Introduction to Data Assimilation with 4D-Var and its relation to Tikhonov regularisation

Melina Freitag

Department of Mathematical Sciences
University of Bath

INVERT Centre Bath
22nd August 2008
1 Introduction

2 Variational Data Assimilation
   - Least square estimation
   - Examples
   - Kalman Filter
   - Problems and Issues

3 Tikhonov regularisation

4 Plan and work in progress
Outline

1 Introduction

2 Variational Data Assimilation
   - Least square estimation
   - Examples
   - Kalman Filter
   - Problems and Issues

3 Tikhonov regularisation

4 Plan and work in progress
What is Data Assimilation?

Loose definition

*Estimation* and *prediction* (analysis) of an unknown, true state by combining *observations* and *system dynamics* (model output).
What is Data Assimilation?

**Loose definition**

**Estimation** and **prediction** (analysis) of an unknown, true state by combining **observations** and **system dynamics** (model output).

**Some examples**

- Navigation
- Medical imaging
- **Numerical weather prediction**
Data Assimilation in NWP

Estimate the state of the atmosphere $x_i$.

<table>
<thead>
<tr>
<th>Observations $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Satellites</td>
</tr>
<tr>
<td>Ships and buoys</td>
</tr>
<tr>
<td>Surface stations</td>
</tr>
<tr>
<td>Aeroplanes</td>
</tr>
</tbody>
</table>
Data Assimilation in NWP

Estimate the state of the atmosphere $x_i$.

<table>
<thead>
<tr>
<th>A priori information $x^B$</th>
<th>Observations $y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>background state (usual previous forecast)</td>
<td>Satellites</td>
</tr>
<tr>
<td></td>
<td>Ships and buoys</td>
</tr>
<tr>
<td></td>
<td>Surface stations</td>
</tr>
<tr>
<td></td>
<td>Aeroplanes</td>
</tr>
</tbody>
</table>
Data Assimilation in NWP

Estimate the state of the atmosphere \( x_i \).

A priori information \( x^B \)
- background state (usual previous forecast)

Models
- a model how the atmosphere evolves in time (imperfect)

\[ x_{i+1} = M(x_i) \]

Observations \( y \)
- Satellites
- Ships and buoys
- Surface stations
- Aeroplanes
## Data Assimilation in NWP

Estimate the **state of the atmosphere** $\mathbf{x}_i$.

### A priori information $\mathbf{x}^B$
- background state (usual previous forecast)

### Models
- a model how the atmosphere evolves in time (imperfect)
  \[
  \mathbf{x}_{i+1} = M(\mathbf{x}_i)
  \]
- a function linking model space and observation space (imperfect)
  \[
  \mathbf{y}_i = H(\mathbf{x}_i)
  \]

### Observations $\mathbf{y}$
- Satellites
- Ships and buoys
- Surface stations
- Aeroplanes
## Data Assimilation in NWP

Estimate the **state of the atmosphere** $\mathbf{x}_i$.

### A priori information $\mathbf{x}^B$
- background state (usual previous forecast)

### Models
- a model how the atmosphere evolves in time (imperfect)
  \[ \mathbf{x}_{i+1} = M(\mathbf{x}_i) \]
- a function linking model space and observation space (imperfect)
  \[ \mathbf{y}_i = H(\mathbf{x}_i) \]

### Observations $\mathbf{y}$
- Satellites
- Ships and buoys
- Surface stations
- Aeroplanes

### Assimilation algorithms
- used to find an (approximate) state of the atmosphere $\mathbf{x}_i$ at times $i$ (usually $i = 0$)
- using this state a forecast for future states of the atmosphere can be obtained
- $\mathbf{x}^A$: Analysis (estimation of the true state after the DA)
Figure: Background state $x^B$
Schematics of DA

Figure: Observations $y$
Figure: Analysis $x^A$ (consistent with observations and model dynamics)
## Underdeterminacy

- Size of the state vector $\mathbf{x}$: $432 \times 320 \times 50 \times 7 = \mathcal{O}(10^7)$
## Underdeterminacy

- Size of the state vector \( \mathbf{x} \): \( 432 \times 320 \times 50 \times 7 = \mathcal{O}(10^7) \)
- Number of observations (size of \( \mathbf{y} \)): \( \mathcal{O}(10^5 - 10^6) \)
Data Assimilation in NWP

