Lecture 8:

SciML for Weather Forecasting

Chris Budd OBE

Some papers

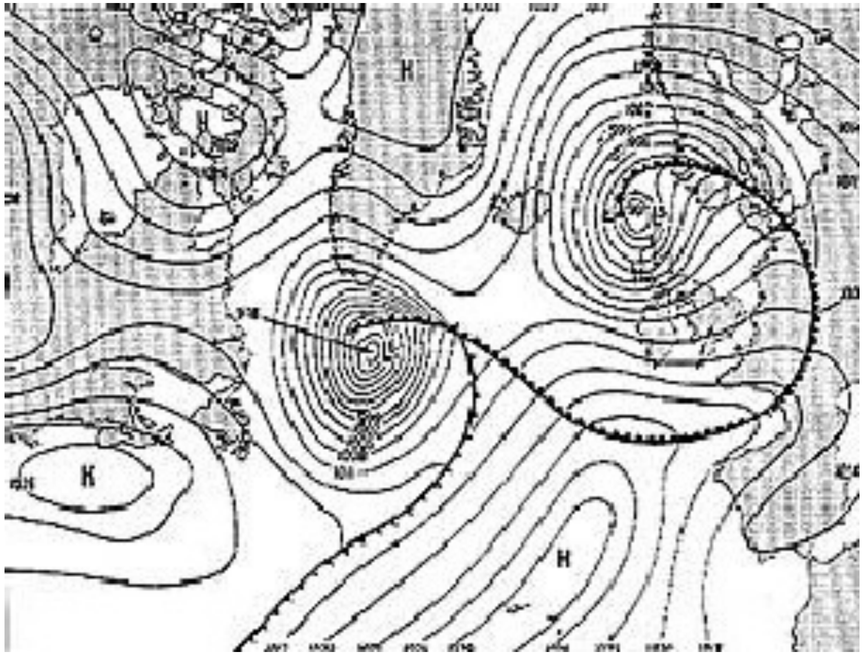
- How PINNs cheat: Predicting chaotic motion of a double pendulum

- Aardvark

- GraphCast
- MODELING CHAOTIC LORENZ ODE SYSTEM USING SCIENTIFIC
- MACHINE LEARNING

- Panda: A pretrained forecast model for universal representation of chaotic dynamics

Vital to get a weather forecast right : D Day June 5th/6th 1944

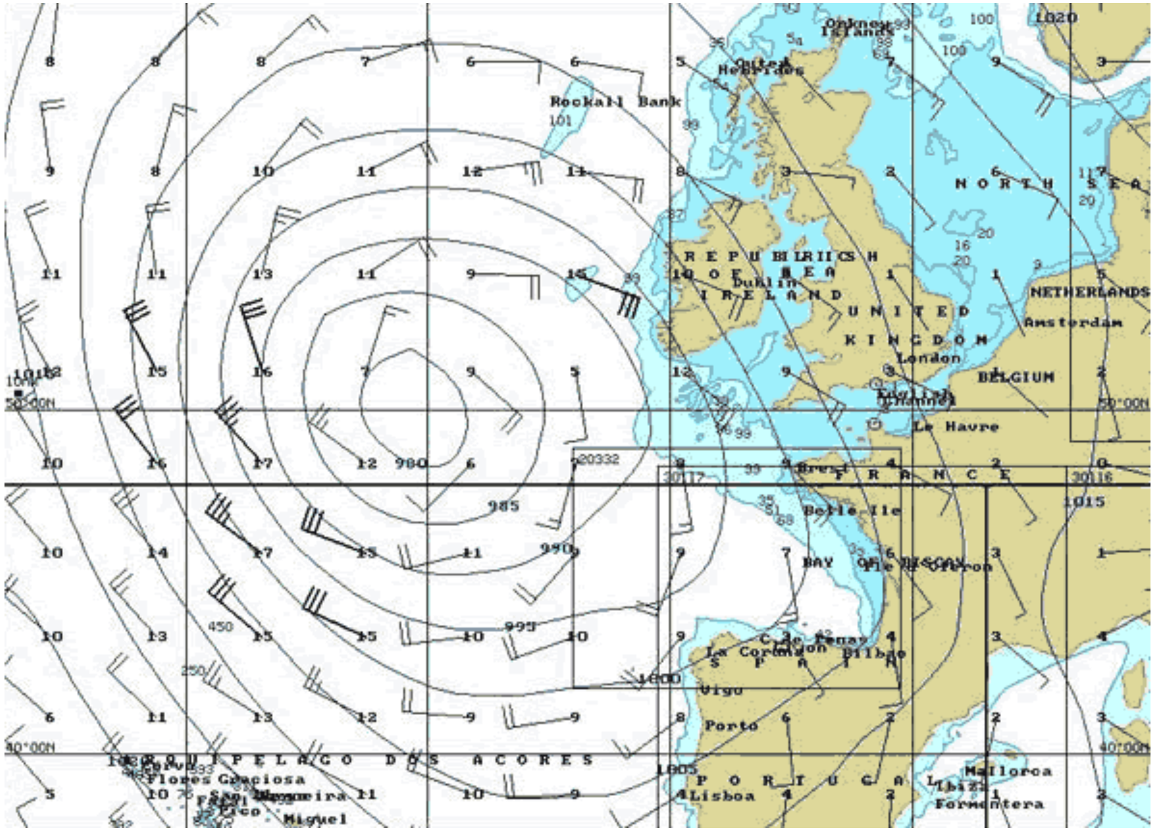


James Stagg, with a bit of Northern Irish help, got the forecast right.

The Germans got it wrong!

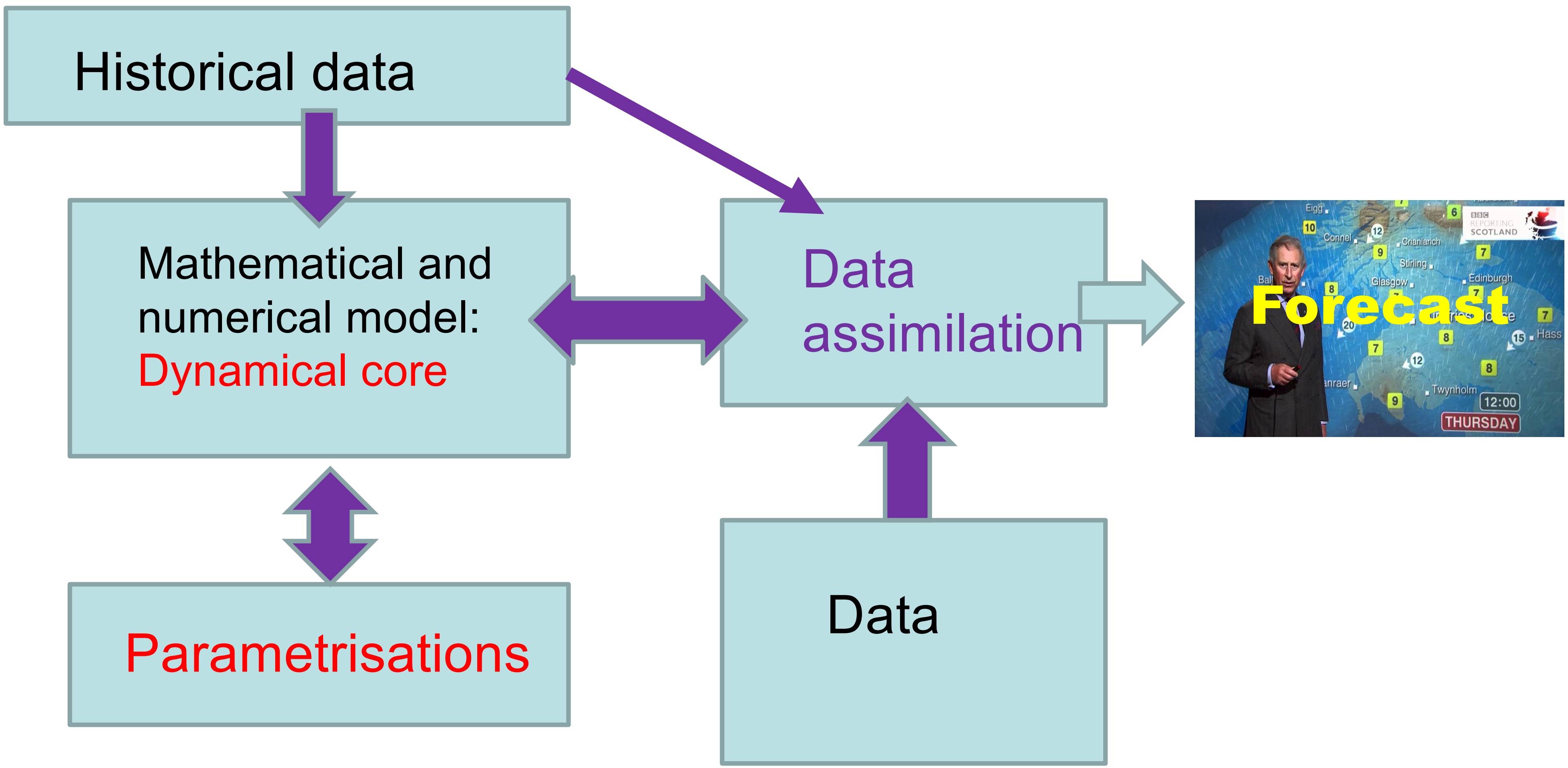
$$u_t + u.\nabla u = -\nabla P + \frac{1}{Re}\nabla^2 u, \qquad \nabla . u = 0.$$

Navier-Stokes equations



Weather forecasting under a week ahead

Weather forecasting process



Areas where ML is used in weather forecasting

1. Post-processing results eg. down scaling

2. Parametrisations eg. clouds

3. Dynamical core/physics: GraphCast

4. End-to-end/data only: Aardvark

Dynamical prediction of the weather

Physics/PINNs/Neural ODEs or similar

Complex interrelated processes described by differential equations

Basic equations: Navier-Stokes which describe the weather

$$\frac{Du}{Dt} + 2f \times u + \frac{1}{\rho}\nabla p + g = \nu\nabla^2 u,$$

Motion

$$\frac{\partial\rho}{\partial t} + \nabla.(\rho u) = 0,$$

Density

$$C\frac{DT}{Dt} - \frac{RT}{\rho}\frac{D\rho}{Dt} = \kappa_h\nabla^2 T + S_h + LP,$$

Temperature

$$\frac{Dq}{Dt} = \kappa_q\nabla^2 q + S_q - P,$$

Moisture

$$p = \rho RT.$$

Pressure

For climate add in ice, CO2, ocean currents, vegetation, …

$$\frac{Du}{Dt} + 2f \times u + \frac{1}{\rho}\nabla p + g = \nu \nabla^2 u,$$

$$\frac{\partial \rho}{\partial t} + \nabla .(\rho u) = 0,$$

$$C\frac{DT}{Dt} - \frac{RT}{\rho}\frac{D\rho}{Dt} = \kappa_h \nabla^2 T + S_h + LP,$$

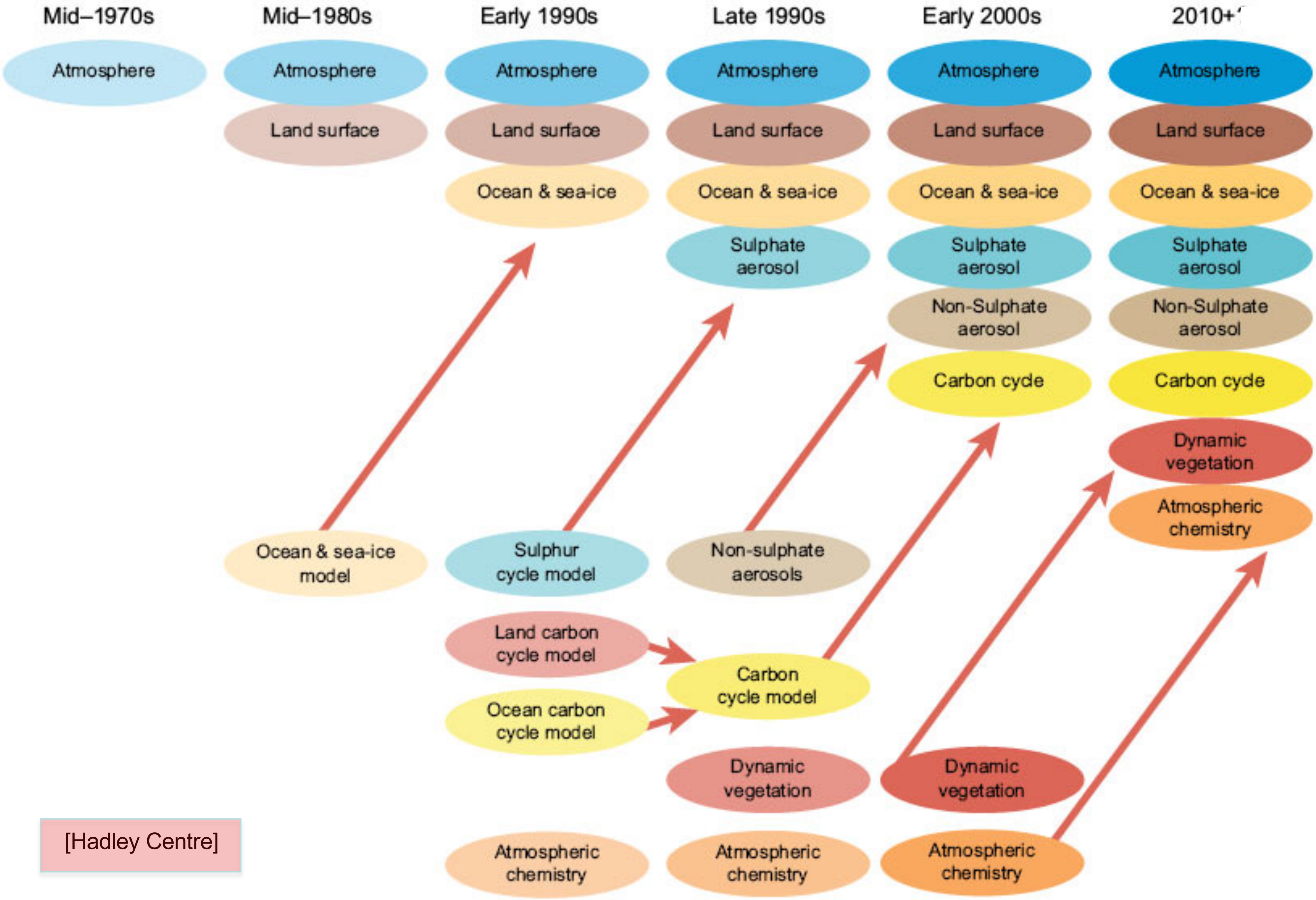
$$\frac{Dq}{Dt} = \kappa_q \nabla^2 q + S_q - P,$$

$$p = \rho RT.$$

Certain modelling simplifications make the models easier to analyse:

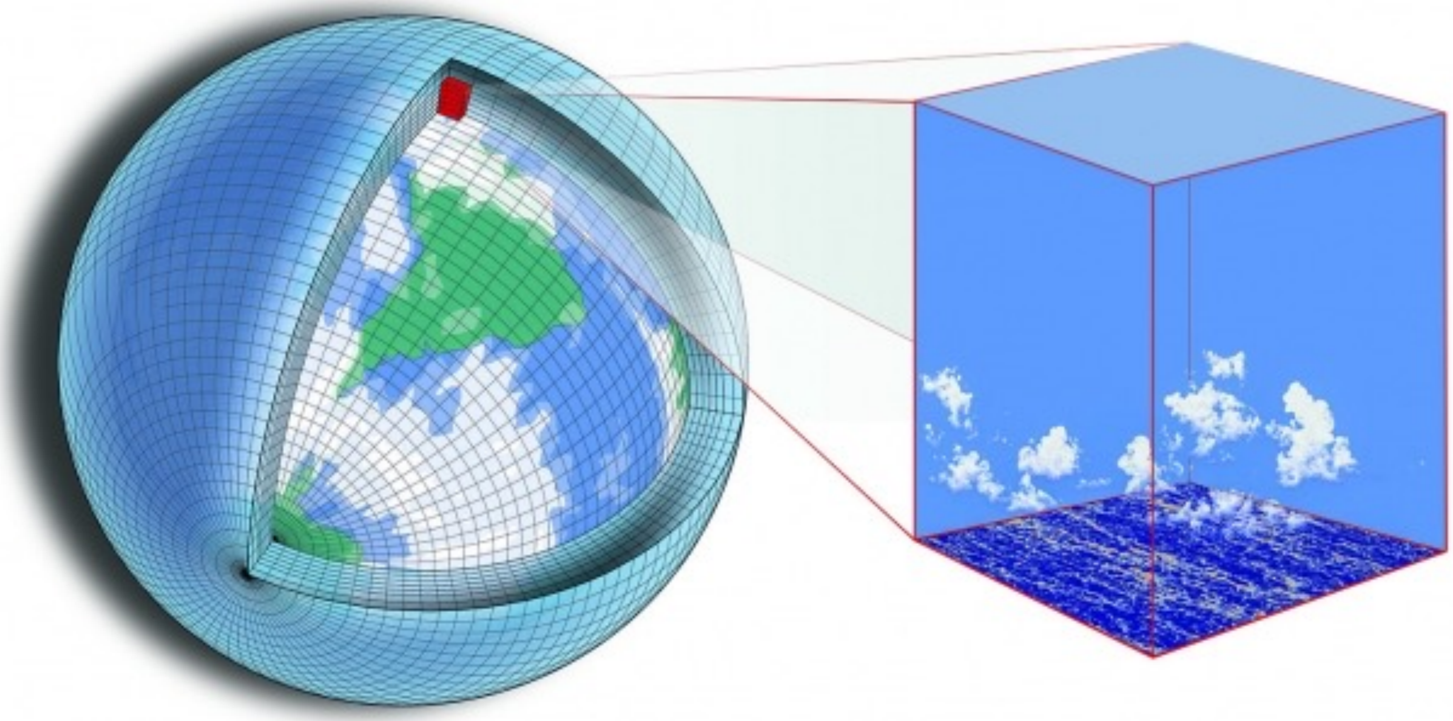
Geostrophic balance
Hydrostatic balance
Stratification

