

PySDM: a novel Pythonic tool for modelling atmospheric clouds with CPU and GPU number-crunching backends

Sylwester Arabas^{0,1}

seminarium Zakładu Zastosowań Metod Obliczeniowych, 2 VI 2022

⁰Mathematics and Computer Science, Jagiellonian University (atmos.ii.uj.edu.pl)

¹Atmospheric Sciences, University of Illinois at Urbana-Champaign (atmos.illinois.edu)

PySDM: a novel Pythonic tool for modelling atmospheric clouds with CPU and GPU number-crunching backends

Sylwester Arabas^{0,1}

co-authors, contributors & collaborators:

@**UJ**: P. Bartman (WMil), M. Olesik (WFAiS), G. Łazaski (WCh), O. Bulenok...

@**Caltech**: E. de Jong, C. Singer, A. Jaruga, B. Mackay, I. Dula, S. Azimi ...

@**UIUC**: N. Riemer, M. West & J. Curtis

⁰Mathematics and Computer Science, Jagiellonian University (atmos.ii.uj.edu.pl)

¹Atmospheric Sciences, University of Illinois at Urbana-Champaign (atmos.illinois.edu)

Sylwester Arabas

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)

- ▶ outside of academia:

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)
- ▶ outside of academia:
 - ▶ 2015–2017: Chatham Financial, Kraków

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)

- ▶ outside of academia:
 - ▶ 2015–2017: Chatham Financial, Kraków
 - ▶ 2017–2018: AETHON Engineering, Athens

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)

- ▶ outside of academia:
 - ▶ 2015–2017: Chatham Financial, Kraków
 - ▶ 2017–2018: AETHON Engineering, Athens

- ▶ back in academia:

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)

- ▶ outside of academia:
 - ▶ 2015–2017: Chatham Financial, Kraków
 - ▶ 2017–2018: AETHON Engineering, Athens

- ▶ back in academia:
 - ▶ 2018–2021: postdoc at WMil UJ (FNP „POWROTY”)

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)
- ▶ outside of academia:
 - ▶ 2015–2017: Chatham Financial, Kraków
 - ▶ 2017–2018: AETHON Engineering, Athens
- ▶ back in academia:
 - ▶ 2018–2021: postdoc at WMil UJ (FNP „POWROTY”)
 - ▶ 2021–2022: postdoc at U. Illinois Urbana-Champaign

Sylwester Arabas

- ▶ alma mater – fuw.edu.pl:
 - ▶ MSc (2008) in observational cloud physics
 - ▶ PhD (2013) in computational cloud physics
 - ▶ postdoc (till 2015): CFD software development (github.com/igfuw)

- ▶ outside of academia:
 - ▶ 2015–2017: Chatham Financial, Kraków
 - ▶ 2017–2018: AETHON Engineering, Athens

- ▶ back in academia:
 - ▶ 2018–2021: postdoc at WMił UJ (FNP „POWROTY”)
 - ▶ 2021–2022: postdoc at U. Illinois Urbana-Champaign
 - ▶ 2022–2024: NCN „SONATA” @ WMił UJ

context: aerosol-cloud-precipitation interactions (scales!)



“Cloud and ship. Ukraine, Crimea, Black sea, view from Ai-Petri mountain”

(photo: Yevgen Timashov / National Geographic)

context: aerosol-cloud-precipitation interactions (uncertainty!)



The screenshot shows the IPCC website homepage. At the top, there is a blue navigation bar with a menu icon on the left and links for ABOUT, DATA, DOCUMENTATION, FOCAL POINTS PORTAL, BUREAU PORTAL, LIBRARY, LINKS, LANGUAGES, and a search icon. Below this is a secondary navigation bar with the IPCC logo and links for REPORTS, SYNTHESIS REPORT, WORKING GROUPS, ACTIVITIES, NEWS, and CALENDAR. There are also icons for FOLLOW and SHARE. The main content area features a large title "The Intergovernmental Panel on Climate Change" and a descriptive paragraph: "The Intergovernmental Panel on Climate Change (IPCC) is the United Nations body for assessing the science related to climate change." At the bottom, there is a grey box for "WORKING GROUP II SIXTH ASSESSMENT REPORT" and logos for WHO, UNEP, and the Nobel Peace Prize.

MENU

ABOUT DATA DOCUMENTATION FOCAL POINTS PORTAL BUREAU PORTAL LIBRARY LINKS LANGUAGES SEARCH

ipcc

REPORTS SYNTHESIS REPORT WORKING GROUPS ACTIVITIES NEWS

CALENDAR

FOLLOW SHARE

The Intergovernmental Panel on Climate Change

The Intergovernmental Panel on Climate Change (IPCC) is the United Nations body for assessing the science related to climate change.

WORKING GROUP II SIXTH ASSESSMENT REPORT

WHO UNEP Nobel PEACE PRIZE OF THE NOBEL FOUNDATION

context: aerosol-cloud-precipitation interactions (uncertainty!)

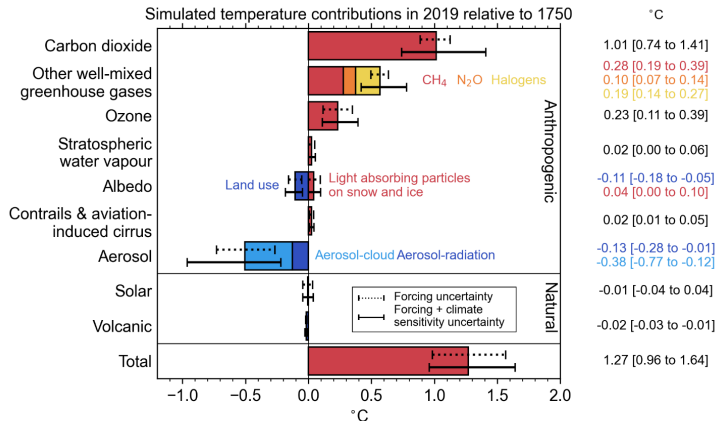
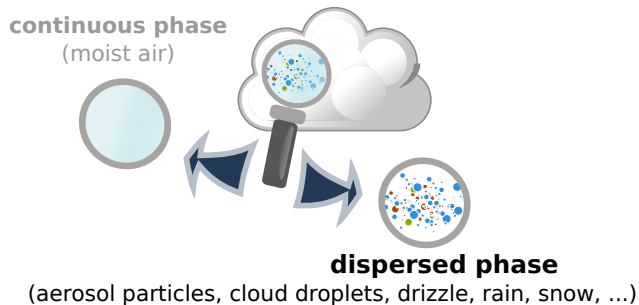


Figure 7.7: The contribution of forcing agents to 2019 temperature change relative to 1750 produced using the two-layer emulator (Supplementary Material 7.SM.2), constrained to assessed ranges for key climate metrics described in Cross-Chapter Box 7.1.

modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

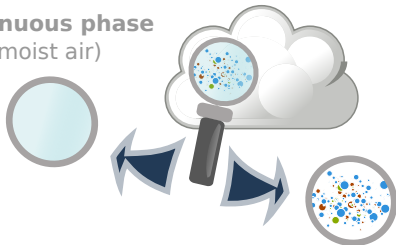


modelling cloud μ -physics: Eulerian vs. Lagrangian approaches



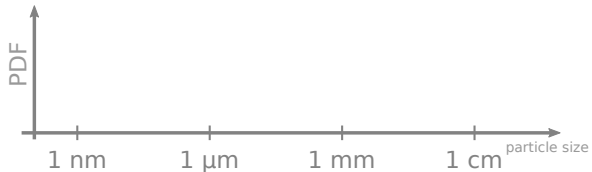
modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

continuous phase
(moist air)



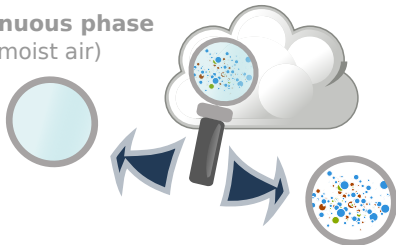
dispersed phase

(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



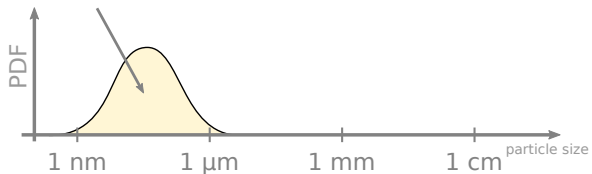
modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

continuous phase
(moist air)



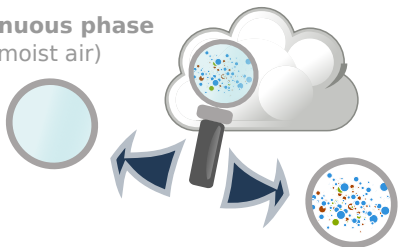
dispersed phase

(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



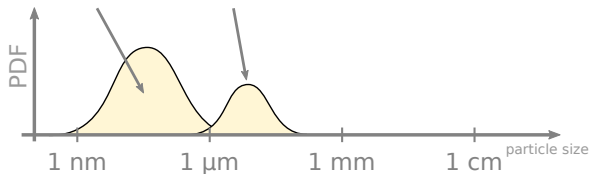
modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

continuous phase
(moist air)



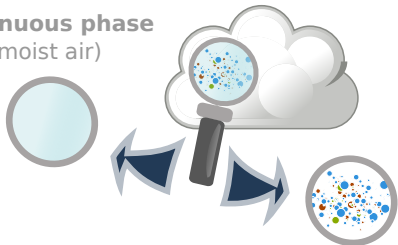
dispersed phase

(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



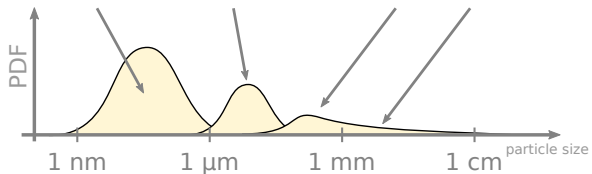
modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

continuous phase
(moist air)



