Introduction to Data Assimilation

Clarice Poon
University of Bath

March 23, 2020

What is Data Assimilation?

Data assimilation (DA) is an approach for combining observation with model output.

This is a special case of inverse problems and approximates the true states of some physical system at a given time by combining time observations with some dynamic model in an optimal way.

What is Data Assimilation?

Data assimilation (DA) is an approach for combining observation with model output.

This is a special case of inverse problems and approximates the true states of some physical system at a given time by combining time observations with some dynamic model in an optimal way.

Used heavily in numerical weather forecasting. You want to estimate the state of the atmosphere (e.g. temperature, pressure, etc). To do this, you have...

What is Data Assimilation?

Data assimilation (DA) is an approach for combining observation with model output.

This is a special case of inverse problems and approximates the true states of some physical system at a given time by combining time observations with some dynamic model in an optimal way.

Used heavily in numerical weather forecasting. You want to estimate the state of the atmosphere (e.g. temperature, pressure, etc). To do this, you have...

1. **a-priori information** (e.g. yesterday's forecast).

What is Data Assimilation?

Data assimilation (DA) is an approach for combining observation with model output.

This is a special case of inverse problems and approximates the true states of some physical system at a given time by combining time observations with some dynamic model in an optimal way.

Used heavily in numerical weather forecasting. You want to estimate the state of the atmosphere (e.g. temperature, pressure, etc). To do this, you have...

1. **a-priori information** (e.g. yesterday's forecast).
2. **observations** at locations scattered in space (e.g. satellites, ships and buoys, surface stations, aeroplanes).

What is Data Assimilation?

Data assimilation (DA) is an approach for combining observation with model output.

This is a special case of inverse problems and approximates the true states of some physical system at a given time by combining time observations with some dynamic model in an optimal way.

Used heavily in numerical weather forecasting. You want to estimate the state of the atmosphere (e.g. temperature, pressure, etc). To do this, you have...

1. **a-priori information** (e.g. yesterday's forecast).
2. **observations** at locations scattered in space (e.g. satellites, ships and buoys, surface stations, aeroplanes).
3. some (possibly imperfect) **physical model** $M(x_i) = x_{i+1}$.

What is Data Assimilation?

Data assimilation (DA) is an approach for combining observation with model output.

This is a special case of inverse problems and approximates the true states of some physical system at a given time by combining time observations with some dynamic model in an optimal way.

Used heavily in numerical weather forecasting. You want to estimate the state of the atmosphere (e.g. temperature, pressure, etc). To do this, you have...

1. **a-priori information** (e.g. yesterday's forecast).
2. **observations** at locations scattered in space (e.g. satellites, ships and buoys, surface stations, aeroplanes).
3. some (possibly imperfect) **physical model** $M(x_i) = x_{i+1}$.

How do we combines all this information?

Typical data assimilation framework

The aim is to estimate a sequence of true states $x_k^t \in \mathbb{R}^n$ for $k = 1, \ldots, K$.
Here, k indexes points in time.

Typical data assimilation framework

The aim is to estimate a sequence of true states $x_k^t \in \mathbb{R}^n$ for $k = 1, \ldots, K$. Here, k indexes points in time.

- We assume some **model**:

$$x_k^t = M_k[x_{k-1}^t] + e_k^m, \qquad e_k^m \sim \mathcal{N}(0, Q_k),$$

  and e_k^m is the so-called modelling error.

Typical data assimilation framework

The aim is to estimate a sequence of true states $x_k^t \in \mathbb{R}^n$ for $k = 1, \ldots, K$. Here, k indexes points in time.

- We assume some **model**:

$$x_k^t = M_k[x_{k-1}^t] + e_k^m, \qquad e_k^m \sim \mathcal{N}(0, Q_k),$$

  and e_k^m is the so-called modelling error.
- We are given **observations** $y \in \mathbb{R}^p$ (typically $p \ll n$):

$$y_k = H_k[x_k^t] + e_k^o, \qquad e_k^o \sim \mathcal{N}(0, R_k).$$

Typical data assimilation framework

The aim is to estimate a sequence of true states $x_k^t \in \mathbb{R}^n$ for $k = 1, \dots, K$. Here, k indexes points in time.

- We assume some **model**:

$$x_k^t = M_k[x_{k-1}^t] + e_k^m, \qquad e_k^m \sim \mathcal{N}(0, Q_k),$$

  and e_k^m is the so-called modelling error.
- We are given **observations** $y \in \mathbb{R}^p$ (typically $p \ll n$):

$$y_k = H_k[x_k^t] + e_k^o, \qquad e_k^o \sim \mathcal{N}(0, R_k).$$

- We have a **background state** (initial guess) x^b. We assume that $x^b = x_0^t + e^b$ where $e^b \in \mathcal{N}(0, B)$.

Typical data assimilation framework

The aim is to estimate a sequence of true states $x_k^t \in \mathbb{R}^n$ for $k = 1, \ldots, K$. Here, k indexes points in time.

- We assume some **model**:

$$x_k^t = M_k[x_{k-1}^t] + e_k^m, \qquad e_k^m \sim \mathcal{N}(0, Q_k),$$

  and e_k^m is the so-called modelling error.
- We are given **observations** $y \in \mathbb{R}^p$ (typically $p \ll n$):

$$y_k = H_k[x_k^t] + e_k^o, \qquad e_k^o \sim \mathcal{N}(0, R_k).$$

- We have a **background state** (initial guess) x^b. We assume that $x^b = x_0^t + e^b$ where $e^b \in \mathcal{N}(0, B)$.

Assume throughout that

- the covariance matrices Q_k, R_k and B are known.
- e_k^m, e_k^o and e^b are independent.

Typical data assimilation framework

The aim is to estimate a sequence of true states $x_k^t \in \mathbb{R}^n$ for $k = 1, \ldots, K$. Here, k indexes points in time.

- We assume some **model**:

$$x_k^t = M_k[x_{k-1}^t] + e_k^m, \qquad e_k^m \sim \mathcal{N}(0, Q_k),$$

  and e_k^m is the so-called modelling error.
- We are given **observations** $y \in \mathbb{R}^p$ (typically $p \ll n$):

$$y_k = H_k[x_k^t] + e_k^o, \qquad e_k^o \sim \mathcal{N}(0, R_k).$$

- We have a **background state** (initial guess) x^b. We assume that $x^b = x_0^t + e^b$ where $e^b \in \mathcal{N}(0, B)$.

Assume throughout that

- the covariance matrices Q_k, R_k and B are known.
- e_k^m, e_k^o and e^b are independent.

Goal: combine the above to obtain an estimate x^a. We call this the **analysis**.

We will discuss two approaches to DA:

1. Statistical interpolation techniques, these include **BLUE** and **Kalman filters**.
2. Variational techniques, these include **3D-Var** and **4D-Var.**

Outline

Best linear unbiased estimate (BLUE)

The Kalman filter

The extended Kalman filter

The stochastic ensemble Kalman filter

3D Var

4D Var

Best linear unbiased estimate (BLUE)

Setup

We want to estimate the true state $x^t \in \mathbb{R}^n$ given

Best linear unbiased estimate (BLUE)

Setup

We want to estimate the true state $x^t \in \mathbb{R}^n$ given

- a background state $x^b \in \mathbb{R}^n$.

Best linear unbiased estimate (BLUE)

Setup

We want to estimate the true state $x^t \in \mathbb{R}^n$ given

- a background state $x^b \in \mathbb{R}^n$.
- observation $y = H[x^t] + e^o \in \mathbb{R}^p$, where $H \in \mathbb{R}^{p \times n}$ is a matrix, typically, $p \ll n$.

Best linear unbiased estimate (BLUE)

Setup

We want to estimate the true state $x^t \in \mathbb{R}^n$ given

- a background state $x^b \in \mathbb{R}^n$.
- observation $y = H[x^t] + e^o \in \mathbb{R}^p$, where $H \in \mathbb{R}^{p \times n}$ is a matrix, typically, $p \ll n$.

Assume that the background error $e^b \stackrel{\text{def.}}{=} x^b - x^t$ and the observation error e^o are uncorrelated with symmetric covariance matrices

$$B \stackrel{\text{def.}}{=} \mathbb{E}[e^b(e^b)^\top] \quad \text{and} \quad R \stackrel{\text{def.}}{=} \mathbb{E}[e^o(e^o)^\top]$$

respectively.

Best linear unbiased estimate (BLUE)

We consider a statistical interpolation technique known as BLUE.

Best linear unbiased estimate (BLUE)

We consider a statistical interpolation technique known as BLUE.

Let us make the Ansatz that

$$x^a = Lx^b + Ky, \quad \text{where} \quad L \in \mathbb{R}^{n\times n} \quad \text{and} \quad K \in \mathbb{R}^{n\times p}.$$

Best linear unbiased estimate (BLUE)

We consider a statistical interpolation technique known as BLUE.

Let us make the Ansatz that

$$x^a = Lx^b + Ky, \quad \text{where} \quad L \in \mathbb{R}^{n\times n} \quad \text{and} \quad K \in \mathbb{R}^{n\times p}.$$

How do we choose L and K such that x^a is "statistically optimal"?

1. an unbiased estimate: $\mathbb{E}[x^a - x^t] = 0$
2. to minimise $\mathbb{E}[\|x^a - x^t\|^2]$.

