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

Outline

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 $\mathbf{x}_i$.

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)

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)

$$\mathbf{x}_{i+1} = M(\mathbf{x}_i)$$

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

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)

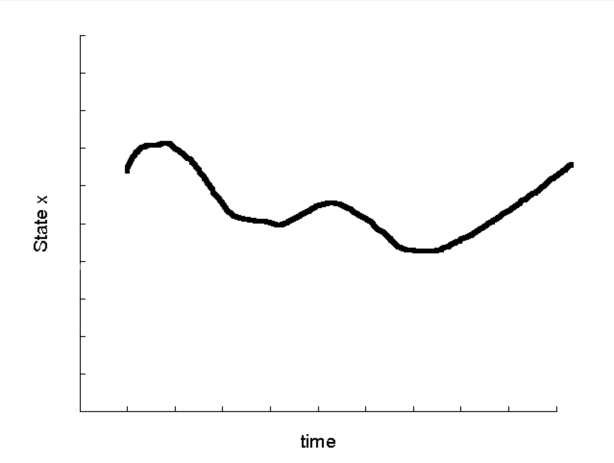


Figure: Background state $\mathbf{x}^B$

Schematics of DA

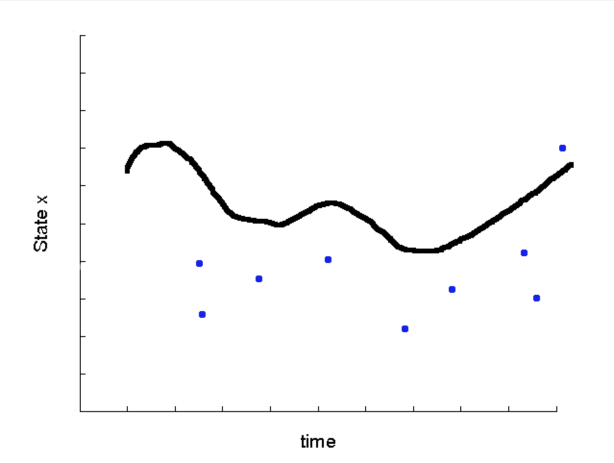


Figure: Observations $\mathbf{y}$

Schematics of DA

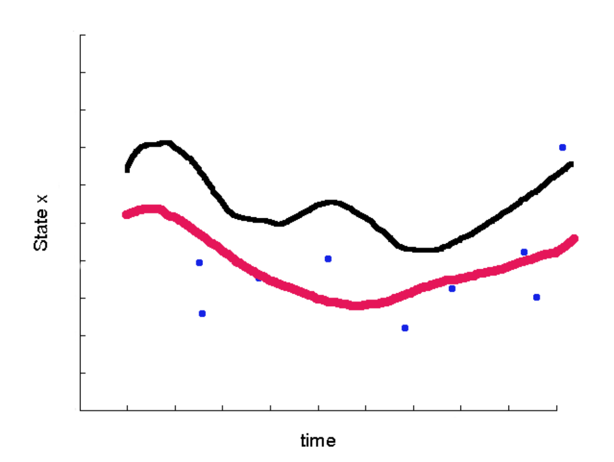


Figure: Analysis $\mathbf{x}^A$ (consistent with observations and model dynamics)

Underdeterminacy

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

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 $\mathbf{y}$): $\mathcal{O}(10^5 - 10^6)$

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: $\mathbf{y} = H(\mathbf{x})$

Any easy scheme

Cressman analysis

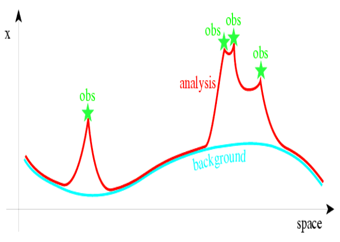


Figure: Copyright:ECMWF

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

UNIVERSITY OF BATH

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)

Error variables

Modelling the errors

- background error $\varepsilon^B = \mathbf{x}^B - \mathbf{x}^{\mathrm{Truth}}$ of average $\overline{\varepsilon}^B$ and covariance

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

UNIVERSITY OF BATH

Error variables

Modelling the errors

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

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

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

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

Error variables

Modelling the errors

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

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

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

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

- analysis error $\varepsilon^A = \mathbf{x}^A - \mathbf{x}^{\text{Truth}}$ of average $\overline{\varepsilon}^A$ and covariance

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

- measure of the analysis error that we want to minimise

$$\text{tr}(\mathbf{A}) = \overline{\|\varepsilon^A - \overline{\varepsilon}^A\|^2}$$

Assumptions

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

Optimal least-squares estimater

Cost function

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

$$\begin{aligned} J(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}^B)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^B) + (\mathbf{y}-H(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-H(\mathbf{x})) \\ &= J_B(\mathbf{x}) + J_O(\mathbf{x}) \end{aligned}$$

UNIVERSITY OF BATH

Optimal least-squares estimater

Cost function

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

$$\begin{aligned} J(\mathbf{x}) &= (\mathbf{x} - \mathbf{x}^B)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^B) + (\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x})) \\ &= J_B(\mathbf{x}) + J_O(\mathbf{x}) \end{aligned}$$

Interpolation equations

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

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \quad \mathbf{K} \ldots \text{gain matrix}$$

Conditional probabilities

Non-Gaussian PDF's (probability density function)

- $P(\mathbf{x})$ is a priori PDF (background)
- $P(\mathbf{y}|\mathbf{x})$ is the observation PDF (likelihood of the observations given background $\mathbf{x}$)

Conditional probabilities

Non-Gaussian PDF's (probability density function)

