Response to Reviewers – gmd-2025-30039

Dear Dr. Räss

We would like to thank you and the reviewers for the time and effort invested in evaluating our manuscript. We appreciate the constructive and detailed comments which, we believe, helped us significantly improve the clarity and robustness of the paper.

In this revised version, we have carefully addressed all reviewer comments. The manuscript has been substantially revised to improve the description of the semi-Lagrangian (SL) implementation and strengthen the comparison with the discontinuous Galerkin (DG) advection scheme. In particular, we have added additional explanations regarding the implementation of the method, improved the discussion of numerical diffusion and computational cost, and extended the application to include DG simulations alongside the SL experiments in the Antarctic damage-transport case. In light of this additional DG analysis, we have also slightly revised the manuscript title to "A semi-Lagrangian advection scheme in Elmer (v26.1): benchmarking against discontinuous Galerkin and application to ice-damage transport", which we believe better represents the scope and content of the study. We have also clarified the parallel implementation and scalability aspects of the solver.

We have also slightly revised the initial state used in the Antarctic damage-transport case following minor adjustments to the model initialization. These changes were introduced to ensure that all solvers operate correctly with the halo elements required by the DG solver when using a partitioned mesh.

We also thank Dr. Kerkweg for clarifying the GMD policy regarding code archiving and versioning. The simulations presented in the manuscript were produced using Elmer/Ice commit 97773a1f2 (November 2024) alongside personal fixes and debugs of solvers used in our simulations. In the previous submission, the Zenodo archive cited corresponded to an earlier release of the code (2023). To comply with the GMD reproducibility policy, we have now created a Zenodo archive (post v26.1) containing the last versions of all solvers allowing to properly run the simulations (https://doi.org/10.5281/zenodo.19051131). The "Code and data availability" section of the manuscript has been updated accordingly.

Below, we provide a detailed, point-by-point response to all reviewer comments. Reviewer comments are reproduced in black, followed by our responses and a description of the changes made in the manuscript in blue.

Response to Reviewer 1

General comments

The authors present a particle-based Lagrangian scheme for advecting damage along glaciers. While the method is interesting, and has certain advantages over existing schemes, the description of how this novel scheme is coupled to existing Eulerian finite element methods could be improved. Additionally, the authors compare results to existing methods for simple cases, but this comparison is missing for application-relevant cases, making it unclear under which circumstances the presented scheme is an improvement. More specific comments, in order of occurrence:

We sincerely thank the reviewer for their detailed and constructive feedback. Their comments helped us clarify the coupling between the Semi-Lagrangian (SL) and Eulerian FEM formulations. We have revised Section 2.1 to explicitly describe how the SL scheme couples to Elmer's Eulerian FEM solver.

We note that the missing application of DG in our ice damage modeling has also been raised by the other reviewer. As requested, we have included a comparison between the SL and DG for this last application and an additional discussion that should address both reviewers' comments consistently.

We believe the revisions and additions substantially strengthened the manuscript and clarify all points raised by the reviewers.

Technical comments

Line 24: "can be solved using different approaches": It might be more appropriate to refer to this as "using different reference frames", as Eulerian and Lagrangian does not refer to a specific method/approach, but merely as to whether the mesh deforms with the material (or in the case of particle-based Lagrangian methods, follows the particles) or is static as material moves through it.

Thanks for pointing us to a better wording. We have corrected the references to Lagrangian/Eulerian method as "frameworks" rather than "methods" where needed in the manuscript. We have kept the reference to Semi-Lagrangian and Discontinuous Galerkin as "methods".

Line 29-30, "In this case, the transport equations are expressed in terms of partial differential equations (PDEs) that describe the spatial and temporal variations of the transported quantities": This is also the case for Lagrangian descriptions, and has nothing to do with the difference between Lagrangian vs Eulerian reference frames. This is the difference between continuum models (either Lagrangian or Eulerian) and particle-based methods (e.g. discrete element methods).

We agree with the reviewer. Our explanation was more to describe the difference between the PDE as in Eq. 1 with respect to the SL description of the same Eq. in Sec. 2.1. We have revised this sentence and some others of this section following this comment and comment on line 55. The main revised paragraphs are presented here:

"Transport processes in a continuum can be described by the advection (or transport) equation, which governs the spatial and temporal evolution of a scalar or tensor quantity q under a velocity field $\boldsymbol{u}$*:*

$$\frac{\partial q}{\partial t} + \nabla \cdot (\boldsymbol{u} q) = S$$

with S *a possible source or sink terms. This partial differential equation (PDE) can be expressed in different reference frames.*

In a Eulerian description, the equations are solved on a fixed spatial mesh, observing how q changes as material flows through control volumes. The numerical discretization and resolution of this equation, such as the Galerkin method widely used in Finite Element Methods (FEMs) […]."

Line 34-35, "Finite Element Models": The acronym FEM stands for Finite Element Methods

This has been corrected in the new version of the manuscript.

Line 37-41: SUPG does not introduce a diffusion term, it modifies the test functions to prioritize information from upstream. One of the terms resulting from this corresponds to a diffusion-like term, but as it is consistently applied to all terms within the PDE, this added diffusion goes towards zero as the residual of the solution approaches zero (e.g. upon mesh and time step refinement). As such, it does not add "significant numerical diffusion", but instead adds the bare minimum amount of diffusion to prevent oscillations as a result of advective terms, with the magnitude of this diffusion-like-term scaling with the velocity and used mesh size.

We agree with the reviewer that our description of SUPG was missing an important point. The SUPG description has been rewritten to accurately describe its residual-based stabilization mechanism. The new text reads:

"[…] a common solution is to rely on the streamline upwind Petrov-Galerkin (SUPG) method (hughes, 1987), which modifies the test functions in the weak formulation, introducing a residual-based term that acts as directional diffusion along streamlines. Theoretically, this diffusion vanishes as the residual approaches zero upon mesh and time-step refinement. In practical simulations, however, the residual remains finite and the term effectively behaves as a form of artificial diffusion, introducing numerical smoothing that can reduce the accuracy of sharp gradients."

