Response to “Review of ‘Consistent Point Data Assimilation in Firedrake and Icepack’ by Nixon-Hill et al.” by Doug Brinkerhoff

Reuben W. Nixon Hill, Daniel Shapero, Colin J. Cotter, David A. Ham

February 16, 2024

1 Summary

We thank the reviewer for his constructive comments. One high level conclusion that we can draw from both this and the other review is that we have been insufficiently clear in what the core message of this paper is. As we write in the response to the other reviewer, the core idea of this manuscript is that point evaluation can be fully integrated as a first class, differentiable operation in a symbolic finite element framework such as Firedrake, and that doing so makes it straightforward to assimilate point data by interpolation, as opposed to the (mathematically problematic) extrapolation methods that have frequently been applied. We will make this explanation much more explicit in the abstract and introduction.

Since the representation of point evaluation as a finite element operation in the UFL abstraction is a core contribution of this manuscript, much of what the reviewer took to be unnecessary mathematical detail is actually necessary. That said, the reviewer is entirely correct that there are a number of points at which our exposition could have been significantly improved, and we will make those changes in the revised manuscript as noted below.

2 Response to Specific Comments

P1 Capitalisation of Partial Differential Equations. This is has been fixed in the revised manuscript.

P2, Para 2 Which versus that. We have gone through the manuscript and corrected this here and in a few other places.

P2, Para 2 Differentiation meaning AD. The reviewer is correct and we have amended this accordingly.

Sec. 2 As suggested, we have removed much of the ”text book” material from chapter 2. We agree with the reviewer that it is still necessary to restate that at finite element field is a weighted sum of basis functions, because the typical GMD reader may not have the reviewer’s intimate acquantence with finite element methods.

Sec. 3 Here we disagree with the reviewer. It is certainly true that the evaluation operator is a linear operator between finite dimensional vector spaces with specified bases, and is hence expressible as a matrix. However, the strength of the finite element method, as exploited by Firedrake, FEniCS and others, is that a high level mathematical expression of the problem to be solved can be used to automatically generate the relevant matrices, and can be algorithmically differentiated to enable the solution of nonlinear systems and optimisation problems. To simply jump to the matrix begs the question in this regard. The linear algebra approach that the reviewer advocates is not simpler if one starts from the position of vector calculus applied to functions in finite element spaces, which is the basic formalism of the finite element method. The derivations in this section are therefore not about data structures, they are about raising the level of mathematical abstraction of point cloud interpolation to that of the rest of the field. Here we note the kind description by the other reviewer of this extension of the UFL formalism as ”remarkable”. We have rewritten part of the introduction to make this motivation for the new formalism explicit.

Now we turn to whether our maths is actually correct. Here we think that the comments of the reviewer are actually dual to the correct situation. Consider a domain comprised of a cloud of points (what we call a vertex only mesh):

$$\Omega_v = \{X_i\}_{i=0}^{N-1}, \tag{1}$$

If we restrict ourselves to real scalar-valued functions for brevity of exposition (nothing important changes in the complex, or vector- or tensor-valued case) then a function defined on a vertex-only mesh, has the form:

$$f : \Omega_v \to \mathbb{R}. \tag{2}$$

f associates with each vertex X_i a value. The space of all such functions, V, is clearly N-dimensional. The natural finite element basis for this is:

$$\psi_i(x) = \begin{cases} 1, & x = X_i \\ 0, & \text{otherwise} \end{cases} \quad 0 <= i < N \tag{3}$$

Which is exactly the P0DG basis. A Dirac delta, in contrast, is a functional or measure. It can't be a basis function for V because it doesn't lie in the space V. It is, however a member of V^* and the corresponding dual basis in the Ciarlet sense is:

$$\psi_i^*(g) = \delta_{X_i}(g) = g(X_i) \quad 0 <= i < N \tag{4}$$

In defence of his position that the basis functions need to be Dirac deltas, the reviewer writes: "the spatial integral of a finite valued basis function that is only defined at a single point in Rd (as in Eq. 7) would be zero". This would be true if the integral were over the spatial domain Ω and the integral measure were therefore $d(>0)$-dimensional. However, in equation 7 and elsewhere where we integrate over the point cloud, the domain of integration is Ω_v and the measure is therefore the finite sum of individual point measures making up the cloud.

