# ECCO Modeling Utilities (EMU)

## Version 1.1a

## User Guide

Ichiro Fukumori, Ou Wang, and Ian Fenty

Jet Propulsion Laboratory, California Institute of Technology

**DRAFT**

May 06, 2025

## Table of Contents

# 1. Introduction

This document describes the ECCO Modeling Utilities (EMU), a set of computational tools summarized in Table 1 for analyzing the ECCO ocean state estimate (presently Version 4 release 4, V4r4, https://www.ecco-group.org/products-ECCO-V4r4.htm). EMU is geared toward analyzing the *model* underlying the ECCO estimate, as opposed to exploring its *discrete state output* in itself (Table 2). The Tools facilitate investigations of physical relationships governing variables across space, time and type that are difficult to infer from the state alone (e.g., causation).

EMU is *menu-driven* with no modeling expertise required for its usage, aimed at separating the technical task of devising and implementing model computations (what EMU does) from the scientific application of these calculations (what EMU enables). The tools are based on those employed in *Fukumori et al. [2021]* where examples of their application can be found.

| | Tool | Abb. | Description |
|---|---|---|---|
| 1 | Sampling | *samp* | Evaluates time-series of *user-specified quantity.* |
| 2 | Forward Gradient | *fgrd* | Computes model's response to *user-specified change in forcing* (forward gradient). |
| 3 | Adjoint | *adj* | Computes sensitivity of *user-specified quantity* to different forcing (adjoint gradient). |
| 4 | Convolution | *conv* | Evaluates convolution of *user-specified adjoint* gradients and *forcing* (adjoint gradient decomposition). |
| 5 | Tracer | *trc* | Computes evolution of *user-defined passive tracer* and *its adjoint.* |
| 6 | Budget | *bud* | Evaluates contributions (fluxes) to *user-specified quantity's* budget. |
| 7 | Modified Simulation | *msim* | Conducts simulation with *user-defined changes* (e.g., forcing, diagnostic output) |
| 8 | Attribution | *atrb* | Evaluates contributions to time-series of *user-specified quantity* from separate *types* of controls. |
| 9 | Auxiliary | *aux* | Generates example user input files for other EMU tools. |

Table 1: ECCO Modeling Utilities (EMUs)[1]

EMU's model is a *flux-forced* version of ECCO's Version 4 release 4 (V4r4) ocean model (*Wang et al., 2021, 2022*). Whereas the V4r4 state estimate employs bulk formulae that diagnostically evaluate the interactions (forcing) among ocean, atmosphere, and sea ice, the *flux-forced* version employs fluxes associated with these interactions as pre-computed input saved from the V4r4 estimate. Results of the two models are virtually identical to each other. The *flux-forced* formulation provides a convenient means for evaluating the effects of separate processes (e.g., heat flux, freshwater flux) as opposed to those of individual atmospheric state (e.g., air

[1] Tools described as "evaluating" analyze *model* output provided with EMU and/or output from other Tools that run the *model* anew.

temperature, humidity). This *flux-forced* version of ECCO's V4r4 ocean model will be referred to simply as the *model* in this guide. The pre-computed V4r4 equivalent output from this *model*, hereafter referred to as the *reference run* (emu_ref), is part of EMU and is downloadable along with the Tools (Section 4).

# 2. What the Individual Tools do

## 2.1. Sampling Tool

The Sampling Tool evaluates time-series of a user-defined variable from the *reference run* (emu_ref) and/or its variations (Section 2.7), hereafter the *Objective Function* (*J,* OBJF), also known as the *Cost Function* or *Quantity of Interest*. The Objective Function can simply be one of the ***standard state variables*** (Table 2) or ***forcing variables*** (Table 3) 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,

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

Here, $t$ is time, α is a scalar multiplication factor (scaling), **T** is a linear operator (weight, mask) in space (**x**), ***v*** is the chosen variable and subscript $i$ distinguishes different number of 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) and, thereby, the suitability of the ECCO *model* and EMU's other tools for an application. Users are advised to look for resources elsewhere in case the ECCO estimate does not resolve the users' particular quantity of interest.

| 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. Forward Gradient Tool

The Forward Gradient Tool computes the model's response to unit 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 Forward Gradient 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 *model'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 ($\phi$). Forcing perturbation is defined weekly. Salt flux (saltflux and spflx) is associated with the ocean's interaction with sea ice. All other fluxes combine ocean's interactions with both atmosphere and sea ice.

The Forward Gradient Tool computes the gradient (Eq 2) by finite difference, i.e., as the difference of the state (*v*) between a model integration 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 **x** and time *t* using forcing $\phi$. $\phi'$ is identical to $\phi$ except at location **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 (**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 the un-perturbed model, viz., the *reference run* (emu_ref).

## 2.3. Adjoint Tool

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

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

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

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

i.e., time-mean of $J$ between some instances $t_{start}$ and $t_g$. Time $t_g$ is the nominal instant of $\bar{J}\left(t_g\right)$ and is, hereafter, called the *target instant*. The denominator $\phi\left(\mathbf{r},s\right)$ 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). (See also Fukumori, 2022.)

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 $\phi$ (Table 3) at different locations $\mathbf{r}$ and time $s$. As in the Forward Gradient Tool (Section 2.2), controls ($\phi$) 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 specified/fixed (the other one spans the entire model space of the corresponding variable). **Whereas the Forward Gradient 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 $\delta\phi$. The product approximates changes in the gradients' Objective Function $\delta\bar{J}$; viz.,

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

Here $\bar{J}$ is defined by Eq (6). The summation is conducted over subscript $i$ that distinguishes different controls $\phi$ (Table 3), **r** that denotes their spatial location (2-dimensional), and $\Delta t$ their temporal lag from $t_g$. Variable $t$ denotes an arbitrary instant. $\Delta t_{max}$ 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 ($\Delta t$) between the gradients and the controls.

The Tool is useful for studying causal relationships between an objective function and its controls, i.e., $J$ and $\phi$ 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,

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

Termed adjoint gradient decomposition (*Fukumori et al., 2015*), Equation (7) provides an explicit causal relationship between forcing and quantity of interest, permitting identification of elements of the former (its type, location, and time) 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 resolution. 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 and its adjoint.

The tool integrates the tracer evolution using the 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.

The initial tracer can be chosen interactively as either a unit tracer at a particular location or a general three-dimensional distribution specified in a user-provided file.

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, the evolution of an adjoint passive 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 evaluates time-series of variables and fluxes for analyzing property budgets of a user-defined domain of the *reference run* (emu_ref) and its variations (Section 2.7). The analyzed budget is of a scalar quantity integrated over the domain and can be either volume, heat, salt, or salinity, following the recipe outlined by *Piecuch [2022]*.

The user-defined domain can be specified as either a single model grid point or a larger body. For the latter, the larger domain can be interactively specified by a range in latitude, longitude, and depth or specified in a file containing an array spanning the model's three-dimensional grid (a value of one indicating a grid point within the volume and zero otherwise).

