doi: 10.32131/jes.3.20

Journal of the Earth Simulator, Volume 3, September 2005, 20 – 28

Dissection of a Sphere and Yin-Yang Grids

Akira Kageyama

The Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology,
3173–25 Showa-machi, Kanazawa-ku, Yokohama, 236-0001 JAPAN
Email: kage@jamstec.go.jp

(Received February 28, 2005; Revised manuscript accepted March 24, 2005)

Abstract A geometrical dissection that divides a spherical surface into two identical pieces is considered. When the piece is symmetric in two perpendicular directions, the two pieces are called yin and yang and the dissection is yin-yang dissection of a sphere. The yin and yang are mapped each other by a rotation M on the sphere where $M^2 = 1$. Therefore, the yin's landscape viewed from yang is exactly the same as the yang's landscape viewed from yin, and vice versa. This complemental nature of the yin-yang dissection leads to the idea of new spherical overset grid named Yin-Yang grid. The flexibility of the yin-yang dissection of a sphere enables one to patch the piece with an orthogonal, quasi-uniform grid mesh. Since the two pieces are identical, one computational routine that involves individual calculation in each grid is used for two times, one for yin grid and another for yang. Other routines that involve data transformation between yin and yang are also recycled for two times because of the complemental nature of the grids. Due to the simplicity of the underlying grid geometry, the Yin-Yang grid suits to massively parallel computers.

Keywords: dissection, Yin-Yang grid, spherical grid

1. Introduction

Recently, we proposed a new spherical grid system named "Yin-Yang grid" [1], which is a kind of overset or Chimera grids applied to a spherical surface or spherical shell. The Yin-Yang grid is already used with good success in simulations of geodynamo [2, 3], mantle convection [4], and coupled simulation of atmosphere and ocean [5, 6]. In this paper, we review the Yin-Yang grid with a special emphasise on the motivation and basic ideas behind it.

In these ten years, we have been performing simulation research of magnetohydrodynamic (MHD) dynamo in spherical shells. For the spatial discretization of the MHD equations, we used a finite difference method with a prospect of its increasing importance in the era of massively parallel computers. The base grid system adopted in our previous spherical shell MHD code was the latitude-longitude (Lat-Lon) grid, which is defined on the spherical polar coordinates (r, θ, ϕ) with radius r, colatitude θ, and longitude ϕ. For the spherical surface S, the grid mesh of the Lat-Lon grid is uniform when it is seen in the computational space

$$S := \{\theta, \phi\}, \ |\theta - \pi/2| \le \pi/2, |\phi| \le \pi, \tag{1}$$

but it is far from uniform when it is seen in the real space.

A numerical problem in the Lat-Lon grid is the existence of the coordinate singularity on the north pole

$$P_n := \{\theta = 0, \phi\}, \tag{2}$$

and the south pole

$$P_s := \{\theta = \pi, \phi\}. \tag{3}$$

In the spherical polar coordinates, all the differential operators should be represented in three different forms for three local regions of S. Take the gradient operator $\nabla = (\nabla_r, \nabla_\theta, \nabla_\phi)$ as an example. Define a local region S' by

$$S' := S - P_n - P_s, \tag{4}$$

which is the spherical surface without the poles. The gradient operator is represented by

$$\nabla = \left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}, \frac{1}{r\sin\theta}\frac{\partial}{\partial \phi}\right) \ \text{for } S', \tag{5}$$

which is a familiar form. On the other hand, the gradient operator should be represented in other, unfamiliar, forms for the north and south poles as

$$\nabla = \left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}, +\frac{1}{r}\frac{\partial^2}{\partial \theta\, \partial \phi}\right) \ \text{for } P_n, \tag{6}$$

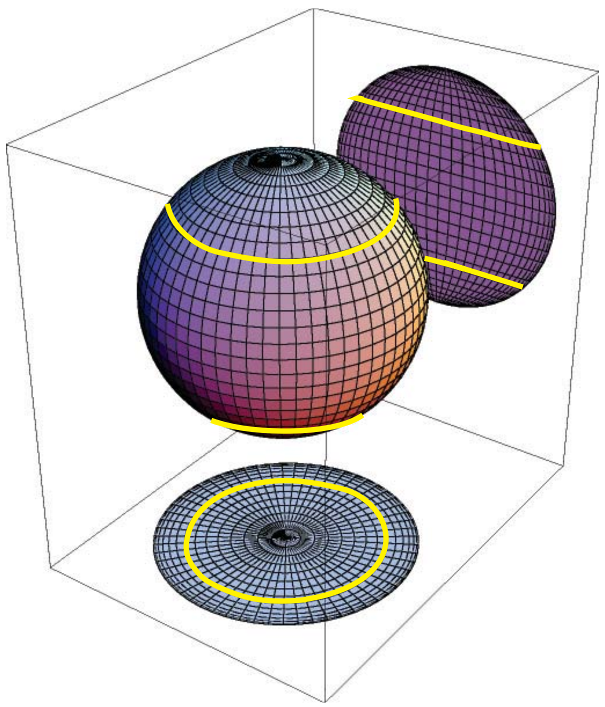

Fig. 1 Latitude-longigude (Lat-Lon) grid. There are two numerical problems in Lat-Lon grid; the coordinate singularity *on* the poles and the grid convergence *near* the poles. If we focus on the low latitude part, the mesh of Lat-Lon grid has desirable features; it is orthogonal and it has quasi-uniform grid spacings.

