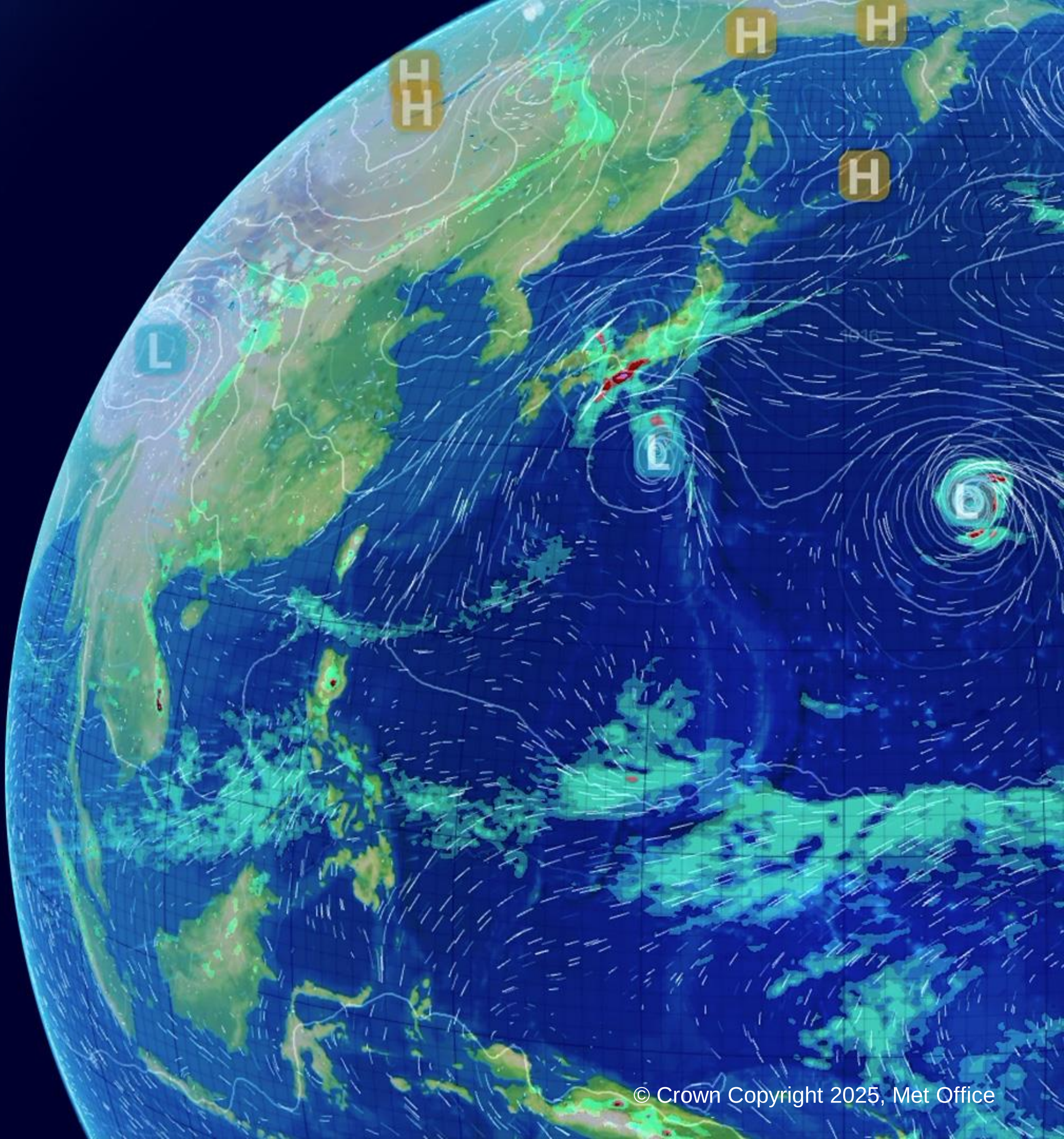
Met Office

The Joint Effort for Data assimilation Integration (JEDI) software

J. A. Waller

With thanks to C. Thomas, D. Simonin and many other members of the Satellite and Surface Assimilation group

11th June 2025



- JEDI Overview
- JEDI Components
- JEDI Configuration
- Summary

Met Office

JEDI Overview

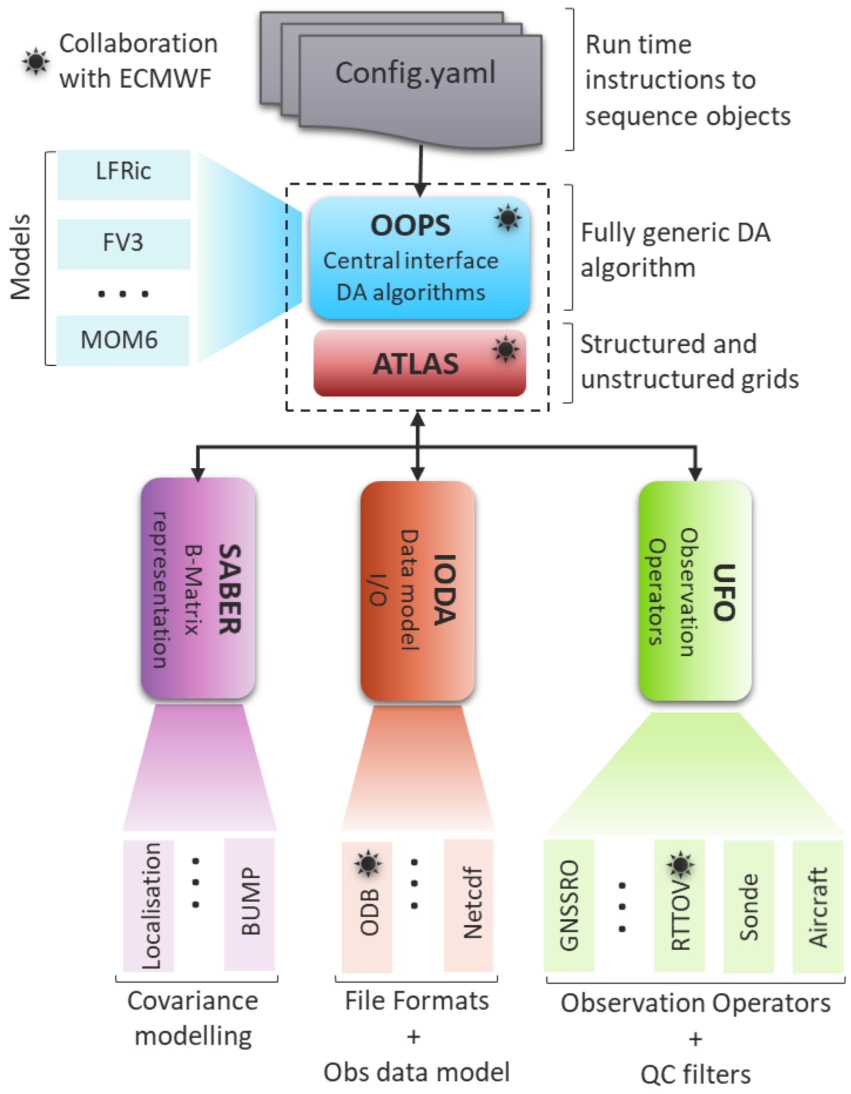
JEDI

JEDI is a collaborative development between JCSDA partners to develop a unified data assimilation system:

- → From toy models to Earth system coupled models
- → Unified observation (forward) operators (UFO)
- → For research and operations
- → Share as much as possible without imposing one approach (one system, multiple methodologies/configurations)

Motivation

Changes in HPC landscape	Fully exploit future generations of supercomputers Scalability, efficient I/O Memory and novel parallelism in era of large (1000s) ensemble DA
Technical	Increased modularity (more object-oriented capabilities) Expanded range of platforms (traditional HPC, cloud, laptop, etc)
More complex science	Flow-dependent QC, ensembles Strongly-coupled earth system DA, etc.
Future Applications	Nowcasting, cloud analysis, city-scale DA, composition DA, multi-models, etc.
Human	Current OPS/VAR too complex for wide-spread use Need to encourage wider collaboration (academic users, those working on other models)



JEDI

JEDI is a collaborative development between JCSDA partners to develop a unified data assimilation system:

→ From toy models to Earth system coupled models

New system

The logical chain of processing is applied "**dynamically**". <u>Code free of any science</u>

Collection of bricks functions, methods, classes, procedures

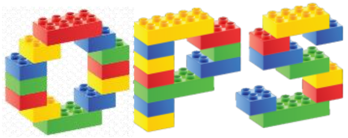
\+

```
window_begin: 2018-04-14T21:00:00Z
window_end: 2018-04-15T03:00:00Z
LinearObsOpTest:
coefTL: 0.1
toleranceTL: 1.0e-13
toleranceAD: 1.0e-11
Observations:
ObsTypes:
- ObsOperator:
name: VertInterp
VertCoord:
air_pressure
```

Configuration file

Instructions how to assemble the bricks

Assemble the bricks using modern computation techniques



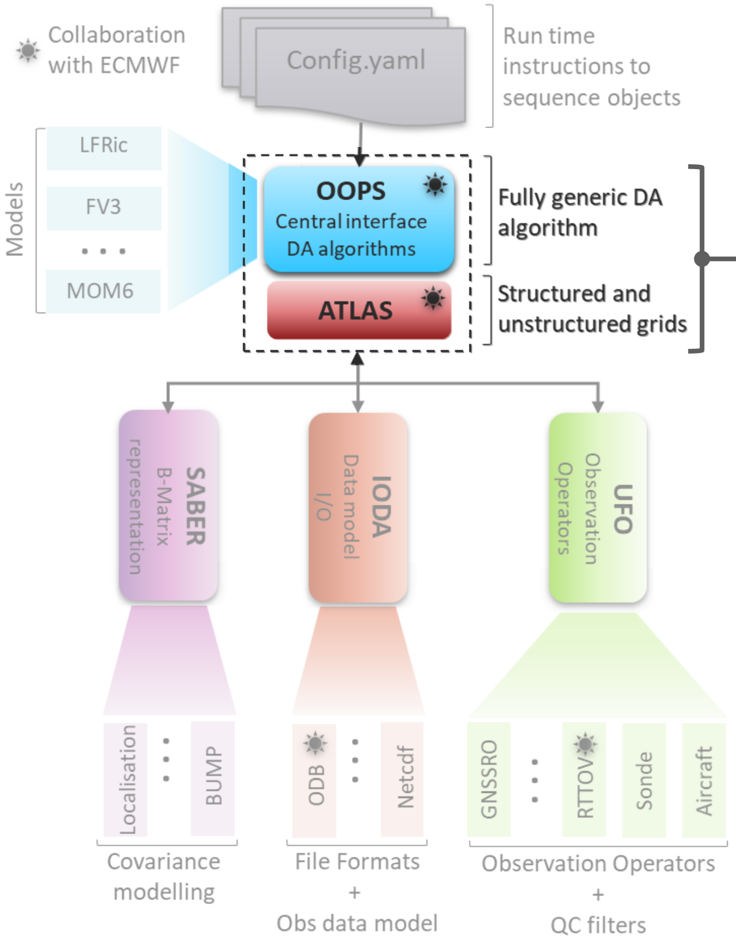
Mot...

