

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

Sylwester Arabas<sup>0,1</sup>

seminarium Zakładu Zastosowań Metod Obliczeniowych, 2 VI 2022

---

<sup>0</sup>Mathematics and Computer Science, Jagiellonian University ([atmos.ii.uj.edu.pl](http://atmos.ii.uj.edu.pl))

<sup>1</sup>Atmospheric Sciences, University of Illinois at Urbana-Champaign ([atmos.illinois.edu](http://atmos.illinois.edu))

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

Sylwester Arabas<sup>0,1</sup>

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

---

<sup>0</sup>Mathematics and Computer Science, Jagiellonian University ([atmos.ii.uj.edu.pl](http://atmos.ii.uj.edu.pl))

<sup>1</sup>Atmospheric Sciences, University of Illinois at Urbana-Champaign ([atmos.illinois.edu](http://atmos.illinois.edu))

# Sylwester Arabas

# Sylwester Arabas

- ▶ alma mater – [fuw.edu.pl](http://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](https://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](https://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](https://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](https://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](https://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](http://fuw.edu.pl):
  - ▶ MSc (2008) in observational cloud physics
  - ▶ PhD (2013) in computational cloud physics
  - ▶ postdoc (till 2015): CFD software development ([github.com/igfuw](https://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](http://fuw.edu.pl):
  - ▶ MSc (2008) in observational cloud physics
  - ▶ PhD (2013) in computational cloud physics
  - ▶ postdoc (till 2015): CFD software development ([github.com/igfuw](https://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](http://fuw.edu.pl):
  - ▶ MSc (2008) in observational cloud physics
  - ▶ PhD (2013) in computational cloud physics
  - ▶ postdoc (till 2015): CFD software development ([github.com/igfuw](https://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
  - ▶ 2022–2024: NCN „SONATA” @ WMiL UJ

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



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

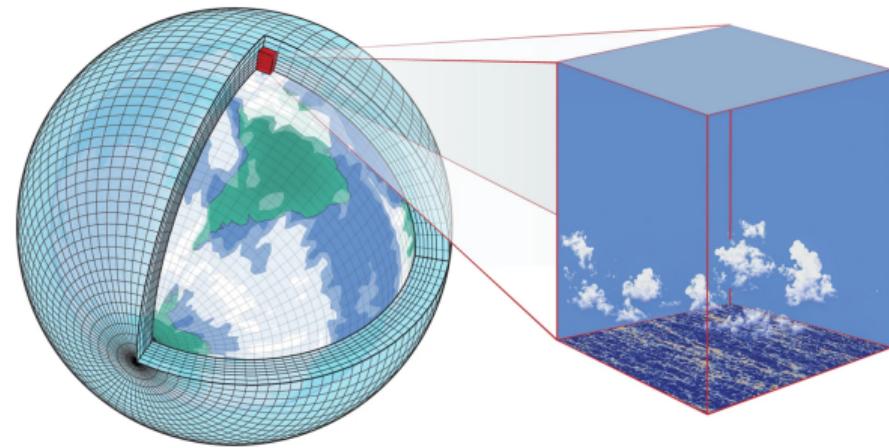
(photo: Yevgen Timashov / National Geographic)

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



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

(photo: Yevgen Timashov / National Geographic)



"Grid cells in a global climate model and a large-eddy simulation of shallow cumulus clouds at 5 m resolution"

(fig. from Schneider et al. 2017)

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

The screenshot shows the official website of the Intergovernmental Panel on Climate Change (IPCC). The header features a blue navigation bar with links for MENU, ABOUT, DATA, DOCUMENTATION, FOCAL POINTS PORTAL, BUREAU PORTAL, LIBRARY, LINKS, LANGUAGES, and SEARCH. Below the header, the IPCC logo is prominently displayed, followed by links for REPORTS, SYNTHESIS REPORT, WORKING GROUPS, ACTIVITIES, NEWS, and CALENDAR. On the right side of the header, there are social media icons for FOLLOW and SHARE. The main content area features a large, stylized title: "The Intergovernmental Panel on Climate Change". To the right of the title is a descriptive text block: "The Intergovernmental Panel on Climate Change (IPCC) is the United Nations body for assessing the science related to climate change." At the bottom left, a button reads "WORKING GROUP II SIXTH ASSESSMENT REPORT". At the bottom right, logos for WHO, UNEP, and the Nobel Peace Prize are shown, along with a series of navigation icons.

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 2007 PEACE PRIZE ON THE NOBEL FOUNDATION

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

Final Government Draft

Chapter 7

IPCC AR6 WGI

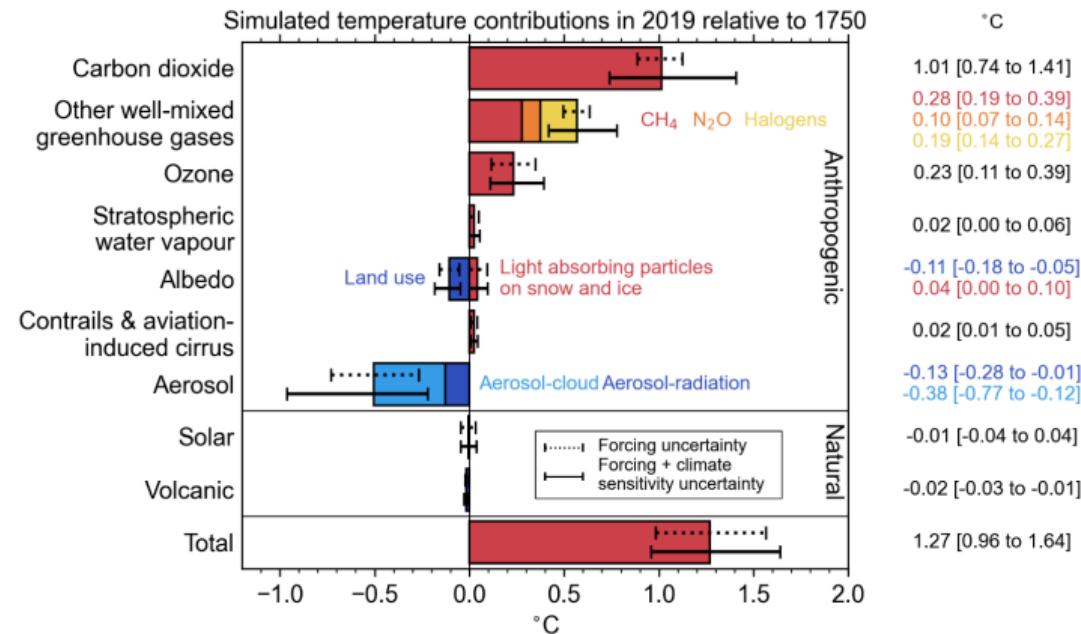
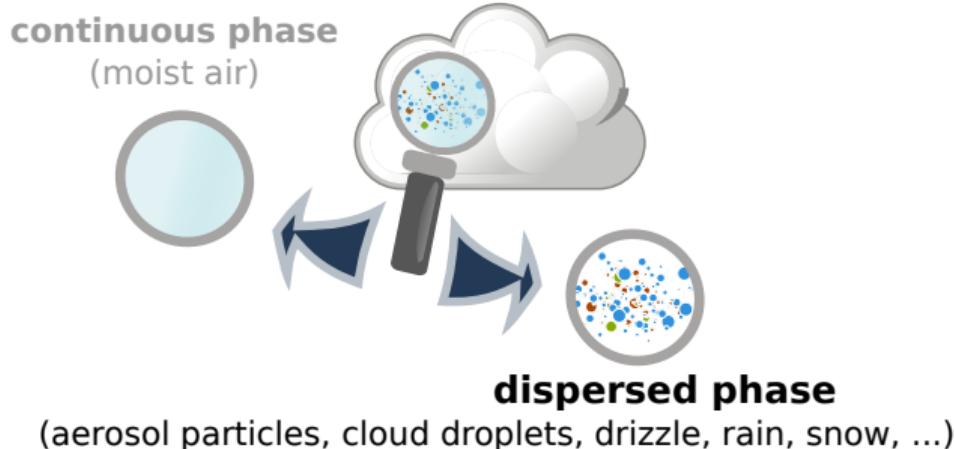


Figure 7.7: The contribution of forcing agents to 2019 temperature change relative to 1750 produced using the two-layer emulator (Supplementary Material 7.SM.2), constrained to assessed ranges for key climate metrics described in Cross-Chapter Box 7.1.

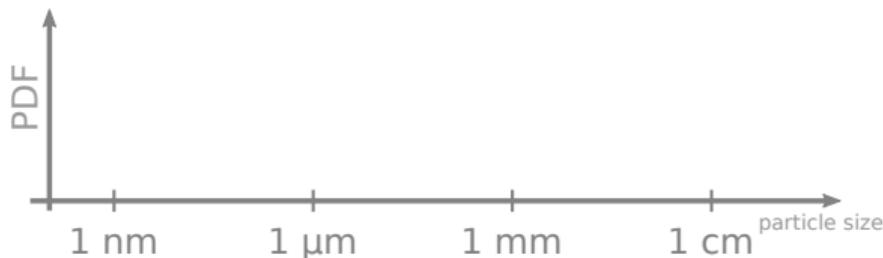
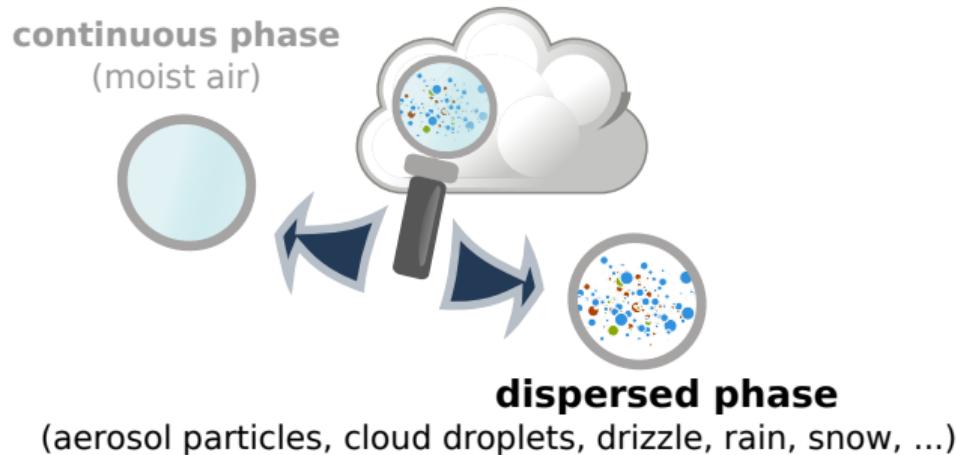
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



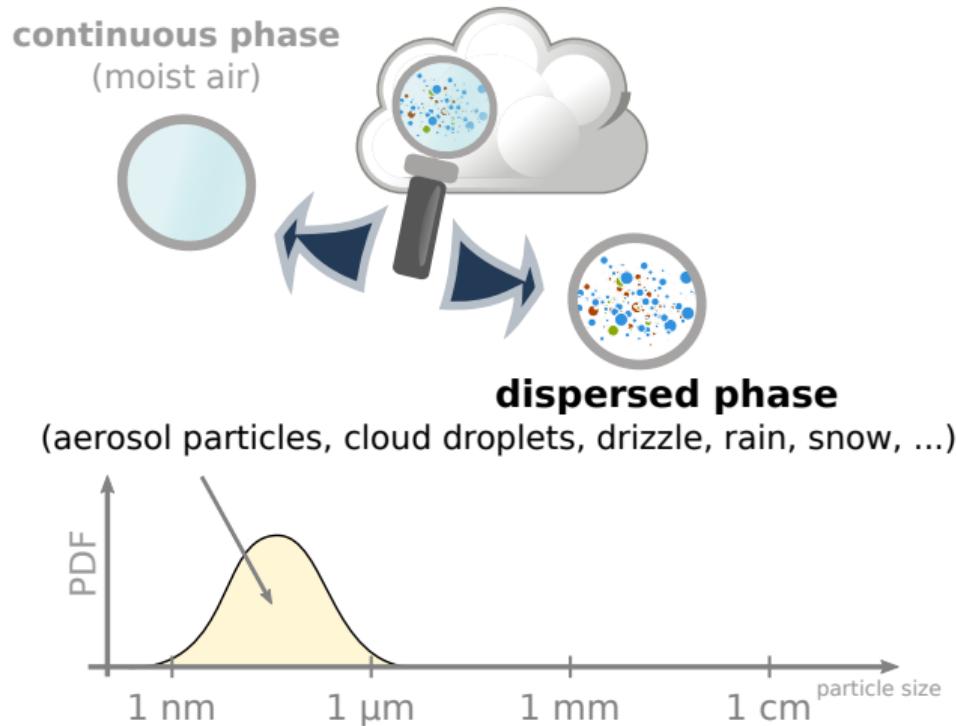
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



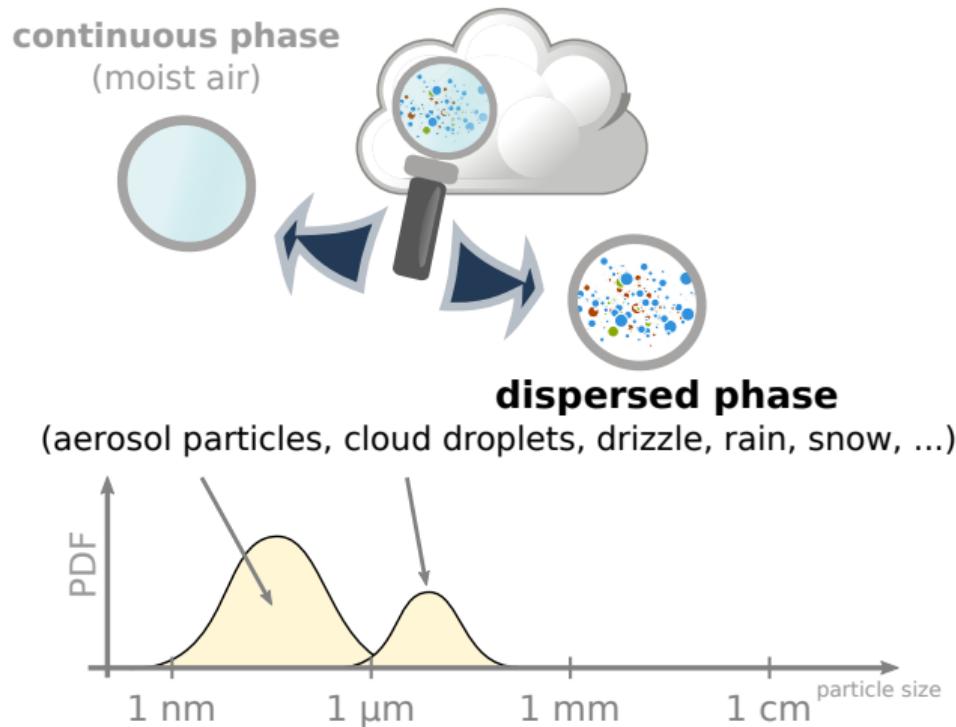
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



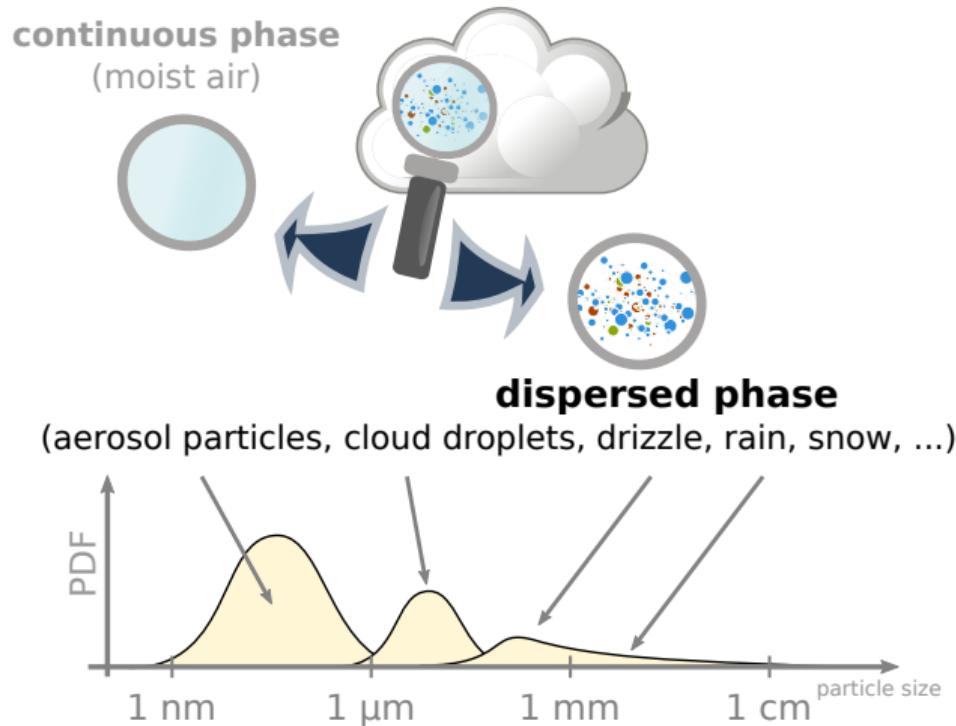
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



