Available online www.ejaet.com

European Journal of Advances in Engineering and Technology, 2023, 10(5): 61-65

Research Article ISSN: 2394 - 658X

Study of the incompressible Navier-Stokes equations in a square domain with prescribed velocities along the boundary in two dimensional

Papa Touty Traore[1], Moussa Dieng, Mor NDIAYE, Issa Diagne

[1]Department of physics, Cheikh Anta Diop University, Dakar, Senegal -11000
papatoutytraore@gmail.com (only corresponding author)

ABSTRACT

The aim of this work is to solves the incompressible Navier-Stokes equations in square domain with prescribed velocities along the boundary. The solution method is finite differencing on a staggered grid with implicit diffusion and a Chorin projection method for the pressure. Visualization is done by a colormap-isoline plot for pressure and normalized quiver and streamline plot for the velocity field. The standard setup solves a lid driven cavity problem.

Key words: Incompressible, Navier -stocks-velocities

INTRODUCTION

In physics, the Navier–Stokes equations [1], named after Claude-Louis Navier and George Gabriel Stokes, each of whom derived these equations independently, are a set of nonlinear partial differential equations which describe the motion of viscous fluids and are the fundamental equations of fluid dynamics. These equations result from applying Newton's second law [2] to fluid dynamics [3], along with the assumption that the stress in the fluid is the sum of a diffusing viscous term [4] (based on the way that the velocity is changing) and a pressure term, describing viscous flow.

The Navier–Stokes equations are based on the work of Leonhard Euler (1707–1783). Euler considered the fluid as a continuum allowing him to derive governing equations for the motion of inviscid (non-viscous) fluids based on differential calculus [6]. Usually, the Navier-Stokes equations are too complicated to be solved in a closed form. However, in some special cases the equations can be simplified and may admit analytical solutions.

The Navier–Stokes equations are very useful because they describe the physics of many different scientific phenomena and are widely used in both science and engineering. Scientists and engineers use the equations in mathematical models of weather, ocean currents, water flow in a pipe, air flow around a wing, drag in race cars, optimizing particle filters, studying environmental particle transport, how stars move inside a galaxy, and much more. The Navier–Stokes equations in their full and simplified forms help with the design of aircraft and cars, the study of blood flow, the design of power stations, the analysis of pollution, and many other things. Together with Maxwell's equations (the equations for electricity and magnetism) they can be used to model and study how things that can flow and conduct electricity can produce (and react to) magnetic fields.

In this work, a code for incompressible [8], viscous flows is developed. It is an example of a simple numerical method for solving the Navier-Stokes equations. It contains fundamental components, such as discretization [9] on a staggered grid, an implicit viscosity step, a projection step, as well as the visualization of the solution over time. The main priorities of the code are 1. Simplicity and compactness: The whole code is one single MATLAB [10] file of about 100 lines. 2. Flexibility: The code does not use spectral methods, thus can be modified to more complex domains, boundary conditions, and flow laws. 3. Visualization: The evolution of the flow field is visualized while the simulation

runs. 4. Computational speed: Full vectorization and pre-solving the arising linear systems in an initialization step results in fast time stepping.
We consider the incompressible Navier-Stokes equations in two space dimensions.

$$u_t + p_x = -(u^2)_x - (uv)_y + \frac{1}{R_e}(u_{xx} + u_{yy}) \quad (1)$$

$$u_t + p_y = -(v^2)_y - (uv)_x + \frac{1}{R_e}(u_{xx} + u_{yy}) \quad (2)$$

$$u_x + v_y = 0 \quad (3)$$

on a rectangular domain Ω = [0, lx]×[0,l_y]. The four domain boundaries are denoted upward, downward, left, and right. The domain is fixed in time, and we consider no-slip boundary conditions on each wall, i.e.

