ECCO Resources and Tools

Overview ECCO V4 datasets, Python & Julia tools and EMU

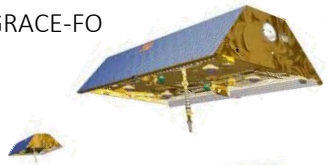
- Ian Fenty (JPL)
- Ichiro Fukumori (JPL)
- Gael Forget (MIT)

ECCO state estimates are *multi-platform, multi-instrument, multi-variable* synthesis products that integrate ocean and ice observations and models

Sentinel-6
Michael Freilich

TOPEX/Poseidon

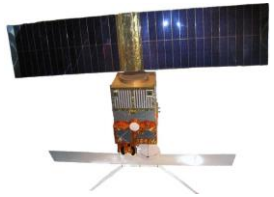
GRACE
GRACE-FO



JASON 1,2, 3

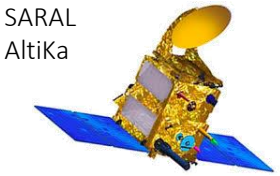
Aquarius
SAC-D

ERS 1,2

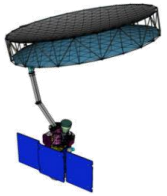


SMOS

SARAL
AltiKa

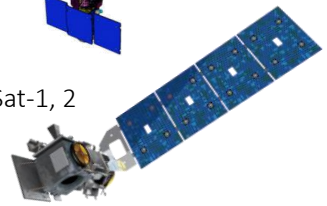


SMAP

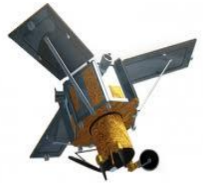


Cryosat-2

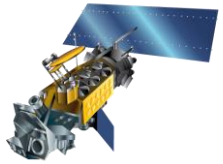
ICESat-1, 2



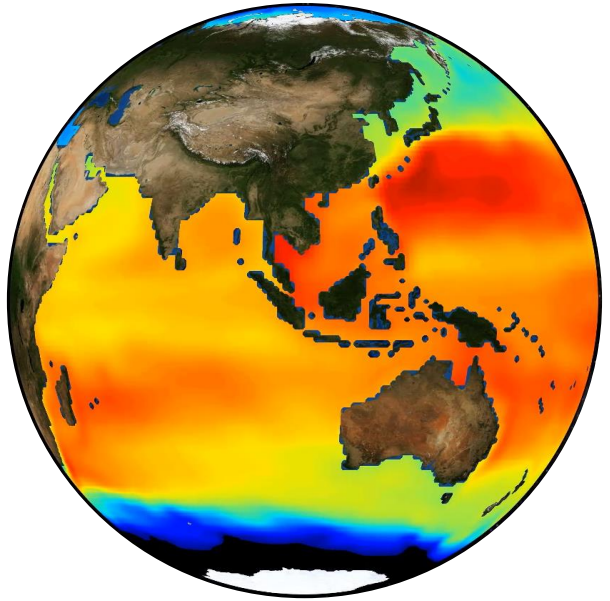
AVHRR



DMSP SMMR,
SSM/I SSMIS

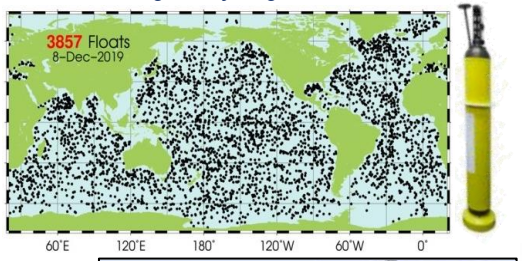


Coupled Ocean/Sea-ice General Circulation Model

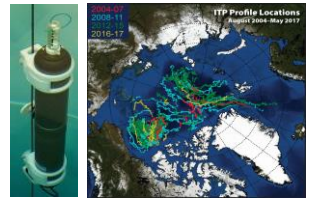


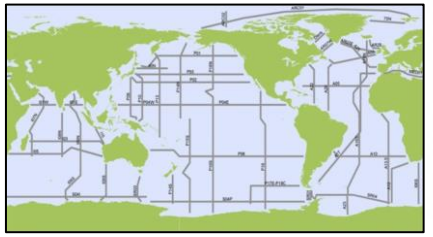
$$
\begin{aligned}
&\frac{\partial \boldsymbol{v}}{\partial t} + (f+\zeta)\hat{\boldsymbol{k}} \times \boldsymbol{v} + \nabla_{z^*}\mathrm{KE} + w\frac{\partial \boldsymbol{v}}{\partial z} + g\nabla_{z^*}\eta + \nabla_h\Phi' \\
&\quad = \boldsymbol{D}_{z^*,\boldsymbol{v}} + \boldsymbol{D}_{\perp,\boldsymbol{v}} + \mathcal{F}_{\boldsymbol{v}}, \\
&\frac{\partial \Phi'}{\partial z} = g\frac{\rho'}{\rho_c}, \\
&\frac{1}{H}\frac{\partial \eta}{\partial t} + \nabla_{z^*}(s^*\boldsymbol{v}) + \frac{\partial w}{\partial z^*} = s^*\mathcal{F}, \\
&\frac{\partial (s^*\theta)}{\partial t} + \nabla_{z^*}(s^*\theta \boldsymbol{v}_{\mathrm{res}}) + \frac{\partial (\theta w_{\mathrm{res}})}{\partial z^*} \\
&\quad = s^*(\mathcal{F}_\theta + D_{\sigma,\theta} + D_{\perp,\theta}), \\
&\frac{\partial (s^*S)}{\partial t} + \nabla_{z^*}(s^*S\boldsymbol{v}_{\mathrm{res}}) + \frac{\partial (Sw_{\mathrm{res}})}{\partial z^*} \\
&\quad = s^*(\mathcal{F}_S + D_{\sigma,S} + D_{\perp,S}),
\end{aligned}
$$

Argo Profiling Floats

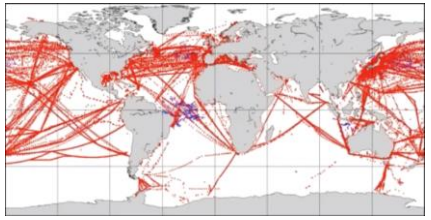


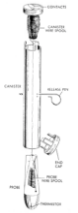
Ice-tethered profilers



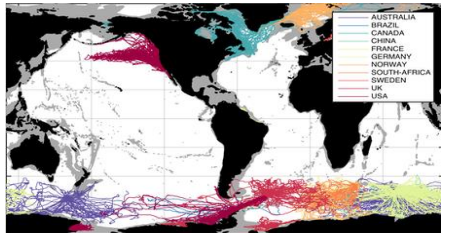


CTD
•GO-SHIP
•WOCE
•others





XBT



Instrumented marine mammals

Instrumented moorings

ECCO Version 4: a global multidecadal ocean and sea-ice reanalysis

Ocean model

Massachusetts Institute of Technology general circulation model (**MITgcm**) coupled ocean, sea-ice, ice-shelf model, and its adjoint.

