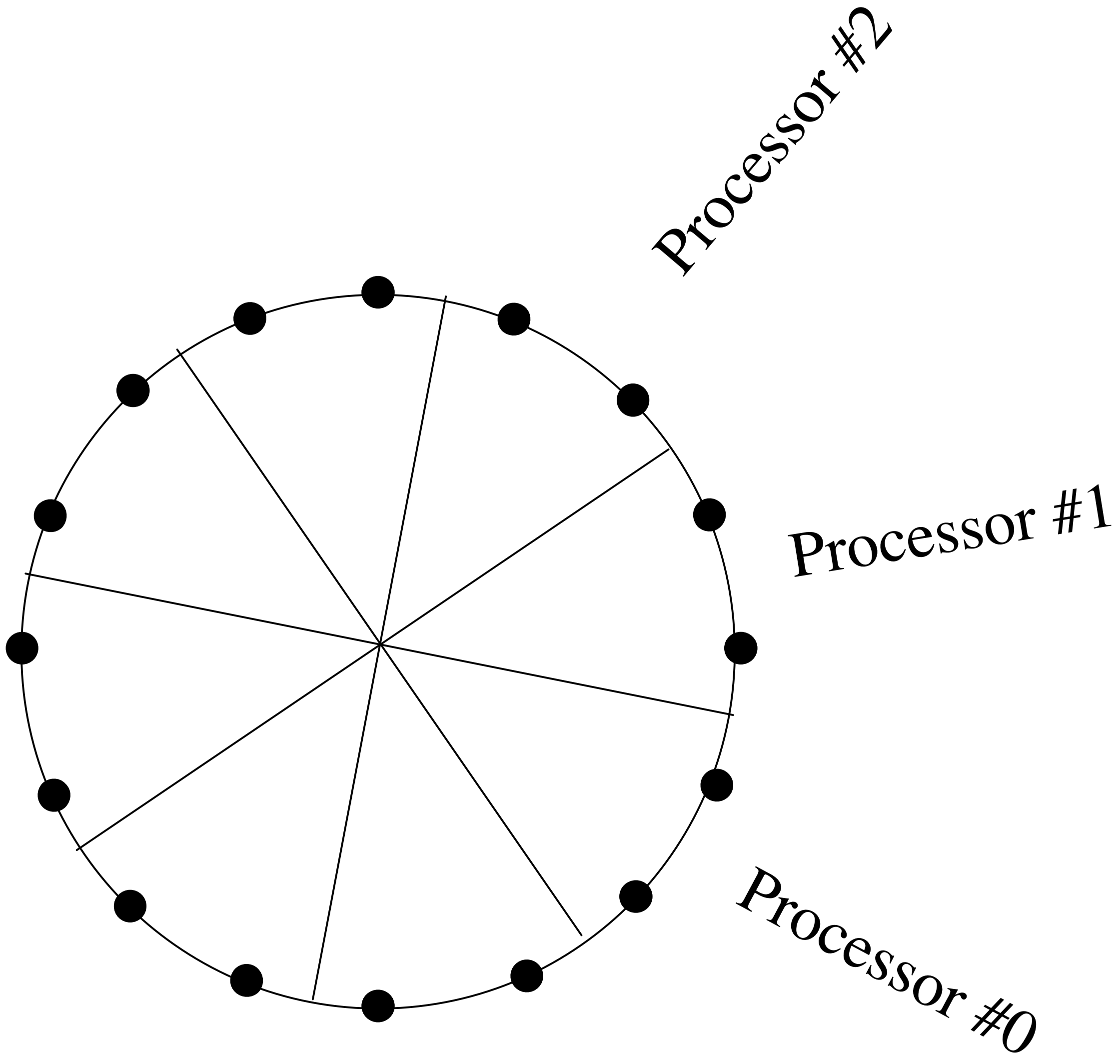
Numerical Methods for Geodynamo Simulation

Akira Kageyama
Earth Simulator Center, JAMSTEC, Japan

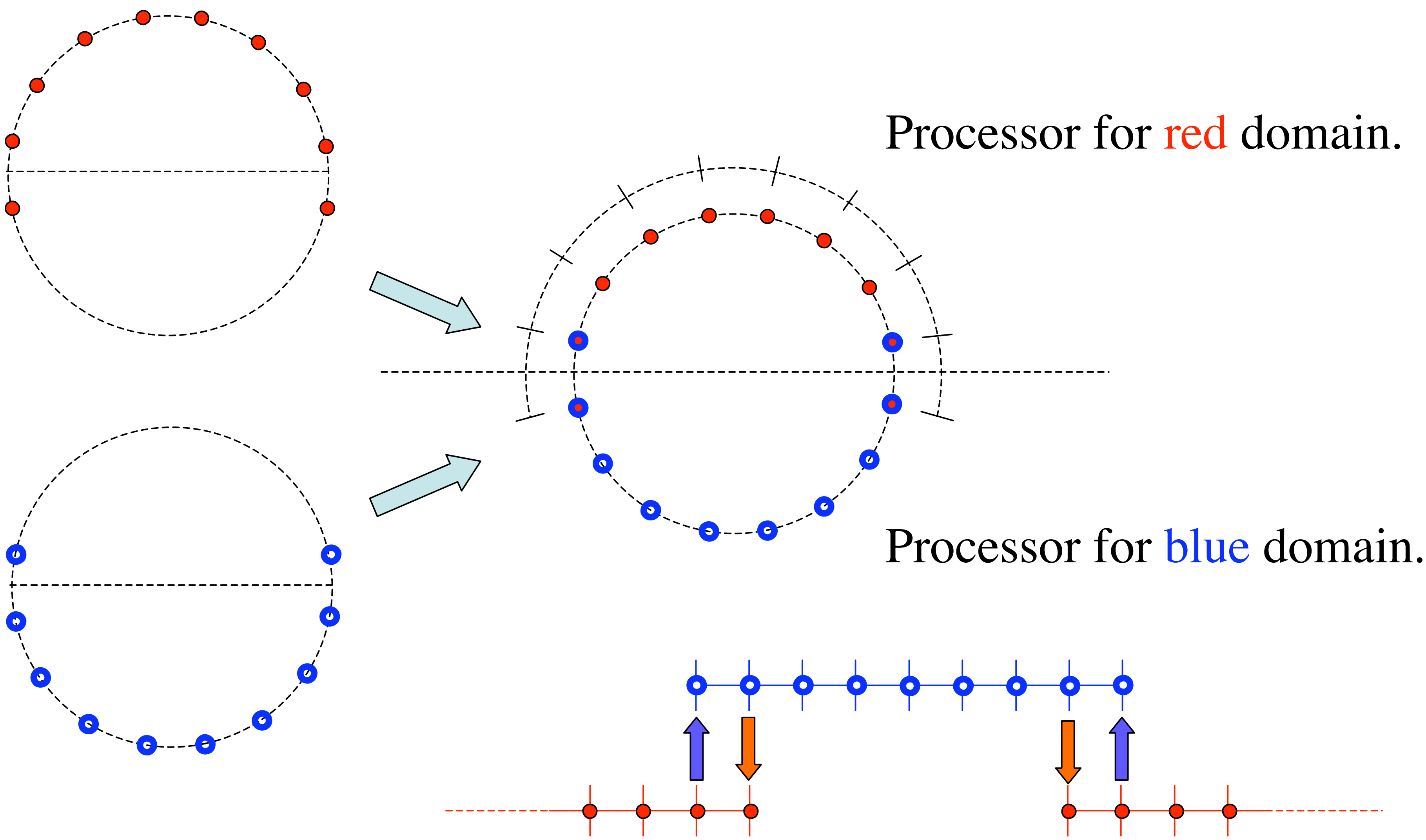
Part 3

2007.08 Les Houches Summer School "Dynamos"

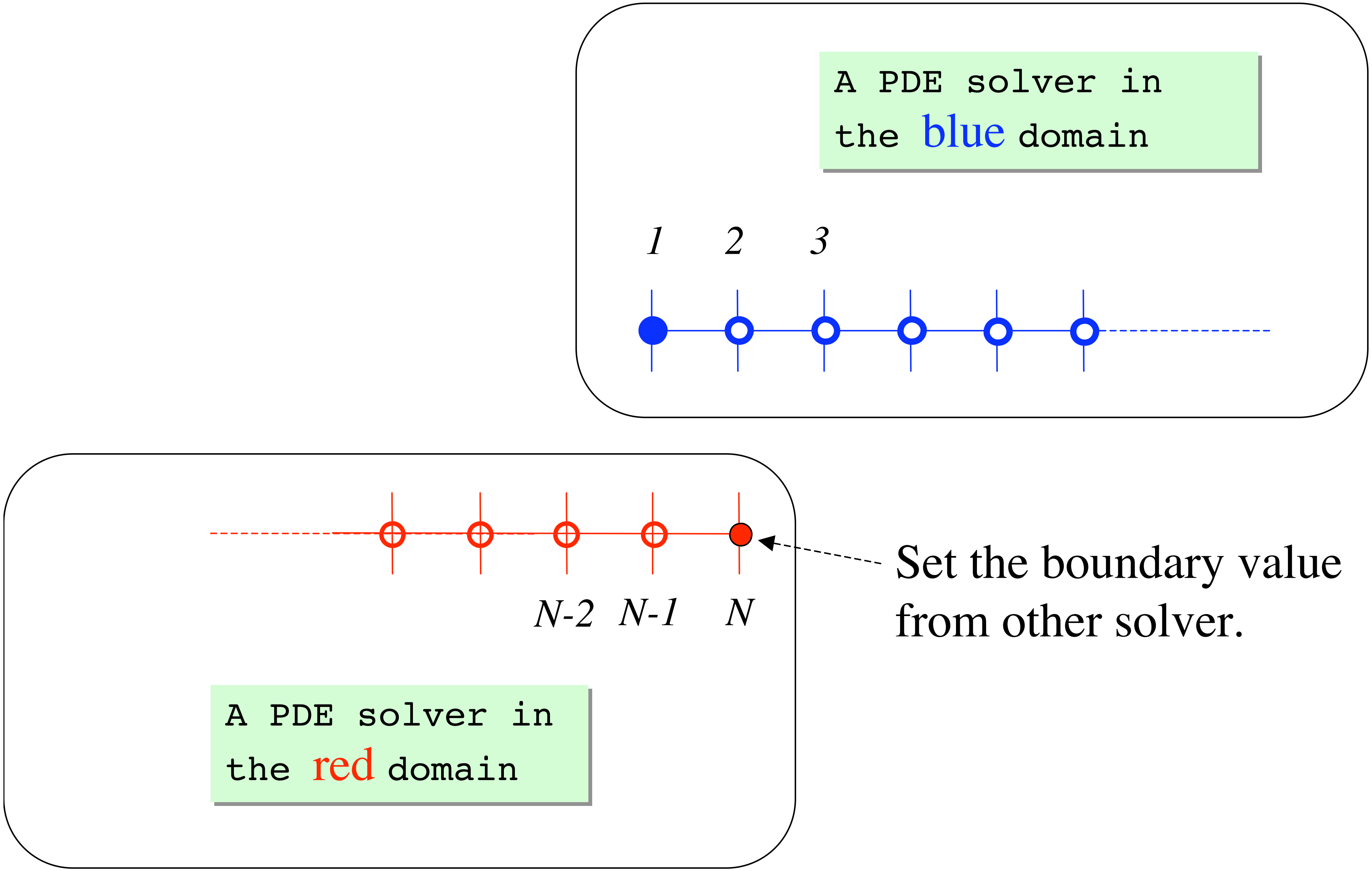
Parallel processing by domain decomposition



