What to use when?
With a brief recap

Lecturer: Ross Bannister
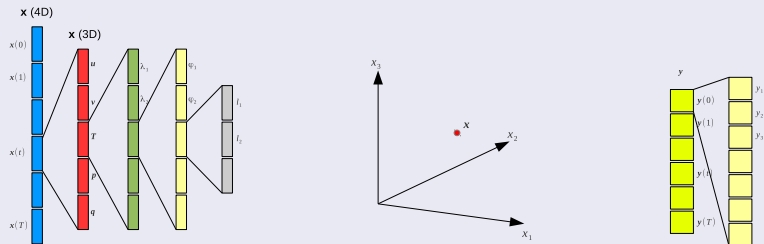
NCEO, Dept. of Meteorology, Univ. of Reading

5th–8th March 2019, Univ. of Reading
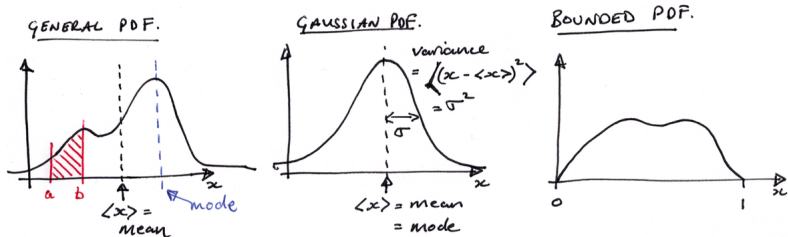
- To blend information from models and observations.
  - State/parameter estimation (some kind of 'optimal' blending).
  - The posterior PDF or reduced moments of it.

State vector and observation vectors

Probability Density Functions

GENERAL PDF.   GAUSSIAN PDF.   BOUNDED PDF.

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

Form of a Gaussian:

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

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

A covariance matrix describes the second moment of a PDF.

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

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

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

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

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

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 is Gaussian distributed:

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

$$
\propto \quad \exp(\text{non-quadratic in } x)
$$

linearize ...

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

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

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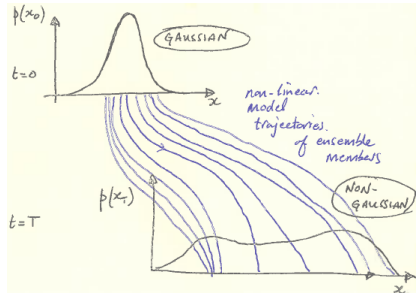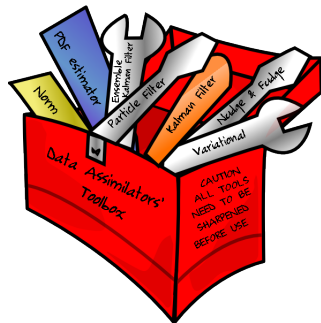

Again non-linearity leads to non-Gaussianity.

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

- Data insertion/nudging
- 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

▷ 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.

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

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

▷ 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 constraints (e.g. weak constraint for model error); ▷ strong theoretical basis; ▷ $\mathscr{M}$ and $\mathscr{H}$ can be non-linear; ▷ usually incremental formulation used

$$
\begin{aligned}
\text{forecast state: } \mathbf{x}_0^{\mathrm{f}} &= \mathscr{M}(\mathbf{x}_{-T}^{\mathrm{a}}) \\
\text{analysis state: } \mathbf{x}_0^{\mathrm{a}} &= \mathbf{x}_0^{\mathrm{f}} + \underset{\delta\mathbf{x_0}}{\arg\min} J[\delta\mathbf{x}_0]
\end{aligned}
$$

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

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, $\mathscr{M}_t(\mathbf{x}_t) = \mathbf{x}_t$).

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

- Assumes Gaussian prior and observations.
- 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.
- Difficult to develop (time and expertise).
- Difficult to parallelize.

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.

$$\mathbf{x}_t^{(i),\mathrm{f}} = \mathscr{M}_t(\mathbf{x}_{t-1}^{(i),\mathrm{a}}) + \beta^{(i)}$$

$$n \times N: \quad \mathbf{X}_t^{\prime\mathrm{f}} = \left(\mathbf{x}_t^{(1),\mathrm{f}} - \bar{\mathbf{x}}_t^{\mathrm{f}} \quad \cdots \quad \mathbf{x}_t^{(N),\mathrm{f}} - \bar{\mathbf{x}}_t^{\mathrm{f}} \right)$$

$$p \times N: \quad \mathbf{Y}_t' = \left(\mathscr{H}_t(\mathbf{x}_t^{(1),\mathrm{f}}) - \mathscr{H}(\bar{\mathbf{x}}_t^{\mathrm{f}}) \quad \cdots \quad \mathscr{H}_t(\mathbf{x}_t^{(N),\mathrm{f}}) - \mathscr{H}_t(\bar{\mathbf{x}}_t^{\mathrm{f}}) \right)$$

$$n \times N: \quad \mathbf{X}_t^{\prime\mathrm{a}} = \left(\mathbf{x}_t^{(1),\mathrm{a}} - \bar{\mathbf{x}}_t^{\mathrm{a}} \quad \cdots \quad \mathbf{x}_t^{(N),\mathrm{a}} - \bar{\mathbf{x}}_t^{\mathrm{a}} \right)$$

