The Work of Japanese Researchers at the Earth Simulator Center

1. Pseudo-Compressibility Method in 3-D Mantle Convection Charley Kameyama

2. Ying-Yang Scheme in discretizing the Spherical Geometry: An Attempt to Overthrow the Spherical Harmonic Reich

Akira Kageyama

ACuTEMan

A multigrid-based mantle convection simulation method and its optimization to the Earth Simulator

Masanori Kameyama

kameyama@jamstec.go.jp

Earth Simulator Center (ESC)

Japan Agency for Marine-Earth Science and Technology

Email: kameyama@jamstec.go.jp

URL: http://www.es.jamstec.go.jp/esc/research/Solid/members/kameyama/

Simulation Procedure in General

(Anelastic fluid and Boussinesq approximations are assumed)

do time-marching loop

solve energy equation (update temperature T)

$$\frac{DT}{Dt} = \nabla \cdot (\kappa \nabla T) + q$$

solve for flow field (update velocity v and pressure p)

$$\begin{aligned} 0 &= \nabla \cdot \boldsymbol{v} \\ 0 \simeq \frac{1}{Pr}\frac{D\boldsymbol{v}}{Dt} &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2}(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \nabla)\right] + Ra\, T \boldsymbol{e}_z \end{aligned}$$

end do time-marching loop

Difficulty in Simulations

Solution for flow field is very time-consuming

$$
\begin{aligned}
0 \simeq \frac{1}{Pr}\frac{D\boldsymbol{v}}{Dt} &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2}\left(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \boldsymbol{v}\right)\right] + Ra\, T \boldsymbol{e}_z \\
0 &= \nabla \cdot \boldsymbol{v}
\end{aligned}
$$

Causes of difficulty

- ❑ incompressible (or anelastic) fluid
  - ⇒ no equation for directly solving pressure field
- ❑ extremely high viscosity ($\eta \sim 10^{22}$ Pa s)
  - ⇒ inertia term becomes negligibly small ($Pr \sim 10^{24}$)
  - ⇒ steady-state flow must be solved at every timestep
- ❑ strong spatial variation in viscosity (in vertical/horizontal direction)
  - ⇒ spectral method is not suitable

Aim of This Study

To develop an efficient algorithm

- (from viewpoint of mantle convection simulations)

  for solving $\left\{ \begin{array}{l} \text{steady-state flow of} \\ \text{incompressible fluid with} \\ \text{strong spatial variation in viscosity} \end{array} \right.$

  in large-scale 3-D domain

$$\begin{aligned} 0 &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2} \left(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \nabla \right) \right] + Ra\, T \boldsymbol{e}_z \\ 0 &= \nabla \cdot \boldsymbol{v} \end{aligned}$$

- (from computational viewpoint)

  for solving $\left\{ \begin{array}{l} \text{a set of elliptic equations} \\ \text{with strongly varying coefficients and} \\ \text{large number of unknowns} \end{array} \right.$

  → making good use of multigrid technique

  → with sufficient vector/parallel efficiency

Proposed Algorithm (1)

For steady-state flow of highly viscous incompressible fluid

$$
\begin{aligned}
0 &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2}(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \nabla)\right] + Ra\,T\boldsymbol{e}_z \\
0 &= \nabla \cdot \boldsymbol{v} \quad (\text{for given temperature } T \text{ and viscosity } \eta)
\end{aligned}
$$

Ingredient 1: Pseudo-Compressibility Method

- integrates pseudo-evolutionary equations for $\boldsymbol{v}$ and p to steady-state
- describes pseudo-evolution of pressure p using artificial compressibility

$$
\begin{aligned}
M\frac{\partial \boldsymbol{v}}{\partial \tau} &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2}(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \nabla)\right] + Ra\,T\boldsymbol{e}_z \\
-K\frac{\partial p}{\partial \tau} &= \nabla \cdot \boldsymbol{v}
\end{aligned}
$$

where τ : pseudo-time
M: pseudo-density
K : pseudo-compressibility

Pseudo-Compressibility Method ?

- The good
  - always converges to incompressible flow field (owing to viscous damping)

$$\frac{\partial^2}{\partial\tau^2}(\nabla\cdot\boldsymbol{v}) = \underbrace{\frac{1}{KM}\nabla^2(\nabla\cdot\boldsymbol{v})}_{\text{"pseudo-sound"}} + \underbrace{\frac{\eta}{M}\frac{\partial}{\partial\tau}\left[\nabla^2(\nabla\cdot\boldsymbol{v})\right]}_{\text{viscous damping}} + \cdots$$

  - deals with primitive variables (velocity $\boldsymbol{v}$ and pressure p)
    → applicable both to 2-D and 3-D problems
  - simple and easy
    → only integrates (pseudo-)evolutionary equations
- The bad
  - converges very slowly
    → very slow reduction in long-wavelength error (because of diffusion equation)

$\Rightarrow$ must be used with multigrid method

Proposed Algorithm (2)

What happens in case with spatial variation in viscosity η ?

$$
\begin{aligned}
M\frac{\partial \boldsymbol{v}}{\partial \tau} &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2}\left(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \nabla\right)\right] + Ra\, T \boldsymbol{e}_z \\
-K\frac{\partial p}{\partial \tau} &= \nabla \cdot \boldsymbol{v}
\end{aligned}
$$

"local convergence rate" for velocities $\boldsymbol{v} \propto \eta$
$\rightarrow$ yields slow convergence where η is small

Ingredient 2: Local Time-Stepping Method

- varies "effective" time-steppings $\Delta\tau/M$ and $\Delta\tau/K$ in space
  - ⇒ simply achieved by varying "artificial" density M and compressibility K in space
  - ⇒ neither M nor K needs to be "realistic" values

Proposed Algorithm (3)

Appropriate form of spatial variations in M and K ?

$$\begin{aligned} M\frac{\partial \boldsymbol{v}}{\partial \tau} &= -\nabla p + \nabla \cdot \left[\frac{\eta}{2}\left(\nabla \otimes \boldsymbol{v} + \boldsymbol{v} \otimes \nabla\right)\right] + Ra\,T\boldsymbol{e}_z \\ -K\frac{\partial p}{\partial \tau} &= \nabla \cdot \boldsymbol{v} \end{aligned}$$

Hint: Pseudo-temporal evolution of flow field is characterized by,

$$\frac{\partial^2}{\partial \tau^2}\left(\nabla \cdot \boldsymbol{v}\right) = \underbrace{\frac{1}{KM}\nabla^2\left(\nabla \cdot \boldsymbol{v}\right)}_{\text{"pseudo-sound" propagation}} + \underbrace{\frac{\eta}{M}\frac{\partial}{\partial \tau}\left[\nabla^2\left(\nabla \cdot \boldsymbol{v}\right)\right]}_{\text{viscous damping}} + \cdots$$

To remove influences of variation in η,

- ❑ minimize spatial variations in

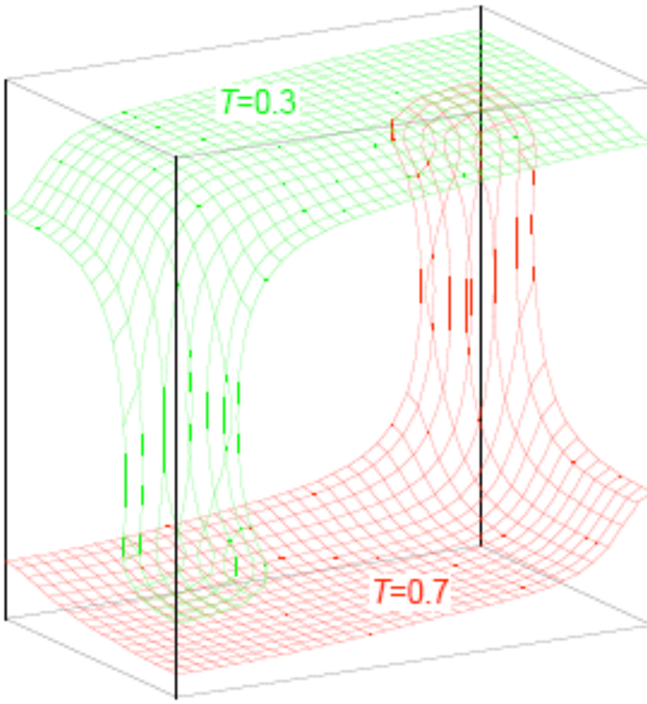
$$\begin{cases} \text{rate of effective viscous "damping" } \eta/M \\ \text{rate of "pseudo-sound" propagation } 1/\sqrt{KM} \end{cases}$$

$$\boxed{\begin{array}{c}\text{Choose} \\ M \propto \eta \\ K \propto \eta^{-1}\end{array}}$$