$u(x,l_y) = u_N(x)$	$v(x,l_y) = 0$
$u(x,0) = u_S(x)$	$v(x,0) = 0$
$u(0,y) = 0$	$v(0,y) = v_W(y)$
$u(l_x,y) = 0$	$v(l_x,y) = v_E(y)$

A derivation of the Navier-Stokes equations can be found in [2]. The momentum equations (1) and (2) describe the time evolution of the velocity field (u,v) under inertial and viscous forces. The pressure p is a Lagrange multiplier to satisfy the incompressibility condition (3). Note that the momentum equations are already put into a numerics-friendly form. The nonlinear terms on the right-hand side equal

$$(u^2)_x + (uv)_y = uu_x + vu_y \quad (4)$$

$$(uv)_x + (v^2)_y = uv_x + vv_y \quad (5)$$

which follows by the chain rule and equation (3). The above right hand side is often written in vector form as $(\mathbf{u} \cdot \nabla)\mathbf{u}$. We choose to numerically discretize the form on the left-hand side, because it is closer to a conservation form.
The incompressibility condition is not a time evolution equation, but an algebraic condition. We incorporate this condition by using a projection approach [1]: Evolve the momentum equations neglecting the pressure, then project onto the subspace of divergence-free velocity fields.

NUMERICAL SOLUTION APPROACH

While u, v, p and q are the solutions to the Navier-Stokes equations, we denote the numerical approximations by capital letters. Assume we have the velocity field U^n and V^n at the n^{th} time step (time t), and condition (3) is satisfied. We find the solution at the $(n + 1)^{st}$ time step (time $t + \Delta t$) by the following three step approach:

1. **Treat nonlinear terms**

The nonlinear terms are treated explicitly. This circumvents the solution of a nonlinear system but introduces a CFL condition which limits the time step by a constant times the spacial resolution.

$$\frac{U^* - U^n}{\Delta t} = -((u^n)^2)_x - (U^n V^n)_y \quad (6)$$

$$\frac{V^* - V^n}{\Delta t} = -(U^n V^n)_x - ((V^n)^2)_y \quad (7)$$

2. **Implicit viscosity**

The viscosity terms are treated implicitly. If they were treated explicitly, we would have a time step restriction proportional to the spatial discretization squared. We have no such limitation for the implicit treatment. The price to pay is two linear systems to be solved in each time step.

$$\frac{U^{**} - U^*}{\Delta t} = \frac{1}{R_e}(U^{xx}_{xx} - U^{xx}_{yy}) \quad (8)$$

$$\frac{V^{**} - V^*}{\Delta t} = \frac{1}{R_e}(V^{xx}_{xx} - V^{xx}_{yy}) \quad (9)$$

1. **Pressure correction**

We correct the intermediate velocity field (U^{**}, V^{**}) by the gradient of a pressure P^{n+1} to enforce incompressibility.

$$\frac{\mathrm{U}^{n+1} - \mathrm{U}^{x*}}{\Delta \mathrm{t}} = -(\mathrm{P}^{n+1})_{\mathrm{x}} \quad (10)$$

$$\frac{V^{n+1} - V^{x*}}{\Delta t} = -(P^{n+1})_x \quad (11)$$

The pressure is denoted P^{n+1} since it is only given implicitly. It is obtained by solving a linear system. In vector notation the correction equations read as

$$\frac{1}{\Delta t}U^{n+1} - \frac{1}{\Delta t}U^n = -\nabla P^{n+1} \quad (12)$$

Applying the divergence to both sides yield the linear system:

$$-\nabla P^{n+1} = -\frac{1}{\Delta t}\nabla . U^n \quad (13)$$

DISCRETIZATION

The spacial discretization is performed on a staggered grid with the pressure *P* in the cell midpoints, the velocities *U* placed on the vertical cell interfaces, and the velocities *V* placed on the horizontal cell interfaces. The stream function *Q* is defined on the cell corners.

Consider to have $n_x \times n_y$ cells. Figure 1 shows a staggered grid with $n_x = 5$ and $n_y = 3$. When speaking of the fields *P*, *U* and *V* (and *Q*), care must be taken about interior and boundary points. Any point truly inside the domain is an interior point, while points on or outside boundaries are boundary points. Dark markers in Figure 1 stand for interior points, while light markers represent boundary points. The fields have the following sizes:

Table -1 Boundary points

Field quantity	Interior resolution	Resolution with boundary points
pressure *P*	$nx \times ny$	$(n_x + 2) \times (n_y + 2)$
velocity component *U*	$(n_x - 1) \times n_y$	$(n_x + 1) \times (n_y + 2)$
velocity component *V*	$n_x \times (n_y - 1)$	$(n_x + 2) \times (n_y + 1)$
stream function *Q*	$(n_x - 1) \times (n_y - 1)$	$(n_x + 1) \times (n_y + 1)$

The values at boundary points are no unknown variables. For Dirichlet boundary conditions they are prescribed, and for Neumann boundary conditions they can be expressed in term of interior points. However, boundary points of *U* and *V* are used for the finite difference approximation of the nonlinear advection terms. Note that the boundary points in the four corners are never used.

