In-situ Parallel Visualizations for Geodynamo Simulations Using Calypso

Hiroaki Matsui[(1)], Yangguang Liao[(1)], Takumi Kera[(2)], Yuto Katoh[(2)], Magali Billien[(1)], and Louise Kellogg[(1)]
(1) Univ. of California, Davis, (2) Tohoku University

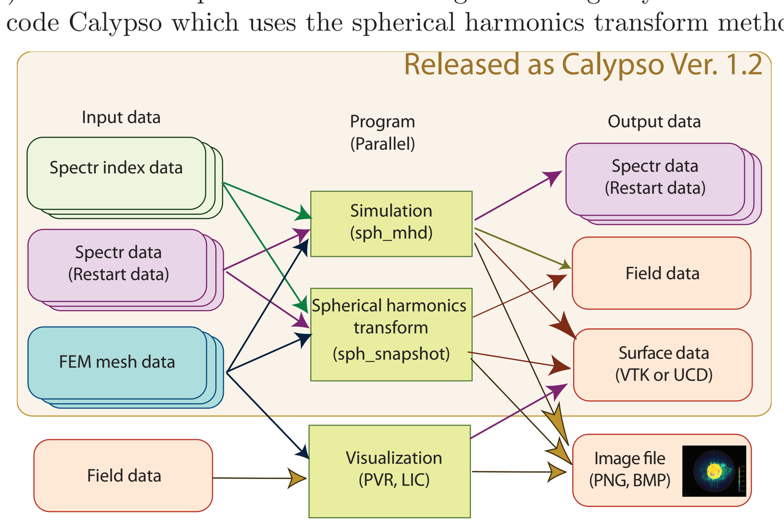
UCDAVIS DEPARTMENT OF EARTH AND PLANETARY SCIENCES

NSF

CIG COMPUTATIONAL INFRASTRUCTURE for GEODYNAMICS

Introduction

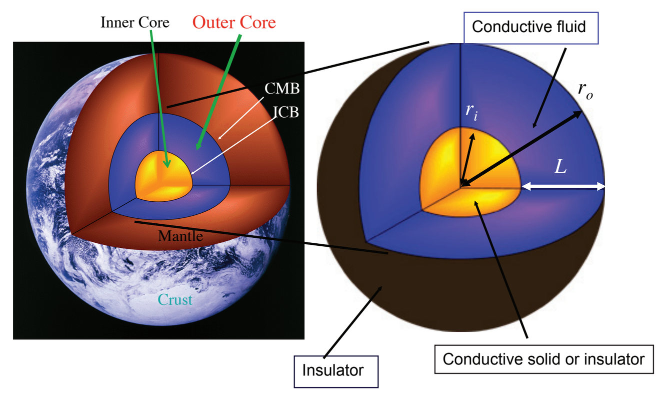
- In the last 20 years, geodynamo simulations have successfully represented some characteristics of the geomagnetic field and clarified the basic dynamics of the fluid motion and generation of the geomagnetic field.
- However, the parameter regime for these numerical dynamos is still far from the actual parameter regime for the outer core. To approach Earth parameter regimes, a massively parallel computational environment is required to achieve significantly finer spatial resolution.
- Large scale simulation on massively parallel computer generate a huge size of data output. We need data visualization during the dynamo simulation to reduce the data output with keeping the time resolution.
- We implement parallel volume rendering and line integral convolution (LIC) modules for a parallel unstructured grid into a geodynamo simulation code Calypso which uses the spherical harmonics transform method.



Model of geodynamo simulation

Model of the dynamo simulation

- A rotating spherical shell from the inner core boundary (ICB) $r = r_i$ to the core mantle Boundary (CMB) $r = r_o$.
- Shell thickness is set to $L = r = r_o - r_i = 1$, and the ratio of the inner core radius to the outer core radius is $r_i/r_o = 0.35$.
- The fluid shell is filled by conductive Boussinesq fluid with constant kinetic viscosity ν, thermal diffusivity κ, and magnetic diffusively η.
- Inner core co-rotates with mantle.
- Inner core has the same thermal conductivity as the outer core, and homogeneous heat source is given to conserve the heat in the entire core. Consequently, $\frac{\partial T}{\partial t} = \frac{1}{Pr}\frac{\partial^2}{\partial x_i x_i}T + Q$ is solved in the inner core.



Rotating spherical shell modeled on the Earth's outer core for geodynamo simulation.

