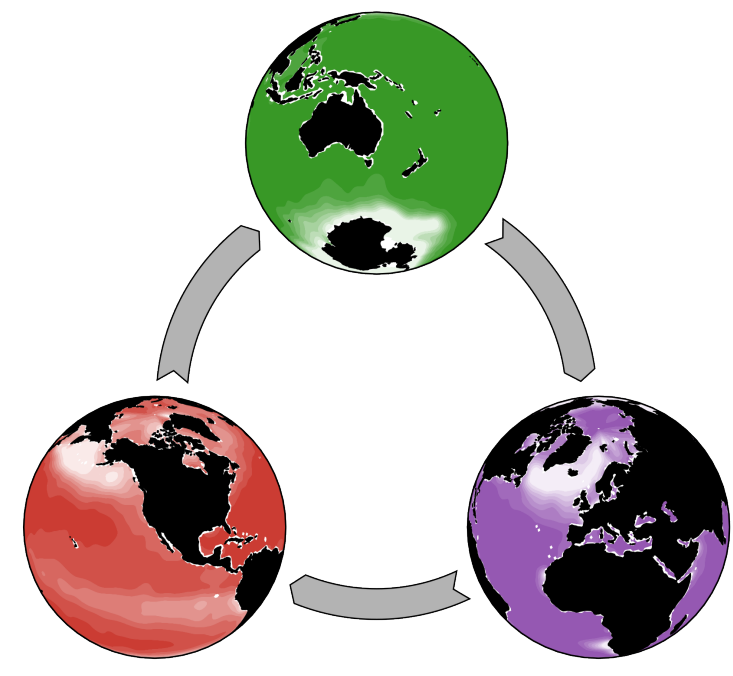


# AIBECS.jl



## The ideal tool for marine biogeochemistry modelling

Benoit Pasquier  
bpasquier.com / bp\_358@usc.edu

University of Southern California

### Motivation

Ideally, a global marine biogeochemistry (BGC) model should only require you to define a few processes and an ocean circulation model. AIBECS [8] aims to deliver that simplicity, following in the footsteps of the AWESOME OCIM [6], but with some additional features.

### How it works

Any discrete steady-state BGC model is entirely described by a single equation of the type

$$(\partial_t + \mathbf{T}(\mathbf{p}))\mathbf{x} = \mathbf{G}(\mathbf{x}, \mathbf{p})$$

where  $\mathbf{x}$  is a vector of all the state variables (the tracer concentrations). The flux divergence of tracers resulting from the transport by the ocean circulation and other processes (such as the biological pump) are given by  $\mathbf{T}(\mathbf{p})\mathbf{x}$  and the local sources and sinks by  $\mathbf{G}(\mathbf{x}, \mathbf{p})$ .

AIBECS requires you to chose/define a transport and the local sources and sinks of each tracer. Under the hood, AIBECS then generates  $\mathbf{T}$ ,  $\mathbf{G}$ , and the state function,  $\mathbf{F}(\mathbf{x}, \mathbf{p}) = \partial_t \mathbf{x}$ . You can then directly solve for the steady-state solution,  $\mathbf{x} = \mathbf{s}(\mathbf{p})$ , which satisfies

$$\mathbf{F}(\mathbf{x}, \mathbf{p}) = 0$$

with an efficient quasi-Newton algorithm provided by AIBECS. This requires the Jacobian of  $\mathbf{F}$ , which is auto-differentiated using [ForwardDiff.jl](#) [12] by AIBECS at virtually zero computational cost.

### Ocean circulations

AIBECS currently comes with a few ocean circulations that you can swap with a single line of code:

- OCIM0.1 [11]
- OCIM1 [3, 5]
- OCIM2 [4]
- simplest 2-box model [13]
- 3-box archer et al. model [1]
- 5-box F. Primeau model

### Using observational data

[WorldOceanAtlasTools.jl](#) allows you to programmatically download World Ocean Atlas (WOA) data to use directly with your model simulations [10].