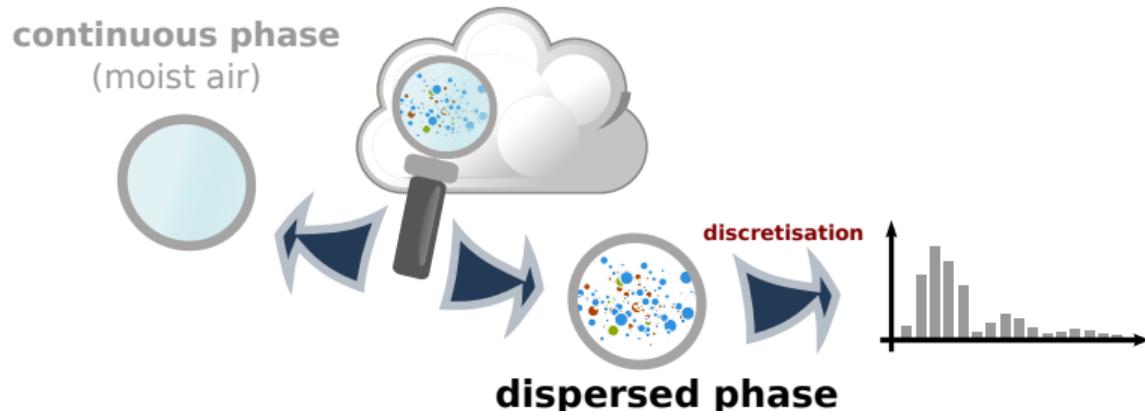
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



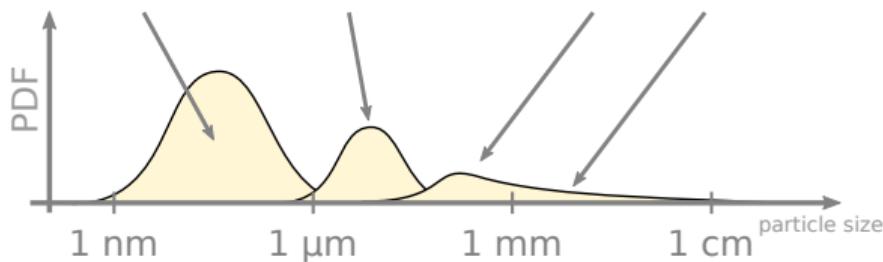
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



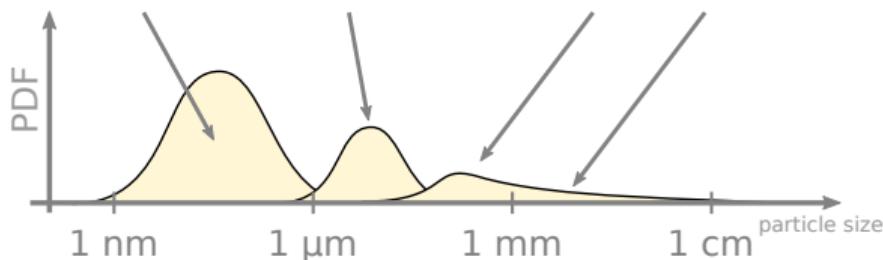
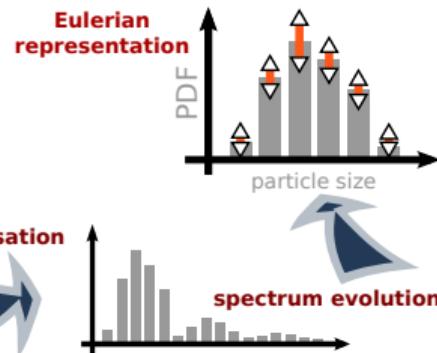
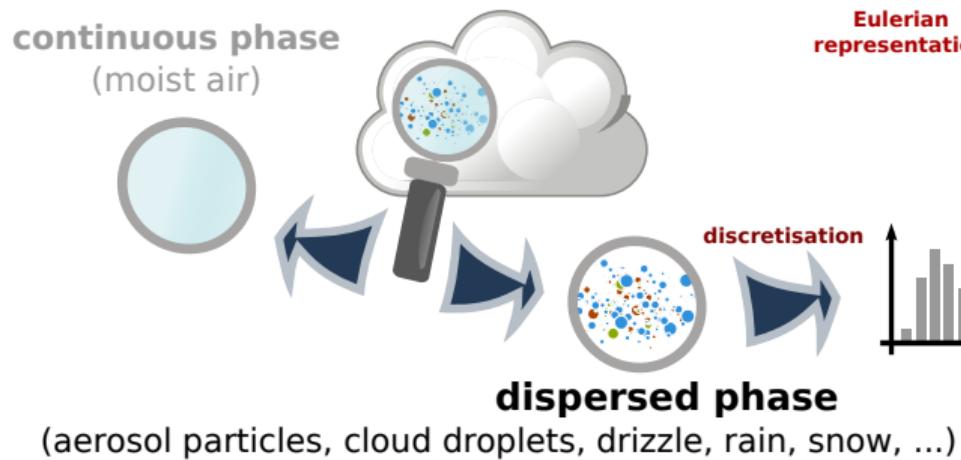
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



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



# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



# 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

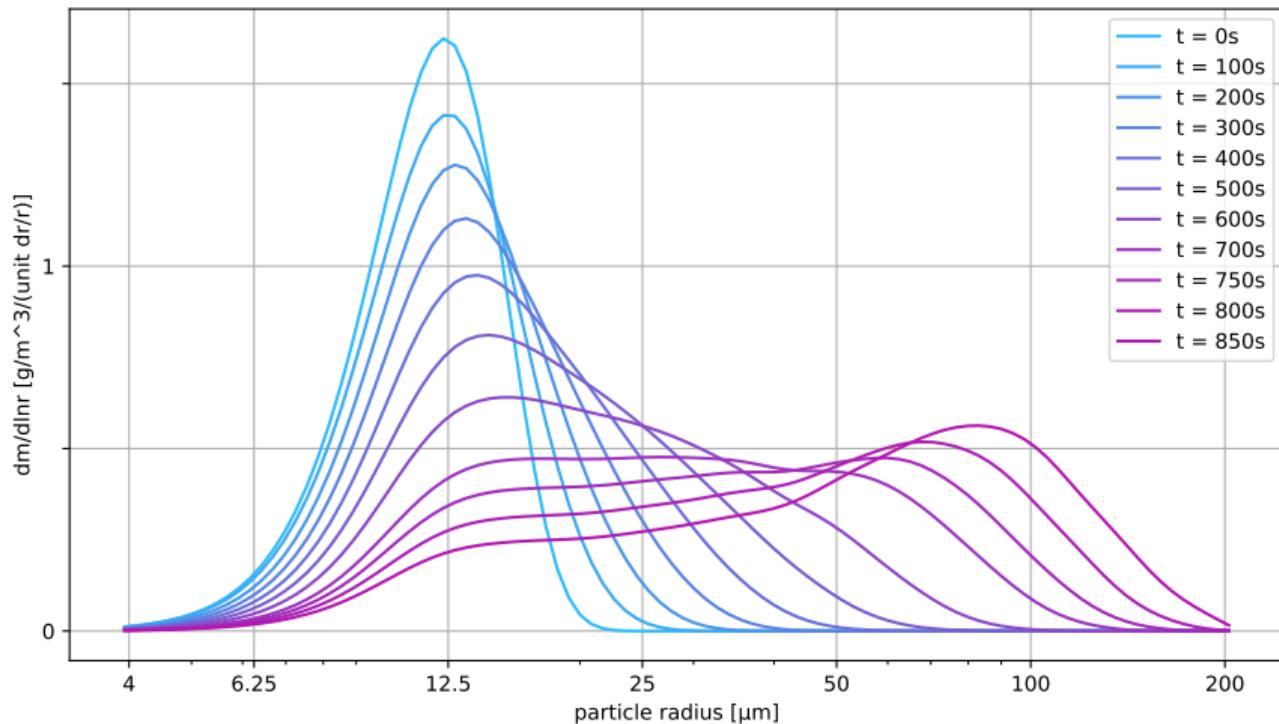


figure (PySDM simulation): Bartman, Arabas et al. 2021, LNCS  
(doi:10.1007/978-3-030-77964-1\_2)

## SCE: challenges/problems

- ▶ analytic solutions known only for simple kernels

## SCE: challenges/problems

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

## SCE: challenges/problems

- ▶ 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.:

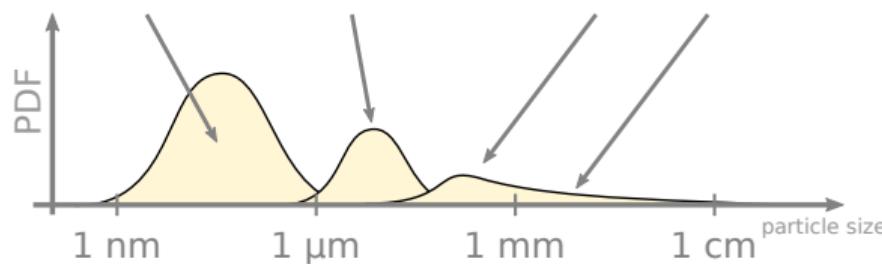
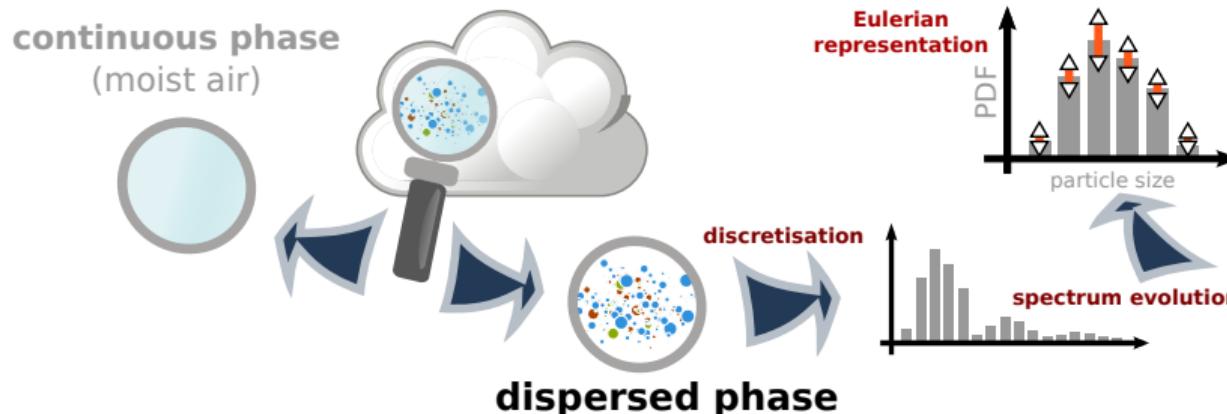
## SCE: challenges/problems

- ▶ 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

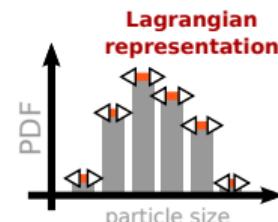
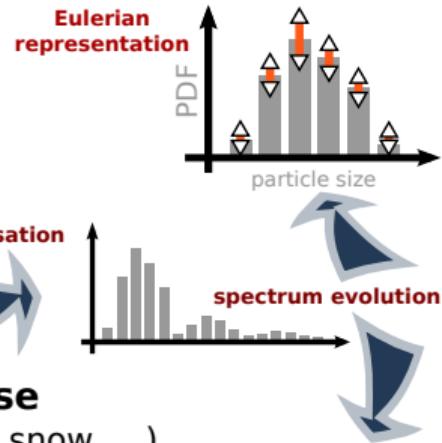
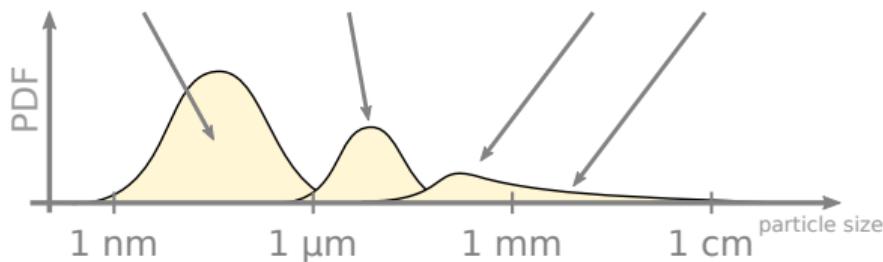
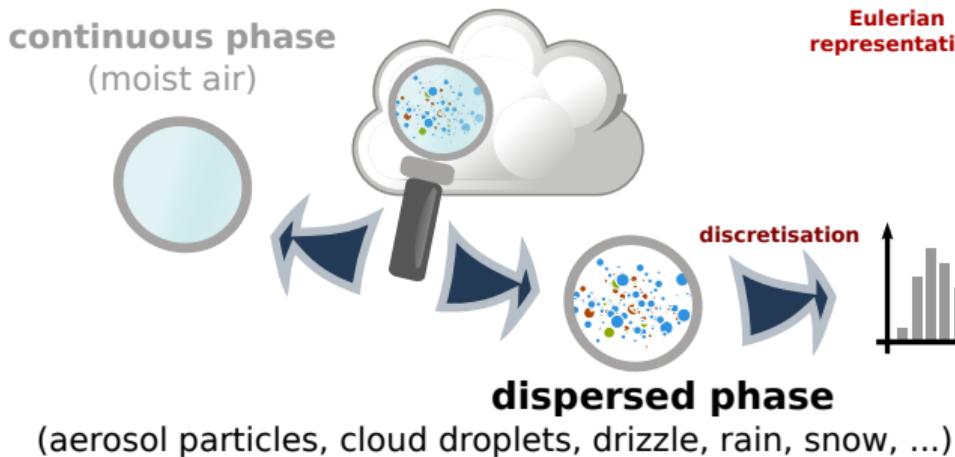
## SCE: challenges/problems

- ▶ 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
- ▶ ...

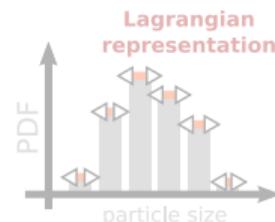
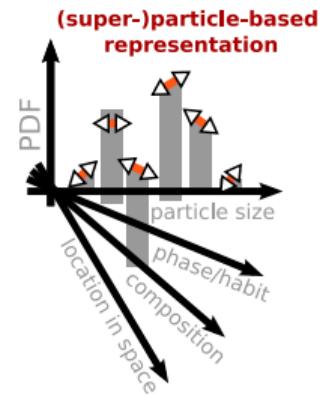
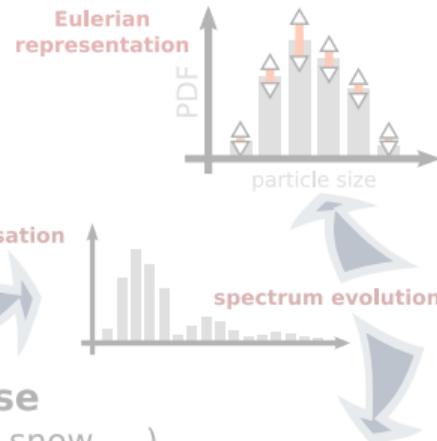
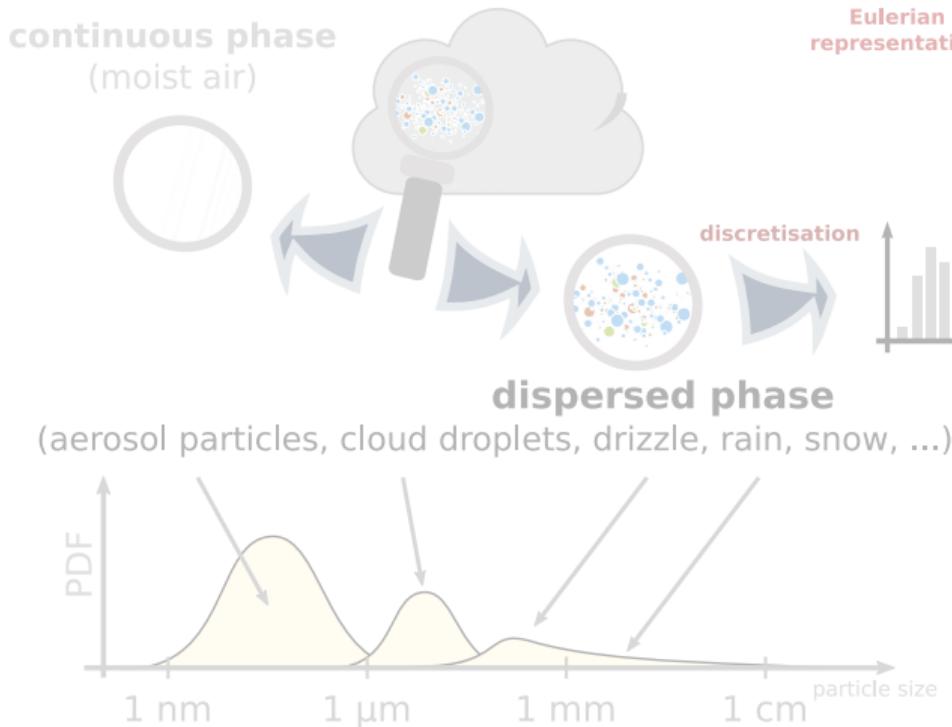
# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches

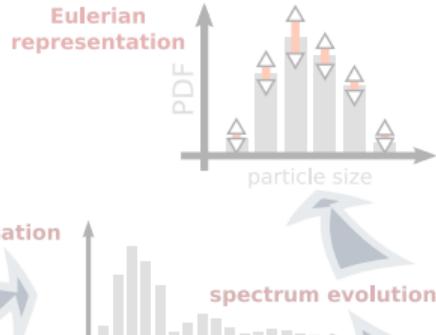


# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches

continuous phase  
(moist air)

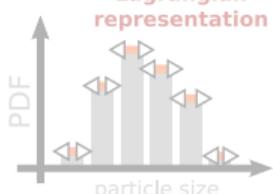


Eulerian representation



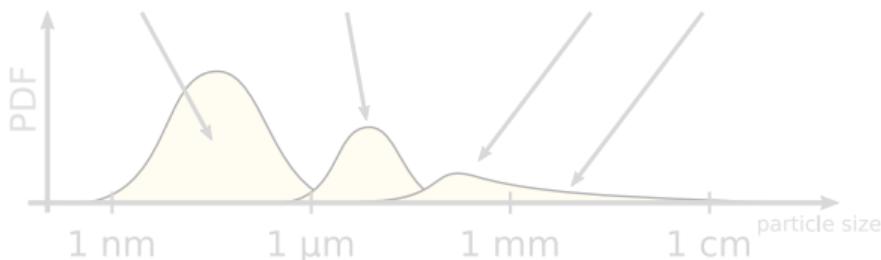
discretisation

Lagrangian representation

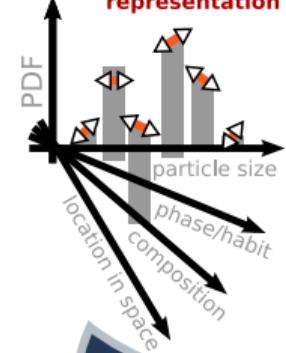


dispersed phase

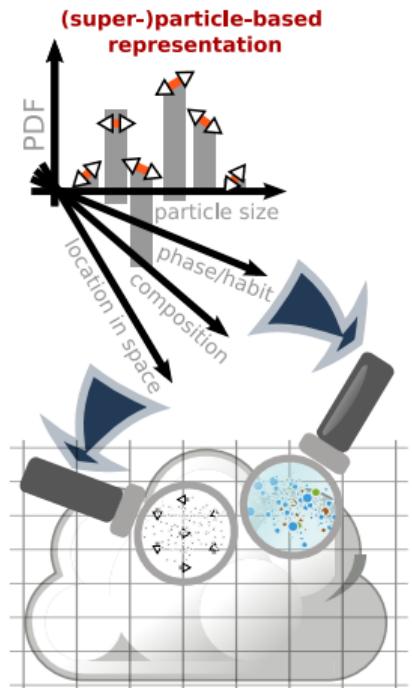
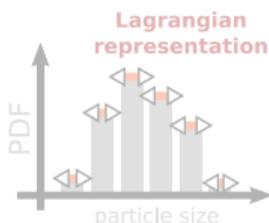
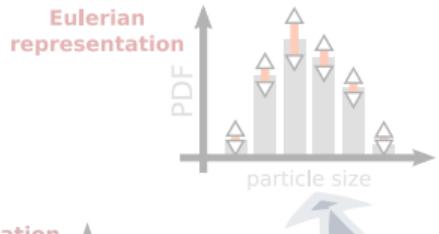
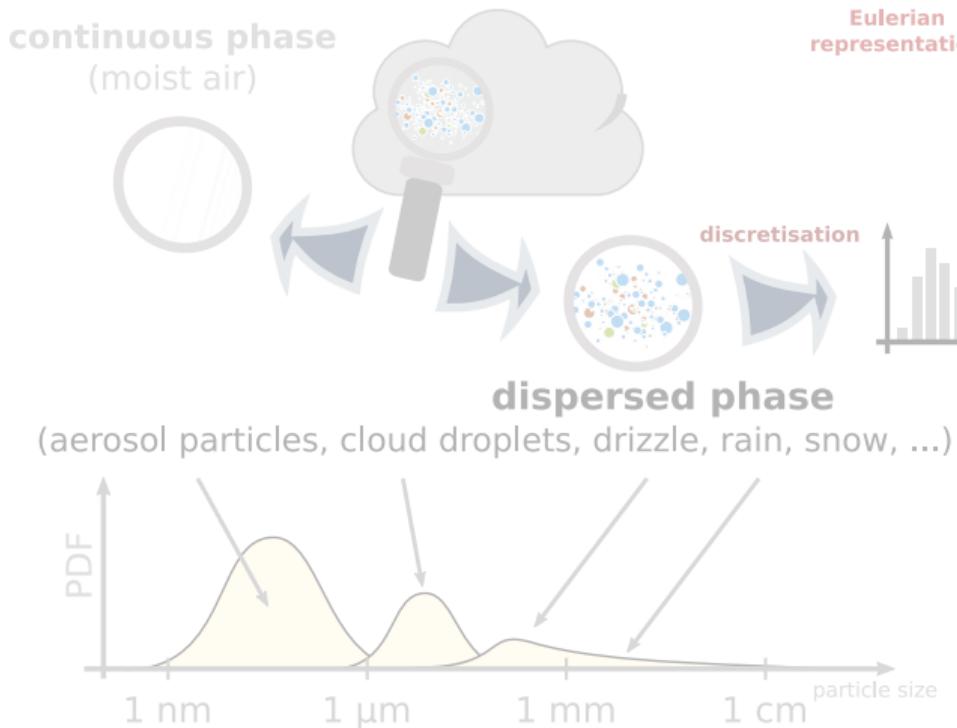
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



(super-)particle-based representation



# modelling cloud $\mu$ -physics: Eulerian vs. Lagrangian approaches



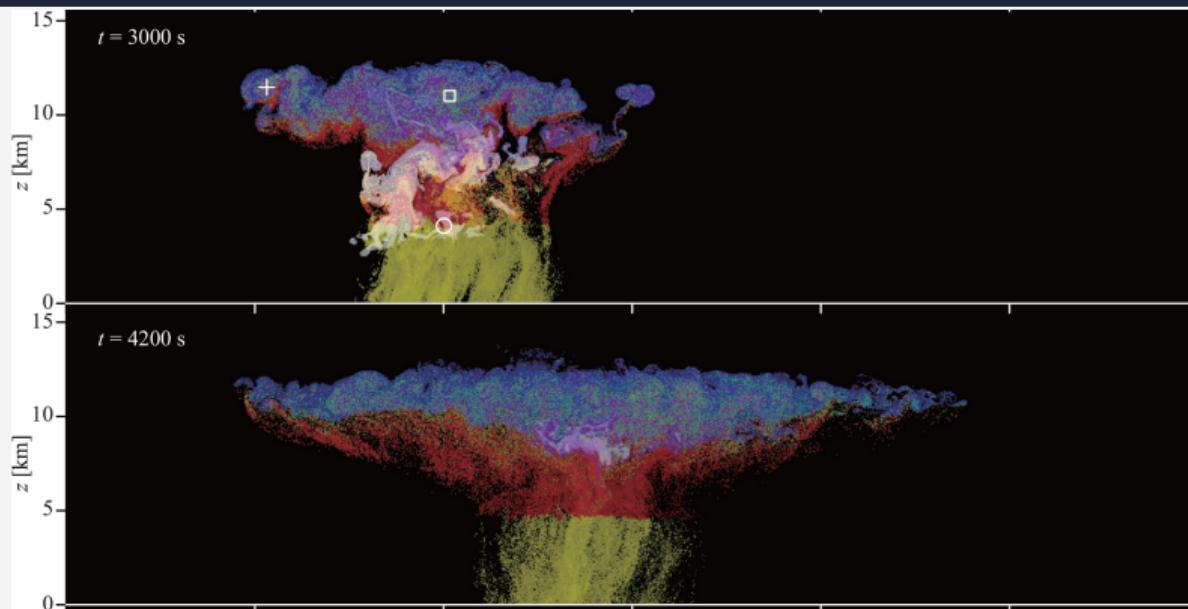
# Monte-Carlo SCE alternatives: e.g., SDM by Shima et al.

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

# PySDM: goals

**Develop an implementation of the SDM algorithm:**

# PySDM: goals

## **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

# PySDM: goals

## 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

# PySDM: goals

## 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

# PySDM: goals

## 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)

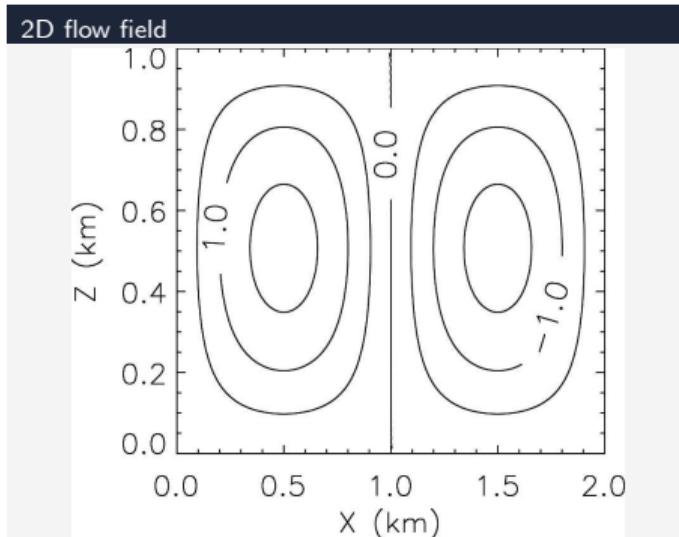
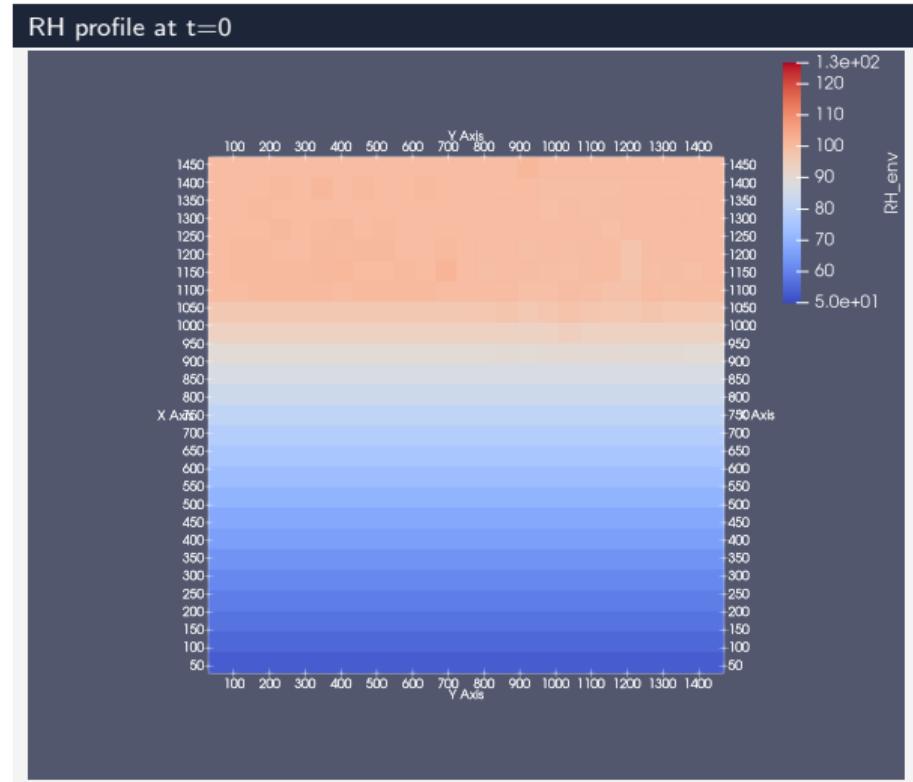
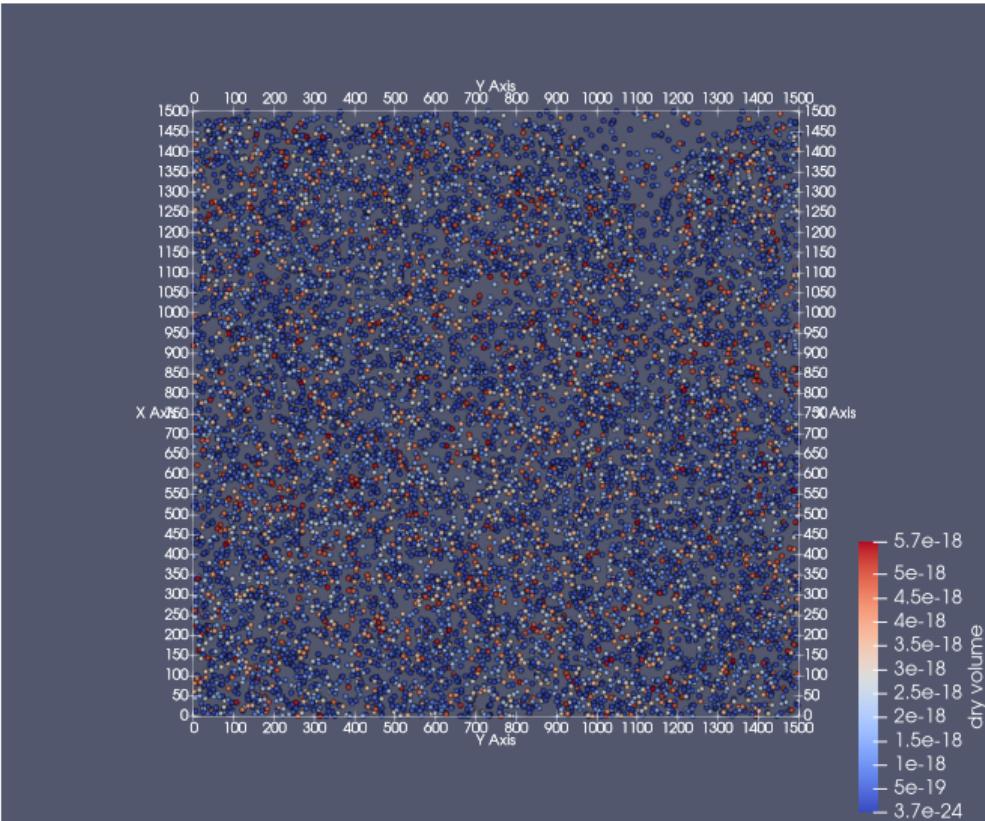


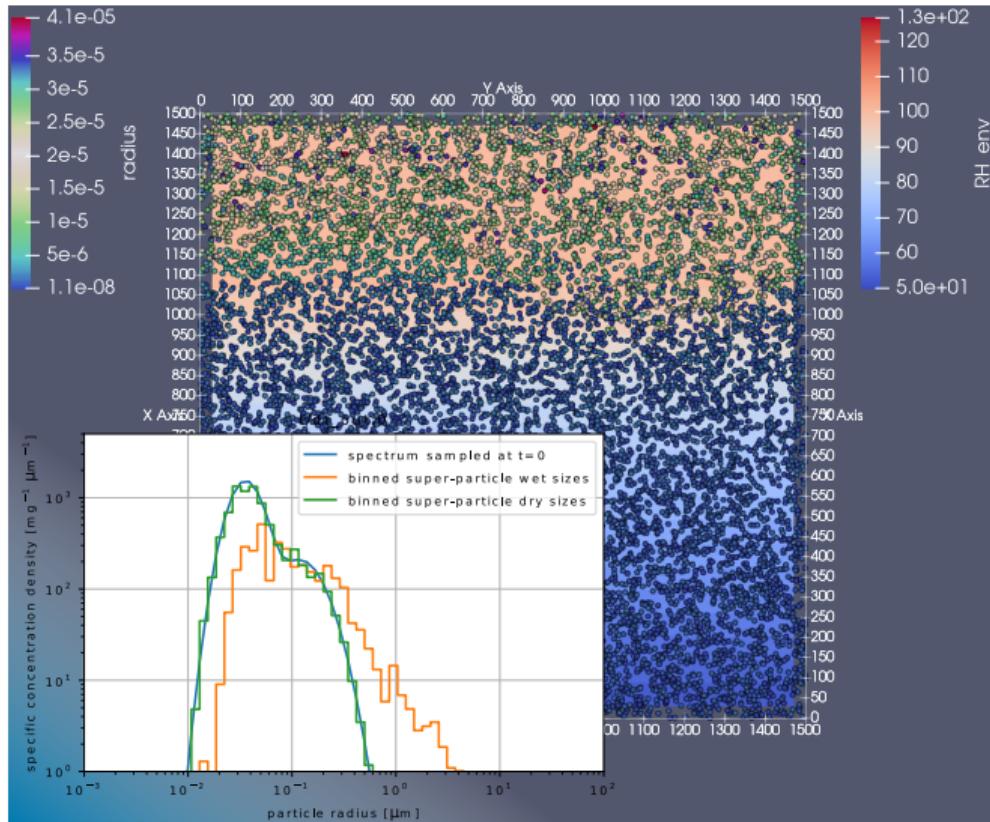
FIG. 1. Time-invariant vertical velocity for the stratocumulus case (contour interval is  $0.5 \text{ m s}^{-1}$ ).



