Adaptive timestepping for condensational growth of droplets in numerical cloud models with Lagrangian microphysics

P. Dziekan, A. Makulska, H. Pawlowska

Institute of Geophysics, Faculty of Physics, University of Warsaw

Japanese collaborators: Shin-ichiro Shima's group, U. of Hyogo, Kobe

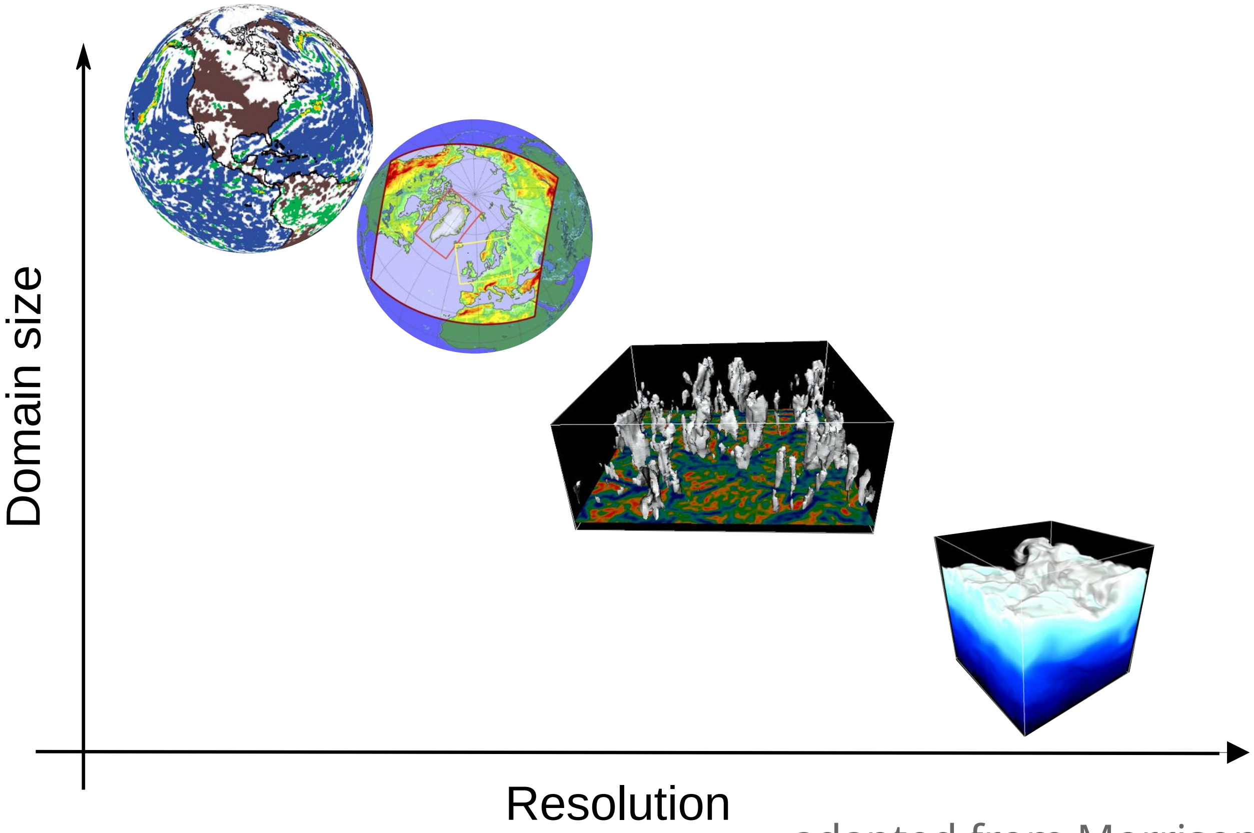
Agenda

1) Overview of the University of Warsaw Lagrangian Cloud Model (UWLCM)
2) Development of adaptive condensation substepping in UWLCM
3) Other UWLCM work related to Hanami

Agenda

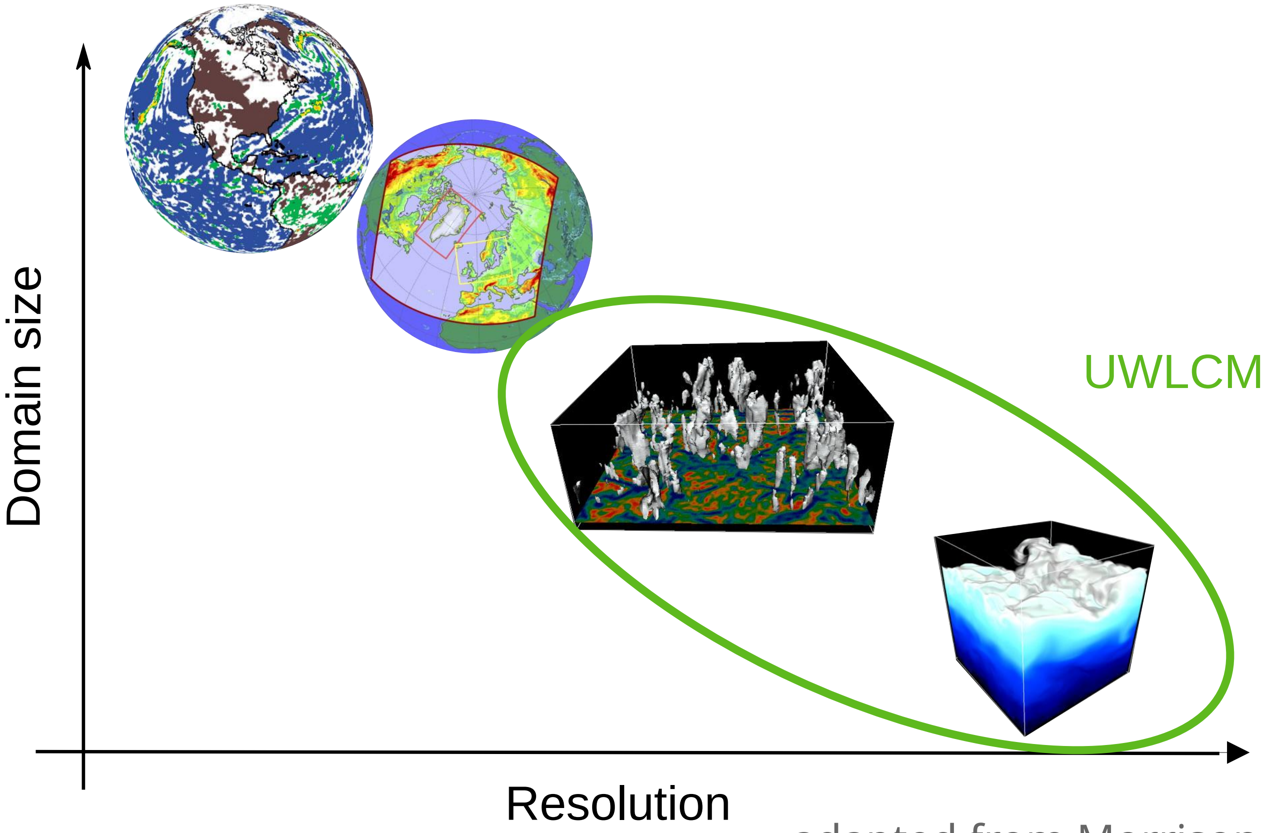
1) Overview of the University of Warsaw Lagrangian Cloud Model (UWLCM)

