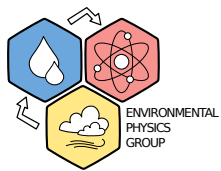
O modelowaniu chmur w chmurze
(i otwartych narzędziach do modelowania chmur rozwijanych na AGH)

Sylwester Arabas



seminarium SKNF Bozon AGH, 19.III.2024

plan prezentacji

- modelowanie chmur: fizyka i informatyka stosowana
- otwarte pakiety oprogramowania rozwijane na AGH
- demo (przykłady prac dyplomowych)
- tech stack, tematy prac domowych i dalsze losy absolwentów

plan prezentacji

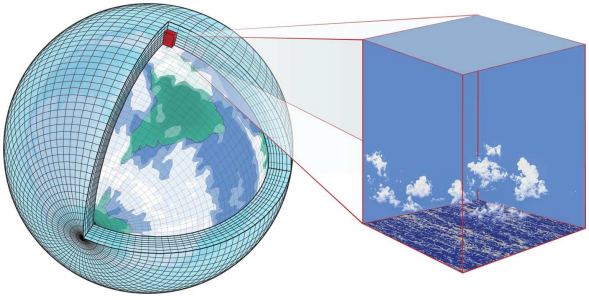
- modelowanie chmur: fizyka i informatyka stosowana
- otwarte pakiety oprogramowania rozwijane na AGH
- demo (przykłady prac dyplomowych)
- tech stack, tematy prac domowych i dalsze losy absolwentów

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

(photo: Yevgen Timashov / National Geographic)

"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)


MENU
ABOUT
DATA
DOCUMENTATION
FOCAL POINTS PORTAL
HELP
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

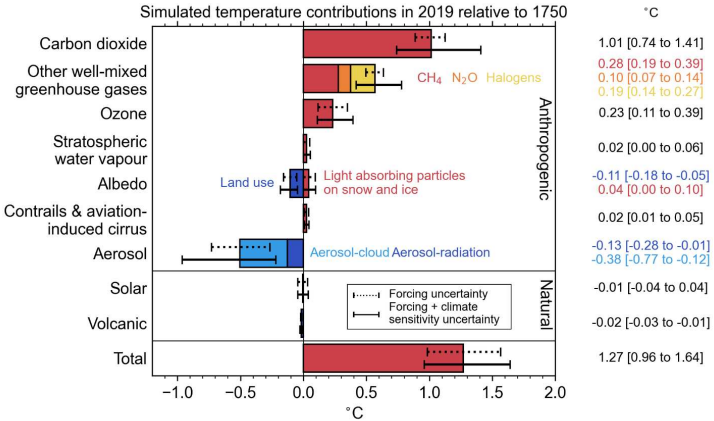
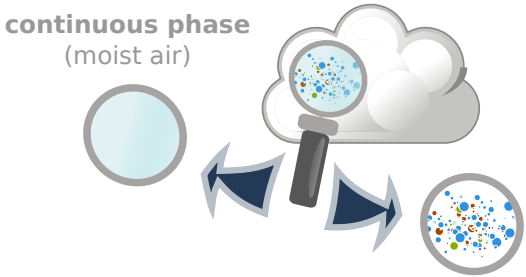
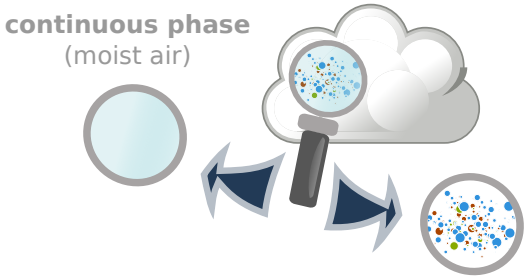


Figure 7.7: **The contribution of forcing agents to 2019 temperature change relative to 1750 produced using the two-layer emulator (Supplementary Material 7.SM.2), constrained to assessed ranges for key climate metrics described in Cross-Chapter Box 7.1.**



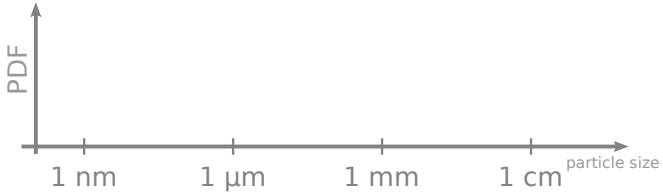
dispersed phase

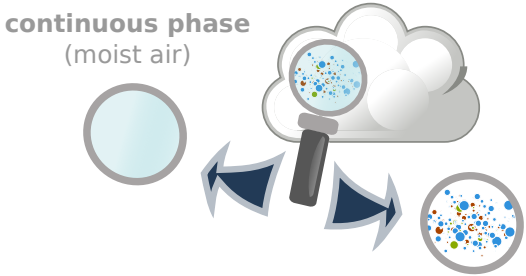
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)



dispersed phase

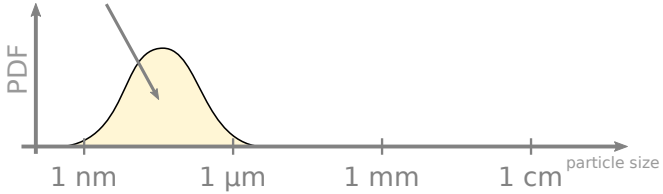
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)

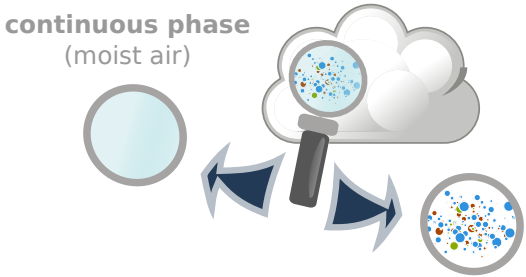




dispersed phase

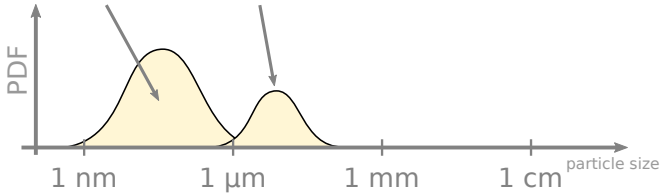
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)

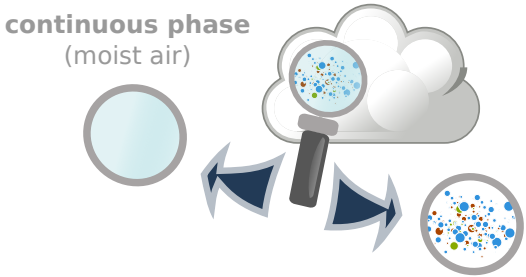




dispersed phase

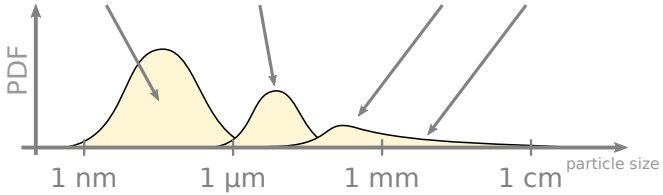
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)

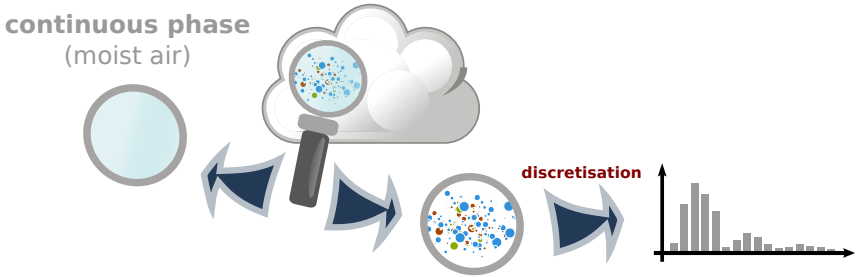




dispersed phase

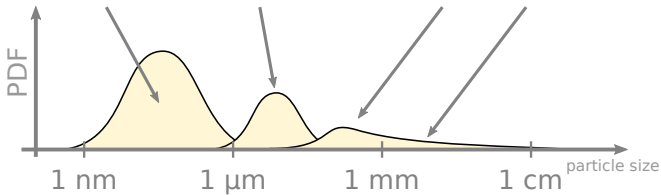
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)

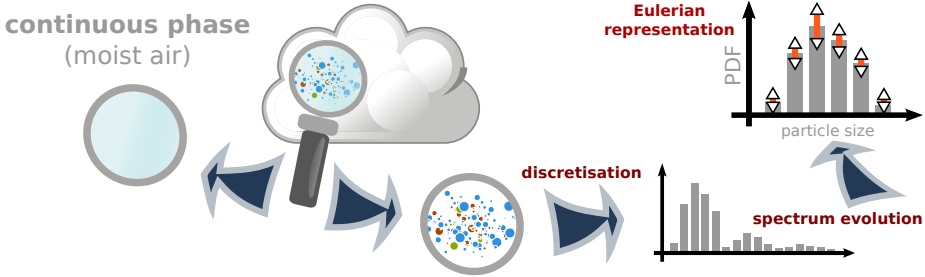




dispersed phase

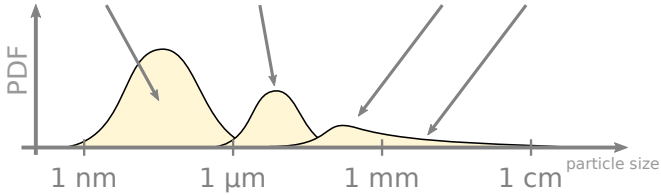
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)





dispersed phase

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



równania koagulacji Smoluchowskiego (SCE)

liczebność cząstek o rozmiarze x w czasie t: $c(x,t)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$
prawdopodobieństwo zderzeń: $a(x_1, x_2)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$

równania koagulacji Smoluchowskiego (SCE)

liczebność cząstek o rozmiarze x w czasie t: $c(x,t)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$
prawdopodobieństwo zderzeń: $a(x_1, x_2)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$