Chan... lands...	Memory and novel parallelism in era of large (1000s) ensemble DA
Technical	Increased modularity (more object-oriented capabilities) Expanded range of platforms (traditional HPC, cloud, laptop, etc)
More complex science	Flow-dependent QC, ensembles Strongly-coupled earth system DA, etc.
Future Applications	Nowcasting, cloud analysis, city-scale DA, composition DA, multi-models, etc.
Human	Current OPS/VAR too complex for wide-spread use Need to encourage wider collaboration (academic users, those working on other models)

Met Office

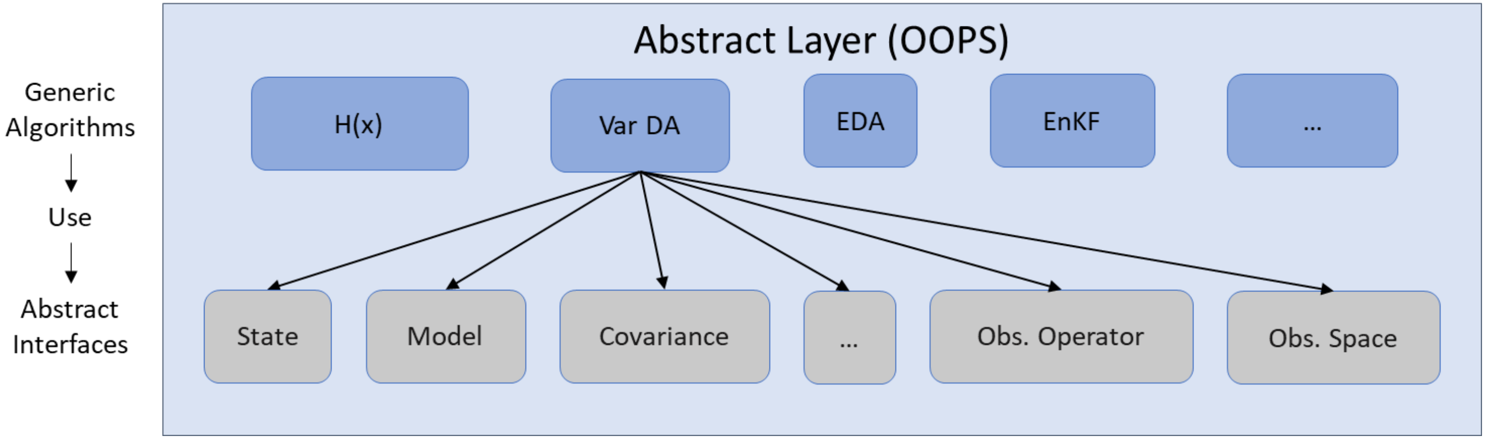
- JEDI Overview
- JEDI Components
- JEDI Configuration
- Summary

OOPS



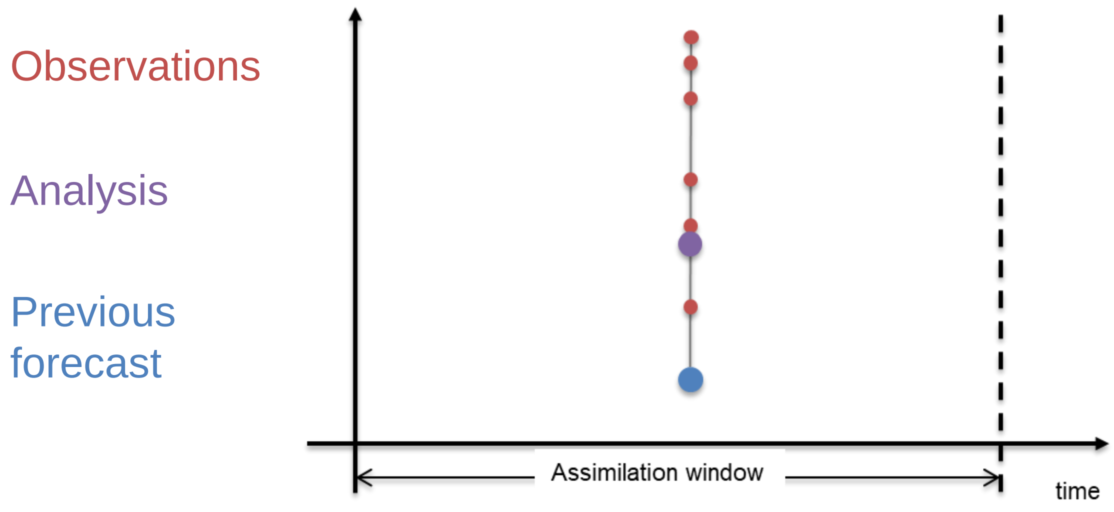
Object Oriented Prediction System

- OOPS is the abstract, model-agnostic DA system. It is top level of JEDI that orchestrates the configuration and execution of applications.
- OOPS contains:
  - Generic data assimilation algorithms
  - Abstract interfaces
  - Applications to run the data assimilation
  - Toy models
- OOPS is independent of the underlying model and physical system.



