COMM. MATH. SCI.
Vol. 1, No. 2, pp. 317–332

GAUGE METHOD FOR VISCOUS INCOMPRESSIBLE FLOWS*

WEINAN E† AND JIAN–GUO LIU‡

Abstract. We present a new formulation of the incompressible Navier-Stokes equation in terms of an auxiliary field that differs from the velocity by a gauge transformation. The gauge freedom allows us to assign simple and specific boundary conditions for both the auxiliary field and the gauge field, thus eliminating the issue of pressure boundary condition in the usual primitive variable formulation. The resulting dynamic and kinematic equations can then be solved by standard methods for heat and Poisson equations. A normal mode analysis suggests that in contrast to the classical projection method, the gauge method does not suffer from the problem of numerical boundary layers. Thus the subtleties in the spatial discretization for the projection method are removed. Consequently, the projection step in the gauge method can be accomplished by standard Poisson solves. We demonstrate the efficiency and accuracy of the gauge method by several numerical examples, including the flow past cylinder.

1. The gauge formulation

In this paper, we introduce a new formulation of the incompressible Navier-Stokes equation and demonstrate that this new formulation is particularly suited for numerical purpose. We start with the classical formulation of the Navier-Stokes equation:

$$\begin{cases} \boldsymbol{u}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p = \frac{1}{Re}\triangle\boldsymbol{u} \\ \nabla\cdot\boldsymbol{u} = 0 \end{cases} \tag{1.1}$$

on Ω, where $\boldsymbol{u} = (u, v)$ is the velocity and p is the pressure, with the simplest physical boundary condition:

$$\boldsymbol{u} = 0 \tag{1.2}$$

at $\Gamma = \partial\Omega$. By taking the divergence of the momentum equation one can obtain a Poisson equation for pressure:

$$-\triangle p = \nabla\cdot(\boldsymbol{u}\cdot\nabla\boldsymbol{u}) \tag{1.3}$$

This can be thought of as a replacement for the incompressibility condition. The task of solving (1.1) would become much easier if we could attach a simple boundary condition to this equation. Unfortunately, the most natural candidate obtained from extending (1.1) to the boundary:

$$\frac{\partial p}{\partial \boldsymbol{n}} = \boldsymbol{n}\cdot\triangle\boldsymbol{u} \tag{1.4}$$

involves evaluating the viscous term at the boundary. Not only is it difficult to enforce this condition accurately, but maintaining consistency between (1.3) and (1.4) (since this is a Neumann problem) in a discrete setting is also very difficult. As a result, a projection method was invented to by-pass the issue of the pressure boundary condition [3, 15, 10]. The price paid is that **special** discretization schemes have to be used to discretize the pressure Poisson equation. This greatly limits the simplicity

* Received: September 30, 2002; accepted (in revised version): October 13, 2002.

† Department of Mathematics and Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544 (weinan@princeton.edu).

‡ Institute for Physical Science and Technology and Department of Mathematics, University of Maryland, College Park, MD 20742 (jliu@math.umd.edu).

and flexibility of the projection method. This issue and a careful comparison between the projection method and the gauge method proposed in this paper will be addressed in Section 3.

The starting point of the gauge method is to replace pressure by a gauge variable ϕ, and introduce the auxiliary field $\boldsymbol{a} = \boldsymbol{u} + \nabla\phi$, and replace (1.1) by

$$\boldsymbol{a}_t + (\boldsymbol{u}\cdot\nabla)\boldsymbol{u} = \frac{1}{Re}\triangle\boldsymbol{a} \tag{1.5}$$

and

$$\triangle\phi = \nabla\cdot\boldsymbol{a} \tag{1.6}$$

It is easy to see that (1.1) and (1.5-1.6) are equivalent, and ϕ is related to the pressure by

$$p = \partial_t\phi - \frac{1}{Re}\triangle\phi \tag{1.7}$$

The main advantage of this formulation is that **we can use the gauge freedom to assign an unambiguous boundary condition for $\boldsymbol{a}$ and ϕ.** Corresponding to (1.2), we can either prescribe:

$$\frac{\partial\phi}{\partial\boldsymbol{n}} = 0, \quad \boldsymbol{a}\cdot\boldsymbol{n} = 0, \quad \boldsymbol{a}\cdot\boldsymbol{\tau} = \frac{\partial\phi}{\partial\boldsymbol{\tau}} \tag{1.8}$$

or

$$\phi = 0, \quad \boldsymbol{a}\cdot\boldsymbol{n} = \frac{\partial\phi}{\partial\boldsymbol{n}}, \quad \boldsymbol{a}\cdot\boldsymbol{\tau} = 0 \tag{1.9}$$

on Γ. Here $\boldsymbol{\tau}$ is the unit vector in the tangential direction.

2. The finite difference methods

(1.5)-(1.9) can be solved using many different numerical techniques, including spectral and finite element methods. In this paper, we choose to concentrate on finite difference methods with Neumann boundary conditions supplemented to the gauge field (1.8).

2.1. Temporal discretization. Depending upon the cell Reynolds number $Rc = Re\frac{\triangle x}{L}$ (L is the typical size of Ω), we can either choose to treat the viscous term explicitly or implicitly. With backward Euler in time, the finite difference scheme can be written as:

$$\frac{\boldsymbol{a}^{n+1} - \boldsymbol{a}^n}{\triangle t} + (\boldsymbol{u}^n\cdot\nabla_h)\boldsymbol{u}^n = \frac{1}{Re}\triangle_h\boldsymbol{a}^{n+1} \tag{2.1}$$

and

$$\triangle_h\phi^{n+1} = \nabla_h\cdot\boldsymbol{a}^{n+1} \tag{2.2}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{a}^{n+1} - \nabla_h\phi \tag{2.3}$$

We will specify the spatial discretization and incorporation of the boundary condition later. Similarly, one replaces (2.1) by

$$\frac{\boldsymbol{a}^{n+1} - \boldsymbol{a}^n}{\triangle t} + (\boldsymbol{u}^n\cdot\nabla_h)\boldsymbol{u}^n = \frac{1}{Re}\triangle_h\boldsymbol{a}^n \tag{2.4}$$

in a forward Euler scheme, with (2.2) and (2.3) unchanged. (2.1) can be easily improved to second order (in time) if we use Crank-Nicolson for the viscous term and Adams-Bashforth for the convection term. In fact, there is no difficulty in going to even higher order. This is particularly easy for explicit methods with Runge-Kutta in time.

