ECCO Modeling Utilities (EMU)

User Guide

Ichiro Fukumori, Ou Wang, and Ian Fenty

DRAFT

December 19, 2022

1. Introduction

This document describes computational tools to analyze the ocean model underlying the ECCO ocean state estimate Version 4 release 4 (V4r4).

The tools include the following (Table 1);

1. Sampling Tool
2. Perturbation Tool (forward gradient)
3. Adjoint Tool (adjoint gradient)
4. Convolution Tool (adjoint gradient decomposition)
5. Budget Tool
6. Tracer Tool (passive tracer and its adjoint)

The tools utilize the flux-forced version of ECCO's Version 4 release 4 (V4r4) ocean model described in Wang et al. [2021], hereafter simply the *model*.

	Tool	Description
1	Sampling	Extract time-series of *user-specified quantity.*
2	Perturbation	Computes model's response to change in *user-specified forcing* (forward gradient).
3	Adjoint	Computes sensitivity of *user-specified quantity* to different forcing (adjoint gradient).
4	Convolution	Computes convolution of adjoint gradients with corresponding forcing (adjoint gradient decomposition).
5	Budget	Extract variables and fluxes underlying the budget of *user-specified quantity*
6	Tracer	Computes evolution of *user-defined passive tracer* and/or *its adjoint*.

Table 1: ECCO Modeling Utilities (EMUs)

2. What the Tools do

2.1. Sampling Tool

The Sampling Tool extracts time-series of a user-defined variable of the model, hereafter the Objective Function (J). The Objective Function can simply be one of the ***standard state variables*** v (Table 2) at a particular model grid point or a user-defined linear function (combination, transformation) of these variables (e.g., spatial integral, steric sea level). In its general form, the Objective Function is written as,

$$J(t) = \sum_i \alpha_i \sum_{\mathbf{x}} \mathbf{T}_i(\mathbf{x}) v_i(\mathbf{x}, t) \tag{1}$$

Here, t is time, α is a scalar multiplication factor (scaling), **T** is a linear operator (weight) in space (**x**), and subscript i distinguishes different variables. (The Tool allows the Objective Function to be a combination of any number of variables.)

The Sampling Tool is useful for assessing the fidelity of V4r4; e.g., comparison to observations.

Index	Variable	Unit	Description	temporal mean
1	ssh	m	dynamic sea level	daily & monthly
2	obp	m	ocean bottom pressure (unit in equivalent sea level)	daily & monthly
3	theta	°C	potential temperature	monthly
4	salt	PSU	salinity	monthly
5	uvel	m/s	i-direction velocity	monthly
6	vvel	m/s	j-direction velocity	monthly

Table 2: **Standard model state variables** (v) available as daily and/or monthly means.

2.2. Perturbation Tool

The Perturbation Tool computes the model's response to changes in forcing (aka control); i.e., forward gradient,

$$\frac{\partial v(\mathbf{x}, t)}{\partial \phi(\mathbf{r}, s)} \tag{2}$$

Here, the numerator $v(\mathbf{x}, t)$ is a standard state variable (Table 2) at location **x** and time t. $\phi(\mathbf{r}, s)$ in the denominator, chosen by the user, is a particular forcing (Table 3) at a specific location **r** and time s. The Perturbation Tool computes this gradient for all standard state variables v (Table 2) at different locations **x** and time t of the model.

The gradients are useful for studying the ocean's response to change in forcing and for assessing the accuracy of the model's adjoint gradients (Adjoint Tool, Section 2.3).

Index	Variable Name	Unit	Description
1	empmr	kg/m^2/s	upward freshwater flux
2	pload	N/m^2	downward surface pressure load
3	qnet	W/m^2	net upward heat flux
4	qsw	W/m^2	net upward shortwave radiation
5	saltflux	g/m^2/s	net upward salt flux
6	spflx	g/m^2/s	net downward salt plume flux
7	tauu	N/m^2	westward wind stress
8	tauv	N/m^2	southward wind stress

Table 3: Model Forcing (ϕ). Forcing perturbation is defined weekly.

The Perturbation Tool computes the gradient (Eq 2) as differences of the state (v) between model integrations with and without the forcing perturbation, divided (normalized) by the amplitude of that perturbation. Namely,

$$\frac{v(\mathbf{x},t;\phi') - v(\mathbf{x},t;\phi)}{\delta\phi} \approx \frac{\partial v(\mathbf{x},t)}{\partial \phi(\mathbf{r},s)} \tag{3}$$

Here on the left-hand-side, the model's forcing used in deriving the state is noted parametrically, i.e., $v(\mathbf{x},t;\phi)$ denotes model state v at location $\mathbf{x}$ and time t using forcing ϕ. ϕ' is identical to ϕ except at location $\mathbf{r}$ and time s where it has been perturbed by $\delta\phi$, viz.,

$$\phi'(\mathbf{r},s) = \phi(\mathbf{r},s) + \delta\phi \tag{4}$$

The user choses $\delta\phi$ among the different controls that are available (Table 3), its magnitude, spatial location ($\mathbf{r}$), and specific instant (s) defined at 7-day intervals, starting from 12Z January 01, 1992, which is the starting instant of ECCO V4r4. The model time-step is 1-hour and the forcing perturbation is interpolated linearly in time. The Tool integrates the model with the perturbation and evaluates the gradient (Eq 3) as daily and/or monthly means corresponding to the standard model state (Table 2). The second term in the numerator on the left-hand-side of Eq 3 is pre-computed as part of the Tool's installation (Section 4).