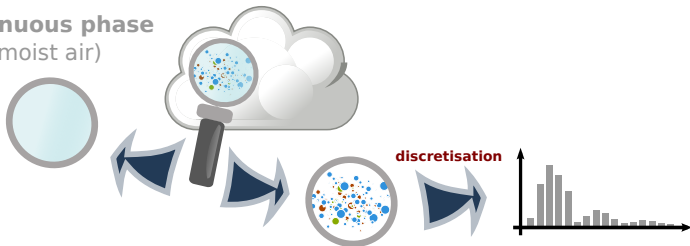
dispersed phase

(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

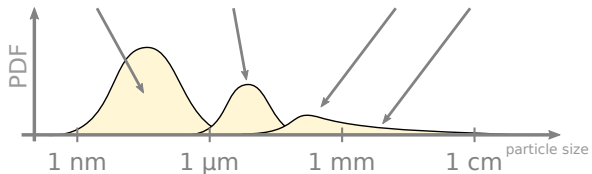
continuous phase
(moist air)



discretisation

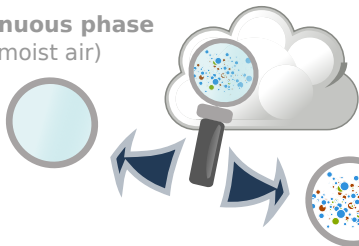
dispersed phase

(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



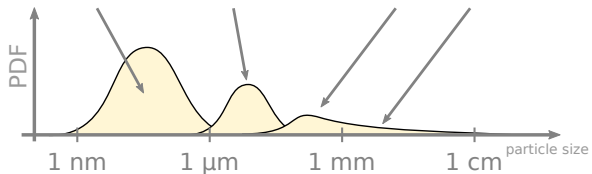
modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

continuous phase
(moist air)

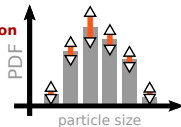


discretisation

dispersed phase
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



Eulerian representation



spectrum evolution

Smoluchowski's coagulation equation (SCE)

concentration of particles of size x at time t : $c(x, t): \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

collision kernel: $a(x_1, x_2): \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

Smoluchowski's coagulation equation (SCE)

concentration of particles of size x at time t : $c(x, t): \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

collision kernel: $a(x_1, x_2): \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$$\dot{c}(x) = \frac{1}{2} \int_0^x a(y, x-y)c(y)c(x-y)dy - \int_0^\infty a(y, x)c(y)c(x)dy \quad (1)$$

Smoluchowski's coagulation equation (SCE)

concentration of particles of size x at time t : $c(x, t): \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

collision kernel: $a(x_1, x_2): \mathbb{R}^+ \times \mathbb{R}^+ \rightarrow \mathbb{R}^+$

$$\dot{c}(x) = \frac{1}{2} \int_0^x a(y, x-y) c(y) c(x-y) dy - \int_0^\infty a(y, x) c(y) c(x) dy \quad (1)$$

discretised particle concentration: $c_i = c(x_i)$ where $x_i = i \cdot x_0$

$$\dot{c}_i = \frac{1}{2} \sum_{k=1}^{i-1} a(x_k, x_{i-k}) c_k c_{i-k} - \sum_{k=1}^{\infty} a(x_k, x_i) c_k c_i \quad (2)$$

cloud droplet collisional growth

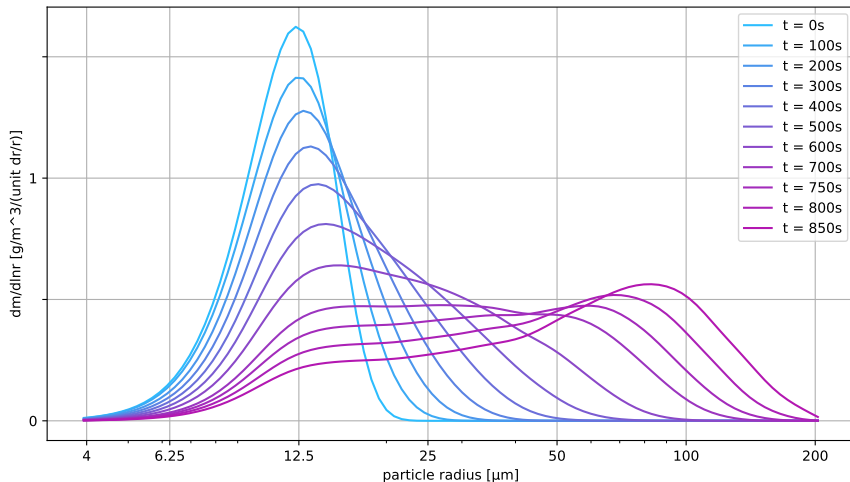


figure (PySDM simulation): Bartman, Arabas et al. 2021, LNCS
(doi:10.1007/978-3-030-77964-1_2)

- ▶ analytic solutions known only for simple kernels

- ▶ analytic solutions known only for simple kernels
- ▶ numerical methods suffer from the curse of dimensionality
when distinguishing particles of same size but different properties

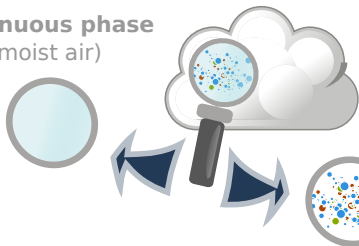
- ▶ analytic solutions known only for simple kernels
- ▶ numerical methods suffer from the curse of dimensionality when distinguishing particles of same size but different properties
- ▶ assumptions behind SCE difficult to meet in practice, e.g.:

- ▶ analytic solutions known only for simple kernels
- ▶ numerical methods suffer from the curse of dimensionality when distinguishing particles of same size but different properties
- ▶ assumptions behind SCE difficult to meet in practice, e.g.:
 - it is assumed that the system is large enough and the droplets inside are uniformly distributed, which in turn is only true for a small volume in the atmosphere

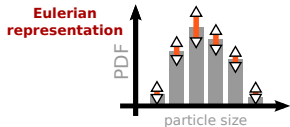
- ▶ analytic solutions known only for simple kernels
- ▶ numerical methods suffer from the curse of dimensionality when distinguishing particles of same size but different properties
- ▶ assumptions behind SCE difficult to meet in practice, e.g.:
 - it is assumed that the system is large enough and the droplets inside are uniformly distributed, which in turn is only true for a small volume in the atmosphere
- ▶ ...

modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

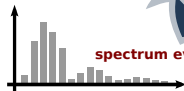
continuous phase
(moist air)



discretisation

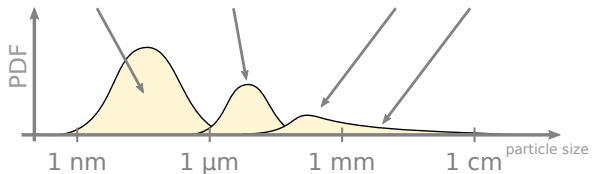


spectrum evolution

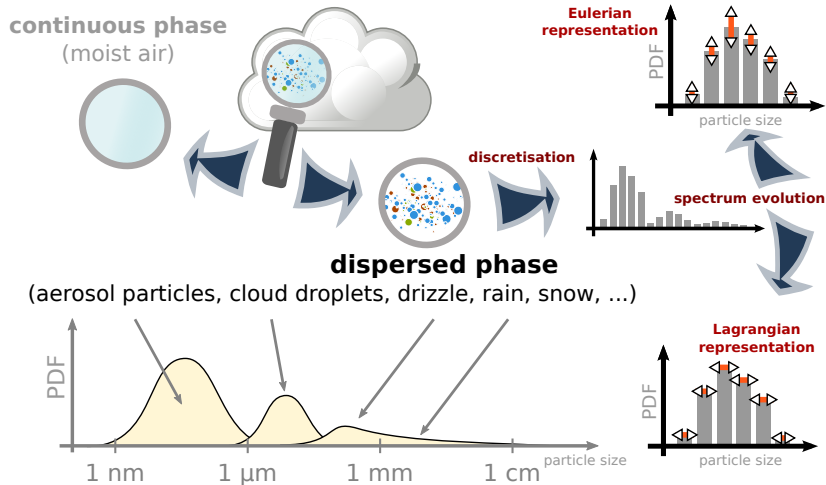


dispersed phase

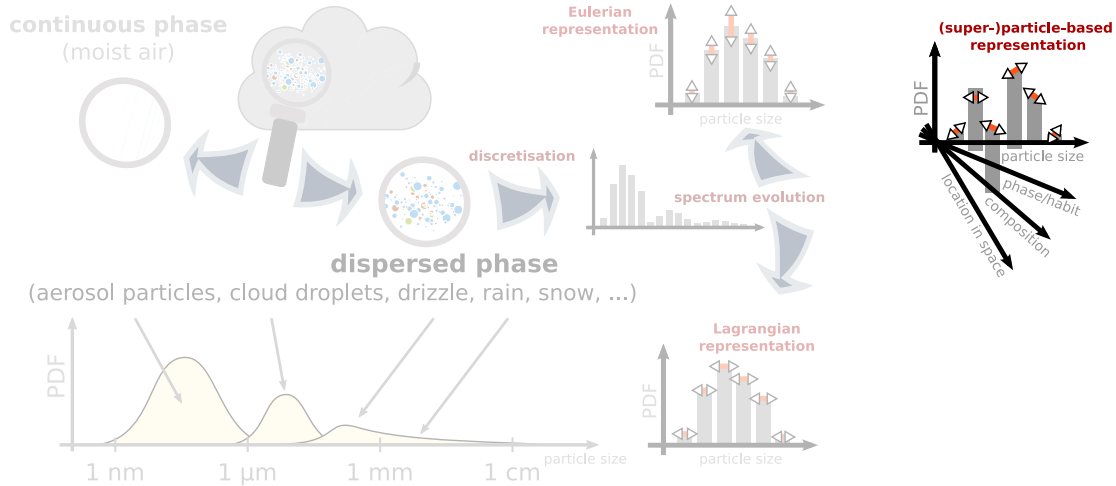
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

