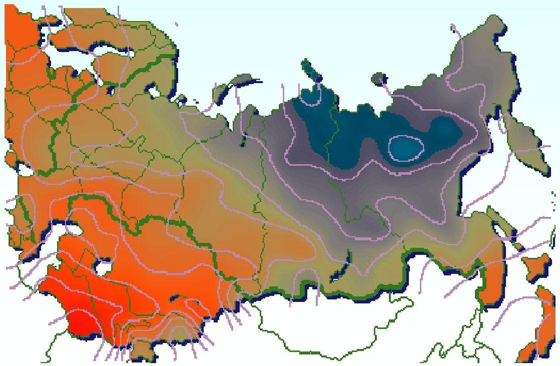
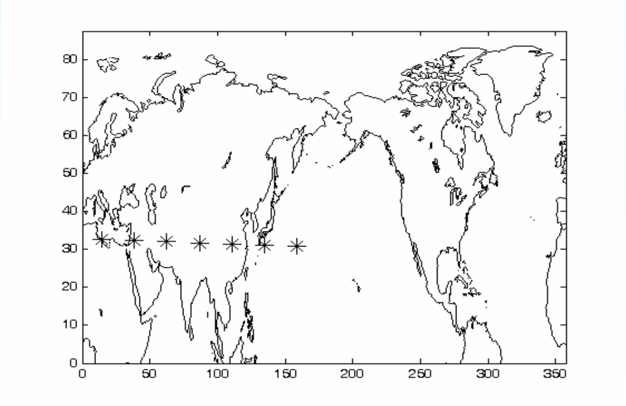
Effective algorithm of data assimilation based on ensemble Kalman filter

Ekaterina Klimova

Institute of Computational Technologies
Russian Academy of Sciences, Siberian Branch,
Novosibirsk, Russia,
klimova@ict.nsc.ru

Siberian Hydrometeorological Research Institute,
Novosibirsk, Russia





Introduction

1. The Kalman filter is one of the most popular approaches to solving problems of data assimilation.
2. The so-called ensemble approach is a leading method in the use of Kalman filter data assimilation. It allows one to calculate the estimation error covariance matrices for nonlinear prognostic models.
3. In the report, an efficient algorithm of observational data assimilation for nonlinear models with an ensemble of forecasts is proposed to evaluate the estimation error covariances. The method is based on ideas taken from automatic control theory
4. The ensemble Kalman filter, much like the conventional Kalman filter, is an algorithm difficult to implement technically, since it involves operations with high-order matrices.

Introduction

5. The operation count for the ensemble π-algorithm is near that for the Local Ensemble Transform Kalman Filter (LETKF) (Hunt et al., 2007; Szunyogh et al., 2008). However, the ensemble π-algorithm formulas are different from LETKF formulas and obtained in a different way.

6. The ensemble Kalman filter differs greatly from the classical Kalman filter in that the forecast error covariances are estimated by deviations of the ensemble elements from mean values. The algorithm proposed in this report can be extended to estimate the covariances by deviations of the ensemble elements from “true” values.

Plan of report

1. The ensemble π–algorithm formulas for nonlinear model and data operators.
2. An extension of the ensemble π-algorithm in which the covariance matrices are estimated by an ensemble of errors (deviations of the ensemble elements from “true” values).
3. A comparative analysis of the ensemble π-algorithm and LETKF formulas.
4. The results of numerical experiments on model data assimilation with the 1D Burgers equation.
5. Conclusions.

Π-algorithm

$$\hat{x}_{k+1} = M_k \hat{x}_k + P_{k+1} M^T_{k+1} R^{-1}_{k+1} (y_0 - H_{k+1} M_k \hat{x}_k)$$

$$x^t_{k+1} = M_k x^t_k + \eta^t_k$$

-the equation for estimated variable

$$\Delta x_k = x^t_k - x_k$$

$$\Delta x_{k+1} = M_k \Delta x_k - P_{k+1} H^T_{k+1} R^{-1}_{k+1} (y_0 - H_{k+1} H_k \hat{x}_k) + \eta^t_k$$

- ***the equation for estimation error***

Covariance matrix of estimation error:

$$P_{k+1} = \overline{\Delta x_{k+1} \Delta x^T_{k+1}}$$

$$P_{k+1} = \Delta x_{k+1} \Delta x^T_{k+1}$$

π ***-algorithm:***

$$\Delta x_{k+1} = M_k x_k + \Delta x_{k+1} \pi_{k+1}$$

$$\Delta x_{k+1} (1 + \pi_{k+1}) = M_k x_k + \eta^t_k$$

$$\pi_{k+1} = \Delta x^T_{k+1} H^T_{k+1} R^{-1}_{k+1} (y_0 - H_{k+1} M_k \hat{x}_k)$$

Krasovsky A.A., Beloglazov I.N., Chigin G.P. The correlation-extreme navigating systems theory. - Moscow, Nauka, 1979, 448 p. (in Russian).

