University of Warsaw Lagrangian Cloud Model (UWLCM)

a modern LES with Lagrangian cloud microphysics

Piotr Dziekan, Maciej Waruszewski, Hanna Pawłowska
Institute of Geophysics, Faculty of Physics, University of Warsaw

EULAG workshop
Warsaw, 05.2018

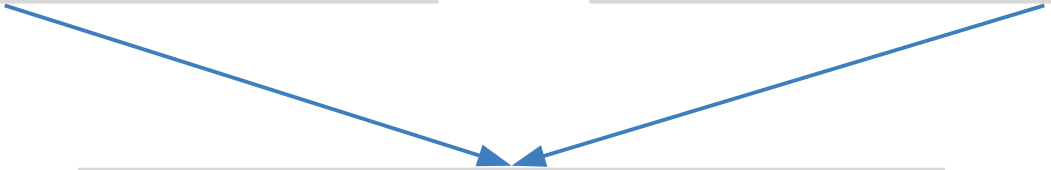
Software developed by our group

libmpdata++

- new implementation of MPDATA
- written in C++
- started ca. 5 years ago

libcloudph++

- cloud microphysics routines
- written in C++ with python bindings
- started ca. 5 years ago



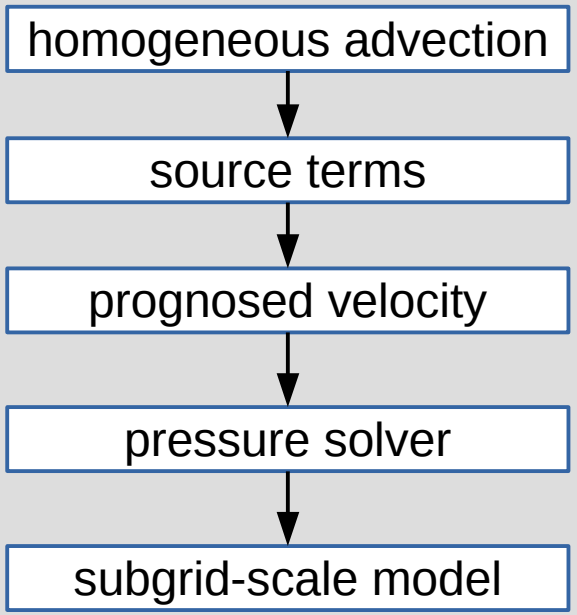
UWLCM

- Large Eddy Simulations of clouds using anelastic approximation
- written in C++
- started ca. 2 years ago

Modern code structure: separation of concerns

libmpdata++

- hierarchy of solvers:

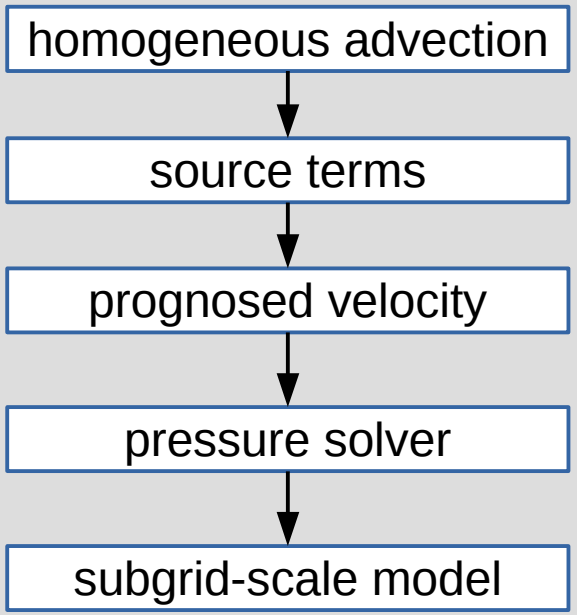


- boundary conditions
- concurrency handlers
- output handlers

Modern code structure: separation of concerns

libmpdata++

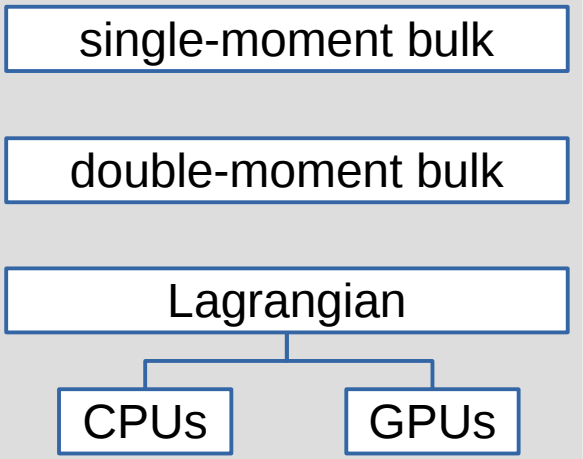
- hierarchy of solvers:



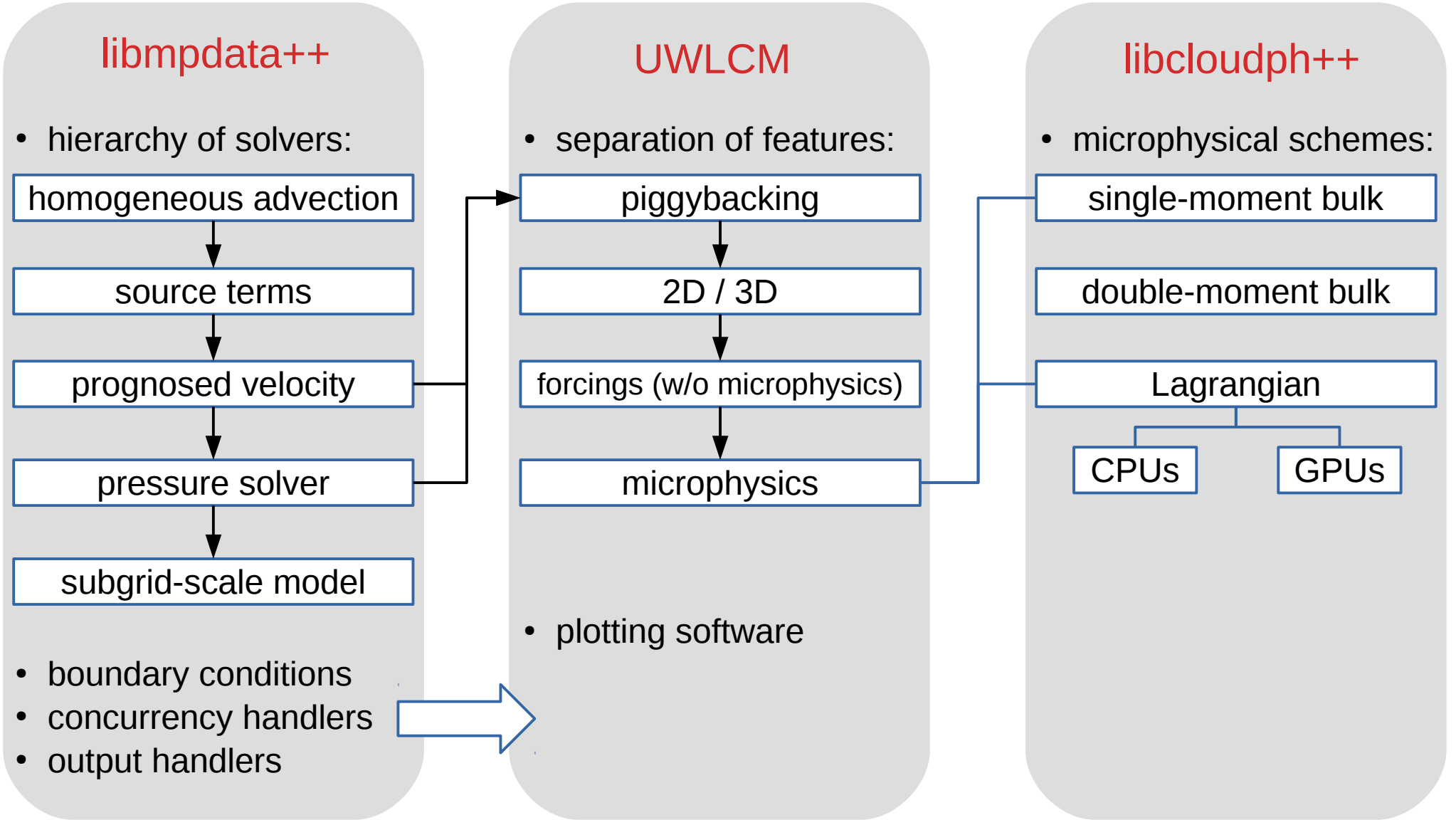
- boundary conditions
- concurrency handlers
- output handlers