Underdeterminacy

- Size of the state vector $\mathbf{x}$: $432 \times 320 \times 50 \times 7 = \mathcal{O}(10^7)$
- Number of observations (size of $\mathbf{y}$): $\mathcal{O}(10^5 - 10^6)$
- Operator $H$ (nonlinear!) maps from state space into observations space: $\mathbf{y} = H(\mathbf{x})$
At each time step $i$

$$
\mathbf{x}^A(k) = \mathbf{x}^B(k) + \frac{\sum_{l=1}^{n} w(lk)(\mathbf{y}(l) - \mathbf{x}^B(l))}{\sum_{l=1}^{n} w(lk)}
$$

$$
w(lk) = \max \left( 0, \frac{R^2 - d_{lk}^2}{R^2 + d_{lk}^2} \right)
$$

$d_{lk}$ measures the distance between points $l$ and $k$. 
Outline

1. Introduction

2. Variational Data Assimilation
   - Least square estimation
   - Examples
   - Kalman Filter
   - Problems and Issues

3. Tikhonov regularisation

4. Plan and work in progress
Data Assimilation in NWP

Estimate the state of the atmosphere $\mathbf{x}_i$.

**Apriori information $\mathbf{x}^B$**

- background state (usual previous forecast) has errors!

**Models**

- a model how the atmosphere evolves in time (imperfect)
  
  \[ \mathbf{x}_{i+1} = M(\mathbf{x}_i) + \text{error} \]

- a function linking model space and observation space (imperfect)
  
  \[ \mathbf{y}_i = H(\mathbf{x}_i) + \text{error} \]

**Observations $\mathbf{y}$ has errors!**

- Satellites
- Ships and buoys
- Surface stations
- Airplanes

**Assimilation algorithms**

- used to find an (approximate) state of the atmosphere $\mathbf{x}_i$ at times $i$ (usually $i = 0$)
- using this state a forecast for future states of the atmosphere can be obtained
- $\mathbf{x}^A$: Analysis (estimation of the true state after the DA)
Modelling the errors

- background error $\varepsilon^B = x^B - x^\text{Truth}$ of average $\bar{\varepsilon}^B$ and covariance

$$
B = (\varepsilon^B - \bar{\varepsilon}^B)(\varepsilon^B - \bar{\varepsilon}^B)^T
$$
### Modelling the errors

- **Background error** $\varepsilon^B = \mathbf{x}^B - \mathbf{x}^{\text{Truth}}$ of average $\overline{\varepsilon}^B$ and covariance

$$ \mathbf{B} = (\varepsilon^B - \overline{\varepsilon}^B)(\varepsilon^B - \overline{\varepsilon}^B)^T $$

- **Observation error** $\varepsilon^O = \mathbf{y} - H(\mathbf{x}^{\text{Truth}})$ of average $\overline{\varepsilon}^O$ and covariance

$$ \mathbf{R} = (\varepsilon^O - \overline{\varepsilon}^O)(\varepsilon^O - \overline{\varepsilon}^O)^T $$
Error variables

Modelling the errors

- background error $\varepsilon^B = x^B - x^\text{Truth}$ of average $\bar{\varepsilon}^B$ and covariance
  
  \[ B = (\varepsilon^B - \bar{\varepsilon}^B)(\varepsilon^B - \bar{\varepsilon}^B)^T \]

- observation error $\varepsilon^O = y - H(x^\text{Truth})$ of average $\bar{\varepsilon}^O$ and covariance
  
  \[ R = (\varepsilon^O - \bar{\varepsilon}^O)(\varepsilon^O - \bar{\varepsilon}^O)^T \]

- analysis error $\varepsilon^A = x^A - x^\text{Truth}$ of average $\bar{\varepsilon}^A$ and covariance
  
  \[ A = (\varepsilon^A - \bar{\varepsilon}^A)(\varepsilon^A - \bar{\varepsilon}^A)^T \]

- measure of the analysis error that we want to minimise

  \[ \text{tr}(A) = \|\varepsilon^A - \bar{\varepsilon}^A\|^2 \]
Assumptions

- Linearised observation operator: $H(x) - H(x^B) = H(x - x^B)$
- Nontrivial errors: $B$, $R$ are positive definite
- Unbiased errors: $x^B - x_{\text{Truth}} = y - H(x_{\text{Truth}}) = 0$
- Uncorrelated errors: $(x^B - x_{\text{Truth}})(y - H(x_{\text{Truth}}))^T = 0$
Optimal least-squares estimator

