Koninklijk Meteorologisch Instituut

Institut Royal Météorologique

Königliche Meteorologische Institut

Royal Meteorological Institute

# EDIPI predictability training with qgs

Jonathan Demaeyer & Francesco Ragone

Royal Meteorological Institute of Belgium

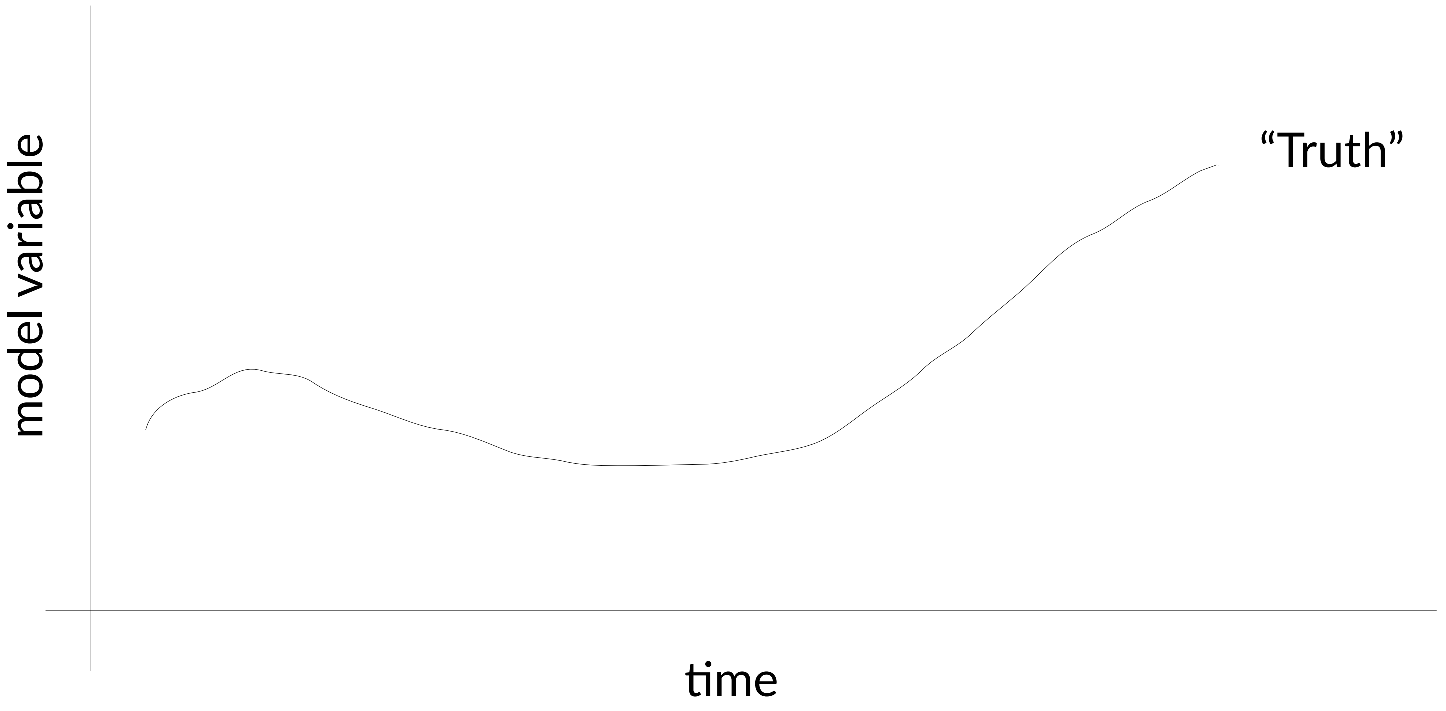
16-20 January 2023

# Outline

- Introduction to forecasting
- Focus on predictability
- Introduction to qgs
- The goals of this training

# Forecasting chain

Weather forecasting is an initial value problem !

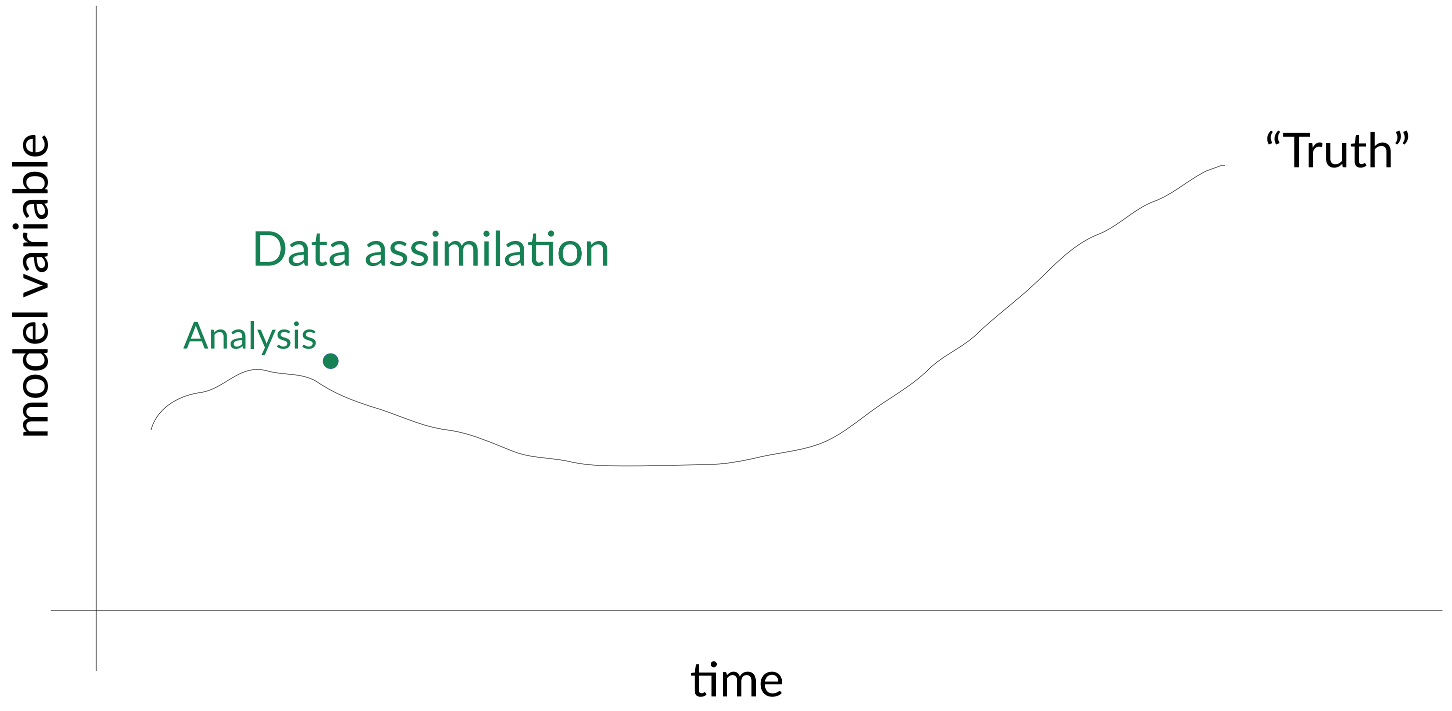


# Forecasting chain

Weather forecasting is an initial value problem !

**1. Data assimilation:**

Estimation of the best initial conditions for the model based on observations.



# Forecasting chain

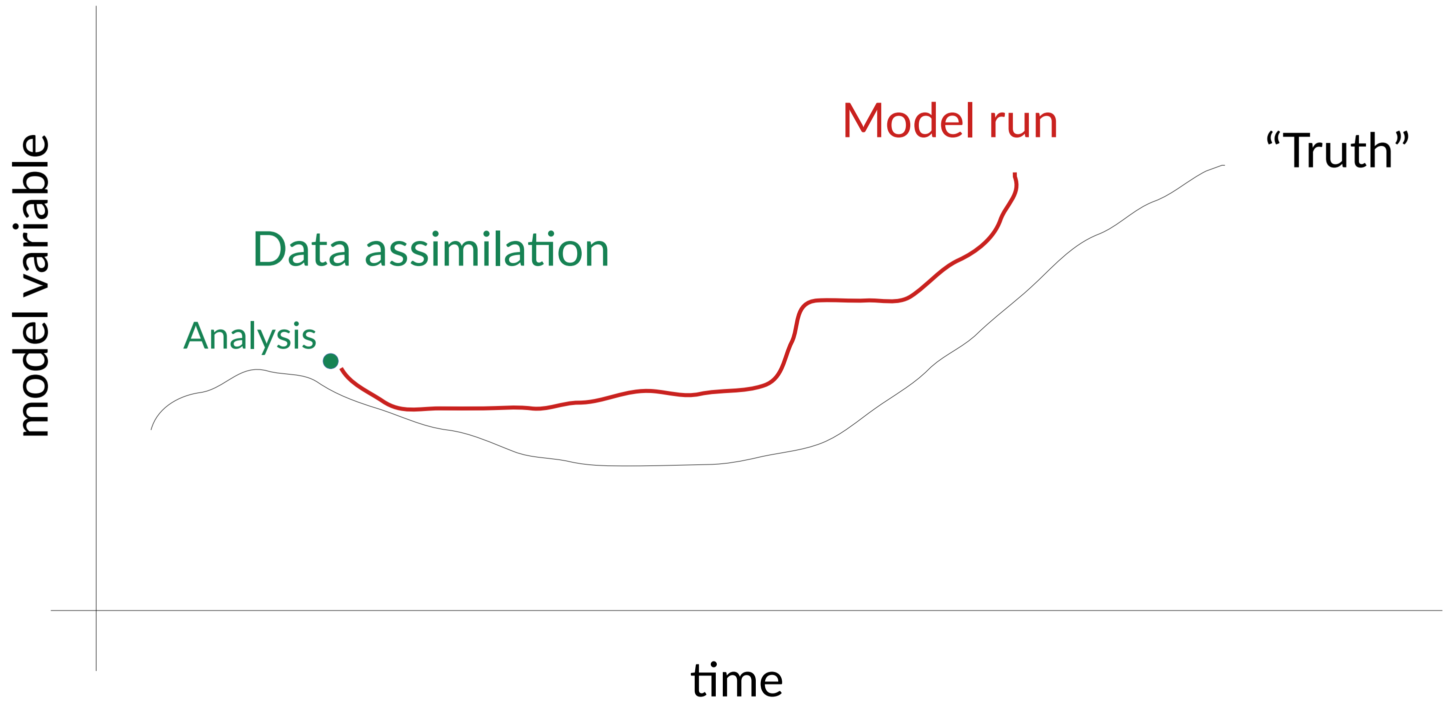
Weather forecasting is an initial value problem !