The ensemble π–algorithm

Let's

$\mathbf{x}^f(t_{k+1})$ is the vector of predicted quantities at time t_{k+1}

$\mathbf{x}^a(t_k)$ is the vector of values obtained after the analysis step at time t_k

M is the model operator

$\boldsymbol{\eta}(t_k)$ is Gaussian white noise with covariance matrix $\mathbf{Q}_k$

The forecast step can be written as

$$\mathbf{x}^f(t_{k+1}) = M(\mathbf{x}^a(t_k)) + \boldsymbol{\eta}(t_k)$$

The ensemble π–algorithm

The analysis step is

$$\mathbf{x}^a(t_k) = \mathbf{x}^f(t_k) + \mathbf{P}_k^a \mathbf{H}^T \mathbf{R}_k^{-1}(\mathbf{y}_{t_k}^0 - H(\mathbf{x}^f(t_k))),$$

$\mathbf{P}_k^a$ is the co variance matrix of analysis errors,

$\mathbf{R}_k$ is the covariance matrix of observation errors,

H is an operator (generally nonlinear) which transforms values at grid nodes to values at observation points,

$\mathbf{H}$ is a linearized operator

$\mathbf{y}_{t_k}^0$ is the observation vector at time t_k

The ensemble π–algorithm

Write the algorithm in the following equivalent form:

$$\mathbf{x}(t_{k+1}) = M(\mathbf{x}(t_k)) + \boldsymbol{\eta}(t_k) + \mathbf{P}_{k+1}\mathbf{H}^T\mathbf{R}_{k+1}^{-1}(\mathbf{y}_{t_k}^0 - H(M(\mathbf{x}(t_k)) + \boldsymbol{\eta}(t_k))),$$

where

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{KH})\mathbf{P}_{k+1}^f, \mathbf{K} = (\mathbf{HP}_{k+1}^f\mathbf{H}^T + \mathbf{R}_{k+1})^{-1}$$

$\mathbf{P}_{k+1}^f$ is the covariance matrix of forecast errors

The ensemble π–algorithm

Specify an ensemble of initial fields

$$\mathbf{x}_n(t_0)=\mathbf{x}(t_0)+\Delta\mathbf{x}_n^0,\; n=1,\cdots;N,$$

where «*n*» is the vector number in the ensemble and

$\Delta\mathbf{x}_n^0$ is a vector of N random error fields with covariance $\mathbf{P}_0$

With these initial fields, we calculate N estimates as follows:

$$\mathbf{x}_n(t_{k+1})=M(\mathbf{x}_n(t_k))+\boldsymbol{\eta}_n(t_k)+\mathbf{P}_{k+1}\mathbf{H}^{T}\mathbf{R}_{k+1}^{-1}(\mathbf{y}_{t_{k+1}}^{0}-H(M(\mathbf{x}_n(t_k))+\boldsymbol{\eta}_n(t_k))).$$

The ensemble π–algorithm

Let a “true” quantity satisfy the equation

$$\mathbf{x}^t(t_{k+1}) = M(\mathbf{x}^t(t_k)),$$
$$\mathbf{x}^t(t_0) = \widehat{x}_0.$$

The ensemble of estimation errors is $\mathbf{\Delta x}_n^{k+1} = \mathbf{x}^t(t_{k+1}) - \mathbf{x}_n(t_{k+1})$

and since the “true” quantity is unknown, we assume
that the estimation errors are close to deviations from mean values

$$\mathbf{dx}_n^{k+1} = \mathbf{x}_n(t_{k+1}) - \overline{\mathbf{x}_n(t_{k+1})}$$

where

$$\overline{\mathbf{x}(t_{k+1})} \cong \frac{1}{N}\sum_{n=1}^{N}\mathbf{x}_n(t_{k+1}).$$

The ensemble π–algorithm

Then $\mathbf{dx}_n^{k+1}$ satisfies the following relation:

$$\mathbf{dx}_n{}^{k+1} = M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k) - \overline{M(\mathbf{x}_n(t_k))} - \mathbf{P}_{k+1}\mathbf{H}^{\,T}\mathbf{R}_{k+1}^{-1}(H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k)) -$$
$$\overline{H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k))}).$$

Estimating $\mathbf{P}_{k+1}$ by the following formula

$$\mathbf{P}_{k+1} = \overline{\mathbf{dx}_n^{k+1}\left(\mathbf{dx}_n^{k+1}\right)^T} \approx \frac{1}{N-1}\sum_{n=1}^{N}\mathbf{dx}_n^{k+1}\left(\mathbf{dx}_n^{k+1}\right)^T ,$$

we have a variant of the ensemble Kalman filter.

