A fast finite difference method for solving Navier-Stokes Equations on irregular domains

Zhilin Li* Cheng Wang†

April 17, 2002

keywords: Navier-Stokes equations, irregular domains, vorticity stream-function formulation, vorticity boundary condition, immersed interface method

AMS Classification; 65M06, 65M12, 76T05

Abstract

A fast finite difference method is proposed to solve the incompressible Navier-Stokes equations defined on a general domain. The method is based on the vorticity stream-function formulation and the fast Poisson solver defined on general domains using the immersed interface method. The key to the new method is the fast Poisson solver for general domains and the interpolation scheme for the boundary condition of the streaming functions. Numerical examples that show second order accuracy of the computed solution are also provided.

1 Introduction

In this paper, we consider the non-dimensionalized incompressible Navier-Stokes equations (NSE) in a general and bounded domain Ω:

$$\left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\nabla)\mathbf{u}\right) + \nabla p = \nu\Delta\mathbf{u}, \quad \mathbf{x}\in\Omega, \tag{1.1}$$

$$\nabla\cdot\mathbf{u} = 0, \tag{1.2}$$

$$\mathbf{u}|_{\partial\Omega} = \mathbf{0}, \quad \text{BC}, \tag{1.3}$$

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0, \quad \text{IC}. \tag{1.4}$$

*Center for Research in Scientific Computation & Department of Mathematics, North Carolina State University, Raleigh, NC 27695-8205, zhilin@math.nscu.edu.

†Department of Mathematics, University of Indiana at Bloomington, Bloomington, IN 47405, cwang@indiana.edu.

where $\mathbf{u} = (u, v)$ is the velocity, p is the pressure, ν is the viscosity. We assume that the boundary of the domain $\partial\Omega$ is piecewise smooth. We wish to solve the NSE numerically use a Cartesian grid that encloses the domain Ω.

Traditionally, finite element methods with a body fitted grid are used to solve such problems defined on irregular geometries. The use of Cartesian grids for solving problems with complex geometry, moving interface and free boundary problems has become quite popular recently, especially after popular Cartesian grid methods such as Peskin's immersed boundary (IB) method, see [21] for an overview, the level set method, see the original paper [20], the Clawpack [16] and other methods are developed. One of advantages of Cartesian grid methods is that there is almost no cost in the grid generation. This is quite significant for moving interface and free boundary problems.

Depending on the magnitude of the Reynolds number $1/\nu$, numerical methods can be divided as two categories for two dimensional problems: (1) numerical methods based on the primitive variables formulation for problems with small to medium sized Reynolds number, particularly the projection method, see [3, 5, 22] for examples. Usually an implicit or semi-implicit discretization in time is needed to take a reasonable time step; (2) the vorticity stream-function formulation for problems with large Reynolds numbers, see for example, [10, 13] and the references therein.

There are a few articles in the literature that use projection typed methods based on Cartesian grids for interface problems or for problems with irregular geometries using an embedding technique. Among them are, just name a few, Peskin's immersed boundary method with numerous applications [7, 8, 11, 21, 23]; the ghost fluid method [14]; the immersed interface method [15]; the finite volume method [2, 22, 24, 26] etc.

However, there are almost no known methods in the literature that solve the full incompressible Navier Stokes equations defined on complex domains using the *vorticity stream-function formulation* based Cartesian grids except the very recent work by Calhoun [6]. In [6], a finite volume method coupled with the software package Clawpack [16] is used to solve the Navier Stokes equations on irregular domains.

In this paper, we use a more direct finite difference approach based on a Cartesian grid to solve the incompressible Navier Stokes equations defined on an irregular domain. We believe that our method is simpler than the one developed in [6]. The computed vorticity and the velocity from our method are nearly second order accurate, see Sec. 5.

The paper is organized as follows. In Sec. 2, we introduce the computational frame of the vorticity stream-function formulation and an outline of our algorithm. The fast Poisson solver for complex domains is explained in Sec. 3. In Sec. 4, we explain how to deal with the vorticity boundary condition. Numerical examples and conclusion are presented in Sec. 5 and Sec. 6 respectively.

2 Computational frame of vorticity stream-function formulation for NSE on irregular domains

We use a rectangular box $R = [a,\ b] \times [c,\ d]$ to enclose the physical domain Ω. For simplicity of presentation, we use a uniform Cartesian grid

$$x_i = a + ih, \quad i = 0, 1, \cdots, M, \qquad y_j = c + jh, \quad j = 0, 1, \cdots, N. \tag{2.5}$$

where $h = (b-a)/M = (d-c)/N$. Usually we choose R in such a way that the distance between the boundaries $\partial\Omega$ and ∂R is at least $2h$ apart.

