THE UNIVERSITY OF MELBOURNE

21st Century Weather
ARC Centre of Excellence

Waves of the future

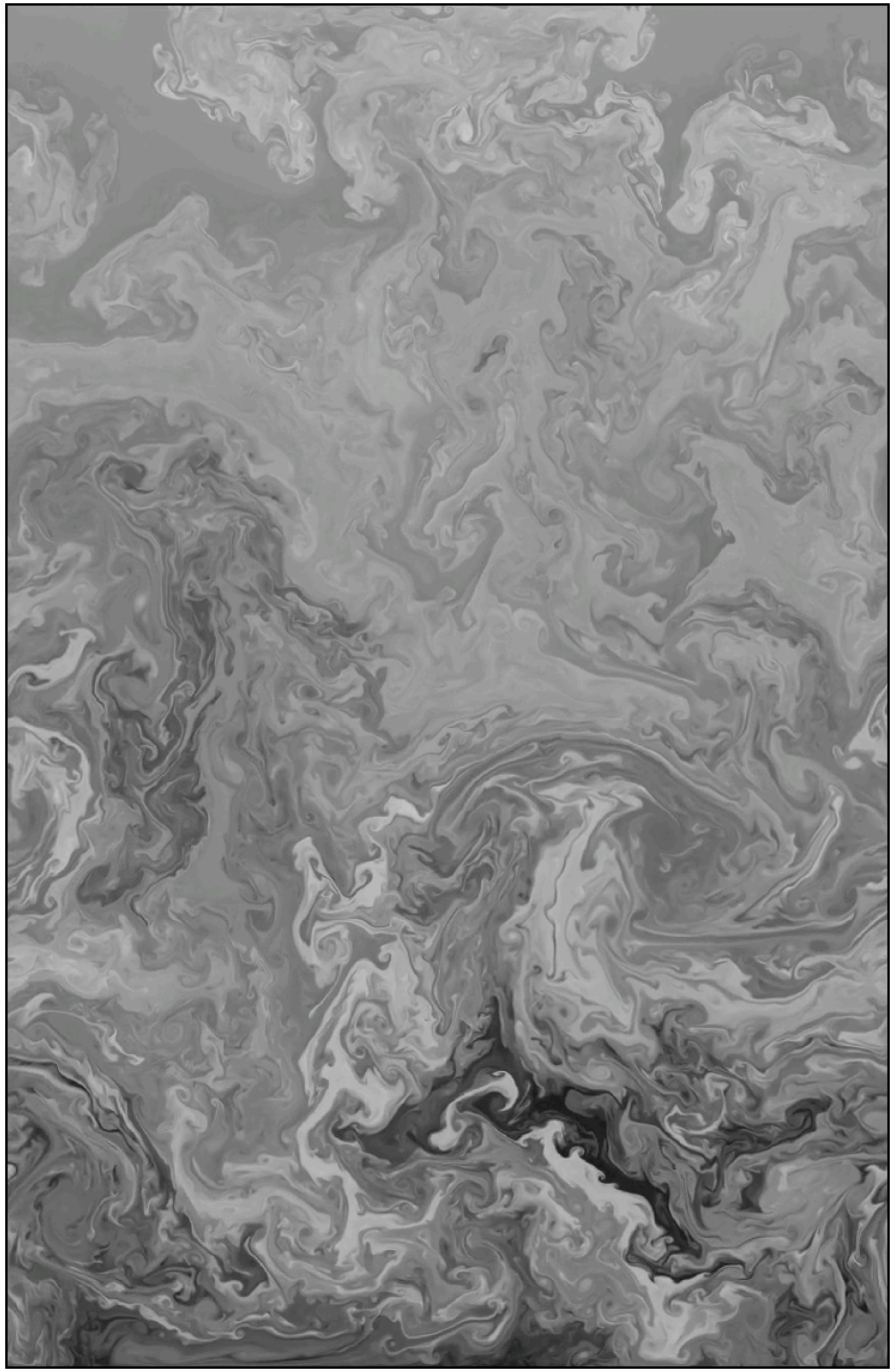
how modern software transforms ocean research & teaching

Navid Constantinou

Sean Haney 2025 Memorial Symposium

Feb 2025

SCRIPPS INSTITUTION OF OCEANOGRAPHY



stratified turbulence simulated with Oceananigans
animation made using Greg Wagner's `TurbulentImages.jl` package

why climate is tad difficult problem? 🌤️ 🌏 🌀

climate models 👩🏾‍💻 💾

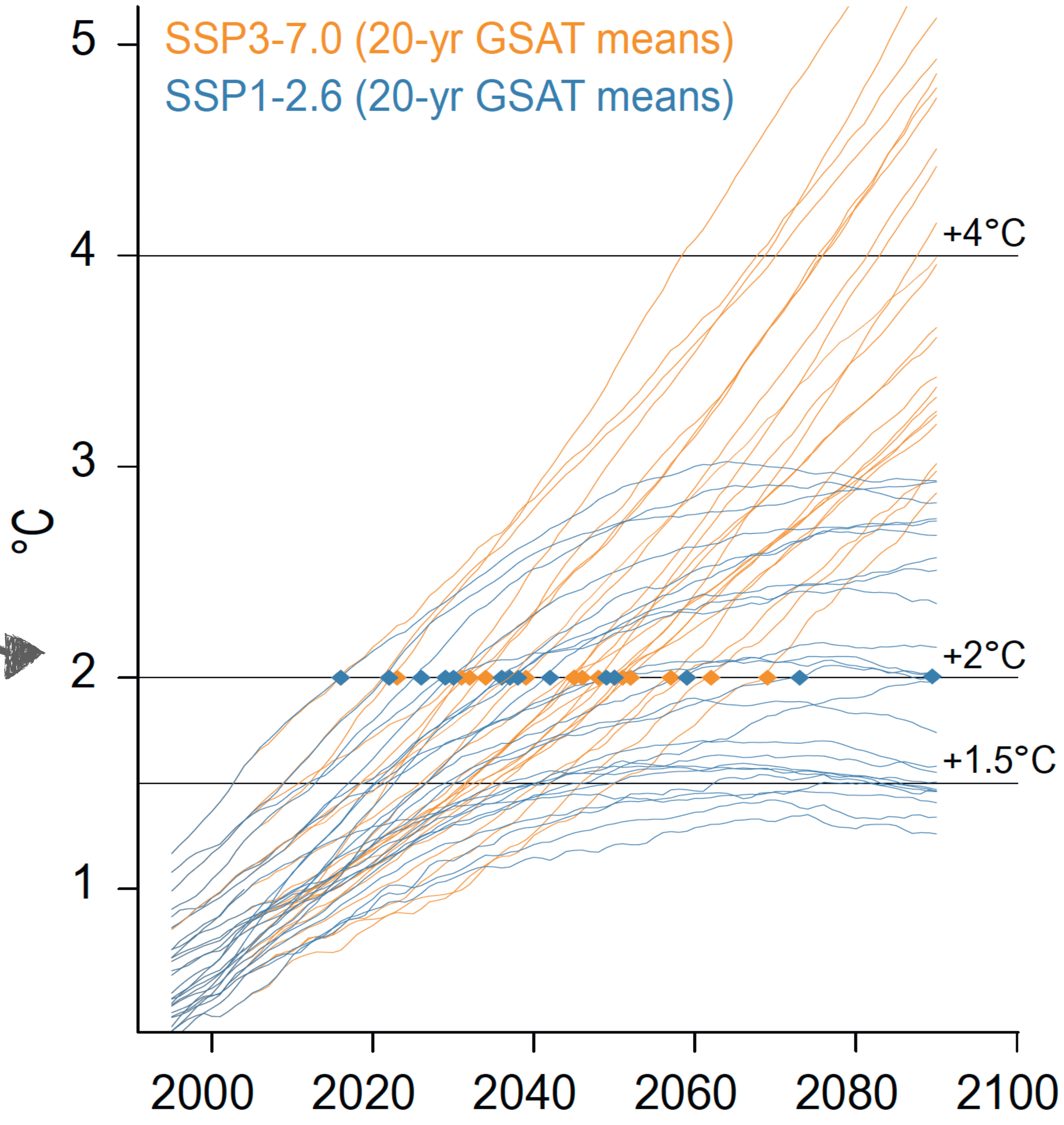
modern software 🛸
& its impact on our work flow in
research 👩🏽‍🔬 and teaching 👨🏽‍🎓

why climate is tad difficult problem? 🌤️ 🌏 🌀

climate 🌏 is what you expect,
weather is what you get 🌦️

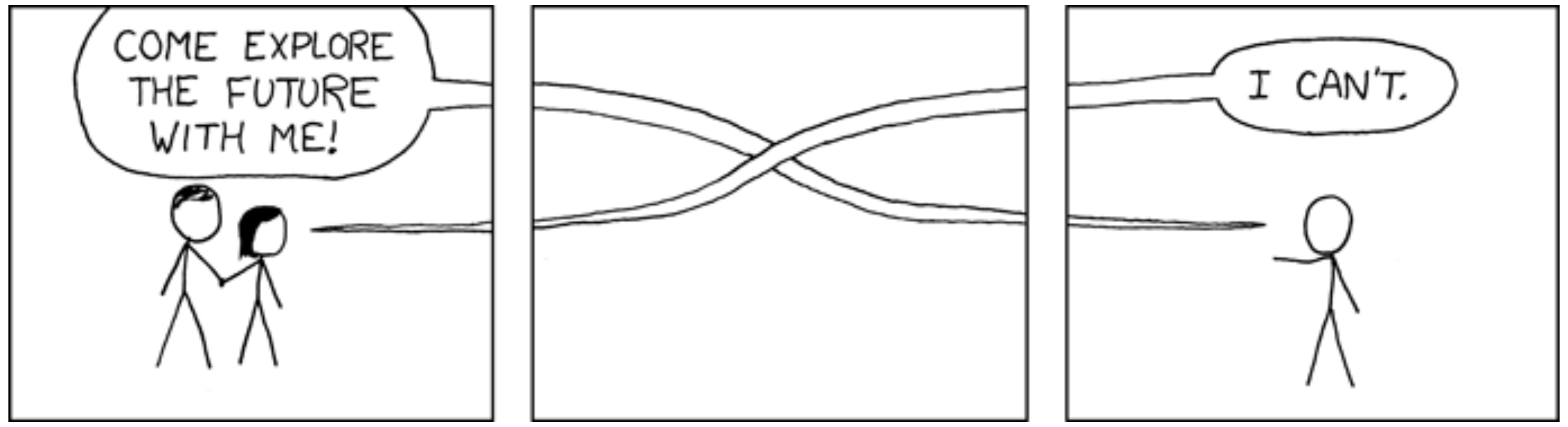
predictions of future warming are *uncertain*

2°C threshold



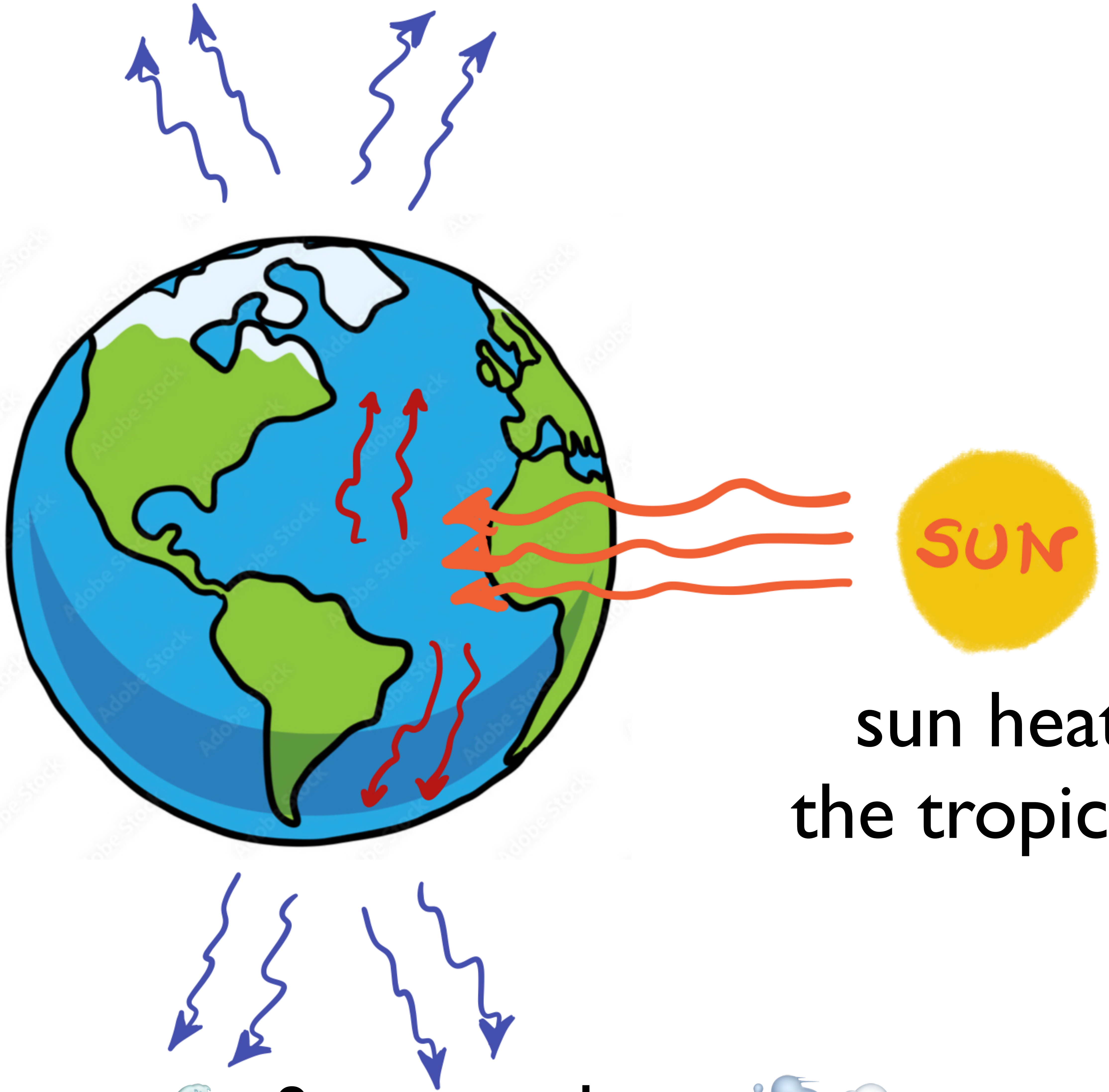
High-emissions (SSP3-7.0)
and
low-emissions (SSP1-2.6)
scenarios in CMIP6

IPCC AR 6 Working Group I



[xkcd.com/338]

