Programming for the Planet

Dominic Orchard

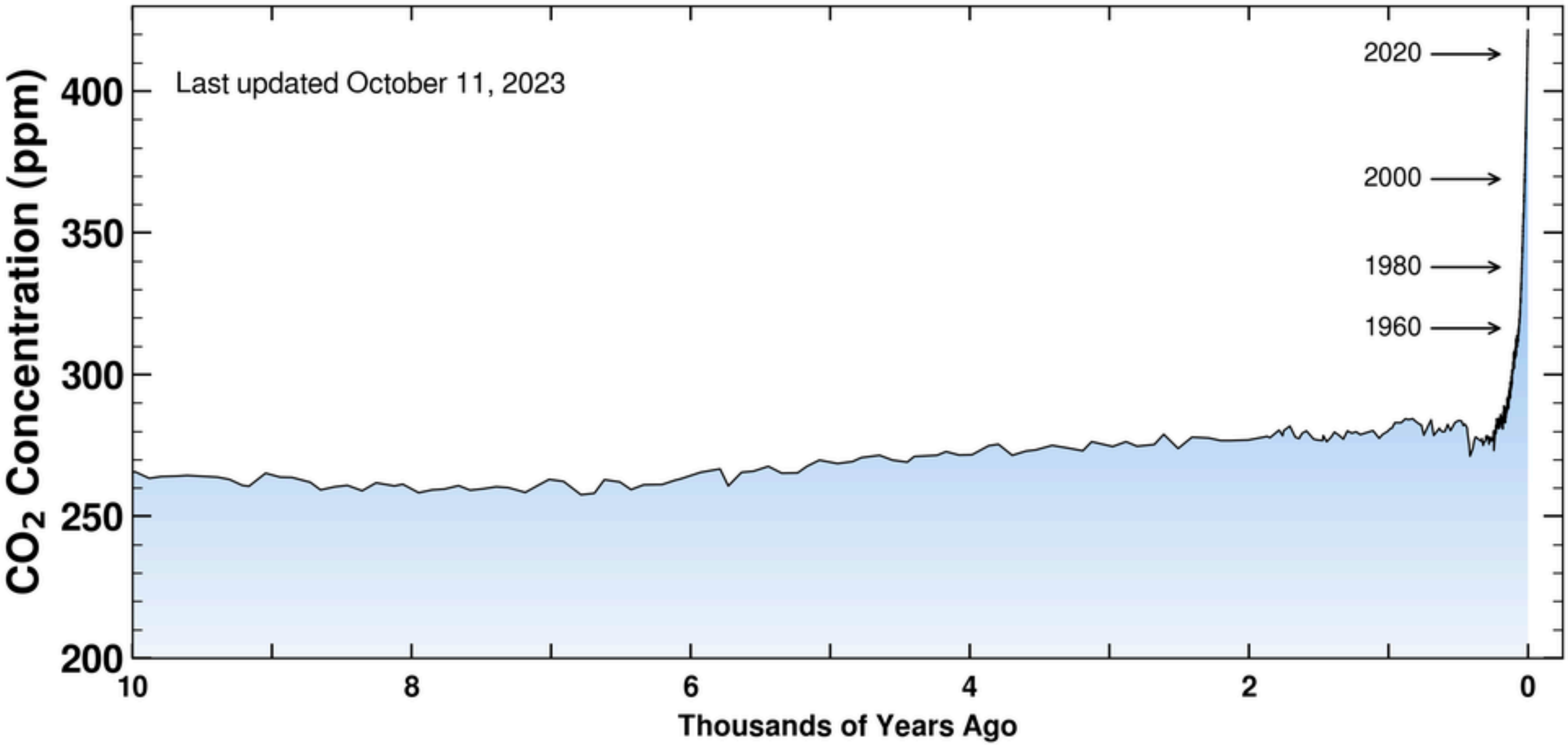
UNIVERSITY OF CAMBRIDGE

Institute of Computing for Climate Science

University of Kent

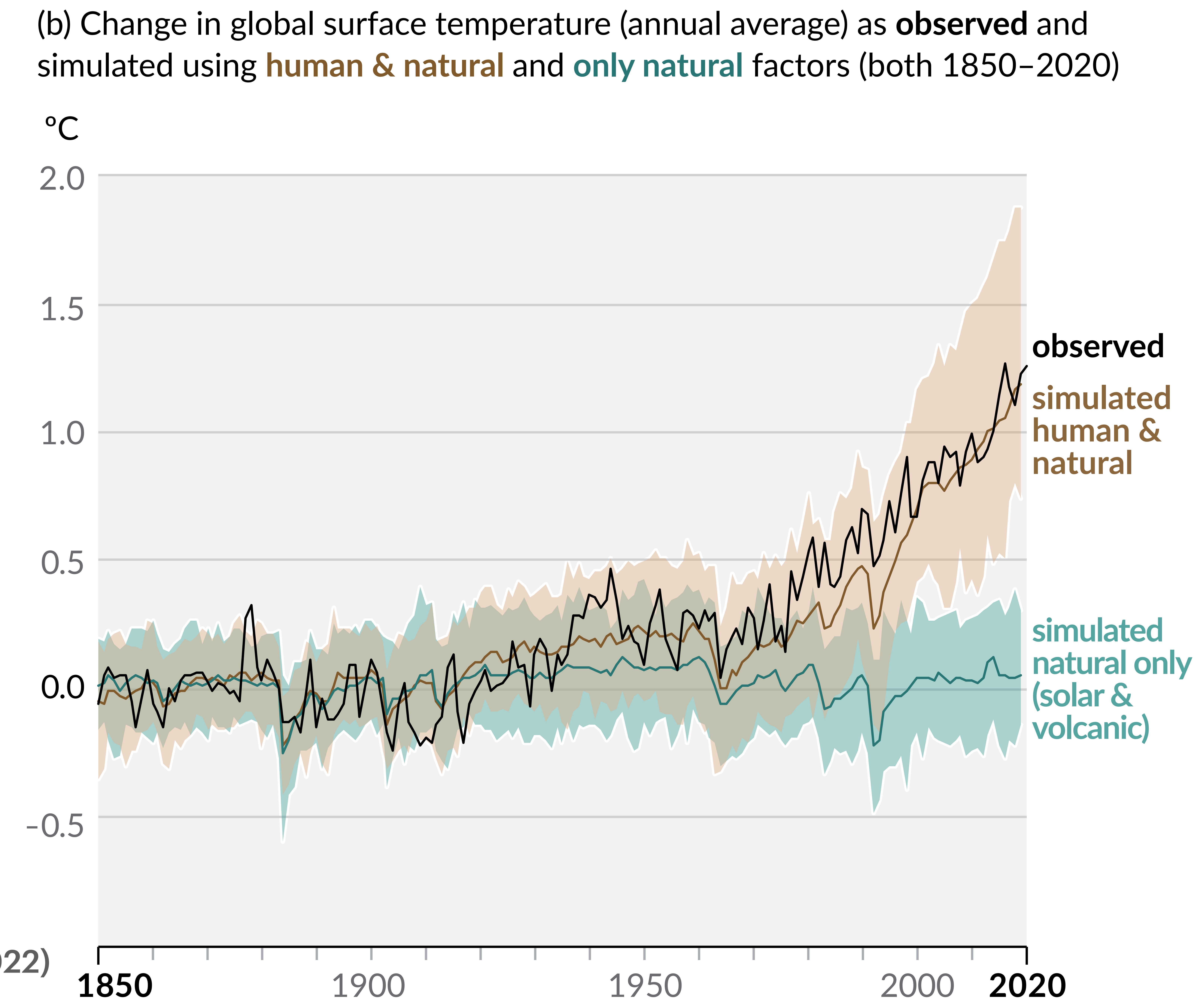
Topos Institute Colloqium, 12th October 2023

1850-2022 (Ed Hawkins “Warming stripes”)



https://keelingcurve.ucsd.edu/

The cause of warming is clear



IPCC 6th Assessment Report (2022)

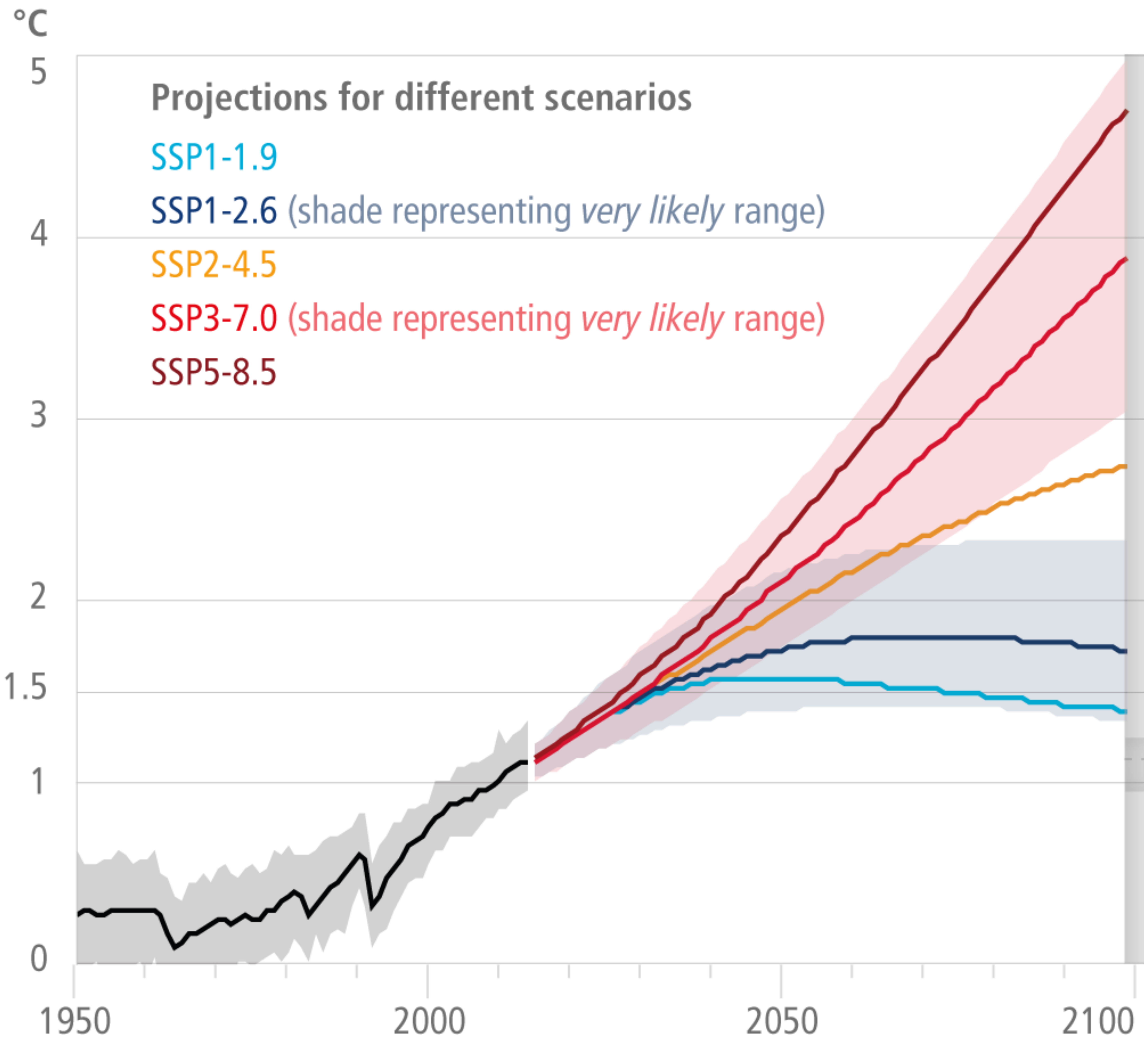
SSP1-1.9 - net zero by 2050

SSP1-2.6 - serious reduction by 2050

SSP2-4.5 - current levels maintained till 2050 then fall to net zero by 2100

SSP3-7.0 - doubling current emissions by 2100

SSP5-8.5 - doubling current emissions by 2050



IPCC 6 (2022)

…and further interlinked crises

Desertification

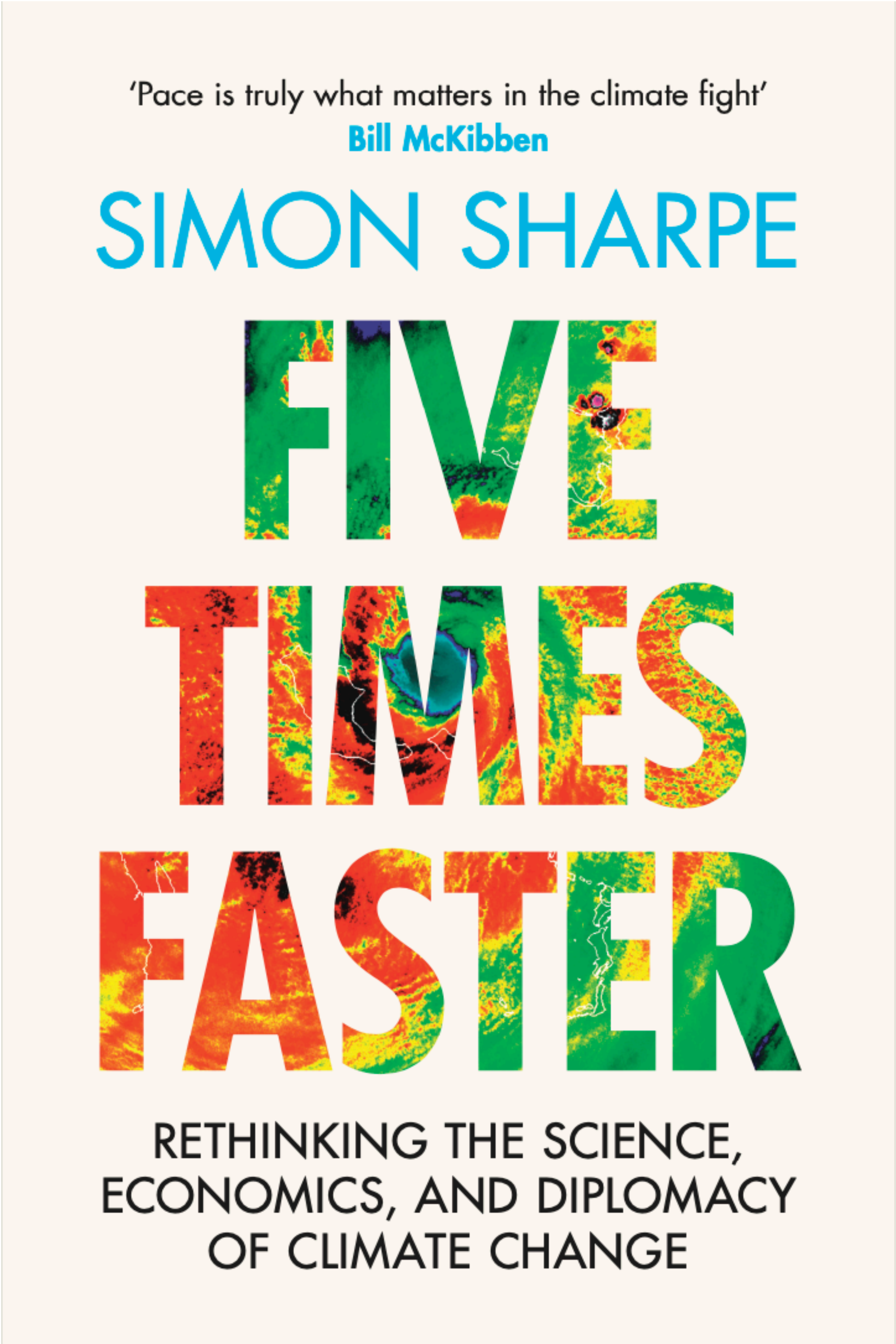
Biodiversity loss

Felton Davis (CC2.0)

https://commons.wikimedia.org/wiki/File:D%C3%A9sertification_des_terres.jpg

"A race we are losing, but a race we can win..."

UN Secretary-General António Guterres



"Still, our appreciation of the risks of climate change is limited by the way our academic institutions encourage each researcher to focus on their own narrow area of expertise."

UNIVERSITY OF CAMBRIDGE

Institute of Computing for Climate Science

Maximise effectiveness of climate science research via...

Computer Science

Software Engineering

Programming Languages & systems

Mathematics

Data Science

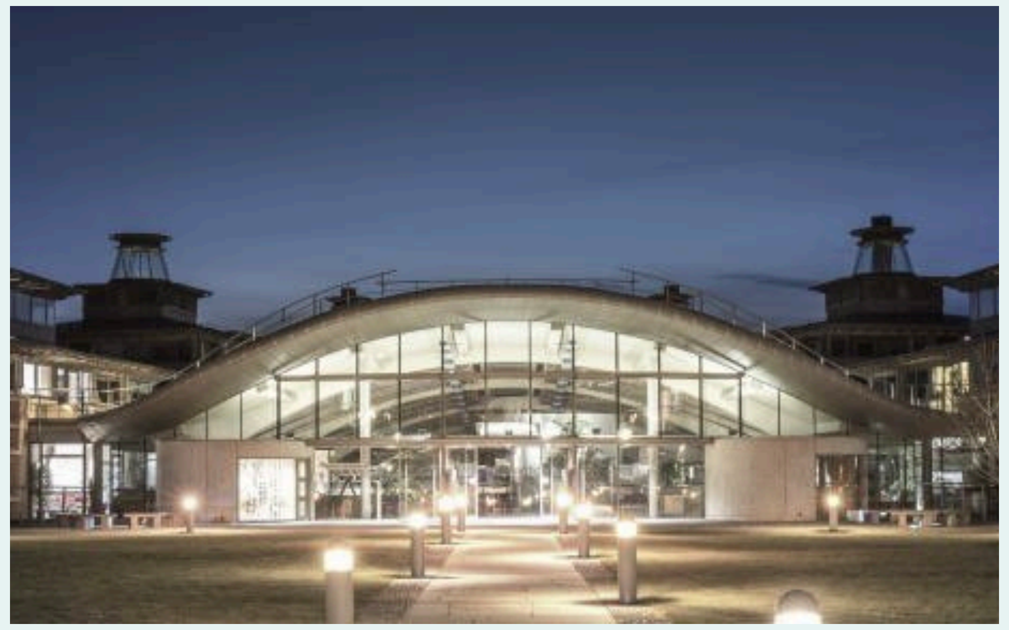
Machine learning

Emily Shuckburgh

Cambridge Zero
+ Computer Science
& Technology

Colm Caulfield

Department of Applied
Maths and Theoretical
Physics



Chris Edsall

Research
Computing
Services

Dominic Orchard

Department of
Computer Science &
Technology

Marla Fuchs

ICCS

Institute of
Computing for
Climate Science

Virtual Earth System Research Institute (VESRI)

DataWave: Collaborative Gravity Wave Research

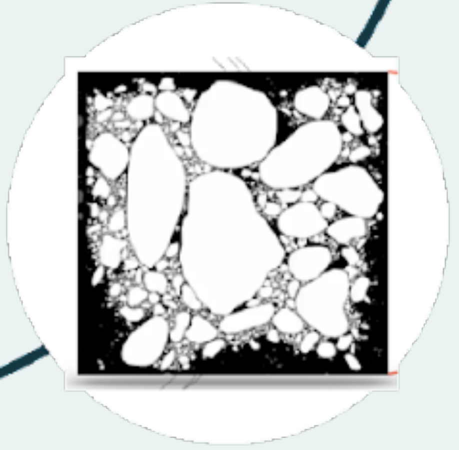
LEMONTREE: Land Ecosystem Models based On New Theory, obseRvations, and ExperimEnts

