open-source Python projects for particle-based aerosol/cloud microphysics modelling

---

Sylwester Arabas
Jagiellonian University, Kraków, Poland

Feb 28 2023

DACoPT meeting @ INRIA Sofia Antipolis

Smoluchowski's coagulation equation (SCE)

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

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

Smoluchowski's coagulation equation (SCE)

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

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

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

Smoluchowski's coagulation equation (SCE)

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

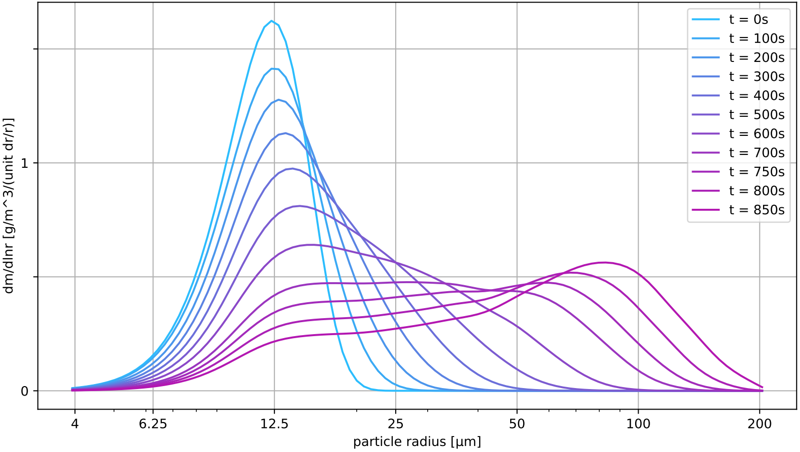
collision kernel: $a(x_1, x_2)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$

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

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

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

cloud droplet collisional growth



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

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

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

(photo: Yevgen Timashov / National Geographic)

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.:

- 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
- ...

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

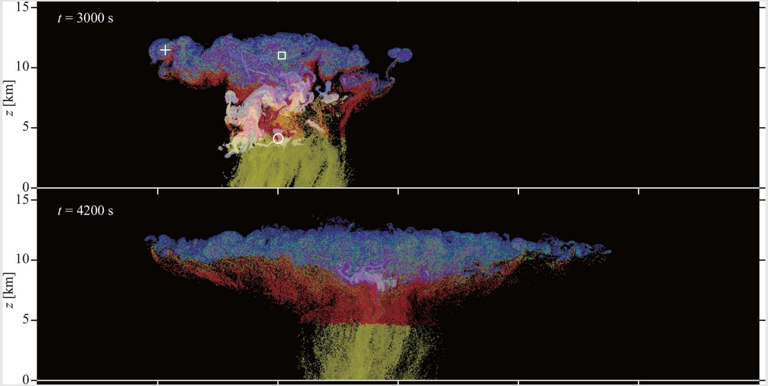


Figure 1. Typical realization of CTRL cloud spatial structures at $t = 2040$, 2460, 3000, 4200, and 5400 s. The mixing ratio of cloud water, rainwater, cloud ice, graupel, and snow aggregates are plotted in fading white, yellow, blue, red, and green, respectively. The symbols indicate examples of unrealistic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

SCE (naïve impl)	SDM
method type	
mean-field, deterministic	Monte-Carlo, stochastic

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

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

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

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")

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)

super-particles as an alternative to bulk or bin μ-phyics

Confronting the Challenge of Modeling Cloud and Precipitation Microphysics

Hugh Morrison, Marcus van Lier-Walqui, Ann M. Fridlind, Wojciech W. Grabowski, Jerry Y. Harrington, Corinna Hoose, Alexei Korolev, Matthew R. Kumjian, Jason A. Milbrandt, Hanna Pawlowska, Derek J. Posselt, Olivier P. Prat, Karly J. Reimel, Shin-Ichiro Shima, Bastiaan van Diedenhoven, Lulin Xue

super-particles as an alternative to bulk or bin μ-phyics

Confronting the Challenge of Modeling Cloud and Precipitation Microphysics

Hugh Morrison, Marcus van Lier-Walqui, Ann M. Fridlind, Wojciech W. Grabowski, Jerry Y. Harrington, Corinna Hoose, Alexei Korolev, Matthew R. Kumjian, Jason A. Milbrandt, Hanna Pawlowska, Derek J. Posselt, Olivier P. Prat, Karly J. Reimel, Shin-Ichiro Shima, Bastiaan van Diedenhoven, Lulin Xue

Journal of Advances in Modeling Earth Systems 10.1029/2019MS001689

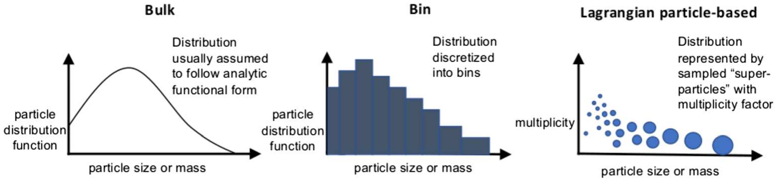


Figure 3. Representation of cloud and precipitation particle distributions in the three main types of microphysics

SDM

PySDM

PySDM goals:

PySDM goals:

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

PySDM goals:

- 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:

- 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:

- 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)

2D flow field

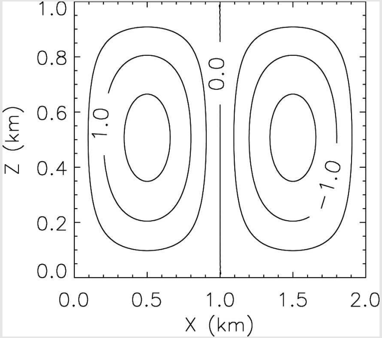
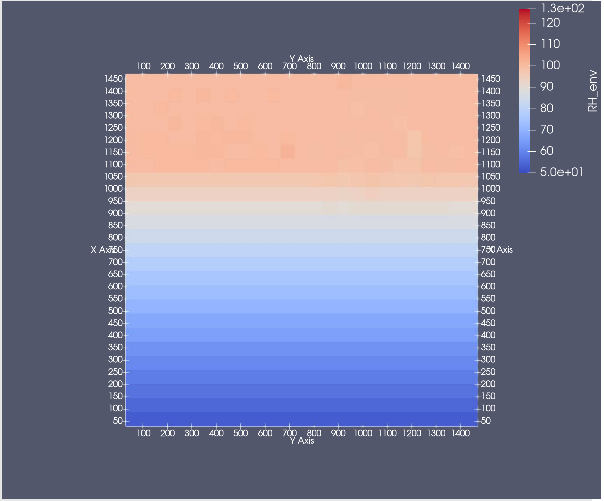
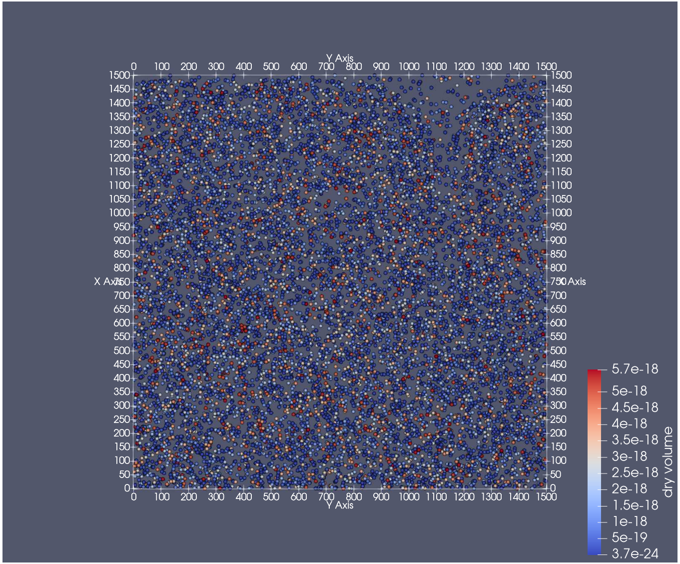


FIG. 1. Time-invariant vertical velocity for the stratocumulus case (contour interval is 0.5 m s^{-1}).

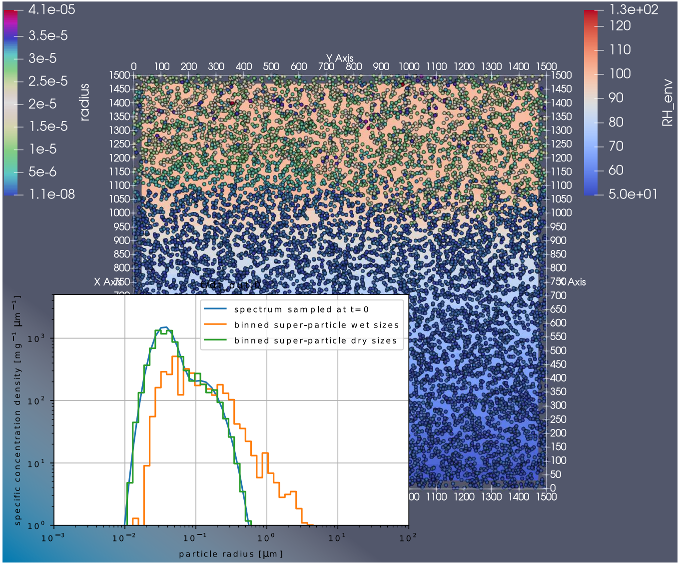
RH profile at t=0



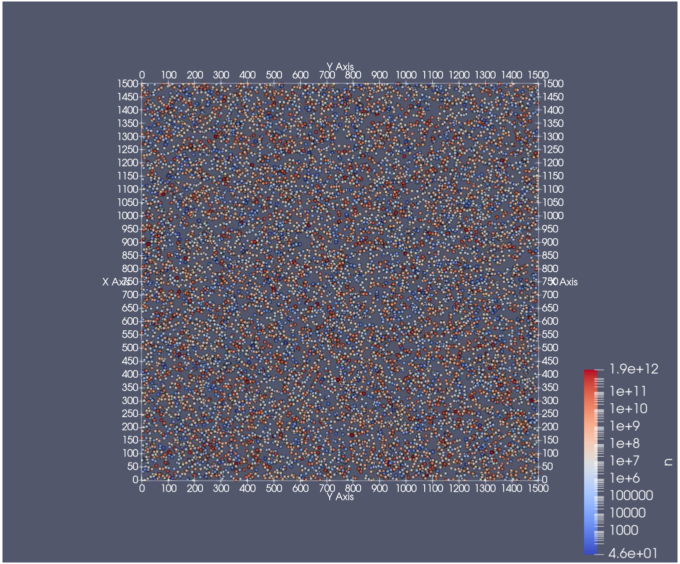
particle attribute initialisation: dry/wet volume



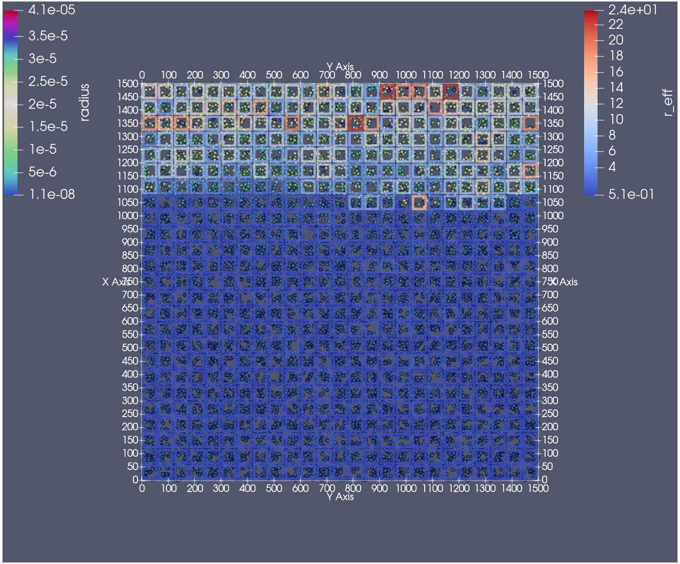
particle attribute initialisation: dry/wet volume