In a similar vein, [GEOTRACES.jl](#) can read GEOTRACES IDP 17 data as long as you manually download the data beforehand. (See right panel for an example using Cadmium data.)

### Inverse modelling

Manual calibrations of global BGC models are increasingly being replaced by more automated approaches such as inverse modelling. Let  $f(\mathbf{x}, \mathbf{p})$  be a given objective function quantifying the model/observations mismatch to be minimized along the manifold of steady-state solutions, i.e., where  $\mathbf{x} = \mathbf{s}(\mathbf{p})$ .

The [F1Method.jl](#) package allows for evaluation of the corresponding gradient and Hessian at virtually no computational cost, using a combination of adjoint methods, hyper-dual numbers, and analytical shortcuts [9]. This in turn allows the use of the gold standard of optimization routines, which leverage Hessian information using, e.g., Julia's [Optim.jl](#) [7].

### Getting started is easy

1. Install Julia [2] from [julia.org](#). (Julia is a high-performance, open-source, and free language that is ideal for scientific computing.)
2. Install AIBECS. After starting Julia, type ] to enter Julia's package manager and then just type:

```
pkg> add AIBECS
```

3. Try the tutorials. Follow the QR code in the central panel and head over to the documentation to find tutorials and how-to guides available as Jupyter notebooks. (The ideal-mean age example on the right panel was taken from the first tutorial.)

### Future work

AIBECS works but it is still in development. In future versions AIBECS will include new circulations, new matrix-free solvers for larger systems and/or seasonal ocean circulations, new and fancier plotting recipes for flawless publication-quality figures, new diagnostics, and more. AIBECS will become part of the JuliaOcean GitHub organization to share interface and data with other ocean-related packages being developed as Julia grows.

# AIBECS

is a new Julia package for simple and efficient modelling of global biogeochemical tracers.

It is free and open-source, thoroughly tested, and regularly updated.

Use it for teaching, for exploring ideas, or for cutting-edge research!