Unbiased analysis for $x^a = Lx^b + Ky$:

Using $y = Hx^t + e^o$:

$$x^a - x^t = L(x^b - x^t + x^t) + K(Hx^t + e^o) - x^t$$

Unbiased analysis for $x^a = Lx^b + Ky$:

Using $y = Hx^t + e^o$:

$$x^a - x^t = L(x^b - x^t + x^t) + K(Hx^t + e^o) - x^t$$

which implies that

$$e^a = Le^b + Ke^o + (L + KH - \mathrm{Id})x^t.$$

Unbiased analysis for $x^a = Lx^b + Ky$:

Using $y = Hx^t + e^o$:

$$x^a - x^t = L(x^b - x^t + x^t) + K(Hx^t + e^o) - x^t$$

which implies that

$$e^a = Le^b + Ke^o + (L + KH - \mathrm{Id})x^t.$$

Taking the expectation leads to

$$\mathbb{E}[e^a] = (L + KH - \mathrm{Id})\mathbb{E}[x^t]$$

Unbiased analysis

Unbiased analysis for $x^a = Lx^b + Ky$:

Using $y = Hx^t + e^o$:

$$x^a - x^t = L(x^b - x^t + x^t) + K(Hx^t + e^o) - x^t$$

which implies that

$$e^a = Le^b + Ke^o + (L + KH - \mathrm{Id})x^t.$$

Taking the expectation leads to

$$\mathbb{E}[e^a] = (L + KH - \mathrm{Id})\mathbb{E}[x^t]$$

So, to ensure that our analysis state is unbiased, a **sufficient** (although not necessary) condition is

$$L = \mathrm{Id} - KH,$$

The analysis covariance

From the choice of $L = \mathrm{Id} - KH$,

$$x^a = (\mathrm{Id} - KH)x^b + Ky \implies x^a = x^b + K(y - Hx^b)$$

The analysis covariance

From the choice of $L = \mathrm{Id} - KH$,

$$x^a = (\mathrm{Id} - KH)x^b + Ky \implies x^a = x^b + K(y - Hx^b)$$

and recalling that $y = Hx^t + e^o$, we have

$$e^a = e^b + K(e^o - He^b) = Le^b + Ke^o$$

The analysis covariance

From the choice of $L = \mathrm{Id} - KH$,

$$x^a = (\mathrm{Id} - KH)x^b + Ky \implies x^a = x^b + K(y - Hx^b)$$

and recalling that $y = Hx^t + e^o$, we have

$$e^a = e^b + K(e^o - He^b) = Le^b + Ke^o$$

The covariance matrix is therefore

$$\begin{aligned} P^a &= \mathbb{E}[e^a(e^a)^\top] = LBL^\top + KRK^\top \\ &= (\mathrm{Id} - KH)B(\mathrm{Id} - KH)^\top + KRK^\top. \end{aligned}$$

Minimal error

We just deduced that $P^a = (\mathrm{Id} - KH)B(\mathrm{Id} - KH)^\top + KRK^\top$.

We want to choose K to minimise $\mathbb{E}[\left\| x^a - x^t \right\|^2] = \mathrm{tr}\,(P^a)$.

Minimal error

We just deduced that $P^a = (\mathrm{Id} - KH)B(\mathrm{Id} - KH)^\top + KRK^\top$.

We want to choose K to minimise $\mathbb{E}[\left\|x^a - x^t\right\|^2] = \mathrm{tr}\,(P^a)$.

Look at the variation of $\mathrm{tr}\,(P^a)$ with respect to K:

$$\begin{aligned}\delta(\mathrm{tr}(P^a)) &= \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}(\mathrm{tr}(P^a_{K+\varepsilon\delta K}) - \mathrm{tr}(P^a_K)) \\ &= 2\,\mathrm{tr}\left((-LBH^\top + KR)(\delta K)^\top\right)\end{aligned}$$

and so

$$-(\mathrm{Id} - KH)BH^\top + KR = 0 \iff K = BH^\top(R + HBH^\top)^{-1}.$$

Minimal error

We just deduced that $P^a = (\mathrm{Id} - KH)B(\mathrm{Id} - KH)^\top + KRK^\top$.

We want to choose K to minimise $\mathbb{E}[\|x^a - x^t\|^2] = \operatorname{tr}(P^a)$.

Look at the variation of $\operatorname{tr}(P^a)$ with respect to K:

$$\begin{aligned} \delta(\operatorname{tr}(P^a)) &= \lim_{\varepsilon \to 0} \frac{1}{\varepsilon}(\operatorname{tr}(P^a_{K+\varepsilon\delta K}) - \operatorname{tr}(P^a_K)) \\ &= 2 \operatorname{tr}\left((-LBH^\top + KR)(\delta K)^\top\right) \end{aligned}$$

and so

$$-(\mathrm{Id} - KH)BH^\top + KR = 0 \iff K = BH^\top(R + HBH^\top)^{-1}.$$

To summarise:

BLUE

Let $K = BH^\top(R + HBH^\top)^{-1}$.

- $x^a = x^b + K(y - Hx^b)$ is called the BLUE estimator with covariance

$$P^a = (\mathrm{Id} - KH)B + \left(-(\mathrm{Id} - KH)BH^\top + KR\right)K^\top = (\mathrm{Id} - KH)B.$$

- We call K the gain matrix and $y - Hx^b$ the innovation.

Example

Scenario

Suppose you are shipwrecked at sea, a few km from shore.

- Before boarding a small lifeboat, you measure your coordinates $(u, v) = (0, v_b)$ with high accuracy. The first axis is parallel to shore, the second axis is perpendicular to shore.
- After 1 hour, you want to estimate your new coordinates. So, you guess your distance to shore: v_o with variance σ_o^2.
- Assume that the probability that the boat remains at (u_b, v_b) follows $\mathcal{N}(0, \sigma_b^2)$.

BLUE

Let $K = BH^\top(R + HBH^\top)^{-1}$. The estimator is $x^a = x^b + K(y - Hx^b)$ with covariance $P^a = (\mathrm{Id} - KH)B$.

Example

BLUE

Let $K = BH^\top(R + HBH^\top)^{-1}$. The estimator is $x^a = x^b + K(y - Hx^b)$ with covariance $P^a = (\mathrm{Id} - KH)B$.

- The state vector is $x = (u, v)^\top$.

Example

BLUE

Let $K = BH^\top(R + HBH^\top)^{-1}$. The estimator is $x^a = x^b + K(y - Hx^b)$ with covariance $P^a = (\mathrm{Id} - KH)B$.

- The state vector is $x = (u, v)^\top$.
- Observation is

$$y \overset{\text{def.}}{=} v_o = Hx + \varepsilon, \quad \text{where} \quad H = (0, 1), \quad \varepsilon \sim \mathcal{N}(0, \sigma_o^2).$$

So, the observation covariance matrix is the scalar $R = \sigma_o^2$.

Example

BLUE

Let $K = BH^\top(R + HBH^\top)^{-1}$. The estimator is $x^a = x^b + K(y - Hx^b)$ with covariance $P^a = (\mathrm{Id} - KH)B$.

- The state vector is $x = (u, v)^\top$.
- Observation is

$$y \stackrel{\text{def.}}{=} v_o = Hx + \varepsilon, \quad \text{where} \quad H = (0, 1), \quad \varepsilon \sim \mathcal{N}(0, \sigma_o^2).$$

So, the observation covariance matrix is the scalar $R = \sigma_o^2$.

Example

BLUE

Let $K = BH^\top(R + HBH^\top)^{-1}$. The estimator is $x^a = x^b + K(y - Hx^b)$ with covariance $P^a = (\mathrm{Id} - KH)B$.

- The state vector is $x = (u, v)^\top$.
- Observation is

$$y \overset{\text{def.}}{=} v_o = Hx + \varepsilon, \quad \text{where} \quad H = (0, 1), \quad \varepsilon \sim \mathcal{N}(0, \sigma_o^2).$$

  So, the observation covariance matrix is the scalar $R = \sigma_o^2$.
- The background is $x^b = \binom{0}{v_b}$ with covariance matrix $B = \sigma_b^2 \mathrm{Id}_2$.

Example

Example

The gain matrix is $\frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}\begin{pmatrix}0 & 1\end{pmatrix}^\top$

Example

The gain matrix is $\frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}\begin{pmatrix}0 & 1\end{pmatrix}^\top$

The BLUE estimator is

$$x^a = x^b + K(y - Hx^b) = \begin{pmatrix}0 \\ v_b\end{pmatrix} + K(v_o - v_b) = \begin{pmatrix}0 \\ v_b + \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}(v_o - v_b)\end{pmatrix}$$

Example

The gain matrix is $\frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}\begin{pmatrix}0 & 1\end{pmatrix}^\top$

The BLUE estimator is

$$x^a = x^b + K(y - Hx^b) = \begin{pmatrix}0\\ v_b\end{pmatrix} + K(v_o - v_b) = \begin{pmatrix}0\\ v_b + \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}(v_o - v_b)\end{pmatrix}$$

The posterior error is

$$P^a = (\mathrm{Id} - KH)B = \begin{pmatrix}\sigma_b^2 & 0\\ 0 & \frac{\sigma_o^2\sigma_b^2}{\sigma_o^2+\sigma_b^2}\end{pmatrix}$$

Example

The gain matrix is $\frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}\begin{pmatrix}0 & 1\end{pmatrix}^\top$

The BLUE estimator is

$$x^a = x^b + K(y - Hx^b) = \begin{pmatrix} 0 \\ v_b \end{pmatrix} + K(v_o - v_b) = \begin{pmatrix} 0 \\ v_b + \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}(v_o - v_b) \end{pmatrix}$$

The posterior error is

$$P^a = (\mathrm{Id} - KH)B = \begin{pmatrix} \sigma_b^2 & 0 \\ 0 & \frac{\sigma_o^2\sigma_b^2}{\sigma_o^2+\sigma_b^2} \end{pmatrix}$$

Our analysis brings no new information to the first variable and uncertainty in the u coordinate will simply grow with time.

Example

The gain matrix is $\frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}\begin{pmatrix}0 & 1\end{pmatrix}^\top$

The BLUE estimator is

$$x^a = x^b + K(y - Hx^b) = \begin{pmatrix}0\\ v_b\end{pmatrix} + K(v_o - v_b) = \begin{pmatrix}0\\ v_b + \frac{\sigma_b^2}{\sigma_b^2+\sigma_o^2}(v_o - v_b)\end{pmatrix}$$

The posterior error is

$$P^a = (\mathrm{Id} - KH)B = \begin{pmatrix}\sigma_b^2 & 0\\ 0 & \frac{\sigma_o^2\sigma_b^2}{\sigma_o^2+\sigma_b^2}\end{pmatrix}$$

Our analysis brings no new information to the first variable and uncertainty in the u coordinate will simply grow with time.

Denoting σ_a^2 as the error on the v coordinate, we have

$$\frac{1}{\sigma_a^2} = \frac{1}{\sigma_b^2} + \frac{1}{\sigma_o^2}$$

This will be dominated by σ_o^2 as time progresses.

Outline

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

Setup

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

1. an initial background state $x^b = x_0^t + e^b \in \mathbb{R}^n$ where $e^b \in \mathcal{N}(0, B)$.

Setup

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

1. an initial background state $x^b = x_0^t + e^b \in \mathbb{R}^n$ where $e^b \in \mathcal{N}(0, B)$.
2. observations $y_k = H_k x_k^t + e_k^o \in \mathbb{R}^p$ for $k = 1, \dots, K$, where $e_k^o \sim \mathcal{N}(0, R_k)$ $H_k \in \mathbb{R}^{p \times n}$, typically, $p \ll n$.

