# Data Assimilation applied to the Three-Body Problem

Melina Freitag

Department of Mathematical Sciences
University of Bath

27th March 2008

# Outline



UNIVERSITY OF BATH

# 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)

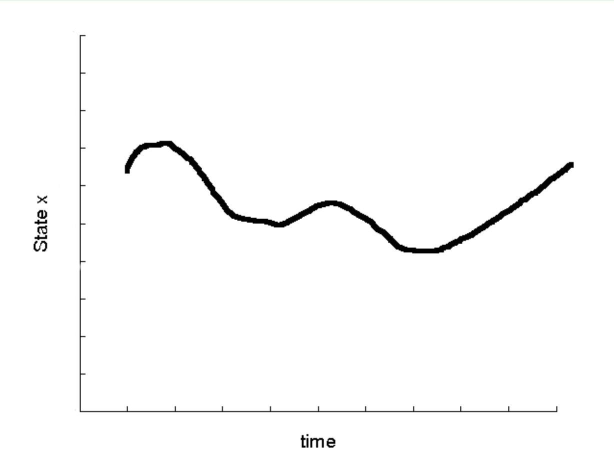



Figure: Background state $\mathbf{x}^B$

# Schematics of DA

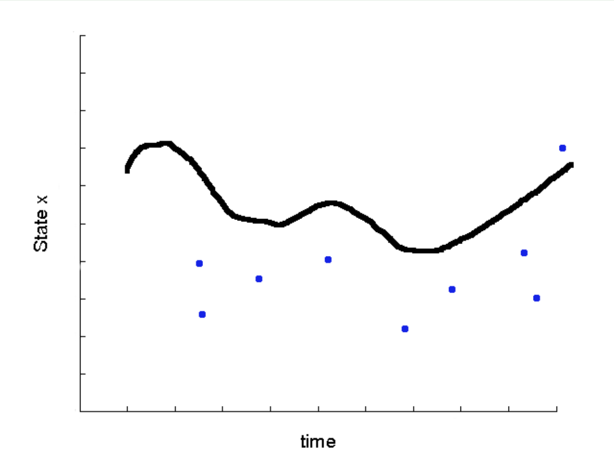



Figure: Observations $\mathbf{y}$

# Schematics of DA

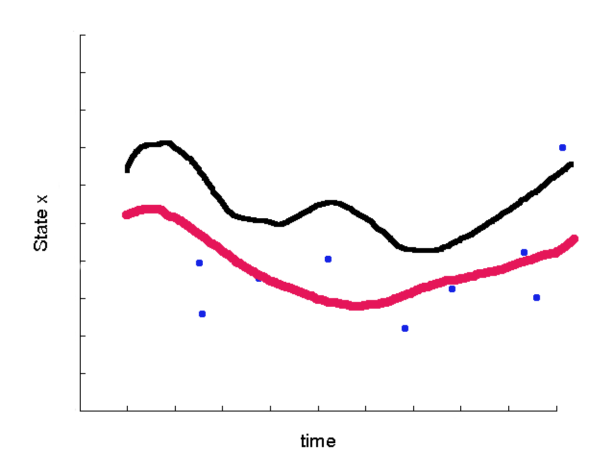



Figure: Analysis $\mathbf{x}^A$ (consistent with observations and model dynamics)

### Underdeterminacy

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

# Outline

# 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
- 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)

# Error variables

## Modelling the errors

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

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

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

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

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

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

- analysis error $\varepsilon^A = \mathbf{x}^A - \mathbf{x}^{\mathrm{Truth}}$ 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

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

# Assumptions

- 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 (3D-Var)

## Cost function

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

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

# Optimal least-squares estimater (3D-Var)

## Cost function

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

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

## Four-dimensional variational assimilation (4D-Var)

Minimisation of the cost function

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

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

UNIVERSITY OF BATH

# Minimisation of the 4D-Var cost function

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

- 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 $J_{Oi} = (\mathbf{y}_i - H_i(\mathbf{x}_i))^T \mathbf{d}_i$
- $\nabla J$ 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., $\ldots \tilde{\mathbf{x}}_0 = -\frac{1}{2}\nabla J_O$

