

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 will revise 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 modelling has also been raised by the other reviewer. As requested, we will include 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 will substantially strengthen the manuscript and clarified 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 will correct 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 u :

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

with q the transported quantity, \mathbf{u} the flow velocity, and S represents 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 will be 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 will be rewritten to accurately describe its residual-based stabilization mechanism. The text will read:

“The SUPG method 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. However, in practical simulations where the residual is finite, this 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. We will rephrase the paragraph as follows:

“Particle-based approaches 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 (e.g., 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. Although particle-based formulations can scale efficiently on modern high-performance computing systems (e.g. Macpherson et al., 2009;

Ketefian et al., 2016), they may still become computationally demanding in large-scale simulations because of the very high number of particles that must be tracked and interpolated at each time step. This overhead is particularly relevant in coupled Eulerian–Lagrangian frameworks, where particle advection and field interpolation introduce additional communication and memory costs.

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 will 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 will correct 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. 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 than 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 add a paragraph in the Discussion to clarify these differences and the practical cases where SL is preferable to DG:

“The Discontinuous Galerkin (DG) scheme and the Semi-Lagrangian (SL) scheme exhibit complementary numerical behaviors. The DG formulation primarily benefits from time-step refinement: as the time increment decreases, the temporal truncation error is reduced, whereas increasing spatial resolution provides only limited improvement once the polynomial order dominates. In contrast, the SL method shows the opposite trend—its accuracy improves with finer spatial resolution and with larger advection time steps, both of which reduce cumulative interpolation errors. This behavior is visible in Fig. 3, where the SL scheme yields lower RMSE(q) values than DG for $\Delta t > \frac{2\pi}{250}$ at R_{s1} and for $\Delta t > \frac{2\pi}{500}$ at R_{s2} . The higher RMSE obtained with smaller Δt stems from the larger number of interpolations required per simulation, which accumulates interpolation diffusion.”

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 will include 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. $\Delta t \sim 10^{-2}$ 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 will make 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 will therefore revise the sentence to correctly state that the SL method reduces the numerical diffusion observed with SUPG in these cases. 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 will make the precision in 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 will be fixed.

The authors might want to check their reference list more carefully, e.g. on line 545 "slope limiter for p -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 will correct/update the bibliography.