

RESEARCH ARTICLE | AUGUST 16 2023

Arbitrarily high-order accurate simulations of compressible rotationally constrained convection using a transfinite mapping on cubed-sphere grids FREE

Special Collection: [Turbulence in Plasmas and Fluids](#)Kuangxu Chen (陈匡旭) ; Chunlei Liang ; Minping Wan (万敏平)  *Physics of Fluids* 35, 086120 (2023)<https://doi.org/10.1063/5.0158146>[CrossMark](#)

Articles You May Be Interested In

Transfinite elements: A highly efficient procedure for modeling open field problems

Journal of Applied Physics (April 1987)

Deterministic Quantum Mechanics Versus Classical Mechanical Indeterminism and Nonlinear Dynamics

AIP Conference Proceedings (April 2007)

Average Exceptional Lie Group Hierarchy and High Energy Physics

AIP Conference Proceedings (May 2008)

Arbitrarily high-order accurate simulations of compressible rotationally constrained convection using a transfinite mapping on cubed-sphere grids

Cite as: Phys. Fluids **35**, 086120 (2023); doi: [10.1063/5.0158146](https://doi.org/10.1063/5.0158146)

Submitted: 15 May 2023 · Accepted: 14 July 2023 ·

Published Online: 16 August 2023



Kuangxu Chen (陈匡旭),¹ Chunlei Liang,^{1,a)} and Minping Wan (万敏平)^{2,3,b)}

AFFILIATIONS

¹Department of Mechanical and Aerospace Engineering, Clarkson University, Potsdam, New York 13699, USA

²Guangdong Provincial Key Laboratory of Turbulence Research and Applications, Department of Mechanics and Aerospace Engineering, Southern University of Science and Technology, Shenzhen 518055, China

³Guangdong-Hong Kong-Macao Joint Laboratory for Data-Driven Fluid Mechanics and Engineering Applications, Southern University of Science and Technology, Shenzhen 518055, China

Note: This paper is part of the special topic, Turbulence in Plasmas and Fluids.

^{a)}Electronic mail: cliang@clarkson.edu

^{b)}Author to whom correspondence should be addressed: wanmp@sustech.edu.cn

ABSTRACT

We present two major improvements over the Compressible High-Order Unstructured Spectral difference (CHORUS) code published in Wang *et al.*, “A compressible high-order unstructured spectral difference code for stratified convection in rotating spherical shells,” *J. Comput. Phys.* **290**, 90–111 (2015). The new code is named CHORUS++ in this paper. Subsequently, we perform a series of efficient simulations for rotationally constrained convection (RCC) in spherical shells. The first improvement lies in the integration of the high-order spectral difference method with a boundary-conforming transfinite mapping on cubed-sphere grids, thus ensuring exact geometric representations of spherical surfaces on arbitrary sparse grids. The second improvement is on the adoption of higher-order elements (sixth-order) in CHORUS++ vs third-order elements for the original CHORUS code. CHORUS++ enables high-fidelity RCC simulations using sixth-order elements on very coarse grids. To test the accuracy and efficiency of using elements of different orders, CHORUS++ is applied to a laminar solar benchmark, which is characterized by columnar banana-shaped convective cells. By fixing the total number of solution degrees of freedom, the computational cost per time step remains unchanged. Nevertheless, using higher-order elements in CHORUS++ resolves components of the radial energy flux much better than using third-order elements. To obtain converged predictions, using sixth-order elements is 8.7 times faster than using third-order elements. This significant speedup allows global-scale fully compressible RCC simulations to reach equilibration of the energy fluxes on a small cluster of just 40 cores. In contrast, CHORUS simulations were performed by Wang *et al.* on supercomputers using approximately 10 000 cores. Using sixth-order elements in CHORUS++, we further carry out global-scale solar convection simulations with decreased rotational velocities. Interconnected networks of downflow lanes emerge and surround broader and weaker regions of upflow fields. A strong inward kinetic energy flux compensated by an enhanced outward enthalpy flux appears. These observations are all consistent with those published in the literature. Furthermore, CHORUS++ can be extended to magneto-hydrodynamic simulations with potential applications to the hydromagnetic dynamo processes in the interiors of stars and planets.

Published under an exclusive license by AIP Publishing. <https://doi.org/10.1063/5.0158146>

I. INTRODUCTION

Turbulent thermal convection is ubiquitous in the interiors of rotating stars and planets, such as the Sun and Jupiter. With advances in computing powers, numerical simulations of rotationally constrained

convection (RCC) are considered for studies of many astrophysical hydrodynamic phenomena, such as differential rotation, solar granulation, and meridional circulation. However, global-scale simulations of the convection zone with realistic conditions are challenging. Given that

the fluid Reynolds number (Re) and Rayleigh number (Ra) in the Sun are on the order of 10^{12} and 10^{20} ,^{1,2} it is hard to resolve the broad range of length scales in turbulent solar convection, but substantial progress has been made in simulations with reduced flow parameters. Early laminar convection simulations^{3–5} reproduced many salient features of solar convection, e.g., the large differential rotation, which takes the form of equatorial acceleration. More recent studies,^{6–9} which were based on turbulent RCC simulations at higher Reynolds numbers, reproduced structures of convective cells on the Sun and revealed how differential rotation and meridional circulation are established. Further research found that convective patterns in laminar and turbulent states are significantly different.⁶ The convective pattern in the laminar case is dominated by banana cells, which is consistent with observations in the early simulations. However, the turbulent case exhibits broad and weak upflow regions surrounded by narrow and strong downflow lanes. This pattern resembles solar granulation but on a much larger scale. A subsequent study based on turbulent RCC simulations with $Re = 400$ revealed that coherent downflow structures associated with giant cells play a significant role in maintaining differential rotation.⁹ Purely hydrodynamic simulations of global-scale RCC were also extended to magnetohydrodynamic (MHD) simulations, and the interplay between convection, rotation, and magnetic fields was investigated.⁸

Although many efficient codes for RCC simulations have been developed, they may still have some limitations. The first limitation is that most of them are based on pseudo-compressible approximations. For example, the anelastic spherical harmonic (ASH) code,¹⁰ one of the most successful codes for RCC simulations in spherical shells, uses the anelastic approximation.¹¹ This approximation is valid when characteristic velocities remain substantially subsonic, and the stratification does not become appreciably superadiabatic. It does not hold near the photosphere of the Sun, where the Mach number approaches unity. The second limitation is that many of them are based on a specific gridding system and cannot be implemented on arbitrary unstructured grids. For example, the ASH code is based on the spherical harmonic basis functions, which only apply to the spherical geometry. The PENCIL code,^{12,13} a popular code for compressible astrophysical fluid dynamics, is designed for structured grids. Although it was successfully applied to a spherical star immersed inside of a Cartesian tube,¹⁴ its early applications^{15,16} were concentrated on the spherical wedge geometry which omits polar regions. The third limitation is that parallel slowdown may appear as a result of a communications bottleneck. The reason is that most codes use anelastic or incompressible approximations and, therefore, need computationally expensive global communication to enforce the divergence-free constraint of the velocity field.

To avoid aforementioned three limitations, a Compressible High-Order Unstructured Spectral difference (CHORUS) code was developed,¹⁷ which uses fully compressible models, unstructured grids, and compact computational stencils favored by massively parallel computing. To provide high fidelity, the high-order accurate spectral difference (SD) method^{18–22} is used. This is a compact high-order method, which uses element-wise polynomials for reconstruction. Similar methods, like the spectral element method (SEM),²³ were applied to direct numerical simulations of rotating convection in a cube.²⁴ By comparing with the SEM, the discontinuous Galerkin (DG) method, which solve the integral form of the Navier–Stokes (NS) equations, the SD method solves the differential form of the NS

equations. It is especially attractive since explicit integral computations are not needed, and all interpolation and differentiation are performed in efficient one-dimensional (1D) procedures.

By comparing with the original CHORUS code, which uses the isoparametric mapping to perform coordinate transformations, the new code, CHORUS++, employs exact transfinite mapping and implements the SD method on a cubed-sphere grid. With transfinite mapping, geometric errors for capturing the curvature of spherical surfaces are exactly zero on arbitrary sparse grids. Dense distribution of computational elements near curved boundaries is no longer needed. We can use element-wise high-order polynomials to reconstruct solutions and fluxes on sparse grids to achieve high accuracy and save computational costs. The superiority of higher-order reconstruction compared with lower-order counterparts is shown for RCC simulations in terms of accuracy and computing costs in Sec. VI. Moreover, the topology of the cubed-sphere grid is optimal since the grid resolution on each horizontal surface is the same in polar and equatorial regions. By contrast, grids used in ASH and PENCIL distribute grid points uniformly in latitudinal, longitudinal, and radial dimensions. Thus, dense grid points are concentrated in polar regions, and sparse grid points are scattered in equatorial regions. However, this distribution of grid points is not optimal since the polar regions, compared to the equatorial regions, are less susceptible to the Coriolis force and may exhibit fewer intricate small-scale flow structures.

CHORUS++ suffers from severe Courant–Friedrichs–Lewy (CFL) constraint when the compressible flows are in the regimes of very low Mach numbers since it is based on fully compressible models and explicit time stepping. To circumvent this issue, the stellar luminosity is artificially increased to achieve higher Mach numbers,^{17,25,26} and the rotational velocity is increased correspondingly to keep a solar-like Rossby number.

This paper is organized as follows. In Sec. II, RCC equations and simulation setups are introduced. Procedures for cubed-sphere boundary-conforming grid generation are given in Sec. III. The implementation of the SD method based on the transfinite mapping is described in Sec. IV. The parallelization of CHORUS++ is discussed in Sec. V. Numerical results on the laminar solar benchmark and its variants with different rotational velocities are presented in Secs. VI and VII.

II. SIMULATION SETUP

To model global-scale RCC, governing equations for the hydrodynamics of an ideal gas are solved in a rotating spherical shell. The radius is equal to $r = r_b$ at the inner spherical surface and $r = r_t$ at the outer spherical surface. These boundaries are impenetrable. A constant heat flux is continuously fed into the spherical shell from the bottom boundary to induce thermal instability and ensuing convection.

A. Governing equations

The simulations are conducted in a reference frame whose origin coincides with the geometric center of the spherical-shell-shaped convection zone. The reference frame is uniformly rotating about the z axis with rotational velocity Ω_0 . The governing equations for the RCC in a rotating reference frame are expressed in the form of hyperbolic conservation law as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \bar{\mathbf{F}} = \mathbf{M}, \quad \nabla \cdot \bar{\mathbf{F}} = \frac{\partial \mathbf{F}}{\partial x} + \frac{\partial \mathbf{G}}{\partial y} + \frac{\partial \mathbf{H}}{\partial z}, \quad (1)$$

where $\mathbf{Q} = (\rho, \rho\mathbf{U}, E)^T$ is the vector of conserved variables, ρ is the density, $\mathbf{U} = (u, v, w)$ is the velocity vector, and E is the total energy. Since an ideal gas is assumed, we have $p = \rho\mathcal{R}T$, where $\mathcal{R}$ is the gas constant. $\bar{\mathbf{F}} = (\mathbf{F}, \mathbf{G}, \mathbf{H})$ is the total flux vector; $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ are three components of the flux in the x , y , and z directions; and $\mathbf{M}$ is the source term. The total flux $\bar{\mathbf{F}} = \bar{\mathbf{F}}_{\text{inv}} - \bar{\mathbf{F}}_{\text{vis}}$ consists of the inviscid flux minus the viscous flux. The inviscid and viscous flux vectors are

$$\bar{\mathbf{F}}_{\text{inv}}(\mathbf{Q}) = \begin{pmatrix} \rho\mathbf{U} \\ \rho\mathbf{U} \otimes \mathbf{U} + p\mathbb{I} \\ \mathbf{U}(E + p) \end{pmatrix}, \quad \bar{\mathbf{F}}_{\text{vis}}(\mathbf{Q}, \nabla\mathbf{Q}) = \begin{pmatrix} 0 \\ \bar{\boldsymbol{\tau}} \\ \mathbf{U} \cdot \bar{\boldsymbol{\tau}} - \mathbf{q} \end{pmatrix},$$

where ρ , p , and E are the density, hydrodynamic pressure, and total energy, respectively. The total energy E is defined as

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

where γ is the specific ratio of ideal gas and $\|\cdot\|$ is the Euclidean vector norm. $\bar{\boldsymbol{\tau}}$ is the shear stress tensor,

$$\bar{\boldsymbol{\tau}} = \mu(\nabla\mathbf{U} + (\nabla\mathbf{U})^T) + \lambda(\nabla \cdot \mathbf{U})\mathbb{I}, \quad (3)$$

where $\mu = \rho\nu$ is the dynamic viscosity, ν is the kinematic viscosity, and $\lambda = -2/3\mu$ based on the Stokes' hypothesis. The heat flux vector $\mathbf{q} = -\kappa\rho T\nabla S - \kappa_r\rho C_p\nabla T$, where T is the temperature, κ is the entropy diffusion coefficient, κ_r is the radiative diffusivity, and S is the specific entropy, which is defined as

$$S = C_p \ln(p^{1/\gamma}/\rho), \quad (4)$$

where C_p is the specific heat at constant pressure. Here, the entropy diffusion term parameterizes the energy flux due to unresolved, subgrid-scale convective motions, which tend to mix entropy.^{6,17,27,28} The source term $\mathbf{M}$ is the combination of Coriolis force term and the gravitational force term, which is defined as

$$\mathbf{M} = \begin{pmatrix} 0 \\ \rho\mathbf{g} - 2\rho\boldsymbol{\Omega}_0 \times \mathbf{U} \\ \rho\mathbf{U} \cdot \mathbf{g} \end{pmatrix}, \quad (5)$$

where $\boldsymbol{\Omega}_0 = (0, 0, \Omega_0)$ is the angular velocity vector. In the rotating reference frame, the Coriolis force is considered. The gravity is modeled as $\mathbf{g} = -g\hat{\mathbf{r}} = -(GM_\odot/r^2)\hat{\mathbf{r}}$, G is the gravitational constant, $M_\odot$ is the total mass of gases in the solar convection zone, and $\hat{\mathbf{r}}$ is the radial unit vector. $\rho\mathbf{U} \cdot \mathbf{g}$ is the work done by buoyancy.

