Mathematical and Computational Modelling of Magnetic Fluids

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 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 magnetization 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. This work focus on the Shliomis equation as a constitutive equation for the magnetism and is still a work in progress.**

***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 [11]. 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 [13] 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 vortical structures 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 [8], 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, but the longer it will take to solve, as there will be more points in which the solution must be computed. All the finite differences used here are of second order in space, which means that the error decays quadratically as the number of points in the mesh increases, and linear in time, which means that the error decays quadratically as the time step is reduced. The mesh that is being used is a staggered grid [5].

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:

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

$$\nabla\cdot\mathbf{v} = 0. \tag{2}$$

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

$$Re = \frac{\rho L U}{\mu}, \tag{3}$$

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

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

$$u(x,1) = \sin^2(\pi x), \quad (4)$$
$$u(x,0) = u(0,y) = u(1,y) = 0, \quad (5)$$
$$v(x,0) = v(x,1) = v(0,y) = v(1,y) = 0 \quad (6)$$

and we start from rest. Note that Equation 4 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,

$$\nabla \cdot \mathbf{B} = 0, \quad (7)$$

and, for a magnetizable medium, we have

$$\mathbf{B} = \mathbf{M} + \mathbf{H}, \quad (8)$$

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

$$\nabla \cdot \mathbf{H} = -\nabla \cdot \mathbf{M}. \quad (9)$$

The magnetostatic regime of the Maxwell's equations implies

$$\nabla \times \mathbf{H} = \mathbf{0}, \quad (10)$$

which indicates there is a potential field ϕ that satisfies

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

With these last equations, it is possible to obtain

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

This is a Poisson equation that will give ϕ according the the local magnetization of the ferrofluid. In our last work [9], the superparamagnetic case was considered, what implies

$$\mathbf{M} = \chi\mathbf{H}. \quad (13)$$

In this work, another constitutive equation is considered, namely the Shliomis equation [12], given by

$$\frac{\partial \mathbf{M}^*}{\partial t^*} = -\frac{1}{\tau}[\mathbf{M}^* - \mathbf{M}_0^*] + \frac{1}{\zeta}[(\mathbf{M}^* \times \mathbf{H}^*) \times \mathbf{M}^*] + + \mathbf{\Omega}^* \times \mathbf{M}^*, \quad (14)$$

in which $\mathbf{\Omega}^* = \frac{1}{2}\nabla \times \mathbf{v}^*$, the macroscopic angular velocity of the fluid, τ is the relaxation time of the magnetisation of the suspension and ζ is the kinetic constant. Our code considers non-dimensional equations, so Equation 14 becomes

$$\frac{\partial \mathbf{M}}{\partial t} = -c_1[\mathbf{M} - \mathbf{M}_0] + c_2[(\mathbf{M} \times \mathbf{H}) \times \mathbf{M}] + + \frac{1}{2}(\nabla \times \mathbf{v}) \times \mathbf{M}, \quad (15)$$

in which $c_1 = \frac{L}{\tau U}$, $c_2 = \frac{LH_0^2}{U\zeta}$ and H_0 is the characteristic magnetic field scale. The definition for $\mathbf{M}_0$ [11] is given by

$$\mathbf{M}_0 = M_S L(\alpha|\mathbf{H}|)\hat{\mathrm{e}}_H, \quad (16)$$

in which M_S is a constant, $L(x) = \coth(x) - \frac{1}{x}$ is the Langevin function, α is a parameter and $\hat{\mathrm{e}}_H$ is a unit vector in the direction of the applied field $\mathbf{H}$, namely

$$\hat{\mathrm{e}}_H = \frac{\mathbf{H}}{|\mathbf{H}|} = \frac{1}{|\mathbf{H}|}\left(H_x\mathbf{i} + H_y\mathbf{j}\right). \quad (17)$$

