Introduction to Data Assimilation

Mike Fisher

ECMWF

May 31, 2015

Outline

ECMWF

The Evolution of Modern Data Assimilation

- Analysis: The process of approximating the true state of a (geo)physical system at a given time.
- For example:
  - Hand analysis of synoptic observations (1850 LeVerrier, Fitzroy).
  - Polynomial Interpolation (1950s Panofsky)
- An important step forward was made by Gilchrist and Cressman (1954), who introduced the idea of using a previous numerical forecast to provide a preliminary estimate of the analysis.
- This prior estimate was called the background.

The Evolution of Modern Data Assimilation

- Bergthorsson and Döös (1955) took the idea of using a background field a step further by casting the analysis problem in terms of increments which were added to the background.
- The increments were weighted linear combinations of nearby observation increments (observation minus background), with the weights determined statistically.
- This idea of statistical combination of background and synoptic observations led ultimately to Optimal Interpolation.
- The use of statistics to merge model fields with observations is fundamental to all current methods of analysis.

The Evolution of Modern Data Assimilation

- An important change of emphasis happened in the early 1970s with the introduction of primitive-equation models.
- Primitive equation models support inertia-gravity waves. This makes them much more fussy about their initial conditions than the filtered models that had been used hitherto.
- The analysis procedure became much more intimately linked with the model. The analysis had to produce an initial state that respected the model's dynamical balances.
- Unbalanced increments from the analysis procedure would be rejected as a result of geostrophic adjustment.
- Initialization techniques (which suppress inertia-gravity waves) became important.

The Evolution of Modern Data Assimilation

- The idea that the analysis procedure must present observational information to the model in a way in which it can be absorbed (i.e. not rejected by geostrophic adjustment) led to the coining of the term data assimilation.
- A final impetus towards the modern concept of data assimilation came from the increasing availability of asynoptic observations from satellite instruments.
- It was no longer sufficient to think of the analysis purely in terms of spatial interpolation of contemporaneous observations.
- The time dimension became important, and the model dynamics assumed the role of propagating observational information in time to allow a synoptic view of the state of the system to be generated from asynoptic data.

Elementary Statistics

Suppose we want to estimate the temperature of this room, given:

- A prior estimate: T_b.
  - E.g., we measured the temperature an hour ago, and we have some idea (i.e. a model) of how the temperature varies as a function of time, the number of people in the room, whether the windows are open, etc.
- A thermometer: T_o.

Denote the true temperature of the room by T^*.

- The errors in T_b and T_o are:

$$\begin{aligned} \epsilon_b &= T_b - T^* \\ \epsilon_o &= T_o - T^* \end{aligned}$$

- We will assume that the error statistics of T_b and T_o are known, and that T_b and T_o have been adjusted (bias corrected) so that their mean errors are zero:

$$\overline{\epsilon_b} = \overline{\epsilon_o} = 0$$

Elementary Statistics

- We estimate the temperature of the room as a linear combination of T_b and T_o:

$$T_a = \alpha T_o + \beta T_b + \gamma$$

- Denote the error of our estimate as $\epsilon_a = T_a - T^*$.
- We want the estimate to be unbiased: $\overline{\epsilon_a} = 0$.
- We have:

$$T_a = T^* + \epsilon_a = \alpha\left(T^* + \epsilon_o\right) + \beta\left(T^* + \epsilon_b\right) + \gamma$$

- Taking the mean and rearranging gives:

$$\overline{\epsilon_a} = \left(\alpha + \beta - 1\right) T^* + \gamma$$

- Since this holds for any T^*, we must have
  - $\gamma = 0$, and
  - $\alpha + \beta - 1 = 0$.
- I.e. $T_a = \alpha T_o + (1 - \alpha) T_b$

Elementary Statistics

- The general Linear Unbiased Estimate is:

$$T_a = \alpha T_o + (1-\alpha) T_b$$

- Now consider the error of this estimate.
- Subtracting T^* from both sides of the equation gives

$$\epsilon_a = \alpha \epsilon_o + (1-\alpha)\epsilon_b$$

- The variance of the estimate is:

$$\overline{\epsilon_a^2} = \alpha^2 \overline{\epsilon_o^2} + 2\alpha(1-\alpha)\overline{\epsilon_o \epsilon_b} + (1-\alpha)^2 \overline{\epsilon_b^2}$$

- The quantity $\overline{\epsilon_o \epsilon_b}$ represents the covariance between the error of our prior estimate and the error of our thermometer measurement.
- There is no reason for these errors to be connected in any way.
- We will assume that $\overline{\epsilon_o \epsilon_b} = 0$.

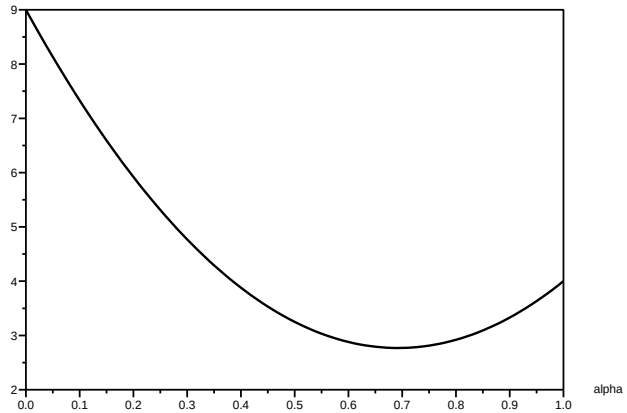
ECMWF

Elementary Statistics

$$\overline{\epsilon_a^2} = \alpha^2 \overline{\epsilon_o^2} + (1-\alpha)^2 \overline{\epsilon_b^2}$$

We can easily derive some properties of our estimate:

- $\frac{d\overline{\epsilon_a^2}}{d\alpha} = 2\alpha\overline{\epsilon_o^2} - 2(1-\alpha)\overline{\epsilon_b^2}$
- For $\alpha = 0$, $\overline{\epsilon_a^2} = \overline{\epsilon_b^2}$ and $\frac{d\overline{\epsilon_a^2}}{d\alpha} = -2\overline{\epsilon_b^2} < 0$
- For $\alpha = 1$, $\overline{\epsilon_a^2} = \overline{\epsilon_o^2}$ and $\frac{d\overline{\epsilon_a^2}}{d\alpha} = 2\overline{\epsilon_o^2} > 0$



From this we can deduce:

- For $0 \leq \alpha \leq 1$, $\overline{\epsilon_a^2} \leq \max(\overline{\epsilon_b^2}, \overline{\epsilon_o^2})$
- The minimum-variance estimate occurs for $\alpha \in (0,1)$.
- The minimum-variance estimate satisfies $\overline{\epsilon_a^2} < \min(\overline{\epsilon_b^2}, \overline{\epsilon_o^2})$

Elementary Statistics

The minimum-variance estimate occurs when

$$\frac{d\overline{\epsilon_a^2}}{d\alpha} = 2\alpha\overline{\epsilon_o^2} - 2(1-\alpha)\overline{\epsilon_b^2} = 0$$

$$\Rightarrow \quad \alpha = \frac{\overline{\epsilon_b^2}}{\overline{\epsilon_b^2} + \overline{\epsilon_o^2}}$$

It is not difficult to show that the error variance of this minimum-variance estimate is:

$$\overline{\epsilon_a^2} = \left(\frac{1}{\overline{\epsilon_b^2}} + \frac{1}{\overline{\epsilon_o^2}} \right)^{-1}$$

Extension to Multiple Dimensions

- Now, let's turn our attention to the multi-dimensional case.
- Instead of a scalar prior estimate T_b, we now consider a vector $\mathbf{x}_b$.
- We can think of $\mathbf{x}_b$ as representing the entire state of a numerical model at some time.
- The elements of $\mathbf{x}_b$ might be grid-point values, spherical harmonic coefficients, etc., and some elements may represent temperatures, others wind components, etc.
- We refer to $\mathbf{x}_b$ as the background
- Similarly, we generalize the observation to a vector $\mathbf{y}$.
- $\mathbf{y}$ can contain a disparate collection of observations at different locations, and of different variables.

Extension to Multiple Dimensions

- The major difference between the simple scalar example and the multi-dimensional case is that there is no longer a one-to-one correspondence between the elements of the observation vector and those of the background vector.
- It is no longer trivial to compare observations and background.
- Observations are not necessarily located at model gridpoints
- The observed variables (e.g. radiances) may not correspond directly with any of the variables of the model.
- To overcome this problem, we must asume that our model is a more-or-less complete representation of reality, so that we can always determine "model equivalents" of the observations.

Extension to Multiple Dimensions

- We formalize this by assuming the existance of an observation operator, $\mathcal{H}$.
- Given a model-space vector, $\mathbf{x}$, the vector $\mathcal{H}(\mathbf{x})$ can be compared directly with $\mathbf{y}$, and represents the "model equivalent" of $\mathbf{y}$.
- For now, we will assume that $\mathcal{H}$ is perfect. I.e. it does not introduce any error, so that:

$$\mathcal{H}(\mathbf{x}^*) = \mathbf{y}^*$$

where $\mathbf{x}^*$ is the true state, and $\mathbf{y}^*$ contains the true values of the observed quantities.

