University of Warsaw Lagrangian Cloud Model (UWLCM): LES with particle microphysics

P. Dziekan, S. Arabas, M. Waruszewski, A. Jaruga, P. Zmijewski, H. Pawlowska

University of Warsaw, Warsaw, Poland

Agenda

1) Model overview
2) Particle microphysics
3) Fractal grid refinement
4) Software design
5) Computational performance

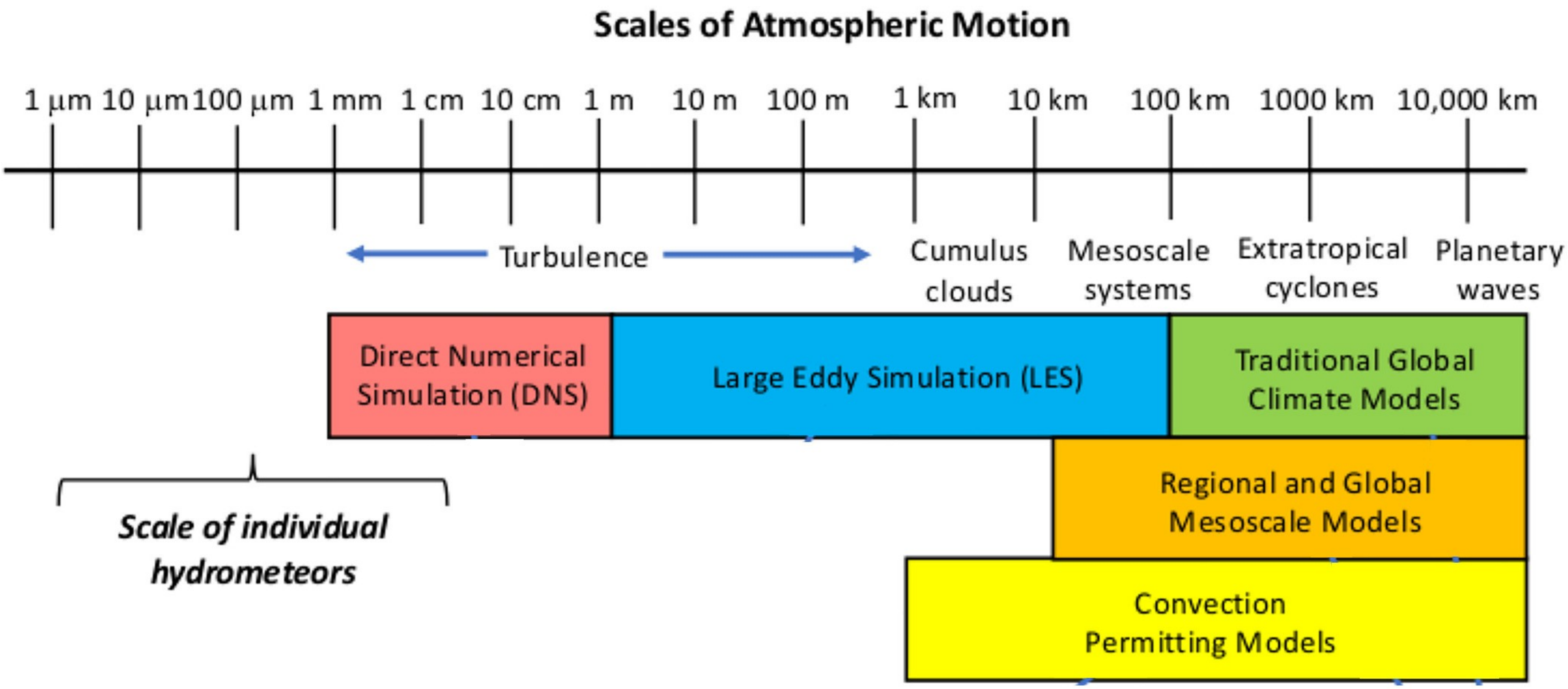
Agenda

1) Model overview

UWLCM basics

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

LES spatial scale



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

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

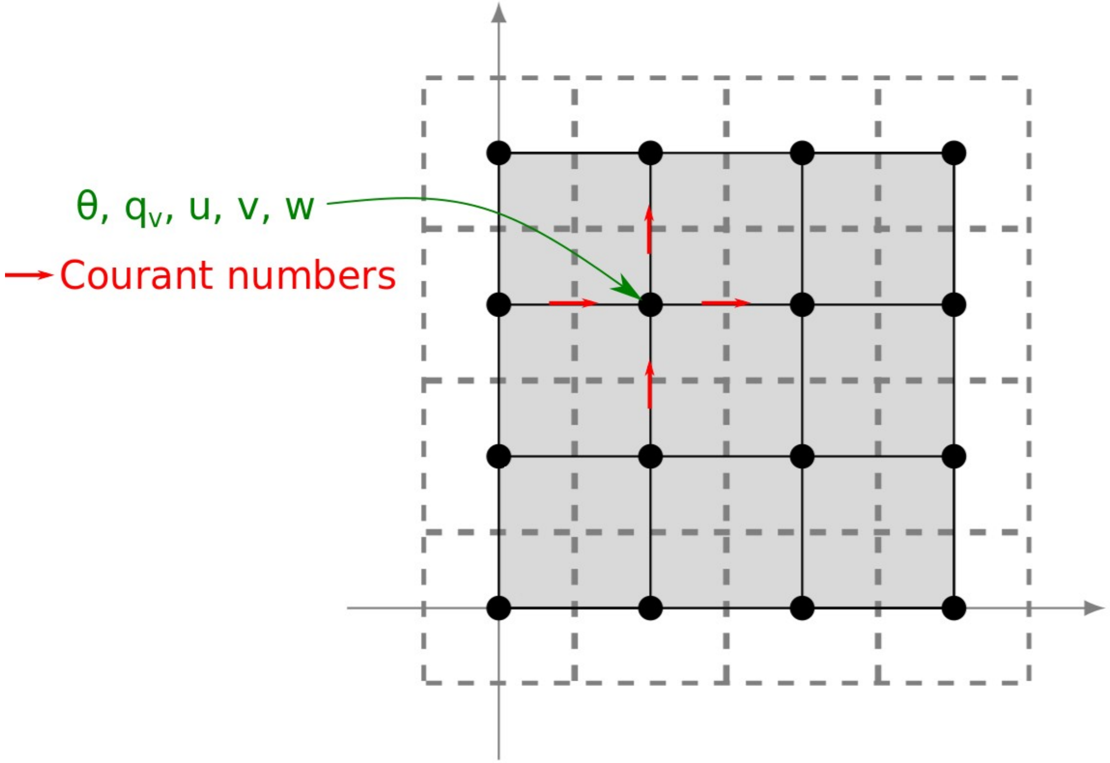
UWLCM basics

- 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

- Staggered rectangular grid with stretching
- Governed by anelastic equations
- Solved with MPDATA
- Subgrid-scale models:
  - Smagorinsky
  - Implicit LES



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
  - Particle microphysics (super-droplet method)

