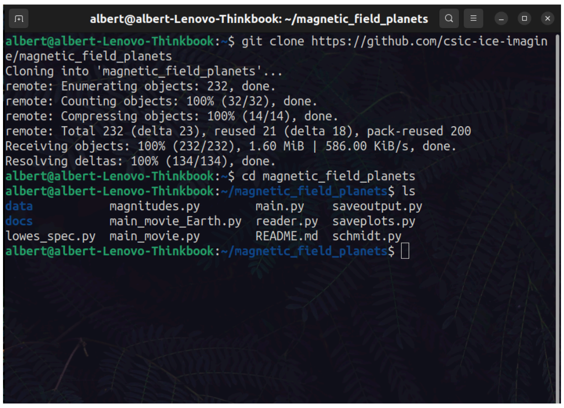
Hands-on session
Planetary magnetic field measurements

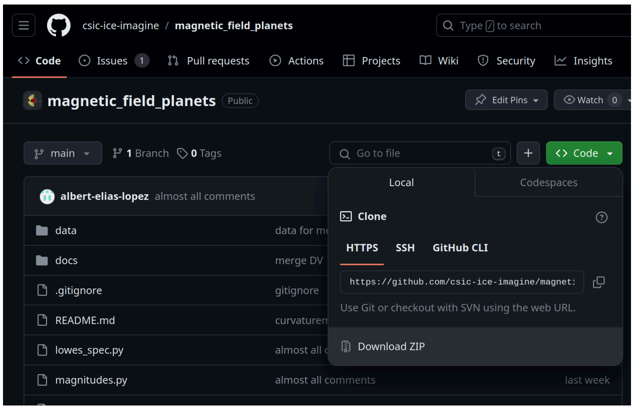
Public repository

0. Have python 3 installed

1. Download the repository (either clone from git or download and decompress directly on the web page)

    git clone https://github.com/csic-ice-imagine/magnetic_field_planets





2. Look around the directory. You will only need to use main.py, as it calls the other functions defined in the other files:

lowes_spec.py	Calculates and plot Lowes spectra
magnitudes.py	Calculates and plot the curl, divergence and curvature of **B**
reader.py	Reads the constants defined in tables
saveplots.py	Defines and save the plots
schmidt.py	Recursively calculates constants, Schmidt polynomials and **B**
saveoutput.py	Saves output for 3D visualisation

main_movie.py and main_movie_Earth.py are versions of main.py to recursively plot in radius and Earth data, respectively. All data tables are located in data/, and some pdfs with all the formulae used are located in docs/.

3. Open main.py.

You will only need to play with the 50ish first lines. Things that can be changed:

- Latitude-longitude resolution
- Radius (in corresponding planetary radii units)
- Save plots in plane/Mollweide projections
- Save Lowes spectrum
- Plot curl, divergence, and curvature

Increasing resolution will exponentially increase the computational time. To run the code you will only need to do:

    python main.py

```python
#---------------------------------------------------------------------------
# Grid resolution
Ntheta = 50       # Latitudinal points (North-South direction)
Nphi = 2*Ntheta   # Longitudial points (East-West direction)
Nr = 1            # Radial points (change only to generate 3D output)

# Radius considered in the map plot, and name of the corresponding images
# This should be the actual radius in kilometers (6371.2/72492 for
# Earth/Jupiter), but we renormalize to 1, since r/a is what matters.
rc = 1.00

# String used for naming the output files
rc_file = '%.2f'%rc
rc_file = rc_file.replace(".","_")

# Planet (or satellite) to choose. Raw data is located in folder data/
planet, year = "My_own", 2020
# You can choose either Earth, Jupiter, Jupiter_2021, Saturn, Neptune, Uranus,
# Mercury and Ganymede or My_own. Anything else will make the code stop.  If
# you choose Earth, you also need to choose a year, which can only be: 1900,
# 1905, 1910, ..., to 2020.

# Definition of the spherical grid matrices
phi    = np.linspace(0, 2*np.pi, num=Nphi)
theta  = np.linspace(np.pi / Ntheta, np.pi * (1 - 1 /  Ntheta), num=Ntheta)
# To calculate curvature/curl it is recommended to use a fixed value
# theta  = np.linspace(np.pi / 20, np.pi * (1 - 1 / 20), num=Ntheta)
# to avoid doing operations too close to the axis.

#---------------------------------------------------------------------------
# Switches to save projections in plane and Mollweide projections. Coastlines
#  are included in Earth plots.
planeproj, mollweideproj = True, True
# If you have successfully installed the ccrs library you can put the Earth
# coastline in the Earth plane projections also, using the boolean ccrs_library
ccrs_library = True

# ATTENTION: To plot using the Mollweide projection you need the ccrs library.
# The combination mollweideproj=True, ccrs_library=False will crash if you have
# not installed this library

# Switch to save the Lowes spectrum for the given radius
lowes = True
# Switch to save the Lowes spectrum for a number of radii
multiple_lowes_r, lowes_radii = False, np.array([1.45,1.30,1.15,1.00,0.85,0.70,0.55])
# Switch to plot the curl, divergence and curvature of the magnetic field
plot_magnitudes = False
```

