EXERCISE 9: Forecast Cycle Observation Sensitivities

Like the forecast cycle observation impacts of Exercise 8, computation of the observation sensitivities for the forecast cycle involves multiple steps. The initial set-up steps are the same as those in Exercise 8, and the only difference is in steps 5 and 8.

Case 1: Measuring observation sensitivities using a verifying analysis

Steps 1 through 4 and step 7 are the same as in Exercise 8, so if you have done these already there is no need to do them again. You will need to copy the NetCDF files from the **WC13/RBL4DVAR_forecast_impact** subdirectories **FCSTAT**, **FCSTA** and **FCSTB** into the corresponding subdirectories of **WC13/RBL4DVAR_forecast_sensitivity** though.

The difference between the forecast cycle observation impact calculations of Exercise 8 and the forecast cycle observation sensitivity calculation considered here is the way that δe is computed. As described in Lecture 5, we can also express δe to 3rd-order as:

$$\delta e = \boldsymbol{d}^T (\partial \boldsymbol{\mathcal{K}} / \partial \boldsymbol{y})^T \boldsymbol{M}_b^T \left[\boldsymbol{M}_g^T \boldsymbol{C}\left(\boldsymbol{x}_g - \boldsymbol{x}_a\right) + \boldsymbol{M}_r^T \boldsymbol{C}(\boldsymbol{x}_r - \boldsymbol{x}_a)\right] \quad (1)$$

where $(\partial \boldsymbol{\mathcal{K}} / \partial \boldsymbol{y})^T$ represents the adjoint of the entire ROMS 4D-Var algorithm.

__**Step 5**__:
Go to the subdirectory **WC13/RBL4DVAR_forecast_sensitivity** and compile the forecast observation sensitivity driver using **build_roms.csh** using the following cpp options:

```
  setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRBL4DVAR_FCT_SENSITIVITY"
  setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DAD_IMPULSE"
 #setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DOBS_IMPACT"
 #setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DOBS_SPACE"
  setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRPCG"
```

The adjoint 4D-Var algorithm $(\partial \boldsymbol{\mathcal{K}} / \partial \boldsymbol{y})^T$ is selected by undefining **OBS_IMPACT**.

Now run the job **romsM**.

Create a new subdirectory, **Case1**, and save the solution in it for analysis and plotting to avoid overwriting solutions when playing with difference CPP options and rerunning and recompiling:

```
mkdir Case1
mv Build_roms rbl4dvar.in *.nc log Case1
cp -p romsM roms_wc13.in Case1
```

where log is the ROMS standard output specified.

Step 6:
To plot the output from step 5, go to the subdirectory **WC13/plotting** and run the Matlab script **plot_rbl4dvar_forecast_sensitivity.m**. Compare with the corresponding figure from the impact calculation of Exercise 8.

Case 2: Measuring observation sensitivities using independent observations

In this case, the 3rd-order approximation for δe becomes:

$$\delta e = \boldsymbol{d}^T (\partial \boldsymbol{\mathcal{K}} / \partial \boldsymbol{y})^T \boldsymbol{M}_b^T \left[\boldsymbol{G}_g^T \boldsymbol{C} \left(\boldsymbol{y}_g - \boldsymbol{y} \right) + \boldsymbol{G}_r^T \boldsymbol{C} (\boldsymbol{y}_r - \boldsymbol{y}) \right]. \quad (2)$$

where $\boldsymbol{G}_g^T$ and $\boldsymbol{G}_r^T$ denote the adjoint model *forced* at the observation points and linearized about green and red forecast (Fig. 1) respectively. Steps 1-3 are identical to Case 1, so there is no need to repeat these. However, we now need to create the forcing functions for the adjoint since they will be different for this case.

Step 8:
Go back to the subdirectory **WC13/RBL4DVAR_forecast_sensitivity** and compile the forecast observation sensitivity driver using **build_roms.csh** using the following cpp options:

```
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRBL4DVAR_FCT_SENSITIVITY"
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DAD_IMPULSE"
#setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DOBS_IMPACT"
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DOBS_SPACE"
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRPCG"
```

Now run the job **romsM**.

Create a new subdirectory, **Case2**, and save the solution in it for analysis and plotting to avoid overwriting solutions when playing with difference CPP options and rerunning and recompiling:

```
mkdir Case2
mv Build_roms rbl4dvar.in *.nc log Case2
cp -p romsM roms_wc13_2hours.in Case2
```

where log is the ROMS standard output specified.

Step 9:
To plot the output from step 8, go to the subdirectory **WC13/plotting** and run the Matlab script **plot_rbl4dvar_forecast_sensitivity_obs_space.m**. Compare with the corresponding figure from the impact calculation of Exercise 8.