B. Boundary conditions

The top and bottom boundaries are impenetrable, which means that radial velocities are zero at all boundaries,

$$U_r = \mathbf{U} \cdot \hat{\mathbf{r}} = 0. \quad (6)$$

Meanwhile, all boundaries are stress-free. Stress tensors $\bar{\boldsymbol{\tau}}$ on boundary faces are computed in the Cartesian coordinate systems by computing the velocity gradients. They are then transformed to spherical coordinate systems, and their tangential components are dropped. After this operation, they are transformed back to the Cartesian coordinate systems.

A fixed heat flux is imposed at the bottom boundary,

$$\mathbf{q} = -\kappa\rho T\nabla S - \kappa_r\rho C_p\nabla T = (L_\odot/A_b)\mathbf{n} \quad \text{at } r = r_b, \quad (7)$$

where $L_\odot$ is the stellar luminosity, $A_b = 4\pi r_b^2$ is the area of the horizontal surface at the bottom boundary, and $\mathbf{n}$ is the outward-pointing unit normal vector. The temperature at the top boundary is fixed.

C. Initial conditions

In this part, how to initialize the flow field to trigger thermal instability and initiate convection is introduced.

Note that initially, all flow variables only vary in the radial dimension. Initially, the fluid is in a hydrostatic balance state ($\mathbf{U} = \mathbf{0}$),

$$dp/dr = -\rho g. \quad (8)$$

The classical polytropic solution for static stratification²⁷ can give the initial profiles for pressure p , density ρ , and temperature T as

$$\begin{aligned} p &= p_b \left[1 - \frac{\Phi - \Phi_b}{C_p T_b} \right]^{\gamma/(\gamma-1)}, \\ \rho &= \rho_b \left[1 - \frac{\Phi - \Phi_b}{C_p T_b} \right]^{1/(\gamma-1)}, \\ T &= T_b \left[1 - \frac{\Phi - \Phi_b}{C_p T_b} \right], \end{aligned} \quad (9)$$

where $\Phi = -GM_\odot/r$ is the Roche potential, the temperature at the bottom boundary is given by

$$T_b = \frac{\Phi_t - \Phi_b}{C_p [1 - \exp(-(\gamma-1)N_\rho)]}, \quad (10)$$

the density at the bottom boundary ρ_b is an input to the codes, which refer to the gas density near the base of the solar convection zone and is set as $\rho_b = 0.21 \text{ g cm}^{-3}$; the subscripts b and t denote the bottom and top boundaries, respectively; and $N_\rho = \ln(\rho_b/\rho_t)$ is the number of density scale heights. However, the profiles in Eq. (9) do not satisfy the thermal equilibrium. To achieve an approximate thermal equilibrium, an almost flux balance approach¹⁷ is utilized. If the simulation starts from a thermally relaxed state, the total flux through horizontal surfaces at any radial position should be constant,⁸

$$\mathbf{q} \cdot \hat{\mathbf{r}} = L_\odot/(4\pi r^2) \quad \text{from } r_b \text{ to } r_t. \quad (11)$$

Using the polytropic solutions of p , ρ , and T in Eq. (9), a target entropy gradient is computed,

$$\frac{dS}{dr} \equiv \Gamma = \frac{1}{\rho T \kappa} \left(\frac{L_\odot}{4\pi r^2} + \rho C_p \kappa_r \frac{dT}{dr} \right). \quad (12)$$

Then, a target entropy profile is computed,

$$S(r) = S_b + \int_{r_b}^r \Gamma dr, \quad (13)$$

where S_b is the specific entropy at the bottom boundary. After this step, the polytropic solutions are not needed anymore in the following computations. Taking spatial derivative of the specific entropy in Eq. (4) along the radial dimension and using Eq. (8), we have

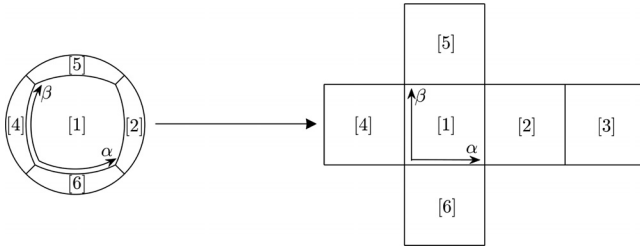


FIG. 1. The cubed-sphere gridding technique maps the spherical surface on the six faces of a cube (labeled as region [1], [2], ..., [6]), which are shown here opened on a plane.

$$\frac{\Gamma}{C_p} = -\left(\frac{1}{\rho} \frac{d\rho}{dr} + \frac{g\rho}{\gamma p}\right). \quad (14)$$

Rearranging the above equation gives an ordinary differential equation with respect to the density ρ ,

$$\frac{d\rho}{dr} = -\left[\rho \frac{\Gamma}{C_p} + \frac{g}{\gamma} \rho^{2-\gamma} \exp\left(-\frac{\gamma S}{C_p}\right)\right] \quad (15)$$

with $\rho = \rho_b$ at $r = r_b$,

where the target entropy S and entropy gradient Γ are already computed. Numerical integration of Eq. (15) gives the initial density profile, which should be a little different from the polytropic solution of the density in Eq. (9). Based on the initial density and entropy profiles, initial profiles of other thermodynamic variables, like pressure p and temperature T , can be computed.

The simulation starts from a thermally relaxed state. Thus, convection will not occur immediately. However, since a constant heat flux flows into the convection zone from the bottom boundary and the temperature at the top boundary is fixed, the temperature gradient will increase and gradually become supercritical. Then, thermal instability can be triggered and convection starts. In fact, how the thermal convection in a rotating spherical shell transitions and enters symmetry-breaking state depends on a wide range of parameters. However, this is not the focus of the current paper. To investigate the dependence of the thermal instability on non-dimensional parameters, Sánchez and Juan²⁹

gave an efficient estimation based on dynamical system tools, which can serve as a parameter setting guide before direct numerical simulations.

III. GRID GENERATION

The physical domain for simulations is a spherical shell. It is partitioned into non-overlapping unstructured hexahedral elements. Traditional grid generation uses isoparametric elements, which are constructed by polynomial-based shape functions defined on nodal points. However, shapes of the hexahedral elements in the current study are described by analytical functions. The current study ensures that elements seamlessly fill in the spherical shell without any singularity.

The grid is generated by stacking a series of surface meshes on horizontal surfaces of different radius of curvature r ($r_b \leq r \leq r_t$) in the radial dimension with equidistant spacings. The number of spacings is N_r . Cubed-sphere gridding technique³⁰ is used to generate the surface meshes. It is based on a decomposition of the sphere into six identical regions, which are obtained by projecting the sides of a circumscribed cube onto a spherical surface, as shown in Fig. 1. Using this projection, tiling a cubed face with $N_a \times N_a$ square elements uniformly is equivalent to constructing the angularly equidistant grids on one of the six identical regions of the sphere. For example, for region,¹ angular variables α and β are chosen,

$$\alpha = \arctan(y/x), \quad \beta = \arctan(z/x), \quad -\pi/4 \leq \alpha, \beta \leq \pi/4,$$

and each element face is enclosed by angularly equidistant curves,

$$[\alpha_i, \alpha_{i+1}] \times [\beta_j, \beta_{j+1}],$$

$$\alpha_i = -\pi/4 + i\pi/2, \quad \beta_j = -\pi/4 + j\pi/2, \quad 0 \leq i, j \leq N_a,$$

as illustrated in Fig. 2(b). Figure 3 shows a sample grid with $N_a = 20$ and $N_r = 8$. The total number of hexahedral elements in the grid is $6N_a^2 N_r$.

IV. NUMERICAL METHODS

A. Transfinite mapping

A transfinite mapping, which can be expressed by analytical functions, is used to transform all elements from the physical domain (x, y, z) into a standard element

$$(\xi, \eta, \zeta) \in [0, 1] \times [0, 1] \times [0, 1] \quad (16)$$

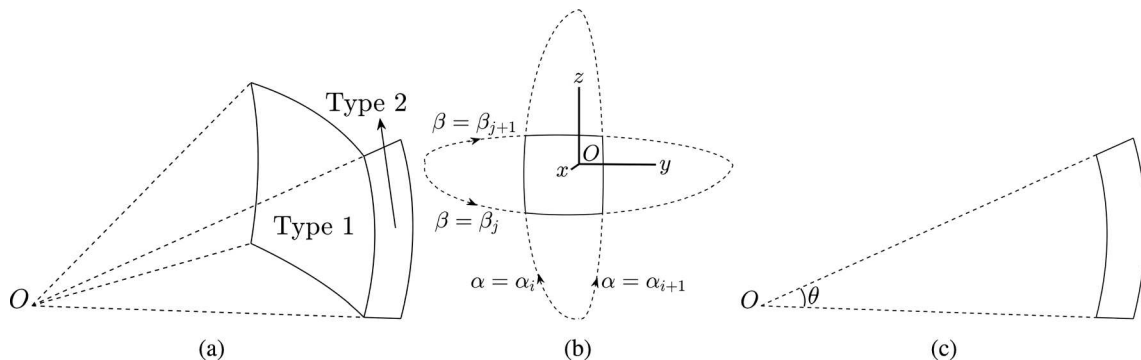


FIG. 2. Element faces belong to two types: type 1 and type 2. (a) Illustration of two faces, which belong to type 1 and type 2, respectively. (b) A face of type 1. It is located in region¹ of the cubed-sphere illustrated in Fig. 1. It is enclosed by four semicircular curves with $\alpha = \alpha_i$ and α_{i+1} and $\beta = \beta_j$ and β_{j+1} . (c) A face of type 2. It is enclosed by two radii and two arcs.

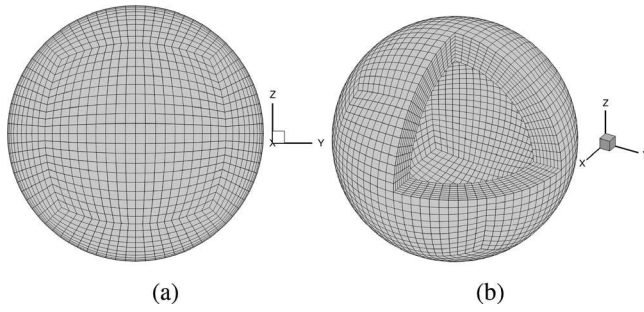


FIG. 3. Illustration of a grid whose $N_\theta = 20$ and $N_\varphi = 8$. (a) Front view of the surface mesh on each horizontal surface generated by the cubed-sphere gridding technique. (b) Volume mesh to partition the simulation domain, which is a spherical shell. The portion of the mesh where $x > 0$, $y > 0$, and $z > 0$ is removed to show the topology of the volume mesh in the interior.