Data Assimilation Schemes within OOPS

- 3DVar



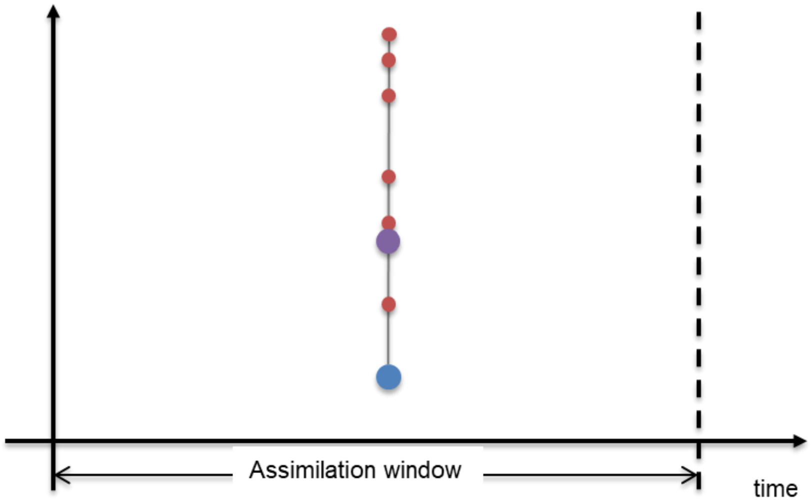
Data Assimilation Schemes within OOPS

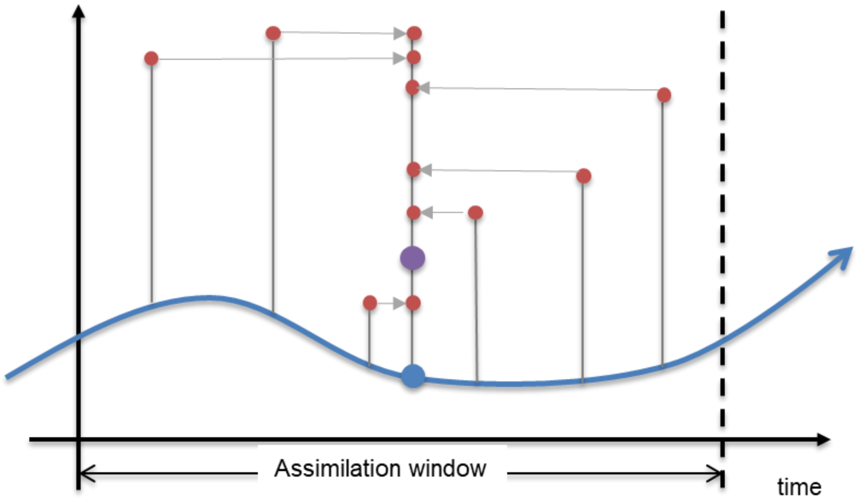
- 3DVar
- 3DFGAT

Observations

Analysis

Previous forecast





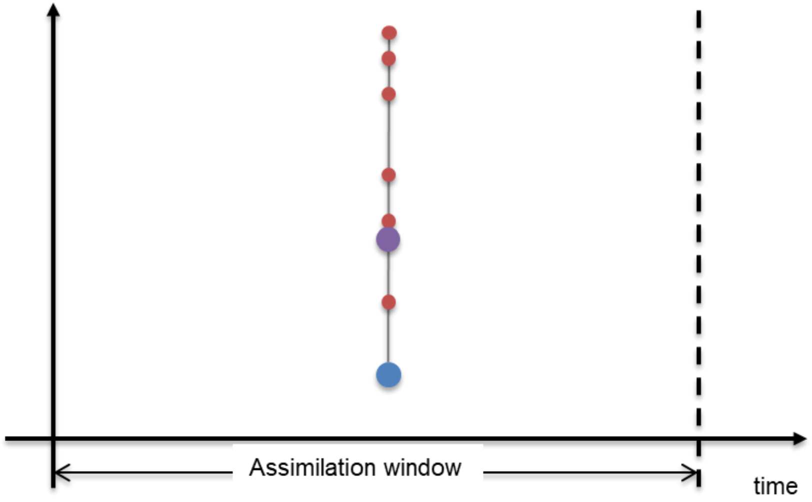
Data Assimilation Schemes within OOPS

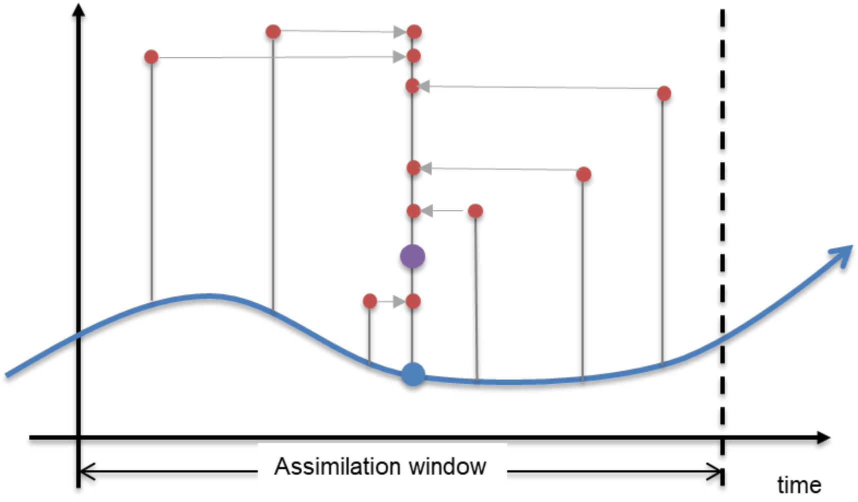
- 3DVar
- 3DFGAT
- 4DVar

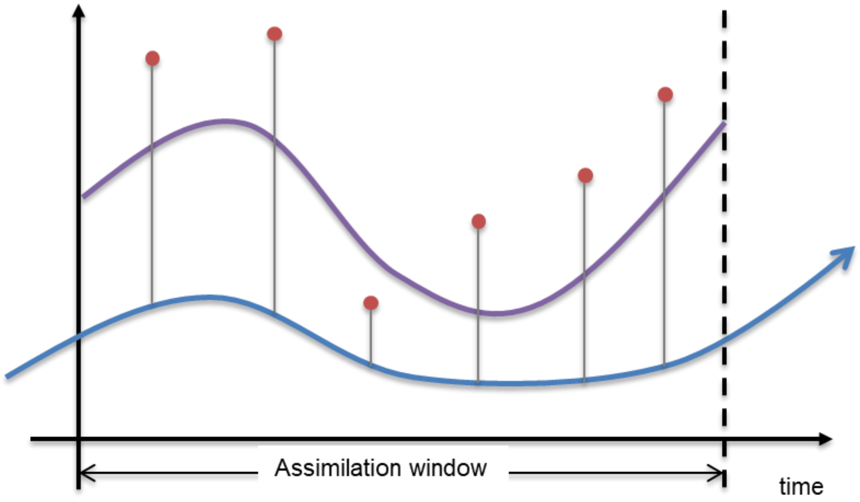
Observations

Analysis

Previous forecast







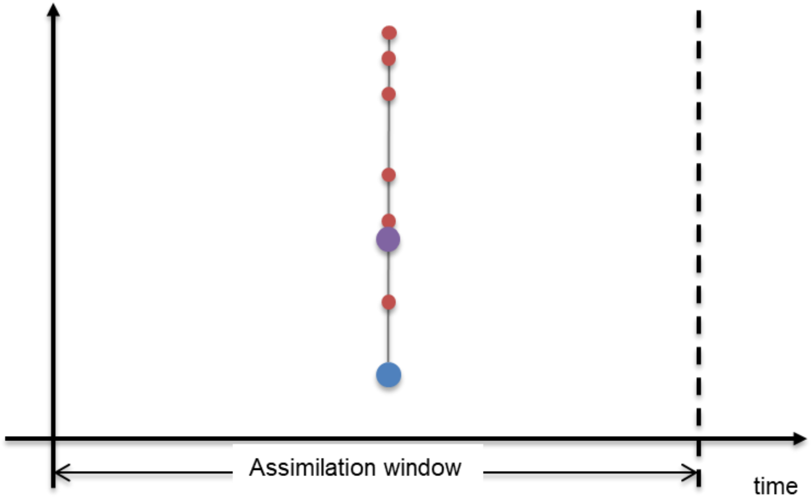
Data Assimilation Schemes within OOPS

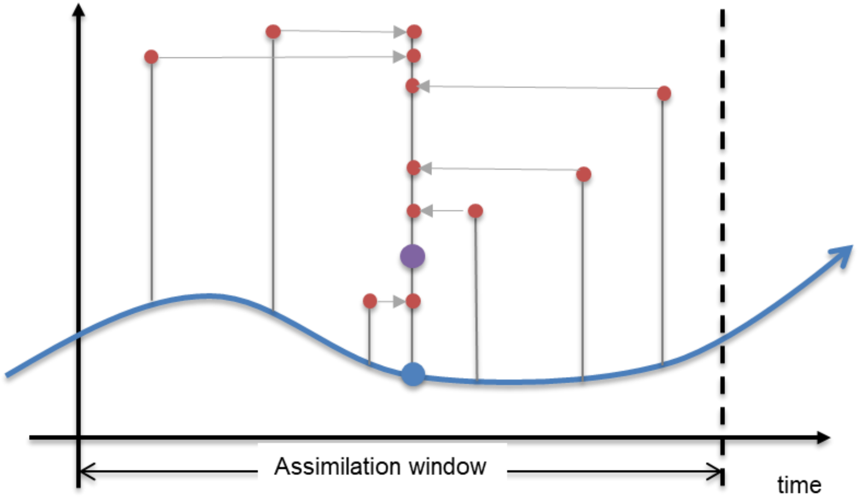
- 3DVar
- 3DFGAT
- 4DVar
- 4DEnsVar

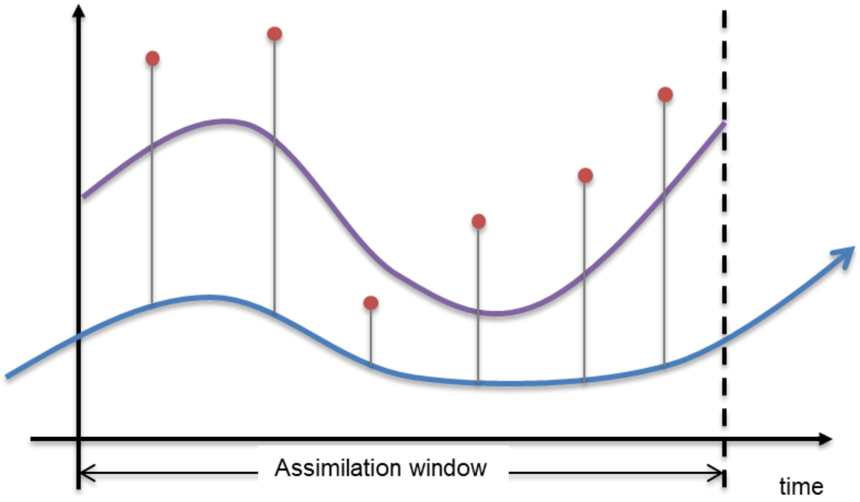
Observations

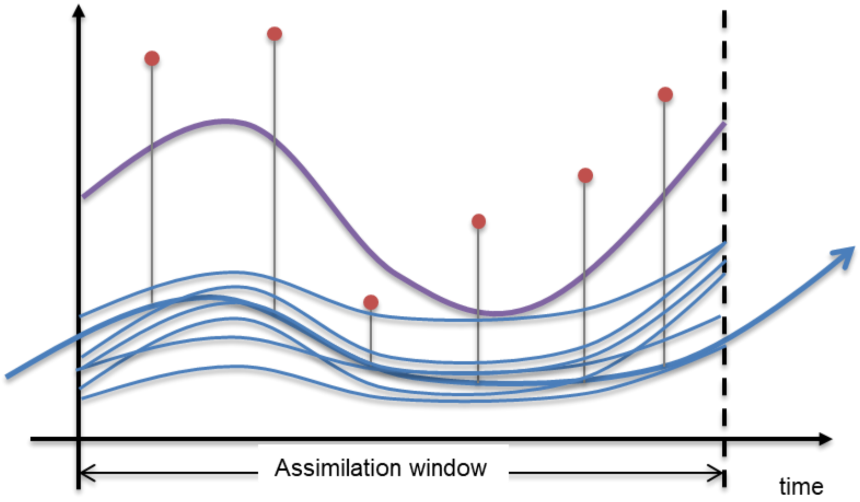
Analysis

Previous forecast









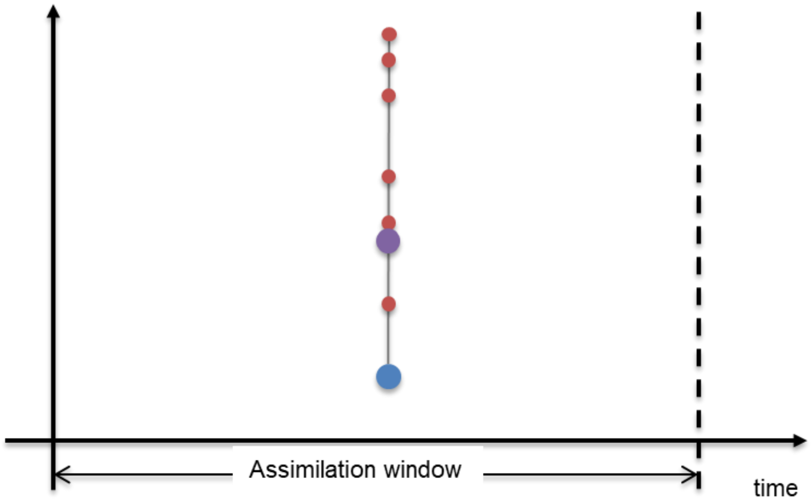
Data Assimilation Schemes within OOPS

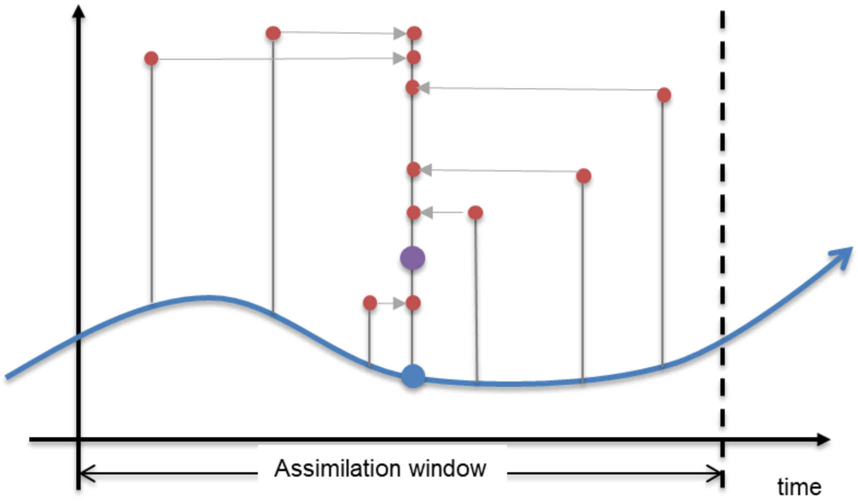
- 3DVar
- 3DFGAT
- 4DVar
- 4DEnsVar
- LETKF
- LGETKF

