Author Response to:
Interactive comment on "A local particle filter for high dimensional geophysical systems" *by* S. G. Penny and T. Miyoshi

C. Snyder (Referee)

chriss@ucar.edu

Received and published: 18 March 2016

Review of: A local particle filter for high dimensional geophysical systems, by S. Penny and T. Miyoshi.

Reviewed by: C. Snyder, NCAR

Recommendation: Requires major revision

This manuscript considers an approach for a spatially local update step in particle filters. This is an interesting and potentially important topic and I think the authors have an idea that is worth pursuing. On the other hand, the manuscript is overly terse and sometimes obscure in describing the method, and includes only limited analysis of results. My comments focus on those issues.

> We thank Chris for his thoughtful and thorough comments. Upon the reviewer's suggestions, we add further explanation of the method via additional equations (as requested below). We also offer that perhaps it would be clearer to add an appendix containing a simplified pseudo-code for the LPF algorithm used in this study, which we have included below.

Major comments:

1. The details of the LPF implementation should be clarified:

a. How is the smoothing of weights implemented? An equation would be helpful.

> The smoothing is applied as such:

For a given model grid point *mi*, ensemble member *ki*, a set of *N* neighbor points N_p, and vector **a** from (13) acting as a resampling index given as a function of the model grid point and ensemble member, we have,

$$X^{LPF}_{\{mi,ki\}} = \frac{1}{2} X^{a}_{\{mi,ki\}} + \frac{1}{2N} \sum_{ni \in N_p} X^{b}_{\{mi,\mathbf{a}(ni,ki)\}}$$

The subscript indices indicate the row and column of the matrix. Here we define the

concept of a 'neighbor point' abstractly as a point near the analyzed grid point based on a specified distance metric.

b. The smoothing of weights will involve a length scale. Is that length scale related to the localization radius? If so, how and why?

> Not explicitly. The smoothing itself has no length scale, as it is applied in ensemble space for only 1 gridpoint at a time. However, we do search 1 gridpoint radius to identify which particle indices should be chosen for averaging at that point (as we describe in the equation above). It is not clear that increasing this radius would necessarily improve the analysis. In fact, in experiments in which we tried larger search radii, the accuracy was degraded vs. the 1-gridpoint search radius (we an example below in Figure 2).

c. The text mentions a transform function T (p 9, l 21; p 10, l 23). Is this the same as the ETKF transformation matrix T that appears in (11)? Is it related to the matrix E in (16), perhaps by a spatial smoothing applied to the columns of E? Please clarify.

> The transform function T in (11) is a general transform that does not explicitly refer to the ETKF. Rather, it is used to generically refer to either E in (16), or $E^{(j)}$ in (19).

2. Further analysis and diagnosis of the results would improve the manuscript.

a. Figure 5 shows that, for given ensemble size, the LPF has smaller error of the prior mean than LETKF only when there are more than 20 observations, despite the fact that the non-Gaussianity of the prior will even greater as the number of observations decreases below 20. This behavior is at odds with expectations and the authors' conclusions (p 16, l 4-8) that more nonlinearity or non-Gaussianity increases the advantage of the LPF relative to the LETKF. What's going on? One possibility is that there are problems with spatial continuity of members after the LPF update when observations are not dense.

> Below 20 observations, in this experiment scenario, both LETKF and the LPF exhibit filter divergence. It appears that in this regime the behavior of LPF, which essentially propagates randomly generated members until one catches an attracting basin to the true state by chance, is not as robust as LETKF, which instead has a forcing term that drives it toward the observations.

b. It would be helpful to include an illustration of the problems encountered without spatial smoothing of weights, and how smoothing ameliorates those problems.

