Assignment 2: Shallow water gravity waves

ENME302, University of Canterbury

Semester 2, 2024

This assignment is an individual piece of assessment, and your report is to be submitted via the Ako | Learn course page by 15 October.

Problem description

We are concerned that an asteroid will impact Lake Taupō and the resulting waves might damage our yacht anchored near the lakefront.

The wave equation can describe the wave dynamics in shallow water[a]:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left(GH\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(GH\frac{\partial u}{\partial y}\right) + s \tag{1}$$

a For theory, refer to https://www.coastalwiki.org/wiki/Shallow-water_wave_theory.

where $u(x, y, t)$ is the surface displacement, G acceleration of gravity, $H(x, y)$ still water depth, and $s(x, y, t)$ source term. The initial displacement is given by $f(x, y)$ and initial velocity by $g(x, y)$.

Part 1: Code verification

We aim to develop and verify code for solving the wave equation with a constant coefficient within a unit square of $0\,\mathrm{m} \leqslant x, y \leqslant 1\,\mathrm{m}$ using homogeneous Dirichlet boundary conditions around the perimeter. The governing equation simplifies to:

$$\frac{\partial^2 u}{\partial t^2} = GH\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + s \tag{2}$$

Discretise using the finite difference method with the central difference, explicit scheme, a uniformly spaced grid, $\Delta x = \Delta y$, and write a Python script to solve the problem. Include your discretisations and code as file attachments to your report. Set $GH = 100\,\mathrm{m}^2/\mathrm{s}^2$.

Verification with analytical solution

Verify your code using the following analytical solution:

$$u_a = \sin(\pi x)\sin(\pi y)\cos\left(\pi t\sqrt{2GH}\right) \tag{3}$$

where the initial displacement is $f(x,y) = u_a(x,y,0\,\mathrm{s})$, likewise the initial velocity $g(x,y)$ is evaluated from the time derivative at $t = 0\,\mathrm{s}$. There is no source term involved in this analytical solution, $s = 0\,\mathrm{m/s^2}$.

Q1 Visualise the numerical and analytical solutions using 3-D surface plots after one oscillation. Include a plot of the numerical solution, and a plot of the difference between the two solutions.

Q2 Plot line graphs of $u(x, 0.5\,\mathrm{m}, t)$ against x for several times across half an oscillation. Include the numerical solution with lines and the analytical solution with markers.

Q3 Plot the discretisation error ε against grid spacing Δx using a constant Courant number and final time. Fit a suitable curve to quantify the convergence rate.

Q4 Discuss your results, including: physical sense of solutions, comparison between numerical and analytical solutions, convergence rates, choice of Courant number, and any other notable results you found.

Verification with method of manufactured solutions

Verify your code using the method of manufactured solutions[a] to ensure that $g(x,y)$ and $s(x,y,t)$ are also implemented correctly. Consider the following two manufactured solutions:

a *If you are interested in the details, refer to Section 2.2.3 of Langtangen, H.P. and Linge, S., (2017) Finite Difference Computing with PDEs: A Modern Software Approach.*

$$\begin{aligned} u_{e1} &= (x - x^2)(y - y^2)(1 + \frac{1}{2}t) \\ s &= GH(t+2)(x - x^2 + y - y^2) \end{aligned} \tag{4a}$$

$$\begin{aligned} u_{e2} &= (x - x^4)(y - y^4)(1 + t) \\ s &= 12GH(t+1)\left(y^2(x - x^4) + x^2(y - y^4)\right) \end{aligned} \tag{4b}$$

Q5 Visualise the evolution of the manufactured solutions using line graphs of $u(x, 0.5\,\mathrm{m}, t)$ against x for several times. Include the numerical solution with lines and the manufactured with markers.

Q6 Plot the error ε against time t for each manufactured solution.

Q7 Discuss your results, including: physical sense of solutions, rates of accumulating discretisation error, and any other notable results.

Verification with COMSOL

Develop a COMSOL model to simulate the second manufactured solution, Equation 4b, and compare with your numerical solutions.

Q8 Demonstrate mesh convergence using a suitable figure.

Q9 Plot the displacement at $u(0.5\,\text{m}, 0.5\,\text{m}, t)$ against t. Include the Python solution with lines and COMSOL with markers.

Q10 Discuss your results, including: mesh convergence analysis, comparison of results with your Python code, and any other notable results.

Part 2: Asteroid impacting Lake Taupō

Now that we have confidence in our codes working as expected from our verifications, we are going to model the waves generated from an asteroid impacting Lake Taupō which has a variable lake level (still water depth), refer to Equation 1 with $s = 0\,\text{m/s}^2$. We are going to model the lake as a circle[a] of radius $R = 14\,\text{km}$ with homogeneous Neumann boundary conditions (yielding more realistic reflections of waves at the lakefront). We are to approximate[b] the depth (units: m):

$$H(r) = -(3.366 \times 10^{-7})r^2 - (8.4983 \times 10^{-3})r + 186 \tag{5}$$

where $r = \sqrt{x^2 + y^2}$ is the radial distance from the centroid of the lake. Assume that the asteroid impact results in a Gaussian distributed dip at the centre of the lake given by:

$$f(r) = -A_0 \exp\left(-\frac{r^2}{2A_s^2}\right) \tag{6}$$

where A_0 is the peak dip, and A_s standard deviation of the dip. The initial speed of the deflection is $g(r) = 0\,\text{m/s}$.

