



The tracer nudging method for correcting and preventing uneven tracer distributions in geodynamical models

Paul James Tackley

Department of Earth and Planetary Sciences, ETH Zurich, Zurich, 8092, Switzerland

Correspondence: Paul James Tackley (ptackley@ethz.ch)

Received: 21 March 2025 – Discussion started: 31 March 2025

Revised: 5 September 2025 – Accepted: 6 October 2025 – Published: 17 November 2025

Abstract. Tracers/markers/particles are commonly used in geodynamical models to track composition and sometimes other quantities throughout the domain. A common problem is that over time, gaps in the tracer distribution can develop, often resulting in cells with no tracers as well as bunching of tracers. These arise when tracer advection does not perfectly respect the mass conservation equation, so here this equation is used to derive a correction method that perturbs or “nudges” the positions of tracers in such a way as to close gaps and eliminate bunching. Test results show that this tracer nudging method is highly effective. Starting from an extremely heterogeneous tracer distribution with large regions of the domain devoid of tracers, it can produce an even distribution in only a few nudge iterations. In a time-stepping situation with a nudge every time-step, the amplitudes of the nudges are small yet sufficient to prevent gaps and bunches, allowing a low-order tracer advection method to be used while maintaining a tracer distribution that is more even than that obtained using higher-order advection methods alone. The computational cost is small – slightly larger than that of a first-order tracer advection step alone – because the method simply requires solving a Poisson equation. If an accuracy threshold is applied, a nudge correction may be necessary in only a fraction of time steps, with tests indicating that it is fastest to use low order advection with more frequent nudges than high order advection with less frequent nudges.

the framework of a so-called “marker-and-cell” or “particle-in-cell” method, in which velocity and pressure are calculated on a fixed Eulerian grid while various other quantities are advected on Lagrangian tracers/markers/particles (e.g. Harlow and Welch, 1965; Tackley and King, 2003; Gerya and Yuen, 2007), because the latter has some advantages to grid-based advection methods, such as lack of numerical diffusion or dispersion and the possibility of representing sub-grid-scale features. All of the major geodynamical modelling codes include this option, including CitcomS (Moresi et al., 2014), Aspect (Heister et al., 2017), Stag3D/StagYY (Tackley and King, 2003; Tackley, 2008), TERRA (e.g. Panton et al., 2025), LaMEM (Kaus et al., 2016), and I3ELVIS (Gerya et al., 2015).

This method relies on many tracers (e.g. 5–50) being present in each cell. Thus, it is problematic that over time, gaps in the tracer distribution typically develop, often resulting in cells containing no tracers. At the same time, bunching of tracers builds up. Such gaps and bunches typically develop when tracer advection does not respect the mass conservation equation, which can be due to (i) interpolation of velocities from grid points to tracer points not respecting conservation of mass (Pusok et al., 2017) and/or (ii) inaccurate advection of tracers in regions with large velocity gradients even with perfect velocities. The development of such gaps and bunches can be minimized by an optimal choice of tracer advection method (Pusok et al., 2017; Gerya et al., 2021) but apparently not eliminated, particularly since geodynamical simulations spanning the age of the Earth may require millions of time steps, giving small inaccuracies plenty of time to build up. Thus, some remedy, preferably one that is based on the equations being solved, is required. One remedy is to

1 Introduction

Tracers, alternatively named markers or particles, are commonly used in geodynamical models to track composition and sometimes temperature and other quantities, typically in

create new tracers to fill the gaps (Gerya, 2019), but this is unphysical.

Here, the mass conservation equation is used to correct non-conservation errors introduced during tracer advection by perturbing or “nudging” the positions of tracers. If performed frequently, such as every time step, the amplitudes of the nudges are small yet prevent large-scale gaps and bunches from building up. The method also works well when starting from an extremely uneven tracer distribution with large fractions of the domain initially devoid of tracers.

Irregularities in the spatial distribution of tracers can be quantified in terms of the number of tracers per unit volume (i.e. number density) calculated on a cell-by-cell basis. Alternatively, if tracers are considered to each have a mass (e.g. equal to the total mass of the domain divided by the number of tracers, although they could have different masses), then this can be thought of as a density, i.e. mass of tracers per unit volume. In the latter usage, the goal of this method is to nudge tracer positions in order to achieve, throughout the model domain, a tracer density that matches the correct physical density, which could vary with position if compressibility is included. The latter usage of “tracer density” is what the subsequent theory will mainly focus on.

In subsequent sections the mathematical theory is presented, followed by various tests of its effectiveness using the accompanying MATLAB program in two and three dimensions.

2 Mathematical Theory

As the goal is to achieve the correct tracer density everywhere, the first step is to calculate the current tracer density $\rho_t(x, y, z)$ on a cell-by-cell basis. Tracer density can be defined either as the mass of tracers per unit volume or (if tracers are massless) the number of tracers per unit volume. It can be dimensional (kg m^{-3} or m^{-3} , respectively) or nondimensional, as in the example MATLAB program. It is important that ρ_t changes smoothly as tracers move around, which it does not if one simply counts the number of tracers in each cell, because a tracer crossing a cell boundary causes an abrupt change in the densities of the two cells. Therefore, linear averaging of tracers to cells is important – termed “shape function” averaging by Tackley and King (2003) and widely recommended (e.g. Gerya, 2019; Ismail-Zadeh and Tackley, 2012). In this, each tracer contributes to the mass in 4 (in 2-D) or 8 (in 3-D) cells, linearly dependent on its distance from the cell centres using bilinear (in 2-D) or trilinear (in 3-D) functions analogous to the shape functions used in the finite element method. Once the tracer-based density in each cell is known, the tracer density error can then be calculated as

$$\Delta\rho_e = \rho_t - \rho_c, \quad (1)$$

where ρ_c is the correct density (e.g. of rock). This can in general vary with position, making the method applicable without modification to compressible flows, but for the purposes of the tests in this paper ρ_c is assumed to be constant.

The required perturbation (“nudging”) of tracer positions can be derived starting with the equation expressing conservation of mass:

$$\frac{\partial\rho}{\partial t} = -\nabla \cdot (\rho\mathbf{v}), \quad (2)$$

where ρ is the density field, $\mathbf{v}$ is the velocity field and t is time. Multiplying Eq. (2) by a finite time interval and substituting $\Delta\mathbf{x} = \mathbf{v}\Delta t$ leads to an approximate equation relating a finite change in density to a finite perturbation in position $\Delta\mathbf{x}$, which is here applied to the tracer density ρ_t :

$$\Delta\rho_t \approx -\nabla \cdot (\rho_t\Delta\mathbf{x}). \quad (3)$$

$\Delta\mathbf{x}$ can conveniently be expressed as the gradient of a mass flux potential ϕ (with units kg m^{-1} if ρ_t has units of kg m^{-3} or m^{-1} if has units of m^{-3}):

$$\rho_t\Delta\mathbf{x} = \nabla\phi. \quad (4)$$

Substituting Eq. (4) into Eq. (3) leads to a Poisson equation for ϕ :

$$\Delta\rho_t = -\nabla^2\phi. \quad (5)$$

The desired change in density $\Delta\rho_t$ is minus the density error, $\Delta\rho_e$, therefore the equation to solve is

$$\nabla^2\phi = \Delta\rho_e. \quad (6)$$

This can easily and efficiently be solved using standard methods such as multigrid. Assuming that the domain boundaries are impermeable, the appropriate boundary condition for ϕ is zero gradient perpendicular to the boundary; for other velocity boundary conditions Eq. (4) can be used to derive the appropriate condition on ϕ .