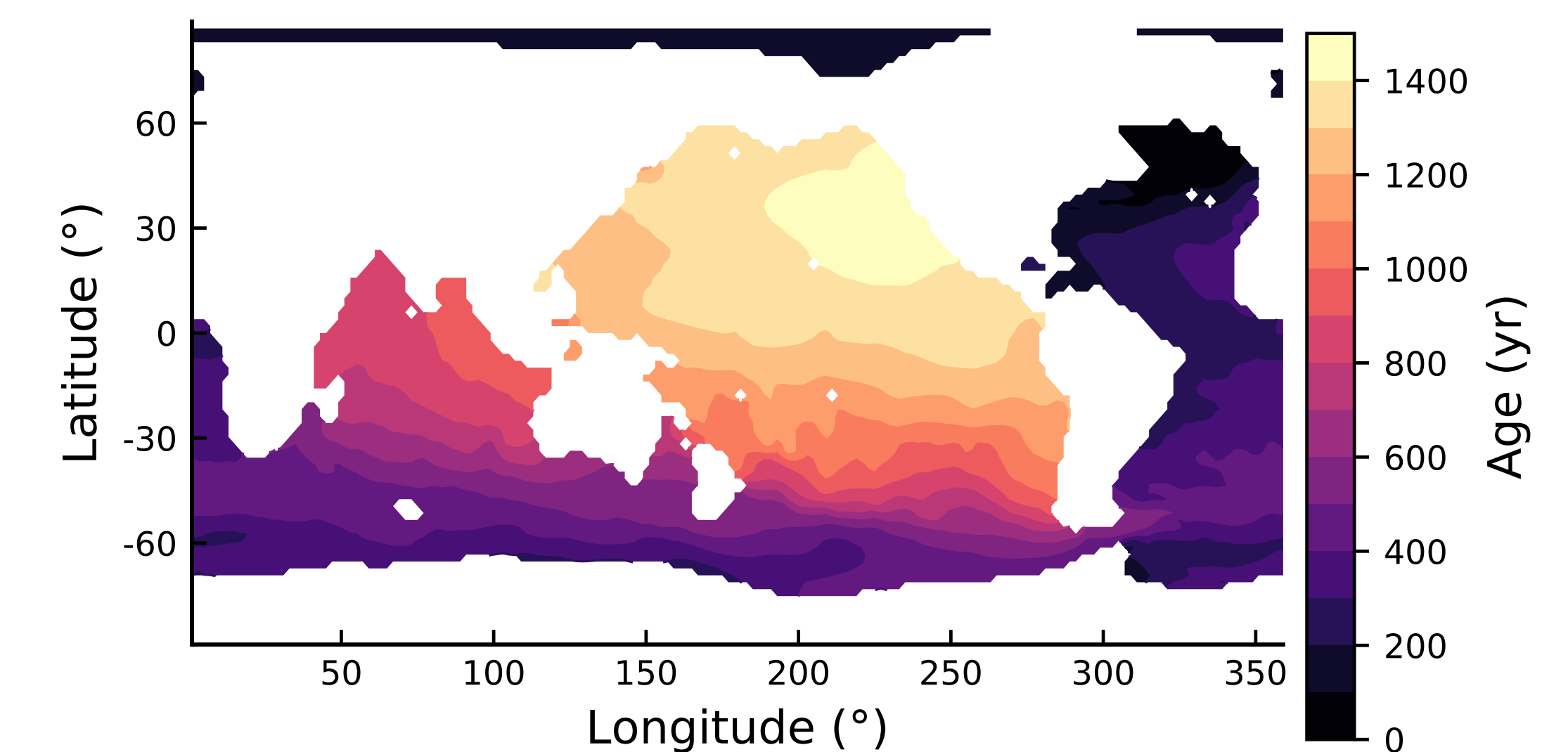


Access the code, documentation, and notebooks at [github.com/briochemc/AIBECS.jl](https://github.com/briochemc/AIBECS.jl) or simply aim your camera at the QR code.

### Example 1: Water age

Solving and plotting the ideal mean age takes less than 20 lines of code:

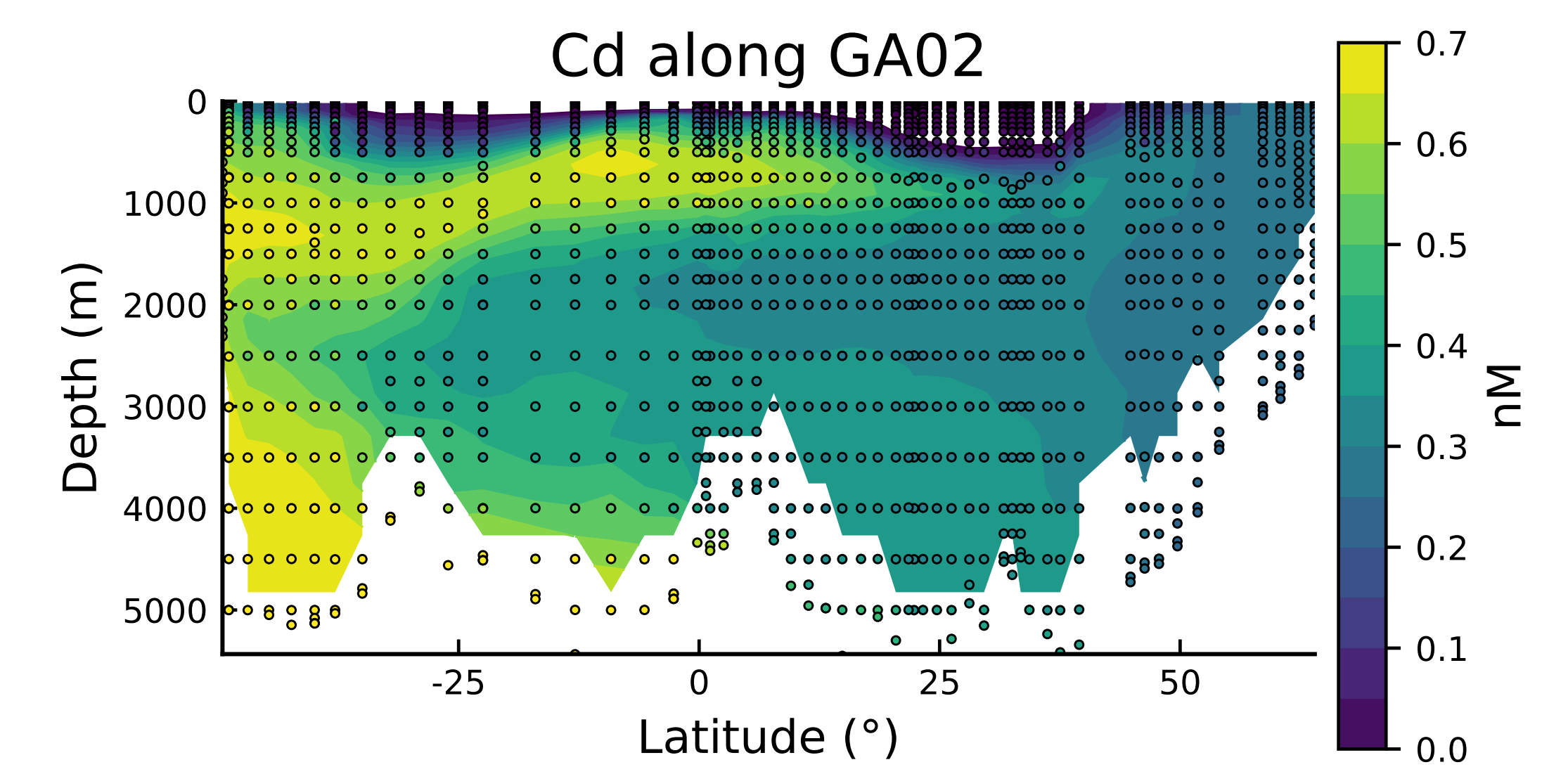
```
using AIBECS, Plots
grd, TOCIM2 = OCIM2.load()
T(p) = TOCIM2
function G(x,p)
    @unpack τ, Z₀ = p
    return @. 1 - x / τ * (Z < Z₀)
end
z = depthvec(grd)
F, ∇.F = state_function_and_Jacobian(T, G)
struct IdealAgeParameters{U} <: AbstractParameters{U}
    τ::U
    Z₀::U
end
p = IdealAgeParameters(1.0, 20.0)
x_init = zeros(count(iswet(grd)))
prob = SteadyStateProblem(F, ∇.F, x_init, p)
age = solve(prob, CTAlg()) * u"s" .|> u"yr"
horizontalslice(age, grd, depth=2"km", colorbar_title="Age", color=:magma)
```



### Example 2: Cadmium

Below is a simple Cadmium model (using a prescribed P uptake) run and plotted along the GA02 transect with GEOTRACES data overlaid.

```
using AIBECS, BSON, OceanographyCruises, GEOTRACES, Plots
grd, T = OCIM1.load()
T_DCD(p) = T
const z = depthvec(grd)
T_PCD(p) = transportoperator(grd, w=w(p))
function w(p)
    @unpack wₑ, w' = p
    return @. wₑ + w' * z
end
BSON.@load "uptake_P.bson" uptake_P
function G_DCD(DCd, PCd, p)
    @unpack Cd/P, τPCd, xgeo, tgeo = p
    return @. -Cd/P * uptake_P + PCd / τPCd + (xgeo - DCd) / tgeo
end
function G_PCD(DCd, PCd, p)
    @unpack Cd/P, τPCd = p
    return @. Cd/P * uptake_P - PCd / τPCd
end
import AIBECS: @units, units, @initial_value, initial_value
@units @initial_value struct Params{Tp} <: AbstractParameters{Tp}
    wₑ::Tp | 0.64 | u"m/d"
    w'::Tp | 0.13 | u"m/d/m"
    Cd/P::Tp | 0.49 | u"nmol/μmol"
    τPCd::Tp | 5.0 | u"d"
    tgeo::Tp | 1.0 | u"Myr"
    xgeo::Tp | 0.63 | u"nM"
end
nb = count(iswet(grd))
x = zeros(2nb)
F, ∇.F = state_function_and_Jacobian((T_DCD, T_PCD), (G_DCD, G_PCD), nb)
s = solve(SteadyStateProblem(F, ∇.F, x, Params{Tp}(), CTAlg()))
t = uconvert(u"nM", GEOTRACES.transects("Cd") * 1.035u"kg/L").transects[1]
DCd = uconvert(u"nM", unpack_tracers(s, grd)[1] * u"mol/m³")
ct = OceanographyCruises.CruiseTrack(t)
meridionaltransect(DCd, grd, ct=ct, clim=(0,0.7), levs=0:0.05:0.7, c=:viridis)
meridionalscattertransect!(t, clim=(0,0.7), c=:viridis, ms=2, msw=0.5)
```