Given Equations 12 and 15, the potential ϕ is determined, which allows one to compute the magnetic force using

$$\mathbf{f} = \mathrm{C_{pm}}\, \mathbf{M} \cdot \nabla\mathbf{H}, \quad (18)$$

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

$$\mathrm{C_{pm}} = \frac{\mu_0 H_0^2}{\rho U^2}. \quad (19)$$

The boundary conditions were not yet determined for the potential field ϕ. To derive those conditions, an applied magnetic field $\mathbf{H}_0$ was considered and the requirement of continuity of normal components of the magnetic induction $\mathbf{B}$ accross the boundaries, Equation 8. This continuity can be mathematically represented by

$$(\mathbf{B}_{\mathrm{out}} - \mathbf{B}_{\mathrm{in}}) \cdot \mathbf{n} = 0. \quad (20)$$

From Equations 8 and 20, we have

$$(H^{\mathrm{n}}_{\mathrm{out}} + M^{\mathrm{n}}_{\mathrm{out}}) = (H^{\mathrm{n}}_{\mathrm{in}} + M^{\mathrm{n}}_{\mathrm{in}}). \quad (21)$$

The medium outside is air and therefore its magnetization $M^{\mathrm{n}}_{\mathrm{out}}$ is zero. The term $H^{\mathrm{n}}_{\mathrm{out}}$ is substituted following Equation 11, resulting in:

$$\nabla\phi^{\mathrm{n}}_{\mathrm{d}} = -H^{\mathrm{n}}_{\mathrm{out}} + M^{\mathrm{n}}_{\mathrm{in}}. \quad (22)$$

From that, four Neumann boundary conditions can be obtained for Equation 12, one for each wall of the cavity [9].

III. Numerical Solution of the Governing Equations

A. Grid configuration

As stated before, a 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.

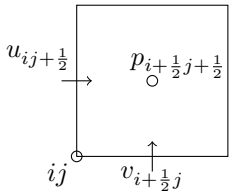

Fig. 1. Cell of the staggered grid

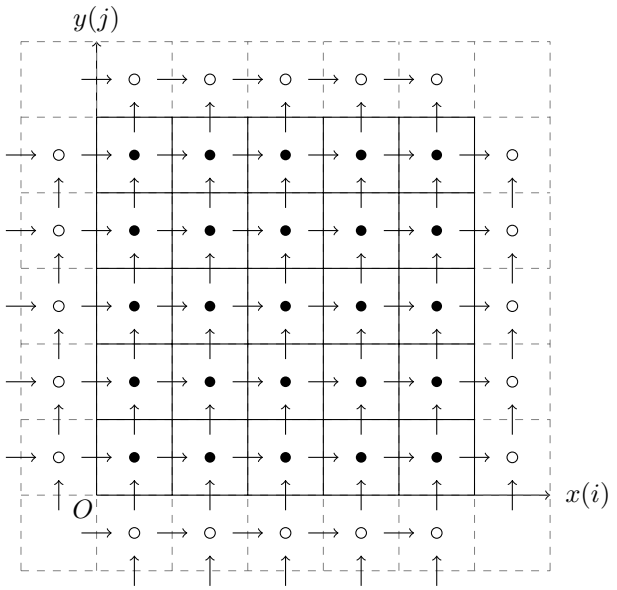

Fig. 2. Staggered-grid. Dark circles are internal points, while empty ones are part of the extra layer

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. Graphics were made using Python's graphical packages [6]. Codes and articles of this project are hosted on GitHub [10].

B. Time-splitting

In order to discretize the Navier-Stokes equation, an expansion is performed in directions x and y to obtain scalar equations:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y}\right) + f_x, \quad (23)$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial^2 x} + \frac{\partial^2 v}{\partial^2 y}\right) + f_y. \quad (24)$$

Finite differences of second order need to be applied to discretize Equations 23 and 24. Nevertheless, a naive application of the discretized formulas will not yield a proper solution, as the fluid flow equation is highly nonlinear 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

$$\frac{\partial \mathbf{v}}{\partial t} \approx \frac{\mathbf{v}^{n+1} - \mathbf{v}^n}{\Delta t} = \frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} + \frac{\mathbf{v}^* - \mathbf{v}^n}{\Delta t}, \quad (25)$$

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 25 a relation with the pressure field:

$$\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} = -\nabla p. \quad (26)$$