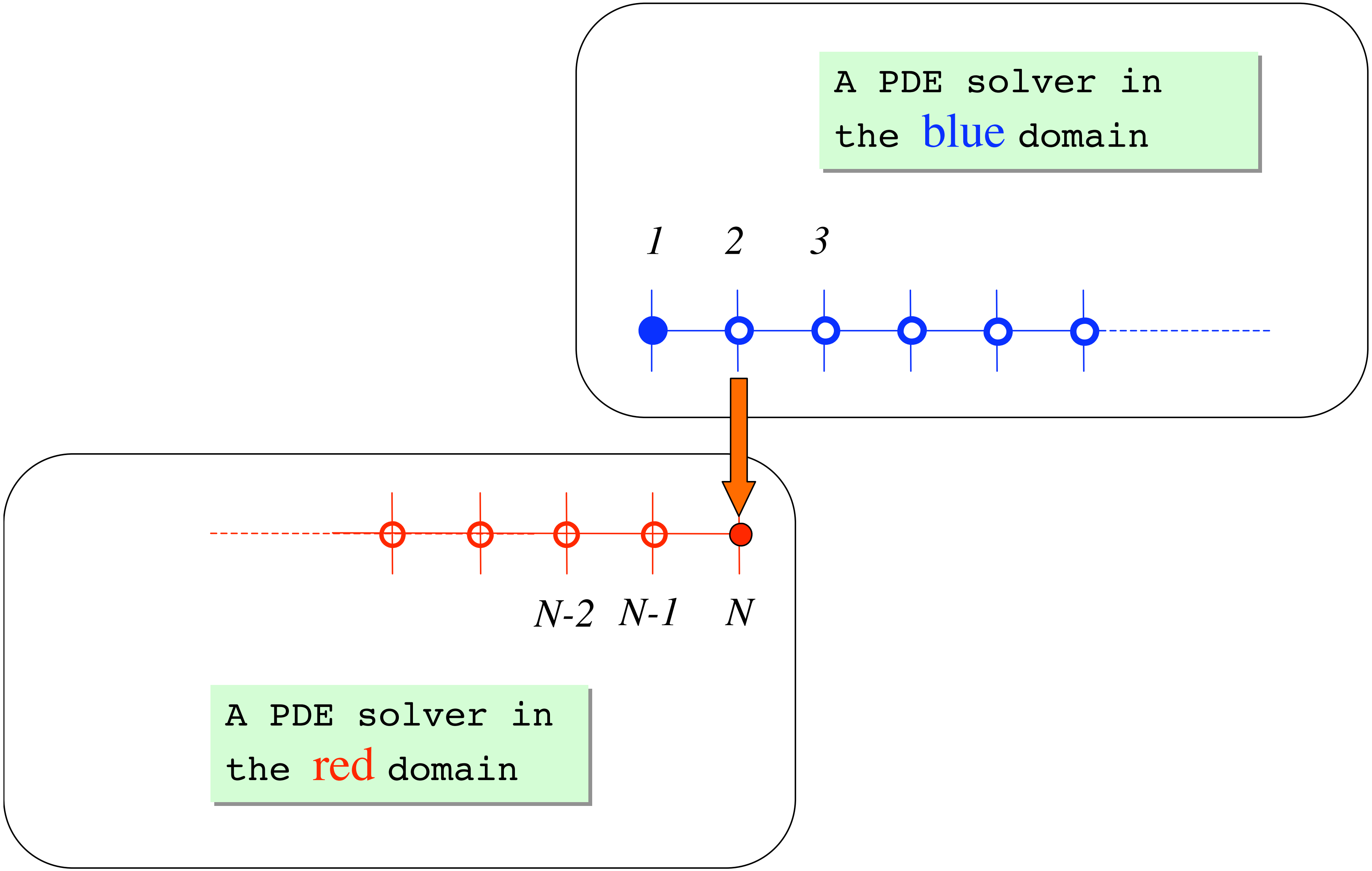
Parallel computation by 2 processors



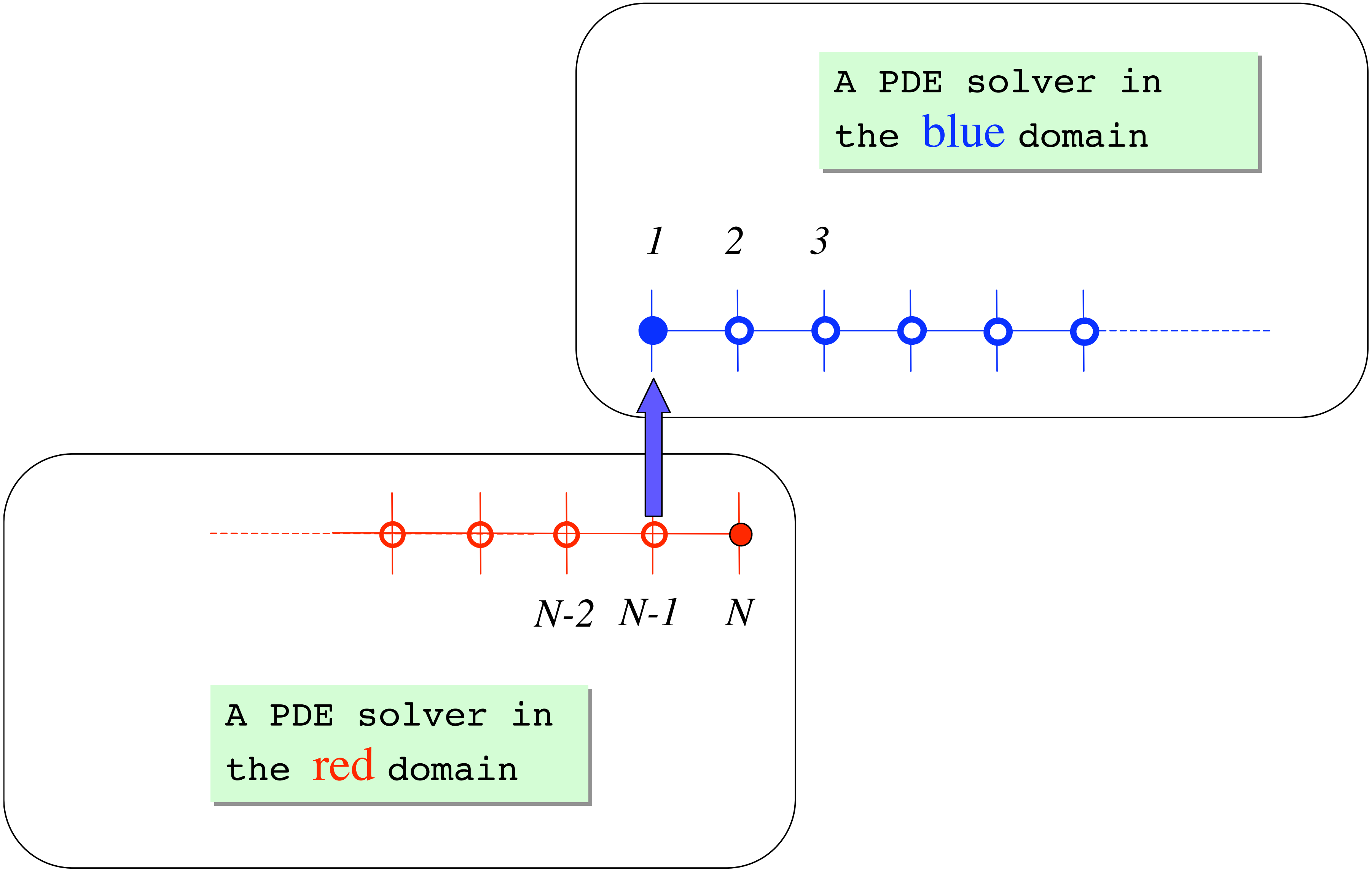
Inter process communication



Inter process communication



Inter process communication



In the code

In the red processor:

```
do i = 2 , nx-1
  diffusion_equation(i) = dx2*(psi(i+1)-2*psi(i)+psi(i-1))
end do
```

Same computation, same code.

In the blue processor:

```
do i = 2 , nx-1
  diffusion_equation(i) = dx2*(psi(i+1)-2*psi(i)+psi(i-1))
end do
```

In the code

In the red processor:

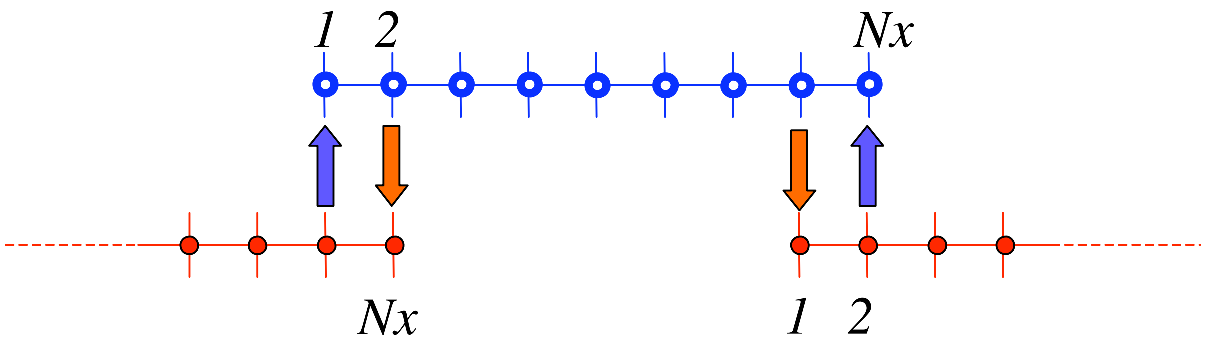
```
do i = 2 , nx-1
  diffusion_equation(i) = dx2*(psi(i+1)-2*psi(i)+psi(i-1))
end do
```

Same computation, same code.

In the blue processor:

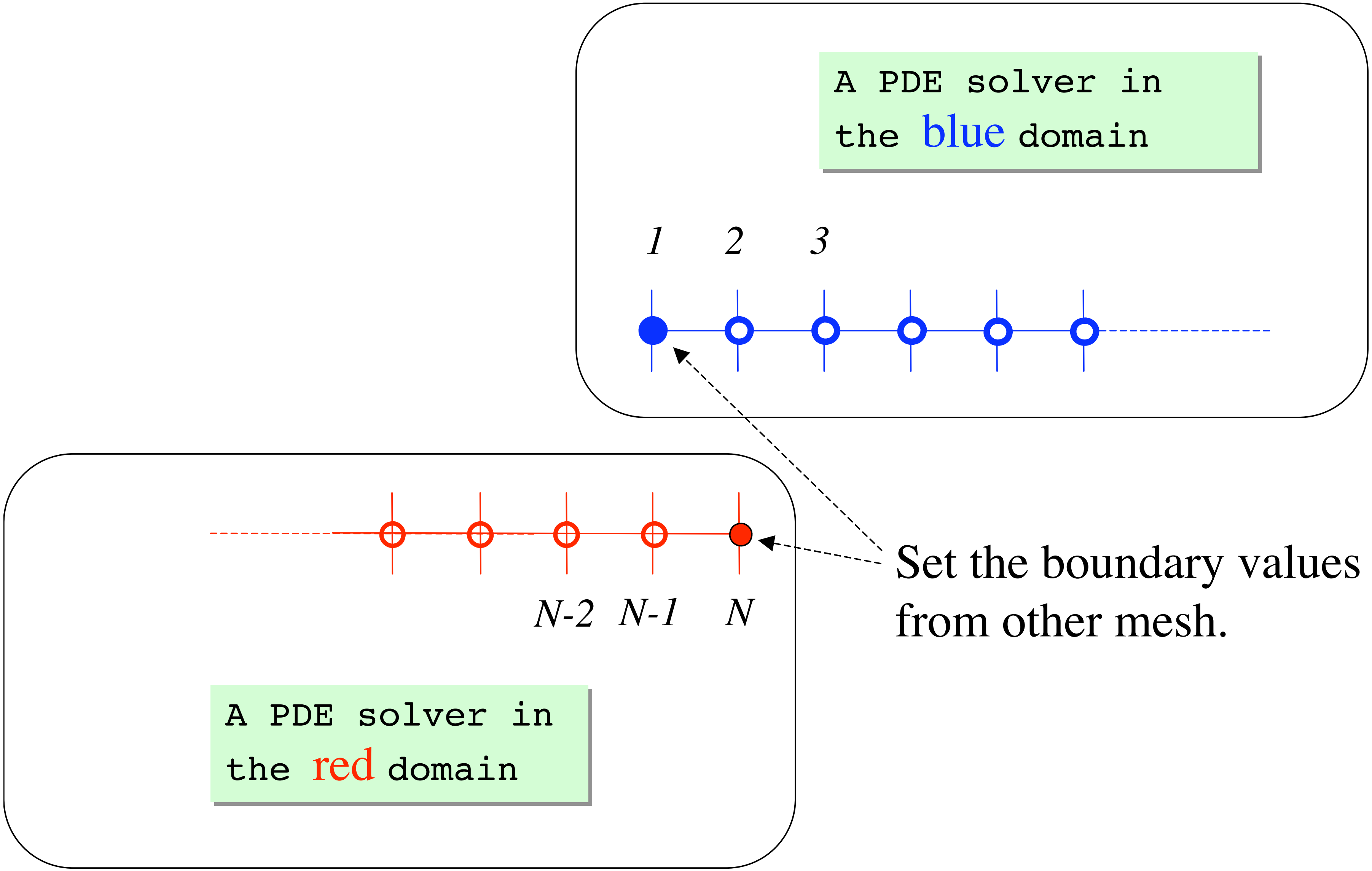
```
do i = 2 , nx-1
  diffusion_equation(i) = dx2*(psi(i+1)-2*psi(i)+psi(i-1))
end do
```

“Boundary condition”



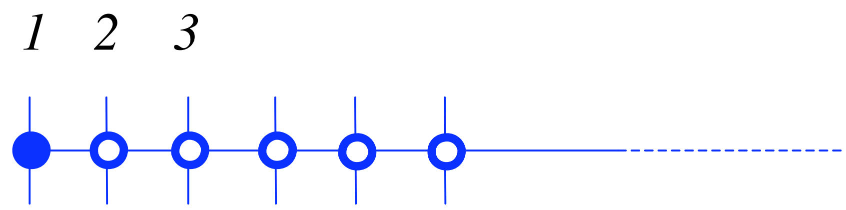
Inter-process communication (by MPI, etc.)

