



Fully Compressible Magnetohydrodynamic Simulations of Solar Convection Zones with CHORUS++

Aidan Paoli* and Chunlei Liang †

Department Mechanical and Aerospace Engineering, Clarkson University, Potsdam, NY, United States

The objective of this study is to develop a fully compressible magnetohydrodynamic solver for fast simulations of the global dynamo of the Sun using unstructured grids and GPUs. Solar activity largely dictates Earth's immediate space environment and atmosphere. Accurate modeling of the Sun's convective layers is vital to predicting the Sun's behavior, including solar dynamo and sunspot cycles. Similarly, better understanding of convection in gas giants like Jupiter and Saturn requires global simulations of their convective layers. Global simulations modeling convective layers inside stars and planets are computationally strenuous. Currently, there are many efficient codes capable of conducting these large simulations; however, many make assumptions of anelastic density distribution. The anelastic assumption is capable of producing accurate results for low mach numbers; however, it fails in regions with a higher mach number and a fully compressible flow must be considered. Many of these codes are also required to use a structured grid for the use of spherical harmonics. To avoid these issues, Wang et al. [1] created a Compressible High-ORder Unstructured Spectral difference (CHORUS) code. CHORUS is a fully compressible, high-order, spectral difference code for simulating fluid dynamics inside stars and planets. CHORUS++ augmented the CHORUS code to adopt a higher degree of polynomials by using cubed-sphere meshing and transfinite mapping to perform simulations on unstructured grids [2]. Two hydrodynamic benchmark tests of Jupiter and the Sun were used to test the CHORUS code [1, 2]. Recently, CHORUS++ was further developed for parallel magnetohydrodynamic (MHD) solutions on GPUs at Clarkson University. This study presents CHORUS-MHD solutions for two dynamo benchmark problems similar to the one proposed by Chen et al. [2]. We extended the solar benchmark problems presented by Chen et al. [2] to unsteady solar dynamo problems, with two different density scale heights. The CHORUS-MHD code is further accelerated by GPUs and used to successfully solve these solar dynamo benchmark problems. Both solar benchmark problems are run using a 6th-order spectral difference method on multiple GPUs. CHORUS-MHD can be further used to simulate the sunspot cycle and model the polar vortices of the Sun [3] because of its flexibility in meshing.

I. Introduction

The formation and development of the solar system were largely dictated by the Sun. Today, the immediate space environment is heavily influenced by solar activity, which is closely correlated to magnetic field activity in the Sun, sunspots, and other solar phenomena. By improving our understanding of the Sun, we can ultimately begin to predict its behavior and understand the evolution of the solar cycle including sunspot cycles, solar flares, and magnetic field polarity. To better understand these phenomena, we must investigate the convective layers of the Sun, where large amounts of charged plasma is moved, producing magnetic fields. Modeling these convective layers is often computationally taxing and difficult due to the vast difference in length scales and timescales in planets and stars, though there are multiple successful approaches. Many models use an anelastic density approximation, allowing for the implementation of spherical harmonics to solve governing equations; however, there are drawbacks. First, the anelastic approximation and spherical harmonics require the code to be used on spherically structured shells, which results in lower meshing quality at the poles. Simultaneously, the anelastic approximation begins to lose accuracy as the Mach number approaches one, as seen in convection in red giants and modeling of convection into the photosphere where sunspots and granulation occurs. To accurately model the convection of the Sun with a fully compressible code, CHORUS was developed. CHORUS is a fully Compressible High-ORder Unstructured Spectral difference code for modeling fluid dynamics in spherical shells, originally developed by Wang et al. [1]. CHORUS uses conservation of mass, energy, and momentum

*Undergraduate Student, AIAA Student Member, Membership Number: 1400547

†Professor, AIAA Associate Fellow, Membership Number: 303971

equations to solve hydrodynamic problems in spherical shells. CHORUS++ extended the code to include transfinite mapping to a computational domain, and an unstructured cubed sphere meshing technique. The unstructured nature allows CHORUS to work for oblate shells such as Saturn and other rapidly rotating bodies. The cubed-sphere meshing technique also provides a more uniform and refined grid resolution, especially in the polar regions, and singularities can be avoided at the north and south poles. In smaller-size stars, where deep convection is typically at a low Mach number and incompressible flow can be assumed, the fully compressible nature of CHORUS is not ideal. However, CHORUS can simulate problems that are not achievable by incompressible codes such as red giant convection, photospheric convection with sunspots, and granulation, where compressible flow needs to be considered. Recently, CHORUS has been extended to CHORUS-MHD, which includes additional induction equations and an equation to clean the divergence of the magnetic field [4]. CHORUS-MHD is also capable of being run on multiple GPUs to speed up simulation time.

II. MHD Governing Equations

CHORUS considers the fluid dynamics in a hollow spherical shell. The reference frame of the simulation rotates uniformly about the z-axis with the spherical shell's angular speed. It is assumed that most of the mass is concentrated inside the shell, resulting in simpler gravitational terms. The governing equations consist of conservation equations for mass, total energy, linear momentum, and the magnetic induction, in addition to a divergence cleaning equation of the magnetic field. The general form of these equations can be expressed in Equation 1.

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \bar{\mathbf{F}} = \mathbf{M} \quad (1)$$

Where \mathbf{Q} is the vector of conserved variables, \mathbf{M} is the vector of the source term, and $\bar{\mathbf{F}}$ is the flux vector with components $F\hat{\mathbf{x}}, G\hat{\mathbf{y}}, H\hat{\mathbf{z}}$. \mathbf{Q} is defined in Equation 2.

