CMIS18 - HW6

Isabela Blucher

June 1, 2018

Introduction

This week we are dealing with the field of magnetostatic. We are given a relation that induces magnetization explicitly and by doing some substitution and cleaning up it is possible to obtain the PDE

$$\nabla \cdot \nabla \phi = \nabla \cdot M$$

which is a poisson equation that we solve for the scalar field ϕ.

To solve this PDE we use the finite volume method (FVM), a discretization method similar to the FDM and FEM introduced previously. Similar to the other methods, the FVM is also used to compute values at discrete places of a meshed geometry. The difference is that we consider small volumes surrounding each node point of the mesh, and thus we derive our linear system based on volume integrals and the Gauss divergence theorem, which converts these terms into surface integrals.

In terms of real-world applications, the FVM is a good discretization technique especially for problems that arise from physical conservation laws. It is commonly used for discretizing computational fluid dynamics equations.

Derivation of the FVM

In this section, we derive the FVM for the given magnetostatisc problem $\nabla \cdot \nabla \phi(x) = \nabla \cdot M(x), \forall x \in \mathcal{A}$, where $\mathcal{A}$ is the closed box $[-3..3] \times [-1..1]$ and $M(x) : \mathbb{R}^2 \to \mathbb{R}^2$ is defined as zero outside the unit disk and $M = (0, -1)^T$ everywhere else.

We start by integrating the PDE over the arbitrary but fixed volume V with surface S for each control volume.

$$\int_V \nabla \cdot \nabla \phi(x)\, dV = \int_V \nabla \cdot M(x)\, dV$$

Using the Gauss divergence theorem, we can transform both sides into surface integrals.

$$\int_S \nabla \phi(x) \cdot n\, dS = \int_S M(x) \cdot n\, dS$$

In the surface integrals, n represents the outward unit normal vector of the control volume. The surface boundary S is piece-wise continuous, so we replace the term with summation over continuous pieces. This means that we can separate the surface integrals into a sum and integrate on each edge of the control volume.

$$\sum_e \int_{S^e} \nabla \phi(x) \cdot n\, dS = \sum_e \int_{S^e} M(x) \cdot n\, dS$$

We can now use the midpoint approximation rule to approximate the defined surface integral. We need to consider that the outward unit normal vector n is constant along each surface element S^e

$$\sum_e [\nabla \phi(x) \cdot n]_e \cdot L_e = \sum_e [M(x) \cdot n]_e \cdot L_e$$

Here we' re computing the values inside the brackets on the midpoint of each surface edge e and L^e is the area of the surface.

It is possible now to apply the FDM to compute the directional derivative for the left term of the equation. With this process we want to assemble the matrices A and b of the linear system $A\phi = b$.

Implementation-wise, what is done to assemble the matrix A is that for each control volume we compute the directional derivatives for each edge (inside and outside of edge) and multiply by the area of the edge surface. To assemble b we see if the midpoint of the edge is inside the unit disk and then act accordingly, multiplying $M(x)$, the outward unit normal vector n of the edge and the surface area of the edge.

Experiments

Original mesh

After implementing the assembly of the linear matrix system described in the section above we can look at some figures. We are given the same triangular mesh on a 6×2 domain, but this time we are dealing with control volumes that have an hexagonal shape. Figure 1 illustrates the mesh layout.