Stochastic EnKF

$$\mathbf{x}_t^{(i)\mathrm{a}} = \mathbf{x}_t^{(i)\mathrm{f}} + \mathbf{K}_t \left(\mathbf{y}_t + \varepsilon_y^{(i)} - \mathscr{H}_t(\mathbf{x}_t^{(i)\mathrm{f}}) \right)$$

$$\mathbf{K}_t = \mathbf{X}_t^{\prime\mathrm{f}} \mathbf{Y}_t^{\prime\mathrm{T}} \left(\mathbf{Y}_t' \mathbf{Y}_t^{\prime\mathrm{T}} + (N-1)\mathbf{R}_t \right)^{-1}$$

Ensemble Transform KF

$$\bar{\mathbf{x}}_t^{\mathrm{a}} = \bar{\mathbf{x}}_t^{\mathrm{f}} + \mathbf{K}_t \left(\mathbf{y}_t - \mathscr{H}_t(\bar{\mathbf{x}}_t^{\mathrm{f}}) \right)$$

$$\mathbf{X}_t^{\prime\mathrm{a}} = \mathbf{X}_t^{\prime\mathrm{f}} \mathbf{T}_t$$

$$\mathbf{K}_t = \mathbf{X}_t^{\prime\mathrm{f}} \mathbf{T}_t \mathbf{T}_t^{\mathrm{T}} \mathbf{Y}_t^{\prime\mathrm{T}} \mathbf{R}_t^{-1}$$

$$\mathbf{T}_t = \left(\mathbf{I} + \mathbf{Y}_t^{\prime\mathrm{T}} \mathbf{R}_t^{-1} \mathbf{Y}_t' \right)^{-1/2}$$

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: $\mathbf{P}_t^{\mathrm{f}} \circ \boldsymbol{\Omega}$ (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, . . .)

▷ As variational DA, but where $\mathbf{B} \rightarrow \mathbf{X}'^{\,f}_0 \mathbf{X}'^{\,f\,T}_0/(N-1)$ 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:

$$\mathbf{x}^a_0 = \mathbf{x}^f_0 + \mathbf{X}'^f \chi_{\text{ens}} \qquad \chi_{\text{ens}} \text{ is an } N\text{-element matrix}$$

$$J[\chi_{\text{ens}}] = \frac{1}{2}\chi^T_{\text{ens}}\chi_{\text{ens}} + \frac{1}{2}\sum_{t=0}^{T}(\mathbf{y}_t - \mathscr{H}_t(\mathbf{x}_t))^T \mathbf{R}^{-1}_t (\bullet)$$

$$\text{subject to } \delta\mathbf{x}_{t+1} = \mathbf{M}_t(\delta\mathbf{x}_t) \quad \text{and } \delta\mathbf{x}_0 = \mathbf{X}'^f \chi_{\text{ens}}$$

$$\mathbf{x}^f_{t+1} = \mathscr{M}_t\left(\mathbf{x}^f_t\right)$$

- Assumes Gaussian prior and observations.
- $\mathbf{P}^f$-matrix $= \mathbf{X}'^{\,f}_0 \mathbf{X}'^{\,f\,T}_0/(N-1)$, rank deficient ($N < n$).
- Needs localization and a separate parallel ensemble.
- 4DEnVar 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.

As variational DA, but where

$$\mathbf{B} \rightarrow (1-\beta)\mathbf{B} + \beta \mathbf{X}'^{\mathrm{f}}_0 \mathbf{X}'^{\mathrm{f}\,\mathrm{T}}_0 /(N-1)$$

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

Traditional 4DVar with control variable transform:

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

subject to $\delta\mathsf{x}_{t+1} = \mathsf{M}_t\left(\delta\mathsf{x}_t\right), \quad \delta\mathsf{x}_t = \mathsf{U}\chi_{\mathsf{B}}$

Hybrid-En4DVar:

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

subject to $\delta\mathsf{x}_{t+1} = \mathsf{M}_t\left(\delta\mathsf{x}_t\right), \quad \delta\mathsf{x}_t = \sqrt{1-\beta}\,\mathsf{U}\chi_{\mathsf{B}} + \sqrt{\beta}\,\mathsf{X}'^{\mathsf{f}}\chi_{\mathrm{ens}}$

- 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:

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

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

- 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).

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 lin/adjoint of $\mathcal{H}$), EnKF, 4DEnVar, Hybrid-4DEnVar, PF.
- How many model runs can you afford per cycle? ~ 1: Var. Dozens: EnKF (loc), EnVar (loc), Hybrid (loc), PF. Large number: as ← (no 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.

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).
Existing and new methods are constantly being developed.