PySDM:

exploring novel tools from the Python ecosystem for super-droplet simulation studies

Piotr Bartman, Sylwester Arabas

Jagiellonian University

University of Hyogo | August 23, 2019

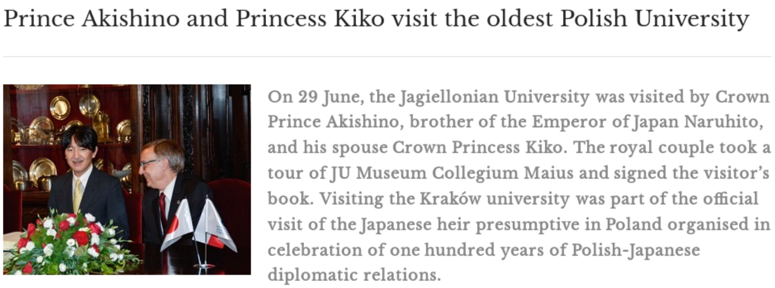
Jagiellonian University, Kraków, Poland

North Sea
SWEDEN
DEN.
Baltic Sea
LAT.
LITH.
RUS.
BELARUS
POLAND
GERMANY
Kraków
UKRAINE
CZECH
SLVK.
AUSTRIA
HUNGARY
ROMANIA

- founded in 1364 (貞治３年)
- among 20 oldest continuously operating univ. in the world
- ca. 40 000 students, 7000 staff (4000 acad.), 16 faculties
- Oriental Studies since 1919, Japanese Studies since 1987

Jagiellonian University, Kraków, Poland

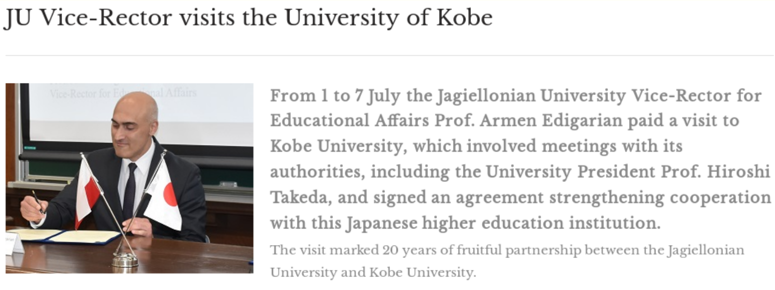
June 2019



Prince Akishino and Princess Kiko visit the oldest Polish University

On 29 June, the Jagiellonian University was visited by Crown Prince Akishino, brother of the Emperor of Japan Naruhito, and his spouse Crown Princess Kiko. The royal couple took a tour of JU Museum Collegium Maius and signed the visitor's book. Visiting the Kraków university was part of the official visit of the Japanese heir presumptive in Poland organised in celebration of one hundred years of Polish-Japanese diplomatic relations.

July 2019



JU Vice-Rector visits the University of Kobe

From 1 to 7 July the Jagiellonian University Vice-Rector for Educational Affairs Prof. Armen Edigarian paid a visit to Kobe University, which involved meetings with its authorities, including the University President Prof. Hiroshi Takeda, and signed an agreement strengthening cooperation with this Japanese higher education institution.

The visit marked 20 years of fruitful partnership between the Jagiellonian University and Kobe University.

motivation

journal/common sense/scientific method requirements

new 2019 GMD journal policy, doi:10.5194/gmd-12-2215-2019

journal/common sense/scientific method requirements

new 2019 GMD journal policy, doi:10.5194/gmd-12-2215-2019

- *„everything required to run the experiment must be provided, apart from the model itself"*

journal/common sense/scientific method requirements

new 2019 GMD journal policy, doi:10.5194/gmd-12-2215-2019

- *„everything required to run the experiment must be provided, apart from the model itself"*

- *„ensure that there is no manual processing of the data: models are run by a script, and all pre- and post-processing is scripted"*

journal/common sense/scientific method requirements

new 2019 GMD journal policy, doi:10.5194/gmd-12-2215-2019

- *„everything required to run the experiment must be provided, apart from the model itself"*

- *„ensure that there is no manual processing of the data: models are run by a script, and all pre- and post-processing is scripted"*

