ECCO Modeling Utilities (EMU)

User Guide

Ichiro Fukumori, Ou Wang, and Ian Fenty

DRAFT

January 23, 2023

Table of Contents

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. Tracer Tool (passive tracer and its adjoint)
6. Budget Tool

The tools utilize the flux-forced version of ECCO's Version 4 release 4 (V4r4) ocean model described in Wang et al. [2021, 2022], hereafter simply the *model*.

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

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, OBJF). 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,

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

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,

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

Here the numerator $\bar{J}(t_g)$ is a user-defined mean Objective Function (Eq 1) defined as,

$$\bar{J}(t_g) \equiv \frac{1}{t_g - t_{start}} \int_{t_{start}}^{t_g} J(t)\,dt \tag{6}$$

i.e., time-mean of J between some instances t_{start} and t_g. The denominator $\phi(\mathbf{r},s)$ is a forcing (Table 3) at location $\mathbf{r}$ and time s. By definition, the gradient is zero for $s > t_g$ due to causality.

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 $\mathbf{x}$ 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 $\mathbf{r}$ 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 $\delta\phi$, which approximates changes in the gradients' Objective Function $\delta\bar{J}$; viz.,

$$\delta\bar{J}(t) \approx \sum_i \sum_{\mathbf{r}} \sum_{\Delta t=0}^{\Delta t_{\max}} \frac{\partial \bar{J}(t_g)}{\partial \phi_i(\mathbf{r}, t_g - \Delta t)} \delta\phi_i(\mathbf{r}, t - \Delta t) \tag{7}$$

Here $\bar{J}$ is defined by Eq (6). The summation is defined over subscript i that distinguishes different controls ϕ (Table 3), $\mathbf{r}$ that denotes their spatial location (2-dimensional), and Δt their temporal lag from t_g and an arbitrary instance t. Δt_{max} defines the maximum lag used in the computation. The target instant t_g being parametrically defined, Equation (7) can be recognized as a convolution over temporal lag (Δt) between the gradients and the controls.

The Tool is useful for studying causal relationships between an objection function and its controls, i.e., J and ϕ in Eq (7). Equation (7) 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,

$$\frac{\partial \bar{J}(t)}{\partial \phi_i(\mathbf{r}, t - \Delta t)} \approx \frac{\partial \bar{J}(t_g)}{\partial \phi_i(\mathbf{r}, t_g - \Delta t)} \tag{8}$$

Termed adjoint gradient decomposition (*Fukumori et al., 2015*), Equation (7) 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. Tracer Tool

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

The tool integrates the tracer evolution using the ECCO model's pre-computed circulation (weekly-mean advection and mixing). Users specify the initial tracer distribution (terminal distribution in case of adjoint) and the start (t_{start}) and end (t_{end}) dates of the integration. When $t_{end} < t_{start}$ the Tool computes the adjoint tracer evolution, otherwise the evolution of the forward tracer.

Passive tracers are useful for studying the origin and fate of water masses and pathways of ocean circulation. Whereas the evolution of a passive tracer describes where the tracer-tagged water goes, that of a passive adjoint tracer describes where the tracer-tagged water comes from (*Fukumori et al., 2004*). Unlike advected particles, tracer evolution accounts for effects of both advection and mixing, including convection.

2.6. Budget Tool

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

3. How to use the Tools

Before using the Tools, create and change to some directory, **USRDIR** below (name as desired), in which the Tools' computations will take place.

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***.

Each Tool is run by following the prompt of a single corresponding shell script (hereafter "Tool Script") summarized in Table 4.

The Tool Scripts, along with brief REAME files and this Guide, should be installed in a user access directory to the Tools, hereafter **FORUSERDIR** (what is specified in step 17 of "Section 4 Installing the Tools"). On Pleiades, **FORUSERDIR** is **/nobackup/ifukumor/ECCO_tools/emu**.

Each Tool Script **emu_***TOOL***.csh** runs the following three steps;

1) Set up interface (shell script **setup_***TOOL***.csh**),
2) Specify what to do (interactive program *****TOOL***.x**),
3) Conduct calculation (shell script **do_***TOOL***.csh**).

where *****TOOL***** is the abbreviation of the Tool's name (Table 4). (Each step can be run individually one after the other in place of the Tool Script.) Prompts for user input are generated in Step 2.

