

Scientific Computation of Two-Phase Ferrofluid Flows

Gareth Johnson

Faculty Adviser: Prof. Ricardo Nochetto

University of Maryland
AMSC 663: Advanced Scientific Computing I
Supported by Johns Hopkins University Applied Physics Lab

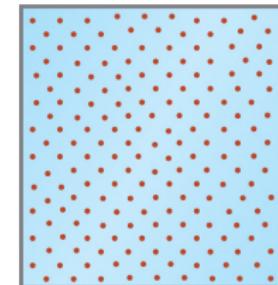
December 4, 2018

What is a Ferrofluid?

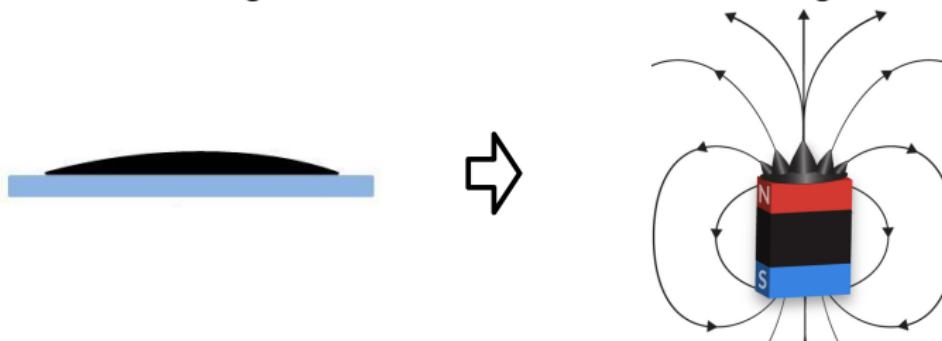
Awesome video of ferrofluids.

What is a Ferrofluid? (Backup)

A ferrofluid is a colloid of nanoscale ferromagnetic particles suspended in a carrier fluid such as oil, water, or an organic solvent.

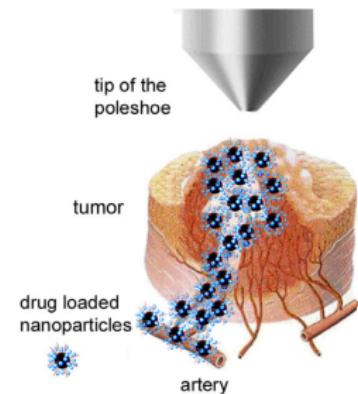


Ferrofluids become magnetized when under the effect of a magnetic field.



Applications

- Initially created to pump rocket fuel once a spacecraft entered a weightless environment.
- Commercial applications:
 - Vibration damping
 - Sensors
 - Acoustics
- Recent research areas:
 - Magnetic drug targeting
 - Adaptive deformable mirrors



PDE Model for Two-Phase Ferrofluid Flow

- Dr. Nochetto and collaborators developed a model for two-phase ferrofluid flows and devised an energy stable numerical scheme [6].
- The model was not derived, but instead was assembled.
- Important results from [6]:
 - Proved an energy law for the PDE model.
 - Proved the numerical scheme was energy stable and the existence of a local solution.
 - For an even simpler model, they proved stability, convergence, and the existence of solutions.

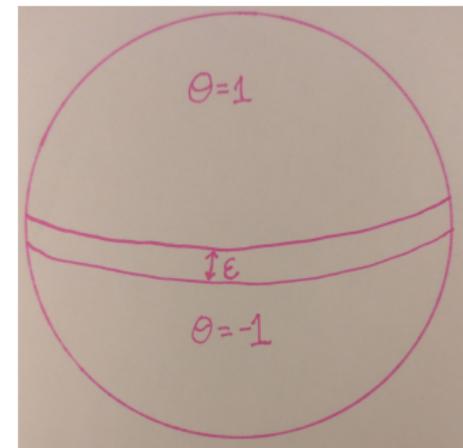
Modeling a Two-Phase Fluid

- In order to track both fluids, a diffuse interface is used.
- The phase variable θ is introduced, which takes values in $[-1, 1]$.
- The evolution of θ is given by a modified Cahn–Hilliard equation:

$$\begin{cases} \theta_t + \operatorname{div}(\mathbf{u}\theta) + \gamma\Delta\psi = 0 & \text{in } \Omega \\ \psi - \epsilon\Delta\theta + \frac{1}{\epsilon}f(\theta) = 0 & \text{in } \Omega \\ \partial_\eta\theta = \partial_\eta\psi = 0 & \text{on } \Gamma, \end{cases}$$

where

- $0 < \epsilon \ll 1$ is related to the interface thickness,
- $\gamma > 0$ is the constant mobility,
- ψ is the chemical potential,
- $f(\theta)$ is the truncated double well potential.



Modeling of the Magnetic Field

- Instead of using the magnetostatics equations, a simplified approach was used.
- Define the magnetic field by

$$\mathbf{h} := \mathbf{h}_a + \mathbf{h}_d,$$

where

- \mathbf{h}_a – smooth harmonic (curl-free and div-free) applied magnetizing field,
- \mathbf{h}_d – demagnetizing field.

