CIG Community Accuracy & Performance Benchmark

CIG Dynamo Working Group

October 29, 2013

Contents

1 Introduction

A number of numerical dynamo models have been developed in the last ten years. As new models and numerical methods continue to be developed, it is important to update and extend benchmarks for testing these models. The first dynamo benchmark of Christensen *et al.* (2001) was applied to models based on spherical harmonic expansion methods. The published results compared the accuracy of several models developed by different research groups. Only a few groups have reported results of the dynamo benchmark using local methods (Harder and Hansen, 2005; Matsui and Okuda, 2005; Chan *et al.*, 2007) because of the difficulty treating magnetic boundary conditions.

The purpose of this benchmark test is to assess numerical geodynamo models on a massively parallel computational platform. All benchmark calculations will be carried out using the CIG allocation on XSEDE's Stampede computer, which affords a high degree of uniformity in benchmark comparisons. Further, Stampede's significant computational power is ideal for performance testing a broad range of codes. To facilitate this, CIG developers Dr. Hiroaki Matsui (hrmatsui@ucdavis.edu) and Dr. Eric Heien (emheien@ucdavis.edu) will *i*) provide participants login accounts on Stampede and *ii*) install participating dynamo codes on Stampede (or assist participants with the installation process).

In order to present our first results at this fall's AGU meeting, the deadline for reporting the accuracy benchmark solutions is **November 1st, 2013**. Once ported onto Stampede, the Calypso code was able to run the relatively inexpensive accuracy benchmark calculations in approximately a day.

We also want to encourage the testing of local methods, such as finite volume or finite element methods. We propose two kinds of benchmark tests. The first is an accuracy benchmark with the goal of establishing the resolution needed to reproduce a known numerical solution within a 1% error tolerance. The second test is a performance benchmark, which will assess the speed and scalability of numerical dynamo models. We strongly encourage participants to run the accuracy benchmark tests using two different choices of magnetic boundary conditions (i.e. an insulated boundary and a pseudo-vacuum boundary). However, participants are welcome to contribute benchmark results using only one choice of boundary condition.

The benchmark tests involve standard solutions to convection of a Boussinesq fluid in a rotating spherical shell with magnetic fields. For the accuracy benchmark, the following two cases are proposed:

- Dynamo simulation using electrically insulating boundaries, as defined by Christensen *et al.* (2001).

- Dynamo simulation using pseudo-vacuum boundaries, following the study of Jackson *et al.* (2013); i.e. http://jupiter.ethz.ch/~ajackson/pseudo.pdf.

For the required accuracy, we choose the 1% difference from the suggested solution for the insulated boundary dynamo case, and Jackson *et al.* (2013) highest resolution solution for the pseudo-vacuum dynamo case. If the simulation does

not obtain the 1% accuracy, then we request that participants report on the convergence of the solution with the spatial resolutions simulated, and the expected spatial resolution to reach 1% accuracy. For spectral codes (i.e., using spherical harmonics), it is required that all reported solutions are de-aliased in the horizontal directions.

In the performance benchmark excerise, Drs. Matsui and Heien will work with participants to test each dynamo code's computational performance on Stampede. Using the same parameters as in the accuracy benchmark cases, these tests will investigate strong and weak parallel scaling efficiency for all the participating codes.

2 Accuracy Benchmark Exercise

2.1 Benchmark Model

2.1.1 Governing Equations

The benchmarks presented here are magnetohydrodynamics (MHD) simulations in a rotating spherical shell modeled on the Earth's outer core (see Figure 1).

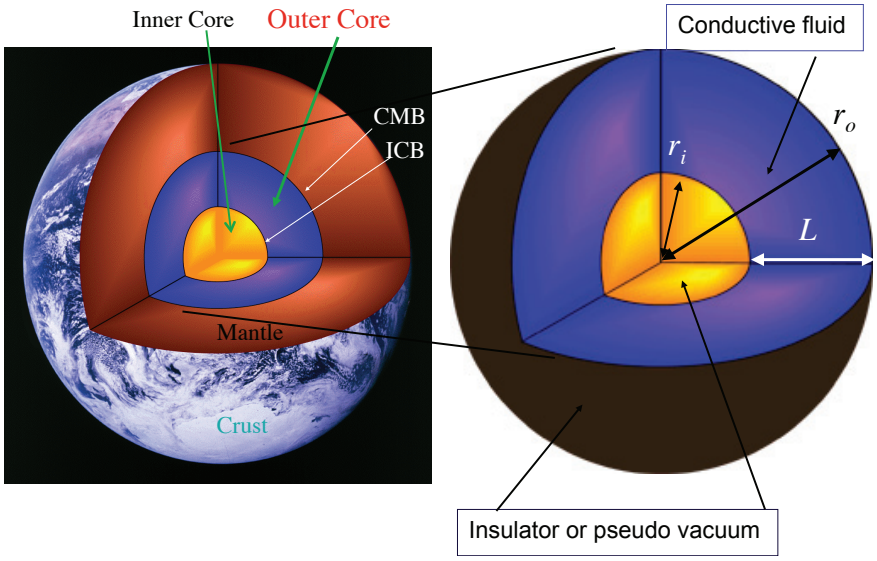

Figure 1: Rotating spherical shell modeled on the Earth's outer core.

We consider a spherical shell from the inner core boundary (ICB) $r = r_i$ to the core mantle Boundary (CMB) $r = r_o$ in a rotating frame which rotates at a constant angular velocity $\mathbf{\Omega}$. The shell is filled with a Boussinesq fluid with constant diffusivities (kinematic viscosity ν, magnetic diffusivity η, thermal

diffusivity κ) and thermal expansivity α. The inner core ($0 < r < r_i$) is co-rotating with the mantle, and assumed to be an electrical insulator or pseudo-vacuum. The region outside of the core is also assumed to be an electrical insulator or pseudo-vacuum.

The governing equations of the MHD dynamo problem are the following:

$$\begin{aligned} E\left[\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,\boldsymbol{u}\right] &= -\nabla P + E\nabla^2 u - 2\,(\hat{z}\times\boldsymbol{u}) \\ &\quad + RaT\frac{\boldsymbol{r}}{r_o} + \frac{1}{Pm}\,(\nabla\times\boldsymbol{B})\times\boldsymbol{B}, \qquad (1) \\ \frac{\partial T}{\partial t} + (\boldsymbol{u}\cdot\nabla)\,T &= \frac{1}{Pr}\nabla^2 T, \qquad (2) \\ \frac{\partial \boldsymbol{B}}{\partial t} &= \frac{1}{Pm}\nabla^2\boldsymbol{B} + \nabla\times(\boldsymbol{u}\times\boldsymbol{B}), \qquad (3) \end{aligned}$$

and

$$\nabla\cdot\boldsymbol{u} = \nabla\cdot\boldsymbol{B} = 0, \qquad (4)$$

where, $\boldsymbol{u}$, P, $\boldsymbol{B}$, and T are the velocity, pressure, magnetic field, and temperature, respectively. To obtain the normalized equations, as given above, the shell width of $L = r_o - r_i$ and a viscous diffusion time of L^2/ν are selected as the length and time scales, respectively. The ratio of the inner core radius to the outer core radius is set to $r_i/r_o = 0.35$. Thus, inner core radius and outer core radius are $r_i = 7/13$ and $r_o = 20/13$, respectively. The magnetic field is normalized by $B_0^2 = \rho\mu_0\eta\Omega$, where ρ and μ_0 are the density of the fluid and the magnetic permeability, respectively. The gravity to the center is proportional to the radius r, and amplitude g_o is defined by the gravity at CMB.

2.1.2 Control Parameters

The dimensionless control parameters in above equations are the Ekman number $E = \nu/\Omega L^2$; the modified Rayleigh number $Ra = \alpha g_o \Delta T L/\Omega\nu$; the Prandtl number $Pr = \nu/\kappa$; and the magnetic Prandtl number $Pm = \nu/\eta$. In the Ra definition, the temperature difference between the inner and outer shell boundaries is denoted by ΔT.

The following parameter values are used in all the benchmark calculations proposed herein:

$$\begin{aligned} E &= 1.0\times10^{-3}, \qquad (5) \\ Ra &= 100, \\ Pr &= 1.0, \\ Pm &= 5.0. \end{aligned}$$

2.1.3 Boundary Conditions for Velocity and Temperature

The boundary conditions exert a significant influence on the motion of the fluid and the overall dynamo process. Non-slip conditions for the velocity field and

constant temperature values are imposed at both boundaries of the fluid shell:

$$\boldsymbol{u} = 0 \qquad \text{at } r = r_i \text{ and } r_o, \tag{6}$$

$$T = 1 \qquad \text{at } r = r_i, \tag{7}$$

$$T = 0 \qquad \text{at } r = r_o. \tag{8}$$

2.1.4 Initial Conditions for Velocity and Temperature

The initial velocity and temperature fields defined by Christensen *et al.* (2001) for both the insulated and the pseudo-vacuum benchmarks.