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

równania koagulacji Smoluchowskiego (SCE)

liczebność cząstek o rozmiarze x w czasie t: $c(x,t)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$
prawdopodobieństwo zderzeń: $a(x_1, x_2)$: $\mathbb{R}^+ \times \mathbb{R}^+ \to \mathbb{R}^+$

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

zdyskretyzowana postać: $c_i = c(x_i)$ gdzie $x_i = i \cdot x_0$

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

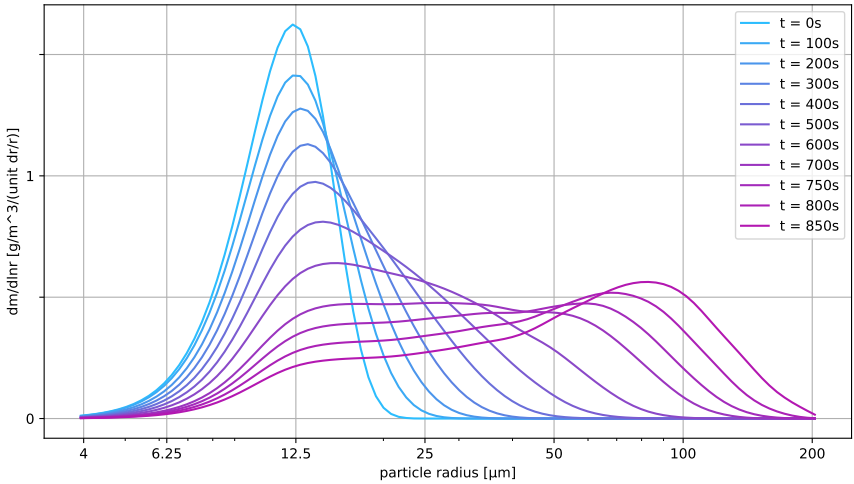


figure (PySDM simulation): Bartman, Arabas et al. 2021, LNCS (doi:10.1007/978-3-030-77964-1_2)

- rozwiązania analityczne znane tylko dla prostych f-cji $a(x, y)$

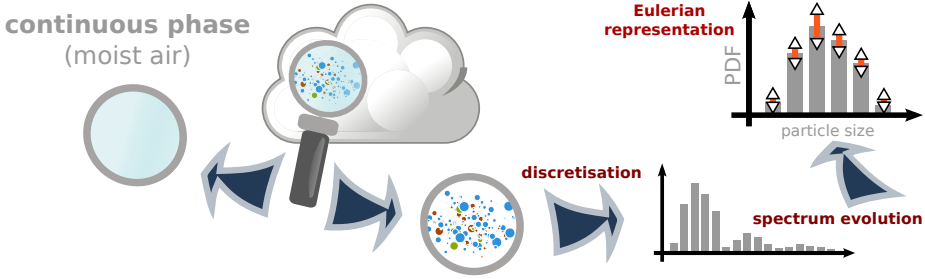
[a] https://pl.wikipedia.org/wiki/Przekle%C5%84stwo_wymiarowo%C5%9Bci

- rozwiązania analityczne znane tylko dla prostych f-cji $a(x, y)$
- "przekleństwo wymiarowości"[a] na przeszkodzie wykorzystania metod numerycznych (gdy rozróżniamy cząstki o tych samych rozmiarach, ale innym składzie/ładunku/...)

[a] https://pl.wikipedia.org/wiki/Przekle%C5%84stwo_wymiarowo%C5%9Bci

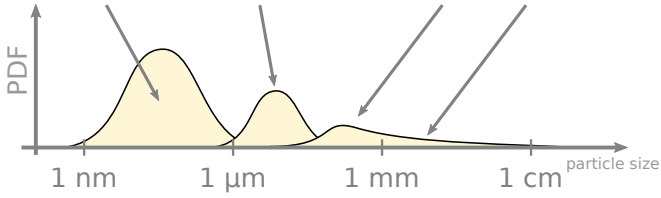
- rozwiązania analityczne znane tylko dla prostych f-cji $a(x, y)$
- "przekleństwo wymiarowości"[a] na przeszkodzie wykorzystania metod numerycznych
  (gdy rozróżniamy cząstki o tych samych rozmiarach, ale innym składzie/ładunku/...)
- założenia potrzebne do wyprowadzenia SCE mogą nie być spełnione w atmosferze!
  (np. wystarczająco duża populacja aby opisać cząstki za pomocą rozkładu prawd.)
- ...

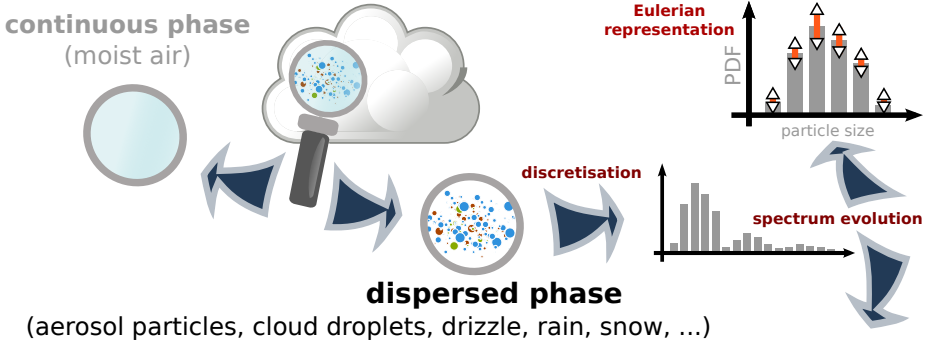
[a] https://pl.wikipedia.org/wiki/Przekle%C5%84stwo_wymiarowo%C5%9Bci



dispersed phase

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




continuous phase
(moist air)
Eulerian
representation
PDF
particle size
discretisation
spectrum evolution
dispersed phase
(aerosol particles, cloud droplets, drizzle, rain, snow, ...)
PDF
1 nm
1 µm
1 mm
1 cm
particle size
Lagrangian
representation
PDF
particle size

SDM: alternatywa dla równań Smoluchowskiego

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

SDM: alternatywa dla równań Smoluchowskiego

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

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

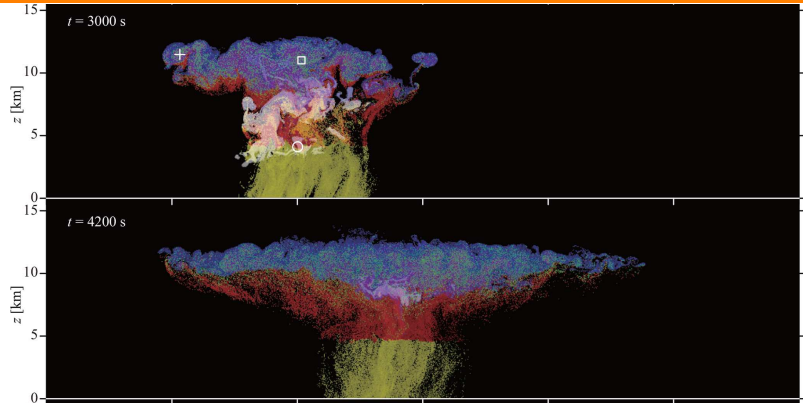


Figure 1. Typical realization of CTRL cloud spatial structures at $t = 2040$, 2460, 3000, 4200, and 5400 s. The mixing ratio of cloud water, rainwater, cloud ice, graupel, and snow aggregates are plotted in fading white, yellow, blue, red, and green, respectively. The symbols indicate examples of unrealistic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

zdjęcie (CC-BY): T. Matsui / `https://en.wikipedia.org/wiki/K_computer`

przykładowa animacja:
`https://zenodo.org/records/3841697/files/Movie01.QHYD_TYP-CTRL_2.2.0.gif`

SCE (naiwna implementacja)	SDM
typ symulacji	
deterministyczna	Monte-Carlo (wymaga wielu realizacji)

SCE (naiwna implementacja)	SDM
typ symulacji	
deterministyczna	Monte-Carlo (wymaga wielu realizacji)
zderzane pary rozmiarów kropel	
wszystkie pary (i,j)	losowy zbiór $n_{sd}/2$ niepokrywających się par, prawdopodobieństwo skalowane o czynnik $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$

SCE (naiwna implementacja)	SDM
typ symulacji	
deterministyczna	Monte-Carlo (wymaga wielu realizacji)
zderzane pary rozmiarów kropel	
wszystkie pary (i,j)	losowy zbiór $n_{sd}/2$ niepokrywających się par, prawdopodobieństwo skalowane o czynnik $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$
złożoność obliczeniowa	
$\mathcal{O}(n_{sd}^2)$ (zależności danych pomiędzy parami)	$\mathcal{O}(n_{sd})$ ("embarrassingly parallel"!)