CALIPSO: Carbon Loss In Plants, Soils and Oceans

Institute of Computing for Climate Science

FETCH$_4$: Fate, Emissions, and Transport of CH_4

M²LInES: Multiscale Machine Learning In Coupled Earth System Modeling



SASIP: The Scale-Aware Sea Ice Project

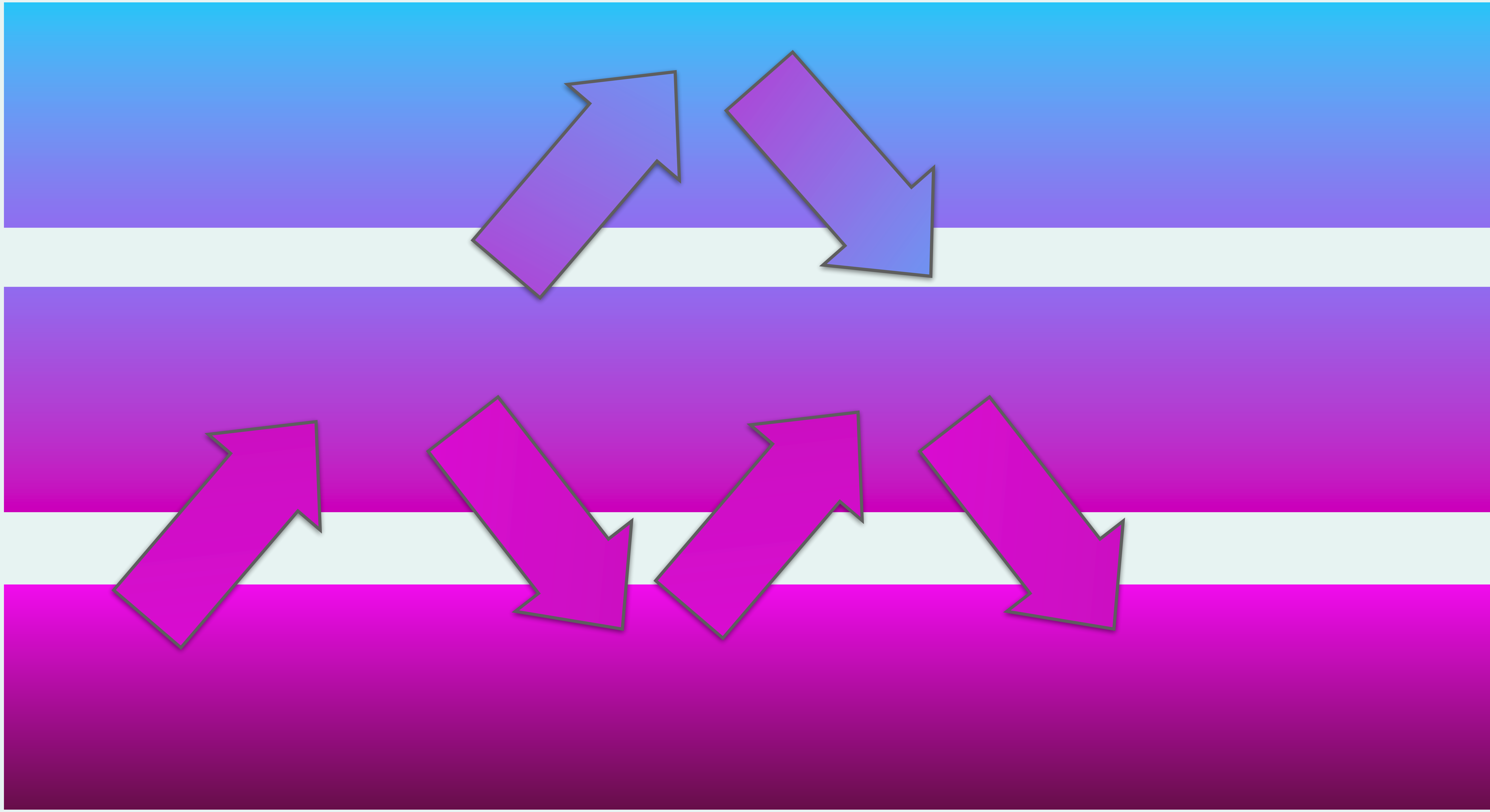
UNIVERSITY OF CAMBRIDGE

Institute of Computing for Climate Science

Open research questions
5–30 years

Cross-cutting concerns
2–5 years

Immediate impact
Reactive
6 months – 2 years



My work:

"Tools for the tool makers" for decision making, understanding, forecasting, monitoring

Programming for the planet

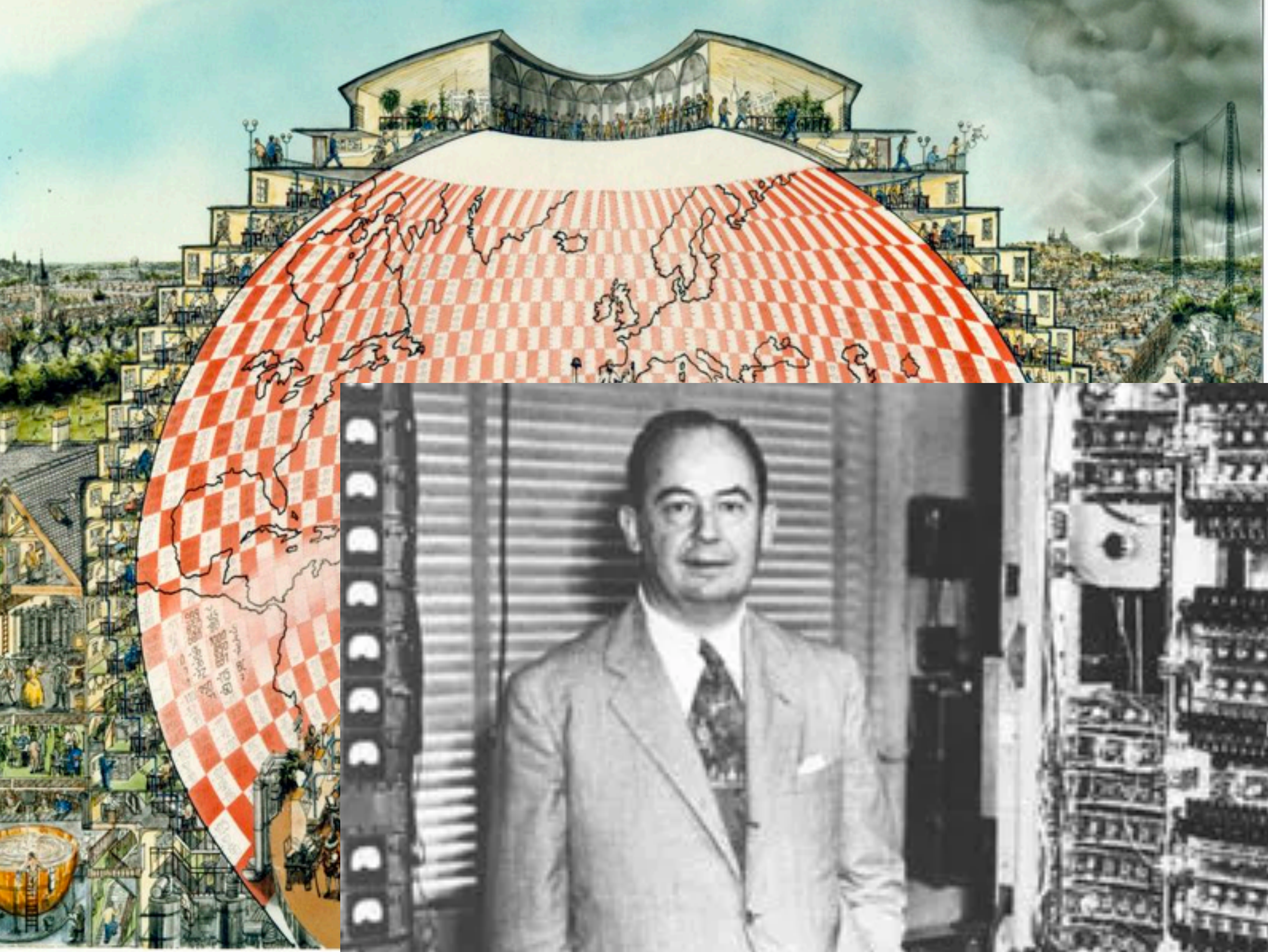
- Climate modelling background
- Challenges of scale
- Role of languages and work in progress
    1. Static analysis and lightweight verification
    2. Categorical abstractions for grids
    3. Transparent and explainable computation
- Ideas for the future

1922

Weather Prediction by Numerical Process

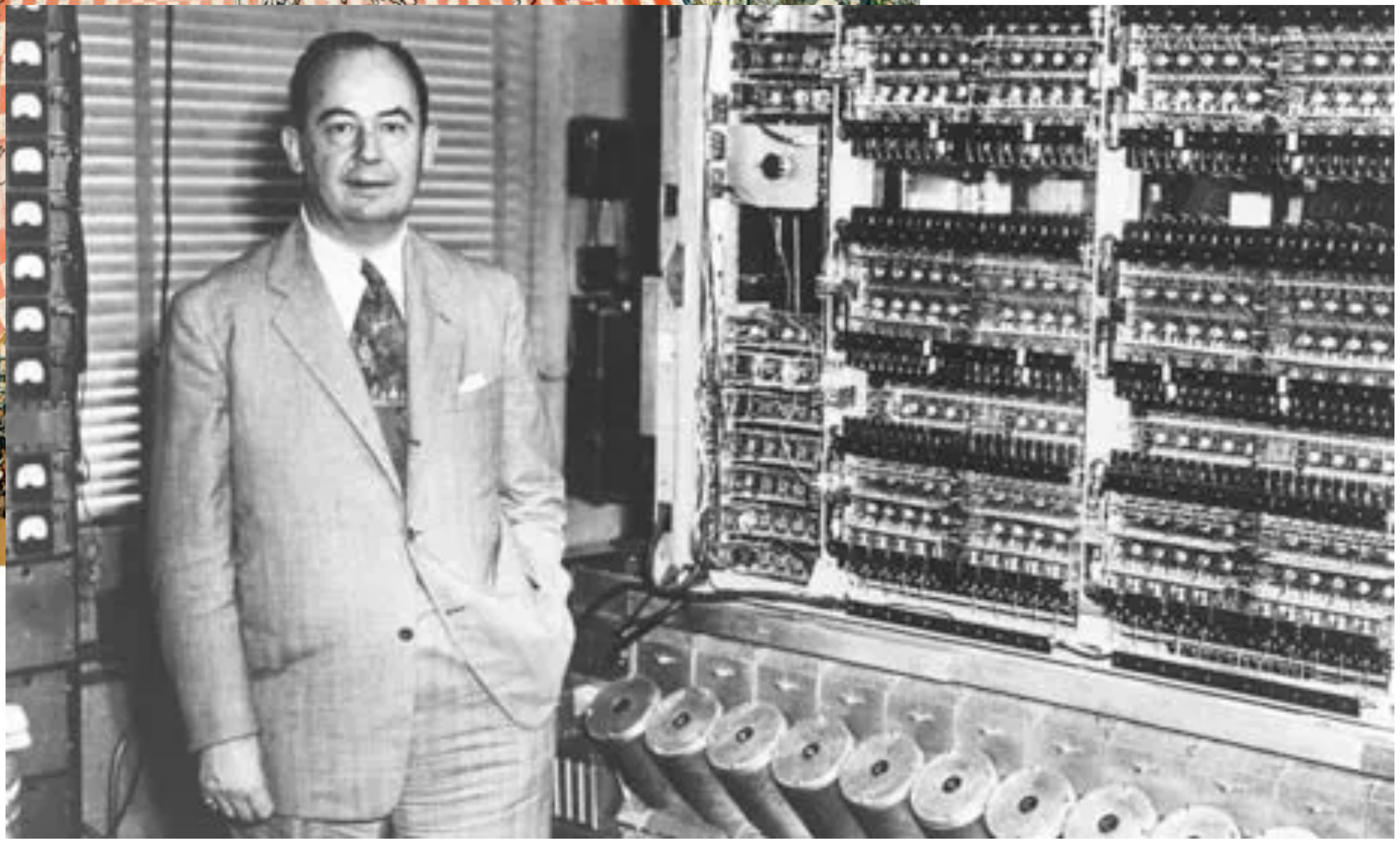
(L.F.Richardson, 1922)

Image: Weather Forecasting Factory Stephen Conlin, 1986.



1922

1945-47



John von Neumann
(with the stored-program computer at the Institute of Advanced Study, Princeton 1945)

Jule Gregory Charney

1922

1945-47

1967-69

"According to our estimate, a doubling of the CO_2 content in the atmosphere has the effect of raising the temperature of the atmosphere by 2C"

(Manabe & Wetherald)

Thermal Equilibrium of the Atmosphere with a Given Distribution of Relative Humidity

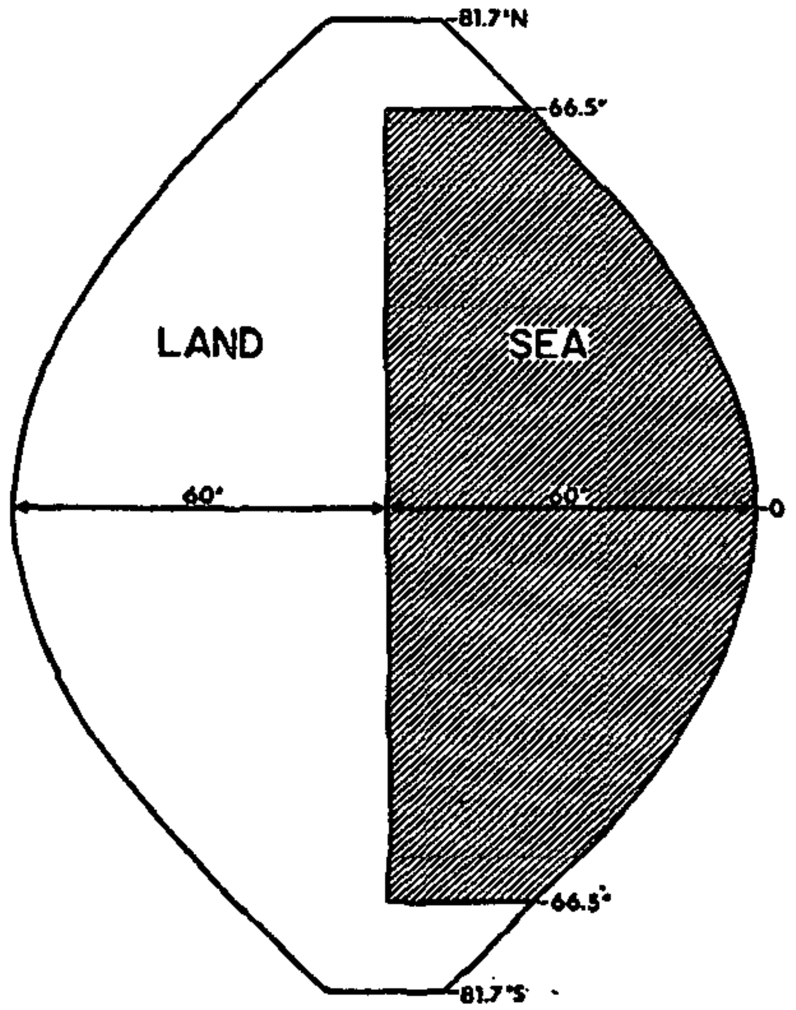


FIG. 1. Ocean-continent configuration of the model.

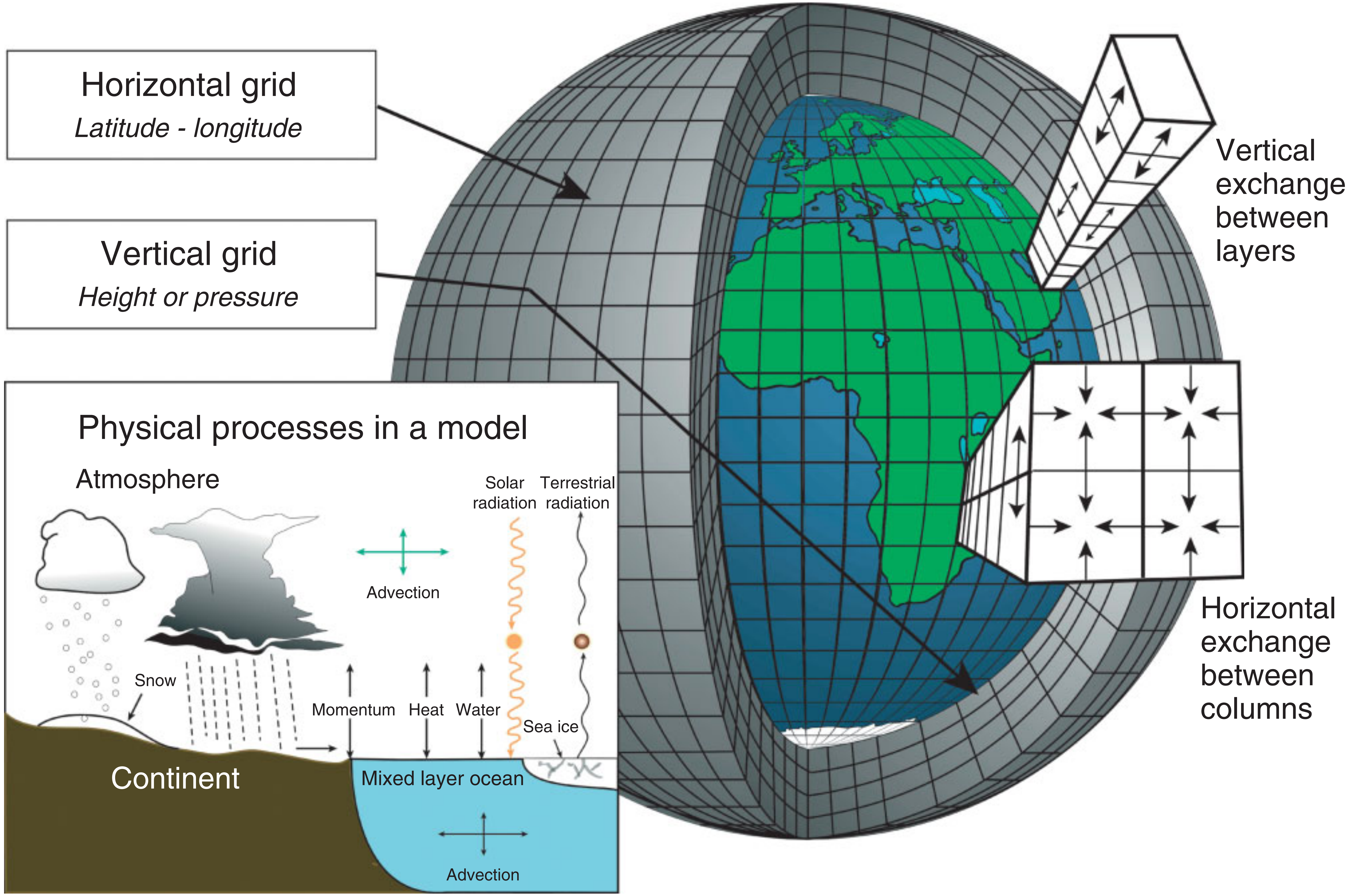
First "coupled" model

"milestone in scientific computing"
(Nature 2006)