particle attribute initialisation: multiplicity

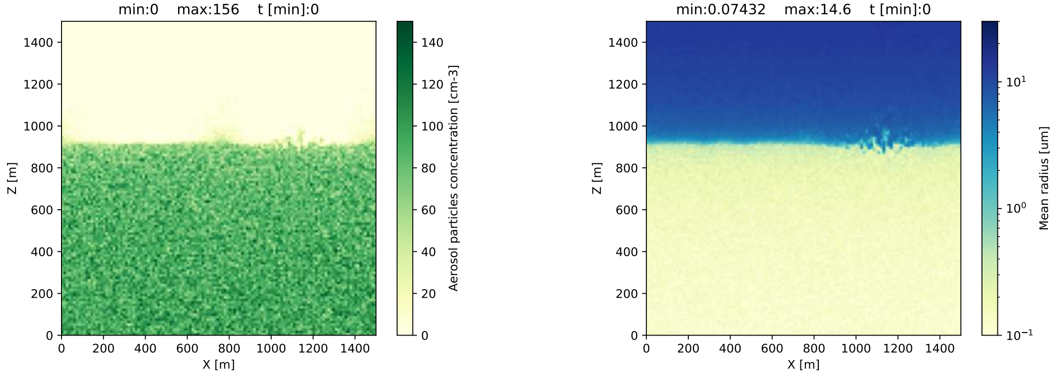


particle attribute evolution: droplet radius



Computational grid: 128x128

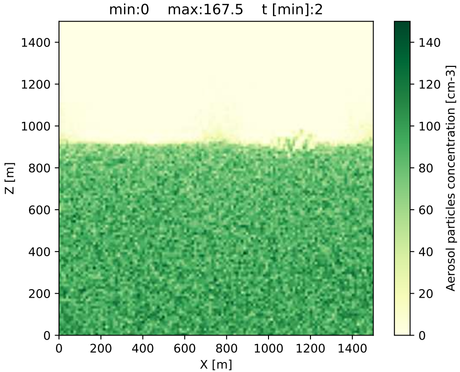
Computational particles: 2^{21}

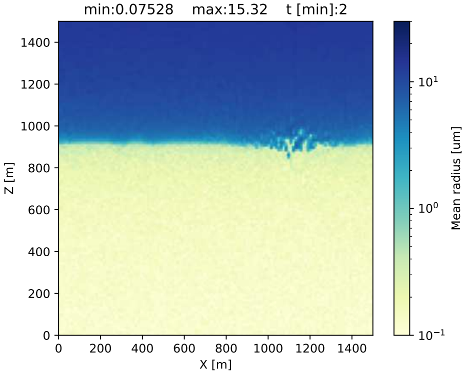


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}



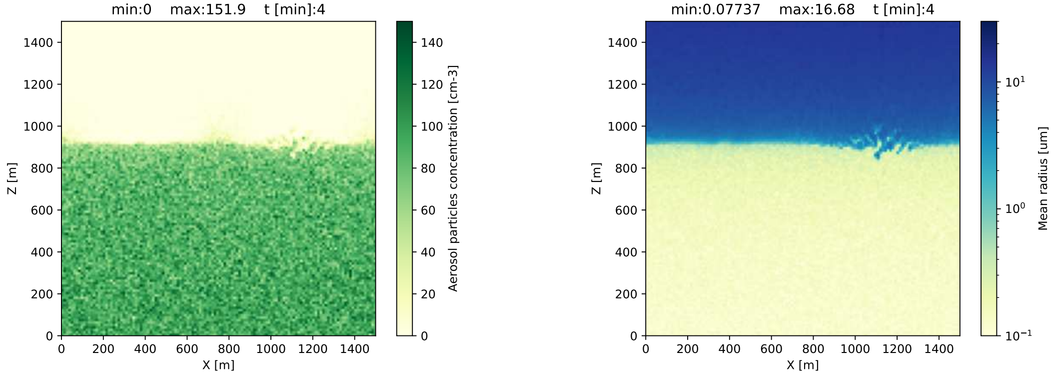


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

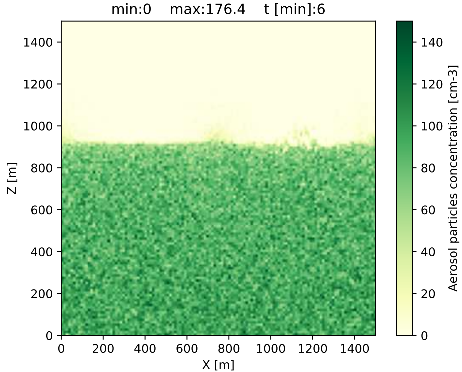
Computational particles: 2^{21}

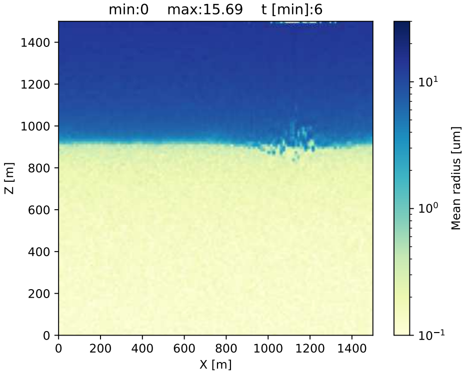


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}

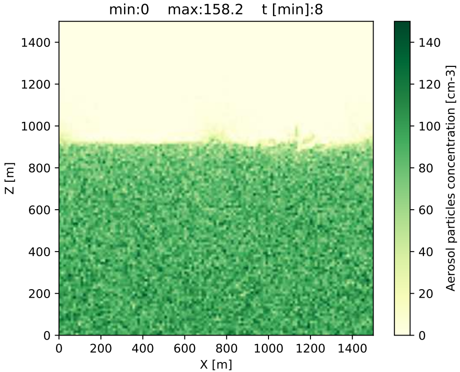


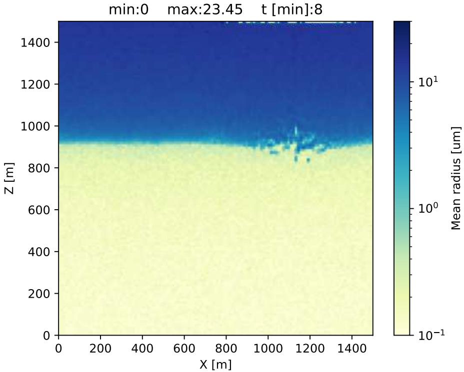


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}



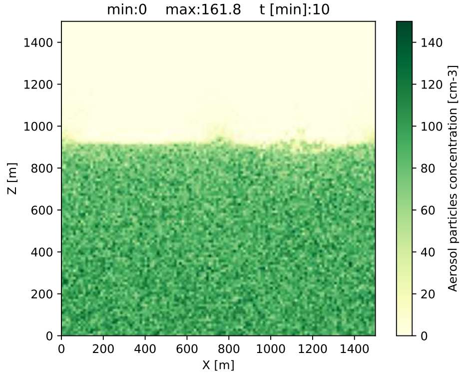


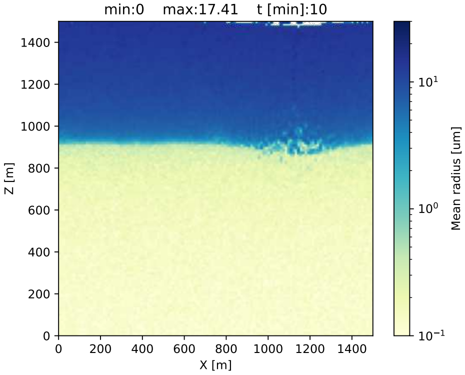
Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

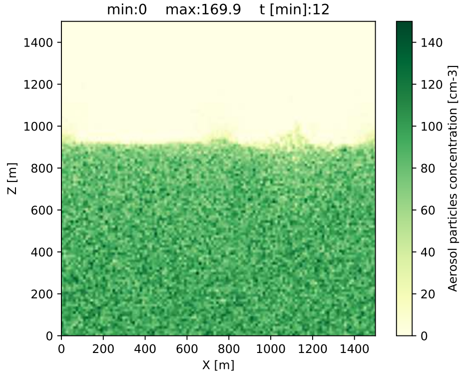


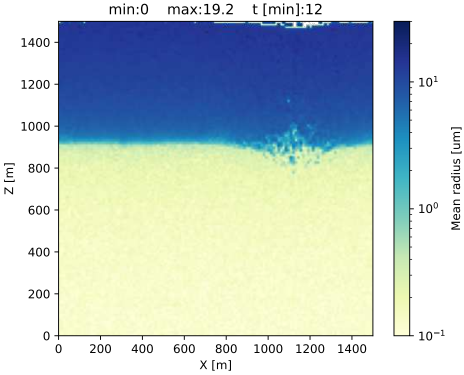


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}



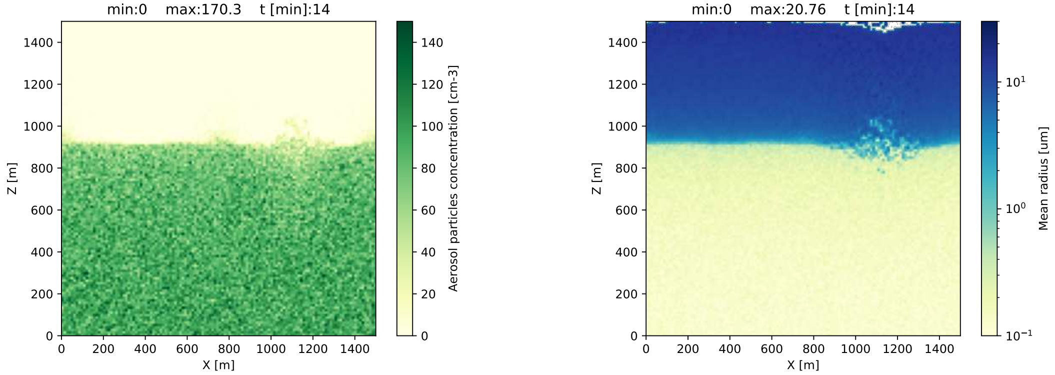


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

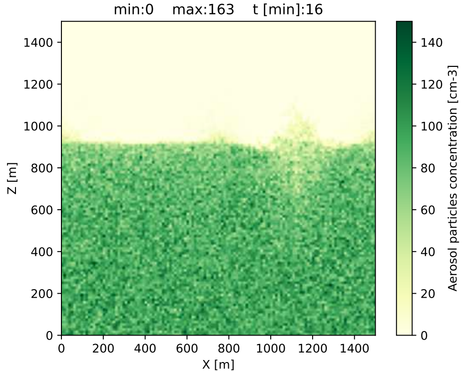
Computational grid: 128x128

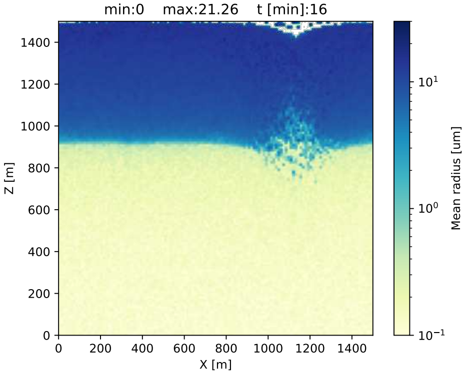
Computational particles: 2^{21}



Computational grid: 128x128

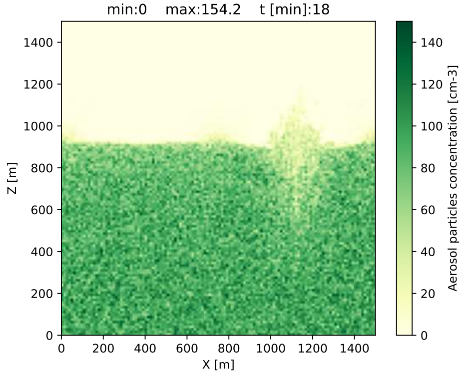
Computational particles: 2^{21}

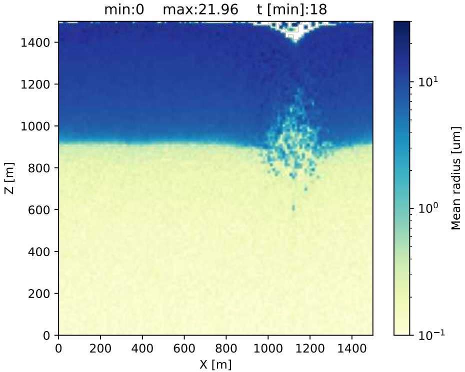




Computational grid: 128x128

Computational particles: 2^{21}

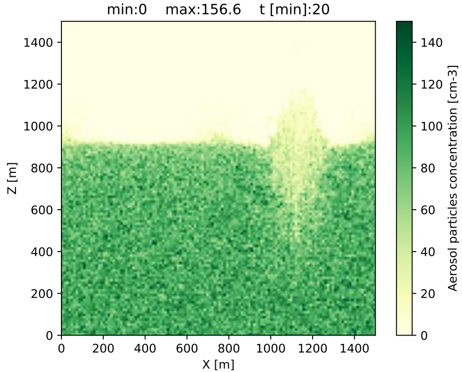


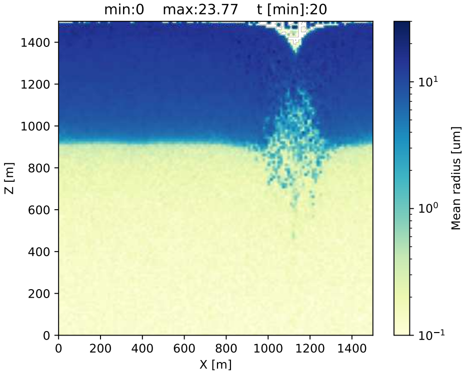


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}

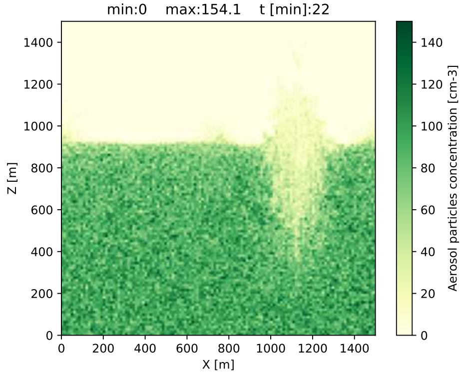


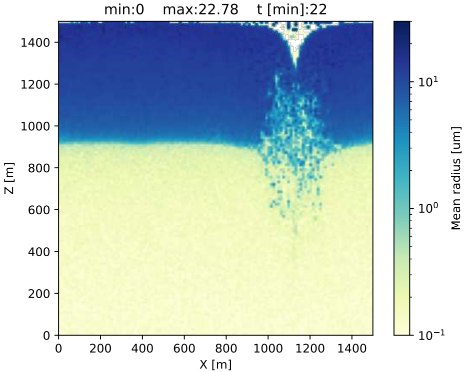


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}



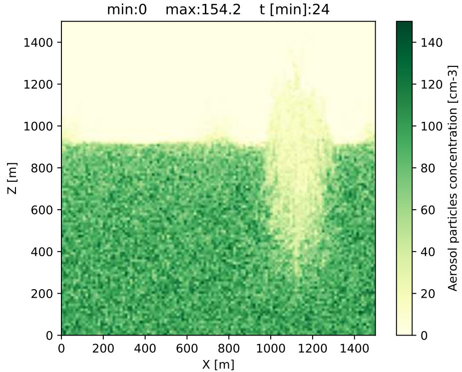


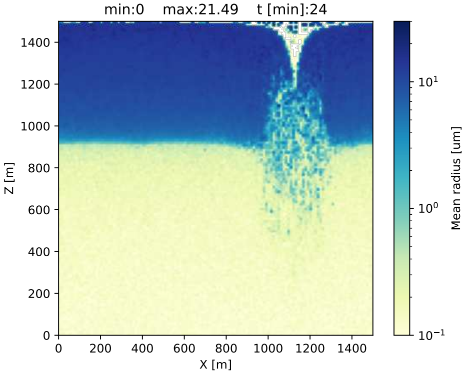
Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