Benchmark Test

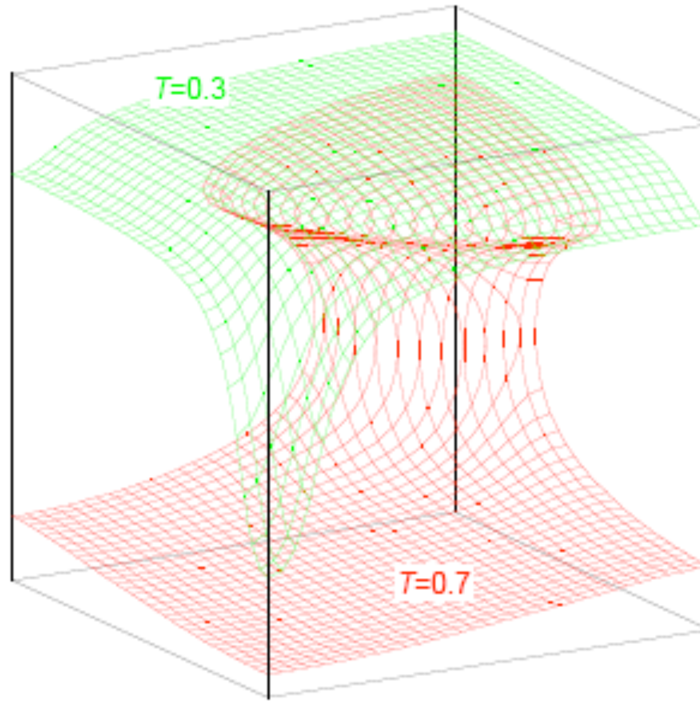
with Busse et al. (1993)

Case 1a
constant viscosity
$Ra = 30000$
$1.0079 \times 0.6283 \times 1$ box



	$32\times16\times32$	$64\times32\times64$	best estimates
Nu	3.6019	3.5549	3.5374 ± 0.0005
V_{rms}	41.382	41.104	40.999 ± 0.004

Case 2
temperature-dependent viscosity
Ra=20000, viscosity contrast=20
$1\times1\times1$ box



	$32\times32\times32$	$64\times64\times64$	best estimates
Nu	3.0503	3.0419	3.0393 ± 0.0050
V_{rms}	35.161	35.130	35.13 ± 0.05

Comparison with Ogawa et al. (1991)

for strongly temperature-dependent viscosity

$1.7 \times 0.5 \times 1$ box

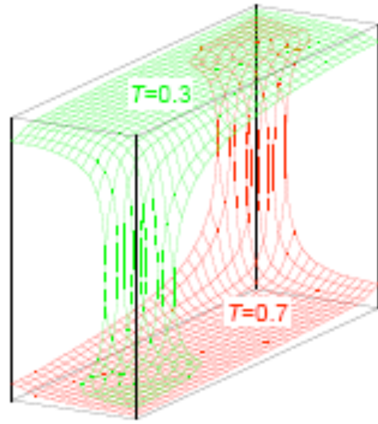
$64 \times 32 \times 64$ mesh

$\eta = \eta_t \exp[E(T_t - T)]$

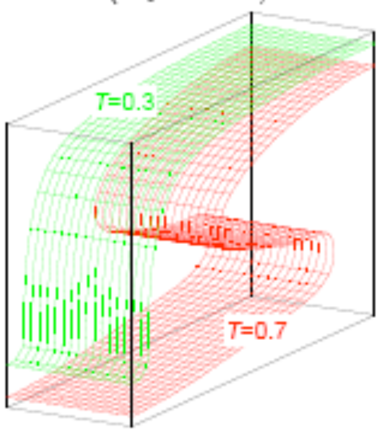
$R_t = \frac{\rho g \alpha (T_b - T_t) d^3}{\kappa \eta_t}$

$r = \exp[E(T_b - T_t)] = \frac{\eta_{\max}}{\eta_{\min}}$

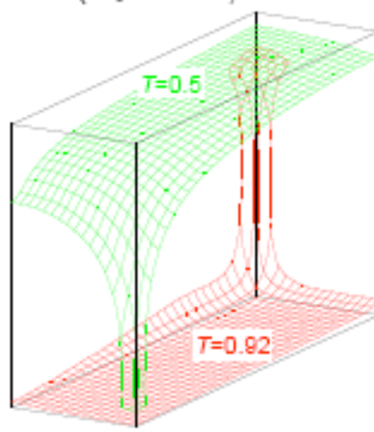
Case 1 ($R_t = 10^5, r = 1$)



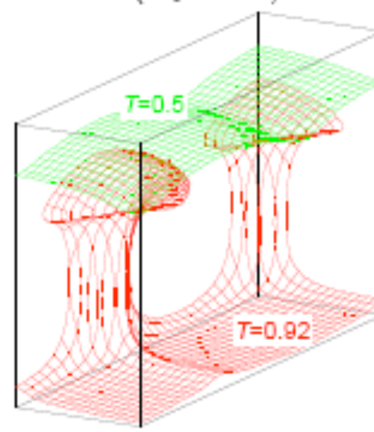
Case 4 ($R_t = 10^3, r = 10^2$)



Case 16 ($R_t = 10^3, r = 3.2 \times 10^3$)



Case 18 ($R_t = 32, r = 10^5$)



computational time: 3 sec/step for Case 1 (with no viscosity contrast)

18 sec/step for Case 18 (with 10^5 viscosity contrast)

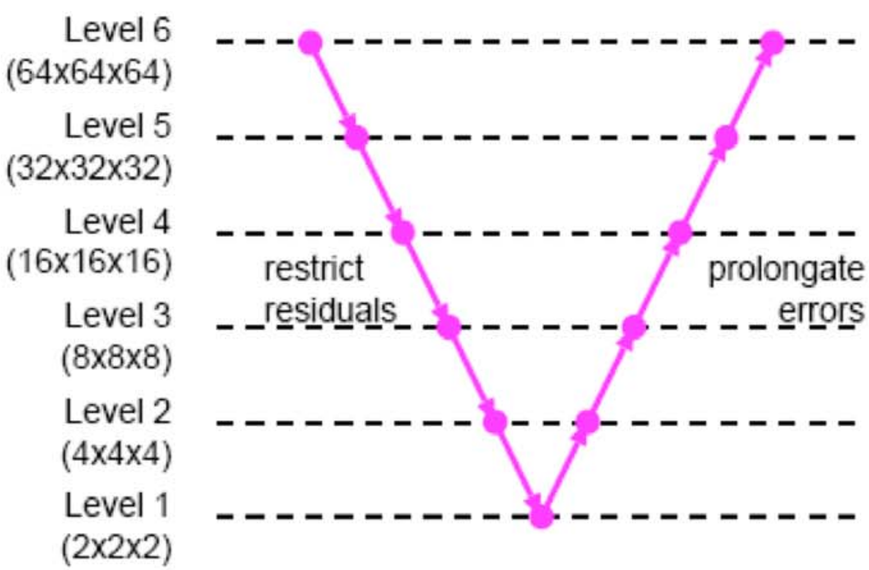
with Pentium IV 2.20GHz (NB: started from different initial conditions)

Convergence Tests (0)

Pseudo-temporal integration is used as a smoother of multigrid method

Example: 3-D thermal convection in a cube with $64 \times 64 \times 64$ mesh divisions and strongly temperature-dependent viscosity

Multigrid V-cycle
(Correction Storage Scheme)