SCE (naiwna implementacja)	SDM
typ symulacji	
deterministyczna	Monte-Carlo (wymaga wielu realizacji)
zderzane pary rozmiarów kropel	
wszystkie pary (i,j)	losowy zbiór $n_{sd}/2$ niepokrywających się par, prawdopodobieństwo skalowane o czynnik $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$
złożoność obliczeniowa	
$\mathcal{O}(n_{sd}^2)$ (zależności danych pomiędzy parami)	$\mathcal{O}(n_{sd})$ ("embarrassingly parallel"!)
zajście koagulacji	
w każdym kroku	wynik porównania prawdopodobieństwa z liczbą losową

SCE (naiwna implementacja)	SDM
typ symulacji	
deterministyczna	Monte-Carlo (wymaga wielu realizacji)
zderzane pary rozmiarów kropel	
wszystkie pary (i,j)	losowy zbiór $n_{sd}/2$ niepokrywających się par, prawdopodobieństwo skalowane o czynnik $(n_{sd}^2 - n_{sd})/2$ to $n_{sd}/2$
złożoność obliczeniowa	
$\mathcal{O}(n_{sd}^2)$ (zależności danych pomiędzy parami)	$\mathcal{O}(n_{sd})$ ("embarrassingly parallel"!)
zajście koagulacji	
w każdym kroku	wynik porównania prawdopodobieństwa z liczbą losową
wynik koagulacji	
koaguluje część z $c_{[i]}$, $c_{[j]}$ kropel	koagulują wszystkie z $\min\{c_{[i]}, c_{[j]}\}$ kropel

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik
- **NTLP** @ncar.ucar.edu Fortran, integralna część pakietu CFD, 1 plik

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik
- **NTLP** @ncar.ucar.edu Fortran, integralna część pakietu CFD, 1 plik
- **libcloudph++** @uw.edu.pl C++, wsparcie dla GPU, biblioteka

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik
- **NTLP** @ncar.ucar.edu Fortran, integralna część pakietu CFD, 1 plik
- **libcloudph++** @uw.edu.pl C++, wsparcie dla GPU, biblioteka
- **LCM1D** @dlr.de Fortran/C/shell

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik
- **NTLP** @ncar.ucar.edu Fortran, integralna część pakietu CFD, 1 plik
- **libcloudph++** @uw.edu.pl C++, wsparcie dla GPU, biblioteka
- **LCM1D** @dlr.de Fortran/C/shell
- **superdroplet** @mit.edu Cython/Numba/C++/Fortran 2008/Julia - "hello world"

otwarte implementacje SDM:

- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik
- **NTLP** @ncar.ucar.edu Fortran, integralna część pakietu CFD, 1 plik
- **libcloudph++** @uw.edu.pl C++, wsparcie dla GPU, biblioteka
- **LCM1D** @dlr.de Fortran/C/shell
- **superdroplet** @mit.edu Cython/Numba/C++/Fortran 2008/Julia - "hello world"
- **PySDM** @agh.edu.pl Python: Numba (LLVM+OpenMP) + ThrustRTC (GPU)

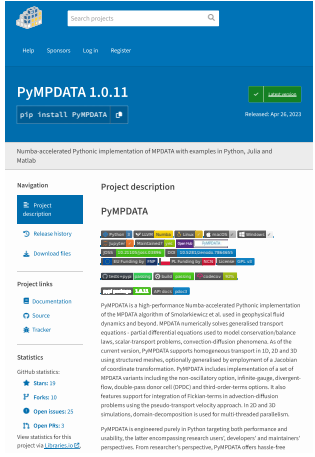
otwarte implementacje SDM:

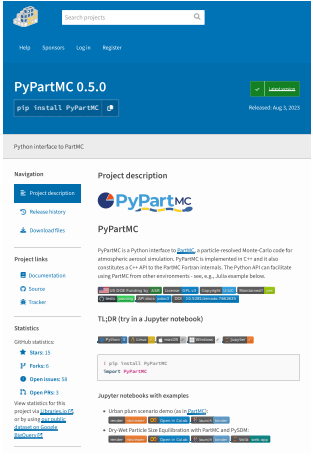
- **SCALE-SDM** @RIKEN.jp (Fortran, integralna część pakietu CFD, 1 plik)
- **Pencil Code** @Nordita.org Fortran, integralna część pakietu CFD, 1 plik)
- **PALM** @uni-hannover.de Fortran, integralna część pakietu CFD, 1 plik
- **NTLP** @ncar.ucar.edu Fortran, integralna część pakietu CFD, 1 plik
- **libcloudph++** @uw.edu.pl C++, wsparcie dla GPU, biblioteka
- **LCM1D** @dlr.de Fortran/C/shell
- **superdroplet** @mit.edu Cython/Numba/C++/Fortran 2008/Julia - "hello world"
- **PySDM** @agh.edu.pl Python: Numba (LLVM+OpenMP) + ThrustRTC (GPU)
- **CLEO** @mpimet.mpg.de C++ (incl. Python API)

plan prezentacji

- modelowanie chmur: fizyka i informatyka stosowana
- otwarte pakiety oprogramowania rozwijane na AGH
- demo (przykłady prac dyplomowych)
- tech stack, tematy prac domowych i dalsze losy absolwentów

github.com/open-atmos





2D flow field

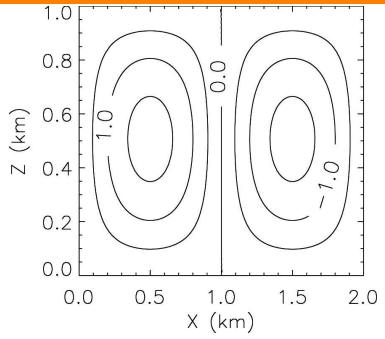
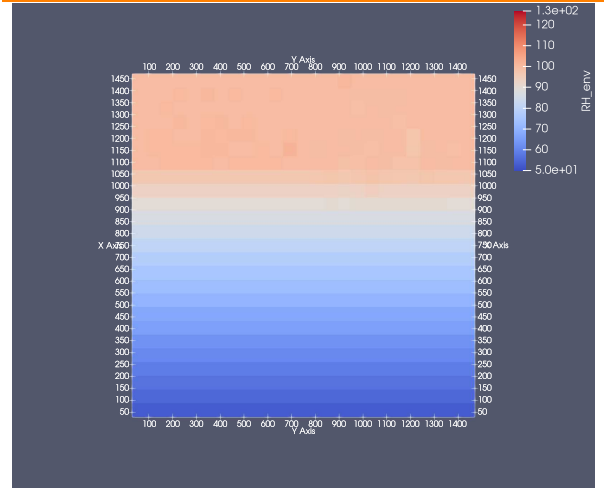
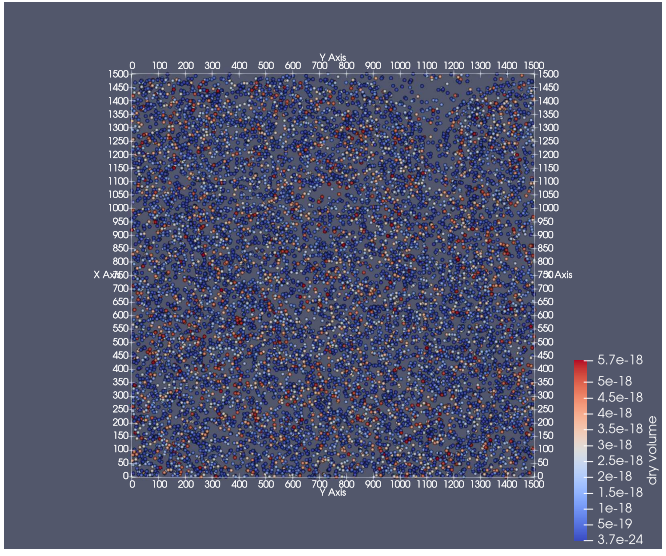
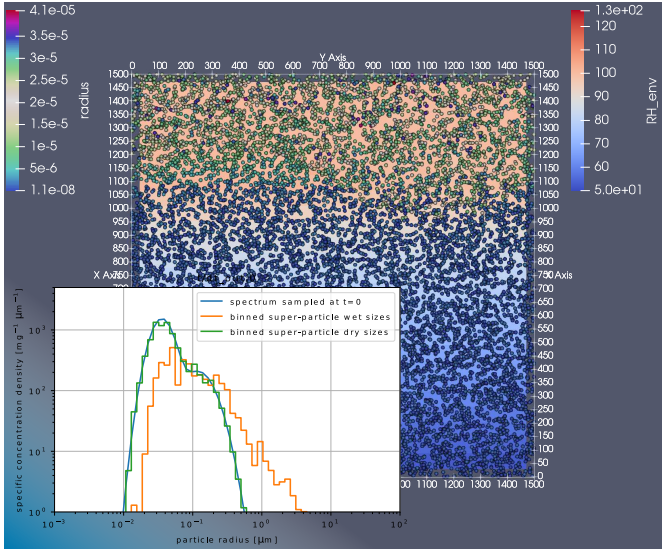


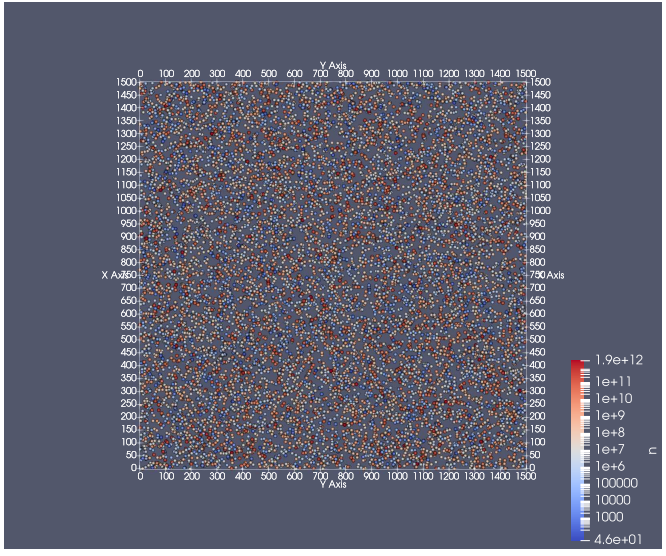
FIG. 1. Time-invariant vertical velocity for the stratocumulus case (contour interval is 0.5 m s^{-1}).