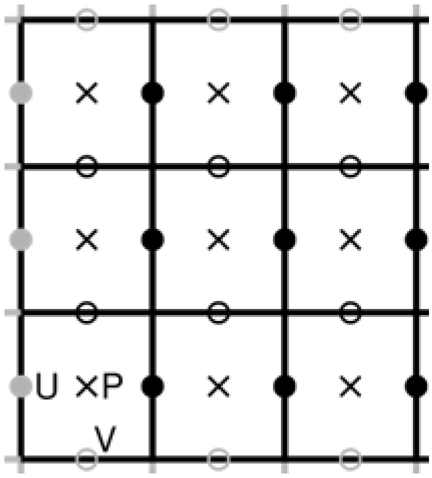

Fig. 1 Discrete model with boundary cells

Finite differences can approximate second derivatives in a grid point by a centered stencil. At an interior point Ui,j we approximate the Laplace operator by

$$\Delta U_{i,j} = ((U_{xx})_{i,j} + (U_{yy})_{i,j}) \approx \frac{U_{i-1,j} - 2U_{i,j} + U_{i+1,j}}{h_x^2} + \frac{U_{i,j-1} - 2U_{i,j} + U_{i,j+1}}{h_y^2}$$

$$\Delta U_{i,j} = ((U_{xx})_{i,j} + (U_{yy})_{i,j}) \approx \frac{U_{i+1,j} - U_{i,j}}{h_x}$$

The first derivative is applied on the corrected pression and the differentiating is applied for the non linear term.

The pressure correction and the implicit discretization of the viscosity terms requires linear systems to be solved in every time step. Additionally, the computation of the stream function requires another system to be solved whenever the data is plotted. Since neither geometry nor discretization change with time, the corresponding system matrices remain the same in every step. This means that all matrices can be constructed in an initialization step. Of course, one would even wish to compute the inverse matrices in an initialization step, but even for medium grid resolutions these could not be saved, since they are full matrices. Of the many possible approaches to do at least some work in the initialization, we propose the following three:

- Use Fourier methods based on the fast Fourier transform in the solution step. Initialize memory and constants in the setup phase.
- Compute good preconditioners in the setup phase. Candidates are ILU or multigrid. Save the preconditioners as sparse matrices and use them in the solution phase.
- Use elimination with reordering to compute the inverse matrices exactly, but in a comparably sparse format. In the code the third approach is implemented. Since the matrices are symmetric positive definite, the sparse Cholesky decomposition can be used.

RESULTS AND DISCUSION

We take advantage of the MATLAB data structures and save the field quantities as matrices. Each quantity is stored without boundary points, yielding matrices of the following sizes.

Table -2 Matrix size

Quantity	Matrix size
U	$(n_x-1).(n_y+1)$
V	$(n_x+1).(n_y-1)$
P	$n_x.n_y$
V	$(n_x-1).(n_y-1)$