It is noted that another possible expression for $\Delta\mathbf{x}$ is

$$\Delta\mathbf{x} = \nabla\varphi \quad (7)$$

where φ is a displacement potential (with units m^2), leading to

$$\Delta\rho_t = \nabla \cdot (\rho_t\nabla\varphi), \quad (8)$$

which is slightly more difficult to solve and problematic in areas where $\rho_t = 0$, if such areas exist. Equation (4) also seems problematic in areas where $\rho_t = 0$ but as there are no tracers in these areas, there is no problem in practice.

This method does not achieve a perfectly uniform tracer distribution in a single nudge because ρ_t changes (towards the correct density ρ_c) during the displacement of tracers: Eq. (2) is an approximation. In areas of too-high ρ_t (decreasing during the correction step), Eq. (4) underpredicts the displacement, whereas in areas of too-low ρ_t (increasing during

the correction step), Eq. (4) overpredicts the displacement. Thus, when calculating the displacement from Eq. (4) it is best to use an average of the initial density and the correct density, rather than only the initial density. Tests indicate that a geometric average gives slightly better convergence than an arithmetic average, but both perform considerably better than using just the starting ρ_t . In summary, when calculating displacement, Eq. (4) is replaced by:

$$\Delta \mathbf{x} = \frac{\nabla \phi}{\sqrt{(\rho_t \rho_c)}}. \quad (9)$$

A single application of this algorithm achieves a considerable reduction of the density error (quantified using the L1 or L2 norm), which is sufficient during a time-stepping situation. If, however, starting from an extremely non-uniform tracer distribution with large portions of the domain being devoid of tracers, several iterations of the algorithm may be needed, as documented in Sect. 4.

3 Accompanying MATLAB scripts

This method is implemented in two and three dimensions in the accompanying MATLAB scripts (Tackley and ETH Zurich, 2025) (main program NUDGE.m), which can run the various test cases documented and discussed in Sect. 4. MATLAB scripts have the advantage of being easy to translate into other science and engineering-oriented high-level languages that include multi-dimensional arrays and array algebra, such as Julia (Bezanson et al., 2017) or modern Fortran (Metcalfe et al., 2024). Indeed, the method has already been implemented in the Fortran geodynamical modelling code StagYY (Tackley, 2008) and is in regular use.

The accompanying program uses a multigrid solver to obtain the displacement potential field. This is highly efficient but does require that the number of cells be a power-of-2 in each direction, or a power-of-2 times a small integer. Resolution is set by the number of cells in each direction nx , ny and nz , and the number of tracers by *tracers_per_cell*. Two-dimensional cases can be run by setting the number of y -points $ny = 1$. Densities are calculated at cell centres, while displacements and velocities are defined at cell boundaries in the standard staggered-grid finite volume arrangement (e.g. Harlow and Welch 1965; Patankar, 1980) as used by many codes in the geodynamical modelling community (e.g. Ogawa et al., 1991; Tackley, 1993; Trompert and Hansen, 1996; Gerya and Yuen, 2007; Kameyama et al., 2008; Tackley, 2008; Kaus et al., 2016). Domain boundaries are coincident with the perpendicular displacement/velocity points. Tracer positions are initialised either on a regular grid (with a smaller grid spacing than that on which the velocities/displacements are calculated), on a regular grid with random perturbations of up to half a grid spacing, or completely randomly. Initialising tracers on a regular grid causes artefacts with tracer alignment when they are advected, so regu-

lar + random is optimal. Completely random positions cause a density error that is typically a factor of 2 larger than regular + random, as shown later. The domain depth is assumed to be 1.0 and the grid spacing is the same in all three physical directions, meaning that the domain width in the x and y directions is given by (nx/nz) and (ny/nz) , respectively.

The MATLAB m-files are:

- *NUDGE.m*. The main program that runs and plots individual tests or test suites.
- *correct_tracer_density.m*. Performs the “nudging” algorithm detailed in Sect. 2.
- *tracer_density.m*. Calculates the cell-based tracer density field.
- *Poisson_solve.m*. Solves Poisson equation in 2-D or 3-D assuming zero-gradient boundary conditions.
- *advect_tracers.m*. Performs 1st-order Euler, 2nd-order or 4th-order Runge-Kutta tracer advection.

The core of the nudging algorithm in *correct_tracer_density.m* is compact, consisting of only four lines (Fig. 1).

4 Tests

Four test cases are presented. The first starts with various extremely non-uniform tracer distributions and tests how rapidly (in terms of number of nudging iterations) the method can create a uniform tracer distribution. The other three test cases involve time stepping, with the first two of these using analytical flow fields (cellular flow and shear flow along an interface) but the third being full thermal convection. After these, timings of the various routines are presented. Finally, adaptive use of the nudge correction (i.e. using when needed rather than once every time step) is tested.

4.1 Highly non-uniform tracer distribution tests

Various idealized initial tracer distributions are tested:

- i. *Half-empty*. Half of the domain is empty of tracers.
- ii. *Rectangular hole*. A rectangular region in the middle of the domain is empty of tracers.
- iii. *Spherical hole*. A spherical region in the centre of the domain is empty of tracers.
- iv. *Sphere*. All tracers are in a sphere in the centre of the domain, the rest being empty.
- v. *Random*. Tracers are placed randomly in the entire domain.

```

dens = tracer_density(tra,nx,ny,nz);           % calculate tracer density
phi = Poisson_solve (dens-mean(dens,"all"));   % calculate displacement potential
displ = displacement (phi,dens);               | % calculate displacement field
tra = advect_tracers(tra,displ,1,true,L);      % nudge tracers

```

Figure 1. MATLAB implementation of the algorithm in Sect. 2, in file `correct_tracer_density.m`.

Figure 2 (top row) shows these initial conditions and Fig. 2 (rows 2–5) shows the results of the first four correction nudges. After two nudges (a “nudge-nudge”; Fig. 2 middle row) tracers fill the domain; the subsequent nudge-nudge evens them out further. The evenness of the tracer distribution is quantified by tracer density plots in Fig. 3. After one nudge-nudge there is still significant unevenness, but this becomes difficult to discern after a further nudge-nudge. Random initial tracer positions (right column) lead to substantial initial unevenness in tracer density.

Figure 4 shows how the L1 norm of tracer density error decreases with number of nudges for the 2-D tests (Figs. 2 and 3) and for 3-D versions of the tests. For highly non-uniform initial conditions the reduction in tracer density error is more than an order of magnitude after 2 nudges, then becomes less rapid. Again, the random initial condition has substantial tracer density error approaching 0.2. 3-D cases are similar but with slightly slower convergence for the “sphere” case.

A problem in initial tests of the “sphere” case was that many tracers were nudged through the domain boundaries. This is due to the extreme nature of this test and is not a problem in a normal time-stepping application, but nevertheless a solution has been found. An approach that does not work is to place these tracers at the closest point inside the domain, although this does work for normal tracer advection by a velocity field that does not cross the boundaries. However, in this application the displacement field can substantially cross the boundaries, leading to a build-up of tracers at the boundaries, tracers that are not easily nudged away from there (close to the boundaries the perpendicular displacement is 0). What does work is to detect tracers that are nudged beyond external boundaries and instead apply only a fraction of the displacement to them. A fraction of 70 % was found to be optimal. That is, tracers that are initially calculated as crossing boundaries are instead moved only 70 % of the calculated distance.