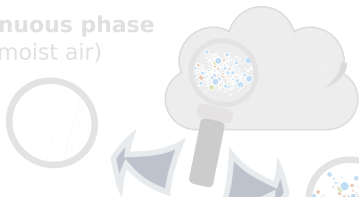


modelling cloud μ -physics: Eulerian vs. Lagrangian approaches



modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

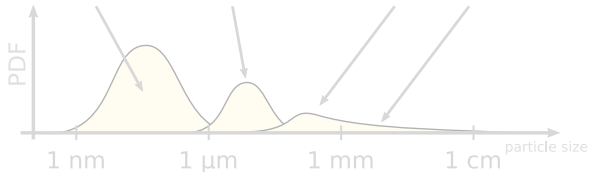
continuous phase
(moist air)



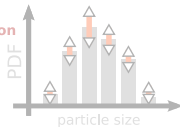
discretisation



dispersed phase
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



Eulerian representation

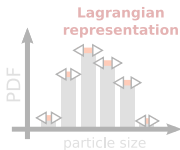


spectrum evolution

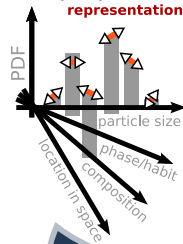
discretisation

discretisation

Lagrangian representation

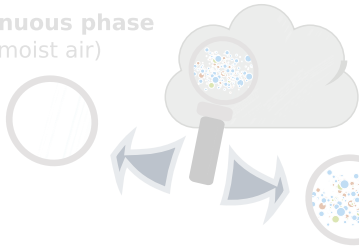


(super-)particle-based representation

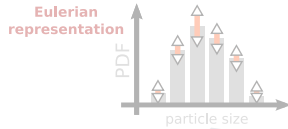


modelling cloud μ -physics: Eulerian vs. Lagrangian approaches

continuous phase
(moist air)

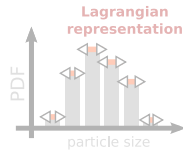
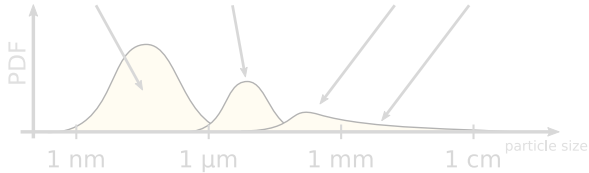


discretisation

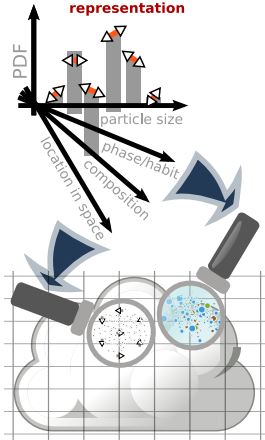


spectrum evolution

dispersed phase
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



(super-)particle-based representation



Monte-Carlo SCE alternatives: e.g., SDM by Shima et al.

Shima et al. 2009 (doi:10.1002/qj.441): warm-rain

Monte-Carlo SCE alternatives: e.g., SDM by Shima et al.

Shima et al. 2009 (doi:10.1002/qj.441): warm-rain

Shima et al. 2020 (doi:10.5194/gmd-13-4107-2020): mixed-phase

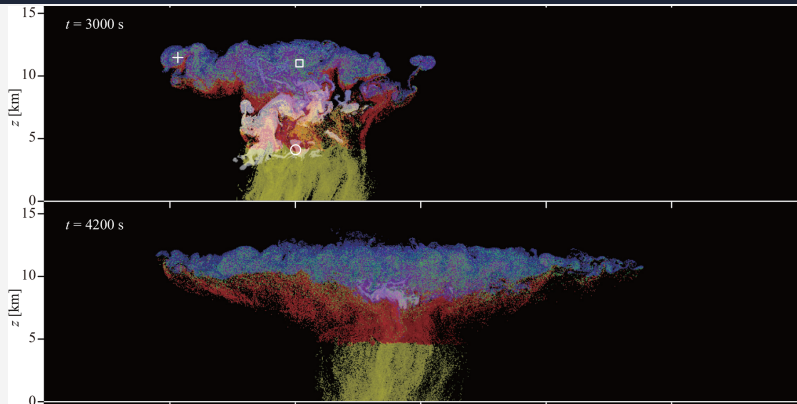


Figure 1. Typical realization of CTRL cloud spatial structures at $t = 2040, 2460, 3000, 4200,$ and 5400 s. The mixing ratio of cloud water, rainwater, cloud ice, graupel, and snow aggregates are plotted in fading white, yellow, blue, red, and green, respectively. The symbols indicate examples of unrealistic predicted ice particles (Sects. 7.3 and 9.1). See also Movie 1 in the video supplement.

Super Droplet Method vs. SCE: differences

SCE (naïve impl)

SDM

method type

mean-field, deterministic

Monte-Carlo, stochastic

Super Droplet Method vs. SCE: differences

SCE (naïve impl)

SDM

method type

mean-field, deterministic

Monte-Carlo, stochastic

considered pairs

all (i,j) pairs

random set of $n_{sd}/2$ non-overlapping pairs,
probability up-scaled by $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$ ratio

Super Droplet Method vs. SCE: differences

SCE (naïve impl)

SDM

method type

mean-field, deterministic

Monte-Carlo, stochastic

considered pairs

all (i,j) pairs

random set of $n_{sd}/2$ non-overlapping pairs,
probability up-scaled by $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$ ratio

computation complexity

$\mathcal{O}(n_{sd}^2)$

$\mathcal{O}(n_{sd})$

Super Droplet Method vs. SCE: differences

SCE (naïve impl)

SDM

method type

mean-field, deterministic

Monte-Carlo, stochastic

considered pairs

all (i,j) pairs

random set of $n_{sd}/2$ non-overlapping pairs,
probability up-scaled by $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$ ratio

computation complexity

$\mathcal{O}(n_{sd}^2)$

$\mathcal{O}(n_{sd})$

collisions triggered

every time step

by comparing probability with a random number

Super Droplet Method vs. SCE: differences

SCE (naïve impl)

SDM

method type

mean-field, deterministic

Monte-Carlo, stochastic

considered pairs

all (i,j) pairs

random set of $n_{sd}/2$ non-overlapping pairs,
probability up-scaled by $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$ ratio

computation complexity

$\mathcal{O}(n_{sd}^2)$

$\mathcal{O}(n_{sd})$

collisions triggered

every time step

by comparing probability with a random number

collisions

colliding a fraction of $\xi_{[i]}$, $\xi_{[j]}$

collide all of $\min\{\xi_{[i]}, \xi_{[j]}\}$ ("all or nothing")

Super Droplet Method vs. SCE: differences

SCE (naïve impl)

SDM

method type

mean-field, deterministic

Monte-Carlo, stochastic

considered pairs

all (i,j) pairs

random set of $n_{sd}/2$ non-overlapping pairs,
probability up-scaled by $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$ ratio

computation complexity

$\mathcal{O}(n_{sd}^2)$

$\mathcal{O}(n_{sd})$

collisions triggered

every time step

by comparing probability with a random number

collisions

colliding a fraction of $\xi_{[i]}$, $\xi_{[j]}$

collide all of $\min\{\xi_{[i]}, \xi_{[j]}\}$ ("all or nothing")

interpretation

concentration " c_i " in size bin " i "

besides c_i , each "particle" i carries other physicochemical attributes, e.g. position (x_i, y_i, z_i)

SDM

PySDM

Develop an implementation of the SDM algorithm:

Develop an implementation of the SDM algorithm:

- ▶ applicable in research on aerosol-cloud-interactions (and beyond)
KPI: reproduction of results from classic and recent literature

Develop an implementation of the SDM algorithm:

- ▶ applicable in research on aerosol-cloud-interactions (and beyond)

KPI: reproduction of results from classic and recent literature

- ▶ **easy to reuse**: code (Python), examples (Jupyter), extensibility (modular, high test coverage), interoperability (other languages, i/o), leveraging modern hardware (GPUs, multi-core CPUs)

KPI: user feedback & contributions

Develop an implementation of the SDM algorithm:

- ▶ applicable in research on aerosol-cloud-interactions (and beyond)
KPI: reproduction of results from classic and recent literature
- ▶ **easy to reuse**: code (Python), examples (Jupyter), extensibility (modular, high test coverage), interoperability (other languages, i/o), leveraging modern hardware (GPUs, multi-core CPUs)
KPI: user feedback & contributions
- ▶ **accessibility**: seamless Linux/macOS/Windows installation (pip)
KPI: continuous integration on all targeted platforms

Develop an implementation of the SDM algorithm:

- ▶ applicable in research on aerosol-cloud-interactions (and beyond)
KPI: reproduction of results from classic and recent literature
- ▶ **easy to reuse**: code (Python), examples (Jupyter), extensibility (modular, high test coverage), interoperability (other languages, i/o), leveraging modern hardware (GPUs, multi-core CPUs)
KPI: user feedback & contributions
- ▶ **accessibility**: seamless Linux/macOS/Windows installation (pip)
KPI: continuous integration on all targeted platforms
- ▶ **curation**: open licensing (GPL), public versioned development (Github)
KPI: instant and anonymous execution on commodity environment

PySDM: 2D kinematic Sc test (Morrison & Grabowski '07)

