

# Digital Twins for Ocean Robots

Gaël Forget<sup>1</sup>

<sup>1</sup>Earth, Atmospheric and Planetary Sciences, Massachusetts Institute of Technology,  
77 Massachusetts Avenue, Cambridge, 02139, MA, USA

## ABSTRACT

DTOR is a framework to access, analyze, and simulate the global fleet of ocean observing devices (or *ocean robots*) that monitor climate change. It brings these observations to Julia and lets us pair ocean robots with virtual counterparts (or *twins*). Digital twins provide a bridge from the data to predictive models and enable machine learning. In turn, observing system simulations in a digital environment can help evaluate observational strategies a priori, during deployment, or afterwards. In this paper we present the DTOR framework, its supported observing systems, capabilities to simulate ocean robots, and envisioned applications.

## Keywords

Julia, Ocean, Climate, Robot, Observation, Observing, Platform, Sensor, Drifter, Buoy, Profiler, Model, Modeling, Machine Learning, Artificial Intelligence, Data Assimilation, Parameter Inference, Climatology, Geospatial, Statistics

## 1. Introduction

The DTOR framework was introduced at the Symposium on Advances in Ocean Observation in 2022, and later presented to US-CLIVAR and JuliaCon in 2023, and the OceanPredict Symposium in 2024. The primary goal of DTOR is to associate ocean observing systems (or *ocean robots*) with virtual counterparts (or *twins*), and do it for the whole fleet of ocean robots that are currently at sea or have observed the Ocean in the past (Fig. 1). The virtual twins are to be created through numerical model simulations, which can come in different flavors and languages. The simulation of observing systems in the future, as climate change progresses, is also part of the scope of the DTOR project. At JuliaCon 2023, we presented a solution to (1) access and manipulate the data collected by ocean robots (inc. OCEANROBOTS.JL and ARGODATA.JL), and (2) simulate such observations in a digital environment using a hierarchy of models (incl. CLIMATEMODELS.JL and MITGCM.JL).

The model hierarchy includes fast climate model emulators, global model output, ocean reanalyses, high-resolution models, and several ways to represent marine ecosystems. Through a streamlined workflow, CLIMATEMODELS.JL (JuliaCon 2021, 2023) makes it easy to operate these models that can provide a digital environment for the virtual ocean robot fleet to observe and navigate. DRIFTERS.JL [9] can be used to predict pathways of ocean robots that tend to follow ocean currents. MESHARRAYS.JL adds basic geospatial support for global climate model grids, providing capabilities such as interpolation and geolocation on a grid. The framework leverages and links to a variety of highly capable Julia pack-

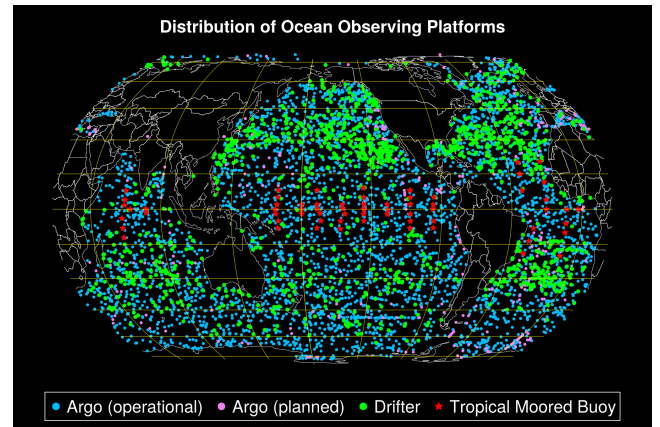


Fig. 1: Ocean robots that were collecting data on 2024/11/29 for three types of observations (blue, green, and red dots). Planned deployments of Argo profilers are also shown (purple dots). Data is from <https://www.ocean-ops.org> and queried via OCEANROBOTS.JL.

ages from the community. It notably provides user-friendly plotting recipes via its MAKIE.JL extension, and tutorial examples in the form of PLUTO.JL notebooks.

