

Interactive comment on “The Land Variational Ensemble Data Assimilation fRamework: LaVEnDAR” by Ewan Pinnington et al.

Anonymous Referee #2

Received and published: 14 June 2019

In this paper Pinnington et al introduce the land modeling community to 4D ensemble variational data assimilation (4DEnVar) and demonstrate its application to a land model, JULES, at a site-scale. This system is applied in two examples, one using simulated data with known parameters, the other using field data, to demonstrate the ability of 4DEnVar to constrain model parameters. The system appears to be updating initial conditions as well, though these were not discussed. The Python code base to support these two examples is given the name LaVEnDAR, and while the approach is touted as being ‘general’, it is not entirely clear whether the system has been applied to other sites, models, or data constraints, nor is it clear what would be required to be able to do so. That said, it is fully worth acknowledging that the work presented in this paper is definitely sufficient to constitute a ‘first application’ of the LaVEnDAR system and

C1

that the system appears to perform well. In the detailed comments below I will raise a number of concerns about the current capabilities of this system, but I want to be clear at the outset that I do not consider these to be fatal flaws for this initial paper, but rather important caveats that need to be stated more explicitly in the paper and then improved upon in subsequent applications.

Page 1, Lines 15-17: Both your land surface model and atmospheric models are completely deterministic. The chaotic behavior of the atmosphere is indicative of its high sensitivity to initial conditions, not stochasticity. The important difference I think you’re trying to get at, which I’ll agree has very important implications for prediction and data assimilation, is that JULES is a stable model that will converge to a steady state.

P1, L19: (i) “problem_of_parameter estimation” (ii) I don’t agree that it is safe to assume that parameter estimation is the only issue here, or even that parameter uncertainty is the dominant uncertainty in land model prediction. I don’t think this has really been shown conclusively, as all existing uncertainty analyses I’m aware of either ignore or confound multiple key uncertainties. I agree calibration is definitely an important problem, but don’t oversell/overstate your position.

P2, L9: “parameters can change over time” – this point is debatable. I’d say that it’s probably more appropriate to say that allowing model parameters to change over time (or space) is a mechanism that can be used to account for model structural inadequacy (processes or covariates that are missing from the model). Of course that inadequacy/incompleteness is an inherent feature all models, so to some degree parameter variability (typically modeled using random effects) is frequently a source of uncertainty that needs to be considered.

P2, L12: If the focus is on efficient approaches to model calibration, I’d recommend mentioning emulator methods as well (e.g. Fer et al 2018 Biogeosciences)

P2, L14: I don’t think “non-Gaussianity” is a word. Maybe “non-Gaussian error” instead?

C2

P2, L21: I'm surprised the paper is adopting the position that parameters should be static in time after arguing just 12 lines ago that parameters change over time.

P3 L31: GPP is not an observation, it is predicted from a simple model based on NEE and environmental covariates, and those simple models are known to have errors and systematic biases. Treating GPP like it is data means you are calibrating your model to another model, which should be treated with extreme caution.

P4 L14-15: I find this notation to be unnecessarily confusing. Specifically, why use i as a subscript instead of t when i is being used to indicate time? It'd be much simpler to just use t and $t-1$. Also, do we really need a subscript on f ? Is the model itself changing with time?

P4, L25: won't this structure change the time invariance of p once you account for process error (unless you define the variance as zero, but that'll probably mess up the inversion of the error covariance matrix)

P6 L6: This bit is really in the weeds and could benefit from a bit more detail/explanation.

P7 L17: Here you say the adjoint is still present, but this is the first mention of an adjoint in the Methods. Needs further explanation.

Figure 1: I'm not sure this figure is useful. I'd either drop or combine it with Figure 2

P9, L6: Why these seven parameters? Were there any sort of uncertainty analyses performed that attributed model uncertainty to these parameters specifically?

P9, L11: Why was this variance chosen? Does this represent a typical or realistic level of parameter uncertainty? My experience has been that the magnitude of parameter variance can differ enormously from parameter to parameter because of the wildly differing amounts of trait data available to constrain different parameters.