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

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

Conditional probabilities

Non-Gaussian PDF's (probability density function)

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

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

Gaussian PDF's

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

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

A simple scalar illustration

Room temperature

- T^O observation with standard deviation σ_O
- T^B background with standard deviation σ_B

A simple scalar illustration

Room temperature

- T^O observation with standard deviation σ_O
- T^B background with standard deviation σ_B
- $T^A = T^B + k(T^O - T^B)$ with error variance $\sigma_A^2 = (1-k)^2\sigma_B^2 + k^2\sigma_O^2$
- optimal k which minimises error variance

$$k = \frac{\sigma_B^2}{\sigma_B^2 + \sigma_O^2}$$

- equivalent to minimising

$$J(T) = \frac{(T - T^B)^2}{\sigma_B^2} + \frac{(T - T^O)^2}{\sigma_O^2}$$

and then $\dfrac{1}{\sigma_A^2} = \dfrac{1}{\sigma_B^2} + \dfrac{1}{\sigma_O^2}$

Optimal interpolation

Optimal interpolation

Computation of

$$\mathbf{x}^A = \mathbf{x}^B + \mathbf{K}(\mathbf{y} - H(\mathbf{x}^B))$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \quad \mathbf{K}\ldots \text{gain matrix}$$

Optimal interpolation

Optimal interpolation

Computation of

$$\mathbf{x}^A = \mathbf{x}^B + \mathbf{K}(\mathbf{y} - H(\mathbf{x}^B))$$

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} \quad \mathbf{K}\ldots \text{gain matrix}$$

- expensive!

Three-dimensional variational assimilation (3D-Var)

3D-Var

Minimisation of

$$
\begin{aligned}
J(\mathbf{x}) &= (\mathbf{x}-\mathbf{x}^B)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^B) + (\mathbf{y}-H(\mathbf{x}))^T\mathbf{R}^{-1}(\mathbf{y}-H(\mathbf{x})) \\
&= J_B(\mathbf{x}) + J_O(\mathbf{x})
\end{aligned}
$$

Three-dimensional variational assimilation (3D-Var)

3D-Var

Minimisation of

$$\begin{aligned} J(\mathbf{x}) &= (\mathbf{x} - \mathbf{x}^B)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}^B) + (\mathbf{y} - H(\mathbf{x}))^T \mathbf{R}^{-1} (\mathbf{y} - H(\mathbf{x})) \\ &= J_B(\mathbf{x}) + J_O(\mathbf{x}) \end{aligned}$$

- avoids computation of $\mathbf{K}$ by using a descent algorithm s

Four-dimensional variational assimilation (4D-Var)

Minimise the cost function

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + \sum_{i=0}^{n} (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1} (\mathbf{y}_i - H_i(\mathbf{x}_i))$$

subject to model dynamics $\mathbf{x}_i = M_{0 \to i} \mathbf{x}_0$

Four-dimensional variational assimilation (4D-Var)

Minimise the cost function

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + \sum_{i=0}^{n} (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1} (\mathbf{y}_i - H_i(\mathbf{x}_i))$$

subject to model dynamics $\mathbf{x}_i = M_{0\to i}\mathbf{x}_0$

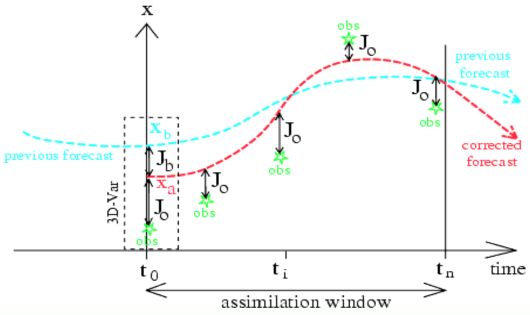


Figure: Copyright:ECMWF

4D-Var analysis

Model dynamics

Strong constraint: model states $\mathbf{x}_i$ are subject to

$$\mathbf{x}_i = M_{0\to i}\mathbf{x}_0$$

nonlinear constraint optimisation problem (hard!)

4D-Var analysis

Model dynamics

Strong constraint: model states $\mathbf{x}_i$ are subject to

$$\mathbf{x}_i = M_{0\to i}\mathbf{x}_0$$

nonlinear constraint optimisation problem (hard!)

Simplifications

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

$$\mathbf{x}_i = M_{i,i-1}M_{i-1,i-2}\dots M_{1,0}\mathbf{x}_0$$

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

$$\mathbf{y}_i - H_i(\mathbf{x}_i) = \mathbf{y}_i - H_i(M_{0\to i}\mathbf{x}_0) = \mathbf{y}_i - H_i(M_{0\to i}\mathbf{x}_0^B) - \mathbf{H}_i\mathbf{M}_{0\to i}(\mathbf{x}_0 - \mathbf{x}_0^B)$$

$\mathbf{M}$ is the tangent linear model.

4D-Var analysis

Model dynamics

Strong constraint: model states $\mathbf{x}_i$ are subject to

$$\mathbf{x}_i = M_{0\to i}\mathbf{x}_0$$

nonlinear constraint optimisation problem (hard!)

Simplifications

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

$$\mathbf{x}_i = M_{i,i-1}M_{i-1,i-2}\dots M_{1,0}\mathbf{x}_0$$

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

$$\mathbf{y}_i - H_i(\mathbf{x}_i) = \mathbf{y}_i - H_i(M_{0\to i}\mathbf{x}_0) = \mathbf{y}_i - H_i(M_{0\to i}\mathbf{x}_0^B) - \mathbf{H}_i\mathbf{M}_{0\to i}(\mathbf{x}_0 - \mathbf{x}_0^B)$$

  $\mathbf{M}$ is the tangent linear model.