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho \mathbf{U} \\ E \\ \mathbf{B} \\ \psi \end{pmatrix} \quad (2)$$

Where ρ is the density, \mathbf{U} is the velocity vector, E is the total energy, \mathbf{B} is the magnetic field vector and ψ is the divergence cleaning term of the generalized Lagrange multiplier (GLM) for the magnetic field [5]. The total energy density is composed of internal energy, kinetic energy, magnetic field energy, and a divergence cleaning term. The total energy density is defined as

$$E = e + \frac{1}{2}\psi^2 = \frac{p}{\gamma - 1} + \frac{1}{2}\rho\|\mathbf{U}\|^2 + \frac{1}{2}\|\mathbf{B}\|^2 + \frac{1}{2}\psi^2 \quad (3)$$

Where p is the pressure, γ is the ratio of specific heats, and $\|\cdot\|$ is the Euclidean vector norm. We assume an ideal gas so that

$$p = \rho RT \quad (4)$$

With R being the specific gas constant, and T is the temperature. The vector of sources term consists of terms for the gravitational force and the gravitational potential energy, Coriolis force, and additional terms for cleaning divergence [5]. The complete source vector is

$$\mathbf{M} = \begin{pmatrix} 0 \\ \rho\mathbf{g} - 2\rho\boldsymbol{\Omega} \times \mathbf{U} - (\nabla \cdot \mathbf{B})\mathbf{B} \\ \rho\mathbf{U} \cdot \mathbf{g} - (\nabla \cdot \mathbf{B})\mathbf{U} \cdot \mathbf{B} \\ -(\nabla \cdot \mathbf{B})\mathbf{U} \\ -a\psi \end{pmatrix} \quad (5)$$

Where $\mathbf{g} = \frac{GM}{r}\hat{\mathbf{r}}$ is the acceleration due to gravity, and $\boldsymbol{\Omega} = \Omega\hat{\mathbf{z}}$ is the rotation rate. The flux term $\bar{\mathbf{F}}$ is composed of

inviscid and viscous fluxes as in $\bar{\mathbf{F}} = \bar{\mathbf{F}}_{inv} - \bar{\mathbf{F}}_{vis}$. Inviscid and viscous fluxes are shown in Equation 6 and Equation 7.

$$\bar{\mathbf{F}}_{inv} = \begin{pmatrix} \rho \mathbf{U} \\ \rho \mathbf{U} \otimes \mathbf{U} + \left(p + \frac{1}{2} \|\mathbf{B}\|^2 \right) I - \mathbf{B} \otimes \mathbf{B} \\ \mathbf{U} \left(e + p + \frac{1}{2} \|\mathbf{B}\|^2 \right) - \mathbf{B}(\mathbf{U} \cdot \mathbf{B}) + C_h \psi \\ \mathbf{U} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{U} + C_h \psi I \\ C_h \mathbf{B} \end{pmatrix} \quad (6)$$

$$\bar{\mathbf{F}}_{vis} = \begin{pmatrix} 0 \\ \bar{\tau} \\ \mathbf{U} \cdot \bar{\tau} - \mathbf{q} + \eta(\mathbf{B} \times \mathbf{J}) \\ \eta(\nabla \times \mathbf{B}) \\ 0 \end{pmatrix} \quad (7)$$

In Equations 6 and 7, I is the identity matrix, η is the magnetic diffusivity, \mathbf{J} is the current density defined as $\mathbf{J} = \nabla \times \mathbf{B}$, $\bar{\tau}$ is the shear stress tensor, and \mathbf{q} is the heat flux. The shear stress tensor and heat flux are defined in Equations 8 and 9 respectively.

$$\bar{\tau} = \mu(\nabla \mathbf{U} + (\nabla \mathbf{U})^T) + \lambda(\nabla \cdot \mathbf{U})I \quad (8)$$

$$\mathbf{q} = -\kappa \rho T \nabla S - \kappa_r \rho C_p \nabla T \quad (9)$$

μ is the dynamic viscosity, $\lambda = -\frac{2}{3}\mu$ based on Stokes' hypothesis, κ is the thermal diffusivity, κ_r is the radiative diffusivity, and S is the specific entropy. The specific entropy is given in Equation 10.

$$S = C_p \ln \left(\frac{p^{\frac{1}{\gamma}}}{\rho} \right) \quad (10)$$

III. Meshing

The spherical shell is generated by layering a series of surface meshes. CHORUS-MHD uses a cubed-sphere meshing technique, which projects the six faces of a cube onto the spherical surface for each radial increment. In total, there are N_r radial increments. Using a cubed-sphere mesh allows for a given face to have angularly equidistant hexahedral elements for all radii. Angularly equidistant elements on each face prevent singularities and allow for better resolution in the polar regions. It is also possible to generate oblate meshes for CHORUS-MHD to run on. For one of the six faces, the angles $\alpha = \arctan(\frac{y}{x})$ and $\beta = \arctan(\frac{z}{x})$ are chosen, with $\alpha = [-\frac{\pi}{4}, \frac{\pi}{4}]$ and $\beta = [-\frac{\pi}{4}, \frac{\pi}{4}]$. On each face there are N_z by N_z zonal elements. An element face at a given radial increment R is defined as the surface enclosed by four arcs. The pairs of arcs enclosing the i^{th} , j^{th} element on a single face are described by the angles

$$[\alpha_i, \alpha_{i+1}] \times [\beta_i, \beta_{i+1}] \quad (11)$$

with

$$\alpha_i = -\frac{\pi}{4} + \frac{i}{N_z} \frac{\pi}{2} \quad \beta_i = -\frac{\pi}{4} + \frac{j}{N_z} \frac{\pi}{2} \quad (12)$$

The total number of elements in the mesh is given by $N_{cells} = 6N_r N_z^2$. The number of degrees of freedom in a three-dimensional mesh is then calculated as $N_{DOF} = N_{cells} \times N_{order}^3$, where N_{order} is the order of the spectral difference method used. Figure 1 shows a cubed sphere mesh with $N_r = 12$ and $N_z = 24$.

