Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER Applied Mathematics and Computation 164 (2005) 679–695

APPLIED MATHEMATICS AND COMPUTATION

www.elsevier.com/locate/amc

Fast direct solver for the biharmonic equation on a disk and its application to incompressible flows

Ming-Chih Lai [a,*], Hsi-Chi Liu [b]

[a] *Department of Applied Mathematics, National Chiao Tung University, 1001, Ta Hsueh Road, Hsinchu 300, Taiwan*

[b] *Department of Mathematics, National Chung Cheng University, Minghsiung, Chiayi 621, Taiwan*

Abstract

We develop a simple and efficient FFT-based fast direct solver for the biharmonic equation on a disk. The biharmonic equation is split into a coupled system of harmonic problems. We first use the truncated Fourier series expansion to derive a set of coupled singular ODEs, then we solve those singular equations by second-order finite difference discretizations. Using a radial grid with shifting a half mesh away from the origin, we can handle the coordinate singularity easily without pole conditions. The Sherman–Morrison formula is then applied to solve the resultant linear system in a cost-efficient way. The computational complexity of the method consists of $O(MN \log_2 N)$ arithmetic operations for $M \times N$ grid points. The numerical accuracy check and some applications to the incompressible Navier–Stokes flows inside a disk are conducted.

Keywords: Biharmonic equation; Polar coordinates; Sherman–Morrison formula; FFT; Vorticity stream function formulation

[*] Corresponding author.
E-mail address: mclai@math.nctu.edu.tw (M.-C. Lai).

0096-3003/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.amc.2004.04.064

1. Introduction

The biharmonic problem arises from several applications in solid mechanics and fluid mechanics. In solid mechanics, finding the displacement of the bending of elastic plates involves solving the biharmonic equation. In fluid mechanics, the stream function of incompressible Stokes flow in two-dimensional space is the solution of a biharmonic equation as well.

The biharmonic problem (of the first kind) on a two-dimensional domain Ω has the form

$$\Delta^2 u(x,y) = f(x,y) \quad \text{in } \Omega, \tag{1.1}$$

$$u = g(x,y), \quad \frac{\partial u}{\partial n} = h(x,y) \quad \text{on } \partial\Omega, \tag{1.2}$$

where the differential operator Δ is the well-known Laplacian defined by $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ and $\Delta^2 u = \Delta(\Delta\, u)$. As mentioned in [2], direct discretization of this fourth-order biharmonic equation (1.1) leads to a very ill-conditioned linear equations. Thus, most of the iterative methods require a large number of iterations in order to obtain some satisfactory solutions. One popular approach to avoid solving such ill-conditioned matrix equations is to introduce an auxiliary variable $v(x,y) = \Delta u(x,y)$ and to split the biharmonic equation (1.1) into a coupled system of Poisson equations as

$$\Delta u(x,y) = v(x,y), \quad \Delta v(x,y) = f(x,y). \tag{1.3}$$

The boundary conditions (1.2) are still the same. One can easily see that under this formulation, there are two boundary conditions for the solution u but no boundary conditions for v. The functions u and v are coupled through the boundary conditions implicitly which turns out to be the main difficulty of solving such problem.

Throughout this paper, we are interested in the domain of a unit disk $\Omega = \{(x,y) : x^2 + y^2 < 1\}$. Therefore, it is natural to apply the polar coordinate transformation $x = r\cos\theta$, $y = r\sin\theta$ to the equations. For simplicity, we use the same notations to represent the functions both in Cartesian and polar coordinates. The coupled system of the biharmonic equation of $u(r,\theta)$ can be written as

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = v(r,\theta), \quad 0 < r < 1, \ 0 \leqslant \theta < 2\pi, \tag{1.4}$$

$$\frac{\partial^2 v}{\partial r^2} + \frac{1}{r}\frac{\partial v}{\partial r} + \frac{1}{r^2}\frac{\partial^2 v}{\partial \theta^2} = f(r,\theta), \quad 0 < r < 1, \ 0 \leqslant \theta < 2\pi, \tag{1.5}$$

$$u(1,\theta) = g(\theta), \quad \frac{\partial u}{\partial r}(1,\theta) = h(\theta). \tag{1.6}$$

Various approaches for the numerical solution of the boundary value problem (1.4)–(1.6), along with the applications to the steady incompressible flow inside circular geometries have been developed in the literature. Those include the spectral method [6], the integral equation method [4], and the spectral/difference method [1,5]. In this paper, we shall develop a fast direct method similar to [1] but differs with the treatments of coordinate singularity (with pole conditions [1] vs. without pole conditions) and the boundary conditions (global integral conditions [1] vs. local difference approximations). Besides, our resultant linear equations can be solved in an efficient algorithm. Unlike those papers aforementioned, we apply the present biharmonic solver to study the unsteady incompressible Navier–Stokes flows.

Our method is a FFT-based fast direct solver for the biharmonic equation (1.4)–(1.6). We first use the truncated Fourier series expansion to derive a set of coupled singular ODEs, then we solve those singular equations by second-order finite difference discretizations. Using a radial grid with shifting a half mesh away from the origin, we can handle the coordinate singularity easily without pole conditions. The Sherman–Morrison formula is then applied to solve the resultant linear system in a cost-efficient way. The computational complexity of the method consists of $O(MN \log_2 N)$ arithmetic operations for $M \times N$ grid points.

