Finite Difference Schemes for Incompressible Flows in Vorticity Formulations

Weinan E[*]
Courant Institute of Mathematical Sciences
New York University
New York, NY 10012, USA

Jian-Guo Liu[†]
Department of Mathematics
Temple University
Philadelphia, PA 19122, USA

Abstract

In this paper, we review some recent progress made in [4, 5, 6] on finite difference schemes for viscous incompressible flows using vorticity formulation. The main purpose of this series of papers [4, 5, 6] is to resurrect the idea of using *local* vorticity boundary condition for unsteady calculation. The emphasis is on simplicity of the methods. Three main issues will be discussed: efficient time-stepping procedures and cell Reynolds number constraints, efficient methods in 3D on non-staggered grids and efficient high order methods using compact differencing.

1   Introduction

In this paper, we review some recent progress made in [4, 5, 6] on finite difference schemes for viscous incompressible flows using vorticity formulation. The main purpose of this series of papers [4, 5, 6] is to resurrect the idea of using *local* vorticity boundary condition for unsteady calculation. The emphasis is on simplicity of the methods. It is very important in these methods to use *explicit* treatment of the viscous term. Therefore questions on stability and cell Reynolds number constraints have to be addressed. We do this in Section 2. Section 3 is about 3D non-staggered grids, and in Section 4 we present stable and efficient 4th order schemes using compact differencing.

2   Basic issues: Local vorticity boundary condition, cell Reynolds number and equivalent MAC formulation

The 2D Navier-Stokes equation in vorticity streamfunction formulation reads: $(\mathbf{u} = (u, v))$

$$
\begin{aligned}
\partial_t \omega + (\mathbf{u} \cdot \nabla)\omega &= \nu \Delta \omega\,, \\
\Delta \psi &= \omega\,, \\
u = -\partial_y \psi\,, \quad v &= \partial_x \psi
\end{aligned}
\tag{2.1}
$$

with the boundary condition

$$
\psi = 0, \quad \frac{\partial \psi}{\partial \mathbf{n}} = 0\,.
$$

2.1   Local vorticity boundary conditions

At the grid points, (2.1) is discretized using:

$$\frac{d\omega}{dt} - \tilde{D}_y\psi \, \tilde{D}_x\omega + \tilde{D}_x\psi \, \tilde{D}_y\omega = \nu\Delta_h\omega \,,$$
$$\Delta_h\psi = \omega \,,$$

(2.2)

where $\tilde{D}_x, \tilde{D}_y$ are the standard centered differences and Δ_h is the standard 5-point Laplacian. We will use i and j to number the grid lines in the x and y directions respectively, with $i = 0$ at the boundary Γ_y and $j = 0$ at Γ_x. The no penetration boundary condition $\psi = 0$ is imposed on Γ in the solution of the discrete Poisson equation. The no-slip condition is imposed (say on Γ_x) via:

$$\frac{\psi_{i,1} - \psi_{i,-1}}{2\triangle y} = 0 \,.$$

(2.3)

where $(i, -1)$ refers to the "ghost" grid point outside of the computational domain. Since $\psi_{i,0} = \psi_{i-1,0} = \psi_{i+1,0} = 0$, (2.3) implies

$$\omega_{i,0} = (\triangle_h\psi)_{i,0} = \frac{\psi_{i+1,0} - 2\psi_{i,0} + \psi_{i-1,0}}{\Delta x^2} + \frac{\psi_{i,1} - 2\psi_{i,0} + \psi_{i,-1}}{\Delta y^2} = \frac{2}{\Delta y^2}\psi_{i,1}$$

(2.4)

which is the well-known Thom's formula [13]. Many variants of Thom's formula were summarized in [4].

Explicit time-stepping procedure: Here we use forward Euler as an illustration. The fully discrete scheme,

$$\frac{\omega^{n+1} - \omega^n}{\Delta t} + (\mathbf{u}^n \cdot \nabla_h)\omega^n = \nu\Delta_h\omega^n \,, \quad \text{for } i, j \geq 1 \,,$$
$$\Delta_h\,\psi^{n+1} = \omega^{n+1} \,, \quad \text{for } i, j \geq 1 \,,$$
$$\psi^{n+1}|_\Gamma = 0 \,,$$
$$D_\mathbf{n}\psi^{n+1}|_\Gamma = 0$$

(2.5)

can be realized by a simple three-step marching procedure: Given $\{\omega_{i,j}^n\}$, $\{\omega_{i,j}^{n+1}\}$ is computed by:

Step 1. Update vorticity at the interior grid points by

$$\frac{\omega^{n+1} - \omega^n}{\Delta t} - \tilde{D}_y\psi^n \, \tilde{D}_x\omega^n + \tilde{D}_x\psi^n \, \tilde{D}_y\omega^n = \nu\Delta_h\omega^n \,.$$

(2.6)

Step 2. Solve

$$\Delta_h\,\psi^{n+1} = \omega^{n+1} \,,$$

(2.7)

with the boundary condition: $\psi^{n+1}|_\Gamma = 0$.

Step 3. Update vorticity at the boundary using

$$\omega_{i,0}^{n+1} = \frac{2}{\Delta y^2}\psi_{i,1}^{n+1} \,, \quad \omega_{0,j}^{n+1} = \frac{2}{\Delta x^2}\psi_{1,j}^{n+1} \,.$$

(2.8)

The key here is that (2.7) can be solved without knowing the boundary value of ω^{n+1}. Hence *there is no need to iterate between ψ^{n+1} and the boundary value of ω^{n+1}.* At every time step, only one *standard* Poisson solve is required.

This is a very simple method. It is spatially second order accurate, and higher order accuracy in time can be achieved by replacing forward Euler with higher order Runge-Kutta [4]. A natural question arises: *Why isn't this method more popular?* In fact, methods like this were very popular in the 60's and early 70's. But many difficulties and controversies were encountered and were not resolved even until recently. The situation at that time was reviewed in [10]. Beginning at the early 80's, global vorticity boundary conditions were invented to overcome the problems encountered earlier [12, 9]. Since then, global vorticity boundary conditions have dominated the field, and local methods such as the one described above were largely abandoned. This is very clearly reflected in the review articles of Gresho [7, 8] and the recent book of Quartapelle [12].