Budgets are useful for assessing processes (e.g., advection vs external forcing) controlling the user-defined quantity of interest.

## 2.7. Modified Simulation Tool

The Modified Simulation Tool carries out a modified *reference run* by replacing the *model*'s default input files with those of the same name from a user specified directory. If there are no corresponding files in the user specified directory, the Tool will reproduce the *reference run* (Section 1). The replaced files could be, for example, the model's forcing, initial condition, or input files that control what is output. Model forcing files are summarized in Table 4.

| Index | File Name Prefix | Unit | Description |
|---|---|---|---|
| 1 | oceFWflx | kg/m$^2$/s | upward freshwater flux |
| 2 | sIceLoadPatmPload_nopabar | N/m$^2$ | downward surface pressure load |
| 3 | TFLUX | W/m$^2$ | net upward heat flux |
| 4 | oceQsw | W/m$^2$ | net upward shortwave radiation |
| 5 | oceSflux | g/m$^2$/s | net upward salt flux |
| 6 | oceSPflx | g/m$^2$/s | net downward salt plume flux |
| 7 | oceTAUX | N/m$^2$ | westward wind stress |
| 8 | oceTAUY | N/m$^2$ | southward wind stress |

Table 4: Model forcing files. The *model*'s forcing files are named PREFIX_6hourlyavg_YEAR where PREFIX is one of the eight listed above and YEAR is year from 1992 to 2017. Files

contain fluxes on the model grid at 6-hour intervals for each year. Variables in each file correspond to those in Table 3.

The Tool is useful for analyzing the model's response to changes made, such as its controls (forcing and initial condition), as well as to save variables not included in standard output (Table 2). The Modified Simulation Tool provides an interface to run the model with a more variety of changes than in the Forward Gradient Tool (Section 2.2).

EMU includes examples (shell scripts and programs) to create time-mean forcing files and time-mean state as initial condition. Results using the Modified Simulation Tool with these examples are part of EMU and are utilized in the Attribution Tool (Section 2.8).

## 2.8. Attribution Tool

The Attribution Tool evaluates contributions from separate types of controls to a user-defined quantity of the *reference run* (emu_ref). Contributions are evaluated for anomalies (relative to respective time-mean) in wind stress (both taux and tauy; cf Table 3), heat flux (qnet and qsw), freshwater flux (empmr), salt flux (saltflux and spflx), pressure load (pload), and the model's initial state. The user-defined quantity is specified interactively as in the Sampling Tool (Section 2.1).

The separate contributions are obtained by comparing the *reference run* (Section 1) with *model* simulations using modified input (Section 2.7). For instance, differences between the *reference run* and a model simulation replacing a particular control with its time-mean (e.g., wind) is regarded as contributions from that replaced control's temporal variation. The standard states (Table 2) from such modified simulations (emu_msim) are pre-computed and are available as part of EMU's installation (Section 4) as is the *reference run* (emu_ref).

The Tool is useful for identifying the type of controls (e.g., wind vs heat flux) responsible for a user-defined quantity. The Tool is also useful in assessing and correcting results of the Convolution Tool (Section 2.3). Results of the Attribution Tool are equivalent to comparing different terms that make up the first summation in Equation (7) of the Convolution Tool. The Attribution Tool entails fewer approximations than the Convolution Tool and is therefore more accurate. (The Attribution Tool only approximates the model linearly with respect to the controls.) The Attribution Tool, however, does not distinguish contributions as a function of the control's location or instance that the Convolution Tool does (the second and third sums in Equation 7).

## 2.9. Auxiliary Tool

The Auxiliary Tool generates examples of what other EMU tools employ as user input files. The example input files include;

1) Masks defining objective function [misc_mask.sh],
2) Replacement files for Modified Simulation Tool;
    a. File (file named 'data') specifying model integration period (i.e., start and duration) [misc_msim_data.sh],

b. File (file named 'data.diagnostics') specifying diagnostic output (i.e., variable and its temporal and spatial sampling) [misc_msim_diagnostics.sh],
c. Time-mean forcing files in place of ECCO's time-variable estimates [misc_msim_forcing.sh],
d. Initial condition file with time-mean state in place of ECCO's estimate [misc_msim_ic.sh],
e. Initial condition file with the end state from another simulation and corresponding 'data' file specifying model integration period [misc_pickup.sh],

Item 1) is provided for reference only; EMU's individual tools provide the same utility. Item 2a) should *NOT* be used in conjunction with 2e) as one will overwrite what the other creates.

## 3. How to use the Tools

EMU's Tools (Table 1) are run by following the prompt of command **emu**, e.g.,

pfe25>**emu**

Here, user input will be noted in RED, system prompts in CYAN; pfe25> denotes the Unix prompt on NAS Pleiades. Names of files (which command emu above is such) and directories are denoted in **bold** type. Brief descriptions of the prompts are given in ***bold italic highlighted in yellow***.

EMU's Tools are accessed through files installed in a user access directory, hereafter denoted **FORUSERDIR** (what is specified in step 17 of "Section 4 Installing the Tools"). On Pleiades, **FORUSERDIR** for the present EMU Version 1 is **/nobackup/ifukumor/emu_v1_access**. If directory **FORUSERDIR** is not in the user's search path, enter the full pathname to **emu** above (e.g., **/nobackup/ifukumor/emu_v1_access/emu**).

The **emu** command will prompt the user to select a tool;

pfe25>**emu**

ECCO Modeling Utilities (EMU) ...
See **/nobackup/ifukumor/emu_v1_access/README** ***The README file provides a brief description of EMU***

Choose among the following tools ...

1) Sampling (samp); Evaluates state time-series from model output.
2) Forward Gradient (fgrd); Computes model's forward gradient.
3) Adjoint (adj); Computes model's adjoint gradient.
4) Convolution (conv); Computes adjoint gradient decomposition.
5) Tracer (trc); Computes passive tracer evolution.
6) Budget (budg); Evaluates budget time-series from model output.
7) Modified Simulation (msim); Re-runs model with modified input.
8) Attribution (atrb); Evaluates state time-series by control type.

9) Auxiliary (aux); Generates user input files for other EMU tools.

Enter choice ... (1-9)? ***Prompt for Tool***

Enter a number from 1 to 9 corresponding to the Tool of interest. Examples of using the different tools are described below.

EMU's Tools will generate files under the directory from which the **emu** command is initiated. Unless noted otherwise, all files are binary files. Numbers in EMU's binary input and output files are 32-bit big endian for both float and integer.

## 3.1. Sampling Tool

a) Example running the Sampling Tool

To run the Sampling Tool, choose 1 in response to command **emu**'s prompt for choice of tool. The example below evaluates monthly-mean time-series of dynamic sea level relative to its global mean at a model grid point closest to 148°W 73.1°N, which is model grid point (85, 604).