## 2. Robots Observing the Ocean

Part of the fleet of ocean robots currently at sea is depicted in Fig. 1, focusing on a few widely used observing platforms. To create this map, OCEANROBOTS.JL queries the meta-data-base from OceanOPS.org, which keeps track of the whole ocean fleet of scientific observing platforms in real time, and provides a RESTful API. The OCEANROBOTS.JL package provides an interface to this API via the ‘OceanOPS’ module. It then defines data structures such as ‘SurfaceDrifter’ and ‘ArgoFloat’ to access and utilize data from widely used ocean observing platforms. The list of ocean observing platforms currently supported by OCEANROBOTS.JL is in Tab. 1. A fraction of the geospatial data obtained by surface ocean drifters in the past (green dots in Fig. 1) is depicted in Fig. 2 using OCEANROBOTS.JL and MAKIE.JL. These drifting buoys tend to follow ocean currents at approximately 15 meters depth. They can measure sea surface temperature and sea level pressure along their trajectory. The data archive currently holds 19396 drifter trajectories, collected over the past 50 years, which allow us to create climatologies such as the one shown in Fig. 3.

Argo profiling floats (blue dots in Fig. 1; Code 1) are one of our main tools to monitor global warming below the sea surface. These devices provide a less detailed view of oceanic pathways (Fig. 4, top panels) than surface drifters do (e.g., Fig. 2), since Argo floats

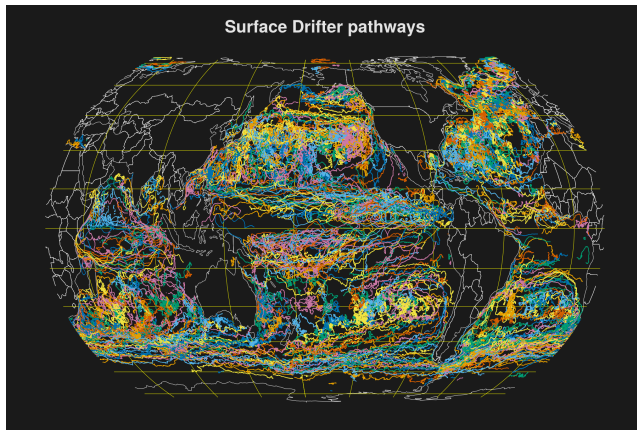


Fig. 2: A few of the ocean drifters (5% of 19396) that have been deployed to follow near surface ocean currents in the past.

Table 1.: Ocean observing systems targeted by OCEANROBOTS.JL, and associated data structures (or blank if not yet implemented).

Platform Type	Data Structure
surface drifters	<i>SurfaceDrifter, CloudDrift</i>
drifting profilers	<i>ArgoFloat</i>
moored buoys	<i>OceanSite, NOAAbuoy</i>
sea gliders	<i>Gliders</i>
expendable bathy-thermographs	
sail drones	
research vessel data	<i>CCHDO</i>
ships of opportunity data	
marine mammals data	

only report their location once every ten days. However, Argo floats have a crucial diving capability that surface drifters don't have – they go up and down the water column to measure temperature and salinity (T,S; bottom panels of Fig. 4). The advent of the Argo array in the early 2000s [19] thus opened up a whole new era of geospatial analysis, state estimation, and parameter inference over the global Ocean [8, 4, 15, 5, 6, 7, 10]. On 2024/05/27 for example there were 3837 Argo floats at sea, and a total of 18730 in the Argo data base. Since the Argo array is such an important observing system, a dedicated package called ARGODATA.JL was created that OCEANROBOTS.JL uses under the hood.

Code 1: Download and visualize one Argo float data as in Fig. 4.

```
using OceanRobots, CairoMakie;
argo=read(ArgoFloat(), wmo=6900900);
fig=plot(argo, option=:standard)
```

OCEANROBOTS.JL brings these key climate data sets to the Julia community. It provides a bridge to climate scientists working on observations and models, who are interested in leveraging the powerful Julia software ecosystem (e.g., for numerical modeling, machine learning, and statistical analysis), and have much expertise to contribute. The development of OCEANROBOTS.JL aims to help advance (1) how we understand and simulate observations that monitor climate change in the oceans, and (2) climate literacy and education by providing simple apps that anyone can use.

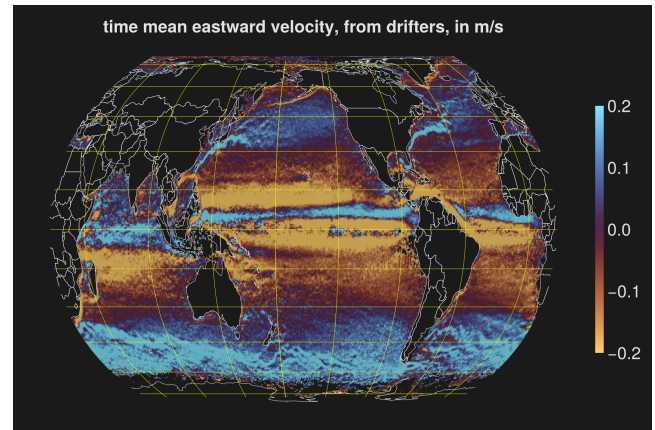


Fig. 3: Sample mean eastward velocity estimated from drifter data, with a grid resolution of 1/2 degree.

