

Introduction to Numeric Simulation of Magnetic Fluid Flows

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

Abstract—This work aims at studying the behavior of magnetic fluids 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, different equations for the evolution of the magnetism of the magnetic fluid were introduced. The governing equations in continuous form and also in discrete form are presented, as well as the dimensionless parameters governing the problem. The Navier-Stokes equations are solved using a projection method to decouple the velocity pressure problem. The solution of the Poisson equation is obtained 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. We observe that the magnetic field strongly influences the flow on the cavity.

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

I. INTRODUCTION

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 [9]. Magnetic fluids have usually around 10^{23} magnetic nano-particles per cubic meter and are opaque at visible light since most of them are oil-based.

This project aims at studying magnetic fluids in a 2D lid-driven cavity. The cavity studied in this work is a square. The lid has a stationary velocity and all the boundaries have null velocities. Even though this may seem a simple problem, papers have been presented not only in the last century but also in the current to study real problems: Garandet [4] uses a similar approach to study solute segregations; on the field of biomagnetic fluids, Tzirtzilakis [10] has also considered a cavity and analyzed the stationary solutions under different conditions of the Reynolds numbers and magnetic parameters. Differently from this last work, our project has not considered a method to find directly the steady-state flow, but has evolved in discrete time steps from rest until the steady

state is reached. The advantage is that more data can be analyzed, so that it is possible to know when vortices start to appear and if there are any that disappear during the evolution to steady-state.

The differential equations governing this problem will be solved using finite differences [7], a method that 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 number of points in the mesh increases. The mesh that is being used is a staggered grid.

II. GOVERNING EQUATIONS

A. Hidrodynamics

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

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

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

In the latter equations, some terms are presented: \mathbf{v} is the velocity, p stands for pressure and \mathbf{f} is an external force acting on the fluid, which is, in our case, created by a magnetic field. The Reynolds number Re is given by

$$Re = \frac{\rho LU}{\mu}, \quad (3)$$

in which ρ is the density of the fluid, L is a characteristic linear dimension, in our case, the size of the cavity, U is maximum speed of the lid and μ is the dynamic viscosity of the fluid.

Computer solutions require certain conditions on the time step, Δt , and spacing of grid points. As well described on [5], the requirements are: stable diffusion,

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

stable advection, the CFL condition,

$$\Delta t < \frac{\Delta x}{U}, \quad (5)$$

and to resolve well the hydrodynamic boundary layer,

$$\Delta x < \frac{1}{Re}. \quad (6)$$

If the time-stepping conditions are not satisfied, the routine will try to solve the flow faster than the laws of physics allow. The result is an incorrect solution or divergence of the solution.

A specific solution of the Navier-Stokes equation is found by stating its boundary and initial conditions. For the current work, the boundary conditions are

$$u(x, 1) = \sin^2(\pi x), \quad (7)$$

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

$$v(x, 0) = v(x, 1) = v(0, y) = v(1, y) = 0 \quad (9)$$

and we start from rest. Note that Equation 7 was used to prevent corner stress singularities.

B. Magnetism

Besides the hydrodynamics equations and conditions, the magnetic force \mathbf{f} in Equation 1 must be computed. From Maxwell's equations,

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

and, for a magnetizable medium, we have

$$\mathbf{B} = (\mathbf{M} + \mathbf{H}), \quad (11)$$

in its dimensionless form. In those equations, \mathbf{B} denotes the magnetic induction, \mathbf{M} the magnetization of the fluid and \mathbf{H} the magnetic field. From the latter equations, it is possible to conclude that

$$\nabla \cdot \mathbf{H} = -\nabla \cdot \mathbf{M}. \quad (12)$$

The magnetostatic regime of the Maxwell's equations implies

$$\nabla \times \mathbf{H} = \mathbf{0}, \quad (13)$$

which indicates there is a potential field ϕ that satisfies

$$\mathbf{H} = -\nabla \phi. \quad (14)$$

With these last equations, it is possible to obtain

$$\nabla^2 \phi = \nabla \cdot \mathbf{M}. \quad (15)$$

This is a Poisson equation that will give ϕ according to the local magnetization of the ferrofluid. In this work, the superparamagnetic case is being considered, what implies

$$\mathbf{M} = \chi \mathbf{H}. \quad (16)$$

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. The magnetic force is

$$\mathbf{f} = C_{pm} \mathbf{M} \cdot \nabla \mathbf{H}, \quad (17)$$

in which C_{pm} , the magnetic pressure coefficient for the superparamagnetic case, is

$$C_{pm} = \frac{\mu_0 H_0^2}{\rho u^2}. \quad (18)$$

The boundary conditions were not yet determined for the potential field ϕ . To derive those conditions, an applied magnetic field \mathbf{H} was considered and the requirement of continuity of normal components of the magnetic induction \mathbf{B} across the boundaries, Equation 11. This continuity can be mathematically represented by

$$(\mathbf{B}_{out} - \mathbf{B}_{in}) \cdot \mathbf{n} = 0. \quad (19)$$

From Equations 11 and 19, we have

$$(H_{out}^n + M_{out}^n) = (H_{in}^n + M_{in}^n). \quad (20)$$

The medium outside is air and therefore its magnetization M_{out}^n is zero. The term H_{out}^n is substituted following Equation 14, resulting in:

$$\nabla \phi_d^n = -H_{out}^n + M_{in}^n. \quad (21)$$

From that, four Neumann boundary conditions can be obtained for Equation 15:

$$\left. \frac{\partial \phi}{\partial x} \right|_{x=0} \approx -H_{2,j}^x + M_{2,j}^x, \quad (22)$$

$$\left. \frac{\partial \phi}{\partial x} \right|_{x=1} \approx -H_{n,j}^x + M_{n,j}^x, \quad (23)$$

$$\left. \frac{\partial \phi}{\partial y} \right|_{y=0} \approx -H_{i,2}^y + M_{i,2}^y, \quad (24)$$

$$\left. \frac{\partial \phi}{\partial y} \right|_{y=1} \approx -H_{i,n}^y + M_{i,n}^y; \quad (25)$$

in which the subscripts are indicating matrix indices.

