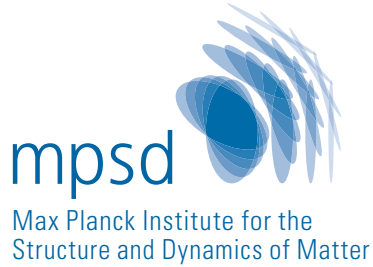
EOSC Jupyter Vision (Fair with Jupyter)



Hans Fangohr

Max Planck Institute for the Structure and Dynamics of Matter
Hamburg, Germany

University of Southampton, Southampton, UK

2022-11-29 Grenoble (France)

hans.fangohr@mpsd.mpg.de
https://fangohr.github.io
@ProfCompMod@fosstodon.org

Outline

- Jupyter Notebook for Science
- FAIR data vision for EOSC
- Binder
- Binder for FAIR data

Jupyter Notebook

- interactive document
- hosted in web browser
- combines text, source code, code output and images

In [1]:
```
import matplotlib.pyplot as plt
import numpy as np
```

Code cells show both code input and output:

In [2]:
```
6*8 - 3*2
```

Out[2]: 42

Markdown cells such as this one can contain text and $\LaTeX$ equations such as $c(a, b) = \sqrt{a^2 + b^2}$. We can use code to define the corresponding functions:

In [3]:
```
def c(a, b):
    return (a**2 + b**2)**0.5
```

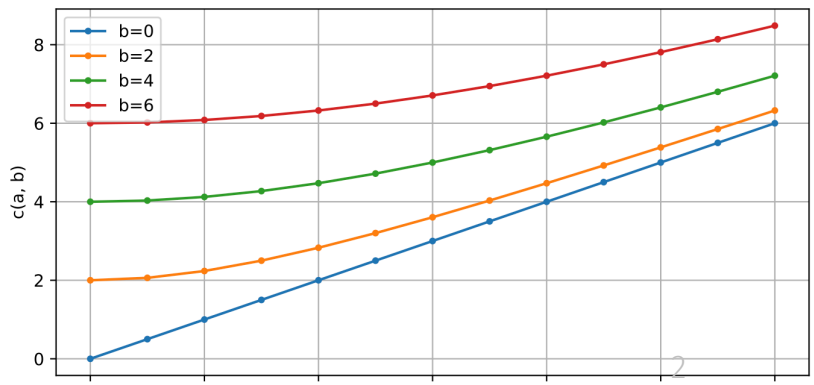
Let us compute `c(a, b)` as a function of `a` and `b` and plot multiple lines (each for a fixed `b`).

In [4]:
```
a = np.linspace(0, 6, 13)

plt.figure(figsize=(8, 4))

for b in [0, 2, 4, 6]:
    c_res = c(a, b)
    plt.plot(a, c_res, '.-', label='b='+str(b))

plt.xlabel('a')
plt.ylabel('c(a, b)')
plt.grid()
plt.legend();
```



Why Jupyter notebooks for science?

"Jupyter: Thinking and Storytelling With Code and Data"

Granger and Pérez,
10.1109/MCSE.2021.3059263 (2021)

"Using Jupyter for reproducible scientific workflows"

Beg, Kluyver, Ragan-Kelly, Fangohr *et al*
10.1109/MCSE.2021.3052101 (2021)

"Data exploration and analysis with Jupyter notebooks"

Fangohr, Kluyver, PaNOSC team *et al*
10.18429/JACoW-ICALEPCS2019-TUCPR02 (2019)

Jupyter notebooks for Photon and Neutron Science [1] (2019)

17th Int. Conf. on Acc. and Large Exp. Physics Control Systems ICALEPCS2019, New York, NY, USA JACoW Publishing
ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-TUCPR02

DATA EXPLORATION AND ANALYSIS WITH Jupyter NOTEBOOKS

H. Fangohr*[1], M. Beg, M. Bergemann, V. Bondar, S. Brockhauser[2,3], C. Carinan, R. Costa, F. Dall'Antonia, C. Danilevski, J. C. E, W. Ehsan, S. G. Esenov, R. Fabbri, S. Fangohr, G. Flucke, C. Fortmann[4], D. Fulla Marsa, G. Giovanetti, D. Goeries, S. Hauf, D. G. Hickin, T. Jarosiewicz[5], E. Kamil, M. Karnevskiy, Y. Kirienko, A. Klimovskaia, T. A. Kluyver, M. Kuster, L. Le Guyader, A. Madsen, L. G. Maia, D. Mamchyk, L. Mercadier, T. Michelat, J. Möller, I. Mohacsi, A. Parenti, M. Reiser, R. Rosca, D. B. Rueck, T. Rüter, H. Santos, R. Schaffer, A. Scherz, M. Scholz, A. Silenzi, M. Spirzewski[5], J. Sztuk, J. Szuba, S. Trojanowski[5], K. Wrona, A. A. Yaroslavtsev, J. Zhu
European XFEL GmbH, Schenefeld, Germany
J. Reppin, F. Schlünzen, M. Schuh, DESY, Hamburg, Germany
E. Fernandez-del-Castillo, G. Sipos, EGI Foundation, Amserdam, Netherlands
T. H. Rod, J. R. Selknaes, J. W. Taylor, ESS, Copenhagen, Denmark
A. Campbell, A. Götz, J. Kieffer, ESRF, Grenoble, France
J. Hall, E. Pellegrini, J. F. Perrin, ILL, Grenoble, France