Extension to Multiple Dimensions

- As we did in the scalar case, we will look for an analysis that is a linear combination of the available information:

$$\mathbf{x}_a = \mathbf{F}\mathbf{x}_b + \mathbf{G}\mathcal{H}(\mathbf{x}_b) + \mathbf{K}\mathbf{y} + \mathbf{c}$$

  where $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{K}$ are matrices, and where $\mathbf{c}$ is a vector.
- If $\mathcal{H}$ is linear, we can proceed as in the scalar case and look for a linear unbiased estimate.
- In the more general case of nonlinear $\mathcal{H}$, we will require that error-free inputs ($\mathbf{x}_b = \mathbf{x}^*$ and $\mathbf{y} = \mathbf{y}^*$) produce an error-free analysis ($\mathbf{x}_a = \mathbf{x}^*$):

$$\mathbf{x}^* = \mathbf{F}\mathbf{x}^* + \mathbf{G}\mathcal{H}(\mathbf{x}^*) + \mathbf{K}\mathcal{H}(\mathbf{x}^*) + \mathbf{c}$$

- Since this applies for any $\mathbf{x}^*$, we must have $\mathbf{c} = 0$ and

$$\mathbf{F} + \mathbf{G}\mathcal{H}(\cdot) \equiv \mathbf{I} - \mathbf{K}\mathcal{H}(\cdot)$$

- Our analysis equation is thus:

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}\left(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)\right)$$

ECMWF

Extension to Multiple Dimensions

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}\left(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)\right)$$

- Remember that in the scalar case, we had

$$\begin{aligned} T_a &= \alpha T_o + (1-\alpha) T_b \\ &= T_b + \alpha(T_o - T_b) \end{aligned}$$

- We see that the matrix $\mathbf{K}$ plays a role equivalent to that of the coefficient α.
- $\mathbf{K}$ is called the gain matrix.
- It determines the weight given to the observations
- It handles the transformation of information defined in "observation space" to the space of model variables.

Extension to Multiple Dimensions

- The next step in deriving the analysis equation is to describe the statistical properties of the analysis errors.
- We define

$$\begin{aligned} \epsilon_a &= \mathbf{x}_a - \mathbf{x}^* \\ \epsilon_b &= \mathbf{x}_b - \mathbf{x}^* \\ \epsilon_o &= \mathbf{y} - \mathbf{y}^* \end{aligned}$$

- We will assume that the errors are small, so that

$$\mathcal{H}(\mathbf{x}_b) = \mathcal{H}(\mathbf{x}^*) + \mathbf{H}\epsilon_b + O(\epsilon_b^2)$$

where $\mathbf{H}$ is the Tangent Linear Operator associated with $\mathcal{H}$.

Extension to Multiple Dimensions

- Substituting the expressions for the errors into our analysis equation, and using $\mathcal{H}(\mathbf{x}^*) = \mathbf{y}^*$, gives (to first order):

$$\epsilon_a = \epsilon_b + \mathbf{K}\left(\epsilon_o - \mathbf{H}\epsilon_b\right)$$

- As in the scalar example, we will assume that the mean errors have been removed, so that $\overline{\epsilon_b} = \overline{\epsilon_o} = 0$. We see that this implies that $\overline{\epsilon_a} = 0$.
- In the scalar example, we derived the variance of the analysis error, and defined our optimal analysis to minimize this variance.
- In the multi-dimensional case, we must deal with covariances.

Covariance

- The covariance between two variables x_i and x_j is defined as

$$\mathrm{cov}(x_i, x_j) = \overline{(x_i - \overline{x_i})(x_j - \overline{x_j})}$$

- Given a vector $\mathbf{x} = (x_1, x_2, \cdots, x_N)^{\mathrm{T}}$, we can arrange the covariances into a covariance matrix, $\mathbf{C}$, such that $C_{ij} = \mathrm{cov}(x_i, x_j)$.
- Equivalently:

$$\mathbf{C} = \overline{(\mathbf{x} - \overline{\mathbf{x}})(\mathbf{x} - \overline{\mathbf{x}})^{\mathrm{T}}}$$

- Covariance matrices are symmetric and positive definite

Extension to Multiple Dimensions

- The analysis error is:

$$\begin{aligned}\epsilon_a &= \epsilon_b + \mathbf{K}(\epsilon_o - \mathbf{H}\epsilon_b) \\ &= (\mathbf{I} - \mathbf{KH})\epsilon_b + \mathbf{K}\epsilon_o\end{aligned}$$

- Forming the analysis error covariance matrix gives:

$$\begin{aligned}\overline{\epsilon_a\epsilon_a^{\mathrm{T}}} &= \overline{[(\mathbf{I} - \mathbf{KH})\epsilon_b + \mathbf{K}\epsilon_o]\,[(\mathbf{I} - \mathbf{KH})\epsilon_b + \mathbf{K}\epsilon_o]^{\mathrm{T}}} \\ &= (\mathbf{I} - \mathbf{KH})\overline{\epsilon_b\epsilon_b^{\mathrm{T}}}(\mathbf{I} - \mathbf{KH})^{\mathrm{T}} + (\mathbf{I} - \mathbf{KH})\overline{\epsilon_b\epsilon_o^{\mathrm{T}}}\mathbf{K}^{\mathrm{T}} \\ &\quad + \mathbf{K}\overline{\epsilon_o\epsilon_b^{\mathrm{T}}}(\mathbf{I} - \mathbf{KH})^{\mathrm{T}} + \mathbf{K}\overline{\epsilon_o\epsilon_o^{\mathrm{T}}}\mathbf{K}^{\mathrm{T}}\end{aligned}$$

- Assuming that the backgound and observation errors are uncorrelated (i.e. $\overline{\epsilon_o\epsilon_b^{\mathrm{T}}} = \overline{\epsilon_b\epsilon_o^{\mathrm{T}}} = 0$), we find:

$$\overline{\epsilon_a\epsilon_a^{\mathrm{T}}} = (\mathbf{I} - \mathbf{KH})\overline{\epsilon_b\epsilon_b^{\mathrm{T}}}(\mathbf{I} - \mathbf{KH})^{\mathrm{T}} + \mathbf{K}\overline{\epsilon_o\epsilon_o^{\mathrm{T}}}\mathbf{K}^{\mathrm{T}}$$

Extension to Multiple Dimensions

$$\overline{\epsilon_a \epsilon_a^{\mathrm{T}}} = (\mathbf{I}-\mathbf{KH})\overline{\epsilon_b \epsilon_b^{\mathrm{T}}}(\mathbf{I}-\mathbf{KH})^{\mathrm{T}} + \mathbf{K}\overline{\epsilon_o \epsilon_o^{\mathrm{T}}}\mathbf{K}^{\mathrm{T}}$$

- This expression is the equivalent of the expression we obtained for the error of the scalar analysis:

$$\overline{\epsilon_a^2} = (1-\alpha)^2\overline{\epsilon_b^2} + \alpha^2\overline{\epsilon_o^2}$$

- Again, we see that $\mathbf{K}$ plays essentially the same role in the multi-dimensional analysis as α plays in the scalar case.
- In the scalar case, we chose α to minimize the variance of the analysis error.
- What do we mean by the minimum-variance analysis in the multi-dimensional case?

Extension to Multiple Dimensions

- Note that the diagonal elements of a covariance matrix are variances $C_{ii} = \mathrm{cov}(x_i, x_i) = \overline{(x_i - \overline{x_i})^2}$.
- Hence, we can define the minimum-variance analysis as the analysis that minimizes the sum of the diagonal elements of the analysis error covariance matrix.
- The sum of the diagonal elements of a matrix is called the trace.
- In the scalar case, we found the minimum-variance analysis by setting $\frac{d\overline{\epsilon_a^2}}{d\alpha}$ to zero.
- In the multidimensional case, we are going to set

$$\frac{\partial \mathrm{trace}(\overline{\epsilon_a \epsilon_a^{\mathrm{T}}})}{\partial \mathbf{K}} = \mathbf{0}$$

- (Note: $\frac{\partial \mathrm{trace}(\overline{\epsilon_a \epsilon_a^{\mathrm{T}}})}{\partial \mathbf{K}}$ is the matrix whose ij^{th} element is $\frac{\partial \mathrm{trace}(\overline{\epsilon_a \epsilon_a^{\mathrm{T}}})}{\partial K_{ij}}$.)

Extension to Multiple Dimensions

- We have: $\overline{\epsilon_a \epsilon_a^{\mathrm{T}}} = (\mathbf{I}-\mathbf{KH})\overline{\epsilon_b \epsilon_b^{\mathrm{T}}}(\mathbf{I}-\mathbf{KH})^{\mathrm{T}} + \mathbf{K}\overline{\epsilon_o \epsilon_o^{\mathrm{T}}}\mathbf{K}^{\mathrm{T}}$.
- The following matrix identities come to our rescue:

$$\frac{\partial \mathrm{trace}(\mathbf{KAK}^{\mathrm{T}})}{\partial \mathbf{K}} = \mathbf{K}(\mathbf{A}+\mathbf{A}^{\mathrm{T}})$$

$$\frac{\partial \mathrm{trace}(\mathbf{KA})}{\partial \mathbf{K}} = \mathbf{A}^{\mathrm{T}} \qquad \frac{\partial \mathrm{trace}(\mathbf{AK}^{\mathrm{T}})}{\partial \mathbf{K}} = \mathbf{A}$$

- Applying these to $\partial \mathrm{trace}(\overline{\epsilon_a \epsilon_a^{\mathrm{T}}})/\partial \mathbf{K}$ gives:

$$\frac{\partial \mathrm{trace}(\overline{\epsilon_a \epsilon_a^{\mathrm{T}}})}{\partial \mathbf{K}} = 2\mathbf{K}\left[\mathbf{H}\overline{\epsilon_b \epsilon_b^{\mathrm{T}}}\mathbf{H}^{\mathrm{T}} + \overline{\epsilon_o \epsilon_o^{\mathrm{T}}}\right] - 2\overline{\epsilon_b \epsilon_b^{\mathrm{T}}}\mathbf{H}^{\mathrm{T}} = \mathbf{0}$$

- Hence: $\mathbf{K} = \overline{\epsilon_b \epsilon_b^{\mathrm{T}}}\mathbf{H}^{\mathrm{T}}\left[\mathbf{H}\overline{\epsilon_b \epsilon_b^{\mathrm{T}}}\mathbf{H}^{\mathrm{T}} + \overline{\epsilon_o \epsilon_o^{\mathrm{T}}}\right]^{-1}$.

