Chapter 2
Basic Equations and Development of Numerical Code

Abstract In this chapter, we show our development of numerical code. The detailed setting for calculations in the solar convection zone is introduced. The near-surface region ($>0.98R_\odot$) is included for the solar global convection calculation for the first time. Our new challenge for including the partial ionization effect of Hydrogen and Helium is explained. In order to deal with large number of CPUs in the huge parallel computer, we adopt efficient parallelizaion method, Peano-Hilbert space-filling curve and Yin-Yang grid. These are explained. Finally the method for analyzing the huge size data outputted by the calculations are introduced.

Keywords Yin-Yang grid · Artificial viscosity · Managing huge data · Equation of state including partial ionization

2.1 Model Setting

2.1.1 Equations

We solve the three-dimensional magnetohydrodynamic equations with the RSST in the spherical geometry (r, θ, ϕ) as follows:

$$\frac{\partial \rho_1}{\partial t} = -\frac{1}{\xi^2}\nabla \cdot (\rho_0 \mathbf{v}), \tag{2.1}$$

$$\rho_0 \frac{\partial \mathbf{v}}{\partial t} = -\rho_0(\mathbf{v}\cdot\nabla)\mathbf{v} - \nabla\left(p_1 + \frac{B^2}{8\pi}\right) + \nabla\cdot\left(\frac{\mathbf{BB}}{4\pi}\right) - \rho_1 g \mathbf{e_r} + 2\rho_0 \mathbf{v}\times\Omega_0, \tag{2.2}$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla\times(\mathbf{v}\times\mathbf{B}), \tag{2.3}$$

$$\rho_0 T_0 \frac{\partial s_1}{\partial t} = -\rho_0 T_0 (\mathbf{v}\cdot\nabla)s_1 + \frac{1}{r^2}\frac{d}{dr}\left(r^2 \kappa_\mathrm{r}\rho_0 c_\mathrm{p}\frac{dT_0}{dr}\right) + \Gamma, \tag{2.4}$$

where ρ, p, s, T, $\mathbf{v}$, and $\mathbf{B}$ are the density, gas pressure, specific entropy, temperature, fluid velocity, and magnetic field, respectively. Subscript 1 denotes the fluctuation

H. Hotta, *Thermal Convection, Magnetic Field, and Differential Rotation in Solar-type Stars*, Springer Theses, DOI 10.1007/978-4-431-55399-1_2

from the time-independent spherically symmetric reference state, which has subscript 0. g, κ_r, and Γ are the gravitational acceleration, the coefficient of the radiative diffusivity, and the surface cooling term, respectively. Ω_0 is the rotating vector of the rotating frame. The setting of these values is explained in the following paragraph. The equation of state required to close the MHD system is explained in Sect. 2.1.5.

Note that we do not have any explicit turbulent thermal diffusivity and viscosity (Miesch et al. 2000) to maximize the fluid and magnetic Reynolds number but use the artificial viscosity introduced in Rempel et al. (2009). This method is explained in Sect. 2.2.2.

2.1.2 Background Stratification and Radiation

Figure 2.1 shows the reference state used in this study in comparison with Model S (Christensen-Dalsgaard et al. 1996). The reference stratification is determined by solving the one-dimensional hydrostatic equation and the realistic equation of state

