University of Colorado at Denver and Health Sciences Center

# Efficient Implementation of the Ensemble Kalman Filter

Jan Mandel

May 2006 UCDHSC/CCM Report No. 231

CENTER FOR COMPUTATIONAL MATHEMATICS REPORTS

# EFFICIENT IMPLEMENTATION OF THE ENSEMBLE KALMAN FILTER

JAN MANDEL*

**Abstract.** We present several methods for the efficient implementation of the Ensemble Kalman Filter (EnKF) of Evensen. It is shown that the EnKF can be implemented without access to the observation matrix, and only an observation function is needed; this greatly simplifies software design. New implementations of the EnKF formulas are proposed, with linear computational complexity in the number of data points. These implementations are possible when the data covariance matrix is easy to decompose, such as a diagonal or a banded matrix, or given in a factored form as sample covariance. Unlike previous methods, our method for the former case uses Choleski decomposition on a small matrix from the Sherman-Morrison-Woodbury formula instead of SVD on a large matrix, and our method in the latter case does not impose any constraints on data randomization. One version of the EnKF formulas was implemented in a distributed parallel environment, using SCALAPACK and MPI.

**1. Introduction.** The Ensemble Kalman Filter (EnKF) is a Monte-Carlo implementation of the Bayesian update problem: Given a probability distribution of the modeled system (the prior, called often the forecast in geosciences) and data likelihood, the Bayes theorem is used to to obtain the probability distribution with the data likelihood taken into account (the posterior or the analysis). The Bayesian update is combined with advancing the model in time, with the data incorporated from time to time. The original Kalman Filter [14] relies on the assumption that the probability distributions are Gaussian, and provided algebraic formulas for the change of the mean and covariance by the Bayesian update, and a formula for advancing the covariance matrix in time provided the system is linear. However, this is not possible computationally for high dimensional systems. For this reasons, EnKFs were developed [7, 12]. EnKFs represent the distribution of the system state using a random sample, called an ensemble, and replace the covariance matrix by the sample covariance of the ensemble. One advantage of EnKFs is that advancing the probability distribution in time is achieved by simply advancing each member of the ensemble. EnKFs, however, still rely on the Gaussian assumption, though they are of course used in practice for nonlinear problems, where the Gaussian assumption is not satisfied. Related filters attempting to relax the Gaussian assumption in EnKF include [3, 4, 15, 17].

This paper is focused on the Bayesian update step for the EnKF version from [6, 8]. This filter involves randomization of data. For filters without randomization of data, see [2, 9, 16].

The paper is organized as follows. In Sec. 2, we state the Bayesian update formulas in EnKF. In Section 3, we show how to evaluate formulas without an observation matrix, involving observation functions. Sec. 4 contains the discussion of several implementations of the EnKF and their computational complexity. Finally, Sec. 5 briefly reports on the experience from a distributed parallel implementation.

**2. Notation and the EnKF Formulas.** We review the EnKF formulas following [6, 8], with only one minor difference. The forecast ensemble consists of $N$ members, which are state vectors of dimension $n$. The ensemble can be written as

*Center for Computational Mathematics, University of Colorado at Denver and Health Sciences Center, Denver, CO 80217-3364

the $N$ by $n$ matrix

$$X^f = [x_1, \ldots, x_N] = [x_i]$$

are the forecast ensemble states. The ensemble mean is

$$E(X) = \frac{1}{N} \sum_{k=1}^{N} x_k.$$

and the ensemble covariance matrix is

$$C = \frac{AA^T}{N-1}$$

where

$$A = X - E(X) = X - \frac{1}{N} (Xe_{N\times 1}) e_{1\times N},$$

and $e$ denotes the matrix of all ones of the indicated size.

The data is given as a measurement vector $d$ size $m$ and error covariance matrix $R$ size $m$ by $m$. Then the measurement matrix $D$ size $m$ by $N$ is defined by

$$D = [d_1, d_2, ..., d_N], \quad d_j = d + v_j, \quad v_j \sim N(0, R)$$

where $v_j$ are independent random perturbations.