The rest of the paper is organized as follows. In Section 2, we present our fast direct solver for the biharmonic equation (1.4)–(1.6). We then apply this solver to develop a numerical scheme for the unsteady incompressible Navier–Stokes flows inside a disk in Section 3. The numerical accuracy check and some test applications have been performed in Section 4.

2. FFT-based fast biharmonic solver

2.1. Fourier mode equations

Since the solution u in Eqs. (1.4) and (1.5) is periodic in θ, we can approximate it by the truncated Fourier series as

$$u(r,\theta) = \sum_{k=-N/2}^{N/2-1} u_k(r)\mathrm{e}^{\mathrm{i}k\theta}, \tag{2.1}$$

where $u_k(r)$ is the complex Fourier coefficient given by

$$u_k(r) = \frac{1}{N}\sum_{j=1}^{N} u(r,\theta_j)\mathrm{e}^{-\mathrm{i}k\theta_j}, \tag{2.2}$$

$\theta_j = 2j\pi/N$, and N is the number of grid points along a circle. The functions $v(r,\theta)$, $f(r,\theta)$, $g(\theta)$, and $h(\theta)$ are defined in the same manner as Eqs. (2.1) and (2.2). The above transformation between the physical space and Fourier space can be efficiently performed using the fast Fourier transform(FFT) with $O(N\log_2 N)$ arithmetic operations.

Substituting those expansions into Eqs. (1.4)–(1.6), we reduce the original PDE to a set of singular ODEs. This common approach is known as the separation of variables in the solution of the linear PDEs. The kth Fourier coefficients $u_k(r)$ and $v_k(r)$ now satisfy the boundary value problems

$$u_k'' + \frac{u_k'}{r} - \frac{k^2}{r^2}u_k = v_k(r), \tag{2.3}$$

$$v_k'' + \frac{v_k'}{r} - \frac{k^2}{r^2}v_k = f_k(r), \tag{2.4}$$

$$u_k(1) = g_k, \quad u_k'(1) = h_k, \tag{2.5}$$

where the prime denotes the derivative with respect to r. The remaining task is to solve those coupled Fourier mode equations for u_k and v_k by second-order finite difference discretizations.

2.2. Spatial discretization and boundary conditions

Throughout this paper, we adapt a radial grid by shifting half mesh width away the origin as

$$r_i = (i - 1/2)\Delta r, \quad i = 1, 2, \ldots M, M+1, \tag{2.6}$$

where the mesh width $\Delta r = 2/(2M + 1)$. Under such grid, we have $r_{M+1} = 1$. The advantage of this grid is that we do not place grid points directly at the origin; thus, as we shall see, the numerical boundary value near the origin is not needed. This radial grid has been intensively used to develop the efficient Poisson solvers in 2D polar [9,7] and 3D cylindrical [10] geometries, and to simulate the compressible Navier–Stokes and Euler equations [11].

For convenience of presentation, we simply denote the discrete values $U_i \approx u_k(r_i)$, $V_i \approx v_k(r_i)$, and $F_i \approx f_k(r_i)$. Using second-order centered difference approximations to discretize Eqs. (2.3) and (2.4), we obtain the difference equations

$$\frac{U_{i+1} - 2U_i + U_{i-1}}{\Delta r^2} + \frac{1}{r_i}\frac{U_{i+1} - U_{i-1}}{2\Delta r} - \frac{k^2}{r_i^2}U_i = V_i, \tag{2.7}$$

$$\frac{V_{i+1} - 2V_i + V_{i-1}}{\Delta r^2} + \frac{1}{r_i}\frac{V_{i+1} - V_{i-1}}{2\Delta r} - \frac{k^2}{r_i^2}V_i = F_i \tag{2.8}$$

for the index $1 \leqslant i \leqslant M$.

In order to close the linear system, the numerical boundary values U_0, V_0, U_{M+1} and V_{M+1} should be supplied. Choosing r_i as described in (2.6), we have $r_1 = \Delta r/2$. When the index of $i = 1$, the coefficients of U_0 and V_0 in the difference equations (2.7) and (2.8) equal to zero, respectively; thus, no approximations for U_0 and V_0 are actually needed. This is the advantage of using a shifted grid (2.6). The outer numerical boundary value U_{M+1} is known by the boundary value g_k. However, there is no direct given value for V_{M+1}. One simple way to obtain the numerical boundary value V_{M+1} is by the local finite difference approximation as follows.