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

1. an initial background state $x^b = x_0^t + e^b \in \mathbb{R}^n$ where $e^b \in \mathcal{N}(0, B)$.
2. observations $y_k = H_k x_k^t + e_k^o \in \mathbb{R}^p$ for $k = 1, \dots, K$, where $e_k^o \sim \mathcal{N}(0, R_k)$ $H_k \in \mathbb{R}^{p \times n}$, typically, $p \ll n$.
3. a model $x_k^t = M_k x_{k-1}^t + e_k^m$ for $k = 1, \dots, K$ where $e_k^m \sim \mathcal{N}(0, Q_k)$ and $M_k \in \mathbb{R}^{n \times n}$.

Assume independence between the background, observation and model.

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

1. an initial background state $x^b = x_0^t + e^b \in \mathbb{R}^n$ where $e^b \in \mathcal{N}(0, B)$.
2. observations $y_k = H_k x_k^t + e_k^o \in \mathbb{R}^p$ for $k = 1, \dots, K$, where $e_k^o \sim \mathcal{N}(0, R_k)$ $H_k \in \mathbb{R}^{p \times n}$, typically, $p \ll n$.
3. a model $x_k^t = M_k x_{k-1}^t + e_k^m$ for $k = 1, \dots, K$ where $e_k^m \sim \mathcal{N}(0, Q_k)$ and $M_k \in \mathbb{R}^{n \times n}$.

Assume independence between the background, observation and model.

NB: we are assuming that the model and observation are linear.

The Kalman filter

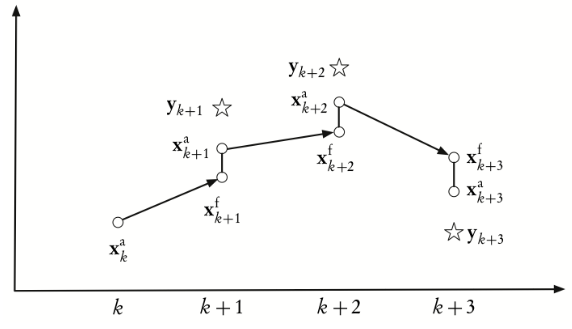
The Kalman filter is a **sequential statistical interpolation method**.

The Kalman filter

The Kalman filter is a **sequential statistical interpolation method**.

It alternates between

1. Analysis. Interpolates the observations and the background at time t_k .
2. Forecast. Advances to the next time point using the model.



The Kalman filter

We initialise with a state $x_0^f = x^b$ and covariance matrix $P_0^f = \mathbb{E}[e^b(e^b)^\top]$.

For $t_k = 1, 2, \ldots$,

The Kalman filter

We initialise with a state $x_0^f = x^b$ and covariance matrix $P_0^f = \mathbb{E}[e^b(e^b)^\top]$.

For $t_k = 1, 2, \ldots,$

1. Compute the **analysis estimate**: Just as for BLUE, let

   $$x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$$

   with

   $$K_k^* = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$$

   which is called the Kalman gain matrix. Moreover, the error covariance is

   $$P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$$

We initialise with a state $x_0^f = x^b$ and covariance matrix $P_0^f = \mathbb{E}[e^b(e^b)^\top]$.

For $t_k = 1, 2, \ldots,$

1. Compute the **analysis estimate**: Just as for BLUE, let

   $$x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$$

   with

   $$K_k^* = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$$

   which is called the Kalman gain matrix. Moreover, the error covariance is

   $$P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$$

2. Given x_k^a, compute the **forecast**

   $$x_{k+1}^f \stackrel{\text{def.}}{=} M_{k+1} x_k^a.$$

   Thanks to the linearity of M_{k+1}, this is unbiased. Moreover,

   $$e_{k+1}^f = M_{k+1} e_k^a + e_{k+1}^m \quad \text{and} \quad P_{k+1}^f = M_{k+1} P_k^a M_{k+1}^\top + Q_{k+1}$$

Example

Scenario

Suppose you are shipwrecked at sea, a few km from shore. You want to estimate your new coordinates on an hourly basis.

- Before boarding a small lifeboat, you measure your coordinates $(u, v) = (0, v_b)$ with high accuracy. The first axis is parallel to shore, the second axis is perpendicular to shore.
- at hour k, you guess your distance to shore: y_k with variance σ_o^2.
- Denote the true coordinates at time k by $x_k = (u_k, v_k)$. Between time k and $k+1$, assume the boat has drifted with $x_{k+1} = x_k + \xi_k$ whre $\xi_k \sim \mathcal{N}(0, \sigma_m^2 \mathrm{Id}_2)$.
- Assume that the probability that the boat remains at (u_b, v_b) follows $\mathcal{N}(0, \sigma_b^2)$.

Example

- The state vectors are of the form $x_k = (u_k, v_k)^\top$.
- As before, $H_k = (0, 1)$ and $R_k = \sigma_o^2$.
- We have $M_k = \mathrm{Id}_2$ and $Q_k = \sigma_m^2 \mathrm{Id}_2$.
- We have $x_0^f = (u, v)$ and $P_0^f = \sigma_b^2 \mathrm{Id}_2$.

Example

- The state vectors are of the form $x_k = (u_k, v_k)^\top$.
- As before, $H_k = (0, 1)$ and $R_k = \sigma_o^2$.
- We have $M_k = \mathrm{Id}_2$ and $Q_k = \sigma_m^2 \mathrm{Id}_2$.
- We have $x_0^f = (u, v)$ and $P_0^f = \sigma_b^2 \mathrm{Id}_2$.

Assume that $P_k^a = \mathrm{diag}(\lambda_{1,k}, \lambda_{2,k})$ and $P_k^f = \mathrm{diag}(\nu_{1,k}, \nu_{2,k})$ are diagonal matrices (we will check this afterwards).

Example: Analysis step

Kalman filter analysis step

Given x_k^f and P_k^f, set $x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$ where

$$K_k^* = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$$

The error covariance is $P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$.

Example: Analysis step

Kalman filter analysis step

Given x_k^f and P_k^f, set $x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$ where

$$K_k^* = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$$

The error covariance is $P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$.

The Kalman gain is

$$K_k^* = \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Example: Analysis step

Kalman filter analysis step

Given x_k^f and P_k^f, set $x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$ where

$$K_k^* = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$$

The error covariance is $P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$.

The Kalman gain is

$$K_k^* = \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The analysis is

$$\begin{pmatrix} u_k^a \\ v_k^a \end{pmatrix} = \begin{pmatrix} u_k^f \\ v_k^f \end{pmatrix} + K_k^* \left(y_k - (0,1) \begin{pmatrix} u_k^f \\ v_k^f \end{pmatrix} \right) = \begin{pmatrix} u_k^f \\ v_k^f + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} (y_k - v_k^f) \end{pmatrix}$$

Example: Analysis step

Kalman filter analysis step

Given x_k^f and P_k^f, set $x_k^a = x_k^f + K_k(y_k - H_k x_k^f)$ where

$$K_k^* = P_k^f H_k^\top (H_k P_k^f H_k^\top + R_k)^{-1}$$

The error covariance is $P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$.

The Kalman gain is

$$K_k^* = \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

The analysis is

$$\begin{pmatrix} u_k^a \\ v_k^a \end{pmatrix} = \begin{pmatrix} u_k^f \\ v_k^f \end{pmatrix} + K_k^* \left(y_k - (0,1) \begin{pmatrix} u_k^f \\ v_k^f \end{pmatrix} \right) = \begin{pmatrix} u_k^f \\ v_k^f + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} (y_k - v_k^f) \end{pmatrix}$$

The covariance $P_k^a = (\mathrm{Id} - K_k^* H_k) P_k^f$ has diagonal entries

$$\lambda_{1,k} = \nu_{1,k} \quad \text{and} \quad \frac{1}{\lambda_{2,k}} = \frac{1}{\sigma_o^2} + \frac{1}{\nu_{2,k}}$$

Example: Forecast step

Kalman filter forecast step

Given x_k^a and P_k^a, the forecast and associated covariance matrix are

$$x_{k+1}^f \stackrel{\text{def.}}{=} M_{k+1} x_k^a \quad \text{and} \quad P_{k+1}^f = M_{k+1} P_k^a M_{k+1}^\top + Q_{k+1}.$$

Example: Forecast step

Kalman filter forecast step

Given x_k^a and P_k^a, the forecast and associated covariance matrix are

$$x_{k+1}^f \stackrel{\text{def.}}{=} M_{k+1} x_k^a \quad \text{and} \quad P_{k+1}^f = M_{k+1} P_k^a M_{k+1}^\top + Q_{k+1}.$$

- The model does not change our estimate

$$\begin{pmatrix} u_{k+1}^f \\ v_{k+1}^f \end{pmatrix} = \begin{pmatrix} u_k^a \\ v_k^a \end{pmatrix}.$$

Example: Forecast step

Kalman filter forecast step

Given x_k^a and P_k^a, the forecast and associated covariance matrix are

$$x_{k+1}^f \stackrel{\text{def.}}{=} M_{k+1} x_k^a \quad \text{and} \quad P_{k+1}^f = M_{k+1} P_k^a M_{k+1}^\top + Q_{k+1}.$$

- The model does not change our estimate

$$\begin{pmatrix} u_{k+1}^f \\ v_{k+1}^f \end{pmatrix} = \begin{pmatrix} u_k^a \\ v_k^a \end{pmatrix}.$$

- The forecast covariance error diagonal entries are

$$\nu_{1,k+1} = \lambda_{1,k} + \sigma_m^2 \quad \text{and} \quad \nu_{2,k+1} = \lambda_{2,k} + \sigma_m^2$$

Example: summary

To summarise:

$$u^f_{k+1} = u^f_k \quad \text{and} \quad v^f_{k+1} = \frac{\sigma_o^2}{\sigma_o^2 + \nu_{2,k}} v^f_k + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} y_k$$

and

$$\nu_{1,k+1} = \nu_{1,k} + \sigma_m^2 = k\sigma_m^2 + \sigma_b^2 \quad \text{and} \quad \frac{1}{\nu_{2,k+1} - \sigma_m^2} = \frac{1}{\sigma_o^2} + \frac{1}{\nu_{2,k}}$$

Example: summary

To summarise:

$$u_{k+1}^f = u_k^f \quad \text{and} \quad v_{k+1}^f = \frac{\sigma_o^2}{\sigma_o^2 + \nu_{2,k}} v_k^f + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} y_k$$

and

$$\nu_{1,k+1} = \nu_{1,k} + \sigma_m^2 = k\sigma_m^2 + \sigma_b^2 \quad \text{and} \quad \frac{1}{\nu_{2,k+1} - \sigma_m^2} = \frac{1}{\sigma_o^2} + \frac{1}{\nu_{2,k}}$$

Estimate on u: The u estimate will not change and uncertainty grows linearly with time.