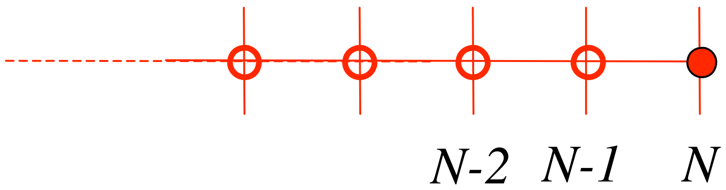
Two independent PDE solvers



Two independent PDE solvers: Grid size can be different

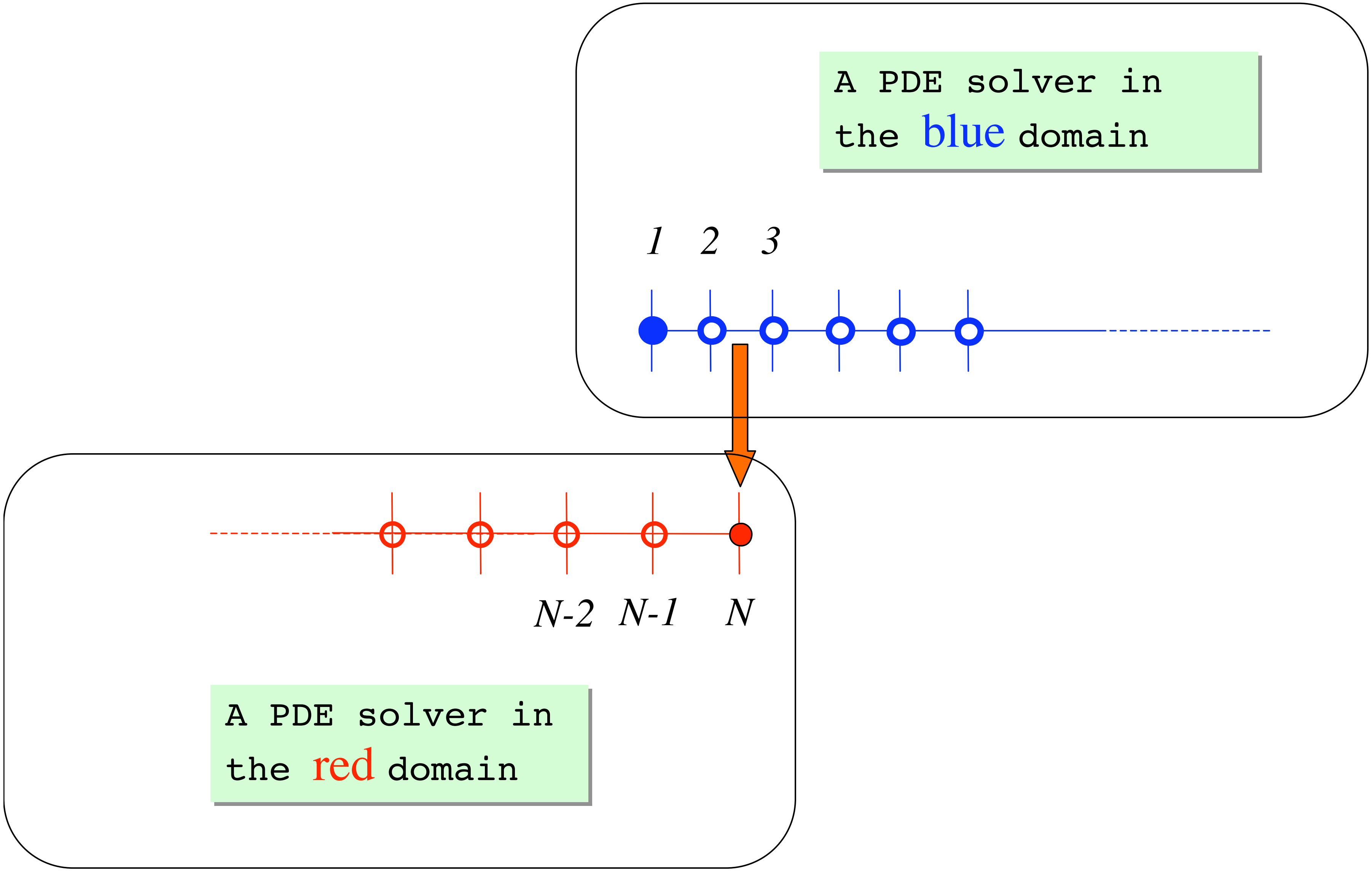
A PDE solver in the blue domain



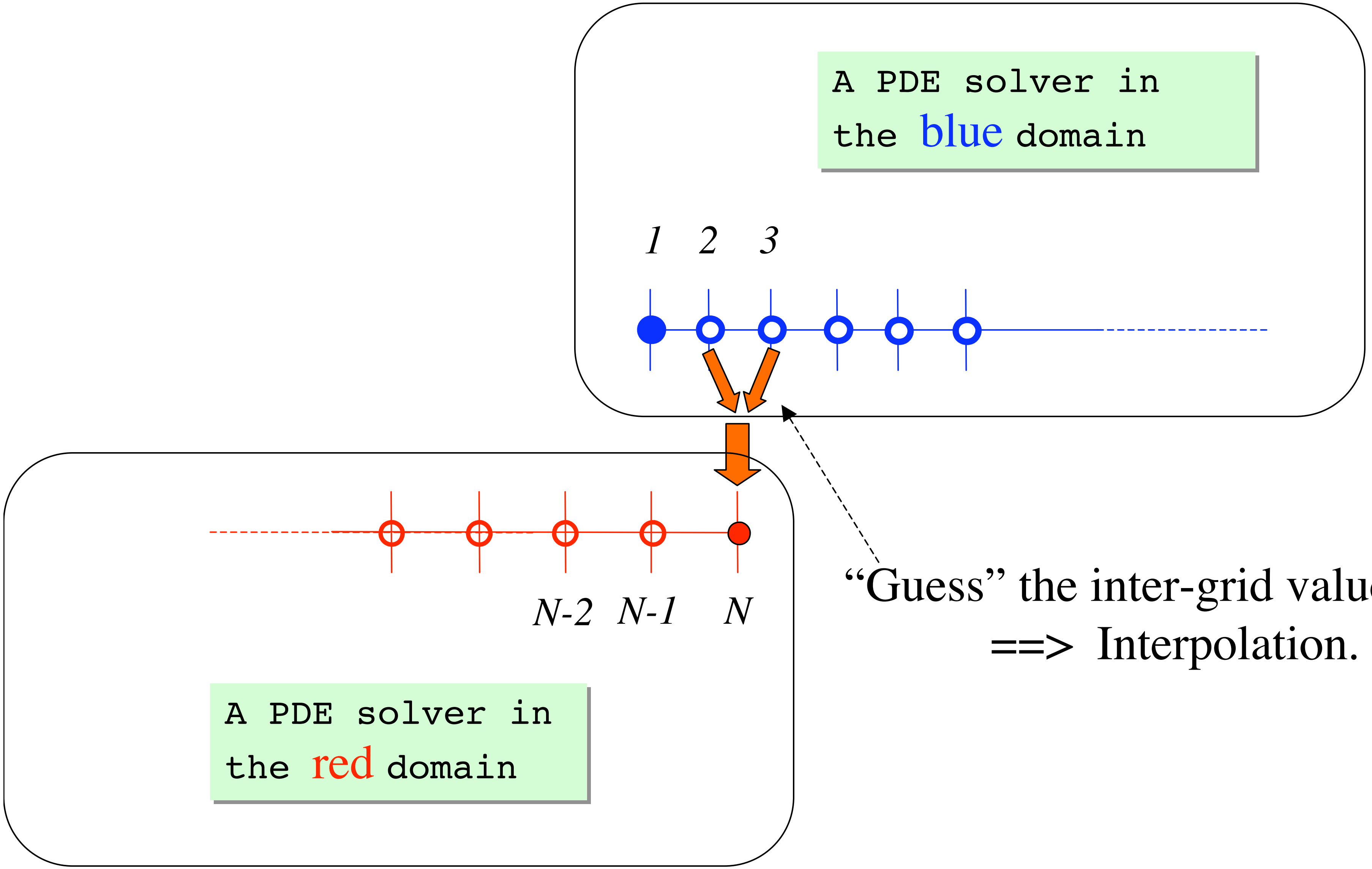


A PDE solver in the red domain

Two independent PDE solvers: Grid size can be different



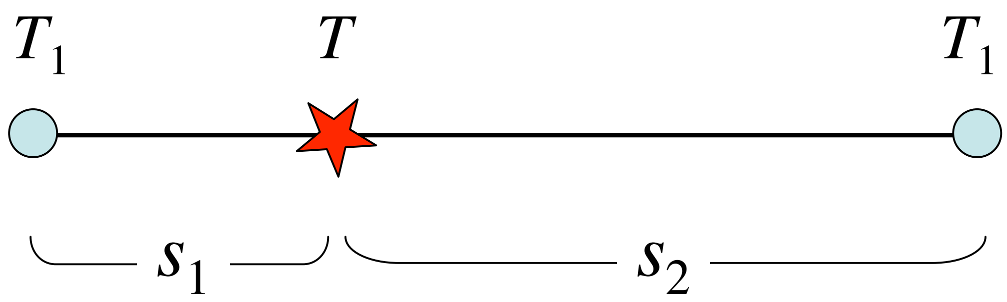
No problem



"Guess" the inter-grid value.

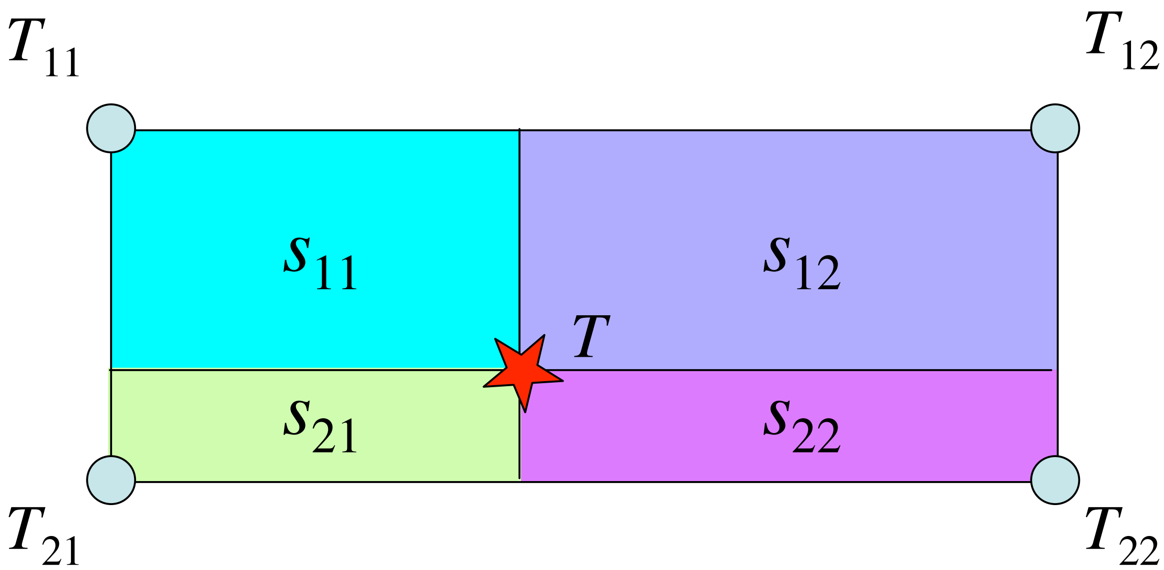
==> Interpolation.

Linear interpolation in 1-D



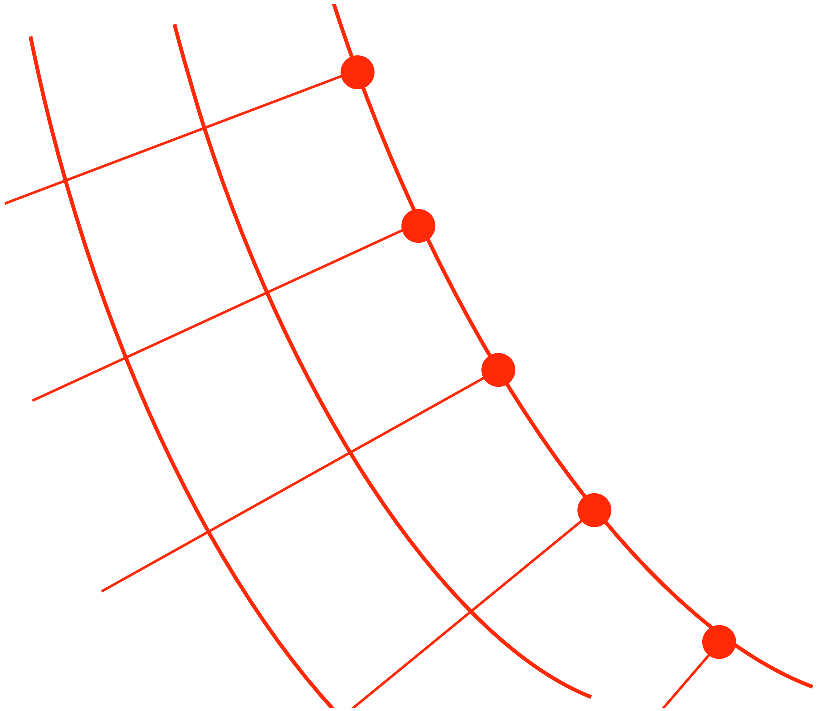
$$T = (s_2 T_1 + s_1 T_2) \,/\, (s_1 + s_2)$$

Bi-linear interpolation in 2-D

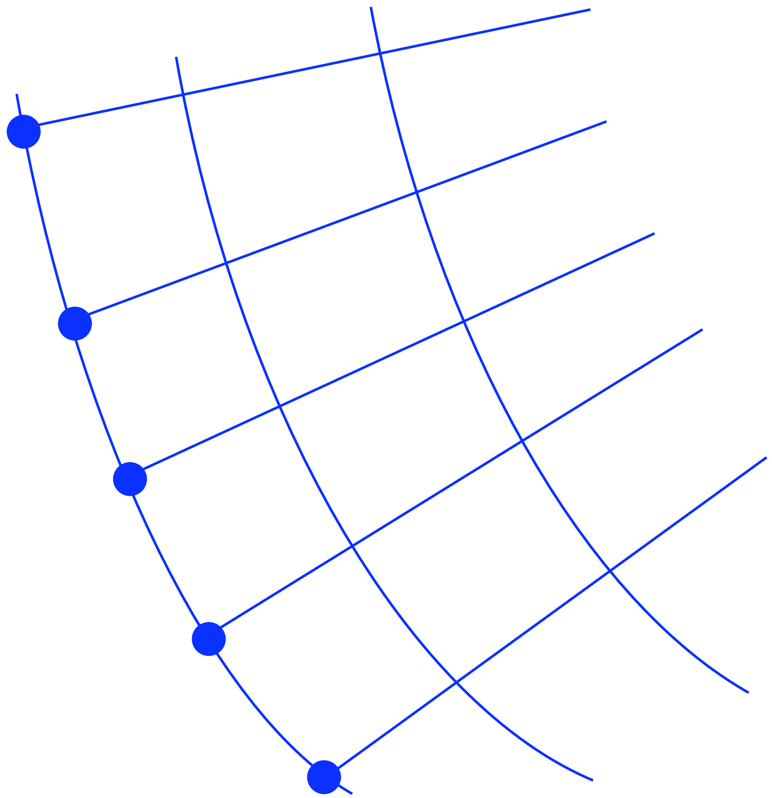


$$T = \frac{(s_{22}\,T_{11} + s_{21}\,T_{12} + s_{12}\,T_{21} + s_{11}\,T_{22})}{(s_{11} + s_{12} + s_{21} + s_{22})}$$

