Implementation of the Fourier Transform Method for Geodynamo Simulation on the Earth Simulator

Yusuke Oishi[1]*, Ataru Sakuraba[1], and Yozo Hamano[2]

[1] *Department of Earth and Planetary Science, University of Tokyo,*
7-3-1 Hongo, Bunkyo-ku Tokyo 113-0033, Japan
[2] *Institute for Research on Earth Evolution, Japan Agency for Marine-Earth Science and Technology,*
2-15 Natsushima-cho, Yokosuka 237-0061, Japan

Abstract Most of the computer codes for the recent simulation studies of geodynamo are based on a spectral transform method in which variables are expanded in spherical harmonics. This method, however, is not suitable for high-resolution simulations because of a large amount of computation for spectral transform with respect to Legendre functions. In this study, we developed a modified code based on the Fourier transform method in which the Legendre transform is avoided and the remaining two-dimensional differential equations in the meridional plane are solved by finite differences. The details of the implementation technique of the Fourier transform method on the Earth Simulator are described. The performance tests of the developed code are also reported.

Keywords: Geodynamo, numerical modeling, magnetohydrodynamics, finite difference, multigrid

1. Introduction

In order to elucidate the generation process of the geomagnetic field inside the liquid outer core, computer simulations of convection-driven spherical magnetohydrodynamic dynamos have been commonly carried out since Kageyama et al. [1] and Glatzmaier and Roberts [2] in 1995 (see Kono and Roberts [3] for review). In these simulations, however, the model parameters are considerably different from the realistic values of the Earth's core because of computational difficulty. In particular, the Ekman number, E, which represents the ratio between viscous and Coriolis forces, was 10^{-5} or greater in most of the previous studies, although it is considered to be $10^{-9} \sim 10^{-15}$ in the core. Growing computer performance has made it possible to access Earth-like parameters to some extent. Takahashi et al. [4] recently reported one of the highest-resolution simulations using the Earth Simulator to achieve $E \sim 10^{-6}$, which is still several orders of magnitude greater than the realistic value. In order to investigate the dynamo in the extreme condition compatible to the Earth's core, more efficient numerical methods are required.

In the previous geodynamo simulations, the most widely used method is the spectral transform method, in which the variables are expanded by spherical harmonics [5]. This method (hereafter we call "spherical harmonic transform method" or "SHTM") is beneficial because it enables accurate computation of the spatial derivatives and simple and exact treatment of the magnetic boundary condition between insulating and conducting media. It does not have problems pertaining to the poles of the spherical coordinates. However, the spectral expansion by the Legendre functions causes huge number of calculations when the number of the terms increases, since there is no practical algorithm of a Legendre transform as fast as a Fourier transform. Recent requirements for higher-resolution calculations make it difficult to continue the SHTM.

Some research groups have applied local methods such as finite difference (e.g., [1]) and finite element methods (e.g., [6]) to geodynamo simulations. In particular, Kageyama and Sato [7] recently developed a spherical grid system that prevents the problems pertaining to the poles, and achieved a high calculation speed on the Earth Simulator by using a finite difference method. In general, the local methods require fewer calculations than the spectral method and are also well suited for parallel computation. However, the local methods have drawbacks in

---

* ***Corresponding author***: *Dr. Yusuke Oishi, Department of Earth and Planetary Science, University of Tokyo, Hongo, Bunkyo-ku, Tokyo 113-0033, Japan. E-mail: yoishi@eps.s.u-tokyo.ac.jp*

respect of accuracy and treatment of the magnetic boundary condition. The latter problem could be avoided by meshing the exterior of the core and solving the Laplace equation for a magnetic potential field, although superfluous calculations are inevitable for that.

Oishi et al. [8] introduced the Fourier transform method (FTM) as another possible replacement for the SHTM. The FTM is basically the same as the SHTM, but only the Fourier transform in longitude is performed. The remaining two-dimensional differential equations are solved by a finite difference method. This method can treat the magnetic boundary condition as easily as in the SHTM and the amount of computation is less than the SHTM. Comparison to the SHTM has been also made by Oishi et al. [8] in respect of computational accuracy. In this paper, we focus on some technical issues particularly regarding parallel execution on the Earth Simulator. We describe the simulation model in section 2, and then we present the details of the FTM with special attention to its parallelization and vectorization in section 3. Finally, in section 4, we analyze the parallel computation performance on the Earth Simulator.