	Tool Name	abbreviation	Tool Script
1	Sampling	samp	**emu_samp.csh**
2	Perturbation	pert	**emu_pert.csh**
3	Adjoint	adj	**emu_adj.csh**
4	Convolution	conv	**emu_conv.csh**
5	Tracer	trc	**emu_trc.csh**
6	Budget	bud	**emu_bud.csh**

Table 4: List of Tool Script

3.1. Sampling Tool

a) Run **emu_samp.csh** to use the Tool.

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).

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

```
************************************
   EMU Sampling Tool
************************************
```

Lists the Tool's name

```
**** Step 1: Tool Setup
    Running setup_samp.csh
```

Setting up the Tool (Step 1)

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

****

Run samp.x to specify sampling.

****

**** Step 2: Specification ***Specifying what to sample (Step 2)***

Running 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 ***This example chooses monthly mean variables for its Objective Function (Eq 1).***

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

Choose OBFJ variable (v in Eq 1 of Guide) # 1 ... (1-5)? ***Variable i=1 in Eq (1).***
(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)?

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
        C-grid is (long E, lat N) = -148.1  73.2
        Depth (m)=  3675.7
```
Information of chosen model grid point.

```
  Enter scaling factor (alpha in Eq 1 of Guide)... ?
1
  amult =   1.0000E+00
```
This is alpha in Eq (1).

```
------------------
  Choose OBFJ variable (v in Eq 1 of Guide) # 2 ... (1-5)?
  (Enter 0 to end variable selection)
0
```
Variable i=2 in Eq (1).

Selecting 0 ends definition of Objective Function.

```
Sampling Tool output will be in : emu_samp_m_1_85_604_1

... Done samp setup of data.ecco


*********************************
   Run "do_samp.csh" to conduct sampling.
*********************************

**** Step 3: Calculation
     Running do_samp.csh
```
Conducting the sampling (Step 3)

```
 nobjf =            1
Sampling MONTHLY means ...
```
Number of terms (i) chosen for Eq (1) in Step 2.

Sampling chosen in Step 2.

```
  Mask file : objf_1_mask_C
  Masks maximum absolute value =   1.0000E+00
       at (i,j) =   85   604

  Masks sum =   1.0000E+00
```
Filename for T in Eq (1) for i=1.

Maximum element of T in Eq (1) for i=1.

Sum of all elements of T in Eq (1) for i=1.

```
********************************************
   Results are in emu_samp_m_1_85_604_1/output.
********************************************
```
Output directory

b) Sampling Tool output.

The Sampling Tool creates files 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). User output is collected in a subdirectory named **output**.

The files in this **output** directory are described below for the example above.

```
pfe24>ls -l emu_samp_m_85_604_1/output
total 28
-rw-r--r-- 1 ifukumor g26113   332 Jan 16 21:57 data.ecco
-rw-r--r-- 1 ifukumor g26113   469 Jan 16 21:57 samp.info
-rw-r--r-- 1 ifukumor g26113  1252 Jan 16 21:57 samp.out_312
-rw-r--r-- 1 ifukumor g26113  1248 Jan 16 21:57 samp.step_312
-rw-r--r-- 1 ifukumor g26113 10642 Jan 16 21:57 samp.txt
```

File **samp.txt** is an ASCII text file with the time-series of the user specified variable, with time (1-hour time-step from 12Z January 1, 1992) and corresponding sampled quantity listed in a table format. The time here is the end instant of the averaging period of the sampled quantity (e.g., end of month).

Files **samp.out_312** and **samp.step_312** are equivalent to **samp.txt** but in binary format. File **samp.out_312** has ***anomaly time-series of the sampled quantity from its time-mean***; 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 (313^{th} in the example above.) File **samp.step_312** has the time record of the time-series. An example FORTRAN code to read these binary 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) Run **emu_pert.csh** to use the Tool.

The example below computes the model's gradient (Eqs 2 & 3) with respect to "tauu" (Table 3) at model grid (87,605) at week 518 using a perturbation magnitude of -0.1 (N/m^2) (Eq 4). (This is similar to the perturbation used in Figure A3 of Fukumori et al., 2021.)

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

```
************************************
   EMU Perturbation Tool                     Lists the Tool's name
************************************

**** Step 1: Tool Setup                      Setting up the Tool (Step 1)
    Running setup_pert.csh

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

********************************************
   Run pert.x to specify computation.
********************************************

**** Step 2: Specification                   Specifying what to perturb (Step 2)
    Running pert.x

Perturbation Tool ...