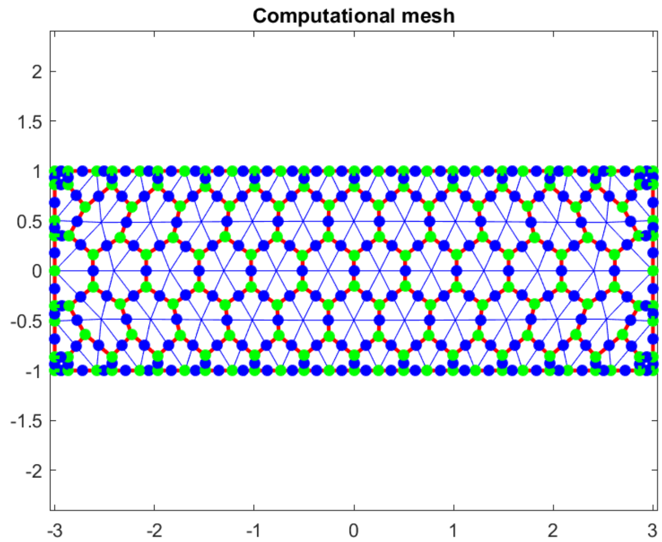
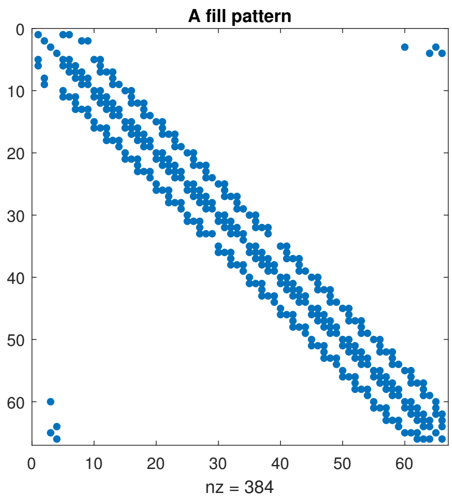
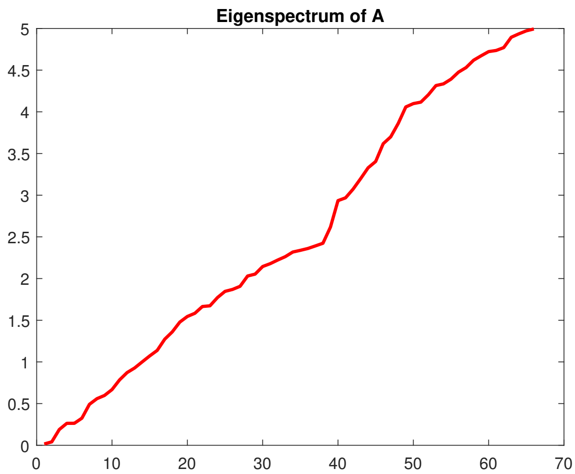


Figure 1: Triangular mesh with control volumes

As we did for the other problems, we can also analyze the fill pattern and eigenspectrum of the matrix A.



(a) Fill pattern of matrix A

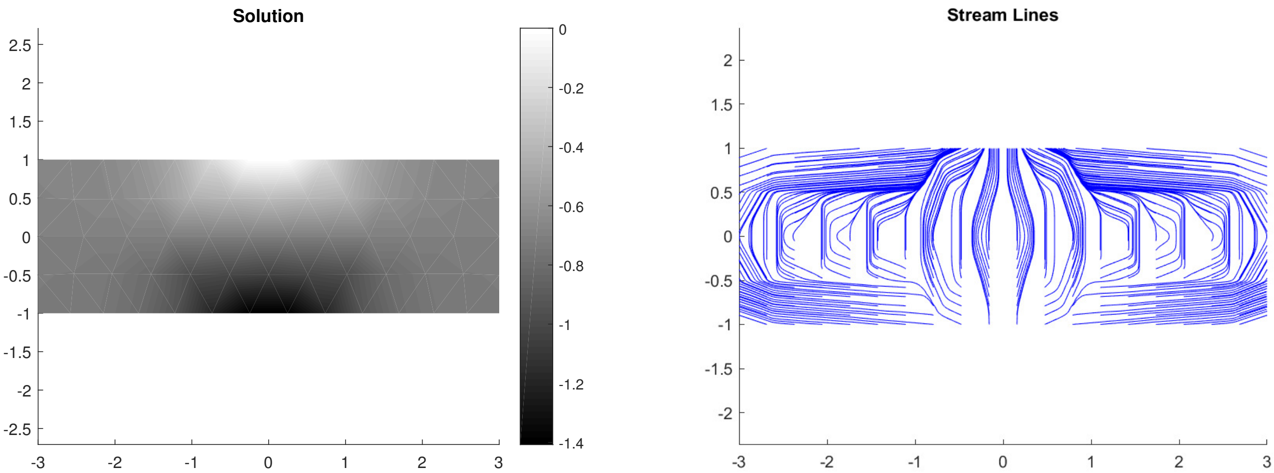


(b) Eigenspectrum of matrix A

Figure 2: Fill pattern and eigenspectrum of A

By looking at the plots in Figure 2 we can see that the assembly process was correctly implemented and matrix A has a regular symmetric appearance which implies that our discretization is correct. As for the eigenvalues, it shows us that since there are no zero eigenvalues the problem is well defined and a solution for it exists.

For the original problem our magnet is unit disk sized, with the north pole on the upper part of the domain. After having computed the field ϕ, we can see how the solution and the magnet's field lines look like.



