Kalman Filter and Ensemble Kalman Filter

Stochastic Data Assimilation for Linear and Nonlinear Models

Christophe Prud'homme

December 18, 2025

University of Strasbourg

Outline

Introduction and Motivation

Context: Where we are in the course

What we have covered

- Reduced Basis (RB) methods for parametrized PDEs
- Empirical Interpolation Method (EIM) for non-affine problems
- Generalized EIM (GEIM) for field reconstruction
- Parametrized Background Data-Weak (PBDW) for state estimation

What comes next

Sequential data assimilation: combining models and observations over time using stochastic filtering approaches.

Key idea: PBDW and GEIM provide *static* state estimation. Kalman filters extend this to *dynamic* systems with time evolution.

Why sequential data assimilation?

In many applications, we need to:

1. **Track** a time-evolving system (weather, ocean, structures...)
2. **Predict** future states based on current knowledge
3. **Update** estimates as new observations become available
4. **Quantify** uncertainty in our predictions

Applications

- Weather forecasting and reanalysis
- Ocean state estimation
- Structural health monitoring
- Digital twins for engineering systems
- Parameter estimation in inverse problems

Dynamical systems + ROM toolkit

Dynamical systems we target

Generic evolution model (discrete-time):

$$\mathbf{x}_{k+1} = \mathcal{M}_k(\mathbf{x}_k, \boldsymbol{\mu}_k) + \mathbf{w}_k, \qquad \mathbf{y}_k = \mathcal{H}_k(\mathbf{x}_k) + \boldsymbol{\epsilon}_k^{\mathrm{o}}$$

- Covers ODEs and time-discretized PDEs with parameters $\boldsymbol{\mu}$
- $\mathbf{w}_k$: model/process error; $\boldsymbol{\epsilon}_k^{\mathrm{o}}$: sensor error
- Objectives: **track** the state, **estimate** parameters, and **predict** ahead
- Toy-but-classic testbed: **Lorenz-63** (chaotic attractor used in meteorology education)

ROM tools for dynamics (RB, GEIM, PBDW)

Offline: build low-dimensional structures

- RB/POD spaces from time-dependent snapshots of the full model
- (G)EIM for non-affine terms and for selecting **informative sensors**
- Certified error estimators provide trust in the reduced surrogate

Online: exploit these structures sequentially

- Fast forecast in reduced space ($N \ll n$) for each ensemble member
- Observation operators naturally derived from GEIM functionals
- PBDW gives a deterministic assimilation map; EnKF brings stochastic UQ

Time-dependent PBDW (Benaceur, A. Ern)

Idea: adapt PBDW to dynamical systems by updating reduced spaces and reconstructions over time.

- Build time-indexed reduced spaces V_n^k via POD-greedy on transient snapshots
- Select sensors with a GEIM-like procedure to stabilize reconstructions at each t_k
- At every time step: solve a PBDW problem in V_n^k using the current measurements y_m^k

Positioning:

- Deterministic alternative to KF/EnKF when certified bounds are needed
- Provides informed backgrounds and sensor layouts that can seed ensemble methods

From Static to Sequential Estimation

Bridging RB/EIM vocabulary to Data Assimilation

RB/EIM/GEIM	Data Assimilation	Meaning
$u(\boldsymbol{\mu}) \in V$	$\mathbf{x} \in \mathbb{R}^n$	State (field or vector)
Parameter $\boldsymbol{\mu} \in \mathcal{D}$	Time t_k or param.	What varies
Reduced space V_N	Background/prior space	Model knowledge
RB solution $u_N(\boldsymbol{\mu})$	Background $\mathbf{x}^{\mathrm{b}}$	Model prediction
Sensors $\{\sigma_m\}$	Obs. operator $\mathbf{H}$	How we observe
Measurements $y_m = \sigma_m(u)$	Observations $\mathbf{y}$	What we measure
GEIM/PBDW estimate	Analysis $\mathbf{x}^{\mathrm{a}}$	Best estimate
A posteriori error	Analysis covariance $\mathbf{P}^{\mathrm{a}}$	Uncertainty

Key parallel:

- PBDW: $u^* = \underset{z \in u_{\mathrm{bg}} + V_N}{\arg\min} \|z - u_{\mathrm{bg}}\|^2$ s.t. $\sigma_m(z) = y_m$
- BLUE: $\mathbf{x}^{\mathrm{a}} = \arg\min_{\mathbf{x}} \|\mathbf{x} - \mathbf{x}^{\mathrm{b}}\|^2_{\mathbf{B}^{-1}} + \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2_{\mathbf{R}^{-1}}$

Notation: $\|\mathbf{x}\|^2_{\mathbf{A}^{-1}} := \mathbf{x}^\top \mathbf{A}^{-1}\mathbf{x}$ is the **Mahalanobis norm** (weighted by precision $\mathbf{A}^{-1}$).

The key objects in Data Assimilation

1. **The state $\mathbf{x} \in \mathbb{R}^n$** (or $u \in V$):

    - In RB: the solution field $u(\cdot; \boldsymbol{\mu})$ discretized on a mesh
    - In DA: temperature, velocity, concentration... at all grid points
    - Dimension n can be 10^6–10^9 for realistic problems

2. **The observation operator $\mathbf{H}$** $: \mathbb{R}^n \to \mathbb{R}^m$:

    - Maps the full state to what sensors can measure
    - In GEIM: $\mathbf{H} = [\sigma_1, \ldots, \sigma_M]^\top$ (linear functionals)
    - Examples: point evaluation, spatial averages, integrals
    - Usually $m \ll n$ (sparse observations)

3. **The observations $\mathbf{y} \in \mathbb{R}^m$**:

    - Actual measurements: $\mathbf{y} = \mathbf{H}\mathbf{x}^{\mathrm{true}} + \boldsymbol{\epsilon}^{\mathrm{o}}$
    - Include measurement noise $\boldsymbol{\epsilon}^{\mathrm{o}}$

The key objects (continued)

4. **The background/prior $\mathbf{x}^{\mathrm{b}}$**:
   - Our best guess *before* seeing observations
   - In PBDW: comes from the reduced model manifold V_N
   - In KF: comes from model propagation (forecast)

5. **Covariance matrices** (new concept!):
   - $\mathbf{B}$ (or $\mathbf{P}^{\mathrm{f}}$): background/forecast error covariance
     - Encodes: "how much do we trust the model?"
     - Encodes spatial correlations of errors
   - $\mathbf{R}$: observation error covariance
     - Encodes: "how much do we trust the sensors?"
     - Often diagonal (uncorrelated sensor noise)

6. **The analysis $\mathbf{x}^{\mathrm{a}}$**: optimal combination of background + observations

Bayesian vocabulary: Prior and Posterior

Prior $p(\mathbf{x})$: belief about the state **before** seeing observations

- In DA: $\mathbf{x}^{\mathrm{b}} \sim \mathcal{N}(\mathbf{x}^{\mathrm{b}}, \mathbf{B})$ — the background/forecast
- Source: model prediction, climatology, previous analysis
- Analogy: "The weather model predicts 20°C tomorrow"

