

Numerical solution of Navier stokes in 2D and turbulence modelling

Submitted by : Shubham P. Raghuvanshi
Guided by : Prof. Pinaki Majumdar

July 13, 2021

1 Particle and field approach

- There are two main approaches to fluid dynamics namely the particle approach or the Lagrangian description and field approach or the Eulerian description. In the Lagrangian description the flow is described by following a very large number of interacting fluid particles as they move in $2N$ dimensional phase space and the overall flow is modelled by the phase space density. The governing equation for time evolution of this phase space density is Navier Stokes equation for incompressible flow. This is derived in appendix A.2.

In the Eulerian description, a flow field is defined over the whole domain of analysis, whose properties over space domain and time describe the fluid flow. The governing equations are the Navier Stokes equations for fluid's velocity and density field. In our analysis we will take the latter approach due to it's easier computer implementation.

- To this end we describe the fluid by a set of fields permeating over the domain of analysis. These fields at least include the mass density field $\rho(\vec{x}, t)$, velocity field $\vec{u}(\vec{x}, t)$, hydrodynamic pressure field $P(\vec{x}, t)$, and vorticity $\omega(\vec{x}, t)$. However if we want to study the heat transfer or transfer of any other fluid property across with the fluid then we can include their corresponding fields. If the fluid is to be studied under the effect of any external force then we can include an external force field.

2 Navier stokes equations for incompressible viscous fluid flow

The governing equations for an incompressible viscous flow are conservation of mass and conservation of momentum as it applies to the density and velocity field of the fluid.

2.1 Conservation of mass

We define material volume $V_m(t)$ as the volume occupied by a specific collection of fluid particles, such a volume moves and deforms within a fluid flow such a way that it always contains the same mass elements, none enter the volume and none leave it, like the volume enclosed by surface of a balloon. Therefore by definition

$$\frac{d}{dt} \int_{V_m(t)} \rho(\vec{x}, t) dV = 0 \quad (1)$$

Which implies

$$\int_{V_m(t)} \left(\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{u}) \right) dV = 0 \quad (2)$$

Where \vec{u} is the normal velocity of the boundary enclosing the material volume. Derivation of above equation can be found at appendix A.1. The above equation 2 can be written in a differential form as

$$\frac{\partial \rho}{\partial t} + \nabla(\rho \vec{u}) = 0 \quad (3)$$

For an incompressible fluid the material volume does not deform as during the course of motion, it just moves along with the local velocity field. This means $\nabla \vec{u} = 0$. This gives the mass conservation for an incompressible fluid as

$$\frac{\partial \rho}{\partial t} + \vec{u} \bullet \nabla \rho = 0 \quad (4)$$

2.2 Conservation of momentum

The calculation of appendix A.1 can also be done for any smooth vector field, repeating the same calculation for momentum density field $\rho \vec{u}$ we get the equations in the component form as

$$\frac{\partial(\rho u_i)}{\partial t} + \partial_j(\rho u_j u_i) + \frac{\partial P}{\partial x_i} = 0 \quad (5)$$

which can also be written as

$$\frac{\partial(\rho u_i)}{\partial t} = -\partial_j T_{ij} \quad (6)$$

where $T_{ij} = \rho u_i u_j + \delta_{ij} P$ is the stress energy tensor for ideal fluid and equation(5) can also be thought of as the Newton's equation of motion for ideal fluid. To model a fluid with viscosity we add another term in the stress energy tensor which is $-\nu S_{ij}$ where ν is coefficient of viscosity.

and $S_{ij} = \frac{1}{2} \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$ is the strain rate tensor. For incompressible flow $S_{ii} = 0$ where summation convention is implied. Using this we can write the Navier stokes momentum equations as

$$\frac{\partial u_i}{\partial t} = -u_j \partial_j u_i - \frac{1}{\rho} \partial_i P + \frac{\nu}{\rho} \partial^2 u_i$$

For numerical computational purposes it is preferable to write these equations in their dimensionless form by making following substitutions $t \rightarrow Tt$, $x_i \rightarrow Lx_i$, $u_i \rightarrow Uu_i$, $P \rightarrow P_0P$. After some algebra this gives

$$[St] \frac{\partial u_i}{\partial t} = -u_j \partial_j u_i - [Eu] \partial_i P + \frac{1}{[Re]} \partial^2 u_i$$

Where $[St] = \frac{L}{TU}$ is Strouhal number, $[Eu] = \frac{P_0}{\rho U^2}$ is the Euler number and $[Re] = \frac{\rho LU}{\nu}$ is the Reynold's number. In the natural units of space and time we can take $[St]=1$. Therefore the equations

$$\frac{\partial u_i}{\partial t} = -u_j \partial_j u_i - [Eu] \partial_i P + \frac{1}{[Re]} \partial^2 u_i \quad (7)$$

Along with the incompressibility condition

$$\partial_i u_i = 0 \quad (8)$$

Constitute $D+1$ coupled non linear partial differential equations to be simultaneously solved in order to get the complete solution of a incompressible viscous fluid flow problem. Navier-Stokes momentum equations are a special class of convection diffusion equation with pressure gradient as the source term. In the following section we briefly describe the numerical solution to convection diffusion type problems.

3 Numerical solution of convection diffusion problems

Consider the following 1D convection diffusion IVP

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = D \frac{\partial^2 T}{\partial x^2}, \quad T(x, t=0) = f(x) \quad (9)$$

In what follows now the superscripts denote discrete time index and subscripts denote discrete space index.

- **Explicit FTCS finite difference $O(\Delta t, \Delta x^2)$**

$$T_i^{n+1} = T_i^n - \frac{u \Delta t}{2 \Delta x} (T_{i+1}^n - T_{i-1}^n) + \frac{D \Delta t}{\Delta x^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (10)$$

The values at time step $n+1$ is computed from the values at time step n . The conditions for stability is derived in appendix (A.5). The result is that for every grid point

$$\Delta t \leq \frac{2D}{u^2} \quad (11)$$

$$\Delta x \geq u \Delta t \quad (12)$$

For $D=0$ the method is always unstable. One possible solution to this is given by replacing T_i^n on right hand side by it average on neighbouring grids. Resulting in the Lax Friedrichs scheme.

- **Lax Friedrichs scheme $O(\Delta t, \Delta x^2)$**

$$T_i^{n+1} = \frac{T_{i+1}^n + T_{i-1}^n}{2} - \frac{u \Delta t}{2 \Delta x} (T_{i+1}^n - T_{i-1}^n) + \frac{D \Delta t}{\Delta x^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (13)$$

After some algebraic manipulations we obtain

$$T_i^{n+1} = T_i^n - \frac{u \Delta t}{2 \Delta x} (T_{i+1}^n - T_{i-1}^n) + \frac{\Delta t}{\Delta x^2} \left(D + \frac{\Delta x^2}{2 \Delta t} \right) (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (14)$$

Therefore the problem of instability at $D=0$ is avoided in this scheme by adding **Artificial viscosity** in explicit FTCS.

- **Upwind convection $O(\Delta t, \Delta x)$** Another possible solution that works for $D=0$ is to use backwards difference for advection term. The condition for stability is $\Delta x > u \Delta t$

$$T_i^{n+1} = T_i^n - \frac{u \Delta t}{\Delta x} (T_i^n - T_{i-1}^n) + \frac{D \Delta t}{\Delta x^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (15)$$

- **Explicit RK2 $O(\Delta t^2, \Delta x^2)$**

$$T_i^{n+1} = T_i^n + (k_i^n + k_i^{n+1})/2 \quad (16)$$

$$k_i^n = \frac{u \Delta t}{2 \Delta x} (T_{i+1}^n - T_{i-1}^n) + \frac{D \Delta t}{\Delta x^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (17)$$

Where we substitute for T_i^{n+1} using explicit FTCS.

- **Implicit FTCS finite difference (BTCS) $O(\Delta t, \Delta x^2)$**

$$T_i^{n+1} = T_i^n - \frac{u\Delta t}{2\Delta x} (T_{i+1}^{n+1} - T_{i-1}^{n+1}) + \frac{D\Delta t}{\Delta x^2} (T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) \quad (18)$$

The values at time step $n+1$ is computed from the nearby values at time step $n+1$. Therefore we need to solve a system linear equations. The method is $O(\Delta t, \Delta x^2)$ accurate and unconditionally stable.

$$T_i^{n+1} = T_i^n - \alpha (T_{i+1}^{n+1} - T_{i-1}^{n+1}) + \beta (T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) \quad (19)$$

$$(\alpha - \beta)T_{i+1}^{n+1} + (1 + 2\beta)T_i^{n+1} - (\alpha + \beta)T_{i-1}^{n+1} = T_i^n \quad (20)$$

In matrix form this reads

$$\begin{bmatrix} 1 + 2\beta & \alpha - \beta & 0 & 0 & \dots & -(\alpha + \beta) \\ -(\alpha + \beta) & 1 + 2\beta & \alpha - \beta & 0 & \dots & 0 \\ 0 & -(\alpha + \beta) & 1 + 2\beta & \alpha - \beta & \dots & 0 \\ 0 & 0 & -(\alpha + \beta) & 1 + 2\beta & \alpha - \beta & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \alpha - \beta & \dots & 0 & 0 & -(\alpha + \beta) & 1 + 2\beta \end{bmatrix} \begin{bmatrix} T_0^{n+1} \\ T_1^{n+1} \\ T_2^{n+1} \\ T_3^{n+1} \\ \vdots \\ T_{N-1}^{n+1} \end{bmatrix} = \begin{bmatrix} T_0^n \\ T_1^n \\ T_2^n \\ T_3^n \\ \vdots \\ T_{N-1}^n \end{bmatrix}$$

- **Predictor-Corrector** In these type of methods the initial guess for the solution(Predictor) is provided by any of the Explicit schemes and then using that guess the set of linear algebraic equations in (18) can be solved using any of the iterative methods such Gauss-Seidel or Jacobi iteration.
- **Explicit-Implicit FTCS(BTCS-RK2) $O(\Delta t^2, \Delta x^2)$** Instead of substituting for T_i^{n+1} using explicit FTCS, We directly solve equation(16) iteratively for T_i^{n+1} . After some algebraic manipulations in eq(16) one can get the following expression

$$T_i^{n+1} = T_i^{n+1/2} - \frac{u(\Delta t/2)}{\Delta x} (T_{i+1}^{n+1} - T_{i-1}^{n+1}) + \frac{D(\Delta t/2)}{\Delta x^2} (T_{i+1}^{n+1} - 2T_i^{n+1} + T_{i-1}^{n+1}) \quad (21)$$

where

$$T_i^{n+1/2} = T_i^n - \frac{u(\Delta t/2)}{2\Delta x} (T_{i+1}^n - T_{i-1}^n) + \frac{D(\Delta t/2)}{\Delta x^2} (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (22)$$

Therefore this is equivalent to first doing an explicit FTCS for time step $\Delta t/2$ by eq(22) then using this value to get the final solution using eq(21) using predictor corrector scheme. The method is $O(\Delta t^2, \Delta x^2)$ and unconditionally stable.