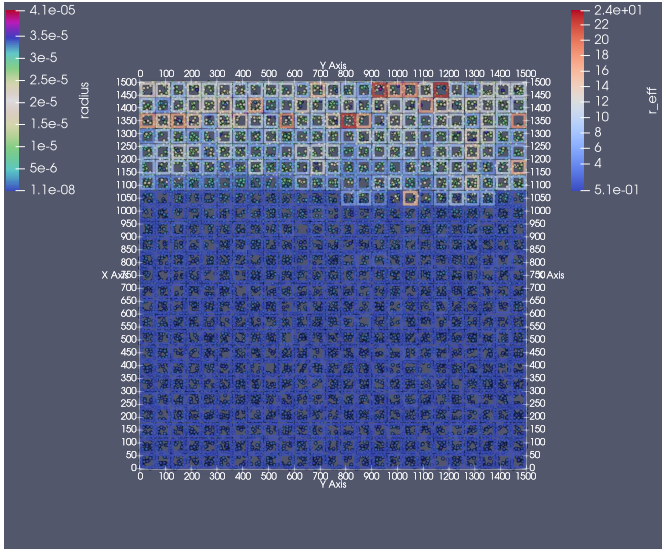
RH profile at t=0



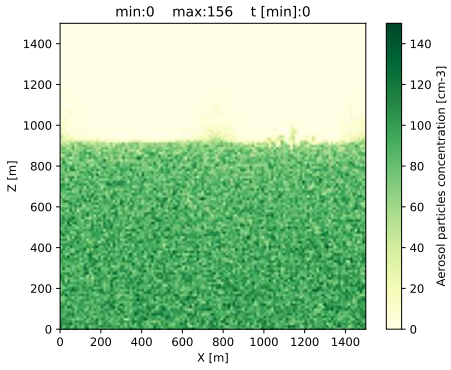

Y Axis
X Axis
Y Axis
X Axis
0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500
5.7e-18
5e-18
4.5e-18
4e-18
3.5e-18
3e-18
2.5e-18
2e-18
1.5e-18
1e-18
5e-19
3.7e-24
dp_volume

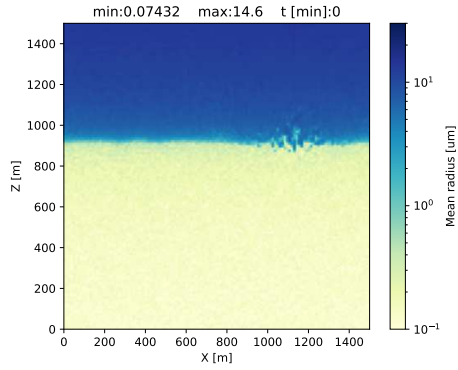

radius
4.1e-05
3.5e-5
3e-5
2.5e-5
2e-5
1.5e-5
1e-5
5e-6
1.1e-08
RH_env
1.3e+02
120
110
100
90
80
70
60
5.0e+01
Y Axis
X Axis
spectrum sampled at t=0
binned super-particle wet sizes
binned super-particle dry sizes
specific concentration density [mg⁻¹ µm⁻¹]
particle radius [µm]


Y Axis
0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500
X Axis
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 1450 1500
Z Axis
0 50 100 150 200 250 300 350 400 450 500 550 600 650 700 750 800 850 900 950 1000 1050 1100 1150 1200 1250 1300 1350 1400 1450 1500
n
1.9e+12
1e+11
1e+10
1e+9
1e+8
1e+7
1e+6
100000
10000
1000
4.6e+01


4.1e-05
3.5e-5
3e-5
2.5e-5
2e-5
1.5e-5
1e-5
5e-6
1.1e-08
radius
2.4e+01
22
20
18
16
14
12
10
8
6
4
5.1e-01
f_eff
Y Axis
X Axis
0 100 200 300 400 500 600 700 800 900 1000 1100 1200 1300 1400 1500

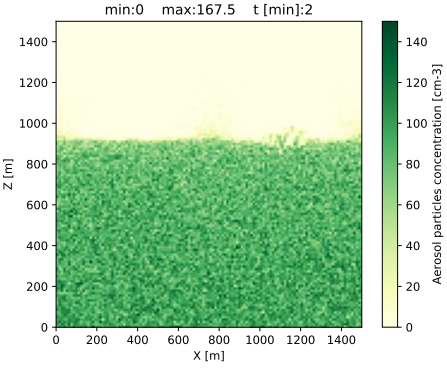
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

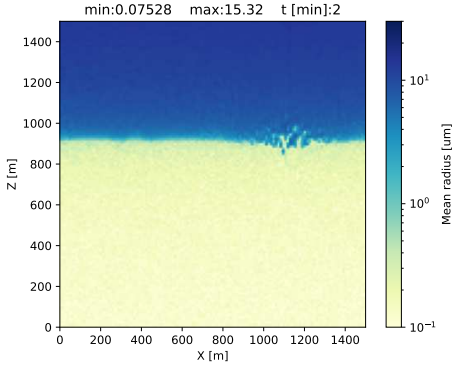




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

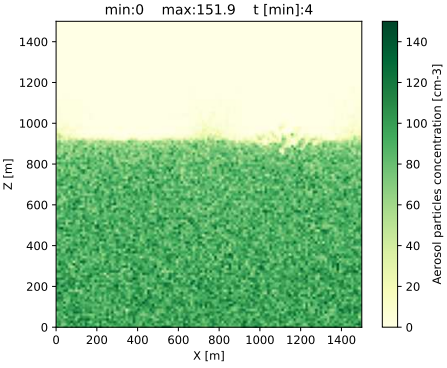
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

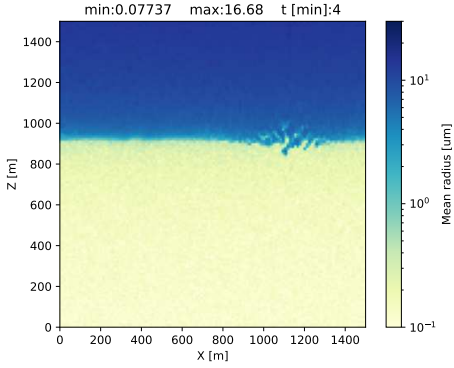




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

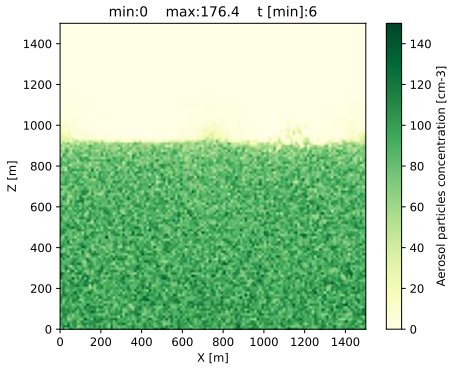
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

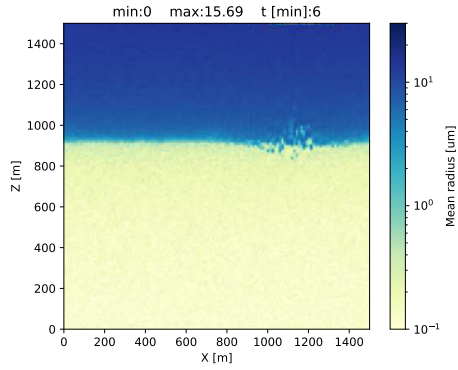




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

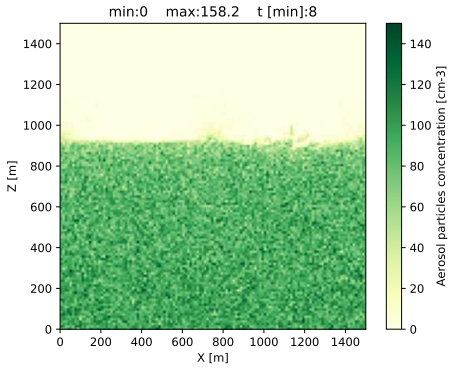
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