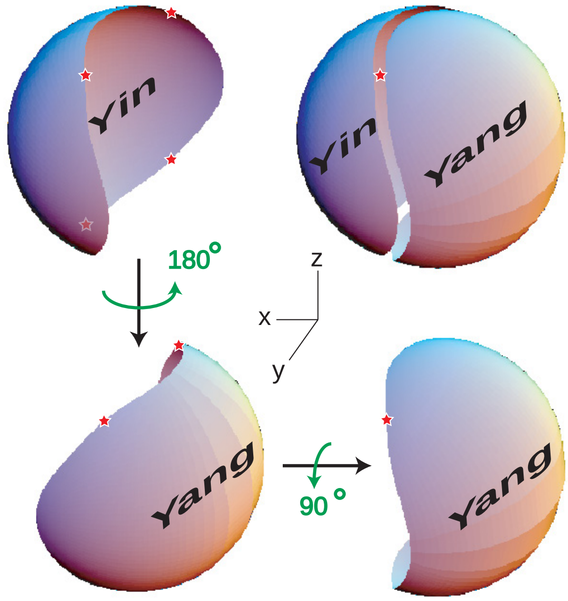

Fig. 2 An example of yin-yang dissection of a sphere. In this dissection, a sphere is divided into two identical pieces, with same shape and size. Each piece has up-down and right-left symmetries. It is constructed by a copy of one piece, followed by two rotations.

and

$$\nabla = \left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}, -\frac{1}{r}\frac{\partial^2}{\partial\theta\,\partial\phi} \right) \quad \text{for } P_s. \tag{7}$$

The above eqs. (6) and (7) are derived from eq. (5) by the L'Hospitals rule. For numerical simulations, therefore, we have to prepare basically three different subroutines for numerical computation of discretized equations. This enhances the complexity of the computer programs.

Another problem in the Lat-Lon grid is the grid convergence near the poles that is obviously seen in Fig. 1. By a simple count, we know that nearly 84% of all grid points on the sphere are located in higher latitudes than 45°N and 45°S (the yellow lines in Fig. 1). In other words, the low latitude region between 45°N and 45°S is covered by only 16% of grid points.

The grid convergence imposes a severe restriction on the time step Δt in explicit time integration schemes. This is recognized in, for example, the metric factor $1/(r \sin\theta)$ of ϕ-component of ∇ in eq. (5). As θ approaches to 0 or π in S', this factor blows up with a constant grid spacing $\Delta\phi$ in the azimuthal direction.

In order to relax the restriction, one has to apply a kind of low-pass filter so that the grid spacing on the sphere becomes *effectively* quasi-uniform. The spherical filter itself has sizable computational costs. But even if you could develop a very high-speed (with high-flops) spherical filter, it merely means that you can perform very futile task very quickly. It is similar to build a hill on a flat land, then remove it, and repeat the cycle every one second. Under the spherical filter method, one integrates physical quantities according the basic equations on all the grid points — 84% of which are concentrated in high-latitude — then quickly throw away a lot of information just obtained, and this cycle is repeated every time step. This is computationally inefficient in total, even if it outputs an apparent high flops value.

Note that the above two problems (the coordinate singularity and the grid convergence) of the Lat-Lon grid come from the region of high latitudes. The remaining part of the Lat-Lon grid — the low latitude region — has rather desirable feature for numerical simulations; it is an orthogonal grid, it has simple metric tensors, and the grid spacings are quasi-uniform.

2. Dissection of a Sphere

There is no grid mesh that is (i) orthogonal, (ii) free of coordinate singularity, (iii) free of grid convergence problem, and (iv) defined over a spherical surface. We have to discard one of these incompatible conditions.

Since the orthogonality is evidently desirable for

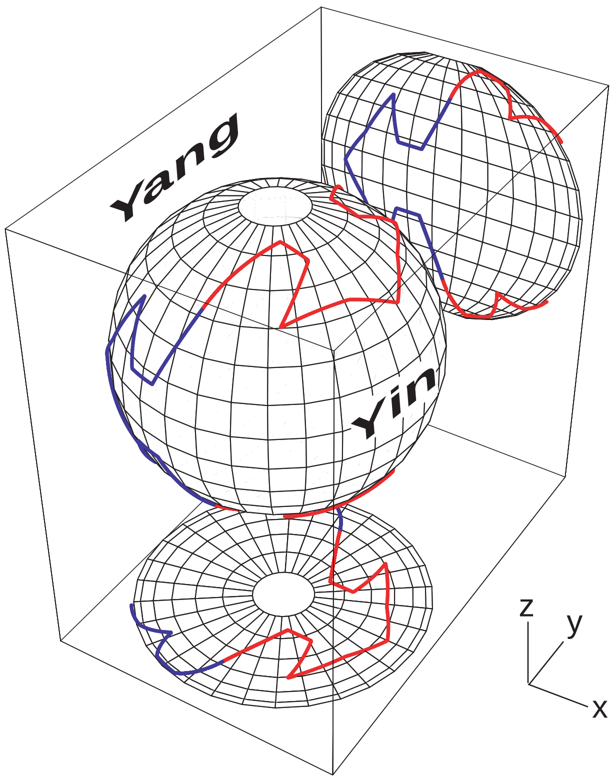

Fig. 3 An example of yin-yang dissection of a sphere. This is no doubt impractical for numerical calculations.

