Computer Practical: Shallow Water Model

Paul Connolly and Robin Hogan, September 2023

1 Overview

In this computer practical, a shallow water model, implemented in Python, is used to demonstrate a number of phenomena, including gravity waves, barotropic instability, orographic Rossby waves, geostrophic turbulence, tsunamis and equatorially trapped waves. It can also be used perform rudimentary weather forecasts and investigate numerical stability issues.

2 Model formulation

The two-dimensional model is implemented in Python. It has three *prognostic* variables: the two wind components u and v, and the depth of the fluid layer h. One of the ways the model is forced is by height of the orography H, a fixed two-dimensional field. Figure 1 illustrates the meaning of h and H. In its default configuration, the model is intended to simulate a channel around the Earth between 20 and 70°N with a spatial resolution of 100 km (around one degree). The domain is periodic in the x direction and has solid north and south boundaries where $v = 0$ and h is fixed at its initial values.

The perimeter of a sphere at a latitude of 45° is $2\pi R_e \cos 45^\circ \cong 28$ thousand kilometres (comparable to the domain size here).

A conservation equation for quantity q in two dimensions may be written as

$$\frac{\partial q}{\partial t} + \frac{\partial(uq+U)}{\partial x} + \frac{\partial(vq+V)}{\partial y} = Q, \tag{1}$$

where Q represents any sources of q, and U and V represent any terms with spatial derivatives that depend on the prognostic variables in the model. In a shallow water context and in the absence of any sources, we wish to conserve the total volume and the total momentum in each direction, so want conservation equations for h, uh and vh. Therefore, the shallow water equations in conservative (or flux) form may be written as:

$$\begin{aligned}
\frac{\partial h}{\partial t} + \frac{\partial(uh)}{\partial x} + \frac{\partial(vh)}{\partial y} &= 0, & (2)\\
\frac{\partial(uh)}{\partial t} + \frac{\partial(u^2h+gh^2/2)}{\partial x} + \frac{\partial(uvh)}{\partial y} &= h\left(fv - g\frac{\partial H}{\partial x}\right), & (3)\\
\frac{\partial(vh)}{\partial t} + \frac{\partial(uvh)}{\partial x} + \frac{\partial(v^2h+gh^2/2)}{\partial y} &= h\left(-fu - g\frac{\partial H}{\partial y}\right), & (4)
\end{aligned}$$

where f is the Coriolis parameter and g is the acceleration due to gravity. In this model, the Coriolis parameter is modelled as varying linearly with y such that $f = f_0 + \beta(y - \bar{y})$. Thus $f = f_0$ in the middle of the domain in the y direction. The pressure gradient terms have been put within the spatial derivatives on the left-hand side.

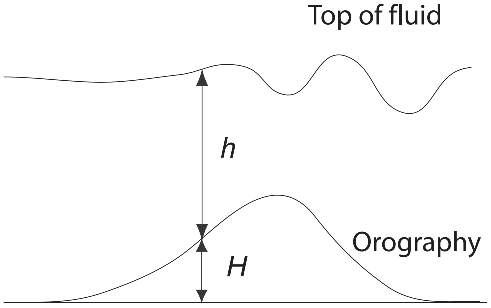

Figure 1: Schematic of the scenario being modelled, where h is the depth of the fluid and H is the height of the orography.

We integrate this scheme forward-in-time using the Lax-Wendroff scheme, which achieves reasonable accuracy for only modest complexity. The scheme is second-order accurate, requires only one previous time-step, and does not need any artificial diffusion in time or space to keep it stable. It works in two steps. Considering generic quantity q from (1), in the first step it estimates q at mid-points in space and time:

$$q^{n+1/2}_{i+1/2,j} = \frac{q^n_{i,j} + q^n_{i+1,j}}{2} - \frac{\Delta t}{2}\frac{(uq+U)^n_{i+1,j} - (uq+U)^n_{i,j}}{\Delta x}, \tag{5}$$

$$q^{n+1/2}_{i,j+1/2} = \frac{q^n_{i,j} + q^n_{i,j+1/2}}{2} - \frac{\Delta t}{2}\frac{(vq+V)^n_{i,j+1} - (vq+V)^n_{i,j}}{\Delta y}. \tag{6}$$