Cloud modeling across scales



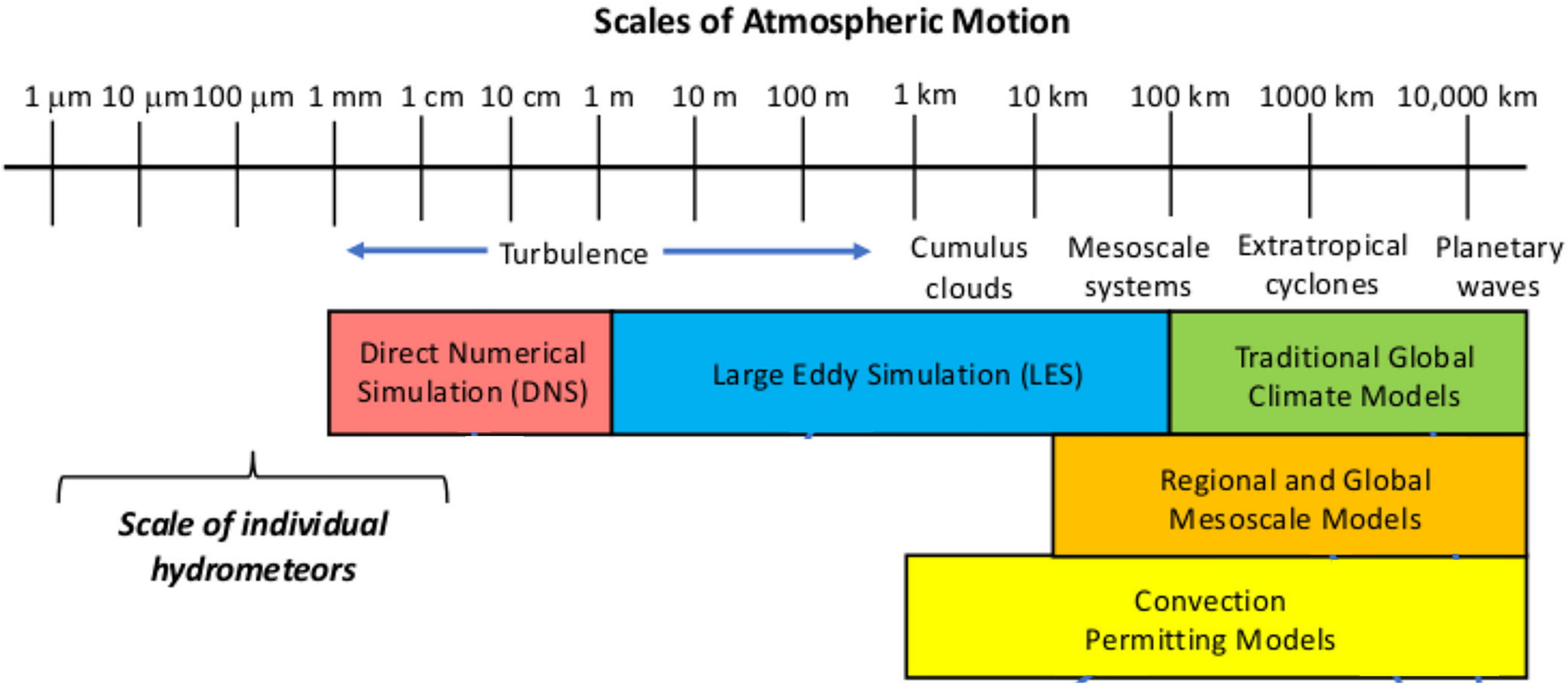
adapted from Morrison et al. *JAMES* (2020)

Cloud modeling across scales



adapted from Morrison et al. *JAMES* (2020)

Cloud modeling across scales



adapted from Morrison et al. *JAMES* (2020)

LES use cases

- Basic research in cloud physics.
- Improvement of parameterizations used in weather and climate models.
- Predictive models are starting to use resolutions close to LES (e.g. project NextGEMS). Methods developed for LES will be used directly in global models.

University of Warsaw Lagrangian Cloud Model (UWLCM)

- Tool for large eddy simulations (LES) of clouds
- Sophisticated cloud microphysics model – super-droplet method (SDM)
- Developed for 10+ years
- Written in C++ (Boost, blitz++, Python bindings)
- Hybrid OpenMP + MPI parallelization
- Can use GPUs (CUDA Thrust) to model microphysics
- Open-source: github.com/igfuw/UWLCM

UWLCM basics

- Modeling air flow (2d or 3d):
  - Large eddy simulations: small-scale turbulence is parameterized

UWLCM basics

- Modeling air flow (2d or 3d):
  - Large eddy simulations: small-scale turbulence is parameterized
- Modeling temperature and water vapor

UWLCM basics

- Modeling air flow (2d or 3d):
  - Large eddy simulations: small-scale turbulence is parameterized
- Modeling temperature and water vapor
- Modeling liquid water
  - Bulk microphysics

