HDF5 in geomagnetic data assimilation and visualisation

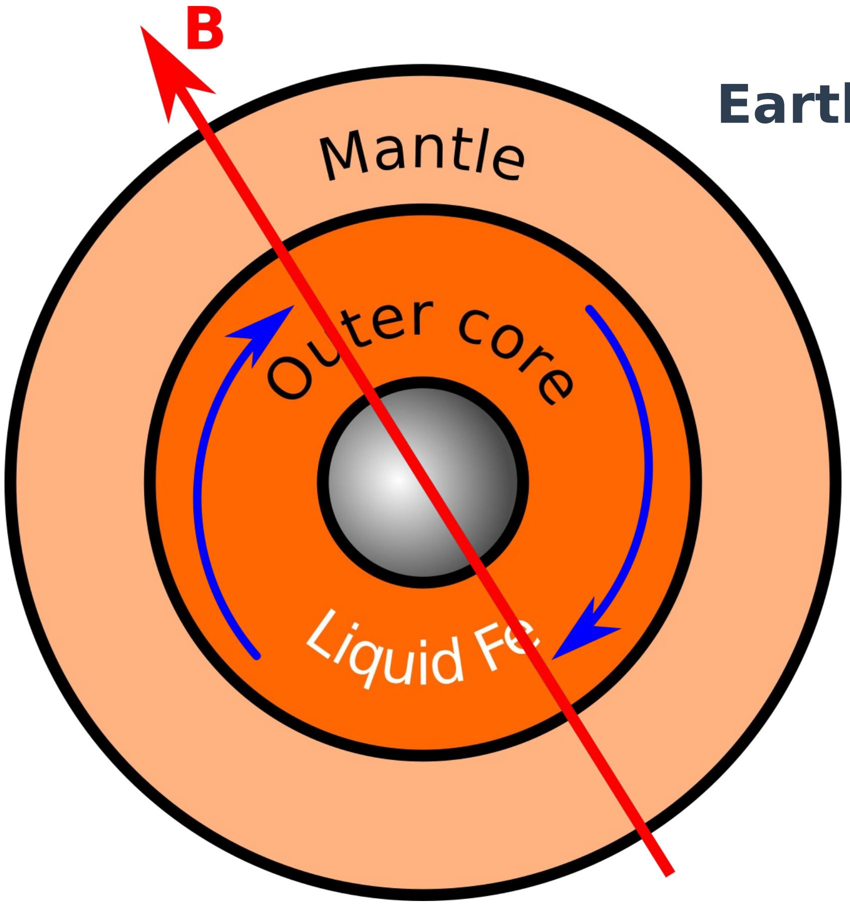
Loïc Huder, Nicolas Gillet, Franck Thollard

ISTerre, CNRS, Université Grenoble Alpes, Grenoble

- ISTerre : Earth sciences lab

- Geodynamo team

The Earth’s magnetic field



Earth’s magnetic field ⇆ Earth’s core flow

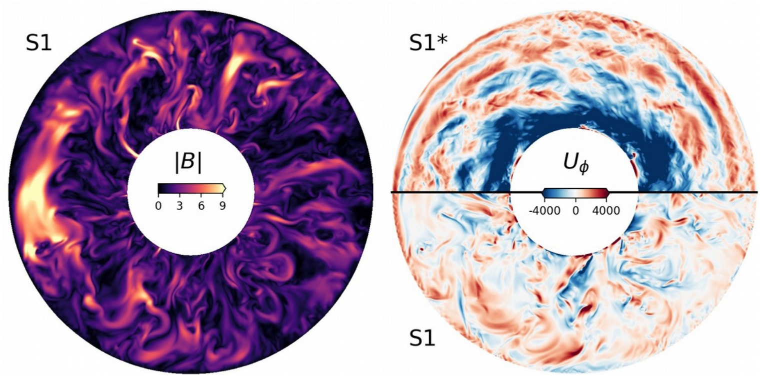
CHALLENGING PROBLEM :