[1] also at University of Southampton, Southampton, United Kingdom
[2] also at University of Szeged, Szeged, Hungary
[3] also at Biological Research Center of the Hungarian Academy of Sciences, Szeged, Hungary
[4] also at Max-Planck-Inst. for Evolutionary Biology, Plön, Germany
[5] also at NCBJ, Otwock, Poland

Abstract

Jupyter notebooks are executable documents that are displayed in a web browser. The notebook elements consist of human-authored contextual elements and computer code, what was done, and a record of the code used and the results, to establish confidence and to serve as a base for further work.

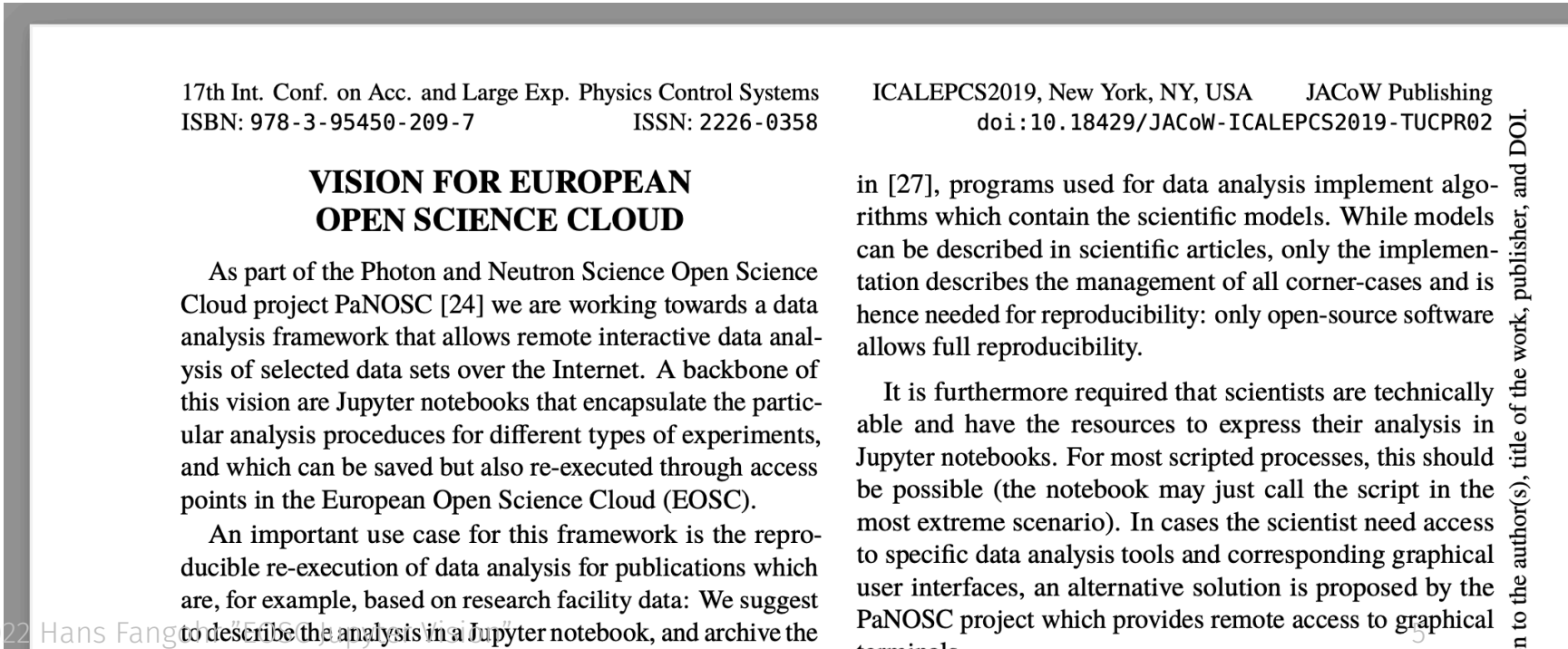
Digital notebook interfaces are an increasingly popular

Jupyter notebooks for Photon and Neutron Science [1]

- driving data analysis from notebook
- collection of notebook recipes for typical tasks
- notebook as a script (detector calibration)
- drive simulation studies from notebook
- documenting software libraries
- JupyterHub and computational environments (remote X11)

Jupyter notebooks for Photon and Neutron Science [1]

- driving data analysis from notebook
- collection of notebook recipes for typical tasks
- notebook as a script (detector calibration)
- drive simulation studies from notebook
- documenting software libraries
- JupyterHub and computational environments (remote X11)
-



17th Int. Conf. on Acc. and Large Exp. Physics Control Systems ICALEPCS2019, New York, NY, USA JACoW Publishing
ISBN: 978-3-95450-209-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2019-TUCPR02

VISION FOR EUROPEAN OPEN SCIENCE CLOUD

As part of the Photon and Neutron Science Open Science Cloud project PaNOSC [24] we are working towards a data analysis framework that allows remote interactive data analysis of selected data sets over the Internet. A backbone of this vision are Jupyter notebooks that encapsulate the particular analysis proceduces for different types of experiments, and which can be saved but also re-executed through access points in the European Open Science Cloud (EOSC).