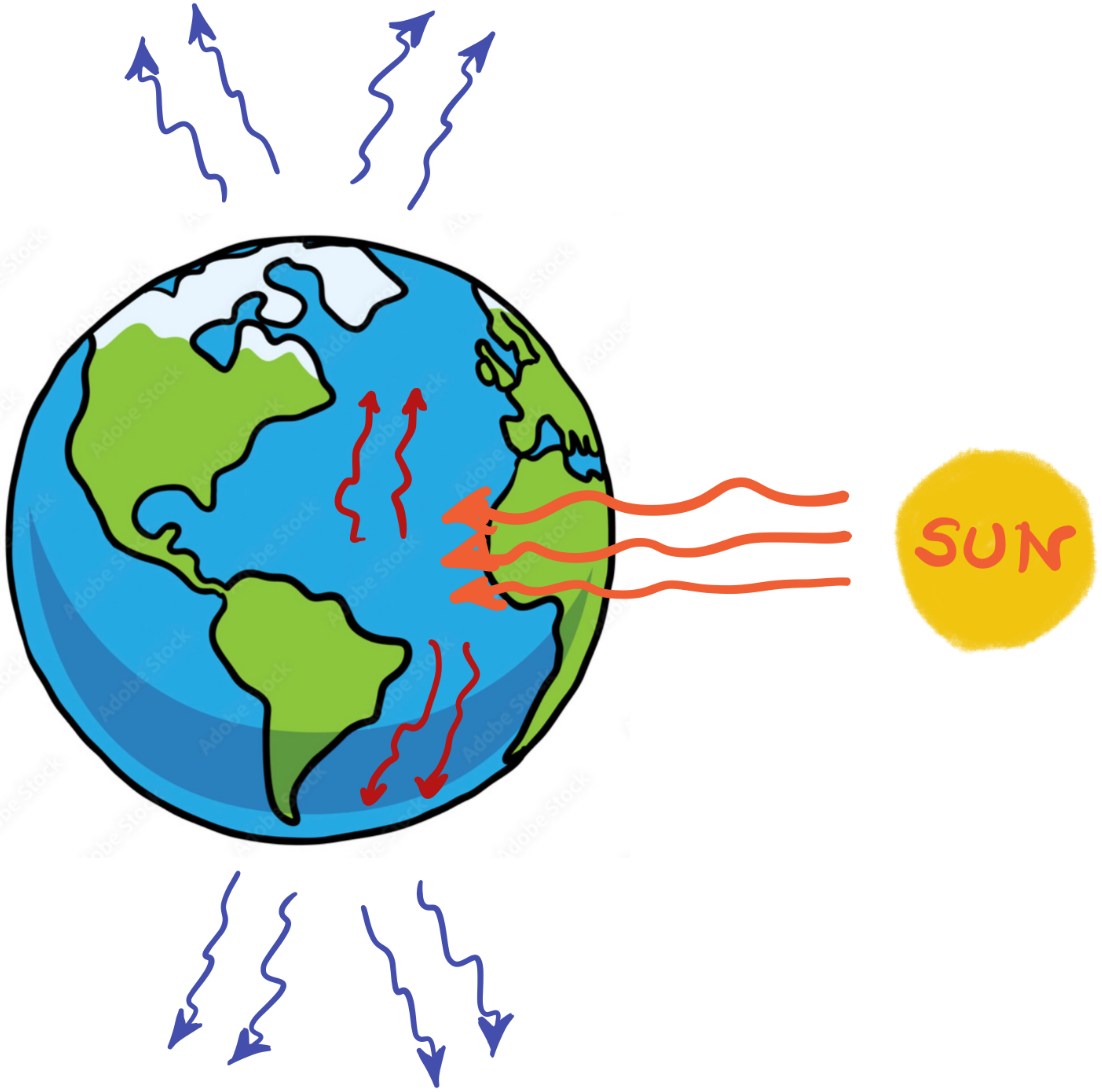
what sets the ocean (and atmosphere) in motion



sun heats
the tropics…

…ocean 🌊 & atmosphere 💨☁️
carry some of the heat to the poles

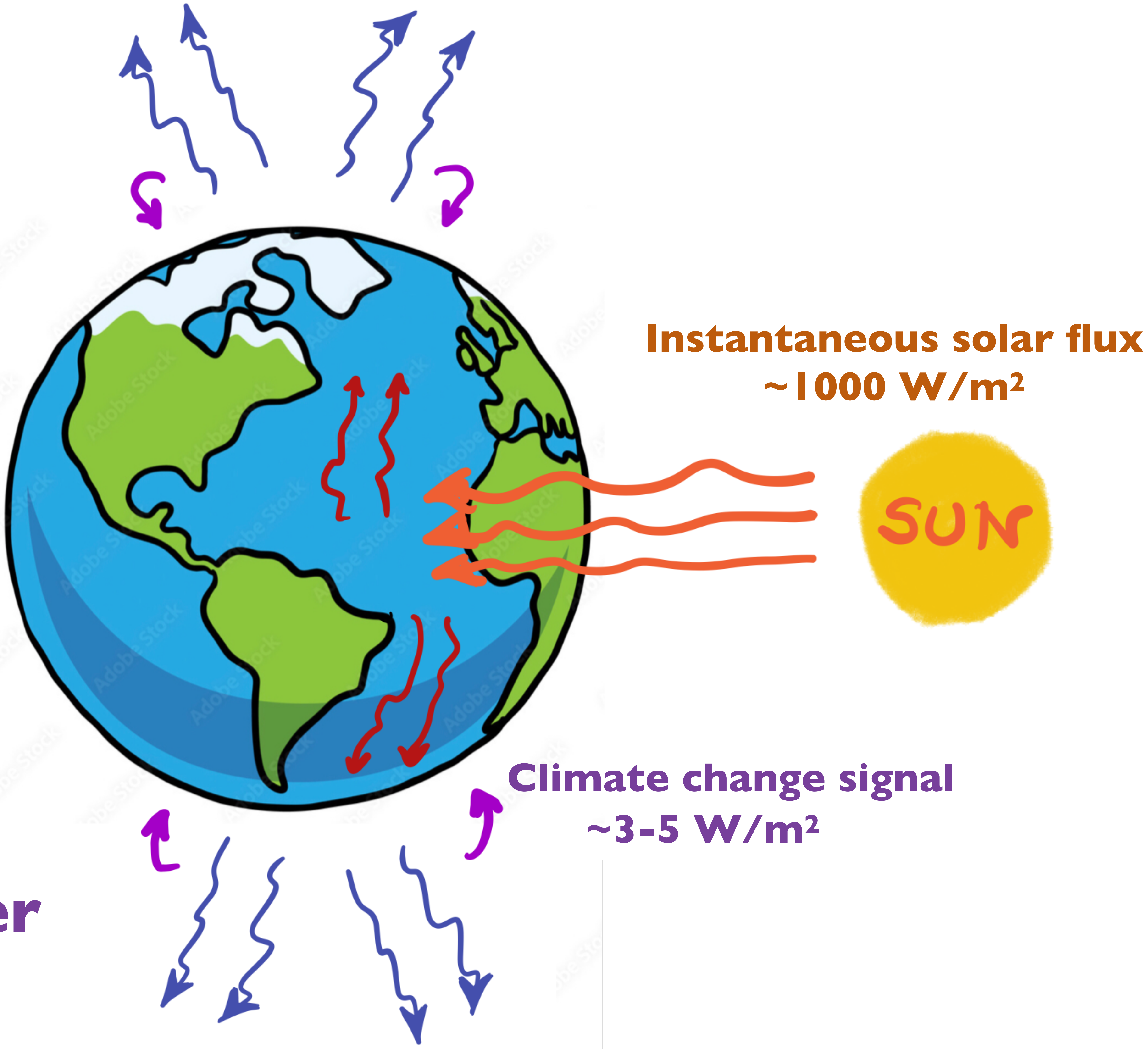
‘steady’ climate / climate in equilibrium



energy in = energy out

energy in is exactly equal to energy out

a changing climate



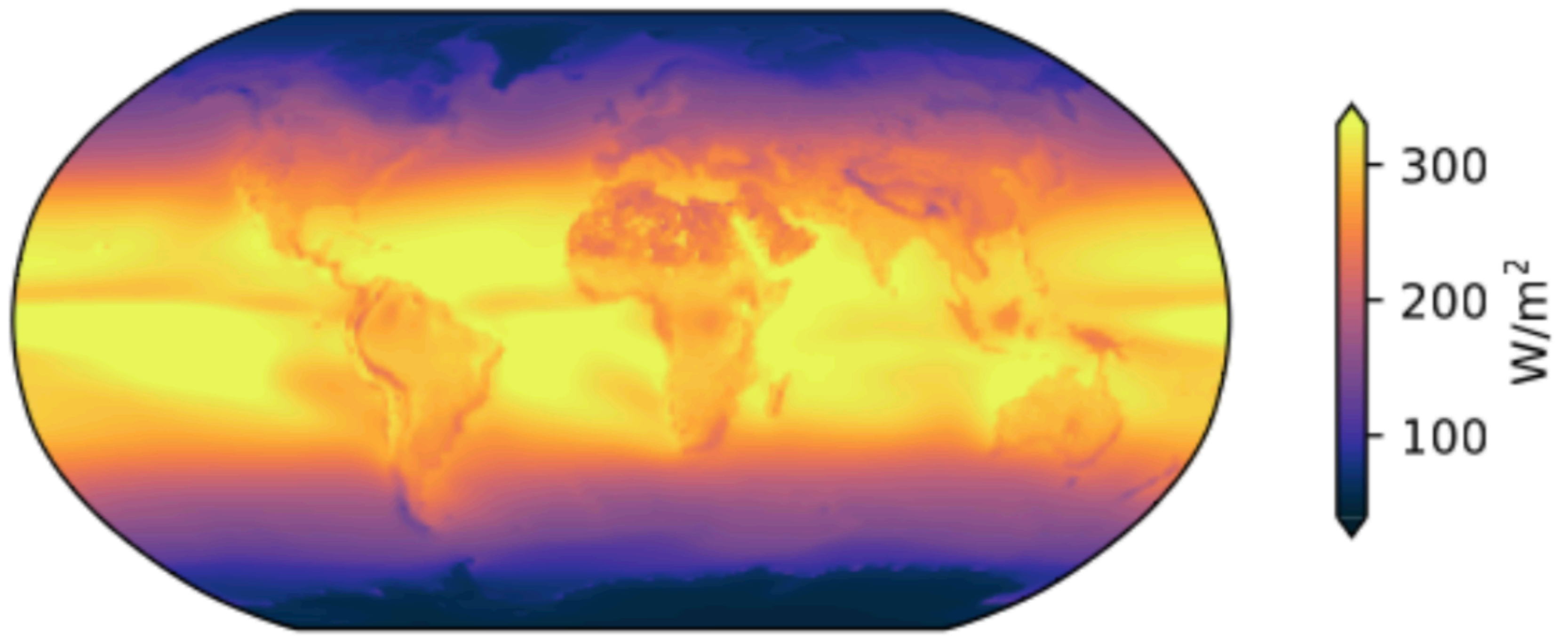
energy in = **energy out**
+ **a little**
bit leftover

energy in is **not quite** equal to **energy out**

if you like being a bit more quantitative..

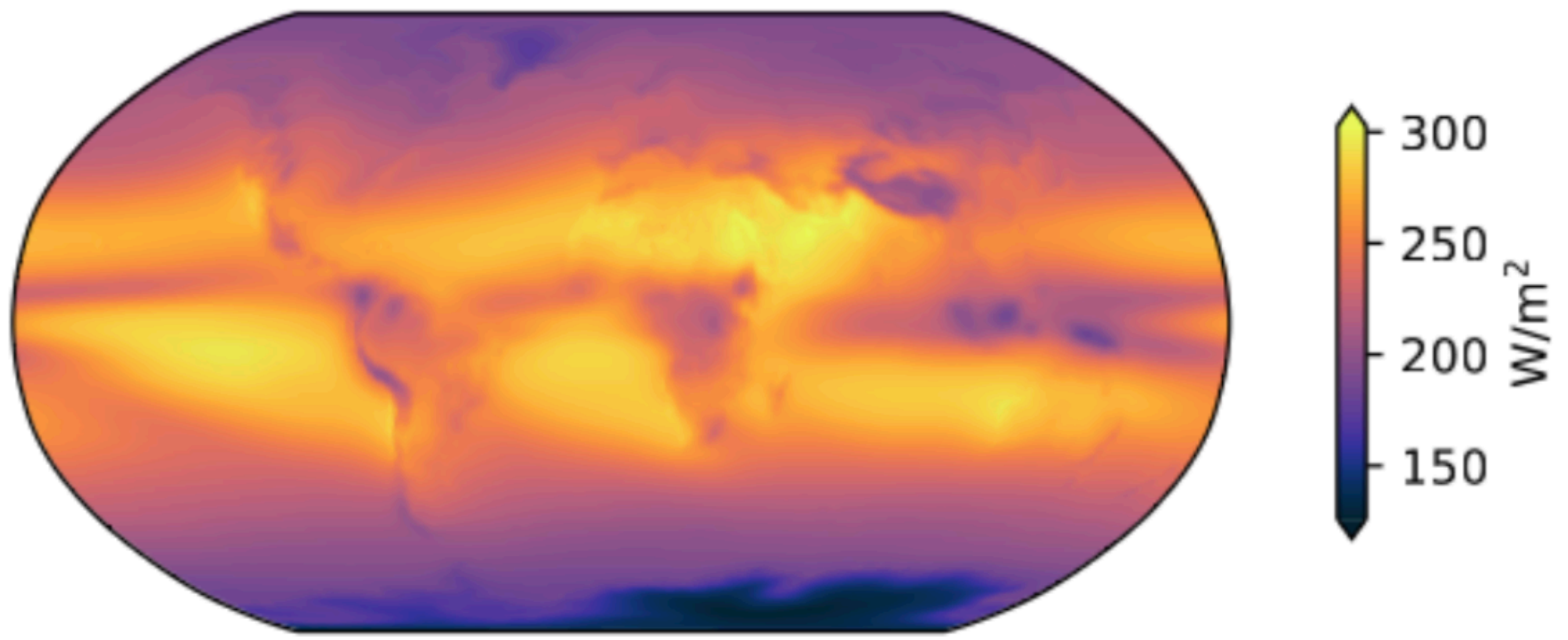
energy in

net shortwave coming in
(incoming – outgoing)



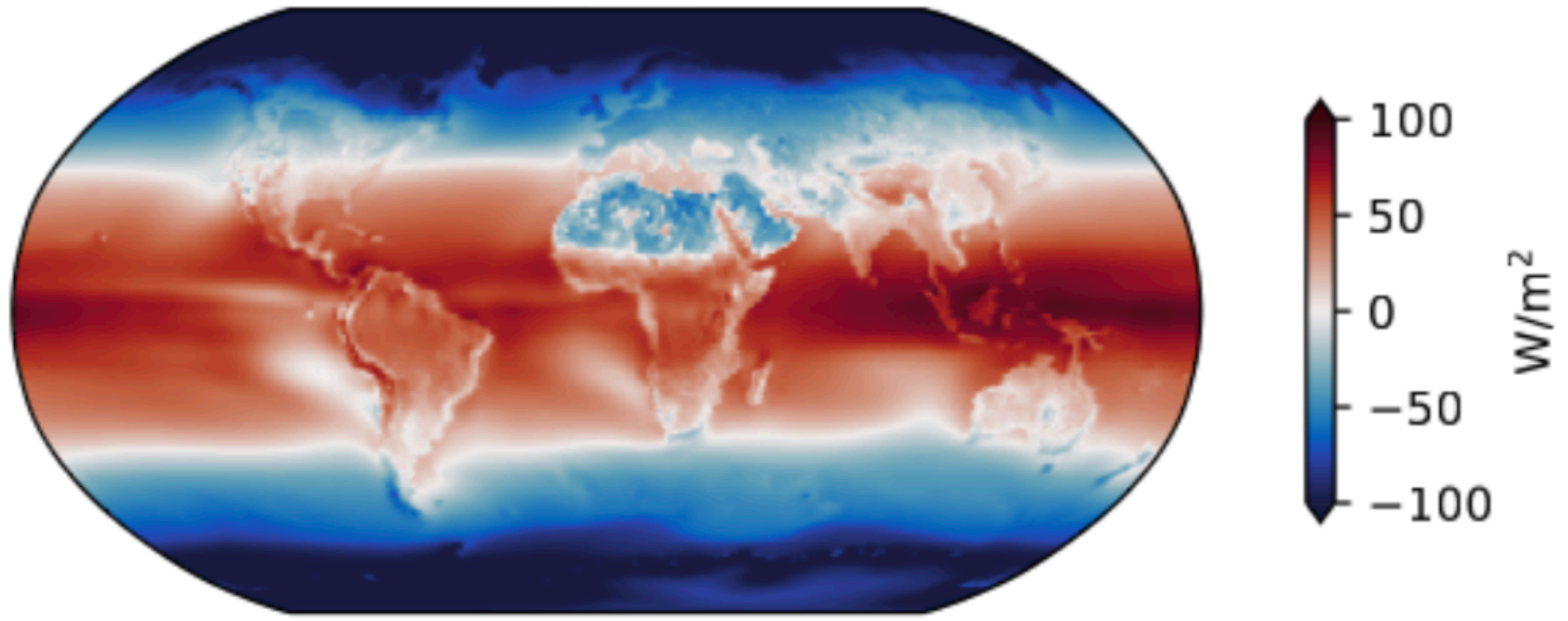
energy out

outgoing longwave



their difference

net radiation coming in



[CESM pre-industrial control climate simulation for CMIP6]

