libcloudph++: warm-rain cloud microphysics in C++

Sylwester Arabas

cloud-aerosol modelling team
Faculty of Physics, University of Warsaw

foss.igf.fuw.edu.pl

TU Delft, Jan. 15th 2015

Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

“HARMONIA” project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

"HARMONIA" project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

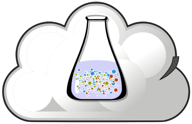
- CCN activation
- condensational growth

"HARMONIA" project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- CCN activation
- condensational growth



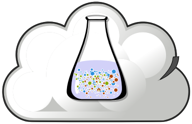
- collisional growth
- aqueous chemistry

“HARMONIA” project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- CCN activation
- condensational growth



- collisional growth
- aqueous chemistry

- precipitation
- wet deposition
- droplet deactivation

"HARMONIA" project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- novel cloud/aerosol microphysics models,

“HARMONIA” project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,

“HARMONIA” project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,
- modern coding techniques
  ⇝ priorities: researchers’ productivity & result reproducibility

"HARMONIA" project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,
- modern coding techniques
  $\rightsquigarrow$ priorities: researchers' productivity & result reproducibility

funding granted!

NATIONAL SCIENCE CENTRE
POLAND

- Title: **Aerosol processing by clouds - a multifaceted object-oriented numerical simulation framework**

“HARMONIA” project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,
- modern coding techniques
  ⇝ priorities: researchers’ productivity & result reproducibility

funding granted!

NATIONAL SCIENCE CENTRE
POLAND

- Title: **Aerosol processing by clouds - a multifaceted object-oriented numerical simulation framework**
- Duration: 3 years (till April 2016)

“HARMONIA” project: architects, goals & funding

Hanna Pawłowska, Piotr Smolarkiewicz & Wojtek Grabowski

develop a new LES tool for aerosol-cloud interactions research

- novel cloud/aerosol microphysics models,
- state-of-the-art numerical schemes,
- modern coding techniques
  ⇝ priorities: researchers’ productivity & result reproducibility

funding granted!

NATIONAL SCIENCE CENTRE
POLAND

- Title: **Aerosol processing by clouds - a multifaceted object-oriented numerical simulation framework**
- Duration: 3 years (till April 2016)
- Budget: $^1/_4$ M€

“HARMONIA” project: the team

the team (`http://foss.igf.fuw.edu.pl/`)

- prof. Hanna Pawłowska (group leader)

“HARMONIA” project: the team

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)

“HARMONIA” project: the team

the team (http://foss.igf.fuw.edu.pl/)

- prof. Hanna Pawłowska (group leader)
- in Warsaw:
  - Sylwester Arabas (postdoc)
  - Piotr Dziekan (postdoc)
  - Anna Jaruga (PhD student)
  - Maciej Waruszewski (PhD student)
  - Anna Zimniak (MSc student)
- overseas:
  - prof. Piotr Smolarkiewicz @ ECMWF
  - prof. Wojciech Grabowski @ NCAR
  - Dorota Jarecka @ NCAR (postdoc)

why develop new tools for aerosol/cloud μ-physics research

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)
- change of major element in the toolchain (e.g., programming language)

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)
- change of major element in the toolchain
  (e.g., programming language)
- reusing existing software not straightforward
  (legal, technological, maintenance issues)

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)
- change of major element in the toolchain (e.g., programming language)
- reusing existing software not straightforward (legal, technological, maintenance issues)
- new design principles

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)
- change of major element in the toolchain (e.g., programming language)
- reusing existing software not straightforward (legal, technological, maintenance issues)
- new design principles
  $\rightsquigarrow$ productivity-oriented software design (reusability, maintainability, reproducibility)

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)
- change of major element in the toolchain (e.g., programming language)
- reusing existing software not straightforward (legal, technological, maintenance issues)
- new design principles
  $\rightsquigarrow$ productivity-oriented software design
  (reusability, maintainability, reproducibility)
- know-how build-up

why develop new tools for aerosol/cloud μ-physics research

- new simulation paradigm (e.g., Lagrangian μ-physics)
- compatibility with novel hardware (e.g., GPUs)
- change of major element in the toolchain (e.g., programming language)
- reusing existing software not straightforward (legal, technological, maintenance issues)
- new design principles
  $\rightsquigarrow$ productivity-oriented software design
  (reusability, maintainability, reproducibility)
- know-how build-up

---

why not to develop everything from scratch:

- have to wait 3 years before tackling scientific problems

Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

libmpdata++ & libcloudph++

project target

LES-type tool featuring:

- robust numerics (MPDATA)
- particle-based aerosol/warm-rain μ-physics (super-droplet)

libmpdata++ & libcloudph++

project target

LES-type tool featuring:

- robust numerics (MPDATA)
- particle-based aerosol/warm-rain μ-physics (super-droplet)

current "products" – C++ libraries

libmpdata++ parallel solvers for systems of transport equations

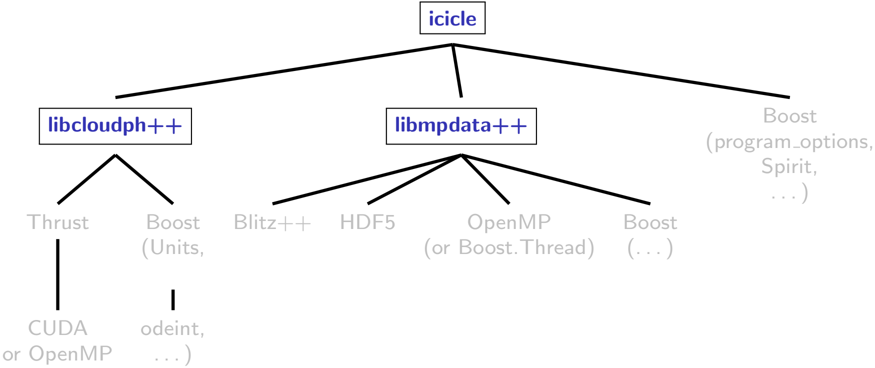
- `http://libmpdataxx.igf.fuw.edu.pl/`
- doi:10.5194/gmdd-7-8179-2014

libcloudph++ aerosol/cloud μ-physics algorithm collection

- `http://libcloudphxx.igf.fuw.edu.pl/`
- doi:10.5194/gmdd-7-8275-2014

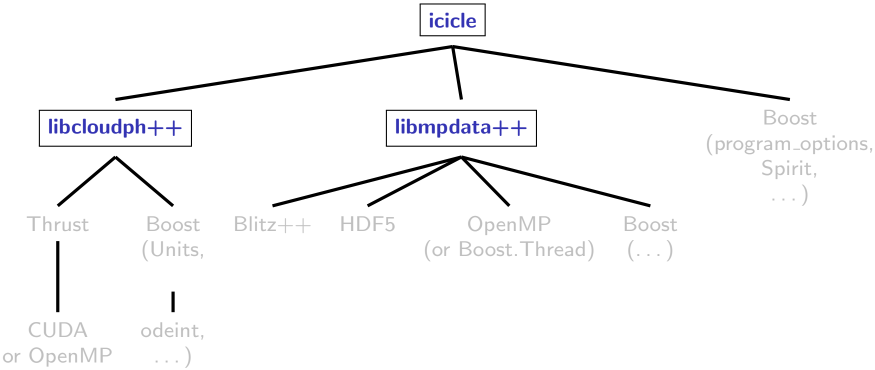
a few words on first design choices

- structure the code into “standalone” libraries



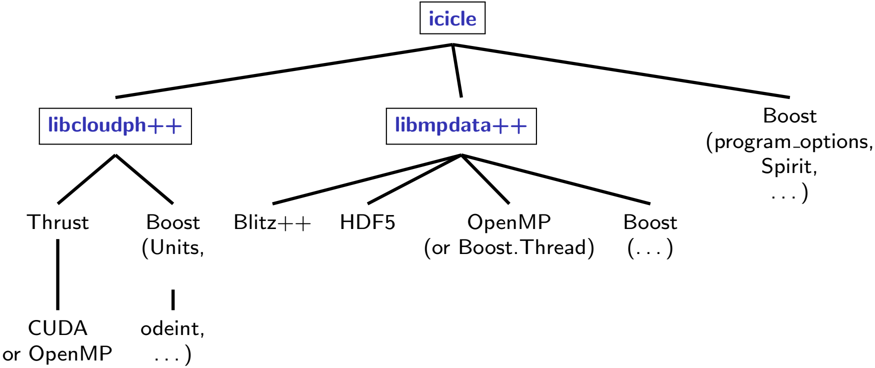
a few words on first design choices

- structure the code into "standalone" libraries
  ⇝ easier to document, to test and to contribute to
  ⇝ easier to be reused by others (in various contexts)



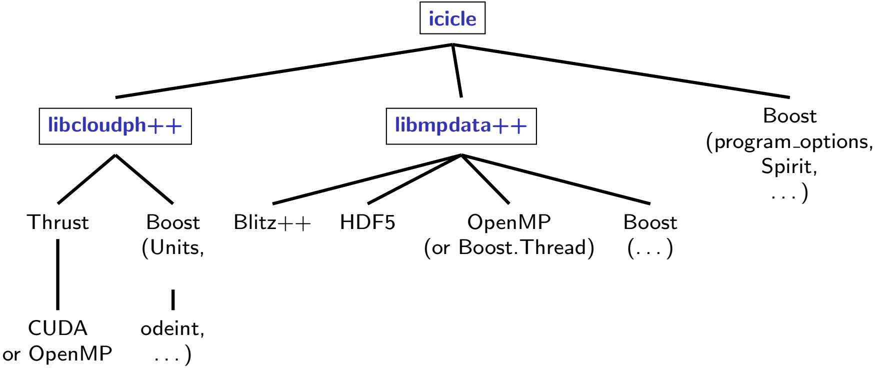
a few words on first design choices

- structure the code into “standalone” libraries
  ⇝ easier to document, to test and to contribute to
  ⇝ easier to be reused by others (in various contexts)
- leverage existing **reusable** software



a few words on first design choices

- structure the code into "standalone" libraries
  ⇝ easier to document, to test and to contribute to
  ⇝ easier to be reused by others (in various contexts)
- leverage existing **reusable** software
  ⇝ save time, benefit from state-of-the-art components



libcloudph++ components

- single-moment bulk saturation-adjustment scheme
  with Kessler's coalescence

- double-moment bulk scheme with predicted supersaturation
  (Morrison & Grabowski 2007)

- particle-based scheme with Monte-Carlo coalescence
  (super-droplet method of Shima et al. 2009)

- . . .

libcloudph++ components

- single-moment bulk saturation-adjustment scheme
  with Kessler's coalescence
- double-moment bulk scheme with predicted supersaturation
  (Morrison & Grabowski 2007)
- particle-based scheme with Monte-Carlo coalescence
  (super-droplet method of Shima et al. 2009)
- ...

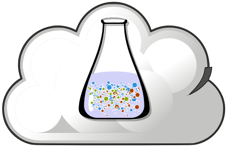
Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

Plan of the talk

- "HARMONIA" project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian "super-droplet" μ-physics
- libcloudph++ / DALES coupling

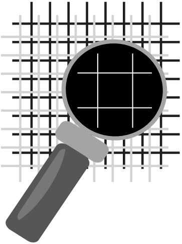
- CCN activation
- condensational growth



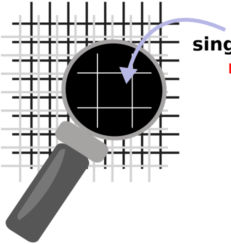
- collisional growth
- aqueous chemistry

- precipitation
- wet deposition
- droplet deactivation

Eulerian vs. Lagrangian μ-physics



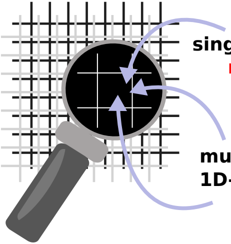
Eulerian vs. Lagrangian μ-physics



single-moment bulk

no influence of aerosol on cloud

Eulerian vs. Lagrangian μ-physics



single-moment bulk

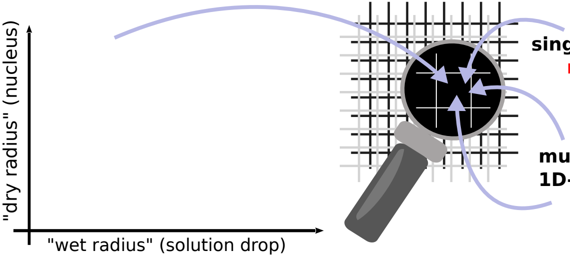
no influence of aerosol on cloud

multi-moment bulk, 1D-bin

aerosol may influence cloud

no "memory" of drop nuclei

Eulerian vs. Lagrangian μ-physics



single-moment bulk

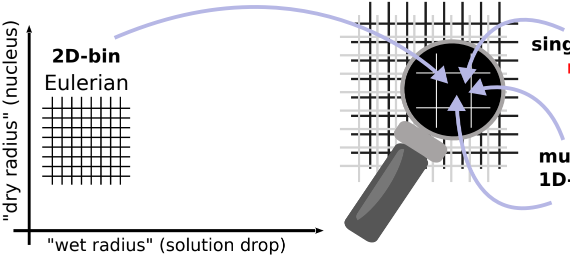
no influence of aerosol on cloud

multi-moment bulk, 1D-bin

aerosol may influence cloud

no "memory" of drop nuclei

Eulerian vs. Lagrangian μ-physics



2D-bin

Eulerian

single-moment bulk

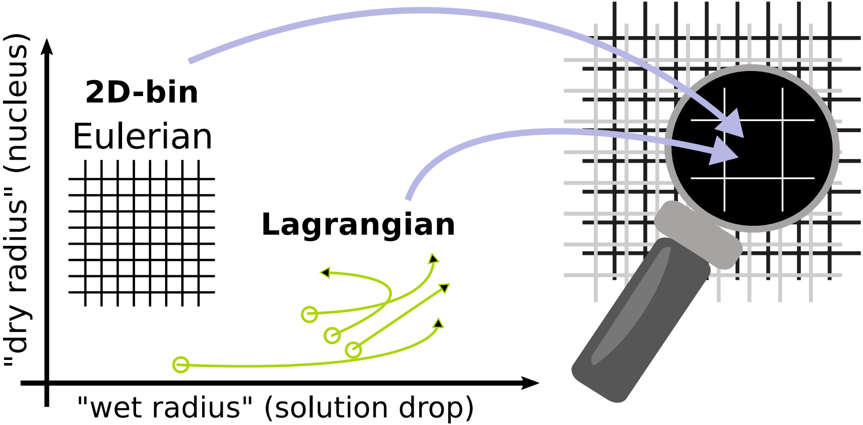
no influence of aerosol on cloud

multi-moment bulk,
1D-bin

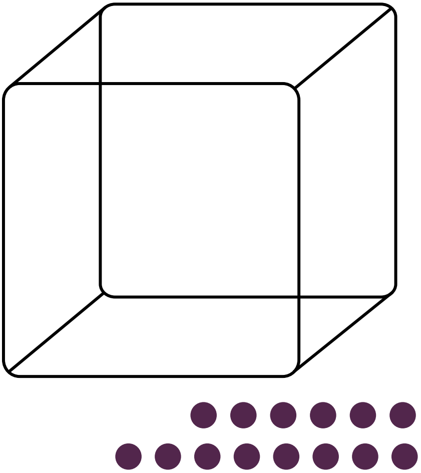
aerosol may influence cloud

no "memory" of drop nuclei

Eulerian vs. Lagrangian μ-physics

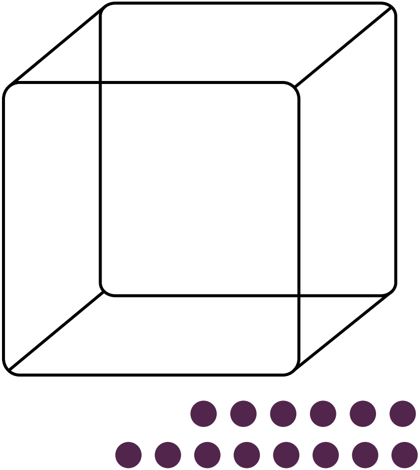


Lagrangian μ-physics: key concepts



The domain is populated with "information carriers" (alias computational particles, super droplets)