An important use case for this framework is the reproducible re-execution of data analysis for publications which are, for example, based on research facility data: We suggest to describe the analysis in a Jupyter notebook, and archive the

in [27], programs used for data analysis implement algorithms which contain the scientific models. While models can be described in scientific articles, only the implementation describes the management of all corner-cases and is hence needed for reproducibility: only open-source software allows full reproducibility.

It is furthermore required that scientists are technically able and have the resources to express their analysis in Jupyter notebooks. For most scripted processes, this should be possible (the notebook may just call the script in the most extreme scenario). In cases the scientist need access to specific data analysis tools and corresponding graphical user interfaces, an alternative solution is proposed by the PaNOSC project which provides remote access to graphical

Vision for European Open Science Cloud

EOSC Jupyter Vision [1]

- a *data analysis framework* that allows *remote interactive data analysis* of selected data sets over the Internet.
- *Jupyter notebooks* that *encapsulate* the particular analysis *procedures* and which
- can be *re-executed* through *access points* in the European Open Science Cloud (EOSC).

---

[1] *Data exploration and analysis with Jupyter*, 10.18429/JACoW-ICALEPCS2019-TUCPR02

Vision for European Open Science Cloud [1]

Use case 1: reproducible publications

- describe the analysis in a Jupyter notebook
- archive the notebook with required software as metadata
- together with (raw or preprocessed) publication data

---

[1] *Data exploration and analysis with Jupyter*, 10.18429/JACoW-ICALEPCS2019-TUCPR02

Example for use case 1: reproducible publications

Nanotechnology **27** (2016) 455502

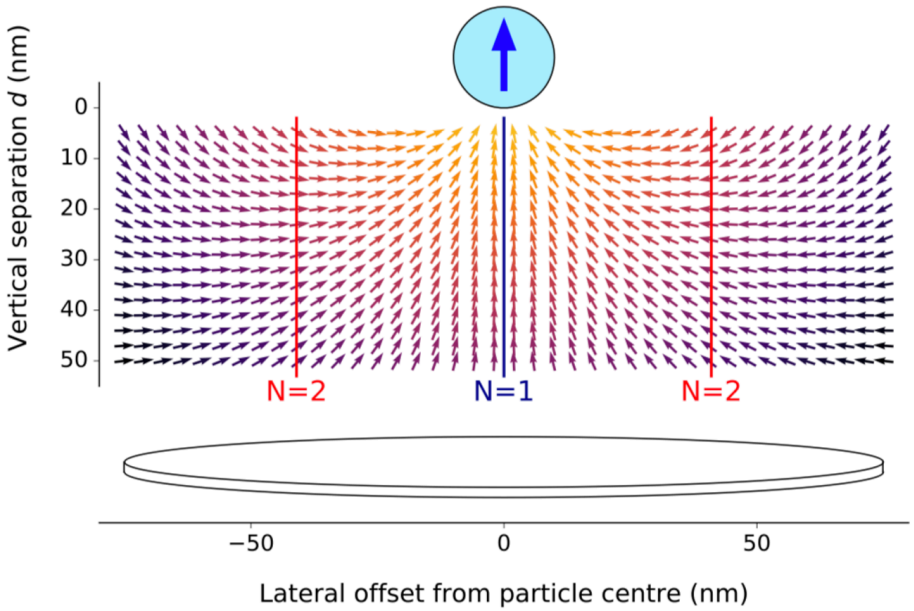
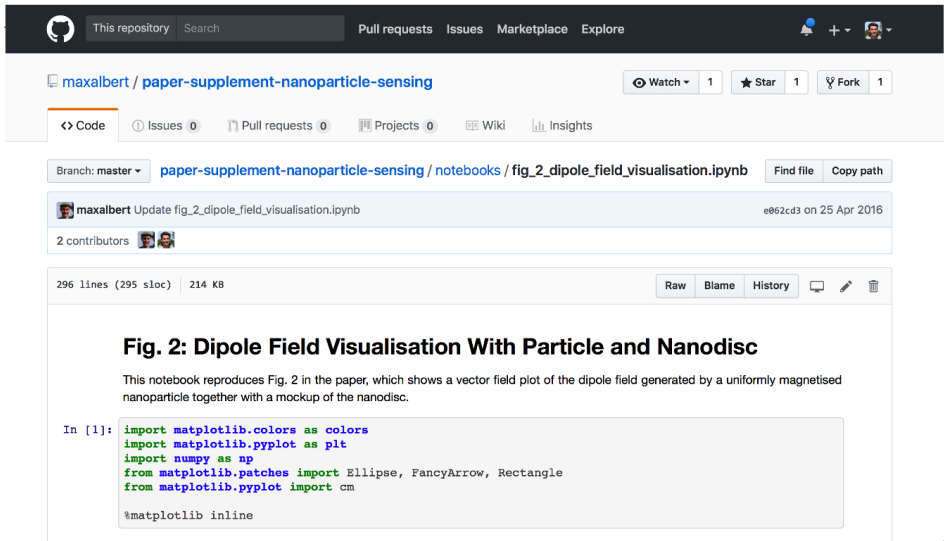
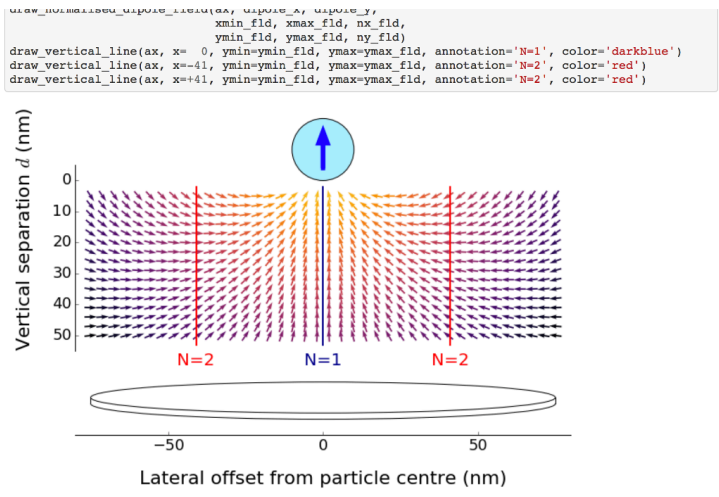


