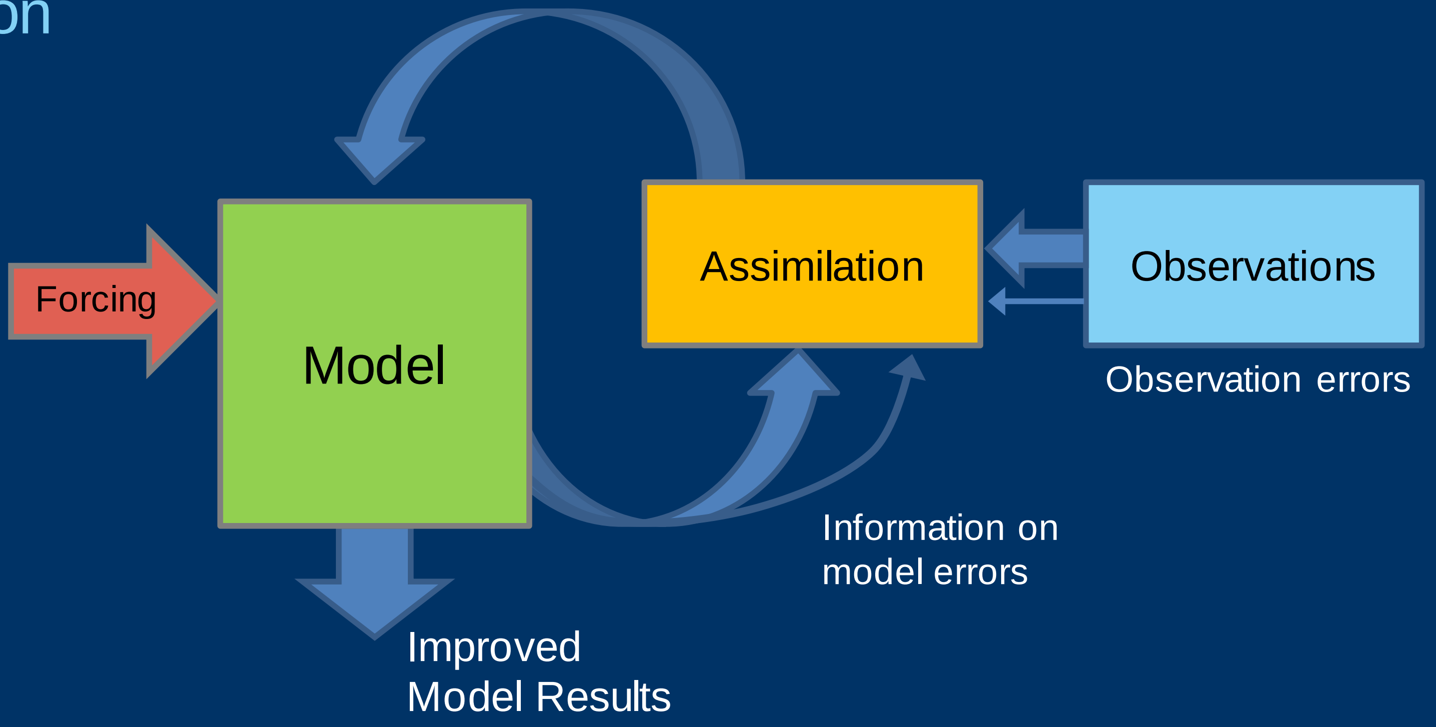
Running EFDC example with Data Assimilation

Data Assimilation



- The data assimilation algorithm uses:
  - Available observations and their uncertainty (observation errors).
  - Model predictions and their uncertainty (model errors).
- The algorithm applies an optimal interpolation scheme based on the available information to improve model results.

Data Assimilation scheme

- The Data Assimilation (DA) scheme is an Optimal Interpolation [1] scheme that uses only spatial information about the errors of the EFDC physical model.
- Model error information is represented by a description of the spatial distribution of the covariance functions of such errors.
- Given that the covariance function of the model itself is unknown, an approximation is obtained by means of assumptions on the covariance structure.
- Stated differently, the DA scheme is a linear state estimator with stationary covariance operators, where the covariance matrices are predetermined in advanced (through inference/assumptions of their covariance structure) instead of being computed from the dynamical equation.
- In the assimilation cycle of the DA scheme, the updated model predictions, or analysis for short, are computed using the **Best Linear Unbiased Estimator** (BLUE) [2].