Line 53-55, "The Lagrangian approach takes a different perspective. Rather than solving equations on a fixed grid or mesh, it directly tracks the motion of individual particles throughout their trajectories defined by a flow field": This is only tangentially related to the use of a Lagrangian reference frame, it might be worth clarifying that the authors mean particle/discrete element based approaches, not the use of a Lagrangian reference frame.

We agree that our description was focused on particle/discrete element-based approaches rather than the Lagrangian reference frame. Following different comments from the three reviewers we have substantially reworked the end of the introduction on the Lagrangian, particle-based and semi-Lagrangian approaches.

Line 56, "However, the Lagrangian approach may face challenges when dealing with large-scale simulations due to the large number of particles that need to be tracked.": Please provide a reference on this, as this does not seem considerably different from the challenges faced with Eulerian formulations where they are restricted due to the size of meshes used. Particle-based approaches are typically extremely efficient, scaling well to high-performance computing systems.

We agree that particle-based methods scale well, as we have actually showed for our own SL implementation in Elmer. However, they can still be computationally expensive at large scales because of the number of tracked fields, communication and, and interpolation cost between particles and mesh (especially in hybrid methods like our SL method). We clarify this in the revised manuscript.

66, "https://csc.fi/elmer/": This link leads to an error 404 page. The authors might want to replace this to use a proper doi, e.g. using https://doi.org/10.5281/zenodo.7892181 (or adding a proper citation).

Sorry for this dead link, we corrected this in the new manuscript.

Section 2.1: Could the authors clarify if the particles are initialized once at the nodes, and then tracked throughout the simulation, or if they are initialized at every time increment. Additionally, as the scheme locates the particles at the nodes, not at integration points, please clarify how the fields

advected via the particles are used within the velocity updates (which requires integration-point values). Are the nodal values interpolated using DG shape functions? (and if so, wouldn't this add a considerable amount of diffusion every time increment by interpolating between the nodal values each time step?)

We clarify here that particles are reinitialized at each time increment (e.g. for the two first experiments) or at a user-defined interval (e.g. for the third experiment). Advected quantities are interpolated from the previous time step using CG shape functions, not DG. Values are interpolated to integration points for source-term evaluation.

Figure 1: Is this figure correct in showing that the velocity and time increment are sufficiently high to advect particles over multiple elements? Wouldn't this cause instability issues related to CFL conditions within the solver for the fluid?

The figure and the test setup are correct. In the 2-D rotation test the velocity field is prescribed analytically and is steady; it is not produced by an explicit time-dependent flow solver, therefore standard CFL stability constraints for explicit solvers do not apply. In our last (and realistic) application we solve the steady/implicit Stokes equations (an elliptic problem) for which there is no explicit CFL time-step restriction as in explicit hyperbolic solvers There are therefore no stability issues.

Figure 2 and 4: These results indicate that the presented scheme is not suitable for advecting quantities over any reasonable amount of time increments. While the DG scheme allows for time refinement to obtain more accurate solutions (which is typically expected within FEM, and follows a similar trend to fluid mechanics solvers improving in accuracy with smaller increments), it seems SL offers no possibility to get more accurate results. Could the authors add a brief discussion on which cases the behaviour of the SL scheme would be preferable over the DG scheme?

It is true that the DG scheme allows for time refinement to obtain more accurate solutions, while the solution does not really profit from an increased spatial resolution. The SL shows the opposite with longer timesteps and/or better spatial resolution leading to better solutions, as both reduce the cumulative interpolation error.

This is illustrated in Fig. 3, which quantifies the visual diffusion we observe in Fig. 2, we can see that in terms of RMSE(q), the SL method does a better job that DG, when $\Delta t > \frac{2\pi}{250}$ at R_{s1} and $\Delta t > \frac{2\pi}{500}$ at R_{s2}.

The higher RMSE of SL at smaller time steps results from the increasing number of interpolations, which would occur in a coupled setting. This interpolation-driven diffusion motivated our choice in the final application to test the impact of a lower coupling frequency between the Stokes solver and the SL damage advection. We have reworked the Sec. 3.1.2 to better explain the limits of both methods.

Section 4: Would the authors be able to also present results for this case using DG? As there is no "correct" solution to compare the predictions to, it would be nice to compare the two to see how much the results differ for realistic cases, especially for cases where the time increment of the flow and advection solvers match in time increment (e.g. Fig 6c).

As requested, we included this comparison in Sec. 4. We note that this point is also raised by Reviewer 2, and our additional discussion now addresses both reviewers' comments consistently.

Line 264, "small time steps (e.g. Δt ∼ 10−2 km)": I presume the authors mean 10^-2 a?

Thanks for noticing this typo. It has been corrected.

Line 267, "In this study, we do not account for feedbacks between damage and viscosity (Eq. (B4)), so the reduced update frequency of damage does not impact the flow solution.": This seems a major shortcoming of the presented scheme. Was the decision to have these fields remain uncoupled decided a priori, or did the authors try this coupling and run into issues?

There might be a misunderstanding here. The decision of having these fields uncoupled (i.e., no feedback of ice damage on the ice viscosity used for the Stokes flow resolution), was made a priori. Indeed, since we assess the SL performance based on the frequency of a potential coupling between the damage and ice flow, activating this coupling would affect differently the ice flow for each simulation, hence also modifying the subsequent ice damage and the computation of the source term. Activating this feature can be done with no issue but would lead to confusion between the diffusion of the solution, the evolution of the flow field, and the evolution of the source terms over time. We have made this clearer in the new version of the manuscript.

Lines 331-336: Could the authors expand on this more? Typically Lagrangian formulations should be perfectly conservative. Is this related to how the Lagrangian particles are coupled and updated (in which case, please expand on how this is done, and why it makes the scheme non-conservative)?