With this formula for $\mathbf{P}_{k+1}$ we obtain a system of equations for

$$\mathbf{dx}_n^{k+1}$$

The ensemble π–algorithm

Let **D** be an $L \times N$ matrix with vector columns

$$\{\mathbf{dx}_n^{k+1}, n = 1, \ldots, N\}$$

We denote $$\mathbf{f}_n^k = M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k) - \overline{M(\mathbf{x}_n(t_k))}$$

let **F** be a matrix with columns $\{\mathbf{f}_n^k, n = 1, \ldots, N\}$

Formula can be written in matrix form as follows:

$$\mathbf{D}^T = \mathbf{F}^T - \mathbf{\Pi}^T \mathbf{D}^T,$$

where $\mathbf{\Pi}$ is the $(N \times N)$ matrix with elements

$$(\mathbf{\Pi})_m^n = \frac{1}{N-1}\left(\mathbf{dx}_m^{k+1}\right)^T \mathbf{H}^T \mathbf{R}_{k+1}^{-1}(H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k)) - \overline{H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k))}),$$

The ensemble π–algorithm

Let $\tilde{\mathbf{f}}_n^k = H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k)) - \overline{H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k))}$

$\tilde{\mathbf{F}}$ a matrix with columns $\{\tilde{\mathbf{f}}_n^k, n = 1, \ldots, N\}$

Formula for $\boldsymbol{\Pi}$ is equivalent to the matrix equality

$$\boldsymbol{\Pi}^T = \frac{1}{N-1}\mathbf{D}^T\mathbf{H}^T\mathbf{R}^{-1}\tilde{\mathbf{F}}.$$

From formula for **D**, we obtain the following relations:

$$(\mathbf{I} + \boldsymbol{\Pi}^T)\mathbf{D}^T = \mathbf{F}^T, \qquad \mathbf{D}^T = (\mathbf{I} + \boldsymbol{\Pi}^T)^{-1}\mathbf{F}^T,$$

From this we obtain for the matrix $\boldsymbol{\Pi}$

$$(\mathbf{I} + \boldsymbol{\Pi}^T)\boldsymbol{\Pi}^T = \frac{1}{N-1}\mathbf{F}^T\mathbf{H}^T\mathbf{R}^{-1}\tilde{\mathbf{F}}.$$

The ensemble π–algorithm

Let $\mathbf{C} = \frac{1}{N-1}\mathbf{F}^T\mathbf{H}^T\mathbf{R}^{-1}\tilde{\mathbf{F}}$. From equation for $\mathbf{\Pi}$ we obtain the relation

$$(\mathbf{\Pi}^T + 0,5\mathbf{I})^2 = \mathbf{C} + 0,25\mathbf{I}.$$

For taking the root from $(\mathbf{C} + 0,25\mathbf{I})$ the matrix must be positive definite. In this case,

$$\mathbf{\Pi}^T = (\mathbf{C} + 0,25\mathbf{I})^{\frac{1}{2}} - 0,5\mathbf{I}.$$

It is evident that $\mathbf{\Pi} \geq 0$. If the operator H is linear, $\tilde{\mathbf{F}} = \mathbf{HF}$, and the matrix $\mathbf{C} = \frac{1}{N-1}\mathbf{F}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{HF}$ is symmetric and positive definite. In this case, the problem of taking the root from the matrix is solved easily and, hence, to determine the matrix $\mathbf{\Pi}$ it is better to use the linearized observation operator $\mathbf{H}$.

The ensemble π–algorithm

Thus, the ensemble π-algorithm consists of the following steps:

1. specify the $(L \times N)$ matrix **F**;
2. calculate the $(N \times N)$ matrix **C**;
3. calculate the $(N \times N)$ matrix $\mathbf{\Pi}$;
4. calculate the $(L \times N)$ matrix **D**;
5. calculate the ensemble of estimates

$$\mathbf{X}_n^{(k+1)T} = \mathbf{F}_2^T + \mathbf{\Pi}_2^T \mathbf{D}^T,$$

$$\left(\mathbf{F}_2^T\right)_n = M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k),$$

$$\left(\mathbf{\Pi}_2^T\right)_m^n = \frac{1}{N-1}\left(\mathbf{dx}_m^{k+1}\right)^T \mathbf{H}^T \mathbf{R}_{k+1}^{-1}(\mathbf{y}_{t_{k+1}}^0 - H(M(\mathbf{x}_k^n) + \boldsymbol{\eta}_k^n))).$$

The ensemble π–algorithm

It should be noted that the formulas of the ensemble π-algorithm presented in this section do not reproduce those of the classical Kalman filter if $\mathbf{y}^0$ is considered without additional perturbations. In fact, if an ensemble element at the analysis step has the form

$$\mathbf{x}_a^n = \mathbf{x}_b^n + \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y}^0 - \mathbf{H}\mathbf{x}_b^n),$$

(the superscript "*a*" denotes the variables obtained at the analysis step, and "*b*" denotes the variables obtained at the forecast step), the analysis error satisfies the equation

$$\mathbf{dx}_a^n = \mathbf{dx}_b^n - \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}\mathbf{dx}_b^n,$$

and the error covariance of the ensemble algorithm is

$$\mathbf{P}_a = (\mathbf{I} - \mathbf{KH})\mathbf{P}_b(\mathbf{I} - \mathbf{KH})^T .$$

This does not correspond to the equation for the error covariances of the Kalman filter (Jazwinsky, 1970)

Jazwinski A.H. 1970. *Stochastic processes and filtering theory*. Academic Press: New York.