2.2. Spatial discretization. We will concentrate on the case in which the viscous term is treated implicitly. There is a lot to be said about the case in which the viscous term is treated explicitly, but we will postpone this discussion to a separate paper.

The simplest way of discretizing in space is to define all numerical variables on grid points and use centered differences for ∇_h and the five-point formula for $\triangle_h$. For simplicity, we will denote the grid points by $\{(i,j), j \geq 0\}$ and specify $j = 0$ to be the physical boundary. Writing $\boldsymbol{a} = (a, b)$, we impose

$$b_{i,0}^{n+1} = 0, \quad a_{i,0}^{n+1} = 2\frac{\phi_{i+1,0}^n - \phi_{i-1,0}^n}{2\triangle x} - \frac{\phi_{i+1,0}^{n-1} - \phi_{i-1,0}^{n-1}}{2\triangle x} \tag{2.5}$$

In (2.5), to avoid coupling the boundary value of $\boldsymbol{a}$ with the interior values of ϕ at the same time step, it was necessary to use extrapolation from previous time steps. For the first order (in time) scheme, we can replace the last formula by

$$a_{i,0}^{n+1} = \frac{\phi_{i+1,0}^n - \phi_{i-1,0}^n}{2\triangle x} \tag{2.6}$$

The Neumann boundary condition for ϕ can be taken into account by using

$$\phi_{i,-1}^{n+1} = \phi_{i,1}^{n+1} \tag{2.7}$$

in the evaluation of $\triangle_h$. To evaluate $\nabla_h \cdot \boldsymbol{a}^{n+1}$ at the boundary $j = 0$, we use simple horizontal extrapolation:

$$f_0 = 3f_1 - 3f_2 + f_3 \tag{2.8}$$

At corner points, the same extrapolation formula can be used in the diagonal direction.

An alternative strategy is to define the numerical variables at the centers of the computational cells. In this case, the simplest way of implementing the boundary conditions is to use the reflection technique:

$$b_{i,-\frac{1}{2}}^{n+1} = -b_{i,\frac{1}{2}}^{n+1}, \quad \phi_{i,-\frac{1}{2}}^{n+1} = \phi_{i,\frac{1}{2}}^{n+1} \tag{2.9}$$

$$\frac{1}{2}(a_{i,-\frac{1}{2}}^{n+1} + a_{i,\frac{1}{2}}^{n+1}) = 2\frac{\phi_{i+1,-\frac{1}{2}}^n - \phi_{i-1,-\frac{1}{2}}^n}{2\triangle x} - \frac{\phi_{i+1,-\frac{1}{2}}^{n-1} - \phi_{i-1,-\frac{1}{2}}^{n-1}}{2\triangle x} \tag{2.10}$$

and use the same strategy as described above to compute $\nabla_h \cdot \boldsymbol{a}^{n+1}$ at the boundary points $\{(i, \frac{1}{2})\}$.

2.3. Numerical result: the driven cavity flow. We implemented the explicit gauge method on a canonical test problem – the driven cavity flow. The computational domain is $[0,1] \times [0,1]$. The boundary condition $u = 0, v = 0$ is imposed everywhere at the boundary except at $\{y = 1\}$ where we impose $u = 1, v = 0$. For initial data we

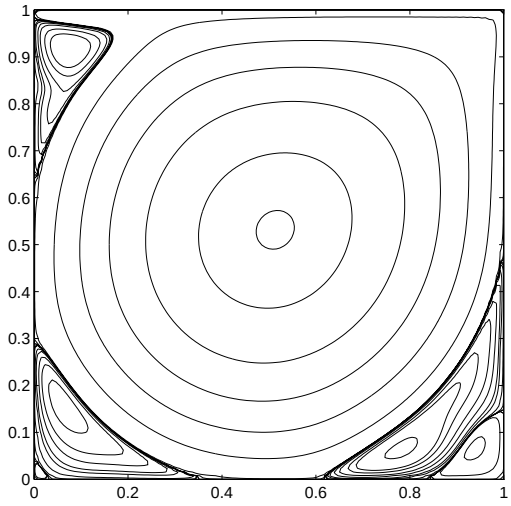

FIG. 1. *Driven cavity flow at* $Re = 10^4, t = 1000$ *with* 128×128 *grid points.*

choose the impulsive start: $u = 0, v = 0$. We used classical Runge-Kutta to discretize time.

Figure 1 is the contour plot of stream function at Reynolds number 10^4 and $t = 1000$ with 128×128 grid points. It agrees very well with previous results obtained in [9, 14] by solving directly the steady state equation.

We used a MAC grid for the space discretization, in which the exact projection can be realized at the discrete level. We shall point out that when the viscosity term is treated explicitly, the MAC scheme is exactly the same as the projection method, as we explain below. It also performs very similarly to the explicitly gauge method. We now use the forward Euler to explain the relationship between the MAC scheme, the explicit projection method and the explicit gauge method. We use $\boldsymbol{u}^n$ to denote the approximation of $\boldsymbol{u}$ at t^n in a MAC staggered grid. That is, the x-component (y-component) of $\boldsymbol{u}$ is discretized on the vertical side (horizontal side) of the cell boundary. And the boundary condition is understand as the reflection boundary condition (see [5] for details). We use the notation of ∇_h, Δ_h as the difference approximation of ∇ and Δ in the MAC scheme. Then, the forward Euler discretization of the MAC scheme is given by,

$$\begin{cases} \frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla_h)\boldsymbol{u}^n + \nabla_h p^{n+1} = \frac{1}{Re}\Delta_h \boldsymbol{u}^n \\ \nabla \cdot \boldsymbol{u}^{n+1} = 0 \\ \boldsymbol{u}^{n+1} \cdot \boldsymbol{n} = 0, \qquad \boldsymbol{u}^{n+1} \cdot \tau = 0 \qquad \text{on } \Gamma\,. \end{cases} \tag{2.11}$$

Above, we have written the boundary condition into the normal and tangent component, which will be used below to explain the equivalentce between different schemes. Introducing an intermittent variable u^*, (2.11) can be written in the following splitting form:

$$\frac{\boldsymbol{u}^*-\boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla_h)\boldsymbol{u}^n = \frac{1}{Re}\Delta_h \boldsymbol{u}^n \tag{2.12a}$$

