Project 5, deadline December 10, 2017

The Rossby wave equation in a basin

Fall semester 2017

For this project you can collaborate with fellow students and you can hand in a common report. This project (together with projects 3 and 4) counts 1/3 of the final mark.

Theoretical background and description of the system

Large scale, time-dependent motion in the atmosphere and ocean is often conceptualized in terms of *Rossby waves*, which have scales of hundreds to thousands of kilometers. The waves are used, for example, to understand the meandering of the atmospheric Jet Stream and the adjustment of the ocean to changes in wind forcing. Rossby waves exist because of the Coriolis acceleration, which acts perpendicular to the velocity of a fluid parcel. The Coriolis acceleration varies with latitude, being negative in the Southern Hemisphere and positive in the Northern (and vanishes at the equator). As a result of this variation, fluid parcels experience a change in their spin or *vorticity* if they move to different latitudes.

We will use the shorthand notations

$$\partial_t = \frac{\partial}{\partial t},$$

$$\partial_x = \frac{\partial}{\partial x}$$

$$\partial_{xx} = \frac{\partial^2}{\partial x^2}$$

etc.

The equation governing the vorticity is:

$$\partial_t \zeta + \beta \partial_x \psi = 0. \tag{1}$$

where ζ is the vorticity and ψ is the streamfunction, which determines the velocities:

$$u = -\partial_y \psi \quad , \quad v = \partial_x \psi, \tag{2}$$

Here u is the velocity component in the east-west direction (x) and v in the north-south direction (y). The vorticity is defined:

$$\zeta = \partial_x v - \partial_y u = \nabla_H^2 \psi. \tag{3}$$

Thus the vorticity is the Laplacian of the streamfunction.

Then there is the Coriolis parameter, defined as:

$$f = 2\Omega \sin(\theta) \tag{4}$$

where θ is the latitude and Ω is the rotation rate of the Earth, $\Omega = 2\pi/day$. We often approximate f as a linear function, centered on a latitude θ_0:

$$f \approx f_0 + \beta y \tag{5}$$

where $f_0 = 2\Omega \sin(\theta_0)$, $\beta = 2\Omega \cos(\theta_0)/R_e$ and $y = R_e(\theta - \theta_0)$, if R_e is the Earth's radius. This linear representation is called the β-plane approximation, and accounts for the β term in (1).

Thus the vorticity equation can be written in terms of one variable, the streamfunction, thus:

$$\boxed{\partial_t \nabla_H^2 \psi + \beta \partial_x \psi = 0.} \tag{6}$$

This is the *barotropic Rossby wave equation.*

We will examine analytic and numerical solutions to the vorticity equation in a *periodic domain* and a *closed domain.* A periodic domain is one that "wraps around", like the atmosphere. A closed domain has walls, like the continents bounding the oceans.

Analytical solutions

Project 5a): Solution to the Rossby equation in a periodic domain. Consider the vorticity equation (6) in one dimension (we'll add the second dimension later). Say the periodic domain extends from $x = [0, L]$ (east-west). A suitable wave solution has the form:

$$\psi = A \cos\left(\frac{2n\pi x}{L} - \omega t\right), \tag{7}$$

where n is an integer ($= 1, 2, \cdots$), and where ω is the wave frequency and A is the amplitude.

Show that the solution is the same at the two boundaries, so that it satisfies the periodic boundary conditions. Then use this wave solution in equation (6) and solve for ω. Use the one-dimensional form for the vorticity:

$$\zeta = \partial_x v = \partial_{xx} \psi \tag{8}$$

The result is known as the wave *dispersion relation.*

The *phase speed* is the speed at which the Rossby waves move. To see how this comes about, consider the phase of the wave:

$$\theta = \frac{n\pi x}{L} - \omega t. \tag{9}$$

For instance, if $\theta = 2\pi$, then $\psi = A\cos(2\pi) = A$. This is a maximum or "high pressure" point. To see how this moves, we solve (9) for x:

$$x = \frac{\theta L}{2n\pi} + \frac{\omega t L}{2n\pi}. \tag{10}$$

Taking a time derivative, we obtain

$$c = \frac{dx}{dt} = \frac{\omega L}{2n\pi} \tag{11}$$

Calculate the phase speed for the Rossby wave, using the dispersion relation. Which direction is the wave moving?