2. Model

We consider a spherical shell filled with an electrically conductive Boussinesq fluid and spun around the *z*-axis with an angular velocity Ω. The ratio between the inner radius r_i and the outer radius r_o is set to 0.35. The regions inside and outside the fluid shell are electrically insulating. The inner and outer boundaries are isothermal with a temperature difference ΔT. Rigid boundary conditions are imposed and both boundaries are co-rotating. The gravity changes linearly with radius. Spherical polar coordinates (r, θ, ϕ) rotating with an angular velocity Ωe_z are used, where e_z is the unit vector parallel to the *z*-axis.

The nondimensional governing equations are

$$E \frac{\partial \boldsymbol{u}}{\partial t} = -\nabla(p + \frac{1}{2} E \boldsymbol{u} \cdot \boldsymbol{u}) + RaT \frac{\boldsymbol{r}}{r_o} + E\boldsymbol{u} \times \omega$$
$$+ 2\boldsymbol{u} \times \boldsymbol{e}_z - E\nabla \times \nabla \times \boldsymbol{u} + Pm^{-1} \boldsymbol{J} \times \boldsymbol{B}, \tag{1}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} = \nabla \times (\boldsymbol{u} \times \boldsymbol{B}) - Pm^{-1} \nabla \times \nabla \times \boldsymbol{B}, \tag{2}$$

$$\frac{\partial T}{\partial t} = -\boldsymbol{u} \cdot \nabla T + Pr^{-1} \nabla^2 T, \tag{3}$$

$$\nabla \cdot \boldsymbol{u} = \nabla \cdot \boldsymbol{B} = 0, \tag{4}$$

where $\boldsymbol{u}$, $\boldsymbol{B}$, $\omega = \nabla \times \boldsymbol{u}$, and $\boldsymbol{J} = \nabla \times \boldsymbol{B}$ are velocity, magnetic field, vorticity, and electric current density, respectively. T and p are deviations from the equilibrium state of temperature and pressure. Length is scaled by D, time

by D^2/ν, magnetic field by $(\rho\mu_0\eta\Omega)^{1/2}$, temperature deviation by ΔT, and pressure deviation by $\rho\nu\Omega$, where D denotes the thickness of shell, μ_0 the magnetic permeability, η magnetic diffusivity, ν kinematic viscosity and ρ density. The resulting four non-dimensional parameters are the modified Rayleigh number Ra, the Ekman number E, the magnetic Prandtl number Pm, and the Prandtl number Pr, which are defined by

$$Ra = \frac{\alpha g_o \Delta T D}{\nu\Omega}, \ E = \frac{\nu}{\Omega D^2}, \ Pr = \frac{\nu}{\kappa}, \ Pm = \frac{\nu}{\eta}$$

where α, g_o, and κ are the thermal expansion coefficient, acceleration due to gravity at the outer radius, and thermal diffusivity, respectively.

3. Method and implementation
3.1 Toroidal-poloidal decomposition

As is usually adopted in the SHTM, the magnetic and velocity fields are split into toroidal and poloidal components,

$$\boldsymbol{u} = \nabla \times \nabla \times (U\boldsymbol{r}) + \nabla \times (V\boldsymbol{r}), \tag{5}$$

$$\boldsymbol{B} = \nabla \times \nabla \times (G\boldsymbol{r}) + \nabla \times (H\boldsymbol{r}), \tag{6}$$

where U and G are the defining scalars of the poloidal components and V and H are those of the toroidal ones. By applying $\boldsymbol{r} \cdot$ and $\boldsymbol{r} \cdot \nabla \times$ to the curl of the equation of motion (1) and to the induction equation (2), we obtain

$$\frac{\partial}{\partial t} \nabla^2 \tilde{U} = \nabla^2 \nabla^2 \tilde{U} - \frac{1}{r_0} Ra E^{-1} \mathcal{L}^2 T$$
$$- \boldsymbol{r} \cdot \nabla \times \nabla \times \boldsymbol{W}, \tag{7}$$

$$\frac{\partial}{\partial t} \tilde{V} = \nabla^2 \tilde{V} + \boldsymbol{r} \cdot \nabla \times \boldsymbol{W}, \tag{8}$$

$$\frac{\partial}{\partial t} \tilde{G} = Pm^{-1} \nabla^2 \tilde{G} + \boldsymbol{r} \cdot \nabla \times \boldsymbol{F}, \tag{9}$$

$$\frac{\partial}{\partial t} \widetilde{H} = Pm^{-1} \nabla^2 \widetilde{H} + \boldsymbol{r} \cdot \nabla \times \nabla \times \boldsymbol{F}, \tag{10}$$