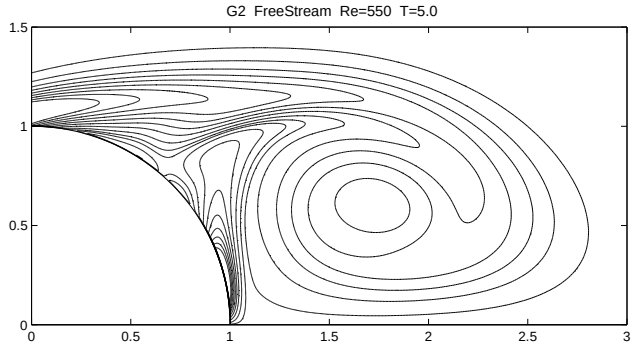

FIG. 2. *Impulsive started flow past an circular cylinder at* $Re = 550, t = 5$ *with* 513×768 *grid points, computed by the second order Gauge method. The free-stream far-field boundary condition is used.*

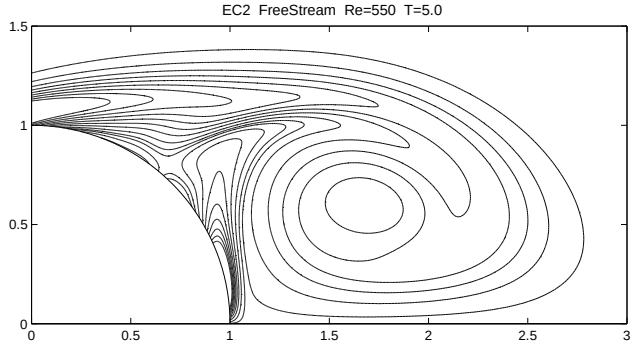

FIG. 3. *All the computational and physical parameters are same as in Figure 2. The computation is done by the second order finite difference in vorticity stream function formulation with Thom's boundary condition and RK4 time stepping.*

$$\begin{cases} \frac{\boldsymbol{u}^{n+1}-\boldsymbol{u}^*}{\triangle t} + \nabla_h p^{n+1} = 0 \\ \nabla \cdot \boldsymbol{u}^{n+1} = 0 \\ \boldsymbol{u}^{n+1} \cdot \boldsymbol{n} = 0 \qquad \text{on } \Gamma \end{cases} \tag{2.12b}$$

and we enforce

$$\boldsymbol{u}^{n+1} \cdot \tau = 0 \qquad \text{on } \Gamma \tag{2.12c}$$

Since (2.12a) is an explicit evaluation, there is no need a boundary condition for $\boldsymbol{u}^*$. However, when (2.12b) is implemented, a boundary value of $\boldsymbol{u}^*$ is needed. We now argue that it can be any number, say α_b; hence, we can set it to be zero. Reformat (2.12b) in the Poisson equation form,

$$\begin{cases} \Delta_h p^{n+1} = \frac{\nabla_h \boldsymbol{u}^*}{\triangle t}, \qquad \boldsymbol{u}^* \cdot \boldsymbol{n} = \alpha_b \\ \frac{D}{D\boldsymbol{n}} p^{n+1} = \triangle t \alpha_b \end{cases} \tag{2.13}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \triangle t \nabla_h p^{n+1} \tag{2.14}$$

Homogenize the boundary condition in a MAC grid, and we will find that (2.13) is equivalent to

$$\begin{cases} \Delta_h p^{n+1} = \frac{\nabla_h \boldsymbol{u}^*}{\triangle t}, & \boldsymbol{u}^* \cdot \boldsymbol{n} = 0 \\ \frac{D}{D\boldsymbol{n}} p^{n+1} = 0 \end{cases} \tag{2.13$'$}$$

Therefore, we have shown that the MAC scheme is equivalent to:

$$\begin{cases} \frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla_h)\boldsymbol{u}^n = \frac{1}{Re}\Delta_h \boldsymbol{u}^n \\ \boldsymbol{u}^* \cdot \boldsymbol{n} = 0 \qquad \text{on } \Gamma \end{cases} \tag{2.15a}$$

$$\begin{cases} \Delta_h p^{n+1} = \frac{\nabla_h \boldsymbol{u}^*}{\triangle t}, \\ \frac{D}{D\boldsymbol{n}} p^{n+1} = 0 \end{cases} \tag{2.15b}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \triangle t \nabla_h p^{n+1} \tag{2.15c}$$

$$\boldsymbol{u}^{n+1} = 0 \qquad \text{on } \Gamma \tag{2.15d}$$

This is exactly the explicit projection method.

In the gauge method, it changes to

$$\begin{cases} \frac{\boldsymbol{a}^{n+1} - \boldsymbol{a}^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla_h)\boldsymbol{u}^n = \frac{1}{Re}\Delta_h \boldsymbol{a}^n \\ \boldsymbol{a}^{n+1} \cdot \boldsymbol{n} = 0, \quad \boldsymbol{a}^{n+1} \cdot \tau = \frac{D}{D\tau}\phi^n, \qquad \text{on } \Gamma \end{cases} \tag{2.16a}$$

$$\begin{cases} \Delta_h \phi^{n+1} = \boldsymbol{a}^{n+1} \\ \frac{D}{D\boldsymbol{n}} \phi^{n+1} = 0 \end{cases} \tag{2.16b}$$

$$\boldsymbol{u}^{n+1} = \boldsymbol{a}^{n+1} - \nabla_h \phi^{n+1} \tag{2.16c}$$

$$\boldsymbol{u}^{n+1} = 0 \qquad \text{on } \Gamma \tag{2.16d}$$

The only difference is that $\boldsymbol{u}^n$ in the right hand side of (2.15a) be replaced by $\boldsymbol{a}^n$, which different from $\boldsymbol{u}^n$ by a gradient function.

A MAC grid not only gives the exact projection. But it also gives a natural way to define vorticity and stream function at the discrete level[6]. It was shown in [6] that the MAC scheme is equivalent to the standard 2nd order finite difference in vorticity stream function coupled with the Thom's boundary condition. If we take the discrete curl operator to both (2.15a) and (2.16a), this leads to same equation. Indeed, it is simply the vorticity transport equation in the discretized level.

$$\frac{\omega^{n+1} - \omega^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla_h)\omega^n = \frac{1}{Re}\Delta_h \omega^n \tag{2.17a}$$

The nonlinear term has a slight difference [6]. The kinetic relation in both (2.15b) and (2.1b) leads to

$$\begin{cases} \Delta_h \psi^{n+1} = \omega^{n+1} \\ \psi^{n+1} = 0, \qquad \text{on } \Gamma \end{cases} \tag{2.17b}$$

and the reflection boundary condition of the tangent velocity on the boundary becomes the Thom's formula [6], explaining the similarity in performance. A different variant formulation of the gauge method in which a standard grid was used and the velocity flux on the cell boundary was computed via averageing. The computation of the driven cavity flow was computed and obtained a similar result.