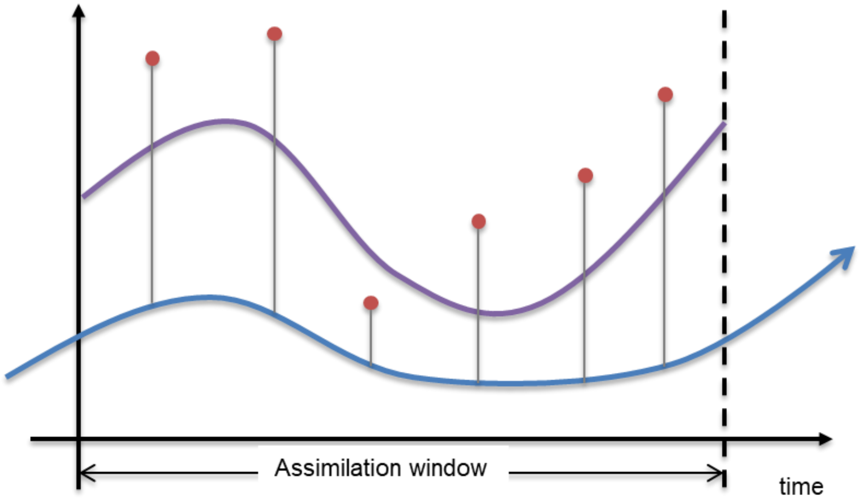
Observations

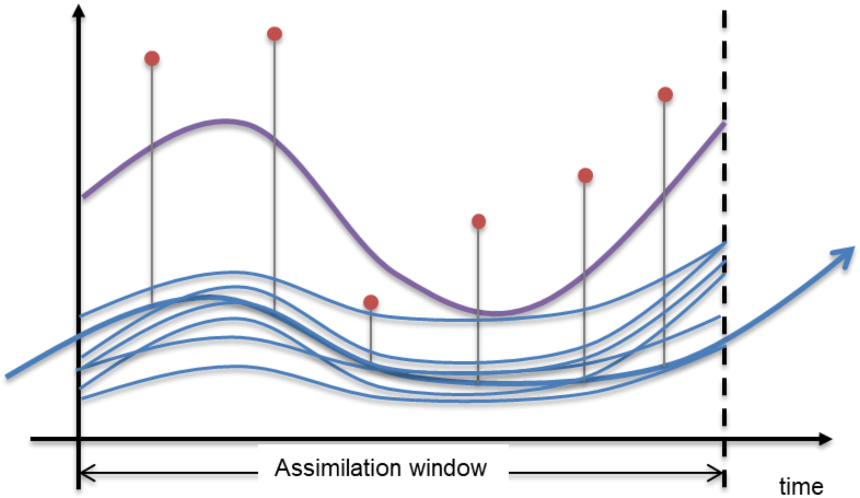
Analysis

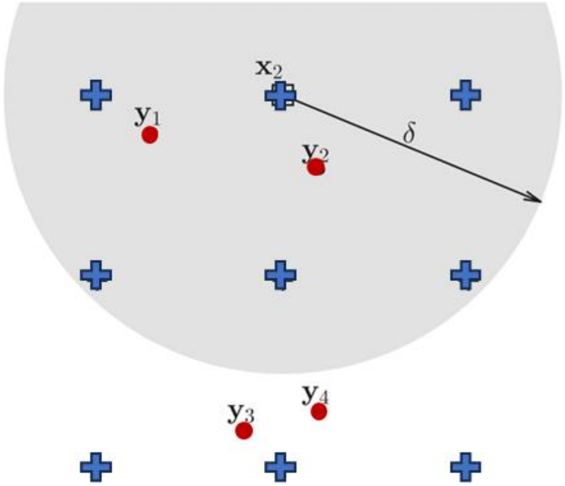
Previous forecast







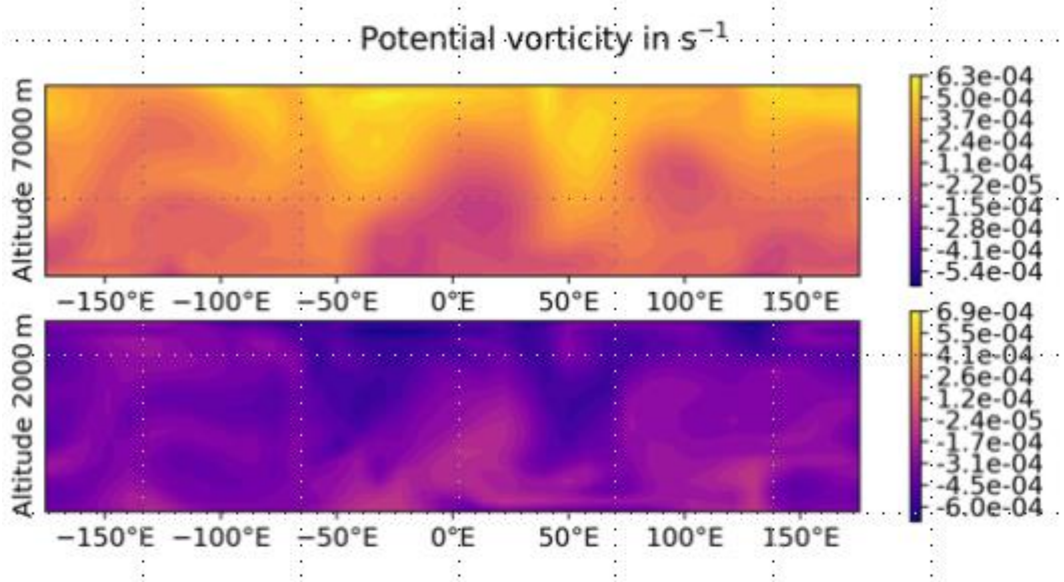




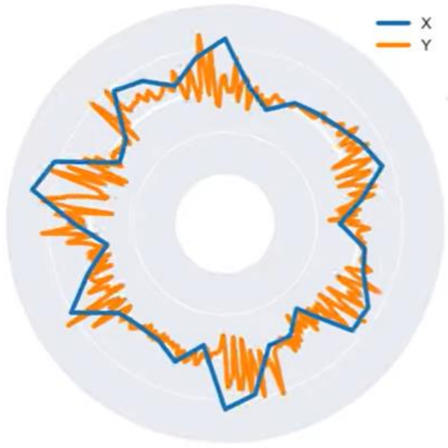
Toy models in OOPS

- Two toy models are available within oops:
  - Quasi Geostrophic model
  - Lorenz 96 model
- Used to test code changes and new developments quickly.
- Can be used to run research experiments.
- They also allow for extensive unit testing that is always run on oops pull requests that provide advance warning of any issues

Quasi Geostrophic Model



Lorenz 96 Model



Met Office

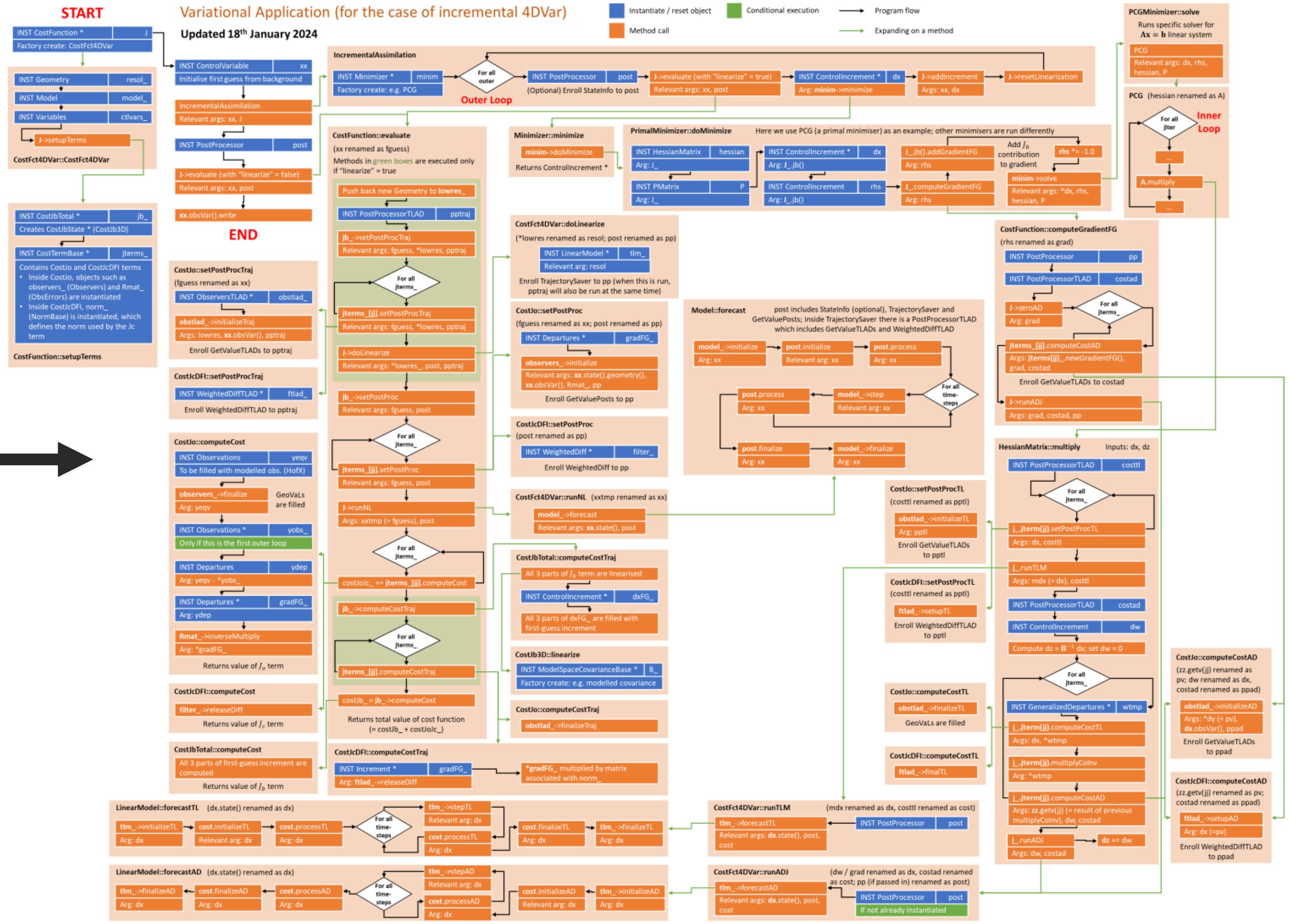
OOPS Applications

- Data assimilation runs are controlled by configuration files (more later) and run via the OOPS applications.
- Example applications include:

```
- HofX.h
- Variational.h
- Forecast.h
- EnsembleApplication.h
- GenEnsPertB.h
- …
```