Integration is a bounded linear functional, and integration of the functions defined at a single point must be linear in the single function value. It is therefore equal to the Dirac delta defined at that point. Once again, the Dirac delta is the functional rather than the function.

One possible cause of this confusion is that we have adopted the UFL convention of writing dx for the volume measure of whichever domain is being integrated over, which is a point measure on a vertex only mesh. We have now disambiguated this by writing dx_v to indicate the point measure when the domain of integration is Ω_v.

Sec. 4 Here we encounter the same disagreement as the previous section, so the response is essentially the same. The reviewer notes that Firedrake has a pre-existing "at" syntax, and claims that this is sufficient to abstract away the point evaluation of functions. This provides an opportune moment to explain why this is not so. The "at" method of a Function in Firedrake does indeed evaluate the function at the provided coordinates. However, it does not support the concept of a persistent set of point cloud locations nor of a set of values associated with them. This matters in the context of algorithmic differentiation and inverse problems, because in order to define the required Gateaux derivatives, we need to have a concept of known and unknown variables of a given type. In this case the type is "values associated with a particular collection of point locations". A large part of the point of these sections are the introduction of the required types (FunctionSpaces) and variables of those types (Functions). This representation of point cloud data in terms of a distinctive type is also useful for performance reasons in providing a location to cache point searches and parallel decomposition, but this is secondary to he main point here.

Equation 17 we have prepended the conventional phrase.

On P9, regarding pointwise evaluation of DG functions Users frequently run problems on regular meshes and then choose evaluation points at round number coordinates that coincide with element boundaries, so this case is nowhere near as uncommon as the reviewer might suspect. The challenge to special-casing the result of interpolation in this case, as the reviewer suggests, is that roundoff error makes it exceptionally difficult, if not impossible, to detect that the point lies on a cell boundary. The most that could be done is to apply the special case to a finite width band around the edge of the cell. This would be unlikely to please users.

Of course the strict answer is that point-evaluating a discontinuous finite element space is not well defined *at any point*. However this is an interpolatory crime that users frequently choose to commit, so the current behaviour of Firedrake is the compromise that best matches user expectations.

P9, last paragraph Yes, interpolation is a linear operator. This doesn't remove the need to represent it symbolically in a way that Dolfin-adjoint/pyadjoint can reason about.

Eq. 19 i is an index into the set of evaluation points $\{X_i\}$. We have made this explicit.

Just after Eq. 20 The reviewer is correct that "guess" is imprecise. We have reworded in accordance with the suggestion.

Note on nomenclature This has been removed as suggested.

Eq. 21 There are other ways out of the dilemma including, as the reviewer points out, using Gaussian processes. This would involve estimating the Gaussian random field f that produced the observations, and then creating a misfit term that penalises the distance to the mean of f, using the covariance operator as a metric in the norm. Nonetheless, using Gaussian processes is not yet standard practice at least in the literature on glaciological inverse problems. It is far more common to interpolate the observational data to an intermediate field in a way that does not account for the sparsity or density of observational data. Improving on this common practice is the main point of this paper. Our approach does allow for the possibility of different weights or variances for each observation point (or indeed covariances) by using a weighted norm in equation 21 and in fact we do this in the Larsen C demo at the end of the paper.

Last line of Sec. 6 We thank the reviewer for referring us to Bouziani and Ham (2023), which one of us coauthored. That paper does not employ point evaluation and so is not directly relevant to the point being made in the last line of section 6. The line in question specifically says that automated code generation systems with adjoint capability (or differentiable programming models for the finite element method, if one chooses to adopt the terminology in Bouziani & Ham) have not previously handled point data. The reviewer objects that one could apply a generic AD tool to a finite element model. This is true, but it's a long way from an automated process and the arguments about efficiency and robustness of generic AD systems that we raised in Farrell et al. (2013) still apply. A more relevant alternative would be to hand-code the adjoint to the point interpolation and to implement a sui generis composition of that code with Firedrake's adjoint, as the authors did using Firedrake in Roberts et al. (2022). However, even that approach lacks the seamlessness, automation and expressiveness of directly incorporating a differentiable point evaluation operator in Firedrake itself.