where

$$\tilde{U} = \mathcal{L}^2 U, \ \tilde{V} = \mathcal{L}^2 V, \ \tilde{G} = \mathcal{L}^2 G, \ \widetilde{H} = \mathcal{L}^2 H, \tag{11}$$

$$\mathcal{L}^2 = -\frac{1}{\sin\theta} \frac{\partial}{\partial\theta} \left(\sin\theta \frac{\partial}{\partial\theta} \right) - \frac{1}{\sin^2\theta} \frac{\partial^2}{\partial\phi^2}, \tag{12}$$

and $\boldsymbol{W}$ and $\boldsymbol{F}$ are non-linear terms defined as

$$\boldsymbol{W} = \boldsymbol{u} \times \boldsymbol{\omega} + 2E^{-1} \boldsymbol{u} \times \boldsymbol{e}_z + Pm^{-1} E^{-1} \boldsymbol{J} \times \boldsymbol{B}, \tag{13}$$

$$F = \mathbf{u} \times B. \tag{14}$$

3.2 Fourier transform

In the SHTM, the scalar variables *U*, *G*, *V*, *H*, and *T* are expanded in spherical harmonics. For example,

$$U(r,\theta,\phi) = \sum_{l=0}^{L} \sum_{m=-l}^{l} U_l^m(r) P_l^m(\cos\theta) e^{im\phi}, \tag{16}$$

where P_l^m is the associated Legendre function of degree *l* and order *m*, and *L* denotes the cutoff wavenumber in the spectral expansion. In the FTM, on the other hand, we apply only the Fourier expansions

$$U(r,\theta,\phi,t) = \sum_{m=-L}^{L} U_m(r,\theta,t) e^{im\phi}. \tag{17}$$

The resulting spectral equations are

$$\frac{\partial}{\partial t}\tilde{Q}_m = \nabla_m^2 \tilde{Q}_m - \frac{1}{r_o} E^{-1} Ra \mathcal{L}_m^2 T_m$$
$$- \frac{1}{2\pi}\int_0^{2\pi}(\mathbf{r}\cdot\nabla\times\nabla\times\mathbf{W}) e^{-im\phi} d\phi, \tag{18}$$

$$\frac{\partial}{\partial t}\tilde{V}_m = \nabla_m^2 \tilde{V}_m + \frac{1}{2\pi}\int_0^{2\pi}(\mathbf{r}\cdot\nabla\times\mathbf{W}) e^{-im\phi} d\phi, \tag{19}$$

$$\frac{\partial}{\partial t}\tilde{G}_m = Pm^{-1}\nabla_m^2 \tilde{G}_m + \frac{1}{2\pi}\int_0^{2\pi}(\mathbf{r}\cdot\nabla\times\mathbf{F}) e^{-im\phi} d\phi, \tag{20}$$

$$\frac{\partial}{\partial t}\tilde{H}_m = Pm^{-1}\nabla_m^2 \tilde{H}_m + \frac{1}{2\pi}\int_0^{2\pi}(\mathbf{r}\cdot\nabla\times\nabla\times\mathbf{F}) e^{-im\phi} d\phi, \tag{21}$$

$$\frac{\partial}{\partial t}T_m = Pr^{-1}\nabla_m^2 T_m - \frac{1}{2\pi}\int_0^{2\pi}(\mathbf{u}\cdot\nabla T) e^{-im\phi} d\phi, \tag{22}$$

where

$$\tilde{Q}_m = \nabla_m^2 \tilde{U}_m, \tag{23}$$

$$\tilde{U}_m = \mathcal{L}_m^2 U_m, \quad \tilde{V}_m = \mathcal{L}_m^2 V_m, \quad \tilde{G}_m = \mathcal{L}_m^2 G_m, \quad \tilde{H}_m = \mathcal{L}_m^2 H_m, \tag{24}$$

and

$$\nabla_m^2 = \frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) - \frac{1}{r^2}\mathcal{L}_m^2, \tag{25}$$

$$\mathcal{L}_m^2 = -\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta\frac{\partial}{\partial\theta}\right) + \frac{m^2}{\sin^2\theta}. \tag{26}$$