Note that Equation 26 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 25 is responsible for all the terms of Equation 1, except the pressure gradient term that is already considered on Equation 26:

$$\frac{\mathbf{v}^* - \mathbf{v}^n}{\Delta t} = \frac{1}{Re}\nabla^2 \mathbf{v}^n + \mathbf{f}^n - \mathbf{v}^n \cdot \nabla \mathbf{v}^n \quad (27)$$

The term $\mathbf{v}^*$ can be isolated in Equation 27 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 27, which results in

$$\begin{aligned} u^*_{ij} &= u^n_{ij} + \Delta t\left[\frac{1}{Re}\left(\frac{u^s_{ij} - 4u^n_{ij}}{\Delta x^2}\right) + Fx^n_{ij}\right] + \\ &- \left[u^n_{ij}(u^n_{i+1j} - u^n_{i-1j}) + \right. \\ &+ \left. v^t_{ij}(u^n_{ij+1} - u^n_{ij-1})\right]\frac{\Delta t}{2\Delta x} \end{aligned} \quad (28)$$

and

$$\begin{aligned} v^*_{ij} &= v_{ij} + \Delta t\left[\frac{1}{Re}\left(\frac{v^s_{ij} - 4v^n_{ij}}{\Delta x^2}\right) + Fy^n_{ij}\right] \\ &- \left[u^t_{ij}(v^n_{i+1j} - v^n_{i-1j}) + \right. \\ &+ \left. v^n_{ij}(v^n_{ij+1} - v^n_{ij-1})\right]\frac{\Delta t}{2\Delta x}. \end{aligned} \quad (29)$$

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^t_{ij} and v^t_{ij} are interpolated velocities that need to be defined in order for the computations to be correct in the staggered-grid and are defined as:

$$u^t_{ij} = \frac{1}{4}(u^n_{ij} + u^n_{i+1j} + u^n_{i+1j-1} + u^n_{ij-1}), \quad (30)$$

$$v^t_{ij} = \frac{1}{4}(v^n_{ij} + v^n_{i-1j} + v^n_{i-1j+1} + v^n_{ij+1}). \quad (31)$$

The terms u^s_{ij} and v^s_{ij} refer to a sum of cell points around the corresponding velocity located in ij:

$$u^s_{ij} = u^n_{i+1j} + u^n_{i-1j} + u^n_{ij+1} + u^n_{ij-1}, \quad (32)$$
$$v^s_{ij} = v^n_{i+1j} + v^n_{i-1j} + v^n_{ij+1} + v^n_{ij-1}. \quad (33)$$

As a means to calculate p, the divergence operator is applied to both sides of Equation 26:

$$-\nabla \cdot \nabla p = \nabla \cdot \left(\frac{\mathbf{v}^{n+1} - \mathbf{v}^*}{\Delta t} \right). \quad (34)$$

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:

$$\nabla^2 p = \frac{1}{\Delta t} \nabla \cdot \mathbf{v}^*. \quad (35)$$

In order to solve Equation 35, its boundary conditions need to be determined. Equations 23 and 24 are valid in the whole domain, including the walls. To obtain the boundary conditions for p, these equations are evaluated according to Equations 4 to 6, which results in:

$$\left.\frac{\partial p}{\partial x}\right|_{\text{wall}} = \left.\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}\right|_{\text{wall}} + f_x, \quad (36)$$
$$\left.\frac{\partial p}{\partial y}\right|_{\text{wall}} = \left.\frac{1}{Re}\frac{\partial^2 v}{\partial y^2}\right|_{\text{wall}} + f_y. \quad (37)$$

Equation 36 gives conditions for the left and right boundaries, while Equation 37 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

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

Equation 38 is then used to evolve the system from time $n\Delta t$ to $(n+1)\Delta t$.

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,

$$\Delta t < \frac{1}{4} Re \Delta x^2, \quad (39)$$

stable advection, the CFL condition,

$$\Delta t < \frac{\Delta x}{U}, \quad (40)$$

and to resolve well the hydrodynamic boundary layer,

$$\Delta x < \frac{1}{Re}. \quad (41)$$

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 divergency of the solution.

C. Poisson equations

A pressure must be found that satisfies Equation 35, 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.