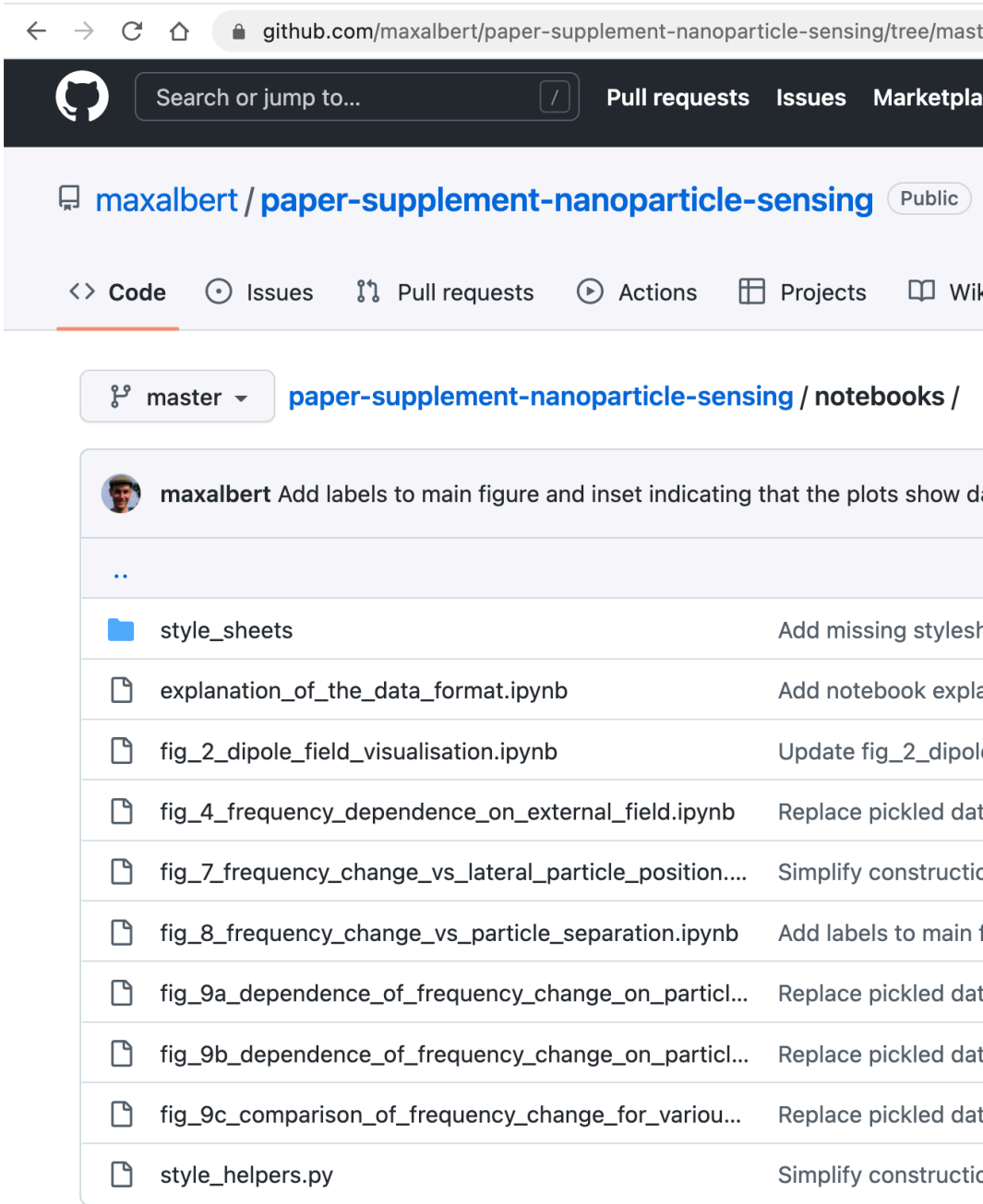
Figure 2. Vector field plot of the dipole field generated by a uniformly $+z$-magnetised MNP. The vectors are scaled to uniform length with their colour indicating the field strength (orange is high and violet/black is low). The vertical lines correspond to the x-values where modes 1 and 2 have maxima in their spin precession amplitude (see figures 3(a)–(b)). A schematic of the nanodisc is shown at the bottom.