First, let us impose Eq. (2.7) at the index $i = M + 1$; that is, we have

$$V_{M+1} = \frac{U_{M+2} - 2U_{M+1} + U_M}{\Delta r^2} + \frac{1}{r_{M+1}}\frac{U_{M+2} - U_M}{2\Delta r} - \frac{k^2}{r_{M+1}^2}U_{M+1}, \tag{2.9}$$

where U_{M+2} is the ghost value outside the computational domain. Approximating the boundary condition $u_k'(1) = h_k$ by the second-order difference formula, we have

$$\frac{U_{M+2} - U_M}{2\Delta r} = h_k. \tag{2.10}$$

Therefore, the ghost value U_{M+2} can be obtained by the formula $U_{M+2} = U_M + 2\Delta r h_k$. Substituting the value of U_{M+2}, $U_{M+1}(=g_k)$, and $r_{M+1} = 1$ into Eq. (2.9), we derive the numerical boundary value for V_{M+1} as

$$V_{M+1} = \frac{2}{\Delta r^2}U_M + \left(-\frac{2}{\Delta r^2} - k^2\right)g_k + \left(\frac{2}{\Delta r} + 1\right)h_k. \tag{2.11}$$

This is also explained that the solutions of U_i and V_i are coupled through the boundary conditions.

2.3. Efficient solver for the resultant linear system

After multiplying Δr^2 in both sides of Eqs. (2.7) and (2.8), the resultant $2M \times 2M$ linear system has the form

$$\begin{bmatrix} T & D \\ E & T \end{bmatrix}\begin{pmatrix} \mathbf{U} \\ \mathbf{V} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{F} \end{pmatrix}. \tag{2.12}$$

Here, T is an $M \times M$ tridiagonal matrix

$$T = \begin{bmatrix} d_1 & s_1 & & & & \\ l_2 & d_2 & s_2 & & & \\ & \cdot & \cdot & \cdot & & \\ & & \cdot & \cdot & \cdot & \\ & & & l_{M-1} & d_{M-1} & s_{M-1} \\ & & & & l_M & d_M \end{bmatrix}, \tag{2.13}$$

whose entries are

$$d_i = -2 - \frac{k^2}{(i-1/2)^2}, \quad s_i = 1 + \frac{1}{2(i-1/2)}, \quad l_i = 1 - \frac{1}{2(i-1/2)} \tag{2.14}$$

for $1 \leqslant i \leqslant M$. The matrix D is the $M \times M$ diagonal matrix with $D = -\Delta r^2 I$. The matrix E has the only one nonzero entry ($E_{M,M} = 2s_M/\Delta r^2$) which represents the coupling of the boundary condition of V with U. One should notice that this coupling comes from the local approximation (2.11). Solving this system by the direct Gaussian elimination needs at least $O(M^2)$ arithmetic operations. However, one can immediately recognize that the matrix E is as simple as a rank-one matrix. Therefore, we can uncouple the linear equations by ignoring the nonzero entry of the matrix E (that is, to replace E by a zero matrix) and solve the new linear system by inverting the tridiagonal matrix T. In such way, the computational cost has been cut to $O(M)$ operations. This idea is completely described by the Sherman–Morrison formula as follows.

Lemma 1 (Sherman–Morrison formula [15]). *If $By = b$ and $Bz = \alpha$, then $(B + \alpha\beta^t)x = b$ has the solution*

$$x = y - \frac{\beta^t y}{1 + \beta^t z} z. \tag{2.15}$$

This formula can be easily verified by direct substitution.

In our implementation of the above formula, the matrices B and $\alpha\beta^t$ are chosen as

$$B = \begin{bmatrix} T & D \\ 0 & T \end{bmatrix}, \quad \alpha\beta^t = \begin{bmatrix} 0 & 0 \\ E & 0 \end{bmatrix}. \tag{2.16}$$

The $2M$ column vectors α and β are chosen as

$$\alpha = (0, 0, \ldots, 0, 2s_M/\Delta r^2)^t \quad \beta = (0, 0, ..1, 0, ..0)^t, \tag{2.17}$$

where the nonzero entries of α and β are at $2M$th and Mth components, respectively. By those choices, we can easily check $E=\alpha\beta^t$. The inversion of the matrix B involves solving the linear system of the tridiagonal matrix T which needs only $O(M)$ operations. One should also notice that the matrix T is the resultant matrix of the Fourier mode equations of Dirichlet Poisson problem which is completely solvable [9].

2.4. Summary of the algorithm

Let us close the section by summarizing the algorithm and the operation counts in the following three steps:

1. Compute the Fourier coefficients for the right-hand side function and the boundary conditions using FFT described in (2.2). This requires $O(MN\log_2 N)$ arithmetic operations.
2. Solve the coupled tridiagonal linear system (2.12) resulting from (2.7) and (2.8) by the Sherman–Morrison formula. This requires $O(MN)$ operations.
3. Convert the Fourier coefficients by inverse FFT (2.1) to obtain the solution, which requires $O(MN\log_2 N)$ operations.

The overall operation count is thus $O(MN\log_2 N)$ for $M\times N$ grid points.