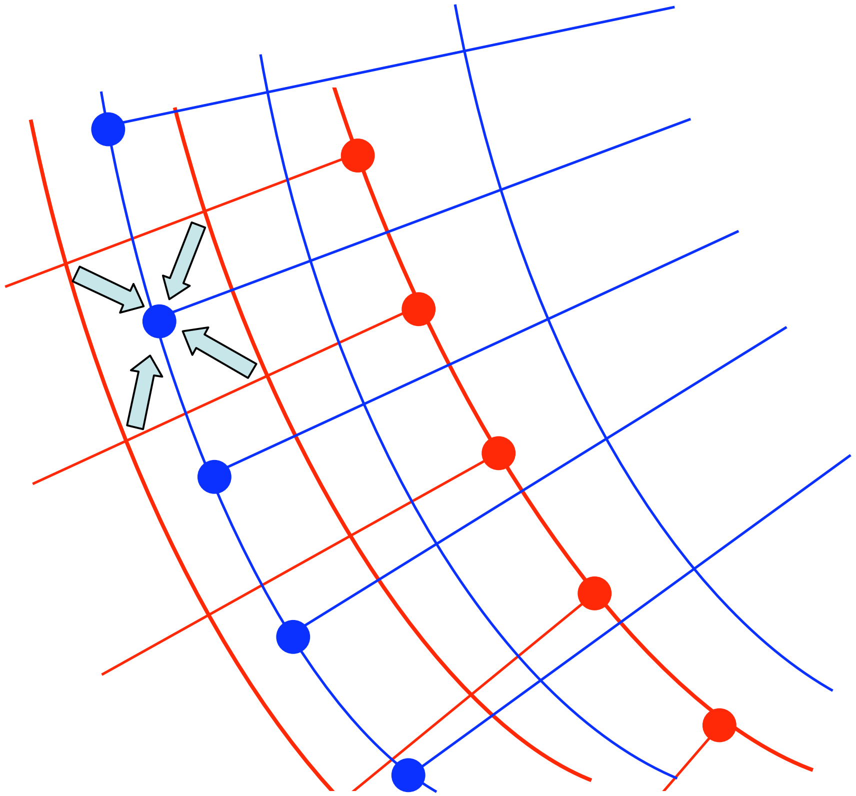
Two independent grids



Two independent grids



Overset grid (Chimera grid) method



- “Divide and conquer” approach

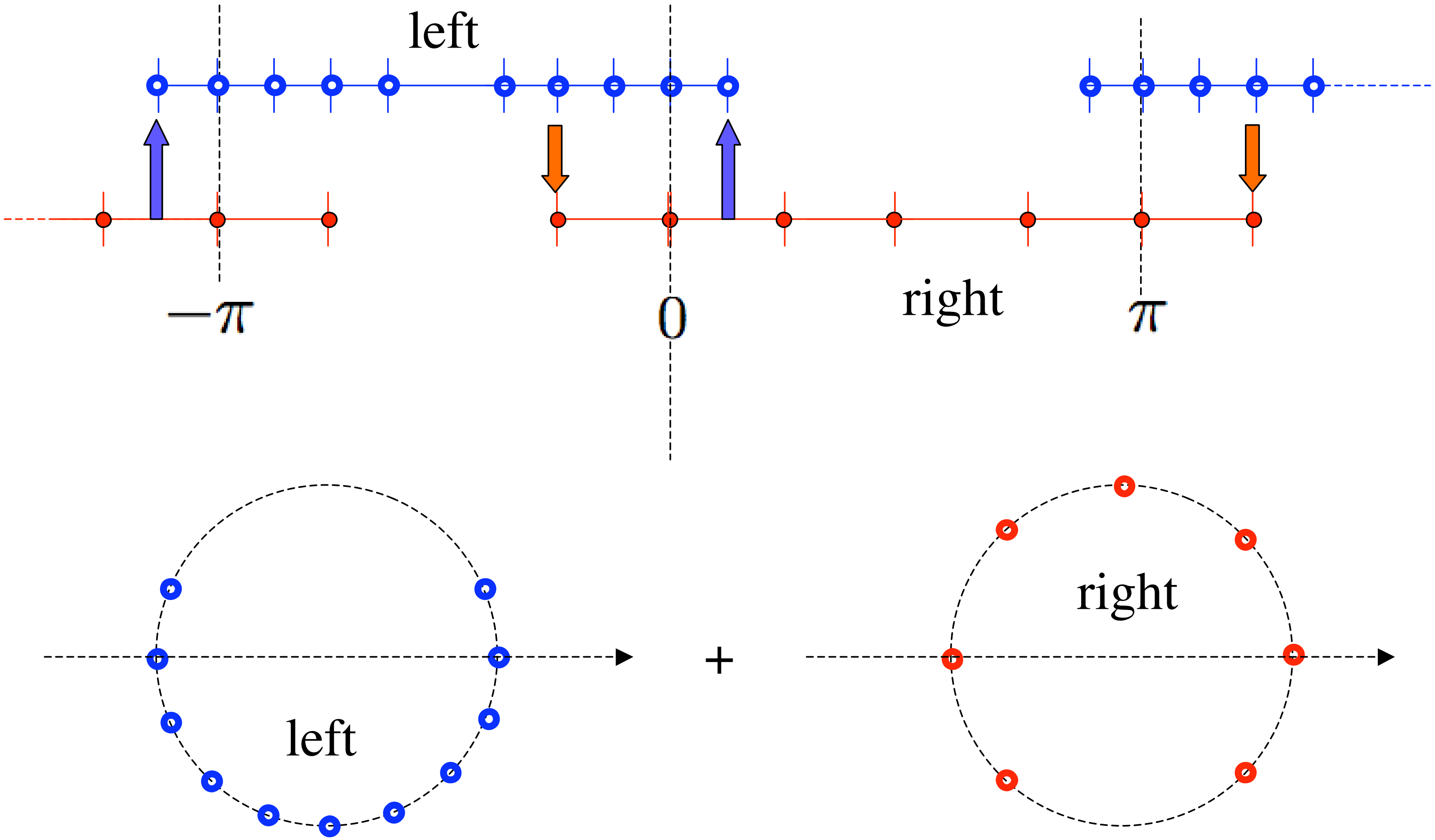
- Partially overlapped meshes.

- Setting boundary values by <u>mutual</u> interpolations.

- Essentially parallel computation.

Chesshire, G., and W. D. Henshaw (1990),
Composite overlapping meshes for the solution of partial differential equations,
J. Comput. Phys., 90, 1 − 64,

Overset grid method example: 1-D Burgers' equation



Let’s run the code

In sourcecode_tar.gz,
- src/SampleChimera

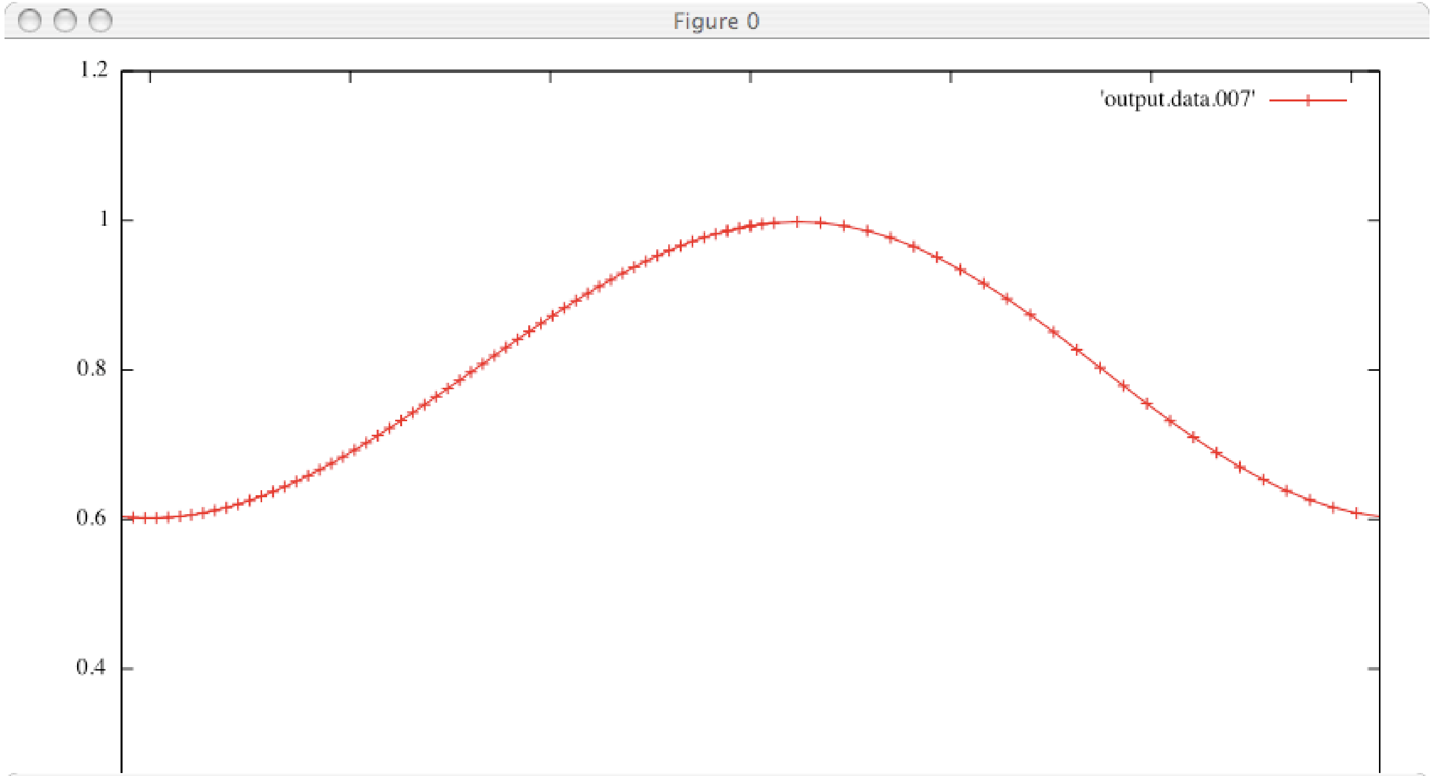


Figure 0
1.2
1
0.8
0.6
0.4
'output.data.007'

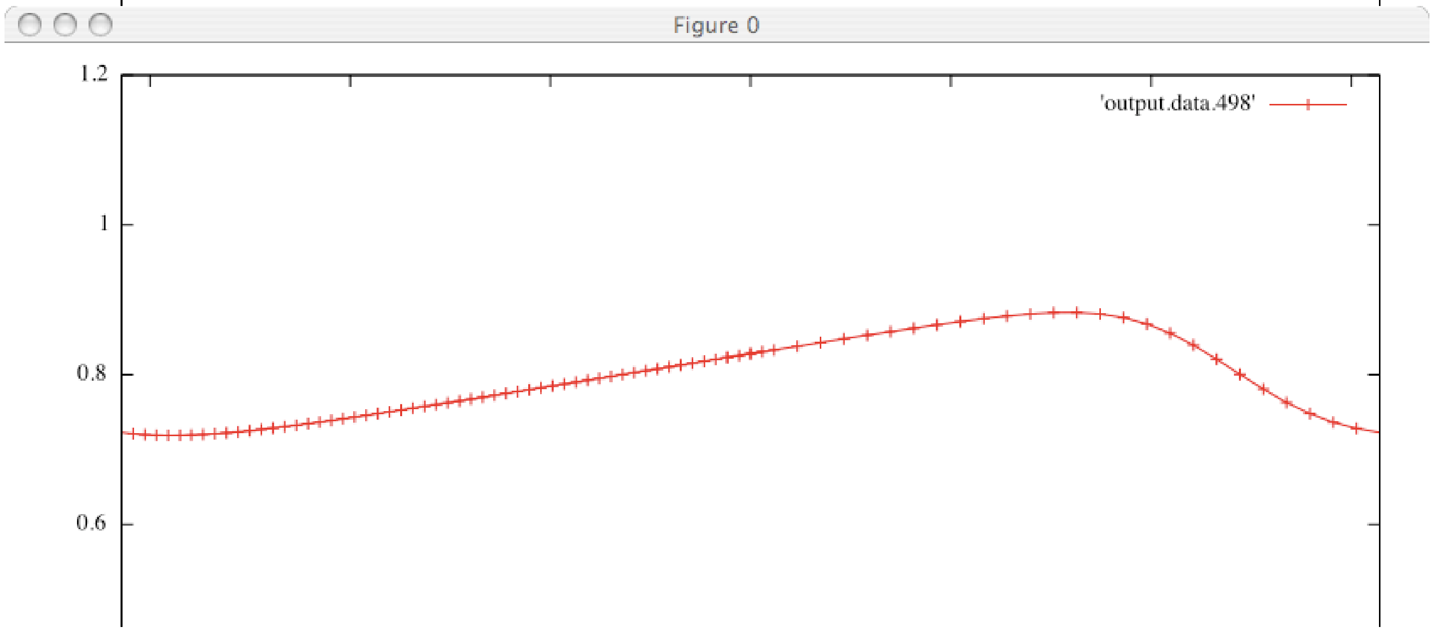


Figure 0
1.2
1
0.8
0.6
'output.data.498'

Coordinate singularity causes two <u>different</u> problems

1. On the poles:

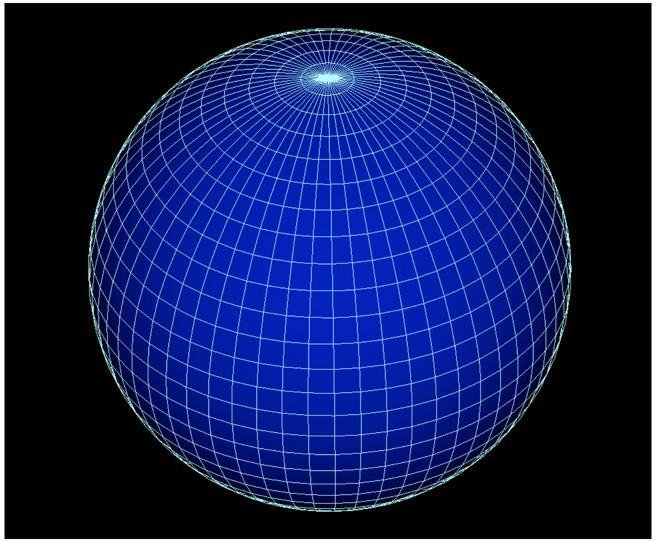
    - Need's special cares;
      e.g., L'Hospital's theorem.

    - Not serious.

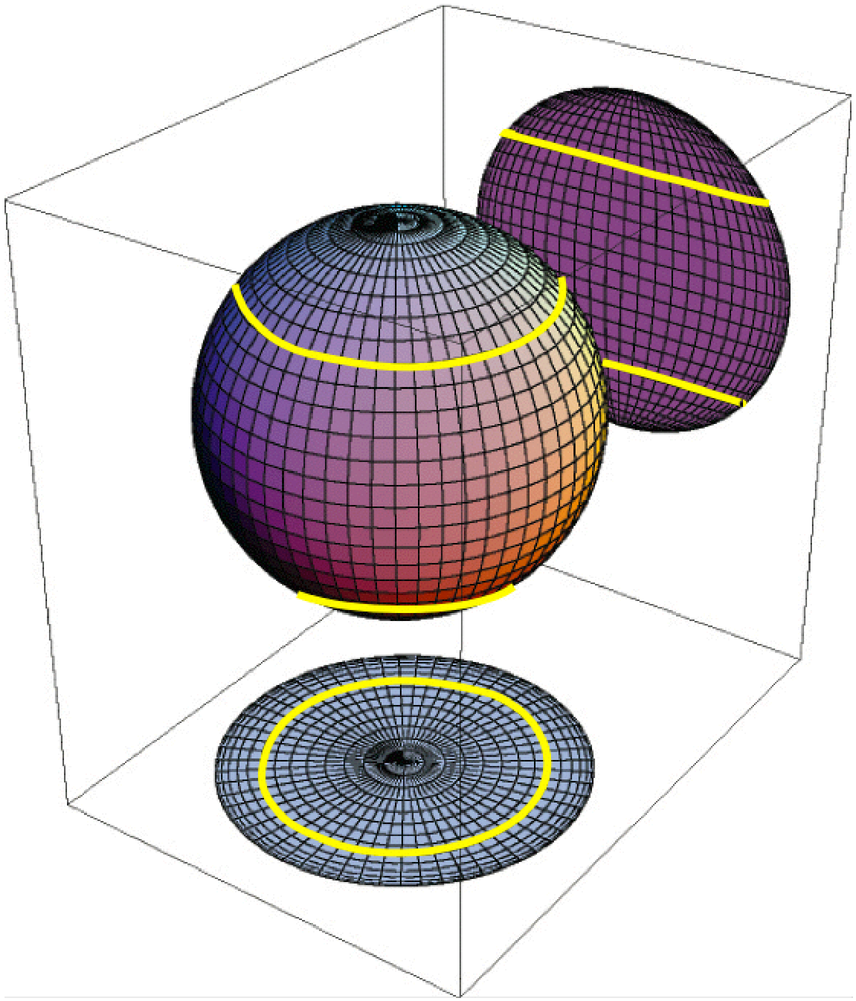
2. Near the poles:

    - Grid convergence.