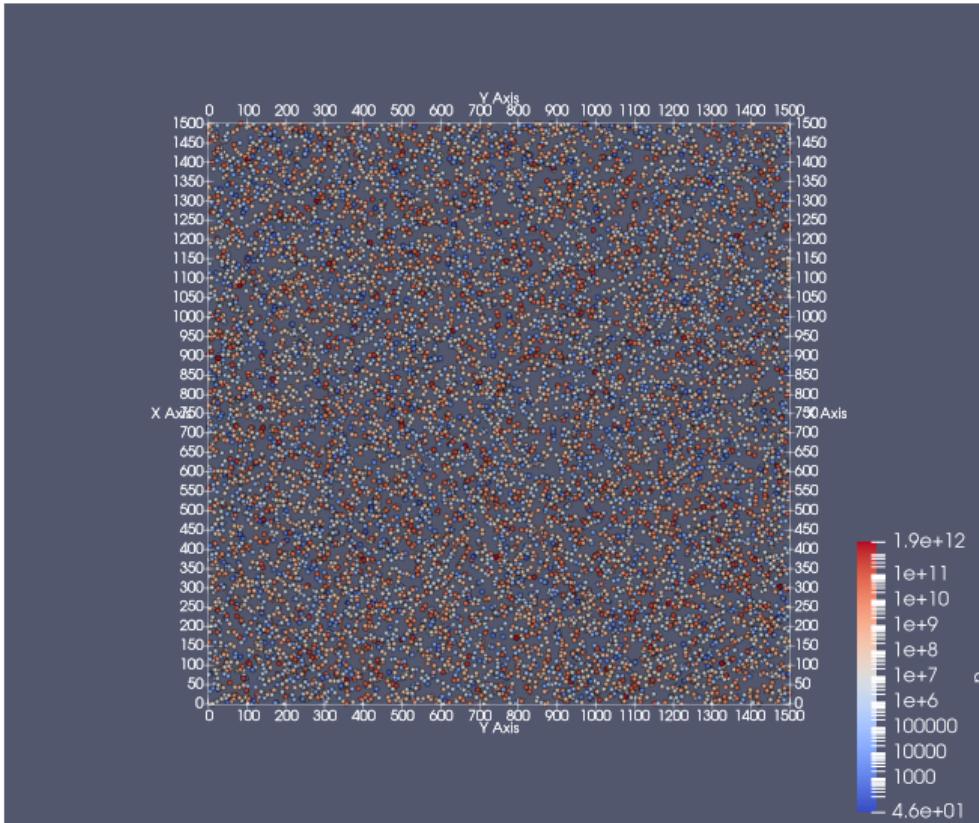
## particle attribute initialisation: dry/wet volume



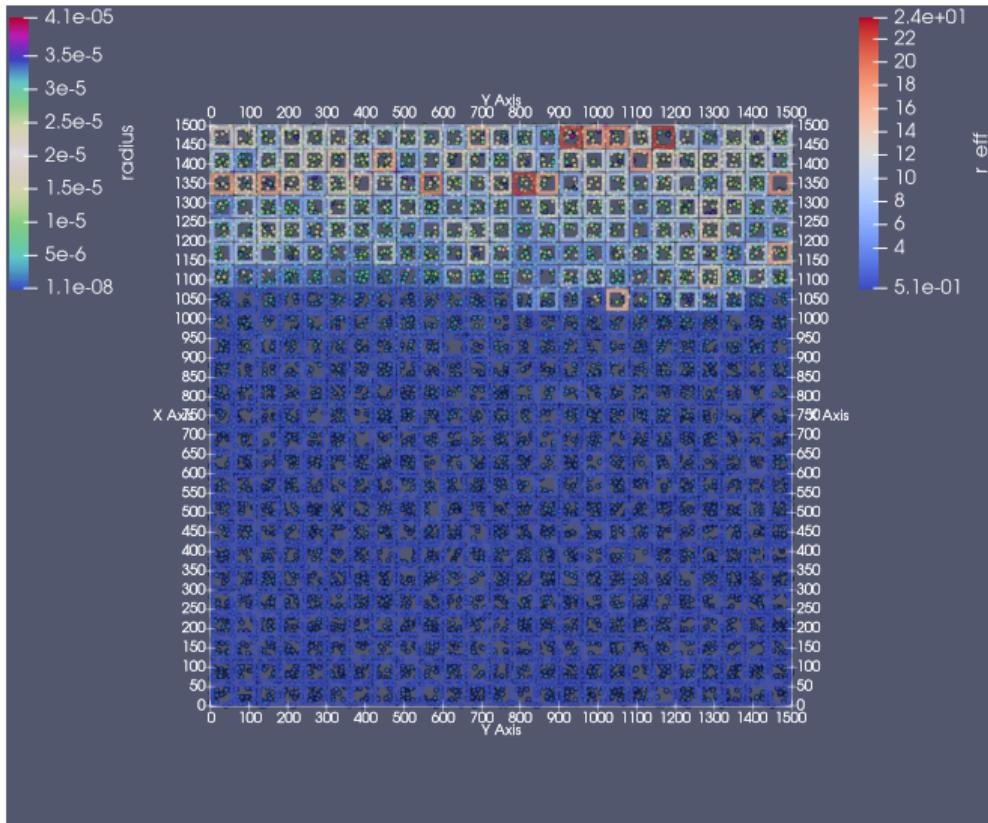
# particle attribute initialisation: dry/wet volume



# particle attribute initialisation: multiplicity

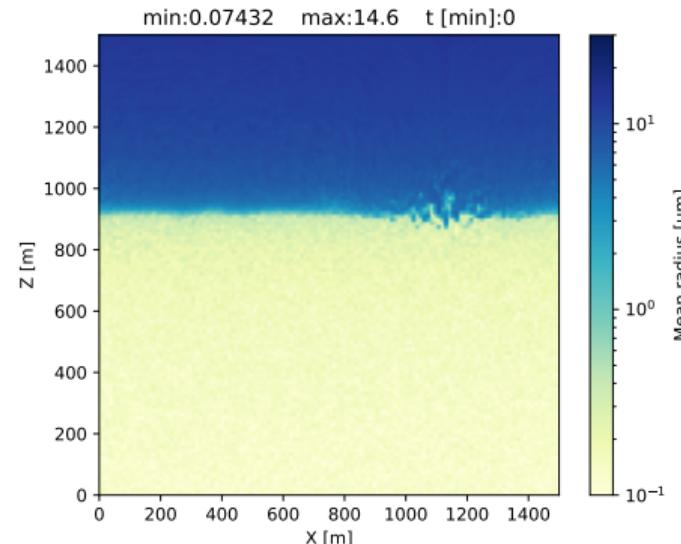
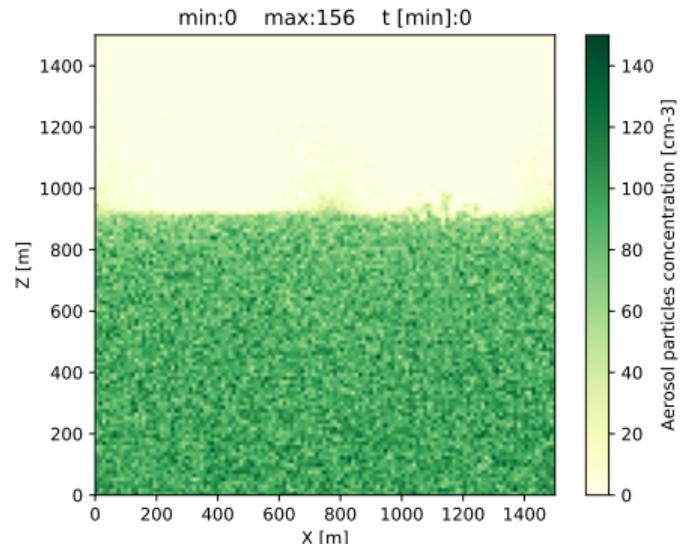


# particle attribute evolution: droplet radius



# sample aerosol-cloud-precipitation interactions simulation

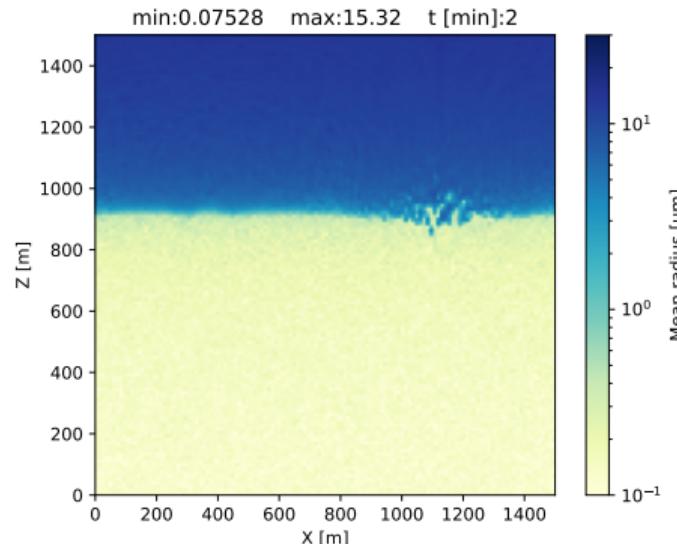
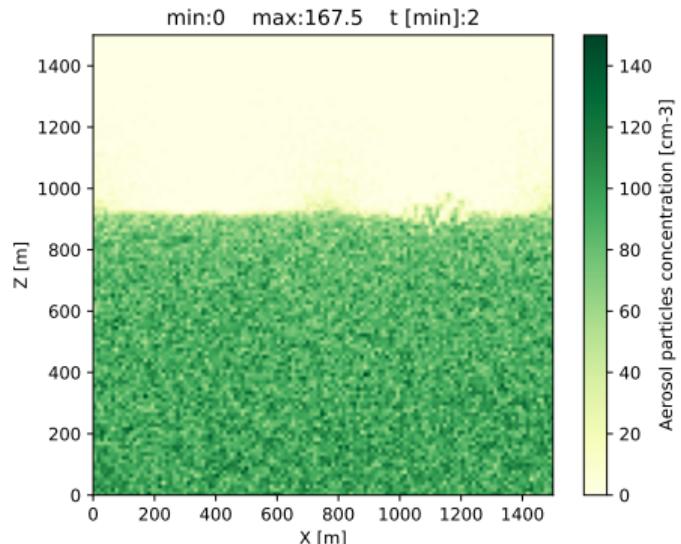
Computational grid: 128x128  
Computational particles:  $2^{21}$



Simulation & visualisation: Piotr Bartman (MSc thesis © WMiL UJ)

# sample aerosol-cloud-precipitation interactions simulation

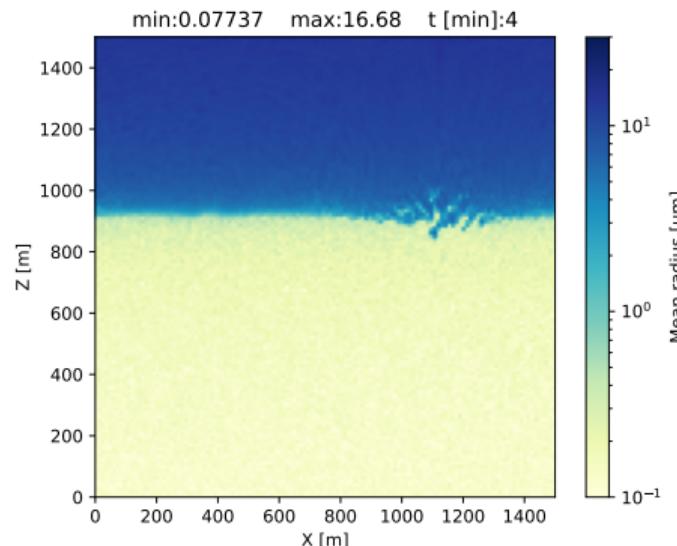
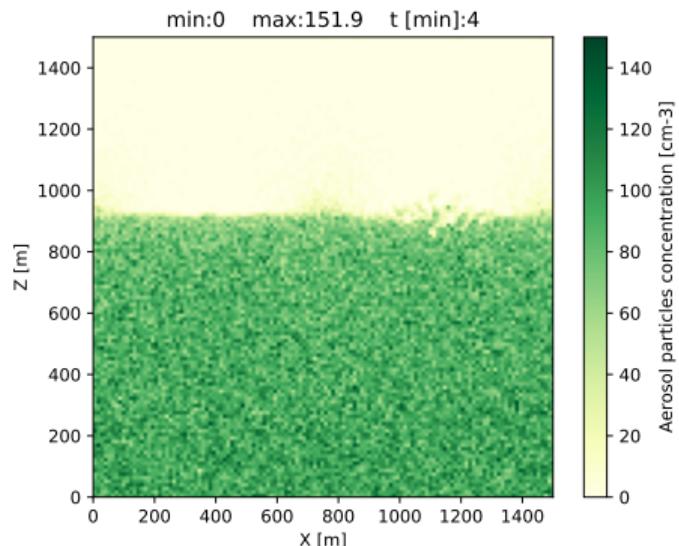
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128  
Computational particles:  $2^{21}$

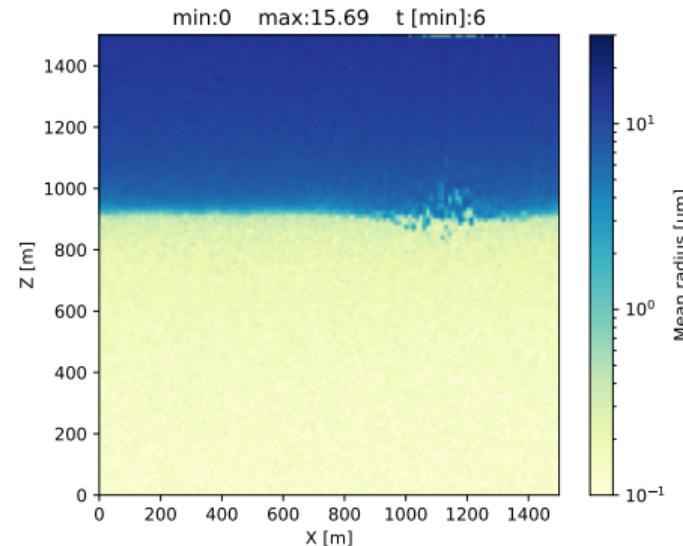
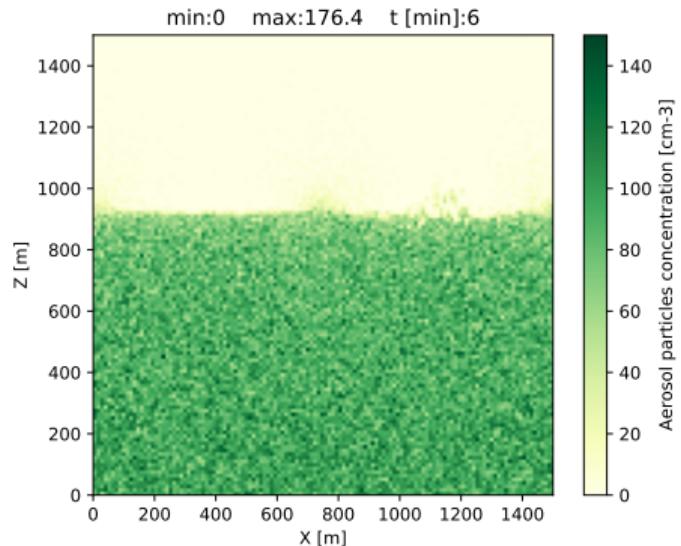


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiL UJ)

# sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

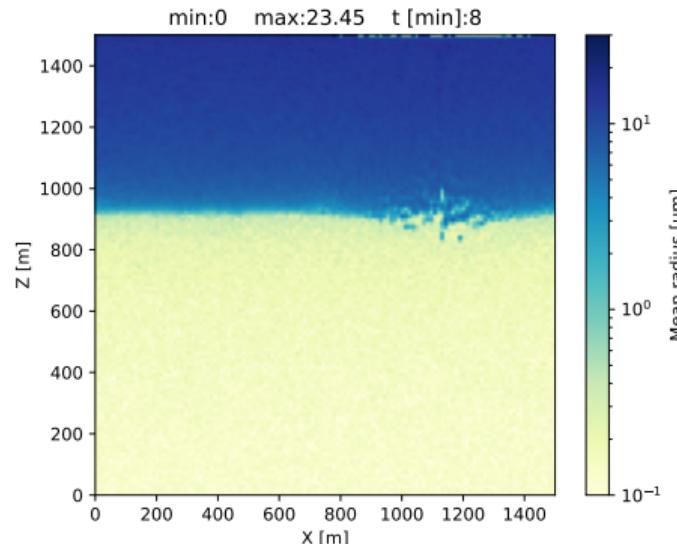
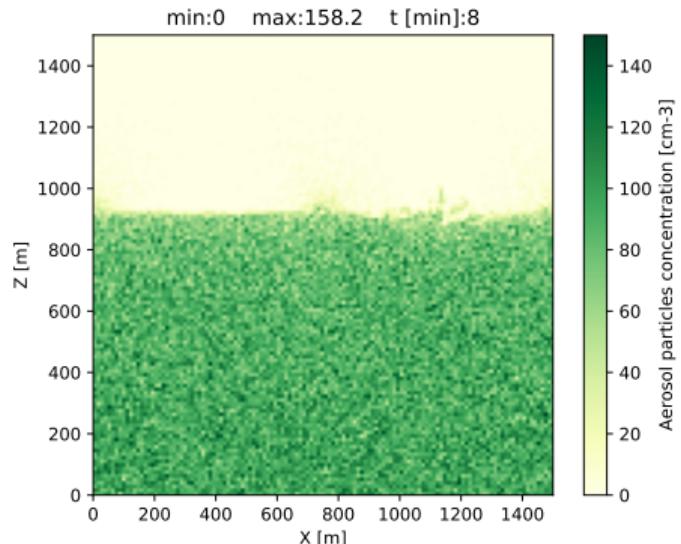
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

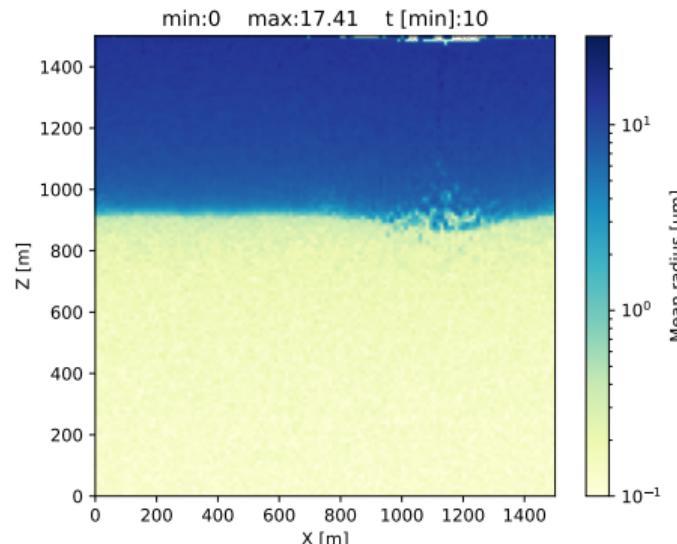
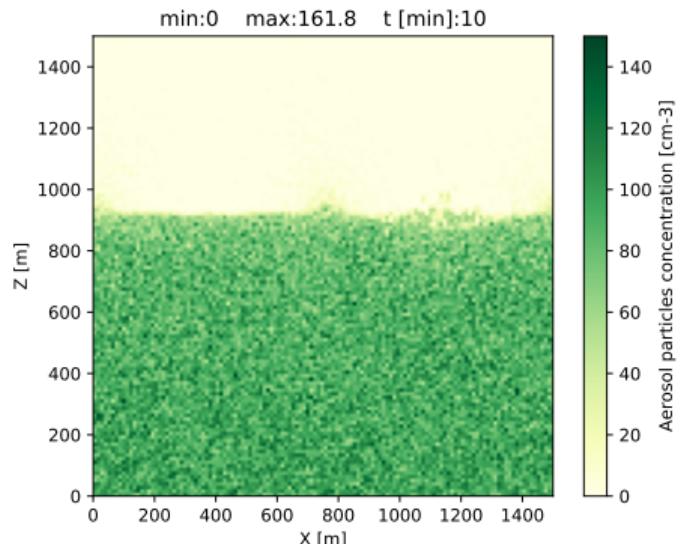
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

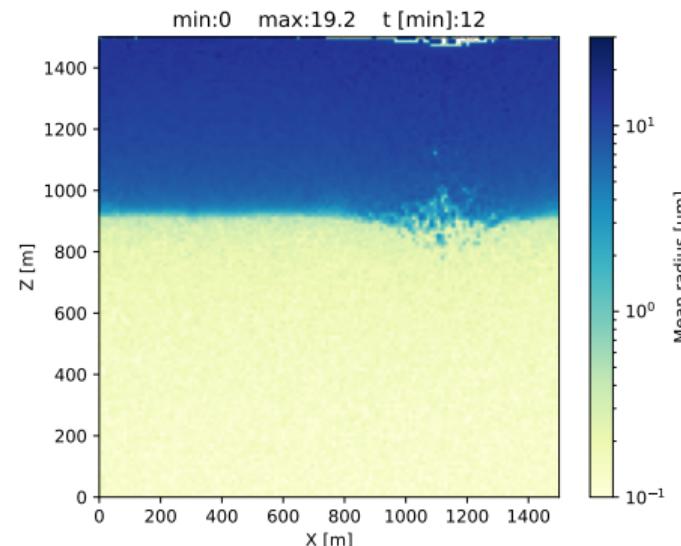
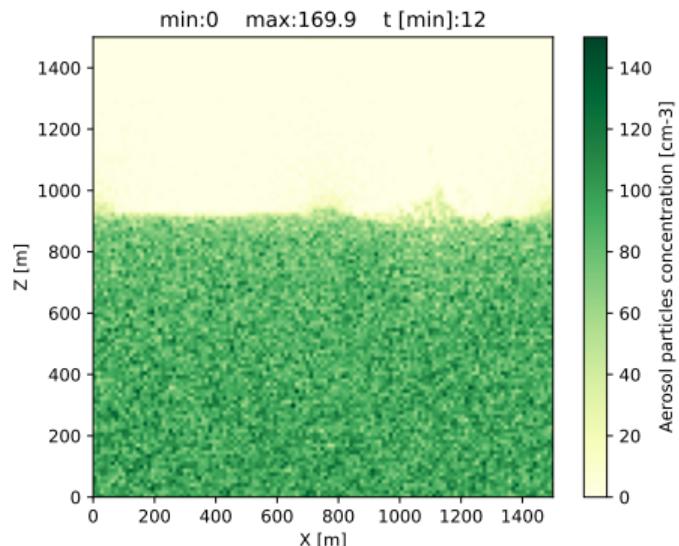
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

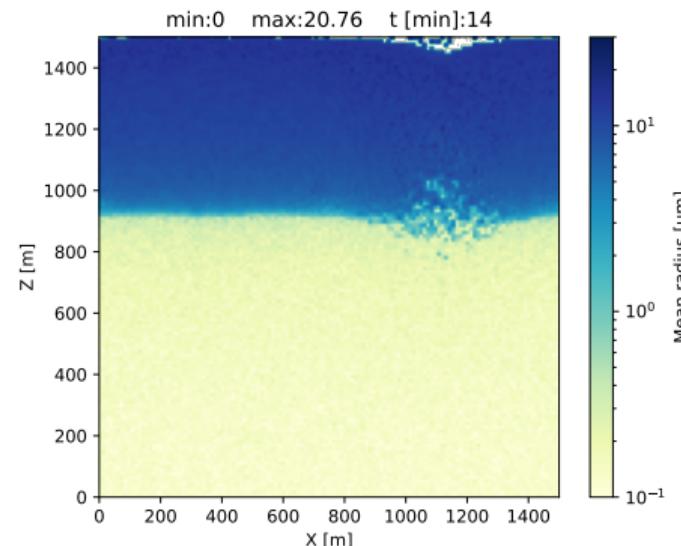
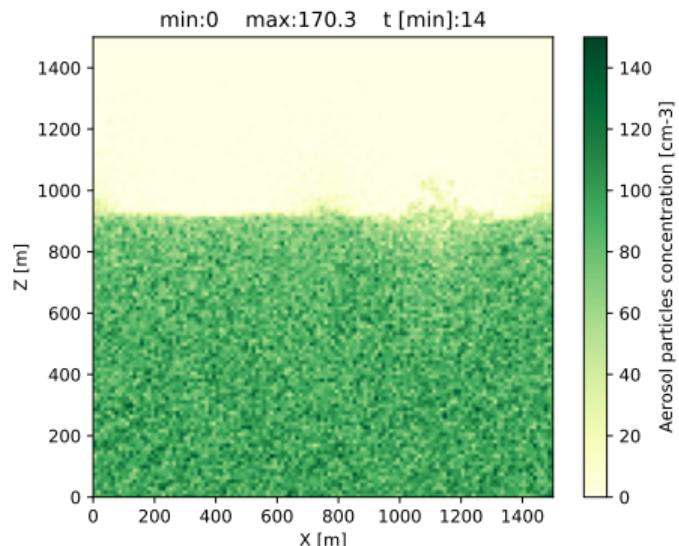
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

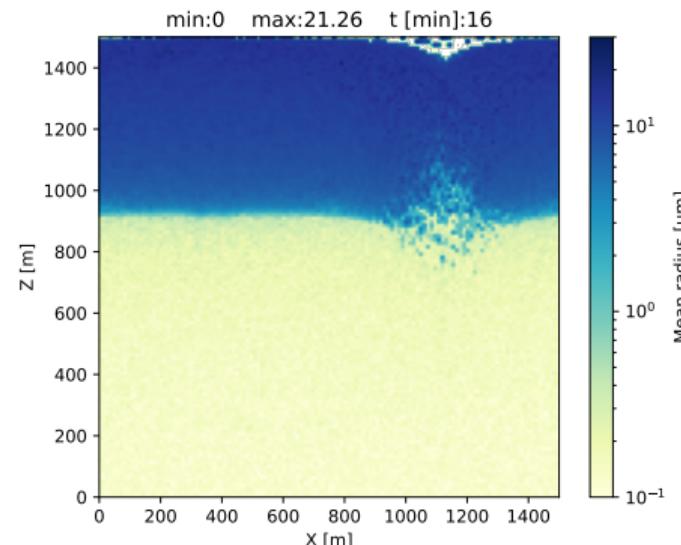
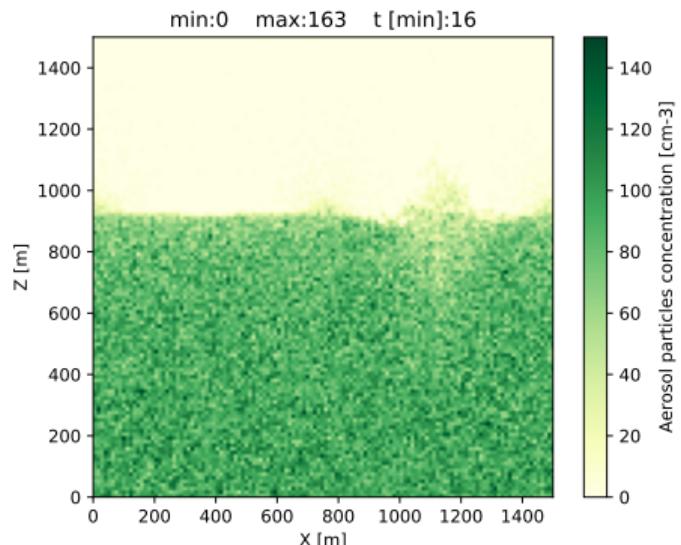
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

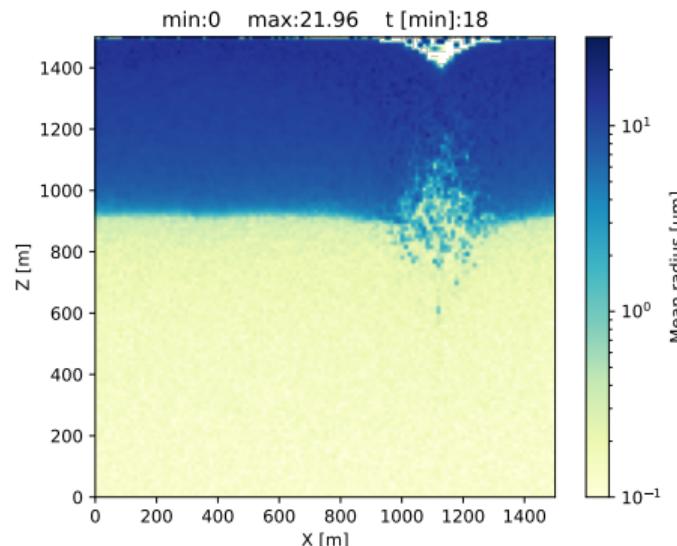
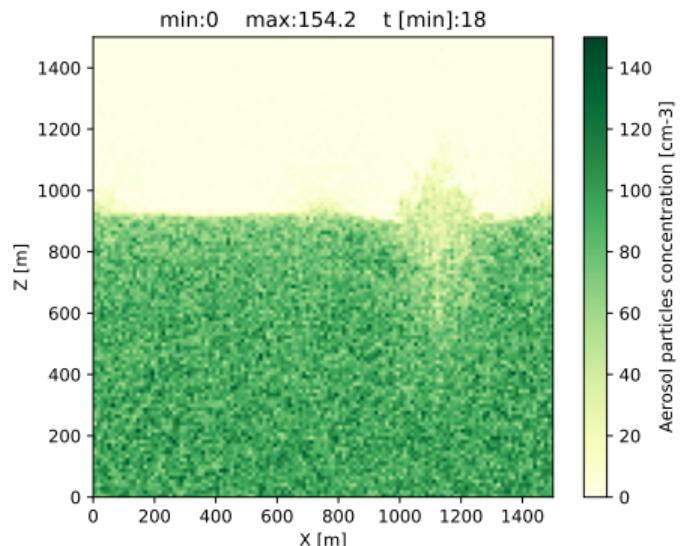
Computational grid: 128x128  
Computational particles:  $2^{21}$



Simulation & visualisation: Piotr Bartman (MSc thesis © WMiL UJ)

# sample aerosol-cloud-precipitation interactions simulation

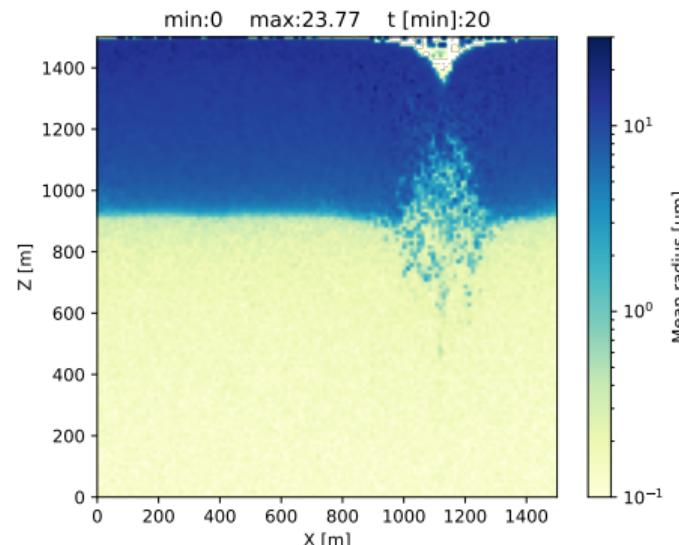
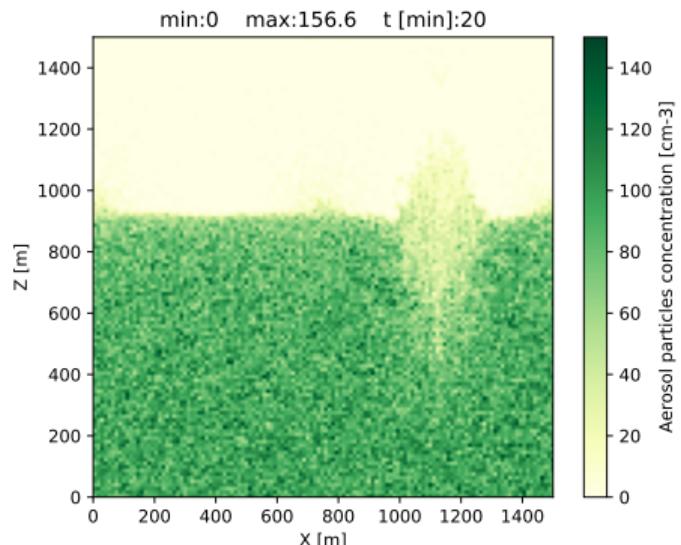
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

