# 2-D Wave Equation Program: `wave2d`

Isaac Dooley

November 24, 2008

The wave equation can be stated in the following form,

$$\frac{\partial^2 p}{\partial p^2} = c^2 \nabla^2 p, \tag{1}$$

where p is a physical measure such as pressure or the depth of a body of water in a container. This equation can be discretized over a 2d grid using a finite differencing scheme. Assume the pressures are located across a rectangular 2-D grid in x and y dimensions. We call the value of the pressure $p^t_{x,y}$ for time $t$ at location $(x, y)$ in the grid. One solution to the differential equation (1) over a grid is

$$p^{t+1}_{x,y} + p^{t-1}_{x,y} - 2p^t_{x,y} = c^2 \left( p^t_{x+1,y} + p^t_{x-1,y} + p^t_{x,y+1} + p^t_{x,y-1} - 4p^t_{x,y} \right). \tag{2}$$

The left hand side term $\frac{\partial^2 p}{\partial p^2}$ is represented in terms of the values of $p$ at three consecutive timesteps at grid location $(x, y)$. The right hand side terms are discretized using the $p$ values for the 4 locations adjacent to $(x, y)$ in the grid. One can simply solve for $p^{t+1}_{x,y}$ in equation (2) to produce an update rule,

$$p^{t+1}_{x,y} = c^2 \left( p^t_{x+1,y} + p^t_{x-1,y} + p^t_{x,y+1} + p^t_{x,y-1} - 4p^t_{x,y} \right) - p^{t-1}_{x,y} + 2p^t_{x,y}. \tag{3}$$

This update rule specifies how the $p$ value for each location in the grid should be computed, using values on the grid from the two previous time steps. This update rule is easy to code in an application. The application simply maintains two timesteps' worth of $p$ grids, and using these, the next timestep's $p$ grid can be computed. The $c$ term determines the wave speed. This value must be small enough such that the wave cannot move across more than one grid square in a single timestep. Smaller values for $c$ will make the simulation take longer to propagate a wave by a fixed distance, but larger values for $c$ can introduce dispersive and diffusive errors.

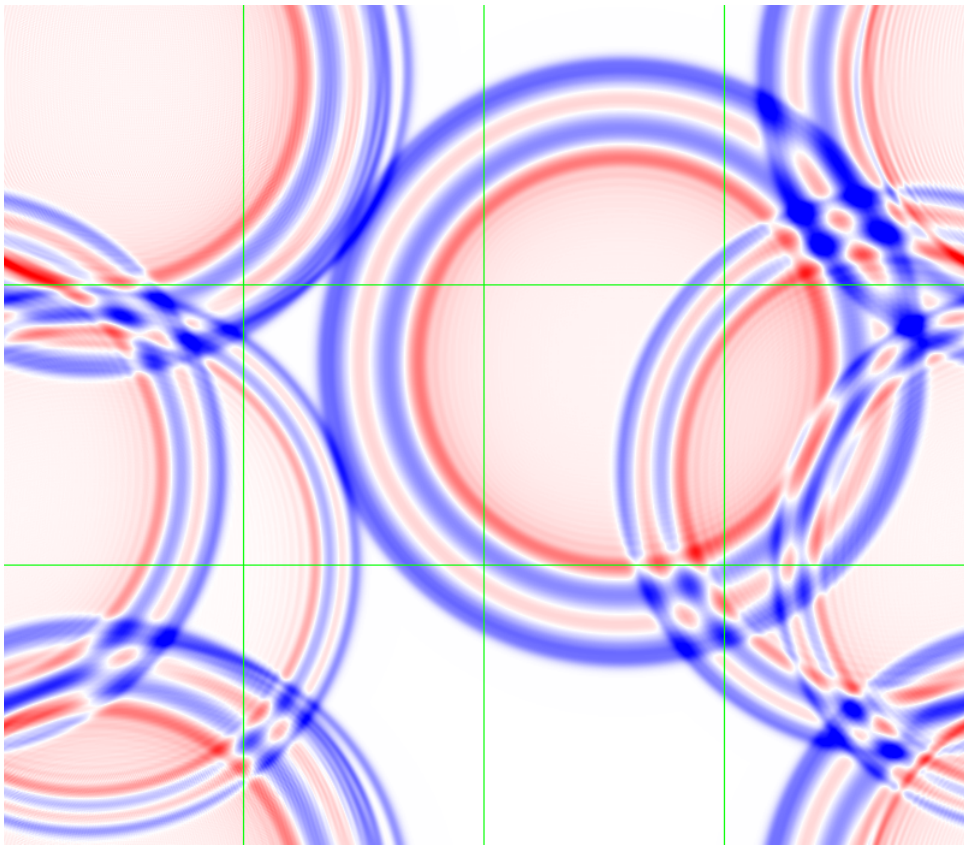

Figure 1: A screenshot of the `wave2d` program. Positive values of $p$ on the grid are colored red, while negative values are colored blue. The green lines show the parallel decomposition of the problem onto a $4 \times 3$ chare array. The grid domain logically wraps around from left to right and top to bottom.