Number of pre/post-smoothing

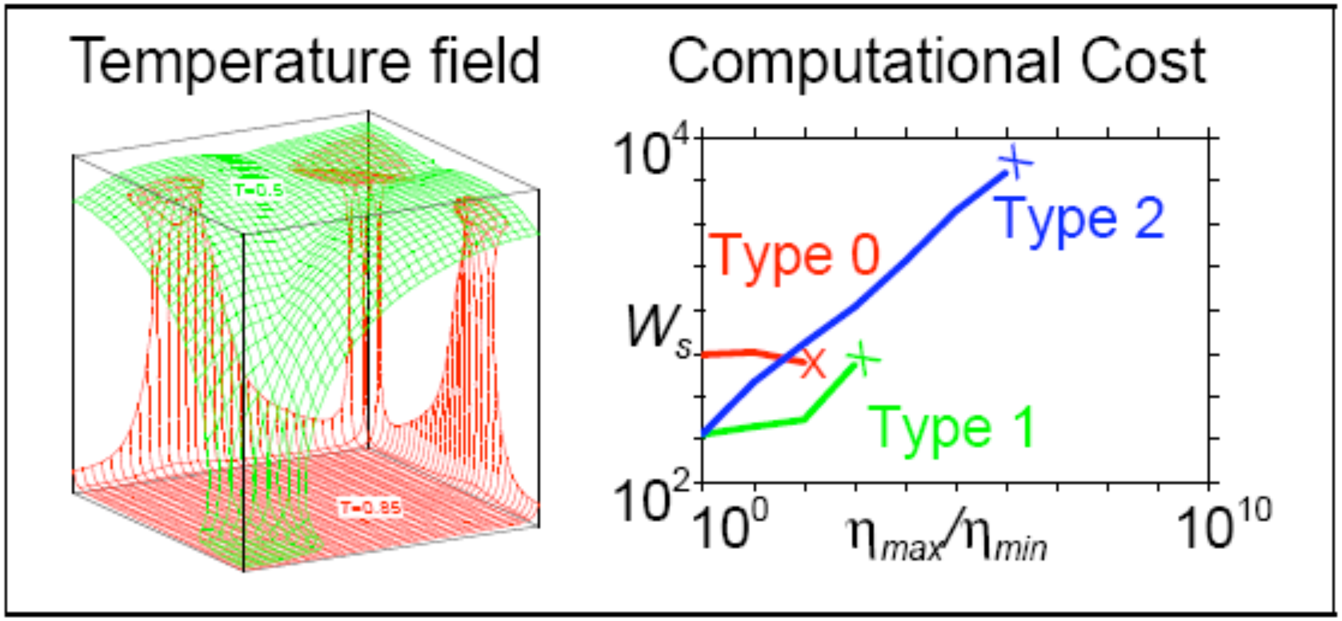
Type 0	Type 1	Type 2
8	8	$8 \times [\ln(\eta_{max}/\eta_{min})+1]$
8	16	$16 \times [\ln(\eta_{max}/\eta_{min})+1]$
8	32	$32 \times [\ln(\eta_{max}/\eta_{min})+1]$
8	64	$64 \times [\ln(\eta_{max}/\eta_{min})+1]$
8	128	$128 \times [\ln(\eta_{max}/\eta_{min})+1]$
8	256	$256 \times [\ln(\eta_{max}/\eta_{min})+1]$

3 Types of pre/post-smoothing iterations are tested
$\rightarrow$ to find robust implementation for strong viscosity variation

Convergence Tests (2)

For the case with larger local temperature variations

Temperature-dependent viscosity $\eta \propto \exp\left[-T \ln\left(\eta_{\max}/\eta_{\min}\right)\right]$

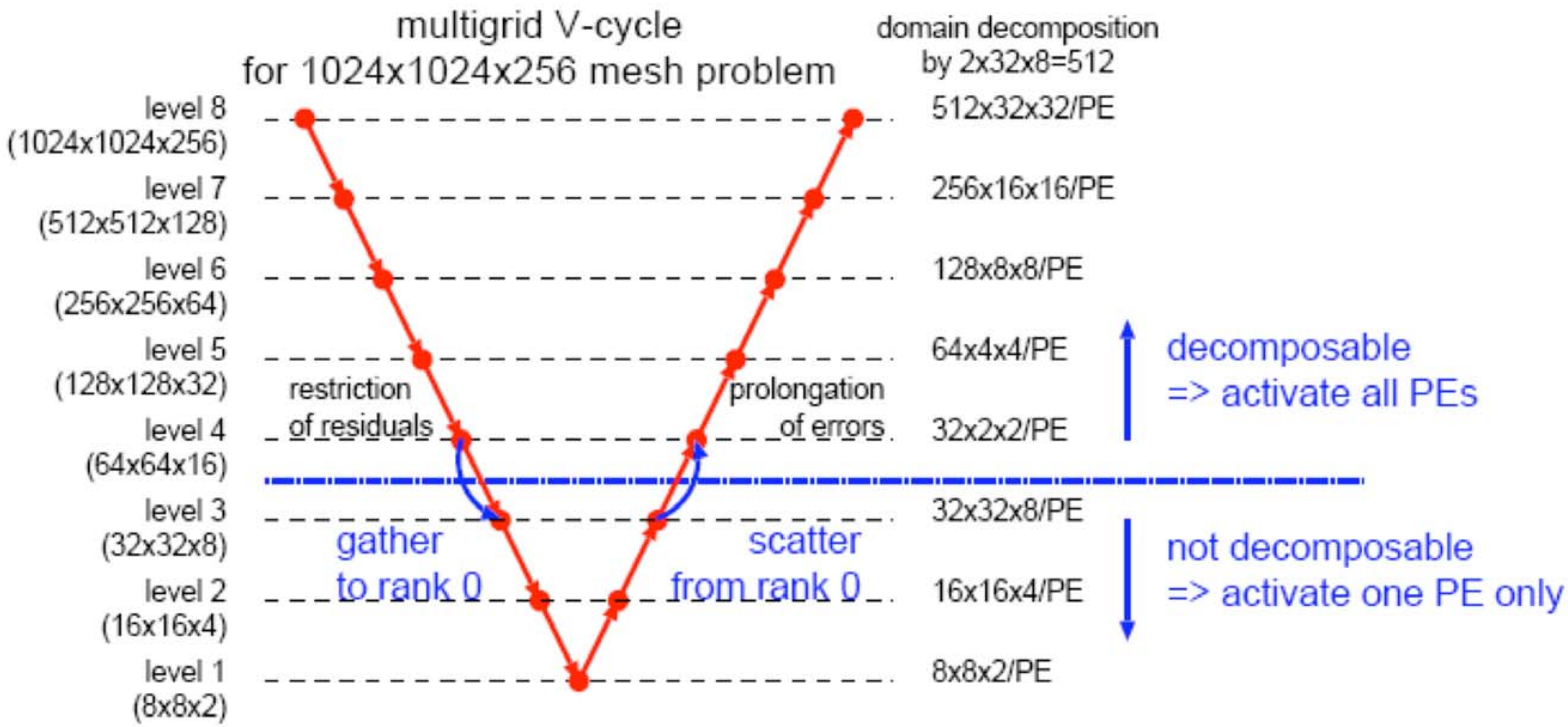


- Convergent up to $\eta_{\max}/\eta_{\min} = 10^6$
- Larger local temperature variation $\rightarrow$ Larger local viscosity variation $\Rightarrow$ More “ill-conditioned”
- Better choice of initial condition can lead to convergence

Multigrid Optimization

- "agglomeration": Key to massive parallelization of multigrid calculations
  - ⇒ activate all PEs in calculations on fine grid levels
  - ⇒ activate 1 PE only in calculations on coarse grid levels

  to reduce communication overheads



Performance on the Earth Simulator

$1024 \times 1024 \times 256$ mesh division, Multigrid 8 levels, 64PN(=512PE)

$Ra = 10^7$, constant viscosity, 1000 timesteps

```
MPI Program Information:
========================
Note: It is measured from MPI_Init till MPI_Finalize.
      [U,R] specifies the Universe and the Process Rank in the Universe.
Global Data of 512 processes:         Min [U,R]            Max [U,R]         Average
=============================
Real   Time (sec)             :      2167.564 [0,79]      2168.180 [0,184]     2168.026
User   Time (sec)             :      2148.951 [0,398]     2161.454 [0,219]     2156.579
System Time (sec)             :         3.240 [0,231]        7.921 [0,280]        3.950
Vector Time (sec)             :      1886.794 [0,70]      1999.746 [0,0]       1907.307
Instruction Count             : 172586180925 [0,56]   204325670446 [0,0]   175903786565
Vector Instruction Count      :  45725028280 [0,1]     59885749447 [0,0]    46845002132
Vector Element Count          : 8972700070279 [0,295] 11284593905563 [0,0]  9115733866864
FLOP Count                    : 3141552699120 [0,1]   4245768380296 [0,0]   3222272951859
MOPS                          :      4219.400 [0,231]     5290.575 [0,0]       4286.788
MFLOPS                        :      1454.453 [0,1]       1965.394 [0,0]       1494.161
Average Vector Length         :       188.435 [0,0]        199.675 [0,1]        194.601
Vector Operation Ratio (%)    :        98.584 [0,124]       98.736 [0,0]         98.604
Memory size used (MB)         :       979.454 [0,1]       1043.454 [0,2]       1011.461

Overall Data:
=============
Real   Time (sec)             :      2168.180
User   Time (sec)             :   1104168.311
System Time (sec)             :      2022.255
Vector Time (sec)             :    976541.389
GOPS    (rel. to User Time)   :      2194.835
GFLOPS  (rel. to User Time)   :       765.010
Memory size used (GB)         :       505.730
```

