RESEARCH ARTICLE
SOLID EARTH: GEODYNAMICS

Earth and Planetary Physics
8: 549–563, 2024
doi: 10.26464/epp2024021

Implementation of a particle-in-cell method for the energy solver in 3D spherical geodynamic modeling

Hao Dong[1,2,3], ZeBin Cao[2], LiJun Liu[2,4*], YanChong Li[4], SanZhong Li[1,3], LiMing Dai[1,3], and XinYu Li[5]

[1]Frontiers Science Center for Deep Ocean Multispheres and Earth System, Key Laboratory of Submarine Geosciences and Prospecting Techniques, Ministry of Education and College of Marine Geosciences, Ocean University of China, Qingdao 266100, China;

[2]State Key Laboratory of Lithospheric Evolution, Institute of Geology and Geophysics, Chinese Academy of Sciences, Beijing 100029, China;

[3]Laboratory for Marine Geology, Laoshan Laboratory, Qingdao 266237, China;

[4]Department of Earth Science & Environmental Change, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA;

[5]Laboratory of Seismology and Physics of Earth's Interior, School of Earth and Space Sciences, University of Science and Technology of China, Hefei 230026, China

Key Points:

- Traditional temperature solvers are usually prone to numerical oscillations that hinder geodynamic interpretation.
- We implement a particle-in-cell method for the energy solver to improve the temperature solution in the finite element code CitcomS with data assimilation.
- This new method generates accurate solutions in spherical coordinates with distributed memory and good parallel efficiency.

Citation: Dong, H., Cao, Z. B., Liu, L. J., Li, Y. C., Li, S. Z., Dai, L. M., and Li, X. Y. (2024). Implementation of a particle-in-cell method for the energy solver in 3D spherical geodynamic modeling. *Earth Planet. Phys.*, *8*(3), 549–563. http://doi.org/10.26464/epp2024021

Abstract: The thermal evolution of the Earth's interior and its dynamic effects are the focus of Earth sciences. However, the commonly adopted grid-based temperature solver is usually prone to numerical oscillations, especially in the presence of sharp thermal gradients, such as when modeling subducting slabs and rising plumes. This phenomenon prohibits the correct representation of thermal evolution and may cause incorrect implications of geodynamic processes. After examining several approaches for removing these numerical oscillations, we show that the Lagrangian method provides an ideal way to solve this problem. In this study, we propose a particle-in-cell method as a strategy for improving the solution to the energy equation and demonstrate its effectiveness in both one-dimensional and three-dimensional thermal problems, as well as in a global spherical simulation with data assimilation. We have implemented this method in the open-source finite-element code CitcomS, which features a spherical coordinate system, distributed memory parallel computing, and data assimilation algorithms.

Keywords: numerical oscillation; overshooting and undershooting; particle-in-cell method; three-dimensional spherical geodynamic modeling; energy solver; finite element method

1. Introduction

The Earth's internal convection, to the first order, is a thermal problem that concerns the behavior of cold downwelling and hot upwelling. In addition, the temperature field of the Earth's interior controls several key mechanical properties, such as density, viscosity, and phase transformation. Therefore, it is important to accurately compute the distribution and evolution of the Earth's thermal structure. However, a long-standing problem in traditional numerical models is that oscillations in the temperature field (overshooting and undershooting) tend to occur in the temporal behavior of the advection–diffusion process (Fletcher, 1988; Lenardic and Kaula, 1993; Samuel and Evonuk, 2010; Zhong SJ et al., 2015). These phenomena are also referred to as numerical dispersion, which is believed to be caused by odd-order derivatives in the governing equation (Fletcher, 1988). In the case of simulating the Earth's thermal evolution, this numerical error is caused by the directional advection process.

In practice, this numerical problem tends to occur at locations where the velocity is parallel to temperature gradients, especially when the latter are sharp (Figure 1a). Because major geodynamic processes, such as subducting slabs and rising plumes, usually possess sharp thermal gradients with large velocities, this numerical oscillation represents a common artifact in such models (Figure 1b). More important, the resulting low- and high-temperature features from these oscillations often grow over time and may severely contaminate the true thermal structure and mantle dynamics (e.g., Lenardic and Kaula, 1993). Consequently, this may

First author: H. Dong, donghao10000@stu.ouc.edu.cn
Correspondence to: L. J. Liu, ljliu@mail.iggcas.ac.cn
Received 07 DEC 2023; Accepted 06 MAR 2024.
First Published online 22 APR 2024.

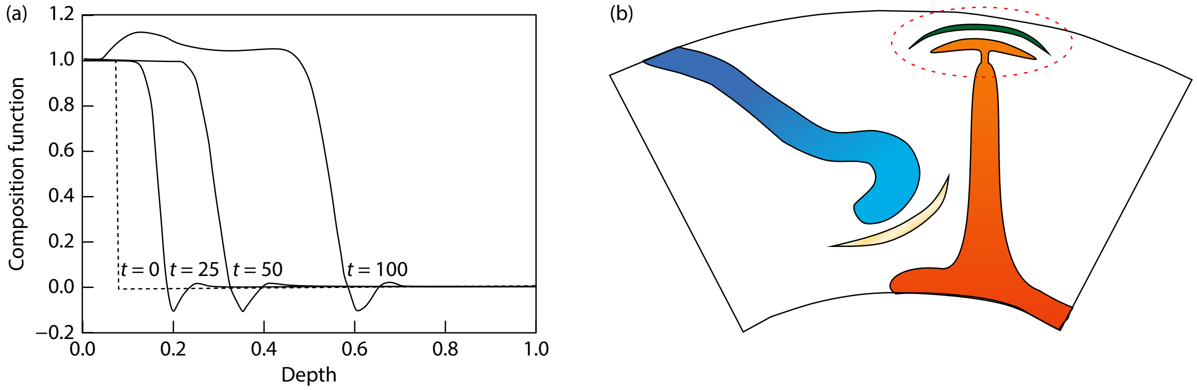

Figure 1. Numerical oscillation in advection–diffusion problems. (a) Numerical dispersion during the advection of a 1D step function (based on Lenardic and Kaula, 1993). (b) Illustration of temperature artifacts caused by high-temperature overshooting (yellow) below a subducting slab (blue) and low-temperature overshooting (green) above a mantle plume (orange).

lead to misrepresentation of temperature-related geological processes. For example, the artificial high-temperature anomalies generated around a sinking slab may accumulate and rise to the base of nearby continents and may form illusionary heat sources for intraplate thermal events (Figure 1b). For plume models, the region above the plume head may be prone to artificial cooling, which affects the thermal budget and motion of the plume (Figure 1b).

Proposed methods for bypassing or depressing such numerical oscillations include (1) an interface tracking method that initializes particles near the material interface and then tracks the interface over time (e.g., Christensen and Yuen, 1984; van Keken et al., 1997; Zhong SJ and Hager, 2003; Samuel and Evonuk, 2010); (2) a numerical filter that truncates the temperatures below and above given thresholds while preserving the total enthalpy of the system (e.g., Hansen and Yuen, 1988; Lenardic and Kaula, 1993); and (3) a Lagrangian method that utilizes particles from the entire solution region to achieve advection (e.g., Weinberg and Schmeling, 1992; Tackley, 1998; Gerya and Yuen, 2003; Tackley and King, 2003; McNamara and Zhong SJ, 2004). The interface tracking method mainly focuses on identifying the interface between two compositional components during their temporal evolution (e.g., Christensen and Yuen, 1984; van Keken et al., 1997; Zhong SJ and Hager, 2003; Samuel and Evonuk, 2010). This method utilizes tracer particles that represent different materials to determine and track the location of the interface over time. However, because of its incapability of reproducing the thermal diffusion process, this method cannot be applied to solving the Earth-like temperature evolution.

A numerical filter (such as the Lenardic filter) is commonly used in suppressing overshooting and undershooting because it is easy to implement and computationally inexpensive (e.g., Hansen and Yuen, 1988; Lenardic and Kaula, 1993). However, this method creates sharp thermal gradients during temperature truncation that may cause new oscillations. In addition, it works only when the artificial temperatures are beyond the maximum and minimum allowable values of the model, meaning that artifacts with values within these bounds (like those in Figure 1b with both slabs and plumes present) cannot be corrected. This problem has become urgent since recent models, especially those with data assimilation, have begun to consider both cold (e.g., slabs) and hot (e.g., plumes) mantle features, to replicate the real Earth (e.g., Schuberth et al., 2009; Gerya et al., 2015; Leonard and Liu LJ, 2016; Cao XZ et al., 2021; Flament et al., 2022; Pal and Ghosh, 2023). In these cases, because the slabs and plumes represent extreme temperature values, the resulting artifacts usually start near the temperature of the ambient mantle, so the filter cannot eliminate these artifacts.

Compared with the other methods, the Lagrangian method is more attractive because of its shock-capture capability (e.g., Weinberg and Schmeling, 1992; Tackley, 1998; Gerya and Yuen, 2003; Tackley and King, 2003; McNamara and Zhong SJ, 2004). However, this method is computationally expensive because of the large number of particles required to ensure its effectiveness. For some recent global models (e.g., Wu BC et al., 2022; Peng DD and Liu LJ, 2023), the number of three-dimensional (3D) cells or elements can reach ~10^8, the number of particles is generally greater than 10^9, and the computational cost is approximately 10^5–10^6 core hours for a 300 Myr simulation. Moreover, existing treatments for the energy equation mainly focus on finite difference (FD) models in Cartesian coordinates, such as I2/3ELVIS (Gerya and Yuen, 2003; Gerya et al., 2015) and LaMEM (Kaus et al., 2016), as well as a two-dimensional (2D) Cartesian-coordinate code with a finite element method (FEM) — MVEP2. However, codes with spherical coordinates, such as CitcomS (Zhong SJ et al., 2000; McNamara and Zhong SJ, 2004; Tan E et al., 2006) and StagYY (Tackley, 2008), still use a Lenardic filter to handle these artifacts in the solution of the energy equation, and some, such as ASPECT (Gassmöller et al., 2019), use the Lagrangian method to solve only nondiffusion problems.

