics A: Mathematics and General* 2000; **33**:8577–8596.
14. Schwarz L. *Theore des Distributions.* Hermann: Paris, 1951.
15. Wei GW, Zhang DS, Kouri DJ, Hoffman DK. Lagrange distributed approximating functionals. *Physical Review Letters* 1997; **79**(5):775–779.

16. Wei GW. Discrete singular convolution method for the sine-Gordon equation. *Physica D* 2000; **137**:247–259.
17. Wei GW. Solving quantum eigenvalue problems by discrete singular convolution. *Journal of Physics B*: *Atomic, Molecular and Optical Physics* 2000; **33**:343–352.
18. Wei GW. Vibration analysis by discrete singular convolution. *Journal of Sound and Vibration* 2001; **244**: 535–553.
19. Wei GW. Discrete singular convolution for beam analysis. *Engineering Structures* 2001; **23**:1045–1053.
20. Guan S, Lai C-H, Wei GW. Bessel–Fourier analysis of patterns in a circular domain. *Physica D* 2001; **151**: 83–98.
21. Wei GW. A new algorithm for solving some mechanical problems. *Computational Methods in Applied Mechanics and Engineering* 2001; **190**:2017–2030.
22. Wei GW. A unified method for computational mechanics. In *Computational Mechanics for the Next Millennium*, Wang CM, Lee KH, Ang KK (eds). Elsevier: New York, 1999; 1049–1054.
23. Hou ZJ, Wei GW. A new approach for edge detection. *Pattern Recognition*, 2002, in press.
24. Qian LW, Wei GW. A note on regularized Shannon's sampling formulae. *Journal of Approximation Theory*, 1999, submitted.
25. Patankar SV. *Numerical Heat Transfer and Fluid Flow.* McGraw-Hill: New York, 1980.
26. Fletcher CAJ. *Computational Techniques for Fluid Dynamics—Fundamental and General Techniques.* Springer: Berlin, 1988.
27. Harlow FH, Welch JE. *The Physics of Fluids* 1965; **8**:2182–2189.
28. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Mathematics of Computation* 1968; **22**:745.
29. Peyret R, Taylor TD. *Computational Methods for Fluid Flow.* Springer Series in Computational Physics. Springer: Berlin, 1980.
30. Hirsch C. *Numerical Computations of Internal and External Flows*, vol. I. Wiley: New York, 1991.
31. Ferziger JH, Peric M. *Computational Methods for Fluid Dynamics.* Springer: New York, 1996.
32. E W, Shu C-W. A numerical resolution study of high order essentially non-oscillatory schemes applied to incompressible flow. *Journal of Computational Physics* 1994; **110**:39–46.
33. E W, Liu J-G. Essentially compact schemes for unsteady viscous incompressible flows. *Journal of Computational Physics* 1996; **126**:122–138.
34. Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; **48**:387–411.
35. Guermond JL, Quartapelle L. Calculation of incompressible viscous flows by an unconditionally stable projection FEM. *Journal of Computational Physics* 1997; **132**:12–23.
36. Schreiber R, Keller HB. Driven cavity flows by efficient numerical techniques. *Journal of Computational Physics* 1983; **49**:310–333.
37. Iwatsu R, Ishii K, Kawamura T, Kuwahara K. Hyun JM. Numerical simulation of three-dimensional flow structure in a driven cavity. *Fluid Dynamics Research* 1989; **5**:173–189.
38. Lou ZJ, Ferraro R. A parallel incompressible flow solver package with a parallel multigrid elliptic kernel. *Journal of Computational Physics* 1996; **125**:225–243.
39. Poliashenko M, Aidun CK. A direct method for computation of simple bifurcations. *Journal of Computational Physics* 1995; **121**:246–260.
40. Liffman K. Comments on a collocation spectral solver for the Helmholtz equation. *Journal of Computational Physics* 1996; **128**:254–258.
41. Bell J, Colella P, Glaz H. A second-order projection method for the incompressible Navier–Stokes equations. *Journal of Computational Physics* 1989; **85**:257–283.
42. Wei GW. Synchronization of single-side averaged coupling and its application to shock capturing. *Physical Review Letters* 2001; **86**:3542–3545.