Horizontal resolution

1-degee with finer resolution at high latitudes

Vertical resolution

50 unequally-spaced levels, dz=10 m at surface

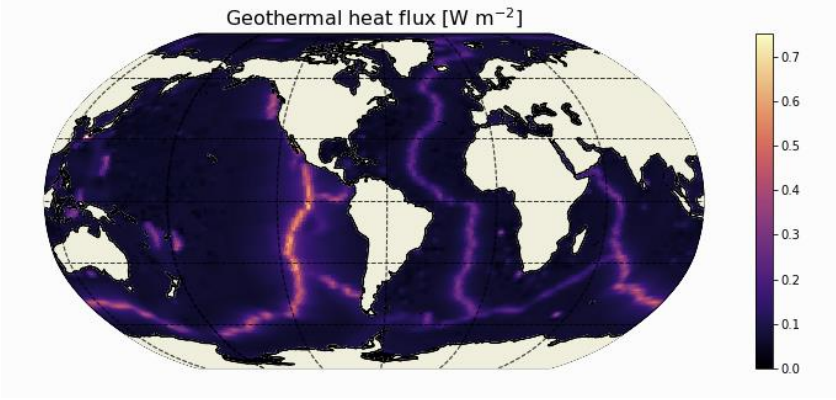
Time window:

1992-2022

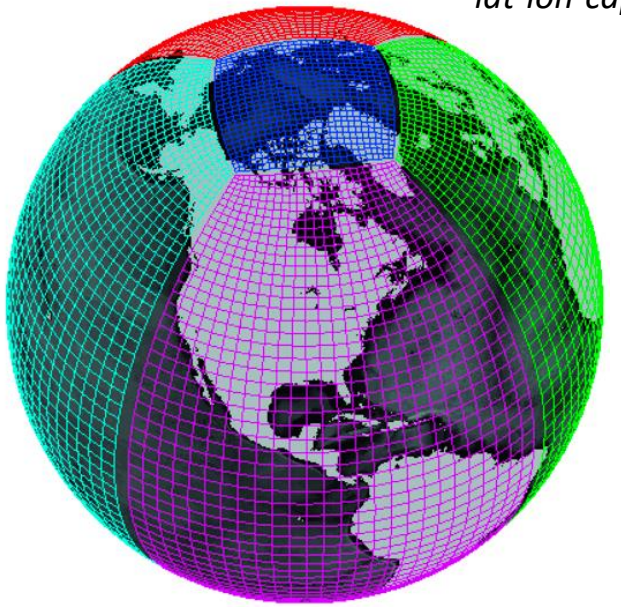
First-guess atmospheric state:

MERRA-2 from NASA GMAO (Gelaro et al., 2017).

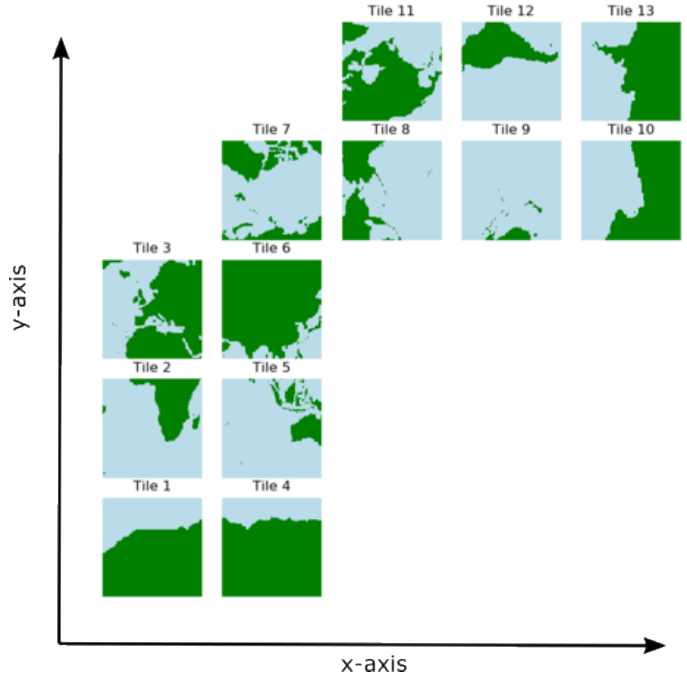
Geothermal heating:



"lat-lon-cap 90"



13 tiles of 90x90x50



ECCO V4 fields provided to the community

Monthly and daily mean:

Ocean and sea-ice

- $T, S, \vec{u}, \eta, \rho, \Phi$
- Horizontal and vertical fluxes of ocean volume, heat, salt, momentum
- Sea-ice and snow *h* and *c*
- Horizontal fluxes of sea-ice volume

Atmosphere

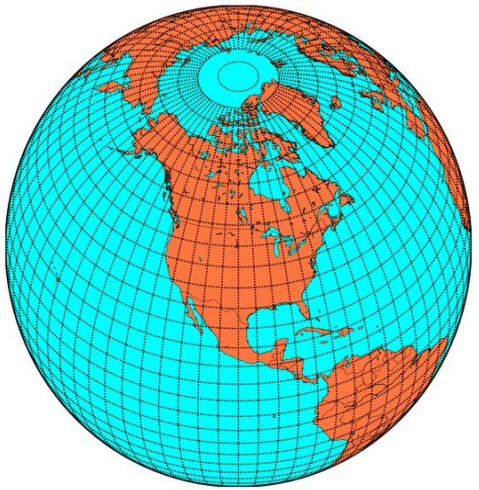
- Surface T, q, $|\vec{u}|$, $\vec{\tau}$, and downwelling radiative fluxes
- Air–sea ice–ocean fluxes of heat, moisture, energy, and momentum

Subgrid-scale mixing parameters

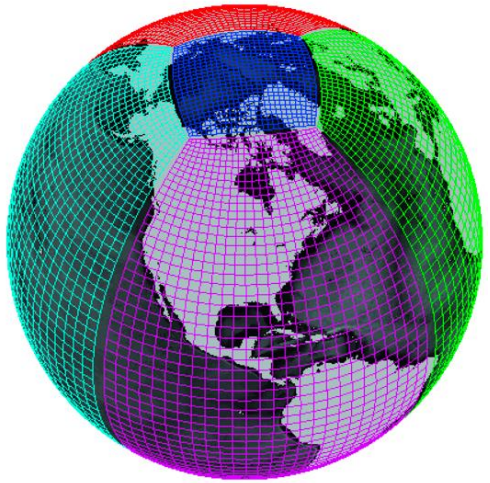
- 3D GM κ and Redi κ
- 3D vertical diffusivity

Fields are provided on two grids

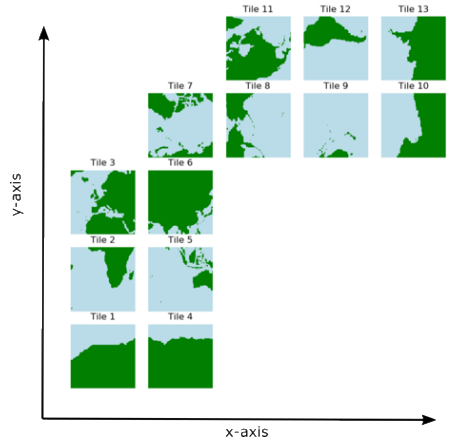
0.5° lat-lon



"lat-lon-cap 90"



13 tiles of 90x90x50



ECCO V4 Py: Python Package

An open-source resource to facilitate analysis of ECCO V4

Subroutines for:

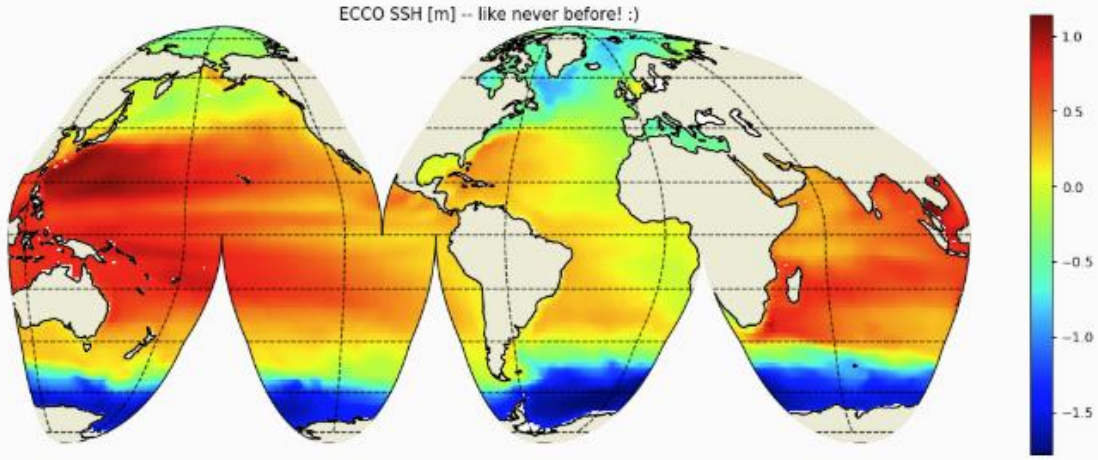
- Global mean sea level, ocean heat content, ocean volume
- Global and regional budgets for volume, heat, salt, salinity, and freshwater
- Volume, heat, and salt transports across arbitrary sections
- Meridional overturning circulation streamfunction
- Plotting llc-grid fields

```
[27]: plt.figure(figsize=(16,6), dpi=90)

      tmp_plt = ecco_ds.SSH.isel(time=1)
      tmp_plt = tmp_plt.where(ecco_ds.hFacC.isel(k=0) !=0)

      ecco.plot_proj_to_latlon_grid(ecco_ds.XC, ecco_ds.YC, \
                                    tmp_plt, \
                                    user_lon_0=-66,\
                                    projection_type='InterruptedGoodeHomolosine',\
                                    plot_type = 'pcolormesh', \
                                    show_colorbar=True,
                                    dx=1, dy=1);

      plt.title('ECCO SSH [m] -- like never before! :)');
```



https://github.com/ECCO-GROUP/ECCOv4-py

ECCO v4 Python Tutorial

http://ecco-v4-python-tutorial.readthedocs.io/

Welcome to the ECCO Version 4 Tutorial

This website contains a set of tutorials about how to use the ECCO Central Production Version 4 (ECCO v4) global ocean and sea-ice state estimate. The tutorials were written in Python and make use of the ecco_v4_py Python library, a library written specifically for loading, plotting, and analyzing ECCO v4 state estimate fields.