Therefore, it is necessary to implement the Lagrangian method in 3D global FEM codes that support parallel computing in distributed memory clusters and to fully examine the effectiveness of this method. In addition, data assimilation methods have been proven to be very effective in recent years for replicating realistic Earth dynamic processes (e.g., Schuberth et al., 2009; Hu JS et al., 2016; Peng DD and Liu LJ, 2023), so it is necessary to adopt the Lagrangian methods for the data assimilation models as well.

In this study, we present a particle-in-cell method for the energy solver (PICES) that satisfactorily resolves the overshooting and undershooting issues. We demonstrate this by using a one-dimensional (1D) thermal model and eventually implementing the method in the 3D global or regional geodynamic models with data assimilation based on the code CitcomS.

2. Strategies to Resolve the Numerical Oscillation Problem

2.1 Problems in the Solution to the Energy Equation

The governing equation for the conservation of energy is

$$\rho c_P \frac{\partial T}{\partial t} = \underbrace{\rho c_P \kappa \Delta T}_{\text{diffusion}} - \underbrace{\rho c_P \boldsymbol{u} \cdot \nabla T}_{\text{advection}} + \underbrace{\rho \left(H_A + H_S + H_L + H_R \right)}_{\text{internal heat production}}, \tag{1}$$

where ρ is density, c_P is specific heat capacity at a constant pressure, T is temperature, t is time, κ is thermal diffusivity, $\boldsymbol{u}$ is velocity, H_A is the adiabatic heat production rate, H_S is the shear heat production rate, H_L is the latent heat production rate, and H_R is the radiative heat production rate.

Two common numerical schemes can be used to solve the advection–diffusion equation above. The first scheme is the Eulerian method:

$$T_{s+1} = T_s + \left(t_{s+1} - t_{s-1} \right) \times \left(\kappa \Delta T_s - \boldsymbol{u} \cdot \nabla T_s + \frac{H_A + H_S + H_L + H_R}{c_P} \right). \tag{2}$$

The second scheme is the modified-Eulerian method (also known as the predictor–corrector method), as adopted in the code CitcomS (Zhong SJ et al., 2000):

$$T_{s+1,c} = T_s + \left(t_{s+1} - t_{s-1} \right) \times \left(\frac{\kappa \Delta T_s - \boldsymbol{u} \cdot \nabla T_s + \kappa \Delta T_{s+1,p} - \boldsymbol{u} \cdot \nabla T_{s+1,p}}{2} + \frac{H_A + H_S + H_L + H_R}{c_P} \right), \tag{3}$$

where T_s is the temperature of time step s, and the subscripts p and c represent the predicted and corrected temperatures, respectively.

As is shown later, both of the numerical schemes above tend to generate oscillatory solutions. A commonly adopted method for reducing numerical oscillation is the Lenardic filter. This method can be divided into two steps (Lenardic and Kaula, 1993). The first step is to truncate the temperature beyond the maximum and minimum allowable values:

$$T_{n,f} = \begin{cases} T_{\min}, & T_{n,u} \in \left(-\infty, T_{\min} + 2 \times \left| T_{\min} - \min\left(T_u \right) \right| \right), \\ T_{n,u}, & T_{n,u} \in \left[T_{\min} + 2 \times \left| T_{\min} - \min\left(T_u \right) \right|, T_{\max} - 2 \times \left| T_{\max} - \max T_u \right| \right], \\ T_{\max}, & T_{n,u} \in \left(T_{\max} - 2 \times \left| T_{\max} - \max\left(T_u \right) \right|, +\infty \right). \end{cases} \tag{4}$$

The second step is to update the entire temperature field to maintain the conservation of energy:

$$T_{n,\mathrm{mod}} = T_{n,f} + \frac{\sum_{n=1}^{N} \left(\rho c_P T_{n,o} \right) - \sum_{n=1}^{N} \left(\rho c_P T_{n,f} \right)}{\sum_{n=1}^{N} \left(\rho c_P \right)}, \tag{5}$$

where T_n is the filtered temperature of node n, N is the total number of nodes, and the subscripts o, f, and mod represent the unfiltered temperature, the filtered temperature, and the modified temperature, respectively. Additionally, $T_{\min}$ is the minimum temperature set by users and usually equals T_0, whereas $T_{\max}$ is the maximum temperature set by users and usually equals $T_0 + T_{\mathrm{range}}$, with T_{range} being an allowable temperature range.

In many cases, the Lenardic filter can control the numerical oscillations caused by overshooting and undershooting. However, this is not true for models that include both subducting slabs and mantle plumes because the filtering effect of this method takes place only for temperatures exceeding the system bounds, but the oscillations happen around the middle of the temperature range and are thus not detected by the filter. This is an inherent disadvantage of the filter method.

2.2 Particle-in-Cell Method for the Energy Solver in 3D Spherical Geodynamic Modeling

On the basis of the analysis above, we present a particle-in-cell strategy for the energy solver and implement it in the 3D spherical FEM code CitcomS.

2.2.1 Numerical Scheme of PICES

According to previous studies dealing with the instability problem in the energy evolution based on the FD method (Gerya and Yuen, 2003; Gerya et al., 2015; Kaus et al., 2016), separating the advection term from the energy equation is a preferred strategy:

$$\rho c_P \frac{\partial T}{\partial t} = \underbrace{\rho c_P \kappa \Delta T}_{\text{diffusion}} + \underbrace{\rho \left(H_A + H_S + H_L + H_R \right)}_{\text{internal heat production}}. \tag{6}$$

The advection term needs to be expressed and solved in a Lagrangian form by particle advection. Particles carrying temperature gradually move over time, resulting in an advection effect, thus avoiding solving advection on the grid. In this way, the advection–diffusion equation is simplified to a diffusion equation.

As a result, the solution of the diffusion equation of energy is expressed as

$$T_{s+1} = T_{\mathrm{interpolation}} + \left(t_{s+1} - t_s \right) \times \left(\kappa \Delta T_s + \left(H_A + H_S + H_L + H_R \right) / c_P \right), \tag{7}$$

where the subscript interpolation represents the interpolation of particles to elements.

The change in the temperature of one particle consists of two parts. One part is the temperature change caused by subgrid diffusion from all particles near a node so as to converge to a surrounding temperature, and the other part is the change due to thermal diffusion between elements. Thus, the temperature update scheme for particles is

$$T_{s+1} = T_s + \delta T^{\mathrm{subgrid}} + \delta T^{\mathrm{remain}}. \tag{8}$$

For the first part, the process of convergence of the particle temperature to the surrounding temperature is the relaxation process of the particle temperature. The first part is also called subgrid diffusion (Gerya and Yuen, 2003). We call the time required for the particle temperature to reach that surrounding temperature the relaxation time. This is the time it takes for the system to reach equilibrium from a nonequilibrium state.

On the basis of the definition of κ, the relaxation time is generally considered to be

$$t^{\text{feature}} = \frac{L_e^2}{\kappa}, \tag{9}$$

where L_e is the length of the element, and κ is the thermal diffusion coefficient of particle properties.

According to Newton's law of cooling,

$$\frac{\mathrm{d}T}{\mathrm{d}t} = \frac{T_{\text{environment}} - T}{t^{\text{feature}}}, \tag{10}$$

where the rate of change of the temperature of an object is equal to the ratio of the temperature difference between the environmental and actual temperatures to the relaxation time.

Integrating both sides of the differential equation above and using current particle (T_s) and surrounding ($T_s^{\text{environment}}$) temperatures, we can obtain the subgrid diffusion term of particles:

$$\delta T^{\text{subgrid}} = \left(T_s^{\text{environment}} - T_s\right)\left(1 - \exp\left(-\left(t_{s+1} - t_s\right)/t^{\text{feature}}\right)\right), \tag{11}$$

where exp represents an exponential function, which describes the temperature change of a particle that follows an exponential decay when it has a thermal discrepancy relative to the environment.

The second part depends on the environmental temperature change caused by thermal diffusion between elements. We first solve the total thermal diffusion on the grid, and then remove the subgrid diffusion, as shown in the first part, to obtain the change in environmental temperature:

$$\delta T^{\text{remain}} = \left(T_{S+1} - \delta T^{\text{subgrid}}_{\text{inverse}}\right)_{\text{interpolation}}, \tag{12}$$

where the subscript inverse represents the interpolation of elements to particles.

During model simulation, the initial temperature of particles can be directly assigned a dimensionless temperature. Subsequently, evolving boundary conditions, especially with data assimilation, need to be considered when solving the evolution of temperature. In this case, the temperature of particles must be updated when interpolated to nodes. In addition, the temperature variation δT in Equation (8) and the rate of temperature change $\dot{T}$ need to be recalculated before calculating the particle temperature.

2.2.2 Two Interpolation Options

To interpolate the temperature from particles to nodes (Figure 2a), the general form of the interpolation is

$$T_n^{\text{interpolation}} = \left(\sum\nolimits_{m=1}^{M} T_m w_m\right) \Big/ \left(\sum\nolimits_{m=1}^{M} w_m\right), \tag{13}$$

where m is the index of particles within a given element, M is the number of particles within the element, and w is the weight of the particles.

To interpolate the temperature from nodes to particles (Figure 2b), the general form of the interpolation is

$$T_m^{\text{inverse}} = \left(\sum\nolimits_{n=1}^{N} T_n w_n\right) \Big/ \left(\sum\nolimits_{n=1}^{N} w_n\right), \tag{14}$$

where w_n is the weight of the nodes.