As we emphasized in [4], the difficulties mentioned earlier were largely due to the fact that the viscous term is treated implicitly in almost all early applications of the local vorticity boundary condition. As a result, a coupled system involving both interior and boundary values of vorticity and streamfunction has to be solved at each time step, and this is the source of most complications. The above time-stepping procedure avoids this problem. But since the viscous term was treated explicitly, one naturally worries about the stability constraint on the time step size. To see whether this is a problem, let us compare

$$\Delta t_1 = \frac{\Delta x}{U}, \qquad \Delta t_2 = \frac{1}{4}\frac{\Delta x^2}{\nu} \tag{2.9}$$

The first is the time step given by standard convective CFL condition and the second comes from the viscous term, U is the maximum of velocity. We have that if the cell Reynolds number

$$R_c = \frac{U\Delta x}{\nu} \geq 4 \tag{2.10}$$

then

$$\Delta t_1 \leq \Delta t_2 \tag{2.11}$$

i.e. the stability constraint coming from the viscous term is less severe than the one from the convective term. Remember that we do want to obey the convective stability constraint since we are interested in accurately following the dynamics of the flow.

Another cell Reynolds number restriction comes from the following consideration. It is well-known that if we use second order centered difference in space and forward Euler in time for the simple advection equation

$$u_t + au_x = 0 \tag{2.12}$$

the resulting scheme is unconditionally unstable. This has the consequence that for the advection-diffusion equation

$$u_t + au_x = \nu u_{xx} \tag{2.13}$$

this scheme is stable only under the constraint

$$\Delta t\left(\frac{a^2}{2\nu} + \frac{2\nu}{\Delta x^2}\right) < 1 \tag{2.14}$$

Therefore we must have

$$\nu \frac{\Delta t}{\Delta x^2} < \frac{1}{2}, \quad \text{and} \quad \Delta t < \frac{2\nu}{a^2} \tag{2.15}$$

The first condition in (2.15) is the standard diffusive constraint on time steps. The second one reflects the fact that the scheme is unstable if $\nu = 0$.

From a slightly different point of view, if we demand the second condition in (2.15) to be less restrictive than the standard diffusive condition, we should take

$$\frac{2\nu}{a^2} > \frac{\Delta x^2}{2\nu}, \quad \text{i.e.} \quad Rc = \frac{a\Delta x}{\nu} < 2 \tag{2.16}$$

This cell Reynolds number constraint has often been used as an argument against using centered differencing for the convection term at high Reynolds number.

We remark that this cell Reynolds number problem is special to incompressible flows. For viscous compressible flows, the thickness of the shock layers is on the order of Re^{-1}. Therefore the cell Reynolds number is limisted by accuracy considerations. In other words, resolving viscous shock layer is intrinsically difficult. However for incompressible flows, the thickness of the shear layers are on the order of $Re^{-1/2}$ for 2D and $Re^{-3/4}$ for 3D. Therefore in principle at high enough Reynolds number, accuracy considerations should not present a constraint for the cell Reynolds number. Any cell Reynolds number constraint can only be a consequence of the stiffness of the method.

At a first sight, (2.11) and (2.16) contradict each other and the method seems doomed. However, what causes the cell Reynolds number constraint (2.13) is the fact that the stability region of the forward Euler method does not contain any part of the imaginary axis. Same is true for the standard two-stage second order explicit Runge-Kutta methods, but not for 3rd and 4th order ones. The Fourier symbol for the centered difference operator $-a\tilde{D}_x + \nu D_x^2$ is $C(\xi) = ia\frac{\sin \xi}{\Delta x} - \frac{4\nu}{\Delta x^2}\sin^2\frac{\xi}{2}$. Therefore if we use 4th order Runge-Kutta in time, the two stability conditions are

$$\frac{a\Delta t}{\Delta x} \le C_1, \quad 4\nu \frac{\Delta t}{\Delta x^2} \le C_2 \tag{2.17}$$

where C_1 and C_2 are some constants (for example, they can be taken as 1.5). *There is no cell Reynolds number constraint imposed by stability considerations.* Same conclusions can be drawn for the 3rd order Runge-Kutta method. Indeed in the calculations presented in [4, 5], the cell Reynolds number was as high as 10^2 and even 10^3.

Remark 1: Global vorticity boundary condition. Strictly speaking, incompressible flows are nonlocal and vorticity boundary conditions cannot be truly local. This is one reason why global vorticity boundary conditions are preferred in the literature. When we refer to local boundary conditions, it is a purely a numerical object. We call a vorticity boundary condition local if boundary points are not coupled directly. In other words, one does not have to solve a coupled system in order to evaluate the boundary values of vorticity.

It is shown in [4] that Anderson's global vorticity boundary conditions [1] can always be written as local ones if interpreted slightly differently. Global vorticity boundary conditions discussed in [12] are necessary only because the viscous term is treatly *implicitly*. These methods suffer from extreme complexity, both in implementation and in execution. This is particularly true in 3D. As we discussed earlier, if the cell Reynolds number is not too small, this complexity can be avoided entirely by treating the viscous term explicitly.

Remark 2: Low cell Reynolds number calculation. In situations when the cell Reynolds number has to be small, such as nearly steady or Stokes flows, global vorticity boundary conditions have to be used to avoid stiffness. From a computational point of view, this situation is more difficult than the case when the flow is very dynamic although the physical flow may very well be simpler. Better fast solvers will have to be developed to implement the global vorticity boundary conditions.

Remark 3: Connection with MAC scheme. MAC scheme is the prototype in the primitive variable formulation. In 2D it is shown in [4] that they are equivalent to the vorticity methods (2.6–2.8): There is a very natural way of defining discrete vorticity and streamfunction at the MAC grid points which will yield the same values as the ones computed using (2.6–2.8). This gives an alternative way of understanding the numerical boundary conditions. For example, under this, Thom's formula is translated to the reflection boundary condition used in MAC scheme; Orszag-Israeli's formula is translated to the formula of Peyret and Taylor [11].

3   Finite difference methods on 3D non-staggered grids

It is well-known that unlike the primitive variable formulation, the various vorticity formulations differ drastically between 2D and 3D. This has always been regarded as a disadvantage for the vorticity-based numerical methods particularly since the 3D situation is often far more complicated than 2D.