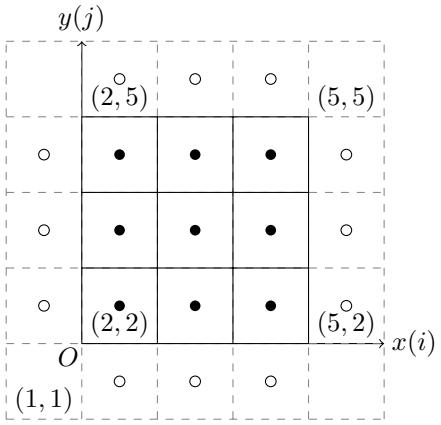

Fig. 3. Example grid to solve for pressure

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

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

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

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

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

$$-4p_{22} + p_{32} + \boldsymbol{p_{12}} + p_{23} + \boldsymbol{p_{21}} = \Delta x^2 g_{22}, \quad (44)$$
$$-4p_{23} + p_{33} + \boldsymbol{p_{13}} + p_{24} + p_{22} = \Delta x^2 g_{23}, \quad (45)$$
$$-4p_{24} + p_{34} + \boldsymbol{p_{14}} + \boldsymbol{p_{25}} + p_{23} = \Delta x^2 g_{24}. \quad (46)$$

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 and b are obtained [9]. Matrix A has an eigenvalue equal to zero, so the system 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, A_{11} and one element of the vector b, namely b_1 [9].

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 `x = A\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 fatorized matrix is used, what is a performance-wise decision.

The magnetization 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 12. The discrete formulas for the constitutive equation 15 are given by

$$\begin{aligned} M_{x,ij}^{n+1} &= M_{x,ij}^{n} - c_1 \Delta t [M_{x,ij} - M_{x_0,ij}] \\ &- c_2 \Delta t M_{y,ij}^{t} (M_{x,ij} H_{y,ij}^{t} - M_{y,ij}^{t} H_{x,ij}) \\ &- \frac{1}{2} \Delta t M_{y,ij}^{t} \left(\frac{v_{ij} + v_{ij+1}}{2\Delta x} - \frac{v_{i-1j} + v_{i-1j+1}}{2\Delta x} \right. \\ &\left. - \frac{u_{ij+1} - u_{ij-1}}{2\Delta x} \right), \end{aligned} \quad (51)$$

$$\begin{aligned} M_{y,ij}^{n+1} &= M_{y,ij}^{n} - c_1 \Delta t [M_{y,ij} - M_{y_0,ij}] \\ &+ c_2 \Delta t M_{x,ij}^{t} (M_{x,ij}^{t} H_{y,ij} - M_{y,ij} H_{x,ij}^{t}) \\ &+ \frac{1}{2} \Delta t M_{x,ij}^{t} \left(\frac{v_{i+1j} - v_{i-1j}}{2\Delta x} - \frac{u_{ij} + u_{i+1j}}{2\Delta x} \right. \\ &\left. + \frac{u_{ij-1} + u_{i+1j-1}}{2\Delta x} \right), \end{aligned} \quad (52)$$

in which the superscript t denotes an average in the staggered grid done so that we multiply values that correspond to the same physical location. This is similar to what the t superscript used in Equations 28 and 29.

IV. Results

As part of the solution method, tests of the Poisson routine and of the discretization of the Navier-Stokes equation were made and presented in [9]. The Results section will present a comparison of the results for the flow without magnetism and with magnetism. 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 [10].

A. Applied magnetic field

The applied magnetic field in dimensionless form is $\mathbf{H} = (H_x, H_y)$, whose components are given by:

$$H_x(x,y) = \frac{1}{2\pi} \frac{y-b}{(x-a)^2 + (y-b)^2}, \quad (53)$$

$$H_y(x,y) = -\frac{1}{2\pi} \frac{x-a}{(x-a)^2 + (y-b)^2}, \quad (54)$$

where (a,b) is the point where the magnetic source is placed [13]. The terms just presented are already dimensionless. The field can have a greater impact on the force depending on the parameter $\mathrm{C_{pm}}$. There parameters a and b were chosen both equal to -0.2. Remember that the cavity has side equal to 1. Figure 4 shows this magnetic field.

