

# Practical Session: data assimilation experiments

Patrick Laloyaux, Marcin Chrust, Massimo Bonavita, Xavier Abellan

*Illustrate the main concepts of data assimilation*

*Hands-on exercises to improve critical thinking*

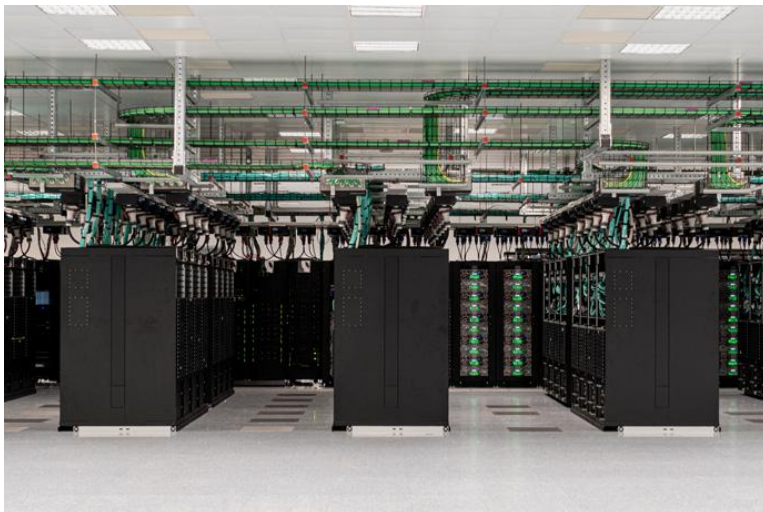
*Engaging environment, please ask questions!*

# The operational IFS and 4D-Var system

## The ECMWF model and assimilation software in numbers

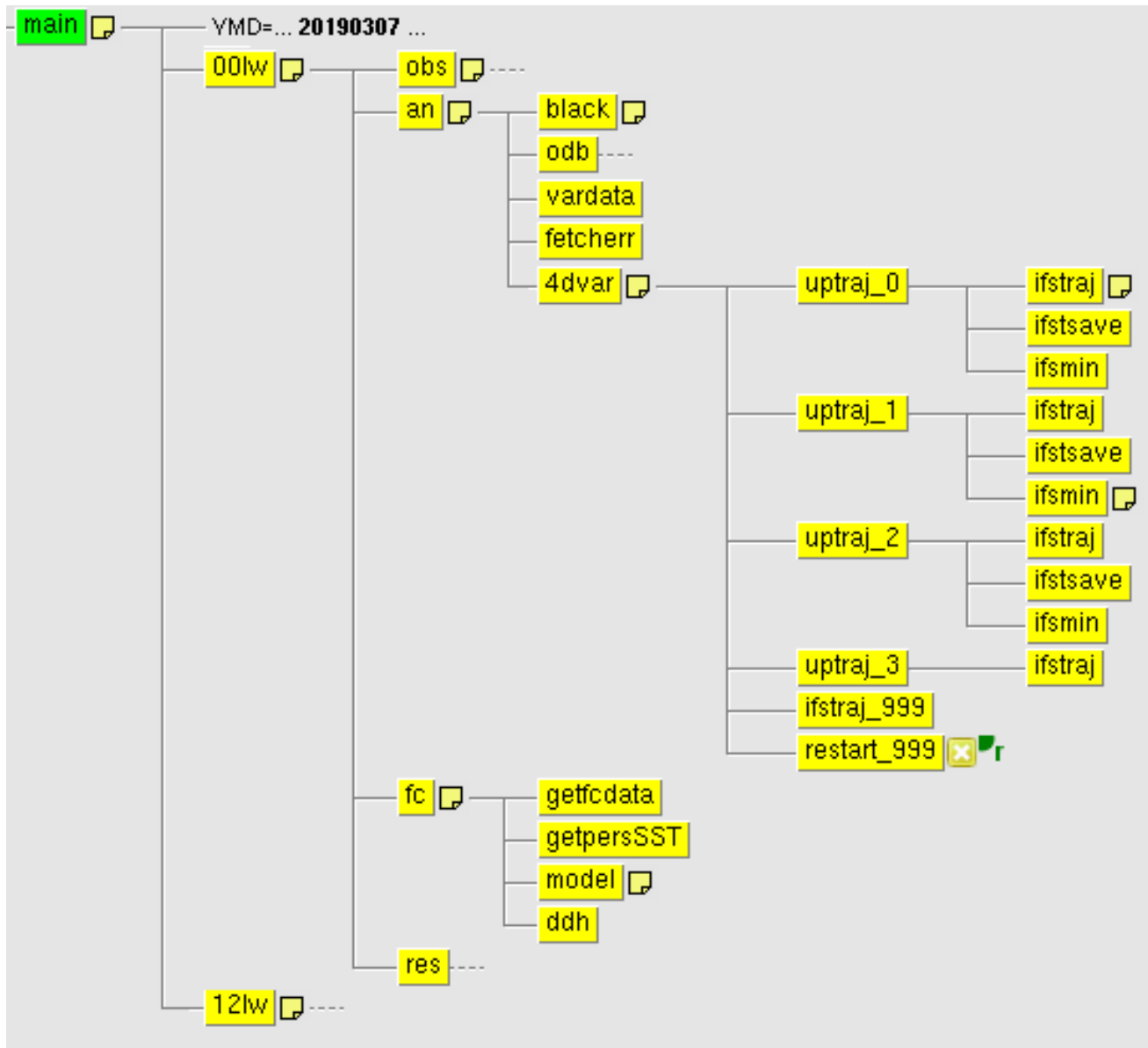
- 18,000 source code files
- 1,000,000 lines of code
- 100,000 if statements
- 45 minutes to solve one 4D-Var problem... using 25,000 CPUs

```
!-----  
! 1. Convert X into X-Xb : subtract background  
!-----  
IF (LHOOK) CALL DR_HOOK('CVAR2',0,ZHOOK_HANDLE)  
ASSOCIATE(YDDIM=>YDGEOMETRY%YRDIM, &  
& YDDIMV=>YDGEOMETRY%YRDIMV, YDGM=>YDGEOMETRY%YRGM, YDMP=>YDGEOMETRY%YRMP)  
ASSOCIATE(NPROMA=>YDDIM%NPROMA, &  
& NFLEVG=>YDDIMV%NFLEVG, NFLEVEL=>YDDIMV%NFLEVEL, &  
& NGPTOT=>YDGM%NGPTOT, &  
& MYLEVS=>YDMP%MYLEVS)  
IF (YD_JB_STRUCTURE%JB_DATA%LSUBBG) CALL SBSBGS(YDGEOMETRY, YDGMV, YDGMV5, YDFIELDS)  
  
! 3. Apply B matrix  
!-----  
  
! Parameters to be estimated  
  
IF (LVARBC) CALL YDVARBC%PARAM_GET(YD_JB_STRUCTURE%JB_DATA%LSUBBG, YDVAZX%PARAMS)  
IF (YDVAZX%LAM1D) THEN  
  II=0  
  DO JF=1, CVA_DATA%NVA1D  
    DO JS=1, NFLEVEL  
      IL=MYLEVS(JS)  
      II=II+1  
      IF (YD_JB_STRUCTURE%JB_DATA%TMEANUVER(IL, JF)/=0.) THEN  
        YDVAZX%LAMCV(II) = YD_JB_STRUCTURE%JB_DATA%SPJB%SP1D(JS, JF) /&  
          & YD_JB_STRUCTURE%JB_DATA%TMEANUVER(IL, JF)  
      ELSE  
        YDVAZX%LAMCV(II) = 0.0_JPRB  
      ENDIF  
    ENDDO  
  ENDDO  
ENDIF
```