There is no flow at the start. That is,

$$\boldsymbol{u} = 0 \tag{9}$$

Thermal convection is initiated by the following temperature perturbation:

$$T = \frac{r_o r_i}{r} - r_i + \frac{21}{\sqrt{17920\pi}} \left(1 - 3x^2 + 3x^4 - x^6\right) \sin^4\theta \cos 4\phi, \tag{10}$$

where θ is the colatitude, ϕ the longitude, and $x = 2r - r_i - r_o$. By using real part of the spherical harmonics $\Re\left(Y_l^m\right)$, the horizontal perturbation of the temperature can be described as

$$\frac{21}{\sqrt{17920\pi}} \sin^4\theta \cos 4\phi = 0.2\Re\left(Y_4^4\right), \tag{11}$$

where we have defined our spherical harmonics Y_l^m to be

$$Y_l^m = \sqrt{\frac{2l+1}{4\pi} \frac{(l-|m|)!}{(l+|m|)!}} P_l^m e^{im\phi}. \tag{12}$$

This chosen normalization ensures orthonormality spherical harmonics over the sphere, namely

$$\int_0^{2\pi} \int_0^{\pi} Y_l^m Y_l^{*m} \sin\theta d\theta d\phi = 1. \tag{13}$$

2.2 Insulating Boundaries Case

2.2.1 Magnetic Boundary Conditions

The regions outside the fluid shell are assumed to be electrical insulators in the present benchmark. In the electric insulator, current density vanishes,

$$\boldsymbol{J}_{ext} \quad = \quad 0, \tag{14}$$

where the suffix $_{ext}$ indicates fields outside of the fluid shell. At the boundaries of the fluid shell, the magnetic field $\boldsymbol{B}_{fluid}$, current density $\boldsymbol{J}_{fluid}$, and electric field $\boldsymbol{E}_{fluid}$ in the conductive fluid satisfy:

$$(\boldsymbol{B}_{fluid} - \boldsymbol{B}_{ext}) = 0, \tag{15}$$
$$(\boldsymbol{J}_{fluid} - \boldsymbol{J}_{ext}) \cdot \hat{r} = 0, \tag{16}$$

and

$$(\boldsymbol{E}_{fluid} - \boldsymbol{E}_{ext}) \times \hat{r} = 0, \tag{17}$$

where, $\hat{r}$ is the radial unit vector (i.e. normal vector for the spherical shell boundaries). Consequently, radial current density $\boldsymbol{J}$ vanishes at the boundary as

$$\boldsymbol{J} \cdot \hat{r} = 0 \text{ at } r = r_i, r_o \tag{18}$$

In an electrical insulator the magnetic field can be described as a potential field

$$\boldsymbol{B}_{ext} = -\nabla W_{ext}, \tag{19}$$

where W_{ext} is the magnetic potential. The boundary conditions can be satisfied by connecting the magnetic field in the fluid shell at boundaries to the potential fields. If the magnetic field is decomposed using spherical harmonics coefficients with poloidal and toroidal components as

$$\begin{aligned} \boldsymbol{B}(r,\theta,\phi) &= \sum_{l=1}^{l_{max}} \sum_{m=-l}^{l} [\nabla \times \nabla \times (B_{Sl}^{\ m}(r) Y_l^m(\theta,\phi)\hat{r}) \\ &\quad + \nabla \times (B_{Tl}^{\ m}(r) Y_l^m(\theta,\phi)\hat{r})], \end{aligned} \tag{20}$$

then the boundary condition is described by

$$\frac{l+1}{r} B_{Sl}^{\ m}(r) = \frac{\partial B_{Sl}^{\ m}}{\partial r} \text{ at } r = r_i, \tag{21}$$
$$\frac{l}{r} B_{Sl}^{\ m}(r) = -\frac{\partial B_{Sl}^{\ m}}{\partial r} \text{ at } r = r_o, \tag{22}$$

and

$$B_{Tl}^{\ m}(r) = 0 \text{ at } r = r_i, r_o. \tag{23}$$

2.2.2 Initial Magnetic Field

Initial values are important for these benchmark tests. In particular, simulations must be started with a strong magnetic field to sustain the magnetic field in this parameter regime.

The initial magnetic field for the insulating magnetic boundaries case is defined in Christensen *et al.* (2001) as

$$B_r = \frac{5}{8}\left(8r_o - 6r - 2\frac{r_i^4}{r^3}\right)\cos\theta, \tag{24}$$

$$B_\theta = -\frac{5}{8}\left(8r_o - 9r + \frac{r_i^4}{r^3}\right)\sin\theta, \tag{25}$$

and

$$B_\phi = 5\sin\left(\pi\left(r - r_i\right)\right)\sin 2\theta. \tag{26}$$

2.3 Pseudo-Vacuum Case

2.3.1 Magnetic Boundary Conditions

Under the pseudo-vacuum boundary condition, the magnetic field has only a radial component at the boundaries (e.g., Harder and Hansen, 2005). Given the conservation of the magnetic field (4), the magnetic boundary condition will be

$$\frac{\partial}{\partial r}\left(r^2 B_r\right) = B_\theta = B_\phi = 0 \text{ at } r = r_i, r_o. \tag{27}$$

The present boundary condition is also described by using the poloidal and toroidal coefficients (20) as

$$\frac{\partial B_{Sl}^m}{\partial r} = B_{Tl}^m(r) = 0 \text{ at } r = r_i, r_o. \tag{28}$$

2.3.2 Initial Magnetic Field

The initial magnetic field for the pseudo-vacuum benchmark is defined by Jackson *et al.* (2013) as

$$B_r = \frac{5}{8}\frac{-48 r_i r_o + \left(4r_o + r_i\left(4 + 3r_o\right)\right)6r - \left(16 + 12\left(r_i + r_o\right)\right)r^2 + 9r^3}{r}\cos\theta, \tag{29}$$

$$B_\theta = -\frac{15}{4}\frac{\left(r - r_i\right)\left(r - r_o\right)\left(3r - 4\right)}{r}\sin\theta, \tag{30}$$

and

$$B_\phi = \frac{15}{8}\sin\left(\pi\left(r - r_i\right)\right)\sin 2\theta. \tag{31}$$

Note: This initial field generates strong Lorentz force. Thus, shorter time steps may be required in the beginning of these simulation, relative to the insulating magnetic boundaries case.

2.4 Diagnostics and Data Outputs

Six data are requested once a given dynamo simulation has reached a quasi-steady state. These are:

1. The kinetic energy averaged over the fluid shell E_{kin} defined by

$$E_{kin} = \frac{1}{V}\int \frac{1}{2}u^2 dV, \tag{32}$$

   where V is the volume of the fluid shell.

2. The magnetic energy averaged over the fluid shell E_{mag} defined by

$$E_{mag} = \frac{1}{V}\frac{1}{EPm}\int \frac{1}{2}B^2 dV. \tag{33}$$

3. Angular drift velocity of the field patterns in the zonal direction, ω.

4. Local temperature T.

5. Local zonal velocity u_ϕ.

6. Local θ-component of the magnetic field B_θ.

The local values (4, 5, and 6) should be measured at a specific point, but solution propagates in the longitudinal direction. The local values are measured at mid-depth of the fluid shell ($r = (r_o + r_i)$ / 2) in the equatorial plane ($\theta = \pi/2$), with a ϕ-coordinate given by the conditions $u_r = 0$ and $\partial u_r/\partial\phi > 0$. Further details are given in Appendix A and Christensen *et al.*(2001).

We set the threshold of the convergence to be 1% of the suggested solution for the all six parameters. Suggested solutions for the insulated boundary and pseudo-vacuum boundary cases are proposed by Christensen *et al.*(2001) and Jackson *et al.* (2013), respectively. We follow these suggested solution as a reference. The reference solutions are shown in Table 2 and Table 3 for insulated dynamo and pseudo-vacuum dynamo cases, respectively. The solution of local magnetic field B_θ can have an opposite sign if initial magnetic field has opposite direction from them given in sections 2.2.2 or 2.3.2. For example, if pseudo-vacuum case is started by using the initial magnetic field for insulated boundary case, B_θ should have the opposite sign from the solution in Table 3 and the other solutions should be the solutions in Table 3.

All values are to be evaluated when the simulation achieves a steady state in a frame drifting in longitude at a constant angular velocity ω relative to the rotating frame of reference (positive value represents prograde drifting). As seen in Tables 2 and 3, fields drifts retrograde in the insulated boundary case and prograde in pseudo-vacuum case. The time variation of the kinetic and magnetic energies can be used to deduce when a quasi-steady state has been reached at time t_e. The relative changes of kinetic and magnetic energies are denoted by

$$\epsilon_{kin} = \frac{1}{E_{kin}(t=\infty)}\frac{dE_{kin}}{dt} \quad \text{and} \quad \epsilon_{mag} = \frac{1}{E_{mag}(t=\infty)}\frac{dE_{mag}}{dt}, \tag{34}$$

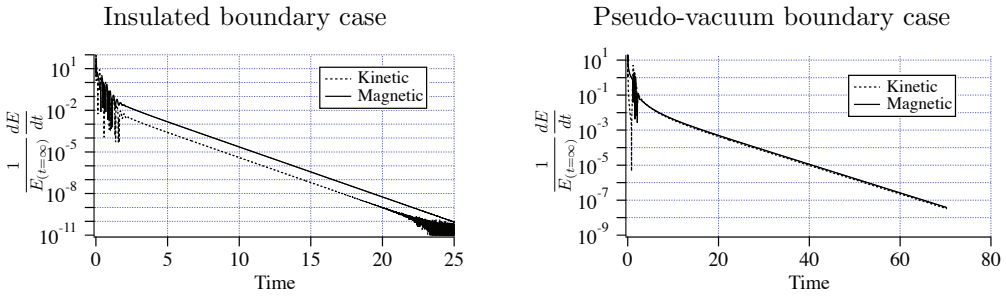

Figure 2: Rate of change of the relative kinetic and magnetic energies averaged over the fluid shell plotted as a function of viscous diffusion time.

The time variations of ϵ_{kin} and ϵ_{mag} are plotted in Figure 2 based on Calypso simulations. Both quantities decrease exponentially with time. Thus, we can estimate the fractional change in magnetic energy, $\Gamma_{mag} = \int_{t_e}^{\infty} \epsilon_{mag} dt$ were it possible to integrate from t_e to $t = \infty$. The number of viscous diffusion times needed to reach Γ_{mag} values of order 10^{-5} and 10^{-7} are given in Table 1.

The left panel of Figure 2 shows the insulating boundaries Calypso solution made with $N_r = 96$ radial grid points shown in Table 4. The magnetic energy differs by 0.661% from the insulated dynamo benchmark solution at $t = 11.1$ viscous diffusion times. This error will change only from 0.661% to 0.660% by continuing the simulation, because the temporal decay of ϵ_{mag} implies that the magnetic energy will change by at most 0.001% between $t = 11.1$ and $t = \infty$.