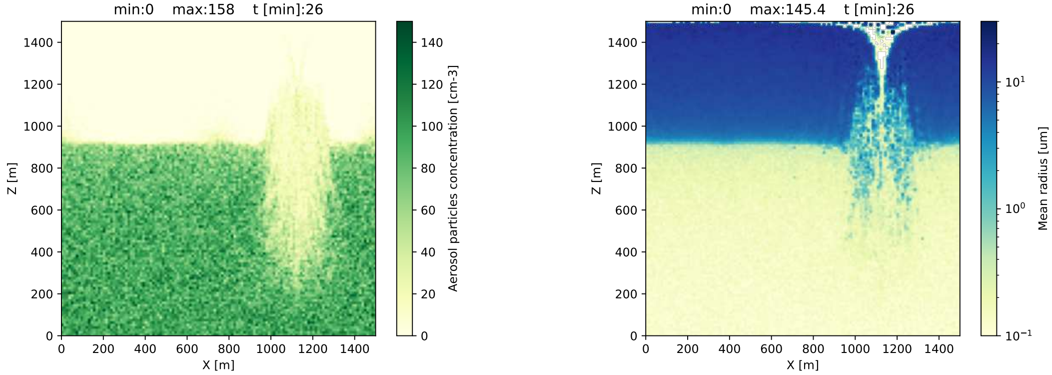




sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

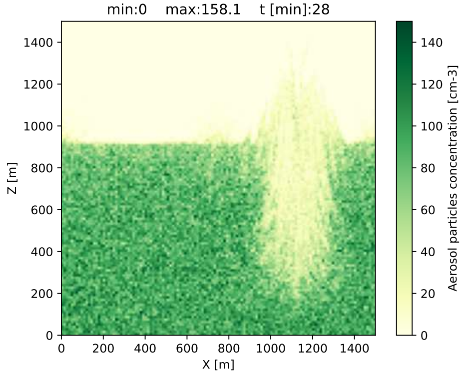


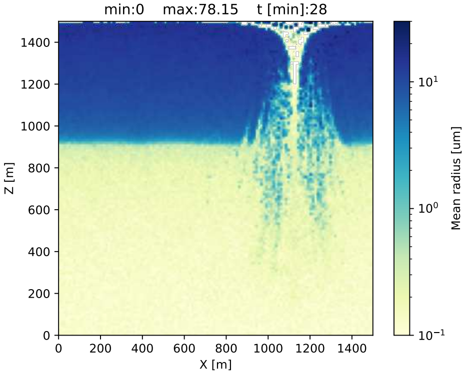
Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

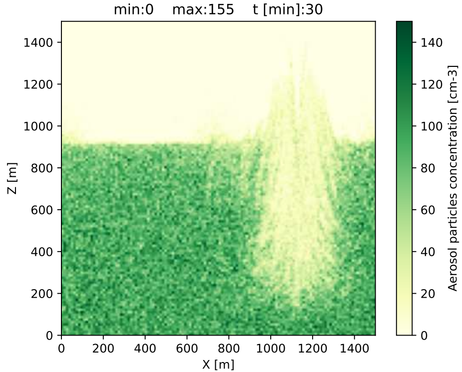


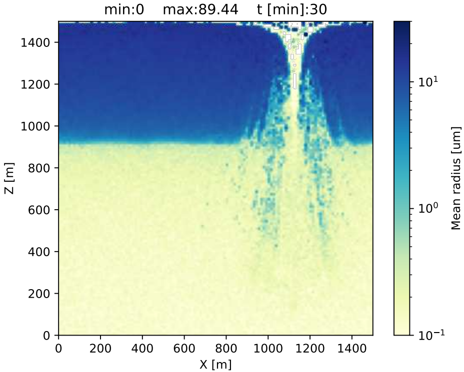


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}



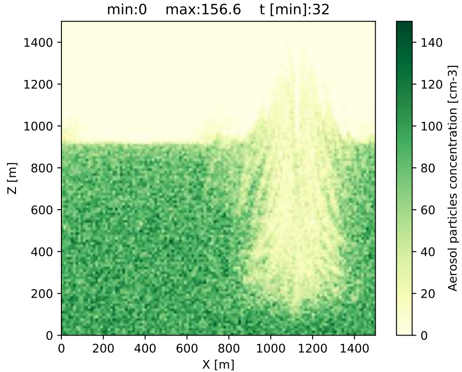


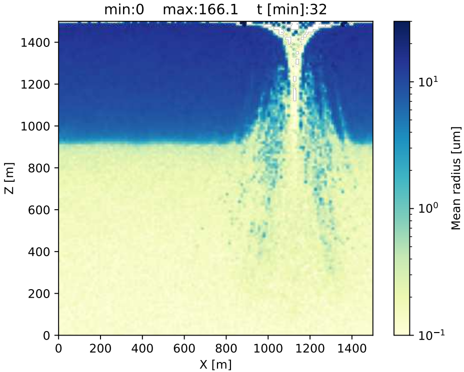
Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}



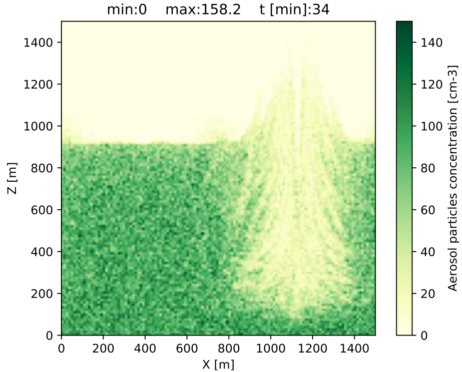


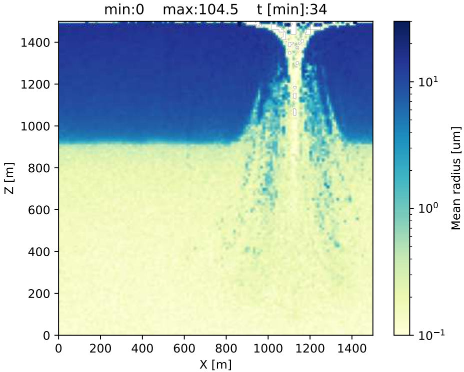
Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

Computational grid: 128x128

Computational particles: 2^{21}

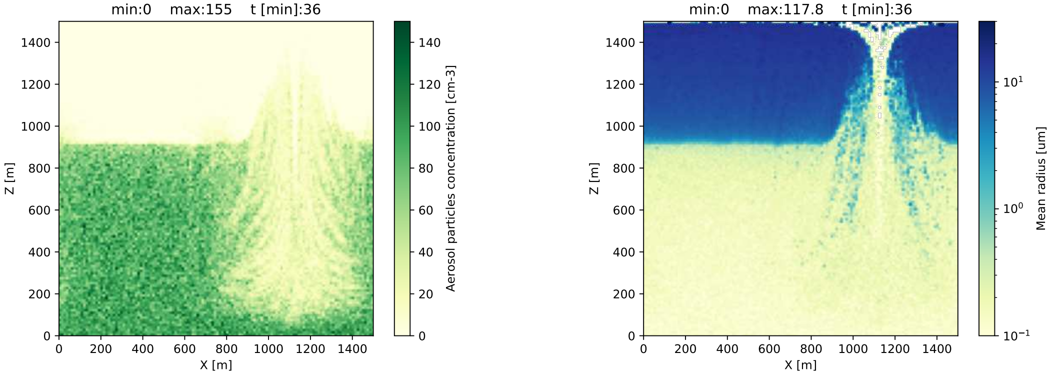




Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

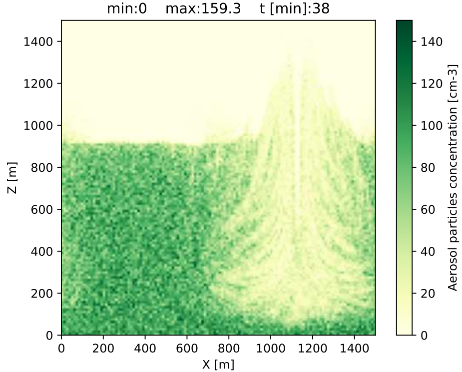
Computational particles: 2^{21}

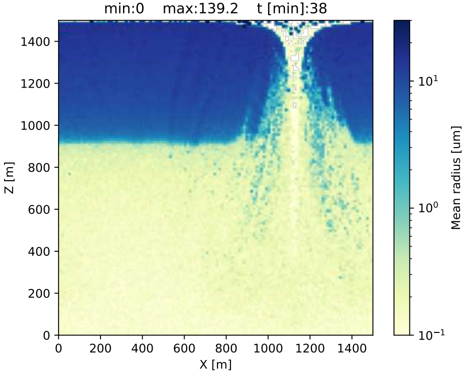


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

Computational grid: 128x128

Computational particles: 2^{21}



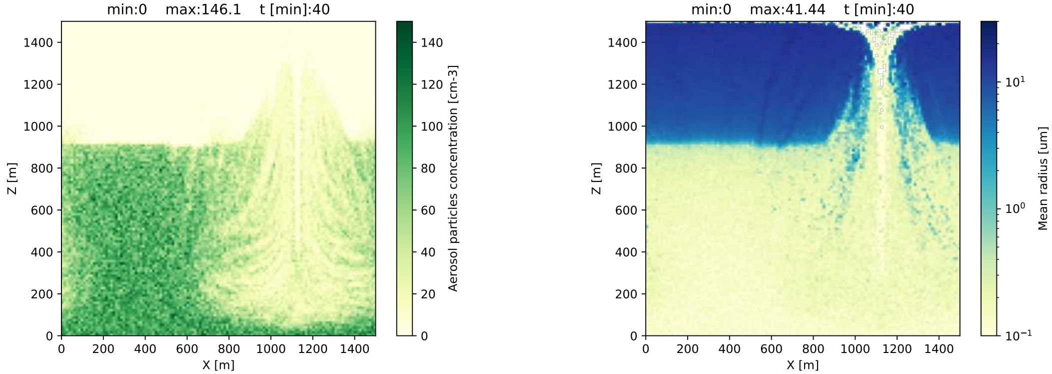


Simulation & visualisation: Piotr Bartman (MSc thesis @ WMiI UJ)

sample aerosol-cloud-precipitation interactions simulation

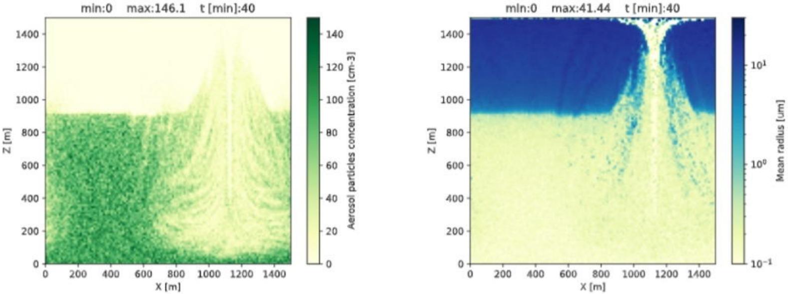
Computational grid: 128x128

Computational particles: 2^{21}



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

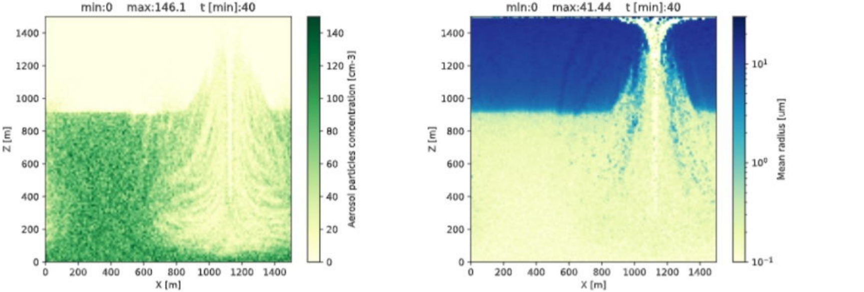
PySDM:



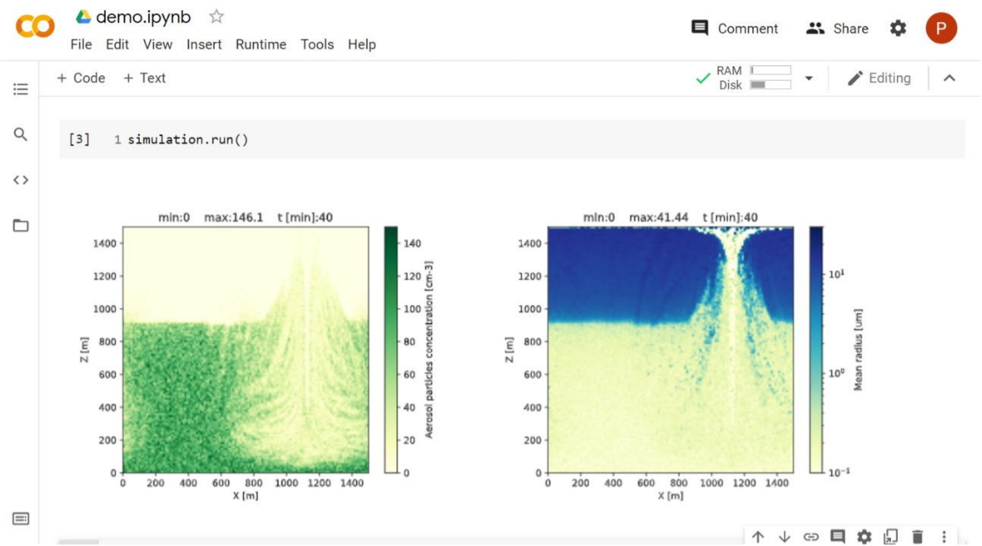
min:0 max:146.1 t [min]:40
Z [m]
X [m]
Aerosol particles concentration [cm-3]
min:0 max:41.44 t [min]:40
Z [m]
X [m]
Mean radius [um]