<table>
<thead>
<tr>
<th>Cost function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solution of the variational optimisation problem $x^A = \arg \min J(x)$ where</td>
</tr>
<tr>
<td>$J(x) = (x - x^B)^T B^{-1} (x - x^B) + (y - H(x))^T R^{-1} (y - H(x))$</td>
</tr>
<tr>
<td>$= J_B(x) + J_O(x)$</td>
</tr>
</tbody>
</table>
Optimal least-squares estimator

### Cost function

Solution of the variational optimisation problem $x^A = \arg \min J(x)$ where

$$
J(x) = (x - x^B)^T B^{-1} (x - x^B) + (y - H(x))^T R^{-1} (y - H(x))
$$

$$
= J_B(x) + J_O(x)
$$

### Interpolation equations

$$
x^A = x^B + K (y - H(x^B)), \quad \text{where}
$$

$$
K = BH^T (HBH^T + R)^{-1} \quad K \ldots \text{gain matrix}
$$
Conditional probabilities

Non-Gaussian PDF’s (probability density function)

- $P(x)$ is a priori PDF (background)
- $P(y|x)$ is the observation PDF (likelihood of the observations given background $x$)
Conditional probabilities

Non-Gaussian PDF’s (probability density function)

- $P(x)$ is a priori PDF (background)
- $P(y|x)$ is the observation PDF (likelihood of the observations given background $x$)
- $P(x|y)$ conditional probability of the model state given the observations, Bayes theorem:

$$\arg_x \max P(x|y) = \arg_x \max \frac{P(y|x)P(x)}{P(y)}$$
Conditional probabilities

Non-Gaussian PDF’s (probability density function)

- $P(x)$ is a priori PDF (background)
- $P(y|x)$ is the observation PDF (likelihood of the observations given background $x$)
- $P(x|y)$ conditional probability of the model state given the observations, Bayes theorem:

$$\arg_x \max P(x|y) = \arg_x \max \frac{P(y|x)P(x)}{P(y)}$$

Gaussian PDF’s

$$P(x|y) = c_1 \exp \left( - (x - x^B)^T B^{-1} (x - x^B) \right) \cdot c_2 \exp \left( - (y - H(x))^T R^{-1} (y - H(x)) \right)$$

$x^A$ is the maximum a posteriori estimator of $x^{\text{Truth}}$. Maximising $P(x|y)$ equivalent to minimising $J(x)$.
A simple scalar illustration

Room temperature

- $T^O$ observation with standard deviation $\sigma_O$
- $T^B$ background with standard deviation $\sigma_B$
A simple scalar illustration

Room temperature

- $T^O$ observation with standard deviation $\sigma_O$
- $T^B$ background with standard deviation $\sigma_B$
- $T^A = T^B + k(T^O - T^B)$ with error variance $\sigma^2_A = (1 - k)^2\sigma^2_B + k^2\sigma^2_O$
- optimal $k$ which minimises error variance

$$k = \frac{\sigma^2_B}{\sigma^2_B + \sigma^2_O}$$

- equivalent to minimising

$$J(T) = \frac{(T - T^B)^2}{\sigma^2_B} + \frac{(T - T^O)^2}{\sigma^2_O}$$

and then

$$\frac{1}{\sigma^2_A} = \frac{1}{\sigma^2_B} + \frac{1}{\sigma^2_O}$$
Optimal interpolation

Computation of

\[ x^A = x^B + K(y - H(x^B)) \]

\[ K = BH^T(HBH^T + R)^{-1} \quad K \ldots \text{gain matrix} \]
Optimal interpolation

Computation of

\[ x^A = x^B + K(y - H(x^B)) \]

\[ K = BH^T (HBH^T + R)^{-1} \quad K \ldots \text{gain matrix} \]

- expensive!
3D-Var

Minimisation of

\[ J(x) = (x - x^B)^T B^{-1} (x - x^B) + (y - H(x))^T R^{-1} (y - H(x)) \]

\[ = J_B(x) + J_O(x) \]
Three-dimensional variational assimilation (3D-Var)

3D-Var
Minimisation of

\[ J(x) = (x - x^B)^T B^{-1} (x - x^B) + (y - H(x))^T R^{-1} (y - H(x)) = J_B(x) + J_O(x) \]

- avoids computation of \( K \) by using a descent algorithm
Four-dimensional variational assimilation (4D-Var)

