Adaptive Time-Stepping within the Super-Droplet Method Monte-Carlo Coagulation

UW IGF Seminar May 16th, 2025

Emma Ware

P.Bartman (Jagiellonian), S. Arabas (AGH)

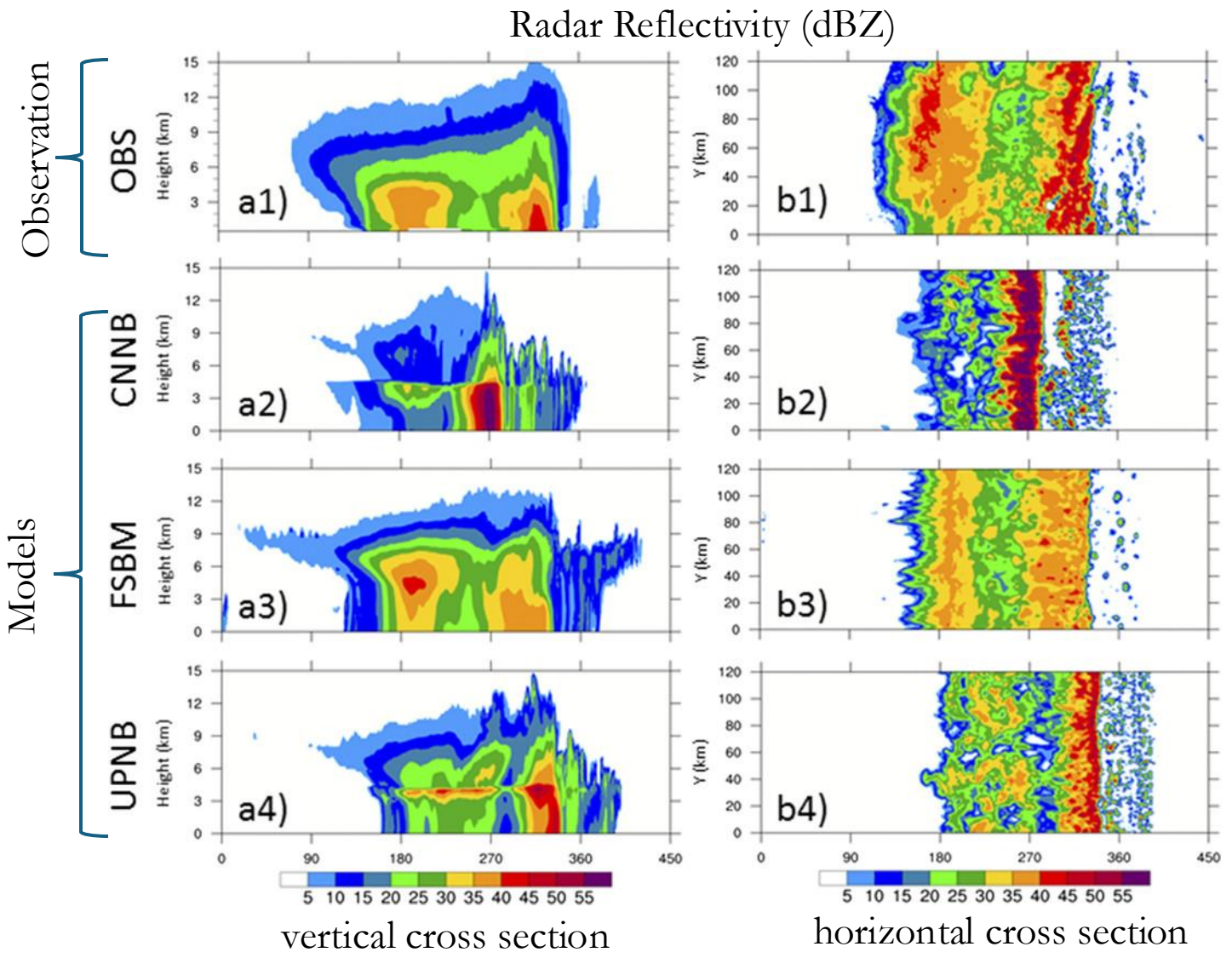
- **Background on Cloud Processes**
- Superdroplet Method for collision coalescence
- Defining the Deficit Problem/Adaptive Timestepping
- Adaptivity in a Box model
- Adaptivity in 2d prescribed flow

Clouds are important and hard to model accurately!