2.2 sec/step

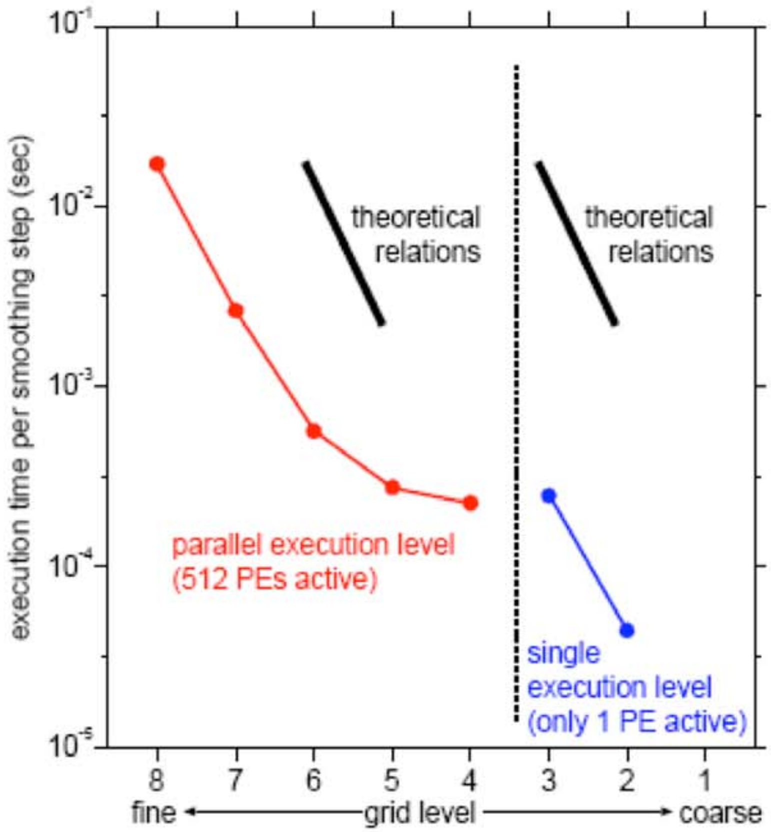
18.68% of peak performance

99.91% parallelization efficiency

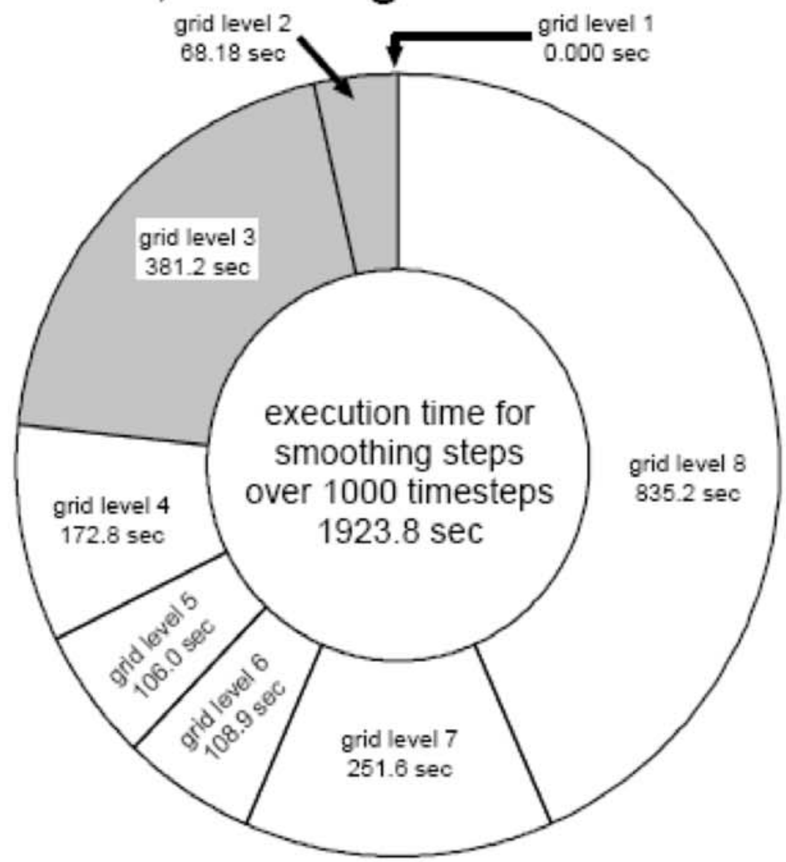
fastest "made-in-Japan" code

Impact of "Agglomeration"

Results from $1024 \times 1024 \times 256$ meshes, 8 multigrid levels



plots of execution times of one smoothing calculation against grid levels



ratios of execution times of total smoothing calculations for all grid levels

- "agglomeration" considerably improves the efficiency of coarse-grid calculations (by removal of communication overheads)
  $\rightarrow$ no serious loss of overall efficiency

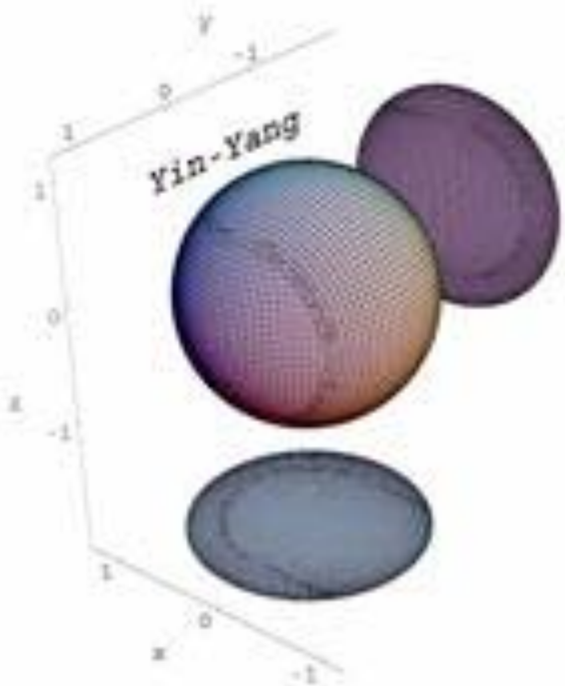
References

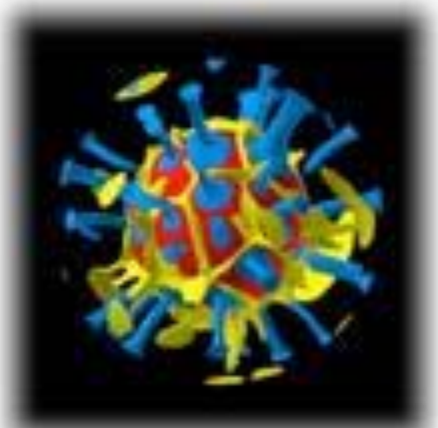
- Masanori Kameyama, Akira Kageyama, and Tetsuya Sato, Multigrid iterative algorithm using pseudo-compressibility for three-dimensional mantle convection with strongly variable viscosity,
  Journal of Computational Physics, 206 (1), 162-181, 2005.
- Masanori Kameyama,
  ACuTEMan: A multigrid-based mantle convection simulation code and its optimization to the Earth Simulator,
  Journal of the Earth Simulator, 4, 2-10, 2005.
  available via:
  http://www.es.jamstec.go.jp/esc/eng/publications/