2.3. Adjoint Tool

The Adjoint Tool computes the model's sensitivity to different forcing; i.e., adjoint gradient,

$$\frac{\partial J(t_g)}{\partial \phi(\mathbf{r},s)} \tag{5}$$

Here the numerator $J(t_g)$ is a user-defined scalar quantity of the model, i.e., the Objective Function as described in Section 2.1 (Eq 1), at a particular time t_g defined as the gradient's "target instant". The denominator $\phi(\mathbf{r},s)$ is a forcing (Table 3) at location **r** and time *s*.

Adjoint gradients are useful for studying the sensitivity of the model to different forcing, including identification of forcing responsible for the model's variation (Convolution Tool, Section 2.4).

The Tool allows *J* to be chosen as a particular state variable at a specific location **x** or, more generally, a user-defined linear function of the state such as a spatial integral as in the Sampling Tool (Eq 1). In time, *J* for this Tool is based on monthly means; for example, *J* could be an average of a particular month or an average over a longer period based on monthly means such as a particular year or over the entire period of V4r4. Using the adjoint of the model, the Adjoint Tool computes this gradient for different controls ϕ (Table 3) at different locations **r** and time *s*. As in the Perturbation Tool (Section 2.2), controls (ϕ) are defined weekly, starting from 12Z January 01, 1992, that are interpolated linearly in time.

Adjoint gradients (Eq 5) are closely related to forward gradients (Eq 2). The two tools differ in whether it is the numerator or the denominator that is fixed (with the other one varying). **Whereas the Perturbation Tool computes the gradients for a particular denominator, the Adjoint Tool computes the gradients for a particular numerator.** Otherwise, the two gradients are mathematically the same for corresponding numerators and denominators. (Numerical differences arise from approximations.)

2.4. Convolution Tool

The Convolution Tool computes the product of a particular set of adjoint gradients (Section 2.3) and the variations of corresponding controls $\delta\phi$ that approximates changes in the gradients' Objective Function δJ; viz.,

$$\delta J(t) \approx \sum_i \sum_{\mathbf{r}} \sum_{\Delta t} \frac{\partial J(t_g)}{\partial \phi_i(\mathbf{r}, t_g - \Delta t)} \delta\phi_i(\mathbf{r}, t - \Delta t) \tag{6}$$

Here, *J* is the gradients' Objective Function and t_g is its target instant. The summation is defined over subscript *i* that distinguishes different controls ϕ (Table 3), **r** that denotes their spatial location (2-dimensional), and Δt their temporal lag from t_g and an arbitrary instance *t*. The target instant t_g being parametrically defined, Equation (6) can be recognized as a convolution over temporal lag (Δt) between the gradients and the controls.

Equation (6) is useful for studying causal relationships between *J* and ϕ. The equation is an approximation of a first-order Taylor Series expansion of variations in *J* at time *t*, $\delta J(t)$, using gradients at a particular target instant, t_g, rather than the actual instant of *J*, namely,

$$\frac{\partial J(t)}{\partial \phi_i(\mathbf{r}, t-\Delta t)} \approx \frac{\partial J(t_g)}{\partial \phi_i(\mathbf{r}, t_g-\Delta t)} \tag{7}$$

Termed adjoint gradient decomposition (*Fukumori et al., 2015*), Equation (6) provides an explicit relationship between forcing and quantity of interest, permitting identification of elements of the former responsible for the latter.

By definition, the Tool treats the controls in the denominator of the gradeints and the forcing used in the convolution having the same temporal resoution. For consistency with the Adjoint Tool (Section 2.3), the Convolution Tool assumes that this temporal resolution is weekly. The Tool by default employs gradients computed by the Adjoint Tool and forcing used by the model (7-day averaged), but also allows Users to substitute them with alternate files (e.g., *Fukumori et al., 2021*).

2.5. Budget Tool

The Budget Tool extracts time-series of a user-defined variable and the fluxes underlying its temporal evolution.

2.6. Tracer Tool

The Tracer Tool computes the temporal evolution of a user-defined passive tracer or its adjoint.

3. How to use the Tools

Before using the Tools, create and change to directory **USRDIR** (name as desired) in which the Tools' computation will take place. The tools are designed to be run one by one. Simultaneous executions should be run in separate **USRDIR** directories to avoid conflict among the Tools' execution.

pfe25>mkdir **USRDIR**

pfe25>cd **USRDIR**

User commands and input are noted in RED. System prompts, file names, directories, and variables are indicated by CYAN. (File names and directories are in **bold**.) pfe25> denotes Unix prompt on NAS Pleiades. Descriptions of the prompts are ***highlighted in bold italic***.

All Tools require three steps,

a) Set up interface,
b) Specify what to do,

c) Conduct calculation.
summarized in Table 4.

	Tool	Setup tool	Specification	Calculation
1	Sampling	setup_samp.csh	samp.x	do_samp.x
2	Perturbation	setup_pert.csh	pert.x	pbs_pert.csh
3	Adjoint	setup_adj.csh	adj.x	pbs_adj.csh
4	Convolution	setup_conv.csh	conv.x	do_conv.csh
5	Budget	setup_bud.csh	bud.x	do_bud.csh
6	Tracer	setup_trac.csh	trac.x	pbs_trac.csh

Table 4: Summary of scripts and programs

3.1. Sampling Tool

a) Set up user interface (files used in steps b & c below).