**1. Data assimilation:**

Estimation of the best initial conditions for the model based on observations.

**2. Model run:**

Computing the model time evolution.



# Forecasting chain

Weather forecasting is an initial value problem !

**1. Data assimilation:**

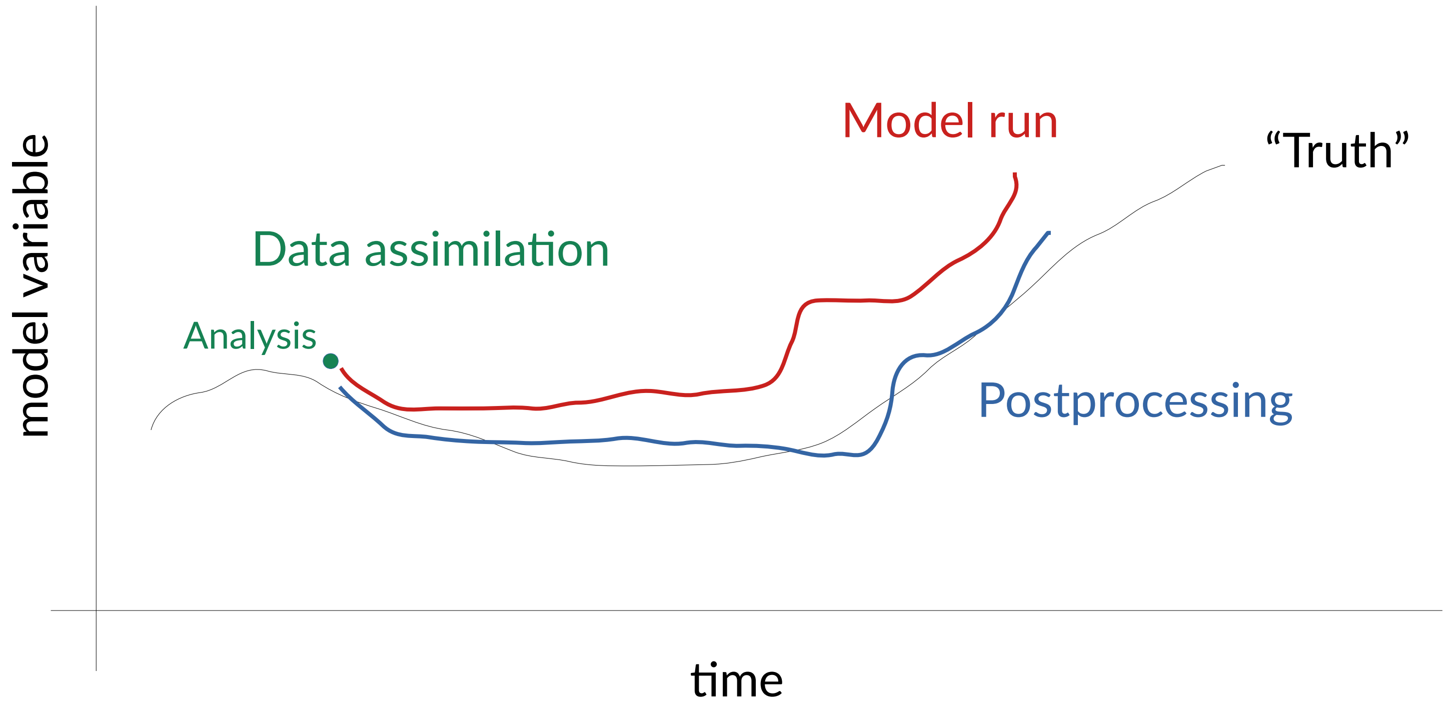
Estimation of the best initial conditions for the model based on observations.

**2. Model run:**

Computing the model time evolution.

**3. Postprocessing**

Bringing back the model output to reality: bias correction, spread, ...



# Forecasting chain is a cycle !

Weather forecasting is an initial value problem !

**1. Data assimilation:**

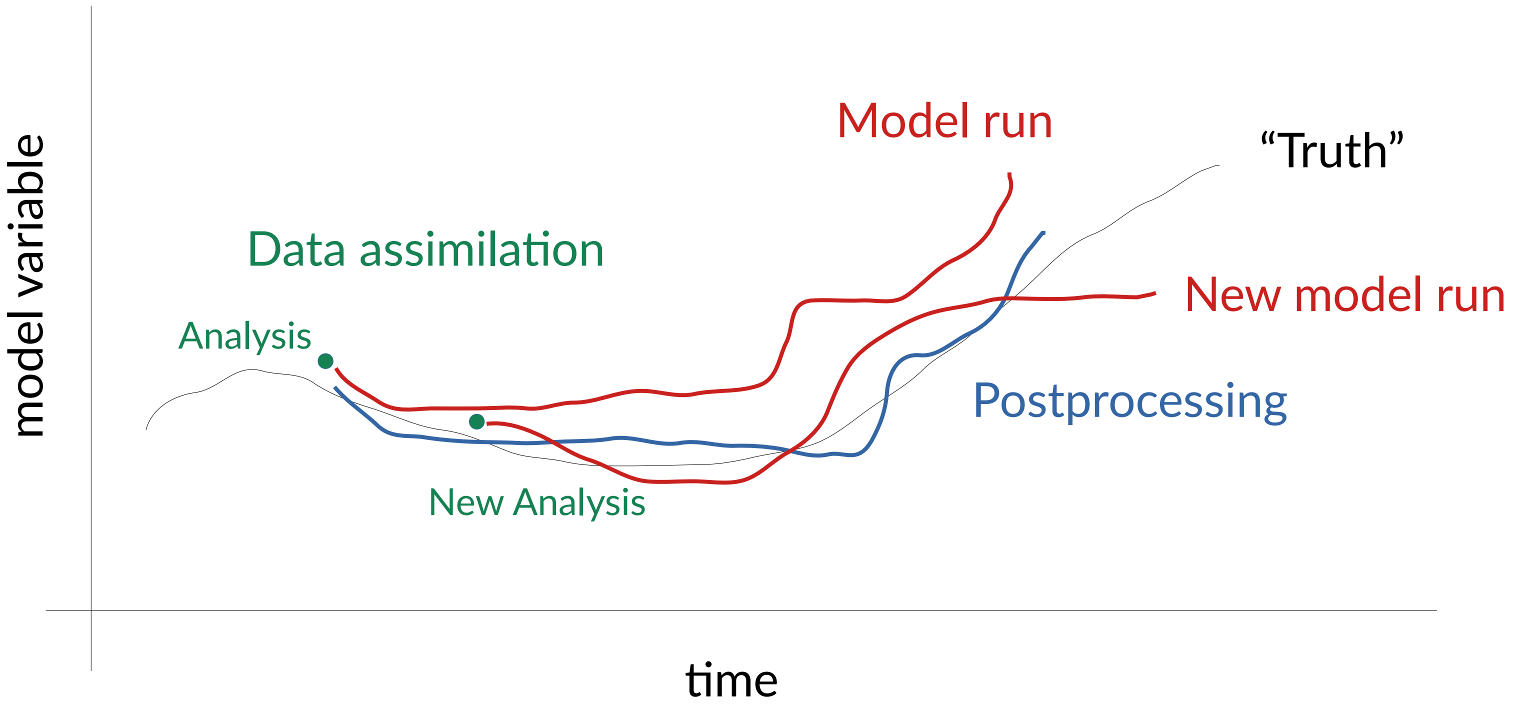
Estimation of the best initial conditions for the model based on observations.

**2. Model run:**

Computing the model time evolution.

**3. Postprocessing**

Bringing back the model output to reality: bias correction, spread, ...