For steady or quasi-steady solutions, please report the instantaneous solution and the measured time. The benchmark cases have azimuthally symmetrical solutions (m=4), and these solutions also have a symmetry (e.g. velocity, temperature, and current density) or anti-symmetry (e.g. vorticity and magnetic field) with respect to the equatorial plane. Please only report solutions from calculations made with no imposed symmetries. For some local methods, however, the solutions can be remain weakly oscillatory. These oscillations come either from the interpolation procedures necessary to obtain the local fields, or from interference between physical field patterns and the pattern of the numerical mesh (Harder and Hansen, 2005; Matsui and Okuda, 2005). Should such oscillations exist, please average the solution over more than 10 oscillations and report the averaging time window.

In order to compare spatial resolutions employed, Christensen *et al.* (2001) defines the spatial resolution as the cubic root of the degree of freedom of spherical harmonics coefficients for a scalar value as

$$R_{sph} = N_r^{1/3} \left(l_{max} \left(2m_{max} + 1 \right) - m_{max}^2 + m_{max} + 1 \right)^{1/3}, \tag{35}$$

where N_r, l_{max}, and m_{max} are the number of grid points in the radial direction, the truncation degree, and the truncation order, respectively. This definition

Table 1: Viscous diffusion times to reach the quasi-steady state t_e and number of time steps by Calypso.

	$\int_{t_e}^{\infty} \epsilon_m dt = 10^{-5}$		$\int_{t_e}^{\infty} \epsilon_m dt = 10^{-7}$	
	time	time steps	time	time steps
Insulated boundary	10.8	2.2×10^5	16.3	3.3×10^5
pseudo-vacuum boundary	31.8	6.4×10^6	47.9	9.6×10^6

Table 2: Reference solution for the insulating magnetic boundary conditions case.

	1% smaller	Solution	1% larger
E_{kin}	30.465	30.773	31.081
E_{mag}	620.15	626.41	632.67
T	0.36965	0.37338	0.37711
u_ϕ	-7.701	-7.625	- 7.549
B_θ	-4.9782	-4.9289	-4.8796
ω	-3.1327	-3.1017	-3.0707

can be used for codes that use spectral transform methods.

For local method, R_{lcl} is defined by the cubic root of the degree of freedom for a scalar value in the fluid shell. For example, Matsui and Okuda (2005) used cubed sphere for the FEM mesh. They defined the resolution parameter as

$$R_{lcl} = (N_r N_{sphere})^{1/3}, \tag{36}$$

where N_r and N_{sphere} are the number of nodes in the radial direction and on a spherical surface, respectively.

Solutions for the accuracy benchmark exercises are given in Appendix B. An output data reporting template is found in Appendix C.

Table 3: Reference solution for the pseudo-vacuum dynamo case.

	1% smaller	Solution	1% larger
E_{kin}	40.271	40.678	41.086
E_{mag}	217.20	219.39	221.58
T	0.42163	0.42589	0.43014
u_ϕ	-11.752	-11.636	-11.519
B_θ	1.3903	1.4043	1.4184
ω	0.74240	0.74990	0.7574

3 Performance Benchmark Exercise

The performance dynamo benchmark is the essentially novel part of this project. After establishing that a code can solve the dynamo equations with sufficient accuracy, it is then of interest to the dynamo community to determine which codes can do so efficiently on modern (and future) massively-parallelized computing platforms. Because all the participating codes will have already been ported to Stampede for the accuracy benchmark calculations, it will be straightforward to use Stampede to carry out a suite of performance calculations with all the participating codes.

CIG programmers Matsui and Heien will be in charge of carrying out this performance suite. Thus, participants are not required to take part in this task, but those who are interested are of course invited to collaborate. In doing so, Drs. Matsui and Heien will do basic optimization and possibly make some suggestions to developers. However, it will not be possible for them to work on significantly improving the parallelization schemes of all the codes.

Here, we will use the same the model, parameter values and boundary conditions as employed in the two accuracy benchmark cases, but the simulations will be carried out at much higher resolutions than was necessary for the accuracy benchmarks. In doing this, we will investigate computational performance via measurements of

- Elapsed time for initialization.
- Elapsed time for each time step.
- Elapsed time for data communication for each step.
- Elapsed time for time stepping excluding the data communication.
- Elapsed time for linear solvers. (most time consuming part for the local methods).

Note that the elapsed time for IO will not be explicitly evaluated because there are too many control parameters to control (e.g. the specific data format).

We expect that performance will depend on the magnetic boundary conditions for the local methods. Thus, assessing differences in performance due to the choice of magnetic boundary condition will be another one of the focuses of the performance exercise.

To investigate parallel efficiency, we will investigate the scalability of the participating codes by two approaches. One approach is to measure the so-called "strong scalability". In strong scalability tests, spatial resolution is fixed, and performance is evaluated by using different numbers of CPU cores. The elapsed time should decrease linearly with the number of cores in the ideal scalability.

Another scaling test is to measure so-called "weak scalability". Here the spatial resolution is increased linearly with the number of cores (i.e. the degrees of freedom for each processor core is fixed). If the number of calculations increase linearly with the degree of freedom, then the ideal value of the elapsed time does not change with the number of processor cores.

The number of cores used in each scalability test will depend on the parallel performance of the code from the accuracy benchmarks. Based on our experience using the Calypso code on Stampede (see Figures 11-14), we will test scalability starting at 16 cores (one node) and going up from there. For codes that scale well, we can access up to 4096 cores (256 nodes) without special request. Furthermore, if a special access request is made, we can use a greater numbers of cores. However, it is unlikely we will need to do so for the problem sizes investigated here.

References

[1] K.H. Chan, K. Zhang, L. Li and X. Liao, A new generation of convection-driven spherical dynamos using EBE finite element method , *Physics of the Earth and Planetary Interiors*, **163** ,251 - 265 (2007).

[2] U.R. Christensen, J. Aubert, P. Cardin, E. Dormy, S. Gibbons, G.A. Glatzmaier, E. Grote, H. Honkura, C. Jones, M. Kono, M. Matsushima, A. Sakuraba, F. Takahashi, A. Tilgner, J. Wicht and K. Zhang, A numerical dynamo benchmark, *Physics of the Earth and Planetary Interiors*, **128**, 25–34 (2001).

[3] H. Harder and U. Hansen, A Finite-Volume Solution Method for Thermal Convection and Dynamo Problems in Spherical Shells, *Geophys. J. Int.*, **161**, 522–532 (2005).

[4] A. Jackson, A. Sheyko, P. Marti, A. Tilgner, D. Cebron, S.Vantieghem, R.Simitev, F.Busse, X. Zhan, G. Schubert, S. Takehiro, Y. Sasaki, Y.-Y. Hayashi, A. Ribeiro, C. Nore, and J.-L. Guermond, A spherical shell numerical dynamo benchmark with pseudo-vacuum magnetic boundary conditions, under review at *Geophysical Journal International* (2013).

[5] H. Matsui and H. Okuda, MHD dynamo simulation using the GeoFEM platform–verification by the dynamo benchmark test, *International Journal of Computational Fluid Dynamics*, **19**, 15-22 (2005).

[6] F. Takahashi, Implementation of a high-order combined compact difference scheme in problems of thermally driven convection and dynamo in rotating spherical shells, *Geophysical & Astrophysical Fluid Dynamics* , **106**, 231-249 (2012).

[7] J. Wicht, S. Stellmach, and H. Harder. Numerical models of the geodynamo: From fundamental cartesian models to 3d simulations of field reversals. In

Geomagnetic Field Variations, Advances in Geophysical and Environmental Mechanics and Mathematics, pages 107158. Springer Berlin Heidelberg, 2009.

Appendix A Evaluation of Local Fields in Spectral Codes

For the accuracy benchmarks, it is necessary to report local fields (T, u_ϕ, and B_θ) at mid-depth of the fluid, in the equatorial plane, and with a ϕ-coordinate given by the conditions $u_r = 0$ and $\partial u_r/\partial\phi > 0$. To find these observation points, two approaches are considered here:

1. Find the observation point using the Newton-Raphson method from the poloidal velocity spectrum data.

2. Calculate the local fields in the equatorial plane via Legendre and Fourier transform. Then find the observation point by linear interpolation of the local fields onto an equidistant longitudinal grid along a circle at mid-depth.

By using approach 1, the solution should decay exponentially in time. Then please report the instantaneous solutions for the local quantities of interest. If approach 2 is taken, oscillations due to the liner interpolation may be observed in the local solution, as in the case using local methods. In that case, please report time averaged solutions over 10 oscillations and report the time-averaging window. This oscillation can be minimized by using a larger number of grid points in the linear interpolation process.

Appendix B Accuracy Benchmark Results

B.1 Solutions for insulating magnetic boundaries case

Solutions for the insulated boundary dynamo case have been reported by the following authors:

- Christensen *et al.* (2001) using spherical methods by various groups
- Matsui and Okuda (2005) using FEM in hemispherical shell
- Chan *et. al.* (2007) using FEM
- Harder using FVM (in Wicht *et. al.*, 2009)
- Takahashi (2012) using spherical harmonics transform and combined compact difference scheme (CCDS)

Solutions in these publications and solutions by Calypso are listed in Tables 4 and 5, and these solutions are plotted as a function of resolution parameter R_{sph} or R_{lcl} in Figure 5 and Figure 5. Radial magnetic field at CMB and radial velocity at mid-depth of the shell at quasi-steady state are shown in Figure 3. For a quick check of the simulation, the evolution of the average kinetic energy E_{kin} in the beginning of the benchmark is shown Figure 4.