Two options exist for the interpolation weights. For spherical coordinate systems, the first option is distance interpolation:

$$\begin{aligned} w_m &= 1/\sqrt{(x_n - x_m)^2 + (y_n - y_m)^2 + (z_n - z_m)^2} \\ &= 1/\sqrt{r_n^2 + r_m^2 - 2r_n r_m\left[\sin\theta_n \sin\theta_m \cos\left(\phi_n - \phi_m\right) + \cos\theta_n \cos\theta_m\right]}. \end{aligned} \tag{15}$$

Similarly, the interpolation weight for spherical distance is

$$w_n = w_m. \tag{16}$$

The second option is specifically designed for the FEM, where the shape function is used as the interpolation weight. Because the shape function is defined as

$$T^{\text{inverse}} = [N_e][T_e], \tag{17}$$

where

$$[N_e] = [\omega_{e1}, \omega_{e2}, \cdots, \omega_{eK}], \tag{18}$$

and

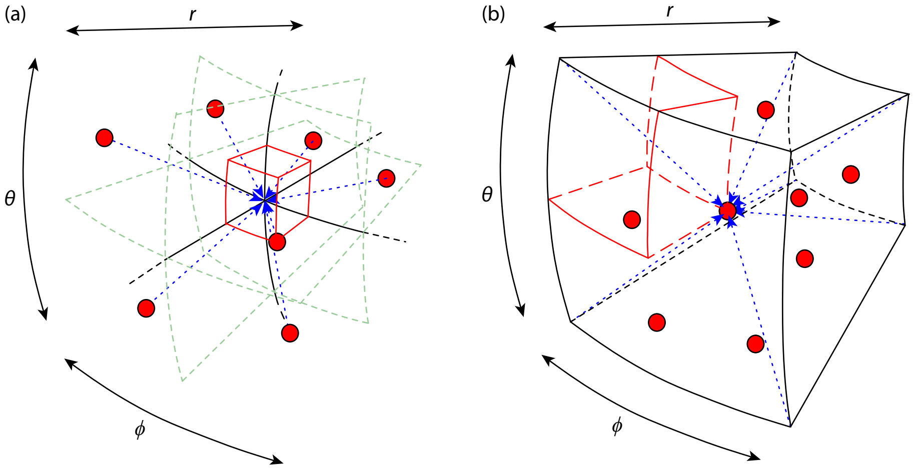

Figure 2. Cartoon diagram of interpolation in the spherical coordinate system: (a) Temperature interpolation from particles to nodes; and (b) temperature interpolation from nodes to particles. The red circles represent particles, and the solid red lines indicate the position of the particles. The blue dashed arrows indicate the direction of interpolation. The black lines represent the actual boundaries of elements, whereas the green lines indicate that the element is not fully represented.

$$[T_e] = [T_{e1}, T_{e2}, \cdots, T_{eK}]^{\mathrm{T}}, \tag{19}$$

where the subscript *e* represents a finite element, and *K* represents the number of nodes contained in each element. The $[N]$ is a matrix composed of the shape function ω_e. Therefore, the shape function of a node can be used as a weight for interpolation from particles to the node:

$$w_m = \omega_{ek},\ k \in [1, K]. \tag{20}$$

2.2.3 Implementation in CitcomS

In the initial state, the particle temperature and mesh temperature should be the same. In practice, we determine the temperature of elements and then interpolate that temperature to the particles. After initializing the temperature of particles and cells, the main algorithmic steps can be summarized in two parts: the cell part and the particle part. The algorithmic flowchart of the new particle-in-cell energy solver is shown in Figure 3.

The cell part is mainly responsible for calculating the diffusion equation and handling the boundary and time step conditions:
(1) Interpolate the temperature from particles to cells by using the interpolation algorithm in the previous section;
(2) To ensure consistency across different processers, average the temperatures and weights at these cells via message passing interface (MPI) communication;
(3) Apply data assimilation conditions to the boundary cells;
(4) Solve the diffusion equation by using the Eulerian method;
(5) Verify the boundary conditions;
(6) Verify the time step constraints;
(7) Calculate the solved temperature change due to diffusion.

The particle part is mainly responsible for updating the particle temperature, including the subgrid temperature changes and the environmental temperature changes. The main steps are as follows:
(1) Interpolate the solved temperature from cells to particles;
(2) Calculate the rate of change in temperature at the subgrid scale;
(3) Interpolate the subgrid scale temperature rate into the cells;
(4) Average the subgrid scale temperature rate of cells across processers by using MPI communication;
(5) Calculate the environmental temperature change rate;
(6) Interpolate the environmental temperature change rate back to the particles by using the shape function;
(7) Obtain the final temperature of particles.

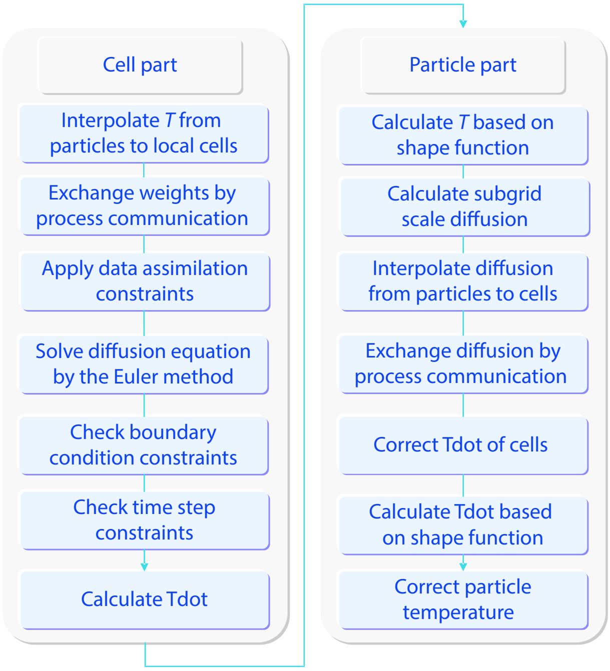
Cell part
Interpolate *T* from particles to local cells
Exchange weights by process communication
Apply data assimilation constraints
Solve diffusion equation by the Euler method
Check boundary condition constraints
Check time step constraints
Calculate Tdot
Particle part
Calculate *T* based on shape function
Calculate subgrid scale diffusion
Interpolate diffusion from particles to cells
Exchange diffusion by process communication
Correct Tdot of cells
Calculate Tdot based on shape function
Correct particle temperature

Figure 3. Algorithm flowchart of the new energy solver. The left panel is mainly for the cell part, whereas the right panel is mainly for the particle part.

The large number of interpolation operations can be accelerated by parallel computing. At this point, the weight of the local boundaries of each process requires the calculation of its average value through process communication. The temperature and weights of the nodes have the same topological structure, so boundary temperature exchangers can also be used to exchange weights. For the particle part, the particle temperature can be directly included in the extra properties of tracers (McNamara and Zhong SJ, 2004).

3. Results

3.1 A 1D Advection–Diffusion Problem

Here we demonstrate the effectiveness of the PICES by using a 1D advection–diffusion problem solved on an FEM mesh with three numerical approaches: the modified-Eulerian method, the Eulerian method, and the PICES method (Figure 4). The mesh domain is equally divided into 100 elements. In the PICES model, each element further contains 3 tracers. The initial temperature of the models is represented as a step function, and the advection process is expressed as a constant leftward velocity. The initial temperature of the mesh domain is set to 100 K, except for the temperature near the right boundary, which is set to −100 K. The results are obtained after the step function moves from the right boundary to the center of the mesh domain.

The modified-Eulerian (Figure 4a) result shows obvious overshooting or undershooting near sharp temperature gradients, and the Eulerian (Figure 4b) result shows inaccurate diffusion. In comparison, the PICES result (Figure 4c) is the most consistent with the exact (analytical) solution, and both overshooting and undershooting are suppressed to negligible ranges relative to those from the other two methods (Figure 4d). In this case, the PICES method clearly achieves better results than the other methods.

3.2 Simulation of a 3D Spherical Cold Downwelling

The second test is a regional model with a cold blob in a spherical coordinate system. The model is divided into 256 × 256 × 112 elements. In the PICES model, each element contains an average of 27 tracers. A cold blob falls from right below the surface to the core–mantle boundary while being internally stretched. Initially, the background temperature is set to 0.7 and the temperature of the cold blob is set to 0. The models evolve for 20 Myr as the blob reaches the core–mantle boundary. Theoretically, without internal heating, the temperature range of the model should always be

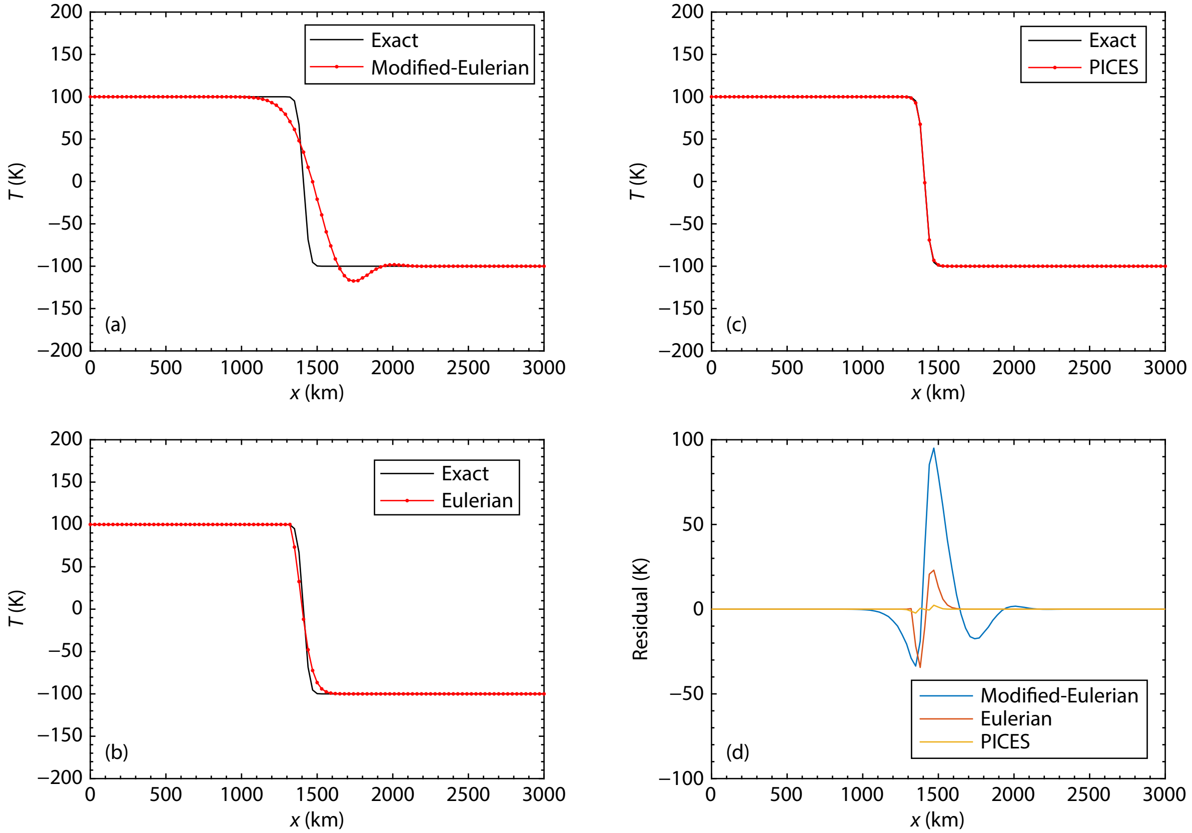

Figure 4. Comparison of the temperature solutions for a 1D step-function model: (a) the modified-Eulerian method; (b) the Eulerian method; (c) the PICES method; and (d) the residual of results from these three methods relative to the analytical solution.