IV. Spectral Difference Method

To perform calculations in CHORUS-MHD, all elements are transformed into a standard cubic element in the computational domain. The transformation is described as

$$(x, y, z) = \mathbf{P}(\xi, \eta, \zeta) \quad (13)$$

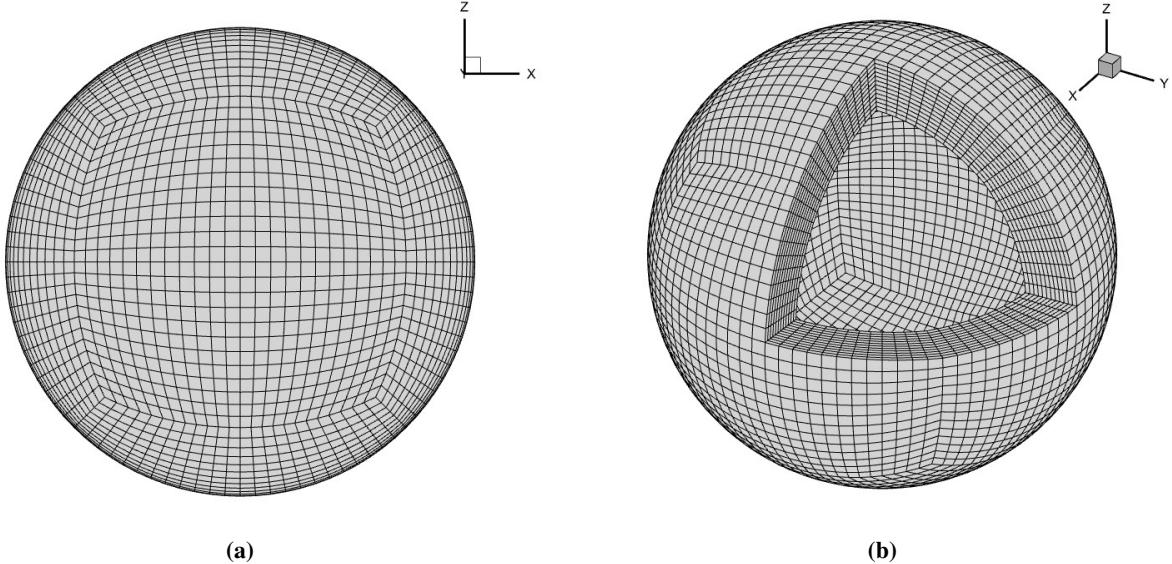


Fig. 1 Cubed Sphere Mesh with 12 Radial Cells and 24 Zonal Cells

Where ξ , η , and ζ are the coordinates in the computational domain defined as

$$(\xi, \eta, \zeta) \in [0, 1] \times [0, 1] \times [0, 1] \quad (14)$$

The transfinite mapping $\mathbf{P}(\xi, \eta, \zeta)$ is created using linear combinations of projectors, including bilinear projectors and a trilinear projector as described in [2]. To perform computations the governing equations are transformed to the computational domain using a Jacobi matrix for the mapping \mathbf{P} . The Jacobi matrix is defined as:

$$J = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{bmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{bmatrix} \quad (15)$$

Using the Jacobi matrix, the governing equations in the form of Equation 1 can be written as

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial t} + \nabla \cdot \tilde{\mathbf{F}} = \tilde{\mathbf{M}} \quad (16)$$

The solution vector and source vector are transformed as $\tilde{\mathbf{Q}} = |J|\mathbf{Q}$ and $\tilde{\mathbf{M}} = |J|\mathbf{M}$. The flux vector is transformed as

$$\tilde{\mathbf{F}} = |J|J^{-1}\mathbf{F} \quad (17)$$

Once in the computational domain, for an N^{th} order spectral difference method, a given cubic element has N^3 solution points, forming a 3-D grid of solution points. For each iteration, the divergence of the flux vector must be determined, which requires a $(N + 1)^3$ grid of flux points in each cubic element to maintain order. A standard 2-D element of flux points and solution points is shown in Figure 2 for reference. Both the solution point and flux point grids are created by repeating 1-D spacings in each coordinate direction. The solution points are chosen to be Chebyshev–Gauss quadrature points. The n^{th} solution point is given as

$$X_{s,n} = \frac{1}{2} \left[1 - \cos \left(\frac{2n - 1}{2N} \pi \right) \right], \quad n = 1, 2, \dots, N \quad (18)$$

The flux points X_f are roots of the Legendre polynomials of N^{th} degree with two additional endpoints on the cell boundaries at zero and one. As seen in Figure 2, each string of solution points, in either unit direction, has an associated string of flux points in the same unit direction used for Lagrange interpolation. For a two-dimensional grid, the

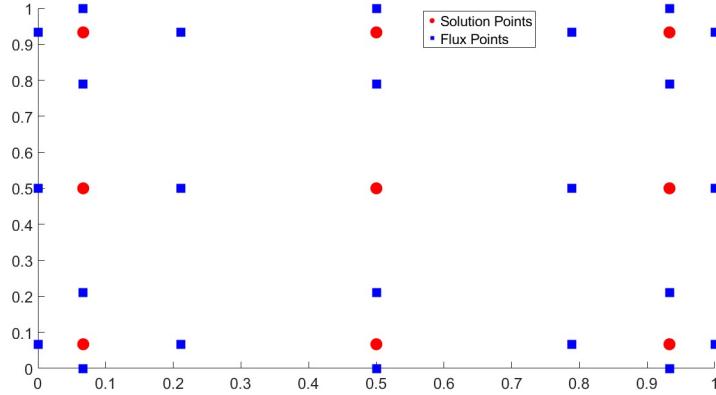


Fig. 2 Standard element solution Points (red) and flux Points (blue) for a two dimensional, third order spectral difference method

flux points are split into two families. In the ξ direction the family of points is given as: $(\xi, \eta) = (X_{f,i}, X_{s,j})$ for $i = [1, N+1]$ and $j = [1, N]$. Similarly in the η direction, the family is described as: $(\xi, \eta) = (X_{s,i}, X_{f,j})$ for $i = [1, N]$ and $j = [1, N+1]$.