# Chaos and ensemble forecasting

## The atmospheric models are chaotic systems :
## → Sensitivity to initial conditions
## → Forecasts become “wrong” over time

### Predictability experiments in 1982: global rate of the errors growth

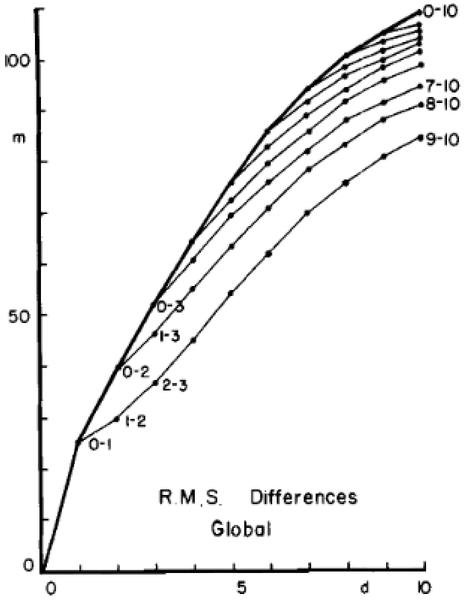


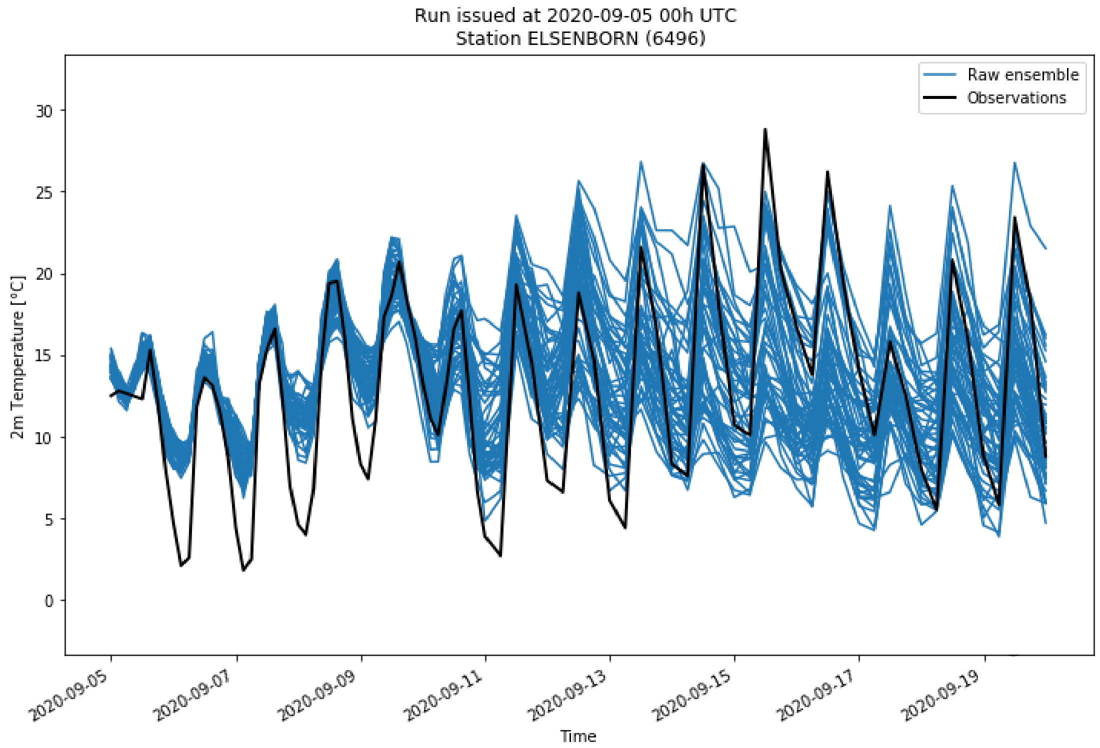

*Fig. 1.* Global root-mean-square 500-mb height differences $E_{jk}$, in meters, between $j$-day and $k$-day forecasts made by the ECMWF operational model for the same day, for $j < k$, plotted against $k$. Values of $(j,k)$ are shown beside some of the points. Heavy curve connects values of $E_{0k}$. Thin curves connect values of $E_{jk}$ for constant $k - j$.

The doubling time quoted in most predictability studies is the doubling time for errors of very small amplitude. To estimate this time from Fig. 1 we should have to extrapolate the thin curves backward until they approached zero—a rather difficult task. The task becomes easy, however, if we

The rate at which separate solutions of the model diverge is supposed to approximate the rate at which separate solutions of the true atmospheric equations diverge. If it does, and if, at some time during the forecast, the model could suddenly be replaced by the true equations, the remainder of the heavy curve would follow one of the thin curves.

*Lorenz 1982*

### Modern predictability experiments: local properties and ensemble forecasts



# Local predictability

**How much wrong models become over time depend on the model state.**
**→ In other words, how predictable is a model is a local property.**

*Figure 1: The scientific basis for ensemble prediction illustrated by the prototypical Lorenz model of low-order chaos, showing that in a nonlinear system, predictability is flow dependent. (a) A forecast with high probability, (b) forecast with moderate predictability, (c) forecast with low predictability.*

*Palmer et al. 2007*

# How to define the local predictability?

Consider a uniform ball of initial conditions around a model trajectory in the state space.
Over time:

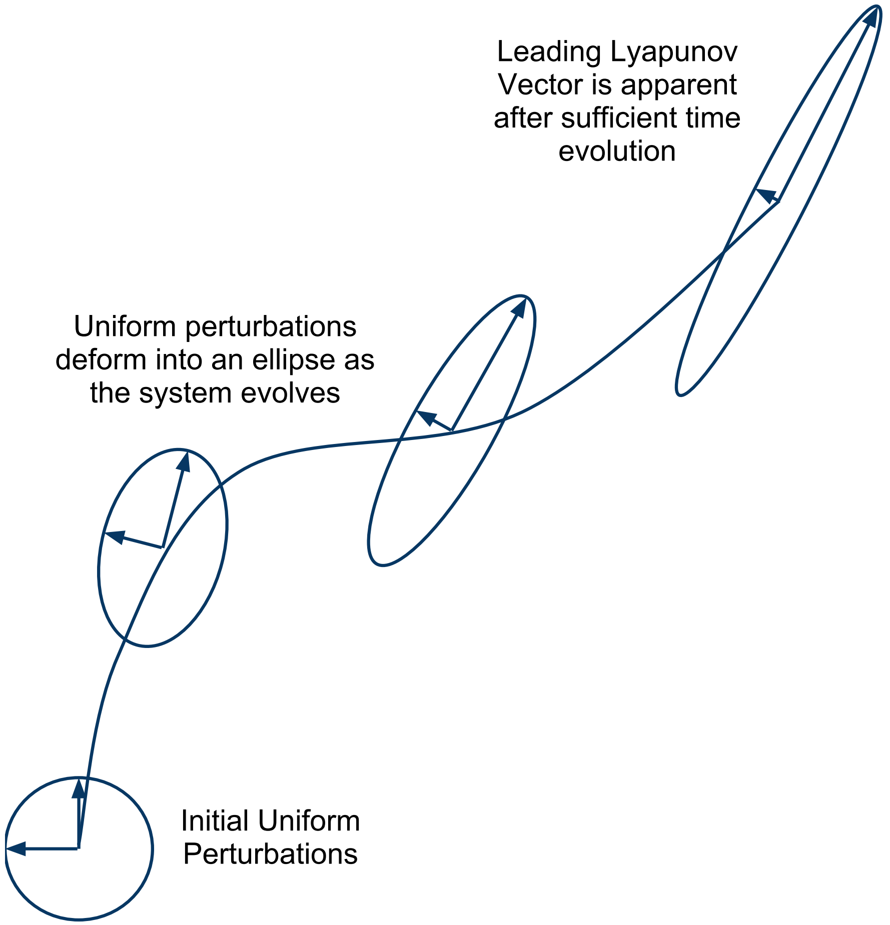
1) The stable directions around the trajectory will get closer to it, and the ball "shrinks" along these directions.

2) The unstable directions around the trajectory will get away from it, and the ball expands along these directions.

→ Over time, the ball deforms into an ellipsoïd.
→ After a while, the most unstable direction becomes apparent.

If the length of the semiaxes of the ellipsoid are denoted $\sigma_i$ , then the rates at which they stretch are called the Lyapunov exponents:

$$\sigma_i \sim e^{\lambda_i t} \qquad \text{for long times } t$$



# How to define the local predictability?

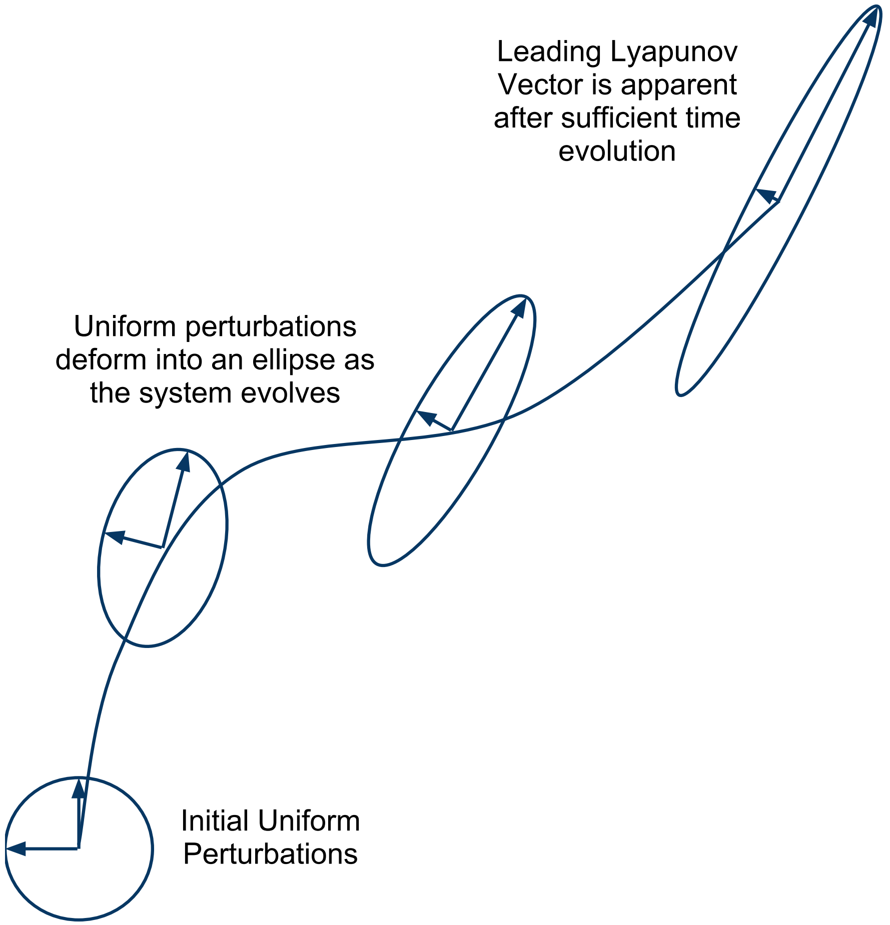
Consider a uniform ball of initial conditions around a model trajectory in the state space.
Over time:

1) The stable directions around the trajectory will get closer to it, and the ball "shrinks" along these directions.
2) The unstable directions around the trajectory will get away from it, and the ball expands along these directions.