Example: summary

To summarise:

$$u^f_{k+1} = u^f_k \quad \text{and} \quad v^f_{k+1} = \frac{\sigma_o^2}{\sigma_o^2 + \nu_{2,k}} v^f_k + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} y_k$$

and

$$\nu_{1,k+1} = \nu_{1,k} + \sigma_m^2 = k\sigma_m^2 + \sigma_b^2 \quad \text{and} \quad \frac{1}{\nu_{2,k+1} - \sigma_m^2} = \frac{1}{\sigma_o^2} + \frac{1}{\nu_{2,k}}$$

Estimate on u: The u estimate will not change and uncertainty grows linearly with time.

Estimate on v: we note that $\nu_{2,k} \geqslant \sigma_m^2$, and

$$\nu_{2,k}\sigma_o^2 = \nu_{2,k}\nu_{2,k+1} - \sigma_m^2\nu_{2,k} + \sigma_o^2\nu_{2,k+1} - \sigma_o^2\sigma_m^2,$$

Example: summary

To summarise:

$$u_{k+1}^f = u_k^f \quad \text{and} \quad v_{k+1}^f = \frac{\sigma_o^2}{\sigma_o^2 + \nu_{2,k}} v_k^f + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} y_k$$

and

$$\nu_{1,k+1} = \nu_{1,k} + \sigma_m^2 = k\sigma_m^2 + \sigma_b^2 \quad \text{and} \quad \frac{1}{\nu_{2,k+1} - \sigma_m^2} = \frac{1}{\sigma_o^2} + \frac{1}{\nu_{2,k}}$$

Estimate on u: The u estimate will not change and uncertainty grows linearly with time.

Estimate on v: we note that $\nu_{2,k} \geqslant \sigma_m^2$, and

$$\nu_{2,k}\sigma_o^2 = \nu_{2,k}\nu_{2,k+1} - \sigma_m^2\nu_{2,k} + \sigma_o^2\nu_{2,k+1} - \sigma_o^2\sigma_m^2,$$

This leads to a fixed point equation $\nu_*^2 - \sigma_m^2\nu_* - \sigma_o^2\sigma_m^2 = 0$ with solution

$$\nu_*^2 = \frac{\sigma_m^2}{2}\left(1 + \sqrt{1 + 4\frac{\sigma_o^2}{\sigma_m^2}}\right)$$

Example: summary

To summarise:

$$u_{k+1}^f = u_k^f \quad \text{and} \quad v_{k+1}^f = \frac{\sigma_o^2}{\sigma_o^2 + \nu_{2,k}} v_k^f + \frac{\nu_{2,k}}{\sigma_o^2 + \nu_{2,k}} y_k$$

and

$$\nu_{1,k+1} = \nu_{1,k} + \sigma_m^2 = k\sigma_m^2 + \sigma_b^2 \quad \text{and} \quad \frac{1}{\nu_{2,k+1} - \sigma_m^2} = \frac{1}{\sigma_o^2} + \frac{1}{\nu_{2,k}}$$

Estimate on u: The u estimate will not change and uncertainty grows linearly with time.

Estimate on v: we note that $\nu_{2,k} \geqslant \sigma_m^2$, and

$$\nu_{2,k}\sigma_o^2 = \nu_{2,k}\nu_{2,k+1} - \sigma_m^2\nu_{2,k} + \sigma_o^2\nu_{2,k+1} - \sigma_o^2\sigma_m^2,$$

This leads to a fixed point equation $\nu_*^2 - \sigma_m^2\nu_* - \sigma_o^2\sigma_m^2 = 0$ with solution

$$\nu_*^2 = \frac{\sigma_m^2}{2}\left(1 + \sqrt{1 + 4\frac{\sigma_o^2}{\sigma_m^2}}\right)$$

Compromise between the uncertainty due the the observation and the uncertainty due to the uncontrolled drift of the boat.

Outline

Best linear unbiased estimate (BLUE)

The Kalman filter

The extended Kalman filter

The stochastic ensemble Kalman filter

3D Var

4D Var

The extended Kalman filter

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

- an initial background state $x^b = x_0^t + e^b \in \mathbb{R}^n$ where $e^b \in \mathcal{N}(0, B)$.
- observations $y_k = H_k[x_k^t] + e_k^o \in \mathbb{R}^p$ for $k = 1, \ldots, K$, where $e_k^o \sim \mathcal{N}(0, R_k)$.
- a model $x_k^t = M_k[x_{k-1}^t] + e_k^m$ for $k = 1, \ldots, K$ where $e_k^m \sim \mathcal{N}(0, Q_k)$.

Assume independence between the background, observation and model.

The extended Kalman filter

Setup

We want to estimate a sequence of true state $(x_k^t)_{k=1}^K \subset \mathbb{R}^n$ given

- an initial background state $x^b = x_0^t + e^b \in \mathbb{R}^n$ where $e^b \in \mathcal{N}(0, B)$.
- observations $y_k = H_k[x_k^t] + e_k^o \in \mathbb{R}^p$ for $k = 1, \ldots, K$, where $e_k^o \sim \mathcal{N}(0, R_k)$.
- a model $x_k^t = M_k[x_{k-1}^t] + e_k^m$ for $k = 1, \ldots, K$ where $e_k^m \sim \mathcal{N}(0, Q_k)$.

Assume independence between the background, observation and model.

What to do if H_k and M_k are nonlinear operators?

The tangent linear

Linearise in the computation of the covariance matrices P_k^a, P_k^f and the Kalman gain matrix K_k.

The tangent linear

Linearise in the computation of the covariance matrices P_k^a, P_k^f and the Kalman gain matrix K_k.

The tangent linear (Jacobian) of M_k and H_k at x are defined to be the $n \times n$ matrices

$$(\mathbb{M}_k(x))_{ij} \stackrel{\text{def.}}{=} \frac{\partial (M_k)_i}{x_j}(x) \quad \text{and} \quad (\mathbb{H}_k(x))_{ij} \stackrel{\text{def.}}{=} \frac{\partial (H_k)_i}{x_j}(x)$$

The extended Kalman filter: the forecast step

The forecast step is simply $x^f_{k+1} = M_{k+1}(x^a_k)$.

To compute the forecast covariance, we look at the linearisation of the error:

$$e^f_{k+1} = x^f_{k+1} - x^t_{k+1} =$$

The forecast step is simply $x^f_{k+1} = M_{k+1}(x^a_k)$.

To compute the forecast covariance, we look at the linearisation of the error:

$$e^f_{k+1} = x^f_{k+1} - x^t_{k+1} = \quad \underbrace{M_{k+1}(x^a_k)}_{} \quad - \quad \underbrace{x^t_{k+1}}_{}$$

The extended Kalman filter: the forecast step

The forecast step is simply $x^f_{k+1} = M_{k+1}(x^a_k)$.

To compute the forecast covariance, we look at the linearisation of the error:

$$e^f_{k+1} = x^f_{k+1} - x^t_{k+1} = \underbrace{M_{k+1}(x^a_k)}_{M_{k+1}(x^a_k - x^t_k + x^t_k)} - \underbrace{x^t_{k+1}}$$

The extended Kalman filter: the forecast step

The forecast step is simply $x^f_{k+1} = M_{k+1}(x^a_k)$.

To compute the forecast covariance, we look at the linearisation of the error:

$$e^f_{k+1} = x^f_{k+1} - x^t_{k+1} = \underbrace{M_{k+1}(x^a_k)}_{M_{k+1}(x^a_k - x^t_k + x^t_k)} - \underbrace{x^t_{k+1}}_{M_{k+1}(x^t_k) + e^m_{k+1}}$$

The extended Kalman filter: the forecast step

The forecast step is simply $x^f_{k+1} = M_{k+1}(x^a_k)$.

To compute the forecast covariance, we look at the linearisation of the error:

$$\begin{aligned} e^f_{k+1} = x^f_{k+1} - x^t_{k+1} = & \underbrace{M_{k+1}(x^a_k)}_{M_{k+1}(x^a_k - x^t_k + x^t_k)} - \underbrace{x^t_{k+1}}_{M_{k+1}(x^t_k) + e^m_{k+1}} \\ & \approx \mathbb{M}_{k+1}(x^t_k) e^a_k - e^m_{k+1}. \end{aligned}$$

The extended Kalman filter: the forecast step

The forecast step is simply $x_{k+1}^f = M_{k+1}(x_k^a)$.

To compute the forecast covariance, we look at the linearisation of the error:

$$
\begin{aligned}
e_{k+1}^f = x_{k+1}^f - x_{k+1}^t = \quad & \underbrace{M_{k+1}(x_k^a)}_{M_{k+1}(x_k^a - x_k^t + x_k^t)} \quad - \quad \underbrace{x_{k+1}^t}_{M_{k+1}(x_k^t) + e_{k+1}^m} \\
\approx\ & \mathbb{M}_{k+1}(x_k^t) e_k^a - e_{k+1}^m .
\end{aligned}
$$

Computation of P_{k+1}^f is replaced by

$$P_{k+1}^f = \mathbb{M}_{k+1}(x_k^t) P_k^a \mathbb{M}_{k+1}(x_k^t)^\top + Q_{k+1}$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

We linearise the analysis error:

$$e_k^a = x_k^a - x_k^t = x_k^f - x_k^t + K_k(y_k - H_k[x_k^f])$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

We linearise the analysis error:

$$\begin{aligned} e_k^a = x_k^a - x_k^t &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^f]) \\ &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^t] + H_k[x_k^t] - H_k[x_k^f - x_k^t + x_k^t]) \end{aligned}$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

We linearise the analysis error:

$$\begin{aligned} e_k^a = x_k^a - x_k^t &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^f]) \\ &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^t] + H_k[x_k^t] - H_k[x_k^f - x_k^t + x_k^t]) \\ &\approx e_k^f + K_k(e_k^o - \mathbb{H}_k[x_k^t]e_k^f) \end{aligned}$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