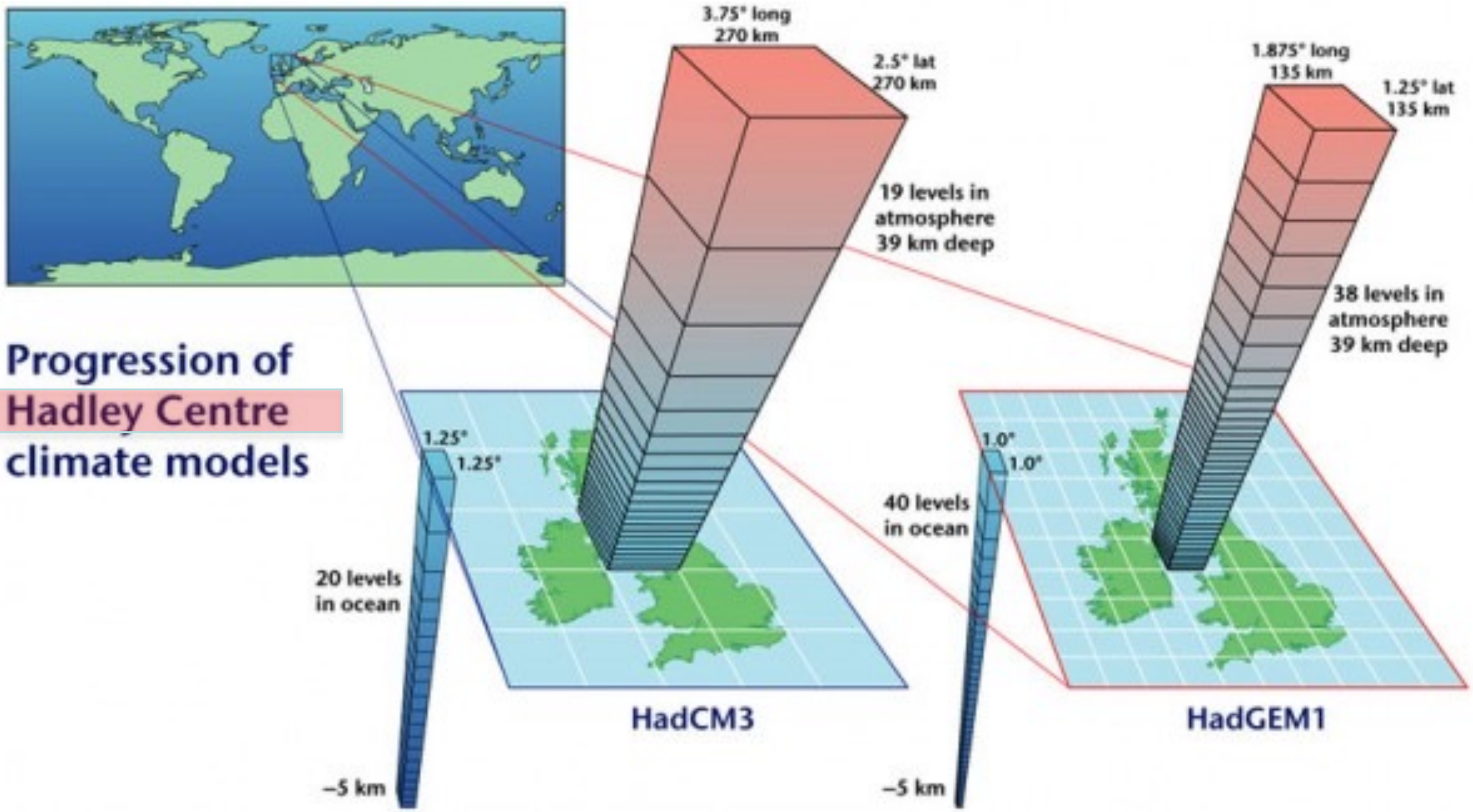
The Development of Climate Models:



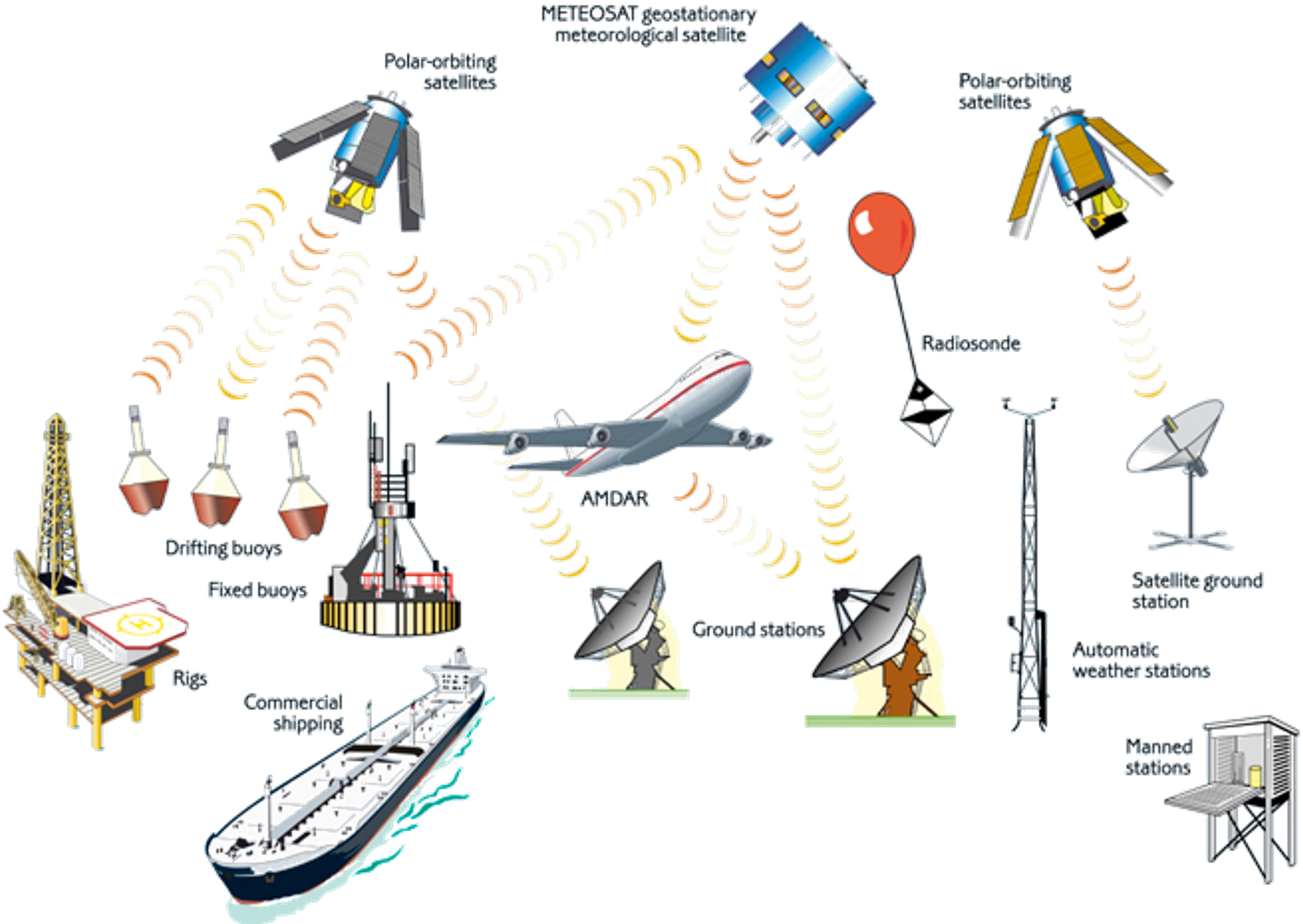
[Hadley Centre]

Equations are solved numerically




Progression of
Hadley Centre
climate models
3.75° long
270 km
2.5° lat
270 km
19 levels in
atmosphere
39 km deep
1.25°
1.25°
20 levels
in ocean
–5 km
HadCM3
1.875° long
135 km
1.25° lat
135 km
38 levels in
atmosphere
39 km deep
1.0°
1.0°
40 levels
in ocean
–5 km
HadGEM1

Data: Sources of observation



High performance numerical methods:

Finite volume, semi-Lagrangian, parallel, deep learning

Ten year testing programme!

The importance of chaos

Chaotic motion is complex, irregular and otherwise unpredictable behaviour which arises from a ‘simple’ system which can be exactly described by ‘simple’ mathematical laws.

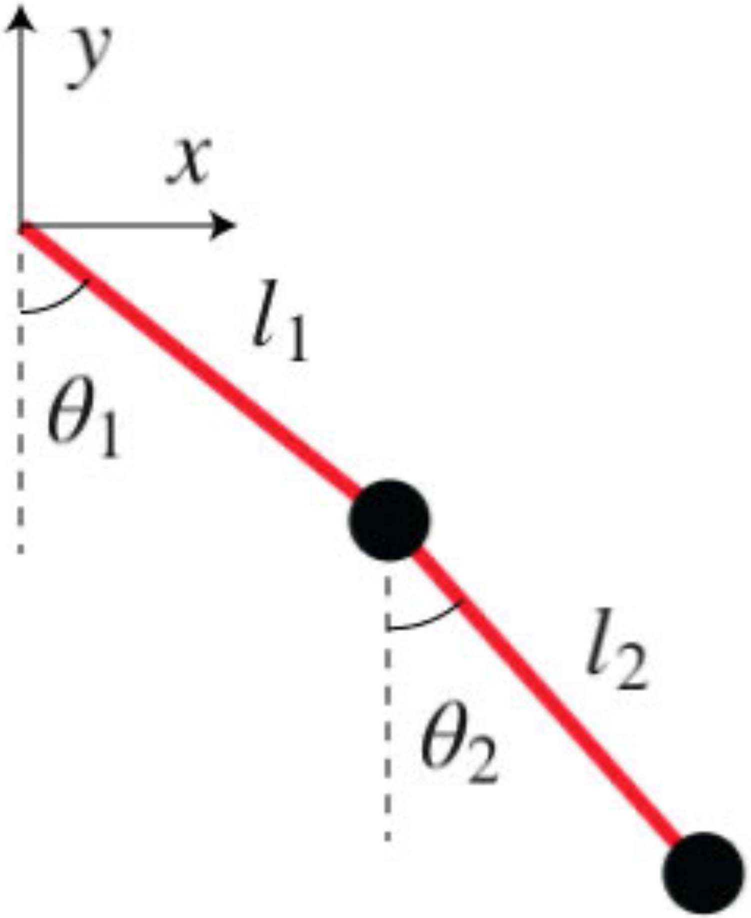
Manifested as sensitivity to initial conditions after the Lyapunov time

and complex trajectories on strange attractors

Newton's laws apply to the double pendulum!

θ_1 Angle of top part

θ_2 Angle of bottom part

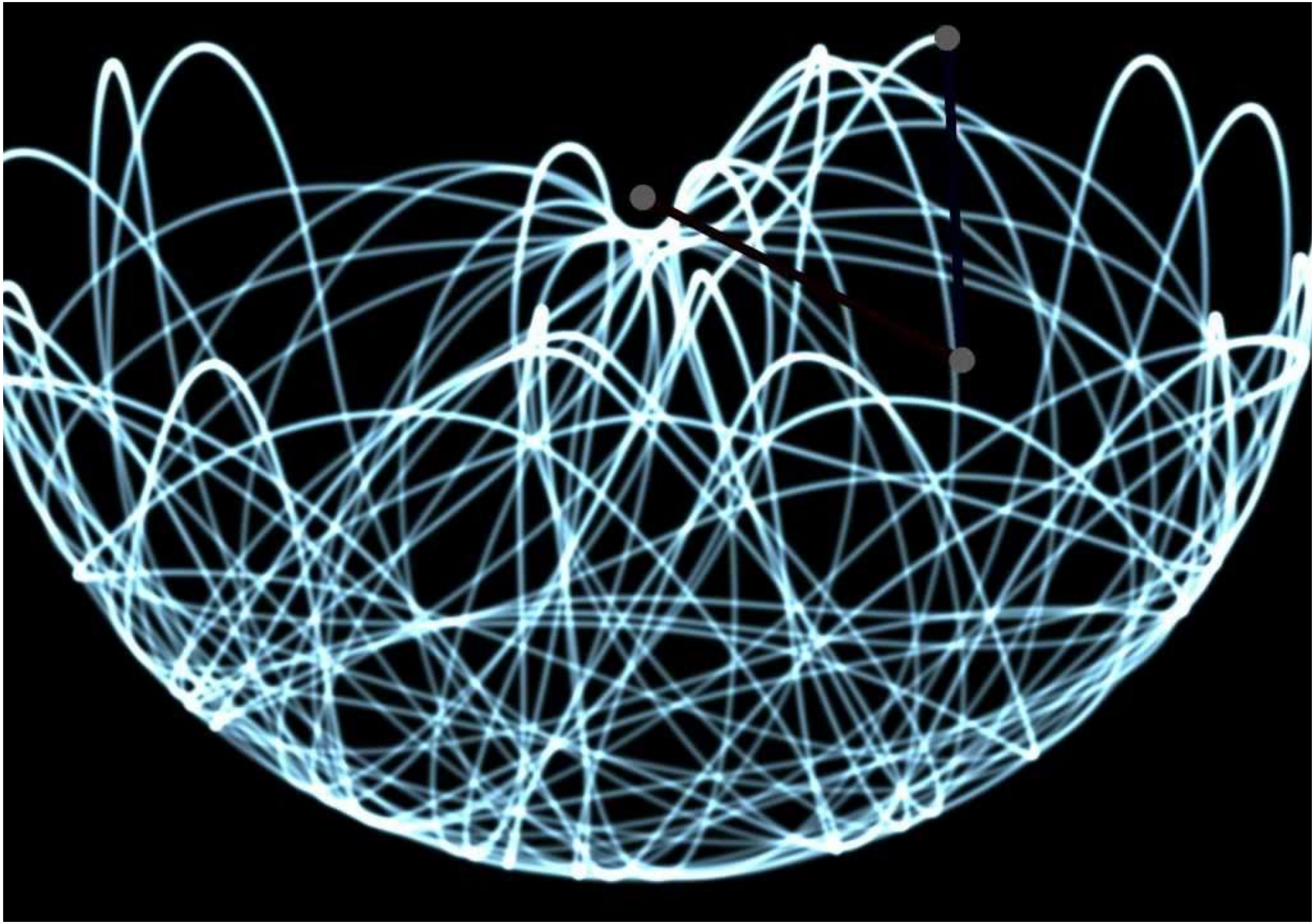


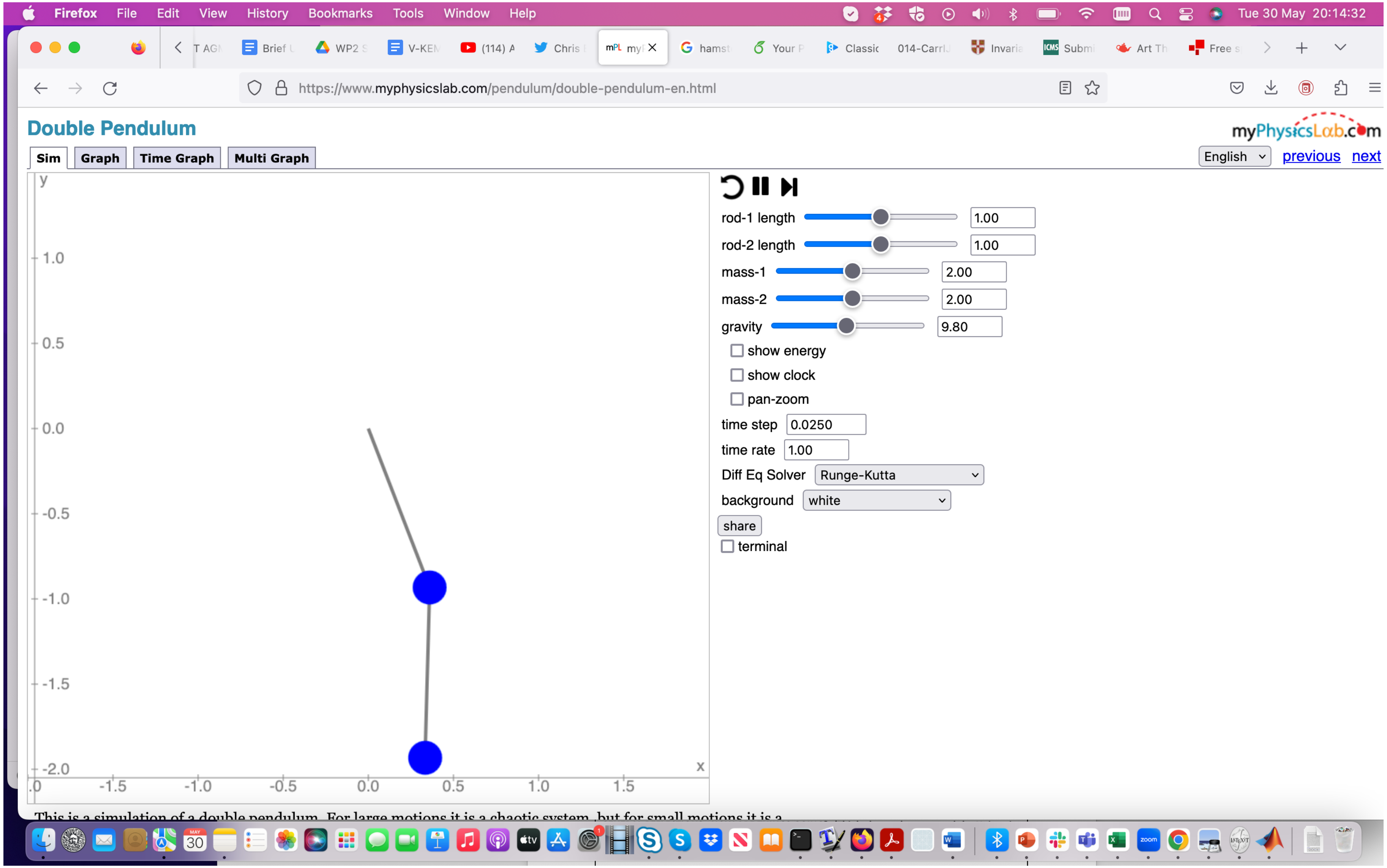
Motion is described by a coupled pair of second order nonlinear ordinary differential equations

$$(m_1 + m_2)l_1\ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1 - \theta_2) + g(m_1 + m_2)\sin(\theta_1) = 0$$

$$m_2 l_2 \ddot{\theta}_2 + m_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - m_2 l_1 \dot{\theta}_1^{\ 2} \sin(\theta_1 - \theta_2) + m_2 g \sin(\theta_2) = 0$$