0–0.7. A temperature higher than this range should be a high-temperature artifact, whereas a value lower than this range is a low-temperature artifact. Figure 5 shows the evolution of the thermal structure by using the same methods as in Section 3.1, and Figure 6 highlights the configuration of their respective thermal artifacts due to overshooting. Figure 7a shows the 1D temperature profiles of the final state through the center of the cold blob, and Figure 7b shows the evolution of the highest temperature of the models over time.

In the modified-Eulerian model, the cold blob gradually evolves into a pear shape because of the competition between negative buoyancy and viscous drag; during the evolution, the surface temperature of the blob remains sharp and clear (Figure 5a). However, high-temperature oscillations (T > 0.7) appeared below and around the blob while the amplitude of these oscillations increases and the oscillations spread over time (Figure 6a). A low-temperature overshooting feature also occurs inside the blob (Figure 6a), gradually cooling the interior of the blob, contradictory to reality. It is worth noting that the low-temperature region in Figure 6a is bell-shaped in the 3D view, with a hollow interior. Therefore, the anomalous region in Figure 6a appears broader than the 1D curve in Figure 7a.

In the Eulerian model, the thermal diffusion is more prominent than in the other cases, so its temperature profile is notably blurred (Figure 5b). In the 1D thermal profile (Figure 7a), the temperature range of the Eulerian model is smaller than that in the other two models. In this model, the temperature artifacts appear mainly behind the sinking blob, and their amplitude is smaller than that in the modified-Eulerian model (Figure 6).

In comparison, the PICES model also has a sharp surface (Figure 5c), similar to the modified-Eulerian model. The number of temperature artifacts is the least among all three models, and they appear mainly at the top of the model; no low-temperature artifacts appear (Figure 6c). Regarding the maximum temperature, the amount of deviation from the background temperature (0.7) in the PICES model is small enough to be considered negligible compared with other models (Figure 7b).

3.3 Simulation of a 3D Spherical Hot Plume

The model setup for the hot plume cases is the same as that for the cold blobs. Initially, the background temperature is 0.7 and the hot blob temperature is 0.9. Theoretically, the temperature range should always stay within the range of 0.7 to 0.9. A temperature higher than this range is a high-temperature artifact, whereas a temperature lower than this range is a low-temperature artifact. Figure 8 shows the evolving thermal structure of model results based on the three methods, and Figure 9 shows their associated thermal artifact. Figure 7c shows the 1D temperature profile at the final state through the center of the hot plume, and Figure 7d shows the evolution of the highest temperature over time for the plume model.

The numerical results for the hot plume cases are similar to those for the cold blob cases. With the modified-Eulerian method, the plume gradually upwells and exhibits a mushroom shape (Figure

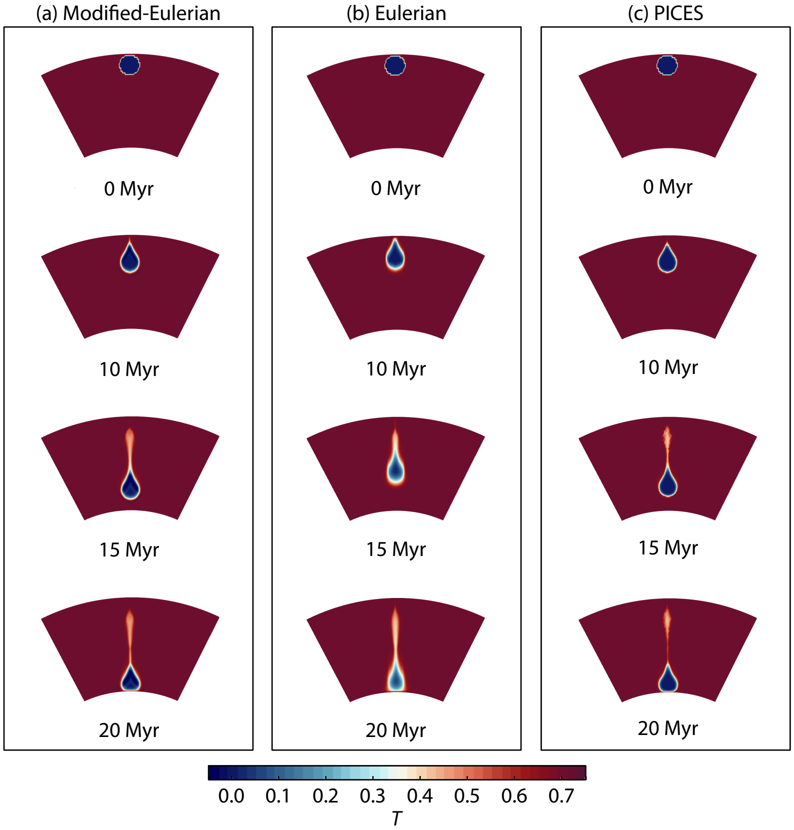

Figure 5. Evolution of a cold blob based on three methods: (a) the modified-Eulerian method; (b) the Eulerian method; and (c) the PICES method.

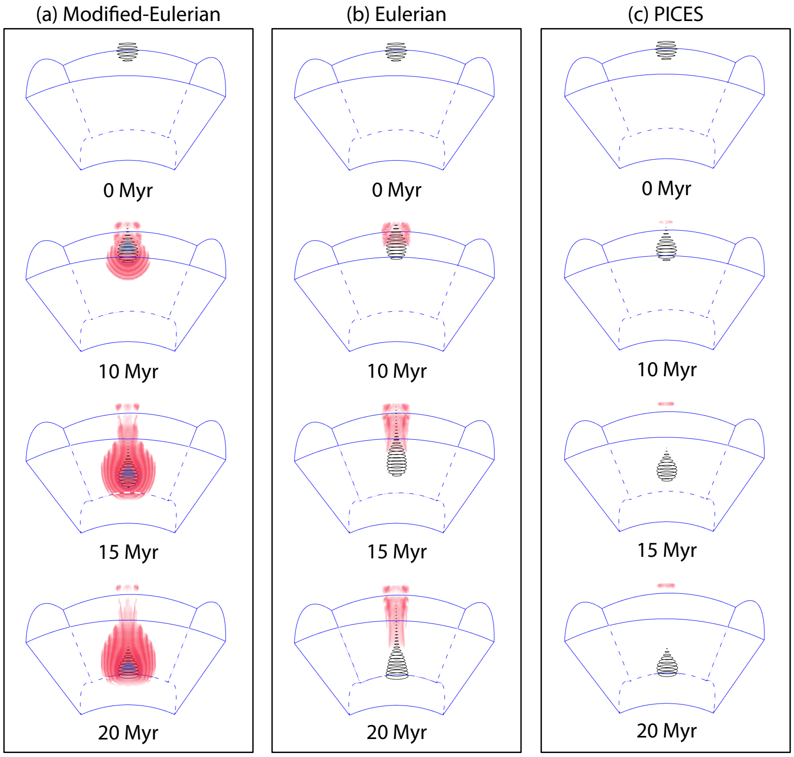

Figure 6. Comparison of the temperature artifacts associated with the sinking cold blob from three methods: (a) the modified-Eulerian method; (b) the Eulerian method; and (c) the PICES method. The blue lines outline the modeling domains. The black lines outline the low-temperature blob ($T < 0.6$). The red volumes represent high-temperature artifacts ($T > 0.7$). The blue volumes represent low-temperature artifacts ($T < 0$).