> A reasonable request. Here is a sample,

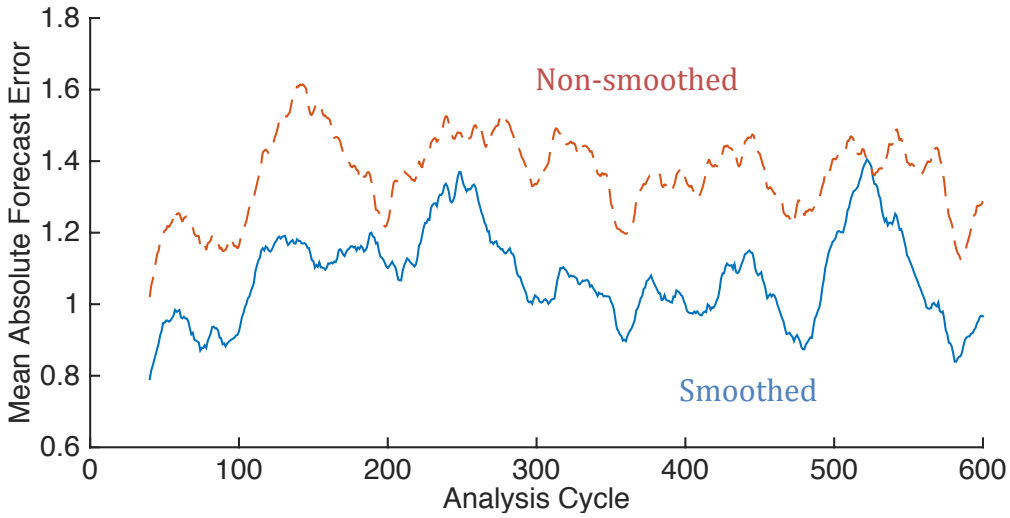

Figure 1. 40-cycle moving average of the mean absolute forecast error, comparing the Non-smoothed (dashed red) and Smoothed (solid blue) LPF analyses. The ensemble space smoothing improves the forecast accuracy.

By applying ensemble smoothing of the weights, we find that the mean absolute forecast error (averaged over the model domain) is reduced versus a non-smoothed analysis (Figure 1). Here we are considering the example of LPF applied to L96 with the analysis cycle dt=0.5 (producing a relatively non-linear error growth), a localization radius of 2 grid points, k=100 ensemble members, l=40 observations per cycle, and observation error of 0.5. Increasing the search radius to 2 gridpoints (Figure 2) for the same example case does not produce a clear benefit.

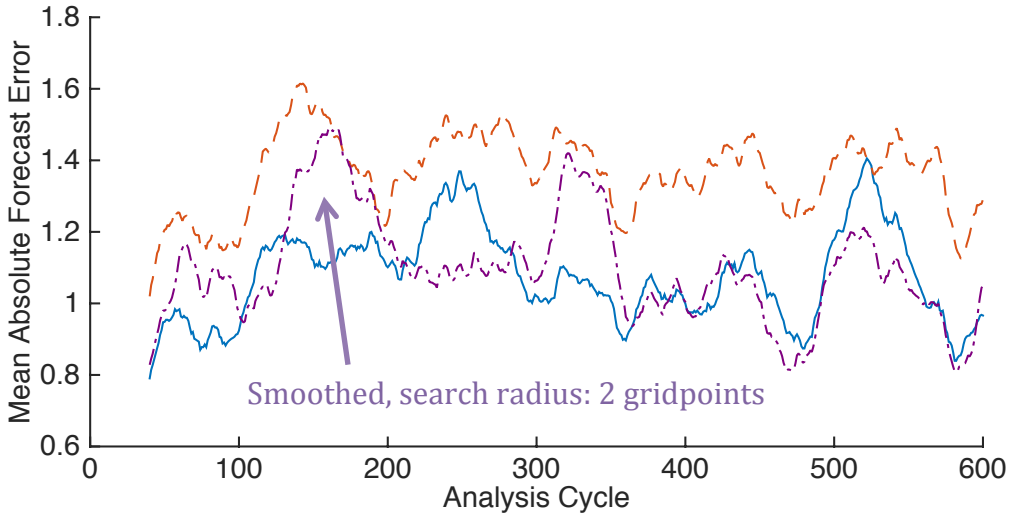

Figure 2. 40-cycle moving average of the mean absolute forecast error, comparing the Non-smoothed (dashed red) and Smoothed (solid blue) LPF analyses, with the addition of a Smoothed analysis using a 2-gridpoint search radius (dash-dot violet line) for the ensemble space smoothing. Increasing this search radius does not provide an obvious

improvement.