Define control perturbation (denominator in Eq 2 of Guide) ...
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 (phi in Eq 2 of Guide) ... (1- 8)?     phi in in denominator Eq (2)
```

7
..... perturbing tauu

Choose location for perturbation (r in Eq 2 of Guide) ... ***r in denominator of Eq (2)***
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
C-grid is (long E, lat N) = -147.8 72.3
Depth (m) = 3539.5

Enter week to perturb (s in Eq 2) ... (1-1358) ? ***s in denominator of Eq (2)***
518
...... perturbing week = 518

Default perturbation (delta_phi in Eq 4 of Guide) : ***delta_phi in Eq (4)***
-0.1000E+00 in unit N/m2 (westward wind stress)
Enter 1 to keep, 9 to change ... ? ***option to choose magnitude of delta_phi***
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 months to integrate (Max t in Eq 2)... (1-312)? ***Max t in Eq (2).***
Tool always integrates from 1/1/1992.
312 ***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.
... Estimated wallclock hours is 12 ***Tool sets wallclock period of computation.***

Wrote pert_xx.nml
Wrote pert_xx.str

Perturbation Tool output will be in : emu_pert_7_87_605_518_-1.00E-01

****
Run "do_pert.csh" to compute model response.
****

```
**** Step 3: Calculation
    Running do_pert.csh
```
Conduct gradient computation (Step 3)

```
15264481.pbspl1.nas.nasa.gov
```
A batch job has been submitted for the computation[1]

```
... Batch job pbs_pert.csh has been submitted
   to compute the model's response to perturbation.

   Estimated wallclock time:
#PBS -l walltime=12:00:00

********************************************
   Results will be in emu_pert_7_87_605_518_-1.00E-01/output.
********************************************
```
Output directory

b) Analyze the results.

The Perturbation Tool creates files in a directory bearing the perturbation's specification in its name, which is **emu_pert_7_87_605_518_-1.00E-01** for the case above. Here "**emu_pert**" indicates output from the Perturbation Tool, "**7**" for perturbing tauu, "**87_605**" for the perturbation's location (i,j)=(87, 605), "**518**" for perturbing week 518, and the last "**-1.00E-01**" for perturbation magnitude. User output is collected in a subdirectory named **output**.

The files in the Tool's **output** directory are described below for the example above.

pert_xx.nml:
Namelist file with specifics of the perturbation saved for reference.

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**

---

[1] Step 3 of the Perturbation Tool may require many hours to complete. Progress of this computation can be monitored by the number of intermediate daily-mean standard model state files (one file per day, 9497-days for V4r4's 26-year integration) of the perturbed run written in subdirectory **temp/diags**; e.g.,

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

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) Run **emu_adj.csh** to use the Tool.

The example below computes the model's adjoint gradient (Eq 5) of mean dynamic sea-level averaged over the Beaufort Sea for December 1993. (The gradients are similar to those used in Fukumori et al., 2021.)

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

```
************************************
   EMU Adjoint Tool
************************************
```

Lists the Tool's name

```
**** Step 1: Tool Setup
    Running setup_adj.csh

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

Setting up the Tool (Step 1)

```
********************************************
   Run adj.x to specify computation.
********************************************

**** Step 2: Specification
    Running adj.x
```

Specifying what to perturb (Step 2)

Define objective function (OBJF; J^bar in Eq 5 of Guide) ...

$\bar{J}$ in Eq (5)

First define OBJF time-period (t_start and t_g in Eq 6 of Guide) ... t_{start} and t_g in Eq (6)

```
  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.
  Enter FIRST month of OBJF period (t_start in Eq 6 of Guide) ... (1-312)?
```

t_{start} in Eq (6)

```
24
  Enter LAST month of OBJF period (t_g in Eq 6 of Guide) ... (1-312)?
```

t_g in Eq (6)

```
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) (v in Eq 1 of Guide) ...

Variable *v* in Eq (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 (v in Eq 1 of Guide) # 1 ... (1-5)? ***Variable i=1 in Eq (1)***
(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 (T in Eq 1 of Guide) ... ?
../mask.beaufort ***Example name of a mask file (T n Eq 1)***

fmask = ../mask.beaufort

Mask file : ../mask.beaufort
Masks maximum absolute value = 2.2296E-03
at (i,j) = 86 597 ***Lists maximum value of mask and its location for reference.***
Masks sum = -1.0863E-06. ***Sum of mask's elements computed for reference.***
Here the mask's sum value is virtually nil (much smaller than the maximum) as the mask corresponds to mean sea-level of the Beaufort Sea relative to global mean sea-level.

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

Choose OBFJ variable (v in Eq 1 of Guide) # 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_mask.beaufort_1**
Output directory.