The analysis ensemble is then given by

$$X^a = X + CH^T \left(HCH^T + R\right)^{-1} (D - HX), \tag{2.1}$$

cf., [8, eq. (20)].

The difference between (2.1) and [8, eq. (20)] is that we use the covariance matrix $R$ of the measurement error rather than the sample covariance $DD^T/(N-1)$ of the randomized data. Because $R$ is always positive definite, there is no difficulty with the inverse in (2.1), unlike [8, eq. (20)].

The analysis formula (2.1) can be rewritten similarly as in [8, eq. (54)] as

$$X^a = X + \frac{1}{N-1} A (HA)^T P^{-1} (D - HX) \tag{2.2}$$

where

$$HA = HX - \frac{1}{N} ((HX) e_{N\times 1}) e_{1\times N}, \tag{2.3}$$

$$P = \frac{1}{N-1} HA (HA)^T + R. \tag{2.4}$$

**3. Observation Matrix-Free Method.** Clearly, the matrix $H$ in (2.2) – (2.4) is needed only in the matrix vector product $HX$, which needs to be computed only once. However, it is very inconvenient to operate with the matrix $H$ explicitly for several reasons:

1. An observation function $h(x)$ that creates synthetic data from the state can be quite complicated. Even when the observation function is affine, thus of the form

   $$h(x) = Hx + f, \tag{3.1}$$

   creating the matrix $H$ and the vector $f$ is an additional effort, which is typically much harder than programming the observation function itself.

2. Computing the matrix-vector product $HX$ takes computational resources.
3. The matrix $H$ is typically sparse and it can be very large. E.g., in the quite common case when every measurement coincides with the value of a state variable, the matrix will have just one one in every row and the rest of its entries are zeros. If the measurement is an image, the number of measurements can be very large, but each pixel in the image is interpolated from just a few entries of the state vector. Assimilation of image data into a geophysical model may easily require $m \approx 10^6$ and $n \approx 10^6$, which makes manipulation of the matrix $H$ stored as full impossible on current computers. So, the matrix $H$ must be stored as sparse, which is an additional complication.

Fortunately, creating the matrix $H$ explicitly is not needed. Assume that we only have access to the evaluation of $h(x)$ in (3.1) but not to the values of $H$ or $f$. To compute the data residual, note that

$$d - Hx = d + f - (Hx + f) = \widetilde{d} - h(x),$$

where

$$\widetilde{d} = d + f$$

is the data actually given. To compute $HA$, note that

$$\begin{aligned}
[HA]_i &= Hx_i - H\frac{1}{n}\sum_{j=1}^{n} x_j \\
&= (Hx_i + f) - \frac{1}{N}\sum_{j=1}^{N} (Hx_j + f) \\
&= h(x_i) - \frac{1}{N}\sum_{j=1}^{N} h(x_j).
\end{aligned}$$

Consequently, the ensemble update can be computed by evaluating the observation function $h$ on each ensemble member once.

**4. Computational Complexity.** All operations in (2.2) – (2.4) are evaluations of the observation function $h(x)$, cf., Sec. 3, and matrix-matrix and matrix-vector operations, which can be implemented efficiently by calls to the LAPACK, BLAS, and SCALAPACK libraries [1, 5]. These libraries contain routines for operations including Choleski decomposition of a symmetric positive definite matrix (CHOL) $LL^T = A$, and matrix multiply (GEMM), $A = \alpha A + \beta B^{(T)} C^{(T)}$, where $(T)$ is either transpose or nothing, for rank update. Recall that $m$ is the number of data points, $n$ is the number degrees of freedom, $N$ is the ensemble size, so $X$ is $n$ by $N$, $HX$ and $HA$ are $m$ by $N$, $R$ is $m$ by $m$. Assume that evaluation of $h(x)$ costs $O(m)$, and that multiplication of matrices of size $n_1$ by $n_2$ and $n_2$ by $n_3$ costs $O(n_1 n_2 n_3)$.