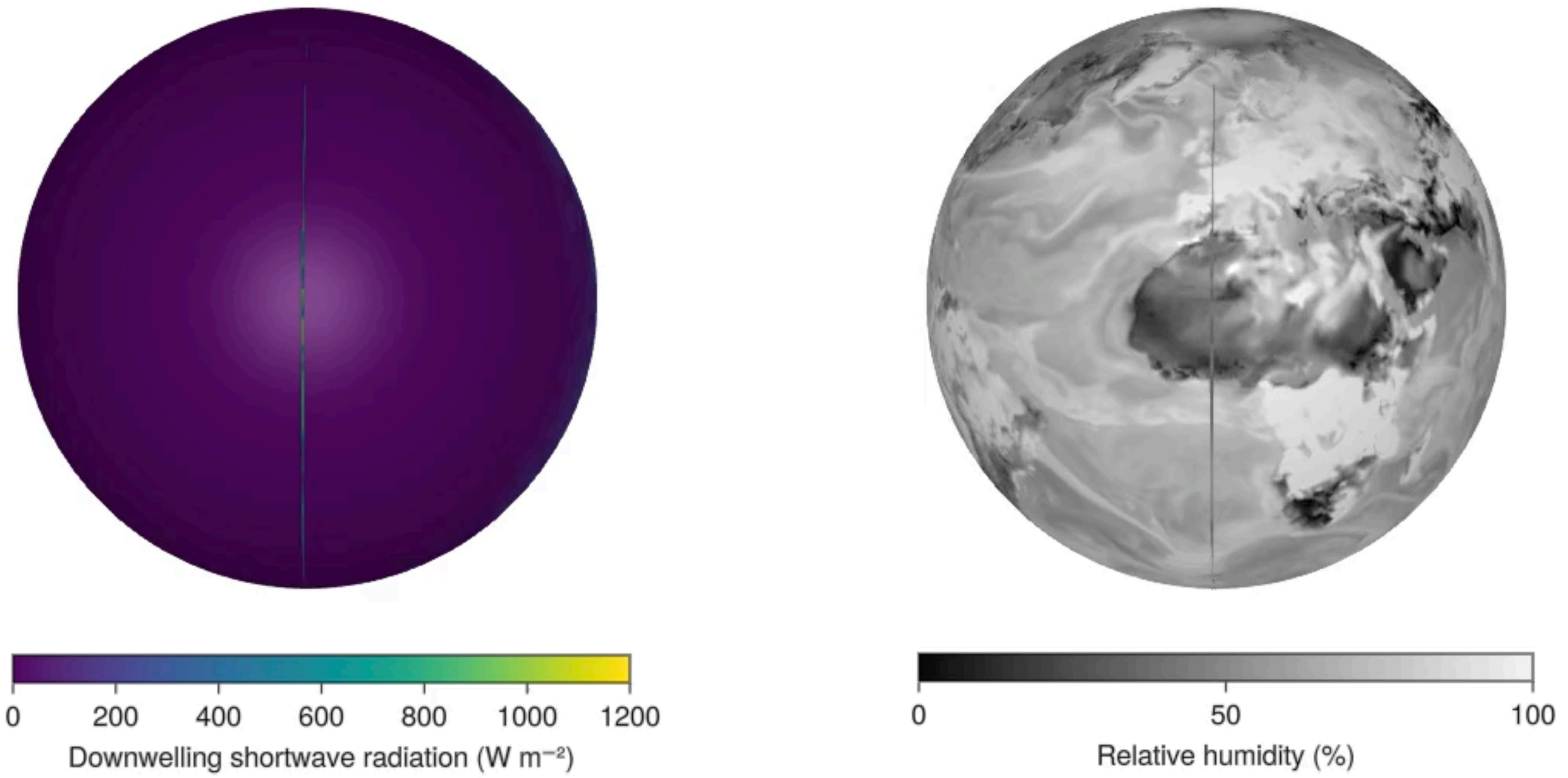
why is climate prediction so hard?

The Earth system is *strongly* forced
but very close to equilibrium

Climate change signal: ~3-5 W/m²
Instantaneous solar flux: ~1300 W/m²

JRA55 forcing on year-day 0.0

"energy in"

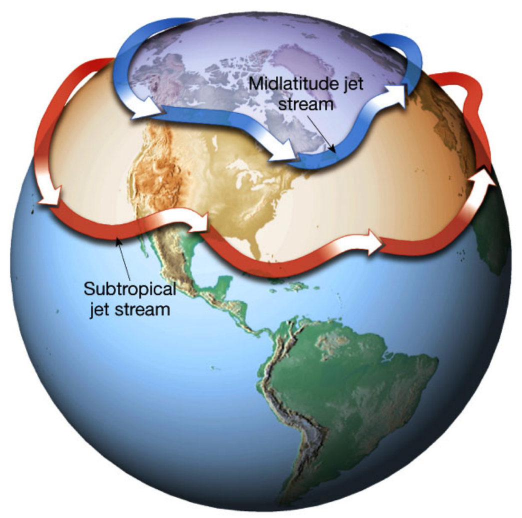


Earth system models redistribute these fluxes across all components
(atmosphere, ocean, land, ice, trees, etc)

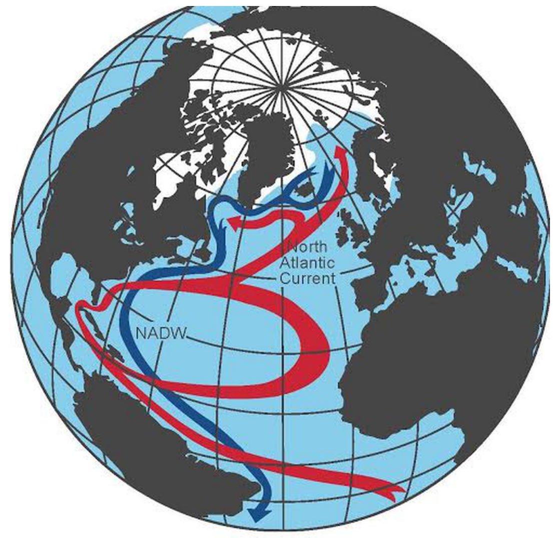
why is climate prediction so hard?

Interconnections among many processes

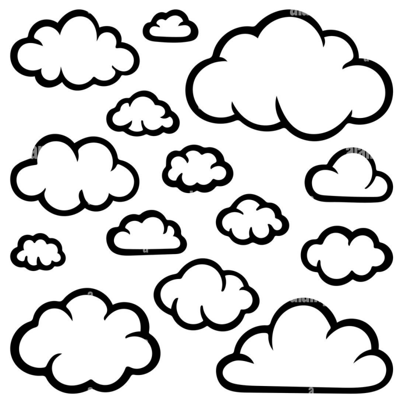
Some things we can't resolve..!
Other things we don't even know the equations!



atmosphere



ocean



clouds

trees

[made by Claire Yung]

ice-ocean interactions

air-sea exchanges

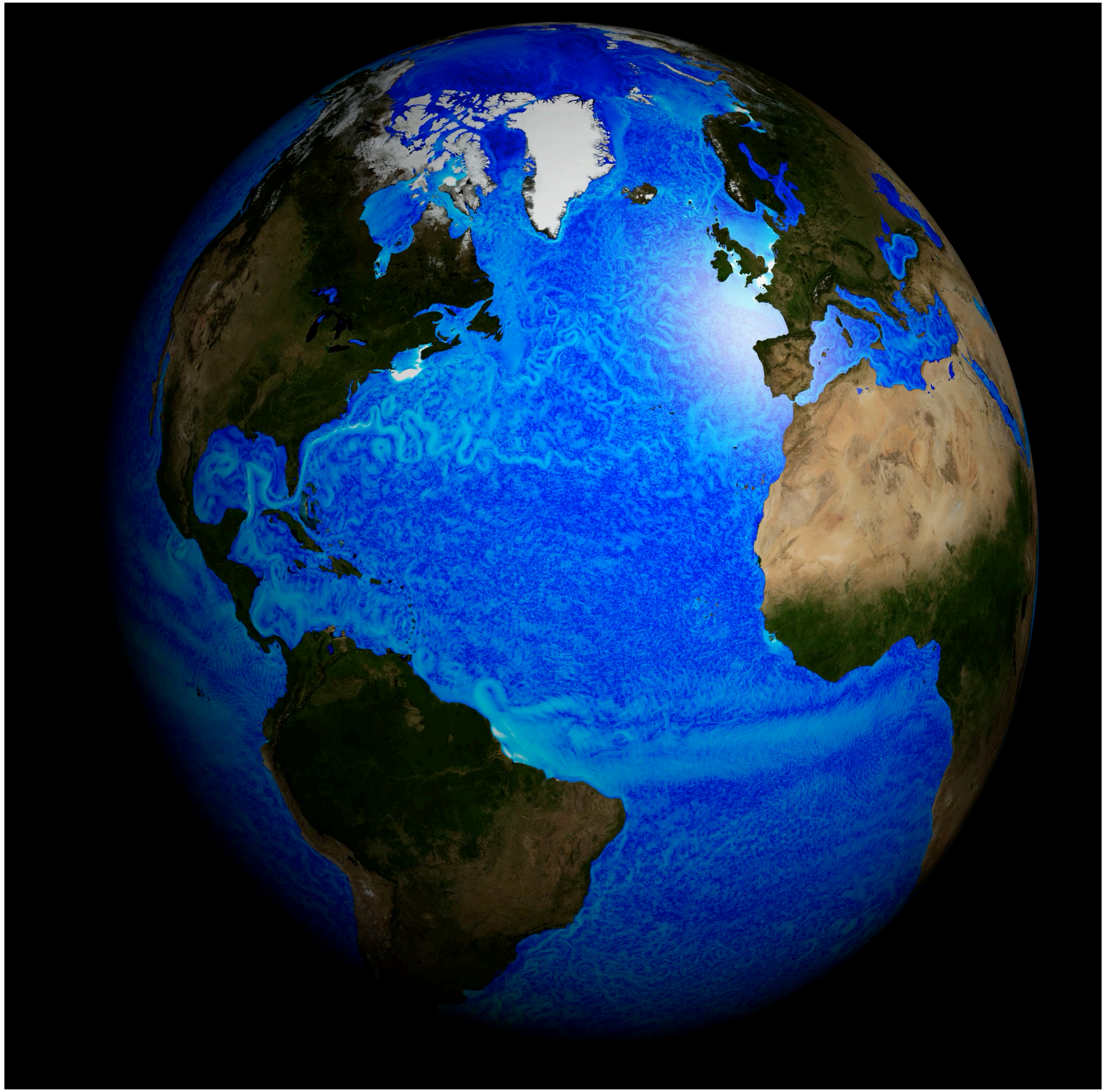
. . .

many more processes

Approximate models for many processes (parameterizations) dominate the uncertainty of climate projections

🌊 ocean: processes at many scales interact

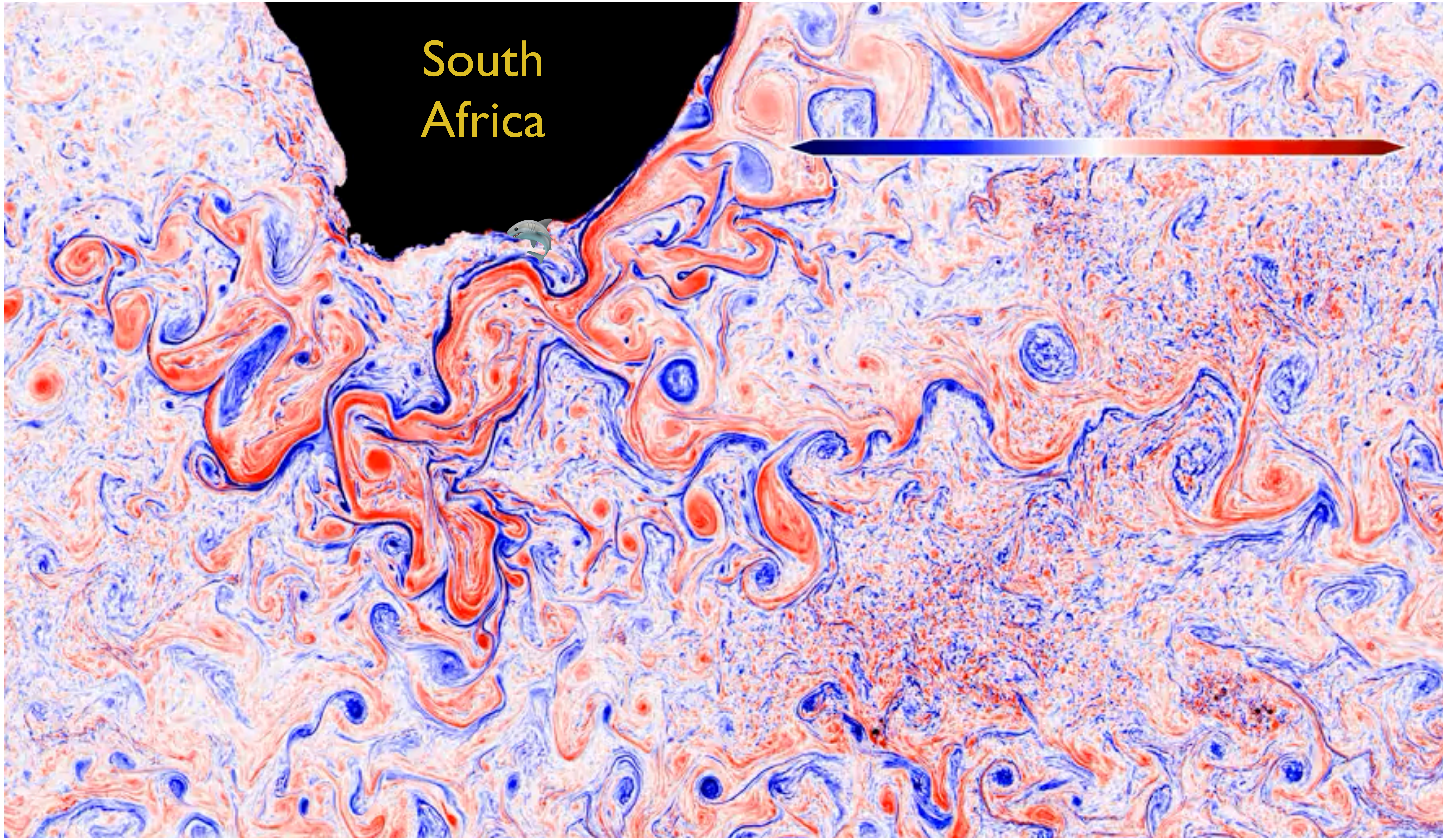
turbulence	~1-100 m ~minutes
eddies	~200 km ~months
currents	~1000 km ~years-decades
overturning circulation	~10.000 km ~decades-centuries



LLC4320 sea surface speed animation
by Henze and Menemenlis (NASA/JPL)
1/48th degree or ~1-2 km & 90 vertical levels
(one of the biggest ocean simulation ever run…)

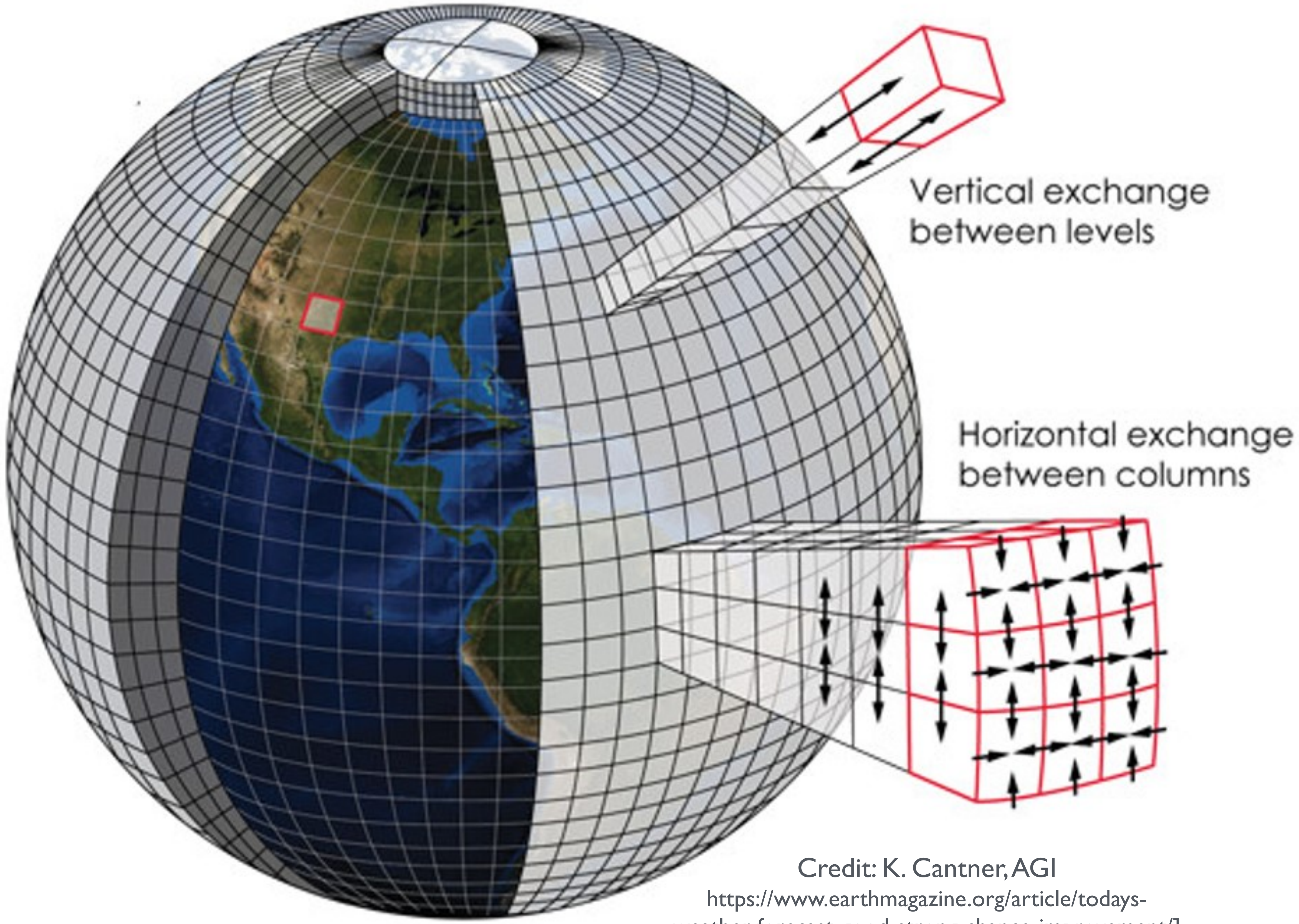
turbulence comes at many scales!

the more we zoom in, the more see…



by C. Henze and D. Menemenlis (NASA/JPL) climate models 👩🏾‍💻 💾

how do we simulate climate? 🌎



Credit: K. Cantner, AGI
https://www.earthmagazine.org/article/todays-weather-forecast-good-strong-chance-improvement/]

Newton's 2nd law
(mass × acceleration = force)
for every grid box!