$$\frac{dp_0}{dr} = -\rho_0 g, \tag{2.5}$$

$$\rho_0 = \rho_0(p_0, s_0), \tag{2.6}$$

$$\frac{ds_0}{dr} = 0. \tag{2.7}$$

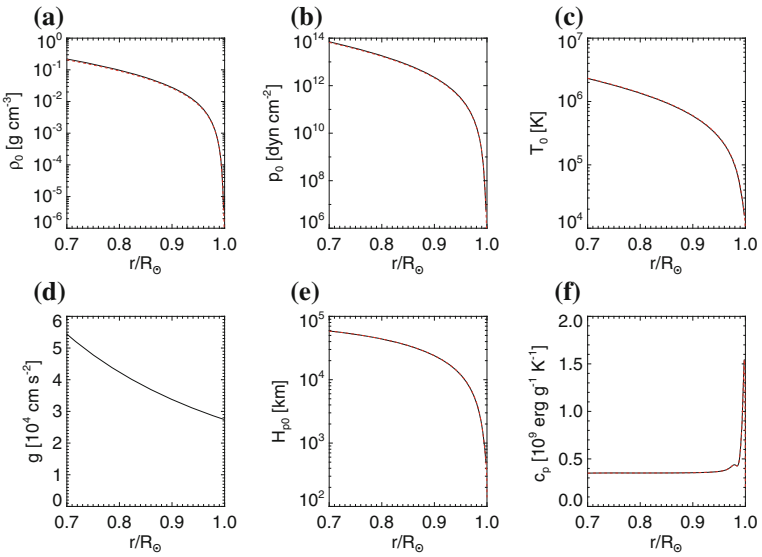

Fig. 2.1 The values at the reference state for **a** density, **b** gas pressure, **c** temperature, **d** gravitational acceleration, **e** pressure scale height, **f** heat capacity at constant pressure. The *black* and *red lines* show the reference state in this study and the values from Model S, respectively. The value of the gravitational acceleration in this study is exactly the same as Model S (Color figure online)

Equation (2.6) is calculated with the OPAL repository including partial ionization. The adiabatic stratification is set as the reference state and initial state. The stratification becomes superadiabatic after the development of convection as a consequence of radiative heating near the bottom and radiative cooling at the top as described below. The gravitational acceleration and the radiative diffusion are adopted from Model S. The boundary is set at $r = 0.998R_\odot$ with the values from Model S and the equations are integrated inward with the fourth-order Runge-Kutta method. A total of 3,072 grid points are used in the integration and stored. For each of the time-dependent calculations in this study, the interpolated values to each grid (i.e., variables with subscript 0 in Eqs. (2.1) and (2.4)) are from the stored data.

In the real sun, the surface is continuously cooled by radiation. Because the adopted boundary is not located at the real solar surface even though it is unprecedentedly closer to the real surface, we add artificial cooling (Γ) in Eq. (2.4)

$$\Gamma(r) = -\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 F_\mathrm{s}), \tag{2.8}$$

$$r^2 F_\mathrm{s}(r) = r_\mathrm{min}^2 F_\mathrm{r}(r_\mathrm{min}) \exp\left[-\left(\frac{r - r_\mathrm{max}}{d_\mathrm{c}}\right)^2\right], \tag{2.9}$$

$$F_\mathrm{r}(r) = -\kappa_\mathrm{r}\rho_0 c_\mathrm{p}\frac{dT_0}{dr}, \tag{2.10}$$

where r_min and r_max denote the location of the bottom boundary and top boundary, respectively. This procedure ensures that the radiative luminosity inputted from the bottom is released through the top boundary. The realistic simulation for the near surface layer shows that the thickness of the cooling layer by radiation is similar to the local pressure scale height (e.g. Stein et al. 2009). Typically, two pressure scale heights are adopted for the thickness of the cooling layer, i.e., $d_\mathrm{c} = 2H_{p0}(r_\mathrm{max})$, where $H_{p0} = p_0/(\rho_0 g)$ is the pressure scale height.

2.1.3 Setting for RSST

In this thesis, the factor of the RSST is set to make the adiabatic reduced speed of sound uniform in space. The adiabatic speed of sound is defined as:

$$c_\mathrm{s}(r) = \sqrt{\left(\frac{\partial p}{\partial \rho}\right)_s}. \tag{2.11}$$

Then the factor of the RSST is set as

$$\xi(r) = \xi_0 \frac{c_\mathrm{s}(r)}{c_\mathrm{s}(r_\mathrm{min})}. \tag{2.12}$$

In this thesis, we adopt $\xi_0 = 120$ for calculations in Chap. 3. Thus, the reduced speed of sound $\hat{c}_s \equiv c_s/\xi = 1.88$ km s^{-1} at all depths. Hotta et al. (2012) suggested that the RSST is valid for thermal convection under the criterion of $v_{rms}/\hat{c}_s < 0.7$, where v_{rms} is the RMS (root mean square) convection velocity. Thus, the we can properly treat the convection with $v_{rms} < 1.3$ km s^{-1} in this model. In Chap. 4, since $\xi = 200$ is adopted, the valid convection speed is less than 0.79 km s^{-1}. Hotta et al. (2012) suggest that the total mass is not conserved with inhomogeneous ξ and long-term drift is not avoided from the reference state, i.e., mass continuously decreases or increases. In this study, however, we adopt different way to avoid this type of long-term drift. When the equation of continuity is treated as

$$\frac{\partial}{\partial t}\left(\xi^2\rho_1\right) = -\nabla\cdot(\rho_0\mathbf{v})\,, \tag{2.13}$$

the value $\hat{M}$ is conserved in the rounding error with appropriate boundary conditions, in which

$$\hat{M} = \int_V \xi^2\rho_1 dV. \tag{2.14}$$

Although the radial distribution of the density is different from the original, the fluctuation remains small (e.g., $\rho_1/\rho_0 \sim 10^{-6}$) and does not affect the character of the thermal convection. Hotta et al. (2012) confirmed that the statistical features are not influenced by the inhomogeneous ξ.

