What to use when?

With a brief recap

Lecturer: Ross Bannister

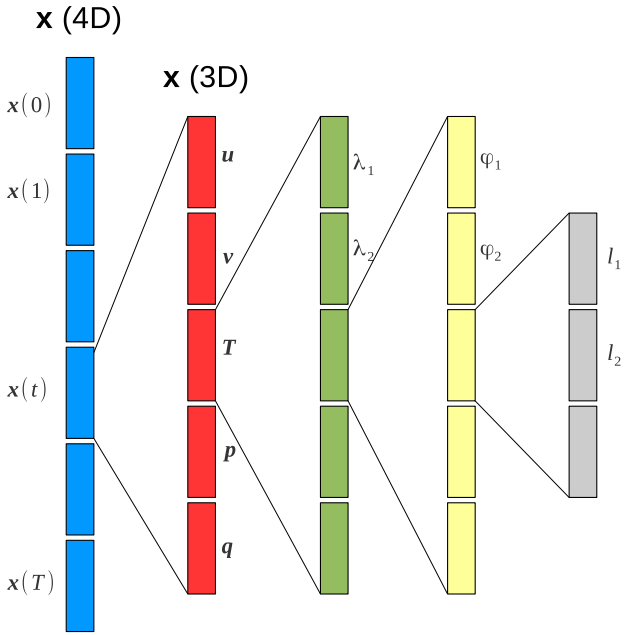
NCEO, Dept. of Meteorology, Univ. of Reading

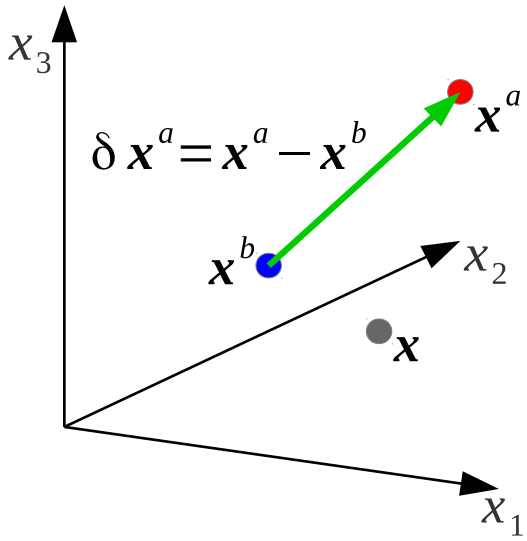
21–24 June 2022, Univ. of Reading

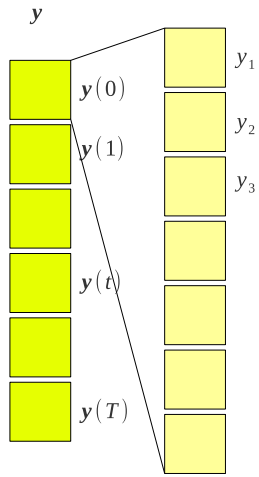
Reminder: what is data assimilation?

- To blend information from models and observations.
  - State/parameter estimation (some kind of 'optimal' blending).
  - The posterior PDF or certain moments of it.