# The operational IFS and 4D-Var system

## Job scheduler



Retrieve observations

Preprocess observations

Read B matrix

Run nonlinear model

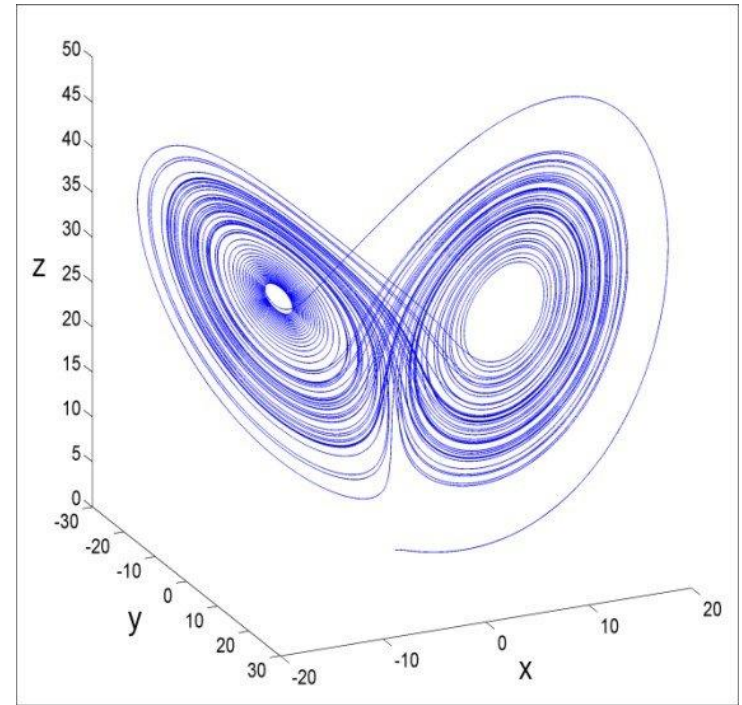
Minimise quadratic cost function

Produce forecast

# The Lorenz Model

The Lorenz system is a system of ordinary differential equations.