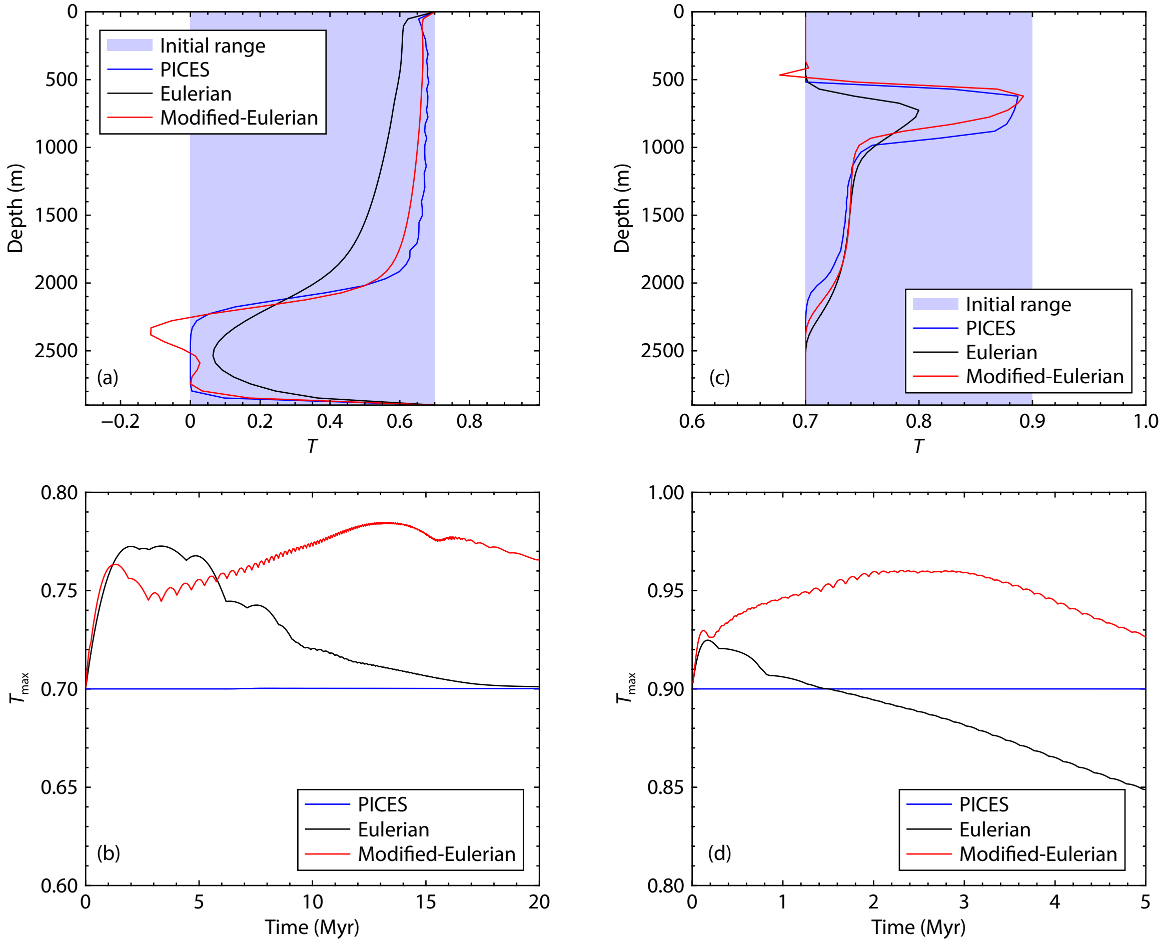

Figure 7. One-dimensional temperature profiles and maximum temperatures of the models: (a) The 1D temperature structure profile at the final state through the center of the cold blob; (b) evolution of the highest temperature of the cold blob models over time; (c) the 1D temperature structure profile at the final state through the center of the hot plume; and (d) evolution of the highest temperature of the hot plume models over time.

8a). Low-temperature ($T < 0$) oscillations are visible above the plume. In the Eulerian model, the diffusion of the plume thermal structure is also the most intense among all three cases, causing its temperature profile to be significantly blurred (Figure 8b) and the maximum temperature value to be notably reduced (Figure 7b). In this model, temperature artifacts appear mainly behind the rising plume, but the amplitude of oscillation is weaker than in the modified-Eulerian case (Figure 6).

The PICES model has clear and sharp thermal structures, similar to the modified-Eulerian case, and the overshooting effects are not visible (Figures 8c and 9c). The maximum temperature is always consistent with the analytical value of 0.9 (Figure 7d). In addition, we also test the conservation of energy, where we calculate the volume-integral of thermal energy in the model. At the final state of the PICES model, the total energy change of the model does not exceed 0.0013%, which is 3 to 4 orders of magnitude lower than that in other methods. From this result, it can be seen that the conservation is guaranteed.

3.4 Test with a 3D Spherical Global Data-Assimilation Model

To further examine the effectiveness of the PICES method, we implement a fourth test using a global model within a spherical coordinate system, based on which we compare the results from the modified-Eulerian and the PICES methods. The model domain is divided into 12 spherical caps, and each cap is further divided into 256 × 256 × 112 elements. Each element contains an average of 27 tracers. At the initial time corresponding to 270 Ma, the background temperature is equal to 0.7. Figure 10 shows the evolved thermal structure after 170 Myr of simulation.

In the modified-Eulerian model (Figure 10a), thermal oscillations around the subducting slabs and rising plumes can be clearly observed. These artifacts are concentrated in front of the sinking slab and above the mantle plume, similar to that depicted in Figure 1b. In contrast, the slabs and plumes in the PICES model have a continuous thermal profile across their edges, without any oscillations (Figure 10b). Because temperature influences both viscosity and density, the growing artifacts could also affect the dynamics of the model over an extended modeling time. For example, the slab structures within the same spatial locations display notable differences, and both slabs and plumes that are prone to nearby artifacts become thicker, consistent with the opposite-signed artifacts hindering the motion of the main features. Other disparities between the two models may also be

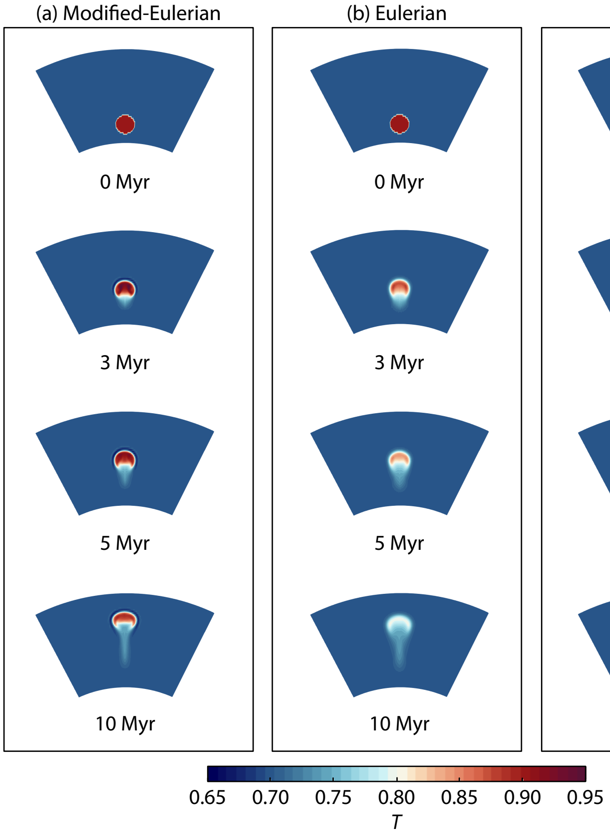

Figure 8. Comparison of the hot plume model based on three methods: (a) the modified-Eulerian method; (b) the Eulerian method; and (c) the PICES method.

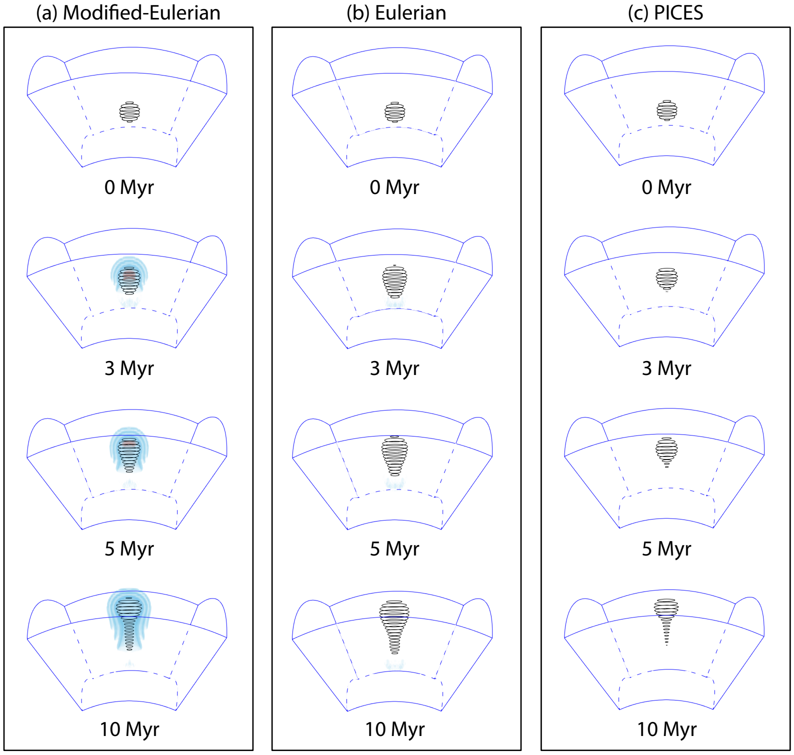

Figure 9. Comparison of the temperature artifacts associated with the hot plume from three methods: (a) the modified-Eulerian method; (b) the Eulerian method; and (c) the PICES method. The blue lines outline the modeling domains. The black lines outline the blob with high temperatures ($T > 0.71$). The red volumes represent high-temperature artifacts ($T > 0.9$) whose temperature is higher than the blob. The blue volumes represent low-temperature artifacts ($T < 0.7$) whose temperature is lower than the background mantle.

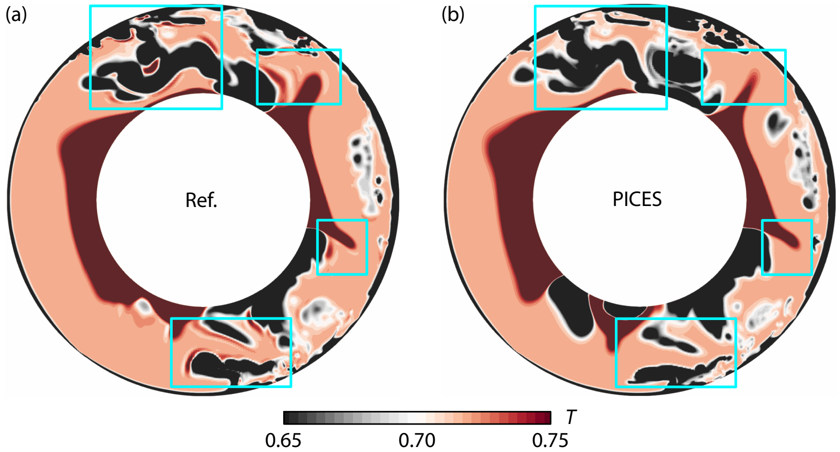

Figure 10. Comparison of global model results using two different methods: (a) the control group (the modified-Eulerian method), and (b) the experimental group (the PICES method). The cyan rectangles outline common regions where the two different methods show different model results.

explained similarly.

3.5 Parallel Efficiency Analysis

Taking the 3D regional CitcomS model with a hot plume as an example, we test the parallel efficiency of the PICES method while taking the modified-Eulerian method, which is more commonly used in current geodynamic codes, as a control group (Figure 11). For different parallel scales, the CPU execution time is calculated by executing 100 steps with a constant step size of 0.01 Myr, and other parameters remain unchanged. We test the penalization efficiency with 8, 16, 32, 64, and 128 nodes, respectively, where each node contains 56 cores.

As the number of cores increases, the execution time gradually decreases (Figure 11a). As the number of nodes increases from 8 to 64, the computation time is almost linearly related to the number of cores used (Figure 11b). When the computation scale reaches 128 nodes, the parallel efficiency deteriorates as the computation time plateaus. This behavior remains largely invariant between the control model (black symbols in Figure 11) and the experimental model (red symbols in Figure 11). This result shows that the PICES algorithm scales well within a large range of node numbers and that its implementation in CitcomS does not adversely affect the parallel efficiency of the code.