Eulerian

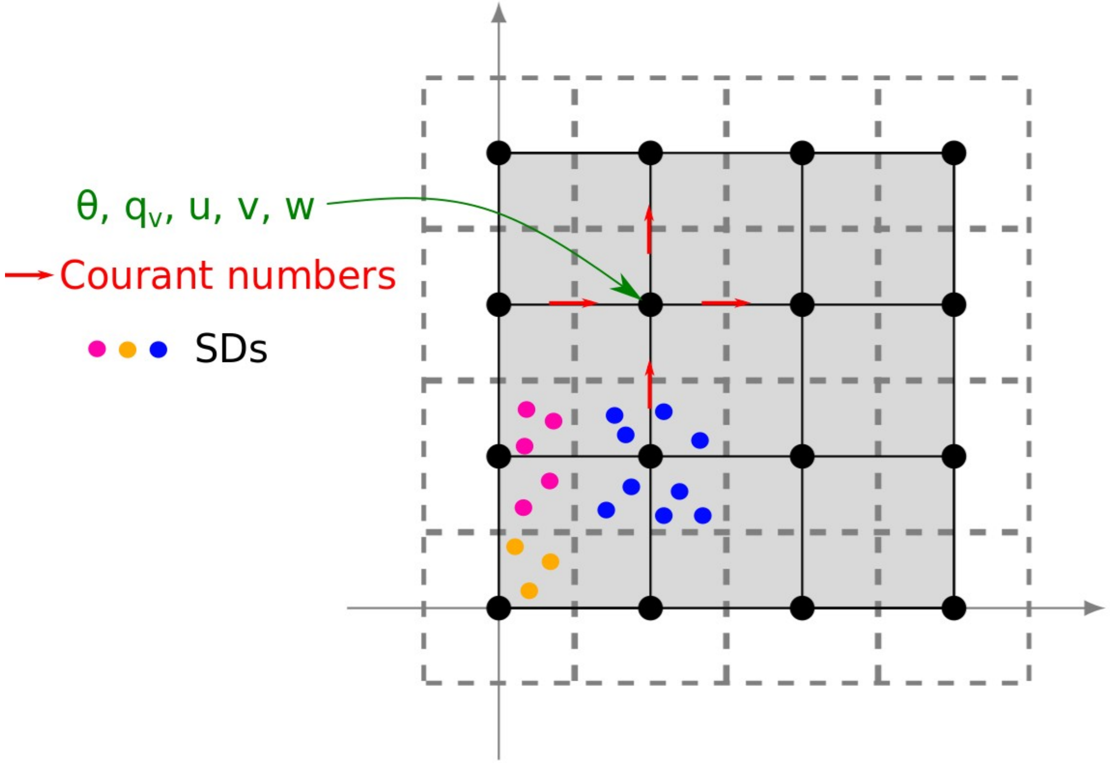
Lagrangian

Agenda

1) Model overview
2) Particle microphysics

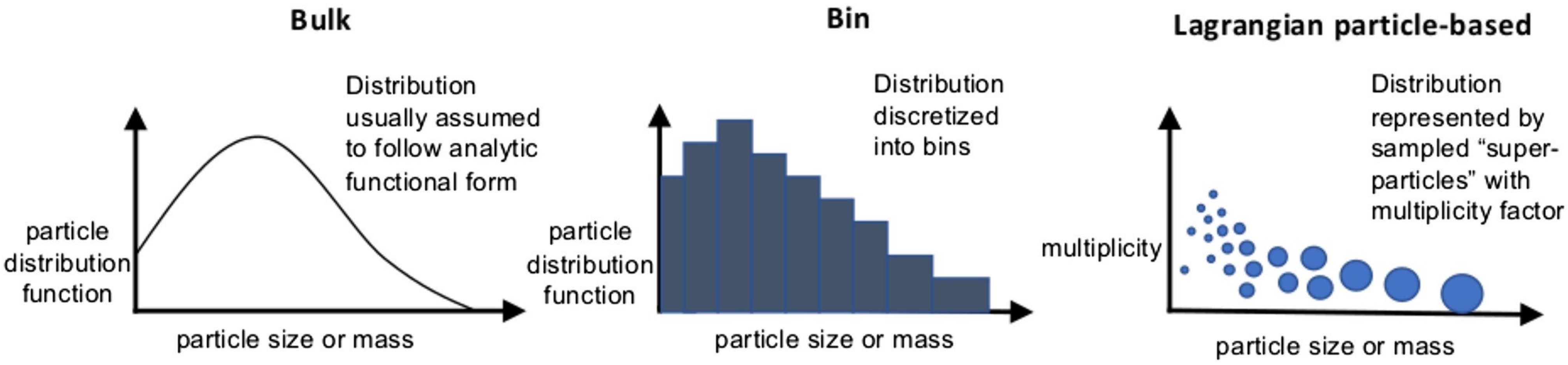
Super-droplets (SD)

- Computational particle-like objects called super-droplets represent:
  - Humidified aerosols
  - Cloud droplets
  - Rain drops



SD: Droplet size distribution

- Each SD represent multiple real hydrometeors (multiplicity) with same properties (e.g. radius)
- Evolution of the DSD is resolved, like in bin microphysics