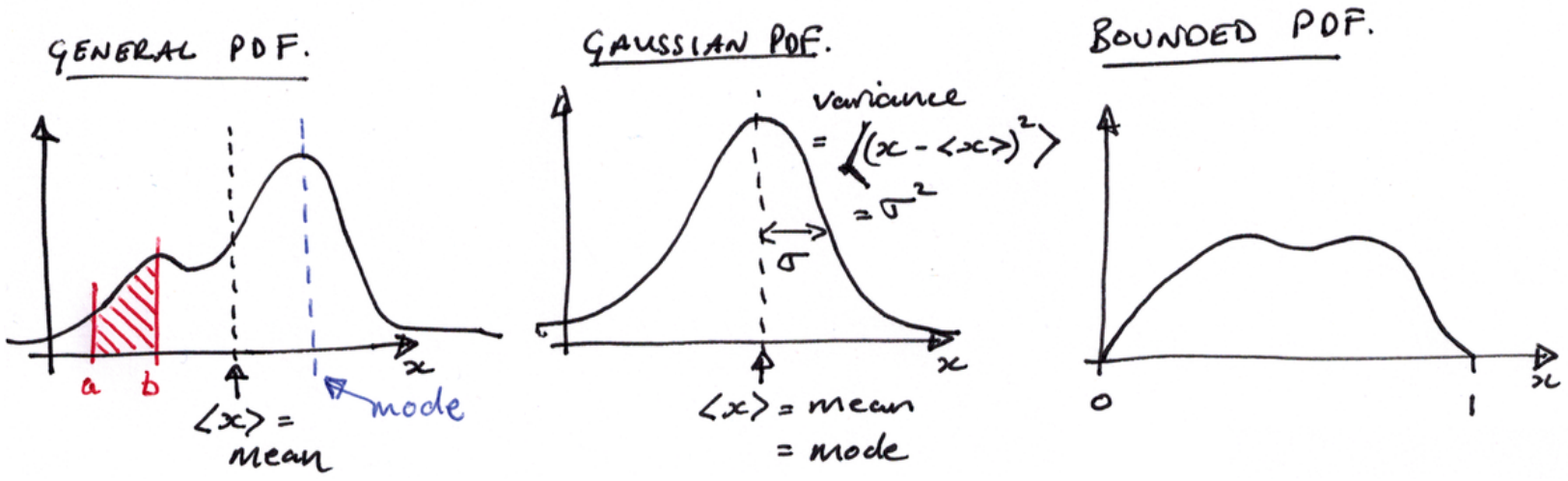
State vector and observation vectors







Probability Density Functions



$\int_a^b p(x)dx$ is the probability that X lies between values a and b.

Form of a Gaussian:

$$\mathbf{x} \sim N(\mathbf{x}^{\mathrm{b}}, \mathbf{B})$$

$$p(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n \det(\mathbf{B})}} \exp -\frac{1}{2} \left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right)^{\mathrm{T}} \mathbf{B}^{-1} \left(\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right)$$

Covariance matrices

A covariance matrix describes the second moment of a PDF.

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}, \qquad \mathbf{x}' = \mathbf{x} - \langle \mathbf{x} \rangle$$

$$\mathrm{cov}(\mathbf{x}') = \langle \mathbf{x}' \mathbf{x}'^{\mathrm{T}} \rangle = \begin{pmatrix} \langle x_1'^2 \rangle & \langle x_1' x_2' \rangle & \cdots & \langle x_1' x_n' \rangle \\ \langle x_2' x_1' \rangle & \langle x_2'^2 \rangle & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ \langle x_n' x_1' \rangle & \cdots & \cdots & \langle x_n'^2 \rangle \end{pmatrix}$$

Bayes' Theorem – the foundation of DA

$$p(\mathbf{x}|\mathbf{y}) = \frac{p(\mathbf{x}) \times p(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}$$

$$\text{posterior dist.} = \frac{\text{prior dist.} \times \text{likelihood}}{\text{normalizing constant}}$$

- Prior distribution: PDF of the state before observations are considered (e.g. PDF of model forecast).
- Likelihood: PDF of observations given that the state is $\mathbf{x}$.
- Posterior distribution: PDF of the state given the observations.

Association between non-linearity and non-Gaussianity

Example with observation operator (observation of a scalar, non-linear $\mathscr{H}$)

$$\begin{aligned} y^{\mathrm{m}} &= \mathscr{H}(x) = \mathscr{H}(x^{\mathrm{R}} + \delta x) \\ y^{\mathrm{m}} &\approx \mathscr{H}(x^{\mathrm{R}}) + H\delta x \end{aligned}$$

Suppose the likelihood term has the form:

$$p(y|x) \propto \exp -\frac{(y - \mathscr{H}(x))^2}{2\sigma_y^2} \quad \propto \quad \exp(\text{non-quadratic in } x)$$