To find the time derivative of the solution vector at a given solution point, the divergence of the flux must be evaluated at X_s , as seen in Equation 16. Each iteration starts with the solutions known at the solution points, X_s . The solutions and solution gradient are then extrapolated to their respective families of flux points using one-dimensional Lagrange polynomials. The flux is then calculated using Equation 6 and Equation 7, and a Lagrange polynomial of order $N+1$ is constructed of the flux. The gradient of the flux polynomial is then calculated and evaluated at the solution points. Lagrange polynomials are constructed using Lagrange basis functions at the solution points and the flux points in the (ξ, η, ζ) directions. For the solution points the Lagrange basis function is

$$h_i(X) = \prod_{s=1, s \neq i}^N \left(\frac{X - X_s}{X_i - X_s} \right) \quad (19)$$

Similarly for the flux points,

$$l_i(X) = \prod_{f=1, f \neq i}^{N+1} \left(\frac{X - X_f}{X_i - X_f} \right) \quad (20)$$

To reconstruct solution and flux polynomials in the computational domain, is the tensor product of the one dimensional Lagrange polynomials. For the solutions in the computational domain

$$\tilde{\mathbf{Q}}(\xi, \eta, \zeta) = \sum_i^N \sum_j^N \sum_k^N \frac{\tilde{\mathbf{Q}}_{i,j,k}}{|J_{i,j,k}|} h_i(\xi) h_j(\eta) h_k(\zeta) \quad (21)$$

For the flux in the computational domain the components of the flux vector $\tilde{\mathbf{F}} = \tilde{F}\hat{\xi} + \tilde{G}\hat{\eta} + \tilde{H}\hat{\zeta}$ are given as

$$\tilde{F}(\xi, \eta, \zeta) = \sum_i^N \sum_j^N \sum_k^N \tilde{\mathbf{F}}_{i,j,k} l_i(\xi) h_j(\eta) h_k(\zeta) \quad (22)$$

$$\tilde{G}(\xi, \eta, \zeta) = \sum_i^N \sum_j^N \sum_k^N \tilde{\mathbf{G}}_{i,j,k} h_i(\xi) l_j(\eta) h_k(\zeta) \quad (23)$$

$$\tilde{H}(\xi, \eta, \zeta) = \sum_i^N \sum_j^N \sum_k^N \tilde{\mathbf{H}}_{i,j,k} h_i(\xi) h_j(\eta) l_k(\zeta) \quad (24)$$

To ensure solution polynomials are continuous across element faces, the common solution evaluated at element interfaces is taken as the average of the polynomials from both cells, known as the BR1 scheme [6]. A similar procedure is used for the viscous flux across cell interfaces. For inviscid flux, the Rusanov solver [7] is used to ensure a common flux. The cell face averaging processes are performed during the previously described iteration procedure, after the construction of a solution or flux polynomial.

Once the spatial derivatives of the flux polynomials are computed, the time derivative of the solutions can be found. For time stepping, CHORUS MHD uses a five-stage third-order explicit strong stability-preserving Runge-Kutta method [SSPRK(5,3)]. The coefficients for the method are tabulated in Table 1 of Ruth [8].

V. Initial Conditions

Initially all conserved variables are only a function of radius, and the shell is in hydrostatic balance. Mathematically, $\mathbf{U} = 0$, and

$$\frac{dp}{dr} = -\rho g \quad (25)$$

The classical solutions for static stratification give initial profiles for thermodynamic variables ρ, p, T are

$$p = p_b \left[1 - \frac{\Phi - \Phi_b}{C_p T_b} \right]^{\frac{\gamma}{\gamma-1}} \quad \rho = \rho_b \left[1 - \frac{\Phi - \Phi_b}{C_p T_b} \right]^{\frac{1}{\gamma-1}} \quad T = T_b \left[1 - \frac{\Phi - \Phi_b}{C_p T_b} \right] \quad (26)$$

Where Φ is the gravitational potential given by $\Phi = \frac{GM_\odot}{r} \hat{r}$ and T_b is the temperature at the bottom surface given by

$$T_b = \frac{\Phi_t - \Phi_b}{C_p (1 - e^{-(\gamma-1)N_\rho})} \quad (27)$$

Where N_ρ is the number of density scale heights defined as $N_\rho = \ln \left(\frac{\rho_t}{\rho_b} \right)$. The relations listed in Equation 26 do not satisfy thermal equilibrium. To satisfy an approximate thermal equilibrium, an almost flux-balanced approach is used [1]. In the initial state, the total heat flux from the shell is constant through each radial layer. At the bottom boundary, the heat flux can be defined by

$$\mathbf{q} = -\kappa \rho T \nabla S - \kappa_r \rho C_p \nabla T = \frac{L_\odot}{4\pi r_b^2} \hat{r} \quad (28)$$

Where L_\odot is the stellar luminosity of the spherical shell, defined as an input. The gradients ∇S and ∇T simplify to $\frac{dS}{dr}$ and $\frac{dT}{dr}$ respectively. Rearranging Equation 28 for the entropy gradient gives

$$\frac{dS}{dr} = \frac{1}{\kappa \rho T} \left(\frac{L_\odot}{4\pi r_b^2} + \kappa_r \rho C_p \frac{dT}{dr} \right) \quad (29)$$

Using Equations in 26 and Equation 29, the target entropy gradient can be solved for using the form

$$S = S_b + \int_{r_b}^r \frac{dS}{dr} dr \quad (30)$$

Once the entropy profile is computed, we can then solve for a new density gradient. Using Equations 10 and 25, it can be shown that

$$\frac{d\rho}{dr} = - \left(\frac{\rho}{C_p} \frac{dS}{dr} + \frac{g}{\gamma} \rho^{2-\gamma} e^{\frac{\gamma S}{C_p}} \right) \quad (31)$$

With the entropy S and its gradient $\frac{dS}{dr}$ already computed, Equation 31 can be solved numerically for the new density distribution. Once the new density distribution is calculated, the pressure and temperature distributions can also be calculated. The system starts thermally stable, without convection. The constant temperature flux from the bottom boundary of the shell increases the temperature gradient until it becomes critical, resulting in thermal instability and convection.