The differential equations in the remaining two-dimensional space (*rθ*-plane) are solved by second-order central differences with equally spaced grid spanning (Fig. 1).
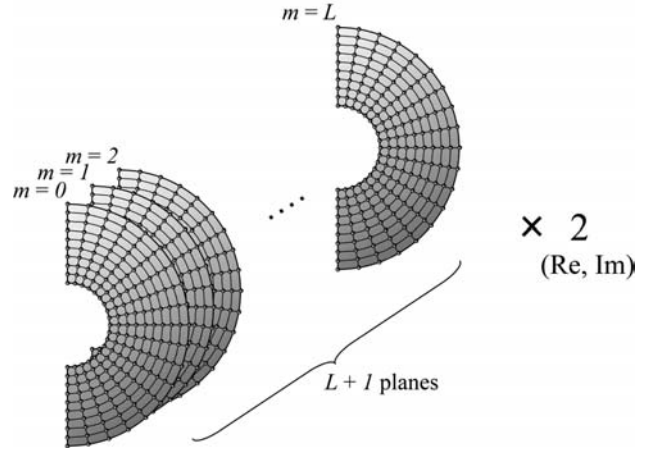

Fig. 1 Grid system in spectral space. Since Fourier coefficients are imaginary numbers, we solve a differntial equation on 2(*L* + 1) meridional planes.

The time integration is performed in spectral space with the third-order Runge-Kutta scheme. All the operations in spectral space do not couple different components of the Fourier expansion. This independence is utilized for the vectorization in spectral space in our model.

3.3 Non-linear terms

Non-linear terms are calculated in real space by the following processes: (a) $\tilde{U}_m$ is obtained by solving the two-dimensional Poisson equation (23) via the multigrid method; (b) U_m, V_m, G_m, and H_m are obtained by solving the onedimensional Poisson equations (24) via the cyclic reduction algorithm; (c) $\mathbf{u}$, B, and *T* are obtained in real space by calculating

$$u_r = \sum_{m=-L}^{L} \frac{1}{r}\tilde{U}_m e^{im\phi}, \tag{27}$$

$$u_\theta = \sum_{m=-L}^{L}\left[\frac{1}{r}\frac{\partial}{\partial\theta}\frac{\partial}{\partial r}(rU_m) + \frac{im}{\sin\theta}V_m\right]e^{im\phi}, \tag{28}$$

$$u_\phi = \sum_{m=-L}^{L}\left[\frac{im}{r\sin\theta}\frac{\partial}{\partial r}(rU_m) - \frac{\partial V_m}{\partial\theta}\right]e^{im\phi}, \tag{29}$$

$$B_r = \sum_{m=-L}^{L} \frac{1}{r}\tilde{G}_m e^{im\phi}, \tag{30}$$

$$B_\theta = \sum_{m=-L}^{L}\left[\frac{1}{r}\frac{\partial}{\partial\theta}\frac{\partial}{\partial r}(rG_m) + \frac{im}{\sin\theta}H_m\right]e^{im\phi}, \tag{31}$$

$$B_\phi = \sum_{m=-L}^{L}\left[\frac{im}{r\sin\theta}\frac{\partial}{\partial r}(rG_m) - \frac{\partial H_m}{\partial\theta}\right]e^{im\phi}, \tag{32}$$
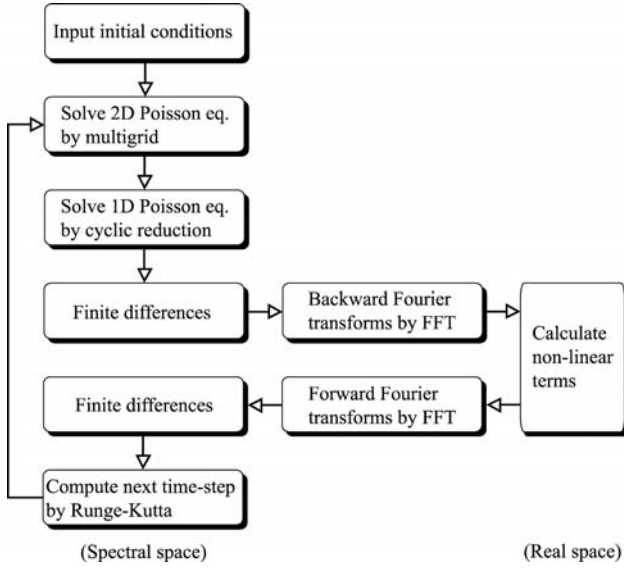
Fig. 2 Flowchart of the Fourier transform method.

