

---

# waveEquation

Milad H. Mobarhan & Sverren-Arne Dragly

October 06, 2013

## 1 Project 2: 2D wave equation

*Summary.* The aim of this project is to develop a solver for the two-dimensional, standard, linear wave equation, with damping, and verify the solver.

---

### 1.1 Mathematical problem

The general wave equation in  $d$  space dimensions, with variable coefficients, can be written in the compact form

$$\varrho \frac{\partial^2 u}{\partial t^2} + b \frac{\partial u}{\partial t} = \nabla \cdot (q \nabla u) + f \text{ for } \mathbf{x} \in \Omega \subset \mathbb{R}^d, t \in (0, T],$$

which in 2D becomes

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

To save some writing and space we may use the index notation, where subscript  $t$ ,  $x$  or  $y$  means differentiation with respect to that coordinate, i.e.

$$\begin{aligned} u_t &= \frac{\partial u}{\partial t}, & u_{tt} &= \frac{\partial^2 u}{\partial t^2} \\ u_x &= \frac{\partial u}{\partial x}, & u_{xx} &= \frac{\partial^2 u}{\partial x^2} \\ u_y &= \frac{\partial u}{\partial y}, & u_{yy} &= \frac{\partial^2 u}{\partial y^2} \\ (qu_x)_x &= \frac{\partial}{\partial x} \left( q(x, y) \frac{\partial u}{\partial x} \right) \\ (qu_y)_y &= \frac{\partial}{\partial y} \left( q(x, y) \frac{\partial u}{\partial y} \right). \end{aligned}$$

The 2D versions of the two model PDEs, with and without variable coefficients, can with now with the aid of the index notation for differentiation be stated as

$$\varrho u_{tt} + bu_t = (qu_x)_x + (qu_y)_y + f$$

Since this PDE contains a second-order derivative in time, we need two initial conditions; (1) the initial shape of the string,  $I$ , and (2) the initial velocity of the string,  $V$ ;

$$\begin{aligned} u(x, y, 0) &= I(x, y), \\ u_t(x, y, 0) &= V(x, y). \end{aligned}$$

In addition, PDEs need boundary conditions, which are of three principal types:

- $u$  is prescribed ( $u = 0$  or a known time variation for an incoming wave)
- $\partial u / \partial n = \mathbf{n} \cdot \nabla u$  prescribed (zero for reflecting boundaries)
- An open boundary condition (also called radiation condition) is specified to let waves travel undisturbed out of the domain.

## 1.2 Discretization

In this section we will derive the discrete set of equations to be implemented in a program. We will for simplicity assume constant spacing between the mesh points. Our mesh points are

$$x_i = i\Delta x, \quad i = 0, \dots, N_x, \quad y_j = j\Delta y, \quad j = 0, \dots, N_y, \quad t_n = n\Delta t, \quad n = 0, \dots, N_t.$$

The solution  $u(x, y, t)$  is sought at the mesh points. We introduce the mesh function  $u_{i,j}^n$ , which approximates the exact solution at the mesh point  $(x_i, y_j, t_n)$  for  $i = 0, \dots, N_x, j = 0, \dots, N_y$  and  $n = 0, \dots, N_t$ .

### Discretizing the variable coefficient

The principal idea is to first discretize the outer derivative. We define

$$\phi^x = q(x, y) \frac{\partial u}{\partial x},$$

and use a centered derivative around  $x = x_i$  for the derivative of  $\phi$ :

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

Then discretize

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

Similarly,

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

These intermediate results are now combined to

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

With operator notation we can write the discretization as

$$\left[ \frac{\partial}{\partial x} \left( q \frac{\partial u}{\partial x} \right) \right]_i^n \approx [D_x q D_x u]_i^n$$

Similarly we have

$$\left[ \frac{\partial}{\partial y} \left( q \frac{\partial u}{\partial y} \right) \right]_i^n \approx [D_y q D_y u]_i^n$$

for

$$\phi^y = q(x, y) \frac{\partial u}{\partial y}.$$

In order to compute  $[D_x q D_x u]_i^n$  and  $[D_y q D_y u]_i^n$  we need to evaluate  $q_{i \pm \frac{1}{2}, j}$  and  $q_{i, j \pm \frac{1}{2}}$ . If  $q$  is a known function, we can easily evaluate  $q_{i \pm \frac{1}{2}, j}$  simply as  $q(x_{i \pm \frac{1}{2}}, y_j)$  with  $x_{i \pm \frac{1}{2}} = x_i \pm \frac{1}{2} \Delta x$ . However, in many cases  $q$  is only known as a discrete function, and evaluating must be done by averaging. The most commonly used averaging technique is the arithmetic mean:

$$q_{i + \frac{1}{2}, j} \approx \frac{1}{2} (q_{i+1, j} + q_{i, j}) = [\bar{q}^x]_{i + \frac{1}{2}, j}, \quad q_{i - \frac{1}{2}, j} \approx \frac{1}{2} (q_{i, j} + q_{i-1, j}) = [\bar{q}^x]_{i - \frac{1}{2}, j},$$

which we are going to use in the following.

## Replacing derivatives by finite differences

The second-order derivatives can be replaced by central differences. The most widely used difference approximation of the second-order derivative is

$$\frac{\partial^2}{\partial t^2} u(x_i, y_j, t_n) \approx \frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}}{\Delta t^2}.$$

It is convenient to introduce the finite difference operator notation

$$[D_t D_t u]_{i,j}^n = \frac{u_{i,j}^{n+1} - 2u_{i,j}^n + u_{i,j}^{n-1}}{\Delta t^2}.$$

The first-order derivative in time in the damping term can be approximated by

$$[D_{2t} u]_{i,j}^n = \frac{u_{i,j}^{n+1} - u_{i,j}^{n-1}}{2\Delta t}.$$

## Algebraic version of the initial conditions

The initial conditions are given as

$$\begin{aligned} u(x, y, t_0) &= I(x, y), \\ u_t(x, y, t_0) &= V(x, y). \end{aligned}$$

The first condition can be computed by

$$u_i^0 = I(x_i), \quad i = 0, \dots, N_x.$$

For the second one we use a centered difference of type

$$\frac{\partial}{\partial t} u(x_i, y_j, t_0) \approx \frac{u_{i,j}^1 - u_{i,j}^{-1}}{2\Delta t} = [D_{2t}u]_{i,j}^0 = V(x_i, y_j).$$

## Algebraic version of the PDE

### Interior spatial mesh points

The PDE with variable coefficients is discretized term by term:

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

When written out and solved for the unknown  $u_{i,j}^{n+1}$  one gets the scheme

$$\begin{aligned} u_{i,j}^{n+1} = & \left\{ 2u_{i,j}^n + u_{i,j}^{n-1} \left[ \frac{b\Delta t}{2\varrho_{i,j}} - 1 \right] + \frac{\Delta t^2}{\varrho_{i,j}} f_{i,j}^n \right. \\ & + \frac{1}{\varrho_{i,j}} \frac{\Delta t^2}{\Delta x^2} \left[ \frac{1}{2}(q_{i,j} + q_{i+1,j})(u_{i+1,j}^n - u_{i,j}^n) - \frac{1}{2}(q_{i-1,j} + q_{i,j})(u_{i,j}^n - u_{i-1,j}^n) \right] \\ & \left. + \frac{1}{\varrho_{i,j}} \frac{\Delta t^2}{\Delta y^2} \left[ \frac{1}{2}(q_{i,j} + q_{i,j+1})(u_{i,j+1}^n - u_{i,j}^n) - \frac{1}{2}(q_{i,j-1} + q_{i,j})(u_{i,j}^n - u_{i,j-1}^n) \right] \right\} \left( \frac{1}{1 + \frac{b\Delta t}{2\varrho_{i,j}}} \right). \end{aligned}$$

### Modified scheme for the first step

A problem with algebraic version of the PDE equation arises when  $n = 0$  since the formula for  $u_{i,j}^1$  involves  $u_{i,j}^{-1}$ , which is an undefined quantity outside the time mesh (and the time domain). However, we can use the initial condition to arrive at a special formula for  $u_{i,j}^1$ . From initial condition we have

$$u_{i,j}^{-1} = u_{i,j}^1 - 2\Delta t V.$$

Inserting this into the algebraic version of the PDE equation for  $n = 0$  gives:

$$\begin{aligned} u_{i,j}^1 = & \left\{ 2u_{i,j}^0 + (u_{i,j}^1 - 2\Delta t V) \left[ \frac{b\Delta t}{2\varrho_{i,j}} - 1 \right] + \frac{\Delta t^2}{\varrho_{i,j}} f_{i,j}^0 \right. \\ & + \frac{1}{\varrho_{i,j}} \frac{\Delta t^2}{\Delta x^2} \left[ \frac{1}{2}(q_{i,j} + q_{i+1,j})(u_{i+1,j}^0 - u_{i,j}^0) - \frac{1}{2}(q_{i-1,j} + q_{i,j})(u_{i,j}^0 - u_{i-1,j}^0) \right] \\ & \left. + \frac{1}{\varrho_{i,j}} \frac{\Delta t^2}{\Delta y^2} \left[ \frac{1}{2}(q_{i,j} + q_{i,j+1})(u_{i,j+1}^0 - u_{i,j}^0) - \frac{1}{2}(q_{i,j-1} + q_{i,j})(u_{i,j}^0 - u_{i,j-1}^0) \right] \right\} \left( \frac{1}{1 + \frac{b\Delta t}{2\varrho_{i,j}}} \right). \end{aligned}$$

Thus, the special formula for  $u_{i,j}^1$  is

$$u_{i,j}^1 = \left\{ 2u_{i,j}^0 - 2\Delta t V \left[ \frac{b\Delta t}{2\rho_{i,j}} - 1 \right] + \frac{\Delta t^2}{\rho_{i,j}} f_{i,j}^0 \right. \\ \left. + \frac{1}{\rho_{i,j}} \frac{\Delta t^2}{\Delta x^2} \left[ \frac{1}{2}(q_{i,j} + q_{i+1,j})(u_{i+1,j}^0 - u_{i,j}^0) - \frac{1}{2}(q_{i-1,j} + q_{i,j})(u_{i,j}^0 - u_{i-1,j}^0) \right] \right. \\ \left. + \frac{1}{\rho_{i,j}} \frac{\Delta t^2}{\Delta y^2} \left[ \frac{1}{2}(q_{i,j} + q_{i,j+1})(u_{i,j+1}^0 - u_{i,j}^0) - \frac{1}{2}(q_{i,j-1} + q_{i,j})(u_{i,j}^0 - u_{i,j-1}^0) \right] \right\} \left( \frac{1}{1 + \frac{b\Delta t}{2\rho_{i,j}}} \right) \left( \frac{1}{2 - \frac{b\Delta t}{2\rho_{i,j}}} \right).$$


---

### 1.3 Boundary condition

In a rectangular spatial domain  $\Omega = [0, L_x] \times [0, L_y]$ , the homogeneous Dirichlet condition, is given by

$$u_{i,j}^1 = u_{i,j}^{n+1} = 0$$

for  $n = 1, 2, \dots, N_t - 1$ , when  $i = 0$ ,  $i = N_x$ ,  $j \in [0, L_y]$  and when  $j = 0$ ,  $j = N_y$ ,  $i \in [0, L_x]$ . Note that  $u_{i,j}^0$  is given by  $I$ , i.e.

$$u_{i,j}^0 = I(x_i, y_j) \quad \text{for } i = 0, \dots, N_x \quad j = 0, \dots, N_y$$

The Neumann boundary condition is given by

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial y} = 0, \quad (1)$$

at  $x = 0, N_x$  and  $y = 0, N_y$ . Since we have used central differences in all the other approximations to derivatives in the scheme, it is tempting to implement this condition by the difference

$$\begin{aligned} [-D_{2x}u = 0]_{0,j}^n, \quad [D_{2x}u = 0]_{L_x,j}^n &\Rightarrow u_{-1,j}^n = u_{1,j}^n, \quad u_{N_x+1,j}^n = u_{N_x-1,j}^n \\ [-D_{2y}u = 0]_{i,0}^n, \quad [D_{2y}u = 0]_{i,L_y}^n &\Rightarrow u_{i,-1}^n = u_{i,1}^n, \quad u_{i,N_y+1}^n = u_{i,N_y-1}^n. \end{aligned}$$

The problem is that  $u_{-1,j}^n$ ,  $u_{i,-1}^n$ ,  $u_{N_x+1,j}^n$  and  $u_{i,N_y+1}^n$  are not  $u$  values that are being computed since these points are outside the mesh. However we can use the relations given above and the algebraic version of the PDE equation to arrive at a modified scheme for the boundary points. The implementation of the special formulas for the boundary points can benefit from using the general formula for the interior points also at the boundaries, but replacing  $u_{i-1,j}^n$  by  $u_{i+1,j}^n$  when computing  $u_{i,j}^{n+1}$  for  $i = 0$ , and  $u_{i+1,j}^n$  by  $u_{i-1,j}^n$  when computing  $u_{i,j}^{n+1}$  for  $i = N_x$ . A similar modification must be done for  $j = 0, N_y$ .

