Name: Hongda Li
Class: AMATH 581

Repos

(Click this to access the public repo)
My Repo
Files:

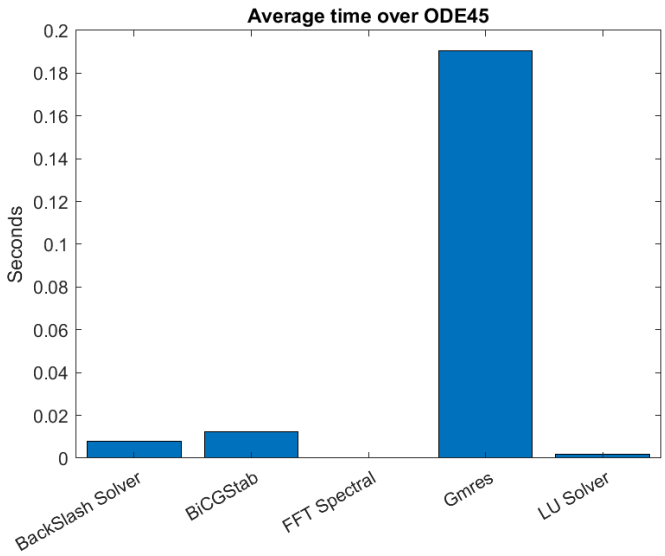
- Psiguess.m:
  This is a handler class that is designed to hold the last value of ω for the iterative solver.
- Parameters.m
  A class that holds all the relevant parameters for the solver to perform.
- VectorizedInitialDistribution.m
  A function that will package the initial conditions given a function, in the form of $f(x, y)$ for the 2d plane.
- SolveForPsi.m
  The function will solve for Psi.
- solution.m
  Gradescope Submission file.
- Rhs.m
  The right hand side of the Vorticity function.
- GetFramer.m
  Get an objects for storing the frames from the animations.
- FiniteDiffMatrix.m
  A function that will returns sparse diagonal matrix with periodic conditions.
- EllipticTransform.m
  Transform a Mesgrid Domain by rotating it and stretching it.
- ScratchPaper.m
  Scratch paper.
- CompiledToHW4.m
  Just another Scratch Paper.
- BecnhMark.m
  Run all 5 ways of solving the system for the ODE45 and then plot the average time it takes for it to solve them.
- AsCoolAsItGets.m Run the simulations for different initial conditions and save them as animations for this HW. **This is the main File**
- Test.1/2/3/4.avi
- benchmark.png
- ⟨This file ⟩

Methodology: Choice of Solver

We are using the FFT for simulations because it's experimentally the fastest, and it's also the fastest in theory.

The iterative solver is not fast even if a close guess is provided, this might due to the large size of the sparse matrix. Biconjugate Gradient Descend is way faster than Minimal Residual Method.

Here is a plot:



Note, the time it takes for FFT is so small that it's just not showing up on the graph.

Choice of Color

The color for the first 3 simulations is chosen to be "pink" because it feels like Mocha, and I like Mocah.

The color of the last simulations is "jet" because I want all the Guassian Bumps with different magnitude to be clearly distinguisable.

The color can be tweaked in the "AsCoolAsItGets.m" file, under the "SaveToAnimations" function.

Simulations Produced

- File: Test1.avi
  2 Oppositely charged Guassian Vorties next to each other without colliding.
- File: Test2.avi
  Same charged Guassian vorties next to each other, collided after a while.
- File: Test3.avi
  Pairs of Oppositely charged vortices which collided during simulations.
- File: Test4.avi
  A random assortment of Vortices, like really random, I wrote codes to produce true randomness, but it gets a bit finnicky with the boundary, it's not really looping back, but I think it's good enough, and it's looking like the surface of a bubbles.