as shown in Fig. 4. The mapping can be expressed as

$$(x, y, z) = \mathbf{P}(\xi, \eta, \zeta). \quad (17)$$

The mapping can be calculated as algebraic combinations of some projectors,³¹

$$\mathbf{P} = \mathbf{P}_\xi + \mathbf{P}_\eta + \mathbf{P}_\zeta - \mathbf{P}_\xi \mathbf{P}_\eta - \mathbf{P}_\xi \mathbf{P}_\zeta - \mathbf{P}_\eta \mathbf{P}_\zeta + \mathbf{P}_\xi \mathbf{P}_\eta \mathbf{P}_\zeta, \quad (18)$$

where projectors $\mathbf{P}_\xi$, $\mathbf{P}_\eta$, and $\mathbf{P}_\zeta$ are computed as

$$\mathbf{P}_\xi(\xi, \eta, \zeta) = (1 - \xi)\mathbf{P}(0, \eta, \zeta) + \xi\mathbf{P}(1, \eta, \zeta), \quad (19)$$

$$\mathbf{P}_\eta(\xi, \eta, \zeta) = (1 - \eta)\mathbf{P}(\xi, 0, \zeta) + \eta\mathbf{P}(\xi, 1, \zeta), \quad (20)$$

$$\mathbf{P}_\zeta(\xi, \eta, \zeta) = (1 - \zeta)\mathbf{P}(\xi, \eta, 0) + \zeta\mathbf{P}(\xi, \eta, 1), \quad (21)$$

projectors of the form of bilinear products are computed as

$$\begin{aligned} \mathbf{P}_\xi \mathbf{P}_\eta(\xi, \eta, \zeta) &= \psi_1(\xi, \eta)\mathbf{P}(0, 0, \zeta) + \psi_2(\xi, \eta)\mathbf{P}(1, 0, \zeta) \\ &\quad + \psi_3(\xi, \eta)\mathbf{P}(1, 1, \zeta) + \psi_4(\xi, \eta)\mathbf{P}(0, 1, \zeta), \end{aligned}$$

$$\begin{aligned} \mathbf{P}_\xi \mathbf{P}_\zeta(\xi, \eta, \zeta) &= \psi_1(\xi, \zeta)\mathbf{P}(0, \eta, 0) + \psi_2(\xi, \zeta)\mathbf{P}(1, \eta, 0) \\ &\quad + \psi_3(\xi, \zeta)\mathbf{P}(1, \eta, 1) + \psi_4(\xi, \zeta)\mathbf{P}(0, \eta, 1), \end{aligned}$$

$$\begin{aligned} \mathbf{P}_\eta \mathbf{P}_\zeta(\xi, \eta, \zeta) &= \psi_1(\eta, \zeta)\mathbf{P}(\xi, 0, 0) + \psi_2(\eta, \zeta)\mathbf{P}(\xi, 1, 0) \\ &\quad + \psi_3(\eta, \zeta)\mathbf{P}(\xi, 1, 1) + \psi_4(\eta, \zeta)\mathbf{P}(\xi, 0, 1) \end{aligned}$$

with the basis functions

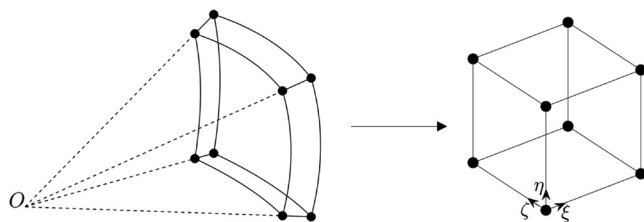


FIG. 4. Transfinite mapping of a hexahedral element with curved surfaces from the physical domain (x, y, z) to the standard cubic domain (ξ, η, ζ) . Point O is the origin of the reference frame.

$$\begin{aligned} \psi_1(\xi, \eta) &= (1 - \xi)(1 - \eta), \\ \psi_2(\xi, \eta) &= \xi(1 - \eta), \\ \psi_3(\xi, \eta) &= \xi\eta, \\ \psi_4(\xi, \eta) &= (1 - \xi)\eta, \end{aligned}$$

and the trilinear projector, which corresponds to the trilinear isoparametric mapping, is computed as

$$\begin{aligned} \mathbf{P}_\xi \mathbf{P}_\eta \mathbf{P}_\zeta(\xi, \eta, \zeta) &= (1 - \xi)(1 - \eta)(1 - \zeta)\mathbf{P}(0, 0, 0) \\ &\quad + (1 - \xi)\eta(1 - \zeta)\mathbf{P}(0, 1, 0) \\ &\quad + (1 - \xi)\eta(1 - \zeta)\mathbf{P}(0, 1, 0) \\ &\quad + (1 - \xi)(1 - \eta)\zeta\mathbf{P}(0, 0, 1) + \xi\eta(1 - \zeta)\mathbf{P}(1, 1, 0) \\ &\quad + \xi(1 - \eta)\zeta\mathbf{P}(1, 0, 1) + (1 - \xi)\eta\zeta\mathbf{P}(0, 1, 1) \\ &\quad + \xi\eta\zeta\mathbf{P}(1, 1, 1). \end{aligned}$$

Notice that $\mathbf{P}(\xi, \eta, \zeta)$ is a 3D mapping from a physical volume to a standard cubic volume, but relying on the above computation, we can instead compute one-dimensional (1D) mappings from standard straight lines to element edges in the physical domain, like $\mathbf{P}(\xi, 0, 0)$, and two-dimensional (2D) mappings from standard square faces to curved element faces, like $\mathbf{P}(\xi, \eta, 0)$. For all hexahedral elements, which are depicted in Fig. 4, there are four straight edges in the radial dimension and eight edges, which are arcs on spherical surfaces. 1D mappings from $[0, 1]$ to straight edges are simply 1D isoparametric mappings. 1D mappings from $\xi \in [0, 1]$ to arcs are linear mappings between ξ and the arc angle ϑ . For 2D mappings, all elements have six faces, which can be divided into two types, as illustrated in Fig. 2. There are two faces of type 1. They are located on spherical surfaces and generated by the cubed-sphere surface meshing. Therefore, the 2D mapping $\mathbf{P}(\xi, \eta, 0)$, which maps a standard square face onto a face enclosed by the semicircular curves $\alpha = \alpha_i$ and α_{i+1} and $\beta = \beta_j$ and β_{j+1} with radius r , can be done by

$$\alpha = \alpha_i + \xi\Delta\alpha, \quad \beta = \beta_j + \eta\Delta\beta, \quad \Delta\alpha = \Delta\beta = \pi/(2N_\theta),$$

$$x = r/\sqrt{1 + \tan^2\alpha + \tan^2\beta}, \quad y = x \tan \alpha, \quad z = x \tan \beta.$$

Remaining four faces belong to type 2 and are composed of two straight radial edges and four arc edges. The 2D mapping is a tensor product of an isoparametric mapping along the radial dimension and a linear mapping of a co-ordinate onto the arc angle ϑ .

When mapping one point (ξ, η) in the standard square face to a point (x, y, z) in 3D space on a $1/6$ spherical surface with radius r , one face among the six faces in Fig. 1, geometric errors are measured by $\Delta r = \sqrt{x^2 + y^2 + z^2} - r$. Figure 5 shows that geometric errors introduced by the isoparametric mappings are exactly zero at nodal points where the shape functions are defined, for example, like corner points and mid-edge points. However, errors are extremely large in the center of the face. The maximum non-dimensional geometric error $|\Delta r|/r$ is 42%, 16%, and less than $10^{-10}\%$ for linear isoparametric mapping, quadratic isoparametric mapping, and transfinite mapping, respectively. Such large geometric errors introduced by the isoparametric mappings cannot be removed without massive meshing. By contrast, the geometric error introduced by the transfinite mapping is always near the error of machine precision and is irrelevant with the mesh resolution.

The Jacobian matrix J for the mapping $\mathbf{P}(\xi, \eta, \zeta)$ is

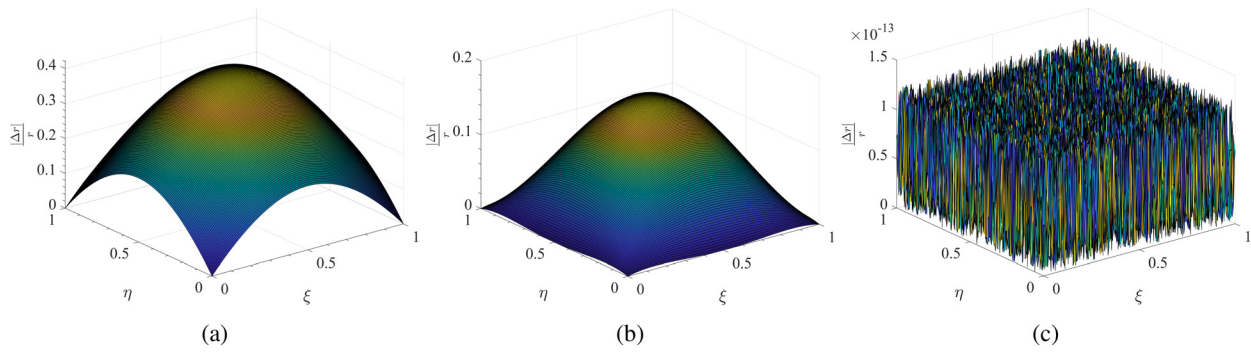


FIG. 5. Comparison of (a) linear isoparametric mapping, (b) quadratic isoparametric mapping, and (c) transfinite mapping in terms of non-dimensional geometric errors introduced by mapping the standard square face $(\xi, \eta) \in [0, 1]^2$ to a 1/6 spherical surface.

$$J = \frac{\partial(x, y, z)}{\partial(\xi, \eta, \zeta)} = \begin{pmatrix} x_\xi & x_\eta & x_\zeta \\ y_\xi & y_\eta & y_\zeta \\ z_\xi & z_\eta & z_\zeta \end{pmatrix}. \quad (22)$$

Using the transfinite mapping, the governing equations (1) are then transformed into the computational domain within the standard cubic element

$$\frac{\partial \tilde{\mathbf{Q}}}{\partial t} + \frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} + \frac{\partial \tilde{\mathbf{H}}}{\partial \zeta} = \tilde{\mathbf{M}}, \quad (23)$$

where $\tilde{\mathbf{Q}} = |J|\mathbf{Q}$, $\tilde{\mathbf{M}} = |J|\mathbf{M}$ and fluxes are transformed as

$$\begin{pmatrix} \tilde{\mathbf{F}} \\ \tilde{\mathbf{G}} \\ \tilde{\mathbf{H}} \end{pmatrix} = |J|J^{-1} \begin{pmatrix} \mathbf{F} \\ \mathbf{G} \\ \mathbf{H} \end{pmatrix}, \quad (24)$$

in which fluxes can be written explicitly as

$$\begin{aligned} \mathbf{F}_{\text{inv}} &= (\rho u, \rho u^2 + p, \rho uv, \rho uw, u(E + p))^T, \\ \mathbf{G}_{\text{inv}} &= (\rho v, \rho vu, \rho v^2 + p, \rho vw, v(E + p))^T, \\ \mathbf{H}_{\text{inv}} &= (\rho w, \rho wu, \rho wv, \rho w^2 + p, w(E + p))^T, \end{aligned} \quad (25)$$

and

$$\begin{aligned} \mathbf{F}_{\text{vis}} &= (\tau_{xx}, \tau_{yx}, \tau_{zx}, u\tau_{xx} + v\tau_{yx} + w\tau_{zx} + q_x)^T, \\ \mathbf{G}_{\text{vis}} &= (\tau_{xy}, \tau_{yy}, \tau_{zy}, u\tau_{xy} + v\tau_{yy} + w\tau_{zy} + q_y)^T, \\ \mathbf{H}_{\text{vis}} &= (\tau_{xz}, \tau_{yz}, \tau_{zz}, u\tau_{xz} + v\tau_{yz} + w\tau_{zz} + q_z)^T, \end{aligned} \quad (26)$$

where q_x , q_y , and q_z are three components of the heat flux vector $\mathbf{q}$ in the x , y , and z directions and τ_{ij} are the components of the shear stress tensor in Eq. (3).

B. Spectral difference method

Within the computational domain, in order to construct a degree $(N - 1)$ polynomial along the ξ , η or ζ dimension, N solution points (SPs) are required in each dimension. In each dimension, the SPs are chosen as the Chebyshev–Gauss points,

$$X_s = \frac{1}{2} \left[1 - \cos \left(\frac{2s - 1}{2N} \pi \right) \right], \quad s = 1, 2, \dots, N. \quad (27)$$

To be consistent with the conservation law, where spatial derivatives of the fluxes and the conserved variables are on the same stage, the polynomial to reconstruct the fluxes should have degree N . Hence, $(N + 1)$ flux points (FPs) are needed in each dimension. They are denoted as $X_{s+1/2}$, $s = 0, 1, \dots, N$ and chosen as $X_{1/2} = 0$ and $X_{N+1/2} = 1$ plus $(N - 1)$ roots of the N th order Legendre polynomial. In each dimension, Lagrange interpolation is used to construct solution polynomials and reconstruction flux polynomials with nodal Lagrange basis functions at the SPs and FPs,

$$h_i(X) = \prod_{s=1, s \neq i}^N \left(\frac{X - X_s}{X_i - X_s} \right), \quad (28)$$

$$l_{i+1/2}(X) = \prod_{s=0, s \neq i}^N \left(\frac{X - X_{s+1/2}}{X_{i+1/2} - X_{s+1/2}} \right). \quad (29)$$

For the 3D computational domain, the SPs are located at $(\xi, \eta, \zeta) = (X_i, X_j, X_k)$, where $i, j, k = 1, 2, \dots, N$. The FPs can be divided into three families along three dimensions, which are

$$\begin{aligned} (\xi, \eta, \zeta) &= (X_{i+1/2}, X_j, X_k), & \text{family along } \xi\text{-dimension,} \\ & i = 0, 1, \dots, N, & j, k = 1, 2, \dots, N, \\ (\xi, \eta, \zeta) &= (X_i, X_{j+1/2}, X_k), & \text{family along } \eta\text{-dimension,} \\ & j = 0, 1, \dots, N, & i, k = 1, 2, \dots, N, \\ (\xi, \eta, \zeta) &= (X_i, X_j, X_{k+1/2}), & \text{family along } \zeta\text{-dimension,} \\ & k = 0, 1, \dots, N, & i, j = 1, 2, \dots, N. \end{aligned}$$