Replace **FORUSERDIR** below to where the Tool is installed (what was specified in step 15 of "Section 4 Installing the Tools"); On NAS Pleiades the Tool can be found in **/nobackup/ifukumor/ECCO_tools/emu**.

pfe25>source **FORUSERDIR/setup_samp.csh**

... Setting up ECCO V4r4 Sampling Tool ...
See **FORUSERDIR/README_samp**

b) Specify the quantity of interest (Objective Function) by following the prompt to **samp.x**.

The example below extracts monthly-mean time-series of dynamic sea level at a model grid point closest to 148°W 73.1°N, which is model grid point (85, 604).

pfe27>**samp.x**

Extracting model time-series ...

Define objective function (OBJF) ...
Available VARIABLES are ... ***List of variables for Objective Function. cf Table 2***
1) SSH (m)
2) OBP (equivalent sea level m)
3) THETA (deg C)
4) SALT (PSU)
5) UV (m/s)

Monthly or Daily mean ... (m/d)?
(NOTE: daily mean available for SSH and OBP only.)

m. ***Choosing monthly mean variables in defining Objective Function.***

fmd = m
==> Sampling MONTHLY means ...

Choose OBFJ variable # 1 ... (1-5)? ***Choose variable from 1 to 5 for i=1 in Eq (1).***
(Enter 0 to end variable selection)
1 ***Choosing SSH as variable.***
OBJF variable 1 is SSH
Choose either VARIABLE at a point (1) or VARIABLE weighted in space (2) ...
(1/2)?
1 ***Choosing 1 causes samp.x to form a sampling operator as weight T in Eq (1), i.e., T=1 at the chosen point but zero otherwise. See Section 3.3 for an example choosing 2.***
... OBJF will be a scaled VARIABLE at a point

i.e., MULT * VARIABLE

Choose horizontal location ...
Enter 1 to select native grid location (i,j),
or 9 to select by longitude/latitude ... (1 or 9)?
9

Enter location's lon/lat (x,y) ...
longitude ... (E)?
-148
latitude ... (N)?
73.1

...... Chosen point is (i,j) = 85 604. ***Lists information of chosen model grid point.***
C-grid is (long E, lat N) = -148.1 73.2
Depth (m)= 3675.7

Enter scaling factor MULT ... ? ***This is alpha in Eq (1).***
1.
amult = 1.0000E+00

Choose OBFJ variable # 2 ... (1-5)? ***Choosing variable for i=2 in Eq (1).***
(Enter 0 to end variable selection)
0 ***Selecting 0 ends definition of Objective Function.***

Sampling Tool output will be in : **emu_samp_m_1_85_604_1.** ***Output directory.***

... Done samp setup of **data.ecco**

c) Run program **do_samp.x** to extract time-series of the Objective Function specified in step b). (Sampling Tool does not require a PBS job.)

pfe20>**do_samp.x**

Sampling Tool output will be in : **emu_samp_m_2_85_604_1**

nobjf = 1 ***Number of terms (i) chosen for Eq (1) in step b)***
Sampling MONTHLY means ... ***Sampling according to user definition in b)***

Mask file : objf_1_mask_C ***Filename for T in Eq (1) for i=1.***
Masks maximum absolute value = 1.0000E+00
at (i,j) = 85 604 ***Maximum element of T in Eq (1) for i=1.***

... Done.

d) Analyze the results.

The Sampling Tool creates output in a directory bearing specification of the extracted variable (Objective Function) in its name, which is **emu_samp_m_1_85_604_1** for the case above. Here "**emu_samp**" indicates output from the Sampling Tool, "**m**" for monthly mean variable, "**1**" for SSH, "**85_604**" for location (i,j)=(85, 604), and the last "**1**" for number of variables defining the Objective Function (nobjf=1).

The files in this directory for the example above are.

pfe24>ls -l **emu_samp_m_85_604_1**

```
total 16
-rw-r--r-- 1 ifukumor g26113  332 Dec  1 19:13 data.ecco
-rw-r--r-- 1 ifukumor g26113  465 Dec  1 19:13 samp.info
-rw-r--r-- 1 ifukumor g26113 1252 Dec  1 19:13 samp.out_312
-rw-r--r-- 1 ifukumor g26113 1248 Dec  1 19:13 samp.step_312
```

The sampled variable is in file **samp.out_312** as an ***anomaly time-series from its time-mean in binary format***; The last number after "_" indicates the number of records in the anomaly time-series, which in this case is 312 monthly mean values from 1992 to 2017 of V4r4's analysis period. The time-mean reference value is given as the last variable in the file (313th in the example above.) File **samp.step_312** has the time record of the time-series, defined as the end instant of each averaging period (e.g., end of the month), in terms of the model's time-step (1-hour time-step from 12Z January 1, 1992.) An example FORTRAN code to read these output files is given below.

File **samp.info** is a text file summarizing the user-defined Objective Function and file **data.ecco** is an ECCO MITgcm input file defining the objective function.

FORTRAN

```
integer nrec
parameter (nrec=312)
real*4 anom(nrec), ref
character*256 f_in
integer istep(nrec)

f_in = 'samp.out _312'
open(60, file=trim(f_in), action='read', access='stream')
read(60) anom          Anomaly time-series of the Objective Function.
read(60) ref           Time-mean reference of the anomaly.
close(60)

f_in = 'samp.step _312'
open(61, file=trim(f_in), action='read', access='stream')
read(61) istep         Time of variable "anom" read above.
close(61)
```

3.2. Perturbation Tool

a) Set up user interface (files used in steps b & c below).