Enter choice ... (1-9)?
1 ***Entering 1 to select EMU's Sampling Tool***
choice is 1) Sampling Tool (samp)
See **/nobackup/ifukumor/emu_v1_access/README_samp** ***File provides brief description of Sampling Tool***

```
************************************
   EMU Sampling Tool (native)
************************************
```

***"Native" indicates locally compiled version of the tool being invoked, not a containerized version.***

**** Step 1: Tool Setup
Running **setup_samp.sh**

**** Step 2: Specification

By default, tool will sample EMU reference run from state files in directory
**/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_ref/diags** ***This is the directory where model's diagnostic output is.***

Enter return or enter an alternate directory if sampling another run ... ?
<return> ***Press Enter/Return to choose default option above.***
... sampling default EMU reference run.

Running **samp.x**

State will be read from :
**/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_ref/diags**

Evaluating 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) ***Choose from 1 to 5 from list above.***
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

Should value at point be relative to global mean ... (enter 1 for yes)?
1  ***Choosing 1 modifies the weight T in Eq (1) accordingly by subtracting the fractional area at each grid point (i.e., area of each grid point relative to area over the globe.)***
... OBJF will be relative to global mean

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

------------------
Choose OBFJ variable (v in Eq 1 of Guide) # 2 ... (1-5)?  ***Variable 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**

... Done samp setup of **data.ecco.**  ***This is the end of the Tool's Specification (Step 2).***

**** Step 3: Calculation  ***This step does the actual sampling.***
Running **do_samp.x**
inputdir read : **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_ref/diags**
nobjf =  1  ***Number of terms (i) chosen for Eq (1) in Step 2.***
Sampling MONTHLY means ...  ***Sampling chosen in Step 2.***

Mask file : **objf_1_mask_C**  ***Filename for T in Eq (1) for i=1.***
***Values for T below are output for confirmation.***
Masks maximum absolute value =   9.9999E-01  ***Maximum value of T is not 1, because sampling in this example is at a point relative to global mean.***
at (i,j) =   85   604  ***Maximum element of T in Eq (1) for i=1.***
Masks sum =  -2.5539E-05  ***The sum of elements in T. This is virtually zero, because sampling in this example is at a point relative to global mean.***

********************************************
Results are in **emu_samp_m_1_85_604_1/output**  ***Directory where output can be found***
********************************************

b) Sampling Tool output.

The Sampling Tool creates files in a directory bearing specification of the evaluated 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   654 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*** (float); The last number after “_” in the file name 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 (float) 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 (integer). 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. Forward Gradient Tool

a) Example running the Forward Gradient Tool.

To run the Forward Gradient Tool, choose 2 in response to command **emu**'s prompt for choice of 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.)

```
Enter choice ... (1-9)?
2            Entering 2 to select EMU's Forward Gradient Tool
choice is 2) Forward Gradient Tool (fgrd)
See /nobackup/ifukumor/emu_v1_access/README_fgrd   File provides a brief
                                                   description of the
                                                   Forward Gradient Tool
************************************
   EMU Forward Gradient 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[2]***

... 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 Forward Gradient 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_fgrd**" 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.

**fgrd_xx.nml**
Namelist file with specifics of the perturbation saved for reference.

---

[2] Step 3 of the Forward Gradient 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

**fgrd_spec.info**

A text file summarizing the user's response to the Tool's prompt describing its computation.

**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 **fgrd.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) Example running the Adjoint Tool

To run the Adjoint Tool, choose 3 in response to command **emu**'s prompt for choice of 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.)

Enter choice ... (1-9)?
3 *Entering 3 to select EMU's Adjoint Tool*
choice is 3) Adjoint Tool (adj)
See **/nobackup/ifukumor/emu_v1_access/README_adj** *File provides brief description of Adjoint Tool*

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

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

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

```
********************************************
   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                  Calculating the adjoint gradient (Step 3)
     Running do_adj.csh

15266902.pbspl1.nas.nasa.gov.  A batch job has been submitted for the computation[3]

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

   Estimated wallclock time:
#PBS -l walltime=9:00:00              Wallclock time of batch job.

*******************************************
```

[3] 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
```

Results will be in **emu_adj_24_24_1_mask.beaufort_1/output.** ***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**:
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) Example running the Convolution Tool

To run the Convolution Tool, choose 4 in response to command **emu**'s prompt for choice of 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.)

Enter choice ... (1-9)?
4 ***Entering 4 to select EMU's Convolution Tool***
choice is 4) Convolution Tool (conv)
See **/nobackup/ifukumor/emu_v1_access/README_conv** ***File provides a brief description of the Convolution Tool***

```
***********************************
   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)? ***$\Delta t_{max}$ 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 ($\mathbf{r}$) 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 $\delta\bar{J}\left(t\right)$ 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),

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

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 $\delta\bar{J}\left(t\right)$ 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) Example running the Forward Gradient Tool.

To run the Tracer Tool, choose 5 in response to command **emu**'s prompt for choice of 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.

```
Enter choice ... (1-9)?
5            Entering 5 to select EMU's Tracer Tool
choice is 5) Tracer Tool (trc)
```

See **/nobackup/ifukumor/emu_v1_access/README_trc** ***File provides a brief description of the Tracer Tool***

```
************************************
   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).

## 3.6. Budget Tool

a) Example running the Budget Tool

To run the Budget Tool, choose 6 in response to command **emu**'s prompt for choice of tool. The example below evaluates variables and fluxes for the heat budget of the top 50m of the Nino 3.4 area (5N-5S, 170W-120W).

Enter choice ... (1-9)?
6 ***Here, we are entering 6 to select EMU's Budget Tool***
choice is 6) Budget Tool (budg)
See **/nobackup/ifukumor/emu_v1_access/README_budg** ***File provides brief description of Budget Tool***

************************************
EMU Budget Tool (native) ***"Native" indicates locally compiled version of the tool being invoked, not a containerized version.***
************************************

**** Step 1: Tool Setup
Running setup_budg.sh

... Setting up ECCO V4r4 Budget Tool ...
See **/nobackup/ifukumor/emu_v1_access/README_budg**

**** Step 2: Specification

By default, tool will sample EMU reference run from state files in directory
**/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_ref/diags** ***Reference run is installed as part of EMU implementation (cf Section 4).***

Enter return or enter an alternate directory if sampling another run ... ?
<return> ***Hit Enter or Return key to choose default option above.***

... sampling default EMU reference run.
Running **budg.x**
inputdir read : **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced** ***Directory where EMU is installed (cf Section 4).***
srcdir read : **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_ref/diags** ***Directory where budget files will be read from (chosen above).***
tool files read : **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced**

Evaluating budget time-series ...