Several 3D finite element codes, such as CitcomS and ASPECT, have implemented the particle-in-cell method to trace the composition fields. We note that particles are present in both demonstrated models, as they solve thermal-chemical convection. Because the computation cost of PICES itself is minimal relative to other operations, its implementation does not notably increase the computation cost. In addition, the PICES is applicable in a spherical coordinate system and provides good parallelism with negligible extra communication costs.

When implemented in the existing code CitcomS, only the

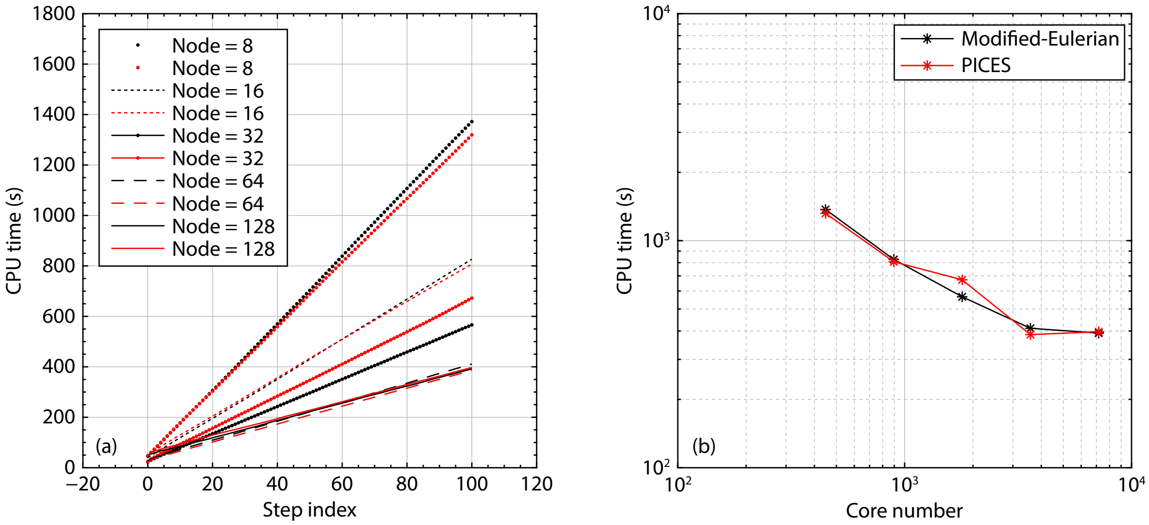

Figure 11. Parallel efficiency analysis: (a) CPU execution time versus the computation step index; and (b) the total CPU execution time of each case versus the number of cores. The red symbols represent the PICES method, and the black symbols represent the modified-Eulerian method. Each compute node contains 56 cores.

temperature interpolation between particles and elements and several mathematical functions need to be added. Modern CPUs, such as those used by the TianHe new generation supercomputer, are able to handle this case by automatic vectorization. Furthermore, for parallel jobs with more than 5000 cores, each process needs to handle only a small portion of the interpolation.

3.6 Spatial and Temporal Resolution

To illustrate the accuracy and convergence of the PICES method, we calculate the L_2 norm of errors at different spatial and temporal resolutions using first-order elements. Figures 12a–12b show the variation of the L_2 norm of errors and the order of convergence at different spatial resolutions.

As the number of elements gradually increases, the L_2 norm of errors decreases linearly (Figure 12a) and the order of convergence tends toward 1 (Figure 12b). However, the errors do not vary with the temporal resolution (Figure 12c) and their order of convergence tends toward zero (Figure 12d) when the spatial resolution is high enough. In other words, the errors of the PICES method depend mainly on spatial resolution but not on temporal resolution.

The result presented here shows that the PICES method exhibits convergence of $O(N)$ in first-order elements. This indicates that the PICES method ensures accuracy and convergence while effectively suppressing numerical oscillations.

4. Discussion

In practice, the mantle temperature strongly influences its viscosity, density, and other mechanical or phase properties that have been taken into account by most numerical geodynamic models (e.g., Zhong SJ et al., 2000; Gerya and Yuen, 2003; McNamara and Zhong SJ, 2004; Tan E et al., 2006; Gerya et al., 2015; Kaus et al., 2016; Luo T and Leng W, 2021; Ma ZQ et al., 2022; Zhao H and Leng W, 2023). However, overshooting and undershooting, which generate low- and high-temperature artifacts within the model, can affect the accuracy of the model solutions, both thermally and dynamically (Fletcher, 1988; Lenardic and Kaula, 1993; Samuel and Evonuk, 2010; Zhong SJ et al., 2015). Identifying and eliminating these artifacts has become increasingly difficult as models have become more complex, especially for models that include both subducting slabs and rising plumes (e.g., Schuberth et al., 2009; Gerya et al., 2015; Leonard and Liu LJ, 2016; Cao XZ et al., 2021; Flament et al., 2022; Pal and Ghosh, 2023), because existing numerical filter methods, such as the Lenardic filter (Lenardic and Kaula, 1993), are unable to effectively eliminate thermal anomalies that exist in the middle of the temperature range.

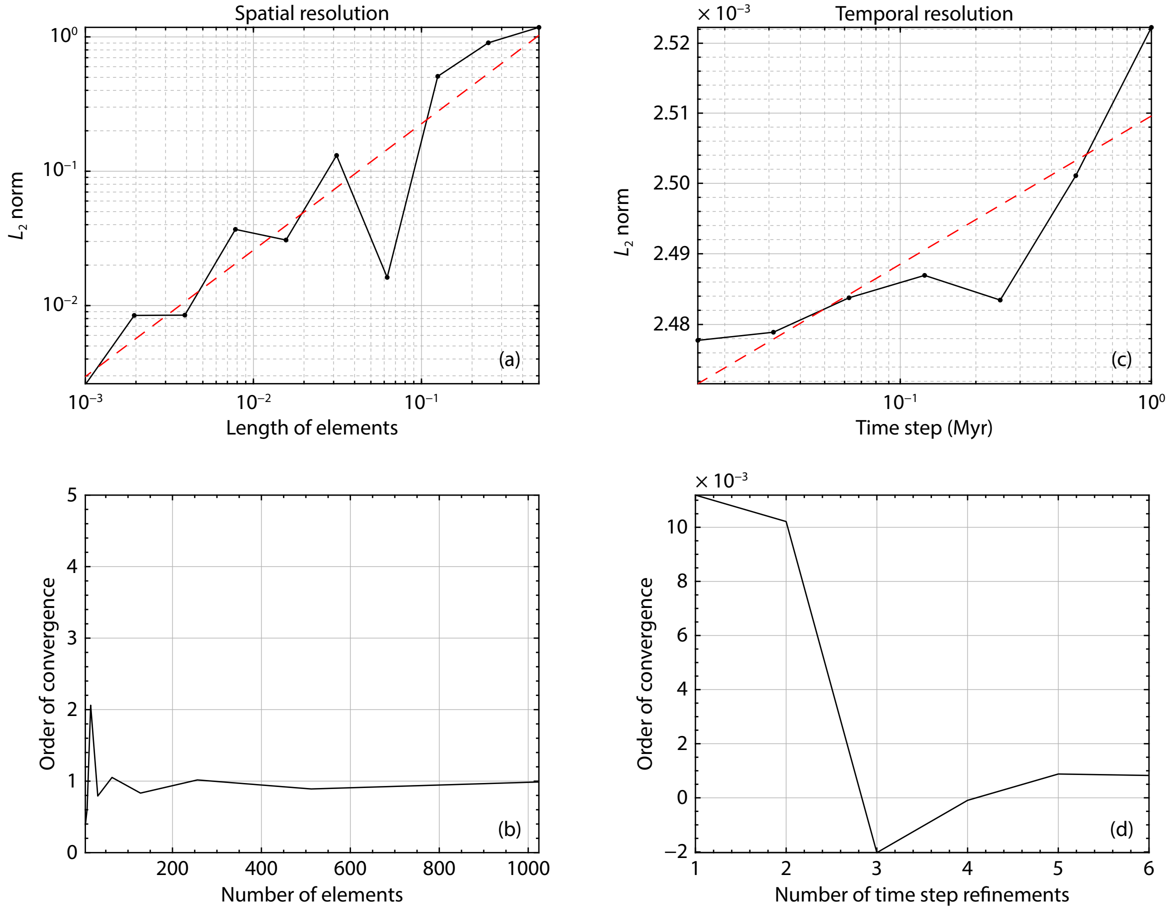

Figure 12. (a) The L_2 norm of errors versus the lengths of elements. The red dashed line is the result of an ordinary least squares fit to the black line. (b) The order of convergence versus the number of elements. (c) The L_2 norm of errors versus the time step. The red dashed line is the result of an ordinary least squares fit to the black line. (d) The order of convergence versus the number of time step refinements.

More specifically, without correcting these artifacts, the sinking or stagnant slabs may produce sharp temperature gradients in the direction of slab motion while the resulting temperature artifacts are warmer than the ambient mantle. In the absence of mantle plumes, these artifacts may be removed by using filters, such as the Lenardic filter (Lenardic and Kaula, 1993). However, the filter can only handle artifacts beyond the allowable temperature range. Once the model also considers mantle plumes, whose temperature is notably higher than that of the ambient mantle, the filter method can no longer eliminate the artifacts around the middle temperature values. The probability then arises that these artifacts may be misidentified as thermal anomalies generated by a mantle plume. In addition, these temperature artifacts may eventually accumulate and ascend to beneath the continent, leading to incorrect interpretations of the intraplate heat source.

Particle-based methods provide a better solution to this problem (e.g., Weinberg and Schmeling, 1992; Tackley, 1998; Gerya and Yuen, 2003; Tackley and King, 2003; McNamara and Zhong SJ, 2004). Compared with the interface tracking method (e.g., Christensen and Yuen, 1984; van Keken et al., 1997; Zhong SJ and Hager, 2003; Samuel and Evonuk, 2010), the Lagrangian methods generally fill the entire model domain with particles, making it more suitable for solving continuous field problems such as temperature. With the Lagrangian methods, a feasible way to handle the temperature artifacts is to separate the advection term from the governing equation. This treatment has been implemented in FD codes in the Cartesian coordinate system in previous studies and has been widely used in the field of geodynamics (Gerya and Yuen, 2003; Gerya et al., 2015; Kaus et al., 2016). However, for some 3D FEM models, especially those suitable for realistic data assimilation, this method has not yet been implemented.