As in Section 3.1, replace **FORUSERDIR** below to where the Tool is installed (what was specified in step 15 of "Section 4 Installing the Tools"); On NAS Pleiades the Tool can be found in **/nobackup/ifukumor/ECCO_tools/emu**.

pfe25>source **FORUSERDIR/setup_pert.csh**

... Setting up ECCO V4r4 Perturbation Tool ...
See **FORUSERDIR/README_pert**

b) Specify the control perturbation by following the prompt to **pert.x**.

The example below perturbs "tauu" (Table 1) at model grid (87,605) at week 518 with magnitude -0.1 (N/m^2). (This is similar to the perturbation used in Figure A3 of Fukumori et al., 2021.) The program **pert.x** will create a namelist file **pert_xx.nml** and a text file **pert_xx.str** used by the PBS job in step c). The namelist file **pert_xx.nml** is used by the tool to perturb the control and the text file **pert_xx.str** contains a shorthand text describing the perturbation used to name the directory where the Perturbation Tool computation will take place ("7_87_605_518_-1.00E-01" for the example noting, in the order of, control, grid point, week, and amplitude).

```
pfe25>pert.x
 Available control variables to perturb ...        List of controls. cf Table 3
    1) empmr
    2) pload
    3) qnet
    4) qsw
    5) saltflux
    6) spflx
    7) tauu
    8) tauv
   Enter control ... (1- 8) ?
7                                    Choosing tauu as perturbed control.
  ..... perturbing tauu

Choose location for perturbation ...
    Enter 1 to choose native grid location (i,j),
         9 to select by longitude/latitude ... (1 or 9)?
1
    Enter native (i,j) grid to perturb ...
   i ... (1-90) ?
87
   j ... (1-1170) ?
605
  ...... perturbation at (i,j) =          87        605.   Lists the chosen model grid point.
      C-grid is (long E, lat N) =  -147.8  72.3
      Depth (m) =   3539.5

Enter week to perturb ... (1-1358) ?        Contol perturbation is defined weekly.
518
  ...... perturbing week =          518

 Default perturbation =   -0.1000E+00
      in unit N/m2 (westward wind stress)
 Enter 1 to keep, 9 to change ... ?'
1
Perturbation amplitude =   -0.1000E+00
      in unit N/m2 (westward wind stress)

 V4r4 integrates 312-months from 1/1/1992 12Z to 12/31/2017 12Z
 which requires 10-hours wallclock time.     Rough measure of required wallclock time.
 Enter how many months to integrate here ... (1-312)?
312                       Enter number of months to integrate starting from 1/1/1992.
Will integrate model over 312 months.

   ... Program has set computation periods in files data and pbs_pert.csh accordingly.
```

The Tool dynamically sets the period of integration.

```
Wrote pert_xx.nml
Wrote pert_xx.str
```

c) Submit a batch job to compute the model's response to perturbation specified in step b).

```
pfe25>qsub pbs_pert.csh
```

The batch job will create a ***job directory*** ("**emu_pert_7_87_605_518_-1.00E-01**", for the example above, named using the string in **pert_xx.str**) in which the computation will take place under **USRDIR** (cf Section 3). Two subdirectories will be created under this ***job directory***; Subdirectory **temp** will have output of the model (first term in the numerator on the left-hand-side of Eq 1) and subdirectory **pert_result** will have the gradients of the Perturbation Tool (Eq 1).

Progress of this computation can be monitored by the number of daily mean standard model state output files (one file per day) written in **temp/diags**; e.g.,

```
pfe27>ls emu_pert_7_87_605_518_-1.00E-01/temp/diags/*2d*day*data | wc -l
```

d) Analyze the results.

The final results of the Perturbation Tool can be found in subdirectory **pert_result** under the Tool's ***job directory*** (cf step c) and will have the following files;

pert_xx.nml:
Namelist file with specifics of the perturbation saved for reference. Created by **pert.x** and used by **pert_xx.x** to perturb the model's control.

state_2d_set1_day.*TIMESTEP***.data**
state_2d_set1_day.*TIMESTEP***.meta**

state_2d_set1_mon.*TIMESTEP***.data**
state_2d_set1_mon.*TIMESTEP***.meta**

state_3d_set1_mon.*TIMESTEP***.data**
state_3d_set1_mon.*TIMESTEP***.meta**

Forward gradient in MITgcm diagnostic output format; "data" are binary, "meta" are text files with "data" file information. The ***TIMESTEP*** in the filenames are model time-steps (center step of average); each file corresponds to a particular instant. The fields are on the model's native grid.

Files "**state_2d_set1_day**" have gradients of daily mean dynamic sea level (ssh) and ocean bottom pressure (obp) on the model's 2-dimensional horizontal grid. Files "**state_2d_set1_mon**" have monthly means of these same variables. Units are meters for both variables (equivalent sea level for obp) per unit perturbation of the chosen control.

Files "**state_3d_set1_mon**" have gradients of monthly mean temperature (theta; deg C), salinity (salt; PSU), i-direction velocity (uvel; m/s), and j-direction velocity (vvel; m/s) on the model's 3-dimensional grid per unit perturbation of the chosen control. (NOTE: Although controls tauu and tauv are westward and southward on the native grid, uvel and vvel are in the model's i- and j-directions.)