Define budget variable ...
Available VARIABLES are ... ***Choice of budget variable.***
1) Volume (m^3)
2) Heat (theta) (degC)

3) Salt (PSU)
4) Salinity (PSU)
==> Budget is MONTHLY ... ***Budgets are evaluated monthly.***

-------------------
Choose budget variable (v in Eq 1 of Guide) ... (1-5)? ***Choose 1-5 from list above.***
2 ***Here, we are entering 2 to conduct heat budget.***
Budget variable is Heat (theta)

Choose budget for a single model grid point (1) or
over a larger volume (2) ... (1/2)?
2 ***Here, we are entering 2 to analyze the budget over a volume.***
... Budget will be over a volume

Choose either a lat/lon/depth volume (1) or
a volume specified in a user-provided file (2) ... (1/2)?
(user file must be in model's native binary format)
1 ***Here, we are entering 1 to specify the volume interactively by selecting a range for latitude, longitude and depth.***
... Budget will be over a lat/lon/depth volume
Enter west most longitude (-180E to 180E)... x1?
-170 ***Numbers entered below correspond to the example volume.***
Enter east most longitude (-180E to 180E)... x2?
(choose x2=x1 for zonally global volume)
-120
Enter south most latitude (-90N to 90N)... y1?
-5
Enter north most latitude (-90N to 90N)... y2?
5
Enter deepest depth (0-6000m) ... z1?
50
Enter shallowest depth (0-6000m)... z2?
0
min/max longitude -170.0 -120.0
min/max latitude -5.0 5.0
min/max depth 50.0 0.0

***All model grid points within this range defines the volume over which the budget will be evaluated.***

Budget Tool output will be in : **emu_budg_m_2_-170.0_-120.0_-5.0_5.0_50.0_0.0**

... Done budg setup of **data.ecco** ***This is the end of the Tool's Specification (Step 2).***

**** Step 3: Calculation ***This step does the actual evaluation of variables.***
Running **do_budg.x**
18675644.pbspl1.nas.nasa.gov
... Batch job **pbs_budg.sh** has been submitted ***Computation is performed in batch.***

to compute the budget.

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

```
**********************************************
   Results will be in emu_budg_m_2_-170.0_-120.0_-5.0_5.0_50.0_0.0/output
**********************************************
```

***Output will be placed in this directory upon completion of the batch job.***

b) Budget Tool output.

The Budget Tool creates files in a directory bearing specification of the budget in its name, which is **emu_budg_m_2_-170.0_-120.0_-5.0_5.0_50.0_0.0** for the case above. Here "**emu_budg**" indicates output from the Budget Tool, "**m**" for monthly mean variable, "**2**" for Heat Budget, "**-170.0_-120.0_-5.0_5.0_50.0_0.0**" for the interactively chosen range of longitude, latitude, and depth of the volume. User output is collected in a subdirectory named **output**.

The files in this **output** directory are described below for the example above. (Here, command **lss** is a shorthand alias of **ls -log --time-style=+""**.)

```
pfe20>lss emu_budg_m_2_-170.0_-120.0_-5.0_5.0_50.0_0.0/output
total 15364
-rw-r--r-- 1      554  budg.info
-rw-r--r-- 1      307  data.ecco
-rw-r--r-- 1   624005  emu_budg.mkup_adv_x
-rw-r--r-- 1   249605  emu_budg.mkup_adv_y
-rw-r--r-- 1  2496005  emu_budg.mkup_adv_z
-rw-r--r-- 1  6240005  emu_budg.mkup_atm
-rw-r--r-- 1   624005  emu_budg.mkup_mix_x
-rw-r--r-- 1   249605  emu_budg.mkup_mix_y
-rw-r--r-- 1  2496005  emu_budg.mkup_mix_z_e
-rw-r--r-- 1  2496005  emu_budg.mkup_mix_z_i
-rw-r--r-- 1    80004  emu_budg.msk3d_a
-rw-r--r-- 1    80004  emu_budg.msk3d_v
-rw-r--r-- 1     8004  emu_budg.msk3d_x
-rw-r--r-- 1     3204  emu_budg.msk3d_y
-rw-r--r-- 1    32004  emu_budg.msk3d_z
-rw-r--r-- 1     8828  emu_budg.sum_tend
-rw-r--r-- 1     8828  emu_budg.sum_tint
```

**budg.info**

A text file, created by **budg.x**, summarizing the user-specified budget computation.

**data.ecco**

An ECCO MITgcm namelist file (text file) defining the objective function, modified by

**budg.x**, and used by **do_budg.x** to conduct the budget computation.

**emu_budg.mkup_****

Time-series of spatially varying converging fluxes making up individual terms of the budget. These files are useful for analyzing the spatial location of the fluxes contributing to the budgeted quantity. Time-series of the spatial sum of these files are summarized in **emu_budg.sum_tend**. The string ******** indicates particular terms in the budget summarized below for heat budget. See *Piecuch (2022)* for description of the terms. (Some of these files will be absent if the budget has no corresponding elements, as in geothermal flux in the example above.) Per divergence theorem, the fluxes are those along the bounding surface of the budget's domain (2d surface in 3d space), except for short-wave radiation that penetrates the volume (included in **emu_budg.mkup_atm** for this heat budget example.) (Sea ice's salt-plume flux, a component for salt and salinity budgets, is also a penetrative flux deposited inside a budget's volume.)

- **adv_x**: Advection in the horizontal i-direction.
- **adv_y**: Advection in the horizontal j-direction.
- **adv_z**: Advection in the vertical r-direction.
- **atm**: Fluxes from atmosphere & sea ice.
- **geo**: Geothermal heating.
- **mix_x**: Mixing in the horizontal i-direction.
- **mix_y**: Mixing in the horizontal j-direction.
- **mix_z_e**: Explicit mixing in the vertical r-direction.
- **mix_z_i**: Implicit mixing (convection) in the vertical r-direction.

For volume budget, possible terms are

- **adv_x**: Advection in the horizontal i-direction.
- **adv_y**: Advection in the horizontal j-direction.
- **adv_z**: Advection in the vertical r-direction.
- **srf**: Surface fluxes.

For salt budget, possible terms are

- **adv_x**: Advection in the horizontal i-direction.
- **adv_y**: Advection in the horizontal j-direction.
- **adv_z**: Advection in the vertical r-direction.
- **frc_sflux**: Surface salt flux from sea ice.
- **frc_oceSP**: Penetrative salt flux from sea ice.
- **mix_x**: Mixing in the horizontal i-direction.
- **mix_y**: Mixing in the horizontal j-direction.
- **mix_z_e**: Explicit mixing in the vertical r-direction.
- **mix_z_i**: Implicit mixing (convection) in the vertical r-direction.

The salinity budget does not take a divergence form and does not have directional information. For brevity, the spatial distribution of the fluxes that make up the budget are not individually output by the Tool. (Each term is 3-dimensional spanning the domain of interest making the file sizes much larger, if saved, than those for fluxes on the domain's boundary.) Spatially integrated summary of the salinity budget is, as are other budgets, given in files **emu_budg.sum_tend** and **emu_budg.sum_tint** described below.