As an example consider the one dimensional convection diffusion equation with constant u .

$$\phi_t(x, t) + u\phi_x(x, t) - D\phi_{xx}(x, t) = 0, \phi(x, t=0) = \phi(x) \quad (23)$$

The solution is

$$\phi(x, t) = \int_{-\infty}^{\infty} \Phi(k) e^{-ik(x-ut)-Dk^2t} dk$$

where $\Phi(k) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi(x') e^{ikx'} dx'$

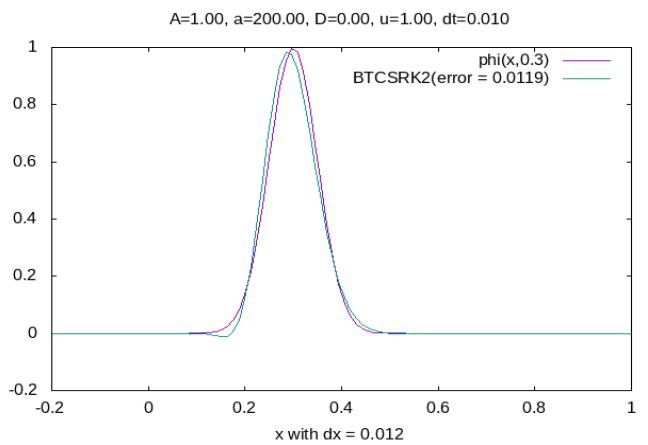
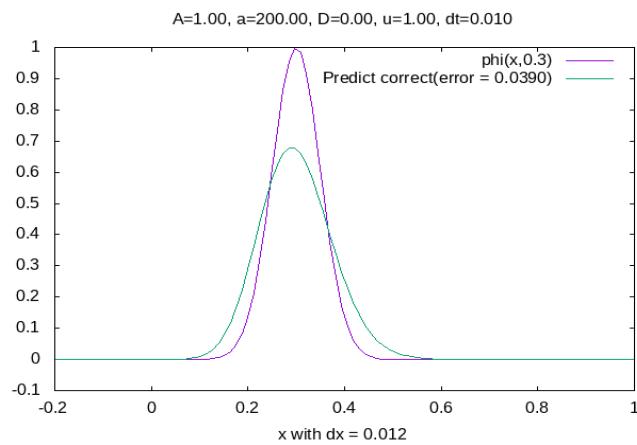
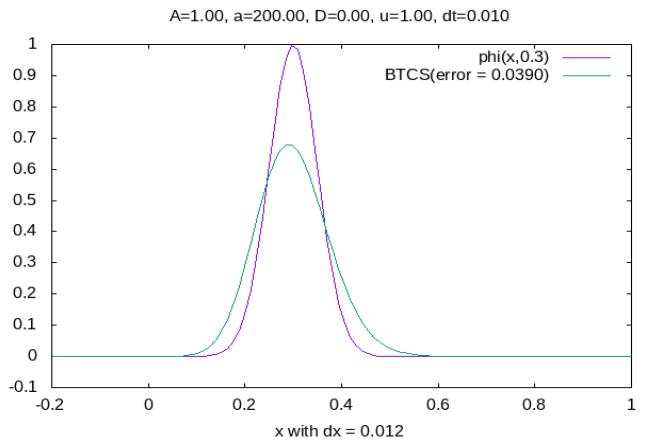
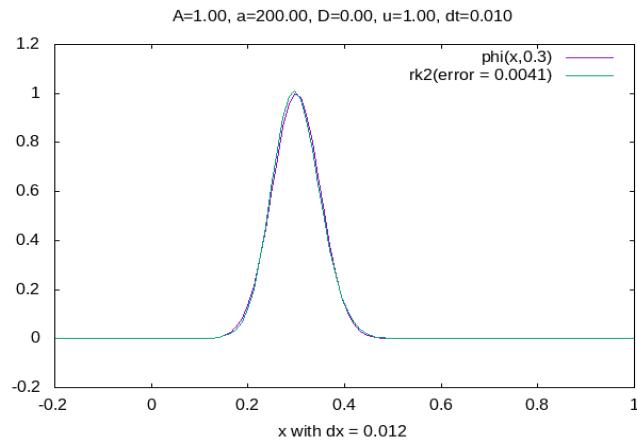
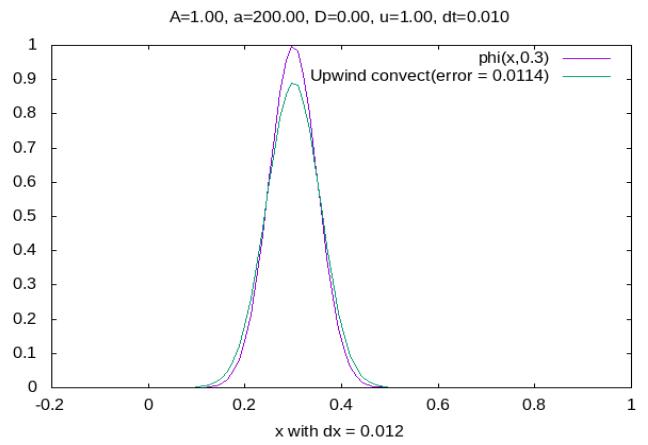
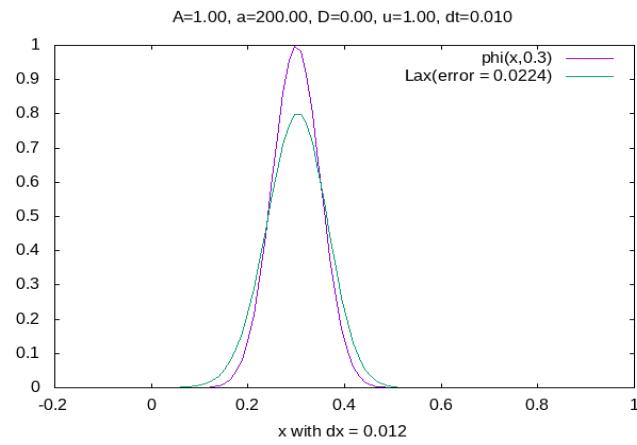
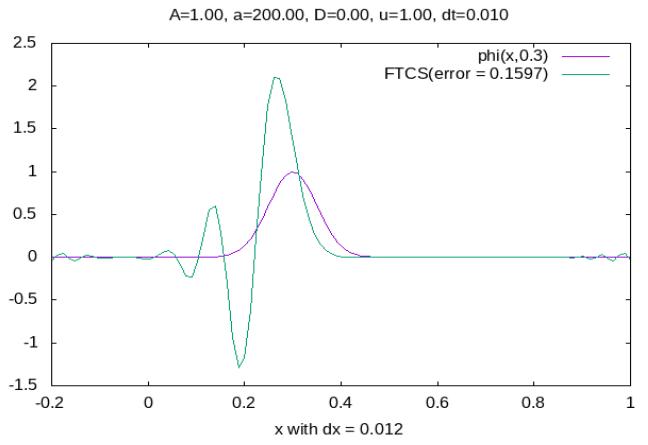
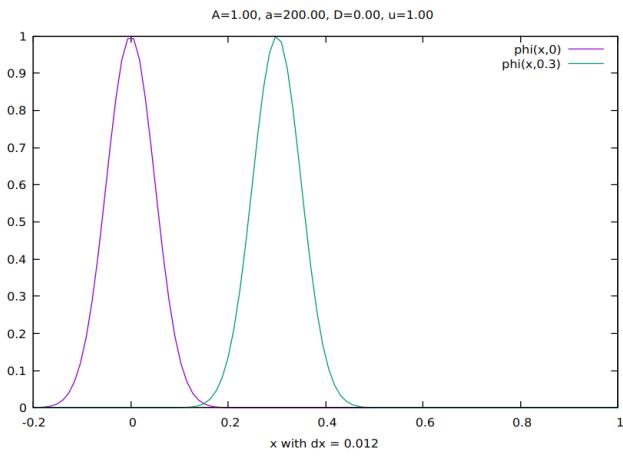
For $\phi(x) = Ae^{-ax^2}$ the solution is

$$f(x, t) = \frac{A}{\sqrt{1+4aDt}} \exp\left(\frac{-a(x-ut)^2}{1+4aDt}\right)$$

Note that for $u = 0$ and $D = \frac{i\hbar}{2m}$ we get the solution for time dependant Schroedinger's equation. Also the impulse response can be obtained by setting $A = \sqrt{a/\pi}$ and letting $a \rightarrow \infty$

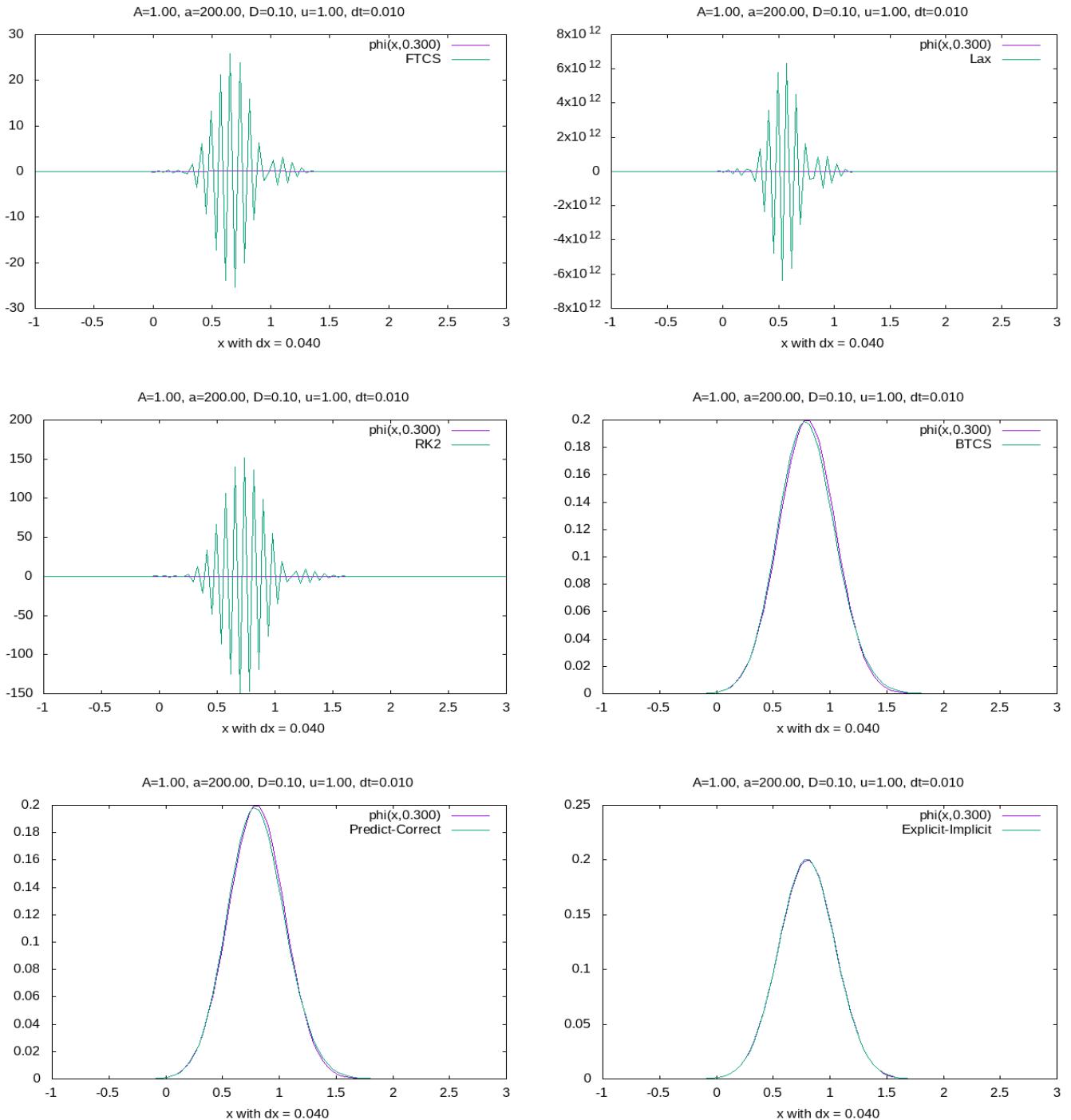
$$f(x, t) = \frac{1}{\sqrt{4\pi Dt}} \exp\left(\frac{-(x-ut)^2}{4Dt}\right)$$

Using our code we are able to reproduce the results in section 3 of the following paper [4], which also attempts to solve the given problem numerically. The results are shown below.