The boundary of Ω is expressed as the zero level set of a two dimensional function $\varphi(x, y)$

$$\varphi(x,y) \begin{cases} < 0, & \text{if } (x,y) \text{ is in the inside of } \Omega, \\ = 0, & \text{if } (x,y) \text{ is on the boundary of } \Omega, \\ > 0, & \text{if } (x,y) \text{ is in the outside of } \Omega. \end{cases} \tag{2.6}$$

Since $\partial\Omega$ is piecewise smooth, the level set function φ should be chosen to be at least Lipschitz continuous in the neighborhood of $\partial\Omega$. Usually φ is chosen as the signed distance function. Note that, the level set function can be easily defined at grid points.

Let

$$\begin{aligned} \varphi_{i,j}^{max} &= \max\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\}, \\ \varphi_{i,j}^{min} &= \min\{\varphi_{i-1,j}, \varphi_{i,j}, \varphi_{i+1,j}, \varphi_{i,j-1}, \varphi_{i,j+1}\}. \end{aligned} \tag{2.7}$$

We define (x_i, y_j) as an ***irregular grid point*** if $\varphi_{i,j}^{max}\ \varphi_{i,j}^{min} \le 0$. Otherwise the grid point is a *regular grid point.*

The two dimensional Navier Stokes equations in vorticity stream-function formulation is

$$\partial_t\omega + u\omega_x + v\omega_y = \nu\Delta\omega, \tag{2.8}$$

$$\Delta\psi = \omega, \quad \psi|_\Omega = 0, \tag{2.9}$$

$$u = -\psi_y, \qquad v = \psi_x, \tag{2.10}$$

where $\omega = \nabla \times \mathbf{u} = -u_y + v_x$ is the vorticity, ψ is the stream function which satisfy the non-slip boundary condition

$$\psi_n = \frac{\partial\psi}{\partial\mathbf{n}} = 0, \tag{2.11}$$

where $\mathbf{n}$ is the unit normal vector of the boundary $\partial\Omega$ pointing outward.

We define the standard central finite difference operators applied to the grid function u_{ij} below

$$D_x u_{i,j} = \frac{u_{i+1,j} - u_{i-1,j}}{2h}, \qquad D_y u_{i,j} = \frac{u_{i,j+1} - u_{i,j-1}}{2h}, \tag{2.12}$$

$$\Delta_h u_{i,j} = \left(D_x^2 + D_y^2\right) u_{i,j} = \frac{u_{i+1,j} + u_{i-1,j} + u_{i,j+1} + u_{i,j-1} - 4u_{i,j}}{h^2}. \tag{2.13}$$

With these notations, the semi-discretized Navier Stokes equations in vorticity stream-function formulation is

$$\partial_t \omega + u D_x \omega + v D_y \omega = \nu \Delta_h \omega. \tag{2.14}$$

As discussed in [9, 10], the above scheme can be implemented very efficiently via explicit treatment of the convection and diffusion terms. Such explicit treatment does not result in any problem caused by the cell-Reynolds number constraint if a Runge-Kutta method is applied.

2.1 Outline of the vorticity stream-function formulation for NSE defined on irregular domains

We use the classical fourth order Runge-Kutta method, which is a multi-stage explicit time stepping procedure, to treat the semi-discretized equations (2.14). The explicit treatment of convection and diffusion terms appearing in the momentum equations makes the whole scheme very easy to implement. Such explicit treatment can avoid any stability concern caused by the cell-Reynolds number constraint if the high order Runge-Kutta method, such as classical RK4, is applied. This observation was first made by E and Liu in [10]. As a result, only one Poisson solver, which will be explained in detail in the next section, is required at each Runge-Kutta time stage. The rest is standard finite-difference calculation. That makes the method extremely efficient. We refer the readers to [25] for the discussion of the method of the vorticity stream-function formulation for rectangular domains.

We present the following outline for the explicit Euler method from time level t^k to t^{k+1} to demonstrate the essence of our algorithm. The extension to Runge-Kutta method is straightforward.

1. Solve the Poisson equation

$$\Delta \psi^{k+1} = \omega^k, \qquad \psi^{k+1}\Big|_{\partial\Omega} = 0,$$

   to get the stream-function ψ^{k+1}.

2. Update the velocity using

$$u^{k+1} = -D_y \psi^{k+1}, \qquad v^{k+1} = D_x \psi^{k+1}.$$

3. Update the vorticity from

$$\frac{\omega^{k+1} - \omega^k}{\Delta t} = \nu \Delta_h \omega^k - u^{k+1} D_x \omega^k - v^{k+1} D_y \omega^k, \qquad \omega^{k+1}\Big|_{\partial\Omega} = \Delta_h \psi^{k+1}.$$

In our implementation, we actually use a fourth order Runge-Kutta method to update the velocity and the vorticity from t^k to t^{k+1} so that second order accuracy can be obtained.

