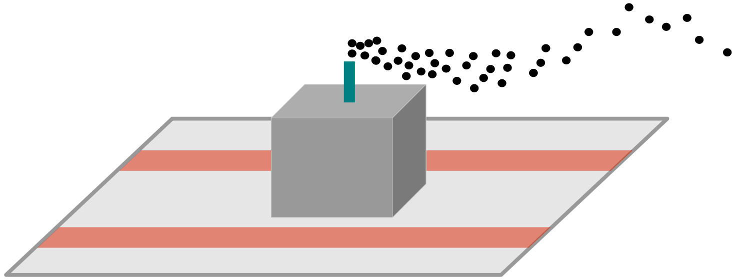
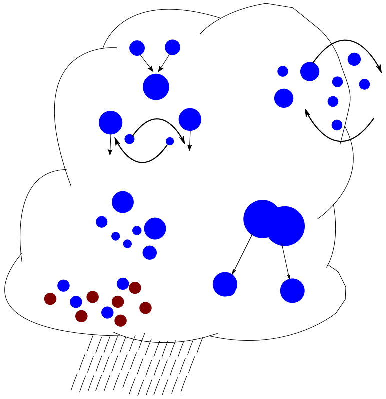
Lagrangian particle model

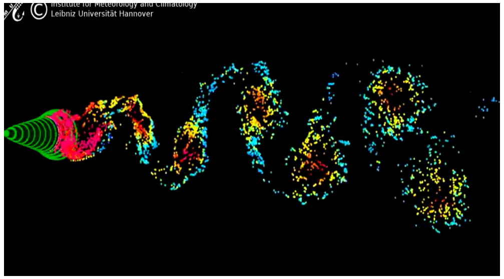
Institute of Meteorology and Climatology, Leibniz Universität Hannover

- Physics of Lagrangian Particle Model
- Physics and concept of Lagrangian Cloud Model
- Steering
- Implementation
- Applications

- The Lagrangian particle model (LPM) embedded in PALM can be used for different purposes:
  - Dispersion modeling / Footprint analysis
  - Cloud droplet simulations
  - Visualization



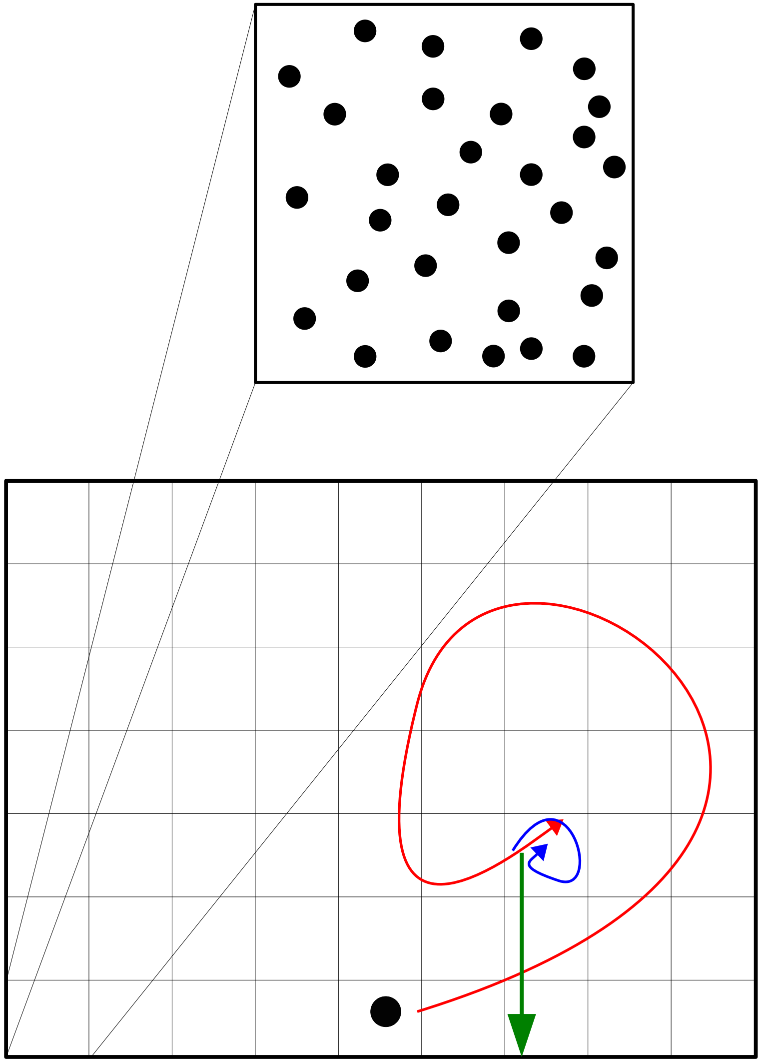




Lagrangian particle model

General

- Particles are not fixed on the Eulerian grid
- Particles are implemented and treated as objects
- Each particle contains several attributes

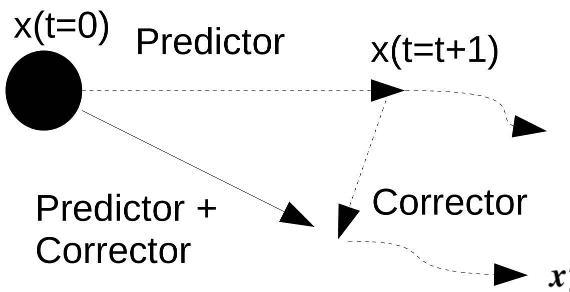


- Particles can be transported **passively** with the **resolved-scale** flow
- Particle transport by **subgrid-scale (SGS) turbulence**
- Particles can be given a mass and thus an **inertia** and a **radius** which results in gravitational settling velocity.