It is famous for having chaotic solutions for certain parameter values and initial conditions



In the practical sessions, we will use a more complex model

- Lorentz-95 model (40 variables)
- Twin experiments (the true state is known)
- 3D-Var and 4D-Var systems

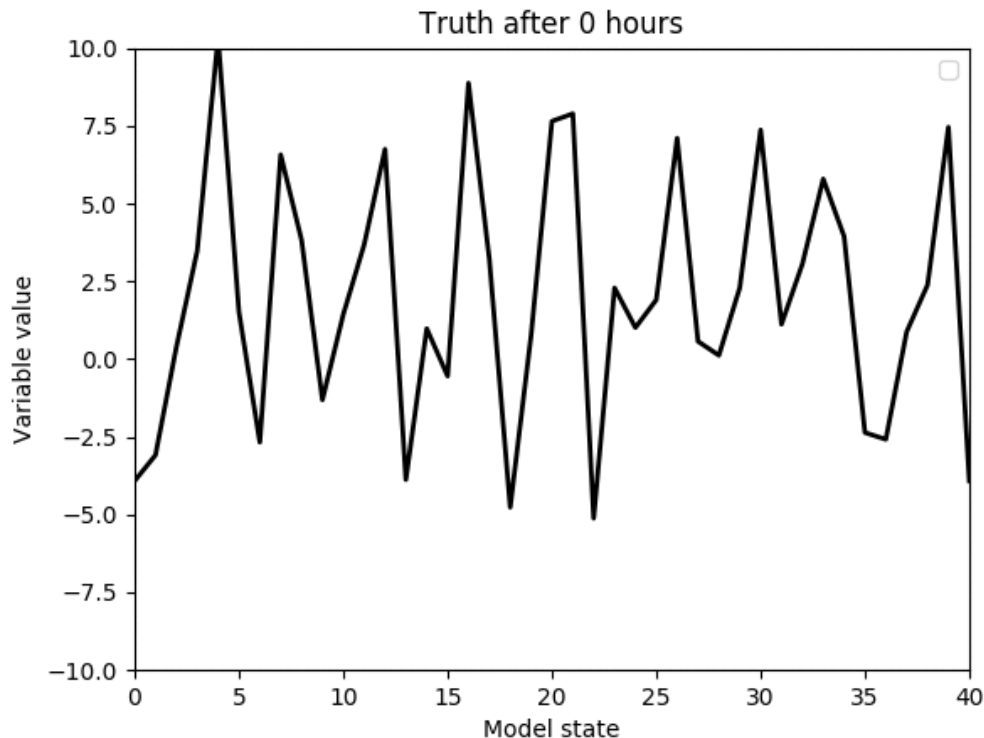
It is good enough to illustrate to most important concepts and issues

# The Lorenz Model

- The Lorenz-95 model is a widely-used low-dimensional dynamical system for data assimilation studies
- The system is defined by a set of coupled ordinary differential equations

$$dx_i/dt = -x_{i-2} x_{i-1} + x_{i-1} x_{i+1} - x_i + F \text{ for } i = 1, 2 \dots N$$

- For a range of values of  $F \approx 8$ , the system is chaotic, and has similar characteristics as an operational NWP system



The model state  $\mathbf{x}$  includes 40 variables

# How to start?

The image shows a web browser window displaying the JupyterLab interface. The browser's address bar shows the URL `traininglab35.ecmwf.europeanweather.cloud`. The browser tabs include "Home | Microsoft 365 Copilot", "Mail - Patrick Lalouaux - Outlook", "Training course: Data assimilation &...", "DA training course - Day 3 - All Docu...", "JupyterLab", and "Jupyter Remote Desktop Proxy".

The JupyterLab interface has a top menu bar with "File", "Edit", "View", "Run", "Kernel", "Git", "Tabs", "Settings", and "Help". Below the menu bar is a "Launcher" tab. On the left side, there is a file browser showing a tree view of the file system with columns for "Name" and "Modified". The file browser shows a root directory with subdirectories: "DAcourse" (modified 5d ago), "Desktop" (modified 5d ago), "Documents", "Downloads", "Music", "Pictures", "Public", "Templates", and "Videos" (all modified "now").

The main area of the Launcher is divided into sections:

- Notebook**: Contains two options: "Python 3 (ipykernel)" and "X11 Desktop [?]"
- Console**: Contains one option: "Python 3 (ipykernel)"
- Other**: Contains five options: "Terminal", "Text File", "Markdown File", "Python File", and "Show Contextual Help".

At the bottom left of the interface, there is a "Simple" toggle switch and a status bar showing "0 \$ \_ 0". At the bottom right, it says "Launcher 1" with a notification bell icon.

# How to start?

The screenshot displays a JupyterLab environment. On the left, a terminal window shows the output of a model run, including statistics and state information. On the right, a PDF viewer displays a plot and task instructions.