Example for use case 1: reproducible publications

- publish repository to complement manuscript, containing
- one notebook per figure / main result
- Zenodo for long term preservation



`https://github.com/maxalbert/paper-supplement-nanoparticle-sensing`
`https://doi.org/10.5281/zenodo.60605`

Binder software and project

Binder needs:

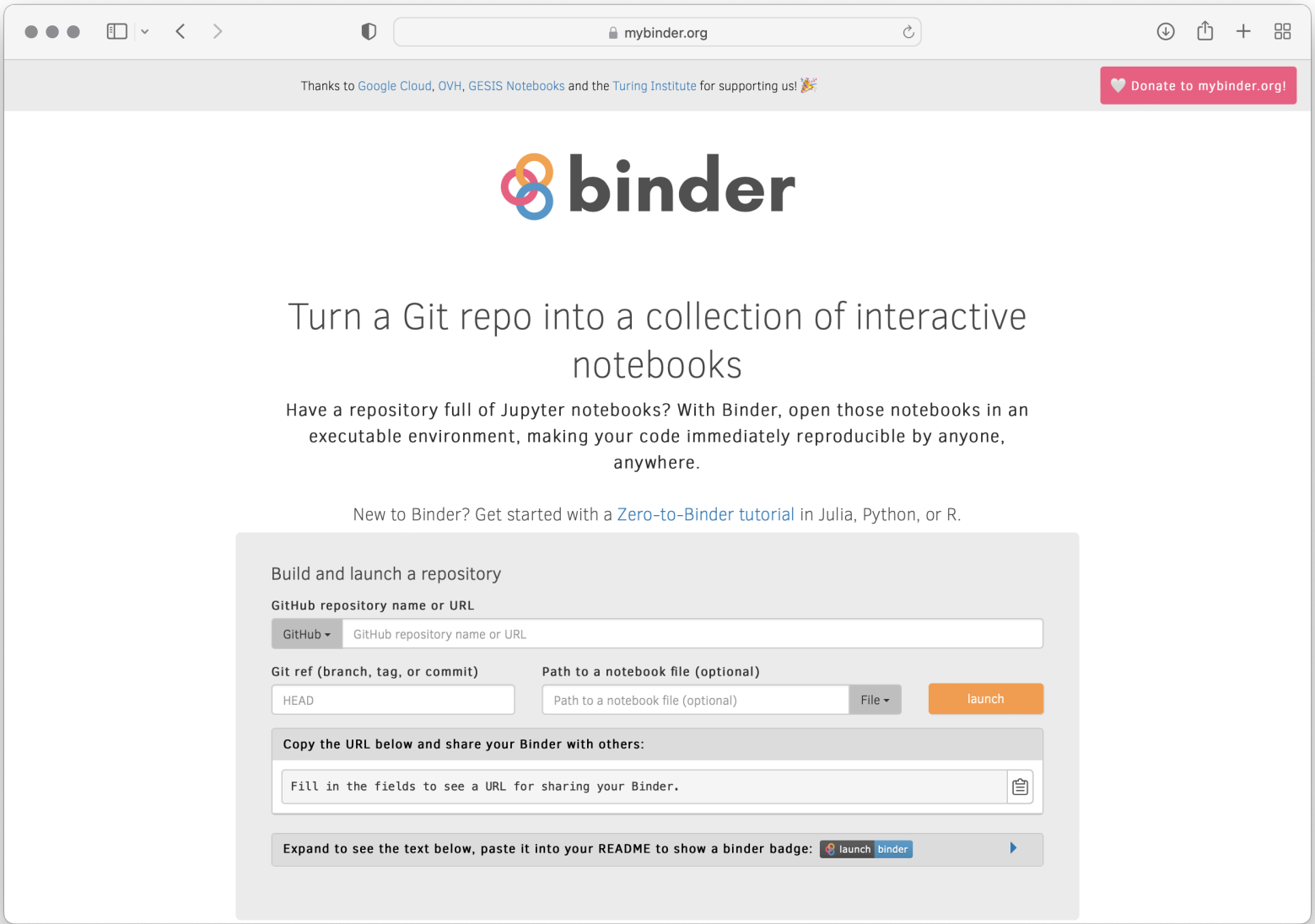
- data repository (e.g. git repository, Zenodo, Figshare, DataVerse)
- notebooks to execute
- software requirements (e.g. `requirements.txt`)

