INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **00**:1–6 *Prepared using* fldauth.cls *[Version: 2002/09/18 v1.01]*

Fourth-order finite difference scheme for the incompressible Navier-Stokes equations in a disk

Ming-Chih Lai

Department of Applied Mathematics, National Chiao Tung University
1001, Ta Hsueh Road, Hsinchu 30050, TAIWAN
mclai@math.nctu.edu.tw

SUMMARY

We develop an efficient fourth-order finite difference method for solving the incompressible Navier-Stokes equations in the vorticity-stream function formulation on a disk. We use the fourth-order Runge-Kutta method for the time integration and treat both the convection and diffusion terms explicitly. Using a uniform grid with shifting a half mesh away from the origin, we avoid placing the grid point directly at the origin; thus, no pole approximation is needed. Besides, on such grid, a fourth-order fast direct method is used to solve the Poisson equation of the stream function. By Fourier filtering the vorticity in the azimuthal direction at each time stage, we are able to increase the time step to a reasonable size. The numerical results of the accuracy test and the simulation of a vortex dipole colliding with circular wall are presented. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Navier-Stokes equations; Vorticity-stream function formulation; Polar coordinates; Fast Poisson solver; Runge-Kutta method.

1. INTRODUCTION

For the past three decades, finite difference approximation of the incompressible Navier-Stokes equations on Cartesian geometry has been developed and implemented extensively. However, for the fluid flow in polar/cylindrical or spherical domain, the relevant literature is still quite limited. The reason may be attributed to introduce some new additional difficulties when the governing equations are expressed by those orthogonal curvilinear coordinates. For instance, those difficulties might include the coupling of velocity components, the coordinate singularities, and the restrictive CFL constraint. Let us describe those difficulties briefly in the following.

The coupling of velocity components is not unusual even in the case of Cartesian geometry. This can be seen through the nonlinear convection term in the governing equations. For the case of other orthogonal curvilinear coordinates systems, the velocity components are coupled

Contract/grant sponsor: National Science Council of Taiwan; contract/grant number: NSC-91-2115-M-009-016.

Received
Revised

even in the linear diffusion terms. This can be explained by when the vector elliptic equation is written in those coordinates, the Laplacian acting on the vector field results a coupled system of elliptic equations. Fortunately, this kind of coupling can be eliminated by applying a similarity transformation to the vector field as described in [10] for the polar/cylindrical coordinates. A detailed summary for how to separate the vector elliptic equations in other different coordinates can be found in the book of Quartapelle [12].

The second difficulty arises from the coordinate singularities occurring at the origin/centerline ($r = 0$) of the polar/cylindrical domain, and the north and south poles ($\phi = 0, \pi$) of spherical domain. It is important to note that the occurrence of those singularities is due to the representation of the governing equation in those coordinates and it is irrelevant to the solution at those singular points. For simplicity, we will just name those singularities as the "poles" in different domains. In order to capture the solution behavior near the poles, some suitable treatments must be employed. For instance, in [6], the authors deduced a new governing equation which is valid at the centerline in polar/cylindrical domain. This traditional technique is to use a regular grid with imposing conditions at the poles. Some natural pole conditions can be found in the book of Canuto et. al.[3]. Note that, in finite difference setting, these pole conditions can be used to approximate numerical boundary values at the poles.

Another technique is to use a uniform grid by shifting a half mesh away from the poles to avoid placing the grid points directly at the poles; thus, no pole conditions are needed [5, 11]. This simple and elegant approach has been successfully applied to simulate the convective flow over a sphere [5] and compressible Navier-Stokes equations [11]. Using this kind of grid, Lai and Wang have been able to develop an efficient class of second- and fourth-order accurate fast direct solvers for Poisson/Helmholtz equations on 2D polar and spherical domains without pole conditions [8, 9].

The third difficulty arising is the severe CFL stability restriction. This is because when using a uniform grid to discretize the governing equations in polar/cylindrical coordinates, the azimuthal grid resolution is proportional to the distance from the poles. In other words, the physical mesh spacing of the azimuthal grid points becomes finer and finer as the poles were approached. Thus, near the poles, this leads to an over-resolution in the azimuthal direction which requires choosing a very small time step to fulfill the CFL constraint for the time-dependent problem if an explicit time-integration is used. Several techniques were proposed to overcome this difficulty. For instance, Akselvoll and Moin [1] treated all terms with derivatives in the azimuthal direction implicitly in the core region (near the pole) but all other terms are treated explicitly. While in the outer region (away from the pole), the diffusive terms and the derivatives in the radial direction are treated implicitly but all other terms are treated explicitly. The idea behind their approach is significant, however, the treatment of the interface between the two regions and the maintenance of the required accuracy could be quite complex. Another simple treatment introduced by [15, 5] is to filter out the high wavenumber modes of the dependent variable in the azimuthal direction. This treatment has the same effect to coarsen the grid near the poles thus allowing us to increase the time step to a reasonable size.