- Thermal coupling
- Fluid mechanics
- Electromagnetism

Tackling the geodynamo problem

Numerical simulations

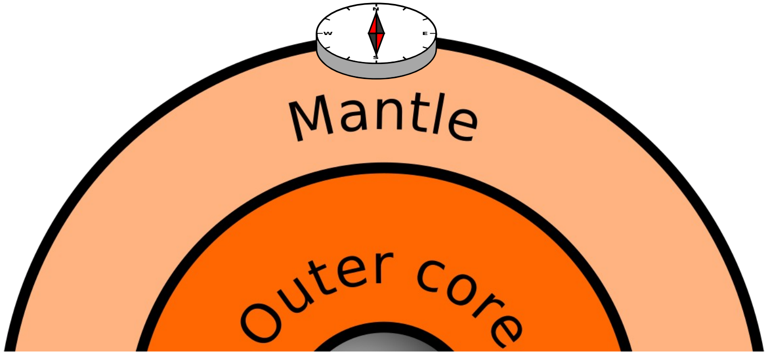
fast rotating fluid mechanics with turbulence



Schaeffer et al. GJI 2017

Physical measurements

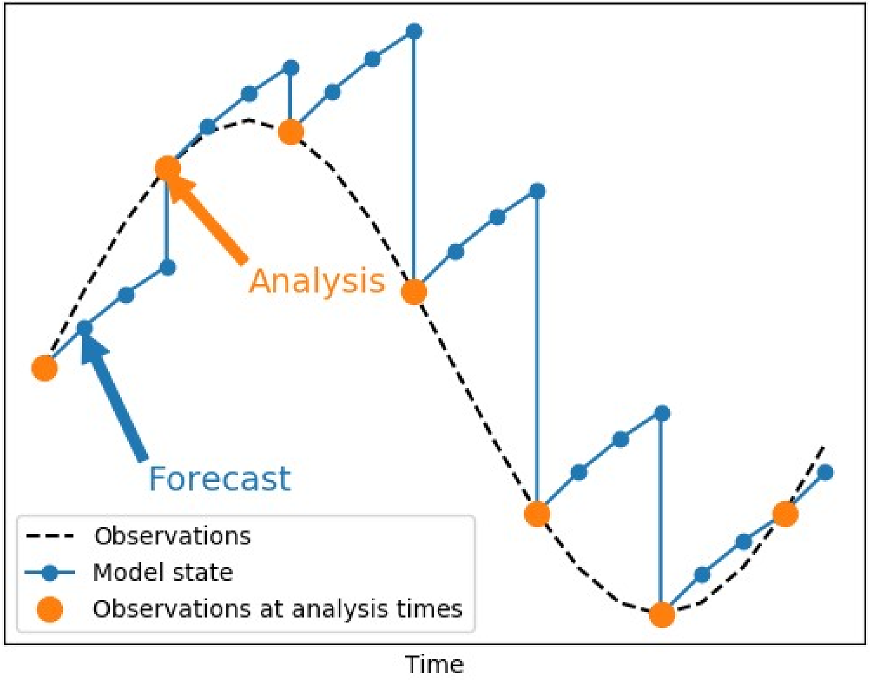
only magnetic field at the Earth's surface and its evolution in time...



Sequential data assimilation

- Forecast in time
  with numerical model

- Assimilate measure data
  in analyses



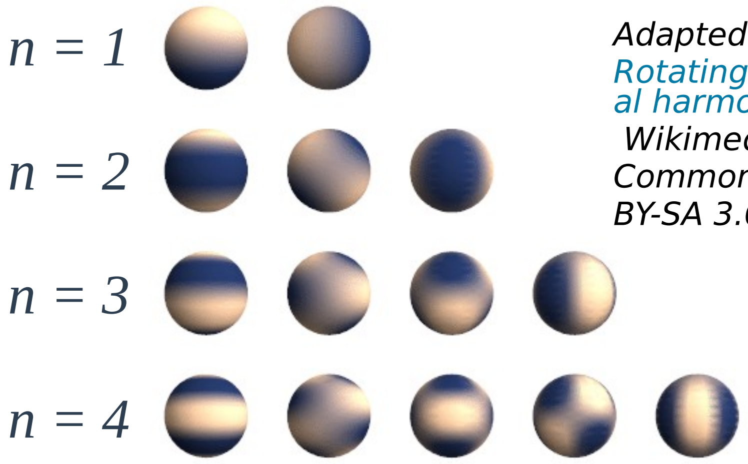
pygeodyn package

- Python package for geomagnetic data assimiliation
  - Forecast with reduced numerical model anchored to numerical simulations (J. Aubert, IPGP)
  - Analysis with up-to-date ground and satellite magnetic data