Governing equations

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

and

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

where, $\boldsymbol{u}$, P, $\boldsymbol{B}$, and T are the velocity, pressure, magnetic field, and temperature, respectively.

Control parameters

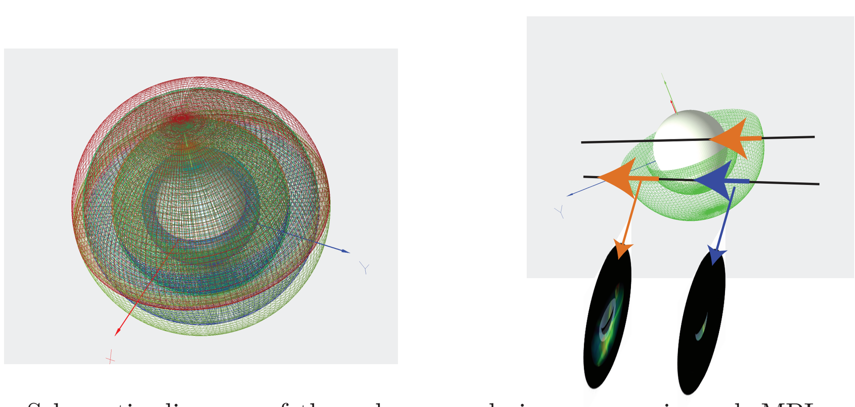
The following dimensionless numbers are required for the thermally driven dynamo model.

$$
\begin{aligned}
\text{Prandtl number: } Pr &= \frac{\nu}{\kappa} \\
\text{Ekman number: } E &= \frac{\nu}{\Omega L^2} \\
\text{Rayleigh number: } Ra &= \frac{\alpha g \beta_o L^4}{\kappa\nu\chi} \\
\text{magnetic Prandtl number: } Pm &= \frac{\nu}{\eta}
\end{aligned}
$$

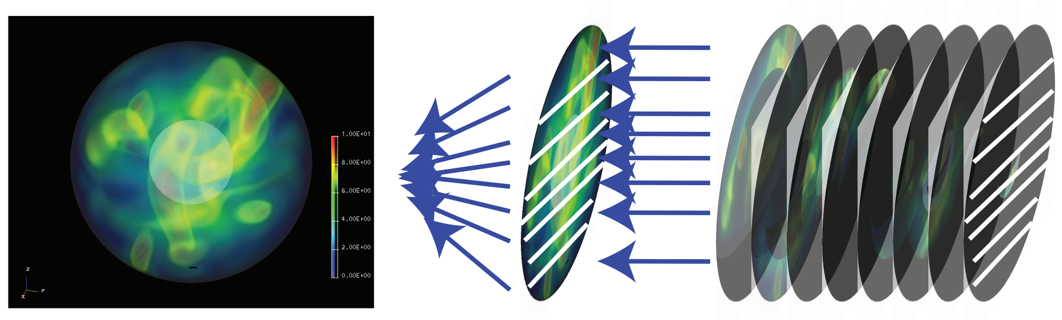
where, $-\beta_o$ and $\chi = r_o/r_i$ are the temperature gradient at the outer boundary and ratio of the inner core radius to the outer core radius, respectively.

Parallel rendering module (Volume and surface renderings)

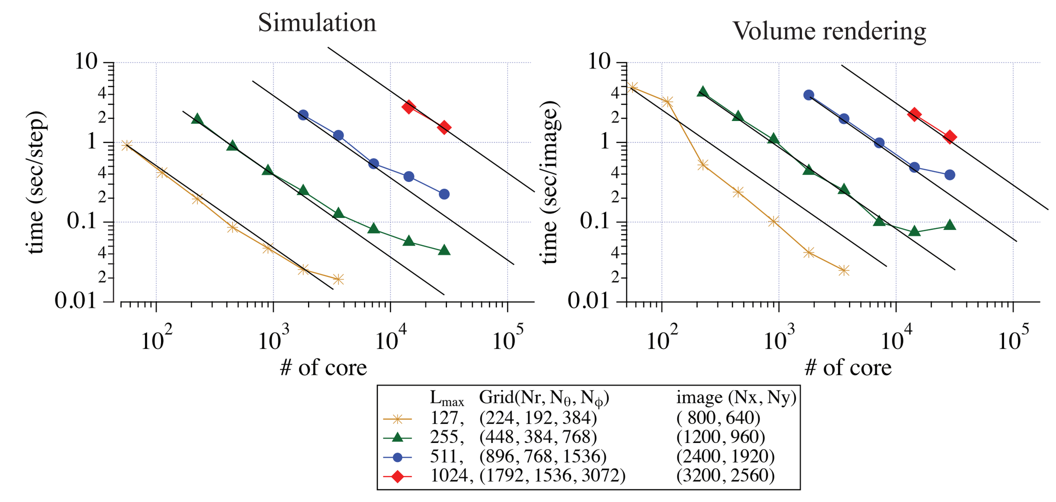
Volume rendering routine is connected to the dynamo simulation program. Consequently, volume rendering images are written during the simulation.