Fig. 3 The grid interval in the azimuthal direction corresponding to Fourier expansion truncated up to L' is plotted as a function of the colatitude (see Equations (33)–(35)). When L' does not depend on θ, the grid interval is expressed as $2\pi\sin\theta/L$, as shown by a dash-dotted line. When L' changes in proportion to $\sin\theta$, the interval does not depend on θ (a dashed line). Here, $L = 64$ and $C = 3$ are assumed.

by the finite differences and backward discrete Fourier transforms; (d) F and W are calculated by (14); and (e) the nonlinear terms (Fourier integrals) are obtained by using the forward discrete Fourier transforms and finite differences. Discrete Fourier transforms are performed by the Fast Fourier transform (FFT). Note that the number of grid points in longitude N_ϕ in the real space is set to $3(L + 1)$ in order to eliminate aliasing errors. All the processes in this method are summarized in Fig. 2. After calculating the non-linear terms, the linear terms are calculated by using the finite differences and the next time-step is computed.

3.4 Pole problems

As we use spherical coordinates, some numerical problems arise in treating the poles. Singularity at the pole can be easily avoided by the L'Hospidal's rule (e.g., [1]). The Courant-Friedrichs-Levy (CFL) condition is more serious problem in time integration because many grid points gather at the pole. To circumvent this difficulty, we employ the cutoff wavenumber L' at each colatitude θ as

$$L' = 0 \ (\theta = 0, \pi), \tag{33}$$

$$L' = [CL\sin\theta] \ (0 < \sin\theta < 1/C), \tag{34}$$

$$L' = L \ (\sin\theta > 1/C), \tag{35}$$

where $C > 1$ is a constant to make the resolution near the pole comparable to that of lower latitudes and $[x]$ represents the minimum integer not smaller than x. Fig. 3 illustrates the azimuthal grid width corresponding to the Fourier expansion truncated at L' as a function of the colatitude. We used $C = 3$ in numerical simulations described in Section 4.
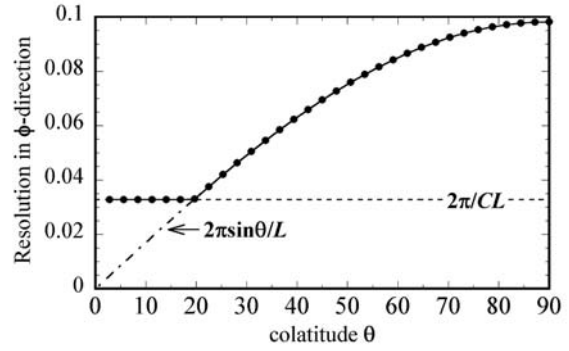
3.5 Magnetic boundary condition

The boundary conditions of the poloidal magnetic field between the conducting fluid shell and the insulators outside can be expressed in terms of the spherical harmonics component of the defining scalar G_l^m as

$$\left.\frac{dG_l^m}{dr}\right|_{r = r_o} = -\frac{l+1}{r_o} G_l^m(r_o), \quad \left.\frac{dG_l^m}{dr}\right|_{r = r_i} = \frac{l}{r_i} G_l^m(r_i). \tag{36}$$

The relation between G_m and G_l^m is

$$G_m(r, \theta_j) = \sum_{l=m}^{L} G_l^m(r) P_l^m(\cos\theta_j), \tag{37}$$

where θ_j is the latitude at a grid point. From the orthogonality of the Legendre functions we have

$$G_l^m(r) = \sum_{k=0}^{N_\theta} G_m(r, \theta_k) P_l^m(\cos\theta_k) \sin\theta_k \Delta\theta, \tag{38}$$

where N_θ is the number of grid points in the θ-direction. On differentiating (37) by r and substituting (36) and (38) into (37), we obtain

$$\frac{\partial}{\partial r} G_m(r, \theta_j) = \sum_{k=0}^{N_\theta} A_{jk} G_m(r, \theta_k) \tag{39}$$

where $[A_{jk}]$ is the coefficient matrix, whose boundary values at $r = r_o$ and r_i are estimated by

$$A_{jk} = \begin{cases} -\displaystyle\sum_{l=m}^{L} \frac{l+1}{r_o} P_l^m(\cos\theta_k) P_l^m(\cos\theta_j) \sin\theta_k \Delta\theta \ (r = r_o), \\ \displaystyle\sum_{l=m}^{L} \frac{l}{r_i} P_l^m(\cos\theta_k) P_l^m(\cos\theta_j) \sin\theta_k \Delta\theta \quad (r = r_i). \end{cases} \tag{40}$$

These coefficient matrices can be prepared before time integration. Then from (39) we obtain the magnetic field at the boundaries that satisfies the boundary conditions. Since evaluation of (39) is required only at the boundaries, the additional amount of computation is not so significant.