Minimise the cost function

\[
J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + \sum_{i=0}^{n} (y_i - H_i(x_i))^T R_i^{-1} (y_i - H_i(x_i))
\]

subject to model dynamics \( x_i = M_{0\rightarrow i} x_0 \)
Four-dimensional variational assimilation (4D-Var)

Minimise the cost function

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + \sum_{i=0}^{n} (y_i - H_i(x_i))^T R_i^{-1} (y_i - H_i(x_i)) \]

subject to model dynamics \( x_i = M_{0 \rightarrow i} x_0 \)

Figure: Copyright: ECMWF
Model dynamics

Strong constraint: model states $x_i$ are subject to

$$x_i = M_{0 \rightarrow i} x_0$$

nonlinear constraint optimisation problem (hard!)
4D-Var analysis

Model dynamics

Strong constraint: model states $x_i$ are subject to

$$x_i = M_{0 \rightarrow i} x_0$$

nonlinear constraint optimisation problem (hard!)

Simplifications

- **Causality** (forecast expressed as product of intermediate forecast steps)

  $$x_i = M_{i,i-1} M_{i-1,i-2} \cdots M_{1,0} x_0$$

- **Tangent linear hypothesis** ($H$ and $M$ can be linearised)

  $$y_i - H_i(x_i) = y_i - H_i(M_{0 \rightarrow i} x_0) = y_i - H_i(M_{0 \rightarrow i} x_0^B) - H_i M_{0 \rightarrow i} (x_0 - x_0^B)$$

  $M$ is the tangent linear model.
Model dynamics

Strong constraint: model states $x_i$ are subject to

$$x_i = M_{0\rightarrow i}x_0$$

nonlinear constraint optimisation problem (hard!)

Simplifications

- **Causality** (forecast expressed as product of intermediate forecast steps)
  $$x_i = M_{i,i-1}M_{i-1,i-2}\ldots M_{1,0}x_0$$

- **Tangent linear hypothesis** ($H$ and $M$ can be linearised)
  $$y_i - H_i(x_i) = y_i - H_i(M_{0\rightarrow i}x_0) = y_i - H_i(M_{0\rightarrow i}x_0^B) - H_iM_{0\rightarrow i}(x_0 - x_0^B)$$

  $M$ is the tangent linear model.

- **Unconstrained quadratic optimisation problem** (easier).
Minimisation of the 4D-Var cost function

Efficient implementation of $J$ and $\nabla J$:
- forecast state $x_i = M_{i,i-1}M_{i-1,i-2} \ldots M_{1,0}x_0$
Minimisation of the 4D-Var cost function

Efficient implementation of $J$ and $\nabla J$:

- forecast state $x_i = M_{i,i-1}M_{i-1,i-2} \ldots M_{1,0}x_0$
- normalised departures $d_i = R_i^{-1}(y_i - H_i(x_i))$
- cost function $J_{Oi} = (y_i - H_i(x_i))^T d_i$
Minimisation of the 4D-Var cost function

Efficient implementation of $J$ and $\nabla J$:

- forecast state $x_i = M_{i,i-1}M_{i-1,i-2} \ldots M_{1,0}x_0$
- normalised departures $d_i = R_i^{-1}(y_i - H_i(x_i))$
- cost function $J_{Oi} = (y_i - H_i(x_i))^T d_i$
- $\nabla J$ is calculated by

$$\frac{1}{2} \nabla J_O = -\frac{1}{2} \sum_{i=0}^{n} \nabla J_{Oi}$$

$$= \sum_{i=0}^{n} M_{i,i-1}^T H_i^T d_i$$

$$= H_0^T d_0 + M_{1,0}^T [H_1^T d_1 + M_{2,1}^T [H_2^T d_2 + \ldots + M_{n,n-1}^T H_n^T d_n] \ldots]$$
Minimisation of the 4D-Var cost function

Efficient implementation of $J$ and $\nabla J$:

- forecast state $x_i = M_{i,i-1}M_{i-1,i-2} \ldots M_{1,0}x_0$
- normalised departures $d_i = R^{-1}_i(y_i - H_i(x_i))$
- cost function $J_{Oi} = (y_i - H_i(x_i))^T d_i$
- $\nabla J$ is calculated by

$$-\frac{1}{2} \nabla J_O = -\frac{1}{2} \sum_{i=0}^{n} \nabla J_{Oi}$$

$$= \sum_{i=0}^{n} M_{1,0}^T \ldots M_{i,i-1}^T H_i^T d_i$$

$$= H_0^T d_0 + M_{1,0}^T [H_1^T d_1 + M_{2,1}^T [H_2^T d_2 + \ldots + M_{n,n-1}^T H_n^T d_n] \ldots]$$