→ Over time, the ball deforms into an ellipsoïd.
→ After a while, the most unstable direction becomes apparent.

If the length of the semiaxes of the ellipsoid are denoted $\sigma_i$ , then the rates at which they stretch are called the Lyapunov exponents:

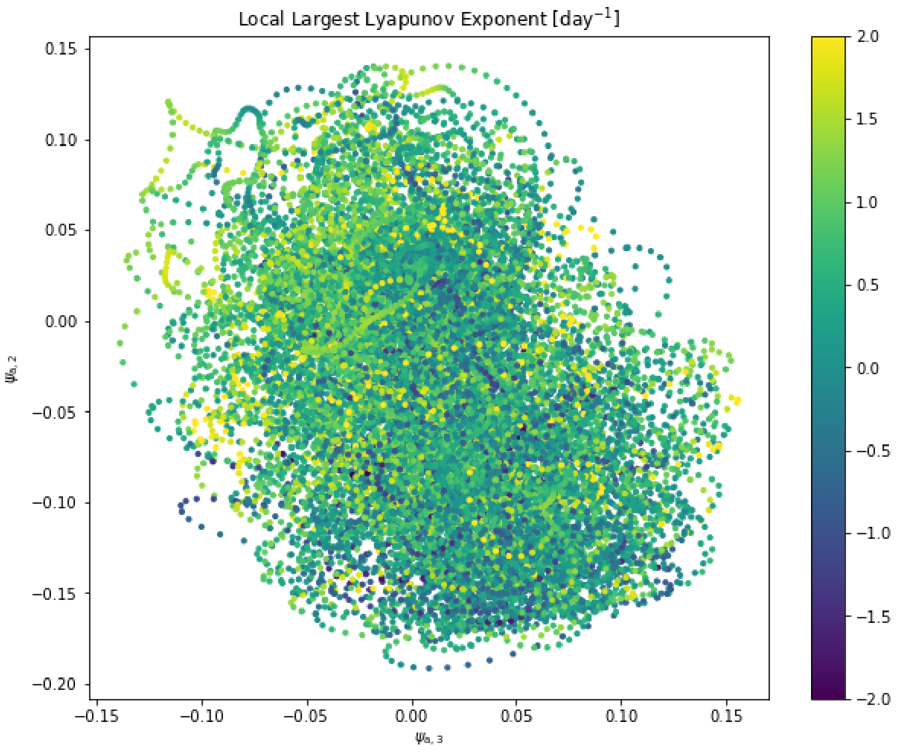
$$\sigma_i \sim e^{\lambda_i t} \quad \text{for long times } t$$

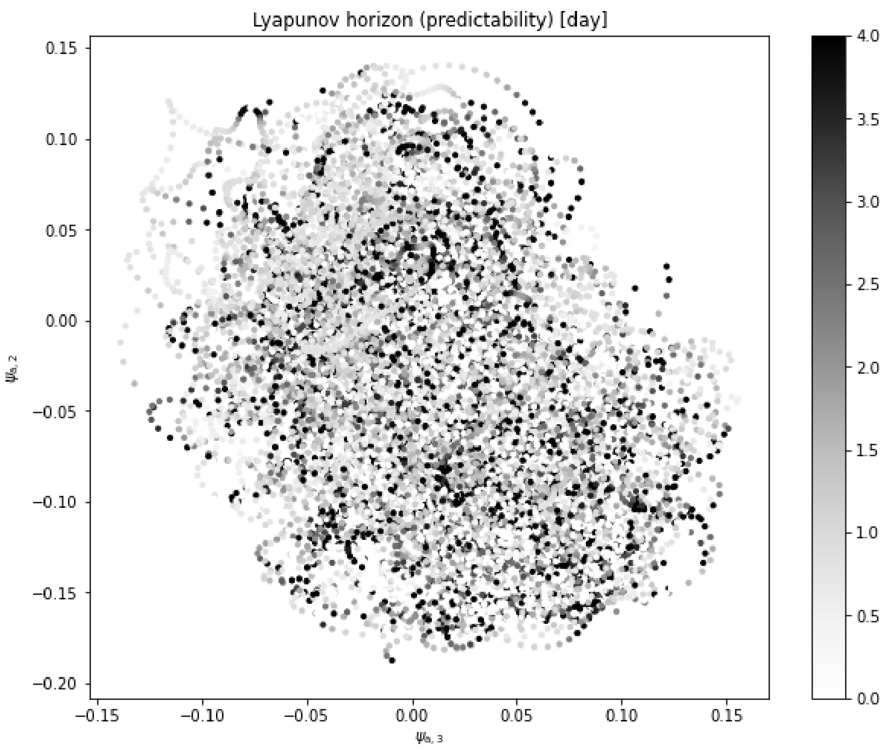



**Predictability: timescale 1 / $\lambda_1$ where $\lambda_1$ is the stretching rate of the most unstable direction.**

# Local predictability over a model attractor

Attractor = set of states toward which a system tends to evolve
Dissipative system !



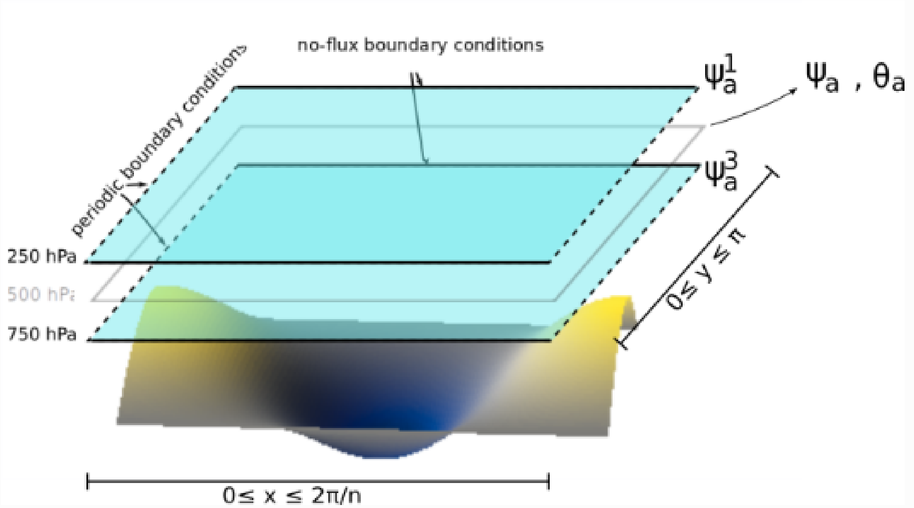




2-dimensional projection of the qgs atmospheric model attractor with the local predictability superimposed

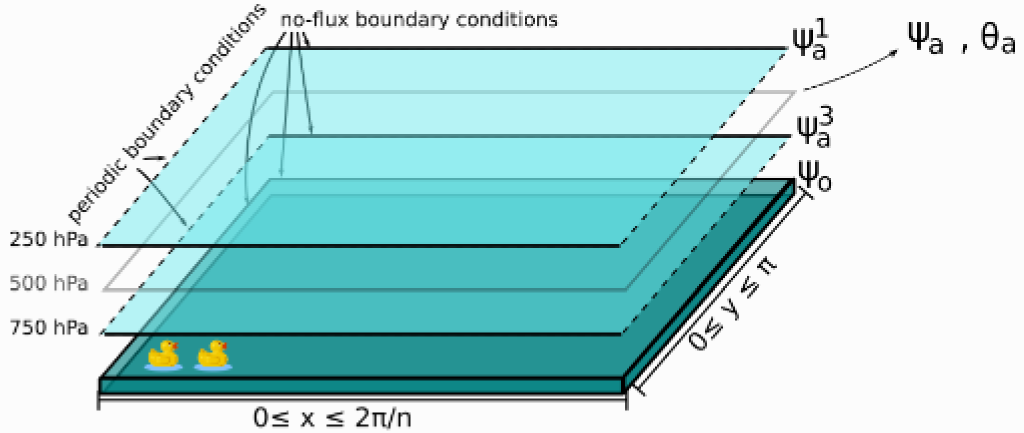
# qgs description

qgs is an atmospheric model for midlatitudes.
It models the dynamics of:

- a 2-layer quasi-geostrophic channel atmosphere
- on a beta plane (linearized Coriolis parameter)
- Coupling with a simple land (orography) or shallow-water ocean component (i.e. MAOOAM):
    - Momentum exchange (friction)
    - Heat exchange scheme or relaxation to climatological temperature




Sketch of the atmospheric model layers with a simple orography. The domain ($\beta$-plane, zonal and meridional coordinates are labeled as the $x$ and $y$ variables.