UNIVERSITY OF BATH

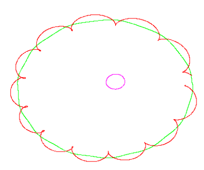
# Outline



UNIVERSITY OF BATH

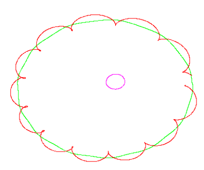
# Example - Three-Body Problem

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 (nondimensionalised)

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

# 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

# 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)

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

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

- subsequent forecast is take 5000 time steps
- $\mathbf{R}$ is diagonal with variances between $10^{-3}$ and $10^{-5}$

# Example- Three-Body problem

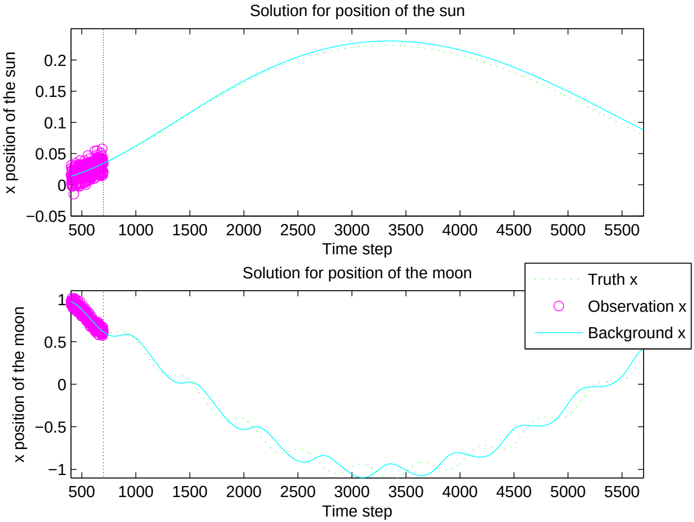



Figure: Truth trajectory with observations and background

# Example- Three-Body problem

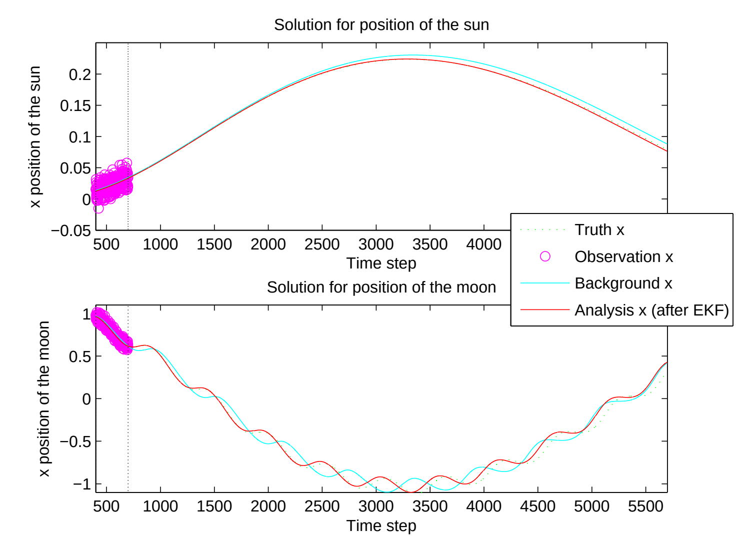



Figure: Analysis

# Example- Three-Body problem

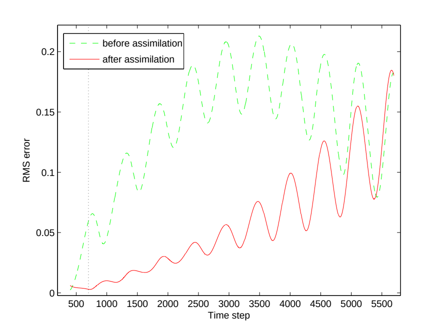



Figure: RMS error

# Example- Three-Body problem

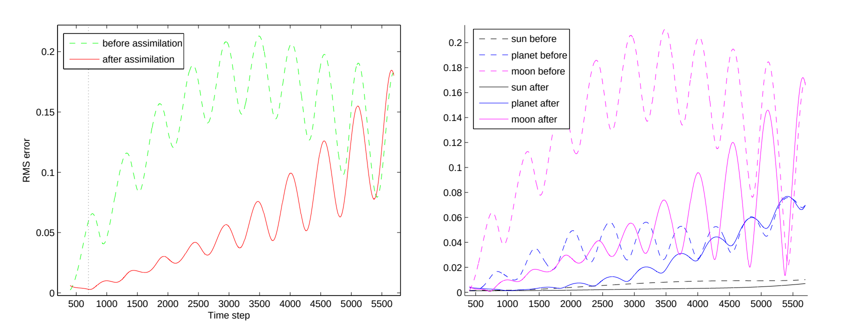