Tensor products of the three one-dimensional Lagrange basis functions construct solutions,

$$\mathbf{Q}(\xi, \eta, \zeta) = \sum_{k=1}^N \sum_{j=1}^N \sum_{i=1}^N \frac{\tilde{\mathbf{Q}}_{i,j,k}}{|\tilde{\mathbf{Q}}_{i,j,k}|} h_i(\xi) \cdot h_j(\eta) \cdot h_k(\zeta). \quad (30)$$

After extrapolating solutions from the SPs to the FPs, inviscid fluxes are computed at the FPs. However, they are only element-wise continuous, but discontinuous across element interfaces. For the inviscid flux, the Rusanov solver³² (also known as Local Lax–Friedrichs solver) is employed to compute common inviscid fluxes at interfaces to ensure flux conservation. The Rusanov solver is implemented in the computational domain along interface normal direction. At the same time, to ensure that solutions are continuous across element interfaces, the

common solution at element interfaces is computed as the average of values interpolated from the left and right elements (also known as the BR1 scheme³³). Sun *et al.*³⁴ proposed an efficient way to compute solution gradients in 3D unstructured grids,

$$\begin{aligned}\nabla \mathbf{Q} &= \frac{\partial \mathbf{Q}}{\partial \xi} \nabla \xi + \frac{\partial \mathbf{Q}}{\partial \eta} \nabla \eta + \frac{\partial \mathbf{Q}}{\partial \zeta} \nabla \zeta \\ &= \frac{1}{|J|} \left[\frac{\partial(\mathbf{Q} \vec{S}_\xi)}{\partial \xi} + \frac{\partial(\mathbf{Q} \vec{S}_\eta)}{\partial \eta} + \frac{\partial(\mathbf{Q} \vec{S}_\zeta)}{\partial \zeta} \right],\end{aligned}$$

where $\vec{S}_\xi = |J| \nabla \xi$, $\vec{S}_\eta = |J| \nabla \eta$, $\vec{S}_\zeta = |J| \nabla \zeta$ and the derivatives at the SPs along each dimension are computed,

$$\begin{aligned}\left(\frac{\partial \vec{S}_\xi}{\partial \xi} \right)_{i,j,k} &= \sum_{r=0}^N (\mathbf{Q} \vec{S}_\xi)_{r+1/2,j,k} \cdot l'_{r+1/2}(X_i), \\ \left(\frac{\partial \vec{S}_\eta}{\partial \eta} \right)_{i,j,k} &= \sum_{r=0}^N (\mathbf{Q} \vec{S}_\eta)_{i,r+1/2,k} \cdot l'_{r+1/2}(X_j), \\ \left(\frac{\partial \vec{S}_\zeta}{\partial \zeta} \right)_{i,j,k} &= \sum_{r=0}^N (\mathbf{Q} \vec{S}_\zeta)_{i,j,r+1/2} \cdot l'_{r+1/2}(X_k).\end{aligned}\quad (31)$$

Then, the solution gradients are extrapolated from the SPs to the FPs. Again, the common solution gradients are averaged on element interfaces. After obtaining both solutions and solution gradients at the FPs, viscous fluxes are computed at the FPs. The total fluxes, which combine both the inviscid and viscous fluxes, are reconstructed,

$$\begin{aligned}\tilde{\mathbf{F}} &= \sum_{k=1}^N \sum_{j=1}^N \sum_{i=0}^N \tilde{\mathbf{F}}_{i+1/2,j,k} l_{i+1/2}(\xi) \cdot h_j(\eta) \cdot h_k(\zeta), \\ \tilde{\mathbf{G}} &= \sum_{k=1}^N \sum_{j=0}^N \sum_{i=1}^N \tilde{\mathbf{G}}_{i,j+1/2,k} h_i(\xi) \cdot l_{j+1/2}(\eta) \cdot h_k(\zeta), \\ \tilde{\mathbf{H}} &= \sum_{k=0}^N \sum_{j=1}^N \sum_{i=1}^N \tilde{\mathbf{H}}_{i,j,k+1/2} h_i(\xi) \cdot h_j(\eta) \cdot l_{k+1/2}(\zeta).\end{aligned}\quad (32)$$

Spatial derivatives of fluxes at the SPs are then computed,

$$\left(\frac{\partial \tilde{\mathbf{F}}}{\partial \xi} \right)_{i,j,k} = \sum_{r=0}^N \tilde{\mathbf{F}}_{r+1/2,j,k} \cdot l'_{r+1/2}(X_i), \quad (33)$$

$$\left(\frac{\partial \tilde{\mathbf{G}}}{\partial \eta} \right)_{i,j,k} = \sum_{r=0}^N \tilde{\mathbf{G}}_{i,r+1/2,k} \cdot l'_{r+1/2}(X_j), \quad (34)$$

$$\left(\frac{\partial \tilde{\mathbf{H}}}{\partial \zeta} \right)_{i,j,k} = \sum_{r=0}^N \tilde{\mathbf{H}}_{i,j,r+1/2} \cdot l'_{r+1/2}(X_k). \quad (35)$$

C. Time stepping scheme

After computing the spatial derivatives of fluxes, Eq. (23) can be written in a residual form,

$$\frac{\partial \mathbf{Q}}{\partial t} = -\frac{1}{|J|} \left(\frac{\partial \tilde{\mathbf{F}}}{\partial \xi} + \frac{\partial \tilde{\mathbf{G}}}{\partial \eta} + \frac{\partial \tilde{\mathbf{H}}}{\partial \zeta} \right) + \mathbf{M}. \quad (36)$$

A five-stage third-order explicit strong stability-preserving Runge–Kutta method [SSPRK(5,3)] is used in the present work, whose coefficients are tabulated in Table 1 of Ruuth.³⁵

V. PARALLEL COMPUTING

The CHORUS++ code is written in Fortran 90, and the message passing interface (MPI) is used for interprocessor communication. It uses the METIS package³⁶ for domain decomposition. Before the parallel computations start, the root processor calls the `mpmetis` program, a serial program, to partition the mesh into parts. Then, each processor reads in its own part of the mesh. Due to the compactness of the SD method, most computations are done locally, except information communications occurring at element interfaces whose two neighboring elements are located in different processors. By comparing with the load for local computations, the load for communications is negligible since the number of FPs on interprocessor element interfaces is much smaller than the number of other FPs.

Figure 6 shows the strong scaling for CHORUS++ using third-, fourth-, and sixth-order elements. This test is carried on a cluster whose nodes are equipped with 2-sockets Intel Xeon Gold 6148 CPU with 20 cores each at 2.40 GHz for a total of 40 cores per node. Linear speedup is well achieved up to 40 processors for all orders. With the increase in the number of processors, CHORUS++ gradually deviates from the linear speedup due to increased portion of interprocessor communications in overall computational costs. When the number of processors becomes very large, CHORUS++ using sixth-order elements has better speedup than CHORUS++ using lower-order elements. This is consistent with the fact that the computational cost in the interiors of elements is proportional to N^3 while the communication cost on the element interfaces is proportional to N^2 .

VI. LAMINAR SOLAR BENCHMARK

To test the accuracy of the CHORUS++ code, it is applied to the laminar solar benchmark proposed by Wang *et al.*¹⁷ This benchmark has been investigated thoroughly using both the original CHORUS code and ASH code and can serve as a test case for verification of CHORUS++. Parameters for the laminar solar benchmark are listed

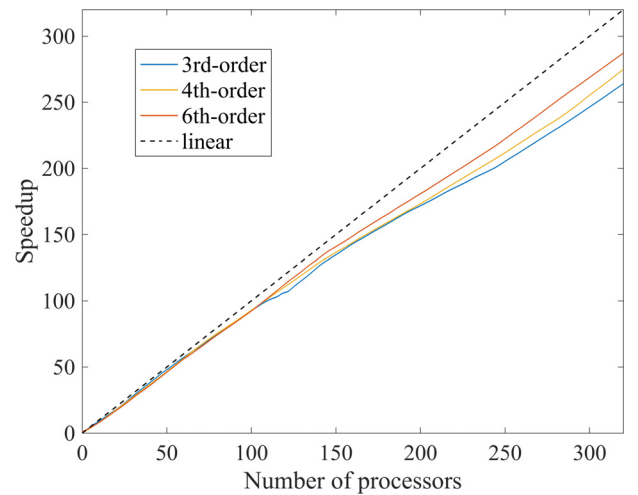


FIG. 6. Strong scaling of CHORUS++.

TABLE I. Summary of parameters for the laminar solar benchmark.

Top boundary $r_t = 6.61 \times 10^{10}$ cm, bottom boundary $r_b = 4.87 \times 10^{10}$ cm, depth of the convection zone $d = r_t - r_b = 1.74 \times 10^{10}$ cm, stellar luminosity $L_\odot = 3.846 \times 10^{36}$ ergs s ⁻¹ , gravitational constant $G = 6.67 \times 10^{-8}$ g ⁻² cm ² , stellar mass $M_\odot = 1.98891 \times 10^{33}$ g, $\rho_b = 0.21$ g cm ⁻³ , $N_\rho = 3$, $\gamma = 5/3$, gas constant $\mathcal{R} = 1.4 \times 10^8$ ergs g ⁻¹ K ⁻¹ , $C_p = 3.5 \times 10^8$ ergs g ⁻¹ K ⁻¹ , kinematic viscosity $\nu = 6.0 \times 10^{13}$ cm ² s ⁻¹ , entropy diffusion coefficient $\kappa = 6.0 \times 10^{13}$ cm ² s ⁻¹ , $\Omega_0 = 8.1 \times 10^{-5}$ s ⁻¹

in Table I. We use the Sun values for the length of the solar convection zone, as reflected by values of r_t and r_b . The radiative diffusivity is parameterized as $\kappa_r = \lambda_r(c_0 + c_1\omega + c_2\omega^2)$, where $c_0 = 1.560\,097\,5 \times 10^8$, $c_1 = -4.563\,171\,8 \times 10^7$, $c_2 = 3.337\,036\,8 \times 10^6$, and $\omega = r \times 10^{-10}$. The parameter λ_r is set by making sure that $F_r = L_\odot/(4\pi r^2)$ at $r = r_b$, where F_r is the radiative flux in Eq. (39). The current study artificially increases the stellar luminosity $L_\odot$ 1000 times to achieve higher Mach numbers^{17,25,26} so that the severity of the CFL constraint can be alleviated to some extent. However, with the boosted luminosity, the convective flow is still in a low-Mach regime since the maximum Mach number in the whole domain during simulations is less than 0.05. The flow velocities in the near-photospheric layers would not become supersonic since the number of density scale heights is chosen as 3, a small value representing that the change in the Mach number along the radial dimension would not be as drastic as the real Sun. As a consequence of the boosted luminosity, the root mean square (rms) of fluid velocity is enlarged ten times according to the mixing-length theory of convection that $U'_{\text{rms}} \propto L_\odot^{1/3}$, where $U'_{\text{rms}} = \sqrt{\langle U'^2 \rangle}$, $\langle \cdot \rangle$ represents the time-averaged operation, the overbar represents the spatially averaged operation, $U'^2 = u'^2 + v'^2 + w'^2$, and $u' = u - \langle u \rangle$. To preserve a solar-like Rossby number $\text{Ro} = U'_{\text{rms}}/(2\Omega_0 d)$, the rotational velocity should be enlarged for ten times accordingly. Given that the rotational velocity for the real Sun is $\Omega_0^{\text{Sun}} = 2.6 \times 10^{-6}$ s⁻¹, the laminar solar benchmark considers a fast-rotating Sun and uses $\Omega_0 = 8.1 \times 10^{-5}$ s⁻¹ $\approx 31.2\Omega_0^{\text{Sun}}$. Note that even if the value of $(L_\odot)^{1/3}/\Omega_0$ in the simulation matches that of the real Sun, the convective structure and predicted convective velocities cannot match that on the Sun exactly. Much evidence suggests that the large-scale convective velocities obtained by solar convection simulations might be over-estimated. This is called the convective conundrum.³⁷ There are many ongoing works^{38–40} about exploring factors that may affect the amplitude of the convective velocities and approaching the solar-like convection regime. Table II shows that the non-dimensional parameters for the Sun and laminar solar benchmark and disparity in Reynolds number can be noticed. Data for the Sun is from Jones *et al.*⁴¹

Six different grids, G1, G2, ..., G6, are used for computations of the laminar solar benchmark, as shown in Table III. According to the number of solution degrees of freedom (DoFs), they can be divided into two groups. G1, G3, and G5 belong to the first group, and G2, G4, and G6, which are refined 1.5 times along all dimensions, belong to the second group. For the SD method, DoFs measure computational costs per time step. Comparisons within a group allow us to investigate the superiority of high-order accuracy objectively.