P9, L12: This is definitely an unrealistically low amount of noise on any sort of land

C3

observations, especially in light of the fact that process error is not included. It might be useful to develop some additional analyses that explore larger, more realistic observation errors.

P10, L2-4: Choice of variances here (initial conditions and observation errors) are similarly not given any sort of justification and strike me as much too tight.

P19, L8: But this raises the question about why this parameter was selected for inclusion in the calibration, out of the 90 PFT-level parameters in JULES http://jules-lsm.github.io/vn4.9/namelists/pft_params.nml.html, if model outputs are not sensitive to it.

P20, L4-14: I find it odd that this paragraph discusses the ability of 4DEnVar to accurately retrieve parameters as if it were a bad thing. At the heart of the issue is the (unstated) problem of filter divergence, where the model ensemble becomes sufficiently confident in itself that it ignores (diverges from) the observations. The authors' concern here suggests a misrepresentation of the uncertainties that control the ensemble spread. Specifically, of the five uncertainties that control the spread of the ensemble (initial conditions, external drivers, parameter uncertainty, parameter variability [i.e. random effects], and process error; see Dietze 2017 Ecol Appl), the current analysis is only considering two (IC and parameter uncertainty). Because, as discussed at P1 L15, the model is stable the IC uncertainty will decline exponentially toward zero with time. Similarly, parameter uncertainty will decline asymptotically toward zero with more data (and since the data are timeseries, that implies that this uncertainty also declines with time). So it is unsurprising that the ensemble is converging toward zero variance, as that's exactly what we know it should do from first principles. By contrast, the three uncertainties not included in the current analysis (drivers, random effects, process error) all systematically increase the ensemble variance with time. There are also ~80 other PFT-level parameters in JULES whose (prior) uncertainty isn't being propagated. Rather than suggesting the use of methods to inflate the ensemble variance, I'd argue that the authors would be much better served by including the missing uncertainties

C4

that do this naturally. As discussed in my overall summary, I'm not asking the authors to redo their current analysis, but I am asking that they revise their Discussion to acknowledge the future steps that need to be taken to incorporate these missing uncertainties.

P20, L8: If parameters trading off is an issue, I'd recommend reporting the posterior covariances (or correlations) as a supplement. You fundamentally can't see equifinality in the parameter means or posterior marginal distributions, as it is a property of the JOINT posterior distribution.

P20, L13: While I previously suggested that you drop ensemble inflation methods altogether from the Discussion, if you do retain this I'll note that this statement is not explained sufficiently to the reader.

P21, L3: Either explain what ensemble localization is, or drop.

P21, L17: As noted P1, L19 I don't think this issue is settled. It's worth noting that the paper cited as justification likewise only considers a subset of the uncertainties mentioned in my comment on P20, L4-14

P21, L22: (i) as noted in P20, L4-14 comment, don't equate process error with "stochastic noise" or inflation. (ii) It is also worth noting that data assimilation frameworks are usually applied iteratively, but the current proof-of-concept application of LaVENDAR is a completely 'offline' problem that's never applied to more than a year's worth of data. I think this requires more discussion and acknowledgement of the current system's limitations. There are a number of additional modules that would need to be added to LAVENDAR to support this, as well as features that need to be present in the model itself. Specifically, while LAVENDAR claims to be able to work on any existing model based just on it's outputs, iteration would require the ability to save the full state of the model and then restart the model from updated initial conditions. I'd recommend expanding this discussion. Similarly, by only applying the 4DEnVar to one year of data (neither assimilating a second year nor validating against a second year)

C5

the authors were able to skip over the problem that using fixed parameters will most likely leave the model unable to capture interannual variability. The single-year application may thus be leaving readers with an overly-optimistic view of how well the system is performing. I'd recommend discussing this explicitly or (even better) demonstrating it with additional validation years.

Interactive comment on Geosci. Model Dev. Discuss., <https://doi.org/10.5194/gmd-2019-60>, 2019.

C6