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

SD pros

- Easy to add new attributes
  - Chemistry
  - Aerosol hygroscopicity
- Scales down to DNS
- No numerical diffusion of droplet size distribution
- Stochastic collision-coalescence
- Activation without parameterization
- Solute effect after activation
- Straightforwad aerosol processing

SD cons

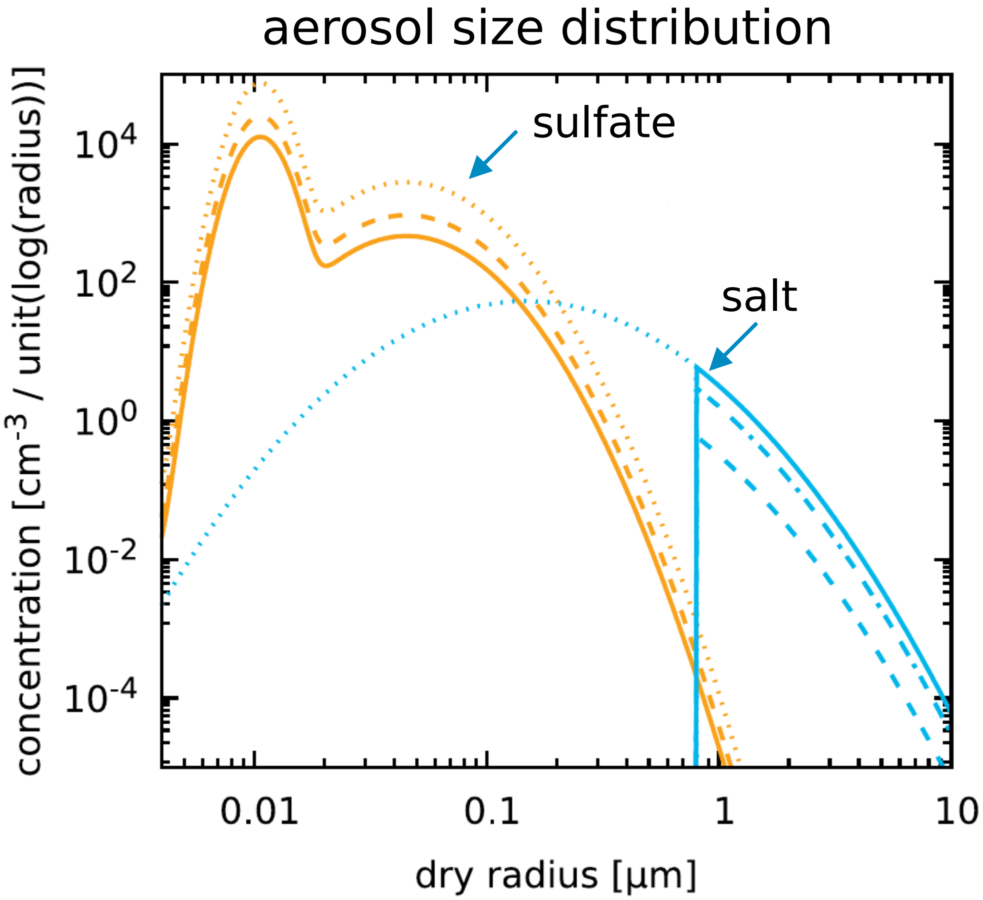
- Inconsitent results of collision-coalescence modeling with different implementations
- Increased spatial and temporal variances; requires averaging