Yes, we agree that Lagrangian formulations are conservative. It is also true that the non-conservation of the SL algorithm is actually due to the coupling with the flow and the reinitialization of the particles. Each time a particle value is update to the FE mesh, the conservation is not strictly ensured because field values are reconstructed by interpolation rather than flux balancing, introducing small non-conservative errors. One solution would be to introduce a conservative interpolation in Elmer. While this is one of our goals for the future (Sec. 5 Discussion and Future adaptations), we remind that conservative interpolations such as flux balancing or remapping schemes that redistribute quantities to neighboring elements, could introduce other numerical biases (e.g., artificial oscillations or loss of monotonicity).

Line 340: "and ensuring the stability of the advection with respect to Eulerian FEM implementations of SUPG methods": Eulerian methods with SUPG are perfectly stable in advective cases. Do the authors mean limiting the diffusion compared to SUPG instead?

We thank the reviewer for this clarification. We agree that Eulerian formulations using SUPG stabilization are generally stable for advection-dominated problems. Our intention was not to question the stability of SUPG, but rather to highlight that, in our experience with Elmer, the SUPG formulation introduces excessive numerical diffusion when advecting sharp features, such as damage. We have therefore revised the sentence to correctly state that the SL method ensures "the stability of the advection while limiting numerical diffusion when advecting sharp features". We do not directly refer to SUPG here to avoid confusion since different implementations from the one in Elmer could also help limiting diffusion. For completeness, we note that we still use SUPG for ice-thickness advection, where gradients are smooth and the method performs well.

Line 346, "increasing time steps and increasing spatial resolution": Please clarify if the authors mean "increased time step size" or "increased amount of time steps" (presumably, the former).

Yes, we meant increasing time step size. We made the precision in the new version of the manuscript.

Line 362 onwards: Why would the SL scheme benefit from increased computational power? It seems DG would benefit from this (smaller and more time increments) whereas SL is more appropriate for when only a reduced number of time steps are taken, thus having less computational cost.

We agree that both schemes can benefit from increased computational power, though in different ways. The SL method benefits most from higher spatial resolution and a larger number of particles, which reduce interpolation errors—the main source of diffusion in the SL approach. In contrast, DG (as implemented in Elmer) shows limited improvement with spatial refinement (see Fig. 2); its accuracy primarily depends on the time-step size. Consequently, when more computational resources are available, the SL method can typically exploit them more effectively—by increasing spatial resolution without reducing the global time step—whereas DG is constrained by its time-stepping stability limit.

Eqs. B7 and B9: Please use \cdot to indicate dot products (as is done in B1)

This has been be fixed. Thanks.

The authors might want to check their reference list more carefully, e.g. on line 545 "slope limiter for p<math><mi is="true">p</mi></math>-adaptive discontinuous", line 547 where a full title is given in capitals, or the many entries where journal articles are formatted as references to book chapters

Thank you for pointing these inconsistencies, we have corrected and updated the bibliography.

Response to Reviewer 2

We thank Albert de Montserrat Navarro for their constructive and technically detailed review. Their feedback allowed us to clarify the implementation details, improve consistency, and expand the discussion of particle-based methods and parallelization. Below we address each of their general comments, as well as their specific comments.

General Comments

1. The manuscript discusses the Discontinuous Galerkin (DG) and Semi-Lagrangian (SL) advection methods. However, the Particles-in-Cell (PiC) approach (also known as Markers-in-Cell) is another particle-based technique that is widely applied in fields such as fluid and plasma dynamics as well as computational geodynamics. This method is known to produce substantially less numerical diffusion and can capture certain subgrid-scale features more accurately. Its implementation would likely be straightforward, as many components of the current SL scheme could be reused. I therefore recommend that the authors include a short discussion explaining why the PiC method was not considered, or outlining why it may not be suitable for their particular application.

We thank the reviewer for this insightful suggestion. We agree that the Particles-in-Cell (PiC, or Markers-in-Cell) method is closely related to our Semi-Lagrangian (SL) scheme and could, in principle, replace it. Elmer already contains the code for forward particle-tracking that could serve as the basis for such an implementation. The PiC approach has clear advantages, such as avoiding the need to reinitialize particles at each time step and improving reversibility between backward tracking and forward integrations.

However, a PiC implementation would require initializing a large number of particles per element to ensure statistically meaningful averages (since the sampling error scales as $1/\sqrt{N}$). This considerably increases the computational cost. Particle-location searches are also expensive in FEM unstructured meshes. In our current Elmer framework, which is optimized for FEM-centric data structures (element connectivity, handling of boundaries, …), PiC simulations are relatively slow for this reason. Implementing an efficient PiC scheme would likely require substantial restructuring of the FEM data structure to minimize the cost of particle-search.

A short discussion has been added to Section 5 of the revised manuscript explaining why the PiC approach was not implemented. We highlight its conceptual similarities with the SL formulation, as well as its additional complexity and computational demands. We also note that PiC schemes can face the same conservation issues as SL and would require additional flux-balancing steps. A conservative PiC extension is therefore identified as a potential direction for future work:

"Finally, particle–grid approaches such as Particle-in-Cell (PiC) or Markers-in-Cell, introduced in Sec.~1, could provide an alternative strategy for tracer advection. In these methods, particles are advected forward in time while carrying tracer properties, and their values are periodically projected back onto the Eulerian mesh using conservative weighting, which can improve mass conservation and represent sub-grid variability. However, such schemes typically require maintaining a large number of particles per element to ensure adequate sampling, and particle–mesh communication as well as particle repopulation can become costly on unstructured meshes. While implementing an efficient PiC framework in Elmer would require substantial modifications to the current FEM-based data structures, they represent a promising avenue for the future evolution of Elmer particle-based implementations".

2. It is not entirely clear whether the Stokes solver also employs DG or another FEM formulation. Providing a brief and explicit description of the Stokes solver in Elmer would improve the manuscript's clarity. If DG is also used for the flow solution, the velocity field would be discontinuous across element boundaries. In this case, when combining DG with the SL method, it would be useful to clarify how this discontinuity is handled. Do the authors use the discontinuous field directly, or is the velocity averaged (e.g., at element nodes) before advection?

