Finite difference simulation of 2D waves

Obligatory project No. 1 in IN5270/IN9270

1 General information

- Collaboration/discussion among students is encouraged, but each student should submit a set of files, including a project report, which are programmed/written by her/himself.
- Please organize the your submission as a directory named `wave_project` to hold all your files of this project. Make suitable subdirectories. Include a `README` file with a short overview of the different files. (Info about when, where and how to make the submission will be given at the semester webpage.)
- Write a short report summarizing the main results. LaTeX is probably the preferred format, but there are several other options[1] too. Regardless of format, the report must be in an easy-to-read format like PDF or HTML (or Jupyter notebook).

Background material

> **Main source of information.**
>
> The various building blocks needed in this project can be found in Chapter 2 of the OpenAccess textbook *Finite Difference Computing with PDEs.*

Depending on your familiarity with finite difference methods before this course, it might be useful to consult Chapter 1 of the same textbook, which describes the fundamentals of the time discretization needed in the present project.

[1] `http://hplgit.github.io/teamods/writing_reports/index.html`

2 The core parts of the project

2.1 Mathematical problem

The project addresses the two-dimensional, standard, linear wave equation, with damping,

$$\frac{\partial^2 u}{\partial t^2} + b\frac{\partial u}{\partial t} = \frac{\partial}{\partial x}\left(q(x,y)\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(q(x,y)\frac{\partial u}{\partial y}\right) + f(x,y,t)\,. \qquad (1)$$

in a rectangular spatial domain $\Omega = [0, L_x] \times [0, L_y]$. The associated boundary condition is the homogeneous Neumann condition

$$\frac{\partial u}{\partial n} = 0\,. \qquad (2)$$

The initial conditions are

$$u(x,y,0) = I(x,y), \qquad (3)$$

$$u_t(x,y,0) = V(x,y)\,. \qquad (4)$$

2.2 Discretization

Derive the discrete set of equations to be implemented in a program:

- the general scheme for computing $u_{i,j}^{n+1}$ at interior spatial mesh points,
- the modified scheme for the first step,
- the modified scheme at boundary points (first step and subsequent steps), unless you use the interior scheme also at the boundary with extra ghost cells.

2.3 Implementation

Implement the numerical method for the PDE problem in a program. You may use `wave2D_u0.py`[2] as a starting point (this program solves the 2D wave equation with constant wave velocity and $u = 0$ on the boundary and is explained in the textbook). It is optional to also include a vectorized version of the implemention, in addition to a scalar (pointwise) version.

3 Verification

3.1 Constant solution

1. Construct a test case with constant solution $u(x,y,t) = U$, where U is a non-zero constant. (Hint: Fit the parameters f, b, I, and V such that

[2] `http://tinyurl.com/nm5587k/wave/wave2D_u0/wave2D_u0.py`

$u(x,y,t) = U$ fulfills the PDE problem. For simplicity, you can choose $q(x,y) = 1$.)

2. Show that the constant solution is also an exact solution of the discrete equations.

3. Make a test function to verify your code for this special case.

4. Inject at least two types of possible bugs in the implementation of the mathematical formulas. See how many of them that lead to a wrong non-constant solution.

3.2 Standing, undamped waves

The goal here is to compute the error and see how it approaches zero as $\Delta t, \Delta x, \Delta y \to 0$, with the help of an exact analytical solution of the PDE. (Please read about "Method of manufactured solution" in the textbook.)

With no damping ($b = 0$) and constant wave velocity ($q(x,y)$ being a constant), our wave equation problem without any source term ($f(x,y,t) = 0$) admits a standing wave solution:

$$u_{\mathrm{e}}(x,y,t) = A\cos(k_x x)\cos(k_y y)\cos(\omega t), \quad k_x = \frac{m_x\pi}{L_x},\ k_y = \frac{m_y\pi}{L_y}, \tag{5}$$

for arbitrary amplitude A, arbitrary integers m_x and m_y, and a suitable choice of ω. This solution can be used to test the convergence rate of the numerical method.

Compute the true error $e^n_{i,j} = u_{\mathrm{e}}(x_i, y_j, t_n) - u^n_{i,j}$ on a series of refined meshes. The physical parameters (A, m_x, m_y) can be kept at some chosen values. A suitable error norm can be

$$E = ||e^n_{i,j}||_{\ell^\infty} = \max_i \max_j \max_n |e^n_{i,j}|,$$

Introduce a common discretization parameter h such that Δt, Δx, and Δy are proportional to h. This leads to an error model $E = \hat{C}h^r$ for constant $\hat{C}$ and r. Theoretical analysis (e.g., via truncation errors) leads to the convergence rate $r = 2$. Compute a sequence of r values by comparing two consecutive experiments (as shown in the course material) and see if r approaches 2.