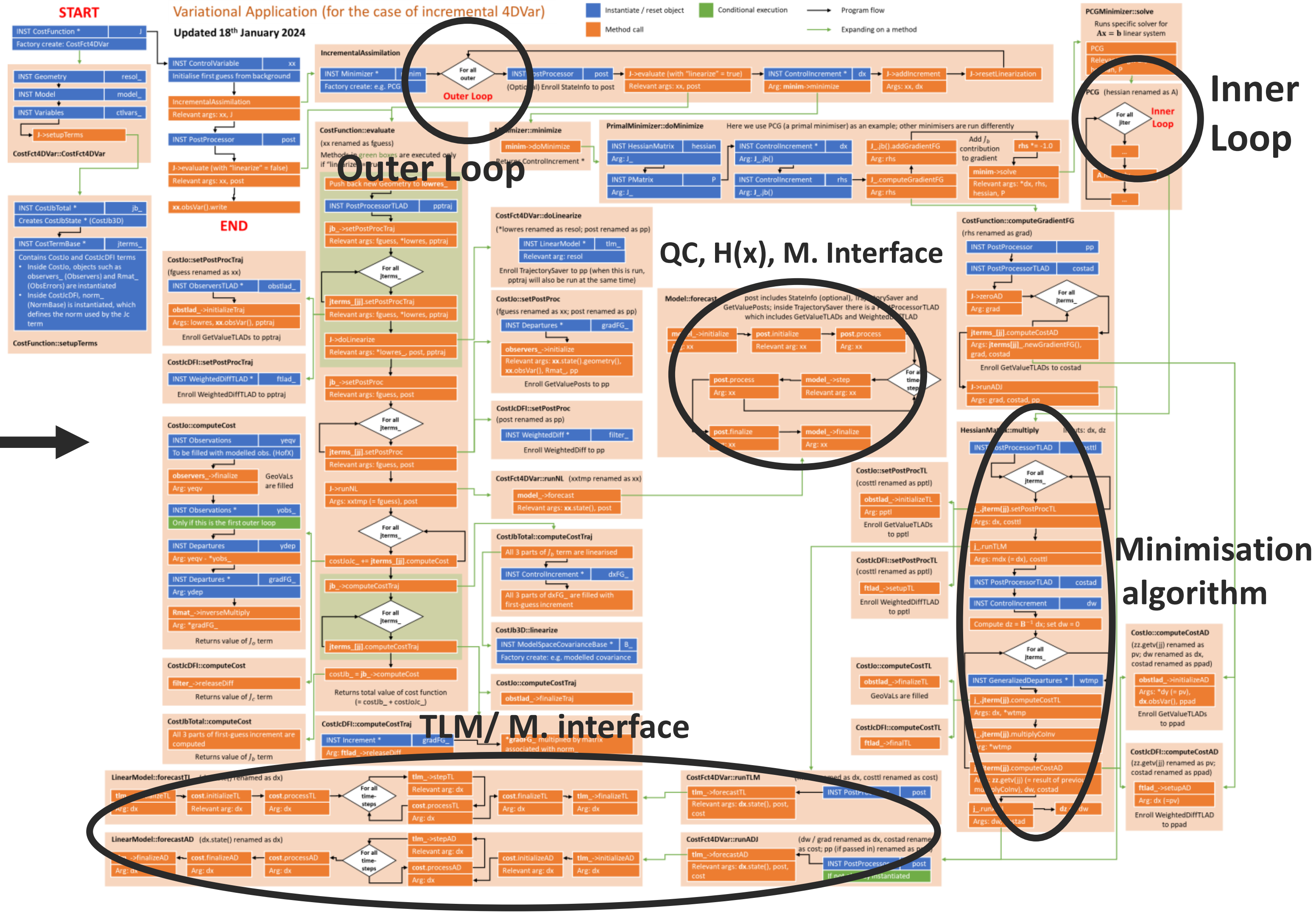
OOPS Applications

- Data assimilation runs are controlled by configuration files (more later) and run via the OOPS applications.
- Example applications include:
  - HofX.h
  - Variational.h
  - Forecast.h
  - EnsembleApplication.h
  - GenEnsPertB.h
  - …



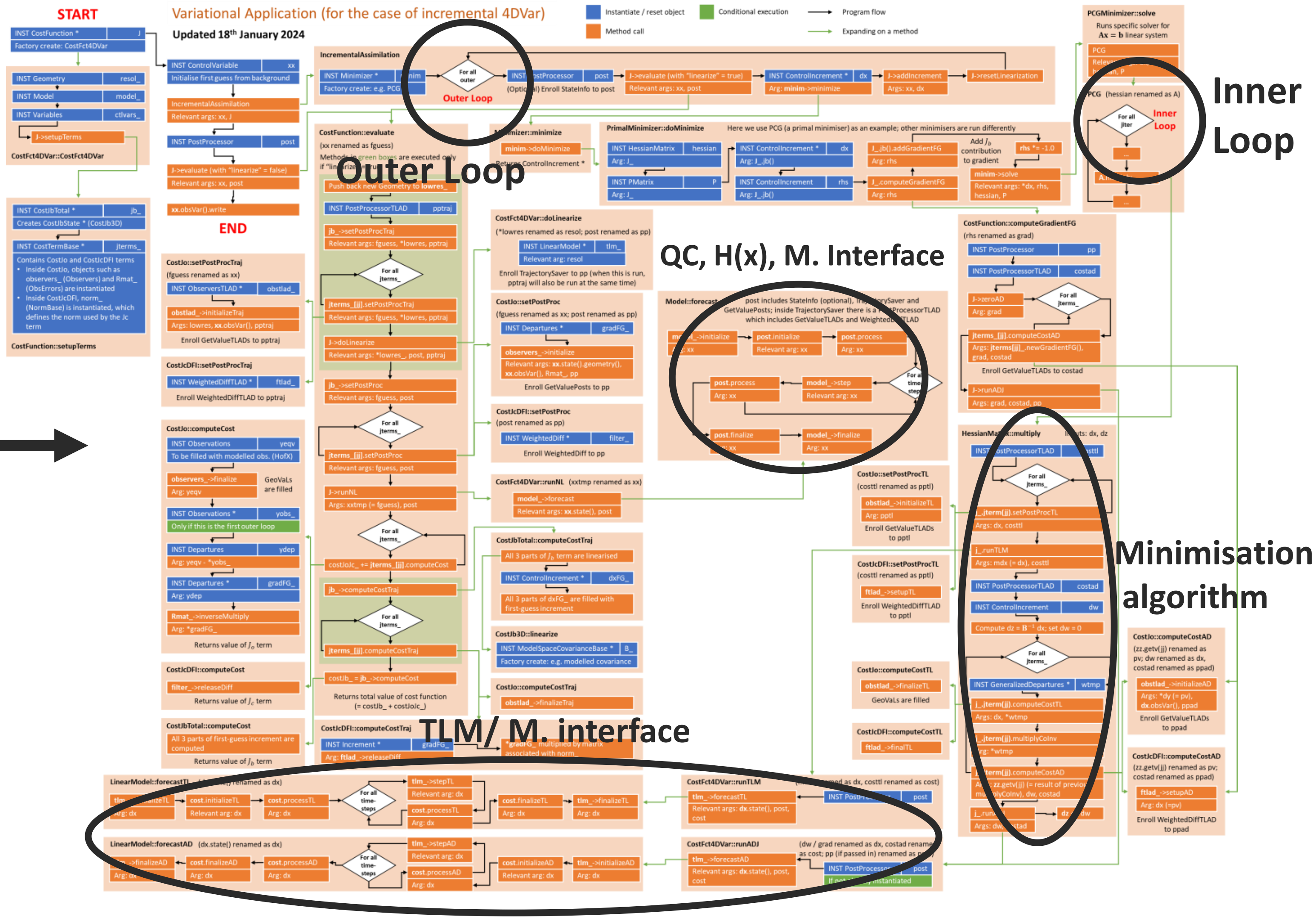
OOPS Applications

- Data assimilation runs are controlled by configuration files (more later) and run via the OOPS applications.
- Example applications include:
  - HofX.h
  - Variational.h
  - Forecast.h
  - EnsembleApplication.h
  - GenEnsPertB.h
  - …

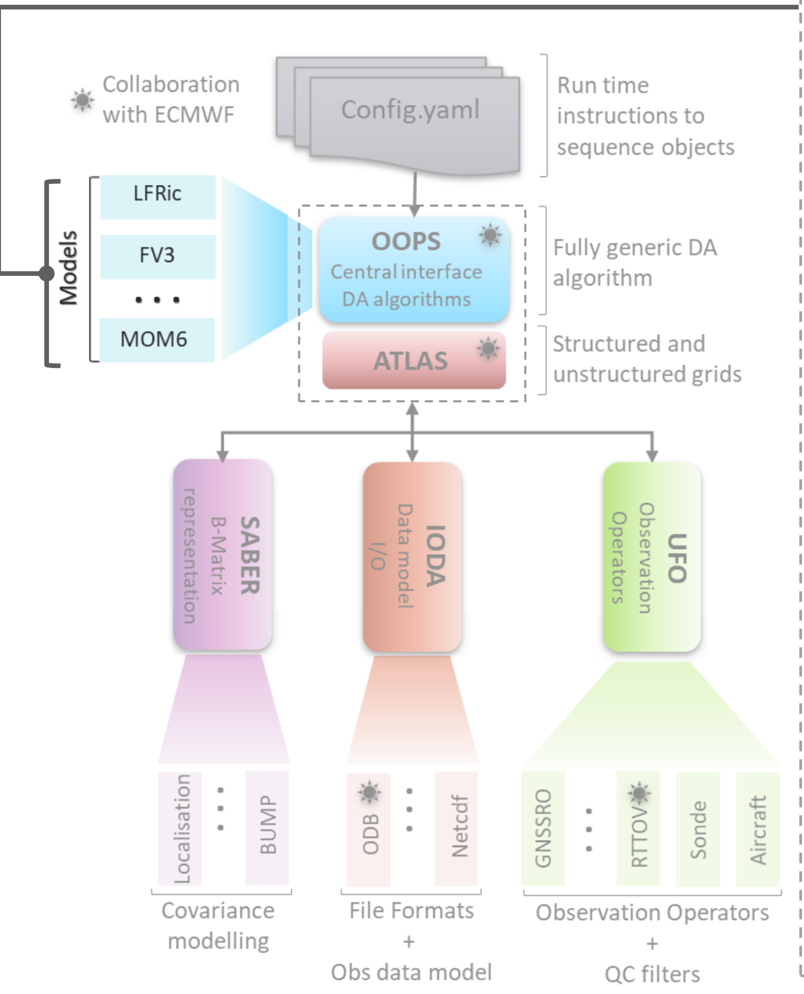


OOPS Applications

- Data assimilation runs are controlled by configuration files (more later) and run via the OOPS applications.
- Example applications include:
  - HofX.h
  - Variational.h →
  - Forecast.h
  - EnsembleApplication.h
  - GenEnsPertB.h
  - …
- Code can be run without understanding or recompiling/building code!