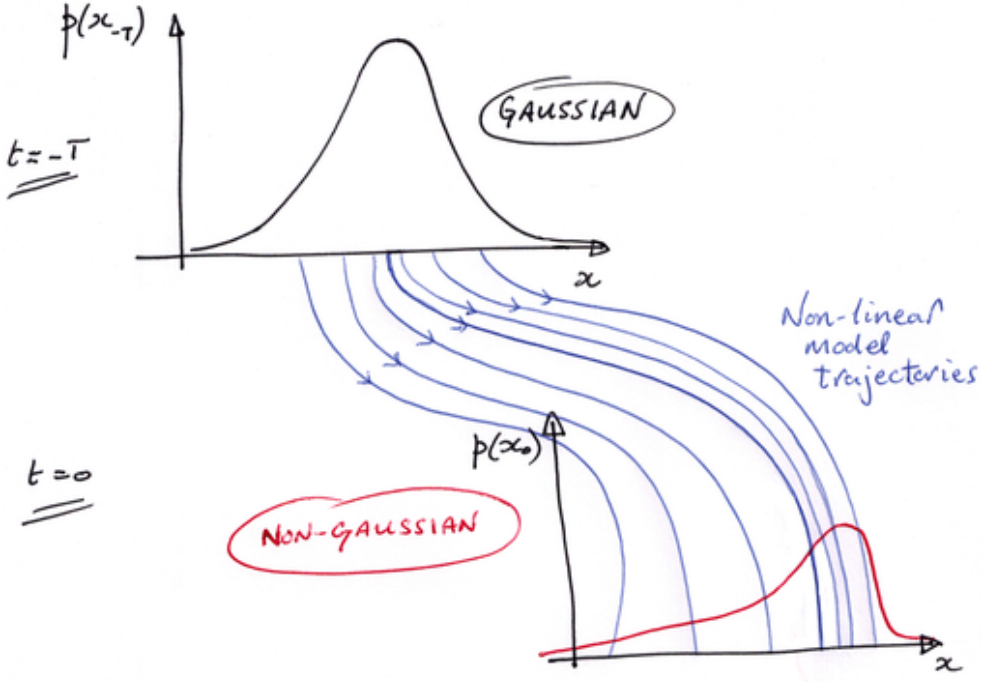
$$\text{linearize } \ldots$$

$$p(y|x^{\mathrm{R}}, \delta x) \sim \exp -\frac{\left(y - \mathscr{H}(x^{\mathrm{R}}) - H\delta x\right)^2}{2\sigma_y^2} \quad \propto \quad \exp(\text{quadratic in } \delta x)$$

Non-linearity leads to non-Gaussianity; Gaussianity can be approximately preserved if non-linearity is weak.

Association between non-linearity and non-Gaussianity (cont)

A non-linear forecast model $\mathbf{x}_0 = \mathscr{M}(\mathbf{x}_{-T})$

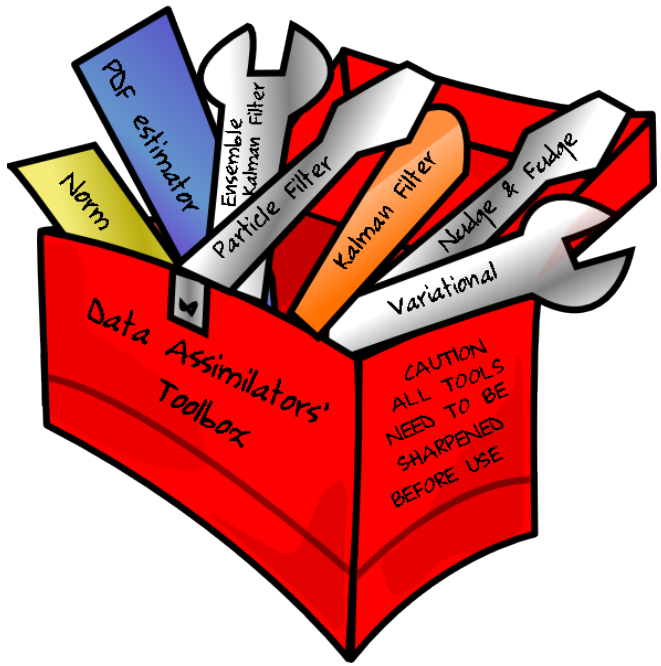


Again non-linearity leads to non-Gaussianity.