Between Eqs. 24 and 25 As noted above, the definition of the basis functions is correct.

P13 We have fixed the capitalisation of L-curve.

Sec. 7.1.1 We have rewritten this section to avoid reference to posterior consistency. The point can be made more clearly without reference to Bayesian inversion at all. The point is that if the observations are first interpolated to the computational grid and then incorporated as a misfit term penalising deviation between the interpolated data and the solution of the inverse problem, then the error will saturate as the number of observations is increased, whilst if the interpolation operator is used and the misfit term is properly defined as a sum over observation points, the error can continue to decrease with observations below the saturation point.

Eq. 32 This was a mistake, we have corrected it in the revision.

P18, paragraph 3 We have elaborated on this more in the text. In this example, there are only 3 parameters to infer, so much of the machinery of Bayesian inference is not necessary.

P18, last paragraph We have rewritten this to remove mention of posterior consistency as 7.1.1.

Sec. 7.3.1 We disagree. Although this is not a glaciology paper as such, we feel that it is important to give some justification for why we used this simplified equation set. If we had submitted this paper to The Cryosphere then we might assume this knowledge on the part of readers.

Sec. 7.3.2 Corrected in the revised version.

P21, para 3 We have added a statement to the text to this effect. Ideally, we would have conducted this experiment using only the raw chip matches or displacements that are obtained from repeat-image feature tracking, which are not on a regular grid, rather than the regular grid of velocity values that are interpolated after the fact. Instead we made do with what is readily publicly available, i.e. a gridded velocity product.

Fig. 8 This is the units for the regularisation parameter (meters).

P23 L1 This would involve estimating the Gaussian random field f that produced the observations, and then creating a misfit term that penalises the distance to the mean of f, using the covariance operator as a metric in the norm. If the observation data are correlated, then this can be modelled using a covariance matrix in the misfit in equation 21. This would be a worthwhile exercise for a glaciology / remote sensing paper. However, the purpose of this exercise to conduct an experiment that is not possible by the conventional approach of first interpolating the observational data to the finite element mesh and then using the 2-norm misfit between the computed and interpolated velocity fields.

P23 Para 2 We have clarified the text. Briefly, if you are willing to assume that (1) the model is correct, and (2) that the error estimates are correct in a relative but not an absolute sense, then it is possible using cross-validation to estimate the absolute scale that the error estimates should have been on.

P23 Para 3 and 4 We have cut most of this text and revised what remains. We do not claim to have taken the most principled approach possible and we agree that a full Bayesian treatment would be superior. Instead, we argue that (1) accounting for the point-like nature of the observations, either directly in the formulation of the inference problem or by using Gaussian process regression, is a substantial improvement over interpolating and then fitting, and (2) Firedrake is now provides seamless forward and adjoint support for the point-like nature of the observations, which is relatively uncommon among finite element modelling packages. We do not need to take the most principled approach imaginable in order to improve upon existing practice in the glaciological community or in other disciplines. Point data assimilation enables substantial improvements to current practice.

Sec. 8 We think that the reviewer's core question about this section reflects the fact that we were insufficiently clear that the point data abstraction and implementation are core to the paper. That said, we have revised this section to make it clearer that it is about further work which is facilitated by a point data abstraction. The section has also been updated to account for development which has occurred in the extended period that the paper has been in review.

References

Bouziani, N. and Ham, D. A.: Physics-driven machine learning models coupling PyTorch and Firedrake, in: ICLR 2023 Workshop on Physics for Machine Learning, https://doi.org/https://doi.org/10.48550/arXiv.2303.06871, 2023.

Farrell, P. E., Ham, D. A., Funke, S. W., and Rognes, M. E.: Automated Derivation of the Adjoint of High-Level Transient Finite Element Programs, SIAM Journal on Scientific Computing, 35, C369–C393, https://doi.org/10.1137/120873558, publisher: Society for Industrial and Applied Mathematics, 2013.

Roberts, K. J., Olender, A., Franceschini, L., Kirby, R. C., Gioria, R. S., and Carmo, B. S.: spyro: a Firedrake-based wave propagation and full-waveform-inversion finite-element solver, Geoscientific Model Development, 15, 8639–8667, https://doi.org/10.5194/gmd-15-8639-2022, 2022.