2.1.4 Divergence Free Condition for Magnetic Field

The divergence free condition, i.e. $\nabla\cdot\mathbf{B} = 0$, is maintained with the diffusion scheme for each Runge-Kutta loop. This was also introduced by Rempel et al. (2009). The original idea for this is that when the diffusion equation for the divergence of magnetic field with an appropriate boundary condition, i.e., $\nabla\cdot\mathbf{B} = 0$, the numerical generated divergence error disappears. Thus the following equation is intended to be solved

$$\frac{\partial}{\partial t}(\nabla\cdot\mathbf{B}) = \nabla\cdot[\mu\nabla(\nabla\cdot\mathbf{B})], \tag{2.15}$$

where μ is the diffusion coefficient and adopted as large as possible for the CFL condition. In the numerical calculation, the equation:

$$\frac{\partial\mathbf{B}}{\partial t} = \mu\nabla(\nabla\cdot\mathbf{B}), \tag{2.16}$$

is solved and is mathematically identical with Eq. (2.15).

2.1.5 Equation of State

Since our upper boundary is at $r = 0.99R_\odot$ at maximum, in the near-surface region partial ionization is important (see Fig. 2.1f) and is included in our treatment by using the OPAL repository with solar abundances of $X = 0.75$, $Y = 0.23$, and $Z = 0.02$, where X, Y, and Z specify the mass fraction of hydrogen, helium and other heavy elements, respectively.

In the numerical simulations of the convection in the near surface layer, the ordinary tabular equation of state is widely used (Vögler et al. 2005; Rempel et al. 2009). However, this is not a good approach in our current simulations, because the deviations from the reference state are small, e.g. $\rho_1/\rho_0 \sim 10^{-6}$ around the base of the convection zone. Thus we adopt another way to treat the ionization effect in the near surface layer. The fluctuations from the reference state are calculated as

$$p_1 = \left(\frac{\partial p}{\partial \rho}\right)_s \rho_1 + \left(\frac{\partial p}{\partial s}\right)_\rho s_1, \tag{2.17}$$

$$T_1 = \left(\frac{\partial T}{\partial \rho}\right)_p \rho_1 + \left(\frac{\partial T}{\partial p}\right)_\rho p_1, \tag{2.18}$$

$$e_1 = \left(\frac{\partial e}{\partial \rho}\right)_T \rho_1 + \left(\frac{\partial e}{\partial T}\right)_\rho T_1, \tag{2.19}$$

where e is the internal energy. The first derivatives, such as $(\partial p/\partial \rho)_s$, are described by the background variables, $\rho_0(r), p_0(r) \ldots$ and are regarded as functions of depth r. In the OPAL routine (Rogers et al. 1996), the values $(\partial e/\partial \rho)_T$, $(\partial e/\partial T)_\rho$, $(\partial p/\log \rho)_T$ and $(\partial p/\log T)_\rho$ are provided for given ρ_0, T_0, and the mass fractions of hydrogen (X), helium (Y) and other heavy elements (Z). The relations between the OPAL-provided variables and the required variables derived from the first law of thermodynamics are (Mihalas and Mihalas 1984):

$$\left(\frac{\partial p}{\partial \rho}\right)_s = \frac{c_\mathrm{p}}{\kappa_\mathrm{t}\rho_0 c_\mathrm{v}}, \tag{2.20}$$

$$\left(\frac{\partial p}{\partial s}\right)_\rho = \frac{\beta_\mathrm{p} T_0}{\kappa_\mathrm{t} c_\mathrm{v}}, \tag{2.21}$$

$$\left(\frac{\partial T}{\partial \rho}\right)_p = -\frac{1}{\rho_0 \beta_\mathrm{p}}, \tag{2.22}$$

$$\left(\frac{\partial T}{\partial p}\right)_\rho = T_0 \Big/ \left(\frac{\partial p}{\partial \log T}\right)_\rho, \tag{2.23}$$