Figure: RMS error

# The Kalman Filter Algorithm

## State and error covariance forecast

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

## State and error covariance analysis

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

# Example - Three-Body Problem

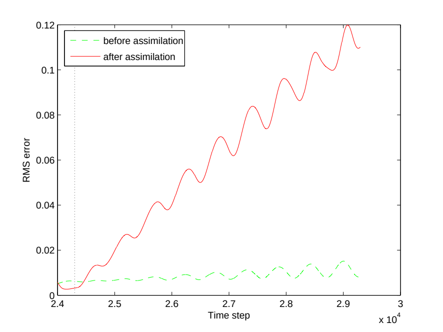



Figure: 4D-Var with $\mathbf{B} = \mathbf{I}$

# Example - Three-Body Problem

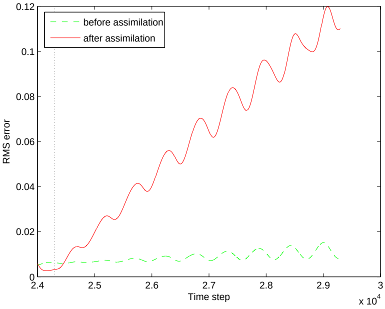



Figure: 4D-Var with $\mathbf{B} = \mathbf{I}$

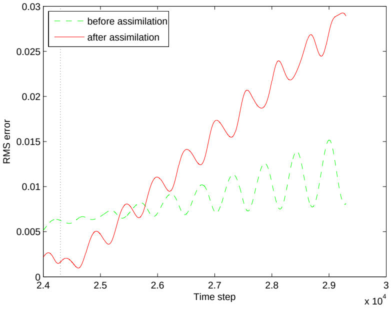



Figure: 4D-Var with $\mathbf{B} = \mathbf{P}^A$

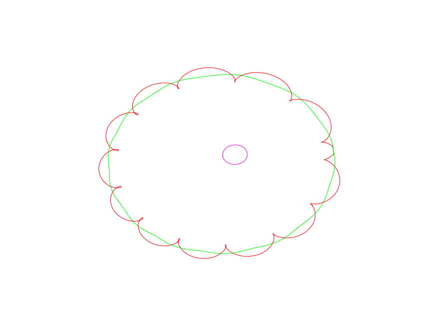
UNIVERSITY OF BATH

# Outline

# Changing the masses of the bodies

DA needs Model error!

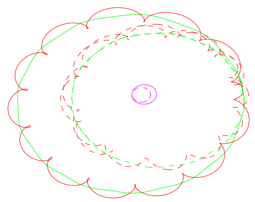
$$
\begin{aligned}
m_s = 1.0 &\quad\longrightarrow\quad m_s = 1.1\\
m_p = 0.1 &\quad\longrightarrow\quad m_p = 0.11\\
m_m = 0.01 &\quad\longrightarrow\quad m_m = 0.011
\end{aligned}
$$



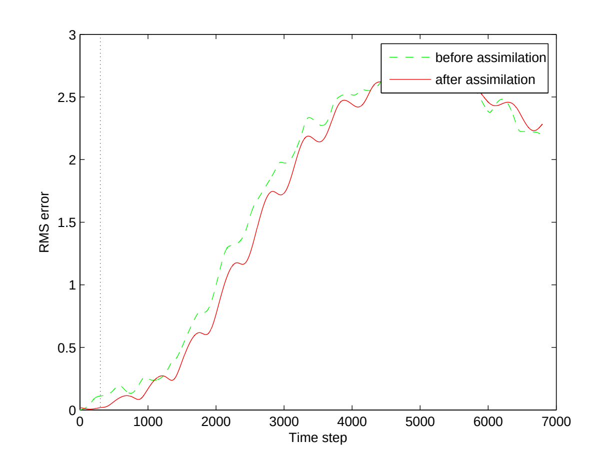
# Changing the masses of the bodies

DA needs Model error!