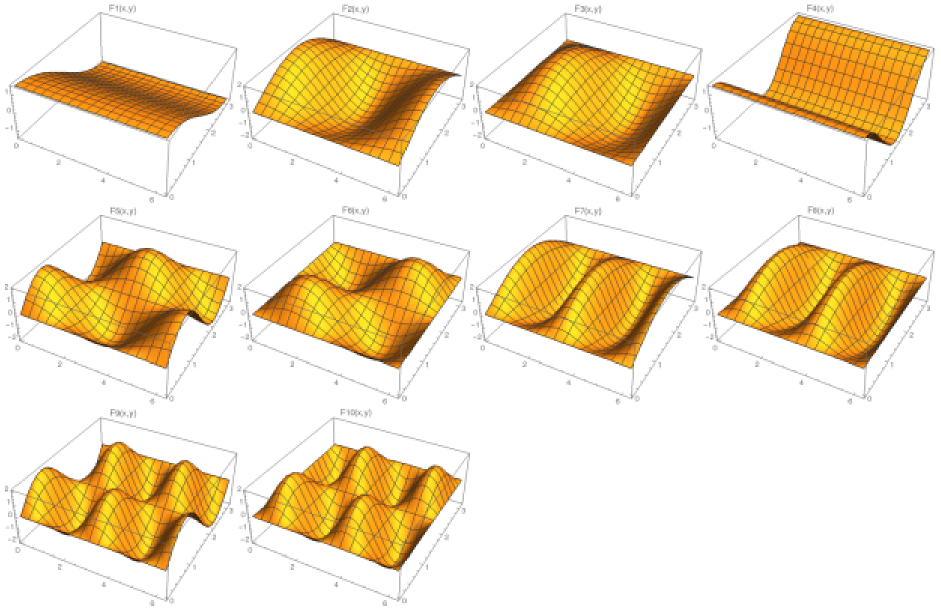



Sketch of the ocean-atmosphere coupled model layers. The domain ($\beta$-plane) zonal and meridional coordinates are labeled as the $x$ and $y$ variables.

# qgs is a spectral modeling framework

- Spectral approach:
  Fourier expansion of the stream function fields $\psi_{\mathrm{a}} = (\psi_{\mathrm{a}}^1 + \psi_{\mathrm{a}}^3)/2$
  and the temperature fields $\theta_{\mathrm{a}} = (\psi_{\mathrm{a}}^1 - \psi_{\mathrm{a}}^3)/2$

10 atmospheric modes: $F_i(x, y)$




Atm. streamfunction $\psi_{\mathrm{a}}(x,y) = \sum_{i=1}^{n_{\mathrm{a}}} \psi_{\mathrm{a},i}\, F_i(x,y)$

Atm. temperature $\theta_{\mathrm{a}}(x,y) = \sum_{i=1}^{n_{\mathrm{a}}} \theta_{\mathrm{a},i}\, F_i(x,y)$

→ 20-dimensional system

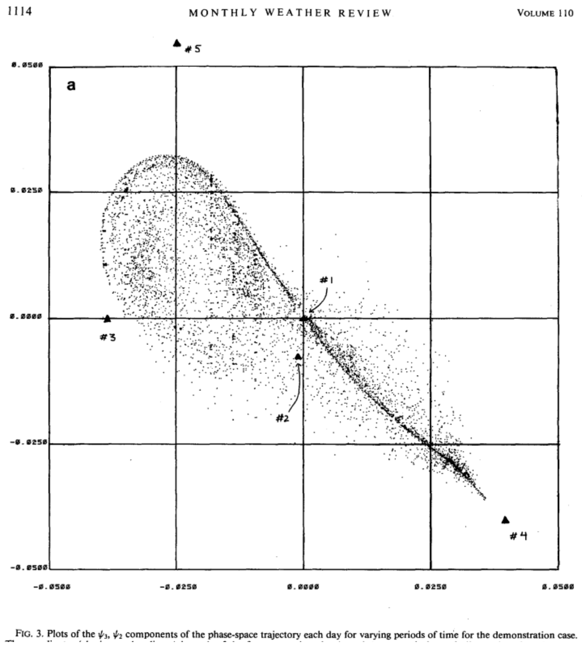
→ Same for the other components (land, ocean)

→ More modes can be added

# qgs is a QG spectral modeling framework

## Advantage of quasi-geostrophic (QG) spectral models

- Offer a good representation of the dry atmospheric dynamics
- Allow one to identify typical features of the atmospheric circulation, such as blocked and zonal circulation regimes, and low-frequency variability
- Reduced- or intermediary-order model → understanding of the processes at play
- Fast generation of long time series




Reinhold & Pierrehumbert, MWR 1982

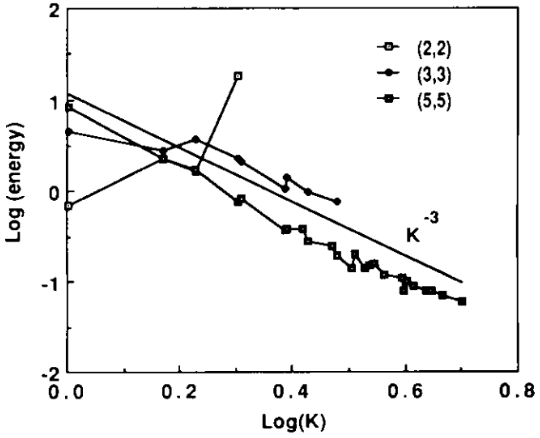



*Fig. 10.* Kinetic energy spectrum as a function of total wavenumber, $K$, for (2, 2), (3, 3) and (5, 5) models. A $K^{-3}$ profile is also shown.

O'Brien & Branscome, Tellus 1988

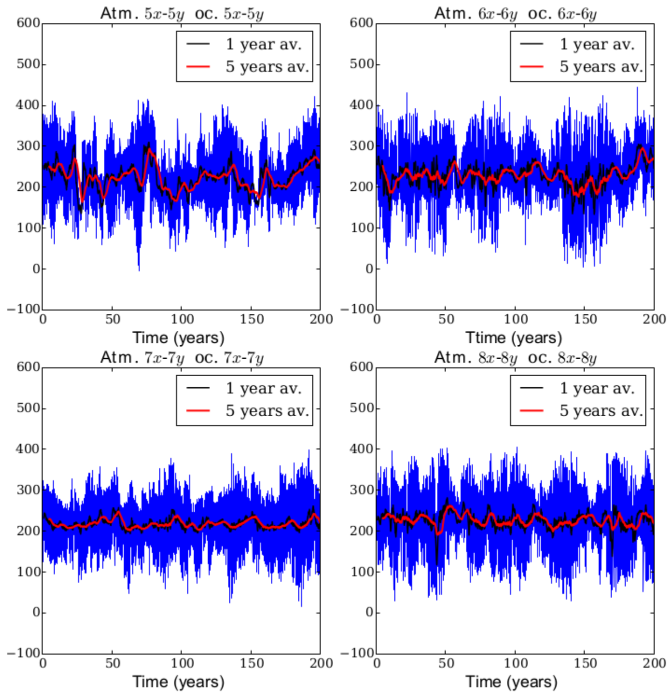
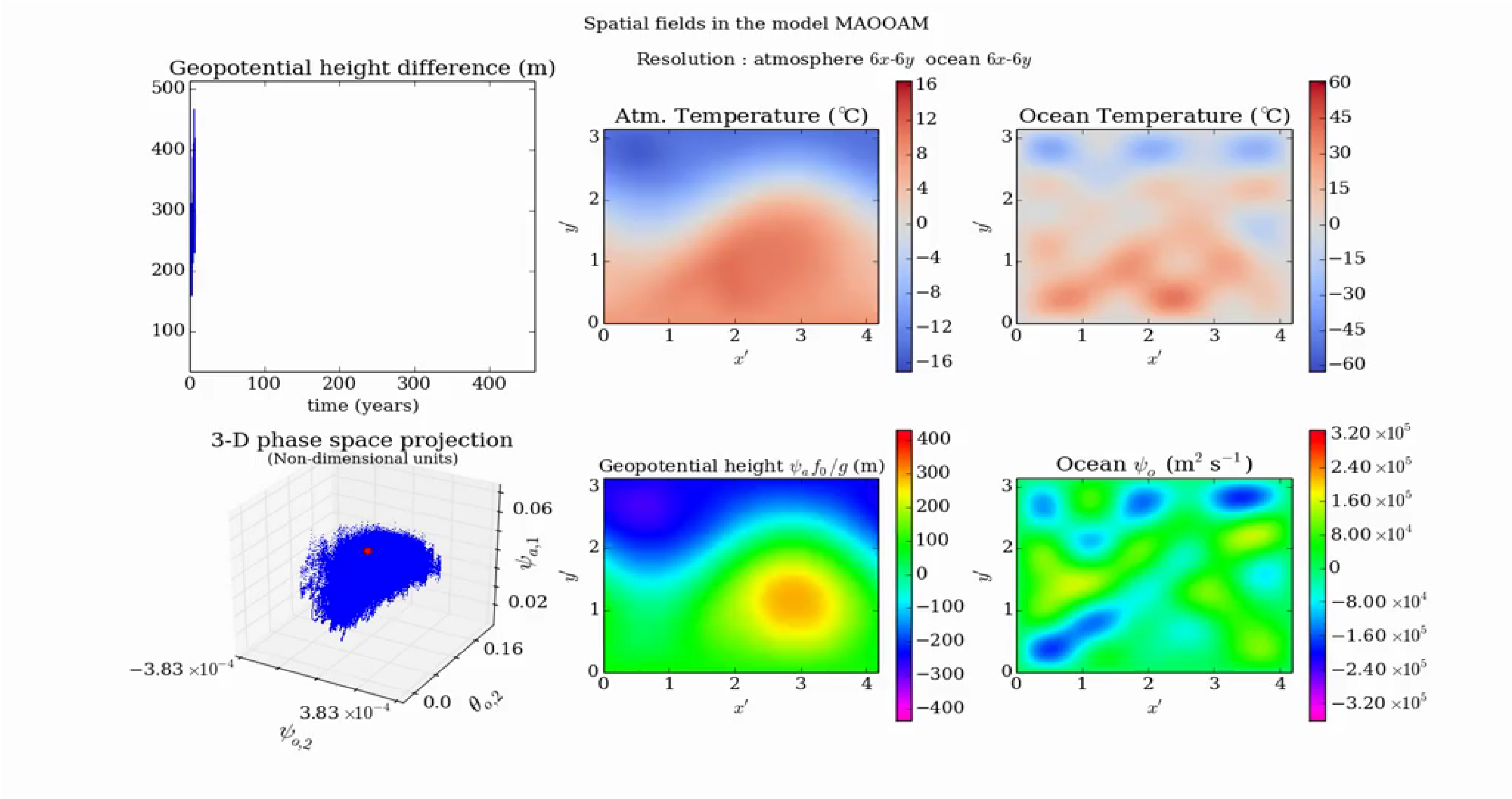