All simulations run up to the time when the equilibration of components of the energy flux almost reaches. Neglecting structural stellar evolution, the thermal relaxation occurs in the thermal

timescale or Kelvin–Helmholtz timescale $T_{KH} = E_t/L_\odot$, where E_t is the total thermal energy. For rotating stars, T_{KH} is generally very large and can exceed 10^5 years. However, a more relevant timescale for the equilibration of the convection is the thermal diffusion timescale $T_d = d^2/\kappa$, which equals to $58.44T_\odot$ for the solar benchmark, where $T_\odot = 2\pi/\Omega_0$ is one rotational period of time for stars. To ensure final equilibration, all simulations run up to $t = T_f = 4.8 \times 10^5$ s $\approx 61.88T_\odot$. This simulation time is much longer than the affordable simulation time for the original CHORUS code,¹⁷ which is $15T_\odot$. The time interval $T_f < t < T_f + 30T_\odot$ is chosen as the sampling time for computing time-averaged values. $U'_{\text{rms}} = 4.0398 \times 10^4$ cm s⁻¹ for the laminar solar benchmark, and non-dimensional parameters in Table II are computed based on this value.

Figure 7 shows that the kinetic energy,

$$\text{KE} = \frac{1}{V} \int_V \frac{1}{2} \rho (\mathbf{U} \cdot \mathbf{U}) dV, \quad (37)$$

achieves a nonlinear saturation and reaches a dynamic equilibrium after a fast exponential growth stage and a descending stage. Lack of grid resolution would underestimate KE, as shown by the predicted line of G1. Figure 8 shows convective patterns predicted from the six grids. Light fluids with high temperature rise to the top and form the upflow indicated by the red color while heavy fluids with low temperature sink to the bottom and form the downflow indicated by the blue color. Given the same DoFs, compared with G3 and G5, G1 with third-order elements gives the most crude solution: there are many

TABLE II. Comparison between some non-dimensional parameters for the Sun and the laminar solar benchmark. For the benchmark, $\Delta S = 7.798 \times 10^5$ erg g⁻¹ K⁻¹ and $U'_{\text{rms}} = 4.040 \times 10^4$ cm s⁻¹.

Parameters	Sun	benchmark
Rayleigh number	10^{20}	1.429×10^6
$\text{Ra} = GM_\odot d \Delta S / (\nu \kappa C_p)$		
Reynolds number	10^{12}	11.72
$\text{Re} = U'_{\text{rms}} d / \nu$		
Ekman number	10^{-14}	2.447×10^{-3}
$\text{Ek} = \nu / (\Omega_0 d^2)$		
Taylor number	$10^{19} \sim 10^{27}$	6.682×10^5
$\text{Ta} = 4\Omega_0^2 d^4 / \nu^2$		
Prandtl number	$10^{-6} \sim 10^{-4}$	1
$\text{Pr} = \nu / \kappa$		
Rossby number	0.1–1	1.433×10^{-2}
$\text{Ro} = U'_{\text{rms}} / (2\Omega_0 d)$		
Number of density scale heights $N_\rho = \ln(\rho_b/\rho_t)$	16	3

TABLE III. Computational efforts for the laminar solar benchmark using CHORUS++ on G1, G2, ..., G6 are shown. The number of elements in grids are $6 \times (N_a^2 \times N_r)$ and $N_a/N_r = 1.5$ for all grids. Computational efforts for the same benchmark using CHORUS and ASH are attached for comparison. The sparsest grid needed for a third-order CHORUS++ to get converged components of energy flux is given in the last column.

	G1 $6 \times (24^2 \times 16)$ =55 296	G2 $6 \times (36^2 \times 24)$ =186 624	G3 $6 \times (18^2 \times 12)$ =23 328	G4 $6 \times (27^2 \times 18)$ =78 732	G5 $6 \times (12^2 \times 8)$ =6912	G6 $6 \times (18^2 \times 12)$ =233 328
No. of elements						
SD order	Third-order	Third-order	Fourth-order	Fourth-order	Sixth-order	Sixth-order
DoFs	1 492 992	5 038 848	1 492 992	5 038 848	1 492 992	5 038 848
Max Δt (s)	17.9	11.5	13.8	8.6	8.5	5.0
CPU hours per time step	1.03×10^{-3}	3.48×10^{-3}	1.05×10^{-3}	3.54×10^{-3}	1.01×10^{-3}	3.41×10^{-3}
CPU hours to run 10 days	44.6	234.7	59.0	319.3	92.2	529.0
Valid or not	×	×	×	×	✓	✓
	CHORUS		ASH		Third-order CHORUS++	
DoFs	19 660 800		$100 \times 256 \times 512 = 2\,907\,000$		$6 \times (75^2 \times 50) \times 3^3 = 45\,562\,500$	
Δt (s)	4		20		5.3	
CPU hours per time step	8.03×10^{-2}		5.32×10^{-3}		3.16×10^{-2}	
CPU hours to run 10 days	17 352.0		230.0		4624.9	

small artificial pothole-shaped convective cells in the mid-latitude region, and contours of convective cells near the equator are not quite smooth. Using sixth-order elements, the so-called banana cells are captured without obvious oscillations shown on the contour lines: columnar convective rolls are aligned with the rotation axis but sheared slightly in the prograde direction near the equator by the differential rotation. After refining the grids, solution smoothness of G2 is much higher than that of G1, but convective patterns predicted from G5 and G6 are almost the same, which means that the simulation is almost converged for G6. By comparing with results from G6, banana cells predicted from G2 are wider in the longitudinal direction and shorter in the latitudinal direction.

To check whether the convection reaches equilibration, the radial energy flux needs to be examined. Four components of the energy flux are involved in transporting energy in the radial direction, namely, the enthalpy flux F_e , radiative flux F_r , entropy flux F_u , and kinetic energy

flux F_k . Appendix B in Ref. 26 derives the equation of flux balance in detail. When the system reaches a statistically steady state, the sum of these four fluxes should equal to the full luminosity imposed at the bottom boundary

$$F_e + F_r + F_u + F_k = F_\odot = \frac{L_\odot}{4\pi r^2}, \quad (38)$$

where

$$\begin{aligned} F_e &= \bar{\rho} C_p U_r (T - \bar{T}), \\ F_r &= -\kappa_r \bar{\rho} C_p \frac{\partial \bar{T}}{\partial r}, \\ F_u &= -\kappa_p \bar{\rho} T \frac{\partial \bar{S}}{\partial r}, \\ F_k &= \frac{1}{2} \bar{\rho} U_r (\mathbf{U} \cdot \mathbf{U}), \end{aligned} \quad (39)$$

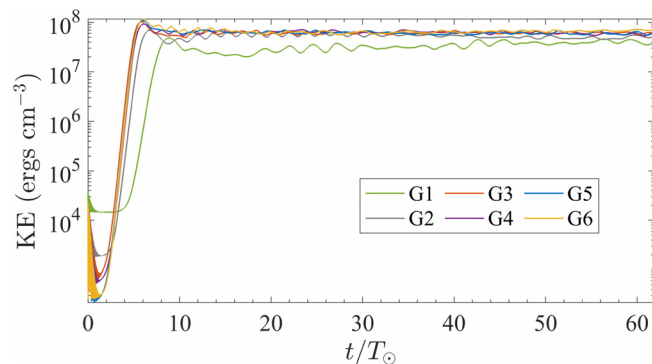


FIG. 7. Time evolution of the kinetic energy computed from six different grids in the laminar solar benchmark.

the overbars denote mean values averaged over horizontal surfaces. Figure 9 shows the decomposition of the energy flux at $t = T_f$ for solutions computed from CHORUS++ on the six grids. To make a comparison, results based on CHORUS and ASH are also attached.¹⁷ Due to prohibitive computational costs, even with Yellowstone high-performance computing clusters at National Center for Atmospheric Research (NCAR), CHORUS fails to run the benchmark until equilibration of the energy flux, as $\max((F_e + F_r + F_u + F_k)/F_\odot - 1) \approx 30\%$ in Fig. 9(g). Using sixth-order CHORUS++, predicted components of the energy flux are smooth in Figs. 9(e) and 9(f). High consistency between them verifies the convergence of simulations. The energy flux balance is almost achieved for G5 and G6 at $t = T_f$ since $\max((F_e + F_r + F_u + F_k)/F_\odot - 1) \approx 5\%$. Results show that the radiative flux F_r dominates the energy flux in the lower convection zone while the entropy flux F_u dominates in the upper convection zone.

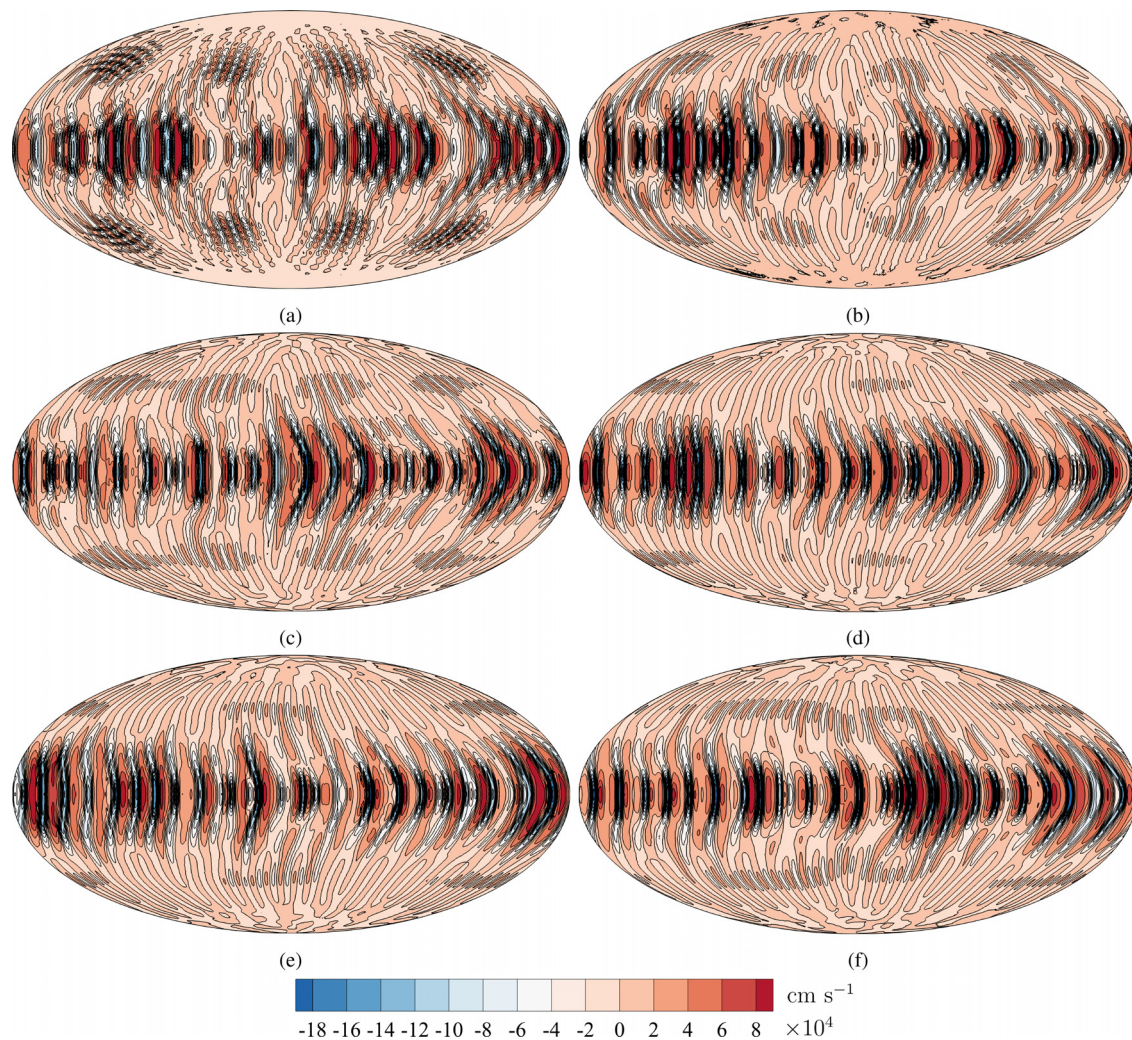


FIG. 8. Instantaneous radial velocity $U_r = \mathbf{U} \cdot \hat{r}$ for the laminar solar benchmark at the horizontal surface $r = 0.95r_t$. Red and blue tones denote the upflow and downflow, respectively, as indicated by the color bar. The horizontal surfaces are displayed in a Mollweide projection that includes all 360° of longitude and in which lines of constant latitude are horizontal. Shown are results computed on (a) G1, (b) G2, (c) G3, (d) G4, (e) G5, and (f) G6.

Both F_e and F_k decrease to 0 at boundaries due to the impenetrable boundary conditions. The enthalpy flux F_e peaks at about $r = 0.95r_t$ with a peak value close to $0.3L_\odot$. By contrast, all third- and fourth-order CHORUS++ fail at the same DoFs since spurious oscillations of F_u appear. These oscillations are not physical and stem from lack of spatial discretization accuracy. Increasing mesh resolution can alleviate this issue. Refining G3 to G4, the amplitude of unphysical oscillations in F_u decreases significantly. Spurious oscillations of F_u computed from G1 and G2 are too violent to show with the same scale of other components. Refining G1 to G2 also results in a physical prediction of the enthalpy flux F_e . With improved mesh resolution, boundary conditions are better satisfied. It can be noticed that for G3, the boundary condition of heat flux at the bottom boundary is not even satisfied well. After refining the mesh from G3 to G4, $F_r + F_u = F_\odot$ satisfies extremely well at both the bottom and top boundaries. However, the sixth-order

CHORUS++ does not have all these problems at the same DoFs. It demonstrates that given the same DoFs, sixth-order accuracy provides enhanced simulation fidelity. To achieve the same level of accuracy, higher-order CHORUS++ needs much fewer DoFs.