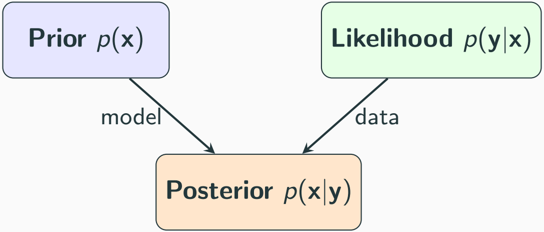
Posterior $p(\mathbf{x}|\mathbf{y})$: updated belief **after** incorporating observations

- In DA: $\mathbf{x}^{\mathrm{a}} \sim \mathcal{N}(\mathbf{x}^{\mathrm{a}}, \mathbf{P}^{\mathrm{a}})$ — the analysis
- Combines prior knowledge with observational evidence
- Analogy: "Given thermometer reads 23°C, I now estimate 22.4°C"

Likelihood $p(\mathbf{y}|\mathbf{x})$: probability of observing $\mathbf{y}$ if state were $\mathbf{x}$

- In DA: $\mathbf{y}|\mathbf{x} \sim \mathcal{N}(\mathbf{H}\mathbf{x}, \mathbf{R})$
- "How well does candidate state $\mathbf{x}$ explain the data?"

Prior $\to$ Posterior: the Bayesian update



$$\underbrace{p(\mathbf{x}|\mathbf{y})}_{\text{posterior}} \propto \underbrace{p(\mathbf{y}|\mathbf{x})}_{\text{likelihood}} \times \underbrace{p(\mathbf{x})}_{\text{prior}}$$

Connection to PBDW:

PBDW (deterministic)	**Bayesian DA (stochastic)**
Background $u_{\mathrm{bg}} \in V_N$	Prior mean $\mathbf{x}^{\mathrm{b}}$
RB space V_N	Prior covariance structure
PBDW estimate u^*	Posterior mean $\mathbf{x}^{\mathrm{a}}$
A posteriori error bound	Posterior covariance $\mathbf{P}^{\mathrm{a}}$

How to obtain the covariance matrices?

Observation error covariance $\mathbf{R}$:

- Often the **easiest** to estimate
- From **sensor specifications**: manufacturer provides accuracy
- From **calibration experiments**: repeated measurements of known state
- From **representativeness error**: mismatch between point measurement and grid-cell average
- Typically **diagonal**: $\mathbf{R} = \mathrm{diag}(s_1^2, \dots, s_m^2)$ where s_i = std. dev. of sensor i

Background error covariance $\mathbf{B}$ (much harder!):

- We don't know the true state $\mathbf{x}^{\mathrm{true}}$, so we can't compute $\mathbf{B} = \mathbb{E}[(\mathbf{x}^{\mathrm{b}} - \mathbf{x}^{\mathrm{true}})(\mathbf{x}^{\mathrm{b}} - \mathbf{x}^{\mathrm{true}})^\top]$ directly
- Dimension: $n \times n$ with $n \sim 10^6$–10^9 $\rightarrow$ cannot store explicitly!

Practical approaches for B

1. **NMC method** (National Meteorological Center):
    - Use forecast differences as proxy: $\mathbf{x}^{48h} - \mathbf{x}^{24h}$
    - Idea: differences between forecasts approximate forecast errors
    - Estimate: $\mathbf{B} \approx \frac{1}{K} \sum_{k=1}^{K} (\mathbf{x}_k^{48h} - \mathbf{x}_k^{24h})(\mathbf{x}_k^{48h} - \mathbf{x}_k^{24h})^\top$
2. **Innovation statistics** (Desroziers et al., 2005):
    - Use observation-minus-background: $\mathbf{d}^{\mathrm{b}} = \mathbf{y} - \mathbf{H}\mathbf{x}^{\mathrm{b}}$
    - Consistency relation: $\mathbb{E}[\mathbf{d}^{\mathrm{b}}(\mathbf{d}^{\mathrm{b}})^\top] = \mathbf{H}\mathbf{B}\mathbf{H}^\top + \mathbf{R}$
3. **Ensemble methods** (used in EnKF):
    - Run an ensemble of N_e forecasts: $\{\mathbf{x}^{\mathrm{f},(i)}\}_{i=1}^{N_e}$
    - Sample covariance: $\mathbf{P}^{\mathrm{f}} \approx \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{x}^{(i)} - \bar{\mathbf{x}})^\top$
    - **Key advantage**: never form $\mathbf{P}^{\mathrm{f}}$ explicitly!

Structure of B: correlation models

Since **B** is too large to store, we model its **structure**:

$$B_{ij} = s_i \, s_j \, \rho(d_{ij})$$

where s_i = standard deviation at point i, ρ = correlation function, d_{ij} = distance.

Note: We use s for std. dev. to distinguish from GEIM functionals σ_m. In KF notation, **B** here plays the role of the forecast covariance $\mathbf{P}^{\mathrm{f}}$.

Common correlation models:

- Gaussian: $\rho(d) = \exp\left(-\frac{d^2}{2L^2}\right)$
- SOAR (Second-Order Auto-Regressive): $\rho(d) = \left(1 + \frac{d}{L}\right) \exp\left(-\frac{d}{L}\right)$
- Matérn family (flexible smoothness)

Correlation length L:

- Controls how far information spreads from observations
- Estimated from data or physical considerations

From PBDW to BLUE: the role of uncertainty

PBDW (deterministic):

$$u^* = \arg \min_{z \in u_{\mathrm{bg}} + V_N} \|z - u_{\mathrm{bg}}\|_V^2 \quad \text{s.t.} \quad \sigma_m(z) = y_m$$

- Observations are **exact constraints**
- Background error quantified by $\inf_{v \in V_N} \|u - v\|_V$
- No observation noise model

BLUE/3D-Var (stochastic):

$$\mathbf{x}^{\mathrm{a}} = \arg\min_{\mathbf{x}} \underbrace{\|\mathbf{x} - \mathbf{x}^{\mathrm{b}}\|_{\mathbf{B}^{-1}}^2}_{\text{background term}} + \underbrace{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_{\mathbf{R}^{-1}}^2}_{\text{observation term}}$$

- Observations are **soft constraints** (weighted by $\mathbf{R}^{-1}$)
- Both model and observations have uncertainty
- Weights $\mathbf{B}^{-1}, \mathbf{R}^{-1}$ = inverse covariances = precisions

Static linear Bayesian estimation (BLUE)

Consider a linear observation model

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}^{\mathrm{o}}, \quad \boldsymbol{\epsilon}^{\mathrm{o}} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}),$$

with a Gaussian prior on the state

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

Bayes' theorem gives the posterior:

$$p(\mathbf{x}|\mathbf{y}) \propto p(\mathbf{y}|\mathbf{x})\, p(\mathbf{x})$$

For Gaussian distributions, the posterior is also Gaussian with:

$$\mathbf{x}^{\mathrm{a}} = \mathbf{x}^{\mathrm{b}} + \mathbf{K}\big(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathrm{b}}\big), \qquad \mathbf{K} = \mathbf{B}\mathbf{H}^\top\big(\mathbf{H}\mathbf{B}\mathbf{H}^\top + \mathbf{R}\big)^{-1}.$$

Bayes in action: a simple example

Situation: Estimate temperature x at one location.