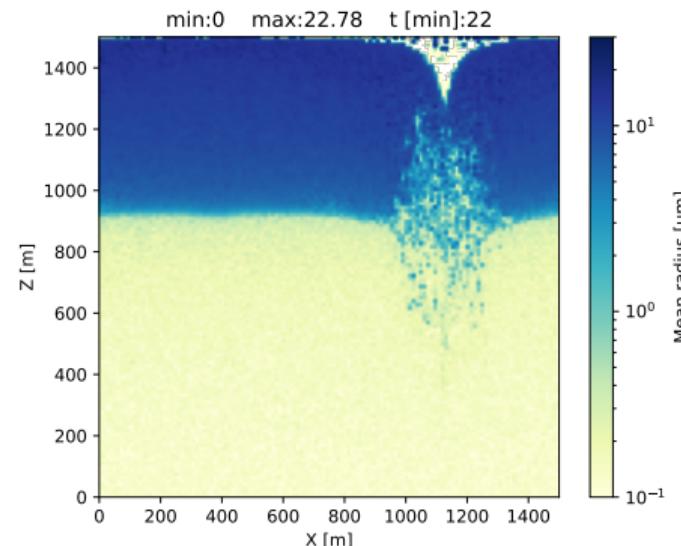
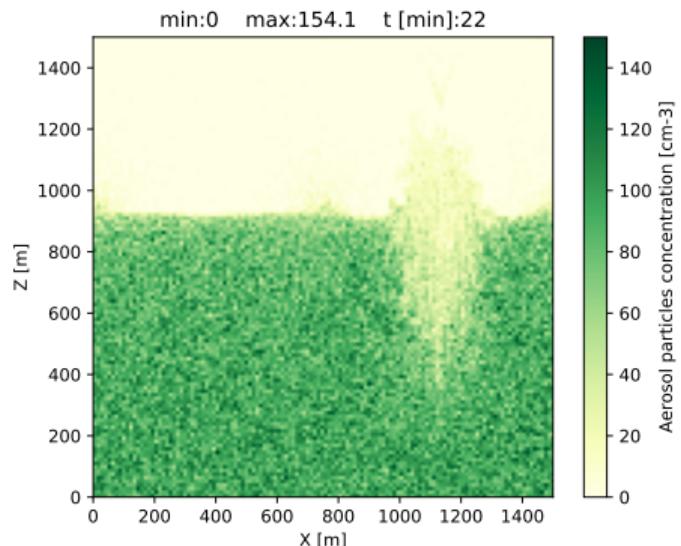
Computational grid: 128x128  
Computational particles:  $2^{21}$



Simulation & visualisation: Piotr Bartman (MSc thesis © WMiL UJ)

# sample aerosol-cloud-precipitation interactions simulation

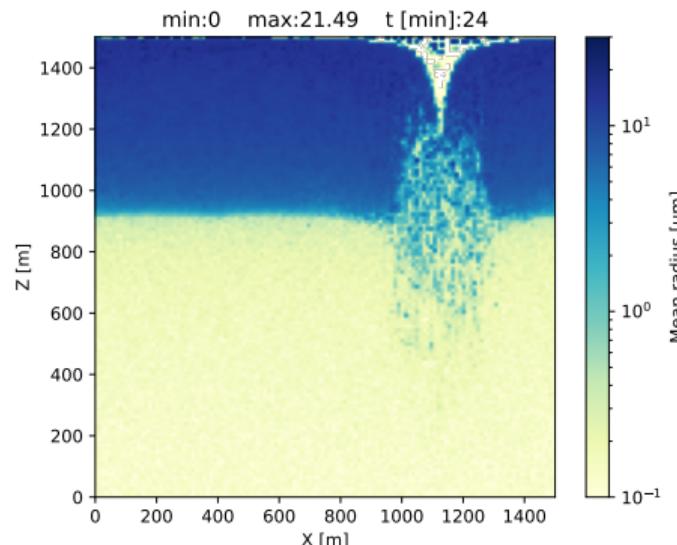
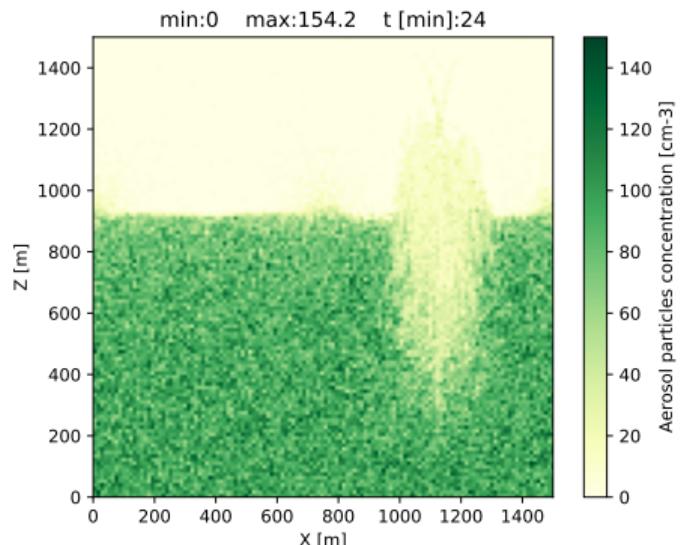
Computational grid: 128x128  
Computational particles:  $2^{21}$



Simulation & visualisation: Piotr Bartman (MSc thesis © WMiL UJ)

# sample aerosol-cloud-precipitation interactions simulation

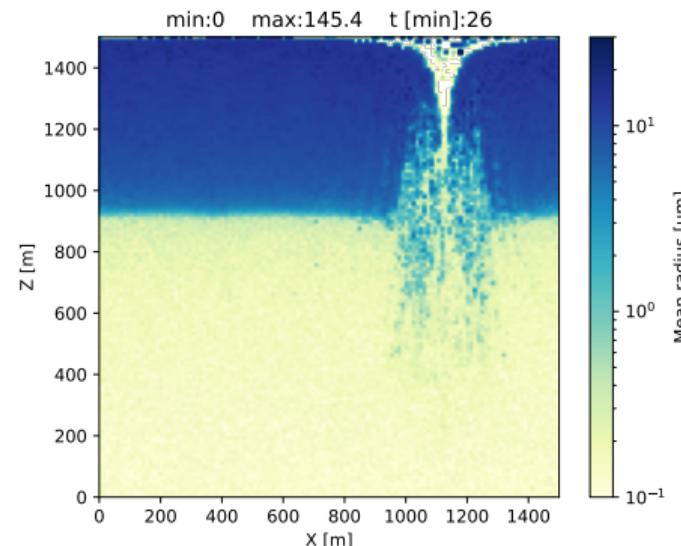
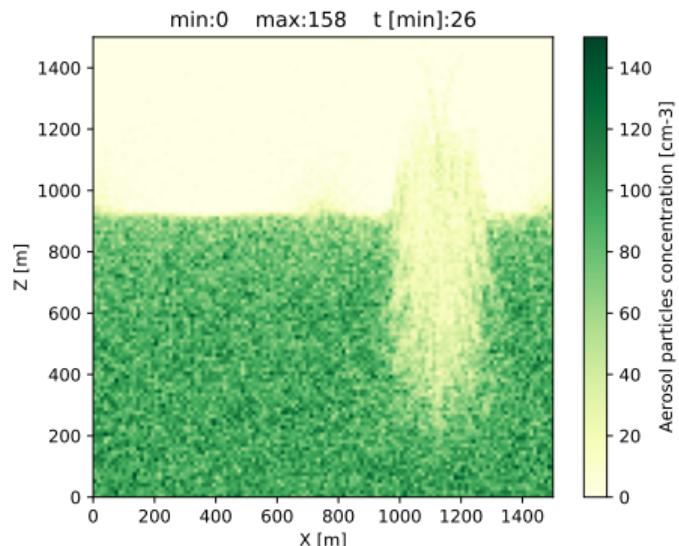
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

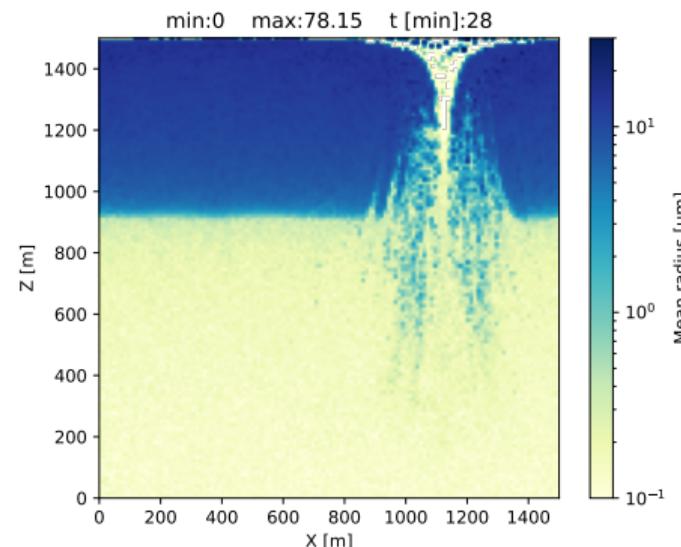
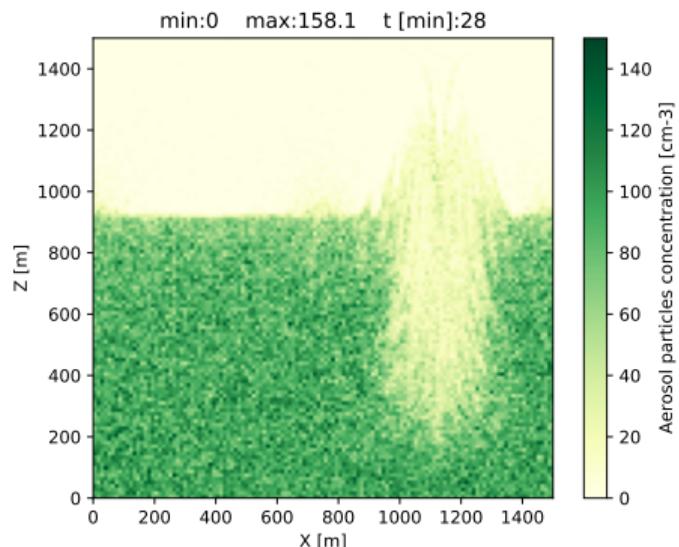
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

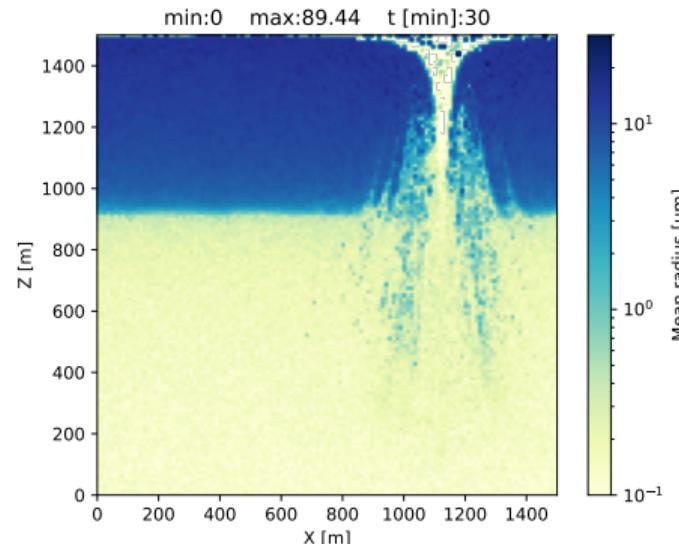
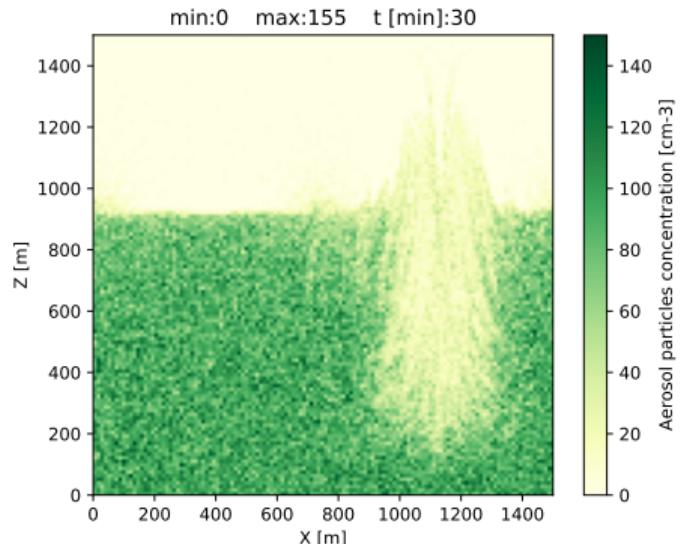
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

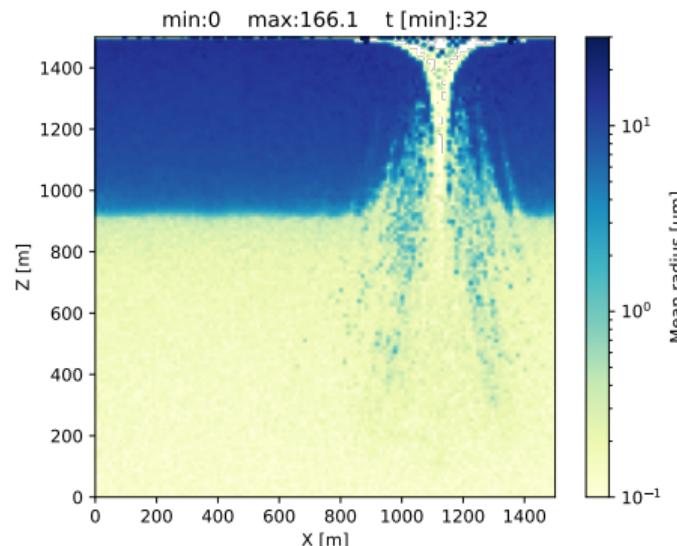
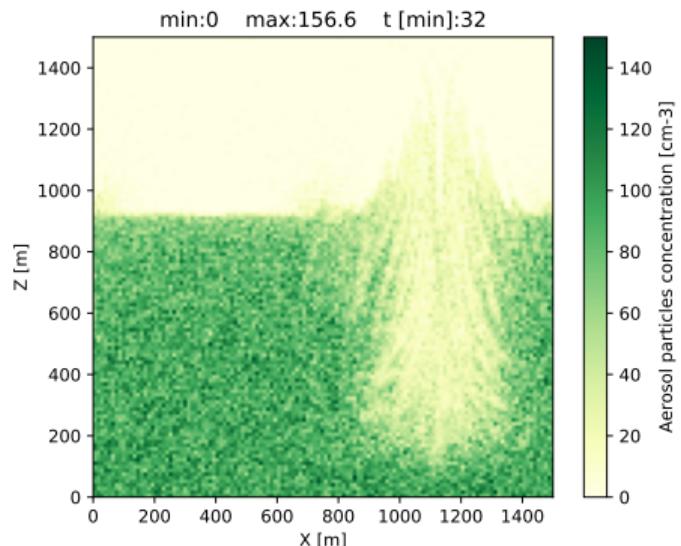
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

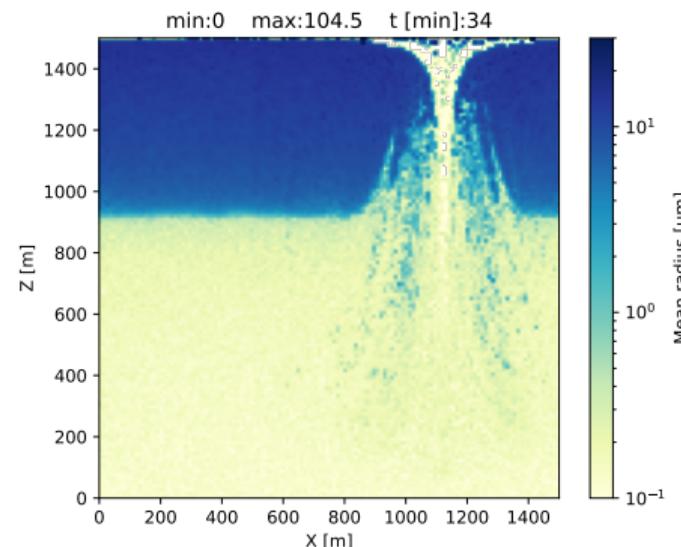
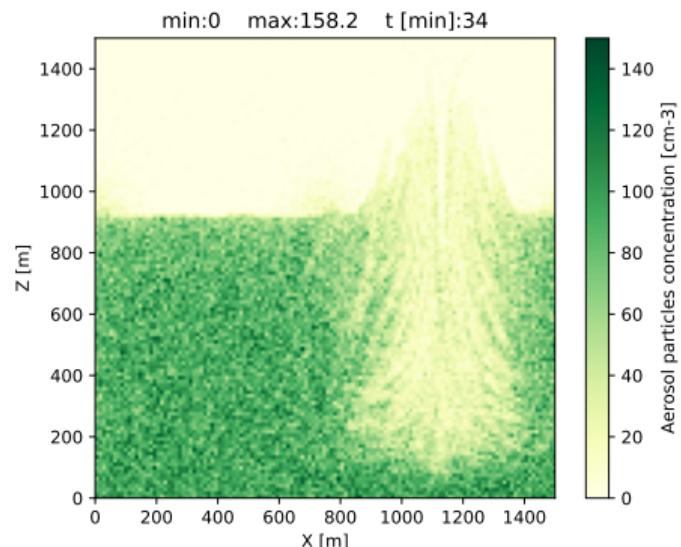
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