   The Stokes equations are solved using continuous Galerkin FEMs with linear equal-order elements for velocity and pressure. The velocity field is continuous, ensuring smooth particle trajectories. DG is only used for advection, not for the flow solution. We added this specification by adding a comment to section 2.1 to explain how the velocity field is obtained:

   "The velocity and pressure fields are typically solved following Stokes-like equations with Elmer's standard continuous Galerkin formulation with stabilized equal-order linear elements. The

resulting velocity field is continuous at element nodes, ensuring smooth trajectories for particle advection."

And one to section 2.2 to explain that the velocity field is the same for the SL and DG method.

"Similarly to the SL algorithm, the flow solution does not rely on a DG formulation; only the advection equation employs a DG formulation."

3. The manuscript refers to "linear square elements" and "linear edge elements" in the numerical experiments. Are these the same for both velocity and pressure fields? Additionally, line 329 mentions that wedge elements have 44 integration points—are these the same "linear" elements referred to earlier? It would be beneficial to provide a concise description of these elements along with the relevant Elmer implementation details.

   We have clarified that all simulations use linear Lagrange elements. The 2D test uses quadrilateral elements ('linear square') while the 3D synthetical test use hexahedral elements. The 3D application to ice damage in the Amundsen Sea sector uses linear wedge elements but with 44 integration points for quadrature. For the ice damage application, we have added the following sentence to the description of the numerical setup:

   "The model is discretized on an unstructured finite-element mesh using stabilized equal-order linear wedge elements for velocity and pressure. Numerical integration is performed using 44 Gauss points per element (11 points on the triangular face and 4 along the vertical direction). Although the velocity and pressure are approximated with first-order Lagrange elements, this relatively high quadrature order is required to accurately integrate the nonlinear viscosity and stabilization terms within each element. Stresses are then computed at these integration points and subsequently interpolated to the mesh nodes, where the damage variable is evaluated."

4. In Section 3.2.2, the authors state that the SL scheme seldom randomly leads to either an artificial increase or decrease of the advected field accumulated over the whole domain, and this potentially being caused by the number of internal time steps or loss of particles. Figure 5 shows ± oscillations over time for all time steps used, so assuming the number of internal time steps remains constant throughout the simulation, it is unclear why such oscillations would occur as a result of the number of internal time steps. Could the authors elaborate on this behavior? Similarly, the observed particle loss when moving across partitions requires clarification. If particles are correctly reassigned to their respective MPI ranks after each internal time step, such losses should not occur.

   Oscillations stem from interpolation and rounding errors during particle reinitialization, not from sub-step count. Particle loss occurs when particles cross MPI domain boundaries between communication steps. These particles are reassigned at the next synchronization (i.e., timestep). We have added the following paragraph in Section 3.2.2:

   The small oscillations visible in Fig. 5 arise from interpolation and rounding errors that accumulate as particles are reinitialized at each simulation time step (Δt). The number of internal sub-steps (δt , kept constant here) influences the oscillation amplitude only indirectly, through its effect on trajectory accuracy. The small particle losses observed in parallel runs result from particles crossing sub-domain boundaries between MPI synchronizations; when communication is performed only

after the final internal sub-step, a few particles can transiently exit the active domain before being reassigned at the next exchange. These events are rare and decrease with more frequent synchronization.

5. In Section 4, only the SL method is applied. It would strengthen the study to include a comparison using the DG method for the same experiment, thereby demonstrating the relative performance of both methods in a realistic application, rather than only in synthetic tests. This would better illustrate the practical trade-offs between the two schemes.

We note that this comment overlaps with Reviewer 1's request. As requested, we have included this comparison in Sec. 4.

Minor technical and stylistic corrections

All stylistic and consistency issues in the revised manuscript have been corrected, including:

- Abbreviation: we agree that both "a" (for annum) and "yr" (for year) are used in the literature. "a" is widely adopted in glaciology and geosciences and is consistent with conventions in The Cryosphere (EGU journal) and other journals. It seems that yr is preferred in GMD and we have made the correction following the reviewer suggestion.
- Line 35: FEM acronym corrected
- Line 96: Thank you for this suggestion. We did not test an explicit RK4 integrator, as we do not expect it to significantly improve the overall accuracy of the Semi-Lagrangian (SL) scheme. When a sufficiently large number of internal time steps is used for particle trajectory integration, the dominant source of error in the SL method arises from the interpolation between particles and the Eulerian mesh rather than from the trajectory integration itself. RK4 integrator could be useful in certain situations: (1) very long simulations where errors accumulate over time, (2) a small number of internal time step is used, or (3) in complex flow fields with rapid directional variations. However, we expect that increasing the number of timesteps is a more effective strategy.
- Line 113-114: We have removed the term 'traditional cell search' and rewrite the sentence: *"The simplest way to find this position is to localize the particle's new location using the coordinate of the particle to retrieve its position into the mesh."*
- Line 159 and elsewhere: RMS consistently renamed RMSE.
- Line 165: We add a short introduction to Section 3: *"This second experiment extends the analysis to a three-dimensional setting and focuses on the influence of time stepping,*
- *parallel domain decomposition, and computational cost on tracer advection."*
- Line 200: Clarified 3D slab-flow setup (layers of prismatic elements) by adding the sentence: "We use 3-D linear hexahedral elements obtained by vertically extruding the 2-D rectangular mesh into three layers"
- Sec. 3.2.1.: Yes, we use 3D elements constructed by extruding the 2D squares on the vertical axis. We have made that clearer in the new version of the manuscript.
- Line 210-213: we added a short paragraph describing that simulations use distributed-memory MPI only (No OpenMP is used), with communication performed after each internal sub-step. It also refers to a new appendix (A3) testing the scalability of the SL algorithm.