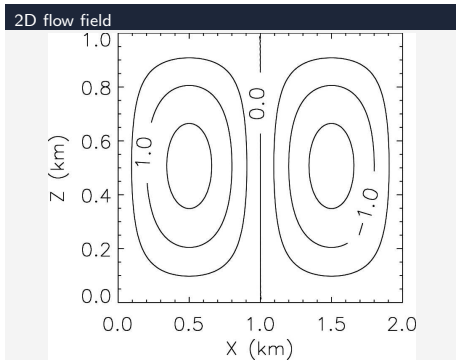
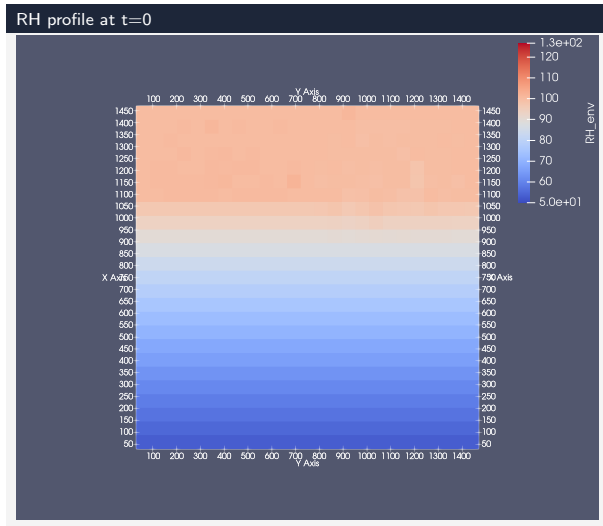
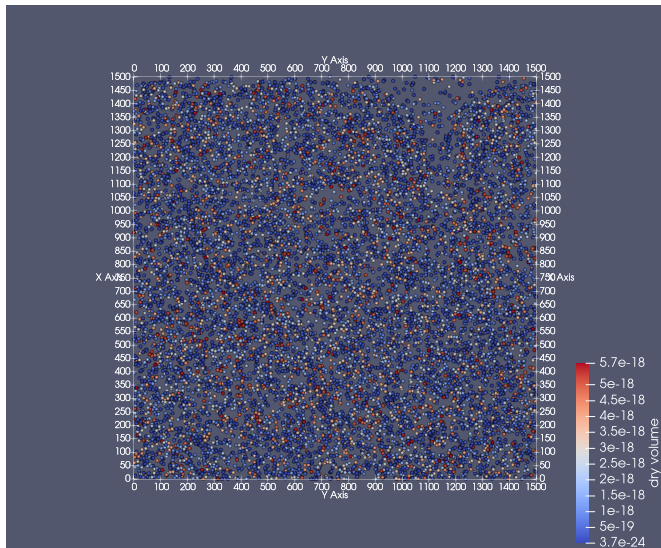


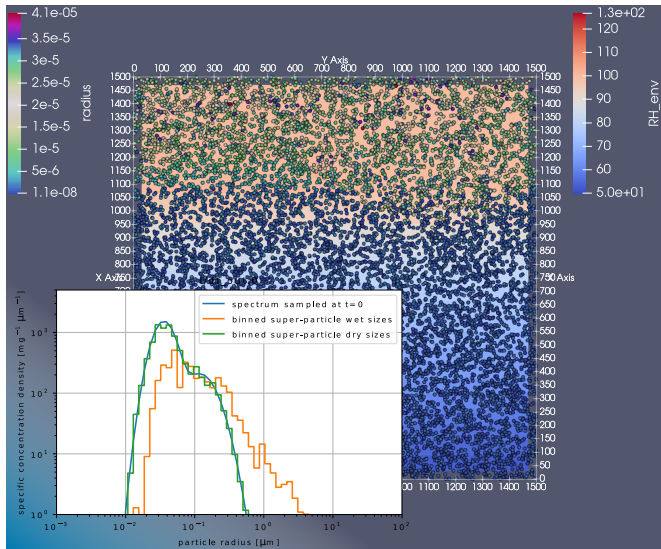
FIG. 1. Time-invariant vertical velocity for the stratocumulus case (contour interval is 0.5 m s^{-1}).



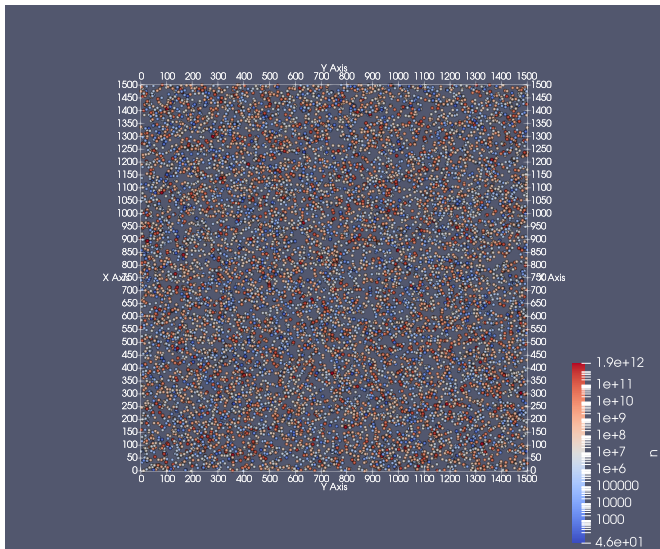
particle attribute initialisation: dry/wet volume



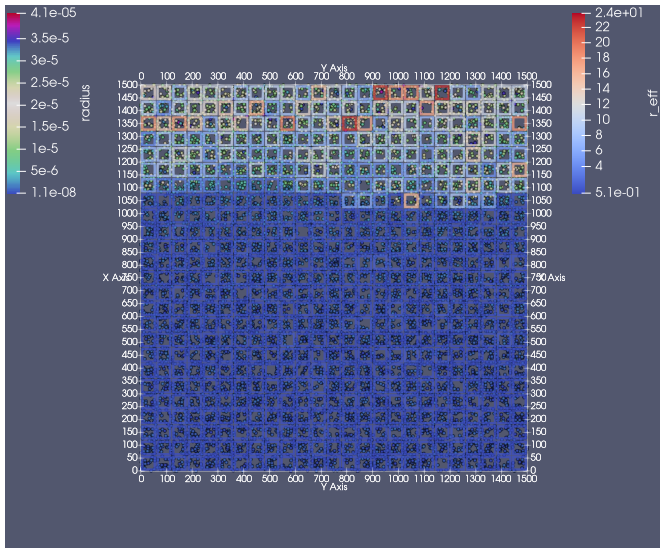
particle attribute initialisation: dry/wet volume



particle attribute initialisation: multiplicity



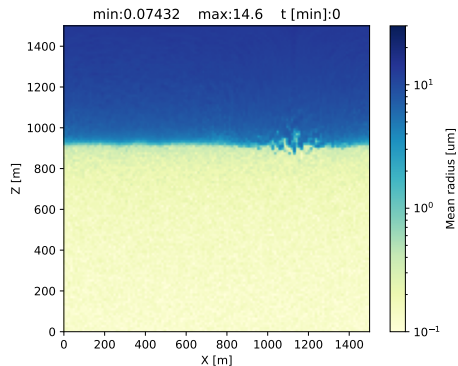
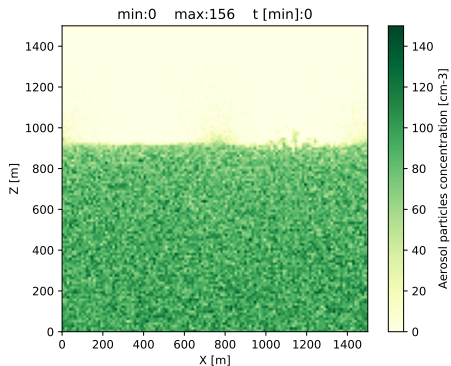
particle attribute evolution: droplet radius



sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

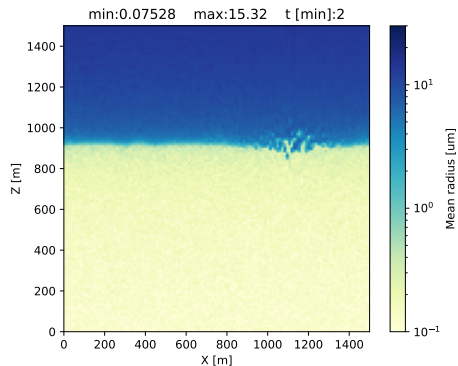
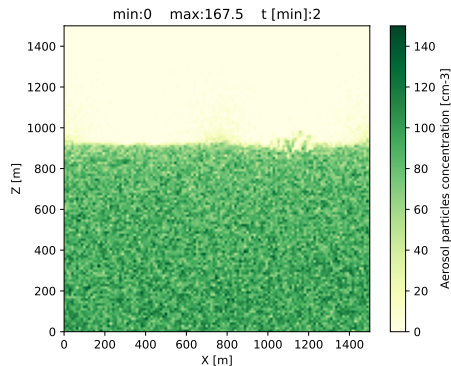


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

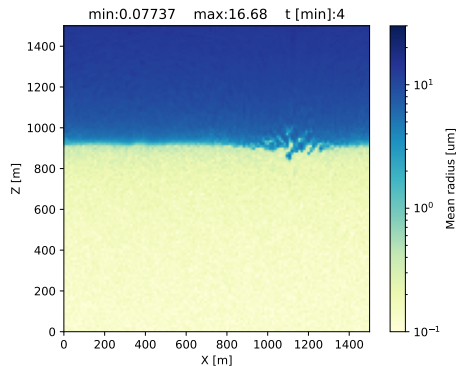
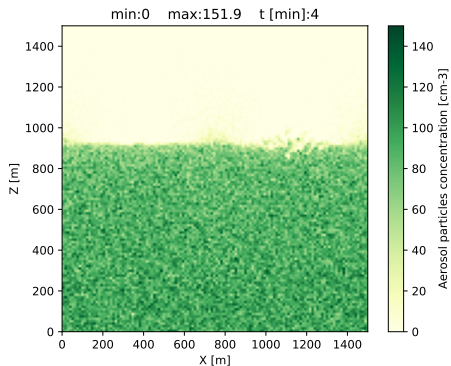


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMIl UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

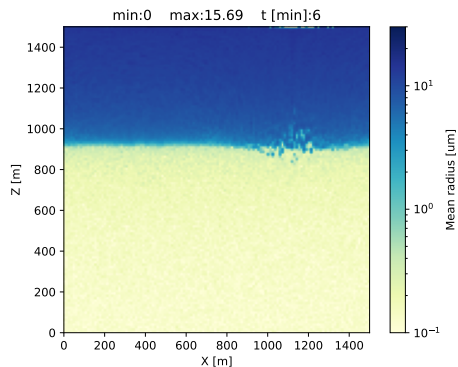
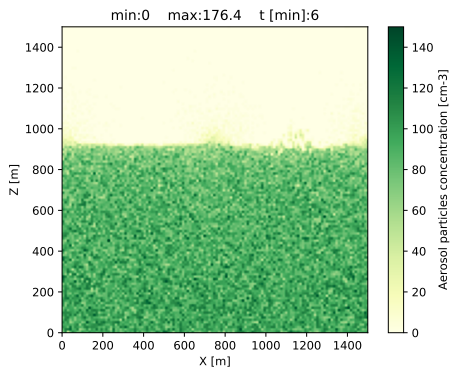