### 3. Simulating Ocean Robots

Our ability to simulate the data produced by ocean robots is directly linked to our ability, or lack thereof, to explain observed variations and decipher mechanisms from data. Numerical modeling is thus often motivated and driven by observations. Many research activities involve combining ocean models and observations, with the typical goal of learning model parameters, statistics, and dynamics from data [4, 5, 6, 7, 10]. Observing system simulations in a digital environment (using models) enable a wide range of applications – to test out innovative platform or sensor designs ahead of deployment, to optimize global monitoring strategies, or to guide deployed assets in real time for example.

The two simple examples presented below illustrate the simulation of (1) environmental sensors and (2) observing platforms. First, in Fig. 5 we simulate an Argo data collection for temperature (T) and salinity (S) by sampling an ocean climatology along the track of the Argo float (the one from Fig. 4). This calculation requires (1) a model prediction of  $T(x, y, z, t)$  and  $S(x, y, z, t)$ , which can be based on statistical or mechanistic models on a grid, and (2) tools that deal with the Earth geography, and can localize observations on a grid, and perform interpolation tasks.

To create Fig. 5, we use CLIMATOLOGY.JL to access the OCCA climatology [4], and then interpolate  $T(x, y, z, t)$ ,  $S(x, y, z, t)$  via geospatial tools from MESHARRAYS.JL. The OCCA climatology is a previously trained model that consists of 12 monthly three-dimensional fields of  $T, S$  (one per calendar month). The fact that Figs. 4 and 5 show broadly similar patterns reflects that the predictive model is skillful, and that  $T, S$  contrasts seen in Fig. 4 largely result from the sensor moving with the Argo float across the Ocean's  $T, S$  geography. Differences between Figs. 4 and 5 can in turn provide a basis for improving the predictive model, for example by including time variability beyond the seasonal cycle, or through various methods for data assimilation, artificial intelligence, or parameter inference [4, 5, 6, 7, 11].

Our second example focuses on the simulation of observing platforms moving with ocean currents. This basic simulation of surface drifter pathways uses the geostatistical average of velocities calculated in Fig. 3 as a predictive model for the  $U(x, y, t)$ ,  $V(x, y, t)$  velocity fields. In Code 2, this gridded data set is provided as input to DRIFTERS.JL, which then releases virtual drifters and calculates their trajectories following the flow field (Fig. 7). While this simple drifter trajectory prediction model neglects stochastic variability and small scales in ocean currents altogether (Fig. 8), it is

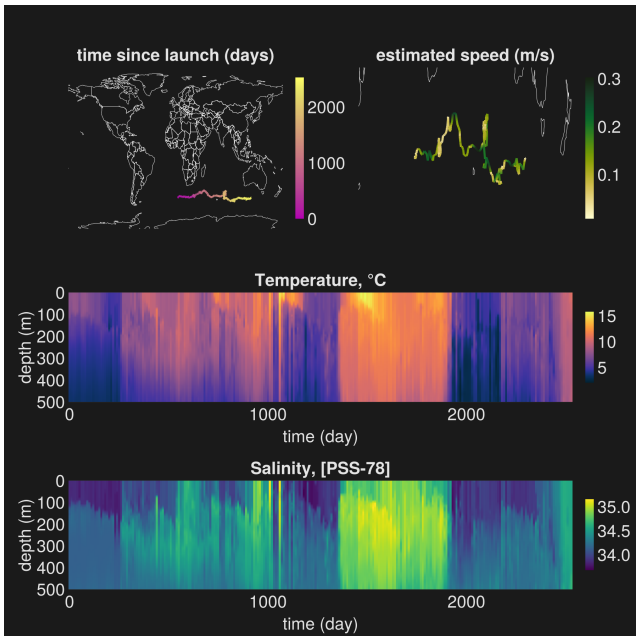


Fig. 4: Data collected by a profiling float from the Argo array. Temperature and salinity profiles, taken every ten days, extend to 2000m depth.