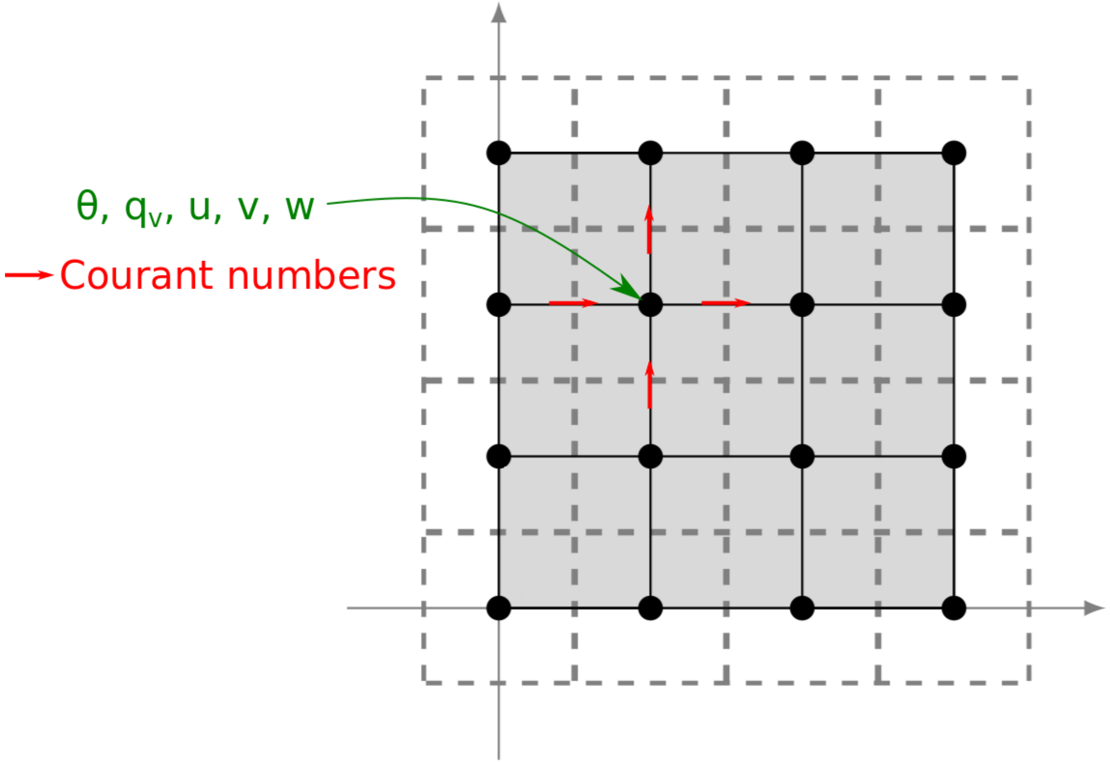
UWLCM basics

- Modeling air flow (2d or 3d):
  - Large eddy simulations: small-scale turbulence is parameterized
- Modeling temperature and water vapor
- Modeling liquid water
  - Bulk microphysics

Eulerian

Eulerian variables

- Governed by anelastic equations
- Staggered rectangular grid with stretching
- Solved with MPDATA



UWLCM basics

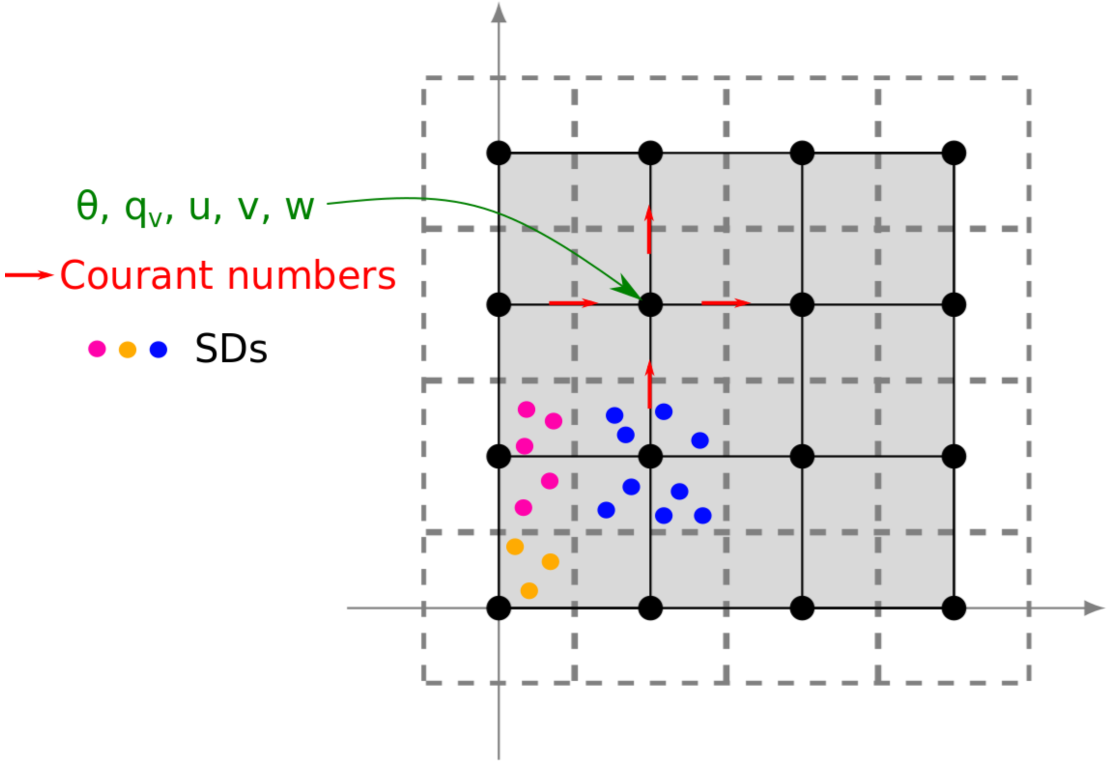
- Modeling air flow (2d or 3d):
  - Large eddy simulations: small-scale turbulence is parameterized
- Modeling temperature and water vapor
- Modeling liquid water
  - Bulk microphysics
  - Super-droplet method