There are two crucial components in our algorithm described above. The first one is how to solve the Poisson equation on an irregular domain which will be explained in the next section. The second one is the treatment of the boundary condition since there are two boundary conditions for ψ. The Dirichlet boundary condition $\psi = 0$ on $\partial\Omega$ was implemented to solve the stream function via the vorticity computed from (2.14). Yet the normal boundary condition, $\frac{\partial\psi}{\partial\mathbf{n}} = 0$ in (2.11), can not be enforced directly. The way to overcome this difficulty is to convert it into the boundary condition for the vorticity. The detailed process of vorticity boundary condition will be given in Sec. 4.

3 The fast Poisson solver on irregular domains

Using the vorticity stream-function formulation to solve the Navier Stokes equations, we need to solve the Poisson equation $\Delta\psi = \omega$ on the irregular domain Ω. In this section, we outline our fast Poisson solver for irregular domains. For theoretical and implementation details, we refer the readers to the references [17, 12, 19]. The two-dimensional fast Poisson solver is also available to the public [18].

Our Poisson solver on irregular domains is based on the fast immersed interface method (IIM) for interface problems. The main idea is to extend the Poisson equation from Ω to the entire rectangular domain R. This procedure allows use of fast Poisson solvers such as FFT on a fixed Cartesian grid independent of the shape of the irregular domain.

We extend the source term of the Poisson equation by zero outside Ω but inside R. We require the normal derivative of the solution ψ to be continuous across the immersed boundary $\partial\Omega$. The solution itself is allowed to have a finite jump g. In the language of potential theory this requirement is equivalent to introducing a double-layer source on $\partial\Omega$. This extension leads to the following interface problem,

$$\begin{aligned}
&\Delta\psi = \begin{cases} \omega(x,y), & \text{if } (x,y) \in \Omega, \\ 0, & \text{if } (x,y) \in R - \Omega, \end{cases} \\
&\left[\frac{\partial\psi}{\partial\mathbf{n}}\right] = 0, \qquad [\psi] = g, \qquad \text{on } \partial\Omega, \\
&\psi = 0, \qquad \text{on } \partial R,
\end{aligned} \tag{3.15}$$

where $[\cdot]$ denotes the jump across $\partial\Omega$. We need to determine the particular g, which is defined along the boundary $\partial\Omega$, so that the solution ψ of (3.15) satisfies Dirichlet boundary condition

$$\psi^- = 0, \qquad \text{on } \partial\Omega, \tag{3.16}$$

where ψ^- is the limiting value of the solution on the boundary from within the domain Ω. Note that the solution of the interface problem above is a functional of g. There is a unique solution $\psi(g)$ which is in piecewise H^2 space if ω and g are in L^2 space, and the interface $\partial\Omega$

is Lipschitz continuous. We refer the readers to [4] for the discussion of the regularity of the interface problem.

To numerically compute the solution of (3.15)–(3.16) for ψ and g, we use the standard central finite difference

$$\Delta_h \psi_{ij} = \omega_{ij} \tag{3.17}$$

at regular grid points, inside and outside of R. At irregular grid points, the finite difference scheme is

$$\Delta_h \psi_{ij} = \omega_{ij} + C_{ij}, \tag{3.18}$$

where C_{ij} is a correction term that depends on the jump g. Therefore, the solution to (3.15) satisfies a linear system of equations of the form

$$A\Psi + BG = F_1, \tag{3.19}$$

where A is the matrix obtained from the discrete Laplacian, Ψ is the approximate solution ψ at grid points, and G is the discrete value of g defined on a set of points on the boundary $\partial\Omega$, BG is the correction terms at irregular grid points, and F_1 is the source term inside Ω and zero outside Ω. The dimension of G is much smaller than the dimension of Ψ.

