Assimilation Algorithms
Lecture 3: 4D-Var

Mike Fisher

ECMWF

March 12, 2014

Outline

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 $\mathcal{G}$:

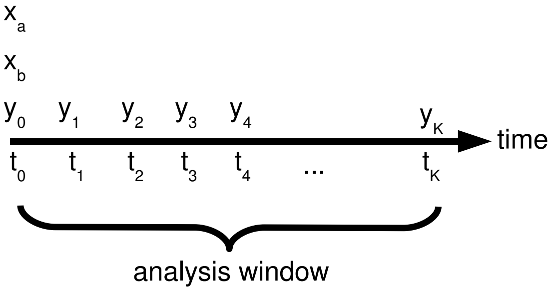
$$J(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}_b - \mathbf{x}\right)^{\mathrm{T}} \left(\mathbf{P}_b\right)^{-1} \left(\mathbf{x}_b - \mathbf{x}\right) + \frac{1}{2}\left(\mathbf{y} - \mathcal{G}(\mathbf{x})\right)^{\mathrm{T}} \mathbf{R}^{-1} \left(\mathbf{y} - \mathcal{G}(\mathbf{x})\right)$$

- 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:

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

Strong Constraint 4D-Var

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



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).

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

- 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:

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

- 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 $\mathcal{G}_k = \mathcal{H}_k \mathcal{M}_{t_0 \to t_k}$, this implies that the model is perfect:

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

- 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.

Strong Constraint 4D-Var

- In general, unconstrained minimization problems are easier to solve than constrained problems.
- To minimize the cost function, we write it as a function of $\mathbf{x}_0$:

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

- However, when evaluating the cost function, we can avoid repeated integrations of the model by using the following algorithm:
  - $J := \frac{1}{2}(\mathbf{x}_b - \mathbf{x}_0)^{\mathrm{T}} (\mathbf{P}_b)^{-1} (\mathbf{x}_b - \mathbf{x}_0)$
  - Repeat for $k = 0, 1, \cdots, K$:
  - $J := J + \frac{1}{2}(\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{H}_k(\mathbf{x}_k))$.
  - $\mathbf{x}_{k+1} := \mathcal{M}_{t_k \to t_{k+1}}(\mathbf{x}_k)$.

Strong Constraint 4D-Var

- As in 3D-Var, efficient minimization of the cost function requires us to calculate its gradient.
- Differentiating the unconstrained version of the cost function with respect to $\mathbf{x}_0$ gives:

$$\nabla J(\mathbf{x}_0) = (\mathbf{P}_b)^{-1}(\mathbf{x}_b - \mathbf{x}_0) + \sum_{k=0}^{K} \mathbf{G}_k^{\mathrm{T}} \mathbf{R}_k^{-1} (\mathbf{y}_k - \mathcal{G}_k(\mathbf{x}_0))$$

- Now, $\mathbf{G}_k$ is the Jacobian of $\mathcal{G}_k$, and:

$$\begin{aligned} \mathcal{G}_k &= \mathcal{H}_k \mathcal{M}_{t_0 \to t_k} \\ &= \mathcal{H}_k \mathcal{M}_{t_{k-1} \to t_k} \mathcal{M}_{t_{k-2} \to t_{k-1}} \cdots \mathcal{M}_{t_0 \to t_1} \end{aligned}$$

- Hence:

$$\begin{aligned} \mathbf{G}_k &= \mathbf{H}_k \mathbf{M}_{t_{k-1} \to t_k} \mathbf{M}_{t_{k-2} \to t_{k-1}} \cdots \mathbf{M}_{t_0 \to t_1} \\ \Rightarrow \mathbf{G}_k^{\mathrm{T}} &= \mathbf{M}_{t_0 \to t_1}^{\mathrm{T}} \cdots \mathbf{M}_{t_{k-2} \to t_{k-1}}^{\mathrm{T}} \mathbf{M}_{t_{k-1} \to t_k}^{\mathrm{T}} \mathbf{H}_k^{\mathrm{T}} \end{aligned}$$