- Model predicts: $x^{\mathrm{b}} = 20^\circ\mathrm{C}$ with std. dev. $s_b = 2^\circ\mathrm{C}$
- Sensor measures: $y = 23^\circ\mathrm{C}$ with std. dev. $s_o = 1^\circ\mathrm{C}$

Bayesian analysis (1D Kalman):

$$x^{\mathrm{a}} = x^{\mathrm{b}} + K(y - x^{\mathrm{b}}), \quad K = \frac{s_b^2}{s_b^2 + s_o^2} = \frac{4}{4+1} = 0.8$$

$$x^{\mathrm{a}} = 20 + 0.8 \times (23 - 20) = 22.4^\circ\mathrm{C}$$

Interpretation:

- Analysis is closer to observation (sensor more accurate)
- If $s_o = 4^\circ\mathrm{C}$ instead: $K = 0.2$, $x^{\mathrm{a}} = 20.6^\circ\mathrm{C}$ (trust model more)
- Posterior uncertainty: $s_a^2 = (1-K)s_b^2 = 0.2 \times 4 = 0.8\,(^\circ\mathrm{C})^2$

$\Rightarrow$ **Both estimate and uncertainty are updated!**

BLUE: Derivation (1/2)

Optimality criterion

Find $\mathbf{x}^{\mathrm{a}}$ that minimizes the analysis error variance:

$$\min_{\mathbf{x}^{\mathrm{a}}} \mathbb{E}\left[\|\mathbf{x}^{\mathrm{a}} - \mathbf{x}^{\mathrm{t}}\|^2\right] = \min_{\mathbf{x}^{\mathrm{a}}} \mathrm{tr}(\mathbf{A})$$

where $\mathbf{A} = \mathrm{Cov}(\mathbf{x}^{\mathrm{a}} - \mathbf{x}^{\mathrm{t}})$.

Assume a linear estimator of the form:

$$\mathbf{x}^{\mathrm{a}} = \mathbf{L}\mathbf{x}^{\mathrm{b}} + \mathbf{K}\mathbf{y}$$

Unbiasedness constraint: $\mathbb{E}[\mathbf{x}^{\mathrm{a}}] = \mathbb{E}[\mathbf{x}^{\mathrm{t}}]$

Since $\mathbf{y} = \mathbf{H}\mathbf{x}^{\mathrm{t}} + \boldsymbol{\epsilon}^{\mathrm{o}}$ and $\mathbf{x}^{\mathrm{b}} = \mathbf{x}^{\mathrm{t}} + \boldsymbol{\epsilon}^{\mathrm{b}}$:

$$\mathbb{E}[\mathbf{x}^{\mathrm{a}}] = \mathbf{L}\mathbf{x}^{\mathrm{t}} + \mathbf{K}\mathbf{H}\mathbf{x}^{\mathrm{t}} \implies \mathbf{L} = \mathbf{I} - \mathbf{K}\mathbf{H}$$

BLUE: Derivation (2/2)

With $\mathbf{L} = \mathbf{I} - \mathbf{KH}$, the analysis error is:

$$\boldsymbol{\epsilon}^{\mathrm{a}} = \mathbf{x}^{\mathrm{a}} - \mathbf{x}^{\mathrm{t}} = (\mathbf{I} - \mathbf{KH})\boldsymbol{\epsilon}^{\mathrm{b}} - \mathbf{K}\boldsymbol{\epsilon}^{\mathrm{o}}$$

The analysis error covariance:

$$\mathbf{A} = (\mathbf{I} - \mathbf{KH})\mathbf{B}(\mathbf{I} - \mathbf{KH})^\top + \mathbf{KRK}^\top$$

Minimizing $\mathrm{tr}(\mathbf{A})$ with respect to $\mathbf{K}$:

$$\frac{\partial \mathrm{tr}(\mathbf{A})}{\partial \mathbf{K}} = 0 \implies \mathbf{K} = \mathbf{BH}^\top(\mathbf{HBH}^\top + \mathbf{R})^{-1}$$

Result: BLUE formulas

$$\mathbf{x}^{\mathrm{a}} = \mathbf{x}^{\mathrm{b}} + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathrm{b}}), \qquad \mathbf{A} = (\mathbf{I} - \mathbf{KH})\mathbf{B}$$

Variational interpretation: 3D-Var

The BLUE is equivalent to the solution of the optimization problem:

$$\mathbf{x}^{\mathrm{a}} = \arg\min_{\mathbf{x}} \mathcal{J}(\mathbf{x})$$

where the **cost function** is:

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\left\|\mathbf{x} - \mathbf{x}^{\mathrm{b}}\right\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\left\|\mathbf{y} - \mathbf{H}\mathbf{x}\right\|^2_{\mathbf{R}^{-1}}$$

Interpretation:

- First term: penalize departure from background (prior)
- Second term: penalize misfit to observations
- Weights $\mathbf{B}^{-1}$, $\mathbf{R}^{-1}$: inverse covariances (precisions)

Connection to PBDW: similar structure with model-based prior and observation constraint!

From static to sequential estimation

- Static BLUE / 3D-Var:
    - One shot assimilation at a given time.
    - No explicit time propagation of error covariance.
- Many applications are **time-dependent**:
    - PDEs with time (parabolic, hyperbolic),
    - control and tracking problems,
    - forecasting in geophysics, meteorology, etc.
- We need a **sequential** formulation:
    - propagate the state in time,
    - propagate the uncertainty in time,
    - assimilate new observations as they arrive.

Key insight

The Kalman filter applies BLUE recursively in time, using the previous analysis as the new background.

The Kalman Filter

Linear Gaussian state-space model

We consider a discrete-time linear dynamical system:

State equation (dynamics)

$$\mathbf{x}_{k+1} = \mathbf{M}_k \mathbf{x}_k + \mathbf{w}_k, \qquad \mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$$

Observation equation

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{x}_k + \boldsymbol{\epsilon}_k^{\mathrm{o}}, \qquad \boldsymbol{\epsilon}_k^{\mathrm{o}} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$$

Assumptions:

- Initial state: $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{x}_0^{\mathrm{b}}, \mathbf{P}_0^{\mathrm{b}})$
- Model error $\mathbf{w}_k$ and observation error $\boldsymbol{\epsilon}_k^{\mathrm{o}}$ are independent
- All random variables are jointly Gaussian
- Errors are uncorrelated in time: $\mathbb{E}[\mathbf{w}_j \mathbf{w}_k^{\top}] = \mathbf{Q}_k \delta_{jk}$

