Global Geomagnetic Field Modeling with Python - **pymaginverse**

Utrecht University

Frenk Out[1], Maximilian Schanner[2,3], Liz van Grinsven[1], Monika Korte[3], and Lennart V. de Groot[1]

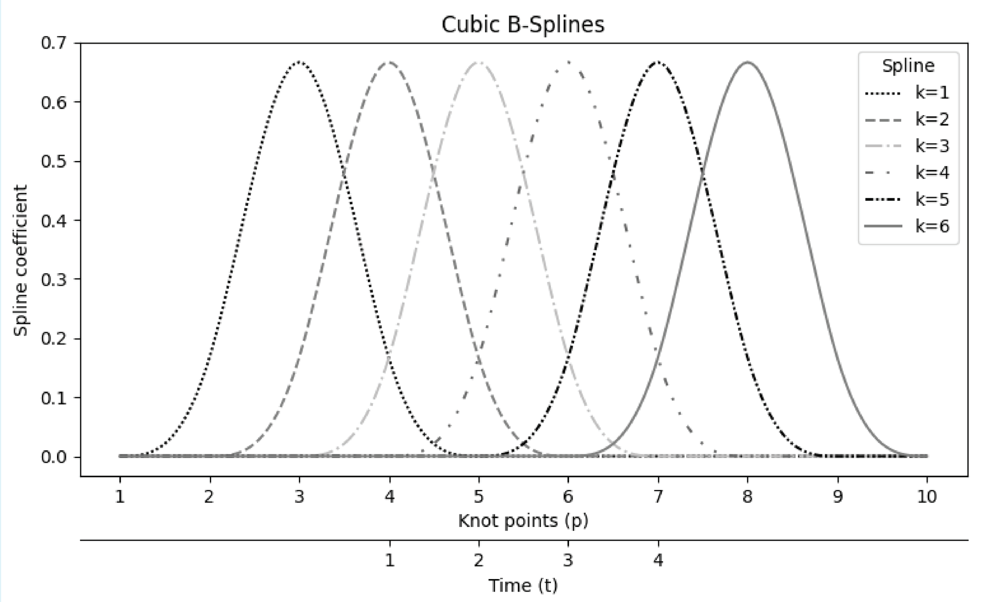
NWO

Introduction & Motivation

- Regularized least squares models (RLS) are used in geomagnetic field modeling since the 1980's, written in **Fortran**[a,b,c,d]
- **Fortran's lack of error clarity, documentation, and version control leads to maintenance challenges and compatibility issues.[e]**
- **We translated RLS models into a Python version with accompanying literature overview and published on github**
- **This Python version is, in terms of speed, on par with Fortran**
- RLS models require geomagnetic data and spatial and temporal smoothness constraints to invert for Gauss coefficients (g_1^0, g_1^1, ...)
- RLS models are used for modeling historical-to-millennial scale magnetic fields[b,e,f], snapshots thereof[g], and reversals[h]

Methods

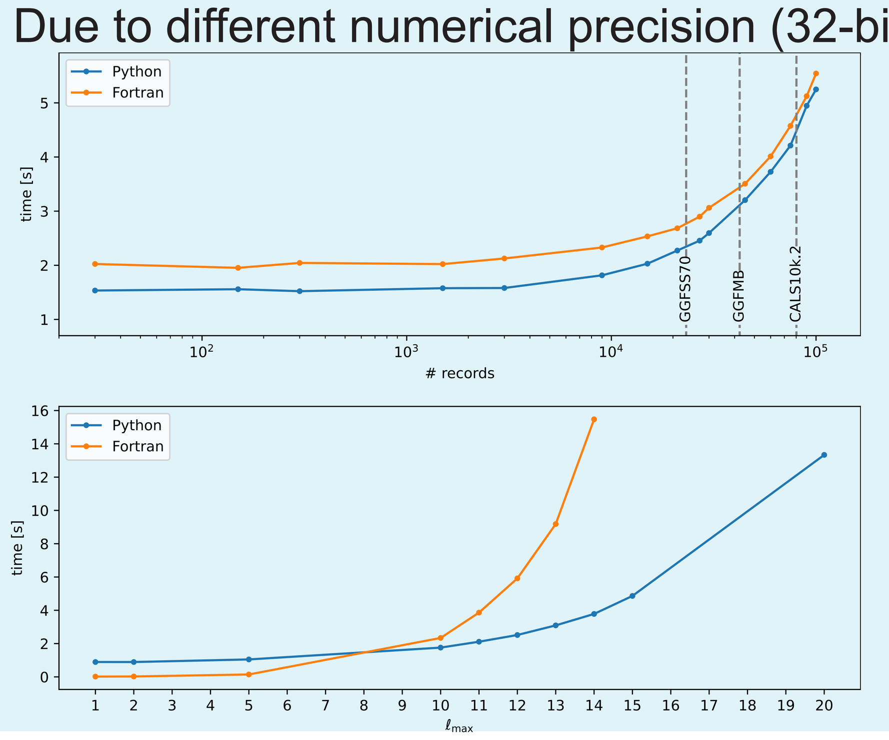
The algorithm is based on solving Laplace's equation in spherical coordinates using the source free assumption. Time-dependence is introduced with B-Splines[i,j]. The algorithm employs an iterative scheme to use non-linear data, where 8 different damping types can be imposed. However, damping parameters have lost physical basis, so model uncertainties are to be interpreted relatively. We lifted heavy computational weight with Cython and changed the structure of the algorithm to attain a further speed-up.



