

INF5620 — Project 2

2D Wave Equation

Emilie Fjørner
`e.s.fjorner@fys.uio.no`

Jonas van den Brink
`j.v.d.brink@fys.uio.no`

October 28, 2013

Description

Partial Differential Equation

The PDE we will be solving has the following form,

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

This is a two-dimensional, standard, linear wave equation, with damping. Here $q(x, y) = c^2$ is the wave velocity, which is generally a field. The constant b , is a damping factor, and $f(x, y, t)$ is a source term that will be used to verify our solver.

We solve the equation on the spatial domain $\Omega = [0, L_x] \times [0, L_y]$, with a Neumann boundary condition

$$\frac{\partial u}{\partial n} = 0 \text{ on } \delta\Omega,$$

here $\partial/\partial n$ denotes the directional derivative out of the domain at the boundary.

Our PDE also has the initial conditions

$$u(x, y, 0) = I(x, y), \quad u_t(x, y, 0) = V(x, y).$$

Discretizing the PDE

We discretize both the temporal domain $[0, T]$, and both spatial dimensions, using uniform meshes. This means we define mesh points

$$\begin{aligned} x_i &= i\Delta x, \text{ for } i = 0, \dots, N_x, \\ y_j &= j\Delta y, \text{ for } j = 0, \dots, N_y, \\ t_n &= n\Delta t, \text{ for } n = 0, \dots, N_t. \end{aligned}$$

We now evaluate our PDE in the point (x_i, y_j, t_n) and introduce the shorthand notation

$$u_{i,j}^n \equiv u(x_i, y_j, t_n).$$

We will use central difference approximations for the time derivatives, meaning we have

$$\left[\frac{\partial^2 u}{\partial t^2} \right]^n \approx \frac{u^{n+1} - 2u^n + u^{n-1}}{\Delta t^2} = \left[D_t D_t u \right]^n,$$

and

$$b \frac{\partial u}{\partial t} \approx b \frac{u^{n+1} - u^{n-1}}{2\Delta t} = \left[D_{2t} u \right]^n.$$

Hvorfor har vi med b her? (Burde i såfall vært med i siste leddet og?)

For the spatial derivatives, we first approximate the outer derivative using a central difference. In order to do this we first introduce $\phi \equiv q \partial u / \partial x$, and find

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi_{i+\frac{1}{2}} - \phi_{i-\frac{1}{2}}}{\Delta x} = [D_x \phi]_i.$$

where we approximate $\phi_{i+\frac{1}{2}}$ and $\phi_{i-\frac{1}{2}}$, using a central difference yet again

$$\phi_{i+1/2} = q_{i+\frac{1}{2}} \left[\frac{\partial u}{\partial x} \right]_{i+\frac{1}{2}} \approx q_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} = [q D_x u]_{i+\frac{1}{2}}.$$

$$\phi_{i-1/2} = q_{i-\frac{1}{2}} \left[\frac{\partial \phi}{\partial x} \right]_{i-\frac{1}{2}} \approx q_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x} = [q D_x u]_{i-\frac{1}{2}}.$$

If we have access to a continuous q , evaluating q in $x_{i+\frac{1}{2}}$ is no problem, but we would also like to be able to use a discretized q known only in the mesh points, so we approximate $q_{i+\frac{1}{2}}$ using an arithmetic mean

$$q_{i+\frac{1}{2}} \approx \frac{q_{i+1} + q_i}{2}, \quad q_{i-\frac{1}{2}} \approx \frac{q_i + q_{i-1}}{2}.$$

Inserting this, we have

$$\left[\frac{\partial}{\partial x} \left(q \frac{\partial u}{\partial x} \right) \right]_i \approx \frac{1}{2\Delta x^2} \left[(q_{i+1} + q_i)(u_{i+1} - u_i) + (q_{i-1} + q_i)(u_{i-1} - u_i) \right].$$

And we use the exact same approximation for the other spatial derivative.

Our discrete equation then becomes

$$[D_t D_t u + b D_{2t} u = D_x \bar{q}^x D_x u + D_y \bar{q}^y D_y u + f]_{i,j}^n.$$

Which written out and solved for $u_{i,j}^{n+1}$ gives the following numerical scheme

$$\begin{aligned} u_{i,j}^{n+1} = & \left(\frac{2}{2 + b\Delta t} \right) \left[2u_{i,j}^n - \left(1 - \frac{b\Delta t}{2} \right) u_{i,j}^{n-1} \right. \\ & + \frac{h_x}{2} \left((q_{i+1,j} + q_{i,j})(u_{i+1,j}^n - u_{i,j}^n) + (q_{i-1,j} + q_{i,j})(u_{i-1,j}^n - u_{i,j}^n) \right) \\ & + \frac{h_y}{2} \left((q_{i,j+1} + q_{i,j})(u_{i,j+1}^n - u_{i,j}^n) + (q_{i,j-1} + q_{i,j})(u_{i,j-1}^n - u_{i,j}^n) \right) \\ & \left. + \Delta t^2 f(x_i, y_j, t_n) \right], \end{aligned}$$

where $h_x = \Delta t^2 / \Delta x^2$.