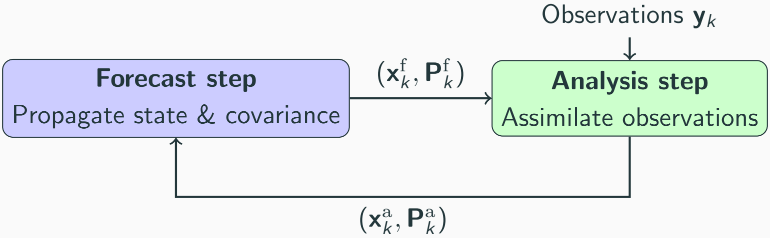
Notation and terminology

At each time step k:

Notation	Description
$\mathbf{x}_k^{\mathrm{f}}$	**Forecast** (prior/background) state
$\mathbf{P}_k^{\mathrm{f}}$	Forecast error covariance
$\mathbf{x}_k^{\mathrm{a}}$	**Analysis** (posterior) state
$\mathbf{P}_k^{\mathrm{a}}$	Analysis error covariance
$\mathbf{y}_k$	Observation vector
$\mathbf{H}_k$	Observation operator (linear)
$\mathbf{R}_k$	Observation error covariance
$\mathbf{M}_k$	Model (state transition) operator
$\mathbf{Q}_k$	Model error covariance
$\mathbf{K}_k$	Kalman gain matrix

Note: "forecast" = "background" = "prior" in DA literature.

Kalman Filter: the two-step cycle



1. **Forecast**: use model to predict state and uncertainty
2. **Analysis**: correct forecast using observations (BLUE update)
3. **Repeat**: analysis becomes background for next cycle

Kalman Filter: forecast step

Given $(\mathbf{x}_k^{\mathrm{a}}, \mathbf{P}_k^{\mathrm{a}})$, the **forecast** (or prediction) to time $k+1$ is:

State forecast

$$\mathbf{x}_{k+1}^{\mathrm{f}} = \mathbf{M}_k \mathbf{x}_k^{\mathrm{a}}$$

Covariance forecast

$$\mathbf{P}_{k+1}^{\mathrm{f}} = \mathbf{M}_k \mathbf{P}_k^{\mathrm{a}} \mathbf{M}_k^{\top} + \mathbf{Q}_k$$

Interpretation:

- Mean: propagated through the linear model
- Covariance: propagated via similarity transform + model error
- Uncertainty *grows* during forecast (unless $\mathbf{Q}_k = 0$)

Kalman Filter: analysis step

When observations $\mathbf{y}_{k+1}$ are available, the **analysis** step computes:

Kalman gain

$$\mathbf{K}_{k+1} = \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\top}_{k+1}\left(\mathbf{H}_{k+1}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\top}_{k+1} + \mathbf{R}_{k+1}\right)^{-1}$$

State update

$$\mathbf{x}^{\mathrm{a}}_{k+1} = \mathbf{x}^{\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}\underbrace{\left(\mathbf{y}_{k+1} - \mathbf{H}_{k+1}\mathbf{x}^{\mathrm{f}}_{k+1}\right)}_{\text{innovation } \mathbf{d}_{k+1}}$$

Covariance update

$$\mathbf{P}^{\mathrm{a}}_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}^{\mathrm{f}}_{k+1}$$

Interpretation of the Kalman gain

The Kalman gain $\mathbf{K}_k$ balances:

- the **forecast uncertainty** $\mathbf{P}_k^{\mathrm{f}}$
- the **observation uncertainty** $\mathbf{R}_k$

Limiting cases:

- If $\mathbf{R}_k \to 0$ (perfect observations): in the observed subspace the analysis collapses to $\mathbf{y}_k$ (when $\mathbf{H}_k$ is invertible there)
- If $\mathbf{P}_k^{\mathrm{f}} \to 0$ (perfect forecast): $\mathbf{K}_k \to 0$, analysis $\to$ forecast
- If $\mathbf{R}_k \gg \mathbf{H}_k \mathbf{P}_k^{\mathrm{f}} \mathbf{H}_k^\top$: trust model more
- If $\mathbf{R}_k \ll \mathbf{H}_k \mathbf{P}_k^{\mathrm{f}} \mathbf{H}_k^\top$: trust observations more

Key property: $\mathbf{P}_k^{\mathrm{a}} \leq \mathbf{P}_k^{\mathrm{f}}$ (in matrix sense)

$\Rightarrow$ Assimilating observations always *reduces* uncertainty!

Kalman Filter: Algorithm

Algorithm 1 Kalman Filter

Initialize: $\mathbf{x}_0^{\mathrm{a}} = \mathbf{x}_0^{\mathrm{b}}$, $\mathbf{P}_0^{\mathrm{a}} = \mathbf{P}_0^{\mathrm{b}}$

for $k = 0, 1, 2, \ldots$ **do**

    Forecast step:

        $\mathbf{x}_{k+1}^{\mathrm{f}} = \mathbf{M}_k \mathbf{x}_k^{\mathrm{a}}$

        $\mathbf{P}_{k+1}^{\mathrm{f}} = \mathbf{M}_k \mathbf{P}_k^{\mathrm{a}} \mathbf{M}_k^{\top} + \mathbf{Q}_k$

    Analysis step (if observations $\mathbf{y}_{k+1}$ available):

        $\mathbf{K}_{k+1} = \mathbf{P}_{k+1}^{\mathrm{f}} \mathbf{H}_{k+1}^{\top} (\mathbf{H}_{k+1} \mathbf{P}_{k+1}^{\mathrm{f}} \mathbf{H}_{k+1}^{\top} + \mathbf{R}_{k+1})^{-1}$

        $\mathbf{x}_{k+1}^{\mathrm{a}} = \mathbf{x}_{k+1}^{\mathrm{f}} + \mathbf{K}_{k+1} (\mathbf{y}_{k+1} - \mathbf{H}_{k+1} \mathbf{x}_{k+1}^{\mathrm{f}})$

        $\mathbf{P}_{k+1}^{\mathrm{a}} = (\mathbf{I} - \mathbf{K}_{k+1} \mathbf{H}_{k+1}) \mathbf{P}_{k+1}^{\mathrm{f}}$

end for

Optimality of the Kalman Filter

Theorem (Kalman, 1960)

For a linear Gaussian state-space model, the Kalman filter provides:

1. *The* **minimum variance** *unbiased estimator*
2. *The* **maximum likelihood** *estimator*
3. *The* **exact posterior mean** $\mathbb{E}[\mathbf{x}_k|\mathbf{y}_{1:k}]$

Proof sketch: By induction, if $p(\mathbf{x}_k|\mathbf{y}_{1:k})$ is Gaussian, then $p(\mathbf{x}_{k+1}|\mathbf{y}_{1:k+1})$ is also Gaussian with mean and covariance given by the KF formulas.