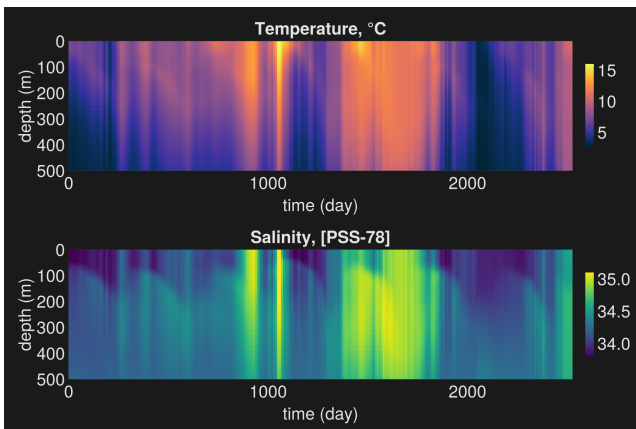


Fig. 5: Virtual Argo profiles predicted using the OCCA climatology, interpolated to the positions of the float shown in Fig. 4.

sufficient to capture the main mean pathway of sea water through the Gulf of Mexico region – the well-known loop current feeding the Gulf Stream via Florida Strait (Fig. 7).

Code 2: Simulate surface drifter pathways in the Gulf of Mexico (Fig. 6) from flow fields  $u, v$  and initial positions  $x, y$ . Visualization code is provided in the DRIFTERS.JL documentation.

```
using OceanRobots, Drifters, CairoMakie ;
P=Drifters.Gulf_of_Mexico_setup();
F=FlowFields(u=P.u,v=P.v,period=P.T);
I=Individuals(F,P.x0,P.y0);
[solve!(I,P.T .+P.dT*(n-1)) for n in 1:P.nt];
```

Much more could be done to improve the details and skill of the simple models used here. Higher-order and higher-resolution

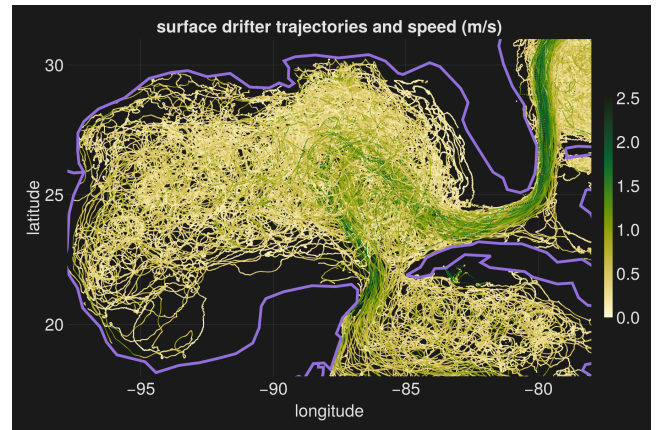


Fig. 6: Drifter trajectories in the Gulf of Mexico region. Large velocities highlight the path of the Gulf Stream, being fed by the loop current.

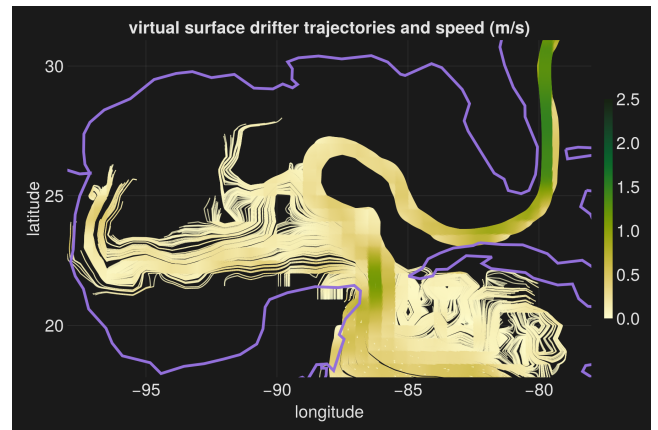


Fig. 7: Virtual drifter trajectories predicted using (1) just the climatological mean flow field estimated from drifters (Fig. 3, plus the corresponding northward component) and (2) DRIFTERS.JL to calculate trajectories.

model output are available to represent important aspects of what is being observed, but neglected in Figs. 5 and 7, such as the turbulent dispersion seen in Fig. 6 or the small-scale variations visible in Fig. 4. Small scale temperature fronts and currents are present everywhere in the ocean, with a lot of heterogeneity across regions, as illustrated in Fig. 8. Hence it is important to include global km-scale ocean simulations in our model hierarchy (see section 4).