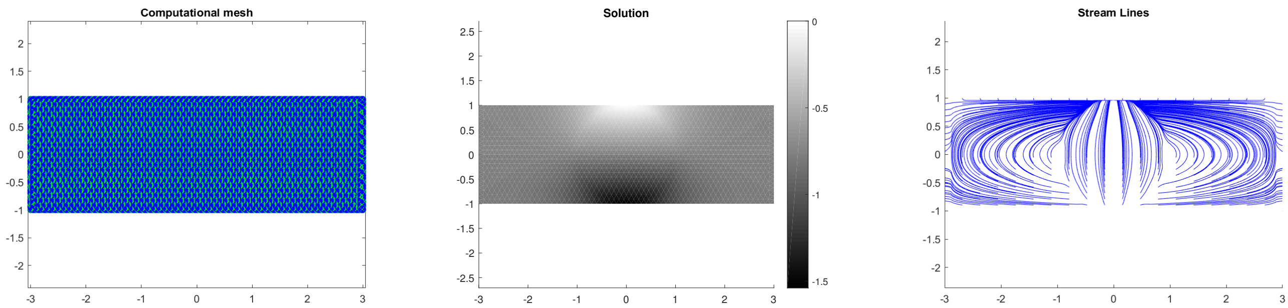
(a) Solution ϕ of linear system (b) Stream lines of magnet

Figure 3: Appearance of ϕ, solution and stream lines

Different meshes

One possible experiment here is looking at different mesh resolutions for meshes generated with DistMesh and seeing how it affects not only the mesh and the control volumes, but also the magnet behavior.

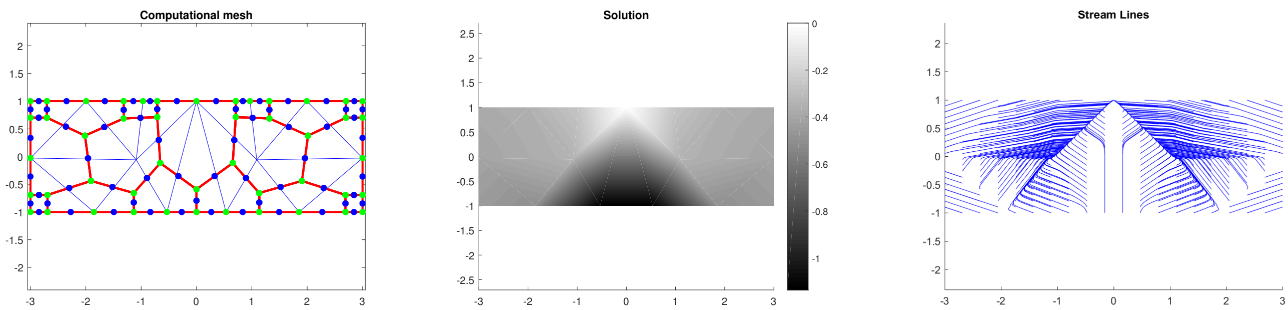
We will try finer and larger mesh resolutions. The resolution of the mesh we are using is 0.5, so we will try 0.1 and 1 as new resolutions and see the changes obtained. We keep the magnet as the unit disk for both resolutions.



(a) Mesh layout with control volumes (b) Solution ϕ of linear system (c) Stream lines of magnet

Figure 4: Plots of magnetostatic problem with 0.1 DistMesh resolution

Figure 4 shows us that a finer mesh resolution gives us more details in our solution and magnetic field lines. The solution shows a very round magnet and the field lines seem to be much smoother than for the original case. The expectation for the larger mesh resolution would be the opposite, where the solution and field lines would get a lot harsher and not really reflect the real-world physics of the problem, due to small amount of triangles on the mesh.