These can be thought of as forward-time, centred-space steps in one dimension, except that they move only half a step in time and space. The source term Q has been neglected in this first step for simplicity. These equations are applied to the quantities h, uh and vh to obtain $h^{n+1/2}_{i+1/2,j}$, $h^{n+1/2}_{i,j+1/2}$, $(uh)^{n+1/2}_{i+1/2,j}$, $(uh)^{n+1/2}_{i,j+1/2}$, $(vh)^{n+1/2}_{i+1/2,j}$ and $(vh)^{n+1/2}_{i,j+1/2}$.

From the midpoint quantities, we calculate the new values of $uq+U$ and $vq+V$ for each of the quantities being advected. For example, to advect uh we need $u^2h + gh^2/2$ at the midpoints, which can be computed from $(uh)^2/h + gh^2/2$. We then do the second step as follows:

$$q^{n+1}_{i,j} = q^n_{i,j} + \Delta t\left[-\frac{(uq+U)^{n+1/2}_{i+1/2,j} - (uq+U)^{n+1/2}_{i-1/2,j}}{\Delta x} - \frac{(vq+V)^{n+1/2}_{i,j+1/2} - (vq+V)^{n+1/2}_{i,j-1/2}}{\Delta y} + Q^n_{i,j}\right]. \tag{7}$$

This can be thought of a centred-time, centred-space step in two dimensions (which is what gives the Lax-Wendroff scheme second-order accuracy), except that the scheme here uses the values at midpoints to jump to the next time-step, rather than the leapfrog scheme which uses the adjacent full points to jump two time-steps.

3 Download the code

The code address on GitHub is:
`https://github.com/EnvModelling/shallow-water-model-practical`
Download the code using the `git clone` command, and change into the directory.

4 Running the model

The Python files required are `shallow_water_model.py`, a script that runs the model, `lax_wendroff.py`, a function which implements the numerical scheme, and `animate.py`, a script that animates the height of the top of the fluid $h+H$ and the relative vorticity $\zeta = \partial v/\partial x - \partial u/\partial y$. Later experiments also require the data files `reanalysis.mat` and `digital_elevation_map.mat`.
The instructions below describe running the model in Python:

1. Make sure your working directory is `python` within the code repository.
2. Edit Section 1 of `shallow_water_model.py` to configure the model on lines 43-70 and save it.
3. Run the model by typing `python3 shallow_water_model.py` at the prompt, followed by `enter` to run the model. This will run the model and generate an animated gif file of the output.
4. Finally, get the output file using SFTP by typing `get /tmp/<username>/animation.gif`

Table 1 describes the parameters in Section 1 of `shallow_water_model` that can be modified. Note that the typical fluid depth is $h = 10,000 \pm 500$ m, roughly corresponding to the depth of the troposphere. Multiplying this by g gives the surface pressure in Pa, so $h = 10$ km corresponds approximately to $p_s = 1000$ hPa. Highs and lows in h then correspond proportionately to highs and lows in surface pressure. Note that it its actually $h+H$ that is plotted (see Figure 1), which over orography is equivalent to the pressure at sea level.

Table 1: Parameters controlling the behaviour of the model. Note that n_x and n_y have to be the values shown here if you are to load orography (via `orography=EARTH_OROGRAPHY`) or an initial pressure field (via `initial_conditions=REANALYSIS`) from a file.

Variable	Default value	Description
g	9.81 m s^{-2}	Acceleration due to gravity
f_0	10^{-4} s^{-1}	Mean Coriolis parameter
β	1.6×10^{-11} m^{-1} s^{-1}	Meridional (latitudinal) gradient of Coriolis parameter
Δx, Δy	10^5 m	Horizontal grid spacing
n_x	254	Number of zonal (longitudinal) gridpoints
n_y	50	Number of meridional (latitudinal) gridpoints
`dt_mins`	1 minutes	Timestep
`output_interval_mins`	60 minutes	Time between outputs
`forecast_length_days`	4 days	Total simulation time
`orography`	`FLAT`	Set the orography (the H field)
`initial_conditions`	`GAUSSIAN_BLOB`	The initial height field
`initially_geostrophic`	`False`	Is the wind initially in geostrophic balance (or at rest)?
`add_random_noise`	`False`	Add random noise to the initial height field?