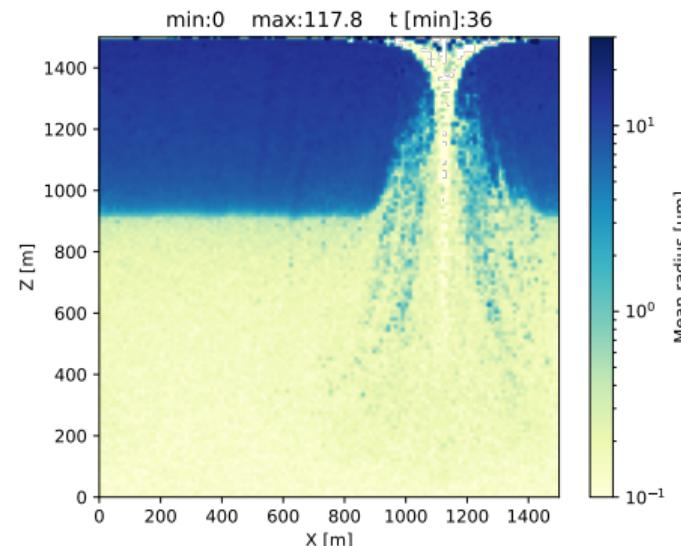
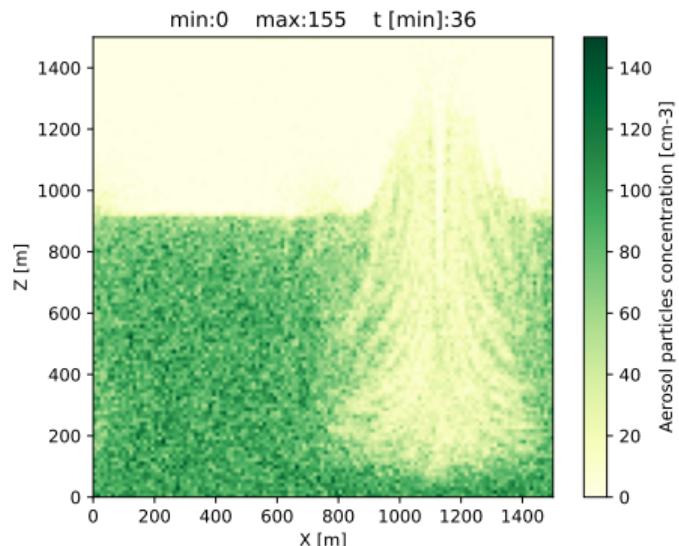
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

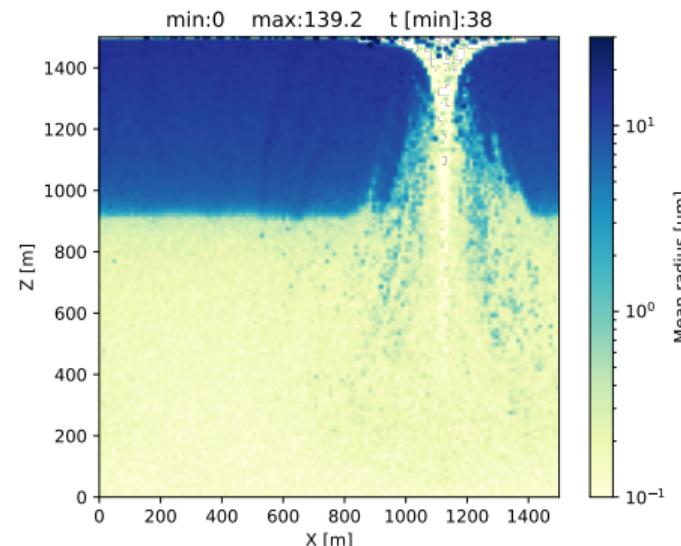
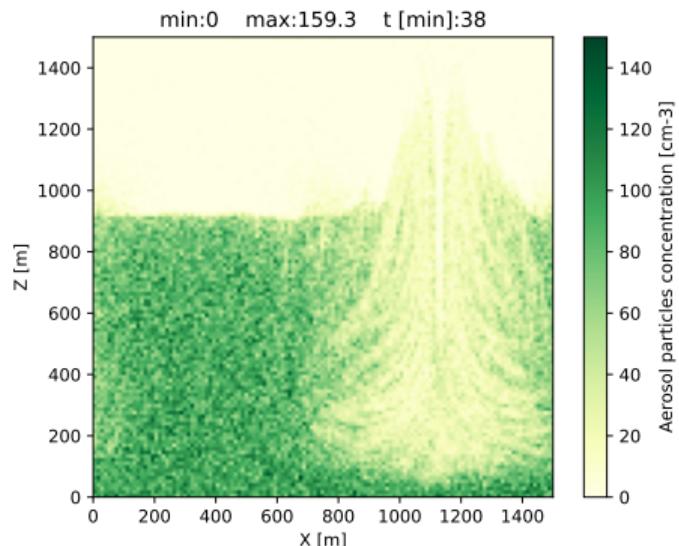
Computational grid: 128x128  
Computational particles:  $2^{21}$



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

# sample aerosol-cloud-precipitation interactions simulation

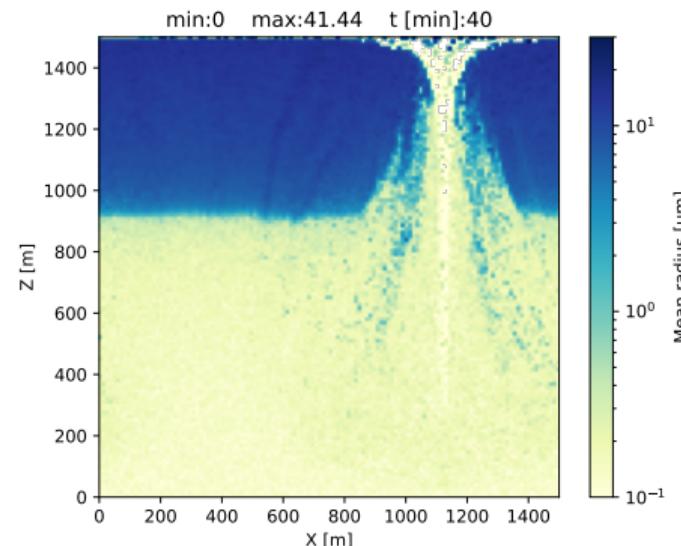
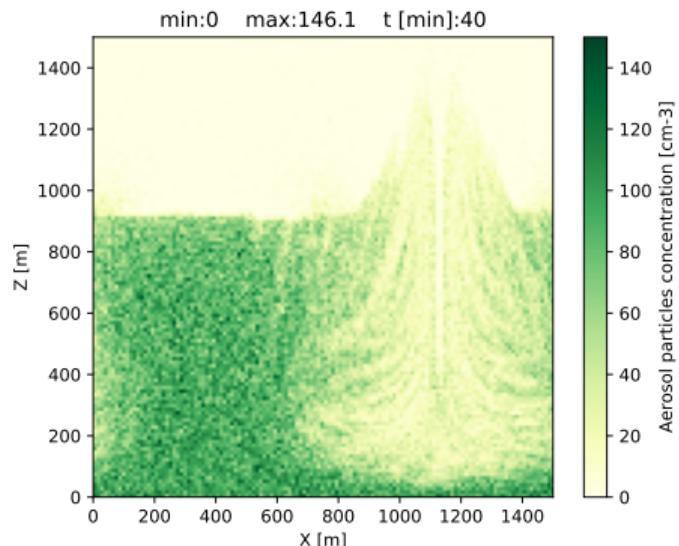
Computational grid: 128x128  
Computational particles:  $2^{21}$



Simulation & visualisation: Piotr Bartman (MSc thesis © WMiL UJ)

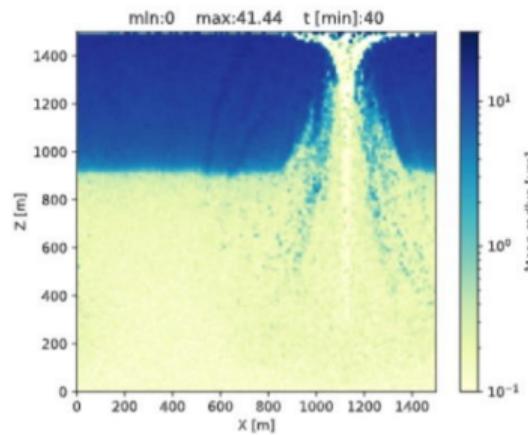
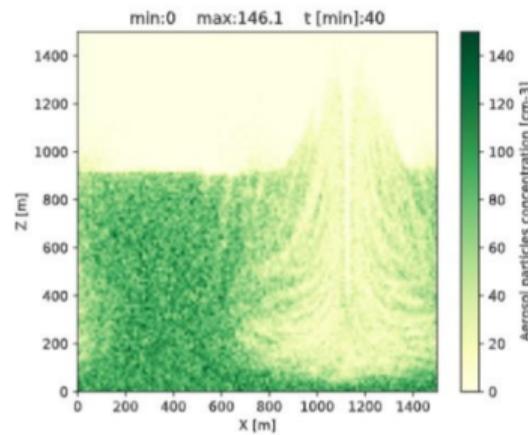
# sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128  
Computational particles:  $2^{21}$



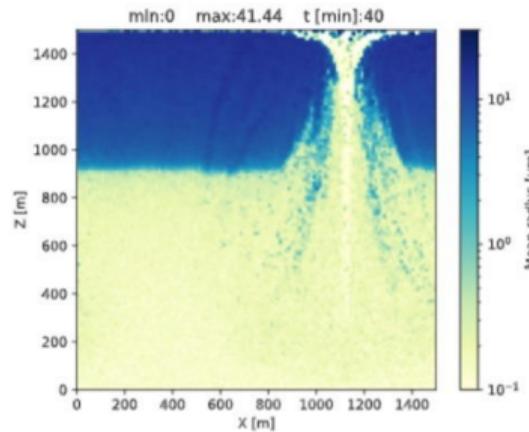
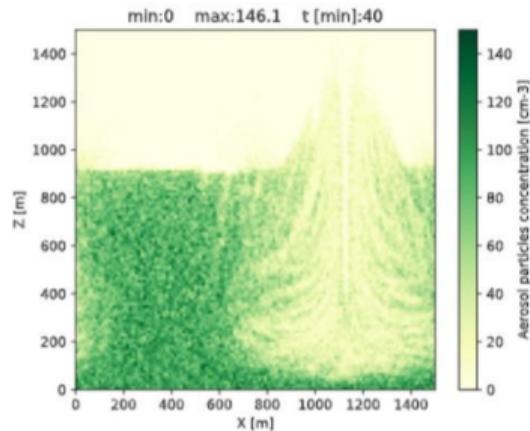
Simulation & visualisation: Piotr Bartman (MSc thesis © WMiL UJ)

# PySDM:

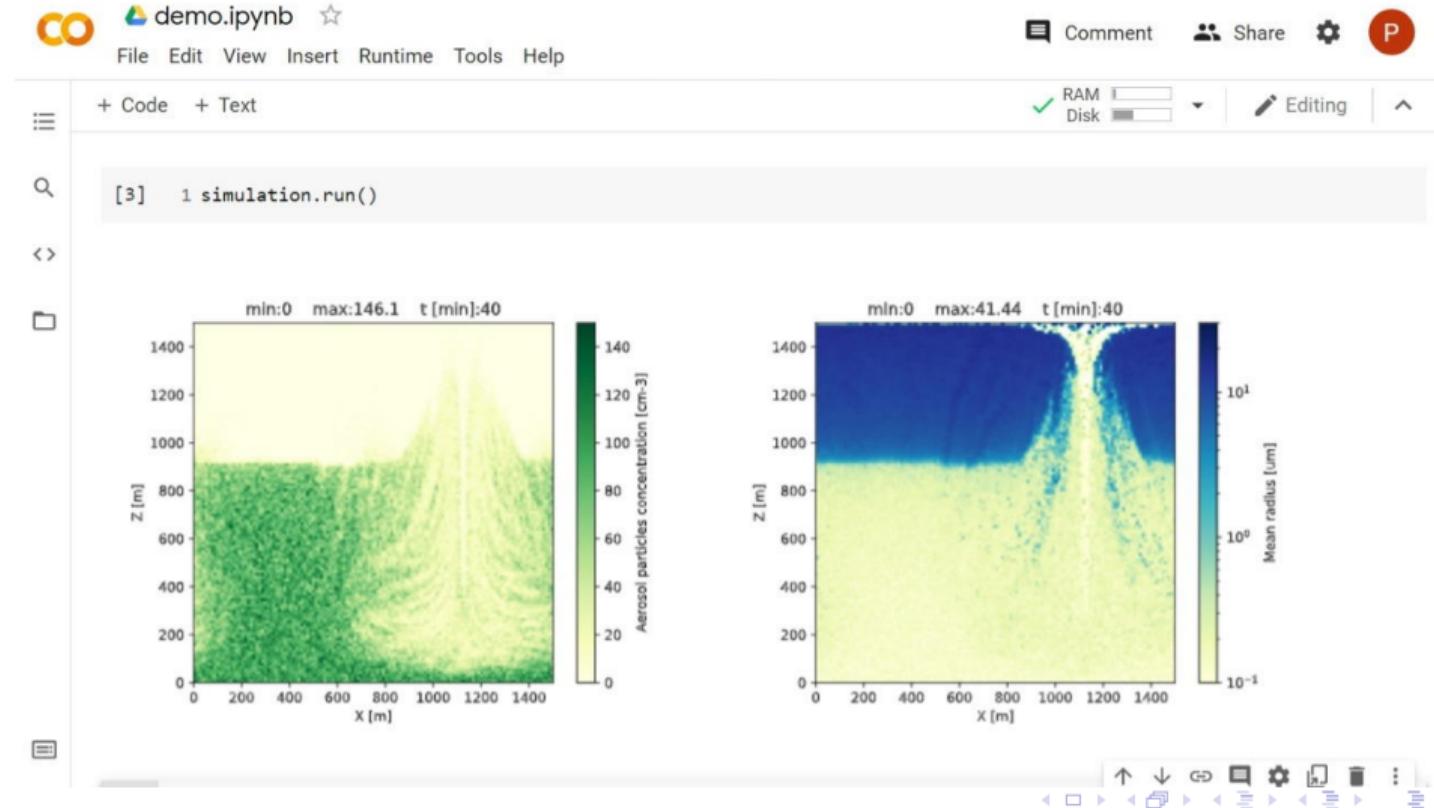


# PySDM: Pythonic

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



# PySDM: Pythonic, Jupyter-friendly



# PySDM: Pythonic, Jupyter-friendly, GPU-enabled

The screenshot shows a Jupyter Notebook interface in Google Colab. The notebook title is "demo.ipynb". The menu bar includes File, Edit, View, Insert, Runtime, Tools, Help, and a status message "All changes saved". The toolbar includes Comment, Share, and a profile icon. A progress bar indicates RAM and Disk usage.

In the code cell [3], the command `simulation.run()` is run, resulting in a visualization of a simulation. The plot shows a 2D field with axes X [m] and Z [m]. A color scale at the bottom left indicates values from 0 to 40, labeled "Aerosol". A vertical color bar on the right indicates "Mean radius [μm]" on a logarithmic scale from  $10^{-1}$  to  $10^1$ .

A modal dialog titled "Notebook settings" is open. It contains the following options:

- Hardware accelerator: Set to GPU (with a dropdown arrow and a help icon).
- To get the most out of Colab, avoid using a GPU unless you need one. [Learn more](#)
- Omit code cell output when saving this notebook

At the bottom of the dialog are "CANCEL" and "SAVE" buttons.

# 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

[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

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

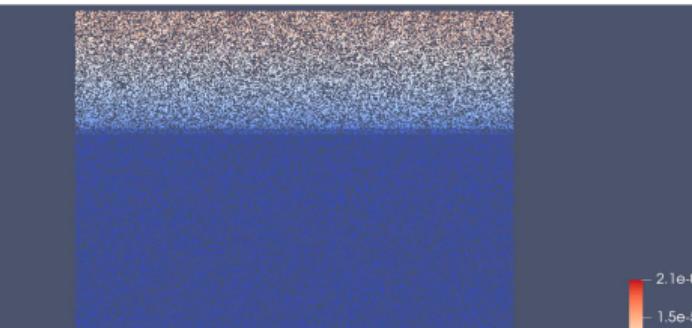
PySDM is a particle-based aerosol/cloud microphysics package written entirely in Python.