In this paper, we shall present an efficient finite difference scheme for solving the incompressible Navier-Stokes equations in a two-dimensional disk. The reason that we choose the flow problem in a disk domain is because it suffices to have all aforementioned numerical difficulties occurring in higher-dimensional cylindrical and spherical geometries. Note that here we are not trying to compare or compete with the finite element or spectral methods for solving such kind of problems. We simply want to introduce some simple finite difference

treatments for the difficulties such as the coordinate singularity and the restrictive CFL constraint. Another main feature of our method is the fourth-order accuracy which is sufficient for most of flow computations. The rest of the paper is organized as follows. In Section 2, we formulate the incompressible Navier-Stokes equations using the vorticity-stream function form in polar coordinates. The complete fourth-order numerical scheme for solving the equations is described in Section 3. A simple fourth-order fast direct solver for Poisson equation in a disk is reviewed in Section 4. The numerical results are given in Section 5 and followed by some conclusion.

2. INCOMPRESSIBLE NAVIER-STOKES EQUATIONS

The incompressible Navier-Stokes equations has the standard form

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

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

where $\mathbf{u}(\mathbf{x}, t)$ is the fluid velocity, $p(\mathbf{x}, t)$ the pressure, and ν the fluid viscosity. 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 (1)-(2) by so called the vorticity-stream function formulation. By taking the curl of Eq. (1) to eliminate the pressure gradient term, we have

$$\frac{\partial \omega}{\partial t} + J(\omega, \psi) = \nu \Delta \omega, \tag{3}$$

where ω is the nonzero z component of the vorticity, ψ the stream function, defined by $\mathbf{u} = \nabla \psi \times \mathbf{e}_z$, and $J(\omega, \psi)$ is the Jacobian determinant. Using this formulation, the velocity $\mathbf{u}$ automatically satisfies the incompressibility constraint (2). By the definition of ω, we can easily derive the relation of ω and ψ as

$$\Delta \psi = -\omega. \tag{4}$$

Therefore, the original 2D Navier-Stokes equations (1)-(2) with three primitive variables now has an alternate formulation described by (3)-(4) with only two unknown variables.

We are interested in the numerical approximations of Eqs. (3)-(4) in a unit disk $\Omega = \{0 < r \leq 1, 0 \leq \theta \leq 2\pi\}$. Thus, it is natural to rewrite the equation by the polar coordinates. The nonlinear Jacobian describing the vorticity transport now has the form

$$J(\omega, \psi) = \frac{1}{r} \left[\frac{\partial \psi}{\partial \theta} \frac{\partial \omega}{\partial r} - \frac{\partial \psi}{\partial r} \frac{\partial \omega}{\partial \theta} \right], \tag{5}$$

and the Laplacian operator Δ is

$$\Delta \equiv \frac{\partial^2}{\partial r^2} + \frac{1}{r} \frac{\partial}{\partial r} + \frac{1}{r^2} \frac{\partial^2}{\partial \theta^2}. \tag{6}$$

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

$$u_r = \frac{1}{r} \frac{\partial \psi}{\partial \theta}, \qquad u_\theta = -\frac{\partial \psi}{\partial r}. \tag{7}$$

In this paper, we restrict our attention to the flow inside a unit disk with the no-slip velocity conditions specified on the boundary; that is, $u_r = u_\theta = 0$ at $r = 1$. From the relation of (7), the boundary conditions for Eqs. (3)-(4) become

$$\psi(1,\theta) = 0, \qquad \frac{\partial \psi}{\partial r}(1,\theta) = 0. \tag{8}$$

So the governing equations consist of Eqs. (3)-(4) and the boundary conditions (8).

It is interesting to note that there are two boundary conditions for the stream function ψ but no boundary condition for the vorticity ω. This is troublesome since Eqs. (3)-(4) are also coupled by the boundary conditions. There are several different approaches summarized in [12] to overcome this difficulty. In this paper, we will just use an explicit time integration to solve Eqs. (3)-(4), and transfer one boundary condition of the stream function to the vorticity.

3. NUMERICAL METHOD

3.1. Time integration

We employ a fourth-order Runge-Kutta method for the time integration of Eqs. (3)-(4) as

$$\frac{\omega_1 - \omega^n}{\frac{1}{2}\Delta t} + J(\omega^n, \psi^n) = \nu \Delta \omega^n, \qquad \Delta \psi_1 = -\omega_1, \psi_1(1,\theta) = 0; \tag{9}$$

$$\frac{\omega_2 - \omega^n}{\frac{1}{2}\Delta t} + J(\omega_1, \psi_1) = \nu \Delta \omega_1, \qquad \Delta \psi_2 = -\omega_2, \psi_2(1,\theta) = 0; \tag{10}$$

$$\frac{\omega_3 - \omega^n}{\Delta t} + J(\omega_2, \psi_2) = \nu \Delta \omega_2, \qquad \Delta \psi_3 = -\omega_3, \psi_3(1,\theta) = 0; \tag{11}$$

$$K_4 = -J(\omega_3, \psi_3) + \nu \Delta \omega_3,$$

$$\omega^{n+1} = \frac{1}{3}(-\omega^n + \omega_1 + 2\omega_2 + \omega_3) + \frac{\Delta t}{6} K_4, \qquad \Delta \psi^{n+1} = -\omega^{n+1}, \psi^{n+1}(1,\theta) = 0, \tag{12}$$

where the superscript on a variable represents the time step index, and Δt is the time step.