Eulerian

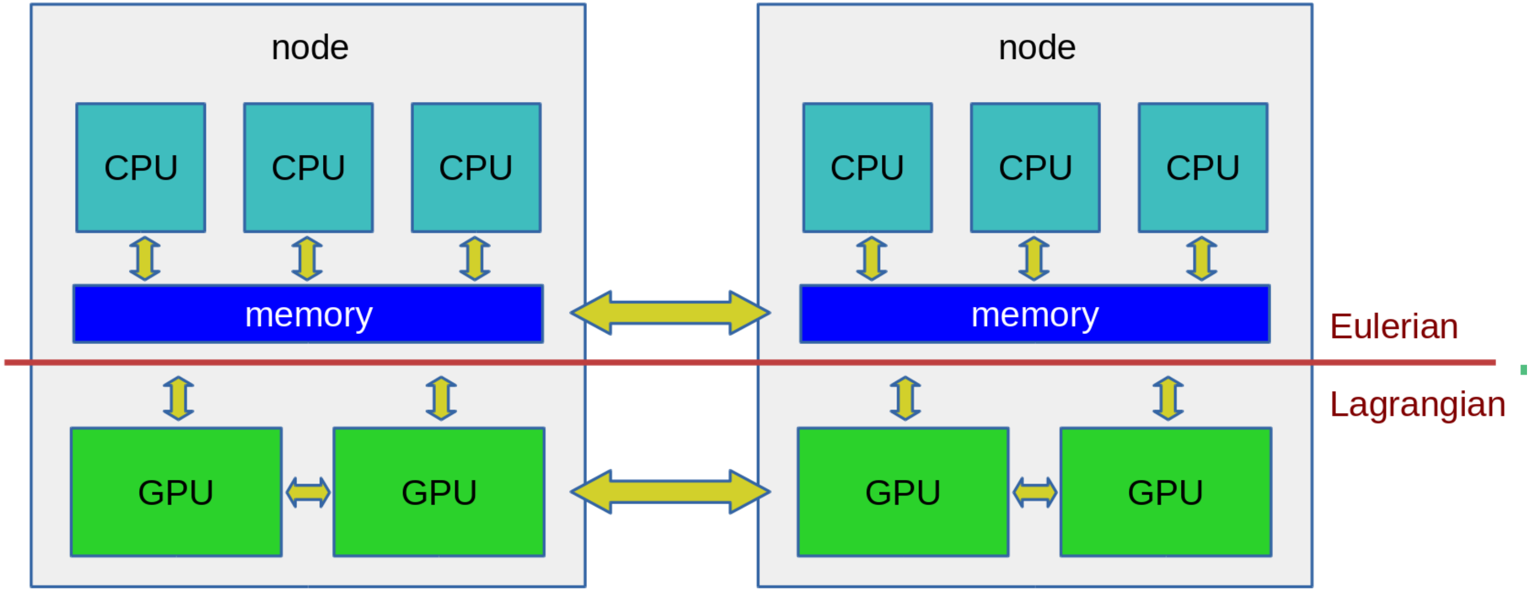
Lagrangian

Super-droplets (SD)

- Computational particle-like objects called super-droplets represent:
  - Humidified aerosols
  - Cloud droplets
  - Rain drops
- Each super-droplet represents lage number of identical real hydrometeors

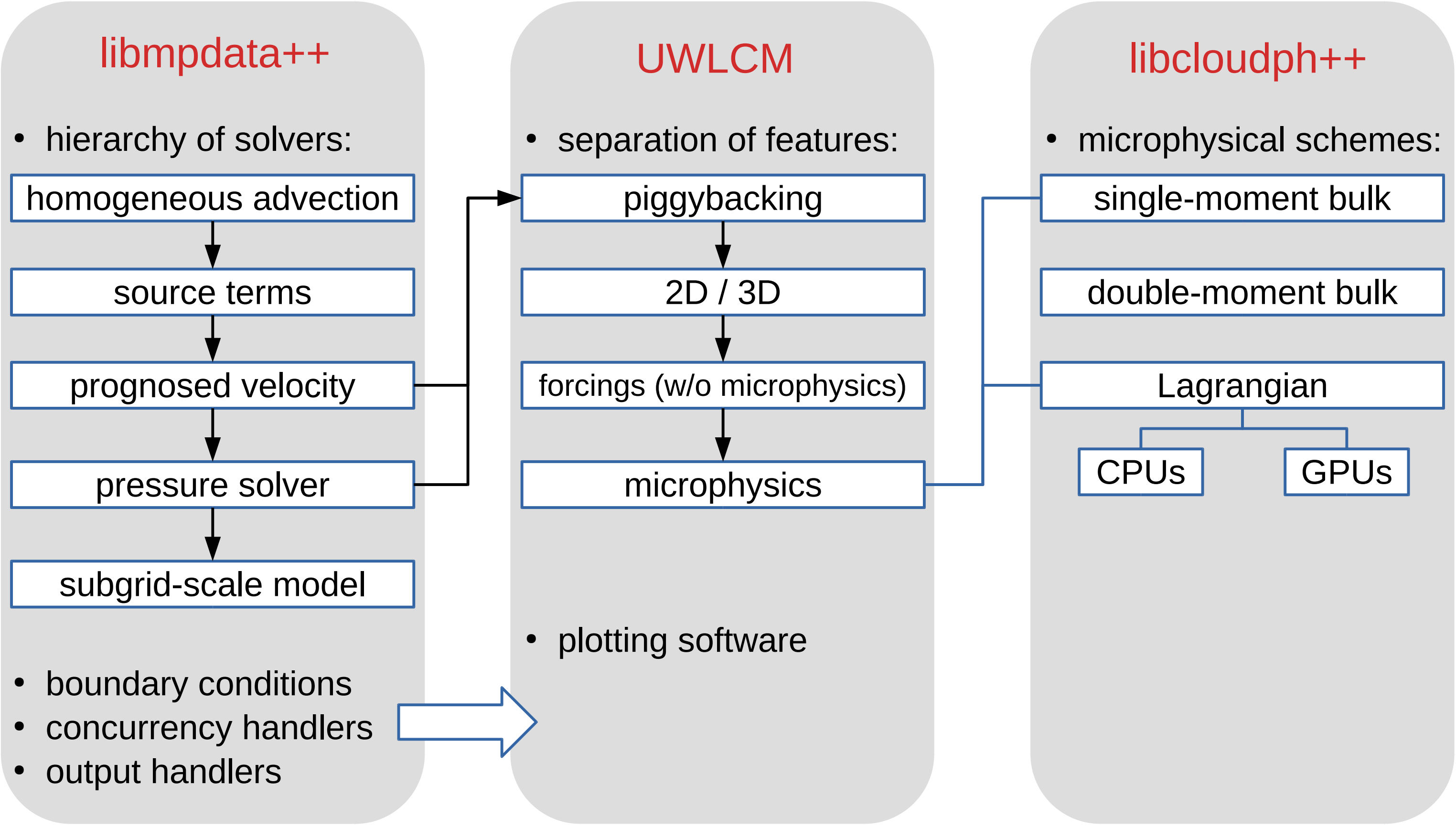


Use of heterogeneous (CPU+GPU) clusters



- Eulerian component: resides in RAM, computed by CPUs
- Lagrangian component: resides in GPU RAM, computed by GPUs
- Simultaneous computation of Eulerian and Lagrangian components