Benchmark

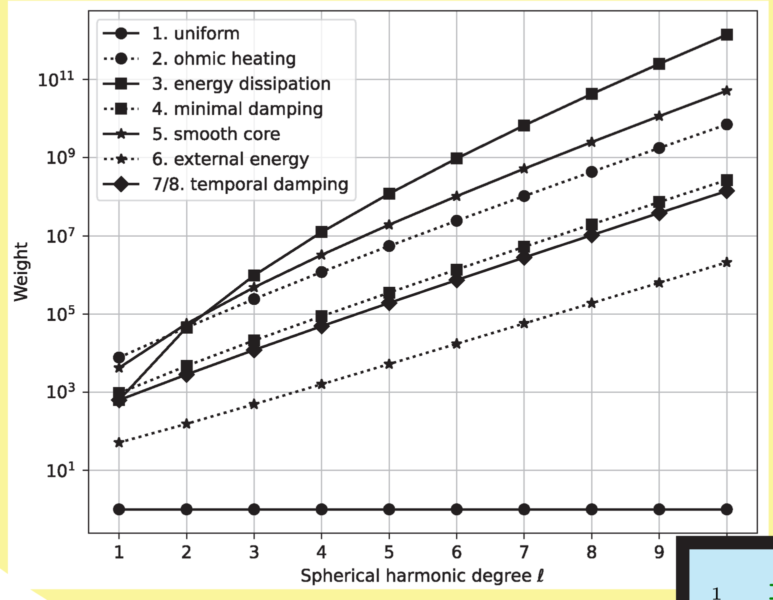
To ensure consistency we compared this Python algorithm to the Fortran version. In our benchmark we employed non-linear data randomly generated using pymagglobal[m]. This benchmark shows deviations for Python less than 17 nT with a mean of 0.3 nT from the Fortran version.

> Due to different numerical precision (32-bit vs 64-bit).



Damping

Constrain fast oscillations and large overswingings of model through **spatial and temporal damping**



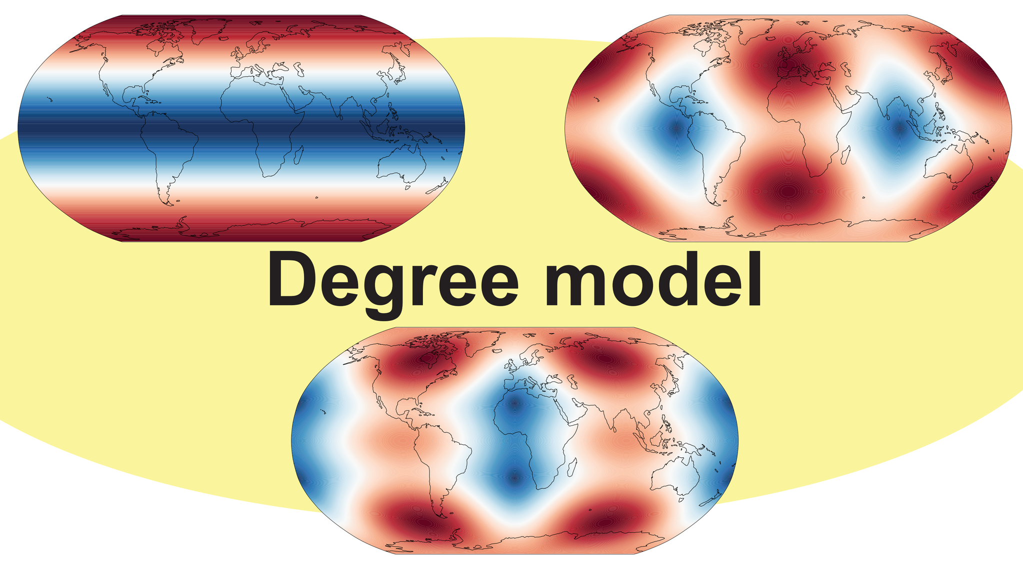
Tutorials included to demonstrate the capabilities of the algorithm