Units and direction of the different controls are (as noted by **pert.x** prompts),

control (1) = 'empmr' 'kg/m2/s (upward freshwater flux)'
control (2) = 'pload' 'N/m2 (downward surface pressure loading)'
control (3) = 'qnet' 'W/m2 (net upward heat flux)'
control (4) = 'qsw' 'W/m2 (net upward shortwave radiation)'
control (5) = 'saltflux' 'g/m2/s (net upward salt flux)'
control (6) = 'spflx' 'g/m2/s (net downward salt plume flux)'
control (7) = 'tauu' 'N/m2 (westward wind stress)'
control (8) = 'tauv' 'N/m2 (southward wind stress)'

Example code to read temperature, theta (the first record; irec), from file **state_3d_set1_mon.0000012396.data** as variable "fvar".

FORTRAN

```
      integer nx, ny, nr
      parameter (nx=90, ny=1170, nr=50)
      integer irec
      real*4 fvar(nx,ny,nr)
      character*256 f_in

      f_in = 'state_3d_set1_mon.0000012396.data'
      open(60, file=f_in, access='direct',
     $     recl=nx*ny*nr*4, form='unformatted')

      irec = 1
      read(60,rec=irec) fvar
```

IDL

```
nx = 90
ny = 1170
nr = 50

f_in = 'state_3d_set1_mon.0000012396.data'
```

```
close,1 & openr,1,f_in,/swap_if_little_endian
d_file = assoc(1,fltarr(nx,ny,nr))

irec = 0
fvar = d_file(irec)
```

MATLAB

```
nx = 90;
ny = 1170;
nr = 50;

f_in = 'state_3d_set1_mon.0000012396.data';
fid=fopen(f_in,'r','ieee-be');

irec = 1;
status=fseek(fid,(irec-1)*(nx*ny*nr*4),'bof');
fvar=fread(fid, [nx*ny*nr], 'single');
fvar=reshape(fvar, [nx,ny,nr]);
fclose(fid);
```

PYTHON

```
import numpy as np
nx = 90
ny = 1170
nr = 50

f_in = 'state_3d_set1_mon.0000012396.data'
dt =  np.dtype([ ('fld', '>f4', (nr,ny,nx))])
d_file = np.fromfile(f_in,dtype=dt)

irec = 0
fvar = d_file['fld'][irec]
```

3.3. Adjoint Tool

a) Set up user interface (files used in steps b & c below).

As in Section 3.1, replace **FORUSERDIR** below to where the Tool is installed (what was specified in step 15 of "Section 4 Installing the Tools"); On NAS Pleiades the Tools can be found in **/nobackup/ifukumor/ECCO_tools/emu**.

```
pfe25>source FORUSERDIR/setup_adj.csh
```

... Setting up ECCO V4r4 Adjoint Tool ...
See **FORUSERDIR/README_adj**

b) Specify Objective Function (OBJF) by following the prompt to **adj.x**.

The example below sets OBJF to be monthly-mean sea level of December 1993 averaged over the Beaufort Sea. (This is similar to that used in Figure A3 of Fukumori et al., 2021.)

```
pfe25>adj.x

Define objective function (OBJF) ...

First define OBJF time-period ...
```

The Tool uses monthly means to define OBJF.

```
  V4r4 can integrate from 1/1/1992 12Z to 12/31/2017 12Z
     which is 26-years (312-months).

  Select FIRST and LAST month of OBJF averaging period.
```

Here, mean of month 24 is chosen as OBJF.

```
  Enter FIRST month of OBJF period ... (1-312)?
24
  Enter LAST month of OBJF period ... (1-312)?
24
  PERIOD start & end months = 24 24.

  ... Program has set computation periods in files data and pbs_adj.csh accordingly.
  ... Estimated wallclock hours is    9
```

The Tool dynamically sets the period of integration from 01 January 1992 to the end of OBJF and adjusts the wallclock resource request accordingly. Gradients are computed for controls during this period.

```
Next define OBJF variable(s) ...
```

This is similar to the Sampling Tool (Section 3.1).

```
  Available VARIABLES are ...
   1) SSH (m)
   2) OBP (equivalent sea level m)
   3) THETA (deg C)
   4) SALT (PSU)
   5) UV (m/s)

------------------
```

Choose OBFJ variable # 1 ... (1-5)?
(Enter 0 to end variable selection)
1
OBJF variable 1 is SSH
Choose either VARIABLE at a point (1) or VARIABLE weighted in space (2) ...
(1/2)?
2   ***Choosing 2 causes T in Eq (1) to be read from a user-specified file.***
... OBJF will be a linear function of selected variable
i.e., MULT * SUM(MASK * VARIABLE)

!!!!! MASK must be uploaded (binary native format) before proceeding ...

Enter MASK filename ... ?
f21_a_1_a.beaufort   ***Example name of a mask file (T n Eq 1)***

fmask = f21_a_1_a.beaufort

Mask file : f21_a_1_a.beaufort.
Masks maximum absolute value =   2.2296E-03
at (i,j) =   86   597   ***Lists maximum value of mask for reference.***

Enter scaling factor MULT ... ?   ***This is alpha in Eq (1).***

1.
amult =   1.0000E+00

Choose OBFJ variable # 2 ... (1-5)?   ***Choosing variable for i=2 in Eq (1).***
(Enter 0 to end variable selection)
0   ***Selecting 0 ends definition of Objective Function.***

Adjoint Tool output will be in : emu_adj_24_24_1_f21_a_1_a.beaufort_1
Output directory.