3. Incompressible Navier–Stokes solver on a disk

3.1. Vorticity stream function formulation

The incompressible Navier–Stokes equations has the standard form

$$\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p=\frac{1}{Re}\Delta\mathbf{u}, \tag{3.1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{3.2}$$

where $\mathbf{u}(\mathbf{x},t)$ is the fluid velocity, $p(\mathbf{x},t)$ the pressure, and Re is the Reynolds number. The first equation describes the conservation of momentum and the second one is the conservation of mass. In 2D geometry, we can express the Navier–Stokes equations (3.1) and (3.2) by so called the vorticity stream function formulation. By taking the curl of Eq. (3.1) to eliminate the pressure gradient term, we have

$$\frac{\partial \omega}{\partial t}+J(\omega,\psi)=\frac{1}{Re}\Delta\omega, \tag{3.3}$$

where ω is the nonzero vorticity of the z component, ψ the stream function, defined by $\mathbf{u}=\mathbf{e}_z\times\nabla\psi$, and $J(\omega,\psi)$ is the Jacobian determinant. Note that, the velocity **u** automatically satisfies the incompressibility constraint (3.2). Now using the definition of ω, it yields the relation of ω and ψ by

$$\Delta\psi = \omega. \tag{3.4}$$

Therefore, the original 2D Navier–Stokes equations (3.1) and (3.2) with three primitive variables now has an alternate formulation described by (3.3) and (3.4) with only two unknown variables.

We are interested in the numerical approximations of Eqs. (3.3) and (3.4) in a unit disk geometry $\Omega = \{0 < r \leqslant 1, 0 \leqslant \theta \leqslant 2\pi\}$; thus, the polar coordinates is used. The nonlinear Jacobian describing the vorticity transport is written as

$$J(\omega,\psi) = \frac{1}{r}\left(\frac{\partial\psi}{\partial r}\frac{\partial\omega}{\partial\theta} - \frac{\partial\psi}{\partial\theta}\frac{\partial\omega}{\partial r}\right). \tag{3.5}$$

The radial and azimuthal velocity components can be recovered from the stream function by the formulas

$$u_r = -\frac{1}{r}\frac{\partial\psi}{\partial\theta}, \quad u_\theta = \frac{\partial\psi}{\partial r}. \tag{3.6}$$

The vorticity can be written as

$$\omega = \frac{\partial u_\theta}{\partial r} + \frac{u_\theta}{r} - \frac{1}{r}\frac{\partial u_r}{\partial\theta}. \tag{3.7}$$

We restrict our attention to the flow inside a unit disk with some particular velocity specified on the boundary as $u_r = 0$ and $u_\theta = h(\theta)$ at $r = 1$. (This particular boundary condition corresponds to the non-normal flow condition. In addition, if $h(\theta) = 0$, the velocity is no-slip at the boundary.) From the relation of (3.6), the above boundary conditions become

$$\psi(1,\theta) = 0, \quad \frac{\partial\psi}{\partial r}(1,\theta) = h(\theta). \tag{3.8}$$

So the complete governing equations include Eqs. (3.3) and (3.4) and the boundary conditions (3.8). Again, one can easily see that there are two boundary conditions for the stream function ψ but no boundary condition for the vorticity ω. This is exactly the same situation as the case of solving the coupled system of biharmonic equations (1.4)–(1.6). It should not be surprising since the 2D Navier–Stokes equations can be actually formulated to a time-dependent biharmonic problem of the stream function ψ. This pure stream function formulation can be obtained by simply substituting the equation (3.4) into (3.3).

3.2. Time integration

We employ a second-order IMEX (implicit–explicit) backward integration scheme for Eqs. (3.3) and (3.4) as

$$\frac{3\omega^{n+1}-4\omega^{n}+\omega^{n-1}}{2\Delta t}+[2J(\omega^{n},\psi^{n})-J(\omega^{n-1},\psi^{n-1})]=\frac{1}{Re}\Delta\omega^{n+1}, \tag{3.9}$$

$$\Delta\psi^{n+1}=\omega^{n+1}, \tag{3.10}$$

$$\psi^{n+1}(1,\theta)=0,\quad \frac{\partial\psi^{n+1}}{\partial r}(1,\theta)=h(\theta). \tag{3.11}$$

The superscript on a variable represents the time step index where Δt is the time step. One can easily see that the above time integration scheme has local truncation error $O(\Delta t^2)$. Here, we treat the nonlinear convection term explicitly and the linear viscous term implicitly so that at each time step we need to solve a coupled system of Poisson problems just like Eqs. (1.4)–(1.6). Therefore, the fast biharmonic solver described in the previous section can be applied without much modifications.

Recently, the first author has introduced a finite difference scheme for Eqs. (3.3) and (3.4) which uses the Runge-Kutta method as a time integrator and treats the convection and viscous terms explicitly [8]. The method involves solving a single Poisson problem for the stream function at each time stage. However, the price to be paid for such simplicity is that the time step has to be chosen very small in order to guarantee the numerical stability. To have a reasonable time step size, a Fourier filtering must be implemented to the vorticity near the center at each time stage.

There are other numerical schemes for the unsteady Navier–Stokes equations (3.3) and (3.4) on a disk in the literature. For instance, Torres and Coutsias [12] have implemented a pseudospectral method with a third-order IMEX backward differencing for time integration to Eqs. (3.3) and (3.4). The pesudospectral method involves expanding the vorticity and the stream function in a truncated Chebyshev–Fourier series in $r-\theta$ directions. Therefore, it leads to solve the similar singular Fourier mode equations (2.3) and (2.4) by Chebyshev method. In order to keep the spectral accuracy, some complicated preconditioning techniques must be employed for different modes which makes solving the resultant linear equations quite complex. On the other hand, as we discussed before, our resultant linear equations can be solved in a simple and efficient way without any preconditioning techniques.