- Tab. 1: We also explained that slight decreases in computation time per particle with N (Table 1) mostly results from the interpolation, which is done one time per time step, whatever the value of N. The SL average time step time (s) follows the relation: $t = t_{interp\ and\ other\ operations} + N \times t_{\delta t\ advection}$ with $t_{interp\ and\ other\ operations} \sim 0.035$ s and $t_{\delta t\ advection} \sim 0.075$ s, resulting in :
    - $t(N = 1) = 0.035 + 0.075 = 0.11$ s
    - $t(N = 5) = 0.035 + 5 \times 0.075 = 0.41$ s
    - $t(N = 10) = 0.035 + 10 \times 0.075 = 0.78$ s

  The reviewer is therefore right to think that the time per N is approximatively constant. We added the following sentence to clear this:

  "This behavior is approximately linear, as interpolation and mesh-related operations are performed once per simulation time step, whereas the particle advection cost scales with N"

- Sec. 4.1: The grid resolution and the type of elements is presented in Appendix B2 but, following the reviewer's suggestion we added a short summary in Sec 4.1. We also better explained how the damage variable is computed:
  "The model is discretized on an unstructured finite-element mesh using stabilized equal-order linear elements for velocity and pressure. Once initialized, we use the model to simulate the evolution of damage over 50 years. […] The damage source term is evaluated at the nodes, using stresses interpolated from integration points.

We also accounted for the remaining minor comments in the reviewer's annotated version of the manuscript:

- "this may be a bit misleading, SL does not require any particles". We agree with the reviewer that this wording may be misleading. In a strict sense, Semi-Lagrangian schemes do not require particles, as they are based on characteristic back-tracing rather than on discrete material. In the present implementation, the SL scheme is realized using a particle-based formulation to approximate the characteristic paths and source-term integration. We have clarified this distinction at the end of our introduction and consistently refer to this approach as a particle-based implementation of a Semi-Lagrangian method.
- In the Appendix of ice flow model initialization: "what do you mean by 2D-plan? you mean a mesh of the surface". Not exactly, it is a 2D-plan mesh of the footprint of the glacier basin, it is then extruded between the upper surface and the lower surface (described through a DEM).
- Figure 4. The DG solution looks smoother at t0 because of how the field is represented and visualized in DG, not because diffusion has already occurred.
- The damage source is computed at the node. We have made this clearer with the following sentence: *"[…] The damage source term is evaluated at the nodes, using stresses interpolated from integration points."*.
- Regarding the construction of the 3D mesh, the domain is first discretized as a 2D footprint using triangular elements. This 2D mesh is then vertically extruded into 11 layers, with the bottom and top surfaces adjusted to match the glacier bed and surface digital elevation models. This has been clarified in Appendix B2 with this updated sentence:

"The 2-D mesh is then vertically extruded into 11 layers, with the bottom and top surfaces adjusted to match the glacier bed and surface digital elevation models, leading to a vertical resolution ranging from 10 to 300~m locally depending on the ice thickness"

We thank the reviewer again for their detailed review. Their suggestions have substantially improve the clarity, completeness, and technical precision of the manuscript.

Response to Reviewer 3

This manuscript evaluates a semi-Lagrangian (SL) advection scheme that seems to be already developed and implemented in Elmer/Elmer-Ice. The results from the SL scheme are then compared with those from the discontinuous Galerkin (DG) method, which is also already implemented in Elmer/Elmer-Ice. The authors evaluate both approaches using two benchmark test (bodies in rotation and donut transport) and apply the SL method to simulate ice-damage transport in Amundsen Sea Sector of Antarctica. The study addresses a persistent numerical challenge in ice sheet modeling – to maintain accuracy and reduce spurious diffusion while lowering computational cost and ensuring stability – particularly when advecting sharp crevasse features.

My main concern/comment is that the amount of new contribution made in this article is not significant in terms of model development as the SL and DG methods were previously developed and implemented in Elmer/Elmer-Ice. Also, the results from the two benchmark tests visually seem to show excessive diffusion although the RMSE is less than 5%, which I found a bit confusing. Moreover, the Antarctic ice damage study is too simple as it lacks the coupling between ice flow and damage processes. Perhaps, the most significant finding of this article is that SL scheme suffers from less spurious diffusion than the DG method with larger time steps, which is useful information for the ice sheet modelers. However, it is not clear how one would estimate the optimal time step size with the SL scheme in a general continental-scale ice sheet simulation. Is there a theoretical way of estimating the optimal time step size or does one have to figure by trial and error.

Looking at the other two reviews, it is fair that the authors get a chance to respond to reviewer comments and make substantial revisions. I have a few detailed comments for the authors to consider below. Based on the authors' response regarding the paper's new contribution, I could be convinced that the article warrants publication.

– Ravindra Duddu

We thank the reviewer for this important comment regarding the level of new model development. We agree that both the DG and SL schemes already existed in Elmer. It is true that we did not introduce any new changes on the DG scheme over the course of this study but a substantial rework of the SL scheme has been conducted. We understand the reviewer's confusion, as the SL solver has indeed been available for many years. This scheme was originally implemented by one of the main authors (Peter Råback). However, several important limitations that had remained undetected at the time made the solver unsuitable for the advection of active tracers. In particular, these limitations affected the correct treatment of source terms.

As part of this work, we implemented and validated the following major improvements:

- Fixes for source terms that explicitly depend on the advected variable

- Improvement in the path-integral formulation by allowing forward integration of the source term, as schematically presented in Figure 1.
- Better handling of initial conditions, which allows for simulation restarts that were problematic.
- Better handling of boundary conditions that were leading to undesired particle loss
- Fixes related to parallel execution and MPI communication.

The activity on the Elmer Github repository for the ParticleUtils.F90 and the ParticleAdvector.F90 documents the progression of these developments. These fixes have been made possible through extensive testing in both idealized and more realistic configurations, with iterative refinement of the implementation.

In addition to these developments, our study **(i)** provides the first **systematic quantitative comparison of SL and DG for sharp, damage-like tracers under ice-flow conditions**, **(ii)** identifies their **opposite convergence behaviors with respect to time stepping and spatial resolution (as mentioned by the reviewer),** and **(iii)** proposes practical guidelines for using SL in the context of ice-damage evolution.

This work also presents the **first 3D and large-scale application of Semi-Lagrangian damage advection in Elmer/Ice.** This application reveals practical limitations related to coupling frequency, conservation errors, and computational cost. These insights are directly relevant for the ice-sheet modeling community and go beyond what is currently documented in the literature for SL schemes in glaciological damage transport.