Strong Constraint 4D-Var

- Let us consider how to evaluate the second term of $\nabla J(\mathbf{x}_0)$:

$$
\begin{aligned}
\sum_{k=0}^{K} \mathbf{G}_k^{\mathrm{T}} \mathbf{R}_k^{-1} \left(\mathbf{y}_k - \mathcal{G}_k(\mathbf{x}_0)\right) = \\
& \mathbf{H}_0^{\mathrm{T}} \mathbf{R}_0^{-1} \left(\mathbf{y}_0 - \mathcal{G}_0(\mathbf{x}_0)\right) \\
& + \mathbf{M}_{t_0 \to t_1}^{\mathrm{T}} \mathbf{H}_1^{\mathrm{T}} \mathbf{R}_1^{-1} \left(\mathbf{y}_1 - \mathcal{G}_1(\mathbf{x}_0)\right) \\
& + \mathbf{M}_{t_0 \to t_1}^{\mathrm{T}} \mathbf{M}_{t_1 \to t_2}^{\mathrm{T}} \mathbf{H}_2^{\mathrm{T}} \mathbf{R}_2^{-1} \left(\mathbf{y}_2 - \mathcal{G}_2(\mathbf{x}_0)\right) \\
& \quad \vdots \\
& + \mathbf{M}_{t_0 \to t_1}^{\mathrm{T}} \mathbf{M}_{t_1 \to t_2}^{\mathrm{T}} \cdots \mathbf{M}_{t_{K-1} \to t_K}^{\mathrm{T}} \mathbf{H}_K^{\mathrm{T}} \mathbf{R}_K^{-1} \left(\mathbf{y}_K - \mathcal{G}_K(\mathbf{x}_0)\right)
\end{aligned}
$$

$$
\begin{aligned}
= \quad & \mathbf{H}_0^{\mathrm{T}} \mathbf{R}_0^{-1} \left(\mathbf{y}_0 - \mathcal{G}_0(\mathbf{x}_0)\right) + \mathbf{M}_{t_0 \to t_1}^{\mathrm{T}} [\mathbf{H}_1^{\mathrm{T}} \mathbf{R}_1^{-1} \left(\mathbf{y}_1 - \mathcal{G}_1(\mathbf{x}_0)\right) \\
& + \mathbf{M}_{t_1 \to t_2}^{\mathrm{T}} [\mathbf{H}_2^{\mathrm{T}} \mathbf{R}_2^{-1} \left(\mathbf{y}_2 - \mathcal{G}_2(\mathbf{x}_0)\right) + \mathbf{M}_{t_2 \to t_3}^{\mathrm{T}} [\cdots \\
& \cdots + \mathbf{M}_{t_{K-1} \to t_K}^{\mathrm{T}} \mathbf{H}_K^{\mathrm{T}} \mathbf{R}_K^{-1} \left(\mathbf{y}_K - \mathcal{G}_K(\mathbf{x}_0)\right)] \cdots]]]
\end{aligned}
$$

Strong Constraint 4D-Var

- Hence, to evaluate the gradient of the cost function, we can ues the following algorithm:
  - Set $\nabla J := 0$.
  - Repeat for $k = K, K-1, \ldots 1$:
    - ⋆ $\nabla J := \nabla J + \mathbf{H}_k^{\mathrm{T}} (\mathbf{y}_k - \mathcal{G}_k(\mathbf{x}_k))$
    - ⋆ $\nabla J := \mathbf{M}_{t_{k-1} \to t_k}^{\mathrm{T}} \nabla J$
  - Finally add the contribution from the observations at t_0, and the contribution from the background term:
    $\nabla J := \nabla J + \mathbf{H}_0^{\mathrm{T}} (\mathbf{y}_0 - \mathcal{G}_0(\mathbf{x}_0)) + (\mathbf{P}_b)^{-1} (\mathbf{x}_b - \mathbf{x}_0)$.
- Note that the gradient can be evaluated with one application of each $\mathbf{M}_{t_{k-1} \to t_k}^{\mathrm{T}}$ for each k.
- Each $\mathbf{M}_{t_{k-1} \to t_k}^{\mathrm{T}}$ corresponds to a timestep of the adjoint model.
- Note that the adjoint model is integrated backwards in time, starting from t_K and ending with t_0.