where β_p, c_v, c_p and κ_t are the coefficient of thermal expansion, the specific heat at constant volume and pressure, and the coefficient of isothermal compressibility, respectively and they are defined as follows:

$$\beta_{\rm p} = -\left(\frac{\partial \log \rho}{\partial T}\right)_p = \frac{1}{T_0}\left(\frac{\partial p}{\partial \log T}\right)_\rho \Big/ \left(\frac{\partial p}{\partial \log \rho}\right)_T, \tag{2.24}$$

$$c_{\rm v} = \left(\frac{\partial e}{\partial T}\right)_\rho, \tag{2.25}$$

$$c_{\rm p} = c_{\rm v} - T_0\beta_{\rm p}\left[\left(\frac{\partial e}{\partial \rho}\right)_T - \left(\frac{p_0}{\rho_0^2}\right)\right], \tag{2.26}$$

$$\kappa_{\rm t} = \left(\frac{\partial \log \rho}{\partial p}\right)_T. \tag{2.27}$$

2.2 Numerical Method

2.2.1 Space Derivative and Time Integration

The numerical method used in this thesis is the same as that in the MURaM code (Vögler et al. 2005). We use the fourth-order space-centered difference for each derivative. The first spatial derivatives of quantity u are given by

$$\left(\frac{\partial u}{\partial x}\right)_i = \frac{1}{12\Delta x}\left(-u_{i+2} + 8u_{i+1} - 8u_{i-1} + u_{i-2}\right), \tag{2.28}$$

where i denotes the index of the grid position along a particular spatial direction. The numerical solution of the system is advanced in time with an explicit fourth-order Runge-Kutta scheme. The system of partial equations is written as

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{R}(\mathbf{U}), \tag{2.29}$$

for $\mathbf{U}_{n+1}$, which is the value at $t_{n+1} = (n+1)\Delta t$. This is calculated in four steps as

$$\mathbf{U}_{n+\frac{1}{4}} = \mathbf{U}_n + \frac{\Delta t}{4}\mathbf{R}\left(\mathbf{U}_n\right), \tag{2.30}$$

$$\mathbf{U}_{n+\frac{1}{3}} = \mathbf{U}_n + \frac{\Delta t}{3}\mathbf{R}\left(\mathbf{U}_{n+\frac{1}{4}}\right), \tag{2.31}$$

$$\mathbf{U}_{n+\frac{1}{2}} = \mathbf{U}_n + \frac{\Delta t}{2}\mathbf{R}\left(\mathbf{U}_{n+\frac{1}{3}}\right), \tag{2.32}$$

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t\mathbf{R}\left(\mathbf{U}_{n+\frac{1}{2}}\right). \tag{2.33}$$

2.2.2 Artificial Viscosity

The artificial viscosity used in this thesis is introduced by Rempel et al. (2009). The defined diffusive flux is

$$F_{i+1/2} = -\frac{1}{2} c_{i+1/2} \phi_{i+1/2} \left(u_\mathrm{r} - u_\mathrm{l}, u_{i+1} - u_i\right) \left(u_\mathrm{r} - u_\mathrm{l}\right), \tag{2.34}$$

$$\phi = \begin{cases} \left[\dfrac{(u_\mathrm{r} - u_\mathrm{l})}{(u_{i+1} - u_i)}\right]^2 & \text{for } (u_\mathrm{r} - u_\mathrm{l}) \cdot (u_{i+1} - u_i) > 0, \\ 0 & \text{for } (u_\mathrm{r} - u_\mathrm{l}) \cdot (u_{i+1} - u_i) \leq 0, \end{cases} \tag{2.35}$$

where $c_{i+1/2} = c_\mathrm{s} + v + c_\mathrm{a}$ is the characteristic velocity which is the sum of the speed of sound (c_s), fluid velocity (v) and the Alfven velocity (c_a). To decrease the effect of viscosity, a multiplier less than unity is sometimes used. In the code, the physical variables u_i are defined at the center of the cell. To calculate the diffusive flux, the variables u_r and u_l at a boundary of the cells are defined as:

$$u_\mathrm{l} = u_i + \frac{1}{2} \Delta u_i, \tag{2.36}$$

$$u_\mathrm{r} = u_{i+1} - \frac{1}{2} \Delta u_{i+1}, \tag{2.37}$$

where the tilt of the variable (Δu_i) is defined as:

$$\Delta u_i = \mathrm{minimod}\left(\varepsilon(u_{i+1} - u_i), \frac{u_{i+1} - u_{i-1}}{2}, \varepsilon(u_i - u_{i-1})\right), \tag{2.38}$$

where ε is the factor for the minimod function ($1 < \varepsilon < 2$).