We use the least squares interpolation scheme [17] to discretize the boundary condition (3.16) for a given Ψ. At a given point $\mathbf{x}^* = (x^*, y^*) \in \partial\Omega$ where G is defined, the interpolation scheme has the form

$$\sum_{|\mathbf{x}_{ij} - \mathbf{x}^*| \le \epsilon} \gamma_{ij} \psi_{ij} = 0, \tag{3.20}$$

where $\mathbf{x}_{i,j} = (x_i, y_j)$, ϵ is taken between $2h \sim 3h$ meaning that we use the information of the grid function ψ_{ij} in the neighborhood of $\mathbf{x}^*$. We refer the readers to [17] about how to find the coefficients γ_{ij} in the interpolation scheme. The interpolation scheme in the matrix-vector form can be written as

$$C\Psi + DG = F_2. \tag{3.21}$$

Thus we obtain the following system of equations for the solution Ψ and the intermediate discrete jump G on the boundary

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Psi \\ G \end{bmatrix} = \begin{bmatrix} F_1 \\ F_2 \end{bmatrix}. \tag{3.22}$$

The Schur complement of (3.22) for G is

$$(D - CA^{-1}B)G = Q, \tag{3.23}$$

where

$$Q = F_2 - CA^{-1}F_1.$$

Equation (3.23) for G is a much smaller system than equation (3.22) for Ψ. We use the generalized minimum residue (GMRES) method to solve the Schur complement system. Each iteration of the GMRES method involves one matrix-vector multiplication by $(D - CA^{-1}B)G$ with a specified G. This involves a fast Poisson solver for computing $A^{-1}BG$ on the rectangular domain R, and the interpolation scheme to compute the residual of (3.23). The dominant cost in each iteration is the fast Poisson solver from the Fishpack [1].

This Poisson solver that we outlined above for irregular domains is second order accurate. The number of calls of the fast Poisson solver on the rectangular domain is the same as the number of GMRES iterations, and it is almost independent of the mesh size but depends only on the geometry of the domain.

In our implementation, the matrices and vectors are never explicitly formed. The fast solvers with examples for Poisson/Helmholtz equations on irregular domains are available to the public through anonymous ftp at *ftp ftp.ncsu.edu* under the directory */pub/math/zhilin/Packages*.

4 The vorticity boundary condition

To solve the momentum equation (2.14), we need a boundary condition for vorticity. Physically, the vorticity boundary condition enforces the no-slip boundary condition for the velocity. The no-penetration boundary condition, which can be converted into the Dirichlet boundary condition for ψ: $\psi|_{\partial\Omega} = 0$, is used in the Poisson equation (2.9) for ψ. Therefore we only need the information of the vorticity at interior grid points to solve the stream-function ψ with the homogeneous boundary condition. On the other hand, the no-slip boundary condition $\frac{\partial\psi}{\partial \boldsymbol{n}}|_{\partial\Omega} = 0$, along with the one-sided approximation of $\omega = \Delta\psi$, is converted into a local vorticity boundary condition.

4.1 Review of local vorticity boundary formula in a rectangular domain

Let us assume that Ω is a rectangle for the moment. The Dirichlet boundary condition $\psi = 0$ on $\partial\Omega$ is implemented to solve the stream-function via the vorticity obtained from (2.14). Yet the normal boundary condition, $\frac{\partial\psi}{\partial \boldsymbol{n}} = 0$, can not be enforced directly. The way to overcome this difficulty is to convert it into the boundary condition for the vorticity. We use $\psi|_{\partial\Omega} = 0$ and $\frac{\partial\psi}{\partial \boldsymbol{n}} = 0$ to approximate the vorticity on the boundary. Take the bottom part of $\partial\Omega$, where the subscript j is zero, as example, we use the central finite difference scheme to approximate the Laplacian which is simply $D_y^2\psi$ since $D_x^2\psi = 0$ from the boundary condition $\psi = 0$. The second order finite difference approximation yields

$$\omega_{i,0} = D_y^2\psi_{i,0} = \frac{\psi_{i,1} + \psi_{i,-1} - 2\psi_{i,0}}{h^2} = \frac{2\psi_{i,1}}{h^2} - \frac{2}{h}\frac{\psi_{i,1} - \psi_{i,-1}}{2h}, \tag{4.24}$$

where $\psi_{i,0} = 0$ and $(i, -1)$ refers to the "ghost" grid point outside of the computational domain. Since

$$\frac{\psi_{i,1} - \psi_{i,-1}}{2h} \approx \frac{\partial \psi}{\partial \mathbf{n}} + O(h^2) = O(h^2),$$

we have $\psi_{i,-1} = \psi_{i,1} + O(h^2)$ which leads to **Thom's formula**

$$\omega_{i,0} = \frac{2\psi_{i,1}}{h^2} \,. \tag{4.25}$$

We should mention here that by formal Taylor expansion, one can prove that the Thom's formula is only first order approximation to the boundary condition for ω. However, more sophisticated consistency analysis show that the scheme actually is indeed second order accurate. The result was first proved in [13].