The ensemble π–algorithm

With “perturbed” observations

$$\tilde{\mathbf{y}}^{0}_{t_k} = \mathbf{y}^{0}_{t_k} + \boldsymbol{\varepsilon}^{0}_{k},$$

and similar calculations, we obtain formulas that coincide with formulas 1)-5) and a matrix **C** that differs from the matrix obtained above:

$$\mathbf{C} = \frac{1}{N-1}\mathbf{F}^{T}\mathbf{H}^{T}\mathbf{R}^{-1}(\mathbf{HF}+\mathbf{E}) = \mathbf{C}_1 + \mathbf{C}_2.$$

Here $\mathbf{E}$ is the matrix whose columns are equal to the vector $\boldsymbol{\varepsilon}^{0}_{k}$. The matrix $\mathbf{C}_2$ is

$$\mathbf{C}_2 = \frac{1}{N-1}\left\{\mathbf{H}(\mathbf{Mdx}^{k}_{n} - \boldsymbol{\eta}_{n}(t_k))^{T}\right\}\mathbf{R}^{-1}\boldsymbol{\varepsilon}^{0}_{k+1}.$$

Under the assumption of space ergodicity, the formula for $\mathbf{C}_2$ is an estimate of the covariance

$$\mathrm{cov}\{[\mathbf{H}(\mathbf{Mdx}^{k}_{n} - \boldsymbol{\eta}_{n}(t_k))],[\left(\mathbf{R}^{-1}_{k+1}\boldsymbol{\varepsilon}^{0}_{k+1}\right)]\}\times\frac{J-1}{N-1}.$$

The ensemble π–algorithm

One can see from formula for the elements of matrix $\mathbf{\Pi}$ that this matrix is the same for all grid nodes at which the analysis is made. Owing to this property, the algorithm can be easily used for each grid node or for a group of nodes. Steps 1) - 4) of the ensemble π-algorithm can be made in the same way as in the Local Ensemble Transform Kalman Filter (LETKF) algorithm (Hunt et al., 2007) independently of grid nodes using the subvector $y^{0}_{t_{k+1},l}$ and the corresponding eigenmatrices **H** and **R** for each grid node.

At step 5), the calculations must be made for all grid nodes simultaneously. To calculate $\mathbf{X}_{n}^{(k+1)T}$ at a grid node, the matrix $\mathbf{\Pi}_{2}^{T}$ is calculated with the matrices **H** and **R** corresponding to this node and the subvector $y^{0}_{t_{k+1},l}$.

Extension of the ensemble π-algorithm

Let the “true” value x_t satisfy the equation

$$\mathbf{x}_t(t_{k+1}) = M(\mathbf{x}_t(t_k)),$$

$$\mathbf{x}_t(t_0) = \widehat{\mathbf{x}}_0.$$

Consider an ensemble of initial fields

$$\mathbf{x}_n(t_0) = \mathbf{x}_t(t_0) + \Delta\mathbf{x}_n^0,$$

an ensemble of “model noise” $\boldsymbol{\eta}_n(t_k)$, and data of observations

$$\mathbf{y}_{t_k}^0 = H(\mathbf{x}_t(t_k)) + \boldsymbol{\varepsilon}_k^0,$$

where $\boldsymbol{\varepsilon}_k^0$ are random observation errors with zero mean and covariance matrix $\boldsymbol{R}_k$.

The ensemble of estimates has the following form:

$$\mathbf{x}_n(t_{k+1}) = M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k) + \mathbf{P}_{k+1}\mathbf{H}^T\mathbf{R}_{k+1}^{-1}(\mathbf{y}_{t_{k+1}}^0 - H(M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k))).$$

Extension of the ensemble π-algorithm

Determine the ensemble of estimation errors $\mathbf{dx}_n^{k+1} = \mathbf{x}_t(t_{k+1}) - \mathbf{x}_n(t_{k+1})$. The errors satisfy the following equation:

$$\mathbf{dx}_n{}^{k+1} = M(\mathbf{x}_t(t_k)) - M(\mathbf{x}_n(t_k)) - \boldsymbol{\eta}_n(t_k) -$$

$$\mathbf{P}_{k+1}\mathbf{H}^{\,T}\mathbf{R}_{k+1}{}^{-1}(H(M(\mathbf{x}_t(t_k)) + \boldsymbol{\varepsilon}_k^0 - H(M(\mathbf{x}_n(t_k) + \boldsymbol{\eta}_n(t_k))).$$

Consider the case when linearized operators $\mathbf{M}$ and $\mathbf{H}$ are used to calculate the error:

$$\mathbf{dx}_n{}^{k+1} = \mathbf{Mdx}_n{}^{k} - \boldsymbol{\eta}_n(t_k) - \mathbf{P}_{k+1}\mathbf{H}^{\,T}\mathbf{R}^{-1}(\mathbf{H}(\mathbf{Mdx}_n{}^{k} - \boldsymbol{\eta}_n(t_k)) + \boldsymbol{\varepsilon}_k^0).$$

After calculations similar to those in the previous section (omitting the superscript “*k*”), we obtain:

$$\mathbf{D}^T = (\mathbf{I} + \boldsymbol{\Pi}^T)^{-1}\mathbf{F}^T,$$

$$\boldsymbol{\Pi}^T = (\mathbf{C} + 0,25\mathbf{I})^{1/2} - 0,5\mathbf{I},$$

$$\mathbf{C} = \frac{1}{N-1}\mathbf{F}^T\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{HF} + \mathbf{E}) = \mathbf{C}_1 + \mathbf{C}_2.$$