Lagrangian particle model

Physics – Passive resolved Advection

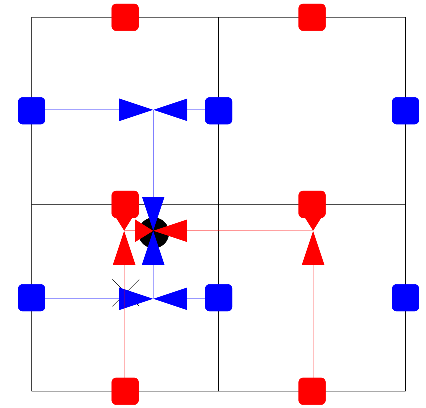
- Particles are advected passively with the underlying LES fields interpolated to the position of the particle
- Currently, two interpolation methods are availible:
  - `trilinear`
  - `simple_predictor`
  - `simple_corrector` **(Recommended)**
- Furthermore, latter involves a predictor-corrector timestep:

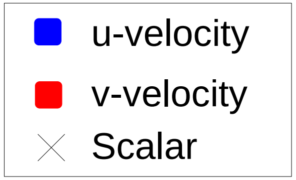


$$x_{\mathrm{p}}^{n+1} = x^n + u^n(x^n)\Delta t$$

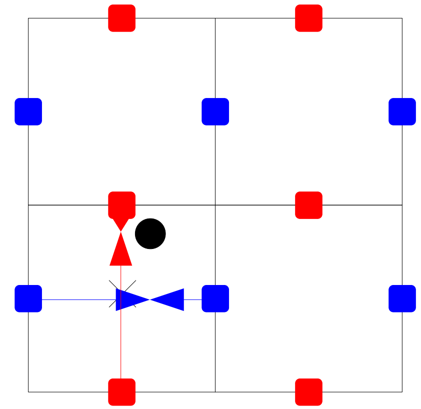
$$x_{\mathrm{c}}^{n+1} = x^n + \left[u^{n+1}(x_{\mathrm{p}}^{n+1}) + u^n(x^n)\right]\frac{\Delta t}{2}$$

Trilinear

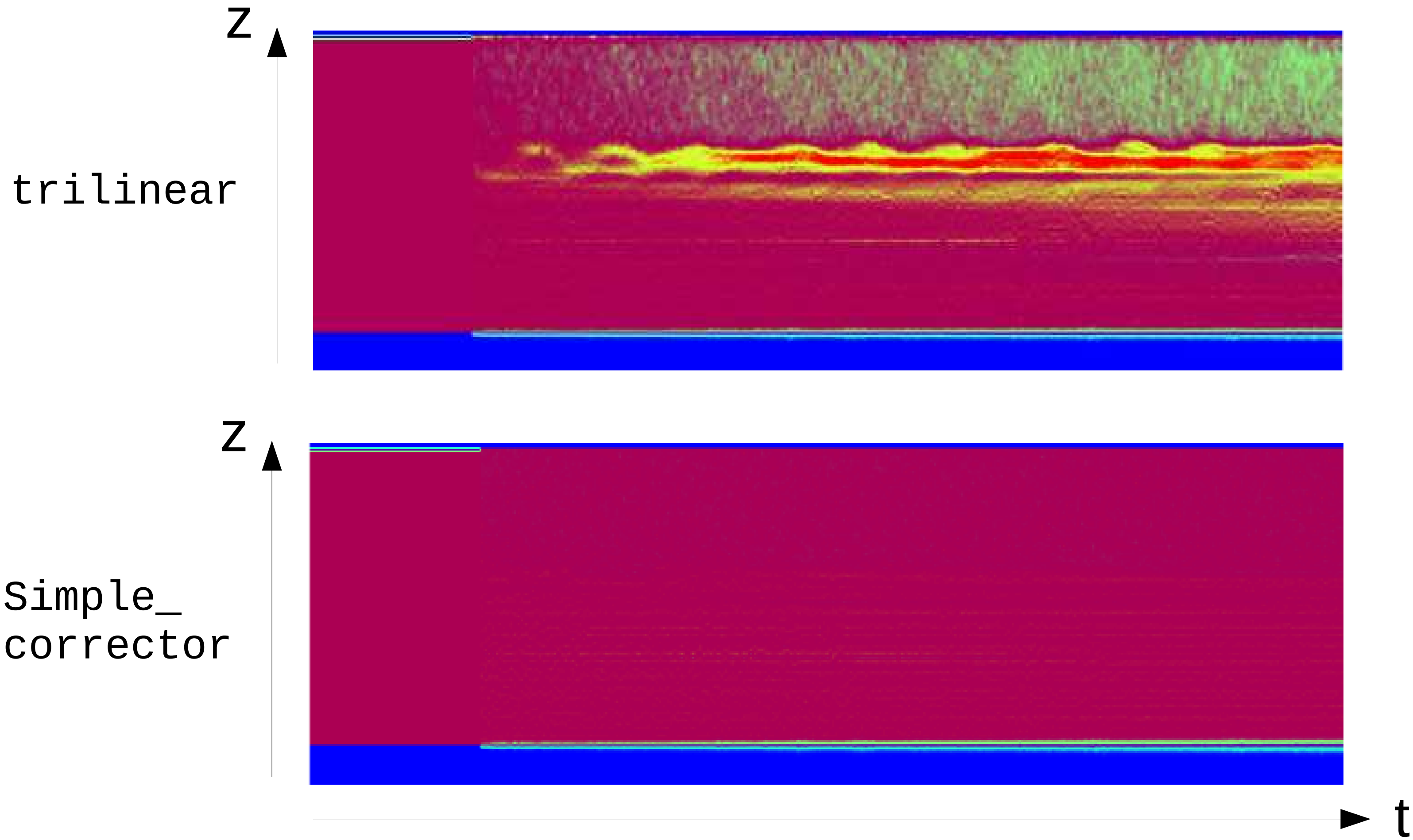




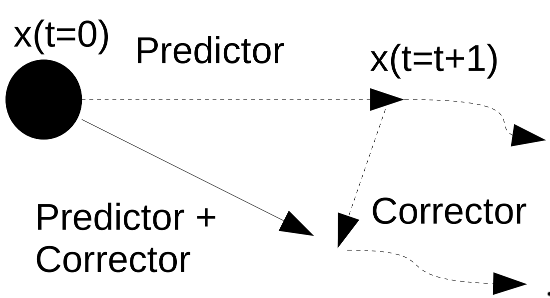
Simple



Particle concentration



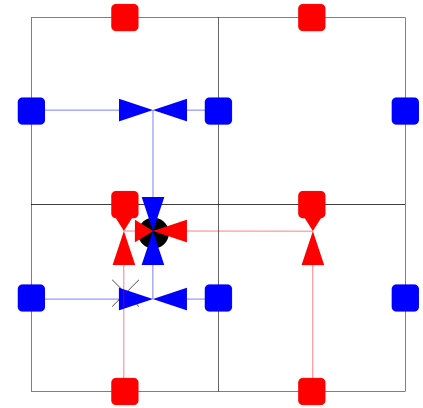
- Particles are advected passively with the underlying LES fields interpolated to the position of the particle
- Currently, two interpolation methods are availible:
  - `trilinear`
  - `simple_predictor`
  - `simple_corrector` **(Recommended)**
- Furthermore, latter involves a predictor-corrector timestep:

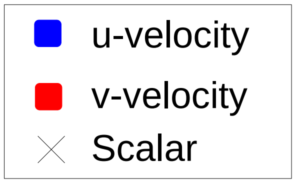


$$\boldsymbol{x}_{\mathrm{p}}^{n+1} = \boldsymbol{x}^{n} + \boldsymbol{u}^{n}(\boldsymbol{x}^{n})\Delta t$$

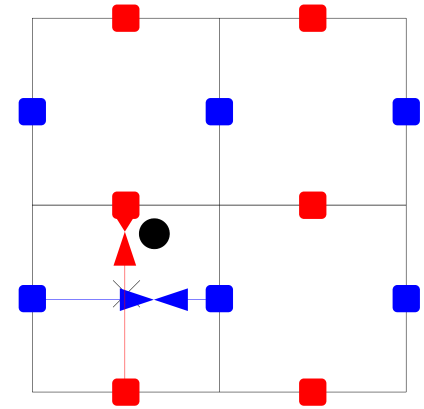
$$\boldsymbol{x}_{\mathrm{c}}^{n+1} = \boldsymbol{x}^{n} + \left[\boldsymbol{u}^{n+1}(\boldsymbol{x}_{\mathrm{p}}^{n+1}) + \boldsymbol{u}^{n}(\boldsymbol{x}^{n})\right]\frac{\Delta t}{2}$$

Trilinear





Simple



- Besides the resoved part of the velocity, particles can have velocities on the subgrid scale and fall velocities

$$\vec{V}_{\text{particle}} = \vec{V}_{\text{res}} + \vec{V}_{\text{sgs}} + \delta_{i3} U_{\infty}$$

- SGS-component important in areas where resolved scale is small (e.g. building walls) or for entrainment processes (e.g. cloud edge)
- Currently, different methods for calculating SGS-velocities are applied.

SGS - passive Particles

$$du_{\mathrm{p},i}^{\mathrm{sgs}} = -\frac{3 c_{\mathrm{sgs}} C_{\mathrm{L}} \epsilon}{4} \frac{u_{\mathrm{p},i}^{\mathrm{sgs}}}{e} \Delta t_{\mathrm{L}} + \frac{1}{2} \left(\frac{1}{e} \frac{\mathrm{d}e}{\Delta t_{\mathrm{L}}} u_{\mathrm{p},i}^{\mathrm{sgs}} + \frac{2}{3} \frac{\partial e}{\partial x_i} \right) \Delta t_{\mathrm{L}} + \left(c_{\mathrm{sgs}} C_{\mathrm{L}} \epsilon \right)^{\frac{1}{2}} \mathrm{d}\zeta_i$$

After Weil et al., 2004

SGS - cloud droplets

$$\vec{V}_{\text{sgs}}(t) = R_L \vec{V}_{\text{sgs}}(t - \Delta t) + \sqrt{1 - R_L^2} \left(\sqrt{e}\, \zeta \right)$$

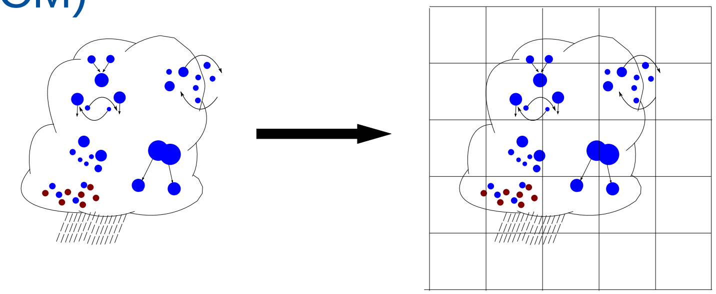
After Sölch and Kärcher, 2010

Deposition velocities

$$U_{\infty}(r) = \begin{cases} 8000\, r \left[1 - \exp(24000\, r) \right] & r \leq 372.5 \mu\text{m} \\ 9.65 - 10.43 \exp(-1200\, r) & r > 372.5 \mu\text{m} \end{cases}$$

After Rogers et al., 1993

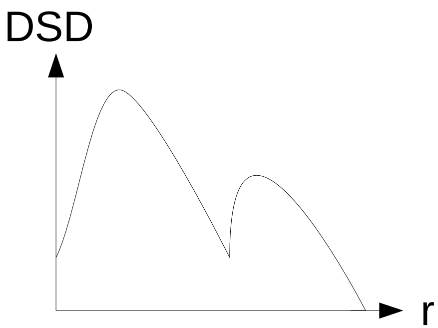
How to simulate clouds

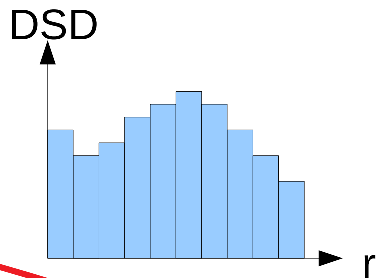


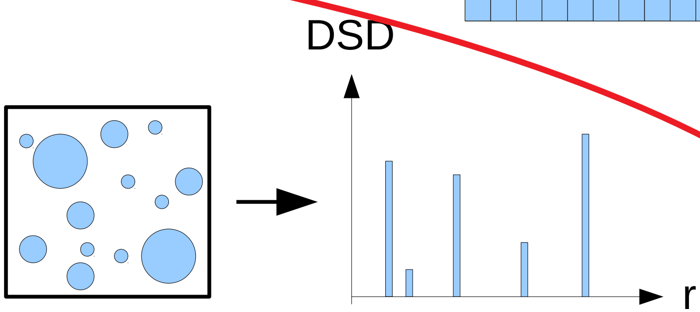
3 basic approaches

demand of computational power

- **Bulk cloud models**
- **Spectral-bin models**
- **Particle-based models/ Lagrangian Cloud models (LCM)**