    - Serious; waste of CPU time.



Grid convergence = inefficiency

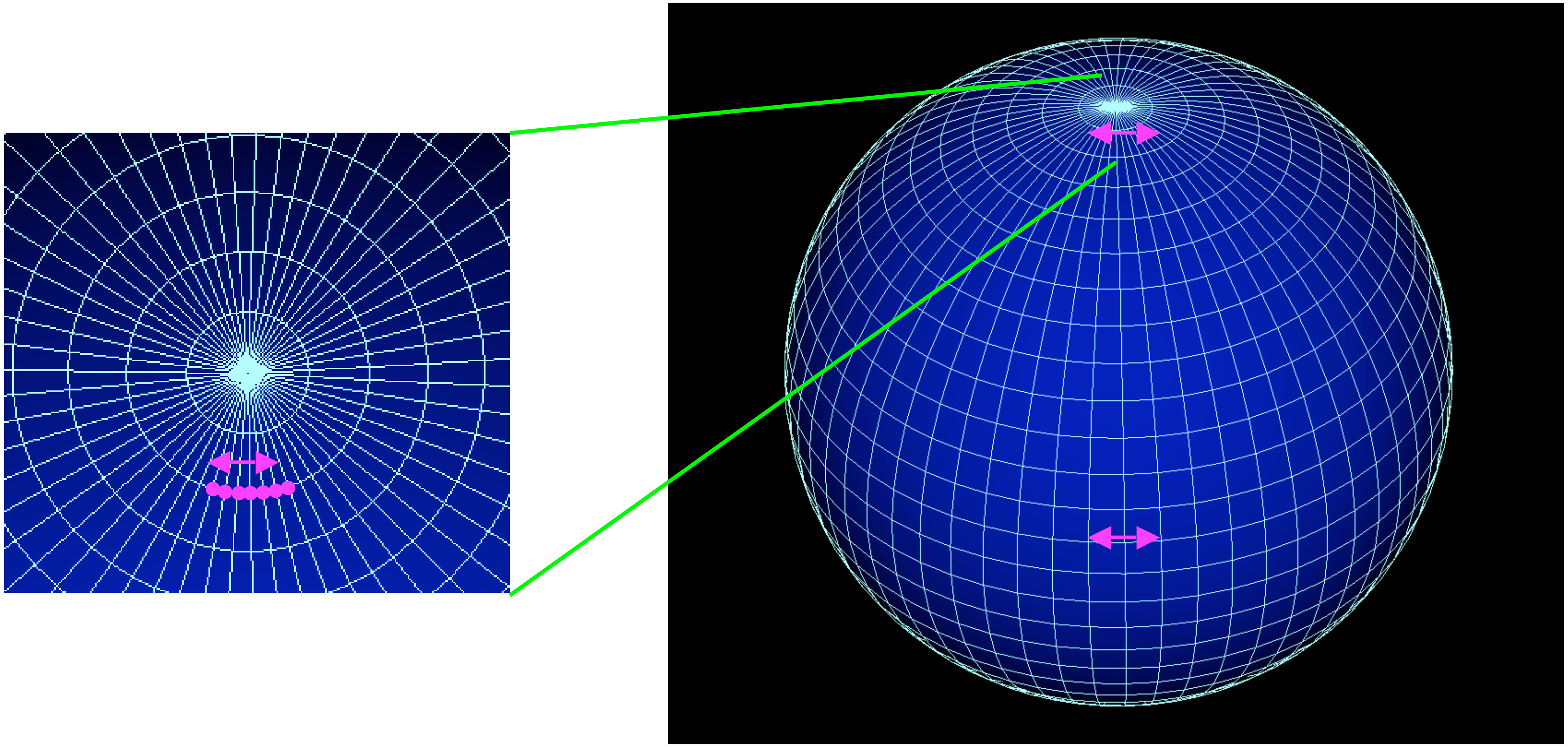


84% of grid points are located in high-latitude part (>45^{o} N and S).

Low latitude part (between 45^{o} N and S) is covered by only 16%.

Grid Spacing Problem in Lat-Lon Grid

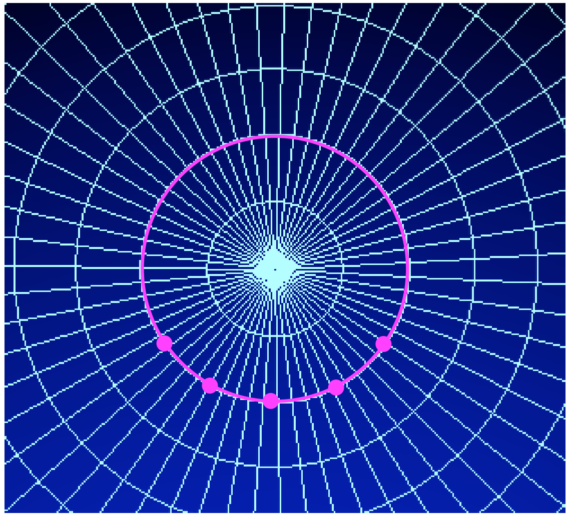
Severe CFL condition (short time step) near the poles.

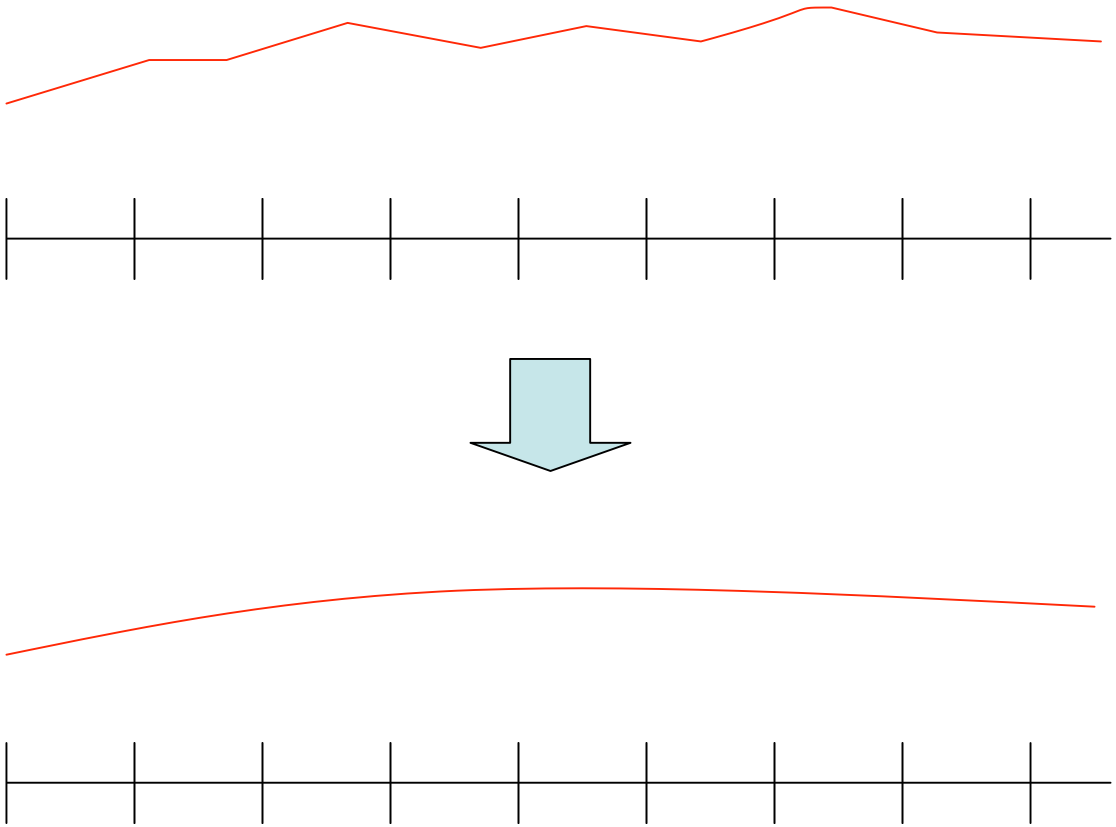


Spherical Filter

Retain the grids, but drop useless high wave number modes.

→ Filtering





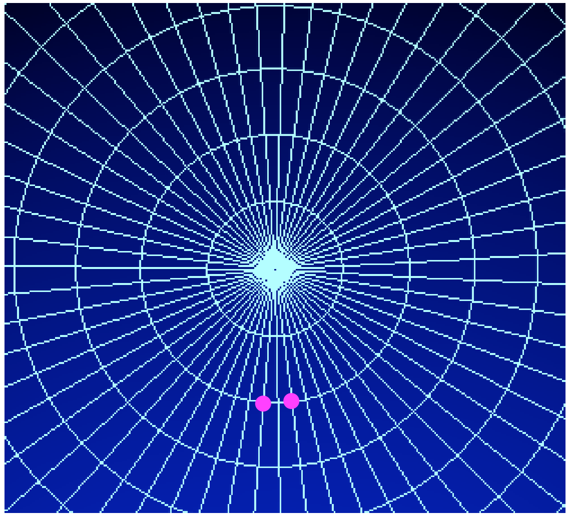
Inefficiency of Lat-Lon Grid

- Too many useless grids in high-latitudes.

  (1) Place many grid points near the poles.
  (Spoiling the low-latitude's resolution.)

  (2) Work hard to calculate data on the grids.

  (3) Throw away most of the data!

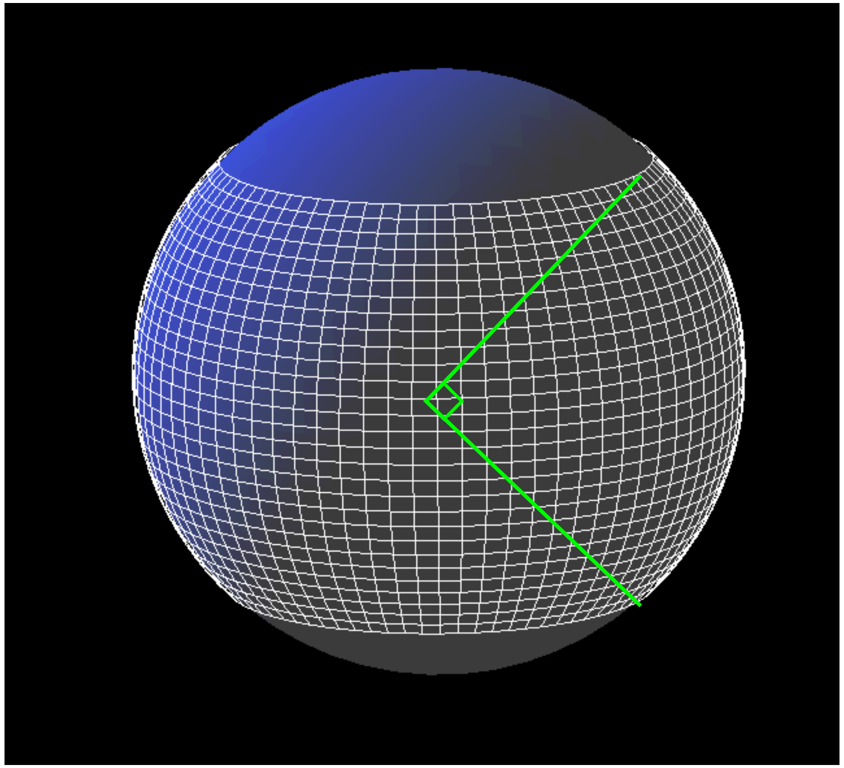


This is true for other spherical discretization methods:
- Double FFT spectral method (FFT both in latitude & longitude).
- Single FFT, hybrid method (FD in latitude & FFT in longitude).

Re-view the latitude-longitude grid

It is almost an ideal grid in the *low latitude region*.