Schematic diagram of the volume rendering process in each MPI process. The spherical shell is divided into subdomains with respect to the parallelization of the dynamo simulation. Each MPI process has one subdomain, but each process needs to generate more than one sub images because the ray for rendering may go through the subdomain more than once.



Schematic diagram of the image collection for the volume rendering. Images for subdomains in each process are re-distributed to the all process for composition, and all process compose segmented images with respect to the distance from viewpoint. finally, the composited images are collected into one process and output as a PNG or BMP file. Consequently, global communication requires twice to construct each image.



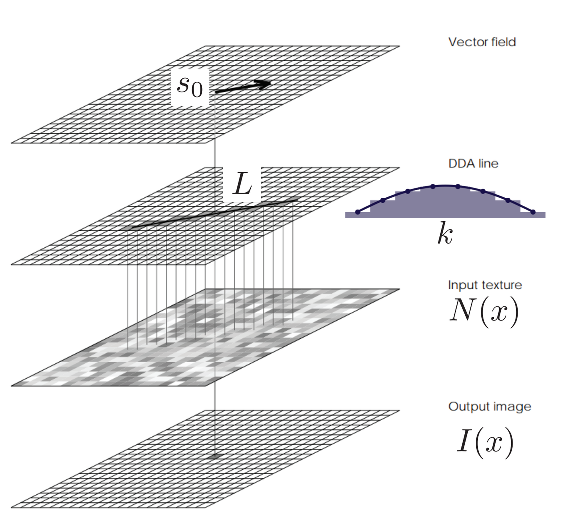
Strong scaling on TACC frontera for time integration and volume rendering module of Calypso. The elapsed times for each time step is plotted on the left, and elapsed time to generate each image is plotted on the right. The volume rendering is performed with time integration. Ideal scaling is plotted by black lines. Volume rendering for one image requires approximately three times of the one time integration. However, required time for visualization is small enough because less than 10 images are written every 200 too 1000 time steps in the productive runs.

Parallel line integeral convolution (LIC) rendering module

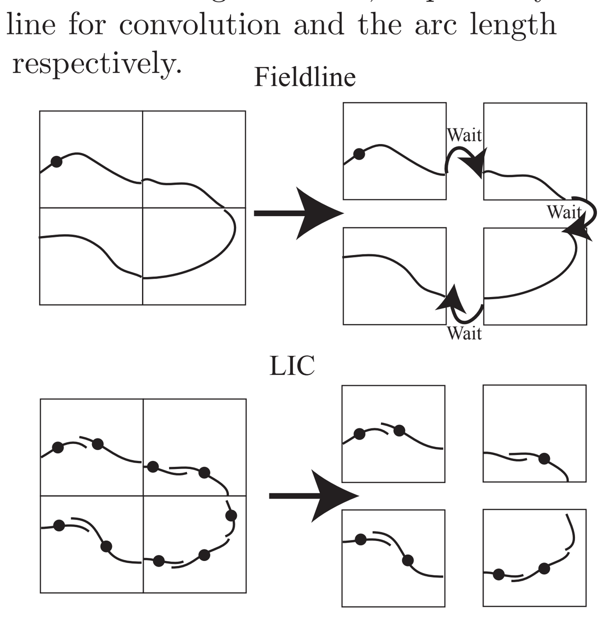
LIC is first introduced by Cabral and Leedom (1993) as

$$I(x_0) = \int_{s_0-L/2}^{s_0+L/2} k(s-s_0)N(\sigma(s))ds,$$

where I, $N(x)$, k are the intensity of the pixel, represents the noise texture, and a kernel function to smoothly taper off contributing field lines, respectively. L and s are the the length of the field line for convolution and the arc length parameterization of the field line curve, respectively.