Additional Resources

The ECCO v4 state estimate is the output of a free-running simulation of a global ca. 1-degree configuration of the MITgcm. Prior to public release, the model output files model are assembled into NetCDF files. If you would like to work directly with the flat binary "MDS" files provided by the model then take a look at the xmitgcm Python package. The xgcm Python package provides tools for operating on model output fields loaded with xmitgcm. If you wish to analyze the MITgcm model output using Matlab then we strongly recommend the extensive set of tools provided by the gcmfaces toolbox.

The ecco_v4_py package used in this tutorial was inspired by both the xmitgcm package and gcmfaces toolbox.

Getting Started

- The ECCO Ocean and Sea-Ice State Estimate
- ECCO v4 state estimate ocean, sea-ice, and atmosphere fields
- Python and Python Packages
- How to get the ECCO v4 State Estimate
- Tutorial Overview

ECCO Version 4 Tutorial

Getting started
The ECCO Ocean and Sea-Ice State Estimate
ECCO v4 state estimate ocean, sea-ice, and atmosphere fields
Python and Python Packages
Using Python to Download ECCO Datasets
Downloading Subsets of ECCO Datasets
Using *wget* to Download ECCO Datasets from PO.DAAC
AWS Cloud: getting started and retrieving ECCO datasets
Tutorial Overview

ECCO data structures
The Dataset and DataArray objects used in the ECCOv4 Python package
Coordinates and Dimensions of ECCOv4 NetCDF files

Input/output, data structure manipulation
Loading the ECCOv4 native model grid parameters
Loading the ECCOv4 state estimate fields on the native model grid
ECCOv4 Loading llc binary files in the 'compact' format
Combining multiple Datasets
Saving Datasets and DataArrays to NetCDF

Operating on data variables
Accessing and Subsetting Variables
Operating on Numpy arrays

Plotting & interpolation
Plotting Tiles
Interpolating fields from the model llc grid to a regular lat lon grid

Scalar and vector calculations
Example calculations with scalar quantities
Calculating gradients and curl on the ECCO native grid

Intro to PO Tutorials
Intro to PO Tutorials: Getting Started
Part 1: Geostrophic balance
Part 2: Thermal Wind
Part 3: Steric height

More advanced calculations
Compute meridional heat transport
Compute MOC along the approximate OSNAP array from ECCO
ECCOv4 Global Volume Budget Closure
Global Heat Budget Closure
Salt, Salinity and Freshwater Budgets
Calculate ocean thermal forcing from ECCOv4r4 data, direct from PO.DAAC S3 storage

Search ⌘ + K

Welcome to ECCO HackWeek!

Details

Event Logistics

Schedule

Team

Participant Conduct

Preparation

What to Expect

Checklist

Intro to Open Science Studio and JupyterHub

Set up `git` access

Set up OSS with your hackweek repository

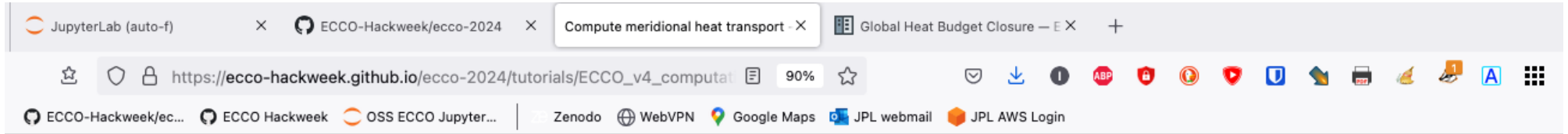
ECCO v4 computations

The ECCO version 4 Python Tutorial website has a wide range of tutorials to help users get started with accessing and using ECCOv4 output. These tutorials cover subjects ranging from loading data files using Python's *xarray* package, to more advanced computations like the steric height, meridional heat transport, and budgets.

Tutorials hosted in `ecco-2024`

A subset of these tutorials have been copied over to the `ecco-2024` Jupyter book and can be accessed below, to give you a sense of their range and structure. The tutorials hosted here already have settings enabled for cloud access, and will use the `efs_ecco` mounted drive for any downloads.

Tutorial	Topics
The Dataset and DataArray objects used in the ECCOv4 Python package	*xarray* intro with ECCOv4 output
Coordinates and Dimensions of ECCOv4 NetCDF files	Arakawa C-grid and ECCOv4 coordinates
Accessing and Subsetting Variables	*xarray* accessing and subsetting
Memory management in Python	Memory management when using *numpy*, *dask*, and *xarray*
Example calculations with scalar quantities	Sample calculations (e.g., SSH global mean)
Calculating gradients and curl on the ECCO native grid	Horizontal derivatives on the C-grid
Steric height	Steric height and thermosteric/halosteric components



Compute meridional heat transport

This notebook shows how to compute meridional heat transport (MHT) across any particular latitude band. Additionally, we show this for both global and basin specific cases.

Oceanographic computations:

- use xgcm to compute masks and grab values at a particular latitude band
- use ecco_v4_py to select a specific basin
- compute meridional heat transport at one or more latitude bands

Python basics on display:

- how to open a dataset using xarray (a one liner!)
- how to save a dataset using xarray (another one liner!)
- one method for making subplots
- some tricks for plotting quantities defined as dask arrays

Note that each of these tasks can be accomplished more succinctly with ecco_v4_py functions, but are shown explicitly to illustrate these tools. Throughout, we will note the ecco_v4_py (python) and gcmfaces (MATLAB) functions which can perform these computations.

Datasets to download

If you don't have any of the following datasets already, you will need to download them to complete the tutorial. Aside from the grid geometry file (which has no time dimension), you will need the monthly datasets for the full time span of ECCOv4r4 output (1992 through 2017). The ShortNames of the datasets are:

- **ECCO_L4_GEOMETRY_LLC0090GRID_V4R4**
- **ECCO_L4_OCEAN_3D_TEMPERATURE_FLUX_LLC0090GRID_MONTHLY_V4R4** (1992-2017)

Available ECCO fields for the EHW?