**4.1. Reference Implementation.** Straightforward implementation of the formulas (2.2) – (2.4) leads to the following algorithm:

| computation | operation | size | cost |
|---|---|---|---|
| $HX$ | $N$ times $h(x)$ | $m \times N$ | $O(mN)$ |
| $z = (HX)\, e_{N\times 1}$ | matrix multiply | $m \times N \times 1$ | $O(mN)$ |
| $HA = HX - \frac{1}{N} z e_{1\times N}$ | matrix multiply | $m \times 1 \times N$ | $O(mN)$ |
| $Y = D - HX$ | matrix add | $m \times N$ | $O(mN)$ |
| $P = R + \frac{1}{N-1} HA\,(HA)^T$ | matrix multiply | $m \times N \times m$ | $O(m^2N)$ |
| $LL^T = P$ | Choleski | $m$ | $O(m^3)$ |
| $M = P^{-1}Y$ | solution | $m \times m \times N$ | $O(m^2N)$ |
| $Z = (HA)^T M$ | matrix multiply | $N \times m \times N$ | $O(mN^2)$ |
| $X^a = X + \frac{1}{N-1} AZ$ | matrix multiply | $n \times N \times N$ | $O(nN^2)$ |

(4.1)

The total computational complexity of the algorithm (4.1) is

$$O\left(m^3 + m^2N + mN^2 + nN^2\right).$$

So, this method is suitable for a large number of degrees of freedom $n$, but not for a large number of observations $m$.

**4.2. Large Number of Data Points.** In practice, the number of data points $m$ is often large, while the data error covariance matrix $R$ is often diagonal (when the data errors are uncorrelated), nearly diagonal, or easy to decompose. This happens, e.g., in the assimilation of images, or in regularized EnKF where the gradient of fields in the state is assimilated as an additional artificial observation [13]. In this case, the following algorithm, which has only linear complexity in $m$, provides a significant advantage. Assume that multiplication by $R^{-1}$ is dominated by other costs. Using the Sherman-Morrison-Woodbury formula [11]

$$(R + UV^T)^{-1} = R^{-1} - R^{-1}U(I + V^T R^{-1} U)^{-1} V^T R^{-1},$$

with

$$U = \frac{1}{N-1} HA, \quad V = HA,$$

we have

$$\begin{aligned} P^{-1} &= \left(R + \frac{1}{N-1} HA\,(HA)^T\right)^{-1} \\ &= R^{-1}\left[I - \frac{1}{N-1}(HA)\left(I + (HA)^T R^{-1} \frac{1}{N-1}(HA)\right)^{-1} (HA)^T R^{-1}\right]. \end{aligned}$$

The computation (4.1) then becomes

$$\left.\begin{array}{llll}
\text{computation} & \text{operation} & \text{size} & \text{cost} \\
HX & N \text{ times } h(x) & m \times N & O(mN) \\
z = (HX)\, e_{N\times 1} & \text{matrix multiply} & m \times N \times 1 & O(mN) \\
HA = HX - \frac{1}{N} z e_{1\times N} & \text{matrix multiply} & m \times 1 \times N & O(mN) \\
Y = D - HX & \text{matrix add} & m \times N & O(mN) \\
Q = I + (HA)^T R^{-1} \frac{1}{N-1} (HA) & \text{matrix multiply} & N \times m \times M & O(mN^2) \\
LL^T = Q & \text{Choleski} & N & O(N^3) \\
Z = (HA)^T R^{-1} Y & \text{matrix multiply} & N \times m \times N & O(mN^2) \\
W = Q^{-1} Z & \text{solution} & N \times N & O\left(N^3\right) \\
M = R^{-1}\left[I - \frac{1}{N-1}(HA)\, W\right] & \text{matrix multiply} & m \times N \times N & O(mN^2) \\
Z = (HA)^T M & \text{matrix multiply} & N \times m \times N & O(mN^2) \\
X^a = X + \frac{1}{N-1} AZ & \text{matrix multiply} & n \times N \times N & O(nN^2)
\end{array}\right\} \tag{4.2}$$

This gives overall complexity $O\left(N^3 + mN^2 + nN^2\right)$, which is suitable for large $n$ and large $m$.

