Physics 750: Exercise 10

Tuesday, October 17, 2017

The equation of motion for a vibrating string has the form

$$\rho\frac{\partial^2 u}{\partial t^2} + R\frac{\partial u}{\partial t} = T\frac{\partial^2 u}{\partial x^2} - \kappa\frac{\partial^4 u}{\partial x^4} + \cdots, \tag{1}$$

where $u(x,t)$ is the displacement field, ρ is the mass density per unit length, R is the loss coefficient, T is the line tension, and κ is the line stiffness. The ellipses denote additional terms with higher order derivatives and nonlinear contributions. In the limit of small oscillations and small gradients, and at times $t \ll 1/R$, the string's behaviour is governed by the wave equation

$$\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2},$$

which is linear and lossless. As we discussed in class, any right- or left-travelling wave of the form $u(x,t) = F(x \pm ct)$, propagating at constant velocity $c = \sqrt{T/\rho}$, is a solution.

The general solution, due to d'Alembert, is as follows. Given the initial conditions

$$u(x,0) = F(x),$$
$$\frac{\partial u}{\partial t}(x,0) = P(x),$$

the displacement field at all future times is

$$u(x,t) = \frac{1}{2}\left[F(x-ct) + F(x+ct) + \frac{1}{c}\int_{x-ct}^{x+ct} P(\xi)\,d\xi\right].$$

Pure right- and left-travelling solutions are recovered with the choice $P(x) = \pm cF'(x)$, which you can derive by backpropagating $u(x,0) = F(x)$ to $u(x,-dt) = F(x \mp c\,dt)$. A curious feature of the general solution is that a wave pulse released from rest ($P = 0$) splits into two oppositely-directed pulses.

In this exercise, we will attempt to solve the wave equation using finite-difference methods. In the usual way, we create a mesh in both space and time ($x_i = i\Delta x$ and $t_n = n\Delta t$) and track the displacement values at points on a two-dimensional grid, $u_i^{(n)} = u(x_i, t_n)$. Approximating both second-order partial derivatives (in x and t) using centred finite differences,

$$\frac{u_i^{(n+1)} + u_i^{(n-1)} - 2u_i^{(n)}}{(\Delta t)^2} \approx c^2\frac{u_{i+1}^{(n)} + u_{i-1}^{(n)} - 2u_i^{(n)}}{(\Delta x)^2},$$

we arrive at the recurrence relation

$$u_i^{(n+1)} = 2(1-s^2)u_i^{(n)} - u_i^{(n-1)} + s^2\left(u_{i+1}^{(n)} + u_{i-1}^{(n)}\right). \tag{2}$$

Each subsequent time step is computed in terms of displacements at two previous time steps; there is one independent scale factor $s = c\Delta t/\Delta x$. This scheme can be represented by the diagram in Fig. 1(a).

Another approach involves breaking the second-order wave equation into two coupled first-order equations. For instance, if we define new variables $v = \partial u/\partial t$, $\theta = -c\partial u/\partial x$, then we get

$$\frac{\partial^2 u}{\partial t^2} - c\frac{\partial^2 u}{\partial x^2} = \frac{\partial v}{\partial t} + c\frac{\partial \theta}{\partial x} = 0$$

and

$$\frac{\partial^2 u}{\partial t\partial x} - \frac{\partial^2 u}{\partial t\partial x} = \frac{\partial \theta}{\partial t} + c\frac{\partial v}{\partial x} = 0.$$

