https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Lagrangian tracking methods applied to free surface boundaries in numerical geodynamic models

Timothy Gray[1], Paul Tackley[1], and Taras Gerya[1]

[1]Institute of Geophysics, Department of Earth and Planetary Sciences, ETH Zürich, Sonneggstrasse 5, 8092 Zurich, Switzerland

Correspondence: Timothy Gray (timothy.gray@eaps.ethz.ch)

Abstract. A desirable characteristic of mantle convection models is the ability to determine surface topography evolution over time on a global scale. This capability is increasingly important due to growing interest in coupling planetary geodynamics with climate, landscape, habitats, and biological evolution systems. A common way of achieving this in numerical geodynamic models with a fixed Eulerian grid is through the implementation of a free-surface boundary condition using the sticky air method in conjunction with an appropriate method of tracking the free surface. Although existing methods for tracking the interface between the air and rock layers are available, they often struggle to provide high-resolution results on a global scale without incurring significant computational costs. We propose a method for representing surfaces directly using Lagrangian markers that can track the location of a free surface in 2D and 3D models and test it using the finite-volume mantle convection code StagYY. This approach offers a direct, high-resolution representation of the surface without the need for a large number of high-cost tracers throughout the model domain. Benchmarks demonstrate the effectiveness of this method compared to the commonly-used marker-in-cell method. The direct representation of the surface enables additional features such as the direct tracking of sea levels over time, potential for coupling with surface process models on the global scale, and enables the implementation of alternative discretisations of the Stokes equations to improve Stokes solver accuracy near the free surface boundary.

1 Introduction

Modelling the evolution of planetary surfaces over time has attracted considerable attention in geodynamics, since surface topography is a primary observable of a planet's internal dynamics and can also change the dynamics compared to assuming a flat surface (e.g. Crameri et al. (2012b)). The growing interest in coupled planetary systems, particularly the coupling between geodynamic processes, climatic processes, and biological evolution, further motivates the study of topographic evolution due to the coupling between these systems occurring primarily at the surface. The study of these couplings, also referred to as *biogeodynamics* (Zerkle, 2018; Stern and Gerya, 2023), requires accurate methods of determining how mantle convection affects the surface, and in turn affects other systems in order to study fundamental questions regarding the habitability of planetary bodies.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Geodynamic models commonly represent a free surface using a fixed, Eulerian grid with "sticky air" representing material above the surface (e.g. the various modelling codes in Crameri et al. (2012a) and Schmeling et al. (2008)). Modelling free surfaces on fixed Eulerian grids in numerical geodynamic models requires two key components: an appropriate solver that can model or approximate the solution to the Stokes equation with a free surface boundary condition and a method of tracking the location of the surface over time.

This study implements a method for tracking free surfaces on a global scale. The method is implemented into StagYY (Tackley, 2008), which uses the staggered-grid finite difference discretisation (equivalent to the finite volume discretisation), as is often used in geodynamic modelling codes (e.g. (Patankar, 1980; Ogawa et al., 1991; Tackley, 1993; Trompert and Hansen, 1996; Gerya and Yuen, 2007; Tackley, 2008; Gerya et al., 2015; Kaus et al., 2016)) due to its simplicity in discretising the governing equations and compatibility with efficient multigrid solvers in both 2D and 3D. However, the free surface tracking methods are also generally applicable to other Eulerian-grid geodynamic codes.

For clarity, the Lagrangian points already implemented in StagYY and distributed throughout the model domain are hereafter referred to as "tracers". The existing surface tracking approach—the marker-in-cell technique—uses these tracers to infer the surface location. In contrast, Lagrangian points introduced in this study specifically for tracking the free surface are referred to as “surface markers”.

In StagYY, as in many other codes, the standard technique for interface tracking employs Lagrangian tracers using a marker-in-cell method. In this approach, Lagrangian tracers throughout the model are used to represent information about material properties (in this case whether they are rock or air), which are then averaged to determine the density and viscosity fields used by the Stokes solver, as well as the location of the free surface. However, when running large-scale global models, particularly in 3D, computational cells are typically large due to computational limitations. A typical global scale 3D spherical model can only achieve a vertical resolution of approximately 10 km near the surface while remaining computationally tractable, even after refining the vertical grid spacing near the surface (Coltice et al., 2019). For modelling topography on Earth-like planets, where the maximum topographic variations are on the order of ± 10 km, this vertical cell resolution necessitates a method for tracking topography on a fine sub-grid scale. Using the marker-in-cell method, finer resolution can be achieved trivially by increasing the number of tracers. However, this approach scales poorly: doubling the vertical tracer density requires 4 times as many tracers in 2D and 8 times as many in 3D. This increases both memory usage and computational costs for storing and advecting these tracers. This inefficiency motivates alternative methods of tracking the free surface.

A successful surface tracking method must satisfy several criteria:

- Provide fine sub-grid resolution of topography on the global scale.
- Remain stable over many timesteps and across diverse geodynamic regimes, including subduction.
- Ensure compatibility with the underlying Stokes solver (e.g. preserve mass conservation).
- Reduce computational cost relative to the marker-in-cell method.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Interface-tracking approaches that meet these criteria include level-set and volume-of-fluid methods. However, one of the conceptually simplest options is to track interfaces directly using Lagrangian markers—often referred to as direct interface tracking in the CFD literature (Tezduyar, 2006). In this approach, the surface is explicitly represented by Lagrangian markers that follow the surface flow. Using such markers only at the surface avoids the scaling limitations of the full marker-in-cell approach, offering potentially higher accuracy at lower computational cost.

Several characteristics of surfaces in global scale geodynamics make Lagrangian interface tracking attractive:

- For Earth-like planets, the maximum surface deviation (± 10 km) is relatively small relative to the mantle's total thickness (2,890 km).
- The surface height can be considered a one-to-one function of position (i.e. the surface cannot overlap itself).
- Planetary surfaces are sharp interfaces; there is no diffused mixed region of air and rock in the vicinity of the surface.
- Many geodynamic codes, including StagYY, already use Lagrangian tracers, enabling simple implementation through the repurposing of existing code.
- Lagrangian markers can provide fine sub-grid level resolution, essential for the global scale where the Eulerian mesh is coarse.
- The number of markers required to track the surface is generally small in comparison to the total number of markers used for other purposes throughout the mantle.
- Lagrangian markers are well-suited for parallelisation, including on GPUs.

In this study, we implement Lagrangian interface tracking in StagYY and introduce two methods: the *bilinear method* and the *integral method*, which differ in how they transfer surface information to the underlying grid. We assess their computational performance and their ability to track a free-surface boundary condition, comparing them with the existing marker-in-cell approach.

2 Limitations of the marker-in-cell method

2.1 General approach

The marker-in-cell method is the approach used to track free surfaces in StagYY and many other geodynamic codes. Lagrangian tracers throughout the model domain track material type (rock or air in the examples here; often many other quantities). To obtain the position of the surface, the tracer distribution is first turned into a compositional field representing air fraction, then the vertical location of the surface is inferred from this air fraction field. Air fraction is calculated as ratio of the number of air tracers to total (rock+air) tracers in a cell, with the numbers of tracers being calculating using one of two sampling (shape) functions, as in Tackley and King (2003).

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

2.1.1 Cell averaging/boxcar averaging

The simplest method of determining the location of the free surface is by first computing a compositional field C by averaging numbers of tracers of each type per cell (Figure 1). A cell containing only rock tracers will have $C_{\text{rock}} = 1$ and $C_{\text{air}} = 0$, a cell containing only air tracers will have $C_{\text{rock}} = 0$ and $C_{\text{air}} = 1$, while a cell containing the surface will have both Cs between 0 and 1. The surface location is then derived from this boxcar averaged field.

While this method enables vertical sub-grid level topographic resolution, doing so accurately requires large numbers of tracers per cell. For example, providing 100 m topography sensitivity on a grid with a vertical resolution of 10 km would require at least 100 tracers per cell. Random noise inherent in the tracers also hinders the ability of this method to accurately track surfaces. As a tracer moved from one cell to another, the composition of both cells changes discontinuously.

2.1.2 Linear shape function averaging

It is possible to overcome some of the limitations of the simple cell averaging method by additionally taking into account the positions of the tracers. This is done by using linear shape function averaging, where the compositional field C is determined for a given cell by weighted averaging of tracers, with this weighting function being linear and centred at the cell centre, resulting in a triangular shape (Figure 1), similar to linear shape functions in the finite element method. By weighting the markers, information about the position of the tracers is also taken into account. Determining the surface location when using this method then requires the solution of a geometrical problem relating to the triangular shape of the shape function, which causes two cells vertically to have C_{air} between 0 and 1. As this is much more accurate than the cell averaging approach, linear shape function averaging is the default surface tracking method used by StagYY and thus what is used in the presented tests.

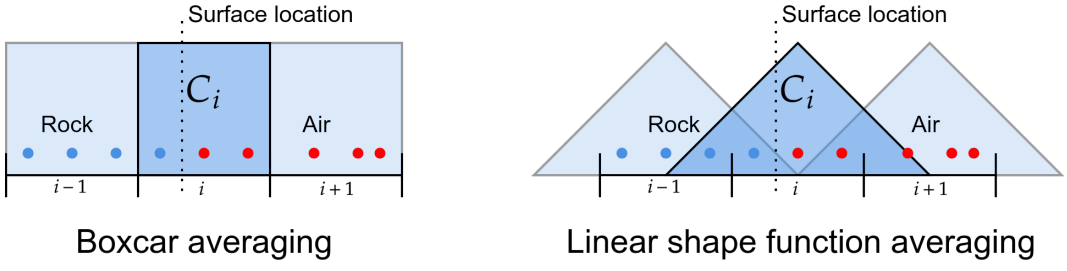


Figure 1. There are two averaging methods used in determining the surface location from a tracer field (illustrated in 1D): boxcar averaging (or cell averaging), and linear shape function averaging. With boxcar averaging, a simple count of rock and air tracers (blue and red) in a given cell is performed, and the location of the surface inferred from C_i. With linear shape function averaging, a weighting function is first applied which includes information on both number and location of tracers in a cell.

Although an improvement on the simple cell averaging approach, linear shape function averaging retains a dependence on number of tracers per cell, which can result in poor performance when modelling small variations in topography.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

2.2 Testing the limitations of the marker-in-cell method

To quantify tracer density–dependent noise in the existing method (where tracer density here means the number of tracers per cell), we use a simple dimensionless 3D Cartesian StagYY model (Figure 2). The viscosity is constant and set to a nondimensional value of $\eta = 1$, convecting with a Rayleigh number of 10^7. Surface topography is expressed as a fraction of the normalised mantle thickness (set to 1). Temperature is also nondimensionalised, ranging from $T = 0$ at the surface to $T = 1$ at the core-mantle boundary. To better illustrate the effect of tracer density, a relatively coarse resolution of 64x64x32 using uniform cubic cells and no surface mesh refinement. Large surface cells highlight the need for sub-grid resolution to capture topography accurately.

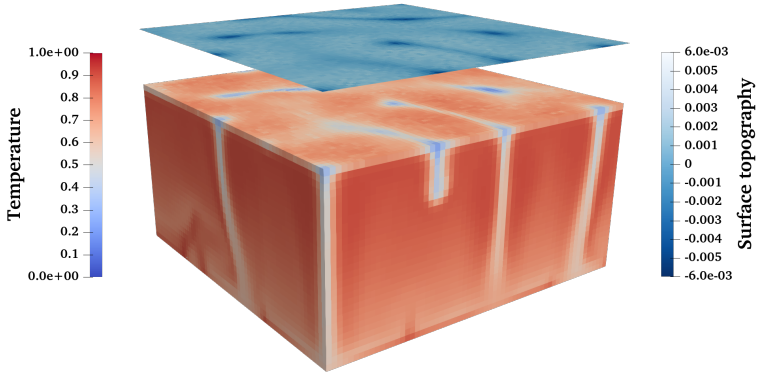


Figure 2. A nondimensional constant-viscosity model ($\mathrm{Ra} = 10^7$) in 3D Cartesian geometry is used to investigate the dependence of the quality of surface topography reconstruction on tracer density (Figure 3). Topography is tracked with tracers using the marker-in-cell method, and a sticky air discretisation with a viscosity contrast of $\eta_{\mathrm{air}}/\eta_{\mathrm{rock}} = 10^{-3}$ is used. A relatively coarse resolution of 64x64x32 is used to better highlight the effect of low marker densities on topography.

After 5,000 time steps, the simple models have reached statistically steady-state convection, and their surface topographies can be compared. Visual inspection of the results at the final time step confirms that low tracer densities (best seen in the 10 or 20 tracer per cell models in Figure 3) result in grainy, unrefined topography, while increasing tracer density results in much smoother topography, which would be expected in a constant viscosity model.

While visually apparent, it is also possible to quantify the noise present at different tracer densities by computing the distribution of surface topography. Figure 4 shows this distribution for a selection of tracer densities as computed by sampling the surface topography of each model at all time steps, excluding the first 200 in order to avoid including the initial warm-up period before steady-state convection is reached. The results show a distribution of topography slightly skewed above the initial level (0.0), which is consistent with visual inspection of Figure 3 that shows the majority of topography is above the initial level with relatively small cold plumes forming deeper topographic lows. The key result is that the distribution is much wider at lower tracer densities, indicating the presence of noise in the topography data. The amount of noise appears to become smaller

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

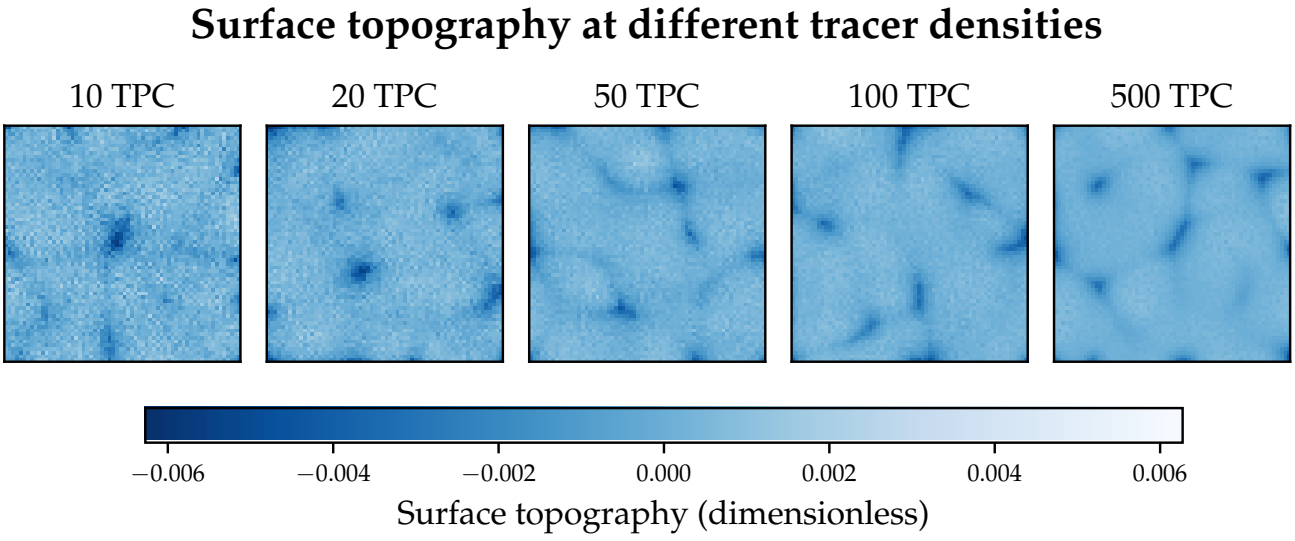


Figure 3. The quality of surface topography using the marker-in-cell method is strongly dependent on number of tracers per cell (TPC). Low tracer densities result in grainy, unrefined topography, with the quality of the topography increasing with tracer density. The magnitude of topography is a dimensionless fraction of the total mantle thickness (normalised to 1).

with increasing tracer density, until the point where it is reasonable to assume the amount of noise is negligible at 500 tracers per cell. Further increases beyond this tracer density differ minimally.

This result demonstrates that a trivial solution to achieve greater topographic resolution is to increase the tracer density. However, this is not always possible in large scale 3D models due to computational limitations. For the same models as above, we compare the computational effort (measured in total computational time), as a function of tracers per cell (Figure 5). Tracers are advected using a 4th-order Runge-Kutta scheme with velocities interpolated to tracer locations using a divergence-free quadratic spline method, while the Stokes and continuity equations are solved using a geometric multigrid solver (Tackley, 2008). These results show a linear dependence of computational effort involved in advecting tracers with tracer density. Memory usage also scales linearly. While this computational trade-off is reasonable for relatively fast and low resolution models such as this one, when scaling to much larger models the computational effort involved in advecting tracers can quickly dominate, which presents a major limitation when attempting to model topography accurately on the global scale. In the models run for Figure 4, tracer computation accounted for 9% of total runtime at 10 tracers per cell, and 77% at 500 tracers per cell, with the 50% crossover point occurring between 100 and 150 tracers per cell.

These findings highlight the limitations of the marker-in-cell method for tracking surface locations. Developing a tracer density independent method is critical for improving topographic accuracy while reducing computational cost in global-scale geodynamic models.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

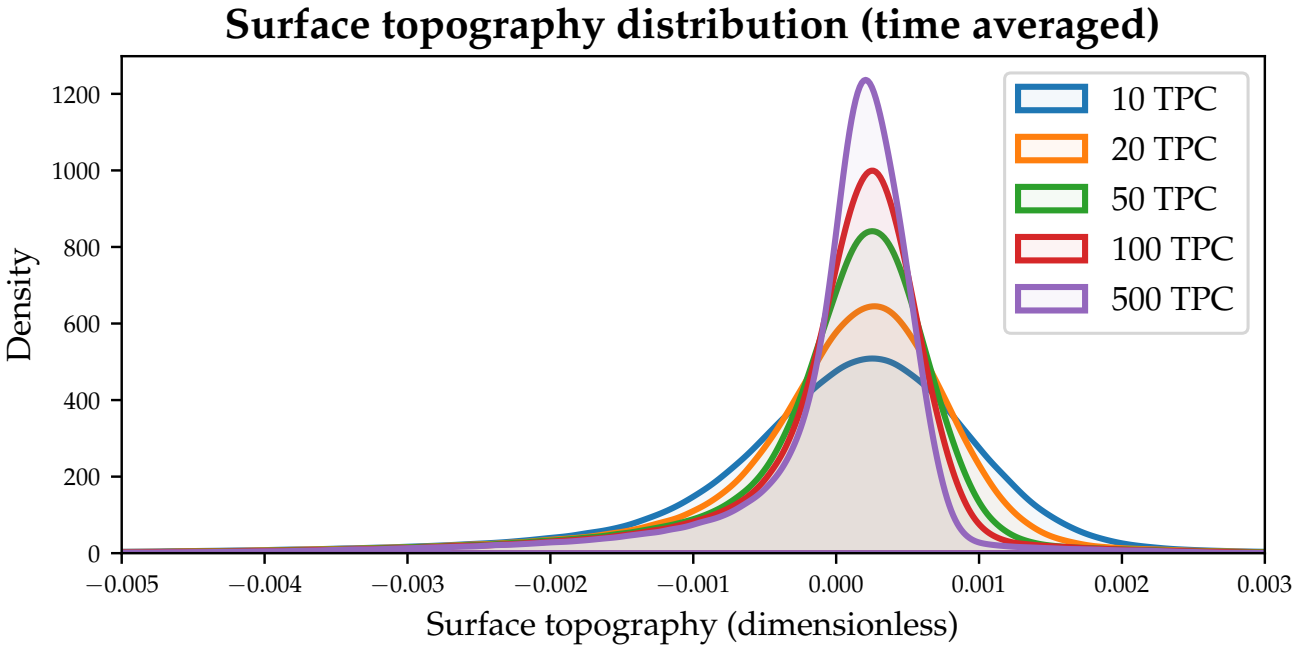


Figure 4. Surface topography probability density function averaged over all timesteps for the five models as shown in Figure 3. At low tracer densities, the distribution is relatively wide indicating that the noise that is more apparent with low marker densities is affecting the quality of the topography generated.