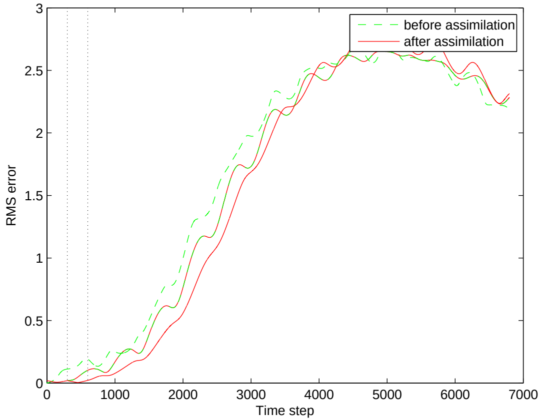
$$m_s = 1.0 \quad \rightarrow \quad m_s = 1.1$$
$$m_p = 0.1 \quad \rightarrow \quad m_p = 0.11$$
$$m_m = 0.01 \quad \rightarrow \quad m_m = 0.011$$



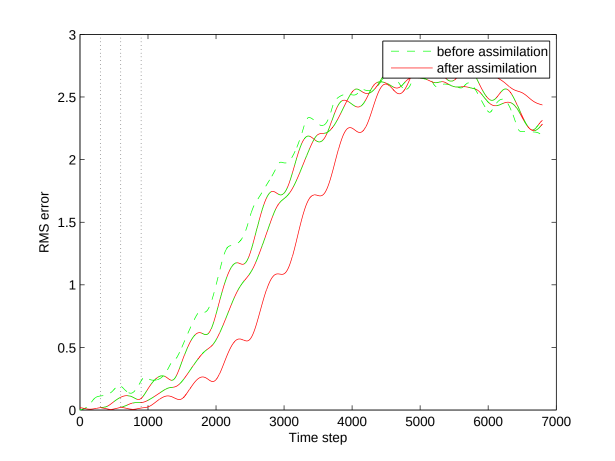
# Changing the masses of the bodies



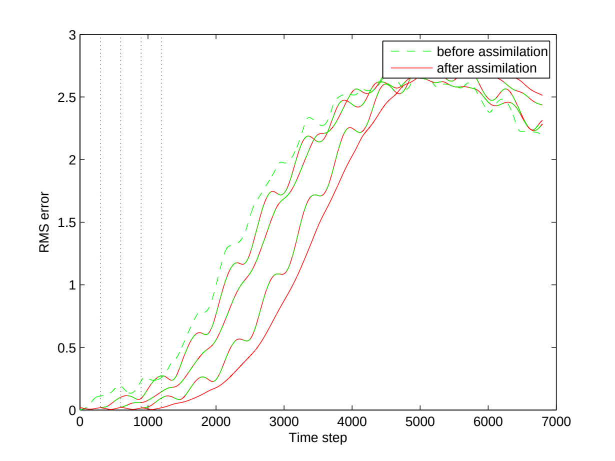
# Changing the masses of the bodies



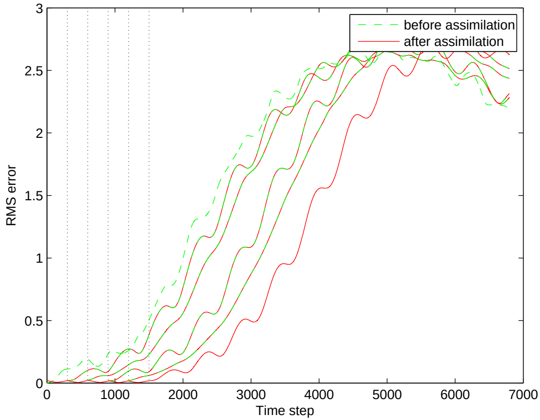