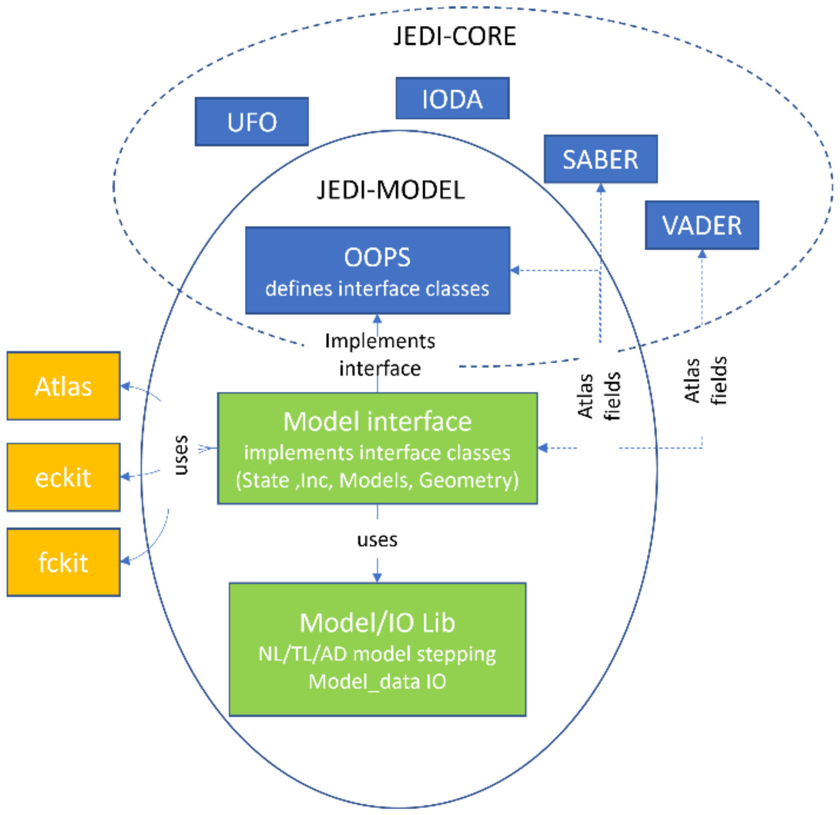


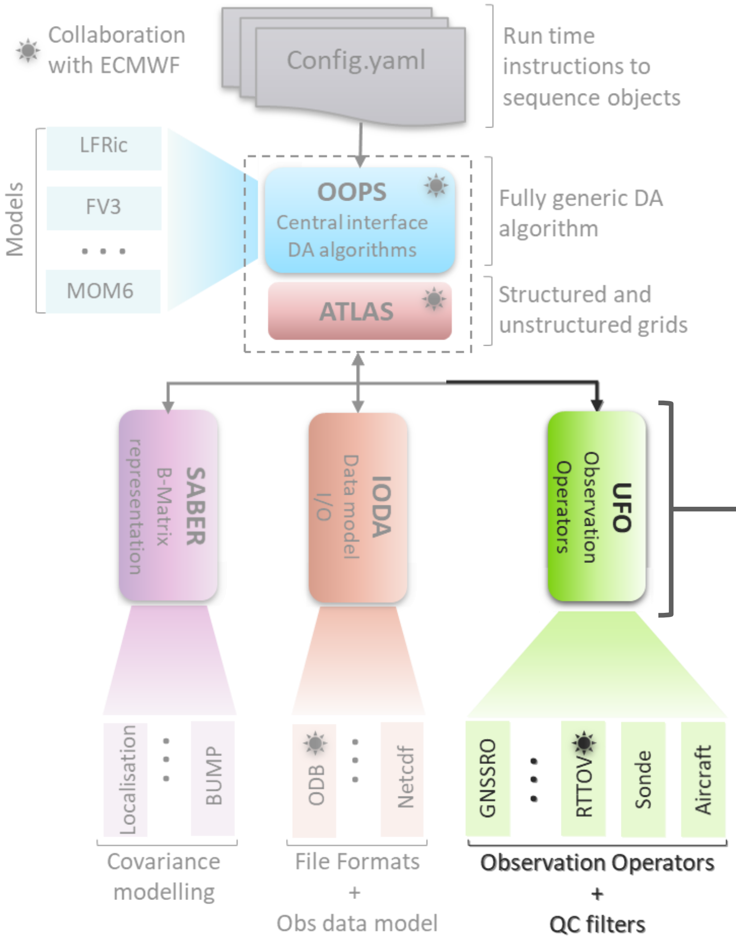
Model Interface



Model Interface

- Provide the link between the individual models and the agnostic JEDI code.
- Enables use of model values at various stages of the observation processing and data assimilation.
- Full NWP to toy models.
- Generic interface with one specific implementation per model:
  - UM (existing Met Office model)
  - LFRic (upcoming Met Office model)
  - FV3 (GFDL), …
  - Toy models
- Inter-changeable.





Unified Forward Operator

- Collection of forward (observation) operators:
  - Easy to explore differences between different versions
- Collection of abstract observation filters:
  - Called before and after the forward operators
  - Exposed to:
    - Observation values and metadata
    - Model values at observations locations

Different radiative transfer operators produce different results

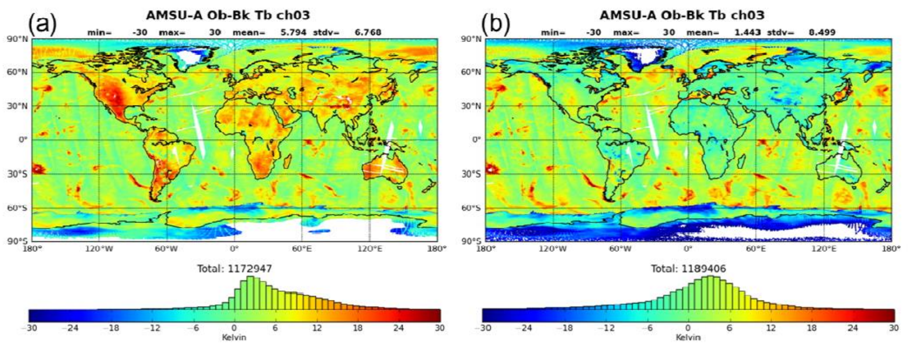
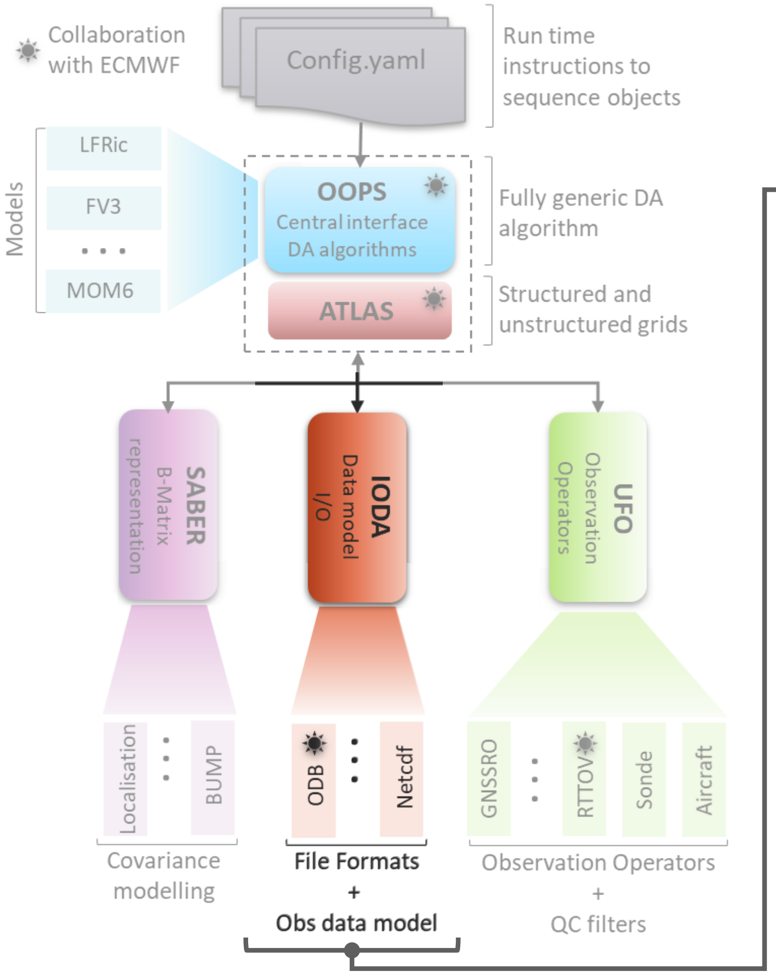


Figure 3: (a) AMSU-A channel 3 FG departures using (a) RTTOV and (b) CRTM at 00 UTC on 7 November 2012.

Example quality control filters:

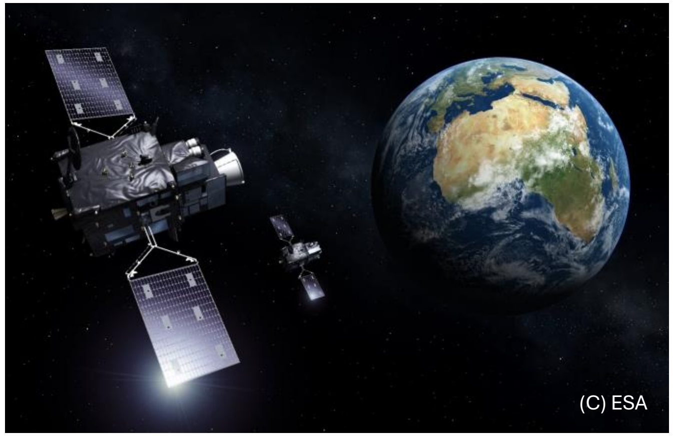
- Thinning
- Missing or bad data / blacklisted stations
- Out of physically plausible bounds / out of domain (e.g. ships over land)
- O-B ('observation minus background') checks
- Satellites: channel selection

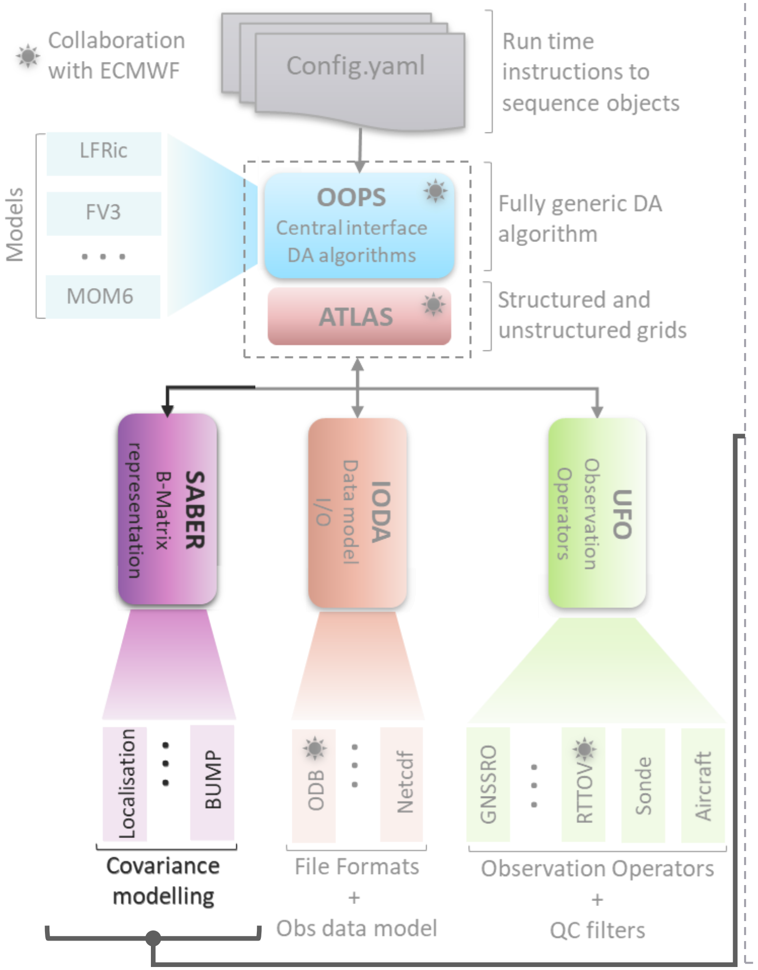
Pre-H(x)
Prior
Post-H(x)