Yin-Yang Grid

The Earth Simulator Center
JAMSTEC, Japan

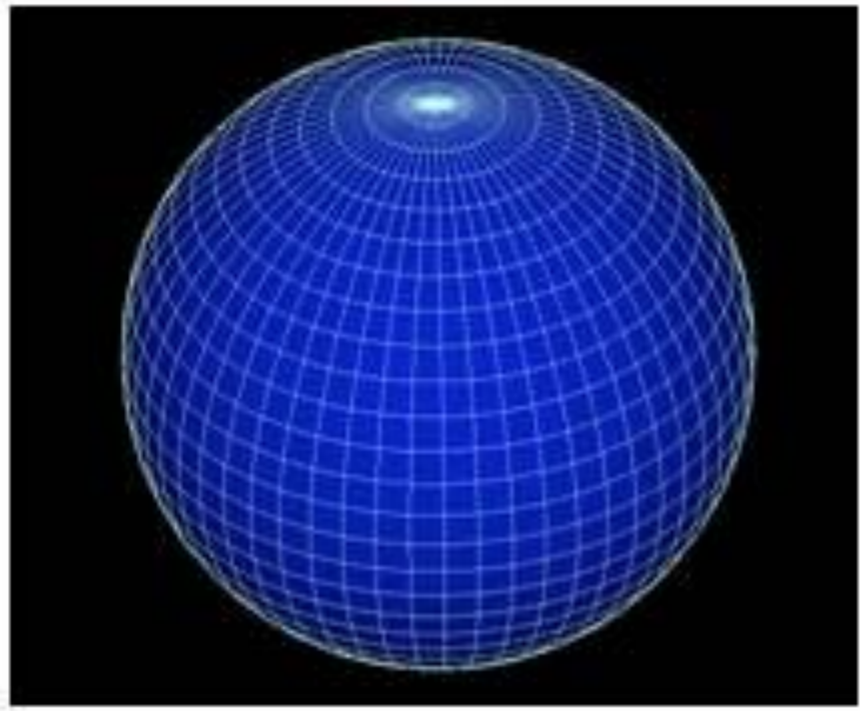
Akira Kageyama





Two "Pole Problems" of the Latitude-Longitude (lat-lon) Grid

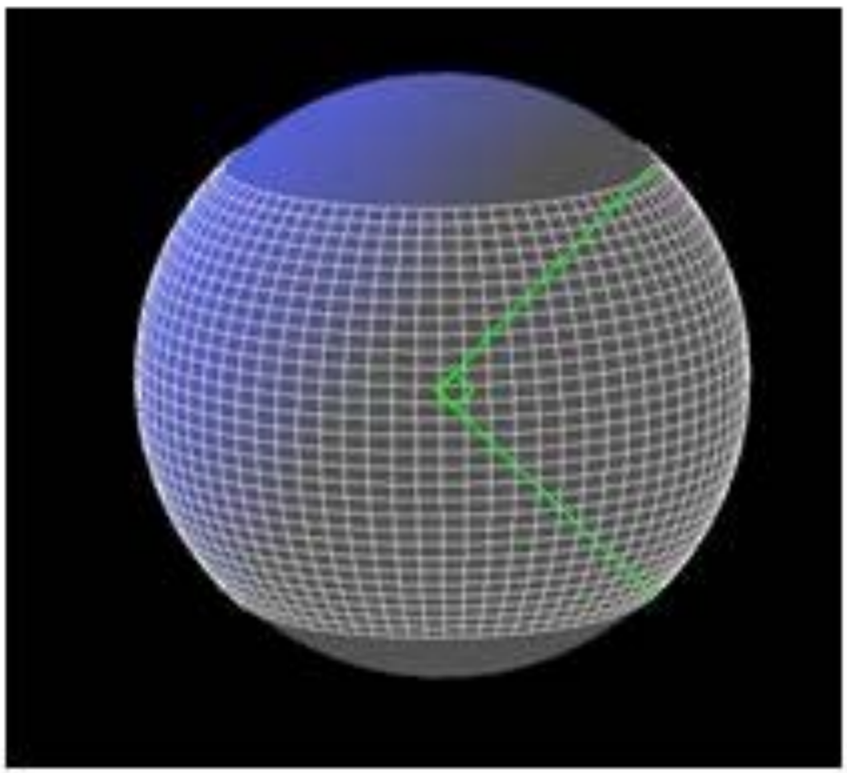
1. Coordinate singularity *on* the poles
    - special care should be taken
2. Grid convergence *near* the poles
    - needs a spherical filter for CFL
    - waste of CPU time



Re-view the Lat-Lon Grid

It is almost ideal grid in the low latitude region.