Climate calculations with a combined ocean-atmosphere model

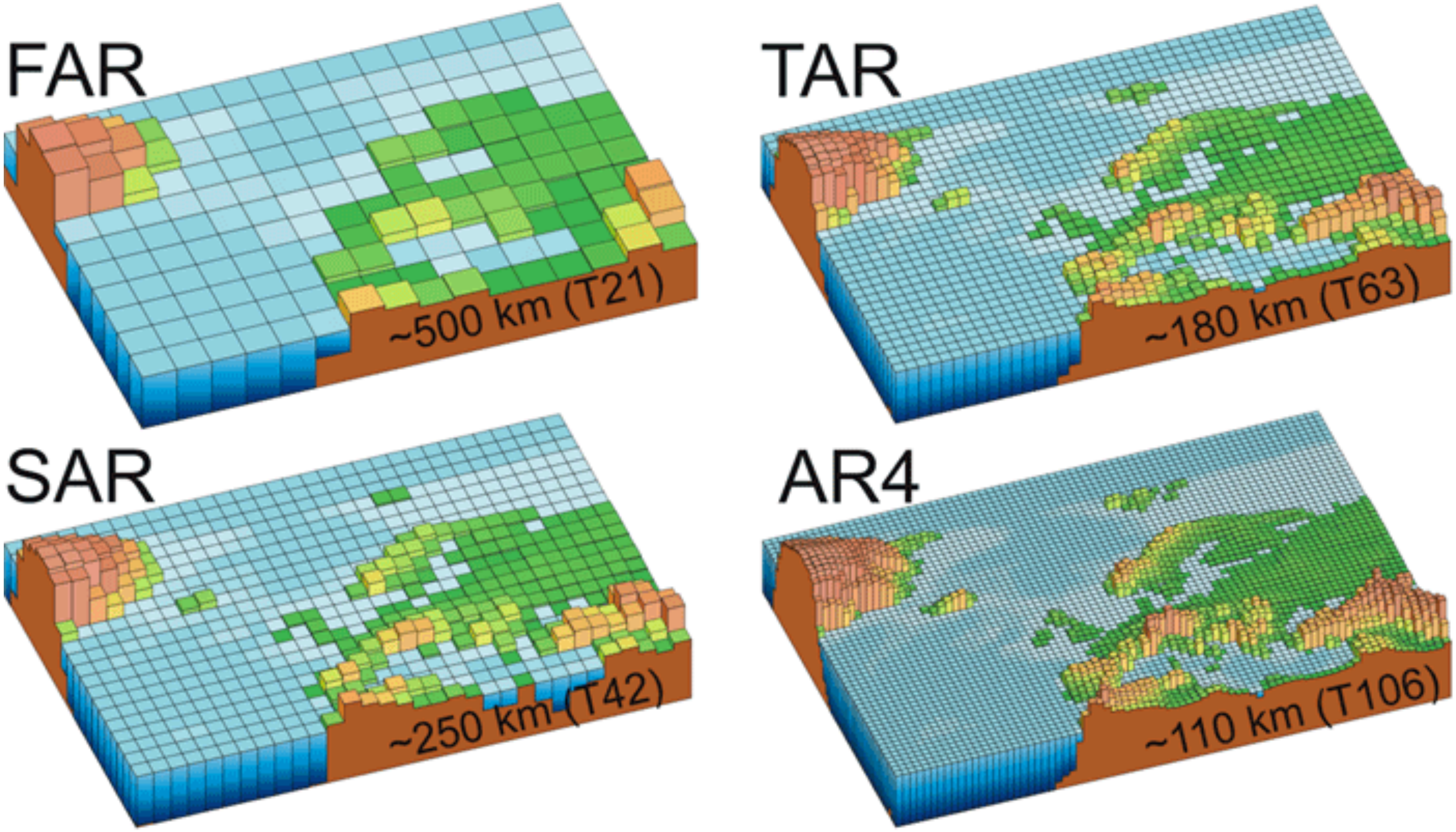
(Manabe & Bryan)

Modern GCMs (Global Circulation Models)



Edwards (2011)

Increasing resolution over IPCC models (1990,1995,2001,2007)



graphics from 4th IPCC report (2007)

AR6 - Model resolutions

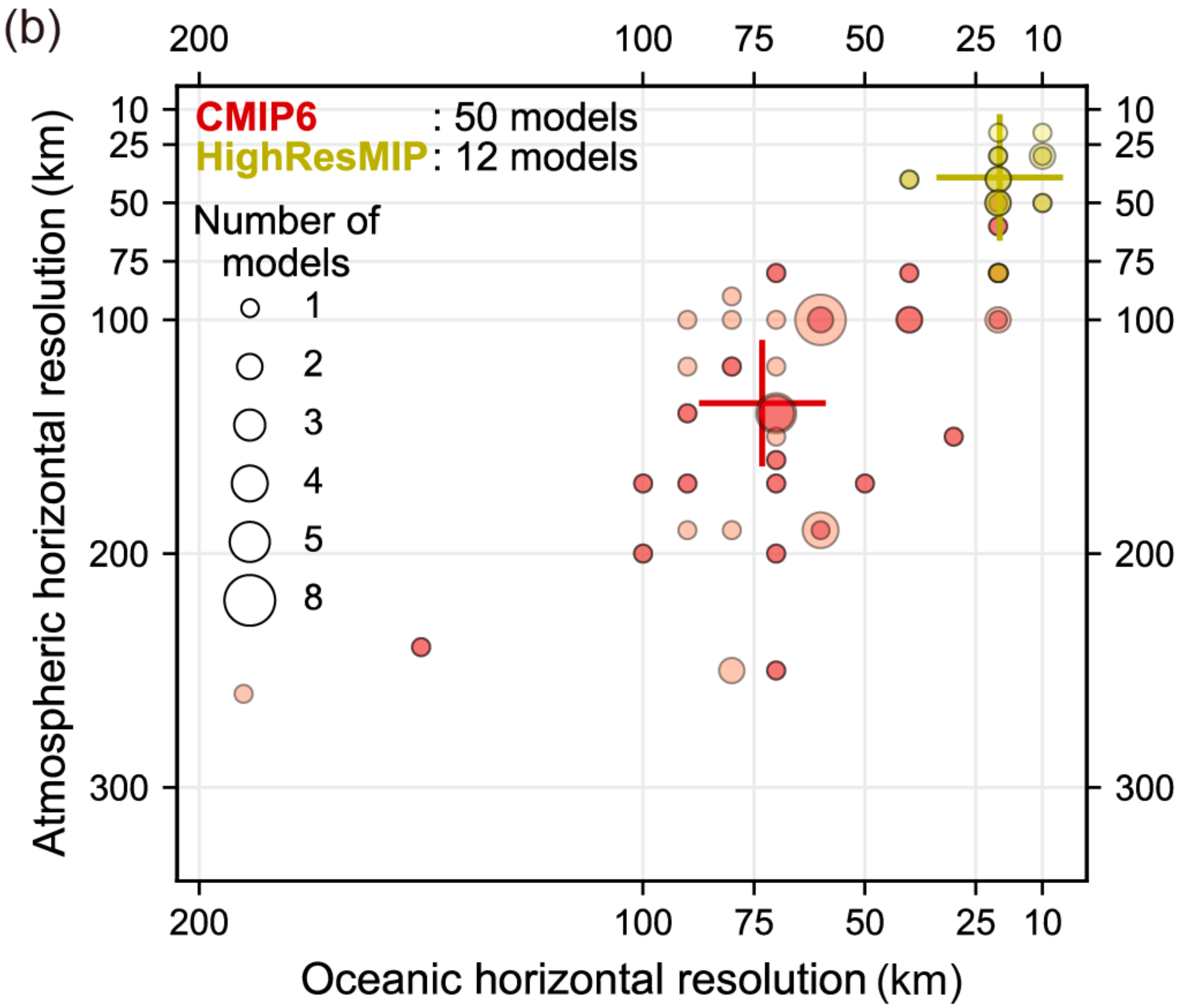
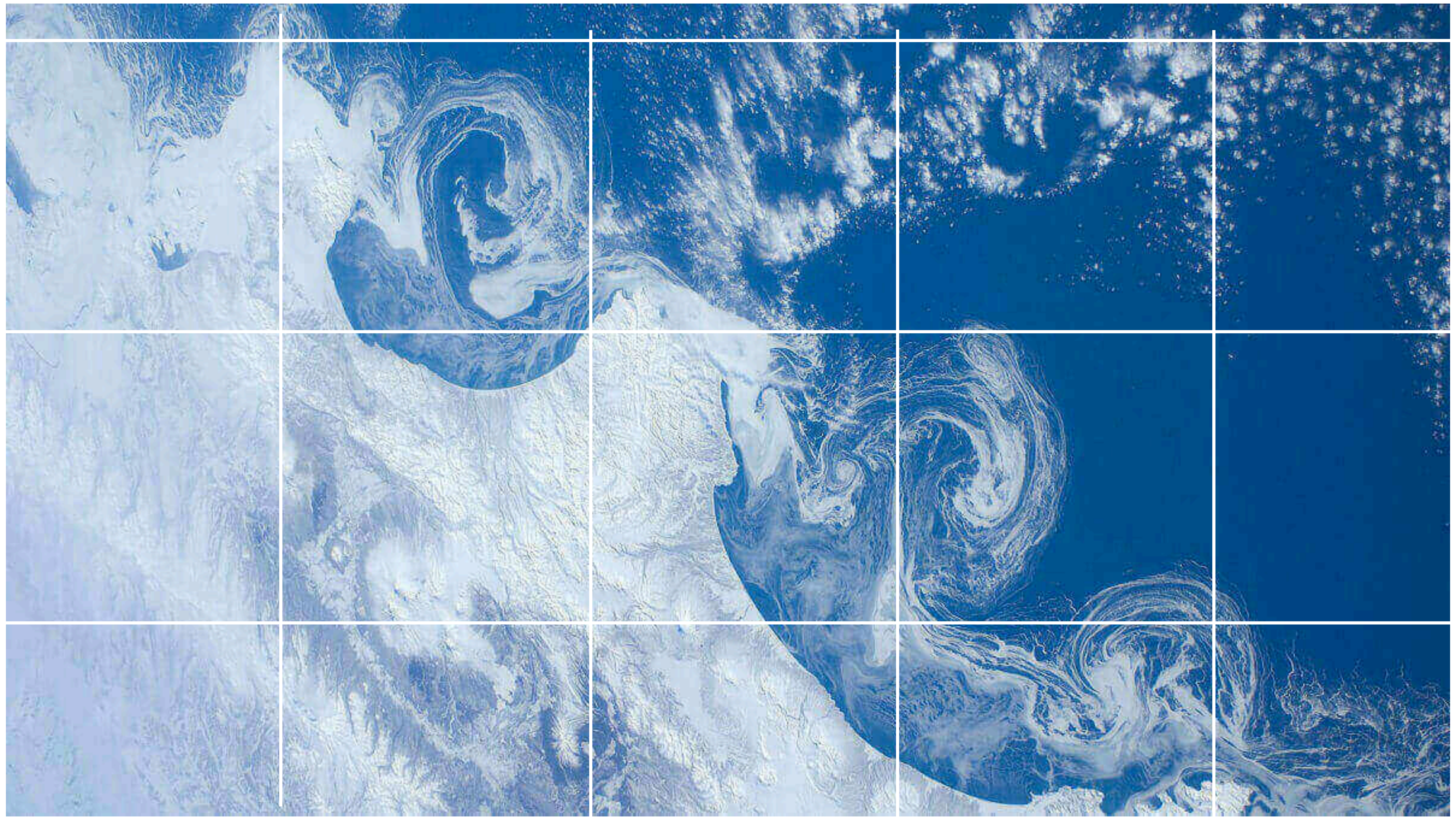
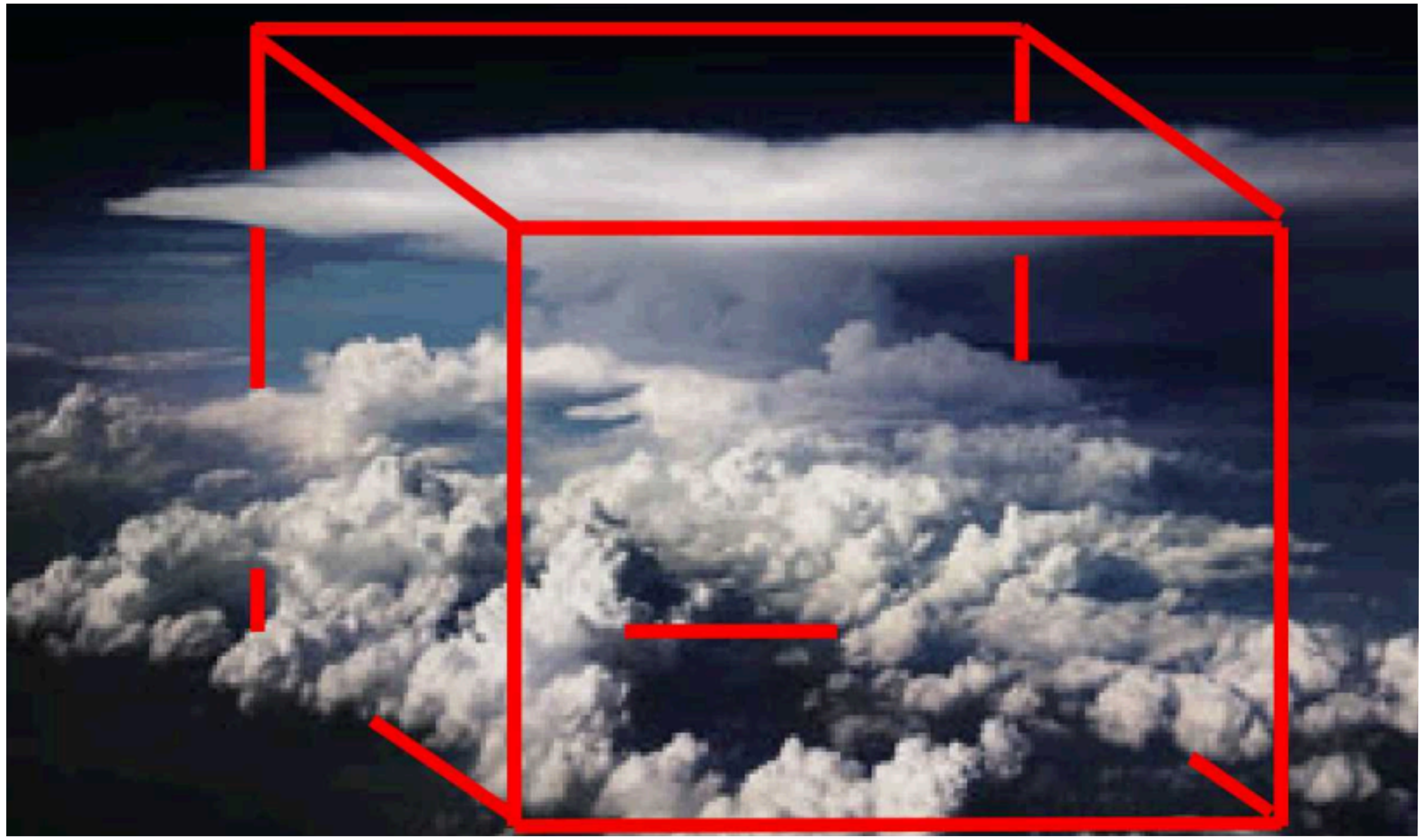


Figure 1.19 in IPCC, 2021: Chapter 1. In: *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change [Chen, et al.]*

Approximating sub grid processes



NASA / Wikimedia Commons



Hillman et al. 2020

Uncertainty / error *vs.* expense

Behavioural specification via conservation law

General form
$$\frac{\delta\phi}{\delta t} = D(\phi) + P(\phi) + U(\phi) + F$$

t - time

ϕ - time-varying ("prognostic") variables in 3D *(state vector)*

e.g. temperature, pressure, wind-speed, humidity, etc.

D - (resolved) dynamics

based on PDEs of fluid **motion**

P - *physics (& chemistry)*

e.g., radiative transfer, convection, (bio-)geochemistry

U - *unresolved processes (subgrid models)*

e.g., eddies, clouds, other waves

F - *forcings (external factors, not simulated)*

*e.g. solar radiation (**insolation**), anthropogenic emissions, geothermal heating*

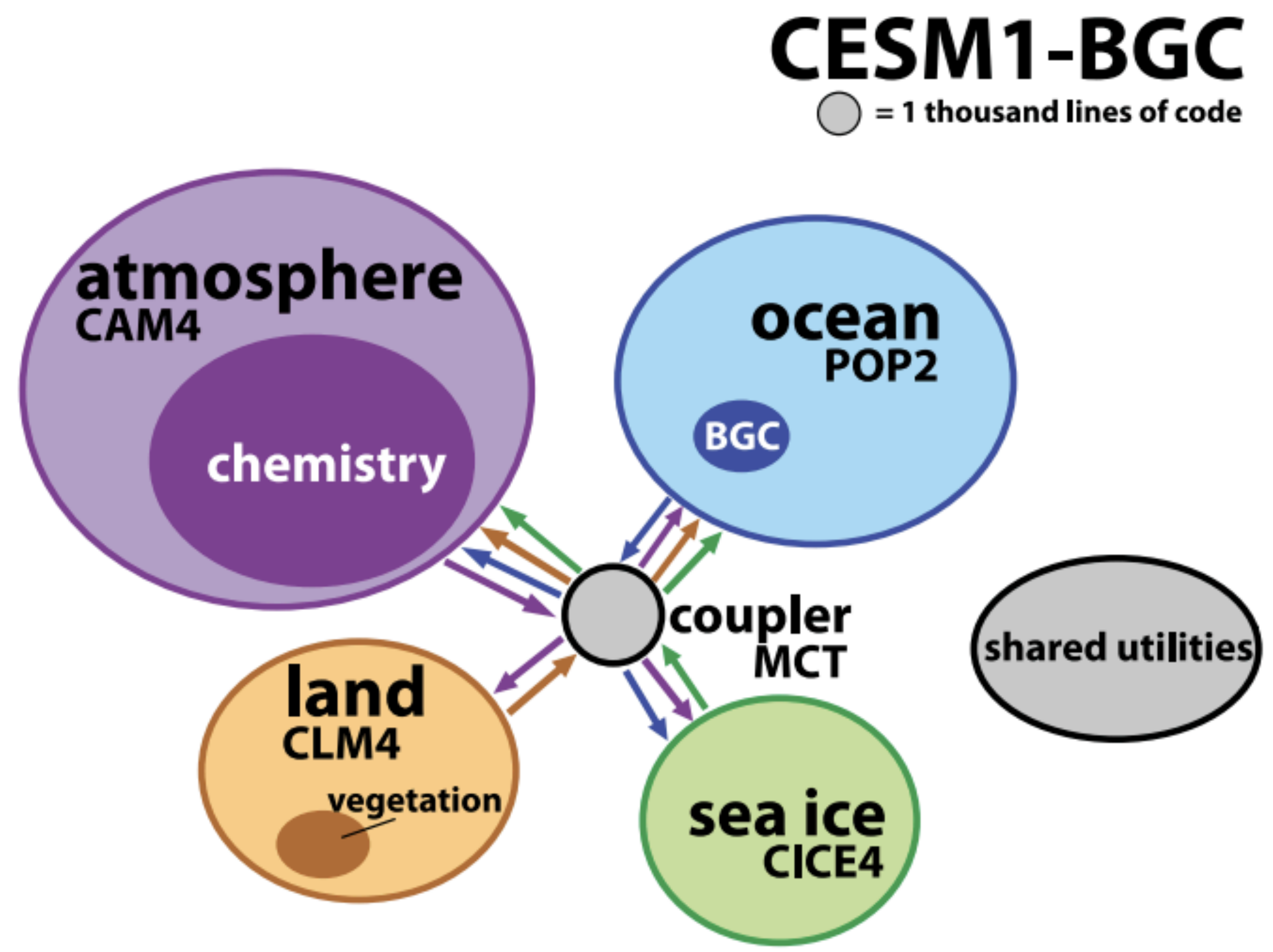


Figure 1. Architecture diagram for CESM1-BGC.

