Geophysical Fluid Dynamics, theory and modeling

S. Thual, Fudan University 2019

Small Experiments with Shallow-Water Model

1 Instructions

This document proposes to code, analyze and play with a simple shallow-water model (with rotation) for geophysical fluid dynamics. The content was used for a midterm evaluation. Some necessary notions of Computational Fluid Dynamics are provided in the document Lecture_4_Handout.pdf.

1.1 Starting system

Consider the shallow-water equations:

$$
\begin{aligned}
&\tfrac{\partial}{\partial t}u - fv = -\tfrac{\partial}{\partial x}p \\
&\tfrac{\partial}{\partial t}v + fu = -\tfrac{\partial}{\partial y}p \\
&\tfrac{\partial}{\partial t}p + G(\tfrac{\partial}{\partial x}u + \tfrac{\partial}{\partial y}v) = 0
\end{aligned}
\tag{1}
$$

defined over the domain $\Delta x \leq x \leq L_x, \Delta y \leq y \leq L_y, \Delta t \leq t \leq L_t$. Here u, v are zonal velocities, p is pressure (or height), and f and G are parameters. We take values $f = 0$ and $G = 0.01$. Note that $G = \sqrt{c}$ where $c = 0.1$ is the propagation speed.

The boundary conditions are periodic in x and reflecting in y, which reads:

$$\begin{aligned}
&u(0,y,t) = u(L_x,y,t) \qquad && u(x,\Delta y,t) = 0 \qquad && u(x,L_y,t) = 0 \\
&v(0,y,t) = v(L_x,y,t) \qquad && v(x,\Delta y,t) = 0 \qquad && v(x,L_y,t) = 0 \\
&p(0,y,t) = p(L_x,y,t) \qquad && \tfrac{\partial p}{\partial y}(x,\Delta y,t) = 0 \qquad && \tfrac{\partial p}{\partial y}(x,L_y,t) = 0
\end{aligned} \tag{2}$$

We solve these equations over the model grid detailed at the end of Lecture_4_Handout.pdf, with: $x_i = i\Delta x$ for $i = 1, ..., N_x$, $y_i = j\Delta y$ for $j = 1, ..., N_y$, and $t_k = k\Delta t$ for $k = 1, ..., N_t$, where $L_x = \Delta x N_x$, $L_y = \Delta y N_y$ and $L_t = \Delta t N_t$. We take values $\Delta x = 2.5$, $N_x = 200$ ($L_x = 500$), $\Delta y = 2.5$, $N_y = 200$ ($L_y = 500$), and $\Delta t = 0.5$, $N_t = 10000$ ($L_t = 5000$).

We consider an explicit Euler method in time and a centered scheme 2nd order in space to approximate the derivatives.

The initial conditions read:

$$\begin{aligned}
u(x,y,0) &= 0 \\
v(x,y,0) &= 0 \\
p(x,y,0) &= \exp(-a_x(x-x_a)^2 - a_y(y-y_a)^2)
\end{aligned} \tag{3}$$

where $a_x = 0.002$, $a_y = 0.002$, $x_a = L_x/2$, $y_a = L_y/2$.

1.2 Task 1

Do the following:

a) Write a code in matlab to solve the discrete version of the shallow-water equations. Make sure that the code is readable (use comments) and that parameter values can easily be modified later on.

b) Integrate the numerical solutions over time.

c) Use the provided matlab function animatesw.m to create a time-animation of u,v,p (you may want to use frameskip=1 and dolevels=1 at first). Save results in a gif file with suitable name (for example test1.gif).

d) Compute the Courant number (for both x and y) and determine if the CFL conditions are satisfied.

Note: solution is provide in the file midterm.m

1.3 Task 2

Consider a new setup and repeat steps a,b,c,d from Task 1 (use a different matlab file !). The new setup must change at least three conditions among the followings:

1) Initial conditions (change the location of initial perturbation or consider entirely new initial conditions)

2) Boundary conditions (all periodic, all reflecting, or reflecting in x and periodic in y). You may even consider open boundary conditions.

4) Physical parameters c and/or f (you may try for example $f = 0.01$, $f = \beta(y - L_y/2)$ with $\beta = 2\,10^{-5}$, $f = -0.02\cos(\pi y/L_y)$, etc).

6) Resolution Δx, Δy, Δt and/or domain size L_x, L_y, L_t (try to keep same Courant number as in Task 1d or at least ensure CFL conditions)

8) Finite difference method in time: choose among methods in Table 2 from Lecture 4.

9) Any other modifications that are physically sound and correctly implemented.

Try to find inventive setups where something different happens that is interesting. Then prepare comments on the physical processes (propagations, boundary conditions, rotation...) and/or the numerical errors (numerical diffusion, aliasing..) and other practical aspects (computation cost, implementation, etc). Note that we have not covered rotation effects for $f \neq 0$ in the class yet so there is no need to understand these in details. It is also very good to show cases when the setup **doesnt work** and understand the reasons behind it. You will also get a higher score if the setup was more difficult to implement (open boundary conditions, implicit methods, etc).

Note: some examples are documented in the folder tests/