To conserve total energy, the heat from the dissipated kinetic energy and the magnetic energy should be properly treated. The treatment for ensuring that the dissipated kinetic and magnetic energies with artificial viscosity are converted to internal energy is the following. The equation of motion and the induction equation are expressed as

$$\begin{aligned} \rho_0 \frac{\partial \mathbf{v}}{\partial t} &= [\cdots] - \nabla \cdot [\rho_0 \mathbf{F}_\mathrm{diff}(\mathbf{v})], \\ \frac{\partial \mathbf{B}}{\partial t} &= [\cdots] - \nabla \cdot [\mathbf{F}_\mathrm{diff}(\mathbf{B})], \end{aligned} \tag{2.39}$$

where $\mathbf{F}_\mathrm{diff}$ is the diffusive flux calculated with Eq. (2.35). The heat caused by the artificial viscosity is estimated and added in the equation of entropy as

$$\rho T \frac{Ds}{Dt} = -[\rho_0 \mathbf{F}_\mathrm{diff}(\mathbf{v}) \cdot \nabla] \cdot \mathbf{v} - \frac{1}{4\pi} [\mathbf{F}_\mathrm{diff}(\mathbf{B}) \cdot \nabla] \cdot \mathbf{v}. \tag{2.40}$$

2.2.3 Peano-Hilbert Space Filling Curve for MPI Communication

For parallel computation by using the MPI library, the data for the spatial elements, each of which corresponds to an MPI process, should be loaded to reduce the communication among processes. For example, 64 MPI processes are used to divide the calculation space in two dimensions as $S_x \times S_y = 8 \times 8$, where S_x, and S_y are the number of data elements for the x-, and y-direction, respectively. As is often the case, the communication between neighbors is more effective than that between apart two, the communication in the x-direction is relatively effective, because the neighborhood in the x-direction is also a neighborhood in the MPI process number. Regarding the y-direction, it is not the case. In this case, the MPI processes are first assigned in the x-direction, and after one line is filled in the x-direction, they move to the y-direction. The neighborhood in the y-direction is always apart 8 in MPI process numbers. When the computer system is one-dimensional, this assignment decreases the efficiency of the communication cost. When the MPI process number becomes more than 10^3, this becomes even more problematic.

To avoid this problem, the Peano-Hilbert space-filling curve is adopted for ordering the MPI rank numbers in the numerical code. This type of space filling curve is typically adopted by codes using the adaptive mesh refinement (Matsumoto 2007). The space-filling curve assigns the nodes whose MPI rank is close to the node number. In this section, the algorithm of the three-dimensional Peano-Hilbert space-filling curve is introduced.

First, we define the connection vectors for the first-order Peano-Hilbert curve as:

$$\mathbf{B_2} = (0, 1, 0), \tag{2.41}$$
$$\mathbf{B_3} = (1, 0, 0), \tag{2.42}$$
$$\mathbf{B_4} = (0, -1, 0), \tag{2.43}$$
$$\mathbf{B_5} = (0, 0, 1), \tag{2.44}$$
$$\mathbf{B_6} = (0, 1, 0), \tag{2.45}$$
$$\mathbf{B_7} = (-1, 0, 0), \tag{2.46}$$
$$\mathbf{B_8} = (0, -1, 0). \tag{2.47}$$

The first-order Peano-Hilbert curve has eight points that are connected with the connection vectors as follows:

$$^{\mathbf{1}}\mathbf{P_1} = (0, 0, 0), \tag{2.48}$$
$$^{\mathbf{1}}\mathbf{P_i} = {}^{\mathbf{1}}\mathbf{P_{i-1}} + \mathbf{B_i} \quad \text{for } i = 2, 3, \ldots, 8. \tag{2.49}$$

The first-order Peano-Hilbert curve only consists of connection vectors. The first-order curve is shown in Fig. 2.2a.

