

The two RCs are extremely helpful in pointing out clarifications, further explanations and further tests that would make the manuscript more helpful to the community, so I certainly thank the referees for their efforts.

There are two recommendations in common to both RCs: to include a test or tests that directly resemble geodynamic problems, and to examine the trade-off in execution time between this method versus high-accuracy advection methods, so I have added two more tests and studied the CPU time trade-off.

RC1

This manuscript tackles an important numerical difficulty in geodynamic modelling: the development of uneven tracer distributions during Lagrangian particle advection. The authors propose a *tracer-nudging* algorithm, derived from the requirement that material density remain constant, which iteratively redistributes tracers until a uniform spatial density is achieved. The idea is elegant and, if widely adopted, could mitigate one of the longest-standing practical problems in high-resolution mantle-convection and lithosphere-deformation studies. While I acknowledge the novelty and potential impact of the work, several aspects of the presentation and validation need to be strengthened before the paper is suitable for publication.

Major comments

1. Demonstrate the method on realistic geodynamic problems

The manuscript shows simple circulation tests only. Please include at least one geologically meaningful application—e.g. a high-viscosity-contrast convection benchmark or a 2-D subduction experiment—to illustrate how tracer nudging behaves in complex, time-dependent flow fields and when and how often the nudging is needed.

I have added two more tests, one of which is a “realistic geodynamic problem”, namely high-viscosity-contrast thermal convection.

2. Compare with established schemes

The paper states that the computational cost of tracer nudging is small, yet no comparison is provided. As far as I can see from the manuscript, the additional computational cost is probably higher than the existing remedies that only add a correction items to the velocity interpolation. It would be nice if the author make a comparison with other method and emphasize the advantages (and limitations) of this method

I have added a timing analysis (new Section 4.5). This shows that a nudge correction takes slightly more CPU time than performing first-order tracer advection. Since the correction procedure includes performing first-order tracer advection, it means that calculating the displacements takes a relatively short time. Examining the trade-off between order of advection (1st, 2nd or 4th) and how frequently a nudge correction is made to keep tracer density error lower than its original value indicates that optimal CPU time efficiency is obtained using 1st order advection with relatively more frequent nudges, rather than using higher order advection with less frequent nudges (new Section 4.6).

I am not aware of any other scheme that can reduce the error in tracer density (distribution). It's usually a question of minimising how rapidly things get worse; they all start failing at some point. Age-of-Earth global geodynamical simulations often require of order (1 million) time steps making it virtually impossible to find an uncorrected scheme that can prevent substantial degradation in tracer distribution.

3. Clarify applicability to compressible versus incompressible flow

Lines 34–35 imply the method corrects non-divergence-free advection errors. However, many geodynamic models employ *compressible* Stokes flow, where $\nabla \cdot \mathbf{v} \neq 0$ by definition.

1.
 - State explicitly that the current formulation targets incompressible Stokes problems.
 - Discuss whether, and how, the nudging algorithm could be adapted to compressible flow, and what errors might arise if it were used without modification.
 -

The algorithm works for compressible flows without modification. This is because ρ_c in Equation (1) can be an arbitrary function of position, which was already stated - but the earlier sentence that the referee pointed out indeed implies something different. Gone through the manuscript making sure that this is clear.

4. Streamline the figures

Figures 2 and 3 convey nearly the same information. I suggest keeping the one that best illustrates the tracer-density evolution and moving the other to the supplement or removing it entirely.

Figures 2 and 3 are really complementary because by comparing the two figures the reader can see clearly how a tracer distribution converts to a tracer density field (something that is not normally plotted in publications and therefore unfamiliar to the readers). So I would argue that it's best to keep both figures close to each other in the main text.

RC2

The study by P. Tackley presents a new algorithm for tracer advection for the marker-in-cell method. The algorithm corrects the displacement of tracers by solving for mass conservation of tracers. It is a neat, physics-based approach; however, I find that the choice of the potential function requires more discussion, and, if this algorithm is to be useful to the geodynamics community—or more broadly, to anyone using the marker-in-cell method, the writing can be improved and more tests would be useful. I detail these points below. Considering that the revision work I propose is considerable, I recommend a major revision of the manuscript.