Important

Optimality holds **only** for:

- Linear dynamics and observations
- Gaussian noise distributions

Alternative formulations of the analysis

Several equivalent forms for the covariance update:

Joseph form (numerically stable):

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

Sherman-Morrison-Woodbury form:

$$(\mathbf{P}_k^{\mathrm{a}})^{-1} = (\mathbf{P}_k^{\mathrm{f}})^{-1} + \mathbf{H}_k^\top\mathbf{R}_k^{-1}\mathbf{H}_k$$

Alternative Kalman gain:

$$\mathbf{K}_k = \left((\mathbf{P}_k^{\mathrm{f}})^{-1} + \mathbf{H}_k^\top\mathbf{R}_k^{-1}\mathbf{H}_k\right)^{-1}\mathbf{H}_k^\top\mathbf{R}_k^{-1}$$

Choice depends on dimensions:

- Standard form: invert $m \times m$ matrix ($m = \#$ observations)
- SMW form: invert $n \times n$ matrix ($n =$ state dimension)

Limitations of the Kalman Filter

1. **Computational cost**:
    - Storage: $\mathbf{P}_k \in \mathbb{R}^{n \times n}$ ($O(n^2)$ memory)
    - Operations: matrix products, inversions ($O(n^3)$ per step)
    - Prohibitive for high-dimensional systems (e.g., $n \sim 10^6$)
2. **Linearity assumption**:
    - Real models are often nonlinear: $\mathbf{x}_{k+1} = \mathcal{M}_k(\mathbf{x}_k) + \mathbf{w}_k$
    - Observation operators may be nonlinear: $\mathbf{y}_k = \mathcal{H}_k(\mathbf{x}_k) + \boldsymbol{\epsilon}_k^{\mathrm{o}}$
3. **Gaussianity assumption**:
    - Errors may have heavy tails, be bounded, etc.
    - Nonlinear transformations destroy Gaussianity

Solutions: Extended KF, Unscented KF, Ensemble KF, Particle filters

Extended Kalman Filter (EKF)

For nonlinear models $\mathbf{x}_{k+1} = \mathcal{M}_k(\mathbf{x}_k) + \mathbf{w}_k$:

Idea

Linearize around the current estimate:

$$\mathcal{M}_k(\mathbf{x}) \approx \mathcal{M}_k(\mathbf{x}_k^{\mathrm{a}}) + \mathbf{M}_k(\mathbf{x} - \mathbf{x}_k^{\mathrm{a}})$$

where $\mathbf{M}_k = \frac{\partial \mathcal{M}_k}{\partial \mathbf{x}}\Big|_{\mathbf{x}_k^{\mathrm{a}}}$ is the **tangent linear model**.

EKF forecast:

$$\mathbf{x}_{k+1}^{\mathrm{f}} = \mathcal{M}_k(\mathbf{x}_k^{\mathrm{a}}), \qquad \mathbf{P}_{k+1}^{\mathrm{f}} = \mathbf{M}_k \mathbf{P}_k^{\mathrm{a}} \mathbf{M}_k^{\top} + \mathbf{Q}_k$$

Drawbacks:

- Requires tangent linear and adjoint models (complex to implement)
- First-order approximation may be poor for strong nonlinearities
- Can diverge if linearization is inaccurate

The Ensemble Kalman Filter (EnKF)

Motivation for Ensemble Methods

Problems with standard KF in high dimensions:

- Covariance matrix $\mathbf{P} \in \mathbb{R}^{n \times n}$ with $n \sim 10^6 - 10^9$
- Storage: $O(n^2)$ — impossible!
- Matrix operations: $O(n^3)$ — way too expensive!

Key insight (Evensen, 1994):

Monte Carlo approximation

Represent the probability distribution by an **ensemble** of N_e states $\{\mathbf{x}^{(i)}\}_{i=1}^{N_e}$ and estimate statistics from samples:

$$\text{Mean: } \bar{\mathbf{x}} \approx \frac{1}{N_e} \sum_{i=1}^{N_e} \mathbf{x}^{(i)}, \qquad \text{Covariance: } \mathbf{P} \approx \frac{1}{N_e - 1} \sum_{i=1}^{N_e} (\mathbf{x}^{(i)} - \bar{\mathbf{x}})(\mathbf{x}^{(i)} - \bar{\mathbf{x}})^\top$$

Cost: $O(N_e \cdot n)$ storage, $O(N_e \cdot n)$ operations per step

Typically $N_e \sim 20 - 100 \ll n$!

Idea of the Ensemble Kalman Filter

- Represent the state uncertainty by an ensemble of states $\{\mathbf{x}_k^{(i)}\}_{i=1}^{N_e}$.
- Use the **sample mean** and **sample covariance** as approximations of the true mean and covariance:

$$\overline{\mathbf{x}}_k^{\mathrm{f}} = \frac{1}{N_e}\sum_{i=1}^{N_e}\mathbf{x}_k^{(i)}, \qquad \mathbf{P}_k^{\mathrm{f}} \approx \frac{1}{N_e - 1}\sum_{i=1}^{N_e}(\mathbf{x}_k^{(i)} - \overline{\mathbf{x}}_k^{\mathrm{f}})(\mathbf{x}_k^{(i)} - \overline{\mathbf{x}}_k^{\mathrm{f}})^\top$$

- Propagate each ensemble member through the (possibly nonlinear) model.
- Apply a Kalman-like analysis step to each member.

No linearization needed! The ensemble implicitly captures nonlinear error propagation.

Ensemble representation

Define the **ensemble matrix** and **anomaly matrix**:

$$\mathbf{X} = [\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N_e)}] \in \mathbb{R}^{n \times N_e}$$

$$\mathbf{X}' = \mathbf{X} - \bar{\mathbf{x}}\mathbf{1}^\top = [\mathbf{x}^{(1)} - \bar{\mathbf{x}}, \dots, \mathbf{x}^{(N_e)} - \bar{\mathbf{x}}]$$

The sample covariance can be written as:

$$\mathbf{P} \approx \frac{1}{N_e - 1}\mathbf{X}'(\mathbf{X}')^\top$$

Key observation

$\mathbf{P}$ is a **low-rank** matrix of rank at most $N_e - 1$.

We never need to form $\mathbf{P}$ explicitly — only products $\mathbf{Pv}$ for vectors $\mathbf{v}$.

EnKF: Forecast step

For a (possibly nonlinear) model $\mathbf{x}_{k+1} = \mathcal{M}_k(\mathbf{x}_k) + \mathbf{w}_k$:

Forecast each ensemble member

For each $i = 1, \ldots, N_e$:

$$\mathbf{x}_{k+1}^{(i),\mathrm{f}} = \mathcal{M}_k(\mathbf{x}_k^{(i),\mathrm{a}}) + \mathbf{w}_k^{(i)},$$

where $\mathbf{w}_k^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$ are i.i.d. samples.

Compute ensemble statistics

$$\bar{\mathbf{x}}_{k+1}^{\mathrm{f}} = \frac{1}{N_e} \sum_{i=1}^{N_e} \mathbf{x}_{k+1}^{(i),\mathrm{f}}, \qquad \mathbf{P}_{k+1}^{\mathrm{f}} = \frac{1}{N_e - 1} (\mathbf{X}_{k+1}^{\mathrm{f}})' ((\mathbf{X}_{k+1}^{\mathrm{f}})')^\top$$

Note: No tangent linear model needed!