Usage:

- Given the URL of the repository, Binder builds a (Docker) container to provide the software
- Starts Jupyter notebook server inside that container
- and connects to user's browser

- Binder is part of Project Jupyter `https://github.com/jupyterhub/binder`
- Example: `https://github.com/fangohr/reproducibility-repository-example`

mybinder.org service (a public Binder instance)



Public service “MyBinder”: `http://mybinder.org`

Software specification makes repo ”binder-enabled”

master paper-supplement-nanoparticle-sensing / environment.yml

maxalbert Eliminate explicit dependency on brewer2mpl (we now hard-code the col... ...

1 contributor

11 lines (11 sloc) | 158 Bytes

```
1  name: particle-sensing
2  dependencies:
3    - python
4    - future
5    - ipython
6    - jupyter
7    - matplotlib>=1.5
8    - numexpr
9    - numpy
10   - pandas>=0.18
11   - statsmodels
```

- software requirements are recorded in repository:
  - `environment.yml` for conda
  - `requirements.txt` for python
  - `install.R` for R
  - More options at `https://repo2docker.readthedocs.io/en/latest/config_files.html`

Use cases for Binder-enabled repositories

- reproducibility
  - Beg et.al., *Using Jupyter for reproducible scientific workflows*, 10.1109/MCSE.2021.3052101 (2021)
- zero-install software provision (only web browser needed)
  - interactive documentation
  - workshops
  - test-drive software
    - Example: `https://ubermag.github.io` → "Try in your browser"
- data access
  - deposit data together with software to read data

EOSC Jupyter Vision [1]

Use case 1: reproducible publications

- describe the *analysis* in Jupyter notebook
- archive the notebook with required software
- together with *publication* data

Use case 2: data access

- describe the *data access* in Jupyter notebook
  - *perhaps include common analysis examples*
- archive the notebook with required software
- together with *experiment* data

---

[1] *Data exploration and analysis with Jupyter*, 10.18429/JACoW-ICALEPCS2019-TUCPR02

EOSC Jupyter Vision [1]

Use case 1: reproducible publications

- describe the *analysis* in Jupyter notebook
- archive the notebook with required software
- together with *publication* data

Use case 2: data access

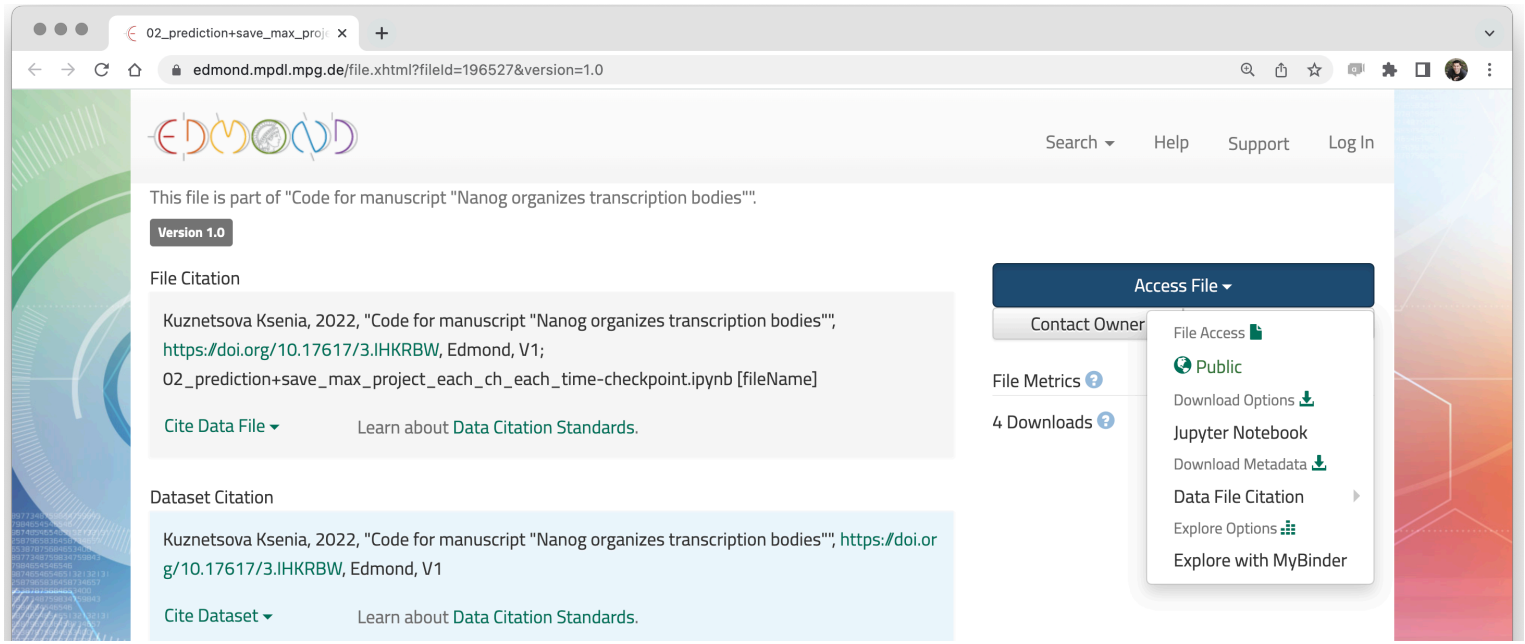
- describe the *data access* in Jupyter notebook
  - *perhaps include common analysis examples*
- archive the notebook with required software
- together with *experiment* data