At each Runge-Kutta stage, we discretize both the convection and diffusion terms in the vorticity equation explicitly such that the vorticity can be computed directly (without solving any linear system of equations). The spatial derivatives with respect to r and θ in both convection and diffusion terms are approximated by the standard fourth-order centered difference discretization, see next subsection in detail. Meanwhile, to avoid the severe CFL stability constraint near the origin, increasingly many high wavenumber modes for the vorticity in θ direction are removed as the grid point toward to the origin. The resultant vorticity is then used as a right-hand side function of Poisson equation for solving the stream function. The scheme is very efficient in terms of computational complexity since it only involves one Poisson equation to be solved at each time stage. In Section 4, we will introduce a fourth-order accurate fast Poisson solver for solving the stream function. The employment of Fourier filtering will be discussed more detail in Section 5.

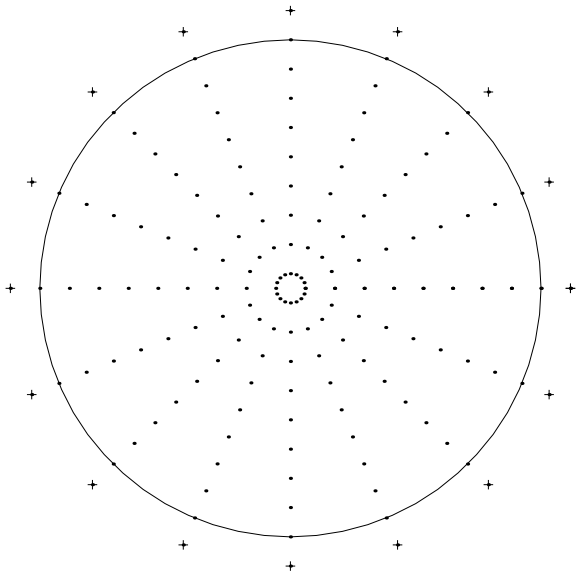

Figure 1. A polar grid in a disk based on $\overline{G}$ with $M = 8$ and $N = 16$. The circle denotes the boundary. The vorticity ω are defined on those '.' grid points. The stream function ψ are defined on '.' points as well as the outer ghost points '+'.

3.2. Spatial discretization

First, let us choose a $M \times N$ computational grid inside the disk domain Ω with the grid point locations

$$G = \{(r_i, \theta_j) = (\,(i - 1/2)\,\Delta r, (j-1)\,\Delta\theta\,) \,|\, 1 \le i \le M, 1 \le j \le N\}, \tag{13}$$

where $\Delta r = 2/(2M+1)$ and $\Delta\theta = 2\pi/N$. By the choice of Δr, we have $r_{M+1} = 1$. By shifting a half mesh away from the origin, we avoid placing the grid point directly at the origin; thus, hopefully no pole approximation is needed. Because a five-point stencil will be used for approximating the derivatives, we define a larger grid set by adding the computational grid G with a few ghost points along the radial direction as

$$\overline{G} = \{(r_i, \theta_j) = (\,(i - 1/2)\,\Delta r, (j-1)\,\Delta\theta\,) \,|\, -1 \le i \le M+2, 1 \le j \le N\}. \tag{14}$$

The unknowns are computed at the grid points of G whereas $\overline{G}$ consists of all grid points involved. This polar grid $\overline{G}$ is illustrated in Fig. 1 with $M = 8$ and $N = 16$.

The computation of the nonlinear convection and linear diffusion terms in the vorticity equation involves the spatial discretization for the first and second derivatives in radial and azimuthal directions. Those derivatives can be approximated by the standard fourth-order centered difference formulas as

$$\left.\frac{\partial \omega}{\partial r}\right|_{ij} \approx \frac{-\omega_{i+2,j} + 8\omega_{i+1,j} - 8\omega_{i-1,j} + \omega_{i-2,j}}{12\,\Delta r}, \tag{15}$$

$$\left.\frac{\partial^2 \omega}{\partial r^2}\right|_{ij} \approx \frac{-\omega_{i+2,j} + 16\omega_{i+1,j} - 30\omega_{ij} + 16\omega_{i-1,j} - \omega_{i-2,j}}{12\,\Delta r^2}. \tag{16}$$

Similarly, we can write down the same formulas for the approximations of $\partial\omega/\partial\theta$ and $\partial^2\omega/\partial\theta^2$.