- ECCO V4r4 1992-2017: PO.DAAC AWS S3 Bucket
  - Daily and monthly means, snapshots, native and lat-lon grid
- ECCO V4r5 1992-2020: ECCO AWS S3 Bucket
  - Monthly means, and snapshots, native grid
- ECCO V4r5 extension 2020-2024: ECCO AWS S3 Bucket
  - Monthly means, and snapshots, native grid

ECCO Modeling Utilities (EMU)

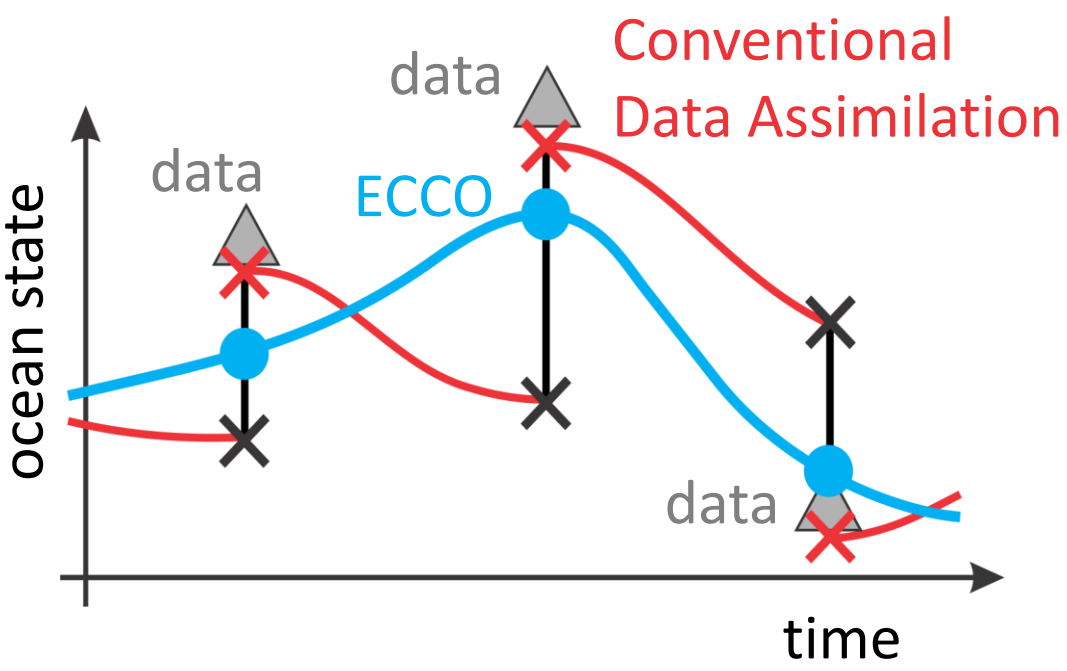
A set of menu-driven tools
for analyzing the ECCO model.

Ichiro Fukumori, Ou Wang, Ian Fenty
Jet Propulsion Laboratory, California Institute of Technology

14 October 2024, ECCO HackWeek @ KISS/Caltech

ECCO Modeling Utilities (EMU)

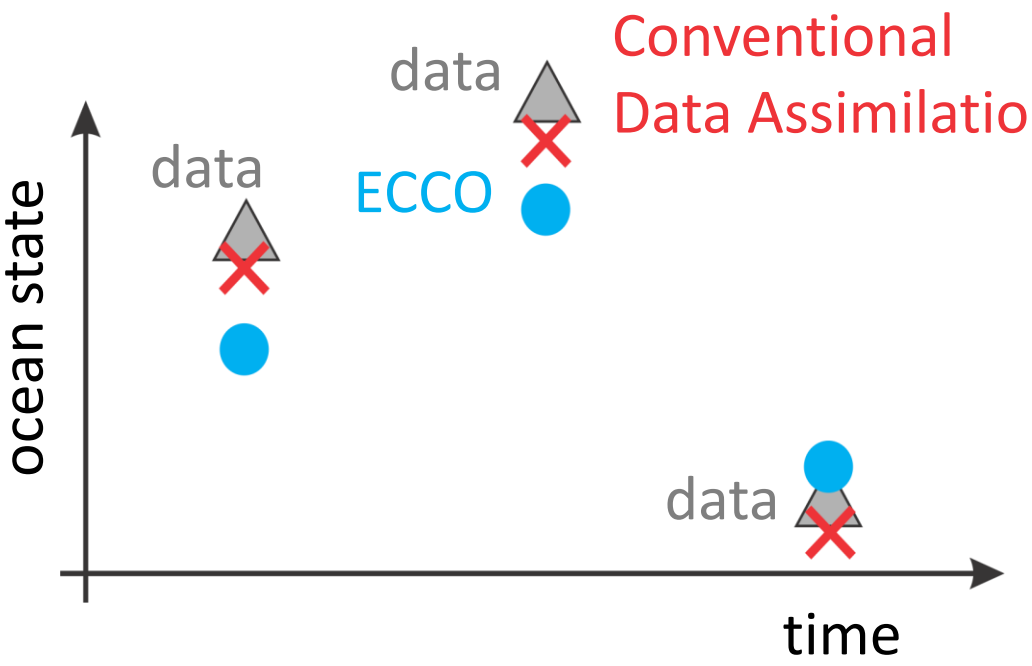
Why? A set of menu-driven tools for analyzing the ECCO model.



1) ECCO is a state-of-the-art synthesis of ocean observations,
2) ECCO is characterized by its physical consistency (e.g., closed budgets),
3) The physics of this consistency is embodied in the model,
4) EMU permits analyses of the model's physics (e.g., causation) without needing modeling expertise.

ECCO Modeling Utilities (EMU)

Why? A set of menu-driven tools for analyzing the ECCO model.

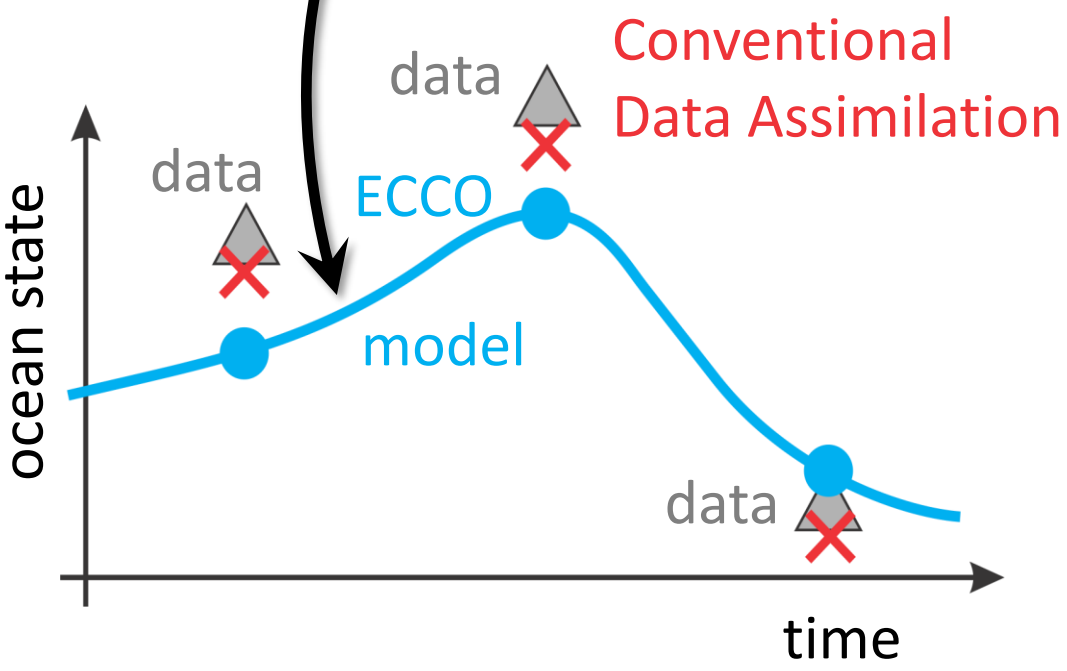


1) ECCO is a state-of-the-art synthesis of ocean observations,
2) ECCO is characterized by its physical consistency (e.g., closed budgets),
3) The physics of this consistency is embodied in the model,
4) EMU permits analyses of the model's physics (e.g., causation) without needing modeling expertise.

ECCO Modeling Utilities (EMU)

This is Why!

A set of menu-driven tools for analyzing the ECCO model.



1) ECCO is a state-of-the-art synthesis of ocean observations,
2) ECCO is characterized by its physical consistency (e.g., closed budgets),
3) The physics of this consistency is embodied in the model,
4) EMU permits analyses of the model's physics (e.g., causation) without needing modeling expertise.

ECCO Modeling Utilities (EMU)

A set of menu-driven tools
for analyzing the ECCO model.