VI. Problem Formulation and Boundary Conditions

CHORUS considers a hollow fluid shell with most of the mass inside the shell. A constant heat flux is imposed on the inner surface to induce convection, and the outer surface has a constant temperature. The inner and outer surfaces are assumed to be stress-free and impenetrable. For the magnetic boundary conditions, the surfaces can be defined as perfectly electrically conducting, or a perfectly radial magnetic field can be defined. The constant heat flux is at the bottom surface is given by

$$\mathbf{q} = \frac{L_{\odot}}{4\pi r_b^2} \hat{\mathbf{r}} \quad (32)$$

L_{\odot} is the stellar luminosity of the spherical shell, specified as an input, and r_b is the radius of the inner surface. Impenetrable boundaries are described by Equation 33.

$$\mathbf{U} \cdot \hat{\mathbf{f}} = 0 \quad (33)$$

For the stress-free surfaces, the shear stress tensor $\bar{\tau}$ is converted to spherical coordinates, and the angular components are set to zero for cells at $r = r_b$ and $r = r_t$. The perfectly conducting boundary conditions are described as $\mathbf{J} = J_r \hat{\mathbf{r}}$ at $r = r_s$ where r_s is the surface to which the boundary conditions are applied. Using $\mathbf{J} = \nabla \times \mathbf{B}$ leads to the magnetic field conditions that $B_r = \frac{\partial}{\partial r} (r B_{\theta}) = \frac{\partial}{\partial r} (r B_{\phi}) = 0$ at $r = r_s$. In CHORUS-MHD, this is implemented by transforming the Jacobian of the magnetic field into spherical coordinates and setting the appropriate derivatives to zero. For perfectly radial conditions $B_{\theta} = B_{\phi} = 0$ at $r = r_s$, and to maintain the divergence-free magnetic field, $\frac{\partial}{\partial r} (r^2 B_r) = 0$ at $r = r_s$. For perfectly insulating boundaries, $\mathbf{J} = 0$ at $r = r_s$ and outside of the spherical shell. To model perfectly insulating boundary conditions, now that $\nabla \times \mathbf{B} = 0$, the magnetic field can now be represented as a scalar potential, and the magnetic field inside of the shell can be matched to the magnetic field outside of the shell. For codes using spherical harmonics, the perfectly insulated boundary conditions are rather easy to apply by mapping the magnetic field to the scalar magnetic field potential. For fully compressible codes using a method other than spherical harmonics, perfectly insulating boundary conditions would be much more cumbersome. Mapping the magnetic field to the potential magnetic field requires solving the Laplace equation numerically for the scalar magnetic field at the given boundary of the shell. Currently, perfectly insulating boundary conditions are not implemented in CHORUS-MHD. In this study, benchmark tests were run with perfectly radial boundary conditions, which allow surface current on the shell.

VII. Benchmark Problem Definition

Two benchmark problems are presented for CHORUS-MHD. The first benchmark test is an extension of a hydrodynamic model of the Sun presented by Wang et al. [1] to an MHD model. The key parameters defining the solar benchmark test are shown in Table 1. The second benchmark test increases the density scale height from $N_{\rho} = 3$ to $N_{\rho} = 4$. Both simulations use a sixth-order spectral difference method.

Table 1 Solar Benchmark Parameters

Physical Input Parameters:		
$r_t = 6.61 \times 10^{10} \text{ cm}$	$r_b = 4.87 \times 10^{10} \text{ cm}$	$M_{\odot} = 1.98891 \times 10^{33} \text{ g}$
$\Omega = 8.1 \times 10^{-5} \text{ s}^{-1}$	$\rho = 0.21 \text{ g cm}^{-3}$	$G = 6.67 \times 10^{-8} \text{ g}^{-1} \text{ cm}^3 \text{ s}^{-2}$
Fluid Properties:		
$R = 1.4 \times 10^8 \text{ erg g}^{-1} \text{ K}^{-1}$	$\kappa = 6.0 \times 10^{13} \text{ cm}^2 \text{ s}^{-1}$	$\nu = 6.0 \times 10^{13} \text{ cm}^2 \text{ s}^{-1}$
$\gamma = \frac{5}{3}$	$\eta = 1.2 \times 10^{13} \text{ cm}^2 \text{ s}^{-1}$	
Thermodynamic Properties:		
$L_{\odot} = 3.846 \times 10^{36} \text{ erg s}^{-1}$	$C_p = 3.5 \times 10^8 \text{ erg g}^{-1} \text{ K}^{-1}$	$N_{\rho} = 3$

The radiative flux of the solar benchmark is given by $\kappa_r = \lambda (c_0 + c_1 \omega + c_2 \omega^2)$ with $c_0 = 1.5600975 \times 10^8$, $c_1 = -4.5631718 \times 10^7$, $c_2 = 3.3370368 \times 10^6$, and $\omega = r \times 10^{-10}$. For the solar benchmark, the lower surface is assumed to be perfectly conducting as the magnetic flux is typically very small. For the upper surface, the magnetic field is assumed to be perfectly radial, following observations of the Sun's magnetic field near the surface.

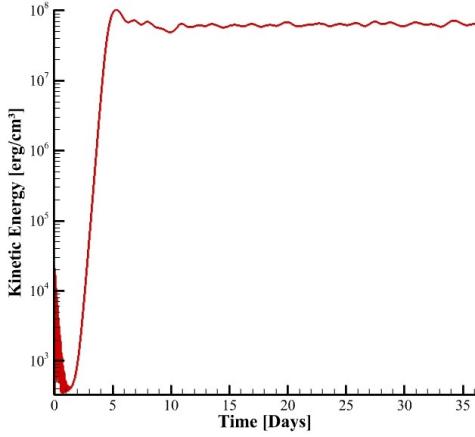


Fig. 3 Globally Average Kinetic Energy for First Benchmark Test

VIII. Results

For both simulations, the globally averaged kinetic energy, globally averaged magnetic field, meridional circulation, differential rotation, poloidal magnetic energy, and toroidal magnetic energy are calculated. The averaged kinetic energy and magnetic field energy are given as:

$$KE = \frac{1}{V} \int_V \frac{1}{2} \rho \mathbf{U} \cdot \mathbf{U} dV \quad (34)$$

$$E_B = \frac{1}{V} \int_V \frac{1}{2} \mathbf{B} \cdot \mathbf{B} dV \quad (35)$$