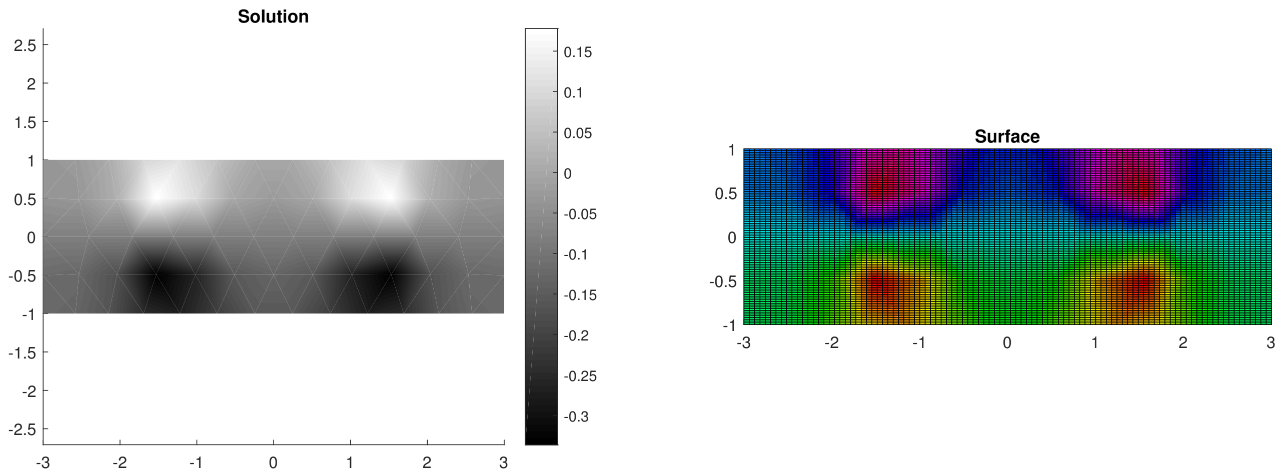
(a) Mesh layout with control volumes (b) Solution ϕ of linear system (c) Stream lines of magnet

Figure 5: Plots of magnetostatic problem with 1 DistMesh resolution

As expected, Figure 5 shows us that the solution and magnetic field lines are not as smooth and reflective of the physics of the problem as finer mesh resolutions. This is explained due to the mesh layout, which has less control volumes and thus, cannot discretize the problem as much as the previous mesh.

More magnets

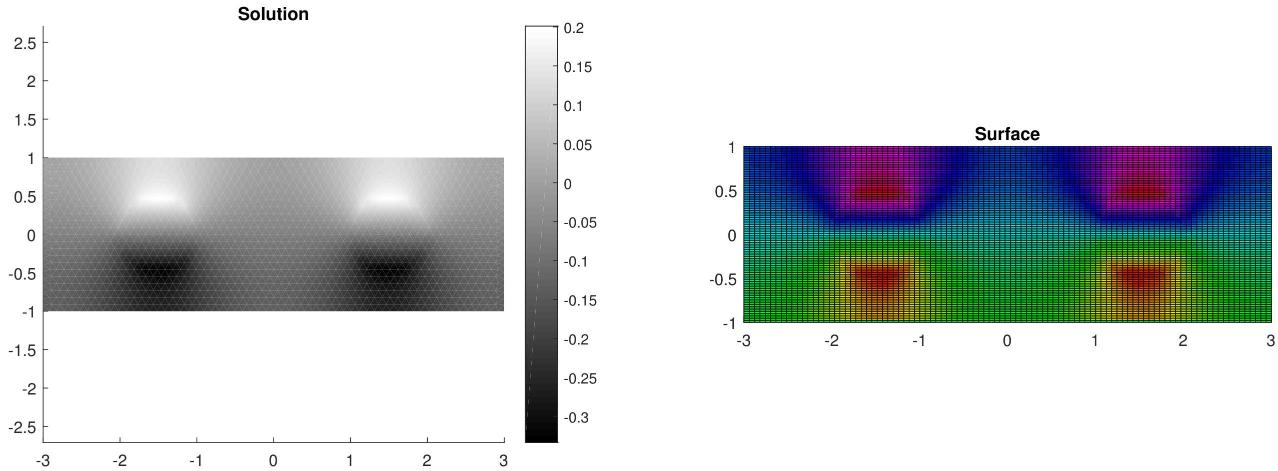
Another experiment that can be devised for the magnetostatic problem is changing the size and amount of magnets on the domain. Let's start by inserting two magnets that do not overlap, still circular, but with a radius of 0.5. For this setup we can see the following results for the original mesh and a finer one.



(a) Solution ϕ of linear system

(b) Surface of ϕ

Figure 6: Solution and surface for two magnets of radius 0.5 on 0.5 resolution mesh



