Hyperbolic PDEs—the Wave Equation

The wave equation usually has 'boundary value' conditions in x and 'initial value' conditions in t. The extra twist is that, as this is a second-order equation in t, we are usually given ϕ *and* $\partial\phi/\partial t$ at [eg] $t = 0$. If we, as usual, replace the original equation

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

by a difference equation for the approximation $u_{i,j}$ at the i-th x-strip and j-th t-strip, we have

$$\frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{k^2} \approx c^2\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{h^2},$$

where h and k are the usual strip widths, and so

$$u_{i,j+1} \approx 2u_{i,j} - u_{i,j-1} + (kc/h)^2(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}).$$

Assuming that we start at $t = 0$, $j = 0$, this equation enables us to advance to the next time step *provided that* $j \geqslant 1$.

Unfortunately, we can't immediately take the very first step, since that describes $u_{i,1}$ in terms of the values of u at time 0 and [sadly!] $u_{i,-1}$ at time $-k$. Luckily, it isn't too difficult to get a decent approximation to $u_{i,1}$, since we know $\partial\phi/\partial t$ at $t = 0$. By the Maclaurin expansion,

$$u_{i,1} \approx u_{i,0} + k\dot{u}_{i,0} + \tfrac{1}{2}k^2\ddot{u}_{1,0},$$

and $\dot{u}$ is known from the initial conditions, and $\ddot{u} = c^2u''$ from the wave equation, where u'' is known if we have an explicit differentiable function for the initial u [and can otherwise be replaced by the usual approximation from the initial $u_{i,0}$]. We could take the expansion further, but this is usually overkill, given the accuracy of the rest of the process.

Example

Let us take a very simple case: $c = 1$, $\phi(x,0) = \sin\pi x$, $\dot\phi(x,0) = 0$, $\phi(0,t) = \phi(1,t) = 0$. [Of course, in *really* interesting problems, c is variable.] Then the exact solution is

$$\phi(x,t) = \sin\pi x\cos\pi t.$$

Numerically, we therefore take

$$u_{i,0} = \sin h\pi i;\ u_{i,1} = u_{i,0} - \tfrac{1}{2}k^2\pi^2\sin h\pi i.$$

Then we can use the above scheme to find $u_{i,2}$ and so on. For many reasonable values of h and k, this works surprisingly well: the table shows the maximum error [for any $u_{i,j}$] if we integrate out to $t = 1$.

h	k	maximum error
0.1	0.1	0.00131
0.1	0.01	0.00730
0.1	0.001	0.00747
0.01	0.01	0.00000129
But:		
0.01	0.1	8.48×10^{9}
0.01	0.02	1.05×10^{41}

As with the heat/diffusion equation, there is a partial instability if we take k too large. Essentially, this happens if [in general] $kc > h$, that is, if the distance travelled by ‘sound’ in one time-step is greater than the space-step. The numerical analysis is bounded by the speed of ‘sound’! Note that once stability is achieved, nothing seems to be gained by taking smaller time steps. [Detailed error analysis beyond the present scope.] As usual, it is important that h and k be small enough that any important features in the initial problem feature adequately in the discretisation.

Note that we do not need to invert any matrices or solve any complicated linear equations for this solution. We need to be able to store the two most recent time steps in order to generate the next; but solving two- or even three-dimensional problems is not that hard, as [eg] $100 \times 100 \times 100$ space steps will need the storage of [only!] two million u values at a time.

So hyperbolic PDEs are usually quite easy, as long as the space- and time-steps are carefully chosen; and as long as the boundary conditions are reasonable [such as: vibrating strings, membranes, water waves, *etc.*, provided that the boundary is not 'moving']. Unfortunately, many problems in real life have the boundaries either vibrating [so that, in the current example, $\phi(0,t)$ and/or $\phi(1,t)$ depends on t] or moving [so that ϕ is specified at changing values of x]. If the vibration/motion is [significantly] slower than the 'speed of sound', then there will not usually be any problems. But many real applications involve either forced motion [eg, mechanical vibrations driving musical instruments] or rapid motion [speedboats, jet planes] producing either resonance or shockwaves. These are numerically a serious mess; avoid! You really have to get analytic approximations to what is going on, and hope that the numerical work can be confined to other parts of the problem.

Finite-Element methods for PDEs

[Introduced here so that you know the term!]

One of the big problems with real-life PDEs is the nature of the boundary. Real machines, buildings, capacitors, *etc.* are not just rectangles, bars, cubes, but include complicated shapes, internal corners, holes and so on. Most of the simple PDEs have somewhat analytic solutions, like the $f(x+ct)+g(x-ct)$ solutions of the wave equation, meaning that the solution can often be expressed formally as [for example] a Fourier series, or similar.

The idea in finite-element methods is to divide the problem up into several/many components [similar to a triangulation of a complex surface], each of which can be approximated by a simple shape [such as a triangle]. Then we assume that each component has its own 'solution', given, for example, by the first few terms of a Fourier series [in two or three dimensions] or a Taylor series. For elements 'on the edge', there will be formal conditions on those terms given by the actual boundary conditions. For internal boundaries, there will be 'match-up' conditions, by which we require that the solution on one component turns smoothly into the solution on the next; there are devices such as 'splines' to help with this.

The next step is usually to try to minimise the total error. The PDE will not usually be exactly satisfied by the truncated series, so at each point there will be a discrepancy. Integrate the square [or modulus] of that over the area/volume concerned, and minimise the result as a function of the coefficients in the Fourier/Taylor series. [Note that the integration usually involves quite straightforward trig/polynomial functions, so we do not need NA to do it, but you will usually need a computer to do all the arithmetic.]

The process corresponds roughly to trying to analyse sound waves into fundamental frequencies and the first few harmonics. The expectation is that as we do more work and include more and more harmonics, the solution converges rapidly.

Very similar methods can be used for ODEs. Look up the Rayleigh–Ritz and Galerkin methods.