Introduction to Numeric Simulation of Magnetic Fluid Flows

Ataias Pereira Reis, Yuri Dumaresq Sobral, Francisco Ricardo da Cunha
Universidade de Brasília
Departamento de Matemática
Brasília, Brasil
ataiasreis@gmail.com

***Abstract*—This work aims at studying how magnetic fluids behave in a 2D lid-driven cavity. In order to achieve this goal, a computer program was developed to solve the Navier Stokes equation using finite differences. In addition to that, models of the magnetostatic force were introduced. Solving the Poisson equation is done by means of an implicit method with sparse matrices and Cholesky factorization. Once the hydrodynamic and magnetostatic parts are solved, vector fields and streamlines are plotted for analysis of the flow. A few distinct Reynolds numbers are tested, as well as some different magnetic parameters. The governing equations in continuous form and also in discrete form are presented, as well as the vectors fields resulted from computations.**

***Index Terms*—Magnetic fluid, Driven Cavity, Finite differences, Magnetohydrodynamics**

I. Introduction

This project aims at studying magnetic fluids in a 2D lid-driven cavity. The ideal cavity used here is a square of side equal to unity. All the boundaries but the top are with null velocities. The lid has a stationary velocity pattern that will be presented. Even though this may seem a simplistic method, papers have been presented not only in the last century but also in this century to study real problems. Garandet [2] uses a similar approach to study solute segregations. On the field of biomagnetic fluids, Tzirtzilakis [6] has also considered a cavity and analysed the stationary solutions under different conditions of the Reynolds numbers and magnetic parameters. Differently from this last researcher, our project has not considered a method to find directly the steady state, but has evolved in discrete time steps from time 0 to an end time defined. The advantage is that more data can be analysed, so that it is possible to know when vortices start to appear and if there is any that disappear. Animations were made in order to investigate this.

A colloidal magnetic fluid, or ferrofluid, consists typically of a suspension of monodomain ferromagnetic particles such as magnetite in a nonmagnetic carrier fluid. Particle-to-particle agglomeration is avoided by surfactants covering the particles, and Brownian motion prevents particle sedimentation in gravitational or magnetic fields. [5]. A magnetic fluid has usually around 10^{23} particles per cubic meter and is opaque at visible light.

The method chosen to solve the differential equations is finite differences [4]. This method approximates each derivative by discrete terms involving differences. The domain in which the problem is to be solved must be turned into a mesh. The smaller the cells of the mesh, the better the finite differences solution will be. All the finite differences used here are of second order, which means that the error decays quadratically as the mesh size is lowered. The mesh that is being used is not the trivial one, but a staggered grid. In such a grid, not every information is stored in the same point. For cell ij, the pressure will be in position $i + \frac{1}{2}j + \frac{1}{2}$, while velocity in the x direction will be $u_{ij+\frac{1}{2}}$ and the y direction will be $v_{i+\frac{1}{2}j}$. This avoids spurious pressure modes [3]. A cell of this grid is presented in Figure 1. An example of the mesh is presented in Figure 2. The simulation will have to use all the round points presented, and almost all the velocities marked, except the extreme left and bottom ones. After the solution is obtained, an interpolation can be made to obtain the values at a specified point. It was decided to save the results in the middle point of each cell. The pressure is already in that position, but the velocities are not. Therefore, averages were computed to obtain those values in the middle point.

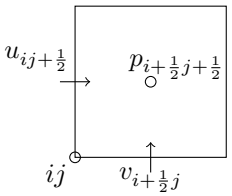

Fig. 1. Cell of the staggered grid

The code for the simulator was made with aid of programming language Julia[1]. This language offers several linear algebra packages that were easy to use. Graphics were made using Python's graphical packages. Codes and articles are hosted on GitHub[2]. In the Julia language, matrices are indexed from 1 to n. This is different from C, because its indexing starts from 0.

