Geosci. Model Dev. Discuss.,
doi:10.5194/gmd-2016-208-RC3, 2016

Geoscientific Model Development Discussions
Open Access
EGU

Interactive comment on "A mask-state algorithm to accelerate volcanic ash data assimilation" *by* Guangliang Fu et al.

Anonymous Referee #3

Received and published: 28 December 2016

The authors presented a "mask-state algorithm[2] which reduces the dimension of full ensemble state matrix into a relatively smaller one and consequently reduce the computational cost in the analysis step of Ensemble Kalman Filter (EnKF) for data assimilation. Computational cost for the analysis step is studied in detail. Numerical tests show that computational times for analysis step is reduced to less than 1/3 of the original time for analysis step. The overall computational time is reduced to a level that is smaller than given time window. The idea of reducing work by only working with the non-zero rows identified by the [3]mas-state[2] is rather straightforward but the notion of exploiting sparsity and parallelism of the matrix is necessary. I am appreciative of the hard work in the complicated task of managing the irregular sparsity. Effectively this is a good problem specific reordering scheme for the matrix. This paper makes the case for the need of sparsity aware processing by comparing the performance (computation

Printer-friendly version

Discussion paper

and memory) to a full matric implementation. The overall contribution while not exciting or deep likely has an impact on data assimilation(DA) and given the standard practice in the DA community it potentially represents an advance. A true test of the quality of this work would be comparison to standard sparse matrix methods (like Compressed Sparse Row or Column or more many more advanced variants tune to multi-core and multi-processor architectures) that have been much studied (see for e.g.[1-6]) and are literally graduate textbook material.

More numerically sophisticated areas like the modeling community will find this standard but it is likely to have an impact on the DA community so I recommend publication after the authors carefully make the case for the Mask State scheme relative to more standard sparse matrix methods not just comparison to full storage dense matrices. If the authors were to attack or even attempt to tackle the harder problem of programming for parallelism with changing sparsity this would be also be a much stronger and impactful paper.

Other limitations of the proposed algorithm:

1) The proposed method is based on the fact that volcanic ash only occupies portion of the whole domain and hence there will be many zero rows in the ensemble state matrix. So such a method can only be implemented for flows that have such feature. It seems that few applications of Ensemble Kalman Filter (EnKF) data assimilation method have this feature.

2) According to [3]the analysis step takes 72% of total runtime[2], it means the analysis step is the bottle neck. However, such case might not be general for all ash forecasts. As the computational cost for initialization and forecast greatly depends on the [3]forecast model[2] that is used. If a more expensive ash forecasting model is used, then, I would guess, the bottle neck would be ash forecasting.3) It is not clear to me that a good parallel linear algebra library like PeTSC which allows users to specify their orderings and comes with machine optimized parallel matrix-matrix multiply operations

Printer-friendly version

Discussion paper

will not outperform the versions coded up here.

Modification suggestions and questions:

1) The paper focusses on "Ensemble Kalman Filter (EnKF) data assimilation method[2] and the new algorithm proposed in this paper is specific for "Ensemble Kalman Filter (EnKF) data assimilation method[2]. It would be more precise to use [3]data assimilation based on Ensemble Kalman Filter (EnKF)[2] instead of just [3]data assimilation[2] in the title.

2) What is the subscript [3]q[2] in Eq. (3), should it be [3]N[2]?

3) It would be interest to see the additional cost for reducing the size of the ensemble state matrix.

4) The "non-zero rows[2] is around 0.393 of total rows. According to cost analysis, the cost is O(nNˆ2). So theoretically speaking, the minimum time would be 0.393*3.14h $\sim$ 1.234h. Numerical test shows that the time goes down to 0.88h. It is interesting to know what are the other contributions to this time decrease. My guess is memory access cost also goes down when the size of matrix reduced.

5) Again, regarding questions in 4) 6) The most expensive sub-step in the analysis step is actually a matrix multiplication. It should be trivial to parallelize it by yourself or utilize libraries like scaLAPACK. This would be a more general and more powerful way to reduce time on analysis step.

7) Mathematics symbols need to be a little bit more clear, for example, in the paper, y is used as both observational space and a vector in the space?

[1] Bank, Randolph E.; Douglas, Craig C. (1993), "Sparse Matrix Multiplication Package (SMMP)" <http://www.cs.yale.edu/homes/douglas-craig/Preprints/pub34.pdf>(PDF), Advances in Computational Mathematics, 1

[2] Buluç, Aydžn; Fineman, Jeremy T.; Frigo, Matteo; Gilbert, John R.; Leis-

erson, Charles E. <https://en.wikipedia.org/wiki/Charles_E._Leiserson> (2009). Parallel sparse matrix-vector and matrix-transpose-vector multiplication using compressed sparse blocks <http://gauss.cs.ucsb.edu/∼aydin/csb2009.pdf> (PDF). ACM Symp. on Parallelism in Algorithms and Architectures. CiteSeerX <https://en.wikipedia.org/wiki/CiteSeerX> 10.1.1.211.5256 <https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.211.5256>

[3] M. Kreutzer, G. Hager, G. Wellein, H. Fehske, and A. R. Bishop: A unified sparse matrix data format for modern processors with wide SIMD units, 2014 [4] Barrett, et al., [3]Templates for the solution of linear systems: Building Blocks for Iterative Methods, 2nd Edition[2], SIAM, 1994 (book online)

[5] Sparse BLAS standard: http://www.netlib.org/blas/blast-forum

[6] BeBOP: http://bebop.cs.berkeley.edu/

Interactive comment on Geosci. Model Dev. Discuss., doi:10.5194/gmd-2016-208, 2016.