(Navier−Stokes equations + thermodynamics)

IBM Blue Gene/P supercomputer
with 164,000 processors!

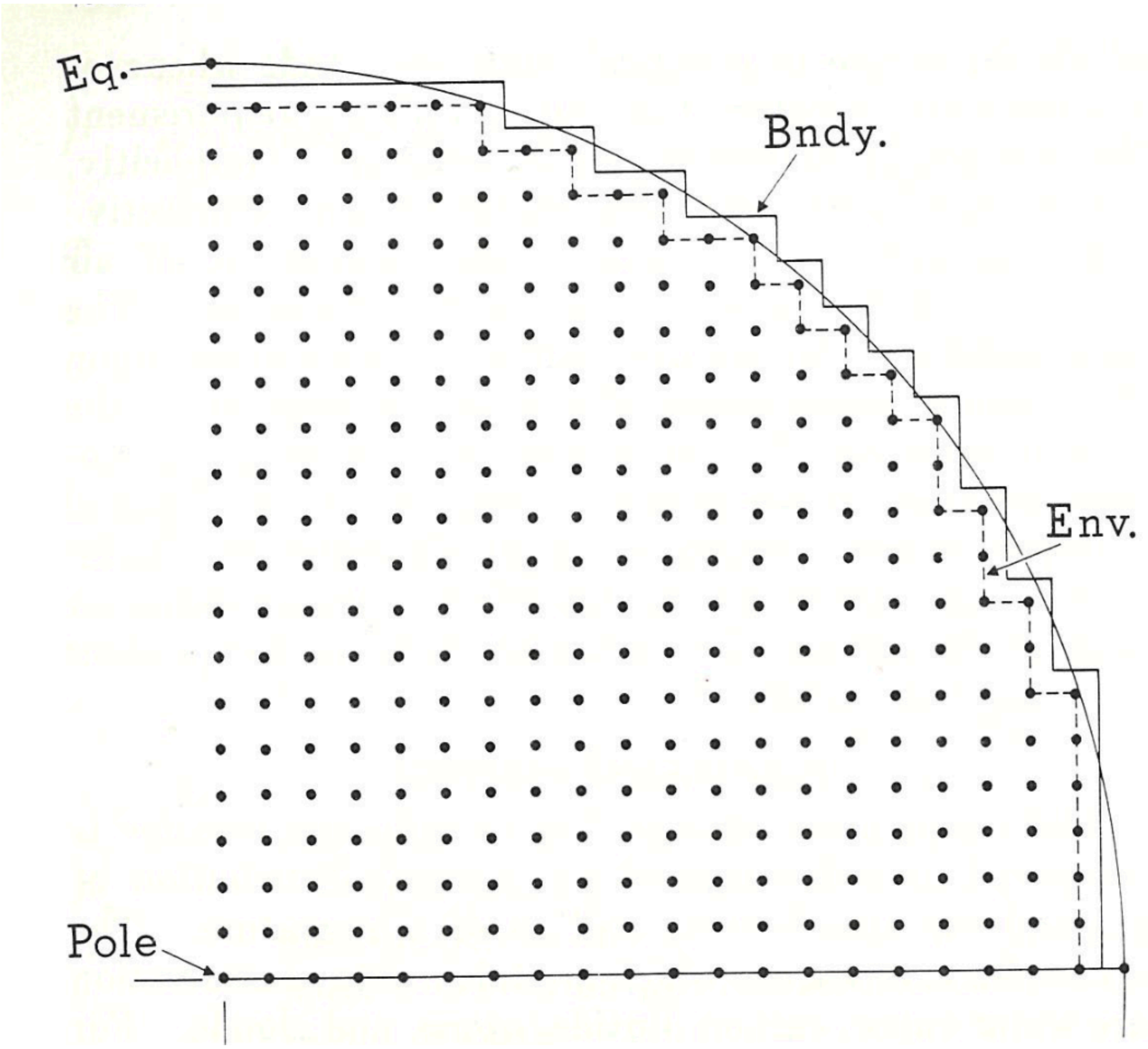
(imagine 164,000 laptops side by side)

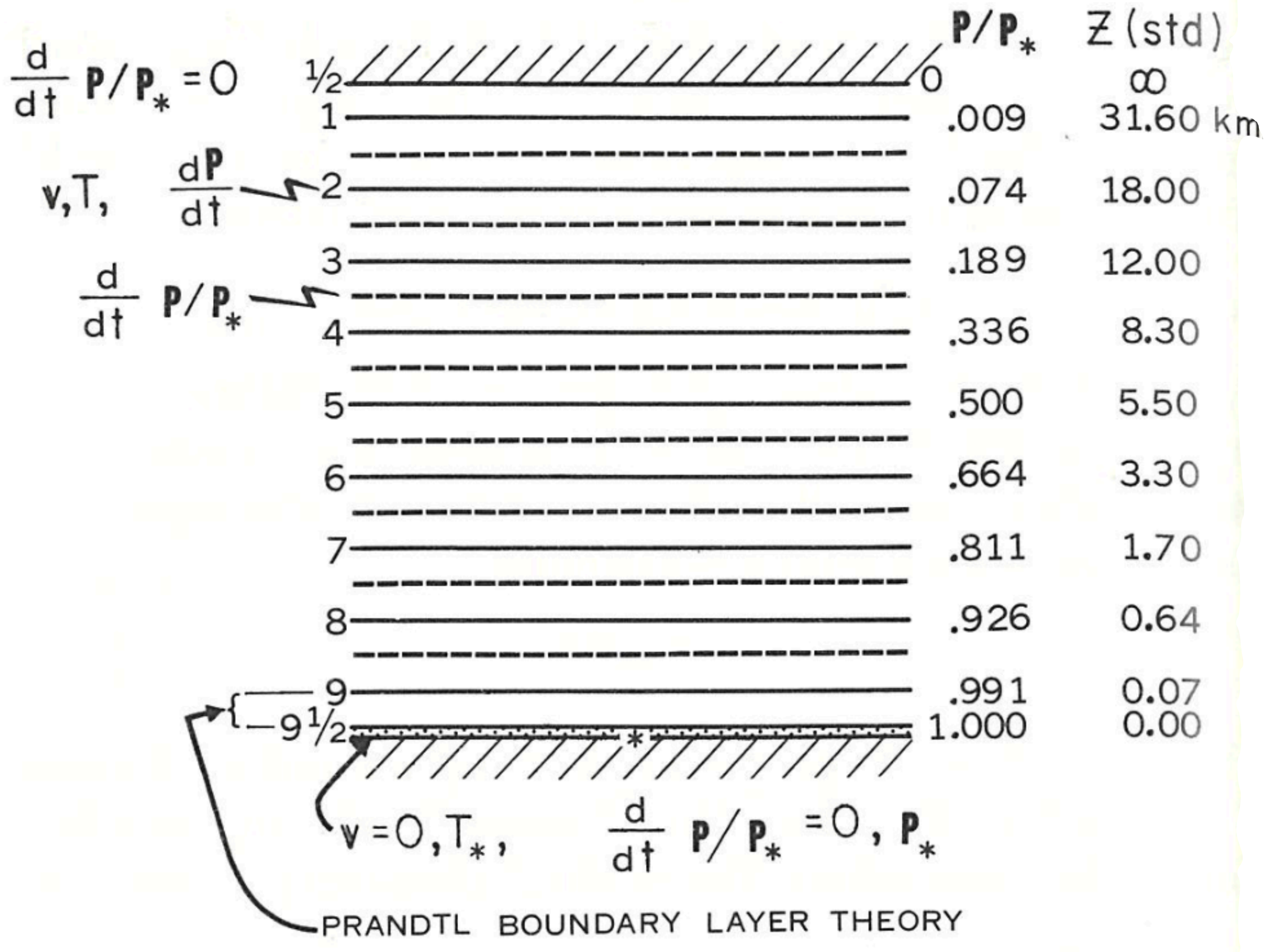
a bit of history of climate modeling

first atmosphere
☁️ model 💾

Joseph Smagorinsky, Suki Manabe, & J. Leith Holloway
Numerical results from a nine-level general circulation model of the atmosphere.

Mon.Wether Rev. 1965





a bit of history of climate modeling

first atmosphere
☁️ model 💾

Joseph Smagorinsky, Suki Manabe, & J. Leith Holloway
Numerical results from a nine-level general circulation model of the atmosphere.

Mon. Wether Rev. 1965

first ocean
🌊 model 💾

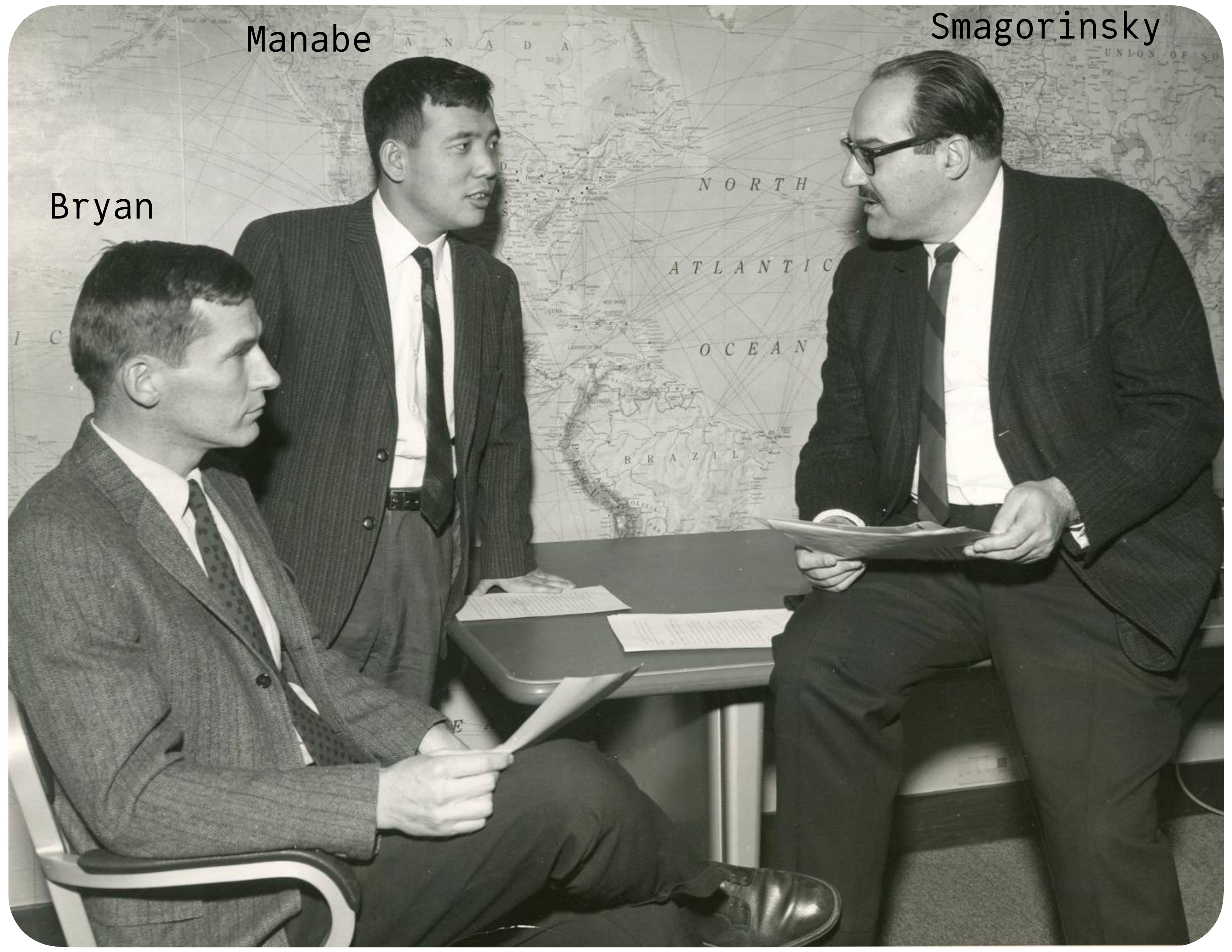
Kirk Bryan & Michael Cox
A numerical investigation of the oceanic general circulation

Tellus 1967
(paper received 1965)

first coupled atmos-ocean
☀️ 🌧️ 🌊 model 💾

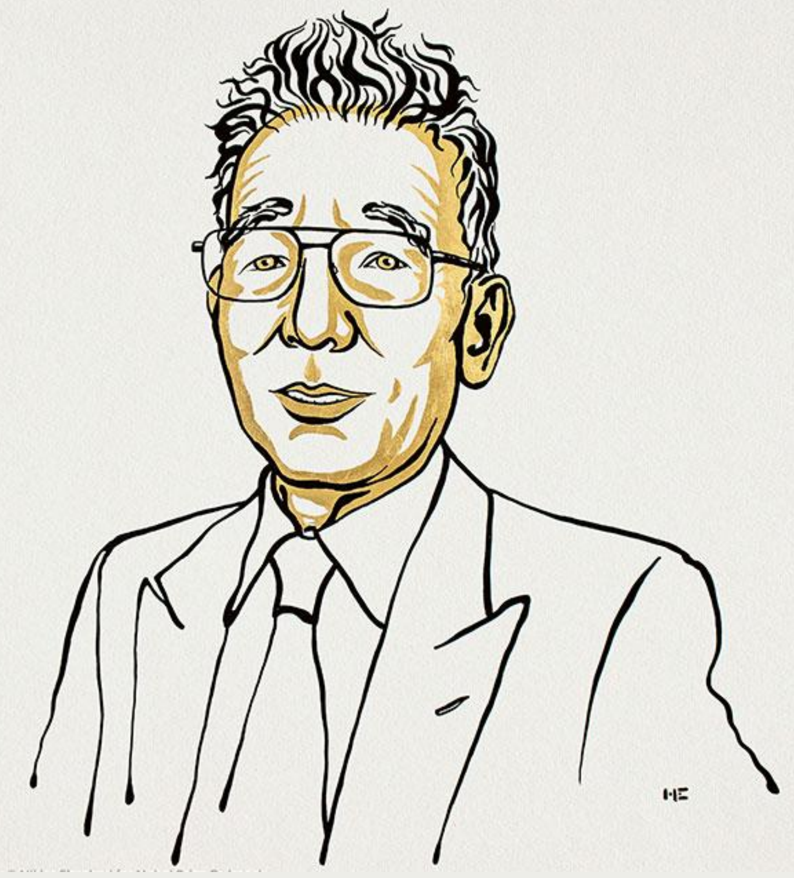
Suki Manabe & Kirk Bryan
Climate calculations with a combined ocean-atmosphere model

