

## 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 reliability 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. As programming language we prefer that you choose between C/C++, Fortran or Python. The following prescription should be followed when preparing the report:

- Use Fronter to hand in your projects, log in at [blyant.uio.no](http://blyant.uio.no) and choose 'fellesrom fys3150'. Thereafter you will see an icon to the left with 'hand in' or 'innlevering'. Click on that icon and go to the given project. There you can load up the files within the deadline.

- Upload **only** the report file and the source code file(s) you have developed. 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.
- Comments from us on your projects, approval or not, corrections to be made etc can be found under your Fronter domain and are only visible to you and the teachers of the course.

Finally, we do prefer that you work two and two together. Optimal working groups consist of 2-3 students. You can then hand in a common report.

## Project 4, Two-dimensional wave equation and simple models for Tsunamis, deadline November 3 12am (mid-night)

In this project will first study the simple two-dimensional wave equation and compare our numerical solution with analytic results. Thereafter we introduce a simple model for a tsunami.

Consider first the two-dimensional wave equation for a vibrating square membrane given by the following initial and boundary conditions

$$\left\{ \begin{array}{ll} \lambda \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = \frac{\partial^2 u}{\partial t^2} & x, y \in [0, 1], t \geq 0 \\ u(x, y, 0) = \sin(\pi x) \sin(2\pi y) & x, y \in (0, 1) \\ u = 0 \text{ boundary} & t \geq 0 \\ \partial u / \partial t|_{t=0} = 0 & x, y \in (0, 1) \end{array} \right. .$$

The boundary is defined by  $x = 0$ ,  $x = 1$ ,  $y = 0$  and  $y = 1$ .

- Find the analytic solution for this equation using the technique of separation of variables.
- Write down the algorithm for the explicit method for solving this equation and set up a program to solve the discretized wave equation. Describe in particular how you treat the boundary conditions and initial conditions. Compare your results with the analytic solution. Use a quadratic grid.

Check your results as function of the number of mesh points and in particular against the stability condition

$$\Delta t \leq \frac{1}{\sqrt{\lambda}} \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1/2}$$

where  $\Delta t$ ,  $\Delta x$  and  $\Delta y$  are the chosen step lengths. In our case  $\Delta x = \Delta y$ . It can be useful to make animations of the results (a simple recipe with gnuplot and python for this is available under the project link for project 4 at the webpage).

We modify now the wave equation in order to consider a 2+1 dimensional wave equation with a position dependent velocity, given by

$$\frac{\partial^2 u}{\partial t^2} = \nabla \cdot (\lambda(x, y) \nabla u).$$

If  $\lambda$  is constant, we obtain the standard wave equation discussed in the two previous points. The solution  $u(x, y, t)$  could represent a model for water waves. It represents then the surface elevation from still water. We will model  $\lambda$  as

$$\lambda = gH(x, y),$$

with  $g$  being the acceleration of gravity and  $H(x, y)$  is the still water depth.

The function  $H(x, y)$  simulates the water depth using for example measurements of still water depths in say a fjord or the north sea. The boundary conditions are then determined by the coast lines as discussed in point d) below. We have assumed that the vertical motion is negligible and that we deal with long wavelenghts  $\tilde{\lambda}$  compared with the depth of the sea  $H$ , that is  $\tilde{\lambda}/H \gg 1$ . We will also neglect Coriolis effects.

You can discretize

$$\nabla \cdot (\lambda(x, y) \nabla u) = \frac{\partial}{\partial x} \left( \lambda(x, y) \frac{\partial u}{\partial x} \right) + \frac{\partial}{\partial y} \left( \lambda(x, y) \frac{\partial u}{\partial y} \right),$$

as follows using again a quadratic domain for  $x$  and  $y$ :

$$\frac{\partial}{\partial x} \left( \lambda(x, y) \frac{\partial u}{\partial x} \right) \approx \frac{1}{\Delta x} \left( \lambda_{i+1/2, j} \left[ \frac{u_{i+1, j}^l - u_{i, j}^l}{\Delta x} \right] - \lambda_{i-1/2, j} \left[ \frac{u_{i, j}^l - u_{i-1, j}^l}{\Delta x} \right] \right),$$

and

$$\frac{\partial}{\partial y} \left( \lambda(x, y) \frac{\partial u}{\partial y} \right) \approx \frac{1}{\Delta y} \left( \lambda_{i, j+1/2} \left[ \frac{u_{i, j+1}^l - u_{i, j}^l}{\Delta y} \right] - \lambda_{i, j-1/2} \left[ \frac{u_{i, j}^l - u_{i, j-1}^l}{\Delta y} \right] \right).$$

- c) Show that this equation has the same truncation error as the expressions used in a) and b) and that they result in the same equations when  $\lambda$  is a constant.