Since the function defined in a disk is periodic in θ, the approximation of θ-derivative does not have any numerical boundary value problem. By this, we mean that $\omega_{ij} = \omega_{ij'}$, where $j' \equiv j \,(\mathrm{mod}\, N)$. However, the numerical boundary values for the approximation of r-derivatives are more complicated. First, the centered difference formulas (15)-(16) are applied to all interior points except the last radial interior point (r_M, θ_j) adjacent to the boundary. At the point (r_M, θ_j), we use the following one-sided difference approximations described in [4]

$$\left.\frac{\partial\omega}{\partial r}\right|_{Mj} \approx \frac{3\omega_{M+1,j} + 10\omega_{M,j} - 18\omega_{M-1,j} + 6\omega_{M-2,j} - \omega_{M-3,j}}{12\,\Delta r}, \tag{17}$$

$$\left.\frac{\partial^2\omega}{\partial r^2}\right|_{Mj} \approx \frac{11\omega_{M+1,j} - 20\omega_{Mj} + 6\omega_{M-1,j} + 4\omega_{M-2,j} - \omega_{M-3,j}}{12\,\Delta r^2}. \tag{18}$$

Second, due to the five-point stencil of (15)-(16), the numerical boundary approximations in the vicinity of the origin must also be provided. For instance, when $i = 1$ in the formula of (15), the numerical approximations of $\omega_{-1,j}$ and ω_{0j} at the corresponding ghost points $(r_{-1}, \theta_j), (r_0, \theta_j)$ need to be provided. Those approximations can be found in the following. When 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 $\omega(-r, \theta) = \omega(r, \theta + \pi)$ if the domain of the function is extended to negative values of r. So we have

$$\omega_{0j} = \omega(-\Delta r/2, \theta_j) = \omega(\Delta r/2, \theta_j + \pi) = \omega_{1,j+N/2}. \tag{19}$$

Similarly, we have $\omega_{-1,j} = \omega_{2,j+N/2}$.

3.3. Vorticity boundary condition

As we mentioned before, the no-slip conditions become two boundary conditions of the stream function as in (8). This causes that the Poisson equation of the stream function is over-determined. One easy way to overcome this difficulty is to translate one boundary condition of the stream function to the vorticity, which is first derived by Thom [13]. A very detailed discussion of the vorticity boundary condition in Cartesian coordinates and different variants of Thom's formula can be found in the work of E and Liu [4]. Next, we will derive the fourth-order vorticity boundary condition in polar coordinates.

As in the scheme (9)-(12), the Poisson equation of the stream function is solved with the Dirichlet boundary condition $\psi(1, \theta) = 0$. The other Neumann boundary condition of ψ can be used to obtain the boundary condition of the vorticity as follows. Using the one-sided difference formula (17) to approximate the Neumann condition on the boundary points (r_{M+1}, θ_j), we have

$$\left.\frac{\partial\psi}{\partial r}\right|_{M+1,j} \approx \frac{3\psi_{M+2,j} + 10\psi_{M+1,j} - 18\psi_{Mj} + 6\psi_{M-1,j} - \psi_{M-2,j}}{12\,\Delta r} = 0. \tag{20}$$

Since $\psi_{M+1,j} = 0$ for all j, we obtain

$$\psi_{M+2,j} = 6\psi_{M,j} - 2\psi_{M-1,j} + \frac{1}{3}\psi_{M-2,j}. \tag{21}$$

Using the fact that $\psi_{M+1,j} = 0$ and $\left.\frac{\partial \psi}{\partial r}\right|_{M+1,j} = 0$, the boundary vorticity becomes

$$\omega_{M+1,j} = -(\Delta\psi)_{M+1,j} = -\left.\frac{\partial^2 \psi}{\partial r^2}\right|_{M+1,j}. \tag{22}$$

Applying the one-sided difference (18) and using the approximation of (21), we obtain

$$\omega_{M+1,j} = \frac{-108\psi_{M,j} + 27\psi_{M-1,j} - 4\psi_{M-2,j}}{18\Delta r^2}. \tag{23}$$

Therefore, the vorticity at the boundary can be approximated by the neighboring values of the stream function.

It is interesting to note that the vorticity boundary condition derived here (under the assumption of (8)) is exactly the same one first derived by Briley [2] in the case of Cartesian coordinates. Although our derivation is for the no-slip boundary conditions, the extension to slip boundary conditions can be derived straightforwardly. In fact, in our numerical test in Section 5, we have implemented the slip boundary conditions for Example 1.

4. REVIEW OF FOURTH-ORDER FAST POISSON SOLVER

In the numerical scheme (9)-(12), we know that at each time stage requires solving one Dirichlet Poisson equation for the stream function. In this section, we review a simple FFT-based fast Poisson solver in a disk [9] which is applied to our numerical computations. This solver is based on truncating the Fourier series expansion, and then solving the differential equations of Fourier coefficients by finite difference discretization. Note that, this kind of approach is not a new one, since we can find it in different literature, eg. [7]. However, our method differs from the one in [7] by two folds: without pole conditions and fourth-order accuracy [9] vs. with pole conditions and second-order accuracy [7]. One can see that our method has a very simple treatment for the coordinate singularity and it can be applied to different boundary conditions as well.