Confused? Overwhelmed?

In realistic practical applications we cannot represent the PDFs explicitly, so we need approximate DA methods

- Kalman filter (+ extended KF)
- Variational data assimilation
- Ensemble Kalman filters
- En-Var filters
- Hybrid methods
- Particle filters



Which method to use for your application?

Data insertion/nudging

Data insertion

Overwrite model values with observations, $x_i \rightarrow y$.

Dangerous – e.g. sudden jump in value.

Nudging

Introduce observations gradually, e.g. for one observation of gridpoint i:

$$\frac{\partial \mathbf{x}}{\partial t} = m(\mathbf{x}) - \mathbf{f}_i \frac{(x_i - y)}{\tau}$$

$\mathbf{f}_i$ structure function associated with obs position, τ timescale

- No account of uncertainty of model or obs.
- How to treat indirect observations?
- No information in observation voids

The Kalman filter (and extended Kalman filter)

▷ Propagates the mean state and its error covariance sequentially; ▷ forecast/analysis is mean of the prior/posterior; ▷ the analysis is the state that has minimum variance; ▷ strong theoretical basis.

$$
\begin{aligned}
\text{forecast state: } \mathbf{x}_t^{\mathrm{f}} &= \mathscr{M}_t(\mathbf{x}_{t-1}^{\mathrm{a}}) \\
\text{forecast covariance: } \mathbf{P}_t^{\mathrm{f}} &= \mathbf{M}_t \mathbf{P}_{t-1}^{\mathrm{a}} \mathbf{M}_t^{\mathrm{T}} + \mathbf{Q}_t \\
\text{analysis state: } \mathbf{x}_t^{\mathrm{a}} &= \mathbf{x}_t^{\mathrm{f}} + \mathbf{K}_t \left(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t^{\mathrm{f}}) \right) \\
\text{analysis covariances: } \mathbf{P}_t^{\mathrm{a}} &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}_t) \mathbf{P}_t^{\mathrm{f}} \\
\text{Kalman gain: } \mathbf{K}_t &= \mathbf{P}_t^{\mathrm{f}} \mathbf{H}_t^{\mathrm{T}} \left(\mathbf{H}_t \mathbf{P}_t^{\mathrm{f}} \mathbf{H}_t^{\mathrm{T}} + \mathbf{R}_t \right)^{-1}
\end{aligned}
$$

- Assumes Gaussian prior and observations.
- Assumes $\mathscr{M}$ and $\mathscr{H}$ are linear (weak non-linearity is allowed in the extended KF).
- Unfeasible when n is large as matrices are treated explicitly.

Traditional variational data assimilation

▷ Forecast is mean of the prior, analysis is mode of the posterior (minimises a cost fn); ▷ OK when n is large; ▷ iterative method of solution; ▷ can add extra terms (e.g. weak constraint for model error); ▷ strong theoretical basis; ▷ $\mathscr{M}$ and $\mathscr{H}$ can be non-linear; ▷ usually incremental formulation used

$$\begin{aligned} \text{forecast state: } \mathbf{x}_0^{\mathrm{f}} &= \mathscr{M}_{-T}(\mathbf{x}_{-T}^{\mathrm{a}}) \\ \text{analysis state: } \mathbf{x}_0^{\mathrm{a}} &= \mathbf{x}_0^{\mathrm{f}} + \underset{\delta\mathbf{x}_0}{\arg\min}\, J[\delta\mathbf{x}_0] \end{aligned}$$

$$\begin{aligned} J[\delta\mathbf{x}_0] &= \frac{1}{2}\delta\mathbf{x}_0^{\mathrm{T}}\mathbf{B}_0^{-1}\delta\mathbf{x}_0 + \frac{1}{2}\sum_{t=0}^{T}\left(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t^{\mathrm{f}}) - \mathbf{H}_t\delta\mathbf{x}_t\right)^{\mathrm{T}}\mathbf{R}_t^{-1}(\bullet) \\ &\text{subject to } \delta\mathbf{x}_{t+1} = \mathbf{M}_t(\delta\mathbf{x}_t) \\ &\mathbf{x}_{t+1}^{\mathrm{f}} = \mathscr{M}_t\left(\mathbf{x}_t^{\mathrm{f}}\right) \end{aligned}$$

Flavours: weak-constraint 4DVar (additional control vectors); strong-constraint 4DVar (as above); 3DFGAT (set $\mathbf{M}_t = \mathbf{I}$); 3DVar (use persistence model in the cost function, $\mathscr{M}_t(\mathbf{x}_t) = \mathbf{x}_t$).