- It is orthogonal coordinates (simple metrics)
- Nearly uniform grid spacing

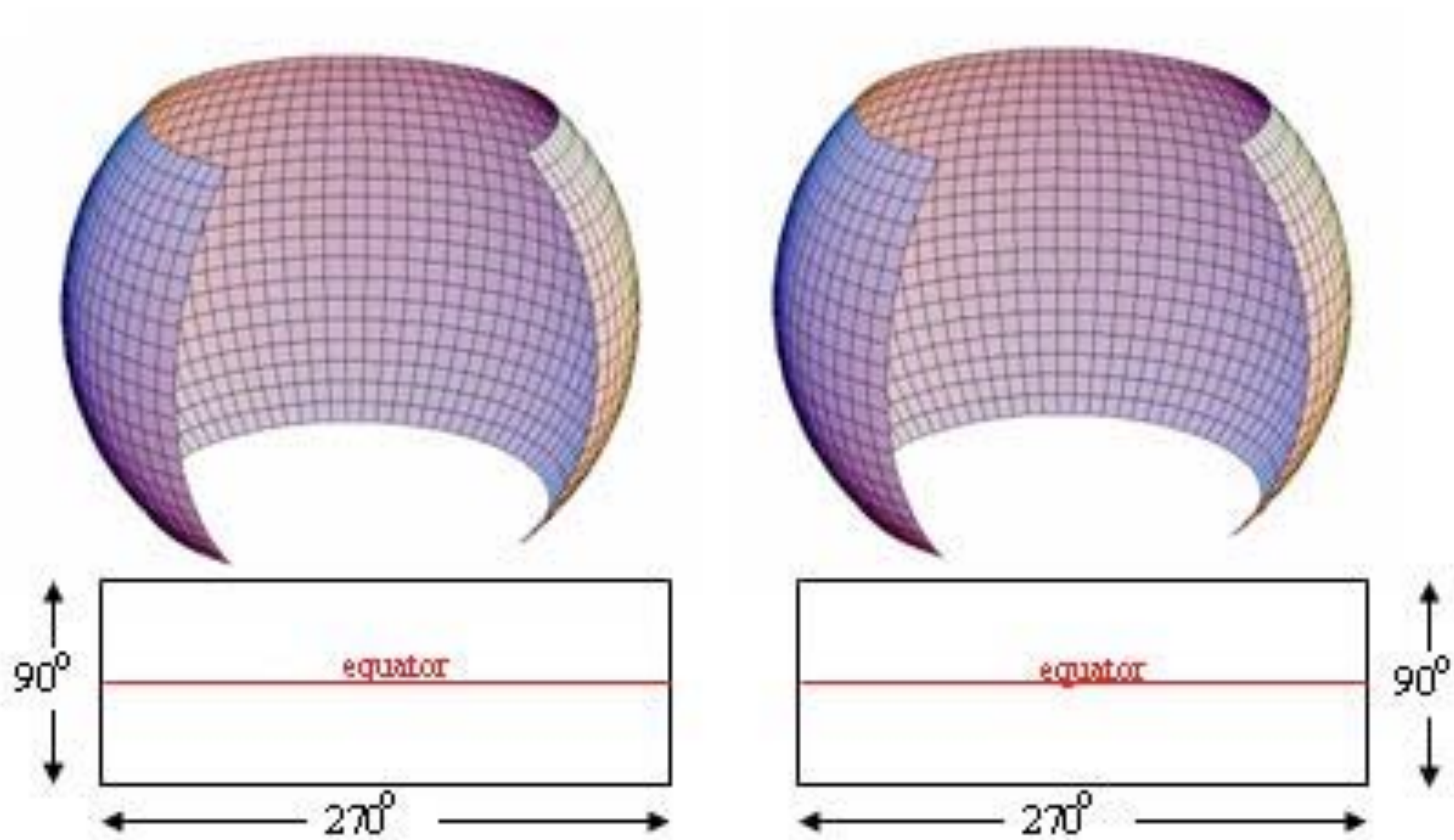


Baseball

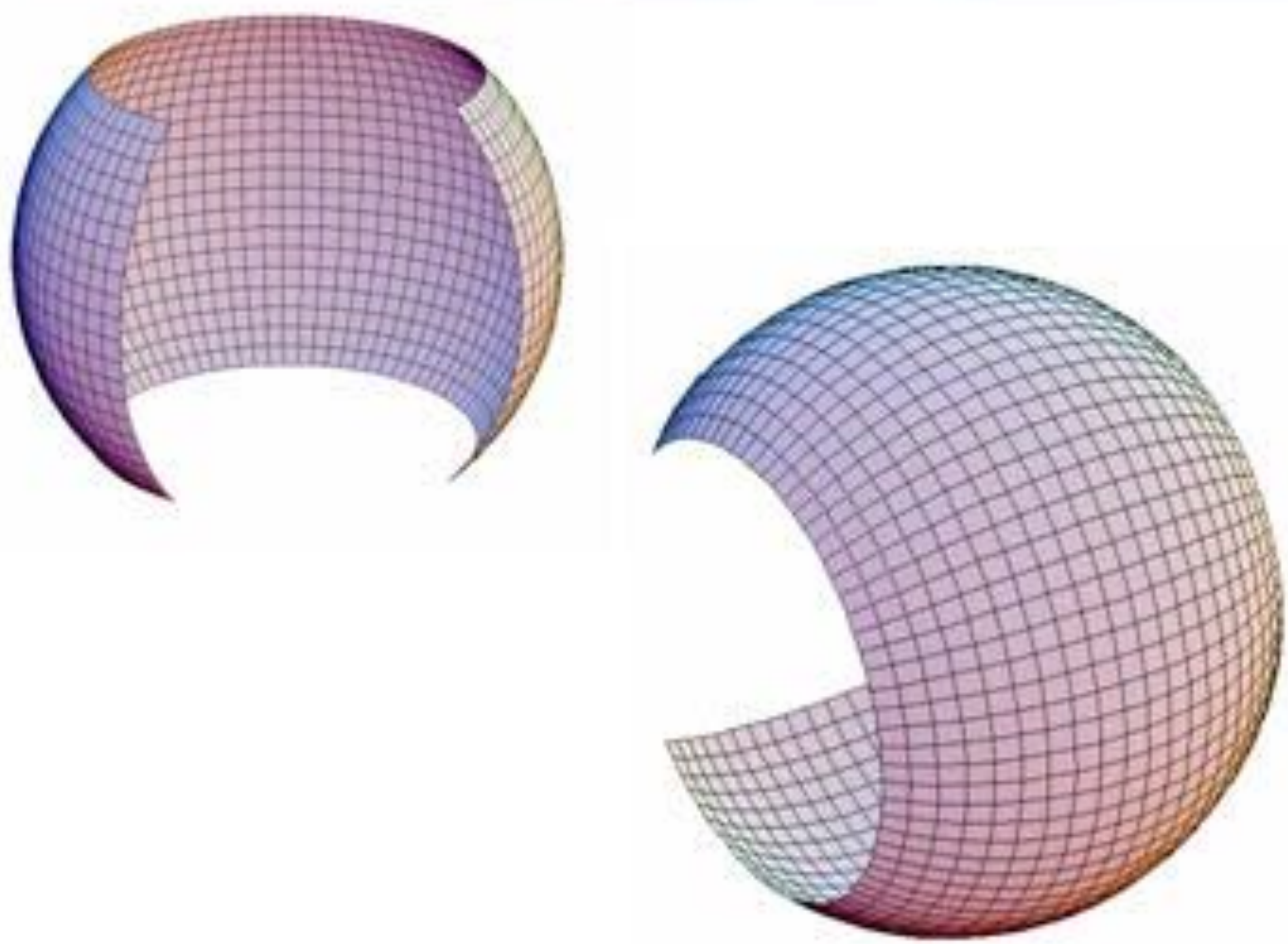
A spherical surface is covered by

- combination of two identical parts (patches).
- one seam.

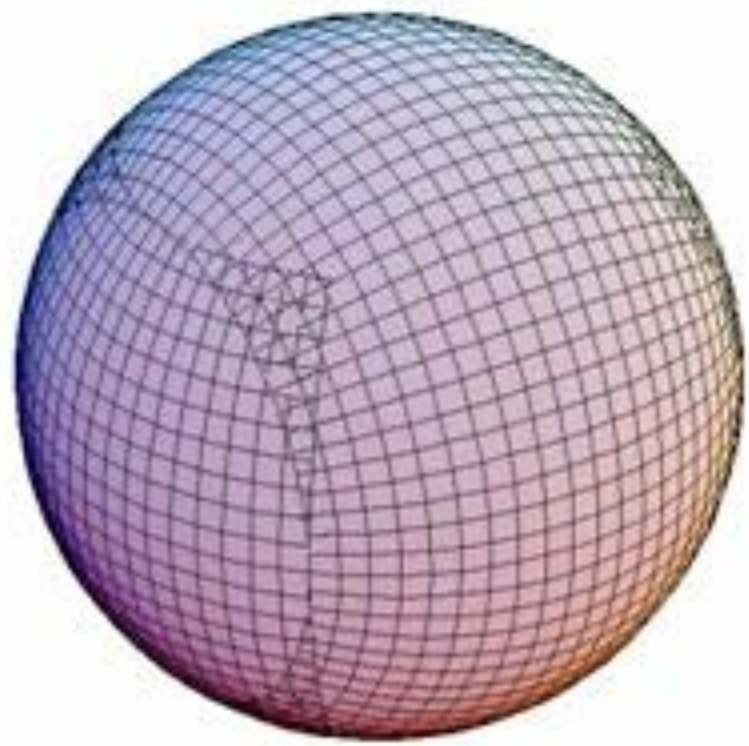
Combining Two Identical Sub-grids to Cover a Full Sphere