[1] http://julialang.org
[2] https://github.com/ataias/ferrofluidos

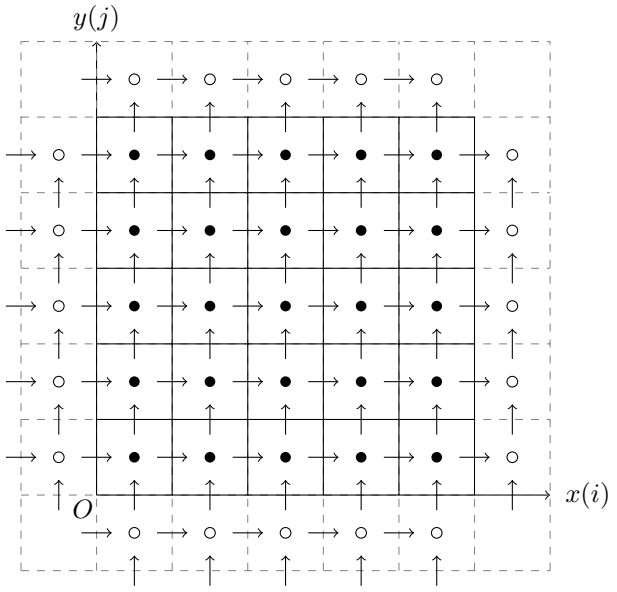

Fig. 2. Staggered-grid. Dark circles are internal points, while empty ones are part of the extra layer

II. Governing Equations

A. Hidrodynamics

Equations 1 and 2 govern the flow of an incompressible fluid. The first one is the Navier Stokes equation, while the second one is the condition for incompressibility.

$$\rho\left(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v}\cdot\nabla\mathbf{v}\right) = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f} \quad (1)$$

$$\nabla\cdot\mathbf{v} = 0 \quad (2)$$

Computer solutions require certain conditions on the time step and spacing of grid points. The requirements are: stable diffusion, Equation 3; stable advection, Equation 4 and hydrodynamic boundary layer, Equation 5. U is the maximum speed in the cavity. If those equations are not satisfied, the solution will be wrong as the program may be trying to solve faster than the laws of physics allow. The program may not even converge and stay on an infinite loop.

$$\Delta t < \frac{1}{4}Re\Delta x^2 \quad (3)$$

$$\Delta t < \frac{\Delta x}{U} \quad (4)$$

$$\Delta x < \frac{1}{Re} \quad (5)$$

The Navier Stokes equation is thorough, encompassing several types of flow. A specific solution is found by specifying its boundary conditions. For the current work, the conditions are

$$u(x,1) = \sin^2(\pi x), \quad (6)$$

$$u(x,0) = u(0,y) = u(1,y) = 0 \quad (7)$$

$$v(x,0) = v(x,1) = v(0,y) = v(1,y) = 0 \quad (8)$$

B. Magnetism

Besides the hydrodynamics equations and conditions, the magnetic force - **f** - must be computed. From Maxwell's theory, Equations 9 and 10 are obtained.

$$\mathbf{B} = \mu_0(\mathbf{M}+\mathbf{H}) \quad (9)$$

$$\nabla\cdot\mathbf{B} = 0 \quad (10)$$

From the latter equations, it is possible to conclude that

$$\nabla\cdot\mathbf{M} = -\nabla\cdot\mathbf{H}. \quad (11)$$

The magnetostatic regime implies $\nabla\times\mathbf{H} = \mathbf{0}$ which indicates there is a potential field ϕ that satisfies $\mathbf{H} = -\nabla\phi$. With these last equations, it is possible to obtain Equation 12. This is a Poisson equation that will give ϕ according the the applied field **H**. The superparamagnetic case is being considered, what implies $\mathbf{M} = \chi\mathbf{H}$.