The Incremental Method

- We have seen how the 4D-Var cost function and gradient can be evaluated for the cost of
  - one integration of the forecast model
  - one integration of the adjoint model
- This cost is still prohibitive:
  - A typical minimization will require between 10 and 100 evaluations of the gradient.
  - The cost of the adjoint model is typically 3 times that of the forward model.
  - The analysis window in the ECMWF system is 12-hours.
- Hence, the cost of the analysis is roughly equivalent to between 20 and 200 days of model integration.
- The incremental algorithm reduces the cost of 4D-Var by reducing the resolution of the model.

The Incremental Method

- The incremental method can be applied to both 3D-Var and 4D-Var, so let's return to the general expression for the cost function:

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

- We introduce a linearization state $\mathbf{x}^{(m)}$, and write

$$\mathbf{x} = \mathbf{x}^{(m)} + \delta\mathbf{x}^{(m)}$$

- The cost function can be written in terms of the increment $\delta\mathbf{x}^{(m)}$, and approximated by the quadratic function:

$$\begin{aligned} J(\delta\mathbf{x}^{(m)}) \quad = \quad & \frac{1}{2}\left(\mathbf{x}_b - \mathbf{x}^{(m)} - \delta\mathbf{x}^{(m)}\right)^{\mathrm{T}} (\mathbf{P}_b)^{-1} \left(\mathbf{x}_b - \mathbf{x}^{(m)} - \delta\mathbf{x}^{(m)}\right) \\ & + \frac{1}{2}\left(\mathbf{d}^{(m)} - \mathbf{G}\delta\mathbf{x}^{(m)}\right)^{\mathrm{T}} \mathbf{R}^{-1} \left(\mathbf{d}^{(m)} - \mathbf{G}\delta\mathbf{x}^{(m)}\right) \end{aligned}$$

where $\mathbf{d}^{(m)} = \mathbf{y} - \mathcal{G}\left(\mathbf{x}^{(m)}\right)$.

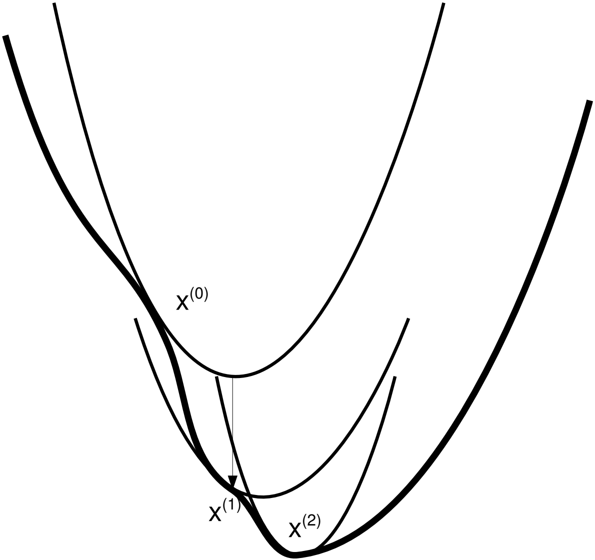
The Incremental Method

- The incremental method treats the minimization of J as a sequence of quadratic problems:
  - Repeat for $m = 0, 1, \cdots$ until convergence:
  - Minimize the quadratic cost function $J(\delta \mathbf{x}^{(m)})$.
  - Set $\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \delta \mathbf{x}^{(m)}$.
- In this form, if the minimization converges, it will converge to the solution of the original problem.
- However, to reduce the computational cost of the analysis, we can make a further approximation, and evaluate the quadratic cost function at lower resolution:

$$
\begin{aligned}
J(\delta \tilde{\mathbf{x}}^{(m)}) \quad = \quad & \frac{1}{2}\left(\tilde{\mathbf{x}}_b - \tilde{\mathbf{x}}^{(m)} - \delta \tilde{\mathbf{x}}^{(m)}\right)^{\mathrm{T}} \left(\tilde{\mathbf{P}}_b\right)^{-1} \left(\tilde{\mathbf{x}}_b - \tilde{\mathbf{x}}^{(m)} - \delta \tilde{\mathbf{x}}^{(m)}\right) \\
& + \frac{1}{2}\left(\mathbf{d}^{(m)} - \tilde{\mathbf{G}} \delta \tilde{\mathbf{x}}^{(m)}\right)^{\mathrm{T}} \mathbf{R}^{-1} \left(\mathbf{d}^{(m)} - \tilde{\mathbf{G}} \delta \tilde{\mathbf{x}}^{(m)}\right)
\end{aligned}
$$

where $\tilde{\cdot}$ indicates low resolution, and where $\tilde{\mathbf{x}}_b$, etc. are interpolated from the corresponding full-resolution fields.

The Incremental Method



The Incremental Method

$$J(\delta\tilde{\mathbf{x}}^{(m)}) = \frac{1}{2}\left(\tilde{\mathbf{x}}_b - \tilde{\mathbf{x}}^{(m)} - \delta\tilde{\mathbf{x}}^{(m)}\right)^{\mathrm{T}}\left(\tilde{\mathbf{P}}_b\right)^{-1}\left(\tilde{\mathbf{x}}_b - \tilde{\mathbf{x}}^{(m)} - \delta\tilde{\mathbf{x}}^{(m)}\right) + \frac{1}{2}\left(\mathbf{d}^{(m)} - \tilde{\mathbf{G}}\delta\tilde{\mathbf{x}}^{(m)}\right)^{\mathrm{T}}\mathbf{R}^{-1}\left(\mathbf{d}^{(m)} - \tilde{\mathbf{G}}\delta\tilde{\mathbf{x}}^{(m)}\right)$$

- When the quadratic cost function is approximated in this way, 4D-Var no longer converges to the solution of the original problem.
- The analysis increments are calculated at reduced resolution and must be interpolated to the high-resolution model's grid.
- Note, however that $\mathbf{d}^{(m)} = \mathbf{y} - \mathcal{G}\left(\mathbf{x}^{(m)}\right)$ is evaluated using the full-resolution versions of $\mathcal{G}$ and $\mathbf{x}^{(m)}$.
- I.e. the observations are always compared with the *full resolution* linearization state. The reduced-resolution observation operator only appears applied to increments: $\tilde{\mathbf{G}}\delta\tilde{\mathbf{x}}^{(m)}$.

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:

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

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

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

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:

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

- 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:

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

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

Weak Constraint 4D-Var

- In strong constraint 4D-Var, we can use the constraints to reduce the problem of minimizing a function of $\mathbf{x}_0 \cdots \mathbf{x}_K$ to that of minimizing a function of the initial state $\mathbf{x}_0$ only.
- This is not possible in weak constraint 4D-Var — we must either:
  - minimize the function $J(\mathbf{x}_0 \cdots \mathbf{x}_K)$, or:
  - express the cost function as a function of $\mathbf{x}_0$ and $\eta_1 \cdots \eta_K$.
- Although the two approaches are mathematically equivalent, they lead to very different minimization problems, with different possibilities for preconditioning.
  - It is not yet clear which approach is the best.
  - Formulation of an incremental method for weak constraint 4D-Var also remains a topic of research.
- Finally, note that model error is unlikely to be temorally uncorrelated.
  - Indeeed, initial attempts to account for model error in the ECMWF analysis are concentrated on representing only the bias component of model error (i.e. model error is assumed constant in time).

Summary

- Strong Constraint 4D-Var is an extension of 3D-Var to the case where observations are distributed in time.
- The observation operators are generalised to include an integration of the forecast model.
- The model is assumed to be perfect, so that the four-dimensional analysis state corresponds to an integration (trajectory) of the model.
- The incremental method allows the computational cost to be reduced to acceptable levels.
- Weak Constraint 4D-Var allows the perfect model assumption to be removed.
- This allows longer windows to be contemplated.
- However, it requires knowledge of the statistics of model error, and the ability to express this knowledge in the form of covariance matrices.
- The statistical description of model error is one of the main current challenges in data assimilation.