Extension to Multiple Dimensions

$$\mathbf{K} = \overline{\epsilon_b \epsilon_b^{\mathrm{T}}} \mathbf{H}^{\mathrm{T}} \left[\mathbf{H} \overline{\epsilon_b \epsilon_b^{\mathrm{T}}} \mathbf{H}^{\mathrm{T}} + \overline{\epsilon_o \epsilon_o^{\mathrm{T}}} \right]^{-1}$$

- This optimal gain matrix is called the Kalman Gain Matrix.
- Note the similarity with the optimal gain we derived for the scalar analysis: $\alpha = \overline{\epsilon_b^2}/(\overline{\epsilon_b^2} + \overline{\epsilon_o^2})$.
- The variance of analysis error for the optimal scalar problem was:

$$\overline{\epsilon_a^2} = \left(\frac{1}{\overline{\epsilon_b^2}} + \frac{1}{\overline{\epsilon_o^2}} \right)^{-1}$$

- The equivalent expression for the multi-dimensional case is:

$$\overline{\epsilon_a \epsilon_a^{\mathrm{T}}} = \left[\left(\overline{\epsilon_b \epsilon_b^{\mathrm{T}}} \right)^{-1} + \mathbf{H}^{\mathrm{T}} \left(\overline{\epsilon_o \epsilon_o^{\mathrm{T}}} \right)^{-1} \mathbf{H} \right]^{-1}$$

Notation

- The notation we have used for covariance matrices can get a bit cumbersome.
- The standard notation is:

$$\begin{aligned} \mathbf{P}^a &\equiv \overline{\epsilon_a \epsilon_a^{\mathrm{T}}} \\ \mathbf{P}^b &\equiv \overline{\epsilon_b \epsilon_b^{\mathrm{T}}} \\ \mathbf{R} &\equiv \overline{\epsilon_o \epsilon_o^{\mathrm{T}}} \end{aligned}$$

- In many analysis schemes, the true covariance matrix of background error, $\mathbf{P}^b$, is not known, or is too large to be used.
- In this case, we use an approximate background error covariance matrix. This approximate matrix is denoted by $\mathbf{B}$.

Alternative Expression for the Kalman Gain

Finally, we derive an alternative expression for the Kalman gain:

$$\mathbf{K} = \mathbf{P}^b\mathbf{H}^{\mathrm{T}}\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]^{-1}$$

Multiplying both sides by $\left[(\mathbf{P}^b)^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right]$ gives:

$$\begin{aligned}\left[(\mathbf{P}^b)^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right]\mathbf{K} &= \left[\mathbf{H}^{\mathrm{T}} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right]\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]^{-1} \\ &= \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left[\mathbf{R} + \mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right]\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]^{-1} \\ &= \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\end{aligned}$$

Hence:

$$\mathbf{K} = \left[(\mathbf{P}^b)^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}$$

Optimal Interpolation

- Optimal Interpolation is a statistical data assimilation method based on the multi-dimensional analysis equations we have just derived.
- The method was used operationally at ECMWF from 1979 until 1996, when it was replaced by 3D-Var.
- The basic idea is to split the global analysis into a number of boxes which can be analysed independently:

$$\mathbf{x}_a^{(i)} = \mathbf{x}_b^{(i)} + \mathbf{K}^{(i)} \left(\mathbf{y}^{(i)} - \mathcal{H}^{(i)}(\mathbf{x}_b) \right)$$

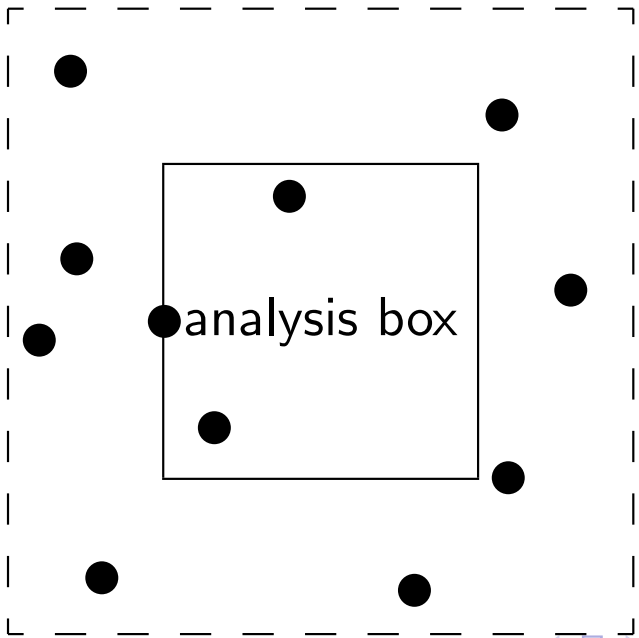
where

$$\mathbf{x}_a = \begin{pmatrix} \mathbf{x}_a^{(1)} \\ \mathbf{x}_a^{(2)} \\ \vdots \\ \mathbf{x}_a^{(M)} \end{pmatrix} \qquad \mathbf{x}_b = \begin{pmatrix} \mathbf{x}_b^{(1)} \\ \mathbf{x}_b^{(2)} \\ \vdots \\ \mathbf{x}_b^{(M)} \end{pmatrix} \qquad \mathbf{K} = \begin{pmatrix} \mathbf{K}^{(1)} \\ \mathbf{K}^{(2)} \\ \vdots \\ \mathbf{K}^{(M)} \end{pmatrix}$$

Optimal Interpolation

$$\mathbf{x}_a^{(i)} = \mathbf{x}_b^{(i)} + \mathbf{K}^{(i)} \left(\mathbf{y}^{(i)} - \mathcal{H}^{(i)}(\mathbf{x}_b) \right)$$

- In principle, we should use *all* available observations to calculate the anaysis for each box. However, this is too expensive.
- To produce a computationally-feasible algorithm, Optimal Interpolation (OI) restricts the observations used for each box to those observations which lie in a surrounding selection area:



Optimal Interpolation

- The gain matrix used for each box is:

$$\mathbf{K}^{(i)} = \left(\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right)^{(i)} \left[\left(\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right)^{(i)} + \mathbf{R}^{(i)}\right]^{-1}$$

- Now, the dimension of the matrix $\left[\left(\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right)^{(i)} + \mathbf{R}^{(i)}\right]$ is equal to the number of observtions in the selection box.
- Selecting observations reduces the size of this matrix, making it feasible to use direct solution methods to invert it.
- Note that to implement Optimal Interpolation, we have to specify $\left(\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right)^{(i)}$ and $\left(\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}}\right)^{(i)}$. This effectively limits us to very simple observation operators, corresponding to simple interpolations.
- This, together with the artifacts introduced by observation selection, was one of the main reasons for abandoning Optimal Interpolation in favour of 3D-Var.

From Optimal Interpolation to 3D-Var

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{K}\left(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)\right) \qquad \text{where} \qquad \mathbf{K} = \mathbf{P}^b\mathbf{H}^{\mathrm{T}}\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]^{-1}$$

- Optimal Interpolation (OI) applies direct solution methods to invert the matrix $\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]$.
- Data selection is applied to reduce the dimension of the matrix.
- Direct methods require access to the matrix elements. In particular, $\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}}$ must be available in matrix form.
- This limits us to very simple observation operators.

From Optimal Interpolation to 3D-Var

- Iterative methods have significant advantages over the direct methods used in OI.
- They can be applied to much larger problems than direct techniques, so we can avoid data selection.
- They do not require access to the matrix elements.
- Typically, to solve $\mathbf{Ax} = \mathbf{b}$, requires only the ability to calculate matrix-vector products: $\mathbf{Ax}$.
- This allows us to use operators defined by pieces of code rather than explicitly as matrices.
- Examples of such operators include radiative-transfer codes, numerical models, Fourier transforms, etc.

Example: Conjugate Gradients

To solve $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ is real, symmetric and positive-definite:

$$\mathbf{r}_0 := \mathbf{b} - \mathbf{A}\mathbf{x}_0 \qquad \mathbf{p}_0 := \mathbf{r}_0 \qquad k := 0$$

repeat until $\mathbf{r}_{k+1}$ is sufficiently small

$$\begin{aligned}
\alpha_k &:= \frac{\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k}{\mathbf{p}_k^{\mathrm{T}}\mathbf{A}\mathbf{p}_k} \\
\mathbf{x}_{k+1} &:= \mathbf{x}_k + \alpha_k \mathbf{p}_k \\
\mathbf{r}_{k+1} &:= \mathbf{r}_k - \alpha_k \mathbf{A}\mathbf{p}_k \\
\beta_k &:= \frac{\mathbf{r}_{k+1}^{\mathrm{T}}\mathbf{r}_{k+1}}{\mathbf{r}_k^{\mathrm{T}}\mathbf{r}_k} \\
\mathbf{p}_{k+1} &:= \mathbf{r}_{k+1} + \beta_k \mathbf{p}_k \\
k &:= k+1
\end{aligned}$$

The result is x_{k+1}

From Optimal Interpolation to 3D-Var

- There are two ways to apply iterative methods to the linear analysis equation, depending which expression we adopt for $\mathbf{K}$:
- For $\mathbf{K} = \mathbf{P}^b\mathbf{H}^{\mathrm{T}}\left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]^{-1}$ we have:

$$\mathbf{x}_a = \mathbf{x}_b + \mathbf{P}^b\mathbf{H}^{\mathrm{T}}\mathbf{z} \quad \text{where} \quad \left[\mathbf{H}\mathbf{P}^b\mathbf{H}^{\mathrm{T}} + \mathbf{R}\right]\mathbf{z} = \mathbf{y} - \mathcal{H}(\mathbf{x}_b)$$

- For $\mathbf{K} = \left[(\mathbf{P}^b)^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right]^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}$, we have:

$$\mathbf{x}_a = \mathbf{x}_b + \delta\mathbf{x} \quad \text{where} \quad \left[(\mathbf{P}^b)^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right]\delta\mathbf{x} = \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y} - \mathcal{H}(\mathbf{x}_b)\right)$$

- The first of these alternatives is called PSAS
- The second (although it may not look like it yet) is 3D-Var

3D-Var

- As we have seen, (linear) 3D-Var analysis can be seen as an application of iterative solution methods to the linear analysis equation.
- Historically, 3D-Var was not developed this way.
- We will now consider this alternative derivation.
- We will need to apply Bayes' theorem:

$$p(A|B) = \frac{p(B|A)p(A)}{p(B)}$$

where $p(A|B)$ is the probability of A given B, etc.