We assume that we can approximate the coastline with a quadratic grid. As boundary condition at the coastline we will employ

$$\frac{\partial u}{\partial n} = \nabla u \cdot \mathbf{n} = 0,$$

where  $\partial u / \partial n$  is the derivative in the direction normal to the boundary.

We are going to model the impact of an earthquake on sea water. This is normally modelled via an elevation of the sea bottom. We will assume that the movement of the sea bottom is very rapid compared with the period of the propagating waves. This means

that we can approximate the bottom elevation with an initial surface elevation. The initial conditions are then given by (with  $L$  the length of the grid)

$$u(x, y, 0) = f(x, y) \quad x, y \in (0, L),$$

and

$$\partial u / \partial t |_{t=0} = 0 \quad x, y \in (0, L).$$

We will approximate the initial elevation with the function

$$f(x, y) = A_0 \exp \left( - \left[ \frac{x - x_c}{\sigma_x} \right]^2 - \left[ \frac{y - y_c}{\sigma_y} \right]^2 \right),$$

where  $A_0$  is the elevation of the surface and is typically 1 – 2 m. The variables  $\sigma_x$  and  $\sigma_y$  represent the extensions of the surface elevation. In this project we will let  $\sigma_x = 80$  km and  $\sigma_y = 200$  km. The 2004 tsunami had extensions of approximately 200 and 1000 km, respectively.

The variables  $x_c$  and  $y_c$  represent the epicentre of the earthquake.

We need also to model the sea bottom and the function  $\lambda(x, y) = gH(x, y)$ . We assume that we can model the sea bottom as depicted in the following figure, with a water depth of 5000 m and a surface elevation of 2 m. We assume the sea bottom depends only on the

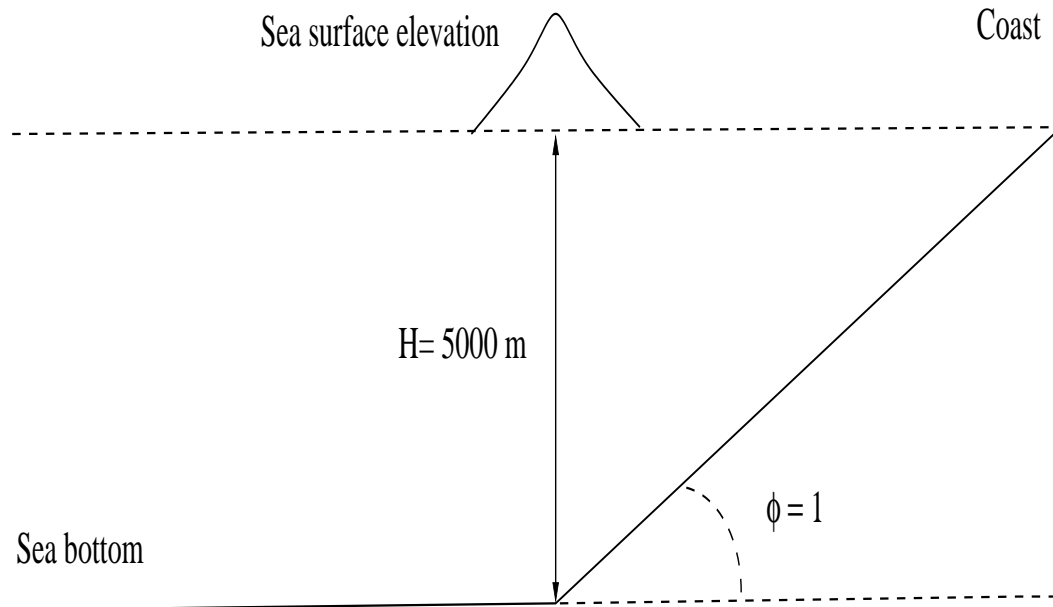


Figure 1: The sea bottom towards one of the coastlines has a shape with an inclination of 1 degree and depth where the earthquake takes place of 5000 m. The surface elevation is exaggerated on the figure.

variable  $x$  and has depth 5000 m before it starts increasing towards the coastline. We fix

the angle  $\theta = 1$  degree. From the figure you will be asked below to model the  $x$  dependence of  $H(x)$ .