$$\nabla^2\phi = \nabla\cdot\mathbf{M} \quad (12)$$

Nevertheless, the potential ϕ cannot be introduced directly in the Navier Stokes equation. What is needed is a relation that gives the magnetic force, which could then be incorporated in the fluid equations. Equation 13 stands for the magnetic force, while Equation 14 is the magnetic reynolds number.

$$\mathbf{f} = \mathrm{Cpm}\mathbf{M}\cdot\nabla\mathbf{H} \quad (13)$$

$$\mathrm{Cpm} = \frac{\mu_0 H_0^2}{\rho u^2} \quad (14)$$

One thing that is missing is the boundary conditions for the potential field ϕ. To understand how they are derived, first consider an applied magnetic field **H**. A condition that is imposed is continuity of the normal component of **B** - **B** can be obtained by Equation 9. This continuity can be mathematically represented by Equation 15. From this, Equation 16 can be obtained.

$$(\mathbf{B}_{\mathrm{out}} - \mathbf{B}_{\mathrm{in}})\cdot\mathbf{n} = 0 \quad (15)$$

$$\mu_0(H^{\mathrm{n}}_{\mathrm{out}} + M^{\mathrm{n}}_{\mathrm{out}}) = \mu_0(H^{\mathrm{n}}_{\mathrm{in}} + M^{\mathrm{n}}_{\mathrm{in}}) \quad (16)$$

The medium outside is air and therefore its magnetization, $M^{\mathrm{n}}_{\mathrm{out}}$ is zero. It is possible to use $\mathbf{H} = -\nabla\phi$ in the remaining outer term to obtain Equation 17.

$$\nabla\phi^{\mathrm{n}}_{\mathrm{d}} = -H^{\mathrm{n}}_{\mathrm{f}} + M^{\mathrm{n}}_{\mathrm{d}} \quad (17)$$

From that, Neumann boundary conditions can be obtained. For the sake of clarity, they are presented in Equations 18 to 21. The subscripts are indicating matrix indices.

$$\left.\frac{\partial\phi}{\partial x}\right|_{\text{left}} \approx -H^x_{2,j} + M^x_{2,j} \quad (18)$$

$$\left.\frac{\partial\phi}{\partial x}\right|_{\text{right}} \approx -H^x_{n,j} + M^x_{n,j} \quad (19)$$

$$\left.\frac{\partial\phi}{\partial y}\right|_{\text{lower}} \approx -H^y_{i,2} + M^y_{i,2} \quad (20)$$

$$\left.\frac{\partial\phi}{\partial y}\right|_{\text{upper}} \approx -H^y_{i,n} + M^y_{i,n} \quad (21)$$

III. Solution Method

In order to solve the flow with a computer, the latter equations should be expanded to its 2D form and then discretized. Nevertheless, the fluid flow equation is highly non-linear due to its convective term - $\mathbf{v}\cdot\nabla\mathbf{v}$. Another problem is that the pressure needed at each time step is not known and need to be solved. The pressure is needed to know the velocity and the velocity is needed to obtain the pressure. To solve this dilemma, the Chorin method [1] was used. This method is the time splitting. To understand this method, one can start by using finite differences for the velocity derivative with respect to time and then doing a smart manipulation with two new terms. This is presented in Equation 22.

$$\frac{\partial\mathbf{v}}{\partial t} \approx \frac{\mathbf{v}^{n+1}-\mathbf{v}^n}{\Delta t} = \frac{\mathbf{v}^{n+1}-\mathbf{v}^*}{\Delta t} + \frac{\mathbf{v}^*-\mathbf{v}^n}{\Delta t} \quad (22)$$

The idea of time-splitting is to impose each parcel of Equation 22 to a different part of the Navier Stokes equation. The last term in the right hand side is used to solve Equation , in which pressure is not considered, while the term just before it is used to obtain the next time step from the pressure calculated with $\mathbf{v}^*$.