1. Choice of the potential function for displacement vector in Eq. 3.

It is not clear why the author chose Eq. 4 (instead of Eq. 7) besides convenience to solve a linear Poisson equation. The justification is not there. I agree that it is more convenient to define $\Delta \vec{x} = \text{grad}(\phi) / \rho_t$, but what is the physical meaning of this equation? What would be the physical units of such a displacement? Maybe the author has thought about this, but the displacement should not depend on tracer density, and only Eq. 7 is a physical choice. Neither Eq. 4 nor Eq. 9 is balanced in terms of units. Has the author tried solving the non-linear Poisson equation in Eq. 8?

The meaning of Equation (4) is clearer if written as

$$\rho_t \Delta \vec{x} = \nabla \phi$$

The left-hand side is mass flux, thus ϕ is a mass flux potential. I have renamed it in the text and rewritten equation (4) in this form.

As the continuity equation is about $\text{div}(\text{mass_flux})$ and this correction algorithm redistributes mass (to achieve the correct density), it is logical and physical to use a mass flux potential. The (SI) unit of displacement is metres and the units of mass flux above are kg/m^2 , the physical meaning being kilograms flowing through a 1 m^2 area in one correction step. (In the continuity equation, mass flux is per second rather than per correction step). The units of ϕ are kg/m and equations (4) and (9) are dimensionally consistent with this. I have added the units to the text.

The alternative definition of ϕ used in equations (7) and (8) is a displacement potential with units of m^2 . I have changed the symbol to φ so that it's a clearly different quantity.

The example code is nondimensional so there are no units.

L77, Eqs. 7-8: Why is the non-linear equation problematic in areas where ρ_t is zero? Not sure this is an issue because in the marker-in-cell method, zero tracer density in control volume is a violation, so this is avoided (which makes the models in Fig. 2 somewhat artificial).

Discretization of partial differential equations such as (8) generally leads to a matrix problem of the form:

$$[\text{coefficients}][\text{unknowns}] = [\text{rhs}]$$

where $[\text{coefficients}]$ is a square matrix, $[\text{unknowns}]$ is a column vector containing the quantity being solved for at each grid point (in this case ϕ) and $[\text{rhs}]$ is a column vector of the known fixed right-hand-side. If too many coefficients are 0 then the matrix is singular and thus no solution is possible. That is the case here. The presence of $\rho_t=0$ in one or more cells results in a singular matrix and thus no solution.

Yes in principle the marker-in-cell method should have at least one marker in each cell, something that this method is intended to enforce. But after an advection step, it can be that there are cell(s) with no tracers and therefore this method would not work.

Minor comments regarding mathematical theory:

L46, Eq. 1: Need definition (equation) for tracer density in the main text. It was obtained from the code but was not obvious. $\text{Density} = \#n_{\text{local_mark}} * n_x * n_y * n_z / \#n_{\text{tracers}}$

Equally, for the mean density, which is taken equal to 1 here.

$\text{Mean density} = n_{\text{markx_cell}} * n_{\text{markz_cell}} * n_x * n_y * n_z / \#n_{\text{tracers}} = 1$

It was already stated that tracer density is mass of tracers per unit volume. Depending on the application, this could be nondimensional mass and density (as in the example code, this is why the mean value is 1.0) or dimensional mass and density (kg/m^3), in which case the code line pasted by the referee would be different. I have added more explanation of this.

L47: tracer density is "defined as the mass (or number) of tracers per unit volume". This statement is correct only if it is assumed that all tracers have the same mass, and the density relates to the number of tracers. This needs to be clarified.

This is an "or" statement, it's not both at the same time.

People use the marker-in-cell approach in different ways. Sometimes markers do not have mass, in which case the "number density" is the relevant quantity, and this is what Pusok et al. (2016) plot, for example; actually they used number per cell rather than number per volume.