We linearise the analysis error:

$$\begin{aligned}
e_k^a = x_k^a - x_k^t &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^f]) \\
&= x_k^f - x_k^t + K_k(y_k - H_k[x_k^t] + H_k[x_k^t] - H_k[x_k^f - x_k^t + x_k^t]) \\
&\approx e_k^f + K_k(e_k^o - \mathbb{H}_k[x_k^t]e_k^f) \\
&= (\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])e_k^f + K_k e_k^o
\end{aligned}$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

We linearise the analysis error:

$$\begin{aligned}
e_k^a = x_k^a - x_k^t &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^f]) \\
&= x_k^f - x_k^t + K_k(y_k - H_k[x_k^t] + H_k[x_k^t] - H_k[x_k^f - x_k^t + x_k^t]) \\
&\approx e_k^f + K_k(e_k^o - \mathbb{H}_k[x_k^t]e_k^f) \\
&= (\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])e_k^f + K_k e_k^o
\end{aligned}$$

So, set

$$P_k^a = (\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])P_k^f(\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])^\top + K_k R_k K_k^\top$$

The extended Kalman filter: the analysis step

For **the analysis step**, we compute

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

We linearise the analysis error:

$$\begin{aligned} e_k^a = x_k^a - x_k^t &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^f]) \\ &= x_k^f - x_k^t + K_k(y_k - H_k[x_k^t] + H_k[x_k^t] - H_k[x_k^f - x_k^t + x_k^t]) \\ &\approx e_k^f + K_k(e_k^o - \mathbb{H}_k[x_k^t]e_k^f) \\ &= (\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])e_k^f + K_k e_k^o \end{aligned}$$

So, set

$$P_k^a = (\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])P_k^f(\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])^\top + K_k R_k K_k^\top$$

The Kalman gain matrix which minimises $\mathbb{E}[\mathrm{tr}\,(P_k^a)]$ is therefore

$$K_k = P_k^f\mathbb{H}_k^\top(R_k + \mathbb{H}_k P_k^f \mathbb{H}_k^\top)^{-1}$$

and the analysis covariance is now $P_k^a = (\mathrm{Id} - K_k\mathbb{H}_k[x_k^t])P_k^f$

The extended Kalman filter

We initialise with a state $x_0^f = x^b$ and covariance matrix $P_0^f = \mathbb{E}[e^b(e^b)^\top]$.

For $t_k = 1, 2, \ldots$,

1. Compute the **analysis** estimate: let $\mathbb{H}_k \overset{\text{def.}}{=} \mathbb{H}_k[x_k^f]$.

$$x_k^a = x_k^f + K_k(y_k - H_k[x_k^f])$$

with Kalman gain matrix.

$$K_k = P_k^f \mathbb{H}_k^\top (\mathbb{H}_k P_k^f \mathbb{H}_k^\top + R_k)^{-1}$$

The error covariance is

$$P_k^a = (\mathrm{Id} - K_k^* \mathbb{H}_k) P_k^f$$

2. Given x_k^a, compute the **forecast**

$$x_{k+1}^f \overset{\text{def.}}{=} M_{k+1}[x_k^a].$$

and covariance

$$P_{k+1}^f = \mathbb{M}_{k+1} P_k^a \mathbb{M}_{k+1}^\top + Q_{k+1}$$

where $\mathbb{M}_{k+1} \overset{\text{def.}}{=} \mathbb{M}_{k+1}[x_k^a]$.

Disadvantages of the Kalman filter

1. The storage of the covariance matrices P_k^f. This requires $n(n+1)/2$ scalars to be stored. Much more expensive than storing only the state vectors.
2. We require $2n$ computations with the model M_k for the computation of P_k^f since we apply M_k and $M_k^\top$ on the left and right hand sides.
3. In the extended Kalman filter for nonlinear models, this is an approximation, and the approximation may diverge if the timestep between consecutive updates is too large.

Outline

Best linear unbiased estimate (BLUE)

The Kalman filter

The extended Kalman filter

The stochastic ensemble Kalman filter

3D Var

4D Var

The Kalman filter is expensive in memory due to the storage of covariance matrices. Moreover, the linearisation in the extended Kalman filter results in breaks down when the timestep becomes too large.

The stochastic ensemble Kalman filter

The Kalman filter is expensive in memory due to the storage of covariance matrices. Moreover, the linearisation in the extended Kalman filter results in breaks down when the timestep becomes too large.

The stochastic ensemble Kalman filter (Geir Evensen, 1994) aims to alleviate these issue problems.

The stochastic ensemble Kalman filter

The Kalman filter is expensive in memory due to the storage of covariance matrices. Moreover, the linearisation in the extended Kalman filter results in breaks down when the timestep becomes too large.

The stochastic ensemble Kalman filter (Geir Evensen, 1994) aims to alleviate these issue problems.

Idea: instead of propagating the covariance matrices P_k^f,

- maintain a collection of state vectors (particles) whose variability represent the uncertainty of the system's state.
- the particles are propagated by the model without any linearisation.

The stochastic ensemble Kalman filter: naive formulation

Suppose we now have an ensemble of m particles from the previous forecast step $\{x_i^f\}_{i=1}^m \subset \mathbb{R}^n$. At each time, we have observation y.

1. Compute the empirical mean and covariance:

$$\bar{x}^f \stackrel{\text{def.}}{=} \frac{1}{m}\sum_{i=1}^m x_i^f \quad \text{and} \quad P^f = \frac{1}{m-1}\sum_{i=1}^m (x_i^f - \bar{x}^f)(x_i^f - \bar{x}^f)^\top.$$

2. For each $i = 1, \dots, m$, let

$$x_i^a = x_i^f + K(y_i - H(x_i^f)), \quad \text{where} \quad y_i = y.$$

3. The analysis covariance is

$$P^a = \frac{1}{m-1}\sum_{i=1}^m (x_i^a - \bar{x}^a)(x_i^a - \bar{x}^a)^\top$$

How would this compare with BLUE analysis?

How would this compare with BLUE analysis?

Suppose now that H is linear, from

$$\begin{cases} x_i^a = x_i^f + K(y - H(x_k^f)) \\ \bar{x}^a \overset{\text{def.}}{=} \frac{1}{m}\sum_i x_i^a = \bar{x}^f + K(y - H(\bar{x}^f)) \end{cases}$$

How would this compare with BLUE analysis?

Suppose now that H is linear, from

$$\begin{cases} x_i^a = x_i^f + K(y - H(x_k^f)) \\ \bar{x}^a \stackrel{\text{def.}}{=} \frac{1}{m}\sum_i x_i^a = \bar{x}^f + K(y - H(\bar{x}^f)) \end{cases}$$

we have the ensemble anomalies $e_i^a \stackrel{\text{def.}}{=} x_i^a - \bar{x}^a$ satisfy

$$e_i^a = e_i^f + K(0 - He_i^f) = (\mathrm{Id} - KH)e_i^f$$

How would this compare with BLUE analysis?

Suppose now that H is linear, from

$$\begin{cases} x_i^a = x_i^f + K(y - H(x_k^f)) \\ \bar{x}^a \overset{\text{def.}}{=} \frac{1}{m}\sum_i x_i^a = \bar{x}^f + K(y - H(\bar{x}^f)) \end{cases}$$

we have the ensemble anomalies $e_i^a \overset{\text{def.}}{=} x_i^a - \bar{x}^a$ satisfy

$$e_i^a = e_i^f + K(0 - He_i^f) = (\mathrm{Id} - KH)e_i^f$$

which leads to

$$P^a = \frac{1}{m-1}\sum_{i=1}^{m}(e_i^a)(e_i^a)^\top = (\mathrm{Id} - KH)P^f(\mathrm{Id} - KH)^\top.$$

There is a $KRK^\top$ term missing if we wanted to emulate the BLUE analysis.

The analysis step

Fix:

1. Perturb $y_i = y + u_i$ where $u_i \sim \mathcal{N}(0, R)$.
2. Define $R_u = \frac{1}{m-1} \sum_{i=1}^{m} u_i u_i^\top \quad \overset{m\to\infty}{\longrightarrow} \quad R$
3. compute the Kalman gain matrix as $K = P^f H^\top (HP^f H^\top + R_u)^{-1}$.

The analysis step

Fix:

1. Perturb $y_i = y + u_i$ where $u_i \sim \mathcal{N}(0, R)$.
2. Define $R_u = \frac{1}{m-1} \sum_{i=1}^{m} u_i u_i^\top \quad \xrightarrow{m \to \infty} \quad R$
3. compute the Kalman gain matrix as $K = P^f H^\top (HP^f H^\top + R_u)^{-1}$.

Let $e_i^o \stackrel{\text{def.}}{=} u_i - \bar{u}$ and $\bar{u} = \frac{1}{m} \sum_{i=1}^{m} u_i$. We then have

$$e_i^a = e_i^f + K(e_i^o - He_i^f) = (\mathrm{Id} - KH)e_i^f + Ke_i^o.$$

Then, letting E^a be the matrix with columns $\frac{e_i^a}{\sqrt{m-1}}$ and E^o be the matrix with columns $\frac{e_i^o}{\sqrt{m-1}}$, we have

$$P^a = (E^a)(E^a)^\top = (\mathrm{Id} - KH)P^f(\mathrm{Id} - KH) + KR_uK^\top = (\mathrm{Id} - KH)P^f$$

The forecast step

We let

$$x_i^f = M(x_i^a), \qquad i = 1, \dots, m$$

and

$$\bar{x}^f \stackrel{\text{def.}}{=} \frac{1}{m}\sum_{i=1}^{m} x_i^f \quad \text{and} \quad P^f = \frac{1}{m-1}\sum_{i=1}^{m}(x_i^f - \bar{x}^f)(x_i^f - \bar{x}^f)^\top.$$

The forecast step

We let

$$x_i^f = M(x_i^a), \qquad i = 1, \ldots, m$$

and

$$\bar{x}^f \overset{\text{def.}}{=} \frac{1}{m} \sum_{i=1}^{m} x_i^f \quad \text{and} \quad P^f = \frac{1}{m-1} \sum_{i=1}^{m} (x_i^f - \bar{x}^f)(x_i^f - \bar{x}^f)^\top.$$

NB: Previously, $P_{k+1}^f = \mathbb{M}_{k+1} P_k^a \mathbb{M}_{k+1}^\top + Q_{k+1}$, so we avoid having to apply $\mathbb{M}_{k+1}$ and there is also no need to explicitly compute P_k^a.

No need for linearisation

For the extended Kalman filter, we would have computed using the tangent linear $\mathbb{H}$ in the computation of K: $K = P^f \mathbb{H}^\top (\mathbb{H} P^f \mathbb{H}^\top + R)^{-1}$.