3 Surface tracking

3.1 Marker chain/mesh

In 2D models, the surface can be tracked by a marker chain which is initialised as an ordered array of Lagrangian markers, each of which stores its position $\mathbf{x}_i$ and a type variable to facilitate manipulation of the chain. The position vector for each marker $\mathbf{x}_i$ may be represented in either Cartesian coordinates $\mathbf{x}_i = (x_i, z_i)$, or polar coordinates $\mathbf{x}_i = (r_i, \theta_i)$, and may be dimensional or dimensionless. The height of each marker (z_i or r_i) is set by an initial surface height function $z_{\text{init}}(x)$ or $r_{\text{init}}(\theta)$.

In 3D models the surface representation extends to a mesh. The initial surface consists of markers arranged in a grid with coordinates $\mathbf{x}_i = (x_i, y_i, z_i)$ in Cartesian geometry, or $\mathbf{x}_i = (r_i, \theta_i, \phi_i)$ in spherical geometry. As with the 2D case, the initial height of the markers z_i or r_i is given by an initial surface height function $z_{\text{init}}(x, y)$ or $r_{\text{init}}(\theta, \phi)$.

Surface markers are advected with the velocity field, determined from the Stokes solver, using a spatially fourth-order Runge-Kutta scheme, similar to the treatment of the tracers used in StagYY. Velocities, which are face-centred in the staggered-grid finite-volume discretisation used by StagYY, are interpolated to surface markers using a second-order quadratic spline interpolation method.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

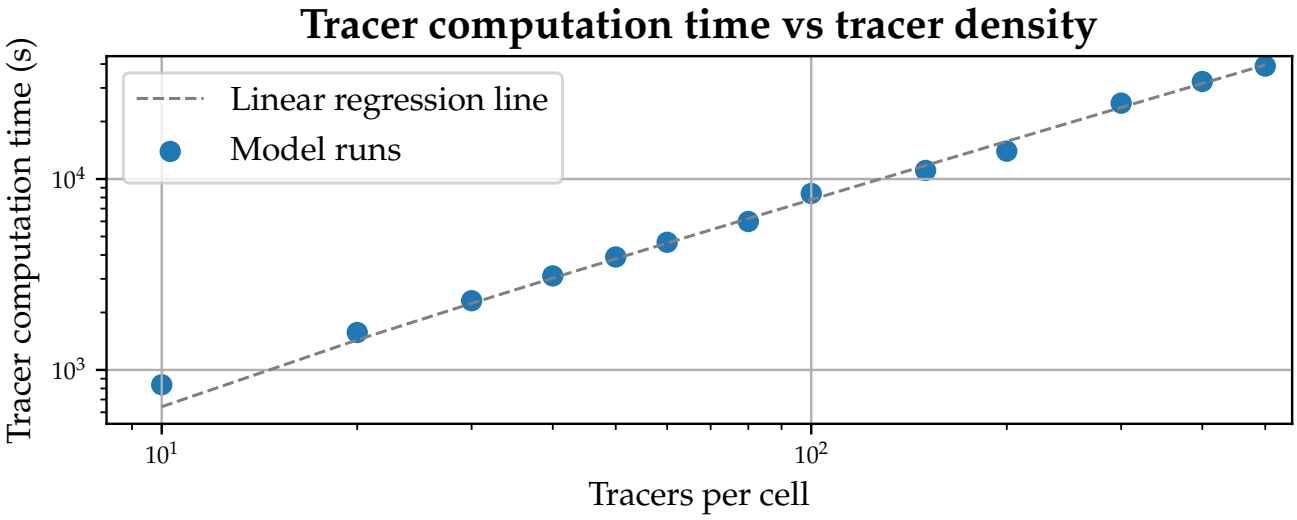


Figure 5. Tracer density (measured in tracers per cell) is linearly correlated with the total runtime of tracer calculations in the model (total CPU time in seconds, 8 CPU cores, AMD Ryzen 9 3900XT). At higher tracer densities, tracer computation time exceeds the computation time of the Stokes solver, becoming the largest component of total runtime. In this example, tracer computation accounts for between 9% of total runtime at 10 tracers per cell, and 77% at 500 tracers per cell.

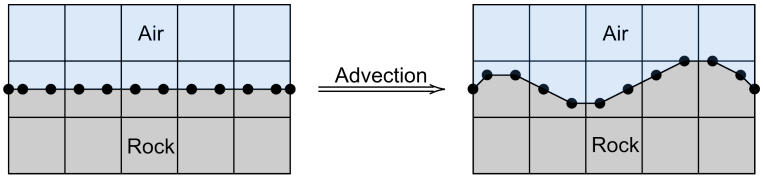


Figure 6. The marker chain divides the domain into regions or rock and air. Lagrangian markers are advected with the velocity field at each timestep to model the evolution of the surface over time.

3.2 Piecewise linear/bilinear surface reconstruction method (bilinear method)

It is necessary to be able to determine the location of the surface as a function of position through interpolation, i.e. finding $z(x)$ or $r(\theta)$ in 2D, or $z(x,y)$ or $r(\theta,\phi)$ in 3D as a function of the position of the Lagrangian surface markers. A straightforward approach is to construct a piecewise continuous linear or bilinear representation of the surface using a first-order accurate distance-weighted linear interpolation (Gerya, 2019). This method was employed with a 2D marker chain in the context of modelling free surfaces in numerical geodynamic models by Duretz et al. (2016), and a similar method is used in 3D models using the regional scale geodynamic code LaMEM (Kaus et al., 2016). The method requires two steps: the interpolation of surface height data from Lagrangian surface markers to nodal points, and the linear or bilinear interpolation of nodal values to arbitrary locations within the domain.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

For consistency, this approach is referred to throughout this chapter as the *bilinear method*, reflecting the bilinear interpolation used in 3D models. In 2D, the process is technically linear, but the term *bilinear method* is retained for uniformity.

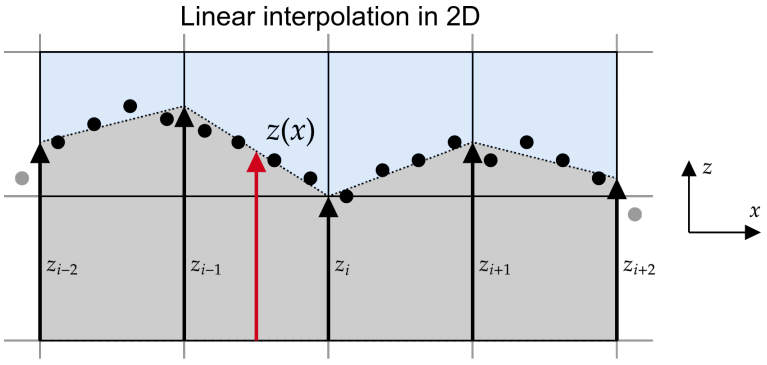


Figure 7. When using the linear/bilinear representation of the surface, the surface markers are interpolated to the nodes of the Eulerian grid through distance-weighted linear interpolation. The surface height at a given point e.g. $z(x)$ (red arrow) may then be computed using linear interpolation.

3.2.1 Interpolating surface markers to nodes

Interpolation of surface heights from markers to the nodes of the Eulerian grid (denoted by z_i) in 2D models is carried out using a 1D distance-weighted linear interpolation (Gerya, 2019):

$$z_i = \frac{\sum_m z_m w_{m(i)}}{\sum_m w_{m(i)}}, \tag{1}$$

$$w_{m(i)} = 1 - \frac{\Delta x_m}{\Delta x}. \tag{2}$$

Where z_m is the height of the m'th surface marker, $w_{m(i)}$ refers to the weight of the m'th surface marker for the i'th node, Δx_m denotes the distance from the m'th surface marker to the i'th node, and Δx denotes the grid spacing. Each node is interpolated using only surface markers found within one grid spacing Δx of the node. In 3D, the method can be extended to determine surface heights at nodes denoted by $z_{i,j}$:

$$z_{i,j} = \frac{\sum_m z_m w_{m(i,j)}}{\sum_m w_{m(i,j)}}, \tag{3}$$

$$w_{m(i,j)} = \left(1 - \frac{\Delta x_m}{\Delta x}\right) \times \left(1 - \frac{\Delta y_m}{\Delta y}\right) \tag{4}$$

Where z_m is the height of the m'th surface marker, $w_{m(i,j)}$ refers to the weight of the m'th surface marker for the i,j'th node, Δx_m and Δy_m denote the x and y distances from the m'th marker to the i,j'th node, and Δx and Δy denote the grid spacing.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

When using spherical coordinates, r, θ, and ϕ may be directly substituted for z, x, and y in the above to obtain the surface height at nodal points r_i or $r_{i,j}$.

In both 2D and 3D, the performance of this operation scales linearly $O(n)$ with respect to the number of surface markers n.

3.2.2 Linear/bilinear interpolation

In the second step of surface reconstruction, the location of the surface at arbitrary points in the domain can be determined by interpolating between nodal values. This is achieved by using linear interpolation in 2D, or bilinear interpolation in 3D (Equations 5, 6).

In 2D, to interpolate the surface height at a point $z(x)$, the surface heights at the neighbouring nodes to the left and right are used:

$$z(x) = \frac{1}{\Delta x}\left(z_i(x_{i+1} - x) + z_{i+1}(x - x_i)\right), \tag{5}$$

Where x_i and x_{i+1} represent the x coordinate of the nodes immediately to the left and right of the point x, z_i and z_{i+1} are the surface heights at those nodes, and Δx is the grid spacing.

In 3D models, four nodes which form a quadilateral surrounding the point of interest must be used to construct a bilinear interpolation of the surface location $z(x, y)$, with their locations denoted by (x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), and (x_{i+1}, y_{j+1}). The bilinear interpolation of the surface is then given by:

$$z(x,y) = \frac{1}{\Delta x \Delta y}(z_{i,j}(x_{i+1} - x)(y_{j+1} - y) + z_{i+1,j}(x - x_i)(y_{j+1} - y) + z_{i,j+1}(x_{i+1} - x)(y - y_j) + z_{i+1,j+1}(x - x_i)(y - y_i)). \tag{6}$$

Again, when using spherical coordinates, r, θ, and ϕ may be substituted directly for z, x, and y in the above to obtain the surface height at nodal points $r(\theta)$ or $r(\theta, \phi)$.

For both 2D and 3D cases, this operation is independent of the number of surface markers, and can be accomplished in constant $O(1)$ time.

3.3 Direct interpolation in 2D (integral method)

Another method of determining the location of the surface $z(x)$ or $r(\theta)$ in 2D at arbitrary points within the domain is through directly interpolating the surface location from the marker chain. This can be achieved by locating the markers on the chain that are immediately to the left and right of the interpolation point, and then linearly interpolating the surface height between these two points.

In Cartesian coordinates, linear interpolation of $z(x)$ at the point x located between the surface marker lying immediately to the left $\mathbf{x}_i = (x_i, z_i)$ and right $\mathbf{x}_{i+1} = (x_{i+1}, z_{i+1})$ of x is accomplished using:

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

$$z(x) = \frac{z_i(x_{i+1} - x) + z_{i+1}(x - x_i)}{x_{i+1} - x_i}. \tag{7}$$

It is possible for higher order interpolation methods to be used to interpolate the marker chain such as a piecewise quadratic interpolation, or cubic spline interpolation. In practice however, adequate results can be achieved with a simple linear interpolation without recourse to higher order methods.

3.3.1 Sorting the marker chain in 2D

In order to directly interpolate on the chain, it is necessary to locate the markers on the chain immediately to the left and right of a given point. In order to efficiently do this, a $O(\log n)$ binary search can be used which requires the chain to be sorted by the horizontal coordinate. As advection of the surface markers allows them to move laterally, it is possible for the chain to become disordered over time, necessitating a sorting step after every timestep. As the chain is often fully or nearly fully sorted after advection, adaptive sorting algorithms such as insertion sort are well-suited for maintaining the order of the chain.

3.4 Direct interpolation in 3D (integral method)

Direct interpolation of the marker mesh in 3D is more difficult, as the surface is no longer represented with lines, but with planes defined by the points of the marker mesh. As the surface markers are generally unstructured, this necessitates the use of a meshing algorithm such as Delaunay triangulation.

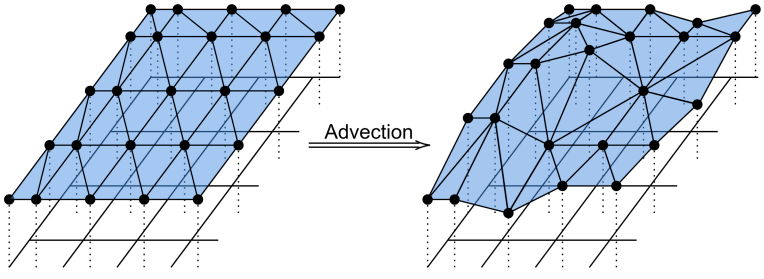


Figure 8. In 3D, Lagrangian markers are advected with the velocity field, and the Delaunay triangulation recomputed at each timestep in order to create a mesh which continuously tracks the surface.

The Delaunay triangulation is computed in 2D using the horizontal (x_i, y_i) coordinates of each marker, not including the height coordinates z. In spherical geometry it is computed using (θ_i, ϕ_i) coordinates for simplicity. This leads to minor errors in the Delaunay triangulation from projecting the spherical patch into a Cartesian plane, but these errors are negligible due to the limited range of θ used by the yin-yang grid. Additionally, the triangulation itself does not have to satisfy the Delaunay criteria, as any triangulation of the surface marker mesh is sufficient for the purposes of this study. Methods for performing Delaunay triangulations on spheres also exist, for example Renka (1997).

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Computing the Delaunay triangulation of n surface markers can be performed in $O(n \log n)$ time using an efficient divide and conquer method as employed by GEOMPACK2, while the naive incremental method takes $O(n^2)$ (Joe, 1991).

The Delaunay triangulation must be computed at each time step to preserve the properties of the triangulation as markers may move relative to one another due to advection. Algorithms for adaptive Delaunay triangulation exist which are able to exploit the fact that typically after an advection step, the existing triangulation will be close to the Delaunay triangulation (Phongthanapanich and Dechaumphai, 2004).

3.4.1 Interpolation of the mesh

An efficient $O(\log n)$ method exists in GEOMPACK2 for looping through a Delaunay triangulation to locate the (2D) triangle that encloses a given point (x, y) if such a point exists (Joe, 1991). This Delaunay triangle corresponds to three markers in 3D, $\mathbf{a}, \mathbf{b}$, and $\mathbf{c}$, which together define a plane.

This plane can be defined through the equation $\mathbf{n} \cdot \mathbf{x} = \alpha$, where $\mathbf{n}$ is a normal vector and α is a scalar value. The normal vector $\mathbf{n}$ can be derived from the three points as:

$$\mathbf{n} = (\mathbf{b} - \mathbf{a}) \times (\mathbf{c} - \mathbf{a}). \tag{8}$$

With some rearrangement, the equation of this plane allows for determination of surface height $z(x, y)$ as a function the three points in the triangulation (as shown in Figure 9):

$$z(x, y) = \frac{n_x(a_x - x) + n_y(a_y - y)}{n_y} + n_y a_y. \tag{9}$$

Similarly, in spherical geometry $r(\theta, \phi)$ can be given by:

$$z(\theta, \phi) = \frac{n_\theta(a_\theta - \theta) + n_\phi(a_\phi - \phi)}{n_\phi} + n_\phi a_\phi. \tag{10}$$

If the point of interest (x, y) lies outside the Delaunay triangulation, extrapolation of the nearest Delaunay triangle is used using the same method with the nearest triangle to the point.

3.5 Interpolating to the Eulerian grid

In order to couple the surface tracking method to the Stokes solver and to output the results, it is necessary to interpolate the surface heights of the marker chain/mesh to the Eulerian grid. When using the bilinear method, this step is handled automatically by interpolating surface heights to the nodes of the Eulerian grid. When using the integral method, there are several possible methods for interpolating to the grid. In this study, a cellwise integration approach was used in both 2D and 3D.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

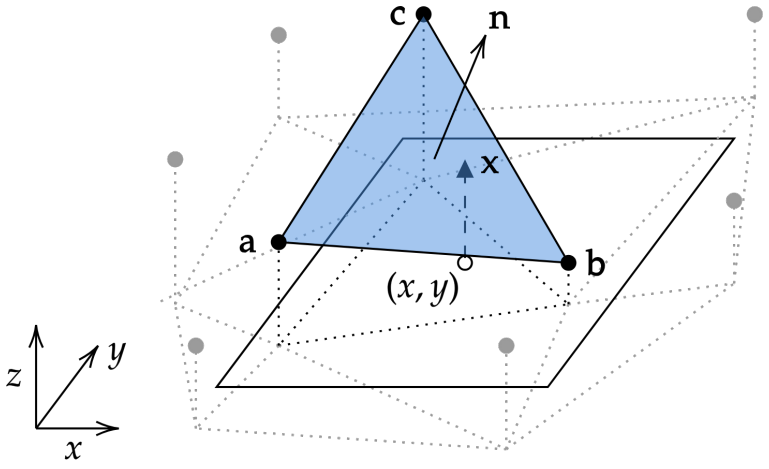


Figure 9. Interpolation of the marker mesh within one triangular surface mesh element in 3D Cartesian geometry.

3.5.1 Cellwise integration

The cellwise integration method is to compute the area (in 2D) or volume (in 3D) enclosed by the surface within the computational cell, and to then compute the equivalent topographic height in that cell that preserves this area or volume. This is physically motivated by the need to maintain conservation of mass of both rock and air within the model.

In 2D, the surface height z_i is a given cell is given as a function of surface location $z(x)$ by:

$$z_i = \frac{1}{\Delta x} \int_{x_{i,l}}^{x_{i,r}} z(x)\, dx, \tag{11}$$

Where $x_{i,l}$ and $x_{i,r}$ are the left and right horizontal boundaries of the i'th cell, and $\Delta x = x_{i,r} - x_{i,l}$ is the grid spacing. The numerical integration of $z(x)$ is performed using the trapezium rule by looping over all the intervals of points in the marker chain, computing the area under each interval, and adding that area to the correct cells before dividing by grid spacing to obtain the surface height. As the marker chain is already sorted by horizontal coordinate, the integration can be accomplished in linear time ($O(n)$) and with no additional array allocations, and also works where the chain is sparse.