SD use case 1: sea salt GCCN

- Solute effect is important for GCCN

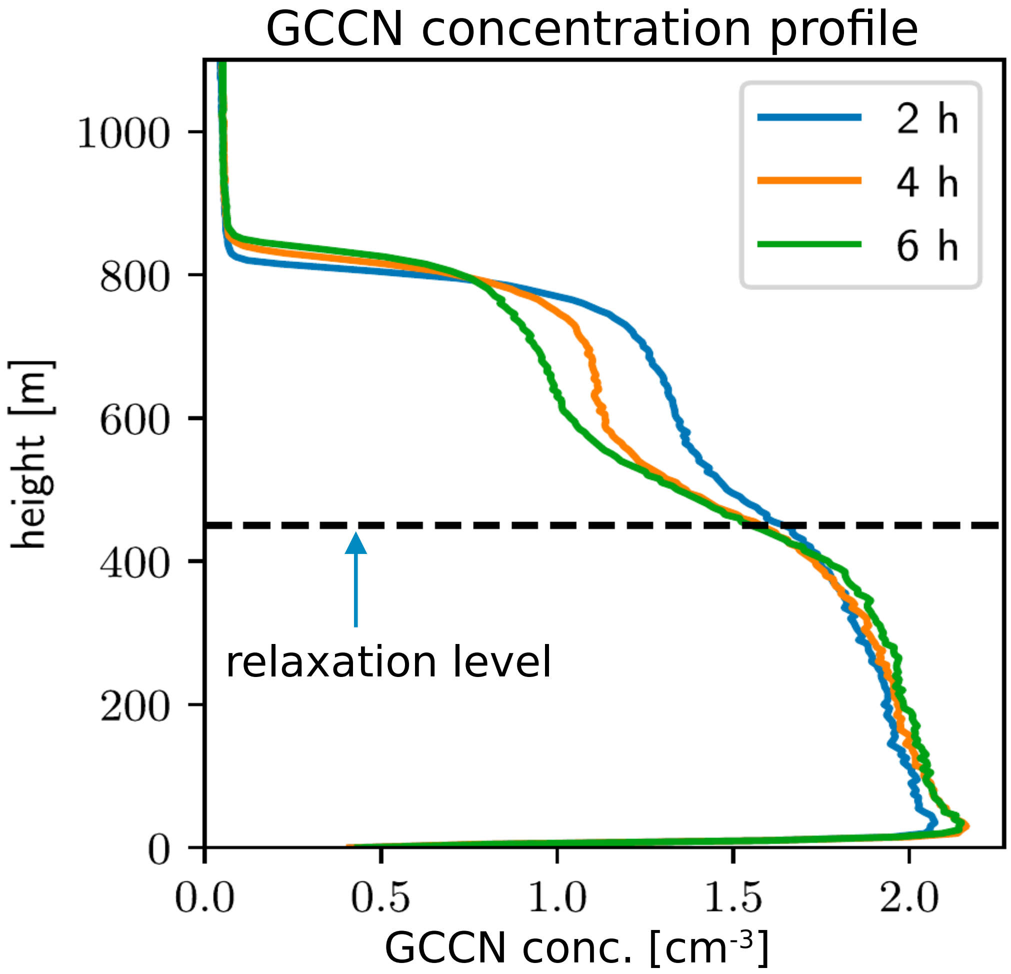
SD use case 1: sea salt GCCN

- Solute effect is important for GCCN
- Two types of aerosols: sulfate CCN and salt GCCN