Consider the incompressible Navier-Stokes equation in vorticity form:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \times (\boldsymbol{\omega} \times i\mathbf{u}) = \nu \Delta \boldsymbol{\omega},$$
$$\boldsymbol{\omega} = \nabla \times \mathbf{u}, \quad \nabla \cdot \mathbf{u} = 0,$$
$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}).$$

$$(3.1)$$

We introduce the vector potential $\boldsymbol{\psi}$ such that

$$\mathbf{u} = \nabla \times \boldsymbol{\psi}.$$

Unlike streamfunction in 2D, vector potential in 3D is far from being uniquely defined: If $\boldsymbol{\psi}$ satisfies the above relation and χ is a smooth scalar function, so does $\boldsymbol{\psi} + \nabla \chi$. This is a key difference between 2D and 3D.

Since $\boldsymbol{\omega} = \nabla \times \mathbf{u}$, we have

$$\boldsymbol{\omega} = \nabla \times (\nabla \times \boldsymbol{\psi}) = \nabla(\nabla \cdot \boldsymbol{\psi}) - \Delta \boldsymbol{\psi}.$$

$$(3.2)$$

From a computational point of view, it is desirable to decouple the three components of (3.2). This essentially forces us to choose the gauge of $\boldsymbol{\psi}$ such that

$$\nabla \cdot \boldsymbol{\psi} = 0.$$

$$(3.3)$$

This does not seem to be a progress since now (3.3) couples the three components of $\boldsymbol{\psi}$ together. However if the numerical method guarantees

$$\nabla \cdot \boldsymbol{\omega} = 0$$

then it is enough to have

$$\nabla \cdot \boldsymbol{\psi} \Big|_{\partial \Omega} = 0.$$

in order to ensure (3.3).

This is the new difficulty that arises in 3D for designing efficient numerical methods in any $\boldsymbol{\omega}$–$\boldsymbol{\psi}$ formulations,namely the enforcement of the three divergence-free conditions:

$$\nabla \cdot \boldsymbol{\psi} = 0, \quad \nabla \cdot \mathbf{u} = 0, \quad \nabla \cdot \boldsymbol{\omega} = 0 \tag{3.4}$$

New vorticity-vector potential formulation. We review the vorticity-vector potential formulation presented in [6] which is particularly suited to our purpose of constructing efficient finite difference schemes. For simplicity of presentation we will take the computational domain to be $\Omega = \{x_3 > 0\}$ with $\Gamma = \partial \Omega = \{x_3 = 0\}$. Extension to more general domains is quite straightforward. We impose the boundary condition:

$$u_3(x_1, x_2, 0) = 0, \quad u_1(x_1, x_2, 0) = u_{1b}(x_1, x_2), \quad u_2(x_1, x_2, 0) = u_{2b}(x_1, x_2), \tag{3.5}$$

or in shorthand,

$$\mathbf{u}|_\Gamma = \mathbf{u}_b, \quad \mathbf{u}_b \cdot \mathbf{n} = 0,$$

where $\mathbf{n} = (0, 0, 1)$. Again we define the vector potential $\boldsymbol{\psi}$ by:

$$\mathbf{u} = \nabla \times \boldsymbol{\psi}. \tag{3.6}$$

Proposition: (3.1)–(3.6) is equivalent to

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \nabla \times [\boldsymbol{\omega} \times (\nabla \times \boldsymbol{\psi})] = \nu \Delta \boldsymbol{\omega}, \tag{3.7}$$
$$-\Delta \boldsymbol{\psi} = \boldsymbol{\omega},$$

with boundary conditions

$$\psi_1 = \psi_2 = 0, \quad \frac{\partial \psi_3}{\partial x_3} = 0, \quad \text{on } \Gamma$$

$$\omega_3 = \frac{\partial u_{2b}}{\partial x_1} - \frac{\partial u_{1b}}{\partial x_2}, \quad \frac{\partial \psi_3}{\partial x_2} - \frac{\partial \psi_2}{\partial x_3} = u_{1b}, \quad \frac{\partial \psi_1}{\partial x_3} - \frac{\partial \psi_3}{\partial x_1} = u_{2b}, \quad \text{on } \Gamma \tag{3.8}$$

and initial condition

$$\boldsymbol{\omega}_0(\mathbf{x}) = \nabla \times \mathbf{u}_0(\mathbf{x}). \tag{3.9}$$

The proof of this result is given in [6].

For more general situations the boundary condition (3.7) should be replaced by a Dirichlet boundary condition for the tangential components of $\boldsymbol{\psi}$ and Neumann boundary condition for the normal component of $\boldsymbol{\psi}$. Similarly (3.8) should be replaced by the normal component of the $\boldsymbol{\omega}$–$\mathbf{u}$ relation and the tangential components of the $\boldsymbol{\psi}$–$\mathbf{u}$ relation.

Efficient difference schemes and local vorticity boundary condition in 3D. The numerical grid is defined by $\Omega_h = \{(x_i, y_j, z_k), x_i = i\Delta x, y_j = j\Delta y, z_k = k\Delta z, i, j \in Z, k \in Z^+\}$. For boundary

conditions, we need to define the "ghost" points: $\{(x_i, y_j, z_{-1}), i, j \in Z, z_{-1} = -\Delta z\}$. We will use the difference operators:

$$\widetilde{D}_x g(x, y, z) = \frac{g(x + \Delta x, y, z) - g(x - \Delta x, y, z)}{2\Delta x}$$

$$D_x^2 g(x, y, z) = \frac{g(x + \Delta x, y, z) - 2g(x, y, z) + g(x - \Delta x, y, z)}{\Delta x^2} \, .$$

Similarly we define $\widetilde{D}_y g$, $D_y^2 g$, etc.