5 Experiments

5.1 Gravity waves

The simplest wave modelled by the shallow water equations is the gravity wave. Gravity waves can be excited by specifying initial conditions consisting of a non-uniform height field (specifically in this example a field that is flat except for a Gaussian blob of fluid towards the left of the domain) and the winds at rest. The settings to use are as follows:

- `orography = FLAT`
- `initial_conditions = GAUSSIAN_BLOB`
- `initially_geostrophic = False`
- `f=0.`
- `beta=0.`

Since gravity waves move fast, you may also wish to set:

```
output_interval_mins=15.
forecast_length_days=0.5
```

in order to clearly see them propagate. You will note that gravity waves have no significant rotational component so cannot be seen in the vorticity field. This is one way in which they may easily be distinguished from Rossby waves.
Run the model for both cases and get the output file as described in Section 4.

Exercise: Estimate the speed of the gravity waves: does it agree with the theoretical phase speed of $c = \sqrt{gh}$?

5.2 Tsunami

If we modify the previous experiment but add some orography then we model something like a tsunami where the fluid is now treated as a very deep ocean with a varying bottom topography. The Gaussian blob in the initial height field can be thought of as a displacement of the ocean surface due to an earthquake (although of much larger magnitude than in a real tsunami in order that we don't need to change the colour scale). The model will fail if H is so high that h goes negative, so we add topography in the form of a Gaussian sea mount which at its highest point is only 500 m below the surface:

- `orography = SEA_MOUNT`
- `output_interval_mins=60.`
- `forecast_length_days=4`

Run the model and get the output file as described in Section 4.

Exercise: What features of a tsunami are apparent in the gravity wave as it approaches shallower seas?

5.3 Barotropic instability

Barotropic instability occurs if the meridional (latitudinal) potential-vorticity gradient, $\partial[(f+\zeta)/h]/\partial y$, changes sign. This can occur in regions of strong shear, such as at the edges of a jet. The following example considers an idealized *Bickley jet* which is of the form $u \propto \mathrm{sech}^2(y-\bar{y})$, or equivalently $h \propto \tanh(y-\bar{y})$:

- `orography = FLAT`
- `initial_conditions = ZONAL_JET`
- `initially_geostrophic = True`
- `add_random_height_noise = True`
- `f=1.e-4;`
- `beta=1.6e-11;`
- `output_interval_mins = 60;`

The following alternative example illustrates how *cyclogenesis* can occur on a front with strong shear. The height field is initialized such that the initial wind is uniform westerly in the northern half of the domain and uniform easterly in the southern half of the domain, with a strong wind shear at the interface:

- `initial_conditions = SHARP_SHEAR`

Run the model for both cases and get the output file as described in Section 4.

Exercise: Try the simulation with `add_random_height_noise = False`. What happens? Why? (think about forces and how the model is initialised)

5.4 Jupiter's Great Red Spot

Jupiter's Great Red Spot is an anticyclone that has persisted for at least 350 years, presenting a puzzle as to how such a feature can be so long lived. One property of 2D geostrophic turbulence, which can be simulated by the shallow-water equations, is that over time small-scale structures often tend to merge into larger ones in an "upscale" cascade of energy. In this example we initialize the height field such that as we move from south to north the zonal (longitudinal) wind switches from easterly to westerly to easterly to westerly, just like conditions on Jupiter:

- `orography = FLAT`
- `initial_conditions = SINUSOIDAL`
- `initially_geostrophic = True`
- `add_random_height_noise = True`

Run the model and get the output file as described in Section 4.