3.6 Parallelization

Parallelization is performed by the two-dimensional domain decomposition on the $r\theta$-plane to avoid global communication between distributed memories, which is required when the parallelization in the azimuthal wavenumber is employed. All the data transfer is handled by MPI (flat-MPI). Vectorization is applied to the do-loop with respect to the azimuthal wavenumber in spectral space to ensure the vector length of $2L$, where the factor of 2 arises from the real and imaginary parts of the complex coefficients of the Fourier expansion.

The data transfer is needed to manipulate the finite differences and to solve the one- and two-dimensional Poisson equations. Since we employ the secondorder central differences, it is enough to transfer the data adjacent to the domain boundary. To solve the one-dimensional Poisson equations, we employ a parallelized cyclic reduction algorithm [9]. In this algorithm, communications are required only when reduced tridiagonal systems, whose dimension is in proportion to $N_{p\theta}$, are solved, where $N_{p\theta}$ is the number of processors in the θ-direction. The number of synchronizations required in these communications is proportional to $\log N_{p\theta}$.

The two-dimensional Poisson equations are solved by the V-cycle multigrid method (Fig. 4) based on the red-black Gauss-Seidel relaxation, full weighting restriction, and bilinear interpolation. The communications can be performed in a similar way to those in the finite differences when the grid is not so coarse. Suppose that the meridional plane is approximated by the grid of $N_{pr} \times N_{p\theta}$, where N_{pr} is the number of processors in the r-direction, and that each processor contains only one data. Since the grid at this stage is, in most cases, too fine to be treated by one processor, the data distributed to adjacent four processors are gathered in one of the four processors to form 2×2 grid. Then the new linear system is solved in parallel between the $N_{pr} \times N_{p\theta}/4$ processors. This gathering procedure is repeated until the grid becomes coarse enough to be treated by one processor.

In the gathering procedure, we perform bidirectional data transfers in order that the adjacent four processors have the same 2×2 data, by which no data transfers are needed in the (coarse-to-fine) prolongation procedures until the system returns to the $N_{pr} \times N_{p\theta}$ grid. If a three-dimensional Poisson equation is solved in parallel or the number of the processors is not so large, it may be effec-

Fig. 4 Structure of the V-cycle to solve a two-dimensional Poisson equation. Here, we suppose that the size of the problem is $(N_r, N_\theta) = (64, 64)$ and 16 processors are used in parallel. The shaded spheres denote Gauss-Seidel smoothing. The downward- and upward-sloping single lines between the spheres denote restrictions and prolongations, respectively. The double lines indicate restrictions with the data gathering procedures. The total grid size (N_r, N_θ) and the grid size of a subdomain assigned to one processor (M_r, M_θ) are given in the right-hand side.

tive to gather all the data at one processor once the system becomes sufficiently coarse. In the Earth Simulator, this approach is adopted by Kameyama [10] in his mantle simulation code.
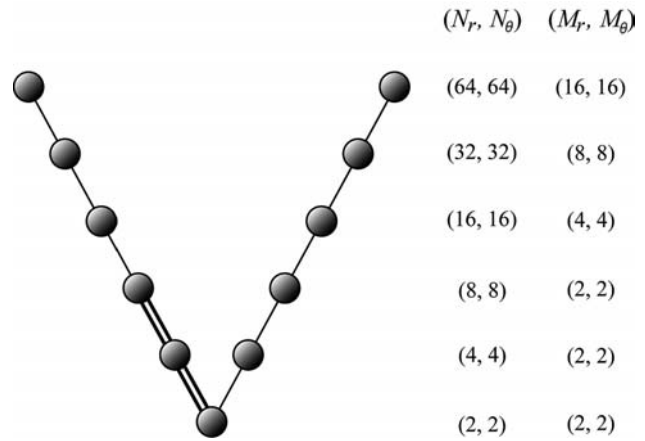
4. Numerical tests

Oishi et al. [8] performed verification tests of the FTM by comparing with the SHTM and ensured that the FTM can simulate self-exciting dynamos in accordance with the SHTM. Here, we show an example of solution obtained by the FTM in Fig. 5, which is the well-known dynamo benchmark problem [11]. Dipole dominant magnetic fields with four sets of magnetic flux concentrations at mid-latitude and four sets of columnar convective cells are observed. The characteristics of the solution are consistent with the solution provided by the SHTM in Christensen et al. [11].