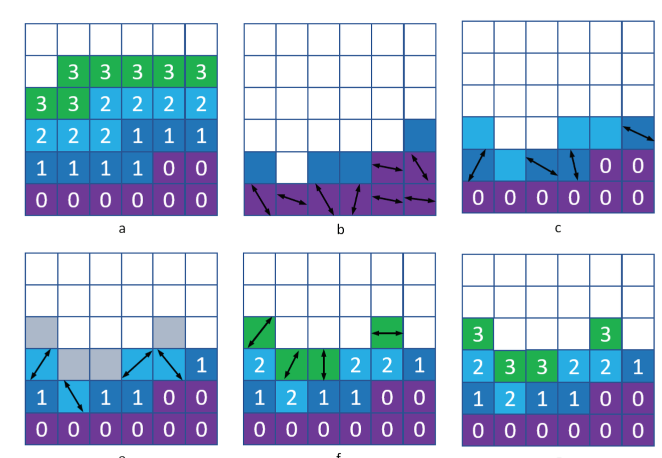


2D schematic diagram for LIC calculation for each pixel. In 3D LIC, obtained LIC value $I(x)$ is rendered by volume rendering.

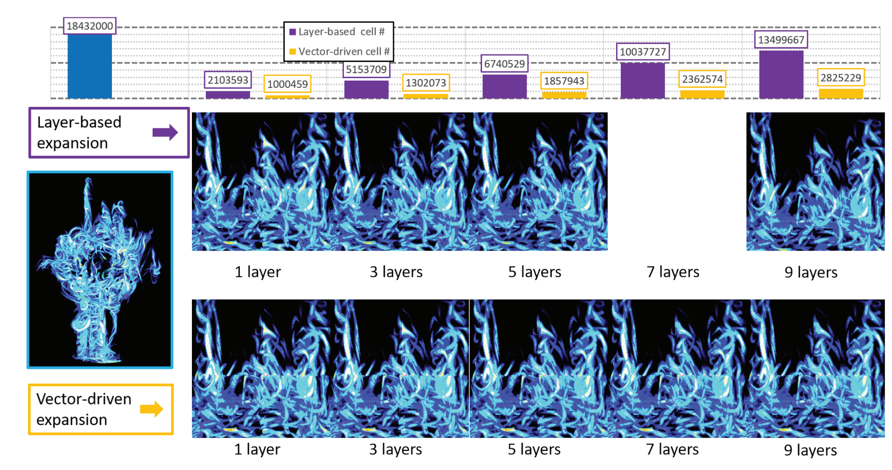


Comparison between parallel field line tracing and LIC. Longer line tracing makes longer waiting time to finish line tracing in other subdomains

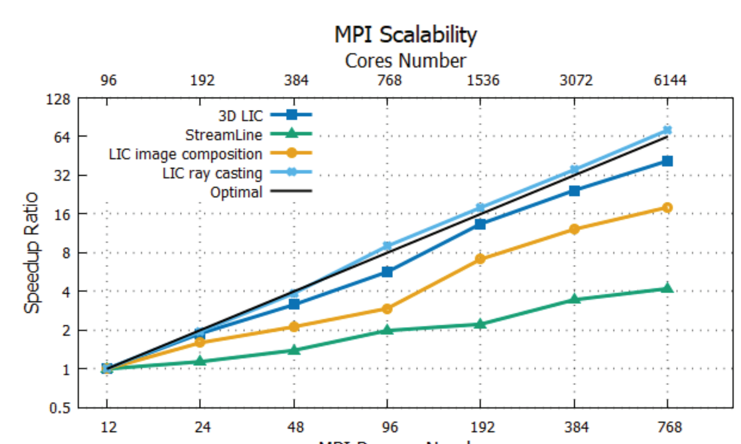
The image quality of the LIC visualization depends on sleeve size of each subdomain. We need to find optimal sleeve size to conserve memory size.



A 2D schematic diagram of the difference between vector-driven external expansion method and simple layer-based expansion method. (a) is an example of 3 layers expansion by using the simple layer-based method. Panels from (b) to (g) show the procedure of our method to expand the external cells by the vector features of each cell.



Comparison of image quality (bottom panels) of the sleeve size and number of total external cells (top panels). The top graph shows the number of external cells generated by different methods using different numbers of layers.