#### Implementation of Neumann conditions via ghost cell

Instead of modifying the scheme at the boundary, we can introduce extra points outside the domain such that the fictitious values  $u_{-1,j}^n$ ,  $u_{i,-1}^n$ ,  $u_{N_x+1,j}^n$  and  $u_{i,N_y+1}^n$  are defined in the mesh. Thus we add the points  $i = -1$ ,  $i = N_x + 1$ ,  $j = -1$  and  $j = N_y + 1$ , which are known as ghost points, and values at these points, are known as ghost points. The important idea is to ensure that we always have

$$\begin{aligned} u_{-1,j}^n &= u_{1,j}^n, & u_{N_x+1,j}^n &= u_{N_x-1,j}^n \\ u_{i,-1}^n &= u_{i,1}^n, & u_{i,N_y+1}^n &= u_{i,N_y-1}^n \end{aligned}$$

because then the application of the standard scheme at a boundary point will be correct and guarantee that the solution is compatible with the boundary condition  $u_x = 0$  and  $u_y = 0$ .

---

## 1.4 Truncation error

Truncation error analysis provides a widely applicable framework for analyzing the accuracy of finite difference schemes and is a measure for the extent the exact solution  $u_e$  fits the discrete equations. We define

$$R = \mathcal{L}_\Delta(u_e),$$

where the residual  $R$  is known as the truncation error of the finite difference scheme  $\mathcal{L}_\Delta(u) = 0$ .

The algebraic version of the wave equation with variable coefficients and damping can be written as:

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

Inserting the exact solution  $u_e(x, y, t)$  in this equation makes this function fulfill the equation if we add the term  $R$ :

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

Our purpose is to calculate the truncation error  $R$  and this can be done by Taylor expansion of the terms in the PDE:

$$[D_t D_t u_e]_{i,j}^n = u_{e,tt}(x_i, y_j, t_n) + \frac{1}{12} u_{e,tttt}(x_i, y_j, t_n) \Delta t^2 + \mathcal{O}(\Delta t^4)$$

$$[D_{2t} u_e]_{i,j}^n = u_{e,t}(x_i, y_j, t_n) + \frac{1}{6} u_{e,ttt}(x_i, y_j, t_n) \Delta t^2 + \mathcal{O}(\Delta t^4)$$

$$[D_x \bar{q}^x D_x u_e]_{i,j}^n = (q u_{e,x})_x + \mathcal{O}(\Delta x^2).$$

$$[D_y \bar{q}^y D_y u_e]_{i,j}^n = (q u_{e,y})_y + \mathcal{O}(\Delta y^2).$$

Inserting all these terms into the discrete equation gives us

$$\begin{aligned} & \varrho \left( u_{e,tt}(x_i, y_j, t_n) + \frac{1}{12} u_{e,tttt}(x_i, y_j, t_n) \Delta t^2 + \mathcal{O}(\Delta t^4) \right) + b \left( u_{e,t}(x_i, y_j, t_n) + \frac{1}{6} u_{e,ttt}(x_i, y_j, t_n) \Delta t^2 + \mathcal{O}(\Delta t^4) \right) \\ &= (q(x_i, y_j) u_{e,x}(x_i, y_j, t_n))_x + \mathcal{O}(\Delta x^2) + (q(x_i, y_j) u_{e,y}(x_i, y_j, t_n))_y + \mathcal{O}(\Delta y^2) + f(x_i, y_j, t_n) + R_{i,j}^n \end{aligned}$$

Because  $u_e$  fulfills the partial differential equation (PDE):

$$\varrho u_{tt} + b u_t = (q u_x)_x + (q u_z)_z + (q u_z)_z + f$$

some of the terms cancel out, and we are left with

$$R_{i,j}^n = \mathcal{O}(\Delta x^2) + \mathcal{O}(\Delta y^2) + \varrho \left( \frac{1}{12} u_{e,tttt}(x_i, y_j, t_n) \Delta t^2 \right) + b \left( \frac{1}{6} u_{e,ttt}(x_i, y_j, t_n) \Delta t^2 \right)$$

The scheme is of second-order in time and space. The key ingredients for second order are the centered differences and the arithmetic mean for  $q$ : all those building blocks feature second-order accuracy.