c. The manuscript has little discussion or analysis of how the weight-smoothing length scale or the localization radius might be chosen. This is a significant hole. The statements that do appear seem questionable: “For a given ensemble size, increasing the localization radius [beyond 2 grid points] degraded the accuracy of both methods.” (p14, l 3) I would expect the LETKF results to improve by increasing the localization radius as the ensemble size increases, at least for dt = 0.05.

> We have discussed the weight smoothing above. The choice of localization radius is problem dependent, and in this case a smaller radius is typically better. We avoided this discussion as the impact of varying localization radius on LETKF for this L96 system has been addressed by Penny (2014; MWR). There it was shown that for a fixed ensemble size, the analysis errors reduce when the localization radius decreases, and for a fixed localization radius, the analysis errors reduce when the ensemble size increases. The LPF showed similar behavior to LETKF in this regard.

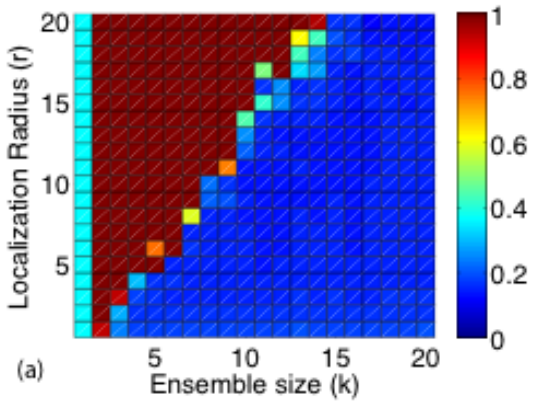

Figure 3. Analysis error when varying localization radius and ensemble size for LETKF, with a fixed observation density of l=20. For a fixed ensemble size, analysis errors reduce when localization radius decreases. For a fixed localization radius, analysis errors reduce when ensemble size increases. *Reproduced from (Penny, 2014).*

d. Some examination of the behavior of the weights would also be helpful (e.g. statistics of maximum weight or effective sample size). Figure 5 shows that the performance of the LPF is almost independent of ensemble size beyond ~75, which suggests that the localization is sufficient to keeps the weights reasonably distributed. But if that is the case, what is limiting the performance of the LPF for ~20 observations or fewer?

> As an example, using the smoothed analysis the mean effective ensemble size N_{eff},

$$N_{eff} = \left[\sum_{i=1}^{j} \left(w_t^i \right)^2 \right]^{-1},$$

with w_t^i defined as in (8), calculated at each gridpoint and averaged, is increased versus

the non-smoothed approach.