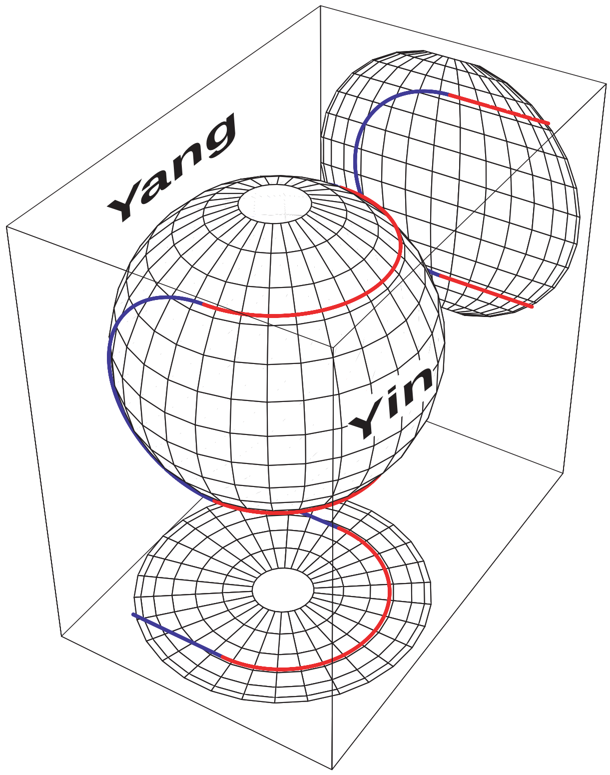

Fig. 4 One of (probably the simplest) yin-yang dissection of a sphere.

numerical calculations, we keep this feature in our grid design. Instead, we discard the condition (iv) above; the single nature of the spherical surface by taking a *divide-and rule* approach: We decompose a spherical surface into subregions. The decomposition, or dissection, will enable us to cover each subregion by a grid system that is individually orthogonal and singularity-free.

How many subregions should we take? There are infinite variations of spherical dissections depending on the number n (≥ 2) of divided pieces of the sphere. Among them, here we consider the minimum case of $n = 2$, i.e., the spherical dissections by two pieces. There are still infinite number of such dissections. For example, one can divide a sphere into two parts by cutting along a small circle at any latitude. Among the indefinite possibilities of spherical dissection with $n = 2$ pieces, we concentrate on a class of dissections which divides a sphere into two parts that are geometrically identical, i.e., the two pieces have exactly same size and shape. Another condition we impose is the symmetry of the piece; the piece is symmetric in two perpendicular directions; up-down and right-left. Here we call such dissection as yin-yang dissection of a sphere.

A trivial example of the yin-yang dissection is obtained by cutting along the equator or any great circle, which produces two hemispheres. Other yin-yang dissections are obtained by modifying the cut curve from the great circle. Let S_{yin} be a piece of a sphere S of radius r with the surface area $2\pi r^2$. An example of S_{yin} is shown in the upper left panel in Fig. 2. Make a copy of S_{yin} and call it S_{yang} which is rotated for 180° around z-axis. (See lower left panel of Fig. 2.) Then, rotate it again, but this time for 90° degree around x-axis, as shown in the lower right panel. Then the original piece S_{yin} (the upper left) and the rotated copy S_{yang} (the lower right) are combined so that they reproduce S as shown in the upper right in this figure.

The shape of S_{yin} is not arbitrary. What is the condition for the curves for the yin-yang dissection of a sphere?

A careful inspection of Fig. 2 leads us to a necessary condition of the cut curve. Suppose that the spherical radius is $r = \sqrt{2}$. The curve passes the following four points on the sphere at $(x, y, z) = (0, -1, +1)$, $(0, +1, +1)$, $(0, -1, -1)$, $(0, +1, -1)$. These four points form a square in the plane $x = 0$.

Consider an arbitrarily chosen curve between upper two vertices $(0, -1, +1)$ and $(0, +1, +1)$ (see the red curve in Fig. 3), and copy (or project) the curve to the southern counterpart between $(0, -1, -1)$ and $(0, +1, -1)$. Then we rotate the pair of red curves by the following consecutive rotations; first 180° around the z-axis, then 90° degree around x-axis, which leads to the blue curves in Fig. 3.

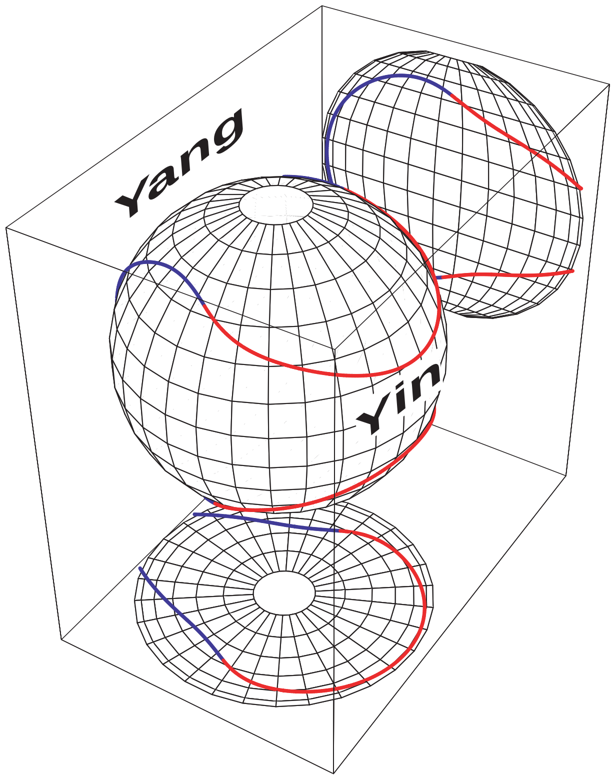

Fig. 5 Another example of yin-yang dissection. This reminds us a baseball, which is a practical application of the yin-yang dissection.