B. Preliminary studies of flow patterns

We have performed some tests to assess the influence of the dimensionless parameters Re, $\mathrm{C_{pm}}$ and Pe 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 $\mathrm{C_{pm}} = 10.0$, $\alpha = 5.0$ and $Pe \in [0.01,\ 1]$. The results here are for Reynolds 50.

Figure 5 is the solution without a magnetic field. Figures 6 through 9 present streamlines for the velocity when $Pe = 1$. Example of a force for this case is presented in Figure 10. There is another set of figures for $Pe = 1$, namely Figures 11 and 12.

Regarding the result with no magnetic field, Figure 5 presents a standard result. The flow is more asymmetric and the center of the main vortex shifts to the right as Re increases, as shown in [9]. Here, we will see how the position and number of vortices change according to the magnetic field.

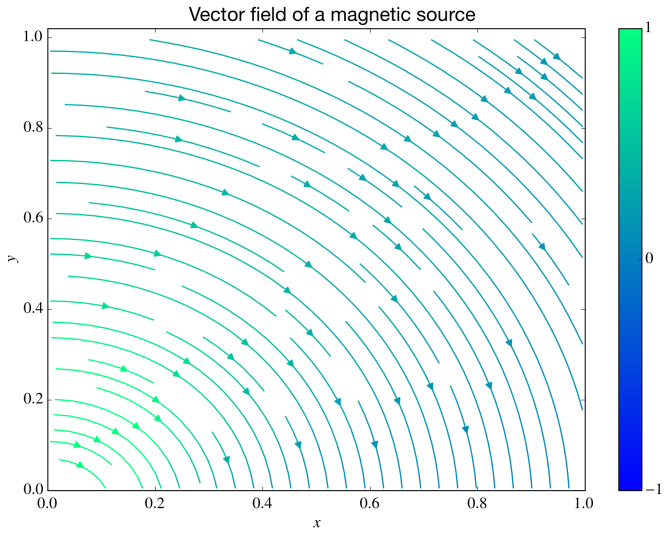

Fig. 4. Applied magnetic field **H**.

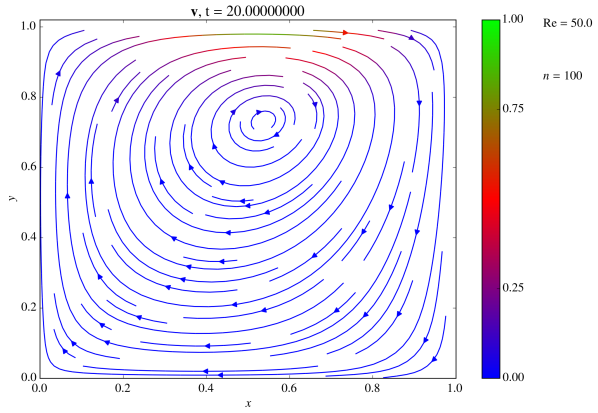

Fig. 5. Solution for $t = 20$ without magnetic field, $Re = 50$. There us only one vortex.

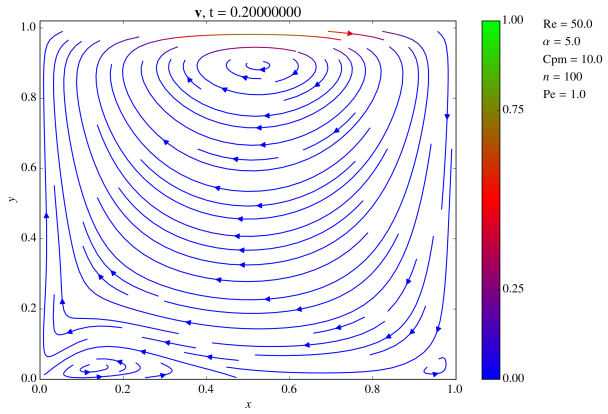

Fig. 6. Solution for $t = 0.2$ with magnetic field, $Re = 50$, $Pe = 1$. A new vortex appears on the bottom.

