Finite Element Modelling of a Mega-Tsunami

Kristof Cools

November 18, 2022

1 Problem Statement

A megatsunami is created when a large land mass slides into the ocean. To understand its potential to cause harm we wish to model this phenomenon. In this project assignment you will:

- From the wave equation that models propagation of waves on the ocean, you will derive a weak formulation and Galerkin discretisation
- You will encounter various boundary conditions: encounters of waves with coastal lines will be modelled by a simple Dirichlet boundary condition.
- The domain of simulation (essentially the total surface of planet earth) needs to be bounded. You can do this in various ways, resulting in various levels of realism and accuracy. You can opt to enforce a Robin type absorbing boundary condition, a toroidal or spherical periodic boundary condition, or you can choose to model the earth's surface as an actual sphere, removing the need to bound the domain of simulation.
- You will use an open source meshing tool (gmsh) to create a discrete geometry that approximates the actual map of the world.
- You will report on your work and findings in a final report.

Let Ω denote the part of the earth's surface covered by oceans. The pressure u associated to a wave is a solution to the wave equation:

$$-\operatorname{div}\operatorname{grad} u - k^2 u = f \quad \text{in } \Omega \tag{1}$$

Here k is the wave number, which is related to the wavelength by $k = 2\pi/\lambda$. Assume λ to be 4,000 km. The right hand side is related to the source of the wave and can be chosen to be a quickly decaying Gaussian bell function centred on the epicentre.
The domain Ω is bounded on one hand by coastal lines Γ_1 and on the other hand by the end of the simulation domain (for all you flat-earthers out there: think of it as the edge of the world) Γ_2, such that $\partial\Omega = \bar{\Gamma}_1 \cap \bar{\Gamma}_2$.
We will model the interaction of the wave with the coastal lines Γ_1 by enforcing a homogeneous Dirichlet boundary condition:

$$u = 0 \quad \text{on } \Gamma_1 \tag{2}$$

As a first approximation, we will define an arbitrary line to act as the end of the world (a rectangular of circular line enclosing all continents) where we wish to enforce an absorbing boundary condition. Such a condition will cause any waves that impinge on this boundary to just keep going, without creating spurious reflections that would give away the presence of this artificial line.
Consider a plane wave solution in a one-dimensional setting:

$$u(x) = e^{-ikx} \tag{3}$$

In addition to being a solution to the wave equation, this function also solves $\partial u/\partial x + iku = 0$. Also note that the left travelling wave e^{+ikx} does not fulfill this condition. This suggests the following absorbing boundary condition to be enforced:

$$\frac{\partial u}{\partial n} + iku = 0, \quad \text{on } \Gamma_2 \tag{4}$$

2 Assigment

In order to solve this problem, answer the following questions:

1. Give the weak formulation of the problem (partial differential equation + boundary conditions). Explain which boundary conditions are essential and which are natural. Describe the function space over which the weak formulation is posed. Because of the absorbing boundary condition, there will be a term that involves an integral over the boundary.

2. Describe the finite dimensional space of finite elements you will be using to discretise the weak formulation. Describe carefully the basis functions for this space and in particular which boundary degrees of freedom you need to include and exclude.

3. Describe the spaces of local shape functions (i.e. the space of all possible restrictions of functions in the finite element space to a fixed triangle or segment).

4. Implement the finite element method for this choice of finite element spaces. Think of a suitable data structure to store the geometric data. It should be rich enough to provide access to vertex-triangle adjacency data and it should allow you to get access to segments in the two types of boundaries.

5. Assembly of the global interaction matrix should be based on a single iteration over the triangles of the domain and the segments its boundary. Using the assembly data (information on which local shape function contributes to which global degree(s) of freedom) scatter the local triangle or segment interactions into their correct entries in the global system matrix. For example, say global function 7 can be written in terms of local function 1 on triangle 13 and local function 1 on triangle 4, i.e.:

   $$f_7 = \phi_{13,1} + \phi_{4,1}. \tag{5}$$

   Now we know that the local interactions computed on triangle 13 contain interactions that will contribute to entries in row and column 7 of the global system matrix.

6. Create better meshes that (i) resemble more the outlines of the continents, and (ii) are made up from more (and smaller) triangles, resulting in a better approximation of the solution on the mesh. Device a numerical experiment to assess the convergence of the solution and report on your findings. Include in your report these studies and your recommendations on how to determine an appropriate number of triangles to use.

Obviously the behaviour at the boundary is not realistic. In addition we have not taken into account the curvature of the earth. To improve upon this situation, choose one of the following two avenues to increase the model accuracy and realism in a second modelling cycle:

1. Adapt the boundary condition on Γ_1 to be periodic: waves disappearing of the left of the map should reappear on the right:

$$u(x, c) = u(x, d) \tag{6}$$

with c, d the y-coordinates corresponding to the left and right boundary. Similarly, implement the boundary condition

$$u(a, y) = u(b, y) \tag{7}$$

to allow waves disappearing at the top/bottom of the simulation domain to reappear on the other side. Is this the behaviour you expect on a spherical surface? If not, what geometry are you modelling? Carefully list the natural and/or essential boundary conditions you have introduced.

2. Use gmsh (`https://gmsh.info`) to create a spherical surface and mesh that represents earth and its oceans. This extension is suited more for those of you interested in CAD. Some research in the required tools and methods is required. Note that the provided library CompScienceMeshes only reads gmsh file format 2. Take this into account when exporting your mesh.

3 Notes

- You can base your implemntation on the solver at

  `https://github.com/krcools/AP3001-FE-Lab`.

  This solver solves the Laplace equation with Dirichlet conditions on the complete boundary. Take not on how the Dirichlet boundary condition is implemented; Dirichlet boundary vertices are ignored during the assembly procedure. This requires a renumbering of the unknowns and a smart index lookup approach. The approach offered here is certainly not the only/best approach.

- The example is written in Julia. To run, navigate on the command line to the example directory, start julia, and run the example by executing `include("tsunami.jl")`. Make sure the project file `Project.toml` is in the same directory as the script. The first time you do this, this can take some time, because the example dependencies will be downloaded, installed, and compiled. You can either build on top of this example, or code your solution in your own Julia, Python, or Matlab package. For both Matlab and Python there are tools you can google for exporting/reading gmsh meshes into your scripts.

- Looking at the absorbing boundary condition it is clear that the matrices and the solution for u are complex valued. The field u should be interpreted as a field of phasors.

4 Report

You are allowed to work in groups of three.
Your final report should be more or less 10 pages. It should contain the following sections:

1. Introduction and Problem Statement

2. Mathematical Model and Discretisation

3. Algorithm, Data Structures, Implementation and Numerical Results

4. Conclusion, recommendations, and future directions

The report should be uploaded as a single pdf file to Ufora. Include in your report a link to a **private** (!!) github.com repository containing the code. Add user `krcools` as a collaborator so that I can access your work. Using your UGent email address, you can sign up for a GitHub Student Developer Pack, which should give you access to collaboration in private repositories.
The deadline for submission is Monday January 9, 2023.