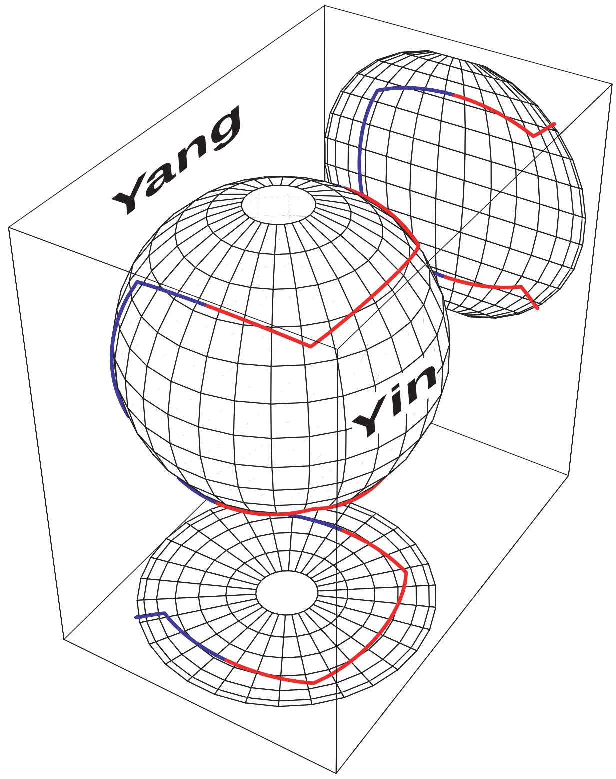

Fig. 6 Yin-yang dissection of a sphere that is constructed from a projection of a cube in the sphere.

From this construction, it is obvious that the spherical dissection defined by a closed red-blue curve in Fig. 3 generates identical two pieces. There are infinite number of variations of yin-yang dissection of a sphere since the original red curve was arbitrarily chosen.

Among infinite patterns of the yin-yang dissection, we believe that the dissection shown in Fig. 4 is geometrically the simplest. In this case, the starting red curve is the equi-latitude at $\theta = \pi/4$ that passes $(x, y, z) = (0, -1, +1)$ and $(0, +1, +1)$.

Fig. 5 shows other cut curve for yin-yang dissection that is obtained by a slight modification of the curve in Fig. 4. This curve in Fig. 5 reminds us the seam of a baseball. Another simple yin-yang dissection is shown in Fig. 6. The red-blue cut curve here is defined by a projection of a cube in the sphere from the origin $(x, y, z) = (0, 0, 0)$. This shape reminds us the "cubed sphere" [7] .

3. Overset Grid Method

In general, a dissection of a computational domain generates internal borders or boundaries between the subregions. In the overset grid methodology [8], the subdomains are permitted to partially overlap one another on their borders. The overset grid is also called as overlaid grid, or composite overlapping grid, or Chimera grid [9]. The validity and importance of the overset approach in the aerodynamical calculations was pointed out by Steger [10]. Since then this method is widely used in this field. It is now one of the most important grid techniques in the computational aerodynamics; see for example, whole aircraft with wing and store [11], tiltrotor aircraft [12], Boeing 747 [13, 14], Space Shuttle [15], helicopter [16], for impressive applications of the overset grid method.

In the computational geosciences, the idea of the overset grid approach appeared rather early. Phillips proposed a kind of composite grid in 1950's to solve partial differential equations on a hemisphere, in which the high latitude region of the latitude-longitude grid is "capped" by another grid system that is constructed by a stereographic projection to a plane on the north pole [17, 18, 19]. After a long intermission, the overset grid method seems to attract growing interest in geoscience these days. The "cubed sphere" [7] is an overset grid that covers a spherical surface with six component grids that correspond to six faces of a cube. The "cubed sphere" is recently applied to the mantle convection simulation [20]. In the atmospheric research, other kind of spherical overset grid is used in a global circulation model [21], in which the spherical surface is covered by two component grids — improved stereographic projection grids — in northern

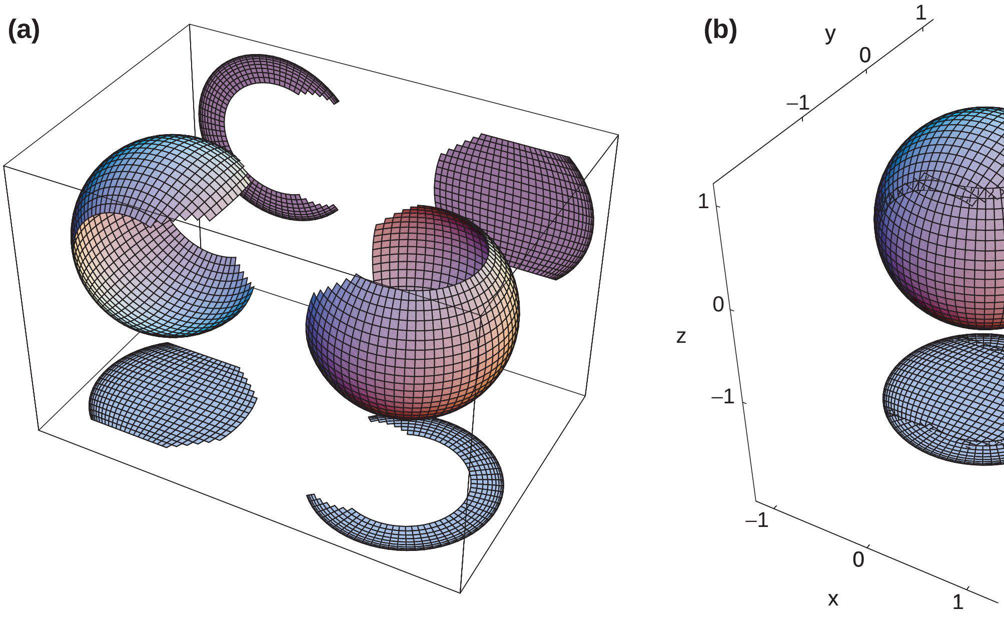

Fig. 7 A Yin-Yang grid based on the yin-yang dissection of Fig. 4.