Combining Two Identical Sub-grids to Cover a Full Sphere

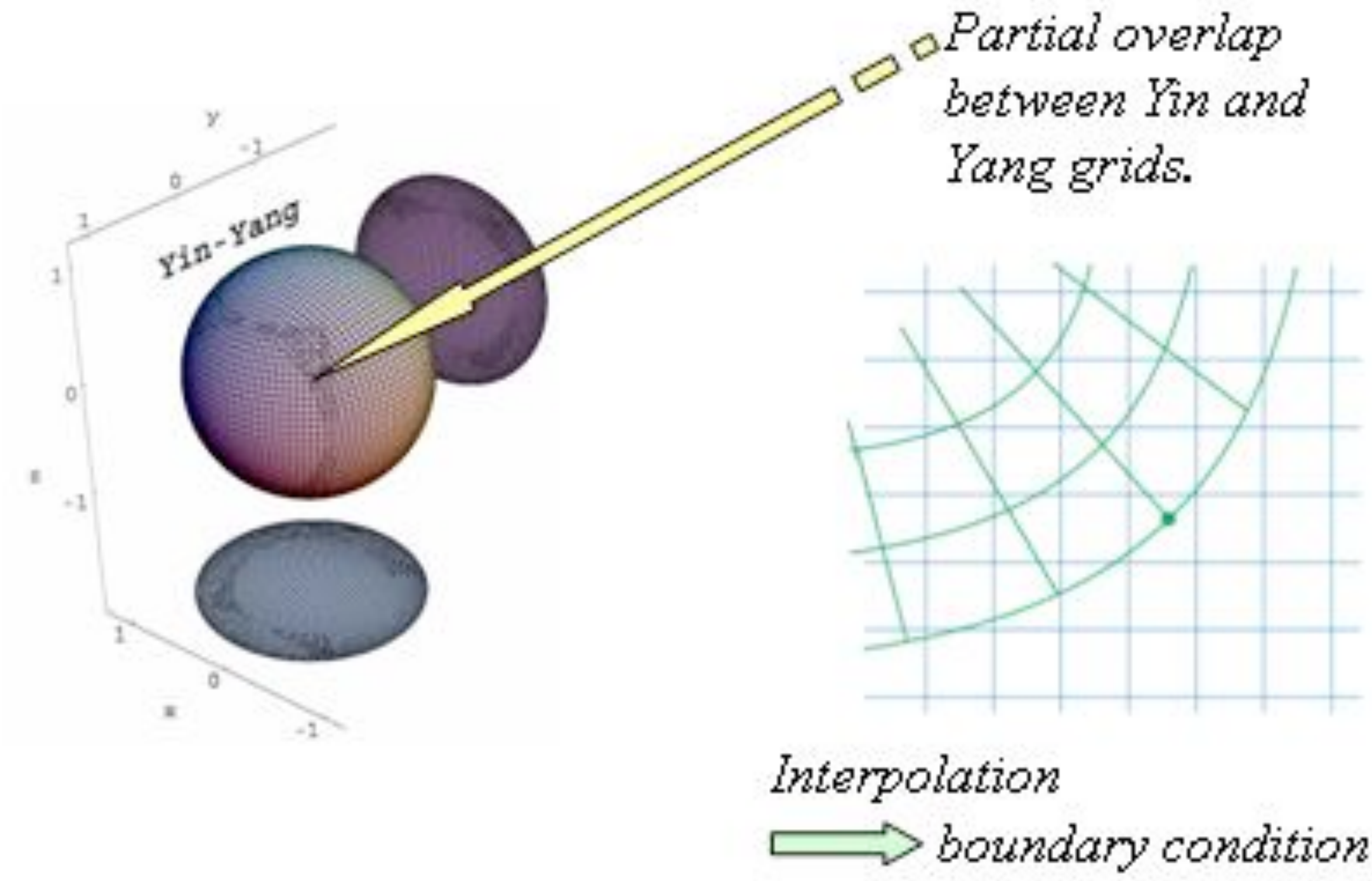


Combining Two Identical Sub-grids to Cover a Full Sphere

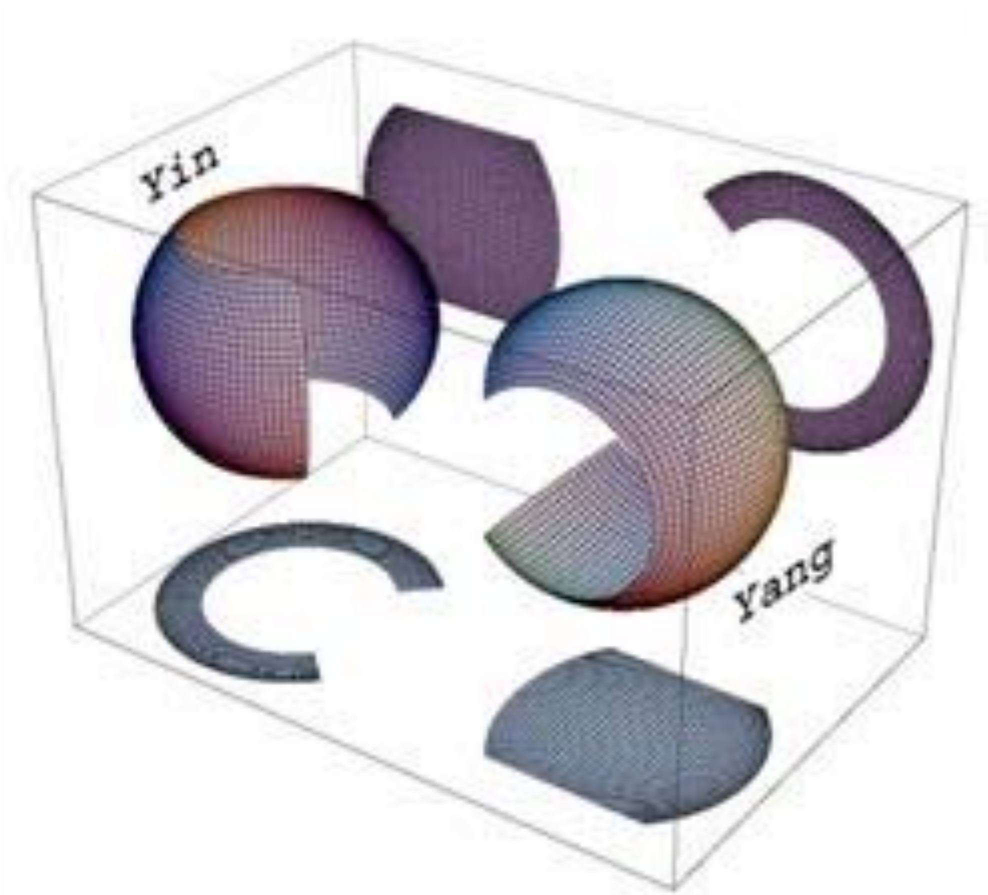


Yin-Yang 陰陽

Yin-Yang grid is an Overset Grid on the Spherical Geometry



Two Component Grids of Yin-Yang:
Yin grid & Yang grid



Suppose a point on the sphere with Yin (or n) coordinates given by

$$(x^n, y^n, z^n),$$

and with Yang (or e) coordinates given by

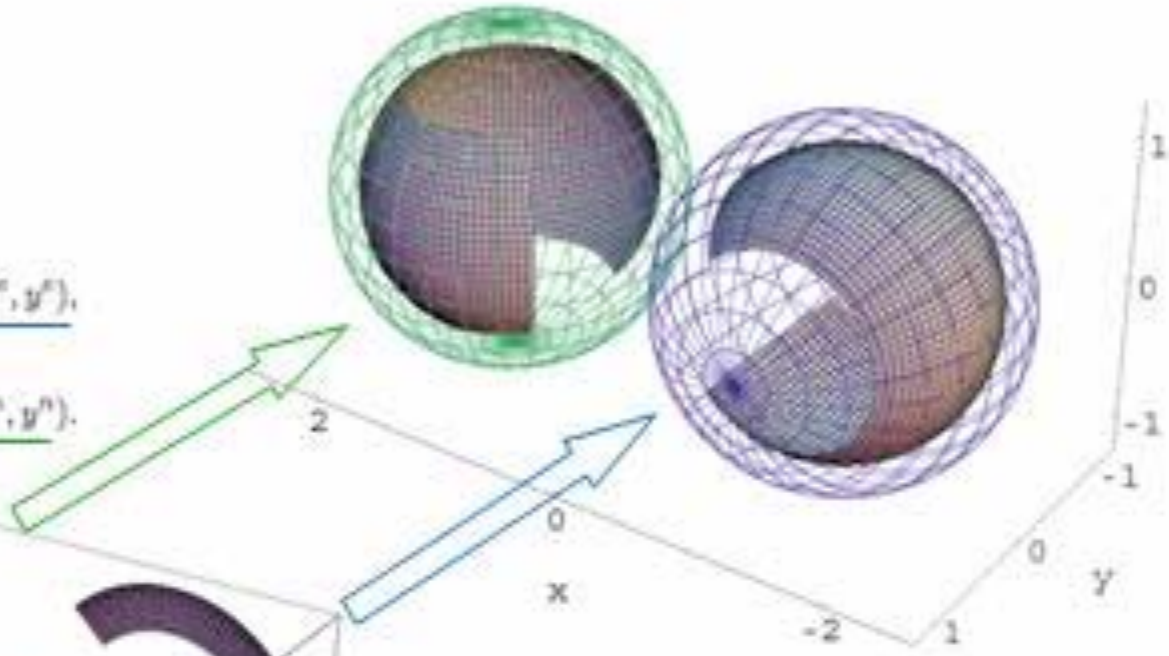
$$(x^e, y^e, z^e).$$

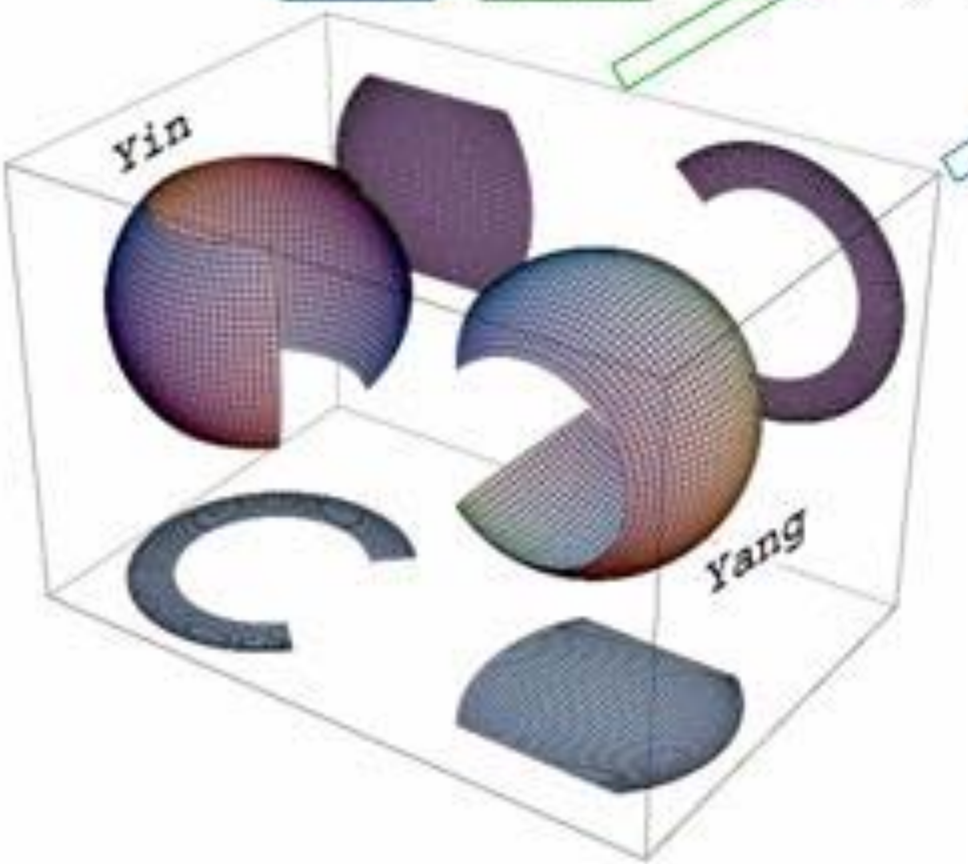
Their relation is given by

$$(x^n, y^n, z^n) = (-x^e, z^e, y^e),$$

or

$$(x^e, y^e, z^e) = (-x^n, z^n, y^n).$$

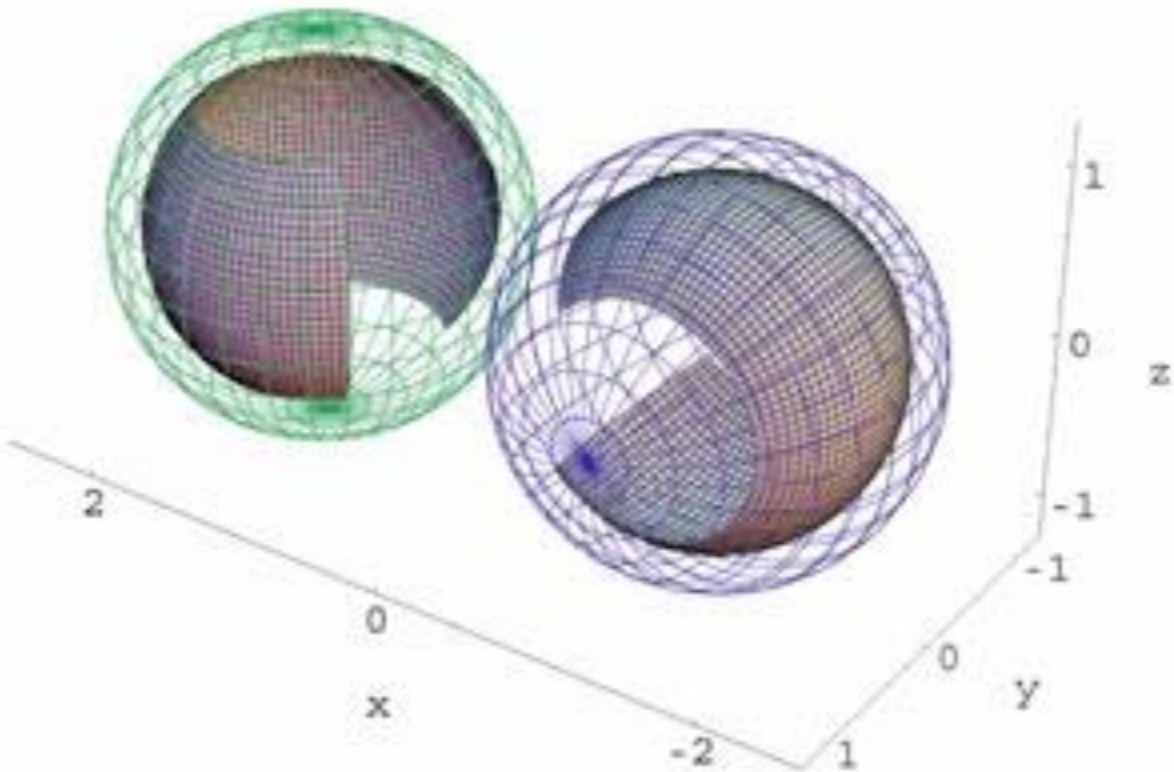




The coordinates relations are symmetric (complemental).

Concise Coding of Yin-Yang Grid:

- *Make one routine on the (partial) latitude-longitude grid.*
- *Recycle it for two times; one for Yin and another for Yang.*

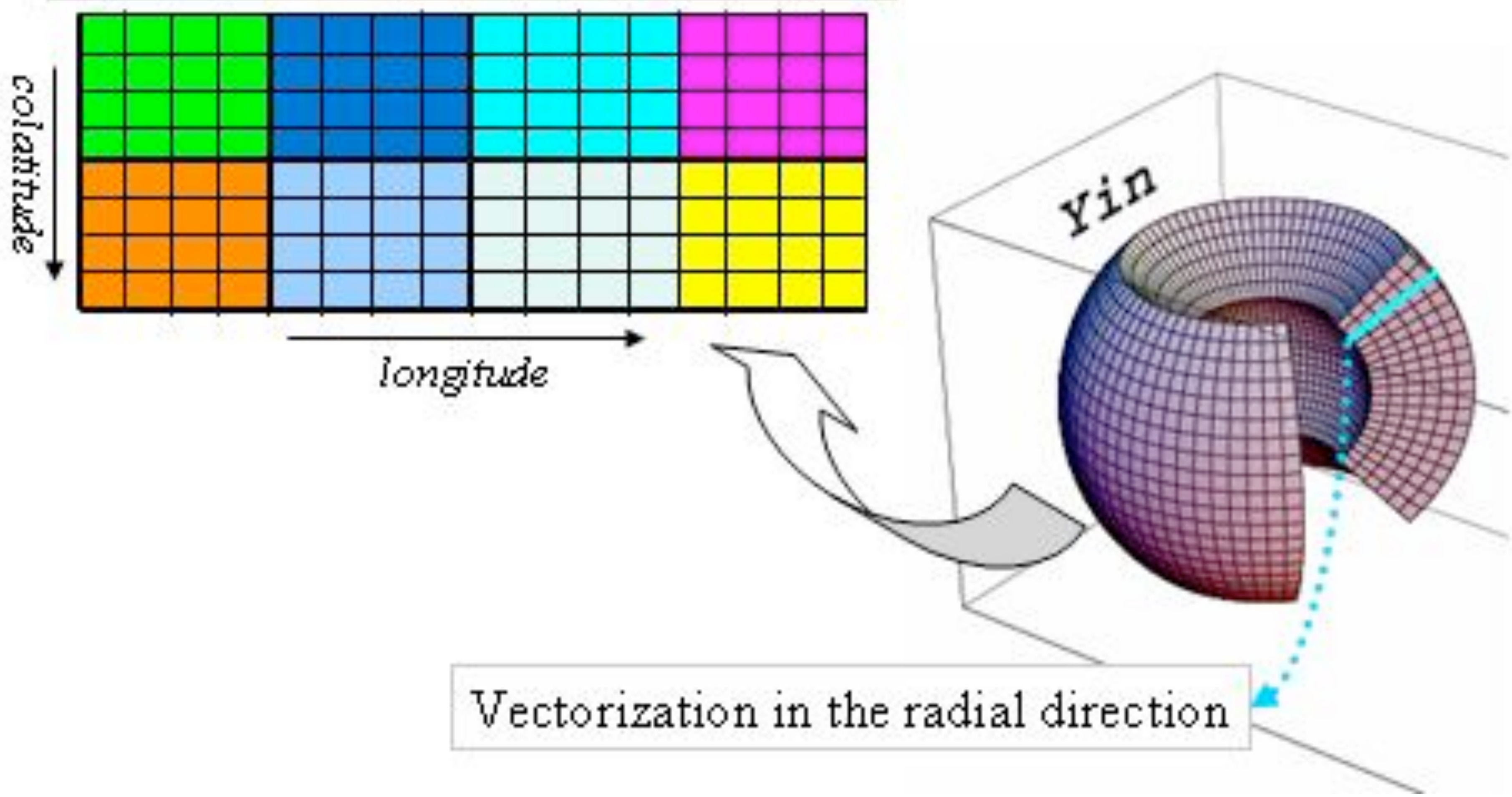


Routines for

- *MHD solver*
- *boundary conditions*
- *interpolations*

Vector-Parallel Computation on Yin-Yang Grid

2-dimensional domain decomposition in the horizontal computational space.



Vectorization in the radial direction

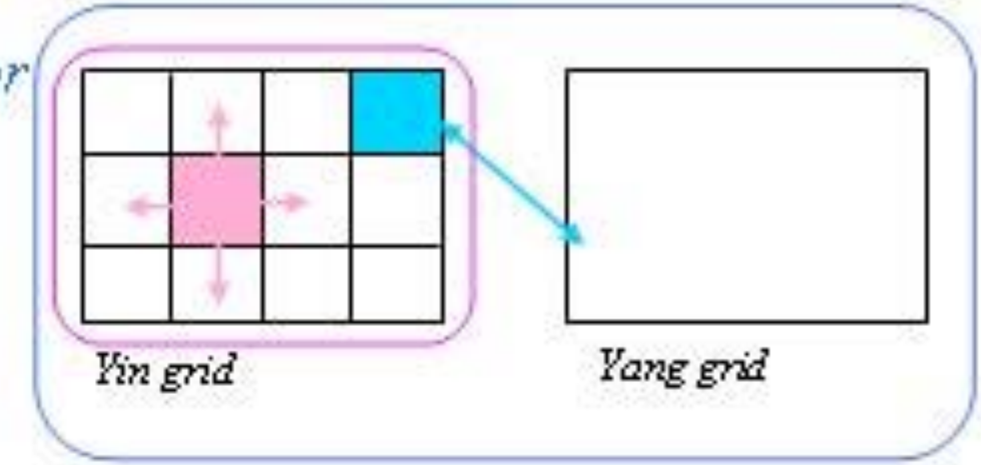