- Then the magnetic field is induced via the scalar potential φ by

$$\mathbf{h} = \nabla \varphi,$$

along with,

$$-\Delta \varphi = \operatorname{div}(\mathbf{m} - \mathbf{h}_a) \quad \text{in } \Omega, \quad \partial_\eta \varphi = (\mathbf{h}_a - \mathbf{m}) \cdot \eta \quad \text{on } \Gamma.$$

Modeling of Ferrohydrodynamics

- A simplified version of Shliomis model is used, which couples an advection–reaction equation for the magnetization \mathbf{m} :

$$\mathbf{m}_t + (\mathbf{u} \cdot \nabla) \mathbf{m} = -\frac{1}{\mathcal{T}}(\mathbf{m} - \varkappa_\theta \mathbf{h}),$$

with the Navier–Stokes equations of incompressible fluids for the velocity–pressure pair (\mathbf{u}, p) :

$$\begin{aligned}\mathbf{u}_t + (\mathbf{u} \cdot \nabla) \mathbf{u} - \operatorname{div}(\nu_\theta \mathbf{T}(\mathbf{u})) + \nabla p &= \mu_0(\mathbf{m} \cdot \nabla) \mathbf{h} + \frac{\lambda}{\epsilon} \theta \nabla \psi, \\ \operatorname{div} \mathbf{u} &= 0,\end{aligned}$$

where

- \mathcal{T} is the relaxation time of the ferrofluid,
- \varkappa_θ is the magnetic susceptibility of the phase variable,
- ν_θ is the viscosity of the phase variable,
- μ_0 is the constitutive parameter related to the Kelvin force,
- $\frac{\lambda}{\epsilon} \theta \nabla \psi$ is the capillary force.

- This is supplemented with a no-slip condition on the boundary:

$$\mathbf{u} = 0 \quad \text{on } \Gamma.$$

- The model reads: Consider a bounded convex polygon/polyhedron domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) with boundary Γ . The evolution of the system is given by the following set of equations in strong form in Ω

$$\theta_t + \operatorname{div}(\mathbf{u}\theta) + \gamma\Delta\psi = 0, \quad (1a)$$

$$\psi - \epsilon\Delta\theta + \frac{1}{\epsilon}f(\theta) = 0, \quad (1b)$$

$$\mathbf{m}_t + (\mathbf{u} \cdot \nabla)\mathbf{m} = -\frac{1}{\mathcal{T}}(\mathbf{m} - \varkappa_\theta \mathbf{h}), \quad (1c)$$

$$-\Delta\varphi = \operatorname{div}(\mathbf{m} - \mathbf{h}_a), \quad (1d)$$

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} - \operatorname{div}(\nu_\theta \mathbf{T}(\mathbf{u})) + \nabla p = \mu_0(\mathbf{m} \cdot \nabla)\mathbf{h} + \frac{\lambda}{\epsilon}\theta\nabla\psi, \quad (1e)$$

$$\operatorname{div}\mathbf{u} = 0, \quad (1f)$$

for every $t \in [0, t_F]$, where $\mathbf{T}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ denotes the symmetric gradient and $\mathbf{h} = \nabla\varphi$. The system (1) is supplemented with the boundary conditions

$$\partial_\eta\theta = \partial_\eta\psi = 0, \quad \mathbf{u} = 0, \quad \text{and} \quad \partial_\eta\varphi = (\mathbf{h}_a - \mathbf{m}) \cdot \eta \quad \text{on } \Gamma. \quad (2)$$

Define the backward difference operator $\delta f^k = f^k - f^{k-1}$.

For given smooth initial data $\{\Theta^0, \mathbf{M}^0, \mathbf{U}^0\}$ and timestep τ , compute $\{\Theta^k, \Psi^k, \mathbf{M}^k, \Phi^k, \mathbf{U}^k, P^k\} \in \mathbb{G}_h \times \mathbb{Y}_h \times \mathbb{M}_h \times \mathbb{X}_h \times \mathbb{U}_h \times \mathbb{P}_h$ for every $k \in \{1, \dots, K\}$ that solves

$$\left(\frac{\delta \Theta^k}{\tau}, \Lambda \right) - (\mathbf{U}^k \Theta^{k-1}, \nabla \Lambda) - \gamma(\nabla \Psi^k, \nabla \Lambda) = 0, \quad (3a)$$

$$(\Psi^k, \Upsilon) + \epsilon(\nabla \Theta^k, \nabla \Upsilon) + \frac{1}{\epsilon}(f(\Theta^{k-1}), \Upsilon) + \frac{1}{\eta}(\delta \Theta^k, \Upsilon) = 0, \quad (3b)$$

$$\left(\frac{\delta \mathbf{M}^k}{\tau}, \mathbf{Z} \right) - \mathcal{B}_h^m(\mathbf{U}^k, \mathbf{Z}, \mathbf{M}^k) + \frac{1}{\mathcal{T}}(\mathbf{M}^k, \mathbf{Z}) = \frac{1}{\mathcal{T}}(\varkappa_\theta \mathbf{H}^k, \mathbf{Z}), \quad (3c)$$

$$(\nabla \Phi^k, \nabla X) = (\mathbf{h}_a^k - \mathbf{M}^k, \nabla X), \quad (3d)$$

$$\left(\frac{\delta \mathbf{U}^k}{\tau}, \mathbf{V} \right) + \mathcal{B}_h(\mathbf{U}^{k-1}, \mathbf{U}^k, \mathbf{V}) + (\nu_\theta \mathbf{T}(\mathbf{U}^k), \mathbf{T}(\mathbf{V})) - (P^k, \operatorname{div} \mathbf{V}) = \mu_0 \mathcal{B}_h^m(\mathbf{V}, \mathbf{H}^k, \mathbf{M}^k) \quad (3e)$$

$$+ \frac{\lambda}{\epsilon}(\Theta^{k-1} \nabla \Psi^k, \mathbf{V}),$$

$$(Q, \operatorname{div} \mathbf{U}^k) = 0. \quad (3f)$$

Numerical Implementation Details

Discretization of the Numerical Scheme:

- Time Discretization: Backward Euler is used.
- Space Discretization: A mix of Continuous and Discontinuous Galerkin is used, approximating the spaces with polynomials of degree 2 in each variable (i.e. Q_2 elements).
 - Continuous: Cahn–Hilliard, Magnetic potential, and Navier Stokes equations.
 - Discontinuous: Magnetization equations.

