begin thorn SimpleWaveScriptCaKernel2

use cakernel

# Set up parameters begin parameters
amp : real "The amplitude of the wave",
default: 1.0, range: -infinity to infinity
kfac : real "The wave number",
default: 2 PI, range: 0 to 10 PI
c0 : real end parameters

# Initial data calculation begin calculation initial_sine_calc scheduled at initial
phi = amp * \sin(kfac * x)
pi = -amp * c0 * \cos(kfac * x) end calculation

# Main evolution equation begin calculation calc_rhs scheduled at MoL_CalcRHS
D_t phi = pi
D_t pi = c0**2 \text{Euc}^ij \cdot D_{ij} \phi end calculation

# Boundary conditions begin calculation calc_bound_rhs scheduled at MoL_CalcRHS on boundary
D_t phi = pi
D_t pi = 0 end calculation

begin variables
phi pi end variables

###

# Define the stencil for derivatives begin operators
# Applies to x, y, or z directions in 3D Df_ii xdiv = (xdiv[i+1] + xdiv[i-1] - 2 * xdiv[i]) \cdot del_i**(-2)
# Applies to x-y, y-z, or x-z in 3D Df_ij xdiv = (xdiv[i+1,j] - xdiv[i-1,j] - xdiv[i,j+1] + xdiv[i,j-1]) / (4 \cdot del_j \cdot del_i) end operators

end thorn

The fundamental difficulty with scientific software is that the existing languages are either too general-purpose, too low-level, or both. Thus, physics equations are entangled with the way they are implemented, even to low-level details such as grid structure, cache optimization, etc. Because everything is coupled, it is difficult for the various scientific communities to adapt to algorithm or hardware advances. More importantly, it means that researchers cannot specialize as easily.

In principle, it should be possible to independently specify (1) the scientific equations to be solved; (2) the type of grids; (3) the type of numerical methods, e.g. time integrators, elliptic solvers; (4) the intended execution platform (desktop, accelerated cluster, etc.); (5) performance goals, e.g. as fast as possible, minimum cost, etc.

At left, we show how we have partially realized this idea in the Chemora project. We specify physics equations, grid variables, and parameters (scalars) in a high-level way that's easy to parse. Operators are used inside the equations, but must be specified in a separate section (bottom) as they are part of the implementation.

Note that it's still useful to specify this detail at a high level because it allows the machine-specific optimizations to be kept separate from the stencil definition.