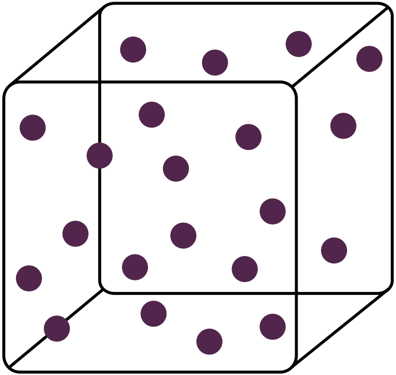
Lagrangian μ-physics: key concepts



The domain is populated with "information carriers" (alias computational particles, super droplets)

attributes:

Lagrangian μ-physics: key concepts

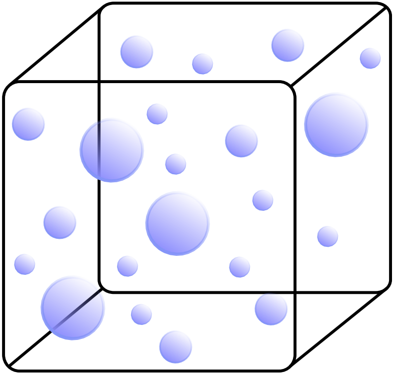


The domain is populated with “information carriers” (alias computational particles, super droplets)

attributes:

- spatial coords

Lagrangian μ-physics: key concepts

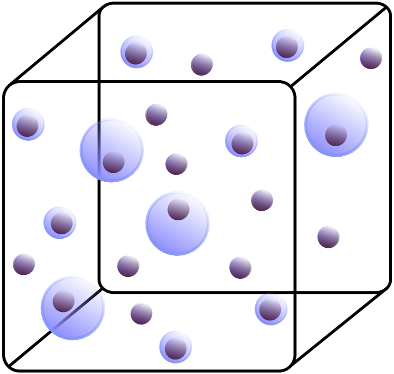


The domain is populated with "information carriers" (alias computational particles, super droplets)

attributes:

- spatial coords
- wet radius

Lagrangian μ-physics: key concepts

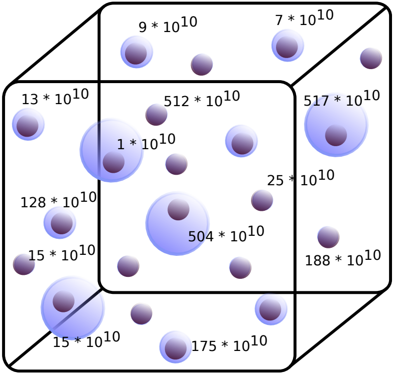


The domain is populated with "information carriers" (alias computational particles, super droplets)

attributes:

- spatial coords
- wet radius
- dry radius

Lagrangian μ-physics: key concepts

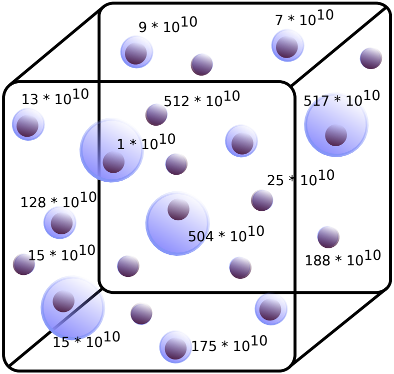


The domain is populated with "information carriers" (alias computational particles, super droplets)

attributes:

- spatial coords
- wet radius
- dry radius
- multiplicity

Lagrangian μ-physics: key concepts

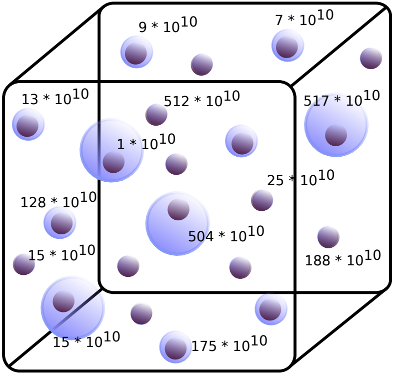


The domain is populated with "information carriers" (alias computational particles, super droplets)

attributes:

- spatial coords
- wet radius
- dry radius
- multiplicity
- ...

Lagrangian μ-physics: key concepts



The domain is populated with "information carriers" (alias computational particles, super droplets)

attributes:

- spatial coords
- wet radius
- dry radius
- multiplicity
- ...

transport does not incur numerical diffusion!

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) ⇝ many ”similar” real-world particles

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) $\rightsquigarrow$ many "similar" real-world particles
- attributes: multiplicity, dry radius, wet radius, nucleus type, ...

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) $\rightsquigarrow$ many "similar" real-world particles
- attributes: multiplicity, dry radius, wet radius, nucleus type, ...
- aerosol, cloud, precip. particles not distinguished, subject to same processes

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) $\rightsquigarrow$ many "similar" real-world particles
- attributes: multiplicity, dry radius, wet radius, nucleus type, . . .
- aerosol, cloud, precip. particles not distinguished, subject to same processes

Eulerian / PDE	Lagrangian / ODE
advection of heat	particle transport by the flow
advection of moisture	condensational growth
	collisional growth
	sedimentation

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) $\rightsquigarrow$ many "similar" real-world particles
- attributes: multiplicity, dry radius, wet radius, nucleus type, ...
- aerosol, cloud, precip. particles not distinguished, subject to same processes

Eulerian / PDE	Lagrangian / ODE
advection of heat	particle transport by the flow
advection of moisture	condensational growth
	collisional growth
	sedimentation
$\partial_t(\rho r) + \nabla(\vec{v}\rho r) = \rho\dot{r}$	$\dot{r} = \sum\limits_{\text{particles} \in \Delta V} \ldots$
$\partial_t(\rho\theta) + \nabla(\vec{v}\rho\theta) = \rho\dot{\theta}$	$\dot{\theta} = \sum\limits_{\text{particles} \in \Delta V} \ldots$

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) $\rightsquigarrow$ many ”similar” real-world particles
- attributes: multiplicity, dry radius, wet radius, nucleus type, ...
- aerosol, cloud, precip. particles not distinguished, subject to same processes

Eulerian / PDE	Lagrangian / ODE
advection of heat	particle transport by the flow
advection of moisture	condensational growth
	collisional growth
	sedimentation
$\partial_t(\rho r) + \nabla(\vec{v}\rho r) = \rho\dot{r}$	$\dot{r} = \sum\limits_{\text{particles} \in \Delta V} \ldots$
$\partial_t(\rho\theta) + \nabla(\vec{v}\rho\theta) = \rho\dot{\theta}$	$\dot{\theta} = \sum\limits_{\text{particles} \in \Delta V} \ldots$
advection of trace gases	in-particle aqueous chemistry
...	...

Lagrangian μ-physics: key concepts

- each particle (aka super-droplet) $\rightsquigarrow$ many "similar" real-world particles
- attributes: multiplicity, dry radius, wet radius, nucleus type, ...
- aerosol, cloud, precip. particles not distinguished, subject to same processes

Eulerian / PDE	Lagrangian / ODE
advection of heat	particle transport by the flow
advection of moisture	condensational growth
	collisional growth
	sedimentation
$\partial_t(\rho r) + \nabla(\vec{v}\rho r) = \rho\dot{r}$	$\dot{r} = \sum\limits_{\text{particles} \in \Delta V} \ldots$
$\partial_t(\rho\theta) + \nabla(\vec{v}\rho\theta) = \rho\dot{\theta}$	$\dot{\theta} = \sum\limits_{\text{particles} \in \Delta V} \ldots$
advection of trace gases	in-particle aqueous chemistry
...	...

- recent examples in context of precipitating clouds:
  - Shima et al. 2009, QJ
  - Andrejczuk et al. 2010, JGR
  - Riechelmann et al. 2012, NJP

Monte-Carlo coalescence scheme (Shima et al. 2009)

- for all n super-droplets in a grid box of volume ΔV in timestep Δt

Monte-Carlo coalescence scheme (Shima et al. 2009)

- for all n super-droplets in a grid box of volume ΔV in timestep Δt
- each representing ξ real particles (aerosol/cloud/drizzle/rain)

Monte-Carlo coalescence scheme (Shima et al. 2009)

- for all n super-droplets in a grid box of volume ΔV in timestep Δt
- each representing ξ real particles (aerosol/cloud/drizzle/rain)
- the probability of coalescence of i-th and j-th super-droplets is:

$$P_{ij} = max(\xi_i, \xi_j) \cdot \underbrace{E(r_i, r_j) \cdot \pi(r_i + r_j)^2 \cdot |v_i - v_j|}_{\text{coalescence kernel}} \cdot \frac{\Delta t}{\Delta V}$$

where r – drop radii, $E(r_i, r_j)$ – collection efficiency, v – drop velocities

Monte-Carlo coalescence scheme (Shima et al. 2009)

- for all n super-droplets in a grid box of volume ΔV in timestep Δt
- each representing ξ real particles (aerosol/cloud/drizzle/rain)
- the probability of coalescence of i-th and j-th super-droplets is:

$$P_{ij} = max(\xi_i, \xi_j) \cdot \underbrace{E(r_i, r_j) \cdot \pi(r_i + r_j)^2 \cdot |v_i - v_j|}_{\text{coalescence kernel}} \cdot \frac{\Delta t}{\Delta V}$$

where r – drop radii, $E(r_i, r_j)$ – collection efficiency, v – drop velocities

- coalescence takes place once in a number of timesteps (def. by P_{ij})
- all min(ξ_i,ξ_j) droplets coalesce
  $\rightsquigarrow$ there's always a "bin" of the right size to store the collided particles

Monte-Carlo coalescence scheme (Shima et al. 2009)

- for all n super-droplets in a grid box of volume ΔV in timestep Δt
- each representing ξ real particles (aerosol/cloud/drizzle/rain)
- the probability of coalescence of i-th and j-th super-droplets is:

$$P_{ij} = max(\xi_i, \xi_j) \cdot \underbrace{E(r_i, r_j) \cdot \pi (r_i + r_j)^2 \cdot |v_i - v_j|}_{\text{coalescence kernel}} \cdot \frac{\Delta t}{\Delta V}$$

  where r – drop radii, $E(r_i, r_j)$ – collection efficiency, v – drop velocities
- coalescence takes place once in a number of timesteps (def. by P_{ij})
- all min(ξ_i,ξ_j) droplets coalesce
  $\rightsquigarrow$ there's always a "bin" of the right size to store the collided particles
- collisions triggered by comparing a uniform random number with P_{ij}

Monte-Carlo coalescence scheme (Shima et al. 2009)

- for all n super-droplets in a grid box of volume ΔV in timestep Δt
- each representing ξ real particles (aerosol/cloud/drizzle/rain)
- the probability of coalescence of i-th and j-th super-droplets is:

$$P_{ij} = max(\xi_i, \xi_j) \cdot \underbrace{E(r_i, r_j) \cdot \pi(r_i + r_j)^2 \cdot |v_i - v_j|}_{\text{coalescence kernel}} \cdot \frac{\Delta t}{\Delta V}$$

  where r – drop radii, $E(r_i, r_j)$ – collection efficiency, v – drop velocities

- coalescence takes place once in a number of timesteps (def. by P_{ij})
- all min(ξ_i,ξ_j) droplets coalesce
  $\rightsquigarrow$ there's always a "bin" of the right size to store the collided particles
- collisions triggered by comparing a uniform random number with P_{ij}
- extensive parameters summed ($\rightsquigarrow$ conserved), intensive averaged

Monte-Carlo coalescence scheme (Shima et al. 2009)

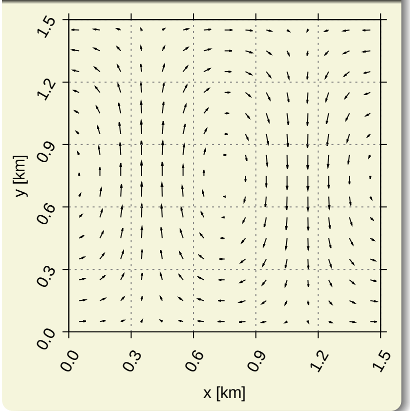
- for all n super-droplets in a grid box of volume ΔV in timestep Δt
- each representing ξ real particles (aerosol/cloud/drizzle/rain)
- the probability of coalescence of i-th and j-th super-droplets is:

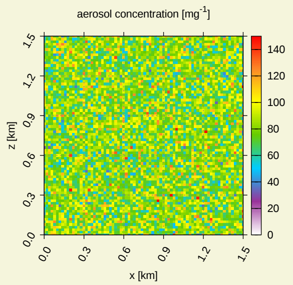
$$P_{ij} = max(\xi_i, \xi_j) \cdot \underbrace{E(r_i, r_j) \cdot \pi(r_i + r_j)^2 \cdot |v_i - v_j|}_{\text{coalescence kernel}} \cdot \frac{\Delta t}{\Delta V} \cdot \frac{n \cdot (n-1)}{2} / \left[\frac{n}{2}\right]$$

  where r – drop radii, $E(r_i, r_j)$ – collection efficiency, v – drop velocities
- coalescence takes place once in a number of timesteps (def. by P_{ij})
- all min(ξ_i,ξ_j) droplets coalesce
  $\rightsquigarrow$ there's always a "bin" of the right size to store the collided particles
- collisions triggered by comparing a uniform random number with P_{ij}
- extensive parameters summed ($\rightsquigarrow$ conserved), intensive averaged
- [n/2] random non-overlapping (i,j) pairs examined instead of all (i,j) pairs
  cost: $O(n^2) \rightsquigarrow O(n)$, probability upscaled by $\frac{n \cdot (n-1)}{2} / \left[\frac{n}{2}\right]$

libcloudph++: VOCALS-inspired aerosol processing set-up

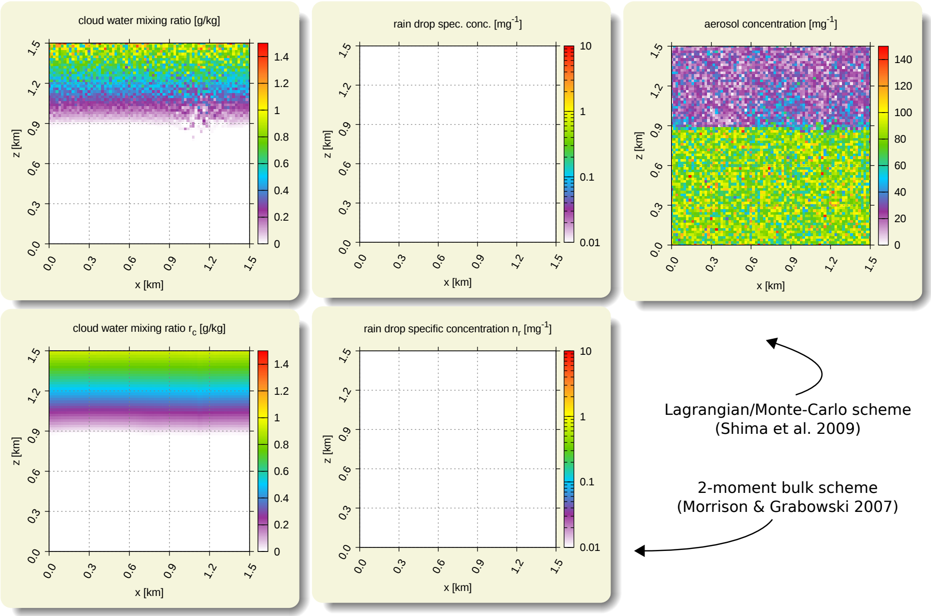
single-eddy velocity field





- set-up: Grabowski & Lebo (ICMW 2012)
- 2D prescribed flow
- advection: libmpdata++ (2-pass FCT)
- μ-physics: libcloudph++