The files contain the following items in this order;

> **msk**: A single character identifying the spatial location of the fluxes (character). The character corresponds to string **?** in file **emu_budg.msk3d_?** below that defines this location.
>
> **i31**: The term number in file **emu_budg.sum_tend** that fluxes in this file are aggregated in (integer)
>
> **b3d**: An array of **n3d** elements with the fluxes where **n3d** is given in file **emu_budg.msk3d_?** (float)

Array **b3d** is repeated for each month that is available (which totals **nmonths** in file **emu_budg.sum_tend**).

### emu_budg.msk3d_?

Spatial location of the convergence in files **emu_budg.mkup_**********. The string **?** is a letter denoting the type of location described below. The location string (**?**) is also the first record of each convergence file (**emu_budg.mkup_**********) indicating the corresponding location for the converging fluxes in that file.

> **a**: Location of fluxes at the ocean surface including shortwave radiation.
> **v**: Location spanning the entire volume of the domain.
> **x**: Location of fluxes in the horizontal i-direction.
> **y**: Location of fluxes in the horizontal j-direction.
> **z**: Location of fluxes in the vertical r-direction.
>
> **s**: Location of fluxes at the ocean surface.
> **g**: Location of geothermal fluxes (ocean bottom).

The files contain the following items in this order;

> **n3d**: number of locations in file (integer)
> **f3d**: array with n3d elements used as weights in evaluating fluxes (float)
> **i3d**: array with n3d elements indicating i-index of the flux (integer)
> **j3d**: array with n3d elements indicating j-index of the flux (integer)
> **k3d**: array with n3d elements indicating k-index of the flux (integer)

### emu_budg.sum_tend

Spatially integrated summary scalar time-series of each term in the budget (tendency budget). (File is created as unit 31 in **do_budg.f**) The file contains the following items in this order;

> **ibud**: type of budget (integer, 2 for heat budget)
> **nmonths**: number of months (integer)
> **tname**: name of variable (fixed-length string with 12 characters)
> **tvar**: variable (float array with **nmonths** elements)

The pair of records **tname** and **tvar** are repeated for all items that make up the budget, which differ with budget type and budget domain. The first pair is always the length of time (hours) for each month (**tname**=dt) and the second pair is always the left-hand-side of the budget (**tname**=lhs). The left-hand-side is given here for reference purpose only in checking consistency with terms on the right-hand-side of the budget in this file. The left-hand-side in this file is based on instantaneous states at the end of the month, except for the first and last months whose tendency is artificially set to zero due to missing output.

For heat budget, there are an additional six possible pairs for **tname** and **tvar** depending on domain. The variable name **tname** for these six are;

advh: horizontal advection (sum of **emu_budg.mkup_adv_x** and **emu_budg.mkup_adv_y**)
mixh: horizontal mixing (sum of **emu_budg.mkup_mix_x** and **emu_budg.mkup_mix_y**.)
advv: vertical advection
mixv: vertical mixing (sum of **emu_budg.mkup_mix_z_e** and **emu_budg.mkup_mix_z_i**.)
tfrc: surface forcing (atmosphere & sea ice)
geo: geothermal forcing

For volume budget, there are an additional three possible pairs for **tname** and **tvar** depending on domain. The variable name **tname** for these three are;

advh: horizontal advection (sum of **emu_budg.mkup_adv_x** and **emu_budg.mkup_adv_y**)
advv: vertical advection
vfrc: surface forcing

For salt budget, there are an additional five possible pairs for **tname** and **tvar** depending on domain. The variable name **tname** for these five are;

advh: horizontal advection (sum of **emu_budg.mkup_adv_x** and **emu_budg.mkup_adv_y**)
mixh: horizontal mixing (sum of **emu_budg.mkup_mix_x** and **emu_budg.mkup_mix_y**.)
advv: vertical advection
mixv: vertical mixing (sum of **emu_budg.mkup_mix_z_e** and **emu_budg.mkup_mix_z_i**.)
sfrc: surface forcing (sea ice)

For salinity budget, there are an additional five possible pairs for **tname** and **tvar** depending on domain. The variable name **tname** for these five are;

advh_slt: horizontal advection of salt
advh_vol: horizontal advection of volume
mixh: horizontal mixing
advv_slt: vertical advection of salt
advv_vol: vertical advection of volume
mixv: vertical mixing
sfrc_slt: surface forcing of salt (sea ice)
sfrc_vol: surface forcing of volume

**emu_budg.sum_tint**

Time-integral of the tendency budget (**emu_budg.sum_tend**). This file is useful for assessing processes controlling the quantity of interest itself (volume, heat, salt, salinity) instead of its tendency. The file content is the same as **emu_budg.sum_tend** but with all variables (except **tname**=dt) time-integrated relative to the second month.

## 3.7. Modified Simulation Tool

a) Example running the Modified Simulation Tool

To run the Modified Simulation Tool, choose 7 in response to command **emu**'s prompt for choice of tool. The example below replaces wind forcing with its time-mean.

Enter choice ... (1-9)?
7 ***Entering 7 to select EMU's Modified Simulation Tool***
choice is 7) Modified Simulation Tool (msim)
See **/nobackup/ifukumor/emu_v1_access/README_msim** ***File provides a brief description of the Modified Simulation Tool***

************************************
EMU Modified Simulation Tool (native)
************************************

**** Steps 1 & 2: Setup & Specification

NB: V4r4's forcing is in
**/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/forcing/other/flux-forced/forcing** ***Model forcing file installed with EMU listed for reference.***
Enter directory name with user replacement files ... ?
tau ***Directory where user-created replacement files are located.***

Replacement files will be read from
**/nobackupp17/ifukumor/temp53/tau** ***Full pathname of user-directory above. where***

Output directory will be **/nobackup/ifukumor/temp53/emu_msim_tau**
***Tool creates directory where computation will take place.***

1) Set up files for MITgcm
Output budget (fluxes) ... (YES/NO)?
y ***Option to output fluxes for budget analysis.***

... outputting budget

Replacement file ... **oceTAUX_6hourlyavg_1992** ***List of replacement files identified in the user-created replacement file directory. For brevity, only a few are shown here.***
Replacement file ... **oceTAUX_6hourlyavg_1993**
…….
Replacement file ... **oceTAUY_6hourlyavg_2016**
Replacement file ... **oceTAUY_6hourlyavg_2017**
Total # of files to be replaced ... 52 ***Number of files to be replaced. The example is from replacing all wind stress files (26 yearly files for each component of wind).***

Proceed to replace and run the model ... (Y/N)?
y ***Option to abort the Tool. Users could abort here to examine files in the Tool-created directory above to examine pre-replacement files.***

Optionally, enter short description about the replacement files ... ?
(Will be copied in output file msim.info for reference.)
(Skip if not needed.)
Using 1992-2017 time-mean wind. ***Short description of the simulation to be conducted.***

Replacing ... **oceTAUX_6hourlyavg_1992** ***Confirming files replaced. For brevity, only a few are shown here.***
Replacing ... **oceTAUX_6hourlyavg_1993**
…….
Replacing ... **oceTAUY_6hourlyavg_2016**
Replacing ... **oceTAUY_6hourlyavg_2017**
Total # of files replaced ... 52

3) Run MITgcm ***Tool submits a batch job to conduct the modified simulation.***
18841731.pbspl1.nas.nasa.gov
... Batch job **pbs_msim.sh** has been submitted
to compute the model's response to modified input.