3.3. Spatial discretization

In the computation of the nonlinear convection term, we need to compute the first derivatives of ψ and ω. This can be easily approximated by the second-order centered difference method. Here, we use the same $M \times N$ grid as in the biharmonic problem, that is,

$$(r_i, \theta_j) = ((i - 1/2)\Delta r, j\Delta\theta), \tag{3.12}$$

where $\Delta r = 2/(2M + 1)$ and $\Delta\theta = 2\pi/N$. Again, by shifting a half mesh in radial direction, we avoid placing grid points directly at the origin.

Let the discrete values of the scalar function ψ be denoted by $\psi_{i,j} \approx \psi(r_i,\theta_j)$. Then, the first derivatives of r and θ can be approximated by

$$\left(\frac{\partial\psi}{\partial r}\right)_{ij} = \frac{\psi_{i+1,j} - \psi_{i-1,j}}{2\Delta r}, \tag{3.13}$$

$$\left(\frac{\partial\psi}{\partial\theta}\right)_{i,j} = \frac{\psi_{i,j+1} - \psi_{i,j-1}}{2\Delta\theta}. \tag{3.14}$$

Since the function is periodic in θ, the approximation of θ-derivative does not run into any trouble. However, at $i = 1$, the numerical boundary value $\psi_{0,j}$ must be provided. One appropriate choice of the value is $\psi_{0,j} = \psi_{1,j+\frac{N}{2}}$. This is because if we replace $-r$ by r and θ by $\theta + \pi$ in the Cartesian-polar transformation, the Cartesian coordinates of a point remain fixed. Therefore, any scalar function satisfies $\psi(-r,\theta) = \psi(r,\theta + \pi)$ if the domain of the function is extended to negative values of r. The same spatial discretization is applied to the vorticity ω.

3.4. Vorticity boundary conditions

After finishing one time step in our scheme (3.9) and (3.10), we obtain the vorticity ω^{n+1} and the stream function ψ^{n+1} at those interior grid points. However, the boundary vorticity is needed for the approximation of the convection term near the boundary. This can be derived using Thom's formula [16] as follows. We first approximate the second boundary condition of (3.11) at the boundary $r_{M+1} = 1$ by

$$\frac{\psi^{n+1}_{M+2,j} - \psi^{n+1}_{M,j}}{2\Delta r} = h_j, \tag{3.15}$$

where $\psi^{n+1}_{M+2,j}$ is a ghost value outside the computational domain. Therefore, we have $\psi^{n+1}_{M+2,j} = \psi^{n+1}_{M,j} + 2\Delta r h_j$. Substituting the value of $\psi^{n+1}_{M+2,j}$ and using the fact of $\psi^{n+1}_{M+1,j} = 0$ for all j, we can compute the boundary vorticity by the discrete Laplacian as

$$\omega^{n+1}_{M+1,j} = \Delta\psi^{n+1}_{M+1,j} = \frac{2\psi^{n+1}_{M,j} + 2\Delta r h_j}{(\Delta r)^2} + h_j. \tag{3.16}$$

The readers who are interested in the vorticity boundary conditions and the related issues can refer to [3].

4. Numerical results

In this section, we first perform the accuracy check for our numerical schemes to the biharmonic and Navier–Stokes equations on a disk. Then we demonstrate the numerical validness of our Navier–Stokes solver by simulating the moving wall and tripole formation problems. In particular, those problems are picked to test our correct treatments near the boundary (moving wall problem) and near the center (tripole formation).

4.1. Accuracy check for the biharmonic solver

We start our numerical tests by checking the accuracy of our FFT-based fast biharmonic solver on a disk. We simply test two exact solutions of Eq. (1.1) as

$$u_1(r,\theta) = \frac{1}{4}(1-r^2)(1+r\cos\theta), \quad f_1(r,\theta) = 0, \tag{4.1}$$

$$u_2(r,\theta) = \mathrm{e}^{r(\cos\theta+\sin\theta)}, \quad f_2(r,\theta) = 4u_2(r,\theta). \tag{4.2}$$

The first solution is chosen as the same one used in [6].

Table 1 shows the L_∞ errors for our test problems. Here, we fix the number of grid points in the azimuthal direction as $N = 64$ and vary the number of grid points M in the radial direction. It is clear that both convergent rates approach two. Therefore, our method is indeed second-order accurate for the biharmonic equation.

4.2. Accuracy check for the Navier–Stokes solver

In this example, we check the accuracy of our scheme for the Navier–Stokes equations. We have taken the exact solution for the Navier–Stokes equations as

Table 1
L_∞ errors for the biharmonic equation

M	L_∞ error (u_1)	Rate	L_∞ error (u_2)	Rate
16	2.4736E−04	–	1.2272E−03	–
32	6.3760E−05	1.96	3.1760E−04	1.92
64	1.6186E−05	1.98	8.0719E−05	1.98
128	4.0786E−06	1.99	2.0340E−05	1.99
256	1.0236E−06	2.00	5.1052E−06	1.99

$$\omega(x,y,t) = 2\mathrm{e}^{-2t/Re}\cos x\cos y, \quad \psi(x,y,t) = \mathrm{e}^{-2t/Re}\cos x\cos y. \tag{4.3}$$

The functions are described in Cartesian coordinates for simplicity of presentation. The actual computations are all in polar coordinates.