2.4. Numerical result: flow past a circular cylinder. The next example is a computation of the impulse started flow past a cylinder at Reynolds number Re=550[1]. The polar coordinate (r, θ) was used and the computational domain was chosen to be $[1, 8] \times [0, \pi]$. An exponentially scaled grid based on the variable $z = \ln r$ was used to cluster more points near cylinder and to regain a constant coefficient in the Laplace operation so that the FFT can be used to solve the resulted linear systems.

The explicit second order difference gauge method was used on a standard 513x768 grid and the free stream boundary condition was used at the far-field. The result of time $t = 5$ is plot in Figure 2. We also computed and plotted in Figure 3 the flow with exact computational and physical parameters for the second order difference in the vorticity-stream function with Thom's boundary, which we have extensively studied before [6]. The results are remarkably similar.

3. Comparison with the projection method

The projection method is a fractional step method that decouples the dynamic momentum equation from the kinematic constraint of incompressibility. A simple example is the following: Knowing $\boldsymbol{u}^n$, compute $\boldsymbol{u}^{n+1}$ by first computing an intermediate velocity field $\boldsymbol{u}^*$ using

$$\frac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla_h)\boldsymbol{u}^n = \triangle_h \boldsymbol{u}^* \tag{3.1}$$

with boundary condition $\boldsymbol{u}^* = 0$, and then projecting $\boldsymbol{u}^*$ to the space of divergence-free vector fields:

$$\begin{cases} \boldsymbol{u}^{n+1} + G_h \triangle t p^{n+1} = \boldsymbol{u}^* \\ D_h \cdot \boldsymbol{u}^{n+1} = 0 \end{cases} \tag{3.2}$$

The projection step can be written as

$$\tilde{\triangle}_h \triangle t p^{n+1} = D_h \cdot \boldsymbol{u}^*, \quad \partial_n p^{n+1} = 0 \tag{3.3}$$

with Neumann boundary condition, and

$$\boldsymbol{u}^{n+1} = \boldsymbol{u}^* - G_h \triangle t p^{n+1} \tag{3.4}$$

Here $\tilde{\triangle}_h = D_h G_h$.

At a first sight this is very close to the the gauge method (2.1-2.3). A momentum-like equation is solved, followed by a step of projecting to the space of divergence-free

[1]This computation was computed by Hans E. Johnston.

vector fields. The auxiliary variable $\boldsymbol{a}$ plays the role of the intermediate velocity field and ϕ (or $\frac{1}{\triangle t}\phi$) plays the role of pressure. However, as we discuss in this section, such similarities are rather superficial and it is their subtle differences that make the gauge method advantageous.

The most important difference is that the gauge method is **not** a fractional step method. (2.1) is a consistent discretization of a PDE, not a fractional step. While in the projection method, $\boldsymbol{u}^*$ never deviates much from the space of divergence-free fields, $\boldsymbol{a}$ in the gauge method is $O(1)$ away from it. We can formally write (3.1) in a way similar to (2.1) using $\boldsymbol{u}^n = \tilde{\boldsymbol{u}}^* - \triangle t p^n, q^n = \triangle t p^n$:

$$\begin{cases} \dfrac{\boldsymbol{u}^* - \tilde{\boldsymbol{u}}^*}{\triangle t} + (\boldsymbol{u}^n \cdot \nabla) \cdot \boldsymbol{u}^n + \dfrac{1}{\triangle t}\nabla q^n = \triangle \boldsymbol{u}^* \\ \\ \triangle q^{n+1} = \nabla \cdot \boldsymbol{u}^* \\ \boldsymbol{u}^{n+1} = \boldsymbol{u}^* - \nabla q^{n+1} \end{cases} \tag{3.5}$$

This looks very close to (2.1-2.3) except for the term $\frac{1}{\triangle t}\nabla q^n$. It turns out that this term makes a big difference as we discuss below. In fact, from a computational point of view, (2.1-2.3) resembles much more the vorticity-velocity formulation in which vorticity is first updated and velocity is then evaluated from the vorticity.

This difference has several profound consequences.

3.1. The projection method introduces additional numerical modes into the problem by formally increaseing the order of the PDE [4], i.e. the numerical method is a singular perturbation of the original PDE. Very often, the spurious numerical modes are of a numerical boundary layer type. This can be seen clearly from the following simple example: We consider the unsteady Stokes equation on $[-1,1] \times (0, 2\pi)$ with no-slip boundary conditions at $x = -1, 1$ and periodic boundary conditions in y. After a Fourier transform in the y variable, the problem is reduced to a family of one-dimensional problems indexed by $k \in Z$. For the gauge formulation, this is

$$\begin{cases} \partial_t a = (\partial_x^2 - k^2)a\,, \\ \partial_t b = (\partial_x^2 - k^2)b\,, \\ (\partial_x^2 - k^2)\phi = \partial_x a + ikb\,, \end{cases} \tag{3.6}$$

and

$$a = u + \partial_x \phi \qquad b = v + ik\phi \tag{3.7}$$

The boundary condition becomes

$$a(\pm 1, t) = \phi'(\pm 1, t) = 0\,, \qquad b(\pm 1, t) = ik\phi(\pm 1, t) \tag{3.8}$$

Here and in what follows, we use the notation $\boldsymbol{a} = (a, b)$,$\boldsymbol{u} = (u, v)$ and

$$\Delta_k = (\partial_x^2 - k^2)\,, \qquad \nabla_k = \begin{pmatrix} \partial_x \\ ik \end{pmatrix}\,,$$

The normal mode solutions of (3.6) are of the form:

$$(\boldsymbol{a}, \phi)(x, t) = e^{\sigma t}\,(\hat{\boldsymbol{a}}, \hat{\phi})(x)\,. \tag{3.9}$$

where $(\hat{a}, \hat{\phi})$ satisfies

$$\begin{cases} \sigma \hat{\boldsymbol{a}} = \Delta_k \hat{\boldsymbol{a}}\,, \\ \Delta_k \hat{\phi} = \nabla_k \cdot \hat{\boldsymbol{a}}\,, \end{cases} \tag{3.10}$$

with boundary conditions

$$\hat{a}(\pm 1) = \hat{\phi}'(\pm 1) = 0\,, \qquad \hat{b}(\pm 1) = ik\hat{\phi}(\pm 1) \tag{3.11}$$