- Small swings: Can be solved exactly to give in phase and out of phase solutions
- Large swings: Solvable numerically. Solutions are chaotic



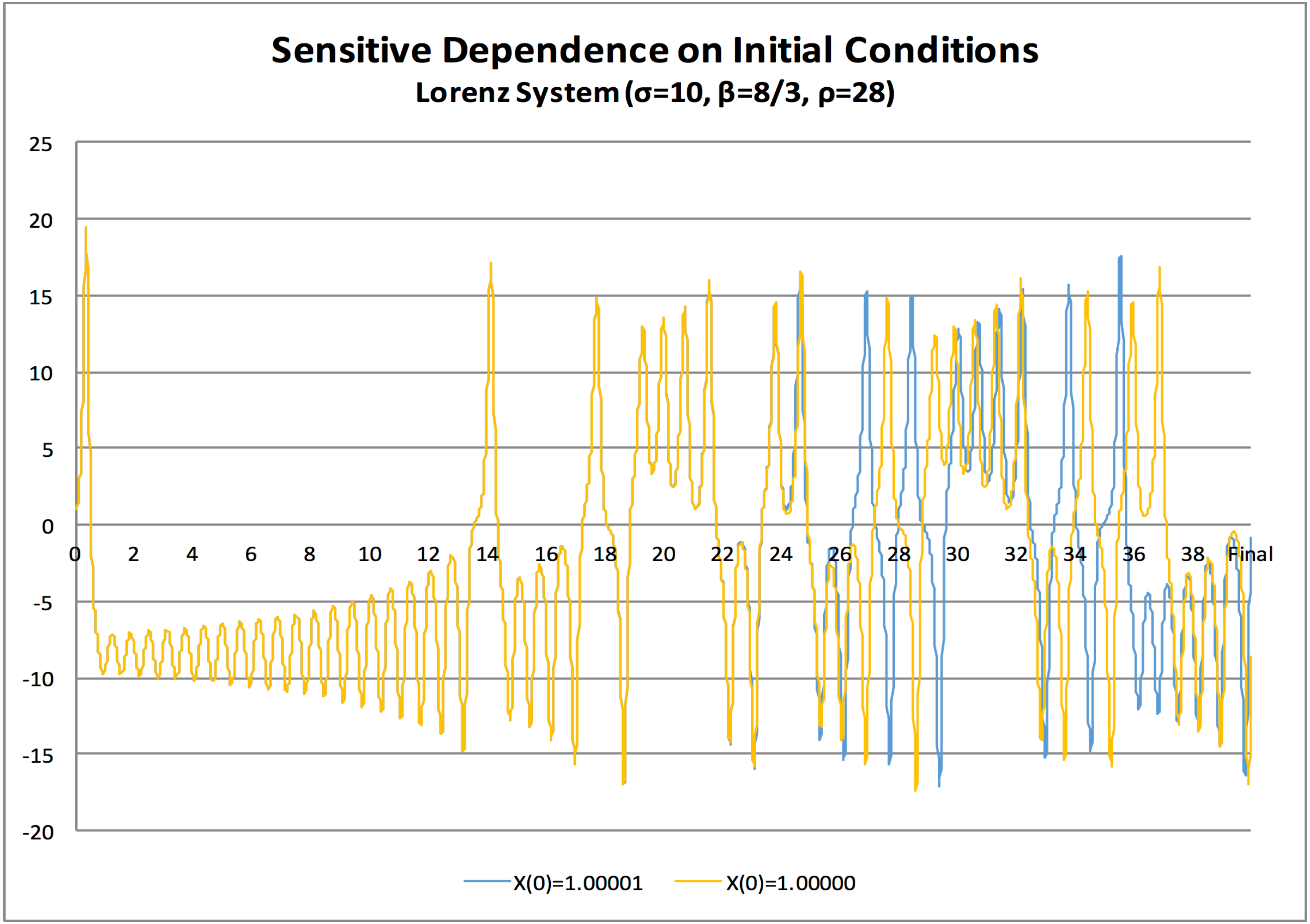

Firefox File Edit View History Bookmarks Tools Window Help
Tue 30 May 20:14:32
https://www.myphysicslab.com/pendulum/double-pendulum-en.html
Double Pendulum
myPhysicsLab.com
Sim
Graph
Time Graph
Multi Graph
English
previous
next
rod-1 length 1.00
rod-2 length 1.00
mass-1 2.00
mass-2 2.00
gravity 9.80
show energy
show clock
pan-zoom
time step 0.0250
time rate 1.00
Diff Eq Solver Runge-Kutta
background white
share
terminal
y
1.0
0.5
0.0
-0.5
-1.0
-1.5
-2.0
x
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
This is a simulation of a double pendulum. For large motions it is a chaotic system, but for small motions it is a

Ed Lorenz

Meteorologist Lorenz (1960s) was studying convection in the atmosphere and derived the Lorenz equations:

$$\boxed{\begin{aligned} \frac{dx}{dt} &= \sigma(y - x), \\ \frac{dy}{dt} &= x(\rho - z) - y, \\ \frac{dz}{dt} &= xy - \beta z. \end{aligned}}$$

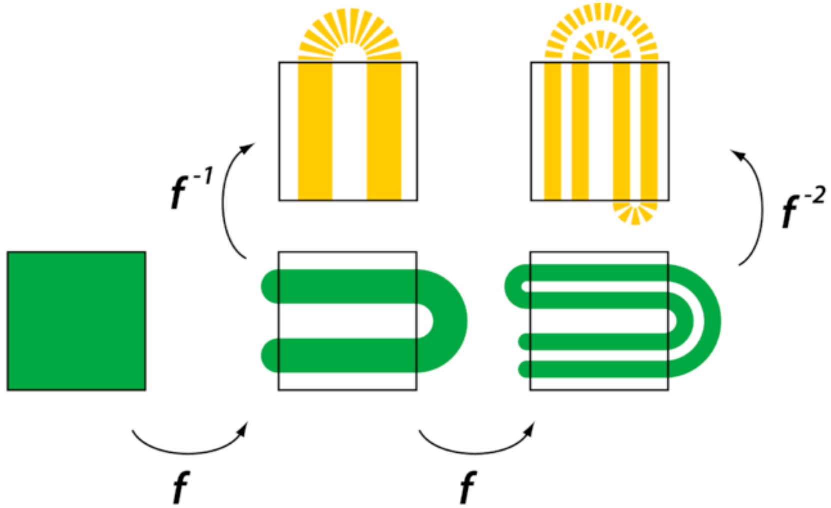
Computer studies showed the existence of chaotic solutions. This came as a complete surprise!



Sensitivity seen after about t = 25

Basic idea:

Chaotic systems locally stretch and fold phase space



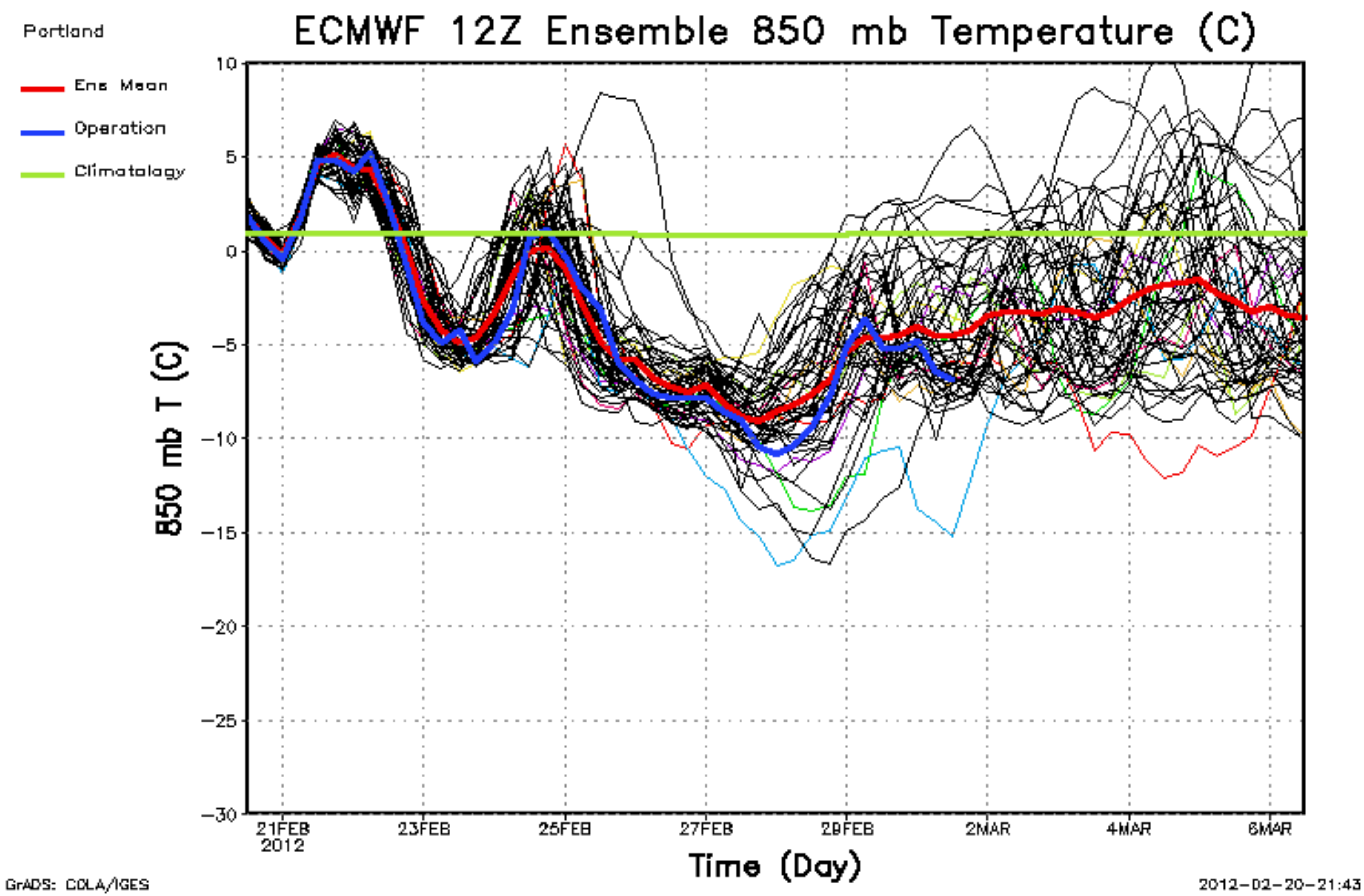
Perturbations p(t) initially grow exponentially fast

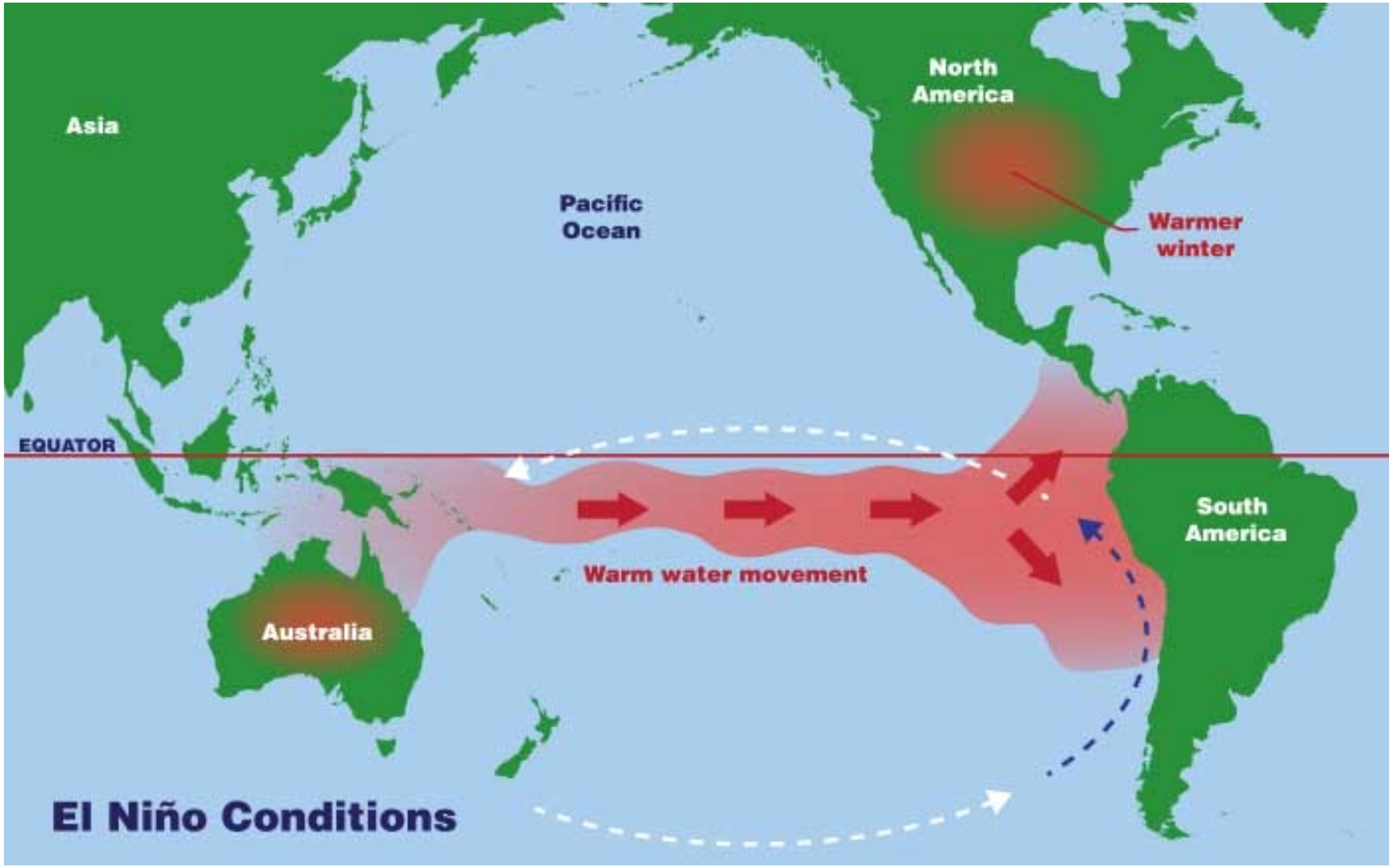
$$p(t) = p(0)\, e^{\lambda t}$$

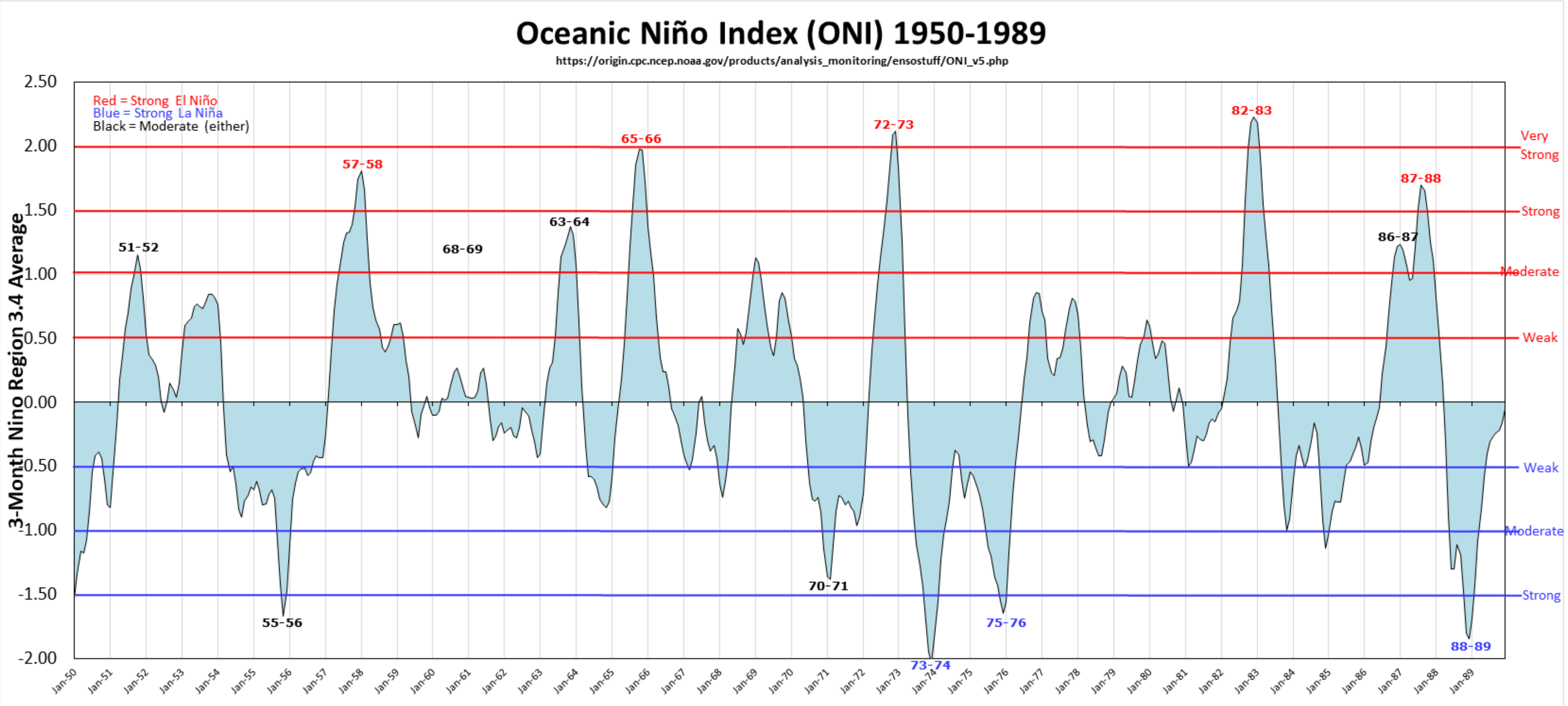
λ: Lyapunov exponent, $1/\lambda$ Lyapunov time