A similar concept can be applied for 3D models by evaluating the volume underneath the surface. In 3D Cartesian coordinates, the surface height in the i, j'th cell z_i, j is given by:

$$z_{i,j} = \frac{1}{\Delta x \Delta y} \iint_{C_{i,j}} z(x,y)\, dA, \tag{12}$$

Where $C_{i,j}$ denotes the domain of the i, j'th cell.

Numerically computing the integral can be accomplished by looping through the list of Delaunay triangles, and then applying a polygon intersection algorithm to compute the intersection between the current triangle and footprint of each column of cells

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

(Figure 10). Polygon intersections can be computed efficiently in linear time using an algorithm such as the Sutherland-Hodgman algorithm or O'Rourke et al. (1982). Once the intersection polygon has been determined, it can be split up into triangles with vertices $\mathbf{a}$, $\mathbf{b}$, and $\mathbf{c}$, the heights recomputed from the equation of the plane through the original three points, and the volume of each triangular prism can be computed. Finally, the sum of all triangular prisms in each column of cells can be summed up and divided by the basal area of the column of cells to obtain the surface height in each column.

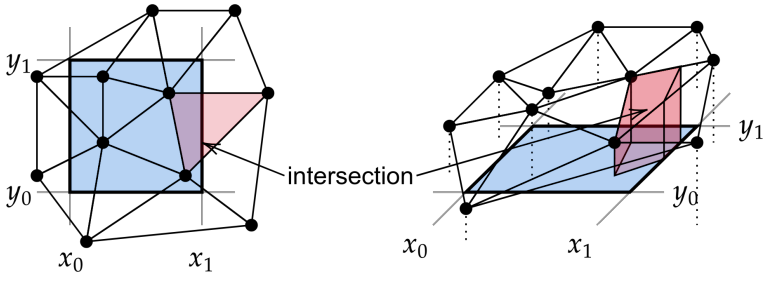


Figure 10. Piecewise integration under the Delaunay mesh using a polygon clipping algorithm is used to compute the volume enclosed by the surface within a given computational cell.

This operation can be accomplished in linear time $O(n)$ with respect to the number of surface markers by simply looping through all Delaunay triangles, albeit with some overhead associated with the need to use polygon intersection.

3.6 Parallelisation

When solving on multiple CPU cores, the model domain is split up into a number of parallel subdomains. These subdomains are not independent of one another, and certain operations such as passing tracers or fields from one subdomain to the other require the use of Message Passing Interface (MPI).

For the purposes of interpolation of surface markers, it is not sufficient to use the existing treatment used for tracers in the model, as this could potentially leave a region within the domain where the surface is not defined. It is therefore necessary to exchange more tracers in order to create an overlap region for the purposes of this interpolation. By adapting existing methods for passing regular tracers to exchange this overlap region, it is possible to form this overlap region such that the surface is defined at all locations within the domain for all subdomains (Figure 11).

4 Stabilisation

When implemented naively, the surface marker chain/mesh does not satisfy the criterion of being stable over multiple timesteps. There are three main reasons for this. Firstly, while the solution to the Stokes equations ensure conservation of mass, the surface marker chain/mesh itself is not and may shift up and down over time, causing errors in mass conservation. Secondly, regions of divergence can cause the surface markers to become widely spaced, compromising the accuracy of the method. Thirdly,

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

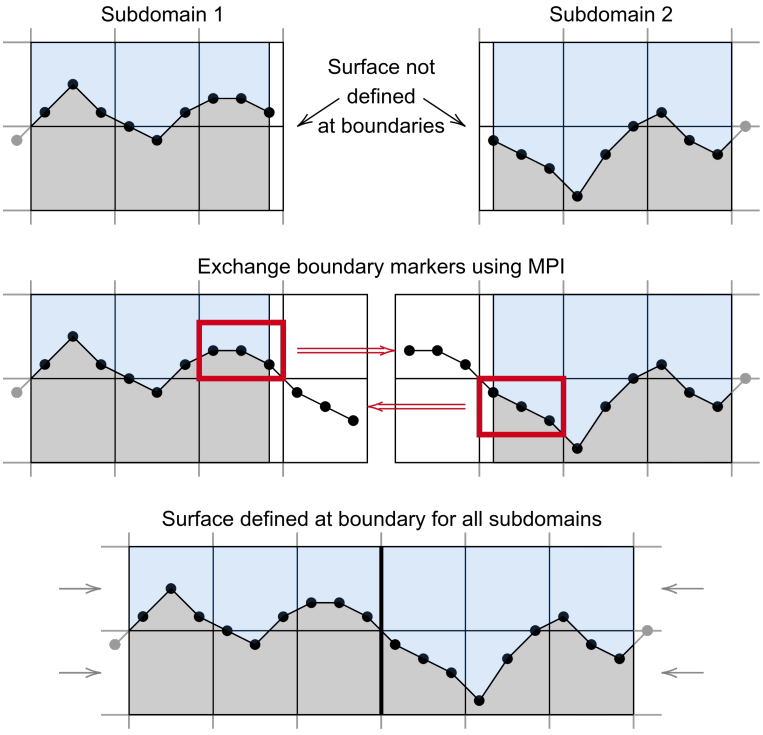


Figure 11. Parallelisation of the marker chain/mesh involves the use of an overlap region which is communicated between subdomains. As the surface itself is not necessarily defined at subdomain boundaries when using markers within the same subdomain, a one cell wide overlap region is taken from the adjacent subdomain. Communication of surface markers between subdomains is achieved using MPI. This approach is also applicable to 3D cases.

regions of convergence can cause entrainment of markers into the mantle, a feature which is unphysical and must be corrected. There are several ways that these stability issues can be addressed.

4.1 Surface height adjustment

While total mass of rock and air is supposed to be conserved in the numerical model, a fact that should be reflected in the solution to the Stokes equations, it is possible for minor errors to accumulate, which may shift the surface up or down over time. The first stabilisation method adjusts the height of the surface markers to preserve the total area or volume of rock and air in the model, and therefore mass assuming near incompressibility of the rock. This can be achieved by integration under the marker chain/mesh.

4.1.1 Integral correction

As the distribution of surface markers is not necessarily even, a more accurate method of ensuring that the volume of rock and air in the model remains the same is by computing the total area/volume under the surface through integration.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

In 2D, the integral correction takes a similar form to that presented for cellwise integration of volume fractions in Section 3.5.1.

$$z_{i,\text{new}} = z_i - \frac{1}{L_x}\int_{x_L}^{x_R} z(x)\,dx - z_{\text{init}}, \tag{13}$$

where x_L and x_R denote left and right boundaries of the domain, respectively, and L_x is the total domain width. Similarly to the situation for cellwise integration, under the assumption of surface displacements being small a simplified approximate form of the integral term can be derived, resulting in an expression for the integral surface correction in spherical geometries:

$$r_{i,\text{new}} = r_i - \frac{1}{L_\theta}\int_{\theta_L}^{\theta_R} r(\theta)\,d\theta - r_{\text{init}}. \tag{14}$$

With left and right domain boundaries θ_L and θ_R, and domain width (in terms of θ) L_θ. In both Cartesian and spherical geometries this integral can be evaluated analytically in linear time when using a sorted marker chain.

These integrals can easily be obtained in linear time by looping through each interval on the marker chain or nodes and adding up the total area underneath the surface using the trapezium rule.

When extending the method to 3D, a similar method can be used to the above, however the required integral becomes a volume integral over the entire domain, which is then divided by the total area of the domain. These volume integrals in Cartesian and spherical coordinates are given by:

$$V = \iint_D z(x,y)dA \quad \text{in Cartesian coordinates,} \tag{15}$$

$$V = \iint_D r(\theta,\phi)dA \quad \text{in spherical coordinates} \tag{16}$$

Which should remain constant for all timesteps to ensure that volume conservation is achieved.

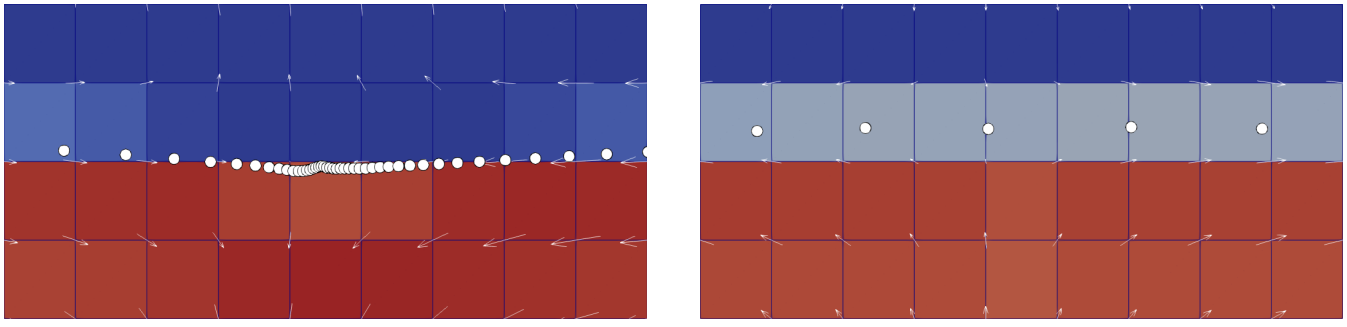
While in Cartesian coordinates the infinitesimal area dA is simply given by $dA = dy\,dx$, in spherical coordinates the area correction must be derived from the definition of an infinitesimal surface element (Kreyszig, 2010):

$$dA = \left\|\frac{\partial \mathbf{r}}{\partial\theta} \times \frac{\partial \mathbf{r}}{\partial\psi}\right\| d\theta\,d\phi = \left|r\hat{\boldsymbol{\theta}} \times r\sin\theta\hat{\boldsymbol{\phi}}\right| = r^2\sin\theta\,d\theta\,d\phi, \tag{17}$$

$$\text{where } \mathbf{r} = \begin{bmatrix} r\sin\theta\cos\phi \\ r\sin\theta\sin\phi \\ r\cos\theta \end{bmatrix}, \tag{18}$$

Where setting $r = R$, the radius of the planet, obtains the infinitesimal area near the surface.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026



(a) Convergence (b) Divergence

Figure 12. Without reinitialisation, regions of convergence experience bunching of the surface markers, while in regions of divergence the marker density becomes low. These effects must be corrected using reinitialisation to maintain surface marker density.

These integrals are evaluated similarly to the cell integration method presented in Section 3.5. When using a Delaunay triangulation, it is possible to loop over all Delaunay triangles using a polygon intersection algorithm to isolate the parts of triangles which are within the modelling domain, and then to compute the volume of each triangular prism underneath. A correction for variable dA can then be applied for the case of spherical geometry. This is relatively efficient, and can be evaluated in linear time with respect to the number of Delaunay triangles.

When using the bilinear method, the integral can be easily evaluated using a weighted sum of the nodal surface heights, with corrections for varying dA as needed.

This method runs in linear time ($O(n)$) with respect to the number of surface markers.

Although reinitialisation is essential for preserving the integrity of the marker chain over multiple time steps, it does come with trade-offs. Specifically, the process may degrade information about the surface geometry and introduce artificial numerical diffusion. Therefore, reinitialisation should only be applied when needed to avoid excessive smoothing.

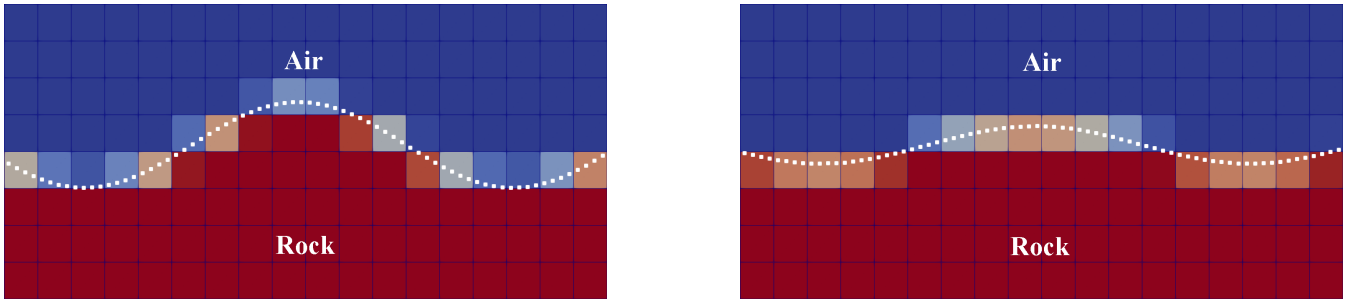
4.1.2 Periodic global reinitialisation

One of the simplest reinitialisation strategies is periodic global reinitialisation. In this approach, the entire surface marker chain/mesh is reinitialised to align with the current surface level using one of the interpolation methods described earlier or based on the Eulerian representation of the surface.

Periodic global reinitialisation is computationally efficient, as it maintains a consistent number of surface markers. However, a significant drawback is the loss of surface information and the introduction of numerical diffusion into the model. For instance, global reinitialisation can overly smooth surface features, as can be demonstrated when using a test case with an initially sinusoidal surface (Figure 13). In this test, the markers are not advected with the velocity field, but are smoothed at each timestep due to the use of periodic global reinitialisation at each timestep.

Despite these limitations, periodic global reinitialisation of surface markers does not require the marker chain/mesh to be previously sorted (or the Delaunay triangulation to be computed), making it suitable for use in conjunction with the bilinear

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026



(a) Timestep 1 (b) Timestep 20

Figure 13. Illustration of the smoothing effect caused by global periodic reinitialisation over 20 timesteps using bilinear interpolation. There is no coupling between the surface marker chain and the underlying model; all deformation of the surface is caused by numerical diffusion resulting from global reinitialisation.

interpolation method. Additionally, while it can reduce the fidelity of some small scale topographic variations, some level of smoothing may be beneficial, as it would otherwise have to be introduced deliberately to maintain surface stability. However, balancing this smoothing effect with the need to preserve surface details remains a challenge.

Global reinitialisation can be accomplished in linear time $O(n)$ when used in conjunction with the bilinear method, and $O(n \log n)$ when used in conjunction with direct interpolation methods due to the $O(\log n)$ time required to sample the surface height at a given location with those methods.

4.1.3 Interval reinitialisation

Interval based reinitialisation in 2D is based on the interval between adjacent markers. If the interval exceeds the maximum interval size, which is a user-defined limit proportional to the original marker spacing Δx_{init}, a marker is added to the chain at the midpoint of the interval. If the interval is too small, the two markers that make up the interval are merged into one at the midpoint of the interval. It is desirable to have an upper threshold no greater than 2 to ensure that markers are added at the same or greater density as the initial starting density, and a lower threshold sufficiently small to allow markers to bunch up in regions where it is needed subduction zones, but not so large as to introduce large numbers of new markers into the model. Empirical testing found that the thresholds $0.2 \leq \Delta x / \Delta x_{\text{init}} \leq 2$ provides good results.

In 3D, a similar concept is applied to the areas of the Delaunay triangles. In cases where the ratio of triangle area to initial A/A_{init} exceeds a certain amount, a new marker is introduced at the midpoint in order to subdivide the triangle, while in cases where it is small the three points of the triangle are merged into one at the midpoint. To ensure that markers are added at the same density as the initial density, the upper threshold should be no greater than 3, as 3 new triangles are created from each triangle that is subdivided. Likewise the lower threshold should be sufficiently small to enable high resolution but not so small as to introduce large numbers of new markers, affecting performance. Empirical testing found that a range of $0.2 \leq A/A_{\text{init}} \leq 3$ provides good results.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Interval based reinitialisation has the benefit of not modifying the marker chain where it is unnecessary, thus avoiding the numerical diffusion inherent in global or cellwise reinitialisation schemes. It is relatively efficient; iterating through the markers to check the intervals takes linear time $O(n)$, while inserting a single new marker generally also takes $O(n)$. A downside is that there is an increased upper bound on the number of markers that can be introduced into the model, requiring more memory and computational resources, however, this can be controlled by modifying the lower interval bounds. Additionally, it requires either sorting or Delaunay triangulation of the surface markers, which precludes its use in conjunction with the bilinear method.

4.2 The need for smoothing

A fundamental problem with the Lagrangian surface marker method is the undesirable entrainment of markers into the mantle or air layer instead of remaining at the surface. The effect is most pronounced when using the integral method, and results in the formation of unrealistically deep trenches or high mountains that are not physically reasonable and can result in numerical instabilities in the model (Figure 14). Solving this problem requires a method of smoothing the Lagrangian markers.

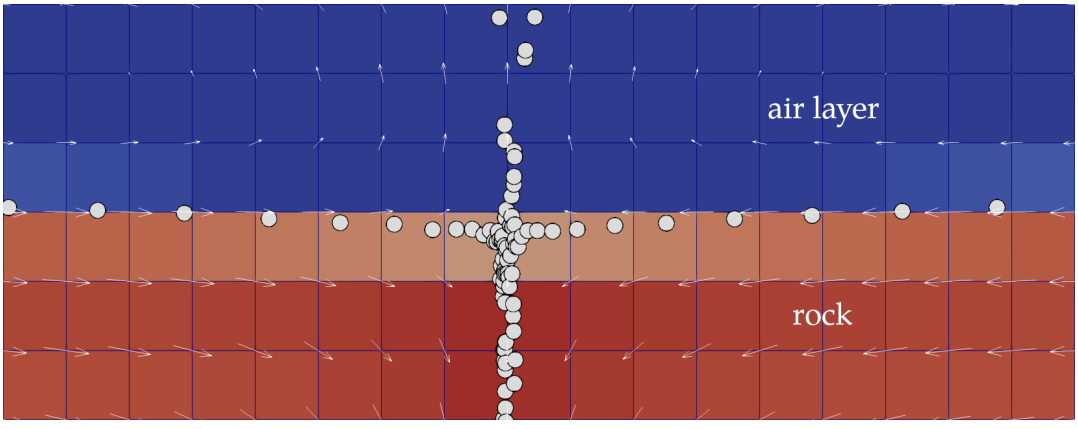


Figure 14. In the absence of an appropriate smoothing algorithm, surface markers can become entrained within rock (red) or the sticky air layer (blue), causing significant numerical instability.

A natural choice for smoothing surfaces is a diffusion process. Two key characteristics of diffusion make it a logical choice: it is a conservative process, ensuring that the total amounts of rock and air in the model remain constant, and it is physically motivated, with the material flux being proportional to the slope. However, this approach can add diffusion to the surface unnecessarily, where model features are smoothed out over time as a result of diffusion of the marker chain, as opposed to any physical process. As noted above, periodic global reinitialisation, if used, already results in some numerical diffusion.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

4.3 Slope limiting

In an attempt to overcome the limitations of pure diffusion based smoothing, alternative methods of locally smoothing the marker chain/mesh were investigated. The method implemented for smoothing when using the integral method is a novel 'slope limiting method'.

The physical motivation for the slope limiting method comes from examining the slope of piles of loosely packed materials such as sand. A sand pile under the influence of gravity will naturally form a conical shape with maximum angle of repose ϕ which is related to its internal coefficient of static friction μ_s by:

$$\tan(\phi) = \mu_s. \tag{19}$$

The angle of repose ϕ in typical materials (such as sand) is around 30 - $40°$. Materials that are sloped below the angle of repose are stable, while materials which are above are unstable and will erode until the angle is reached. Cohesion of the material is not considered in this model.

