Geosci. Model Dev. Discuss.,
https://doi.org/10.5194/gmd-2019-185-AC2, 2019

Geoscientific Model Development Discussions

Interactive comment on "Geostatistical inverse modeling with very large datasets: an example from the OCO-2 satellite" *by* Scot M. Miller et al.

Scot M. Miller et al.

scot.m.miller@gmail.com

Received and published: 26 December 2019

We would like to thank Peter Rayner for his feedback and constructive ideas on the manuscript. These ideas have been very helpful for improving the overall quality of the manuscript and strength of the analysis. We have listed Dr. Rayner's comments in bold typeface below and discuss the associated changes we have made to the manuscript.

- **My first question concerns the code. This is likely to be a significant part of the contribution of the paper, at least for those who use MATLAB. Yet most people are not solving exactly the same problem as the authors. So the question arises how to make such code more generally useful, and from the journal's viewpoint, how to have its utility reviewed. I wonder if a short**

appendix to the paper or a document attached to the code describing any particular problems the authors had to overcome to implement the method and the approaches they took might be more generally useful than learning this from the code directly.

We think this is a great suggestion and have added additional text to the user guide for the associated model code on Github and Zenodo. Much of this text is specific to the practicalities of coding the concepts described in the paper, so we have included this text with the code instead of as an appendix to the manuscript. Below, we have pasted the additional text that has been added to the user guide.

Additional text that will be added to the software user guide:

The computational approaches implemented in this code are designed for large inverse problems, but it is nevertheless important to keep computational considerations in mind when adapting the code for a specific inverse problem. We discuss several of these considerations below:

1. The number of iterations required by the iterative solver to estimate the fluxes can be an important limiting factor when using certain types of adjoint atmospheric models but may not be a limiting factor when using other types of atmospheric models. For trajectory models like the Stochastic Time-Inverted Lagrangian Transport (STILT) model, $\mathbf{H}$ is formulated explicitly and can be read in directly. In that case, the computing resources required to run numerous STILT trajectories, not the number of iterations required by the solver, is likely to be the computational bottleneck. By contrast, the number of iterations required to converge on a solution is likely to be the bottleneck for gridded chemical transport models like GEOS-Chem or TM5. These models do not produce an explicit $\mathbf{H}$ and $\mathbf{H}^T$ matrices, and one must instead run the forward and adjoint models once per iteration of the solver. These calculations often become time-intensive when numerous iterations are required to converge on a flux estimate. Furthermore, some adjoint

models (i.e., GEOS-Chem) cannot be run in parallel for greenhouse gas applications, though we expect that these capabilities will change in the future with the development of an adjoint for models like GEOS-Chem-High Performance (GC-HP).

2. The matrices $\mathbf{D}$ and $\mathbf{E}$ (Eqs. 9-10 in the manuscript) are usually straightforward to store in memory and/or invert given the dimensions of most atmospheric inverse models to date. However we anticipate that this will change in the future as atmospheric models like GEOS-Chem have better parallel computing capabilities and can be run at higher spatial resolution. In those cases, it may be important to structure $\mathbf{D}$ and $\mathbf{E}$ as hierarchical matrices or circulant matrices to avoid problems with storing these matrices in memory or inverting these matrices.
3. The choice of covariance function can have a large impact on the wall clock time and memory required for matrix calculations using $\mathbf{D}$ and $\mathbf{E}$. An exponential covariance model is very common in existing GIM studies in hydrology and atmospheric science. For large inverse problems, this choice may not be practical; an exponential model will never decay to zero. As a result, $\mathbf{D}$ and $\mathbf{E}$ will never be sparse matrices. By contrast, other covariance models, like a spherical model, do decay to zero, and $\mathbf{D}$ and $\mathbf{E}$ can be formulated as memory-saving sparse matrices.
4. The code here can be re-written for other languages if a different language is more convenient than Matlab. We recommend that users exercise caution if doing so because different commonly-used languages can exhibit very different performance. For example, we found that R is far slower than Matlab at linear algebra and often requires more memory than Matlab for the same matrix inversion.

- **My other question concerns section 5.2. The general finding here is that the reduced rank approximation will overestimate posterior uncertainty since**

it reduces the size of the update made via the ShermanMorrisonWoodbury matrix lemma. I agree with that but doesn't it also reduce the generalised variance of the prior by, for example, limiting the number of eigen-values in the decomposition? If that is correct do we have any sense of how this balance plays out?

We have clarified this point in the text. In this setup, the prior is taken to be a positive definite, full rank matrix and is not affected at all by the approximation; however, the posterior covariance is written as an update of the prior covariance matrix involving selected eigenpairs. Since we are subtracting a positive semidefinite update, this ensures that the variance is reduced. An intuitive way of understanding is that by observing data, the variance (i.e., the uncertainty) is reduced since we know more about the parameters of interest.

Interactive comment on Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2019-185, 2019.