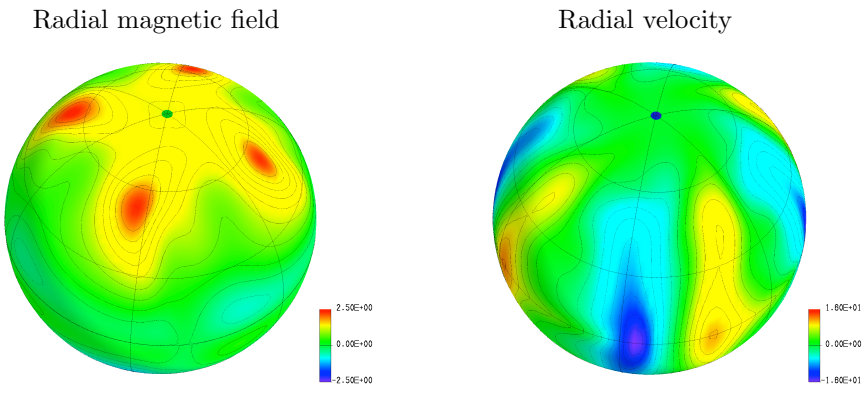

Figure 3: Radial magnetic field B_r at CMB $r = r_o = 20/13$ (left) and radial velocity u_r (right) at mid-depth of the shell in quasi-steady state for insulated magnetic boundary case. Step of contour lines for B_r and u_r are 0.25 and 2.0, respectively.

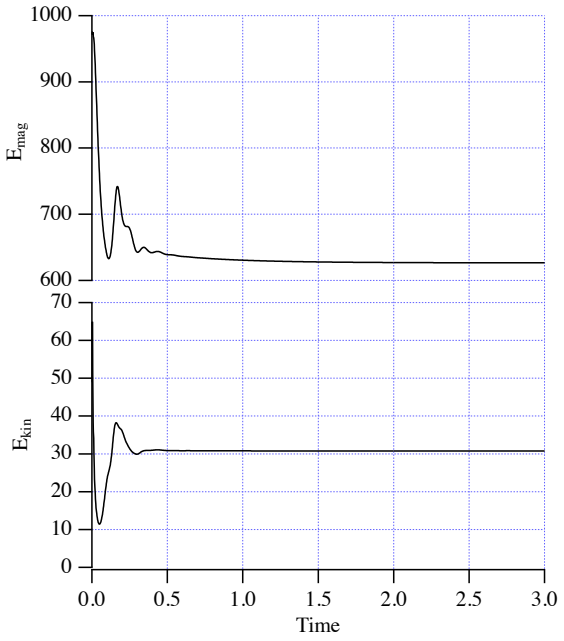

Figure 4: Time evolution of magnetic energy E_{mag} (top) and kinetic energy E_{kin} (bottom) for insulated magnetic boundary case to $t = 3.0$. These time series data sets are available at http://geodynamics.org/cig/community/workinggroups/geodyn/benchmark/.

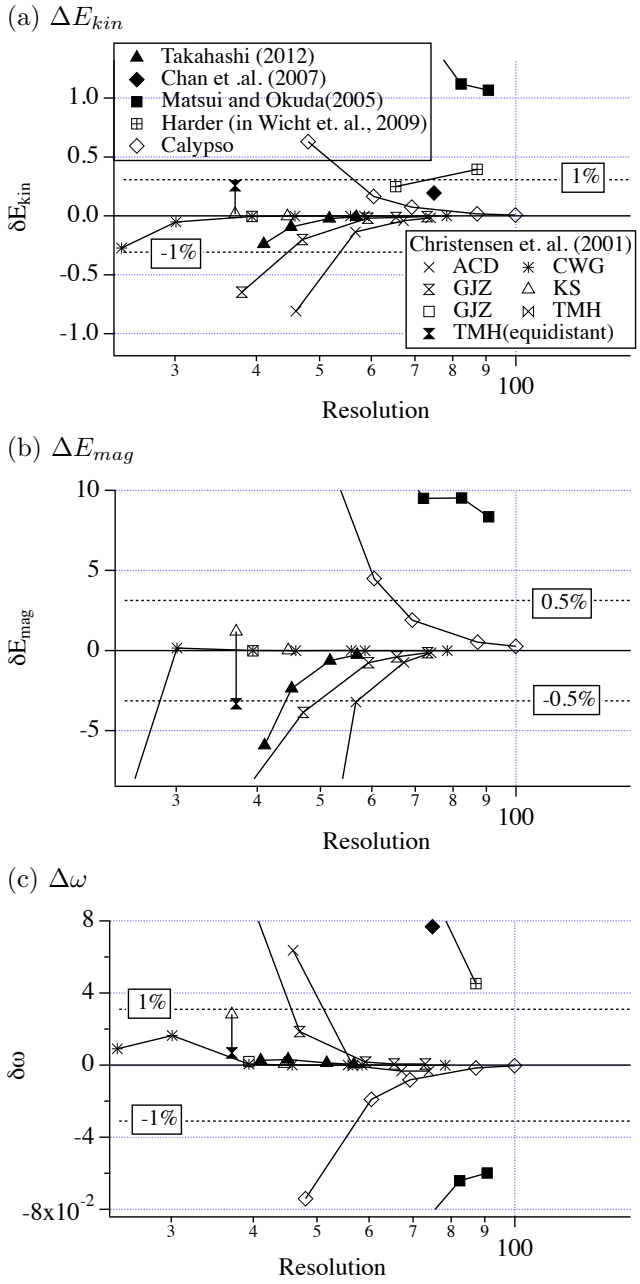

Figure 5: Difference of the solution from suggested solution for global solution in insulated magnetic boundary case. Scales of each error is same as the nonmagnetic case. Relative error from the suggested solution ($\pm 1\%$ for E_{kin}, $\pm 0.5\%$ for E_{mag}, and $\pm 1\%$ for ω,) is plotted by dotted line in each plot.

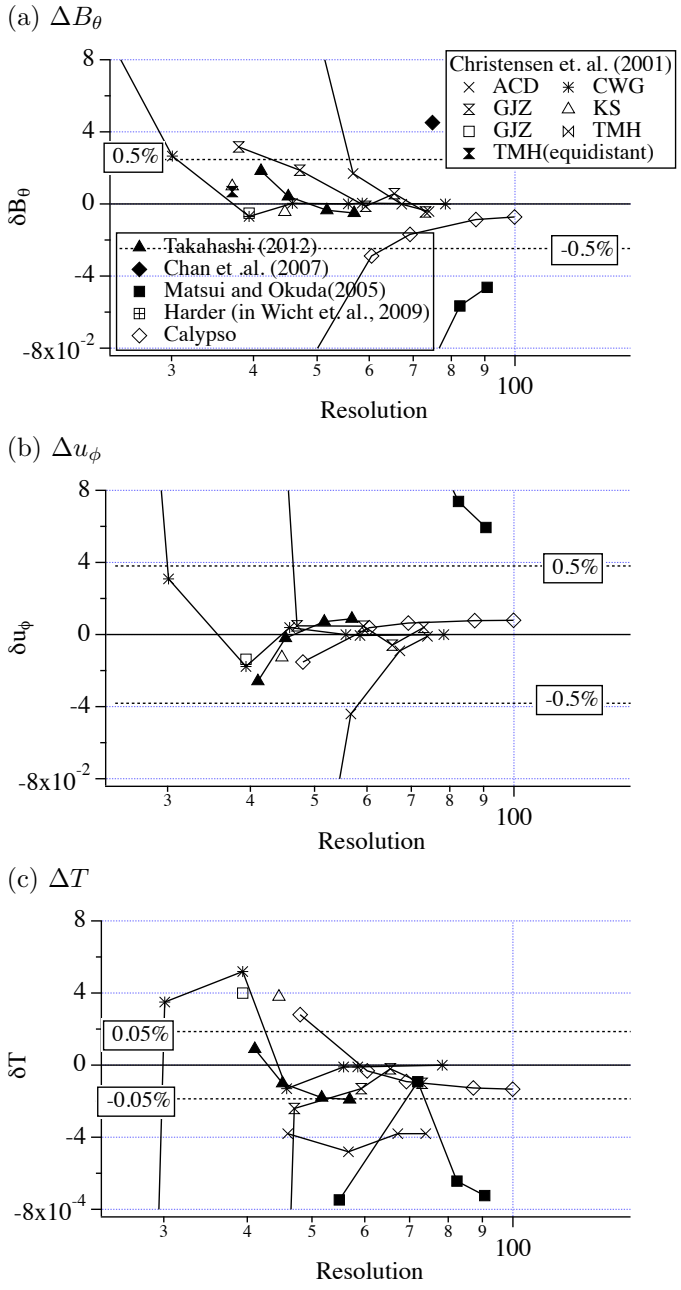

Figure 6: Difference of the solution from suggested solution for local fields in insulated magnetic boundary case. Relative error from the suggested solution ($\pm 0.5\%$ for B_θ and u_ϕ, and $\pm 0.05\%$ for T) is plotted by dotted line in each plot.