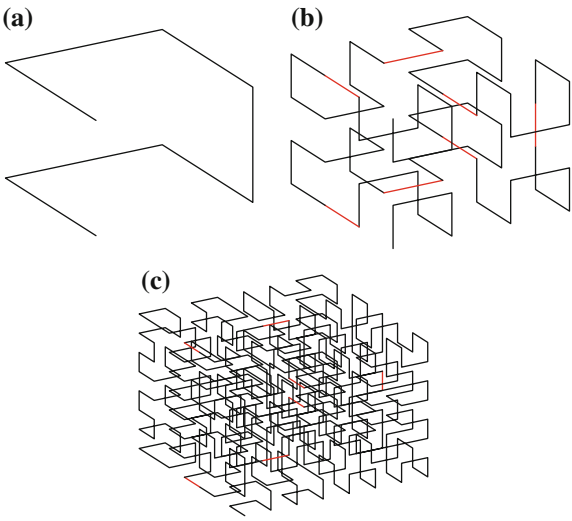

Fig. 2.2 **a** First-order, **b** second-order, and **c** third-order Peano-Hilbert space-filling curve. The connection vectors are shown in *red* (Color figure online)

The second-order Peano-Hilbert curve is made by rotating and connecting the first-order curve. The rotation matrices are defined as

$$\begin{aligned}
\mathbf{A_1} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{A_2} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \\
\mathbf{A_3} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{A_4} = \begin{pmatrix} 0 & 0 & 1 \\ -1 & 0 & 0 \\ 0 & -1 & 0 \end{pmatrix}, \\
\mathbf{A_5} &= \begin{pmatrix} 0 & 0 & -1 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \quad \mathbf{A_6} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\
\mathbf{A_7} &= \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad \mathbf{A_8} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & -1 & 0 \end{pmatrix}.
\end{aligned} \tag{2.50}$$

Then, the eight rotated first-order curves are prepared as

$$\mathbf{^1Q_j^i} = \mathbf{A_i}\,\mathbf{^1P_j}, \tag{2.51}$$

Then the second-order curve is defined by connecting them as follows:

$$\mathbf{^2P_i} = \begin{cases} \mathbf{^1Q_i^1} & \text{for } i = 1, 2, \ldots, 8 \\ \mathbf{^2P_{i-1}} + \mathbf{B_n} & \text{for } m = 1\ (i \neq 1)\,, \\ \mathbf{^2P_{8(n-1)}} + \mathbf{^1Q_m^n} & \text{for others} \end{cases} \tag{2.52}$$

where $m = i \bmod 8$ and $n = (i - m)/8 + 1$. The second order curve is shown in Fig. 2.2b. The connected vector is highlighted in red. Then, the same procedure is repeatedly applied and the higher-order curves are generated. The third-order curve is shown in Fig. 2.2c. We prepare the smallest Peano-Hilbert curve that covers all the nodes.

2.2.4 Yin-Yang Grid

To include the entire spherical shell, we adopt the Yin-Yang grid (Kageyama and Sato 2004). The Yin-Yang grid is a set of two congruent spherical geometries combined in a complementary way to cover the whole spherical shell. The boundary condition for each grid is calculated using the interpolation of the other grid. In all calculation, each grid covers $0.715R_\odot < r < r_{\max}, \pi/4 - \delta_\theta < \theta < 3\pi/4 + \delta_\theta$, and $-3\pi/4 - \delta_\phi < \phi < 3\pi/4 + \delta_\phi$, where δ_θ and δ_ϕ are the margins for interpolation. We use $\delta_\theta = 3\Delta\theta/2$ and $\delta_\phi = 3\Delta\phi/2$ for the interpolation with the third-order function, where $\Delta\theta$ and $\Delta\phi$ are the grid spacings in latitudinal and longitudinal directions. Although the boundary condition for the Yin grid is applied on the edge of the Yin grid, the boundary condition for the Yang grid is applied on the edge of the Yin grid to avoid the double solution in the overlapping area of the Yin-Yang grid (Fig. 2.3). The red thick lines show the location of the horizontal boundary for both Yin and Yang grids.

2.2.5 Big Data Management

The largest number of grid points in this thesis is $512(r) \times 1{,}024(\theta) \times 3{,}072(\phi) \times 2$(Yin Yang). For each output, the data is reorganized from the Yin-Yang grid to the ordinal spherical geometry with $512(r) \times 2{,}048(\theta) \times 4{,}096(\phi)$ grid points. With eight single-precision variables, it costs $512 \times 2{,}048 \times 4{,}096$(grid points) $\times 8$(variables) $\times$ 4(single precision) $\sim$137 GB per time step and $\sim$27 TB for 200 steps. Thus, the treatment of such large number of data is no longer trivial. We adopt two strategies. First, the two-dimensional data in each layer with constant r is output to one file. Second, statistical values are calculated in the supercomputer and only the results that are likely smaller than the original are output. Figure 2.4 summarizes the first strategy. In the calculations, on account of the ordering using the Peano-Hilbert curve, the data distribution is rather complex which makes the analysis difficult. It is almost impossible to construct three-dimensional full data on a personal computer after the data transfer because of the computer memory requirements and reduction time. Thus, by using the first strategy, we can draw a two-dimensional map of the entire sphere with small tasks for download and reproduction. In the second strategy, the required statistical data are almost always the zonal average or at least the horizontal average, which can be generated from the layered data obtained by using the first