But are globally bounded eg. $p(t) = p(0) \sin\left(e^{\lambda t}\right)$

Sensitive dependence of the weather after Lyapunov time: about 10 days



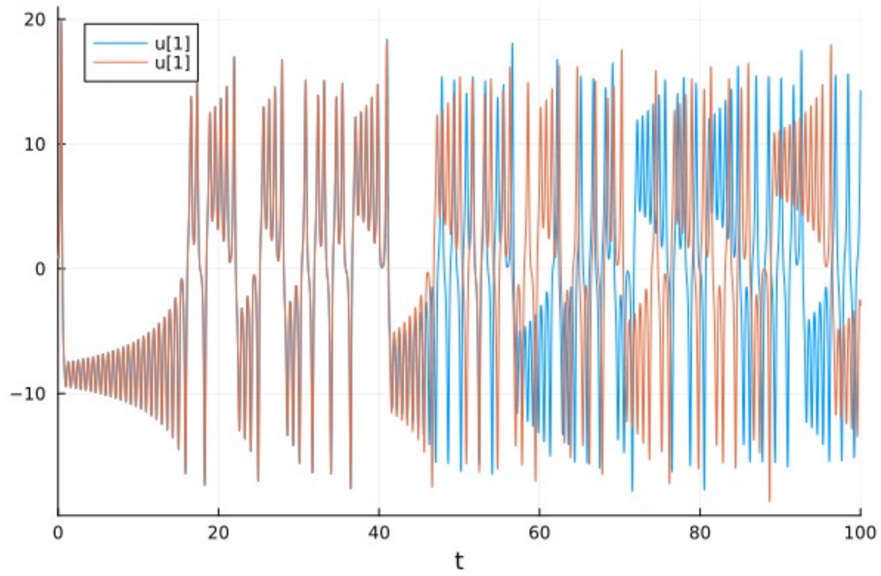

Asia
Pacific
Ocean
North
America
Warmer
winter
EQUATOR
South
America
Warm water movement
Australia
El Niño Conditions



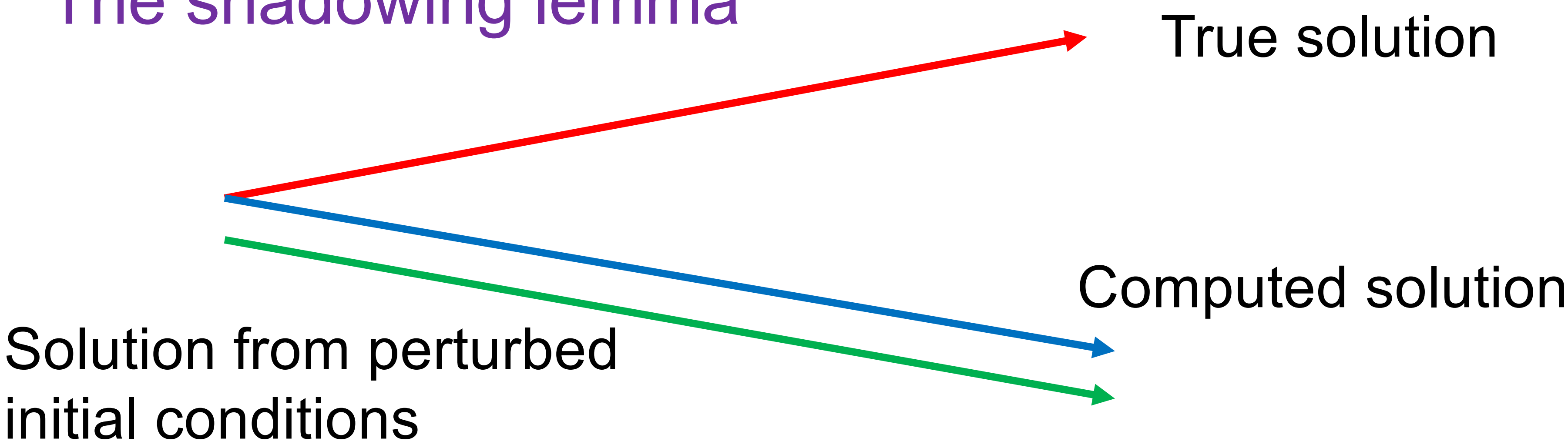
Oceanic Niño Index (ONI) 1950-1989
https://origin.cpc.ncep.noaa.gov/products/analysis_monitoring/ensostuff/ONI_v5.php
Red = Strong El Niño
Blue = Strong La Niña
Black = Moderate (either)
3-Month Nino Region 3.4 Average
2.50
2.00
1.50
1.00
0.50
0.00
-0.50
-1.00
-1.50
-2.00
Very Strong
Strong
Moderate
Weak
Weak
Moderate
Strong
51-52
55-56
57-58
63-64
65-66
68-69
70-71
72-73
73-74
75-76
82-83
86-87
87-88
88-89
Jan-50
Jan-51
Jan-52
Jan-53
Jan-54
Jan-55
Jan-56
Jan-57
Jan-58
Jan-59
Jan-60
Jan-61
Jan-62
Jan-63
Jan-64
Jan-65
Jan-66
Jan-67
Jan-68
Jan-69
Jan-70
Jan-71
Jan-72
Jan-73
Jan-74
Jan-75
Jan-76
Jan-77
Jan-78
Jan-79
Jan-80
Jan-81
Jan-82
Jan-83
Jan-84
Jan-85
Jan-86
Jan-87
Jan-88
Jan-89

Help is at hand: Chaotic systems can still be 'predicted'

- The shadowing lemma
- Ergodic properties
- Dense chaotic attractors
- Data assimilation



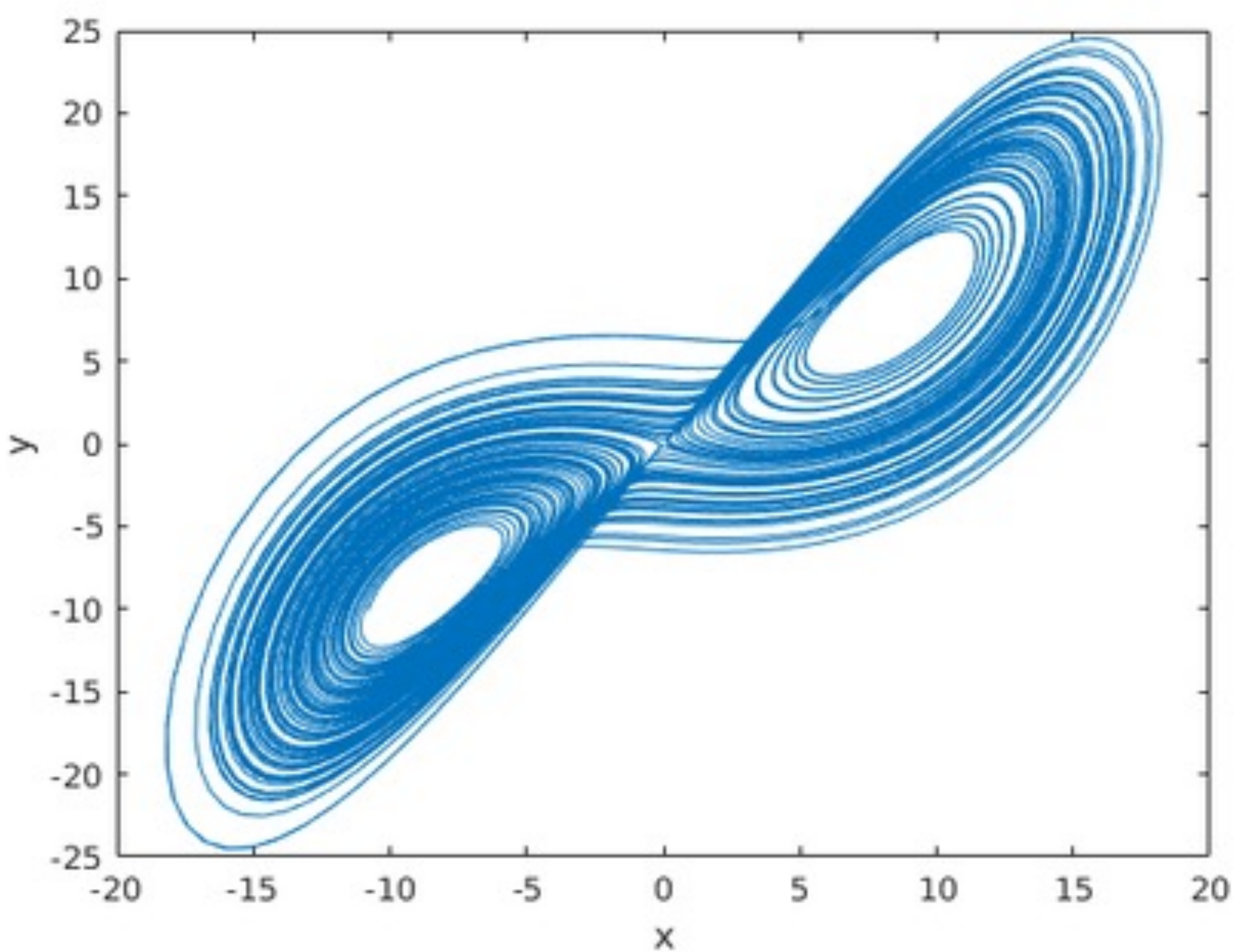
The shadowing lemma



The perturbed solution we get due to small errors in a numerical (eg. ML) computation is not the real solution, but it is close to a solution of the system from some other initial condition (or parameters) that are epsilon close.

i.e. The numerical solution gets the overall topology of the solution of the system correct

Lorenz Eqns again: Low dimensional Strange Attractor



Numerical method makes correct ergodic predictions: ‘climate’

Be warned about over hyped claims of PINNS for predicting chaotic systems

- System may not be chaotic for the parameters chosen
- System may not have been run for long enough (i.e. before the Lyapunov time)
- We may be seeing a shadowing phenomenon (not a bad thing but any decent solver can do that)

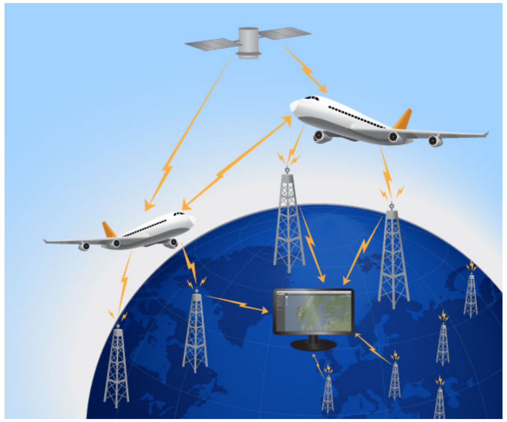
“Our results demonstrate that PINNs do not exhibit any sensitivity to perturbations in the initial condition. Instead, the PINN optimization consistently converges to physically correct solutions that violate the initial condition only marginally, but diverge significantly from the desired solution due to the chaotic nature of the system. In fact, the PINN predictions primarily exhibit low-frequency components with a smaller magnitude of higher-order derivatives, which favors lower physics loss values compared to the desired solution. We thus hypothesize that the PINNs “cheat” by shifting the initial conditions to values that correspond to physically correct solutions that are easier to learn.”