Fixed Point Solver:

- A Picard–like iteration is used.
- Utilizes the "lagging" of the velocity \mathbf{U} to solve each subsystem.
- Iterates until a fixed point for \mathbf{U}^k is reached.
- Given \mathbf{U}^{k-1}
 - 1) Compute Θ^k and Ψ^k substituting \mathbf{U}^{k-1} for \mathbf{U}^k .
 - 2) Next compute \mathbf{M}^k and Φ^k using (Θ^k, Ψ^k) from the previous iteration and substituting \mathbf{U}^{k-1} for \mathbf{U}^k .
 - 3) Finally, compute \mathbf{U}^k and P^k using $(\Theta^k, \Psi^k, \mathbf{M}^k, \Phi^k)$ from the previous two iterations.
 - 4) Repeat steps 1-3 using \mathbf{U}^k from the previous iteration as input until \mathbf{U}^k does not change between iterations.

Subsystem Solvers

Cahn–Hilliard system (3a)–(3b):

- Linearized using convex–concave splitting:

$$f(\Theta^k) \rightarrow f(\Theta^{k-1}) + \eta \delta \Theta^k, \quad \text{where } \eta \leq (\max_{\theta} f'(\theta))^{-1}.$$

- The resulting system is linear but non–symmetric.
- Solved using GMRES preconditioned with algebraic multigrid.

Magnetization system (3c)–(3d):

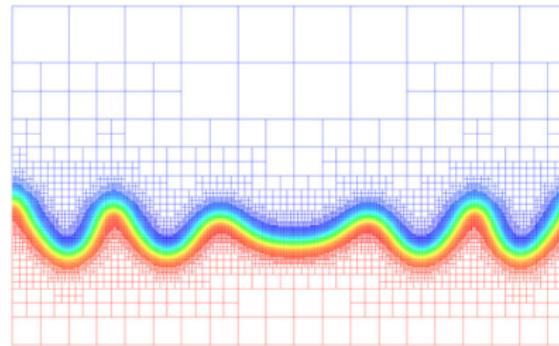
- Equation (3c) is mass dominated and non–symmetric.
- Solved using BiCGstab.
- Equation (3d) is a Laplacian, which is symmetric.
- Solved using CG preconditioned with algebraic multigrid.

Navier–Stokes system (3e)–(3f):

- It is a non–symmetric saddle point problem.
- Solved using GMRES with a block preconditioner.
- Will explore preconditioners presented in Elman's book [3].

Adaptive Mesh Refinement/Coarsening

- In order to resolve the interface, we need $0 < \epsilon \ll 1$.
- This requires the mesh to be highly dense near the interface.
- If a uniform meshsize h is used, this would lead to very large linear systems.
- To overcome this, we will use adaptive mesh refinement/coarsening.



- The adaptive mesh refinement/coarsening will use the simplest element indicator η_T :

$$\eta_T^2 = h_T \int_{\partial T} \left| \left[\frac{\partial \Theta}{\partial \eta} \right] \right|^2 dS \quad \forall T \in \mathcal{T}_h.$$

Generation of the Magnetic Field

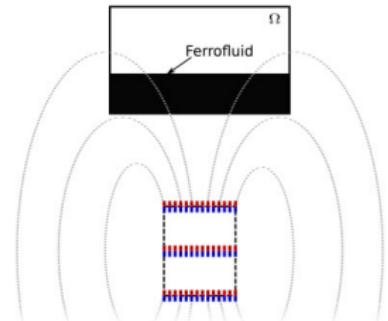
Define the magnetic dipole ϕ_s by

$$\phi_s(\mathbf{x}) = \frac{\mathbf{d} \cdot (\mathbf{x}_s - \mathbf{x})}{|\mathbf{x}_s - \mathbf{x}|},$$

where $|\mathbf{d}| = 1$ and $\mathbf{d}, \mathbf{x}_s, \mathbf{x} \in \mathbb{R}^2$. Then the applied magnetizing field \mathbf{h}_a is given by

$$\mathbf{h}_a = \sum_s \alpha_s(t) \nabla \phi_s,$$

where $\alpha_s(t)$ is the intensity of each dipole.



Project Goals

- 1) Finite Element Code: Develop a finite element code to solve two-phase ferrofluid flows using the numerical scheme (3). This code will be written in a “dimensional-less” way, explained later, so that the code can be easily transitioned from 2d to 3d.
- 2) Solvers: In order to solve (3), three different solvers are required.
- 3) Scientific Questions: Investigate the structure of the velocity field of the fluid flow and the magnetic field around the spike deformation of the ferrofluid for both the Rosenswieg instability and the ferrofluid hedgehog configuration in 2d.

If time permits, the following extensions of the project will be explored:

- 4) Parallelization: Implement parallel adaptive mesh refinement/coarsening.
- 5) Ferrofluid Droplets: Investigate if the model can accurately capture various effects of ferrofluid droplets, such as the coalescence of droplets [4] and the equilibrium shape of droplets under a uniform magnetic field [7].

Finite Element Code:

- 1) Write codes to handle the generation of the finite element spaces given in [6].
- 2) Write codes to handle the generation and adaptive refinement/coarsening of the mesh.
- 3) Write a code to handle the generation of the matrices, which will combine information from the mesh and the finite elements.
- 4) Write codes that solve each of the three subsystems, namely the Cahn–Hilliard system (3a)–(3b), the magnetization system (3c)–(3d), and the Navier–Stokes system (3e)–(3f).
- 5) Write a code to solve the full system (3) at each time step using a Picard–like iteration.
- 6) Write a code to handle the generation of the applied harmonic field, given the locations of each magnetic dipole.
- 7) Include functionality for the numerical simulation to be restarted from the last completed iteration.