The vorticity on the boundary can also be determined by other approximations to $\psi_{i,-1}$. For example, using a third order one-sided finite difference scheme to approximate the normal boundary condition $\frac{\partial \psi}{\partial \boldsymbol{n}} = 0$, we can write

$$\begin{aligned} (\partial_y \psi)_{i,0} &= \frac{-\psi_{i,-1} + 3\psi_{i,1} - \frac{1}{2}\psi_{i,2}}{3h} = 0 + O(h^2)\,, \quad \text{which leads to} \\ \psi_{i,-1} &= 3\psi_{i,1} - \tfrac{1}{2}\psi_{i,2} + O(h^3). \end{aligned} \tag{4.26}$$

Plugging this back to the difference vorticity formula $\omega_{i,0} = \dfrac{1}{h^2}(\psi_{i,1} + \psi_{i,-1})$ in (4.24), we have **Wilkes-Pearson's formula**

$$\omega_{i,0} = \frac{1}{h^2}(4\psi_{i,1} - \frac{1}{2}\psi_{i,2})\,. \tag{4.27}$$

This formula is second order accurate for the vorticity on the boundary.

From the descriptions above, we see that the vorticity boundary condition can be be derived from the combination of the no-slip boundary condition $\frac{\partial \psi}{\partial \boldsymbol{n}}|_{\partial\Omega} = 0$ and some one-sided approximations to $\omega = \Delta\psi$.

4.2 The extension to a curved domain

The extension of the above methodology for a domain with a curved boundary is similar but a little more complicated.

We use Fig. 1 as an illustration. Fig. 1 shows several grid points near the boundary $\partial\Omega$. In Fig. 1, D, E, F, G are regular grid points, AB is an arc section on the boundary. Special attention is needed at the grid points close to the boundary, such as the point C. We denote $a = |AC|/h$, $b = |BC|/h$. Note that $0 < a, b < 1$.

We explain how to determine the vorticity value on the boundary points like A and B from the combined information of the no-slip boundary condition $\frac{\partial \psi}{\partial \boldsymbol{n}}\, |_{\partial\Omega}= 0$ and the one-sided

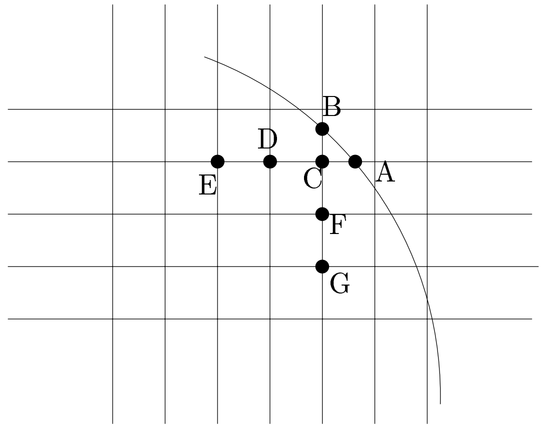

Figure 1: A diagram of the geometry near the boundary

approximation of $\omega = \Delta\psi$. We will derive a simple local formula, analogous to (4.25) or (4.27) to obtain a vorticity boundary value at A, which is an intersection of the grid line and the boundary $\partial\Omega$. The discussion for point B is similar. We need the vorticity value at those points when we solve the vorticity via (2.14) and the stream function ψ in (2.9).

The combination of the Dirichlet boundary condition $\psi\mid_{\partial\Omega}= 0$ and the Neumann boundary condition $\frac{\partial\psi}{\partial\boldsymbol{n}} = 0$ implies that

$$\nabla\psi = 0\,, \quad \text{at} \quad A\,. \tag{4.28}$$

In other words, the partial derivative of the stream function along any direction is zero on the boundary. This can also be seen by the fact that both $u = -\partial_y\psi$ and $v = \partial_x\psi$ vanishes on the boundary.

The local Taylor expansion at the boundary point A gives

$$\psi_C = \frac{a^2h^2}{2}(\partial_x^2\psi_A) - \frac{a^3h^3}{6}(\partial_x^3\psi_A) + O(h^4)\,, \tag{4.29}$$

$$\psi_D = \frac{(1+a)^2h^2}{2}(\partial_x^2\psi_A) - \frac{(1+a)^3h^3}{6}(\partial_x^3\psi_A) + O(h^4)\,, \tag{4.30}$$

where the information of $\psi = 0$ and $\partial_x\psi = 0$ at point A was used in the derivation for (4.29) and (4.30).

The Taylor expansion (4.29) gives a first order approximation to $\partial_x^2\psi$ at the boundary point A

$$(\partial_x^2\psi)_A = \frac{2}{a^2h^2}\psi_C + O(h)\,, \tag{4.31}$$

which corresponds to Thom's formula (4.25). Or the combination of (4.29) and (4.30) gives and second order approximation to $\partial_x^2\psi$ at point A

$$(\partial_x^2\psi)_A = \frac{2}{a^2h^2}\Big((1+a)\psi_C - \frac{a^3}{(1+a)^2}\psi_D\Big) + O(h^2)\,, \tag{4.32}$$

which corresponds to Wilkes' formula (4.27).