The parallel computation performance of the FTM is investigated on the Earth simulator. The resolution examined is $(N_r, N_\theta, L) = (256, 512, 256)$. The number of multigrid cycles is restricted to 2 here, though it depends on convergence character of the problem to be solved. The results are listed in Table 1. When 256 processors are used, the calculation speed reaches 634 GFLOPS, which is 31% of the peak performance. The vector operation ratio is 99.24%. The most time-consuming part is the FFT in this case. However, when a large number of multigrid cycles are required, the multigrid method could
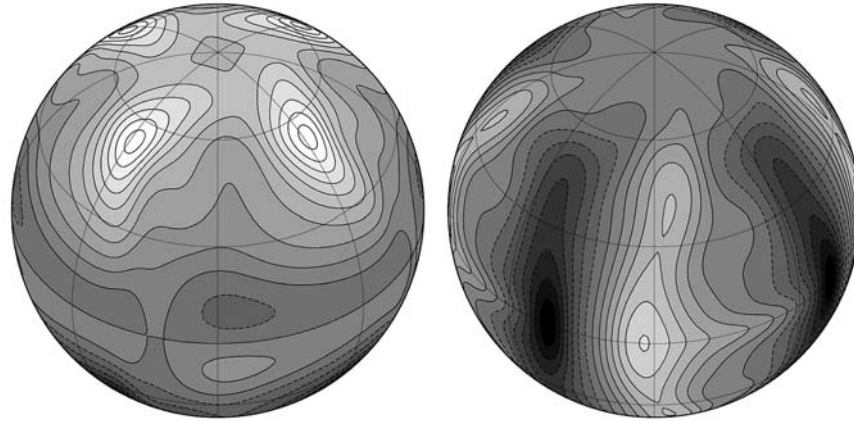
Fig. 5 Numerical solution of dynamo benchmark problem (Case 1). Left is B_r at the outer boundary with a contour interval 0.25, and right is u_r at mid-depth with a contour interval 2. Positive and zero contours are denoted by solid lines and negative contours by dashed lines.
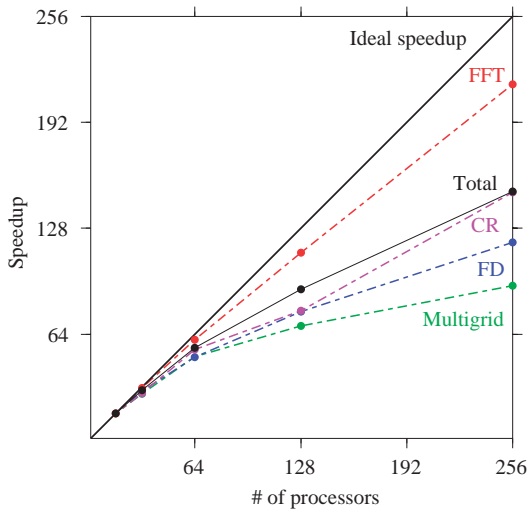

Fig. 6 Speedup of each subroutine. FFT: the operations for FFT; FD: the operations for finite differences and non-linear terms; Multigrid: the operations for the multigrid method; CR: the operations for the cyclic reduction.

become the dominant part.

The speedups (measured on the basis of calculation time on 16 processors, T_{16}, and defined by $16T_{16}/T_p$, where T_p denotes the computation time on the examined number of processors) of total and each computation process are shown in Fig. 6. The FFT is effectively parallelized, because it does not include any communication and the MFLOPS value is not so much affected by the number of processors as shown in Table 1. The multigrid method, on the other hand, shows relatively poor efficiency because communications between processors are iteratively required. To make matters worse, the averaged problem size solved on one processor is smaller than other operations, because coarser grid systems must be solved in the multigrid algorithm. These shortcomings concerned with the communications in the multigrid method can be avoided by employing parallelization about the azimuthal wavenumber m. However, in that case, global all-to-all

Table. 1 Parallel performances in solving 100 time-steps. N_p: number of processors; COMM.TIME: communication time; MFLOPS: Million Floating-point Operations Per Second; V.OP RATIO: vector operation ratio. Total: total of all the operations; FFT: the operations for the FFT; FD & nonlinear: the operations for finite differences and the non-linear terms; Multigrid: the operations for the multigrid method; CR: the operation for the cyclic reduction.