EnKF: Analysis step (stochastic formulation)

Observations: $\mathbf{y}_{k+1}$ with operator $\mathbf{H}_{k+1}$ and error covariance $\mathbf{R}_{k+1}$.

1. Compute the **Kalman gain** using sample covariances:

$$\mathbf{K}_{k+1} = \mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\top}_{k+1}\left(\mathbf{H}_{k+1}\mathbf{P}^{\mathrm{f}}_{k+1}\mathbf{H}^{\top}_{k+1} + \mathbf{R}_{k+1}\right)^{-1}$$

2. For each member i, draw a **perturbed observation**:

$$\mathbf{y}^{(i)}_{k+1} = \mathbf{y}_{k+1} + \boldsymbol{\epsilon}^{(i)}_{k+1}, \quad \boldsymbol{\epsilon}^{(i)}_{k+1} \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_{k+1})$$

3. **Update** each ensemble member:

$$\mathbf{x}^{(i),\mathrm{a}}_{k+1} = \mathbf{x}^{(i),\mathrm{f}}_{k+1} + \mathbf{K}_{k+1}\left(\mathbf{y}^{(i)}_{k+1} - \mathbf{H}_{k+1}\mathbf{x}^{(i),\mathrm{f}}_{k+1}\right)$$

Why perturb the observations?

Without perturbation (deterministic update):

$$\mathbf{x}^{(i),\mathrm{a}} = \mathbf{x}^{(i),\mathrm{f}} + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^{(i),\mathrm{f}})$$

This would give:

$$\mathrm{Var}(\mathbf{x}^{\mathrm{a}}) = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathrm{Var}(\mathbf{x}^{\mathrm{f}})(\mathbf{I} - \mathbf{K}\mathbf{H})^{\top} \neq (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{\mathrm{f}}$$

Problem

The analysis ensemble would be **too narrow —** variance underestimated!

Solution (Burgers et al., 1998): Perturbing observations with samples from $\mathcal{N}(0, \mathbf{R})$ ensures:

$$\mathbb{E}[\text{sample covariance of } \mathbf{x}^{\mathrm{a}}] = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{\mathrm{f}}$$

This is the **stochastic EnKF**.

Efficient implementation of EnKF

Key trick: Never form $\mathbf{P}^{\mathrm{f}}$ explicitly!

Let $\mathbf{X}' = [\mathbf{x}^{(1)} - \bar{\mathbf{x}}, \ldots, \mathbf{x}^{(N_e)} - \bar{\mathbf{x}}]$ be the anomaly matrix.

Then:

$$\mathbf{P}^{\mathrm{f}} = \frac{1}{N_e - 1}\mathbf{X}'(\mathbf{X}')^\top$$

The Kalman gain becomes:

$$\mathbf{K} = \frac{1}{N_e - 1}\mathbf{X}'(\mathbf{X}')^\top\mathbf{H}^\top\left(\frac{1}{N_e - 1}\mathbf{H}\mathbf{X}'(\mathbf{X}')^\top\mathbf{H}^\top + \mathbf{R}\right)^{-1}$$

Define $\mathbf{Y}' = \mathbf{H}\mathbf{X}' \in \mathbb{R}^{m \times N_e}$ (observation anomalies):

$$\mathbf{K} = \frac{1}{N_e - 1}\mathbf{X}'(\mathbf{Y}')^\top\left(\frac{1}{N_e - 1}\mathbf{Y}'(\mathbf{Y}')^\top + \mathbf{R}\right)^{-1}$$

Cost: Invert $m \times m$ matrix (number of observations)!

EnKF Algorithm

Algorithm 2 Stochastic Ensemble Kalman Filter

Initialize: Draw $\mathbf{x}_0^{(i)} \sim \mathcal{N}(\mathbf{x}_0^{\mathrm{b}}, \mathbf{P}_0^{\mathrm{b}})$, $i = 1, \ldots, N_e$
for $k = 0, 1, 2, \ldots$ **do**
    Forecast: For $i = 1, \ldots, N_e$:
        $\mathbf{x}_{k+1}^{(i),\mathrm{f}} = \mathcal{M}_k(\mathbf{x}_k^{(i),\mathrm{a}}) + \mathbf{w}_k^{(i)}$
    Compute $\bar{\mathbf{x}}_{k+1}^{\mathrm{f}}$ and anomaly matrix $\mathbf{X}'_{k+1}$
    Analysis (if observations $\mathbf{y}_{k+1}$ available):
        Compute $\mathbf{Y}' = \mathbf{H}_{k+1}\mathbf{X}'_{k+1}$
        $\mathbf{S} = \frac{1}{N_e-1}\mathbf{Y}'(\mathbf{Y}')^\top + \mathbf{R}_{k+1}$
        For $i = 1, \ldots, N_e$: perturb $\mathbf{y}_{k+1}^{(i)} = \mathbf{y}_{k+1} + \boldsymbol{\epsilon}_{k+1}^{(i)}$
        For $i = 1, \ldots, N_e$:
            $\mathbf{x}_{k+1}^{(i),\mathrm{a}} = \mathbf{x}_{k+1}^{(i),\mathrm{f}} + \frac{1}{N_e-1}\mathbf{X}'(\mathbf{Y}')^\top\mathbf{S}^{-1}(\mathbf{y}_{k+1}^{(i)} - \mathbf{H}_{k+1}\mathbf{x}_{k+1}^{(i),\mathrm{f}})$
end for

Deterministic EnKF variants

Problem with stochastic EnKF: Perturbing observations introduces additional sampling noise.

Deterministic alternatives update the ensemble without perturbations:

- **Ensemble Square Root Filter (EnSRF)** (Whitaker & Hamill, 2002):
  - Update mean using standard Kalman formula
  - Update anomalies using a "square root" of the covariance update
- **Ensemble Transform Kalman Filter (ETKF)** (Bishop et al., 2001):
  - Transform ensemble in the N_e-dimensional subspace
  - Exact covariance update (no sampling error in analysis)
- **Ensemble Adjustment Kalman Filter (EAKF)** (Anderson, 2001):
  - Serial processing of observations
  - Efficient for large observation networks

Ensemble Square Root Filter (EnSRF)

Mean update (same as KF):

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

Anomaly update: Find $\tilde{\mathbf{K}}$ such that:

$$(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{H})\mathbf{P}^{\mathrm{f}}(\mathbf{I} - \tilde{\mathbf{K}}\mathbf{H})^{\top} = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^{\mathrm{f}}$$

For scalar observations ($m = 1$):

$$\tilde{\mathbf{K}} = \mathbf{K}\left(1 + \sqrt{\frac{R}{R + \mathbf{H}\mathbf{P}^{\mathrm{f}}\mathbf{H}^{\top}}}\right)^{-1}$$

Ensemble update:

$$\mathbf{x}^{(i),\mathrm{a}} - \bar{\mathbf{x}}^{\mathrm{a}} = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{H})(\mathbf{x}^{(i),\mathrm{f}} - \bar{\mathbf{x}}^{\mathrm{f}})$$

Advantage: No sampling noise from observation perturbations.