The top left figure shows the actual analytical solution for $t = 0, 0.3$ with $D=0$ and $u=1$, the rest of the figures show the numerical solutions at $t = 0.3$ constructed using $\Delta t = 0.01$ and compare with the actual solution.

- After $t = 0.3$ the oscillations begin to take over in Explicit FCTS give off unstable solution unconditionally. For Small values of D the method is always unstable therefore this cannot be used for high Reynold's number fluid flow.
- Solution from Lax scheme has slightly larger width and dissipation due to the added artificial viscosity term.
- Upwind convection gives stable solution for $D=0$ as long as $\Delta x < u\Delta t$.
- Explicit RK2 has least numerical error, it also gives unstable solutions unconditionally after some time like any other explicit scheme.
- Implicit methods are unconditionally stable. Predictor corrector scheme converge to solution in less number of iterations for the same error as BTCS.
- BTCSRK2 is unconditionally stable and has $O(\Delta t^2, \Delta x^2)$ accuracy. We will use this for our simulations of fluid flow unless otherwise specified.



Figures show the numerical solutions at $t = 0.3$ constructed using $\Delta t = 0.01$ for $D = 0.01$, only implicit schemes are able to give stable solutions.

Numerical solution in 2D: The generalization of the aforementioned methods to 2D is straightforward. The plots below show the analytic and numerical solutions of the following IVP

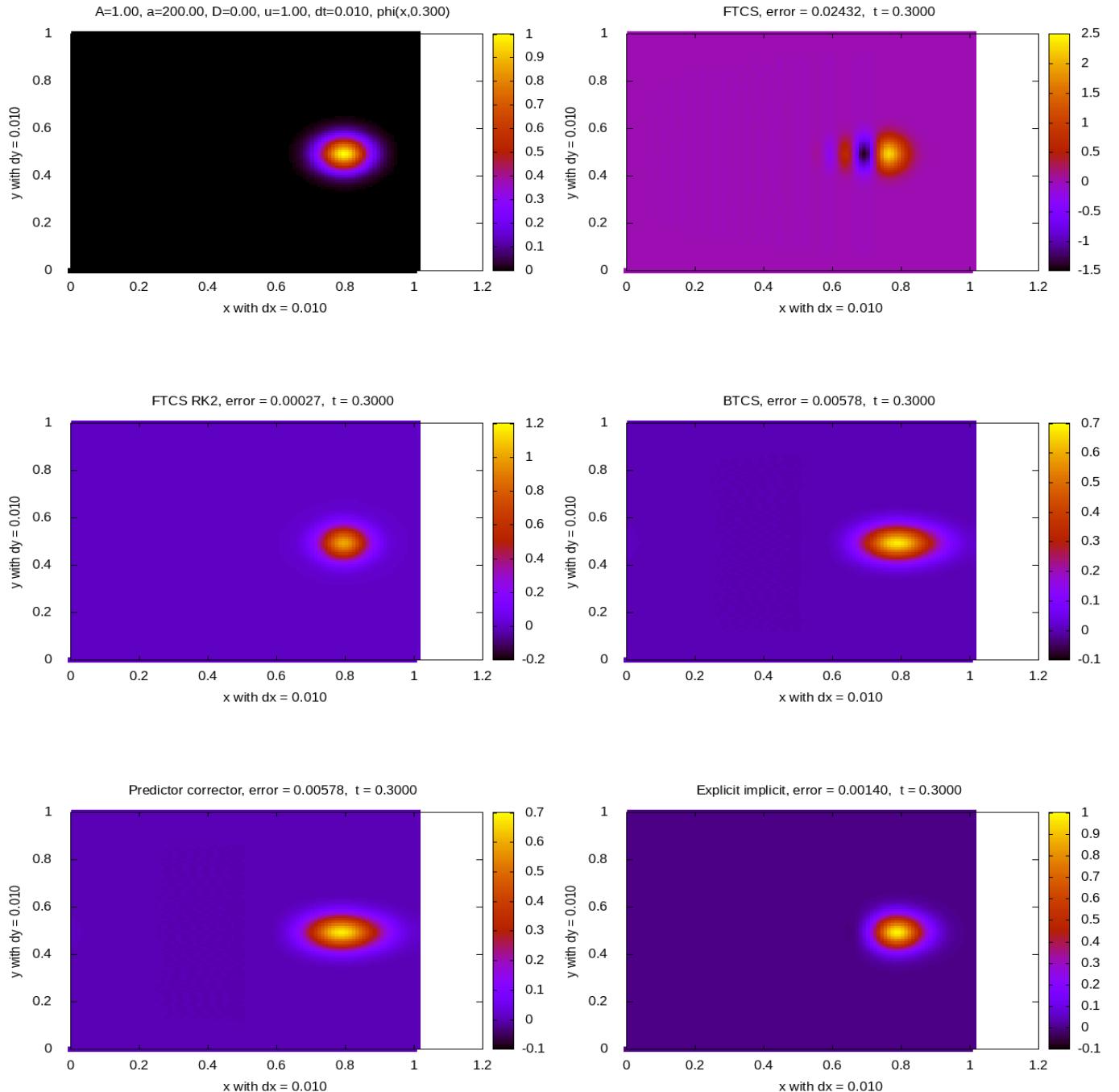
$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = D \frac{\partial^2 T}{\partial x^2}, \quad T(x, y, t = 0) = f(x, y) \quad (24)$$

with

$$f(x, y) = A \exp(-a(x^2 + y^2))$$

The analytic solution for constant u and v is

$$f(x, t) = \frac{A}{1 + 4aDt} \exp\left(\frac{-a(x - ut)^2}{1 + 4aDt}\right) \exp\left(\frac{-a(y - vt)^2}{1 + 4aDt}\right)$$



Contour plots of time evolution of a Gaussian centered at $(0.5, 0.5)$ at $t=0$. Till $t=0.3$, for $u=1$, $v=0$.
Top left figure shows the analytic solution .

4 Numerical solution of Navier stokes equation in 2D

In this section we describe the strategy for simultaneously numerically solving the Navier stokes equations for incompressible fluid flow

$$\begin{aligned}\frac{\partial u_i}{\partial t} + u_j \partial_j u_i - \frac{1}{[Re]} \partial^2 u_i &= -[Eu] \partial_i P \\ \partial_i u_i &= 0\end{aligned}$$

with arbitrary boundary conditions. Few things to note

- The Navier stokes momentum equation is non-linear multivariate boundary value problem with Dirichlet boundary conditions.
- The momentum equation contains a pressure gradient as the source term in RHS but there is no governing equation for pressure. For compressible flows one can use an equation of state for pressure such as thermodynamic pressure, and solve for the pressure field by solving corresponding convection diffusion equation for temperature. Then use that solution for evolution of velocity field.
- For an incompressible fluid one can get the pressure by solving the Poisson's equation for pressure derived in appendix (A.3).

$$\partial^2 P = -\frac{1}{[Eu]} (\partial_i u_j) (\partial_j u_i) \quad (25)$$

This however cannot be solved until the velocity field itself is known. Numerically there are mainly two ways to get solution for incompressible flows (1) Vorticity stream-function approach also known as non-primitive variable approach and (2) primitive variable approach. We describe both of them briefly in the following sections

4.1 Vorticity stream-function approach

This approach only works in 2D and uses the fact that $\nabla \times \nabla P = 0$ and eliminates the pressure term by taking curl of NS momentum equation. Doing this we get transport equation for vorticity. Derivation is given in appendix A.4.

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{u} \bullet \nabla) \vec{\omega} = (\vec{\omega} \bullet \nabla) \vec{u} + \frac{1}{[Re]} \nabla^2 \vec{\omega}$$

For 2D let $\vec{u} = (u, v)$ then the above equation reduces to

$$\frac{\partial \omega}{\partial t} = -u \partial_x \omega - v \partial_y \omega + \frac{1}{[Re]} (\partial_x^2 + \partial_y^2) \omega \quad (26)$$

Where $\omega = \partial_x v - \partial_y u$. The incompressibility condition in 2D can be satisfied by assuming $u = \partial_y \psi$ and $v = -\partial_x \psi$. Where ψ can be obtained from the poisson equation

$$\omega = \partial_x v - \partial_y u = -(\partial_x^2 + \partial_y^2) \psi \quad (27)$$

Now since the pressure has been eliminated equation(26) and (27) can be solved numerically to get ω and ψ and the velocity field can be derived from these. Even though pressure has been eliminated from these equation it can be obtained once velocity field is known by solving the pressure Poisson equation. The algorithm we used to implement this method in the code is the following

1. Initialize u, v and impose noslip boundary conditions.
2. Impose boundary conditions on ψ
3. Impose boundary conditions on ω .
4. Update ω using the finite difference version of convection diffusion equation (26) .
5. Impose boundary conditions on ω .
6. Solve discretized version of Poisson equation for ψ with the boundary conditions on ψ and latest ω .
7. Calculate $u = \partial_y \psi, v = -\partial_x \psi$
8. Increase time by one time step.
9. Go back to step 3.

Even though the method is only applicable in 2D, the advantage is that the velocities are exactly divergence less, not just up to machine accuracy. The method therefore handles incompressibility quite well. The boundary conditions on the non-primitive variables ω, ψ depend on the geometry of the problem. For example consider the **lid driven cavity flow problem** with $u = v = 0$ on the left right and bottom walls and $u = U, v = 0$ on the top wall. Since the velocities don't change along the walls, they are surface of constant ψ . We take this constant to be zero. For vorticity we do the Taylor expansion at the boundary point (i, j) and for this example take $\Delta x = \Delta y$

$$\begin{aligned}\psi_{i+n,j+m} &= \psi_{i,j} + n(\partial_x \psi)_{i,j} \Delta x + m(\partial_y \psi)_{i,j} \Delta y + \frac{1}{2}(\partial_x^2 \psi \Delta x^2 + \partial_y^2 \psi \Delta y^2)_{i,j}, \quad n, m = \pm 1, 0 \\ \psi_{i+n,j+m} &= \psi_{i,j} + (mu_{i,j} - nv_{i,j})\Delta x - \frac{1}{2}\omega_{i,j}\Delta x^2 \\ \omega_{i,j} &= \frac{2}{\Delta x^2}(\psi_{i,j} - \psi_{i+n,j+m}) + \frac{2}{\Delta x}(mu_{i,j} - nv_{i,j})\end{aligned}$$