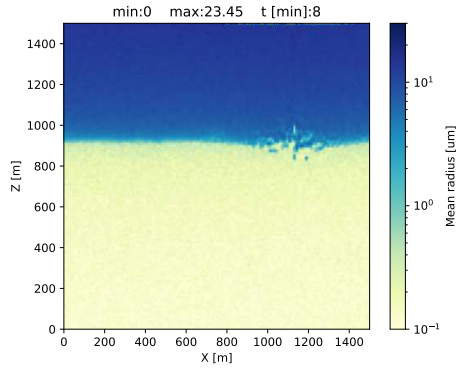




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

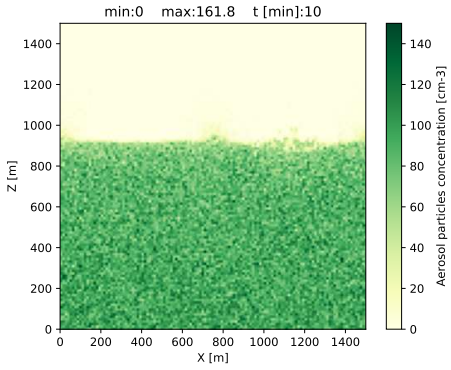
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

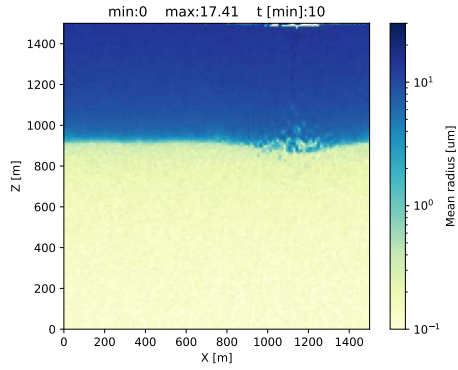




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

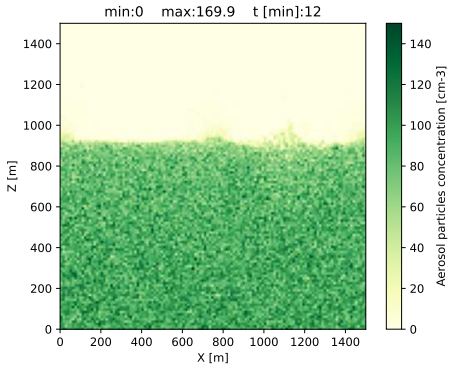
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

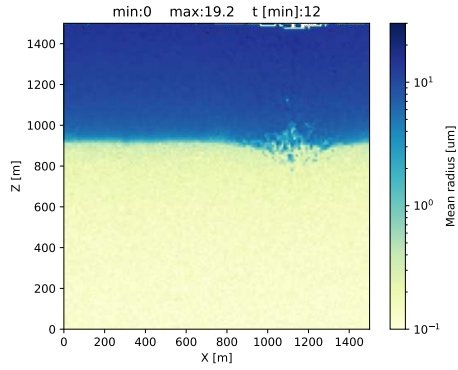




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

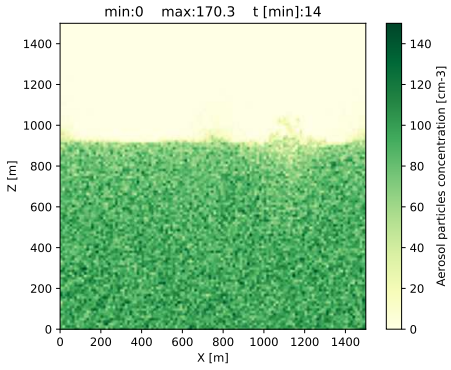
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

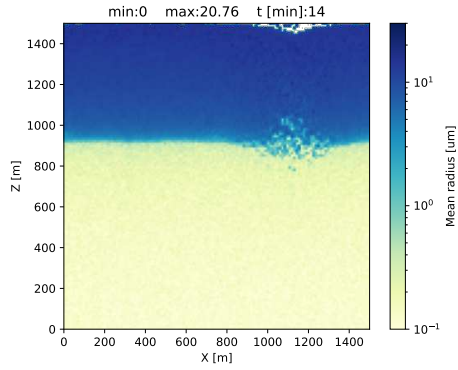




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

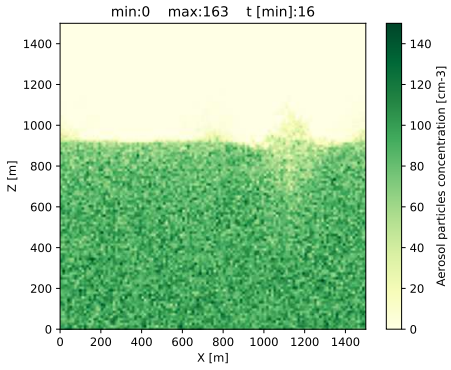
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

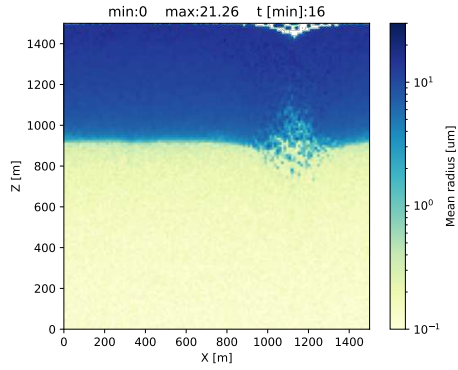




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

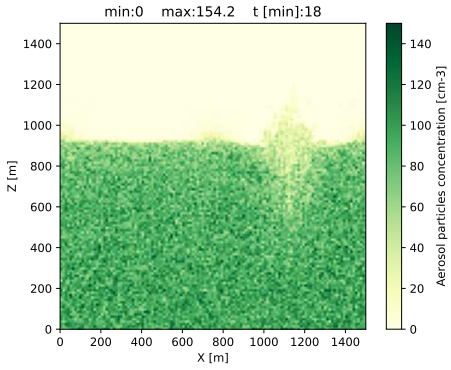
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

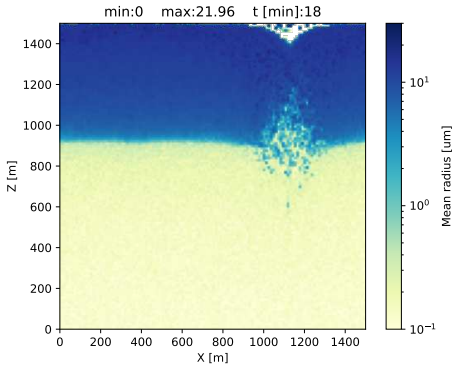




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

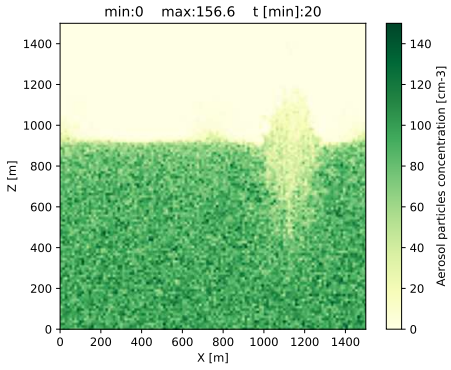
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

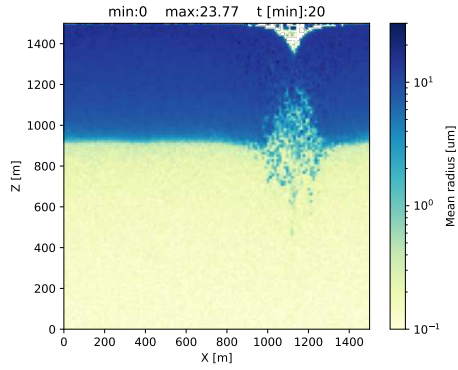




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

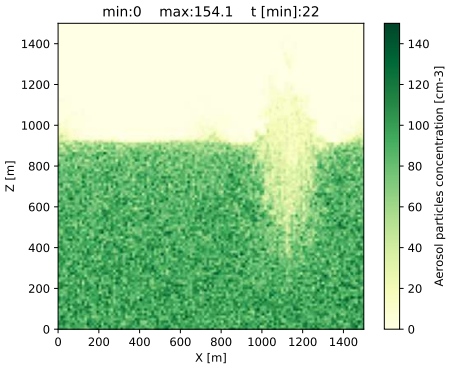
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

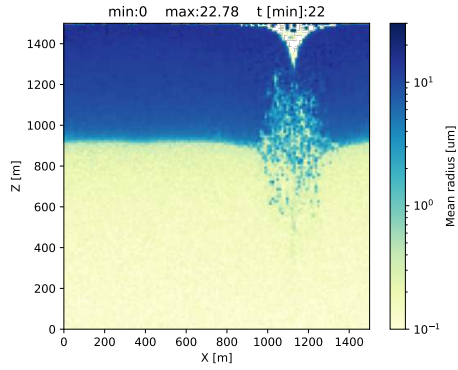




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

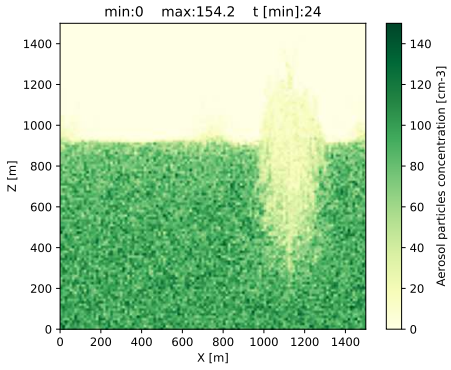
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