In our test, we use $M \times N$ grid points in the disk so that there are M points in the radial direction and N points in the azimuthal direction. The Reynolds number is $Re = 20$. The time step is chosen as $\Delta t = 0.01$ and the approximate solutions were computed up to $T = 2$. Table 2 shows the L_∞ errors for different number of grid points. One can easily see that the second-order accuracy has been achieved for both the stream function and the vorticity.

Table 2
L_∞ errors for the Navier–Stokes equations

$M \times N$	L_∞ error (ψ)	Rate	L_∞ error (ω)	Rate
16×32	3.2481E−04	–	1.6555E−03	–
32×64	8.3609E−05	1.96	6.0361E−04	1.46
64×128	2.1269E−05	1.97	1.7393E−04	1.80
128×256	5.3372E−06	1.99	4.1738E−05	1.88

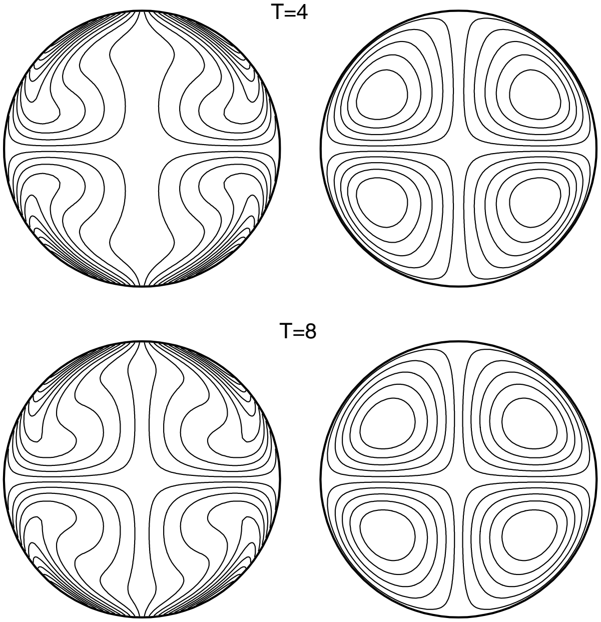

Fig. 1. Vorticity contours (left) and streamlines (right) of the moving wall problem with $Re = 100$.

4.3. Moving-wall problem

The moving-wall problem is a flow problem generated by the tangential motion of the boundary of the disk. Thus, the radial boundary velocity is always kept to be $u_r(1,\theta) = 0$. Here, we choose the same azimuthal (tangential) boundary velocity as in [4]

$$u_\theta = \cos\theta \sin\theta. \tag{4.4}$$

This problem is chosen to test the numerical treatments near the boundary for our Navier–Stokes solver since the vorticity is generated by the movement of the boundary. One can expect that the maximal vorticity occurs near the boundary.

In our run, we use a 128 × 128 grid and the time step $\Delta t = 0.01$. The computations were computed up to time $T = 8$. Figs. 1 and 2 show the vorticity contours and the streamlines at time $T = 4$ and 8 for two different Reynolds numbers $Re = 100$ and 300, respectively. One can see the flow is breaking into four quadrants which is because the azimuthal velocity (4.4) changes sign four times on the boundary. Besides, the gradients of the vorticity contours of the case $Re = 300$ are steeper than the one observed at $Re = 100$ which reflects the more diffusion of the vortex in the latter case.

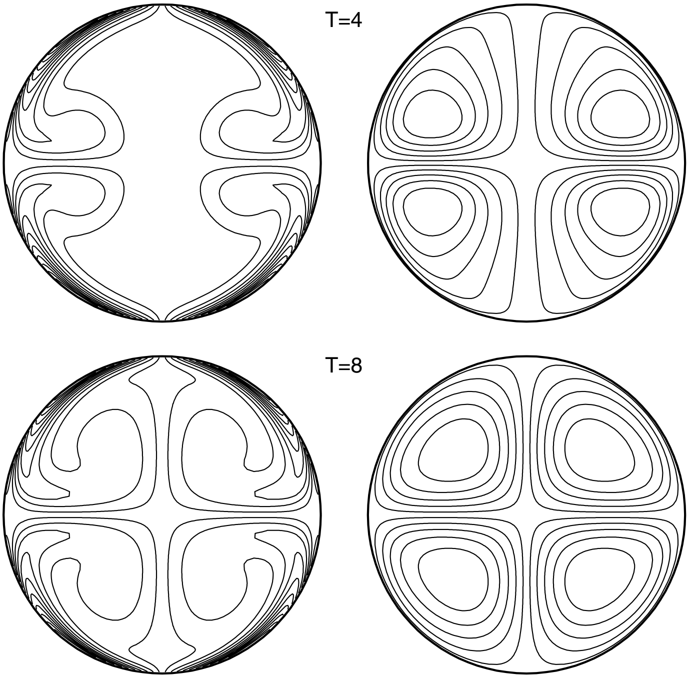

Fig. 2. Vorticity contours (left) and streamlines (right) of the moving wall problem with $Re = 300$.