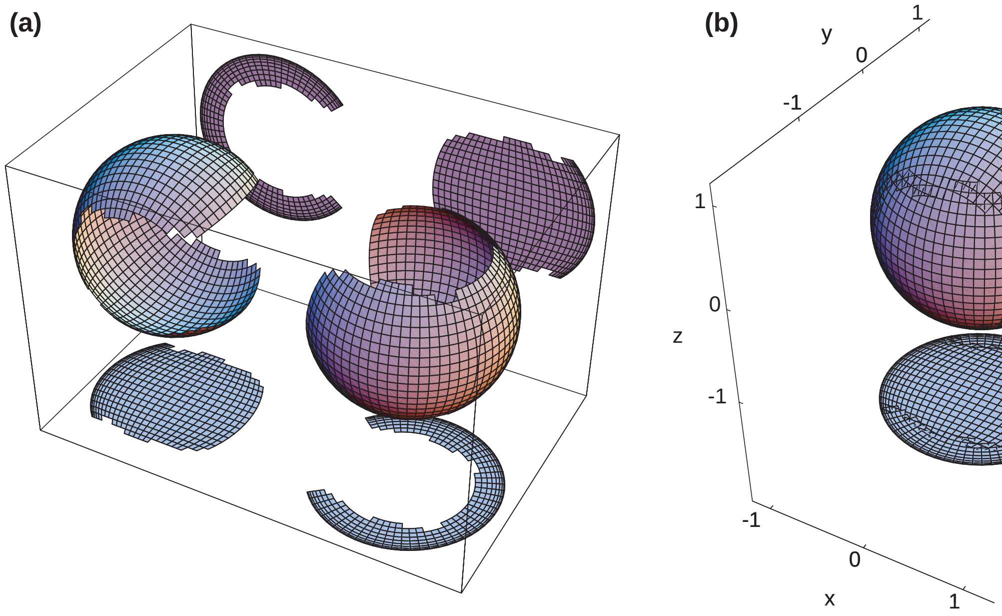

Fig. 8 A Yin-Yang grid based on the yin-yang dissection of Fig. 6.

and souther hemispheres that overlap in the equator.

4. Yin-Yang Grids

The combination of the yin-yang dissection discussed in section 2 and the overset grid method in section 3 leads us to the Yin-Yang grid.

Take the simplest yin-yang dissection shown in Fig. 4. It is a natural idea to use (a part of) usual Lat-Lon grid in the area denoted by "Yin" in Fig. 4 as suggested by the background Lat-Lon grid mesh in the figure. How about Yang part? (Obviously, we should not use the original Lat-Lon mesh for this area since it has both of the "trouble maker"; the north and south poles.) Remember that the Yin part and Yang part are geometrically identical. So, we can patch a grid mesh to the Yang part with the same mesh in the same way used in the Yin part. This is one of the basic ideas of the Yin-Yang grid. Fig. 7 shows the Yin-Yang grid based on the yin-yang dissection of

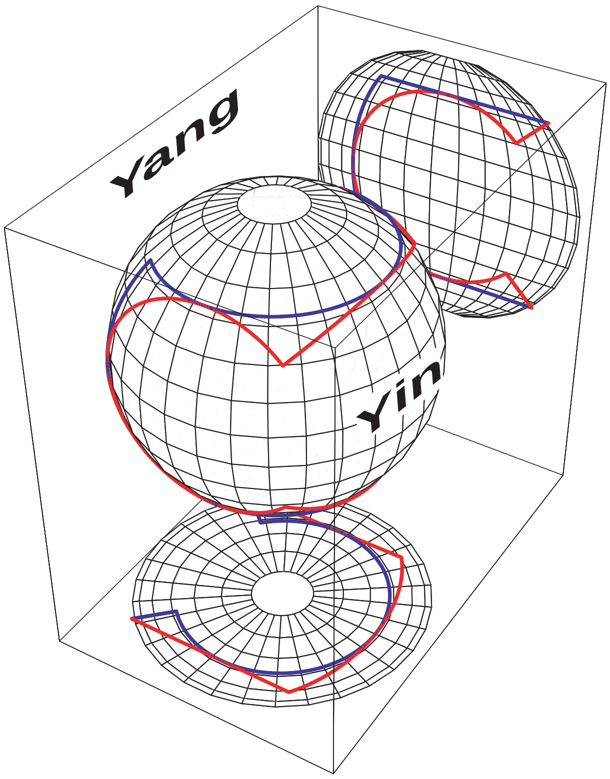

Fig. 9 A dissection of a sphere into two identical pieces — Yin and Yang — with a partial overlap. The blue (red) curve is closed and it is the border of Yin (Yang) piece. In contrast to Figs. 3 – 6, the blue curve is always located in either constant latitudes or constant longitudes. In other words, the Yin (Yang) piece is a rectangle in the computational (θ, ϕ) space of the Yin (Yang) grid.