- unconstrained quadratic optimisation problem (easier).

Minimisation of the 4D-Var cost function

Efficient implementation of J and ∇J:

- forecast state $\mathbf{x}_i = M_{i,i-1} M_{i-1,i-2} \dots M_{1,0} \mathbf{x}_0$

Minimisation of the 4D-Var cost function

Efficient implementation of J and ∇J:

- forecast state $\mathbf{x}_i = M_{i,i-1} M_{i-1,i-2} \ldots M_{1,0} \mathbf{x}_0$
- normalised departures $\mathbf{d}_i = \mathbf{R}_i^{-1}(\mathbf{y}_i - H_i(\mathbf{x}_i))$
- cost function $J_{Oi} = (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{d}_i$

Minimisation of the 4D-Var cost function

Efficient implementation of J and ∇J:

- forecast state $\mathbf{x}_i = M_{i,i-1} M_{i-1,i-2} \ldots M_{1,0} \mathbf{x}_0$
- normalised departures $\mathbf{d}_i = \mathbf{R}_i^{-1}(\mathbf{y}_i - H_i(\mathbf{x}_i))$
- cost function $J_{Oi} = (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{d}_i$
- ∇J is calculated by

$$
\begin{aligned}
-\frac{1}{2}\nabla J_O &= -\frac{1}{2}\sum_{i=0}^{n} \nabla J_{Oi} \\
&= \sum_{i=0}^{n} \mathbf{M}_{1,0}^T \ldots \mathbf{M}_{i,i-1}^T \mathbf{H}_i^T \mathbf{d}_i \\
&= \mathbf{H}_0^T \mathbf{d}_0 + \mathbf{M}_{1,0}^T [\mathbf{H}_1^T \mathbf{d}_1 + \mathbf{M}_{2,1} [\mathbf{H}_2^T \mathbf{d}_2 + \ldots + \mathbf{M}_{n,n-1}^T \mathbf{H}_n^T \mathbf{d}_n] \ldots]
\end{aligned}
$$

UNIVERSITY OF BATH

Minimisation of the 4D-Var cost function

Efficient implementation of J and ∇J:

- forecast state $\mathbf{x}_i = M_{i,i-1} M_{i-1,i-2} \ldots M_{1,0} \mathbf{x}_0$
- normalised departures $\mathbf{d}_i = \mathbf{R}_i^{-1}(\mathbf{y}_i - H_i(\mathbf{x}_i))$
- cost function $J_{Oi} = (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{d}_i$
- ∇J is calculated by

$$\begin{aligned}
-\frac{1}{2}\nabla J_O &= -\frac{1}{2}\sum_{i=0}^{n} \nabla J_{Oi} \\
&= \sum_{i=0}^{n} \mathbf{M}_{1,0}^T \ldots \mathbf{M}_{i,i-1}^T \mathbf{H}_i^T \mathbf{d}_i \\
&= \mathbf{H}_0^T \mathbf{d}_0 + \mathbf{M}_{1,0}^T [\mathbf{H}_1^T \mathbf{d}_1 + \mathbf{M}_{2,1} [\mathbf{H}_2^T \mathbf{d}_2 + \ldots + \mathbf{M}_{n,n-1}^T \mathbf{H}_n^T \mathbf{d}_n] \ldots]
\end{aligned}$$

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

Minimisation of the 4D-Var cost function

Efficient implementation of J and ∇J:

- forecast state $\mathbf{x}_i = M_{i,i-1} M_{i-1,i-2} \dots M_{1,0} \mathbf{x}_0$
- normalised departures $\mathbf{d}_i = \mathbf{R}_i^{-1}(\mathbf{y}_i - H_i(\mathbf{x}_i))$
- cost function $J_{Oi} = (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{d}_i$
- ∇J is calculated by

$$
\begin{aligned}
-\frac{1}{2}\nabla J_O &= -\frac{1}{2}\sum_{i=0}^{n} \nabla J_{Oi} \\
&= \sum_{i=0}^{n} \mathbf{M}_{1,0}^T \dots \mathbf{M}_{i,i-1}^T \mathbf{H}_i^T \mathbf{d}_i \\
&= \mathbf{H}_0^T \mathbf{d}_0 + \mathbf{M}_{1,0}^T [\mathbf{H}_1^T \mathbf{d}_1 + \mathbf{M}_{2,1}[\mathbf{H}_2^T \mathbf{d}_2 + \dots + \mathbf{M}_{n,n-1}^T \mathbf{H}_n^T \mathbf{d}_n] \dots]
\end{aligned}
$$

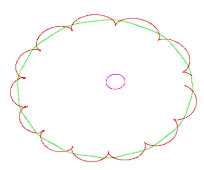
- initialise adjoint variable $\tilde{\mathbf{x}}_n = \mathbf{0}$ and then $\tilde{\mathbf{x}}_{i-1} = \mathbf{M}_{i,i-1}^T(\tilde{\mathbf{x}}_i + \mathbf{H}_i^T \mathbf{d}_i)$ etc., $\dots \tilde{\mathbf{x}}_0 = -\frac{1}{2}\nabla J_O$