**4.3. Square Root Alternative.** Decompose first $R = SS^T$, e.g., by Choleski decomposition, and multiplication by $S^{-1}$ is cheap to compute. Let $\widetilde{U} = S^{-1}U$, $\widetilde{V} = S^{-1}V$. Then the Sherman-Morrison-Woodbury formula becomes

$$(SS^T + UV^T)^{-1} = S^{-T}\left[I - \widetilde{U}\left(I + \widetilde{V}^T\widetilde{U}\right)^{-1}\widetilde{V}\right] S^{-1},$$

which gives, again with $R = SS^T$, $B = S^{-1}HA$

$$P^{-1} = S^{-T}\left[I - \frac{1}{N-1} B\left(I + \frac{1}{N-1} B^T B\right)^{-1} B^T\right] S^{-1}$$

and one proceeds just as in Sec. 4.2. The asymptotic complexity is same as in Sec. 4.2, but the formulas involve symmetric products of matrices, which is numerically more stable and allows to save memory by storing just one triangle.

**4.4. SVD Method for Full Data Error Covariance.** Assume again that $R = SS^T$ and multiplication by $S^{-1}$ is cheap to compute. Write

$$P = \frac{1}{N-1} HA\,(HA)^T + R = S\left[\frac{1}{N-1} S^{-1} HA\,(HA)^T\, S^{-T} + I\right] S^T$$

and use the singular value decomposition (SVD) [10]

$$S^{-1}HA = U\Sigma V^T,$$

where $U$ and $V$ are orthogonal square matrices, and $\Sigma = \text{diag}_{m\times N}(\sigma_1, ..., \sigma_k)$ is the diagonal matrix size $m$ by $N$ with the singular values $\sigma_1, ..., \sigma_k$, $k = \min\{m, N\}$, on the diagonal; then, using the orthogonality relations $V^TV = I$, $UU^T = I$, we have

$$\frac{1}{N-1} S^{-1} HA\,(HA)^T\, S^{-T} + I = \frac{1}{N-1} U\Sigma U^T + I = U\left(\frac{\Sigma^2}{N-1} + I\right) U^T$$

thus

$$P^{-1} = R^{-1/2} U^T \operatorname{diag}\left( \frac{1}{\frac{\sigma_i^2}{N+1} + 1} \right) U R^{-1/2}$$

The complexity is again $O(mN^2)$, assuming that $m > N$ and that computation of SVD of matrix size $m$ by $N$ costs $O\left(mN \min(m, N)\right)$.

**4.5. SVD and Eigenvalue Methods for Sample Data Error Covariance.** It should be noted that the use of SVD in Sec. 4.4 is different than in [8], where spectral methods and SVD were advocated as a device to overcome the singularity of the data sample covariance matrix $R$, given as $R = SS^T/(N-1)$ with $S$ size $m \times N$. In that case, [8, eq. (56)] suggests the eigenvalue decomposition for the matrix

$$P = \frac{1}{N-1}\left( HA\,(HA)^T + SS^T \right) = Z \Lambda Z^T, \tag{4.3}$$

which is size $m$ by $m$ but there are only $N$ nonzero eigenvalues, i.e., diagonal entries of $\Lambda$. If the matrix $P$ is created explicitly, the cost of the decomposition (4.3) is between $O\left(m^2 N\right)$ and $O\left(m^3\right)$.