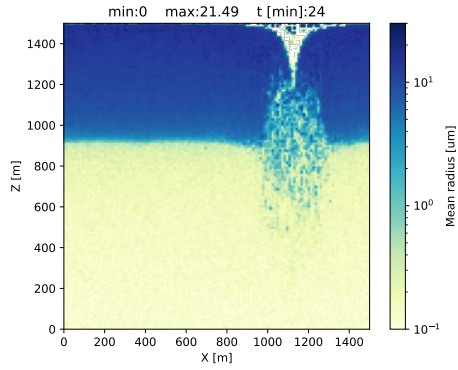




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

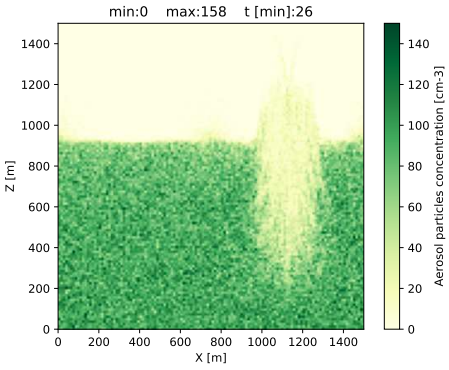
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

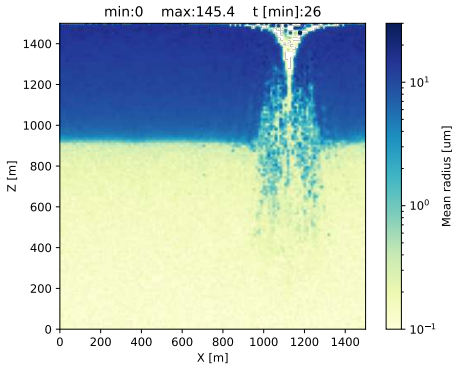




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

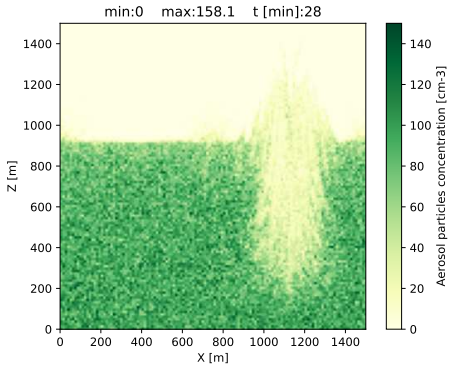
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

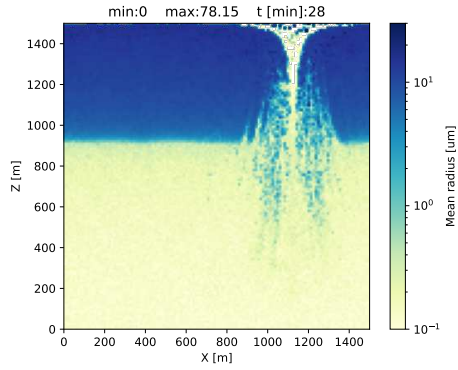




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

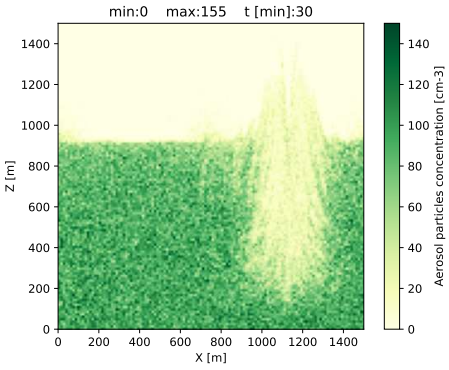
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

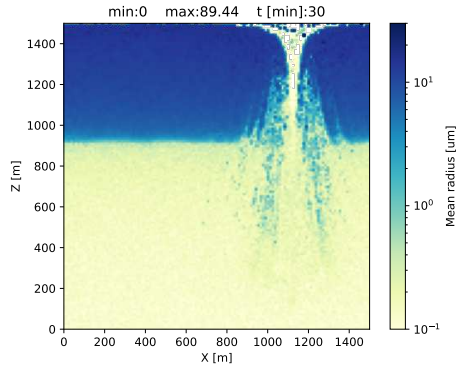




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

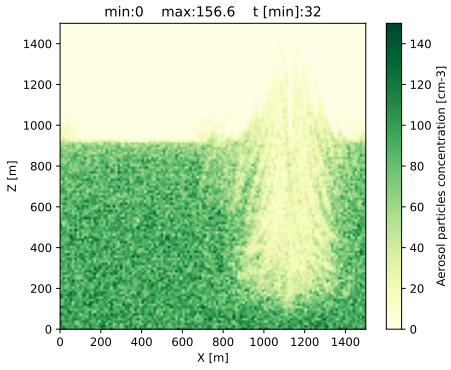
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

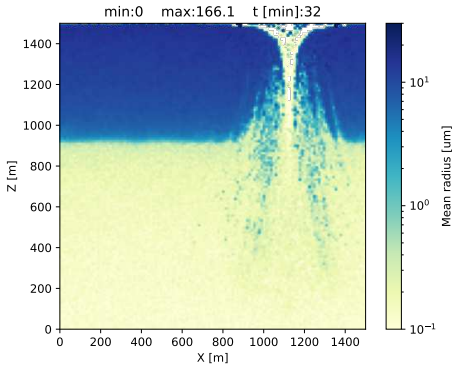




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

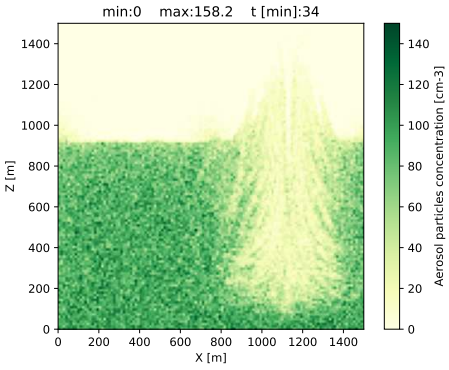
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

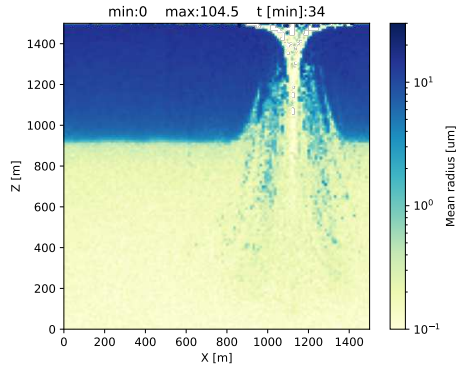




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

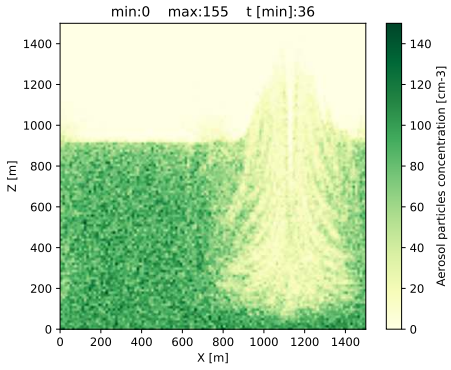
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

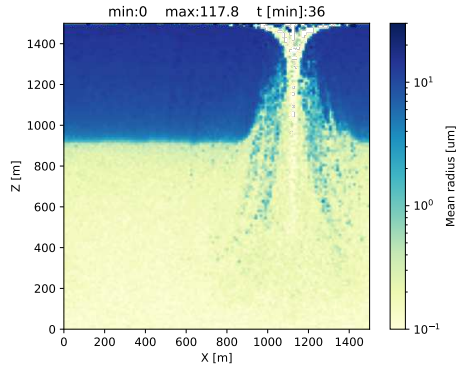




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

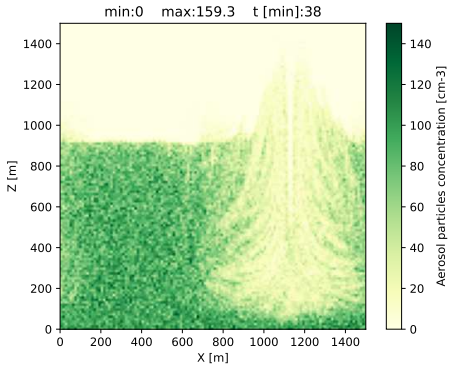
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

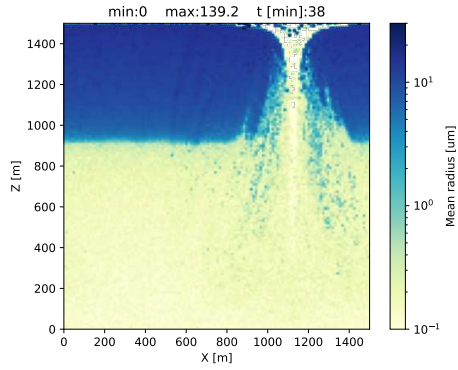




symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

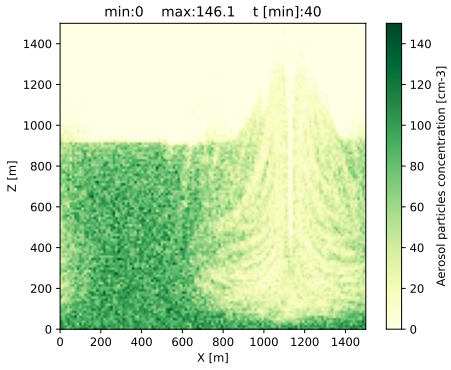
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}