- *„All figures and tables must be scientifically reproducible from the scripts"*

journal/common sense/scientific method requirements

new 2019 GMD journal policy, doi:10.5194/gmd-12-2215-2019

- *„everything required to run the experiment must be provided, apart from the model itself"*
- *„ensure that there is no manual processing of the data: models are run by a script, and all pre- and post-processing is scripted"*
- *„All figures and tables must be scientifically reproducible from the scripts"*
- *„It is the opinion of the GMD editors that if the code is not ready, then neither is the manuscript"*

journal/common sense/scientific method requirements

new 2019 GMD journal policy, doi:10.5194/gmd-12-2215-2019

- *„everything required to run the experiment must be provided, apart from the model itself"*
- *„ensure that there is no manual processing of the data: models are run by a script, and all pre- and post-processing is scripted"*
- *„All figures and tables must be scientifically reproducible from the scripts"*
- *„It is the opinion of the GMD editors that if the code is not ready, then neither is the manuscript"*
- *„During the review process, the ease of model download, compilation, and running of test cases may be assessed"*

programmers' job market

Nature 2019 "toolbox" column (on Julia), doi:10.1038/d41586-019-02310-3

- *„Among climate scientists, the lingua franca is Fortran: speedy, but — with roots dating to the 1950s — not terribly exciting"*
- *„Younger programmers prefer languages that can accommodate the latest trends in software and hardware design"*
- *„'two-language problem' — researchers often prototype algorithms in a user-friendly language such as Python but then have to rewrite them in a faster language ..."*

foci of this talk

Python ecosystem examples to address (simultaneously):

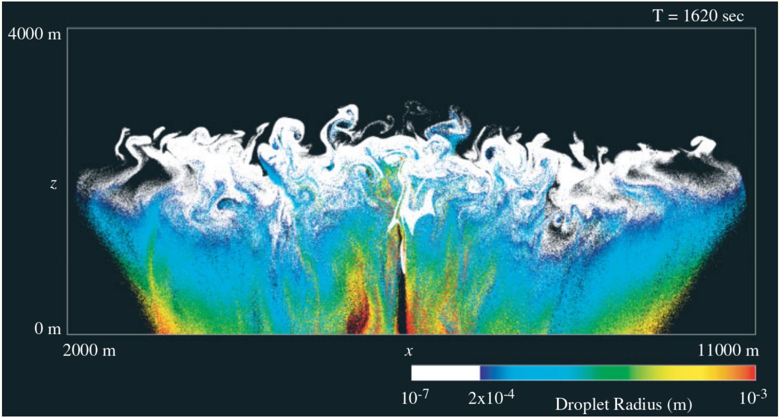
- code readibility ⇝ maintainability
- performance (incl. ability to leverage modern hardware)
- automation (incl. analysis and figure generation)
- dissemination (incl. ease of setting up the environment)

scientific context

Aerosol-cloud-precipitation interactions

"Cloud and ship. Ukraine, Crimea, Black sea, view from Ai-Petri mountain"
(photo: Yevgen Timashov / National Geographic)

Probabilistic particle-based simulations



Super-droplet simulation of a shallow convective cloud
(figure: Shima et al. 2009, QJRMS)