Introducing μ such that

$$\sigma = -k^2 - \mu^2\,, \tag{3.12}$$

Then the first equation in (3.10) becomes

$$(\partial_x^2 + \mu^2)\hat{a} = 0\,, \qquad (\partial_x^2 + \mu^2)\hat{b} = 0\,, \qquad \hat{a}(\pm 1) = 0\,. \tag{3.13}$$

and the second becomes

$$(\partial_x^2 + \mu^2)(\partial_x^2 - k^2)\hat{\phi} = 0\,, \qquad \hat{\phi}'(\pm 1) = 0\,. \tag{3.14}$$

The odd solutions to (3.13) and (3.14) can be written as

$$\hat{a}(x) = 0, \qquad \hat{b}(x) = A \sin \mu x \tag{3.15}$$

and

$$\hat{\phi}(x) = \frac{\sinh kx}{k \cosh k} - \frac{\sin \mu x}{\mu \cos \mu}\,. \tag{3.16}$$

Now using the second equation in (2.10), we get

$$A = \frac{-\sigma}{ik\mu \cos \mu} \tag{3.17}$$

(3.11) From the boundary condition, we have

$$ik\left(\frac{1}{\mu}\tan \mu - \frac{1}{k}\tanh k\right) = \frac{\sigma}{ik\mu}\tan \mu \tag{3.18}$$

or

$$\mu \tan \mu + k \tanh k = 0 \tag{3.19}$$

Going back to $\boldsymbol{u}$ and p variables, we obtain

$$\begin{cases} \hat{u}(x) = \dfrac{\cos \mu x}{\cos \mu} - \dfrac{\cosh kx}{\cosh k} \\ \\ \hat{v}(x) = \dfrac{\mu}{ik}\dfrac{\sin \mu x}{\cos \mu} - i\dfrac{\sinh kx}{\cosh k}\,, \\ \\ \hat{p}(x) = \dfrac{\sigma}{k}\dfrac{\sinh kx}{\cosh k} \end{cases} \tag{3.20}$$

These are the symmetric solutions to the Stokes equation derived earlier in [4, 12]. The antisymmetric solutions can be find in a similar way.

Now we derive the normal mode solutions to the first order time discretization scheme (2.1-2.3), keeping space continuous. These are of the form:

$$(\boldsymbol{a}^n, \phi^n)(x) = \kappa^n \, (\hat{\boldsymbol{a}}, \hat{\phi})(x) \, . \tag{3.21}$$

where $(\hat{\boldsymbol{a}}, \hat{\phi})$ satisfies

$$\begin{cases} \dfrac{\kappa - 1}{\kappa \triangle t} \hat{\boldsymbol{a}} = \Delta_k \hat{\boldsymbol{a}} \, , \\ \\ \Delta_k \hat{\phi} = \nabla_k \cdot \hat{\boldsymbol{a}} \, , \end{cases} \tag{3.22}$$

and boundary conditions

$$\hat{a}(\pm 1) = \hat{\phi}'(\pm 1) = 0 \, , \qquad \kappa \hat{b}(\pm 1) = ik\hat{\phi}(\pm 1) \tag{3.23}$$

The factor κ at the front of $\hat{b}(\pm 1)$ comes from the boundary condition (2.6). Let

$$\bar{\sigma} = \frac{\kappa - 1}{\kappa \triangle t} \tag{3.24}$$

and

$$\bar{\sigma} = -k^2 - \bar{\mu}^2 \tag{3.25}$$

The solutions have exactly the same form as before, except (3.18) and (3.19) are replaced by

$$ik(\frac{1}{\bar{\mu}} \tan \bar{\mu} - \frac{1}{k} \tanh k) = \kappa \frac{\bar{\sigma}}{ik\bar{\mu}} \tan \bar{\mu} \tag{3.26}$$

Hence

$$\bar{\mu} \tan \bar{\mu} + k \tanh k = (\kappa - 1) \frac{\bar{\sigma}}{\bar{\mu}} \tan \bar{\mu} = \triangle t \frac{\bar{\sigma}^2}{\bar{\mu}(1 - \bar{\sigma} \triangle t)} \tan \bar{\mu} \tag{3.27}$$

Thus, we have

$$\bar{\mu} - \mu = O(\triangle t), \quad \kappa - e^{\sigma \triangle t} = O(\triangle t^2) \tag{3.28}$$

resulting in a first order accuracy of the overall scheme. Notice that there are no spurious numerical modes and the numerical method is uniformly first order accurate. The second order scheme can be analyzed in the same way.

In contrast, the solution to the projection method is of the form:

$$\begin{cases} \tilde{u}(x) = \dfrac{\cos\tilde{\mu}x}{\cos\tilde{\mu}} - \dfrac{\cosh kx}{\cosh k}\,, \\ \\ \tilde{v}(x) = \dfrac{\tilde{\mu}}{ik}\dfrac{\sin\tilde{\mu}x}{\cos\tilde{\mu}} - i\dfrac{\sinh kx}{\cosh k}\,, \\ \\ \tilde{u}^*(x) = \dfrac{\cos\tilde{\mu}x}{\cos\tilde{\mu}} - \dfrac{\cosh kx}{\cosh k} - \beta\triangle t\left(\dfrac{\cosh\lambda x}{\cosh\lambda} - \dfrac{\cosh kx}{\cosh k}\right)\,, \\ \\ \tilde{v}^*(x) = \dfrac{\tilde{\mu}}{ik}\dfrac{\sin\tilde{\mu}x}{\cos\tilde{\mu}} - i\dfrac{\sinh kx}{\cosh k} - ik\beta\triangle t\left(\dfrac{1}{\lambda}\dfrac{\sinh\lambda x}{\cosh\lambda} - \dfrac{1}{k}\dfrac{\sinh kx}{\cosh k}\right)\,, \\ \\ \tilde{p}(x) = -\beta\left(\dfrac{1}{\lambda}\dfrac{\sinh\lambda x}{\cosh\lambda} - \dfrac{1}{k}\dfrac{\sinh kx}{\cosh k}\right) \end{cases} \tag{3.29}$$

where

$$\lambda = (k^2 + \triangle t^{-1})^{1/2}\,, \quad \beta = -k^2 - \tilde{\mu}^2 \tag{3.30}$$

$\tilde{\mu}$, β and λ satisfies

$$\tilde{\mu}\tan\tilde{\mu} + k\tanh k = \beta k\,\triangle t\,(\tanh k - \frac{k}{\lambda}\tanh\lambda)\,. \tag{3.31}$$

This reveals clearly the existence of a numerical boundary layer mode associated with λ. Not only is the first order accuracy in pressure is lost, but spatial discretization also becomes subtle because of this spurious boundary layer mode.