PySDM: Pythonic

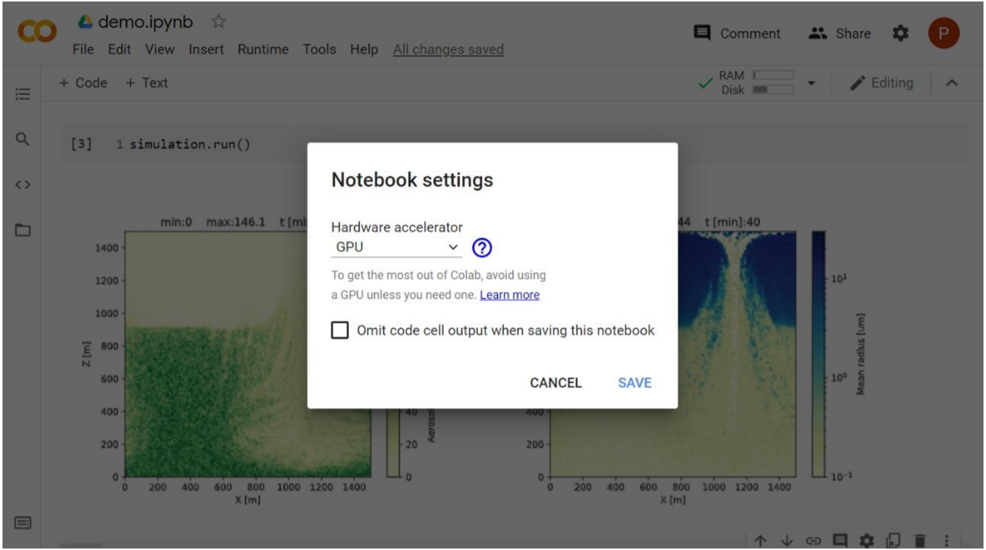
```
[3]  1 simulation.run()
```



PySDM: Pythonic, Jupyter-friendly



PySDM: Pythonic, Jupyter-friendly, GPU-enabled



PySDM: packages

Search projects

Help Sponsors Log in Register

PySDM 2.15

`pip install PySDM`

Latest version

Released: Dec 30, 2022

Pythonic particle-based (super-droplet) warm-rain/aqueous-chemistry cloud microphysics package with box, parcel & 1D/2D prescribed-flow examples in Python, Julia and Matlab

Navigation

- Project description
- Release history
- Download files

Project description

PySDM

Python 3 | LLVM Numba | CUDA ThrustRTC | Linux ✓ | macOS ✓ | Windows ✓

Jupyter ✓ | Maintained? yes | OpenHub | JOSS 10.21105/joss.03219 | DOI 10.5281/zenodo.7640495

EU Funding by FNP | PL Funding by NCN | US DOE Funding by ASR

License GPL v3 | Copyright Jagiellonian University

PySDM: users

PySDM v1: particle-based cloud modelling package for warm-rain microphysics and aqueous chemistry

☐ Search within citing articles

An Efficient Bayesian Approach to Learning Droplet Collision Kernels: Proof of Concept Using "Cloudy," a New *n*-Moment Bulk Microphysics Scheme

M Bieli, ORA Dunbar, EK De Jong… - Journal of Advances …, 2022 - Wiley Online Library

The small-scale microphysical processes governing the formation of precipitation particles cannot be resolved explicitly by cloud resolving and climate models. Instead, they are …

☆ Save 99 Cite Cited by 4 Related articles All 12 versions

Spanning the gap from bulk to bin: A novel spectral microphysics method

EK De Jong, T Bischoff, A Nadim… - Journal of Advances in …, 2022 - Wiley Online Library

Microphysics methods for climate models and numerical weather prediction typically track one, two, or three moments of a droplet size distribution for various categories of liquid, ice …

☆ Save 99 Cite Related articles All 7 versions

Breakups are Complicated: An Efficient Representation of Collisional Breakup in the Superdroplet Method

E de Jong, JB Mackay, A Jaruga, S Arabas - EGUsphere, 2022 - egusphere.copernicus.org

A key constraint of particle-based methods for modeling cloud microphysics is the conservation of total particle number, which is required for computational tractability. The …

☆ Save 99 Cite Related articles »

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

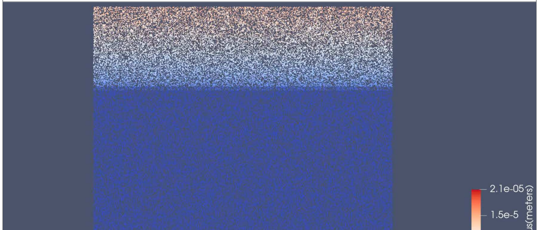
This PR includes a coupling logic for ClimateMachine.jl and PySDM.

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:

output.mp4



Reviewers

- slayoo
- charleskawczynski
- claresinger
- jakebolewski
- edejong-caltech
- tapios

Assignees

trontrytel

Labels

Microphysics

Projects

None yet

Milestone

No milestone

Development

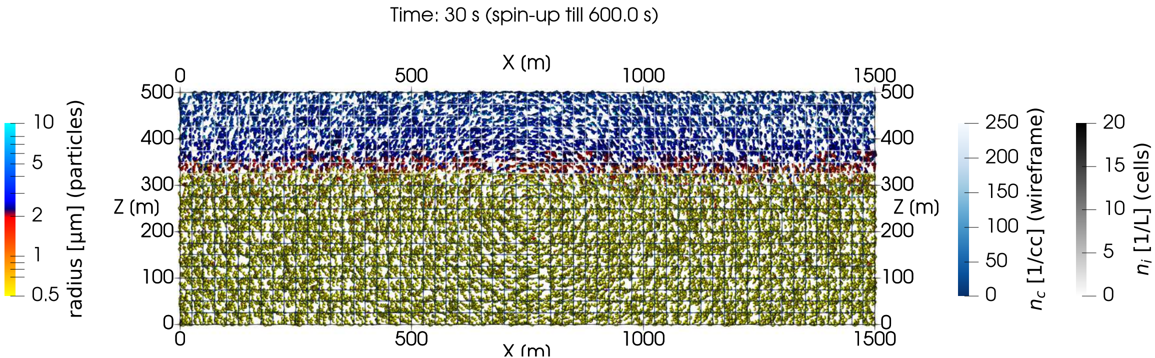
Successfully merging this pull request may close these issues.

None yet

teaser: Monte-Carlo immersion freezing in PySDM (singular or time-dependent)

https://www.reuters.com/markets/commodities/making-snow-stick-wind-challenges-winter-games-slope-makers-2021-11-29/

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

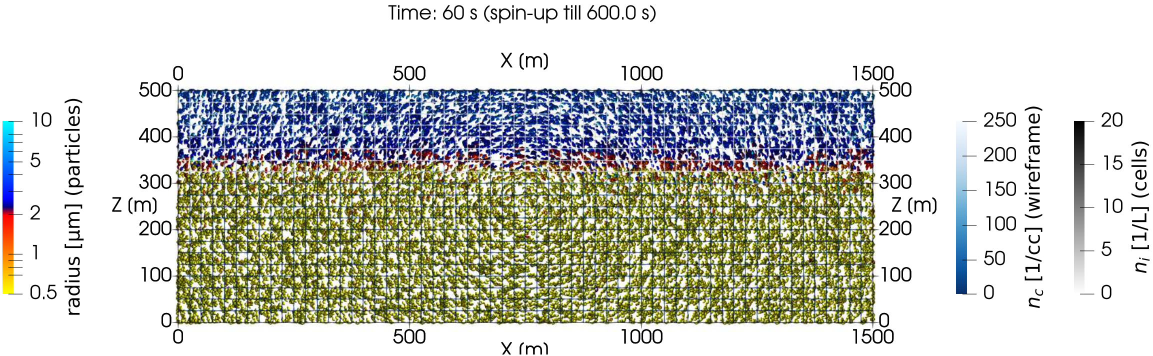


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

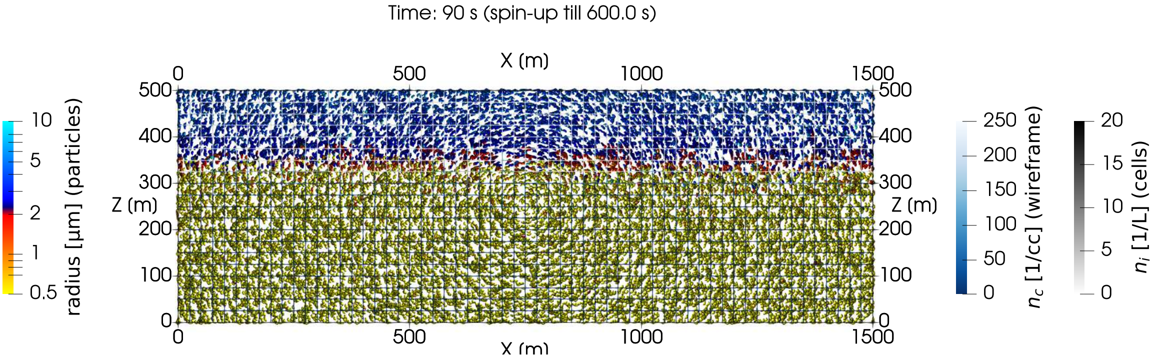


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

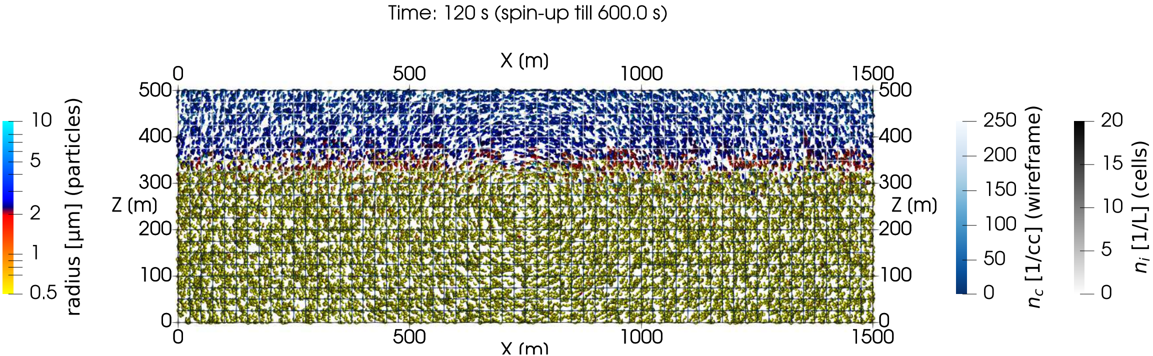


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

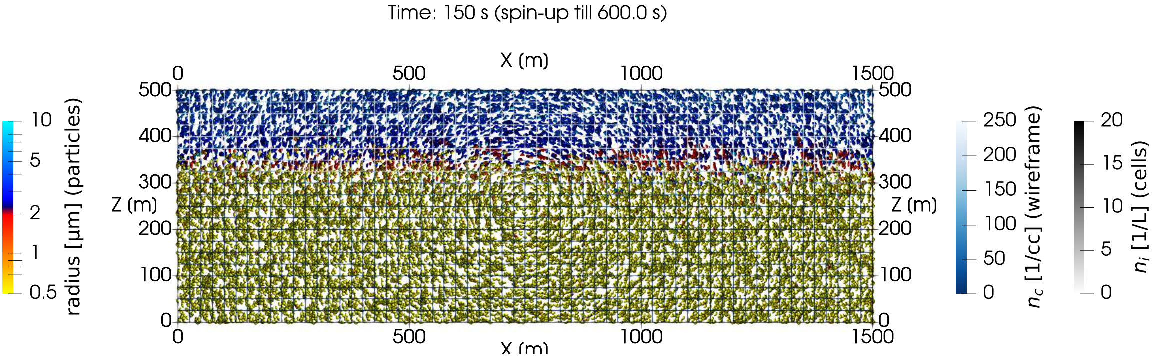


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

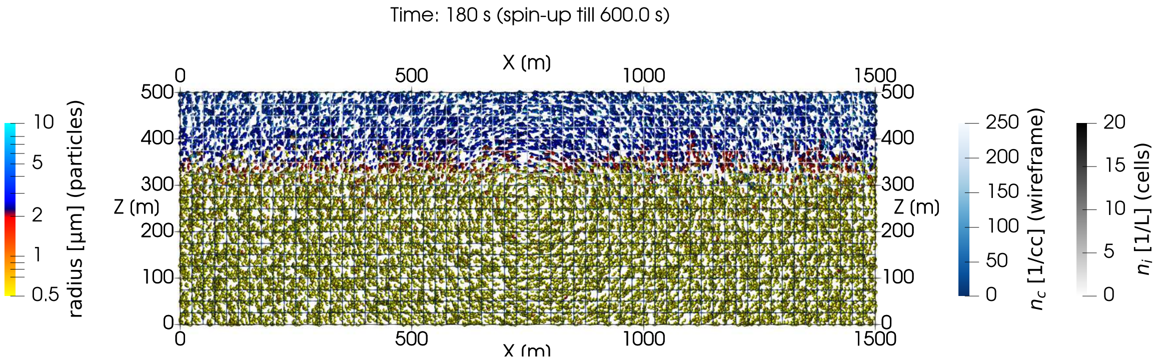


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ µm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