SDM: Monte-Carlo scheme of Shima et al. 2009

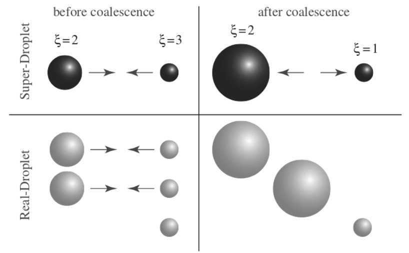
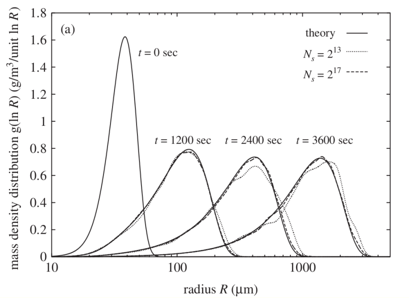
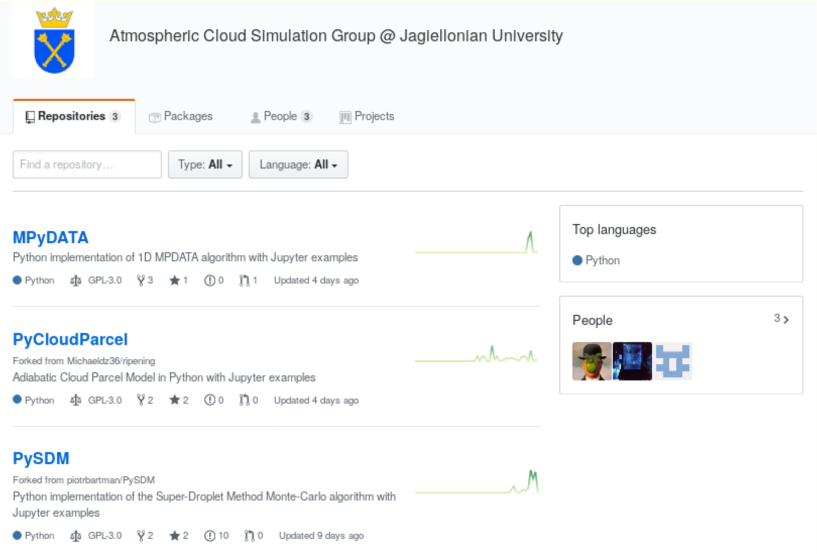


Figure 1. Schematic view of the coalescence of super-droplets. Two super-droplets with multiplicity 2 and 3 undergo coalescence (upper left panel). This represents the coalescence of two droplet pairs (lower panels). As a result, the super-droplet with multiplicity 2 becomes larger and the multiplicity of the other super-droplet decreases 3 → 1 (upper right panel).



PySDM

github.com/atmos-cloud-sim-uj



Atmospheric Cloud Simulation Group @ Jagiellonian University

MPyDATA

Python implementation of 1D MPDATA algorithm with Jupyter examples

● Python GPL-3.0 3 ★ 1 0 1 Updated 4 days ago

PyCloudParcel

Forked from Michaeldz36/ripening

Adiabatic Cloud Parcel Model in Python with Jupyter examples

● Python GPL-3.0 2 ★ 2 0 0 Updated 4 days ago

PySDM

Forked from piotrbartman/PySDM

Python implementation of the Super-Droplet Method Monte-Carlo algorithm with Jupyter examples

● Python GPL-3.0 2 ★ 2 10 0 Updated 9 days ago

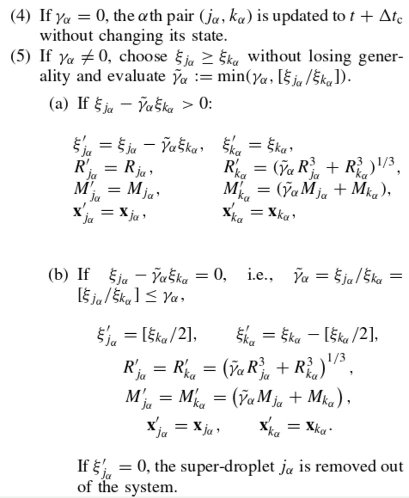
Top languages

● Python

People 3 ›

PySDM: sample code

maths



(4) If $\gamma_\alpha = 0$, the αth pair (j_α, k_α) is updated to $t + \Delta t_c$ without changing its state.

(5) If $\gamma_\alpha \neq 0$, choose $\xi_{j\alpha} \geq \xi_{k_\alpha}$ without losing generality and evaluate $\tilde{\gamma}_\alpha := \min(\gamma_\alpha, [\xi_{j_\alpha}/\xi_{k_\alpha}])$.

(a) If $\xi_{j_\alpha} - \tilde{\gamma}_\alpha \xi_{k_\alpha} > 0$:

$$\begin{aligned} \xi'_{j_\alpha} &= \xi_{j_\alpha} - \tilde{\gamma}_\alpha \xi_{k_\alpha}, & \xi'_{k_\alpha} &= \xi_{k_\alpha}, \\ R'_{j_\alpha} &= R_{j_\alpha}, & R'_{k_\alpha} &= (\tilde{\gamma}_\alpha R^3_{j_\alpha} + R^3_{k_\alpha})^{1/3}, \\ M'_{j_\alpha} &= M_{j_\alpha}, & M'_{k_\alpha} &= (\tilde{\gamma}_\alpha M_{j_\alpha} + M_{k_\alpha}), \\ \mathbf{x}'_{j_\alpha} &= \mathbf{x}_{j_\alpha}, & \mathbf{x}'_{k_\alpha} &= \mathbf{x}_{k_\alpha}, \end{aligned}$$