Verifying the solver using constant solution

As a test case, we try implementing a constant solution

$$u_e = C = \text{const.},$$

by inserting this solution into our discrete equation, we see that the source term vanishes

$$f(x, y, t) = 0.$$

We implement the test case as a nose test and verify that it succeeds.

Verifying the solver using a seperable solution

We will verify our solver by attempting to reproduce a seperable exact solution

$$u_e(x, y, t) = X(x)Y(y)T(t),$$

where X, Y are cubic polynomials and T is quadratic. To satisfy our Neumann boundary conditions, we will have to construct X and Y so that they vanish on the boundaries, so

$$u_e(x, y, t) = 0 \text{ on } \delta\Omega, \quad X(0) = X(L_x) = Y(0) = Y(L_y) = 0.$$

For ease of implementation, we won't treat the polynomials as completely general. We let $T(t) = Dt$, this implies that $I(x, y) = 0$, and $V(x, y) = DX(x)Y(y)$. We then have

$$\begin{aligned} X(x) &= Ax^3 - \frac{3}{2}L_xAx^2 \\ Y(y) &= By^3 - \frac{3}{2}L_yBy^2 \\ T(t) &= Dt, \end{aligned}$$

where A, B , and D are all real constants. To fit our source term to the exact solution, we insert it into our discrete equation.

$$[D_tD_tu_e + bD_{2t}u_e = D_x\bar{q}^x D_xu + D_y\bar{q}^y D_yu + f]_{i,j}^n.$$

In our verification we will ignore damping ($b = 0$) and we will let the wave velocity be constant, $q(x, y) = \text{const.}$, so the equation simplifies

$$[X \cdot Y \cdot D_tD_tT = q \cdot Y \cdot T \cdot D_xD_xX + q \cdot X \cdot T \cdot D_yD_yY + f]_{i,j}^n.$$

This is our discrete equation for all internal mesh points, for the points next to the boundaries, the discrete equation will look slightly different, which we will discuss shortly. As all finite differences in the discrete equation are 2. order we know that they exactly reproduce the derivative of a third degree polynomial (this can be shown from simple insertion), so we have

$$\begin{aligned} D_tD_tT(t) &= D_tD_tDt = 0, \\ D_xD_xX(x) &= D_xD_x\left(Ax^3 - \frac{3}{2}L_xAx^2\right) = A(6x - 3L_x), \\ D_yD_yY(y) &= D_yD_y\left(By^3 - \frac{3}{2}L_yBy^2\right) = B(6y - 3L_y), \end{aligned}$$

So when we write out the discrete equation we have

$$0 = qY_j \cdot T_n \cdot (6Ax_i - 3L_xA) + qX_i \cdot T_n \cdot (6By_i - 3L_yB) + f_{i,j}^n.$$

So the fitted source term is

$$f_{i,j}^n = -q\left(A(6x - 3L_x)Y_j + B(6y - 3L_y)X_i\right) \cdot T_n.$$

If we use this source term, the discrete equation is solved exactly, and the solver should reproduce the exact separable solution. This is the case for the internal mesh points, let us now look closer at the points at the boundary of our domain.

Source term at the boundary

Due to our Neumann condition, $\partial u / \partial n = 0$ on $\delta\Omega$, the discrete equation we are solving on the boundary is slightly different than the discrete equation for the internal mesh points. For the internal mesh points we had

$$[D_t D_t u = q D_x D_x u + q D_y D_y u + f]_{i,j}^n.$$

Where the spatial derivatives are the 2. order central differences

$$D_x D_x u = \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2},$$

however, at the boundary $x = 0$, we have $i = 0$, meaning we need to know the value outside the mesh u_{-1} . In our implementation, this point is included as a ghost cell, so there is no problem evaluating this point. This is possible because we know that $u_{-1} = u_1$, due to our reflective Neumann boundary condition. Inserting this equality, effectively means we are using the discrete equation

$$[D_t D_t u = q D_x D_x u + q \frac{2}{\Delta x} D_{2x} u + q D_y D_y u + f]_{i,j}^n.$$

at the boundary $x = 0$. For $x = L_x$ we get the opposite sign on the central difference term. A same reasoning works for the boundaries at $y = 0$ and $y = L_y$.

The first order central difference does *not* exactly reproduce a cubic polynomial, which is easy to show explicitly

$$q \frac{2}{\Delta x} D_{2x} u_e.$$

This means that we must use an extra contribution to the source term at the boundary:

$$g_1 = -[q \frac{2}{\Delta x} D_{2x} u_e]_{i,j}^n = -2q Y_j T_n A \Delta x,$$

the expression g_1 applies at the boundary $x = 0$. Similar, slightly different, corrections have to be added at $x = L_x$ and $y = 0$ and $y = L_y$.