Fig. 4. Note that there is an overlap between the Yin grid and Yang grid which is required by the overset grid method.

Other type of the Yin-Yang grid is shown in Fig. 8. This Yin-Yang grid is constructed from the yin-yang dissection shown in Fig. 6. We notice in Fig. 8(a) that the perimeter of the component grid (Yin or Yang) has rather irregular shape. Suppose that physical quantities are defined on mesh points on (θ_j, ϕ_k) of Yin grid and Yang grid of Fig. 8 with integer indexes j and k. The irregular shape of perimeter of the component grid in Fig. 8 means that the integer indexes cannot span simply as $j_{min} \le j \le j_{max}$, and $k_{min} \le k \le k_{max}$. One needs to invoke some kind of mask procedure to skip unnecessary j and k in the simulation code. The mask would degrade the program code's simplicity, and possibly the calculation speed, too. The mask procedure is also required in the Yin-Yang grid shown in Fig. 7. From numerical point of view, the ideal shape of the component grid is a rectangle in the computational (θ, ϕ) space.

The overset grid methodology gives us a freedom to design the shape of the component grid as long as the grids has *minimum* overlap one another. Therefore, we can take the component grid as a rectangle in (θ, ϕ) space. Fig. 9 shows a spherical dissection by two identical pieces, in which the two pieces are partially overlapped one another. Note that the northern and southern borders of the Yin piece denoted by the blue curve in Fig. 9 are located in constant latitudes and the western and eastern borders are located in constant longitudes. In other words, the Yin piece in Fig. 9 is a rectangle in the (θ, ϕ) space of the Yin's spherical coordinates, and therefore, the Yang piece is also (the same) rectangle in Yang's coordinates that is perpendicular to the Yin's. The Yin-Yang grid based on this partially overlapped spherical dissection is shown in Fig. 10. Here, each component grid spans the following subregion of the sphere S:

$$S_y := \{\theta, \phi\}, \;\; |\theta - \pi/2| \le \pi/4 + \delta, |\phi| \le 3\pi/4 + \delta, \quad (8)$$

with a small buffer δ which is necessary to keep the overlap between Yin and Yang.

For any type of the Yin-Yang grids described above as well as possible other variations, the relation between Yin coordinates (x^n, y^n, z^n) — denoted in the Cartesian coordinates — and Yang coordinates (x^e, y^e, z^e) is given by

$$\begin{pmatrix} x^e \\ y^e \\ z^e \end{pmatrix} = M \begin{pmatrix} x^n \\ y^n \\ z^n \end{pmatrix}, \quad (9)$$

where

$$M = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \quad (10)$$

Note that

$$M^{-1} = M, \quad (11)$$

which indicates a complemental relation between Yin coordinates and Yang coordinates: The coordinate transformation from Yin to Yang is mathematically the same as that from Yang to Yin. From a programming point of view, this enables us to make only one, instead of two, subroutines that involve any data transformation between Yin and Yang.

Another advantage of the Yin-Yang grid resides in the fact that the component grid is nothing but (a part of) the Lat-Lon grid. We can directly deal with the equations to be solved with the usual spherical polar coordinates. The analytical form of metric tensors are familiar in the spher-

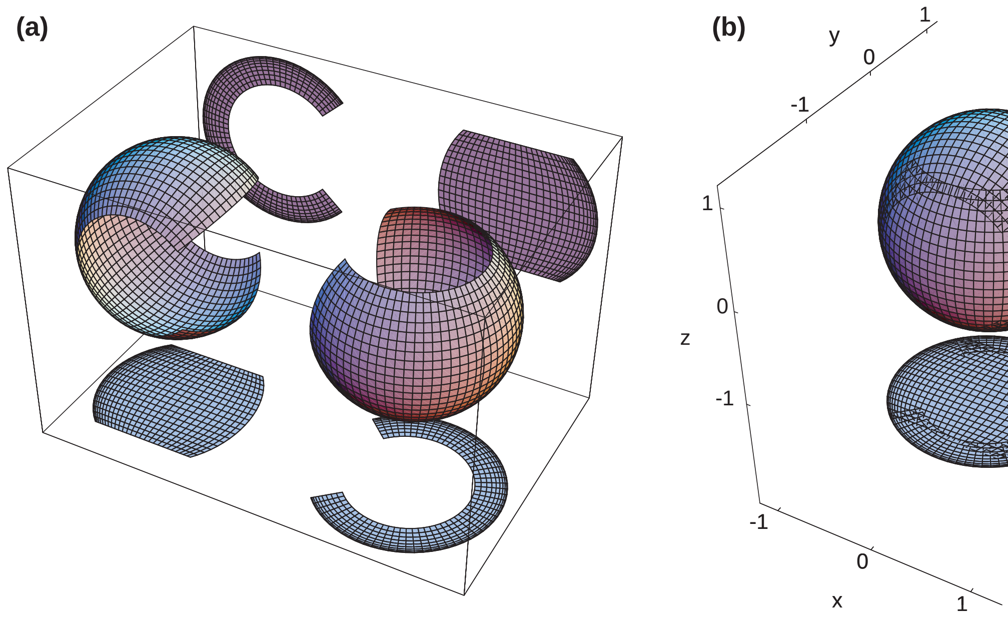

Fig. 10 A Yin-Yang grid based on the yin-yang dissection with partial overlap shown in Fig. 9. Each component grid is rectangle in the computational (θ, ϕ) space.

ical coordinates. We can directly code the basic equations in the program as they are formulated in the spherical coordinates. For example, the gradient operator ∇ is implemented in the finite difference method based on the usual formula on the defining space of Yin grid or Yang grid:

$$\nabla = \left(\frac{\partial}{\partial r}, \frac{1}{r}\frac{\partial}{\partial \theta}, \frac{1}{r\sin\theta}\frac{\partial}{\partial \phi} \right) \quad \text{for } S_y. \tag{12}$$

Note also that the factor $1/(r \sin \theta)$ does not blow up in S_y. We can make use of various resources of mathematical formulas, program libraries, and tools that have been developed in the spherical polar coordinates.