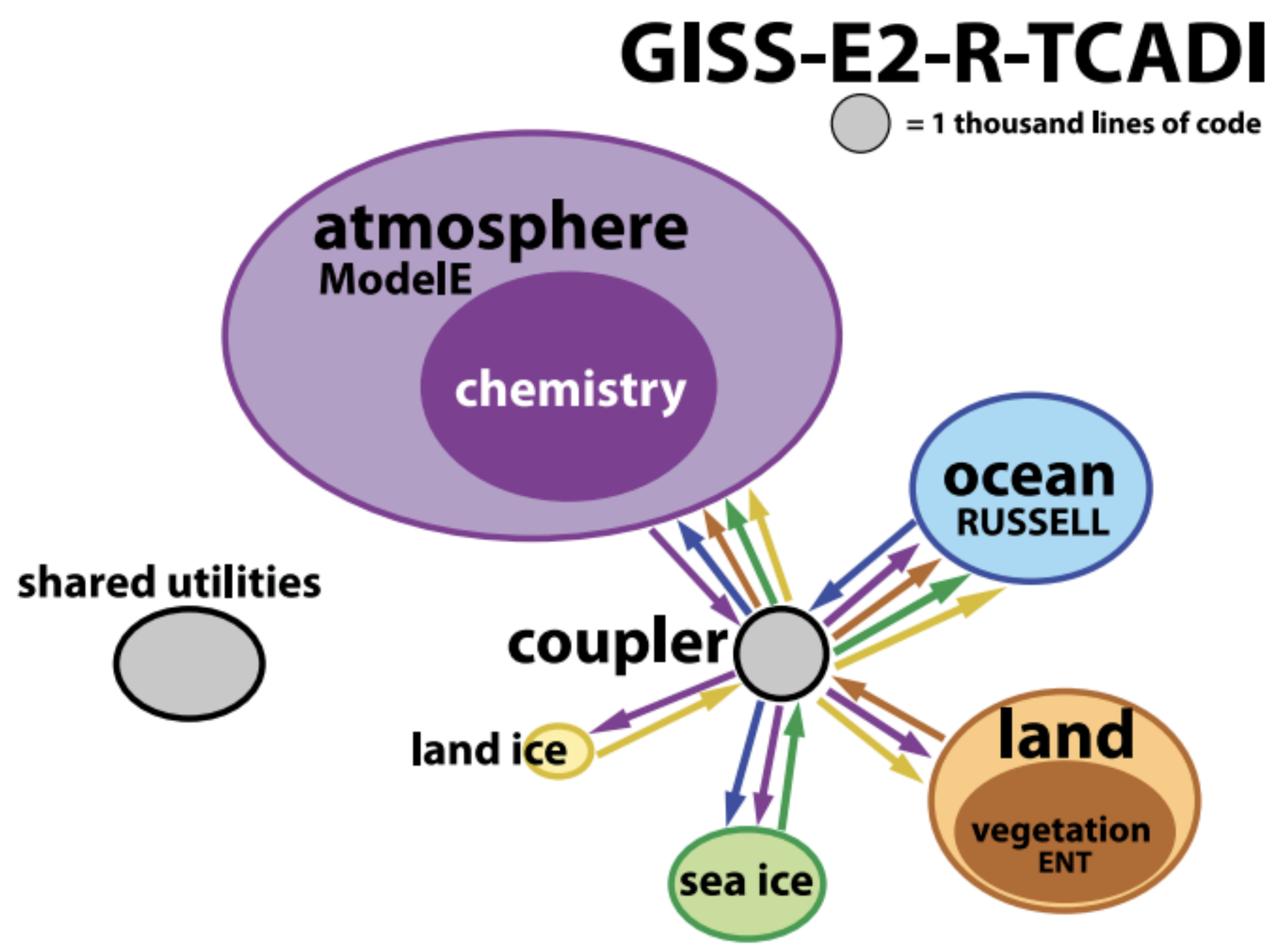


Figure 3. Architecture diagram for GISS-E2-R-TCADI.

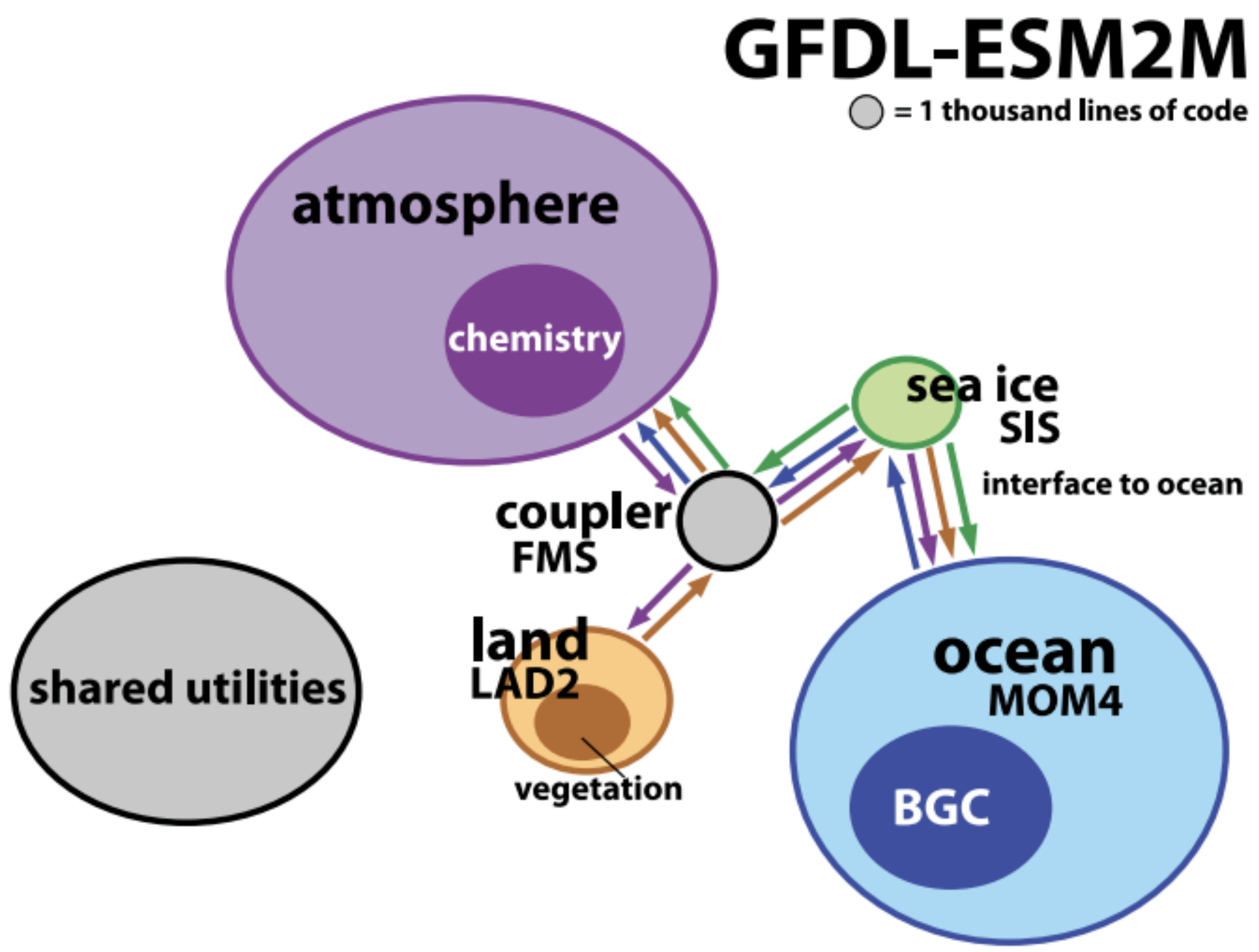


Figure 2. Architecture diagram for GFDL-ESM2M.

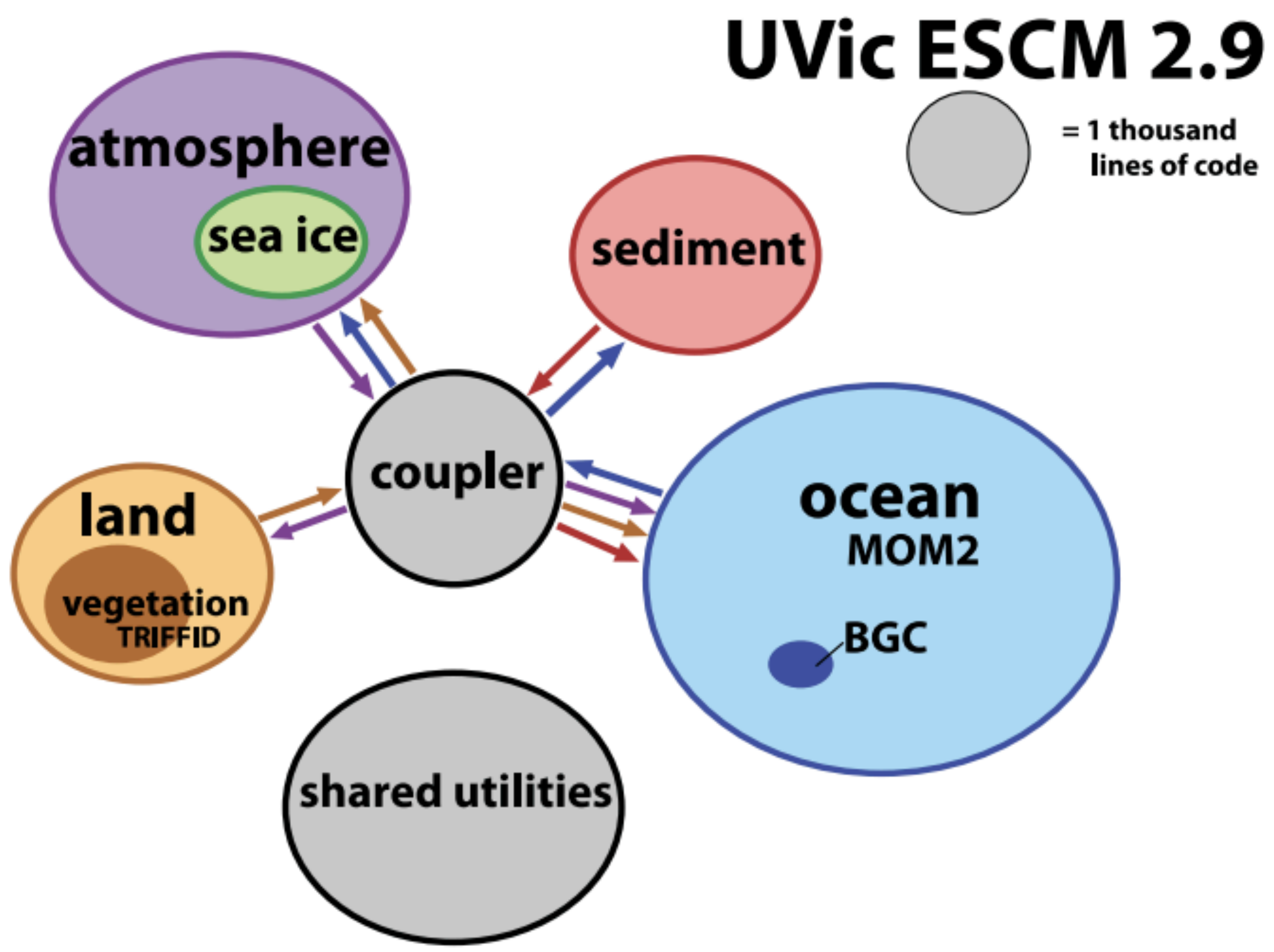


Figure 4. Architecture diagram for UVic ESCM 2.9.

*intermediate complexity model

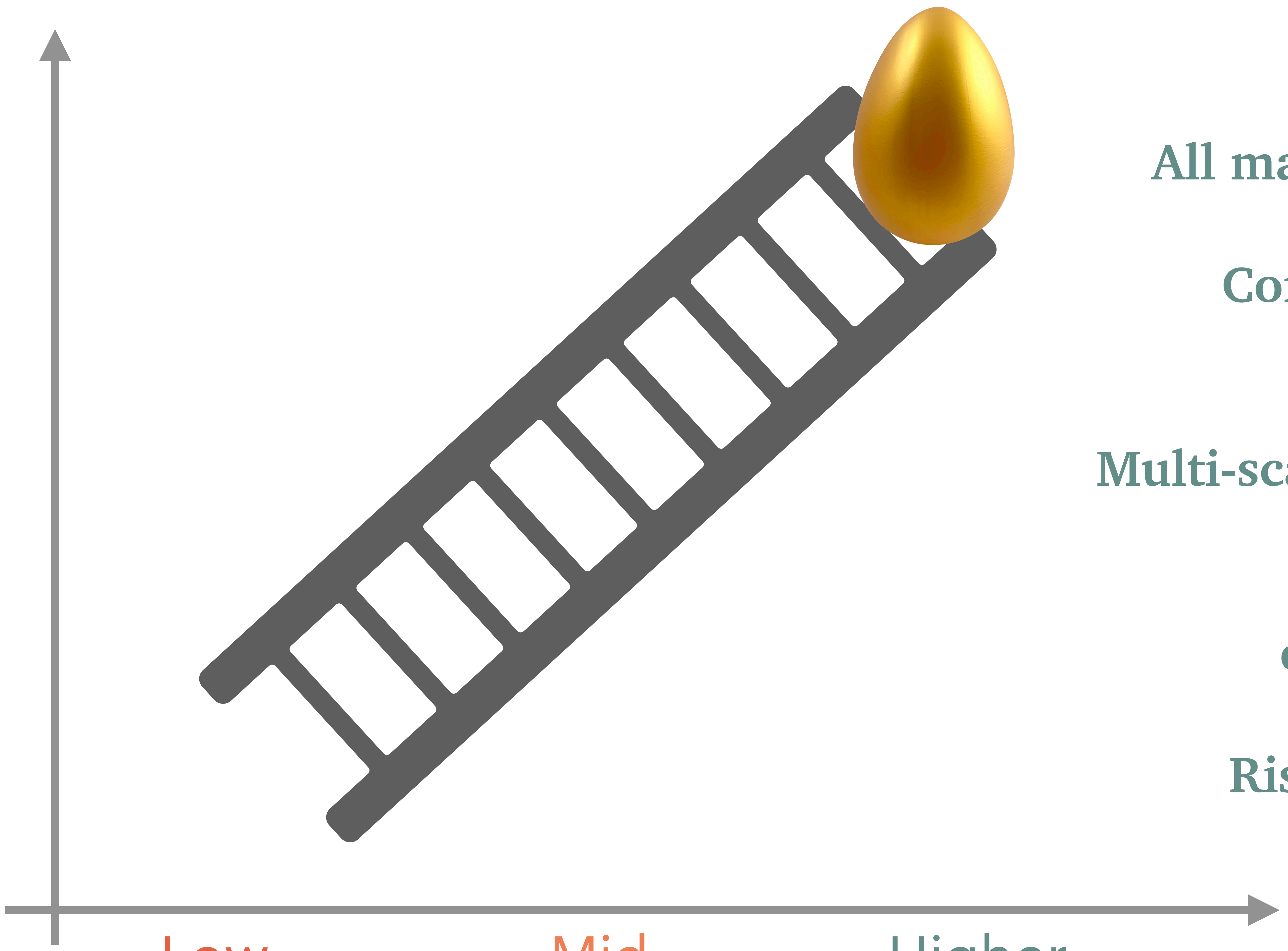
$O(1\text{Msloc})$

The software architecture of climate models: a graphical comparison of CMIP5 and EMICAR5 configurations
Naughten (née Alexander) and Easterbrook (2015)

Better prediction: “climbing the ladder” (Charney)

More processes

Fewer processes



Low resolution 300km

Mid 25-100km (typical GCM)

Higher resolution 1-5km

1k resolution

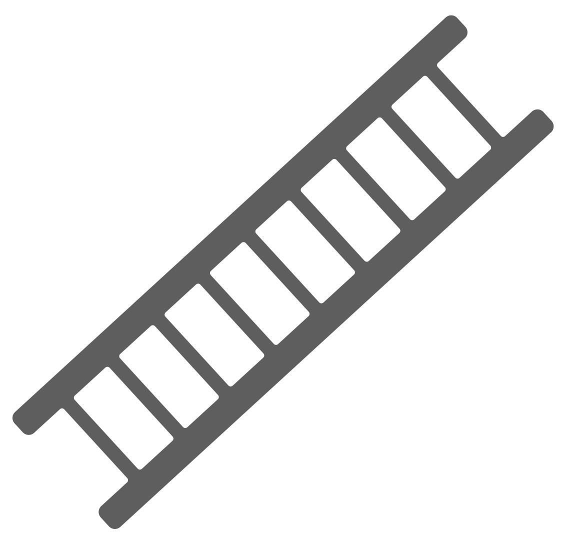
All major processes

Continuous data assimilation

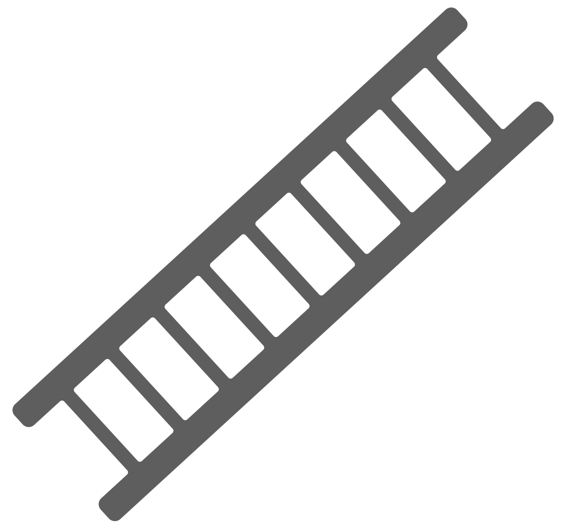
Multi-scale prediction

Uncertainty quantification

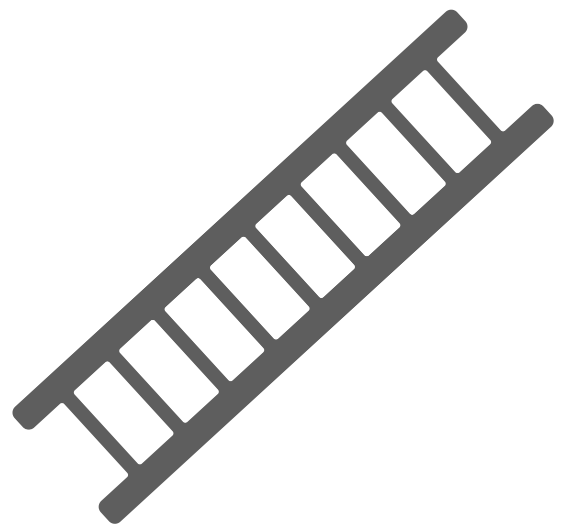
Risk assessment



Computation



Collaboration



Communication

Scaling computation

The challenge of increasing resolution

Doubling horizontal resolution means…

$$\Delta x' = \frac{\Delta x}{2} \implies 4 \times \text{grid points (in horizontal, since } \Delta x = \Delta y)$$

$$\implies 2 \times \text{time steps } \textbf{at least } \left(\Delta t' \leq \frac{\Delta x'}{\text{maxprop}_x} = \frac{\Delta t}{2}\right)$$