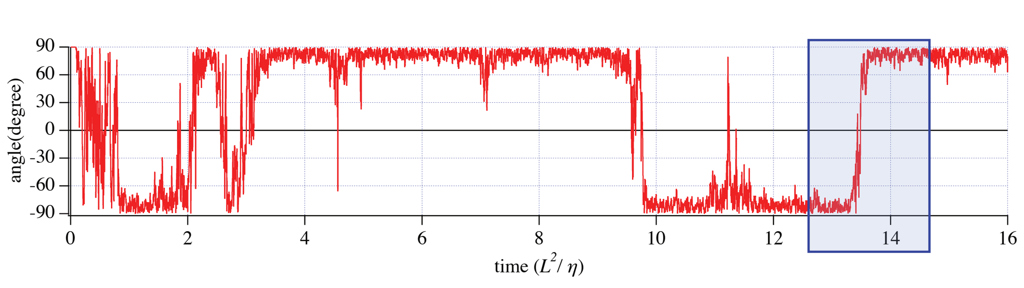
Strong scaling of LIC computation on TACC Stampede 2. LIC computation has much better scaling than the field line tracing. In LIC visualization, collecting images for subdomains into one image reduces the scaling due to the data communication.

Visualization results by Calypso

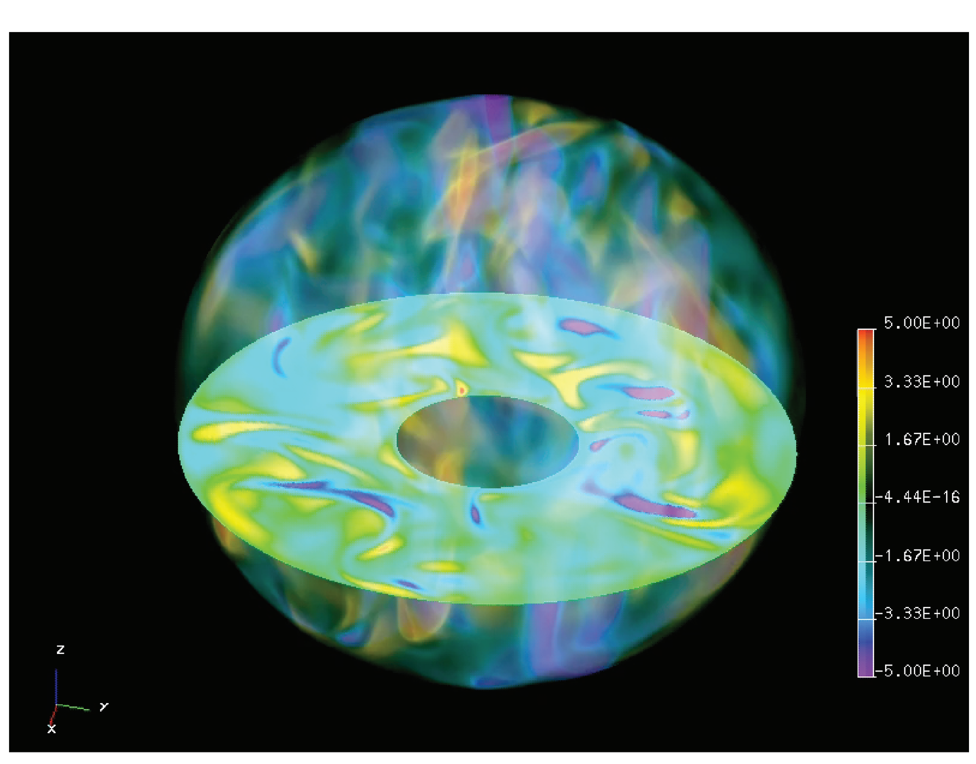
The first example is a dynamo simulation includes magnetic dipole reversal to display volume rendering results of the z-component of the magnetic field, and the second example is the simulation with lower Ekman number to visualize a snapshot and time evolution of the magnetic field.