Extension of the ensemble π-algorithm

Matrix **C** must be such that the square root from $(\mathbf{C}+0,25\mathbf{I})$ can be taken. For $\mathbf{C}_2 \cong \mathbf{0}$, **C** is symmetric and positive. Therefore, the square root can be calculated by determining the eigenvectors and eigennumbers of **C**. The matrix $\mathbf{C}_2$, under the assumption of space ergodicity, is an estimate of the covariance

$$\operatorname{cov}\{[\mathbf{H}(\mathbf{Mdx}_n^k-\boldsymbol{\eta}_n(t_k))],[\left(\mathbf{R}_{k+1}^{-1}\boldsymbol{\varepsilon}_{k+1}^0\right)]\}\times\frac{J-1}{N-1}$$

And the elements of $\mathbf{C}_2$ can be considered close to zero.

This variant of the ensemble π-algorithm can be implemented in the following form:

$$\mathbf{X}_n^{(k+1)T} = \mathbf{F}_2^T + \mathbf{\Pi}_2^T\mathbf{D}^T,$$

$$\left(\mathbf{F}_2^T\right)_n = M(\mathbf{x}_n(t_k)) + \boldsymbol{\eta}_n(t_k),$$

$$\left(\mathbf{\Pi}_2^T\right)_m^n = \frac{1}{N-1}\left(\mathbf{dx}_m^{k+1}\right)^T \mathbf{H}^T\mathbf{R}_{k+1}^{-1}(\mathbf{y}_{t_{k+1}}^0 - H(M(\mathbf{x}_k^n)+\boldsymbol{\eta}_k^n))).$$

If $(M(\mathbf{x}_k^n)-M(\mathbf{x}_k^t))$ is assumed to be approximately equal to $(M(\mathbf{x}_k^n)-M(\overline{\mathbf{x}}_k))$, $\mathbf{dx}_{k+1}^n$ can be calculated without the linearized operator $\mathbf{M}$.

Extension of the ensemble π-algorithm

Consider a more general variant with nonzero elements of $\mathbf{C}_2$, and solve the following nonlinear matrix equation for $\mathbf{D}^T$:

$$(\mathbf{I}+\mathbf{\Pi}^T)\mathbf{D}^T=\mathbf{F}^T,$$

where

$$\mathbf{\Pi}^T=\frac{1}{N-1}\mathbf{D}^T\mathbf{H}^T\mathbf{R}^{-1}\tilde{\mathbf{F}},$$

$\mathbf{F}$ is the matrix with columns $\{\mathbf{f}_n^k, n=1,\ldots,N\}$, $\mathbf{f}_n^k=\mathbf{Mdx}_n^k-\boldsymbol{\eta}_n(t_k)$ and

$\tilde{\mathbf{F}}$ is the matrix with columns $\{\tilde{\mathbf{f}}_n^k, n=1,\ldots,N\}$, $\tilde{\mathbf{f}}_n^k=\mathbf{y}_{t_{k+1}}^0-H(M(\mathbf{x}_n(t_k)))$.

The nonlinear equation can be solved by the following iterative method:

$$(\mathbf{I}+\mathbf{\Pi}^T)\left(\mathbf{D}^T\right)^{l+1}=\mathbf{F}^T,$$

$$\mathbf{\Pi}^T=\frac{1}{N-1}\left(\mathbf{D}^T\right)^l\mathbf{H}^T\mathbf{R}^{-1}\tilde{\mathbf{F}},$$

where l is the iteration number. The iterative process will converge if $\left\|(\mathbf{I}+\mathbf{\Pi}^T)\right\|$ is bounded, since in this case the convergence conditions of Newton's method are satisfied.

Extension of the ensemble π-algorithm

In the algorithm that is proposed here, $\mathbf{x}-\mathbf{x}_t$ (where $\mathbf{x}_t$ is a "true" value), but not $\mathbf{x}-\bar{\mathbf{x}}$ is used as the estimation error, which makes the algorithm closer to the extended Kalman filter. In this case, $\|\mathbf{dx}\|\to 0$ means that the error but not the deviation from a "mean" decreases.

It should be noted that this algorithm can be implemented independently of grid nodes from some subdomains; in this case, the subvector $y^0_{t_{k+1},l}$ and the corresponding eigenmatrices **H** and **R** must be used in each subdomain. To decrease the "false" covariances at large distances, the covariance $\mathbf{P}_{k+1}\mathbf{H}^T$ can be multiplied by a matrix function $\mathbf{\Phi}(\boldsymbol{\rho})>0$, where $\boldsymbol{\rho}$ is the distance between the grid node and the observations. Here the matrix **C** has the following form:

$$\mathbf{C}=\frac{1}{N-1}\mathbf{F}^T\mathbf{H}^T\mathbf{\Phi}(\boldsymbol{\rho})\mathbf{R}^{-1}(\mathbf{HF}+\mathbf{E}).$$