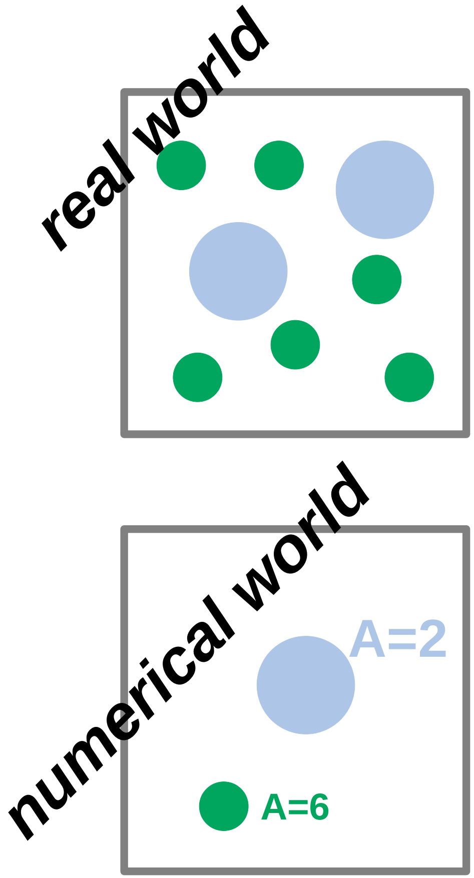




- Relatively new method to simulate cloud microphysical processes using single particles

  Models e.g. by *Andrejczuck et al., 2008, Shima et al., 2009, Sölch and Kärcher, 2011, Riechelmann et al., 2012, Arabas et al., 2015, Naumann et al., 2015, Grabowski et al., 2018*

- Here a simulated particle, so-called superdroplets, represents a large number of real droplets or aerosols
- Microphysical processes such as activation, diffusional growth and collision are considered explicitly for each superdroplet, depending on their size
- The thermodynamic coupling is done by the release of latent heat during the condensation or evaporation process altering the LES fields of water vapor and potential temperature



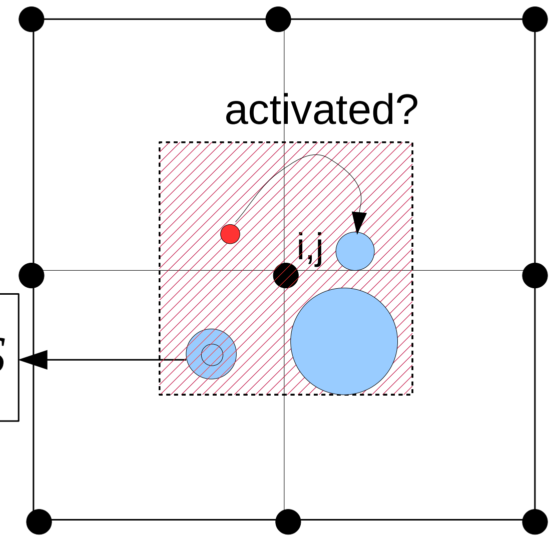
Activation and Diffusional growth

- Calculated for every particle individually
- Activation explicitly treated with:

`curvature_and_solution_effects=.TRUE.`

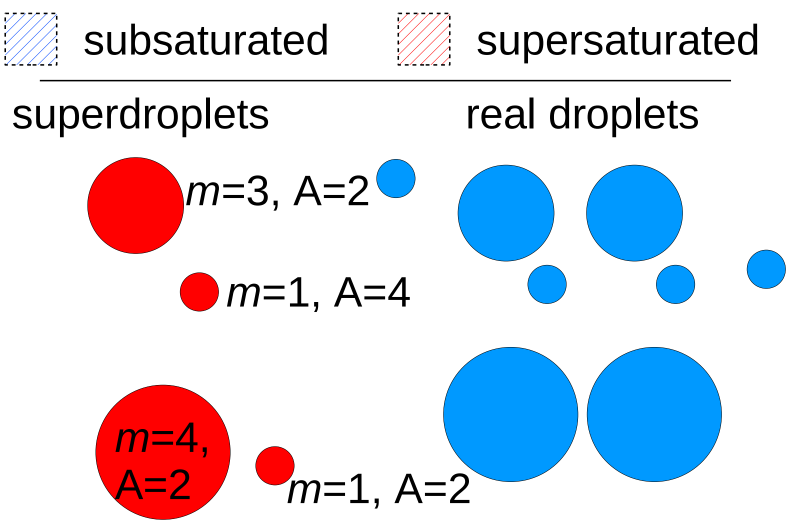
→ Particle's curvature and physical and chemical properties of aerosol

$$r\frac{\mathrm{d}r}{\mathrm{d}t} \propto S$$



Collision and Coalescence

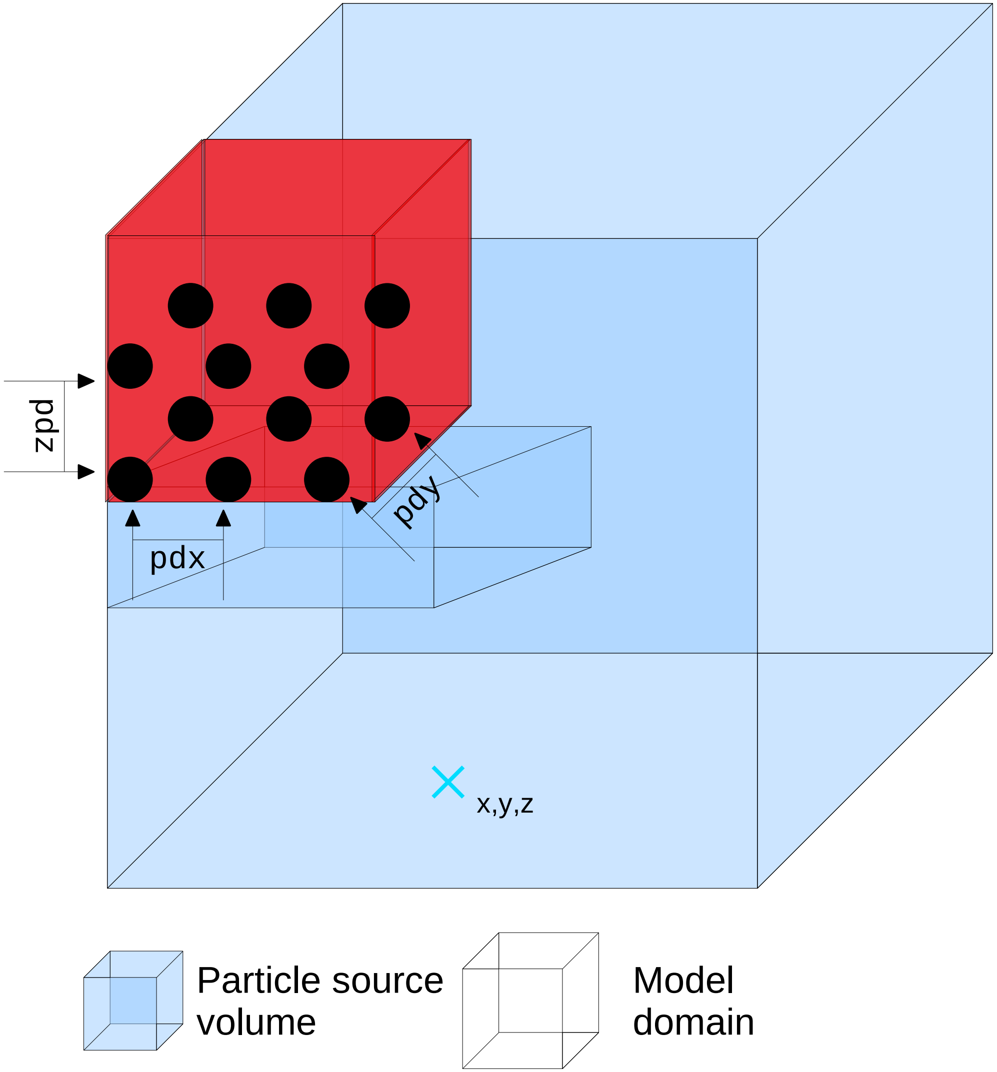
- Two prognostic quantities:
  - **weighting factor** *A*
  - **total mass** *m* of super-droplet
- Superdroplets of one grid box are allowed to collect each other (independent of particle positions within grid box)
- Collection takes place, if calculated probability is bigger than random number
- Turbulence effects can be considered
- `Splitting` improves collisional growth