Basic Idea of Data Assimilation

True state: of a system is x_t

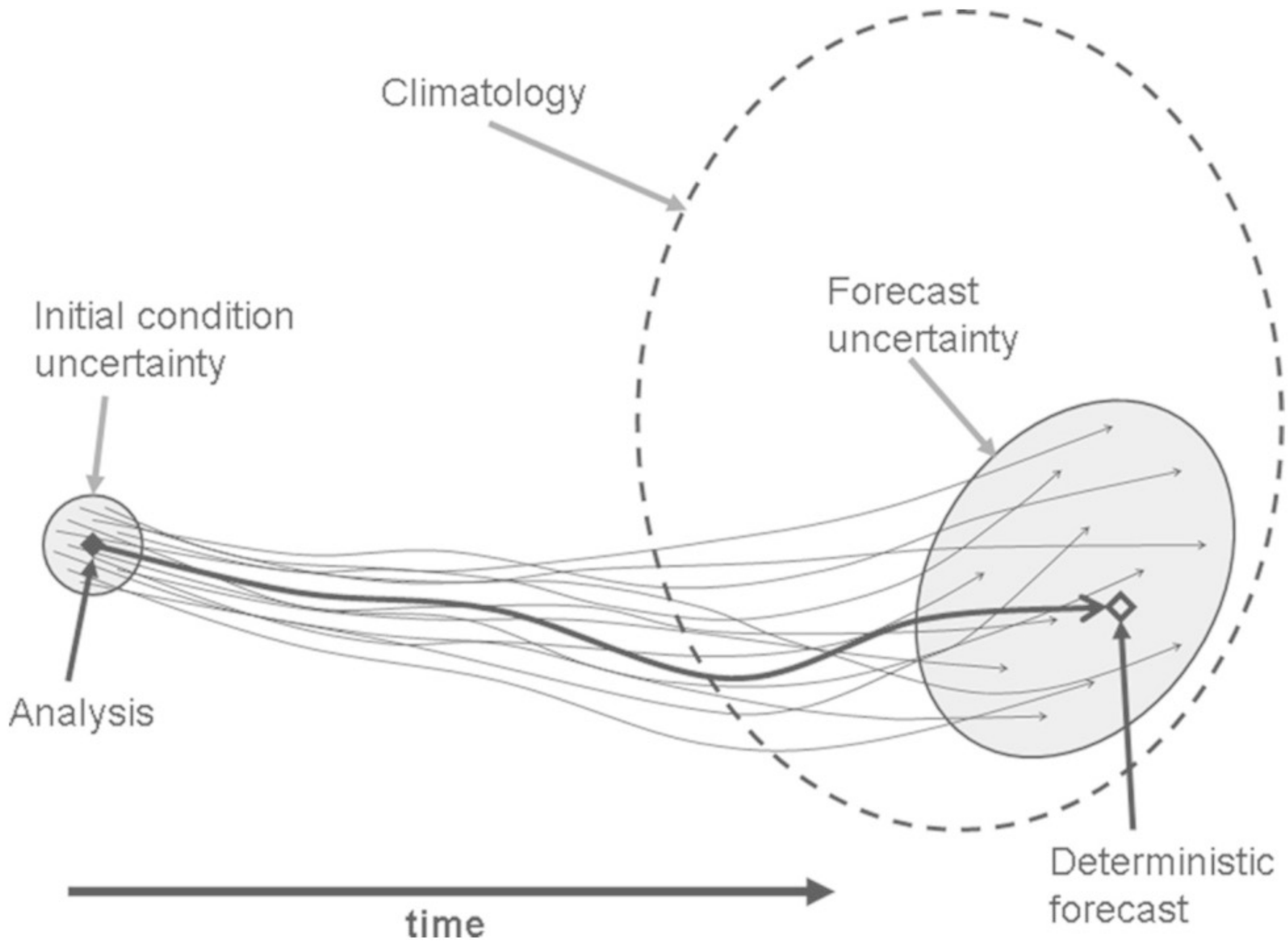
Predicted state: x_b

Data: observations y of some function

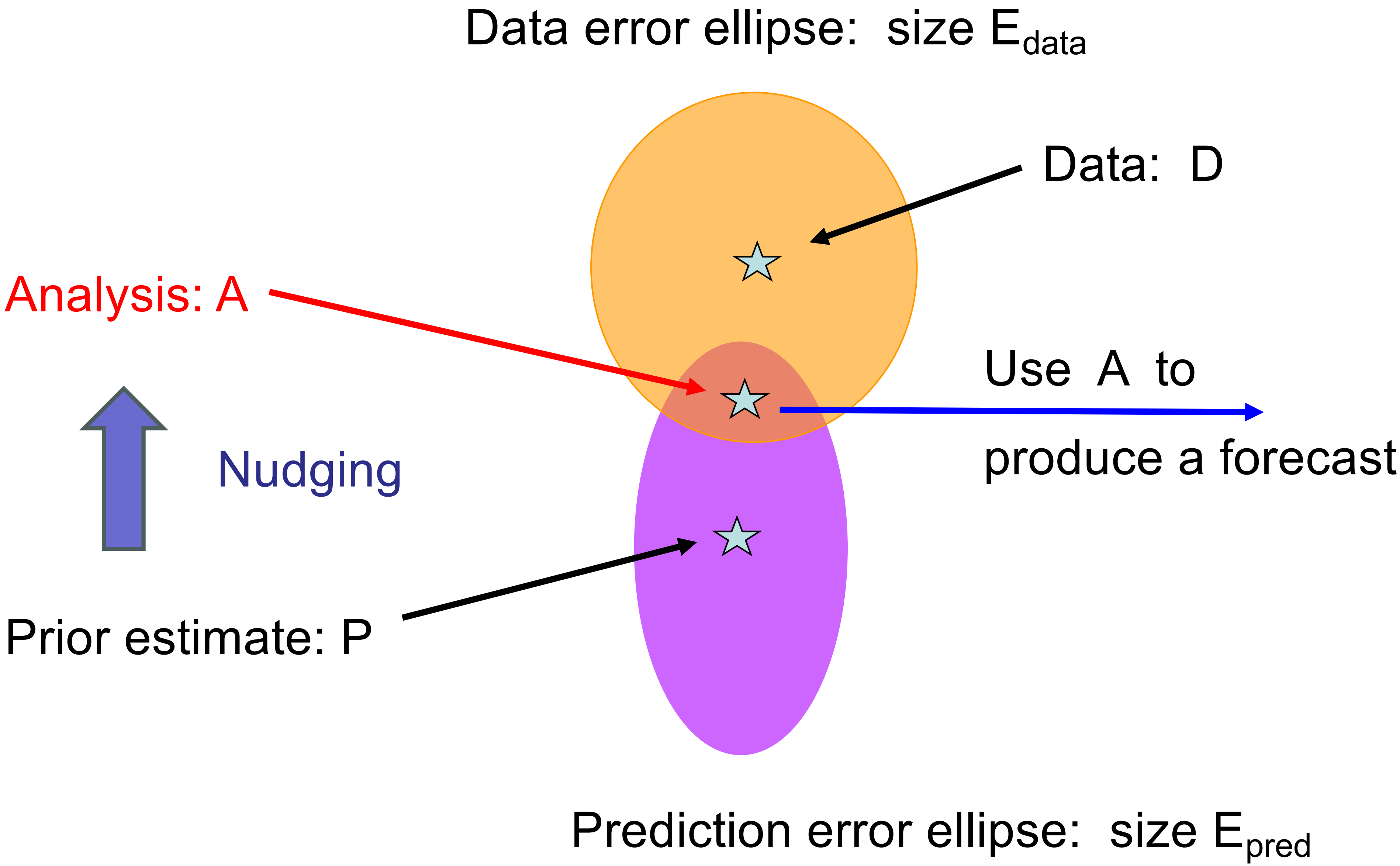


$H(x_t)$ of the true state

Both the prediction and the data have errors


Climatology
Initial condition
uncertainty
Forecast
uncertainty
Analysis
Deterministic
forecast
time

Data assimilation optimally estimates the system state which is consistent with both the prediction and the data and estimates the resulting error



Data error ellipse: size E_{data}
Data: D
Analysis: A
Use A to
produce a forecast
Nudging
Prior estimate: P
Prediction error ellipse: size E_{pred}

Data error: Gaussian, Covariance R

Background prediction error: Gaussian, Covariance B

Maximum likelihood of data y given truth x is

$$M = P(x|y)/P(x) = e^{-J(x)}$$

$$J(x_a) = \frac{1}{2}\left(x_a - x_b\right)^T B^{-1}\left(x_a - x_b\right) + \frac{1}{2}(Hx_a - y)^T R^{-1}(Hx_a - y)$$

BLUE: Find x_a which maximises M or minimizes J

Data sets you can play with created using this approach

Reanalysis:

UKCP18. (Climate, based on HadGEM3)

ERA5 (1979-2017 Weather predictions based on HRES)

All created using the previous methods ie. highly accurate PDE solver (FEM/Finite Volume/Spectral) plus data assimilation (pre and post the forecast where possible)

Historical Data: **HadUK-Grid**

UK Spatial resolution: 1km x 1km

Time span: 1862 to present

Temporal resolution: daily, monthly, seasonal and annual timescales, as well as long term averages for a set of climatological reference periods

Variables: air temperature (maximum, minimum and mean), precipitation, sunshine, mean sea level pressure, wind speed, relative humidity, vapour pressure, days of snow lying, and days of ground frost

ECMWF Reanalysis v5 (ERA5)

iew Datasets search

ERA5 is the fifth generation ECMWF atmospheric reanalysis of the global climate covering the period from January 1940 to present. ERA5 is produced by the Copernicus Climate Change Service (C3S) at ECMWF.

ERA5 provides hourly estimates of a large number of atmospheric, land and oceanic climate variables. The data cover the Earth on a 31km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80km. ERA5 includes information about uncertainties for all variables at reduced spatial and temporal resolutions.

ERA5 is available on:

- Single levels
- Pressure levels: 1000/975/950/925/900/875/850/825/800/775/750/700/650/600/550/500/450/400/350/300/250/225/200/175/150/125/100/70/50/30/20/10/7/5/3/2/1
- Potential temperature levels:

Explore this dataset:

Climate Data Store ›

Access to the data portals and their features depends on who you are

View licence

UKCP18
National Climate Projections

Jason A. Lowe, Dan Bernie, Philip Bett, Lucy Bricheno, Simon Brown, Daley Calvert, Robin Clark, Karen Eagle, Tamsin Edwards, Giorgia Fosser, Fai Fung, Laila Gohar, Peter Good, Jonathan Gregory, Glen Harris, Tom Howard, Neil Kaye, Elizabeth Kendon, Justin Krijnen, Paul Maisey, Ruth McDonald, Rachel McInnes, Carol McSweeney, John F.B. Mitchell, James Murphy, Matthew Palmer, Chris Roberts, Jon Rostron, David Sexton, Hazel Thornton, Jon Tinker, Simon Tucker, Kuniko Yamazaki, and Stephen Belcher.

Department for Environment Food & Rural Affairs
Department for Business, Energy & Industrial Strategy
Met Office Hadley Centre
Environment Agency

Working together on
UK Climate Projections

www.metoffice.gov.uk

Probable Scenarios

Effort to curb emissions	Energy generation	New technology	Transport		Temperature 2081–2100 (average increase relative to 1986-2005)	Sea level 2081–2100 (average rise relative to 1986-2005)	Extreme weather 2081–2100	Adaptation required
Low	Coal-fired power		Cars, trucks	RCP 8.5	3.7 °C	0.63 m	Large increase	High level at high cost
Medium	Mix		Mix	RCP 6.0	2.2 °C	0.48 m	Moderate increase	Medium level at medium cost
Medium	Renewable		Mix	RCP 4.5	1.8 °C	0.47 m	Moderate increase	Medium level at medium cost
High	Renewable	Emissions capture	Bicycles, public transport	RCP 2.6	1.0 °C	0.4 m	Small increase	Low level at low cost

Big claim papers which make use of these data sets

GraphCast (2023)

Aardvark (2025)

Methodology: GNN/Neural operator based methods

Trained on ERA5 data and satellite data

Very rapid prediction times, but expensive to train nature

View all journals | Search | Log in

Explore content | About the journal | Publish with us | Subscribe | Sign up for alerts | RSS feed

nature > news > article

NEWS | 14 November 2023

DeepMind AI accurately forecasts weather — on a desktop computer

The machine-learning model takes less than a minute to predict future weather worldwide more precisely than other approaches.

By Carissa Wong

You have full access to this article via **JISC Springer Compact (publishing + reading)**.

WEATHER FORECASTING

Learning skillful medium-range global weather forecasting

Remi Lam[1]*†, Alvaro Sanchez-Gonzalez[1]*†, Matthew Willson[1]*†, Peter Wirnsberger[1]†, Meire Fortunato[1]†, Ferran Alet[1]†, Suman Ravuri[1]†, Timo Ewalds[1], Zach Eaton-Rosen[1], Weihua Hu[1], Alexander Merose[2], Stephan Hoyer[2], George Holland[1], Oriol Vinyals[1], Jacklynn Stott[1], Alexander Pritzel[1], Shakir Mohamed[1]*, Peter Battaglia[1]*

Global medium-range weather forecasting is critical to decision-making across many social and economic domains. Traditional numerical weather prediction uses increased compute resources to improve forecast accuracy but does not directly use historical weather data to improve the underlying model. Here, we introduce GraphCast, a machine learning–based method trained directly from reanalysis data. It predicts hundreds of weather variables for the next 10 days at 0.25° resolution globally in under 1 minute. GraphCast significantly outperforms the most accurate operational deterministic systems on 90% of 1380 verification targets, and its forecasts support better severe event prediction, including tropical cyclone tracking, atmospheric rivers, and extreme temperatures. GraphCast is a key advance in accurate and efficient weather forecasting and helps realize the promise of machine learning for modeling complex dynamical systems.

It is 05:45 UTC (coordinated universal time) in mid-October 2022 in Bologna, Italy, at the recently opened high-performance computing facility of the European Centre for Medium-Range Weather Forecasts (ECMWF). For the past several hours, the Integrated Forecasting System (IFS) has been running sophisticated calculations to forecast Earth's weather over the next days and weeks, and its first predictions have just begun to be disseminated to users. This process repeats every 6 hours, every day, to supply the world with the most accurate weather forecasts available.

The IFS, and modern weather forecasting more generally, are triumphs of science and

days ahead—a possibility that was unthinkable even a few decades ago.

But while traditional NWP scales well with compute, capitalizing on the vast amount of historical weather data to improve accuracy is not straightforward. Rather, NWP methods are improved by highly trained experts innovating better models, algorithms, and approximations, which can be a time-consuming and costly process.

Machine learning–based weather prediction (MLWP)—wherein forecast models are trained from historical data, including observations and analysis data—offers an alternative to traditional NWP. MLWP has the potential to im-

tively weak, for example, in subseasonal heat wave prediction (*3*) and precipitation nowcasting from radar images (*4*–*7*), where accurate equations and robust numerical methods are not as available.

In medium-range weather forecasting—the prediction of atmospheric variables up to 10 days ahead—NWP-based systems such as the IFS are still most accurate. The top deterministic operational system in the world is ECMWF's high-resolution forecast (HRES), a configuration of IFS that produces global 10-day forecasts at 0.1° latitude and longitude resolution, in around an hour (*8*). However, over the past several years, MLWP methods for medium-range forecasting trained on reanalysis data have been steadily advancing, facilitated by benchmarks such as WeatherBench (*8*). Deep learning architectures based on convolutional neural networks (*9*–*11*) and Transformers (*12*) have shown promising results at latitude and longitude resolutions coarser than 1.0°, and recent works—which use graph neural networks (GNNs), Fourier neural operators, and Transformers (*13*–*16*)—have reported performance that begins to rival IFS's at 1.0° and 0.25° for a handful of variables and lead times up to 7 days.

GraphCast

Here, we introduce an MLWP approach for global medium-range weather forecasting called GraphCast, which produces an accurate 10-day forecast in under a minute on a single Google Cloud TPU (Tensor Processing Unit) v4 device and supports applications including predicting tropical cyclone tracks, atmospheric rivers, and extreme temperatures.

GraphCast takes as input the two most re-

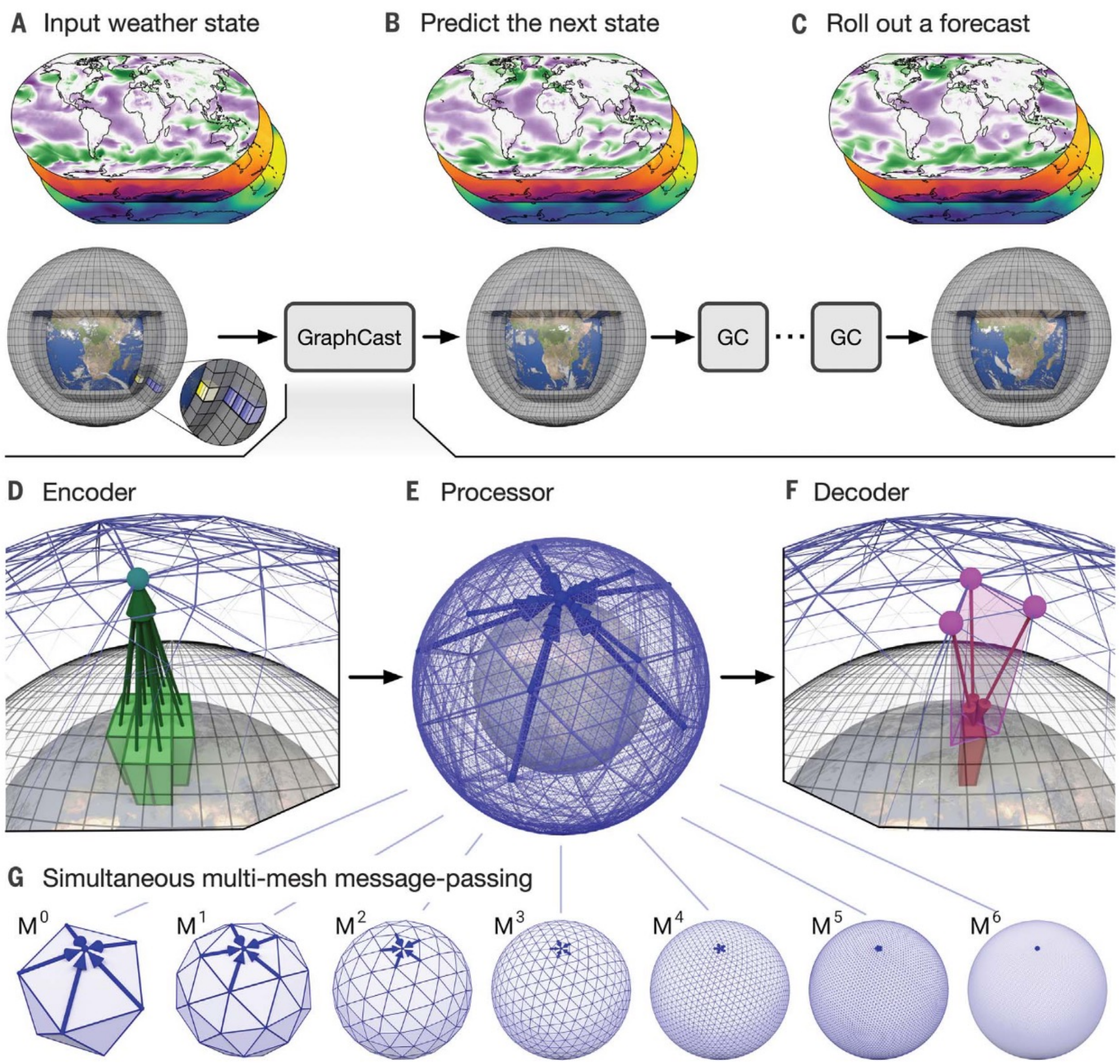


Fig. 1. Model schematic. (**A**) The input weather state(s) are defined on a 0.25° latitude-longitude grid comprising a total of 721 × 1440 = 1,038,240 points. Yellow layers in the close-up pop-out window represent the five surface variables, and blue layers represent the six atmospheric variables that are repeated at 37 pressure levels (5 + 6 × 37 = 227 variables per point in total), resulting in a state representation of 235,680,480 values. (**B**) GraphCast predicts the next state of the weather on the grid. (**C**) A forecast is made by iteratively applying GraphCast (GC) to each previous predicted state, to produce a sequence of states that represent the weather at successive lead times. (**D**) The encoder component of the GraphCast architecture maps local regions of the input (green boxes) into nodes of the multimesh graph representation (green, upward arrows that terminate in the green-blue node). (**E**) The processor component updates each multimesh node using learned message-passing (heavy blue arrows that terminate at a node). (**F**) The decoder component maps the processed multimesh features (purple nodes) back onto the grid representation (red, downward arrows that terminate