Figures 6 through 9 show the flow at specific times for $Pe = 1$ so that it is possible to see how the vortices evolve over time. At $t = 0.2$, a vortex appears at the bottom

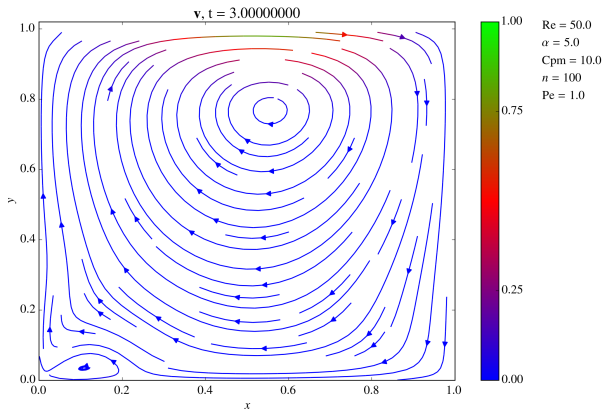

Fig. 7. Solution for $t = 3$ with magnetic field, $Re = 50$, $Pe = 1$. A new vortex appears on the bottom.

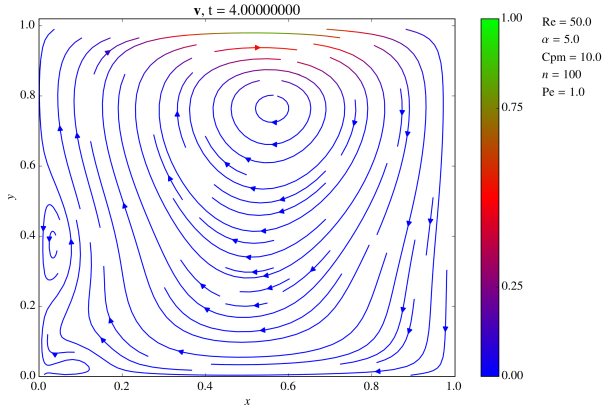

Fig. 8. Solution for $t = 4$ with magnetic field, $Re = 50$, $Pe = 1$. The vortex was advected with the flow to the left wall.

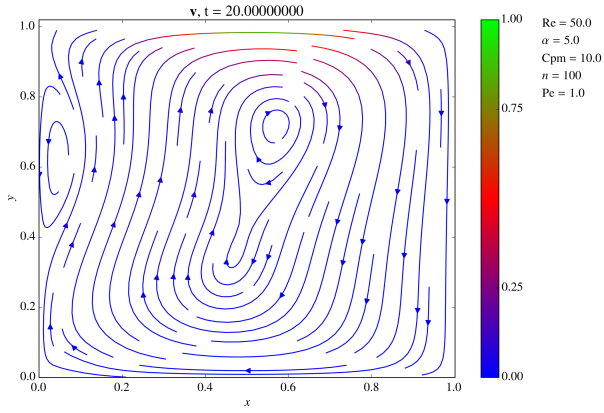

Fig. 9. Solution for $t = 20$ with magnetic field, $Re = 50$, $Pe = 1$. The vortex is allocated at the left wall.

and moves to the left for some time, as one can observe at at $t = 3$. It then moves up, $t = 4$. Up to this point, the main vortex has not changed much. As times passes, both vortices grow in size, as one can see at $t = 20$, Figure 9. Streamlines of the force are presented in Figure 10.

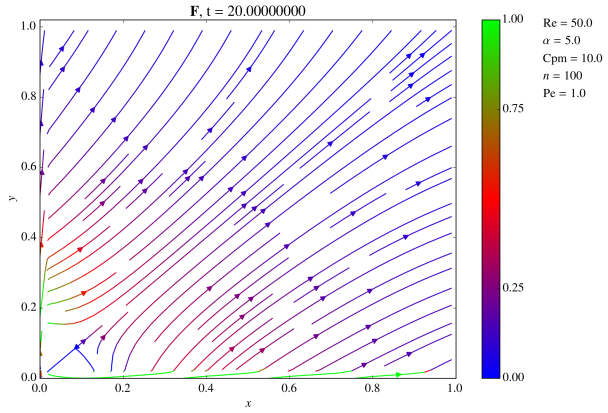

Fig. 10. Magnetic force at $t = 20$, $Re = 50$, $Pe = 1$.