Table III compares the computing efficiency of CHORUS++, CHORUS, and ASH. Data for CHORUS and ASH is collected from Ref. 17. Sixth-order CHORUS++ is far more efficient than CHORUS. However, since CPU hours for CHORUS++ and CHORUS are measured on different computing architectures, this conclusion needs to be further verified. Based on the current third-order CHORUS++, we refine the mesh resolution until the energy flux is fully resolved. Numerical experiments show that $N_r = 50$ is needed to get converged components of the energy flux. The sixth-order CHORUS++ runs $4624.9/529.0 \approx 8.7$ times faster than the third-order CHORUS++. This amazing speedup enables us to carry out the RCC simulation on G6 until equilibration of

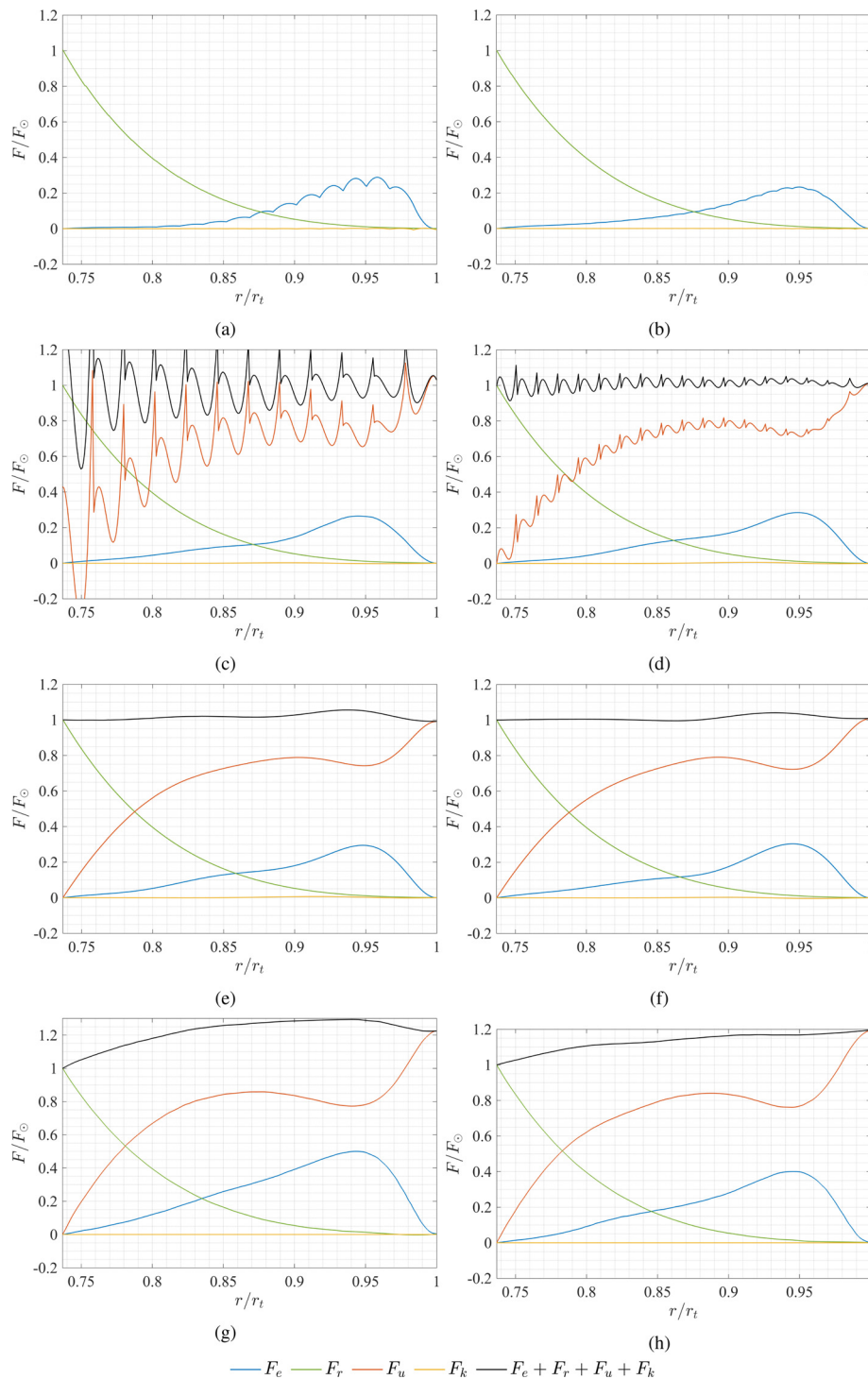


FIG. 9. Components of the normalized, horizontally integrated radial energy flux for the laminar solar benchmark computed on (a) G1, (b) G2, (c) G3, (d) G4, (e) G5, and (f) G6 using CHORUS++. Spurious oscillations of the entropy flux F_u computed using the third-order accuracy on G1 and G2 are too violent to show directly. Predictions for the same problem using CHORUS and ASH are shown in (g) and (h), respectively.¹⁷

components of the energy flux using just one Intel Xeon Gold 6148 Processor about 1.5 weeks. Here, one Intel Xeon Gold 6148 Processor has 40 total threads and 2.4 GHz base frequency. If we neglect the difference of computing facilities, the computing

efficiency of the sixth-order CHORUS++ code is even close to that of the ASH code. However, CHORUS++ is implemented on unstructured grids, based on equations of compressible flows, and orders of spatial accuracy can be chosen arbitrary.

TABLE IV. Summary of non-dimensional parameters for solar convection simulations with varied rotational velocity Ω_0 .

Ω_0 (s^{-1})	2.6×10^{-5}	3.7×10^{-5}	4.8×10^{-5}	5.9×10^{-5}	7.0×10^{-5}	8.1×10^{-5}
U'_{rms} (cm s^{-1})	8.573×10^4	7.548×10^4	6.697×10^4	5.745×10^4	4.922×10^4	4.040×10^4
Reynolds number Re	24.86	21.89	19.42	16.66	14.27	11.72
Rossby number Ro	3.041×10^{-2}	2.678×10^{-2}	2.376×10^{-2}	2.038×10^{-2}	1.746×10^{-2}	1.433×10^{-2}

The CFL constraint for discontinuous high-order methods with explicit Runge–Kutta time marching schemes⁴² is formulated as

$$\Delta t \propto h/(p_o^2) \quad (40)$$

with empirical data close to

$$\Delta t \propto h/(p_o^{1.4}), \quad (41)$$

where Δt is the time step, h is the element size, and p_o is the degree of the polynomials for local reconstruction. For the SD method, $p_o = N - 1$, where N is the order of the SD method. Given the computational domain size and the number of computational elements in the radial direction is l and N_e and the total number of DoFs in the radial direction is fixed as N_{DoFs} , it follows that $h = l/N_e$, $N = N_{\text{DoFs}}/N_e$. We then reach the expression that

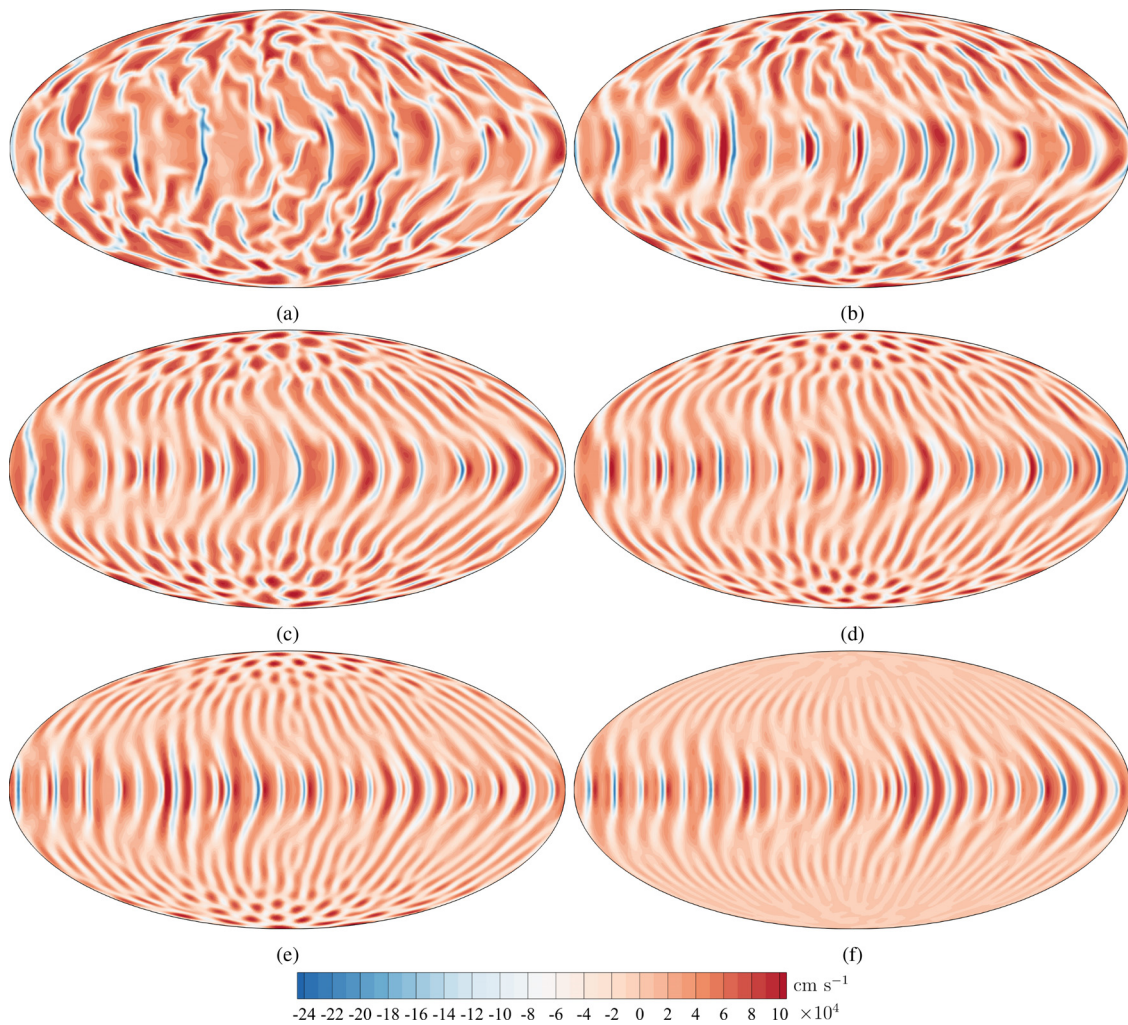


FIG. 10. Instantaneous radial velocity $U_r = \mathbf{U} \cdot \hat{\mathbf{r}}$ at the horizontal surface $r = 0.95r_i$ for the cases of varied rotational velocity. The horizontal surfaces are displayed in a Mollweide projection. (a) $\Omega_0 = 2.6 \times 10^{-5}$, (b) $\Omega_0 = 3.7 \times 10^{-5}$, (c) $\Omega_0 = 4.8 \times 10^{-5}$, (d) $\Omega_0 = 5.9 \times 10^{-5}$, (e) $\Omega_0 = 7.0 \times 10^{-5}$, and (f) $\Omega_0 = 8.1 \times 10^{-5} \text{ s}^{-1}$.

$$\Delta t \propto \frac{N_e^{0.4}}{(N_{\text{DoFs}} - N_e)^{1.4}}. \quad (42)$$

According to the above equation, changing the mesh from G1 to G3 would reduce the maximum time step by a factor of 1.32, which is computed as $17.9/13.8 = 1.30$ in real numerical computations. The slight discrepancy is acceptable due to the nonlinearity of compressible NS equations, non-uniform distribution of the source term, and radiative diffusivity κ_r . Overall, increasing the polynomial orders significantly improves the spatial accuracy at the sacrifice of acceptably shortened time steps.

VII. VARYING THE ROTATIONAL VELOCITY

Vasil *et al.*⁴³ predicted that the dynamical Rossby number for real solar convection is less than unity below the near-surface shear layer, and, therefore, the convection is rotationally constrained. Investigation of the rotational effect is of great important to studies of the solar convection. Rotation tends to suppress convection.⁴⁴ For a weak rotation, small-scale flow structures are reduced with increased rotational velocity.⁴⁵

By varying the rotational velocity Ω_0 in the laminar solar benchmark, we investigate the rotationally constrained effect on convection. Five simulations with $\Omega_0 = 2.6 \times 10^{-5}$, 3.7×10^{-5} , 4.8×10^{-5} , 5.9×10^{-5} , and $7.0 \times 10^{-5} \text{ s}^{-1}$ are computed on G6 using sixth-order elements. Along with the original solar benchmark with $\Omega_0 = 8.1 \times 10^{-5} \text{ s}^{-1}$, six cases are listed in Table IV with non-dimensional parameters quoted.