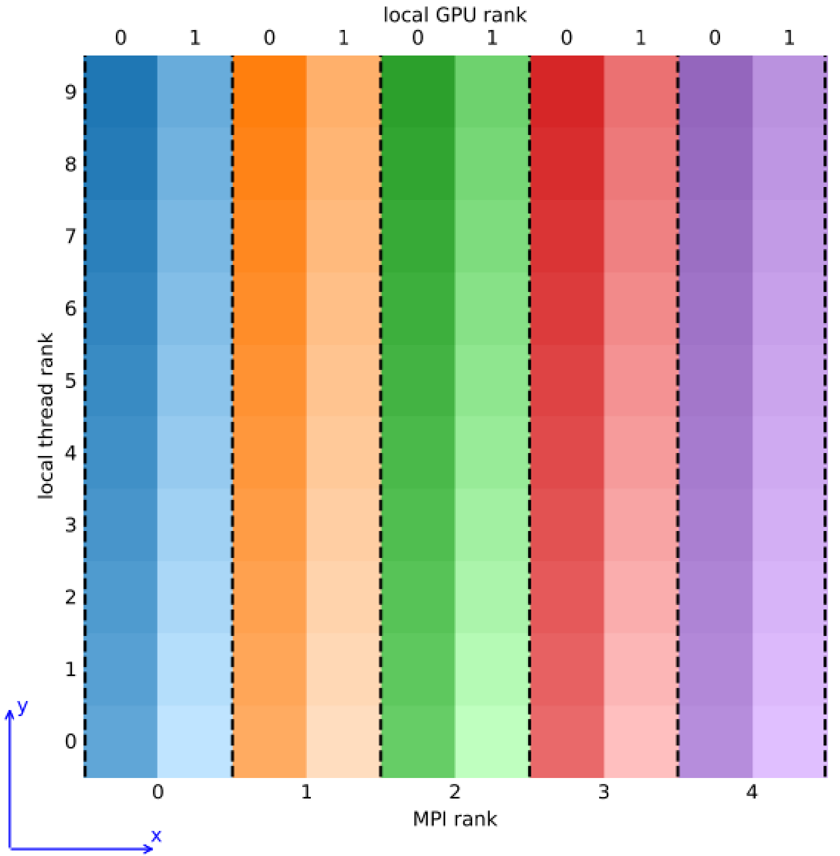
Modular code structure



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

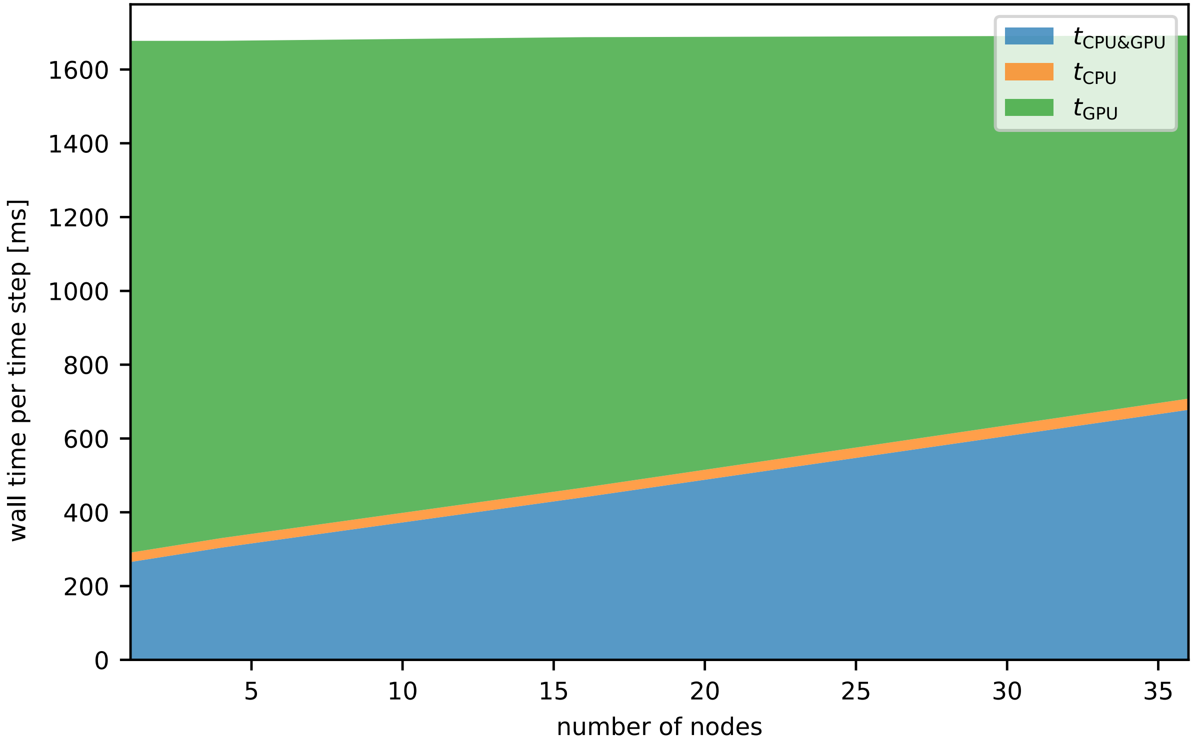
- version control system
- automated tests
- open-source code hosted on github

Domain decomposition



Top-down view of modeled domain; squares are Eulerian grid cells; coloring shows MPI, thread and GPU ranks.