# Changing the masses of the bodies



# Changing the masses of the bodies

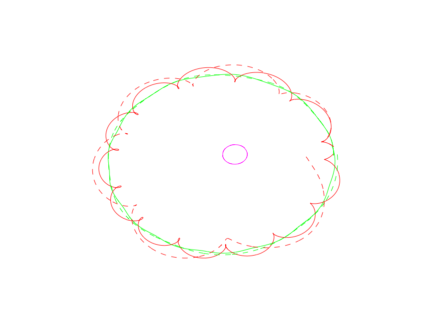


# Changing the masses of the bodies

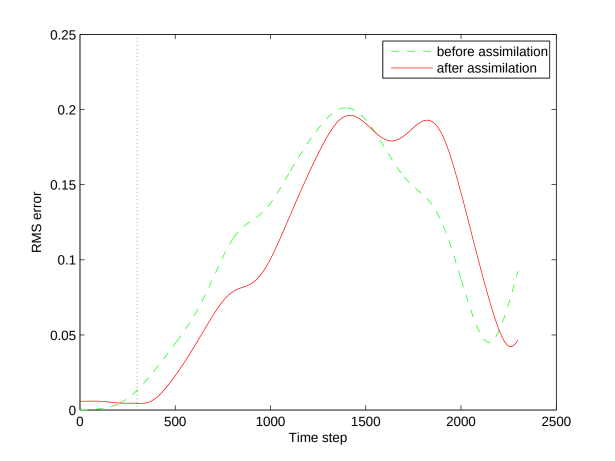


# Changing numerical method

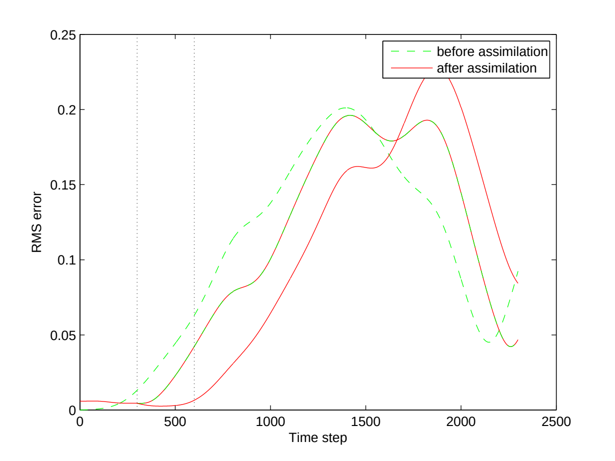
- Truth trajectory: 4th order Runge-Kutta method with local truncation error $\mathcal{O}(\Delta t^5)$
- Model trajectory: Explicit Euler method with local truncation error $\mathcal{O}(\Delta t^2)$