**Figure 8.** Time series of the geopotential height difference (continued from Fig. 7).

De Cruz et al., GMD 2016.

# Example: MAOOAM with qgs

If you couple the atmosphere to a shallow-water ocean component...




*Click on the image to play the video.*

# Inside qgs

## qgs was written with some key points in mind :

- A spectral QG model where it is easy to add new features and scales to the model (modularity)
- Coupling of the atmospheric QG model to different components (ocean, land, ...)
- Straightforward configuration and usage with modern tools (Jupyter notebook)
- A **reasonably fast** Python implementation (thanks to Numba)
- Free lunch: Easily interface with amazing libraries available (e.g: SciPy, pyDMD, machine learning libraries, etc...)
- Symbolic core using the SymPy library
- Didactical and comprehensive documentation

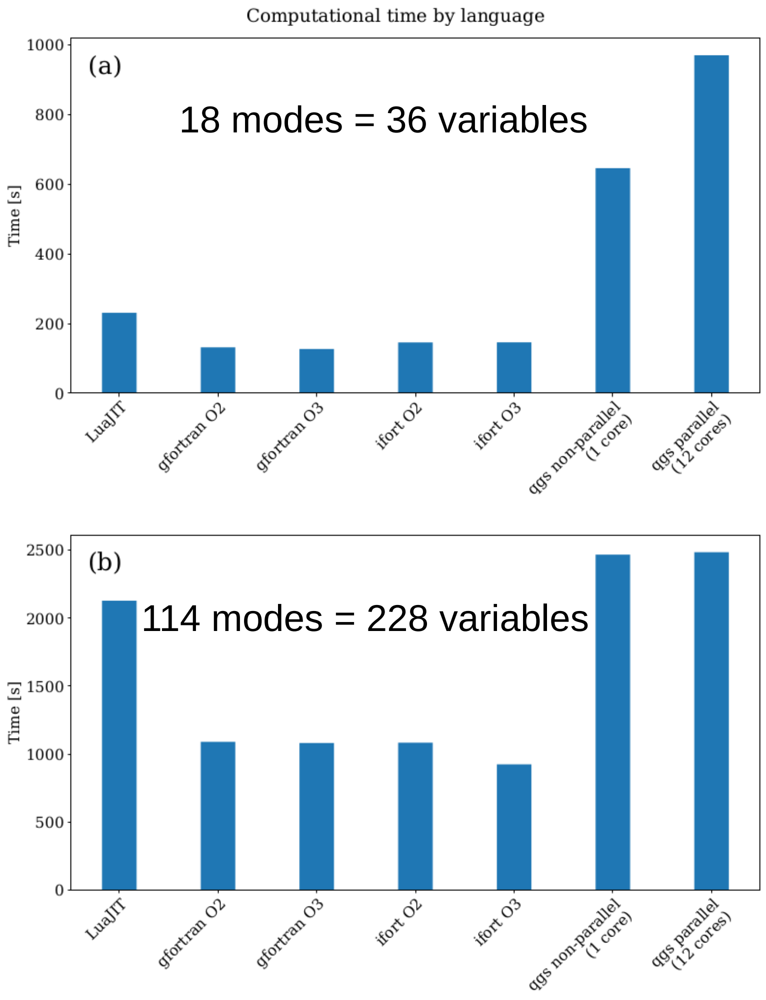



**Figure 1:** Computational times in seconds of different MAOOAM implementations: (a) time to compute a $10^7$ timeunits trajectory with a low-order model configuration (36 variables). (b) time to compute a $10^6$ timeunits trajectory with a higher-order model configuration (228 variables).

Demaeyer et al., JOSS (2020)

# Jupyter example

See full example at :
https://github.com/Climdyn/qgs/blob/master/notebooks/simple_run.ipynb

## Simple example: An atmosphere with an orography

### Loading modules

Entrée [5]:

```python
from params.params import QgParams
from integrators.integrator import RungeKuttaIntegrator, RungeKuttaTglsIntegrator
from functions.tendencies import create_tendencies
from plotting.util import std_plot
```

### Setting and printing the model parameters

Entrée [7]:

```python
# Model parameters instantiation with some non-default specs
model_parameters = QgParams({'phi0_npi': np.deg2rad(50.)/np.pi, 'hd':0.3})
# Mode truncation at the wavenumber 2 in both x and y spatial coordinate
model_parameters.set_atmospheric_channel_fourier_modes(2, 2)

# Changing (increasing) the orography depth and the meridional temperature gradient
model_parameters.ground_params.set_orography(0.4, 1)
model_parameters.atemperature_params.set_thetas(0.2, 0)
```

Entrée [8]:

```python
# Printing the model's parameters
model_parameters.print_params()
```

```
Qgs parameters summary
======================

General Parameters:
'time_unit': days,
'rr': 287.058  [J][kg^-1][K^-1]  (gas constant of dry air),
'sb': 5.67e-08  [J][m^-2][s^-1][K^-4]  (Stefan-Boltzmann constant),

Scale Parameters:
'scale': 5000000.0  [m]  (characteristic space scale (L*pi)),
'f0': 0.0001032  [s^-1]  (Coriolis parameter at the middle of the domain),
'n': 1.3    (aspect ratio (n = 2 L_y / L_x)),
'rra': 6370000.0  [m]  (earth radius),
'phi0_npi': 0.2777777777777778    (latitude expressed in fraction of pi),
'deltap': 50000.0  [Pa]  (pressure difference between the two atmospheric layers),

Atmospheric Parameters:
'kd': 0.1  [nondim]  (atmosphere bottom friction coefficient),
'kdp': 0.01  [nondim]  (atmosphere internal friction coefficient),
```

# Jupyter example

See full example at :
https://github.com/Climdyn/qgs/blob/master/notebooks/simple_run.ipynb

## Simple example: An atmosphere with an orography

### Integrating the model

Creating the tendencies function

Entrée [9]:
```
%%time
f, Df = create_tendencies(model_parameters)
```

```
CPU times: user 4.34 s, sys: 86.6 ms, total: 4.43 s
Wall time: 2.3 s
```

### Time integration

Defining an integrator

Entrée [10]:
```
integrator = RungeKuttaIntegrator()
integrator.set_func(f)
```

Start on a random initial condition and integrate over a transient time to obtain an initial condition on the attractors

Entrée [11]:
```
%%time
ic = np.random.rand(model_parameters.ndim)*0.1
integrator.integrate(0., 200000., dt, ic=ic, write_steps=0)
time, ic = integrator.get_trajectories()
```

```
CPU times: user 13.4 ms, sys: 12.4 ms, total: 25.8 ms
Wall time: 7.83 s
```

Now integrate to obtain a trajectory on the attractor

Entrée [12]:
```
%%time
integrator.integrate(0., 100000., dt, ic=ic, write_steps=write_steps)
time, traj = integrator.get_trajectories()
```

# Jupyter example

See full example at :
https://github.com/Climdyn/qgs/blob/master/notebooks/simple_run.ipynb

## Simple example: An atmosphere with an orography

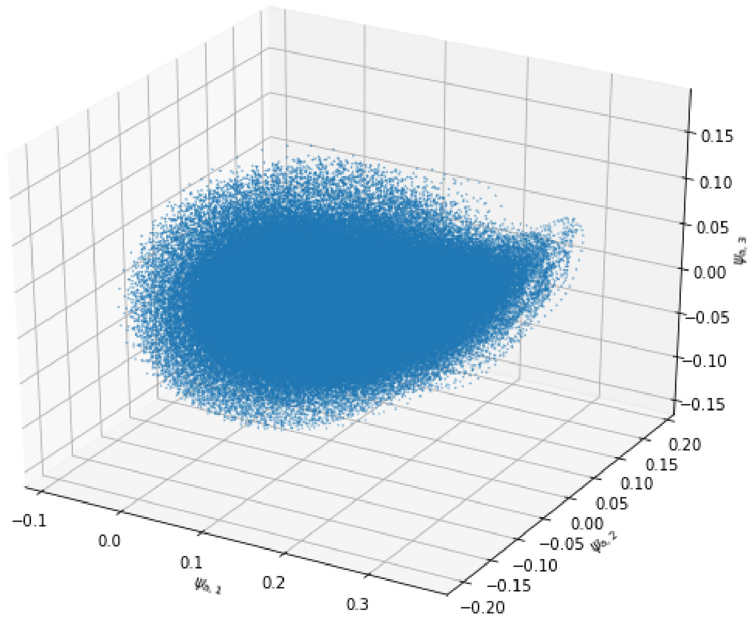
### Plotting the result

Entrée [13]:
```
varx = 0
vary = 1
varz = 2

fig = plt.figure(figsize=(10, 8))
axi = fig.gca(projection='3d')

axi.scatter(traj[varx], traj[vary], traj[varz], s=0.2);

axi.set_xlabel('$'+model_parameters.latex_var_string[varx]+'$')
axi.set_ylabel('$'+model_parameters.latex_var_string[vary]+'$')
axi.set_zlabel('$'+model_parameters.latex_var_string[varz]+'$');
```