a *Surface area of* $616\,\text{km}^2 \implies R = \sqrt{\frac{616\,\text{km}^2}{\pi}}$

b *By assuming the maximum depth of* $186\,\text{m}$ *occurs at the centroid, the average depth of* $110\,\text{m}$ *at the midpoint, and the lakefront is at a depth of* $1\,\text{m}$, *and then fitting a quadratic curve.*

Axisymmetric assumption

We start by exploiting axisymmetry of the problem by modelling the wave equation in 1-D with cylindrical coordinates:

$$\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial r}\left(GH\frac{\partial u}{\partial r}\right) + \frac{GH}{r}\frac{\partial u}{\partial r} \tag{7}$$

Develop a COMSOL model for this axisymmetric 1-D case. Remember to ensure the governing equation, boundary and initial conditions are implemented correctly in your model.

Q11 Perform a mesh convergence study on the displacement at our yacht, $u(r = R, t)$, for at least two peaks[c]. Set $A_0 = 10\,\mathrm{m}$, and $A_s = 1000\,\mathrm{m}$. Create a figure to summarise your mesh convergence study. Create another figure to visualise the evolution of the displacement throughout the lake $u(r, t)$ for your chosen mesh.

c *Note: after the first wave reflects back from the lakefront, a second wave hits the yacht later on.*

Q12 Perform a sensitivity analysis on the two parameters: A_0 and A_s. First, plot $u(r = R, t)$ for various A_0, and second, plot for A_s.

Q13 Given the hydrograph in `hydrograph1D.dat`, recover the corresponding A_0 and A_s parameters (to three significant figures) through an optimisation study[a]. Plot the matching hydrographs on a figure with the parameters stated.

a *You could use a parameter estimation study step to simplify the process. For step by step instructions, refer to* `https://www.comsol.com/blogs/how-to-use-the-parameter-estimation-study-step-for-inverse-modeling`.

Q14 Discuss your results, including: physical sense of solutions, time taken for waves to reach yacht, maximum deflections experienced by the yacht, mesh convergence and numerical settings, hydrograph sensitvity on A_0 and A_s, and any other notable results.

Circular 2-D domain

We will finally model the lake without the axisymmetric assumption, using a 2-D Cartesian domain and solving the governing equation given by Equation 1 with $s = 0\,\mathrm{m/s^2}$. The yacht remains at the lakefront $u(x = R, y = 0\,\mathrm{m}, t)$. Develop a COMSOL model for this case.

Q15 Generate a structured OH-grid, and show the resulting mesh.

Q16 Perform a mesh convergence study, similar to Q11, and compare your results to the 1-D case.

Q17 We are worried about damage to our yacht due to these potential waves generated from an asteroid. We decide to alter the lakebed by digging (increasing $H(x, y)$) and dumping (decreasing $H(x, y)$) dirt nearby. We must avoid detection, and so the dirt cannot create an island ($H(x, y) \geqslant 1\,\mathrm{m}$), and the mass of dirt is conserved. Lastly, we aim to be efficient and move as little dirt for as much wave reduction as practical and therefore we will implement an optimisation study. Show suitable figures of your resulting lakebed topography and impact on the wave dynamics.

Q18 Discuss your results, including: physical sense of solutions, mesh convergence, the methodology and outcomes of your wave mitigation attempt, a critical review of the modelling assumptions made in this assignment (are they valid?), and any other notable results.

Report

Write your report by answering each of the questions sequentially. Remember to proofread your text: conciseness, formatting, grammar, spelling and clarity of figures. Include the following supplementary material (file attachments) to your submission:

- Discretisation working (scanned or a photo is suitable).
- Commented Python scripts.
- COMSOL models[a].
- Animation of wave dynamics for Q17.
- ENME302-related meme.

a *May require deleting solution data to reduce file size for* `.mph`

General hints & tips:

- Assume a constant time step size Δt that depends on the node spacing and stability criterion (i.e. Courant number).
- Lambda functions are helpful for defining brief functions. Note, when using lambda functions, for example for the initial condition $f(x, y)$, the input variables x and y are coordinates in space, and not the indices i and j.
- Local functions might be helpful for reusing code, for example parameter sweeps for mesh convergence studies.
- The discretisation error for each uniform grid resolution can be measured at the same final time using:

$$\varepsilon = \sqrt{\Delta x^2 \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} \left(u_{i,j}^{N_t} - u(x_i, y_i, N_t) \right)^2} \tag{8}$$

  where N_x, N_y, N_t are the number of points in space and time.
- The first argument of Python matrices correspond to the row, the second for column, and third for depth. However we usually use i for an index in the x-direction instead.[b] In this assignment, consider using $u(j, i, n)$ where n is the time level.
- For debugging, try running short snippets of your complete code, in order to find the cause of the problem (if not immediately obvious from the error message). Use a small grid (large Δx) and work through the same equations on paper to check you're getting the answer you expect from the code.

b *In order to have consistent names for the indices i and j, we have used $T(j, i)$ in Section 4 of the lecture notes (i.e. $T(1, 1)$ is the origin located at the lower left corner).*