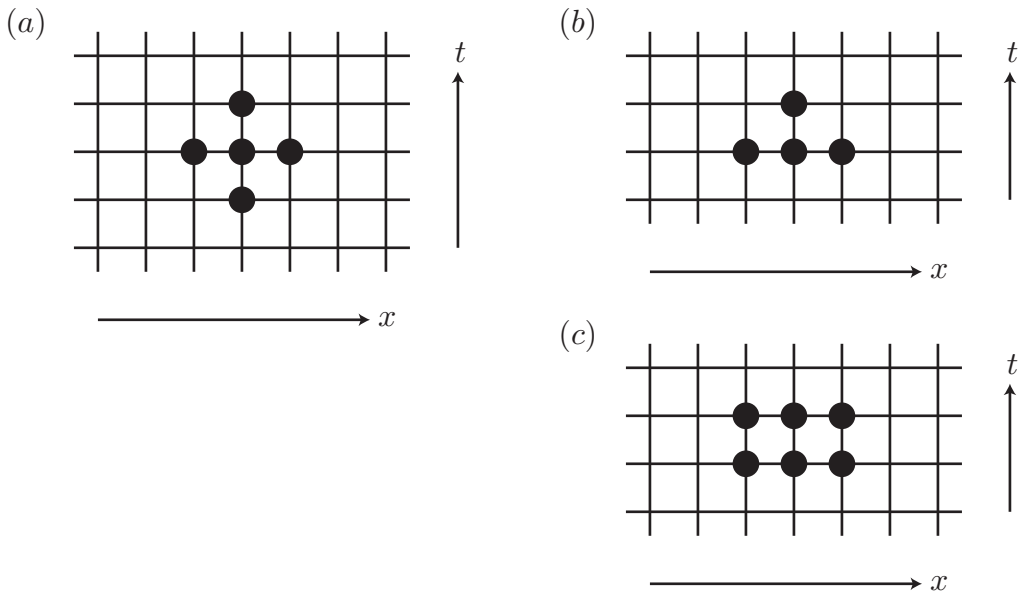

Figure 1: Discretization "stencils" for the (a) 5-point explicit, (b) 4-point explicit, and (c) Crank-Nicholson implicit schemes.

These two equations are symmetric under interchange of θ and v and are known as advection equations. The situation here is similar to what we encountered with first order ODEs. If we integrate the time step using the box rule, we get

$$\frac{v_i^{(n+1)} - v_i^{(n)}}{\Delta t} + c\frac{\theta_{i+1}^{(n)} - \theta_{i-1}^{(n)}}{2\Delta x} = 0 \quad (\text{and } v \leftrightarrow \theta), \tag{3}$$

whereas if we integrate using the trapezoid rule, we get

$$\frac{v_i^{(n+1)} - v_i^{(n)}}{\Delta t} + \frac{c}{2}\left(\frac{\theta_{i+1}^{(n)} - \theta_{i-1}^{(n)}}{2\Delta x} + \frac{\theta_{i+1}^{(n+1)} - \theta_{i-1}^{(n+1)}}{2\Delta x}\right) = 0 \quad (\text{and } v \leftrightarrow \theta). \tag{4}$$

The latter is the more accurate ($O(\Delta t^2, \Delta x^2)$ rather than $O(\Delta t, \Delta x^2)$), but it leads to the same problem we had in our derivation of the Picard rule: we need to know the value at points in the future.

Equation (3) leads to an *explicit 4-point* update rule

$$v_i^{(n+1)} = v_i^{(n)} - \frac{s}{2}\left(\theta_{i+1}^{(n)} - \theta_{i-1}^{(n)}\right),$$

depicted in Fig. 1(b). The value of $v_i^{(n+1)}$ depends only on other quantities evaluated at the previous times slice. On the other hand, Eq. (4) leads to an *implicit 6-point* update rule (referred to as Crank-Nicholson)

$$v_i^{(n+1)} = v_i^{(n)} - \frac{s}{4}\left(\theta_{i+1}^{(n)} - \theta_{i-1}^{(n)} + \theta_{i+1}^{(n+1)} - \theta_{i-1}^{(n+1)}\right),$$

in which the both the left- and right-hand sides have terms at time t_{n+1}; see Fig. 1(c). This requires that $v_i^{(n+1)}$ and $\theta_i^{(n+1)}$ be determined self-consistently (by solving the set of linear equations via matrix inversion or relaxation). The additional computational expense involved in implicit methods is usually worth the gain in numerical stability.

Unlike in the case of ODEs, which can always be controlled with an appropriate choice of method and a small enough time step, finite difference methods for PDEs are prone to instability. A good way to understand this is to use Von Neumann stability analysis. First, we look at a single mode: $u(x,t) = \hat{u}(t)e^{ikx}$. Substituting this into the wave equation gives $\hat{u}''(t) = -c^2k^2\hat{u}(t)$. This has a purely oscillatory solution $\hat{u}(t) = Ae^{ickt}$ obeying $|\hat{u}(t)| = |A| = \text{const}$. On the other hand, the trial solution $u_i^{(n+1)} = \hat{u}^{(n)}e^{ikx_i}$ plugged into Eq. (2) gives

$$\hat{u}^{(n+1)}e^{ikx_i} = 2(1-s^2)u^{(n)}e^{ikx_i} - u^{(n-1)}e^{ikx_i} + s^2\left(\hat{u}^{(n)}e^{ikx_{i+1}} + \hat{u}^{(n)}e^{ikx_{i-1}}\right).$$

This can be reorganized as

$$\begin{aligned}\hat{u}^{(n+1)} &= 2\left[1 - s^2(1-\cos k\Delta x)\right]u^{(n)} - \hat{u}^{(n-1)} \\ &= 2\left[1 - 2s^2\sin^2(k\Delta x/2)\right]u^{(n)} - \hat{u}^{(n-1)}.\end{aligned} \tag{5}$$