Table 4: Simulation results for insulated magnetic boundary case using spherical harmonics expansion

				E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Suggested solution				30.773	626.41	0.37338	-7.625	-4.9289	-3.1017
Christensen *et al.* (2001)									
Group	N_r	N_θ	l_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
ACD	48	46	44	29.965	601.912	0.373	-7.864	-4.779	-3.038
ACD	90	46	44	30.637	623.204	0.3729	-7.669	-4.912	-3.102
ACD	150	46	44	30.732	625.681	0.3730	-7.634	-4.929	-3.105
ACD	200	92	44	30.758	626.284	0.3730	-7.626	-4.933	-3.105
CWG	21	40	26	30.5015	616.085	0.36281	-7.2228	-4.8456	-3.0926
CWG	25	48	32	30.7214	626.572	0.37373	-7.5941	-4.9023	-3.0852
CWG	33	64	42	30.7686	626.420	0.37390	-7.6427	-4.9358	-3.1011
CWG	33	80	53	30.7714	626.406	0.37325	-7.6211	-4.9285	-3.1016
CWG	41	96	64	30.7715	626.416	0.37337	-7.6250	-4.9288	-3.1016
CWG	49	96	63	30.7716	626.413	0.37337	-7.6255	-4.9284	-3.1016
CWG	65	128	85	30.7734	626.409	0.37338	-7.6250	-4.9289	-3.1017
KS	48	64	42	30.7709	626.434	0.36226	-7.6376	-4.9333	-3.1009
TG	33	64	42	30.7695	626.402	0.37378	-7.6387	-4.934	-3.0997
TMH	70(e)		26	31.0298	623.092	0.3679	-7.3525	-4.9220	-3.0949
TMH	70		26	30.7901	627.607	0.3675	-7.3330	-4.9190	-3.0735
GJZ	40	38	36	30.1263	617.462	0.36226	-7.0899	-4.8972	-2.992
GJZ	40	52	50	30.5724	622.558	0.37314	-7.6201	-4.9101	-3.083
GJZ	80	52	50	30.7541	625.656	0.37325	-7.6203	-4.9301	-3.100
GJZ	100	54	52	30.7605	626.02	0.37336	-7.6308	-4.9232	-3.101
GJZ	150	78	50	30.7677	626.282	0.37328	-7.6210	-4.9333	-3.101
Takahashi (2012): Spherical harmonics & CCDM									
	N_r	N_θ	l_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
	30	80	47	30.534	620.50	0.37347	-7.6508	-4.9105	-3.0990
	40	80	47	30.679	624.05	0.37328	-7.6269	-4.9247	-3.0985
	60	80	47	30.751	625.78	0.37320	-7.6178	-4.9323	-3.1005
	80	80	47	30.765	626.15	0.37319	-7.6162	-4.9339	-3.1012
Calypso: spherical harmonics expansion									
	N_r	N_θ	l_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
	48	72	47	31.4042	641.636	0.373660	-7.64018	-5.02010	-3.17576
	96	72	47	30.9394	630.910	0.373348	-7.62113	-4.95774	-3.12064
	144	72	47	30.8475	628.310	0.373289	-7.61863	-4.94572	-3.10980
	288	72	47	30.7915	626.941	0.373253	-7.61728	-4.93747	-3.10319
	432	72	47	30.7810	626.683	0.373247	-7.61704	-4.93603	-3.10196

(e) indicates equidistant grid.

Table 5: Simulation results for insulated magnetic boundary case using local methods

		E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Suggested solution		30.773	626.41	0.37338	-7.625	-4.9289	-3.1017
Chan *et. al.,* (2007): Finite Element method (FEM)							
N_r	N_{sphere}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
21	2,562	32.335	557.47	0.37007	-8.2152	-5.1459	-3.2522
41	10,242	30.968	674.57	0.37053	-7.2652	-4.8837	-3.0249
Matsui and Okuda (2005) : Finite Element method (FEM)							
N_r	N_{sphere}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
48	3456	33.266	643.37	0.372633	-7.4634	-5.0262	-3.2223
48	7776	32.335	635.92	0.373288	-7.5067	-5.0293	-3.1906
48	7776	31.894	635.94	0.372737	-7.5511	-4.9855	-3.1658
82	7776	31.841	634.76	0.372657	-7.5656	-4.9751	-3.1615
Harder (in Wicht *et. al.,* 2009): Finite Volume method (FVM)							
N_r	N_{sphere}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
36	7776	31.022	683.30	0.3722	-7.0611		-2.9615
48	13824	31.169	666.83	0.3725	-7.2790		-3.0564

B.2 Benchmark results for pseudo-vacuum dynamo case

This benchmark is proposed by Jackson *et al.* (2013), but is not published yet. Solutions in the present nondimensional variables are listed in Tables 6 and 7, and are plotted as a function of resolution parameter R_{sph} or R_{lcl} in Figures 9 and 10. Convergence of solution by Calypso is very similar to the non-magnetic and insulated boundary cases. Radial magnetic field at CMB and radial velocity at mid-depth of the shell at quasi-steady state are shown in Figure 7. For a quick check, the evolution of kinetic and magnetic energies in the beginning of simulation are shown in Figure 8.

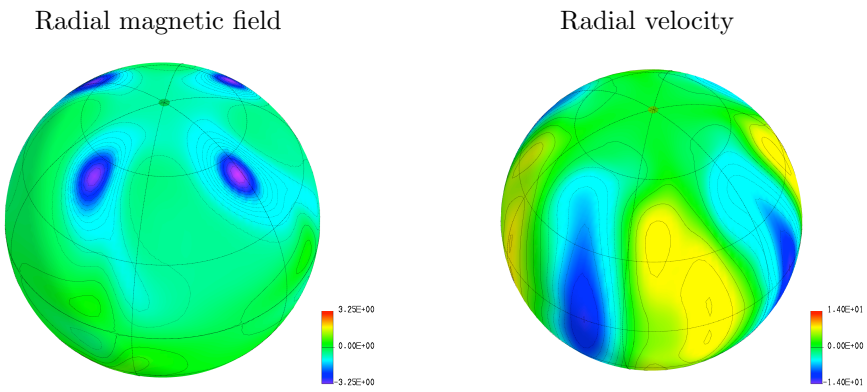

Figure 7: Radial magnetic field B_r at CMB $r = r_o = 20/13$ (left) and radial velocity u_r (right) at mid-depth of the shell in quasi-steady state for pseudo-vacuum boundary case. Step of contour lines for B_r and u_r are 0.25 and 2.0, respectively.

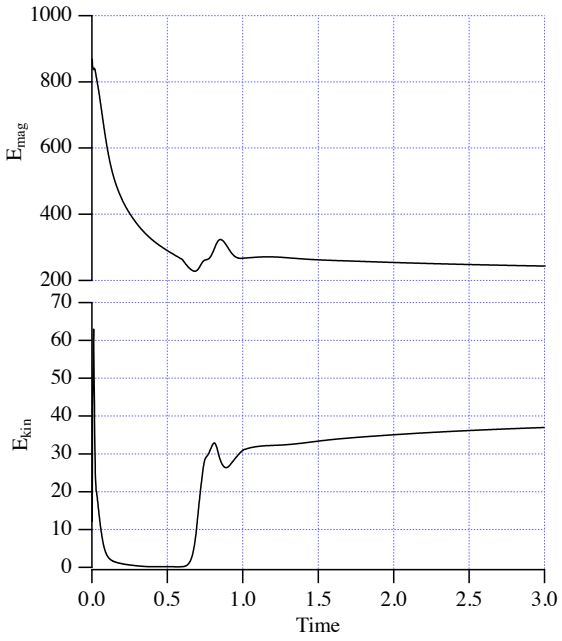

Figure 8: Time evolution of magnetic energy E_{mag} (top) and kinetic energy E_{kin} (bottom) for pseudo-vacuum boundary case to $t = 3.0$. These time series data sets are available at http://geodynamics.org/cig/community/workinggroups/geodyn/benchmark/.

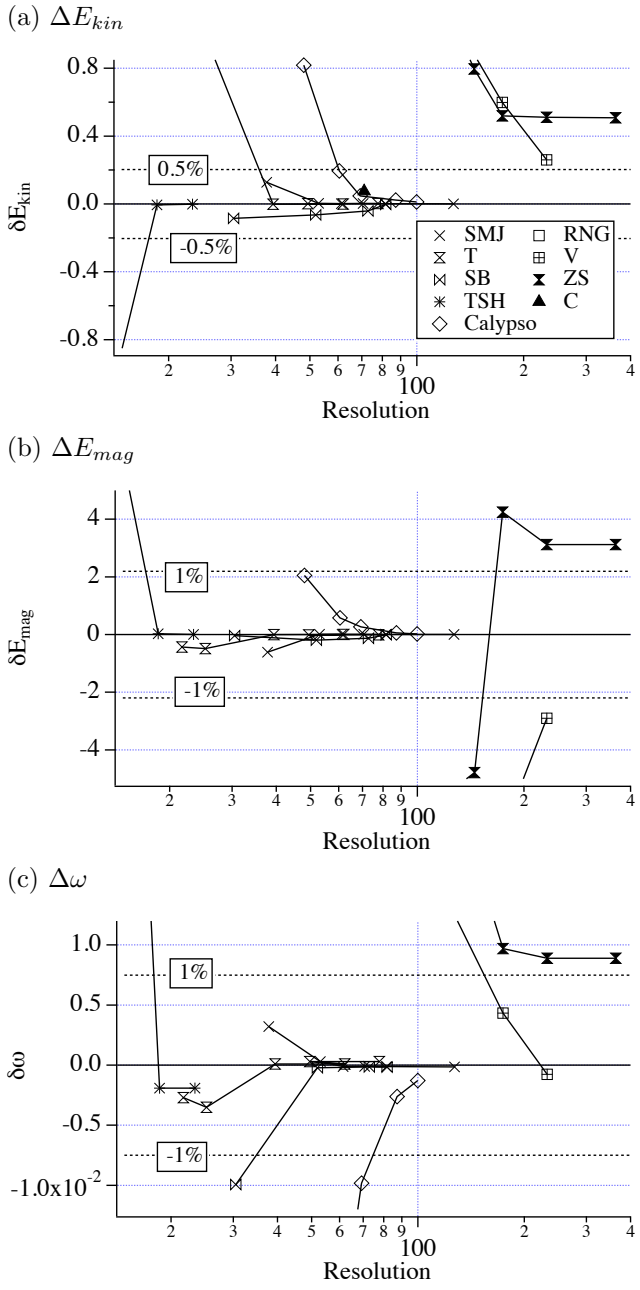

Figure 9: Convergence of the solution for pseudo-vacuum case. Error is evaluated from solution by highest resolution in Jackson *et al.*'s (2013) model. Relative error from highest solution by Jackson *et al.* (2013) ($\pm 0.5\%$ for E_{kin} and $\pm 1\%$ for E_{mag} and ω) is plotted by dotted line in each plot.

