Due May 1, 2024

This project examines the numerical solution of the wave equation in a liquid in two dimensions, subject to various boundary conditions. Ultimately you will account for variable depth and see how it changes the propagation of the waves.

1 2-D Wave equation

Let's examine the propagation of a 2-D wave confined to a rectangular region of dimension $0 \le x \le L_x$ by $0 \le y \le L_y$. Since the surface of the liquid now varies with both x and y location and with time t, we will refer to $u(x_i, y_j, t_k) = u^k_{i,j}$. If one assumes the depth of the fluid measured down from the equilibrium surface position is $h(x, y)$, then the governing partial differential equation is

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left(h(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(h(x,y)\frac{\partial u}{\partial y}\right). \tag{1}$$

The two initial conditions will be a flat surface at equilibrium, $u(x, y, 0) = 0$, and zero velocity, $u_t(x, y, 0) = 0$. To simplify your code, assume "free-floating" conditions on three out of the four sides of the rectangular region in the x-y plane. Specifically, $u_x(L_x, y, t) = 0$, $u_y(x, 0, t) = 0$, $u_y(x, L_y, t) = 0$. The remaining boundary condition along $x = 0$ is, again, a short sinusoidal wiggle. Specifically $u(0, y, t) = \sin(2\pi t)$ for $0 \le t \le 0.5$, and $u(0, y, t) = 0$ for $0.5 \le t$. You may also assume that $\Delta x = \Delta y$.

Note that in this document, we refer to $u(x, y, t)$ values at different times and locations as $u^k_{i,j}$. However in lab we were using Matlab variable names such as `uInit`, `uCur`, and `uFut` to visually simplify the appearance of the code. Relative to any given spatial location (x_i, y_j) and time t_k, the specific translation is `uInit(i,j)` $= u^{k-1}_{i,j}$, `uCur(i,j)` $= u^k_{i,j}$, and `uFut(i,j)` $= u^{k+1}_{i,j}$.

1.1 2-D uniform depth

To start the 2-D analysis, take a look at the simple case of uniform depth, $h(x, y) = 1$. In this case, the general 2-D wave equation (1) becomes

$$\frac{1}{c^2}\frac{\partial^2 u}{\partial t^2} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \tag{2}$$

- Convert (2) to finite difference form using the notation $u(x_i, y_j, t_k) = u^k_{i,j}$. The result should involve values of $u^k_{i,j}$ at five spatial locations in the t_k plane. It should also involve one value of u in the t_{k-1} plane, and one value in the t_{k+1} plane, both at location (x_i, y_j). Except for the very first time step $t = \Delta t$, all $u^k_{i,j}$ values are assumed to be known for times t_{k-1} and t_k. Hence the only unknown $u^k_{i,j}$ would be at time t_{k+1}.
- Call your code for this `Wave2D.m`. You may keep your code simple by assuming "free–floating" boundary conditions on the top, bottom and right boundaries of the rectangular region. In other words, $u_n = 0$ where n is x on the right side, and n is y on the top and bottom.
- Make a nice illuminated surface plot of the fluid surface over the rectangular region for each time step. You might want to use the "`AKWaterColormap.mat`" colormap file that is included in this distribution package.

1.2 2-D variable depth

Finally, address the more general problem of solving (1) on a rectangular region with the same boundary and initial conditions from the previous section.

- Convert (1) to finite difference form. The result should still involve five values of u in the t_k plane. It should also still involve one value of u in the t_{k-1} plane, and one value in the t_{k+1} plane, both at location (x_i, y_j). However, the coefficients on the various u terms should now involve spatially local values of $h(x, y)$, dh/dx and dh/dy.

- Make a copy of `Wave2D.m` and name it `Tsunami2D.m`. Modify this file to incorporate the finite difference version of (1).

- Your `Tsunami2D.m` file should call a supporting file named `Depth2D.m` that accepts two-dimensional arrays of x and y coordinates and returns three two-dimensional arrays of h, $\partial h/\partial x$ and $\partial h/\partial y$ values at specific (x, y) locations.

  Be careful to keep the bottom of your "ocean" below the water surface. In other words, don't let $h(x, y)$ become negative at any location in your $L_x \times L_y$ region.

- You may still simplify your `Tsunami2D.m` code by assuming "free-floating" boundary conditions on the top, bottom and right sides of the rectangular region. In other words, $u_n = 0$ where n is x on the right side, and n is y on the top and bottom.

- Make a nice illuminated surface plot of the surface of the fluid over the rectangular region for each time step.

Electronic work should:

- have all code and any output inside a directory (folder) named `YourLastNamePR02`. This directory should be "zipped" in an archive file named `YourLastNamePR02.zip` and submitted via email to **adam@colorado.edu** with a header **YourLastName Submission Proj02**.

- contain the following comment lines at the beginning of each file, (with the appropriate information filled in):

```
% A one-line description of the program
% Your name
% Today's date
% APPM 3050, Project #02
```

- contain sufficient comments to clearly explain what is being done at various points in the program.