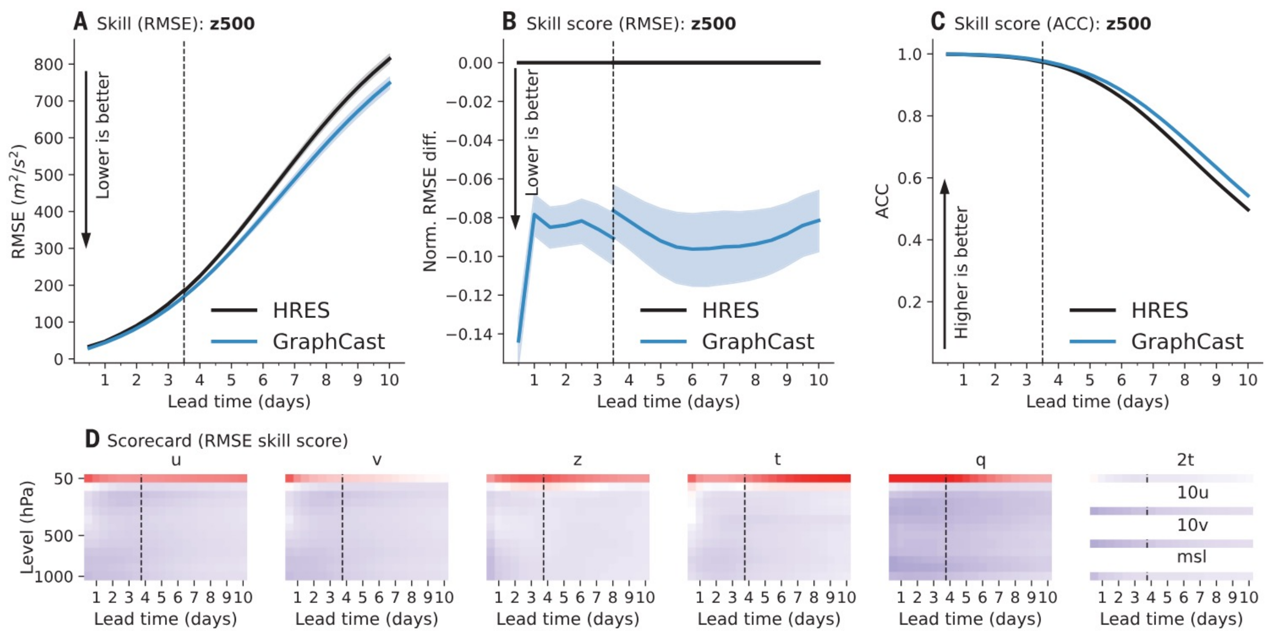


Fig. 2. Global skill and skill scores for GraphCast and HRES in 2018. (**A**) RMSE skill (*y* axis) for GraphCast (blue lines) and HRES (black lines), on *Z*500, as a function of lead time (*x* axis). Error bars represent 95% confidence intervals. The vertical dashed line represents 3.5 days, which is the last 12-hour increment of the HRES 06z/18z forecasts. The black line represents HRES, where lead times earlier and later than 3.5 days are from the 06z/18z and 00z/12z initializations, respectively. (**B**) RMSE skill score (*y* axis) for GraphCast versus HRES, on *Z*500, as a function of lead time (*x* axis). Error bars represent 95% confidence intervals for the skill score. We observe a discontinuity in GraphCast's curve because skill scores up to 3.5 days are computed between GraphCast (initialized at 06z/18z) and HRES's 06z/18z initialization, whereas skill scores after 3.5 days are computed with respect to HRES's 00z/12z initializations. (**C**) ACC skill (*y* axis) for GraphCast (blue lines) and HRES (black lines), on *Z*500, as a function of lead time (*x* axis). (**D**) Scorecard of RMSE skill scores for GraphCast, with respect to HRES. Each subplot corresponds to one variable: *U*, *V*, *Z*, *T*, *Q*, *2T*, *10U*, *10V*, and *MSL*. The rows of each heatmap correspond to the 13 pressure levels (for the atmospheric variables), from 50 hPa at the top to 1000 hPa at the bottom. The columns of each heatmap correspond to the 20 lead times at 12-hour intervals, from 12 hours on the left to 10 days on the right. Each cell's color represents the skill score, as shown in (B), where blue represents negative values (GraphCast has better skill) and red represents positive values (HRES has better skill).

transition from 06z/18z to 00z/12z initializations for HRES induced a small discontinuity in our plots, which is indicated by a vertically dashed line at the appropriate lead time. Supplementary materials section 5 contains further verification details, including details of the comparisons protocol between GraphCast and HRES (supplementary materials section 5.2) and the effect of initialization lookahead on both models' performance (supplementary materials section 5.2.2).

Forecast verification results

We find that GraphCast has greater weather forecasting skill than HRES when evaluated on 10-day forecasts at a horizontal resolution of 0.25° for latitude and longitude and at 13 vertical levels. Figure 2, A to C, shows how GraphCast (blue lines) outperforms HRES (black lines) on the *Z*500 (geopotential at 500 hPa) "headline" field in terms of RMSE skill, RMSE skill score [i.e., the normalized RMSE difference between model *A* and baseline *B* defined as $(RMSE_A - RMSE_B)/(RMSE_B)$], and ACC skill. Using *Z*500, which encodes the synoptic-scale pressure distribution, is common in the literature, as it has strong meteorological importance (*8*). The plots show that GraphCast has better skill scores across all lead times, with a skill score improvement of around 7 to 14%. Plots for additional headline variables are given in supplementary materials section 7.1.

Figure 2D summarizes the RMSE skill scores for all 1380 evaluated variables and pressure levels, across the 10-day forecasts, in a format analogous to the ECMWF Scorecard. The cell colors are proportional to the skill score, where blue indicates that GraphCast had better skill, and red indicates that HRES had better skill. GraphCast outperformed HRES on 90.3% of the

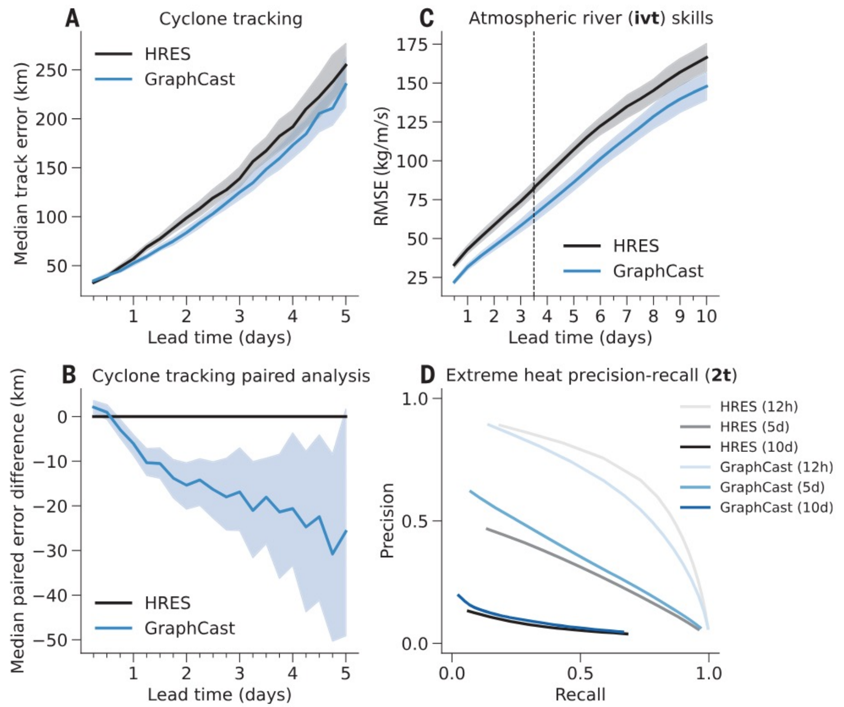


Fig. 3. Severe event prediction. (**A**) Cyclone tracking performances for GraphCast and HRES. The *x* axis represents lead times (in days), and the *y* axis represents median track error (in kilometers). Error bars represent bootstrapped 95% confidence intervals for the median. (**B**) Cyclone tracking paired error difference between GraphCast and HRES. The *x* axis represents lead times (in days), and the *y* axis represents median paired error difference (in kilometers). Error bars represent bootstrapped 95% confidence intervals for the median difference (see supplementary materials section 8.1). (**C**) Atmospheric river prediction (*IVT*) skills for GraphCast and HRES. The *x* axis represents lead times (in days), and the *y* axis represents RMSE. Error bars are 95% confidence intervals. (**D**) Extreme heat prediction precision-recall for GraphCast and HRES. The *x* axis represents recall, and the *y* axis represents precision. The curves represent different precision-recall trade-offs when sweeping over gain applied to forecast signals (see supplementary materials section 8.3).

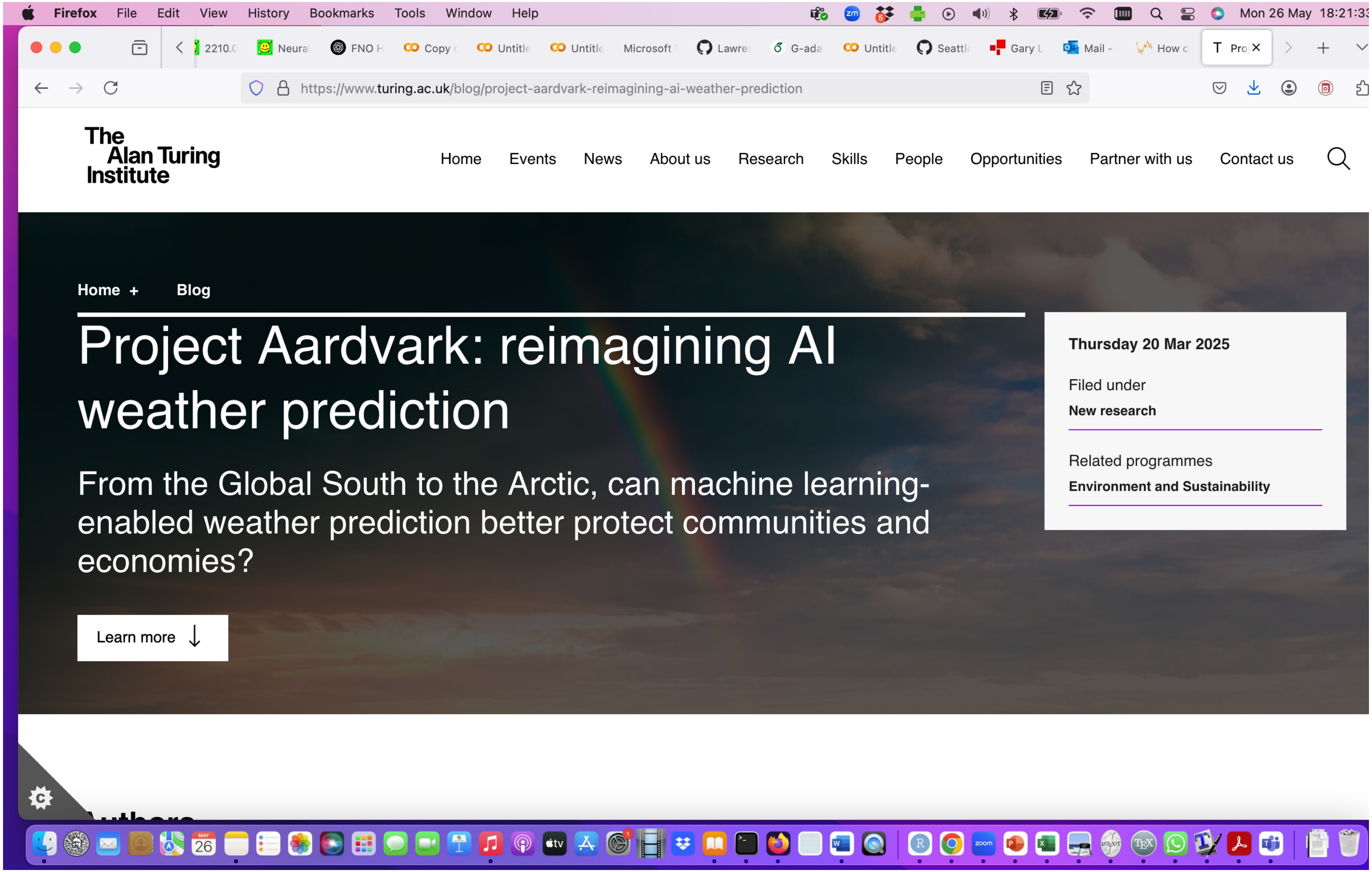
GraphCast's forecast skill and efficiency compared with HRES shows that MLWP methods are now competitive with traditional weather forecasting methods. Additionally, GraphCast's performance on severe event forecasting, which it was not directly trained for, demonstrates its robustness and potential for downstream value. We believe this marks a turning point in weather forecasting, which helps open new avenues to strengthen the breadth of weather-dependent decision-making by individuals and industries by making cheap prediction more accurate and accessible as well as suitable for specific applications. With 36.7 million parameters, GraphCast is a relatively small model by modern ML standards, chosen to keep the memory footprint tractable. And while HRES is released on 0.1 resolution, 137 levels, and up to 1-hour time steps, GraphCast operates on 0.25° latitude and longitude resolution, 37 vertical levels, and 6-hour time steps, because of the ERA5 training data's native 0.25° resolution and engineering challenges in fitting higher-resolution data on hardware. Generally, GraphCast should be viewed as a family of models, with the current version being the largest we can practically fit under current engineering constraints, but which have the potential to scale much further in the future with greater computer resources and higher-resolution data

Criticisms of Graph-Cast

- Heavily based on a very skillfully created data set
- Makes predictions based on the data set, NOT on real (eg. satellite) data
- Suffers from spectral bias, so not resolving small scale phenomena well. Good for (smooth) pressure, not good for (localized) precipitation and fog.
- Norm used to compare with HRES is flattering to smoother predictions

Predicts the wrong weather at the right time, rather than the right weather at the wrong time

2025: Aardvark. Predictions from raw data only



Assessing opportunities and challenges

Aardvark reimagines current weather prediction methods, offering a range of opportunities.

Alongside requiring less computing power, Aardvark is fast. Traditional forecasts can take hours to produce on a supercomputer whereas, once trained, Aardvark can create forecasts within minutes and can be run on a desktop computer.

There are highly promising signs of Aardvark's accuracy too. Globally, Aardvark is already as accurate as America's Global Forecast System (GFS), but it is only using about 10% of the available data to make its forecasts, meaning that further improvements in accuracy should be possible. We are excited to see what happens as we increase the amount of data and optimise Aardvark end-to-end to provide more accurate forecasts. This new paradigm could replace the traditional numerical approach in developing countries.

A streamlined system like this could also play a significant role in democratising access to advanced forecasting tools, empowering developing or data-sparse countries to build capacity and create bespoke weather forecasting systems that previously would have required large teams to operate, deploy and maintain.

There are of course challenges and it's important to acknowledge that, whilst machine learning weather tools are moving at great pace, this is still an experimental technology that will require rigorous evaluation over a period of time.

Weather prediction tools must accurately predict all types of weather, and extremes like hurricanes and floods are especially important. Unfortunately, rare events like these are less represented in the training data, meaning that AI systems may struggle more on these phenomena.

We also need to ensure we account for our changing climate, which could render models trained on past data less accurate.

Early signs suggest that we can rise to these challenges.

What's next for Aardvark?

Article

End-to-end data-driven weather prediction

https://doi.org/10.1038/s41586-025-08897-0

Received: 10 July 2024

Accepted: 12 March 2025

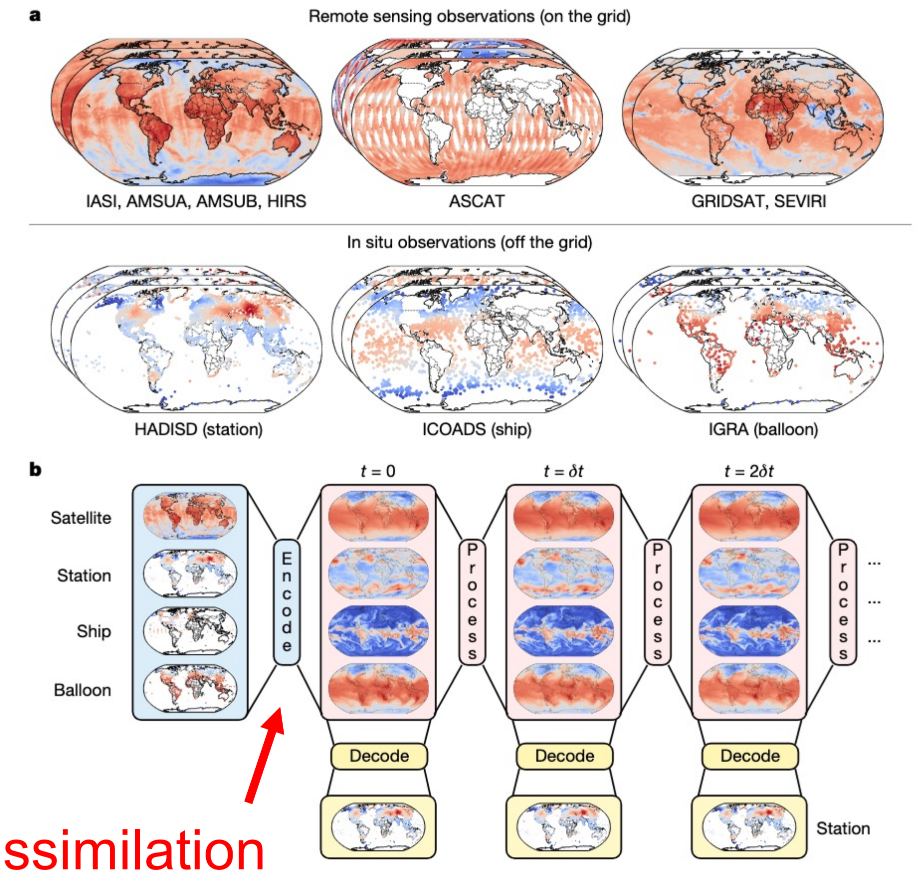
Published online: 20 March 2025

Open access

Check for updates

Anna Allen[1,11] ✉, Stratis Markou[2,11] ✉, Will Tebbutt[2,9], James Requeima[3], Wessel P. Bruinsma[4], Tom R. Andersson[5,10], Michael Herzog[6], Nicholas D. Lane[1], Matthew Chantry[7], J. Scott Hosking[5,8] & Richard E. Turner[2,8] ✉

Weather prediction is critical for a range of human activities, including transportation, agriculture and industry, as well as for the safety of the general public. Machine learning transforms numerical weather prediction (NWP) by replacing the numerical solver with neural networks, improving the speed and accuracy of the forecasting component of the prediction pipeline[1–6]. However, current models rely on numerical systems at initialization and to produce local forecasts, thereby limiting their achievable gains. Here we show that a single machine learning model can replace the entire NWP pipeline. Aardvark Weather, an end-to-end data-driven weather prediction system, ingests observations and produces global gridded forecasts and local station forecasts. The global forecasts outperform an operational NWP baseline for several variables and lead times. The local station forecasts are skilful for up to ten days of lead time, competing with a post-processed global NWP baseline and a state-of-the-art end-to-end forecasting system with input from human forecasters. End-to-end tuning further improves the accuracy of local forecasts. Our results show that skilful forecasting is possible without relying on NWP at deployment time, which will enable the realization of the full speed and accuracy benefits of data-driven models. We believe that Aardvark Weather will be the starting point for a new generation of end-to-end models that will reduce computational costs by orders of magnitude and enable the rapid, affordable creation of customized models for a range of end users.