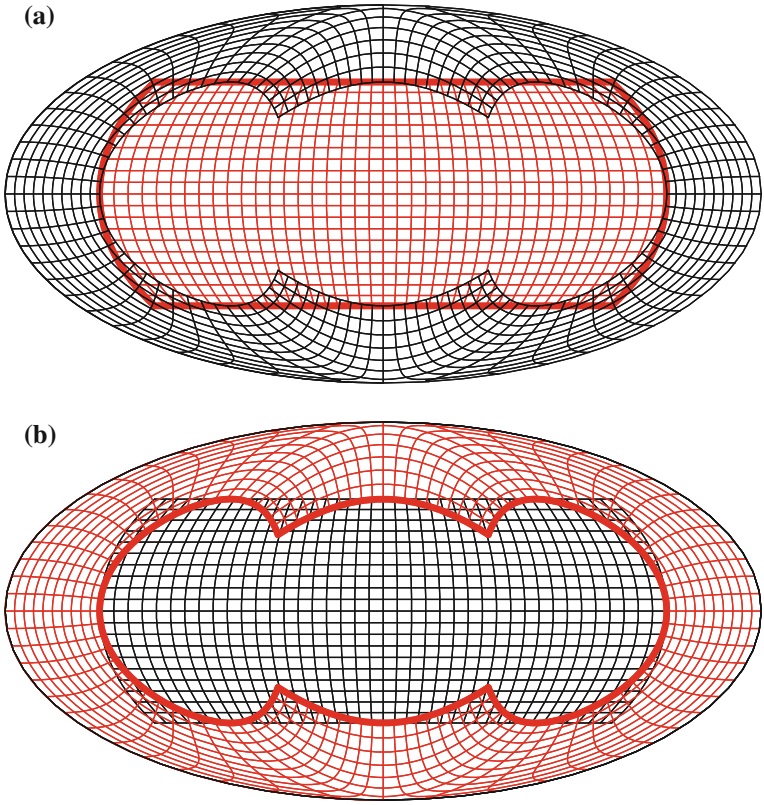

Fig. 2.3 *Red lines* and *Black lines* indicate the Yin and Yang grid, respectively. **a** and **b** shows the geometry on the Mollweide projection in the different view points. The *thick red lines* show the boundaries for both Yin and Yang grids (Color figure online)

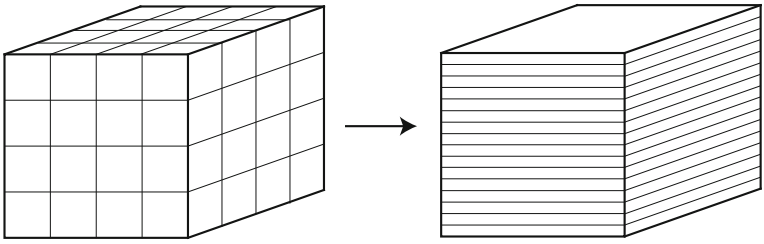

Fig. 2.4 Schematic for the first strategy. The complexly distributed data are reordered for analysis

strategy. We choose the required data in each two-dimensional layer, such as the RMS velocity, the energy flux and so on. This procedure significantly suppresses the required storage, analysis time, and the required memory a personal computer.