4.2 Time-stepping cellular-flow test

The goal in this test is to determine whether the tracer nudging method can prevent gaps and bunches from building up in a time-stepping situation, as this is what is typically used in geodynamical simulations. Tracers are advected according to an analytically defined velocity field given by the curl of a two-dimensional stream function $S(x, z)$:

$$v_x = \frac{\partial S}{\partial z} \quad v_y = 0 \quad v_z = -\frac{\partial S}{\partial x}. \quad (10)$$

The resulting flow field is divergence-free for any S . In the presented tests, S is defined by

$$S(x, z) = \frac{1}{\pi} \sin\left(\pi \frac{x}{L_x}\right) \sin\left(\pi \frac{z}{L_z}\right) \quad (11)$$

where L_x is the length of the domain in the x -direction and L_z is the length of the domain in the z -direction. This gives a one-cell circulation pattern with no flow through the boundaries and velocities given by:

$$v_x = \frac{1}{L_z} \sin\left(\pi \frac{x}{L_x}\right) \cos\left(\pi \frac{z}{L_z}\right) \\ v_z = -\frac{1}{L_x} \cos\left(\pi \frac{x}{L_x}\right) \sin\left(\pi \frac{z}{L_z}\right) \quad (12)$$

In order to maximize the challenge of maintaining a uniform tracer distribution, tracers are advected using the first order forward Euler method, which usually makes them spiral outwards and concentrate towards the outside of the domain. This combination (Euler advection, 1 nudge per time step) is compared to three advection methods without any nudging: Euler, 2nd-order Runge-Kutta and 4th-order Runge-Kutta methods. Velocities at the staggered grid points are calculated using Eq. (12) and linearly interpolated to tracer positions. Tracers are initialized on a (regular+random) grid as discussed earlier, except in an additional (Euler advection, 1 nudge per time step) case with tracers initialized in completely random positions, to test what difference that makes.

Figure 5 shows tracer distributions and density error fields after 100 time-steps of nondimensional time 0.05 on a 32×32 grid with an average of 10 tracers per cell. As the maximum velocity given by Eq. (12) is 1, tracers move a maximum distance of 0.05 in one step. As expected, the Euler method (1st column) is quite inaccurate, with tracers spiraling outwards and building up towards the domain boundaries and corners. With the addition of a single nudge per step (right two columns), however, the tracer distribution remains even and negligible tracer density error is visible. The 2nd- and 4th-order Runge-Kutta methods produce similar results to each other, with significant unevenness visible in the tracer density error field.

The time-evolution of tracer density error is quantified in Fig. 6, which shows the L1-norm versus time step. The “Euler” case rapidly develops a large density error, whereas in both Runge-Kutta cases the error increases steadily from the initial condition, surprisingly at a similar rate for the 2nd- and 4th-order schemes. Adding a single nudge per step to Euler advection causes a reduction of density error to a low

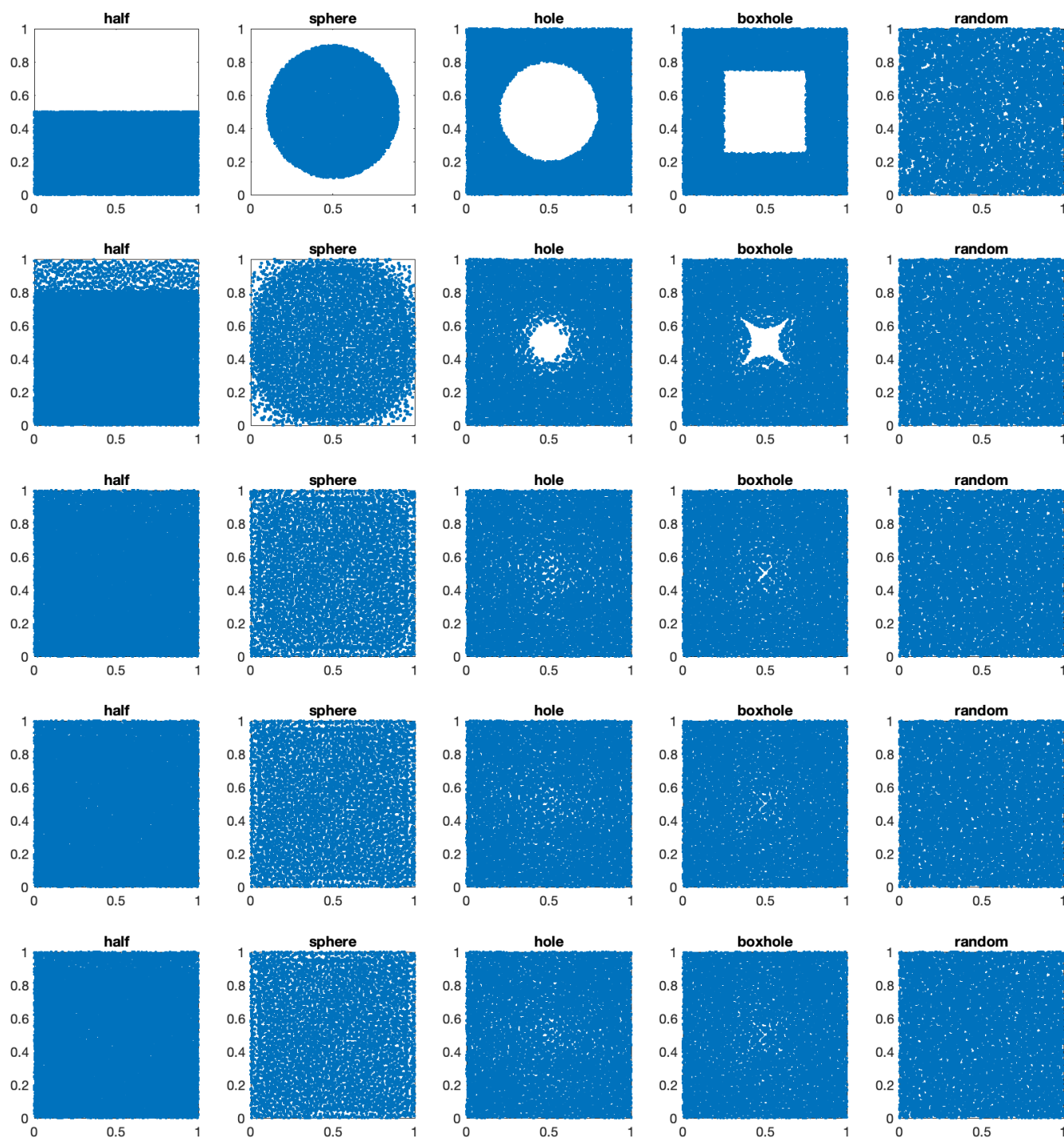


Figure 2. Tracer positions in the five highly nonuniform tests performed in 2-D with 32×32 cells and 10 tracers per cell on average. Each column is one test case and each blue dot is a tracer. Shown are (top row) the initial condition and (rows 2–5) nudges 1–4.

value, which is subsequently maintained. A completely random initial condition has a density error of $\sim$ a factor of 4 higher than (even + random), indicating that the latter initial condition is much better. Even so, adding a nudge correction per step rapidly reduces the error.