III. NUMERICAL SOLUTION OF THE GOVERNING EQUATIONS

A. Staggered-grid

As stated before, an staggered-grid is used to eliminate spurious pressure modes. In such a grid, not every information is stored in the same point. For instance, on the 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 as described in [5]. A cell of this grid is presented in Figure 1, while the mesh is presented in Figure 2. After the solution is obtained, an interpolation can be made to obtain the values at any specified point. We have chosen to plot all 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. The discretizations of the following sections all consider a staggered-grid.

The code for the simulator was made with aid of programming language Julia, more information about this language is available at [3]. Julia offers several linear algebra packages that are easy to use and very efficient.

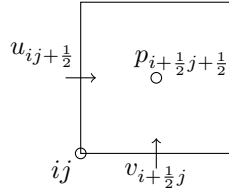


Fig. 1. Cell of the staggered grid

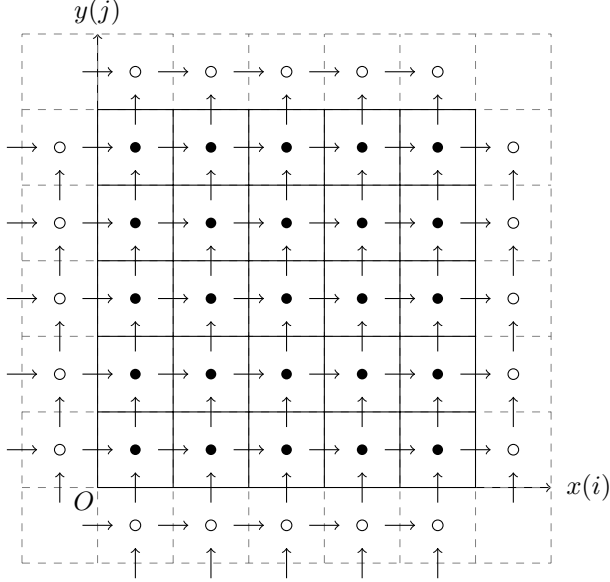


Fig. 2. Staggered-grid. Dark circles are internal points, while empty ones are part of the extra layer

Graphics were made using Python's graphical packages [6]. Codes and articles of this project are hosted on GitHub [8].

B. Time-splitting

In order to discretize the Navier-Stokes equation, an expansion is performed in directions x and y to obtain scalar equations and then finite differences of second order are applied:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f_x, \quad (26)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) + f_y. \quad (27)$$

Nevertheless, a naive application of the discretized formulas will not yield a proper solution, as the fluid flow equation is highly non-linear due to its convective term $\mathbf{v} \cdot \nabla \mathbf{v}$. Also, pressure is not known in advance so, while it is needed at each time step, it is also an unknown that must be found. A method developed by Chorin, named time-splitting, and described in [2] was used to resolve this problem. Firstly, one starts by applying finite differences for the velocity derivative with respect to time and then manipulating by summing and subtracting an arbitrary

quantity \mathbf{v}^* in the numerator, as in

$$\frac{\partial \mathbf{v}}{\partial t} \approx \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} = \frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} + \frac{\mathbf{v}^* - \mathbf{v}^n}{\Delta t}, \quad (28)$$

in which \mathbf{v}^n is the velocity at the current time-step and \mathbf{v}^{n+1} is the velocity one time-step in the future.

The aforementioned technique imposes on the first term of the right hand side of Equation 28 a relation with the pressure field:

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} = -\nabla p. \quad (29)$$

Note that Equation 29 already shows a relation between \mathbf{v}^{n+1} and \mathbf{v}^* , a correlation that is pertinent because it is used to obtain the velocity in the next time step after pressure is calculated.

The second term on the RHS of Equation 28 is responsible for all the terms of Equation 1, except the pressure gradient term that is already considered on Equation 29:

$$\frac{\mathbf{v}^* - \mathbf{v}^n}{\Delta t} = \frac{1}{Re} \nabla^2 \mathbf{v} + \mathbf{f} - \mathbf{v} \cdot \nabla \mathbf{v} \quad (30)$$

The term \mathbf{v}^* can be isolated in Equation 30 and then calculated easily as only known terms from time step n are involved in the aforementioned equation. Nevertheless, only time was discretized so far. Before any computation, we need to apply finite differences to Equation 30, which results in

$$\begin{aligned} u_{ij}^* &= u_{ij}^n + \Delta t \left[\frac{1}{Re} \left(\frac{u_{ij}^s - 4u_{ij}^n}{\Delta x^2} \right) + Fx_{ij}^n \right] + \\ &- [u_{ij}^n(u_{i+1j}^n - u_{i-1j}^n) + \\ &+ v_{ij}^n(u_{ij+1}^n - u_{ij-1}^n)] \frac{\Delta t}{2\Delta x} \text{ and} \end{aligned} \quad (31)$$

$$\begin{aligned} v_{ij}^* &= v_{ij}^n + \Delta t \left[\frac{1}{Re} \left(\frac{v_{ij}^s - 4v_{ij}^n}{\Delta x^2} \right) + Fy_{ij}^n \right] \\ &- [u_{ij}^n(v_{i+1j}^n - v_{i-1j}^n) + \\ &+ v_{ij}^n(v_{ij+1}^n - v_{ij-1}^n)] \frac{\Delta t}{2\Delta x}. \end{aligned} \quad (32)$$