Lagrangian particle model

Steering - Spatial Initialization

- Define boundaries of particle source volume, e.g.:
  - Arbitrary volume
  - Point Source(s)
  - Whole model domain
- Bottom (`psb`) and Top (`pst`)
- Left (`psl`) and Right (`psr`)
- South (`pss`) and North (`psr`)
- Afterwards, define distance between particles within source volume
- Or prescribe number of particles per grid box
- Optional: Add random fluctuations

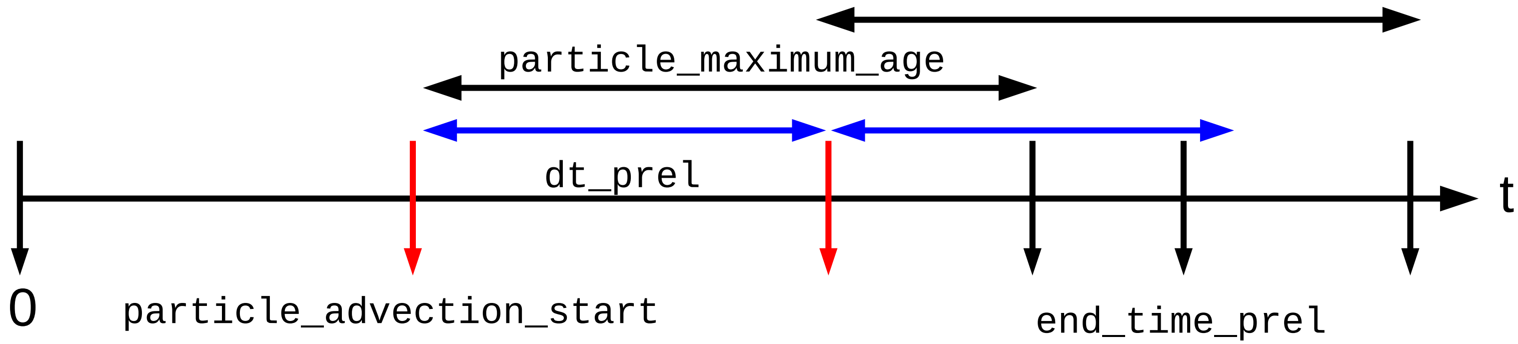


Lagrangian particle model

Steering - Temporal Initialization

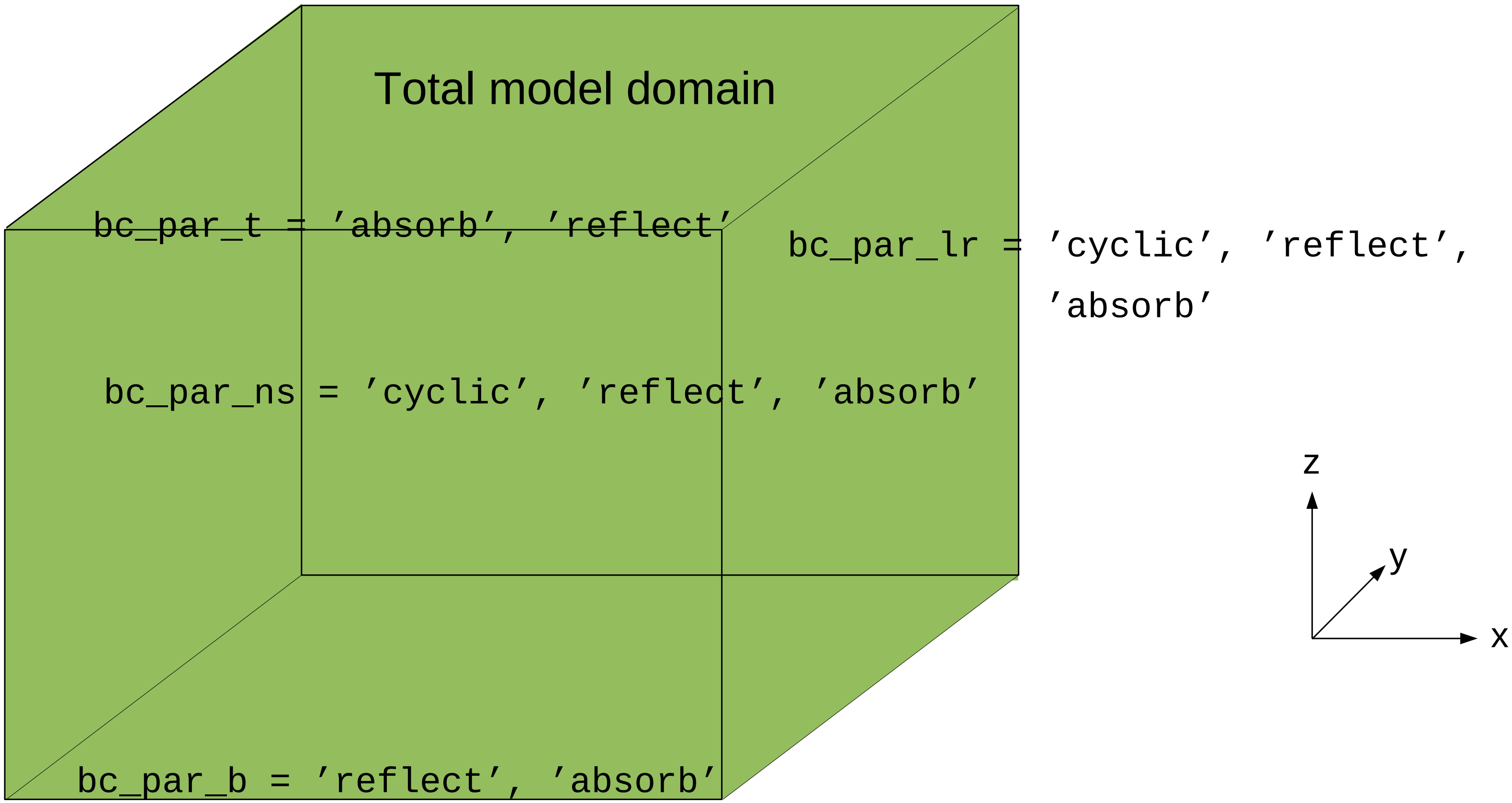
Parameters that define the period of particle release:

- `particle_advection_start`: First release of particles (in s)
- `dt_prel`: Interval at which particles are to be released (in s)
- `particle_maximum_age`: Maximum allowed age of particles (in s)
- `end_time_prel`: Time of the last release of particles (in s)



Start of LES, determination of particle release points	Calculation of particle model begins (default value: 0.0)	Release of a new set of particles	Deletion of the first set of particles	No release of new particles after this time	Deletion of the second set of particles

Parameters that define the boundary conditions for particles



&particles_parameters

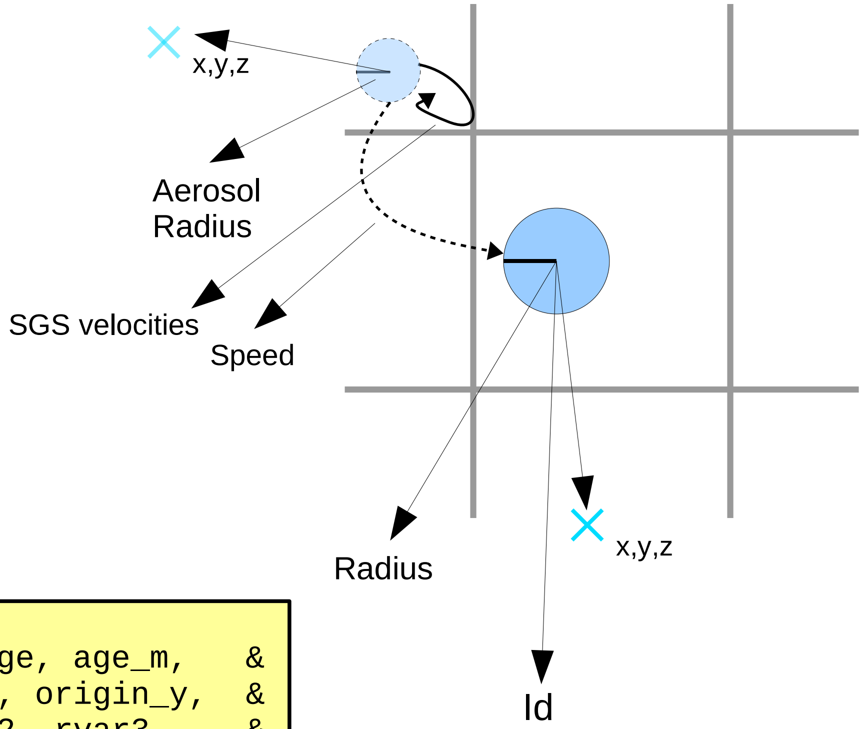
General parameters

psb, pst, psl, psr, pss, psn	Source Volume
pdx, pdy, pdz	Distance between particles
number_particles_per_gridbox	Or set particles per gridbox
particles_per_point	Particles per source point
dt_dopts	Output intervall particle timeseries
particle_advection_start	Start of particle advection
particle_advection_interpolation	Interpolation method for particle advection
number_of_particle_groups	Number of particle groups
use_sgs_for_particles	Switch on sgs velocities for particles

LCM parameters

cloud_droplets (&initializations_parameters)	Switch on cloud droplets
collision_kernel	Switch on collision process
curvature_solution_effects	Switch on explixit activation
density_ratio	Set inertia for particles
number_concentration	Set number concentration
initial_weighting_factor	Or initialize weighting factor