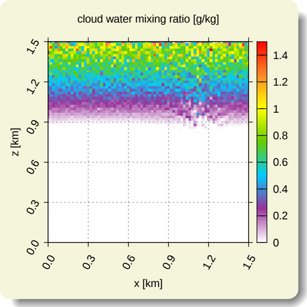
libcloudph++: VOCALS-inspired aerosol processing set-up

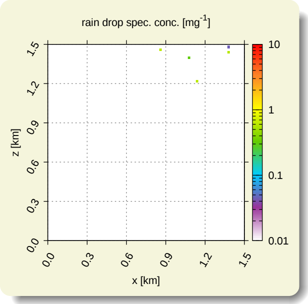


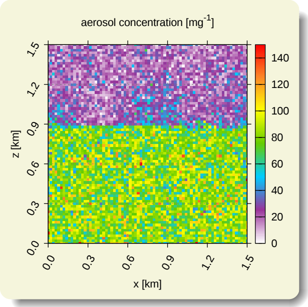
Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

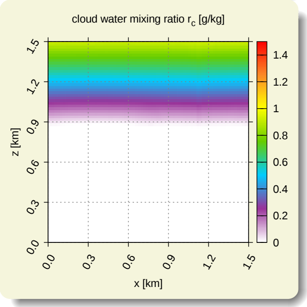
2-moment bulk scheme
(Morrison & Grabowski 2007)

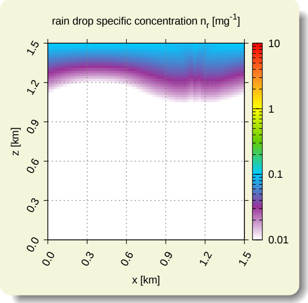
libcloudph++: VOCALS-inspired aerosol processing set-up







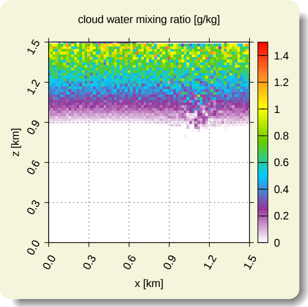


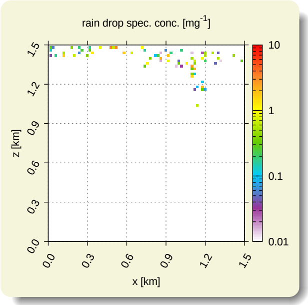


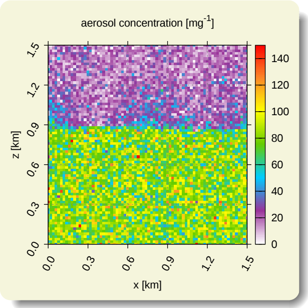
Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

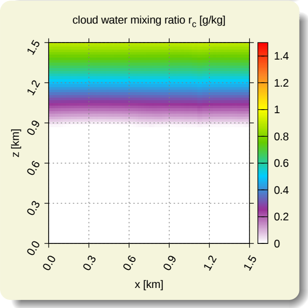
2-moment bulk scheme
(Morrison & Grabowski 2007)

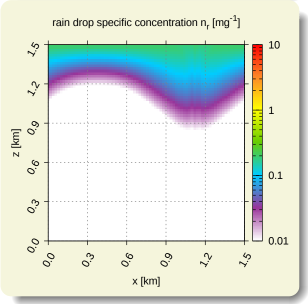
libcloudph++: VOCALS-inspired aerosol processing set-up







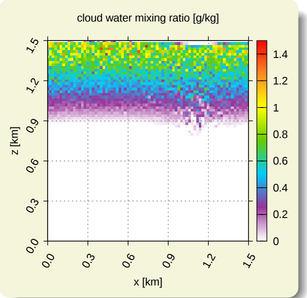


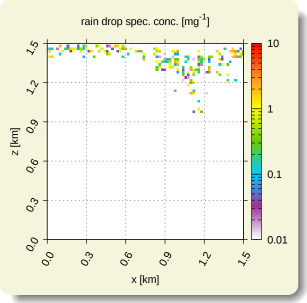


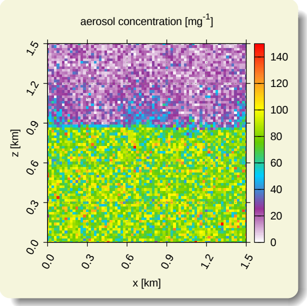
Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

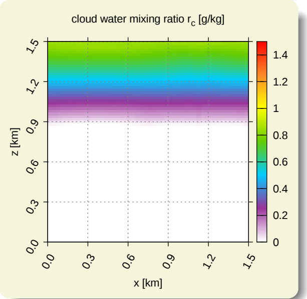
2-moment bulk scheme
(Morrison & Grabowski 2007)

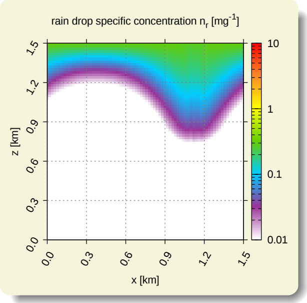
libcloudph++: VOCALS-inspired aerosol processing set-up







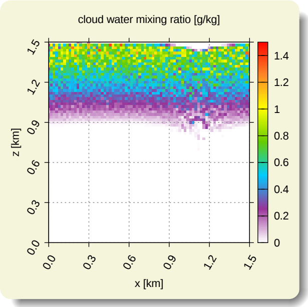


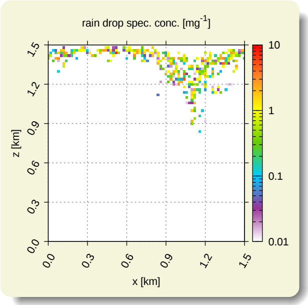


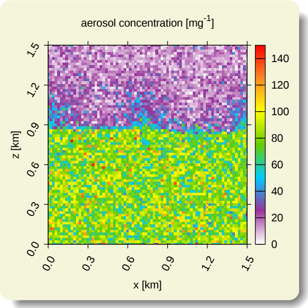
Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

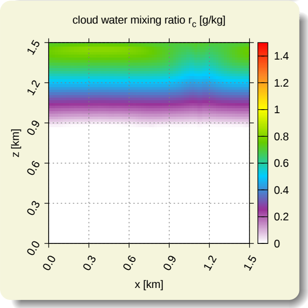
2-moment bulk scheme
(Morrison & Grabowski 2007)

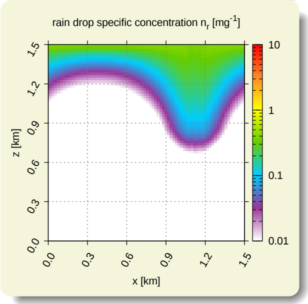
libcloudph++: VOCALS-inspired aerosol processing set-up


cloud water mixing ratio [g/kg]


rain drop spec. conc. [mg^{-1}]


aerosol concentration [mg^{-1}]

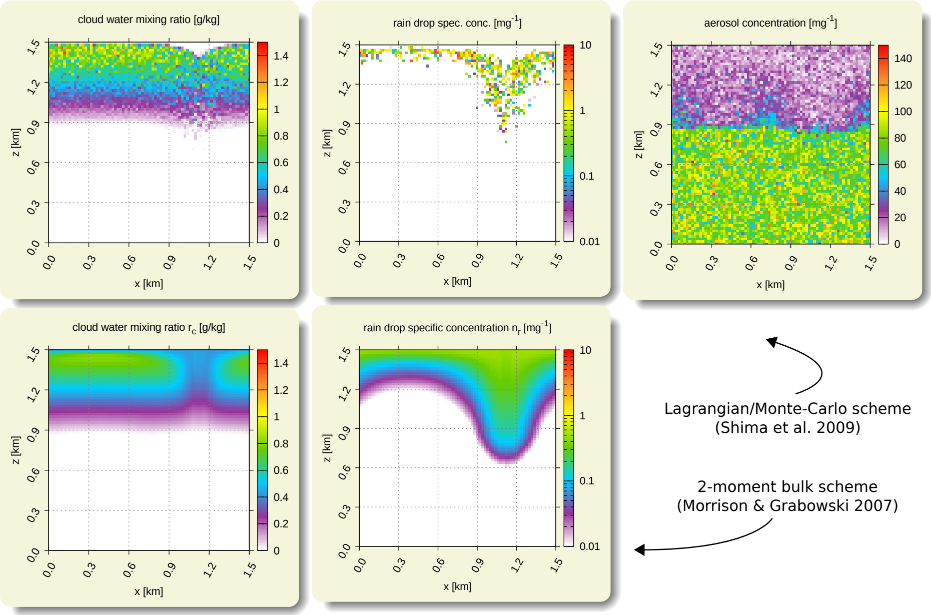

cloud water mixing ratio r_c [g/kg]


rain drop specific concentration n_r [mg^{-1}]

Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

2-moment bulk scheme
(Morrison & Grabowski 2007)

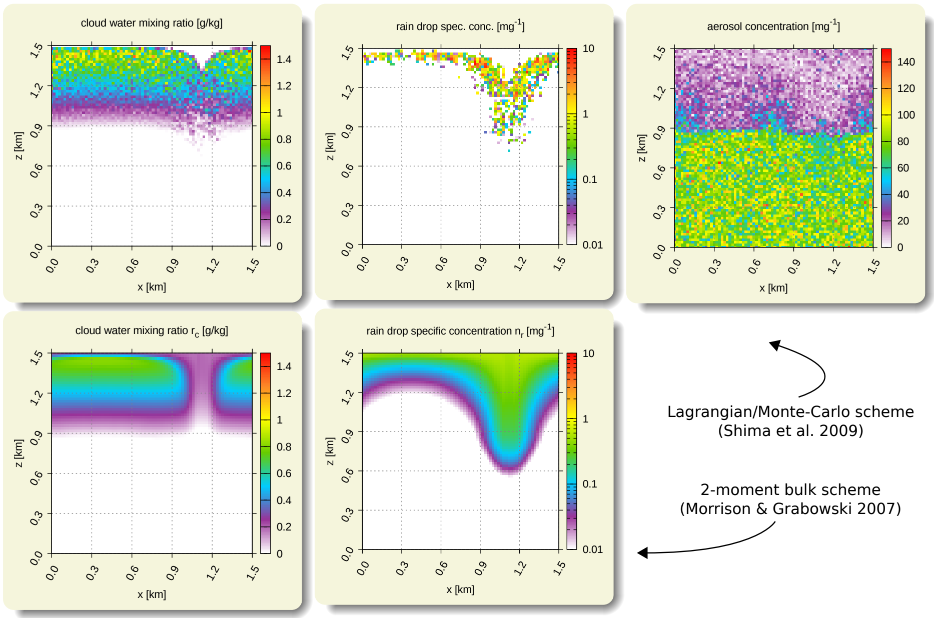
libcloudph++: VOCALS-inspired aerosol processing set-up



Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

2-moment bulk scheme
(Morrison & Grabowski 2007)

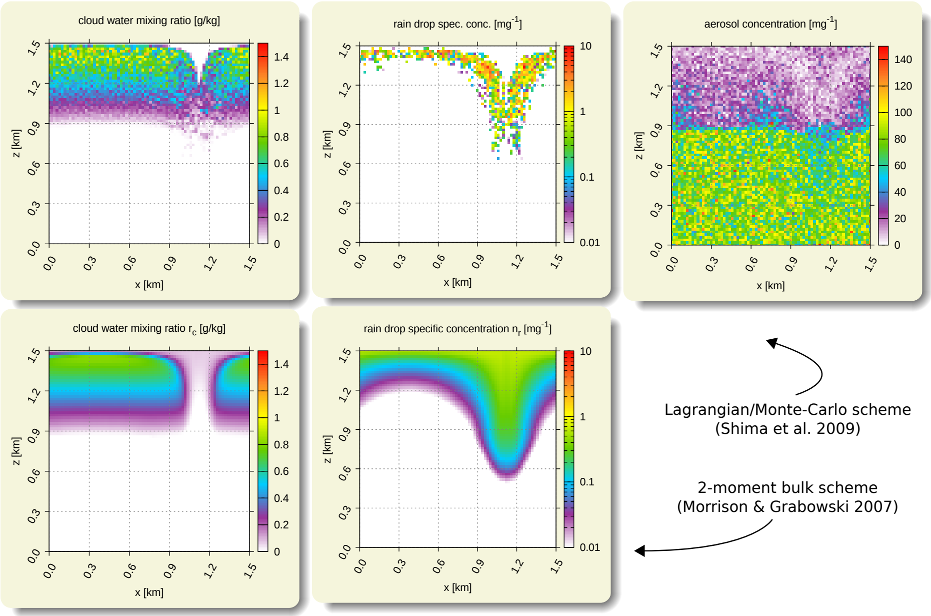
libcloudph++: VOCALS-inspired aerosol processing set-up



Lagrangian/Monte-Carlo scheme (Shima et al. 2009)

2-moment bulk scheme (Morrison & Grabowski 2007)

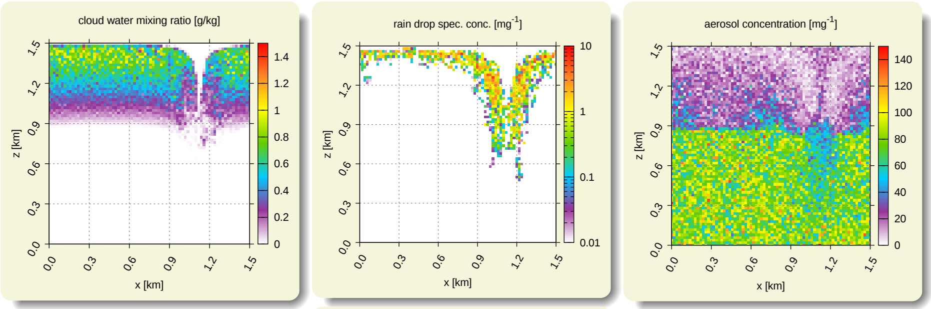
libcloudph++: VOCALS-inspired aerosol processing set-up