4.3.1 Governing equation

The behaviour described above can be expressed as a partial differential equation based on the diffusion equation:

$$\frac{\partial z}{\partial t} = k \frac{\partial^2 z}{\partial x^2} H\left(\frac{\partial z}{\partial x} - s_{\text{crit}}\right), \tag{20}$$

where H is the Heaviside step function, s_{crit} is the critical slope, and k is a diffusion coefficient. The effect of the Heaviside function is to ensure that regions where the slope is below the critical threshold s_{crit} are not modified. In the implementation, only the steady state form of this equation is considered, which is given by:

$$\frac{\partial^2 z}{\partial x^2} H\left(\frac{\partial z}{\partial x} - s_{\text{crit}}\right) = 0. \tag{21}$$

This equation extends to 3D by replacing the second derivative of z with $\nabla^2 z = \partial^2 z/\partial x^2 + \partial^2 z/\partial y^2$, and the first derivative with the norm of the gradient. The equation becomes:

$$\nabla^2 z \, H\left(||\nabla z|| - s_{\text{crit}}\right) = 0. \tag{22}$$

The Heaviside function introduces a nonlinearity to the otherwise linear diffusion equation, which somewhat increases the difficulty of the solution. Iterative methods are able to solve the problem in both 2D and 3D.

The principal effect of slope limiting is to cap gradients at a fixed value. Applied to simple sinusoidal profiles in 2D and 3D (Figures 15 and 16), this flattens the wave sides where gradients peak, mimicking the behaviour of piles of sand. A key advantage is that slope limiting leaves regions unaffected where the gradient does not exceed s_{crit}. This limits the effect to specific

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

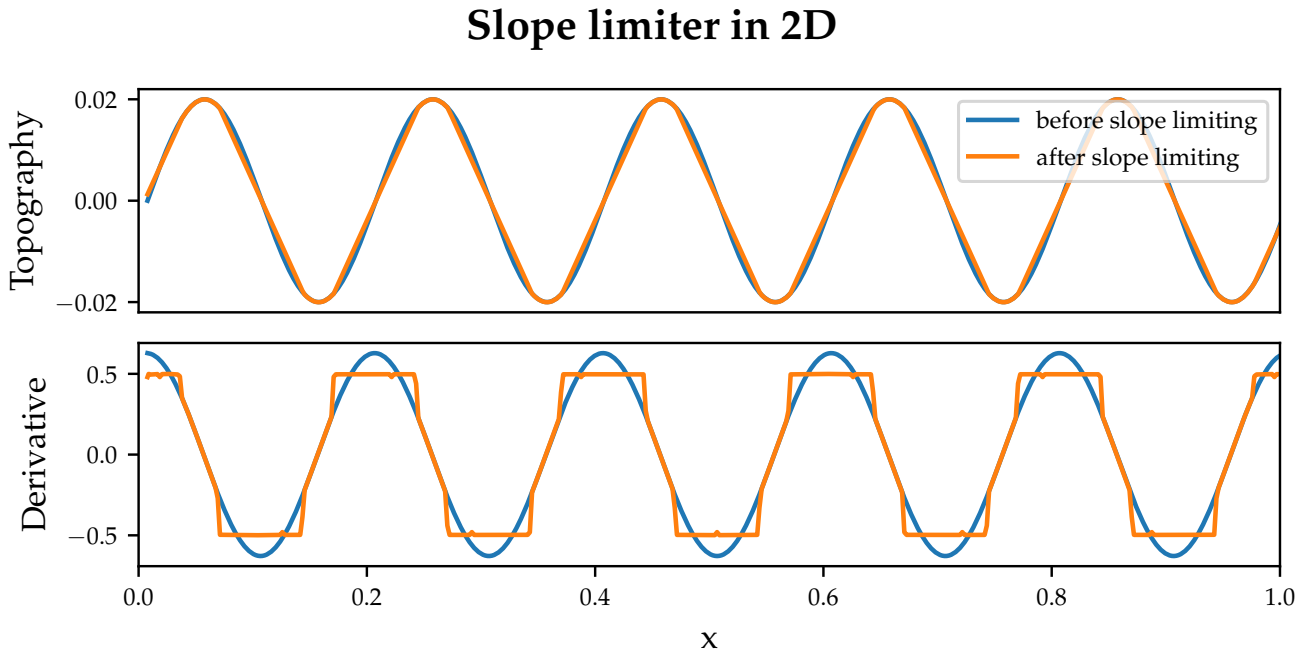
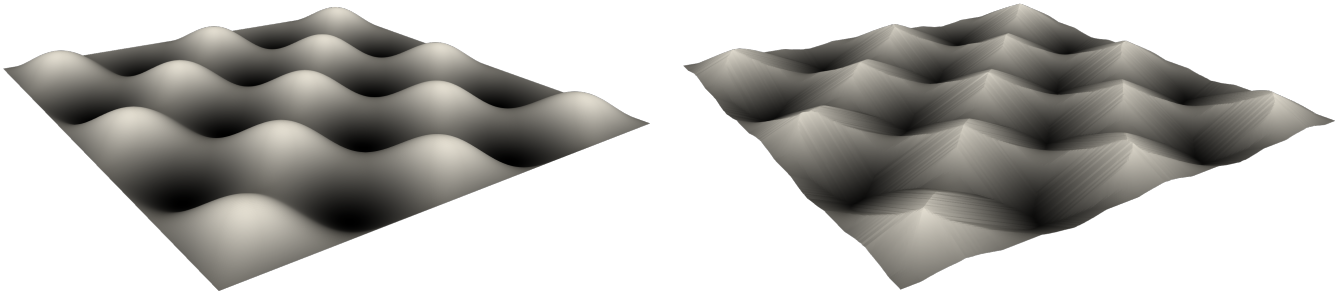


Figure 15. Slope limiter in 2D showing topography (top) before and after the slope limiter has been applied to an initial surface $z = 0.02\sin(5\pi x)$, and its derivative (bottom) showing the limiting of the slope to a maximum of $s_{\text{crit}} = 0.5$



(a) Before slope limiting

(b) After slope limiting

Figure 16. A test on a Cartesian grid showing the effect of the slope limiting algorithm with $s_{\text{crit}} = 0.5$ in 3D on an initial surface $z = 0.05\sin(5\pi x)\sin(5\pi y)$, where $x, y \in [0, 1]$.

regions of high deformation, such as subduction zones or collision zones, where smoothing is necessary to ensure stability. This approach also improves performance when the surface is minimally deformed, as gradient checking can be accomplished in linear time. Additionally, slope limiting is timestep independent, unlike diffusion smoothing or bilinear methods, meaning that it will produce the same results over a variety of different temporal and spatial scales.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

4.3.2 Limitations of slope limiting

A key limitation of the slope limiting algorithm is its relatively poor performance in both 2D and 3D, as can be seen in the performance results presented in Figures 26 and 37.

Another limitation common to all methods of smoothing a surface marker chain/mesh is that it only modifies the location of the surface, and not the underlying tracers that may exist in the model. A result of modifying the surface location independently of the actions of the velocity field is that tracers can become unmixed. Solutions to this problem are discussed in Section 5.1.

5 Coupling

Although it is useful to track the location of the free surface by itself, the utility of the method can be increased by coupling the representation of the surface to the mechanical solver. There are two methods of coupling the model to the solver: indirectly through manipulation of the existing Lagrangian tracers in the model (which then determine the density and viscosity fields), and directly through modification of the density and viscosity fields through the use of volume fraction functions.

5.1 Tracer unmixing

During advection, it is possible for rock tracers to end up above the surface, and likewise for air tracers to end up below the surface. This situation can also occur if the location of the surface is directly manipulated, as it is when applying a smoothing method to the surface. This behaviour is undesirable as it leads to numerical instability, and must be corrected by an appropriate method of unmixing erroneous tracers. Two methods were implemented to unmix tracers near the surface, tracer bouncing and tracer exchange (Figure 17).

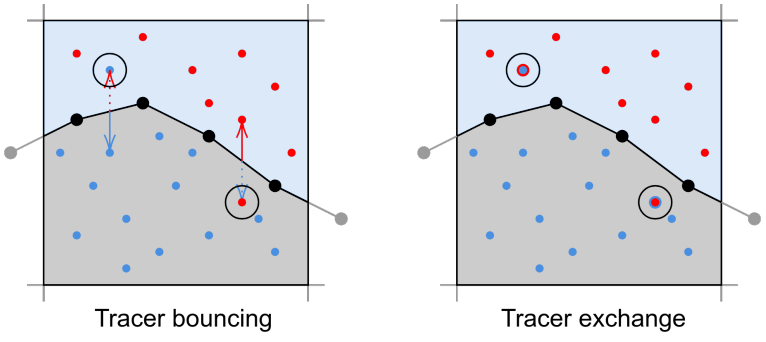


Figure 17. Two methods of dealing with tracers which are erroneously on the wrong side of the surface. Air tracers (red) should remain above the surface, while rock tracers (blue) should remain below. In the bouncing method, erroneous tracers (circled) are bounced back to their side, while with the exchange method they are directly converted to the opposite type.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

5.1.1 Tracer bouncing

A simple way of unmixing tracers is to use "bouncing", where erroneous tracers are simply "bounced" back to their correct side. In 3D, the bouncing distance is the distance to the surface at the x, y position of the tracer:

$$z_{\text{new}} = \begin{cases} z(x,y) + |z_{\text{old}} - z(x,y)| & \text{if air tracer below surface} \\ z(x,y) - |z_{\text{old}} - z(x,y)| & \text{if rock tracer above surface} \end{cases} \tag{23}$$

A downside of the bouncing method is that it is possible to create gaps where there are no tracers in cases where there is considerable deformation of the surface without motion of the underlying tracers. This problem is particularly apparent in convergent regions, where the surface position is modified through smoothing algorithms but the underlying tracers are not. Over time, these gaps can cause severe numerical instability due to the low tracer densities. However, as no tracers are created or destroyed by this method, it conserves total tracer mass and is applicable to cases where many different tracer types are used.

5.1.2 Tracer exchange

Tracer exchange involves exchanging tracers on the opposite side of the surface for the appropriate type. Air tracers erroneously below the surface are directly exchanged for rock tracers, and rock tracers erroneously above the surface are exchanged for air tracers. This method avoids the possibility of large gaps forming near the surface due to tracers being bunched together by the bouncing method.

While exchanging rock tracers into air tracers is trivial as there is only one type of air tracer, exchanging air tracers to rock tracers require a choice of which type of rock tracer to exchange. The simplest and most efficient choice is to select a random template rock tracer from the cell below the free surface. While this method works well for simple model setups where the number of unique compositions is small and tracers are not used to advect physical properties such as temperature and viscosity, for more complicated setups this method becomes unsuitable due to the potentially large number of potential templates.

A further caveat to this method is the fact that it is not strictly volume conserving due to directly creating and destroying tracer mass through creating and destroying rock and air tracers. On average, however, the number of rock tracers exchanged for air and air tracers exchanged for rock remains roughly equal, and so total tracer mass is preserved on average, if not on the small scale.

5.2 Density and viscosity coupling

StagYY typically uses the marker-in-cell method to calculate the density and viscosity fields, which are essential inputs for the Stokes solver. However, with Lagrangian surface tracking, it is possible to achieve greater accuracy by directly coupling the surface location to these fields.

The ability to do so requires the use of volume fractions, denoted by ϕ. These scalar variables are defined as the fraction of material in a given control volume, with the material for geodynamic applications being rock. The volume fraction function is

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

therefore 0 if the control volume is completely empty, and 1 if it is completely full (Figure 18). In the vicinity of the surface, partially filled cells can be found in which $0 < \phi < 1$.

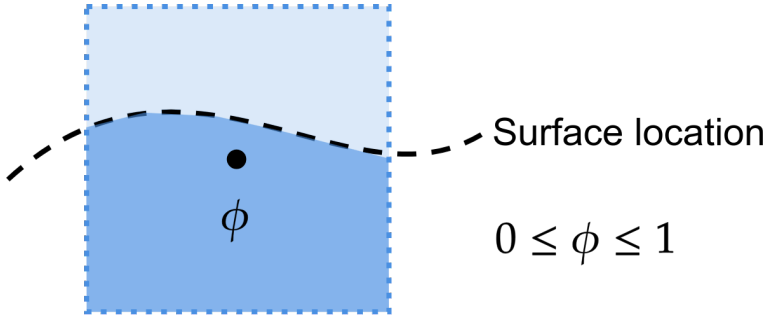


Figure 18. The volume fraction function ϕ for a given control volume is defined as 0 if the control volume is completely empty, and 1 if it is completely full.

StagYY uses a fully staggered grid, where variables are located at different points within the cells (unlike collocated grids). The staggered grid layout in 2D and 3D is illustrated in Figure 19 and is a standard layout in geodynamic modelling (e.g. (Patankar, 1980; Ogawa et al., 1991; Gerya, 2019)). Consequently, volume fractions for specific variables correspond to control volumes offset from the main cells.

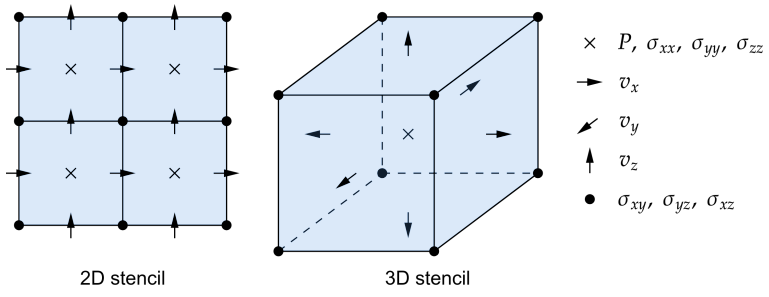


Figure 19. The staggered-grid stencil in 2D and 3D used by StagYY. Note that the vertical axis is denoted by z for both 2D and 3D.

In StagYY, the density field is defined for two locations on the staggered grid:

- Cell centres (pressure P points)
- v_z points

Likewise, viscosities are defined at up to four locations on the staggered grid:

- P points
- σ_{xz} points

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

- σ_{xy} points (3D only)
- σ_{yz} points (3D only)

As an example, to modify the density field used in the z-momentum equation to account for the location of the surface, a reference density ρ_{ref} corresponding to the near-surface density of rock, is multiplied by the appropriate volume fraction (ϕ_{v_z}. The resulting effective density is given by:

$$\rho_{\text{eff}} = \rho_{\text{ref}} \phi_{v_z} \tag{24}$$

Figure 20 illustrates this process for the case of density field modification. Through the use of a volume fraction, this procedure directly incorporates the surface location into the density field. A similar approach can be applied to the viscosities at the various locations used in the momentum equations.

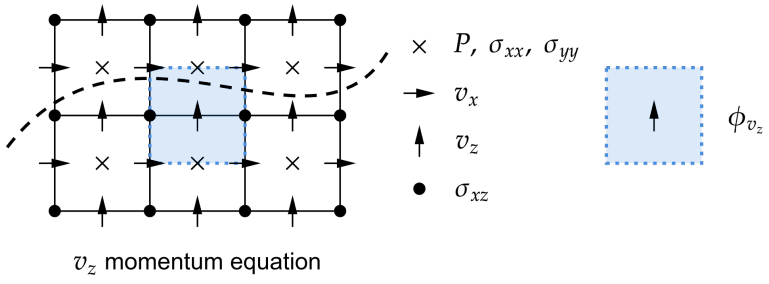


Figure 20. The ϕ_{v_z} volume fraction is used to couple the surface location to the density used in the z-momentum equation near the surface. A reference density is multiplied by this volume fraction in order to obtain a density that takes into account the location of the surface.

In order to work properly, the density and viscosity fields must not already include the effect of the sticky air layer, otherwise the effect of an air layer would be counted twice. It is therefore necessary to set the air density and viscosity to the reference density and viscosity. A caveat of this approach is that it does not produce exactly correct results when there are large lateral variations in surface density or viscosity, as it is no longer possible to choose a single reference value. A solution to this is to copy reference values from the cell immediately below the surface; the closest cell that does not contain air.

5.2.1 Computing volume fractions

In 2D, volume fractions are computed geometrically using the marker chain to determine if a cell lies fully below, above, or intersects the surface. This process, conceptually similar to that in Section 3.5.1, is efficient with linear time complexity $O(n)$. In 3D, the bilinear method allows for exact integration when the surface lies entirely within a single cell, or numerical integration otherwise.

The integral method in 3D follows a similar approach to that in Section 3.5.1, using Delaunay triangles to subdivide the domain into triangular prisms. The volumes of these prisms provide the required volume fractions, also in linear time $O(n)$.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Table 1. Summary of Lagrangian surface tracking options used in 2D benchmarks

Method	Surface MPC	Reinitialisation	Smoothing method	Unmixing method
Marker-in-cell	-	-	-	bouncing
Bilinear method	8	global reinitialisation after each timestep	-	bouncing
Integral method	8	interval method $\Delta x/\Delta x_{\text{init}} \in [0.2, 2.0]$	slope limiter $s_{\text{crit}} = 0.5$	exchanging

A summary of the combination of options used for the three methods compared in 2D benchmarks. Two new methods, referred to as the bilinear method and integral method, are compared with the existing marker-in-cell method. An initial density of 8 surface markers per cell (MPC) and 100 volumetric tracers per cell are used in all 2D models.

In all benchmark tests, computing volume fractions incurred negligible cost compared to overall runtime and other surface marker operations.

6 Results

Three methods, consisting of two Lagrangian surface marker tracking methods and the existing marker-in-cell method, were chosen for testing and benchmarking purposes. In order to isolate the effect of surface tracking method, all methods were tested using the sticky air method to approximate the solution of the Stokes equations with a free surface. The methods are:

Marker-in-cell The existing method in StagYY, which uses tracers to determine the location of the surface via calculation of a C field using linear shape function averaging. This is the baseline against which potential improvements to surface tracking must be compared.

Bilinear method Using surface markers with linear interpolation of the surface in 2D or bilinear interpolation in 3D, similar to the method used by Kaus et al. (2016) and Duretz et al. (2016).

Integral method Using a marker chain or mesh, and evaluating the surface heights on the Eulerian grid using the integral method.

6.1 2D benchmark summary

A summary of the default combination of parameters selected for use with each of the five methods in 2D is shown in Table 1. Unless otherwise stated, all 2D results use these combinations of parameters.

A summary of the 2D benchmarks run can be found in Table 2.