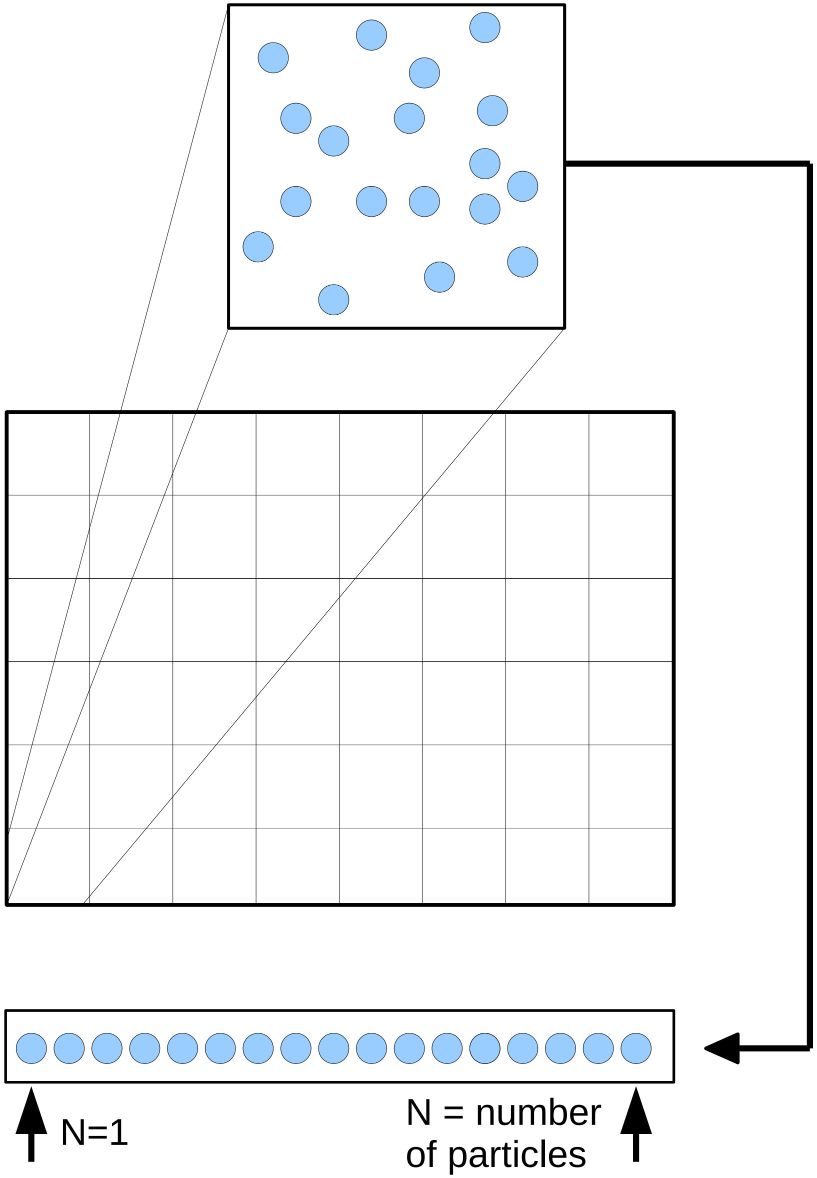
- Particles are stored in a FORTRAN derived data type
- A derived data type consists of several elements, which can be accessed by the % operator
- Advantage: Attributes of particles are easily extendable




```
TYPE particle_type
    REAL(wp)        :: aux1, aux2, radius, age, age_m,    &
                       dt_sum, e_m, origin_x, origin_y,   &
                       origin_z, rvar1, rvar2, rvar3,     &
                       speed_x, speed_y, speed_z,         &
                       weight_factor, x, y, z,
    INTEGER(iwp)    :: class, group
    INTEGER(idp)    :: id
    LOGIACAL        :: particle_mask
    INTEGER(iwp)    :: block_nr

END TYPE particle_type
```

- Handling hundreds of millions of particles, efficient storing is essential for a good performance
- Most applications demand particles located at a certain location (e. g., advection, collision process is computed for all particles located in a certain grid box)
- Sorting the particles by their respective grid-box increases the computability of the code, but needs time for the sorting itself
- A efficient approach for storing particles is implemented by storing in a four-dimensional array



There are different files containing particle Output data

Particle timeseries

- Name: `DATA_1D_PTS_NETCDF`
- Steering: `dt_dopts`
- Contains: e.g. time series of the total of particles, mean particle velocity, mean subgrid scale part of the particle velocity, mean particle location etc.

2d-cross section and 3D-data

- Name: `DATA_2D/3D_[XYZ] NETCDF`
- Steering: in `&runtime_parameters`
- Quantities: `pc` and `pr` (particle concentration per grid box and averaged particle radius per grid box)

All particle data

Very large

- Name: `PARTICLE_DATA`
- Steering: `dt_write_particle_data`
- Contains: all particle data for given time interval. Each core create one file. Data is in binary format. Program to process data is supplied in ../trunk/UTIL/