4.3 Opposing flow on an interface test

Pusok et al. (2017) thoroughly tested many marker advection methods using four different tests, of which arguably the most challenging was the first one, in which two rigid blocks move in opposite directions along an interface oriented at 45°

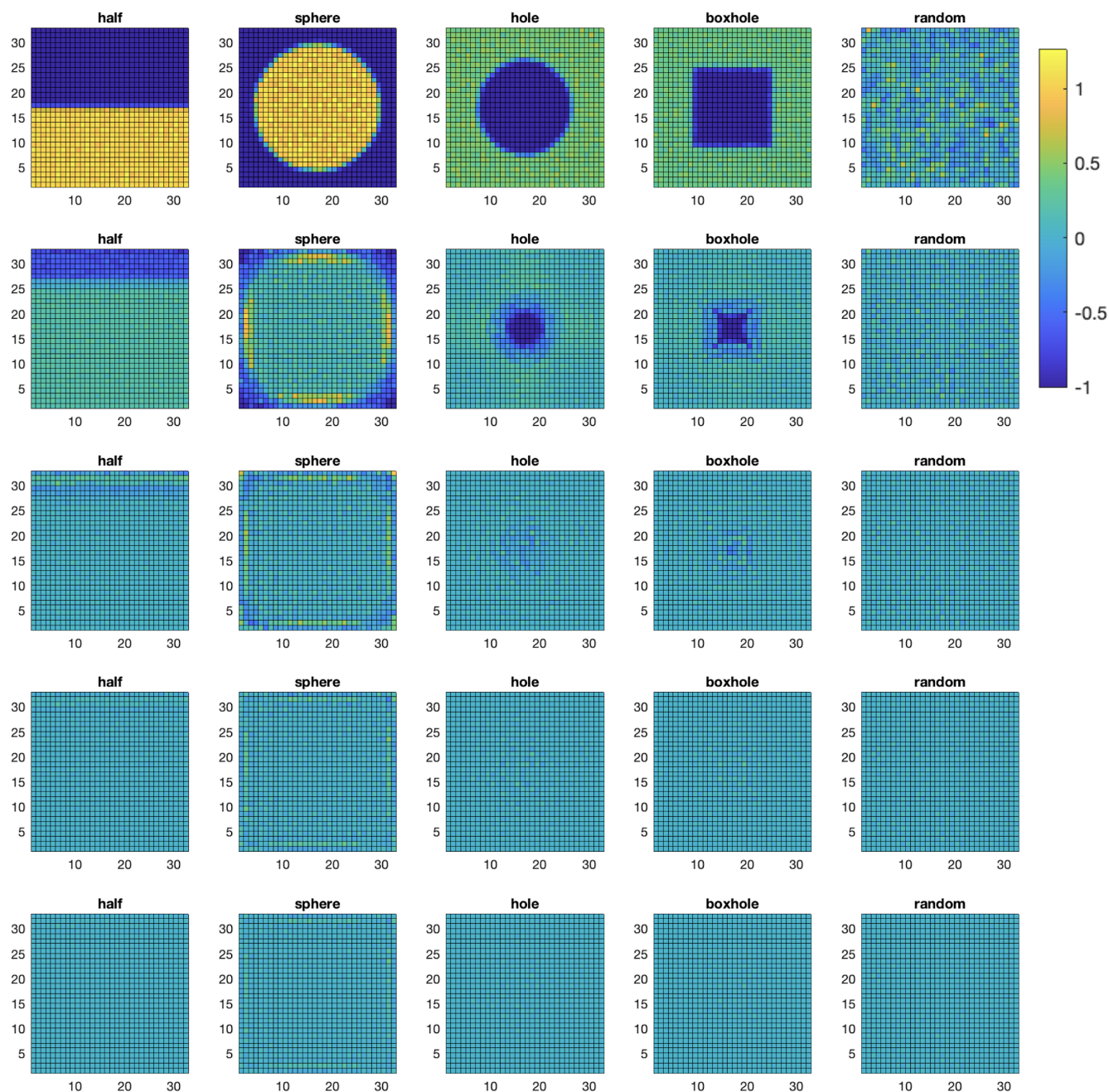


Figure 3. Tracer density error fields for the tracer distributions shown in Fig. 2. The colour bar is the same for all frames.

to the grid, mimicking to a subduction interface, for example. Material above the interface has a velocity $(v_x, v_z) = (1, 1)$ while material below the interface has a velocity of $(-1, -1)$, thus creating a large (shear) velocity change over one grid spacing. Tracers advected out of the domain are wrapped around at the appropriate place on a 45° line from where they left the domain. The advection methods tested in Pusok et al. (2017) almost all created a gap along the interface. It is here tested whether the nudge correction can avoid the gap along the interface.

Figure 7 shows the tracers and tracer density error fields using Euler, Runge-Kutta 2nd-order or Runge-Kutta advection, either on their own or with a single nudge correction per time step. Euler or 2nd-order Runge-Kutta methods indeed create a gap, while with the 4th-order Runge-Kutta method there is a band of tracers inside the gap. Away from the interface, the initial relative positions of tracers are preserved.

With one nudge correction each time step, tracer density error maps are greatly improved by eliminating the gap as well as reducing errors away from the gap. However, exam-

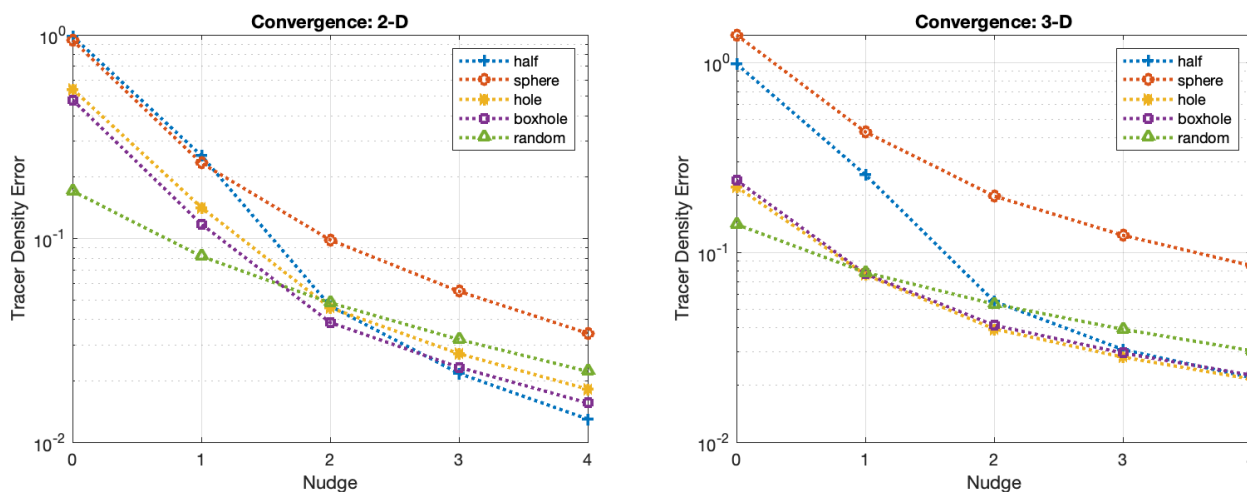


Figure 4. L1 norm of tracer density error versus number of nudges for the 5 initial tracer distributions in (left) 2-D 32×32 cells and (right) 3-D $32 \times 32 \times 32$ cells, in both cases with 10 tracers per cell on average.