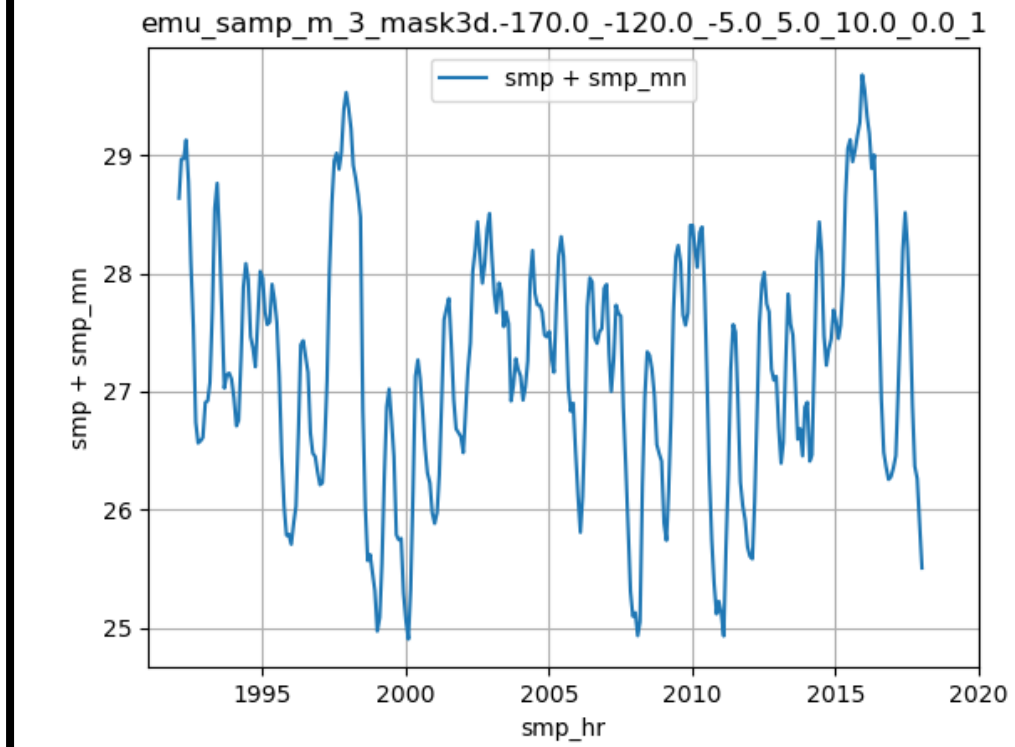
	Tool	Description
1	**Sampling**	Evaluates time-series of model state.
2	**Forward Gradient**	Computes model's response to change in forcing.
3	**Adjoint**	Computes model's sensitivity to forcing (adjoint gradient).
4	**Convolution**	Evaluates convolution of adjoint gradients with forcing (adjoint gradient decomposition).
5	**Tracer**	Computes evolution of passive tracer and its adjoint.
6	**Budget**	Evaluates contributions to a quantity's budget.
7	**Modified Simulation**	Reruns simulation with changes (e.g., forcing).
8	**Attribution**	Evaluates effects by control type.

ECCO Modeling Utilities (EMU)

A set of menu-driven tools
for analyzing the ECCO model.

- ✓ Evaluates time-series of the model state
- ✓ Options to use latitude, longitude, depth as criteria
- ✓ Useful for assessing fidelity of ECCO

Nino3.4 SST



OBJF

Scale Weight Variable

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

Multi-variable Space

	Tool	
1	Sampling	
	ward Gradient	Computes model's re
		l's s
		lutio
		dec
		tion of passive tracer and its adjoint.
		butions to a quantity's budget.
		on with changes (e.g., forcing).
		s by control type.

ECCO Modeling Utilities (EMU)

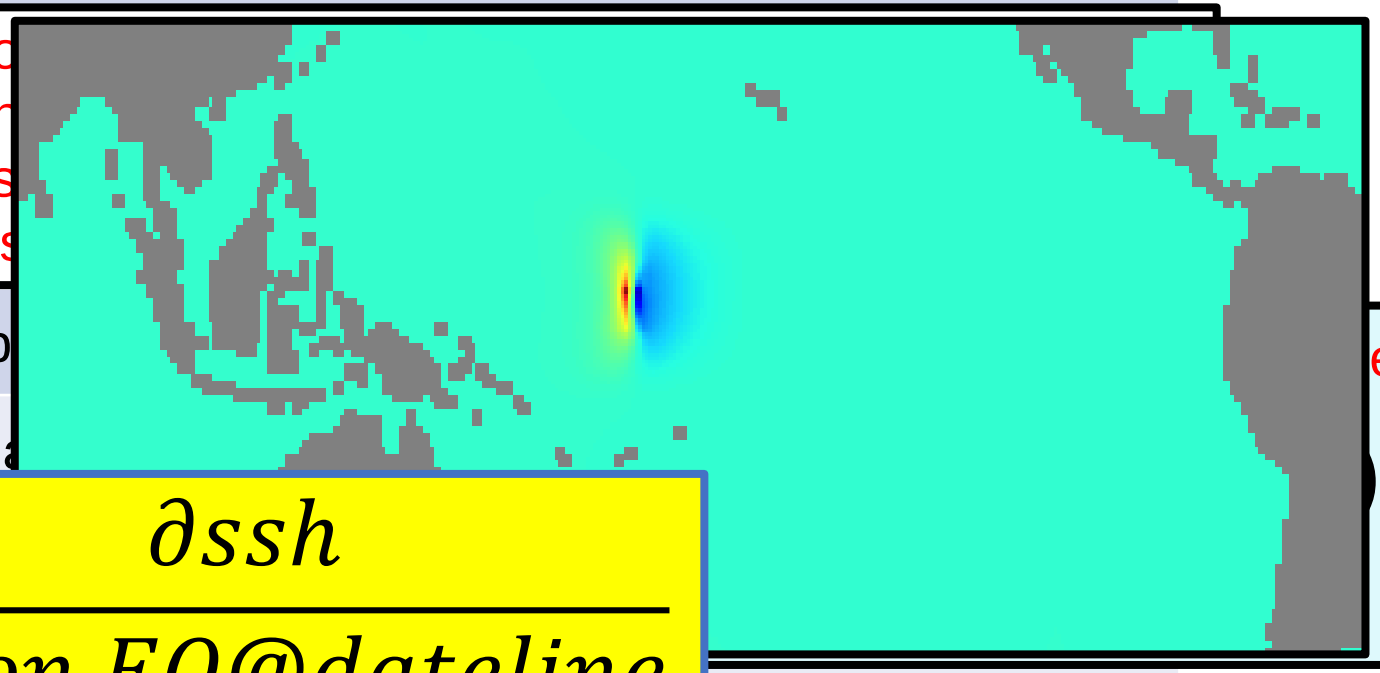
A set of menu-driven tools
for analyzing the ECCO model.

	Tool	
1	**Sampling**	
2	**Forward Gradient**	
3	**Adjoint**	
4	**Convolution**	
5	**Tracer**	Comp
6	**Budget**	Evalu
7	**Modified Simulati**	
8	**Attributi**	

- ✓ Computes forward gradient
- ✓ Choice of denominator
- ✓ Useful for insight into physics
- ✓ Also useful for validating adjoint

$$\frac{\partial\,(\text{model state})}{\partial\,(\text{control})}$$

- ✓ Co
- ✓ Ch
- ✓ Us
- ✓ Als

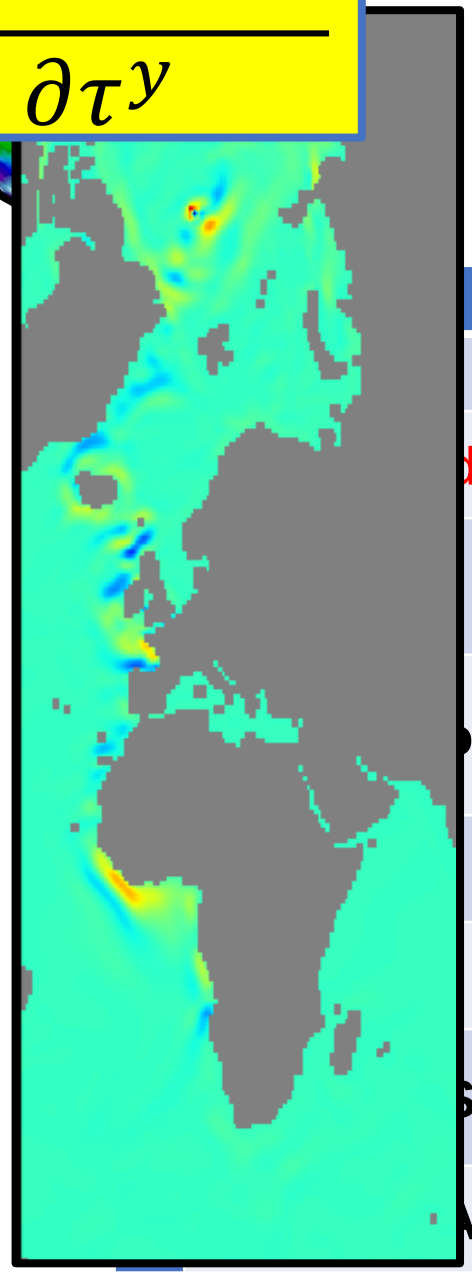


$$\frac{\partial ssh}{\partial \tau^x \text{ on } EQ@dateline}$$

$$\frac{\partial OBP@NP}{\partial \tau^y}$$

CCO Modeling Utilities (EMU)

A set of menu-driven tools
for analyzing the ECCO model.