The reviewer also raises a fair point about the *estimation of the "optimal time step size with the SL scheme in a general continental-scale ice sheet simulation".* It is important to distinguish between the global coupling timestep used to synchronize advection with the flow solver, and the internal substepping employed by the SL solver. The SL solver allows several strategies for selecting internal substeps (δt), including fixed fractions of the global timestep, bounds on physical displacement, or limits on the local Courant number. These criteria can improve the accuracy and robustness of particle trajectories but do not alter the frequency at which the advected field is coupled to the flow.

The choice of the global coupling timestep (Δt) remains governed by physical and modeling considerations, such as the timescale of damage evolution, consistency with the Stokes solver, and acceptable coupling error. In simple and idealized configurations, an optimal timestep can in principle be identified. In practice, such an optimal SL timestep satisfies $\Delta t = k\frac{\Delta x}{|\boldsymbol{u}|}$, where $k \in \mathbb{Z}$ is chosen such that particle displacements remain comparable to the local element size.

In large-scale ice-sheet simulations, however, heterogeneous velocities and mesh resolutions make it impractical to define a universal coupling timestep based on local Courant criteria. As a result, selecting the main timestep inevitably relies on sensitivity tests rather than a theoretical optimum, with the goal of balancing interpolation-induced diffusion against physically meaningful coupling frequency. An alternative strategy would be to adapt the spatial resolution to the SL scheme by varying Δx, since locally adapting k would introduce physical inconsistencies, as particles/ nodes would be updated at different rates. Systematically designing meshes that adapt Δx to local flow speeds would require additional development –and potential remeshing may itself introduce diffusion diffusion– and therefore lies beyond the scope of the present study.

We also understand the reviewer's interest in a fully coupled damage–viscosity formulation. The present application is intentionally simplified and should not be interpreted as a physically complete damage simulation. Introducing damage–viscosity coupling would make it difficult to isolate numerical

diffusion effects, as damage could then develop outside the prescribed source regions through dynamical feedbacks. Our goal here is to characterize the numerical limitations of the SL method so that Elmer/Ice users can more reliably assess and design future simulations of ice-damage evolution.

We believe that our work will facilitate further development of particle-based studies in Elmer/Ice and help future users make informed methodological choices when using the solver. It also improves awareness of accuracy and spurious diffusion issues that are often underdiscussed in ice sheet modeling studies using advection schemes.

Detailed comments

1. Page 2, line 45 – I would assume the SL scheme and SUPG have a long history of application in geosciences and glaciology. The limitations of SUPG can be overcome with newer methods based on variational multiscale (VMS) stabilization, which reduces cross-wind instability and robustly handles non-uniform meshes. The authors could at the least improve the literature review here and cite the following works.

   Masud, A., & Khurram, R. A. (2004). A multiscale/stabilized finite element method for the advection–diffusion equation. Computer Methods in Applied Mechanics and Engineering, 193(21-22), 1997-2018.

   Cheng, G., Morlighem, M., & Gudmundsson, G. H. (2024). Numerical stabilization methods for level-set-based ice front migration. Geoscientific Model Development, 17(16), 6227-6247.

   While we were familiar with the work of Cheng et al. (2024), we thank the reviewer for directing us to Masud et al. (2024). We have added these two cites in the new version of the manuscript. We have added the following text to the paragraph:

   Variational multiscale (VMS) stabilization frameworks have been proposed to improve upon classical SUPG formulations by better controlling cross-wind oscillations and enhancing robustness on non-uniform meshes (e.g., Masud and Khurram, 2004; Cheng et al., 2024). While these approaches significantly improve the behavior of stabilized continuous Galerkin methods in many advection-dominated regimes, they still rely on residual-based modeling and may introduce some degree of numerical diffusion, particularly when sharp fronts or strongly localized features must be preserved. This motivates the exploration of alternative strategies, such as discontinuous Galerkin and semi-Lagrangian formulations, which address advection errors through different numerical mechanisms.

2. Page 2, line 56 – Lagrangian approaches can also involve moving mesh methods (Jimenez et al., 2017), it does not have to be particle-based approaches. Semi-Lagrangian or hybrid methods also include Arbitrary Lagrangian-Eulerian (ALE) and coupled Eulerian–Lagrangian (CEL) formulations, including the material point method (MPM) that we utilized. In Huth et al. (2021a), we proposed an advection benchmark for the MISMIP+ configuration that is more relevant to ice sheet modeling than the Zalesack disc test. In Huth et al. (2021b) we show that MPM along with creep damage model can handle the feedback between ice flow and fracture, damage production and advection and ice-ocean boundary evolution. In Huth et al. (2023) we show that the creep damage model can reproduce the last two years of rift propagation in Larsen C ice shelf that led to the calving of A-68 iceberg, which is perhaps the only such prognostic modeling study in the literature. Bassis and Berg (2022) also use tracers using particles in cell method to track crevasse advection. Citing these works and putting them in context with the current work would improve the discussion of this paper and inform the reader about the pros and cons of different SL schemes.

Jiménez, S., Duddu, R., & Bassis, J. (2017). An updated-Lagrangian damage mechanics formulation for modeling the creeping flow and fracture of ice sheets. Computer Methods in Applied Mechanics and Engineering, 313, 406-432.

Huth, A., Duddu, R., & Smith, B. (2021). A generalized interpolation material point method for shallow ice shelves. 1: Shallow shelf approximation and ice thickness evolution. Journal of Advances in Modeling Earth Systems, 13(8), e2020MS002277.

Huth, A., Duddu, R., & Smith, B. (2021). A generalized interpolation material point method for shallow ice shelves. 2: Anisotropic nonlocal damage mechanics and rift propagation. Journal of Advances in Modeling Earth Systems, 13(8), e2020MS002292.

Huth, A., Duddu, R., Smith, B., & Sergienko, O. (2023). Simulating the processes controlling ice-shelf rift paths using damage mechanics. Journal of Glaciology, 69(278), 1915-1928.