J. Atmos. Sci. 1969



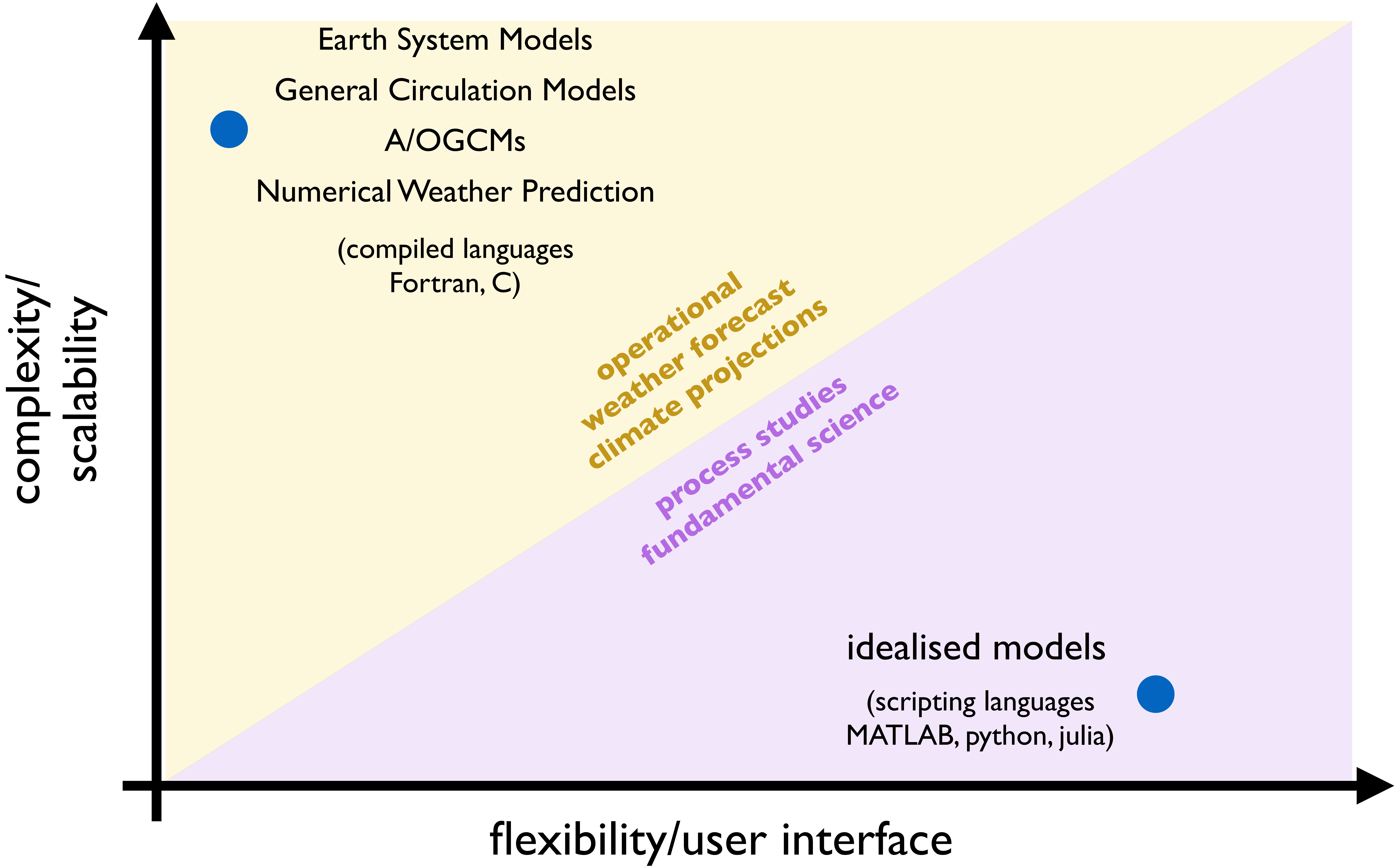
[photo courtesy of GFLD]

Suki Manabe, Nobel Prize Physics 2021

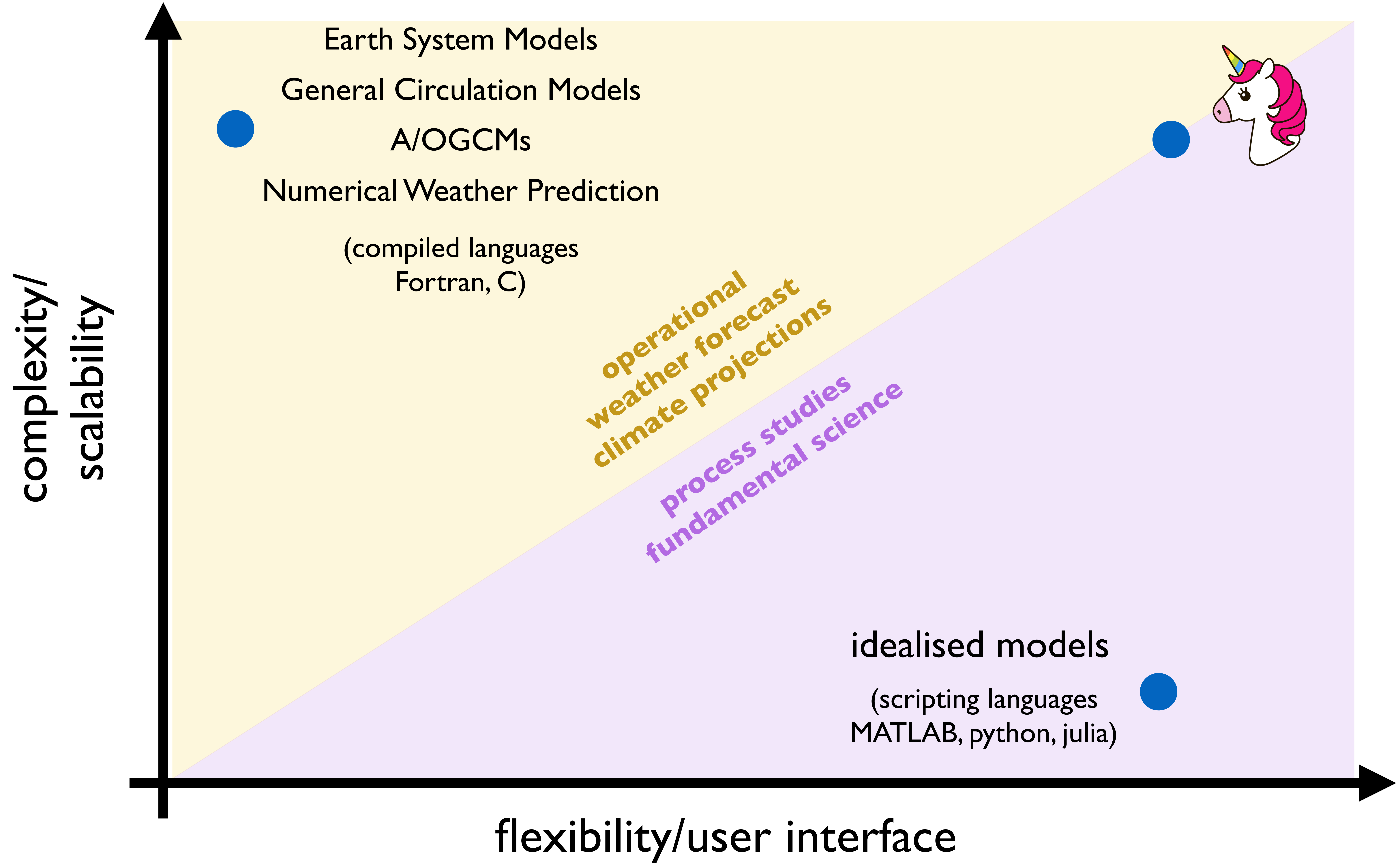


[Credit: Niklas Elmehed for Nobel Prize Outreach]

different motive leads to different model



different motive leads to different model



is user interface important?

internet on mobile phones

all devices have internet capabilities

but, when did people *actually* start using internet on their phones?

[no advertisement intended]

namelist = list of parameter values

allow users to change parameters
without modifying the source code

The NAMELIST input data file specifies the following:

```
NFIRST=1         start from scratch
NLAST=42         stop after timestep 42
NNERGY=20        print fields every 20 timesteps
NTSI=1           print single line of information every timestep
NMIX=10          do a mixing timestep every 10 timesteps
AMF=1.E9         horizontal mixing of momentum
AHF=2.E7         horizontal mixing of heat, salinity, tracers
FKPMF=1.0        vertical mixing of momentum
FKPHF=1.0        vertical mixing of heat, salinity, tracers
DTTSF=172800.    length of timestep on temp., sal., tracers
DTUVF=7200.      length of timestep on internal mode velocities
DTSFF=7200.      length of timestep on stream function
MXSCAN=25        cut off relaxation at 25 scans if not converged
SORF=1.50        coefficient of over-relaxation
CRITF=4.E9       criterion for convergence of relaxation
ACORF=0.5        weight forward and backward timestep equally in
                   the semi-implicit Coriolis term
TINITF=...       initial values of temperature, salinity, tracers
                   (note..  for purposes of precision, salinity is
                    carried in the model in units of parts per part,
                    with .035 subtracted off, however TINITF is ex-
                    pected in parts per part; .035 is subtracted
                    after read-in)
ISIS=13,...      the island and its perimeter points are included
                   in a box between I=13-16 and J=9-12
```

a namelist from the GFDL ocean model back in 1984

Model 1 is a duplicate of the sample model Semtner supplies in his report. It consists of a closed, rectangular basin with a shelf along the western wall and a small island in the center. The update file, NAMELIST input data file, and output file for this model are given on the following pages. The option record specifies that an island is included, that the model is to be run core contained, and that comments are desired in the code listing. The updates which follow simply alter the base code to yield a model like that of Semtner's sample. Note that many of the updates arise because an alternate equation of state is used. There are differences in the manner in which the computation is handled between the two models, so that the answers will not be exactly the same. In particular, Semtner's version uses a relaxation which does not converge as rapidly as that used here, and, in fact, is cut off on most timesteps at 26 scans, before convergence occurs. In the present model, the convergence is completed for all but a few timesteps. The result is that the answers are somewhat different but presumably more accurate in the present model.

Cox (1984), GFDL Ocean Group Technical Report No. 1, GFLD/NOAA

namelist = list of parameter values

allow users to change parameters
without modifying the source code

```
NIGLOBAL = 1440                 !
                                ! The total number of thickness grid points in the x-direction in the physical
                                ! domain. With STATIC_MEMORY_ this is set in MOM_memory.h at compile time.
NJGLOBAL = 450                  !
                                ! The total number of thickness grid points in the y-direction in the physical
                                ! domain. With STATIC_MEMORY_ this is set in MOM_memory.h at compile time.
! LAYOUT = 24, 12               !
                                ! The processor layout that was actually used.

! === module MOM ===
THICKNESSDIFFUSE_FIRST = True   !   [Boolean] default = False
                                ! If true, do thickness diffusion before dynamics. This is only used if
                                ! THICKNESSDIFFUSE is true.
DT = 600.0                      !   [s]
                                ! The (baroclinic) dynamics time step.  The time-step that is actually used will
                                ! be an integer fraction of the forcing time-step (DT_FORCING in ocean-only mode
                                ! or the coupling timestep in coupled mode.)
DT_THERM = 3600.0               !   [s] default = 600.0
                                ! The thermodynamic and tracer advection time step. Ideally DT_THERM should be
                                ! an integer multiple of DT and less than the forcing or coupling time-step,
                                ! unless THERMO_SPANS_COUPLING is true, in which case DT_THERM can be an integer
                                ! multiple of the coupling timestep.  By default DT_THERM is set to DT.
```

a namelist from an
IPCC-class GCM
ocean model

github.com/NOAA-GFDL/MOM6-examples/blob/dev/gfdl/ocean_only/MESO_025_23L/MOM_input

parameters.namelist

```
Nx = 360                                  ! Number of points in longitude
Ny = 180                                  ! Number of points in latitude

force_with_dataset = True                 ! JRA55 atmospheric reanalysis dataset
```

what if we want to add surface heating fluxes $SHF_{\mathrm{surf}} = Q_0 \cos(\mathrm{lat})$?

parameters.namelist

```
Nx = 360                                  ! Number of points in longitude
Ny = 180                                  ! Number of points in latitude

force_with_dataset = False                ! JRA55 atmospheric reanalysis dataset

force_with_coslat = True                  ! Surface heat flux ~ cos(lat)
amplitude_heating_forcing = 340           ! W/m^2, only if force_with_coslat = True
```

what if we want to add surface heating fluxes $SHF_{\text{surf}} = Q_0 \cos(\text{lat})$?

given that such thing is
implemented in main codebase

setting up and running a simulation

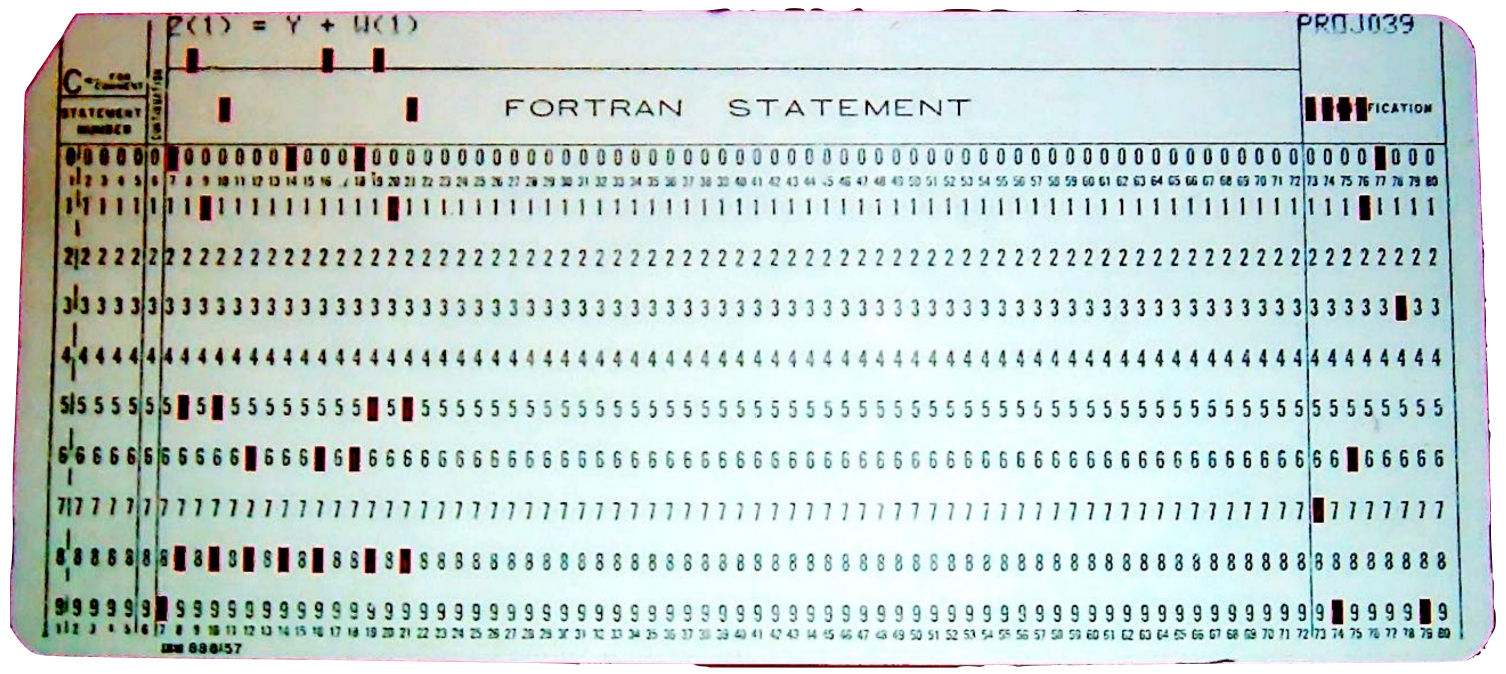
ocean_simulation.script

```
Nx = 360                                  # Number of points in longitude
Ny = 180                                  # Number of points in latitude

grid = GlobalGrid(Nx, Ny)

realistic_atmosphere = DataSet(path = "/data/atmospheric_reanalysis/")

idealised_atmosphere = IdealizedAtmosphere(grid)
set!(idealized_atmosphere, heat_flux = 340 * cos(deg2lat(grid.latitude)))

model = OceanModel(atmosphere = idealized_atmosphere, ...)

...
```