We first describe the semi-discrete version of the method. In the interior of Ω_h (i.e., $k > 0$), we use

$$\begin{aligned}
\frac{\partial \boldsymbol{\omega}}{\partial t} + \tilde{\nabla}_h \times [\boldsymbol{\omega} \times (\nabla \times \boldsymbol{\psi})] &= \nu \Delta_h \boldsymbol{\omega} \, , \\
-\Delta_h \boldsymbol{\psi} &= \boldsymbol{\omega} \, , \\
\mathbf{u} &= \tilde{\nabla}_h \times \boldsymbol{\psi} \, .
\end{aligned} \tag{3.10}$$

The difference operators appeared here are defined by

$$\Delta_h = D_x^2 + D_y^2 + D_z^2 \, , \quad \tilde{\nabla}_h = (\widetilde{D}_x, \widetilde{D}_y, \widetilde{D}_z) \, .$$

The first set of boundary conditions, to be supplemented to the Poisson equations for $\boldsymbol{\psi}$, is

$$\psi_1(x_i, y_j, 0) = \psi_2(x_i, y_j, 0) = 0 \, , \quad \frac{\psi_3(x_i, y_j, z_1) - \psi_3(x_i, y_j, z_{-1})}{2\Delta z} = 0 \, , \quad i, j, \in Z \, . \tag{3.11}$$

Given $\{\boldsymbol{\omega}(x_i, y_j, z_k), i, j \in Z, k \geq 1\}$, we can solve $-\Delta \boldsymbol{\psi} = \boldsymbol{\omega}$ together with (3.11) to obtain $\{\boldsymbol{\psi}(x_i, y_j, z_k), i, j \in Z, k \geq 0\}$. In order to update $\boldsymbol{\omega}$, we need its numerical boundary conditions. These are to be obtained from the velocity boundary condition.

First of all $\omega_3|_\Gamma$ can be readily evaluated:

$$\omega_3(x_1, x_2, 0) = \left(\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2}\right)(x_1, x_2, 0) = \left(\frac{\partial u_{2b}}{\partial x_1} - \frac{\partial u_{1b}}{\partial x_2}\right)(x_1, x_2) \, . \tag{3.12}$$

Next we notice that (3.11) implies already that

$$u_3 = \frac{\partial \psi_2}{\partial x_1} - \frac{\partial \psi_1}{\partial x_2} = 0 \quad \text{on } \Gamma \, ,$$

so only two velocity boundary conditions are left (the two tangential ones):

$$\frac{\partial \psi_3}{\partial x_2} - \frac{\partial \psi_2}{\partial x_3} = u_{1b} \, , \quad \frac{\partial \psi_1}{\partial x_3} - \frac{\partial \psi_3}{\partial x_1} = u_{2b} \tag{3.13}$$

They should give the boundary condition for the tangential components of $\boldsymbol{\omega}$.

Approximate (3.13) by

$$(\widetilde{D}_{x_2} \psi_3)(x_i, y_j, z_0) - \frac{\psi_2(x_i, y_j, z_1) - \psi_2(x_i, y_j, z_{-1})}{2\Delta z} = u_{1b}(x_i, y_j) \tag{3.14}$$

and

$$\frac{\psi_1(x_i, y_j, z_1) - \psi_1(x_i, y_j, z_{-1})}{2\Delta z} - (\widetilde{D}_{x_1} \psi_3)(x_i, y_j, z_0) = u_{2b}(x_i, y_j) \tag{3.15}$$

These formulas define $\{\psi_1(x_i, y_j, z_{-1}), i, j \in Z\}$ and $\{\psi_2(x_i, y_j, z_{-1}), i, j \in Z\}$:

$$\psi_2(x_i, y_j, z_{-1}) = \psi_2(x_i, y_j, z_1) - 2\Delta z\big((\widetilde{D}_{x_2}\psi_3)(x_i, y_j, z_0) - u_{1b}(x_i, y_j)\big) \tag{3.16}$$

$$\psi_1(x_i, y_j, z_{-1}) = \psi_1(x_i, y_j, z_1) - 2\Delta z\big((\widetilde{D}_{x_1}\psi_3)(x_i, y_j, z_0) + u_{2b}(x_i, y_j)\big) \tag{3.17}$$

Now we can evaluate ω_1, ω_2 at Γ:

$$
\begin{aligned}
\omega_1(x_i, y_j, z_0) &= -(D_x^2 + D_y^2 + D_z^2)\psi_1(x_i, y_j, z_0) \\
&= -D_z^2\psi_1(x_i, y_j, z_0) = -\frac{\psi_1(x_i, y_j, z_1) + \psi_1(x_i, y_j, z_{-1})}{\Delta z^2} \\
&= -\frac{2}{\Delta z^2}\psi_1(x_i, y_j, z_1) + \frac{2}{\Delta z}\big(\widetilde{D}_{x_1}\psi_3(x_i, y_j, z_0) + u_{2b}(x_i, y_j)\big)
\end{aligned} \tag{3.18}
$$

$$
\begin{aligned}
\omega_2(x_i, y_j, z_0) &= -(D_x^2 + D_y^2 + D_z^2)\psi_2(x_i, y_j, z_0) \\
&= -D_z^2\psi_2(x_i, y_j, z_0) = -\frac{\psi_2(x_i, y_j, z_1) + \psi_2(x_i, y_j, z_{-1})}{\Delta z^2} \\
&= -\frac{2}{\Delta z^2}\psi_2(x_i, y_j, z_1) + \frac{2}{\Delta z}\Big(\widetilde{D}_{x_2}\psi_3(x_i, y_j, z_0) - u_{1b}(x_i, y_j)\Big).
\end{aligned} \tag{3.19}
$$

These are the analog of Thom's formula for 2D problems. This completes the description of the semi-discrete scheme.

For the fully discrete scheme, it is important to treat the viscous terms explicitly. This point was discussed at length in [6].

This is clearly a very efficient method. At every step or Runge-Kutta stage, only three separate scalar Poisson equations are solved. Moreover, standard fast Poisson solvers can be used.

Remark 1: Discretization of the convective terms. It turns out the discretization of the convective term near the boundary becomes much more of an issue in 3D. This is due to the special structure of the error when local vorticity boundary conditions are used. This is discussed carefully in [6].

Remark 2: Importance of the divergence-free condition for vorticity. A variant of this method was introduced in [6] in which the divergence free condition for vorticity was enforced directly through boundary conditions. It was found that this drastically improved the accuracy of the divergence of vorticity, but *not* the overall results. An explanation was given in [6].

4   Essentially compact 4th order schemes (EC4)

In this section we present a very efficient 4th order accurate compact scheme in the vorticity formulation. For simplicity of presentation we will restrict ourselves to 2D. Extension to 3D can be found in [6].