#### 4. Digital Twin Framework

Digital twins can be defined as per the U.S. Committee on Foundational Research Gaps and Future Directions for Digital Twins [16]:

**DEFINITION 1.** A digital twin is a set of virtual information constructs that mimics the structure, context, and behavior of a natural, engineered, or social system (or system-of-systems), is dynamically updated with data from its physical twin, has a predictive capability, and informs decisions that realize value. The bidirectional interaction between the virtual and the physical is central to the digital twin.

In the digital twins for ocean robots framework (DTOR), interactivity is facilitated by Julia and its large ecosystem of software packages. In particular, MAKIE.JL [2] and PLUTO.JL [22] let us op-

erate even complex modeling workflows from notebooks and apps. The tutorial examples in OCEANROBOTS.JL, ARGODATA.JL, CLIMATEMODELS.JL, and MITGCM.JL [11] are notably available as Pluto notebooks. Interaction between model and data is further enabled by the rich Julia ecosystem for machine learning, data assimilation, parameter inference, etc. Fig. 9 provides an example where FLUX.JL [14] is used to train neural networks to predict chlorophyll concentration (linked to marine microbe abundance) from environmental parameters as done in the CANYON model [21, 1].

The core of DTOR is formed by OCEANROBOTS.JL, ARGODATA.JL, and MESHARRAYS.JL for the physical twins, along with CLIMATOLOGY.JL, CLIMATEMODELS.JL, MITGCM.JL, and DRIFTERS.JL for the virtual twins (i.e., predictive modeling). The CLIMATEMODELS.JL interface streamlines the use of models implemented in various languages. The model hierarchy already includes models listed in this paper, and is easy to extend via the CLIMATEMODELS.JL interface. DTOR can also simulate observations in the future based on climate model predictions (Fig. 10), and CLIMATEMODELS.JL provides two options for this kind of applications – either querying the CMIP6 archive of model output [3] or using a fast emulator such as Hector [13].

Within our model hierarchy, future climate scenarios like Fig. 10 can be downscaled using gridded climatologies and reanalyses [4, 5, 10]. The ECCO4 and OCCA2 reanalyses are simple to rerun with perturbed surface forcing fields via MITGCM.JL, which is particularly convenient for such applications. Multi-decadal solutions produced in this way can then be further downscaled using km-scale model output (Fig. 8). At the end of this modeling workflow, OCEANROBOTS.JL, MESHARRAYS.JL and DRIFTERS.JL enable calculations such as Figs. 5 and 7, for any of our models. DTOR can also take advantage of gridded satellite data, incl. sea surface temperature and sea level anomalies via CLIMATOLOGY.JL, which can sometimes be used instead of ocean reanalyses.

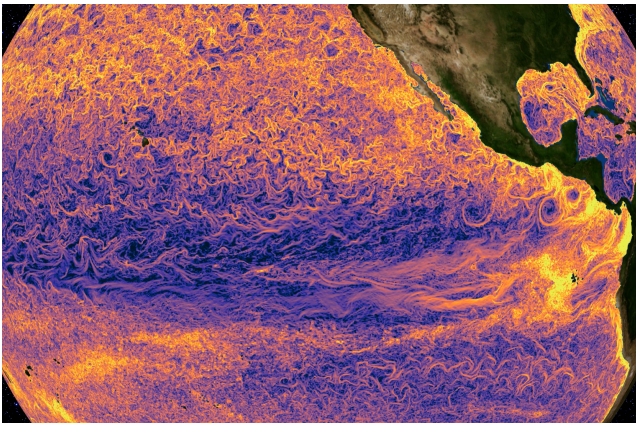


Fig. 8: Temperature fronts in a global km-scale MITgcm simulation. Plotted is the logarithm of the spatial gradient of a temperature snapshot.