Applying this on left, right bottoms and top walls respectively we get

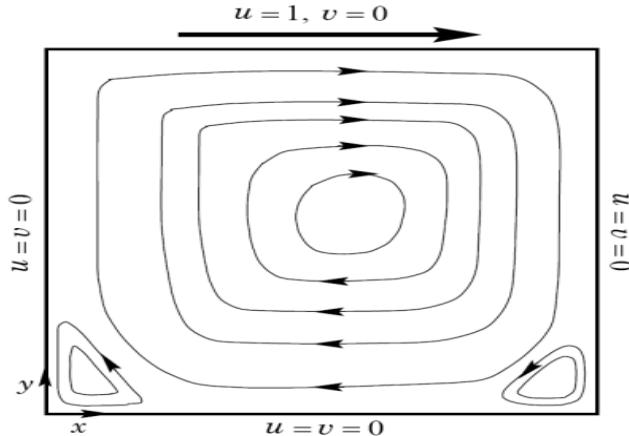
$$\omega_{1,j} = \frac{2}{\Delta x^2}(\psi_{1,j} - \psi_{2,j}) \quad (28)$$

$$\omega_{N,j} = \frac{2}{\Delta x^2}(\psi_{N,j} - \psi_{N-1,j}) \quad (29)$$

$$\omega_{i,1} = \frac{2}{\Delta x^2}(\psi_{i,1} - \psi_{i,2}) \quad (30)$$

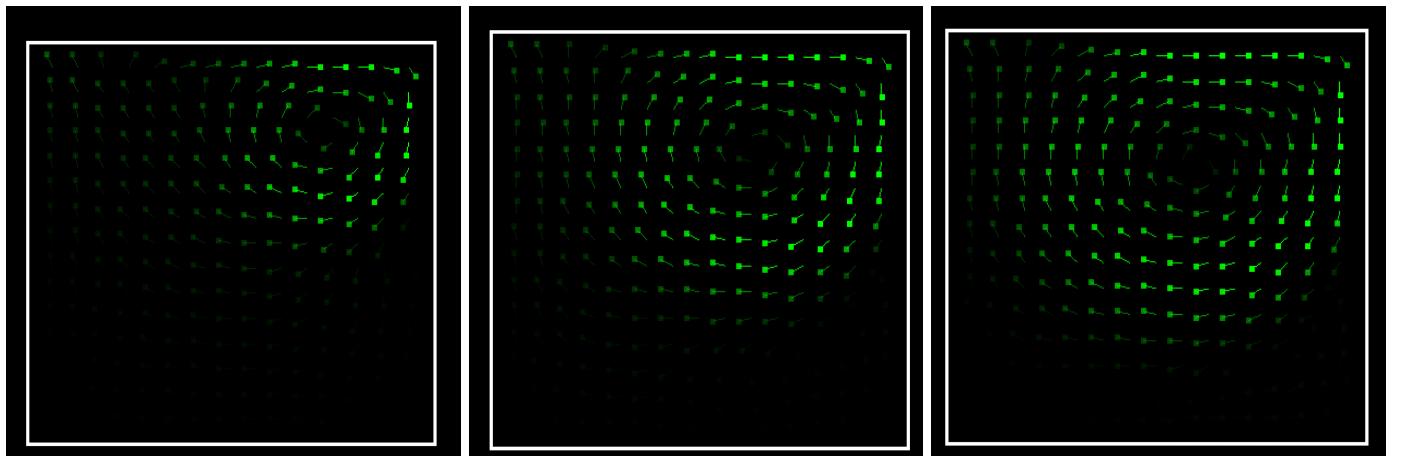
$$\omega_{i,M} = \frac{2}{\Delta x^2}(\psi_{i,M} - \psi_{i,M-1}) - \frac{2U}{\Delta x} \quad (31)$$

The driven cavity flow problem serves as the benchmark for fluid flow solvers. We therefore use it to verify our our results and compare with the numerical examples given in the following papers [5], [6]



Stream lines for Lid driven cavity flow problem

The results from the paper [5] are shown on the next page. The figures below show the computed velocity field from our simulation at times $t = 2.34, 4.69, 7.03, 9.38$ respectively for the fluid flow with $[Re] = 400$. Note that all the arrows are drawn with equal length and the magnitude of a vector is denoted by it's brightness.



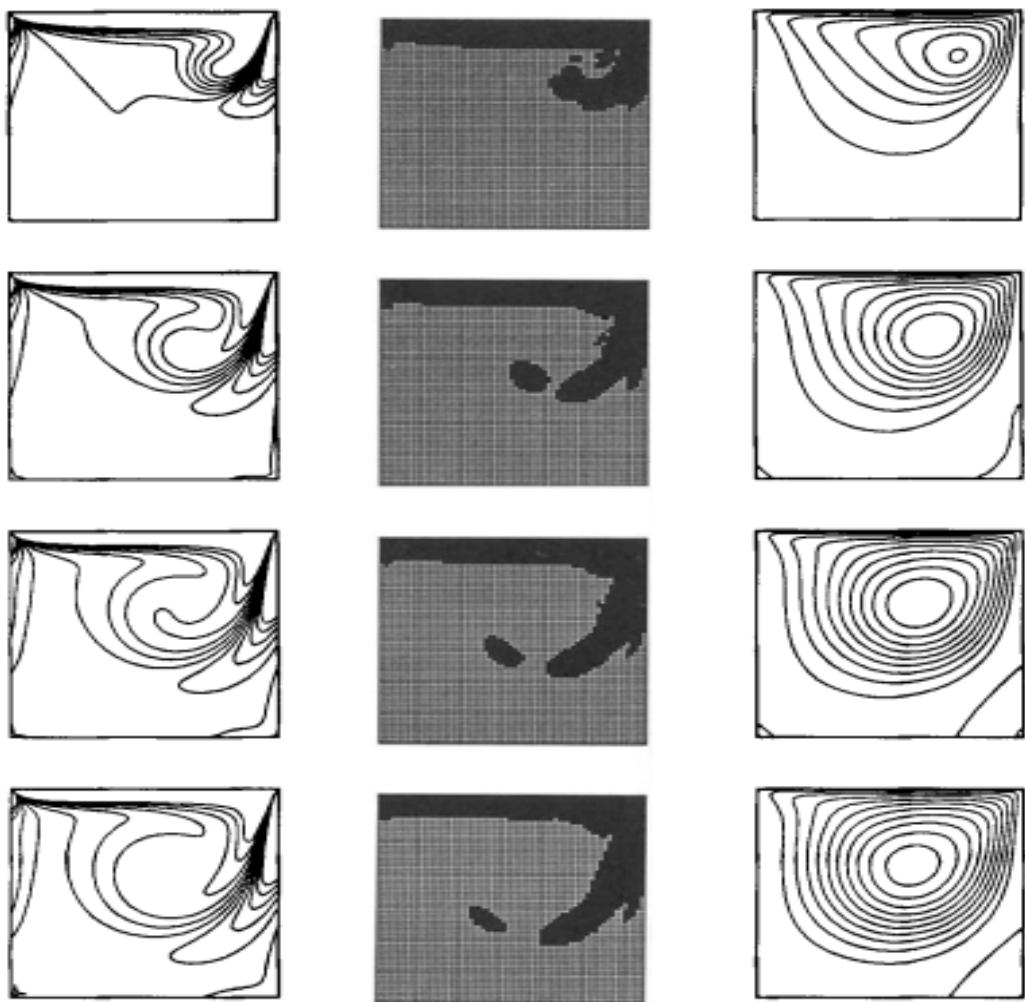
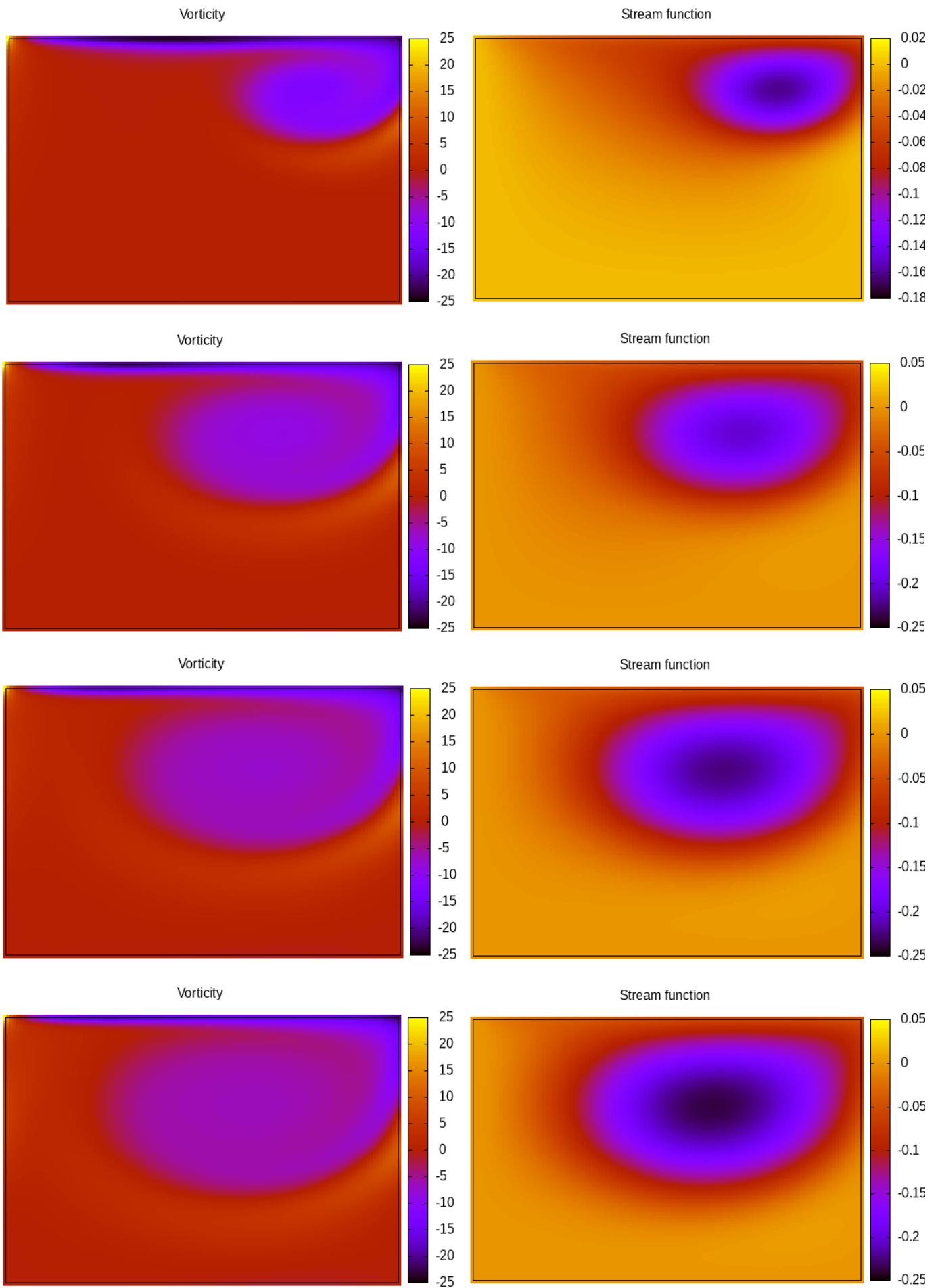


Figure 2. Driven cavity flow at Reynolds number 400: solution obtained by the AIE method; vorticity, distribution of the implicit elements, and streamfunction at $t = 2.34, 4.69, 7.03$ and 9.38

The figures on the left show the vorticity and the figures on the right show stream function at times $t = 2.34, 4.69, 7.03, 9.38$ respectively for the fluid flow with $[Re] = 400$. The corresponding results from our simulation are shown on the next page.



Vorticity and stream function at times $t = 2.34, 4.69, 7.03, 9.38$ respectively for the fluid flow in lid driven cavity with $[Re] = 400$ computed on 128×128 grid.