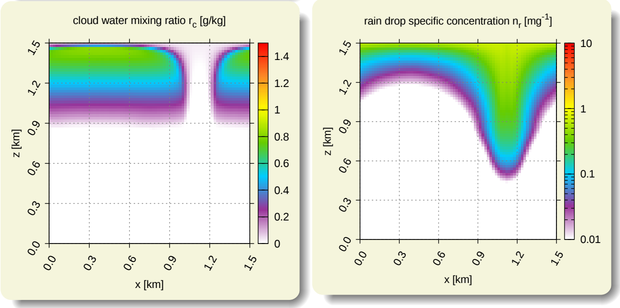


Lagrangian/Monte-Carlo scheme (Shima et al. 2009)

2-moment bulk scheme (Morrison & Grabowski 2007)

libcloudph++: VOCALS-inspired aerosol processing set-up

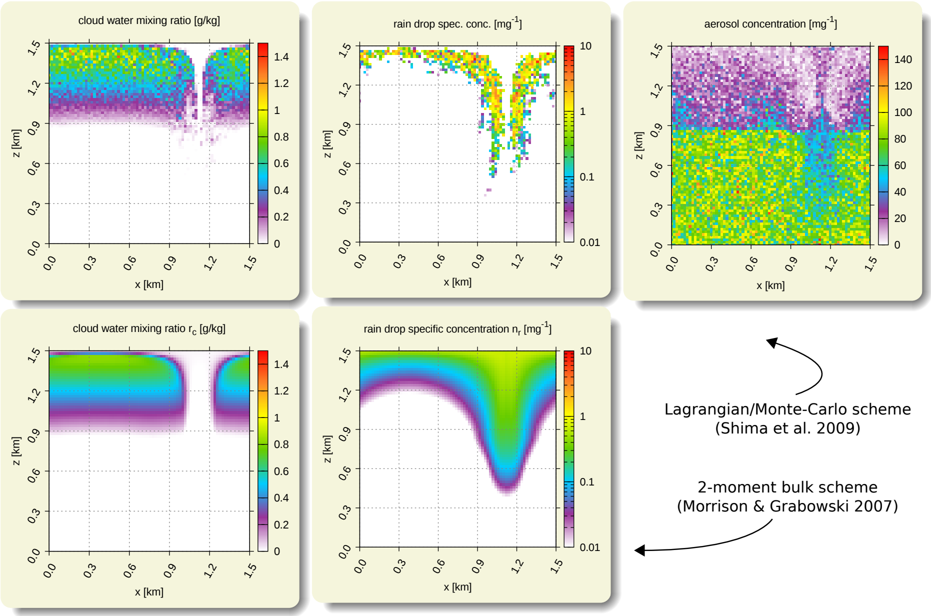




Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

2-moment bulk scheme
(Morrison & Grabowski 2007)

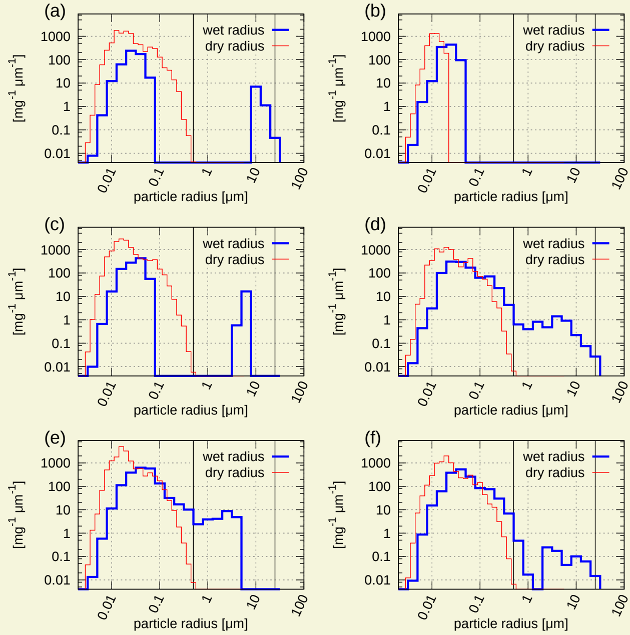
libcloudph++: VOCALS-inspired aerosol processing set-up

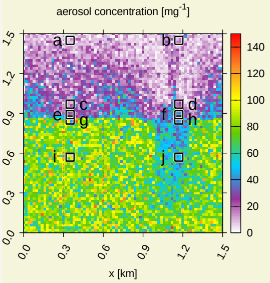


Lagrangian/Monte-Carlo scheme
(Shima et al. 2009)

2-moment bulk scheme
(Morrison & Grabowski 2007)

2×2 cell particle-derived spectra





libcloudph++: summary & some technicalities

key features:

- three schemes (all written from scratch):
    - 1-moment: Kessler
    - 2-moment: Morrison & Grabowski 2008
    - Lagrangian: Shima et al. 2009 (Monte-Carlo coalescence)

libcloudph++: summary & some technicalities

key features:

- three schemes (all written from scratch):
  - 1-moment: Kessler
  - 2-moment: Morrison & Grabowski 2008
  - Lagrangian: Shima et al. 2009 (Monte-Carlo coalescence)
- Lagrangian scheme optionally GPU-resident (via Thrust)

libcloudph++: summary & some technicalities

key features:

- three schemes (all written from scratch):
  - 1-moment: Kessler
  - 2-moment: Morrison & Grabowski 2008
  - Lagrangian: Shima et al. 2009 (Monte-Carlo coalescence)
- Lagrangian scheme optionally GPU-resident (via Thrust)
- compact code (500 / 1000 / 4500 LOC)

libcloudph++: summary & some technicalities

key features:

- three schemes (all written from scratch):
  - 1-moment: Kessler
  - 2-moment: Morrison & Grabowski 2008
  - Lagrangian: Shima et al. 2009 (Monte-Carlo coalescence)
- Lagrangian scheme optionally GPU-resident (via Thrust)
- compact code (500 / 1000 / 4500 LOC)
- written using Boost.units – compile-time dimensional analysis

libcloudph++: summary & some technicalities

key features:

- three schemes (all written from scratch):
    - 1-moment: Kessler
    - 2-moment: Morrison & Grabowski 2008
    - Lagrangian: Shima et al. 2009 (Monte-Carlo coalescence)
- Lagrangian scheme optionally GPU-resident (via Thrust)
- compact code (500 / 1000 / 4500 LOC)
- written using Boost.units – compile-time dimensional analysis
- reusable:
    - design: no assumptions on dimensionality or dyn-core type
    - documentation: API described in the paper/manual
    - legal/practical matters: open source, GPL, hosted on github

Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

Plan of the talk

- “HARMONIA” project: goals and the team
- libcloudph++: design choices and their rationale
- libcloudph++: Lagrangian “super-droplet” μ-physics
- libcloudph++ / DALES coupling

DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++

DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++
- 2014.05: funding application submitted (fnp.org.pl)

DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++
- 2014.05: funding application submitted (fnp.org.pl)
- 2014.08: travel funds granted! (4 meetings within one year)

DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++
- 2014.05: funding application submitted (fnp.org.pl)
- 2014.08: travel funds granted! (4 meetings within one year)
- 2014.09: a kick-off meeting, plans:
  - 1-st stage: off-line simulation
  - 2-nd stage: fully-coupled set-up
  - ...

DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++
- 2014.05: funding application submitted (fnp.org.pl)
- 2014.08: travel funds granted! (4 meetings within one year)
- 2014.09: a kick-off meeting, plans:
  - 1-st stage: off-line simulation
  - 2-nd stage: fully-coupled set-up
  - ...
- 2014.12: most technical issues solved

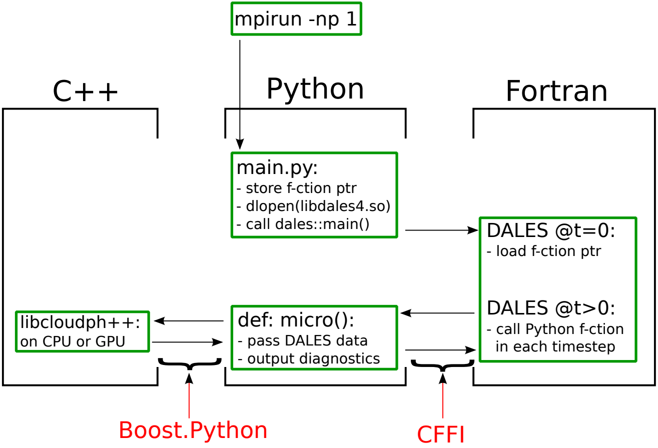
DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++
- 2014.05: funding application submitted (fnp.org.pl)
- 2014.08: travel funds granted! (4 meetings within one year)
- 2014.09: a kick-off meeting, plans:
  - 1-st stage: off-line simulation
  - 2-nd stage: fully-coupled set-up
  - ...
- 2014.12: most technical issues solved
- 2015.01: this visit at TUDelft:
  - first successful off-line simulations
  - getting it running on "thor" @ TUDelft

DALES/libcloudph++ coupling

- 2014.04: let's try to couple DALES with libcloudph++
- 2014.05: funding application submitted (fnp.org.pl)
- 2014.08: travel funds granted! (4 meetings within one year)
- 2014.09: a kick-off meeting, plans:
  - 1-st stage: off-line simulation
  - 2-nd stage: fully-coupled set-up
  - ...
- 2014.12: most technical issues solved
- 2015.01: this visit at TUDelft:
  - first successful off-line simulations
  - getting it running on “thor” @ TUDelft
- ...

DALES/libcloudph++ coupling: technicalities



mpirun -np 1

C++

Python

Fortran

main.py:
- store f-ction ptr
- dlopen(libdales4.so)
- call dales::main()

DALES @t=0:
- load f-ction ptr

DALES @t>0:
- call Python f-ction in each timestep

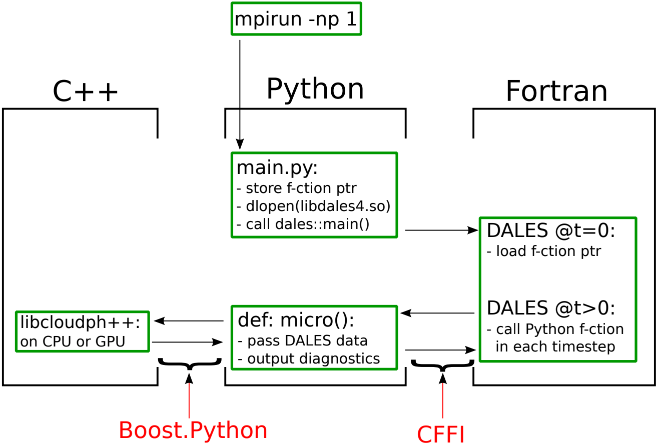
libcloudph++: on CPU or GPU

def: micro():
- pass DALES data
- output diagnostics

Boost.Python

CFFI

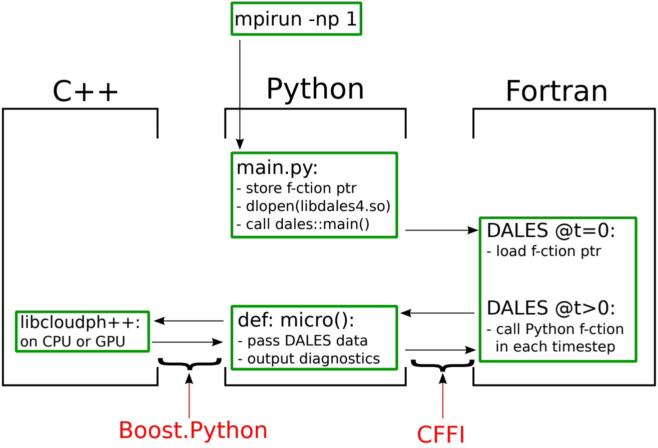
DALES/libcloudph++ coupling: technicalities



rationale:

- minimal changes to DALES
- no changes to libcloudph++
- interfacing C++ & Fortran from Python using pre-existing packages

DALES/libcloudph++ coupling: technicalities



acknowledgement:

- the Python-Fortran coupling concept & code by Dorota Jarecka

DALES/libcloudph++ coupling: "physics"

libcloudph++/Lagrangian:

- $q_v(t; x, y, z)$
- $\theta_d(t; x, y, z) = \left({}^{p_0}/_{p_d}\right)^{R_d/c_{pd}}$
- $\rho_d(z)$ or $\rho_d(t; z) = \frac{p_d}{R_d T}$
- $\rho\vec{u}(t; x, y, z)$
- supersaturation allowed ($q_v \neq q_s$)

DALES/bulk:

- $q_t(t; x, y, z) = q_v + q_l$
- $\theta_l(t; x, y, z) = \theta - q_l \frac{L}{c_{pd}}$
- $\rho_b(z)$ & $\rho(t; z)/\pi(t; z)$
- $\vec{u}(t; x, y, z)$
- q_c diagnosed assuming $q_v = q_s$

DALES/libcloudph++ coupling: “physics”

libcloudph++/Lagrangian:

- $q_v(t;x,y,z)$
- $\theta_d(t;x,y,z) = \left({}^{p_0}\!/_{p_d}\right)^{R_d/c_{pd}}$
- $\rho_d(z)$ or $\rho_d(t;z) = \frac{p_d}{R_d\,T}$
- $\rho\vec{u}(t;x,y,z)$
- supersaturation allowed ($q_v \neq q_s$)

DALES/bulk:

- $q_t(t;x,y,z) = q_v + q_l$
- $\theta_l(t;x,y,z) = \theta - q_l\,\frac{L}{c_{pd}}$
- $\rho_b(z)$ & $\rho(t;z)/\pi(t;z)$
- $\vec{u}(t;x,y,z)$
- q_c diagnosed assuming $q_v = q_s$

- $\rho_d(z) = \rho_d\left(\pi\big|_{t=0}\,,\ \bar{q}_t\big|_{t=0}\,,\ \bar{\theta}_l\big|_{t=0}\right)$
- $q_v = q_t - \overbrace{\sum_{i\in cell} \frac{{}^4\!/_3\ \pi\ n_i\ r_i^3 \rho_w}{\Delta v\ \ \rho_d}}^{q_l}$ ⇝ no $q_v = q_s$ assumption
- $\theta_d = \underbrace{\left(\theta_l + \frac{q_t - q_v}{\pi}\,\frac{L}{c_{pd}}\right)}_{\theta} \cdot \left(1 + q_v\,\frac{R_v}{R_v}\right)^{R_d/c_{pd}}$
- $\rho\vec{u} = \rho_b\vec{u}$ ⇝ consistent with heat/moisture advection

DALES/libcloudph++ coupling: “physics”

libcloudph++/Lagrangian:

- $q_v(t; x, y, z)$
- $\theta_d(t; x, y, z) = \left({}^{p_0}\!/_{p_d}\right)^{R_d/c_{pd}}$
- $\rho_d(z)$ or $\rho_d(t; z) = \frac{p_d}{R_d T}$
- $\rho\vec{u}(t; x, y, z)$
- supersaturation allowed $(q_v \neq q_s)$

DALES/bulk:

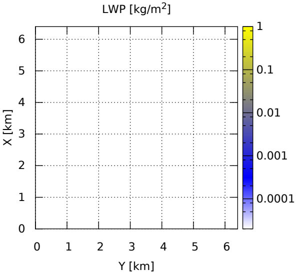
- $q_t(t; x, y, z) = q_v + q_l$
- $\theta_l(t; x, y, z) = \theta - q_l \frac{L}{c_{pd}}$
- $\rho_b(z)$ & $\rho(t; z)/\pi(t; z)$
- $\vec{u}(t; x, y, z)$
- q_c diagnosed assuming $q_v = q_s$

- $\rho_d(z) = \rho_d\left(\pi|_{t=0}\,, \bar{q}_t|_{t=0}\,, \bar{\theta}_l|_{t=0}\right)$
- $q_v = q_t - \overbrace{\sum_{i \in cell} \frac{{}^4\!/_3\ \pi\ n_i\ r_i^3 \rho_w}{\Delta v\ \ \rho_d}}^{q_l}$ ⇝ no $q_v = q_s$ assumption
- $\theta_d = \underbrace{\left(\theta_l + \frac{q_t - q_v}{\pi}\frac{L}{c_{pd}}\right)}_{\theta} \cdot \left(1 + q_v \frac{R_v}{R_v}\right)^{R_d/c_{pd}}$
- $\rho\vec{u} = \rho_b\vec{u}$ ⇝ consistent with heat/moisture advection

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

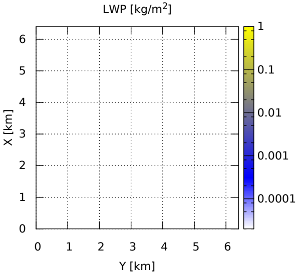
DALES (BOMEX, bulk μ-physics, t=17m)

LWP [kg/m^2]

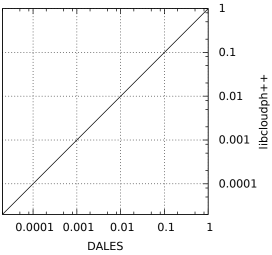


off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

LWP [kg/m^2]

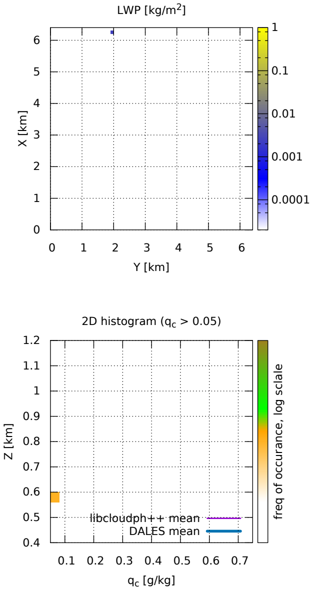


LWP [kg/m^2] (scatter plot)

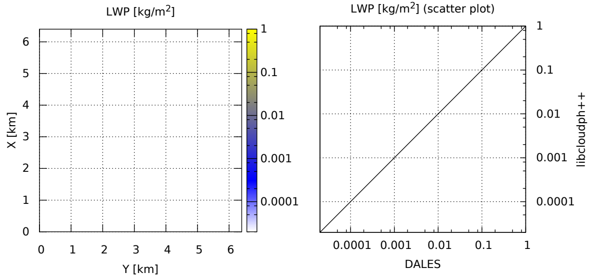


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=18m)

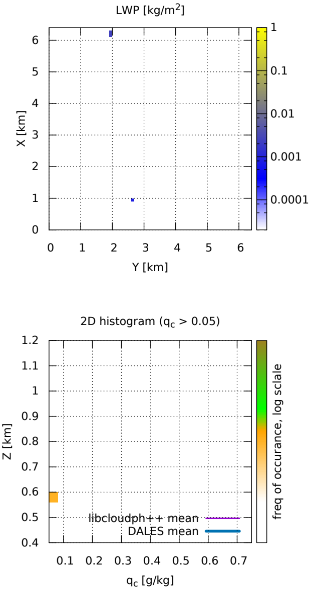


off-line libclouph++ Lagrangian/Monte-Carlo μ-physics on a GPU

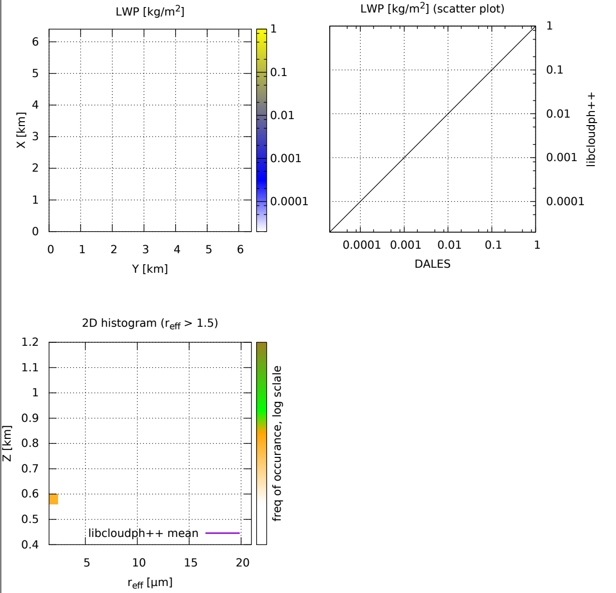


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=19m)

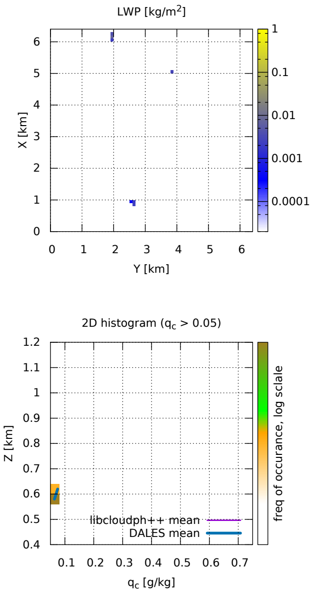


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

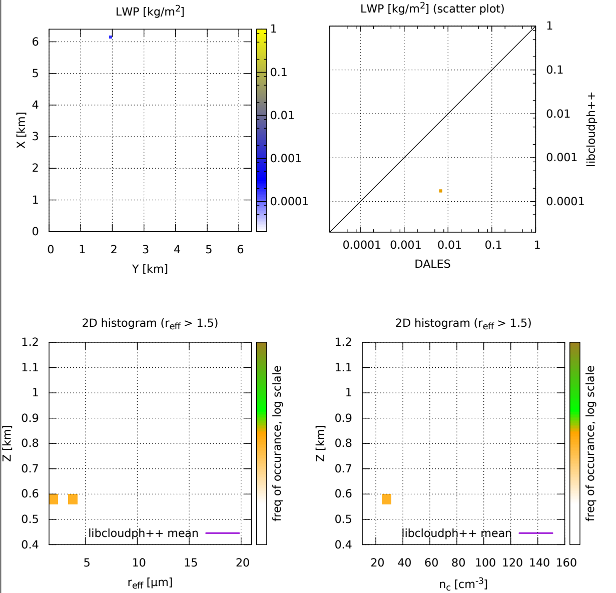


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=20m)

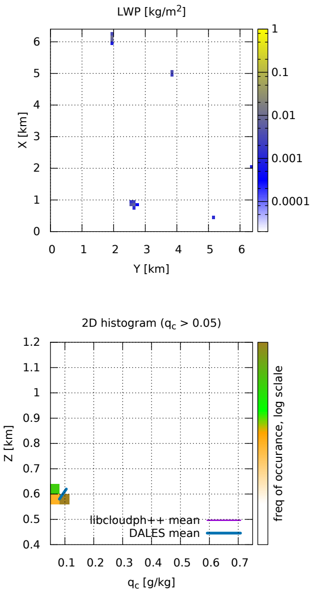


off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

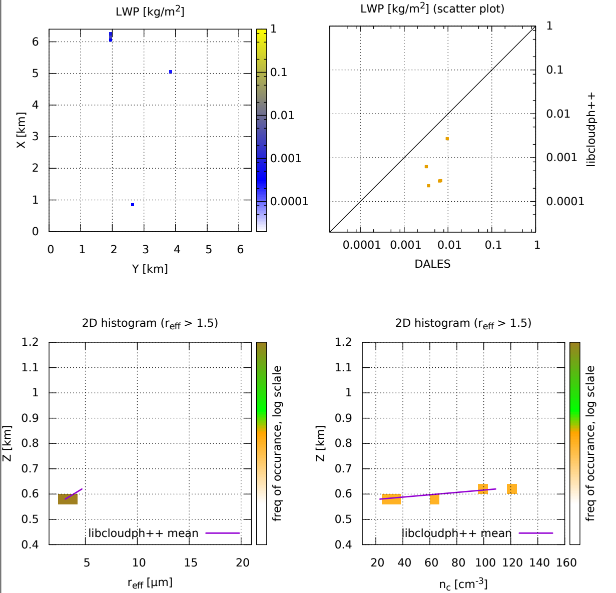


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=21m)

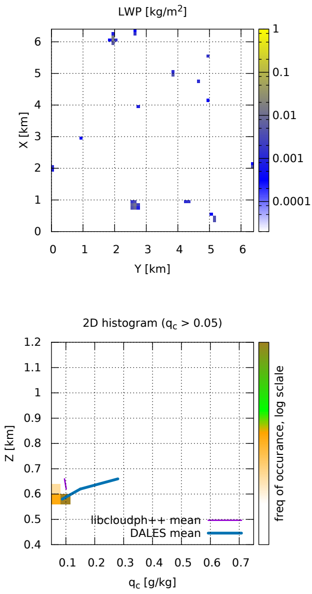


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

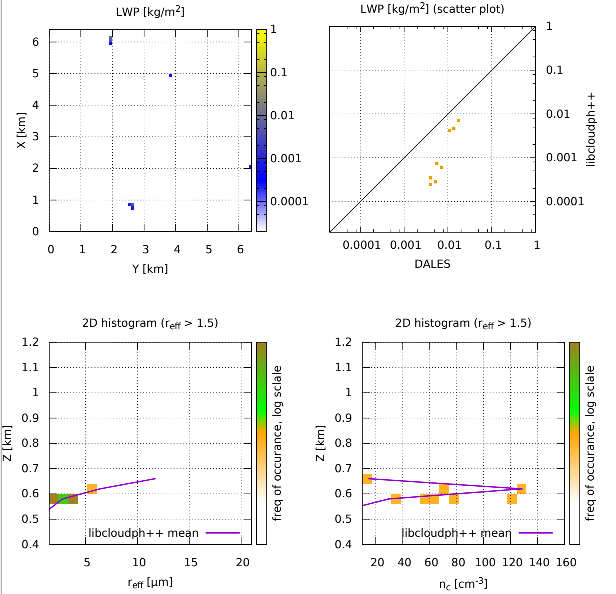


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=22m)

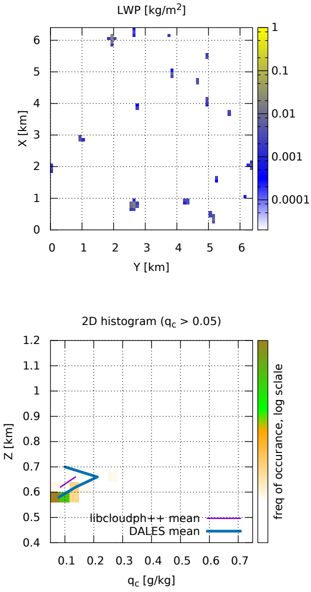


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

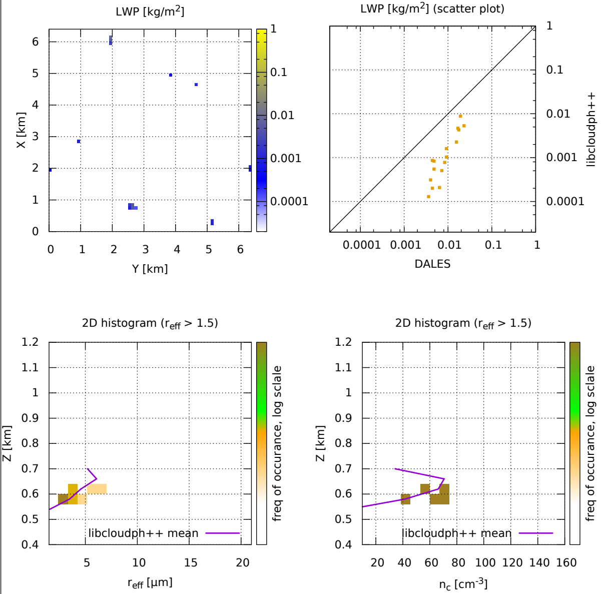


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=23m)

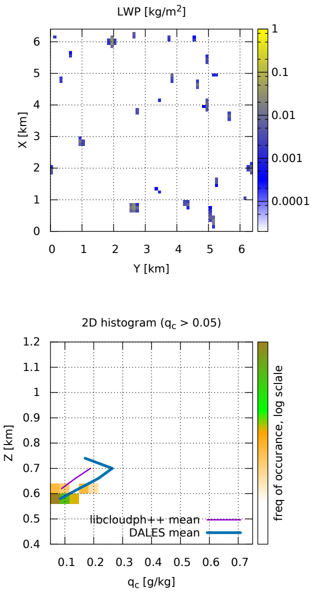


off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

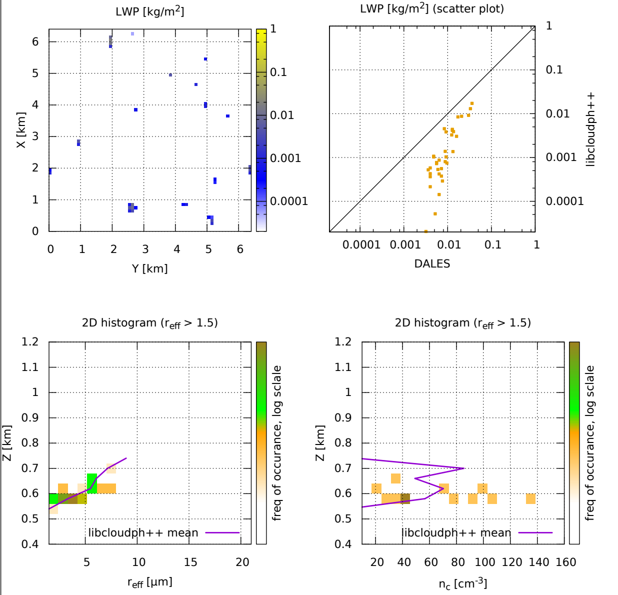


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=24m)