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

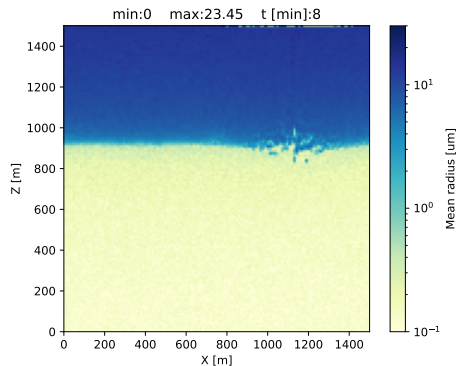
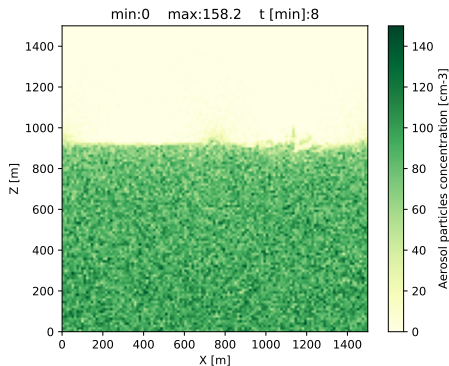


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

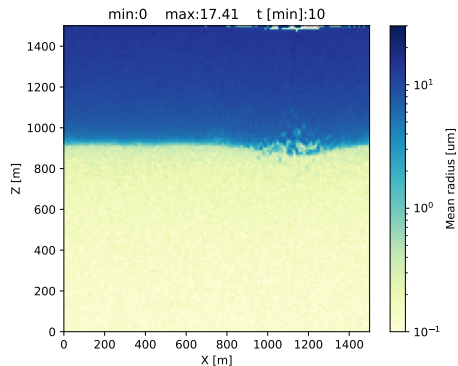
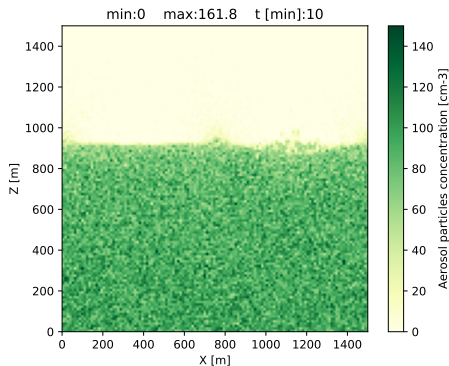


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

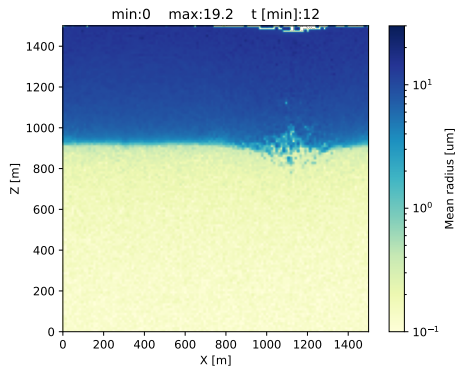
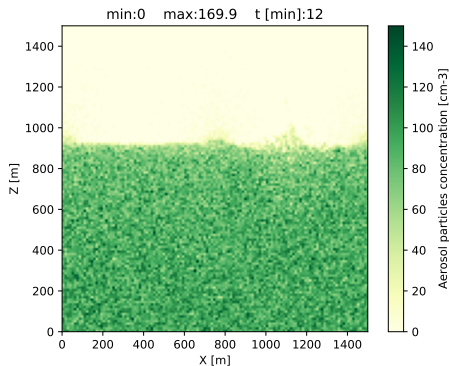


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

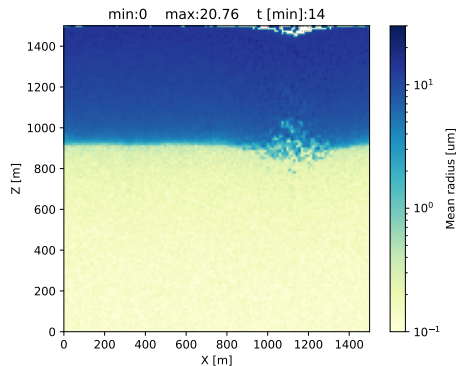
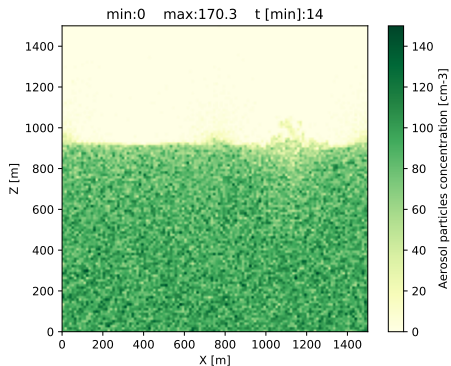


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

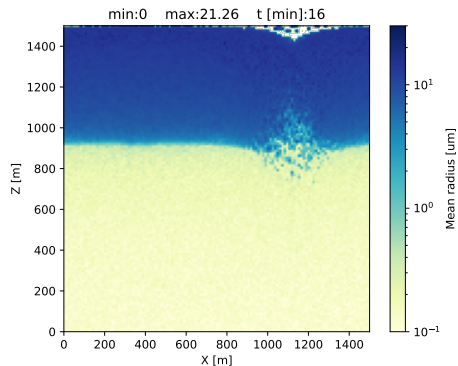
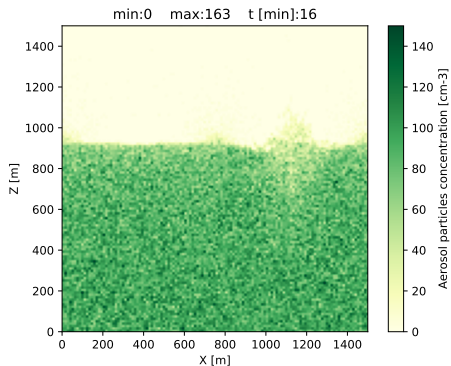


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

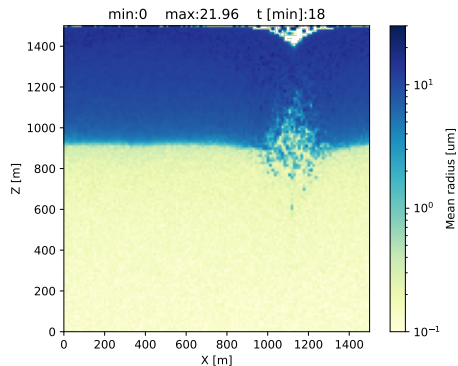
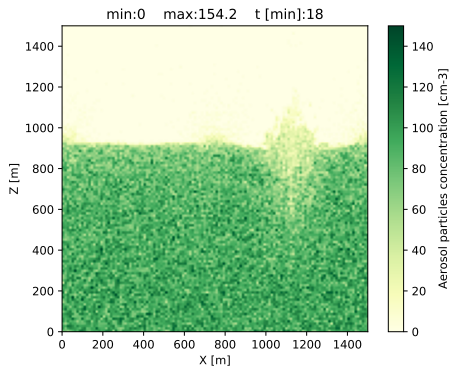


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMIl UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

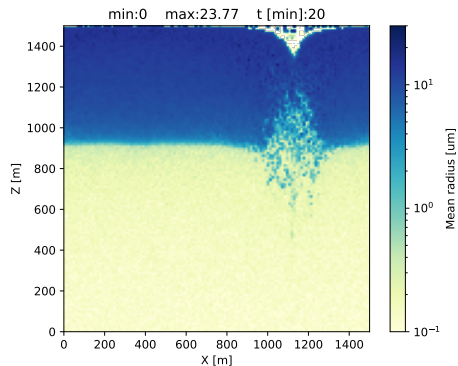
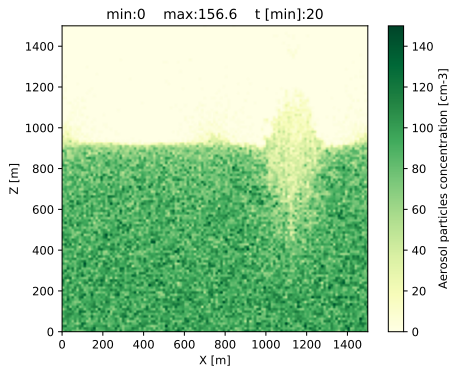


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

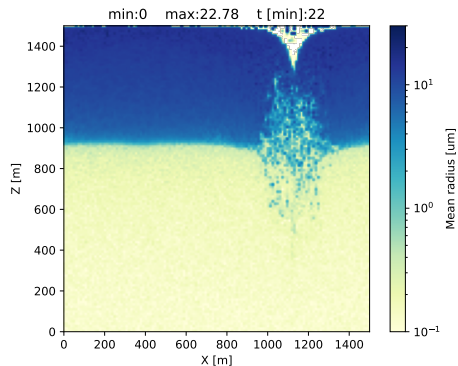
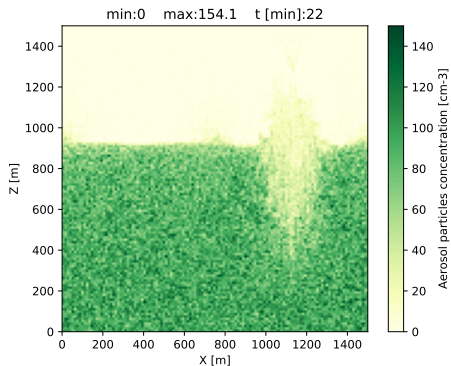


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiU UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