Remark 4.1. In the case of $a = 1$, i.e., the boundary point A happens to be on regular grid, the formula (4.31) becomes

$$(\partial_x^2 \psi)_A = \frac{2}{h^2} \psi_C \,, \tag{4.33}$$

and (4.32) becomes

$$(\partial_x^2 \psi)_A = \frac{4\psi_C - \frac{1}{2}\psi_D}{h^2} \,, \tag{4.34}$$

which are exactly the same as Thom's formula and Wilkes' formula in the case of rectangular domain, as in (4.25), (4.27), respectively.

However, in a domain with a curved boundary, the value of $\partial_x^2 \psi$ on the boundary point like A is not enough to determine $\omega = \Delta\psi = (\partial_x^2 + \partial_y^2)\psi$, due to the fact that $\partial_x^2 \psi$ can not be determined in the same way as we did for ψ_A and ψ_B. To deal with this difficulty, we need to use information of the stream function around the boundary point A and the relation between $\Delta\psi$ and $\partial_x^2 \psi$.

Let θ be the angle between the tangential direction of the boundary $\partial\Omega$ passing through A and the horizontal line, i.e., $\tan\theta$ is the slope of the tangent line for the boundary curve at A. Some simple manipulations indicate

$$\partial_x \psi = \cos\theta \, \partial_\tau \psi + \sin\theta \, \partial_{\boldsymbol{n}} \psi \,, \quad \partial_y \psi = -\sin\theta \, \partial_\tau \psi + \cos\theta \, \partial_{\boldsymbol{n}} \psi \,, \tag{4.35}$$

where τ is the unit vector along the tangential direction. It is easy to show that

$$\partial_x^2 + \partial_y^2 = \partial_\tau^2 + \partial_{\boldsymbol{n}}^2 \,, \tag{4.36}$$

from (4.35). In other words, the Laplacian operator is invariant with respect to orthogonal coordinate systems.

We also can verify that

$$\partial_x^2 \psi = \cos^2\theta \, \partial_\tau^2 \psi + \sin^2\theta \, \partial_{\boldsymbol{n}}^2 \psi + 2\cos\theta \sin\theta \, (\partial_\tau \partial_{\boldsymbol{n}} \psi) \tag{4.37}$$

form (4.35). Meanwhile, we have $\partial_\tau \partial_{\boldsymbol{n}} \psi = 0$, due to the fact that $\partial_{\boldsymbol{n}} \psi$ is identically zero on the boundary, because of the no-slip boundary condition. Thus we arrive at

$$\partial_x^2 \psi = \cos^2\theta \, \partial_\tau^2 \psi + \sin^2\theta \, \partial_{\boldsymbol{n}}^2 \psi \,. \tag{4.38}$$

It is obvious that

$$\partial_\tau^2 \psi = 0 \,, \quad \text{on } \partial\Omega \,, \tag{4.39}$$

because of no penetration boundary condition $\psi|_{\partial\Omega} = 0$. Therefore, we get the following identity on the boundary:

$$\partial_x^2 \psi = \sin^2\theta\, \partial_{\boldsymbol{n}}^2 \psi\,, \tag{4.40}$$

which results in

$$\partial_{\boldsymbol{n}}^2 \psi = \csc^2\theta\, \partial_x^2 \psi\,, \quad \text{at } A\,. \tag{4.41}$$

The combination of (4.36) and (4.41), along with the argument in (4.39), gives

$$\omega = (\partial_x^2 + \partial_y^2)\psi = \partial_{\boldsymbol{n}}^2 \psi = \csc^2\theta\, \partial_x^2 \psi\,, \quad \text{at } A\,. \tag{4.42}$$

Thus we obtain the relationship between ω and $\partial_x^2\psi$ on the boundary.

Plugging (4.31), which is a first order approximation of $\partial_x^2\psi$ at A, into the identity (4.42), we get

$$\omega = \csc^2\theta \cdot \frac{2}{a^2h^2}\psi_C\,, \tag{4.43}$$

which can be regarded as the Thom's formula for a curved boundary. We can get an analogue of Wilkes' formula on a curved boundary by plugging (4.32), which is a second order approximation to $\partial_x^2\psi$ at A, into the identity (4.42):

$$\omega = \csc^2\theta \cdot \frac{2(1+a)\psi_C - \frac{a^3}{(1+a)^2}\psi_D}{a^2h^2} \tag{4.44}$$

