PHYS 410/555 Computational Physics

The Method of Lines for the Wave Equation

One approach to the numerical solution of time-dependent *partial differential equations* (PDEs) is to use a discretization technique, such as finite-differencing, but only apply it explicitly to the *spatial* part(s) of the PDE operator(s) under consideration. Following the spatial discretization, one is left with a set of coupled *ordinary differential equations* in t, which can then often be solved by a "standard" ODE integrator such as `LSODA`.

As an example of this technique, consider the *wave equation* in one space dimension (often called the "1D wave equation"):

$$\frac{\partial^2}{\partial t^2}u(x,t) = c^2\frac{\partial^2}{\partial x^2}u(x,t) \tag{1}$$

Introducing the notation that a subscript denotes partial differentiation, and suppressing the explicit x and t dependence, (1) can also be written as

$$u_{tt} = c^2 u_{xx} \tag{2}$$

As you probably know, the wave equation describes propagation of disturbances, or waves, at a speed c: waves can either travel to the right (velocity $+c$), or to the left (velocity $-c$). Without loss of generality, we can always choose units such that $c = 1$, and, for convenience, we will do so. Our wave equation then becomes:

$$u_{tt} = u_{xx} \tag{3}$$

As with any differential equation, boundary conditions play a crucial role in fixing a solution of (3). Here, we will solve the wave equation on the domain

$$0 \le x \le 1 \qquad t \ge 0 \tag{4}$$

and will thus have to provide boundary conditions at $x = 0$ and $x = 1$, as well as initial conditions at $t = 0$.

For concreteness, we will prescribe *Dirichlet boundary conditions*:

$$u(0,t) = u(1,t) = 0 \tag{5}$$

as well as the following *initial conditions*:

$$u(x,0) = u_0(x) = \exp\left(-\left(\frac{x-x_0}{\Delta}\right)^2\right) \tag{6}$$

$$u_t(x,0) = 0 \tag{7}$$

where x_0 ($0 < x_0 < 1$) and Δ are specified constants.

If we think in terms of small-amplitude waves propagating on a string, then the Dirichlet conditions correspond to keeping the ends of the string fixed. The interpretation of the initial conditions is as follows: In solving (3) we have the freedom to specify the amplitude of the disturbance for all values of x, as well as the time-rate of change of that amplitude, again for all values of x.

We thus have two functions worth of freedom in specifying our initial conditions. We set the initial amplitude to some functional form given by $u_0(x)$; here we use a "gaussian pulse" that is centred at x_0, and that has an overall effective width of a few $\times\ \Delta$. We also set the initial time rate of change of the amplitude to be 0 for all x.

Such data is known as *time symmetric*, since it defines an instant in the evolution of the wave equation where there is a $t \to -t$ symmetry. In other words, with time symmetric initial data, if we integrate *backward* in time, we will see exactly the same solution as a function of $-t$ as we see integrating forward in time.

Since the wave equation describes propagating disturbances, and given that the initial conditions *are* time symmetric, a little reflection might convince you that the initial conditions (6) and (7) must represent a superposition of equal amplitude right-moving and left-moving pulses. Thus, we should expect the solution of (3), subject to (5), (6) and (7) to describe the propagation of two equal-amplitude pulses that are initially coincident, but that subsequently move apart reflect off $x = 0$ and $x = 1$ respectively, move together, through each other, then apart, etc. etc. Indeed, this is precisely the behaviour we will observe in our subsequent numerical solution.

As mentioned above, the *method of lines*, involves an explicit discretization *only of the spatial part of the PDE operator*. Here we will use the familiar $O(h^2)$ finite-difference approach to the treatment of $u_{xx} \equiv \partial^2 u/\partial x^2$.

However, before proceeding to the spatial discretization, we first note that (3) is a second-order-in-time equation. In order that our approach eventually produce a set of *first order* ODEs in t, we introduce an auxiliary variable, $v(x, t)$,

$$v(x,t) \equiv u_t(x,t) \equiv \frac{\partial u}{\partial t}(x,t) \tag{8}$$

and then rewrite (3) as the *system:*

$$u_t = v \tag{9}$$

$$v_t = u_{xx} \tag{10}$$

The boundary conditions become

$$u(0,t) = u(1,t) = v(0,t) = v(1,t) = 0 \tag{11}$$

while the initial conditions are now

$$u(x,0) = u_0(x) = \exp\left(-\left(\frac{x - x_0}{\Delta}\right)^2\right) \tag{12}$$

$$v(x,0) = 0 \tag{13}$$

We can now proceed with the spatial discretization. To that end, we replace the continuum spatial domain $0 \le x \le 1$ by a uniform finite difference mesh, x_j:

$$x_j \equiv (j-1)h \qquad j = 1, 2, \cdots N \qquad h \equiv (N-1)^{-1} \tag{14}$$

and introduce the discrete unknowns, u_j and v_j:

$$u_j \equiv u_j(t) \equiv u(x_j, t) \tag{15}$$

$$v_j \equiv v_j(t) \equiv v(x_j, t) \tag{16}$$

Using the usual centred, $O(h^2)$ approximation for the second spatial derivative,

$$u_{xx}(x_j) = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} + O(h^2) \tag{17}$$

eqs. (9) and (10) become a set of $2(N-2)$ coupled ODEs for the $2(N-2)$ unknowns $u_j(t)$ and $v_j(t)$, $j = 2, \cdots N-1$:

$$\frac{du_j}{dt} = v_j \qquad j = 2, \cdots N-1 \tag{18}$$

$$\frac{dv_j}{dt} = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} \qquad j = 2, \cdots N-1 \tag{19}$$

We can implement the Dirichlet boundary conditions as follows: if the boundary conditions are satisfied at the initial time, $t = 0$, then they will be satisfied at all future times provided that the time derivatives of u and v vanish at the boundaries. Using this observation, we can now write down a complete set of $2N$ coupled ODEs in the $2N$ unknowns $u_j(t)$ and $v_j(t)$ which can then be solved using `LSODA`:

$$\frac{du_1}{dt} = 0 \tag{20}$$

$$\frac{du_j}{dt} = v_j \qquad j = 2, \cdots N-1 \tag{21}$$

$$\frac{du_N}{dt} = 0 \tag{22}$$

$$\frac{dv_1}{dt} = 0 \tag{23}$$

$$\frac{dv_j}{dt} = \frac{u_{j+1} - 2u_j + u_{j-1}}{h^2} \qquad j = 2, \cdots N-1 \tag{24}$$

$$\frac{dv_N}{dt} = 0 \tag{25}$$

This solution procedure is implemented by the the program `wave` (See ~`phys410/ode/wave`). You will follow an analogous approach to solve the *diffusion equation* in the final homework.