Wrote adj.str

c) Submit a batch job to compute the adjoint gradients for OBJF specified in step b). (NOTE: This computation can be lengthy; see wallclock estimate in step b or that specified in job script **pbs_adj.csh**.)

pfe25>qsub **pbs_adj.csh**

The batch job will create a ***job directory*** in which the computation will take place under **USRDIR** (cf Section 3). The ***job directory*** bears the description of the computation in its name (the character string in **adj** in step b), which is "**emu_adj_24_24_1_f21_a_1_a.beaufort_1**" for the example above. Here "**emu_adj**" indicates output from the Adjoint Tool, "**24_24**" for the first and last months of the Objective Function's averaging period, "**f21_a_1_a.beaufort**" for the file name of the spatial mask used, and "**1**" for the number of variables defining the Objective Function (nobjf=1).

Two subdirectories will be created under the ***job directory***; Subdirectory **temp** will have output of the model and its adjoint and subdirectory **adj_result** will have the main output of the Adjoint Tool including the adjoint gradients (Eq 5).

Progress of this computation can be monitored by variable "**ad_time_tsnumber**" printed in PBS job output file **STDOUT.0000**. This variable is the time-step counter of the model. The time-step size is 1-hour and ***counts down backward*** from the ending of OBJF's definition to zero at the beginning of 01 January 1992, V4r4's initial condition. The variable is printed out every 10-days; e.g.,

```
pfe27>cd emu_adj_24_24_1_f21_a_1_a.beaufort_1
pfe27>grep ad_time_tsnumber temp/STDOUT.0000 | tail -n 3
(PID.TID 0000.0001) %MON ad_time_tsnumber          =              720
(PID.TID 0000.0001) %MON ad_time_tsnumber          =              480
(PID.TID 0000.0001) %MON ad_time_tsnumber          =              240
```

d) Analyze the results.

The final results of the Adjoint Tool can be found in subdirectory **adj_result** under the Tool's ***job directory*** (cf step c) and will have the following files;

adj.info:
A text file summarizing the objective function created by **adj.x**.

adxx_*CTRL***_0000000129.data**
adxx_*CTRL***_0000000129.meta**

Adjoint gradient in MITgcm output format; "**data**" files are binary, "**meta**" files are text files with "**data**" file information. *****CTRL***** is the name of the model's different forcing (Table 3). (**0000000129** is the "iteration" number of the particular ECCO estimate.)

Example code to read the adjoint gradient with respect to tauu at 10-weeks lag from the end of OBJF averaging period.

<u>**FORTRAN**</u>

```
integer nx, ny
parameter (nx=90, ny=1170)
integer irec, f_size, nrec, nlag
real*4 fvar(nx,ny)
character*256 f_in
```

```
      f_in = 'adxx_tauu.0000000129.data'
      inquire(file=f_in, size=f_size)
      nrec =  f_size / (nx*ny*4)          Number of records in file.

      open(60, file=f_in, access='direct',
     $     recl=nx*ny*4, form='unformatted')

      nlag = 10
      irec = nrec - nlag            Record number for 10 week lag.

      read(60,rec=irec) fvar
```

3.4. Convolution Tool

a) Set up user interface (files used in steps b & c below).

As in Section 3.1, replace **FORUSERDIR** below to where the Tool is installed (what was specified in step 15 of "Section 4 Installing the Tools"); On NAS Pleiades the Tools can be found in **/nobackup/ifukumor/ECCO_tools/emu**.

pfe25>source **FORUSERDIR/setup_conv.csh**

... Setting up ECCO V4r4 Convolution Tool ...
   See **FORUSERDIR/README_conv**

b) Specify forcing, adjoint gradients, and the maximum lag used for the convolution by following the prompt to **conv.x**.

The example below specifies V4r4's forcing, adjoint gradients for monthly-mean sea level of December 1993 averaged over the Beaufort Sea computed in the example of Section 3.3, and a maximum lag of 105 weeks.

pfe25>**conv.x**
Convolution Tool ...

Specify forcing, adjoint gradient, and maximum lag below ...

V4r4 weekly forcing is in directory
   SETUPDIR/forcing/other/flux-forced/forcing_weekly.
                    SETUPDIR is where the tool is set up (cf Section 4.13).
Use V4r4's weekly forcing ... (Y/N)?
Y   ***Enter Y (y) to use V4r4's forcing.***

Reading forcing from directory *Confirmation of forcing used.*
SETUPDIR/forcing/other/flux-forced/forcing_weekly

Specify adjoint gradients ...
Gradients must have equivalent file and directory names as Adjoint Tool output.
Gradient files must be named **adxx_***CTRL***..0000000129.data** etc
and be present in a directory named '**adj_result**'
under a parent directory prefixed '**emu_adj_**'

Enter directory name of Adjoint Tool output or its equivalent ... ?
emu_adj_24_24_1_f21_a_1_a.beaufort_1/adj_result *Example from Section 3.3.*

Reading adxx from
emu_adj_24_24_1_f21_a_1_a.beaufort_1/adj_result

number of adxx records = 107 *Number of records in adxx_tauu file.*

Zero lag at (weeks) = 106 *Last record in adxx_tauu that is not zero.*

Enter maximum lag (weeks) to use in convolution ... (0-105)?