We first discuss the treatment of the Stokes part, treating the nonlinear convection terms as if they were some known forcing functions. We will use the notation: $f = \partial_x(u\omega) + \partial_y(v\omega)$ and write the vorticity transport equation as

$$\frac{\partial \omega}{\partial t} = \nu \triangle \omega - f \tag{4.1}$$

The most obvious and well-known compact discretization of (4.1) is

$$\left(1 + \frac{h^2}{12}\triangle_h\right)\frac{\partial\omega}{\partial t} = \nu\left(\triangle_h + \frac{h^2}{6}D_x^2 D_y^2\right)\omega - \left(1 + \frac{h^2}{12}\triangle_h\right)f. \tag{4.2}$$

We have assumed, without loss of generality, that $\triangle x = \triangle y = h$.

(4.2) is an $O(h^4)$ approximation to (4.1). Similarly we discretize the kinematic equation relating vorticity to streamfunction ψ using

$$\left(\triangle_h + \frac{h^2}{6}D_x^2 D_y^2\right)\psi = \left(1 + \frac{h^2}{12}\triangle_h\right)\omega \tag{4.3}$$

So far it has been completely standard. We now come to the boundary conditions. Assume that the mesh is as depicted in Fig. 1. The no normal flow condition translates to

$$\psi|_\Gamma = 0$$

In order to obtain the numerical values at the "ghost points" outside the physical domain Ω, we use the no-slip condition $\frac{\partial\psi}{\partial\mathbf{n}}|_\Gamma = 0$ twice at the physical boundary $\Gamma = \Gamma_x \cup \Gamma_y$: once using a 4th order one-sided approximation, once using a 4th order centered approximation. For example, at the boundary Γ_y we have

$$\left(\frac{\partial\psi}{\partial x}\right)_{0,j} \approx \frac{-3\psi_{-1,j} - 10\psi_{0,j} + 18\psi_{1,j} - 6\psi_{2,j} + \psi_{3,j}}{12h} \tag{4.4}$$

$$\left(\frac{\partial\psi}{\partial x}\right)_{0,j} \approx \frac{\psi_{-2,j} - 8\psi_{-1,j} + 8\psi_{1,j} - \psi_{2,j}}{12h} \tag{4.5}$$

Consequently, we have

$$\psi_{-1,j} = 6\psi_{1,j} - 2\psi_{2,j} + \frac{1}{3}\psi_{3,j} - 4h\left(\frac{\partial\psi}{\partial x}\right)_{0,j} \tag{4.6}$$

$$\psi_{-2,j} = 40\psi_{1,j} - 15\psi_{2,j} + \frac{8}{3}\psi_{3,j} - 20h\left(\frac{\partial\psi}{\partial x}\right)_{0,j} \tag{4.7}$$

To evaluate the vorticity at the boundary, we use a 4th order approximation of $\omega = \triangle\psi = \psi_{xx}$:

$$\omega_{0,j} = \frac{1}{12h^2}\left(16(\psi_{-1,j} + \psi_{1,j}) - (\psi_{-2,j} + \psi_{2,j})\right) \tag{4.8}$$

Using (4.6-4.7), we obtain

$$\omega_{0,j} = \frac{108\psi_{1,j} - 27\psi_{2,j} + 4\psi_{3,j}}{18h^2} - \frac{11}{3h}\left(\frac{\partial\psi}{\partial x}\right)_{0,j} \tag{4.9}$$

This is known as Briley's formula [2].

Now the velocity $\mathbf{u} = (u, v)$ can be evaluated readily at all interior grid points by using the standard 4th order accurate formulas:

$$u = -\widetilde{D}_y\left(1 - \frac{h^2}{6}D_y^2\right)\psi; \quad v = \widetilde{D}_x\left(1 - \frac{h^2}{6}D_x^2\right)\psi, \tag{4.10}$$

or some standard compact difference operators. On the boundary Γ, we use naturally

$$u = 0, \quad v = 0. \tag{4.11}$$

To treat the convection terms, we note that

$$\left(1 + \frac{h^2}{12}\triangle_h\right)\widetilde{D}_x\left(1 - \frac{h^2}{6}D_x^2\right) = \widetilde{D}_x\left(1 + \frac{h^2}{6}D_y^2\right) - \frac{h^2}{12}\triangle_h\widetilde{D}_x + O(h^4). \tag{4.12}$$

Hence we can approximate $\left(1 + \frac{h^2}{12}\triangle_h\right)f$ to 4th order using

$$\begin{aligned}
\left(1+\frac{h^2}{12}\triangle_h\right)&f \\
&= \left(1 + \frac{h^2}{12}\triangle_h\right)\left\{\widetilde{D}_x\left(1 - \frac{h^2}{6}D_x^2\right)(u\omega) + \widetilde{D}_y\left(1 - \frac{h^2}{6}D_y^2\right)(v\omega)\right\} + O(h^4) \\
&= \widetilde{D}_x\left(1 + \frac{h^2}{6}D_y^2\right)(u\omega) + \widetilde{D}_y\left(1 + \frac{h^2}{6}D_x^2\right)(v\omega) - \frac{h^2}{12}\triangle_h\left(\widetilde{D}_x(u\omega) + \widetilde{D}_y(v\omega)\right) + O(h^4) \\
&= \widetilde{D}_x\left(1 + \frac{h^2}{6}D_y^2\right)(u\omega) + \widetilde{D}_y\left(1 + \frac{h^2}{6}D_x^2\right)(v\omega) - \frac{h^2}{12}\triangle_h\left(u\widetilde{D}_x\omega + v\widetilde{D}_y\omega\right) + O(h^4)
\end{aligned} \tag{4.13}$$

The operator appearing in the first term of the right hand side of (4.13) has a 9-point compact stencil, and so does the operator in the second term. The third term, however, is not compact. Nevertheless this does not present any problem computationally for two reasons: In the interior of the domain (for $i, j \geq 2$), this term can be evaluated very efficiently since the convection terms will be treated explicitly in the fully discrete scheme. Near the boundary (i, or $j = 1$), we need the boundary value of $u\widetilde{D}_x\omega + v\widetilde{D}_y\omega$ on Γ. In the present case, we can set

$$u\widetilde{D}_x\omega + v\widetilde{D}_y\omega|_\Gamma = 0 \tag{4.14}$$

and still preserve 4th order accuracy. More general situations, including the treatment of in- and out-flow boundary conditions, are discussed in [5].