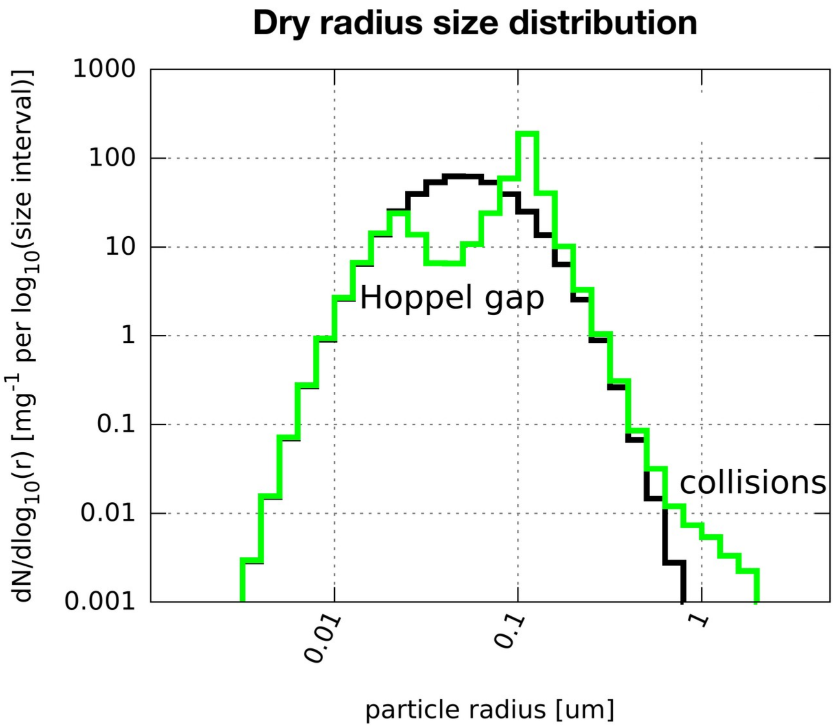


SD use case 1: sea salt GCCN

- Solute effect is important for GCCN
- Two types of aerosols: sulfate CCN and salt GCCN
- Strong GCCN washout and source near surface: relaxation



SD use case 2: chemistry



- Oxidation reaction is irreversible and affects the sizes of aerosol particles that served as condensation nuclei for cloud droplets
- As a result a gap in sizes between the activated and not activated aerosols is formed (Hoppel gap)
- Additionally a tail of larger aerosol particles is formed due to collisions between water drops

A. Jaruga

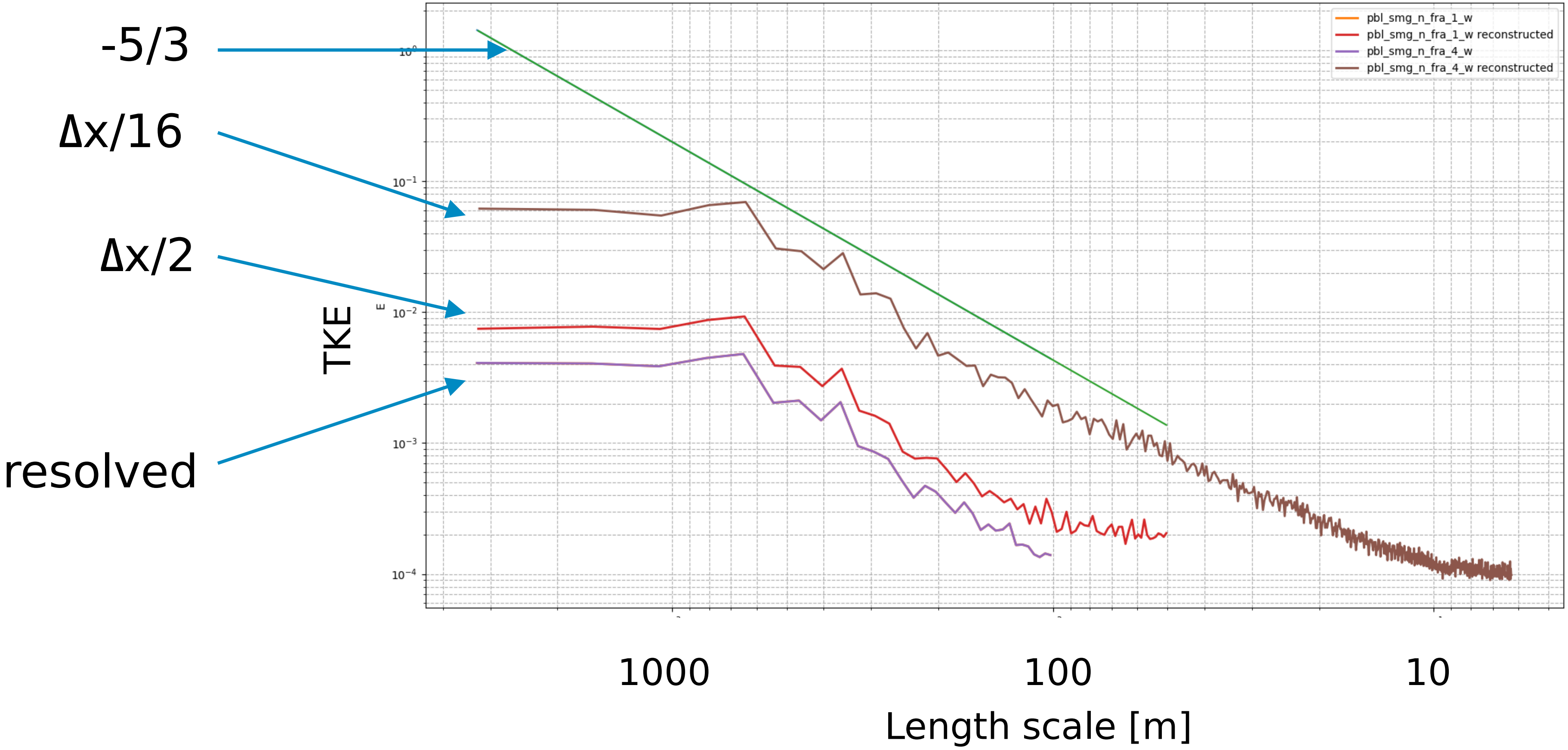
Agenda

1) Model overview
2) Particle microphysics
3) Fractal grid refinement

Reconstruction of SGS eddies

- Uses self-similarity of turbulence
- Applicable in the inertial range
- Each iteration doubles resolution, multiple iterations can be done