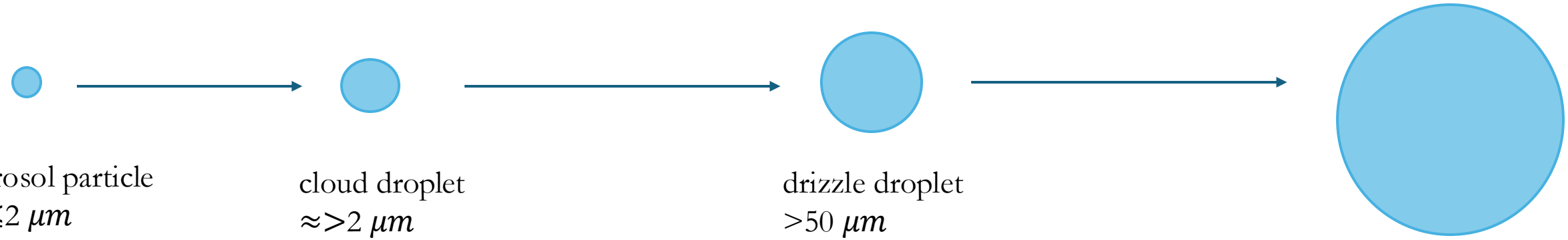
- Weather Forecasting
- Radiation Budget and Climate

We need accurate and efficient representation of cloud microphysical properties and dynamics

Understanding small scale processes leads to accurate parameterizations of larger ones!!!!



Xue L, Fan J, Lebo ZJ, et al. (2017)