$$\frac{\mathbf{v}^*-\mathbf{v}^n}{\Delta t} = \nu\nabla^2\mathbf{v} + \frac{\mathbf{f}}{\rho} - \mathbf{v}\cdot\nabla\mathbf{v} \quad (23)$$

$$\frac{\mathbf{v}^{n+1}-\mathbf{v}^*}{\Delta t} = -\frac{1}{\rho}\nabla p \quad (24)$$

Pelo fato deste primeiro método abordado ser iterativo, com um *loop* sendo executado até a convergência ser obtida, é natural escolher um método de parada. Isto poderia ser: (1) escolher um tempo limite de execução, (2) ver a diferença entre duas matrizes após uma iteração completa em seus pontos internos, para analisar se a diferença entre elas é desprezível e se indica convergência, ou ainda (3) pode-se calcular a Equação **??** em cada ponto interno, que seria o melhor a se fazer, e analisar se a equação está dentro de uma faixa de erro desejada. No Algoritmo 1, é apresentada a formulação inicial utilizada.

O conjunto das Equações **??** a **??** é um sistema linear, e é possível representá-lo na forma matricial $Ax = b$. O que torna este problema interessante é que a matriz A e o vetor b tem padrões muito bem definidos. A matriz A segue o padrão apresentado na Equação 25.

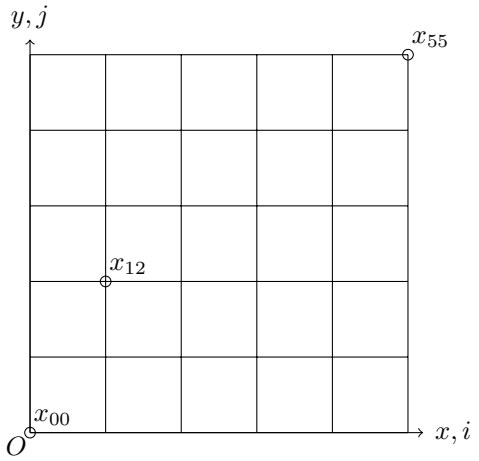

Fig. 3. Malha padrão para discretizar Laplace e Poisson

input : Matriz $n \times n$ com as condições de contorno de Dirichlet
output: Matriz com a solução da equação de Laplace

```
STOP = 0;
err ← 10^−6;
while STOP ≠ 1 do
    u^old ← u;
    for i ← 1 to n − 2 do
        for j ← 1 to n − 2 do
            u_ij ← (u_{i+1j} + u_{i−1j} + u_{ij+1} + u_{ij−1})/4;
        end
    end
    ERROR ← Norma(u^old − u);
    if ERROR < err then
        STOP ← 1;
    end
end
```

Algorithm 1: Resolvendo a equação de Laplace Iterativamente

$$\begin{pmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{pmatrix} \quad (25)$$

Observando o padrão de formação da matriz A pentadiagonal e também o padrão do vetor b, obtém-se um sistema linear cuja incógnita são os pontos internos x. A solução é $x = A^{-1}b$.