[disclaimer: pseudo code☝🏽]

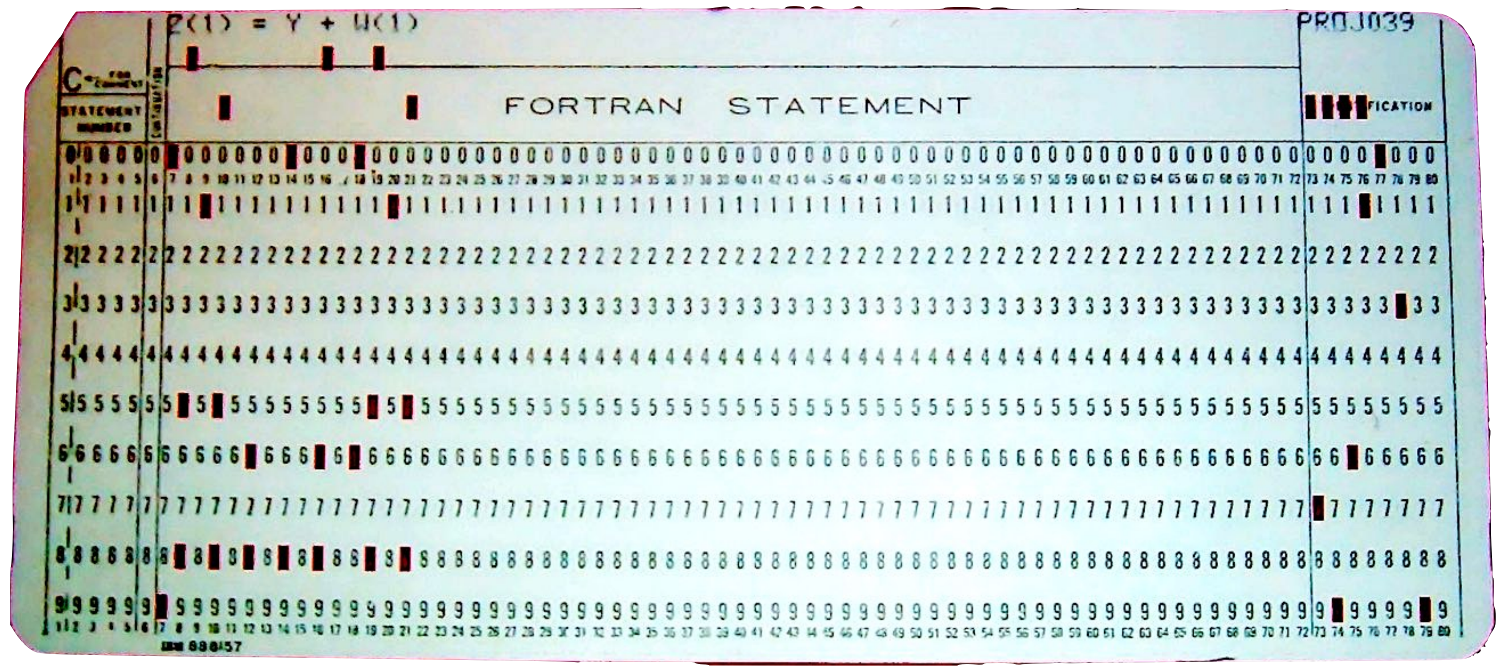
But we'd like more than namelists…

models should be easy and fun!



But we'd like more than namelists…

models should be easy and fun!



[by Dall-E]

But we'd like more than namelists…

models should be easy and fun!

what if

want to switch model components in and out?

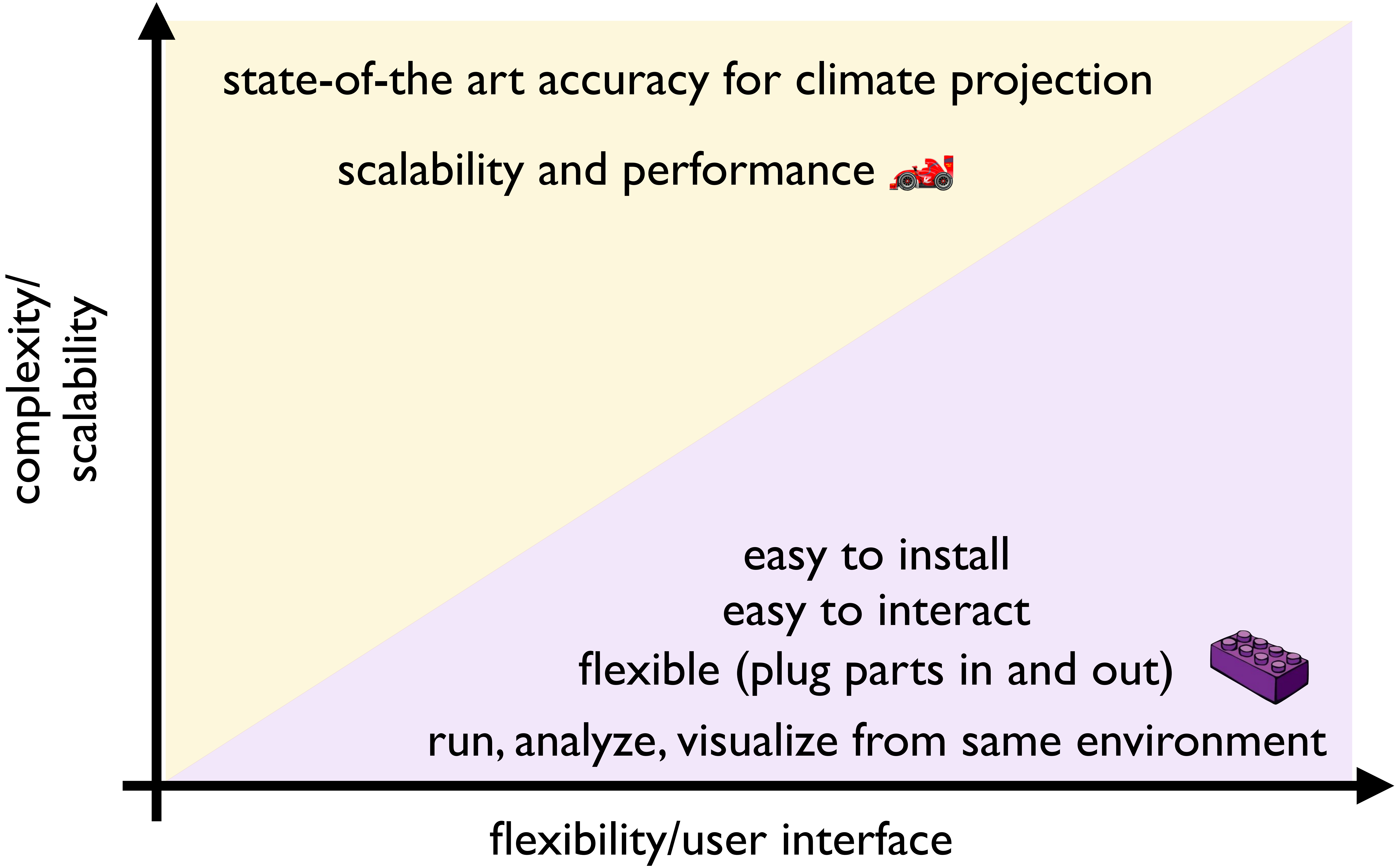
modify forcing, boundary conditions, bathymetry ⛰️,…?

wanna share with other to reproduce your results?

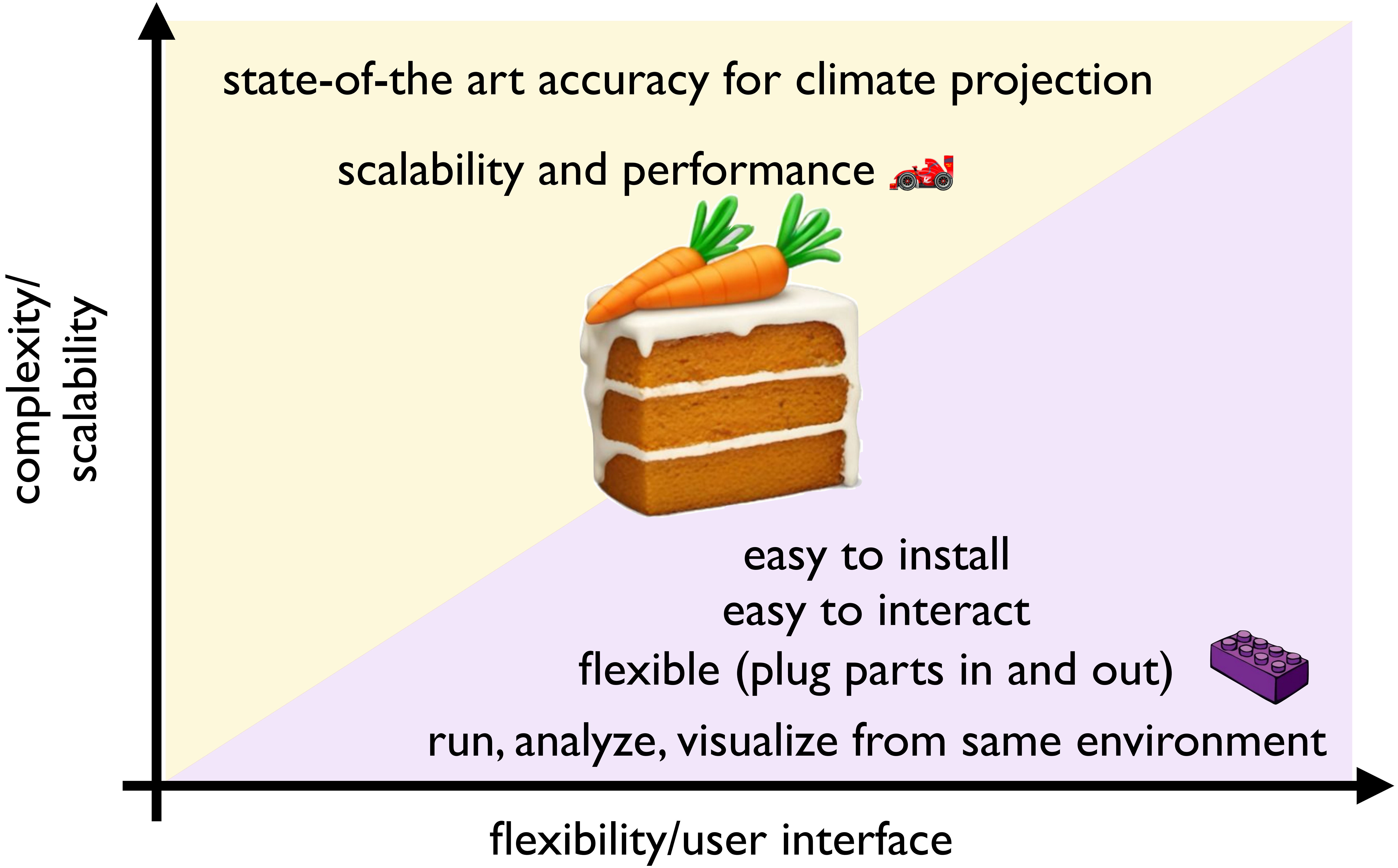
✅

modern software 🛸
& its impact on our work flow in
research 👩🏽‍🔬 and teaching 🧑🏽‍🎓

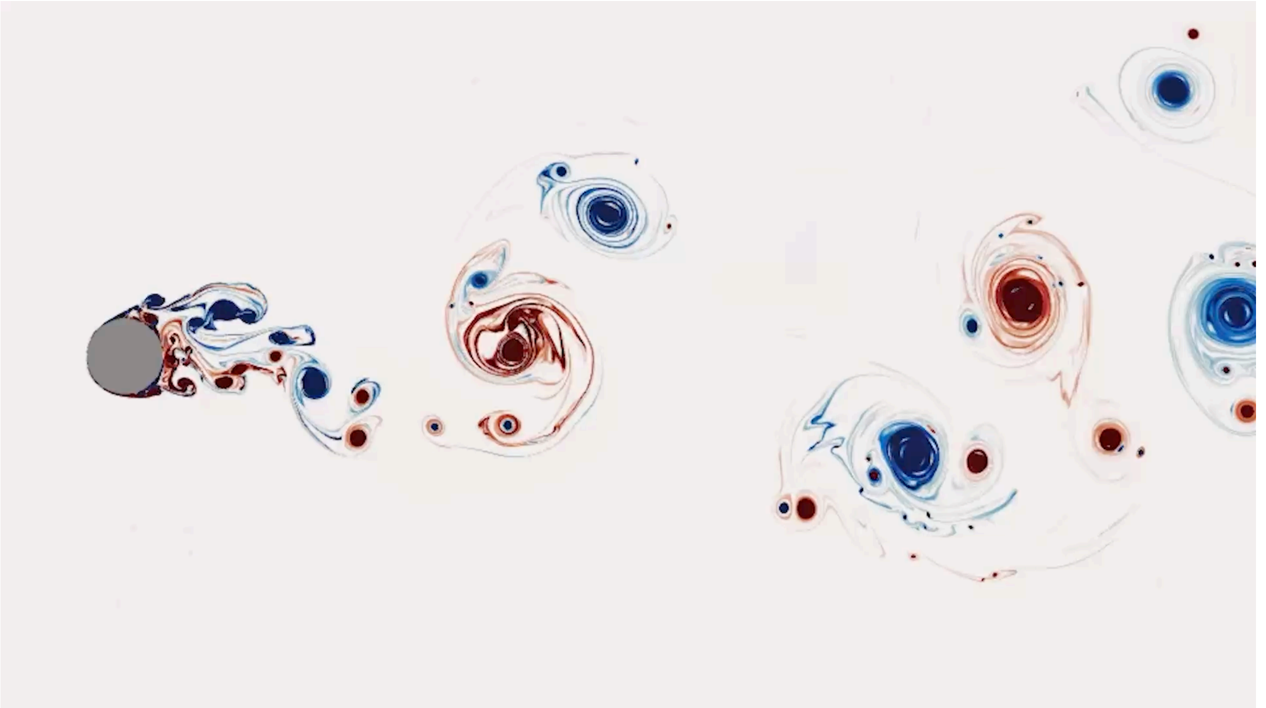
ocean model


state-of-the art accuracy for climate projection
scalability and performance
easy to install
easy to interact
flexible (plug parts in and out)
run, analyze, visualize from same environment
complexity/
scalability
flexibility/user interface