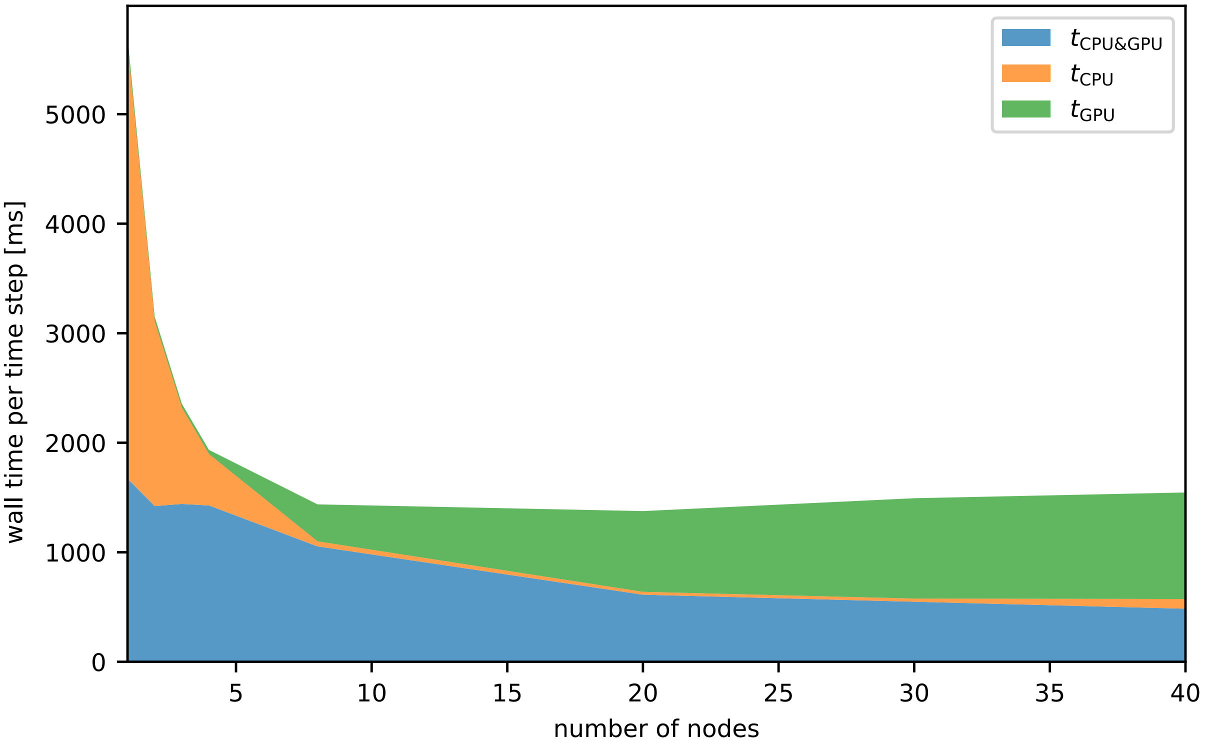
Weak scaling test



Wall time per time step vs number of nodes. Timings of simultaneous CPU and GPU computations (blue), CPU-only computations (orange) and GPU-only computations (green) are stacked.

- GPU time scales better than CPU time
- Simultaneous CPU and GPU usage should be maximized for an optimal number of nodes (larger than shown)
- Up to the optimal number of nodes, scaling efficiency of the total wall time is ca. 100%

Strong scaling on CPU, weak on GPU



Wall time per time step vs number of nodes. Timings of simultaneous CPU and GPU computations (blue), CPU-only computations (orange) and GPU-only computations (green) are stacked.

- Good balance of CPU and GPU computations (ca. 80%) for an optimal number of nodes (5-10 in this case)

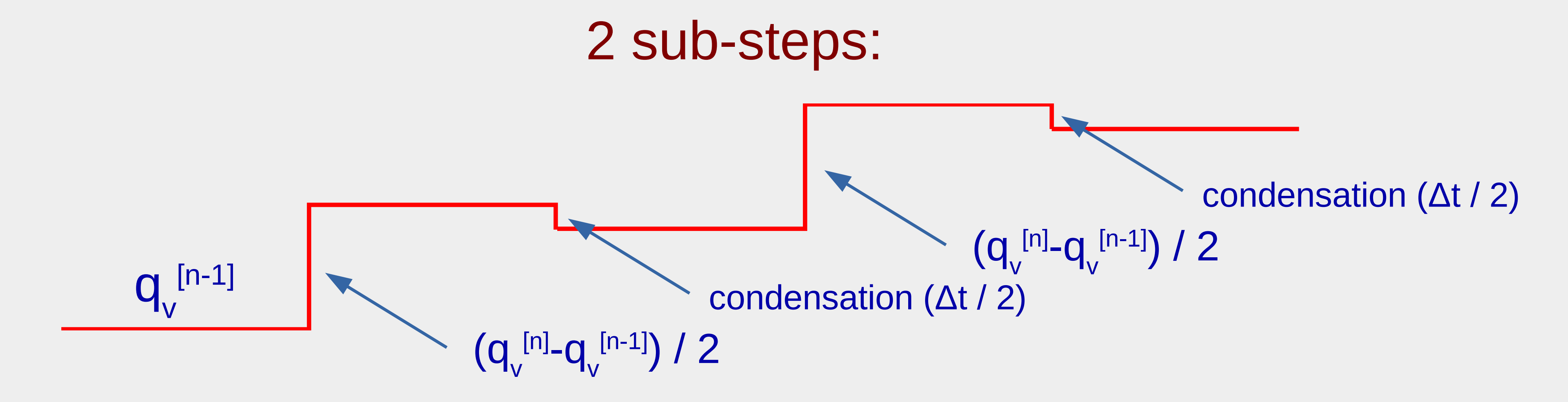
Agenda

1) Overview of the University of Warsaw Lagrangian Cloud Model (UWLCM)
2) Development of adaptive condensation substepping in UWLCM

Need for substepping

- Model timestep: ~1s
- Processes that need to be modeled with shorter timesteps: condensation, collision-coalescence, ...
- Change of radius due to condensation is modeled using an implicit-explicit scheme solved with a predictor-corrector method. Converges for a ~0.1s timestep.
- Substepping: multiple timesteps for specific processes per model timestep.