- Use cases are similar.

---

[1] *Data exploration and analysis with Jupyter*, 10.18429/JACoW-ICALEPCS2019-TUCPR02

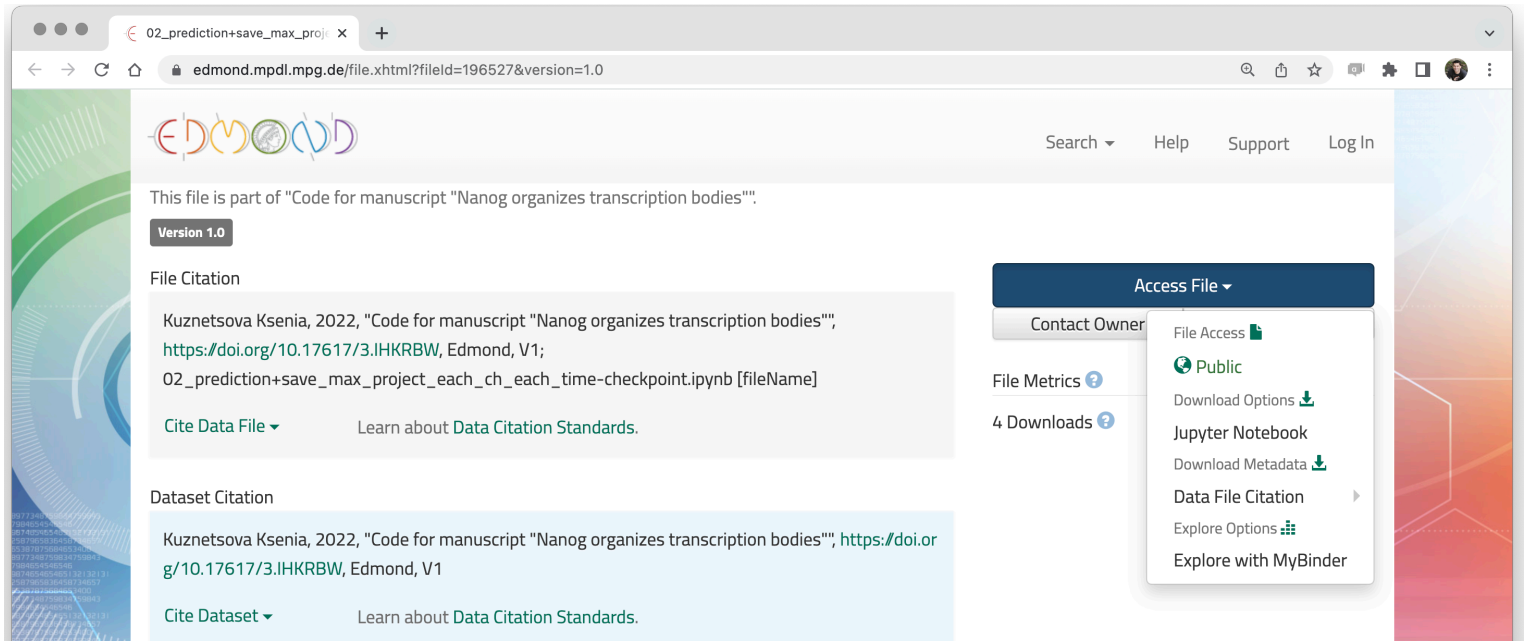
Example for use case 2: Data Access through Binder



- Realised for 50 institutions[1] through Dataverse software including Harvard, Johns Hopkins, Max Planck Society.
- Workflow: the whole data set is copied from the (DataVerse) source to the Binder-container when the binder session starts

[1] https://github.com/jupyterhub/repo2docker/blob/main/repo2docker/contentproviders/dataverse.json

Example for use case 2: Data Access through Binder



- Realised for 50 institutions[1] through Dataverse software including Harvard, Johns Hopkins, Max Planck Society.
- Workflow: the whole data set is copied from the (DataVerse) source to the Binder-container when the binder session starts
- → not practical for larger data sets