3.2. Even though both ϕ^{n+1} and $\triangle t p^{n+1}$ solve a Poisson equation, in the projection method this Poisson equation has to be solved with very special schemes. This is highlighted in (3.3) by writing $\tilde{\triangle}_h$ instead of $\triangle_h$. The consistent projection method demands that a discretization of (3.3)has to come from

$$\begin{cases} \boldsymbol{u}^{n+1} + G_h\triangle t p^{n+1} = \boldsymbol{u}^* \\ D_h\cdot\boldsymbol{u}^{n+1} = 0 \end{cases} \tag{3.32}$$

with some conjugacy conditions between D_h and G_h. This is a mixed formulation for (3.3) and has all the complications of a mixed formulation. The most well-known example is the existence of parasitic modes. If standard second order centered differences are used for D_h and G_h on a non-staggered grid, then the resulting discrete Laplace operator $\triangle_h = D_h\cdot G_h$ has the spurious checkerboard modes in its kernel. Several ways of relaxing this framework have been proposed (see for example [13] and [1]). Still, stringent constraints seem to exist on schemes for (3.3).

The gauge method, on the other hand, is very robust and free of such subtleties. Straightforward methods can be used to discretize (2.2). To demonstrate this difference, we show in the following a comparison of the numerical results obtained from the first and second order gauge method and the projection method. For the first order projection method, we implemented (3.1-3.4) with both the homogeneous

boundary condition $\boldsymbol{u}^* = 0$ (denoted by PM1.1) and the improved boundary condition $\boldsymbol{u}^* \cdot \boldsymbol{n} = 0$, $\boldsymbol{u}^* \cdot \boldsymbol{\tau} = \triangle t \partial_{\boldsymbol{\tau}} p^n$ [8, 11, 18] (denoted by PM1.2) for the intermediate velocity field. Two different formulations of the second order projection method are chosen for this comparison, one based on the improved velocity boundary condition [8, 11, 18] (denoted by PM2.1). This is same as PM1.2 except that backward Euler is replaced by Crank-Nicolson, and Adams-Bashforth is used to compute the convective term. The second one is based on the pressure increment form [16, 2] (denoted by PM2.2):

$$\begin{cases} \dfrac{\boldsymbol{u}^* - \boldsymbol{u}^n}{\triangle t} + (\boldsymbol{u}^{n+1/2} \cdot \nabla)\boldsymbol{u}^{n+1/2} + \nabla p^{n-1/2} = \Delta \dfrac{\boldsymbol{u}^* + \boldsymbol{u}^n}{2}\,, \\ \boldsymbol{u}^* = 0\,, \qquad \text{on } \partial\Omega\,, \\ \boldsymbol{u}^* = \boldsymbol{u}^{n+1} + \triangle t(\nabla p^{n+1/2} - \nabla p^{n-1/2})\,, \\ \nabla \cdot \boldsymbol{u}^{n+1} = 0\,, \\ \boldsymbol{u}^{n+1} \cdot \boldsymbol{n} = 0\,, \qquad \text{on } \partial\Omega\,. \end{cases}$$

For the spatial discretization we used the two methods described earlier in Section 2.2

We choose the following exact solution of the Navier-Stokes equation:

$$\begin{cases} u(x,y,t) = -\cos(t)\sin^2(\pi x)\sin(2\pi y) \\ v(x,y,t) = \cos(t)\sin(2\pi x)\sin^2(\pi y) \\ \phi(x,y,t) = \cos(t)(2+\cos(\pi x))(2+\cos(\pi y))/4 \end{cases} \tag{3.34}$$

Appropriate forcing terms are added to ensure that (3.34) is an exact solution. We set $\nu = 1$ and $\triangle t = \triangle x$ so that the time step size is set by the convective term.

Tables 1-6 summarize the results for the case when numerical values are defined at the cell centers. Tables 7-12 summarizes the results for the second example in which numerical values are defined at the grid points. In both cases, the gauge methods achieve the expected accuracy. This is clearly not the case for the projection method. In the worst cases, the accuracy is entirely lost in the maximum norm.

One curious but not well-understood point is that the projection methods do not perform badly in keeping the velocity field approximately divergence-free. In fact, it often outperforms the gauge method in this respect.

We should remark that the poor performance of the projection method documented here does not imply that projection method cannot be used. It only implies that naive spatial discretization schemes for the PPE as the ones used here does not work for the projection method.

GM1	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	5.01E-3	1.37E-3	3.57E-4	1.91	6.37E-3	1.69E-3	4.37E-4	1.93	1.68E-2	4.98E-3	1.44E-3	1.77
$\boldsymbol{u}$	1.25E-2	7.37E-3	3.99E-3	0.83	1.48E-2	8.57E-3	4.60E-3	0.85	4.10E-2	2.51E-2	1.39E-2	0.79
$\boldsymbol{a}$	7.99E-2	4.33E-2	2.25E-2	0.92	8.89E-2	4.82E-2	2.50E-2	0.92	1.83E-1	1.01E-1	5.28E-2	0.90
ϕ	2.10E-2	1.14E-2	5.91E-3	0.92	2.59E-2	1.41E-2	7.32E-3	0.92	7.41E-2	4.09E-2	2.14E-2	0.90

GM2	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	5.00E-3	1.37E-3	3.57E-4	1.91	6.43E-3	1.71E-3	4.38E-4	1.94	1.87E-2	4.83E-3	1.22E-3	1.97
$\boldsymbol{u}$	2.01E-3	5.25E-4	1.33E-4	1.96	2.34E-3	5.97E-4	1.50E-4	1.98	4.22E-3	1.07E-3	2.68E-4	1.99
$\boldsymbol{a}$	2.57E-3	6.69E-4	1.69E-4	1.97	2.90E-3	7.57E-4	1.91E-4	1.97	7.44E-3	1.99E-3	5.11E-4	1.93
ϕ	8.24E-4	2.14E-4	5.40E-5	1.97	1.03E-3	2.66E-4	6.72E-5	1.97	2.95E-3	7.84E-4	1.99E-4	1.95