- In development for more than 1 year
  - Starting from Fortran code snippets
  - **Output format ?**

HDF5 with h5py

Output : Gauss coefficients



Adapted from Rotating spherical harmonics.gif, Wikimedia Commons, CC BY-SA 3.0

$$\boldsymbol{B} = -\nabla V$$

$$V(r,\theta,\phi,t) = a \sum_{n=0}^{N} \left(\frac{a}{c}\right)^{n+1} \sum_{m=0}^{n} \left(g_n^m(t) \cos(m\phi) + h_n^m(t) \sin(m\phi) \right) P_n^m(\cos\theta)$$

$\boldsymbol{b} = [g_n^m, h_n^m]$ is a vector of size $N(N+2)$

Usually, we have $N = 14$ → 224 coefficients at each timestep

Physical quantities to store

For each timestep (forecast or analysis):

$$\dot{\boldsymbol{b}} = A(\boldsymbol{b})\boldsymbol{u} + \boldsymbol{e}_r$$

- Secular variation (SV): 224
- Magnetic field (MF): 224
- Subgrid errors (ER): 224
- Core flow (U): 720 (N=18)

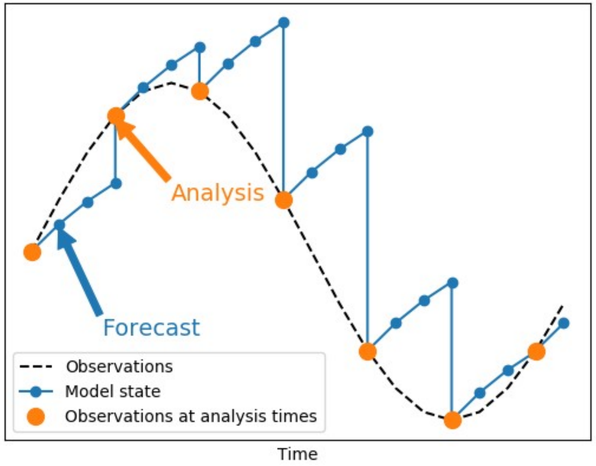
Output array shapes

- Forecasts, for each quantity (MF, SV, U, ER):

  NumPy array of shape

  (nb_model_realisations, nb_forecasts, nb_coefficients)

  Typically: (20, 100, 224 or 720) ~ 10 Mo



- Analyses, for each quantity (MF, SV, U, ER):

  NumPy array of shape

  (nb_model_realisations, nb_analyses, nb_coefficients)

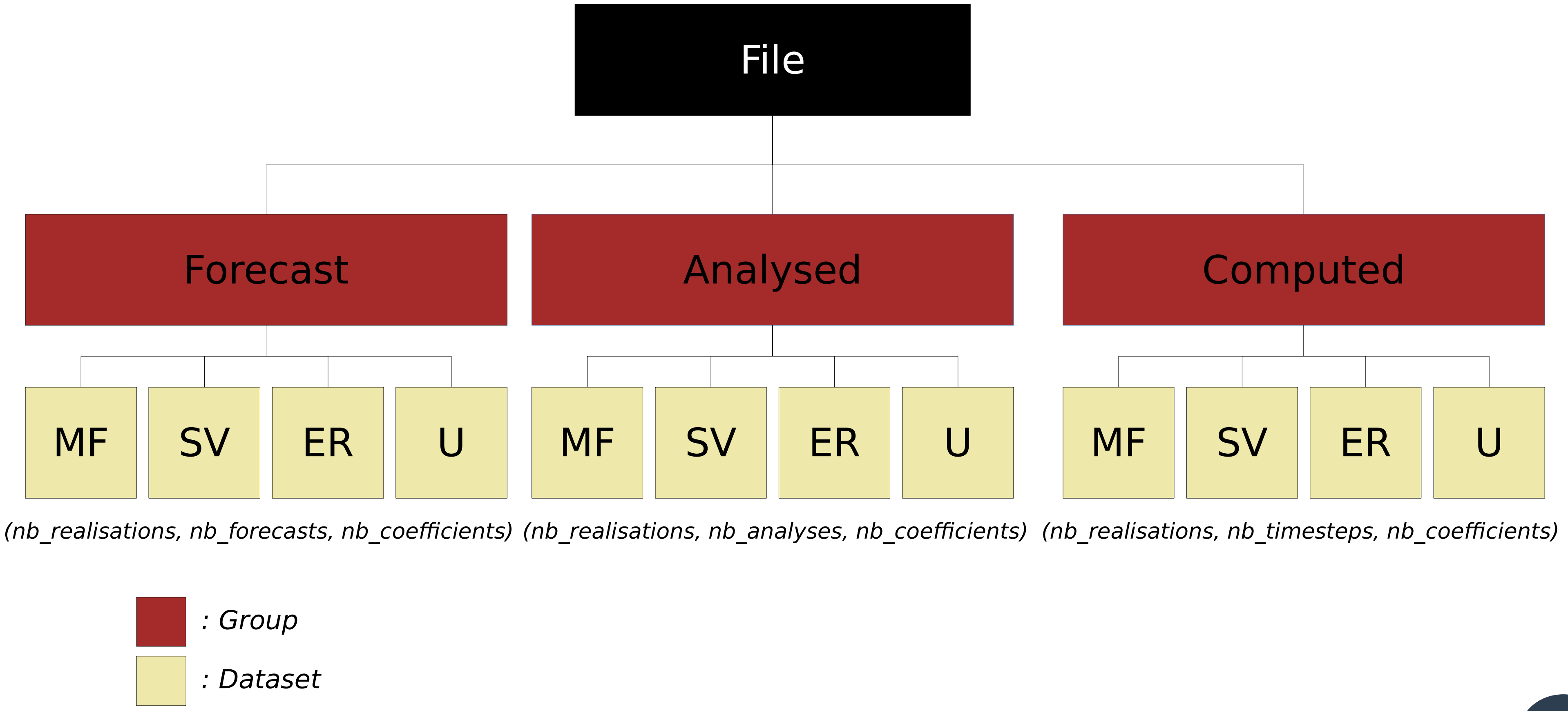
  Typically: (20, 50, 224 or 720) ~ 5 Mo