Condensation substepping

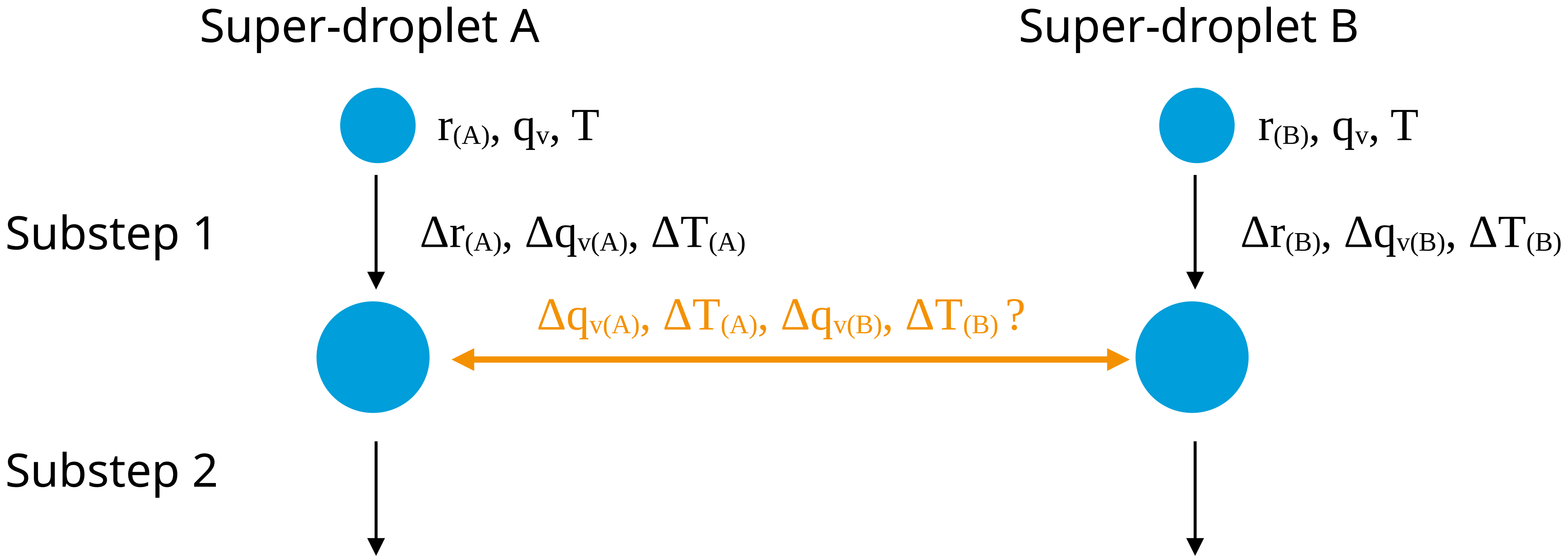


- $q_v^{[n]}$ – water vapor at n-th model time step
- Δt – model time step length
- Change in thermodynamic conditions (q_v, T) is incrementally applied in substeps

Adaptive condensation substepping

- Inspired by: Matsushima T. et al. GMD (2023), Bartman P. Msc thesis (2020)
- Per-cell vs <u>per-superdroplet</u>?
- Number of substeps can vary between superdroplets in a cell
- Number of substeps changes between time steps

The issue of „mixing“



The issue of „mixing“

- Mean distance between droplets: ~2 mm
- Turbulent mixing timescale at a 2 mm distance: ~0.1 s
- Turbulent mixing timescale at a 1 cm distance: ~0.5 s
- No clear answer if it’s better to „mix“ (homogenize) cell every substep (~0.1s) or every step (~1s), so we choose the latter in adaptive substepping.
- Preliminary results don’t show much sensitivity to mixing.

Adaptation strategy

1. $\Delta r^2(\Delta t)$
2. $\Delta r^2(\Delta t/2)$
3. $\Delta r^2(\Delta t/4)$
4. …

r – droplet radius

Stop condition:

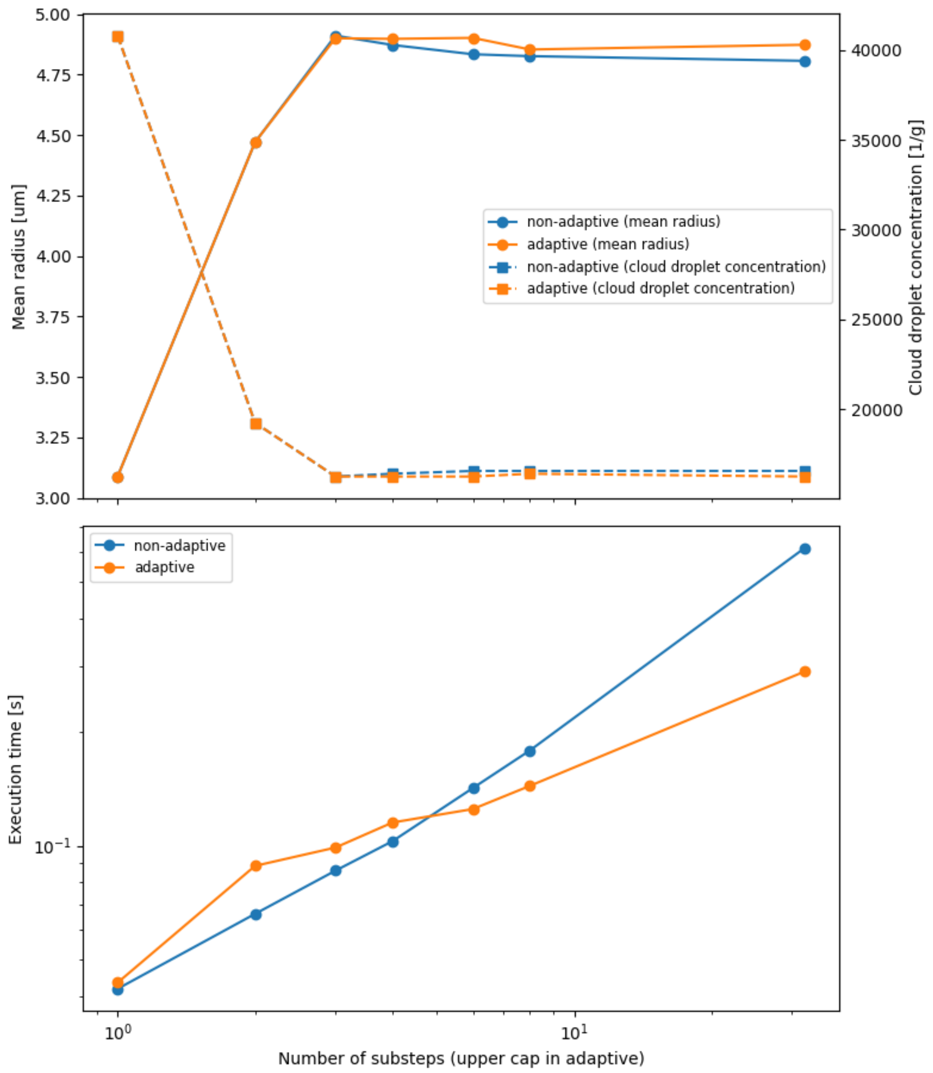
$\Delta r^2(\Delta t') - 2\,\Delta r^2(\Delta t' / 2) < \varepsilon_{\Delta r}\, r^2$; $\varepsilon_{\Delta r}=10^{-3}$

