EXERCISE 3: Dual Formulation RBL4D-Var - R4D-Var

Introduction

As discussed in Lecture 3, 4D-Var can be cast in a dual form in which the search for the best ocean circulation estimate proceeds in the subspace spanned only by the observations, as opposed to the full space spanned by the model (i.e. the primal form and I4D-Var of Exercises 1 and 2). Formally, the primal and dual formulations yield identical estimates of the ocean circulation so one might wonder if there is any advantage of one form over the other? In this exercise and Exercise 4, we will explore this question, and discover that there are indeed practical advantages to both approaches.

Running RBL4D-Var

The physical-space statistical analysis approach to the dual form is run in the directory **WC13/RBL4DVAR**. Follow the instructions in the appropriate **Readme** file and run the model.

Notice that the nonlinear trajectory can be written either daily (**NHIS=48** if using **roms_wc13_daily.in**) or every two-hours (**NHIS=4** if using **roms_wc13_2hours.in**). It the basic state trajectory used to linearize the tangent linear and adjoint models. It turns out that the **daily** sampling is over the limit where the tangent linear approximation is valid. The results are much better when using the two-hours snapshots. The two set-ups are provided to make the user aware of the validity of the tangent linear approximation in highly nonlinear circulations. The differences will be noticeable when computing observation impacts and observation sensitivities.

As in Exercise 1, **roms_wc13_2hours.in** is configured to perform one outer-loop and **25** inner-loops.

You will be running dual 4D-Var algorithm three times: first using the standard $\mathbf{R}^{-1/2}$ preconditioning with a conjugate gradient (CG) method, second using the $\mathbf{R}^{-1/2}$ preconditioning and a minimum residual algorithm (MINRES), and finally using a restricted **B**-preconditioned conjugate gradient (RPCG) approach.

Case 1: $\boldsymbol{R}^{-1/2}$ preconditioning with CG

First edit the file **build_roms.csh** or **build_roms.sh** and choose the following cpp options

```
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRBL4DVAR"
#setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DMINRES"
#setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRPCG"
```

Recompile the model and then run it.

Create a new subdirectory **EX3_CONGRAD**, and save the solution in it for analysis and plotting to avoid overwriting solutions when playing with different CPP options and rerunning and recompiling:

```
mkdir EX3_CONGRAD
mv Build_roms rbl4dvar.in *.nc log EX3_CONGRAD
cp -p romsM roms_wc13_2hours.in EX3_CONGRAD
```

where log is the ROMS standard output specified

Case 2: $\boldsymbol{R}^{-1/2}$ preconditioning with MINRES

First edit the file **build_roms.csh** or **build_roms.sh** and choose the following cpp options

```
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRBL4DVAR"
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DMINRES"
#setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRPCG"
```

Recompile the model. Before running it again, be sure to execute **job_rbl4dvar.sh**. Then run the model.

Create a new subdirectory **EX3_MINRES**, and save the solution in it for analysis and plotting to avoid overwriting solutions when playing with different CPP options and rerunning and recompiling:

```
mkdir EX3_MINRES
mv Build_roms rbl4dvar.in *.nc log EX3_MINRES
cp -p romsM roms_wc13_2hours.in EX3_MINRES
```

where log is the ROMS standard output specified

*Case 3: Restricted **B**-preconditioned conjugate gradient (RPCG)*

First edit the file **build_roms.csh** or **build_roms.sh** and choose the following cpp options

```
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRBL4DVAR"
#setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DMINRES"
 setenv MY_CPP_FLAGS "${MY_CPP_FLAGS} -DRPCG"
```

Recompile the model. **NOTE:** The RPCG algorithm works differently than the other two previous cases, and to run 25 inner-loops you need to choose **Ninner=26** in **roms_wc13_2hours.in**. Before running the model again, be sure to execute **job_rb4dvar.sh**. Then run the model.

Create a new subdirectory **EX3_RPCG**, and save the solution in it for analysis and plotting to avoid overwriting solutions when playing with different CPP options and rerunning and recompiling:

```
mkdir EX3_RPCG
mv Build_roms rbl4dvar.in *.nc log EX3_RPCG
cp -p romsM roms_wc13_2hours.in EX3_RPCG
```

where log is the ROMS standard output specified. **EX3_RPCG** are the files you will need in future exercises.

Plotting your results

1. Plot the 4D-Var cost function *J* for the three cases using the script **plot_rbl4dvar_cost_compare.m** which can be found in **WC13/plotting**. The cost function of the primal case is also plotted for comparison.
2. Using **plot_rbl4dvar_increments.m** plot next a selection of the increments from the RPCG case which should be the last case that you ran.