3.3 Waves with damping and variable wave velocity

Choose a non-constant $q(x,y)$ and find $f(x,y,t)$, along with suitable choices of A, B, ω and d, such that the following formula

$$u_{\mathrm{e}}(x,y,t) = (A\cos(\omega t) + B\sin(\omega t))\,e^{-dt}\cos(k_x x)\cos(k_y y), \quad k_x = \frac{m_x\pi}{L_x},\ k_y = \frac{m_y\pi}{L_y} \tag{6}$$

is a solution to the general 2D wave equation problem with damping and variable wave velocity. Find corresponding $I(x,y)$ and $V(x,y)$, and make a convergence test that recovers the expected convergence rate.

Hint. You may want to explore sympy for automating the analytical work. The following is an example of how to use sympy:

```
>>> from sympy import *
>>> x = Symbol(’x’)
>>> q=x**2
>>> u=sin(x)
>>> r = diff(q*diff(u, x), x)   # Derivative: (q*u_x)_x
>>> simplify(r)
x*(-x*sin(x) + 2*cos(x))
```

4 Investigate a physical problem

The purpose of this part is to explore what happens to a wave (non-damping, zero source term) that enters a medium with different wave velocities. A particular physical interpretation can be wave propagation of a tsunami over a subsea hill. The unknown $u(x, y, t)$ is then the elevation of the ocean surface, and the boundary condition $\partial u/\partial n = 0$ means that the waves are perfectly reflected, because of a steep hill at the shore, or the condition expresses symmetry in the solution. The square of the wave velocity is in this case given by $q(x, y) = gH(x, y)$, where g is the acceleration of gravity and $H(x, y)$ is the stillwater depth.

Hint: It can be wise to take a look at Problem 2.24 in the textbook first, because that 1D program, which corresponds to the present 2D problem, allows for much faster experimentation with parameters and effects.

The initial surface (which is symmetric in the y direction) is taken as a smooth Gaussian function

$$I(x; I_0, I_a, I_m, I_s) = I_0 + I_a \exp\left(-\left(\frac{x - I_m}{I_s}\right)^2\right), \tag{7}$$

with $I_m = 0$ reflecting the location of the peak of $I(x)$ and I_s being a measure of the width of the function $I(x)$ (I_s is $\sqrt{2}$ times the standard deviation of the familiar normal distribution curve). The second initial condition adopts $V(x, y) = 0$.

Three different bottom shapes can be investigated (such that $H(x, y) = H_0 - B(x, y)$). A 2D Gaussian hill can be modeled by

$$B(x, y; B_0, B_a, B_{mx}, B_{my}, B_s, b) = B_0 + B_a \exp\left(-\left(\frac{x - B_{mx}}{B_s}\right)^2 - \left(\frac{y - B_{my}}{bB_s}\right)^2\right), \tag{8}$$

where b is a scaling parameter: $b = 1$ gives a circular Gaussian function with circular contour lines, while $b \neq 1$ gives an elliptic shape with elliptic contour lines.

A less smooth hill is modeled by the "cosine hat" function

$$B(x,y;B_0,B_a,B_{mx},B_{my},B_s) = B_0 + B_a \cos\left(\pi\frac{x-B_{mx}}{2B_s}\right)\cos\left(\pi\frac{y-B_{my}}{2B_s}\right), \tag{9}$$

when $0 \leq \sqrt{x^2+y^2} \leq B_s$ and $B = B_0$ outside this circle.

A more dramatic hill shape is a box:

$$B(x,y;B_0,B_a,B_m,B_s,b) = B_0 + B_a \tag{10}$$

for x and y inside a rectangle

$$B_{mx} - B_s \leq x \leq B_{mx} + B_s, \quad B_{my} - bB_s \leq y \leq B_{my} + bB_s,$$

and $B = B_0$ outside this rectangle. The b parameter controls the rectangular shape of the cross section of the box.

Investigate how different hill shapes, different sizes of the water gap above the hill, and different resolutions $\Delta x = \Delta y = h$ and Δt influence the numerical quality of the solution. One anticipates that the less smooth hill shapes will introduce more numerical noise. Presenting the results as movies (or series of plots) of the surface elevation is effective.