In the latter equations, u_{ij}^* is a velocity in the x -direction at row i and column j in the mesh, while v_{ij}^* is a velocity in the y -direction with similar indexing. It is important to notice that those equations have truncated the $1/2$ that would appear when the value is stored in the middle of the cell, or at the middle of a wall. In order to the computations to be valid, the values involved must be related to the same physical point. The terms u_{ij}^t and v_{ij}^t are interpolated velocities that need to be defined in order for the computations to be correct in the staggered-grid and are defined as:

$$u_{ij}^t = \frac{1}{4}(u_{ij}^n + u_{i+1j}^n + u_{i-1j-1}^n + u_{ij-1}^n), \quad (33)$$

$$v_{ij}^t = \frac{1}{4}(v_{ij}^n + v_{i-1j}^n + v_{i-1j+1}^n + v_{ij+1}^n). \quad (34)$$

The terms u_{ij}^s and v_{ij}^s refer to a sum of cell points around the corresponding velocity located in ij :

$$u_{ij}^s = u_{i+1j}^n + u_{i-1j}^n + u_{ij+1}^n + u_{ij-1}^n, \quad (35)$$

$$v_{ij}^s = v_{i+1j}^n + v_{i-1j}^n + v_{ij+1}^n + v_{ij-1}^n. \quad (36)$$

As a means to calculate p , the divergence operator is applied to both sides of Equation 29:

$$-\nabla \cdot \nabla p = \nabla \cdot \left(\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} \right). \quad (37)$$

As \mathbf{v}^{n+1} has to obey the incompressibility condition, its divergence is zero, with the remaining equation being transformed into a Poisson equation:

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^*. \quad (38)$$

In order to solve Equation 38, its boundary conditions need to be determined. Equations 26 and 27 are valid in the whole domain, including the walls. To obtain the boundary conditions for p , these equations are evaluated according to Equations 7 to 9, which results in:

$$\left. \frac{\partial p}{\partial x} \right|_{\text{wall}} = \frac{1}{Re} \frac{\partial^2 u}{\partial x^2} \Big|_{\text{wall}} + f_x, \quad (39)$$

$$\left. \frac{\partial p}{\partial y} \right|_{\text{wall}} = \frac{1}{Re} \frac{\partial^2 v}{\partial y^2} \Big|_{\text{wall}} + f_y. \quad (40)$$

Equation 39 gives conditions for the left and right boundaries, while Equation 40 is for top and bottom. As the four walls have flux conditions, Neumann conditions, there are infinitely many solutions that differ only by a constant. Therefore, a point in the mesh is specified arbitrarily so that the program converges to a specific solution. This is allowed in our problem, as pressure itself is not directly needed, but only its gradient.

At this stage, \mathbf{v}^* and p are known, so that \mathbf{v}^{n+1} can be calculated by

$$\mathbf{v}^{n+1} = \mathbf{v}^* - \Delta t \cdot \nabla p. \quad (41)$$

Equation 41 is then used to evolve the system from time $n\Delta t$ to $(n+1)\Delta t$.

C. Poisson equations

A pressure must be found that satisfies Equation 38, which is a Poisson equation. The approach to that will be, again, finite differences, but following an implicit approach. Codes that implement explicit difference equations were tested against the implicit ones with the result being that the implicit is much faster. For the implicit case, a linear system must be formed from the difference equations and then the system must be solved to find the pressure. This means that matrix A and vector b must be determined and then system $Ax = b$ must be solved to find x which, in our case, contains pressure values. This system gets very large, but most of the elements in matrix A are 0, so A is a sparse matrix.

Obtaining A and b can be done by first analyzing a small grid and then generalizing the result for an $n \times n$ grid. First, a 5×5 grid was considered, as presented in Figure 3. The equation to be solved is

$$\nabla^2 p = \frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} = g(x, y), \quad (42)$$

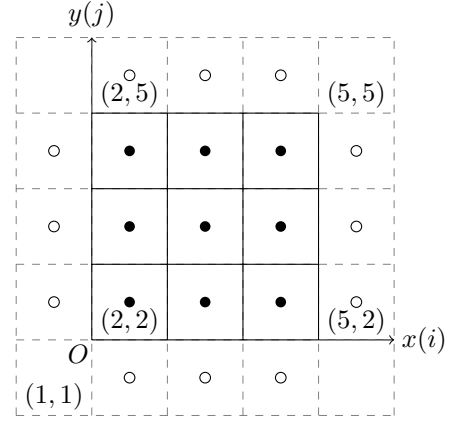


Fig. 3. Example grid to solve for pressure

in which $g(x, y)$ is any known function of x and y . Applying second order central differences to this equation results in

$$\frac{p_{i-1j} - 2p_{ij} + p_{i+1j}}{\Delta x^2} + \frac{p_{i,j-1} - 2p_{ij} + p_{i,j+1}}{\Delta x^2} = g_{ij}, \quad (43)$$

which is a valid equation for internal points (the black dots in Figure 3). Isolating term p_{ij} from it results in an explicit equation for the internal points. Explicit equations for the external points are obtained from their boundary conditions. Nevertheless, what is needed is an implicit representation of the system. For that, all the equations, without isolating terms, need to be considered and then organized in matrix form, like

$$-4p_{22} + p_{32} + \mathbf{p}_{12} + p_{23} + \mathbf{p}_{21} = \Delta x^2 g_{22}, \quad (44)$$

$$-4p_{23} + p_{33} + \mathbf{p}_{13} + p_{24} + p_{22} = \Delta x^2 g_{23}, \quad (45)$$

$$-4p_{24} + p_{34} + \mathbf{p}_{14} + \mathbf{p}_{25} + p_{23} = \Delta x^2 g_{24}. \quad (46)$$