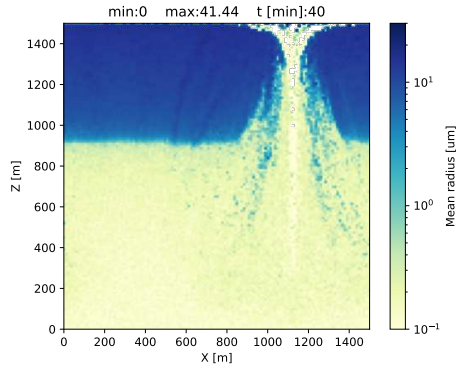




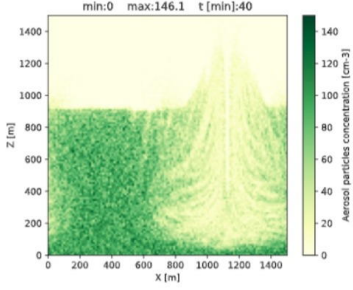
symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)

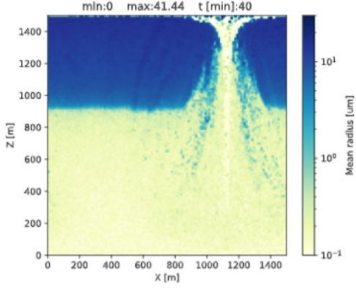
Siatka obliczeniowa: 128×128
Populacja cząstek obliczeniowych: 2^{21}



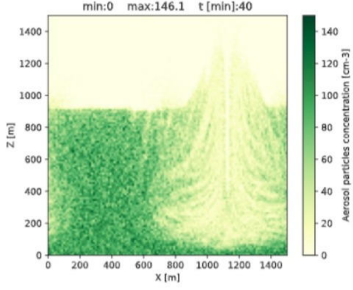


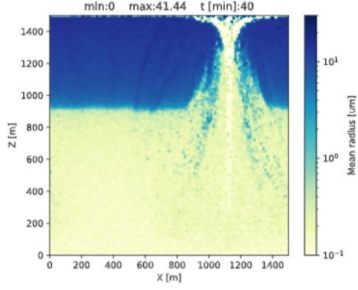
symulacja i wizualizacja: Piotr Bartman (praca magisterska @ WMiI UJ)


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 [μm]

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





demo.ipynb ☆

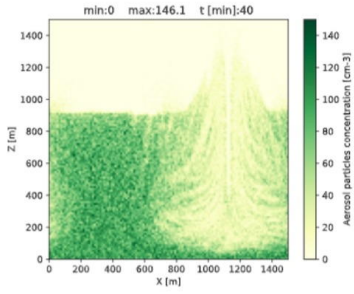
File Edit View Insert Runtime Tools Help

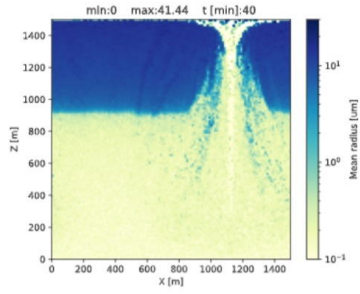
Comment Share P

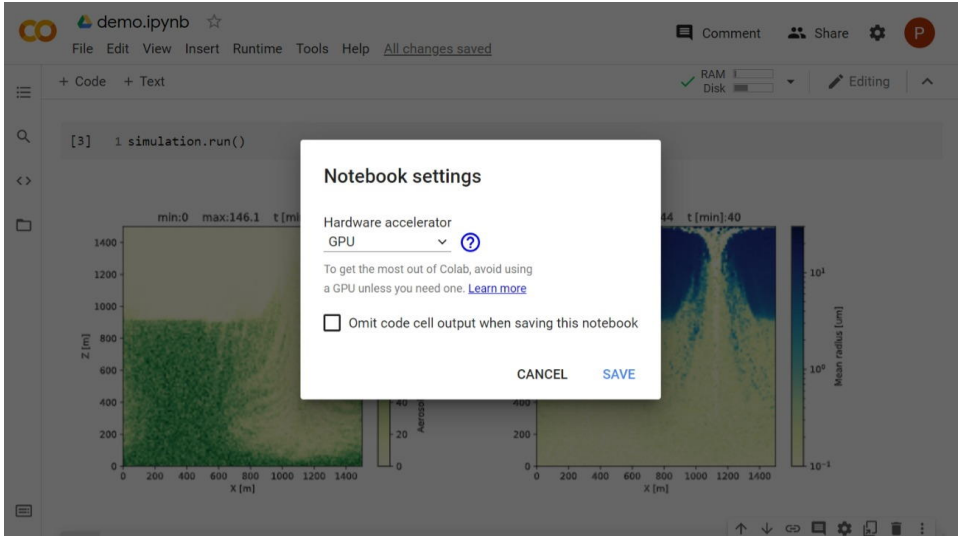
+ Code + Text

RAM Disk Editing

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

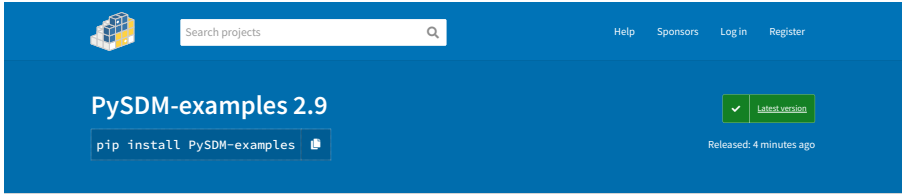





demo.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
[3] 1 simulation.run()
Notebook settings
Hardware accelerator
GPU
To get the most out of Colab, avoid using a GPU unless you need one. Learn more
Omit code cell output when saving this notebook
CANCEL
SAVE
min:0 max:146.1 t [min]:40
Z [m]
X [m]
Mean radius [um]

plan prezentacji

- modelowanie chmur: fizyka i informatyka stosowana
- otwarte pakiety oprogramowania rozwijane na AGH
- demo (przykłady prac dyplomowych)
- tech stack, tematy prac domowych i dalsze losy absolwentów


Search projects
Help
Sponsors
Log in
Register
PySDM-examples 2.9
pip install PySDM-examples
Latest version
Released: 4 minutes ago
PySDM usage examples reproducing results from literature and depicting how to use PySDM from Python Jupyter notebooks
Navigation
Project description
Release history
Download files
Project links
Homepage
Statistics
GitHub statistics:
Stars: 2
Forks: 10
Open issues/PRs: 2
Project description
License GPL v3
Copyright Jagiellonian University
DOI 10.5281/zenodo.6604645
PySDM-examples passing
pull requests 2 open
pull requests 159 closed
pypi package 2.0
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

nature COMMUNICATIONS

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

Key drivers of cloud response to surface-active organics

S.J. Lowe[1,2], D.G. Partridge[3], J.F. Davies[4], K.R. Wilson[5], D. Topping[6] & I. Riipinen[1,2,7]*

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

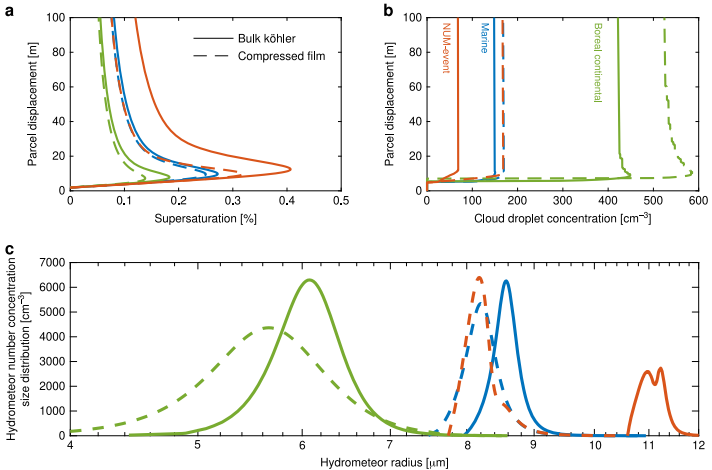


Fig. 2 Simulated microphysics of cloud events on marine (MA, blue), boreal (HYY, green) and NUM-event (NE, orange) aerosol populations. Cloud-formation event simulations using bulk Köhler BK (solid lines) and approximate compressed film CF (dotted lines) models of cloud droplet activation with initial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

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

File Edit View Run Kernel Tabs Settings Help

Launcher | fig_2.ipynb

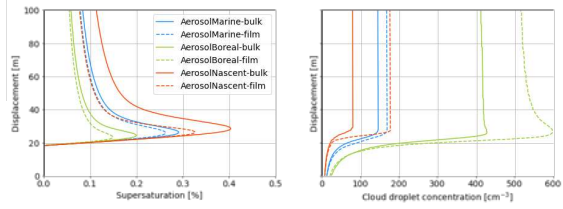
Download GitHub Binder Code

Filter files by name

/ PySDM_examples / Lowe_et_al_2019 /

Name	Last Modified
__init__.py	4 hours ago
aerosol.py	4 hours ago
fig_1.ipynb	4 hours ago
fig_2.ipynb	4 hours ago
settings.py	4 hours ago
simulation.py	4 hours ago

```
[4]: figsize = (15, 5)
     pylab.rc('font', size=14)
     fig, axs = 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]['z'])
             axs[idx].plot(output[key][var], Y, label=key,
                           color=output[key]['color'],
                           linestyle='-' if key.endswith('-bulk') else '--'
                           )
         axs[idx].set_ylim(0, 100)

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

     for ax in axs:
         ax.grid()
     axs[0].legend()
```



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

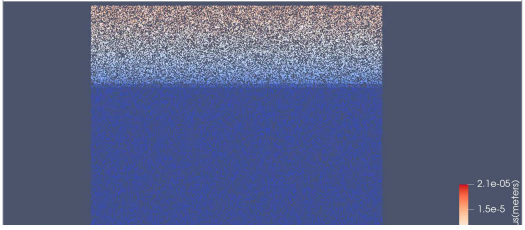
This PR includes a coupling logic for ClimateMachine.jl and PySDM.