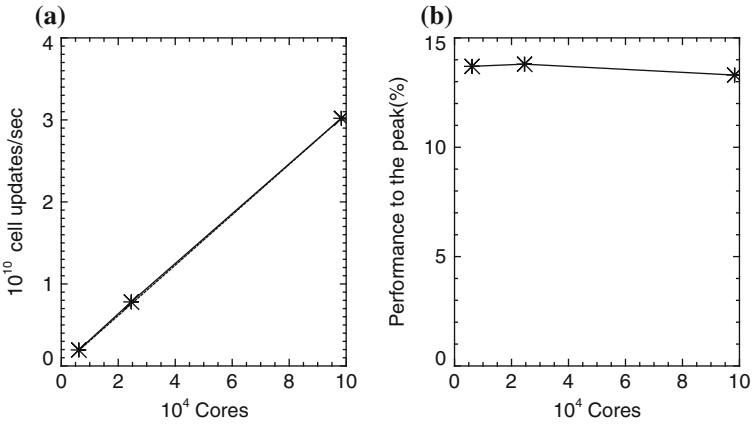

Fig. 2.5 **a** Performance for updating the grid point per second versus the number of cores and **b** peak floating point performance

2.2.6 Code Performance

Using a hybrid MPI and automatic intra-node parallelization approach, and the method explained above, the code efficiently scales up to 10^5 core counts with almost linear weak scaling and achieves 14 % performance at maximum on the RIKEN K-computer in Japan. The performance tests are shown in Fig. 2.5. Because the code includes almost no global communication among cores, this linear scaling is expected to hold further with larger cores. The code performs 3×10^5 grid update/core/s, which allows to investigate the interaction of small-scale and large-scale convection in the spherical shell.

References

J. Christensen-Dalsgaard, W. Dappen, S.V. Ajukov, E.R. Anderson, H.M. Antia, S. Basu, V.A. Baturin, G. Berthomieu, B. Chaboyer, S.M. Chitre, A.N. Cox, P. Demarque, J. Donatowicz, W.A. Dziembowski, M. Gabriel, D.O. Gough, D.B. Guenther, J.A. Guzik, J.W. Harvey, F. Hill, G. Houdek, C.A. Iglesias, A.G. Kosovichev, J.W. Leibacher, P. Morel, C.R. Proffitt, J. Provost, J. Reiter, E.J. Rhodes Jr., F.J. Rogers, I.W. Roxburgh, M.J. Thompson, R.K. Ulrich, The current state of solar modeling. Science **272**, 1286–1292 (1996). doi:10.1126/science.272.5266.1286

H. Hotta, M. Rempel, T. Yokoyama, Y. Iida, Y. Fan, Numerical calculation of convection with reduced speed of sound technique. A&A **539**, 30 (2012). doi:10.1051/0004-6361/201118268

A. Kageyama, T. Sato, "Yin-Yang grid": An overset grid in spherical geometry. Geochem. Geophys. Geosyst. **5**, 9005 (2004). doi:10.1029/2004GC000734

T. Matsumoto, Self-gravitational magnetohydrodynamics with adaptive mesh refinement for protostellar collapse. PASJ **59**, 905 (2007)

M.S. Miesch, J.R. Elliott, J. Toomre, T.L. Clune, G.A. Glatzmaier, P.A. Gilman, Three-dimensional spherical simulations of solar convection. I. Differential rotation and pattern evolution achieved with laminar and turbulent states. ApJ **532**, 593–615 (2000). doi:10.1086/308555

D. Mihalas, B.W. Mihalas, *Foundations of Radiation Hydrodynamics* (Oxford University Press, New York, 1984)

M. Rempel, M. Schüssler, M. Knölker, Radiative magnetohydrodynamic simulation of sunspot structure. ApJ **691**, 640–649 (2009). doi:10.1088/0004-637X/691/1/640

F.J. Rogers, F.J. Swenson, C.A. Iglesias, OPAL equation-of-state tables for astrophysical applications. ApJ **456**, 902 (1996). doi:10.1086/176705

R.F. Stein, Å. Nordlund, D. Georgoviani, D. Benson, W. Schaffenberger, Supergranulation-scale convection simulations, in *Solar-Stellar Dynamos as Revealed by Helio- and Asteroseismology: GONG 2008/SOHO 21*. Astronomical Society of the Pacific Conference Series, vol. 416, ed. by M. Dikpati, T. Arentoft, I. González Hernández, C. Lindsey, F. Hill (2009), p. 421

A. Vögler, S. Shelyag, M. Schüssler, F. Cattaneo, T. Emonet, T. Linde, Simulations of magneto-convection in the solar photosphere. Equations, methods, and results of the MURaM code. A&A **429**, 335–351 (2005). doi:10.1051/0004-6361:20041507

http://www.springer.com/978-4-431-55398-4

Thermal Convection, Magnetic Field, and Differential Rotation in Solar-type Stars

Hotta, H.

2015, XII, 81 p. 49 illus., 11 illus. in color., Hardcover
ISBN: 978-4-431-55398-4