Old output format (<0.2)

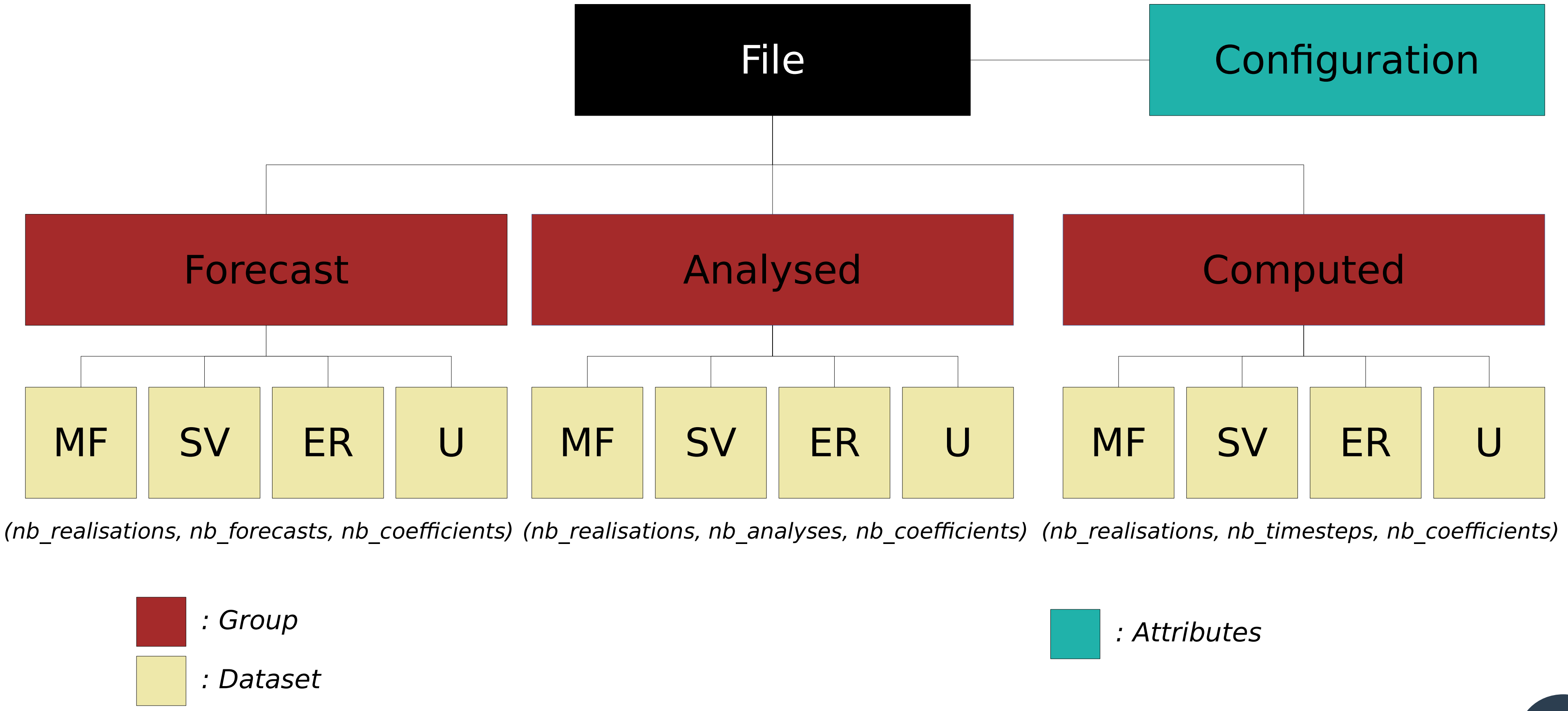
- Regular ASCII with a file for each:
  - Model realisation
  - Step type (forecast, analysis)
  - Quantity (MF, SV, U, ER)

