

OpenDA bias aware model

Nils van Velzen

October 28, 2013

1 Bias aware modelling

Data assimilation methods normally assume no bias in the model and the observation errors. In real life this is unfortunately not always the case. There are some mathematical methods available to detect and estimate structural differences between observations and mode (bias). OpenDA contains a wrapper model that enables perform experiments to estimate and the bias between model and observations.

2 Algorithm

OpenDA uses state augmentation to estimate the bias. The method is explained in more details in [1] and .. A model is defined in openDA according to

$$x^{k+1} = M(x^k, u^k, p, w^k) \quad (1)$$

With the model state x , forcings y , parameters p and noise w . The interpolation operation to compare the model predictions to the observations y^k is defined by

$$Hx^k \quad (2)$$

We assume there is a bias b between model prediction Hx and observations y . If we would know this bias we can correct the observations y with b before assimilating.

$$\begin{bmatrix} x^{k+1} \\ b^{k+1} \end{bmatrix} = \begin{bmatrix} M(x^k, u^k, p, w^k) \\ b^k + n^k \end{bmatrix} \quad (3)$$

The interpolation operation for this bias correcting model is

$$Hx^k - b^k \tag{4}$$

The bias aware model in OpenDA is a generic wrapper model that extends an arbitrary OpenDA model, implementing Equations 1 and 2 into a model that implements Equations 3 and 4.

3 configuration

The model configuration contains of two parts:

1. Definition of the child model (xml-tag `stochModelFactory`)
2. Definition of the bias model (xml-tag `state`)

The format of the `stochModelFactory` is exactly the same as in the definition of a stochastic model factory in the main OpenDA configuration file. It contains:

- the attribute `className` (mandatory), specifying the implementation of the model factory
- the tag `workingDirectory` (mandatory), specifying the main directory of the model configuration
- the tag `configFile` (mandatory), file containing the model configuration

The second part with tag `state` defines the augmented state that is used to model the bias. This tag contains:

- the attribute `maxSize` (optional), the size of the augmented state. This attribute can only be left out when all observations are individually specified using the `observation` attributes.
- the attribute `localization` (optional, default="true") If set "true" we assume all elements in the augmented state to be non correlated. If the filter uses localisation, only the single matching observation is used to update each element of the augmented state. If set to "false" no localization is used, all localization weights of the augmented state are set to one.

- the attribute `standard_deviation`. The standard deviation of the random walk noise for a one day period.
- the attribute `checkObservationID` (optional, default="true"). Match elements in the augmented state using the ID of the observations. If set to "false" the algorithm expects that the i-th observation at each assimilation step corresponds to the i-th element of the augmented state.
- the tag `observation` (optional, repetitive). Specification of biases corresponding to individual observations. This attribute has the following attributes:
 - `id` (mandatory) name of the observation device/location
 - `standard_deviation` (mandatory) standard deviation of the random walk noise for a one day period

It is sometimes not known in advance what the id's are of the observations. A trick is to perform a small run, not individually specifying the observations. In the message file you will then find the id's of the observations when they are assigned to an element of the augmented state for the first time.

A typical configuration where all observations have the same bias uncertainty looks like:

```
<BiasAwareModelConfig>
<stochModelFactory
className="org.openda.models.lorenz.LorenzStochModelFactory">
<workingDirectory>.</workingDirectory>
<configFile></configFile>
</stochModelFactory>
<state standard_deviation="3.0"
maxSize="10"  localization="false"  checkObservationID="true">
</state>
</BiasAwareModelConfig>
```

A configuration that individually configures the biases looks like:

```
<?xml version="1.0" encoding="UTF-8"?>
<BiasAwareModelConfig>
<stochModelFactory
```

```
className="org.openda.models.lorenz.LorenzStochModelFactory">
<workingDirectory>.</workingDirectory>
<configFile></configFile>
</stochModelFactory>
<state localization="false" checkObservationID="true">
<observation id="point1.waterlevel"
    standard_deviation="0.01"></observation>
<observation id="point2.waterlevel"
    standard_deviation="0.02"></observation>
<observation id="point3.waterlevel"
    standard_deviation="0.03"></observation>
<observation id="point4.waterlevel"
    standard_deviation="0.04"></observation>
<observation id="point5.waterlevel"
    standard_deviation="0.05"></observation>
</state>
</BiasAwareModelConfig>
```

References

- [1] Jean-Philippe Drécourt, Henrik Madsen, Dan Rosbjerg, *Advances in Water Resources* 29 (2006) 707-718.