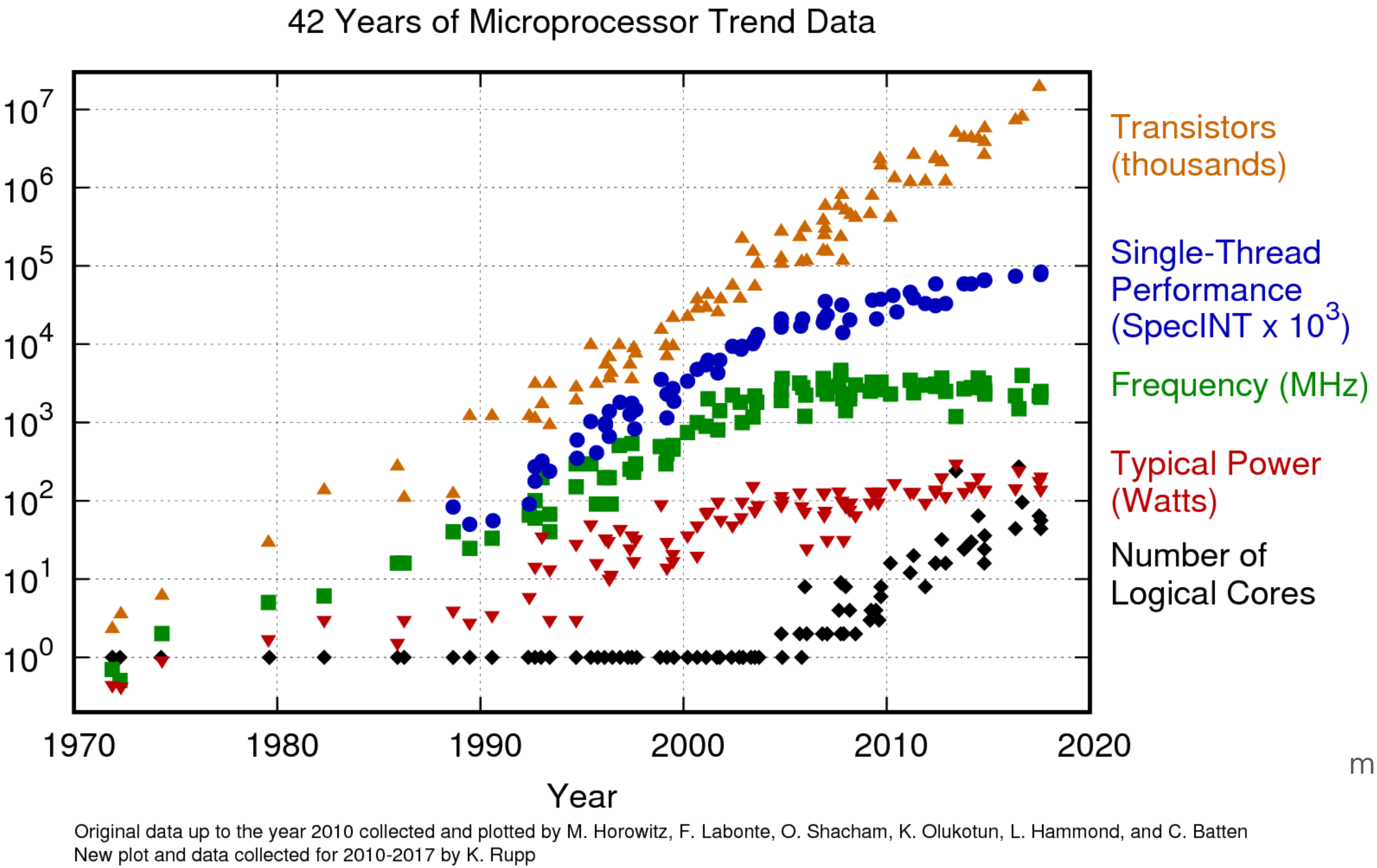
(see Courant-Friedrichs-Lewy condition)

$\implies$ 8 $\times$ more computation!

That's even before we consider the vertical…

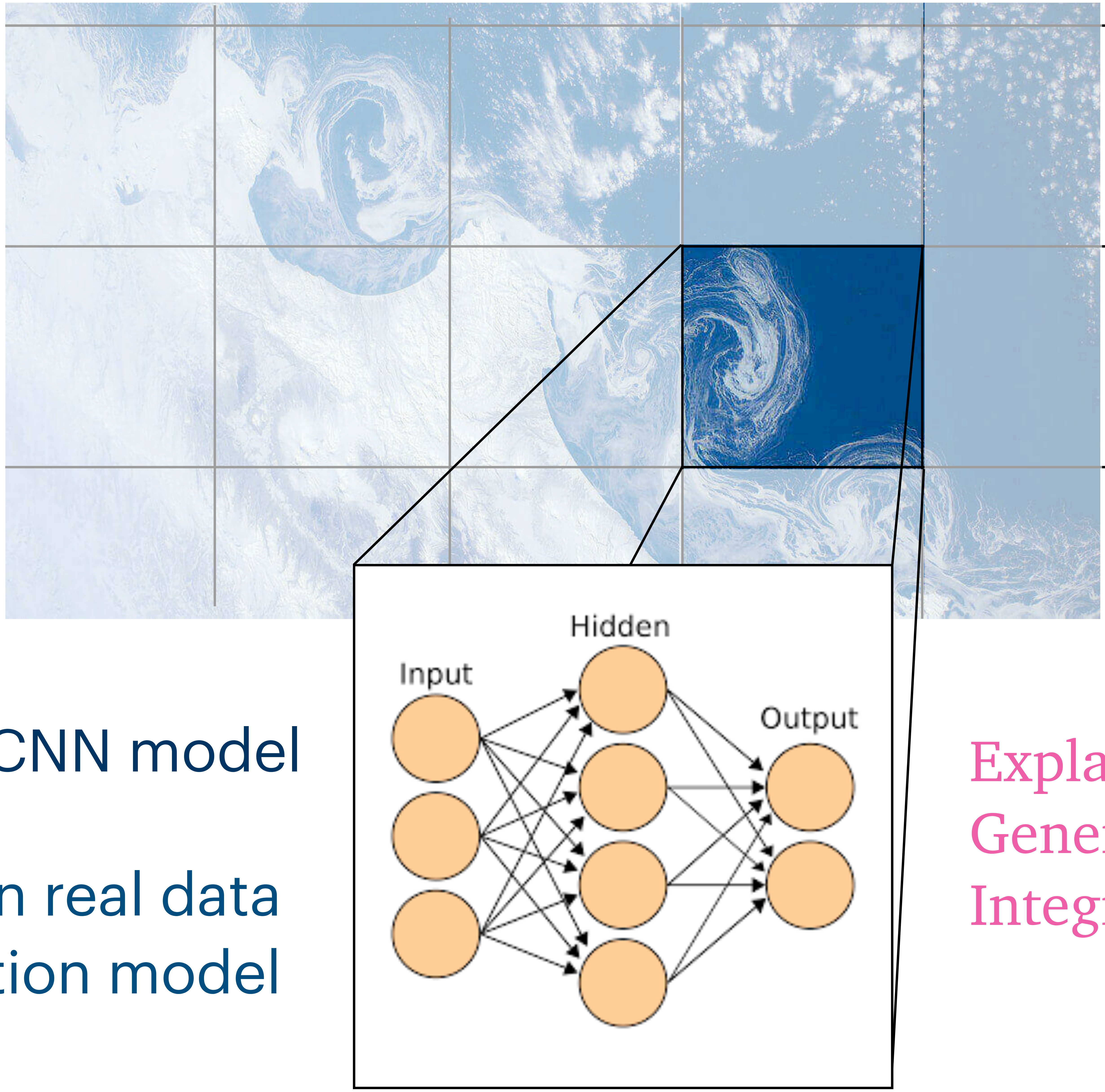
Further challenges of scenario testing, uncertainty estimation, short-term forecasting

Scaling computation



Are general circulation models obsolete? (Balaji et al. 2022)

Recent approach: Data-driven (machine learning) subgrid models



ANN or CNN model

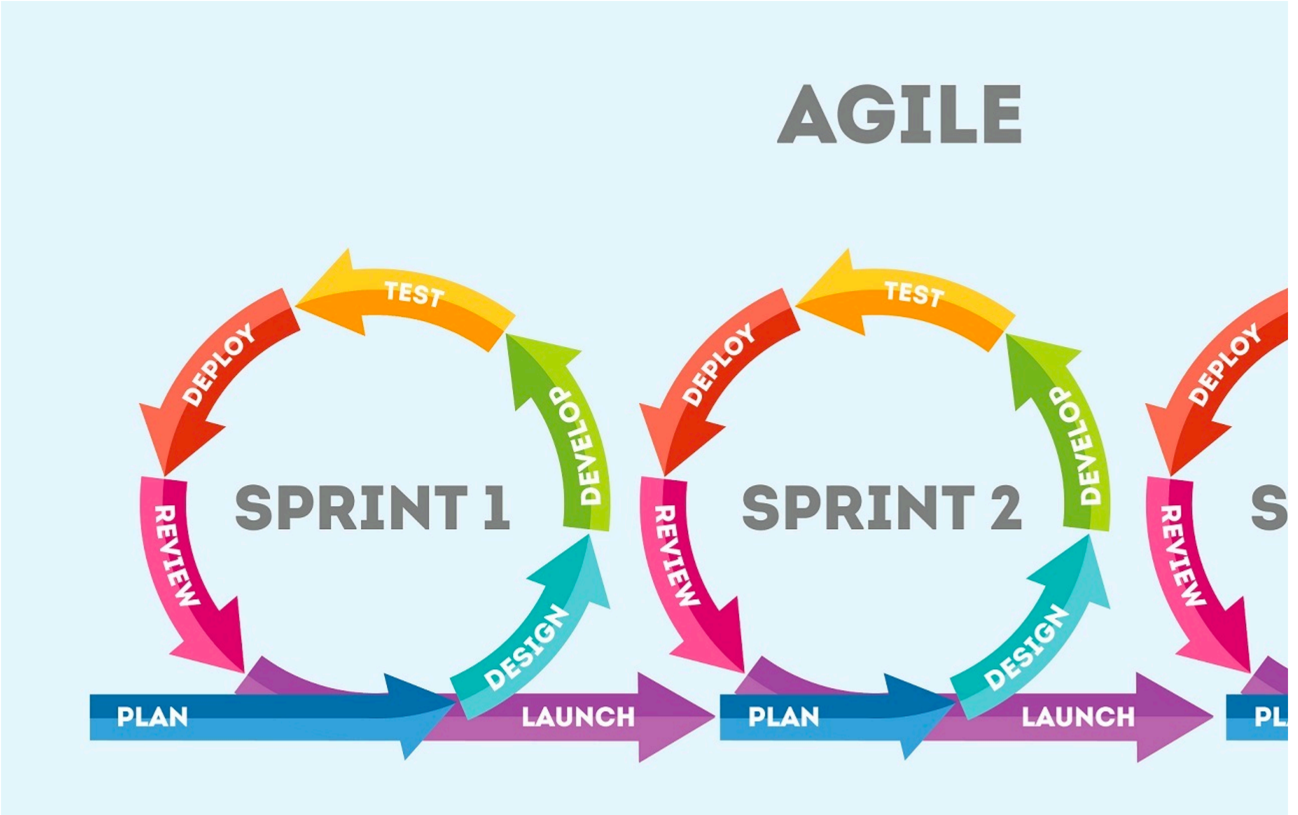
Train on real data
or high-resolution model

Explainability?
Generalisability?
Integration into GCM?

Scaling collaboration

Deploy and train in software engineering tools & techniques

Processes



Version control & public curators

Build systems & containers

Debugging

Profiling

Testing and verification

Structural and cultural/sociological change happening

Software Sustainability Institute

RSE

Society of Research Software Engineers

software carpentry

BETTER SOFTWARE BETTER RESEARCH

The Turing Way

<ReSA>

Research Software Alliance

Scaling communication

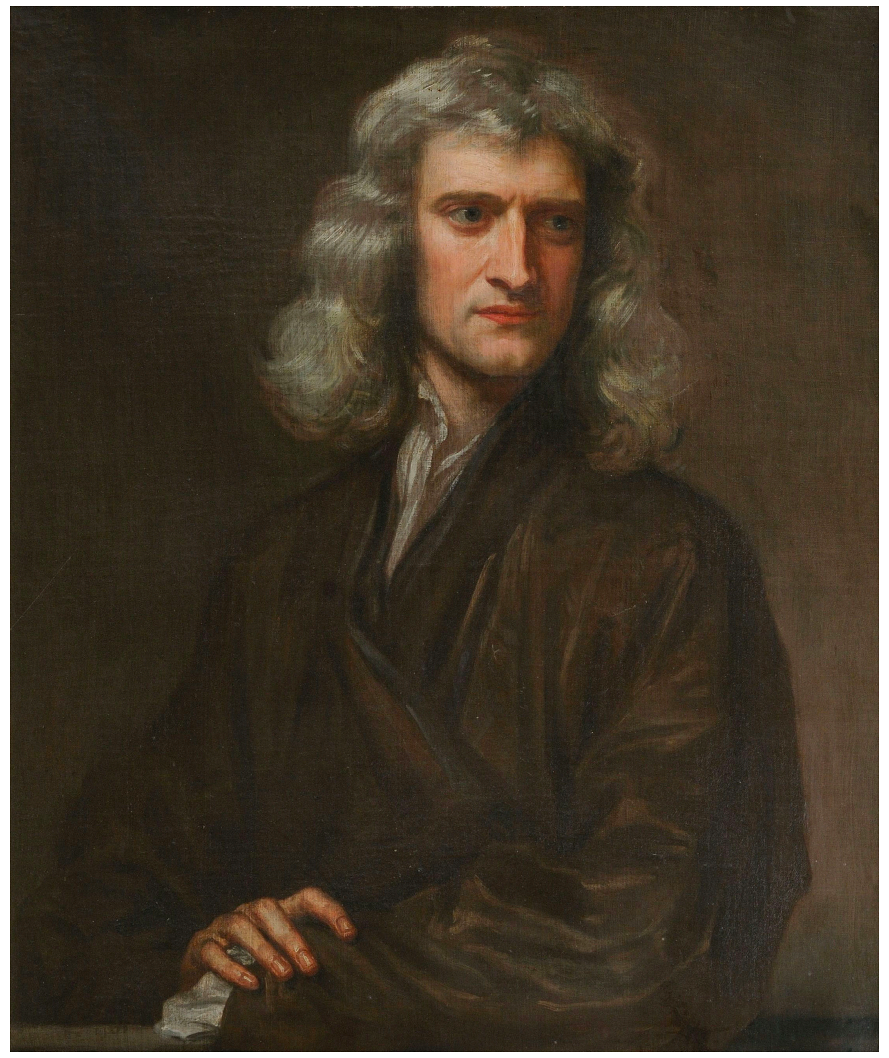
Models in the past...

= maths! (equations in $\mathbb{R}$)

$$F = G\frac{m_1 m_2}{r^2}$$

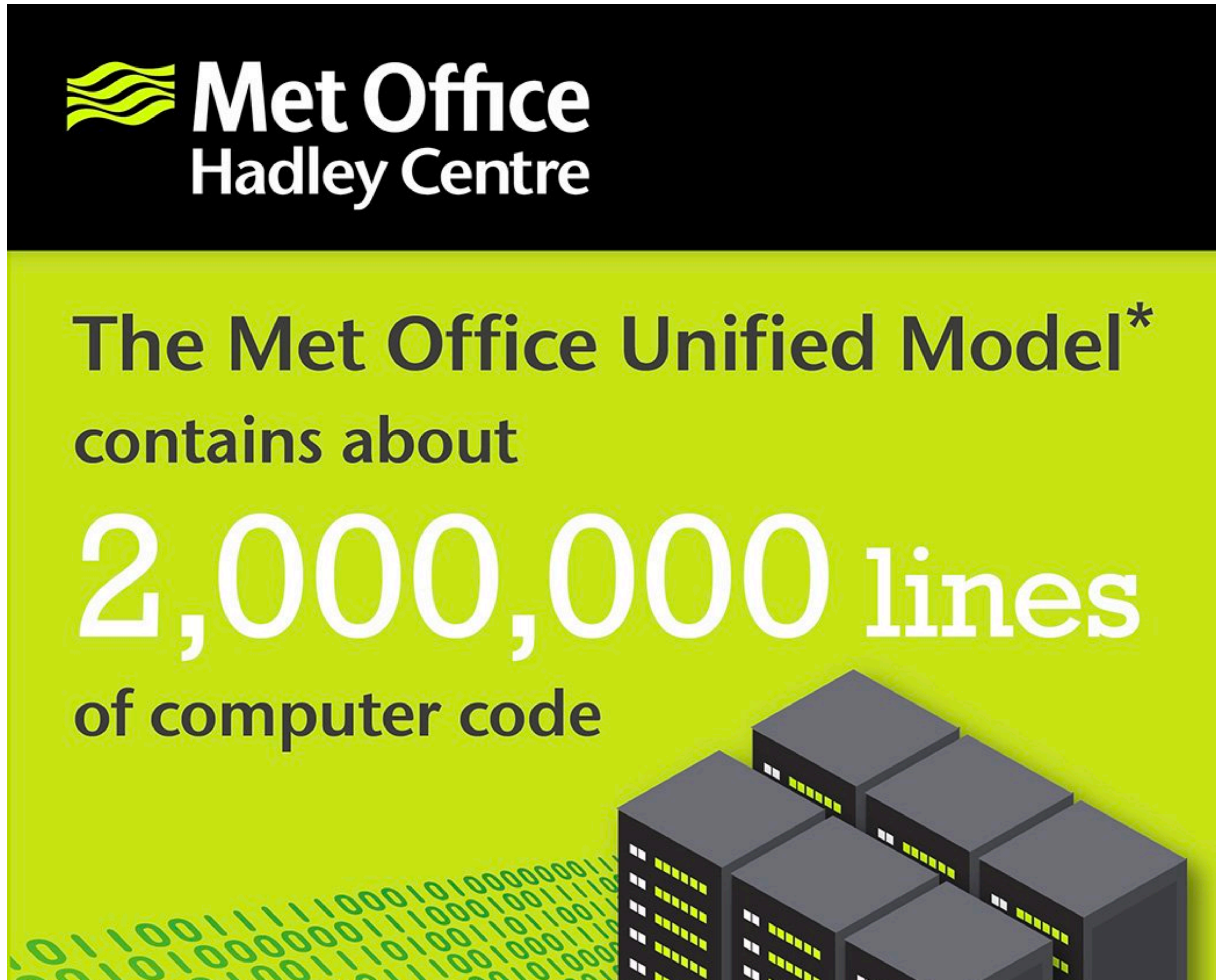
Models now...