Finally, we discuss the temporal discretization procedure. Again we will use forward Euler as an illustration. The computations reported in [5] were done using the classical 4th order Runge-Kutta.

Initialization: Given $\{\omega_{ij}^0\}$, compute

$$\left(1 + \frac{h^2}{12}\triangle_h\right)\omega^0 = \bar{\omega}^0$$

Time-stepping: Given $\{\omega_{ij}^n\}$, we compute $\{\omega_{ij}^{n+1}\}$ via the following steps.

Step 1. Update $\{\bar{\omega}_{i,j}^{n+1}\}_{i\geq 1, j\geq 1}$ using

$$\begin{aligned}
\frac{\bar{\omega}^{n+1} - \bar{\omega}^n}{\Delta t} &+ \widetilde{D}_x\left(1 + \frac{h^2}{6}D_y^2\right)(u^n\omega^n) + \widetilde{D}_y\left(1 + \frac{h^2}{6}D_x^2\right)(v^n\omega^n) \\
&- \frac{h^2}{12}\triangle_h(u^n\widetilde{D}_x\omega^n + v^n\widetilde{D}_y\omega^n) = \nu\left(\triangle_h + \frac{h^2}{6}D_x^2D_y^2\right)\omega^n \quad (4.15)
\end{aligned}$$

Step 2. Solve for $\left\{\psi_{i,j}^{n+1}\right\}_{i\geq 1, j\geq 1}$ using

$$\left(\Delta_h + \frac{h^2}{6}D_x^2 D_y^2\right)\psi^{n+1} = \bar{\omega}^{n+1} \tag{4.16}$$

with the boundary condition

$$\psi^{n+1}|_\Gamma = 0.$$

Compute ψ^{n+1} at the "ghost points" using (4.6).

Step 3. Solve for $\left\{\omega_{i,j}^{n+1}\right\}_{i\geq 1, j\geq 1}$ using

$$\left(1 + \frac{h^2}{12}\Delta_h\right)\omega^{n+1} = \bar{\omega}^{n+1} \tag{4.17}$$

with the boundary condition (4.9).

Step 4. Update the velocity using

$$u^{n+1} = -\widetilde{D}_y\left(1 - \frac{h^2}{6}D_y^2\right)\psi^{n+1}, \quad v^{n+1} = \widetilde{D}_x\left(1 - \frac{h^2}{6}D_x^2\right)\psi^{n+1} \tag{4.18}$$

or some compact 4th order differencing [3] for $i, j \geq 1$, and $u^{n+1}|_\Gamma = 0$, $v^{n+1}|_\Gamma = 0$.

The efficiency of this method is obvious. Only two Poisson-like equations, namely (4.16) and (4.17), are solved at each step or each stage of the Runge-Kutta method. Both can be solved using the standard fast Poisson solvers developed for the *second* order schemes. We have seen consistently that this scheme costs slightly more than twice the cost of the second order scheme for each time step.

This scheme is named essentially compact (EC4 for short) since the only non-compact part occurs in the treatment of the convection terms, and the non-compact differencing does not increase the need for numerically supplemented boundary conditions. So the most attractive features of the compact schemes are retained. Numerical results presented in [5] demonstrate that this scheme is substantially better than the second order scheme discussed in [4] in the ability to resolve boundary layers, small scales and even gross features.

Proof of 4th order convergence. For any $\alpha > 0$, let $\psi_e \in L^\infty\left([0,T]; C^{6,\alpha}(\bar{\Omega})\right)$ be the solution of the Navier-Stokes equation (1.1)-(2.1) and $\mathbf{u}_h$ be the approximate solution of EC4, then we have

$$\sup_{0\leq t\leq T} \|\mathbf{u}(\cdot, t) - \mathbf{u}_h(\cdot, t)\|_{L^2} \leq Ch^4 \sup_{0\leq t\leq T} \|\psi_e(\cdot, t)\|_{C^{6,\alpha}(\bar{\Omega})}^2$$

The proof of this result for very smooth solutions (i.e. right hand side replaced by higher norms) can be found in [5]. The proof under weaker regularity assumptions as stated above can be obtained directly from the authors.

5 Numerical results

We present some numerical results using the second order schemes presented here for the standard test problem: the driven cavity flow. We refer to [5] for numerical results on EC4. The set-up of the problem is standard: velocities vanish everywhere at the boundary except at the upper boundary

where the x-component of the velocity is 1 for 3D problem and $16x^2(1-x)^2$ for 2D. The initial data is take as zeros (impulsive start) for 3D and $\phi_0(x,y) = 16x^2(1-x)^2(y^2-y^3)$ for 2D [4, 6].

The set of results are reported, one for 2D and one for 3D. Figure 2 is the contour plot of vorticity in 2D at Reynolds number 10^5, $t = 5$. Figure 3 is the contour plot of stream-wise vorticity at the center plane $x = 1/2$. Here Reynolds number is 3200, t=35. In [6], we used two other different methods to compute this problem and obtained very similar results.

6   Conclusions

We discussed three basic issues for finite difference methods in vorticity formulation using local vorticity boundary condition: what makes them work? how to make them work in 3D and how to make simple 4th order schemes. These discussions are not limited to finite difference schemes. They apply equally well to finite element and finite volume methods which are more suited for complex geometries. We will leave these extensions to future publications.

References

[1] Anderson, C.R., "Vorticity Boundary Conditions and Boundary Vorticity Generation for Two-Dimensional Viscous Incompressible Flows," *J. Comp. Phys.*, **80**, pp. 72–97, 1989.

[2] Briley, W.R., "A Numerical Study of Laminar Separation Bubbles using the Navier-Stokes Equations," *J. Fluid Mech.*, **47**, pp. 713–736, 1971.

[3] Carpenter, M.H., Gottlieb, D. and Abarbanel, S., "Time-Stable Boundary Conditions for Finite-Difference Schemes Solving Hyperbolic Systems: Methodology and Application to High-Order Compact Schemes," *J. Comp. Phys.*, **111**, pp. 220–236, 1994.