**Terminal Output:**

```
Info : Min=-5.86661, Max=8.26851, Average=2.06758
Info : fc
Info : Valid time: 2010-01-02T00:00:00Z
Info : Min=-6.06334, Max=8.62331, Average=2.05496
Info : fc
Info : Valid time: 2010-01-02T01:30:00Z
Info : Min=-6.2274, Max=8.92875, Average=2.03762
Info : fc
Info : Valid time: 2010-01-02T03:00:00Z
Info : Min=-6.35826, Max=9.18161, Average=2.01468
Info : ModelState:forecast: Finished forecast, time is 2010-01-02T03:00:00Z
Info : End NL
Info : Valid time: 2010-01-02T03:00:00Z
Info : Min=-6.35826, Max=9.18161, Average=2.01468
Info : MakeObs: Finished observation generation.
Info : Final state:
Info : Valid time: 2010-01-02T03:00:00Z
Info : Min=-6.35826, Max=9.18161, Average=2.01468
Info : Generated observation:
Info : Lorenz 95 nobs= 15 Min=-0.539456, Max=7.04957, Average=3.06201
student@traininglab35:~/DAcourse/DA_TC_2021$
student@traininglab35:~/DAcourse/DA_TC_2021$
student@traininglab35:~/DAcourse/DA_TC_2021$
student@traininglab35:~/DAcourse/DA_TC_2021$
```

**PDF Content:**

**python I95\_plotTrajMod.py**

**Task 3: Run a cycle of 6-hour 3D-Var with many observations**

Run a 3D-Var analysis  
I95\_4dvar.x I95\_3dvar\_6h.xml

Plot the truth, background, 3dvar analysis in the middle of the assimilation window (2010-01-02T00:00:00)  
python I95\_plotTraj.py 3dvar\_6h &

Plot the background error, analysis error and analysis increment in the middle of the assimilation window (2010-01-02T00:00:00)  
python I95\_plotDiffs.py 3dvar\_6h &

**Questions:**

- Where are the largest differences between the background and the analysis?
- What is the relationship between background error, analysis error and analysis increment?

# Task 1: Compilation and make the executables

Run the magic script!

```
source DAcourse/DA_TC_2021/makeoops
```

→ the file TC\_oops\_2021.pdf is a copy of the slides

## Task 2: Generate input data for the assimilation

Generate the "truth":

`I95_forecast.x I95_truth.xml`

Generate observations from the truth

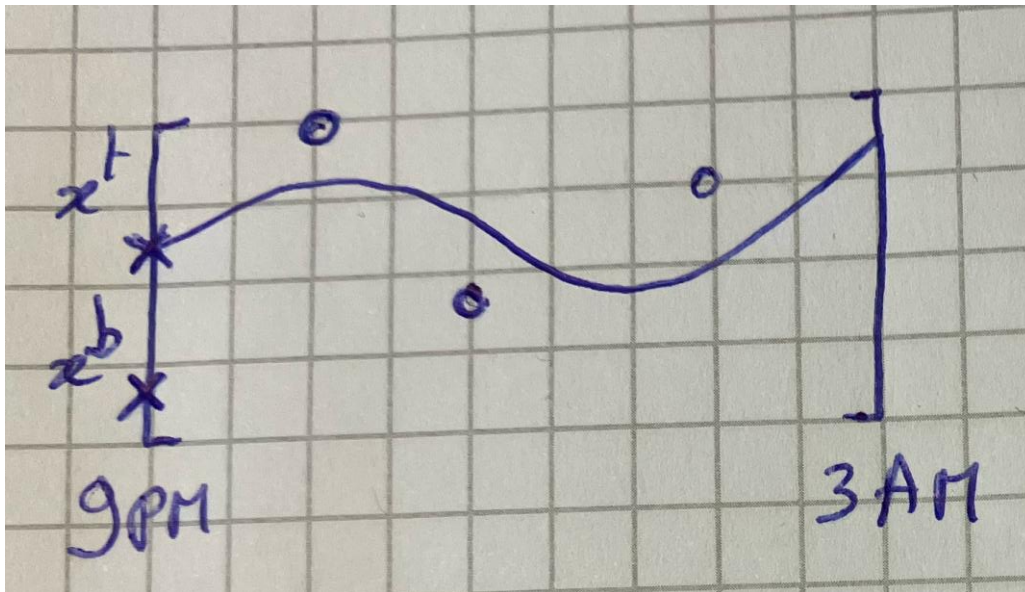
`I95_makeobs.x I95_makeobs_6h.xml`

Generate the background trajectory:

`I95_forecast.x I95_forecast.xml`

Plot the true model trajectory over the first three days

`python I95_plotTrajMod.py`



# Task 3: Run a cycle of 6-hour 3D-Var with many observations

Run a 3D-Var analysis

```
I95_4dvar.x I95_3dvar_6h.xml
```

Plot the truth, background, 3dvar analysis in the middle of the assimilation window (2010-01-02T00:00:00)

```
python I95_plotTraj.py 3dvar_6h &
```

Plot the background error, analysis error and analysis increment in the middle of the assimilation window (2010-01-02T00:00:00)

```
python I95_plotDiffs.py 3dvar_6h &
```

Questions:

- Where are the largest differences between the background and the analysis?
- What is the relationship between background error, analysis error and analysis increment?