Input

Data
GEOMAGIA[k,l] or own CSV dataset

```
>>>pip install pymaginverse
```

Data → **Time knots** → **Damping parameters** → **Degree model**



Degree model

Code

```python
import numpy as np
import pandas as pd
from pathlib import Path
from pymaginverse import InputData, FieldInversion

# load csv-file into InputData class
path = Path().absolute()
dataset = pd.read_csv(path / 'example_data.csv', index_col=0)
inputdata = InputData(dataset)

##### start geomagnetic field inversion #####
# set time array and maximum spherical degree
test_inv = FieldInversion(t_min=-2000, t_max=1990, t_step=10, maxdegree=10)
# load data-class and set damping types: Ohmic heating and minimum acceleration
test_inv.prepare_inversion(inputdata, spat_type='ohmic_heating',
                           temp_type='min_acc')
# set starting model, should have 120 elements
x0 = np.zeros(test_inv._nr_coeffs)
x0[0] = -30000
# run inversion by setting start model, damp factors, and max # iterations
test_inv.run_inversion(x0, spat_damp=1.0e-13, temp_damp=1.0e-3, max_iter=5)
# save gauss coefficients results
test_inv.save_coefficients(path / 'output', file_name='example')
```

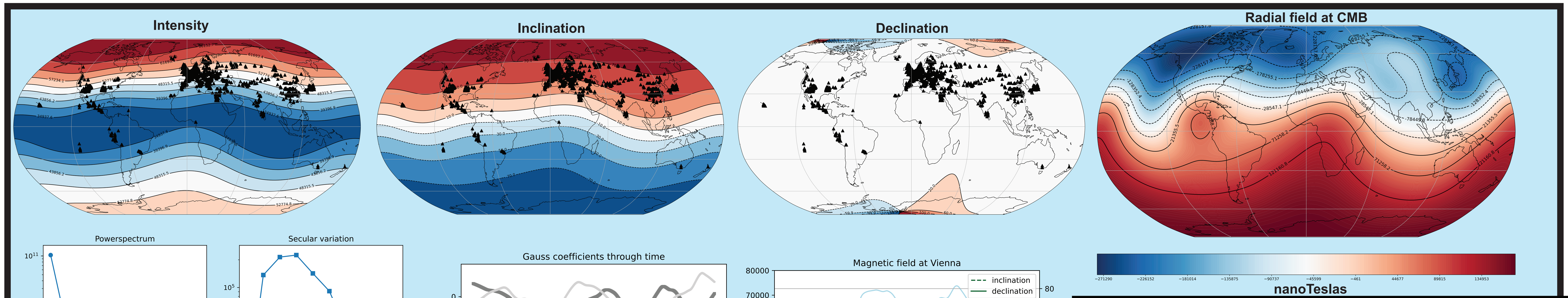
LOAD DATA

INITIATE CLASS

RUN MODEL

Save results in Fortran, pymagglobal[m], or CSV-format

Output



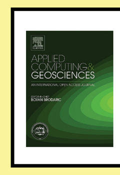
References

[a] Shure et al. (1982). Physics of the Earth and Planetary Interiors 28.
[b] Bloxham & Jackson (1992). Journal of Geophysical Research 97.
[c] Holme & Bloxham (1996). Journal of Geophysical Research: Planets 101.
[d] Jackson et al. (2000). Phil. Trans. of the Royal Society of London. Series A: Math., Phys. and Eng. Sc. 358.
[e] Korte et al. (2009). Geochemistry, Geophysics, Geosystems 10.
[f] Finlay et al. (2020). Earth, Planets and Space 72.
[g] Constable et al. (2000). Phil. Trans. of the Royal Society of London. Series A: Math., Phys. and Eng. Sc. 358.
[h] Mahgoub et al. (2023a). Journal of Geophysical Research: Solid Earth 128.
[i] Constable & Parker (1988). Journal of Computational Physics 78.
[j] De Boor (1978). springer-verlag New York volume 27.
[k] Brown et al. (2015a). Earth, Planets and Space 67.
[l] Brown et al. (2015b). Earth, Planets and Space 67
[m] Schanner et al. (2020). pymagglobal-Python interface for global geomagnetic field models

[1]Paleomagnetic laboratory Fort Hoofddijk, Utrecht University, NL
[2]Institute of applied mathematics, Potsdam University, De
[3]Helmholtz Centre Potsdam, Deutsches GeoForschungsZentrum GFZ, De

email:
f.out@uu.nl

Link to github pymaginverse



Link to manuscript