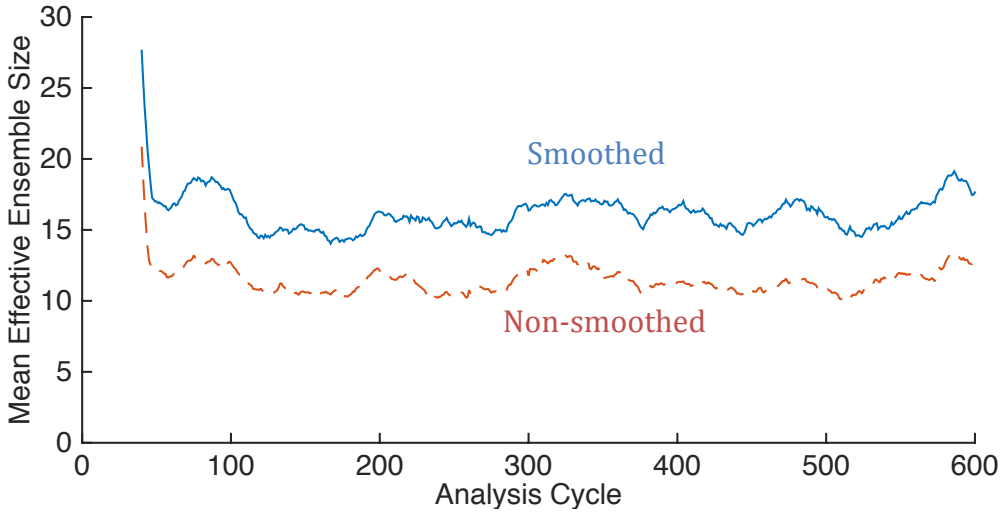

Figure 4. Illustrating the impact of ensemble space smoothing on the effective ensemble size N_{eff}.. Here we show the 40-cycle moving average of N_{eff} for the Smoothed (solid blue) and Non-smoothed (dashed red) cases.

As discussed earlier, using less than 20 observations with an analysis cycle of dt=0.5 leads to filter divergence of both the LETKF and LPF methods.

3. Literature that should be referenced and discussed:

a. Reich (2013, SIAM J. Sci. Computing) introduces the notion of a particle filter based on transformations. Another take on the transformation view of nonlinear filtering is Metref et al. (2014, NPG); they point out the appeal of such techniques in allowing smooth spatial localization.

> We would be happy to cite the contributions of both Reich (2013) and Metref et al. (2014):

Reich, S., 2013: A nonparametric ensemble transform method for Bayesian inference. SIAM J. Sci. Comput., 35, A2013–A2024, doi:10.1137/130907367.

Metref, S., E. Cosme, C. Snyder, and P. Brasseur, 2014: A non-Gaussian analysis scheme using rank histograms for ensemble data assimilation. *Nonlin. Processes Geophys.*, **21**, 869–885.

b. Lei and Bickel (2011) introduce the notion of computing local weights in a non-Gaussian filter, based on subset of observations that are near a given location. They apply this idea to computing the posterior mean. (Lei and Bickel are cited, but not for this contribution.)

> We will include this contribution from Lei and Bickel.

c. Bengtsson et al. (2003, JGR) were the first to point to spatially local updating, using a local subset of observations, as a solution to difficulties of high-dimensional non-Gaussian filtering.

> We will reference Bengtsson et al. (2003) as the originators of the spatially local updating in the geophysical filtering problem.

d. Houtekamer and Mitchell (1998) deserve citation. They introduced localization.

> We will add a reference to Houtekamer and Mitchell (1998) in our introduction of the localization approach.

4. The conclusions should temper the claim that the dense-observation results from the simple model will be relevant to the atmosphere. Yes, many observing systems provide spatially dense observations, but typically only a subset of prognostic variables (or, even worse, a complicated function of a subset of variables) is observed. In that case, it will still be important to use information from the prior in updating unobserved variables, a situation that is more analogous to the few-observations portion of Fig. 5 where the LPF does not perform well.

> We agree further tests are needed for a realistic system. Ultimately, we envision a type of hybrid filter being employed in such a scenario. We suggest this modification to the text:

"*While the PF provides a means of assimilating observations with non-Gaussian errors (e.g. precipitation, sea-ice concentration), we caution that the covariances utilized by the EnKF play a critical role in constraining the unobserved variables*. Thus while the LPF is not optimal for all possible data assimilation scenarios, there is great potential for the LPF to be combined with more traditional approaches to create adaptive hybrid systems that can avoid catastrophic filter divergence and manage multi-modal forecast distributions, nonlinear observation operators, non-Gaussian observations."

Minor comments: 1. Section 2.4:

a. A brief description would be welcome of how the amplitude of the additive noise was chosen and the sensitivity (or not) of the LPF results to that amplitude.