and

$\Delta r^2(\Delta t') < \varepsilon_{sr}\, r^2$; $\varepsilon_s=2$

Box model test

- Condensation in a single cell
- Non-adaptive vs adaptive
- Similar results for adaptive and non-adaptive
- Results converge for #substeps > 5
- Adaptation gives speedup only for #substeps > 5

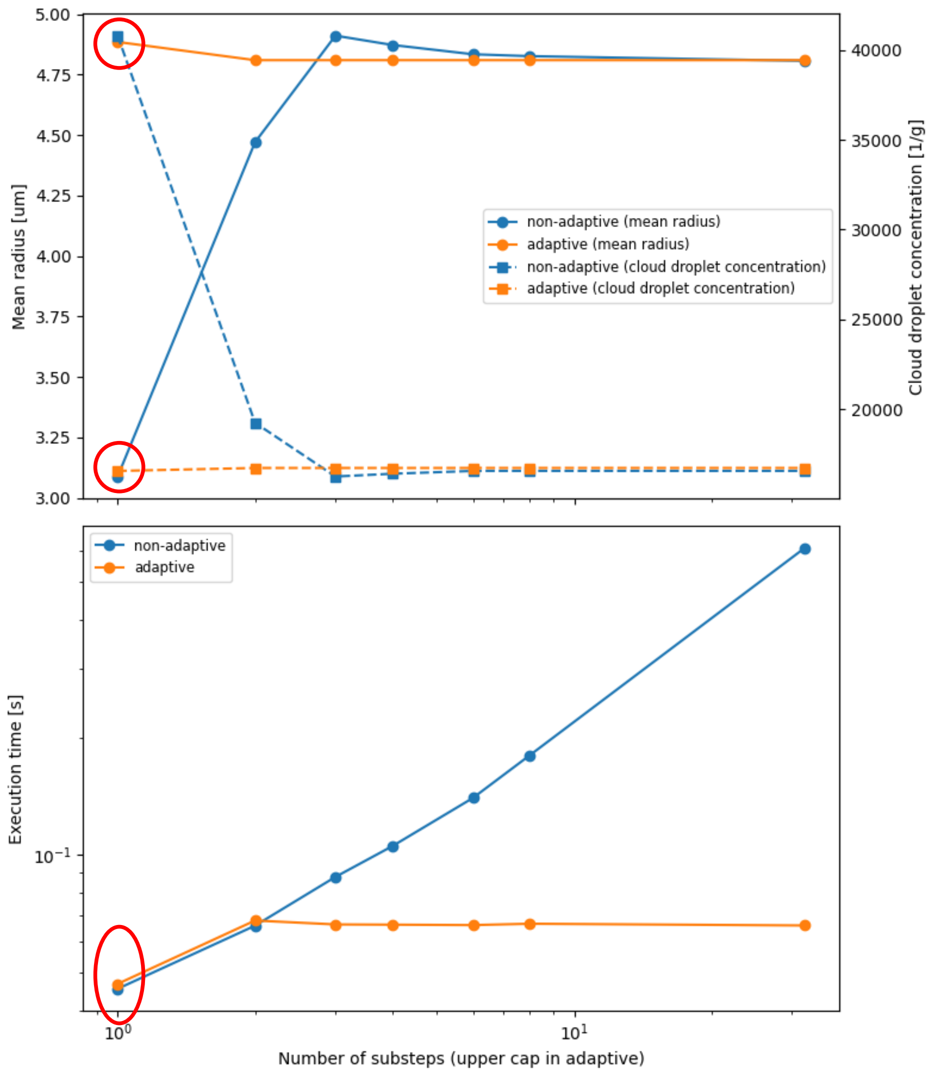


Adaptation for activation

- Short timesteps are mostly required to resolve droplet activation (formation of new cloud droplets).
- Additional condition: if a droplet is to (de)activate: $r^2 < r_c^2$, $r^2 + \Delta r^2(\Delta t) > r_c^2$, use a fixed number of substeps (8).
- The critical radius r_c weakly depends on thermodynamic conditions, so it can be calculated only once.

Box model test

- Activation adaptation used.
- Correct results even if substeps are done only for activating droplets.
- 3x speedup in modeling condensation (possibly more in more realistic simulations).
- Computing condensation can take up to 40% of simulation time.



Adaptation: next steps

- Test in a realistic cloud simulation.
- Test on GPUs.

Agenda

1) Overview of the University of Warsaw Lagrangian Cloud Model (UWLCM)
2) Development of adaptive condensation substepping in UWLCM
3) Other UWLCM work related to Hanami

Benchmarking UWLCM on Fugaku

- Non-MPI runs work.
- Work in progress on a Singularity image with MPI.

Ice microphysics in UWLCM

- Inspired by Shima et al. GMD (2020).
- Implemented processes:
  - Ice nucleation and melting (singular and time-dependent).
  - Deposition and sublimation.
- Tested in parcel model.
- Ongoing work on using it in an Arctic stratocumulus simulation.