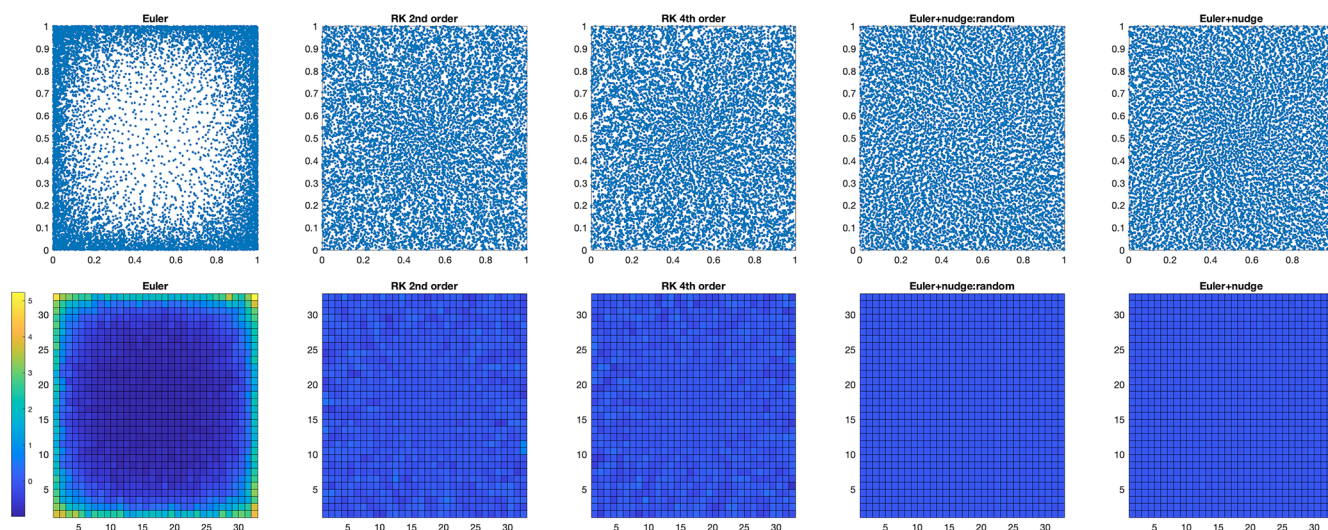


Figure 5. Tracer distributions (top row) and associated density error fields (bottom row) for the 4 advection methods on a 32×32 grid with an average of 10 tracers per cell. The right-most combination (Euler advection plus nudge correction) is performed with both completely random initial tracer positions (4th column) as well as the default positions. The colour bar is the same for all density error fields.

ination of tracer positions does show artifacts around the interface. The Euler and 2nd-order Runge-Kutta cases now display a series of small gaps oriented at 45° instead of one big gap. These are sub-grid-scale features that do not affect the cell-based tracer density field and so are not eliminated by tracer nudging. It could be that the rather artificial 45° angle of the interface allows these features to persist. The 4th-order Runge-Kutta test is much better. This is the only test in this study in which the 4th-order Runge-Kutta shows a distinct advantage over 2nd-order Runge-Kutta.

Graphs of tracer density error vs. time for advection (Fig. 8) show a rapid increase in error to begin with, subsequently stabilising and increasing only slowly. In the nudged-corrected cases, the error decreases then stabilises at a value

roughly an order of magnitude lower than that of the uncorrected cases.

4.4 Thermal convection test

The final test setup is that of thermal convection in an infinite Prandtl number fluid with strongly temperature-dependent viscosity and is thus representative of an actual geodynamical simulation. The Boussinesq approximation is assumed, with the fluid heated from below (nondimensional temperature $T = 1.0$), cooled from above ($T = 0$), having no internal heating and an exponentially varying viscosity $\eta(T) = \exp[-13.8155(T - 0.5)]$, which gives a factor of 10^6 viscosity variation. The Rayleigh number (at $T = 0.5$) is 10^6

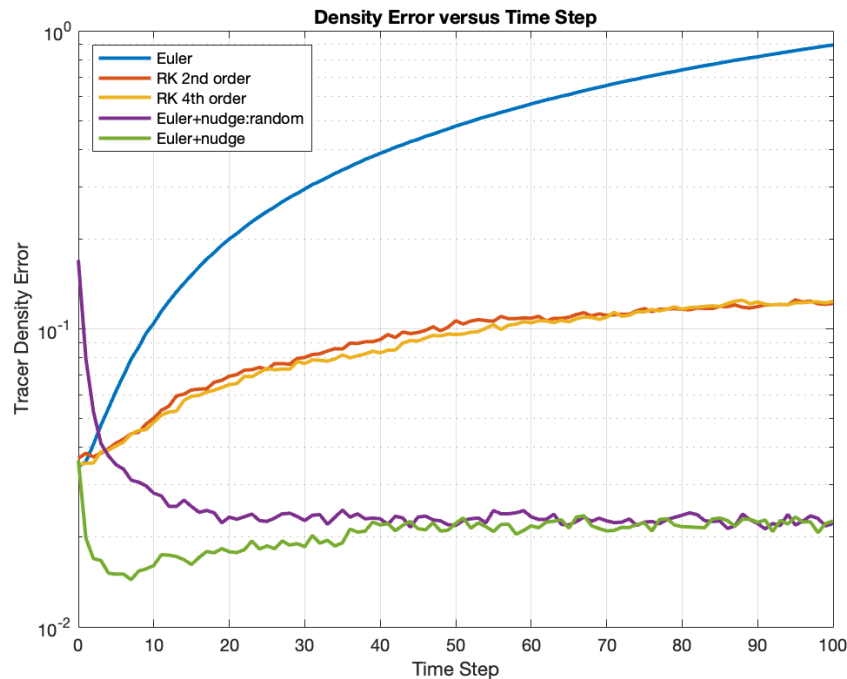


Figure 6. L1-norm of tracer density error versus time step for the tests in Fig. 5.

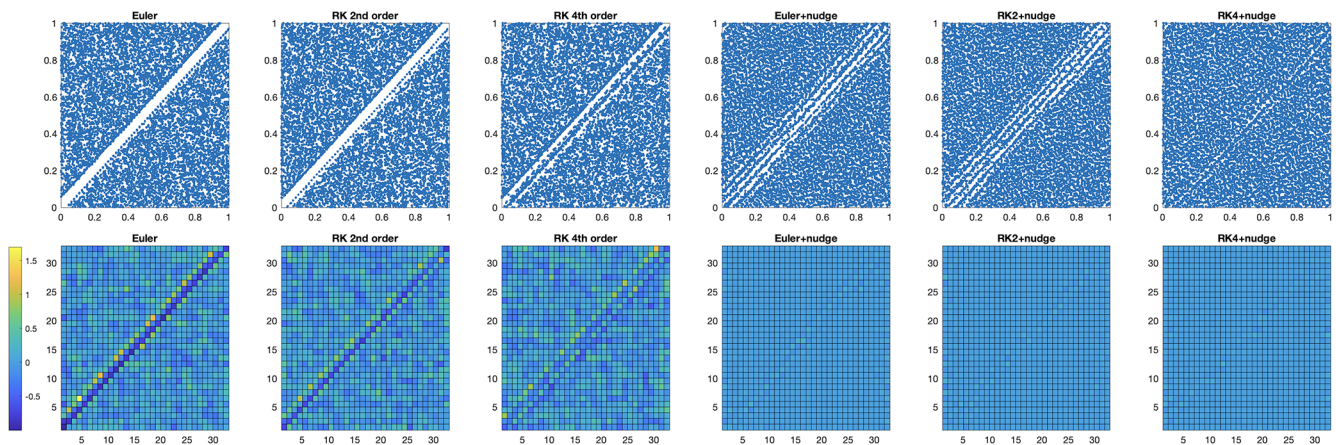


Figure 7. Tracer positions (blue) and tracer density error fields for the opposing flow on an interface test after 500 time steps and a 32×32 grid. See Pusok et al. (2017) for details of this test and comparisons with additional advection methods.