## Task 4: Run a cycle of 6-hour 3D-Var with a single observation

Generate a single observation from the truth at the beginning of the window

```
I95_makeobs.x I95_makeobs_6h_single_begin.xml
```

Run a 3D-Var analysis with a single observation

```
I95_4dvar.x I95_3dvar_6h_single_begin.xml
```

Plot the truth, background, 3dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 3dvar_6h_single &
```

Plot the background error, analysis error and analysis increment

```
python I95_plotDiffs.py 3dvar_6h_single &
```

# Task 4: Run a cycle of 6-hour 3D-Var with a single observation

Edit the file `I95_3dvar_6h_single_begin.xml` and change the parameters of the covariance matrix:

- standard deviation
- length scale

Questions:

- How does the increment size evolve when the background standard deviation is increased/decreased?
- How does the increment spread when the background correlation lengthscale is increased/decreased?
- Would you have the same results if the same observation was available at the end of the window?

## Task 4: Run a cycle of 3D-Var with a single observation

Remember the **Linear Analysis Equation**:

$$x_a = x_b + K(y - Hx_b)$$
$$\text{where } K = BH^T (HBH^T + R)^{-1}$$

For a single observation, located at a gridpoint:

$$H = (0, \dots, 0, 1, 0, \dots, 0)$$

Hence

$$x_a - x_b = K(y - Hx_b)$$
$$= B \begin{pmatrix} 0 \\ \vdots \\ z \\ \vdots \\ 0 \end{pmatrix} \quad \text{where } z = (HBH^T + R)^{-1} (y - Hx_b)$$

That is,  $x_a - x_b \propto$  a column of B

# Task 5: Run a cycle of 24-hour 3D-Var with many observations

Generate observations from the truth:

```
I95_makeobs.x I95_makeobs_24h.xml
```

Run a 3D-Var analysis

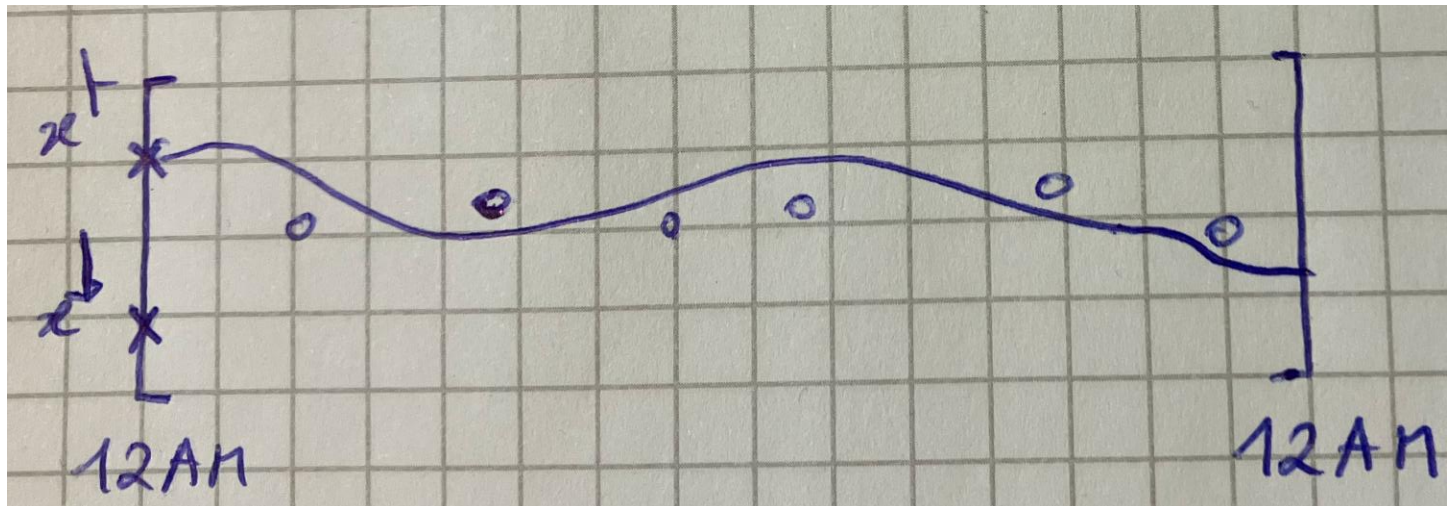
```
I95_4dvar.x I95_3dvar_24h.xml
```

Plot the truth, background, 3dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 3dvar_24h &
```

Plot the background error, analysis error and analysis increment

```
python I95_plotDiffs.py 3dvar_24h &
```



## Task 5: Run a cycle of 24-hour 3D-Var with many observations

Compare the analysis error from the 6-h and the 24-h 3dvar

```
python I95_plotErr.py 3dvar_6h 3dvar_24h &
```

Plot the histogram of departures for the 6h and the 24h windows

```
python I95_plotHist.py 3dvar_6h 3dvar_24h &
```

Plot the departures with respect to the time in the assimilation window

```
python I95_plotTime.py 3dvar_6h 3dvar_24h &
```

Questions:

- What is the impact of having more observations over a longer window on the analysis error?
- What is the problem to run a 3D-Var with a long assimilation window?
- How can we assess the performance of an assimilation system when the true state is unknown?