Parallelization Technique:
2 Levels of MPI Communicator

-*Overall world communicator*

-*Yin/Yang communicator*

  -*Yin's communicator*

  -*Yang's communicator*



Fortran90 code:

```
gRunner%world%communicator      ! communicator for entire Yin-Yang.
gRunner%panel%communicator      ! communicator for Yin or Yang.
gRunner%panel%rank%me           ! rank index in Yin or Yang.
gRunner%panel%rank%north        ! rank index of north neighbor
gRunner%panel%grid%nr           ! grid size in radial direction
gRunner%panel%grid%jstt         ! grid point of north border
```

Performance of the Yin-Yang Geodynamo Simulation code on the Earth Simulator

flat MPI

processors	grid points	Tflops	efficiency
3888	$511 \times 514 \times 1538 \times 2$	13.8	44%
3888	$255 \times 514 \times 1538 \times 2$	12.1	39%
2560	$511 \times 514 \times 1538 \times 2$	10.3	50%
2560	$255 \times 514 \times 1538 \times 2$	9.17	45%
1200	$255 \times 514 \times 1538 \times 2$	5.40	56%
4096	511 x 514 x 1538 x 2	15.2	46.3%

Epilogue:

Ying-Yang has been found to be not too efficient in variable viscosity, high Rayleigh number convection. The parallel version does not run that fast and Charley's code is much faster, when cast in spherical-internal boundary.