and the mechanical boundary conditions are all free slip. The initial condition has

$$T(x) = 0.5 + 0.01 \sin \left(\pi \left(\frac{1}{2} + 3 \frac{x}{L_x} \right) \right), \quad (13)$$

which leads to the formation of two hot plumes from the lower thermal boundary layer, as shown in Fig. 9. Flow is more rapid in these plumes due to their low viscosity. The time step is limited by the Courant condition because a finite-volume scheme is used for temperature advection and for thermal diffusion. The test is run for 500 time steps on a 32×32 grid with an average of 10 tracers per cell.

Tracer positions and density errors (Fig. 10) show large artifacts for the Euler advection method, which are much reduced by using Runge-Kutta advection. With one nudge iteration per time step, tracer density errors are greatly reduced for all three advection schemes.

The L1-norm of density error (Fig. 11) shows a rapid and continuing increase for the Euler scheme, but a much less rapid increase for the Runge-Kutta schemes. 4th order and 2nd order schemes give almost the same result, as is also visible by comparing the tracer distributions in Fig. 10. With a single nudge iteration per step, tracer density error is reduced to a much lower value than that of the initial condition, where

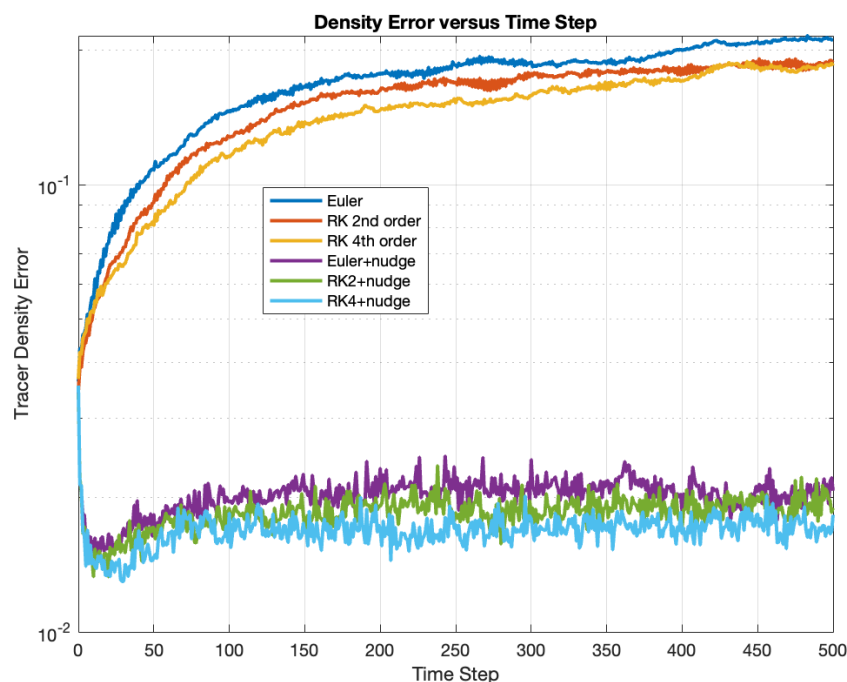


Figure 8. L1 norm of the tracer density error versus time step for the tests in Fig. 7.

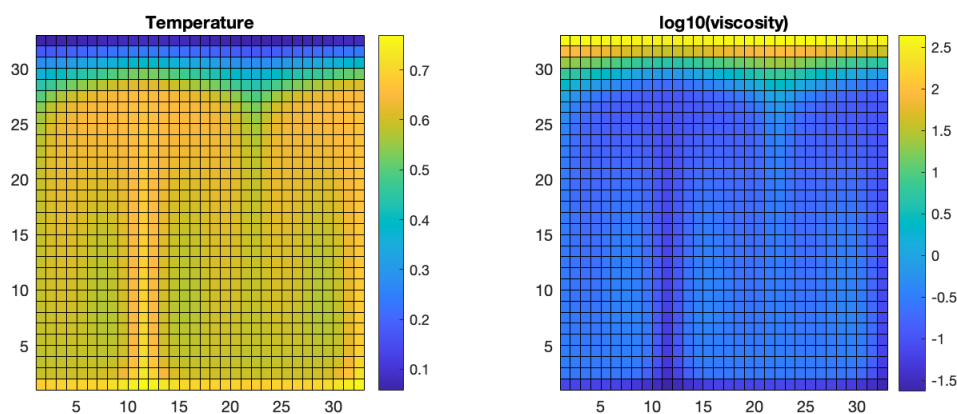


Figure 9. Non-dimensional temperature and viscosity fields for the thermal convection test after 500 time steps. They are the same regardless of tracer advection method because tracers are purely passive.

it remains stable at around an order of magnitude lower than the Runge-Kutta advection schemes. The error has the same magnitude for Euler and Runge-Kutta schemes. Thus, for pure advection there is no advantage to using 4th order instead of 2nd order, while with a nudge correction each step there is no advantage to using Runge-Kutta instead of Euler.

4.5 Timing analysis

Table 1 lists the CPU time taken for various order tracer advection steps, compared to one nudge correction and a 2D Stokes (v, p) solve. The increase in CPU time from 1st order (Euler) to 2nd-order Runge-Kutta to 4th-order Runge-Kutta is notable, with the latter taking about three times as

long as first-order Euler. A nudge correction takes slightly more CPU time than an Euler advection step, indicating that moving the tracers dominates the time; calculating the displacement field is relatively fast. The Stokes solve (see Matlab script `direct_solve_Stokes_2D.m`) solves for (v_x, v_y, p) on a staggered grid using Matlab's built-in “\” sparse direct solver, which uses UMFPACK. While the Stokes solve here seems fast compared to tracer advection, it is important to note that this is using a compiled, highly optimised solver while the tracer routines here use interpreted Matlab – if implemented in a compiled language like C, Fortran or Julia than they would likely be much faster, while the Stokes solve would not be.