A comparison of the ensemble π-algorithm and LETKF algorithm

The analysis step of the proposed algorithm can be represented in the following form:

$$\mathbf{x}_a^n = \mathbf{x}_b^n + \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y}^0 + \boldsymbol{\varepsilon}_0^n - \mathbf{H}\mathbf{x}_b^n),$$

$$\mathbf{P}^a = \frac{1}{N-1} \mathbf{D}\mathbf{x}_a \mathbf{D}\mathbf{x}_a{}^T,$$

$$\mathbf{dx}_a^n = \mathbf{dx}_b^n - \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}\mathbf{dx}_b^n + \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} \boldsymbol{\varepsilon}_0^n.$$

A comparison of the ensemble π-algorithm and LETKF algorithm

In LETKF, the analysis step is made only for ensemble-average values (Hunt et al., 2007):

$$\overline{\mathbf{x}}_a = \overline{\mathbf{x}}_b + \mathbf{P}^a \mathbf{H}^T \mathbf{R}^{-1} (\mathbf{y}^0 - \mathbf{H}\overline{\mathbf{x}}_b),$$

and

$$\mathbf{P}_a \mathbf{H}^T \mathbf{R}^{-1} = \mathbf{P}_b \mathbf{H}^T (\mathbf{H}\mathbf{P}_b \mathbf{H}^T + \mathbf{R})^{-1},$$

$$\mathbf{P}_b = \frac{1}{N-1} \mathbf{Dx}_b \mathbf{Dx}_b{}^T,$$

$\mathbf{Dx}_a$ is found from the condition $\mathbf{P}_a = \mathbf{Dx}_b \hat{\mathbf{A}} \mathbf{Dx}_b{}^T = \frac{1}{N-1} \mathbf{Dx}_a \mathbf{Dx}_a{}^T$. The matrix $\hat{\mathbf{A}}$ has the following form (Hunt et al., 2007):

$$\left[(N-1)\mathbf{I} + (\mathbf{H}\mathbf{Dx}_b)^T \mathbf{R}^{-1} (\mathbf{H}\mathbf{Dx}_b) \right]^{-1}.$$

Hunt B.R., Kostelich E.J., Szunyogh I. 2007. Efficient data assimilation for statiotemporal chaos: A local ensemble transform Kalman filter. *Physica D.* 230: 112-126.

A comparison of the ensemble π-algorithm and LETKF algorithm

The ensemble element after the analysis step in LETKF can be written as

$$\begin{aligned}&\mathbf{x}_a^n = \mathbf{x}_b^n + \mathbf{K}(\mathbf{y}^0 - \mathbf{H}\mathbf{x}_b^n) + (\mathbf{I} - \mathbf{KH})(\overline{\mathbf{x}}_b - \mathbf{x}_b^n) - \mathbf{K}\boldsymbol{\varepsilon}_0^n + \mathbf{dx}_a^n,\\ &\mathbf{K} = \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1}.\end{aligned}$$

That is, the equation for the ensemble element $\mathbf{x}_a^n$ in LETKF differs from equation for $\mathbf{x}_a^n$ in the ensemble π-algorithm by the term $\mathbf{x}' = (\mathbf{I} - \mathbf{KH})(\overline{\mathbf{x}}_b - \mathbf{x}_b^n) - \mathbf{K}\boldsymbol{\varepsilon}_0^n + \mathbf{dx}_a^n$ in the right-hand side.
In this case, we have

$$\overline{\mathbf{x}'} = 0, \overline{\mathbf{x}'\mathbf{x}'^T} = 0.$$

It should be noted that the matrices $\mathbf{P}^a$ in this equations are different. Since in the ensemble algorithms $\overline{(\cdot)} \cong \frac{1}{N}\sum_{n=1}^{N}(\cdot)_n$, $\overline{\mathbf{x}'} = 0$ with a large error, since this estimate converges as $O\left(1\big/\sqrt{N}\right)$.

Numerical experiments

To assess the possibilities of practical implementation of the proposed algorithm, a series of numerical experiments has been performed with the 1D nonlinear Burgers equation:

$$\frac{\partial u}{\partial t}+\frac{u}{a\sin\theta_0}\frac{\partial u}{\partial\lambda}=\alpha\frac{1}{(a\sin\theta_0)^2}\frac{\partial^2 u}{\partial\lambda^2}.$$

The equation was solved for a circle of latitude $\theta_0 = 45^\circ$ with periodic boundary conditions. The equation was solved by a Leap Frog/DuFort-Frankel finite-difference scheme on the basis of an example presented in (Kalnay E., 2002). The solution was sought for grid nodes with $\Delta\lambda = 2{,}5^\circ$, and the time step was $\Delta t = 1$ hour. The parameter α was taken to be 0,001. The initial data were taken as a large-scale wave,

$$u(\lambda) = U_0 \sin(\lambda),$$

$U_0 = 10 ms - 1$.

Kalnay E. 2002. *Atmospheric Modeling, Data Assimilation and Predictability*. Cambridge Univ. Press.