Project 5b): Solution to the Rossby equation with solid boundaries. More appropriate boundary conditions for the ocean are no flow at the two end points. In the present problem, we can enforce this with simple Dirichlet conditions: $\psi = 0$ at $x = 0$ and $x = L$. Use a wave solution with the form:

$$\psi = A(x)\cos(kx - \omega t), \tag{12}$$

where k is the wavenumber. Substitute this into equation (6). You'll obtain terms multiplied by $\sin(kx - \omega t)$ and others multiplied by $\sin(kx - \omega t)$. Set each of these expressions to zero. One will give you the dispersion relation for the waves. The other will give you the form for $A(x)$, which must satisfy the boundary conditions. You see that k will be *quantized*: only specific values will be allowed.

Waves like these are known as Rossby "basin modes". The phase speed can be calculated as in the preceding problem. What is it? Discuss the structure of the wave (which is unusual).

Numerical solution

Now we consider solving equation (6) numerically. This involves two steps. If we know the velocity or streamfunction initially, we can advance the vorticity in time to a new time. Then the streamfunction at the new time is found by inverting (3). Having done this we may advance the vorticity to the next time level using (6), and so forth. We note that (6) is a *prognostic equation* and may therefore be solved numerically using a time stepping method. In contrast, (3) is a *diagnostic equation*. We will solve this using an elliptic solver.

To discretize the equations, we assume a grid of equally-spaced points:

$$x_j = j\Delta x \tag{13}$$

where Δx is the grid spacing. We discretize time in a similar way:

$$t^n = n\Delta t \tag{14}$$

where Δt is the time step. Thus $t^0 = 0$, $t^1 = \Delta t$, and so on.

We then approximate the derivatives by finite differences. For the spatial derivatives, we use centered-differences:

$$\partial_x \psi \approx \frac{\psi^n_{j+1} - \psi^n_{j-1}}{2\Delta x}, \tag{15}$$

$$\partial_{xx} \psi \approx \frac{\psi^n_{j+1} - 2\psi^n_j + \psi^n_{j-1}}{\Delta x^2}, \tag{16}$$

For the time stepping, we will test two methods. One involves a forward difference:

$$\partial_t \psi \approx \frac{\psi^{n+1}_j - \psi^n_j}{\Delta t}, \tag{17}$$

while the second involves a centered difference:

$$\partial_t \psi \approx \frac{\psi^{n+1}_j - \psi^{n-1}_j}{2\Delta t}. \tag{18}$$

Project 5c): Setting up the algorithms. Now we will examine numerical solutions in one dimension in the east-west direction. When calculating these, it is convenient to use *non-dimensional* forms of the equations:

$$\partial_t \zeta + \partial_x \psi = 0. \tag{19}$$

$$\partial_{xx} \psi = \zeta. \tag{20}$$

These are obtained by *scaling* each term by typical values. Doing this, the domain extends from $x = 0$ to $x = 1$, which is much simpler than worrying about realistic sizes. For the latter equation you can use your code from project 1.

For the boundary conditions, we'll consider both the periodic domain (that wraps around) and the bounded domain (with solid walls). For the bounded domain, the streamfunction is zero at the ends, while for the periodic one the boundary values can vary.

Write the algorithms for the two time-stepping methods and the equations you need to implement for the one-dimensional case, including the numerical analogues of the boundary conditions.

Project 5d): Truncation errors and stability. Determine the truncation errors of the two time-stepping schemes and investigate their stability properties. What do you conclude about the two schemes? What is the stability criterion for the stable algorithm.

Project 5e): Implementation. Now we'll implement the model, using both time-stepping schemes. Hereafter we'll use two (non-dimensional) initial conditions, a sine wave:

$$\psi(x,0) = \sin(4\pi x) \tag{21}$$

and a Gaussian:

$$\psi = \exp - \left(\frac{x - x0}{\sigma} \right)^2, \tag{22}$$

Here $\sigma = 0.1$ the width of the Gaussian.

Compare the time-stepping routines, using the sine wave in the periodic domain. Use $\Delta x = 1/40$ and Δt as determined from the results of (5d). Store the results at three times, in addition to the initial condition, e.g. $t = 0$, $t = 50$, $t = 100$ and $t = 150$. Is one of the time-stepping routines better than the other?

Project 5f): Visualization and discussion. Now we'll examine a suite of solutions, using the stable time-stepping algorithm. Consider the following:

a) Calculate the phase speed of the sine wave in the periodic domain. Do this by using a *Hovmuller diagram*. For this you construct a matrix with $\psi(x,t)$ at different times. $t = 0$ can be on the bottom row, then $t = \Delta t$ in the next row, $t = 2\Delta t$ in the next and so. Contour the matrix and extract the phase speed. How does it compare to the theoretical prediction in (5a)?

b) Now consider the sine wave in the bounded domain. How does this evolve differently? Rationalize the results using the theoretical prediction from (5b).

c) Now examine the Gaussian, in the periodic domain. How does the Hovmuller diagram differ? Can you find a phase speed? Try varying the width, σ. How does this affect the phase speed.

d) Lastly, describe the evolution of the Gaussian in the bounded domain. How does this compare to the sine wave in the same domain?