Further simplifications

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

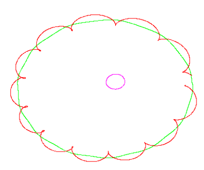
Example - Three-Body Problem

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



Example - Three-Body Problem

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



Equations of motion

$$
\begin{aligned}
H(\mathbf{q}, \mathbf{p}) &= \frac{1}{2} \sum_{\alpha} \frac{|\mathbf{p}_\alpha|^2}{m_\alpha} - \sum \sum_{\alpha<\beta} \frac{m_\alpha m_\beta}{|\mathbf{q}_\alpha - \mathbf{q}_\beta|} \\
\frac{d\mathbf{q}_\alpha}{dt} &= \frac{\partial H}{\partial \mathbf{p}_\alpha} \\
\frac{d\mathbf{p}_\alpha}{dt} &= -\frac{\partial H}{\partial \mathbf{q}_\alpha}
\end{aligned}
$$

UNIVERSITY OF BATH

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)

$$\nabla J(\mathbf{x}_0) = 0$$

$$\nabla\nabla J(\mathbf{x}_0^j)\Delta\mathbf{x}_0^j = -\nabla J(\mathbf{x}_0^j), \quad \mathbf{x}_0^{j+1} = \mathbf{x}_0^j + \Delta\mathbf{x}_0^j$$

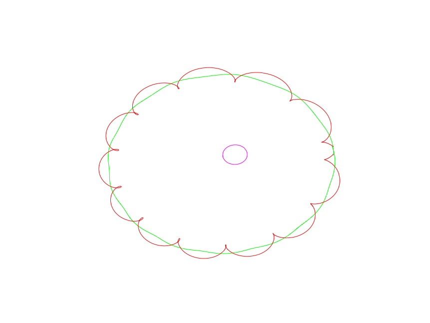
- subsequent forecast is take 3000 time steps
- R is diagonal with variances between 10^{-3} and 10^{-5}

UNIVERSITY OF BATH

Changing the masses of the bodies

DA needs Model error!

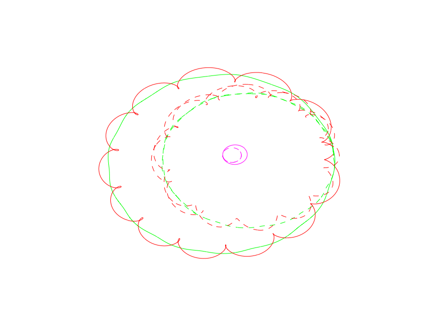
$$\begin{aligned} m_s &= 1.0 \quad \longrightarrow \quad m_s = 1.1 \\ m_p &= 0.1 \quad \longrightarrow \quad m_p = 0.11 \\ m_m &= 0.01 \quad \longrightarrow \quad m_m = 0.011 \end{aligned}$$



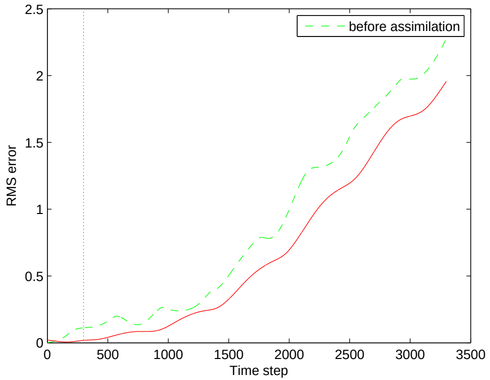
Changing the masses of the bodies

DA needs Model error!

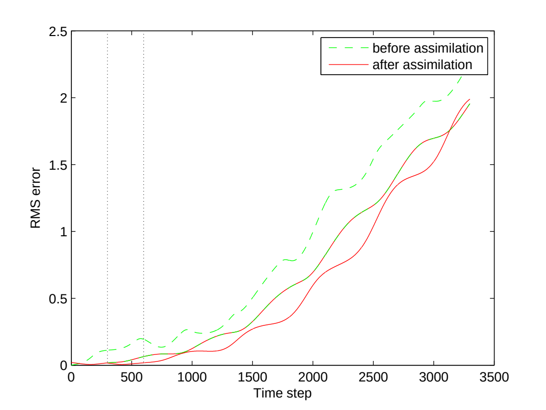
$$\begin{aligned} m_s = 1.0 &\quad\rightarrow\quad m_s = 1.1 \\ m_p = 0.1 &\quad\rightarrow\quad m_p = 0.11 \\ m_m = 0.01 &\quad\rightarrow\quad m_m = 0.011 \end{aligned}$$