The latter equations are three of nine equations that are needed for the internal points of the grid. The rest of the equations can be obtained similarly. Boundary points are marked in bold and equations for them come exactly from boundary conditions. There are twelve external points, what requires the same number of equations, summing up to a total of 21 equations in a 5×5 grid. The four corner points are not taken into account, as they are not used in the grid style adopted, what justifies that only 21 points are used of 25. Equations 47 to 50 can have its indices substituted in order to get the 12 equations for the boundary points:

$$\frac{\partial p}{\partial x}(0, y) = p_x(0, y) \approx \frac{(p_{2j} - p_{1j})}{\Delta x}, \quad (47)$$

$$\frac{\partial p}{\partial x}(1, y) = p_x(1, y) \approx \frac{(u_{5j} - u_{4j})}{\Delta x}, \quad (48)$$

$$\frac{\partial p}{\partial y}(x, 0) = p_y(x, 0) \approx \frac{(p_{i2} - p_{i1})}{\Delta x}, \quad (49)$$

$$\frac{\partial p}{\partial y}(x, 0) = u_y(x, 0) \approx \frac{(p_{i5} - p_{i4})}{\Delta x}. \quad (50)$$

Arranging appropriately the 21 equations obtained, $A =$

$$\begin{bmatrix} -2 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -3 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -3 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -3 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -3 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -2 \end{bmatrix} \quad (51)$$

and $b =$

$$\Delta x \begin{bmatrix} p_x(0, \frac{\Delta x}{2}) + p_y(\frac{\Delta x}{2}, 0) \\ u_x(0, \frac{3\Delta x}{2}) \\ p_x(0, \frac{5\Delta x}{2}) - p_y(\frac{\Delta x}{2}, 1) \\ p_y(\frac{3\Delta x}{2}, 0) \\ 0 \\ -p_y(\frac{3\Delta x}{2}, 1) \\ -p_x(1, \frac{\Delta x}{2}) + p_y(\frac{5\Delta x}{2}, 0) \\ -u_x(1, \frac{3\Delta x}{2}) \\ -p_x(1, \frac{5\Delta x}{2}) - p_y(\frac{5\Delta x}{2}, 1) \end{bmatrix} + \Delta x^2 \begin{bmatrix} g_{22} \\ g_{23} \\ g_{24} \\ g_{32} \\ g_{33} \\ g_{34} \\ g_{42} \\ g_{43} \\ g_{44} \end{bmatrix}. \quad (52)$$

Even though A and b were obtained, one of the Eigenvalues of this system is zero, so the matrix A is not invertible. The reason for that is related to the fact that there are only Neumann boundary conditions and there are infinitely many solutions that differ only by a constant. To solve this problem, one value of the mesh must be specified and the cell to be chosen can be any but one of the corner points, as they don't play any role on the equations. For our problem, $p_{12} = 0$ was chosen, what results in the update of one element of the matrix and one element of the vector:

$$A[1, 1] = -3, \quad (53)$$

$$b[1] = p_y\left(\frac{\Delta x}{2}, 0\right) \Delta x + \Delta x^2 g_{22}. \quad (54)$$

A generic relation to obtain A and b grids bigger than 5×5 is not hard to obtain. After coding a routine to generate A and b according to $g(x, y)$ and n , a sparse matrix solver like the one available in Julia [3] is used to obtain x . Julia has a package that uses Cholesky factorization to solve the system by just typing $\mathbf{x} = \mathbf{A} \backslash \mathbf{b}$. Nevertheless, this factorizes the matrix every time this line is executed. As matrix A is fixed, the factorization can be done only once and saved in a variable. After that, only the factorized matrix is used as in $\mathbf{x} = \mathbf{A} \text{factorized} \backslash \mathbf{b}$, what is a performance-wise decision.

The magnetism equations must be discretized according to second order finite differences as in this section. The Poisson routine obtained in this section is also used to obtain the potential field ϕ , Equation 15. The discrete formulas are not presented here, but they can be found in the code available on GitHub.

IV. RESULTS

It is important to validate the code. As part of the solution method, tests of the Poisson routine and of the discretization of the Navier-Stokes equation are made. The Poisson test is to check the error when comparing a computed result with the actual analytical function that solves the PDE. The second test applies to the Navier-Stokes equation and also for the Poisson one: verifying how the error decays according to an increase in mesh size. As all the discrete formulas used are of second order, this error should decay quadratically. The Results section will present both of these tests. Also, Navier-Stokes with magnetism will be tested to check the order of the discretization. Animations were made in order to better examine those results. The code to generate those is written in Python and is available at the GitHub repository [8].