Project 5g): Moving to two dimensions. Extend the code you have developed here to two dimensions, that is, $\psi = \psi(x,y,t)$ and $\zeta = \zeta(x,y,t)$. It means that we deal with a $2+1$ dimensional problem. Our differential equations become

$$\partial_t \zeta + \partial_x \psi = 0, \quad t > 0 \text{ and } \quad x, y \in [0,1], \tag{23}$$

and

$$(\partial_{xx}\psi + \partial_{yy}\psi) = \zeta, \quad t > 0 \text{ and } \quad x, y \in [0,1], \tag{24}$$

where we now have made a model with a square lattice for x and y. The last equation is just another example of Poisson's equation which can be solved by Jacobi's method, or the Gauss-Seidel method or the so-called Successive Over Relaxation method discussed in chapters 6 and 10 of the lecture notes.

How would you extend the boundary conditions from one dimension to two dimensions? And can you find a closed form solution here as well? It is left to you to decide upon what kind of boundary conditions you deem appropriate.

Project 5h): Solving the two-dimensional equations numerically. Use an explicit scheme for (23). To solve (24) you need to use for example an iterative method like Jacobi's or Gauss-Seidel method or the Successive Over Relaxation (SOR) method. These methods are described in the lecture notes, see chapters 6 and 10.

Outline the algorithm for solving the two-dimensional equations and implement the scheme as function of Δx (assuming $\Delta x = \Delta y$) and Δt. Solve the equations numerically and give a critical discussion of your results. Compare your results with the closed-form answer if possible. Discuss the stability of the solution as function of different values of Δx and Δt.

References. A very good reference is the textbook by Winther and Tveito on partial differential equations. It is available online from the University library.

Introduction to numerical projects

Here follows a brief recipe and recommendation on how to write a report for each project.

- Give a short description of the nature of the problem and the eventual numerical methods you have used.
- Describe the algorithm you have used and/or developed. Here you may find it convenient to use pseudocoding. In many cases you can describe the algorithm in the program itself.
- Include the source code of your program. Comment your program properly.
- If possible, try to find analytic solutions, or known limits in order to test your program when developing the code.
- Include your results either in figure form or in a table. Remember to label your results. All tables and figures should have relevant captions and labels on the axes.
- Try to evaluate the reliabilty and numerical stability/precision of your results. If possible, include a qualitative and/or quantitative discussion of the numerical stability, eventual loss of precision etc.
- Try to give an interpretation of you results in your answers to the problems.
- Critique: if possible include your comments and reflections about the exercise, whether you felt you learnt something, ideas for improvements and other thoughts you've made when solving the exercise. We wish to

keep this course at the interactive level and your comments can help us improve it.

- Try to establish a practice where you log your work at the computerlab. You may find such a logbook very handy at later stages in your work, especially when you don't properly remember what a previous test version of your program did. Here you could also record the time spent on solving the exercise, various algorithms you may have tested or other topics which you feel worthy of mentioning.

Format for electronic delivery of report and programs

The preferred format for the report is a PDF file. You can also use DOC or postscript formats or as an ipython notebook file. As programming language we prefer that you choose between C/C++, Fortran2008 or Python. The following prescription should be followed when preparing the report:

- Use Devilry to hand in your projects, log in at `http://devilry.ifi.uio.no` with your normal UiO username and password and choose either 'fys3150' or 'fys4150'. There you can load up the files within the deadline.
- Upload **only** the report file! For the source code file(s) you have developed please provide us with your link to your github domain. The report file should include all of your discussions and a list of the codes you have developed. Do not include library files which are available at the course homepage, unless you have made specific changes to them.
- In your git repository, please include a folder which contains selected results. These can be in the form of output from your code for a selected set of runs and input parametxers.
- In this and all later projects, you should include tests (for example unit tests) of your code(s).
- Comments from us on your projects, approval or not, corrections to be made etc can be found under your Devilry domain and are only visible to you and the teachers of the course.

Finally, we encourage you to work two and two together. Optimal working groups consist of 2-3 students. For this specific report you can hand in a common report.