Estimated wallclock time:
#PBS -l walltime=18:00:00

```
********************************************
```
Results will be in **emu_msim_tau** ***Tool's output location. The standard output (Table 2) is located in subdirectory diags. Fluxes for budget analysis, when output, are in individual subdirectories under diags.***
```
********************************************
```

b) Modified Simulation Tool output.

The Modified Simulation Tool creates files in a directory, **emu_msim_tau** for the case above, which bears the name of the user specified directory with the replacement files. Here "**emu_msim**" indicates output from the Modified Simulation Tool and "**tau**" for the name of the user specified directory.

**msim.info**

A text file listing replacement *model* input files.

The Tool creates standard state output (Table 2) and budget output under subdirectory **diags** under the Tool's main output directory (**emu_msim_tau** for the case above). This output is controlled by the namelist file **data.diagnostics**, which can itself be modified (cf https://mitgcm.readthedocs.io/en/latest/).

**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**

With the default **data.diagnostics** file, these files contain the standard state (Table 2) 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 (for time-mean variables, the instant is the end time-step of the averaging period). The fields are on the model's native grid.

Files "**state_2d_set1_day**" have 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).

Files "**state_3d_set1_mon**" have 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. (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.)

These model states can be sampled with the Sampling Tool by specifying the **diags** directory in response to the Sampling Tool's prompt.

**ADVr_SLT_mon_mean**
**ADVr_TH_mon_mean**
**ADVx_SLT_mon_mean**
**ADVx_TH_mon_mean**
**ADVy_SLT_mon_mean**
**ADVy_TH_mon_mean**
**DFrE_SLT_mon_mean**
**DFrE_TH_mon_mean**
**DFrI_SLT_mon_mean**
**DFrI_TH_mon_mean**

**DFxE_SLT_mon_mean**
**DFxE_TH_mon_mean**
**DFyE_SLT_mon_mean**
**DFyE_TH_mon_mean**
**ETAN_mon_inst**
**ETAN_mon_mean**
**oceFWflx_mon_mean**
**oceQsw_mon_mean**
**oceSPtnd_mon_mean**
**SALT_mon_inst**
**SFLUX_mon_mean**
**TFLUX_mon_mean**
**THETA_mon_inst**
**UVELMASS_mon_mean**
**VVELMASS_mon_mean**
**WVELMASS_mon_mean**

These subdirectories under **diags** contain variables (fluxes and instantaneous state) for the modified simulation's budget analysis. Budgets can be analyzed with EMU's Budget Tool by specifying the **diags** directory in response to the Budget Tool's prompt.

## 3.8. Attribution Tool

c) Example running the Attribution Tool

To run the Attribution Tool, choose 8 in response to command **emu**'s prompt for choice of tool. Prompts of the Attribution Tool are nearly the same as the Sampling Tool. The example below evaluates steric sea level (equal to difference between dynamic sea level and ocean bottom pressure) averaged over the Beaufort Sea relative to its global mean.

Enter choice ... (1-9)?
8 ***Entering 8 to select EMU's Attribution Tool***
choice is 8) Attribution Tool (atrb)
See **/nobackup/ifukumor/emu_v1_access/README_atrb** ***File provides a brief description of the Attribution Tool***

************************************
EMU Attribution Tool (native)
************************************

**** Steps 1 & 2: Setup & Specification

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) ***Choose from 1 to 5 from list above.***
1 ***Dynamic sea level chosen as first variable of OBJF.***
OBJF variable 1 is SSH
Choose either VARIABLE at a point (1) or VARIABLE weighted in space (2) ...
(1/2)?
2. ***A spatial weight allows computation of Beaufort Sea mean relative to global mean.***
... 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) ... ?
**/nobackup/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu/emu_input/beafortsea.msk** ***The weight in this file consists of the difference between two weights, i.e., the weight for computing the Beaufort Sea mean minus that for computing the global mean. This particular file is included in EMU for reference in subdirectory emu_input.***

Mask file : **/nobackup/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu/emu_input/beafortsea.msk**
Masks maximum absolute value = 2.2279E-03 ***The maximum value of the weight,***
at (i,j) = 85 595 ***its location, and its global sum are printed for reference.***
Masks sum = -7.2392E-07 ***The global sum in this instance is essentially zero as the weight is the difference between two area means.***

Enter scaling factor (alpha in Eq 1 of Guide)... ?
1. ***A unit scaling factor is chosen for dynamic sea level, as steric sea level is dynamic sea level minus ocean bottom pressure.***
amult = 1.0000E+00

------------------
Choose OBFJ variable (v in Eq 1 of Guide) # 2 ... (1-5)? ***Variable i=2 in Eq (1).***
(Enter 0 to end variable selection)
2 ***Ocean bottom pressure chosen as second variable of OBJF.***
OBJF variable 2 is OBP
Choose either VARIABLE at a point (1) or VARIABLE weighted in space (2) ... (1/2)?
2 ***A spatial weight allows computation of Beaufort Sea mean relative to global mean.***
... 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) ... ?
**/nobackup/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu/emu_input/beafortsea.msk** ***The same weight as dynamic sea level is chosen.***

Mask file : **/nobackup/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu/emu_input/beafortsea.msk**
Masks maximum absolute value = 2.2279E-03
at (i,j) = 85 595
Masks sum = -7.2392E-07

Enter scaling factor (alpha in Eq 1 of Guide)... ?
-1. ***A scaling factor of -1 is chosen for ocean bottom pressure, as steric sea level is dynamic sea level minus ocean bottom pressure.***
amult = -1.0000E+00

------------------
Choose OBFJ variable (v in Eq 1 of Guide) # 3 ... (1-5)? ***Variable i=3 in Eq (1).***
(Enter 0 to end variable selection)
0 ***Selecting 0 ends definition of Objective Function.***

Done interactive specification.
Begin extracting time-series ...

Tool output will be in: **emu_atrb_m_1_beafortsea.msk_2.** ***Tool creates directory where computation will be conducted.***

***In following, the Tool samples OBJF from model results provided with EMU. These model results are the reference run and results from the Modified Simulation Tool with individual controls replaced by their corresponding time-mean.***

Sampling reference run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_ref/diags**

Sampling time-mean wind run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_msim/mean_oceTAUX_oceTAUY/diags**

Sampling time-mean heat flux run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_msim/mean_TFLUX_oceQsw/diags**

Sampling time-mean freshwater flux run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_msim/mean_oceFWflx/diags**

Sampling time-mean salt flux run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_msim/mean_oceSflux_oceSPflx/diags**

Sampling time-mean pressure load run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_msim/mean_sIceLoadPatmPload_nopabar/diags**

Sampling time-mean initial condition run ...
from: **/nobackupp17/ifukumor/emu_v1/MITgcm/ECCOV4/release4/flux-forced/emu_msim/mean_IC/diags**

Computing individual control contribution ...