Maximum Likelihood

- We developed the linear analysis equation by searching for a linear combination of observation and background that minimized the variance of the error.
- An alternative approach is to look for the most probable solution, given the background and observations:

$$\mathbf{x}_a = \arg\max_{\mathbf{x}} \left(p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}_b) \right)$$

- It will be convenient to define a cost function

$$J = -\log \left(p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}_b) \right) + const.$$

- Then, since log is a monotonic function:

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} \left(J(\mathbf{x}) \right)$$

Maximum Likelihood

- Applying Bayes' theorem gives:

$$p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}_b) = \frac{p(\mathbf{y} \text{ and } \mathbf{x}_b|\mathbf{x})p(\mathbf{x})}{p(\mathbf{y} \text{ and } \mathbf{x}_b)}$$

- Now, $p(\mathbf{y} \text{ and } \mathbf{x}_b)$ is independent of $\mathbf{x}$.
- *A Priori* we know nothing about $\mathbf{x}$ – all values of $\mathbf{x}$ are equally likely.
- Hence, we can regard $p(\mathbf{x})/p(\mathbf{y} \text{ and } \mathbf{x}_b)$ as independent of $\mathbf{x}$, and write:

$$p(\mathbf{x}|\mathbf{y} \text{ and } \mathbf{x}_b) \propto p(\mathbf{y} \text{ and } \mathbf{x}_b|\mathbf{x})$$

- Furthermore, if observation errors and backgound errors are uncorrelated, then

$$p(\mathbf{y} \text{ and } \mathbf{x}_b|\mathbf{x}) = p(\mathbf{y}|\mathbf{x})p(\mathbf{x}_b|\mathbf{x})$$

$$\Rightarrow \qquad J(\mathbf{x}) = -\log\left(p(\mathbf{y}|\mathbf{x})\right) - \log\left(p(\mathbf{x}_b|\mathbf{x})\right) + const.$$

Maximum Likelihood

- The maximum likelihood approach is applicable to any probability density functions $p(\mathbf{y}|\mathbf{x})$ and $p(\mathbf{x}_b|\mathbf{x})$.
- However, let us consider the special case of Gaussian p.d.f's:

$$p(\mathbf{x}_b|\mathbf{x}) = \frac{1}{(2\pi)^{N/2}|\mathbf{P}_b|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{x}_b-\mathbf{x})^{\mathrm{T}}(\mathbf{P}_b)^{-1}(\mathbf{x}_b-\mathbf{x})\right]$$

$$p(\mathbf{y}|\mathbf{x}) = \frac{1}{(2\pi)^{M/2}|\mathbf{R}|^{1/2}} \exp\left[-\frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{x}))\right]$$

- Now, $J(\mathbf{x}) = -\log(p(\mathbf{y}|\mathbf{x})) - \log(p(\mathbf{x}_b|\mathbf{x})) + const.$
- Hence, with an appropriate choice of the constant *const.*:

$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_b-\mathbf{x})^{\mathrm{T}}(\mathbf{P}_b)^{-1}(\mathbf{x}_b-\mathbf{x}) + \frac{1}{2}(\mathbf{y}-\mathcal{H}(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}-\mathcal{H}(\mathbf{x}))$$

- This is the 3D-Var cost function

ECMWF

Maximum Likelihood

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}_b - \mathbf{x}\right)^{\mathrm{T}}\left(\mathbf{P}_b\right)^{-1}\left(\mathbf{x}_b - \mathbf{x}\right) + \frac{1}{2}\left(\mathbf{y} - \mathcal{H}(\mathbf{x})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y} - \mathcal{H}(\mathbf{x})\right)$$

- The maximum likelihood analysis corresponds to the global minimum of the cost function
- At the minimum, the gradient of the cost function ($\nabla J(\mathbf{x})$ or $\partial J/\partial \mathbf{x}$) is zero:

$$\nabla J(\mathbf{x}) = \left(\mathbf{P}_b\right)^{-1}\left(\mathbf{x} - \mathbf{x}_b\right) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathcal{H}(\mathbf{x}) - \mathbf{y}\right) = \mathbf{0}$$

Maximum Likelihood

$$\nabla J(\mathbf{x}) = (\mathbf{P}_b)^{-1}(\mathbf{x} - \mathbf{x}_b) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathcal{H}(\mathbf{x}) - \mathbf{y}) = \mathbf{0}$$

- Now, if $\mathcal{H}$ is linear (or if we neglect second-order terms) then

$$\mathcal{H}(\mathbf{x}) = \mathcal{H}(\mathbf{x}_b) + \mathbf{H}(\mathbf{x} - \mathbf{x}_b)$$

- Hence: $(\mathbf{P}_b)^{-1}(\mathbf{x} - \mathbf{x}_b) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathcal{H}(\mathbf{x}_b) + \mathbf{H}(\mathbf{x} - \mathbf{x}_b)) - \mathbf{y}) = \mathbf{0}$
- Rearranging a little gives:

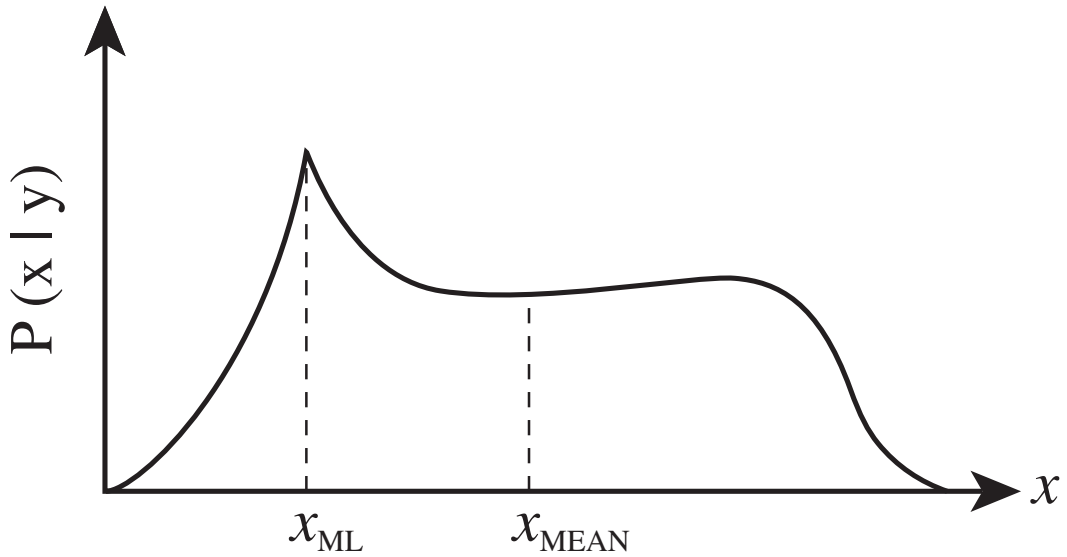
$$\left[(\mathbf{P}_b)^{-1} + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}\mathbf{H}\right]\delta\mathbf{x} = \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}_b))$$

  where $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}_b$

- This is exactly the equation for the minimum-variance analysis we derived earlier!

Maximum Likelihood

- We have shown that the maximum likelihood approach is naturally expressed in terms of a cost function representing minus the log of the probability of the analysis state.
- The minimum of the cost function corresponds to the maximum likelihood (probability) solution.
- For Gaussian errors and linear observation operators, the maximum likelihood analysis coincides with the minimum variance solution.
- This is not the case in general:



Maximum Likelihood

- In the nonlinear case, the minimum variance approach is difficult to apply.
- The maximum-likelihood approach is much more generally applicable
- As long as we know the p.d.f's, we can define the cost function
  - However, finding the global minimum may not be easy for highly non-Gaussian p.d.f's.
- In practice, background errors are usually assumed to be Gaussian (or a nonlinear transformation is applied to *make* them Gaussian).
  - This makes the background-error term of the cost function quadratic.
- However, non-Gaussian observation errors are taken into account. For example:
  - Directionally-ambiguous wind observations from scatterometers
  - Observations contaminated by occasional gross errors, which make outliers much more likely than implied by a Gaussian model.

Minimization

- In 3D-Var, the analysis is found by minimizing the cost function:

$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}_b - \mathbf{x}\right)^{\mathrm{T}}\left(\mathbf{P}_b\right)^{-1}\left(\mathbf{x}_b - \mathbf{x}\right) + \frac{1}{2}\left(\mathbf{y} - \mathcal{H}(\mathbf{x})\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{y} - \mathcal{H}(\mathbf{x})\right)$$

- This is a very large-scale ($\dim(\mathbf{x}) \approx 10^8$) minimization problem.
- The size of the problem restricts on the algorithms we can use.
- Derivative-free algorithms (which require only the ability to calculate $J(\mathbf{x})$ for arbitrary $\mathbf{x}$) are too slow.
- This is because each function evaluation gives very limited information about the shape of the cost function.
  - E.g. a finite difference, $J(\mathbf{x}+\delta\mathbf{v}) - J(\mathbf{x}) \approx \delta\mathbf{v}^{\mathrm{T}}\nabla J(\mathbf{x})$, gives a single component of the gradient.
  - We need $O(10^8)$ components to work out which direction is "downhill".

Calculating the Gradient

- To minimize the cost function, we must be able to calculate gradients.
- The gradient (with respect to $\mathbf{x}$) is:

$$\nabla J(\mathbf{x}) = (\mathbf{P}_b)^{-1}(\mathbf{x} - \mathbf{x}_b) + \mathbf{H}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathcal{H}(\mathbf{x}))$$

- Typically, $\mathbf{R}$ is diagonal — observation errors are treated as being mutually uncorrelated.
- $\mathbf{P}_b$ can be eliminated by a change of variable: $\chi = (\mathbf{P}_b)^{-1/2}(\mathbf{x} - \mathbf{x}_b)$.
- However, the matrix $\mathbf{H}^{\mathrm{T}}$ cannot be eliminated, and is much too large to be represented explicitly.
- Instead, we represent $\mathbf{H}^{\mathrm{T}}$ as an adjoint operator.