105
nlag = 105 *Variable nlag signifies maximum lag used.*

Convolution Tool output will be in :
emu_conv_24_24_1_f21_a_1_a.beaufort_1_105
*Here "***emu_conv_***" indicates output of the Convolution Tool, "***24_24_1_f21_a_1_a.beaufort_1***" corresponds to the adjoint gradient, and the last "***105***" is the maximum lag to be used.*

... Done conv setup (**conv.out**). *Specification of convolution is written to file conv.out.*

The forcing and adjoint gradients used by the Tool can be substituted above with user-created versions of the files. The forcing files must have the same file structure and file names as V4r4's. Each forcing file must be a particular 7-day average for one of the model's controls and named

*****CTRL***_weekly_v1.***TIMESTEP***.data**

where *****CTRL***** is the control's name (Table 3) and *****TIMESTEP***** is an integer (Fortran format i10.10) denoting the ending time-step of the 7-day average. Forcing files must be provided for the entire period of interest. The forcing files contain the controls on the model's native grid in binary format; e.g.,

```
integer nx, ny
parameter (nx=90, ny=1170)
real*4 ctrl(nx,ny)
character*256 f_in
```

```
f_in = 'tauu_weekly_v1.0000206808.data'
open(60, file=f_in, access='stream')
read(60) ctrl
close(60)
```

reads zonal wind stress averaged between 28 July 2015 and 04 August 2015.

User-created adjoint gradients must have the same format and file names as output of the Adjoint Tool (Section 3.3-d) and placed in a directory named "**adj_result**" under a parent directory prefixed "**emu_adj_**". The gradient files must be present for all controls with the same number of records among them.

c) Compute the convolution by running shell script **do_conv.csh**. (NOTE: This computation is done interactively for each control in parallel using GNU Parallel.)

```
pfe25>source do_conv.csh

Conducting adxx-ctrl convolution ...

ctrl read from = SETUPDIR/forcing/other/flux-forced/forcing_weekly
adxx read from = emu_adj_24_24_1_f21_a_1_a.beaufort_1/adj_result
number of adxx records = 107
Zero lag at (weeks) = 106
maximum lag (weeks) = 105
Output will be in : emu_conv_24_24_1_f21_a_1_a.beaufort_1_105

Conv for ... 3  qnet
Conv for ... 1  empmr
Conv for ... 2  pload
Conv for ... 4  qsw
Conv for ... 6  spflx
Conv for ... 7  tauu
Conv for ... 5  saltflux
Conv for ... 8  tauv

  ... reading ctrl

  ... reading adxx

  ... computing convolution
  lag (wks) = 0
  lag (wks) = 12
  lag (wks) = 24
  lag (wks) = 36
  lag (wks) = 48
  lag (wks) = 60
  lag (wks) = 72
```

Confirmation of controls being evaluated in parallel. Output/printout is not necessarily in order.

Lag being processed, printed for monitoring progress.

lag (wks) = 84
lag (wks) = 96

... Done convolution.

d) Analyze the results.

Results of the Convolution Tool in its output directory are the following.

recon2d_*CTRL***.data**

Two-dimensional time-series of the convolution for individual controls, *****CTRL***** (Table 3),

$$\sum_{\Delta t=0}^{nlag} \frac{\partial J\left(t_g\right)}{\partial \phi_i\left(\mathbf{r}, t_g-\Delta t\right)} \delta \phi_i\left(\mathbf{r}, t-\Delta t\right) \tag{8}$$

The output is a function of space (**r**) and time (t).

recon1d_*CTRL***.data**

Time-series of global sum of the convolution at different maximum lags (k) for individual controls, *****CTRL***** (Table 3),

$$\sum_{\mathbf{r}} \sum_{\Delta t=0}^{k} \frac{\partial J\left(t_g\right)}{\partial \phi_i\left(\mathbf{r}, t_g-\Delta t\right)} \delta \phi_i\left(\mathbf{r}, t-\Delta t\right) \tag{9}$$

The output is a function time (t) and maximum lag (k).

istep_*CTRL***.data**

Time (t) of the convolution time-series for individual controls, *****CTRL***** (Table 3), defined as the end instant of each period (e.g., end of the 7-day mean), in terms of the model's time-step (1-hour time-step from 12Z January 1, 1992.) (Different forcing files can span different periods.)

conv.info

Specification of convolution set by **conv.x**, identifying forcing, adjoint gradients, maximum lag used, and name of output directory.

conv.out

Same as **conv.info** but without the comments (read by **do_conv.x**.)

conv.log

Logfile of GNU Parallel with timing information.

Example code to read the Convolution Tool's time-series output.

FORTRAN

```
integer nx, ny, nwks, nlag
```

```
      parameter (nx=90, ny=1170, nwks=1357, nlag=105)
      real*4 fvar2d(nx,ny,nwks), fvar1d(nwks,nlag+1)
      integer istep(nwks)
      character*256 f_in

      f_in = 'recon2d_tauu.data'
      open(60, file=f_in, access='direct',
     $     recl=nx*ny*4, form='unformatted')
      do i=1,nwks
        read(60,rec=i) fvar2d(:,:,i)
      enddo

      f_in = 'recon1d_tauu.data'
      open(60, file=f_in, access='direct',
     $     recl=nwks*4, form='unformatted')
      do i=1,nlag+1                 Records correspond to maximum lag of 0 to nlag.
        read(60,rec=i) fvar1d(:,i)
      enddo

      f_in = 'istep_tauu.data'
      open(60, file=f_in, access='stream')
      read(60) istep
      enddo
```

4. Installing the Tools

This section describes steps to install the tool on NAS Pleiades, which should be similar for other computing systems.

User commands/input are given in RED. (Steps 1-4 in italic are the same as those for setting up the bulk-formula version of the model described in Wang et al. [2020].) System prompts, file names, and variables are in CYAN. (File names and directories are in **bold**.) pfe25> denotes Unix prompt.

Commands are summarized in file **install_cheatsheet.txt** for reference. Groups of commands are also available in shell script files **install_***.csh** where *** refers to steps below.