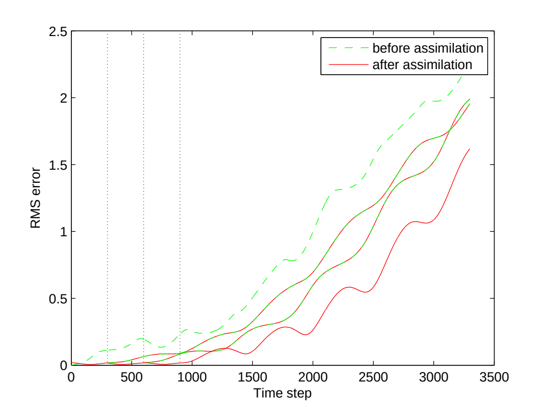
Changing the masses of the bodies



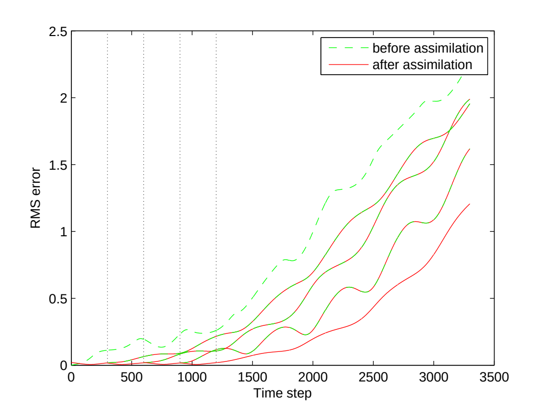
Changing the masses of the bodies



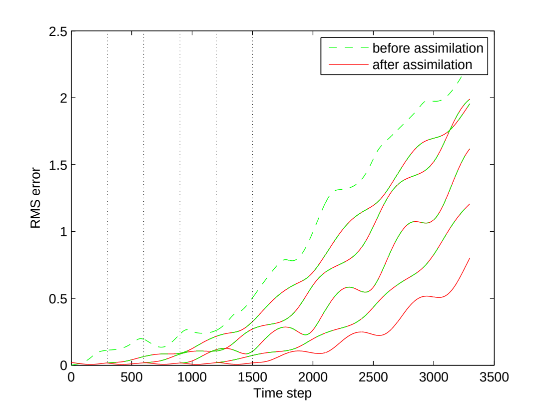
Changing the masses of the bodies



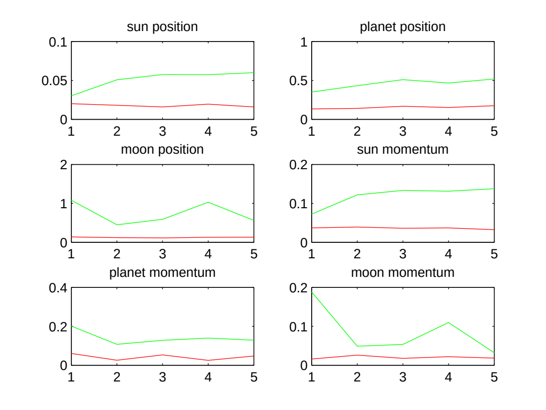
Changing the masses of the bodies



Changing the masses of the bodies

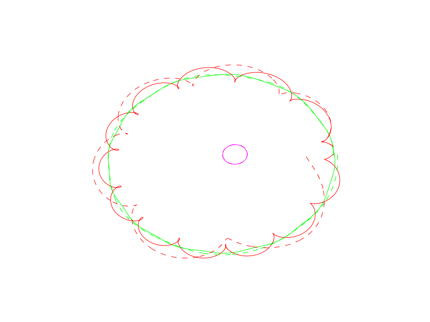


Root mean square error over whole assimilation window

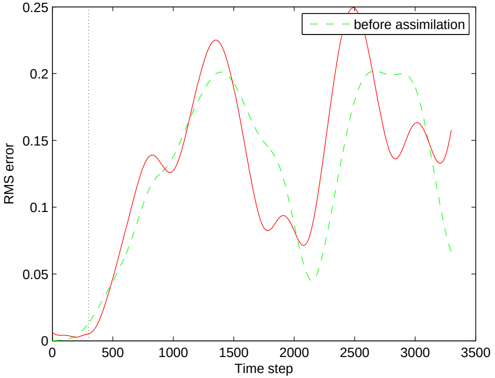


Changing numerical method

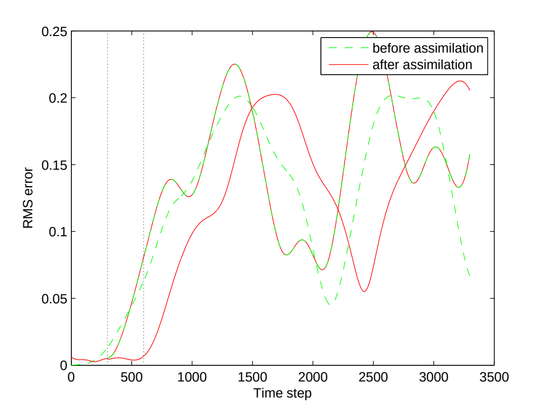
- Truth trajectory: 4th order Runge-Kutta method with local truncation error $\mathcal{O}(\Delta t^5)$
- Model trajectory: Explicit Euler method with local truncation error $\mathcal{O}(\Delta t^2)$