4.2 Pressure corrector or Primitive variable approach

This method makes use of the fact that the pressure in momentum equation can be derived from the incompressibility condition. We therefore do not eliminate pressure from the momentum equation but use it to correct the velocity field such that it satisfies the incompressibility condition. The algorithm we use is known as MAC scheme. The algorithm works as follows. Using our initial guess for pressure \bar{p} we can update the velocities from the equations

$$\bar{u}^{n+1} = u^n + \Delta t (-u\partial_x u - v\partial_y u + \mu\partial^2 u)^n - \Delta t [Eu]\partial_x \bar{p} \quad (32)$$

$$\bar{v}^{n+1} = v^n + \Delta t (-u\partial_x v - v\partial_y v + \mu\partial^2 v)^n - \Delta t [Eu]\partial_y \bar{p} \quad (33)$$

Using any of the stable methods we discussed for solving convection diffusion equations. This gives us updated values of velocities $\bar{u}^{n+1}, \bar{v}^{n+1}$. The bar on the velocities and pressure indicate that this is just our guess for u, v and p since they may not satisfy incompressibility condition.

$$\partial_x u^{n+1} + \partial_y v^{n+1} = 0 \quad (34)$$

The correct velocities can be obtained if we have correct values of pressure i.e.

$$u^{n+1} = u^n + \Delta t (-u\partial_x u - v\partial_y u + \mu\partial^2 u)^n - [Eu]\Delta t \partial_x p \quad (35)$$

$$v^{n+1} = v^n + \Delta t (-u\partial_x v - v\partial_y v + \mu\partial^2 v)^n - [Eu]\Delta t \partial_y p \quad (36)$$

Subtracting the two sets of equation we get

$$u^{n+1} = \bar{u}^{n+1} - \Delta t [Eu]\partial_x p' \quad (37)$$

$$v^{n+1} = \bar{v}^{n+1} - \Delta t [Eu]\partial_y p' \quad (38)$$

Where $p' = p - \bar{p}$ is the pressure correction required to make velocities divergenceless. Taking derivatives and adding above two equations and using the incompressibility condition we get

$$(\partial_x^2 + \partial_y^2)p' = \frac{1}{\Delta t [Eu]} (\partial_x \bar{u}^{n+1} + \partial_y \bar{v}^{n+1}) \quad (39)$$

This is poisson's equation for pressure correction, once it is solved we can update the pressure as $p = p' + \bar{p}$ and from it we can calculate the correct velocities. Using central difference for space derivatives we can write

$$(\partial_x^2 + \partial_y^2)p' = \frac{p'_E - 2p'_P + p'_W}{\Delta x^2} + \frac{p'_N - 2p'_P + p'_S}{\Delta y^2} \quad (40)$$

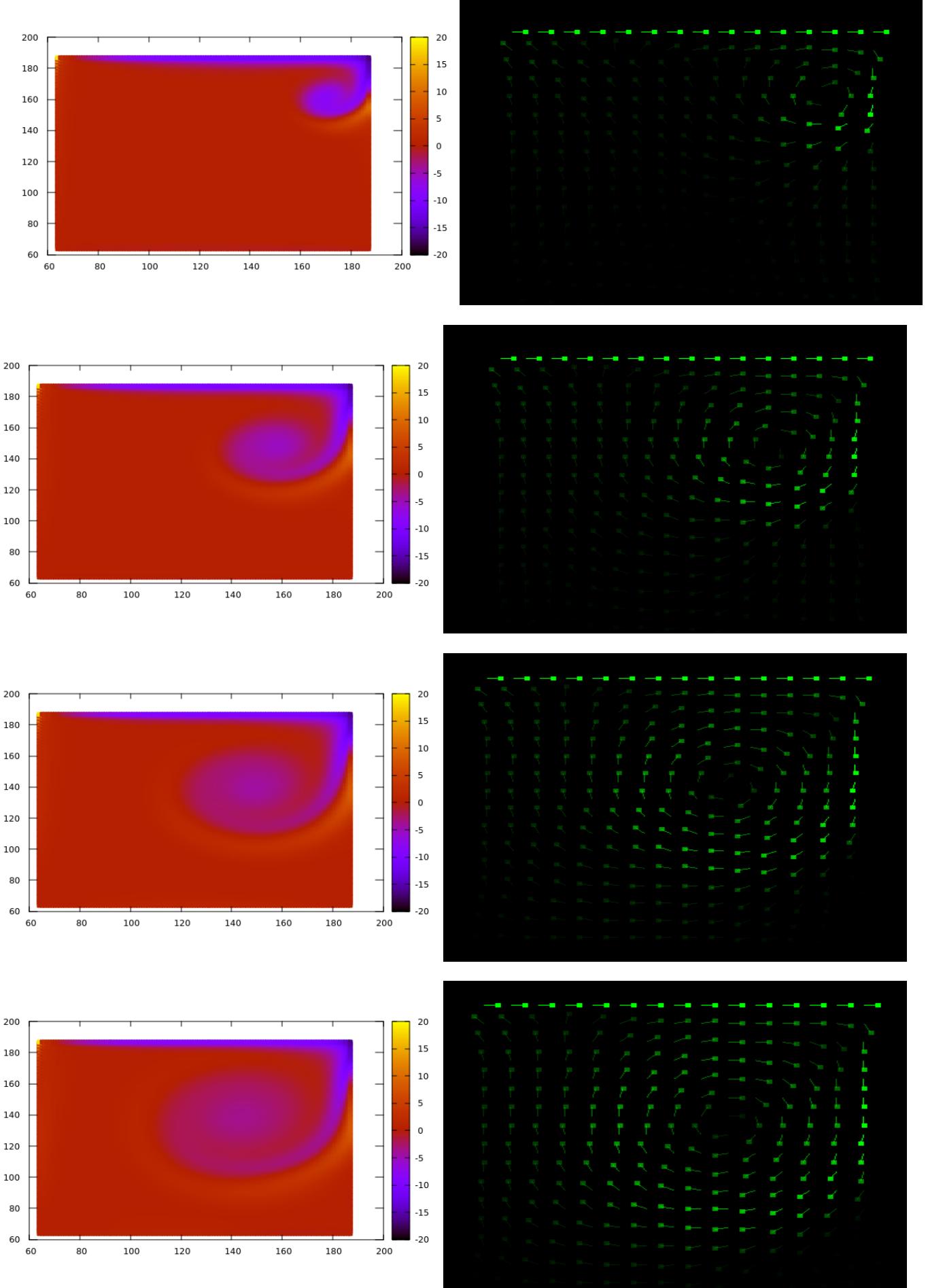
Solving poisson equation is a recursive and time consuming process which introduces it's own numerical errors. Many of the pressure corrector schemes are same till this point. Now comes the crucial step where various pressure corrector schemes differ. MAC scheme assumes that velocities at a point correct pressures only at that point that is to say $p'_W = p'_N = p'_E = p'_S = 0$. This gives

$$p'_P = -\frac{1}{2\Delta t [Eu]} \frac{\left(\frac{\bar{u}_E - \bar{u}_W}{2\Delta x} + \frac{\bar{v}_N - \bar{v}_S}{2\Delta y}\right)}{\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)} \quad (41)$$

Once this pressure correction is known the correct pressure can be calculated from $p = p' + \bar{p}$. From which the corrected velocities can also be calculated from equation (37) (21). One can check that velocities corrected this way are indeed divergence less. The algorithm works as follows

1. Initialize u, v and p .
2. Impose noslip boundary conditions on velocities.
3. Predict the new u and v from the discretized NS equations.
4. Impose noslip boundary conditions on velocities.
5. Calculate the pressure correction from equation(41) and calculate the corrected pressure from $p = p' + \bar{p}$
6. Calculate the updated velocities from equations (37) (38).
7. Calculate the average divergence of the velocity field per unit grid point, if it is not less than the required accuracy then go back to step 2. Break this loop once it is within required accuracy limit.
8. Update u and v for next time step and go back to step 2.

Again as the bench mark problem we solve the lid driven cavity flow for the conditions we used before for the vorticity stream approach. The plots for the velocity field and vorticity are shown on below.



Vorticity and velocity fields at times $t = 2.34, 4.69, 7.03, 9.38$ respectively for the fluid flow in lid driven cavity with $[Re] = 400$ computed on 128×128 grid.

5 Turbulence in Two dimensions

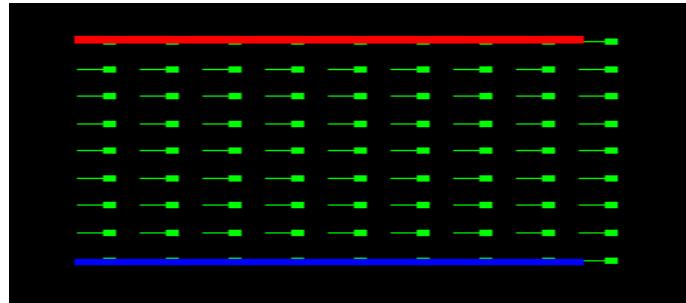
Turbulent flows are characterised by randomness and presence of eddies of varied length scales in the flow, largest of which can be of the system size and smallest apparently only limited by the resolution of the numerical simulation. A dye placed in a fluid under turbulent motion would diffuse and mix with the fluid at more varied length scales than it would under a low velocity laminar flow, and so would the vorticity. Which is apparent from the fact that the equation of motion of a dye with mass density $\zeta(x, t)$ placed in an incompressible fluid are similar to vorticity transport equations apart from one term. For a dye the equation of motion in 3D is

$$\frac{\partial \zeta}{\partial t} + u_j \partial_j \zeta = -[Eu] \partial_i P + \frac{1}{[Re]} \partial^2 \zeta \quad (42)$$

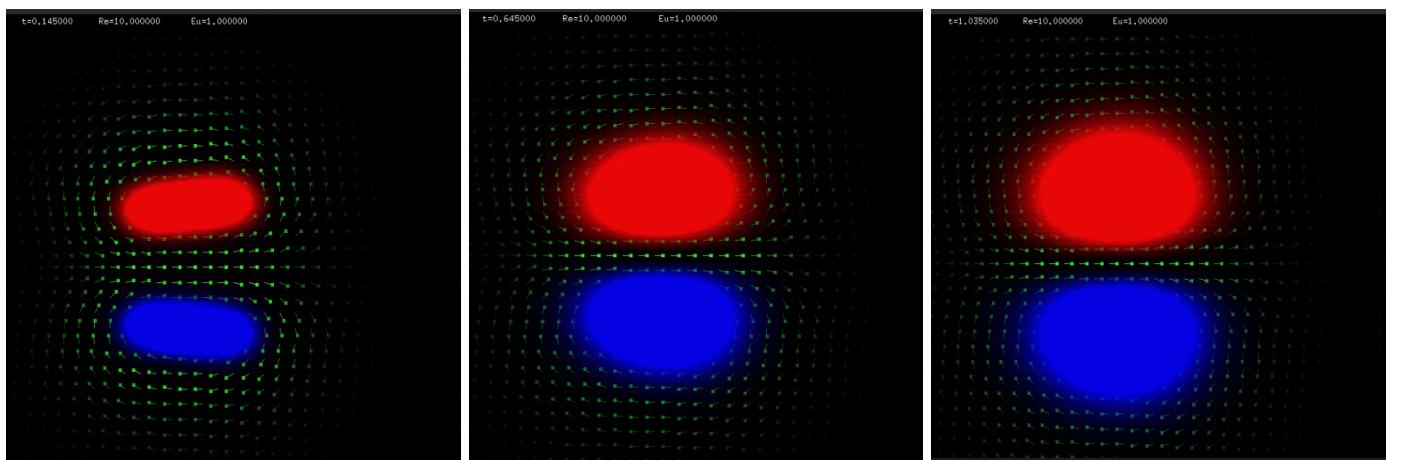
while the vorticity transport equations are

$$\frac{\partial \vec{\omega}}{\partial t} + (\vec{u} \bullet \nabla) \vec{\omega} = (\vec{\omega} \bullet \nabla) \vec{u} + \frac{1}{[Re]} \nabla^2 \vec{\omega}$$