Tool	
Sampling	
d Gradient	✓ Computes forward gradient ✓ Choice of denominator ✓ Useful for insight into physics ✓ Also useful for validating adjoint $\frac{\partial (\text{model state})}{\partial (\text{control})}$
Adjoint	✓ Computes adjoint gradient ✓ Choice of numerator ✓ Useful for insight into physics ✓ Also useful for model calculus $\frac{\partial (\text{model state})}{\partial (\text{control})}$
nvolution	
Tracer	Computes evolution
Budget	Evaluates contributio
imulation	Reruns simulation w
ttribution	Evaluates effects by control type.

OBJF

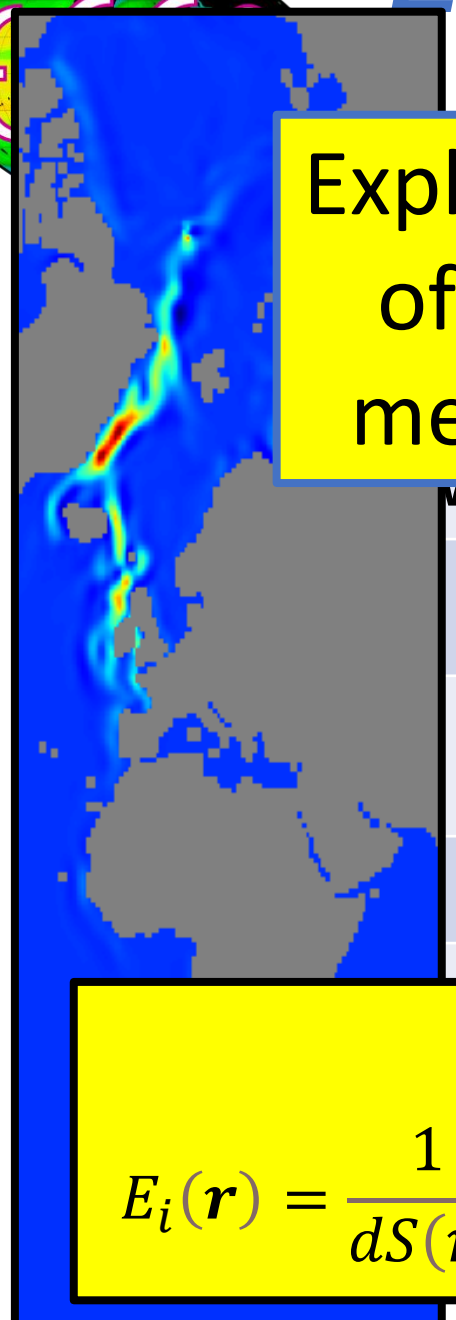
Scale Weight Variable

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

Multi-variable Space

ECCO Modeling Utilities (EMU)

A set of menu-driven tools
g the ECCO model.



Explained variance of OBP@NP by meridional wind

	on
	time-series of model state.
ward Gradient	Computes model's response to change in forcing.
Adjoint	
Convolution	
Tracer	

✓ Expands quantities of interest into their controls

$$\delta J(t) \approx \sum_{i}^{\text{type}} \sum_{\mathbf{r}}^{\text{location}} \sum_{\Delta t}^{\text{lag}} \frac{\partial J(t_g)}{\partial \phi_i(\mathbf{r}, t_g - \Delta t)} \delta\phi_i(\mathbf{r}, t - \Delta t)$$

adjoint gradient

✓ Options to use modified gradients & forcing

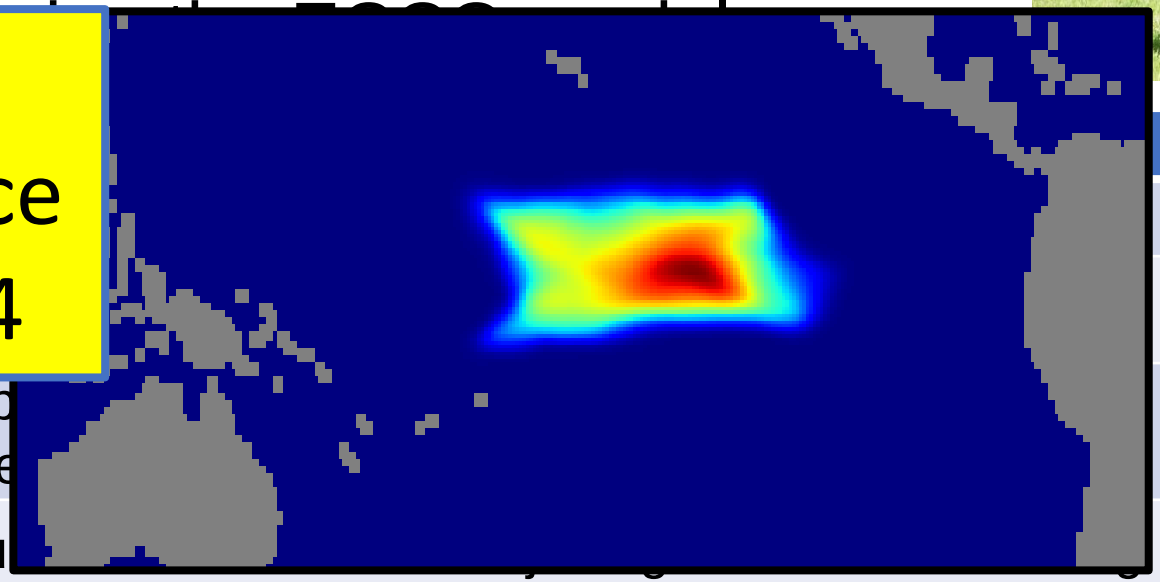
✓ Useful in identifying causal mechanisms

$$E_i(\boldsymbol{r}) = \frac{1}{dS(\boldsymbol{r})}\left[1 - \frac{var\left(\delta J(t) - \Sigma_{\Delta t} \frac{\partial J(t_g)}{\partial \phi_i(\boldsymbol{r}, t_g - \Delta t)} \delta\phi_i(\boldsymbol{r}, t - \Delta t)\right)}{var(\delta J(t))}\right]$$

ECCO Modeling Utilities (EMU)