Figure 10 shows that with the decrease in the rotational velocity, asymmetry between upflows and downflows enhances and downflow lanes become more narrow and strong. For slowly rotating cases, the banana cells become more irregular, indicating a more turbulent state. The downflow lanes near the poles are generally more isotropic and possess smaller spatial scales while that near the equator is influenced

by the rotation with obvious longitudinal shear toward the rotational direction. Figure 11 shows that for the slowest rotating case, an intricate network of downflow lanes appears near the top boundary and the width of the downflow lanes is generally smaller in the upper convection zone and larger in the deep convection zone. By comparing with the fast rotating laminar solar benchmark, convective structures near the poles in the slowest rotating case are more complex. Figure 12 shows that with the rotational velocity decreasing to $\Omega_0 = 3.7 \times 10^{-5} \text{ s}^{-1}$, downflows become so strong that an inward kinetic energy flux indicated by negative F_e is witnessed and the inward F_e grows in the case of $\Omega_0 = 2.6 \times 10^{-5} \text{ s}^{-1}$. A strong inward kinetic energy flux must be compensated by an enhanced outward enthalpy flux,⁹ which is consistent with our results. To investigate differential rotation, we define longitudinally averaged angular velocity of the fluid around the z -axis

$$\Delta\Omega = \int_0^{2\pi} \frac{-yu + xv}{x^2 + y^2} d\theta, \quad \theta = \arctan \frac{y}{x}. \quad (43)$$

Figure 13 shows contours of non-dimensional angular velocity $\Delta\Omega/\Omega_0$ at $t = T_f$ for varied rotational velocity of the spherical shell. Although all current simulations predict a fast-rotating equator and slow-rotating poles, the predicted contrast of angular velocity is far from that in the Sun. For the Sun, $\Omega_0^{\text{Sun}} = 2.6 \times 10^{-6} \text{ s}^{-1}$, corresponding to $T_{\odot}^{\text{Sun}} \approx 28 T_{\odot}^{\text{Earth}}$. The rotational period of time for the Sun is approximately $25 T_{\odot}^{\text{Earth}}$ near the equator and $35 T_{\odot}^{\text{Earth}}$ near the poles. This corresponds to $\Delta\Omega/\Omega_0 \approx 0.12$ near the equator and -0.2 near the poles. Decreasing Ω_0 would enhance the contrast of angular velocity in general, and there is one exception. The amplitude of contrast does not change significantly for cases of $\Omega_0 = 2.6 \times 10^{-5}$ and $\Omega_0 = 3.7 \times 10^{-5} \text{ s}^{-1}$. To further enhance the differential rotation, decreasing the viscosity of the fluid and increasing the Rossby number can be choices.

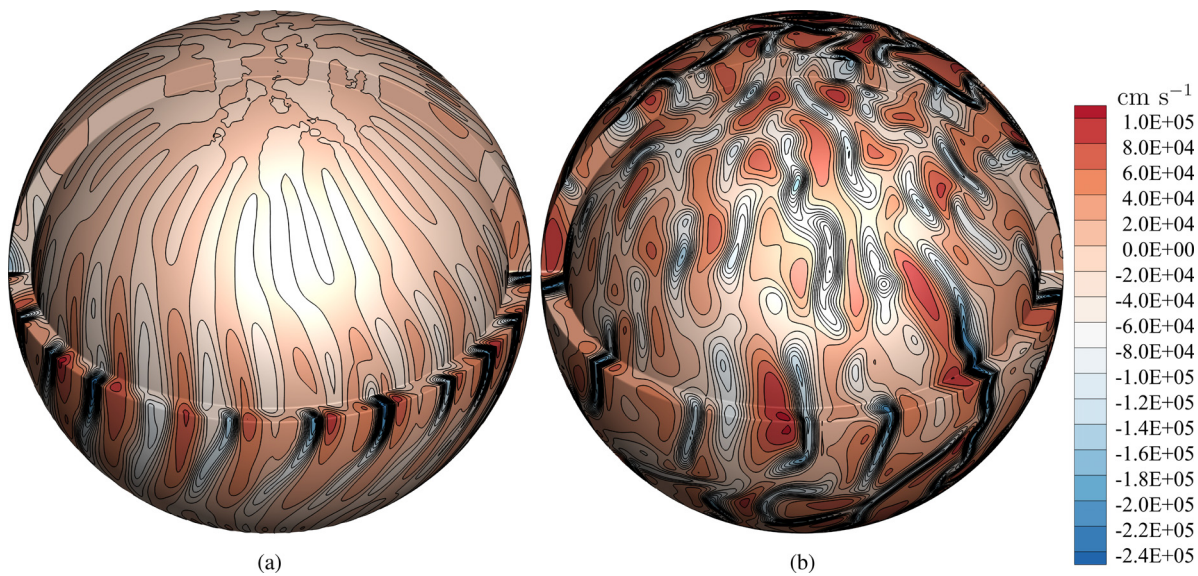


FIG. 11. 3D visualization of instantaneous radial velocity $U_r = \mathbf{U} \cdot \hat{\mathbf{r}}$ for the cases of (a) $\Omega_0 = 8.1 \times 10^{-5}$ and (b) $\Omega_0 = 2.6 \times 10^{-5} \text{ s}^{-1}$. The volume $0.868r_i < r < 0.95r_i$ is shown and the portion where $x > 0$ and $z > 0$ is removed to show the convection structures in the interior.

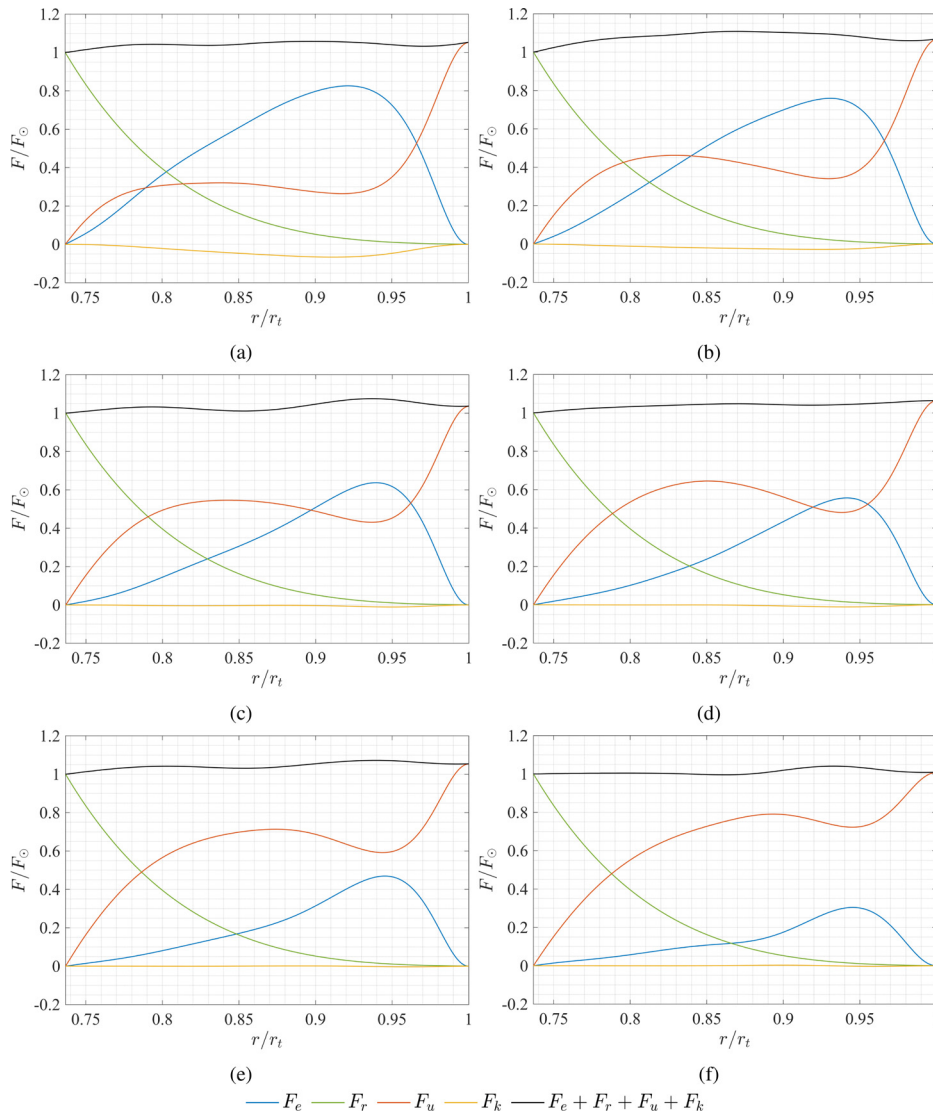


FIG. 12. Components of the normalized, horizontally integrated radial energy flux for the cases of (a) $\Omega_0 = 2.6 \times 10^{-5}$, (b) $\Omega_0 = 3.7 \times 10^{-5}$, (c) $\Omega_0 = 4.8 \times 10^{-5}$, (d) $\Omega_0 = 5.9 \times 10^{-5}$, (e) $\Omega_0 = 7.0 \times 10^{-5}$, and (f) $\Omega_0 = 8.1 \times 10^{-5} \text{ s}^{-1}$.

VIII. CONCLUSIONS AND DISCUSSIONS

CHORUS is the first stellar convection code that employs unstructured grids. In the current research, we further improve it to CHORUS++ for rotationally constrained convection simulations in spherical shells. Like CHORUS, CHORUS++ is based on fully compressible models, flexible on grids and suitable for massively parallel computing. However, compared with CHORUS which uses the isoparametric mapping, CHORUS++ embeds a boundary-conforming transfinite mapping into the spectral difference (SD) method on cubed-sphere grids, thus achieving exact representations of spherical surfaces on arbitrary sparse grids. This allows us to carry out high-fidelity simulations of a laminar solar benchmark using sixth-order elements on very coarse grids. Banana-shaped convective cells are well resolved without spurious oscillations. Given a relatively small number of solution degrees of freedom, using sixth-order elements produces smooth and converged predictions for components of the radial

energy flux while using third- and fourth-order elements fails. To give converged predictions, using sixth-order elements in CHORUS++ is 8.7 times faster than using third-order elements. Moreover, using high-order elements is shown to have enhanced speedup when the number of processors becomes very large. It takes only 1.5 weeks to run the global-scale laminar solar convection until equilibration of energy flux using just one Intel Xeon Gold 6148 Processor of 40 threads, which is not possible for the original CHORUS code even on supercomputers. This efficiency is even comparable to the anelastic spherical harmonic (ASH) code. To test the applicability of using sixth-order elements in CHORUS++ to cases with more complex convective patterns, CHORUS++ is applied to rotating convection simulations with decreased rotational velocity. Asymmetry between upflows and downflows enhances. Downflow lanes become narrow and strong, separating broad and weak upflow regions. Calculations of the energy flux are aligned with observations of strong downflow lanes:

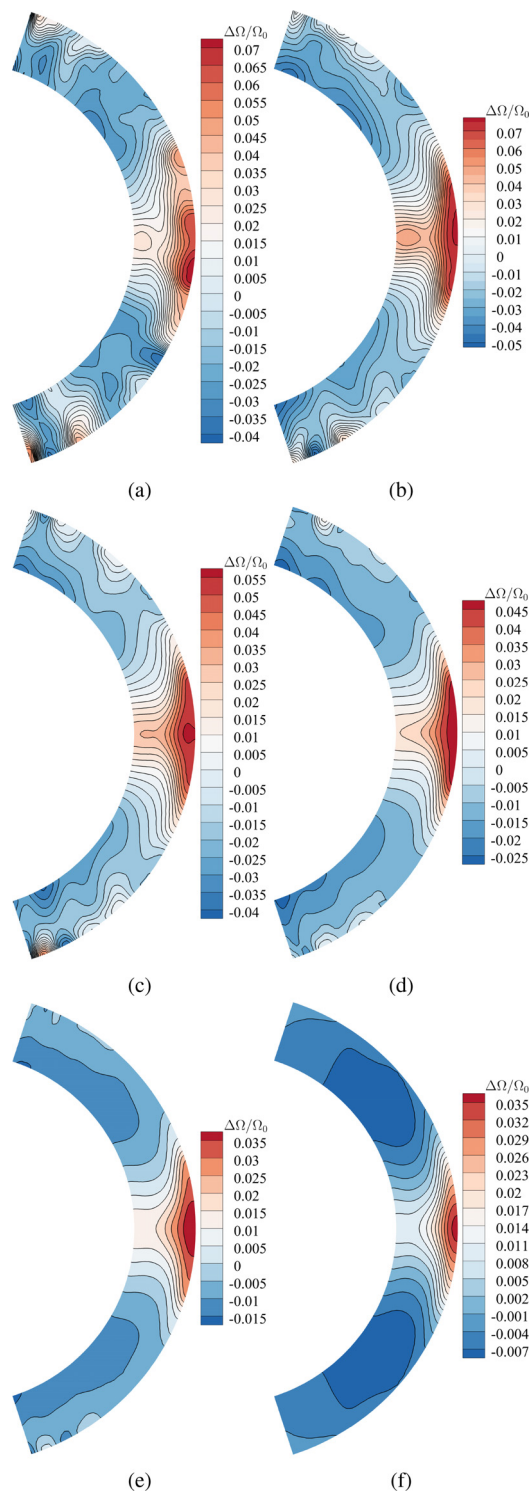


FIG. 13. Snapshots of differential rotation at $t = T_f$, which is expressed in terms of non-dimensional longitudinally averaged contrast of angular velocity around the z-axis. (a) $\Omega_0 = 2.6 \times 10^{-5}$, (b) $\Omega_0 = 3.7 \times 10^{-5}$, (c) $\Omega_0 = 4.8 \times 10^{-5}$, (d) $\Omega_0 = 5.9 \times 10^{-5}$, (e) $\Omega_0 = 7.0 \times 10^{-5}$, and (f) $\Omega_0 = 8.1 \times 10^{-5} \text{ s}^{-1}$.

a strong inward kinetic energy flux appears, and as a compensation, the outward enthalpy flux enhances. These observations are consistent with decreasing rotational constraints and other results from the literature.