[1] https://github.com/jupyterhub/repo2docker/blob/main/repo2docker/contentproviders/dataverse.json

Challenges for EOSC Jupyter vision 1/2

Binder-enabled data access and analysis with Notebooks:

- 1. need analysis content/data access software in repository and notebooks
- 2. need to execute notebooks in the correct computational environment
  - what software is needed
  - what versions
  - how should it be compiled

Challenges for EOSC Jupyter vision 1/2

Binder-enabled data access and analysis with Notebooks:

- 1. need analysis content/data access software in repository and notebooks
  - user behaviour $\rightarrow$ ✓
- 2. need to execute notebooks in the correct computational environment
  - what software is needed
  - what versions
  - how should it be compiled

Challenges for EOSC Jupyter vision 1/2

Binder-enabled data access and analysis with Notebooks:

- 1. need analysis content/data access software in repository and notebooks
  - user behaviour → ✓
- 2. need to execute notebooks in the correct computational environment
  - what software is needed
  - what versions
  - how should it be compiled
  - Binder software specifications → ✓

Challenges for EOSC Jupyter vision 2/2

Binder-enabled data access and analysis with Notebooks:

- 3. for *small data* sets:
  - can use `mybinder.org` (perhaps with added EOSC resources?)
  - transfer whole data set when container is created

- 3. for *large data* sets we need either:
  - transparent data access to files at remote location

  or
  - Need `BinderHub` instances close to (large) data sets

[2] https://github.com/jupyterhub/repo2docker/tree/main/repo2docker/contentproviders

[3] See T4.3 in https://github.com/minrk/horizon-widera-2022/blob/main/submitted-SOURCE-2022-04-20.pdf

Challenges for EOSC Jupyter vision 2/2

Binder-enabled data access and analysis with Notebooks:

- 3. for *small data* sets:
  - can use `mybinder.org` (perhaps with added EOSC resources?)
  - transfer whole data set when container is created
  - working for `providers`[2] defined in `binder` → ✓
- 3. for *large data* sets we need either:
  - transparent data access to files at remote location

  or
  - Need `BinderHub` instances close to (large) data sets

[2] https://github.com/jupyterhub/repo2docker/tree/main/repo2docker/contentproviders

[3] See T4.3 in https://github.com/minrk/horizon-widera-2022/blob/main/submitted-SOURCE-2022-04-20.pdf

Challenges for EOSC Jupyter vision 2/2

Binder-enabled data access and analysis with Notebooks:

- 3. for *small data* sets:
  - can use `mybinder.org` (perhaps with added EOSC resources?)
  - transfer whole data set when container is created
  - working for `providers`[2] defined in `binder` → ✓
- 3. for *large data* sets we need either:
  - transparent data access to files at remote location
    - protocol and location identifier unclear [3] → ?

  or
  - Need `BinderHub` instances close to (large) data sets

[2] https://github.com/jupyterhub/repo2docker/tree/main/repo2docker/contentproviders

[3] See T4.3 in https://github.com/minrk/horizon-widera-2022/blob/main/submitted-SOURCE-2022-04-20.pdf

Challenges for EOSC Jupyter vision 2/2

Binder-enabled data access and analysis with Notebooks:

- 3. for *small data* sets:
  - can use **`mybinder.org`** (perhaps with added EOSC resources?)
  - transfer whole data set when container is created
  - working for **`providers`**[2] defined in **`binder`** → ✓
- 3. for *large data* sets we need either:
  - transparent data access to files at remote location
    - protocol and location identifier unclear [3] → ?

  or
  - Need **`BinderHub`** instances close to (large) data sets
    - provided by data hosting organisations → ? (✓)

[2] https://github.com/jupyterhub/repo2docker/tree/main/repo2docker/contentproviders

[3] See T4.3 in https://github.com/minrk/horizon-widera-2022/blob/main/submitted-SOURCE-2022-04-20.pdf

FAIR with project Jupyter

Findable

Accessible

- Technology: Browser sufficient for access ✓

Interoperable

- If kernel(=language) used in the notebook is acceptable ✓

Re-usable (and reproducible)

- Software environment provides immediate executability ✓

Summary

EOSC Jupyter Vision [1]

- make data available *together* with software to read data
- provide example/documentation/analysis notebooks
- provide access through Binder instance

Does not formally enforce presence of metadata but code embeds some of it.

Blockers (for large data sets)

- either need data set location and remote access/protocol
- or need Binder instances close to data

---

[1] *Data exploration and analysis with Jupyter*,
`https://doi.org/10.18429/JACoW-ICALEPCS2019-TUCPR02`

Acknowledgements

We acknowledge support from

- Photon and Neutron Open Science Cloud (PaNOSC) project (#823852), `https://www.panosc.eu/`
- OpenDreamKit Horizon 2020 European Research Infrastructures project (#676541), `https://opendreamkit.org/`
- Binder team