Numerical Investigation:

- Generate and analyze contour and vector plots of the velocity and magnetic field for the three experiments performed in [6].

Target Platform

Hardware:

- A Linux desktop system.
- Using desktop owned by Dr. Nocetto with a Intel Xeon CPU with 24 cores and 68 GB of ram

Software:

- Developed in C++.
- The code will utilize the deal.II library [1, 2]. The library provides functionality to
 - create meshes,
 - generate finite elements,
 - aid in adaptive mesh refinement/coarsening,
 - solve linear algebra systems with preconditioners.

Distribution:

- Source code and user guide will be hosted on Github.

Validation Methods

Generated Solutions:

- Each of the three subsystems will be verified using a generated solution.
- Simple Example: For $u = \sin(x)$ to be a solution to

$$u' + u = 0,$$

the forcing $f(x) = \sin(x) + \cos(x)$ will be added to the RHS.

Mesh Verification:

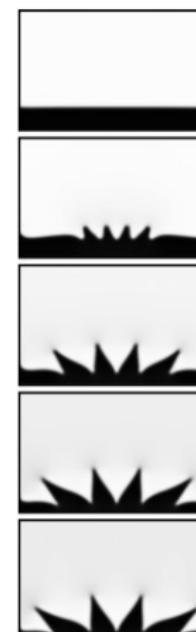
- Visually will verify input mesh.
- Adaptive mesh refinement/coarsening verified using a simple mesh with assigned error values.

Verification Methods (continued)

Comparison with prior works:



Uniform Magnetic Field



Non-uniform magnetic field
 $\mathbf{h} := \mathbf{h}_a + \mathbf{h}_d$



Non-uniform magnetic field
 $\mathbf{h} := \mathbf{h}_a$

Solving PDEs using deal.II

Each program in deal.II can be broken down into 5 main steps:

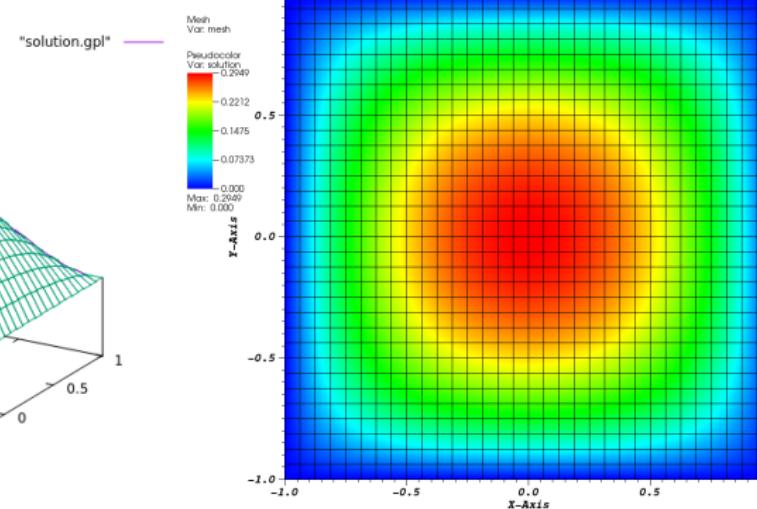
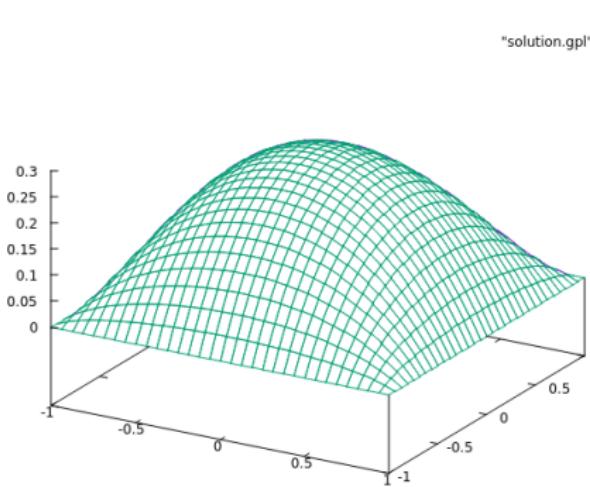
- 1) Generate the mesh.
- 2) Distribute degrees of freedom and set up the matrices for the associated system.
- 3) Compute the left and right hand sides of the system.
- 4) Solve the system numerically.
- 5) Output the solution in a specific format for visualization or post-processing.

Example 1

The first PDE solved was Poisson's equation:

$$\begin{cases} -\Delta u = f(x) & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $f(x) = 1$ and $\Omega = [0, 1]^2$.