4.4. Tripole formation

In order to verify the validness of our numerical treatment near the center, we perform the similar numerical simulation of the tripole formation as in [13]. It has been shown in laboratory and computer experiments [14] that a shielded monopolar vortex, when perturbed, will produce a tripolar vortex. The physics and characteristics of such tripole formation are well explained in [14]. This example serves a perfect test for the coordinate singularity treatment if we place the monopolar vortex at the center.

As in [13], the initial vorticity profile is given by

$$\omega(r) = \left[1 - \frac{\alpha}{2}\left(\frac{r}{\rho}\right)^{\alpha}\right] \mathrm{e}^{-(r/\rho)^{\alpha}}, \tag{4.5}$$

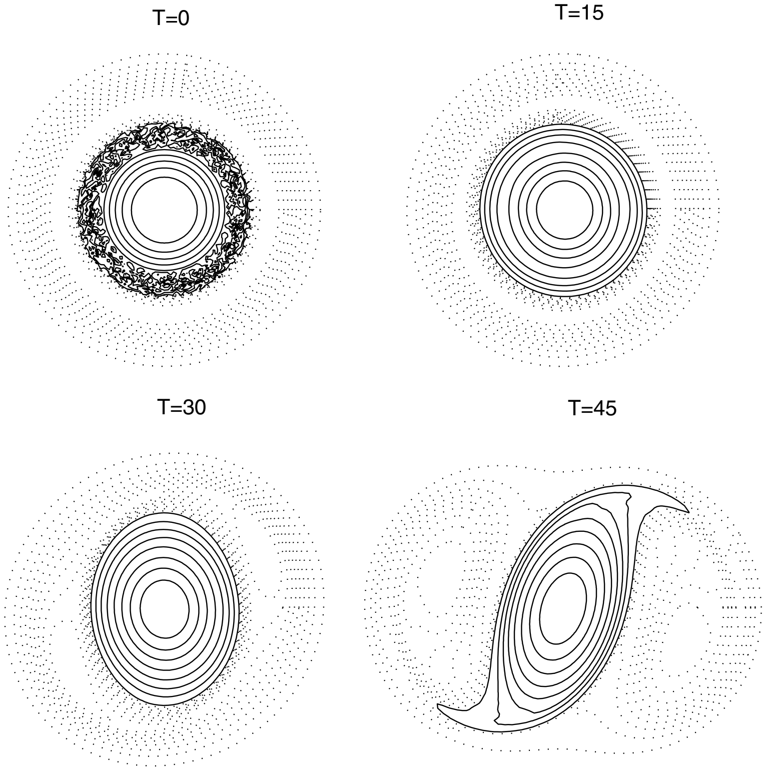

Fig. 3. Vorticity contour plots of the tripole formation from $T = 0$ to 45: '-' positive values, '.' negative values.

where the parameter α controls the steepness of the vorticity gradients and ρ controls the size of the monopole. One can easily see from (4.5), the maximal vorticity occurs at $r = 0$. In our test, the parameters are chosen as $\alpha = 3$, $\rho = 1$, and the Reynolds number $Re = 2000$. We extend the radial computational domain to $r = 4$ where the no-slip conditions are imposed on the boundary. We use a 128×128 grid and the time step $\Delta t = 0.005$.

To speed up the tripole formation, a random perturbation has been added to the vorticity distribution of (4.5) in the neighboring region where the vorticity changes sign. Fig. 3 and Fig. 4 show the formation process of the tripolar vortex from $T = 0$ to 105. As the time evolves, the initial vortex structure soon breaks the symmetry. The central (positive) vortex becomes elliptical and along the longer sides the negative vorticity organizes into two satellite vortices. This steady configuration then rotates about the center of the positive vortex core in the sense of the positive vorticity. The time evolution of the peak vorticity of

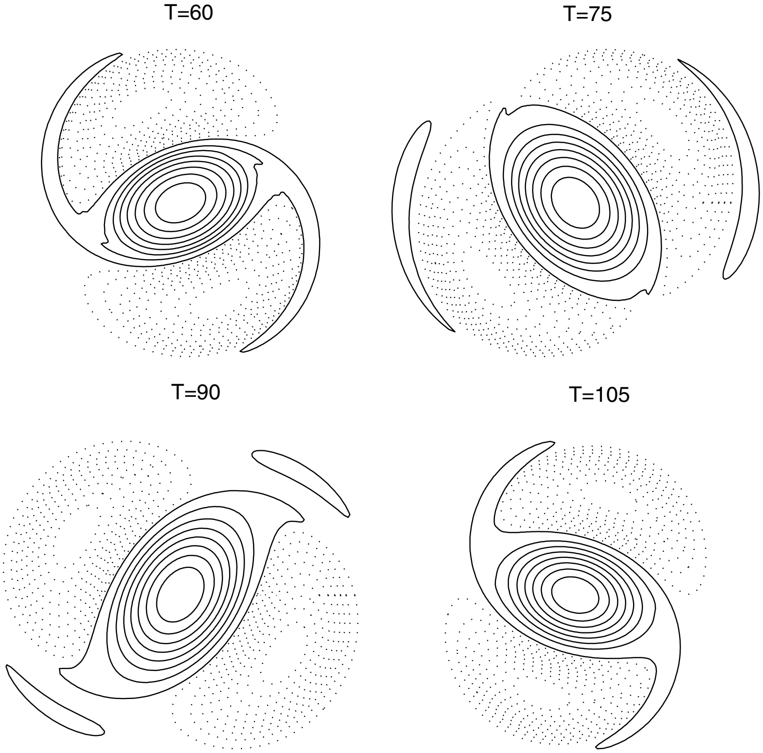

Fig. 4. Vorticity contour plots of the tripole formation from $T = 60$ to 105: '-' positive values, '.' negative values.

