# TP 1: Gravity waves

Please write a report (2-5 pages) on this activity containing your figures, results and some discussion of the different experiments: this will be used as CC for this UE. You should: explain what the problem to be addressed by the experiments is (*Introduction*), what experiments you performed (*Methods*), what the results show (*Results*), and discuss the physical processes that are involved (*Discussion / Conclusions*). For the second part, you might also discuss the approximations that have been made, and how the calculations could be made more realistic.

The report is due on Friday 23rd February (please upload to Moodle). You may write the report in French or English.

## Part 1: Waves in a shallow water model

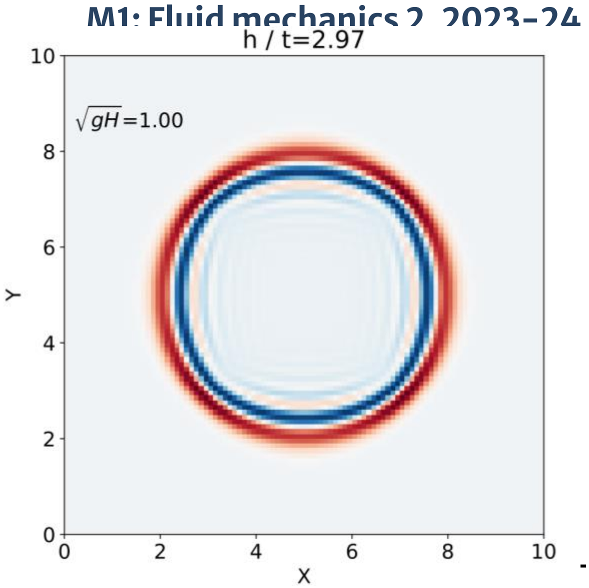
In this activity we will use a model that resolves the shallow-water equations to illustrate some properties of surface gravity waves.

### Download and run the model

The files are available on Moodle. You can choose to work in a notebook, or use the script.

If you use the script, the plot should be updated automatically for you to see the evolution of the fluid depth as the model runs. This is not possible in the notebook, but in both cases, it is possible to generate an mp4 file, which you can look at to see the animation at the end.

You can run the script at the command line with the command: `python rsw.py`




You can customize the size and resolution of the
Coriolis parameter (f), duration of the run (tend
etc.

By default, the code will output an animation ar
surface height hC. It will also output a numpy fil
and horizontal velocities. You can use (and mod
the execution of the script or use the outputted
You can also modify the code to output data at a
(netcdf, hdf5) if you prefer.

## Gravity wave experiments

### 1. Propagation speed

a) Run experiments of a geostrophic adjustment with a flat bottom and different depths H. Check how the propagation speed changes.
b) Run experiments with a sloping bottom. Check how the properties of the wave change as it moves over the topography.
c) Create your own topography (see examples such as topography = 'jump' ) to illustrate the different propagation properties of the waves depending on the depth of the flow.

### 2. Deformation and non-linear effects

Define an initial perturbation in the form of a local plane wave with a chosen wavenumber, k, propagating along the x direction. A plane wave is defined by:

$\eta = A \cos(kx - \omega t)$

a) Choose k and H to be in the "shallow-water" approximation. Check what happens if you increase or decrease k by a factor of 5.
a) Check what happens when you are not in a linear regime anymore (i.e. the initial perturbation is not small).

## Part 2: Ray tracing

During this second activity, you will implement the ray tracing equations to track the paths that would be taken by waves (rays) as they move from deeper to shallower topography.

You will need to download the file topography.pickle from Moodle. This is a data file, containing the topography for the ocean region around Finistère. You can load the file, and access the variables using the commands:

```
import pickle
data = pickle.load(open("topography.pickle", "rb"))
lat = data['latitude']  # 1D array, latitude data
lon = data['longitude'] # 1D array, longitude data
```

```
xdist = data['xdist']   # 1D array,
                        # distance along x-axis (longitude) in m
ydist = data['ydist']   # 1D array,
                        # distance along y-axis (latitude) in m
H = data['depth']       # 2D array, topography data, in m
                        # data is negative in the ocean, and positive
                        # over land
```

In order to trace the rays, we can use the following steps:

1. Choose a starting point somewhere in the region **[y0,x0]**. Store this in an array.
1. Choose values for the zonal and meridional wavenumbers **[kx0,ky0]** for your incident wave. You can choose any reasonable value. Wind waves generally have wavelengths in the range 1 – 1000 m. Store these values in arrays.
2. You will also need a time step for the calculation. To begin with, use dt=60s. You can try changing this later if you wish.
3. At your chosen location obtain the value of the ocean depth, H, using the supplied topography file. You can define a function for the interpolation at any given (y,x):

```
import scipy.interpolate as interp
Hji = interp.RectBivariateSpline(ydist,xdist,H) # function Hji: (y,x) -> H(y,x)
H0 = Hji(y0,x0)
```

4. Calculate the increment in distance **[dx,dy]** during one time step: $dx = \frac{\partial \Omega}{\partial k_x} dt$ You will need a formula for $\Omega(k_x, k_y, x, y)$: you may apply the shallow water approximation to the dispersion relation.
5. The supplied matrices, **xdist** and **ydist**, give the distance along the x-axis (longitude) and y-axis (latitude) starting from the point in the southwest corner with index [0,0] in the array. Use these matrices to find the new position of the wave after 1 time step using your results from step 4. Append this new position to the array of step 1, containing your starting position.
6. Find the value of **H** at your new location in the supplied topography file.
7. Update the wavenumber by adding the incremental change given by: $dk_x = -\left.\frac{\partial \Omega}{\partial H}\right|_k \frac{\partial H}{\partial x} dt$ to your previous wavenumber value (the formula for the meridional wavenumber, l, will be the same, with k replaced by l, and x replaced by y). You will need to calculate the gradient in H using the values of H at the points surrounding your grid point in the supplied topography data. You can also use the function np.gradient and then interpolate at any given (y,x) using the following functions:

```
dHdx,dHdy = np.gradient(np.sqrt(H),ydist,xdist)

dHdxji = interp.RectBivariateSpline(ydist,xdist,dHdx)
dHdyji = interp.RectBivariateSpline(ydist,xdist,dHdy)
```

Append your new wavenumber values to the arrays of step 2, containing your initial wavenumber values.

8. Repeat steps 5-8 until the waves reach the coast.

When you have finished, make a contour plot of the topography, and use your array of the position values to plot the path followed by the wave as it approaches the shore on the same figure. Make a separate plot to show how the wavenumber changes over time using your stored wavenumber values.