= code (and lots of it)



Isaac Newton

Robert Hooke



Scaling communication- Handling the Two Complexities

Inherent

Inadequately supported

Accidental

Too easy to introduce

Both hinder scientific progress, only one is necessary

Scaling communication - Loss of abstract meaning

Example 1D heat equation

Abstract model

$$\frac{\partial \phi}{\partial t} = \alpha \frac{\partial^2 \phi}{\partial x^2}$$

Solution strategy

$$\phi_x^t = \phi_x^{t-1} + \frac{\alpha \Delta t}{\Delta x^2}\left(\boxed{\phi_{x+1}^{t-1}} + \boxed{2\phi_x^{t-1}} + \boxed{\phi_{x-1}^{t-1}}\right)$$

Prediction calculation

```
tend = ...          % end time
xmax = ...          % length of material
dt   = ...          % time resolution
dx   = ...          % space resolution
alpha = ...         % diffusion coefficient
nt = tend/dt        % # of time steps
nx = xmax/dx        % # of space steps
r = alpha*dt/dx^2   % constant in solution

real h(0,nx),        % heat fun. (discretised
     h_old(0, nx);   % in space) at t and t-1

do t = 0, nt
  h_old = h
  do x = 1, nx - 1
       h(i) = h_old(i) + r*(h_old(i-1))
                 - 2*h_old(i) + h_old(i+1)
  end do
end do
```

Gap in explanation….

```fortran
module simulation_mod
  use helpers_mod
  implicit none

  contains

  subroutine compute_tentative_velocity(u, v, f, g, flag, del_t)
    real u(0:imax+1, 0:jmax+1), v(0:imax+1, 0:jmax+1), f(0:imax+1, 0:jmax+1),  &
         g(0:imax+1, 0:jmax+1)
    integer flag(0:imax+1, 0:jmax+1)
    real, intent(in) :: del_t

    integer i, j
    real du2dx, duvdy, duvdx, dv2dy, laplu, laplv

    do i = 1, (imax-1)
      do j = 1, jmax
        ! only if both adjacent cells are fluid cells */
        if (toLogical(iand(flag(i,j), C_F)) .and.                              &
            toLogical(iand(flag(i+1,j), C_F))) then

          du2dx = ((u(i,j)+u(i+1,j))*(u(i,j)+u(i+1,j))+                        &
                  gamma*abs(u(i,j)+u(i+1,j))*(u(i,j)-u(i+1,j))-                &
                  (u(i-1,j)+u(i,j))*(u(i-1,j)+u(i,j))-                         &
                  gamma*abs(u(i-1,j)+u(i,j))*(u(i-1,j)-u(i,j)))                &
                  /(4.0*delx)
          duvdy = ((v(i,j)+v(i+1,j))*(u(i,j)+u(i,j+1))+                        &
                  gamma*abs(v(i,j)+v(i+1,j))*(u(i,j)-u(i,j+1))-                &
                  (v(i,j-1)+v(i+1,j-1))*(u(i,j-1)+u(i,j))-                     &
                  gamma*abs(v(i,j-1)+v(i+1,j-1))*(u(i,j-1)-u(i,j)))            &
                       /(4.0*dely)
          laplu = (u(i+1,j)-2.0*u(i,j)+u(i-1,j))/delx/delx+                    &
                  (u(i,j+1)-2.0*u(i,j)+u(i,j-1))/dely/dely

          f(i,j) = u(i,j) + del_t*(laplu/Re-du2dx-duvdy)
        else
          f(i,j) = u(i,j)
        end if
      end do
    end do

```

Abstract model   Solution strategy   Prediction calculation

Open problem: separating concerns and relating abstractions

Abstract model Solution strategy Prediction calculation

Partial solutions

- Extra technical documentation
- Clear systems design
- High modularity

Could there be better support via a **programming language tailored to science?**

“I don't know what the language of the year 2000 will look like, but I know it will be called Fortran." — Sir Tony Hoare (1982)

Current dominant languages

Are we done?

- Fortran’s evolution shows power of expressivity gains
- But making new languages hard
- Recent breakout success: Julia
- Big bet/opportunity for future climate modelling?

Role of languages and tools and work in progress

Testing difficult, and verification tools not readily deployable

1. Static analysis and lightweight verification

Validation

Did we implement the right equations?

vs

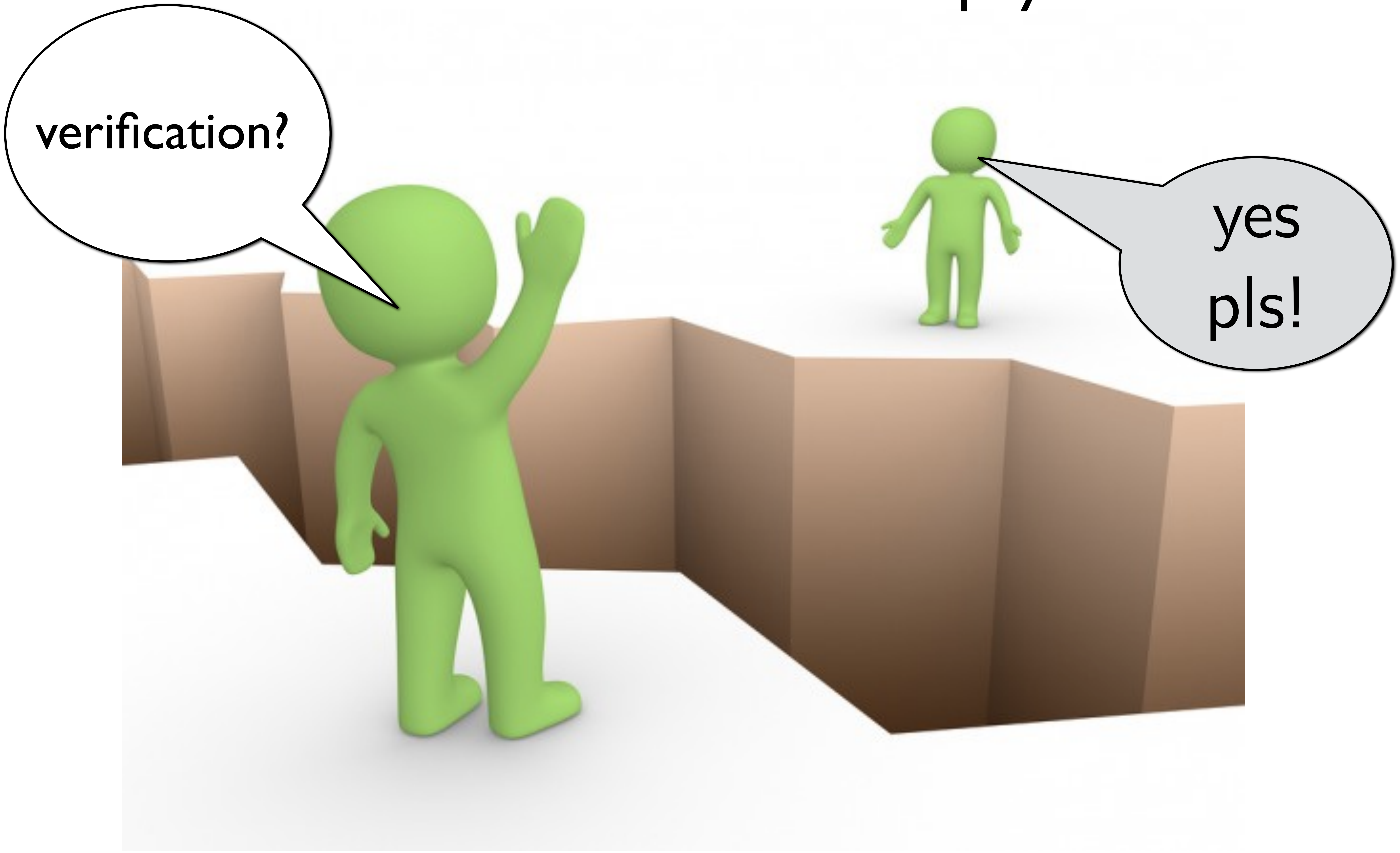
Verification

Did we implement the equations right?

Challenge

Telling these two apart when results are not as expected

natural & physical sciences



computer science

natural & physical sciences

computer science

natural & physical sciences

computer science

Let's bridge the chasm!

CamFort Lightweight verification tools for science

```fortran
program energy
  != unit kg :: mass
  != unit m  :: height
  real :: mass = 3.00, gravity = 9.91, height = 4.20
  != unit kg m**2/s**2 :: potential_energy
  real :: potential_energy

  potential_energy = mass * gravity * height
end program energy
```

```
$ camfort units-check energy1.f90

energy1.f90: Consistent. 4 variables checked.
```

EPSRC Engineering and Physical Sciences Research Council

Bloomberg

Met Office Hadley Centre

+ Static analysis checks

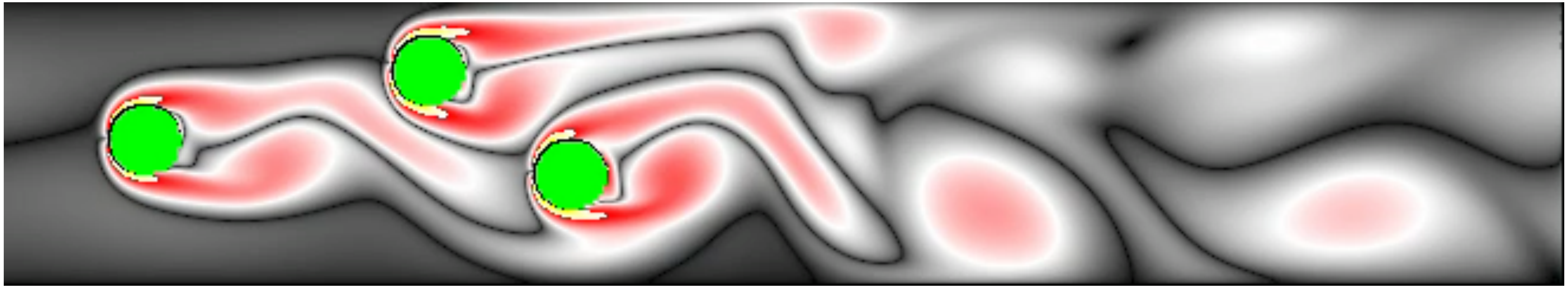
https://camfort.github.io/

Role of languages and tools and work in progress

Code often over-commits to implementation
(see accidental complexity)
What abstractions can avoid this?

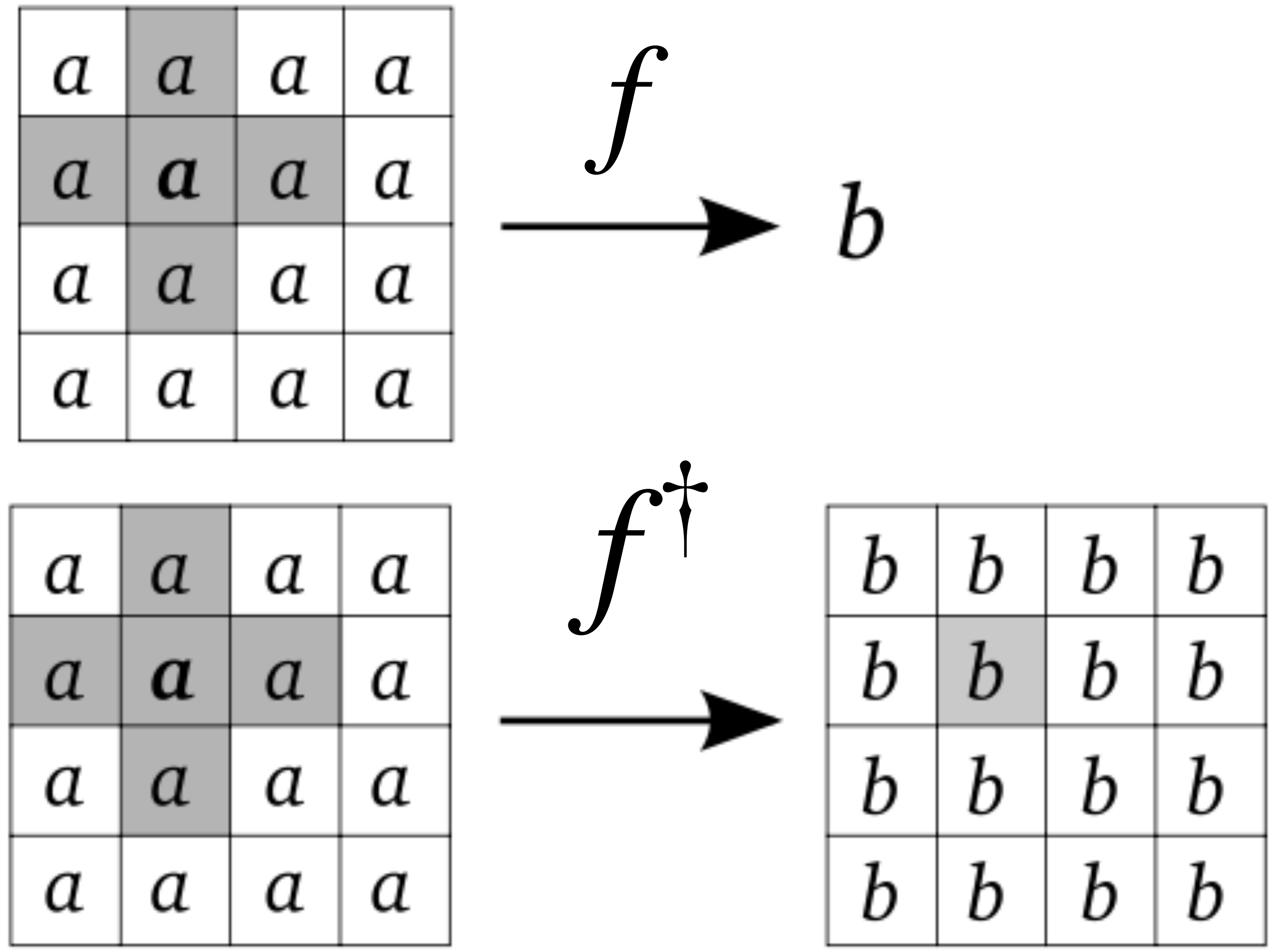
2. Categorical abstraction for grids

Fundamental dynamics (Navier-Stokes equations)



Conservation of momentum + mass for viscous fluid

Discrete approximation of PDE solution via <u>stencil computations</u>



A quantitative analysis of array usage in scientific code

	Package
climate	UM
climate/economics	E3ME
bio/climate	Hybrid4
chem/climate	GEOS-Chem
fluids	Navier
physics	CP
library	BLAS
library	ARPACK-NG
geodynamics	SPECFEM3D
library	MUDPACK
seismology	Cliffs

- Array computations are common in science (133k)
- Majority are *stencils* (55.86% of array comps.)

~2.5 million physical loc (Fortran 77/90)

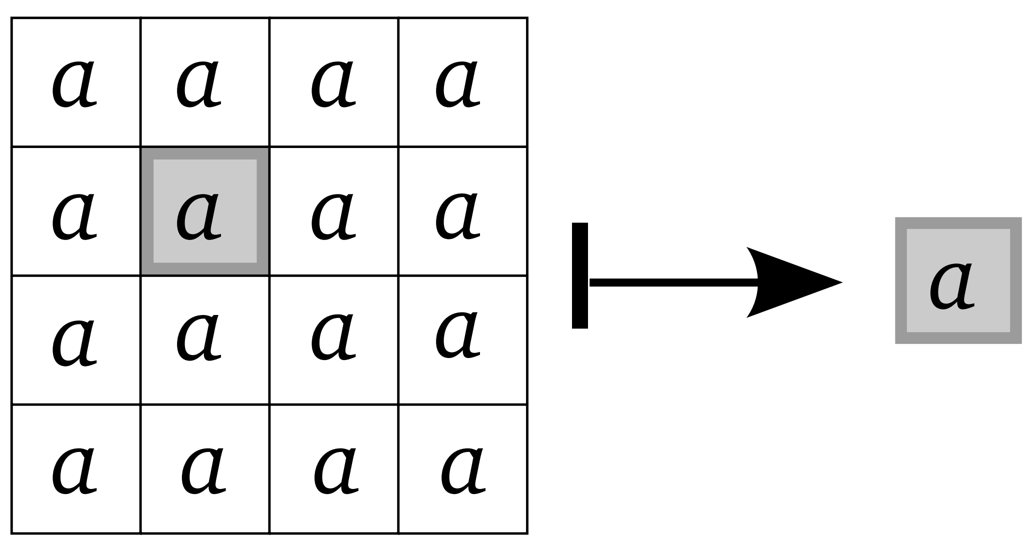
Verifying spatial properties of array computations (Orchard et al. 2017)