> The amplitude of the additive noise was chosen to conform to the dynamics of the growing error subspace, as estimated by the analysis ensemble spread. We note that this amplitude varies spatially and temporally. The results degraded when departing from this approach.

b. Many PFs apply some additional noise or "jitter" at the resampling step, or sample from a mixture of Gaussians centered at the particles, rather than directly from the empirical distribution. It's worth referencing that this is a standard, if empirical, technique.

> We will add references to indicate this is a common technique.

c. I don't see that rank is a relevant concept here. The question is whether the particles are "sufficiently" different. Please re-phrase.

> We will rephrase to make that point clear.

d. The "desirable properties" for the analysis ensemble that the authors quote from Pazo et al. are at best distracting and at worst misleading. The update step for this algorithm (i.e. the computation of weights from the observation likelihoods) assumes that the particles are a draw from the prior distribution. The forecast step achieves this (assuming it accounts appropriately for model error) as long as the analysis ensemble is a draw from the posterior distribution. That's all (!) that we require, yet of the conditions quoted only (3) is directly related to the posterior distribution.

> We acknowledge that under perfect conditions, for which the assumptions of the PF are upheld, that the PF achieves the optimal analysis. However, we are addressing the scenario in which these conditions are not upheld and the PF begins to fail. In that case, we are essentially left with an initialization problem, and the guidelines of Pazo et al. are helpful. For example, we use (4) as a guide to apply the Gaussian noise in the inflation step using an amplitude determined by the magnitude of the analysis ensemble spread.

2. Section 3.3: The authors should note that relatively simple fixes are available for the LETKF in this case. One might implement a basic quality control based on comparing the observation-minus-forecast difference to, say, 3 or 4 times its predicted standard deviation based on the ensemble and the assumed observation-error variance, or explicitly assign the observation error to one component or the other of the mixture (7) by comparing observation likelihoods.

> Previous work in applying such a qc to ocean DA has indicated that a simple approach can be potentially dangerous. For instance, it is possible that LETKF will encounter intermittent ensemble contraction to enough of a degree that it immediately begins discarding observations and accelerating filter divergence.

Considering L96, even with Gaussian observation errors, if we use a slightly more conservative value and eliminate observations when the mean OMFs are greater than 5 times the forecast ensemble spread, comparing to the experiment scenario in section 3.3 it significantly degrades the analysis accuracy with LETKF (Figure 5), and likewise in the more nonlinear scenario (Figure 6).

We acknowledge there may be some modifications to improve the situation for LETKF, but contend that any such modifications would require significant research as this scenario violates the assumptions made in the derivation of LETKF and most other EnKFs in general.