Strong Constraint 4D-Var

- So far, we have tacitly assumed that the observations, analysis and background are all valid at the same time, so that $\mathcal{H}$ includes spatial, but not temporal, interpolation.
- In 4D-Var, we relax this assumption.
- Let's use $\mathcal{G}$ to denote a generalised observation operator that:
  - Propagates model fields defined at some time t_0 to the (various) times at which the observations were taken.
  - Spatially interpolates these propagated fields
  - Converts model variables to observed quantities
- We will use a numerical forecast model to perform the first step.
- Note that, since models integrate forward in time and we do not have an inverse of the forecast model, the observations must be available for times $t_k \geq t_0$.

Strong Constraint 4D-Var

- Formally, the 4D-Var cost function is identical to the 3D-Var cost function — we simply replace $\mathcal{H}$ by $\mathcal{G}$:

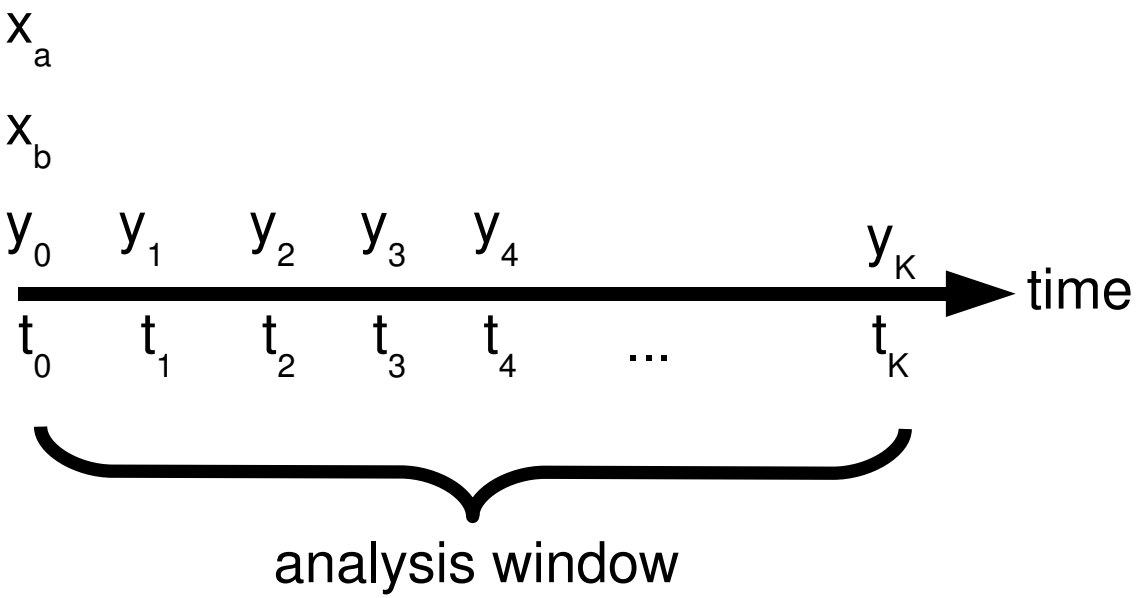
$$J(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_b - \mathbf{x})^{\mathrm{T}}(\mathbf{P}_b)^{-1}(\mathbf{x}_b - \mathbf{x}) + \frac{1}{2}(\mathbf{y} - \mathcal{G}(\mathbf{x}))^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathcal{G}(\mathbf{x}))$$

- However, it makes sense to group observations into sub-vectors of observations, $\mathbf{y}_k$, that are valid at the same time, t_k.
- It is reasonable to assume that observation errors are uncorrelated in time. Then, $\mathbf{R}$ is block diagonal, with blocks $\mathbf{R}_k$ corresponding to the sub-vectors $\mathbf{y}_k$.
- Write $\mathcal{G}_k$ for the generalised observation operator that produces the model equivalents of $\mathbf{y}_k$. Then:

$$\begin{aligned} J(\mathbf{x}) \quad = \quad & \frac{1}{2}(\mathbf{x}_b - \mathbf{x})^{\mathrm{T}}(\mathbf{P}_b)^{-1}(\mathbf{x}_b - \mathbf{x}) \\ & + \frac{1}{2}\sum_{k=0}^{K}(\mathbf{y}_k - \mathcal{G}_k(\mathbf{x}))^{\mathrm{T}}\mathbf{R}_k^{-1}(\mathbf{y}_k - \mathcal{G}_k(\mathbf{x})) \end{aligned}$$

Strong Constraint 4D-Var

$$
\begin{aligned}
J(\mathbf{x}) \quad = \quad & \frac{1}{2}\left(\mathbf{x}_b-\mathbf{x}\right)^{\mathrm{T}}\left(\mathbf{P}_b\right)^{-1}\left(\mathbf{x}_b-\mathbf{x}\right) \\
& +\frac{1}{2}\sum_{k=0}^{K}\left(\mathbf{y}_k-\mathcal{G}_k(\mathbf{x})\right)^{\mathrm{T}}\mathbf{R}_k^{-1}\left(\mathbf{y}_k-\mathcal{G}_k(\mathbf{x})\right)
\end{aligned}
$$



Strong Constraint 4D-Var

- Now, each generalised observation operator can be written as

$$\mathcal{G}_k = \mathcal{H}_k \mathcal{M}_{t_0 \to t_k}$$

  where:
  - $\mathcal{M}_{t_0 \to t_k}$ represents an integration of the forecast model from time t_0 to time t_k.
  - $\mathcal{H}_k$ represents a spatial interpolation and transformation from model variables to observed variables — i.e. a 3D-Var-style observation operator.
- The model integration can be factorised into a sequence of shorter integrations:

$$\mathcal{M}_{t_0 \to t_k} = \mathcal{M}_{t_{k-1} \to t_k} \mathcal{M}_{t_{k-2} \to t_{k-1}} \cdots \mathcal{M}_{t_1 \to t_2} \mathcal{M}_{t_0 \to t_1}$$

Strong Constraint 4D-Var

- Let us introduce model states $\mathbf{x}_k$, which are defined at times t_k.
  - We will also denote the state at the start of the window as $\mathbf{x}_0$ (rather than $\mathbf{x}$, as we have done until now).

$$\begin{aligned}\mathbf{x}_k &= \mathcal{M}_{t_0 \rightarrow t_k}(\mathbf{x}_0) \\ &= \mathcal{M}_{t_{k-1} \rightarrow t_k}(\mathbf{x}_{k-1})\end{aligned}$$

- Then, we can write the cost function as:

$$\begin{aligned}J(\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_k) = & \frac{1}{2}(\mathbf{x}_b - \mathbf{x}_0)^{\mathrm{T}} (\mathbf{P}_b)^{-1} (\mathbf{x}_b - \mathbf{x}_0) \\ & + \frac{1}{2} \sum_{k=0}^{K} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))\end{aligned}$$

Strong Constraint 4D-Var

- Note that, by introducing the vectors $\mathbf{x}_k$, we have converted an unconstrained minimization problem:

$$\mathbf{x}_a = \arg\min_{\mathbf{x}} \left(J(\mathbf{x}_0) \right)$$

into a problem with strong constraints:

$$\begin{aligned} \mathbf{x}_a &= \arg\min_{\mathbf{x}_0} \left(J(\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_k) \right) \\ \text{where} \quad \mathbf{x}_k &= \mathcal{M}_{t_{k-1} \to t_k} \left(\mathbf{x}_{k-1} \right) \quad \text{for } k = 1, 2, \cdots, K \end{aligned}$$

- For this reason, this form of 4D-Var is called strong constraint 4D-Var.

Strong Constraint 4D-Var

- When we derived the 3D-Var cost function, we assumed that the observation operator was perfect: $\mathbf{y}^* = \mathcal{H}(\mathbf{x}^*)$.
- In deriving strong constraint 4D-Var, we have not removed this assumption.
- The generalised observation operators, $\mathcal{G}_k$, are assumed to be perfect.
- In particular, since $\mathcal{G}_k = \mathcal{H}_k \mathcal{M}_{t_0 \rightarrow t_k}$, this implies that the model is perfect:

$$\mathbf{x}_k^* = \mathcal{M}_{t_{k-1} \rightarrow t_k} \left(\mathbf{x}_{k-1}^* \right).$$

- This is called the perfect model assumption.

Strong Constraint 4D-Var

$$J(\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_k) = \frac{1}{2}(\mathbf{x}_b - \mathbf{x}_0)^{\mathrm{T}} (\mathbf{P}_b)^{-1} (\mathbf{x}_b - \mathbf{x}_0) + \frac{1}{2}\sum_{k=0}^{K} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))$$

- When written in this form, it is clear that 4D-Var determines the analysis state at every gridpoint *and at every time within the analysis window*.
- I.e., 4D-Var determines a four-dimensional analysis of the available asynoptic data.
- As a consequence of the perfect model assumption, the analysis corresponds to a trajectory (i.e. an integration) of the forecast model.

Weak Constraint 4D-Var

- The perfect model assumption limits the length of analysis window that can be used to roughly 12 hours (for an NWP system).
- To use longer analysis windows (or to account for deficiencies of the model that are already apparent with a 12-hour window) we must relax the perfect model assumption.
- We saw already that strong constraint 4D-Var can be expressed as:

$$\begin{aligned} \mathbf{x}_a &= \arg\min_{\mathbf{x}_0} \left(J(\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_k) \right) \\ \text{subject to} \quad \mathbf{x}_k &= \mathcal{M}_{t_{k-1} \rightarrow t_k} \left(\mathbf{x}_{k-1} \right) \quad \text{for } k = 1, 2, \cdots, K \end{aligned}$$

- In weak constraint 4D-Var, we define the model error as

$$\eta_k = \mathbf{x}_k - \mathcal{M}_{t_{k-1} \rightarrow t_k} \left(\mathbf{x}_{k-1} \right) \quad \text{for } k = 1, 2, \cdots, K$$

and we allow η_k to be non-zero.