(b) If $\xi_{j_\alpha} - \tilde{\gamma}_\alpha \xi_{k_\alpha} = 0$, i.e., $\tilde{\gamma}_\alpha = \xi_{j_\alpha}/\xi_{k_\alpha} = [\xi_{j_\alpha}/\xi_{k_\alpha}] \leq \gamma_\alpha$,

$$\xi'_{j_\alpha} = [\xi_{k_\alpha}/2], \qquad \xi'_{k_\alpha} = \xi_{k_\alpha} - [\xi_{k_\alpha}/2],$$

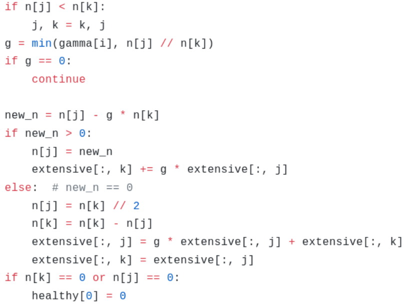
$$R'_{j_\alpha} = R'_{k_\alpha} = \left(\tilde{\gamma}_\alpha R^3_{j_\alpha} + R^3_{k_\alpha}\right)^{1/3},$$

$$M'_{j_\alpha} = M'_{k_\alpha} = \left(\tilde{\gamma}_\alpha M_{j_\alpha} + M_{k_\alpha}\right),$$

$$\mathbf{x}'_{j_\alpha} = \mathbf{x}_{j_\alpha}, \qquad \mathbf{x}'_{k_\alpha} = \mathbf{x}_{k_\alpha}.$$

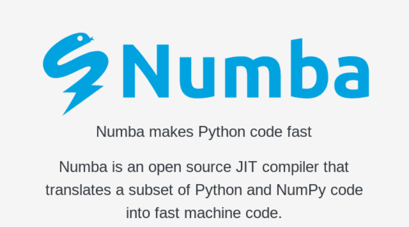
If $\xi'_{j_\alpha} = 0$, the super-droplet j_α is removed out of the system.

Python



```python
if n[j] < n[k]:
    j, k = k, j
g = min(gamma[i], n[j] // n[k])
if g == 0:
    continue

new_n = n[j] - g * n[k]
if new_n > 0:
    n[j] = new_n
    extensive[:, k] += g * extensive[:, j]
else:  # new_n == 0
    n[j] = n[k] // 2
    n[k] = n[k] - n[j]
    extensive[:, j] = g * extensive[:, j] + extensive[:, k]
    extensive[:, k] = extensive[:, j]
if n[k] == 0 or n[j] == 0:
    healthy[0] = 0
```

acceleration with Numba (JIT, multi-threading)

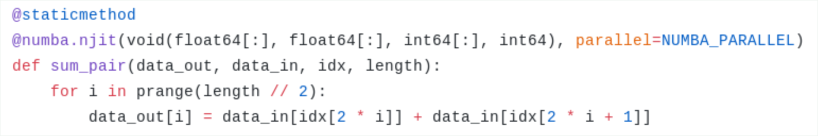


Numba makes Python code fast

Numba is an open source JIT compiler that translates a subset of Python and NumPy code into fast machine code.

https://numba.pydata.org/

PySDM/SDM/backends/numba.py

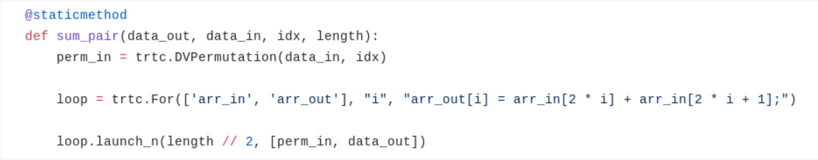


```
@staticmethod
@numba.njit(void(float64[:], float64[:], int64[:], int64), parallel=NUMBA_PARALLEL)
def sum_pair(data_out, data_in, idx, length):
    for i in prange(length // 2):
        data_out[i] = data_in[idx[2 * i]] + data_in[idx[2 * i + 1]]
```

acceleration with ThrustRTC and CURandRTC (GPU)