Sometimes markers do have mass, in which case the usual "mass density" is the relevant quantity. In the latter case, markers can have different masses. The method works for all of these possibilities without changing the equations in the manuscript. The only difference would be that the units of ϕ are different if using "number density" instead of "mass density", as I now point out. I have clarified these things.

Eq. 2 to Eq. 3: the discrete version is missing for $v = dx/dt$.

Most likely the reader can follow this step, but I now point this out just in case.

L55-57, Eq. 1: ρ_c is mean density of tracers in this study. To generalise, better to use 'initial' instead of 'correct' (i.e., ρ_0). Also for the purpose of this study, ρ (without the t subscript) can be used for the tracer density.

In general, ρ_c can vary with position (see response to RC1 above) and can also change with time - it does not have to be the initial or the mean density. Therefore it is best to leave the subscripting as is. I have made this clearer in the text.

2. Generalisation and stopping criterion for nudging

L111: would the equations change for variable grid spacing?

No change. Equations (1)-(9) are physical so there is no change to them when the grid spacing varies, just as there is no change to the continuity, Stokes or energy equations.

Also, is the scheme generalisable for other boundary conditions, other than impermeable (L72)? Can the boundary conditions for Poisson be generalised from the velocity boundary conditions?

Yes. Given some equation(s) for the velocity boundary condition, equation (4) can be used to derive the relevant equation(s) for ϕ . I now point this out.

The number of 'nudges' required seems arbitrary and it is not clear what is the computational cost of each nudge. I wonder if a stopping criterion can be derived? Example, $\text{norm}(\text{density error}) \leq \text{tolerance}$, where tolerance = $1e-1$ or $1e-2$ from Fig. 4. Is that an overkill for using the algorithm?

A good idea, as a nudge might not be necessary every time step.

People don't normally complain about the initial tracer distribution in geodynamic models so a good tolerance to use seems to be the initial L1-norm of the tracer density error, which for the second test is about $3.5e-2$ (see Figure 6). Thus, I have added tests (new Section 4.6) to see how frequently a correction needs to be made for the prescribed-cell-flow test (Section 4.2) and the new convection test (Section 4.5). Indeed, for this tolerance a correction is not needed every time step, saving CPU time.

3. Geodynamic and performance tests

The tests presented are highly simplified, and maybe artificial (Fig. 2). To demonstrate better the impact of the algorithm, I recommend adding another test showing a geodynamic problem—with sharp velocity interfaces or rotational flows near corners. It would be interesting to see, for example, the problem cited in L27 on eruption/intrusion, which was stated, but not explained, and how this algorithm deals with it.

Two more tests have been added (Section 4.3-4.4). Eruption/intrusion algorithms will be the topic of a future manuscript.

Then, there is the issue of the computational cost of this algorithm (with nudges). Simple Matlab scripts run fast, but the marker-in-cell method becomes expensive for 2-D and 3-D problems. What is the cost of this algorithm? How does it vary with grid size (resolution)?

L201-202: says the computational cost of solving Poisson is small, but need to demonstrate. For example, using a geodynamics test case, show time to solve for Stokes and time for tracer correction ($\text{Poisson} \times \text{number_nudges}$).

Now done (Sections 4.5-4.6).

4. Focus and style of writing

The algorithm corrects the displacement of markers by solving for conservation of tracer mass. This does not transpire from the nudging and bunching described. The writing is too focused on the algorithm (which comes out almost as a trick to deal with marker dispersion/clustering), and not so much on the physics or why is it better than other methods. I think it is not doing a great job advertising why this could be a useful technique for geodynamics, and the informal style of writing does not help (e.g. usage of regular+, Euler+nudge+random without clear definition).

Other examples,

L40: the goal shouldn't be to nudge the tracers, but to correct their displacement in order to keep uniform density.

L46: the goal is to preserve the initial tracer distribution.