Weak Constraint 4D-Var

- We can derive the weak constraint cost function using Bayes' rule:

$$p(\mathbf{x}_0 \cdots \mathbf{x}_K | \mathbf{x}_b; \mathbf{y}_0 \cdots \mathbf{y}_K) = \frac{p(\mathbf{x}_b; \mathbf{y}_0 \cdots \mathbf{y}_K | \mathbf{x}_0 \cdots \mathbf{x}_K) p(\mathbf{x}_0 \cdots \mathbf{x}_K)}{p(\mathbf{x}_b; \mathbf{y}_0 \cdots \mathbf{y}_K)}$$

- The denominator is independent of $\mathbf{x}_0 \cdots \mathbf{x}_K$.
- The term $p(\mathbf{x}_b; \mathbf{y}_0 \cdots \mathbf{y}_K | \mathbf{x}_0 \cdots \mathbf{x}_K)$ simplifies to:

$$p(\mathbf{x}_b | \mathbf{x}_0) \prod_{k=0}^{K} p(\mathbf{y}_k | \mathbf{x}_k)$$

- Hence

$$p(\mathbf{x}_0 \cdots \mathbf{x}_K | \mathbf{x}_b; \mathbf{y}_0 \cdots \mathbf{y}_K) \propto p(\mathbf{x}_b | \mathbf{x}_0) \left[\prod_{k=0}^{K} p(\mathbf{y}_k | \mathbf{x}_k) \right] p(\mathbf{x}_0 \cdots \mathbf{x}_K)$$

Weak Constraint 4D-Var

$$p(\mathbf{x}_0 \cdots \mathbf{x}_K | \mathbf{x}_b; \mathbf{y}_0 \cdots \mathbf{y}_K) \propto p(\mathbf{x}_b|\mathbf{x}_0) \left[\prod_{k=0}^{K} p(\mathbf{y}_k|\mathbf{x}_k) \right] p(\mathbf{x}_0 \cdots \mathbf{x}_K)$$

- Taking minus the logarithm gives the cost function:

$$J(\mathbf{x}_0 \cdots \mathbf{x}_K) = -\log\left(p(\mathbf{x}_b|\mathbf{x}_0)\right) - \sum_{k=0}^{K} \log\left(p(\mathbf{y}_k|\mathbf{x}_k)\right) - \log\left(p(\mathbf{x}_0 \cdots \mathbf{x}_K)\right)$$

- The terms involving $\mathbf{x}_b$ and $\mathbf{y}_k$ are familiar. They are the background and observation terms of the strong constraint cost function.
- The final term is new. It represents the *a priori* probability of the sequence of states $\mathbf{x}_0 \cdots \mathbf{x}_K$.

Weak Constraint 4D-Var

- Given the sequence of states $\mathbf{x}_0 \cdots \mathbf{x}_K$, we can calculate the corresponding model errors:

$$\eta_k = \mathbf{x}_k - \mathcal{M}_{t_{k-1} \rightarrow t_k}(\mathbf{x}_{k-1}) \quad \text{for } k = 1, 2, \cdots, K$$

- We can use our knowledge of the statistics of model error to define

$$p(\mathbf{x}_0 \cdots \mathbf{x}_K) \equiv p(\mathbf{x}_0; \eta_1 \cdots \eta_K)$$

- One possibility is to assume that model error is uncorrelated in time. In this case:

$$p(\mathbf{x}_0 \cdots \mathbf{x}_K) \equiv p(\mathbf{x}_0) p(\eta_1) \cdots p(\eta_K)$$

- If we take $p(\mathbf{x}_0) = const.$ (all states equally likely), and $p(\eta_k)$ as Gaussian with covariance matrix $\mathbf{Q}_k$, we see that weak constraint 4D-Var adds the following term to the cost function:

$$\frac{1}{2} \sum_{K=1}^{K} \eta_k^{\mathrm{T}} \mathbf{Q}_k^{-1} \eta_k$$

ECMWF

Weak Constraint 4D-Var

- Hence, for Gaussian, temporally-uncorrelated model error, the weak constraint cost function is:

$$
\begin{aligned}
J(\mathbf{x}_0, \mathbf{x}_1, \cdots \mathbf{x}_k) \quad = \quad & \frac{1}{2}(\mathbf{x}_b - \mathbf{x}_0)^{\mathrm{T}} (\mathbf{P}_b)^{-1} (\mathbf{x}_b - \mathbf{x}_0) \\
& + \frac{1}{2} \sum_{k=0}^{K} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k)) \\
& + \frac{1}{2} \sum_{K=1}^{K} \eta_k^{\mathrm{T}} \mathbf{Q}_k^{-1} \eta_k
\end{aligned}
$$

where $\eta_k = \mathbf{x}_k - \mathcal{M}_{t_{k-1} \to t_k}(\mathbf{x}_{k-1})$.

The Extended Kalman Filter

- Earlier in my lecture, I derived the linear analysis equation:

$$\mathbf{x}_k^a = \mathbf{x}_k^b + \mathbf{K}_k \left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^b) \right)$$

  - NB: I have added a subscript k to show that the analysis, background, observations, etc. are all valid for some time t_k.

- I showed that the optimal choice for $\mathbf{K}_k$ is the Kalman Gain Matrix:

$$\mathbf{K}_k = \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} \left[\mathbf{H}_k \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k \right]^{-1} \equiv \left[(\mathbf{P}_k^b)^{-1} + \mathbf{H}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \mathbf{H}_k \right]^{-1} \mathbf{H}_k^{\mathrm{T}} \mathbf{R}_k^{-1}$$

- I gave the following expression for the covariance matrix of analysis error:

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^b (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}}$$

- Now we will consider how to generate $\mathbf{P}_k^b$ in an optimal way.

The Extended Kalman Filter

- In most applications of data assimilation, we do not just want to produce a single analysis for one given time.
- Rather, we are interested in a sequence of analyses for times $t_0, t_1, \cdots$, etc.
- For each analysis in this sequence, we require background $\mathbf{x}_k^b$ (i.e. a prior estimate of the state at time t_k).
- Our best prior estimate of the state at time t_k is given by a forecast from the preceding analysis:

$$\mathbf{x}_k^b = \mathcal{M}_{t_{k-1} \to t_k} \left(\mathbf{x}_{k-1}^a \right)$$

- What is the error covariance matrix associated with this background?

The Extended Kalman Filter

$$\mathbf{x}_k^b = \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}^a\right)$$

- Subtract the true state at time t_k from both sides:

$$\epsilon_k^b = \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}^a\right) - \mathbf{x}_k^*$$

- Now write $\mathbf{x}_{k-1}^a = \mathbf{x}_{k-1}^* + \epsilon_{k-1}^a$ and assume that ϵ_{k-1}^a is small enough for the following linear approximation to be valid:

$$\mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}^a\right) \approx \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}^*\right) + \mathbf{M}_{t_{k-1}\to t_k}\epsilon_{k-1}^a$$

- Then:

$$\begin{aligned}\epsilon_k^b &= \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}^*\right) + \mathbf{M}_{t_{k-1}\to t_k}\epsilon_{k-1}^a - \mathbf{x}_k^* \\ &= \mathbf{M}_{t_{k-1}\to t_k}\epsilon_{k-1}^a + \eta_k\end{aligned}$$

where $\eta_k = \mathcal{M}_{t_{k-1}\to t_k}\left(\mathbf{x}_{k-1}^*\right) - \mathbf{x}_k^*$ is the model error.

The Extended Kalman Filter

$$\eta_k = \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}^*_{k-1}\right) - \mathbf{x}^*_k$$

- We will assume that $\overline{\epsilon^a_{k-1}} = \overline{\eta_k} = 0 \quad \Rightarrow \quad \overline{\epsilon^b_k} = 0.$
- The covariance matrix of background error is:

$$\overline{\epsilon^b_k(\epsilon^b_k)^{\mathrm{T}}} = \overline{\left(\mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon^a_{k-1} + \eta_k\right)\left(\mathbf{M}_{t_{k-1}\rightarrow t_k}\epsilon^a_{k-1} + \eta_k\right)^{\mathrm{T}}}$$

- Assuming that analysis error and model error are uncorrelated, we can multiply this out to get:

$$\overline{\epsilon^b_k(\epsilon^b_k)^{\mathrm{T}}} = \mathbf{M}_{t_{k-1}\rightarrow t_k}\overline{\epsilon^a_{k-1}(\epsilon^a_{k-1})^{\mathrm{T}}}\mathbf{M}^{\mathrm{T}}_{t_{k-1}\rightarrow t_k} + \overline{\eta_k\eta_k^{\mathrm{T}}}$$

- I.e.

$$\mathbf{P}^b_k = \mathbf{M}_{t_{k-1}\rightarrow t_k}\mathbf{P}^a_{k-1}\mathbf{M}^{\mathrm{T}}_{t_{k-1}\rightarrow t_k} + \mathbf{Q}_k$$

where $\mathbf{Q}_k = \overline{\eta_k\eta_k^{\mathrm{T}}}$ is the covariance matrix of model error.

ECMWF

The Extended Kalman Filter

- We now have all the equations we need to analyse and propagate the state, and to compute and propagate the covariances:

$$
\begin{aligned}
\mathbf{x}_k^b &= \mathcal{M}_{t_{k-1}\rightarrow t_k}\left(\mathbf{x}_{k-1}^a\right) \\
\mathbf{P}_k^b &= \mathbf{M}_{t_{k-1}\rightarrow t_k}\mathbf{P}_{k-1}^a\mathbf{M}_{t_{k-1}\rightarrow t_k}^{\mathrm{T}} + \mathbf{Q}_k \\
\mathbf{K}_k &= \mathbf{P}_k^b\mathbf{H}_k^{\mathrm{T}}\left[\mathbf{H}_k\mathbf{P}_k^b\mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right]^{-1} \\
\mathbf{x}_k^a &= \mathbf{x}_k^b + \mathbf{K}_k\left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^b)\right) \\
\mathbf{P}_k^a &= (\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^b(\mathbf{I}-\mathbf{K}_k\mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^{\mathrm{T}}
\end{aligned}
$$

- These equations define the Extended Kalman Filter.
  - Note: the "extended" qualifier refers to the fact that we allow non-linear observation operators, and propagate the state using a nonlinear model. The standard Kalman filter is a purely linear analysis system in which $\mathcal{H}_k$ and $\mathcal{M}_{t_{k-1}\rightarrow t_k}$ are assumed to be linear.