Interface for Observation Data Access

- Observation data model.
- Provides the interfaces that bridge the external observation data to the components within JEDI that utilize those data, namely OOPS and UFO.
- Can read multiple different file formats (BUFR, netCDF, ODB)

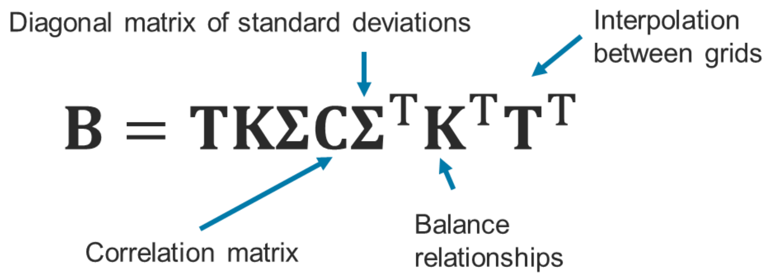




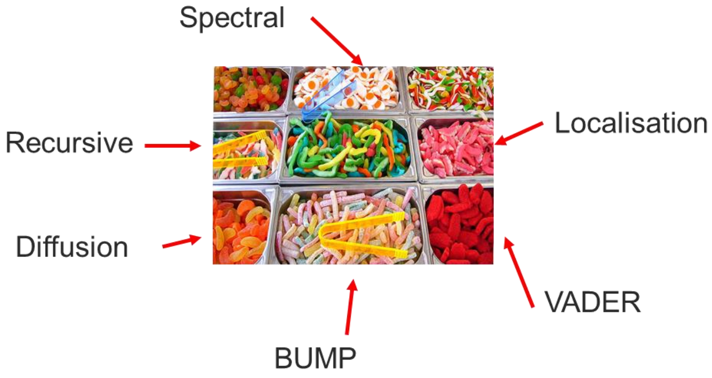
System Agnostic Background Error Representation

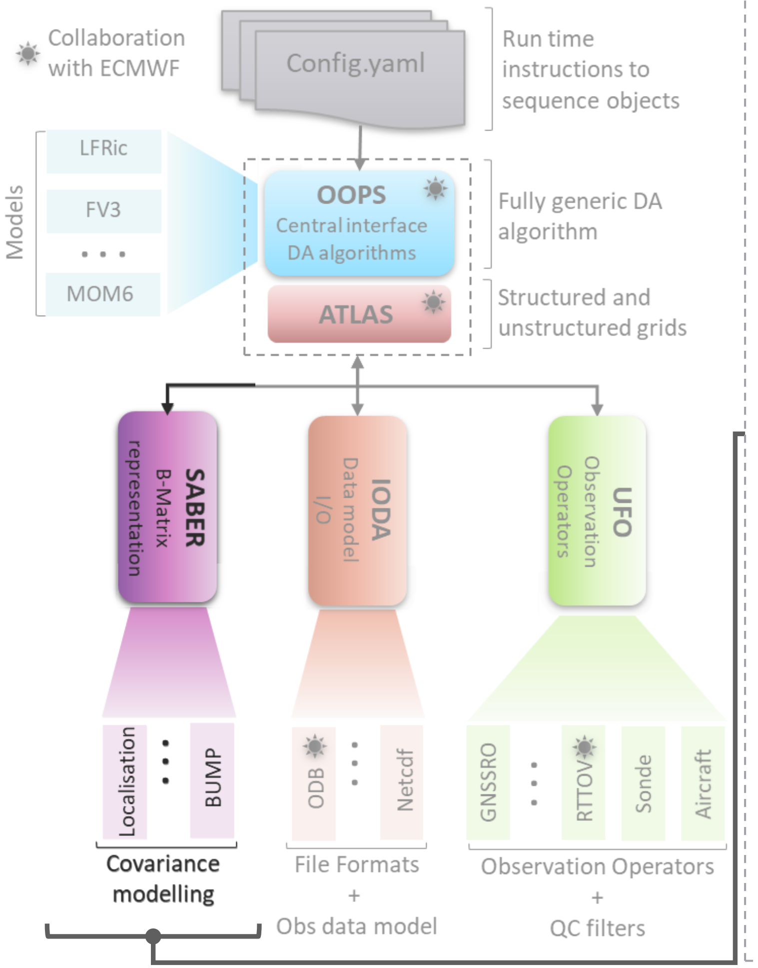
- It is the equivalent of UFO, but for dealing with background error covariance models (both static and ensemble-based versions)
- Model agnostic
- Flexibility: pick and mix approaches
- Can apply different localization schemes.

→ Implementing within SABER-block framework (abstraction of matrices/transform operations that constitute B)



$$\mathbf{B} = \mathbf{T}\mathbf{K}\boldsymbol{\Sigma}\mathbf{C}\boldsymbol{\Sigma}^T\mathbf{K}^T\mathbf{T}^T$$





System Agnostic Background Error Representation

- It is the equivalent of UFO, but for dealing with background error covariance models (both static and ensemble-based versions)
- Model agnostic
- Flexibility: pick and mix approaches
- Can apply different localization schemes.

→ Implementing within SABER-block framework (abstraction of matrices/transform operations that constitute B)

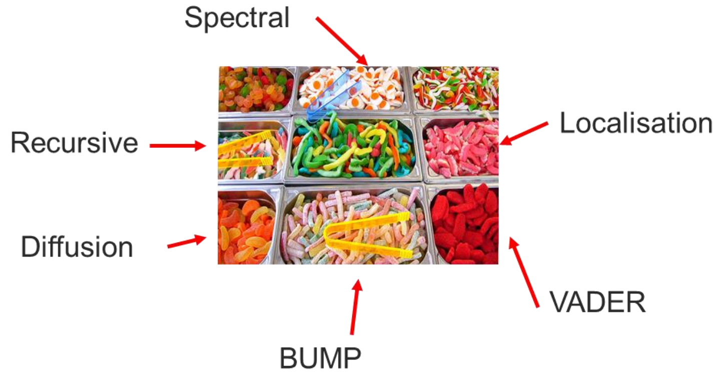
Diagonal matrix of standard deviations

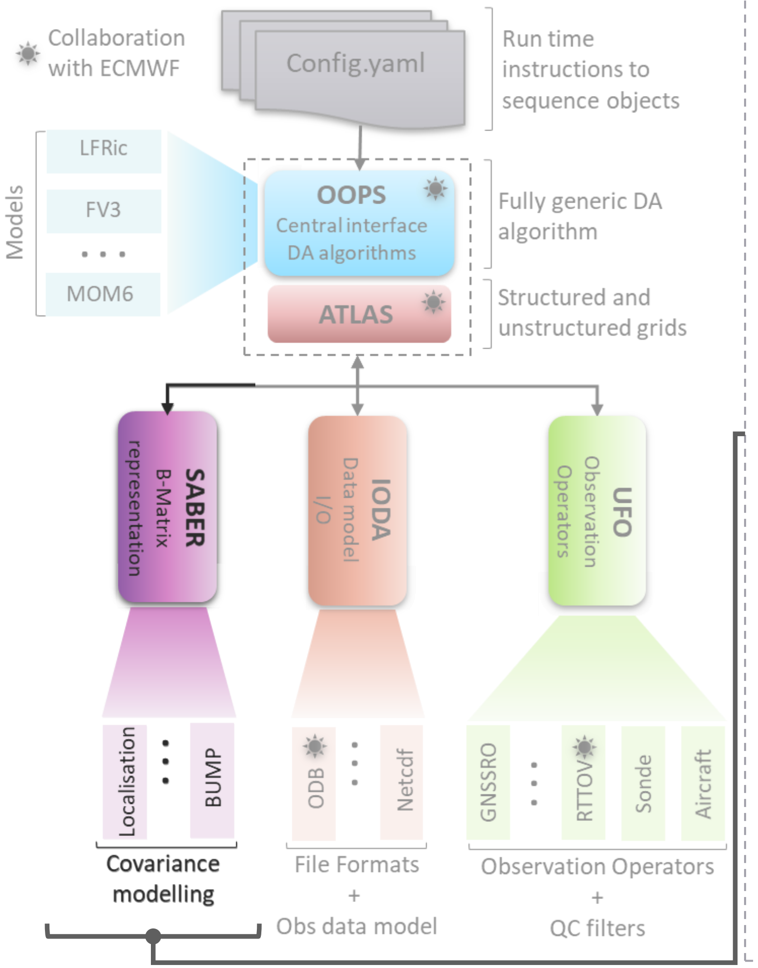
Interpolation between grids

$$\mathbf{B} = \mathbf{T}\mathbf{K}\boldsymbol{\Sigma}\mathbf{C}\boldsymbol{\Sigma}^{T}\mathbf{K}^{T}\mathbf{T}^{T}$$

Correlation matrix

Balance relationships





System Agnostic Background Error Representation

- It is the equivalent of UFO, but for dealing with background error covariance models (both static and ensemble-based versions)
- Model agnostic
- Flexibility: pick and mix approaches
- Can apply different localization schemes.