# Task 6: Run a cycle of 6-hour 4D-Var with many observations

Run 4D-Var

```
I95_4dvar.x I95_4dvar_6h.xml
```

Plot the truth, background, 4dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 4dvar_6h &
```

Compare the analysis error for 3dvar and 4dvar

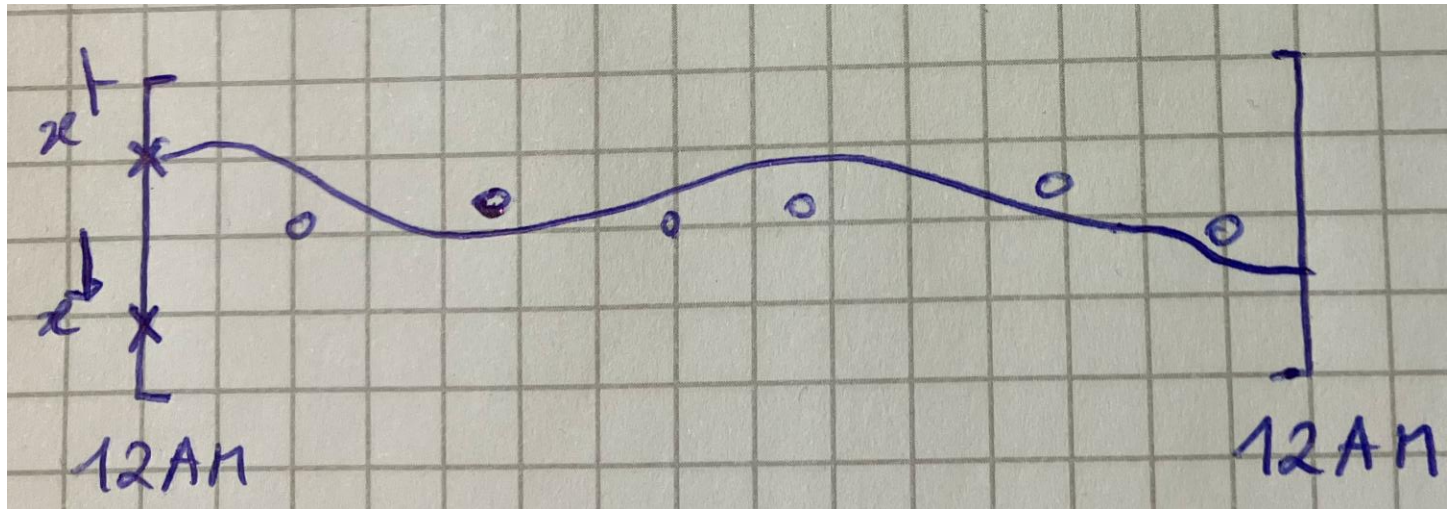
```
python I95_plotErr.py 3dvar_6h 4dvar_6h &
```

Plot the histogram of departures for 3dvar and 4dvar

```
python I95_plotHist.py 3dvar_6h 4dvar_6h &
```

## Task 6: Run a cycle of 6-hour 4D-Var with many observations

4D-Var produces an increment and an analysis at the beginning of the window. To compare with the 3D-Var analysis (valid at 00:00) the 4D-Var analysis is integrated in time from 12:00 to 00:00



Questions:

- Does 4D-Var produce a better analysis compared to 3D-Var?

# Task 7: Run a cycle of 6-hour 4D-Var with a single observation

Generate the observations at the beginning of the window

```
I95_makeobs.x I95_makeobs_6h_single_begin.xml
```

Run 4D-Var

```
I95_4dvar.x I95_4dvar_6h_single_begin.xml
```

Plot the background error, analysis error and analysis increment

```
python I95_plotDiffs.py 4dvar_6h_single_begin &
```

Generate the observations at the end of the window

```
I95_makeobs.x I95_makeobs_6h_single_end.xml
```

Run 4D-Var

```
I95_4dvar.x I95_4dvar_6h_single_end.xml
```

Plot the background error, analysis error and analysis increment

```
python I95_plotDiffs.py 4dvar_6h_single_end &
```

Questions:

- Does the observation time matter in 4D-Var?
- Which mathematical operators control the information spread in 4D-VAR?

## Task 7: Run a cycle of 24-hour 4D-Var with a single observation

Confirm your last answer by running a 24-hour 4D-Var with a single observation at the end of the assimilation window

Generate the observations at the beginning of the window

```
I95_makeobs.x I95_makeobs_24h_single_end.xml
```

Run 4D-Var

```
I95_4dvar.x I95_4dvar_24h_single_end.xml
```

Plot the background error, analysis error and analysis increment