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



Overset grid method applied to a sphere

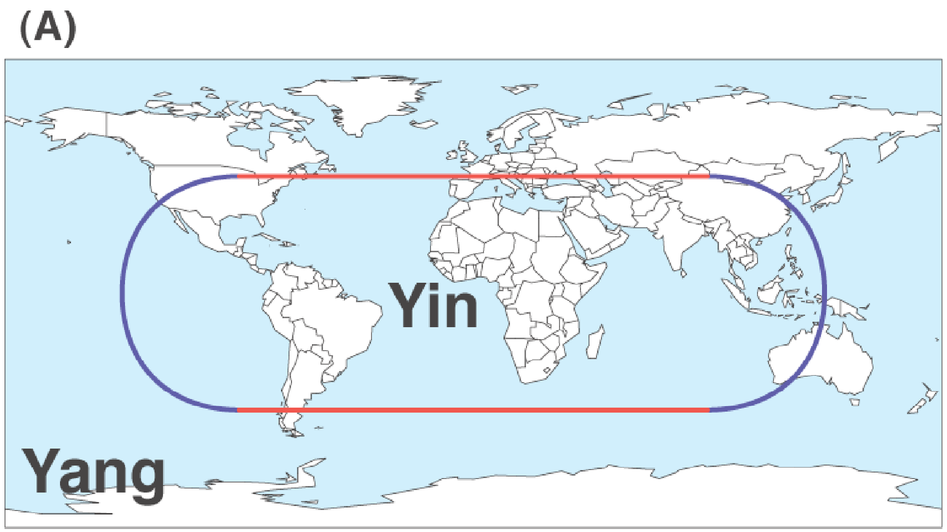
- What is the simplest overset grid on a sphere?
  - Number of component grid = 2
  - The two component grids are the same

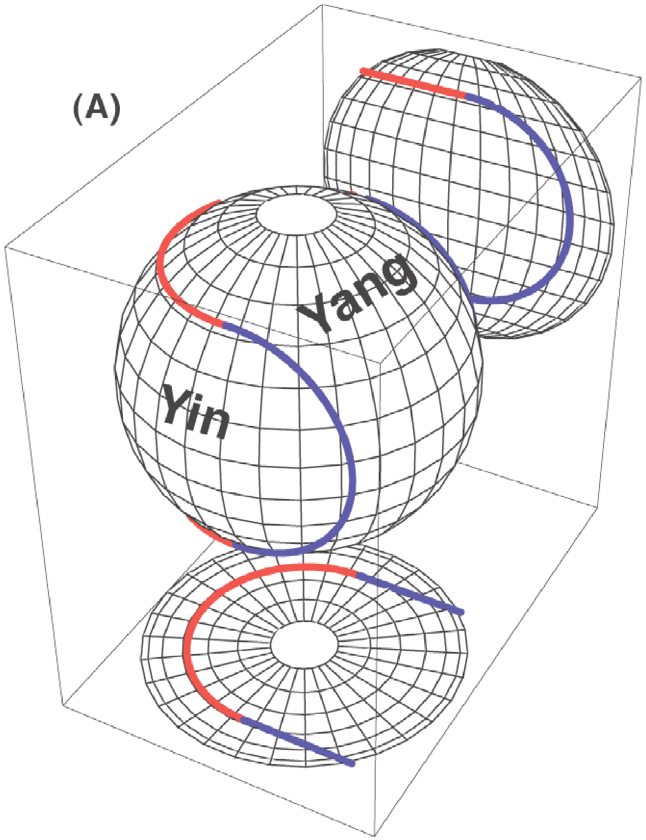
A baseball (or tennis ball)

A spherical surface is covered by

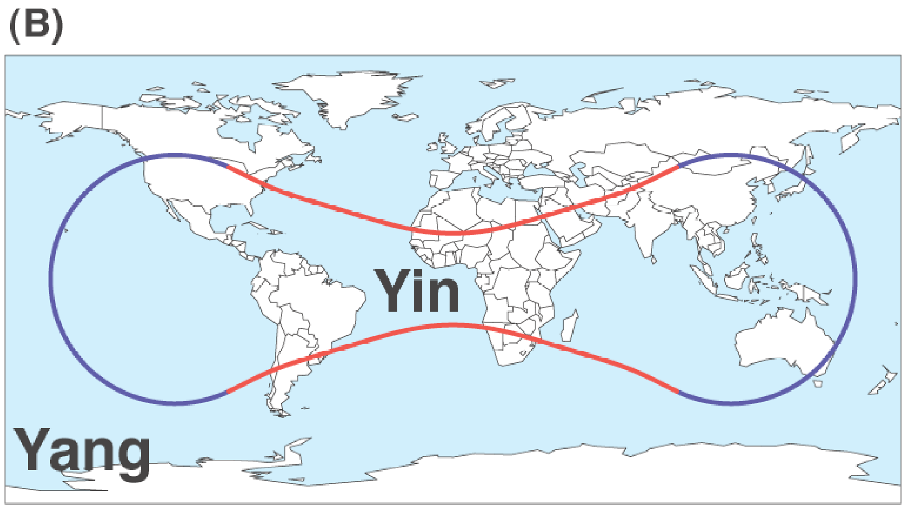
- combination of two identical parts (patches).
- one seam.

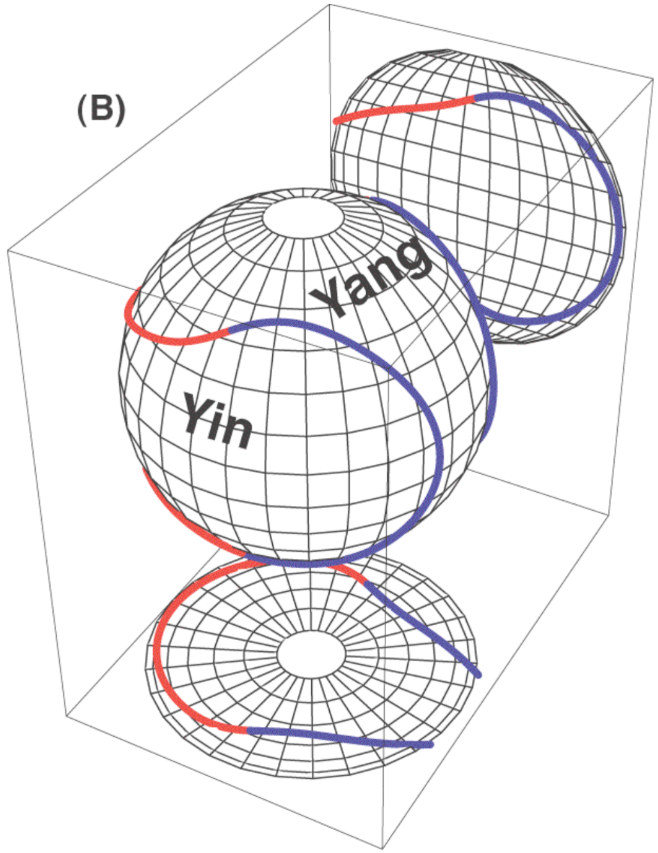
Dissections of a sphere into two identical parts





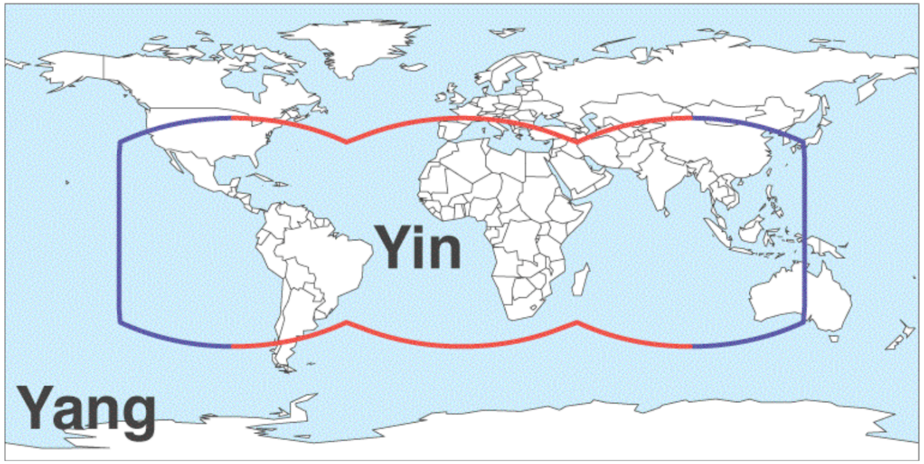
Dissections of a sphere into two identical parts



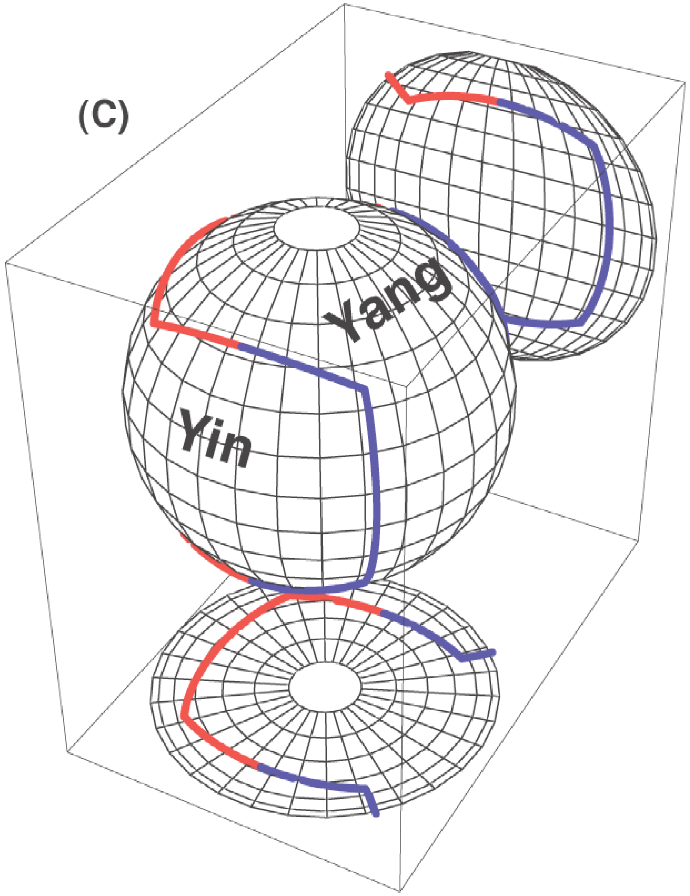


Dissections of a sphere into two identical parts

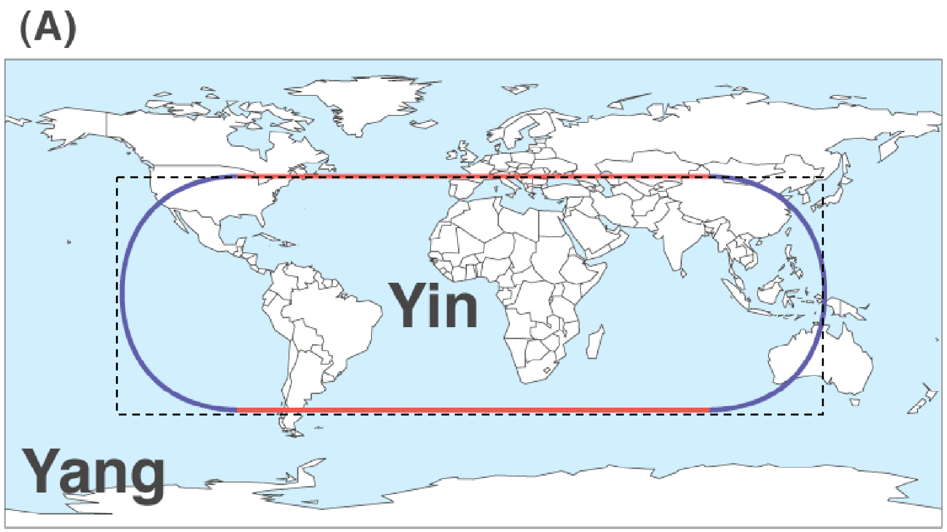
(C)

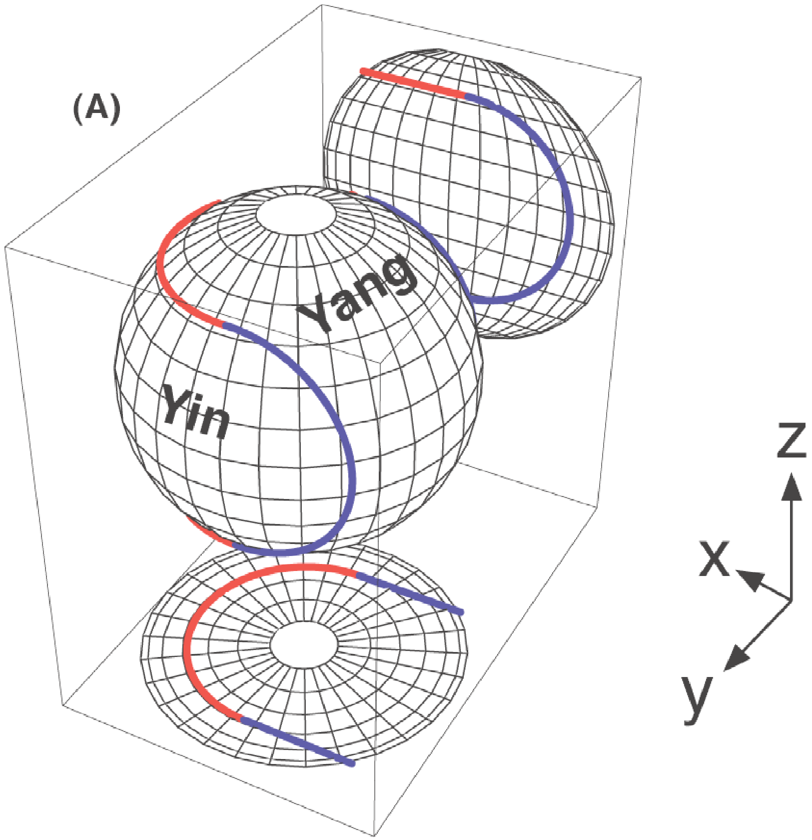


(C)

