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

CHALLENGING PROBLEM:

- Thermal coupling
- Fluid mechanics
- Electromagnetism
Tackling the geodynamo problem

Numerical simulations
fast rotating fluid mechanics with turbulence

Physical measurements
only magnetic field at the Earth’s surface and its evolution in time...

Schaeffer et al. GJI 2017
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

\[ B = - \nabla V \]

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

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

Usually, we have \( N=14 \rightarrow 224 \) coefficients at each timestep

Adapted from Rotating spheric al harmonics.gif, Wikimedia Commons, CC BY-SA 3.0
Physical quantities to store

For each timestep (forecast or analysis):

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

\[ \dot{b} = A(b) u + e_r \]
Output array shapes

- Forecasts, for each quantity (MF, SV, U, ER):
  
  NumPy array of shape

  \[(nb\_model\_realisations, nb\_forecasts, nb\_coefficients)\]

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

- Analyses, for each quantity (MF, SV, U, ER):
  
  NumPy array of shape

  \[(nb\_model\_realisations, nb\_analyses, nb\_coefficients)\]

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

File

Forecast

Analysed

Computed

(nb_realisations, nb_forecasts, nb_coefficients) (nb_realisations, nb_analyses, nb_coefficients) (nb_realisations, nb_timesteps, nb_coefficients)

MF  SV  ER  U

MF  SV  ER  U

MF  SV  ER  U

: Group

: Dataset
Now with HDF5

- **Forecast**
  - MF
  - SV
  - ER
  - U
  - (nb_realisations, nb_forecasts, nb_coefficients)

- **Analysed**
  - MF
  - SV
  - ER
  - U
  - (nb_realisations, nb_analyses, nb_coefficients)

- **Computed**
  - MF
  - SV
  - ER
  - U
  - (nb_realisations, nb_timesteps, nb_coefficients)

- **File**
  - Configuration

- **Group**
  - Dataset

- **Attributes**
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 assimilation

- HDF5 to structure output data and embed computation parameters
  - Tool towards reproducible science/research
HDF5 for geomagnetic data visualisation
HDF5 in the *geodyn* suite

- **pygeodyn**
  - Python package for geomagnetic data assimilation

- **HDF5 file**
  - Python package for geomagnetic data visualisation
    - Web-based tool using a Tornado server
    - Available on PyPI
    - Deployed on
      - [https://geodyn.univ-grenoble-alpes.fr/](https://geodyn.univ-grenoble-alpes.fr/)

**HDF5 (h5py) allows to load directly and efficiently the datasets (faster than ASCII)**
Webgeodyn webpage

This website provides visualisation tools for geomagnetic observations and models. Some models include surface core flows that were inverted directly from geomagnetic data, or from their interpolation through Gauss coefficients, using the radial induction equation at the core mantle boundary (CMB) (see the data assimilation Python package pygeodyn for more details). The visualisation tool described below can also be used locally with the webgeodyn package (also on PyPI).

- **Streamfunctions at the CMB, superimposed with scalar fields**
  - August 2015: Orthographic projection
  - August 2016: Orthographic projection

- **Time-position maps of scalar fields (including geostrophic flow)**
  - (with respect to longitude, latitude or curvature coordinates)

- **Time series of surface core flow spherical harmonics coefficients**

- **Comparison of core flow predictions observatories**
  - (ground-based observatory series and virtual observatory series)

- **Mauersberger-Lowes spectra of the core quantities**
  - for time-average, deviation or at a given date
Visualisation examples

- Time-series of Gauss coefficients

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

- Geomagnetic data assimilation

- 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/pygeodynd

- Git repository of webgeodynd:
  
  https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/webgeodynd

- Article presenting pygeodynd (and a bit of webgeodynd...):
  
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/pygeodyd
https://gricad-gitlab.univ-grenoble-alpes.fr/Geodynamo/webgeodyd
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 _GOV.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 (pygeodynam_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 p0 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/