6.2 2D relaxation benchmark

The 2D surface relaxation benchmark (Figure 21) is based on Case 1 as presented by Crameri et al. (2012a). It considers the time dependent relaxation of an initially non-flat surface with initial topography $z_{\text{init}}(x)$ given by the function:

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Table 2. Summary of Lagrangian surface tracking benchmarks in 2D

Benchmark	Geometry	Resolution	Description
2D relaxation	Cartesian	512×128	Isostatic relaxation of an initially non-flat surface based on Case 1 presented by Crameri et al. (2012a)
2D plume	Cartesian	1024×256	Dynamic topography resulting from a mantle plume based on Case 2 presented by Crameri et al. (2012a)
2D constant viscosity	Spherical annulus	Various	Constant viscosity models with $\mathrm{Ra} = 10^7$ used for investigating performance scaling between different surface marker densities and resolutions

A summary of all the benchmark setups run in 2D.

$$z_{\mathrm{init}}(x) = 7\cos\left(\frac{2\pi x}{2800\ \mathrm{km}}\right)\mathrm{km}. \tag{25}$$

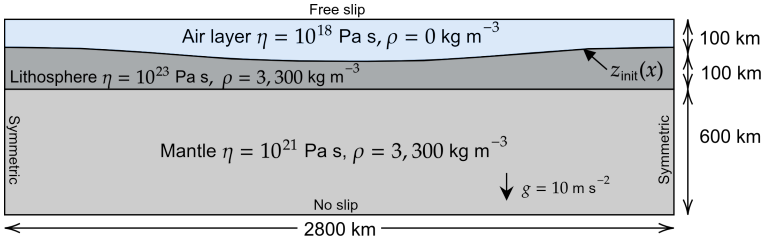


Figure 21. 2D surface relaxation benchmark based on Case 1 presented by Crameri et al. (2012a). An initially non-flat surface in a three-layer model relaxes until equilibrium is reached after approximately 100 ky.

This problem is simple enough to have an analytical solution for surface position as a function of time as derived using a three-layer model (Ramberg, 1981). Starting from an initial maximum of 7 km, the maximum topography as a function of time $z_{\mathrm{max}}(t)$ is given by:

$$z_{\mathrm{max}}(t) = 7\exp\left(\frac{t}{14.825\ \mathrm{ky}}\right)\mathrm{km}. \tag{26}$$

The model domain is a 2800×800 km Cartesian box consisting of three compositional layers: a 600 km thick mantle layer, a more viscous 100 km thick lithosphere, and a 100 km thick air layer. The resolution is 512×128, with 100 tracers per cell for advecting material properties and tracking the surface in the case of the marker-in-cell method. The sticky air layer has a viscosity of 10^{18} Pa s. Symmetric boundary conditions are imposed at the horizontal edges of the model, while a no-slip boundary condition is imposed at the bottom boundary, and a free slip condition is imposed at the top. Acceleration due to

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

gravity g is set to 10 m s^{-2}. For consistency with the benchmarks run by Crameri et al. (2012a), models were run until $t = 100$ ky to show the system reaching equilibrium. Timesteps were limited to 500 years in order to give high temporal resolution for plots of model evolution.

The five surface tracking methods can be compared against the analytical solution when plotting maximum topography with time, as in Figure 22.

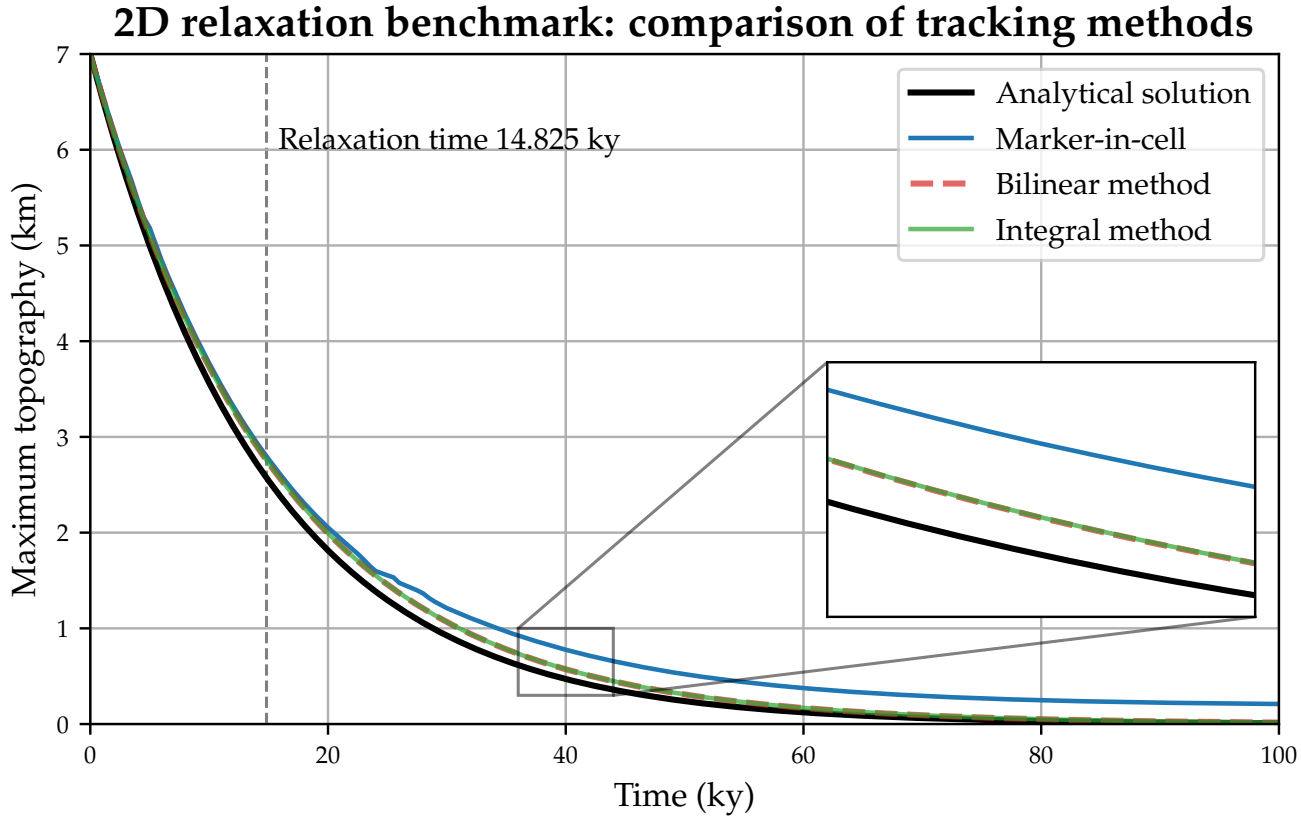


Figure 22. Topography with time for the 2D relaxation benchmark. Both Lagrangian marker based methods closely match the analytical solution. The existing marker-in-cell method, while also tracking the analytical solution closely initially, fails to converge with the analytical solution near the end of the model run.

All methods are qualitatively able to track the analytical solution closely in agreement with the results presented in Crameri et al. (2012a). Minor divergence between the methods becomes more apparent after 30 ky, with the marker-in-cell method failing to match the analytical solution as closely as Lagrangian tracking methods after this point. At 100 ky, while the other methods have converged to a maximum topography near zero, the marker-in-cell method resulted in around 200 m of topography, a result that can also be seen in Figure 2 of Crameri et al. (2012a). This result can be attributed to the maximum topography being affected by the noise inherent in the marker-in-cell method when using a finite number of randomly initialised tracers

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

throughout the model. Meanwhile, the Lagrangian surface marker-based tracking methods all converge to near zero at 100 ky, indicating that Lagrangian surface tracking methods are less susceptible to this noise.

It should be noted that part of the remaining discrepancy between the methods presented here and the analytical solution can also be explained by the limitations of the sticky air method used to produce the solution to the Stokes equations. Therefore, the remainder of the discrepancy in results may not necessarily be a direct consequence of the surface tracking method used.

6.3 2D plume benchmark

The 2D plume benchmark (Figure 23) is based on Case 2 presented in Crameri et al. (2012a). It considers a buoyant plume rising through the mantle over a period of 20 My, and the dynamic topography resulting from this plume's impingement on the lithosphere.

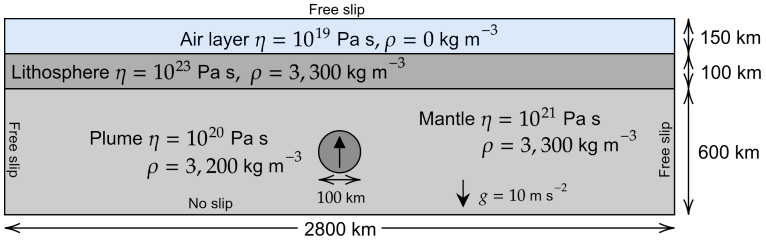


Figure 23. 2D plume benchmark based on Case 2 in Crameri et al. (2012a). A buoyant plume rises through the mantle and impinges upon the lithosphere, producing dynamic topography over a time period of 20 My.

The domain is a 2800×850 km Cartesian box domain run at a resolution of 1024×256, consisting of three compositional layers: a 600 km thick mantle, a 100 km thick lithosphere, and a 150 km thick air layer with a sticky air viscosity of 10^{19} Pa s. The buoyant mantle plume, with radius 100 km, has a density of 3,200 kg m^{-3}; 100 kg m^{-3} less than that of the surrounding mantle and lithosphere. 100 tracers per cell are used for advecting material properties and for tracking the location of the surface in the case of the marker-in-cell method. Acceleration due to gravity g is set to 10 m s^{-2}. Free slip boundary conditions are imposed at the horizontal edges of the model, while a no-slip boundary condition is imposed at the bottom boundary, and a free slip condition is imposed at the top. The model is run for 20 My, which allows for the plume to rise through the mantle and impinge upon the lithosphere causing deformation of the surface.

Analytical solutions for this benchmark are not available. However, results from multiple numerical geodynamic models, including finite element models with true deformable free surfaces, were run as part of the study of Crameri et al. (2012a) and consistently show a maximum surface deformation 800-850 km after 20 My.

The results of the plume benchmark further demonstrate the strengths of Lagrangian tracking methods in isolating the topographic response of the model without the noise associated with the marker-in-cell method. The marker-in-cell method introduces random noise, which results in a ≈100 m discrepancy in topography at the centre of the plume. Lagrangian tracking methods closely matched the results demonstrated by Crameri et al. (2012a) for both the maximum topography at 20 My and the evolution of topography in the intervening time.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

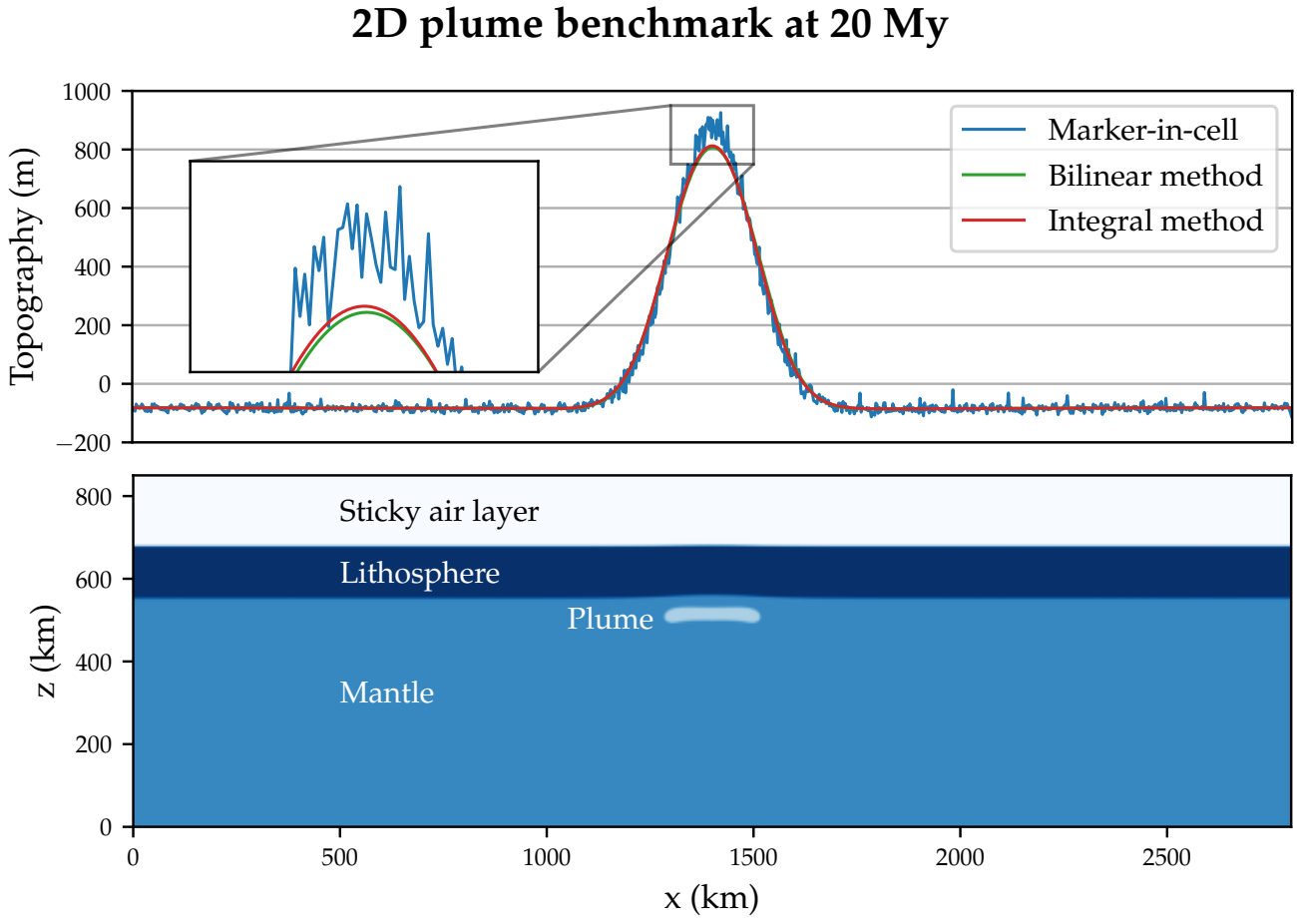


Figure 24. After 20 My all methods produce results similar to those demonstrated in Crameri et al. (2012a), however, the marker-in-cell and integral method demonstrate significantly more noise compared to the other methods and a systematically raised maximum topography.

6.4 2D scaling tests

Scaling tests in 2D were performed using a non-dimensional model in spherical geometry, an example of which is shown in Figure 25. The model setup involves a constant viscosity mantle with Rayleigh number $\mathrm{Ra} = 10^7$, and a sticky air layer with a thickness 10% that of the mantle with a viscosity contrast $\eta_{\mathrm{rock}}/\eta_{\mathrm{air}} = 10^3$. All scaling models were run for a total of 5,000 time steps to ensure that a statistically steady state convection was reached.

Two tests were performed to judge the performance of the methods in 2D: tests of different surface marker densities, and tests of different model resolutions. In both cases, the amount of computational effort spent tracking surfaces using Lagrangian markers can be divided into four main operations: advection (including sorting of the marker chain), interpolation, smoothing,

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

and unmixing. The relative impact of these four operations can be compared to judge the efficiency of the bilinear and integral methods.

6.4.1 2D scaling with surface marker density

The first scaling test examines the effect of marker density on performance. Marker densities ranging from 1 marker per cell (MPC) to 64 MPC were tested, with performance evaluated for both the bilinear and integral methods (Figure 26). The computational time needed for the surface tracking method is expressed as a fraction of the total computational time, which also includes calculating the Stokes solution using the MUMPS direct solver (accessed via PETSc), advecting the temperature field using a TVD method, and advecting the volumetric tracers (as discussed earlier).

The results indicate that the bilinear method's impact on total runtime is minimal, consistently accounting for approximately 3% of computational effort across all tested MPC values. Conversely, the integral method exhibits a significant increase in computational demand as marker density increases, peaking at 11% of total runtime at 64 MPC.

The unmixing operation's inefficiency can be attributed to its requirement to loop through all tracers in the model to determine their position relative to the surface. This process alone accounts for approximately 2–3% of total runtime, as observed in the bilinear method. Additional computational effort is incurred when determining the surface location using linear interpolation and identifying a suitable template for the tracer exchange method. For the bilinear method, determining the surface location at a given point can be performed in constant $O(1)$ time, whereas the integral method requires $O(\log n)$ time.

Other operations, such as advection and interpolation, impose minimal computational overhead in both methods. However, sorting, an operation unique to the integral method, scales with the number of surface markers per cell (MPC). This increased cost is primarily due to the requirement for additional iterations of the slope limiting algorithm at higher marker densities.

6.4.2 2D scaling with resolution

The second scaling test involves investigating the effect of changing resolution. Four model resolutions were tested: 128×16, 256×32, 512×64, and 1024×128, and the model performance measured for the bilinear and integral method at each resolution. For each resolution, model performance was assessed using both the bilinear and integral methods. A fixed marker density of 8 surface markers per cell (MPC) was maintained throughout all tests.