A. Validation of the Poisson Solver

The implementation of the solver was evaluated by testing a few different problems, but only a succinct presentation is done here with one problem:

$$\begin{cases} \nabla^2 u(x, y) = 0, \\ u_x(1, y) = \cos 2\pi y, \\ u_n(x, y) = 0, \text{ otherwise, on the boundary.} \end{cases} \quad (55)$$

Its analytical solution is:

$$u(x, y) = \frac{\cosh 2\pi x}{2\pi \sinh 2\pi} \cos 2\pi y. \quad (56)$$

The Poisson validation routine returns the middle point difference on the grid that was found, this value is then divided by the real value on that point and multiplied by 100 to get a percentage. The maximum grid size used was 320×320 . This one had an error smaller than 0.05%. The results for different grid sizes are presented in Figure 4. It is easy to see that the error decays quadratically, what was expected due to the second order finite differences that were used.

$$u(x, y) = \frac{\cosh 2\pi x}{2\pi \sinh 2\pi} \cos 2\pi y \quad (57)$$

B. Validation of the Navier-Stokes routine

Validation for the hydrodynamics is presented on Figure 5. Reynolds is chosen to be 40. The error will be compared against the solution of grid size 200×200 , as access to the analytical solution is not possible. The important point to evaluate is that the error decays quadratically, what can be assessed from the image for simplicity. Vorticity was used instead of velocity because it is a scalar that accounts for both velocity terms. Note that the error corresponds to the base case of 200×200 .

A similar test was used when magnetism is present. This case has a force \mathbf{f} that differs from zero. Magnetic

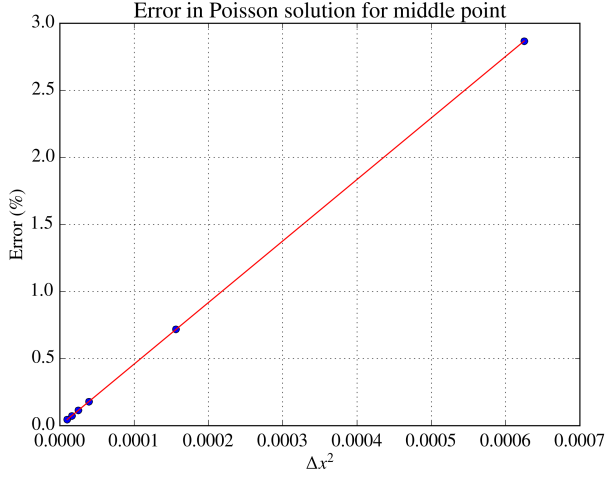


Fig. 4. Error evolving according to grid size. Grid sizes with higher number of points are closer to left-bottom corner. Error decays quadratically. Comparison with analytical solution. Grid sizes from 40×40 to 320×320 .

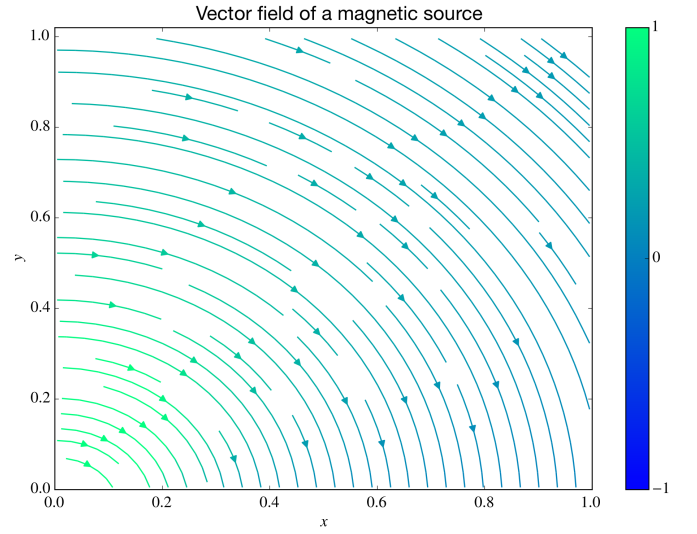


Fig. 6. Applied magnetic field. All tests were made with this specific magnetic field. γ values may change the actual intensity of field, but the streamlines are similar.

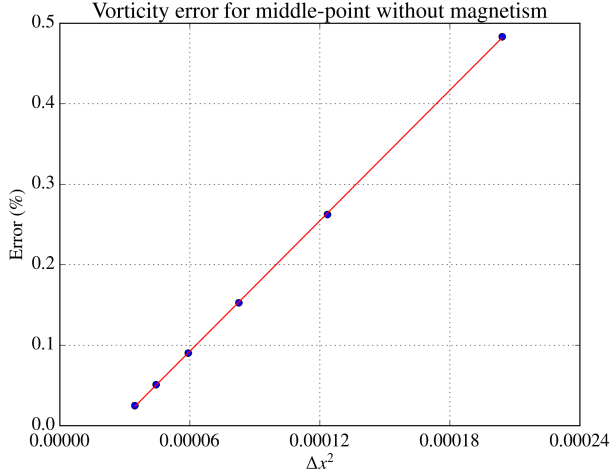


Fig. 5. Error evolving according to grid size. Higher grid sizes are closer to left-bottom corner. Error decays quadratically. Comparison with solution of 200×200 grid. Grid sizes from 70×70 to 170×170 .