Example 2

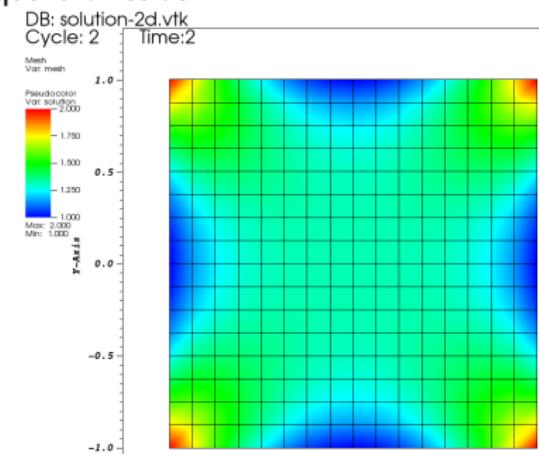
The second PDE solved was Poisson's equation:

$$\begin{cases} -\Delta u = f(x) & \text{in } \Omega \\ u = g(x) & \text{on } \partial\Omega, \end{cases}$$

where

$$f(x) = \begin{cases} 4(x^4 + y^4) & \text{if } \Omega \subset \mathbb{R}^2 \\ 4(x^4 + y^4 + z^4) & \text{if } \Omega \subset \mathbb{R}^3, \end{cases} \quad g(x) = \begin{cases} x^2 + y^2 & \text{if } \Omega \subset \mathbb{R}^2 \\ x^2 + y^2 + z^2 & \text{if } \Omega \subset \mathbb{R}^3, \end{cases}$$

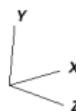
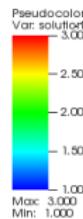
and Ω is the unit square or cube.



3D Plot

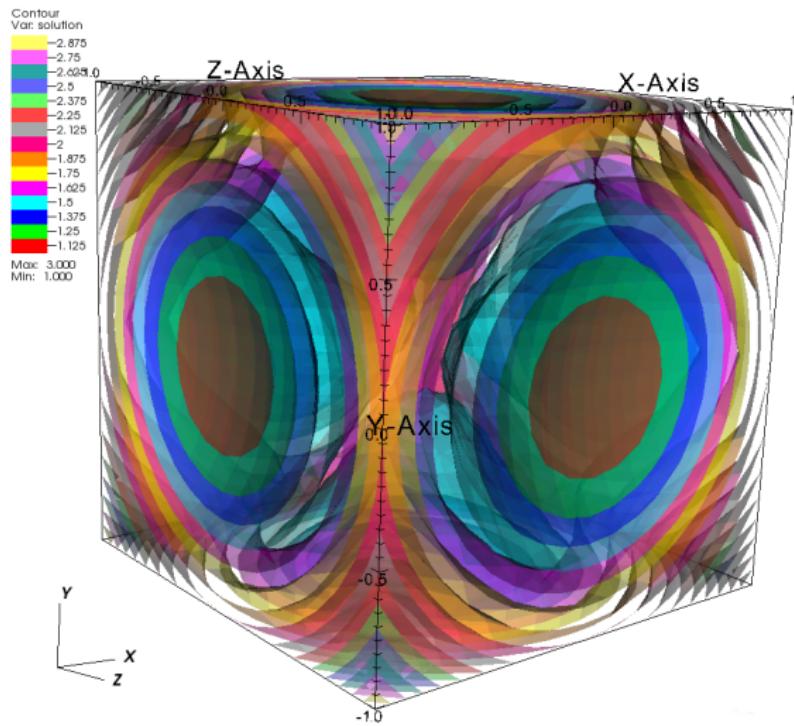
DB: solution-3d.vtk
Cycle: 3 Time: 3

Mesh
Var: mesh



3D Contour Plot

DB: solution-3d.vtk
Cycle: 3 Time: 3



Example 3

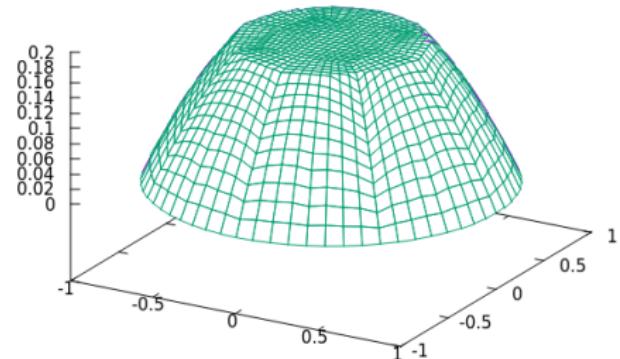
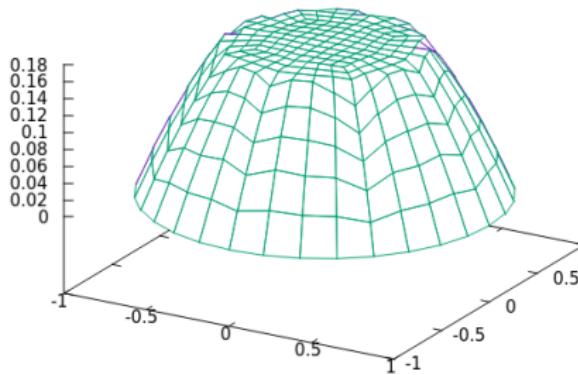
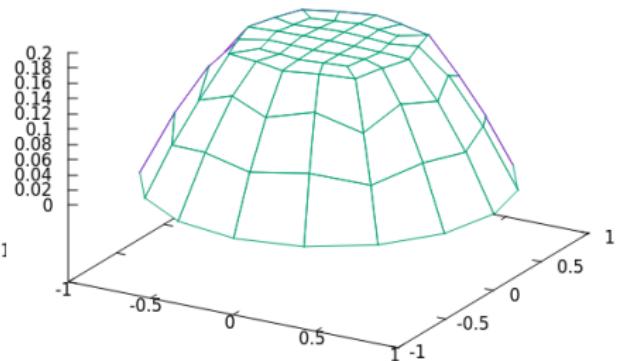
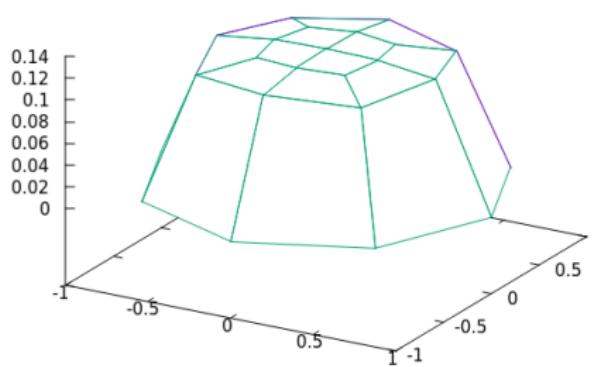
The third PDE solved was Poisson's equation:

$$\begin{cases} -\nabla \cdot (a(x)\nabla u(x)) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = [0, 1]^2$ and

$$a(x) = \begin{cases} 20 & \text{if } |x| < 0.5 \\ 1 & \text{otherwise} \end{cases}.$$

The problem was solved multiple times on increasing global refinements of the mesh.