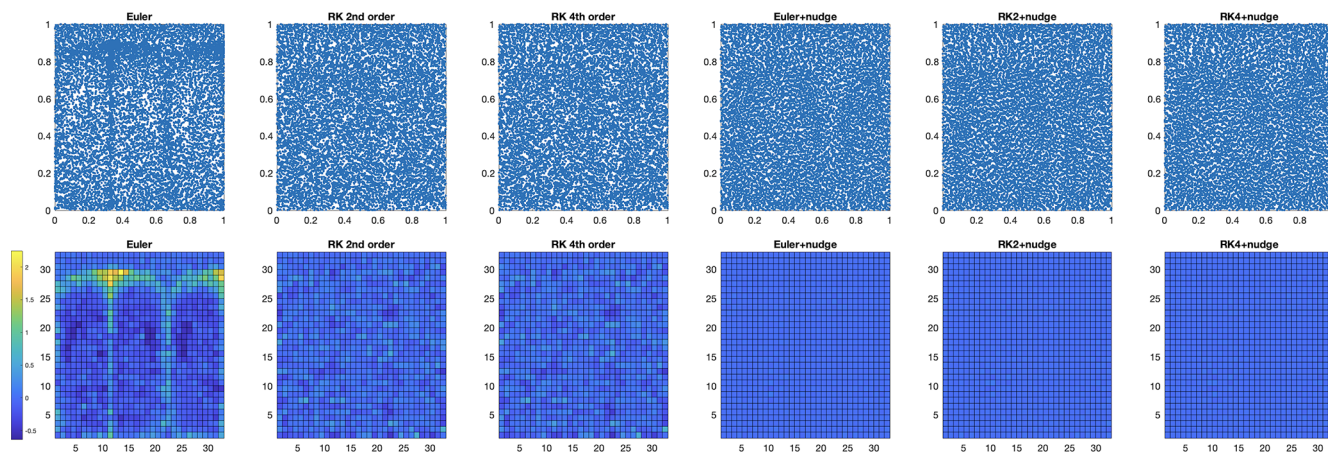


Figure 10. Tracer distributions (top row) and associated density error fields (bottom row) for the three advection methods without (left 3 columns) or with (right 3 columns) a nudge correction each time step.

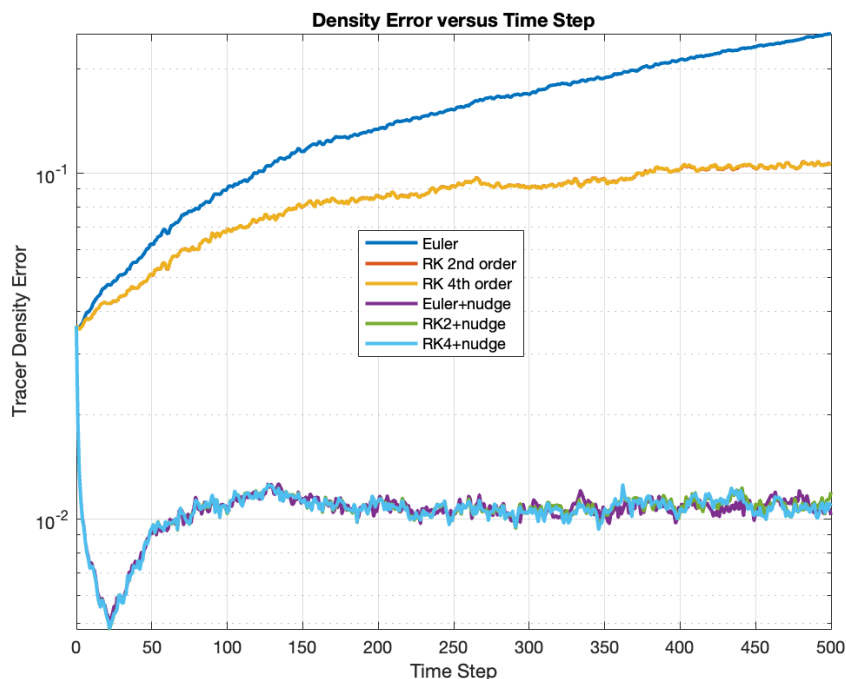


Figure 11. L1 norm of the tracer density error versus time step for the tests in Fig. 10. The RK 4th order curves mostly overlie the RK 2nd order curves, which are therefore not visible.

Comparing the two different resolutions indicates that, as expected, the time taken for tracer operations scales in proportion to the number of tracers: the higher resolution has 32 times as many tracers and requires proportionally more time. In contrast (also as expected) the time required for the Stokes solve increases more rapidly than the number of unknowns: the higher resolution has 16 times as many unknowns but takes 22 times longer.

4.6 Adaptive nudging

Instead of making one nudge correction every time step, another idea is to specify the required level of accuracy (in terms of L1 norm of tracer density error) and make a correction only when needed, or multiple iterations per step if a particularly low error is desired. This approach has been tested using the cellular advection test and the convection test, with results listed in Table 2. The error of 3.5×10^{-2} corresponds to the tracer density error associated with the initial condition (i.e., tracers initialised on a grid plus ran-

Table 1. For the thermal convection test, timings (in milliseconds) for first-order Euler tracer advection, 2nd- and 4th-order Runge-Kutta tracer advection, a nudge correction, and a 2-D Stokes (v , p) solve, at two different resolutions: 32×32 cells with 10 tracers/cell and 128×128 cells with 20 tracers/cell. Measured on a single core of a 3.8 GHz Intel Core i5 in a 2017 iMac, averaged over 100 time steps.

	Euler	RK2	RK4	Nudge	Stokes solve
32×32 10/cell	43.08	69.35	124.2	50.66	16.4
128×128 20/cell	1187	2057	3935	1384	360.1

Table 2. Average number of nudge corrections per time step required to keep tracer density error below a specified threshold, and total execution time of tracer operations for all time steps (in seconds), for the cell test with three different accuracies and the convection test with one accuracy. Timed on an Apple M4 Pro CPU (MacBook Pro Nov 2024).

	Euler	RK2	RK4
Cell; $dt = 0.05$; 100 steps; error = 5.0×10^{-2}			
Nudges/step	38 %	25 %	25 %
t_{total} (s)	6.298	8.218	13.48
Cell; $dt = 0.05$; 100 steps; error = 3.5×10^{-2}			
Nudges/step	82 %	70 %	67 %
t_{total} (s)	8.614	11.13	15.94
Cell; $dt = 0.05$; 100 steps; error = 2.0×10^{-2}			
Nudges/step	292 %	291 %	291 %
t_{total} (s)	19.48	22.07	29.86
Convection; $dt = \text{Courant}$; 500 steps; error = 3.5×10^{-2}			
Nudges/step	14.8 %	13.8 %	13.8 %
t_{total} (s)	7.170	11.02	18.47

dom perturbations), which therefore seems like a reasonable value to stay below. For the cellular advection test with this choice, a nudge needs to be made in 82 % of time steps with 1st order (Euler) advection, dropping to 70 % then 67 % for 2nd- and 4th-order Runge Kutta, respectively. If the error criterion is relaxed to 5.0×10^{-2} then the required number of nudges drops considerably (38 %, 25 %, 25 %), whereas if it is made stricter at 2.0×10^{-2} , then almost three nudges are required per time step regardless of the advection method.

Thus, there is a trade-off between increasing accuracy of advection and decreasing frequency of needed nudge corrections. In terms of execution time, however, it is in every case fastest overall to use the lowest advection accuracy (Euler) with somewhat more frequent nudge corrections. The increase in advection accuracy from 2nd-order to 4th-order Runge-Kutta is not justified, because the total execution time

risers considerably but the frequency of nudges remains almost the same.

In the convection test (Table 2 lowest section), nudges are needed considerably less often (in 14.8 %–13.8 % of steps) than in the cellular advection test with the same error criterion of 3.5×10^{-2} , despite the increased complexity of the flow. This is because the time step is smaller in the convection test: it is limited by the Courant condition such that the advection distance is a maximum of half a grid spacing, whereas in the cell test there is no such limitation and a time step of 0.05 is used, during which tracers may be advected several grid spacings (up to 1.6 for $n_z = 32$).