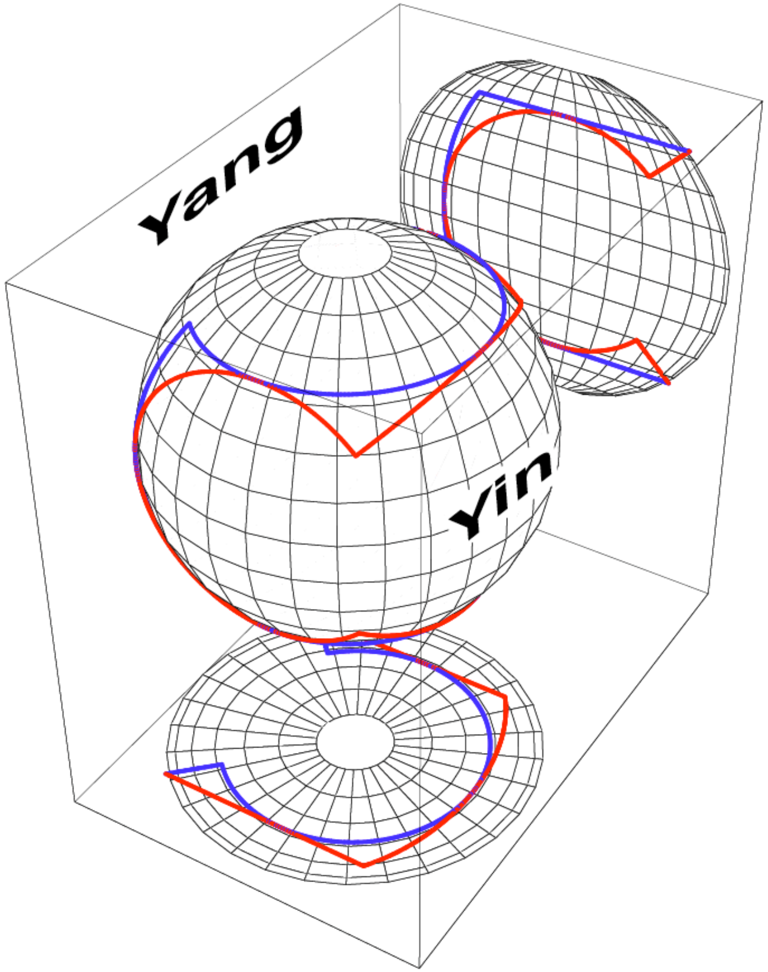


Dissections of a sphere into two identical parts

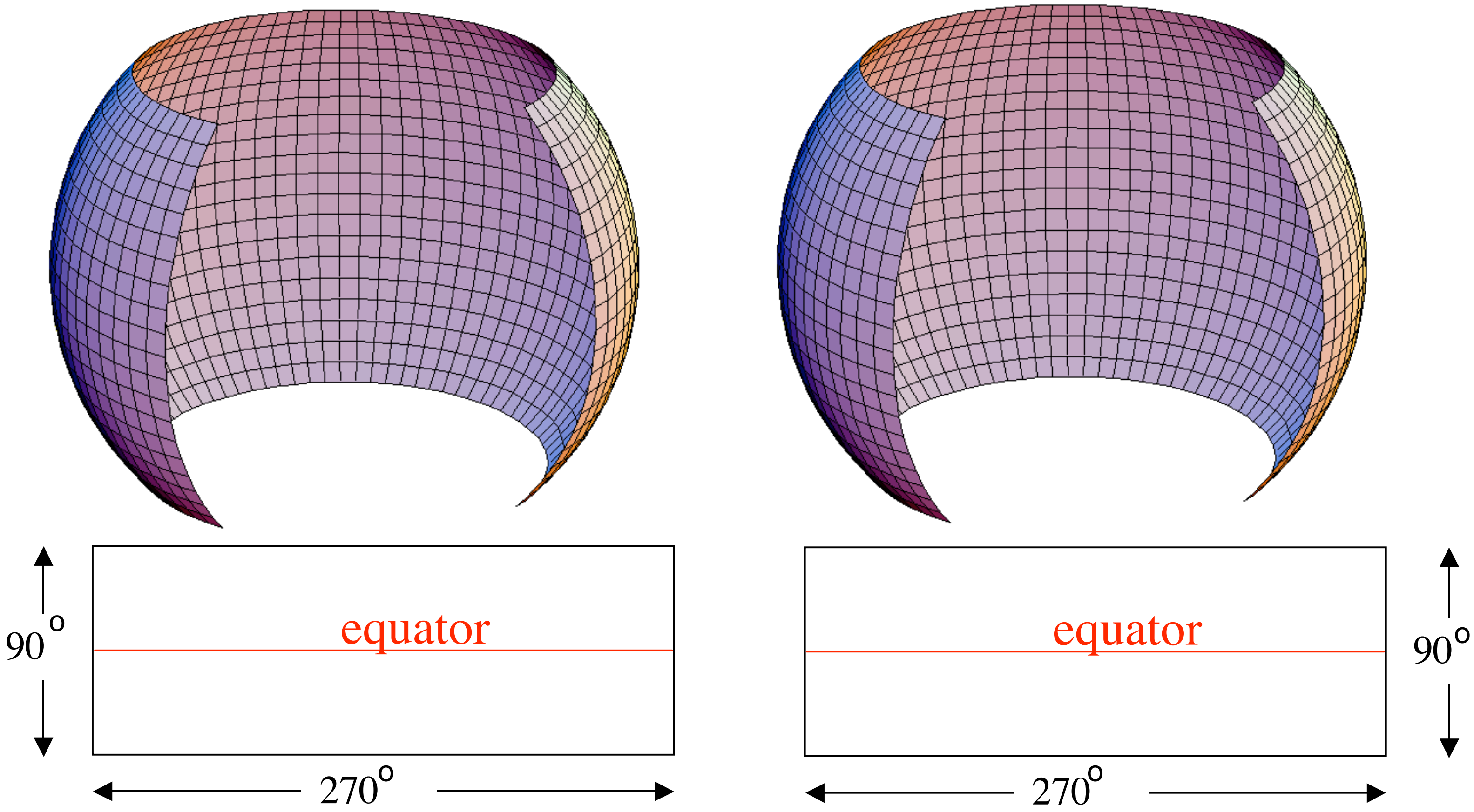




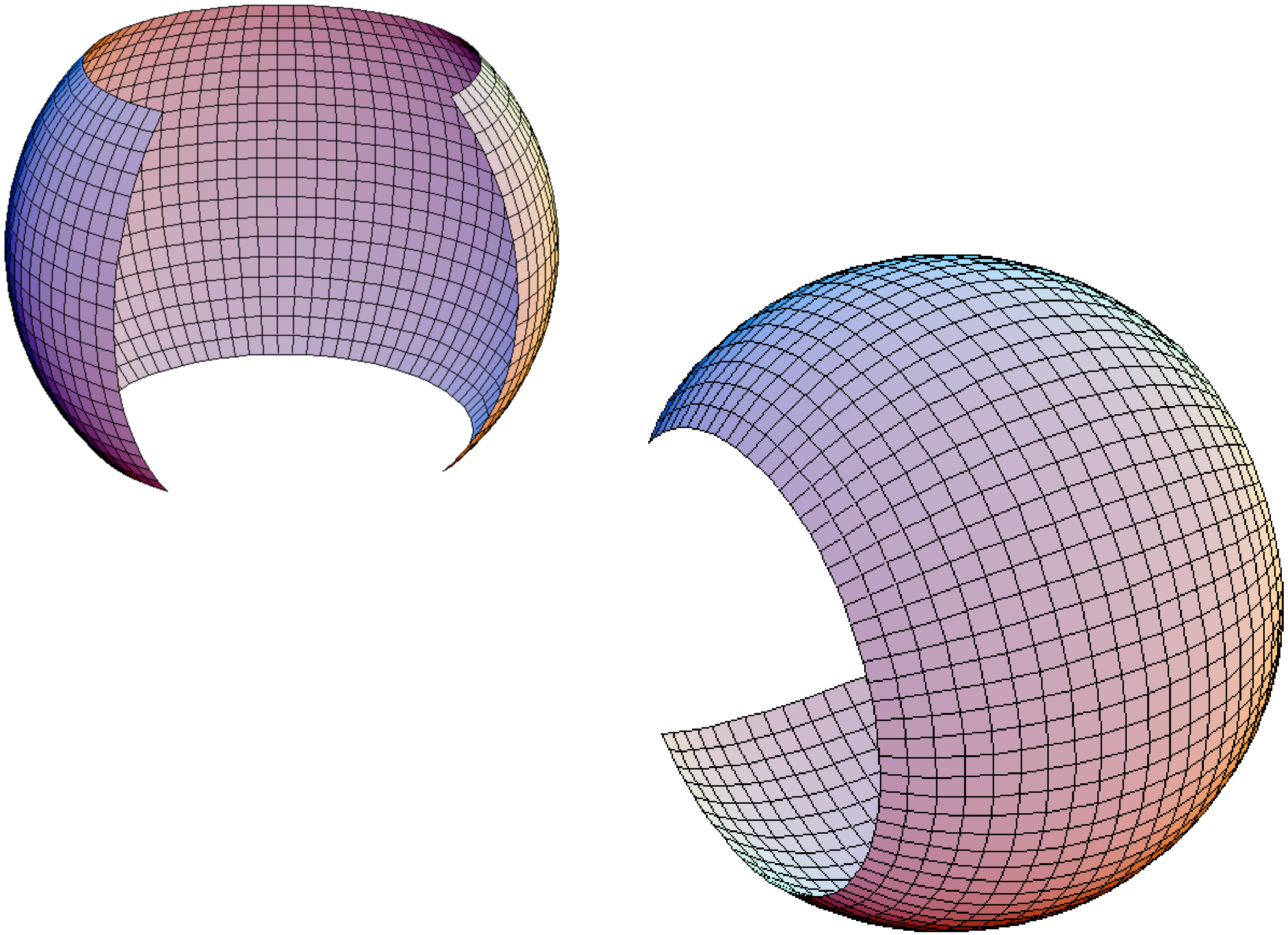
Spherical dissection with partial overlap



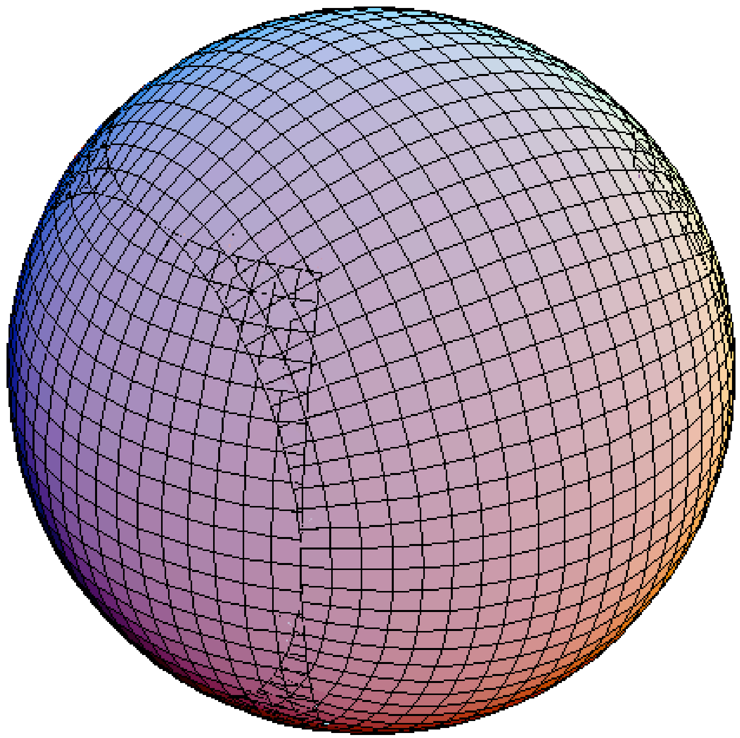
Combining two identical sub-grids to cover a full sphere

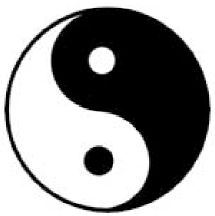


Combining two identical sub-grids to cover a full sphere



Yin-Yang grid

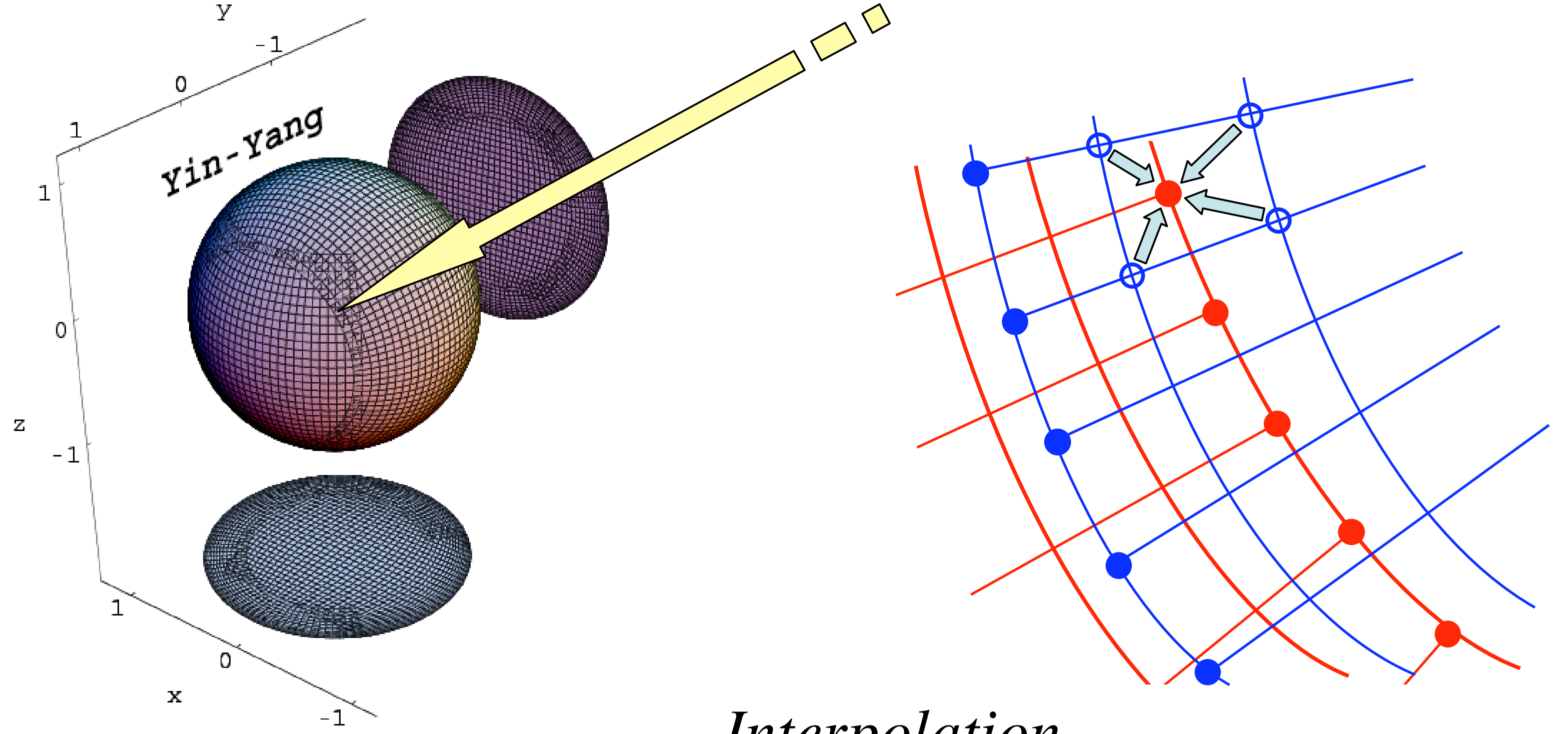




Yin-Yang 陰陽

Yin-Yang grid as an overset grid

Partial overlap between Yin and Yang grids.