Example 4

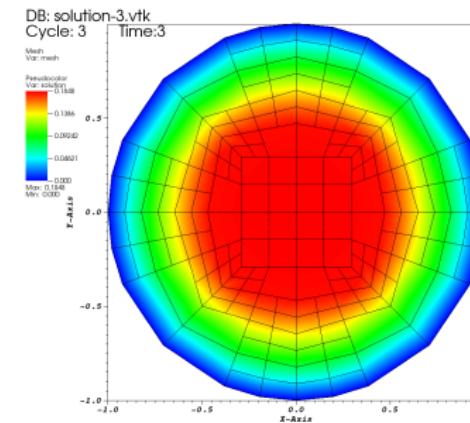
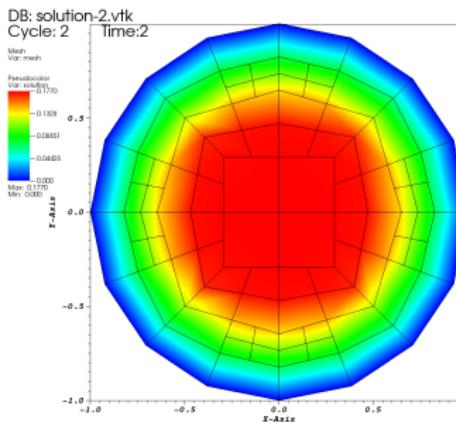
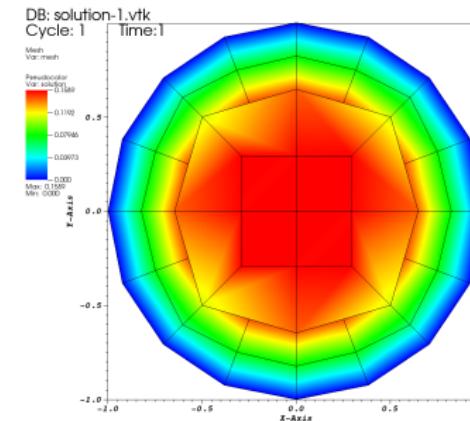
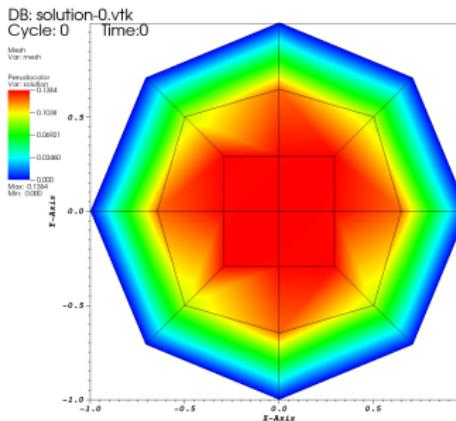
The fourth example was the previous problem:

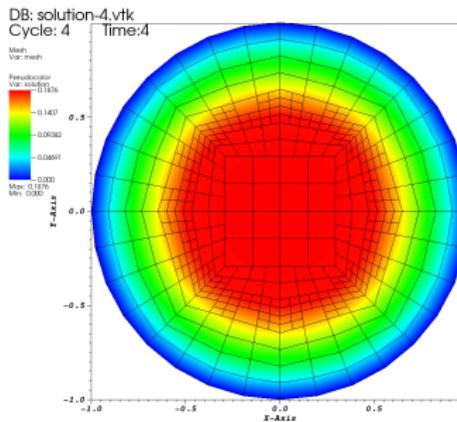
$$\begin{cases} -\nabla \cdot (a(x)\nabla u(x)) = 1 & \text{in } \Omega \\ u = 0 & \text{on } \partial\Omega, \end{cases}$$

where $\Omega = [0, 1]^2$ and

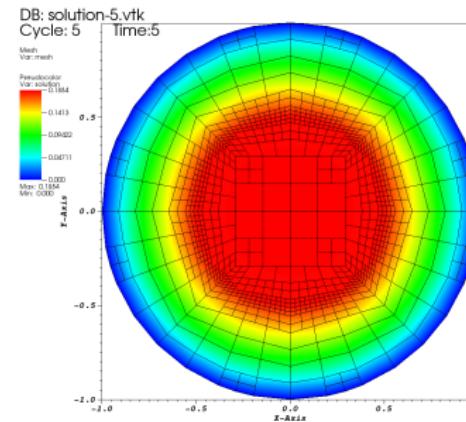
$$a(x) = \begin{cases} 20 & \text{if } |x| < 0.5 \\ 1 & \text{otherwise} \end{cases},$$

except that we used adaptive mesh refinement/coarsening. At each cycle, 30% of the elements with the highest error were refined and 3% of the elements with the lowest error were coarsened.