********************************************
   Done. Results are in
**/nobackupp17/ifukumor/temp53/emu_atrb_m_1_beafortsea.msk_2/output**
********************************************

d) Attribution Tool output.

The Attribution Tool creates files in a directory bearing specification of the analysis in its name, which is **emu_ atrb_m_1_beaufortsea.msk_2** for the case above. Here "**emu_atrb**" indicates output from the Attribution Tool, "**m**" for monthly mean variable, "**1**" for the first variable defining the objective function (1 is sea level), "**beaufortsea.msk**" for the file name defining the mask used for this first variable, and "**2**" for the total number of variables defining the objective function. User output is collected in a subdirectory named **output**.

The files in this **output** directory are described below for the example above. (Files are similar to those of the Sampling Tool, except there are more entries corresponding to different controls for the Attribution Tool.)

pfe20>lss **emu_atrb_m_1_beaufortsea.msk_2/output**

```
total 72
-rw-r--r-- 1   8764 atrb.out_312
-rw-r--r-- 1   1248 atrb.step_312
-rw-r--r-- 1  48202 atrb.txt
-rw-r--r-- 1    544 data.ecco
-rw-r--r-- 1    847 set_samp.info
```

**atrb.out_312**

Monthly anomaly time-series of the objective function for the *Reference Run* (ref) and contributions to it from anomalies in different controls; i.e., surface forcing that consists of wind (*tau*), heat flux (*htflx*), freshwater flux (*fwflx*), salt flux (*sflux*) and pressure load (*pload*), and the initial state of the ocean (*ic*). The last number after "_" in the file name indicates the number of records in the anomaly time-series. The value 312 is the number of monthly mean values available from 1992 to 2017 for V4r4. The time-mean reference for the anomaly is given as the last set of variables in the file (*mean*).

The file contains the following variables in the order given. All variables are 312-element arrays except the last one (**mean**) which is a 7-element array with the time-mean references for the seven anomaly variables *ref, tau, htflx, fwflx, sflux, pload* and *ic*.

- **ref**: Array of anomaly OBJF time-series of the *Reference Run* (float)
- **tau**: Array of anomaly OBJF time-series due to wind anomaly (float)
- **htflx**: Array of anomaly OBJF time-series due to heat flux anomaly (float)
- **fwflx**: Array of anomaly OBJF time-series due to freshwater flux anomaly (float)
- **sflux**: Array of anomaly OBJF time-series due to salt flux anomaly (float)
- **pload**: Array of anomaly OBJF time-series due to pressure load anomaly (float)
- **ic**: Array of anomaly OBJF time-series due to initial state anomaly (float)
- **mean**: Array with time-mean OBJF reference for the variables above (float)

**atrb.step_312**

Time (hour from 12Z January 1, 1992) of the monthly mean values in **atrb.out_312**. Time here is the end instant of the averaging period of the sampled quantity (e.g., end of month). The file contains a single array (float) of 312-elements.

**atrb.txt**

A text file equivalent of binary files **atrb.step_312** and **atrb.out_312**. The file has time-series of the user specified objective function (not its anomaly) listed in table format (see first line of file); time (hour from 12Z January 1, 1992), *Reference Run* (ref) and contributions to it from anomalies in different controls; i.e., surface forcing that consists of wind (*tau*), heat flux (*htflx*), freshwater flux (*fwflx*), salt flux (*sflux*) and pressure load (*pload*), and the initial state of the ocean (*ic*). The time here is the end instant of the averaging period of the sampled quantity (e.g., end of month).

**data.ecco**

An ECCO MITgcm namelist file (text file) defining the objective function, modified by **set_samp.x**, and used by **do_samp.x** to evaluate objective function from different runs.

**set_samp.info**

A text file, created by **set_samp.x**, summarizing the user-specified objective function.

## 3.9. Auxiliary Tool

To run the Auxiliary Tool, choose 9 in response to command **emu**'s prompt for choice of tool. The example below creates input files to run V4r4 from the start of 1995 for 60-days.

Enter choice ... (1-9)?
9 ***Entering 9 to select EMU's Auxiliary Tool***
choice is 9) Auxiliary Tools (aux)
See **/nobackup/ifukumor/emu_v1_access/README_aux** ***File provides a brief description of the Auxiliary Tool***

```
************************************
   EMU Auxiliary Tool
************************************
```

Choose among the following examples ...

1) Masks defining objective function [samp, adj, atrb]
2) Specify model integration period [msim]
3) Specify model diagnostic output [msim]
4) Replace model forcing with its time-mean [msim]
5) Replace model initial condition with time-mean state [msim]
6) Use end state from another simulation as model initial condition [msim]

Enter choice ... (1-6)?
2 ***This example will choose the model's integration period***
choice is 2) Set integration period (running misc_msim_data.sh)

This routine creates example replacement files for EMU Modified Simulation Tool.
This particular example modifies the start and duration of model integration
specified in file data used by MITgcm.

Enter directory name for replacement file to be created in ... (rundir)?
myrun. ***Files will be created in this directory***

Creating /nobackup/ifukumor/runs/myrun ***Tool will create the directory if it doesn't exist***

V4r4 spans 312-months from 1/1/1992 12Z to 12/31/2017 12Z
EMU can re-run V4r4 from the beginning of any of these years
using V4r4's corresponding estimate as initial condition.

```
Enter year to begin integration ... (1992-2017)
1995.   Here we choose to initialize the run on January 1, 1995
Model will be integrated from beginning of year 1995
which is model timestep 26292

Enter number of days to integrate ... ?
60      Here we choose to run the model for over 60-days
Will integrate model over 60 days
which is 1440 model timesteps.

Changed files: The Tool lists the files it generated
misc_msim_data.info
data        These two files have the necessary information to run V4r4
data.ctrl   for 60-days from 01/01/1995. Other files are informational.
my_commands.sh


Successfully modified file data in directory  /nobackup/ifukumor/runs/myrun
Use this directory name as input with the Modified Simulation Tool.

misc_msim_data.sh execution complete.
```

## 4. Installing the Tools

EMU is available on GitHub at https://github.com/ECCO-GROUP/ECCO-EIS and can be installed by following the prompts from its bash script **emu_setup.sh** as follows;

```
pfe25>wget https://raw.githubusercontent.com/ECCO-GROUP/ECCO-EIS/main/emu/emu_setup.sh
pfe25>chmod +x ./emu_setup.sh
pfe25>./emu_setup.sh
```

This script will install EMU's Programs (~1GB), its User Interface (~2MB), and its Input Files (~1TB) to sub-directories where emu_setup.sh is launched, named **emu_dir**, **emu_userinterface_dir**, and **emu_input_dir**, respectively, or to alternate user-specified directories.