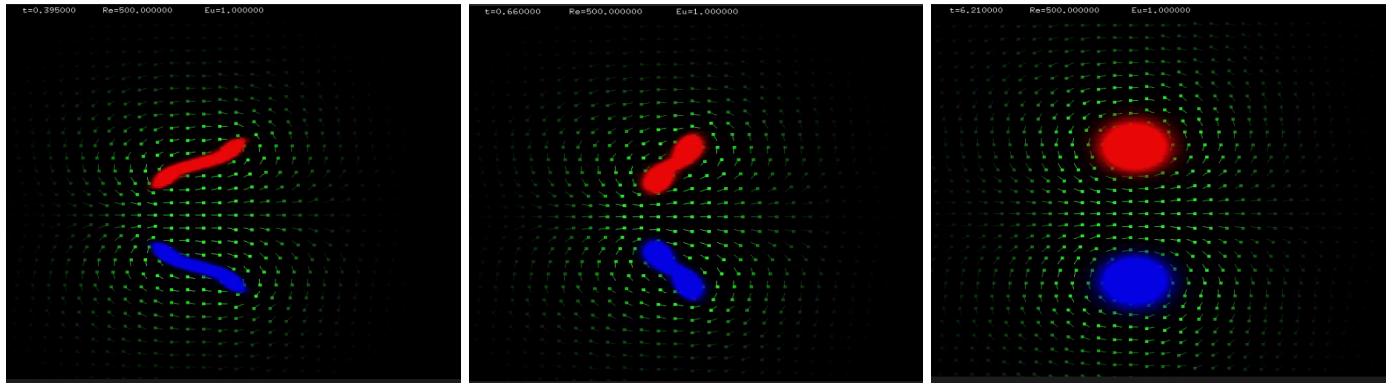
The term $(\vec{\omega} \bullet \nabla) \vec{u}$ is known as the vorticity stretching term, it's responsible for decay of energy from larger eddies to smaller, a feature which is not present in 2D turbulent flows, since this term vanishes in 2D. In two-dimensional inviscid $\nu = 0$ or $[Re] = \infty$ flows the vorticity transport equation reduces to the equation of conservation of the average vorticity over a volume, therefore the vortical structures survive for long periods of time, and keep up their vorticity as they move. Such as vortex sheet or vortex dipole. Smaller structures become visible at higher Reynold's number, and in the course of the evolution of the system the energy is cascaded from small scales to larger scales. Small scale vortices with same sense of rotation merge into bigger vortices and dissipate energy. Consider the following example of blow of air in flatland. The initial conditions are $u(x, y, 0) = 1$, for $x \in [1/4, 1/2]$, $y \in [1/2 - 1/8 \leq y \leq 1/2 + 1/8]$ and $v(x, y) = 0$, for all x and y . The domain of analysis in dimensionless units is $0 \leq x \leq 1$, $0 \leq y \leq 1$. The computations are done on 128×128 uniform mesh with periodic boundary conditions. With our simulation we are able to reproduce some of the results in the following paper [9].



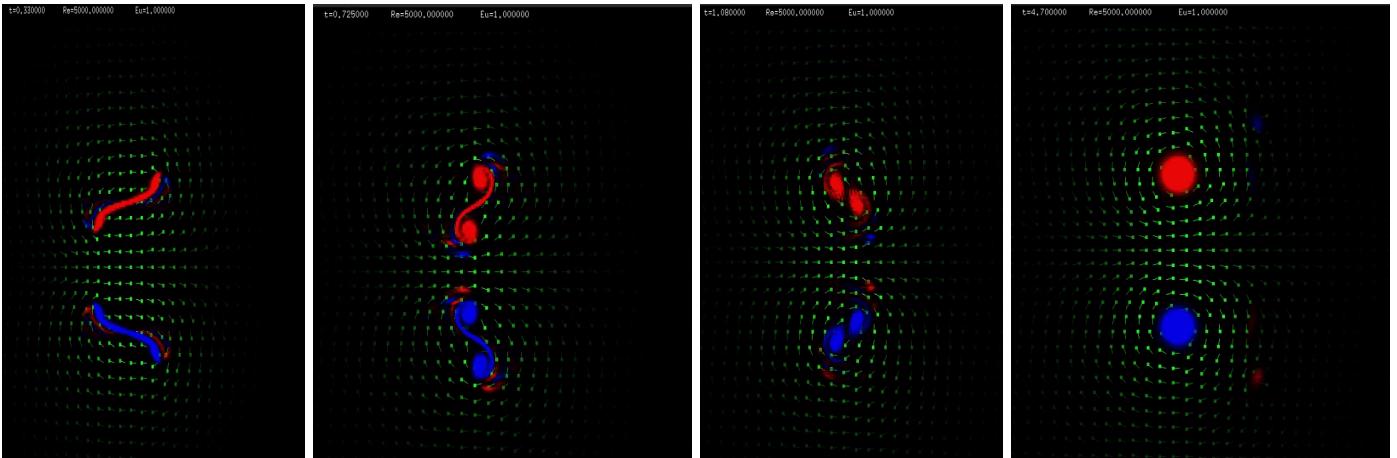
Initial condition of the flow. The arrows denote the velocity field and the the red and blue color denote positive and negative values of vorticity. Magnitude of quantities is denoted by brightness of the pixel.



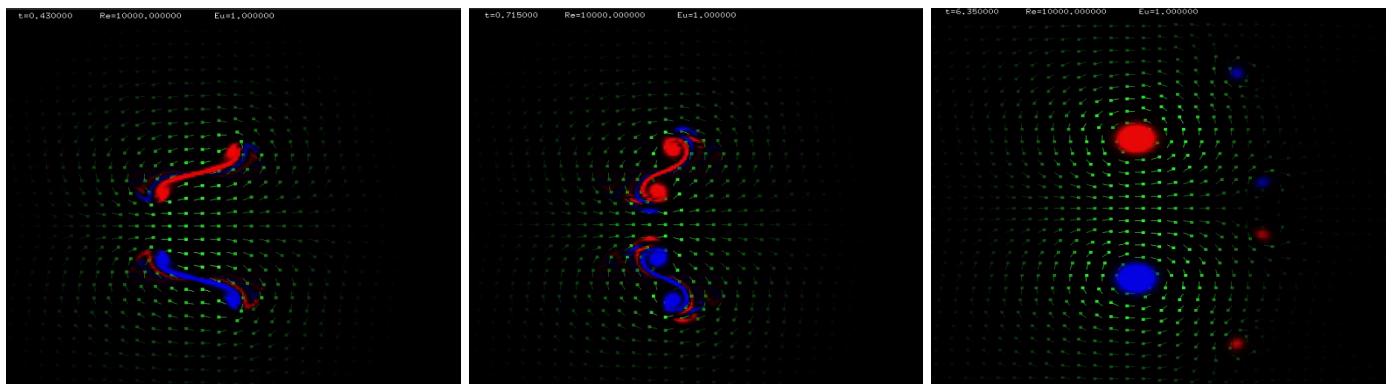
Snapshots of evolution of the system at different times, with $[Re]=10$. The diffusion coefficient $1/[Re]$ is large and the vorticity just diffuses into the surrounding. Creating dipole of vortex which maintain their vorticity and move with the flow. See video file Re10.mp4 for the simulation.



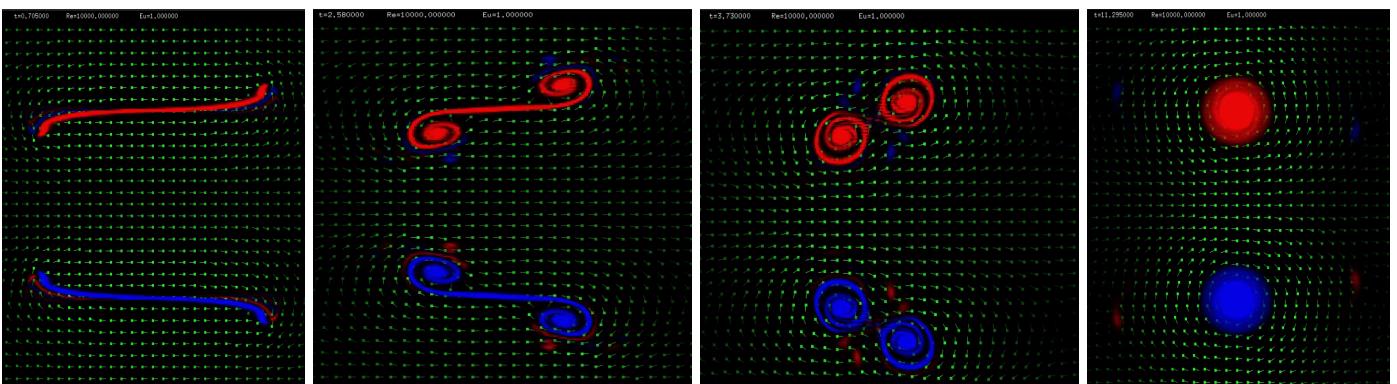
Snapshots of evolution of the system at different times, with $[Re]=500$. Smaller vortices begin to appear for $[Re] > 500$, which then combine and give the final state, the settling time increases $\propto \log[Re]$. See video file Re500.mp4 for the simulation.



Snapshots for $[Re]=5000$, even smaller structure is visible, the separated vortices rotate around each other then combine. The rotational kinetic energy of this collision gives rise to even smaller vortices, which get separated from the parent vortices and move forward to carry the thrust. This process is called filamentation. See video file Re5k.mp4 for the simulation.



$[Re]=10000$. Presence of smaller eddies which get separated is indication of even smaller scale interaction at the parent vortex. We therefore zoom in where the collision happens. See video file Re10k.mp4 for the simulation.



Zoomed in version at $[Re]=10000$. Presence of even smaller structure. The filamentation can be seen. See video file Re10k_z.mp4 for the simulation.

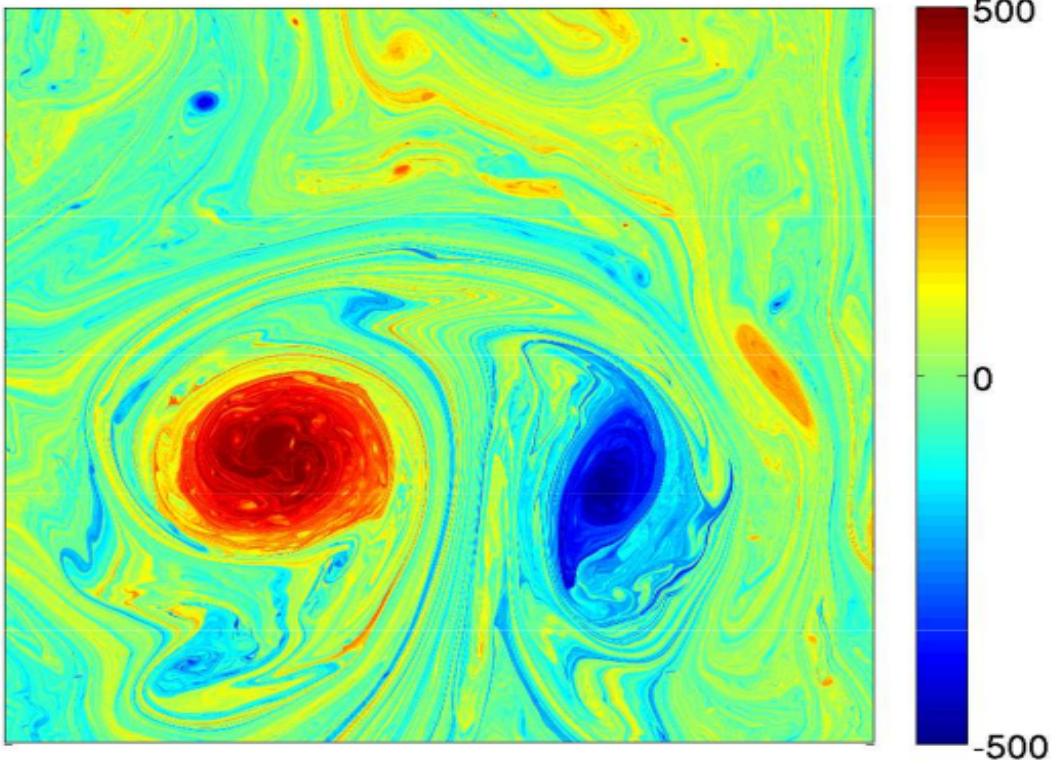


Figure 10: Instantaneous vorticity field $\zeta(x, y)$ in equilibrium 2D turbulence in one-quarter of a doubly periodic domain at high Re . Notice the multiply-folded filaments between the coherent vortices near the forcing scale, the internal fluctuations within the dominant dipole, and the much finer-scale vortices in the far-field regions with weak large-scale strain. This solution has a spectrum close to $E \propto k^{-3}$ on scales smaller than F . (Bracco and McWilliams, 2010)