Berg, B., & Bassis, J. (2022). Crevasse advection increases glacier calving. Journal of Glaciology, 68(271), 977-986.

We thank the reviewer for these valuable references, which we have added to the revised manuscript. We agree that Lagrangian and hybrid formulations include a broad range of methods beyond particle-based SL, including ALE, CEL, and MPM approaches (Jiménez et al., 2017; Huth et al., 2021a,b; 2023). These methods are particularly well suited for fully coupled fracture–damage–flow problems and evolving boundaries.

Our present focus is different, we specifically target advection of damage-like tracers within an Eulerian Stokes solver. We have expanded the discussion to better position SL relative to MPM, PiC, and tracer-based approaches (including Berg & Bassis, 2022). The paragraph now reads:

"*Particle-based approaches, which are conceptually rooted in the Lagrangian framework, offer a distinct perspective on transport processes. Instead of solving the continuum equations on a fixed spatial grid, these methods explicitly track discrete particles (or markers) as they move along trajectories defined by the flow field (Samelson and Wiggins, 2006). Physical quantities are carried by the particles themselves and evolve according to the local flow, thereby avoiding the explicit computation of the advective term on a mesh.*

More broadly, Lagrangian and hybrid formulations encompass a wide spectrum of approaches beyond particle tracking, including moving-mesh and Arbitrary Lagrangian–Eulerian (ALE) methods (e.g. Jiménez et al., 2017), coupled Eulerian–Lagrangian (CEL) techniques, and Material Point Method (MPM) formulations (e.g. Huth et al., 2021a, b, 2023). These methods have demonstrated strong capabilities for simulating fully coupled ice-flow, damage, and fracture processes, including rift propagation and evolving ice–ocean boundaries. Particle-in-cell tracer approaches have also been used to investigate crevasse advection and its impact on calving dynamics (e.g. Berg and Bassis, 2022)."

3. Figures 1–4 can be improved along with the discussion.

In Figure 1, there is not much difference in the two subfigures except for showing two different quantities – position vector r and integral I_t. Perhaps, they can be combined. Also, the SL scheme is explained with a triangular mesh in Figure 1, whereas the mesh used for Figure 2 is noted as a linear square finite element. Was there any triangulation done to apply the SL scheme.

We agree that there is little change between left and right panel. However, we think that combining the two panels make less impacting the difference between the back-tracking of the particle value at previous timestep and the along-flow integral after reversing the tracking in the algorithm. Since we think both arguments are valid to us, we let this to the appreciation of the editors.

In Figure 2, the subfigures (a) – (g) are not labelled but are referred to in the caption. The result for the SL scheme with reinitialization looks quite diffused compared to the reference solution, however, the error is less than 5% in Figure 3 which is puzzling. Please explain this better.

Thanks, we have added the subfigures labels. It is true that the figure looks quite diffuse for small SL timesteps but so does the DG for large timesteps while showing similar RMSE. This is partially due to the fact that, while the sharp U-shaped circle (located on top at the beginning and the end of the simulation) can lose its consistence rapidly, the more diffuse circles are less sensitive to diffusion, at least in regard of the RMSE metric.

In addition, the error metric in Fig. 3 is computed as a domain-wide difference between the advected field and the reference field. Because the tracer occupies only a limited fraction of the domain, large portions of the mesh remain at (near-)zero values for both the reference and the numerical solution. These regions dominate the domain-average and can lead to relatively small values even when visible diffusion occurs near the tracer boundary.

We emphasize that the error metric is not expressed in percent in its original form. Following the reviewer's comment, we verified the implementation and found that the quantity initially reported corresponded to a mean squared error (MSE) rather than a root mean squared error (RMSE). This has been corrected, leading to a maximum RMSE of approximately 0.20.

To better align with a percentage-based interpretation and avoid ambiguity, we additionally report a normalized RMSE defined as:

$$\mathrm{NRMSE} = 1/\Omega \sqrt{\int_{\Omega} (q_i - q_{ref})^2 / \max(q_{ref})}$$

In the present case, the tracer is normalized to $\max(q_{ref}) = 1$, so the NRMSE is identical to the RMSE up to the percent conversion. This clarification has been added to the figure caption and the main text to avoid confusion.

