HIGH-PERFORMANCE COMPUTING IN APPLIED MATHEMATICS

The 1D wave equation

The one dimensional wave equation

$$u_{tt} = c^2 u_{xx}, \quad 0 < x < L, t > 0 \tag{1}$$

may be used to model the vertical displacement of a string from the equilibrium position, assuming that the horizontal displacement may be neglected. If the ends of the string are fixed, the boundary conditions associated to (1) are

$$u(0,t) = 0, \quad u(L,t) = 0, \quad t \geq 0 \tag{2}$$

As initial conditions we must specify the initial configuration and the initial velocity:

$$u(x,0) = f(x), \quad u_t(x,0) = g(x), \quad 0 < x < L \tag{3}$$

To obtain the numerical solution we may use a central difference scheme discretization of the second order derivatives, both in time and in space. Assuming that the spatial domain is discretized with the nodes

$$x_i = i * h, \quad i = 0 : n$$

where $h = L/n$ and the time step is $\tau = \Delta t$, the finite difference equations are written

$$\frac{u_i^{k+1} - 2u_i^k + u_i^{k-1}}{\tau^2} = c^2 \frac{u_{i+1}^k - 2u_i^k + u_{i-1}^k}{h^2} \tag{4}$$

which is a second order accurate scheme in time and space (error of order $\tau^2 + h^2$). In (4), u_i^k is the numerical approximation of the solution at time $t_k = k\tau$ and spatial location x_i.

Denoting

$$s = c^2 \frac{\tau^2}{h^2}$$

from (4) we may express

$$u_i^{k+1} = 2(1-s)u_i^k + s(u_{i-1}^k + u_{i+1}^k) - u_i^{k-1}, \quad i = 1 : n-1, \quad k \geq 1 \tag{5}$$

Notice that three time levels are involved in the finite difference equations (5). To start the computation of u_i^1 we need u_i^0 and u_i^{-1} which may be obtained from the initial conditions (3) as follows:

$$u_i^0 = f(x_i), \quad i = 1 : n-1 \tag{6}$$

and using the central difference scheme for the first derivative in time we may obtain from the initial velocity

$$\frac{u_i^1 - u_i^{-1}}{2\tau} = g(x_i) \Rightarrow u_i^{-1} = u_i^1 - 2\tau g(x_i), \quad i = 1 : n-1 \tag{7}$$

An additional relation between u_i^0, u_i^1 and u_i^{-1} may be obtained from the difference equation (5) at $t = 0$ which corresponds to $m = 0$:

$$u_i^1 = 2u_i^0 - u_i^{-1} + s\left(u_{i-1}^0 - 2u_i^0 + u_{i+1}^0\right) \tag{8}$$

such that after replacing (7) in (8) we obtain

$$u_i^1 = u_i^0 + \frac{1}{2}s\left(u_{i-1}^0 - 2u_i^0 + u_{i+1}^0\right) + \tau g(x_i), \quad i = 1 : n-1 \tag{9}$$

From the boundary conditions we have at all time steps

$$u_0^k = u_n^k = 0, \quad k \geq 0 \tag{10}$$

Equations (5), (6), (9), and (10) provide a complete description of the central time central space (CTCS) discretization of the one dimensional wave equation problem. For stability we must require that $s \leq 1$.

Domain decomposition and algorithm for parallel implementation

- Setup: assume, for simplicity, that the number of processes p evenly divides n.
- Domain decomposition:
  1. Each process defines $loc\_n = n/p$ and works on a local domain $loc\_x(0 : loc\_n + 1)$ defined

     $$loc\_x(i) = (my\_rank * loc\_n + i) * h, \quad i = 0 : loc\_n + 1$$

  2. Each process updates the interior components $1 : loc\_n$ of a local vector $loc\_u(0 : loc\_n + 1)$
  3. The endpoints $loc\_u(0)$ and $loc\_u(loc\_n + 1)$ are local "boundary" points and are used for communication among processes.
- time loop: iteration continues until the end of the time interval is reached
- do while $t < tend$
  1. Communication at each time step:

     Process 0: sends $loc\_u(loc\_n)$ to process 1 and receives $loc\_u(loc\_n + 1)$ from process 1

     Process p-1: sends $loc\_u(1)$ to process p-2 and receives $loc\_u(0)$ from process p-2

     Process 1 to p-2: sends $loc\_u(1)$ to process my_rank - 1 and receives $loc\_u(0)$ from process my_rank - 1; sends $loc\_u(loc\_n)$ to process my_rank + 1 and receives $loc\_u(loc\_n + 1)$ from process my_rank + 1;

     Each process updates loc_unew using the CTCS scheme in terms of loc_u and loc_uold

     Each process sets $loc\_uold = loc\_u$ and $loc\_u = loc\_unew$, $t = t + \Delta t$

See ***wave1d.f***