AOS 180 – Modeling Task 6
QG barotropic vorticity equation
Due date: May 17^{th}11:00AM

Problem description

We are interested in the solution of the QG barotropic vorticity equation system given by

$$\frac{\partial \zeta_g}{\partial t} = -J(\psi, \zeta_g) - \beta \frac{\partial \psi}{\partial x} \tag{1}$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \zeta_g(x, y), \tag{2}$$

where the advection term is written using the 2D Jacobian defined here as

$$J(a, b) = \frac{\partial a}{\partial x}\frac{\partial b}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial b}{\partial x}. \tag{3}$$

We are interested in a solution on the domain $0 \leq x \leq L_x$ and $0 \leq y \leq L_y$, with initial condition prescribe by the vorticity field ζ_g with two Gaussian vortices with the same circulation Γ. The vorticity for each vortex is given by

$$\zeta_g(x, y) = \frac{\Gamma}{\pi a^2} \exp\left(-\frac{(x - x_0)^2 + (y - y_0)^2}{a^2}\right), \tag{4}$$

where a is the radius of the vortex centered at (x_0, y_0). We will place the vortices separated by a distance b. For simplicity, we will look at the evolution of the vorticity field in an enclosed domain, with walls on all four sides (this allows more flexibility because we do not need to require the average vorticity to be zero in the domain). The boundary conditions appropriate for this problem on all four walls are

$$\zeta_g = 0 \tag{5}$$

$$\mathbf{v}_g \cdot \hat{\mathbf{n}} = 0. \tag{6}$$

Here, $\hat{\mathbf{n}}$ is the unit vector normal to the boundary and (6) corresponds to the no-penetration boundary condition. In practice, (6) can be written as $u_g(x = 0, y) = u_g(x = L_x, y) = 0$ and $v_g(x, y = 0) = v_g(x, y = L_y) = 0$, both of which can be satisfied by simply setting $\psi = 0$ on all four walls. Therefore, we have simple Dirichlet boundary conditions $\psi = 0$ and $\zeta_g = 0$ on all walls.

For the problem of two identical vortices of radius a in an inviscid flow (i.e., no viscosity), there is a critical ratio between vortex radius and the distance between the two vortices $(a/b) \approx 0.22$ above which the two vortices quickly merge into one. If the ratio is below critical, the two vortices rotate around each other. For more information about this problem, check out the review paper *Physics of Vortex Merging* by Meunier et al. (2005), which I uploaded to CCLE for your convenience.

Numerical approach

We will use the centered second-order finite difference scheme to discretize both equations with the Arakawa discretization of the Jacobian and the leapfrog scheme for the time advancement of the vorticity equation. We will use SOR to solve the Poisson equation, with the same convergence criteria and tolerance from Modeling Task 5. As a matter of fact, I suggest you actually start by merging your codes from Modeling Tasks 4 and 5, as together they probably contain 70% of the final code you will need here. If you take this approach, set the diffusivity in your code to be $K_x = K_y = 0$.

For the present solution we will use the following configuration: $L_x = L_y = 2000\,\text{km}$, $a = 180\,\text{km}$, $\Gamma/(\pi a^2) = 8 \times 10^{-5}\,\text{s}^{-1}$ and the initial distance between the two vortices $b = 600\,\text{km}$, which I suggest you accomplish by setting the centers of the two vortices to be at $x_0 = L_x/2 \pm b/2$ and $y = L_y/2$ so that they are centered in the domain. I suggest you use 101 grid points in each direction to write and test your code, but then produce a final simulation with a finer resolution so that you can resolve more details in the solution. Our vorticity is quite small, so we will need to integrate for a very long time. Evolve your simulation in time for at least $T_{tot} = 20\,\text{days}$. Run this simulation on an f-plane approximation (i.e., set $\beta = 0$). The choice of Δt is yours.

Note that you will need to solve the Poisson equation every time step of the simulation. For the solution of the first Poisson equation (i.e. the one for the initial condition), start the iterations from $\psi = 0$ everywhere. While starting from the values of ψ in the previous time step makes the convergence of the Poisson solution much faster, it is a dangerous game because you run the risk of setting a tolerance that is too high and almost never get to update ψ. I managed to get really fast runs without setting $\psi = 0$ between time steps by setting the tolerance to 10^{-7} (with the 101×101 grid, I think you need to reduce the tolerance when you reduce Δt). Try it and your simulation will run fast!

I suggest you always keep track of the maximum Courant number. Other things to keep track of are the kinetic energy $\|\mathbf{u}_g\|^2/2$ and the enstrophy ζ_g^2 (integrated over the entire domain). Also keep track of how many iterations you are using for the Poisson equation in each time step, to make sure your SOR is actually working. In fact, I suggest you produce plots of these 4 quantities as a function of time and hand them in in your report.

You may want to use numerical diffusion to improve your solution by filtering out the high-wavenumber noise from the computational mode and the aliasing. If so, all you need to do is add a term $+K_{\text{num}}\nabla^2\zeta_g$ to the right hand side of Eq. (1), and use a simple operator splitting scheme with leapfrog for advection of geostrophic and planetary vorticity and Euler forward with a time step equal to $2\Delta t$ for the artificial diffusion diffusion (this is similar to Modeling Task 4). In this case, also keep track of the von Neumann number during the simulation.

After your are done with the vortex merger, I want you to run three additional simulations. You do not have to include them in your report (but you can if you want to). They all have exactly the same configuration as the base case, except for one change in each:

Simulation 1 Reduce the radius of the vortices to $a = 100\,\text{km}$. Now the behavior should be very different based on the value of a/b.

Simulation 2 Switch the sign of one of the vortices. Now they form a dipole and the results should be completely different.

Simulation 3 Set $\beta = 2 \times 10^{-11}\,\mathrm{m}^{-1}\mathrm{s}^{-1}$ and observe how different the results are (of course, our closed box becomes a problem here).

Report

Please include in your report:

i. Figures: Figures that briefly document your solutions. A few plots of ζ_g and the time evolution of of the 4 control quantities described above are enough. If you want, you can also produce plots of velocity vectores, etc.

ii. Documentation: Please include in your report a brief description of your implementation. How did you choose Δt? What values did you use? How did you initialize ψ for the SOR solver? Did you use numerical diffusion? Which value did you use for K_{num}?

iii. Discussion: Please briefly discuss your results. Is energy and enstrophy conserved? I also want you to write one line or two about the other three simulations (only add figures if you want, but I just want you to tell me what happens in each case).

Please upload a **.pdf** file with your report and **.pdf** file with your code to the dropbox that will be available on the CCLE website for the course. If you also want to upload your original code, that is fine. Please, do not upload word files or other formats, as they do not work well in other operational systems.