Practical Issues and Techniques

Challenges with small ensembles

With $N_e \ll n$, the sample covariance has **rank** $\leq N_e - 1$.

Consequences:

1. **Spurious correlations**: Sample covariance shows correlations between distant/unrelated variables (sampling noise)
2. **Rank deficiency**: Cannot represent uncertainty in all directions
3. **Filter divergence**: Ensemble collapses to a single point, ignoring observations

Two essential techniques

- **Localization**: Remove spurious long-range correlations
- **Inflation**: Artificially increase ensemble spread

Covariance localization

Idea: Physical correlations decay with distance. Multiply sample covariance by a **localization function**:

$$\mathbf{P}^{\mathrm{loc}} = \boldsymbol{\rho} \circ \mathbf{P}^{\mathrm{f}}$$

where $\circ$ is the Schur (element-wise) product.

Common choice: Gaspari-Cohn function (compactly supported):

$$\rho(r) = \begin{cases} 1 - \frac{5}{3}r^2 + \frac{5}{8}r^3 + \frac{1}{2}r^4 - \frac{1}{4}r^5, & 0 \leq r \leq 1 \\ 4 - 5r + \frac{5}{3}r^2 + \frac{5}{8}r^3 - \frac{1}{2}r^4 + \frac{1}{12}r^5 - \frac{2}{3r}, & 1 < r \leq 2 \\ 0, & r > 2 \end{cases}$$

where $r = d/L$, d is distance, L is localization length scale.

Effect:

- Increases effective rank of covariance
- Preserves local correlations, removes distant ones

Covariance inflation

Problem: EnKF tends to **underestimate** uncertainty due to:

- Sampling errors
- Model errors not fully represented
- Nonlinear effects

Solution: Artificially inflate the ensemble spread.

Multiplicative inflation (before analysis):

$$\mathbf{x}^{(i)} \leftarrow \bar{\mathbf{x}} + \alpha(\mathbf{x}^{(i)} - \bar{\mathbf{x}}), \qquad \alpha > 1$$

Typically $\alpha \in [1.01, 1.10]$.

Additive inflation (after analysis):

$$\mathbf{x}^{(i),\mathrm{a}} \leftarrow \mathbf{x}^{(i),\mathrm{a}} + \boldsymbol{\eta}^{(i)}, \qquad \boldsymbol{\eta}^{(i)} \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_{\mathrm{add}})$$

Adaptive inflation: Estimate α from innovation statistics.

Comparison: KF vs EKF vs EnKF

	KF	**EKF**	**EnKF**
Linearity	Required	Linearized	Not needed
Gaussianity	Required	Assumed	Approximated
Covariance	Full $\mathbf{P}$	Full $\mathbf{P}$	Sample
Storage	$O(n^2)$	$O(n^2)$	$O(N_e \cdot n)$
Cost/step	$O(n^3)$	$O(n^3)$	$O(N_e \cdot n)$
TL/Adjoint	No	Yes	No
Parallelizable	Limited	Limited	Highly

EnKF advantages:

- Scales to very high dimensions
- No code differentiation needed
- Naturally parallelizable (independent forecasts)
- Handles nonlinearities implicitly

Connection to Reduced Order Models

Motivation: EnKF + ROM

Even with EnKF, each forecast requires solving the full model:

$$\mathbf{x}_{k+1}^{(i),\mathrm{f}} = \mathcal{M}_k(\mathbf{x}_k^{(i),\mathrm{a}})$$

For PDE-based models with $n \sim 10^6$, running N_e forecasts is expensive!

Idea: Use Reduced Order Models

Replace the full model $\mathcal{M}$ with a reduced model $\mathcal{M}_N$ of dimension $N \ll n$:

- Reduced Basis (RB) methods
- Proper Orthogonal Decomposition (POD)
- Dynamic Mode Decomposition (DMD)

Connection to previous lectures: RB/EIM/GEIM provide efficient surrogates for parametrized PDEs!

ROM-EnKF: Reduced-order ensemble forecasting

Setup:

- Full state space: $\mathbf{x} \in \mathbb{R}^n$
- Reduced basis: $\mathbf{V} = [\phi_1, \ldots, \phi_N] \in \mathbb{R}^{n \times N}$
- Reduced state: $\mathbf{x} \approx \mathbf{V}\hat{\mathbf{x}}$, where $\hat{\mathbf{x}} \in \mathbb{R}^N$

Reduced forecast:

$$\hat{\mathbf{x}}_{k+1}^{(i),\mathrm{f}} = \hat{\mathcal{M}}_k(\hat{\mathbf{x}}_k^{(i),\mathrm{a}})$$

Lift to full space for analysis:

$$\mathbf{x}_{k+1}^{(i),\mathrm{f}} = \mathbf{V}\hat{\mathbf{x}}_{k+1}^{(i),\mathrm{f}}$$

Benefits:

- Forecast cost: $O(N_e \cdot N)$ instead of $O(N_e \cdot n)$
- Ensemble spread lives in low-dimensional subspace
- Can use larger ensembles with same computational budget
- Observation operators $\mathbf{H}$ can be built from GEIM functionals (magic points / sensors) designed offline

Connection to PBDW

PBDW (Parametrized Background Data-Weak) for **static** problems:

$$u^* = \arg \min_{z \in u_{\mathrm{bg}} + V_n} \|z - u_{\mathrm{bg}}\|^2 \quad \text{s.t.} \quad \ell_m(z) = y_m, \; m = 1, \ldots, M$$

Comparison with KF/EnKF analysis:

	PBDW	**KF**	**EnKF**
Background	Model manifold V_n	Forecast $\mathbf{x}^{\mathrm{f}}$	Ensemble mean
Observations	Functionals ℓ_m	$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}$	$\mathbf{y} = \mathbf{H}\mathbf{x} + \boldsymbol{\epsilon}$
Constraint	Exact interp.	Weighted LS	Weighted LS
Uncertainty	A posteriori error	Covariance $\mathbf{P}^{\mathrm{a}}$	Sample cov.
Time	Static	Sequential	Sequential

Key insight: All methods combine model information with observations!

PBDW vs Kalman: Detailed comparison

Aspect	**Time-dep. PBDW**	**Kalman/EnKF**
Philosophy	Deterministic	Stochastic/Bayesian
Background	RB manifold structure	Probabilistic forecast
Update	Projection + constraints	Bayesian conditioning
Error quantif.	A posteriori bounds	Covariance matrix
Model error	Implicit in V_n	Explicit $\mathbf{Q}_k$
Obs. error	Not modeled	Explicit $\mathbf{R}_k$
Offline cost	RB + sensor selection	None (or ROM build)
Online cost	$O(N^3)$	$O(n^3)$ or $O(N_e n)$
Nonlinearity	Via EIM/DEIM	Via ensemble

When to use which?

- **PBDW**: well-understood parametric model, need certified bounds
- **EnKF**: uncertain/complex model, need UQ, high dimensions

Hybrid approaches: PBDW + Kalman

Idea: Combine strengths of both approaches!