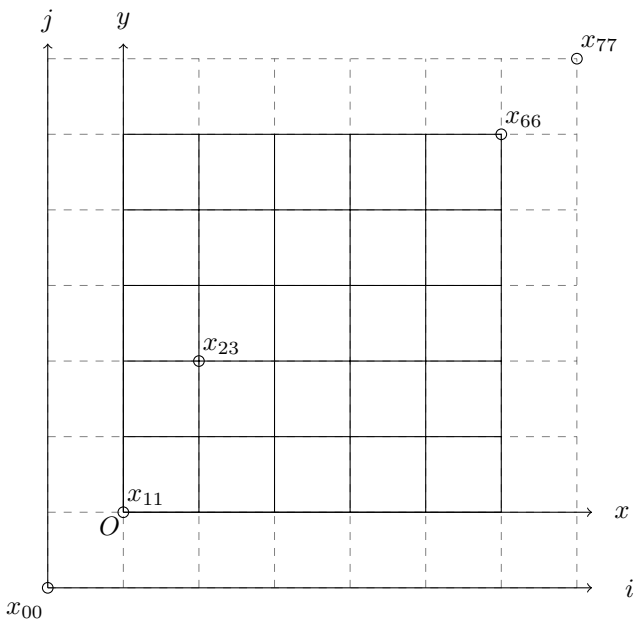

Fig. 4. Mostra dos índices dos pontos do domínio e pontos imaginários

Como se nota, a resolução é direta, mas o algoritmo para se resolver este sistema linear pode ser muito mais complexo. Mas há um problema para este método, que um sistema pode ficar extremamente grande com um n relativamente pequeno, tão grande, ao ponto de necessitar de mais memória do que o computador tem. Isso se resolve utilizando matrizes esparsas, que só salvam na memória elementos diferentes de zero. A matriz A é uma matriz esparsa, na qual a maioria de seus elementos é igual a 0. Utilizaram-se métodos prontos para resolução de sistemas esparsos da biblioteca Eigen. A Figura **??** mostra a relação entre tempo e número de pontos para o método direto. Note que o tempo é muito menor que nos métodos iterativos, nas Figuras **??** e **??**.

IV. Results

V. Conclusion

O problema proposto inicialmente ainda não foi alcançado, que é trabalhar com a instabilidade de Saffman-Taylor com fluido magnético e, ou Anisotrópico. Antes de trabalhar com tal problema, teve-se uma etapa inicial que demandou um certo trabalho. Essa primeira etapa consistiu do estudo da resolução de equações diferenciais parciais, por meio de sua discretização e codificação em um programa de computador.

Foi bem sucedida a resolução das equações de Laplace e Poisson numa malha quadrada, pelos métodos direto e iterativo, com diferenças finitas. No caso da equação de Poisson, ainda teve-se a resolução bem sucedida com condições de fronteira de Neumann, que seria essencial para trabalhar com a pressão na equação de Navier Stokes.

O trabalho foi um aprendizado razoavelmente complexo, e que teve muitas consequências positivas. As habilidades de programação de programas numéricos pelo aluno aumentaram bastante, na realidade, de qualquer programa, pelo fato de se ter treinado bastante o uso do git e da ferramenta de compilação cmake. Apesar da programação ainda ser um problema, pela complexidade do programa, o código[3] está muito mais organizado do que se esperava, e pra isso ele foi refeito algumas vezes.

O programa para resolver a equação de Navier Stokes na malha escalonada está sendo implementado e espera-se obter resultados nas etapas posteriores deste projeto.

Acknowledgments

Primeiramente agradeço a Deus pela oportunidade de ter feito parte deste trabalho. Agradeço também ao professor Yuri Dumaresq pelas orientações e ânimo no ensino que eu vi nele e que também me animaram na resolução destes problemas numéricos. Agradeço também ao CNPq pela bolsa de incentivo à pesquisa.

References

[1] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135(2):118 – 125, 1997.

[2] J.P. Garandet, N. Kaupp, D. Pelletier, and Y. Delannoy. Solute segregation in a lid driven cavity: Effect of the flow on the boundary layer thickness and solute segregation. *Journal of Crystal Growth*, 340(1):149 – 155, 2012.

[3] E.J. Hinch. Lecture notes on computational methods in fluid dynamics: Part i - a first problem., 2006.

[4] L.M. Milne-Thomson. The calculus of finite differences. London: Macmillan & Co., Ltd. XIX, 558 S., 23 Fig. (1933)., 1933.

[5] R E Rosensweig. Magnetic fluids. *Annual Review of Fluid Mechanics*, 19(1):437–461, 1987.

[6] E.E. Tzirtzilakis and M.A. Xenos. Biomagnetic fluid flow in a driven cavity. *Meccanica*, 48(1):187–200, 2013.

[3] Código está acessível online em https://github.com/ataias/ff