Traditional variational data assimilation (cont)

$$\text{grad: } \nabla_{\delta \mathbf{x}_0} J[\delta \mathbf{x}_0] = \mathbf{B}_0^{-1} \delta \mathbf{x}_0 - \sum_{t=0}^{T} \mathbf{M}_0^{\mathrm{T}} \cdots \mathbf{M}_t^{\mathrm{T}} \mathbf{H}_t^{\mathrm{T}} \mathbf{R}_t^{-1} \left(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t^{\mathrm{f}}) - \mathbf{H}_t \delta \mathbf{x}_t \right)$$

- Assumes Gaussian prior and observations.
- Mode of posterior is equivalent to mean of KF if $\mathbf{B}_0 = \mathbf{P}_t^{\mathrm{f}}$, all obs at same time and $\mathscr{H}_t$ linear.
- Analysis is sub-optimal if $\mathscr{M}$ or $\mathscr{H}$ is non-linear; can end up in a local minimum.
- $\mathbf{B}_0$ is modelled/parametrised (e.g. need control variable transforms) – not properly flow-dependent and is too simple.
- Need tangent linear of $\mathscr{M}_t$ and $\mathscr{H}_t$ and their adjoints (for gradient calculation).
- Usually no second moment of analysis found.

Ensemble Kalman filters

▷ Based on KF equations; ▷ propagates N-member ensemble of forecasts to estimate $\mathbf{P}_t^{\mathrm{f}}$; ▷ $\mathscr{M}$ and $\mathscr{H}$ can be non-linear; ▷ works when $N \ll n$ (but see below); ▷ avoids linear/adjoint coding; ▷ easy to code; ▷ parallelization is scalable with N.

$$
\begin{aligned}
& & \mathbf{x}_t^{(i),\mathrm{f}} &= \mathscr{M}_t(\mathbf{x}_{t-1}^{(i),\mathrm{a}}) + \beta^{(i)} \\
n \times N: & & \mathbf{X}_t'^{\mathrm{f}} &= \left(\mathbf{x}_t^{(1),\mathrm{f}} - \bar{\mathbf{x}}_t^{\mathrm{f}} \quad \cdots \quad \mathbf{x}_t^{(N),\mathrm{f}} - \bar{\mathbf{x}}_t^{\mathrm{f}}\right) \\
p \times N: & & \mathbf{Y}_t' &= \left(\mathscr{H}_t(\mathbf{x}_t^{(1),\mathrm{f}}) - \mathscr{H}(\bar{\mathbf{x}}_t^{\mathrm{f}}) \quad \cdots \quad \mathscr{H}_t(\mathbf{x}_t^{(N),\mathrm{f}}) - \mathscr{H}_t(\bar{\mathbf{x}}_t^{\mathrm{f}})\right) \\
n \times N: & & \mathbf{X}_t'^{\mathrm{a}} &= \left(\mathbf{x}_t^{(1),\mathrm{a}} - \bar{\mathbf{x}}_t^{\mathrm{a}} \quad \cdots \quad \mathbf{x}_t^{(N),\mathrm{a}} - \bar{\mathbf{x}}_t^{\mathrm{a}}\right)
\end{aligned}
$$

Stochastic EnKF

$$
\begin{aligned}
\mathbf{x}_t^{(i)\mathrm{a}} &= \mathbf{x}_t^{(i)\mathrm{f}} + \mathbf{K}_t\left(\mathbf{y}_t + \varepsilon_y^{(i)} - \mathscr{H}_t(\mathbf{x}_t^{(i)\mathrm{f}})\right) \\
\mathbf{K}_t &= \mathbf{X}_t'^{\mathrm{f}}\mathbf{Y}_t'^{\mathrm{T}}\left(\mathbf{Y}_t'\mathbf{Y}_t'^{\mathrm{T}} + (N-1)\mathbf{R}_t\right)^{-1}
\end{aligned}
$$