No need for linearisation

For the extended Kalman filter, we would have computed using the tangent linear $\mathbb{H}$ in the computation of K: $K = P^f \mathbb{H}^\top (\mathbb{H} P^f \mathbb{H}^\top + R)^{-1}$.

For the ensemble Kalman filter, this is done without linearisation of H: Consider the matrix products $P^f \mathbb{H}^\top$ and $\mathbb{H} P^f \mathbb{H}^\top$:

$$\bar{y}^f \stackrel{\text{def.}}{=} \frac{1}{m} \sum_{i=1}^{m} H(x_i^f)$$

$$P^f \mathbb{H}^\top = \sum_{i=1}^{m} \frac{(x_i^f - \bar{x}^f)(\mathbb{H}(x_i^f - \bar{x}^f))^\top}{m-1} \approx \sum_{i=1}^{m} \frac{(x_i^f - \bar{x}^f)(H(x_i^f) - \bar{y}^f))^\top}{m-1}$$

No need for linearisation

For the extended Kalman filter, we would have computed using the tangent linear $\mathbb{H}$ in the computation of K: $K = P^f \mathbb{H}^\top (\mathbb{H} P^f \mathbb{H}^\top + R)^{-1}$.

For the ensemble Kalman filter, this is done without linearisation of H: Consider the matrix products $P^f \mathbb{H}^\top$ and $\mathbb{H} P^f \mathbb{H}^\top$:

$$\bar{y}^f \stackrel{\text{def.}}{=} \frac{1}{m} \sum_{i=1}^{m} H(x_i^f)$$

$$P^f \mathbb{H}^\top = \sum_{i=1}^{m} \frac{(x_i^f - \bar{x}^f)(\mathbb{H}(x_i^f - \bar{x}^f))^\top}{m-1} \approx \sum_{i=1}^{m} \frac{(x_i^f - \bar{x}^f)(H(x_i^f) - \bar{y}^f))^\top}{m-1}$$

$$\mathbb{H} P^f \mathbb{H}^\top = \sum_{i=1}^{m} \frac{\mathbb{H}(x_i^f - \bar{x}^f)(\mathbb{H}(x_i^f - \bar{x}^f))^\top}{m-1} \approx \sum_{i=1}^{m} \frac{(H(x_i^f) - \bar{y}^f)(H(x_i^f) - \bar{y}^f))^\top}{m-1}$$

The stochastic ensemble Kalman filter

Initialise $\{x_{i,0}^f\}_{i=1}^m$ and initial error covariance matrix P_0^f.

For $k = 1, 2, \ldots$

Let $y_{i,k} = y_k + u_i$ where $u_i \sim \mathcal{N}(0, R)$.

Analysis step:

- $\bar{x}_k^f \overset{\text{def.}}{=} \frac{1}{m}\sum_{i=1}^m x_{i,k}^f$, $\bar{u} = \frac{1}{m}\sum_{i=1}^m u_i$, $\bar{y}_k^f \overset{\text{def.}}{=} \frac{1}{m}\sum_{i=1}^m H_k[x_{i,k}^f]$.
- $[E_k^f]_i = \frac{1}{\sqrt{m-1}}(x_{i,k}^f - \bar{x}_k^f)_{i=1}^m$ and $[Y_k^f]_i = \frac{H_k[x_{i,k}^f] - u_i - \bar{y}_k^f + \bar{u}}{\sqrt{m-1}}$.
- $K_k = X_k^f(Y_k^f)(Y_k^f(Y_k^f)^\top)^{-1}$.
- For $i = 1, \ldots, m$, $x_{i,k}^a = x_{i,k}^f + K_k(y_{i,k} - H_k(x_{i,k}^f))$,
- $\bar{x}_k^a = \frac{1}{m}\sum_i x_{i,k}^a$ and $P_k^a = \frac{1}{m-1}\sum_{i=1}^m (x_{i,k}^a - \bar{x}^a)(x_{i,k}^a - \bar{x}_k^a)^\top$.

Forecast step:

- For $i = 1, \ldots, m$, $x_{i,k}^f = M(x_{i,k}^a)$
- $P_k^f = \frac{1}{m-1}\sum_{i=1}^m (x_{i,k}^f - \bar{x}_k^f)(x_{i,k}^f - \bar{x}_k^f)^\top$.

Given a positive definite matrix Σ, let $\|v\|_\Sigma \stackrel{\text{def.}}{=} \langle \Sigma v, v \rangle$.

Setup

We want to estimate the true state $x^t \in \mathbb{R}^n$ given

- a background state $x^b \in \mathbb{R}^n$.
- observation $y = H[x^t] + e^o \in \mathbb{R}^p$, where $H \in \mathbb{R}^{p \times n}$ is a matrix, typically, $p \ll n$.

Let $e^b \stackrel{\text{def.}}{=} x^b - x^t$ and $e^o = x^b - x^t$ be the background and observation errors.

They are assumed to be uncorrelated with symmetric covariance matrices $B \stackrel{\text{def.}}{=} \mathbb{E}[e^b(e^b)^\top]$ and $R \stackrel{\text{def.}}{=} \mathbb{E}[e^o(e^o)^\top]$ respectively.

3D Var

Given a positive definite matrix Σ, let $\|v\|_\Sigma \stackrel{\text{def.}}{=} \langle \Sigma v, v \rangle$.

Setup

We want to estimate the true state $x^t \in \mathbb{R}^n$ given

- a background state $x^b \in \mathbb{R}^n$.
- observation $y = H[x^t] + e^o \in \mathbb{R}^p$, where $H \in \mathbb{R}^{p\times n}$ is a matrix, typically, $p \ll n$.

Let $e^b \stackrel{\text{def.}}{=} x^b - x^t$ and $e^o = x^b - x^t$ be the background and observation errors.

They are assumed to be uncorrelated with symmetric covariance matrices $B \stackrel{\text{def.}}{=} \mathbb{E}[e^b(e^b)^\top]$ and $R \stackrel{\text{def.}}{=} \mathbb{E}[e^o(e^o)^\top]$ respectively.

3D Var

We choose x^a to minimise the cost function

$$J(x) = \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\|Hx - y\|^2_{R^{-1}} \tag{3dVar}$$

Notation: For positive definite $A \in \mathbb{R}^{n\times n}$ and $v \in \mathbb{R}^n$, $\|v\|^2_A \stackrel{\text{def.}}{=} \langle Av, v \rangle$.

Lemma 1 (3D Var is equivalent to BLUE)

The minimiser x^a to (3dVar) *is*

$$x^a = x^b + K(y - Hx^b)$$

where $K = (B^{-1} + H^\top R^{-1} H)^{-1} H^\top R^{-1}$. Moreover,

$$K = BH^\top (R + HBH^\top)^{-1} = (B^{-1} + H^\top R^{-1} H)^{-1} H^\top R^{-1}.$$

Lemma 1 (3D Var is equivalent to BLUE)

The minimiser x^a to (3dVar) *is*

$$x^a = x^b + K(y - Hx^b)$$

where $K = (B^{-1} + H^\top R^{-1} H)^{-1} H^\top R^{-1}$. *Moreover,*

$$K = BH^\top (R + HBH^\top)^{-1} = (B^{-1} + H^\top R^{-1} H)^{-1} H^\top R^{-1}.$$

Proof.

Note that

$$\nabla J(x) = B^{-1}(x - x^b) - H^\top R^{-1}(y - Hx).$$

So, the minimiser x^a to (3dVar) satisfies $\nabla J(x^a) = 0$ which yields the first statement.

The second statement follows from the Shermann-Morrison-Woodbury formula. $\square$

Error estimate

Note that $\mathrm{Hess}(J) = B^{-1} + H^\top R^{-1} H$, so $P^a = \mathrm{Hess}(J)^{-1}$ and the analysis precision is proportional to the curvature of J – the narrower the minimum, the better the analysis.

Lemma 2

Let $P^a = \mathbb{E}[(x^a - x^t)(x^a - x^t)^\top]$. *Then* $P^a = \mathrm{Hess}(J)^{-1}$.

Proof.

From $\nabla J(x^a) = 0$, we have $B^{-1}(x^a - x^b) - H^\top R^{-1}(y - Hx^a) = 0$

$\square$

Note that $\mathrm{Hess}(J) = B^{-1} + H^\top R^{-1} H$, so $P^a = \mathrm{Hess}(J)^{-1}$ and the analysis precision is proportional to the curvature of J – the narrower the minimum, the better the analysis.

Lemma 2

Let $P^a = \mathbb{E}[(x^a - x^t)(x^a - x^t)^\top]$*. Then* $P^a = \mathrm{Hess}(J)^{-1}$.

Proof.

From $\nabla J(x^a) = 0$, we have $B^{-1}(x^a - x^b) - H^\top R^{-1}(y - Hx^a) = 0$

$$\implies 0 = B^{-1}(x^a - x^t + x^t - x^b) - H^\top R^{-1}(y - Hx^t + Hx^t - Hx^a)$$

$\square$

Note that $\mathrm{Hess}(J) = B^{-1} + H^\top R^{-1} H$, so $P^a = \mathrm{Hess}(J)^{-1}$ and the analysis precision is proportional to the curvature of J – the narrower the minimum, the better the analysis.

Lemma 2

Let $P^a = \mathbb{E}[(x^a - x^t)(x^a - x^t)^\top]$. *Then* $P^a = \mathrm{Hess}(J)^{-1}$.

Proof.

From $\nabla J(x^a) = 0$, we have $B^{-1}(x^a - x^b) - H^\top R^{-1}(y - Hx^a) = 0$

$$\implies 0 = B^{-1}(x^a - x^t + x^t - x^b) - H^\top R^{-1}(y - Hx^t + Hx^t - Hx^a)$$

$$\implies \underbrace{(B^{-1} + H^\top R^{-1} H)(x^a - x^t)}_{\overset{\text{def.}}{=} v_1} = \underbrace{B^{-1}(x^b - x^t) + H^\top R^{-1}(y - Hx^t)}_{\overset{\text{def.}}{=} v_2}$$

$\square$

Error estimate

Note that $\mathrm{Hess}(J) = B^{-1} + H^\top R^{-1} H$, so $P^a = \mathrm{Hess}(J)^{-1}$ and the analysis precision is proportional to the curvature of J – the narrower the minimum, the better the analysis.

Lemma 2

Let $P^a = \mathbb{E}[(x^a - x^t)(x^a - x^t)^\top]$. Then $P^a = \mathrm{Hess}(J)^{-1}$.