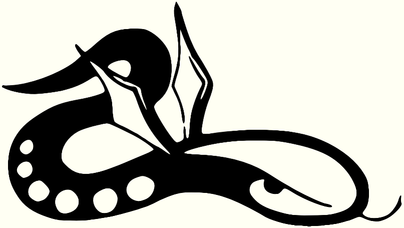
https://github.com/thrust/ (C++)
https://fynv.github.io/ (Python/C++/C#/Java)

PySDM/SDM/backends/thrustRTC.py



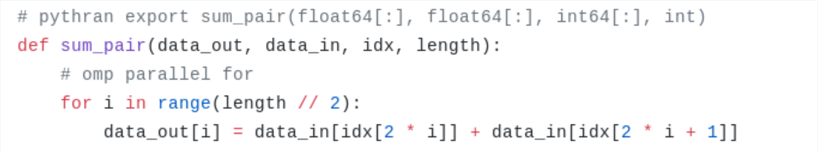
```
@staticmethod
def sum_pair(data_out, data_in, idx, length):
    perm_in = trtc.DVPermutation(data_in, idx)

    loop = trtc.For(['arr_in', 'arr_out'], "i", "arr_out[i] = arr_in[2 * i] + arr_in[2 * i + 1];")

    loop.launch_n(length // 2, [perm_in, data_out])
```

Pythran: ahead-of-time compilation, OpenMP



https://pythran.readthedocs.io/

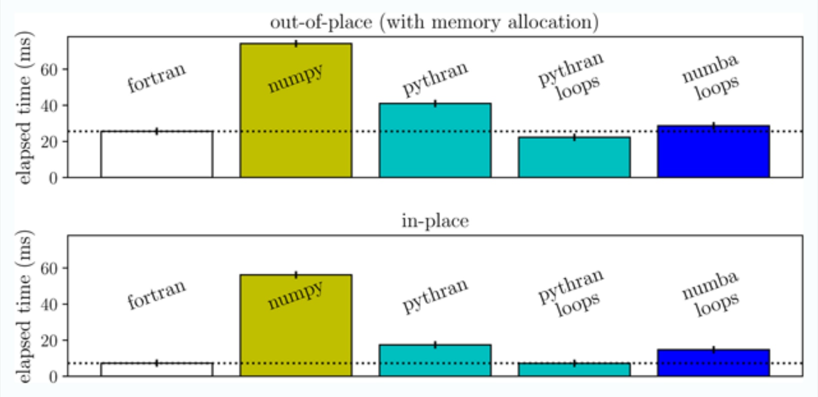
PySDM/SDM/backends/pythran.py



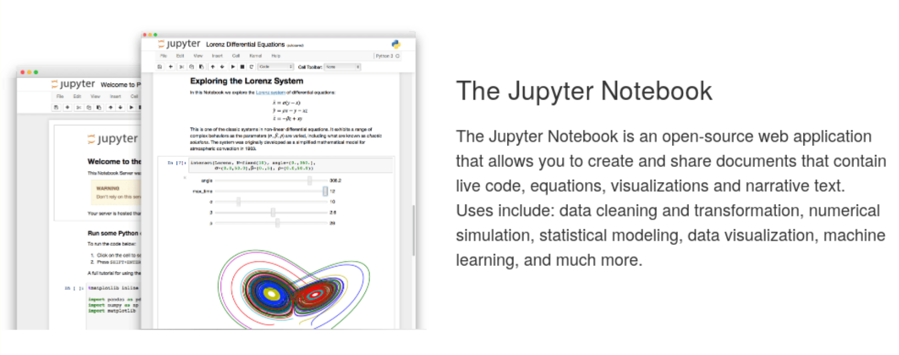
```
# pythran export sum_pair(float64[:], float64[:], int64[:], int)
def sum_pair(data_out, data_in, idx, length):
    # omp parallel for
    for i in range(length // 2):
        data_out[i] = data_in[idx[2 * i]] + data_in[idx[2 * i + 1]]
```

Pythran vs. Numba vs. Fortran ...