The differential rotation presented is simply the longitudinal-time average of V_ϕ , and the meridional flow is the combination of the radial velocity V_r and the polar angle velocity V_θ as $V_{mer} = \sqrt{V_r^2 + V_\theta^2}$, longitudinally time averaged. The poloidal and toroidal contributions to the magnetic field energy are given as

$$E_{Bpol} = \frac{B_r^2 + B_\theta^2}{8\pi} \quad E_{Btor} = \frac{B_\phi^2}{8\pi} \quad (36)$$

A. First Solar Benchmark

The first benchmark was run using Clarkson University's GPU partition on ACRES. The simulation was run using 3 GPUs, with a mesh using 28 radial and 26 zonal cells, for a total of 36 simulated days. CHORUS-MHD produces a lower steady-state kinetic energy density of approximately $6.47 \times 10^7 \text{ erg cm}^{-3}$ in comparison to the hydrodynamic benchmark, as seen in Figure 3. A lower steady state energy is to be expected in a full MHD simulation due to the addition of a magnetic field. Magnetic forces restrict fluid flow perpendicular to the magnetic field lines, and energy is constantly being converted between the magnetic field and the velocity field. Figure 4 shows longitudinally time-averaged plots of the velocity field and magnetic field. Figure 5 shows the mollweide plots of the velocity field and magnetic field. As seen in the longitudinal-time average contours, the MHD simulation presents strong differential rotation and meridional circulation, similar to the hydrodynamic benchmark. The meridional circulation predicts a flow pattern similar to that of Wang et al. [1], with a near-surface circulation, followed by two deeper circulations in the opposite direction. This meridional flow pattern is reflected in both the upper and lower hemispheres. In the Mollweide projections, MHD CHORUS predicts banana cells similar to that of the hydrodynamic benchmark, with many upflow lanes separated by stronger downflow lanes, however with a reduced flow magnitude. The lower kinetic energy and velocity profiles are evident that the magnetic field is key to reducing typical over predictions of fluid velocity. The magnetic field is mostly concentrated near the equator for all three magnetic field components. The concentration of the magnetic field near the equator is likely due to the increased flow speed near the equator, seen in both the differential rotation and the meridional flow. In the shown Mollweide projection, note that all magnetic-field components are concentrated at a similar location in the left hemisphere. In the velocity fields, most notably the radial and azimuthal

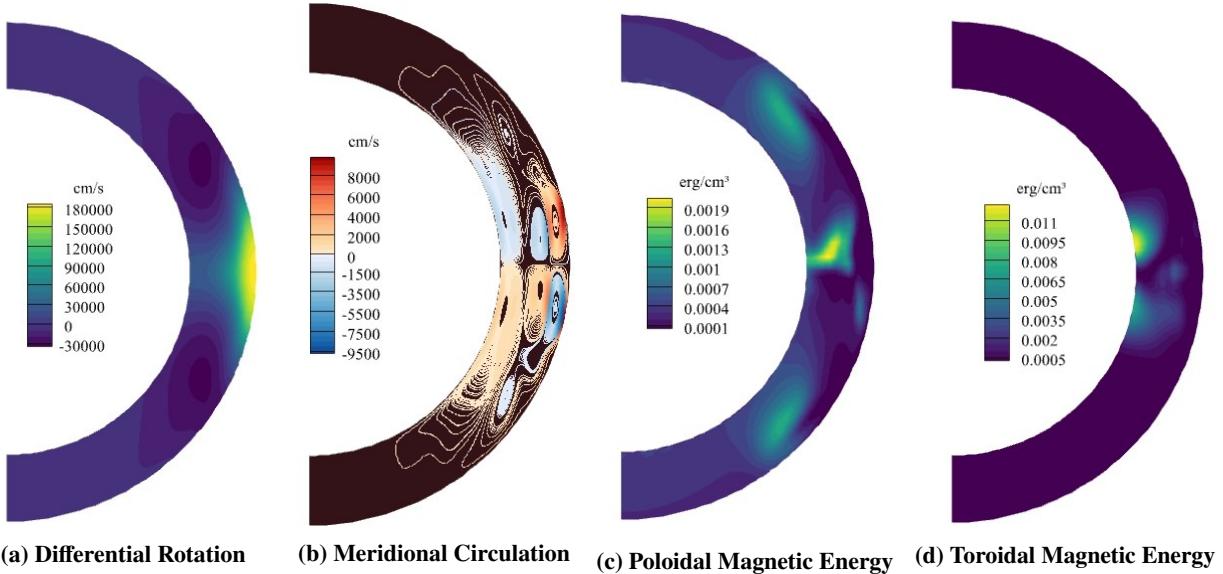


Fig. 4 Longitudinal Time-Averaged Velocity Fields and Magnetic Fields for Benchmark 1

fields, there is a damped zone in the same region where the magnetic field is peaking. This is a similar behavior seen in solar activity like sunspots, where the flow is restricted due to magnetic field lines, and the area's luminosity is lowered due to the reduced supply of plasma.

B. Second Solar Benchmark with Increased Density Scale Height

The second benchmark was run using NASA's Pleiades GPU partition. The simulation was run using 16 GPUS, with a mesh using 32 radial and 34 zonal cells, for a total of simulated 24 days. Figure 6 shows the globally averaged kinetic energy (a) and the globally averaged magnetic energy (b) vs time. Figure 7 shows longitudinally time-averaged plots of the velocity field and magnetic field for the second solar benchmark. Magnetic field plots are plotted on an exponential basis because of the large contrast in magnitudes between the equatorial and polar regions. The kinetic energy density of the second solar benchmark saturates at approximately $9.57 \times 10^7 \text{ erg cm}^{-3}$. The higher kinetic energy of the second benchmark is probably due to the increase of the density scale height without reducing the luminosity or rotational speed of the simulation. From the longitudinally time-averaged plots of the second benchmark, it can be seen that a pure increase in density scale height has large effects on both the velocity and magnetic fields. First, there is a large increase in the magnitude of both differential flow and meridional circulation, with the meridional circulation having a much larger change. The meridional flow pattern in the second solar benchmark is also significantly different from the first benchmark. In the second case, the near-equator meridional flow switches directions on the innermost circulation, and alternating convective cells can be seen to extend all the way up to the poles with strong magnitudes. From the poloidal and toroidal magnetic energy contours, the polar region is much more active, probably corresponding to the strong convective cells near the poles.