However, we note that in this case, $P$ can be written as

$$P = \frac{1}{N-1} F F^T, \quad F = [HA,\ S], \tag{4.4}$$

where $F$ has size $m$ by $2N$. Therefore, applying the SVD decomposition to the matrix $W$, we obtain $W = U \Sigma V^T$, and

$$P^{-1} = (N-1)\, U \Sigma^{-2} U^T. \tag{4.5}$$

The matrix $U$ is size $m$ by $m$ but it has only $2N$ onzero columns, and SVD routines actually return the $m$ by $2N$ submatrix, at the cost $O\left(mN^2\right)$. The resulting algorithm, using the multiplication by factored $P^{-1}$ from (4.5) in (2.1), has again the cost $O\left(N^3 + mN^2 + nN^2\right)$. [8, eq. (57)] discusses a method using SVD, with the same asymptotic cost, but requiring the data perturbation selected in a special way. No such constraint is needed here.

**4.6. Iterative Methods.** The linear system $PM = Y$ can be solved by conjugate gradients for $N$ right-hand sides. However, each iteration costs $O\left(mN^2\right)$, so iterative methods do not seem to be competitive.

**5. Distributed Parallel Implementation.** The method described in Sec. 4.2 was implemented in a distributed parallel environment using MPI and SCALAPACK. EnKF is naturally parallel: each ensemble member can be advanced in time independently. The linear algebra in the Bayesian update step links the ensemble members together. The ensemble matrix $X$ is then naturally distributed so that each process owns a block of columns. However, such distribution is a bottleneck to parallelism: SCALAPACK requires that all matrices involved in an operation must be distributed on the same processor grid (though possibly with different block sizes), and, for best performance, the processor grid should be close to a square. 1D processor grids tend to be particularly inefficient. Therefore, the ensemble matrix must be redistributed before the matrix linear algebra operations.

**6. Acknowledgements.** Section 4.4 is based on a discussion with Andrew Knyazev and Craig Johns. The author would like to thank Craig Johns and Mingshi Chen for useful discussions about the algebra of EnKF, and Jonathan Beezley for reading this paper. Thanks are also due to Jonathan Beezley, Craig Douglas, Deng Li, and Adam Zornes for contributing to an object oriented interface to SCALAPACK, to Craig Douglas and Wei Li for assistance with MPI wrappers, and to Jonathan Beezley for assistance with a parallel distributed implementation of EnKF on top of SCALAPACK. This research was supported by the National Science Foundation under the grant CNS-0325314. Computer time on IBM BlueGene/L supercomputer was provided by NSF MRI Grants CNS-0421498, CNS-0420873, and CNS-0420985, NSF sponsorship of the National Center for Atmospheric Research, the University of Colorado, and a grant from the IBM Shared University Research (SUR) program.

REFERENCES

[1] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, third ed., 1999.

[2] J. L. Anderson, *An ensemble adjustment Kalman filter for data assimilation*, Monthly Weather Review, 129 (1999), pp. 2884–2903.

[3] J. L. Anderson and S. L. Anderson, *A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts*, Monthly Weather Review, 127 (1999), pp. 2741–2758.

[4] T. Bengtsson, C. Snyder, and D. Nychka, *Toward a nonlinear ensemble filter for high dimensional systems*, Journal of Geophysical Research - Atmospheres, 108(D24) (2003), pp. STS 2–1–10.

[5] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley, *ScaLAPACK Users' Guide*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[6] G. Burgers, P. J. van Leeuwen, and G. Evensen, *Analysis scheme in the ensemble Kalman filter*, Monthly Weather Review, 126 (1998), pp. 1719–1724.

[7] G. Evensen, *Sequential data assimilation with nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics*, Journal of Geophysical Research, 99 (C5) (1994), pp. 143–162.

[8] G. Evensen, *The ensemble Kalman filter: Theoretical formulation and practical implementation*, Ocean Dynamics, 53 (2003), pp. 343–367.

[9] ———, *Sampling strategies and square root analysis schemes for the EnKF*, Ocean Dynamics, (2004), pp. 539–560.