5.5 Orographic Rossby waves

A dominant feature of the Earth's atmosphere is the presence of large-scale Rossby waves, or planetary waves. In the northern hemisphere, these are excited by flow over major mountain ranges such as the Himalayas and the Rockies. Let's first simulate the flow over an isolated Gaussian mountain:

- `orography = GAUSSIAN_MOUNTAIN`
- `add_random_height_noise = True`
- `initial_conditions = UNIFORM_WESTERLY`
- `initially_geostrophic = True`
- `forecast_length_days = 4;`

Run the model and get the output file as described in Section 4.

Exercise: Explain the initial meander in terms of vortex stretching and compression.

Exercise: Rerun the simulation with $f = 0$ s^{-1} and $\beta = 0$ m^{-1} s^{-1}. What happens? Why?

Exercise: Rerun the simulation with $f = 1 \times 10^{-4}$ s^{-1} and $\beta = 0$ m^{-1} s^{-1}. What happens? Why? (think about the restoring motion for the Rossby wave as it comes off the mountain.)

A more realistic simulation of northern hemisphere conditions on Earth can be realised with the following:

- `orography = EARTH_OROGRAPHY`

Run the model and get the output file as described in Section 4.

Exercise: What features of the mean large-scale flow on Earth can be identified?

5.6 Equatorially trapped waves

Numerous types of wave can occur in the Tropics that are trapped next to the equator because the Coriolis parameter goes through zero at this point. In this example, values of f_0 and β are chosen to simulate an *equatorial beta plane*, with the equator running horizontally through the middle of the domain. The pressure field is initialized such that there are easterlies near the equator, but with shear to either side on which instabilities can grow:

- $f_0 = 0$ s^{-1}
- $\beta = 2.5 \times 10^{-11}$ m^{-1} s^{-1} (i.e. write `beta = 2.5e-11`)
- `orography = FLAT`
- `initial_conditions = EQUATORIAL_EASTERLY`
- `add_random_height_noise = True`
- `output_interval_mins = 120.`
- `forecast_length_days = 8.`

Run the model and get the output file as described in Section 4.

5.7 Equatorial Kelvin wave

Another type of equatorially trapped wave that is observed in the ocean is the Kelvin wave. This is a kind of gravity wave that occurs in the oceanic mixed layer; because the mixed layer is shallow, it travels slow enough that the Coriolis force can act on it. Specifically the fact that the Coriolis force acts to the right of a parcel's direction of motion in the northern hemisphere and to the left of its direction of motion in the southern hemisphere means that a positive height anomaly propagating along the equator will be reinforced by the Coriolis force (and therefore long-lived), only if it is travelling in one particular direction (it travels east!). This is the equatorial Kelvin wave.

Rather than reduce the depth of the simulation (and the plotting script) to be equivalent to the oceanic mixed layer, the simplest way to simulate an equatorial Kelvin wave in this model is to perform the gravity wave experiment above, but at the equator and with β increased by a factor of 20 to make the Coriolis force relatively much more important.

- $f_0 = 0\ \mathrm{s}^{-1}$
- $\beta = 5 \times 10^{-10}\ \mathrm{m}^{-1}\ \mathrm{s}^{-1}$
- `orography = FLAT`
- `initial_conditions = GAUSSIAN_BLOB`
- `initially_geostrophic = False`
- `add_random_height_noise = False`
- `output_interval_mins=15;`
- `forecast_length_days=0.5;`

Run the model and get the output file as described in Section 4.

Exercise: Which is the equatorial Kelvin wave and which is the equatorial Rossby wave in your simulation?

5.8 Weather forecast

To get an idea of the suitability of the shallow-water equations to make weather forecasts, we will initialize them with the pressure field from the ECMWF reanalysis from 1 July 2000. This is actually from the southern hemisphere, but flipped to be run in the northern hemisphere with no orography. If you have just run simulations at the equator, be sure to reset f_0 and β to the values in Table 1.

- `orography = FLAT`
- `initial_conditions = REANALYSIS`
- `initially_geostrophic = True`
- `output_interval_mins=60;`
- `forecast_length_days = 4;`
- `f = 1e-4;`
- `beta = 1.6e-11;`
- `add_random_height_noise = False;`

Run the model and get the output file as described in Section 4.

Exercise: Initially in the forecast, the height field shows rapid fluctuations. What causes these?