- [1] David E. Archer et al. Atmospheric pCO<sub>2</sub> sensitivity to the biological pump in the ocean. *Global Biogeochemical Cycles*, 14(4):1219–1230, 2000. doi:10.1029/1999GB001216.
- [2] J. Bezanon, A. Edelman, S. Karpinski, and V. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017. doi:10.1137/14100671.
- [3] Tim DeVries. The oceanic anthropogenic CO<sub>2</sub> sink: Storage, air-sea fluxes, and transports over the industrial era. *Global Biogeochemical Cycles*, 28(7):631–647, 2014. doi:10.1002/2013GB004739.
- [4] Tim DeVries and Mark Holzer. Radiocarbon and helium isotope constraints on deep ocean ventilation and mantle-<sup>3</sup>He sources. *Journal of Geophysical Research: Oceans*, 124(5):3036–3057, 2019. doi:10.1029/2018JC014716.
- [5] Tim DeVries and François Primeau. Dynamically and observationally constrained estimates of water-mass distributions and ages in the global ocean. *Journal of Physical Oceanography*, 41:2381–2401, 2011. doi:10.1175/JPO-D-10-05011.1.
- [6] Seth G. John, Hengdi Liang, et al. AWESOME OCIM: A simple, flexible, and powerful tool for modeling elemental cycling in the oceans. *Chemical Geology*, 533:119403, 2020. doi:10.1016/j.chemgeo.2019.119403.
- [7] Patrick Kofod Mogensen and Asbjørn Nilsen Rieðth. Optim: A mathematical optimization package for Julia. *Journal of Open Source Software*, 3(24):615, 2018. doi:10.21105/joss.00615.
- [8] Benoît Pasquier. AIBECS.jl: The ideal tool for exploring global marine biogeochemical cycles, May 2019. doi:10.5281/zenodo.2864051.
- [9] Benoît Pasquier. F1Method.jl: A Julia package for computing the gradient and Hessian of an objective function defined implicitly by the solution to a steady-state problem, May 2019. doi:10.5281/zenodo.2667835.
- [10] Benoît Pasquier. WorldOceanAtlasTools.jl: Download and process World Ocean Data in Julia, May 2019. doi:10.5281/zenodo.2677666.
- [11] F. W. Primeau, M. Holzer, and T. DeVries. Southern Ocean nutrient trapping and the efficiency of the biological pump. *Journal of Geophysical Research*, 118:2547–2564, 2013. doi:10.1002/jgrc.20181.
- [12] J. Revels, M. Lubin, and T. Papamarkou. Forward-mode automatic differentiation in Julia. *arXiv:1607.07892 [cs.MS]*, 2016. URL: <https://arxiv.org/abs/1607.07892>.
- [13] J. L. Sarmiento and N. Gruber. *Ocean Biogeochemical Dynamics*. Princeton Univ. Press, Princeton and Oxford, 2006.