Numerical experiments

The observational data were produced by using the same model ("twin"-type experiments). A prediction for 48 hours was calculated with observational data in a band of 36 grid nodes from number i_1 to number i_2 assumed to be available every 36 hours:

$$i_1 = 1 + (\frac{ntime}{6} - 1) \times 36,$$

$$i_2 = (i_1 - 1) + 36,$$

where *ntime* is the time step number (6, 12, 18, etc.).

As is customary in numerical experiments with model data, a "true" value was simulated by the same model, and the function

$$u^t(\lambda) = u(\lambda) + \zeta_0$$

was taken as initial condition for u^t. Here $\zeta_0 = \sigma_f N(0,1)$, $\sigma_f = 1$, and *N(0,1)* is a random quantity normally distributed with zero mean and unit variance. The observational data were specified by adding a random observation error $\varepsilon = \sigma_0 N(0,1)$, $\sigma_0 = 0{,}1$ to the "true" value $\mathrm{u}^{\mathrm{t}}(\lambda, \mathrm{t})$

Numerical experiments

To implement the ensemble π–algorithm and the LETKF algorithm, an ensemble of *N* initial fields was specified,

$$u_i^0 = u^0 + \Delta u_i^0 \,.$$

Here u^0 was considered to be a preliminary estimate of *x*, $\Delta \mathrm{u}_\mathrm{i}^0 = \sigma_\mathrm{f} \mathrm{N}(0,1)$, $\sigma_f = 1$.

N predictions with assimilation were calculated using the *N* initial fields, and the sought-for field was estimated at each time step by the formula

$$\bar{\mathrm{u}} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \mathrm{u}_\mathrm{i}.$$

As an estimate of the result, the root-mean-square deviation of $\bar{\mathrm{u}}$ from u^t ("truth") was used. The numerical experiments were made for *N=25*, where *N* is the number of ensemble elements. The covariance matrix of forecast errors at the initial time was specified as $\mathbf{P}_0 = \sigma_f{}^2\mathbf{I}$, and the covariance matrix of observation errors, as $\mathbf{R} = \sigma_0^2\mathbf{I}$, where **I** is the unit matrix, **Q**=*0*.

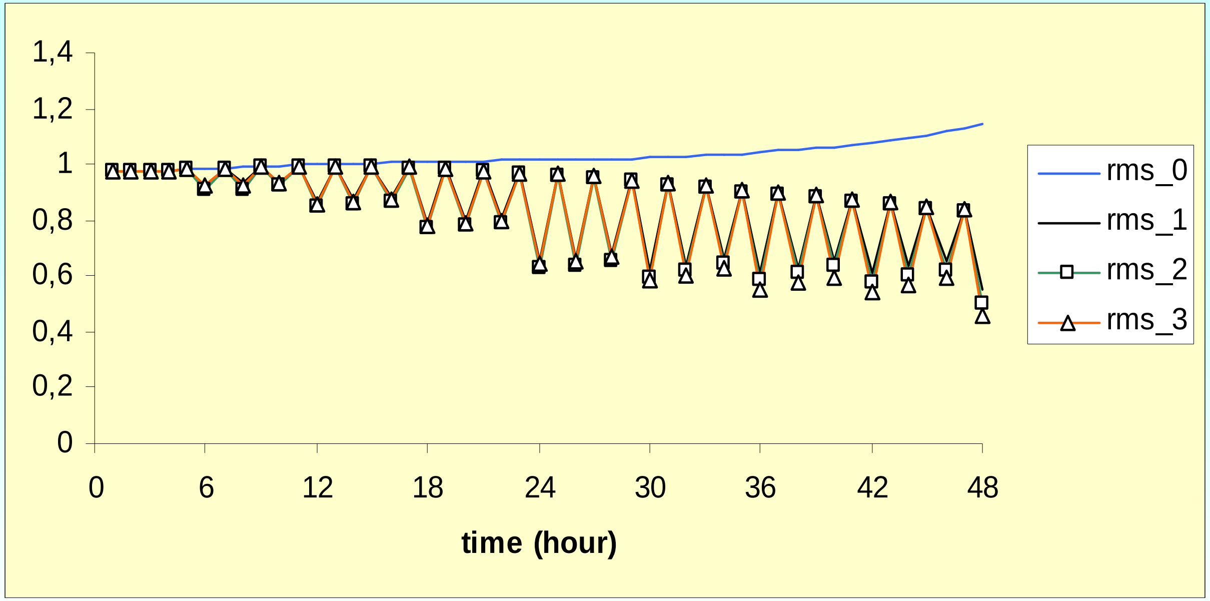
Numerical experiments

Since the covariances were described with a limited number of ensemble elements, some changes were introduced at the analysis step not only locally, at grid nodes located close to observations, but practically in the entire domain. In the first series of numerical experiments, the corrections were made in the domain where observations were specified. In this case, all data for each grid node were used.
The following numerical experiments were made:

1. Data assimilation with the ensemble π-algorithm (Section 2 formulas);
2. Data assimilation with the extended π-algorithm in which the error is considered as a deviation from a “truth” (Section 3 formulas);
3. Data assimilation with the LETKF algorithm