Mohanan et al. 2019, doi:10.5334/jors.238



example automation with Jupyter

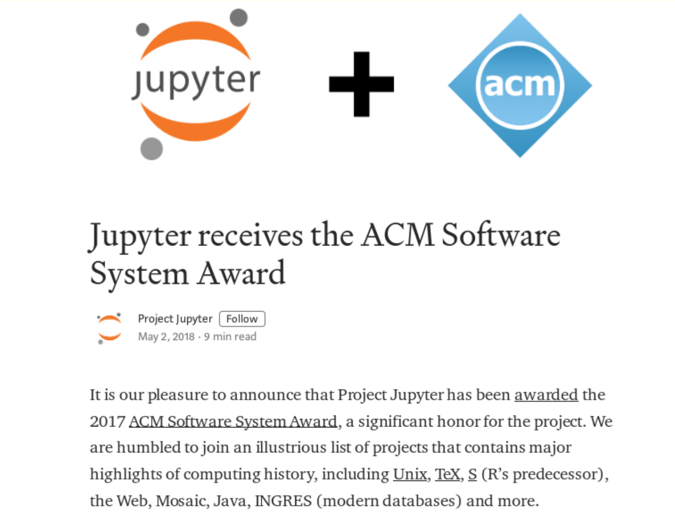


The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

https://jupyter.org/

example automation with Jupyter



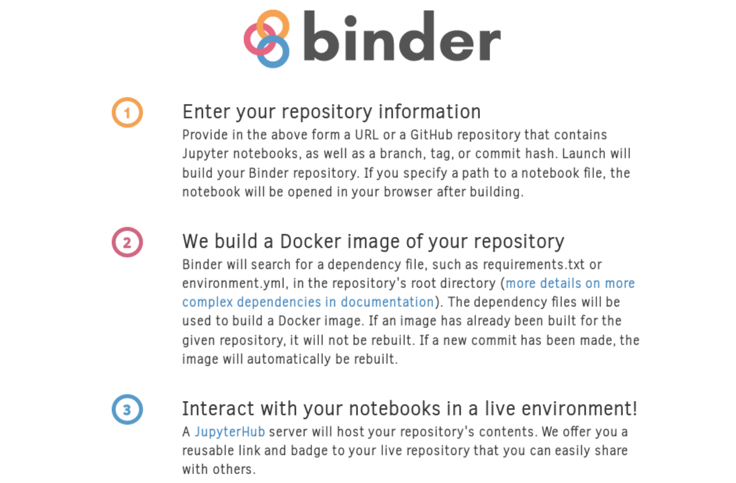
Jupyter receives the ACM Software System Award

Project Jupyter Follow
May 2, 2018 · 9 min read

It is our pleasure to announce that Project Jupyter has been awarded the 2017 ACM Software System Award, a significant honor for the project. We are humbled to join an illustrious list of projects that contains major highlights of computing history, including Unix, TeX, S (R's predecessor), the Web, Mosaic, Java, INGRES (modern databases) and more.

https://jupyter.org/

dissemination with github and mybinder.org



binder

1 **Enter your repository information**

Provide in the above form a URL or a GitHub repository that contains Jupyter notebooks, as well as a branch, tag, or commit hash. Launch will build your Binder repository. If you specify a path to a notebook file, the notebook will be opened in your browser after building.

2 **We build a Docker image of your repository**

Binder will search for a dependency file, such as requirements.txt or environment.yml, in the repository's root directory (more details on more complex dependencies in documentation). The dependency files will be used to build a Docker image. If an image has already been built for the given repository, it will not be rebuilt. If a new commit has been made, the image will automatically be rebuilt.

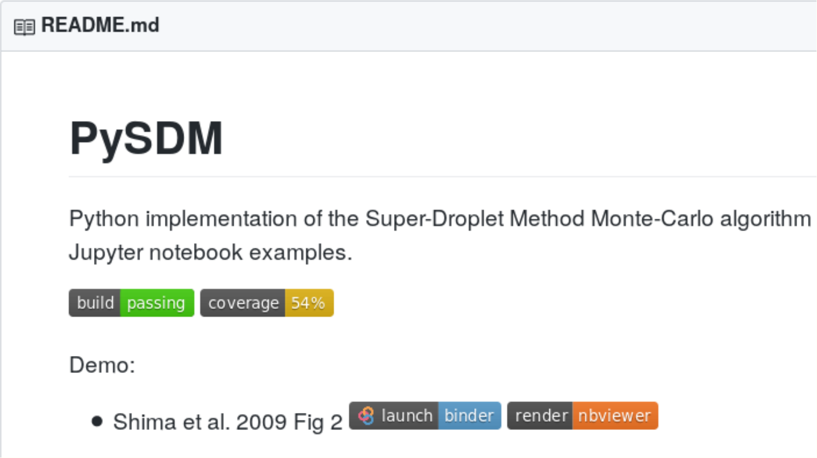
3 **Interact with your notebooks in a live environment!**

A JupyterHub server will host your repository's contents. We offer you a reusable link and badge to your live repository that you can easily share with others.