IX. Concluding Remarks

The CHORUS++ code presented in [2] is now being further developed successfully to solve 3D MHD equations. CHORUS-MHD is also GPU accelerated in this study before it is applied to study two solar dynamo benchmark problems with different density scale heights. One benchmark problem involves an evident exchange of poloidal and toroidal magnetic energies near the equatorial region. The other benchmark problem captures two magnetically active poles that closely correlate with active convective cells in the polar regions. With both solar benchmark tests completed, the next step in our investigation is to perform further parametric studies of the case with increased density scale height. The next simulation will reduce the model luminosity and rotation rate to better control velocity and magnetic fields, with an expectation to achieve a solar-like meridional-flow profile and cyclic magnetic fields at a higher density scale height.

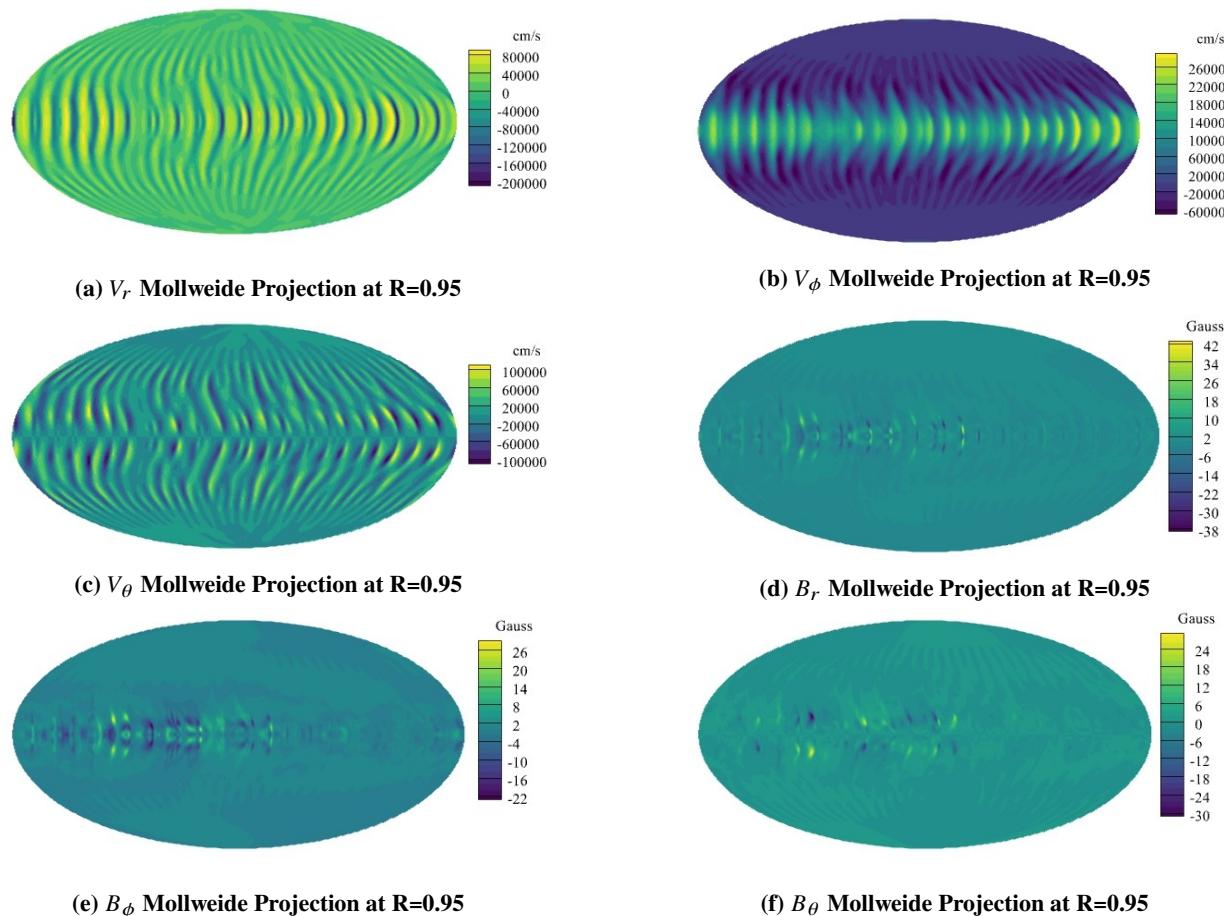


Fig. 5 Mollweide Projections of Velocity Fields and Magnetic Fields for Benchmark 1