The derivation of the vorticity value at the projected boundary point B is similar to that of A. As shown in Fig. 1, we assume that $CF = FG = dy = h$, and $b = |BC|/h$, (note that $0 < b < 1$). By repeating a similar arguments as in (4.28)-(4.42), the vorticity value at point B can be approximated as

$$\omega_B = \sec^2\theta\, \partial_y^2 \psi\,. \tag{4.45}$$

The corresponding Thom's formula for ω_B turns out to be

$$\omega_B = \sec^2\theta \cdot \frac{2}{b^2h^2}\psi_C\,, \tag{4.46}$$

and the corresponding Wilkes' formula for ω_B turns out to be

$$\omega_B = \sec^2\theta \cdot \frac{2(1+b)\psi_C - b^3/(1+b)^2\psi_F}{b^2h^2}\,. \tag{4.47}$$

5 Numerical examples

In this section, we present a numerical experiment for solving the Navier Stokes equations using our algorithm. The computational domain is an elliptic disk whose boundary ($4x^2+16y^2=1$) is the zero level set of the following function $\varphi(x,y)$:

$$\varphi(x,y) = \sqrt{4x^2+16y^2}-1\,, \tag{5.48}$$

inside the square $[-1,1]^2$. The exact mean stream-function is chosen to be

$$\psi_e(x,y,t) = \begin{cases} \Big(r(x,y)-0.25\Big)^2 \cos t\,\sin^2(\pi x)\sin^2(\pi y)\cos t, & \text{if } \varphi(x,y)\le 0, \\ 0, & \text{otherwise.} \end{cases} \tag{5.49}$$

where $r(x,y)=x^2+4y^2$. It is straightforward to verify that the exact stream function is smooth inside the elliptic domain Ω and

$$\psi = 0\,, \quad \frac{\partial\psi}{\partial \boldsymbol{n}} = 0\,, \qquad \text{on} \qquad \partial\Omega\,, \tag{5.50}$$

so that the no penetration, no slip boundary condition is satisfied for the velocity field. The corresponding exact velocity and vorticity functions turn out to be

$$\begin{aligned} u_e(x,y,t) &= -\partial_y\psi_e(x,y,t) = -16y\Big(r(x,y)-0.25\Big)\text{cos}t\,, \\ v_e(x,y,t) &= \partial_x\psi_e(x,y,t) = 4x\Big(r(x,y)-0.25\Big)\text{cos}t\,, \\ \omega_e(x,y,t) &= \Delta\psi_e(x,y,t) = \Big[20\Big(r(x,y)-0.25\Big)+8(x^2+16y^2)\Big]\,\text{cos}t\,, \end{aligned} \tag{5.51}$$

inside the elliptic region Ω and zero outside.

The force term $\boldsymbol{f}$ can be calculated exactly from the vorticity transport equation

$$\begin{cases} \partial_t\omega_e + (\boldsymbol{v}_e\cdot\nabla)\omega_e = \nu\Delta\omega_e + \boldsymbol{f}\,, \\ \Delta\psi_e = \omega_e\,, \\ \psi_e = 0\,, \quad \dfrac{\partial\psi_e}{\partial\boldsymbol{n}} = 0\,, \qquad \text{on} \qquad \partial\Omega\,, \\ \boldsymbol{v}_e = \nabla^{\perp}\psi_e = (-\partial_y\psi_e, \partial_x\psi_e)\,. \end{cases} \tag{5.52}$$

The second order accurate numerical method proposed in this paper is used to solve the above system (5.52). The viscosity is chosen to be $\nu = 0.001$ so the Reynolds number is $Re = 1000$. The final time is taken to be $t = 0.5$.

Table 1 shows the grid refinement analysis for the velocity $\mathbf{u}$ and stream function ψ using Wilkes' formula. Note that the error is quite small. The error obtained from a very coarse grid 32 by 32 is $10^{-4} \sim 10^{-5}$ already. Second order convergence can be clearly seen for L^1, L^2, and L^∞ norms. Therefore, the whole scheme leads to very good approximation to both the variables that we are concerned.

	N	L^1 error	L^1 order	L^2 error	L^2 order	L^∞ error	L^∞ order
	32	5.07e-05		1.01e-04		3.40e-04	
	64	1.49e-05	1.77	2.84e-05	1.83	1.06e-04	1.68
$\boldsymbol{u}$	128	4.00e-06	1.90	7.55e-06	1.91	3.18e-05	1.74
	256	1.05e-06	1.93	2.05e-06	1.88	1.59e-05	1.00
	32	5.66e-06		1.09e-05		3.20e-05	
	64	1.51e-06	1.91	2.90e-06	1.91	8.30e-06	1.95
ψ	128	4.08e-07	1.89	7.71e-07	1.91	2.17e-06	1.95
	256	9.56e-08	2.09	1.81e-07	2.09	5.16e-07	2.07

Table 1: Error and order of accuracy for velocity and stream function at $t = 0.5$ with $\mu = 0.0001$ and the CFL condition $\frac{\Delta t}{h} = 0.5$. The **Wilkes' formula** for a curved boundary is used.