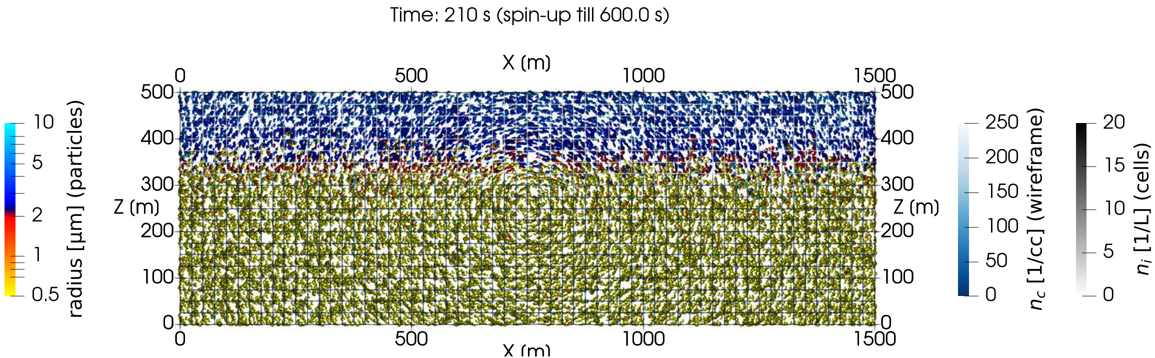


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

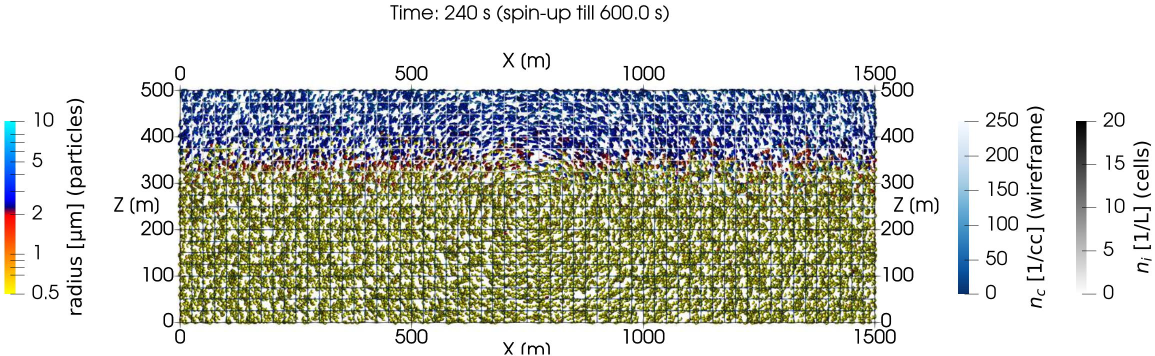


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

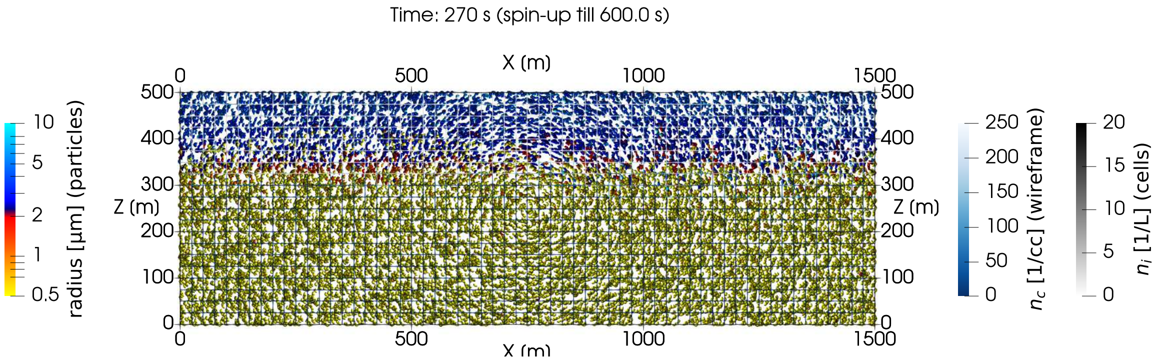


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

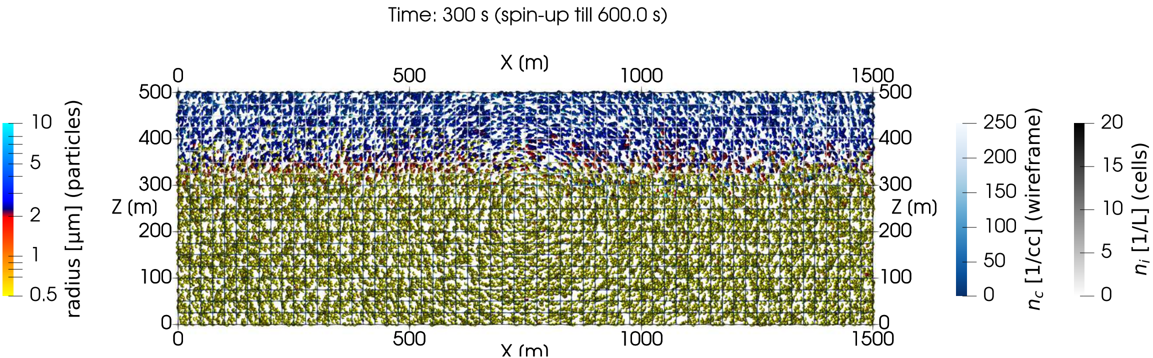


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

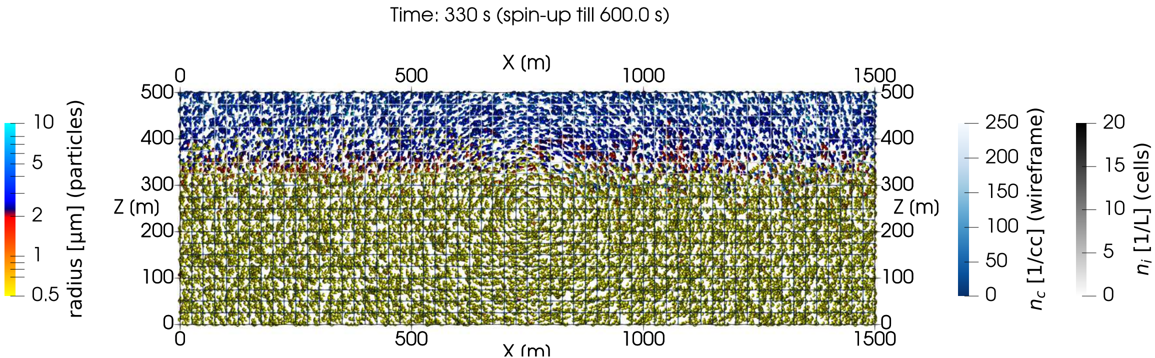


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

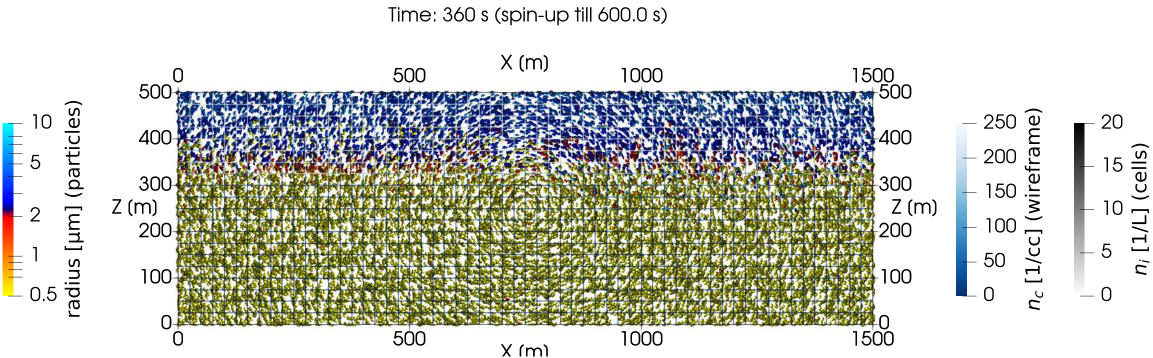


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

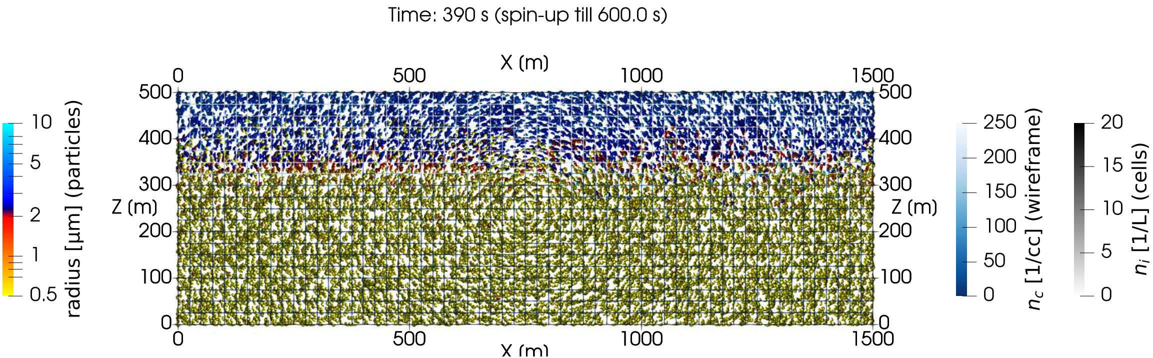


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

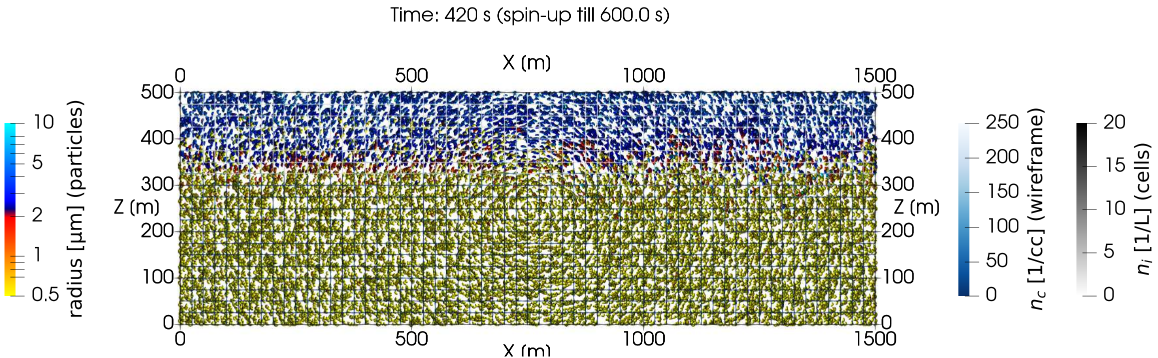


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

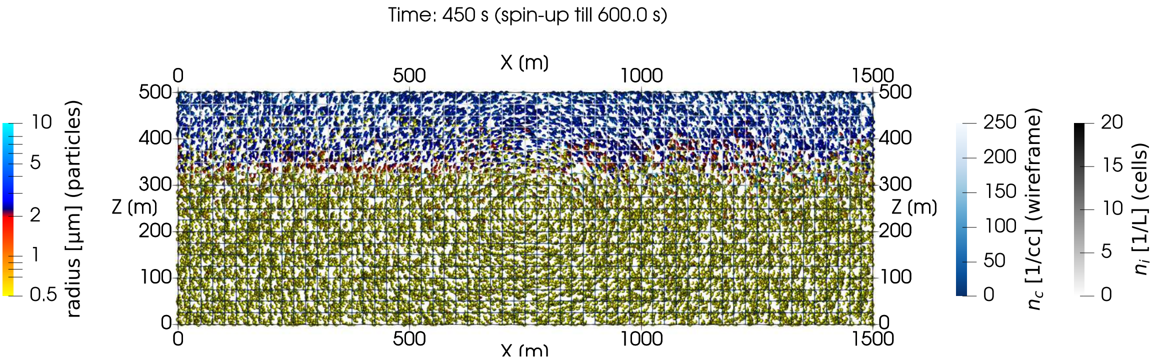


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

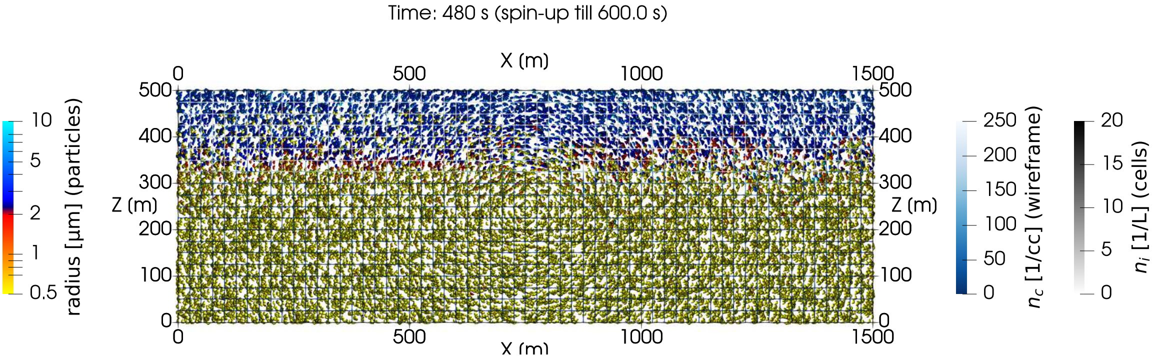


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

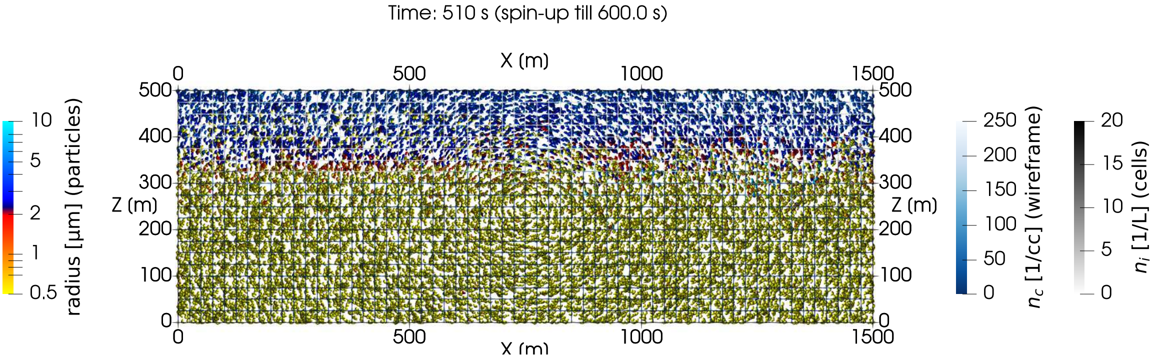


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

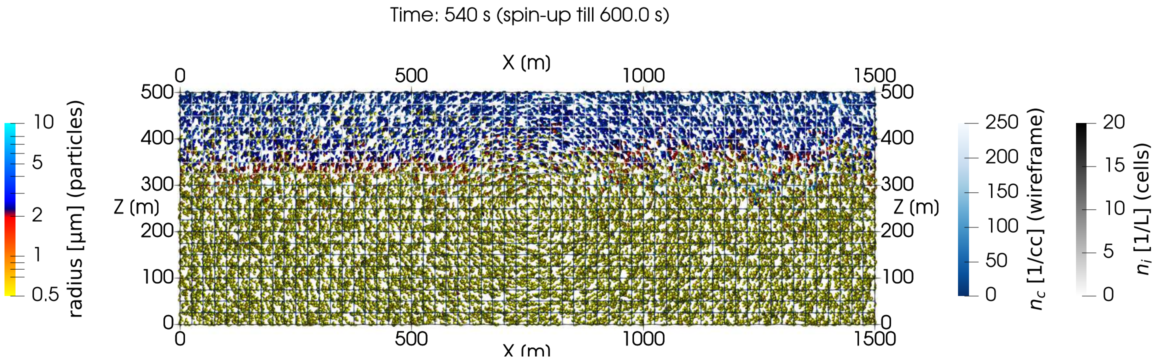


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

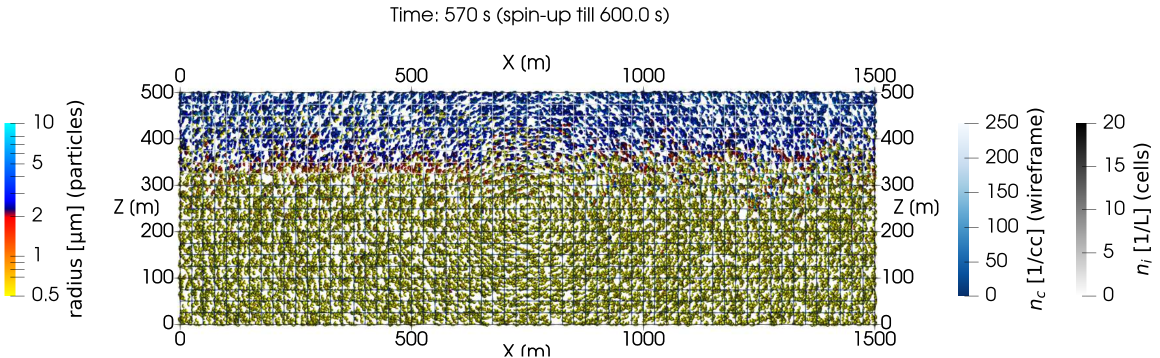


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ µm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