```
Wrote adj.dir_out

*********************************
   Run "do_adj.csh" to compute adjoint gradients.
*********************************

**** Step 3: Calculation
     Running do_adj.csh

15266902.pbspl1.nas.nasa.gov.

... Batch job pbs_adj.csh has been submitted
    to compute the adjoint gradients.

    Estimated wallclock time:
#PBS -l walltime=9:00:00

********************************************
   Results will be in emu_adj_24_24_1_mask.beaufort_1/output.
********************************************
```

Calculating the adjoint gradient (Step 3)

A batch job has been submitted for the computation[2]

Wallclock time of batch job.

Output directory

b) Analyze the results.

The Adjoint Tool creates files in a directory bearing its objective function's specification in its name, which is **emu_adj_24_24_1_mask.beaufort_1** for the case 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, "**mask.beaufort**" for the file name of the spatial mask used, and "**1**" for the number of variables defining the Objective Function (nobjf=1). User output is collected in a subdirectory named **output**.

The files in the Tool's **output** directory are described below for the example above.

adj.info:

---

[2] Progress of this computation can be monitored by variable "**ad_time_tsnumber**" printed in PBS job output file **STDOUT.0000** output in subdirectory **temp**. 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_mask.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
```

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.

FORTRAN

```
      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) Run **emu_conv.csh** to use the Tool.

The example below computes the convolution between the adjoint gradients derived in the example in Section 3.3 and the model's forcing. (This is similar to the convolution conducted in Fukumori et al., 2021.)

```
pfe25>source FORUSERDIR/emu_conv.csh

************************************
   EMU Convolution Tool                       Lists the Tool's name
************************************
```

**** Step 1: Tool Setup *Setting up the Tool (Step 1)*
Running setup_conv.csh

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

****
Run conv.x to specify convolution.
****

**** Step 2: Specification *Specifying what to convolve (Step 2)*
Running conv.x

Convolution Tool ...

Specify forcing, adjoint gradient, and maximum lag below ... ***Defining RHS of Eq (7)***

V4r4 weekly forcing is in directory
/nobackupp17/ifukumor/emu/MITgcm/ECCOV4/release4/flux-forced/forcing/other/flux-forced/forcing_weekly

Use V4r4's weekly forcing for convolution (phi in Eq 7 of Guide) ... (Y/N)?
phi in Eq (7)

y

Reading forcing from directory
/nobackupp17/ifukumor/emu/MITgcm/ECCOV4/release4/flux-forced/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 'output'
under a parent directory prefixed 'emu_adj_'

Enter directory name of Adjoint Tool output or its equivalent ... ? ***Gradients in Eq (7)***
emu_adj_24_24_1_mask.beaufort_1/output

Reading adxx from
emu_adj_24_24_1_mask.beaufort_1/output

number of adxx records = 107

Zero lag at (weeks) = 106

Enter maximum lag (weeks) to use in convolution (delta_t_max in Eq 7 of Guide) ...
(0-105)? ***Δt_{max} in Eq (7)***
105
nlag = 105

Convolution Tool output will be in: emu_conv_24_24_1_mask.beaufort_1_105

... Done conv setup (conv.out)

****
Run "do_conv.csh" to conduct convolution.
****

**** Step 3: Calculation ***Conduct convolution (Step 3)***
Running do_conv.csh

15267426.pbspl1.nas.nasa.gov. ***A batch job has been submitted for the computation***

... Batch job pbs_conv.csh has been submitted
to compute adjoint gradient convolution with control.

Estimated wallclock time:
#PBS -l walltime=02:00:00

****
Results will be in **emu_conv_24_24_1_mask.beaufort_1_105/output**
Output directory
****

b) Analyze the results.

The Convolution Tool creates files in a directory bearing the convolution's specification in its name, which is **emu_conv_24_24_1_mask.beaufort_1_105** for the case above. Here "**emu_conv**" indicates output from the Convolution Tool and "**24_24_1_mask.beaufort_1**" corresponds to the adjoint gradient used, and the last "**105**" is the maximum lag used. User output is collected in a subdirectory named **output**.

The files in the Tool's **output** directory are as follows.

recon2d_*CTRL***.data**

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

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

This sum is a function of space (**r**) and time (t) for a particular control (i). The quantity represents a partial sum of the terms on the RHS of Eq (7), and is useful in analyzing contributions to $\delta \bar{J}(t)$ from different locations of each separate control (e.g., Figure 9 of Fukumori et al., 2021).

recon1d_***CTRL***.data

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

$$\sum_{\mathbf{r}} \sum_{\Delta t=0}^{k} \frac{\partial J(t_g)}{\partial \phi_i(\mathbf{r}, t_g - \Delta t)} \delta \phi_i(\mathbf{r}, t - \Delta t) \qquad (10)$$

This sum is a function time (t) and maximum lag (k) for a particular control (i). The quantity represents a partial sum of the terms on the RHS of Eq (7), and is useful in analyzing contributions to $\delta \bar{J}(t)$ up to different lags of each separate control (e.g., Figure 10 of Fukumori et al., 2021).

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**.)

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
```