The Poisson equation with Dirichlet boundary in a unit disk can be written as

$$\frac{\partial^2 u}{\partial r^2} + \frac{1}{r}\frac{\partial u}{\partial r} + \frac{1}{r^2}\frac{\partial^2 u}{\partial \theta^2} = f(r,\theta), \quad u(1,\theta) = g(\theta). \tag{24}$$

Due to the periodicity of u in θ direction, we can approximate the solution by the truncated Fourier series as

$$u(r,\theta) = \sum_{n=-N/2}^{N/2-1} u_n(r)\, e^{in\theta}, \tag{25}$$

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

$$u_n(r) = \frac{1}{N}\sum_{j=1}^{N} u(r,\theta_j)\, e^{-in\theta_j}. \tag{26}$$

Here, θ_j is a uniform grid in $[0, 2\pi]$ which is defined exactly the same way as in (14). 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 the expansion in (25) into Eq. (24), and equating the Fourier coefficients, the $u_n(r)$ satisfies the boundary value problem

$$\frac{d^2 u_n}{dr^2} + \frac{1}{r}\frac{du_n}{dr} - \frac{n^2}{r^2} u_n = f_n, \qquad u_n(1) = g_n. \tag{27}$$

Here, the complex Fourier coefficients $f_n(r)$ and g_n are defined in the same manner as Eqs. (25)-(26).

Now let us denote the discrete values $U_i \approx u_n(r_i)$ and $F_i \approx f_n(r_i)$, where r_i is chosen exactly the same way as in (14). We now discretize Eq. (27) using the fourth-order difference formulas given in (15)-(16) at the interior points $i = 1, 2, \ldots, M$ as

$$\frac{-U_{i+2} + 16U_{i+1} - 30U_i + 16U_{i-1} - U_{i-2}}{12\,\Delta r^2} + \frac{-U_{i+2} + 8U_{i+1} - 8U_{i-1} + U_{i-2}}{12\, r_i\, \Delta r} - \frac{n^2}{r_i^2} U_i = F_i. \tag{28}$$

This is a penta-diagonal linear system which can be solved by $O(M)$ arithmetic operations. In order to close the above linear system, we need to supply numerical boundary values such as U_{-1}, U_0, U_{M+1} and U_{M+2}. The value U_{M+1} is simply given by g_n. The inner numerical boundary values U_0, U_{-1} can be easily found by the symmetry constraint of Fourier coefficients [9] which are

$$U_0 = u_n(r_0) = u_n(-\Delta r/2) = (-1)^n\, u_n(\Delta r/2) = (-1)^n\, U_1, \tag{29}$$

$$U_{-1} = u_n(r_{-1}) = u_n(-3\,\Delta r/2) = (-1)^n\, u_n(3\,\Delta r/2) = (-1)^n\, U_2. \tag{30}$$

The outer numerical boundary value U_{M+2} can be obtained as follows. Since Eq. (27) also holds at the boundary $r_{M+1} = 1$, we have,

$$u_n''(r_{M+1}) + u_n'(r_{M+1}) = n^2\, u_n(r_{M+1}) + f_n(r_{M+1}). \tag{31}$$

Substituting the one-sided difference formulas (17)-(18) into the above equation and after some careful calculations, we obtain a formula for U_{M+2} in terms of $U_{M+1}, U_M, U_{M-1}, U_{M-2}$ and F_{M+1}. Thus, all the necessary numerical boundary values are derived. We conclude that this fast solver is spectral accurate in the θ direction while is fourth-order accurate in the r direction.

5. NUMERICAL RESULTS

In this section, we first demonstrate the rate of convergence (or the order of accuracy) for our fourth-order scheme (9)-(12) together with the spatial discretization and the vorticity boundary conditions outlined in Section 3. The Poisson equation of the stream function at each Runge-Kutta step is solved by the fourth-order fast direct solver described in Section 4. The second numerical test for our new scheme is to simulate a vortex dipole colliding with a circular wall [14].

Example 1 (Accuracy check): We start our numerical tests by checking the accuracy of our scheme for a computation in a unit disk $\Omega = \{0 < r \le 1, 0 \le \theta \le 2\pi\}$. We have taken the exact solution for Navier-Stokes equations as

$$\omega(x, y, t) = 2\, e^{-2\nu t} \cos x \cos y, \qquad \psi(x, y, t) = e^{-2\nu t} \cos x \cos y. \tag{32}$$

Using the polar coordinate transformation, the above exact solution can be represented by the functions of r and θ. Notice that, the above example has the non-zero vorticity at the origin ($x = y = 0$), but the velocity components are zero at this particular point. In fact, we did another numerical test which has non-zero velocity components and got the similar numerical results as in Table I so we omitted here.