In order to illustrate an example of the programing strategy in the Yin-Yang method, here we take a two-dimensional fluid problem on a sphere S. Suppose that two components of the flow velocity $\mathbf{v} = (v_\theta, v_\phi)$ and the pressure p are written in `vel_t`, `vel_p`, and `press` in a Fortran 90/95 program. They can be combined into one structure or "type" in Fortran 90/95 as

```
type fluid_
 real, dimension(NT, NP) :: vel_t, vel_p, press
end type fluid_
```

where `NT`, `NP` are the grid size integers in θ and ϕ directions in the subregion S_y of eq. (8). Using this structured type, we declare two variables for the fluid; one is for the fluid in the Yin region and another is for the fluid in the Yang region:

```
type(fluid_) :: fluid_yin, fluid_yang
```

Then, we call a fluid solver subroutine, here named `navier_stokes_solver`, that numerically solves the Navier-Stokes equation in the spherical coordinates in the subregion S_y:

```
call navier_stokes_solver(fluid_yin)
call navier_stokes_solver(fluid_yang)
```

The first call of `navier_stokes_solver` solves the fluid motion in the S_y region defined in the Yin's spherical coordinates and the second call is for the same region S_y defined in the Yang's coordinates. But in the program code, we do not have to distinguish the two S_y regions since the basic equations, numerical grid distribution, and therefore, all numerical tasks are identical in the computational space. For a rotating fluid problem with a constant angular velocity Ω, we have the Coriolis force term in the Navier-Stokes equation that seems to break the symmetry between the Yin grid and Yang grid, but it is still possible to write the equation in exactly the same form for the Yin and Yang grids by explicitly writing three components of angular velocity in the Coriolis force term $2\mathbf{v} \times \Omega$ in the subroutine. Then, we call the routine with the angular velocity vector in each grid (Yin or Yang) as the second argument:

```
call navier_stokes_solver(fluid_yin, omega_yin)
call navier_stokes_solver(fluid_yang, omega_yang)
```

where `omega_yin` and `omega_yang` are again structured variables that hold three components of the Ω vector: For example, `omega_yin` holds three components of cartesian vector components in the Yin grid $(\Omega_x^n, \Omega_y^n, \Omega_z^n) = (0, 0, \Omega)$, and `omega_yang` holds $(\Omega_x^e, \Omega_y^e, \Omega_z^e) = (0, \Omega, 0)$. Note that $(\Omega_x^n, \Omega_y^n, \Omega_z^n)$ and $(\Omega_x^e, \Omega_y^e, \Omega_z^e)$ are related each other by the transformation matrix M in eq. (10).

Our experience tells that it is easy to convert an existing Lat-Lon based program into a Yin-Yang based program since there are many shared routines between them. In addition to that the size of the code as well as its complexity is drastically reduced by the conversion. One reason is that we can remove routines that are designed to solve pole grids: We do not have to take special cares to the differential operators in P_n and P_s such as eqs. (6) and (7). Besides, we do not need spherical filter routines in the Yin-Yang grid. Another remark on the code's simplicity in the Yin-Yang grid method is that we can recycle one routine that involves individual grid (Yin grid or Yang grid) for two times as illustrated in the above example.

5. Parallel Computing on the Yin-Yang Grid

Suppose that we have only two processors available. For the computer simulation using the Yin-Yang grid, it is natural to perform a parallel computing by dividing the computational task into Yin part and Yang part. Because the two component grids are identical, we can achieve a perfect balance of the computational loads between the processors.

When we have $2N$ processors, we decompose each component grid into N subdomains for N processors each. The domain decomposition is simple and straightforward in the Yin-Yang grid since the component grids (Yin grid and Yang grid) is geometrically simple; it is a rectangular box in the computational space of (r, θ, ϕ).

We described in [2] an example of two-dimensional domain decomposition in (θ, ϕ) space for MPI-based parallel computation of geodynamo simulation. In the maximum case, we achieved 15.2 Tflops by 4096 processors of the Earth Simulator.

6. Summary

If you knife along a baseball's seam, you will get a couple of two identical patches by which the ball's spherical surface is covered in combination. The seam of the baseball is an example of a family of geometrical dissections that divides a sphere into two identical pieces. When the piece (or patch) of the dissection has up-down and right-left symmetries, the two pieces are transfered each other by a rotation denoted by the matrix M defined by eq. (10). In this paper, we call this kind of dissection as yin-yang dissection of a sphere. Since $M^{-1} = M$, the yang's landscape viewed from a point on the yin is exactly the same as the yin's landscaped viewed from the corresponding point on the yang, and vice versa. This complemental nature is the most remarkable feature of the yin-yang dissection.

Inspired by the yin-yang dissection of a sphere, we have developed a new overset grid for spherical geometry. The two component grids, called Yin grid and Yang grid, are geometrically identical. They are combined in the complemental way to cover the spherical surface with partial overlap one another.