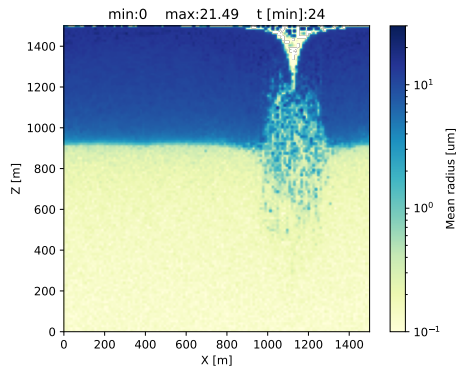
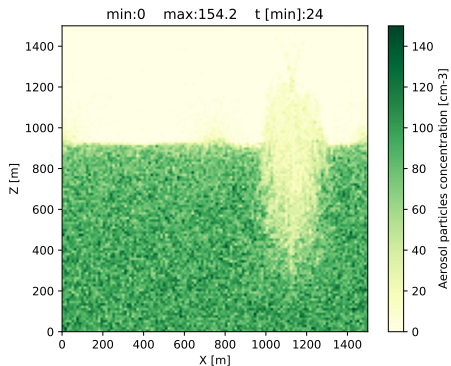


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiU UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

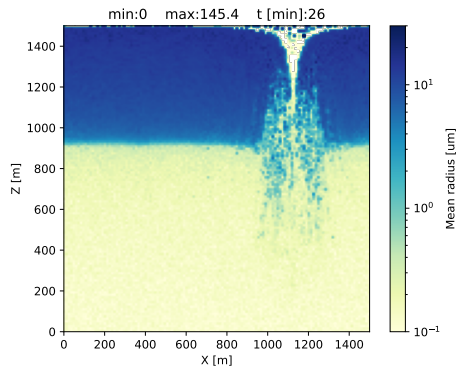
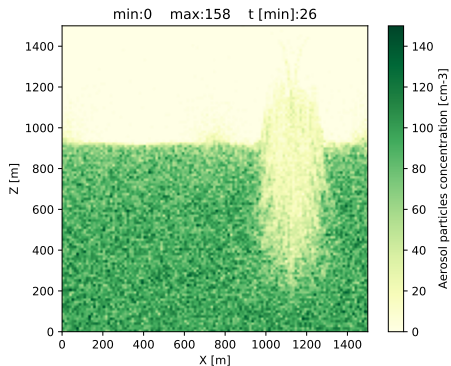


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

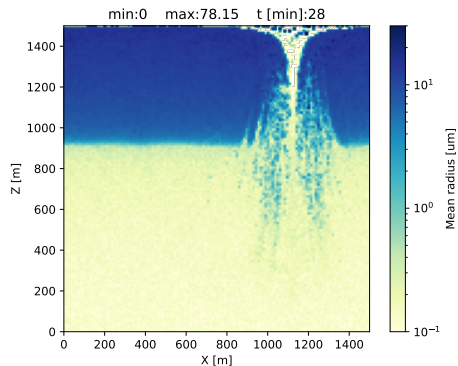
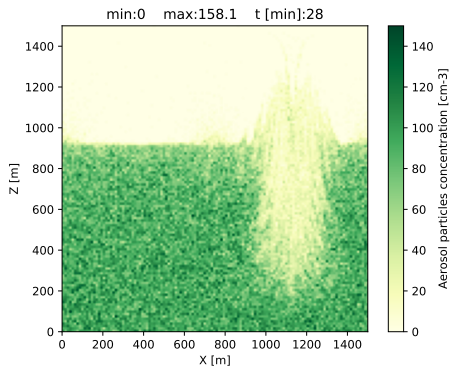


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

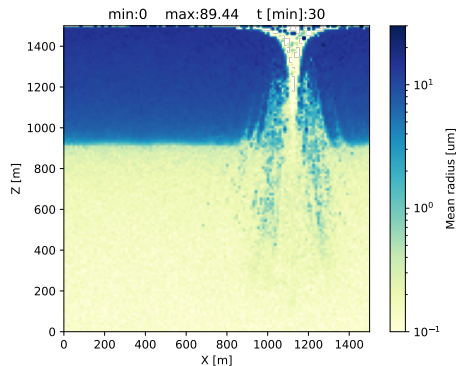
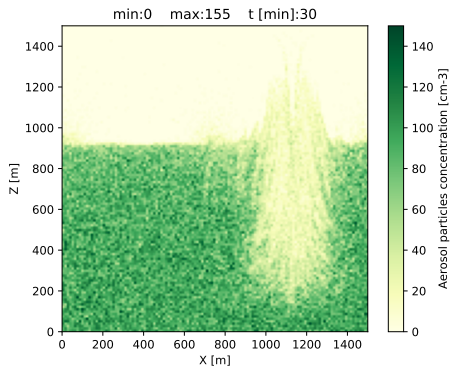


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiU UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

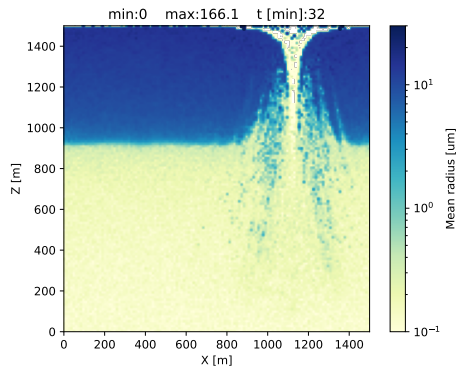
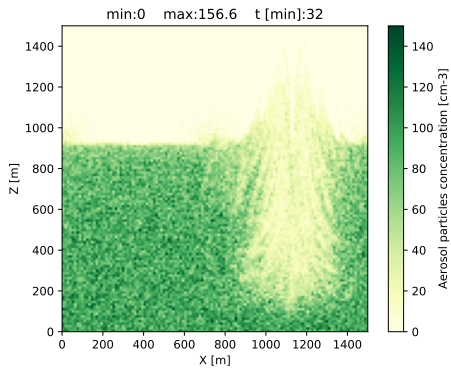


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiU UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

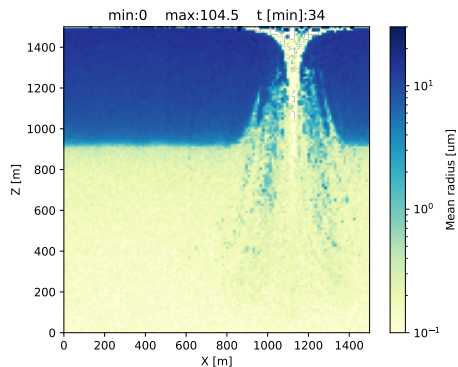
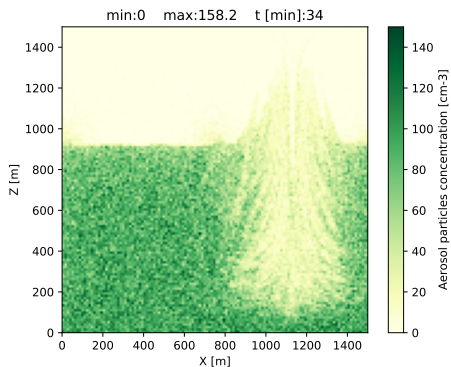


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

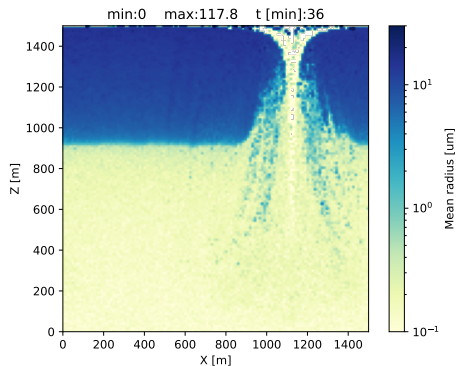
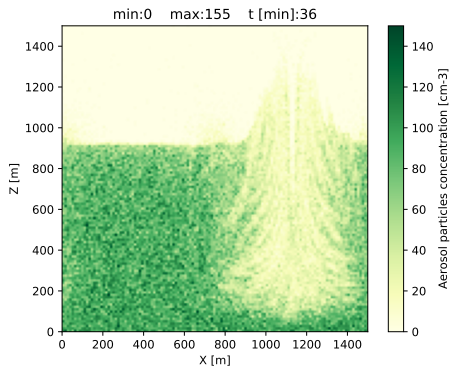


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiU UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

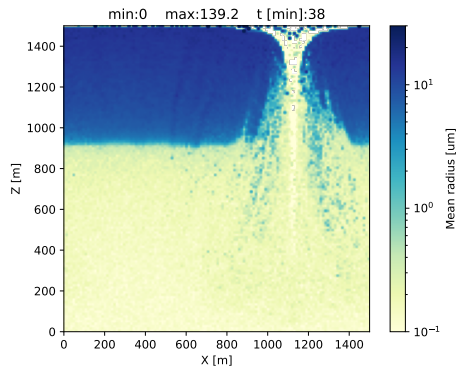
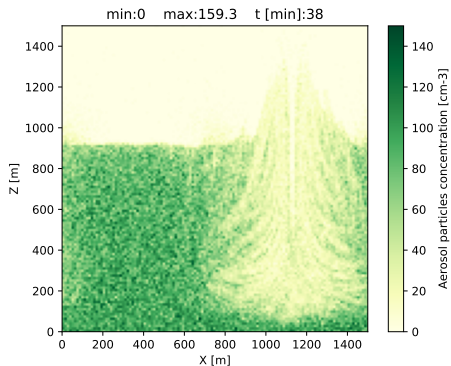


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

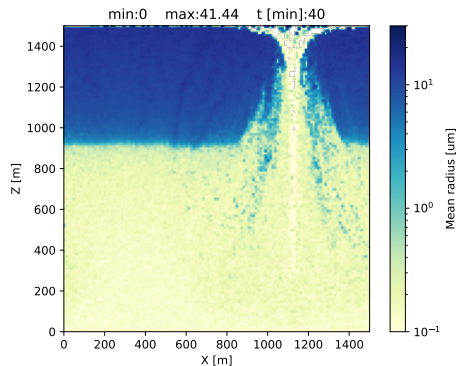
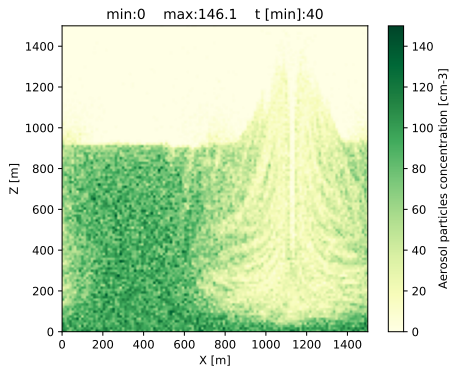