In our test, we choose $N/2 \times N$ grid points in the disk so that there are $N/2$ points in the radial direction and N points in the azimuthal direction. The time steps for the cases of $N = 32, 64, 128, 256$ are $\Delta t = 0.05, 0.025, 0.0125, 0.00625$, respectively. The viscosity is chosen as $\nu = 0.001$. The approximate solution was computed at $T = 3$. Table I shows the L_∞, L_1, and L_2 errors for different number of grid points. In addition to comparing the errors of the vorticity (ω) and stream function (ψ), we also list the errors of the velocity components (u_r and u_θ). One can clearly see that the fourth-order accuracy of our scheme can be indeed achieved for all variables in different norms.

	N	L_∞ error	rate	L_1 error	rate	L_2 error	rate
	32	8.8196E-05		3.5668E-05		4.1421E-05	
ω	64	1.4197E-05	2.64	2.7410E-06	3.70	4.1679E-06	3.31
	128	6.8231E-07	4.38	1.2070E-07	4.51	1.8914E-07	4.46
	256	3.5084E-08	4.28	5.0087E-09	4.59	7.5868E-09	4.64
	32	4.8863E-07		6.1366E-07		4.0599E-07	
ψ	64	1.9435E-08	4.65	1.9195E-08	5.50	1.3896E-08	4.87
	128	7.8116E-10	4.64	9.3920E-10	4.35	6.5151E-10	4.41
	256	6.1184E-11	3.67	6.8173E-11	3.78	5.0133E-11	3.70
	32	7.4158E-04		5.9268E-04		5.0019E-04	
u_r	64	5.3446E-05	3.79	4.3944E-05	3.75	3.5112E-05	3.83
	128	3.5364E-06	3.92	2.9198E-06	3.91	2.2929E-06	3.94
	256	2.2664E-07	3.96	1.8705E-07	3.96	1.4597E-07	3.97
	32	7.6869E-06		2.7102E-06		3.1914E-06	
u_θ	64	5.0719E-07	3.92	1.5524E-07	4.13	1.6942E-07	4.24
	128	2.4950E-08	4.35	7.1842E-09	4.43	7.0078E-09	4.60
	256	1.1435E-09	4.45	3.7646E-10	4.25	3.2495E-10	4.43

Table I. Grid refinement analysis at $T = 3$.

Example 2 (Vortex dipole colliding with a circular wall): In order to demonstrate the performance of our new fourth-order scheme, we consider the problem of a vortex dipole colliding with a circular wall. This example was taken directly from [14] where the authors used it to test their pseudospectral code. Notice that, we are not trying to compare with the pseudospectral method in [14] but just simply introducing an alternative finite difference approach to the simulation of the fluid dynamical problem in a disk. In particular, our approach should be valid for moderate Reynolds number flows since a fourth-order explicit time stepping procedure is used.

The vortex dipole consists of two point vortices with equal magnitude of strength but different signs. Those two vortex centers are initially located at the two points $(0.15, 0)$ and $(-0.15, 0)$ inside a unit disk. The smoothing initial vorticity is chosen as

$$\omega(x, y, 0) = 1.5e^{-20((x-0.15)^2+y^2)} - 1.5e^{-20((x+0.15)^2+y^2)}, \tag{33}$$

whose contour lines can be found in the first plot of Fig. 2. The no-slip boundary conditions are imposed on the wall. The viscosity ν is chosen as 2×10^{-5}.

The calculation was performed on a $M \times N = 512 \times 512$ polar grid. Since the uniform grid is used, the physical mesh spacings along the grid circles in the core region (the region near the origin) are very small. This has the consequence that the CFL constraint is very restrictive for the time-dependent problem. One simple treatment is to employ the Fourier filtering [5, 15] for the vorticity in the θ direction inside the core region at each Runge-Kutta time stage. This treatment has the same effect to coarsen the physical mesh near the origin thus allowing us to increase the time step to a reasonable size. The idea is to remove the high wavenumber modes of the vorticity gradually as the grid points toward to the origin. For simplicity, we choose the core region as the the half disk $r \leq 0.5$. At the inner grid circle $r = \Delta r/2$, we keep the modes with wavenumber $n = 0, 1, -1$ and remove the Fourier modes whose wavenumbers are greater than one. At the second grid circle $r = 3\Delta r/2$, we keep two more modes with wavenumber $n = 0, 1, 2, -1, -2$ and remove the rest of high modes. Continuing in such way, the Fourier modes are no longer to be removed outside the core region. By applying this Fourier filtering, we are able to choose a much larger time step to ensure the numerical stability. In our present computation, we choose the time step $\Delta t = 2.5 \times 10^{-3}$ which is about ten times larger than the one used in [14].