CitcomS uses a streamline-upwind Petrov–Galerkin (SUPG) scheme for spatial discretization, and ASPECT has the option of using a discontinuous Galerkin (DG) scheme, both of which are improvements over the standard Galerkin method. Although CitcomS uses the SUPG scheme, which uses a more general weight function that has a similar effect on the upwind difference, it can reduce overshooting to some extent during actual testing but cannot completely eliminate the numerical errors. Even with the ASPECT regional model using the high-order DG method or others, some numerical errors still occur (see Supplemental Material, Figures S1 and S2). In addition, the DG method has not yet been implemented in global models.

The use of particles to describe the advection term does not lead to numerical dispersion. In geodynamics, this was pioneered in the FD framework, which combines the advantages of Lagrangian points with those of Eulerian points and allows the model to carry more information. It can suppress numerical oscillations well and has been widely used in some regional models.

In this study, a Lagrangian method, PICES, for FEM simulations is presented and implemented in a dimensionless spherical code, CitcomS, and its computational reliability and parallel efficiency are demonstrated. By testing four different cases, we show that the PICES has obvious advantages over the traditional field solver. This scheme successfully eliminates nearly all numerical dispersion while ensuring a reasonable solution for the advection–diffusion problem. In some cases, this new method still leaves behind weak artifacts (Figure 6c), but their amplitude and spatial extent are much smaller than those of the field solvers, with a negligible influence on the temperature solution (Figure 7a). This guarantees accurate modeling of scenarios in which both subducting slabs and mantle plumes are present.

In terms of the implementation scheme, previous applications in FD have used distance-weighted interpolation I2/3ELVIS and LaMEM (Gerya and Yuen, 2003; Gerya et al., 2015; Kaus et al., 2016). Considering that the FD models in Cartesian coordinates generally use structured grids, this scheme has a certain advantage in computational speed. For FEM models, besides the distance function in spherical coordinates, the shape function represents another natural weighting approach.

In comparison, the traditional temperature solvers produce fewer stable results than the PICES method. The modified-Eulerian method generates strong numerical dispersion around the moving object (Figures 5–10). The overshooting causes the interior of a cold blob to cool further and that of a hot plume to heat further (Figures 6a and 9a). Together with the numerical oscillation outside the main geodynamic features that affects both the viscosity and buoyancy, a slab would experience excessive thickness during subduction, whereas a plume would be broadened and retarded. The numerical oscillations generated by the Eulerian method often tend to occur behind the moving object (Figures 6b and 9b) and will cause unrealistically strong diffusion compared with the other two methods (Figure 7). Both will lead to inaccurate temperature solutions and dynamic evolution.

Although the PICES method has clear advantages, its computational speed is slightly slower than that of the conventional modified-Eulerian method (Figure 11a). That said, the PICES method can be further accelerated by using parallel methods that have a parallel efficiency similar to the modified-Eulerian method (Figure 11b). In addition, some local artifacts may be uncorrected by this method (Figure 6c), but with negligible amplitudes (Figure 7b). The origin of these artifacts still requires further investigation. In practice, this problem occurs mainly near the surface of the 3D model. During the process of data assimilation, the shallow structures, such as the lithosphere, are usually constrained by the known boundary conditions. Therefore, the remaining shallow artifacts could be diminished and prevented from growing by the practice of data assimilation, as shown in the global simulations (Figure 10).

5. Conclusions

In this study, we propose a PICES strategy, which is implemented in a dimensionless spherical FEM code, CitcomS, which supports distributed memory parallel computing and data assimilation algorithms. We test four numerical cases, and they all show that the PICES method has obvious advantages over the traditional Eulerian and modified-Eulerian methods. We further demonstrate that the parallel efficiency of the PICES method is similar to that of the traditional methods. This method successfully eliminates most overshooting during thermal evolution. Therefore, it not only improves the accuracy of the temperature field, such as that for a subducting slab and rising plume, but also guarantees the correctness of the associated density and viscosity structures.

Acknowledgments

This work was carried out at the National Supercomputer Center in Tianjin, and the calculations were performed on the TianHe new generation supercomputer. We are grateful to Ma Qingzhen and Li Taihe of the National Supercomputer Center in Tianjin for their patient assistance in providing the compilation environment. We thank the editor, Huajian Yao, for handling the manuscript and Mingming Li and another anonymous reviewer for their constructive comments. The research leading to these results has received funding from National Natural Science Foundation of China projects (Grant Nos. 92355302 and 42121005), Taishan Scholar projects (Grant No. tspd20210305), and others (Grant Nos. XDB0710000, L2324203, XK2023DXC001, LSKJ202204400, and ZR2021ZD09).

References

Cao, X. Z., Flament, N., and Müller, R. D. (2021). Coupled evolution of plate tectonics and basal mantle structure. *Geochem. Geophys. Geosyst.*, *22*(1), e2020GC009244. https://doi.org/10.1029/2020GC009244

Christensen, U. R., and Yuen, D. A. (1984). The interaction of a subducting lithospheric slab with a chemical or phase boundary. *J. Geophys. Res.: Solid Earth*, *89*(B6), 4389–4402. https://doi.org/10.1029/JB089iB06p04389

Flament, N., Bodur, Ö. F., Williams, S. E., and Merdith, A. S. (2022). Assembly of the basal mantle structure beneath Africa. *Nature*, *603*(7903), 846–851. https://doi.org/10.1038/s41586-022-04538-y

Fletcher, C. A. J. (1988). *Computational Techniques for Fluid Dynamics: Specific Techniques for Different Flow Categories*. Berlin: Springer. https://doi.org/10.1007/978-3-642-97071-9

Gassmöller, R., Lokavarapu, H., Bangerth, W., and Puckett, E. G. (2019). Evaluating the accuracy of hybrid finite element/particle-in-cell methods for modelling incompressible Stokes flow. *Geophys. J. Int.*, *219*(3), 1915–1938. https://doi.org/10.1093/gji/ggz405

Gerya, T. V., and Yuen, D. A. (2003). Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties. *Phys. Earth Planet. Inter.*, *140*(4), 293–318. https://doi.org/10.1016/j.pepi.2003.09.006

Gerya, T. V., Stern, R. J., Baes, M., Sobolev, S. V., and Whattam, S. A. (2015). Plate tectonics on the Earth triggered by plume-induced subduction initiation. *Nature*, *527*(7577), 221–225. https://doi.org/10.1038/nature15752

Hansen, U., and Yuen, D. A. (1988). Numerical simulations of thermal-chemical instabilities at the core–mantle boundary. *Nature*, *334*(6179), 237–240. https://doi.org/10.1038/334237a0

Hu, J. S., Liu, L. J., Hermosillo, A., and Zhou, Q. (2016). Simulation of late Cenozoic South American flat-slab subduction using geodynamic models with data assimilation. *Earth Planet. Sci. Lett.*, *438*, 1–13. https://doi.org/10.1016/j.epsl.2016.01.011

Kaus, B. J. P., Popov, A. A., Baumann, T. S., Püsök, A. E., Bauville, A., Fernandez, N., and Collignon, M. (2016). Forward and inverse modelling of lithospheric deformation on geological timescales. *Proceedings of Nic Symposium*, *48*, 978–983.

Lenardic, A., and Kaula, W. M. (1993). A numerical treatment of geodynamic viscous flow problems involving the advection of material interfaces. *J. Geophys. Res.: Solid Earth*, *98*(B5), 8243–8260. https://doi.org/10.1029/92JB02858

Leonard, T., and Liu, L. J. (2016). The role of a mantle plume in the formation of Yellowstone volcanism. *Geophys. Res. Lett.*, *43*(3), 1132–1139. https://doi.org/10.1002/2015GL067131

Luo, T. and Leng, W. (2021). Thermal structure of continental subduction zone: high temperature caused by the removal of the preceding oceanic slab. *Earth Planet. Phys.*, *5*(3), 290–295. https://doi.org/10.26464/epp2021027

Ma, Z. Q., Lu, G., Yang, J. F., and Zhao, L. (2022). Numerical modeling of metamorphic core complex formation: Implications for the destruction of the North China Craton. *Earth Planet. Phys.*, *6*(2), 191–203. https://doi.org/10.26464/epp2022016

McNamara, A. K., and Zhong, S. J. (2004). Thermochemical structures within a spherical mantle: Superplumes or piles?. *J. Geophys. Res.: Solid Earth*, *109*(B7), B07402. https://doi.org/10.1029/2003JB002847

Pal, D., and Ghosh, A. (2023). How the Indian Ocean geoid low was formed. *Geophys. Res. Lett.*, *50*(9), e2022GL102694. https://doi.org/10.1029/2022GL102694

Peng, D. D., and Liu, L. J. (2023). Importance of global spherical geometry for studying slab dynamics and evolution in models with data assimilation. *Earth-Sci. Rev.*, *241*, 104414. https://doi.org/10.1016/j.earscirev.2023.104414

Samuel, H., and Evonuk, M. (2010). Modeling advection in geophysical flows with particle level sets. *Geochem. Geophys. Geosyst.*, *11*(8), Q08020. https://doi.org/10.1029/2010GC003081

Schuberth, B. S. A., Bunge, H. P., Steinle-Neumann, G., Moder, C., and Oeser, J. (2009). Thermal versus elastic heterogeneity in high-resolution mantle circulation models with pyrolite composition: High plume excess temperatures in the lowermost mantle. *Geochem. Geophys. Geosyst.*, *10*(1), Q01W01. https://doi.org/10.1029/2008GC002235

Tackley, P. J. (1998). Self-consistent generation of tectonic plates in three-dimensional mantle convection. *Earth Planet. Sci. Lett.*, *157*(1-2), 9–22. https://doi.org/10.1016/S0012-821X(98)00029-6

Tackley, P. J., and King, S. D. (2003). Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations. *Geochem. Geophys. Geosyst.*, *4*(4), 8302. https://doi.org/10.1029/2001GC000214