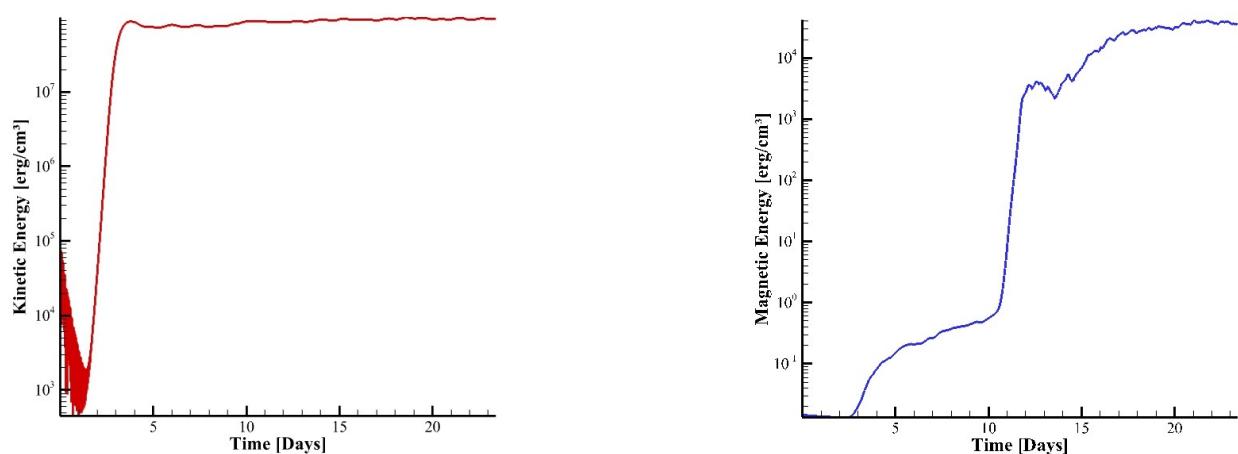


Fig. 6 Globally Average Kinetic Energy (a) and Globally Averaged Magnetic Energy (b)

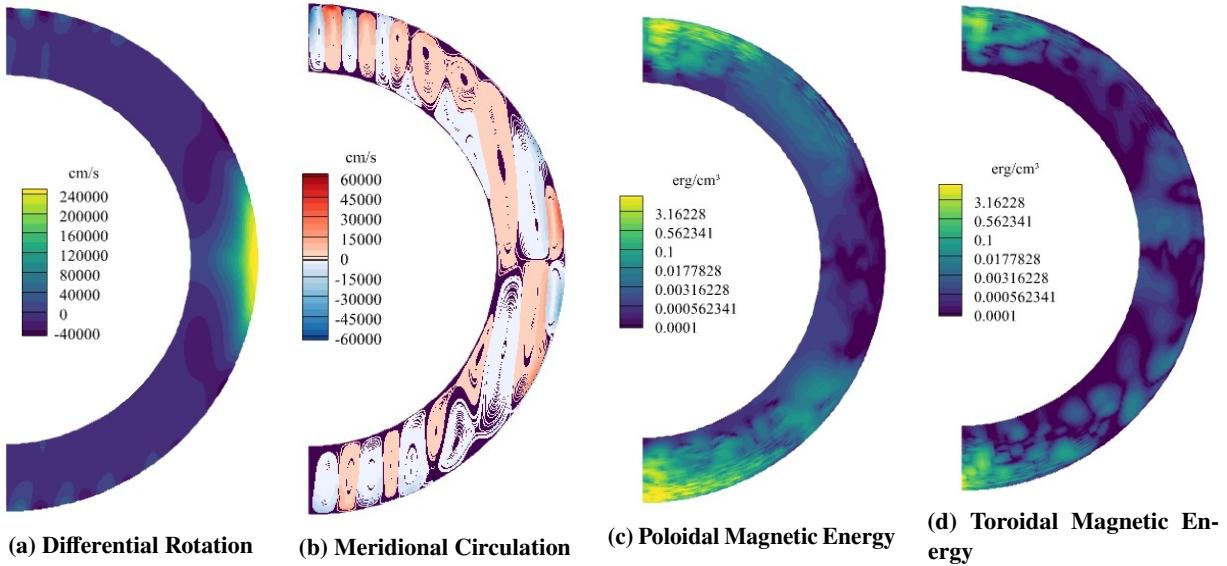


Fig. 7 Longitudinally Time-Averaged Velocity Fields and Magnetic Fields for Benchmark 2

Acknowledgments

The research work in this paper has been financially supported by a National Science Foundation (NSF) award (No. 2310372) monitored by Dr. Lisa Winter and an Air Force Office of Scientific Research (AFOSR) grant (award No. FA9550-23-1-0596) monitored by Dr. Fariba Fahroo. The computational resources for this work were partially supported by the NASA grant 80NSSC20K0602. Our computations were performed on the GPU nodes of Clarkson's ACRES cluster and NASA's Pleiades Supercomputer. CHORUS-MHD was GPU accelerated recently thanks to the M.S. thesis work by Russell Hankey

References

- [1] Wang, J., Liang, C., and Miesch, M. S., "A Compressible High-Order Unstructured Spectral Difference Code for Stratified Convection in Rotating Spherical Shells," *Journal of Computational Physics*, Vol. 290, 2015, pp. 90–111.
- [2] Chen, K., Liang, C., and Wan, M., "Arbitrarily high-order accurate simulations of compressible rotationally constrained convection using a transfinite mapping on cubed-sphere grids," *Physics of Fluids*, Vol. 35, 2023, p. 086120.
- [3] Dikpati, M., Raphaldini, B., McIntosh, S. W., and Gilman, P. A., "A magnetohydrodynamic mechanism for the formation of solar polar vortices," *Proceedings of the National Academy of Sciences*, Vol. 121, 2024, p. e2415157121.
- [4] Chen, K., "High-Order Accurate Simulations of Compressible MHD Dynamo on Unstructured Grids," Ph.D. thesis, Clarkson University, 2023.
- [5] D. Derigs, Winters, A. R., Gassner, G. J., Walch, S., and Bohm, M., "Ideal GLM MHD: About the entropy consistent nine-wave magnetic field divergence diminishing ideal magnetohydrodynamics equations," *Journal of Computational Physics*, Vol. 364, 2018, pp. 420–467.
- [6] Bassi, F., and Rebay, S., "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations," *Journal of Computational Physics*, Vol. 131, 1997, pp. 267–279.
- [7] Rusanov, V. V., "The calculation of the interaction of non-stationary shock waves and obstacles," *USSR Computational Mathematics and Mathematical Physics*, Vol. 1, 1962, pp. 304–320.
- [8] Ruuth, S. J., "Global Optimization of Explicit Strong-Stability-Preserving Runge-Kutta Methods," *Mathematics of Computation*, Vol. 75, No. 253, 2006, pp. 183–207.