The results reveal that while the computational effort required for advection, interpolation, and smoothing decreases (as a fraction of total computational effort) with increasing resolution, the effort required for unmixing scales proportionally with resolution. This behaviour arises because the unmixing process depends on the number of tracers in the model, which increases with resolution. Due to the increased computational overhead of determining surface position for each for these tracers when using the integral method, the bilinear method performs better overall, consistently taking less than 3% of the overall runtime.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

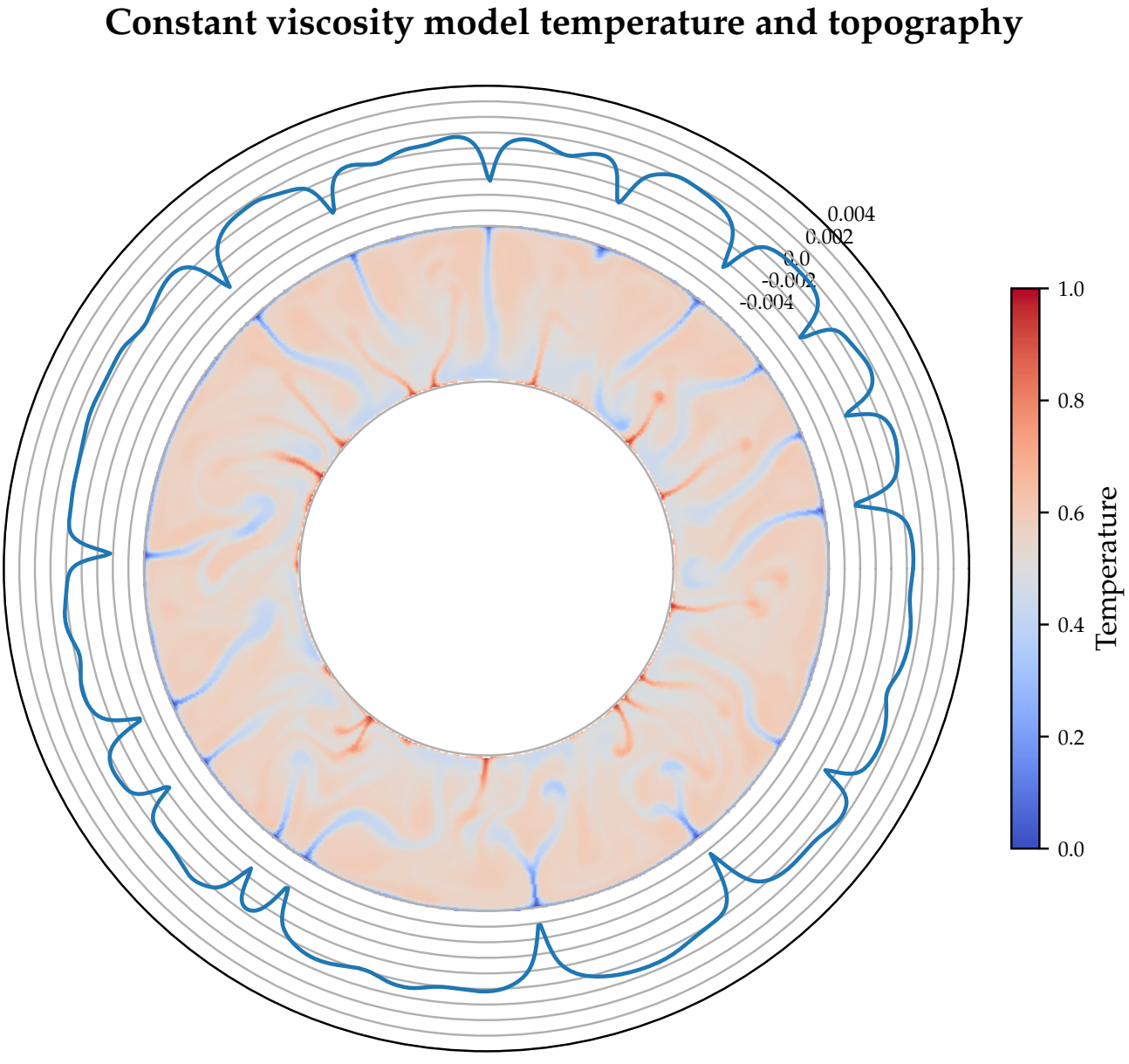


Figure 25. Example of topography generated using a simple constant viscosity model in 2D spherical geometry. The underlying temperature field shows the location of various plumes and downwellings, which are captured in the surface topography generated using the bilinear method. This model forms the basis of the 2D scaling tests performed.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

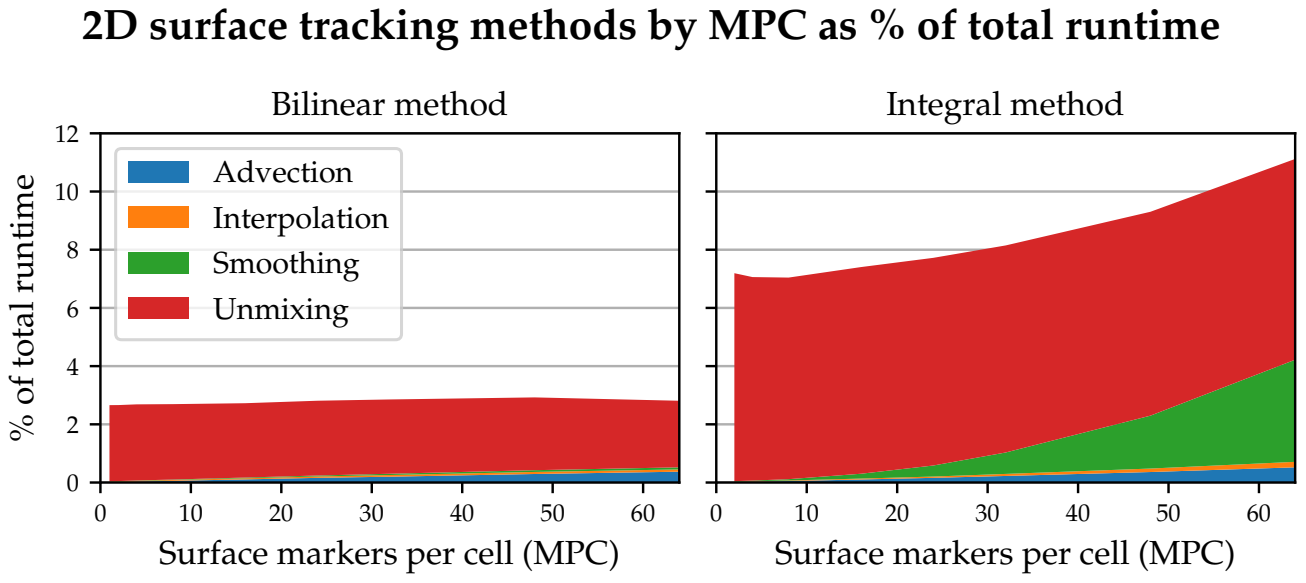


Figure 26. 2D scaling test evaluating the impact of surface marker density (MPC) on tracking performance. The integral method demands more computational effort than the bilinear method, reaching a maximum of 11% of total computational effort at 64 MPC compared to 3% for the bilinear method. Tracer unmixing accounts for the majority of computational cost.

Table 3. Summary of Lagrangian surface tracking options used in 3D benchmarks

Method	Surface MPC	Reinitialisation	Smoothing method	Unmixing method
Marker-in-cell	-	-	-	bouncing
Bilinear method	16	global reinitialisation after each timestep	-	bouncing
Integral method	16	interval method $\Delta x/\Delta x_{\text{init}} \in [0.2, 3.0]$	slope limiter $s_{\text{crit}} = 0.5$	exchanging

A summary of the combination of options used for the three methods compared in 3D benchmarks. Two new methods, referred to as the bilinear method and integral method, are compared with the existing marker-in-cell method. An initial density of $16 = 4 \times 4$ surface markers per cell (MPC) is used in all 3D models.

6.5 3D benchmark summary

Similarly to 2D, it is possible to run a set of benchmarks to test the efficiency and accuracy of Lagrangian surface tracking in 3D. The combinations of parameters used for models in 3D are summarised in Table 3. Unless otherwise stated, all 3D models presented below use these combinations of parameters.

A summary of the 3D benchmarks run can be found in Table 4.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

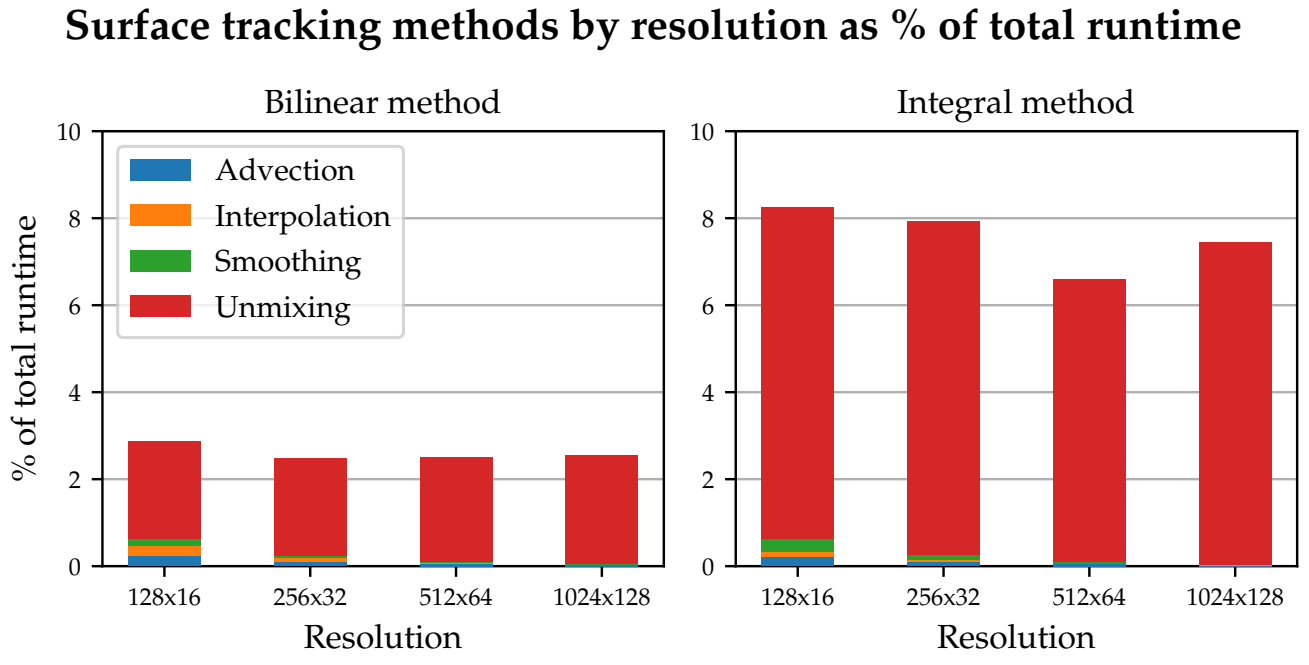


Figure 27. Scaling test in 2D investigating the effect of resolution on surface tracking performance as a percentage of total computational effort. Tracer unmixing dominates runtime for both methods, a result of unmixing being dependent on tracer density instead of surface marker density.

Table 4. Summary of Lagrangian surface tracking benchmarks in 3D

Benchmark	Geometry	Resolution	Description
3D plume	Cartesian	$256 \times 256 \times 128$	Dynamic topography resulting from a rising mantle plume
3D constant viscosity	Cartesian	Various	Constant viscosity models with $\mathrm{Ra} = 10^7$ used for comparison with the marker-in-cell method and for investigating performance scaling with different surface marker densities and resolutions

A summary of the benchmarks run in 3D.

6.6 3D plume benchmark

The 2D plume benchmark presented in Section 6.3 can be extended into 3D. The domain in 3D is a $1400 \times 1400 \times 850$ km Cartesian box with a resolution of $256 \times 256 \times 64$, a resolution chosen to balance the high computational cost of 3D models with accuracy. The model consists of a 600 km thick mantle, a 100 km thick lithosphere, and a 150 km thick air layer. The geometry of the mantle plume is modified to a sphere with an initial diameter of 100 km. A rigid (no slip) boundary condition

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

is imposed at the lower boundary, while free slip boundary condition is imposed on all other boundaries. All other physical parameters remain identical between the 2D and 3D models. As this 3D version is a novel benchmark, there are no previous results available for comparison.

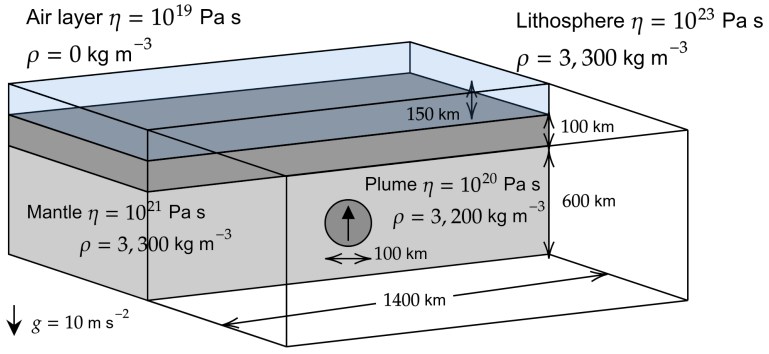


Figure 28. 3D plume benchmark setup, the 3D extension of the 2D plume benchmark above and in Crameri et al. (2012a). 3D models are run at a resolution of $256 \times 256 \times 64$

The performance of different tracking methods, as illustrated in Figure 29, highlights the inherent noise challenges of the marker-in-cell method in 3D. This method struggles with noise generation due to the inherent dependence on tracer density and positioning. Nonetheless, it is still able to capture the main features of the mantle plume, particularly the central topographic uplift created by the buoyant upwelling. The bilinear and integral methods, although somewhat effective at reducing this noise, still display disturbances across varying spatial scales, emphasizing the need for improved coupling techniques.

Figure 30 further emphasizes these findings by providing a cross-sectional view through the model. The marker-in-cell method's pronounced noise is visually prominent in the profile, while the integral and bilinear methods show considerable improvement. Minor differences between the two methods are visible, possibly a side effect of the differing smoothing methods used.

6.7 3D constant-viscosity Cartesian box

As a comparison to the models shown in Figure 3, a simple Cartesian box setup with a relatively coarse $64 \times 64 \times 32$ resolution was used as a basis for testing various aspects of the surface tracking methods. This model considers convection with a constant viscosity using non-dimensional parameters with $\mathrm{Ra} = 10^7$. The topography generated by the five methods can be compared against the topography generated by a high-resolution marker-in-cell method to compare accuracy.

Visual inspection of the results in Figure 31 show that both the Bilinear and Integral method are able to eliminate the noise of the marker-in-cell method and isolate the plume topography.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

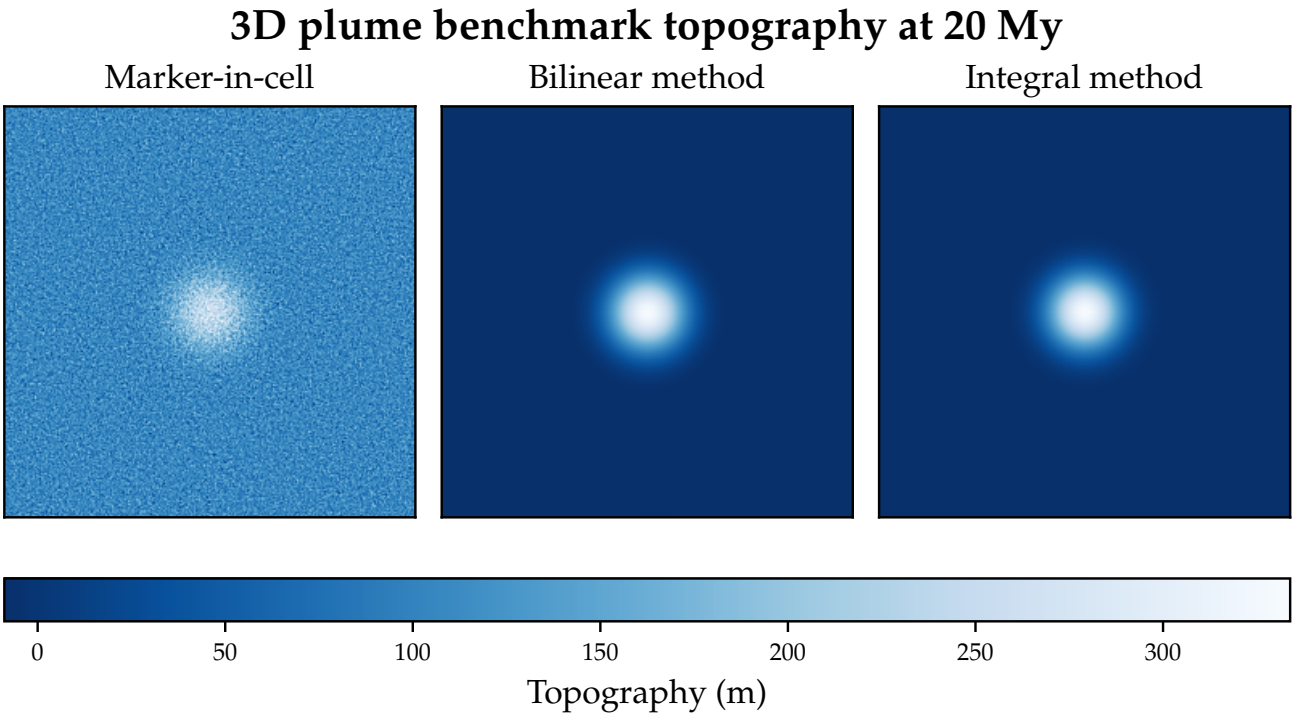


Figure 29. 3D plume benchmark topography at 20 My (top-down view). The marker-in-cell method with 100 tracers per cell is able to capture the subtle topography generated by the mantle plume at the centre of the model, however noise inherent in this method is visually apparent. Both the bilinear and integral method eliminate this noise, providing visually similar results that isolate the plume's influence on topography.

6.8 3D scaling tests

The same Cartesian box model was used in order to test scaling of surface tracking methods with respect to three variables: model resolution, tracer density and surface marker density.

6.8.1 Resolution dependence test

The resolution dependence test involved the simple Cartesian box model evaluated at four resolutions, ranging from $32\times32\times16$ to $256 \times 256 \times 128$, while keeping all other parameters constant.

To qualitatively assess the impact of resolution, a top-down view of the surface topography at the final timestep, as tracked using the integral method, is shown in Figure 33.

The results of this qualitative test indicate that the smooth topography achieved using the integral method is consistent across all length scales, in contrast to the grainy and unrefined topography produced by the marker-in-cell method. Similar

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

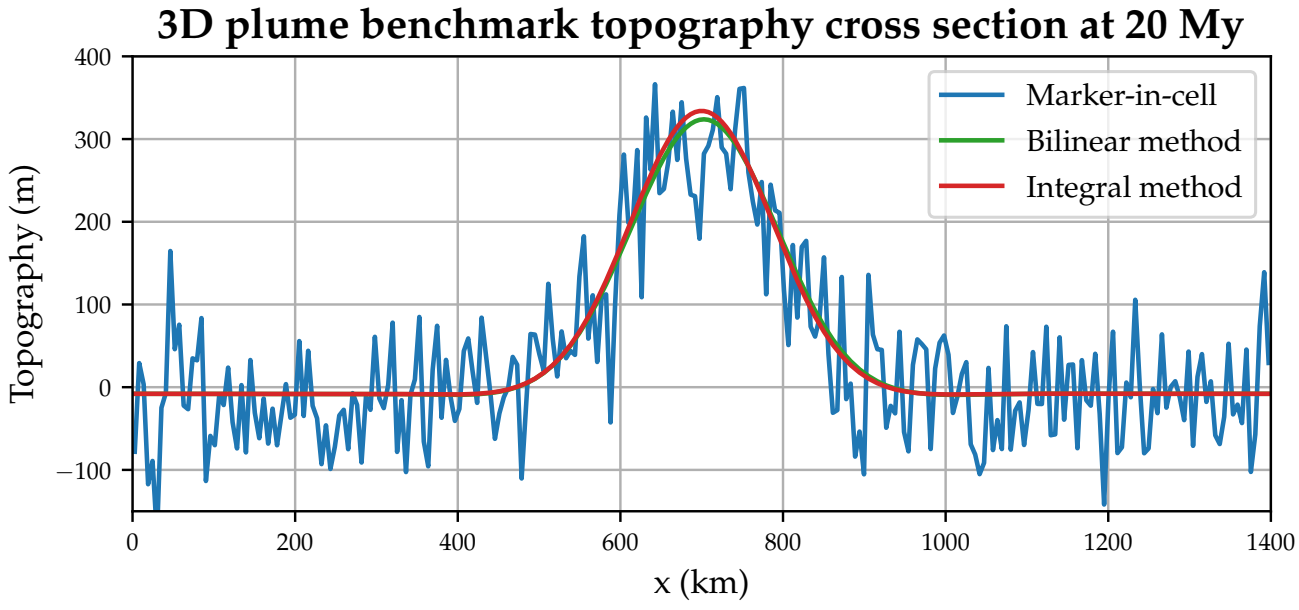


Figure 30. 3D plume benchmark topography at 20 My by surface tracking method (cross section). Similar to the corresponding 2D benchmark, both the bilinear and integral methods are able to effectively capture topography without the noise inherent in the marker-in-cell method.

results were observed for the bilinear method. As expected, higher resolutions reveal greater detail, particularly around regions near downwellings, while these features appear more diffuse at lower resolutions.

