Compulsory project 1, in INF5270

Soran Hussein Mohmmed / soranhm

September 2018

Finite difference simulation of 2D waves

2 The core parts of the project

The project is to addresse the two-dimensional, standard, linear wave equation, with damping:

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

with boundary condition: $\frac{\partial u}{\partial n} = 0$, inital conditions: u(x,y,t) = I(x,y), u_t(x,y,t) = V(x,y), and use it in different situations.

2.2 Discretization

The discrete set:

$$[D_t D_t u + bD_t u = D_x(qD_x u) + D_y(qD_y u) + f]^n_{i,j}$$

After some calculation and inserting the different parts we get:

$$\frac{u^{n+1}_{i,j} - 2u^n_{i,j} + u^{n-1}_{i,j}}{\Delta t^2} + b\frac{u^{n+1}_{i,j} + u^{n-1}_{i,j}}{2\Delta t} = \frac{1}{2}\Big\{\frac{1}{\Delta}x^2((q_{i+1,j} + q_{i,j})(u^n_{i+1,j} - u^n_{i,j}) - (q_{i,j} + q_{i-1,j})(u^n_{i,j} - u^n_{i-1,j}))$$
$$\frac{1}{\Delta}y^2((q_{i,j+1} + q_{i,j})(u^n_{i,j+1} - u^n_{i,j}) - (q_{i,j} + q_{i,j-1})(u^n_{i,j} - u^n_{i,j-1})) + f^n_{i,j}\Big\}$$

The general scheme for computing $u^{n+1}_{i,j}$ at interior spatial mesh points:

$$B = (1 + \frac{b * \Delta t}{2})^{-1}, \quad C_x = (\frac{\Delta t}{\Delta x})^2, \quad C_y = (\frac{\Delta t}{\Delta y})^2$$
$$u^{n+1}_{i,j} = B * \Big\{2u^n_{i,j} - u^{n-1}_{i,j} * B^{-1} + \Delta t^2 f^n_{i,j}$$
$$+ \frac{c_x}{2}[(q_{i+1,j} + q_{i,j})(u^n_{i+1,j} - u^n_{i,j}) - (q_{i,j} + q_{i-1,j})(u^n_{i,j} - u^n_{i-1,j})]$$
$$\frac{c_y}{2}[(q_{i,j+1} + q_{i,j})(u^n_{i,j+1} - u^n_{i,j}) - (q_{i,j} + q_{i,j-1})(u^n_{i,j} - u^n_{i,j-1})]\Big\}$$

The modified scheme for the first step: (using c_x and c_y from last step)

$$
\begin{aligned}
u^0_{i,j} &= I_{i,j}, \quad \frac{\partial u}{\partial t} = V_{i,j} \rightarrow u^{-1}_{i,j} = u^1_{i,j} - 2\Delta t V_{i,j} \\
u^1_{i,j} &= \frac{1}{2} * \Big\{ 2I_{i,j} - 2\Delta t V_{i,j} - b\Delta t^2 V_{i,j} + \Delta t^2 f^0_{i,j} \\
&\quad + \frac{c_x}{2}[(q_{i+1,j} + q_{i,j})(I_{i+1,j} - I_{i,j}) - (q_{i,j} + q_{i-1,j})(I_{i,j} - I_{i-1,j})] \\
&\quad \frac{c_y}{2}[(q_{i,j+1} + q_{i,j})(I_{i,j+1} - I_{i,j}) - (q_{i,j} + q_{i,j-1})(I_{i,j} - I_{i,j-1})] \Big\}
\end{aligned}
$$

the modified scheme at boundary points (first step and subsequent steps), is almost the same as first step with j = 0 and i = 0.

2.3 Implementation

All implementation above is added to the modified file wave2D_u0_S.py. Have implemented all the corners, values at the boundary and the inner points.

3 Verification

3.1 Constant solution

3.1.1 and 3.1.3

Added the function test_constant() were u(x, y, t) = c, where c is a number, and the choosing f,b,q,I, and V such that u = c.

3.1.2

To show that the constant solutions is also a solution of the discrete equations we need first to put u(x,y,t) = 0 in our original equation to find that f(x,y,t) = 0, and then using the discrete equations to find that $f^n_{i,j} = 0$:

$$
\begin{aligned}
\frac{c - 2c + c}{\Delta t^2} + b\frac{c + c}{2\Delta t} = \frac{1}{2}\Big\{ & \frac{1}{\Delta} x^2((q_{i+1,j} + q_{i,j})(c - c) - (q_{i,j} + q_{i-1,j})(c - c) + \\
& \frac{1}{\Delta} y^2((q_{i,j+1} + q_{i,j})(c - c) - (q_{i,j} + q_{i,j-1})(c - c) + f^n_{i,j} \Big\}
\end{aligned}
$$

and this gives us that $f^n_{i,j} = 0$. Hence constant solution is a solution of the discrete equation

3.1.4

five problems bugs: c to big, V and f $\neq 0$, change the exact solution and change initial condition (change I)

3.3 Exact 1D plug-wave solution in 2D

Implemented a test_plug() in the program and chosen b = 0 and the other variables. The test function gives zero error and nosetests runs without any problem.

3.4 Standing, undamped waves

Given:

$$k_x = \frac{m_x x}{L_x}, \quad k_y = \frac{m_y y}{L_y}$$
$$u_e(x, y, t) = Acos(k_x x)cos(k_y y)cos(\omega t)$$

for arbitrary amplitude A, arbitrary integers m_x and m_y, and $\omega = \sqrt{(k_x^2 + k_y^2)}$. The function for the error and calculated a sequence of r values, r = $\frac{log(E[i-1]/E[i])}{log(h[i-1]/h[i]}$. All this is implementet and the r values are printed if you run test_standing_undamped_waves().

3.6 Manufactured solution

have implemented a small test that uses sympy, but the question was a bit hard to understand so i could not find q and f, but after finding q and f the implementation is thr same as the Standing, undamped waves (with q$\neq$ 0=

4 Investigate a physical problem

looking at what happens to a wave that enters a medium with different wave velocity. The unknown u(x,y,t) is then the elevation of the ocean surface, and the boundary condition $\frac{\partial u}{\partial n} = 0$ means that the waves are perfectly reflected. The wave velocity is given as q = gH(x,y), were q is the acceleration of gravity and H(x,y) is the stillwater depth. Used inspiration from the 1D problem, it almost the same but, we need to add the problem inn y direction to. Most of the information is in the program, but we basically run the program with three different bottom shapes Gaussian, cosine hat and box.