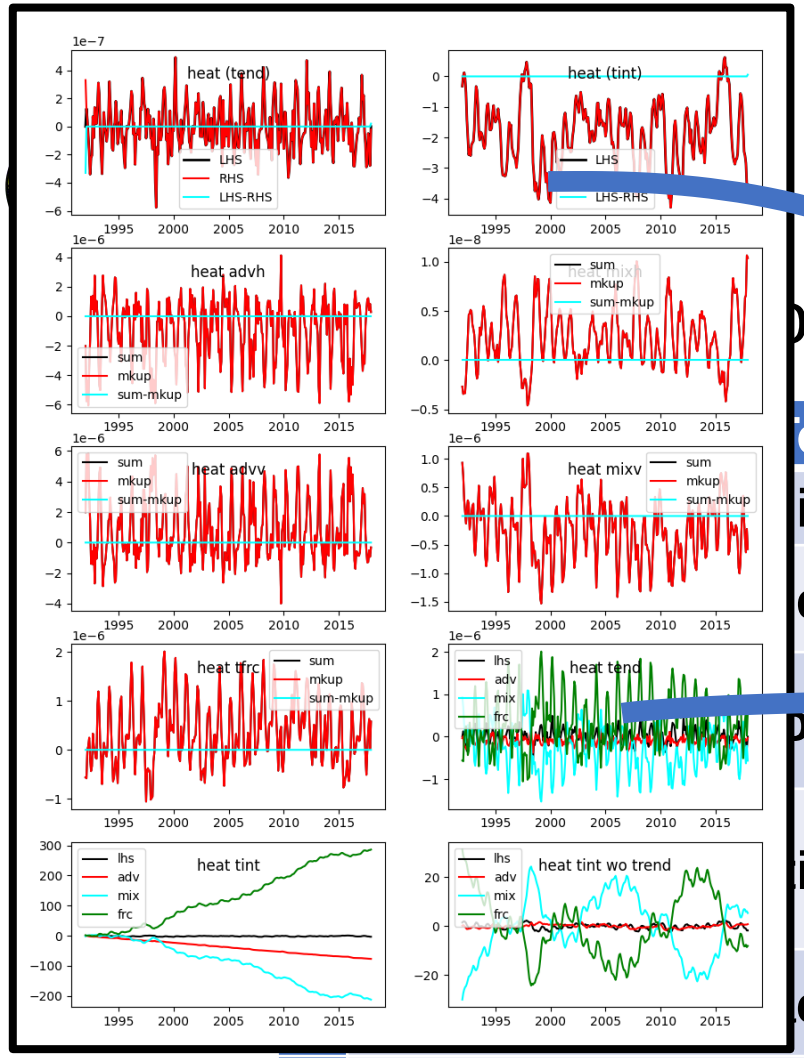
A set of menu-driven tools

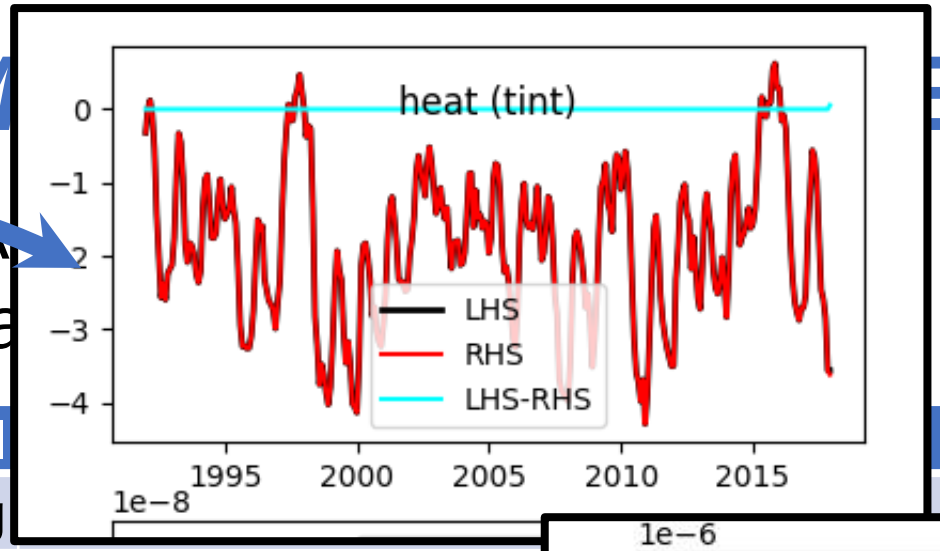
Fate of water occupying surface layer of Nino3.4

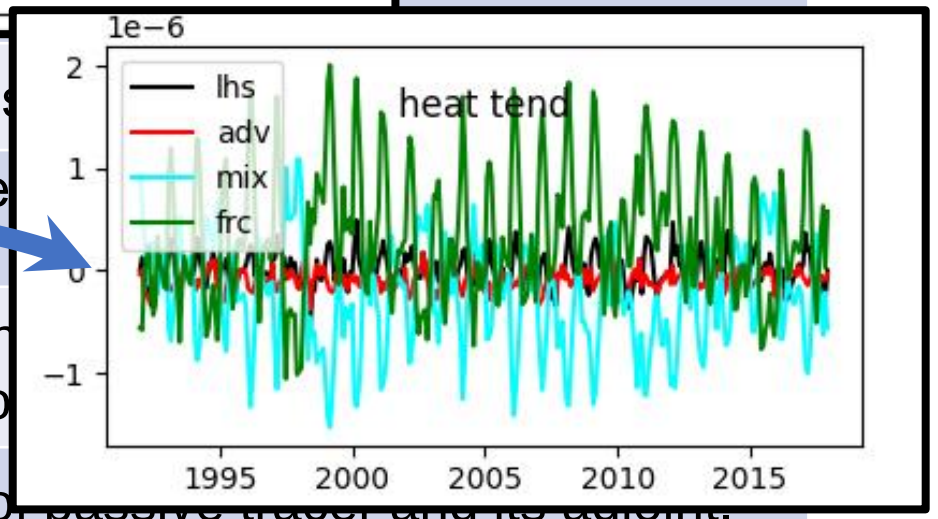


1		
2		
3	**Adjoint**	Comp… gradie…
4	**Convolution**	Evalu…
5	**Tracer**	
6	**Budget**	Evaluates contributions to a quantity's budget.
7	**Modified Simulation**	Reruns simulation with changes (e.g., forcing).
8	**Attribution**	Evaluates effects by control type.

- ✓ Computes evolution of passive tracer and its adjoint
- ✓ Useful in identifying fate & origin of water masses and their circulation pathways







	ing	
	ent	Computes model's res
	oint	Computes model's se gradient).
	ion	Evaluates convolution (adjoint gradient deco
	cer	Computes evolution o
	get	
	ion	Reruns simulation with changes (e.g., forcing).
8	**Attribution**	Evaluates effects by control type.

✓ Evaluates property budgets
✓ Useful in analyzing controlling processes

Nino3.4
SST budget

ECCO Modeling Utilities (EMU)

A set c
for analy

	Tool	Descr
1	**Sampling**	Evalu
4	**Convolution**	Evalu (adjoi
5	**Tracer**	Comp
6	**Budget**	Evalu
7	**Modified Simulation**	
8	**Attribution**	Evaluates effects by control type.

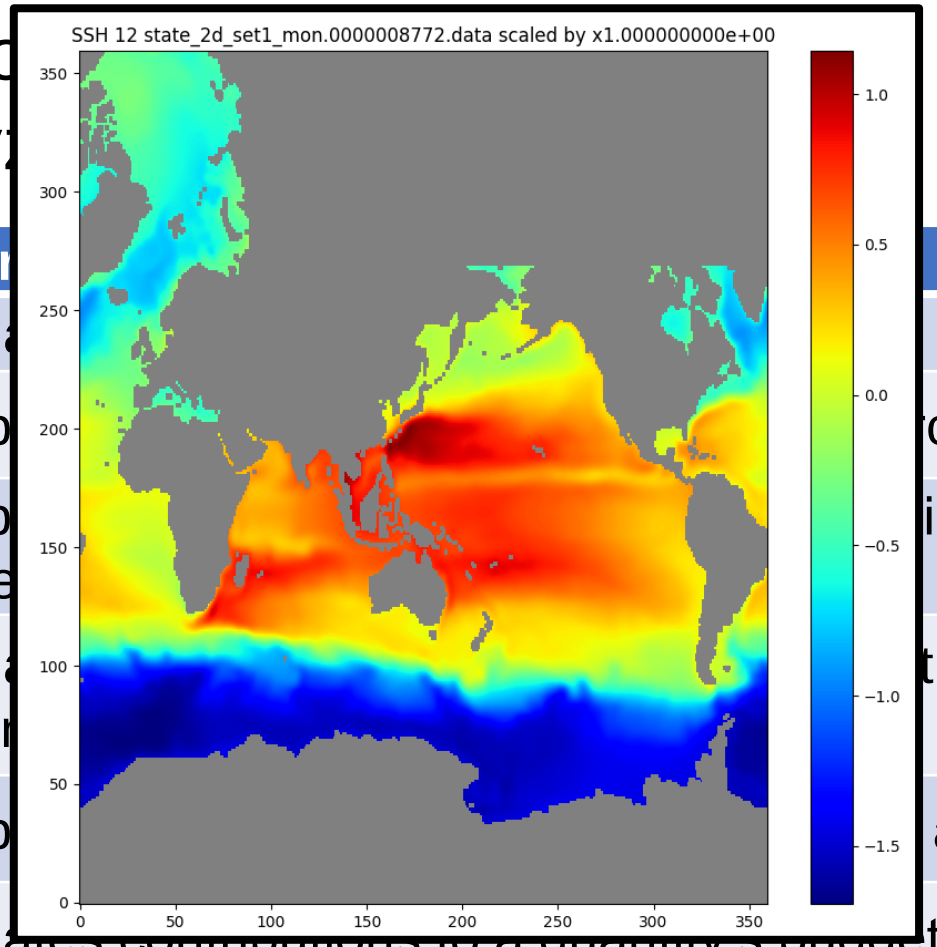
cing.

int

th forcing

adjoint.