In summary, the results of these tests indicate that significant execution time can be saved by taking a nudge only when needed, and that the fastest approach is to use first-order advection, even though nudge corrections are needed slightly more frequently.

5 Conclusions

The tracer nudging method presented here is an effective way of eliminating and preventing gaps and bunching of tracers in geodynamical models/simulations. It uses the mass conservation equation to calculate tracer position perturbations (“nudges”) that correct mass conservation errors introduced by tracer advection. Starting from an extremely heterogeneous distribution with large regions of the domain devoid of tracers, it can produce an even distribution in only a few nudge iterations. In a time-stepping situation it allows a low-order tracer advection method to be used while maintaining a tracer distribution that is more even than that obtained using high-order advection methods alone. The computational cost is small and dominated by performing a first-order tracer advection operation, because the other part simply involves solving a Poisson equation. A nudge correction may not be needed every time step, which further reduces computational cost. It is more time-efficient to use nudge corrections in conjunction with low order tracer advection rather than high order tracer advection, even though the latter reduces the needed frequency of corrections.

Code and data availability. The exact version of the MATLAB code used to produce the results and figures used in this paper is archived on Zenodo under the MIT license under <https://doi.org/10.5281/zenodo.17058894> (Tackley and ETH Zurich, 2025). No input data or additional scripts are required. The data plotted in the figures can be generated by running the code.

Competing interests. The author has declared that there are no competing interests.

Disclaimer. Publisher's note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors. Views expressed in the text are those of the authors and do not necessarily reflect the views of the publisher.

Acknowledgements. The author is grateful to ETH Zurich. The author was motivated to write this method up by discussions on tracer advection with Boris Kaus, Thibault Duretz and Taras Gerya. The author thanks two anonymous reviewers for thoughtful suggestions that improved the manuscript.

Review statement. This paper was edited by Boris Kaus and reviewed by two anonymous referees.

References

- Bezanson, J., Edelman, A., Karpinski, S., and Shah, V. B.: Julia: A fresh approach to numerical computing, *SIAM Review*, 59, 65–98, <https://doi.org/10.1137/141000671>, 2017.
- Gerya, T., Duretz, T., and Ludovic, R.: New continuity-based velocity interpolation scheme for staggered grids, *EGU General Assembly 2021*, online, 19–30 Apr 2021, EGU21-15308, <https://doi.org/10.5194/egusphere-egu21-15308>, 2021.
- Gerya, T. V.: Introduction to Numerical Geodynamic Modelling (2nd Edition), Cambridge University Press, Cambridge, UK, <https://doi.org/10.1017/9781316534243>, 2019.
- Gerya, T. V. and Yuen, D. A.: Robust characteristics method for modelling multiphase visco-elasto-plastic thermo-mechanical problems, *Physics of the Earth and Planetary Interiors*, 163, 83–105, <https://doi.org/10.1016/j.pepi.2007.04.015>, 2007.
- Gerya, T. V., Stern, R. J., Baes, M., Sobolev, S. V. and Whattam, S. A.: Plate tectonics on the Earth triggered by plume-induced subduction initiation, *Nature*, 527, 221–225, <https://doi.org/10.1038/nature15752>, 2015.
- Harlow, F. H. and Welch, J. E.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids A*, 8, 2182–2189, <https://doi.org/10.1063/1.1761178>, 1965.
- Heister, T., Dannberg, J., Gassmüller, J., and Bangerth, W.: High Accuracy Mantle Convection Simulation through Modern Numerical Methods – II: Realistic Models and Problems, *Geophysical Journal International*, 210, 833–851, <https://doi.org/10.1093/gji/ggx195>, 2017.
- Ismail-Zadeh, A. and Tackley, P. J.: Computational Methods for Geodynamics, Cambridge University Press, Cambridge, UK, <https://doi.org/10.1017/CBO9780511780820>, 2012.
- Kameyama, M., Kageyama, A., and Sato, T.: Multigrid-based simulation code for mantle convection in spherical shell using yin-yang grid, *Phys. Earth Planet. Int.*, 171, 19–32, <https://doi.org/10.1016/j.pepi.2008.06.025>, 2008.
- Kaus, B. J. P., Popov, A. A., Baumann, T. S., Püsch, A. E., Bauville, A., Fernandez, N. and Collignon, M.: Forward and inverse modelling of lithospheric deformation on geological timescales, *NIC Series*, 48, 299 pp., ISBN 978-3-95806-109-5, <http://hdl.handle.net/2128/9842> (last access: 9 November 2025), 2016.
- Metcalfe, M., Reid, J., Cohen, M. and Bader, R.: Modern Fortran Explained, Incorporating Fortran 2023, Oxford University Press, 576 pp., ISBN 9780198876588, 2024.
- Moresi, L., Zhong, S., Han, L., Conrad, C., Tan, E., Gurnis, M., Choi, E., Thoutireddy, P., Manea, V., McNamara, A., Becker, T., Leng, W., and Armendariz, L.: CitcomS v3.3.1 (v3.3.1), Zenodo [code], <https://doi.org/10.5281/zenodo.7271920>, 2014.
- Ogawa, M., Schubert, G., and Zebib, A.: Numerical simulations of 3-dimensional thermal convection in a fluid with strongly temperature-dependent viscosity, *J. Fluid Mech.*, 233, 299–328, <https://doi.org/10.1017/S0022112091000496>, 1991.
- Panton, J., Davies, J. H., Koelemeijer, P., Myhill, R., and Ritsema, J.: Unique composition and evolutionary histories of large low velocity provinces, *Sci. Rep.*, 15, 4466, <https://doi.org/10.1038/s41598-025-88931-3>, 2025.
- Patankar, S. V.: Numerical Heat Transfer and Fluid Flow, CRC Press, <https://doi.org/10.1201/9781482234213>, 1980.
- Pusok, A. E., Kaus, B. J. P., and Popov, A. A.: On the Quality of Velocity Interpolation Schemes for Marker-in-Cell Method and Staggered Grids, *Pure and Applied Geophysics*, 174, 1071–1089, <https://doi.org/10.1007/s00024-016-1431-8>, 2017.
- Tackley, P. J.: Effects of strongly temperature-dependent viscosity on time-dependent, 3-dimensional models of mantle convection, *Geophys. Res. Lett.*, 20, 2187–2190, <https://doi.org/10.1029/93GL02317>, 1993.
- Tackley, P. J.: Modelling compressible mantle convection with large viscosity contrasts in a three-dimensional spherical shell using the yin-yang grid, *Phys. Earth Planet. Int.*, <https://doi.org/10.1016/j.pepi.2008.08.005>, 2008.
- Tackley, P. J. and ETH Zurich: Software accompanying manuscript “The tracernudging method for correcting and preventing uneven tracer distributions in geodynamical models” (2.0), Zenodo [code], <https://doi.org/10.5281/zenodo.17058894>, 2025.
- Tackley, P. J. and King, S. D.: Testing the tracer ratio method for modeling active compositional fields in mantle convection simulations, *Geochem. Geophys. Geosyst.*, 4, <https://doi.org/10.1029/2001GC000214>, 2003.
- Trompert, R. A. and Hansen, U.: The application of a finite-volume multigrid method to 3-dimensional flow problems in a highly viscous fluid with a variable viscosity, *Geophys. Astrophys. Fluid Dyn.*, 83, 261–291, <https://doi.org/10.1080/03091929608208968>, 1996.