libcloudph++

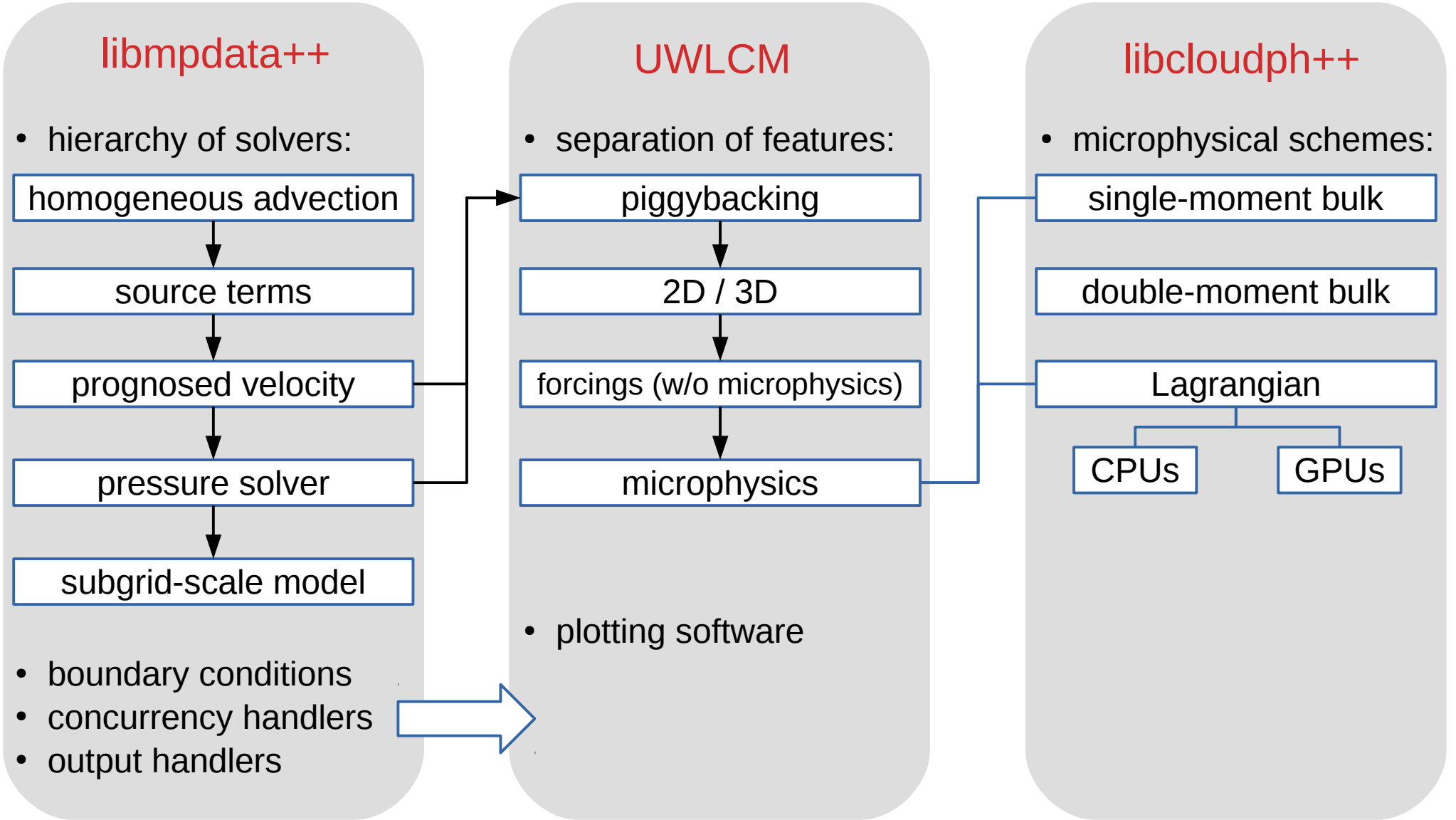
- microphysical schemes:



Modern code structure: separation of concerns



Modern code structure: separation of concerns



- Code sections can be developed independently.
- Code are sections ready to be reused.

- version control system
- automated tests
- open-source code hosted on github

Benefits of the modern code structure: one code, many models

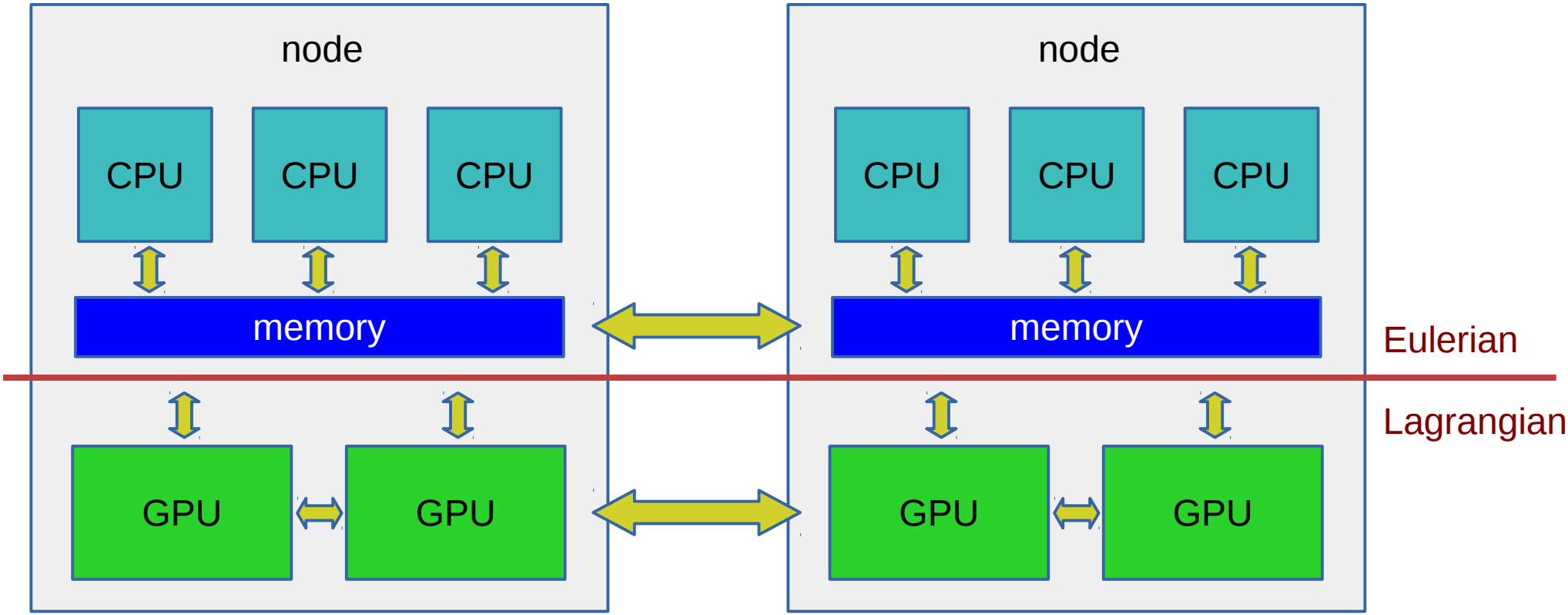
runtime options:

- number of dimensions:
  ./bicycles --ny=[0, X]
- type of microphysics:
  ./bicycles --micro=[blk_1m, lgrngn]
- where to calculate Lagrangian microphysics:
  ./bicycles --backend=[serial, OpenMP, CUDA, multi_CUDA]
- model setup:
  ./bicycles --case=[dycoms, bomex, ...]
- piggybacking:
  ./bicycles --piggy=1 --vel_in=file
- number of CPU threads for dynamics:
  OMP_NUM_THREADS=X ./bicycles
- distributed memory runs:
  mpiexec -np X ./bicycles
- advection, coalescence, condensation timesteps
- number of super-droplets
- ...

compile time options:

- MPDATA options
  - variable-sign option:
    opts = [opts::iga, opts::abs]
  - non-oscillatory option:
    opts = [opts::fct]
  - third-order terms:
    opts = [opts::tot]
  - ...
- microphysics options
  - coalescence kernel
  - terminal velocity
  - ....
- numerical precision
  real_t = [float, double]

Modern HPC architecture



- Domain decomposition between nodes
- Separate intra-node domain decomposition for CPU threads and GPUs
- Bulk microphysics computed on CPUs
- Lagrangian microphysics computed on CPUs or GPUs
- Simultaneous computations of fluid flow on CPUs and microphysics on GPUs synchronized only during condensation

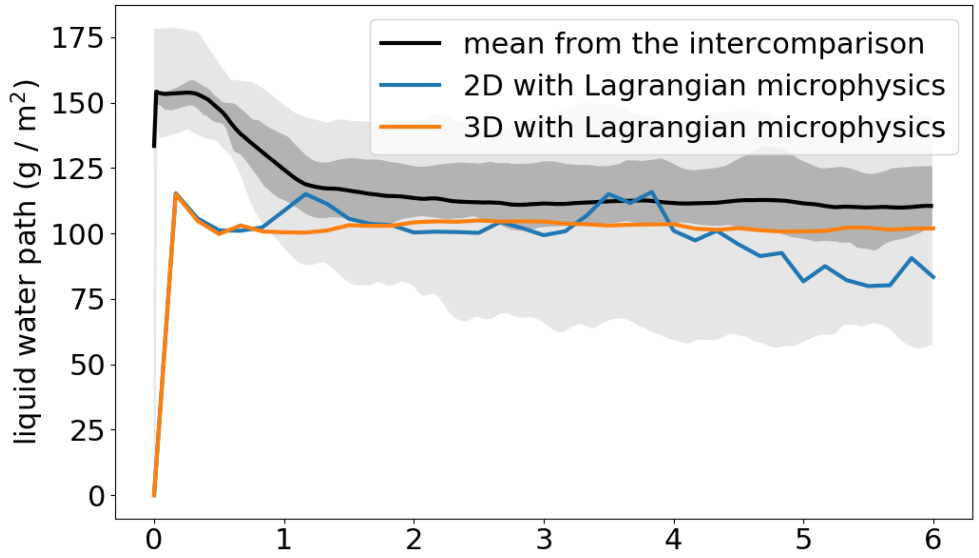
Model validation - stratocumulus

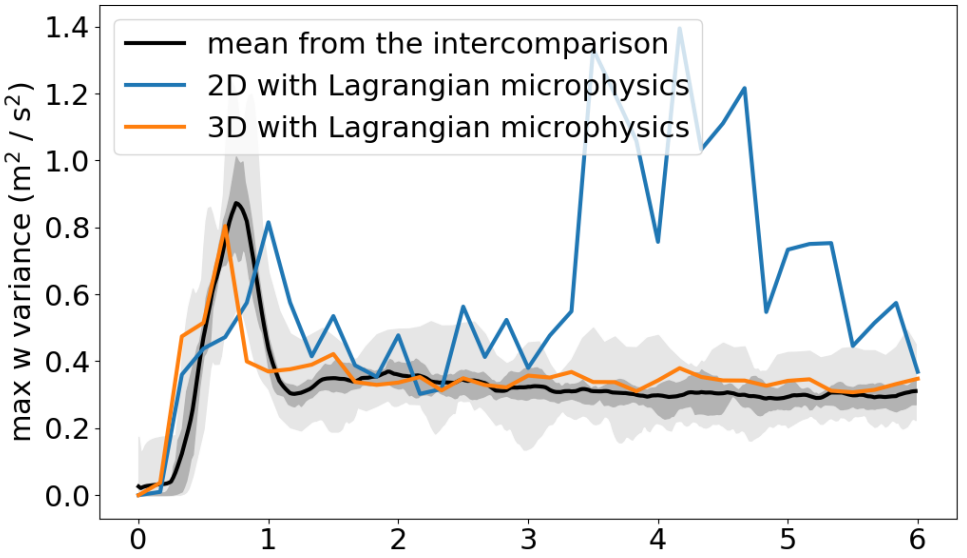
- Test UWLCM against 14 models from the Ackerman et al. 2009 intercomparison
- Microphysical schemes in other models:
  bin, single-moment bulk and double-moment bulk
- implicit LES in UWLCM, other models with subgrid-scale schemes