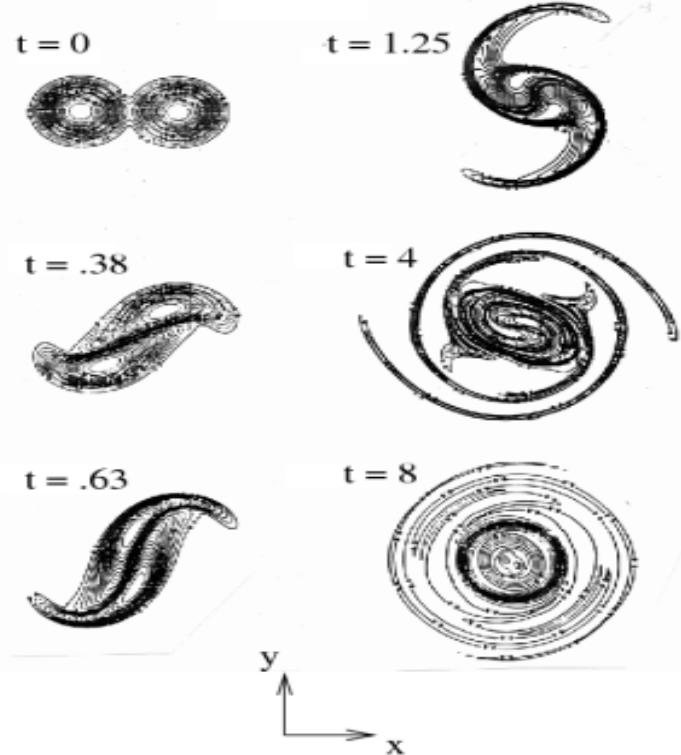


Figure 4: Computational solution for the merger of two like-sign, bare-monopole vortices (in non-dimensional time units scaled by L/V) initially located near each other. The exterior strain field from each vortex deforms the vorticity distribution of the other one so that the ζ fields wrap around each other; their centers move together along spiral trajectories; and ultimately they blend together after viscous diffusion smooths the strong gradients. While this is occurring, vorticity filaments are cast off from the merging vortices, stretched by the exterior strain field, and dissipated by viscosity. (Adapted from McWilliams, 1991.)

These are some of the results from [9] that we reproduced in our simulation. Including the energy spectrum and settling times.

One of the features of 2D turbulence is that the energy is cascaded from small spatial scales to larger. To see this consider the kinetic energy and enstrophy (vorticity variance)

$$E = \frac{1}{2} \int \int dx dy (\mathbf{u}^2)$$

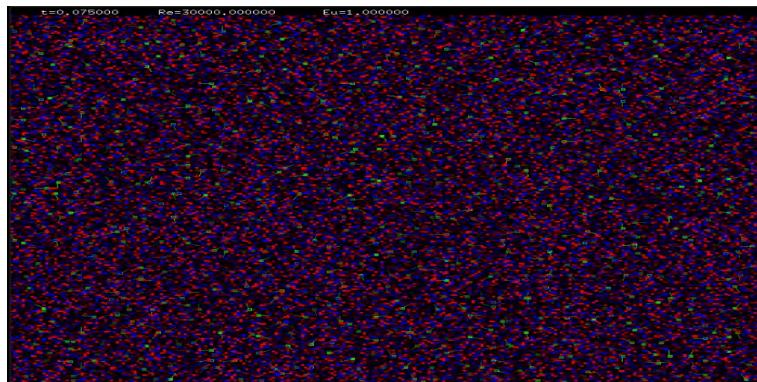
$$E_{ns} = \frac{1}{2} \int \int dx dy \omega^2$$

The rate of change of these can be derived from the NS equations as

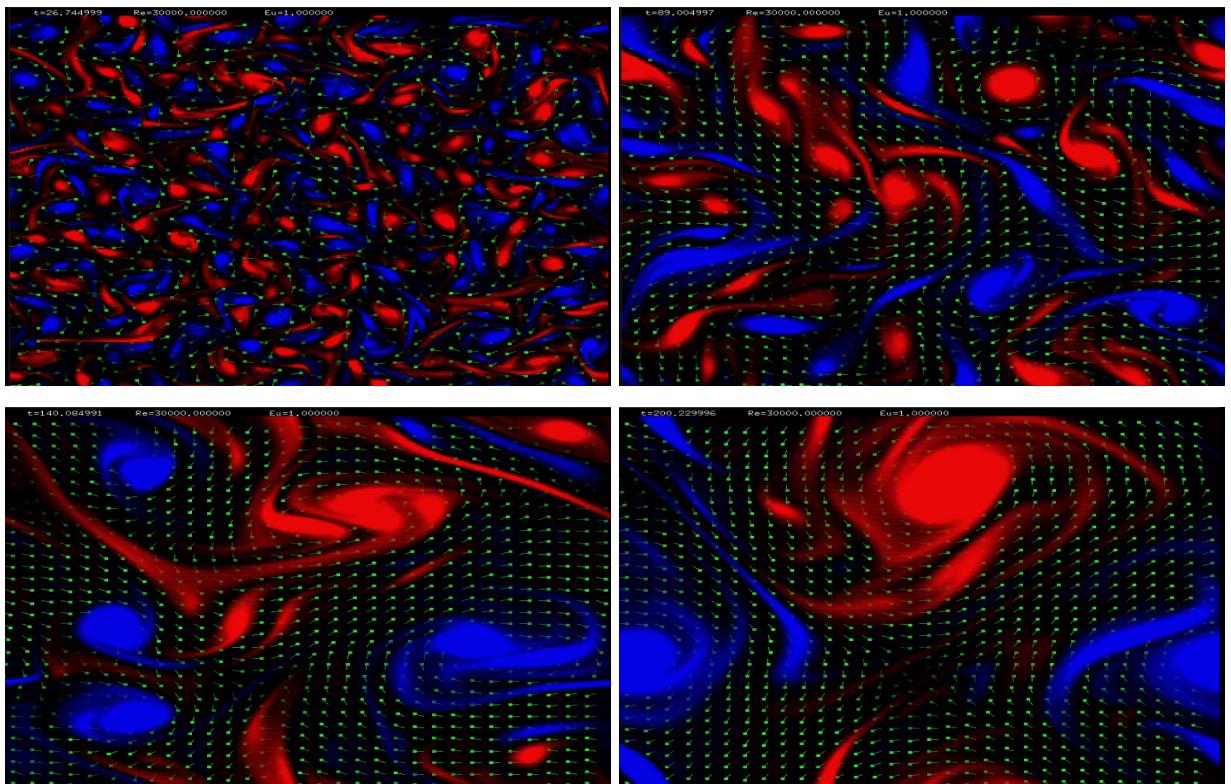
$$\frac{dE}{dt} = -\nu \int \int dx dy (\nabla \mathbf{u})^2$$

$$\frac{dE_{ns}}{dt} = -\nu \int \int dx dy (\nabla \omega)^2$$

Taking the Fourier transform of the equation above we see that $E(k) \propto k^2$, $E_{ns}(k) \propto k^4$. In the inviscid limit both E and E_{ns} are conserved in time, the interaction with the surrounding statistically will tend to broaden the spectra then the only way both the quantities can be conserved is by having more kinetic energy towards smaller values of k (i.e. larger spatial scales) and more enstrophy towards larger k (i.e.) smaller system scales. These are known as inverse energy cascade and the forward enstrophy cascade of 2D turbulence. Consider the following spatially smooth initial conditions where the velocity field is initialised to be uniformly random. The energy cascade in 2D result in emergence of vortex, which in turn merge and dissipate enstrophy. The vortices becomes fewer and larger over time till the largest is about the system size.



Random, spatially smooth initial conditions



State of the system at non dimensional times 26.744, 89.004, 149.084, 200.22 with $[Re]=30000$.See the video cascade.mp4 for the simulation.

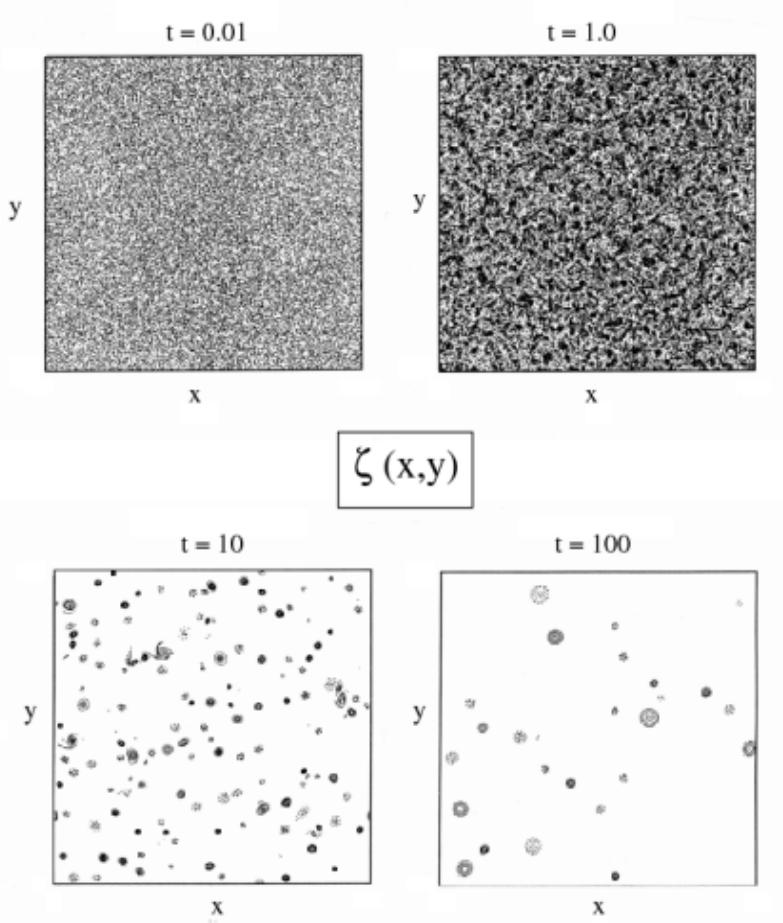


Figure 3: Vortex emergence and evolution in computational 2D turbulence, as seen in $\zeta(x,y)$ at sequential times, with random, spatially smooth initial conditions. Solid contours are for positive ζ , and dashed ones are for negative ζ . The contour interval is twice as large in the first panel as in the others. The times are non-dimensional based on an advective scaling L/V . (Adapted from McWilliams, 1984.)

Result from the paper [9]

6 Conclusion

In this work we have analysed the numerical methods to solve convection diffusion type problems in 1D and 2D and compared their performance, by comparing with analytical solutions. We then made use of these methods and applied them in conjunction with the vorticity-stream function approach and pressure-corrector approach to obtain numerical simulation of viscous fluid flow by solving incompressible Navier Stokes equation in 2D. We compared the methods using the lid driven cavity problem as benchmark. We also discussed in little detail, turbulence in 2D and managed to get numerical solution of fluid flow from Reynold's number 10-50000, the only limitation being the grid resolution. The code developed for this project can be found here, it can be used for any standard fluid flow problem. The code is getting updated periodically and we wish to use it for more advanced problems in future.