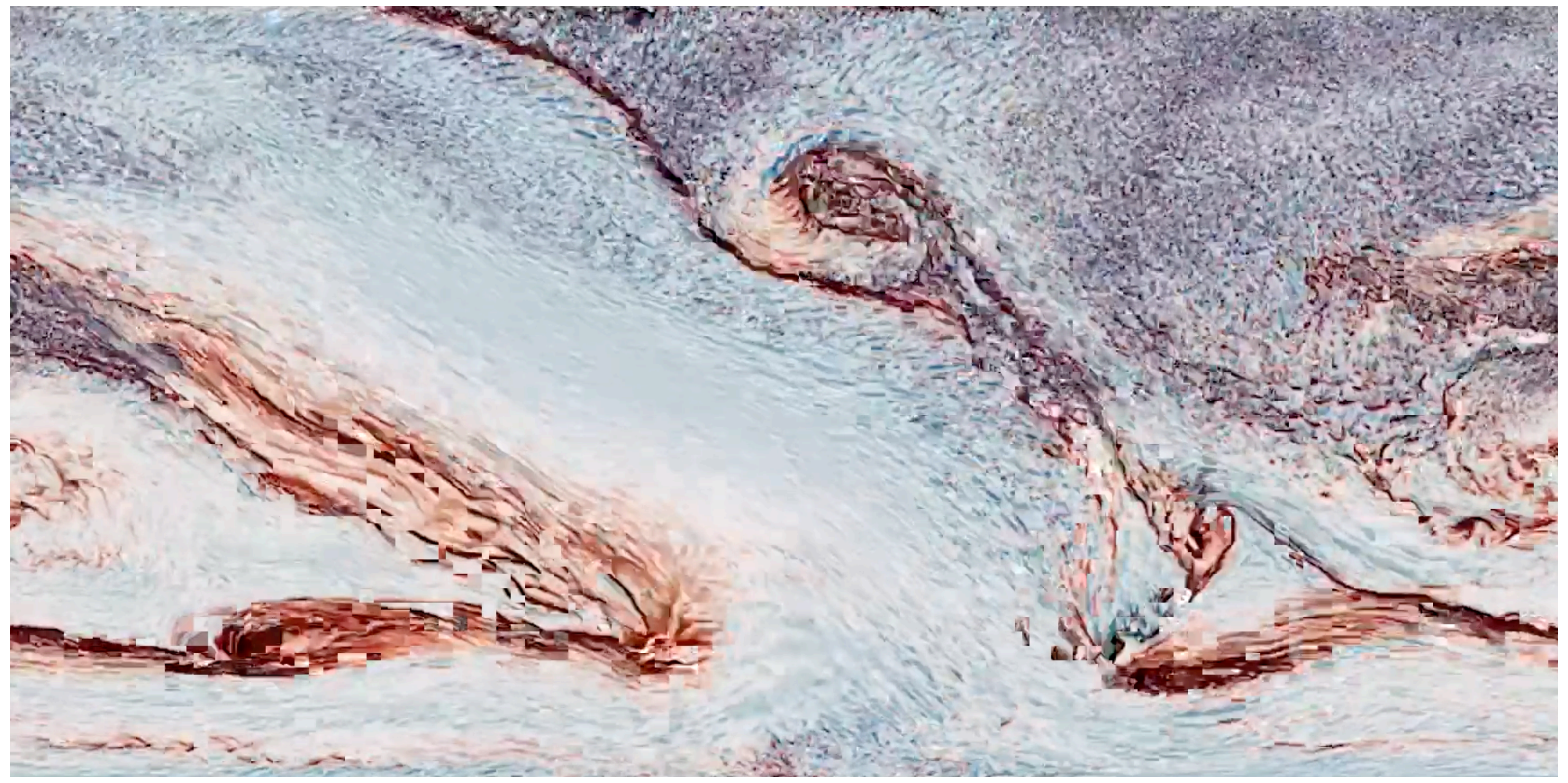
ocean model


state-of-the art accuracy for climate projection
scalability and performance
easy to install
easy to interact
flexible (plug parts in and out)
run, analyze, visualize from same environment
complexity/
scalability
flexibility/user interface

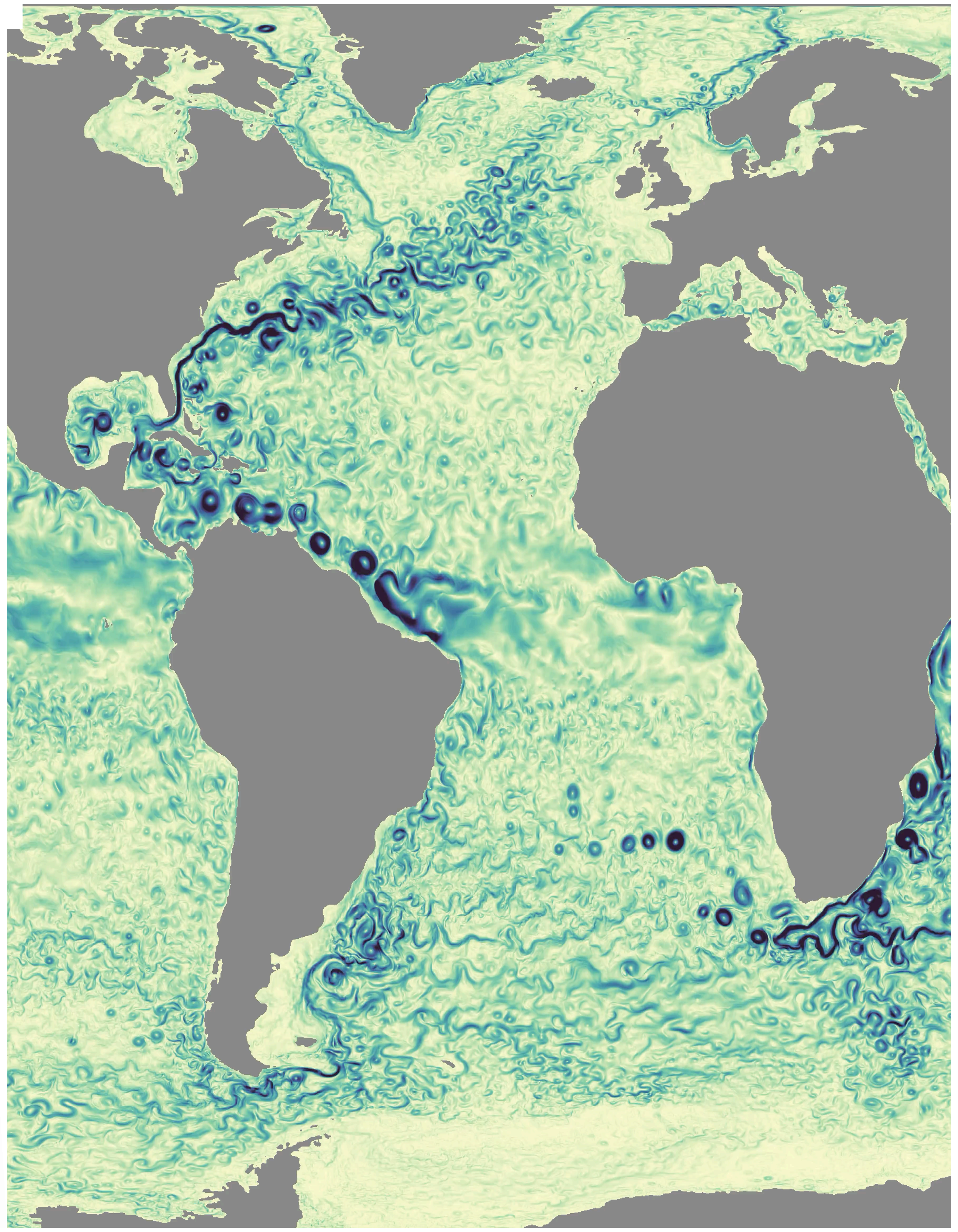
direct numerical simulation (DNS) ⟷ large eddy simulation (LES) ⟷ general circulation modeling (GCM)



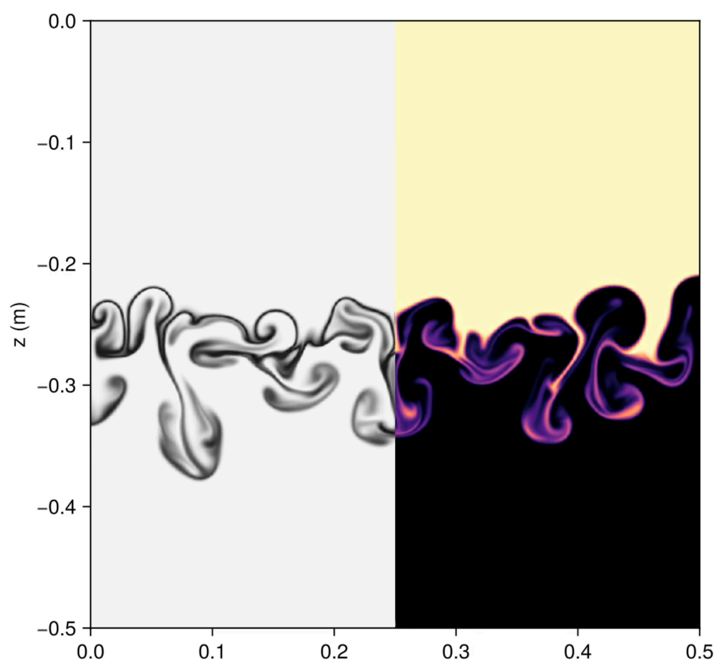
DNS of 2D flow around a cylinder



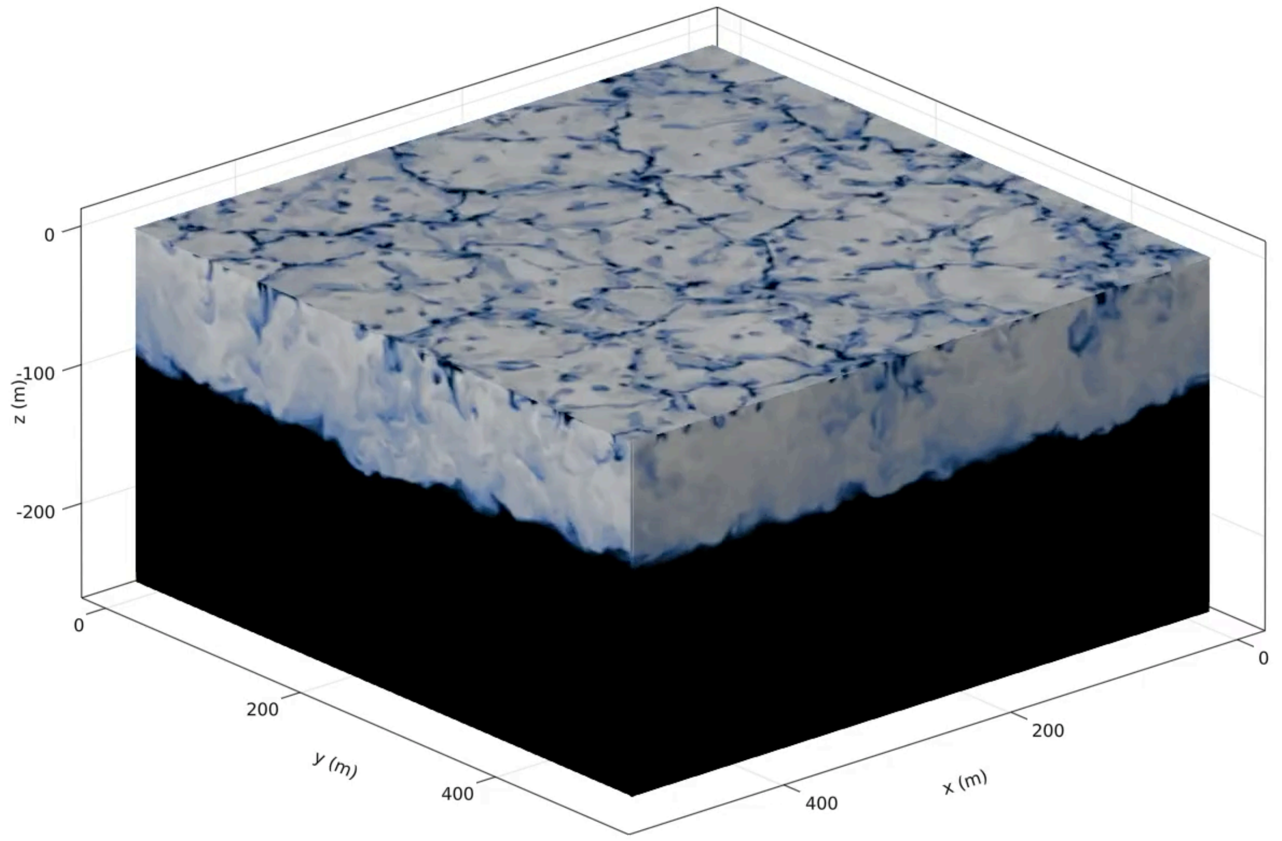
mixed-layer instability LES



near-global simulation
@ 1/12th degree horizontal resolution



DNS of cabbeling in freshwater



Free convection LES

regional hydrostatic simulation

by ClimaOcean dev team
(mostly Greg Wagner & Simone Silvestri)

Oceananigans in a nutshell

- Software for ocean-flavored fluid dynamics using finite-volume numerical methods
- **Fast**: written from scratch for GPUs
- **Friendly**: uses the Julia programming language
  - *Simple simulations are easy*
  - *Complex, creative simulations are possible*
- **Flexible**
  - *Companion packages for biogeochemistry, sea ice, ocean-sea-ice coupled simulations*


Greg Wagner


Simone Silvestri

Many Contributors

I. decaying 2D turbulence

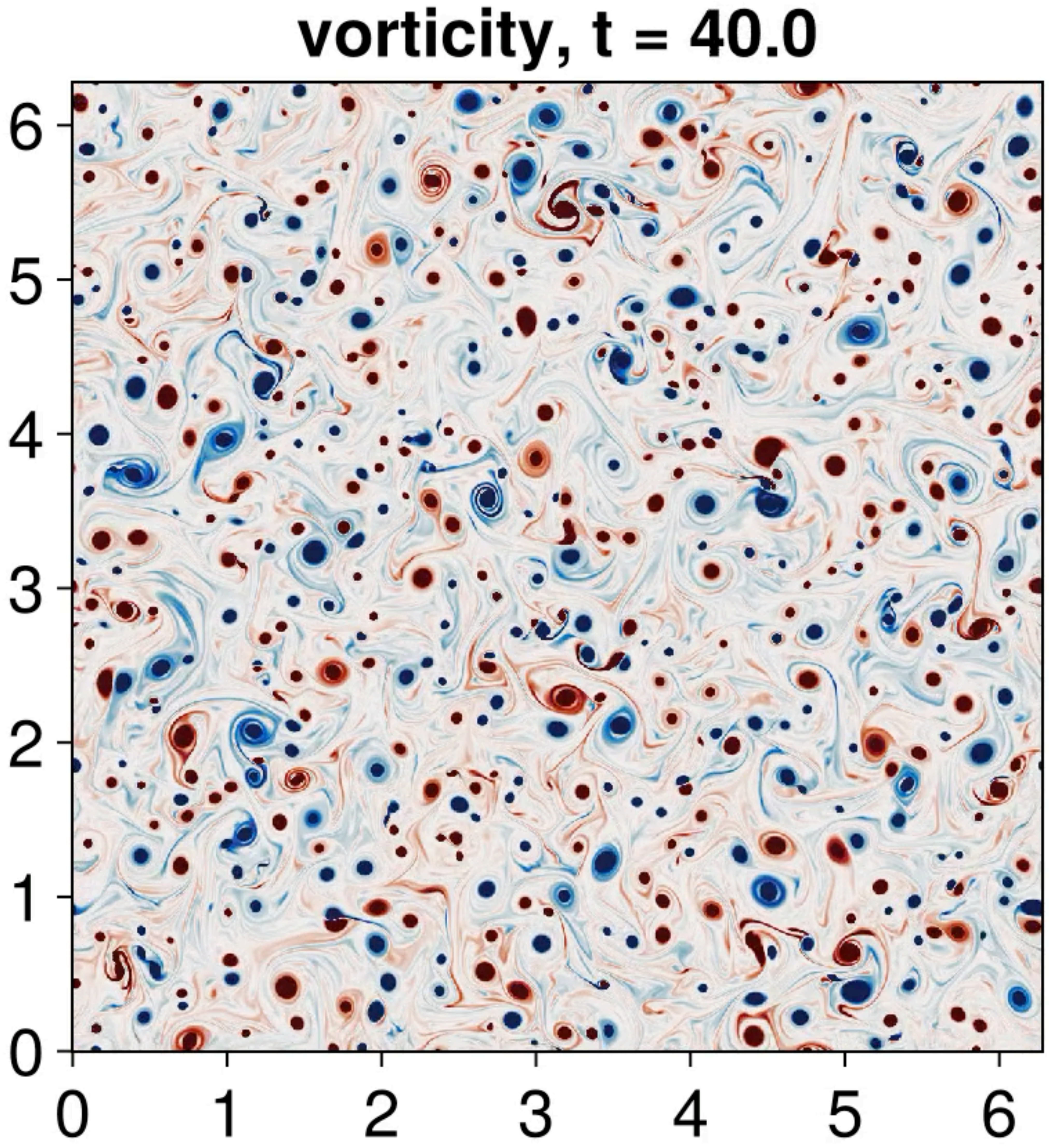
II. baroclinic instability on the sphere