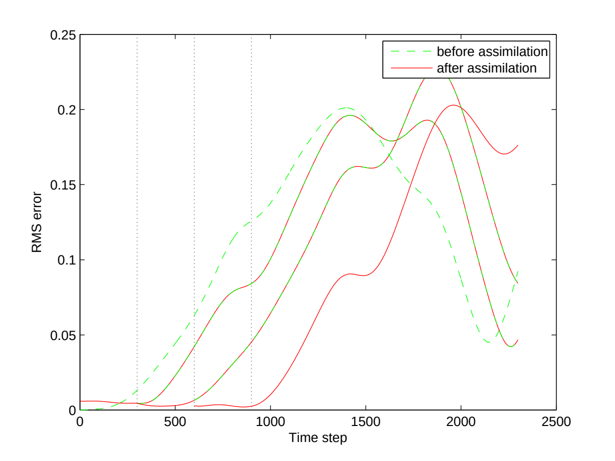
# Changing numerical method



# Changing numerical method

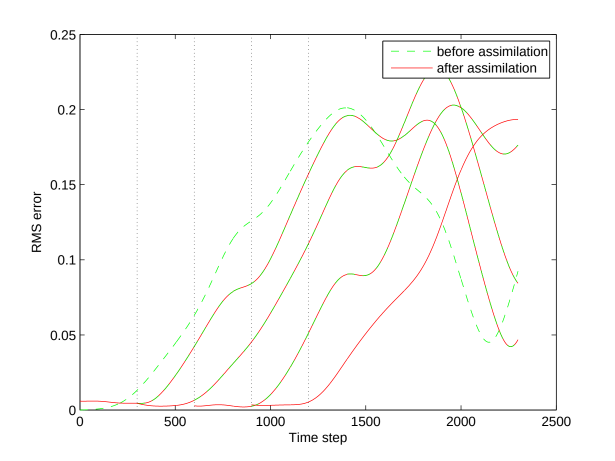


# Changing numerical method




UNIVERSITY OF BATH

# Changing numerical method



# Problem

**Estimation of the background error covariance matrix $\mathbf{B}$**

- Kalman Filter approach does not work

# Problem

**Estimation of the background error covariance matrix $\mathbf{B}$**

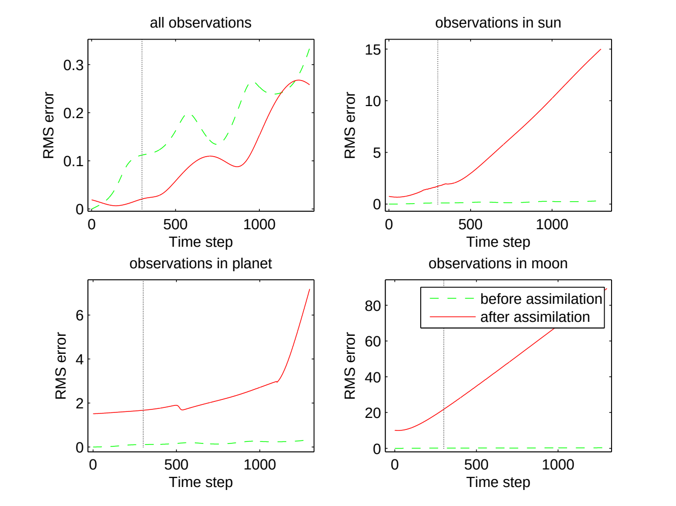
- Kalman Filter approach does not work
- $\mathbf{B} = \mathbf{I}$ appears to be the best result

# Problem

**Estimation of the background error covariance matrix $\mathbf{B}$**

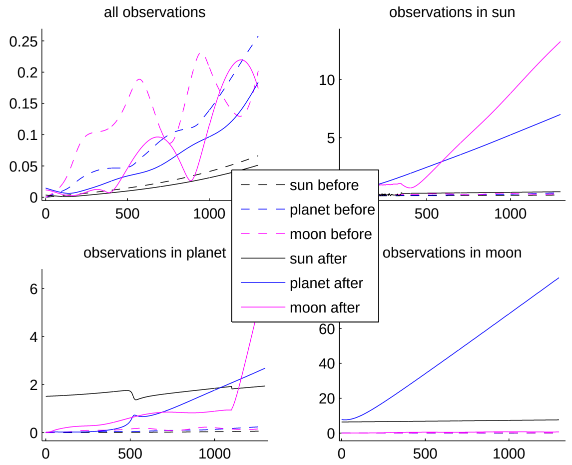
- Kalman Filter approach does not work
- $\mathbf{B} = \mathbf{I}$ appears to be the best result
- Problem too easy?