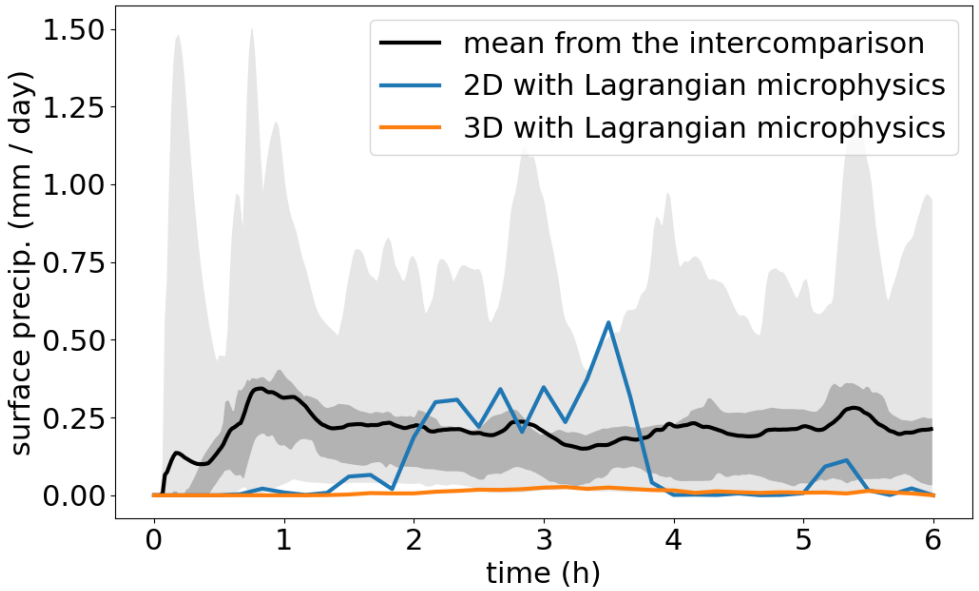
source: Angela Rowe
communitycloudatlas.wordpress.com