PM1.1	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	2.12E-3	4.04E-4	6.87E-5	2.48	3.51E-3	7.32E-4	1.39E-4	2.33	1.87E-2	5.37E-3	1.52E-3	1.81
$\boldsymbol{u}$	3.98E-2	2.80E-2	1.74E-2	0.60	4.85E-2	3.45E-2	2.19E-2	0.58	1.35E-1	1.13E-1	7.75E-2	0.40
p	8.00E-1	5.92E-1	3.82E-1	0.53	1.09E-0	8.08E-1	5.32E-1	0.52	3.73E-0	2.80E-0	1.92E-0	0.50

PM1.2	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	1.07E-3	1.74E-4	2.53E-5	2.70	1.39E-3	2.25E-4	3.28E-5	2.70	3.86E-3	6.84E-4	1.22E-4	2.49
$\boldsymbol{u}$	7.19E-3	3.50E-3	1.71E-3	1.04	8.52E-3	4.10E-3	2.00E-3	1.05	1.60E-2	7.52E-3	3.54E-3	1.09
p	3.95E-1	2.61E-1	1.53E-1	0.68	5.33E-1	3.46E-1	2.01E-1	0.70	1.61E-0	9.99E-1	5.70E-1	0.75

PM2.1	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	1.32E-3	1.90E-4	2.55E-5	2.85	1.80E-3	2.54E-4	3.90E-5	2.77	6.00E-3	1.21E-3	1.01E-3	1.29
$\boldsymbol{u}$	1.71E-3	1.09E-3	6.20E-4	0.73	2.85E-3	2.23E-3	1.66E-3	0.39	1.12E-2	1.21E-2	1.27E-2	-.09
p	2.21E-1	1.29E-1	6.75E-2	0.86	3.08E-1	1.75E-1	8.95E-2	0.89	9.67E-1	5.19E-1	2.56E-1	0.96

PM2.2	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	1.71E-4	3.56E-5	8.12E-6	2.20	2.75E-4	8.69E-5	3.02E-5	1.59	1.91E-3	1.65E-3	1.14E-3	0.37
$\boldsymbol{u}$	3.30E-3	1.42E-3	6.90E-4	1.13	4.36E-3	2.38E-3	1.62E-3	0.71	1.55E-2	1.29E-2	1.26E-2	0.15
p	1.22E-1	5.77E-2	3.53E-2	0.89	1.57E-1	7.36E-2	4.36E-2	0.92	6.34E-1	2.77E-1	1.40E-1	1.09

GM 1	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	5.48E-3	1.45E-3	3.71E-4	1.94	6.88E-3	1.86E-3	4.88E-4	1.91	3.77E-2	3.12E-2	1.87E-2	1.51
$\boldsymbol{u}$	2.34E-2	1.32E-2	7.01E-3	0.87	2.69E-2	1.50E-2	7.94E-3	0.88	6.41E-2	3.74E-2	2.03E-2	0.83
$\boldsymbol{a}$	1.04E-1	5.49E-2	2.81E-2	0.94	1.11E-1	5.93E-2	3.06E-2	0.93	1.87E-1	1.02E-1	5.30E-2	0.91
ϕ	2.33E-2	1.20E-2	6.05E-3	0.97	2.79E-2	1.46E-2	7.44E-3	0.95	7.77E-2	4.18E-2	2.15E-2	0.93

GM 2	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	5.47E-3	1.43E-3	3.65E-4	1.95	6.81E-3	1.75E-3	4.43E-4	1.97	1.96E-2	5.03E-3	1.27E-3	1.97
$\boldsymbol{u}$	2.59E-3	6.52E-4	1.63E-4	2.00	3.10E-3	7.78E-4	1.95E-4	2.00	6.02E-3	1.51E-3	3.78E-4	2.00
$\boldsymbol{a}$	3.41E-3	8.30E-4	2.04E-4	2.03	3.69E-3	9.08E-4	2.24E-4	2.02	8.00E-3	2.01E-3	5.04E-4	1.99
ϕ	9.32E-4	2.25E-4	5.51E-5	2.04	1.13E-3	2.76E-4	6.82E-5	2.03	3.17E-3	7.99E-4	2.00E-4	1.99

PM 1.1	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	2.43E-3	4.38E-4	7.23E-5	2.54	3.96E-3	7.91E-4	1.47E-4	2.38	2.15E-2	5.98E-3	1.66E-3	1.85
$\boldsymbol{u}$	4.38E-2	2.94E-2	1.78E-2	0.65	5.27E-2	3.62E-2	2.25E-2	0.61	1.57E-1	1.22E-1	8.04E-2	0.48
p	1.05E-0	6.82E-1	4.11E-1	0.68	1.31E-0	8.88E-1	5.59E-1	0.61	3.84E-0	2.87E-0	1.93E-0	0.50

PM 1.2	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	1.15E-3	1.83E-4	2.61E-5	2.73	1.47E-3	2.36E-4	3.42E-5	2.71	4.41E-3	1.60E-3	5.42E-4	1.51
$\boldsymbol{u}$	7.73E-3	3.62E-3	1.73E-3	1.08	8.94E-3	4.19E-3	2.02E-3	1.07	1.63E-2	7.54E-3	3.54E-3	1.10
p	8.26E-1	2.97E-1	1.64E-1	1.17	9.70E-1	3.75E-1	2.10E-1	1.10	2.36E-0	9.96E-1	5.68E-1	1.03

PM 2.1	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	1.37E-3	2.10E-4	3.25E-5	2.70	1.76E-3	2.79E-4	5.47E-5	2.50	5.34E-3	2.29E-3	9.05E-4	1.28
$\boldsymbol{u}$	2.89E-3	1.19E-3	5.26E-4	1.23	4.54E-3	2.50E-3	1.63E-3	0.74	1.99E-2	2.29E-2	2.40E-2	-.14
p	3.91E-2	2.19E-2	1.19E-2	0.86	4.85E-2	2.79E-2	1.53E-2	0.83	1.20E-1	7.23E-2	4.10E-2	0.77

PM 2.2	L_1				L_2				L_∞			
	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord	32^2	64^2	128^2	ord
div $\boldsymbol{u}$	1.99E-4	3.64E-5	8.35E-6	2.29	3.88E-4	1.40E-4	5.30E-5	1.44	3.04E-3	2.29E-3	1.81E-3	0.37
$\boldsymbol{u}$	3.47E-3	1.24E-3	5.26E-4	1.36	4.72E-3	2.35E-3	1.57E-3	0.79	1.62E-2	1.86E-2	2.04E-2	-.17
p	2.03E-2	6.78E-3	2.31E-3	1.57	2.59E-2	8.58E-3	2.93E-3	1.57	1.13E-1	4.51E-2	1.78E-2	1.33

Some light can be shed on the performance of both methods by looking at the smoothness of the error. This is displayed in Figure 4 for the second order gauge method, and in Figure 5 for PM2.1. These two figures are representative of the error structure we see in the two methods. The errors (in all variables) in the second order gauge method always behave as in Figure 4, and the errors in the second order projection method always behave as in Figure 5, for both spatial discretization schemes and both formulations of the projection method. We mention that the errors in the first order projection method have a different structure. This is a point we will return to in future work.