Ensemble Transform KF

$$
\begin{aligned}
\bar{\mathbf{x}}_t^{\mathrm{a}} &= \bar{\mathbf{x}}_t^{\mathrm{f}} + \mathbf{K}_t\left(\mathbf{y}_t - \mathscr{H}_t(\bar{\mathbf{x}}_t^{\mathrm{f}})\right) \\
\mathbf{X}_t'^{\mathrm{a}} &= \mathbf{X}_t'^{\mathrm{f}}\mathbf{T}_t \\
\mathbf{K}_t &= \mathbf{X}_t'^{\mathrm{f}}\mathbf{T}_t\mathbf{T}_t^{\mathrm{T}}\mathbf{Y}_t'^{\mathrm{T}}\mathbf{R}_t^{-1} \\
\mathbf{T}_t &= \left(\mathbf{I} + \mathbf{Y}_t'^{\mathrm{T}}\mathbf{R}_t^{-1}\mathbf{Y}_t'\right)^{-1/2}
\end{aligned}
$$

Ensemble Kalman filters (cont)

Flavours

- Stochastic EnKF
- Singular Evolutive Interpolated Kalman Filter (SEIK)
- Ensemble Transform Kalman Filter (ETKF)
- Ensemble Adjustment Kalman Filter (EAKF)
- Ensemble Square Root Filter (EnSRF)
- etc.

Ensemble Kalman filters (cont)

- Assumes Gaussian prior and observations.
- $\mathbf{P}_t^{\mathrm{f}}$-matrix $= \mathbf{X}_t'^{\mathrm{f}} \mathbf{X}_t'^{\mathrm{f}^{\mathrm{T}}} / (N-1)$, rank deficient ($N < n$)
- Sampling noise (e.g. spurious covariances)
  - Needs localization to fix (how depends on the flavour)
    - Localization can be applied in model space: $\mathbf{P}_t^{\mathrm{f}} \circ \mathbf{\Omega}$ (not efficient if done explicitly)
    - Localization can be applied in 'observation' space, e.g. $\mathbf{K}_t \approx \rho \circ \left[\mathbf{X}_t'^{\mathrm{f}} \mathbf{Y}_t'^{\mathrm{T}}\right] \left(\rho \circ \left[\mathbf{Y}_t' \mathbf{Y}_t'^{\mathrm{T}}\right] + (N-1)\mathbf{R}_t\right)^{-1}$
    - Localization can be applied by performing a separate analysis for each grid point and using observations in the locality only (as in LETKF)
    - Localization can disturb physical properties of ensemble (e.g. balance).
- Filter divergence (ensemble under-spread)
  - Needs inflation (additive, multiplicative, relaxation to prior, ...)

EnVar (ensemble-variational)

▷ As variational DA, but where $\mathbf{B} \rightarrow \mathbf{X}_0'^{\mathrm{f}} \mathbf{X}_0'^{\mathrm{f}\mathrm{T}}/(N-1)$ from a parallel ensemble; ▷ has the benefits of variational DA but with a flow-dependent $\mathbf{B}$-matrix; ▷ analysis increment is a linear combination of forecast ensemble perturbations. E.g. En4DVar:

$$
\begin{aligned}
\mathbf{x}_0^{\mathrm{a}} &= \mathbf{x}_0^{\mathrm{f}} + \mathbf{X}'^{\mathrm{f}} \delta\mathbf{v}_{\mathrm{ens}}/\sqrt{N-1} \qquad \delta\mathbf{v}_{\mathrm{ens}} \text{ is an } N\text{-element vector} \\
J[\delta\mathbf{v}_{\mathrm{ens}}] &= \frac{1}{2}\delta\mathbf{v}_{\mathrm{ens}}^{\mathrm{T}}\delta\mathbf{v}_{\mathrm{ens}} + \frac{1}{2}\sum_{t=0}^{T}\left(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t)\right)^{\mathrm{T}}\mathbf{R}_t^{-1}\left(\bullet\right) \\
&\quad \text{subject to } \delta\mathbf{x}_{t+1} = \mathbf{M}_t\left(\delta\mathbf{x}_t\right) \quad \text{and } \delta\mathbf{x}_0 = \mathbf{X}'^{\mathrm{f}}\delta\mathbf{v}_{\mathrm{ens}}/\sqrt{N-1} \\
&\quad \mathbf{x}_{t+1}^{\mathrm{f}} = \mathscr{M}_t\left(\mathbf{x}_t^{\mathrm{f}}\right)
\end{aligned}
$$

- Assumes Gaussian prior and observations.
- $\mathbf{P}^{\mathrm{f}}$-matrix $= \mathbf{X}_0'^{\mathrm{f}} \mathbf{X}_0'^{\mathrm{f}\mathrm{T}}/(N-1)$, rank deficient ($N < n$).
- Needs localization and a separate parallel ensemble.
- EnVar schemes can get very complex with localization scheme.

En4DVar still needs the linear model and adjoint. 4DEnVar uses 4D ensembles and avoids these, but localization becomes very difficult. Worth considering for very large ensemble.

Hybrid methods

As variational DA, but where

$$\mathbf{B}_0 \rightarrow (1-\beta)\mathbf{B}_0 + \beta \mathbf{X}_0^{\prime \mathrm{f}} \mathbf{X}_0^{\prime \mathrm{f}^{\mathrm{T}}} / (N-1)$$

(new matrix is full rank and flow-dependent).

Hybrid methods (cont)

Traditional 4DVar with control variable transform:

$$J[\delta \mathbf{v}_{\mathrm{B}}] = \frac{1}{2}\delta \mathbf{v}_{\mathrm{B}}^{\mathrm{T}}\delta \mathbf{v}_{\mathrm{B}} + \frac{1}{2}\sum_{t=0}^{T}\left(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t^{\mathrm{f}}) - \mathbf{H}_t\delta \mathbf{x}_t\right)^{\mathrm{T}}\mathbf{R}_t^{-1}(\bullet)$$

$$\text{subject to } \delta \mathbf{x}_{t+1} = \mathbf{M}_t(\delta \mathbf{x}_t), \quad \delta \mathbf{x}_0 = \mathbf{U}\delta \mathbf{v}_{\mathrm{B}}$$

Hybrid-En4DVar:

$$J[\delta \mathbf{v}_{\mathrm{B}}, \delta \mathbf{v}_{\mathrm{ens}}] = \frac{1}{2}\delta \mathbf{v}_{\mathrm{B}}^{\mathrm{T}}\delta \mathbf{v}_{\mathrm{B}} + \frac{1}{2}\delta \mathbf{v}_{\mathrm{ens}}^{\mathrm{T}}\delta \mathbf{v}_{\mathrm{ens}} + \frac{1}{2}\sum_{t=0}^{T}\left(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t^{\mathrm{f}}) - \mathbf{H}_t\delta \mathbf{x}_t\right)^{\mathrm{T}}\mathbf{R}_t^{-1}(\bullet)$$

$$\text{subject to } \delta \mathbf{x}_{t+1} = \mathbf{M}_t(\delta \mathbf{x}_t), \quad \delta \mathbf{x}_0 = \sqrt{1-\beta}\mathbf{U}\delta \mathbf{v}_{\mathrm{B}} + \sqrt{\beta}\mathbf{X}'^{\mathrm{f}}\delta \mathbf{v}_{\mathrm{ens}}/\sqrt{N-1}$$