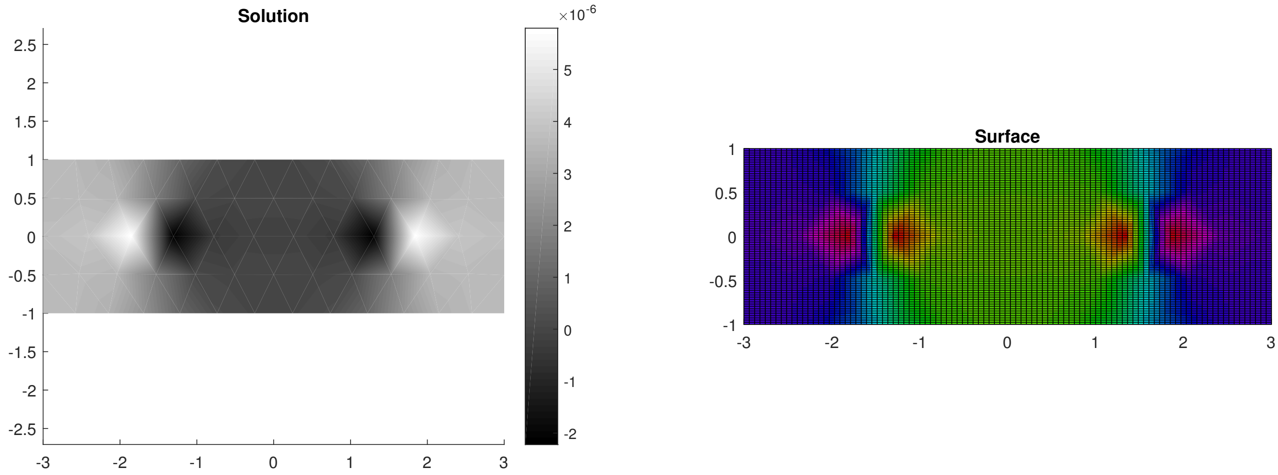
(a) Solution ϕ of linear system

(b) Surface of ϕ

Figure 7: Solution and surface for two magnets of radius 0.5 on 0.1 resolution mesh

For this setup we see the same things that were happening for the original problem. It is possible to confirm that a finer mesh allows for a better discretization of the problem, thus the solution seems much rounder and smoother.

Now by making the radius of the magnets smaller, 0.2, we observe the following behavior for the original mesh.

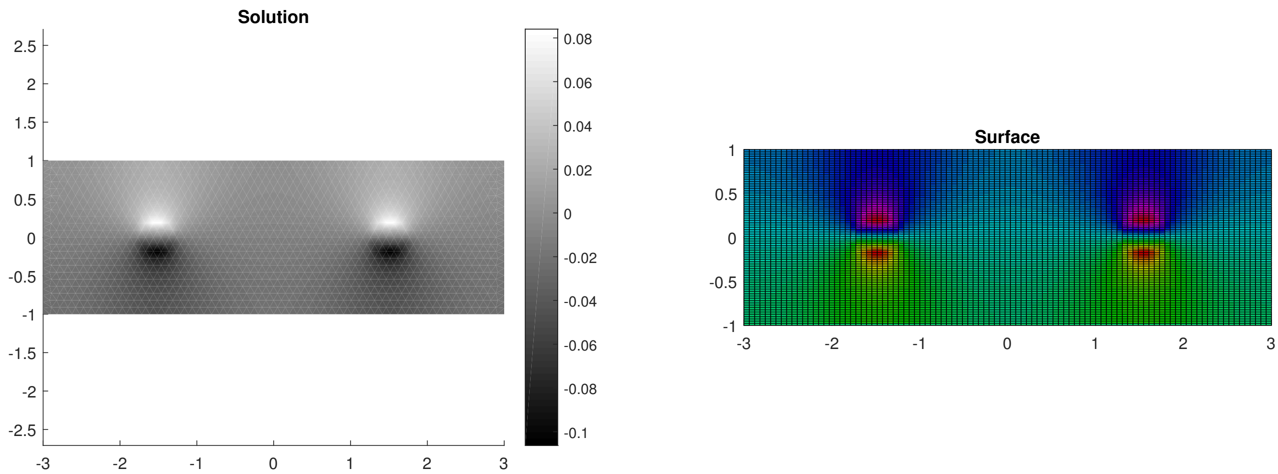


(a) Solution ϕ of linear system

(b) Surface of ϕ

Figure 8: Solution and surface for two magnets of radius 0.2 on 0.5 resolution mesh

It seems that the magnets flip for this mesh resolution and radius size. We'll see if it keeps happening for a finer mesh resolution.



(a) Solution ϕ of linear system

(b) Surface of ϕ

Figure 9: Solution and surface for two magnets of radius 0.2 on 0.1 resolution mesh

It seems that the flipping does not happen for finer resolutions. This can be attributed to a numerical problem that occurs on the previous mesh. When the radius of the magnet gets smaller, its north and south poles are pushed together and the reaction to that is the flipping of the disk. This happens on larger resolutions because the discretization is not as precise as it is for finer meshes. More conclusions could be drawn by inserting more magnets on the domain and varying radius sizes. This implies that the robustness of the FVM has its computational limitations.