EMU can be compiled natively on the user system, which will require a TAF license if installing the Adjoint Tool. Alternatively, a compiled version of EMU is available as a containerized image (Singularity) which does not require a TAF license. The native vs Singularity versions of EMU's installation can be chosen during setup by emu_setup.sh. The singularity version requires Singularity (now known as Apptainer, a free software maintained by the Linux Foundation) to be available on the user system; alias command singularity to apptainer, if necessary. The singularity version of EMU also requires a compatible OpenMPI, which will be downloaded and compiled during setup.

Installation requires obtaining a **NASA Earthdata account** for downloading files from https://ecco.jpl.nasa.gov/drive/.

Following a description of what **emu_setup.sh** does, the script pauses for the user to press the ENTER key.

Press ENTER key to continue ...
***Press the ENTER key to continue***

----------------------
Enter your Earthdata username: YOURUSERNAME
Enter your WebDAV password (*NOT* Earthdata password): YOURWEBDAVPASSWD

----------------------
Enter directory name (emu_dir) to download and set up EMU's Programs (~1 GB)
or press the ENTER key to use EMU's default (emu_dir under the present directory) ... ?
***Press the ENTER key for default directory***

----------------------
Enter directory name (emu_userinterface_dir) to install EMU's User Interface (~2 MB) or press the ENTER key to use EMU's default (emu_userinterface_dir under the present directory) ... ?
***Press the ENTER key for default directory***

----------------------
Enter directory name (emu_input_dir) to download up to 1.1 TB of EMU's Input Files or press the ENTER key to use EMU's default (emu_input_dir under the present directory) .... ?
***Press the ENTER key for default directory***

----------------------
EMU's Programs can be installed in two different ways;
  1) Compiling source code on host (native)
  2) Using Singularity image (singularity)

Option 1) requires a TAF license to derive the MITgcm adjoint used by EMU's Adjoint Tool. Option 2) has compiled versions of the code in containerized form that do not require a separate TAF license to use.

Enter choice for type of EMU implementation ... (1 or 2)?
2. ***Enter 2 to install Singularity version of EMU***

----------------------
EMU uses batch scripts to run some of its tools in PBS (Portable

Batch System). The PBS commands in these shell scripts (pbs_*.sh),
installed in EMU's User Interface directory (emu_userinterface_dir)
may need to be revised for different batch systems and/or different hosts.
Alternatively, these shell scripts can be run interactively if sufficient
resources are available.

Enter the command for submitting batch jobs (e.g., qsub, sbatch,
bsub <, condor_submit, msub) or press the ENTER key to have EMU
run its batch scripts interactively ... ?
***Press the ENTER key to run EMU's batch scripts interactively***

----------------------
EMU's Input Files total 1.1 TB, of which (directory)
  175 GB (emu_ref) is needed by Sampling, Forward Gradient, Adjoint, Tracer, Budget, and Attribution
  195 GB (forcing) is needed by Forward Gradient, Adjoint, Modified Simultion
  380 GB (state_weekly) is needed by Tracer
  290 GB (emu_msim) is needed by Attribution
  (Convolution Tool uses results of the Adjoint Tool and files downloaded by default.)

Choose among the following to download ...
  0) All Input Files (1.1 TB)
  1) Files (~175 GB) needed for Sampling and Budget Tools
  2) Files (~195 GB) needed for Modified Simultion Tools
  3) Files (~370 GB) needed for Adjoint and Forward Gradient Tool
  4) Files (~465 GB) needed for Attribution Tool
  5) Files (~555 GB) needed for Tracer Tool
or press the ENTER key to skip this step, which can take a while
(~13 hours if downloading all input files.)

EMU's Input Files can be downloaded later with shell script
  /net/b230-304-t3/ecco_nfs_1/shared/EMU/singularity2/emu_userinterface_dir/emu_input_setup.sh
See
  /net/b230-304-t3/ecco_nfs_1/shared/EMU/singularity2/emu_userinterface_dir/README_input_setup
for additional detail, including options to download the input
in batch mode.

Enter Input Files download choice ... ?
0 ***Enter 0 to install all input files for EMU***

----------------------
EMU Singularity (emu.sif) requires compilation of compatible OpenMPI
which can take ~30min. Enter 1 to compile OpenMPI now as a background
job or press the ENTER key to skip this step.

EMU compatible OpenMPI can be compiled later with shell script
**emu_userinterface_dir/emu_openmpi_setup.sh**
See
**emu_userinterface_dir/README_openmpi_setup**
for additional detail, including options to install OpenMPI
in batch mode.

Enter 1 to compile now or press the ENTER key to skip ... ?
1 ***Enter 1 to download and compile OpenMPI for EMU***

----------------------
Enter number of CPU cores (nproc) to use for MITgcm employed by EMU.
Available options are ...
13
36
48
68
72
96
192
360

Enter choice for nproc ... ?
48 ***Enter the number of CPU cores to use to run MITgcm***

**********************
End of user input for EMU setup
Rest of this script is conducted without user input.

***Wait until the script ends.***

# 5. What's New

## 5.1. Version 1.1a (1.1 alpha) May 6, 2025

a) Added sampling of control (forcing) to Sampling Tool (samp).
b) Can start model integration from the beginning of any year (1992-2017) for Forward Gradient (fgrd) and Adjoint (adj) Tools, and not just from 1992.
c) Added volume transport as a menu-driven choice for objective function for Sampling (samp), Adjoint (adj), and Attribution (atrb) Tools.
d) As Tool #9, added Auxillary Tool (aux), a collection of tools that generates examples of what other EMU Tools employ as user input. This includes replacement files for the Modified Simulation Tool (msim);

i. Specify model integration period (start and duration),
ii. Specify diagnostic state output (variable and its sampling),
iii. Replace forcing with its time-mean,
iv. Replace model initial condition with the state's time-mean,
v. Specify the end state from another simulation as the initial condition for another simulation.

e) Revised Modified Simulation Tool (msim) to copy all files from the user directory to the Tool's run directory, allowing, for example, starting the model integration from the end state of another integration.
f) Allow Forward Gradient Tool (fgrd) to use runs other than the ECCO estimate (emu_ref) as a reference, such as results of the Modified Simulation Tool (msim). This option allows the Forward Gradient Tool to compare simulations started from the same year to minimize numerical errors associated with restarting the model from different years.
g) Revised EMU Input File download scripts (emu_input_install.sh, emu_input_install_4batch.sh);
    i. Download in parallel as background jobs,
    ii. Only download new files (-N) and resume partial downloads (-c).
h) Created script misc_comp_dirs.sh to check contents of EMU Input Files.

# 6. 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.

Fukumori, I., 2022: “Adjoint Modeling” A Brief Introduction, Zenodo. https://doi.org/10.5281/zenodo.5794446.

Piecuch, C., 2022: A Note on Practical Evaluation of Budgets in ECCO Version 4 Release 3, https://www.ecco-group.org/docs/evaluating_budgets_in_eccov4r3_updated_20220118.pdf

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