**3d projection of the attractor**

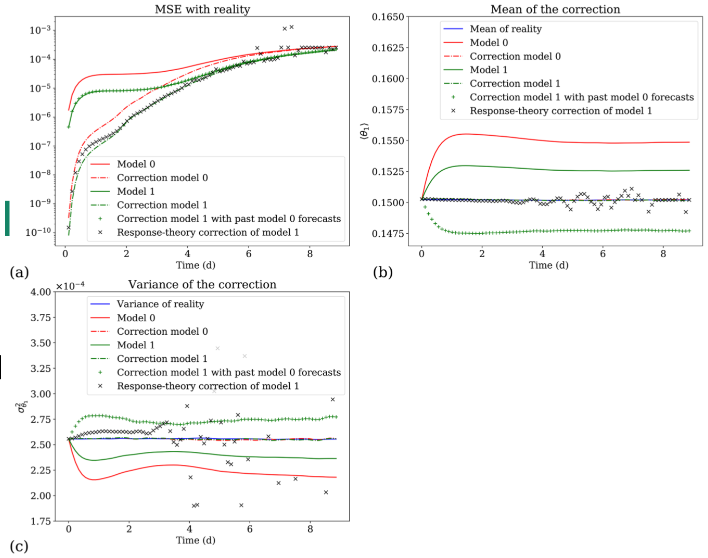
Various other example notebooks available at :
https://github.com/Climdyn/qgs/tree/master/notebooks

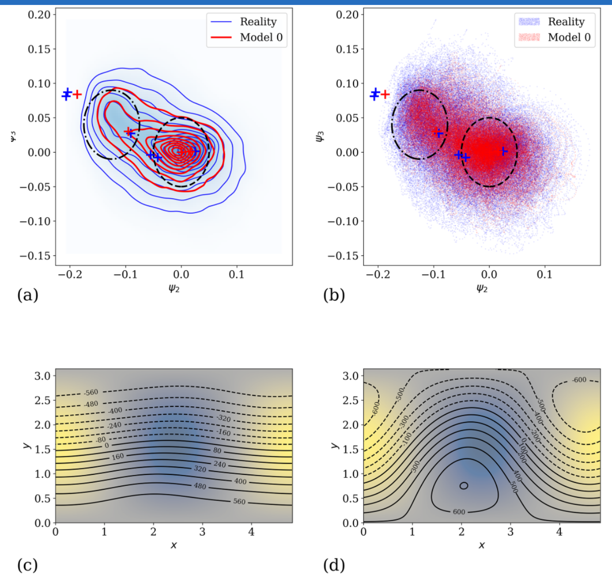
# Recent studies using qgs

## Post-processing

Using qgs with an orography to study the application of response theory for statistical postprocessing

Demaeyer & Vannitsem. Correcting for model changes in statistical postprocessing – an approach based on response theory, *Nonlin. Processes Geophys.*, **27**, 307–327 (2020), https://doi.org/10.5194/npg-27-307-2020.




## Data assimilation

Using qgs with an ocean (MAOOAM) and an ensemble Kalman filter (EnKF)

Carrassi, et al. Data assimilation for chaotic systems. in *Data Assimilation for Atmospheric, Oceanic and Hydrological Applications* (2020). Springer Science

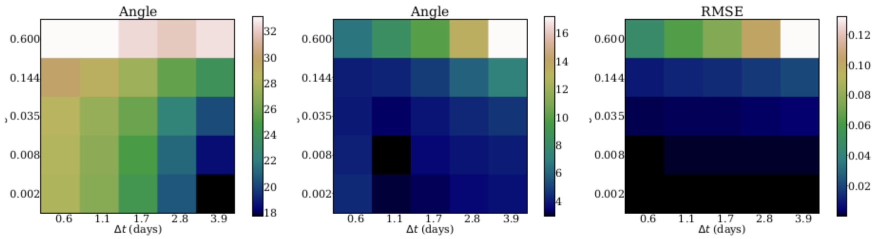



**Fig. 7** MAOOAM coupled Ocean-Atmosphere model. Time- and ensemble-averaged angle given by Eq. (37) between the anomalies of the EnKF and the unstable–neutral subspace (left panel, shadow colours, in degree) or the unstable-near–neutral subspace (mid panel, shadow colours, in degree). The time averaged normalised RMSE of the EnKF analysis is also depicted (right panel, shadow colours). All the figures are on the plane $(x, y) = (\Delta t, \sigma^{\%})$. The set-up is $\mathbf{H} = \mathbf{I}_{N_x}$, $\mathbf{R} = \sigma^{\%}\mathrm{diag}(\sigma^1_{\mathrm{md}}, \ldots, \sigma^{N_x}_{\mathrm{md}})$ and $N = 20$. The unstable-near–neutral subspace is defined as follows: it includes the subspace spanned by the unstable and neutral $n_0 = 6$ directions, but also an additional $n_1 = 10$ stable but near–neutral directions with LEs $\lambda_i \in [-5 \times 10^{-3}, 0[$ day$^{-1}$.

# Recent studies using qgs

## Climate prediction

Using qgs (MAOOAM) forced with a model of ENSO to study its impact on midlatitude climate prediction, through the lens of pullback attractors.

Vannitsem, Demaeyer & Ghil (2021). Extratropical low-frequency variability with ENSO forcing: A reduced-order coupled model study. arXiv preprint arXiv:2103.00517.

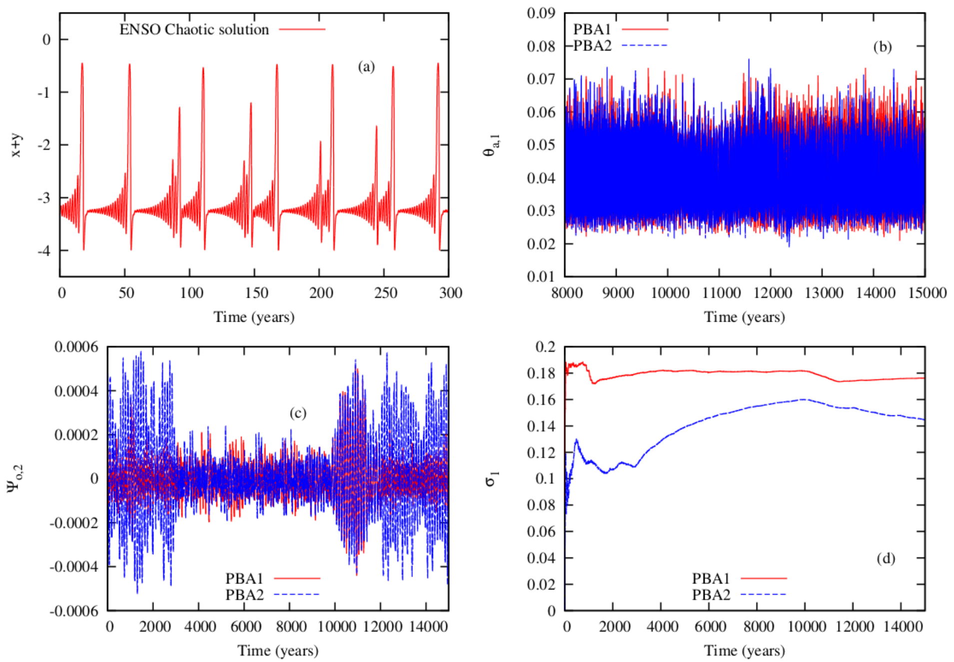



**Figure 10.** As in Fig. 3 but for chaotic ENSO forcing, with $g = 0.03$. (a) Time segment of 300 yr from the chaotic forcing, identical to the one displayed in Fig. 1(b); (b,c) evolution of the