ECMWF

The Extended Kalman Filter

- Subject to the assumptions we have made, the Kalman filter produces an optimal sequence of analyses.
- The analysis $\mathbf{x}_k^a$ is the best (minimum variance) estimate of the state at time t_k, given $\mathbf{x}_0^b$ and all observations up to time t_k (i.e. $\mathbf{y}_0 \cdots \mathbf{y}_k$).
- The inputs to the Kalman filter are:
  - An initial estimate of the state at time t_0, and the corresponding covariance matrix, $\mathbf{P}_0^b$.
  - Observations $\mathbf{y}_k$, and covariances of observation error, $\mathbf{R}_k$ at each analysis time.
  - Covariance matrices of model error, $\mathbf{Q}_k$.
- Note that, unlike OI, 3D-Var and 4D-Var, we do not have to specify the covariance matrix of background error — it is generated and propagated by the filter, using the model dynamics.
- However, we do have to specify $\mathbf{Q}_k$. This is difficult!

Kalman Filters for Large Dimensional Systems

- The Kalman filter is impractical for large dimension systems.
- It requires us to handle matrices of dimension $N \times N$, where $N \sim 10^8$.
  - The World's fastest computer can sustain $\sim 10^{15}$ operations per second.
  - Multiplying two $10^8 \times 10^8$ matrices requires 10^{24} operations, and would take about 32 years on this machine.
  - Evaluating $\mathbf{P}^b_k = \mathbf{M}_{t_{k-1}\rightarrow t_k}\mathbf{P}^a_{k-1}\mathbf{M}^{\mathrm{T}}_{t_{k-1}\rightarrow t_k} + \mathbf{Q}_k$ requires $N \sim 10^8$ model integrations.
- A range of approximate Kalman filters has been developed for use with large systems.
- All of these methods rely on a low-rank approximation of the covariance matrices of background and analysis error.

Kalman Filters for Large Dimensional Systems

- Suppose that $\mathbf{P}_k^b$ has rank $M \ll N$ (e.g. $M \sim 100$).
- Then we can write $\mathbf{P}_k^b = \mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$, where $\mathbf{X}_k^b$ is $N \times M$.
- The Kalman gain becomes:

$$\begin{aligned}\mathbf{K}_k &= \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} \left[\mathbf{H}_k \mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}} + \mathbf{R}_k\right]^{-1} \\ &= \mathbf{X}_k^b \left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} \left[\left(\mathbf{H}_k \mathbf{X}_k^b\right)\left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} + \mathbf{R}_k\right]^{-1}\end{aligned}$$

- Note that, to evaluate $\mathbf{K}$, we apply $\mathbf{H}_k$ to the M columns of $\mathbf{X}_k^b$, rather than to the N columns of $\mathbf{P}_k^b$
- Note also that the analysis increment, $\mathbf{x}_k^a - \mathbf{x}_k^b = \mathbf{K}_k \left(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k^b)\right)$, is a linear combination of the columns of $\mathbf{X}_k^b$.

Kalman Filters for Large Dimensional Systems

$$\mathbf{K}_k = \mathbf{X}_k^b \left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} \left[\left(\mathbf{H}_k \mathbf{X}_k^b\right)\left(\mathbf{H}_k \mathbf{X}_k^b\right)^{\mathrm{T}} + \mathbf{R}_k\right]^{-1}$$

- The analysis error covariance matrix is:

$$\mathbf{P}_k^a = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^b (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^{\mathrm{T}} + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^{\mathrm{T}}$$

- Since $\mathbf{P}_k^b = \mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$, we see that all terms in the expression for $\mathbf{P}_k^a$ contain an initial $\mathbf{X}_k^b$ and a final $\left(\mathbf{X}_k^b\right)^{\mathrm{T}}$.
- Hence

$$\mathbf{P}_k^a = \mathbf{X}_k^b \, \mathbf{W}_k \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$$

for some $M \times M$ matrix $\mathbf{W}_k$.

Kalman Filters for Large Dimensional Systems

$$\mathbf{P}_k^a = \mathbf{X}_k^b \, \mathbf{W}_k \left(\mathbf{X}_k^b \right)^{\mathrm{T}}$$

- The covariance matrix is propagated using:

$$\begin{aligned} \mathbf{P}_{k+1}^b &= \mathbf{M}_{t_k \rightarrow t_{k+1}} \mathbf{P}_{k-1}^a \mathbf{M}_{t_k \rightarrow t_{k+1}}^{\mathrm{T}} + \mathbf{Q}_{k+1} \\ &= \left(\mathbf{M}_{t_k \rightarrow t_{k+1}} \mathbf{X}_k^b \right) \mathbf{W}_k \left(\mathbf{M}_{t_k \rightarrow t_{k+1}} \mathbf{X}_k^b \right)^{\mathrm{T}} + \mathbf{Q}_{k+1} \end{aligned}$$

- Note that this requires only M integrations of the tangent linear model.
- The addition of $\mathbf{Q}_{k+1}$ means that, in general, $\mathbf{P}_{k+1}^b$ is not of low rank.
- However, we can approximate it by projecting onto some suitable M-dimensional subspace. The resulting algorithm is called a reduced-rank Kalman filter.

Kalman Filters for Large Dimensional Systems

- The severe reduction in rank causes significant problems for the reduced-rank kalman filter:
- The analysis increment is restricted to an M-dimensional subspace.
  - There are too few degrees of freedom available to fit the $\sim 10^6$ observations.
- The low-rank approximations of the covariance matrices suffer from spurious long-distance correlations. These cause two problems:
  - The analysis may generate spurious increments in regions where there are no observations.
  - The analysis may be unable to draw to isolated observations (e.g. over Antarctica) if it thinks there is a significant correlation with a well-observed region (e.g. Europe).
- There are two ways around these problems:
  - Local analysis (e.g. Evensen 2003, Ocean Dynamics 343–367; Ott *et al.* 2004, Tellus 415–428).
  - Shur product modification of the covariances (e.g. Houtekamer and Mitchell 2001, MWR 123–137).

ECMWF

Kalman Filters for Large Dimensional Systems

- Local analysis solves the analysis equations independently for each gridpoint, or for each of a set of regions covering the domain.
- Each analysis uses only observations that are local to the gridpoint (or region).
- This guarantees that the analysis at each gridpoint (or region) is not influenced by distant observations.
- The global analysis is constructed by stitching together the independent regional (or gridpoint) analyses, and is thus no longer a linear combination of the columns of $\mathbf{X}_k^b$.
- In effect, the method acts to vastly increase the dimension of the sub-space in which the analysis increment is constructed.
- However, performing independent analyses for each region is not optimal, and the method shares some of the problems of OI (e.g. poor analysis of the large scales, and difficulties in producing balanced analyses).

ECMWF

Kalman Filters for Large Dimensional Systems

- The Schur product approach uses the fact that if $\mathbf{B}$ and $\mathbf{C}$ are covariance matrices, then so is $\mathbf{A} = \mathbf{B} \circ \mathbf{C}$, where $\circ$ denotes the Schur (i.e. element-wise) product: $A_{ij} = B_{ij} C_{ij}$.
- Spurious long-range correlations in $\mathbf{P}_k^b$ may be suppressed by forming the Schur product with a covariance matrix representing a decaying function of distance.
  - The modified covariance matrix is never formed explicitly. Rather, the method deals with terms such as $\mathbf{P}_k^b \mathbf{H}_k^{\mathrm{T}}$.
- The modified covariance matrix is no longer of the form $\mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$.
- Forming the Schur product has the effect of vastly increasing the rank of the matrix.
- Choosing the product function is non-trivial. It is easy to modify $\mathbf{P}_k^b$ in undesirable ways. In particular, balance relationships may be adversely affected.

Ensemble Methods

- Ensemble Kalman filters are reduced-rank Kalman filters that construct their covariance matrices as sample covariance matrices:

$$\mathbf{P}_k^b = \frac{1}{M-1}\sum_{m=1}^{M-1}(\mathbf{x}_{k,m}^b - \overline{\mathbf{x}_{k,m}^b})(\mathbf{x}_{k,m}^b - \overline{\mathbf{x}_{k,m}^b})^{\mathrm{T}}$$

  where the subscript m refers to the sample (ensemble member).

- Note that we can write this as $\mathbf{P}_k^b = \mathbf{X}_k^b \left(\mathbf{X}_k^b\right)^{\mathrm{T}}$, where

$$\mathbf{X}_k^b = \frac{1}{\sqrt{M-1}}\left((\mathbf{x}_{k,1}^b - \overline{\mathbf{x}_{k,1}^b}), \quad (\mathbf{x}_{k,2}^b - \overline{\mathbf{x}_{k,2}^b}), \quad \cdots, \quad (\mathbf{x}_{k,M}^b - \overline{\mathbf{x}_{k,M}^b}) \right)$$

Ensemble Methods

- The Extended Kalman filter includes terms involving $\mathbf{M}_k$, $\mathbf{M}_k^{\mathrm{T}}$, $\mathbf{H}_k$ and $\mathbf{H}_k^{\mathrm{T}}$.
  - I.e. it uses the tangent linear and adjoint model and observation operators.
- In the ensemble Kalman filter, we avoid the need for tangent linear and adjoint operators by approximating:

$$\mathbf{P}_k^b\mathbf{H}_k^{\mathrm{T}} \approx \frac{1}{M-1}\sum_{m=1}^{M}\left(\mathbf{x}_{k,m}^b-\overline{\mathbf{x}_{k,m}^b}\right)\left(\mathcal{H}(\mathbf{x}_{k,m}^b)-\overline{\mathcal{H}(\mathbf{x}_{k,m}^b)}\right)^{\mathrm{T}}$$

$$\mathbf{H}_k\mathbf{P}_k^b\mathbf{H}_k^{\mathrm{T}} \approx \frac{1}{M-1}\sum_{m=1}^{M}\left(\mathcal{H}(\mathbf{x}_{k,m}^b)-\overline{\mathcal{H}(\mathbf{x}_{k,m}^b)}\right)\left(\mathcal{H}(\mathbf{x}_{k,m}^b)-\overline{\mathcal{H}(\mathbf{x}_{k,m}^b)}\right)^{\mathrm{T}}$$

- Not having to code tangent linear an adjoint operators is one of the main attractions of the ensemble Kalman filter!