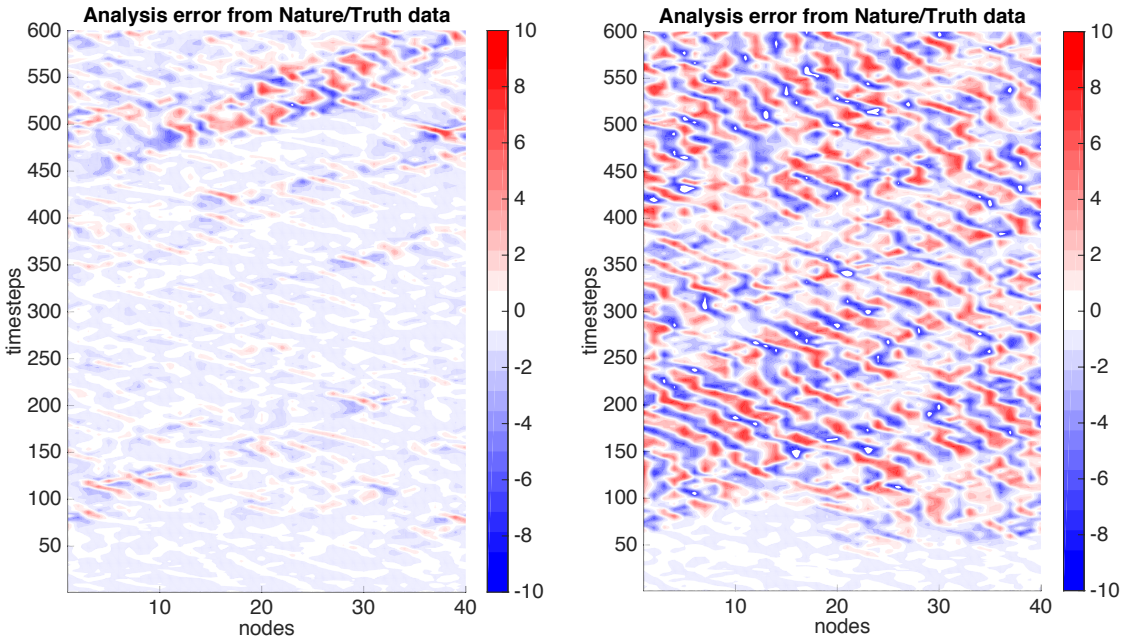

Figure 5. Using parameters from section 3.3 (l=80 observations and k=100 ensemble members, dt=0.05), applying a Gaussian mixture observation error with (left) all observations, and (right) QC that eliminates observations when the mean OMF is greater than 5 times the forecast ensemble spread.

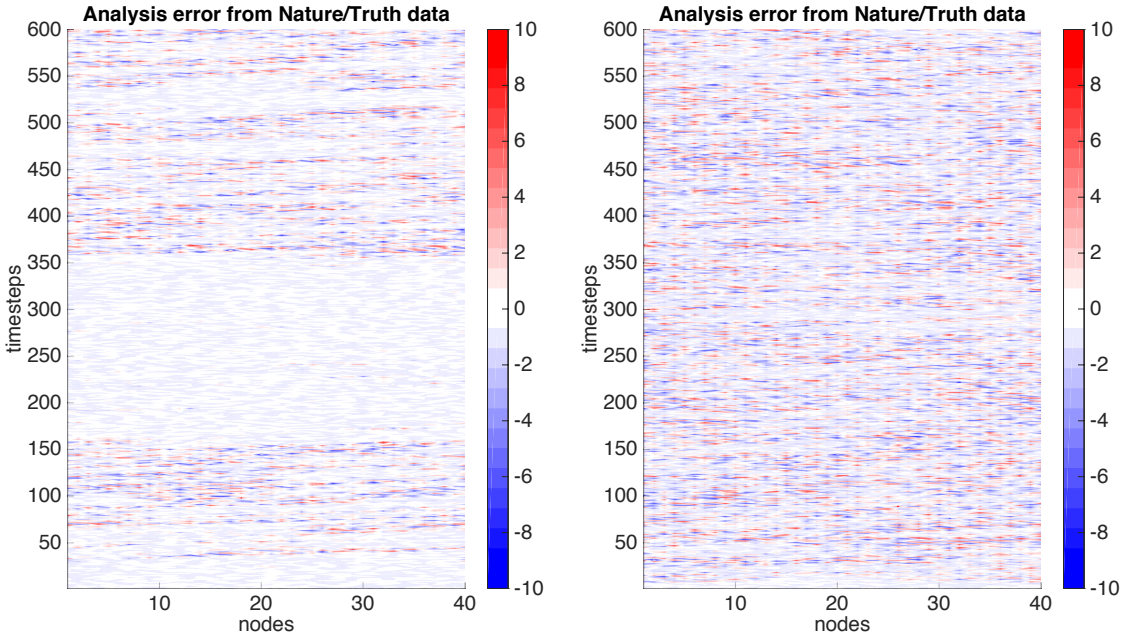

Figure 6. Using parameters from section 3.3 (l=80 observations and k=100 ensemble members, but increasing dt to dt=0.5), applying a (left) standard Gaussian observation

error distribution and a QC that eliminates obs when the mean OMF is greater than 5 times the forecast ensemble spread, and (right) the Gaussian mixture observation error.

p 4, l 7: This summary gives the impression that proposal densities for particle filters may be chosen to avoid the need exponentially large ensemble sizes as the problem size increases. Snyder et al. (2015) in fact show the opposite.