To quantitatively analyse the effects of varying resolutions, the time-averaged surface topography distributions were examined. Variable resolutions pose a challenge for comparison because the timesteps and total runtime differ due to the constant CFL number maintained across all resolutions. At higher resolutions, velocities are more constrained, resulting in longer simulation times. For example, the lowest resolution ($32 \times 32 \times 16$) achieved a final model time approximately 7.3 times greater than the highest resolution ($256 \times 256 \times 128$) after 5000 timesteps. To account for this, only overlapping time intervals were analysed, excluding the initial timesteps to allow for the establishment of a steady-state convective regime. The results are shown in Figure 34.

The analysis indicates a clear dependence on model resolution. The median topography increases with higher resolutions, while the mean topography remains consistently at 0.0. This pattern suggests the presence of deeper trenches at higher resolutions that skew the median. This result aligns with expectations, as higher resolutions better resolve trench features, which appear more diffuse at lower resolutions. Notably, these observations could not have been achieved with marker-in-cell methods, which are heavily influenced by both tracer density and model resolution.

The performance of each method was also evaluated at different resolutions, as shown in Figure 35. Similar to the 2D scaling tests, the performance of both methods is dominated by the unmixing operation. Unmixing scales with the number of tracers

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

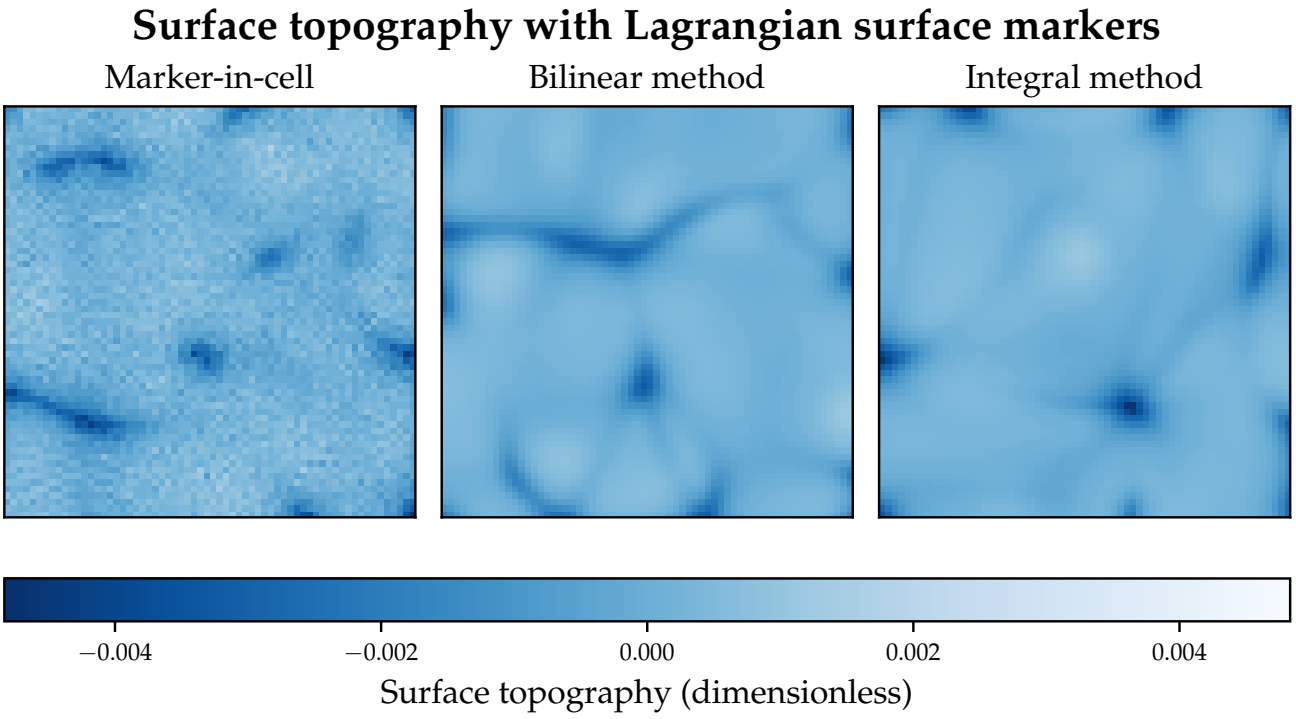


Figure 31. Surface topography as generated by three methods at 100 markers per cell. The marker-in-cell method is the baseline (see also Figure 3). Both Lagrangian tracking methods provide visually smoother topography.

rather than the number of surface markers. For the bilinear method, determining surface location at a given location can be accomplished in constant $O(1)$ time, while the integral method requires iterating through Delaunay triangles, which scales with $O(\log n)$ (where n is the number of surface markers) in the best case. Consequently, unmixing time remains roughly constant for the bilinear method across resolutions, whereas it generally increases with resolution for the integral method. For 3D models, the use of iterative solvers introduces some variability to the overall solution time, accounting for the variation in timing results observed.

Other components, such as advection and interpolation, require minimal computational effort for both methods across all resolutions. However, the integral method's slope limiting algorithm for marker mesh smoothing adds significant computational cost to that method, which scales with resolution due to the increased iterations required to relax the marker mesh at higher resolutions.

6.8.2 Tracer density dependence test

A key question is whether the implemented methods can eliminate the dependence on tracer density and the associated reduction in computation performance that comes with it, as discussed in Section 2.2. This can be tested by running a series of

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

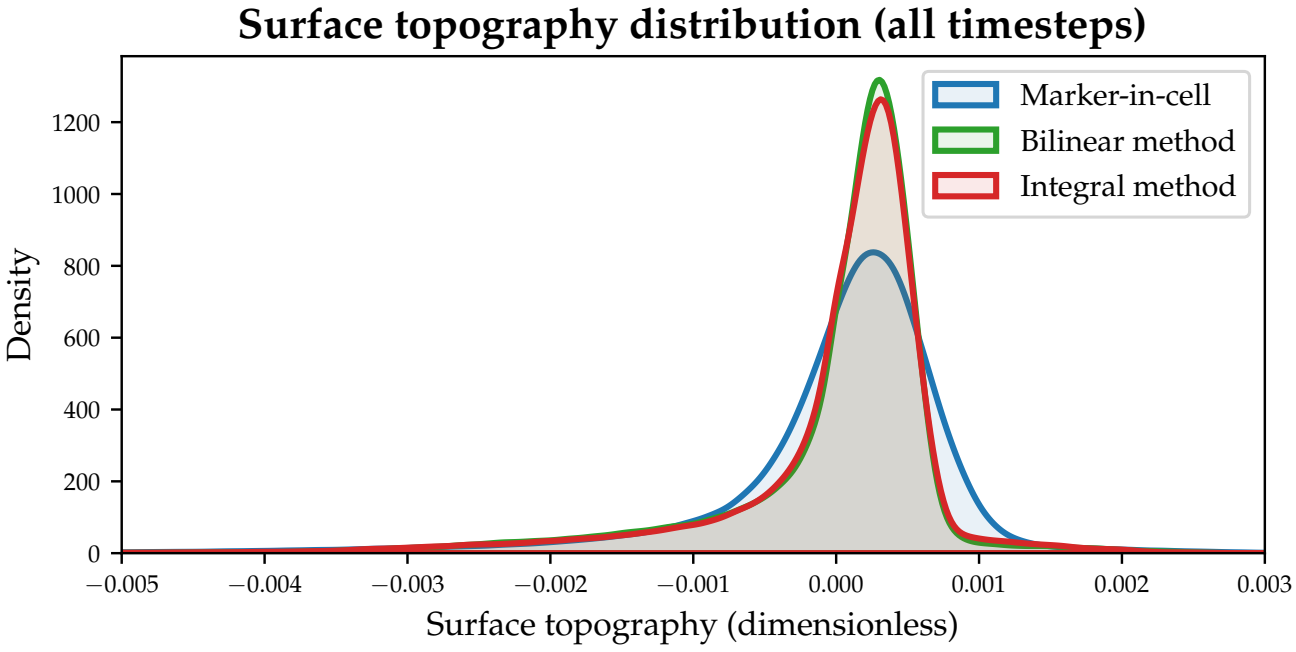


Figure 32. The noise inherent with the marker-in-cell method is shown by its wide distribution of topography.

models with varying tracer densities and comparing their surface topography (Figure 36) to determine both qualitatively and quantitatively whether there is a persistent dependence on tracer density.

The smooth surface topography demonstrates that there is no longer any dependence of surface topography on tracer density, in contrast to the marker-in-cell method, which is as expected because the density and viscosity values near the surface are adjusted directly from the tracked surface position. This is a positive result, as it implies that model performance can be improved by decreasing tracer density without compromising the quality of the surface topography. These results were also verified by comparing time-averaged topographic distributions.

6.8.3 Surface marker density dependence test

A final question to answer through scaling tests is the dependence of the methods on the density of the surface markers themselves. All other models in 3D were run using a surface marker marker density of $4 \times 4 = 16$ markers per cell, however this was chosen arbitrarily. In order to test the surface marker density dependence, scaling tests were performed for marker densities corresponding to all square numbers betwen $1 = 1 \times 1$ markers per cell, and $64 = 8 \times 8$ markers per cell on a constant viscosity $64 \times 64 \times 32$ Cartesian box domain. First, the performance of different surface marker densities was considered for both the integral and bilinear methods, as shown in Figure 37.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

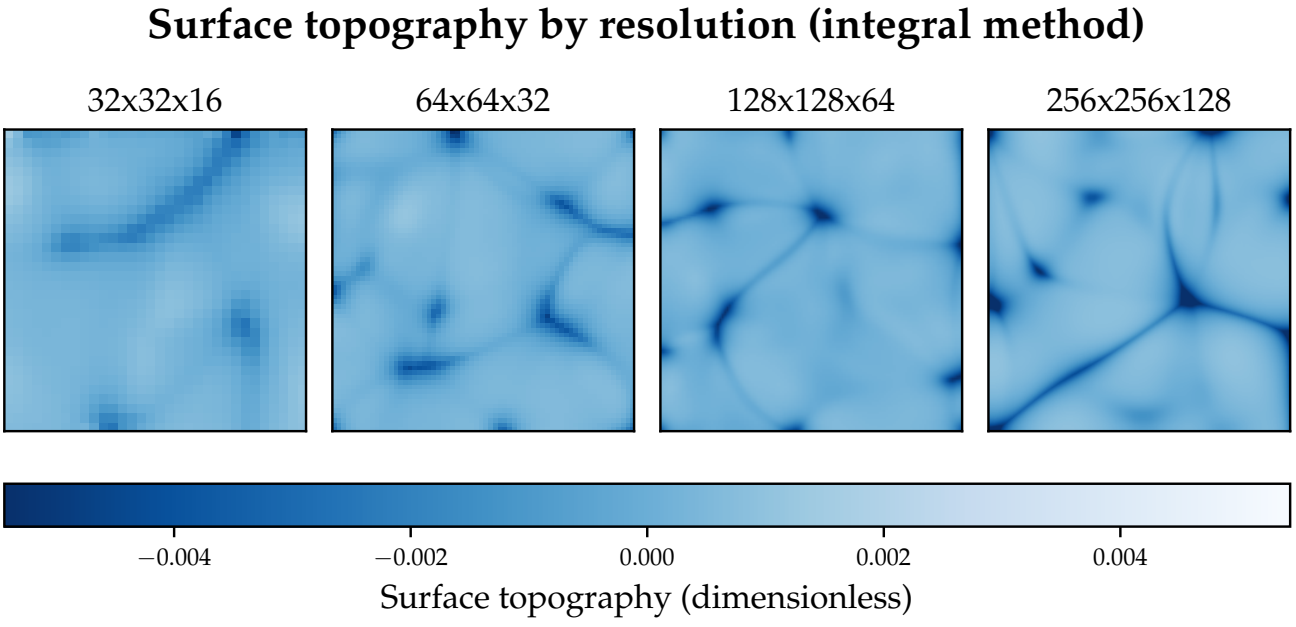


Figure 33. 3D surface topography tracked using the integral method at different resolutions. The smooth topography is consistent across length scales. Higher resolutions resolve greater detail smoothly, while lower resolutions produce more diffuse, but still smooth, features in downwelling regions. Similar results can be obtained when using the bilinear method.

As expected based on previous results, the bilinear method consistently outperforms the integral method across all MPC counts. Unmixing dominates computational effort for the same reasons discussed previously: for the integral method, unmixing depends not only on tracer density but on the number of surface markers, requiring $O(\log n)$ time to determine surface height at a given location.

7 Summary and discussion

7.1 Summary of key results

The key results of this study can be summarised with several points, which are elaborated upon in the discussion below:

- Lagrangian surface markers are a simple and effective way of tracking free surfaces in numerical geodynamic models.
- The bilinear method is able to track surfaces accurately and with good performance, however it does introduce some artificial numerical diffusion due to periodic global reinitialisation.
- The integral method is also able to track surfaces accurately, however it requires additional methods to smooth and stabilise the surface, and is significantly less performant than the bilinear method.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

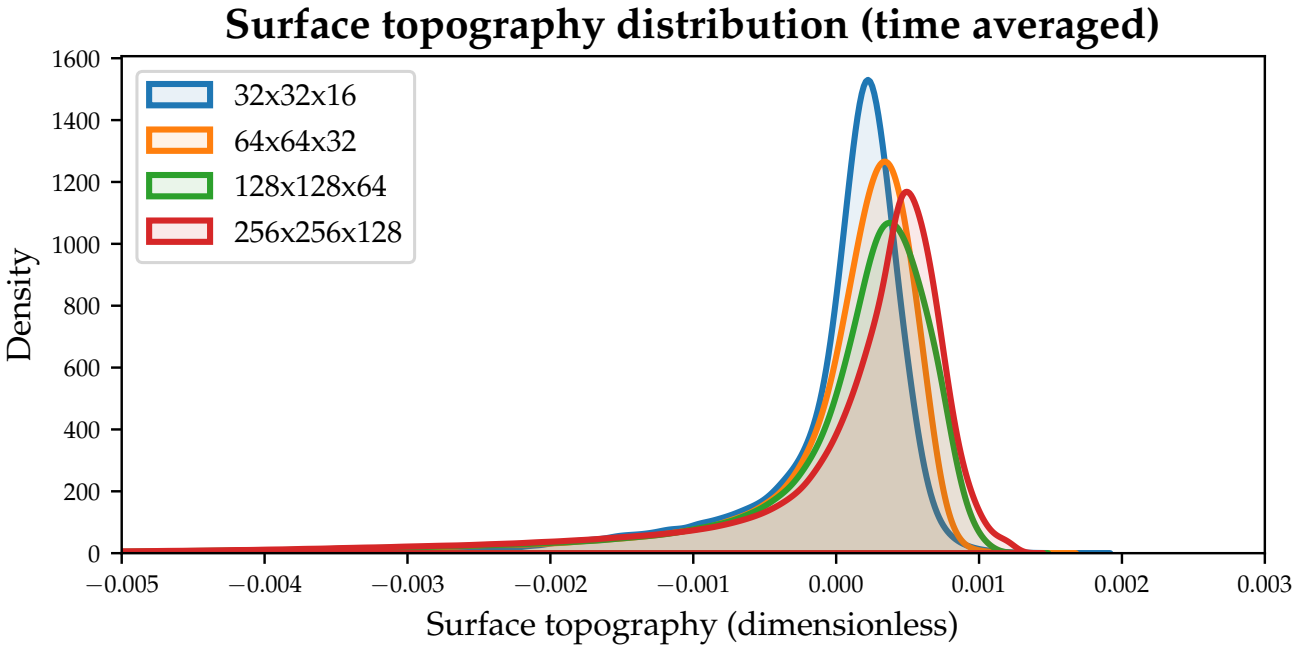


Figure 34. Time averaged topography distributions for a variety of resolutions using the integral method (corrected for variable model runtimes). A general trend towards increased median topography with resolution is seen, which is the expected result of a diffuse representation of trenches at lower resolutions.

- Both methods can be used to obtain volume fractions for coupling the location of the surface to the underlying Stokes solver through the density and viscosity fields (ρ, η coupling).
- Coupling of the surface tracking method to the underlying Stokes solver through ρ, η coupling is essential for smooth and accurate surface tracking results.
- When tracking topography, performance gains are possible by reducing tracer density without sacrificing topographic quality due to the independence of the reconstructed surface on tracer density.

7.2 Discussion of key results

7.2.1 Implementation

The implementation of Lagrangian surface markers, especially the bilinear method, is relatively straightforward. Since Lagrangian tracers already exist in StagYY, as they do for many other numerical geodynamic modelling codes, existing routines for manipulating these tracers were easily adapted for the new surface markers. Reconstructing the surface using the bilinear representation was similarly straightforward and could likely be implemented in other numerical geodynamic codes with min-

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

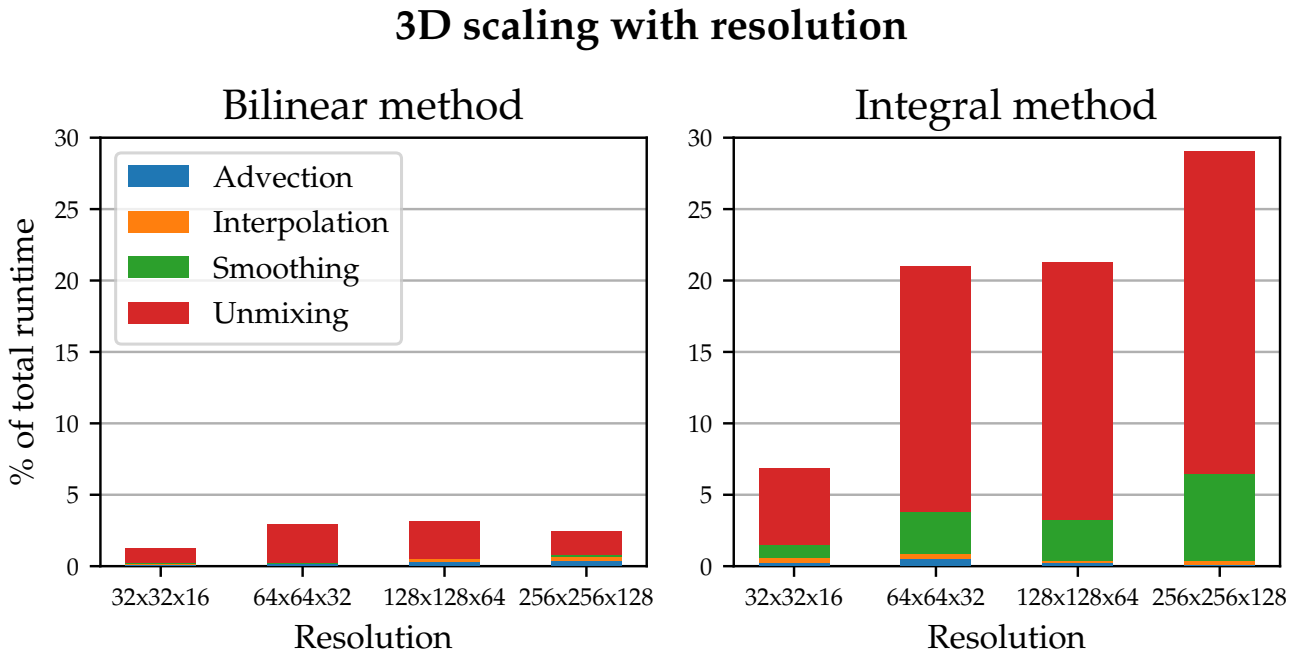


Figure 35. Performance of each method with resolution as a percentage of total runtime. The bilinear method performs significantly better than the integral method in 3D, particularly in terms of tracer unmixing, which was the biggest contributor to the performance of both methods.

imal effort. Indeed, a variation of the bilinear method described here is already integrated into the geodynamic code LaMEM (Kaus et al., 2016), and a simplified 2D version was employed by Duretz et al. (2016) to test free surface discretisation methods.