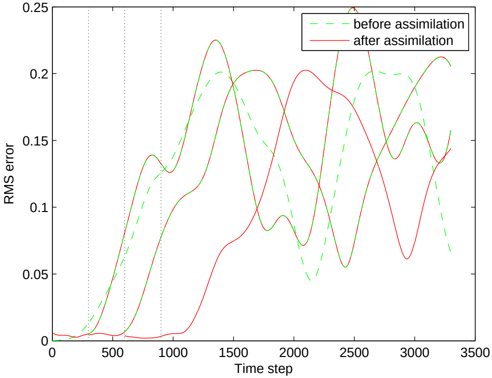
Changing numerical method



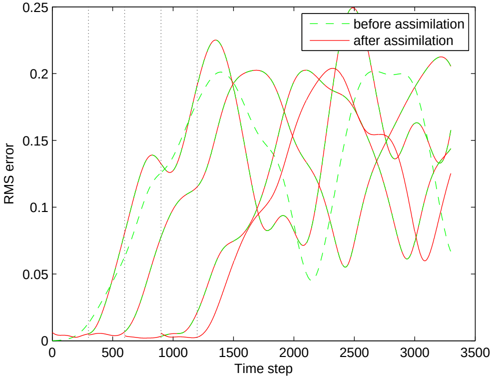
Changing numerical method



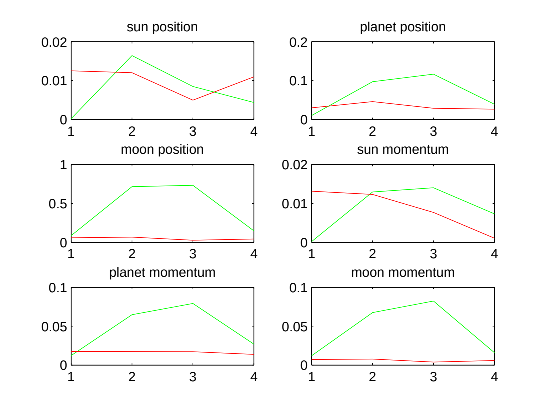
Changing numerical method



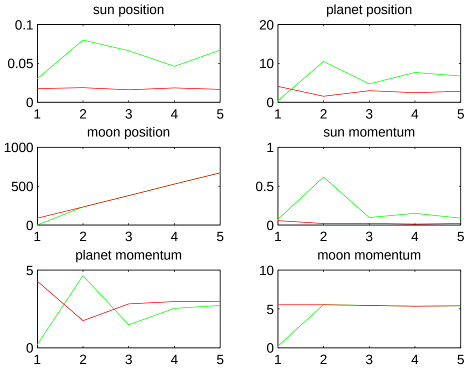
Changing numerical method



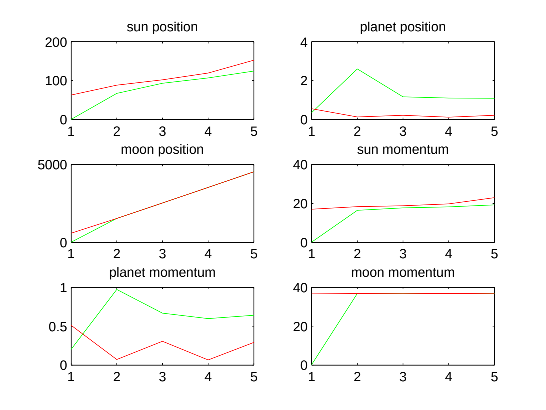
Root mean square error over whole assimilation window



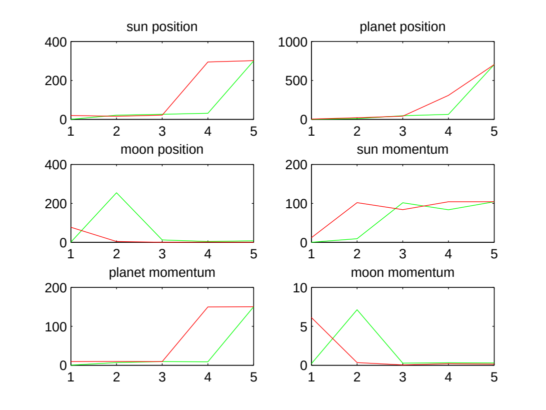
Less observations - observations in sun only



Less observations - observations in planet only

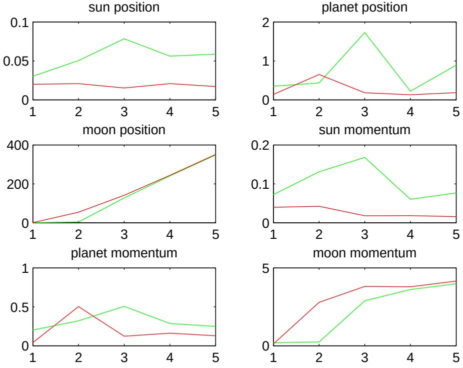


Less observations - observations in moon only

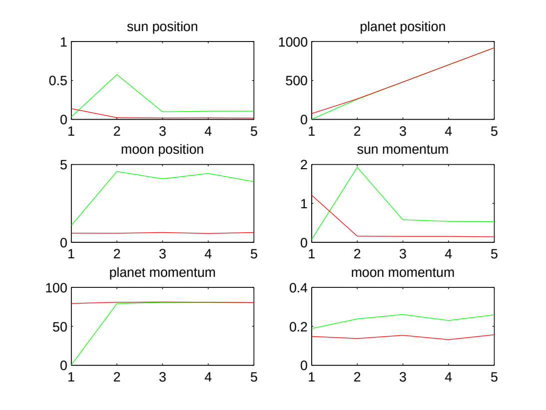


UNIVERSITY OF BATH

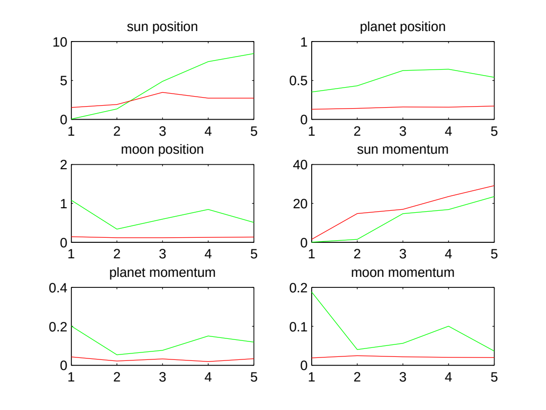
Less observations - observations in sun and planet only



Less observations - observations in sun and moon only



Less observations - observations in planet and moon only



The Kalman Filter Algorithm

- Sequential data assimilation, background is provided by the forecast that starts from the previous analysis
- covariance matrices $\mathbf{B}^F$, $\mathbf{B}^A$
- forecast/model error $\mathbf{x}_{i+1}^{\text{Truth}} = \mathbf{M}_{i+1,i}\mathbf{x}_i^{\text{Truth}} + \eta_i$ where $\eta_i \sim \mathcal{N}(0, \mathbf{Q}_i)$, assumed to be uncorrelated to analysis error of previous forecast