- initialise adjoint variable $\tilde{x}_n = 0$ and then $\tilde{x}_{i-1} = M_{i,i-1}^T (\tilde{x}_i + H_i^T d_i)$
  etc., $\ldots \tilde{x}_0 = -\frac{1}{2} \nabla J_O$
Minimisation of the 4D-Var cost function

Efficient implementation of $J$ and $\nabla J$:

- forecast state $x_i = M_{i,i-1}M_{i-1,i-2} \ldots M_{1,0}x_0$
- normalised departures $d_i = R_i^{-1}(y_i - H_i(x_i))$
- cost function $J_{Oi} = (y_i - H_i(x_i))^T d_i$
- $\nabla J$ is calculated by

$$-\frac{1}{2}\nabla J_O = -\frac{1}{2} \sum_{i=0}^{n} \nabla J_{Oi}$$

$$= \sum_{i=0}^{n} M_{1,0}^T \ldots M_{i,i-1}^T H_i^T d_i$$

$$= H_0^T d_0 + M_{1,0}^T [H_1^T d_1 + M_{2,1}^T [H_2^T d_2 + \ldots + M_{n,n-1}^T H_n^T d_n] \ldots]$$

- initialise adjoint variable $\tilde{x}_n = 0$ and then $\tilde{x}_{i-1} = M_{i,i-1}^T (\tilde{x}_i + H_i^T d_i)$
  etc., $\ldots \tilde{x}_0 = -\frac{1}{2} \nabla J_O$

Further simplifications

- preconditioning with $B = LL^T$ (transform into control variable space) so that $\hat{x} = L^{-1}x$
- Incremental 4D-Var
Example - Three-Body Problem

Motion of three bodies in a plane, two position ($q$) and two momentum ($p$) coordinates for each body $\alpha = 1, 2, 3$
Example - Three-Body Problem

Motion of three bodies in a plane, two position ($q$) and two momentum ($p$) coordinates for each body $\alpha = 1, 2, 3$

Equations of motion

\[
H(q, p) = \frac{1}{2} \sum_\alpha \frac{|p_\alpha|^2}{m_\alpha} - \sum_\alpha \sum_{\alpha<\beta} \frac{m_\alpha m_\beta}{|q_\alpha - q_\beta|}
\]

\[
\frac{dq_\alpha}{dt} = \frac{\partial H}{\partial p_\alpha}
\]

\[
\frac{dp_\alpha}{dt} = -\frac{\partial H}{\partial q_\alpha}
\]
Example - Three-Body problem

- solver: partitioned Runge-Kutta scheme with time step $h = 0.001$
- observations are taken as noise from the truth trajectory
- background is given from a previous forecast
Example - Three-Body problem

- solver: partitioned Runge-Kutta scheme with time step \( h = 0.001 \)
- observations are taken as noise from the truth trajectory
- background is given from a previous forecast
- assimilation window is taken 300 time steps
- minimisation of cost function \( J \) using a Gauss-Newton method (neglecting all second derivatives)

\[
\nabla J(x_0) = 0
\]

\[
\nabla \nabla J(x^j_0) \Delta x^j_0 = -\nabla J(x^j_0), \quad x^{j+1}_0 = x^j_0 + \Delta x^j_0
\]

- subsequent forecast is take 3000 time steps
- \( R \) is diagonal with variances between \( 10^{-3} \) and \( 10^{-5} \)
Changing the masses of the bodies

DA needs Model error!

\[ m_s = 1.0 \rightarrow m_s = 1.1 \]
\[ m_p = 0.1 \rightarrow m_p = 0.11 \]
\[ m_m = 0.01 \rightarrow m_m = 0.011 \]
Changing the masses of the bodies

DA needs Model error!

\[ m_s = 1.0 \rightarrow m_s = 1.1 \]
\[ m_p = 0.1 \rightarrow m_p = 0.11 \]
\[ m_m = 0.01 \rightarrow m_m = 0.011 \]
Changing the masses of the bodies
Changing the masses of the bodies
Changing the masses of the bodies

![Graph showing RMS error before and after assimilation over time steps.](Image)
Changing the masses of the bodies
Changing the masses of the bodies