Tackley, P. J. (2008). Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the yin-yang grid. *Phys. Earth Planet. Inter.*, *171*(1-4), 7–18. https://doi.org/10.1016/j.pepi.2008.08.005

Tan, E., Choi, E., Thoutireddy, P., Gurnis, M., and Aivazis, M. (2006). GeoFramework: Coupling multiple models of mantle convection within a computational framework. *Geochem. Geophys. Geosyst.*, *7*(6), Q06001. https://doi.org/10.1029/2005GC001155

van Keken, P. E., King, S. D., Schmeling, H., Christensen, U. R., Neumeister, D., and Doin, M. P. (1997). A comparison of methods for the modeling of thermochemical convection. *J. Geophys. Res.: Solid Earth*, *102*(B10), 22477–22495. https://doi.org/10.1029/97JB01353

Weinberg, R. F., and Schmeling, H. (1992). Polydiapirs: Multiwavelength gravity structures. *J. Struct. Geol.*, *14*(4), 425–436. https://doi.org/10.1016/0191-8141(92)90103-4

Wu, B. C., Wang, Y. M., and Huang, J. S. (2022). Dynamics of the subducted Izanagi-Pacific plates since the mesozoic and its implications for the formation of big mantle wedge beneath Eastern Asia. *Front. Earth Sci.*, *10*, 829163. https://doi.org/10.3389/feart.2022.829163

Zhao, H., and Leng, W. (2023). Aseismic ridge subduction and flat subduction: Insights from three-dimensional numerical models. *Earth Planet. Phys.*, *7*(2), 269–281. https://doi.org/10.26464/epp2023032

Zhong, S. J., Zuber, M. T., Moresi, L., and Gurnis, M. (2000). Role of temperature-dependent viscosity and surface plates in spherical shell models of mantle convection. *J. Geophys. Res.: Solid Earth*, *105*(B5), 11063–11082. https://doi.org/10.1029/2000JB900003

Zhong, S. J., and Hager, B. H. (2003). Entrainment of a dense layer by thermal plumes. *Geophys. J. Int.*, *154*(3), 666–676. https://doi.org/10.1046/j.1365-246X.2003.01988.x

Zhong, S. J., Yuen, D. A., Moresi, L. N., and Knepley, M. G. (2015). Numerical methods for mantle convection. *Treatise Geophys.*, *7*, 197–222. https://doi.org/10.1016/B978-0-444-53802-4.00130-5

Supplementary Materials of "Implementation of a particle-in-cell method for the energy solver in 3D spherical geodynamic modeling"

This file contains a supplementary text and two additional figures.

Here we present four results of different spatial schemes using the code ASPECT, each under the 2nd-, 4th-, and 8th-order cell. For these tests, only the advection equation is considered. The initial model has a background field of value 0 and a square of value 1 in the center; the left side of the square is fixed, and the right side is free falling. Figures S1 and S2 display model results from four different spatial schemes using elements of three different orders as noted above. As shown in Figures S1 and S2, these methods are able to suppress but still have varying degrees of overshooting.

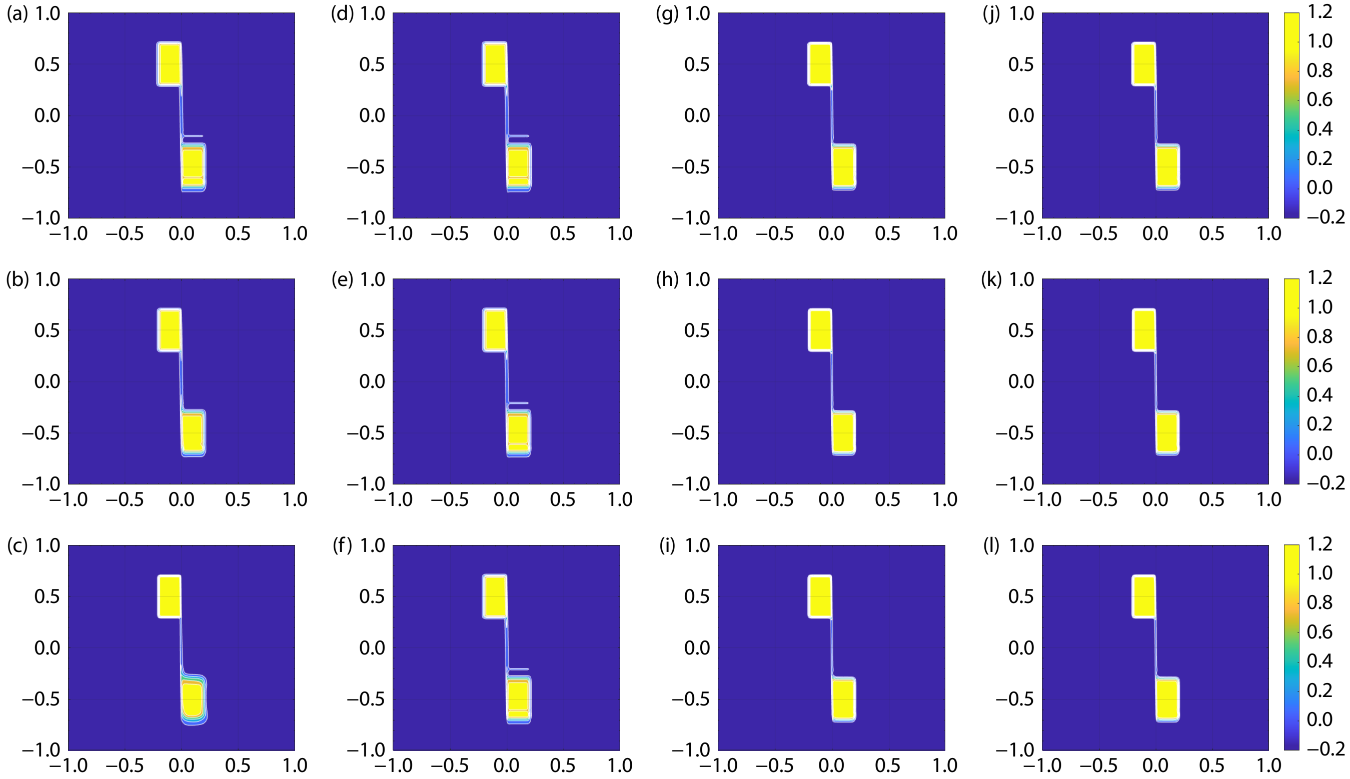

Figure S1. Numerical results of the advection equation for four different spatial discretization methods. (a–c) Continuous finite element method with 2nd-, 4th-, and 8th-order elements, respectively. (d–f) Streamline-Upwind Petrov-Galerkin finite element method with 2nd-, 4th-, and 8th-order elements, respectively. (g–i) Discontinuous Galerkin finite element method with 2nd-, 4th-, and 8th-order elements, respectively. (j–l) Discontinuous finite element Streamline-Upwind Petrov-Galerkin method with 2nd-, 4th-, and 8th-order elements, respectively.

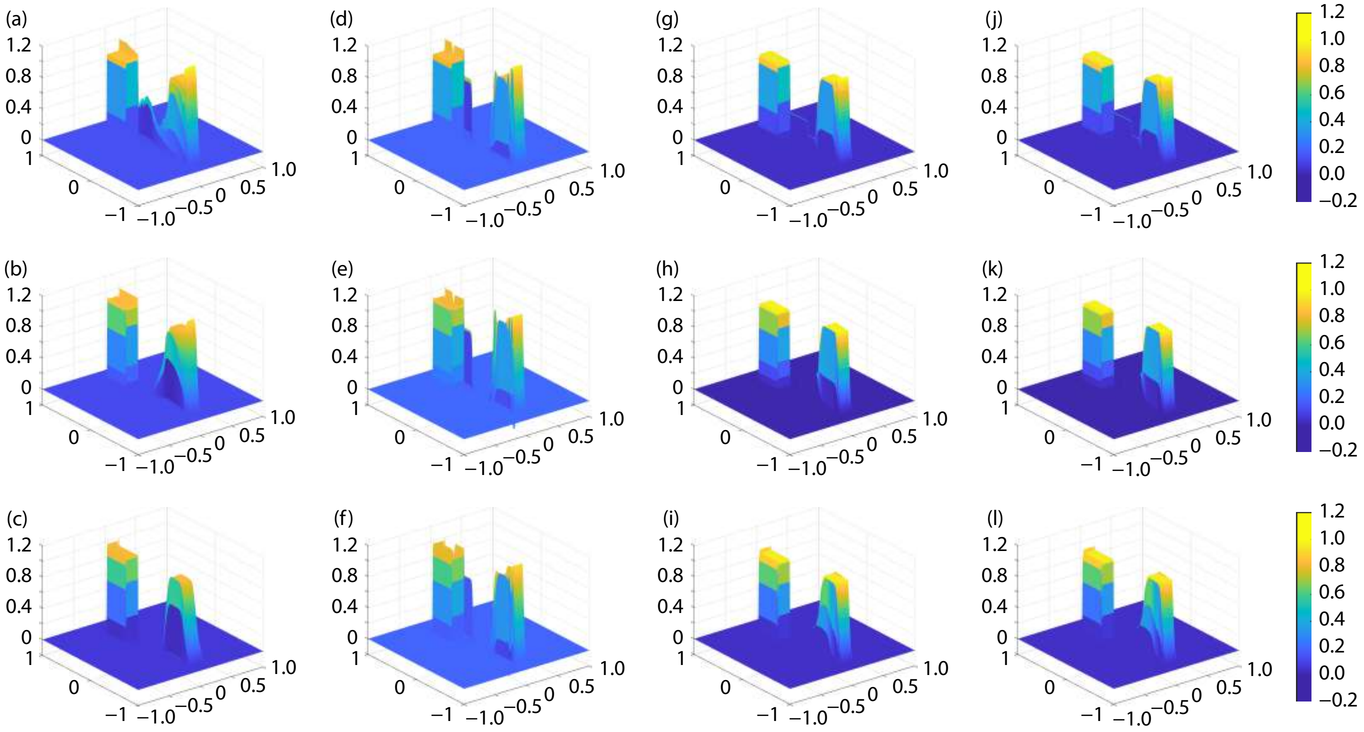

Figure S2. Same as Figure S1 except viewed in 3D.