```
python I95_plotDiffs.py 4dvar_24h_single_end &
```

## Task 7: Run a cycle of 24-hour 4D-Var with a single observation

The increment computed by the 4D-Var is valid at the beginning of the window and can be expressed as:

$$\delta \mathbf{x} = \mathbf{B} \mathbf{M}^T \mathbf{H}^T \left( \mathbf{H} \mathbf{M} \mathbf{B} \mathbf{M}^T \mathbf{H}^T + \mathbf{R} \right)^{-1} (y - \mathcal{G}(x_b))$$

For a single observation at time  $k$ , located at a gridpoint:

$$\mathbf{H} = (0, \dots, 0, 1, 0, \dots, 0)$$

Hence

$$\delta \mathbf{x} = \mathbf{B} \mathbf{M}_k^T \begin{pmatrix} 0 \\ \vdots \\ z \\ \vdots \\ 0 \end{pmatrix} \text{ where } z = \left( \mathbf{H} \mathbf{M}_k \mathbf{B} \mathbf{M}_k^T \mathbf{H}^T + \mathbf{R} \right)^{-1} (y_k - \mathcal{G}_k(x_b))$$

That is,  $\delta \mathbf{x} \propto$  a column of  $\mathbf{B} \mathbf{M}_k^T$

# Practical Session: data assimilation experiments

Patrick Laloyaux, Marcin Chrust, Massimo Bonavita, Xavier Abellan

# Task 8: Run a cycle of 24-hour 4D-Var with many observations

Run a 4D-Var analysis

```
I95_4dvar.x I95_4dvar_24h.xml
```

Plot the truth, background, 3dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 4dvar_24h &
```

Compare the analysis error from 3dvar and 4dvar

```
python I95_plotErr.py 3dvar_24h 4dvar_24h &
```

Plot the histogram of departures for 3dvar and 4dvar

```
python I95_plotHist.py 3dvar_24h 4dvar_24h &
```

Scatter diagram of model and observations

```
python I95_plotScatt.py 4dvar_24h &
```

Plot the departures with respect to the time in the assimilation window

```
python I95_plotTime.py 3dvar_24h 4dvar_24h &
```

Questions:

- Why does 4D-Var outperform 3D-Var with a long assimilation window?

## Task 9: What's going on?

A new version of the model and the assimilation system has been developed and a new set of observations is available... but something is wrong...

Could you find the reason of the poor performance of 4D-Var?

Run a 4D-Var analysis

```
I95_4dvar.x I95_4dvar_24h_bad3.xml
```

Compare the analysis error with the previous 4dvar

```
python I95_plotErr.py 4dvar_24h 4dvar_24h_bad3 &
```

Plot the histogram of departures

```
python I95_plotHist.py 4dvar_24h 4dvar_24h_bad3 &
```

Plot the departures with respect to the time in the assimilation window

```
python I95_plotTime.py 4dvar_24h 4dvar_24h_bad3 &
```

Scatter diagram of model and observations

```
python I95_plotScatt.py 4dvar_24h_bad3 &
```

Plot the truth, background, 4dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 4dvar_24h_bad3 &
```

## Task 9: What's going on?

A new version of the model and the assimilation system has been developed and a new set of observations is available... but something is wrong...

Could you find the reason of the poor performance of 4D-Var?

Run a 4D-Var analysis

```
I95_4dvar.x I95_4dvar_24h_bad2.xml
```

Compare the analysis error with the previous 4dvar

```
python I95_plotErr.py 4dvar_24h 4dvar_24h_bad2 &
```

Plot the histogram of departures

```
python I95_plotHist.py 4dvar_24h 4dvar_24h_bad2 &
```

Plot the departures with respect to the time in the assimilation window

```
python I95_plotTime.py 4dvar_24h 4dvar_24h_bad2 &
```

Scatter diagram of model and observations

```
python I95_plotScatt.py 4dvar_24h_bad2 &
```

Plot the truth, background, 4dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 4dvar_24h_bad2 &
```

## Task 9: What's going on?

A new version of the model and the assimilation system has been developed and a new set of observations is available... but something is wrong...

Could you find the reason of the poor performance of 4D-Var?

Run a 4D-Var analysis

```
I95_4dvar.x I95_4dvar_24h_bad4.xml
```

Compare the analysis error with the previous 4dvar

```
python I95_plotErr.py 4dvar_24h 4dvar_24h_bad4 &
```

Plot the histogram of departures

```
python I95_plotHist.py 4dvar_24h 4dvar_24h_bad4 &
```

Plot the departures with respect to the time in the assimilation window

```
python I95_plotTime.py 4dvar_24h 4dvar_24h_bad4 &
```

Scatter diagram of model and observations

```
python I95_plotScatt.py 4dvar_24h_bad4 &
```

Plot the truth, background, 4dvar analysis in the middle of the assimilation window

```
python I95_plotTraj.py 4dvar_24h_bad4 &
```