Thus, if we assume that the mode grows by a factor z at each time step, i.e., $\hat{u}^{(n)} = z^n\hat{u}^{(0)}$, then Eq. (5) is equivalent to the polynomial

$$z^2 - 2\alpha z + 1 = 0 \quad \text{with} \quad \alpha = s^2\sin^2(k\Delta x/2).$$

The modulus squared of the two roots $z = \alpha \pm \sqrt{\alpha^2 - 1}$ is just

$$|z|^2 = \alpha^2 \pm (\alpha^2 - 1) = \begin{cases} 2\alpha^2 - 1 \\ 1 \end{cases}$$

Requiring that the mode remains bounded ($|z|^2 \le 1$) implies that

$$1 \le 1 - 2s^2\sin^2\frac{k\Delta x}{2} \le 1.$$

The most unstable contribution (occurring at $k\Delta x/2 = \pi/2$) is from the mode with wavelength $2\Delta x$ (the smallest possible wavelength on a grid with spacing Δx), so at worst we need $0 < s \le 1$. We conclude that the method described by Eq. (2) is unstable if $s = c\Delta t/\Delta x > 1$.

1. Use the `curl` command to download from the class website everything you'll need for this exercise.

```
$ WEBPATH=http://www.phy.olemiss.edu/~kbeach/
$ curl $WEBPATH/courses/fall2017/phys750/src/exercise10.tgz -O
$ tar xzf exercise10.tgz
$ cd exercise10
```

2. `cd` into the `exercise10` directory. The `make` command will generate an executable `string` that simulates the wave equation on a line segment $x \in [0, L]$, subject to the boundary conditions $u(0,t) = u(0,L) = 0$. Which integration methods are used is controlled by the boolean variables `use_wave`, `use_advection`, and `implicit`. The solutions computed using the discretized wave and advection equations will be displayed in white and blue, respectively.

3. Run the program with different values of the time step `dt`. Verify that the 5-point scheme is unstable when $s > 1$. Using trial and error, find the values of s at which the 4-point and Crank-Nicholson schemes become unstable. You might want to set `slow_motion = true` so you can watch how the high-frequency noise takes over.

4. By default, the program runs with `initialize(pluck,RIGHT)`, which produces a right-moving Gaussian wave packet. Observe the behaviour when `RIGHT` is replaced by `LEFT` and `STATIONARY`.

5. Set up the string with `initialize(square,RIGHT)`. Try to understand why the square pulse isn't faithfully reproduced.

6. Use the function call `initialize(lowest_harmonic,STATIONARY)` to excite just the lowest mode and `initialize(harmonic,STATIONARY)` to create other linear superpositions of modes. Convince yourself that the behaviour is described by

$$u(x,t) = \sum_{n=1}^{\infty} a_n \sin k_n x \cos \omega_n t,$$

where $k_n = n\pi x/L$ and $\omega_n = ck_n$. Is there a largest mode for the discretized string?

*7. Modify the program so that every few time steps it recomputes the spatial Fouier transform

$$\tilde{u}_n(t) = \frac{2}{L}\int_0^L dx\, u(x,t) \sin k_n x.$$

(You can recycle the integration routines from Exercise 5.) Output the first few Fourier components in columns to a file. Make a plot showing that $\tilde{u}_n(t) = a_n \cos \omega_n t$.

*8. Compute the Fourier components of v and θ. Verify that the total energy,

$$E = \frac{1}{2}\int dx\, (\dot{u}^2 + c^2(\nabla u)^2) = \frac{1}{2}\int dx\, (v^2 + c^2\theta^2) = \frac{1}{2}\sum_{n=1}^{\infty} \omega_n (|\tilde{v}_n|^2 + c^2|\tilde{\theta}_n|^2),$$

is a constant of the motion. You should devise some reasonable scheme for cutting off the sum on the right.

**9. The PDE for a realistic string [Eq. (1)] can be written as

$$\frac{\partial v}{\partial t} + \frac{R}{\rho}v + c\frac{\partial \theta}{\partial x} - \frac{\kappa}{\rho c}\frac{\partial^3 \theta}{\partial x^3} = 0, \quad \frac{\partial \theta}{\partial t} + c\frac{\partial v}{\partial x} = 0.$$

Choose a small value of $r = R/\rho$ and set $\kappa = 0$. Modify the Crank-Nicholson finite-difference code (in the function `update`) to reflect the addition of the dissipation term. Plot the energy as a function of time and extract the decay exponent.