![Graph showing the RMS error before and after assimilation over time steps. The graph compares two lines: one dashed green line labeled "before assimilation" and one solid red line labeled "after assimilation." The x-axis represents time steps, and the y-axis represents RMS error.]
Root mean square error over whole assimilation window
Changing numerical method

- **Truth trajectory**: 4th order Runge-Kutta method with local truncation error $O(\Delta t^5)$
- **Model trajectory**: Explicit Euler method with local truncation error $O(\Delta t^2)$
Changing numerical method

![Graph showing RMS error over time step before assimilation](image)
Changing numerical method

![Graph showing RMS error vs time step before and after assimilation. The graph plots time step on the x-axis and RMS error on the y-axis. Two lines are shown: one dashed green line labeled 'before assimilation' and one solid red line labeled 'after assimilation'. The RMS error is lowest after assimilation compared to before assimilation.]
Changing numerical method

![Graph showing RMS error before and after assimilation over time steps]

- The green dashed line represents the RMS error before assimilation.
- The red solid line represents the RMS error after assimilation.

The graph illustrates the change in RMS error over time steps, with distinct peaks and troughs before and after the assimilation process.
Changing numerical method

![Graph showing RMS error over time step before and after assimilation. The graph displays two lines: one for before assimilation with a green dashed line, and one for after assimilation with a red line. The x-axis represents time step, ranging from 0 to 3500, and the y-axis represents RMS error, ranging from 0 to 0.25. The graph indicates a decrease in RMS error after assimilation.]
Root mean square error over whole assimilation window
Less observations - observations in sun only

- Sun position
- Planet position
- Moon position
- Sun momentum
- Planet momentum
- Moon momentum
Less observations - observations in planet only

![Graphs showing sun position, planet position, moon position, and planet momentum over time.](image)
Less observations - observations in moon only
Less observations - observations in sun and planet only
Less observations - observations in sun and moon only

![Graphs showing sun position, planet position, sun momentum, planet momentum, moon position, and moon momentum over time.](chart)
Less observations - observations in planet and moon only
Sequential data assimilation, background is provided by the forecast that starts from the previous analysis.

Covariance matrices $B^F$, $B^A$.

Forecast/model error $x_{i+1}^{\text{Truth}} = M_{i+1,i} x_i^{\text{Truth}} + \eta_i$ where $\eta_i \sim \mathcal{N}(0, Q_i)$, assumed to be uncorrelated to analysis error of previous forecast.
Sequential data assimilation, background is provided by the forecast that starts from the previous analysis

- covariance matrices \( B^F, B^A \)
- forecast/model error \( x^\text{Truth}_{i+1} = M_{i+1,i}x^\text{Truth}_i + \eta_i \) where \( \eta_i \sim \mathcal{N}(0, Q_i) \), assumed to be uncorrelated to analysis error of previous forecast

State and error covariance forecast

State forecast \( x^F_{i+1} = M_{i+1,i}x^A_i \)

Error covariance forecast \( B^F_{i+1} = M_{i+1,i}B^A_iM^T_{i+1,i} + Q_i \)
The Kalman Filter Algorithm

- **Sequential data assimilation**, background is provided by the forecast that starts from the previous analysis
- Covariance matrices $B^F, B^A$
- Forecast/model error $x_{i+1}^{\text{Truth}} = M_{i+1,i} x_i^{\text{Truth}} + \eta_i$ where $\eta_i \sim \mathcal{N}(0, Q_i)$, assumed to be uncorrelated to analysis error of previous forecast

### State and error covariance forecast

<table>
<thead>
<tr>
<th>State forecast</th>
<th>$x_{i+1}^F = M_{i+1,i} x_i^A$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error covariance forecast</td>
<td>$B_{i+1}^F = M_{i+1,i} B_i^A M_{i+1,i}^T + Q_i$</td>
</tr>
</tbody>
</table>

### State and error covariance analysis

<table>
<thead>
<tr>
<th>Kalman gain</th>
<th>$K_i = B_i^F H_i^T (H_i B_i^F H_i^T + R_i)^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>State analysis</td>
<td>$x_i^A = x_i^F + K_i (y_i - H_i x_i^F)$</td>
</tr>
<tr>
<td>Error covariance of analysis</td>
<td>$B_i^A = (I - K_i H_i) B_i^F$</td>
</tr>
</tbody>
</table>
### Extended Kalman Filter

Extension of the Kalman Filter Algorithm to nonlinear observation operators $H$ and nonlinear model dynamics $M$, where both $H$ and $M$ are linearised.
The Kalman Filter Algorithm

Extended Kalman Filter

Extension of the Kalman Filter Algorithm to nonlinear observation operators $H$ and nonlinear model dynamics $M$, where both $H$ and $M$ are linearised.