The Abstract should also be revised because I had the following questions:

L9: is the correction method physics-based? A sentence describing the method is lacking. Only results are described.

L14: what does the author refer to as non-conservation errors? How do they occur? They are introduced somewhat late, even in the abstract.

Excellent idea to start by stating that the mass conservation equation is used to derive the corrections, which are needed because of tracer advection not perfectly respecting the mass conservation equation. So I have now reworked the relevant introductory sections and all these points should be addressed now.

The details on the content of the Matlab routines in Section 3 (L112-118) and Fig 1. should be in an appendix. Instead, the author can outline a pseudo-algorithm such as

- Initialise tracer distribution, $\rho = \rho_0$
- Time loop:
 - advect tracers with velocity field v ,
 - calculate ρ and density error,
 - correct tracer location by solving for displacement dx (Poisson equation).

I'd like to do what is normal and customary. Some published GMD papers have actual code (not pseudo-code) in the main text as well as details of how the code works, so this does seem acceptable. Examples:

Räss et al. (2022): <https://doi.org/10.5194/gmd-15-5757-2022>

Cheng et al. (2025): <https://doi.org/10.5194/gmd-18-5311-2025>

Therefore I leave it as it is.

Minor comments

L6-7, 17: tracers may track other fields besides composition and temperature throughout the domain. Their advantage is that they may perform better at advecting these fields compared to grid-based methods. L7: why does the problem occur? i.e., rotating fields in a box domain or at sharp interfaces. Addressed in the reworked introductory sections.

L8: Suggest to use 'clustering' or 'accumulation' instead of 'bunching'. I'm not finding a reason why one of these is better => leave as is.

L20: (e.g., \cite not \citep) Changed

L21: why all the major codes use marker-in-cell? It is better to justify advantages rather than state a common practice. Addressed in the reworked introductory sections.

L24: (e.g. 5-50) not necessary. The accuracy of the marker in cell technique relies on having a high density of tracers in each cell. I disagree, as someone who is new to the method might wonder whether "high density" means e.g. 10 or 100000.

L29: why the errors occur in the first place? L30: physics-based remedy is required Addressed in the reworked introductory sections.

L31: one previous solution is to create new tracers, but why is this not a good remedy? It's unphysical.

L44, L96: just because it is already implemented in StagYY, it is not a statement of validation or verification. StagYY should cite the work in the current manuscript to demonstrate the method, not the other way around. Already-published StagYY papers cannot cite this paper, so the point here is mainly to reassure the reader that some already-published papers benefit from this method. Anyway I reduced it to one statement.

L109: first time density bounds are given; this should occur earlier when density is defined. The magnitude of tracer density error is very application-dependent, so belongs where the results are presented.

L149: unclear statement. Tracers at the boundaries were nudged only a fraction of the calculated displacement to avoid crossing the boundaries. Not quite. Tracers that are initially calculated as crossing boundaries are instead moved only 70% of the calculated distance. Now stated like this.

Eqs. 11-12: could plot either the stream function or plot the velocity field to understand why the center particles might get dispersed. The stream function is a sine (0->pi) in x multiplied by the same in z so looks almost circular giving a simple "one cell circulation pattern". This seems too simple to be worth plotting particularly as I already doubled the number of figures.

Section 4.2 notation: need partial differentials (∂). Equation (10) changed accordingly.

Eq. 12: mistake $V_x = 1/L_z \sin() \cos()$ Fixed

L172: define Euler+nudge+random Defined

L174: calculated as in Eq. 12 (not above) Changed

Figures 2 and 3 do not have colorbars or state that blue points represent tracers. Figure 2: Colorbar not needed. Added that each blue dot is a tracer. Figure 3: Added colorbar

Fig 5: what is the computational cost or runtime between the cases? Add colorbars. See Sections 4.5-4.6. Colorbar added.

Fig 6: all cases should start with the same initial condition, and then additionally a case with completely random distribution. The missing case was added.

Matlab: Add in readme how to run the script. I.e. change ntest = 1; choice between 1-11. Added a README.txt