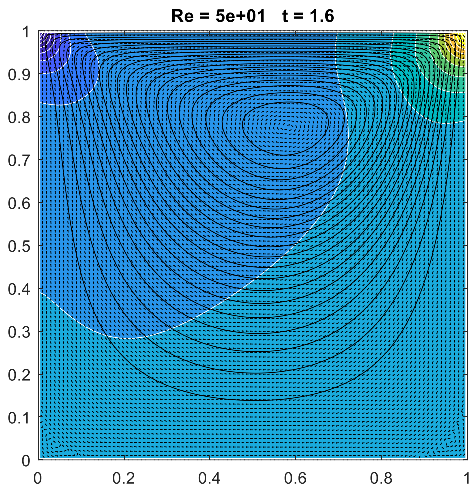

Fig. 2 Evolution of the velocity field in square domain at t= 1.6 seconds

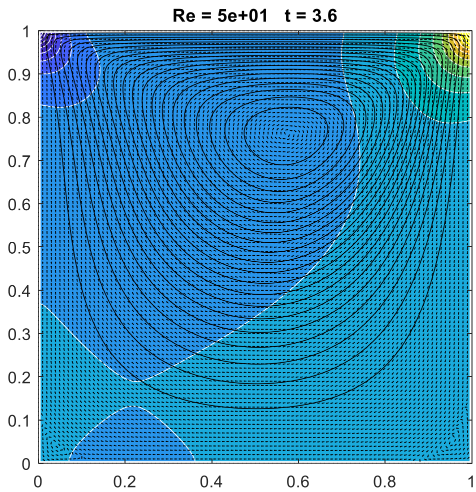

Fig. 3 Evolution of the velocity field in square domain at t= 3.6 seconds

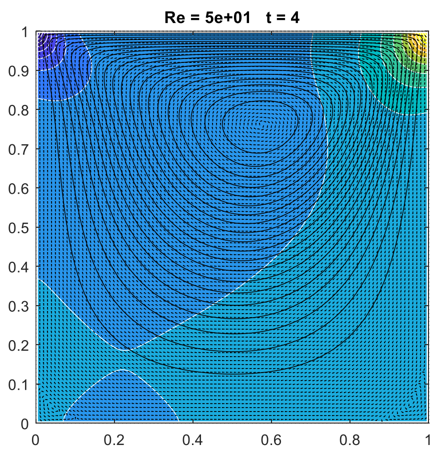

Fig. 4 Evolution of the velocity field in square domain at t= 4 seconds

The simulation is done with the following the parameters.

Re = 0.5e2; Reynolds number
dt = 1e-2; time step
tf = 4e-0; final time
lx = 1; width of box
ly = 1; height of box
nx = 100; number of x-gridpoints
ny = 100; % number of y-gridpoints
nsteps = 10; number of steps with graphic output
Figure 2, figure 3 and figure 4 show the evolution of the velocity field inside the square domain at some specific value time and at constant Reynolds. Number. The results obtained show that the program implemented in MATLAB work. The streamline obtained are shown inside the square but not at wall of the domain.

CONCLUSION

The MATLAB code applied is this work shows correct results that means the numerical resolution of equations and the boundaries conditions applied are stable. The work could be enlarged with others geometrical domain like rectangular cavity, cylinder, or trapezoid.

REFERENCES

[1]. H. Manouzi; Numerical simulations of the Navier-Stokes problem with hyper dissipation; Excerpt from the Proceedings of the COMSOL Multiphysics User's Conference 2005 Boston.

[2]. Salomon F Itza-Ortiz1, Sanjay Rebello and Dean Zollman; Students' Models of Newton's Second Law in Mechanics and Electromagnetism; Institute of Physics Publishing European Journal of Physics Eur. J. Phys. 25 (2004) 81–89.

[3]. Markus Scholle, Florian Marner and Philip H. Gaskell, Potential Fields in Fluid Mechanics: A Review of Two Classical Approaches and Related Recent Advances Water 2020, 12, 1241; doi:10.3390/w12051241.

[4]. Fromm, J. E., The Time Dependent Flow of an Incompressible Viscous Fluid, Meth. Comput. Phys., 3, 345-382 (1964).

[5]. R Marsitin; Analysis of Differential Calculus in Economics; The 1st International Conference on Engineering and Applied Science Journal of Physics: Conference Series 1381 (2019) 012003 IOP Publishing doi:10.1088/1742-6596/1381/1/012003.

[6]. Shuangling Donga, Songping Wub; A modified Navier-Stokes equation for incompressible fluid flow; 7th International Conference on Fluid Mechanics, ICFM7; Procedia Engineering 126 (2015) 169 – 173.

[7]. Papa Touty Traore, M.S. Ould Brahim, Youssou Traore, Alassane Ba, Dame Diao, Seydou Faye, Issa Diagne, and Gregoire Sissoko; Determination of the thermal diffusivity to tow plaster by numerical method: Influence the Biot number and the heat exchange coefficient in transient regime; International Journal of Innovation and Applied Studies ISSN 2028-9324 Vol. 22 No. 4 Mar. 2018, pp. 275-281. http://www.ijias.issr-journals.org/

[8]. T. Aditya Sai Srinvias, A. David Donald, M. Sameena, K. Rekha, I. Dwaraka Srihith; Unlocking the Power of Matlab: A Comprehensive Survey; International Journal of Advanced Research in Science, Communication and Technology (IJARSCT) Volume 3, Issue 1, April 2023; ISSN (Online) 2581-9429.

[9]. A.J. Chorin, J.E. Marsden, A mathematical introduction to fluid mechanics. Third edition, Springer, 2000.

[10]. A.J. Chorin, Numerical solution of the Navier-Stokes equations. Math. Comput. 22, 745–762, 1968.