Equivalence 4D-Var Kalman Filter

Assume

- $Q_i = 0$, $\forall i$ (no model error)
- both 4D-Var and the Kalman filter use the same initial input data
- $H$ and $M$ are linear,

then 4D-Var and the Kalman Filter produce the same state estimate $x^A$ at the end of the assimilation window.
RMS error over whole assimilation window - using 4D-Var
RMS error over whole assimilation window - using Kalman Filter
Example - Three-Body Problem

- solver: partitioned Runge-Kutta scheme with time step $h = 0.001$
- observations are taken as noise from the truth trajectory
- background is given from a perturbed initial condition
- assimilation window is taken 300 time steps
- minimisation of cost function $J$ using a Gauss-Newton method (neglecting all second derivatives)
- application of 4D-Var
Example - Three-Body Problem

- solver: partitioned Runge-Kutta scheme with time step $h = 0.001$
- observations are taken as noise from the truth trajectory
- background is given from a perturbed initial condition
- assimilation window is taken 300 time steps
- minimisation of cost function $J$ using a Gauss-Newton method (neglecting all second derivatives)
- application of 4D-Var
- Compare using $B = I$ with using a flow-dependent matrix $B$ which was generated by a Kalman Filter before the assimilation starts (see G. Inverarity (2007))
Example - Three-Body Problem

Figure: 4D-Var with $B = I$
Example - Three-Body Problem

Figure: 4D-Var with $B = I$

Figure: 4D-Var with $B = P^A$
Problems with Data Assimilation

- DA is computational very expensive, one cycle is much more expensive than the actual forecast
Problems with Data Assimilation

- DA is computational very expensive, one cycle is much more expensive than the actual forecast
- estimation and storage of the $\mathbf{B}$-matrix is hard
  - in operational DA $\mathbf{B}$ is about $10^7 \times 10^7$
  - $\mathbf{B}$ should be flow-dependent but in practice often static
  - $\mathbf{B}$ needs to be modeled and diagonalised since $\mathbf{B}^{-1}$ too expensive to compute (”control variable transform”)

Problems with Data Assimilation

- DA is computational very expensive, one cycle is much more expensive than the actual forecast
- estimation and storage of the $\mathbf{B}$-matrix is hard
  - in operational DA $\mathbf{B}$ is about $10^7 \times 10^7$
  - $\mathbf{B}$ should be flow-dependent but in practice often static
  - $\mathbf{B}$ needs to be modeled and diagonalised since $\mathbf{B}^{-1}$ too expensive to compute ("control variable transform")
- many assumptions are not valid
  - errors non-Gaussian, data have biases
  - forward model operator $M$ is not exact and also non-linear and system dynamics are chaotic
  - minimisation of the cost function needs close initial guess, small assimilation window
Problems with Data Assimilation

- DA is computational very expensive, one cycle is much more expensive than the actual forecast
- estimation and storage of the $B$-matrix is hard
  - in operational DA $B$ is about $10^7 \times 10^7$
  - $B$ should be flow-dependent but in practice often static
  - $B$ needs to be modeled and diagonalised since $B^{-1}$ too expensive to compute (”control variable transform”)
- many assumptions are not valid
  - errors non-Gaussian, data have biases
  - forward model operator $M$ is not exact and also non-linear and system dynamics are chaotic
  - minimisation of the cost function needs close initial guess, small assimilation window
- model error not included
Outline

1. Introduction

2. Variational Data Assimilation
   - Least square estimation
   - Examples
   - Kalman Filter
   - Problems and Issues

3. Tikhonov regularisation

4. Plan and work in progress
Relation between 4D-Var and Tikhonov regularisation

4D-Var minimises

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + \sum_{i=0}^{n} (y_i - H_i(x_i))^T R_i^{-1} (y_i - H_i(x_i)) \]

subject to model dynamics \( x_i = M_{0 \rightarrow i} x_0 \)
Relation between 4D-Var and Tikhonov regularisation