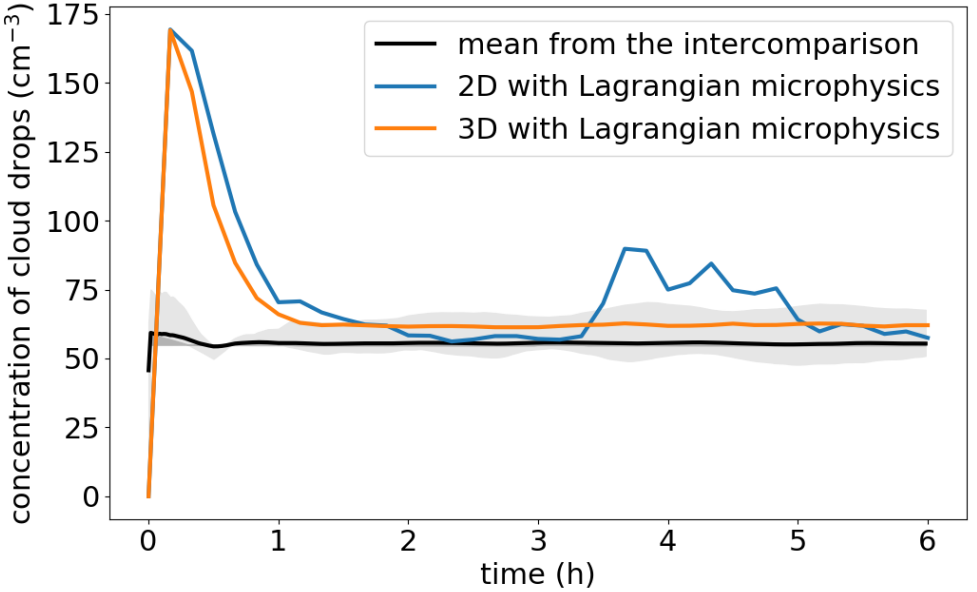
ANIMATION

Stratocumulus time series

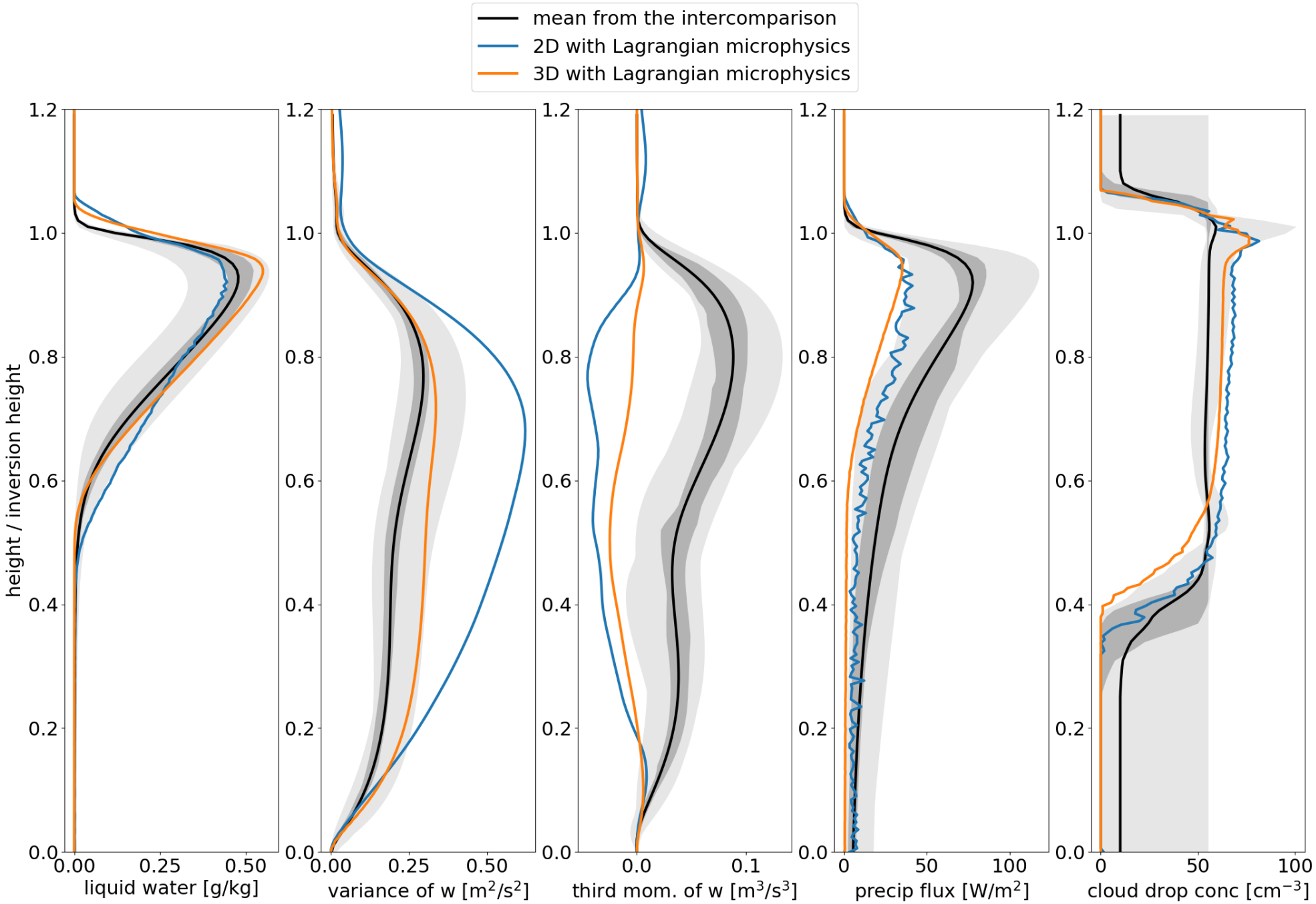




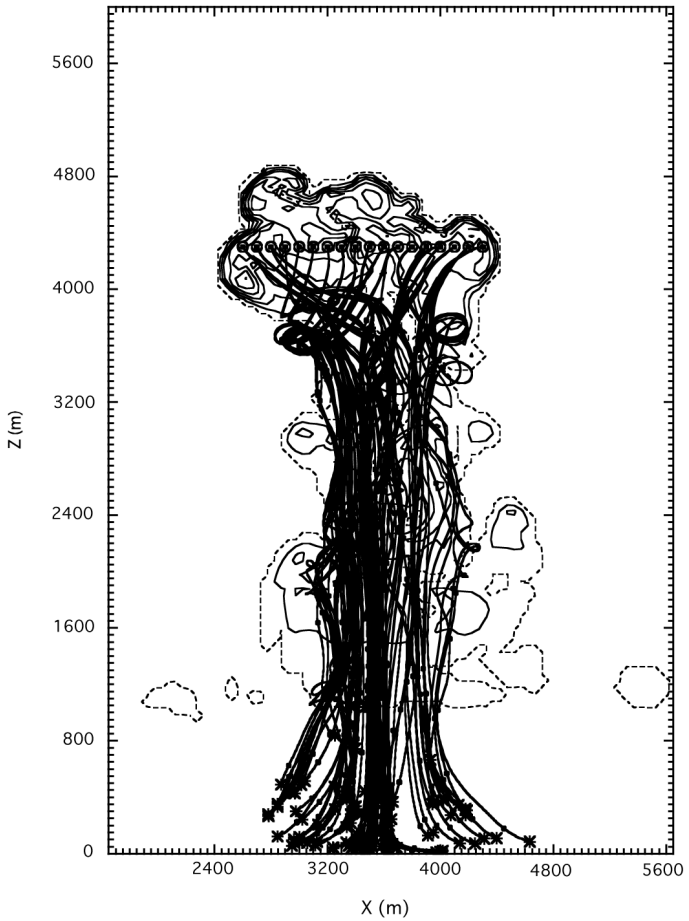