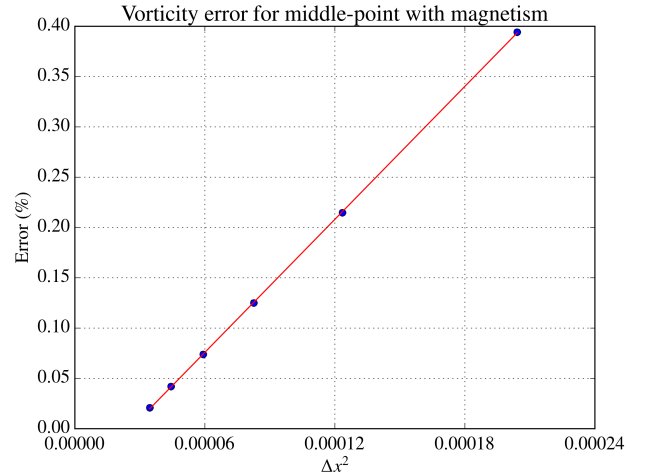


Fig. 7. Error evolving according to grid size. Higher grid sizes are closer to left-bottom corner. Error decays quadratically. Comparison with solution of 200×200 grid. Grid sizes from 70×70 to 170×170 .

parameters chosen are $\chi = 0.5$, $C_{pm} = 0.8$, $\gamma = 3$ and a center for the magnetic field located close the left-bottom corner. The applied magnetic field is that of Figure 6. The error evolution for this case is presented on Figure 7. The values were compared in steady-state.

C. Preliminary studies of flow patterns

We have performed some tests to assess the influence of the dimensionless parameters Re and C_{pm} on the steady-state regimes that were obtained. Some Reynolds numbers were chosen and then several tests were executed for some fixed magnetic parameters. The magnetic parameters are $\chi = 0.5$, $C_{pm} = 0.8$ and $\gamma = 3.5$. Reynolds numbers presented here are 1, 50 and 100. Figures 8 to 19 present streamlines for the velocity and contour lines for the pressure. Images with magnetic parameters set to 0 are

the ones without magnetization. All results are in steady state. We have also coded a routine to calculate the angle between \mathbf{M} and \mathbf{H} for every point in those simulations but, for the case presented here, this is not relevant, as $\mathbf{M} = \chi \mathbf{H}$ is one consideration for the magnetization in the current work (superparamagnetic regime).

Regarding the results with no magnetic field, Figures 8 to 18 present very similar results. Although the Reynolds number changes by a factor of 100, no major changes in the steady state flow were able to be observed. It is noticeable, though, that the flow is more asymmetric and the center of the main vortex shifts to the right as Re increases.

This changes dramatically when a magnetic field is in action. A small vortex already appears in Figure 9 and just grows larger as the Reynolds number increases up to 100 (Figures 11 and 13). Pressure also changes

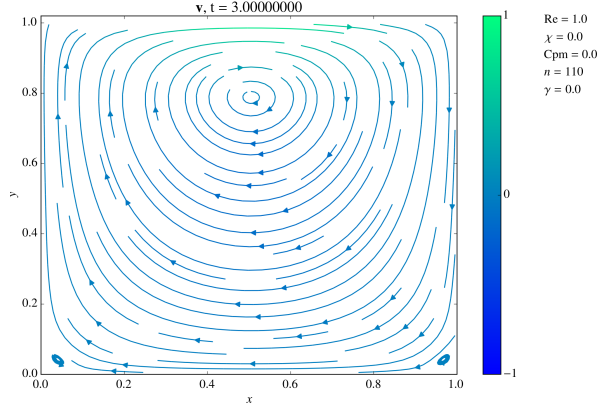


Fig. 8. Steady state solution without magnetic field, $Re = 1$. No extra vortices appear.

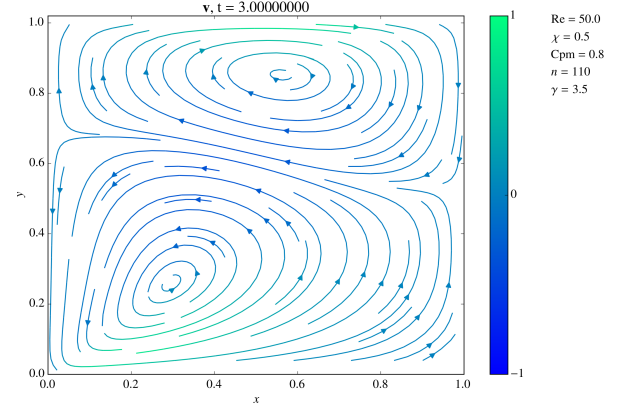


Fig. 11. Steady state solution with magnetic field, $Re = 50$. The new vortex is already bigger than the initial one.

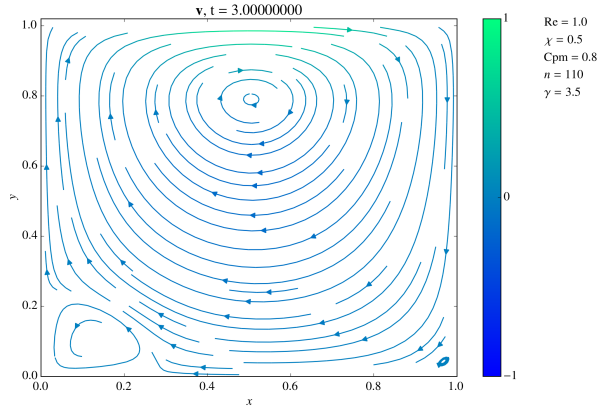


Fig. 9. Steady state solution with magnetic field, $Re = 1$. A new vortex starts to appear.