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

plan prezentacji

- modelowanie chmur: fizyka i informatyka stosowana
- otwarte pakiety oprogramowania rozwijane na AGH
- demo (przykłady prac dyplomowych)
- tech stack, tematy prac domowych i dalsze losy absolwentów

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

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

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

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

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

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

https://github.com/numba-mpi/numba-mpi

numba-mpi

Python 3 | LLVM Numba | Linux ✓ | macOS ✓ | Windows ✓ | tests passing | Pylint passing | Maintained? yes | License GPL v3 | pypi package 0.26 | Anaconda.org 0.26 | DOI 10.5281/zenodo.7385622

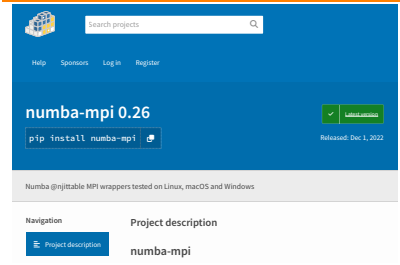
Numba @njittable MPI wrappers

- covering: `size` / `rank` , `send` / `recv` , `allreduce` , `bcast` , `barrier`
- API based on NumPy and supporting numeric and character datatypes
- auto-generated docstring-based API docs on the web: https://numba-mpi.github.io/numba-mpi
- pure-Python implementation with packages available on PyPI and Conda Forge
- CI-tested on: Linux (MPICH, OpenMPI & Intel MPI), macOS (MPICH & OpenMPI) and Windows (MS MPI)

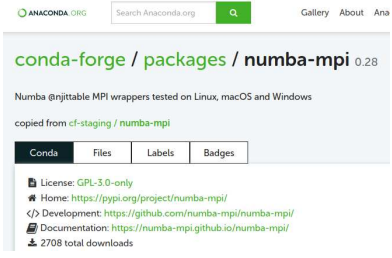
Hello world example:

```
import numba, numba_mpi, numpy

@numba.njit()
def hello():
  print(numba_mpi.rank())
  print(numba_mpi.size())

  src = numpy.array([1., 2., 3., 4., 5.])
  dst_tst = numpy.empty_like(src)

  if numba_mpi.rank() == 0:
    numba_mpi.send(src, dest=1, tag=11)
```

https://pypi.org/project/numba-mpi



https://anaconda.org/conda-forge/numba-mpi



dotychczas zrealizowane prace dyplomowe

- Piotr Bartman (informatyka) – aktualnie @invisiblethingslab.com & doktorant @UJ
  „PySDM v1.0: Pythonic particle-based cloud microphysics package"
  ⇝ `https://ap.uj.edu.pl/diplomas/141204/`
- Michael Olesik (fizyka) – aktualnie doktorant @UJ
  „On higher-order finite differencing for condensational growth in particulate systems"
  ⇝ `https://ap.uj.edu.pl/diplomas/141977/`
- Kacper Derlatka (informatyka) – aktualnie @pega.com
  „Distributed-memory parallelism with Python: MPI support in PyMPDATA..."
  ⇝ `https://ap.uj.edu.pl/diplomas/166883/`
- Oleksii Bulenok (informatyka) – aktualnie @swmansion.com
  „Monte-Carlo collisional breakup in PySDM: GPU support and validation ..."
  ⇝ `https://ap.uj.edu.pl/diplomas/166879/`

propozycje tematów prac dyplomowych

propozycje tematów prac dyplomowych

- **PySDM** (fizyka chmur, metody Monte-Carlo):
  inż: wizualizacje 3D w ramach CI
  mgr: rozszerzenie o śledzenie temperatur kropel

propozycje tematów prac dyplomowych

- **PySDM** (fizyka chmur, metody Monte-Carlo):
  inż: wizualizacje 3D w ramach CI
  mgr: rozszerzenie o śledzenie temperatur kropel
- **PyMPDATA** (równania różniczkowe cząstkowe, numeryka):
  inż: dynamika płynu w przybl. Boussinesq'a (bazując na przykł. z libmpdata++)
  mgr: wycena opcji azjatyckich (matematyka finansowa)

propozycje tematów prac dyplomowych

- **PySDM** (fizyka chmur, metody Monte-Carlo):
  - inż: wizualizacje 3D w ramach CI
  - mgr: rozszerzenie o śledzenie temperatur kropel
- **PyMPDATA** (równania różniczkowe cząstkowe, numeryka):
  - inż: dynamika płynu w przybl. Boussinesq'a (bazując na przykł. z libmpdata++)
  - mgr: wycena opcji azjatyckich (matematyka finansowa)
- **numba-mpi** (komunikacja między węzłami superkomputerów):
  - inż: wsparcie dla komunikacji w "podgrupach" węzłów superkomputera
  - mgr: wsparcie dla komunikacji typów definiowanych przez użytkownika

propozycje tematów prac dyplomowych

- **PySDM** (fizyka chmur, metody Monte-Carlo):
  - inż: wizualizacje 3D w ramach CI
  - mgr: rozszerzenie o śledzenie temperatur kropel
- **PyMPDATA** (równania różniczkowe cząstkowe, numeryka):
  - inż: dynamika płynu w przybl. Boussinesq'a (bazując na przykł. z libmpdata++)
  - mgr: wycena opcji azjatyckich (matematyka finansowa)
- **numba-mpi** (komunikacja między węzłami superkomputerów):
  - inż: wsparcie dla komunikacji w "podgrupach" węzłów superkomputera
  - mgr: wsparcie dla komunikacji typów definiowanych przez użytkownika
- **PyMPDATA-MPI** (PyMPDATA @ Cyfronet):
  - inż: rozszerzenie na 3D (aktualnie przykłady jedynie dla 2D)
  - mgr: przykład rozwiązujący równania płytkiej wody na sferze

propozycje tematów prac dyplomowych

- **PySDM** (fizyka chmur, metody Monte-Carlo):
  - inż: wizualizacje 3D w ramach CI
  - mgr: rozszerzenie o śledzenie temperatur kropel
- **PyMPDATA** (równania różniczkowe cząstkowe, numeryka):
  - inż: dynamika płynu w przybl. Boussinesq'a (bazując na przykł. z libmpdata++)
  - mgr: wycena opcji azjatyckich (matematyka finansowa)
- **numba-mpi** (komunikacja między węzłami superkomputerów):
  - inż: wsparcie dla komunikacji w "podgrupach" węzłów superkomputera
  - mgr: wsparcie dla komunikacji typów definiowanych przez użytkownika
- **PyMPDATA-MPI** (PyMPDATA @ Cyfronet):
  - inż: rozszerzenie na 3D (aktualnie przykłady jedynie dla 2D)
  - mgr: przykład rozwiązujący równania płytkiej wody na sferze
- **PyPartMC**: (interfejs Fortran $\rightsquigarrow$ C++ $\rightsquigarrow$ Python)
  - inż: przykład użycia w C++ (aktualnie: Python, Matlab & Julia)
  - mgr: porównanie PyPartMC vs. PySDM vs. rozwiązanie analityczne

propozycje tematów prac dyplomowych (pl/en; stypendia naukowe NCN!)

- **PySDM** (fizyka chmur, metody Monte-Carlo):
  - inż: wizualizacje 3D w ramach CI
  - mgr: rozszerzenie o śledzenie temperatur kropel
- **PyMPDATA** (równania różniczkowe cząstkowe, numeryka):
  - inż: dynamika płynu w przybl. Boussinesq'a (bazując na przykł. z libmpdata++)
  - mgr: wycena opcji azjatyckich (matematyka finansowa)
- **numba-mpi** (komunikacja między węzłami superkomputerów):
  - inż: wsparcie dla komunikacji w "podgrupach" węzłów superkomputera
  - mgr: wsparcie dla komunikacji typów definiowanych przez użytkownika
- **PyMPDATA-MPI** (PyMPDATA @ Cyfronet):
  - inż: rozszerzenie na 3D (aktualnie przykłady jedynie dla 2D)
  - mgr: przykład rozwiązujący równania płytkiej wody na sferze
- **PyPartMC**: (interfejs Fortran $\rightsquigarrow$ C++ $\rightsquigarrow$ Python)
  - inż: przykład użycia w C++ (aktualnie: Python, Matlab & Julia)
  - mgr: porównanie PyPartMC vs. PySDM vs. rozwiązanie analityczne

Dziękuję za uwagę!

github.com/slayoo

sylwester.arabas@agh.edu.pl