BLUE (Best Linear Unbiased Estimate)

$$T^a = \hat{T} + PH^\top(HPH^\top + R)^{-1}(T^o - H\hat{T})$$

Update(t) = Prediction(t-1) + Gain x [Prediction(t-1) – Observation(t-1)]

Model estimate given information at t-1 (Prior)

Prediction/Observation Difference

Gain: Map observations onto numerical grid
Account for observation quality
Account for model error (structural error)

BLUE equation

$$P \in \mathbb{R}^{N\times N}$$

State error covariance matrix

$$T^a = \hat{T} + PH^\top (HPH^\top + R)^{-1} (T^o - H\hat{T})$$

$$T^a = \begin{bmatrix} T^a_1 \\ \vdots \\ T^a_N \end{bmatrix}$$

Optimally interpolated temperatures

$$\hat{T} = \begin{bmatrix} \hat{T}^a_1 \\ \vdots \\ \hat{T}^a_N \end{bmatrix}$$

Model predicted temperatures

$$H \in \mathbb{R}^{M\times N}$$

Observation/model mapping matrix

$$T^o = \begin{bmatrix} T^o_1 \\ \vdots \\ T^o_M \end{bmatrix}$$

Observed temperatures

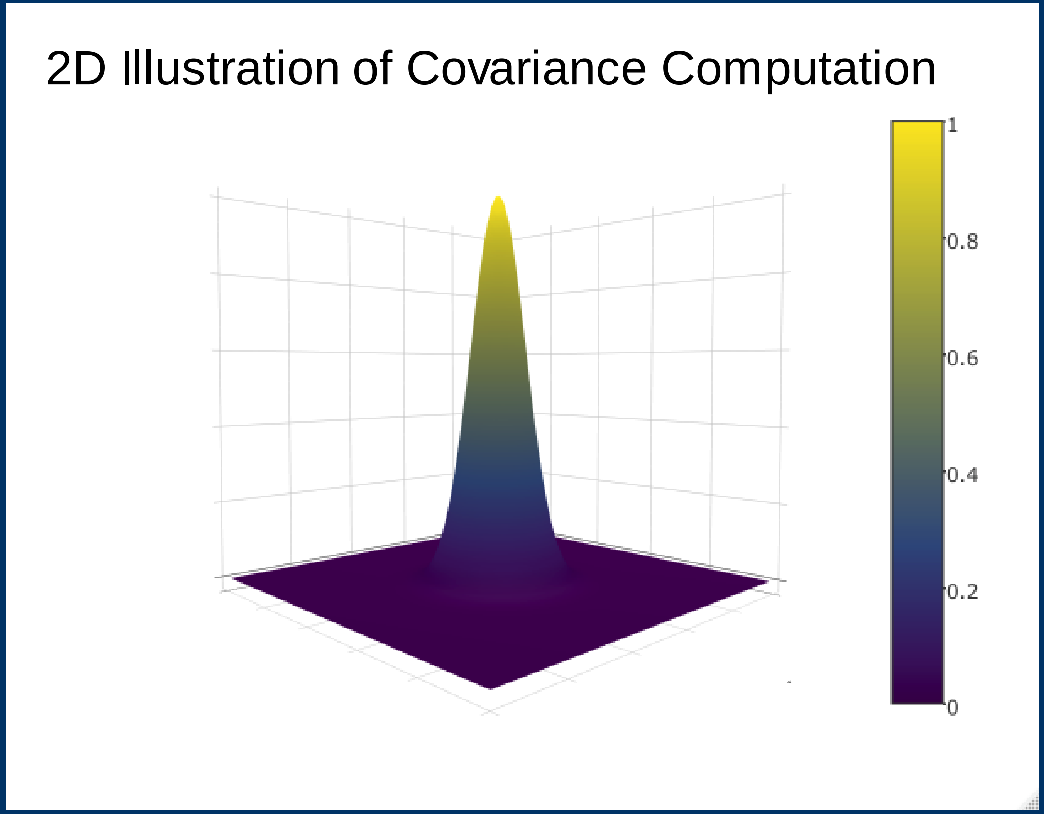
$$R = \begin{pmatrix} \sigma(T^o_1) & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma(T^o_M) \end{pmatrix}$$