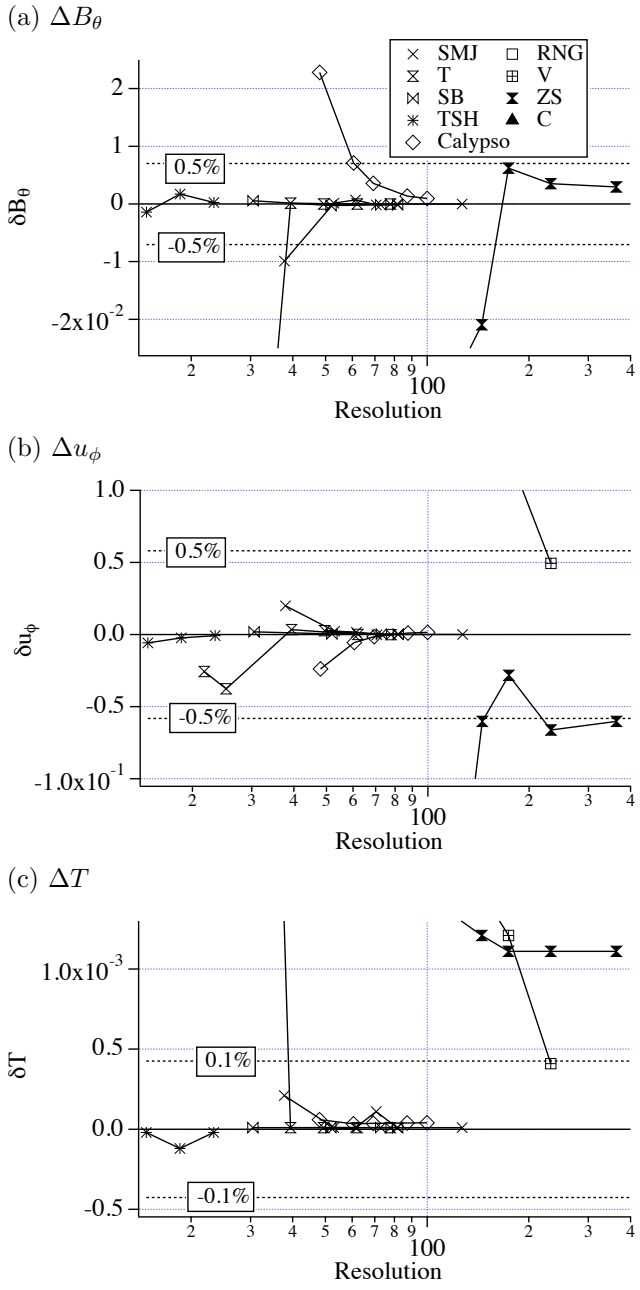

Figure 10: Convergence of the solution for pseudo-vacuum case. Error is evaluated from solution by highest resolution in Jackson *et al.*'s (2013) model. Relative error from highest solution by Jackson *et al.* (2013) ($\pm 0.5\%$ for B_θ and u_ϕ, and $\pm 0.1\%$ for T) is plotted by dotted line in each plot.

Table 6: Benchmark results for pseudo-vacuum boundary case using spherical harmonics expansion in Jackson *et al.* (2013) and from Calypso

				E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Suggested solution				40.677	219.39	0.74990	-11.636	1.4043	0.74990
Jackson *et. al.* (2013)									
Group	N_r	l_{max}	m_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
SMJ	50	32	29	40.806	218.78	0.4261	-11.6158	1.394	0.75312
SMJ	80	42	42	40.681	219.41	0.4259	-11.6334	1.404	0.7502
SMJ	96	48	48	40.680	219.41	0.4259	-11.6341	1.405	0.74978
SMJ	96	60	53	40.681	219.40	0.426	-11.6359	1.404	0.7498
SMJ	128	64	64	40.680	219.40	0.4259	-11.6357	1.404	0.74978
SMJ	200	100	100	40.679	219.40	0.4259	-11.6357	1.404	0.74976
TSH	33	32	64	39.815	225.54	0.42587	-11.6415	1.40291	0.805536
TSH	33	64	128	40.672	219.42	0.42577	-11.638	1.40605	0.747998
TSH	33	128	256	40.677	219.40	0.42587	-11.6365	1.4046	0.747998
T	21	21	21	41.623	218.96	0.4391	-11.6614	1.274	0.7472
T	33	21	21	41.688	218.91	0.4394	-11.6734	1.274	0.7464
T	33	42	42	40.N77	219.39	0.4259	-11.6323	1.404	0.750
T	33	84	84	40.677	219.40	0.4259	-11.6357	1.404	0.750
T	65	42	42	40.680	219.38	0.4259	-11.633	1.404	0.7502
T	65	84	84	40.681	219.38	0.4259	-11.6358	1.404	0.7502
SB	33	42	11	40.593	219.35	0.4259	-11.6339	1.405	0.73996
SB	33	64	65	40.614	219.20	0.4259	-11.6353	1.404	0.74968
SB	41	96	97	40.637	219.27	0.4259	-11.6355	1.404	0.74976
SB	129	64	65	40.675	219.39	0.4259	-11.6354	1.404	0.74974
RNG†	90	180	60	40.649	219.997	0.4224	-11.6597	1.42539	
Calypso									
	N_r	l_{max}	m_{max}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
	48	72	48	41.4953	221.441	0.425950	-11.6596	1.42714	0.663873
	96	72	48	40.8730	219.970	0.425924	-11.6413	1.41142	0.728405
	144	72	48	40.7254	219.660	0.425927	-11.6372	1.40789	0.740086
	288	72	48	40.7011	219.452	0.425929	-11.6347	1.40568	0.747291
	432	72	48	40.6890	219.412	0.425930	-11.6342	1.40526	0.748627

† Ribeiro, Nore, and Guermond (RNG) uses hybrid Fourier and finite element method using a Fourier decomposition in the azimuthal direction and the Lagrange elements P1-P2 in the meridian section. The three resolutions (90, 180, 60) refer to (N_r, N_z, N_ϕ) respectively.

Table 7: Benchmark results for pseudo-vacuum boundary case using local methods in Jackson *et al.* (2013)

			E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Suggested solution			40.677	219.39	0.74990	-11.636	1.4043	0.74990
Group	N_r	R_{lcl}	E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
V	64	116	41.904	205.941	0.4277	-11.3394	1.2858	0.76472
V	96	174	41.2765	212.543	0.4271	-11.511	1.33813	0.75424
V	128	232	40.9373	216.49	0.4263	-11.5864	1.37829	0.74914
ZS	50	91	42.4057	212.739	0.4274	-12.086	1.35962	0.7924
ZS	80	145	41.4735	214.604	0.4271	-11.696	1.38338	0.767
ZS	96	174	41.1971	223.639	0.427	-11.664	1.41054	0.7596
ZS	128	232	41.1888	222.515	0.427	-11.702	1.40785	0.7588
ZS	200	363	41.1856	222.51	0.427	-11.696	1.40728	0.7588
C	200	363	40.7524	204.83				0.786

Appendix C Accuracy Benchmarks Reporting Template

Member name:	
Name simulation code (If there):	
Simulation methods	

Insulated boundary benchmark

Solution at (between):					
E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Truncation of spectrum					
$N_{Chebyshev}$	l_{max}	m_{max}			R_{sph}
Grid resolution					
N_r	N_θ	N_ϕ	N_{sphere}		R_{lcl}

Pseudo vacuum boundary benchmark

Solution at (between):					
E_{kin}	E_{mag}	T	u_ϕ	B_θ	ω
Truncation of spectrum					
$N_{Chebyshev}$	l_{max}	m_{max}			R_{sph}
Grid resolution					
N_r	N_θ	N_ϕ	N_{sphere}		R_{lcl}

Notes for resolution (Truncation type, grid patten, etc.)
Notes for parallelization (Restriction, direction of domain decomposition, etc.)

Appendix D Performance Benchmark Results for Calypso

An example of scaling results of Calypso using insulated magnetic boundary condition is shown in Figure 11. We obtained almost same results for Calypso when we use the pseudo-vacuum boundary condition. We performed the simulation for 100 time steps by fixing the spatial resolution, and evaluate elapsed time for each step and for initialization. We evaluate the elapsed times total time, data communication and computation for s each time step. The elapsed time for initialization is also evaluated. The result show that the elapsed time for time integration keeps linear scalability by using less than 240 cores. The communication time is approximately 20% of the elapsed time when 16 cores are used. However, communication time is getting longer by using more than 320 cores and breaks linear scalability in the elapsed time. The time for initialization does not have any scalability in Calypso, but less than 30 seconds is short enough to run for 12 hours, which is the maximum length of the elapsed time for one job.

The strong scaling for the one time step using various spatial resolutions are also plotted in Figure 12. The results show that Calypso keeps linear scaling to the larger number of cores with increasing spatial resolution.

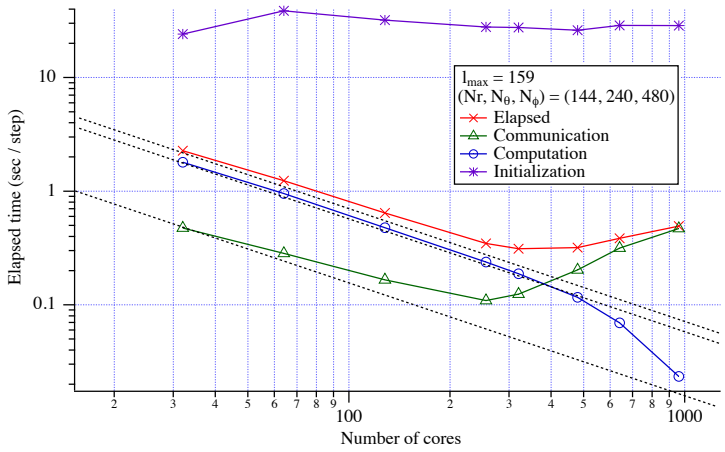

Figure 11: Elapsed time for each times step of Calypso on Stampede using one spatial resolution grid. Ideal scalabilities are plotted by dashed line by using elapsed times with 16 processor cores (One node) or with lowest number of cores. The maximum spherical harmonic degree employed is denoted here by the value of l_{max}.