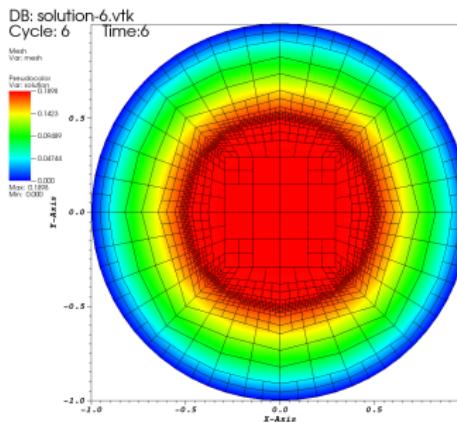




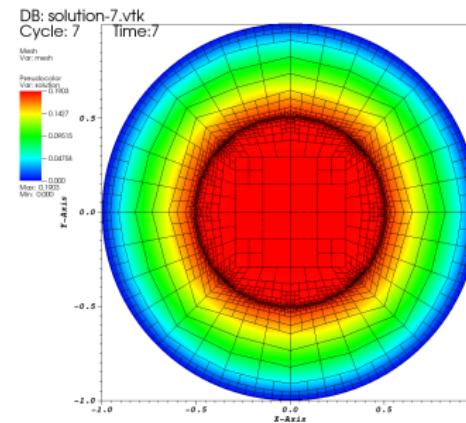
User: werdho
Tue Dec 4 12:17:21 2018



User: werdho
Tue Dec 4 12:17:22 2018



User: werdho



User: werdho



Example 5

The fifth example was the Helmholtz equation:

$$\begin{cases} \Delta u + u = f & \text{in } \Omega \\ u = g_1 & \text{on } \Gamma_1 = \Gamma \cap \{\{x = 1\} \cup \{y = 1\}\}, \\ \mathbf{n} \cdot \nabla u = g_2 & \text{on } \Gamma_2 = \Gamma \setminus \Gamma_1 \end{cases}$$

where $\Omega = [-1, 1]^2$. The solution was chosen to be

$$u(x) = \sum_{i=1}^3 \exp\left(-\frac{|x - x_i|^2}{\sigma^2}\right),$$

with centers $x_1 = (-\frac{1}{2}, \frac{1}{2})$, $x_2 = (-\frac{1}{2}, -\frac{1}{2})$, $(\frac{1}{2}, -\frac{1}{2})$ and $\sigma = \frac{1}{8}$. This was solved using both global and adaptive refinement.

We then computed the error in various norms for each refinement technique:

cycle	# cells	# dofs	L^2 -error	H^1 -error	L^∞ -error
0	64	81	2.380e-01	1.482e+00	3.587e-01
1	256	289	4.283e-02	1.285e+00	1.463e-01
2	1024	1089	1.353e-02	7.557e-01	7.890e-02
3	4096	4225	3.424e-03	3.822e-01	2.343e-02
4	16384	16641	8.588e-04	1.917e-01	6.116e-03

Figure: Error using global refinement.

cycle	# cells	# dofs	L^2 -error	H^1 -error	L^∞ -error
0	64	81	2.380e-01	1.482e+00	3.587e-01
1	124	158	4.333e-02	1.286e+00	1.512e-01
2	280	341	2.445e-02	7.945e-01	8.081e-02
3	571	682	4.950e-02	5.251e-01	8.184e-02
4	1087	1251	6.375e-03	3.118e-01	1.656e-02
5	2122	2384	9.289e-03	2.214e-01	1.632e-02
6	4051	4374	8.039e-03	1.554e-01	1.243e-02
7	7699	8212	8.747e-03	1.142e-01	1.322e-02
8	14746	15461	4.118e-03	8.824e-02	6.901e-03

Figure: Error using adaptive refinement.

Finally, using global refinement we estimated the convergence rate:

n cells		H^1 -error		L^2 -error	
0	64	1.482e+00	-	2.380e-01	-
1	256	1.285e+00	0.21	4.283e-02	2.47
2	1024	7.557e-01	0.77	1.353e-02	1.66
3	4096	3.822e-01	0.98	3.424e-03	1.98
4	16384	1.917e-01	1.00	8.588e-04	2.00

Figure: Estimated convergence using global refinement.

Example 6

The sixth example was the Stokes equation:

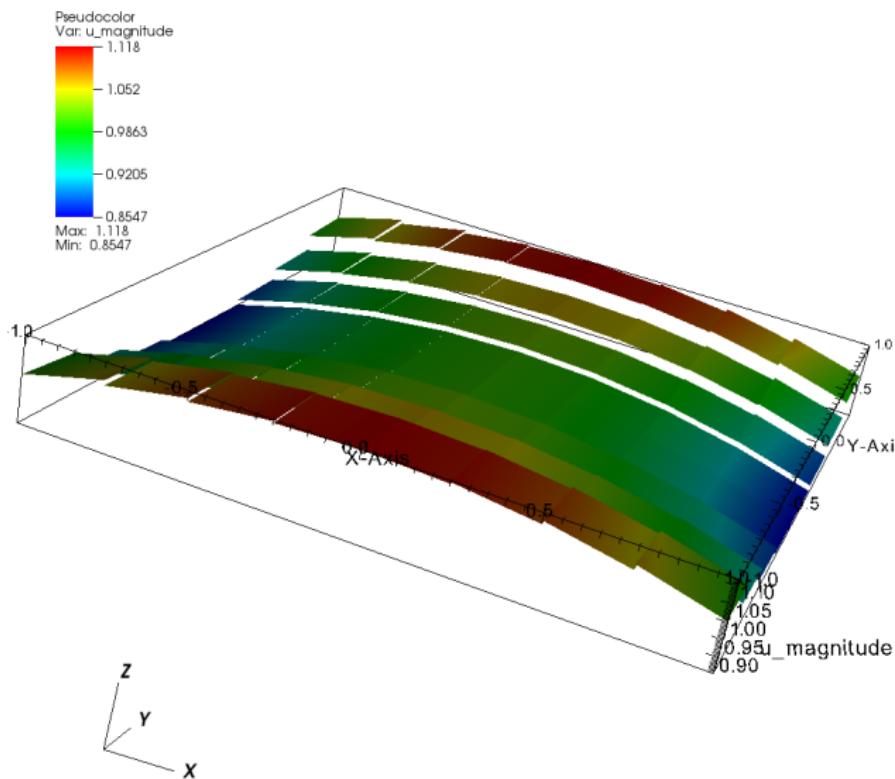
$$\begin{cases} K^{-1}\mathbf{u} + \nabla p = 0 & \text{in } \Omega \\ -\operatorname{div} \mathbf{u} = -f & \text{in } \Omega \\ p = g & \text{on } \partial\Omega, \end{cases}$$