- Ex : results from Barrois *et al. GJI* 2017
  - around 200 files
  - Not accurate/efficient
  - Difficult to manipulate

Now with HDF5



Now with HDF5



HDF5 attributes

```
Lu int 18
Lb int 14
t_start date 1980
t_end date 2015
dt_f months 6
dt_a months 12
```

Examples of configuration parameters

- All parameters stored as HDF5 attributes
  - Integers, floats, strings...
  - Dates as strings (*'1980-01'*)
  - Date arrays as string arrays
- Parameters can be extracted from an output HDF5 file to be reused

Why is it important ?

- Datasets are
  - Ordered
  - Easily accessed
  - **in the same file as parameters**
- In conjunction with other tools :
  - Semantic versioning & continuous release
  - Scientific testing

Reproducible Science

Partial summary

- Geomagnetic data assimiliation

- HDF5 to structure output data and embed computation parameters
  - Tool towards reproducible science/research

HDF5 for geomagnetic data visualisation

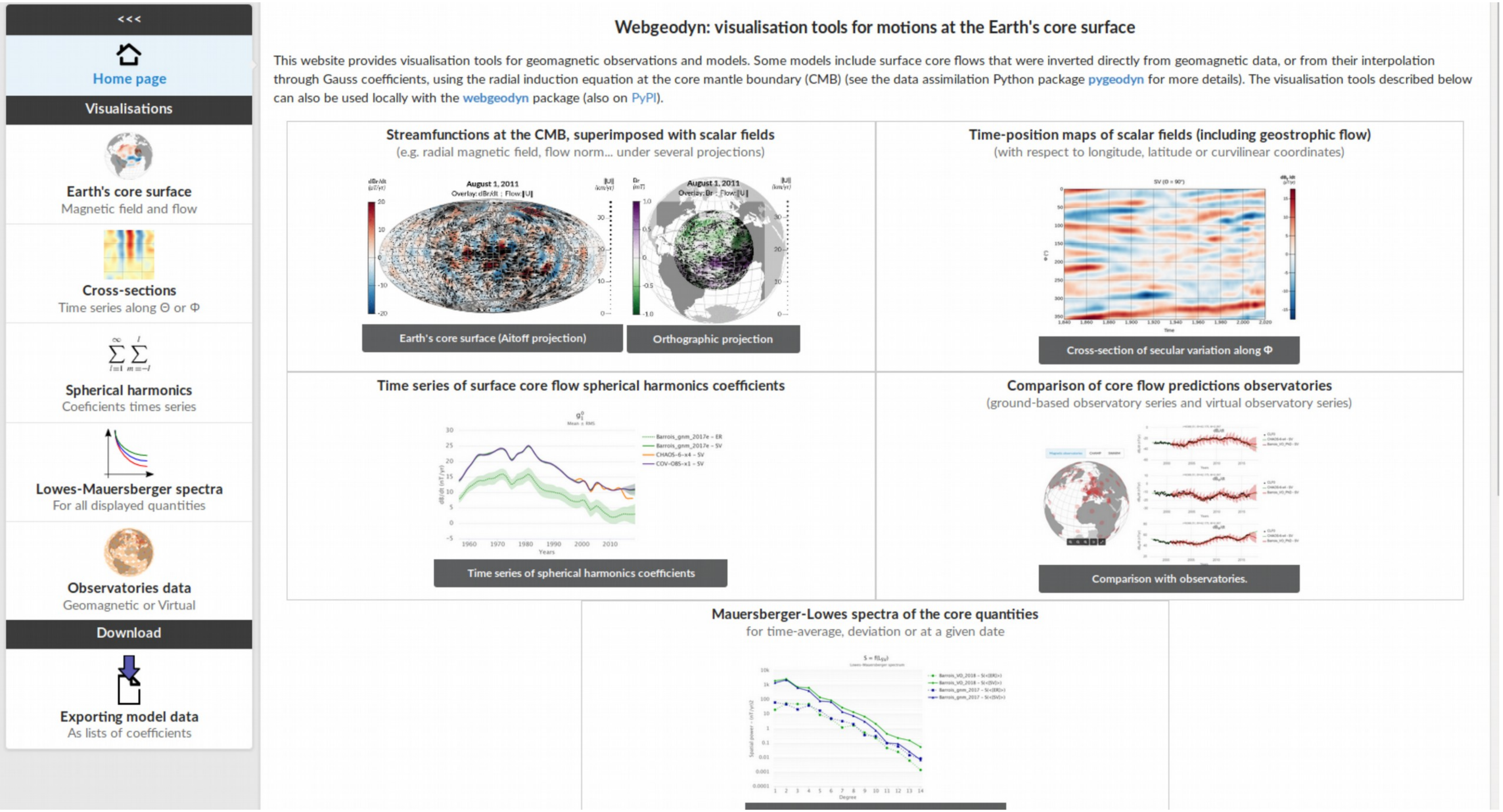
HDF5 in the *geodyn* suite

- Python package for geomagnetic data assimilation

- Python package for geomagnetic data visualisation
  - Web-based tool using a Tornado server
  - Available on PyPI
  - Deployed on *https://geodyn.univ-grenoble-alpes.fr/*

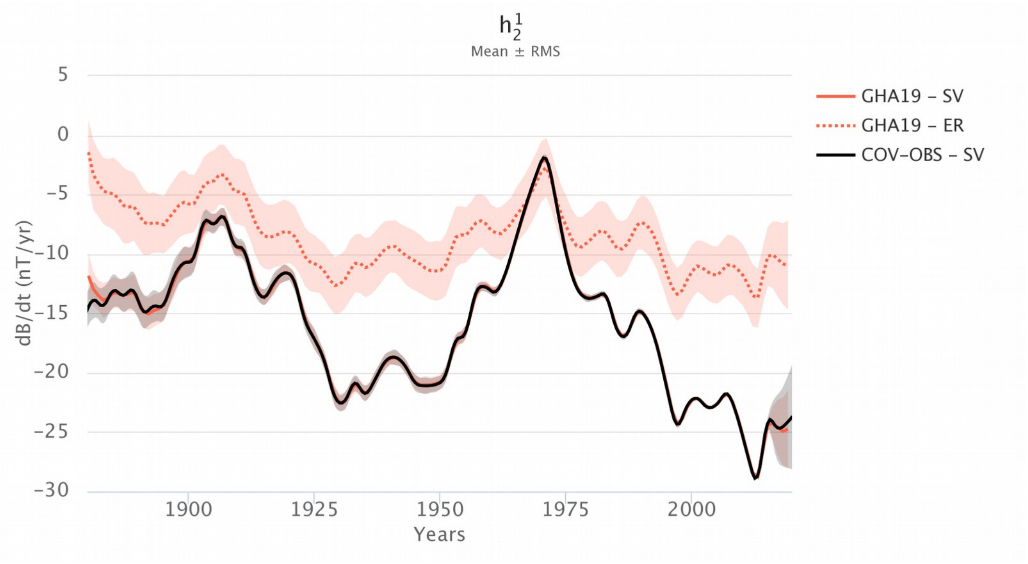
HDF5 (h5py) allows to load directly and efficently the datasets (faster than ASCII)