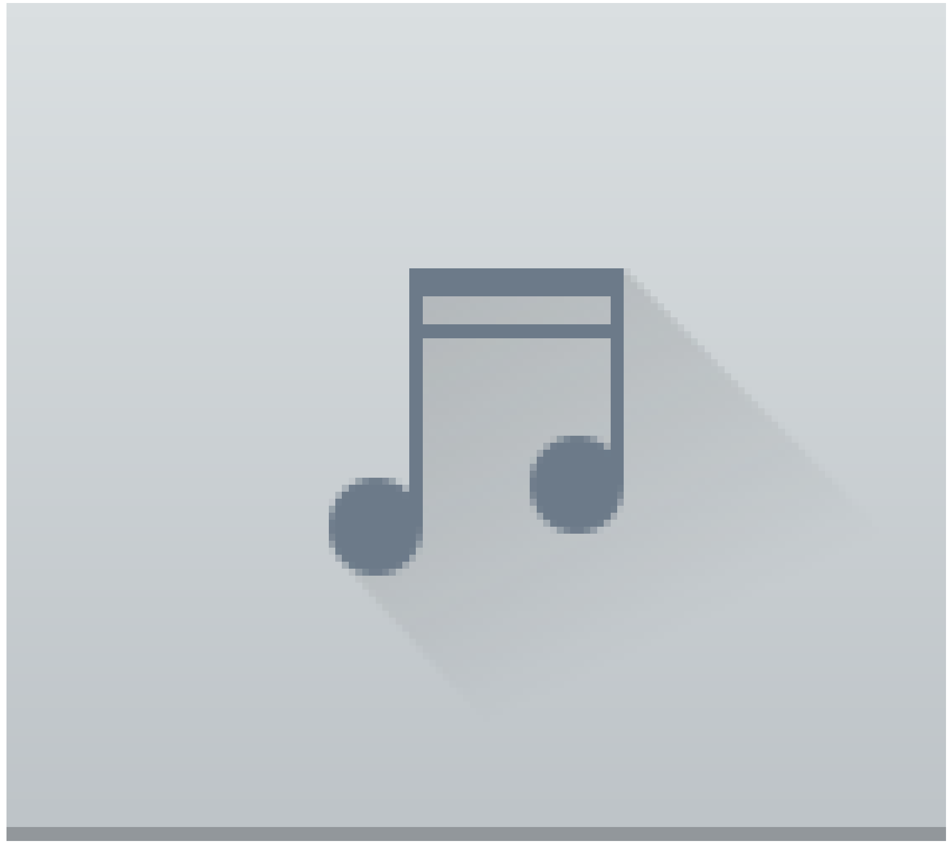
Individual particle timeseries

Requires parallel Netcdf4

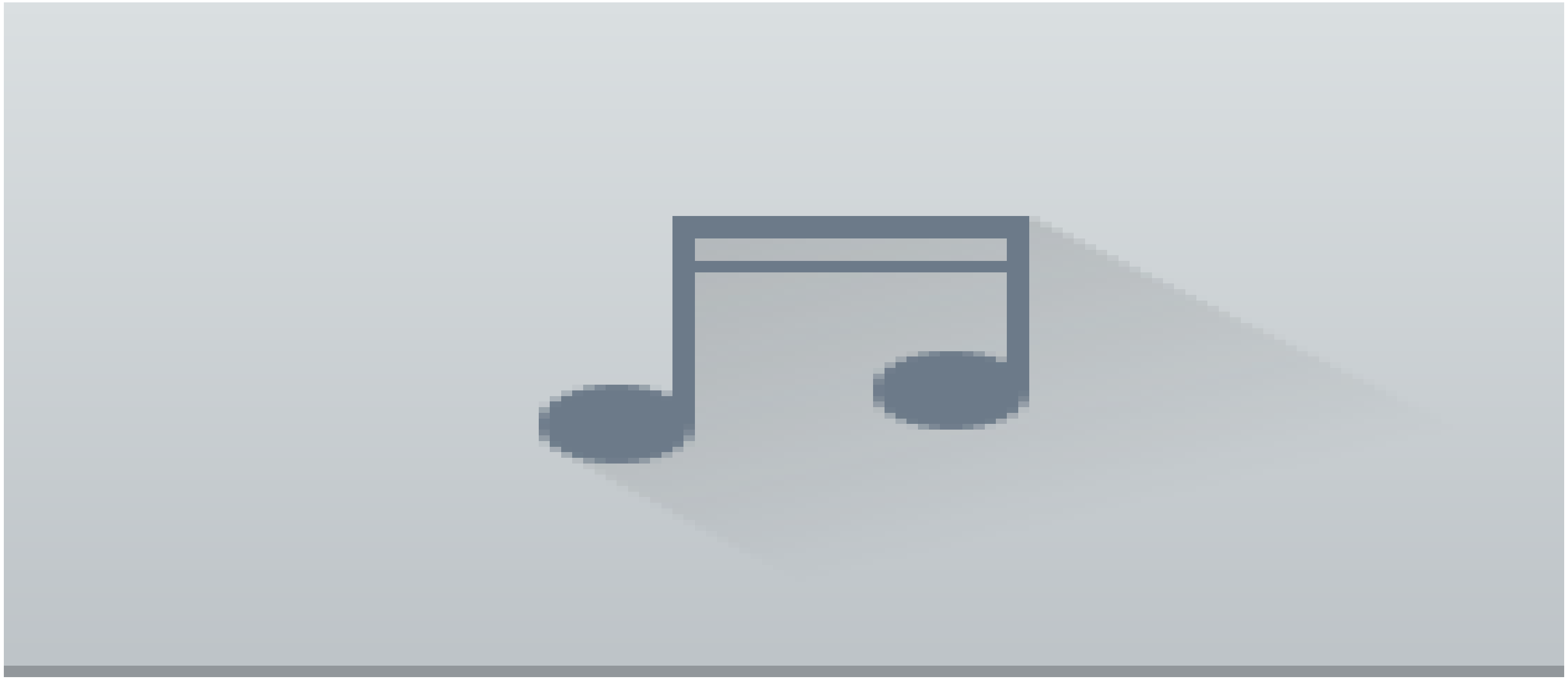
- *Time series in Netcdf format of individual particles of selected or randomly chosen particles.*

Lagrangian particle model

Applications

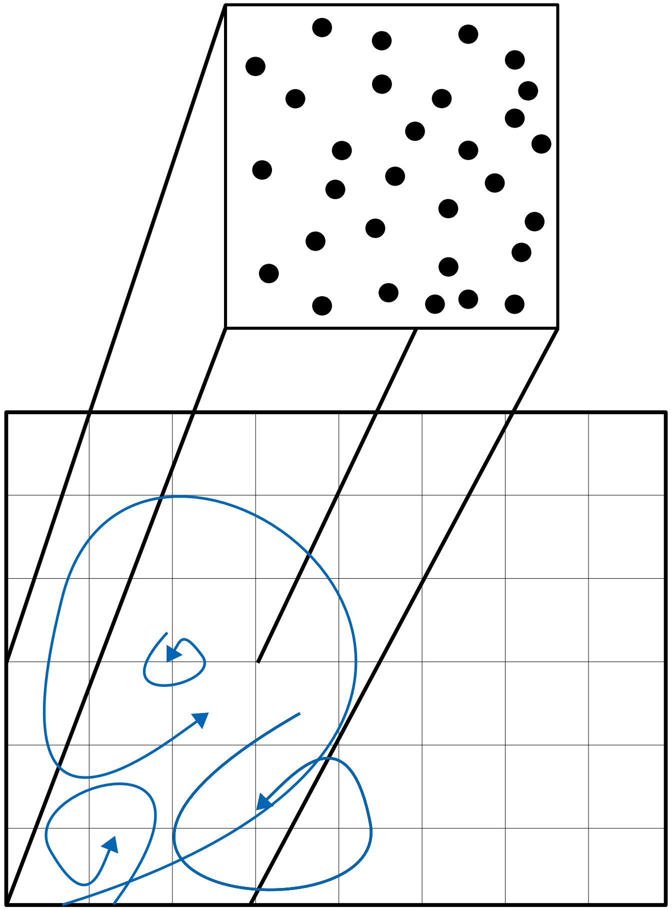


- Development of the LWC and particles' radii during convective situation
- You can find out what a droplet is experiencing during its lifetime



- Development of atmospheric vortex streets consisting of counter rotating mesoscale eddies with vertical axis in the wake of large island
- Particles are released in one layer and act as passive tracers. Their vertical motion is disabled
- Blue/red colors represent a relatively low/high temperature at particles position

- Less numerical errors, point sources can be represented
- History of air parcels can be detected
- Flow can easily visualized
- Many microphysical processes are modeled by first principles
- We are able to simulate cloud microphysics on a very accurate level, but we are also able to cope the macro-scale, i. e., a whole cloud or cloud ensemble by LES
- The LCM provides detailed information, e. g., spatial and temporal evolution of the droplet spectrum, droplet trajectories, …



PALM online:

https://palm.muk.uni-hannover.de

Our YouTube channel:

youtube.com/user/palmhannover