In contrast, the integral method is significantly more complex to implement, particularly due to the need for surface marker reinitialisation and smoothing. The 3D implementation requires a significant amount of code for performing complex operations, such as computing the Delaunay triangulation and determining cellwise volume fractions. Developers considering these methods should carefully weigh the cost-benefit trade-offs, especially given the integral method's performance limitations, which are discussed further below.

7.2.2 Bilinear method

The bilinear method was able to produce high-resolution topography, and demonstrated the best computational performance among the methods tested.

The primary strength of the bilinear method lies in its computational efficiency. Key surface marker chain/mesh operations, such as marker advection and surface reconstruction, are performed in linear time $O(n)$ with respect to the number of surface markers. Once surface locations are interpolated to nodes, determining the surface height at a specific point can be achieved in

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

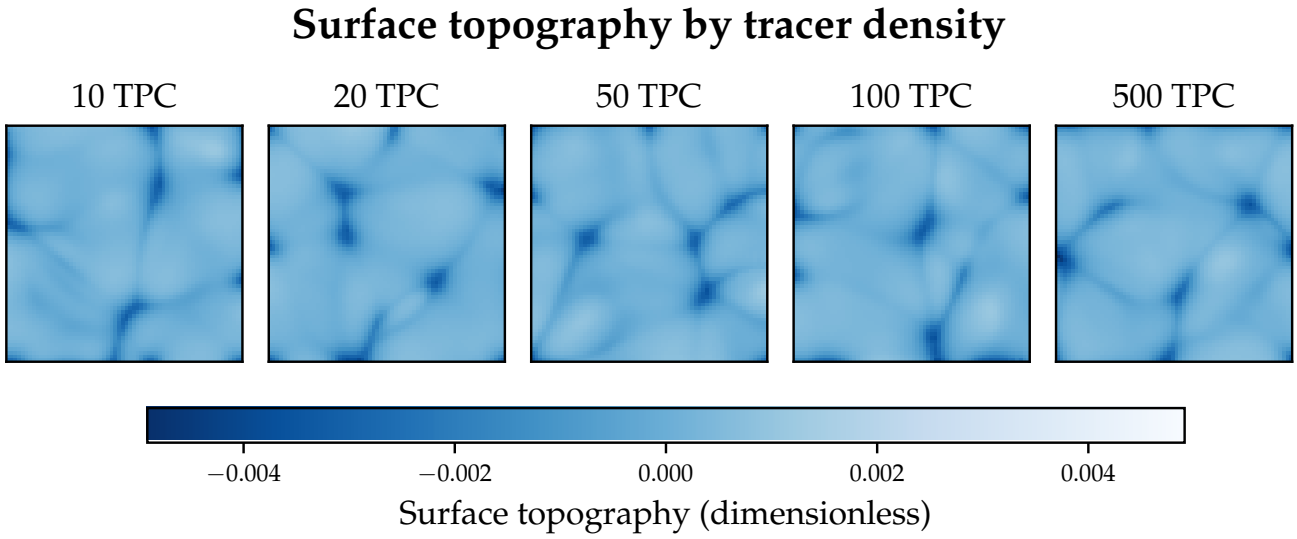


Figure 36. Surface topography as determined using the integral method by tracer density. No qualitative differences are seen in the surface topography reconstruction when changing the underlying tracer density. A similar result can be seen for the bilinear method.

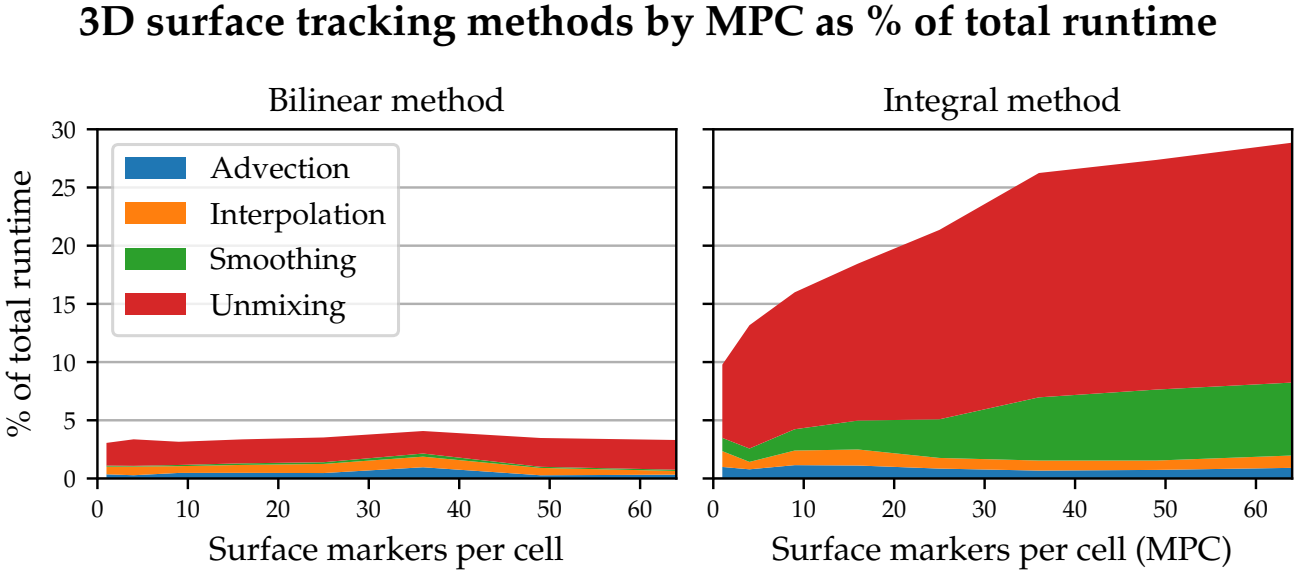


Figure 37. Smoothing and unmixing take up the majority of the runtime in 3D models, especially at high marker per cell counts.

constant time $O(1)$. This efficiency significantly benefits operations that require frequent lookups of surface heights, such as the tracer unmixing operation.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Global reinitialisation introduces implicit numerical diffusion, which helps stabilise the marker chain and avoids the need for explicit smoothing operations, improving performance. However, this same diffusion can gradually reduce the accuracy of the surface representation. While some diffusion is necessary to maintain stability and prevent marker entrainment, it can potentially come at the cost of precision when considering sub-grid scale features.

7.2.3 Integral method

The integral method can produce similarly high-quality topography to the bilinear method but is less performant.

Its main advantage is the absence of artificial numerical diffusion, as surface markers move independently of the Eulerian grid and are only adjusted where needed. In 3D, Delaunay triangulation offers a natural surface representation, suitable for visualising fine-scale features, tracking sea level, and potentially integrating surface process models. However, the method's reliance on several $O(n \log n)$ operations such as marker sorting and reinitialisation limits its efficiency. Although advection and reconstruction remain $O(n)$, slope limiting adds further cost and cannot be solved in linear time. The largest performance bottleneck in both 2D and 3D is tracer unmixing. Unlike the constant-time interpolation of the bilinear method, the integral method requires $O(n \log n)$ per tracer, leading to $O(m \log n)$ overall with m tracers. This cost remains significant even after optimisation of the unmixing algorithm to reduce the number of tracers that are subjected to the unmixing operation.

7.2.4 Comparison of implemented methods with the marker-in-cell method

Both the implemented methods improve upon the existing marker-in-cell method in several ways. Firstly, it bypasses the need for a high tracer density to obtain high-quality surface topography, and therefore allows for significantly decreased computational effort and memory consumption, with the tradeoff of higher computational effort required for the surface tracking methods themselves. Secondly, the implemented methods are more accurate than marker-in-cell based methods, reducing topographic noise on all scales and allowing the accurate modelling of surface topography.

7.2.5 Necessity of density and viscosity coupling

A key result of this study is the necessity of coupling the location of the surface to the underlying physical model through the density and viscosity fields, referred to as ρ, η coupling in Section 5.2. This coupling is essential to avoid topographic noise that arises as a result of determining density and viscosity based on a finite number discrete tracers with a random initial distribution.

The requirement for coupling density directly to the surface was also noted in the results of Duretz et al. (2016), in which a conceptually similar method to the ρ, η coupling used here was applied. However, the methods presented here take this further with application to the viscosity field as well.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

7.3 Applications and future directions

7.3.1 Surface processes

A potential future research direction would be the implementation of more sophisticated smoothing methods, or even the direct implementation of surface processes into StagYY. While the diffusion and slope limiting methods presented here can be considered to be very simple methods of modelling erosion and sedimentation, a direct representation of the surface that is coupled to the underlying model theoretically allows for fully coupled modelling of surface processes on the global scale for the first time.

Potential future research directions might be the incorporation of a surface process model such as Badlands (Salles, 2016) directly into StagYY which could modify the surface at each timestep. With the coupling of the free surface to the underlying model through tracer unmixing and coupling to the Stokes solver, this modified surface would then have a direct effect on the underlying model. This presents an opportunity for future research to incorporate both surface processes and a free surface boundary condition together on the global scale.

7.3.2 Alternative Stokes solvers

A surface tracking method must be coupled with an appropriate solver in order to fully solve the free surface problem. While the results presented in this chapter have all employed the sticky air method to approximate the solution to the free surface problem, other methods of modelling free surfaces exist that can be coupled with Lagrangian surface tracking to form a complete free surface solver.

One such method is the staircase representation of the surface presented by Duretz et al. (2016). In this method, no equations are solved within the air layer, and alternative boundary conditions are applied to cells near the surface in order to approximate a free surface solution without the need for a sticky air layer. A prerequisite of this method is a method to determine whether the midpoint of a given cell is above or below the surface, which is readily achieved using a marker chain or mesh.

The direct representation of the surface also enables the possibility of implementing the variational Stokes discretisation presented in Larionov et al. (2017). The key requirements for the implementation of this method are volume fraction functions, which were already implemented as part of the coupling method described in Section 5.2.

7.3.3 Sea level modelling

Accurately tracking the evolution of sea levels over time is a key capability of the direct surface tracking methods presented here and has several promising applications.

Sea level tracking is a necessary step toward coupling geodynamic models with climate and biological evolution models. Climate models such as FOAM (Tobis et al., 1997) and biological models like Gen3sis (Hagen et al., 2021) require information about continental configurations, which can be more accurately provided by dynamically computing sea level changes from an initial ocean volume.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

Through direct and accurate surface tracking, the evolution of a constant sea volume can be tracked to give information about changes in sea level and coverage area. As StagYY already includes functionality to estimate volatile outgassing rates, including water, it would theoretically be possible to incorporate this into a model which considers a changing sea volume. This would enable for the first time to couple surface processes with interior processes in such a way to allow for more integrated models linking deep Earth dynamics with climate, landscape, and biosphere evolution.

Code availability. The code StagFS, a testbed code based on StagYY, may be used to produce the benchmark results and figures used in this paper. It is archived on Zenodo under the GPLv3 license under https://doi.org/10.5281/zenodo.18096249 (Tackley and Gray, 2025).

Author contributions. **Timothy Gray**: original study design; code and algorithm development; scaling and performance benchmarks; figure creation. **Paul Tackley**: development of StagYY; conceptual input; manuscript review. **Taras Gerya**: conceptual input; manuscript review.

Competing interests. The authors declare that they have no conflict of interest.

Acknowledgements. This work was supported by Swiss National Science Foundation #192296.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

References

Coltice, N., Husson, L., Faccenna, C., and Arnould, M.: What drives tectonic plates?, Science Advances, 5, eaax4295, https://doi.org/10.1126/sciadv.aax4295, publisher: American Association for the Advancement of Science Section: Research Article, 2019.

Crameri, F., Schmeling, H., Golabek, G. J., Duretz, T., Orendt, R., Buiter, S. J. H., May, D. A., Kaus, B. J. P., Gerya, T. V., and Tackley, P. J.: A comparison of numerical surface topography calculations in geodynamic modelling: an evaluation of the 'sticky air' method, Geophysical Journal International, 189, 38–54, https://doi.org/10.1111/j.1365-246X.2012.05388.x, 2012a.

Crameri, F., Tackley, P. J., Meilick, I., Gerya, T. V., and Kaus, B. J. P.: A free plate surface and weak oceanic crust produce single-sided subduction on Earth, Geophysical Research Letters, 39, https://doi.org/10.1029/2011GL050046, _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2011GL050046, 2012b.

Duretz, T., May, D., and Yamato, P.: A free surface capturing discretization for the staggered grid finite difference scheme, Geophysical Journal International, 204, 1518–1530, https://doi.org/10.1093/gji/ggv526, 2016.

Gerya: Introduction to Numerical Geodynamic Modelling, Cambridge University Press, ISBN 978-1-107-14314-2, https://sfx.ethz.ch/sfx_lib4ri?url_ver=Z39.88-2004&ctx_ver=Z39.88-2004&ctx_enc=info:ofi/enc:UTF-8&rfr_id=info:sid/sfxit.com:opac_856&url_ctx_fmt=info:ofi/fmt:kev:mtx:ctx&sfx.ignore_date_threshold=1&rft.object_id=4930000000052441&svc_val_fmt=info:ofi/fmt:kev:mtx:sch_svc&, 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, bandiera_abtest: a Cg_type: Nature Research Journals Number: 7577 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Geodynamics;Tectonics Subject_term_id: geodynamics;tectonics, 2015.

Hagen, O., Flück, B., Fopp, F., Cabral, J. S., Hartig, F., Pontarp, M., Rangel, T. F., and Pellissier, L.: gen3sis: A general engine for eco-evolutionary simulations of the processes that shape Earth's biodiversity, PLOS Biology, 19, e3001 340, https://doi.org/10.1371/journal.pbio.3001340, publisher: Public Library of Science, 2021.

Joe, B.: GEOMPACK — a software package for the generation of meshes using geometric algorithms, Advances in Engineering Software and Workstations, 13, 325–331, https://doi.org/10.1016/0961-3552(91)90036-4, 1991.

Kaus, B. J., Popov, A. A., Baumann, T., Pusok, A., Bauville, A., Fernandez, N., and Collignon, M.: Forward and Inverse Modelling of Lithospheric Deformation on Geological Timescales Forward and Inverse Modelling of Lithospheric Deformation on Geological Timescales, in: Proceedings of nic symposium, vol. 48, pp. 978–983, John von Neumann Institute for Computing (NIC), NIC Series, https://juser.fz-juelich.de/record/507751/files/nic_2016_kaus.pdf, 2016.

Kreyszig, E.: Advanced Engineering Mathematics, John Wiley & Sons, ISBN 978-0-470-45836-5, google-Books-ID: UnN8DpXI74EC, 2010.

Larionov, E., Batty, C., and Bridson, R.: Variational stokes: a unified pressure-viscosity solver for accurate viscous liquids, ACM Transactions on Graphics, 36, 101:1–101:11, https://doi.org/10.1145/3072959.3073628, 2017.

Ogawa, M., Schubert, G., and Zebib, A.: Numerical simulations of three-dimensional thermal convection in a fluid with strongly temperature-dependent viscosity, Journal of Fluid Mechanics, 233, 299–328, https://doi.org/10.1017/S0022112091000496, 1991.

https://doi.org/10.5194/egusphere-2025-6546
Preprint. Discussion started: 23 January 2026

O'Rourke, J., Chien, C.-B., Olson, T., and Naddor, D.: A new linear algorithm for intersecting convex polygons, Computer Graphics and Image Processing, 19, 384–391, https://doi.org/10.1016/0146-664X(82)90023-5, 1982.

Patankar, S. V.: Numerical Heat Transfer and Fluid Flow, Hemisphere Publishing Corporation, New York, 1980.

Phongthanapanich, S. and Dechaumphai, P.: Adaptive Delaunay triangulation with object-oriented programming for crack propagation analysis, Finite Elements in Analysis and Design, 40, 1753–1771, https://doi.org/10.1016/j.finel.2004.01.002, 2004.

Ramberg, H.: Gravity, Deformation, and the Earth's Crust: In Theory, Experiments, and Geological Application, Academic Press, ISBN 978-0-12-576860-3, google-Books-ID: NFESAQAAIAAJ, 1981.

Renka, R. J.: Algorithm 772: STRIPACK: Delaunay triangulation and Voronoi diagram on the surface of a sphere, ACM Trans. Math. Softw., 23, 416–434, https://doi.org/10.1145/275323.275329, 1997.

Salles, T.: Badlands: A parallel basin and landscape dynamics model, SoftwareX, 5, 195–202, https://doi.org/10.1016/j.softx.2016.08.005, 2016.

Schmeling, H., Babeyko, A. Y., Enns, A., Faccenna, C., Funiciello, F., Gerya, T., Golabek, G. J., Grigull, S., Kaus, B. J. P., Morra, G., Schmalholz, S. M., and van Hunen, J.: A benchmark comparison of spontaneous subduction models—Towards a free surface, Physics of the Earth and Planetary Interiors, 171, 198–223, https://doi.org/10.1016/j.pepi.2008.06.028, 2008.

Stern, R. J. and Gerya, T. V.: Chapter 13 - Co-Evolution of Life and Plate Tectonics: The Biogeodynamic Perspective on the Mesoproterozoic-Neoproterozoic Transitions, in: Dynamics of Plate Tectonics and Mantle Convection, edited by Duarte, J. C., pp. 295–319, Elsevier, ISBN 978-0-323-85733-8, https://doi.org/10.1016/B978-0-323-85733-8.00013-5, 2023.

Tackley, P. J.: Effects of strongly temperature-dependent viscosity on time-dependent, three-dimensional models of mantle convection, Geophysical Research Letters, 20, 2187–2190, https://doi.org/10.1029/93GL02317, _eprint: https://onlinelibrary.wiley.com/doi/pdf/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, Physics of the Earth and Planetary Interiors, 171, 7–18, https://doi.org/10.1016/j.pepi.2008.08.005, 2008.

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, doi:10.1029/2001GC000 214, 2003.

Tezduyar, T. E.: Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces, Computer Methods in Applied Mechanics and Engineering, 195, 2983–3000, https://doi.org/10.1016/j.cma.2004.09.018, 2006.

Tobis, M., Schafer, C., Foster, I., Jacob, R., and Anderson, J.: FOAM: Expanding the Horizons of Climate Modeling, in: SC '97: Proceedings of the 1997 ACM/IEEE Conference on Supercomputing, pp. 27–27, https://doi.org/10.1145/509593.509620, 1997.

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, 1996.

Zerkle, A. L.: Biogeodynamics: bridging the gap between surface and deep Earth processes, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 376, 20170 401, https://doi.org/10.1098/rsta.2017.0401, 2018.