Fig. 2 shows the evolutionary contour plots for the vorticity. The solid line denotes the positive vorticity and the dash line denotes the negative values. The magnitude of those plots ranges from -2 to 2. One can see the vortex dipole collides with the circular wall then splits. Each vortex generates another different sign of vortex and forms another dipole and moves to the center. Then they exchange the partners to form two symmetric vortex dipoles which the smaller one moves upward and the larger one moves downward to repeat the colliding process. Fig. 3 shows an enlarging part of the vorticity plot at $T = 120$. One can see that there are four vortex dipoles totally after two complete collisions.

The energy (E) and enstrophy (ε) are the two interesting quantities measured in this test. They are defined by

$$E = \frac{1}{2} \int_0^1 \int_0^{2\pi} (u_r^2 + u_\theta^2)\, r\, dr\, d\theta, \qquad \varepsilon = \int_0^1 \int_0^{2\pi} \omega^2\, r\, dr\, d\theta. \tag{34}$$

We compute those quantities via a discrete approximation of the above integral based on the nodes described in (13). Thus, the computation is almost the same as the usage of the midpoint rule for the double integral integration. Fig. 4-(a) and Fig. 4-(b) are the time evolutionary plots for the energy and enstrophy, respectively. One can see that the energy is decreasing. The enstrophy is oscillating with the maximum occurring around at $T = 30$ when the dipole completes the first hitting to the wall and generates another vortices.

As in [14], we also calculate the energy decay rate by the formula

$$\frac{dE}{dt} = -\nu\varepsilon. \tag{35}$$

The relative error of the decay rate is computed by

$$rel(t) = \frac{(E(t + \Delta t) - E(t - \Delta t))/2\,\Delta t + \nu\,\varepsilon(t)}{-\nu\,\varepsilon(t)}. \tag{36}$$

Fig. 4-(c) shows the relative error of the energy decay rate which is within 1.6E-03 after some earlier time. The computed error at $T = 120$ is 4.3E-05 which is about the magnitude of

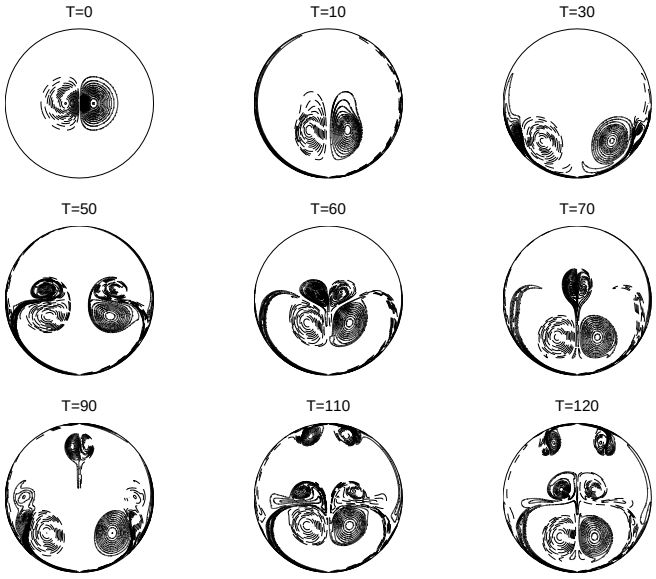

Figure 2. Evolutionary vorticity contours for a vortex dipole colliding with a circular wall. The positive vorticity denotes by '-' and the negative vorticity by '–'. The magnitude ranges from -2 to 2.

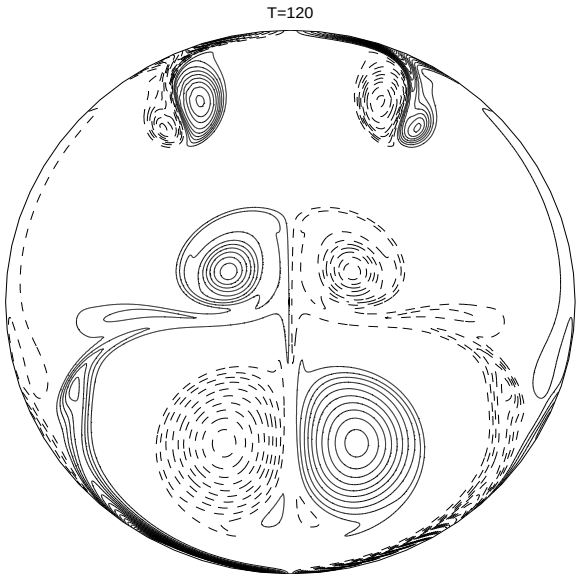

Figure 3. Four vortex dipoles were seen after two complete dipole collisions at $T = 120$.