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:

output.mp4



Reviewers

- slayoo
- charleskawczynski
- claresinger
- jakebolewski
- edeljongs@caltech
- tapios

Assignees

- trontryte

Labels

- Microphysics

Projects

- None yet

Milestone

- No milestone

Development

Successfully merging this pull request may close these issues.

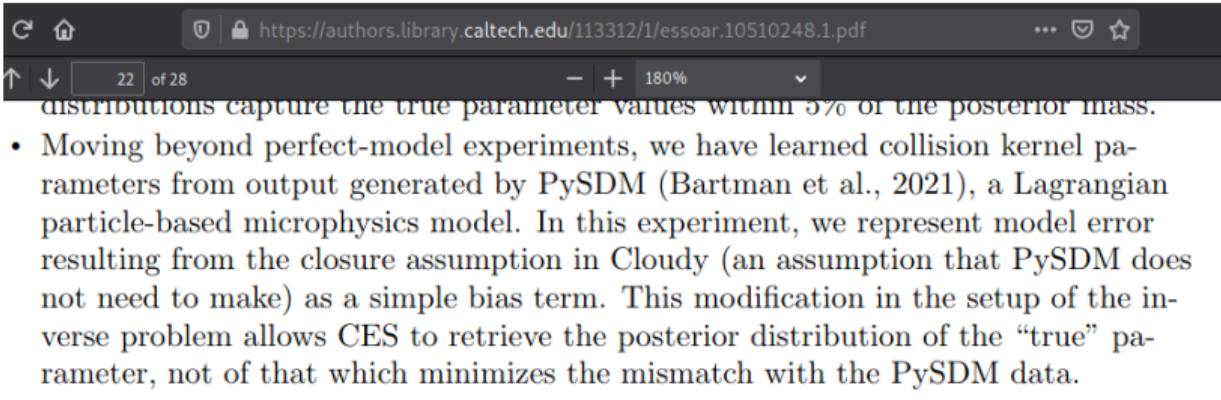
None yet

# first independent development!

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 Bieli<sup>1</sup>, Oliver R. A. Dunbar<sup>1</sup>, Emily K. de Jong<sup>2</sup>, Anna Jaruga<sup>1</sup>,  
Tapio Schneider<sup>1</sup>, Tobias Bischoff<sup>1</sup>



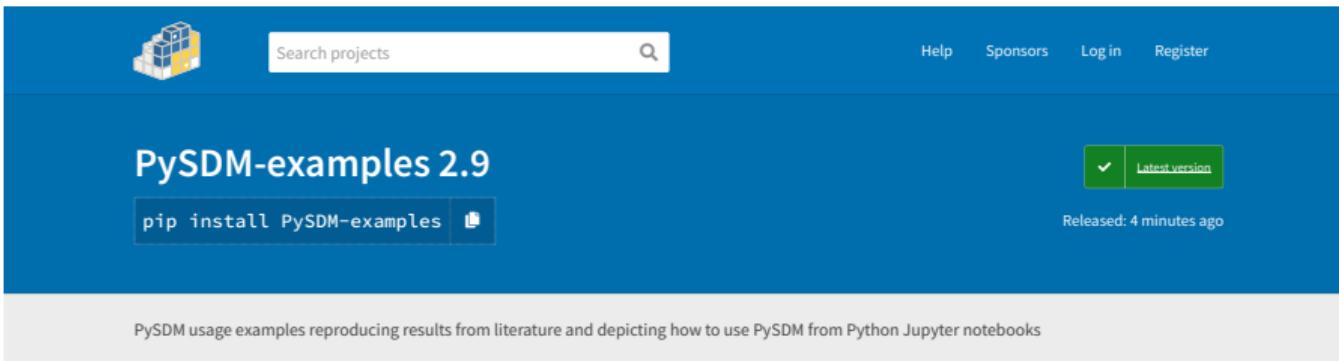
The screenshot shows a web browser window with the following details:

- Address bar: https://authors.library.caltech.edu/113312/1/essoar.10510248.1.pdf
- Page number: 22 of 28
- Zoom level: 180%
- Content preview:

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.

<https://pypi.org/p/PySDM-examples>



The screenshot shows the PySDM-examples project page on the Python Package Index (pypi.org). The top navigation bar includes links for Help, Sponsors, Log in, and Register. The main title is "PySDM-examples 2.9". Below it is a button for "pip install PySDM-examples" and a green "Latest version" badge. A message indicates the package was released 4 minutes ago. The page content describes the repository as containing PySDM usage examples. The left sidebar has a "Navigation" section with "Project description" selected, showing a tree icon. Other options include "Release history" (with a circular arrow icon) and "Download files" (with a download icon). The "Project description" section contains links for "License: GPL v3", "Copyright: Jagiellonian University", and "DOI: 10.5281/zenodo.6604645". It also displays GitHub statistics: 2 open pull requests and 159 closed pull requests. The "pypl package" rating is 2.8, and there is an "API docs: pdoc3" link. The "Project links" section includes a "Homepage" link. The "Statistics" section shows GitHub stats: 2 stars and 10 forks. The "Project description" text states that the repository stores example files for PySDM, depicting its usage from Python via Jupyter. It also mentions the PySDM package itself and examples of usage from Julia and Matlab, pointing to the "PySDM README.md" file. It encourages using the PySDM issue-tracking and discussion infrastructure. The "0D box-model coalescence-only examples" section lists three references with their respective figures and code execution links.

## Navigation

Project description

Release history

Download files

## Project links

Homepage

## Statistics

GitHub statistics:

Stars: 2

Forks: 10

Issues: 0

## Project description

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

PySDM-examples [passing](#)

pull requests [2 open](#) pull requests [159 closed](#)

pypl 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<sup>1,2</sup>, D.G. Partridge<sup>3</sup>, J.F. Davies<sup>4</sup>, K.R. Wilson<sup>5</sup>, D. Topping<sup>6</sup> & I. Riipinen<sup>1,2,7\*</sup>

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.

# PySDM-examples: Lowe et al. 2019

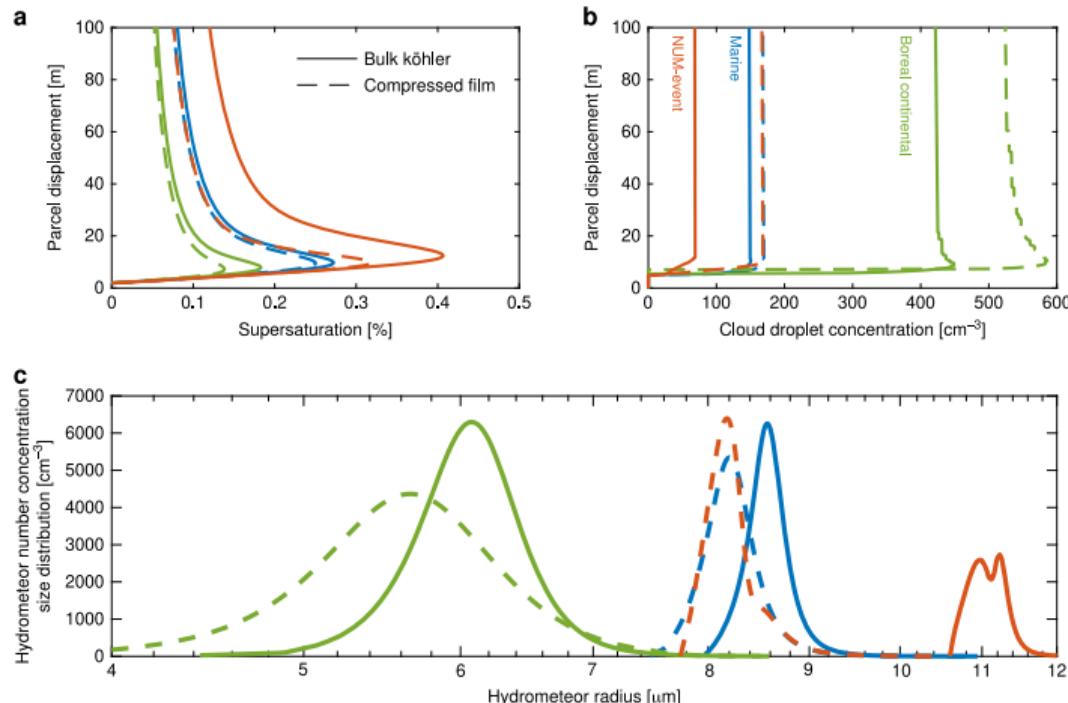


Fig. 2 Simulated microphysics of cloud events on marine (MA, blue), boreal (HYY, green) and NUM-event (NE, orange) aerosol populations. Cloud-formation event simulations using bulk Köhler BK (solid lines) and approximate compressed film CF (dotted lines) models of cloud droplet activation with initial temperature  $T = 280 \text{ K}$ , pressure  $P = 98,000 \text{ Pa}$ , 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

# PySDM-examples: Lowe et al. 2019

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

The screenshot shows a Jupyter Notebook interface with the following details:

- File Menu:** File, Edit, View, Run, Kernel, Tabs, Settings, Help.
- Launcher:** Shows tabs for 'Launcher' (active), 'fig\_1.ipynb', 'GitHub', 'Binder', and 'Code'.
- File Explorer:** Shows a directory structure under '/PySDM\_examples /Lowe\_et\_al\_2019/'. Files listed include: \_\_init\_\_.py, aerosol.py, fig\_1.ipynb, fig\_2.ipynb (selected), settings.py, and simulation.py. All files were modified 4 hours ago.
- Code Cell (Cell 4):**

```
figsize = (15, 5)
pylab.rcParams['font.size']=14)
fig, axes = pylab.subplots(1, 2, figsize=figsize, sharey=True)

for idx, var in enumerate(['S_max', 'n_c_cm3']):
    for key in output.keys():
        Y = np.asarray(output[key][var])
        axes[idx].plot(output[key][var], Y, label=key,
                       color=output[key]['color'],
                       linestyle='-' if key.endswith('-bulk') else '--')
    axes[idx].set_xlim(0, 100)

    axes[idx].set_ylabel('Displacement [m]')
    if var == 'S_max':
        axes[idx].set_xlabel('Supersaturation [%]')
        axes[idx].set_xlim(0, 0.5)
    elif var == 'n_c_cm3':
        axes[idx].set_xlabel('Cloud droplet concentration [cm$^{-3}$]')
        axes[idx].set_xlim(0, 600)
    else:
        assert False

for ax in axes:
    ax.grid()
axes[0].legend()
```
- Plots:** Two side-by-side line plots showing Displacement [m] on the y-axis (0 to 100) against an unlabeled x-axis (0.0 to 0.5).
  - Left Plot:** Shows curves for 'AerosolMarine-bulk' (solid blue), 'AerosolMarine-film' (dashed blue), 'AerosolBoreal-bulk' (solid green), 'AerosolBoreal-film' (dashed green), 'AerosolNascent-bulk' (solid orange), and 'AerosolNascent-film' (dashed orange). The curves generally decrease from higher displacements at lower supersaturation values.
  - Right Plot:** Shows curves for the same six models. The curves are much more vertical, indicating a sharp transition or peak in displacement at specific supersaturation values.
- Page Footer:** Includes navigation icons (left, right, search, etc.) and a page number '28/31'.

# PySDM: technological stack

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



# PySDM: technological stack

- ▶ Python [python.org](https://python.org)
- ▶ Numba (JIT, multi-threading) [numba.pydata.org](https://numba.pydata.org)
- ▶ ThrustRTC (GPU-resident backend)  
[pypi.org/project/ThrustRTC](https://pypi.org/project/ThrustRTC)
  
- ▶ GitHub & GitHub Actions [github.com](https://github.com)
- ▶ Codecov [codecov.io](https://codecov.io)
- ▶ AppVeyor [appveyor.com](https://appveyor.com)



# PySDM: technological stack

- ▶ Python [python.org](https://python.org)
- ▶ Numba (JIT, multi-threading) [numba.pydata.org](https://numba.pydata.org)
- ▶ ThrustRTC (GPU-resident backend)  
[pypi.org/project/ThrustRTC](https://pypi.org/project/ThrustRTC)
  
- ▶ GitHub & GitHub Actions [github.com](https://github.com)
- ▶ Codecov [codecov.io](https://codecov.io)
- ▶ AppVeyor [appveyor.com](https://appveyor.com)
  
- ▶ Jupyter [jupyter.org](https://jupyter.org)
- ▶ Binder [mybinder.org](https://mybinder.org)
- ▶ Colab [colab.research.google.com](https://colab.research.google.com)



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

 Atmospheric Cloud Simulation Group @ Jagiellonian University

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

Overview Repositories Projects Packages Teams People 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](#) Ołeksi 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 Olesiak's paper on an application of PyMPDATA in bin microphysics (published in GMD)

Our technological stack:

- Python NumPy LLVM ThrustRTC/CUDA Numpy pytest
- Colab GitHub Jupyter GitHub Actions Jupyter PyCharm

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

- PySDM: [PyPI package 2.4](#) [codecov 70%](#) [PySDM docs](#) [pdoc3](#)
- PySDM-examples: [PyPI package 2.9](#) [PySDM examples docs](#) [pdoc3](#)
- PyMPDATA: [PyPI package 1.63](#) [codecov 71%](#) [PyMPDATA docs](#) [pdoc3](#)
- PyMPDATA-examples: [PyPI package 1.61](#) [PyMPDATA examples docs](#) [pdoc3](#)
- numba-mpi: [PyPI package 0.3](#) [numba-mpi docs](#) [pdoc3](#)
- atmos-cloud-sim-utils: [PyPI package 0.3](#) [utils docs](#) [pdoc3](#)

Funding:

EU Funding by FNP Polish Funding by NCN US DOE Funding by ASR

Unfollow

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

People



Invite someone

Top languages

Python Jupyter Notebook

Most used topics

#pypi-package #python #atmospheric-modeling  
#numba #atmospheric-physics

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

The screenshot shows the GitHub profile of the Atmospheric Cloud Simulation Group @ Jagiellonian University. The profile includes a logo featuring a crown and a shield, the group's name, and links to Poland and their website. The main page displays news items, a technological stack (Python, NumPy, LLVM, ThrustRTC/CUDA, Numba, Pytest, C++, CUDA, OpenMP, FFTW, Github Actions, Jupyter, PyCharm), Python packages (PySDM, PySDM-examples, PyMPDATA, PyMPDATA-examples, numba-mpi, atmos-cloud-sim-utils), and funding information from EU, NCN, and US DOE.

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

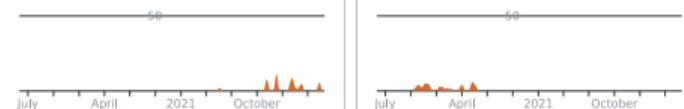
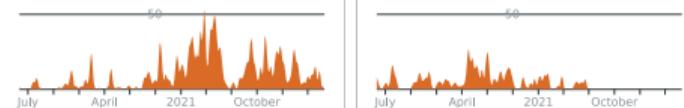
## co-authors, contributors, collaborators

- ▶ @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

Contributions: 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

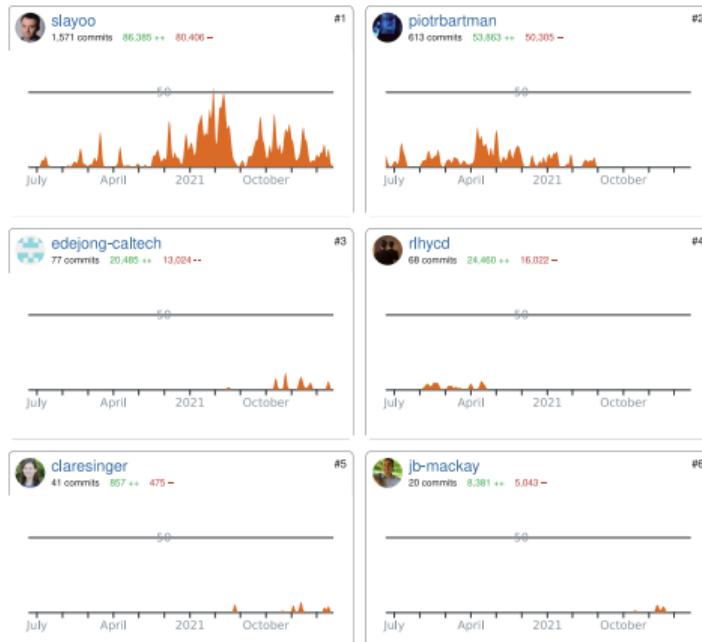
## funding

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

Jun 2, 2019 – Jun 2, 2022

Contributions: 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