4. Before playing with the code, try to install cartopy to enable for the option for Mollweide projection and coastlines (https://scitools.org.uk/cartopy/docs/latest/installing.html). Ideally, these commands should be enough:

```
pip install cartopy
```

or

```
conda install -c conda-forge cartopy
```

In my Ubuntu 22.04 laptop, I had to fight a little...

```
sudo apt-get install libproj-dev proj-data proj-bin
sudo apt-get install libgeos-dev
sudo pip install cython
sudo pip install cartopy
```

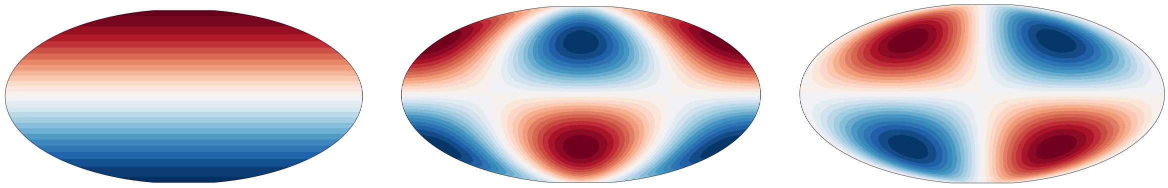
If it does not work, do not worry. Your plots will only be a square projection of the sphere (not as aesthetic). In this case you will have to set `ccrs_library = False` and `mollweideproj = False`.

Q1

Up to which multipole degree (l) do each magnetic field models have? Look in each file in data/

Q2

Using the *"My_own"* option and changing the file in data/my_own_planet.txt play with some multipoles (change some 0's to 1's) to recover plots like g_{10}, g_{21}, or h_{21}, respectively. You should mostly loot at the radial field direction.



Find which multipole (g_{nm} or h_{nm}) creates this Br plot (next image only in the slides).

Q3

You can start running main.py, use `lowes = True`, to save spectral plots. Use some small resolution (at most N_θ=100) to plot all planets available and see the differences. Be aware that if you choose Earth you can specify which year you want (from 1900 to 2020 every 5 years). Put True or False:

- Earth has an almost constant magnetic field modulus throughout its surface.
- Earth magnetic inclination has a nearly perfect horizon with 0º tilt.
- Saturn's magnetic field is aligned with the rotation axis.
- Ganymede's magnetic field seems to be very different from the other planets.
- Uranus' magnetic field is aligned with its rotation axis.
- Neptune's magnetic field is aligned with its rotation axis.
- Jupiter's magnetic field is measured more accurately than Earth's.
- At the dynamo surface Earth's is better measured than any other planet.
- Mercury's magnetic field is stronger than Earth's.

Q4

Use the main_movie_Earth.py to produce the 5-year frequency images. You can play with the resolution and the radius. Try some other radius other than 1 (not less than 0.5). Is the magnetic field static? Towards which direction does it shift to?

Q5

Use the main_movie.py for some planets to produce different plots at different radii. Which are the radii that correspond to a flat magnetic spectrum? Why are we not able to find the same for planets other than Earth and Jupiter?

Possible errors:

1. In some versions of python there is a label that gives problems. On line 47 of saveplots.py, for some this works just fine:

   ```
   '$B_θ$ (Gauss) at $r =$' + str(rc) + '$R_P$',
   ```

   If you have that issue, try:

   ```
   '$B_{\theta}$ (Gauss) at $r =$' + str(rc) + '$R_P$',
   ```

2. Even if you install cartopy correctly, there might still be some bug with ax.coastline(). If you cannot solve this you should comment out lines 94, 139, 224 and 286. You will only not be able to see the Earth continents in the maps.

If you encounter any other errors and how to avoid them, please contact me.