Your tasks are as follows:

- d) Develop an algorithm for solving the new wave equation and write a program which implements it. Pay in particular attention to the implementation of the boundary conditions and the initial conditions. Figure out how to deal with the fictitious values in time and space for the discretized functions. You need also to find the functional form of  $H(x, y) = H(x)$ .

Be careful to scale the equations properly. With the depth of 5000 m, extensions  $\sigma_x = 80$  km and  $\sigma_y = 200$  km you need to figure out the proper dimensions of the grid  $L \times L$ . Scale the equations so that you can use dimensionless quantities.

With the above parameters, initial values and boundary conditions, study the temporal evolution of the wave towards the coastline. Comment your results. It can be useful to make animations of the results (a simple recipe with gnuplot and python for this is available under the project link for project 4 at the webpage).

It also important that you keep in mind the stability condition

$$\Delta t \leq \frac{1}{\sqrt{\max \lambda(x, y)}} \left( \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right)^{-1/2}$$

- e) We keep now the same shape of the sea bottom and the same parameters as in d), but we shift the center of the earthquake to the right, as indicated in Fig. 2.

The surface elevation is moved 40 km to the right. Which one of the two earthquakes will produce the largest impact (wave elevation) at the coastline? Comment your results.

- f) (**Optional/frivillig**) We keep the same shape of the sea bottom as in d) but instead of using the supplied Gaussian shape of the initial surface elevation we will now use realistic data for the surface elevation. These data, with the respective parameters are listed in the file initial.dat on the webpage of the course.

Write a program which reads in these data sets and use spline interpolation to find the desired values of the surface elevation at  $t = 0$ . Your mesh points may not coincide with the tabulated points and you may therefore need to interpolate between the data points.

To perform the interpolation you can use either the spline and splint functions included in lib.cpp or, based on your results from project 1, write your own cubic spline interpolation function.

Compare then the results you obtain in this case with those from d) and comment your results.

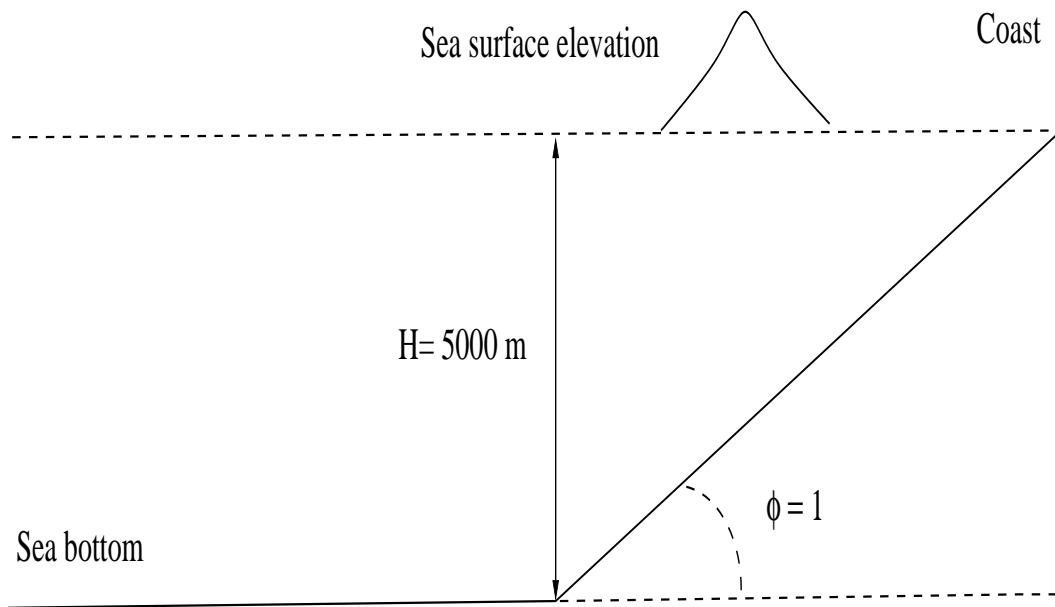


Figure 2: The sea bottom towards one of the coastlines has a shape with an indication of 1 degree. The center of the earthquake is shifted to the right with respect to Fig. 1 as indicated by the exaggerated surface elevation. In this part the surface elevation is moved 40 km to the right.