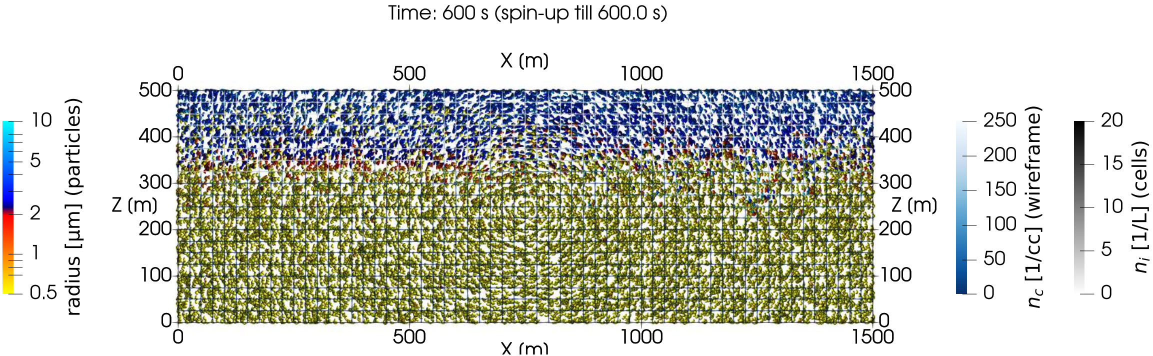


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

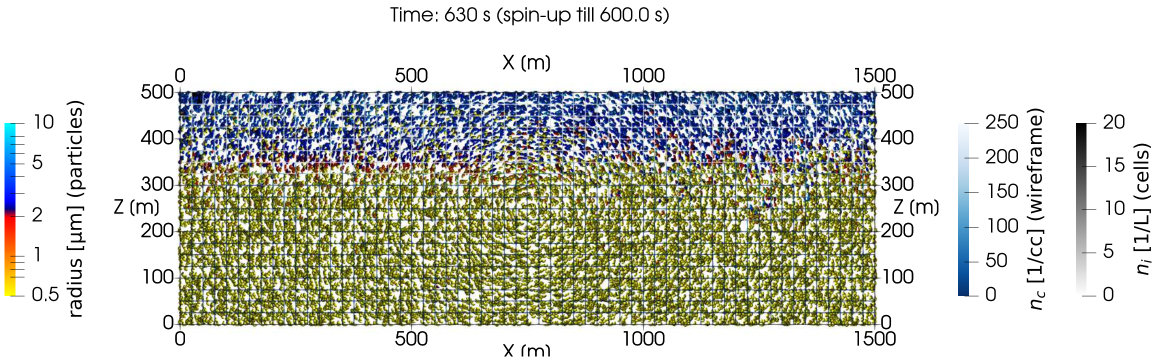


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

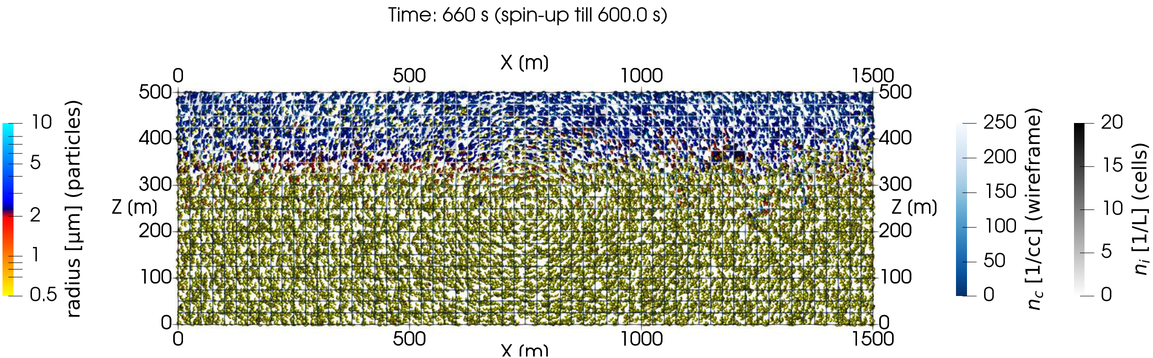


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

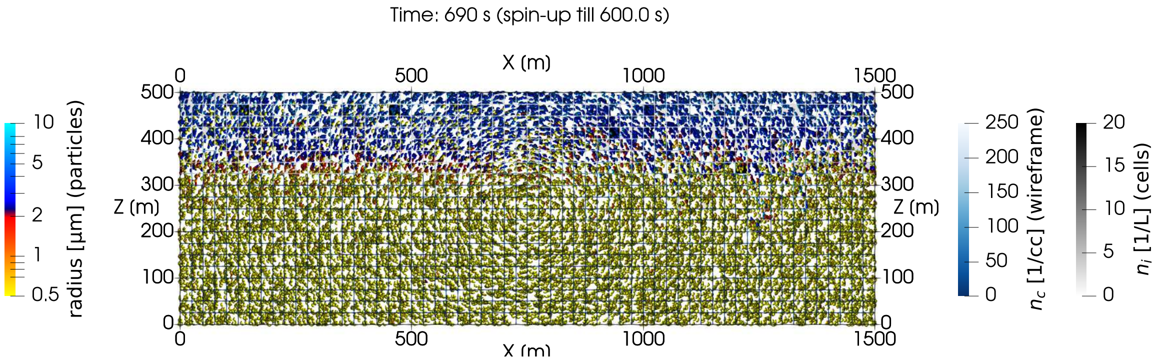


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

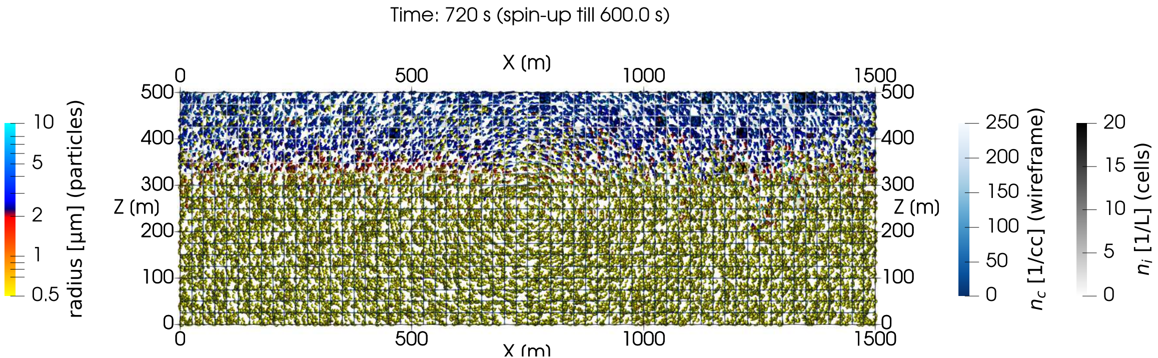


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

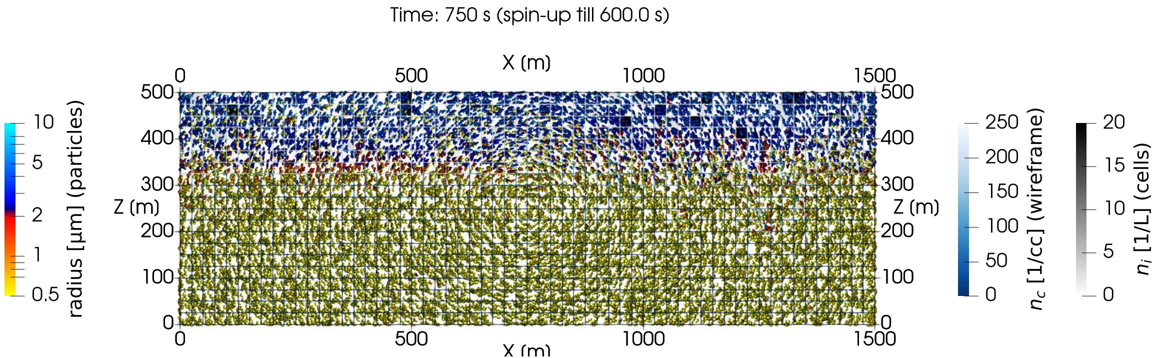


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

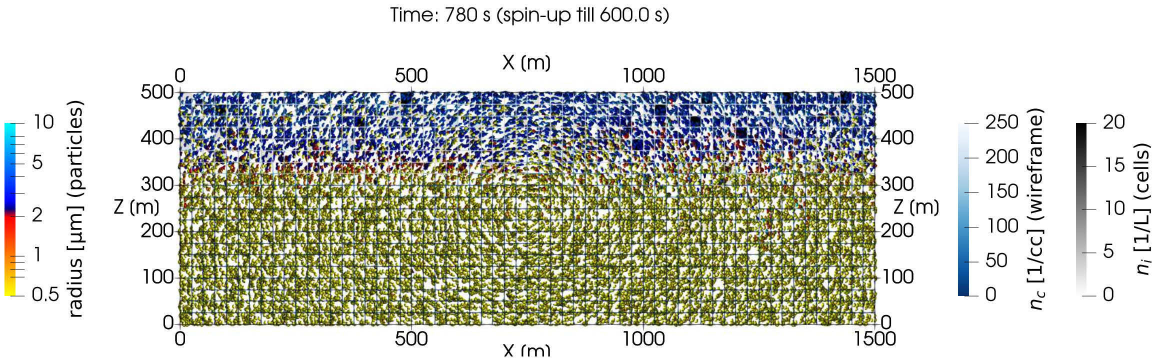


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

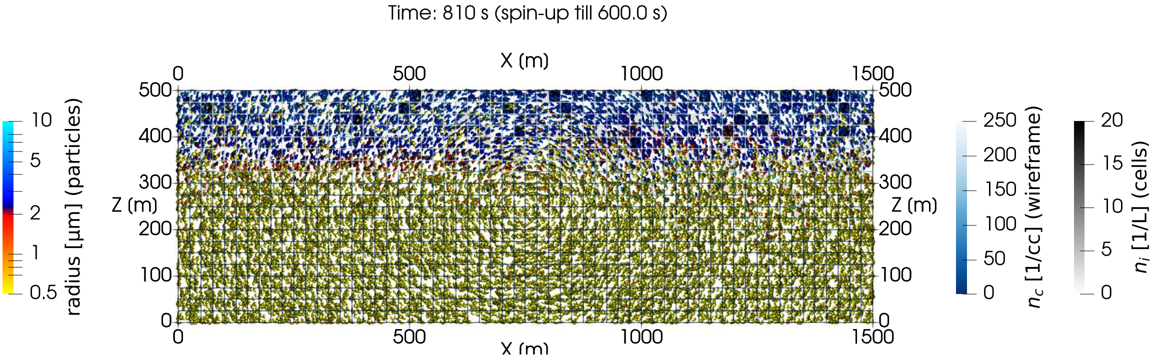


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

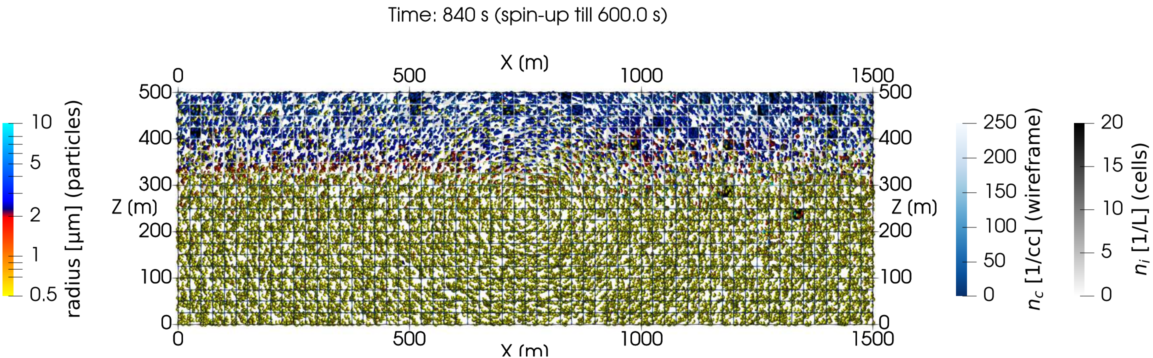


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

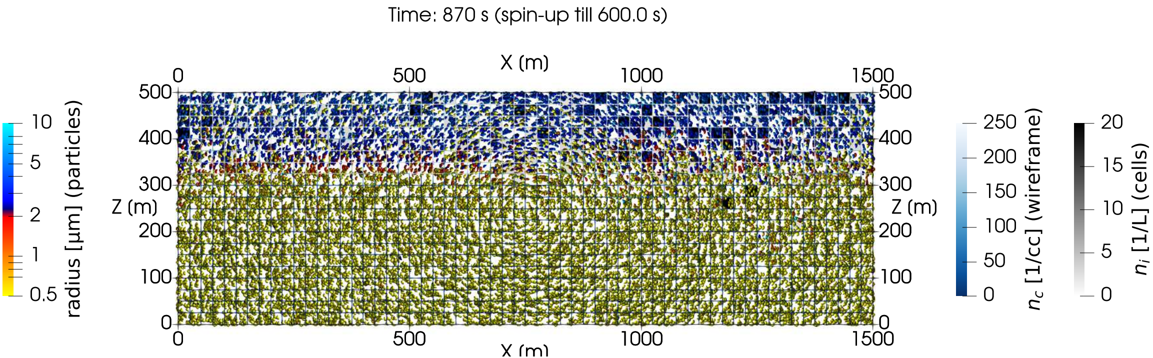


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

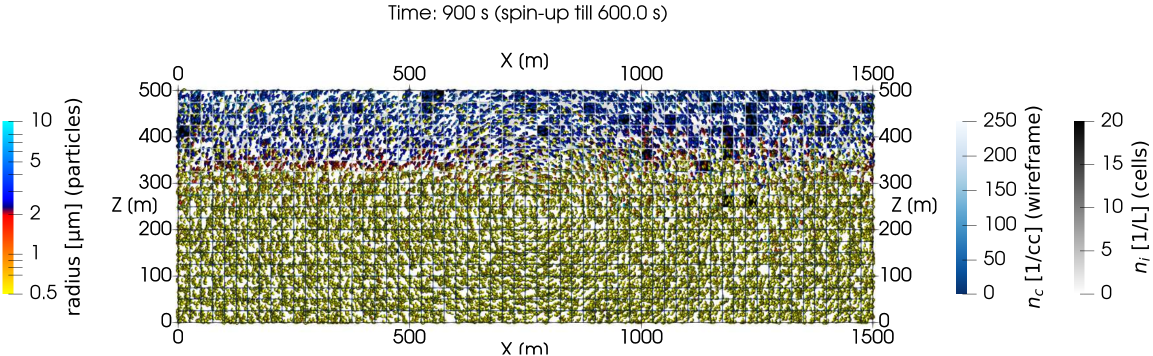


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

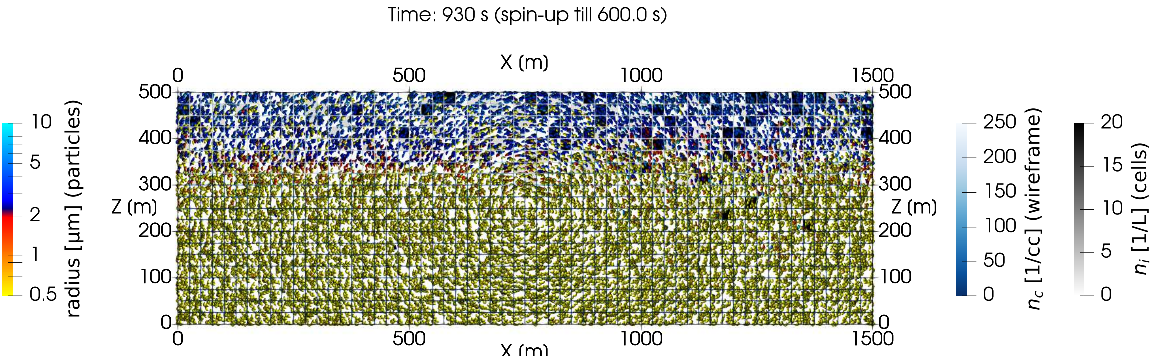