III. near-global ocean simulation
with continents, realistic atmosphere, and whatnot

all scripts/notebooks are available @ `github.com/navidcy/SeanHaneySymposium2025`

two_dimensional_turbulence.jl

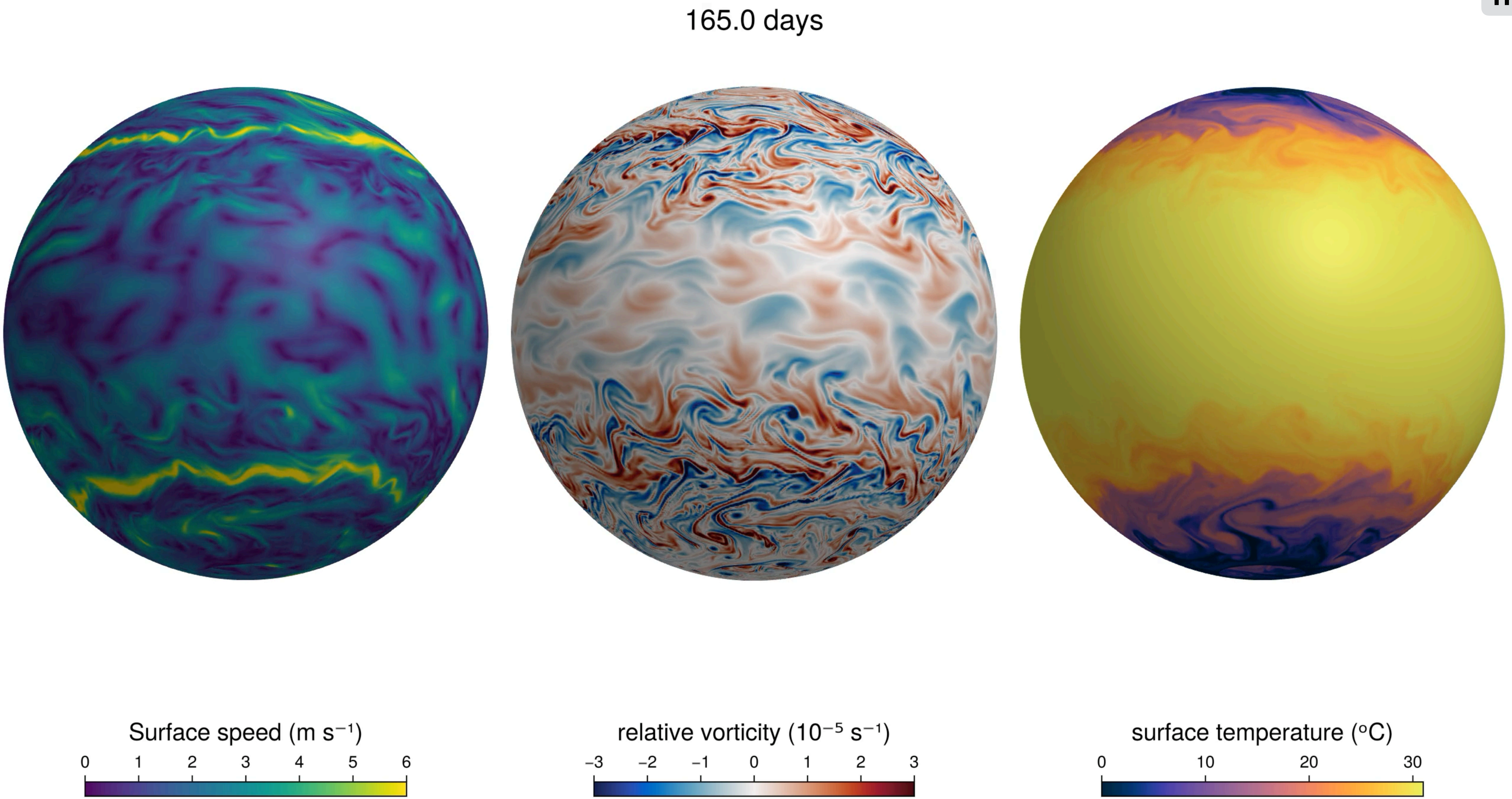
```
grid = RectilinearGrid(size=(2048, 2048),
                       x=(0, 2π), y=(0, 2π),
                       topology = (Periodic, Periodic))

model = NonhydrostaticModel(; grid)

ϵ(x, y) = 2 * rand() - 1 # Uniformly-distributed random numbers ∈ [-1, 1)

set!(model, u=ϵ, v=ϵ)

u, v = model.velocities
ζ = ∂x(v) - ∂y(u)

heatmap(ζ)
```



II.

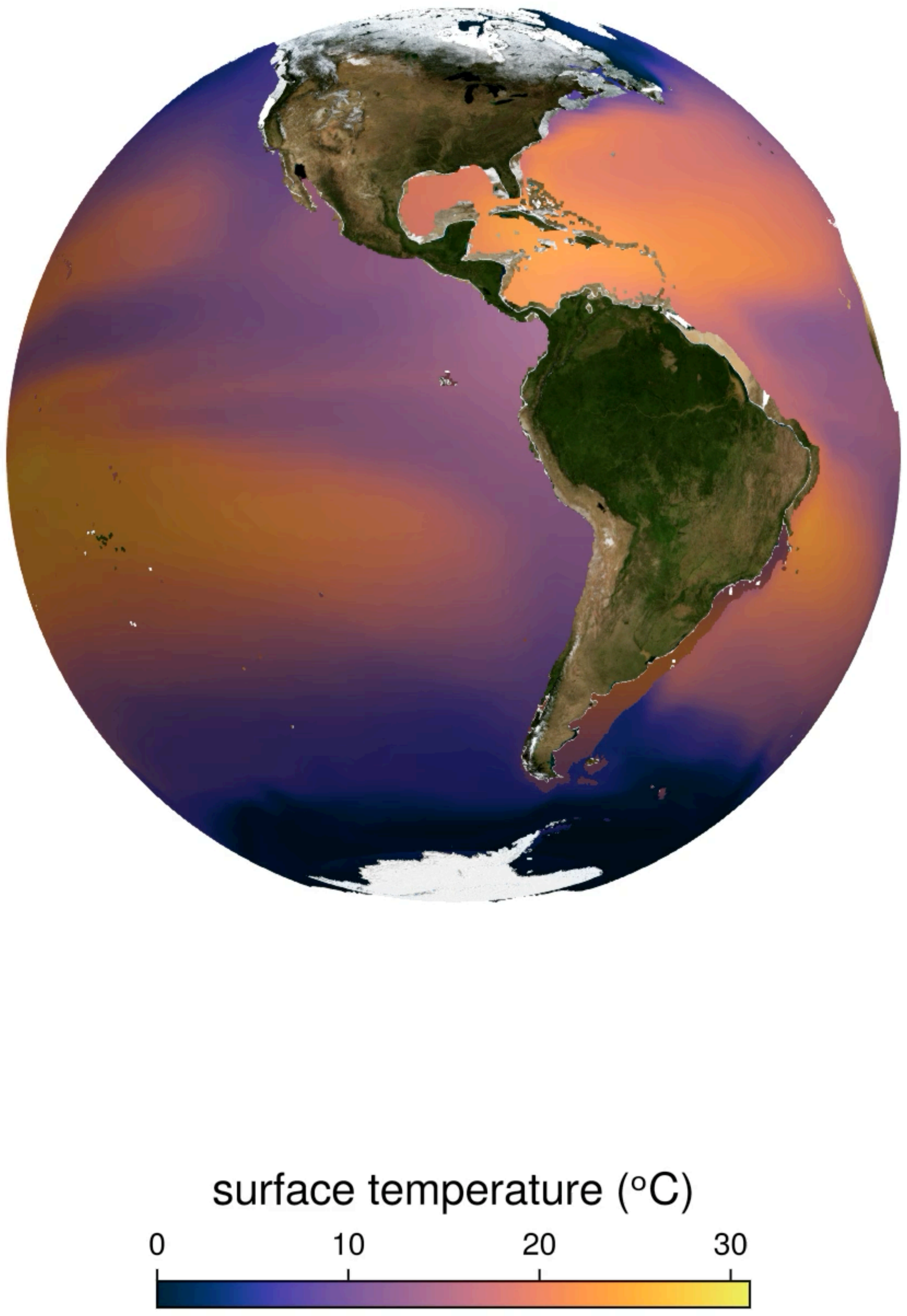
baroclinic_turbulence_sphere.jl

```
Nx, Ny, Nz = 720, 160, 10  # 1/2 degree horizontal resolution

grid = LatitudeLongitudeGrid(GPU(); size = (Nx, Ny, Nz),
                                     latitude = (-80, 80),
                                     longitude = (0, 360),
                                     z = (-3000, 0))

buoyancy = SeawaterBuoyancy(equation_of_state=TEOS10EquationOfState())

model = HydrostaticFreeSurfaceModel(; grid, buoyancy,
                                      coriolis = HydrostaticSphericalCoriolis(),
                                      tracers = (:T, :S),
                                      momentum_advection = WENOVectorInvariant(order=9),
                                      tracer_advection = WENO(order=7))

Tᵢ(λ, φ, z) = 30 * (1 - tanh((abs(φ) - 45) / 8)) / 2 + rand()  # initial temperature
Sᵢ(λ, φ, z) = 28 - 5e-3 * z + rand()                            # initial salinity

set!(model, T=Tᵢ, S=Sᵢ)

simulation = Simulation(model, Δt=5minutes, stop_time=200days)

run!(simulation)
```


165.0 days
Surface speed (m s^{-1})
0 1 2 3 4 5 6
relative vorticity (10^{-5} s^{-1})
−3 −2 −1 0 1 2 3
surface temperature (°C)
0 10 20 30

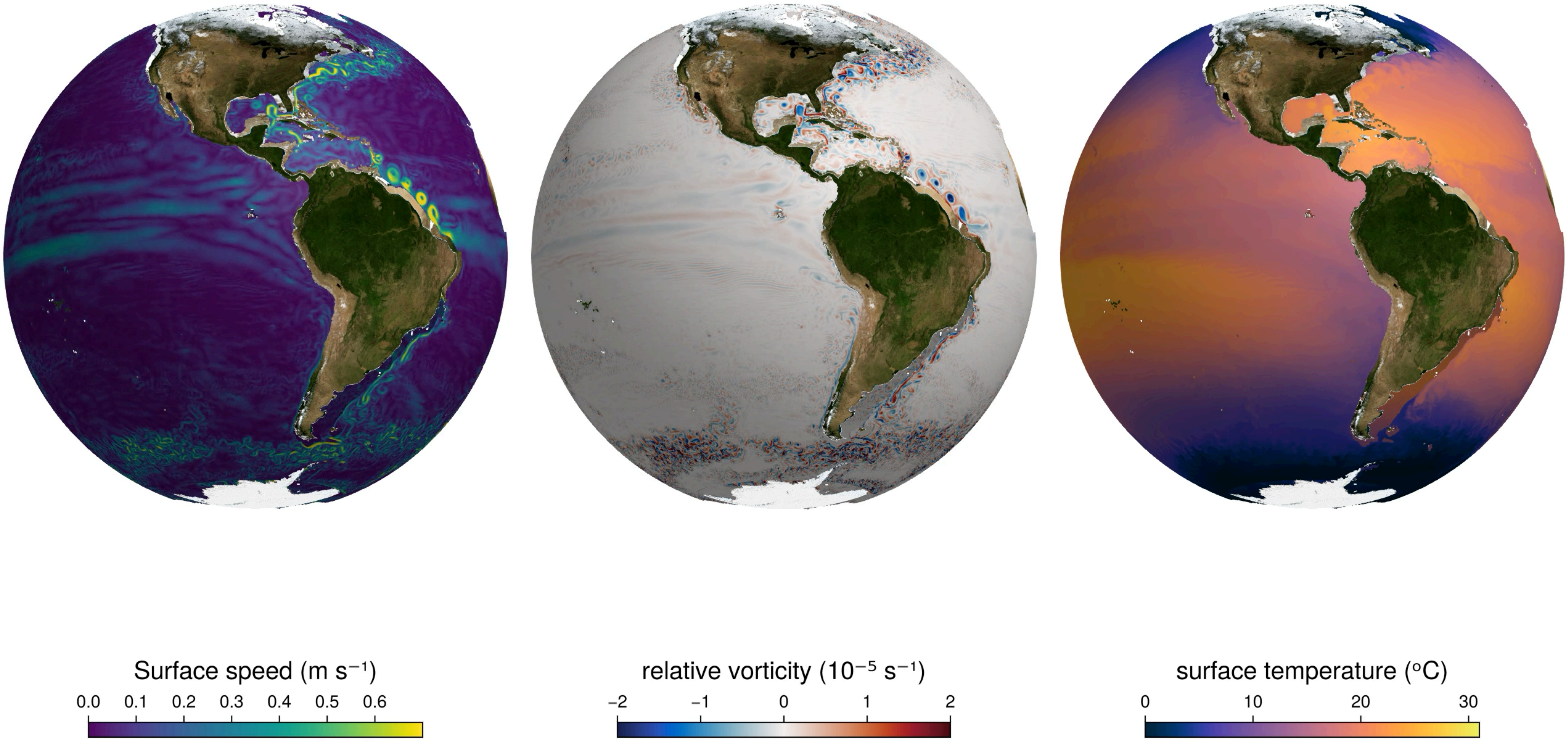
near_global_ocean_simulation.jl

```julia
Nx, Ny, Nz = 1440, 560, 10  # 1/4 degree horizontal resolution

grid = LatitudeLongitudeGrid(GPU(); size = (Nx, Ny, Nz),
                                    latitude = (-70, 70),
                                    longitude = (0, 360),
                                    z = (-3000, 0))

bathymetry = ClimaOcean.regrid_bathymetry(grid) # build gridded bathymetry based on ETOPO1

grid = ImmersedBoundaryGrid(grid, GridFittedBottom(bathymetry))

# build an ocean simulation initialized to the ECCO state estimate on Jan 1, 1993
ocean = ClimaOcean.ocean_simulation(grid)
dates = DateTimeProlepticGregorian(1993, 1, 1)

set!(ocean.model, T = ClimaOcean.ECCOMetadata(:temperature; dates),
                  S = ClimaOcean.ECCOMetadata(:salinity; dates))

# prescribed atmosphere from JRA55 reanalysis
atmosphere = ClimaOcean.JRA55PrescribedAtmosphere(arch)

coupled_model = ClimaOcean.OceanSeaIceModel(ocean; atmosphere)

simulation = Simulation(coupled_model, Δt=5minutes, stop_time=30days)

run!(simulation)
```



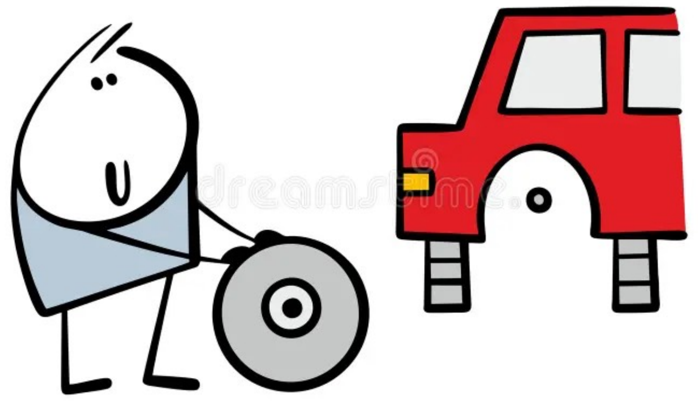
III.

165.0 days



Modeling software 

switch model components in and out easy



modify forcing, boundary conditions,

```
div = ∂x(u) + ∂y(v) + ∂z(w)

set!(windstress = cos(latitude)²)

save(abs(∇T)², “temperature_dissipation.nc”)
```

code that resembles maths/equations

easy to share with other to reproduce results

easy to visualise

[drawing by Alireza Karimi Moghaddam; cartoonist, illustrator)]

Modeling software

anything to take home? 🧳

the ocean is beautiful with
mesmerizing flow patterns!

models are our "eyes" to the future
(climate projections)
+ indispensable for research & teaching

software and science go hand in hand 🤝

but the effort we put in software is very small
($\ll 1$)

[drawing by Alireza Karimi Moghaddam; cartoonist, illustrator)]

cute 🦭… but does it seem tad angry?

because we don’t invest much effort in software for ocean science 🌊 …