### 5. Planned Extensions

Further development of the DTOR framework is expected to proceed in several directions. First, we'd like to integrate additional types of ocean observing platforms, starting with those that do not have a data structure listed yet in Tab. 1. Observational networks that focus on either the coastal ocean, a specific region, or local field experiments could also be supported in the future. In addition,

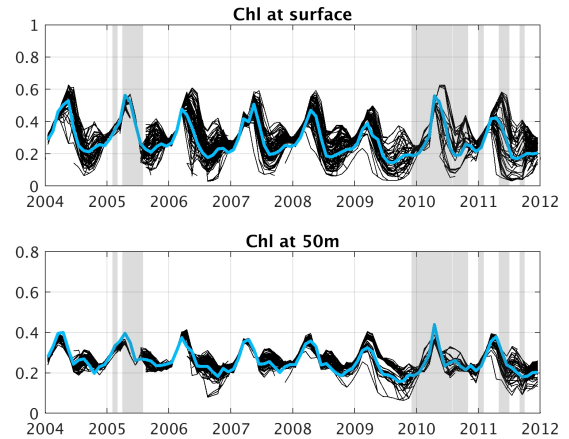


Fig. 9: Ensemble of model predictions (black curves) by a simple multi-layer perceptron (from Flux.jl) trained to predict Chlorophyll concentration (present in green algae and marine microbes) from environmental variables (T, S, but also oxygen, optical backscatter, and solar radiation). The blue line is the model *ground truth* that we seek to estimate, and was obtained through proper spatial averaging of the gridded data set from which the training data itself was generated.

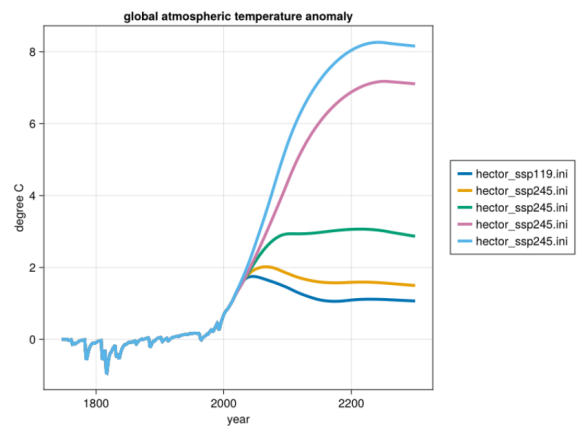


Fig. 10: Prediction of global warming based on different scenarios, called Shared Socioeconomic Pathways (SSPs), as defined by the Intergovernmental Panel on Climate Change (IPCC). These predictions were generated using the Hector model [13] via CLIMATEMODELS.JL.

a top level API to interact with DTOR as a whole through cloud services is envisioned and being developed.

A second direction to pursue is that of further integration with the Julia software stack. Ocean modeling capabilities such as OCEANANIGANS.JL [18], PLANKTONINDIVIDUALS.JL [23], AIBECS.JL [17], OCEANCOLORDATA.JL, and WORLDOCEANATLASTOOLS.JL are of immediate relevance. More broadly, there is a lot of very useful work being done across of number of github organizations that DTOR could further leverage and integrate with. To list a few : JULIACLIMATE, JULIAOCEAN, CLIMA, JULIAGEO, JULIAEARTH, JULIASPACE, JULIAROBOTICS, JULIADYNAMICS, MAKIEORG, PLUTOORG, JULIAHUB, GENIEFRAMEWORK, JULIASTATS, SCIIML, FLUXML, TURING, and JULIAAI.

## 6. Science Applications

Our current focus is on geospatial analyses that track global warming and marine heatwaves using Argo data [10]. Another example from our research is with the C-Streams observational program (UK-US) which is releasing drifters near Florida Strait to track the nutrient stream that is associated with the Gulf Stream. A simulation of these pathways is shown in Fig. 11, which is based on the monthly mean ECCO4 climatology for ocean transports [5, 12, 20], and uses CLIMATOLOGY.JL, MESHARRAYS.JL, and DRIFTERS.JL to calculate pathways. Other applications at the scale of oceanic basins include tracking ocean plastic pollution, monitoring biological impacts of extreme events and global warming, or the optimization of the global climate monitoring fleet of ocean robots. DTOR could be used to pilot field experiments (e.g., S-MODE, to choose an example from the recent past). Related projects on digital twins, not focused on exploiting Julia but interested in ocean observations, include the EU funded *Destination Earth* program, Mercator Ocean International’s *Digital Twin Ocean*, and the UN Decade Action’s DITTO initiative (*Digital Twins of The Ocean*).