3.5. Tracer Tool

a) Run **emu_trc.csh** to use the Tool.

The example below computes the forward tracer evolution from 30 January 1992 to 30 March 1992, initialized to a unit value for the model grid point closest to 0m depth at 160°W 0°N.

```
pfe25>source FORUSERDIR/emu_trc.csh

*************************************
   EMU Tracer Tool                          Lists the Tool's name
*************************************

**** Step 1: Tool Setup                     Setting up the Tool (Step 1)
    Running setup_trc.csh

... Setting up ECCO V4r4 Passive Tracer Tool ...
   See FORUSERDIR/README_trc

*********************************************
   Run trc.x to specify computation.
*********************************************

**** Step 2: Specification                  Specifying what to compute (Step 2)
    Running trc.x


Passive Tracer Tool ...

 Define passive tracer distribution ...

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

Enter START and END days of integration ...
(days since 01 January 1992, between 1 and 9495)

Tool computes forward tracer when START lt END and
adjoint tracer when START gt END.

Enter start day ... (1-9495)?
30 ***Defining initial day of the computation.***
Enter end day ... (1-9495)?
90 ***Defining end day of the computation.***

Start and End days = 30 90. ***Computation will be adjoint if Start > End.***

Forward tracer computation

Enter tracer at start time ... ***Specification of initial tracer distribution.***
Choose either unit tracer at a point (1) or
user-provided distribution in a file (2) ... (1/2)?
1 ***Choosing 1 causes trc.x to create an initial tracer distribution file that is zero everywhere except a single point to be defined below. Choosing 2 will cause trc.x to ask for a user-provided file equivalent to what the tool creates here.***
... starting TRC is unit value at a point.

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)?
-160
latitude ... (N)?
0

...... Chosen point is (i,j) = 15 803 ***Information of chosen model grid point.***
C-grid is (long E, lat N) = -160.5 0.2
Depth (m)= 5053.9

Choose depth ...
Enter 1 to select native vertical level (k),
or 9 to select by meters ... (1 or 9)?
9

Enter location's distance from surface ... (m)?

0

...... closest wet level is (k) = 1   ***Information of chosen model grid point.***
at depth (m) =   5.0

Tracer Tool output will be in : **emu_trc_30_90_15_803_1**

... Done trc setup

****
Run "do_trc.csh" to compute tracer evolution.
****

**** Step 3: Calculation   ***Computing tracer evolution (Step 3)***
Running do_trc.csh

15267761.pbspl1.nas.nasa.gov.   ***A batch job has been submitted for the computation***

... Batch job pbs_trc.csh has been submitted
to compute the tracer evolution.

Estimated wallclock time:
#PBS -l walltime=1:00:00

****
Results will be in **emu_trc_30_90_15_803_1/output**   ***Output directory***
****

b) Analyze the results.

The Tracer Tool creates files in a directory bearing the tracer specification in its name, which is **emu_trc_30_90_15_803_1** for the case above. Here "**emu_trc**" indicates output from the Tracer Tool, "**30_90**" for the start and ending dates (1992-day) of the integration, "**15_803_1**" describing the initial perturbation perturbation which is grid (i,j,k)=(15, 803,1) in the example. User output is collected in a subdirectory named **output**.

The files in the Tool's **output** directory are described below for the example above.

trc.info
A text file summarizing the tracer computation specified by **trc.x**.

ptracer_mon_mean.*TIMESTEP***.data**
ptracer_mon_mean.*TIMESTEP***.meta**