# Task 10: Number of outer and inner iterations in 4D-Var

The cost function can be written in terms of the **increment**  $\delta\mathbf{x}^{(m)}$ , and approximated by the quadratic function:

$$J(\delta\mathbf{x}^{(m)}) = \frac{1}{2} [\delta\mathbf{x}^{(m)} - \delta\mathbf{x}_b]^T \mathbf{P}^{b-1} [\delta\mathbf{x}^{(m)} - \delta\mathbf{x}_b] + \frac{1}{2} [\mathbf{d}^{(m)} - \mathbf{G}\delta\mathbf{x}^{(m)}]^T \mathbf{R}^{-1} [\mathbf{d}^{(m)} - \mathbf{G}\delta\mathbf{x}^{(m)}]$$

The incremental method treats the minimisation of  $J$  as a sequence of quadratic problems:

**for**  $m = 0, 1, \dots, M$  **do**

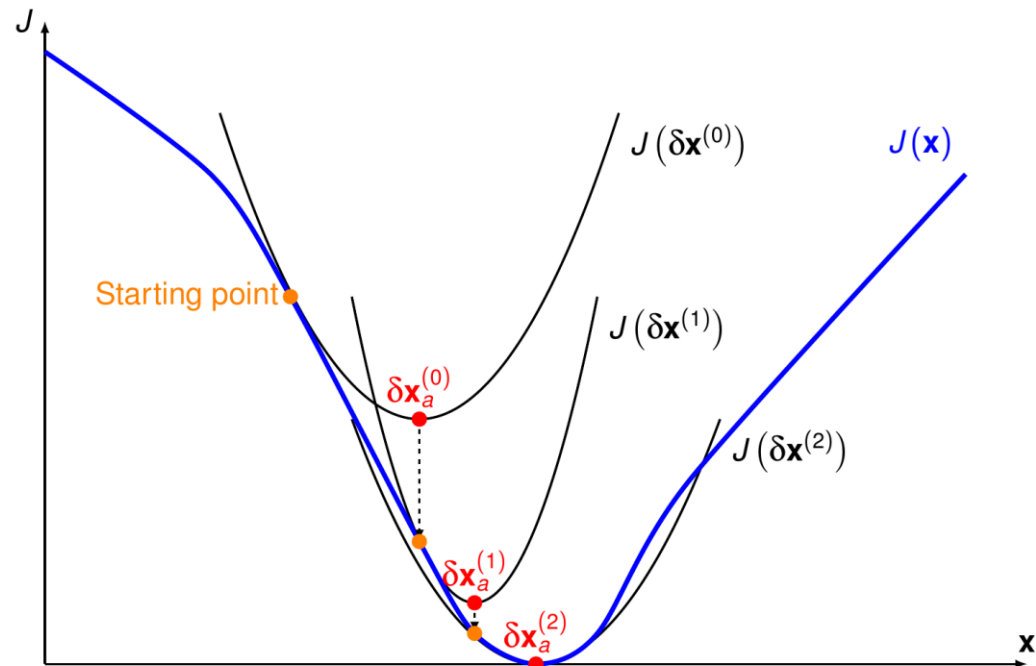
```
/* Minimise the quadratic cost function  $J(\delta\mathbf{x}^{(m)})$ 
```

```
 $\delta\mathbf{x}_a^{(m)} = \arg \min_{\mathbf{x}} [J(\delta\mathbf{x}^{(m)})];$ 
```

```
/* Set the new linearisation state
```

```
 $\mathbf{x}^{(m+1)} = \mathbf{x}^{(m)} + \delta\mathbf{x}_a^{(m)};$ 
```

**end**



## Task 10: Number of outer and inner iterations

Run a 24hour 4D-Var with one, two and three outer iterations (the number of inner iteration is set to 4)

```
I95_4dvar.x I95_4dvar_24h_outer1.xml > 4dvar_24h_outer1.out
```

```
I95_4dvar.x I95_4dvar_24h_outer2.xml > 4dvar_24h_outer2.out
```

```
I95_4dvar.x I95_4dvar_24h_outer3.xml > 4dvar_24h_outer3.out
```

Compare the analyses looking at the fit to the observations

```
python I95_plotTime3.py 4dvar_24h_outer1 4dvar_24h_outer2 4dvar_24h_outer3 &
```

Plot the evolution of the 4D-Var cost function during the minimization

```
python I95_plotConv.py 4dvar_24h_outer3 &
```

Change the number of inner iterations (from 4 to 10) by editing 3 times the variables ninner in I95\_4dvar\_24h\_outer3.xml

Plot the evolution of the 4D-Var cost function during the minimization

```
python I95_plotConv.py 4dvar_24h_outer3 &
```

Questions:

- What's the effect of having more inner iterations?
- Is this always guaranteed?

## Task 10: Number of outer and inner iterations

Run a 6hour 4D-Var with three outer iterations (the number of inner iteration is set to 4)

```
I95_4dvar.x I95_4dvar_6h_outer3.xml > 4dvar_6h_outer3.out
```

Plot the evolution of the 4D-Var cost function during the minimization

```
python I95_plotConv.py 4dvar_6h_outer3 &
```

Questions:

- How does the convergence speed evolve with respect to the assimilation window? Why?