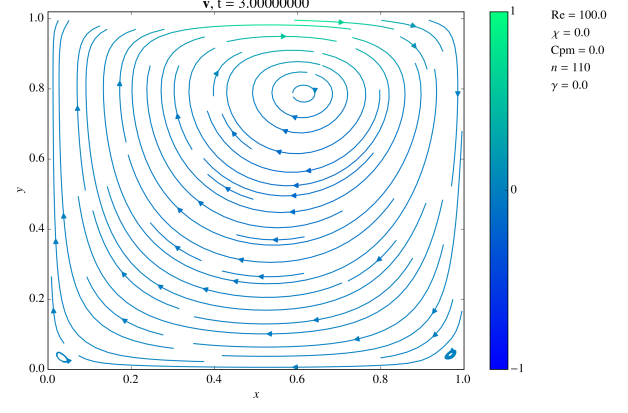


Fig. 12. Steady state solution without magnetic field, $Re = 100$. No extra vortices appear.

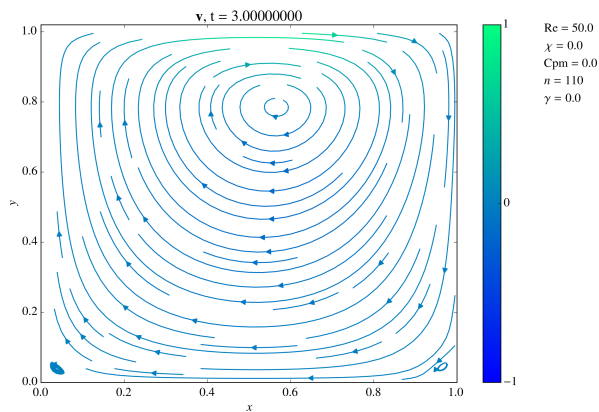


Fig. 10. Steady state solution without magnetic field, $Re = 50$. No extra vortices appear.

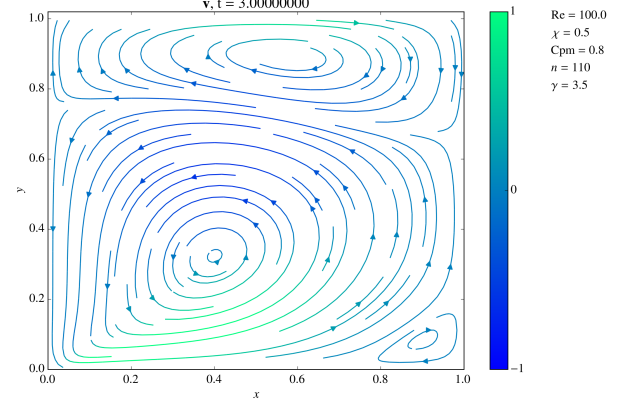


Fig. 13. Steady state solution with magnetic field, $Re = 100$. Vortex increased in comparison with $Re = 50$.

significantly and a particular concentration of contour lines is observable close to the origin (the magnetic field is applied just a little off the origin) as presented in Figures 15, 17 and 19.

V. CONCLUSION

It is important for the code to be correct, otherwise it is meaningless to analyze the simulation results. The validation step shows that certainly the system is behaving

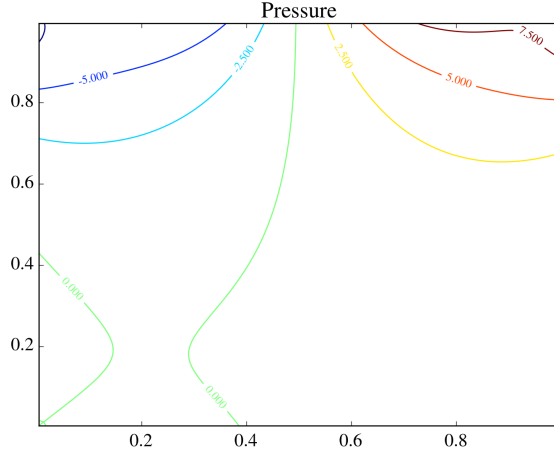


Fig. 14. Contour plot for pressure of the flow in Figure 8, $Re = 1$, no magnetism.

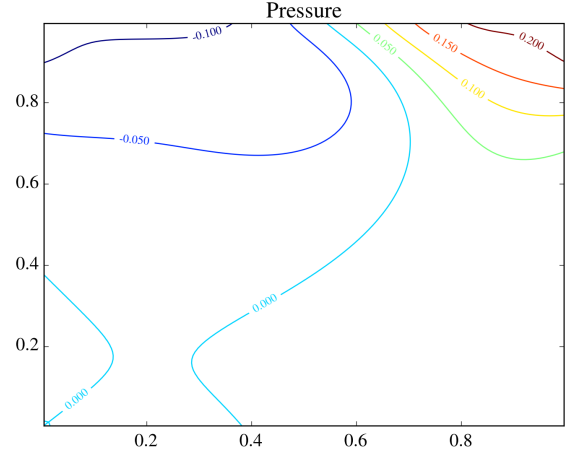


Fig. 16. Contour plot for pressure of the flow in Figure 10, $Re = 50$, no magnetism

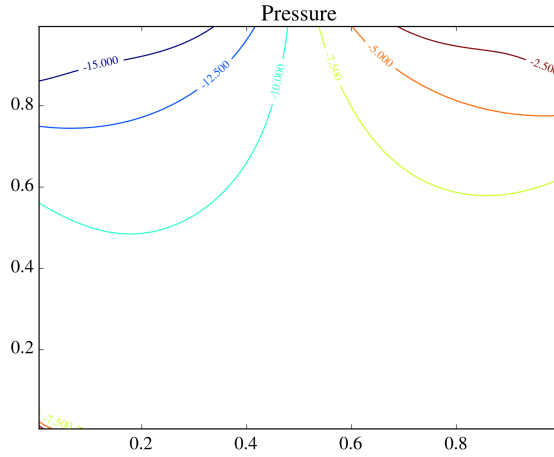


Fig. 15. Contour plot for pressure of the flow in Figure 9, $Re = 1$, with magnetism