Observation error covariance matrix

BLUE Equation – State Covariance Error

$$P = Cov(T_n, T_m) = a \exp\left\{ - \left[\left(\frac{i_n - i_m}{R_1} \right)^2 + \left(\frac{j_n - j_m}{R_2} \right)^2 + \left(\frac{k_n - k_m}{R_3} \right)^2 \right] \right\}$$

Tuning parameters



- The Gaussian covariance function is based on the Euclidean distances across pairs (*n*,*m*) of mesh points.
- The figure illustrates how the covariance varies when applied to a 2D mesh (eliminating third term above) with tuning parameters a, R_1, R_2 equal to 1.

Data Assimilation scheme 3D

- BLUE 2D on the surface
- Depth Projection using Ekman Theory
  - An additional stress will produce Ekman transport
  - Corrections to subsurface velocities are estimated using

$$\delta u(z) = e^{(-z/D_e)}\left[\delta u_s \cos(-z/D_e) - \delta v_s \sin(-z/D_e)\right]$$

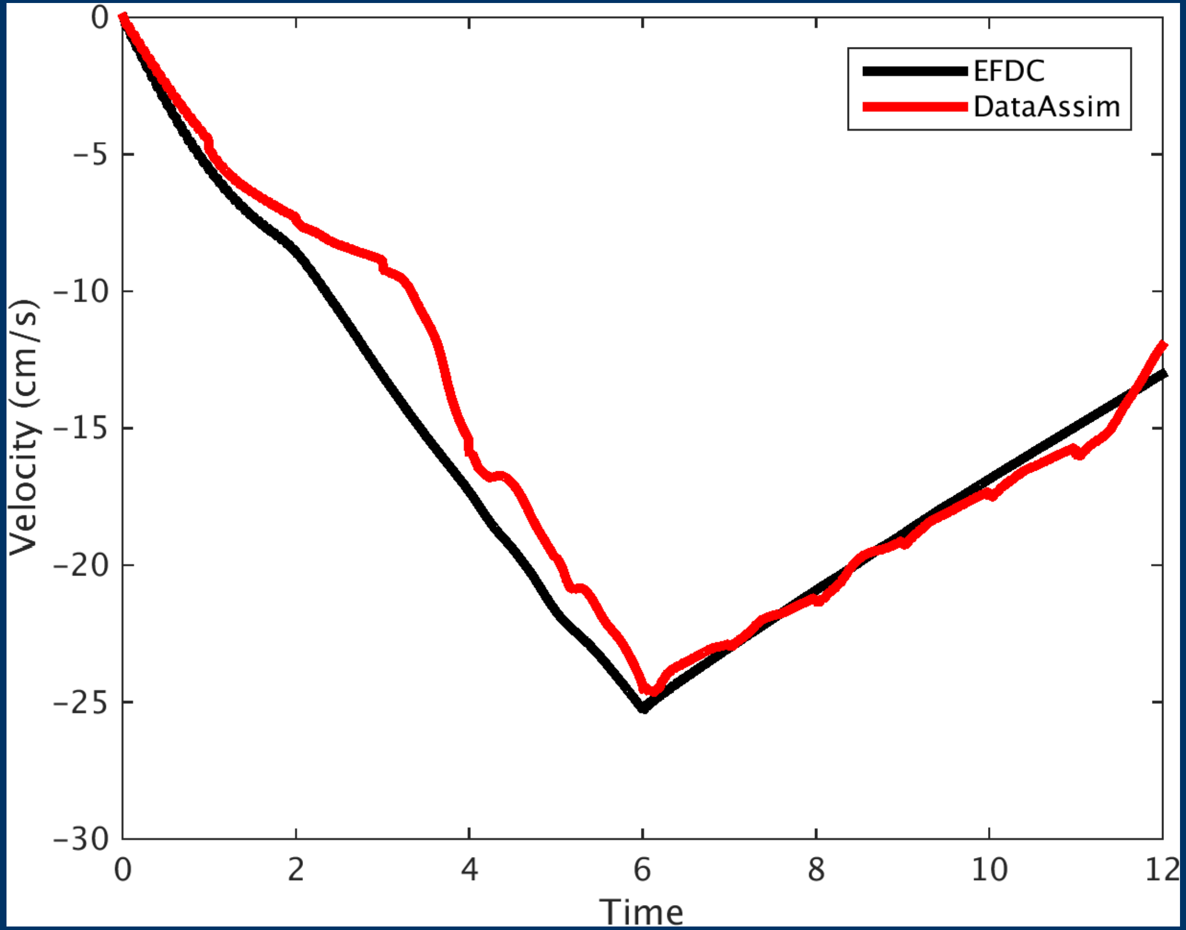
$$\delta v(z) = e^{(-z/D_e)}\left[\delta v_s \sin(-z/D_e) + \delta v_s \cos(-z/D_e)\right]$$