# Observations in different time scales $B = I$, large model error



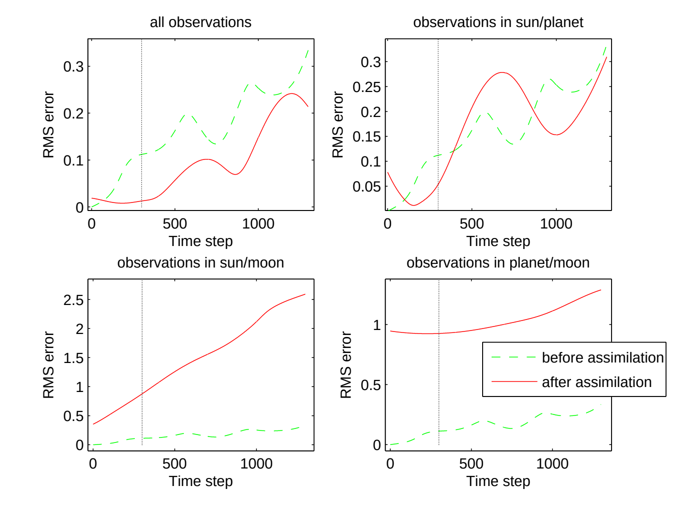

UNIVERSITY OF BATH

# Observations in different time scales $B = I$, large model error



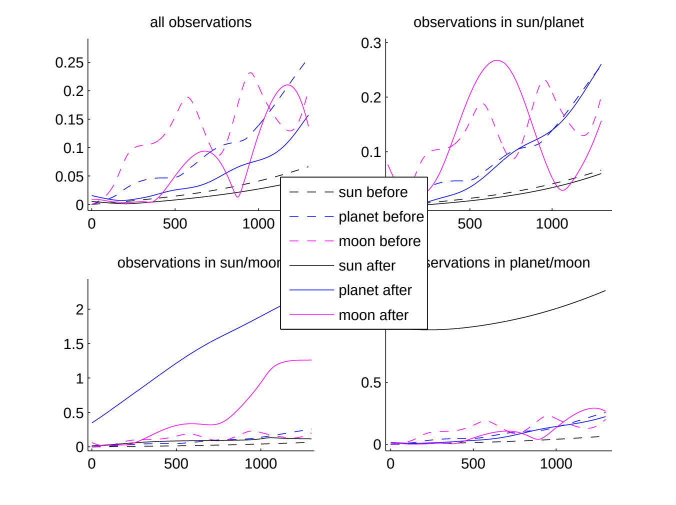

UNIVERSITY OF BATH

# Observations in different time scales $B = I$, large model error

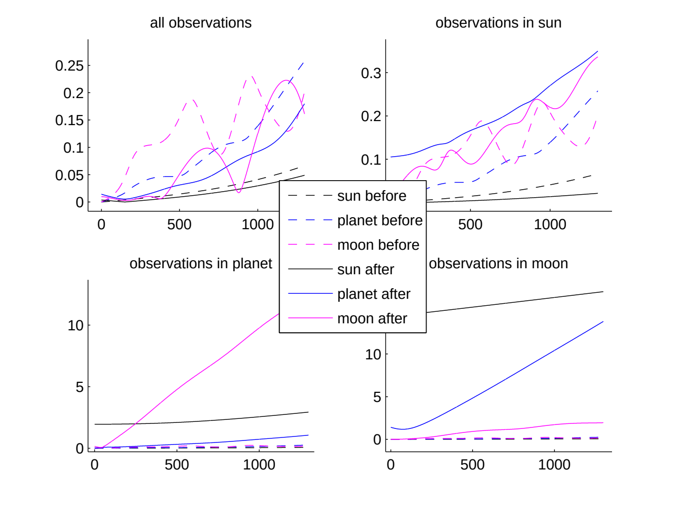



UNIVERSITY OF BATH

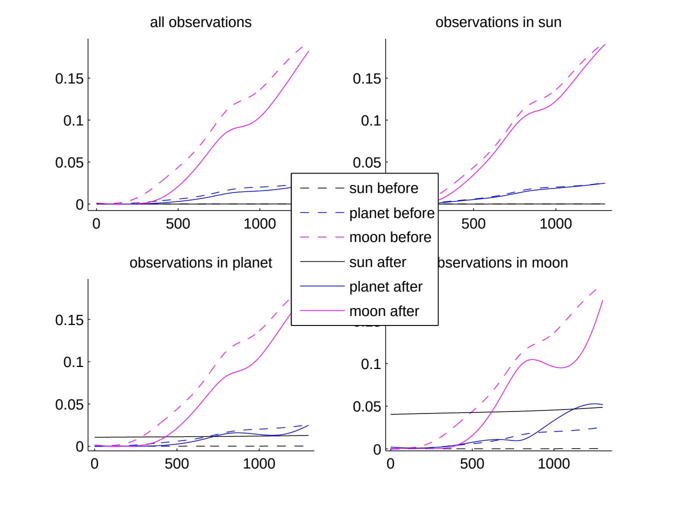
# Observations in different time scales $B = I$, large model error