> We did not mean to give that impression, only that the proposal densities might also be combined with the LPF to improve its performance in the future.

p 15, l 10: The computational requirements for the LETKF scale as the cube of the ensemble size, not exponentially with ensemble size.

> Thanks for the correction.

In the appendix that follows, we attempt to present a simple documented pseduo-code to illustrate the LPF methodology in as clear and precise manner as possible. We left in a few details that we believe were useful in the generation of an accurate analysis.

Appendix:

Pseudo-code for LPF:

```
% INPUTS:
% k    :: ensemble size
% m    :: model dimension
% Yo   :: vector of global observations
% yo   :: vector of local observations
% Xb   :: background ensemble arranged as a matrix
% R    :: observation error covariance matrix
% HXb  :: Xb mapped from model space to observation space
%
% FUNCTIONS:
% obsloc :: finds observations local to a grid point
% cumsum :: cumulative sum of array
% sort   :: sorting of array in descending order
% repmat :: repeat array to form a matrix
%
% OUTPUTS:
% Neff :: effective ensemble sample size
% Xa   :: analysis ensemble arranged as a matrix

%
% Specify minimum of distribution tails
%
mintail = epsilon

%
% Setup global comb for resampling with replacement
%
interval = 1/k
start = interval * random_number
selection_points = [start : interval : start+(k-1)*interval]

%
% Loop over each grid point
%
for mi=1:m
    % Find the observation points within range of focal grid point:
    yo = obsloc(mi,Yo,local_radius)

    % Update (calculate particle weights) using desired distribution
    % (e.g. Gaussian shown here)
    for ki=1:k
       likelihood(ki) =
               exp( -0.5* (yo-HXb(:,ki))' * R^{-1} * (yo-HXb(:,ki)) )
    end

    % Protect against numerical representation problems in the tails
    likelihood = likelihood + mintail

    %
    % Normalize the weights
    %

    % Compute the weights
    weights = likelihood/sum(likelihood)
```

```
    % Calculate effective ensemble sample size
    Neff(mi) = 1/sum(weights.^2)

    % Form cumulative distribution
    [wts, wi] = sort(weights)
    weight  = cumsum(wts)

    %
    % Apply the comb to resample analysis members
    %
    j=1
    for i=1:k
        while selection_points(i) >= weight(j)
            j=j+1
        end

        % Specify the resampling index rs
        rs(mi,i)=wi(j)

        % Assign background value as global analysis
        Xa(mi,i) = Xb(mi,rs(mi,i))
    end

end % loop over model grid points

%
% Apply smoothing by weights in ensemble space.
%
X = 0
for ki=1:k
    for mi=1:m
        for ni={"set of all Np neighbor points"}
            X(mi,ki) = X(mi,ki) + Xa(mi,ki) + Xb(mi,rs(ni,ki))
        end
        Xa(mi,ki) = X(mi,ki)/(2*Np);
    end
end

%
% Apply additive inflation
%

% Compute the analysis ensemble spread to
% determine amplitude of inflation needed
if (mean(Neff) > k/2)
    Xstd=std(Xa,2)
else
    % add a little more noise to protect
    % against ensemble collapse
    Xstd=max(std(Xa,2),maximum_obs_error)
end

% Compute Gaussian random values with standard
% deviation equal to analysis ensemble spread
rmat=randn(m,k).*repmat(Xstd,1,k)

% Apply additive inflation (and remove sample mean)
Xa = Xa + rmat-repmat(mean(rmat,2),1,k)
```