- where $D_e = \sqrt{2\mathrm{A}_v / f}$ is the Ekman depth

EFDC – Example Data assimilation scheme

- Data assimilation requires Blas and Lapack linear algebra libraries (included in Vagrantfile)
- Data assimilation demonstration implemented for a simple harbour example
  - $ cd /vagrant/SampleModels/Simple_DA_example
  - $ cp /vagrant/Src/EFDC .
  - $ ./EFDC
- Configured to assimilate pseudo-generated surface velocities every hour
  - Pseudo-data generated by running model with wind stress activated
  - Assimilated into model with no wind forcing (objective is to use DA to "correct" wind stress term)

DA example performance scheme



Assimilating surface velocities hourly nudges the modeltowards the "correct" (black line) solution

EFDC – Example Data assimilation scheme

- Input file DA.INP defines configurations to activate data assimilation

```
## Data assimilation input file
## Created by Fearghal O'Donncha (3rd August 2017)
## feardonn@ie.ibm.com

## File contains configuration information for
## assimilation of HFR data into EFDC
## Configured for case study application to Chesapeake Bay
## File describes an optional flag to switch on/off DA (IDA_FLAG)
## The spatial extents or domain to which DA applied
## The frequency of data assimilation (in hours)
## Data assimilation parameters (PMatrix_R1, PMatrix_R2, PMatrix_R3)
IDA_FLAG: 1   ! Set to 1 to activate DA, otherwise 0
# Create the extents of the data assimilation grid for DA
# (i.e. for HFR the corners of the rectangular grid
IBEG_DA, 3     #  Data assimilation applied within
IEND_DA, 13    #  this rectangular domain
JBEG_DA, 4     #  and all HFR observations
JEND_DA, 53    #  within this domain integrated
NDAPOINTS: 550  #  NUMBER OF POINTS TO ASSIMILATE
DA_FREQ(HOURS): 1  # How frequent to check for observation data
## DATA Assimilation Tunable parameters
PMatrix_R1: 3.0     # Extent influence east-west of covariance matrix
PMatrix_R2: 3.0     # Extent influence north-south of covariance matrix
PMatrix_A:  3.0     # Magnitude of impact of covariance matrix computation
EKPROJ: 1           # 0/1 flag which dictates if velocities projected into
                    # using empirical relationships (Ekman)
```

- IDA_FLAG 0/1 defines whether data assimilation scheme implemented (or if DA.INP file not present)
- Defines rectangular extents of the data assimilation scheme – points outside this region not ingested
- Defines number of points to assimilate
- Defines data assimilation scheme tuning coefficients: R1, R2 and A
- Flag to dictate whether Ekman Projection applied

Extending DA to other examples

- Assimilation scheme is configured to make compatible with external schemes (e.g. libraries in Python, R, etc.)
- Hence, assimilation acts on input and output files
- At user defined intervals:
  - EFDC state is written to file EFDC_state.csv
  - Observations in files: Observations/bservations_YYYY-MM-DD-HHMM.csv
  - Updated state read back into EFDC model from file BLUE.csv

Extending DA to other examples - Considerations

- DA.INP defines data assimilation configurations mapped to EFDC grid coordinates
- Observation data needs to be mapped to the same grid
  - For Codar HFR data a sample Python script that does mapping online is included (CoordRecon.py)
  - This script also does reconciliation from multiple domains to a single file
  - Data assimilation implemented in serial on a single domain with shared memory optimisations of linear algebra processes