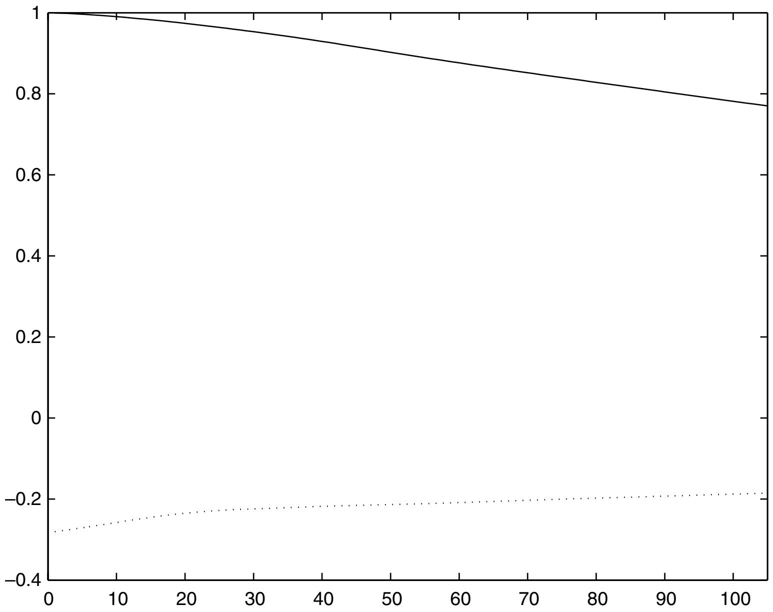

Fig. 5. Time evolution of the peak vorticity for the tripole formation: '-' peak positive vorticity, '.' peak negative vorticity.

the positive and negative vortices are shown in Fig. 5. One can see the peak vorticity decreases due to the diffusion effect.

Acknowledgement

The authors were supported in part by the National Science Council of Taiwan under research grant NSC-92-2115-M-009-012.

References

[1] S.C.R. Dennis, M. Ng, P. Nguyen, Numerical solution for the steady motion of a viscous fluid inside a circular boundary using integral conditions, J. Comput. Phys. 108 (1993) 142–152.

[2] L.W. Ehrlich, M.M. Gupta, Some difference schemes for the biharmonic equation, SIAM J. Numer. Anal. 12 (1975) 773–789.

[3] E. Weinan, J.-G. Liu, Vorticity boundary condition and related issues for finite difference schemes, J. Comput. Phys. 124 (1996) 368.

[4] L. Greengard, M.C. Kropinski, An integral equation approach to the incompressible Navier–Stokes equations in two dimensions, SIAM J. Sci. Comput. 20 (1) (1998) 318.

[5] W. Huang, T. Tang, Pseudospectral solutions for steady motion of a viscous fluid inside a circular boundary, Appl. Numer. Math. 33 (2000) 167–173.

[6] A. Karageorghis, T. Tang, A spectral domain decomposition approach for steady Navier–Stokes problems in circular geometries, Comput. Fluids 25 (1996) 541–549.

[7] M.-C. Lai, A simple compact fourth-order Poisson solver on polar geometry, J. Comput. Phys. 182 (2002) 337–345.

[8] M.-C. Lai, Fourth-order finite difference scheme for the incompressible Navier–Stokes equations in a disk, Int. J. Numer. Meth. Fluids 42 (2003) 909–922.

[9] M.-C. Lai, W.-C. Wang, Fast direct solvers for Poisson equation on polar and spherical geometries, Numer. Meth. Partial Differ. Eq. 18 (2002) 56–68.

[10] M.-C. Lai, W.-W. Lin, W. Wang, A fast spectral/difference method without pole conditions for Poisson-type equations in cylindrical and spherical geometries, IMA J. Numer. Anal. 22 (2002) 537–548.

[11] K. Mohseni, T. Colonius, Numerical treatment of polar coordinate singularities, J. Comput. Phys. 157 (2000) 787–795.

[12] D.J. Torres, E.A. Coutsias, Pseudospectral solution of the two-dimensional Navier–Stokes equations in a disk, SIAM J. Sci. Comput. 21 (1) (1999) 378–403.

[13] R. Verzicco, P. Orlandi, A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates, J. Comput. Phys. 123 (1996) 402–414.

[14] P. Orlandi, G.F. van Heijst, Numerical simulation of tripolar vortices in 2D flow, Fluid Dyn. Res. 9 (1992) 179–206.

[15] J.C. Strikwerda, Finite Difference Scheme and Partial Differential Equations, 1989.

[16] A. Thom, The flow past circular cylinders at low speeds, Proc. Roy. Soc. London A 141 (1933) 651–669.