→ Implementing within SABER-block framework (abstraction of matrices/transform operations that constitute B)

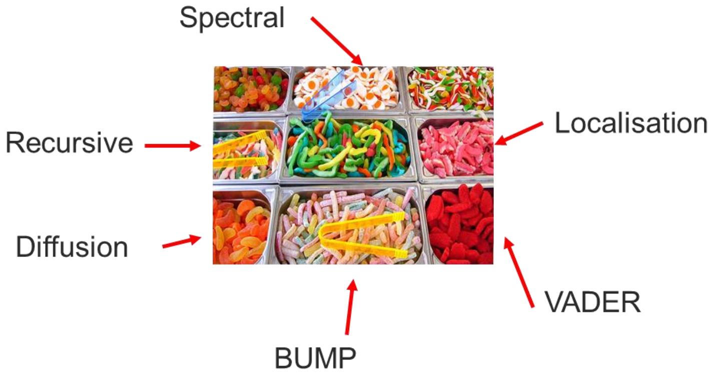
Diagonal matrix of standard deviations

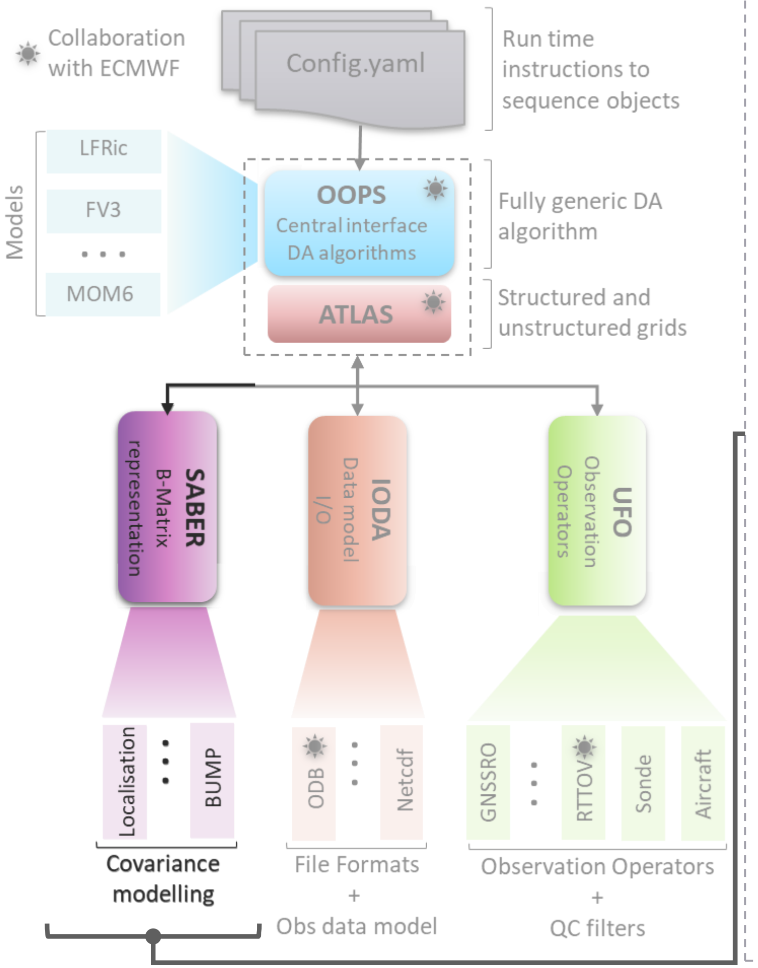
Interpolation between grids

$$\mathbf{B} = \mathbf{T}\mathbf{K}\mathbf{\Sigma}\mathbf{C}\mathbf{\Sigma}^{\mathbf{T}}\mathbf{K}^{\mathbf{T}}\mathbf{T}^{\mathbf{T}}$$

Correlation matrix

Balance relationships

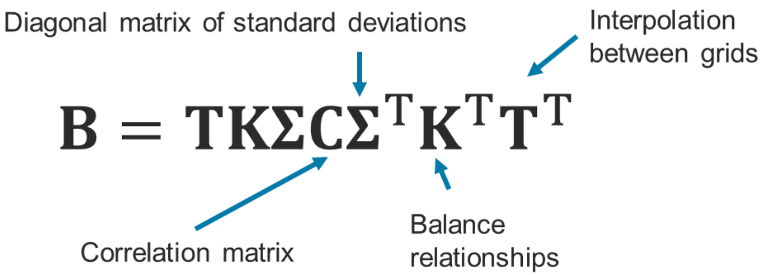


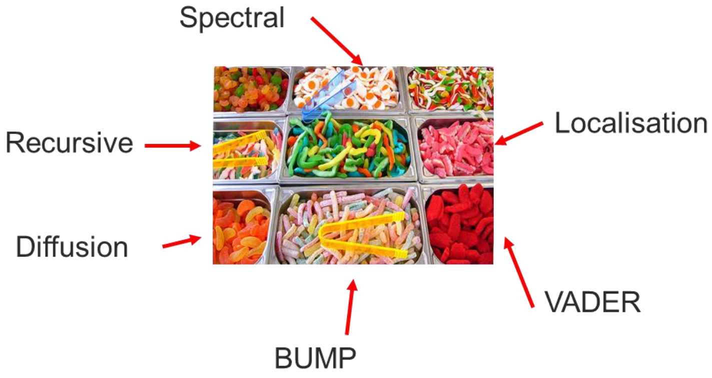


System Agnostic Background Error Representation

- It is the equivalent of UFO, but for dealing with background error covariance models (both static and ensemble-based versions)
- Model agnostic
- Flexibility: pick and mix approaches
- Can apply different localization schemes.

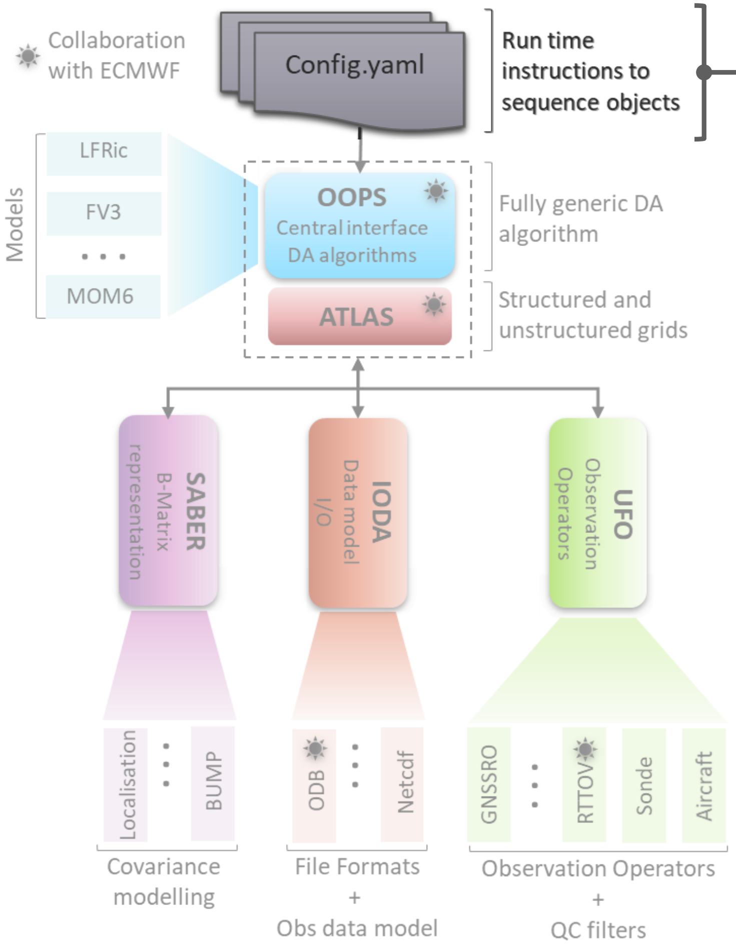
→ Implementing within SABER-block framework (abstraction of matrices/transform operations that constitute B)





- JEDI Overview
- JEDI Components
- JEDI Configuration
- Summary

YAML



JEDI Configuration Files

- Jedi is controlled by a flexible and modular text based configuration system.
- Configuration files provide scientific instruction to the code.
- Goal is to reduce the need for writing custom code when setting up and running JEDI in an operational model.
- Use YAML:
  - User friendly appearance
  - **Key-value** pair specification
  - **Indentation** defines sections
    - Tabs are forbidden (common source of errors)
  - yaml.org

```yaml
geometry:
  nx: 40
  ny: 20
  depths: [4500.0, 5500.0]
initial condition:
  date: 2010-01-01T00:00:00Z
  filename: Data/background.nc
model:
  name: QG
  tstep: PT1H
forecast length: PT12H
time window:
  begin: 2010-01-01T00:00:00Z
  length: PT12H
observations:
  observers:
  - obs space:
      ...
    obs operator: &obsop
      name: VertInterp
    obs pre filters:
    - ...
    obs prior filters:
    - ...
    obs post filters:
    - ...
```

YAML Example

Model, background & data assimilation

```
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```

Observations

```
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```

Minimisation

```
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

Met Office

YAML Example

Model, background & data assimilation


```
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```


Observations


```
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```


Minimisation


```
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

Met Office

YAML Example

Model, background & data assimilation

```
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```

Observations

```
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```

Minimisation

```
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

YAML Example

Model, background & data assimilation

```yaml
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```

Observations

```yaml
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```

Minimisation

```yaml
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

Met Office

YAML Example

Model, background & data assimilation

```yaml
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```

Observations

```yaml
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```

Minimisation

```yaml
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

Met Office

YAML Example

Model, background & data assimilation


```yaml
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```


Observations


```yaml
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```


Minimisation


```yaml
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

Met Office

YAML Example

Model, background & data assimilation

```
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```

Observations

```
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```

Minimisation

```
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

YAML Example

Model, background & data assimilation

```
cost function:
  model:
    name: QG
    tstep: PT15M
  cost type: 4D-Var
  window begin: 2010-01-01T00:00:00Z
  window length: PT24H
  analysis variables: [x]
  geometry:
    nx: 40
    ny: 20
    depths: [4500.0, 5500.0]
  background:
    date: 2010-01-01T00:00:00Z
    filename: Data/forecast.fc.2009-12-31T00:00:00Z.P1D.nc
  background error:
    covariance model: QgError
    horizontal_length_scale: 2.2e6
    maximum_condition_number: 1.0e6
    standard_deviation: 1.8e7
    vertical_length_scale: 15000.0
```

Observations

```
observations:
    observers:
    - obs error:
        covariance model: diagonal
      obs operator:
        obs type: Stream
      obs space:
        obsdatain:
          engine:
            obsfile: Data/truth.obs4d_24h.nc
        obsdataout:
          engine:
            obsfile: Data/4dvar_dripcg.obs4d_24h.nc
        obs type: Stream
```

Minimisation

```
variational:
  minimizer:
    algorithm: DRIPCG
  iterations:
  - gradient norm reduction: 1.0e-10
    ninner: 10
    geometry:
      nx: 40
      ny: 20
      depths: [4500.0, 5500.0]
    linear model:
      trajectory:
        tstep: PT30M
      tstep: PT1H
      variable change: Identity
      name: QgTLM
    diagnostics:
      departures: ombg
```

Met Office

- JEDI Overview
- JEDI Components
- JEDI Configuration
- Summary

Summary

JEDI

JEDI is a collaborative development between JCSDA partners to develop a unified data assimilation system:

- → From toy models to Earth system coupled models
- → Unified observation (forward) operators (UFO)
- → For research and operations
- → Share as much as possible without imposing one approach (one system, multiple methodologies/configurations)

Motivation

Changes in HPC landscape	Fully exploit future generations of supercomputers Scalability, efficient I/O Memory and novel parallelism in era of large (1000s) ensemble DA
Technical	Increased modularity (more object-oriented capabilities) Expanded range of platforms (traditional HPC, cloud, laptop, etc)
More complex science	Flow-dependent QC, ensembles Strongly-coupled earth system DA, etc.
Future Applications	Nowcasting, cloud analysis, city-scale DA, composition DA, multi-models, etc.
Human	Current OPS/VAR too complex for wide-spread use Need to encourage wider collaboration (academic users, those working on other models)