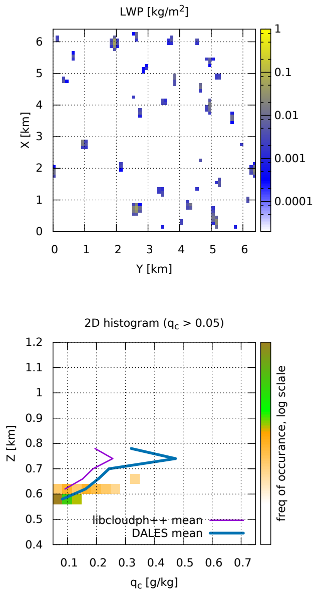


off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

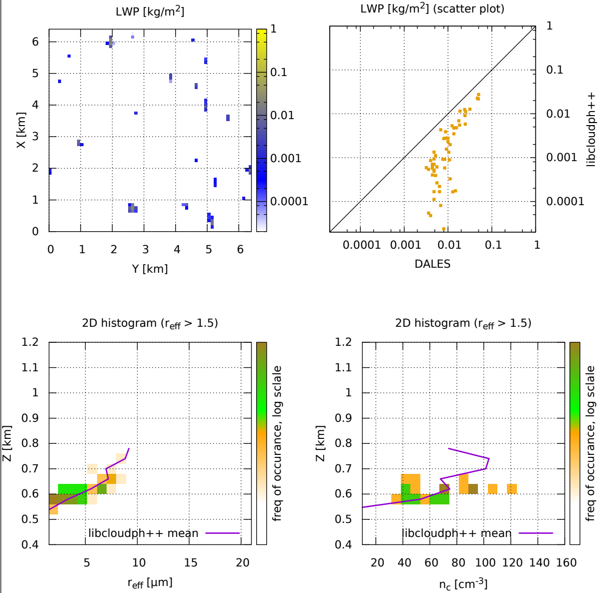


DALES/libcloudph++: proof-of-concept test ($\mathrm{dt}_{\mathrm{adv}}$=20s, $\mathrm{dt}_{\mathrm{cond}}$=0.5s, dz=40m, 10 SD/cell)

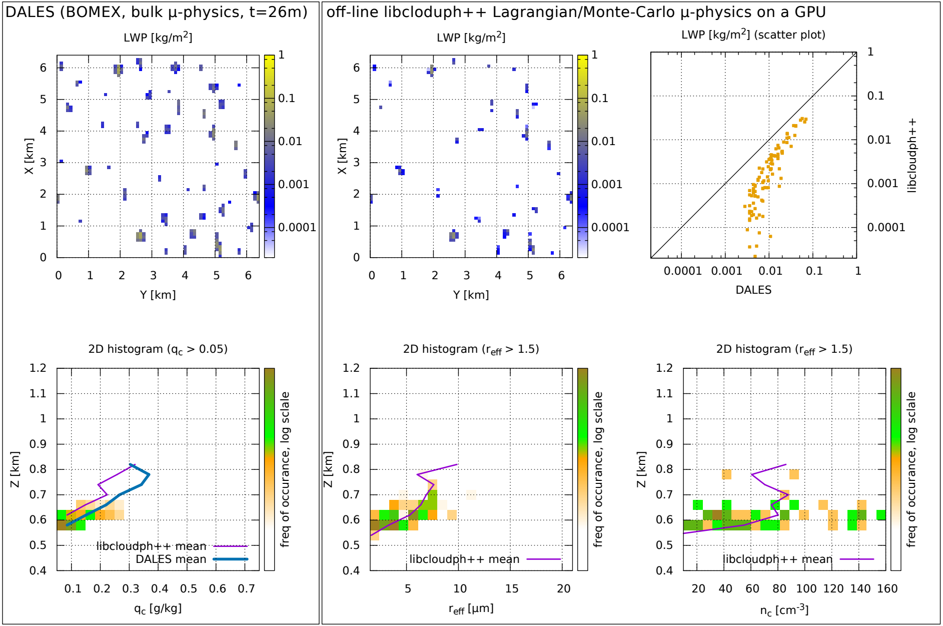
DALES (BOMEX, bulk μ-physics, t=25m)



off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU



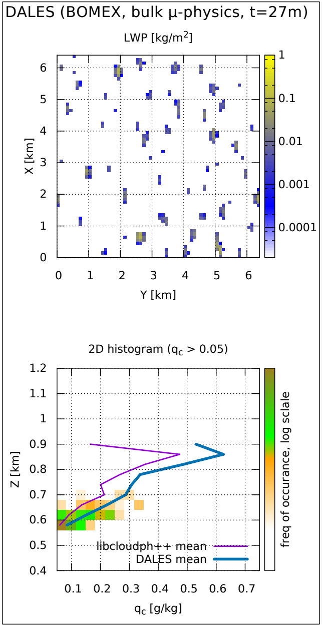
DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)



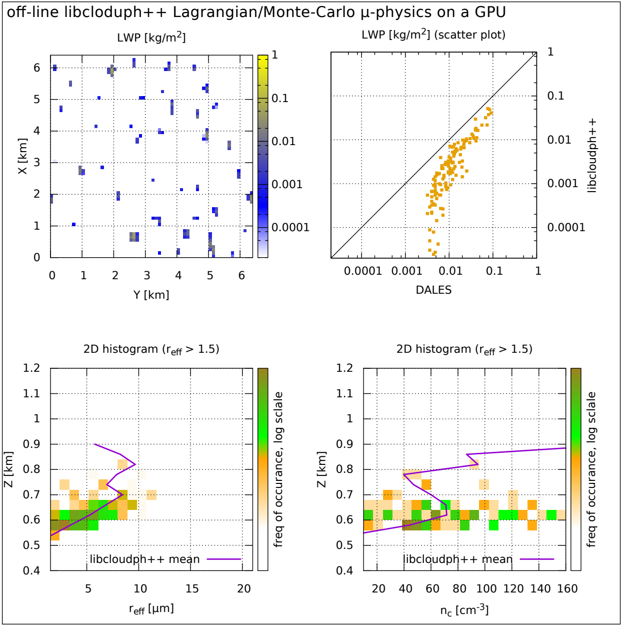
DALES (BOMEX, bulk μ-physics, t=26m)

off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

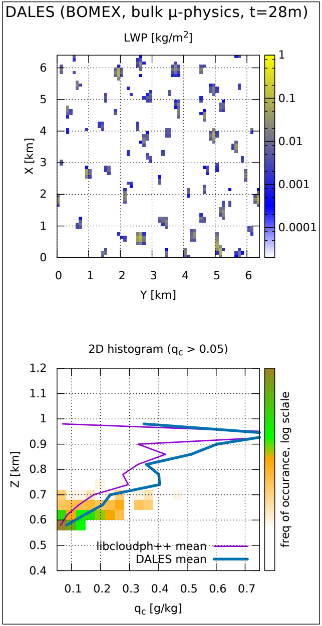


DALES (BOMEX, bulk μ-physics, t=27m)

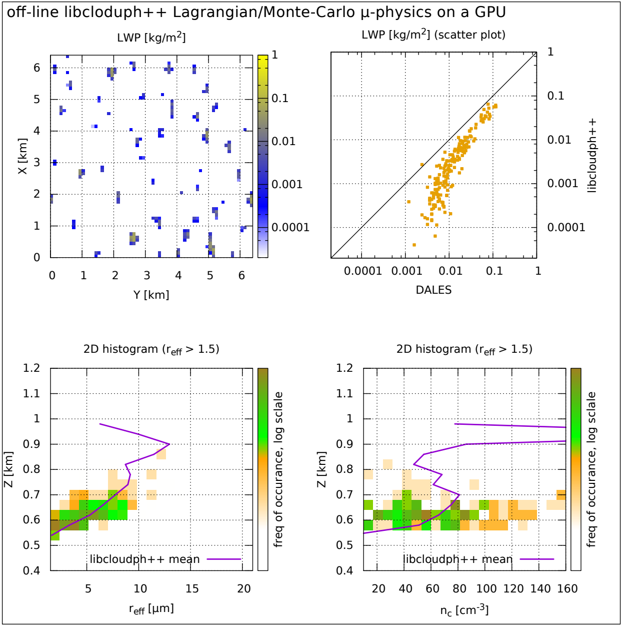


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)



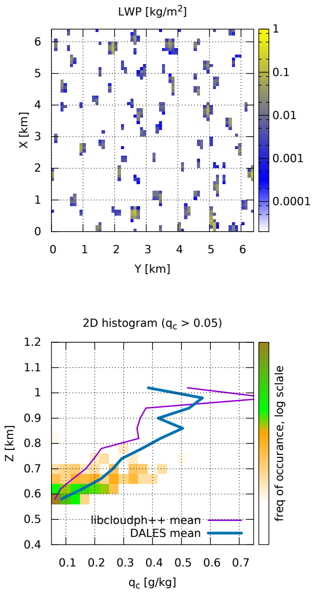
DALES (BOMEX, bulk μ-physics, t=28m)



off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

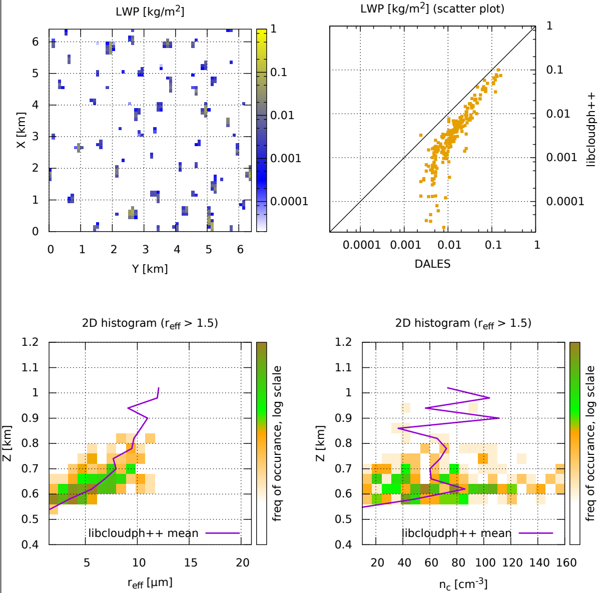
DALES (BOMEX, bulk μ-physics, t=29m)



LWP [kg/m^2]

2D histogram (q_c > 0.05)

off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU



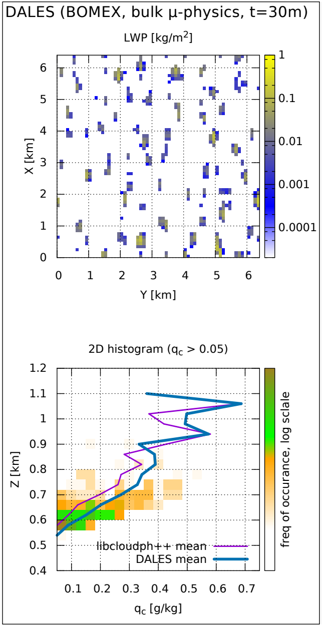
LWP [kg/m^2]

LWP [kg/m^2] (scatter plot)

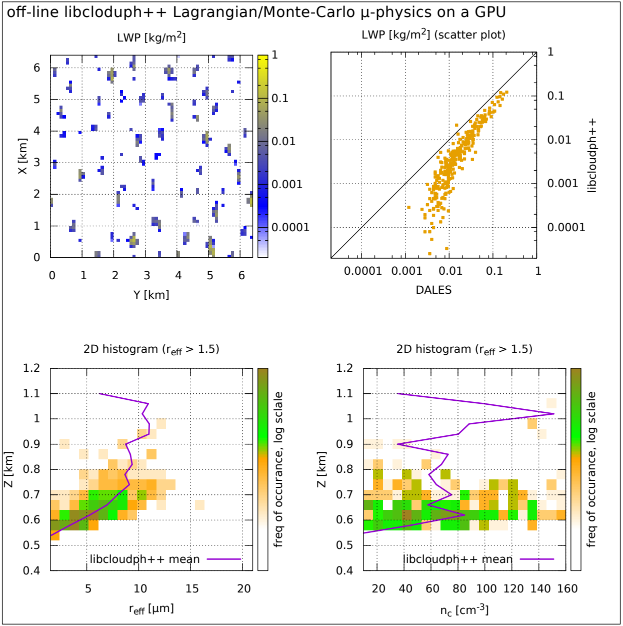
2D histogram (r_{eff} > 1.5)

2D histogram (r_{eff} > 1.5)

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)



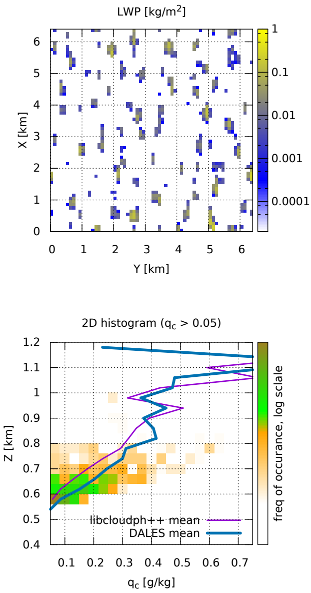
DALES (BOMEX, bulk μ-physics, t=30m)



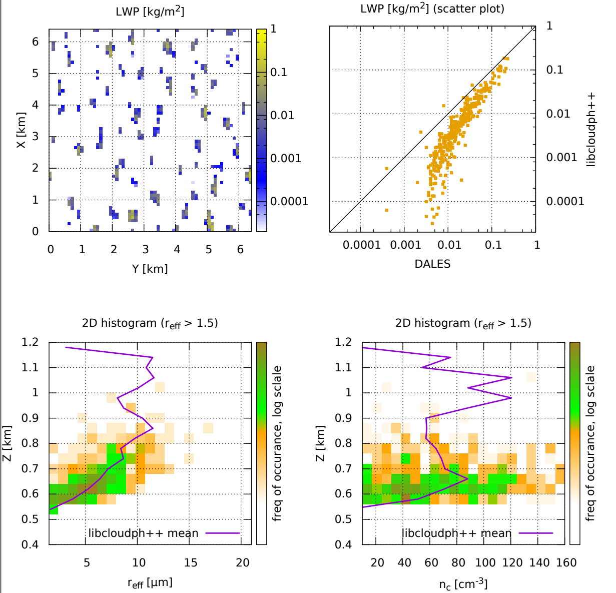
off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

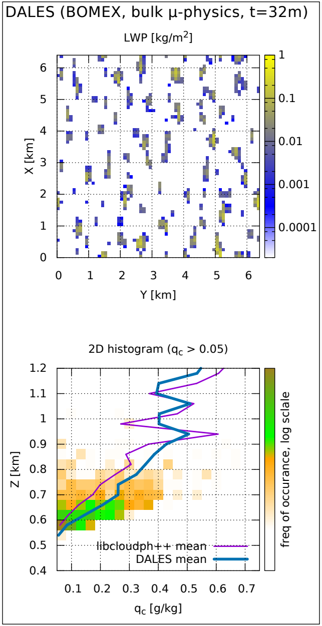
DALES (BOMEX, bulk μ-physics, t=31m)



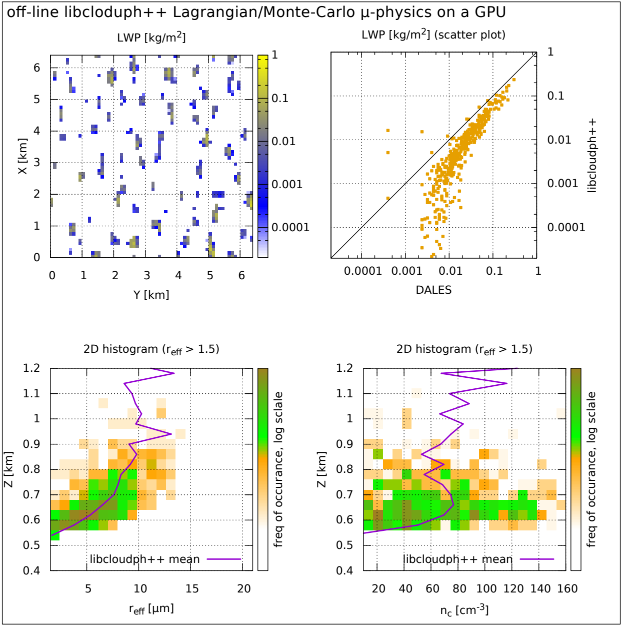
off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU



DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)



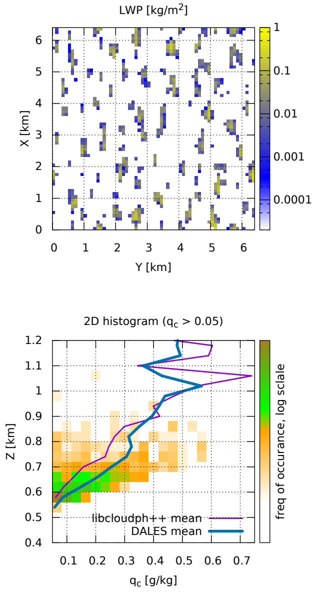
DALES (BOMEX, bulk μ-physics, t=32m)



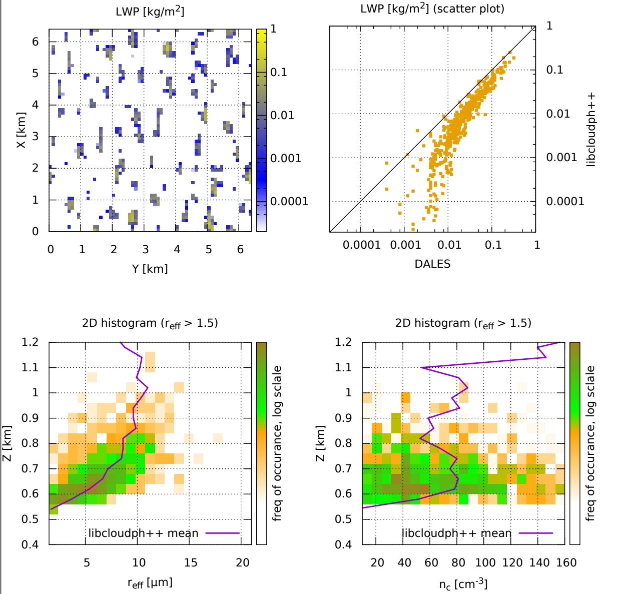
off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

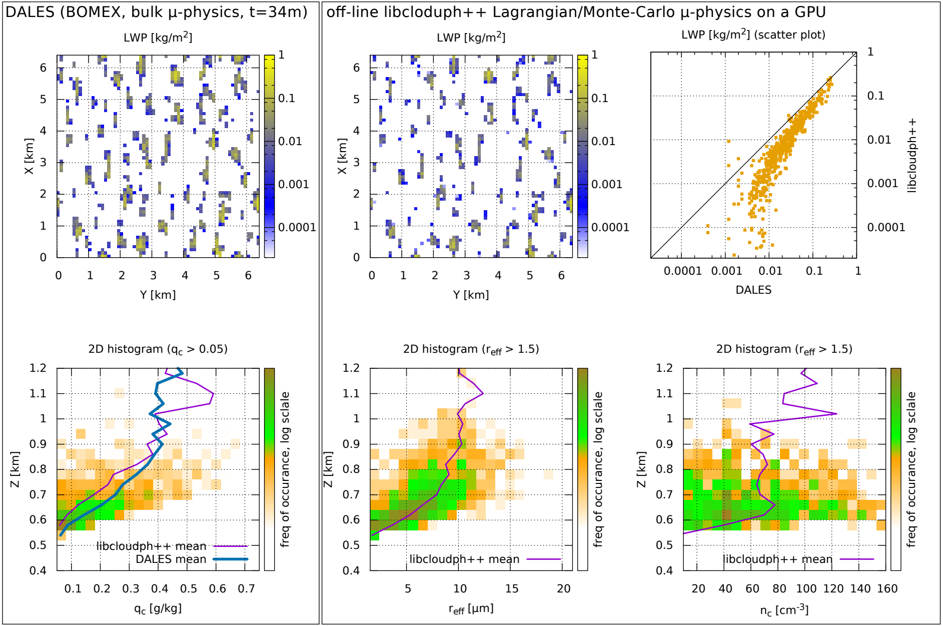
DALES (BOMEX, bulk μ-physics, t=33m)



off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU



DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

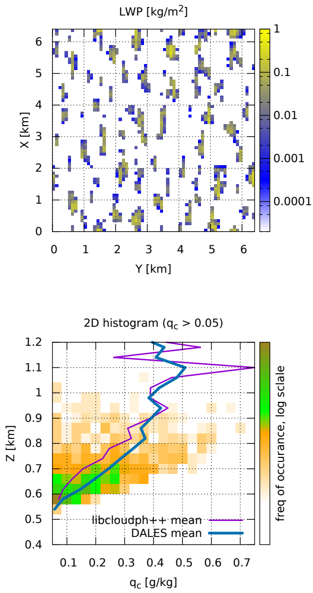


DALES (BOMEX, bulk μ-physics, t=34m)

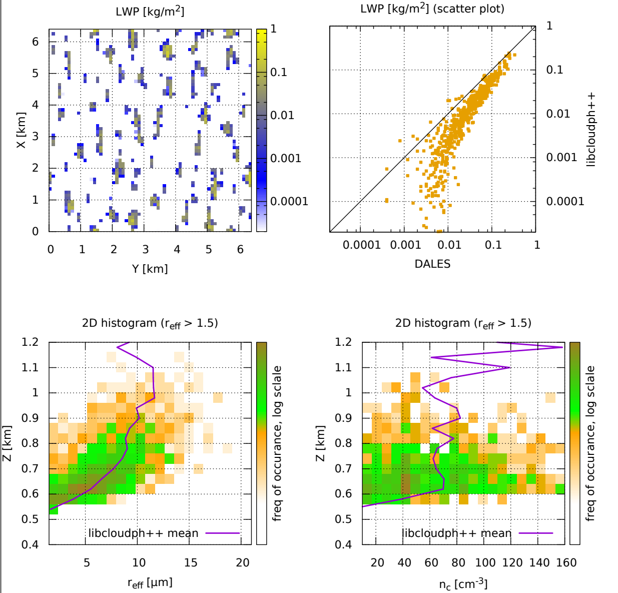
off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

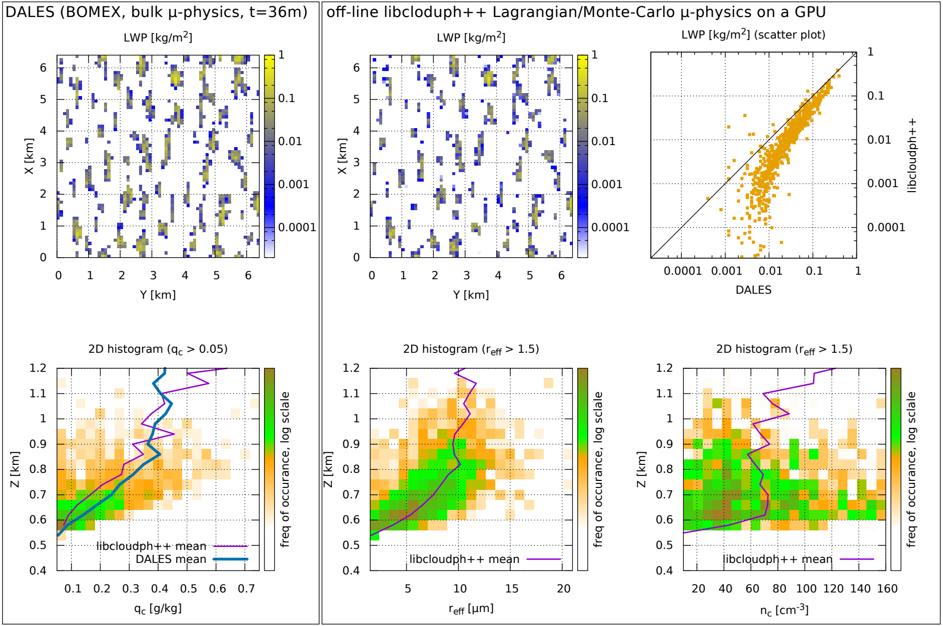
DALES (BOMEX, bulk μ-physics, t=35m)



off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU



DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)



DALES (BOMEX, bulk μ-physics, t=36m)

off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

LWP [kg/m^2]

LWP [kg/m^2]

LWP [kg/m^2] (scatter plot)

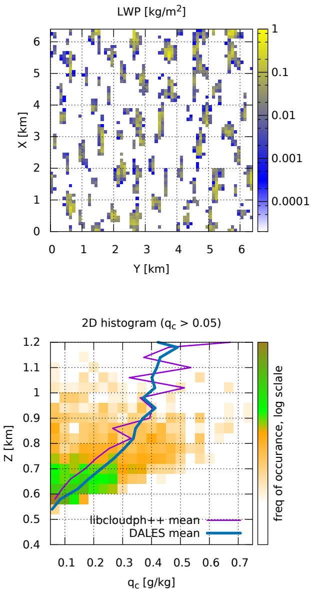
2D histogram (q_c > 0.05)

2D histogram (r_{eff} > 1.5)

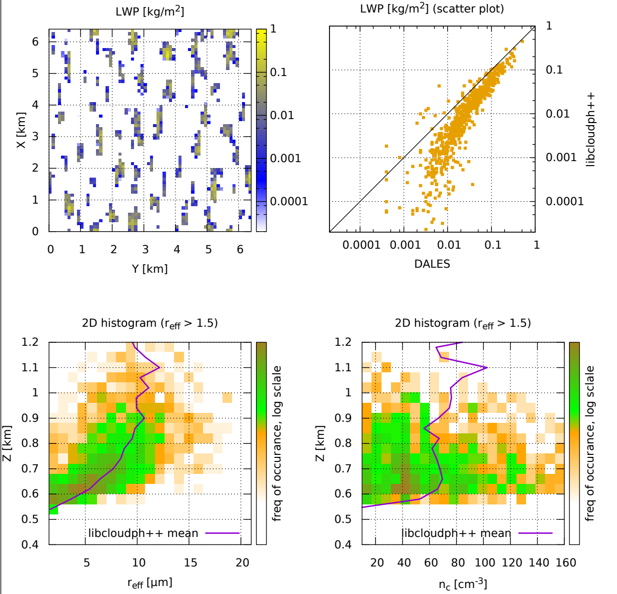
2D histogram (r_{eff} > 1.5)

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=37m)

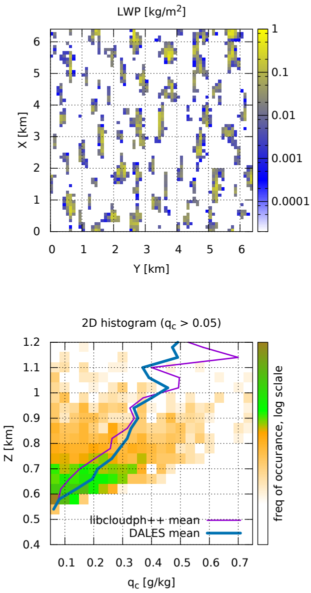


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

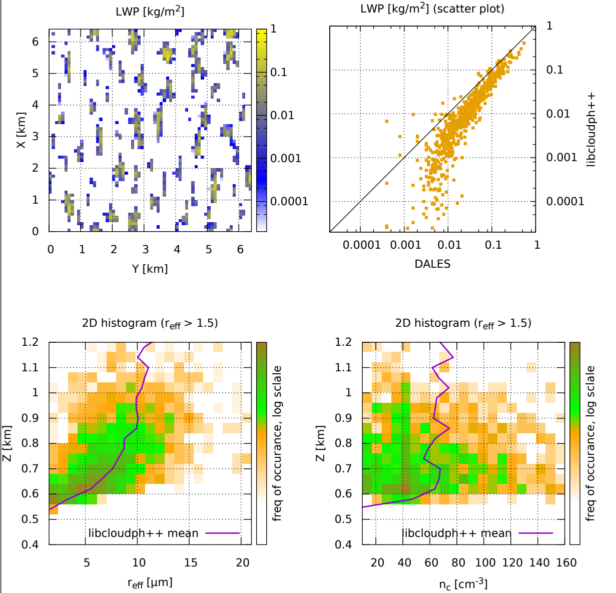


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=38m)

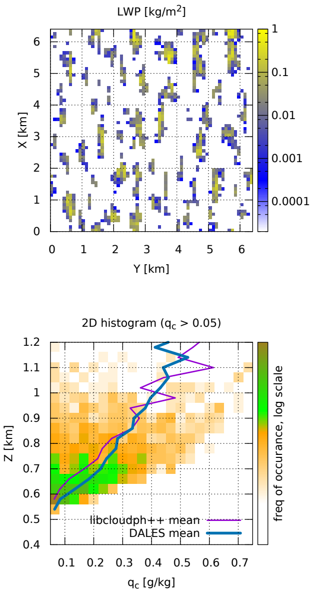


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

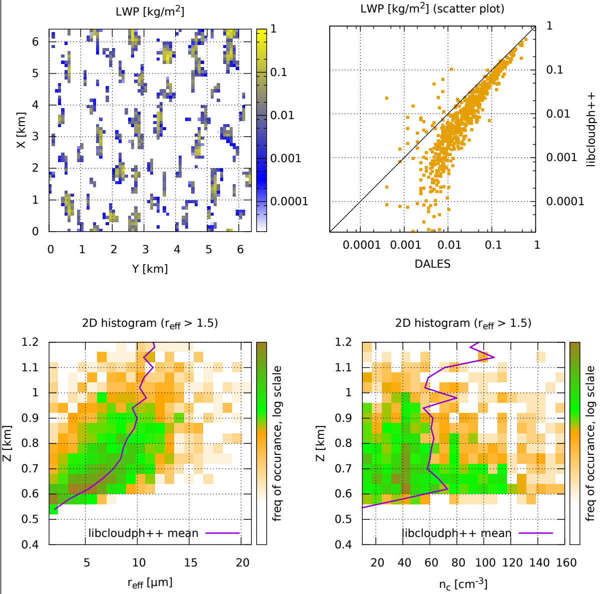


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

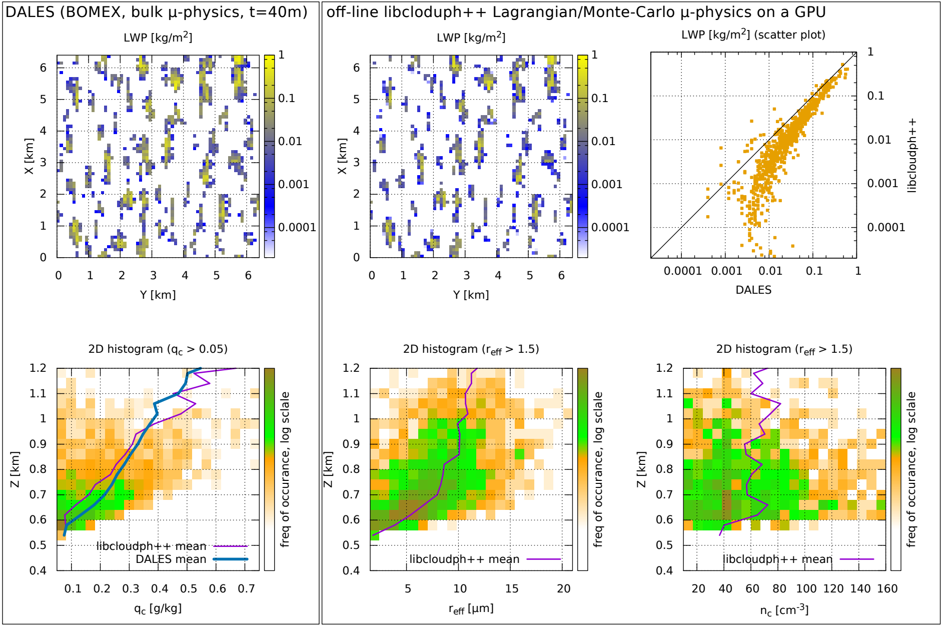
DALES (BOMEX, bulk μ-physics, t=39m)



off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU



DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

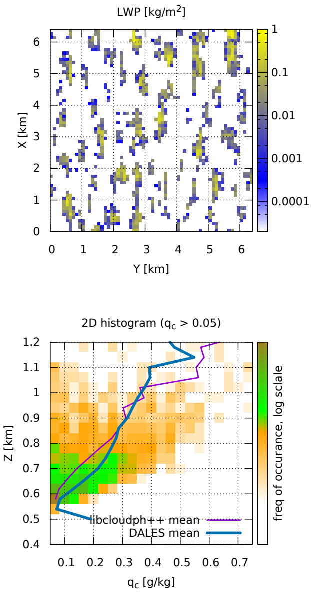


DALES (BOMEX, bulk μ-physics, t=40m)