In reality, the interiors of many stars, including the Sun, and gas giants, planets like Jupiter and Saturn, consist of ionized gases (plasmas), which behave as conducting fluids. To model motions of conducting fluids and ensuing evolution of magnetic fields inside these fluids, CHORUS++ needs to be extended to magnetohydrodynamic (MHD) simulations. To achieve this goal, a potential challenge is an efficient way to deal with the divergence-free constraint of the magnetic field on unstructured grids. An increase in the divergence error would lower the accuracy, affect the stability, and even induce unphysical behaviors of the MHD system.⁴⁶ Chen and Liang⁴⁷ showed that the divergence cleaning approach proposed by Derigs *et al.*⁴⁸ can be coupled with the SD method to control the divergence error on unstructured grids. With the assist of the artificial dissipation method, the resulting method can simulate both subsonic and supersonic MHD flows efficiently. It paves the way for simulations of high-Mach number solar dynamo, for example, dynamo in red giants and photosphere of the Sun.

ACKNOWLEDGMENTS

The research work of the first two authors has been supported by a National Science Foundation (NSF) CAREER Award (No. 1952554) and an Air Force Office of Scientific Research (AFOSR) grant (Award No. FA9550-20-1-0374) monitored by Dr. Fariba Fahroo. M. W. acknowledges the support from NSFC (Grant No. 12225204), Department of Science and Technology of Guangdong Province (Grant No. 2020B1212030001), and the Shenzhen Science and Technology Program (Grant No. KQTD20180411143441009). Numerical simulations have been supported by the ACRES cluster at Clarkson University and the Center for Computational Science and Engineering at Southern University of Science and Technology. The first author would like to acknowledge helpful discussions with Dr. Junfeng Wang at Simetrics Inc.

AUTHOR DECLARATIONS

Conflict of Interest

The authors have no conflicts to disclose.

Author Contributions

Kuangxu Chen: Conceptualization (equal); Formal analysis (lead); Investigation (lead); Validation (lead); Writing – original draft (lead). **Chunlei Liang:** Conceptualization (equal); Formal analysis (supporting); Funding acquisition (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal). **Minping Wan:** Formal analysis (supporting); Funding acquisition (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal).

DATA AVAILABILITY

The data that support the findings of this work are available from the corresponding author upon reasonable request.

REFERENCES

- ¹F. J. Robinson and K. L. Chan, "A large-eddy simulation of turbulent compressible convection: Differential rotation in the solar convection zone," *Mon. Not. R. Astron. Soc.* **321**, 723–732 (2001).
- ²P. Käpylä, "On global solar dynamo simulations," *Astron. Nachr.* **332**, 43–50 (2011).
- ³P. A. Gilman, "Nonlinear dynamics of Boussinesq convection in a deep rotating spherical shell-I," *Geophys. Astrophys. Fluid Dyn.* **8**, 93–135 (1977).
- ⁴G. A. Glatzmaier and P. A. Gilman, "Compressible convection in a rotating spherical shell. V-induced differential rotation and meridional circulation," *Astrophys. J.* **256**, 316–330 (1982).
- ⁵P. Gilman and J. Miller, "Nonlinear convection of a compressible fluid in a rotating spherical shell," *Astrophys. J. Suppl. Ser.* **61**, 585–608 (1986).
- ⁶M. S. Miesch, J. R. Elliott, J. Toomre, T. L. Clune, G. A. Glatzmaier, and P. A. Gilman, "Three-dimensional spherical simulations of solar convection. I. Differential rotation and pattern evolution achieved with laminar and turbulent states," *Astrophys. J.* **532**, 593 (2000).
- ⁷A. S. Brun and J. Toomre, "Turbulent convection under the influence of rotation: Sustaining a strong differential rotation," *Astrophys. J.* **570**, 865 (2002).
- ⁸A. S. Brun, M. S. Miesch, and J. Toomre, "Global-scale turbulent convection and magnetic dynamo action in the solar envelope," *Astrophys. J.* **614**, 1073 (2004).
- ⁹M. S. Miesch, A. S. Brun, M. L. DeRosa, and J. Toomre, "Structure and evolution of giant cells in global models of solar convection," *Astrophys. J.* **673**, 557 (2008).
- ¹⁰T. C. Clune, J. Elliott, M. Miesch, J. Toomre, and G. A. Glatzmaier, "Computational aspects of a code to study rotating turbulent convection in spherical shells," *Parallel Comput.* **25**, 361–380 (1999).
- ¹¹P. A. Gilman and G. A. Glatzmaier, "Compressible convection in a rotating spherical shell. I—Anelastic equations. II—A linear anelastic model. III—Analytic model for compressible vorticity waves," *Astrophys. J. Suppl. Ser.* **45**, 335–388 (1981).
- ¹²A. Brandenburg and W. Dobler, "Hydromagnetic turbulence in computer simulations," *Comput. Phys. Commun.* **147**, 471–475 (2002).
- ¹³W. Dobler, M. Stix, and A. Brandenburg, "Magnetic field generation in fully convective rotating spheres," *Astrophys. J.* **638**, 336 (2006).
- ¹⁴P. J. Käpylä, "Star-in-a-box simulations of fully convective stars," *Astron. Astrophys.* **651**, A66 (2021).
- ¹⁵P. J. Käpylä, M. J. Korpi, A. Brandenburg, D. Mitra, and R. Tavakol, "Convective dynamos in spherical wedge geometry," *Astron. Nachr.: Astron. Notes* **331**, 73–81 (2010).
- ¹⁶P. J. Käpylä, M. J. Mantere, and A. Brandenburg, "Cyclic magnetic activity due to turbulent convection in spherical wedge geometry," *Astrophys. J. Lett.* **755**, L22 (2012).
- ¹⁷J. Wang, C. Liang, and M. S. Miesch, "A compressible high-order unstructured spectral difference code for stratified convection in rotating spherical shells," *J. Comput. Phys.* **290**, 90–111 (2015).
- ¹⁸D. A. Kopriva and J. H. Kolas, "A conservative staggered-grid Chebyshev multidomain method for compressible flows," *J. Comput. Phys.* **125**, 244–261 (1996).
- ¹⁹D. A. Kopriva, "A staggered-grid multidomain spectral method for the compressible Navier-Stokes equations," *J. Comput. Phys.* **143**, 125–158 (1998).
- ²⁰Y. Liu, M. Vinokur, and Z. J. Wang, "Spectral difference method for unstructured grids I: Basic formulation," *J. Comput. Phys.* **216**, 780–801 (2006).
- ²¹Z. J. Wang, Y. Liu, G. May, and A. Jameson, "Spectral difference method for unstructured grids II: Extension to the Euler equations," *J. Sci. Comput.* **32**, 45–71 (2007).
- ²²H. T. Huynh, "A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods," AIAA Paper No. 2007-4079, 2007.
- ²³P. F. Fischer, J. W. Lottes, S. G. Kerkemeier *et al.*, see <https://nek5000.mcs.anl.gov> for "nek5000 web page" (2008).
- ²⁴S. Mandal, M. Ghosh, P. Maity, A. Banerjee, and P. Pal, "Supercritical and subcritical rotating convection in a horizontally periodic box with no-slip walls at the top and bottom," *Phys. Fluids* **34**, 104117 (2022).
- ²⁵P. Käpylä, M. Mantere, G. Guerrero, A. Brandenburg, and P. Chatterjee, "Reynolds stress and heat flux in spherical shell convection," *Astron. Astrophys.* **531**, A162 (2011).
- ²⁶J. Wang, M. S. Miesch, and C. Liang, "Convection in oblate solar-type stars," *Astrophys. J.* **830**, 45 (2016).
- ²⁷C. Jones, P. Boronski, A. Brun, G. Glatzmaier, T. Gastine, M. Miesch, and J. Wicht, "Anelastic convection-driven dynamo benchmarks," *Icarus* **216**, 120–135 (2011).
- ²⁸G. A. Glatzmaier, "Numerical simulations of stellar convective dynamos. I. The model and method," *J. Comput. Phys.* **55**, 461–484 (1984).
- ²⁹J. Sánchez Umbría and M. Net, "Bifurcations to quasiperiodicity of the torsional solutions of convection in rotating fluid spheres: Techniques and results," *Phys. Fluids* **34**, 114112 (2022).
- ³⁰C. Ronchi, R. Iacono, and P. S. Paolucci, "The 'cubed sphere': A new method for the solution of partial differential equations in spherical geometry," *J. Comput. Phys.* **124**, 93–114 (1996).
- ³¹M. Farraskhalvat and J. Miles, *Basic Structured Grid Generation: With an Introduction to Unstructured Grid Generation* (Elsevier, 2003).
- ³²V. V. Rusanov, "The calculation of the interaction of non-stationary shock waves and obstacles," *USSR Comput. Math. Math. Phys.* **1**, 304–320 (1962).
- ³³F. Bassi and S. Rebay, "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier-Stokes equations," *J. Comput. Phys.* **131**, 267–279 (1997).
- ³⁴Y. Sun, Z. J. Wang, and Y. Liu, "High-order multidomain spectral difference method for the Navier-Stokes equations," AIAA Paper No. 2006-301, 2006.
- ³⁵S. Ruuth, "Global optimization of explicit strong-stability-preserving Runge-Kutta methods," *Math. Comput.* **75**, 183–207 (2005).
- ³⁶G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.* **20**, 359–392 (1998).
- ³⁷B. O'Mara, M. S. Miesch, N. A. Featherstone, and K. C. Augustson, "Velocity amplitudes in global convection simulations: The role of the Prandtl number and near-surface driving," *Adv. Space Res.* **58**, 1475–1489 (2016).
- ³⁸B. B. Karak, M. Miesch, and Y. Bekki, "Consequences of high effective Prandtl number on solar differential rotation and convective velocity," *Phys. Fluids* **30**, 046602 (2018).
- ³⁹J. Schumacher and K. R. Sreenivasan, "Colloquium: Unusual dynamics of convection in the Sun," *Rev. Mod. Phys.* **92**, 041001 (2020).
- ⁴⁰H. Hotta and K. Kusano, "Solar differential rotation reproduced with high-resolution simulation," *Nat. Astron.* **5**, 1100–1102 (2021).
- ⁴¹C. A. Jones, M. J. Thompson, and S. M. Tobias, "The solar dynamo," *Space Sci. Rev.* **152**, 591–616 (2010).
- ⁴²J. S. Hesthaven and T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications* (Springer Science & Business Media, 2007).
- ⁴³G. M. Vasil, K. Julien, and N. A. Featherstone, "Rotation suppresses giant-scale solar convection," *Proc. Natl. Acad. Sci.* **118**, e2022518118 (2021).
- ⁴⁴E. M. King and J. M. Aurnou, "Turbulent convection in liquid metal with and without rotation," *Proc. Natl. Acad. Sci.* **110**, 6688–6693 (2013).
- ⁴⁵T. Li, M. Wan, and S. Chen, "Flow structures, nonlinear inertial waves and energy transfer in rotating spheres," *Theor. Appl. Mech. Lett.* **11**, 100312 (2021).
- ⁴⁶J. U. Brackbill and D. C. Barnes, "The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations," *J. Comput. Phys.* **35**, 426–430 (1980).
- ⁴⁷K. Chen and C. Liang, "An arbitrarily high-order spectral difference method with divergence cleaning (SDDC) for compressible magnetohydrodynamic simulations on unstructured grids," *Astrophys. J.* **932**, 16 (2022).
- ⁴⁸D. Derigs, A. R. Winters, G. J. Gassner, S. Walch, and M. Böhm, "Ideal GLM-MHD: About the entropy consistent nine-wave magnetic field divergence diminishing ideal magnetohydrodynamics equations," *J. Comput. Phys.* **364**, 420–467 (2018).