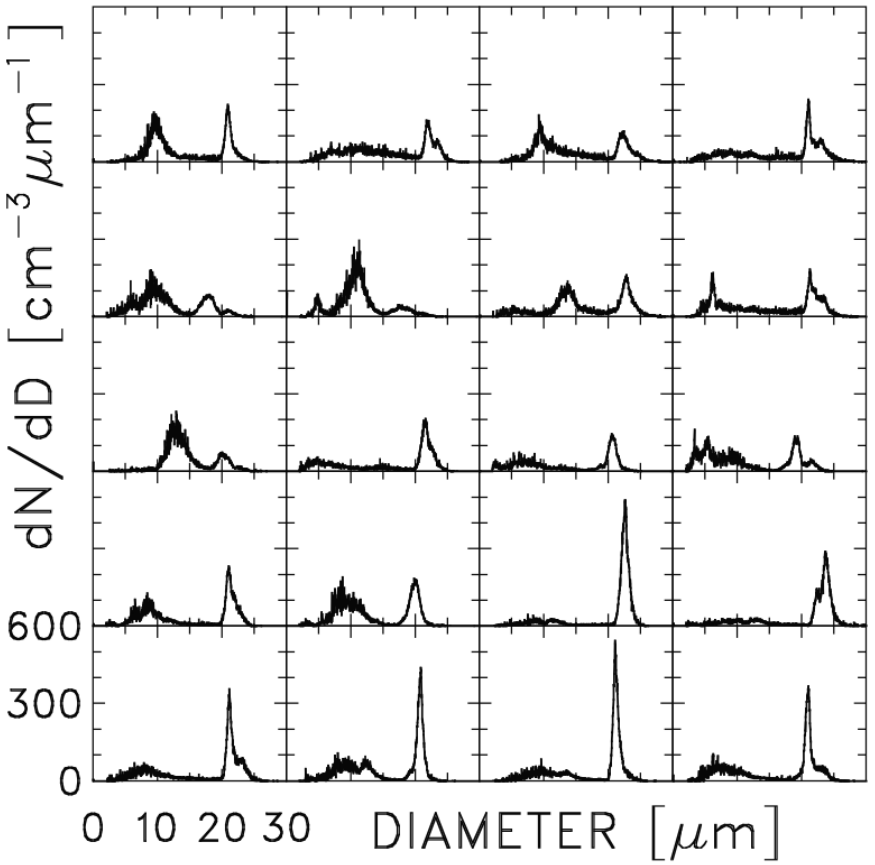


Stratocumulus vertical profiles



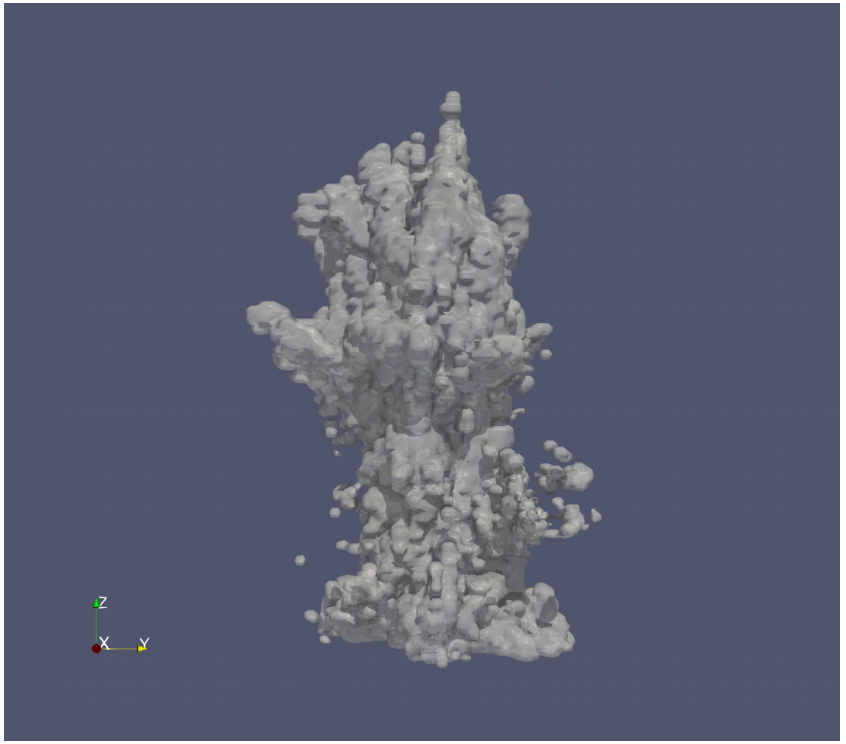
Advantages of particle-based microphysics: Cumulus simulations