Appendix

A.1 Integration of a field on a changing volume

$$\frac{d}{dt} \int_{V(t)} F(x, t) dV = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \left[\int_{V(t+\Delta t)} F(x, t + \Delta t) dV - \int_{V(t)} F(x, t) dV \right]$$

where

$$\begin{aligned} \int_{V(t+\Delta t)} F(x, t + \Delta t) dV &\approx \int_{V(t)} F(x, t) dV + \int_{V(t)} \Delta t \frac{\partial F(x, t)}{\partial t} dV + \int_{\Delta V} F(x, t) dV + \int_{\Delta V} \Delta t \frac{\partial F(x, t)}{\partial t} dV \\ \implies \frac{d}{dt} \int_{V(t)} F(x, t) dV &= \lim_{\Delta t \rightarrow 0} \left[\int_{V(t)} \frac{\partial F(x, t)}{\partial t} dV + \frac{1}{\Delta t} \int_{\Delta V} F(x, t) dV \right] \\ \frac{d}{dt} \int_{V(t)} F(x, t) dV &= \int_{V(t)} \frac{\partial F(x, t)}{\partial t} dV + \int_{A(t)} F(x, t) \vec{u} \bullet \hat{n} dA \\ \frac{d}{dt} \int_{V(t)} F(x, t) dV &= \int_{V(t)} \left(\frac{\partial F(x, t)}{\partial t} + \nabla (F(x, t) \vec{u}) \right) dV \end{aligned} \quad (43)$$

A.2 Mechanics of very large, countably infinite degrees of freedom

The degrees of freedom of a gas of atoms is of the order of Avogadro's number, and a precise description of the initial conditions of such a system is well beyond our ability let alone evolution. Therefore we approximate just like we cut off any irrational number after certain decimal points for the sake of our calculation. We divide the $2n$ dimensional phase space of our system into segments of infinitesimal volume, this means that a point Q, P in phase can lie anywhere within an infinitesimal volume around $\delta Q \delta P$ it. Where

$$\delta Q \delta P = \delta q_1 \delta p_1 \delta q_2 \delta p_2 \dots \delta q_n \delta p_n \quad (44)$$

The number of states that can lie in this infinitesimal volume is given by.

$$\delta N(Q, P, t) = \rho(Q, P, t) \delta Q \delta P \quad (45)$$

Where ρ is defined to be the density of the states. Defined only statistically as the probability of the system to be in "macrostate" (Q, P) at time t . Number of states in a volume $V(t) \subset \mathbb{R}^{2n}$ is given by.

$$N_V(t) = \int_{V(t)} \rho(Q, P, t) \delta Q \delta P \quad (46)$$

We also define the quantity "material volume" $V_m(t)$ as the volume such that it always contains only the states which started inside it. By definition

$$\frac{d}{dt} (N_{V_m}(t)) = 0 \quad (47)$$

Let $A_m(t)$ be the boundary such volume and let \vec{b} be the velocity of the boundary then we have

$$\begin{aligned} \frac{d}{dt} (N_{V_m}(t)) &= \frac{d}{dt} \int_{V_m(t)} \rho(Q, P, t) \delta Q \delta P = \int_{V_m} \left(\frac{\partial \rho}{\partial t} + \nabla (\rho \vec{b}) \right) \delta Q \delta P \\ \implies \int_{V_m} \left(\frac{\partial \rho}{\partial t} + \rho \nabla \vec{b} + \vec{b} \nabla \rho \right) \delta Q \delta P & \end{aligned}$$

For material volume we have $\vec{b} = (\dot{Q}, \dot{P})$

$$\begin{aligned} \implies \int_{V_m} \left(\frac{\partial \rho}{\partial t} + \rho \left(\frac{\partial \dot{Q}}{\partial Q} + \frac{\partial \dot{P}}{\partial P} \right) + \dot{Q} \frac{\partial \rho}{\partial Q} + \dot{P} \frac{\partial \rho}{\partial P} \right) \delta Q \delta P & \\ \implies \int_{V_m} \left(\frac{\partial \rho}{\partial t} + \frac{\partial H}{\partial P} \frac{\partial \rho}{\partial Q} - \frac{\partial H}{\partial Q} \frac{\partial \rho}{\partial P} \right) \delta Q \delta P & \\ \frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + \sum_{i=1}^n \{\rho, H\}_{q_i, p_i} & = 0 \end{aligned} \quad (48)$$

Therefore the density of states in a Hamiltonian system behaves like density of incompressible fluid.

A.3 Pressure poisson equation for incompressible flow

Taking divergence of momentum equation

$$\frac{\partial}{\partial t}(\partial_i u_i) = -(\partial_i u_j)(\partial_j u_i) - (u_j \partial_j)(\partial_i u_i) - [Eu] \partial^2 P + \frac{1}{[Re]} \partial^2(\partial_i u_i) = 0$$

using incompressibility condition we get

$$\partial^2 P = -\frac{1}{[Eu]}(\partial_i u_j)(\partial_j u_i) \quad (49)$$

A.4 vorticity transport equation

Taking curl of momentum equation we get

$$\begin{aligned} \nabla \times \frac{\partial \vec{u}}{\partial t} &= -\nabla \times (\vec{u} \bullet \nabla \vec{u}) + \frac{1}{[Re]} \nabla \times \nabla^2 \vec{u} \\ \frac{\partial}{\partial t}(\nabla \times \vec{u}) &= -\nabla \times (\nabla u^2 + \vec{\omega} \times \vec{u}) + \frac{1}{[Re]} \nabla^2(\nabla \times \vec{u}) \\ \frac{\partial \vec{\omega}}{\partial t} &= -\nabla \times (\vec{\omega} \times \vec{u}) + \frac{1}{[Re]} \nabla^2 \vec{\omega} \\ \frac{\partial \vec{\omega}}{\partial t} + (\vec{u} \bullet \nabla) \vec{\omega} &= (\vec{\omega} \bullet \nabla) \vec{u} + \frac{1}{[Re]} \nabla^2 \vec{\omega} \end{aligned} \quad (50)$$

A.5 Stability of explicit numerical solution to convection diffusion equation

Consider the one-dimensional convection diffusion equation.

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} = D \frac{\partial^2 T}{\partial x^2} \quad (51)$$

The discretized version of this equation with FTCS scheme is

$$T_i^{n+1} = T_i^n - \alpha (T_{i+1}^n - T_{i-1}^n) + \beta (T_{i+1}^n - 2T_i^n + T_{i-1}^n) \quad (52)$$

where $\alpha = u \frac{\Delta t}{2\Delta x}$ and $\beta = D \frac{\Delta t}{\Delta x^2}$

A numerical solution is said to be stable if the roundoff errors in each step decay over time and do not affect the solution. Let us consider the error at a grid point

$$e_i^n = T_i^n - \bar{T}_i^n \quad (53)$$

Where T_i^n is the exact solution and \bar{T}_i^n is the approximate numerical solution due to roundoff. Since the equation is linear in T the error satisfies the same equation i.e.

$$e_i^{n+1} = e_i^n - \alpha (e_{i+1}^n - e_{i-1}^n) + \beta (e_{i+1}^n - 2e_i^n + e_{i-1}^n) \quad (54)$$

Let us take the Fourier transform of the error and since the governing equation is linear let us study only one component of the decomposition i.e.

$$e_i^n \sim g^n(k) e^{i\pi k x_i} \quad (55)$$

$$e_i^{n+1} \sim g^{n+1}(k) e^{i\pi k x_i} \quad (56)$$

Substituting these into the error equation we get

$$\frac{g^{n+1}}{g^n} = [(\alpha + \beta) e^{-i\pi k \Delta x} + (1 - 2\beta) + (\beta - \alpha) e^{i\pi k \Delta x}] \quad (57)$$

The condition for stability is that the magnitude of the error should decay with time i.e.

$$\left| \frac{g^{n+1}}{g^n} \right| \leq 1 \quad (58)$$

This gives

$$\left(1 - 4\beta \sin^2 \left(\frac{\theta}{2} \right) \right)^2 + 4\alpha^2 \sin^2 \theta \leq 1 \quad (59)$$

Where $\theta = k\pi\Delta x$. The stability condition can also be expressed as

$$0 \leq 4\alpha^2 \leq 2\beta \leq 1 \quad (60)$$

Putting the values for α and β we get

$$0 \leq u^2 \frac{\Delta t^2}{\Delta x^2} \leq 2D \frac{\Delta t}{\Delta x^2} \leq 1 \quad (61)$$

Which gives

$$\Delta t \leq \frac{2D}{u^2} \quad (62)$$

$$\Delta x \geq |u|\Delta t \quad (63)$$

Note that for $D=0$ no value to Δt can give stable solution. To get stable solution in that case one must use either forward or backward difference formula for space derivatives.

A.6 Discription of the code

In this section explain the code which does so and how to interpret various outputs produced by the code. The code is constantly improving and getting better, and here we only describe the first iteration of it. The project is written in C++ and the source files can be found at this [github link](#).

The project comprises mainly of two C++ classes named "fluid" and "geometry". The fluid class is created for doing the following things.

- Defining an N by M grid on which the fluid flow equations will be solved.
- Initializing fluid parameters such as grid spacing ($\Delta x = \Delta y$), time step for evolution of solution (Δt), density of the fluid (ρ) and coefficient of viscosity (η). The spatial domain of analysis is $0 \leq x \leq N\Delta x$, $0 \leq y \leq M\Delta y$.
- Defining fields vx, vy, temp_vx, temp_vy for velocities. Temp variables hold intermediate results for the calculations.
- Defining scalar field P for pressure.
- Defining arrays bx and by which hold the locations of all the boundary points.
- Giving initial values of boundary points and velocity fields using functions init_boundary(), init_velocity().
- Imposing noslip boundary conditions using the function noslip().
- Updating velocity field from Navier Stokes equation using the function Vel_evolve().
- Evolving velocity and pressure fields using MAC scheme using function Evolve().
- Evolving a dye placed in our fluid field using the function Den_evolve().

The class geometry does the following jobs for us.

- Gives locations geometrical shapes in 2D such as a point, or a line between two points, a rectangle, circle. These points are used as boundary points in our simulation. For example a flow inside a pipe can be simulated using two parallel lines as boundaries.
- Draws scalar field on each location screen, the brightness of the pixel denotes the magnitude of the field at that location and color indicates the sign i.e. red for positive values and blue for negative value.
- Draws vector fields on screen where each arrow is of equal length and the magnitude of the vector is denoted by the brightness of the arrow on screen.
- Calculates partial derivatives, divergence, laplacian of the vector fields using either central or backwards difference scheme depending on the choice of the user.
- Draws streamlines for the flow.

References

- [1] PIJUSH K. KUNDU, I RA M. COHEN, DAVID R. DOWLING : Fluid Mechanics, fifth edition, Chapter 4,5,6,8,10.
- [2] Paulo S. B. Zdanski , M. A. Ortega & Nide G. C. R. Fico :NUMERICAL SIMULATION OF THE INCOMPRESSIBLE NAVIER-STOKES EQUATIONS
- [3] Jos Stam, Real-Time Fluid Dynamics for Games
- [4] Numerical Methods for the Solution of Partial Differential Equations, Luciano Rezzolla, Albert Einstein Institute, Max-Planck-Institute for Gravitational Physics, Potsdam, Germany
- [5] Solution techniques for the vorticity-streamfunction formulation of two-dimensional unsteady incompressible flows, T. E. TEZDUYAR, J. LIOU, D. K. GANJOO AND M. BEHR
- [6] Streamfunction-Vorticity Formulation. A. Salih Department of Aerospace Engineering Indian Institute of Space Science and Technology, Thiruvananthapuram
- [7] Advanced Computational fluid mechanics Lectures by Prof. Suman Chakraborty
- [8] Jayanta Bhattacharjee - Introduction to fluid dynamics and turbulence
- [9] Two-dimensional turbulence: a physicist approach. Patric kTabeling Laboratoire de Physique Statistique, France