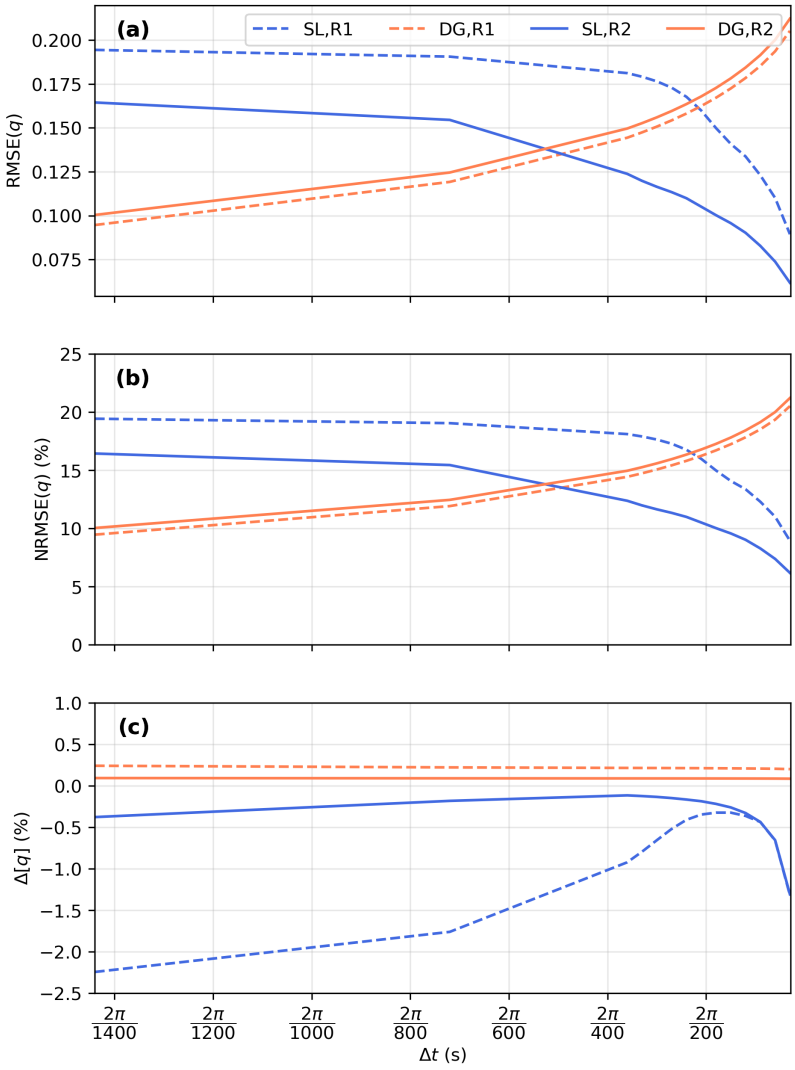

Figure. (a) RMSE, **(b)** NRMSE and **(c)** the concentration of the tracer after one revolution and for different timestep Δt.

Figure 3, please increase the font size of the text and axis labels and tick values.

In Section 3.2.2, the authors refer to Fig. 4a and Fig. 4b, but there is such labeling in Figure 4, only left and right.

Fig. 4 now contains the label (a) left column and (b) right column. Thank you.

4. Page 5 line 129 and Page 6 line 131 – I did not quite follow the argument of linear and logarithmic scaling. Isn't linear scaling better than logarithmic. Please explain.

   We agree with this statement and we believe that this is what we wrote: "*While the method requires some complex geometrical tests, it is shown that the computational time roughly scales linearly [...]. This makes the method faster than most tree-based localizations that are shown to scale logarithmically […]*"

5. Figure 3 – Typically, a numerical method for a time-dependent advection problem is designed to yield more accurate results as the time step is reduced. Depending on the rate of decay of error

with the time step size one would call it first or second order and so on. With the DG method, the error decreases with decreasing time step size, which makes sense, whereas it does not make sense that with the SL method, the error increases with decreasing time step size. Moreover, the results in Figure 4 look basically juxtaposed. The result for DG with dt = 1 looks the same as SL with dt = 0.01. This requires more mathematical analysis and better explanation, perhaps it is explained in the literature and would be useful to discuss here.

We agree that the opposite convergence behaviors of DG and SL with respect to time step size may appear counterintuitive at first sight but this is exactly one of the topic of this paper. For DG, the temporal truncation error dominates, and therefore reducing the time step naturally improves accuracy.

For the SL, when using the quadratic correction for the velocity field (activated through a logical keyword), the particle trajectory integration is formally second-order accurate in time for smooth solutions, such that the trajectory error scales as $O(\Delta t^2)$, compared to $O(\Delta t)$ without correction. Consequently, halving the timestep is expected to reduce the integration error by a factor of approximately four.

However, in the present tests, the dominant source of error is not time integration, but the repeated interpolation between the mesh and the particle (after is localization during the back-tracking process). As the time step decreases, the number of interpolation operations increases proportionally, leading to accumulated numerical diffusion. This explains why SL accuracy can degrade when Δt becomes too small, as observed in Fig. 3.

For the correspondence between the DG with dt = 1 and the SL with dt = 0.01, we believe that is mostly a coincidence but, in both cases, they represent a poor timestep for the respective schemes.

6. Page 16, line 329 – How does it help to use 44 points as the integration of quadratic elements just needs 9 Gauss points. Over-integration sometimes causes issues including numerical instability or increasing computational cost. Perhaps, it is ok to do this with SL scheme but not with DG. Please add a comment on this.

The 44 integration points are not chosen for the particle scheme itself, but follow directly from the quadrature used by the Stokes solver for triangular wedge elements (11 points on the triangle and 4 in depth), allowing for the integration of 4^{th} order polynomials. This quadrature is also required by the use of bubble-function stabilization, which increases the polynomial degree of the integrands to fourth order or higher.

Our suggestion for future work is to reuse these integration points for the SL algorithm to ensure both consistent coupling and dense sampling of the advected fields. While such a number of integration points would be unnecessary for polynomial DG quadrature –and potentially penalizing– it is appropriate in SL framework. We have clarified this by adding the following paragraph to the numerical description of the ice damage experiment (Sec. 4.1):

"The model is discretized on an unstructured finite-element mesh using stabilized equal-order linear wedge elements for velocity and pressure. Numerical integration is performed using 44 Gauss points per element (11 points on the triangular face and 4 along the vertical direction). Although the velocity and pressure are approximated with first-order Lagrange elements, this relatively high quadrature order is required to accurately integrate the nonlinear viscosity and

stabilization terms within each element. Stresses are then computed at these integration points and subsequently interpolated to the mesh nodes, where the damage variable is evaluated."

7. In Appendix A, the authors mention parallelization and the benefits of the SL scheme implementation. Table 1 states simulations were conducted with 8 partitions. Adding strong/weak scaling studies and discussing communication overhead would improve the discussion in the paper. One issue maybe that for problems with small number of degrees of freedom, parallelization may not show optimal scaling.

   We thank the reviewer for this constructive suggestion. Following this comment, we have extended Appendix A with a dedicated scalability analysis (new Appendix A3), in which we assess both strong and weak scaling of the Semi-Lagrangian (SL) advection scheme. We already showed a part of the results in our response to the discussion phase. Since then, we have doubled checked our numbers and have increase the scaling to 256 and 128 partitions.

   We believe that this additional analysis significantly strengthens the discussion of the parallel performance and practical applicability of the SL scheme.

8. I found more than a few typos in the paper. Also, some words have been unnecessarily capitalized, for example you can use lower case for "finite element method" "semi-Lagrangian" "discontinuous Galerkin" "streamline upwind Petrov-Galerkin" Only if it is based on the name of a person like Lagrange or Galerkin it warrants capitalization.

   We thank the reviewer for noting these issues. All typographical errors and unnecessary capitalizations have been corrected in the revised manuscript.