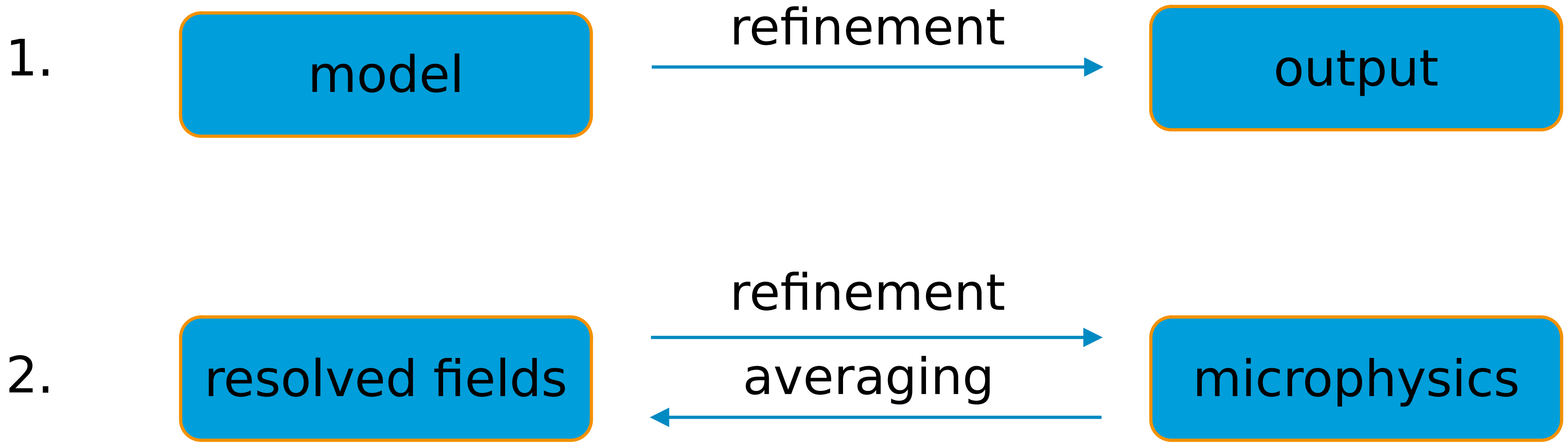
Reconstructed TKE spectrum



Grid refinement use cases

1.

Grid refinement use cases



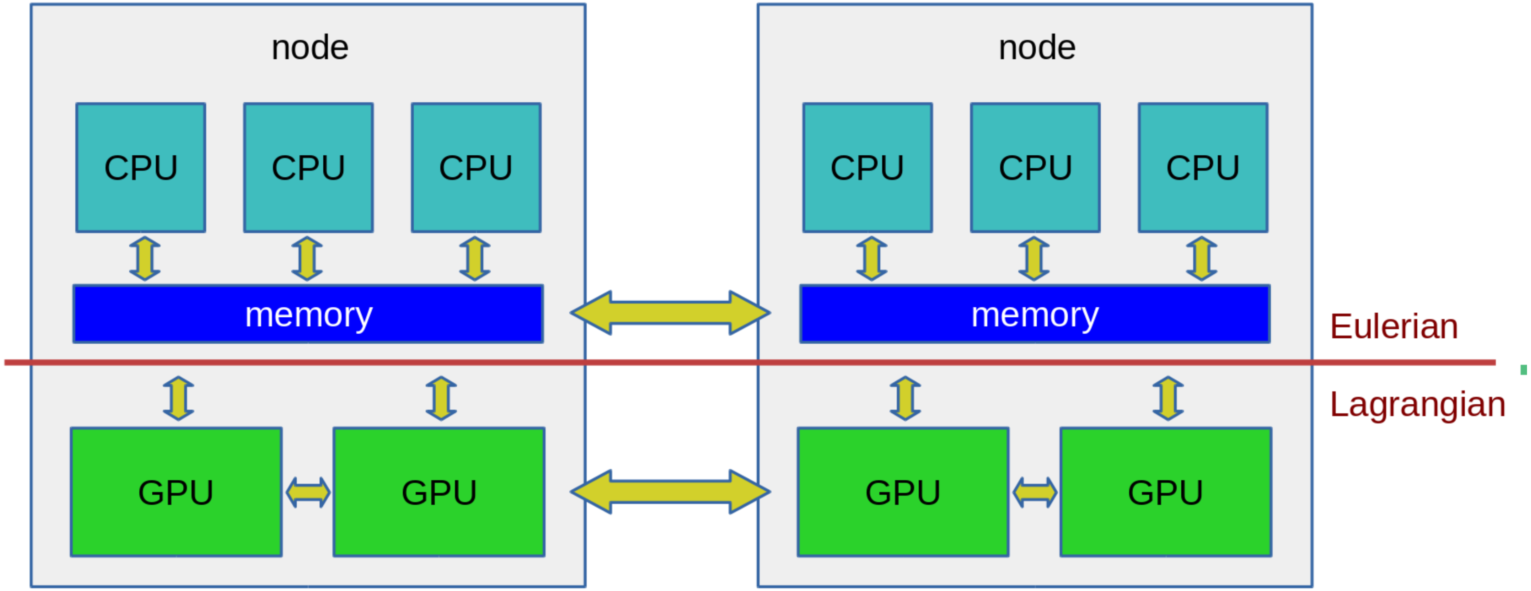
Agenda

1) Model overview
2) Particle microphysics
3) Fractal grid refinement
4) Software design