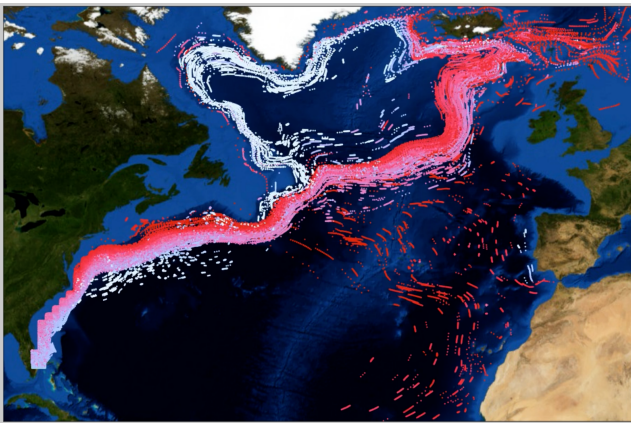


Fig. 11: Tracking Gulf Stream waters from Florida Strait to the subpolar gyre using virtual drifters that follow the three dimensional ocean circulation. Each dot color indicates the virtual drifter depth – red dots near the sea surface, while white dots are below 1500m depth.

## 7. Acknowledgements

Support from NASA awards 80NSSC20K0796, 80NSSC23K0355, 80NSSC22K1697, 1676067, and 1686358, as well as from the Simons Foundation via the CBIOMES and SCOPE-GRADIENTS program is acknowledged for this work.

## 8. References

- [1] Henry C. Bittig, Tobias Steinhoff, Hervé Claustre, Björn Fiedler, Nancy L. Williams, Raphaëlle Sauzède, Arne Körtzinger, and Jean-Pierre Gattuso. An alternative to static climatologies: Robust estimation of open ocean CO<sub>2</sub> variables and nutrient concentrations from T, S, and O<sub>2</sub> data using bayesian neural networks. *Frontiers in Marine Science*, 5, 2018. doi:10.3389/fmars.2018.00328.
- [2] Simon Danisch and Julius Krumbiegel. Makie.jl: Flexible high-performance data visualization for Julia. *Journal of Open Source Software*, 6(65):3349, 2021. doi:10.21105/joss.03349.
- [3] V. Eyring, S. Bony, G. A. Meehl, C. A. Senior, B. Stevens, R. J. Stouffer, and K. E. Taylor. Overview of the coupled model intercomparison project phase 6 (CMIP6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958, 2016. doi:10.5194/gmd-9-1937-2016.
- [4] G. Forget. Mapping ocean observations in a dynamical framework: a 2004-06 ocean atlas. *Journal of Physical Oceanography*, 40(6), 2010. doi:10.1175/2009JPO4043.1.
- [5] G. Forget, J.-M. Campin, P. Heimbach, C. N. Hill, R. M. Ponte, and C. Wunsch. ECCO version 4: an integrated framework for non-linear inverse modeling and global ocean state estimation. *Geoscientific Model Development*, 8(10):3071–3104, oct 2015. doi:10.5194/gmd-8-3071-2015.
- [6] G. Forget, D. Ferreira, and X. Liang. On the observability of turbulent transport rates by Argo: supporting evidence from an inversion experiment. *Ocean Science*, 11(5):839–853, 2015. doi:10.5194/os-11-839-2015.
- [7] G. Forget and R.M. Ponte. The partition of regional sea level variability. *Progress in Oceanography*, 137, 2015. doi:10.1016/j.pocean.2015.06.002.
- [8] G. Forget and C. Wunsch. Estimated global hydrographic variability. *Journal of Physical Oceanography*, 37(8), 2007. doi:10.1175/JPO3072.1.
- [9] Gaël Forget. IndividualDisplacements.jl: a Julia package to simulate and study particle displacements within the climate system. *Journal of Open Source Software*, 6(60):2813, 2021. doi:10.21105/joss.02813.
- [10] Gaël Forget. Energy imbalance in the sunlit ocean layer. (submitted), 2024. doi:10.21203/rs.3.rs-3979671/v1.
- [11] Gaël Forget. MITgcm.jl: a Julia interface to the MITgcm. *Journal of Open Source Software*, submitted, 2024. doi:10.21105/joss.06710.
- [12] Gaël Forget and David Ferreira. Global ocean heat transport dominated by heat export from the tropical Pacific. *Nature Geoscience*, 2019. doi:10.1038/s41561-019-0333-7.
- [13] C. A. Hartin, P. Patel, A. Schwarber, R. P. Link, and B. P. Bond-Lamberty. A simple object-oriented and open-source model for scientific and policy analyses of the global climate system – Hector v1.0. *Geoscientific Model Development*, 8(4):939–955, 2015. doi:10.5194/gmd-8-939-2015.
- [14] Mike Innes. Flux: Elegant machine learning with Julia. *Journal of Open Source Software*, 3(25):602, 2018. doi:10.21105/joss.00602.
- [15] K McCaffrey, B Fox-Kemper, and G Forget. Estimates of ocean macro-turbulence: structure function and spectral slope from Argo profiling floats. *JPO*, 45:1773–1793, 2015. doi:10.1175/JPO-D-14-0023.1.
- [16] National Academies of Sciences Engineering and Medicine. Foundational research gaps and future directions for digital twins. *The National Academies Press*, 2024. doi:10.17226/26894.
- [17] Benoît Pasquier, François W. Primeau, and Seth G. John. AIBECs.jl: a tool for exploring global marine biogeochemical cycles. *Journal of Open Source Software*, 7(69):3814, 2022. doi:10.21105/joss.03814.
- [18] Ali Ramadhan, Gregory LeClaire Wagner, Chris Hill, Jean-Michel Campin, Valentin Churavy, Tim Besard, Andre Souza, Alan Edelman, Raffaele Ferrari, and John Marshall.