To conclude this article, we summarize merits of the Yin-Yang grid: It is an orthogonal system (since it is a part of the Lat-Lon grid); The grid spacing is quasi-uniform (since we picked up only the low latitude region of the Lat-Lon grid); The metric tensors are simple and analytically known (since it is defined based on the spherical polar coordinates); Routines that involve only individual component grid can be recycled for two times (since Yin and Yang are identical); Routines that involve its counterpart component can also be recycled for two times (since Yin and Yang are complemental); And, finally; It suits to massively parallel computers (since the domain decomposition is straightforward).

Acknowledgements

I would like to thank Drs. Masaki Yoshida and Mamoru Hyodo for reading the manuscript.

(This article is reviewed by Dr. Takesi Yukutake.)

References

[1] A. Kageyama and T. Sato, The "Yin-Yang Grid": An overset grid in spherical geometry, *Geochem. Geophys. Geosyst.*, vol.**5**, no.9, doi:10.1029/2004GC000734, 2004.

[2] A. Kageyama, M. Kameyama, S. Fujihara, M. Yoshida, M. Hyodo, and Y. Tsuda, A 15.2 Tflops simulation of geodynamo on the Earth Simulator, *Proc. Supercomputing Conference 2004, SC2004*, ACM/IEEE, Pitssburgh, USA, Nov., 2004.

[3] A. Kageyama, Yin-Yang grid and geodynamo simulation, *Proc. Third M.I.T. Conference on Computational Fluid and Solid Mechanics*, MIT, Cambridge, USA, Jun., 2005 (in press).

[4] M. Yoshida and A. Kageyama, Application of the Yin-Yang grid to a thermal convection of a Boussinesq fluid with infinite Prandtl number in a three-dimensional spherical shell, *Geophys. Res. Lett.*, vol.**31**, no.12, doi:10.1029/2004GL019970, 2004.

[5] K. Takahashi, X. Peng, K. Komine, M. Ohdaira, Y. Abe,

T. Sugimura, K. Hirai, K. Goto, H. Fuchigami, M. Yamada, and K. Watanabe, Development of nonhydrostatic coupled ocean-atmosphere simulation code on the earth simulator. *Proc. 7th International Conference on High Performance Computing and Grid in Asia Pacific Region*, IEEE Computer Society, pp.487–494, Omiya, Japan, Jul., 2004.

[6] K. Komine, K. Takahashi, and K. Watanabe, Development of a global non-hydrostatic simulation code using yin-yang grid system, *Proc. 2004 workshop on the solution of partial differential equations on the sphere*, JAMSTEC, pp.67–69, Yokohama, Japan, Jul., 2004.

[7] C. Ronchi, R. Iacono, and P. S. Paolucci, The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry, *J. Comput. Phys.*, vol.**124**, pp.93–114, 1996.

[8] G. Chesshire and W. D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, *J. Comput. Phys.*, vol.**90**, pp.1–64, 1990.

[9] J. L. Steger, F. C. Dougherty, and J. A. Benek, A Chimera grid scheme, *in Advances in Grid Generation*, edited by K.N. Ghia and U. Ghia, pp.59–69, 1983.

[10] J. L. Steger, On application of body conforming curvilinear grids for finite difference solution of external flow, *in Numerical Grid Generation, J.F. Thomposon, ed.*, North-Holland, New York, pp.295–316, 1982.

[11] R. L. Meakin, Computations of the unsteady flow about a generic wing/pylon/finned-store configurations, *AIAA*, 92-4568-CP, pp.564–580, 1992.

[12] R. L. Meakin, Moving body overset grid methods for complete aircraft tiltrotor simulations, *AIAA*, 93-3350-CP, pp.576–588, 1993.

[13] H. V. Cao, T. Y. Su, and S. E. Rogers, Navier-stokes analysis of a 747 high lift configuration, *AIAA*, 98-2623, pp.402–409, 1998.

[14] S. E. Rogers, H. V. Cao, and T. Y. Su, Grid generation for complex high-lift configurations, *AIAA*, 98-3011, pp.1–11, 1998.

[15] P. G. Buning, I. T. Chiu, S. Obayashi, Y. M. Rizk, and J. L. Steger, Numerical simulation of the integrated space shuttle vehicle in ascent, *AIAA Paper*, 88-4359-Cp, pp.265–283, 1988.

[16] E. P. N. Duque, R. C. Strawn, J. Ahmad, and R. Biswas, An overset grid Navier-Stokes Kirchhoff-surface method for rotorcraft aeroacoustic predictions, *AIAA*, 96-0152, pp.1–13, 1996.

[17] N. A. Phillips, A map projection system suitable for large-scale numerical weather prediction, *J. Meteor. Soc. Japan*, 75th Anniversary Volume, pp.262–267, 1957.

[18] N. A. Phillips, Numerical integration of the primitive equations on the hemisphere, *Month. Weather Rev.*, vol.**87**, pp.333–345, 1959.

[19] G. L. Browning, J. J. Hack, and P. N. Swarztrauber, A comparison of three numerical methods for solving differential equations on the sphere, *Month. Weath. Rev.*, vol.**117**, pp.1058–1075, 1989.

[20] J. W. Hernlund and P. J. Tackley, Three-dimensional spherical shell convection at infinite prandtl number using the 'cubed sphere' method, *Proc. the Second MIT Conference on Computational Fluid and Solid Mechanics*, MIT, Cambridge, USA, Jun., 2003.

[21] J. Dudhia and J. F. Bresch, A global version of the PSU-NCAR mesoscale model, *Month. Weather Rev.*, vol.**130**, pp.2989–3007, 2002.