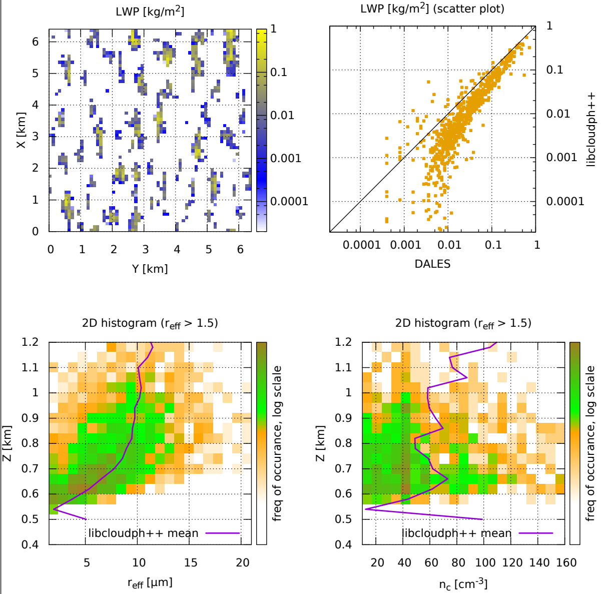
off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=41m)

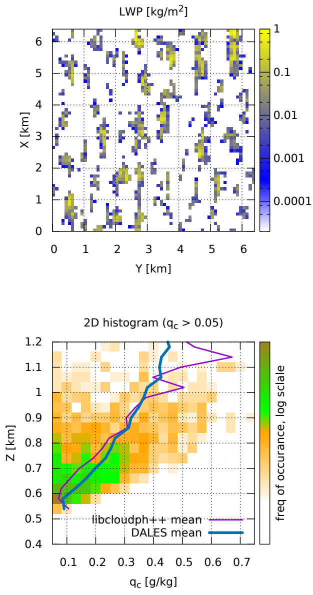


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

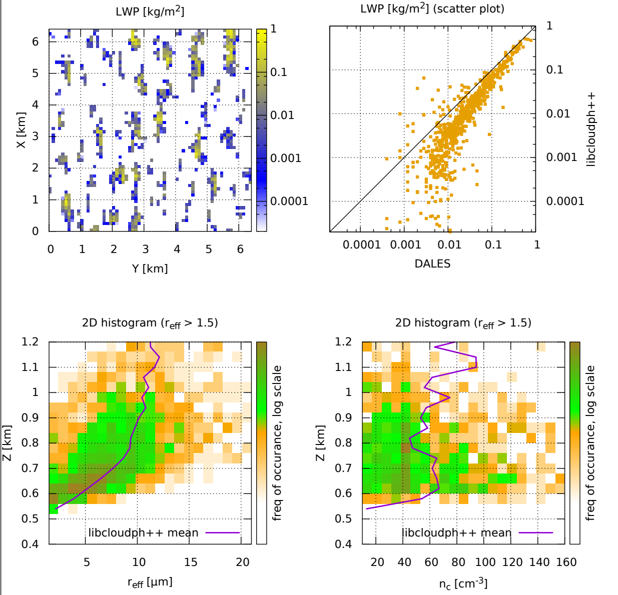


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=42m)

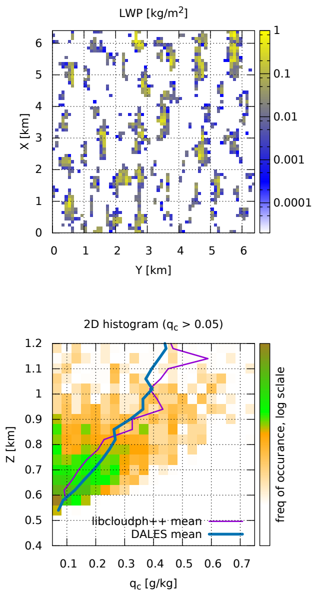


off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU

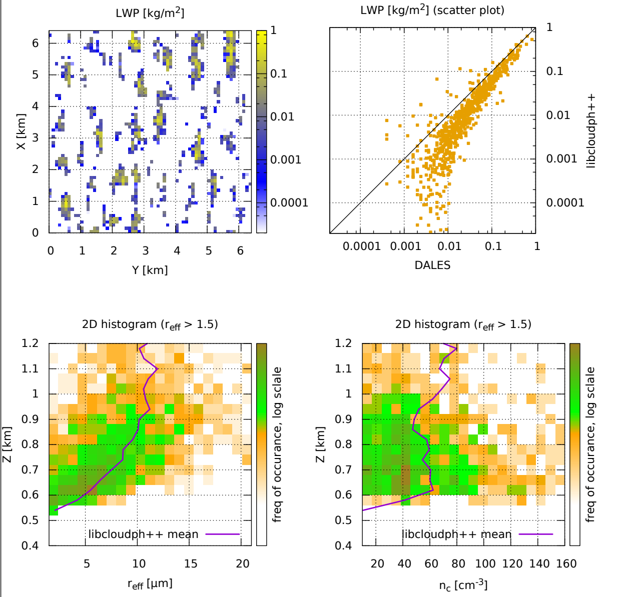


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=43m)



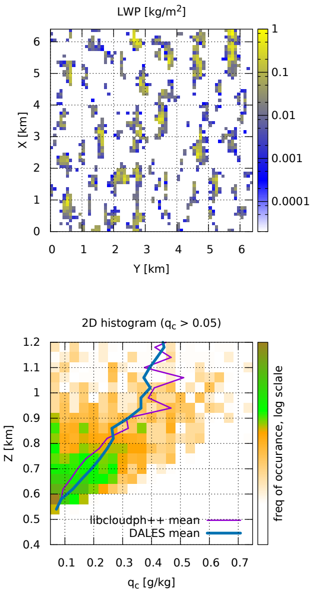
off-line libcloduph++ Lagrangian/Monte-Carlo μ-physics on a GPU



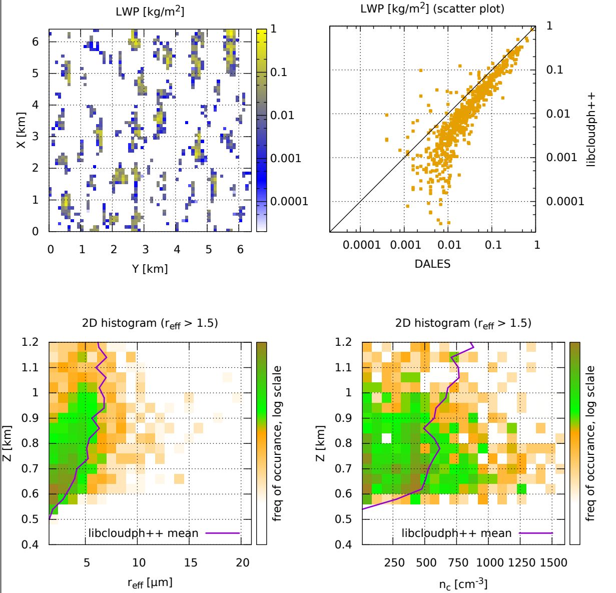
Let's see what happens if we multiply aerosol concentration by a factor of 10...

DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk µ-physics, t=43m)

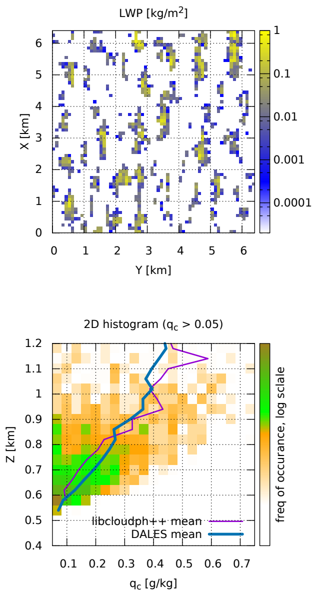


off-line libcloduph++ Lagrangian/Monte-Carlo µ-physics on a GPU

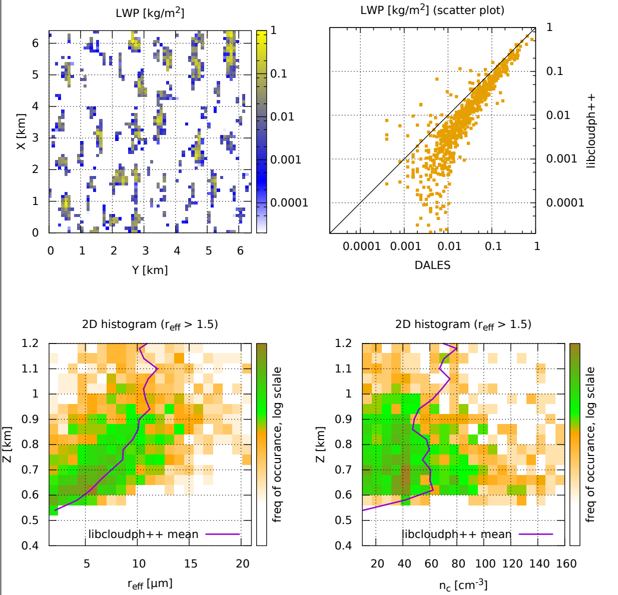


DALES/libcloudph++: proof-of-concept test (dt_{adv}=20s, dt_{cond}=0.5s, dz=40m, 10 SD/cell)

DALES (BOMEX, bulk μ-physics, t=43m)



off-line libcloudph++ Lagrangian/Monte-Carlo μ-physics on a GPU



DALES/libcloudph++ coupling: summary

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:
  - libcloudph++: no modifications

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:
  - libcloudph++: no modifications
  - DALES: ca. 10 LoC changed; ca. 100 LoC added in a new file

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:
  - libcloudph++: no modifications
  - DALES: ca. 10 LoC changed; ca. 100 LoC added in a new file
  - coupling code: ca. 300 LoC in Python

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:
  - libcloudph++: no modifications
  - DALES: ca. 10 LoC changed; ca. 100 LoC added in a new file
  - coupling code: ca. 300 LoC in Python
- next stages:
  - clean-up and test the off-line code
  - work on full bi-directional coupling

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:
  - libcloudph++: no modifications
  - DALES: ca. 10 LoC changed; ca. 100 LoC added in a new file
  - coupling code: ca. 300 LoC in Python
- next stages:
  - clean-up and test the off-line code
  - work on full bi-directional coupling
- potential applications:
  - aerosol-cloud interactions
  - radiative transfer (incl. retrieval algorithms)
  - validating the new “HARMONIA” LES code

DALES/libcloudph++ coupling: summary

- proof-of-concept off-line simulations done
- challenge to the reusability of libcloudph++:
  - libcloudph++: no modifications
  - DALES: ca. 10 LoC changed; ca. 100 LoC added in a new file
  - coupling code: ca. 300 LoC in Python
- next stages:
  - clean-up and test the off-line code
  - work on full bi-directional coupling
- potential applications:
  - aerosol-cloud interactions
  - radiative transfer (incl. retrieval algorithms)
  - validating the new "HARMONIA" LES code
- current libcloudph++ limitations:
  - only geometric collisions, no breakup
  - explicit supersaturation scheme
  - no distributed-memory parallelisation
  - no aerosol sources

Thank you for your attention!

Thank you for your attention!

- "Eulerian vs. Lagrangian methods for cloud microphysics" workshop (Warsaw, April 20-22, 2015, reg. deadline: Jan 31!): `http://foss.igf.fuw.edu.pl/workshop2015/`

Thank you for your attention!

- “Eulerian vs. Lagrangian methods for cloud microphysics” workshop (Warsaw, April 20-22, 2015, reg. deadline: Jan 31!): http://foss.igf.fuw.edu.pl/workshop2015/

- libmpdata++ paper: doi:10.5194/gmdd-7-8179-2014
- libcloudph++ paper: doi:10.5194/gmdd-7-8275-2014

Thank you for your attention!

- "Eulerian vs. Lagrangian methods for cloud microphysics" workshop (Warsaw, April 20-22, 2015, reg. deadline: Jan 31!): `http://foss.igf.fuw.edu.pl/workshop2015/`

- libmpdata++ paper: doi:10.5194/gmdd-7-8179-2014
- libcloudph++ paper: doi:10.5194/gmdd-7-8275-2014

- code repositories: `http://github.com/igfuw/`

Thank you for your attention!

- "Eulerian vs. Lagrangian methods for cloud microphysics" workshop (Warsaw, April 20-22, 2015, reg. deadline: Jan 31!): `http://foss.igf.fuw.edu.pl/workshop2015/`

- libmpdata++ paper: doi:10.5194/gmdd-7-8179-2014
- libcloudph++ paper: doi:10.5194/gmdd-7-8275-2014

- code repositories: `http://github.com/igfuw/`

acknowledgements:

- My visit to TU Delft is supported by Foundation for Polish Science (Mentoring programme)

Thank you for your attention!

- “Eulerian vs. Lagrangian methods for cloud microphysics” workshop (Warsaw, April 20-22, 2015, reg. deadline: Jan 31!): `http://foss.igf.fuw.edu.pl/workshop2015/`
- libmpdata++ paper: doi:10.5194/gmdd-7-8179-2014
- libcloudph++ paper: doi:10.5194/gmdd-7-8275-2014
- code repositories: `http://github.com/igfuw/`

acknowledgements:

- My visit to TU Delft is supported by Foundation for Polish Science (Mentoring programme)
- Development of libmpdata++ and libcloudph++ have been supported by Poland’s National Science Centre (decision no. 2012/06/M/ST10/00434)