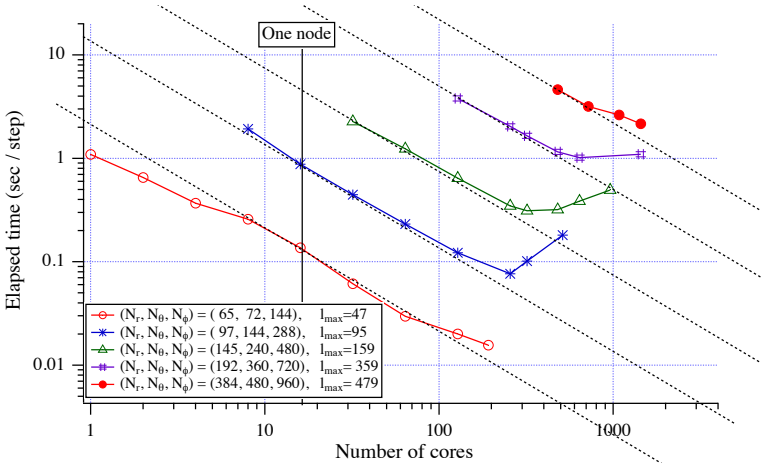

Figure 12: Elapsed time for each times step of Calypso on Stampede using various spatial resolutions. Ideal scalabilities are plotted by dashed line by using elapsed times with 16 processor cores (One node) or with lowest number of cores. The maximum spherical harmonic degree employed is denoted here by the value of l_{max}.

We also show the results for the weak scaling test by Calypso in Figure 13. The elapsed time does not change by changing the number of cores and radial resolution under the ideal scalability for Calypso. The result show that communication time rapidly increases and scalability is broken when 640 cores are used. The weak scaling for the horizontal direction is much more complicated than that for the radial resolution. Number of calculation for Fourier transform increases with $O(l_{max}^2 \log l_{max})$ for each sphere, and number of calculation for Legendre transform increases as $O(l_{max}^3)$ for each sphere. The Calypso's weak scaling for the horizontal resolution is shown in Figure 14. The elapsed time is between the ideal scaling for the Legendre transform and Fourier transform, but communication time rapidly increases when we use more than 256 cores in the present radial resolution.

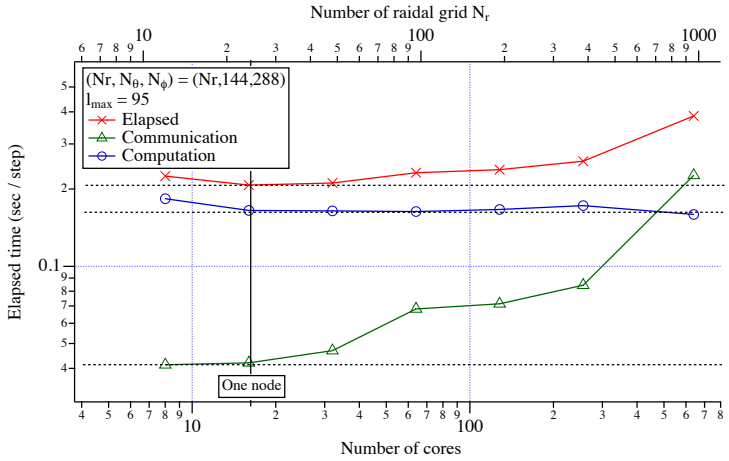

Figure 13: Weak scaling test of Calypso on Stampede with changing radial resolution as a function of number of cores (bottom axis) and radial resolution (top axis). The ideal scalability is plotted by dashed line by using elapsed times with 16 processor cores (One node). The maximum spherical harmonic degree is denoted here by l_{max}.

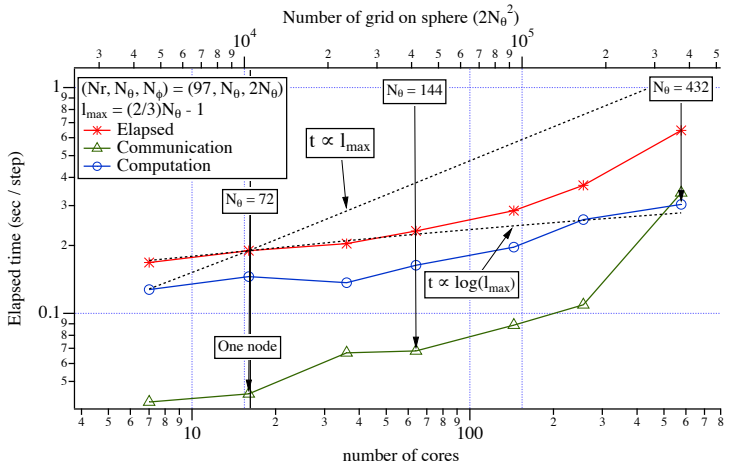

Figure 14: Weak scaling test of Calypso on Stampede with changing horizontal resolution as a function of number of cores (bottom axis) and horizontal resolution (top axis). The ideal scalability for the Legendre transform and Fourier transform are plotted by dashed lines. The maximum spherical harmonic degree is denoted here by l_{max}.

Appendix E Solution dependence on Initial temperature

In the present benchmark tests, initial temperature perturbation has Y_4^4 mode, and all solutions have a 90 degree symmetry ($m = 4$) with respect to the longitude. When we impose a larger temperature variation with the same spatial pattern as the given initial temperature, the solution will achieve the same value given in Tables 2 and 3. However, if initial temperature have a different spatial pattern from the given initial field, simulation goes to a different solutions. For example, If we impose a temperature perturbation with Y_3^3 mode, the simulation results have 120 degree symmetry ($m = 3$) or 72 degree symmetry ($m = 5$) with respect to the longitude and this solution can not sustain the magnetic field. When we impose initial temperature perturbation with Y_5^5 mode, the system can sustain dipolar magnetic field but results have 72 degree symmetry ($m = 5$) with respect to the longitude.

Appendix F Installation on Stampede

We introduce a brief install procedure of Calypso and ASH code on Stampede for an example. The TACC Stampede system is a 10 PFLOPS (PF) Dell Linux Cluster based on 6,400+ Dell PowerEdge server nodes, each outfitted with 2 Intel Xeon E5 (Sandy Bridge) processors and an Intel Xeon Phi Coprocessor (MIC Architecture). Nodes are interconnected with Mellanox FDR InfiniBand technology in a 2-level (cores and leafs) fat-tree topology. All Stampede nodes run CentOS 6.3 and are managed with batch services through SLURM 2.4. The Intel compiler is used, and OpenMP, two MPI implementation and various libraries are supported.

`Module` command is used to manage these installed libraries by each user. To check the available libraries, `module avail` is used as

```
login4$ module avail

------------------ /opt/apps/intel13/mvapich2_1_9/modulefiles ------------------
   amber/12.0                        petsc/3.4-complex
   espresso/5.0.3                    petsc/3.4-complexdebug
   fftw2/2.1.5                       petsc/3.4-cxx
   fftw3/3.3.2                       petsc/3.4-cxxcomplex
...
```

To use the installed libraries, `module load [library name]` command before compile and submitting job.

Stampede supports OpenMP for SMP parallelization within each node and two MPI implementations (MVAPITCH2 and Intel MPI). MVAPITCH2 is chosen as the default MPI, and we can switch the MPI library as following

```
login4$ module swap mvapich2 impi/4.1.1.036
```

or *vice versa*. To save the configuration of loaded modules as default setting, input as

```
login4$ module save
```

F.1 Installation of Calypso on Stampede

In this subsection, we describe the installation process of Calypso on Stampede. Calypso is written in Fortran 90, and fftw3 and parallel version of HDF5 library are required as external libraries. We need the following command before compile and submitting job as,

```
login4$ module load fftw3 phdf5
```

In Calypso, `configure` command is used to set valuables in Makefile. The following options are used in the `configure` command

```
Installation directories:
  --prefix=PREFIX         install architecture-independent files in PREFIX
                          [/usr/local]

Optional Features:
  --disable-option-checking  ignore unrecognized --enable/--with options
  --disable-FEATURE       do not include FEATURE (same as --enable-FEATURE=no)
  --enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
  --enable-fftw3          Use fftw3 library

Optional Packages:
  --with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
  --without-PACKAGE       do not use PACKAGE (same as --with-PACKAGE=no)
  --with-hdf5=yes/no/PATH full path of h5pcc for parallel HDF5 configuration

Some influential environment variables:
  FC          Fortran compiler command
  FCFLAGS     Fortran compiler flags
  PKG_CONFIG  path to pkg-config utility
  MPIFC       MPI Fortran compiler command
```

On Stampede, we use the following configuration option;

```
login4$ pwd
/home1/02179/matsui
login4$ cd Calypso
login4$ ./configure  --prefix='/home1/02179/matsui/local' FC=ifort AR=xiar
MPIFC=mpif90 PKG_CONFIG=/usr/bin/pkg-config FCFLAGS='-O3 -warn all -g'
--enable-fftw3 --with-hdf5
...
```

After generated Makefile, compile starts by using 'make' command as

```
login4$ make
```