Proof.

From $\nabla J(x^a) = 0$, we have $B^{-1}(x^a - x^b) - H^\top R^{-1}(y - Hx^a) = 0$

$$
\begin{aligned}
&\implies 0 = B^{-1}(x^a - x^t + x^t - x^b) - H^\top R^{-1}(y - Hx^t + Hx^t - Hx^a)\\
&\implies \underbrace{(B^{-1} + H^\top R^{-1} H)(x^a - x^t)}_{\overset{\text{def.}}{=} v_1} = \underbrace{B^{-1}(x^b - x^t) + H^\top R^{-1}(y - Hx^t)}_{\overset{\text{def.}}{=} v_2}
\end{aligned}
$$

Considering $\mathbb{E}[v_i^\top v_i]$ for $i = 1, 2$ and since x^b and y are uncorrelated, we have

$$\mathrm{Hess}(J) P_a \mathrm{Hess}(J) = B^{-1} + H^\top R^{-1} H = \mathrm{Hess}(J)$$

which yields $P_a = \mathrm{Hess}(J)^{-1}$. □

Dual formulation

Note that J is convex and we can write $\min_{x\in\mathbb{R}^n} J(x)$ as

$$\min_{x,z\in\mathbb{R}^n} \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\|z\|^2_{R^{-1}} \text{ s.t. } Hx - y = z$$

Dual formulation

Note that J is convex and we can write $\min_{x \in \mathbb{R}^n} J(x)$ as

$$\min_{x,z \in \mathbb{R}^n} \frac{1}{2} \left\| x - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \|z\|_{R^{-1}}^2 \text{ s.t. } Hx - y = z$$

The Lagrange function is a function over $\lambda \in \mathbb{R}^p$

$$H(\lambda) = \min_{x,z} \frac{1}{2} \left\| x - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \|z\|_{R^{-1}}^2 + \langle \lambda, z - Hx + y \rangle$$

Dual formulation

Note that J is convex and we can write $\min_{x\in\mathbb{R}^n} J(x)$ as

$$\min_{x,z\in\mathbb{R}^n} \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\|z\|^2_{R^{-1}} \text{ s.t. } Hx - y = z$$

The Lagrange function is a function over $\lambda \in \mathbb{R}^p$

$$H(\lambda) = \min_{x,z} \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\|z\|^2_{R^{-1}} + \langle\lambda,\, z - Hx + y\rangle$$

The optimum is achieved at $x = x^b + BH^\top\lambda$ and $z = -R\lambda$, so

$$H(\lambda) = -\frac{1}{2}\left\|H^\top\lambda\right\|^2_B - \frac{1}{2}\|\lambda\|^2_R + \langle\lambda,\, -Hx^b + y\rangle$$

The dual problem is therefore

$$\sup_{\lambda\in\mathbb{R}^p} -\frac{1}{2}\langle(HBH^\top + R)\lambda,\, \lambda\rangle - \langle\lambda,\, y - Hx^b\rangle$$

Dual formulation

Note that J is convex and we can write $\min_{x\in\mathbb{R}^n} J(x)$ as

$$\min_{x,z\in\mathbb{R}^n} \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\|z\|^2_{R^{-1}} \text{ s.t. } Hx - y = z$$

The Lagrange function is a function over $\lambda \in \mathbb{R}^p$

$$H(\lambda) = \min_{x,z} \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\|z\|^2_{R^{-1}} + \langle \lambda,\, z - Hx + y\rangle$$

The optimum is achieved at $x = x^b + BH^\top \lambda$ and $z = -R\lambda$, so

$$H(\lambda) = -\frac{1}{2}\left\|H^\top \lambda\right\|^2_{B} - \frac{1}{2}\|\lambda\|^2_{R} + \langle \lambda,\, -Hx^b + y\rangle$$

The dual problem is therefore

$$\sup_{\lambda\in\mathbb{R}^p} -\frac{1}{2}\langle (HBH^\top + R)\lambda,\, \lambda\rangle - \langle \lambda,\, y - Hx^b\rangle$$

The dual formulation is called the **Physical Statistical space Assimilation System (PSAS)** and has the advantage of operating over the measurement space $\mathbb{R}^p$ as opposed to $\mathbb{R}^n$ (recall that often $p \ll n$).

Advantages of 3D Var

Computational:

- BLUE requires the storage and inversion of a large matrix
- 3D var is simply the minimisation of a function J, we only need to compute the product of B^{-1} and R^{-1} against vectors several times – this is computationally less demanding.

Computational:

- BLUE requires the storage and inversion of a large matrix
- 3D var is simply the minimisation of a function J, we only need to compute the product of B^{-1} and R^{-1} against vectors several times – this is computationally less demanding.

The nonlinear case: There is no need to explicitly linearise. For

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

the gradient is

$$\nabla J(x) = B^{-1}(x - x^b) - \mathbb{H}^\top R^{-1}(y - H[x])$$

where $\mathbb{H}$ is the so called *tangent linear* of H at x.

Outline

4D Var

In 4D Var is a sequential extension of 3D Var ('4' refers to the time dimension).

Setup

We assume

- a given background $x^b = x^t + e^b$ with $e^b \in \mathcal{N}(0, B)$
- observations for $k = 1, \ldots, K$,

$$y_k = H_k[x_k^t] + e_k^o, \qquad e_k^o \sim \mathcal{N}(0, R_k).$$

- the true states x_k^t follow some model $x_k^t = M_k[x_{k-1}^t]$ for $k = 1, \ldots, K$.

4D Var

In 4D Var is a sequential extension of 3D Var ('4' refers to the time dimension).

Setup

We assume

- a given background $x^b = x^t + e^b$ with $e^b \in \mathcal{N}(0, B)$
- observations for $k = 1, \ldots, K$,

$$y_k = H_k[x_k^t] + e_k^o, \qquad e_k^o \sim \mathcal{N}(0, R_k).$$

- the true states x_k^t follow some model $x_k^t = M_k[x_{k-1}^t]$ for $k = 1, \ldots, K$.

4D Var

The 4d var estimate is the minimiser of the cost

$$J(x_0) = \frac{1}{2}\left\|x_0 - x^b\right\|_{B^{-1}}^2 + \frac{1}{2}\sum_{k=0}^{K}\|H_k(x_k) - y_k\|_{R_k^{-1}}^2 \qquad \text{s.t. } x_k = M_k(x_{k-1})$$

This is called **strong constraint 4D Var**: the model is assumed to be exact.

4D Var: unconstrained

We first assume that M_k and H_k are both linear.

We can write $x_{k+1} = M_{k+1} M_k \dots M_1 x_0$, so writing

$$d_k = y_k - H_k M_k M_{k-1} \cdots M_2 M_1 x_0 \quad \text{and} \quad \Delta_k = R_k^{-1} d_k$$

we have

$$J(x_0) = \frac{1}{2} \left\| x_0 - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| d_k \|_{R_i^{-1}}^2$$

The gradient is

4D Var: unconstrained

We first assume that M_k and H_k are both linear.

We can write $x_{k+1} = M_{k+1} M_k \dots M_1 x_0$, so writing

$$d_k = y_k - H_k M_k M_{k-1} \cdots M_2 M_1 x_0 \quad \text{and} \quad \Delta_k = R_k^{-1} d_k$$

we have

$$J(x_0) = \frac{1}{2} \left\| x_0 - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| d_k \|_{R_i^{-1}}^2$$

The gradient is

$$\nabla_{x_0} J = B^{-1}(x_0 - x^b) - \sum_{k=0}^{K} (M_1^\top M_2^\top \dots M_k^\top) H_k^\top \Delta_k$$

4D Var: unconstrained

We first assume that M_k and H_k are both linear.

We can write $x_{k+1} = M_{k+1} M_k \dots M_1 x_0$, so writing

$$d_k = y_k - H_k M_k M_{k-1} \cdots M_2 M_1 x_0 \quad \text{and} \quad \Delta_k = R_k^{-1} d_k$$

we have

$$J(x_0) = \frac{1}{2} \left\| x_0 - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| d_k \|_{R_i^{-1}}^2$$

The gradient is

$$\nabla_{x_0} J = B^{-1}(x_0 - x^b) - \sum_{k=0}^{K} (M_1^\top M_2^\top \dots M_k^\top) H_k^\top \Delta_k$$

$$= B^{-1}(x_0 - x^b) - \left(H_0^\top \Delta_0 + M_1^\top \left(H_1^\top \Delta_1 + M_2^\top \left(H_2^\top \Delta_2 + \cdots + M_K^\top \left(H_K^\top \Delta_K \right) \right) \right) \right)$$

4D Var: Computation of the gradient

We can therefore compute the gradient

$$B^{-1}(x_0 - x^b) - \left(H_0^\top \Delta_0 + M_1^\top \left(H_1^\top \Delta_1 + M_2^\top \left(H_2^\top \Delta_2 + \cdots + M_K^\top \left(H_K^\top \Delta_K\right)\right)\right)\right)$$

as follows

We can therefore compute the gradient

$$B^{-1}(x_0 - x^b) - \left(H_0^\top \Delta_0 + M_1^\top \left(H_1^\top \Delta_1 + M_2^\top \left(H_2^\top \Delta_2 + \cdots + M_K^\top \left(H_K^\top \Delta_K\right)\right)\right)\right)$$

as follows

1. First compute x_k recursively using the resolvent matrix M_k and initial state x_0.

We can therefore compute the gradient

$$B^{-1}(x_0 - x^b) - \left(H_0^\top \Delta_0 + M_1^\top \left(H_1^\top \Delta_1 + M_2^\top \left(H_2^\top \Delta_2 + \cdots + M_K^\top \left(H_K^\top \Delta_K\right)\right)\right)\right)$$

as follows

1. First compute x_k recursively using the resolvent matrix M_k and initial state x_0.
2. Compute $\Delta_k = R_k^{-1}(y_k - H_k x_k)$.

We can therefore compute the gradient

$$B^{-1}(x_0-x^b)-\left(H_0^\top\Delta_0+M_1^\top\left(H_1^\top\Delta_1+M_2^\top\left(H_2^\top\Delta_2+\cdots+M_K^\top\left(H_K^\top\Delta_K\right)\right)\right)\right)$$

as follows

1. First compute x_k recursively using the resolvent matrix M_k and initial state x_0.
2. Compute $\Delta_k = R_k^{-1}(y_k - H_k x_k)$.
3. Define $\tilde{x}_k \in \mathbb{R}^n$ (the adjoint state variable). Initialise with $\tilde{x}_K = H_K^\top \Delta_K$. Then, given $\tilde{x}_{k+1}$, $\tilde{x}_k = H_k^\top \Delta_k + M_{k+1}^\top \tilde{x}_{k+1}$.