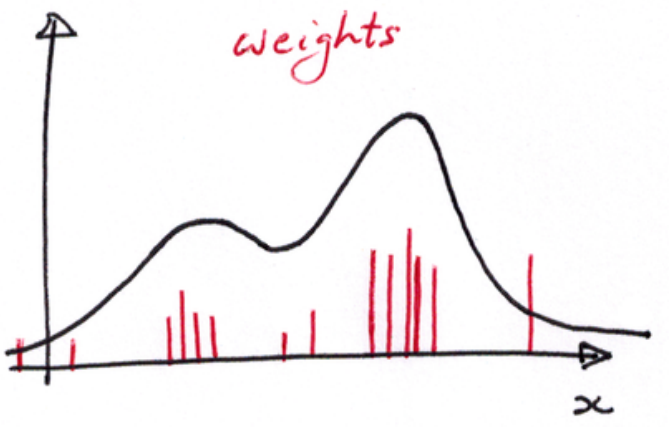
- Assumes Gaussian prior and observations.
- Still needs localization and a separate parallel ensemble.
- Can get very complex to develop.

Particle filters

▷ Non-Gaussian; ▷ fundamentally no need for covariance matrices; ▷ approximates prior and posterior PDFs as summation of 'delta-functions'.
Standard PF:

$$\text{prior PDF: } p(\mathbf{x}) = \sum_{i=1}^{N} w_i^{\text{prior}} \delta(\mathbf{x}-\mathbf{x}_i), \qquad \sum_{i=1}^{N} w_i^{\text{prior}} = 1/N$$

$$\text{posterior PDF: } p(\mathbf{x}|\mathbf{y}) = \sum_{i=1}^{N} w_i^{\text{post}} \delta(\mathbf{x}-\mathbf{x}_i), \qquad w_i^{\text{post}} = \frac{w_i^{\text{prior}} p(\mathbf{y}|\mathbf{x}_i)}{\sum_{i=1}^{N} w_i^{\text{prior}} p(\mathbf{y}|\mathbf{x}_i)}$$



Particle filters (cont)

Standard PF is degenerate (weight tends to accumulate for one particle)

- 'Resampling' – still a problem for lots of obs.
  - 'Localized PF' – weights become a function of position.
    - 'Proposal density' – freedom to sample particles from another distribution to try to even out the weights.
    - Localized adaptive PF (Potthast et al. 2019).
  - Proposal density
    - Sample particles from a chosen proposal density (to be closer to the observations).
    - Adjust the weights to compensate.
  - Particle flow filter.

Which method is right for you?

- Do you have PDFs that are highly non-Gaussian for high-probability regions (e.g. multi-modal)? Yes: PF. No: KF, Var, EnKF, EnVar, Hybrid.
- Is n large (e.g. $n > 100$)? Yes: not the KF!
- Is only a first moment of the posterior required? Yes: Var, EnVar, Hybrid. No: EnKF, PF.
- Is the prior error cov matrix quasi-static? Yes: Var. No: KF, EnKF, EnVar, Hybrid, PF.
- Are linearized/adjoint models available? Yes: KF, 4DVar, En4DVar, Hybrid-En4DVar. No: 3DVar (still need linear/adjoint of $\mathscr{H}$), EnKF, 4DEnVar, Hybrid-4DEnVar, PF.
- How many model runs can you afford per cycle? 1: Hmm. Dozens: Var, EnKF (loc), EnVar (loc), Hybrid (loc), PF. Large number: (less need for loc).
- Is easy parallelization crucial? Yes: EnKF,PF. No: KF, Var, EnVar, Hybrid.
- Do you already have a model, but want minimal development time of the DA system? Yes: EnKF via PDAF/DART/DAPPER/JEDI.

PF=Particle Filter (various flavours); **KF**=Kalman Filter; **EnVar**=Ensemble-Variational (includes En4DVar and 4DEnVar); **Var**=Variational (includes 4DVar, 3DVar); **loc**=localization required; **hybrid** (includes hybrid-En4DVar, hybrid-4DEnVar), **PDAF**=Parallel Data Assimilation Framework (pdaf.awi.de/trac/wiki); **DART**=Data Assimilation Research Testbed (www.image.ucar.edu/DAReS/DART), **DAPPER**=Data assimilation package in Python for experimental research (www.nersc.no/news/data-assimilation-package-python-experimental-research-dapper), **JEDI**=Joint Effort for Data assimilation Integration (www.jcsda.org/jcsda-project-jedi). Existing and new methods are constantly being developed.