6 Conclusions

In this paper, a new second order finite difference method based on the vorticity stream-function formulation is developed for Navier Stokes equations defined on irregular domains. The key of the new method is the fast Poisson solver on irregular domains and the corresponding Thom's and Wilkes' formula on curved boundaries.

7 Acknowledgment

The first author is partially supported by a USA ARO grant 39676-MA, and a USA NSF grant DMS0073403.

References

[1] J. Adams, P. Swarztrauber, and R. Sweet. Fishpack. http://www.netlib.org/fishpack/.

[2] A. Almgren, J. Bell, P. Collella, and T. Marthaler. A cartesian grid projection method for the incompressible Euler equations in complex geometries. *SIAM J. Sci. Comput.*, 18:1289–1309, 1997.

[3] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 85:257–283, 1989.

[4] J. H. Bramble and J. T. King. A finite element method for interface problems in domains with smooth boundary and interfaces. preprint, 1995.

[5] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 168:464, 2001.

[6] D. Calhoun. A cartesian grid method for solving the streamfunction-vorticity equation in irregular regions. to appear in J. Comput. Phys., 2001.

[7] R. Cortez and M. L. Minion. The blob projection method for immersed boundary problems. *J. Comput. Phys.*, 161:428–453, 2000.

[8] R. H. Dillon, L. J. Fauci, and A. L. Fogelson. Modeling biofilm processes using the immersed boundary method. *J. Comput. Phys.*, 129:57–73, 1996.

[9] W. E. and J. Liu. Essentially compact schemes for unsteady viscous incompressible flows. *J. Comput. Phys.*, 126:122–138, 1996.

[10] W. E. and J. Liu. Vorticity boundary condition and related issues for finite difference schemes. *J. Comput. Phys.*, 124:368–382, 1996.

[11] A. L. Fogelson. Continuum models for platelet aggregation: Formulation and mechanical properties. *SIAM J. Numer. Anal.*, 52:1089, 1992.

[12] T. Hou, Z. Li, S. Osher, and H. Zhao. A hybrid method for moving interface problems with application to the Hele-Shaw flow. *J. Comput. Phys.*, 134:236–252, 1997.

[13] T.Y. Hou and B.T.R. Wetton. Convergence of a finite difference scheme for the Navier-Stokes equations using vorticity boundary conditions. *SIAM J. Numer. Anal.*, 29:615–639, 1992.

[14] M. Kang, R. Fedkiw, and X. Liu. A boundary condition capturing method for multiphase incompressible flow. *J. Sci. Comput*, 15:323–360, 2000.

[15] M-C. Lai and Z. Li. The immersed interface method for the Navier-Stokes equations with singular forces. *J. Comput. Phys.*, 171:822–842, 2001.

[16] R. J. LeVeque. Clawpack and Amrclaw – Software for high-resolution Godunov methods. 4-th Intl. Conf. on Wave Propagation, Golden, Colorado, 1998.

[17] Z. Li. A fast iterative algorithm for elliptic interface problems. *SIAM J. Numer. Anal.*, 35:230–254, 1998.

[18] Z. Li. IIMPACK, a collection of fortran codes for interface problems. http://www4.ncsu.edu/~zhilin/iimpack, or anonymous ftp.ncsu.edu in the directory: /pub/math/zhilin/Package, Last updated: 1998.

[19] Z. Li, H. Zhao, and H. Gao. A numerical study of electro-migration voiding by evolving level set functions on a fixed cartesian grid. *J. Comput. Phys.*, 152:281–304, 1999.

[20] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[21] C. S. Peskin and D. M. McQueen. A general method for the computer simulation of biological systems interacting with fluids. *Symposia of the Society for Experimental Biology*, 49:265, 1995.

[22] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider. A high-order projection method for tracking fluid interfaces in variable density incompressible flows. *J. Comput. Phys.*, 130:269–282, 1997.

[23] D. Sulsky and J. U. Brackbill. A numerical method for suspension flow. *J. Comput. Phys.*, 96:339–368, 1991.

[24] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, 100:25–37, 1992.

[25] C. Wang and J. Liu. Analysis of finite difference schemes for unsteady Navier-Stokes equations in vorticity formulation. *Numer. Math.*, in press, 2001.

[26] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundary. *J. Comput. Phys.*, 156:209–240, 1999.