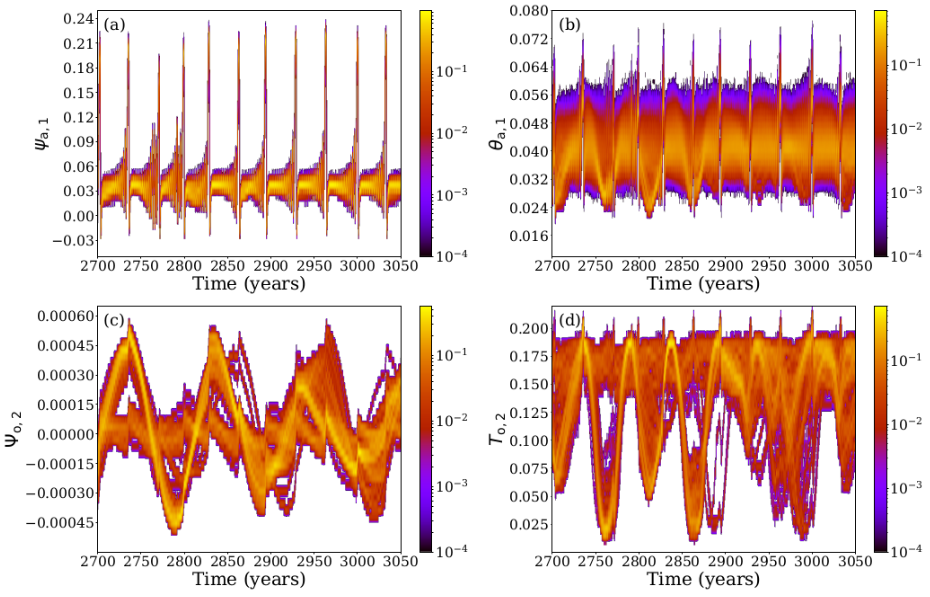



**Figure 13.** As in Fig. 12, but for PBA2.

# More information about qgs

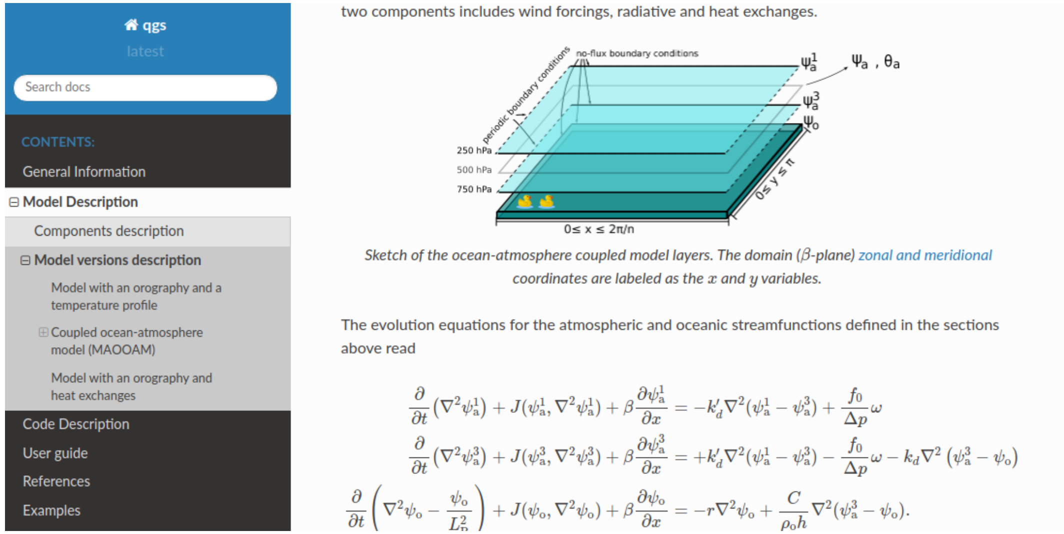
## Github repository

https://github.com/Climdyn/qgs

See also:
https://qgs.readthedocs.io/en/latest/files/general_information.html#contributing-to-qgs

## Documentation

https://qgs.readthedocs.io




## Description paper

Demaeyer, De Cruz & Vannitsem, (2020). qgs: A flexible Python framework of reduced-order multiscale climate models. *Journal of Open Source Software*, **5**(56), 2597, https://doi.org/10.21105/joss.02597

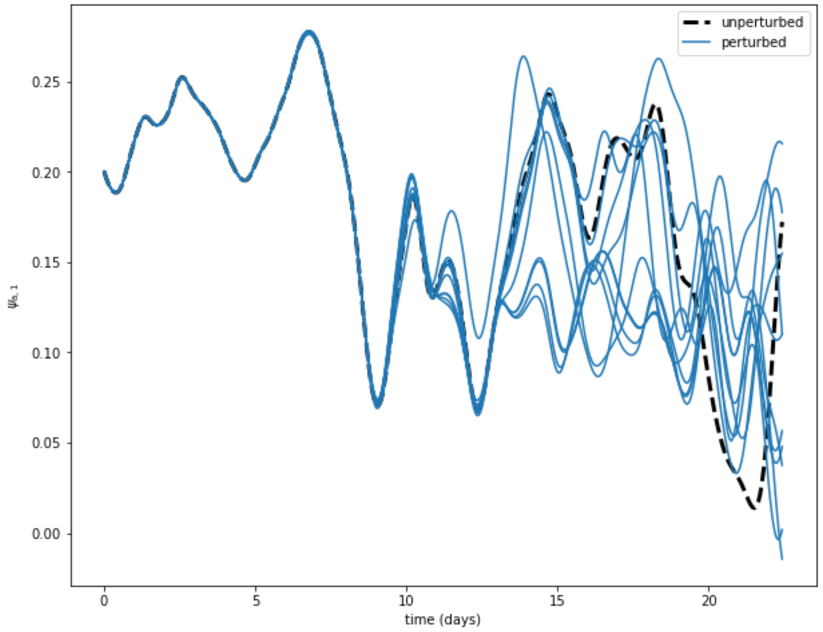
# In this tutorial about predictability

## We are going to use qgs simplest atmospheric model to study its predictability...

qgs can be used to make real-world looking forecast experiment:

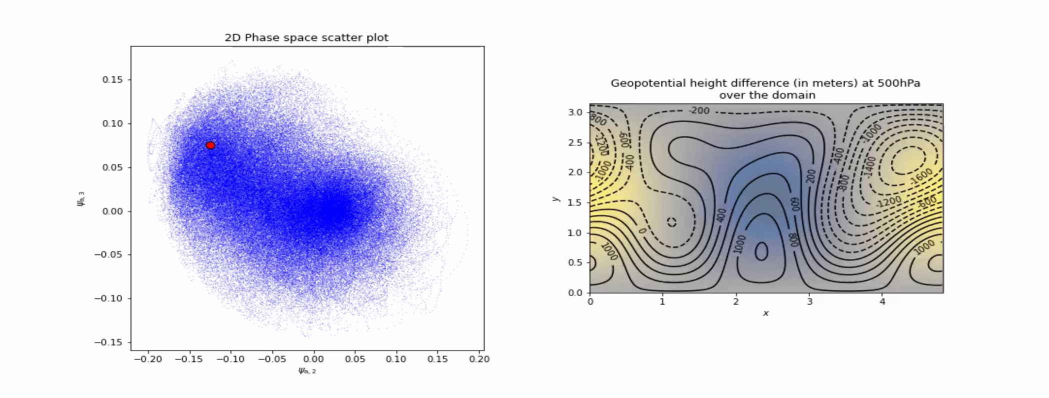
- will allow us to get a grasp on the concept of predictability in high-dimensional forecast models
- computation of relevant quantities indicating the typical timescales at which the model is predictable (autocorrelation time, Lyapunov exponents, ...)

Simple example with an orography: ensemble forecast experiment




Simple example with an orography: animation

*Click on the image to play the video.*



# THANK YOU

**Het Koninklijk Meteorologisch Instituut**

**L'Institut Royal Météorologique**

**Das Königliche Meteorologische Institut**

**The Royal Meteorological Institute**

RMI

The RMI provides reliable public service realized by empowered staff and based on research, innovation and continuity.

Het KMI verleent een betrouwbare dienstverlening aan het publiek en de overheid gebaseerd op onderzoek, innovatie en continuïteit.

L'IRM fournit un service fiable basé sur la recherche, l'innovation et la continuité au public et aux autorités.

Vertrauenswürdige Dienstleistungen für Öffentlichkeit und Behörden begründet auf Forschung, Innovation und Kontinuität.

# Terminology

## Some definitions:

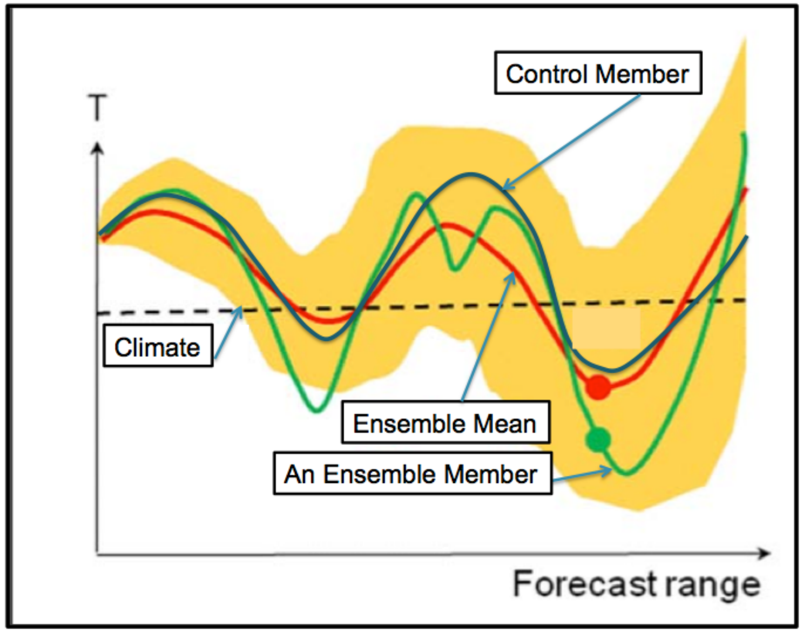
- Lead time: time past since the initial time of the forecast (here called *forecast range* → )

- Ensemble mean: mean of a variable X over the M ensemble members

$$\bar{X} = \sum_{m=1}^{M} X_m$$

- Control member: Member with the best initial condition estimated from observations

- Climate: Range of observed values of the past years forecasts initialized at the same date




*ECMWF Forecast User Guide*