2D Wave Equation – Numerical Solution

Goal: Having derived the 1D wave equation for a vibrating string and studied its solutions, we now extend our results to 2D and discuss efficient techniques to approximate its solution so as to simulate wave phenomena and create photorealistic animations.

I. The 2D Wave Equation with Damping

$$\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \mu \frac{\partial u}{\partial t}, \quad + \quad \text{I.C.:} \ \ u(x,y,0) = f(x,y), u_t(x,y,0) = g(x,y) \quad + \quad \text{B.C...}$$

where:

$u(x,y,t)$ – vertical displacement of the (fluid) surface at 2D point (x,y) at time t

c – wave speed

$\mu \geq 0$ – damping coefficient, related to fluid viscosity

II. Discretizion – Approximating Derivatives with Finite Differences

We fix space and time scales, $\Delta x = \Delta y$, and Δt, and let $u^n_{j,k}$ denote the approximate (numerical) value of $u(x_j, y_k, t^n)$

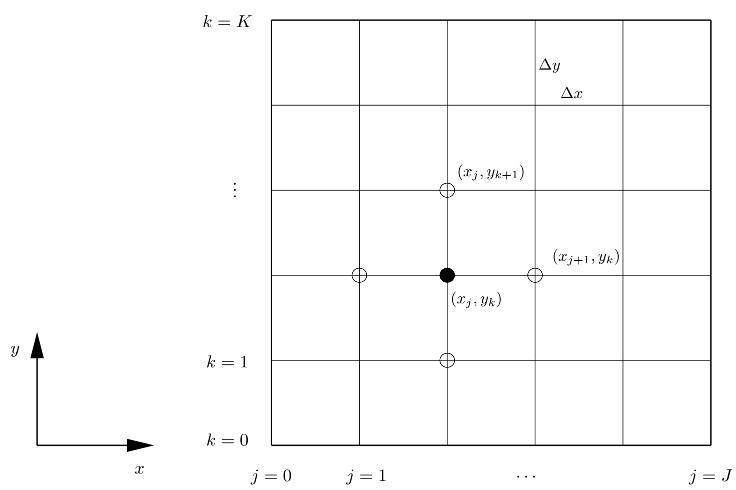

Figure 1: Solution domain

Spatial Derivatives – First Order

Forward difference:

$$\frac{\partial u(x_j, y_k, t^n)}{\partial x} = \lim_{\Delta x \to 0} \frac{u(x_j + \Delta x, y_k, t^n) - u(x_j, y_k, t^n)}{\Delta x} \approx D_+^x u_{j,k}^n = \frac{u_{j+1,k}^n - u_{j,k}^n}{\Delta x}$$

Backward difference:

$$\frac{\partial u_{j,k}^n}{\partial x} \approx D_-^x u_{u,jk}^n = \frac{u_{j,k}^n - u_{j-1,k}^n}{\Delta x}$$

Centered difference:

$$\frac{\partial u_{j,k}^n}{\partial x} \approx D_0^x u_{j,k}^n = \frac{u_{j+1,k}^n - u_{j-1,k}^n}{2\,\Delta x}$$

Spatial Derivatives – Second Order

$$\frac{\partial^2 u_{j,k}^n}{\partial x^2} \approx D_-^x D_+^x u_{j,k}^n = D_-^x \left(\frac{u_{j+1,k}^n - u_{j,k}^n}{\Delta x} \right) = \frac{u_{j+1,k}^n - 2u_{j,k}^n + u_{j-1,k}^n}{\Delta x^2}$$

$$\frac{\partial^2 u_{j,k}^n}{\partial y^2} \approx D_-^y D_+^y u_{j,k}^n = D_-^y \left(\frac{u_{j,k+1}^n - u_{j,k}^n}{\Delta y} \right) = \frac{u_{j,k+1}^n - 2u_{j,k}^n + u_{j,k-1}^n}{\Delta y^2}$$

Remark: The centered second order approximation

$$\frac{\partial^2 u_{j,k}^n}{\partial y^2} \approx D_0^y D_0^y u_{j,k}^n = \frac{u_{j,k+2}^n - 2u_{j,k}^n + u_{j,k-2}^n}{4\Delta y^2}$$

is not a good approximation, **why?** (HW)

Time Derivatives

$$\frac{\partial u_{j,k}^n}{\partial t} \approx \frac{u_{j,k}^{n+1} - u_{j,k}^n}{\Delta t}, \qquad \frac{\partial^2 u_{j,k}^n}{\partial t^2} \approx \frac{u_{j,k}^{n+1} - 2u_{j,k}^n + u_{j,k}^{n-1}}{\Delta t^2}$$

III. Numerical Scheme

Displacement update:

We now *plug* all these approximations into our original equation, and after doing the algebra (HW), and obtain a formula for updating the displacement of the (fluid) surface at the point (x_j, y_k) from time $t = t^n$ to $t = t^n + \Delta t = t^{n+1}$:

$$u_{j,k}^{n+1} = \frac{4 - 8c^2\Delta t^2/\Delta x^2}{\mu\Delta t + 2}\, u_{j,k}^n + \frac{\mu\Delta t - 2}{\mu\Delta t + 2}\, u_{j,k}^{n-1} + \frac{2c^2\Delta t^2/\Delta x^2}{\mu\Delta t + 2} \left[u_{j+1,k}^n + u_{j,k+1}^n + u_{j-1,k}^n + u_{j,k-1}^n \right]$$

Stability:

If the wave speed, c, is too fast or the time scale, Δt, successive iterations may cause the displacement to grow and diverge

If the initial displacement at point (x_j, y_k) is $u^0_{j,k} = h$, we need to make sure that it doesn't get larger at the next time step (damping), so we impose the condition

$$|u^1_{j,k}| \leq |u^0_{j,k}| = |h|$$

or equivalently

$$\left| \frac{2 - 8c^2\Delta t^2/\Delta x^2 + \mu\Delta t}{\mu\Delta t + 2} \right| |h| \leq |h|$$

which leads to the *stability condition* (HW)

$$0 < \Delta t < \frac{\mu + \sqrt{\mu^2 + 32c^2/\Delta x^2}}{8c^2/\Delta x^2}$$

IV. Initial and Boundary Conditions (Next lab)

Given the solution at $t = 0$ (I.C.), we are now ready to update it with our numerical scheme... or almost: to go from $t = t^n$ to $t = t^{n+1}$, we also need the solution at $t = t^{n-1}$. That is, to calculate the solution at $t = 2\Delta t$, from $t = 0$, we first need to construct an approximation of the solution at $t = \Delta t$. What approximation we can build, will depend on the initial conditions $u(x, y, 0)$ and $u_t(x, y, 0)$...

Similarly, to update the boundary values of our solution domain, $j = 0, J$ and $k = 0, K$, we would need information at points (x_j, y_k), for $j = -1, j = J + 1$, and $k = -1, k = K + 1$, which we don't have. For these, we need to take into account the boundary conditions...