Numerical experiments

Figure 1 shows the results of this series of experiments. Here, *rms_0* is the root-mean-square error without assimilation, and *rms_1*, *rms_2*, and *rms_3* are the forecast errors obtained in experiments 1, 2, and 3, respectively. One can see from this figure that the forecast errors *rms_1*, *rms_2*, and *rms_3* are similar in behavior



Numerical experiments

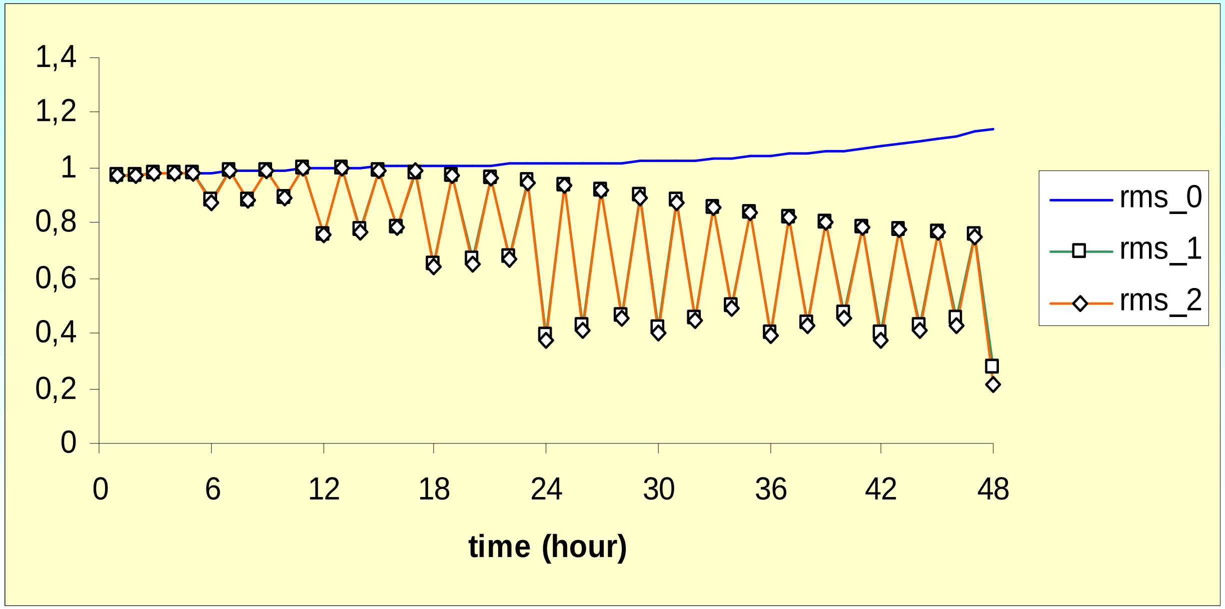
In a second series of experiments, a variant of the algorithms implemented for each grid node separately was considered. In this case, the data for the *k*th node were chosen from the interval *(k-3,k+3).* For each grid node, specific matrices, **H** and **R,** were formed. As can be seen from equations, to implement the ensemble π-algorithm locally, it is necessary first to calculate the matrix **DH**, and then calculate the ensemble of forecasts. The following numerical experiments were made:

1. Data assimilation with the extended π-algorithm in which the error is considered as a deviation from a “truth”;
2. Data assimilation with the LETKF algorithm.

Numerical experiments

Figure 2 shows the results of this series of experiments. Here, *rms_0* is the root-mean-square error without assimilation, and *rms_1* and *rms_2* are the forecast errors obtained in experiments 1 and 2, respectively.

One can see from this figure that the forecast error in the second series of experiments turned out to be smaller than that in the first series, and again the forecast errors with assimilation are practically similar in behavior.



Conclusions

- A suboptimal algorithm of data assimilation based on an ensemble approach was proposed in the report. The algorithm is based on the introduction of an equation for the estimation error and a solution to this equation is sought for to estimate the covariances.

- The major arithmetic operations are made with matrices comparable to the ensemble in size; the algorithm is close to the LETKF method in the number of arithmetic operations.

- The algorithm allows an extension when the estimation error is interpreted as a deviation from a “truth” but not from a mean, as in all ensemble assimilation algorithms.

- The algorithm can be implemented locally for individual grid nodes or a group of nodes. The algorithm formulas can be generalized by multiplying the covariances of the estimation error by a function of the distance between an observation point and a grid node. This decreases the impact of unrealistically high correlations that occur when the size of the ensemble is small at large distances.

Publication

1. Klimova E.G. 2008. A data assimilation technique based on the π-algorithm. *Russian Meteorology and Hydrology*. 33, No. 3: 143-150.

2. Klimova E.G.2008. Data assimilation technique based on the ensemble π-algorithm. *Russian Meteorology and Hydrology*. 33, No. 9: 570-576.

3. Klimova E.G.2010 Use of ensemble Kalman filter at additional observations planning. *Russian Meteorology and Hydrology (in press)*

4. Klimova E.G A suboptimal data assimilation technique based on the ensemble Kalman filter. QJRMS (*in review*).

Thanks for attanshion!