where K was chosen to be the identity. Again, we chose the exact solution to be

$$p = -\left(\frac{\alpha}{2}xy^2 + \beta x - \frac{\alpha}{6}x^3\right), \quad \mathbf{u} = \begin{pmatrix} \frac{\alpha}{2}y^2 + \beta - \frac{\alpha}{6}x^2 \\ \alpha xy \end{pmatrix}.$$

This system was solved using a Schur complement solver.

DB: solution.vtu



Example 6

The final example was the heat equation:

$$\begin{cases} \partial_t u(x, t) - \Delta u(x, t) = f(x, t) & \text{in } \Omega \times (0, 1] \\ u(x, 0) = u_0(x) & \text{on } \Omega \times \{0\} \\ u(x, t) = g(x, t) & \text{on } \partial\Omega \times (0, 1]. \end{cases}$$

The solution was chosen to be

$$u(x, y, t) = \sin(3\pi x)e^{-y-2t}.$$

This problem was solved using a mesh that is adaptively refined every 5 iterations.

Finite Element Code:

- 1) Write codes to handle the generation of the finite element spaces given in [6].
- 2) Write codes to handle the generation and adaptive refinement/coarsening of the mesh.
- 3) Write a code to handle the generation of the matrices, which will combine information from the mesh and the finite elements.
- 4) Write codes that solve each of the three subsystems, namely the Cahn–Hilliard system (3a)–(3b), the magnetization system (3c)–(3d), and the Navier–Stokes system (3e)–(3f).
- 5) Write a code to solve the full system (3) at each time step using a Picard–like iteration.
- 6) Write a code to handle the generation of the applied harmonic field, given the locations of each magnetic dipole.
- 7) Include functionality for the numerical simulation to be restarted from the last completed iteration.

Numerical Investigation:

- Generate and analyze contour and vector plots of the velocity and magnetic field for the three experiments performed in [6].

Below is an updated timeline for the project:

- Mid January: Begun development on the solver for the Cahn–Hilliard system.
- Mid February: Finished and tested the solver for the Cahn–Hilliard system. Begun development on the solver for the Navier–Stokes system.
- Mid March: Finished and tested the solver for the Navier–Stokes system. Begun development on the solver for the magnetization system.
- Mid April: Finished and tested the solver for the magnetization system. Begun development on the Picard iteration solver.
- Semester End: Delivered all required items and the final report to the teaching staff.

Figure References

- Slide 1: https://youtu.be/wHZDgSFzQ_s?t=12
- Slide 2:
 - https://www.researchgate.net/profile/Vikram_Raghavan2/post/What_is_the_effect_of_magnetic_field_on_alignment_of_ferrofluid_droplet/attachment/59d622166cda7b8083a1b9a2/AS%3A273810673078272%401442292959057/download/Effect+of+Magnetic+field.jpg
 - https://ksr-ugc.imgix.net/assets/003/310/641/f0ef73d1fd99f6aa5d96872168478df4_original.png?v=1424378871&w=680&fit=max&auto=format&lossless=true&s=c183d857603c12de82a71f3139283d9e
 - https://opentextbc.ca/chemistry/wp-content/uploads/sites/150/2016/05/CNX_Che_11_05_Colloid.jpg
- Slide 3: [5]
- Slide 12: [6]
- Slide 13: [6]
- Slide 18: [6]

References |

-  W. BANGERTH, C. BURSTEDDE, T. HEISTER, AND M. KRONBICHLER, *Algorithms and data structures for massively parallel generic adaptive finite element codes*, ACM Trans. Math. Softw., 38 (2012), pp. 14:1–14:28.
-  W. BANGERTH, T. HEISTER, AND G. KANSCHAT, *Deal.II differential equations analysis library, technical reference*.
-  H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, 2005.
-  A. GHAFFARI AND S. H. HASHEMABADI, *Parameter study and cfd analysis of head on collision and dynamic behavior of two colliding ferrofluid droplets*, Smart Materials and Structures, 26 (2017).
-  C. JANKO, S. DÜRR, L. E. MUÑOZ, S. LYER, R. CHAURIO, R. TIETZE, S. V. LÖHNEYSEN, C. SCHORN, M. HERRMANN, AND C. ALEXIOU, *Magnetic drug targeting reduces the chemotherapeutic burden on circulating leukocytes*, International Journal of Molecular Sciences, 14 (2013), pp. 7341–7355.
-  R. H. NOCHETTO, A. J. SALDAGO, AND I. TOMAS, *A diffuse interface model for two-phase ferrofluid flows*, Computer Methods in Applied Mechanics and Engineering, 309 (2016), pp. 497–531.
-  P. ROWGHANIAN, C. D. MEINHART, AND O. CAMPÀS, *Dynamics of ferrofluid drop deformations under spatially uniform magnetic fields*, Journal of Fluid Mechanics, 802 (2016), pp. 245–262.