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMil UJ)

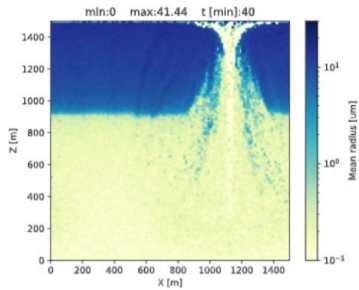
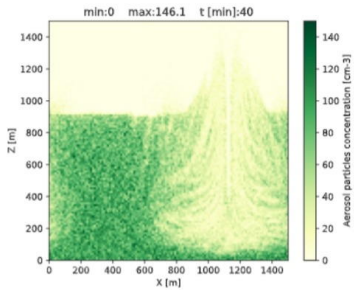
sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

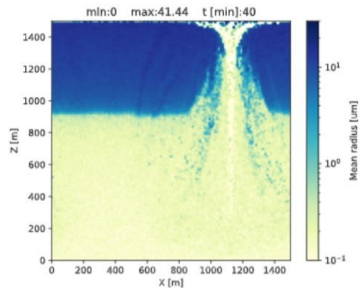
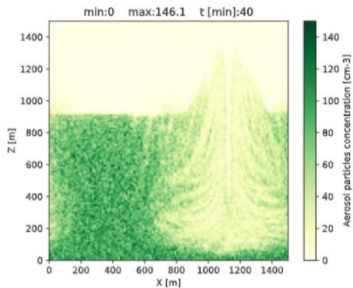


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMIl UJ)



PySDM: Pythonic

```
[3] 1 simulation.run()
```



PySDM: Pythonic, Jupyter-friendly



demo.ipynb ☆

File Edit View Insert Runtime Tools Help



Comment



Share

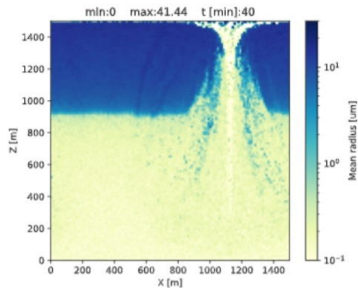
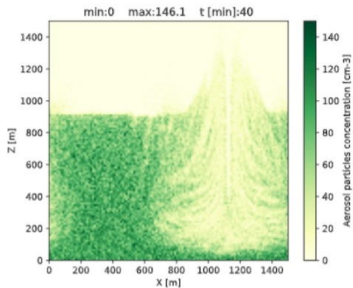


+ Code + Text

✓ RAM
Disk

Editing

```
[3] 1 simulation.run()
```



PySDM: Pythonic, Jupyter-friendly, GPU-enabled

demo.ipynb ☆

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM ✓ Disk

Editing

```
[3] 1 simulation.run()
```

min:0 max:146.1 t[mi] 44 t[mi]:40

Z [m]

X [m]

Aerosol

Mean radius [um]

CANCEL SAVE

first coupling with an external CFD code (Oleksii Bulenok) (<https://github.com/CliMA/ClimateMachine.jl/pull/2244>)

PySDM and ClimateMachine coupling examples in Kinematic setup #2244

[Code](#)

[Open](#) abulenok wants to merge 16 commits into `CliMA:master` from `abulenok:ob-pysdmachine`

Conversation 32

Commits 16

Checks 10

Files changed 17

+2,528 -1



abulenok commented on 27 Oct 2021

Contributor

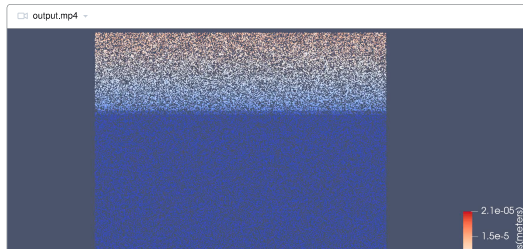
This PR includes a coupling logic for `ClimateMachine.jl` and `PySDM`.

`PySDM` is a particle-based aerosol/cloud microphysics package written entirely in Python.

This PR depicts how Python modules can be leveraged within `ClimateMachine.jl` including the continuous integration setup.

The initial set of tests included here is based on the kinematic 2D example previously used as a test case in both `PySDM` and `ClimateMachine.jl`. In the tests added in this PR, `ClimateMachine.jl` handles air motion and total water transport, while `PySDM` handles representation of aerosol and liquid water transport as well as phase changes leading to formation of cloud water.

Output from `PySDM` is handled using VTK files. Example animation with an evolution of radius computed from particle properties is shown below:



Reviewers

- sdayoo
- charleskawczynski
- claresinger
- jakebolewski
- edejong-csiTech
- tapios

Assignees

- trontrytel

Labels

Microphysics

Projects

None yet

Milestone

No milestone

Development

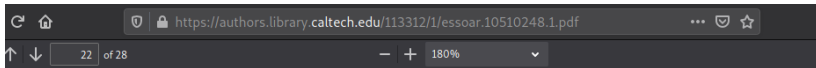
Successfully merging this pull request may close these issues.

None yet

manuscript submitted to *Journal of Advances in Modeling Earth Systems (JAMES)*


An efficient Bayesian approach to learning droplet collision kernels: Proof of concept using “Cloudy”, a new n -moment bulk microphysics scheme

Melanie Bieli¹, Oliver R. A. Dunbar¹, Emily K. de Jong², Anna Jaruga¹,
Tapio Schneider¹, Tobias Bischoff¹




distributions capture the true parameter values within 5% of the posterior mass.


- Moving beyond perfect-model experiments, we have learned collision kernel parameters from output generated by PySDM (Bartman et al., 2021), a Lagrangian particle-based microphysics model. In this experiment, we represent model error resulting from the closure assumption in Cloudy (an assumption that PySDM does not need to make) as a simple bias term. This modification in the setup of the inverse problem allows CES to retrieve the posterior distribution of the “true” parameter, not of that which minimizes the mismatch with the PySDM data.



[Help](#) [Sponsors](#) [Log in](#) [Register](#)

PySDM-examples 2.9


`pip install PySDM-examples` 

 [Latest version](#)

Released: 4 minutes ago

PySDM usage examples reproducing results from literature and depicting how to use PySDM from Python Jupyter notebooks

Navigation

 [Project description](#)

 [Release history](#)

 [Download files](#)

Project links

 [Homepage](#)

Statistics

GitHub statistics:

 Stars: 2

 Forks: 10

Project description

License: [GPL v3](#) Copyright: [Jagiellonian University](#) DOI: [10.5281/zenodo.6604645](#)

 PySDM-examples [passing](#)

 pull requests [2 open](#)  pull requests [158 closed](#)

[pypi package](#) **2.8** [API docs](#) [pdoc3](#)

This repository stores example files for `PySDM` depicting usage of `PySDM` from Python via Jupyter. For information on the `PySDM` package itself and examples of usage from Julia and Matlab, see [PySDM README.md](#) file.

Please use the [PySDM issue-tracking](#) and [discussion](#) infrastructure for `PySDM-examples` as well.

0D box-model coalescence-only examples:

- [Shima et al. 2009](#) (Box model, coalescence only, test case employing Golovin analytical solution):
 - Fig. 2: [render](#) [nbviewer](#) [launch](#) [binder](#) [Open in Colab](#)
- [Berry 1967](#) (Box model, coalescence only, test cases for realistic kernels):
 - Figs. 5, 8 & 10: [render](#) [nbviewer](#) [launch](#) [binder](#) [Open in Colab](#)
- [Bieli et al. 2022](#) (Box model, coalescence and breakup with fixed coalescence efficiency):
 - Fig. 2: [render](#) [nbviewer](#) [launch](#) [binder](#) [Open in Colab](#)



<https://doi.org/10.1038/s41467-019-12982-0>

OPEN

Key drivers of cloud response to surface-active organics

S.J. Lowe^{1,2}, D.G. Partridge³, J.F. Davies⁴, K.R. Wilson⁵, D. Topping⁶ & I. Riipinen^{1,2,7*}

Aerosol-cloud interactions constitute the largest source of uncertainty in global radiative forcing estimates, hampering our understanding of climate evolution. Recent empirical evidence suggests surface tension depression by organic aerosol to significantly influence the formation of cloud droplets, and hence cloud optical properties. In climate models, however, surface tension of water is generally assumed when predicting cloud droplet concentrations. Here we show that the sensitivity of cloud microphysics, optical properties and shortwave radiative effects to the surface phase are dictated by an interplay between the aerosol particle size distribution, composition, water availability and atmospheric dynamics. We demonstrate that accounting for the surface phase becomes essential in clean environments in which ultrafine particle sources are present. Through detailed sensitivity analysis, quantitative constraints on the key drivers – aerosol particle number concentrations, organic fraction and fixed updraft velocity – are derived for instances of significant cloud microphysical susceptibilities to the surface phase.