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

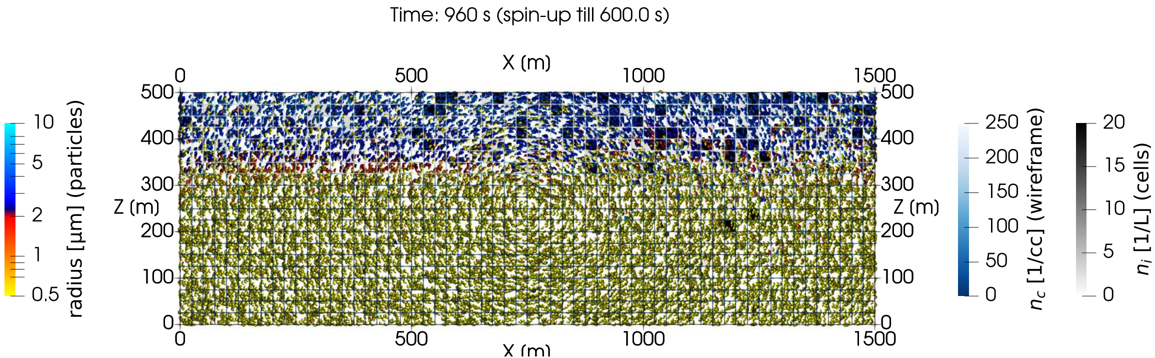


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

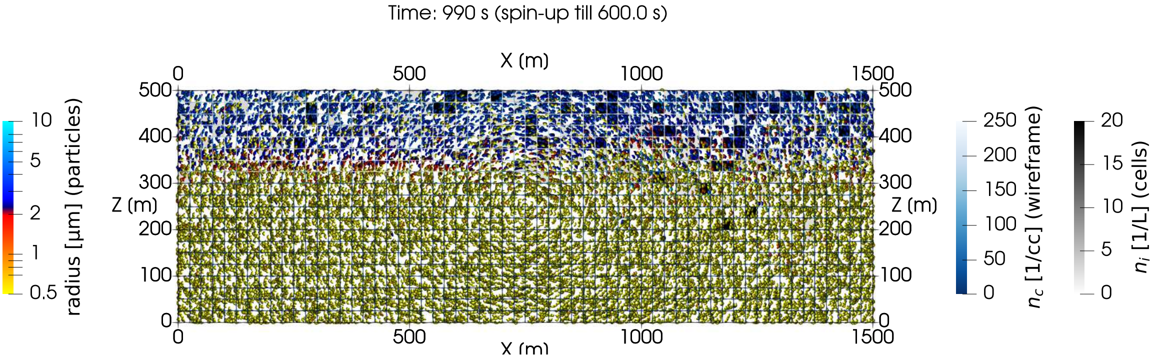


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $\quad N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

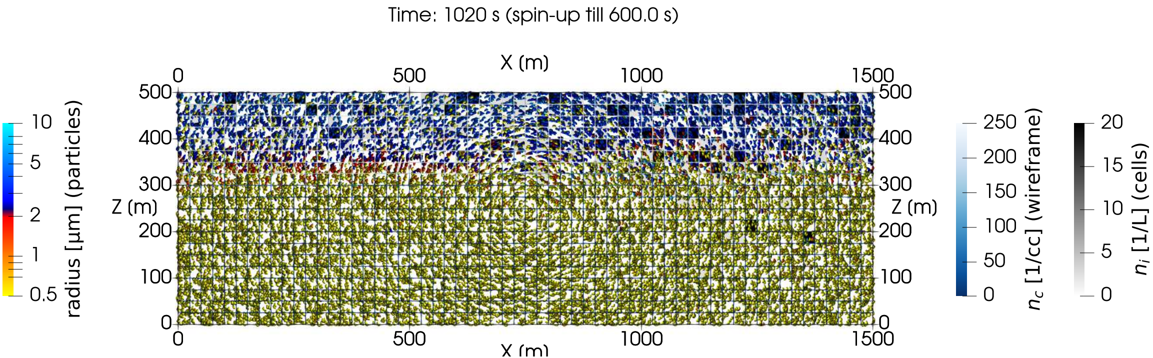


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

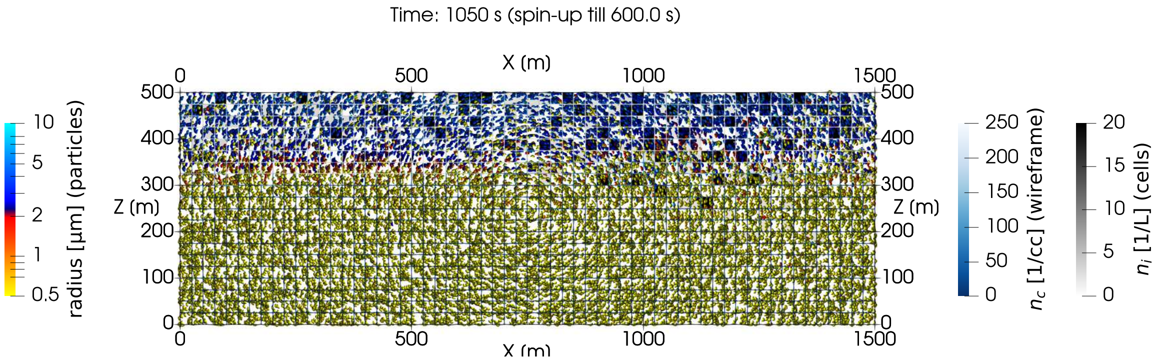


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ µm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

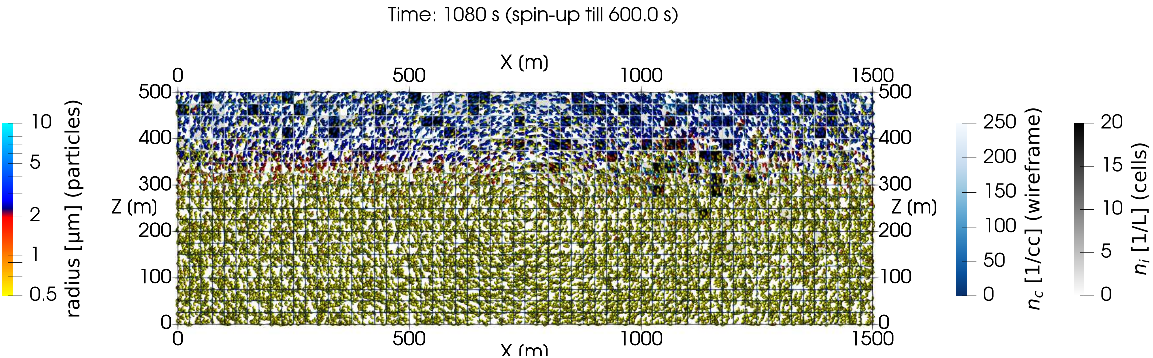


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

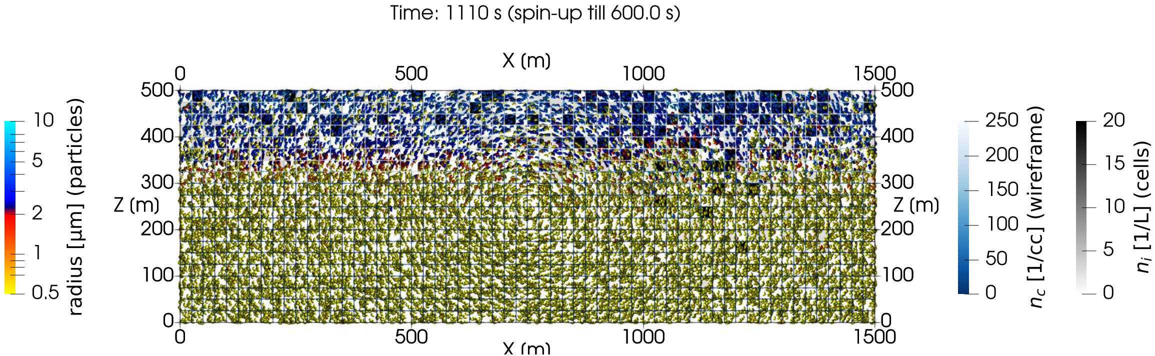


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

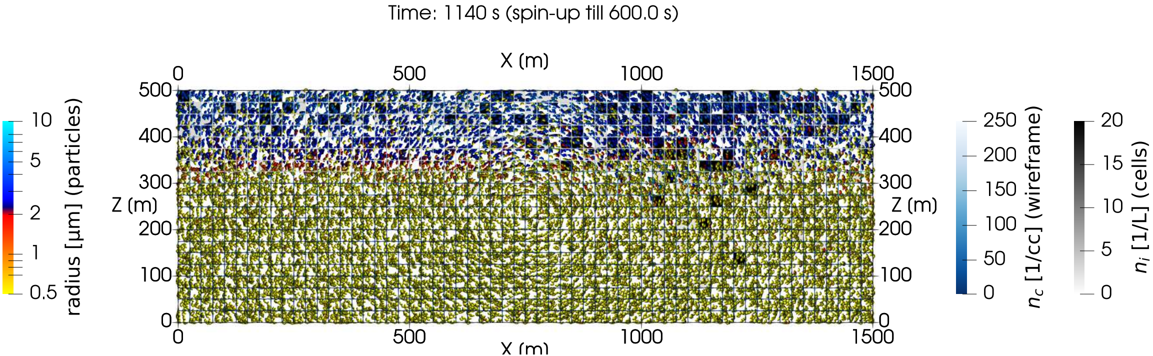


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

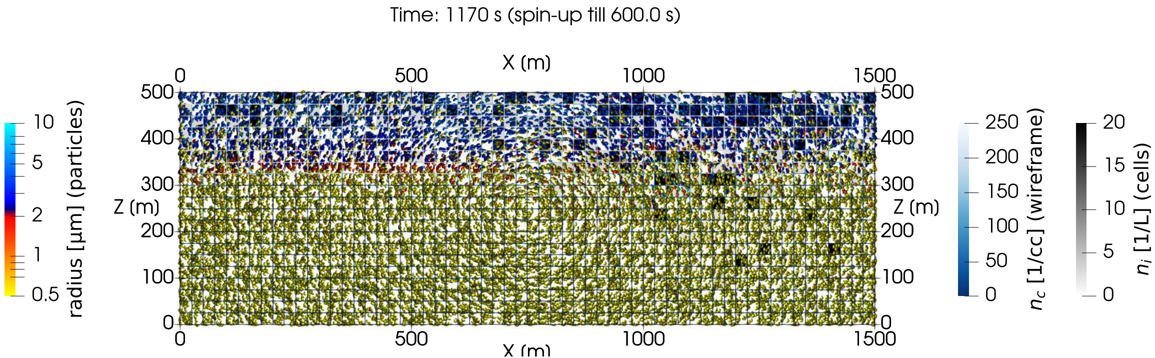


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

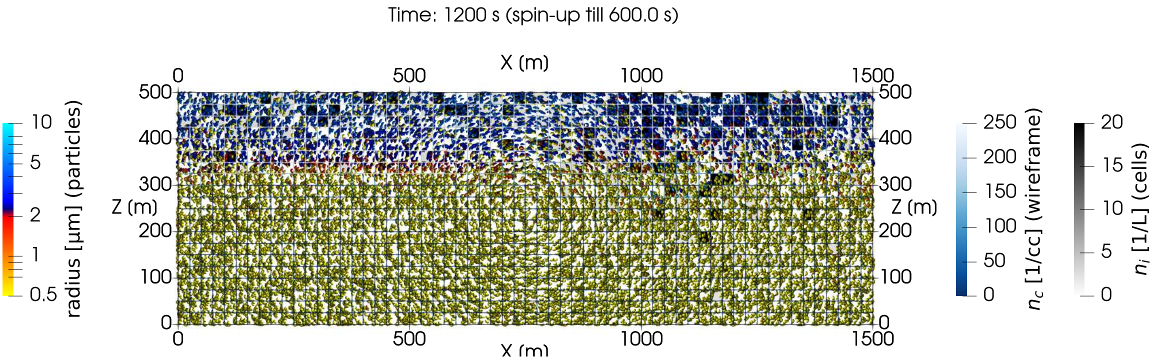


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_g = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

