AMATH 481 & 581 Fall 2018

Homework 4 - Computing Vorticity-Streamfunction Equations

Submission open until 11:59:59pm Monday November 26, 2018

The time evolution of the vorticity $\omega(x, y, t)$ and streamfunction $\psi(x, y, t)$ are given by the governing equations:

$$\omega_t + [\psi, \omega] = \nu \nabla^2 \omega \tag{1}$$

where $[\psi, \omega] = \psi_x \omega_y - \psi_y \omega_x$ and $\nabla^2 = \partial_x^2 + \partial_y^2$. The streamfunction satisfies

$$\nabla^2 \psi = \omega. \tag{2}$$

In most applications, the diffusion is a small parameter. This fact helps the numerical stability considerably. Here, take $\nu = 0.001$.

Boundary Conditions: Assume periodic boundary conditions for both the vorticity and the streamfunction.

Initial Conditions: Assume an elliptic Gaussian bump, stretched in the y-direction, as an initial condition for the vorticity $\omega(x, y, 0) = \exp\left(-x^2 - (y^2/20)\right)$.

Integrate the equations numerically by discretizing both Eq. (1) and (2) and using ODE45 to step forward in time. To solve for the streamline equation, Eq. (2), use the following two methods:

(a) Discretize using central-difference and transform Eq. (2) into a linear problem $A\vec{x} = \vec{b}$ (NOTE: Take $A(1, 1) = 2$ instead of $A(1, 1) = -4$).
Solve the linear system with: **(i)** $A\backslash b$ and **(ii)** LU decomposition. Which method do you expect to be faster in total running time? To justify your answer keep track of the computational speed of each method (check out MATLAB's `tic,toc,cputime` commands). In addition, experiment with the tolerance to see how quickly these iteration schemes converge.
ANSWERS: With $x, y \in [-10, 10]$, $n = 64$, and $tspan = 0 : 0.5 : 4$, write out the solution of your numerical evolution (the vorticity ω) from ode45 as A1.dat for $A\backslash b$ and A2.dat for LU. A1.dat and A2.dat should be matrices of dimensions 9×4096.

BONUS: Solve the linear system with iterative methods: `bicgstab` and `gmres`. For the first few times solving the streamfunction equations, keep track of the residual as a function of the number of iterations needed to converge to the solution. Note that you should adjust the tolerance settings in `bicgstab` and `gmres` to be consistent with your accuracy in the time-stepping. Write out the solution for `bicgstab` and `gmres` as A1b.dat and A2b.dat, respectively and submit the files to **Homework4_Bonus**.

(b) Solve Eq. (2) using FFT (NOTE: set $k_x(1) = k_y(1) = 10^{-6}$). How much time does it take now? Compare with the previous methods.
ANSWERS: With $x, y \in [-10, 10]$, $n = 64$, and $tspan = 0 : 0.5 : 4$, write out the solution of your numerical evolution (the vorticity ω) from ode45 as A3.dat, a matrix of dimensions 9×4096.

(c) Try the initial conditions below with your favorite/**fastest** solver for the streamfunction equations. In these simulations make sure that the Gaussian bumps are well-contained within your spatial domain (you can increase the domain if needed/scale the bumps/play with discretization):

- Two oppositely "charged" Gaussian vortices next to each other, i.e. one with positive amplitude, the other with negative amplitude.
- Two same "charged" Gaussian vortices next to each other.
- Two pairs of oppositely "charged" vortices which can be made to collide with each other.
- A random assortment (in position, strength, charge, ellipticity, etc.) of vortices on the periodic domain. Try 10-15 vortices and watch what happens.

(d) Make a **creative** 2-D movie of the dynamics (combining, mixing initial conditions above to an interesting simulation). **Color and coolness** are key here. (MATLAB commands: `movie, getframe`).

For (c) and (d) turn-in the generating files (and videos) to **"AsCoolAsItGets" canvas assignment**. We would very much like to see everyone's movies.