Option 1: PBDW-informed background for KF

- Use PBDW reconstruction as $\mathbf{x}^{\mathrm{b}}$
- Background covariance $\mathbf{B}$ from PBDW error estimator
- Apply KF analysis for additional observations

Option 2: ROM-EnKF with PBDW analysis

- Ensemble forecast in reduced space
- PBDW-style analysis (exact constraint satisfaction)
- Maintains ensemble spread for UQ

Option 3: Sequential PBDW with model propagation

- PBDW reconstruction at observation times
- ROM propagation between observations
- No stochastic sampling required

Hybrid approaches: GEIM + EnKF

Generalized EIM provides interpolation basis $\{q_m\}$ and magic points $\{x_m\}$.

Observation operator from GEIM:

$$\mathbf{H} = [\ell_1, \ldots, \ell_M]^\top$$

where $\ell_m(u) = u(x_m)$ or $\ell_m(u) = \langle \sigma_m, u \rangle$.

Hybrid algorithm:

1. Use GEIM to select optimal sensor locations
2. Use RB for reduced-order ensemble forecasting
3. Apply EnKF analysis with GEIM-based observations

Advantages:

- Optimal sensor placement from offline stage
- Efficient forecasts with ROM
- Principled uncertainty quantification from EnKF

Live vs. follow-up (timeboxing)

Live 2h (core concepts)

- Dynamical systems + ROM/(G)EIM/PBDW toolbox
- BLUE vs. PBDW; KF cycle; EnKF and practicalities
- ROM-EnKF and hybrid ideas

Follow-up 2h (examples/lab)

- Lorenz-63 EnKF notebook walkthrough
- PDE examples (heat equation, parameter estimation)
- Optional tweaks: inflation/localization experiments

Examples and Applications

Hands-on: Lorenz-63 mini-lab

Why Lorenz-63?

- Simple 3D chaotic system capturing the "butterfly effect" in meteorology
- Perfect sandbox to compare KF linearization vs. EnKF ensemble handling of nonlinearity

Notebook flow (`Slides/rbm/exercises/enkf_lorenz.ipynb`):

1. Quick scalar KF refresher
2. Integrate Lorenz-63 and generate synthetic noisy observations
3. Implement EnKF with small ensemble; plot truth/mean/spread
4. Try inflation or localization radii and discuss forecast skill

Outcome: ready-to-run exercise that mirrors the lecture pipeline.

Example 1: 1D heat equation

Consider the heat equation on $\Omega = (0, 1)$:

$$\frac{\partial u}{\partial t} = \kappa \frac{\partial^2 u}{\partial x^2}, \quad u(0, t) = u(1, t) = 0, \quad u(x, 0) = u_0(x)$$

State estimation problem:

- Unknown: initial condition $u_0(x)$
- Observations: $y_m(t) = u(x_m, t) + \epsilon_m$ at M sensors
- Model: discretized heat equation

EnKF setup:

- State: $\mathbf{x}_k = [u(x_1, t_k), \ldots, u(x_n, t_k)]^\top$
- Model: $\mathbf{M} = e^{\Delta t \kappa \mathbf{L}}$ (discrete Laplacian)
- Observation: $\mathbf{H}$ selects sensor locations

Example 2: Parameter estimation

Joint state-parameter estimation:

Augment the state with unknown parameters:

$$\tilde{\mathbf{x}}_k = \begin{pmatrix} \mathbf{x}_k \\ \boldsymbol{\theta} \end{pmatrix}$$

Augmented dynamics:

$$\tilde{\mathbf{x}}_{k+1} = \begin{pmatrix} \mathcal{M}_k(\mathbf{x}_k; \boldsymbol{\theta}) \\ \boldsymbol{\theta} \end{pmatrix} + \tilde{\mathbf{w}}_k$$

EnKF naturally handles this:

- Ensemble of $(\mathbf{x}^{(i)}, \boldsymbol{\theta}^{(i)})$ pairs
- Cross-covariances $\mathrm{Cov}(\mathbf{x}, \boldsymbol{\theta})$ estimated from samples
- Parameters updated through correlations with observed state

Application: Estimate thermal conductivity κ from temperature measurements.

Example 3: Weather forecasting

Operational NWP (Numerical Weather Prediction):

- State dimension: $n \sim 10^9$ (atmosphere + ocean)
- Ensemble size: $N_e \sim 20 - 100$
- Observations: $\sim 10^7$ per day (satellites, radiosondes, aircraft...)
- Assimilation cycle: every 6 hours

Techniques used:

- Localization: ~ 1000 km horizontal, ~ 3 vertical levels
- Inflation: adaptive multiplicative + additive
- Hybrid methods: EnKF + 4D-Var
- Parallel implementation: $O(10^4)$ processors

Centers using EnKF variants: ECMWF, NCEP, Environment Canada, Météo-France...

Summary and Conclusions

Summary

Kalman Filter

- Optimal linear estimator for Gaussian systems
- Two-step cycle: forecast + analysis
- Propagates mean and full covariance
- Limited to linear, Gaussian, low-dimensional problems

Ensemble Kalman Filter

- Monte Carlo approximation of KF
- Handles nonlinear models naturally
- Scales to very high dimensions
- Requires localization and inflation in practice

Connection to ROM

RB, EIM, GEIM, PBDW can be combined with EnKF for efficient data assimilation in parametrized PDE systems.

Key takeaways

1. **Data assimilation** = optimal combination of models and observations
2. **Kalman filter** = sequential BLUE with covariance propagation
3. **EnKF** = ensemble-based approximation, scalable to high dimensions
4. **Practical EnKF** requires localization and inflation
5. **Time-dependent PBDW** = deterministic alternative using ROM structure
6. **Hybrid approaches** combine strengths of PBDW and EnKF

Next steps in the course:

- Variational methods (3D-Var, 4D-Var)
- Hybrid ensemble-variational methods
- Application to specific engineering problems

References i

- R.E. Kalman (1960). A new approach to linear filtering and prediction problems. *J. Basic Eng.*, 82(1):35–45.
- G. Evensen (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model. *J. Geophys. Res.*, 99(C5):10143–10162.
- G. Burgers, P.J. van Leeuwen, G. Evensen (1998). Analysis scheme in the ensemble Kalman filter. *Mon. Weather Rev.*, 126:1719–1724.
- G. Evensen (2009). *Data Assimilation: The Ensemble Kalman Filter*. Springer.
- M. Asch, M. Bocquet, M. Nodet (2016). *Data Assimilation: Methods, Algorithms, and Applications*. SIAM.

References ii

- Y. Maday, A.T. Patera, J.D. Penn, M. Yano (2015). A parameterized-background data-weak approach to variational data assimilation. *ESAIM: M2AN*, 49(5):1399–1423.
- A. Benaceur (2018). *Réduction de modèles en thermique et mécanique non-linéaires*. PhD thesis, Université Paris-Est Marne la Vallée. HAL: tel-01958278.
- A. Benaceur (2019). A time-dependent parametrized background data-weak approach. *Numerical Mathematics and Advanced Applications ENUMATH 2019*. Springer. DOI: 10.1007/978-3-030-55874-1_11.