Stencil computations on arrays are comonadic

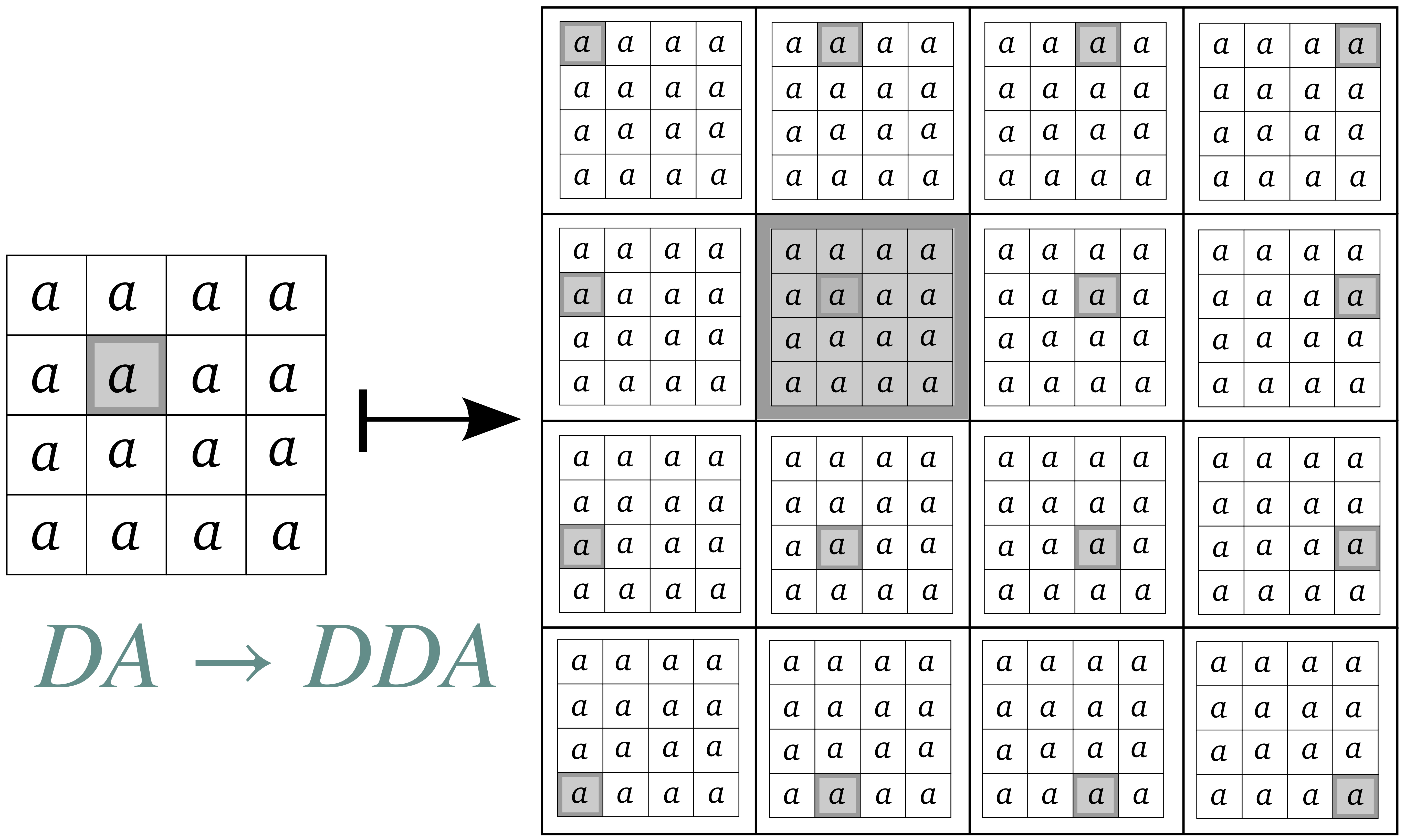
$$DA = \text{Array } I\ A \times I$$

array-data × cursor

Pointed array comonad



$\varepsilon_A : DA \to A$



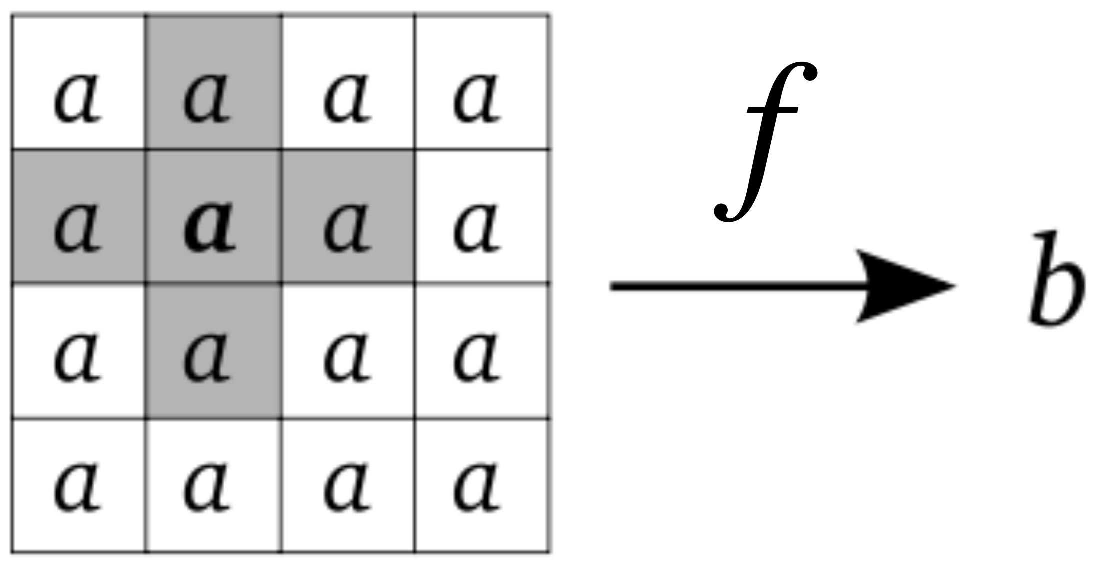
$\delta_A : DA \to DDA$

Ypnos: declarative, parallel structured grid programming (Orchard et al. 2010)
A Categorical Outlook on Cellular Automata (Capobianco, Uustalu, 2010)

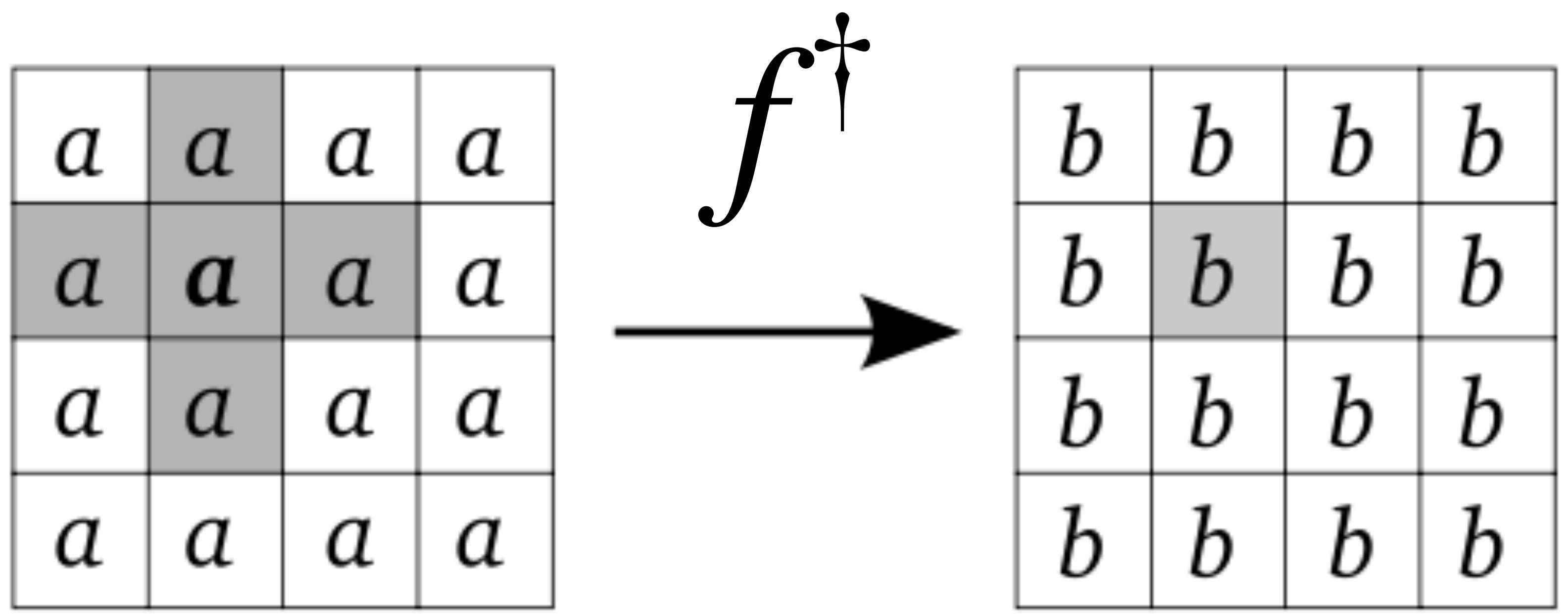
$$DA = \mathsf{Array}\ I\ A \times I$$

array-data × cursor

Pointed array comonad



Local computation (neighbourhood)



Global computation

$$f^{\dagger} = Df \circ \delta$$

$$\frac{DA \xrightarrow{f} B}{DA \xrightarrow{\delta_A} DDA \xrightarrow{Df} DB}$$

Ypnos: declarative, parallel structured grid programming (Orchard et al. 2010)
A Categorical Outlook on Cellular Automata (Capobianco, Uustalu, 2010)

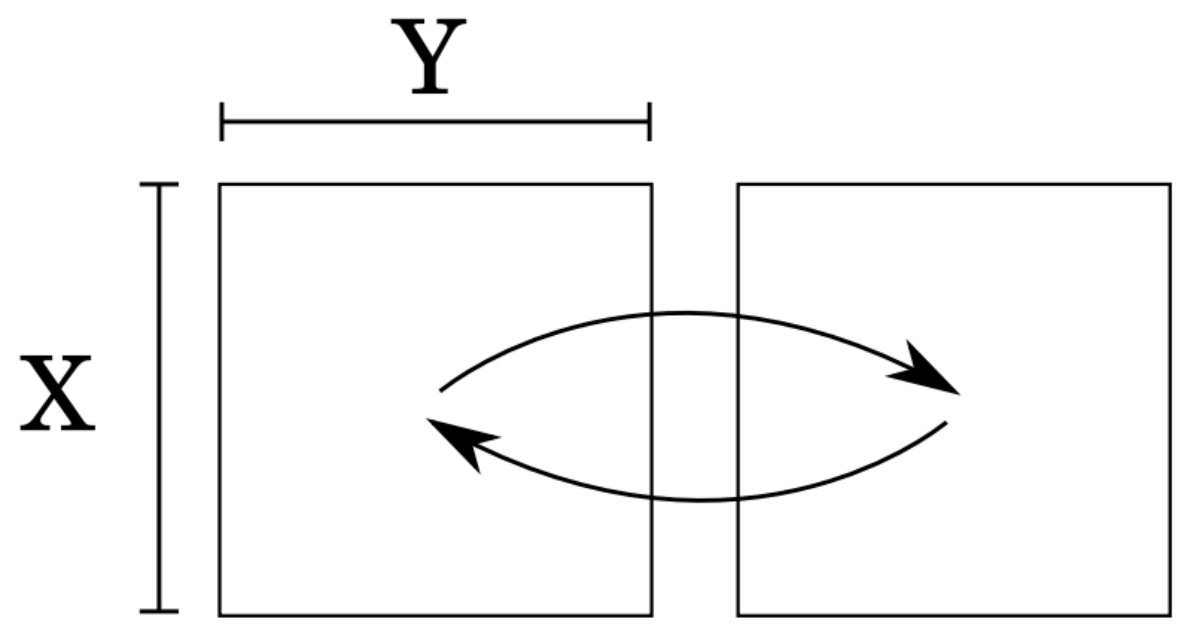
Double-buffering array "comonad"

$$DA = \mathsf{IArray}\ I\ A \times \mathsf{MArray}\ I\ A \times I$$

read-array-data × write-array-data × cursor

$$\frac{DA \xrightarrow{f} A}{DA \xrightarrow{f^\dagger} DA}$$

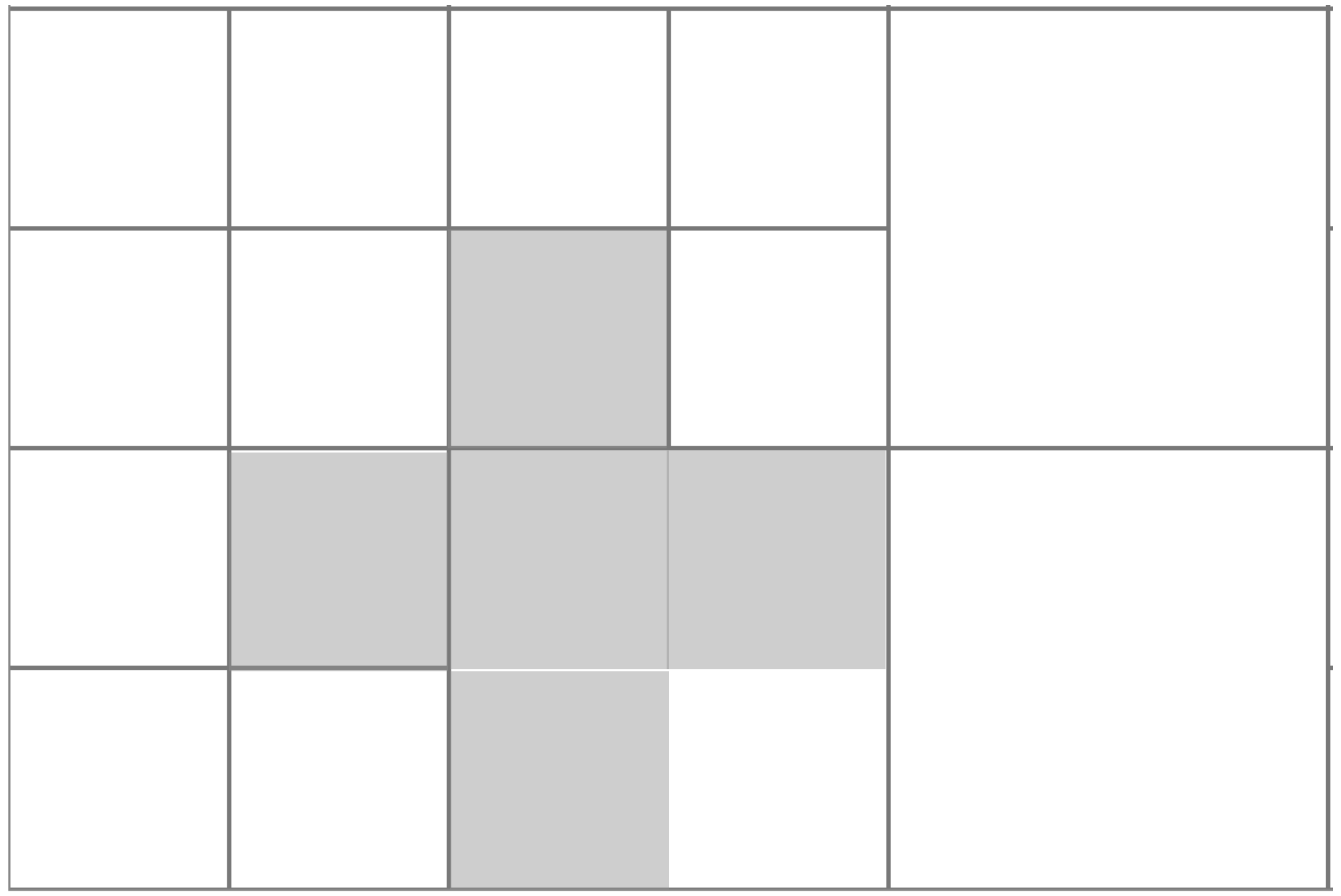
read from IArrray; write to MArray; then swap



General idea: hide optimisations behind abstract interface (e.g., mutation, stencil tiling)

Spatial data structures (quad trees, adaptive mesh)

$$SA = \text{Tree}\ (\text{Array}\ A)$$



Stencil computation
on flat view

$$\frac{SA \xrightarrow{view} DA \xrightarrow{f} B}{SA \xrightarrow{f^\dagger} SB}$$

General idea: hide representation via comonad morphism (*view*)

Categorical abstraction for grids

- Comonads for stencil computations
- Comonad morphisms to map between grid representations

cf. multi-grid configurations in SpeedyWeather.jl

https://github.com/SpeedyWeather/SpeedyWeather.jl

- *Distributive law* with monad for filter and reduction

$$\text{filter} : D(A+1) \to (DA) + 1$$
$$\text{reduce} : D(A \times M) \to (DA) \times M$$

3. Transparent and explainable computation

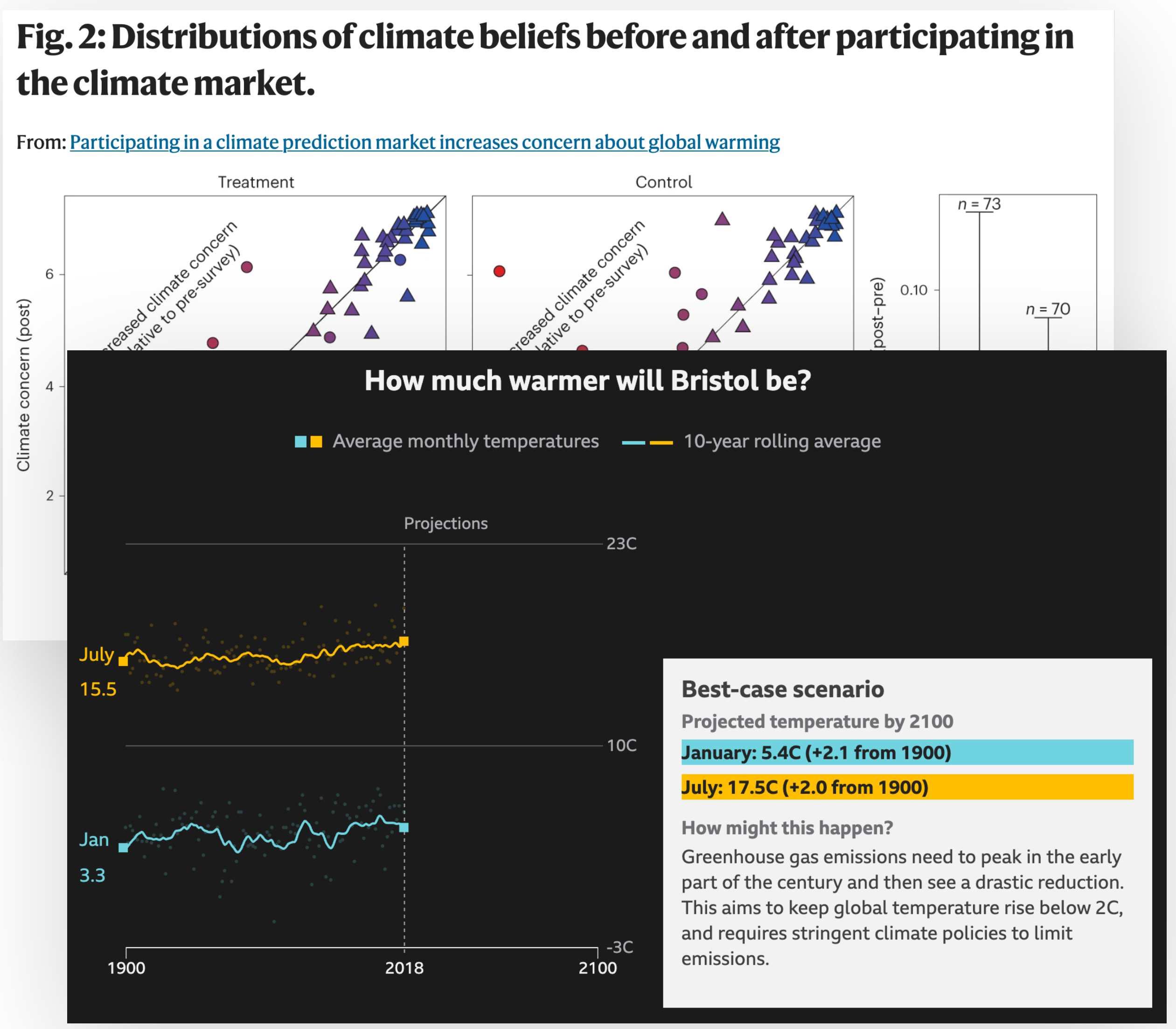
Fluid: A Transparent Programming Language

Research prototype at http://f.luid.org

Roly Perera

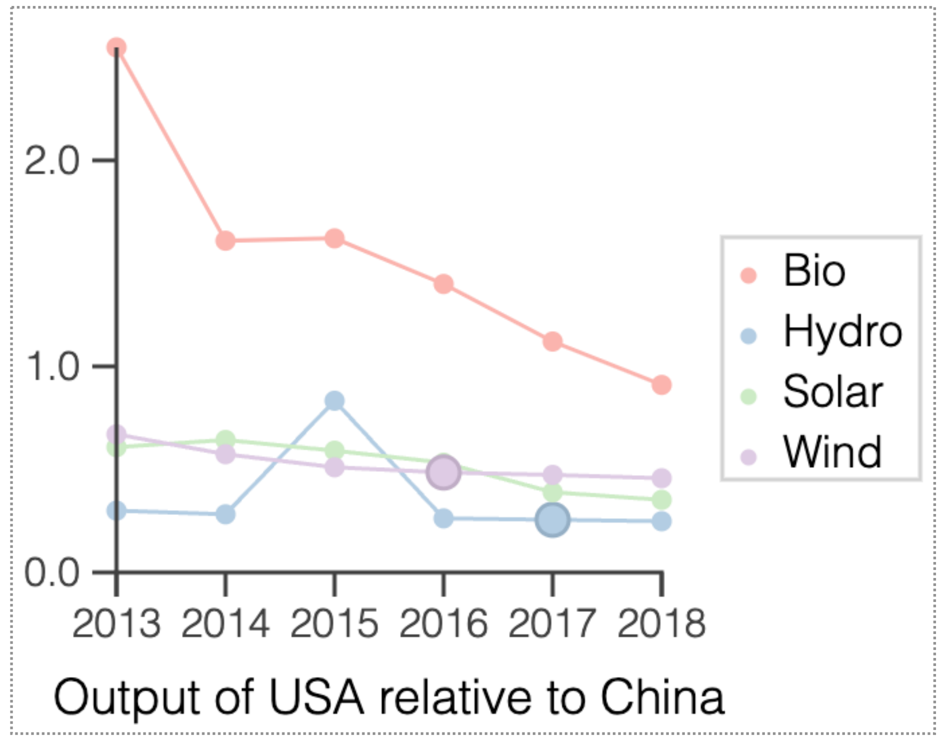
Joe Bond
(Bristol)