4D-Var minimises

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + \sum_{i=0}^{n} (y_i - H_i(x_i))^T R_i^{-1} (y_i - H_i(x_i)) \]

subject to model dynamics \( x_i = M_{0 \rightarrow i} x_0 \)

or

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]

where

\[ \hat{H} = [H_0^T, (H_1 M(t_1, t_0))^T, \ldots (H_n M(t_n, t_0))^T]^T \]

\[ \hat{y} = [y_0^T, \ldots, y_n^T] \]

and \( \hat{R} \) is block diagonal with \( R_i \) on diagonal.
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]

is given by

\[ x_0 = x_0^B + (B^{-1} + \hat{H}^T \hat{R}^{-1} \hat{H})^{-1} \hat{H}^T \hat{R}^{-1} \hat{d}, \quad \hat{d} = \hat{H}(x_0^B - \hat{y}) \]
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]

is given by

\[ x_0 = x_0^B + \left( B^{-1} + \hat{H}^T \hat{R}^{-1} \hat{H} \right)^{-1} \hat{H}^T \hat{R}^{-1} \hat{d}, \quad \hat{d} = \hat{H}(x_0^B - \hat{y}) \]

Singular value decomposition

Assume \( B = \sigma_B^2 I \) and \( \hat{R} = \sigma_O^2 I \) and define the SVD of the observability matrix \( \hat{H} \)

\[ \hat{H} = U \Lambda V^T \]

Then the optimal analysis can be written as

\[ x_0 = x_0^B + \sum_j \frac{\lambda_j^2}{\mu^2 + \lambda_j^2} \frac{u_j^T \hat{d}}{\lambda_j} v_j \]

where \( \mu^2 = \frac{\sigma_O^2}{\sigma_B^2} \).
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]

Variable transformations

\[ B = \sigma_B^2 F_B \text{ and } \hat{R} = \sigma_O^2 F_R \text{ and} \]
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]

Variable transformations

\[ B = \sigma_B^2 F_B \text{ and } \hat{R} = \sigma_O^2 F_R \text{ and define new variable } z := F_B^{-1/2} (x_0 - x_0^B) \]
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1} (x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1} (\hat{y} - \hat{H}(x_0)) \]

Variable transformations

\[ B = \sigma_B^2 F_B \quad \text{and} \quad \hat{R} = \sigma_O^2 F_R \]

and define new variable \( z := F_B^{-1/2} (x_0 - x_0^B) \)

\[ \hat{J}(z) = \mu^2 ||z||_2^2 + ||F_R^{-1/2} \hat{d} - F_R^{-1/2} \hat{H} F_B^{-1/2} z||_2^2 \]

\( \mu^2 \) can be interpreted as a regularisation parameter.
Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

\[ J(x_0) = (x_0 - x_0^B)^T B^{-1}(x_0 - x_0^B) + (\hat{y} - \hat{H}(x_0))^T \hat{R}^{-1}(\hat{y} - \hat{H}(x_0)) \]

Variable transformations

\[ B = \sigma_B^2 F_B \] and \[ \hat{R} = \sigma_O^2 F_R \] and define new variable \( z := F_B^{-1/2}(x_0 - x_0^B) \)

\[ \hat{J}(z) = \mu^2 \|z\|_2^2 + \|F_R^{-1/2} d - F_R^{-1/2} \hat{H} F_B^{-1/2} z\|_2^2 \]

\( \mu^2 \) can be interpreted as a regularisation parameter.
This is the well-known Tikhonov regularisation!
Outline

1 Introduction

2 Variational Data Assimilation
   - Least square estimation
   - Examples
   - Kalman Filter
   - Problems and Issues

3 Tikhonov regularisation

4 Plan and work in progress
Met Office research and plans

- include several time scales (to model the atmosphere)
Met Office research and plans

- include several time scales (to model the atmosphere)
- improve the representation of **multiscale behaviour** in the atmosphere in existing DA methods
- improve the forecast of small scale features (like convective storms)
Met Office research and plans

- include several time scales (to model the atmosphere)
- improve the representation of **multiscale behaviour** in the atmosphere in existing DA methods
- improve the forecast of small scale features (like convective storms)
- compare assimilation algorithms and optimisation strategies to reduce existing errors
- use **regularisation methods from image processing**, for example $L_1$ regularisation to improve forecasts
Met Office research and plans

- include several time scales (to model the atmosphere)
- improve the representation of **multiscale behaviour** in the atmosphere in existing DA methods
- improve the forecast of small scale features (like convective storms)
- compare assimilation algorithms and optimisation strategies to reduce existing errors
- use regularisation methods from image processing, for example $L_1$ regularisation to improve forecasts
- identify and **analyse model error** and analyse influence of this model error onto the DA scheme