Webgeodyn webpage

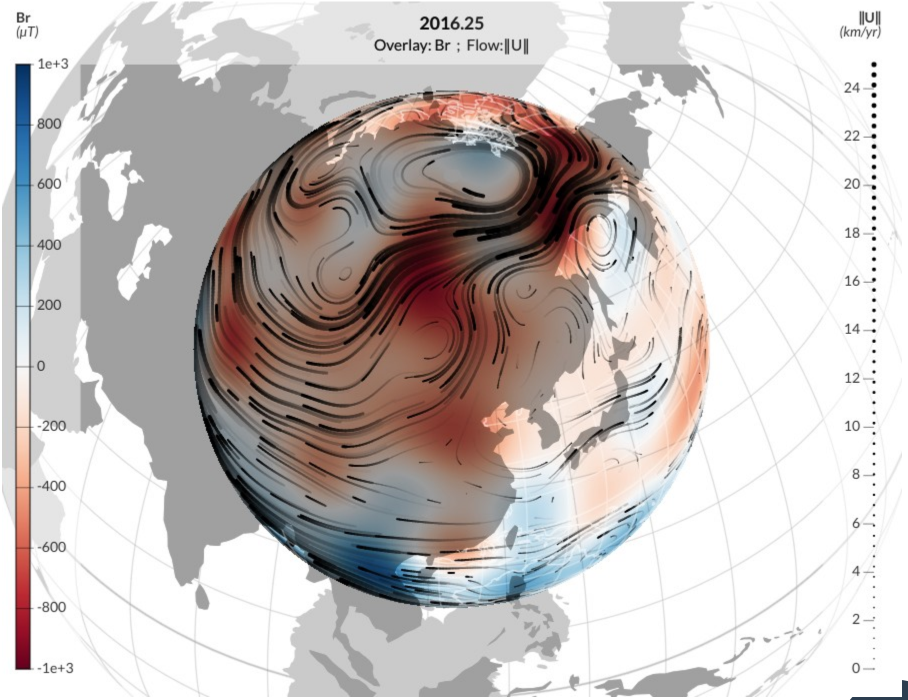


Visualisation examples

- Time-series of Gauss coefficients



- Display of magnetic field and core flow on the core surface



In summary

- Geomagnetic data assimiliation
- HDF5 to structure output data and embed computation parameters
  - Tool towards reproducible science/research
- Efficient interface for visualisation

Thank you for your attention !

For more information :

- Git repository of pygeodyn:

  https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/pygeodyn

- Git repository of webgeodyn:

  https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/webgeodyn

- Article presenting pygeodyn (and a bit of webgeodyn...):

  Huder, L., Gillet, N. & Thollard, F. pygeodyn 1.1.0: a Python package for geomagnetic data assimilation. Geoscientific Model Development 12, 3795–3803 (2019).

Continuous integration and release

Gitlab Pipelines :

- Unitary and scientific tests (against benchmarks)
- Sphinx documentation deployed via Gitlab Pages
- Continuous release triggered by manual pipeline (see dedicated slide)

https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/pygeodyn
https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/webgeodyn

Versioning : CHANGELOG

Changelog of pygeodyn

1.1.2 - 2019-08-27

Updated pygeodyn_data with GOVO_2019 (August 2019) dataset. Minor improvements to doc and CR.

1.1.1 - 2019-08-06

Fixed the reading order for errors of GOVO (wrong format of _COV.obs files was assumed).

1.1.0 - 2019-07-05

Major changes:

- Numerous improvements to the documentation
  - Reorganised into RST files
  - Expanded with tutorials, complete description configuration parameters, ...
  - RST files are now picked up by Sphinx (`make_doc.sh` script) to generate a navigable documentation in HTML also deployed online
- Data in `pygeodyn/data` is now stored as a submodule (`pygeodyn_data`)

In addition:

- `init_algorithm` can now accept a configuration as a file or a dict
- `tmax` was renamed `Nth_legendre` in configuration files
- `TauU` and `TauE` are now stored as `timedelta64`
- Default value for `-m` was set to 20 (when calling `run_algo.py`)
- Added Continuous Release

https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/pygeodyn/blob/master/CHANGELOG.md

Versioning : CR process

- Parse RELEASE.md that describe changes and release type
- Increase version number in _version.py
- Add changes in CHANGELOG.md
- Stage changes with git and add a tag with the new version number

Based on : https://hypothesis.works/articles/continuous-releases/