The Kalman Filter Algorithm

- Sequential data assimilation, background is provided by the forecast that starts from the previous analysis
- covariance matrices $\mathbf{B}^F$, $\mathbf{B}^A$
- forecast/model error $\mathbf{x}_{i+1}^{\text{Truth}} = \mathbf{M}_{i+1,i}\mathbf{x}_i^{\text{Truth}} + \eta_i$ where $\eta_i \sim \mathcal{N}(0, \mathbf{Q}_i)$, assumed to be uncorrelated to analysis error of previous forecast

State and error covariance forecast

$$
\begin{aligned}
\text{State forecast} \quad \mathbf{x}_{i+1}^F &= \mathbf{M}_{i+1,i}\mathbf{x}_i^A \\
\text{Error covariance forecast} \quad \mathbf{B}_{i+1}^F &= \mathbf{M}_{i+1,i}\mathbf{B}_i^A\mathbf{M}_{i+1,i}^T + \mathbf{Q}_i
\end{aligned}
$$

The Kalman Filter Algorithm

- Sequential data assimilation, background is provided by the forecast that starts from the previous analysis
- covariance matrices $\mathbf{B}^F$, $\mathbf{B}^A$
- forecast/model error $\mathbf{x}_{i+1}^{\text{Truth}} = \mathbf{M}_{i+1,i}\mathbf{x}_i^{\text{Truth}} + \eta_i$ where $\eta_i \sim \mathcal{N}(0, \mathbf{Q}_i)$, assumed to be uncorrelated to analysis error of previous forecast

State and error covariance forecast

$$\begin{aligned} \text{State forecast} \quad \mathbf{x}_{i+1}^F &= \mathbf{M}_{i+1,i}\mathbf{x}_i^A \\ \text{Error covariance forecast} \quad \mathbf{B}_{i+1}^F &= \mathbf{M}_{i+1,i}\mathbf{B}_i^A\mathbf{M}_{i+1,i}^T + \mathbf{Q}_i \end{aligned}$$

State and error covariance analysis

$$\begin{aligned} \text{Kalman gain} \quad \mathbf{K}_i &= \mathbf{B}_i^F\mathbf{H}_i^T(\mathbf{H}_i\mathbf{B}_i^F\mathbf{H}_i^T + \mathbf{R}_i)^{-1} \\ \text{State analysis} \quad \mathbf{x}_i^A &= \mathbf{x}_i^F + \mathbf{K}_i(\mathbf{y}_i - \mathbf{H}_i\mathbf{x}_i^F) \\ \text{Error covariance of analysis} \quad \mathbf{B}_i^A &= (\mathbf{I} - \mathbf{K}_i\mathbf{H}_i)\mathbf{B}_i^F \end{aligned}$$

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.

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.

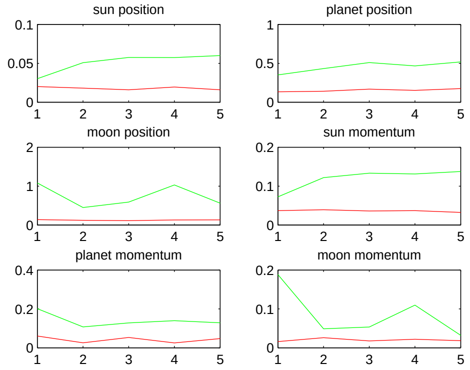
Equivalence 4D-Var Kalman Filter

Assume

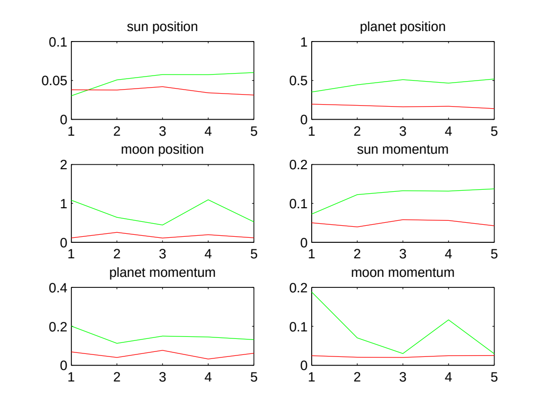
- $\mathbf{Q}_i = 0$, $\forall i$ (no model error)
- both 4D-Var and the Kalman filter use the same initial input data
- H and M are linear,

then 4D-Var and the Kalman Filter produce the same state estimate $\mathbf{x}^A$ 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 $\mathbf{B} = \mathbf{I}$ with using a flow-dependent matrix $\mathbf{B}$ which was generated by a Kalman Filter before the assimilation starts (see G. Inverarity (2007))