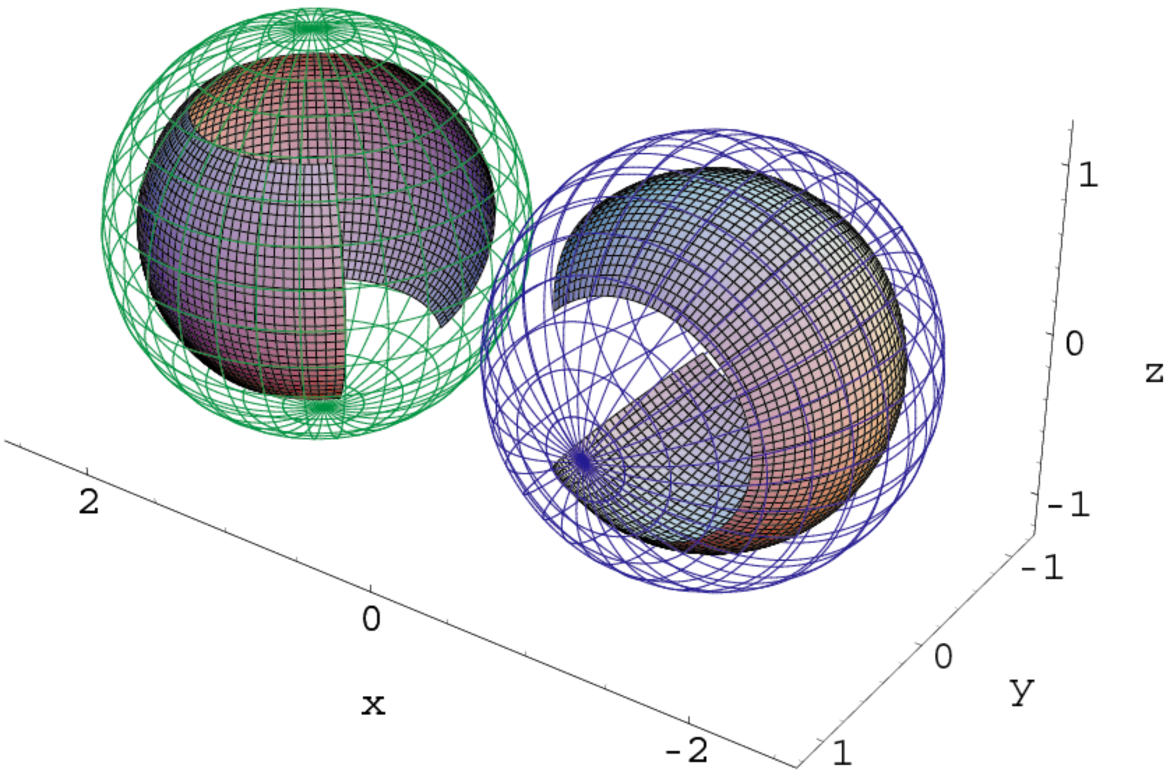


Interpolation

→ *boundary condition*

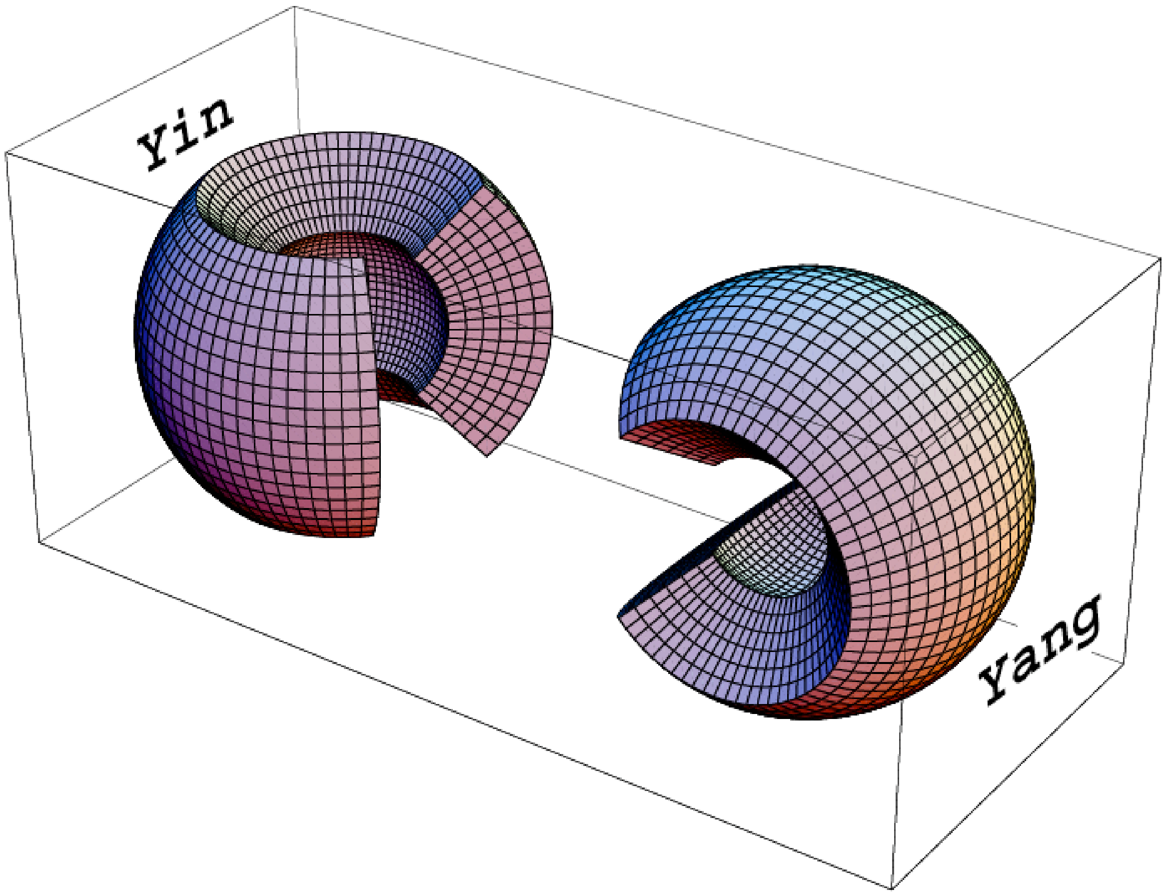
Concise coding of Yin-Yang grid:

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



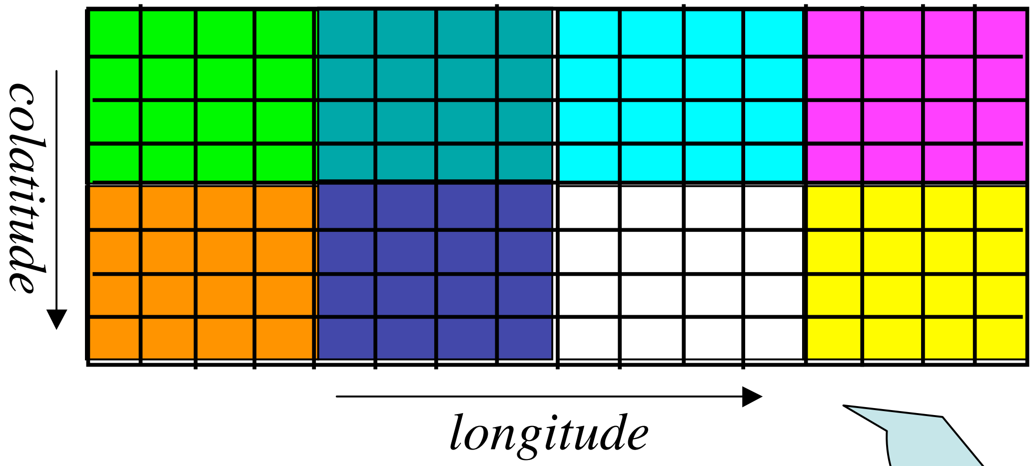
Routines for
- MHD solver
- boundary conditions
- interpolations

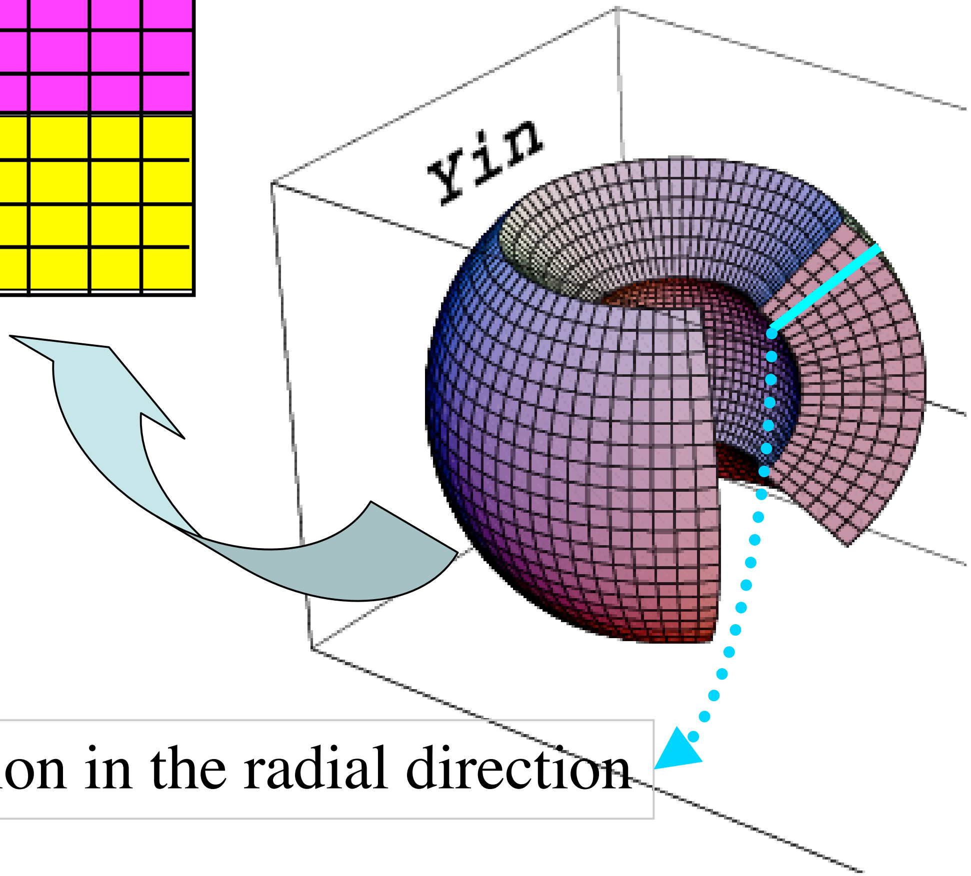
3-D Yin-Yang grid for spherical shells



Parallelization
on the Yin-Yang grid

2-dimensional domain decomposition in the horizontal computational space.





Vectorization in the radial direction

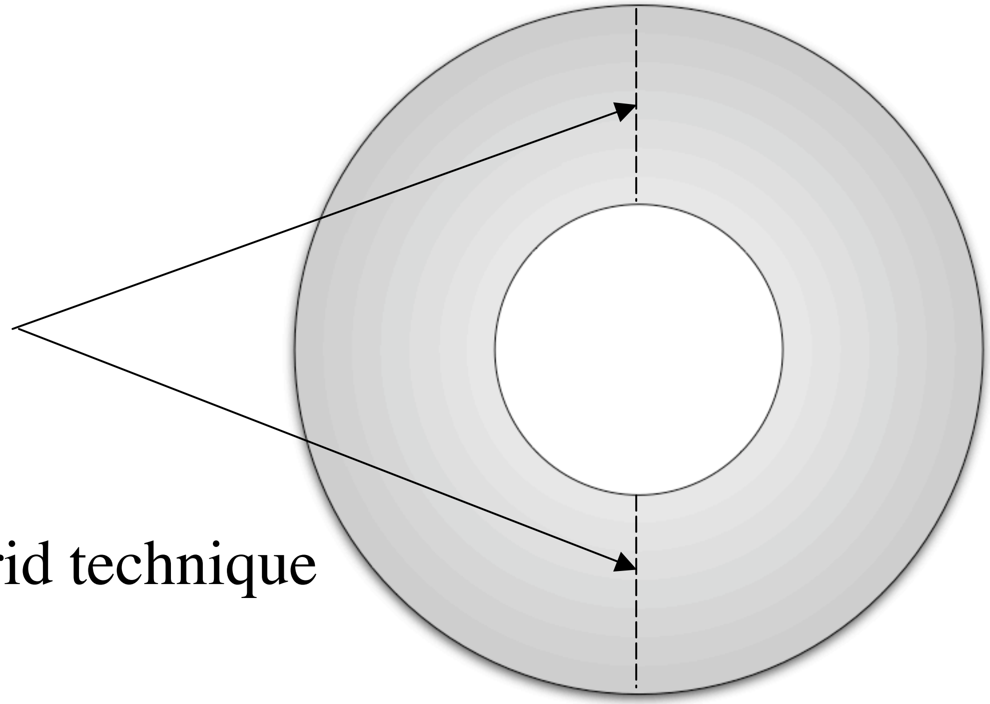
Performance of the Yin-Yang geodynamo simulation code on the Earth Simulator

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%

Overset grid technique

- Outer core

The coordinate singularity
on the poles



→ avoided by the overset grid technique

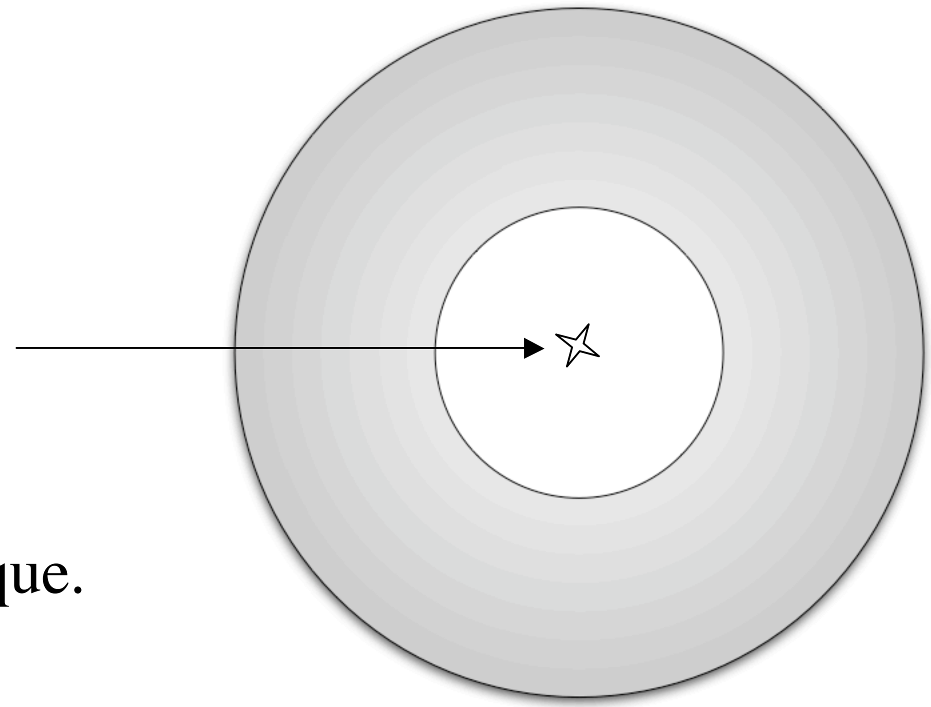
→ Yin-Yang grid

Overset grid technique

- Inner core

Coordinate singularity
on the origin (r=0).

→ Again, overset grid technique.



Cartesian grid for the inner core

- Connect it with the outer core’s Yin-Yang grid