ECMWF

Non-Gaussian Methods

- Kalman filters, as well as 3D-Var, 4D-Var and OI, are essentially Gaussian methods. They assume that the p.d.f. of error is fully described by the mean and covariance.
- Non-Gaussian methods do not make this assumption.
- Particle filters are a class of non-Gaussian method that approximate the p.d.f. by a discrete distribution:

$$p(\mathbf{x}) = \begin{cases} w_m & \text{if } \mathbf{x} = \mathbf{x}_{k,m} \\ 0 & \text{otherwise} \end{cases}$$

- An ensemble of forecasts $\{\mathbf{x}_m; m = 1 \cdots M\}$ is run, and each member is given an associated weight, w_m, according to its probability.
- When an observation, y, is available, the weights are adjusted using Bayes' theorem:

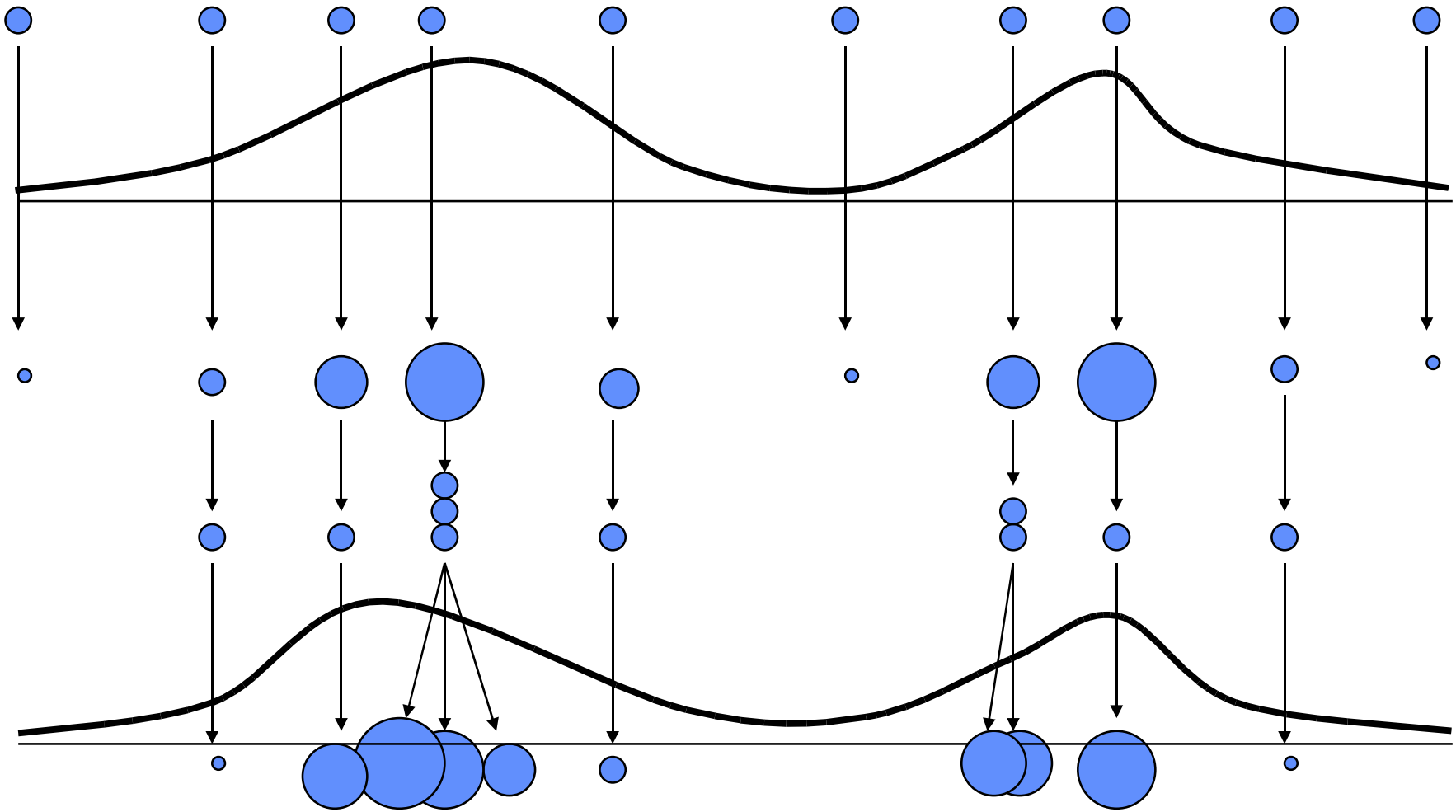
$$w_m^{\text{new}} = \frac{w_m^{\text{old}} p(y|\mathbf{x}_m)}{\sum_{m=1}^{M} w_m^{\text{old}} p(y|\mathbf{x}_m)}$$

Non-Gaussian Methods

- In its most basic form, this is all there is to a particle filter. States that agree with the observations get large weights, whereas states that disagree with the observations get small weights.
- In practice, the weights for some members become tiny. These members are no longer useful and they are dropped from the ensemble and replaced by new, more probable members.
- This is achieved by periodically resampling the p.d.f.
- A new ensemble is generated by randomly picking the old members with probability proportional to their weights. Members with large weight may be picked several times, whereas members with very small weight are unlikely to be picked.
- After resampling, all the weights are reset to $1/M$.
- Resampling may produce some identical members. However these diverge slowly from each other because each member is forced with different random perturbations that represent model error.

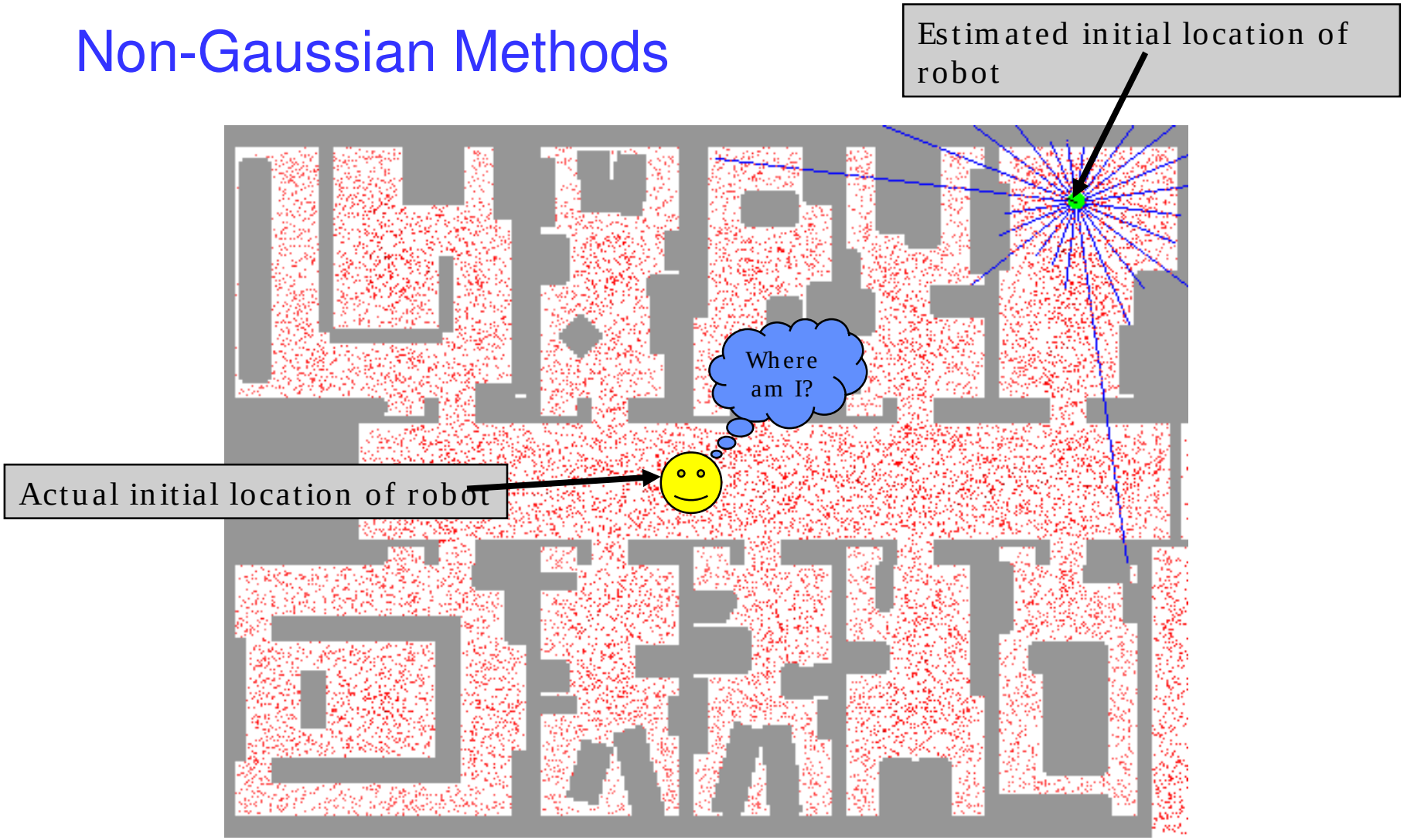
ECMWF

Non-Gaussian Methods



Non-Gaussian Methods

Non-Gaussian Methods



from: Fox et al. 1999, proc 16th National Conference on Artificial