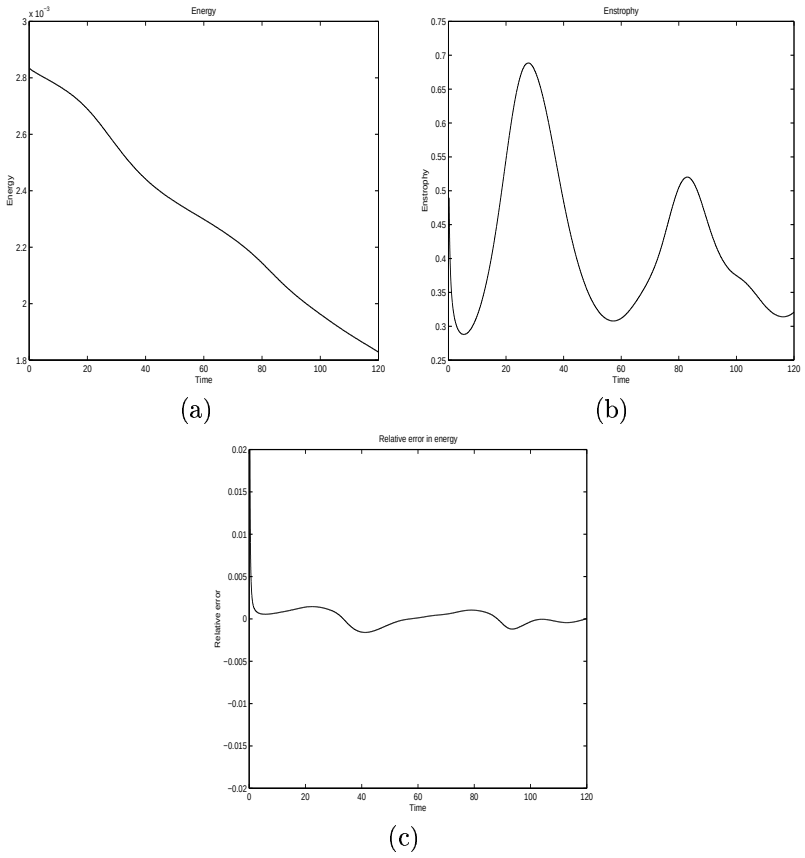

Figure 4. The evolutionary plot of energy, enstrophy, and the relative error of energy decay rate.

$O(\Delta t^2)$. So the numerical evidence shows that our scheme does provide the right energy decay rate.

6. CONCLUSION

Let us summarize our scheme as follows. The Navier-Stokes equations are written in the vorticity-stream function form. We use the fourth-order Runge-Kutta method for the time integration and treat both the convection and diffusion terms explicitly. Using a uniform grid with shifting a half mesh away from the origin, we avoid placing the grid point directly at the origin; thus, no pole approximation is needed. Besides, on such grid, a fourth-order fast direct method is used to solve the Poisson equation of the stream function. By Fourier filtering the vorticity in the azimuthal direction at each time stage, we are able to increase the time step to a reasonable size. The numerical results of the accuracy test and the simulation of a vortex dipole colliding with circular wall are presented.

REFERENCES

1. Akselvoll K, Moin P. An efficient method for temporal integration of the Navier-Stokes equations in confined axisymmetric geometries. *J. Comput. Phys.* 1996; **125**: 454–463.
2. Briley WR, *J. Fluid Mech.* 1971; **47**: 713–736.
3. Canuto C, Hussaini MY, Quarteroni A, Zang TA. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1988; pp.90–91.
4. E W, Liu JG. Vorticity boundary condition and related issues for finite difference schemes. *J. Comput. Phys.* 1996; **124**: 368–382.
5. Fornberg B, Merrill D. Comparison of finite difference and pseudospectral methods for convective flow over a sphere. *Geophys. Res. lett.* 1997; **24**: 3245–3248.
6. Griffin MD, Jones E, Anderson JD. A computational fluid dynamic technique valid at the centerline for non-axisymmetric problems in cylindrical coordinates. *J. Comput. Phys.* 1979; **30**: 352–360.
7. Iserles A. *Numerical Analysis of Differential Equations*, Cambridge University Press, 1996; pp.256–260.
8. Lai MC. A note on finite difference discretizations for Poisson equation on a disk. *Numerical Methods for Partial Differential Equations.* 2001; **17**: 199–203.
9. Lai MC, Wang WC. Fast direct solvers for Poisson equation on polar and spherical geometries, *Numerical Methods for Partial Differential Equations.* 2002; **18**: 56–68.
10. Manna M, Vacca A. An efficient method for the solution of the incompressible Navier-Stokes equations in cylindrical geometries. *J. Comput. Phys.* 1999; **151**: 563–584.
11. Mohseni K, Colonius T. Numerical treatment of polar coordinate singularities. *J. Comput. Phys.* 2000; **157**: 787–795.
12. Quartapelle L. *Numerical Solution of the Incompressible Navier-Stokes Equations*, Birkhauser, 1993.
13. Thom A. *Proc. Roy. Soc. London Sect. A*, 1933; **141**: 651–666.
14. Torres DJ, Coutsias EA. Pseudospectral solution of the two-dimensional Navier-Stokes equations in a disk. *SIAM J. Sci. Comput.* 1999; **21**: 378–403.
15. Williamson DL. Linear stability of finite-difference approximations on a uniform latitude-longitude grid with Fourier filtering, *Mon. Wea. Rev.* 1976; **104**: 31–41.