Design decisions

- Full use of heterogeneous (CPU+GPU) clusters

Heterogeneous cluster schematic



- Eulerian component: resides in RAM, computed by CPUs
- Lagrangian component: resides in RAM, computed by CPUs
  or resides in GPU RAM, computed by GPUs

Design decisions

- Full use of heterogeneous (CPU+GPU) clusters
- Modular code design

Modular code design

- UWLCM uses two libraries:
  - libmpdata++ - collection of solvers that use MPDATA
  - libcloudph++ - collection of microphysics models
- Family of solvers – inheritance from simple to complex
- Benefits:
  - Improves code reusability
  - Simplifies code maintenance

Design decisions

- Full use of heterogeneous (CPU+GPU) clusters
- Modular code design
- Easy to use – Singularity image, multiple runtime options

Multiple single-line options

runtime options:

- number of dimensions:
  ./uwlcm --ny=[0, X]
- type of microphysics:
  ./uwlcm --micro=[blk_1m, lgrngn]
- where to calculate Lagrangian microphysics:
  ./uwlcm --backend=[serial, OpenMP, CUDA, multi_CUDA]
- model setup:
  ./uwlcm --case=[dycoms, bomex, ...]
- piggybacking:
  ./uwlcm --piggy=1 --vel_in=file
- number of CPU threads for dynamics:
  OMP_NUM_THREADS=X ./uwlcm
- MPI processes:
  mpiexec -np X ./uwlcm
- 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]