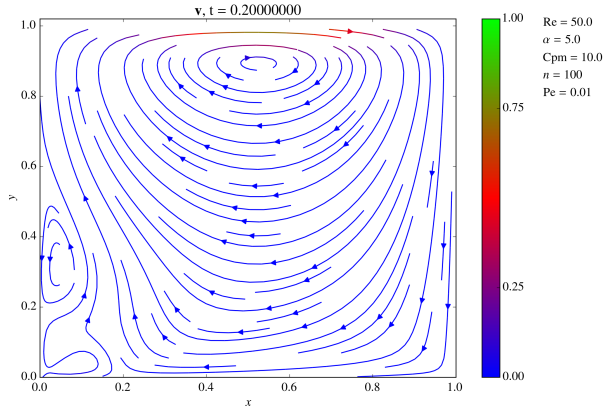

Fig. 11. Solution for $t = 0.2$ with magnetic field, $Re = 50$, $Pe = 100$. The vortex is already at the left wall, at a much earlier time than for $Pe = 1$.

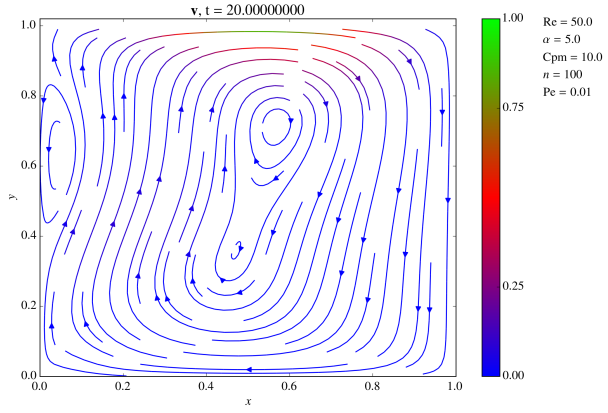

Fig. 12. Solution for $t = 20$ with magnetic field, $Re = 50$, $Pe = 100$. The vortex is at the left wall.

Figures 11 and 12 present the flow for $Pe = 0.01$. The changes due to the magnetic field are much faster to be observed, as $t = 0.2$ already looks like $t = 4$ for $Pe = 1$.

In our last work [9], it was mentioned that we have finalised the step of the construction of the tool that we would use to do detailed study of different models for the evolution of the magnetisation of the ferrofluid in the presence of a flow. The present work is the first study of such studies and it is still a work in progress. There are still some validations that are missing to assure that the Shliomis equations were correctly implemented.

V. Conclusion

The results showed how the vortices in a flow behave when under the influence of a magnetic force. Simulation of the Shliomis equation is not usually done and now there is some code, written in Julia and in the project repository [10], that does that. This was the primary goal and it was accomplished.

An analytical solution is not available for the flow considering the Shliomis equation, so it is important to state the importance of the validation step that was not yet presented. The tools here coded for exploring magnetic fluids are starting to get usable, but still more testing is needed.

VI. Future work

- Implement upwind scheme
- Use magnetic field of Neodymium magnet [7]
- Write code with validations for the Shliomis equations

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

References

[1] G.K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 1967.
[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] A.G. McCaig, M. Clegg. *Permanent Magnets in Theory and Practice*. Wiley, New York, 2 edition, 1987.
[8] L.M. Milne-Thomson. The calculus of finite differences. London: Macmillan & Co., Ltd. XIX, 558 S., 23 Fig. (1933)., 1933.
[9] A.P. Reis, Y.D. Sobral, and F.R. Cunha. Introduction to numerical simulation of magnetic fluids. *Scientific Initiation Program*, 2015.
[10] Ataias Pereira Reis. Git repository: ferrofluidos. Available at https://github.com/ataias/ferrofluidos.
[11] R E Rosensweig. Magnetic fluids. *Annual Review of Fluid Mechanics*, 19(1):437–461, 1987.
[12] M.I. Shliomis. Effective viscosity of magnetic suspensions. *Soviet Physics JETP*, 61(6):2411, 1972.
[13] E.E. Tzirtzilakis and M.A. Xenos. Biomagnetic fluid flow in a driven cavity. *Meccanica*, 48(1):187–200, 2013.