[4] E, W. and Liu, J.-G., "Vorticity Boundary Condition and Related Issues for Finite Difference Schemes," *J. Comp. Phys.*, **124**, pp. 368–382, 1996.

[5] E, W. and Liu, J.-G., "Essentially Compact Schemes for Unsteady Viscous Incompressible Flows," *J. Comp. Phys.*, **126**, pp. 122–138, 1996.

[6] E, W. and Liu, J.-G., "Finite Difference Methods in Vorticity-Vector Potential Formulation on 3D Non-Staggered Grids," submitted to J. Comp. Phys., 1996.

[7] Gresho, P.M., "Incompressible Fluid Dynamics: Some Fundamental Formulation Issues," *Ann. Rev. Fluid Mech.*, **23**, pp. 413–453, 1991.

[8] Gresho, P.M., "Some Interesting Issues in Incompressible Fluid Dynamics, both in the Continuum and in Numerical Simulation," *Adv. Appl. Mech.*, **28**, pp. 45–140, 1992.

[9] Kleiser, L. and Schumann, U., "Treatment of Incompressibility and Boundary Conditions in 3-D Numerical Spectral Simulations of Plane Channel Flows," *Proc. 3rd GAMM Conf. Num. Meth. in Fluid Mech.*, edited by E.H. Hirschel, Vieweg, Braunschweug, pp. 165–173.

[10] Orszag, S.A. and Israeli, M., "Numerical Simulation of Viscous Incompressible Flows," *Ann. Rev. Fluid Mech.*, **6**, pp. 281–318, 1974.

[11] Peyret, R. and Taylor, T., Computational Methods for Fluid Flow, Springer-Verlag, 1983.

[12] Quartapelle, L., Numerical Solution of the Incompressible Navier-Stokes Equations, Birkhäuser, Berlin, 1993.