The makefile generated by configure command is the following;

```
login4$ more Makefile
#
#  Makefile for Calypso geoynamo platform
#    Written by H. Matsui
#
SHELL           = /bin/sh
#
#  directories of Calypso
#
SRCDIR = /home1/02179/matsui/Calypso
INSTDIR= /home1/02179/matsui/local
#
```

```
MHDDIR = $(SRCDIR)/src
MAKEDIR= $(SRCDIR)/work
BUILDDIR= $(SRCDIR)/bin
#
# MPI settings
#
MPICHDIR =
MPICHLIBDIR = $(MPICHDIR)/lib
MPICHBINDIR = $(MPICHDIR)/bin
MPICHINCDIR =
MPILIBS =
#
#    compilers
#
F90_LOCAL = ifort
MPIF90 = mpif90
AR = xiar
RANLIB = ranlib
#
#  optimization flags
#
F90OPTFLAGS= -O3 -warn all -g   -fopenmp
#
#    FFTW3 settings
#
FFTW3_CFLAGS= -I/opt/apps/intel13/mvapich2_1_9/fftw3/3.3.2/include
FFTW3_LIBS=   -L/opt/apps/intel13/mvapich2_1_9/fftw3/3.3.2/lib -lfftw3 -lm
#
#    HDF5 fortran wrapper settings
#
HDF5_FFLAGS=    -I/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib -L/opt/apps/int
el13/mvapich2_1_9/phdf5/1.8.9/lib -I/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/i
nclude
HDF5_LDFLAGS=   -L/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib
HDF5_FLIBS=     -fPIC -I/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/include -L/op
t/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib /opt/apps/intel13/mvapich2_1_9/phdf5
/1.8.9/lib/libhdf5hl_fortran.a /opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib/li
bhdf5_hl.a /opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib/libhdf5_fortran.a /opt
/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib/libhdf5.a -L/opt/apps/intel13/mvapich
2_1_9/phdf5/1.8.9/lib -Wl,-rpath,/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib
-L/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib -lsz -lz -limf -lm -lsz -lz -lm
 -Wl,-rpath -Wl,/opt/apps/intel13/mvapich2_1_9/phdf5/1.8.9/lib
#
#  --- Please do not change the following lines ---
#
MAKE_MOD_DEP= $(BUILDDIR)/make_f90depends
```

```
#

all: makemake
	cd $(MAKEDIR); make

$(MAKE_MOD_DEP): $(MHDDIR)/module_dependency/make_module_dependency.f90
	if [ ! -d $(BUILDDIR) ]; then \
		mkdir $(BUILDDIR); \
	fi
	$(F90_LOCAL) $< -o $@

makemake: $(MAKE_MOD_DEP)
	echo "# Construct Makefile"; \
	cd $(MHDDIR) ; \
		make \
		MAKEDIR="$(MAKEDIR)" \
		BUILDDIR="$(BUILDDIR)" \
		INSTDIR="$(INSTDIR)" \
		MHDDIR="$(MHDDIR)" \
		MPICHDIR="$(MPICHDIR)" \
		MPICHLIBDIR="$(MPICHLIBDIR)" \
		MPILIBS="$(MPILIBS)" \
		MPICHBINDIR="$(MPICHBINDIR)" \
		MPICHINCDIR="$(MPICHINCDIR)" \
		FFTW3_CFLAGS="$(FFTW3_CFLAGS)" \
		FFTW3_LIBS="$(FFTW3_LIBS)" \
		HDF5_FFLAGS="$(HDF5_FFLAGS)" \
		HDF5_LDFLAGS="$(HDF5_LDFLAGS)" \
		HDF5_FLIBS="$(HDF5_FLIBS)" \
		FC_NAME_MANGLE="$(FC_NAME_MANGLE)" \
		OPTFLAGS="$(OPTFLAGS)" \
		F90OPTFLAGS="$(F90OPTFLAGS)" \
		F90LIB="$(F90LIB)" \
		MPIF90="$(MPIF90)" \
		AR="$(AR)" \
		RANLIB="$(RANLIB)" \
		MAKE_MOD_DEP="$(MAKE_MOD_DEP)" \
		makemake


install:
	cd $(MAKEDIR) ; \
		make install

clean:
	for dir in $(MAKEDIR) $(MHDDIR) ; do \
```

```
echo "# cleaning $${dir} directory..."; \
        ( cd $${dir}; \
        make clean )\
done; \
rm -f mpif.h *.o *.mod *~ *.par *.diag *.a *.f90
```

F.2 Installation of ASH code on Stampede

The ASH code is written in Fortran 90/95, parallelized using MPI, and is built relatively straightforwardly using the Intel compiler. It relies on three packages to access external libraries available on Stampede:

1. Intel MPI (parallelization)
2. FFTW (FFTs for spectral transforms)
3. Intel's MKL (matrix LU-decomposition and inversion routines)

Note that is always a good idea to check the TACC website to see which packages they recommend using (http://www.tacc.utexas.edu/user-services/user-guides/stampede-user-guide). Examples of ASH makefiles are provided at the end of this section. These may be useful should you wish to use a similar set of packages for your own code. Before following these examples, you will need to first load the FFTW and Intel MPI modules. This is accomplished via the following steps:

1. First, check which MPI installation you have loaded by typing:
`login2% module list`
2. If mvapich2 is loaded, load Intel's MPI by typing:
`login2% module swap mvapich2 impi`
3. Next, load FFTW by typing:
`login2% module load fftw3`
4. Finally, save this as your default setting by typing:
`login2% module save`

These modules will now be loaded by default whenever you login. The MKL library is included as part of the Intel compiler (loaded by default) and does not need to be explicitly loaded.

The directory structure of ASH contains three directories, largely for historical reasons, that all exist at the same level. The bulk of code resides in a directory named "src." This is where the executable is built. A second directory, "lib," contains modules that are compiled separately and then soft linked into the "src" directory before building the ASH executable. A third directory, "include", contains various macros and preprocessor definitions that are referenced by the files in "src."

The makefiles for "lib" and "src" are provided below. The "lib" makefile is used by typing *make all* from within "lib." This compiles all modules whose source code exists within that directory. The "src" makefile is used to build the executable by typing *make ASH_FD* from within the "src" directory. You will notice that these two makefiles reference the following environment variables:

1. TACC_FFTW3_LIB
2. INCLUDE
3. MLKROOT

These environment variables are each set once you have loaded the modules indicated above.
The lib makefile is structured as:

```
.SUFFIXES: .F .o .m4 .mod .c

FC = mpif90
MKLI = $(MKLROOT)/include
CC = icpc

FFLAGS = -fpp -FR -r8 -O3 -shared-intel -xhost -I$(MKLI) -I$(TACC_FFTW3_INC)

CFLAGS = -xhost -I../include

.F.o:
        $(FC) $(FFLAGS) -I ../include -c $<

.cc.o:
        $(CC) $(CFLAGS) -c *.cc


all :
        make MPI_ASH.o Miscellaneous.o Chebyshev_Transforms.o \
        Numerical_Library.o Fourier_Transforms.o Communication_Library.o \
        Linear_Algebra.o Send_Receive.o Reduce.o All_to_All.o fft.o \
        System_Interface.o


Numerical_Library.o : Linear_Algebra.o Fourier_Transforms.o Chebyshev_Transforms.o
Fourier_Transforms.o : fft.o
Communication_Library.o : Send_Receive.o Reduce.o All_to_All.o
Miscellaneous.o System_Interface.o
Send_Receive.o : MPI_ASH.o
Reduce.o : MPI_ASH.o
fft.o: fft_timing.o
System_Interface.o: System_Interface.F

clean:
        touch temp.o
        touch temp.mod
        rm *.o *.mod
```

The src makefile is structured as:

```
.SUFFIXES: .o .F

FC = mpif90
GENINC = $(INCLUDE)

FFLAGS = -FR -r8 -O3 -shared-intel -xhost -I../include -I$(GENINC) -I$(TACC_FFTW3_INC)

LIBS = -L$(TACC_FFTW3_LIB) -lfftw3f -mkl=sequential

.F.o :
        $(FC) $(FFLAGS) -c $<

ASH_FD: Main.o
        $(FC) $(FFLAGS) -o ash_fd *.o $(LIBS)

Angular_Averages.o: Auxiliaries.o
Auxiliaries.o: Auxiliaries.F Constants.o Derivatives.o Partition.o
ReferenceState.o State.o
        $(FC) $(FFLAGS) -c $<
BoundaryConditions.o: Constants.o Controls.o Partition.o PhysicalParameters.o
ProblemSize.o ReferenceState.o Checkpointing.o
Controls.o: Constants.o
Debugging.o: Constants.o Utilities.o ProblemSize.o
Derivatives.o: Constants.o Debugging.o Partition.o TimeProfile.o
PhysicalParameters.o ProblemSize.o SphericalTransform.o
Implicit.o: BoundaryConditions.o Constants.o Controls.o Derivatives.o
Partition.o TimeProfile.o PhysicalParameters.o ProblemSize.o ReferenceState.o
State.o TemporalParameters.o Utilities.o
Initial_Conditions.o: Constants.o Controls.o Derivatives.o Implicit.o
Partition.o ProblemSize.o ReferenceState.o State.o TemporalParameters.o Auxiliaries.o
Input.o: BoundaryConditions.o Constants.o Controls.o Debugging.o Initial_Conditions.o
PhysicalParameters.o ProblemSize.o ReferenceState.o TemporalParameters.o
Output.o Derivatives.o
Main.o: Constants.o Controls.o Debugging.o Input.o Partition.o TimeProfile.o
Physics.o ReferenceState.o TemporalParameters.o
Output.o: Angular_Averages.o Shells_and_Slices.o Bulk_Output.o Initial_Conditions.o
Partition.o: Constants.o Controls.o Debugging.o ProblemSize.o Utility_IO.o
TimeProfile.o : Partition.o
PhysicalParameters.o: Constants.o Controls.o TemporalParameters.o
Physics.o: Constants.o Controls.o Implicit.o Partition.o TimeProfile.o
PhysicalParameters.o ProblemSize.o ReferenceState.o State.o
TemporalParameters.o Utilities.o
ProblemSize.o: Constants.o Controls.o
ReferenceState.o: Constants.o Controls.o Debugging.o Derivatives.o Partition.o
PhysicalParameters.o State.o TemporalParameters.o Utilities.o Utility_IO.o
SphericalTransform.o: Constants.o MachineParameters.o Partition.o TimeProfile.o
```

```
ProblemSize.o
State.o: Constants.o Partition.o
TemporalParameters.o: Constants.o
Utilities.o: Constants.o
Utility_IO.o: Constants.o ProblemSize.o


clean :
        touch temp.o
        touch temp.mod
        rm *.o
        rm *.mod
        rm ash_fd
        ln -s ../lib/*.mod .
        ln -s ../lib/*.o .
ready :
        ln -s ../lib/*.mod .
        ln -s ../lib/*.o .
```