CactusWave

Gabrielle Allen, Tom Goodale

1999

Abstract

Set of thorns for evolving the standard 3D scalar wave equation. The package includes initial data, evolver, and analysis thorns in a variety of programming languages.

1 Purpose

To demonstrate the use of the Cactus code through a simple, illustrative example.

The model problem solved is the 3D scalar wave equation in Cartesian coordinates,

$$\frac{\partial^2 \phi}{\partial t^2} = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2}$$

The numerical solution of this equation requires initial data to be specified for

$$\phi(t=0), \qquad \frac{\partial \phi}{\partial t}(t=0)$$

The numerical method employed in these thorns to solve for ϕ is a standard 2nd order centered finite difference method. The solution $\phi(t,x,y,z)$ is discretised using

$$\phi(t_i, x_i, y_i, z_i) = \phi^n_{i,j,k}$$

where, for example,

$$x_i = x_0 + i\Delta x$$

The solution at any timeslice can then be found iteratively using the previous two timeslices using the algorithm

$$\begin{aligned} \phi^{n+1} &= 2(1-\rho_x^2-\rho_y^2-\rho_z^2)\phi^n_{i,j,k} - \phi^{n-1}_{i,j,k} + \rho_x^2(\phi^n_{i+1,j,k} - \phi^n_{i-1,j,k}) \\ &\quad +\rho_y^2(\phi^n_{i,j+1,k} - \phi^n_{i,j-1,k}) + \rho_z^2(\phi^n_{i,j,k+1} - \phi^n_{i,j,k-1}) \end{aligned} \tag{1}$$

where we define the Courant factors

$$\rho_x = \frac{\Delta t}{\Delta x} \qquad \rho_y = \frac{\Delta t}{\Delta y} \qquad \rho_z = \frac{\Delta t}{\Delta z}$$

2 Comments

Here we give a brief description of each of the thorns contained in this arrangement

- **IDScalarWave** Different initial data sets, all of which are analytic.
- **IDScalarWaveCXX** The same as IDScalarWave but implemented in C++.
- **IDScalarWaveElliptic** Initial data sets from solving an elliptic equation. At the moment this initial data is rather artificial, and is just here to give a simple demonstration of using an elliptic solver.

- **WaveToyC** The evolver for the scalar field, written in C.
- **WaveToyF77** The same as WaveToyC, but written in F77 to demonstrate the use of `implementation`s.
- **WaveToyF90** The same as the two evolver thorns above, but this time to show the difference between F77 and F90, and to further demonstrate `implementation`s.
- **WaveToyFreeF90** The same as WaveToyF90, but written with free-format F90 rather than fixed.
- **WaveToyCXX** The same as WaveToyC, but written in C++.