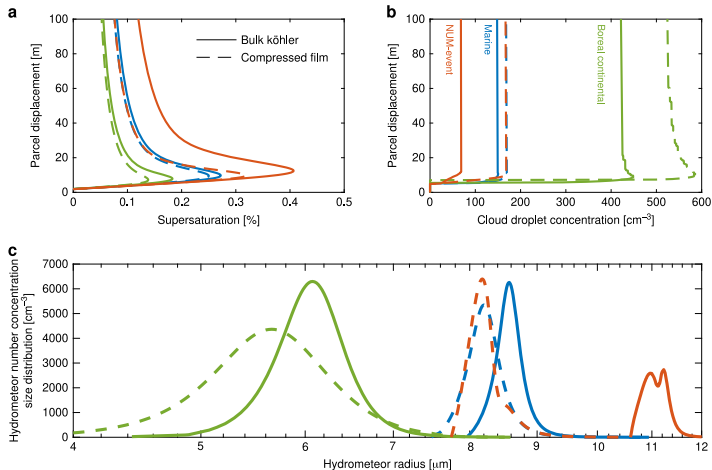
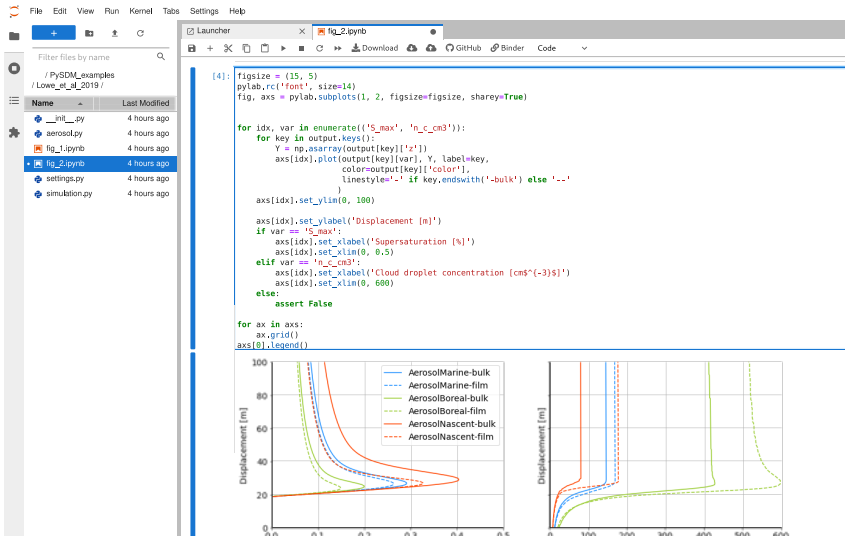


Fig. 2 Simulated microphysics of cloud events on marine (MA, blue), boreal (HYY, green) and NUM-event (NE, orange) aerosol populations. Cloud-formation event simulations using bulk Köhler BK (solid lines) and approximate compressed film CF (dotted lines) models of cloud droplet activation with initial temperature $T = 280$ K, pressure $P = 98,000$ Pa, supersaturation $s = -0.1\%$ and fixed updraft velocity $w = 0.32 \text{ ms}^{-1}$. Simulated (a) ambient parcel supersaturation and (b) cloud droplet number concentration during parcel ascent. c Simulated droplet size distribution at a parcel displacement 200 m above initialisation

example contributed by Clare Singer et al. (<https://claresinger.github.io/>)



PySDM: technological stack

- ▶ Python python.org
- ▶ Numba (JIT, multi-threading) numba.pydata.org
- ▶ ThrustRTC (GPU-resident backend)
pypi.org/project/ThrustRTC




PySDM: technological stack

- ▶ Python python.org
- ▶ Numba (JIT, multi-threading) numba.pydata.org
- ▶ ThrustRTC (GPU-resident backend)
pypi.org/project/ThrustRTC

- ▶ GitHub & GitHub Actions github.com
- ▶ Codecov codecov.io
- ▶ AppVeyor appveyor.com

- ▶ Jupyter jupyter.org
- ▶ Binder mybinder.org
- ▶ Colab colab.research.google.com





Atmospheric Cloud Simulation Group @ Jagiellonian University

Poland <http://atmos.ii.uj.edu.pl/> Unfollow

Overview Repositories 8 Projects Packages Teams People 13 Settings

README .md ✎

News:

- [JOSS under review](#) PySDM v2 outline paper
- [youtube](#) Sylwester's talk at Caltech on PySDM/PyMPDATA mixed-phase cloud simulations
- [PR](#) [Ciekawi](#) [Bulenok's](#) PR to ClimateMachine.jl exemplifying coupling with PySDM
- [JOSS under review](#) PyMPDATA outline paper
- [youtube](#) [Piotr Bartman's](#) Monte-Carlo on GPU with Python talk at NCAR's 2021 Improving Scientific Software conference
- [2103.17238](#) PySDM outline paper (published in JOSS)
- [2101.06318](#) [Piotr Bartman's](#) paper on the PySDM coagulation solver design (published in LNCS)
- [2011.14726](#) [Michael Olesik's](#) paper on an application of PyMPDATA in bin microphysics (published in GMD)

Our technological stack:

[Python](#) [Numba](#) [Jupyter](#) [ThrustRTCUDA](#) [Numpy](#) [pytest](#)
[Colab](#) [Cocotool](#) [PyT](#) [GitHub Actions](#) [Jupyter](#) [PyCharm](#)

Our Python packages (with usage examples for Julia & Matlab):

PySDM: [PySD package 2.8](#) [codcov](#) 76% [PySDM docs](#) [pdoc3](#)

PySDM-examples: [PySD package 2.8](#) [PySDM examples docs](#) [pdoc3](#)

PyMPDATA: [PySD package 1.0.1](#) [codcov](#) 91% [PyMPDATA docs](#) [pdoc3](#)

PyMPDATA-examples: [PySD package 1.0.1](#) [PyMPDATA examples docs](#) [pdoc3](#)

numba-mpi: [PySD package 0.3](#) [numba mpi docs](#) [pdoc3](#)


atmos-cloud-sim-utils: [PySD package 0.5](#) [utils docs](#) [pdoc3](#)

Funding:

[EU Funding by FNP](#) [PL Funding by NCN](#) [US DOE Funding by ASR](#)

View as: Public
You are viewing this page as a public user.

People



[Invite someone](#)

Top languages

[Python](#) [Jupyter Notebook](#)

Most used topics Manage

[#pyd-package](#) [#python](#) [#atmospheric-modeling](#)
[#numba](#) [#atmospheric-physics](#)

The screenshot shows the GitHub profile page for the Atmospheric Cloud Simulation Group @ Jagiellonian University. The profile includes a logo with a crown and a shield, the group name, location (Poland), and website (http://atmos.ii.uj.edu.pl/). Navigation tabs include Overview, Repositories (8), Projects, Packages, Teams (2), People (13), and Settings. The main content area is titled 'News:' and lists several recent publications and talks, such as 'PySDM v2 outline paper', 'Sylwester's talk at Caltech on PySDM/PyMPDATA mixed-phase cloud simulations', and 'Ciekasi Bulenok's PR to ClimateMachine.jl exemplifying coupling with PySDM'. Below the news is a section for 'Our technological stack:' listing tools like Python, Numba, Jupyter, ThrustRTCUDA, Numpy, Pytest, Colab, Cudex, PyT, GitHub Actions, Jupyter, and PyCharm. The 'Our Python packages (with usage examples for Julia & Matlab):' section lists packages like PySDM, PySDM-examples, PyMPDATA, PyMPDATA-examples, numba-mpi, and atmos-cloud-sim-utils, each with version numbers and usage examples. The 'Funding:' section shows logos for EU Funding by FNP, PL Funding by NCN, and US DOE Funding by ASR. On the right side, there are sections for 'View as: Public', 'People' (with avatars), 'Invite someone', 'Top languages' (Python, Jupyter Notebook), and 'Most used topics' (py-pi-package, python, atmospheric-modeling, numba, atmospheric-physics).

we are hiring!

- ▶ 12-month postdoc position within the framework of NCN-funded Ukrainian refugee support (UA researchers only)
- ▶ BSc & MSc stipends (WFAiS students are welcome!)

Acknowledgements

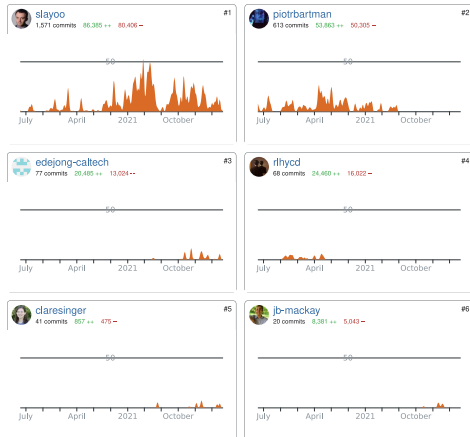
co-authors, contributors, collaborators

- ▶ @uj.edu.pl: P. Bartman, M. Olesik, G. Łazarski, O. Bulenok, ...
- ▶ @caltech.edu: E. de Jong, C. Singer, A. Jaruga, B. Mackay, S. Azimi, ...
- ▶ @illinois.edu: N. Riemer, M. West & J. Curtis

Jun 2, 2019 – Jun 2, 2022

Contributors: Commits

Contributions to main, excluding merge commits and bot accounts



Acknowledgements

co-authors, contributors, collaborators

- ▶ @uj.edu.pl: P. Bartman, M. Olesik, G. Łazarski, O. Bulenok, ...
- ▶ @caltech.edu: E. de Jong, C. Singer, A. Jaruga, B. Mackay, S. Azimi, ...
- ▶ @illinois.edu: N. Riemer, M. West & J. Curtis

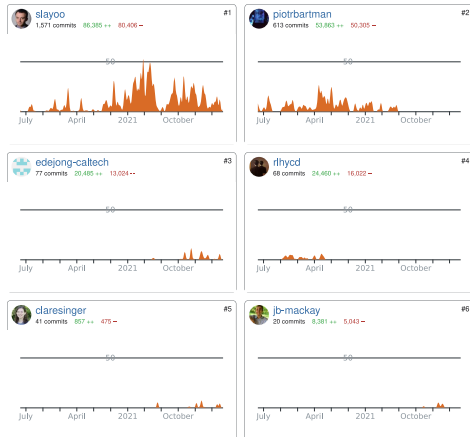
funding

- ▶ PL / National Science Centre
- ▶ EU / Foundation for Polish Science
- ▶ US / DOE Atmospheric System Research & Schmidt Futures

Jun 2, 2019 – Jun 2, 2022

Contributors: Commits

Contributions to main, excluding merge commits and bot accounts



Acknowledgements

co-authors, contributors, collaborators

- ▶ @uj.edu.pl: P. Bartman, M. Olesik, G. Łazarski, O. Bulenok, ...
- ▶ @caltech.edu: E. de Jong, C. Singer, A. Jaruga, B. Mackay, S. Azimi, ...
- ▶ @illinois.edu: N. Riemer, M. West & J. Curtis

funding

- ▶ PL / National Science Centre
- ▶ EU / Foundation for Polish Science
- ▶ US / DOE Atmospheric System Research & Schmidt Futures

Thank you for your attention!

more: <https://atmos.ii.uj.edu.pl/>

contact: sylwester.arabas@uj.edu.pl

Jun 2, 2019 – Jun 2, 2022

Contributors: Commits -

Contributions to main, excluding merge commits and bot accounts