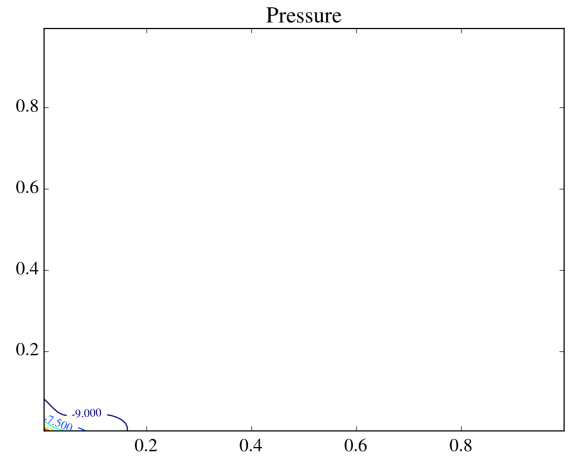


Fig. 17. Contour plot for pressure of the flow in Figure 11, $Re = 50$, with magnetism

as second order.

Although not presented in the last sections, it is interesting to observe that one problem that appeared was when physics was disregarded. Trying to apply an arbitrary magnetic field has a good chance of generating results that converge but are incorrect when the governing equations are checked. The results presented comply to the laws that we claim to be following. This is also good evidence of correctness.

This work was a challenge in the aspect that an analytical solution was not available to check against, so it is important to state the importance of the validation step. The results were satisfactory and the tools here coded for exploring magnetic fluids are starting to get usable, when compared with the stage of one year ago.

VI. FUTURE WORK

The results presented do not include Reynolds greater than 100 because the code is not currently abiding by any

upwind scheme. This is something that is planned to be implemented. Also, a different magnetic field will be used, particularly the one of a Neodymium magnet, instead of the magnetic field of a wire, as used here.

Most importantly, we want to implement the dynamic evolution equations for the magnetization of the ferrofluid that are coupled with the flow field. This will certainly change significantly the dynamics of the flow and will be the subject of our next project.

ACKNOWLEDGMENTS

Firstly, I thank God for this opportunity. I appreciate professor Yuri Dumaesq for his orientations so that this work could be done. Also, many thanks to CNPq for the scholarship.

REFERENCES

- [1] G.K. Batchelor. An Introduction to Fluid Dynamics, 1967.

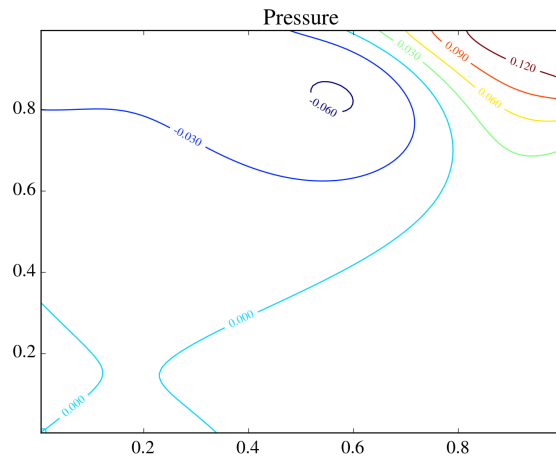


Fig. 18. Contour plot for pressure of the flow in Figure 12, $Re = 100$, no magnetism

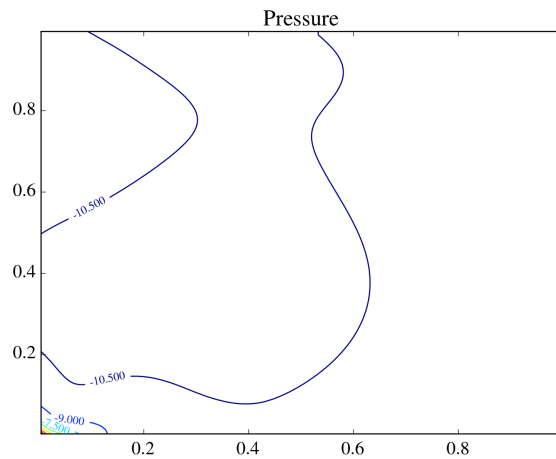


Fig. 19. Contour plot for pressure of the flow in Figure 13, $Re = 100$, with magnetism

- [2] Alexandre Joel Chorin. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics*, 135(2):118 – 125, 1997.
- [3] NumFocus Foundation. The Julia Language. Available at <http://julialang.org>, version 0.3.11.
- [4] 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.
- [5] E.J. Hinch. Lecture notes on computational methods in fluid dynamics: Part i - a first problem., 2006.
- [6] Eric Firing Michael Droettboom John Hunter, Darren Dale and the matplotlib development team. matplotlib: python plotting. Available at <http://matplotlib.org>, version 1.4.3.
- [7] L.M. Milne-Thomson. The calculus of finite differences. London: Macmillan & Co., Ltd. XIX, 558 S., 23 Fig. (1933)., 1933.
- [8] Ataias Pereira Reis. Git repository: ferrofluidos. Available at <https://github.com/ataias/ferrofluidos>.
- [9] R E Rosensweig. Magnetic fluids. *Annual Review of Fluid Mechanics*, 19(1):437–461, 1987.
- [10] E.E. Tzirtzilakis and M.A. Xenos. Biomagnetic fluid flow in a driven cavity. *Meccanica*, 48(1):187–200, 2013.