SSH w/
time-mean wind



- ✓ Reruns model with changes (e.g., forcing)
- ✓ Useful in conducting experiments with the model
- ✓ Also useful in producing non-standard output

ECCO Modeling Utilities (EMU)

A set of menu-driven tools
for analyzing the ECCO model.

	Tool	Description
1	**Sampling**	Evaluates time-series of model state.
2	**Forward Gradient**	Computes model's response to change in forcing.
3	**Adjoint**	Computes model's s… gradient).
4	**Convolution**	Evaluates convolutio… (adjoint gradient dec…
5	**Tracer**	Computes evolution …
6	**Budget**	
7	**Modified Simulation**	
8	**Attribution**	

OBJF

Scale Weight Variable

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

Multi-variable Space

- ✓ Evaluates model's variation by control type
  e.g., effect of variable wind = ECCO_Solution – Modified_Solution_w/o_Variable_Wind
- ✓ Useful in identifying responsible control type
- ✓ Also useful in validating results of Convolution Tool

ECCO Modeling Utilities (EMU)

A set of menu-driven tools
for analyzing the ECCO model.

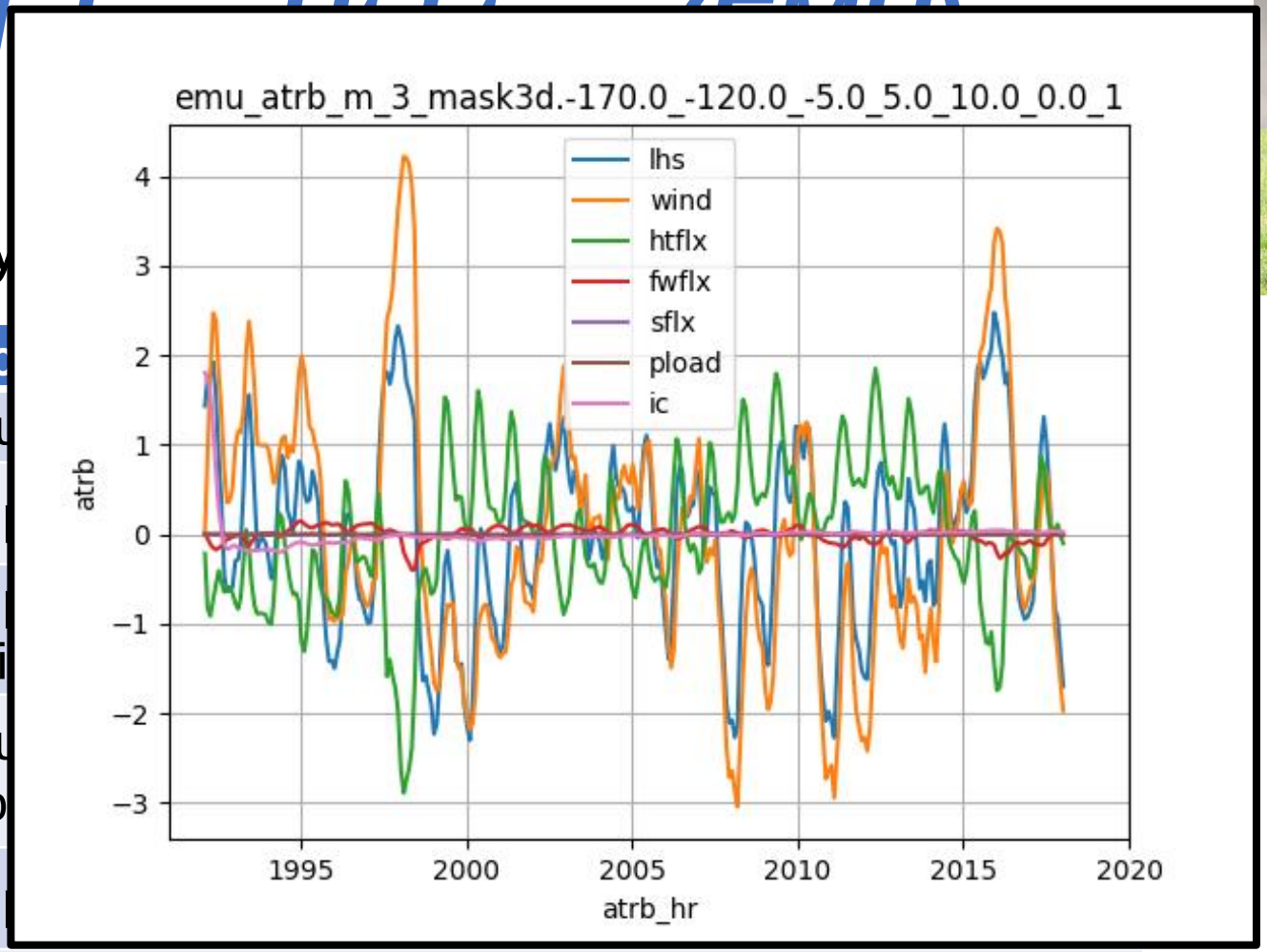
	Tool	Description
1	**Sampling**	Evaluates time-series of model state.
2	**Forward Gradient**	Computes model's response to change in forcing.
3	**Adjoint**	✓ Expands quantities of interest into their controls
4	**Convolution**	$\delta J(t) \approx \sum_{i}^{\text{type}} \sum_{\mathbf{r}}^{\text{location}} \sum_{\Delta t}^{\text{lag}} \frac{\partial J(t_g)}{\partial \phi_i(\mathbf{r}, t_g - \Delta t)} \delta\phi_i(\mathbf{r}, t - \Delta t)$ adjoint gradient
5	**Tracer**	✓ Options to use modified gradients & forcing ✓ Useful in identifying causal mechanisms
6	**Budget**	✓ Evaluates model's variation by control type *e.g., effect of variable wind = ECCO_Solution – Modified_Solution_w/o_Variable_Wind*
7	**Modified Simulation**	✓ Useful in identifying responsible control type
8	**Attribution**	✓ Also useful in validating results of Convolution Tool

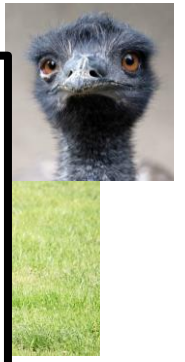
ECCO Mod

A set
for analy

	Tool	Desc
1	**Sampling**	Evalu
4	**Convolution**	Evalu (adjo
5	**Tracer**	Com
6	**Budget**	
7	**Modified Simulation**	
8	**Attribution**	

Nino3.4
SST attribution





- ✓ Evaluates model's variation by control type
  e.g., effect of variable wind = ECCO_Solution – Modified_Solution_w/o_Variable_Wind
- ✓ Useful in identifying responsible control type
- ✓ Also useful in validating results of Convolution Tool

ECCO Modeling Utilities (EMU)

How? A set of menu-driven tools for analyzing the ECCO model.

1) To run EMU, enter command

   emu

2) To read & plot EMU results, run python script

   emu_plot.py

3) To install EMU on your own system, run shell script

   emu_setup.sh

(available as source code & container image)

ECCO Modeling Utilities (EMU)

How can we make EMU more useful?

- What works?
- What doesn't work?
- What's missing?