`https://mybinder.org`

demo

demo (github.com/atmos-cloud-sim-uj/PySDM)



README.md

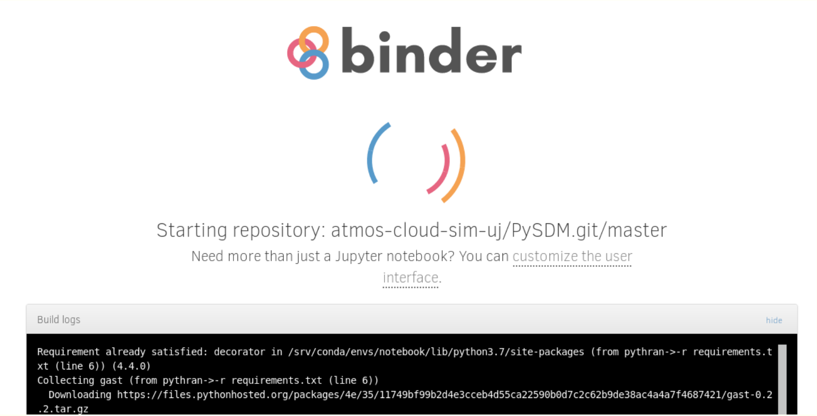
PySDM

Python implementation of the Super-Droplet Method Monte-Carlo algorithm
Jupyter notebook examples.

Demo:

- Shima et al. 2009 Fig 2

demo (mybinder.org)

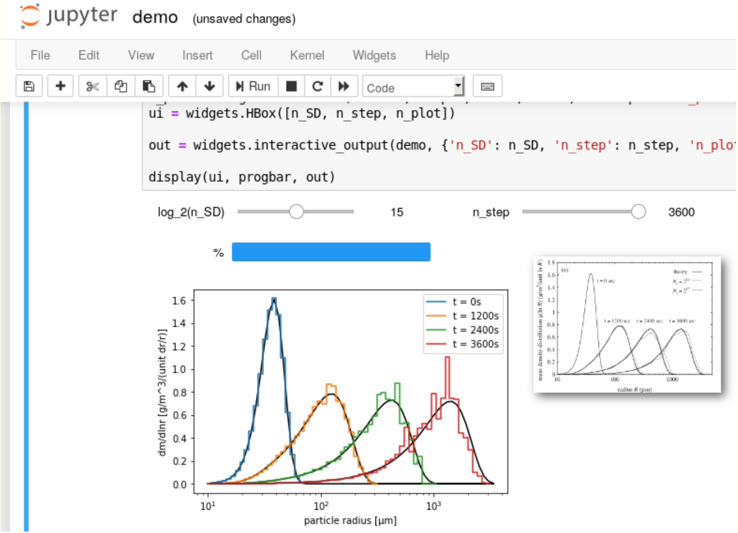


Starting repository: atmos-cloud-sim-uj/PySDM.git/master

Need more than just a Jupyter notebook? You can customize the user interface.

```
Requirement already satisfied: decorator in /srv/conda/envs/notebook/lib/python3.7/site-packages (from pythran->-r requirements.t
xt (line 6)) (4.4.0)
Collecting gast (from pythran->-r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/4e/35/11749bf99b2d4e3cceb4d55ca22590b0d7c2c62b9de38ac4a4a7f4687421/gast-0.2
.2.tar.gz
```

demo (ipywidgets)



```
ui = widgets.HBox([n_SD, n_step, n_plot])

out = widgets.interactive_output(demo, {'n_SD': n_SD, 'n_step': n_step, 'n_plo

display(ui, progbar, out)
```

log_2(n_SD) 15

n_step 3600

ご清聴ありがとうございました。

Thank you for your attention!

funding acknowledgement:
Foundation for Polish Science / European Union