	EXCLUSIVE TIME [sec]					COMM. TIME [sec]				
N_p	Total	FFT	FD & nonlinear	Multigrid	CR	Total	FFT	FD & nonlinear	Multigrid	CR
16	452.4	244.8	100.0	77.9	29.7	18.6	0.0	8.7	9.1	0.9
32	239.5	124.6	53.7	44.3	16.9	17.3	0.0	8.3	7.7	1.3
64	130.2	64.5	32.2	24.8	8.7	13.1	0.0	6.6	5.7	0.8
128	79.1	34.7	20.6	17.8	6.0	14.9	0.0	6.6	7.0	1.3
256	48.4	18.2	13.6	13.4	3.2	12.8	0.0	4.8	7.3	0.7
	MFLOPS					V.OP RATIO				
N_p	Total	FFT	FD & nonlinear	Multigrid	CR	Total	FFT	FD & nonlinear	Multigrid	CR
16	3753.3	4474.8	3908.3	1925.5	2079.6	99.54	99.63	99.56	99.10	99.49
32	3586.0	4462.3	3587.2	1729.1	1988.3	99.52	99.63	99.51	99.01	99.46
64	3383.3	4375.7	3188.6	1553.2	1957.5	99.48	99.62	99.46	98.90	99.43
128	2878.3	4195.9	2527.6	1133.7	1661.9	99.38	99.59	99.35	98.63	99.32
256	2476.3	4114.3	2160.9	772.7	1623.2	99.24	99.54	99.22	98.21	99.24

communications is required to calculate non-linear terms, and the amount of communications will increase.

5. Summary

We reported an implementation technique of the Fourier transform method on the Earth simulator. This method avoids the time-consuming Legendre transform and performs only the Fourier transform. The developed code successfully simulated the self-exciting dynamo. A certain level of numerical performance was achieved. Further improvement of the performance seems to depend on how the Poisson equation is effectively solved. Parallelization in the azimuthal wavenumber might be a remedy for inefficiency of the parallelized multigrid algorithm, although all-to-all global data transfer also costs much time in most of highly parallel computers.

6. Acknowledgements

References

[1] A. Kageyama, T. Sato, and the Complexity Simulation Group, Computer simulation of a magnetohydrodynamic dynamo. II, *Phys. Fluids*, *B*, vol.**2**, pp.2793–2805, 1995.

[2] G. A. Glatzmaier, and P. H. Roberts, A three-dimensional convective dynamo solution with rotating and finitely conducting inner core and mantle, *Phys. Earth Planet. Inter.*, vol.**91**, pp.63–75, 1995.

[3] M. Kono, and P. Roberts, Recent geodynamo simulations and observations of the geomagnetic field, *Rev. Geophys.*, vol.**40**, pp.B1–B46, 2002.

[4] F. Takahashi, M. Matsushima, and Y. Honkura, Simulations of a quasi-Taylor state geomagnetic field including polarity reversals on the Earth Simulator, *Science*, vol.**309**, pp.459–461, 2005.

[5] G. A. Glatzmaier, Numerical simulations of stellar convective dynamos: I. The model and method, *J. Comput. Phys.*, vol.**55**, pp.461–484, 1984.

[6] H. Matsui, and H. Okuda, Development of a simulation code for MHD dynamo processes using the GeoFEM platform, *Int. J. Comput. Fluid. D.*, vol.**18**, pp.323–332, 2004.

[7] 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.

[8] Y. Oishi, A. Sakuraba, and Y. Hamano, Numerical method for geodynamo simulations based on Fourier expansion in longitude and finite difference in meridional plane, *Phys. Earth Planet. Inter.*, 2006 (submitted).

[9] P. Amodio, and N. Mastronardi, A parallel version of the cyclic reduction algorithm on a hypercube, *Parallel comput.*, vol.**19**, no.11, pp.1273–1281, 1993.

[10] M. Kameyama, ACuTEMan: A multigrid-based mantle convection simulation code and its optimization to the Earth Simulator, *J. Earth Sim.*, vol.**4**, pp.2–10, 2005.

[11] U. R. Christensen, J. Aubert, P. Cardin, E. Dormy, S. Gibbons, G. A. Glatzmaier, E. Grote, Y. Honkura, C. Jones, M. Kono, M. Matsushima, A. Sakuraba, F. Takahashi, A. Tilgner, J. Wicht, and K. Zhang, A numerical dynamo benchmark, *Phys. Earth Planet. Inter.*, vol.**128**, pp.25–34, 2001.