4D Var: Computation of the gradient

We can therefore compute the gradient

$$B^{-1}(x_0-x^b)-\left(H_0^\top\Delta_0+M_1^\top\left(H_1^\top\Delta_1+M_2^\top\left(H_2^\top\Delta_2+\cdots+M_K^\top\left(H_K^\top\Delta_K\right)\right)\right)\right)$$

as follows

1. First compute x_k recursively using the resolvent matrix M_k and initial state x_0.
2. Compute $\Delta_k = R_k^{-1}(y_k - H_k x_k)$.
3. Define $\tilde{x}_k \in \mathbb{R}^n$ (the adjoint state variable). Initialise with $\tilde{x}_K = H_K^\top \Delta_K$. Then, given $\tilde{x}_{k+1}$, $\tilde{x}_k = H_k^\top \Delta_k + M_{k+1}^\top \tilde{x}_{k+1}$.
4. $\nabla_{x_0} J = B^{-1}(x_0 - x^b) - \tilde{x}_0$.

4D Var: Computation of the gradient

We can therefore compute the gradient

$$B^{-1}(x_0 - x^b) - \left(H_0^\top \Delta_0 + M_1^\top \left(H_1^\top \Delta_1 + M_2^\top \left(H_2^\top \Delta_2 + \cdots + M_K^\top \left(H_K^\top \Delta_K\right)\right)\right)\right)$$

as follows

1. First compute x_k recursively using the resolvent matrix M_k and initial state x_0.
2. Compute $\Delta_k = R_k^{-1}(y_k - H_k x_k)$.
3. Define $\tilde{x}_k \in \mathbb{R}^n$ (the adjoint state variable). Initialise with $\tilde{x}_K = H_K^\top \Delta_K$. Then, given $\tilde{x}_{k+1}$, $\tilde{x}_k = H_k^\top \Delta_k + M_{k+1}^\top \tilde{x}_{k+1}$.
4. $\nabla_{x_0} J = B^{-1}(x_0 - x^b) - \tilde{x}_0$.

Remark: In practice, M_k is computed using a numerical code of thousands of lines. It is common to resort to auto-differentiation to compute $M_k^\top$, given the code of M_k.

Error estimate Just as for 3D Var, one can show that $P^a = \mathrm{Hess}(J)^{-1}$.

4D Var: error estimate and the nonlinear case

Error estimate Just as for 3D Var, one can show that $P^a = \mathrm{Hess}(J)^{-1}$.

The nonlinear case Let $\mathbb{M}$ and $\mathbb{H}$ be the tangent linear of M and H with respect to x. Let $d_k = y_k - H_k M_k(M_{k-1}(\ldots(M_2(M_1 x_0)))$ and Δ_k be as before. By Leibnitz's rule,

$$\nabla_{x_0} J = -\sum_{k=0}^{K} \left(\mathbb{M}_1^\top(x_0)\mathbb{M}_2^\top(x_1)\ldots\mathbb{M}_k^\top(x_{k-1})\right)\mathbb{H}_k^\top(x_k)\Delta_k.$$

Note only do we need to know the adjoint, but also the differential (tangent linear) of M_k and H_k. This can again be done using automatic differentiation.

Equivalence to Kalman filters

In the case of no model errors and when M_k and H_k are both linear, and M_k are invertible,

- one can show that the output x_K of 4D-Var is precisely the output of the Kalman filter.
- however, there is no equivalence between the intermediate outputs x_k for $k < K$.

1. Minimisation over any x_j is equivalent

The evolution model is assumed to be perfect. So we can minimise over any x_j for $j = 0, \ldots, K$.

Write for $k \geqslant \ell$, $M_{k,\ell} \stackrel{\text{def.}}{=} M_k M_{k-1} \ldots M_{\ell+1}$ and for $k < \ell$, $M_{k,\ell} = M_{\ell,k}^{-1}$.

1. Minimisation over any x_j is equivalent

The evolution model is assumed to be perfect. So we can minimise over any x_j for $j = 0, \ldots, K$.

Write for $k \geqslant \ell$, $M_{k,\ell} \overset{\text{def.}}{=} M_k M_{k-1} \ldots M_{\ell+1}$ and for $k < \ell$, $M_{k,\ell} = M_{\ell,k}^{-1}$.

We have

$$x_0 \in \operatorname{argmin}_x J(x) = \frac{1}{2} \left\| x - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| H_k(M_{k,0} x) - y_k \|_{R_k^{-1}}^2$$

if and only if $M_{j,0} x_0 = x_j$ where writing $x_j^b = M_{j,0} x^b$ and $B_j^{-1} \overset{\text{def.}}{=} M_{0,j}^\top B_0^{-1} M_{0,j}$

$$x_j \in \operatorname{argmin}_x \tilde{J}_j(x) \overset{\text{def.}}{=} \frac{1}{2} \left\| x - x_j^b \right\|_{B_j^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| H_k(M_{k,j} x) - y_k \|_{R_k^{-1}}^2$$

1. Minimisation over any x_j is equivalent

The evolution model is assumed to be perfect. So we can minimise over any x_j for $j = 0, \dots, K$.

Write for $k \geqslant \ell$, $M_{k,\ell} \stackrel{\text{def.}}{=} M_k M_{k-1} \dots M_{\ell+1}$ and for $k < \ell$, $M_{k,\ell} = M_{\ell,k}^{-1}$.

We have

$$x_0 \in \operatorname{argmin}_x J(x) = \frac{1}{2} \left\| x - x^b \right\|_{B^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| H_k(M_{k,0} x) - y_k \|_{R_k^{-1}}^2$$

if and only if $M_{j,0} x_0 = x_j$ where writing $x_j^b = M_{j,0} x^b$ and $B_j^{-1} \stackrel{\text{def.}}{=} M_{0,j}^\top B_0^{-1} M_{0,j}$

$$x_j \in \operatorname{argmin}_x \tilde{J}_j(x) \stackrel{\text{def.}}{=} \frac{1}{2} \left\| x - x_j^b \right\|_{B_j^{-1}}^2 + \frac{1}{2} \sum_{k=0}^{K} \| H_k(M_{k,j} x) - y_k \|_{R_k^{-1}}^2$$

Let $P_j^a = \mathbb{E}[(x_j^a - x_j^t)(x_j^a - x_j^t)^\top]$ and $\mathcal{H}_j$ be the Hessian of J_j at its optimum x_j^a

$$(P_j^a)^{-1} = \mathcal{H}_j = B_j^{-1} + \sum_{k=0}^{K} M_{k,j}^\top H_k^\top R_k^{-1} H_k M_{k,j} = M_{\ell,j}^\top (P_\ell^a)^{-1} M_{\ell,j}.$$

2. Transferability of optimality

Minimising over $[0, K]$ is equivalent to minimising over $[0, m]$ and $[m, K]$:

Lemma 3

Consider the following:

- *Let* x_0 *be the minimiser of* $J(x) = \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\sum_{k=0}^{K}\|H_k(M_{k,0}x) - y_k\|^2_{R_k^{-1}}$.
- *Let* $m < K$ *and* x^a *be the minimiser of* $J_m(x) = \frac{1}{2}\left\|x - x^b\right\|^2_{B^{-1}} + \frac{1}{2}\sum_{k=0}^{m}\|H_k(M_{k,0}x) - y_k\|^2_{R_k^{-1}}$. *Then, by defining* $P^a \stackrel{\text{def.}}{=} \mathcal{H}_{x^a}^{-1}$ *where* $\mathcal{H}_{x^a}$ *is the Hessian of* J_m *at* x^a*, let*

$$\hat{x}_0 \in \operatorname{argmin}_x \frac{1}{2}\|x - x^a\|^2_{\mathcal{H}_{x^a}^{-1}} + \frac{1}{2}\sum_{k=m+1}^{K}\|H_k(M_{k,0}x) - y_k\|^2_{R_k^{-1}}.$$

Then, $\hat{x}_0 = x_0$.

2. Transferability of optimality

Proof.

Define $J_{K,m} \overset{\text{def.}}{=} \frac{1}{2}\sum_{k=m+1}^{K} \|H_k(M_{k,0}x) - y_k\|^2_{R_k^{-1}}$.

NB: $J(x) = J_m(x) + J_{K,m}(x)$.

Since J_m is a quadratic function, it coincides with its 2nd order Taylor expansion around x^a:

$$J_m(x) = J_m(x^a) + \nabla J_m(x^a)(x - x^a) + \frac{1}{2}(x - x^a)^\top \mathcal{H}_{x^a}(x - x^a).$$

By optimality of x^a, $\nabla J_m(x^a) = 0$. Therefore,

$$J(x) = J_m(x^a) + \frac{1}{2}\|x - x^a\|^2_{\mathcal{H}_{x^a}} + J_{K,m}(x).$$

Therefore,

$$\operatorname{argmin}_x J(x) = \operatorname{argmin}_x \frac{1}{2}\|x - x^a\|^2_{\mathcal{H}_{x^a}} + J_{K,m}(x)$$

as required. □

Weak constrained 4D Var

When the model is inexact $x^t_{k+1} = M_{k+1} x^t_k + e^m_k$ and consider the cost

$$J(x_0, \ldots, x_K) = \frac{1}{2} \left\| x - x^b \right\|^2_{B^{-1}} + \sum_{i=0}^{j} \left\| H_i x_i - y_i \right\|_{R_i^{-1}} + \left\| M_i x_{i-1} - x_i \right\|^2_{Q_i^{-1}} .$$

This is called **weak constrained 4D Var**.

However, this requires optimisation over $x_0, \ldots, x_j$ and is therefore much more computationally expensive.

Summary

Data assimilation is a set of statistical tools to improve knowledge of present or future system states by combining experimental data and theoretical knowledge of the system.

- BLUE is a statistical interpolation technique for combining observation with an a-priori guess. The BLUE estimate can be written as the solution to a variational problem, 3d Var.
- The Kalman filter is a sequential statistical interpolation technique. It alternates between performing a analysis step (BLUE), and a forecast step (propagating states using a known model).
- 4D Var is a sequential version of 3D Var, and can be seen to be equivalent to the Kalman filter in the case of linear observation and model operators.
- Extensions of the Kalman filter: extended Kalman filter, ensemble Kalman filter.