Oceananigans.jl: fast and friendly geophysical fluid dynamics on GPUs. *Journal of Open Source Software*, 5(53):2018, 2020. doi:10.21105/joss.02018.

- [19] Dean Roemmich, Matthew H. Alford, Hervé Claustre, Kenneth Johnson, Brian King, James Moum, Peter Oke, W. Brechner Owens, Sylvie Pouliquen, Sarah Purkey, Megan Scanderbeg, Toshio Suga, Susan Wijffels, Nathalie Zilberman, Dorothee Bakker, Molly Baringer, Mathieu Belbeoch, Henry C. Bittig, Emmanuel Boss, Paulo Calil, Fiona Carse, Thierry Carval, Fei Chai, Diarmuid Conchubhair, Fabrizio d’Ortenzio, Giorgio Dall’Olmo, Damien Desbruyeres, Katja Fennel, Ilker Fer, Raffaele Ferrari, Gael Forget, Howard Freeland, Tetsuichi Fujiki, Marion Gehlen, Blair Greenan, Robert Hallberg, Toshiyuki Hibiya, Shigeki Hosoda, Steven Jayne, Markus Jochum, Gregory C. Johnson, KiRyong Kang, Nicolas Kolodziejczyk, Arne Körtzinger, Pierre-Yves Le Traon, Yueng-Djern Lenn, Guillaume Maze, Kjell Arne Mork, Tamaryn Morris, Takeyoshi Nagai, Jonathan Nash, Alberto Naveira Garabato, Are Olsen, Rama Rao Pattabhi, Satya Prakash, Stephen Riser, Catherine Schmechtig, Claudia Schmid, Emily Shroyer, Andreas Sterl, Philip Sutton, Lynne Talley, Toste Tanhua, Virginie Thierry, Sandy Thomalla, John Toole, Ariel Troisi, Thomas W. Trull, Jon Turton, Pedro Joaquin Velez-Belchi, Waldemar Walczowski, Haili Wang, Rik Wanninkhof, Amy F. Waterhouse, Stephanie Waterman, Andrew Watson, Cara Wilson, Annie P. S. Wong, Jianping Xu, and Ichiro Yasuda. On the future of Argo: A global, full-depth, multi-disciplinary array. *Frontiers in Marine Science*, 6, 2019. doi:10.3389/fmars.2019.00439.
- [20] Louise Rousselet, Paola Cessi, and Gael Forget. Coupling of the mid-depth and abyssal components of the global overturning circulation according to a state estimate. *Science Advances*, 7(21):eabf5478, 2021. doi:10.1126/sciadv.abf5478.
- [21] Raphaëlle Sauzède, Henry C. Bittig, Hervé Claustre, Orens Pasqueron de Fommervault, Jean-Pierre Gattuso, Louis Legendre, and Kenneth S. Johnson. Estimates of water-column nutrient concentrations and carbonate system parameters in the global ocean: A novel approach based on neural networks. *Frontiers in Marine Science*, 4, 2017. doi:10.3389/fmars.2017.00128.
- [22] Fons van der Plas and collaborators. fonsp/Pluto.jl: v0.19.42, May 2024. doi:10.5281/zenodo.11144554.
- [23] Zhen Wu and Gaël Forget. Planktonindividuals.jl: a GPU supported individual-based phytoplankton life cycle model. *Journal of Open Source Software*, 7(73):4207, 2022. doi:10.21105/joss.04207.