# Perfect observations in different time scales $B = I$, large model error



# Perfect observations in different time scales $B = I$, small model error



# Making the Three-Body Problem chaotic

## Parameters (Chaotic shuffling of the moon)

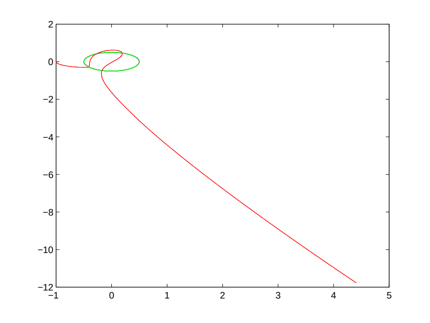
$$
\begin{aligned}
m_s &= 0.5 \\
m_p &= 0.5 \\
m_m &= 0.0
\end{aligned}
$$

Choose initial position and velocity of the moon such that problem becomes chaotic.

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

# Making the Three-Body Problem chaotic

## Solve using PRK

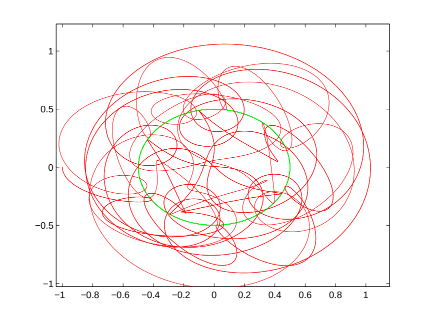



Problem: singularities in the numerical scheme as bodies approach each other:

$$H(\mathbf{q},\mathbf{p}) = \frac{1}{2}\sum_{\alpha}\frac{|\mathbf{p}_\alpha|^2}{m_\alpha} - \sum\sum_{\alpha<\beta}\frac{m_\alpha m_\beta}{|\mathbf{q}_\alpha - \mathbf{q}_\beta|}$$

# Making the Three-Body Problem chaotic

## Solve using PRK with adaptive time stepping




Time step $h = \dfrac{h_{\text{start}}}{r_{12}^{-2} + r_{13}^{-2} + r_{23}^{-2}}$

# Making the Three-Body Problem chaotic

## Adaptive time stepping

- Time step $h = \dfrac{h_{\text{start}}}{r_{12}^{-2} + r_{13}^{-2} + r_{23}^{-2}}$

University of Bath

# Making the Three-Body Problem chaotic

## Adaptive time stepping

- Time step $h = \dfrac{h_{\text{start}}}{r_{12}^{-2} + r_{13}^{-2} + r_{23}^{-2}}$
- Problem: Data Assimilation with adaptive time stepping?

# Making the Three-Body Problem chaotic

## Adaptive time stepping

- Time step $h = \dfrac{h_{\text{start}}}{r_{12}^{-2} + r_{13}^{-2} + r_{23}^{-2}}$
- Problem: Data Assimilation with adaptive time stepping?
- Truth trajectory - Model trajectory

# Making the Three-Body Problem chaotic

## Adaptive time stepping

- Time step $h = \dfrac{h_{\text{start}}}{r_{12}^{-2} + r_{13}^{-2} + r_{23}^{-2}}$
- Problem: Data Assimilation with adaptive time stepping?
- Truth trajectory - Model trajectory
- weighting

# Plans

- design a simple chaotic model of reduced order (Lorenz model)
- include several time scales (to model the atmosphere)

# Plans

- design a simple chaotic model of reduced order (Lorenz model)
- include several time scales (to model the atmosphere)
- identify and analyse model error and analyse influence of this model error onto the DA scheme
- analyse the influence of the error made by the numerical approximation (part of the model error) on the error in the DA scheme
- compare assimilation algorithms and optimisation strategies to reduce existing errors

# Plans

- design a simple chaotic model of reduced order (Lorenz model)
- include several time scales (to model the atmosphere)
- identify and analyse model error and analyse influence of this model error onto the DA scheme
- analyse the influence of the error made by the numerical approximation (part of the model error) on the error in the DA scheme
- compare assimilation algorithms and optimisation strategies to reduce existing errors
- improve the forecast of small scale features (like convective storms)