List of parameters of the simulations

		Case 1	Case 2
Resolution:	(N_r, l_{max})	(225, 127)	(160, 159)
Prandtl:	$Pr = \nu/\kappa$	1.0	1.0
magnetic Prandtl:	$Pm = \nu/\eta$	5.0	1.0
Ekman:	$E = \nu/\Omega L^2$	6.0×10^{-4}	1.0×10^{-5}
Rayleigh:	$Ra = \alpha g \Delta T L^3/\kappa\nu$	4.93×10^5	2.24×10^8



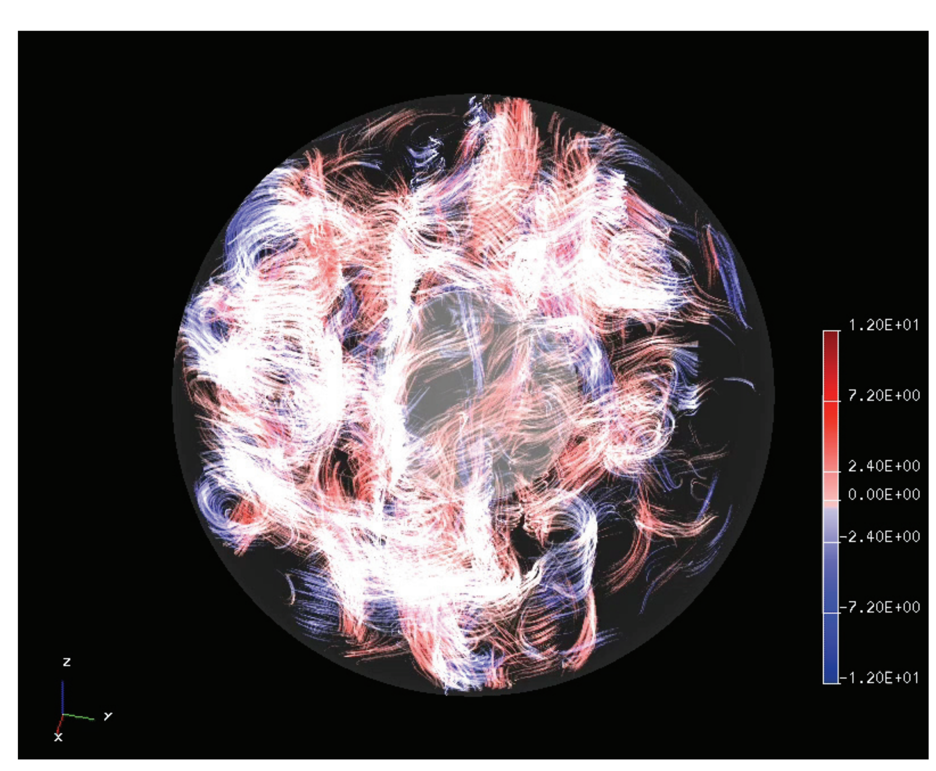
Time evolution of the direction of the dipole component of the magnetic field for Case 1. The duration for the Movie 1 and 2 is shown by the blue box.



Volume rendering of z-component of the magnetic field for Case 1 at the starting time for Movie 1 and 2. Result on the section $z = -0.3 = 0.56r_i$ is also plotted in the image.

Movie 1: Time evolution of the z-component of the magentic field for Case 1.

Movie 2: Time evolution of the temperature for Case 1.



Magnetic field visualized by parallel LIC rendering for Case 2. The outward (i.e. $B_r > 0$) line is displayed with red, and inward magnetic field is displayed blue. The movie for the snapshot with changing viewpoint is shown in Movie3, and time evolution is shown by Movie 4.

Movie 3: Snapshot of the magnetic field for Case 2 with changing the view point.

Movie 4: Time evolution of the magnetic field for Case 2.

Summary

- We develop a magnetohydrodynamics (MHD) simulation code Calypso for rotating spherical shell, and implement a parallel visualization modules into Calypso. Visualization modules consists of volume rendering, cross sectioning, iso-surfacing, and line integral convolution modules.
- These visualization modules are for data on parallel unstructured grid using hexahedral elements.
- We tested the parallel performance of the volume rendering module on TACC Frontera up to 28672 processor cores with connecting simulation model, and the both simulation and volume rendering keeps good strong scaling to 28672 cores.
- To visualize vector fields (e.g. velocity or magnetic field), a parallel visualization system that first uses parallel 3D line integral convolution to visualize flow data, and parallel 3D LIC have a good strong-scaling up to 6144 cores on TACC Stampede 2.
- The vector-driven external cell expansion method for our 3D LIC, which significantly reduces the extra memory cost for external cell.
- Current LIC module has large load imbalance for the in-situ visualization because we apply the same domain decomposition for LIC as that for the simulation. An parallel adjustive volume-based re-partitioning is required to obtain more balanced parallel workload for LIC.

Acknowledgement

This study is supported by NSF awards EAR-1550901. The present study is performed on XSEDE TACC Frontera and Stampede 2.