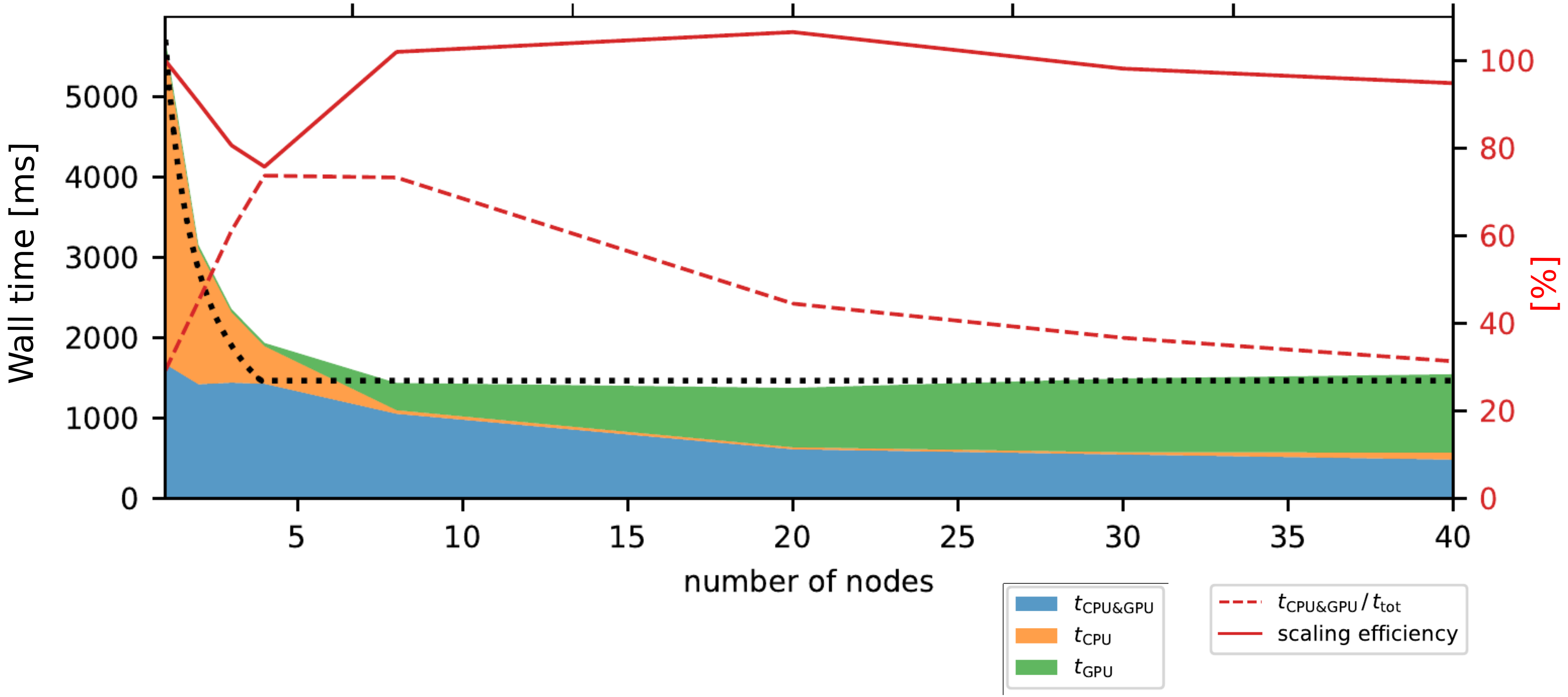
Design decisions

- Full use of heterogeneous (CPU+GPU) clusters
- Modular code design
- Easy to use – Singularity image, multiple runtime options
- Open-source code – hosted on Github
- C++ (CUDA and Python bindings in libcloudph++)
- use existing libraries (Boost, Thrust, blitz++, ...)
- Hybrid OpenMP + MPI parallelization
- Automated tests – Github Actions

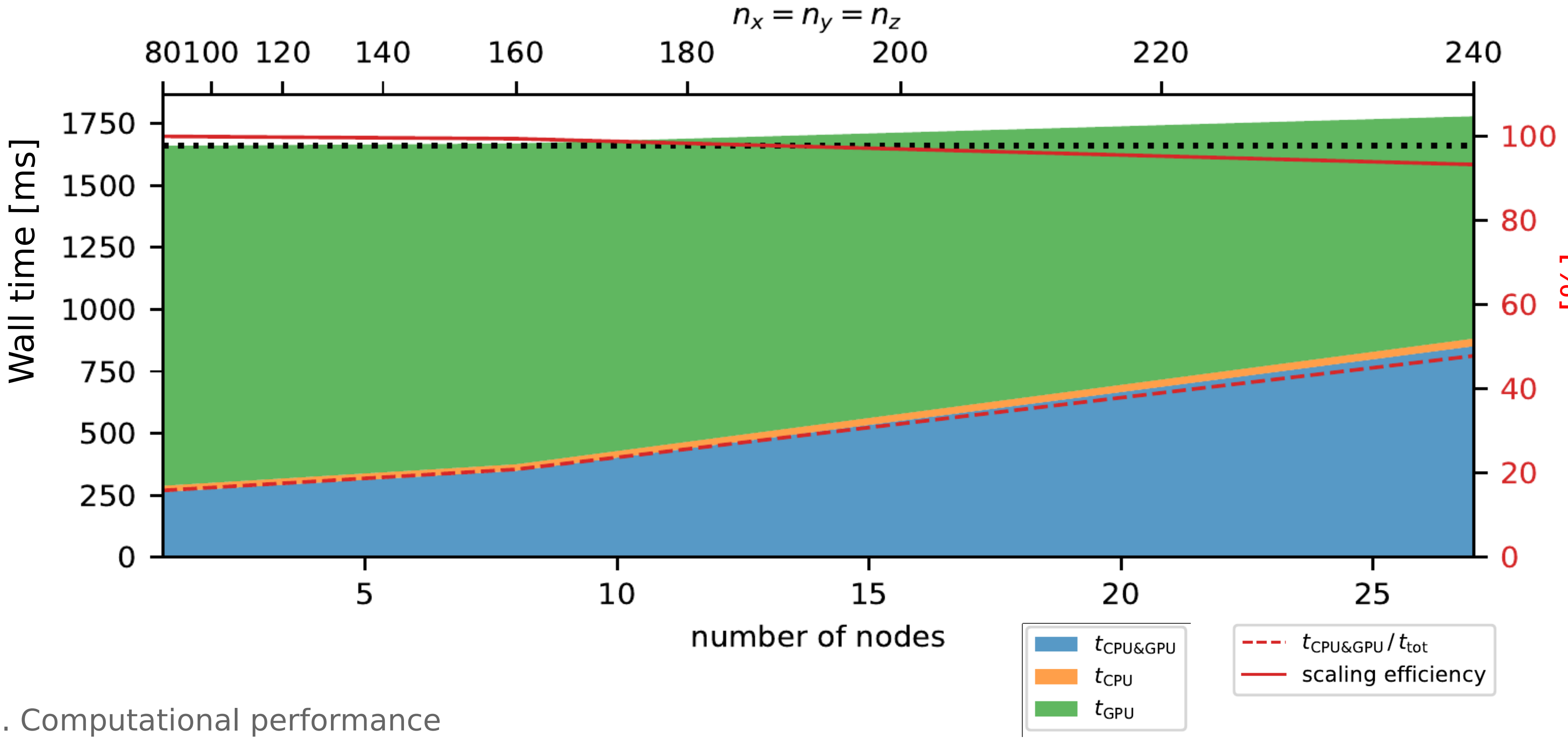
Agenda

1) Model overview
2) Particle microphysics
3) Fractal grid refinement
4) Software design
5) Computational peformance

CPU strong scaling, GPU weak scaling



Weak scaling



Summary

- UWLCM is a tool for LES modeling of clouds with a focus on detailed representation of cloud microphysics
- UWLCM in nextGEMS
  - Simulations of EUREC4A domain
    - Setup: open / periodic boundaries? EUREC4A-MIP?
    - Comparison with ICON-LES
  - Other ideas?
    - Warm cloud parameterization?