PySDM: particle-based μ-physics + prescribed-flow test (PyMPDATA)

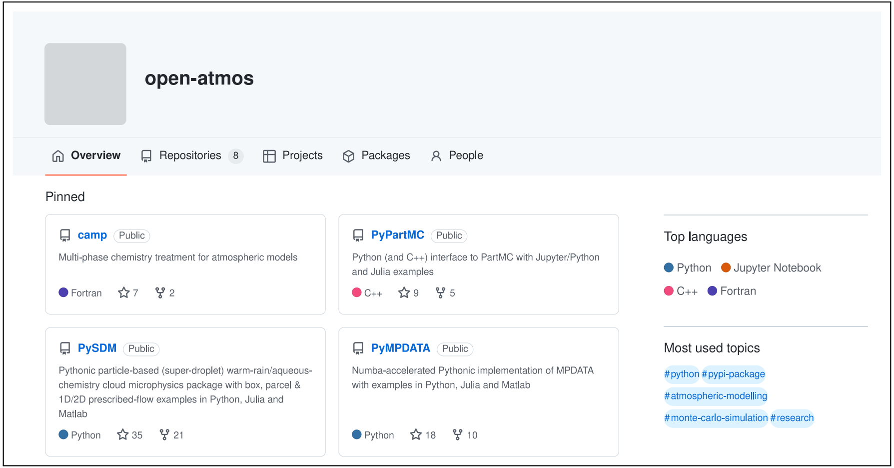


16+16 super-particles/cell for INP-rich + INP-free particles

$N_{\text{aer}} = 300/cc$ (two-mode lognormal) $N_{\text{INP}} = 150/L$ (lognormal, $D_g = 0.74$ μm, $\sigma_{\text{g}} = 2.55$)

spin-up = freezing off; subsequently frozen particles act as tracers

https://github.com/open-atmos



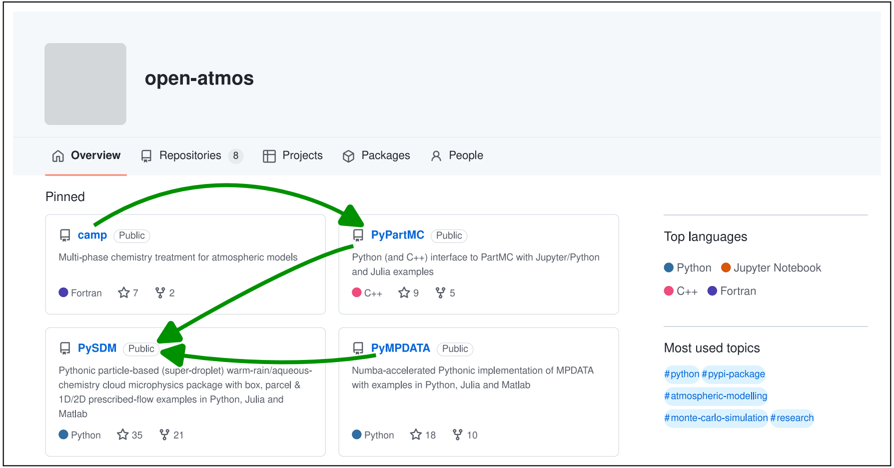
CAMP: BSC, NCAR, UIUC, UCI

PyPartMC: UIUC

PySDM: Jagiellonian, Caltech, UIUC

PyMPDATA: Jagiellonian

https://github.com/open-atmos



CAMP: BSC, NCAR, UIUC, UCI

PyPartMC: UIUC

PySDM: Jagiellonian, Caltech, UIUC

PyMPDATA: Jagiellonian

compdyn/partmc: Particle-resolved stochastic atmospheric aerosol model

PartMC: Particle-resolved Monte Carlo code for atmospheric aerosol simulation

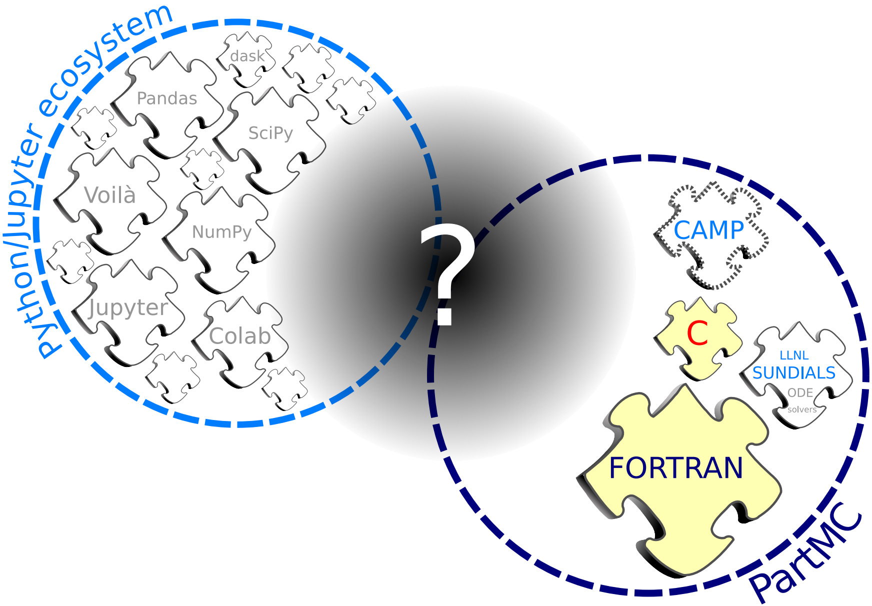
version v2.6.1 docker build automated CI passing license GPL-2.0 DOI 10.5281/zenodo.6144610

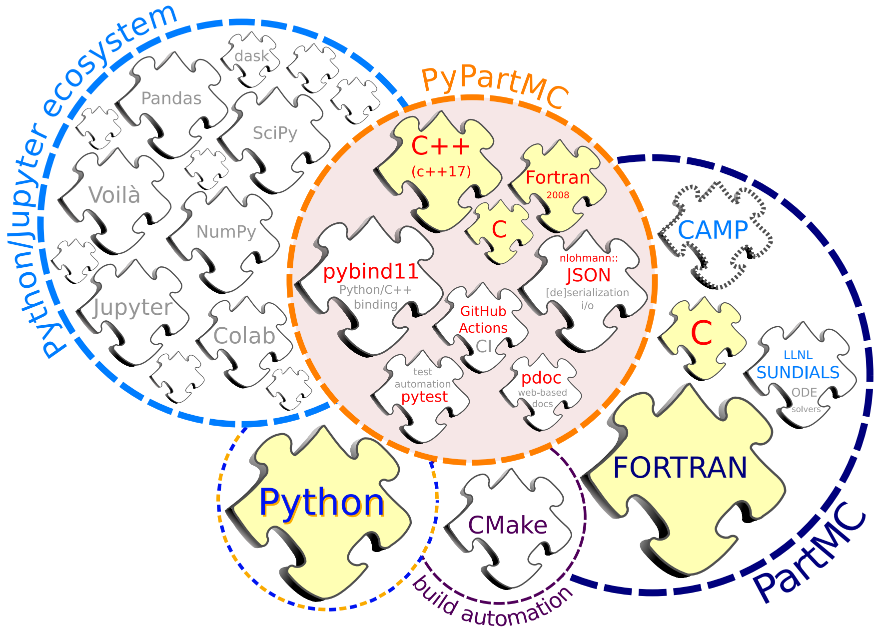
Version 2.6.1
Released 2022-02-18

Source: https://github.com/compdyn/partmc

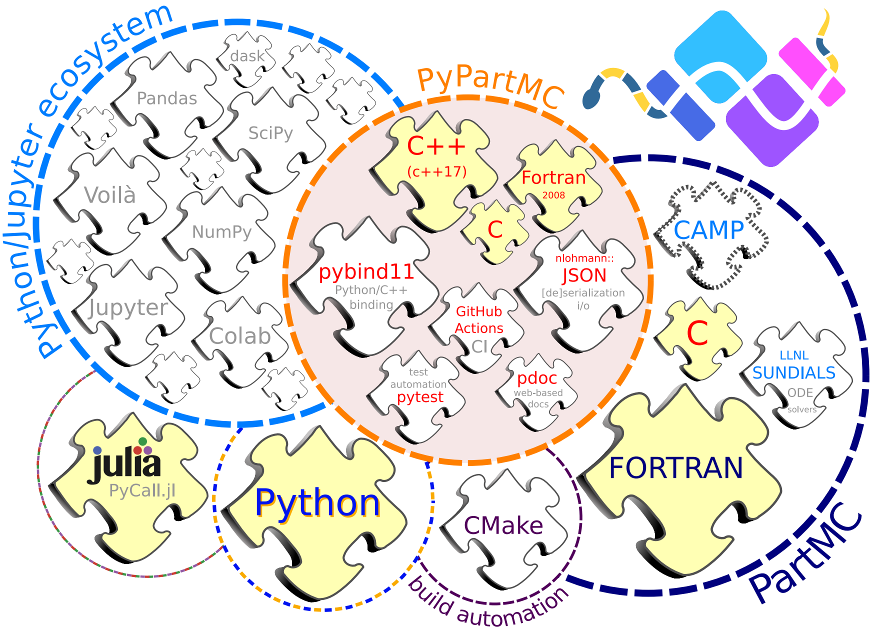
Homepage: http://lagrange.mechse.illinois.edu/partmc/

Cite as: M. West, N. Riemer, J. Curtis, M. Michelotti, and J. Tian (2022) PartMC, version v2.6.1, DOI 10.5281/zenodo.6144610


Python/Jupyter ecosystem
dask
Pandas
SciPy
Voilà
NumPy
Jupyter
Colab
?
CAMP
C
LLNL
SUNDIALS
ODE
solvers
FORTRAN
PartMC



Python/Jupyter ecosystem
dask
Pandas
SciPy
Voilà
NumPy
Jupyter
Colab
PyPartMC
C++
(c++17)
Fortran
2008
C
pybind11
Python/C++
binding
nlohmann::
JSON
[de]serialization
i/o
GitHub
Actions
CI
test
automation
pytest
pdoc
web-based
docs
CAMP
C
LLNL
SUNDIALS
ODE
solvers
FORTRAN
PartMC
Python
CMake
build automation


Python/Jupyter ecosystem
dask
Pandas
SciPy
Voilà
NumPy
Jupyter
Colab
PyPartMC
C++
(c++17)
Fortran
2008
C
pybind11
Python/C++
binding
nlohmann::
JSON
[de]serialization
i/o
GitHub
Actions
CI
test
automation
pytest
pdoc
web-based
docs
CAMP
C
LLNL
SUNDIALS
ODE
solvers
FORTRAN
PartMC
julia
PyCall.jl
Python
CMake
build automation

Search projects

Help Sponsors Log in Register

PyPartMC 0.0.25

```
pip install PyPartMC
```

Latest version

Released: Feb 21, 2023

Python interface to PartMC

Navigation

- Project description
- Release history
- Download files

Project links

- Documentation
- Source

Project description

PyPartMC (pre-alpha!)

PyPartMC is a Python interface to [PartMC](), a particle-resolved Monte-Carlo code for atmospheric aerosol simulation. Since PyPartMC is implemented in C++, it also constitutes a C++ API to the PartMC Fortran internals; the Python API can be used from other environments - see, e.g., Julia example below.

US DOE Funding by ASR | License GPL v3 | Copyright UIUC | Maintained? yes | tests passing | API docs pdoc3

- super-particle-count conserving probabilistic collisional breakup (Emily de Jong @Caltech)
  - $\rightsquigarrow$ https://doi.org/10.5194/egusphere-2022-1243

doctoral network links/prospects

- super-particle-count conserving probabilistic collisional breakup (Emily de Jong @Caltech)
  $\rightsquigarrow$ https://doi.org/10.5194/egusphere-2022-1243

- where turbulence needs to be parameterised:
  - condesation/evaporation vs. supersaturation fluctuations
  - (super-)particle advection and sedimentation
  - collision kernel (mixed-phase!)

doctoral network links/prospects

- super-particle-count conserving probabilistic collisional breakup (Emily de Jong @Caltech)
  $\rightsquigarrow$ https://doi.org/10.5194/egusphere-2022-1243

- where turbulence needs to be parameterised:
  - condesation/evaporation vs. supersaturation fluctuations
  - (super-)particle advection and sedimentation
  - collision kernel (mixed-phase!)

- Python/Jupyter ecosystem in coursework/workshop context – PySDM already used at:
  - "Modelling of Atmospheric Clouds" course at the Jagiellonian Univ.
  - "ESE 134. Cloud and Boundary Layer Dynamics" course at Caltech
  - "High-Performance Computing Summer School" at Univ. Kobe

Merci de votre attention

sylwester.arabas@uj.edu.pl (soon @agh.edu.pl)

collaborators:

N. Riemer, M. West, J. Curtis, Z. D'Aquino, ... (UIUC),

P. Bartman & O. Bulenok, ... (Jagiellonian),

E. de Jong, C. Singer, A. Jaruga, ... (Caltech)

...

funding:

Foundation for Polish Science, Poland's National Science Centre, US DOE (ASR)