“Monthly” average tracer distribution in MITgcm diagnostic output format; "data" are binary files and "meta" are text files with "data" file information. The ***TIMESTEP*** in the filenames are model time-steps (1-hour time-steps) of each average from V4r4’s initial instant (12Z 01 January 1992). These time-steps correspond to the end of each succeeding averaging period, which is nominally 30.5-days.

ptracer_mon_snap.*TIMESTEP***.data**
ptracer_mon_snap.*TIMESTEP***.meta**

Tracer distribution at particular instances (“snapshots”) in MITgcm diagnostic output format; "data" are binary files and "meta" are text files with "data" file information. The ***TIMESTEP*** in the filenames are model time-steps (1-hour time-steps) of each instant from V4r4’s initial instant (12Z 01 January 1992). These files are the same as corresponding **ptracer_mon_mean** files but the output here is instantaneous values instead of time-mean in the latter (30.5-day average).

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) Compile off-line passive tracer version of MITgcm (generates executable "**mitgcmuv**" in directory **build_trc**)

pfe25>mkdir **build_trc**

pfe25>cd **build_trc**

pfe25>/bin/cp -f ../code_offline_ptracer/**OFFLINE_OPTIONS.h.fwd**
../code_offline_ptracer/**OFFLINE_OPTIONS.h**

pfe25>../../../../tools/genmake2 -mods=../code_offline_ptracer

-optfile=../../../../tools/build_options/linux_amd64_ifort+mpi_ice_nas -mpi

pfe25>make depend

pfe25>make all

pfe25>cd ..

11) Compile off-line adjoint passive tracer version of MITgcm (generates executable "**mitgcmuv**" in directory **build_trc_adj**)

pfe25>mkdir **build_trc_adj**

pfe25>cd **build_trc_adj**

pfe25>/bin/cp -f ../code_offline_ptracer/**OFFLINE_OPTIONS.h.adj** ../code_offline_ptracer/**OFFLINE_OPTIONS.h**

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

pfe25>make depend

pfe25>make all

pfe25>cd ..

12) Prepare circulation fields for off-line adjoint passive tracer version of MITgcm

pfe25>cd **forcing/other/flux-forced**

pfe25>cp -p ../../../scripts/* .

pfe25>sh -xv reverseintime_all.sh

pfe25>cd ../../..

13) 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

14) 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

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

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

16) 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**

17) Copy tools (**setup_*.csh**, **README_*** and this Guide) 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 **emu_*.csh** ${useraccessdir}

    pfe25>cp -p **README_*** ${useraccessdir}

    pfe25>cp -p **Guide*.pdf** ${useraccessdir}

5. References

Fukumori, I., T. Lee, B. Cheng, and D. Menemenlis, 2004: The origin, pathway, and destination of Niño3 water estimated by a simulated passive tracer and its adjoint, *J. Phys. Oceanogr.*, **34**, 582-604, https://doi.org/10.1175/2515.1.

Fukumori, I., O. Wang, W. Llovel, I. Fenty, and G. Forget, 2015: A near-uniform fluctuation of ocean bottom pressure and sea level across the deep ocean basins of the Arctic Ocean and the Nordic Seas, *Prog. Oceanogr.*, ***134***, 152-172, https://doi.org/10.1016/j.pocean.2015.01.013.

Fukumori, I., O. Wang, and I. Fenty, 2021: Causal Mechanisms of Sea-level and Freshwater Content Change in the Beaufort Sea. *J. Phys. Oceanogr.*, **51**, 3217-3234, https://doi.org/10.1175/JPO-D-21-0069.1.

Wang, O., I. Fukumori, and I. Fenty, 2022: Configuration for flux-forced version of ECCO Version 4 Release 4, https://github.com/ECCO-GROUP/ECCO-v4-Configurations/blob/master/ECCOv4%20Release%204/flux-forced/doc/README_fluxforced.md

Wang, O., I. Fukumori, and I. Fenty, 2022: Offline passive tracer, https://github.com/ECCO-GROUP/ECCO-v4-Configurations/blob/master/ECCOv4%20Release%204/flux-forced/doc/README_offline_ptracer.md