Problem



Research papers and news articles are opaque — hard to critique, understand or trust

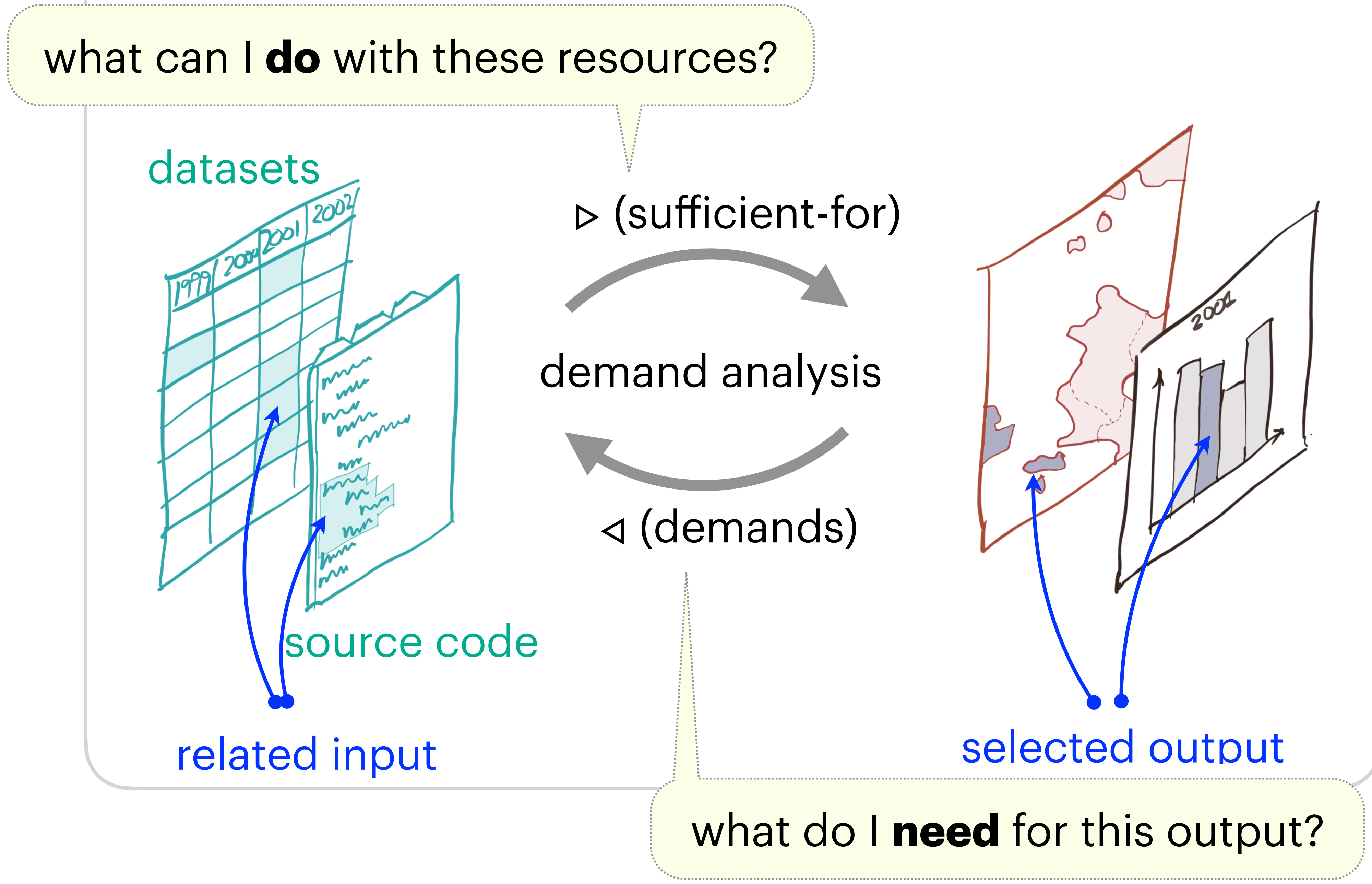
Research goal



year	*energyType*	*China*	*USA*	*=USA/China*
2016	Wind	169	82	0.49
2017	Hydro	313	80	0.26

Data-driven artefacts able to reveal relationship to underlying data

Methodology



Adjoint operators ◁ ⊣ ▷ and De Morgan duals ▶ ⊣ ◀ exposing fine-grained I/O relationships

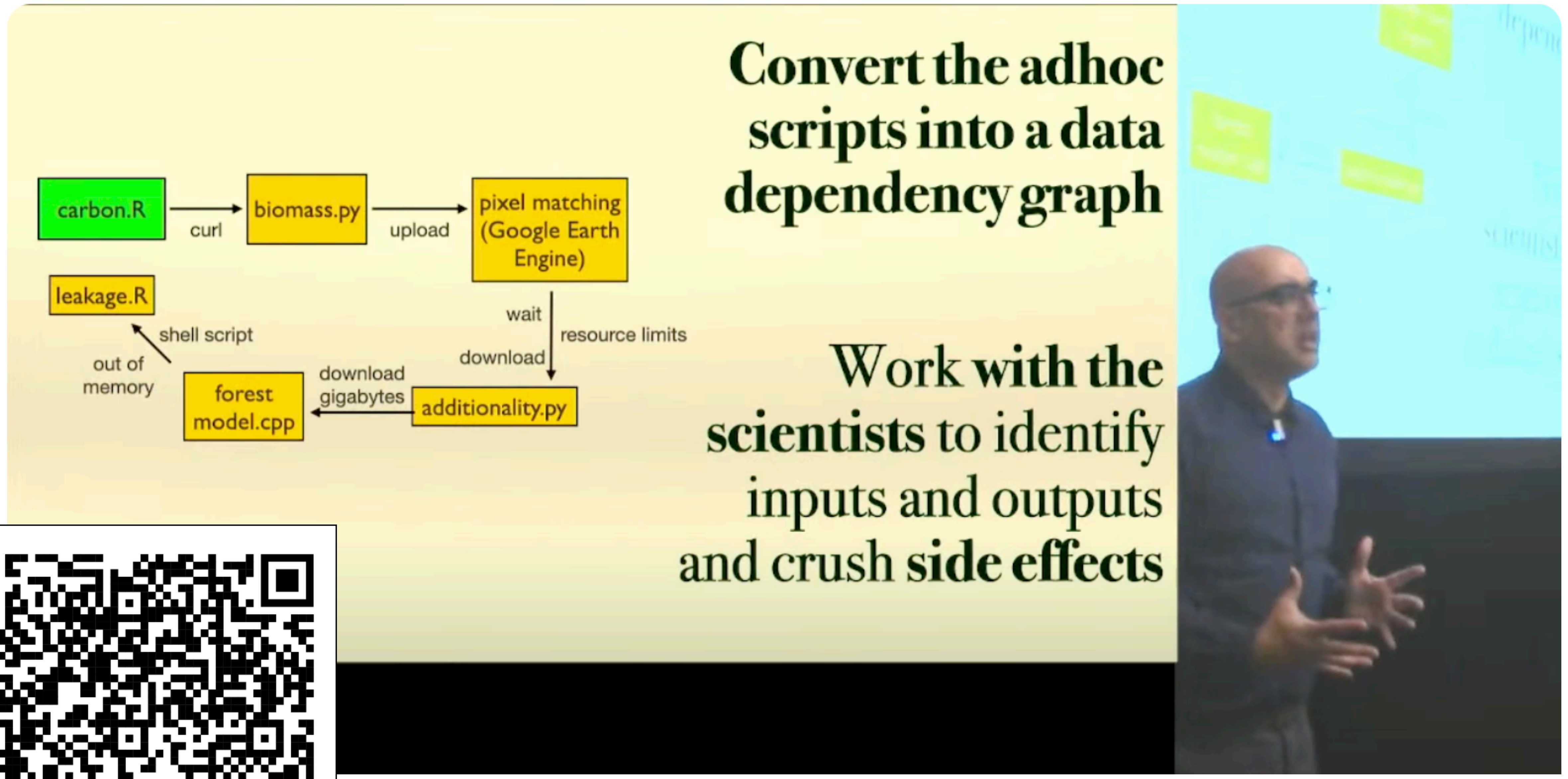
Programming languages for climate modelling through 2030-50?

- Science oriented (cf. Julia)
- Array oriented (cf. Fortran, Matlab, Julia, numpy, xarray, xgcm)
- Fast and predictable performance (cf. Fortran)
- Machine-learning integrated (cf. Python, PyTorch, pyro)
- Interactive (cf. REPLs and Notebooks)
- Heterogeneous compilation (CPUs, GPUs, NPUs, parallel arch.)

- Low-commitment to implementation details
- Lightweight verification (various typing approaches)
- Explainable and transparent
- Integration with program synthesis tooling

Data (a lot of it...)

See Anil Madhavapeddy's ICFP 2023 keynote



A Case for Planetary Computing

Amelia Holcomb, Michael Dales, Patrick Ferris, Sadiq Jaffer, Thomas Swinfield, Alison Eyres, Andrew Balmford, David Coomes, Srinivasan Keshav, Anil Madhavapeddy

Departments of Computer Science & Technology, Plant Sciences and Zoology, University of Cambridge

Abstract

We make a case for *planetary computing*: accessible, interoperable and extensible end-to-end systems infrastructure to process petabytes of global remote-sensing data for the scientific analysis of environmental action. We discuss some pressing scientific scenarios, survey existing solutions and find them incomplete, and present directions for systems research to help reverse the climate and biodiversity crises.

1 INTRODUCTION

There are simultaneous crises across the planet due to rising CO_2 emissions [60], rapid biodiversity loss [59], and desertification [42]. Assessing progress on these complex and interlocked issues requires a global view on the effectiveness of our adaptations and mitigations. To succeed in the coming decades, we need a wealth of new data about our natural environment that we rapidly process into accurate indicators, with sufficient trust in the resulting insights to make decisions that affect the lives of billions of people worldwide.

The scale of the problem demands that we shift beyond depending solely on governmental policies. Tackling the climate and biodiversity emergencies now involves ecologists, climate scientists, executives, journalists, and politicians — all assessing the current environmental state of the world and predicting the impact of changes. They aim to provide information to both policy makers and the public about assessment of ongoing conservation interventions.

A global view on planetary health is possible due to the availability of remote sensing data from satellites in orbit [33], drones flying over natural habitats [57], and networks of ground-based measurement equipment [30]. However, the *systems* required to effectively ingest, clean, collate, process, explore, archive, and derive policy decisions from the raw data are presently not usable by non-CS-experts, not reliable enough for scientific and political decision making, and not widely and openly available to all interested parties. As the climate crisis deepens, the feedback loop between environmental hypotheses and resulting policy action is happening faster than ever, which makes it ripe for abuse from bad actors planetary data, but also building public trust in the resulting policy actions by enforcing standards of transparency, reproducibility, accountability and timeliness in the decision making. We first motivate this with scenarios we have gathered from scientists working on environmental science (§1.1) and distill some common requirements (§1.2). We find that existing solutions only partially solve the systems problems (§2), and so discuss directions towards a planetary computing platform that can be used non-CS-expert users (§3). Our aim is to grow a federated ecosystem that will span individual organisations, and also be survivable beyond any one entity controlling it in the longer term, and be sensitive to the necessity of access control from malicious actors (§4).

1.1 Motivating Environmental Scenarios

Calculating Extinction Rates. Ecologists assess areas of habitat data to generate worldwide extinction statistics [28], but must not reveal individual observation points or else species may come under threat from poachers [45]. To generate this aggregate data they combine satellite data (Landsat, MODIS, Copernicus, GEDI [33]) with readings collected manually over decades. The data is highly variable in quality and requires cleaning and normalisation, before machine learning is used to train models to interpolate missing data. Subsequently, the information gleaned from the data is used to direct habitat regeneration and protection efforts, but must be regenerated monthly as new data arrives. When challenged, it should be possible to reveal the provenance of conclusions to auditors, even from decades-old observations.

Land use policy. Food and fibre production trades off against natural habitats, and understanding where to do this requires jurisdictional land management [25]. A civil servant assessing different methods of evaluating the impact of land use changes on biodiversity needs to access datasets for their country that have a reasonable resolution (<100 metres/pixel and so 100GB/layer storage needed), across all the species on the IUCN extinction list (10000+ entries [39]), and go back 30 years. Similarly, natural resource managers rely on being

Lookout for….

PROPL - Workshop on Programming for the Planet

At POPL 2024
20-21st January
In London

https://popl24.sigplan.org/home/propl-2024

POPL 2024 Wed 17 - Fri 19 January 2024 London, United Kingdom

Attending Tracks Organization Search Series Sign in Sign up

POPL 2024 (series) / PROPL 2024 (series) /

Programming for the Planet (PROPL)

PROPL 2024

About Call for Papers

There are simultaneous crises across the planet due to rising CO_2 emissions, rapid biodiversity loss, and desertification. Assessing progress on these complex and interlocking issues requires a global view on the effectiveness of our adaptations and mitigations. To succeed in the coming decades, we need a wealth of new data about our natural environment that we rapidly process into accurate indicators, with sufficient trust in the resulting insights to make decisions that affect the lives of billions of people worldwide.

However, programming the computer systems required to effectively ingest, clean, collate, process, explore, archive, and derive policy decisions from the planetary data we are collecting is difficult and leads to artefacts presently not usable by non-CS-experts, not reliable enough for scientific and political decision making, and not widely and openly available to all interested parties. Concurrently, domains where computational techniques are already central (e.g., climate modelling) are facing diminishing returns from current hardware trends and software techniques.

PROPL explores how to close the gap between state-of-the-art programming methods being developed in academia and the use of programming in climate analysis, modelling, forecasting, policy, and diplomacy. The aim is to build bridges to the current practices used in the scientific community.

The first edition of this workshop will comprise:

- half day of invited talks
- half day of contributed talks (selected by the programme committee based on short abstracts)
- half day of 'working workshop brainstorming" format

Important Dates AoE (UTC-12h)

Tue 31 Oct 2023
Talk proposals deadline

Wed 15 Nov 2023
Notification

Sat 20 Jan - Sun 21 Jan 2024
Workshop

Chairs

Anil Madhavapeddy
University of Cambridge, UK
United Kingdom

Dominic Orchard
University of Kent, UK and University of Cambridge, UK

Lookout for….

Hiring a 3-year postdoc soon

https://plas4sci.github.io/

Programming Languages and Systems for Science laboratory

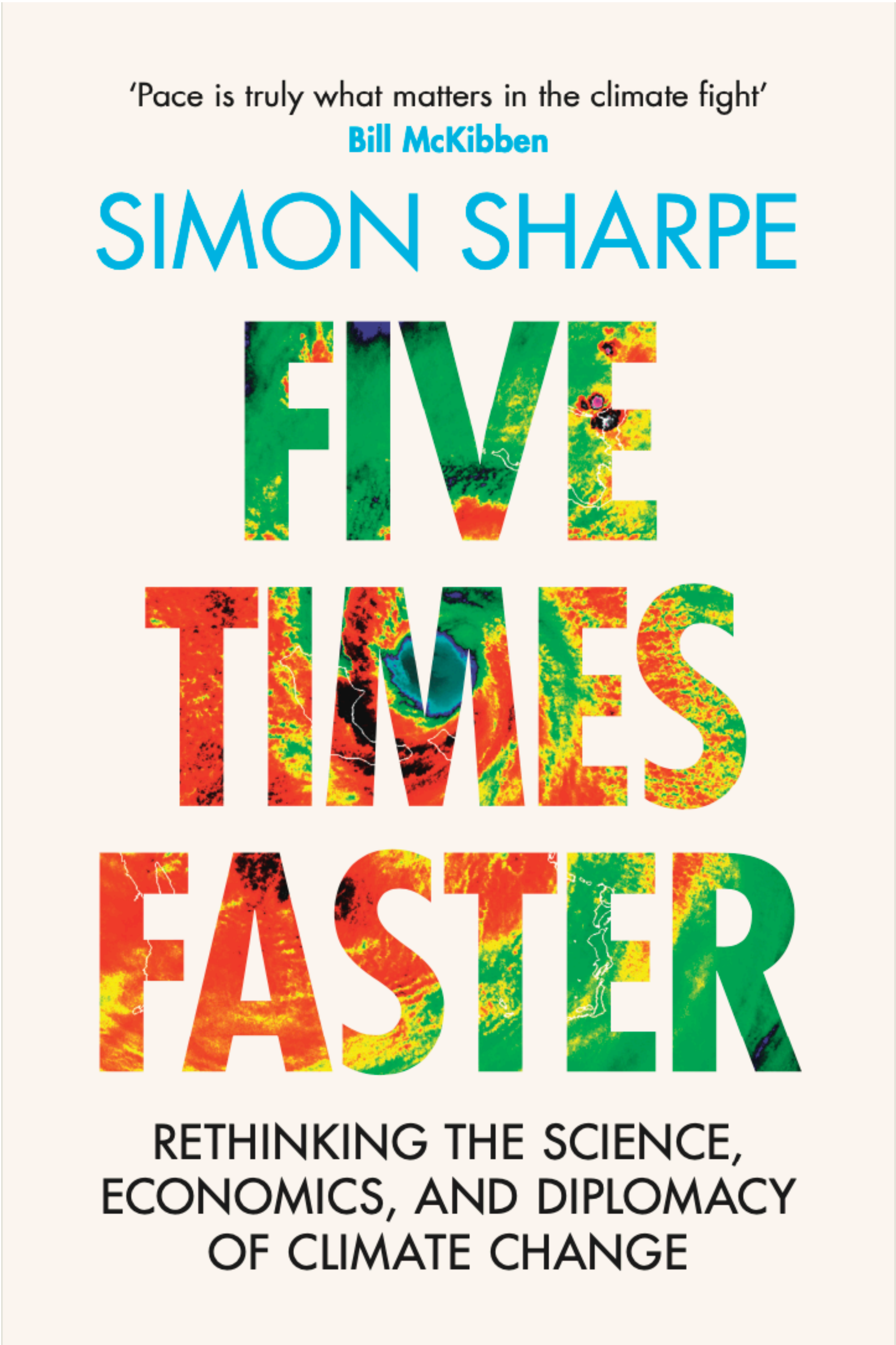
Complex models in modern science and are now routinely expressed as software. The PLAS4Sci lab (Programming Languages and Systems for Science) at the School of Computing, University of Kent is a sub-group of the PLAS group focussed on improving the state-of-the-art in programming languages, programming systems, and programming tools to support the daily work of scientists.

People

- Dominic Orchard - Lab lead
- Benjamin Orchard - Research Assistant and Research Software Engineer
- Laura Bocchi - Reader in Programming Languages
- Vilem-Benjamin Liepelt - PhD student

Partners

Projects



"Still, our appreciation of the risks of climate change is limited by the way our academic institutions encourage each researcher to focus on their own narrow area of expertise."

"Any actor should understand their points of leverage[…] We each have to understand the opportunities presented by our place in the system and do our best to exploit them."

@Cambridge_ICCS

Thanks https://iccs.cam.ac.uk