[10] G. H. Golub and C. F. V. Loan, *Matrix Computations*, Johns Hopkins Univ. Press, 1989. Second Edition.

[11] W. W. Hager, *Updating the inverse of a matrix*, SIAM Rev., 31 (1989), pp. 221–239.

[12] P. Houtekamer and H. L. Mitchell, *Data assimilation using an ensemble Kalman filter technique*, Monthly Weather Review, 126 (1998), pp. 796–811.

[13] C. J. Johns and J. Mandel, *A two-stage ensemble Kalman filter for smooth data assimilation*. Environmental and Ecological Statistics. Conference on New Developments of Statistical Analysis in Wildlife, Fisheries, and Ecological Research, Oct 13-16, 2004, Columbia, MI, in print.

[14] R. E. Kalman, *A new approach to linear filtering and prediction problems*, Transactions of the ASME – Journal of Basic Engineering, Series D, 82 (1960), pp. 35–45.

[15] J. Mandel and J. D. Beezley, *Predictor-corrector ensemble filters for the assimilation of sparse data into high dimensional nonlinear systems*. CCM Report 232, University of Colorado at Denver and Health Sciences Center, 2006; http://www.math.cudenver.edu/ccm/reports, May 2006.

[16] M. K. Tippett, J. L. Anderson, C. H. Bishop, T. M. Hamill, and J. S. Whitaker, *Ensemble square root filters*, Monthly Weather Review, 131 (2003), pp. 1485–1490.

[17] P. van Leeuwen, *A variance-minimizing filter for large-scale applications*, Monthly Weather Review, 131 (2003), pp. 2071–2084.

## CENTER FOR COMPUTATIONAL MATHEMATICS REPORTS

University of Colorado at Denver and Health Sciences Center
P.O. Box 173364, Campus Box 170
Denver, CO 80217-3364

Fax: (303) 556-8550
Phone: (303) 556-8442
http://www-math.cudenver.edu/ccm

220 Andrew V. Knyazev and Merico E. Argentati, "Implementation of a Preconditioned Eigensolver Using Hypre." April 2005.

221 Craig Johns and Jan Mandel, "A Two-Stage Ensemble Kalman Filter for Smooth Data Assimilation." April 2005

222 I. Lashuk, M. E. Argentati, E. Ovchinnikov and A. V. Knyazev, "Preconditioned Eigensolver LOBPCG in hypre and PETSc." September 2005.

223 A. V. Knyazev and M. E. Argentati, "Majorization for Changes in Angles Between Subspaces, Ritz values, and graph Laplacian spectra." September 2005.

224 D. W. Dean, "A Semimartingale Approach For Modeling Multiphase Flow In Heterogeneous Porous Media." November 2005.

225 D.W. Dean, T.H. Illangasekare and T.F. Russell, "A Stochastic Differential Equation Approach For Modeling Of NAPL Flow In Heterogeneous Porous Media." November 2005.

226 J. Mandel and B. Sousedík, "Adaptive Coarse Space Selection in the BDDC and the FETI-DP Iterative Substructuring Methods: Optimal Face Degrees of Freedom." November 2005.

227 J. Mandel and B. Sousedík, "Adaptive Selection of Face Coarse Degrees of Freedom in the BDDC and the FETI-DP Iterative Substructuring Methods." November 2005.

228 Craig C. Douglas, Jonathan D. Beezley, Janice Coen, Deng Li, Wei Li, Alan K. Mandel, Jan Mandel, Guan Qin, and Anthony Vodacek, "Demonstrating the Validity of a Wildfire DDDAS." February 2006.

229 Lynn Schreyer Bennethum, "Flow and Deformation: Understanding the Assumptions and Thermodynamics, Updated." February 2006.

230 Lynn Schreyer Bennethum, "Compressibility Moduli for Porous Materials Incorporating Volume Fraction with Detailed Computations." February 2006.

231 Jan Mandel, "Efficient Implementation of the Ensemble Kalman Filter." May 2006.