Example - Three-Body Problem

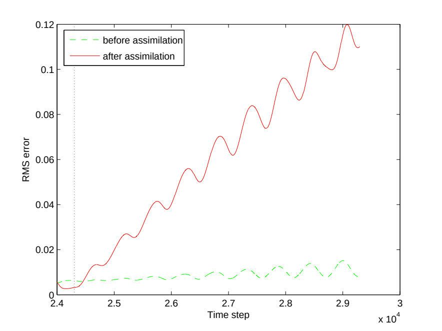


Figure: 4D-Var with $\mathbf{B} = \mathbf{I}$

Example - Three-Body Problem

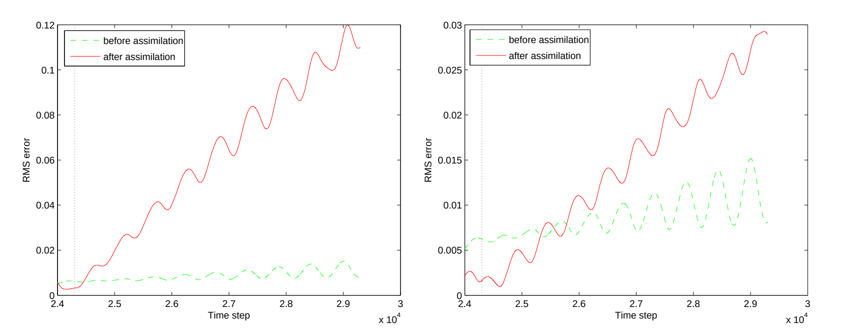


Figure: 4D-Var with $\mathbf{B} = \mathbf{I}$

Figure: 4D-Var with $\mathbf{B} = \mathbf{P}^A$

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 **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 $\mathbf{B}^{-1}$ 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 $\mathbf{B}^{-1}$ 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
- model error not included

Outline

UNIVERSITY OF BATH

Relation between 4D-Var and Tikhonov regularisation

4D-Var minimises

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + \sum_{i=0}^{n} (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1} (\mathbf{y}_i - H_i(\mathbf{x}_i))$$

subject to model dynamics $\mathbf{x}_i = M_{0\to i}\mathbf{x}_0$

Relation between 4D-Var and Tikhonov regularisation

4D-Var minimises

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + \sum_{i=0}^{n} (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1} (\mathbf{y}_i - H_i(\mathbf{x}_i))$$

subject to model dynamics $\mathbf{x}_i = M_{0\to i}\mathbf{x}_0$

or

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

where

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

$$\hat{\mathbf{y}} = [\mathbf{y}_0^T, \ldots, \mathbf{y}_n^T]$$

and $\hat{\mathbf{R}}$ is block diagonal with $\mathbf{R}_i$ on diagonal.

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

is given by

$$\mathbf{x}_0 = \mathbf{x}_0^B + (\mathbf{B}^{-1} + \hat{\mathbf{H}}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{H}})^{-1} \hat{\mathbf{H}}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{d}}, \quad \hat{\mathbf{d}} = \hat{\mathbf{H}}(\mathbf{x}_0^B - \hat{\mathbf{y}})$$

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

is given by

$$\mathbf{x}_0 = \mathbf{x}_0^B + (\mathbf{B}^{-1} + \hat{\mathbf{H}}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{H}})^{-1} \hat{\mathbf{H}}^T \hat{\mathbf{R}}^{-1} \hat{\mathbf{d}}, \quad \hat{\mathbf{d}} = \hat{\mathbf{H}}(\mathbf{x}_0^B - \hat{\mathbf{y}})$$

Singular value decomposition

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

$$\hat{\mathbf{H}} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{V}^T$$

Then the optimal analysis can be written as

$$\mathbf{x}_0 = \mathbf{x}_0^B + \sum_j \frac{\lambda_j^2}{\mu^2 + \lambda_j^2} \frac{\mathbf{u}_j^T \hat{\mathbf{d}}}{\lambda_j} \mathbf{v}_j$$

where $\mu^2 = \dfrac{\sigma_O^2}{\sigma_B^2}$.

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

Variable transformations

$\mathbf{B} = \sigma_B^2 \mathbf{F}_B$ and $\hat{\mathbf{R}} = \sigma_O^2 \mathbf{F}_R$ and

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

Variable transformations

$\mathbf{B} = \sigma_B^2 \mathbf{F}_B$ and $\hat{\mathbf{R}} = \sigma_O^2 \mathbf{F}_R$ and define new variable $z := \mathbf{F}_B^{-1/2}(\mathbf{x}_0 - \mathbf{x}_0^B)$

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

Variable transformations

$\mathbf{B} = \sigma_B^2 \mathbf{F}_B$ and $\hat{\mathbf{R}} = \sigma_O^2 \mathbf{F}_R$ and define new variable $z := \mathbf{F}_B^{-1/2} (\mathbf{x}_0 - \mathbf{x}_0^B)$

$$\hat{J}(\mathbf{z}) = \mu^2 \|\mathbf{z}\|_2^2 + \|\mathbf{F}_R^{-1/2} \hat{\mathbf{d}} - \mathbf{F}_R^{-1/2} \hat{\mathbf{H}} \mathbf{F}_B^{-1/2} \mathbf{z}\|_2^2$$

μ^2 can be interpreted as a regularisation parameter.

Relation between 4D-Var and Tikhonov regularisation

Solution to the optimisation problem

$$J(\mathbf{x}_0) = (\mathbf{x}_0 - \mathbf{x}_0^B)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^B) + (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))^T \hat{\mathbf{R}}^{-1} (\hat{\mathbf{y}} - \hat{\mathbf{H}}(\mathbf{x}_0))$$

Variable transformations

$\mathbf{B} = \sigma_B^2 \mathbf{F}_B$ and $\hat{\mathbf{R}} = \sigma_O^2 \mathbf{F}_R$ and define new variable $z := \mathbf{F}_B^{-1/2} (\mathbf{x}_0 - \mathbf{x}_0^B)$

$$\hat{J}(\mathbf{z}) = \mu^2 \|\mathbf{z}\|_2^2 + \|\mathbf{F}_R^{-1/2} \hat{\mathbf{d}} - \mathbf{F}_R^{-1/2} \hat{\mathbf{H}} \mathbf{F}_B^{-1/2} \mathbf{z}\|_2^2$$

μ^2 can be interpreted as a regularisation parameter.
This is the well-known Tikhonov regularisation!

Outline

UNIVERSITY OF BATH

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