[13] Thom, A., "The Flow past Circular Cylinders at Low Speeds," *Proc. Roy. Soc. Lond. A*, **141**, pp. 651–669, 1933.
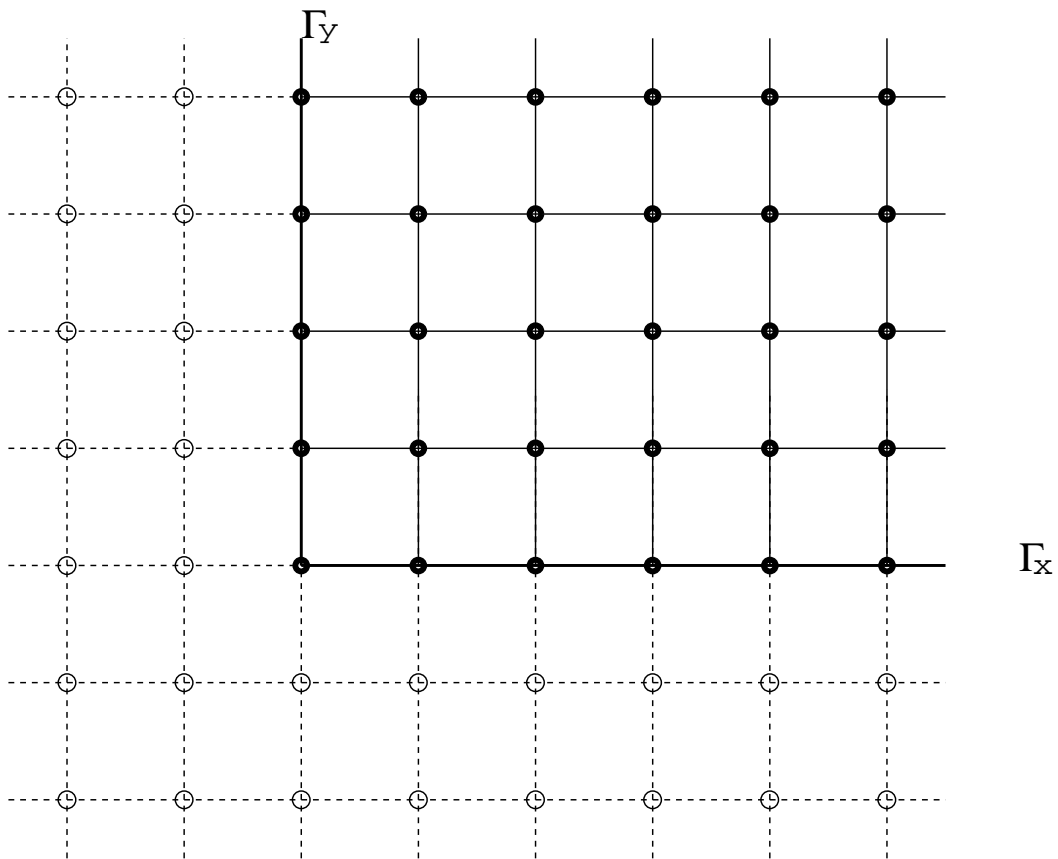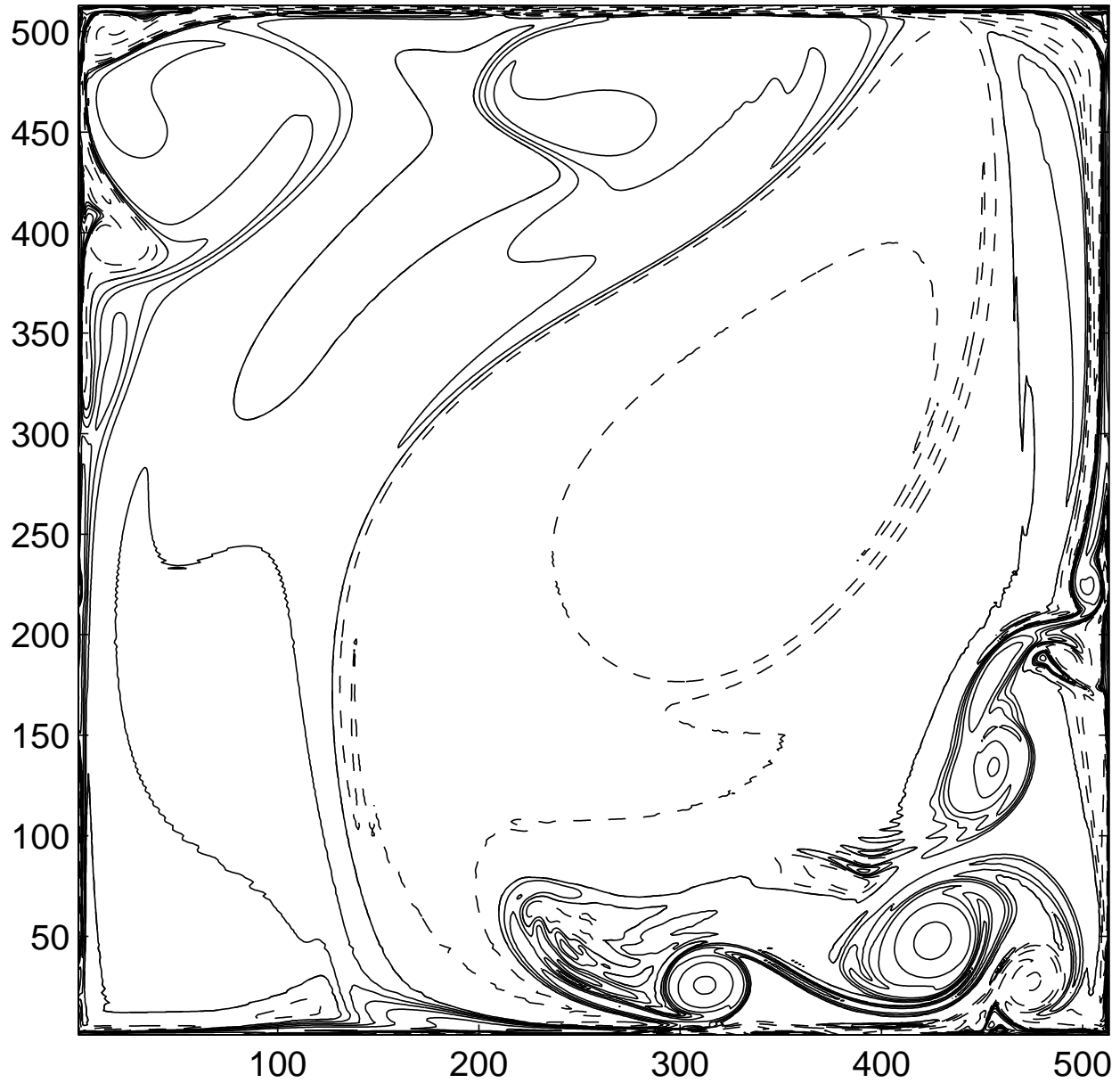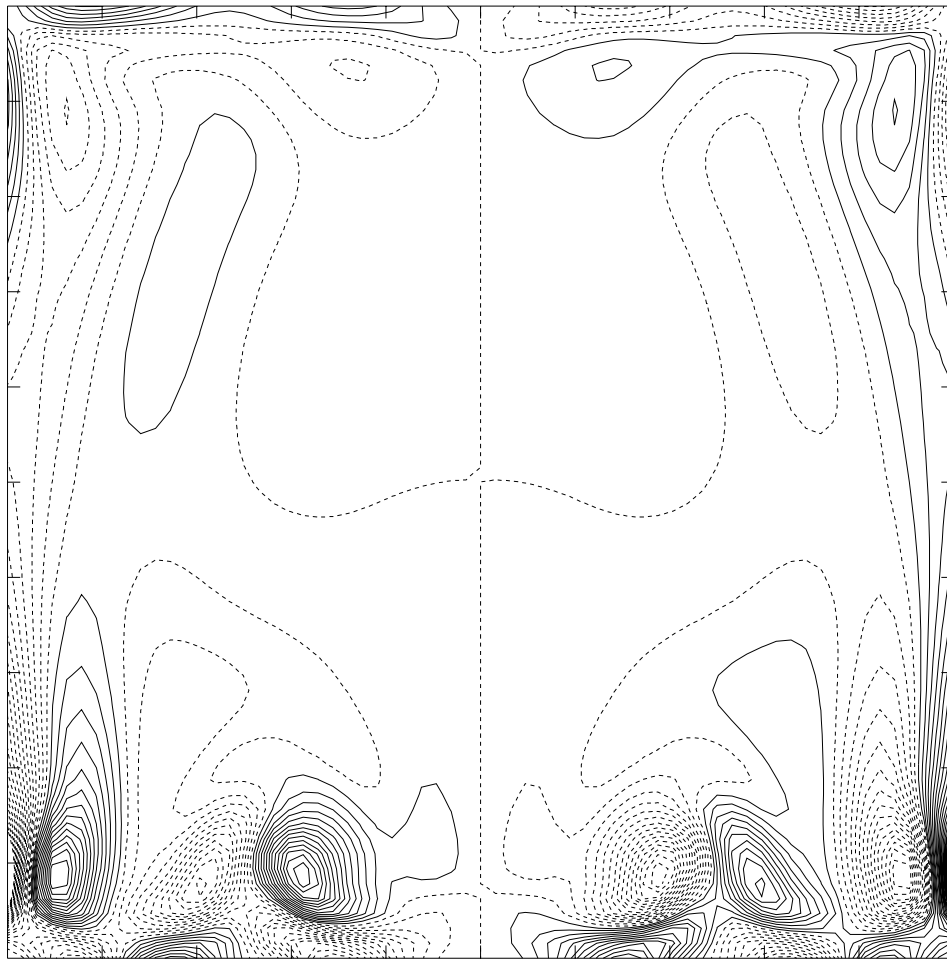
Figure 1: "Ghost" points are indicated with "o".

Figure 2: Contour plot of vorticity at Reynolds number 10^5 computed using second order scheme. Other parameters: $n = 512$, $t = 5$.

CONTOUR FROM -5.0000   TO   4.7500   CONTOUR INTERVAL OF 0.25000   PT(3,3)= -0.74933

Figure 3: Contour plot of ω_1 at the center plane $\{x = \frac{1}{2}\}$ computed using second order 3D scheme. Parameters: $Re = 3200, t = 15, CFL = \frac{\triangle t}{\triangle x} = 2.5, \triangle x = \frac{1}{64}$.