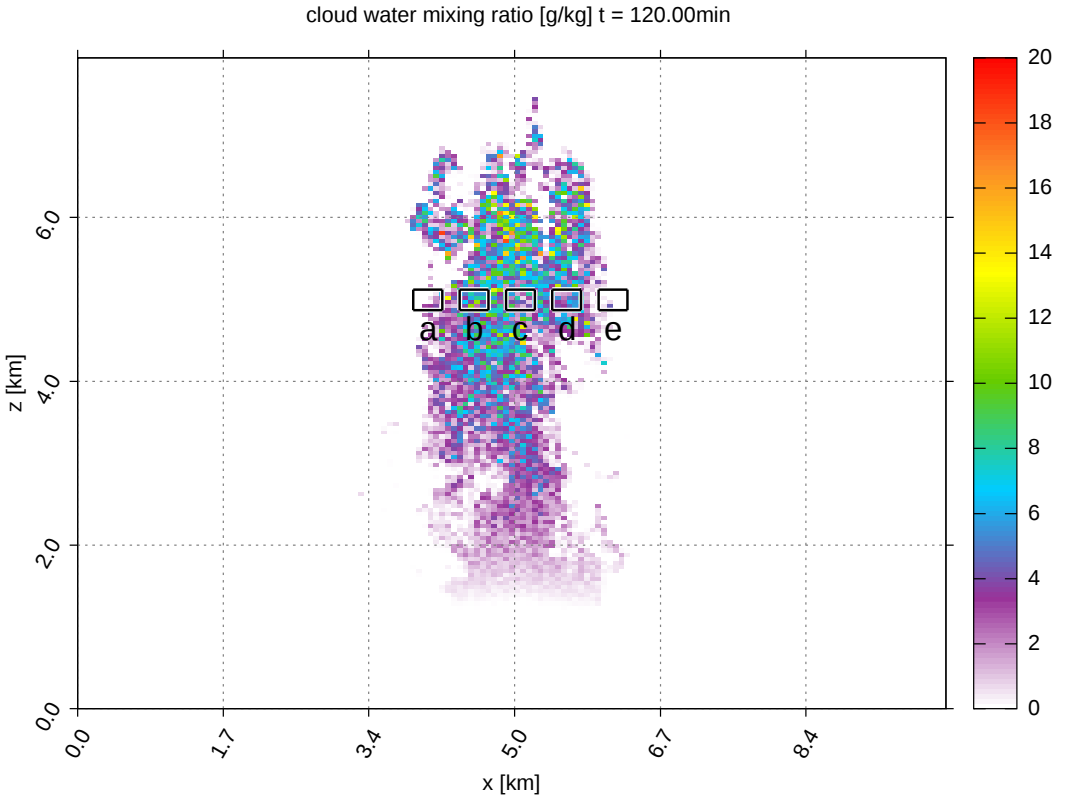


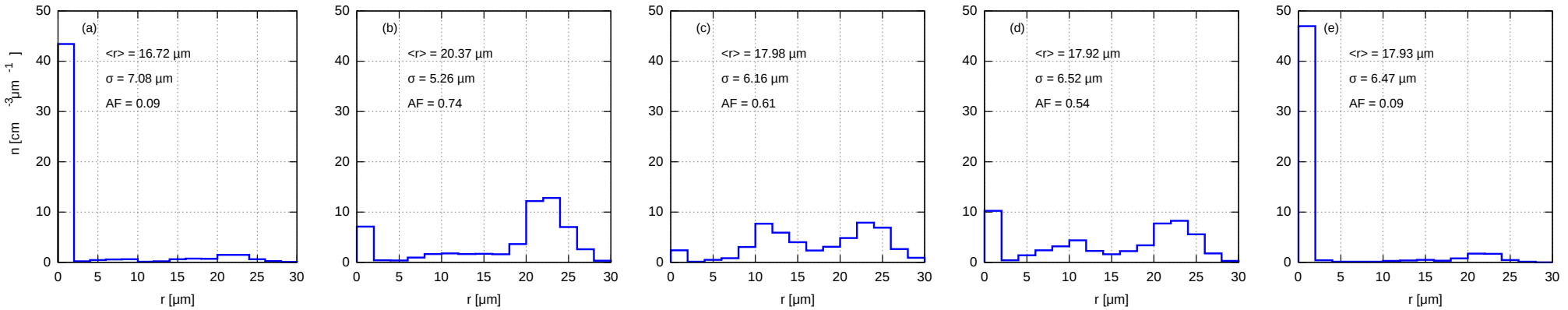


reproduced from Lasher-Trapp et al. *Q. J. R. Met. Soc.* 2005

UWLCM cumulus results







Conclusions

- Particle-based Lagrangian microphysics used in LES give stratocumulus results in agreement with bulk and bin microphysics, except for lower precipitation.
- Implicit LES is in agreement with LES with subgrid-scale models, with the exception of skewness of the vertical velocity distribution.
- Thanks to the use of GPUs, sophisticated microphysics do not slow down simulations.
- LES code available for everyone: https://github.com/igfuw/UWLCM

We have open PhD student positions!