1) *Create and cd to a work directory*

   pfe25>*mkdir* ***WORKDIR***

   pfe25>*cd* ***WORKDIR***

2) *Download MITgcm "checkpoint 66g"*

   pfe25>*git clone https://github.com/MITgcm/MITgcm.git -b checkpoint66g*

3) *Create and cd to subdirectory*

   pfe25>*cd **MITgcm***

   pfe25>*mkdir -p **ECCOV4/release4***

   pfe25>*cd **ECCOV4/release4***

4) *Download V4 configurations*

   pfe25>*git clone https://github.com/ECCO-GROUP/ECCO-v4-Configurations*

5) Extract flux-forced configuration of the model

   pfe25>mv **ECCO-v4-Configurations/ECCOv4\ Release\ 4/flux-forced** .

   pfe25>rm -rf **ECCO-v4-Configurations**

   pfe25>cd **flux-forced**

   pfe25>set **basedir**=`pwd`

   pfe25>mkdir **forcing**

6) Download forcing from ECCO Drive. (Substitute username "*fukumori*" below with your own username and use your WebDAV password, **NOT** your Earthdata account password.) **The second wget will take a while to complete.**

   pfe25>wget -P forcing -r --no-parent --user *fukumori* --ask-password -nH --cut-dirs=4 https://ecco.jpl.nasa.gov/drive/files/Version4/Release4/input_init

   pfe25>wget -P forcing -r --no-parent --user *fukumori* --ask-password -nH --cut-dirs=4 https://ecco.jpl.nasa.gov/drive/files/Version4/Release4/other/flux-forced

7) Load module for compilation.

   pfe25>*module purge*

   pfe25>*module load comp-intel/2020.4.304*

   pfe25>*module load mpi-hpe/mpt.2.25*

pfe25>*module load hdf4/4.2.12*

pfe25>*module load hdf5/1.8.18_mpt*

pfe25>*module load netcdf/4.4.1.1_mpt*

pfe25>*module list*

8) Compile MITgcm program (generates executable "**mitgcmuv**")

   pfe25>mkdir **build**

   pfe25>cd **build**

   pfe25>../../../../**tools**/**genmake2** -mods=../**code** -optfile=../../../../**tools**/**build_options**/linux_amd64_ifort+mpi_ice_nas -mpi

   pfe25>make depend

   pfe25>make all

   pfe25>cd ..

9) Derive adjoint of MITgcm by TAF and compile (generates executable "**mitgcmuv_ad**"). **This step requires a license for TAF. Skip if Adjoint Tool will not be used.**

   pfe25>mkdir **build_ad**

   pfe25>cd **build_ad**

   pfe25> ../../../../**tools**/**genmake2** -mods=../**code** -optfile=../**code**/linux_amd64_ifort+mpi_ice_nas -mpi

   pfe25>make depend

   pfe25>make adtaf

   pfe25>make adall

   pfe25>cd ..

10) Download scripts and programs for the Tools and compile.

pfe25>*git clone https://github.com/ECCO-GROUP/ECCO-EIS.git*

pfe25>mv **ECCO-EIS/emu** .

pfe25>rm -rf **ECCO-EIS**

pfe25>cd **emu**

pfe25>make all

11) Download data files needed by the Tool (**pert_xx.scale**). Substitute username "*fukumori*" below with your own username and use your WebDAV password, **NOT** your Earthdata account password.

pfe25>wget -r --no-parent --user *fukumori* --ask-password -nH --cut-dirs=7 https://ecco.jpl.nasa.gov/drive/files/Version4/Release4/other/flux-forced/tool_pert_data

12) Copy files that will be modified, just in case. (optional)

pfe25>cp -p **setup_samp.csh setup_samp.csh_orig**

pfe25>cp -p **README_samp README_samp_orig**

pfe25>cp -p **pbs_pert_ref.csh pbs_pert_ref.csh_orig**

pfe25>cp -p **pbs_pert.csh pbs_pert.csh_orig**

pfe25>cp -p **setup_pert.csh setup_pert.csh_orig**

pfe25>cp -p **README_pert README_pert_orig**

pfe25>cp -p **setup_adj.csh setup_adj.csh_orig**

pfe25>cp -p **README_adj README_adj_orig**

pfe25>cp -p **pbs_adj.csh pbs_adj.csh_orig**

pfe25>cp -p **setup_conv.csh setup_conv.csh_orig**

pfe25>cp -p **README_conv README_conv_orig**

pfe25>mkdir **orig**

pfe25>mv ***_orig orig**

13) Modify scripts. (Specify directory where the tool files are set up. cf step 5)

pfe25>sed -i -e "s|SETUPDIR|${basedir}|g" ***.csh**

14) Run Perturbation Tool without perturbation to obtain reference results.
This job will produce results under a new directory named **emu_pert_ref** in **basedir** (step 5).

pfe25>qsub **pbs_pert_ref.csh**

15) Copy tools (**setup_*.csh**, **README_***) for user access. Replace **FORUSERDIR** below to a full directory path name where you want to install (copy) the tools at.

pfe25>set useraccessdir=**FORUSERDIR**

pfe25>if (! -d ${useraccessdir}) mkdir ${useraccessdir}

pfe25>sed -i -e "s|PUBLICDIR|${useraccessdir}|g" **setup_*.csh**

pfe25>sed -i -e "s|PUBLICDIR|${useraccessdir}|g" **README_***

pfe25>cp -p ${basedir}/**emu**/**setup_*.csh** ${useraccessdir}

pfe25>cp -p ${basedir}/**emu**/**README_*** ${useraccessdir}

References