It is becoming increasingly clear that the smoothness of the error, particularly near the boundary, plays a very important role in the overall accuracy of the difference scheme. This is one reason why one-sided differences cannot be arbitrarily used at the boundary. This is very well documented for the vorticity-based finite difference methods in [6].

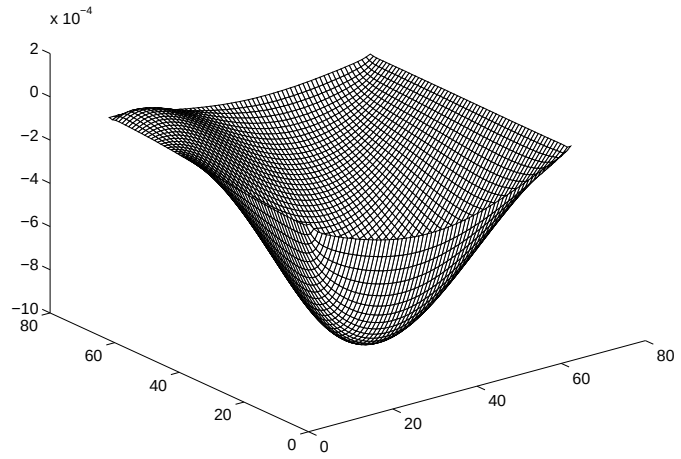

FIG. 4. *Error in velocity for GM2.*

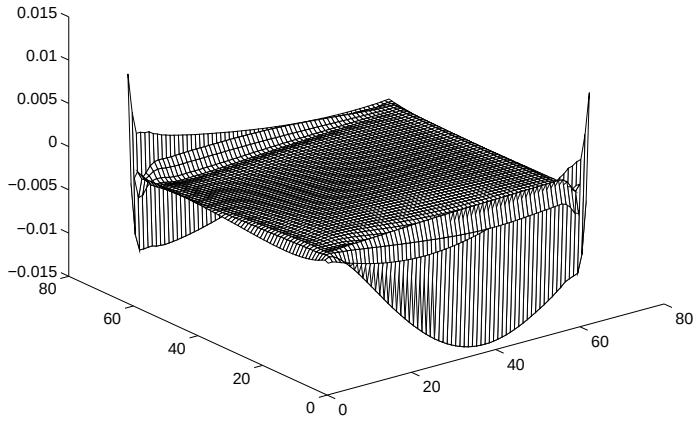

FIG. 5. *Error in velocity for PM2.1.*

4. Conclusions

We presented a new formulation of the Navier-Stokes equation in terms of the gauge variable and demonstrated that the gauge freedom can be used to eliminate the subtleties in the boundary conditions. The most important advantage of this formulation is that it is trouble-free – solving the Navier-Stokes equation is no more difficult than solving the standard heat and Poisson equations. We restricted the discussion to finite difference schemes, but there is no reason why finite element and spectral methods cannot be used. In fact, we expect that the current formulation with finite element discretization should lead to a numerical method most suitable for complicated geometries.

Acknowledgments. We thank P. Smereka for very stimulating discussion. The work of E was supported in part by ONR grant N00014-96-1-1013 and NSF grant DMS-9623137. The work of Liu was supported in part by ONR grant N00014-96-1-1013 and NSF grant DMS-9505275 and DMS-0107218.

REFERENCES

[1] A.S. Almgren, J.B. Bell, and W.G. Szymczak, *A numerical method for the incompressible Navier-Stokes equations based on an approximate projection.* SIAM J. Sci. Comput., 17:358–369, 1996.

[2] 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.

[3] A.J. Chorin, *Numerical solution of the Navier-Stokes equations.* Math. Comp., 22:745-762, 1968.

[4] Weinan E and J.G. Liu, *Projection method I: convergence and numerical boundary layers.* SIAM J. Numer. Anal., 32:1017–1057, 1995. *Projection method II: Godunov-Ryabenki analysis.* SIAM J. Numer. Anal., 33:1597–1621, 1996.

[5] Weinan E and J.G. Liu, *Finite difference schemes for incompressible flows in the velocity-impulse density formulation.* J. Comput. Phys., 130:67–76, 1997.

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

[7] Weinan E and J.G. Liu, *Gauge finite element method for incompressible flows.* Int. J. Num. Meth. Fluids., 34:701-710, 2000.

[8] M. Fortin, R. Peyret, and R. Temam, J. Mecanique., 10:357–390, 1971.

[9] U. Ghia, K.N. Ghia, and C.T. Shin, *High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method.* J. Comput. Phys., 48:387–411, 1982.

[10] P.M. Gresho and R.L. Sani, *On pressure boundary conditions for the incompressible Navier-Stokes equations.* Int. J. Numer. Methods Fluids., 7:1111–1145, 1987.

[11] J. Kim and P. Moin, *Application of a fractional-step method to incompressible Navier-Stokes equations.* J. Comp. Phys., 59:308–323, 1985.

[12] S.A. Orszag, M. Israel, and M.O. Deville, *Boundary conditions for incompressible flows.* J. Scientific Computing, 1:75–111, 1986.

[13] C.M. Rhie and W.L. Chow, AIAA J., 21:1525, 1983.

[14] R. Schreiber and H.B. Keller, *Driven cavity flows by efficient numerical techniques.* J. Comput. Phys., 49:310–333, 1983.

[15] R. Temam, *Sur l'approximation de la solution des equations de Navier-Stokes par la méthode des fractionnarires II.* Arch. Rational Mech. Anal., 33:377–385, 1969.

[16] J. van Kan, *A second-order accurate pressure-correction scheme for viscous incompressible flow.* SIAM J. Sci. Stat. Comp., 7:870–891, 1986.

[17] C. Wang and J.G. Liu, *Convergence of gauge method for incompressible flow.* Math. Comp., 69:1385-1407, 2000.

[18] T. Zang and M.Y. Hussaini, *On spectral multigrid methods for the time-dependent Navier-Stokes equations.* Appl. Math. Comp., 19:359, 1986.