Data assimilation

Fig. 1 | Data and operation of Aardvark Weather. a, Different data sources leveraged in Aardvark. The input data consist of observations from remote sensing instruments (top row), which we pre-grid before passing to the model, as well as in situ observations from land and marine observation platforms and radiosondes (bottom row). Each of these data modalities contains several observational variables, of which we selected a subset here for the purposes of illustration. Here we show remote sensing data[40–45], after performing our gridding step, and raw in situ data[46–48]. Note that the colours in all six plots are meant for illustration purposes. The remote sensing data also include a range of metadata about the measurements, omitted here for simplicity. White areas indicate regions of missing data, which must be handled by the encoder module of Aardvark. **b**, Aardvark at deployment time. First, an encoder module uses raw observations as input to estimate the initial state of the atmosphere across key variables at $t = 0$. Next, a processor module ingests the estimated state to produce a forecast at the next lead time $t = \delta t$. Forecasts at subsequent lead times are produced autoregressively. Finally, a decoder module is applied to the on the grid states to produce off the grid predictions. The modular design of Aardvark allows for pretraining on large high-quality ERA5 reanalysis data[34]. In this figure, the displayed data are the training data used to train each module of Aardvark from the aforementioned sources.

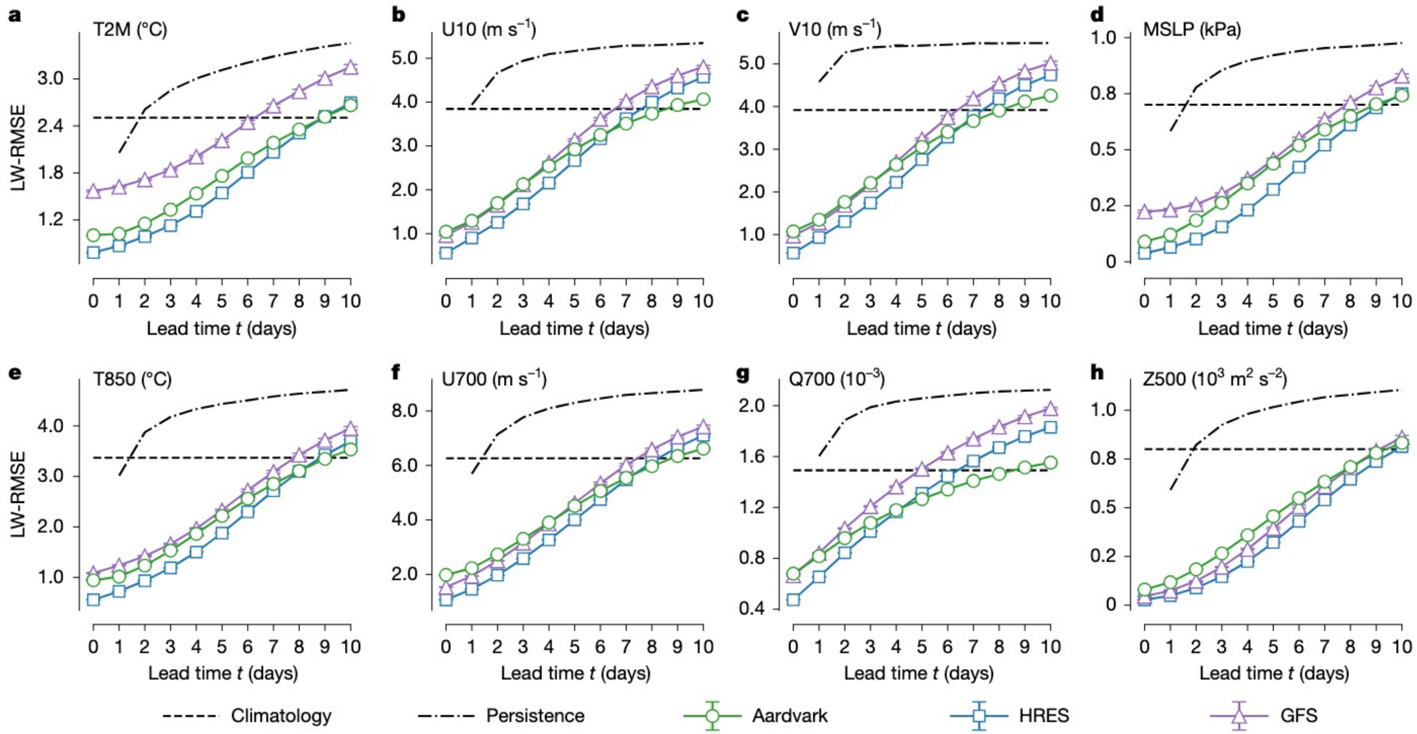


Fig. 2 | Gridded global forecast performance for selected variables. **a**–**h**, Latitude-weighted RMSE using ERA5 (ref. 34) reanalysis data as the ground truth, on the held-out test year (2018), for the four surface variables: 2-m temperature (**a**; T2M), 10-m eastward wind (**b**; U10), 10-m northward wind (**c**; V10) and mean sea level pressure (**d**; MSLP), as well as four headline upper-atmosphere variables: temperature at 850 hPa (**e**; T850), eastward wind at 700 hPa (**f**; U700), specific humidity at 700 hPa (**g**; Q700) and geopotential at 500 hPa (**h**; Z500) as a function of lead time *t*. At lead time $t = 0$, Aardvark predicted the initial atmospheric state from observational data alone. The error at $t = 0$ corresponds to the error in the initial state. Note that HRES has a non-zero error at $t = 0$ compared to ERA5 reanalysis. The HRES forecasts[33] we used have been conservatively re-gridded to prevent aliasing, and we performed the same operation on the GFS forecasts[49]. We report the mean performance of each system together with 98% confidence intervals in our estimate of the mean performance.

use a recurrent update in which the previous forecast is adjusted in light of new observations, similar to Kalman filter recursions in a Markov model. In principle, data assimilation accumulates information from observations across all past time steps. However, in practice, it has been estimated that the effective window size is as short as 4 days (ref. 25). Owing to the complexities of training recurrent neural networks, including the need for a spin-up period and gradient instabilities[26], we opted a way that mimics how it will be deployed. We started by pretraining the encoder module using raw observations as input and reanalysis data as targets. An advantage of this machine learning approach is that the model can learn to correct for biases in the input observations during training; therefore, no bias correction step was performed on the input data. We also pretrained the processor using reanalysis data for both inputs and targets and then fine-tuned the output of the

ML Parametrisations

Augment a NWP forecast by using ML to give sub-grid scale level physics

Example: CARAMEL Project: Cloud fraction prediction

Scale-Aware Parameterization of Cloud Fraction and Condensate for a Global Atmospheric Model Machine-Learned From Coarse-Grained Kilometer-Scale Simulations

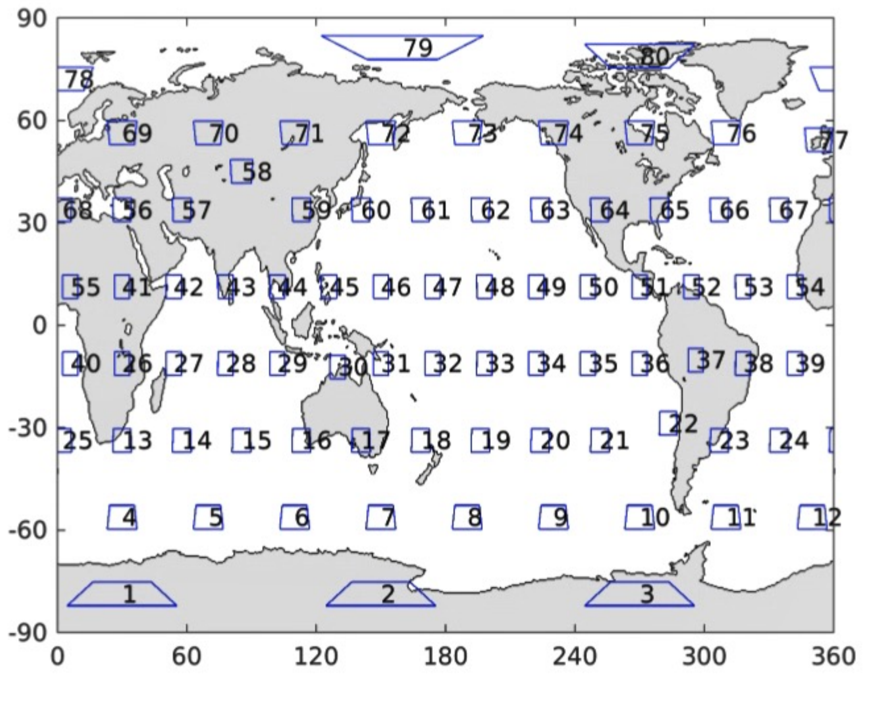
Cyril Morcrette, Tobias Cave, Helena Reid, Joana da Silva Rodrigues, Teo Deveney, Lisa Kreusser, Kwinten Van Weverberg, and B

Data:

Target: Measured Cloud fraction observations

Input: Computed NWP coarse predictions of temperature, pressure, wind speed, moisture content etc.

Output of NN: Estimates of cloud fractions

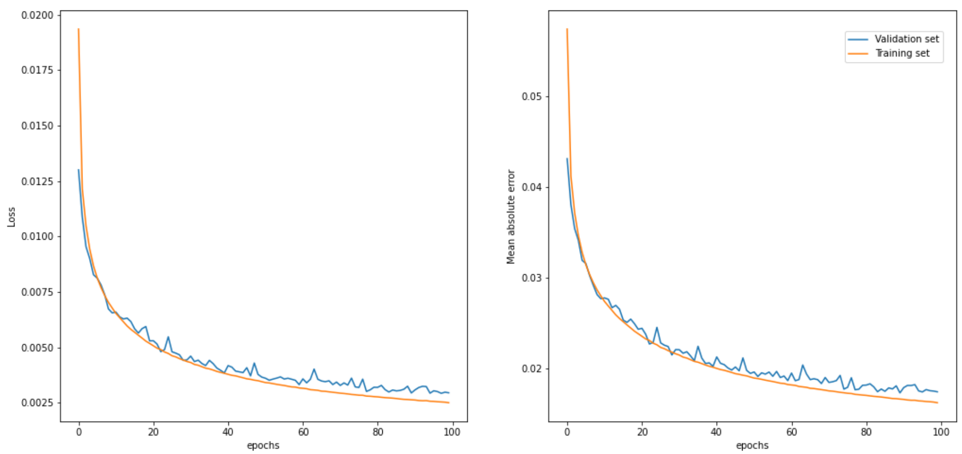


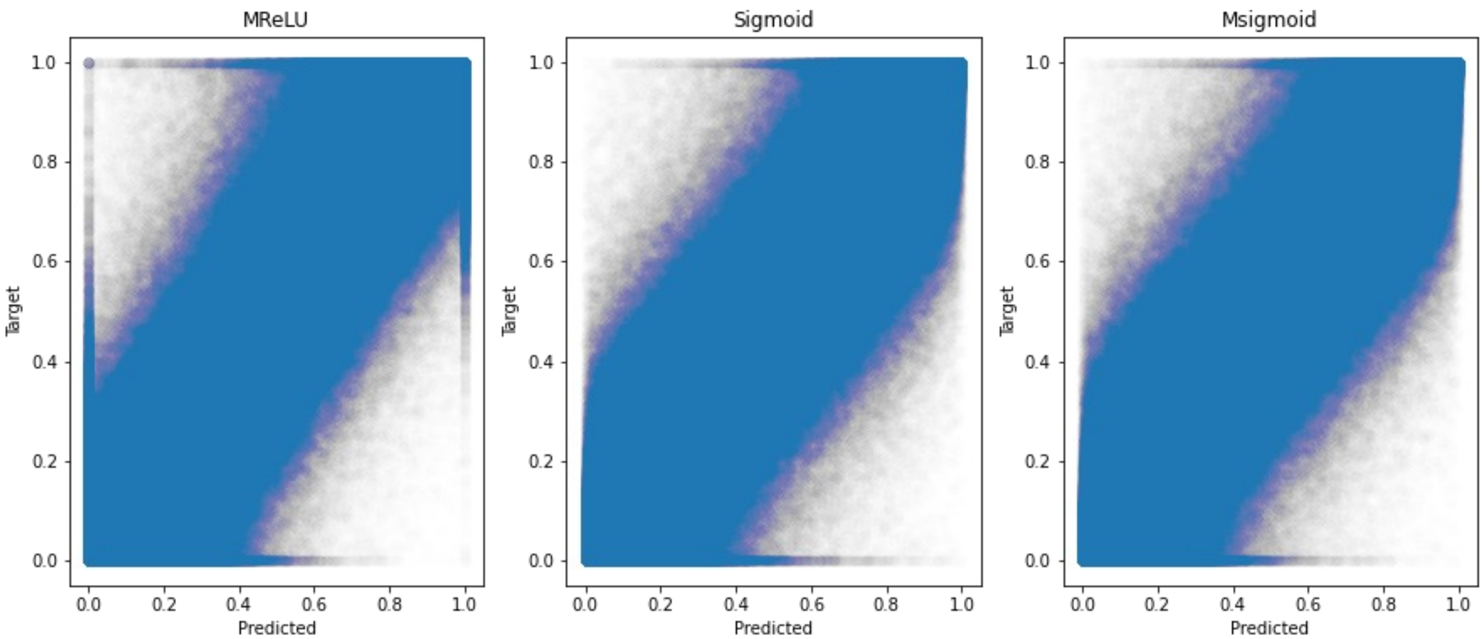
Architecture chosen was a fully connected ANN

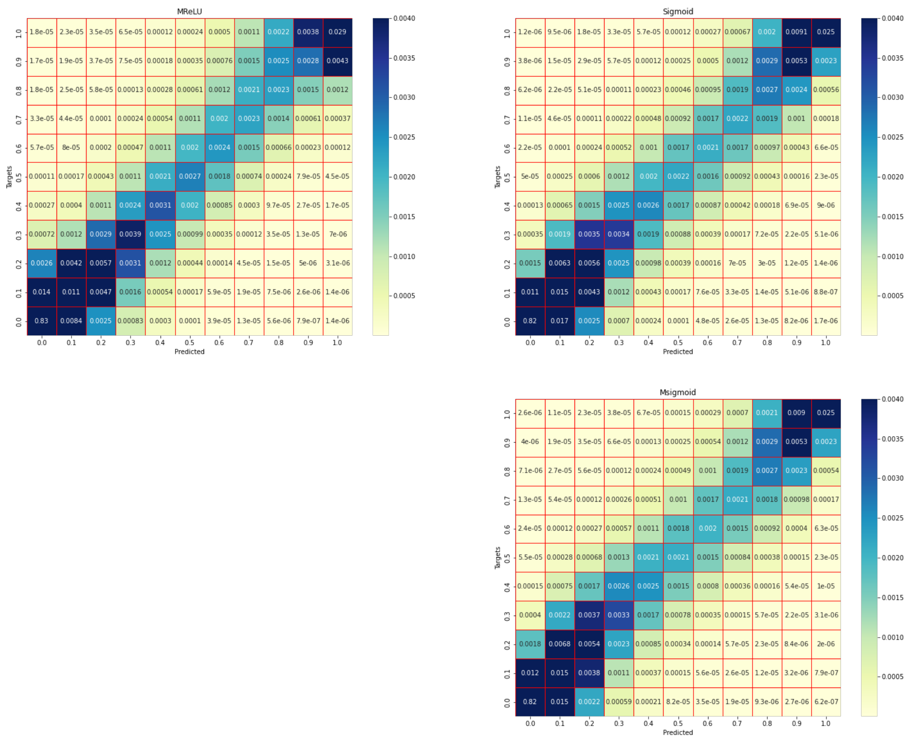
Activation functions of ReLU for all layers apart from the output layer which used sigmoid as the target values are between 0 and 1.

The loss function used was mean squared error.

The optimiser was Adam with a learning rate of 0.0001.




