Implementation of the Yin-Yang grid in PENCIL

Matthias Rheinhardt

August 10, 2016
Motivation

Models in spherical geometry with full $\theta - \phi$ extent

- $\theta = 0$ can’t be a coordinate line
- ghost zones for $\theta$ boundaries lie beyond the poles
- hence: define grid as $\theta_i = \Delta \theta / 2 + i \Delta \theta, i = 0, \ldots, \pi / \Delta \theta - 1$, for grid point at $\phi_j$ fill ghost zones with values from $\phi_j + \pi$
  implemented by Dhruba/Fred
- problem: for $\theta$ grid lines close to poles stepsize in $\phi$ direction $r \sin \theta \Delta \phi$ gets small $\implies \Delta t$ gets (too) small
- possible solution: make grid non-uniform in $\theta$, e.g.:
Yin-Yang grid for simulations over the full $\theta - \phi$ extent

Alternative:

- cover spherical surface by 2 overlapping *identical* grids
- axis singularity of one grid covered regularly by the other
- grid cell size roughly uniform
- tb added: communication between Yin and Yang; algorithms internal to each grid untouched

$\implies$ code extensibility

decomposition without overlap

with overlap

Rheinhardt
coordinate ranges
\[ \frac{\pi}{4} \leq \theta \leq \frac{3\pi}{4}, \quad \Delta \theta = \frac{\pi}{2} \]
\[ \frac{\pi}{4} \leq \phi \leq \frac{7\pi}{4}, \quad \Delta \phi = \frac{3\pi}{2} \]

transformation matrix
\[
M = \begin{bmatrix}
-1 & 0 & 0 \\
0 & 0 & -1 \\
0 & -1 & 0 \\
\end{bmatrix}
\]

\[ M = M^T = M^{-1} \]

\[ \implies \text{only one transformation} \]

**PENCIL CODE:** possible strategies
- double the variables
- **double the processors**

\[ \implies \text{modification of communication only} \]
Implementation in PENCIL CODE

- switch on by `lyinyang=T`
- initialize
  - check constraint `nprocz = 3 nprocy`
  - set implicitly `nprocs = 2 ncpus`
- create a MPI communicator for each grid:
  - `MPI_COMM_GRID ≠ MPI_COMM_WORLD`
- for boundary processors: set outer neighbours
- transform and communicate ghost point coordinates
- calculate interpolation parameters
- transform global input data
- run
  - transform (vectors) and interpolate variables
  - communicate for ghostzone update
  - correct averages
- diagnostics
  - transform/interpolate data on Yang grid
Problems

standard layout:
each processor has exactly 8 neighbours in $\theta - \phi$ plane
$\implies$ restrictions for processor numbers

\begin{align*}
4 \times 12 & \quad 8 \times 24
\end{align*}
standard layout:
each processor has exactly 8 neighbours in $\theta - \phi$ plane
implies restrictions for processor numbers
Implementation in *Pencil Code*

**in code:**
- additional modules `yinyang`, `yinyang_mpi`, `noyinyang`
- + additional subroutines in `general`, `mpicomm`

**in setup:**
- `YINYANG = yinyang` (default `noyinyang`)
- `ncpus` — number of processors for one grid
  (but in submit script: `2*ncpus`!)
Implementation in PENCIL CODE

in visualisation:

- object of \texttt{pc\_read\_var} contains usual variables, but with additional dimension of extent 2 for the two grids

Yin-Yang specific:

- \texttt{YZ}, dimension\((2,*\)} - a linear list of \((\theta, \phi)\) coordinate pairs for the merged grids; technically an irregular grid

- \texttt{TRIANGLES}, dimension\((3,*\)} - a list of triangles describing the triangulation of the merged grid

- \texttt{UU\_MERGE}, dimension\((nxgrid,(size(YZ))(2),3\)} - velocity defined on the merged grids
use merged data by, e.g.:

\[\text{contour, reform}(v.\text{uu}_\text{merge}(ir,*,0)), v.yz(1,*), v.yz(0,*), /fill, nlev=30, \text{tri}=v.\text{triangles}\]

stellar convection:

with grid
Problem

discontinuities/whiggles at grid interface
example: decay of dipolar meridional flow
Implementation in PENCIL CODE

Solution: biquadratic interpolation?

\[ f(y, z) = a_0 + a_1 y + a_2 z + a_3 yz + a_4 y^2 + a_5 y^2 z + a_6 y z^2 + a_7 y^2 z^2 \]

not unique:
Implementation in **PENCIL CODE**

Solution: biquadratic interpolation?

\[ f(y, z) = a_0 + a_1 y + a_2 z + a_3 yz + a_4 y^2 + a_5 y^2 z + a_6 yz^2 + a_7 y^2 z^2 \]

not unique:
Implementation in PENCIL CODE

Solution: biquadratic interpolation?

\[ f(y, z) = a_0 + a_1 y + a_2 z + a_3 y z + a_4 y^2 + a_5 y^2 z + a_6 y z^2 + a_7 y^2 z^2 \]

not unique:
Solution: biquadratic interpolation?

\[ f(y, z) = a_0 + a_1 y + a_2 z + a_3 yz + a_4 y^2 + a_5 y^2 z + a_6 yz^2 + a_7 y^2 z^2 \]

not unique:

How to weigh the 4 variants?
Implementation in \textit{Pencil Code}

\textbf{Status}

- initialization \& communication — done
- linear \& quadratic interpolation — in testing
- \textit{z} averages: for diagnostics — in debugging
  for PDEs — in coding
- \textit{y} and volume averages — missing
- slices: \textit{yz} — done, other — missing
- visualization: reading snapshots, \textit{z} averages \& \textit{yz} slices — done