MReLU
Sigmoid
Msigmoid
Target
Predicted


MReLU
Sigmoid
Msigmoid
Targets
Predicted

Downscaling:

Generating High-Resolution Precipitation Data from Low-Resolution Data using generative diffusion models

B, Libo Chen, Lisa Kreusser and Teo Deveney, Bath and Ben Booth, Met Office

The Problem

- Objective: Use temporally-sequenced low-resolution simulations (GCM) to generat **temporally-sequenced high-resolution precipitation** fields.

- Motivation: High-resolution climate simulations are **computationally expensive**,

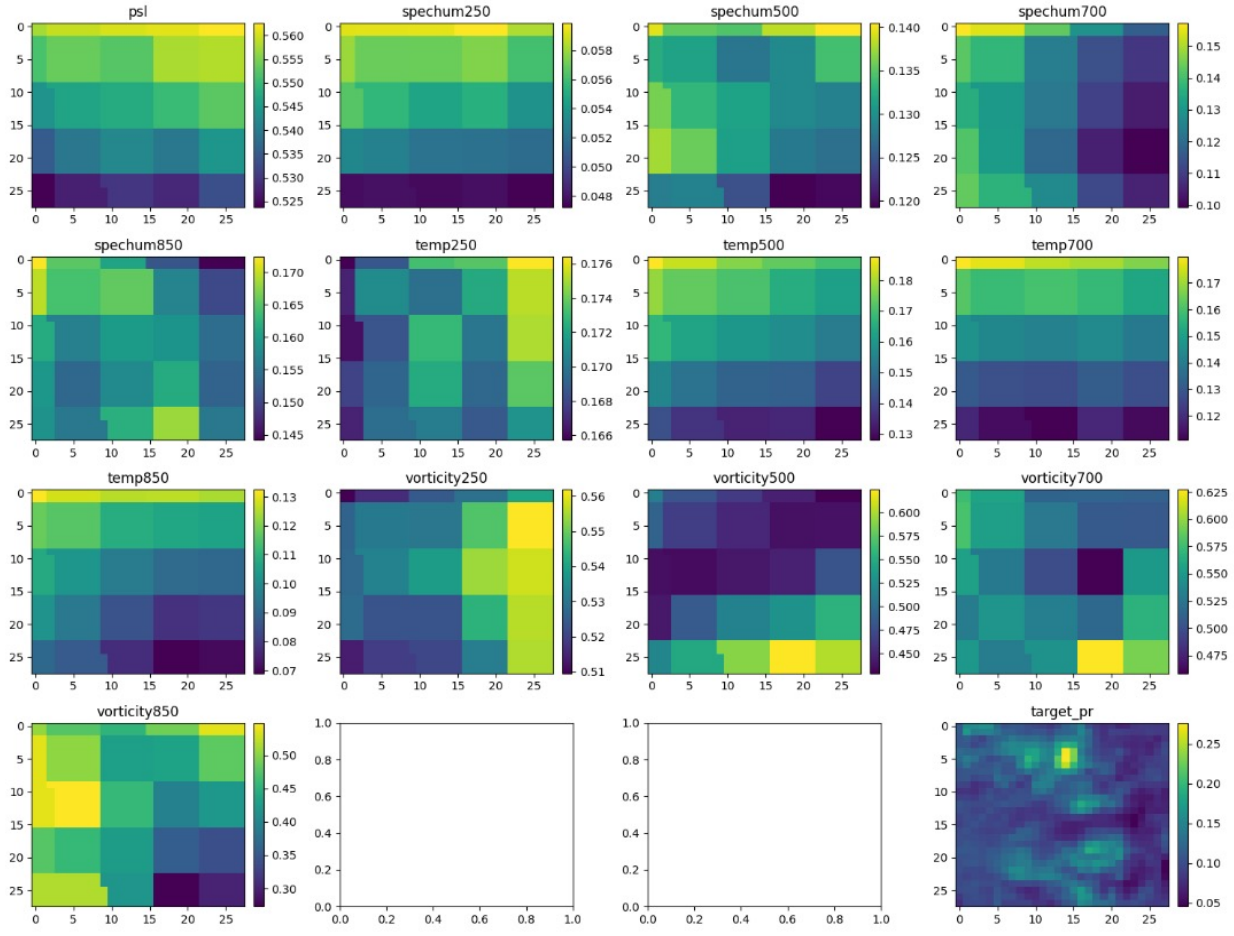
- Diffusion model design for **static data(2D)**

- Data:

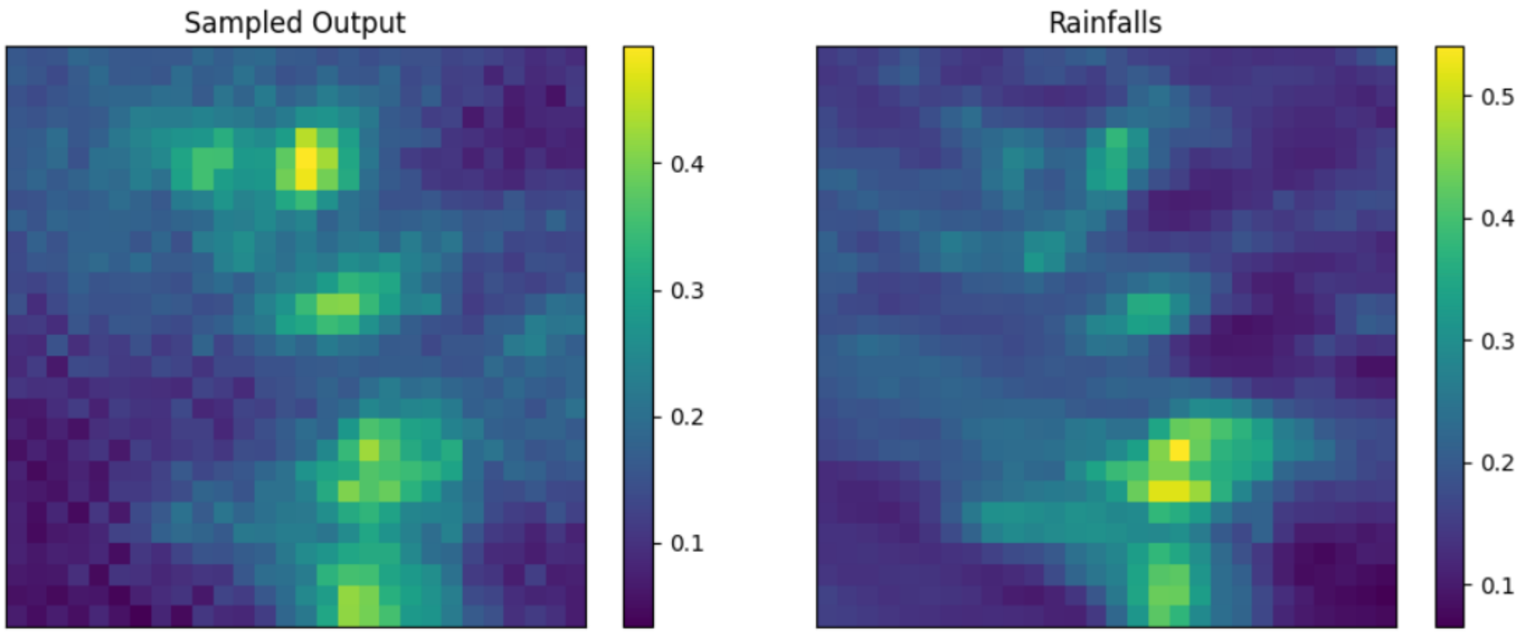
    - $x_t \in \mathbb{R}^{H \times W}$: high-resolution precipitation field 16.6km

    - $y_t \in \mathbb{R}^{C_y \times H \times W}$: GCM outputs 60km, and coarse precipitation 8.8*7 km

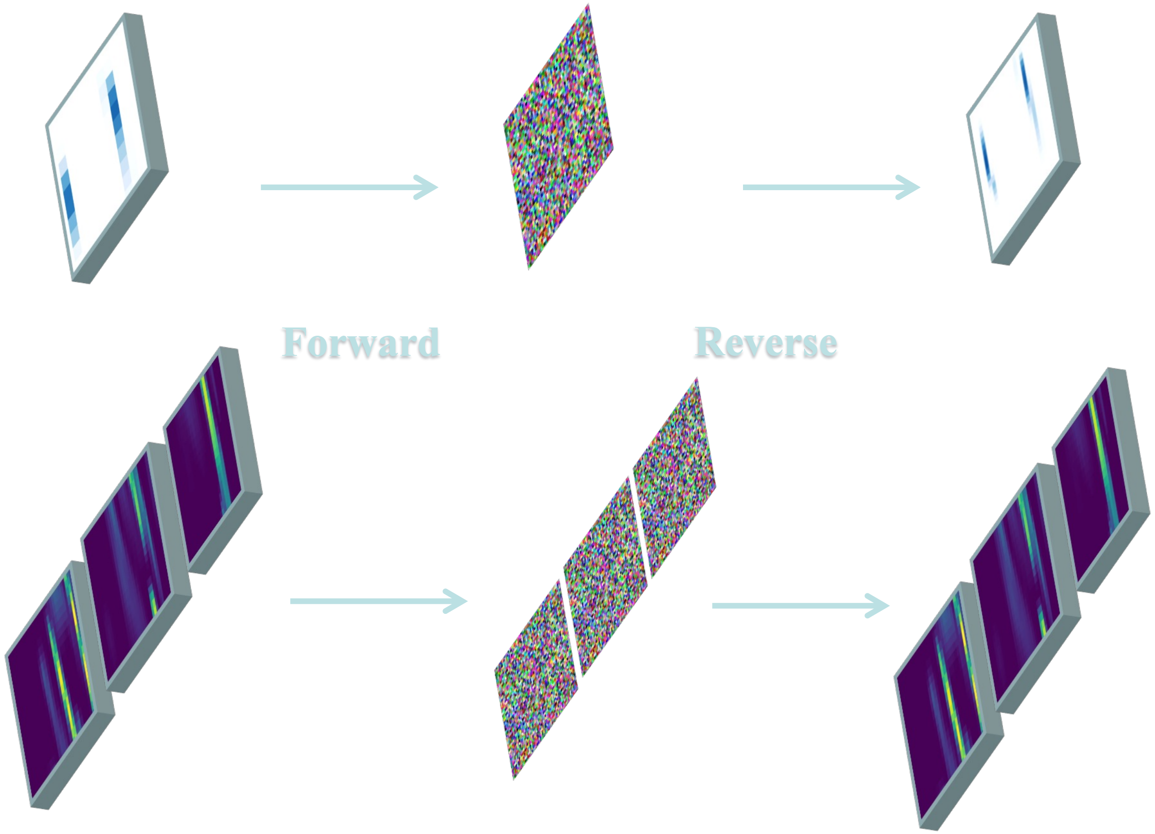
Spatial Distribution of 13 Metorological Variables and Target Precipitation



Result:



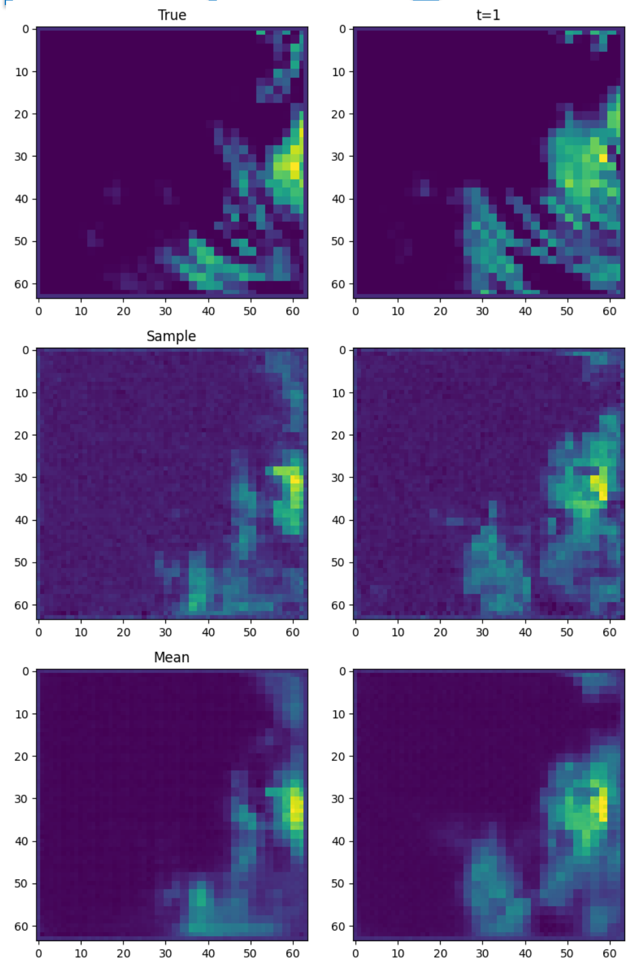
Adding Temporal Information: Diffusion process



$(N, 1, H, W)$

(N', T, H, W)

Temporal Upscaling from Coarse-Resolution Precipitation

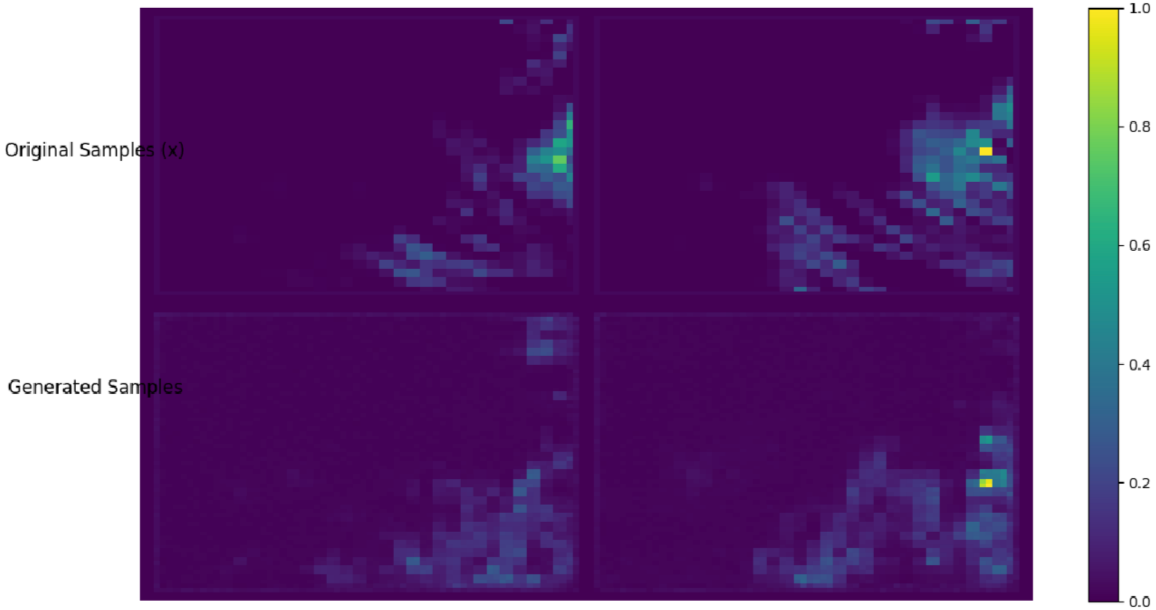


- Input: $x_{low,t-1}, x_{low,t}$

- Output: $x_{high,t-1}, x_{high,t}$

- Model Type: Autoregressive baseline

Physics-informed approach (meteorological variables)

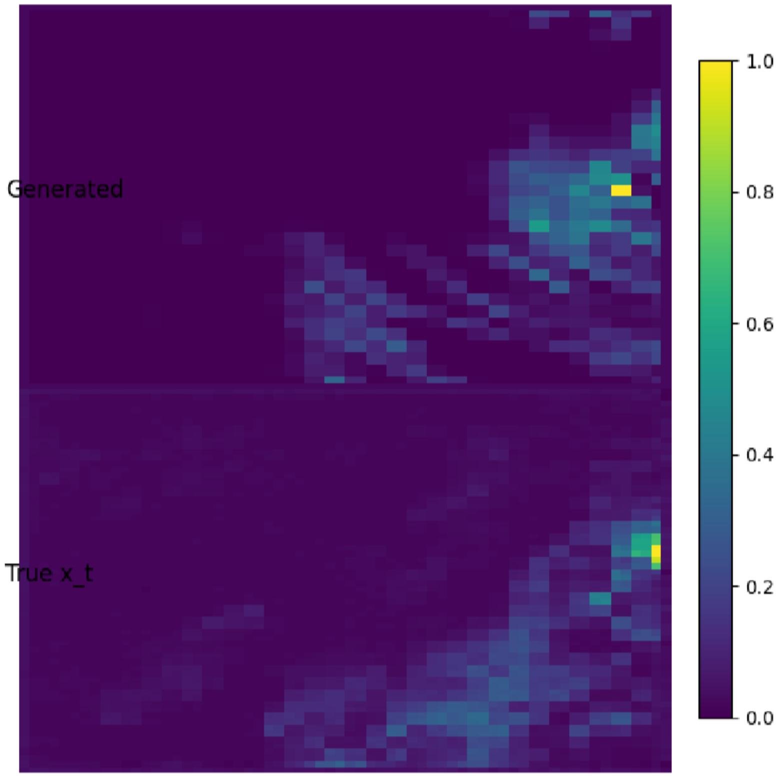


$$y_{low,t-1}, y_{low,t}$$

$$x_{high,t-1}, x_{high,t}$$

- Model Type: Physics-informed approach

Historical precipitation plus meteorological variables



Input: $y_{low,t-1}, y_{low,t}, x_{high,t-1}$

Output: $x_{high,t}$

Model Type: Extended autoregressive approach