Aerosol particle
$\approx \leq 2\ \mu m$

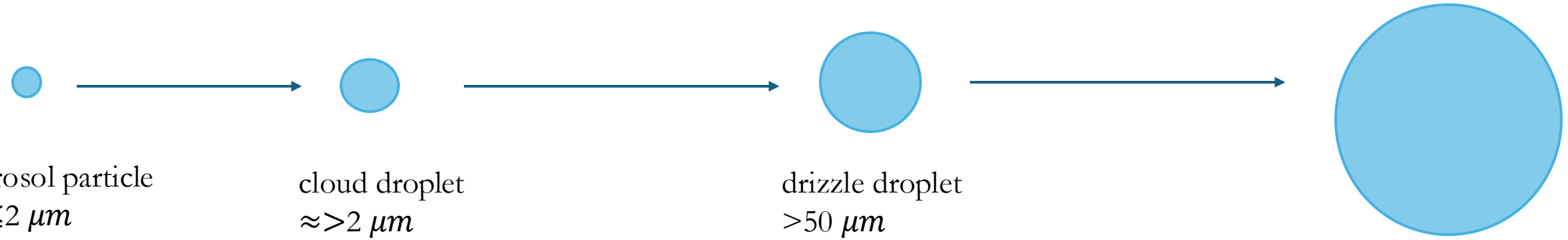
cloud droplet
$\approx > 2\ \mu m$

drizzle droplet
$> 50\ \mu m$

rain droplet
$> 100\ \mu m$

Condensation

Collision-Coalescence



Aerosol particle
$\approx \leq 2\ \mu m$

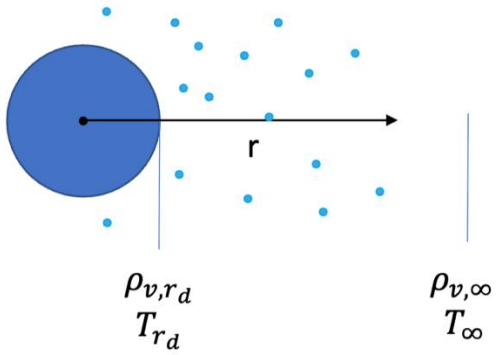
cloud droplet
$\approx > 2\ \mu m$

drizzle droplet
$> 50\ \mu m$

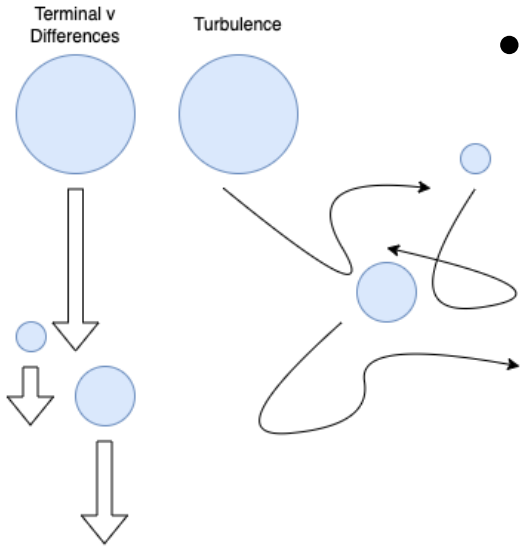
rain droplet
$> 100\ \mu m$

Condensation

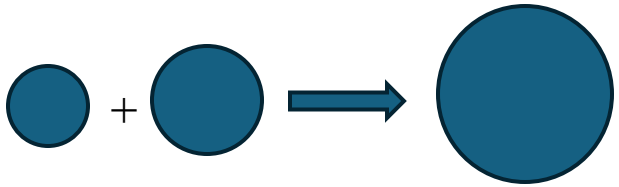
- Diffusional Growth
- Fick's Law

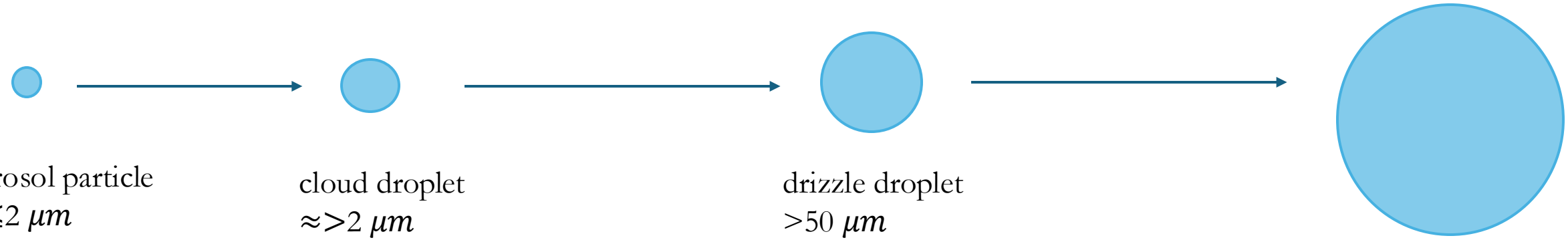


Collision-Coalescence



- Colliding droplets coagulate to form larger droplets





Aerosol particle
$\approx \leq 2\ \mu m$

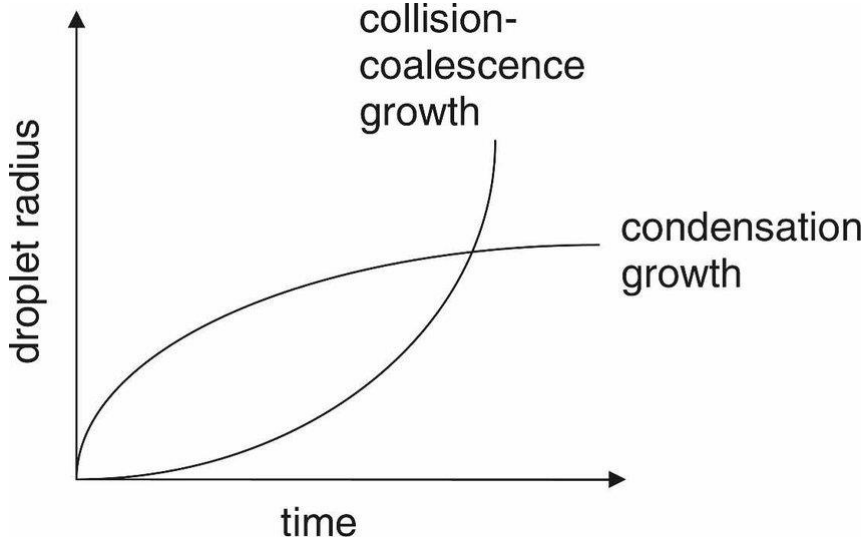
cloud droplet
$\approx > 2\ \mu m$

drizzle droplet
$> 50\ \mu m$

rain droplet
$> 100\ \mu m$

Condensation

Collision-Coalescence



It would take a 2 μm activated cloud droplet ***24 hours*** to grow into 100μm raindrop in average conditions through condensation alone

Smoluchowski Coagulation Equation

$$\frac{\partial n(x,t)}{\partial t} = \frac{1}{2}\int_0^x K(x-y,y)n(x-y,t)n(y,t)\,dy - \int_0^\infty K(x,y)n(x,t)n(y,t)\,dy.$$

GAIN

LOSS

Change in # particles of size X	=	Smaller particles colliding to become size X	-	Particles leaving size X due to collisions with other particles

$K(R_i, R_j)$: $Kernel(j,k)$, describes the likelihood of these interactions happening

(Smoluchowski, 1916)

Smoluchowski Coagulation Equation

$$\frac{\partial n(x,t)}{\partial t} = \frac{1}{2}\int_0^x K(x-y,y)n(x-y,t)n(y,t)\,dy - \int_0^\infty K(x,y)n(x,t)n(y,t)\,dy.$$

GAIN

LOSS

Change in # particles of size X	=	Smaller particles colliding to become size X	-	Particles leaving size X due to collisions with other particles

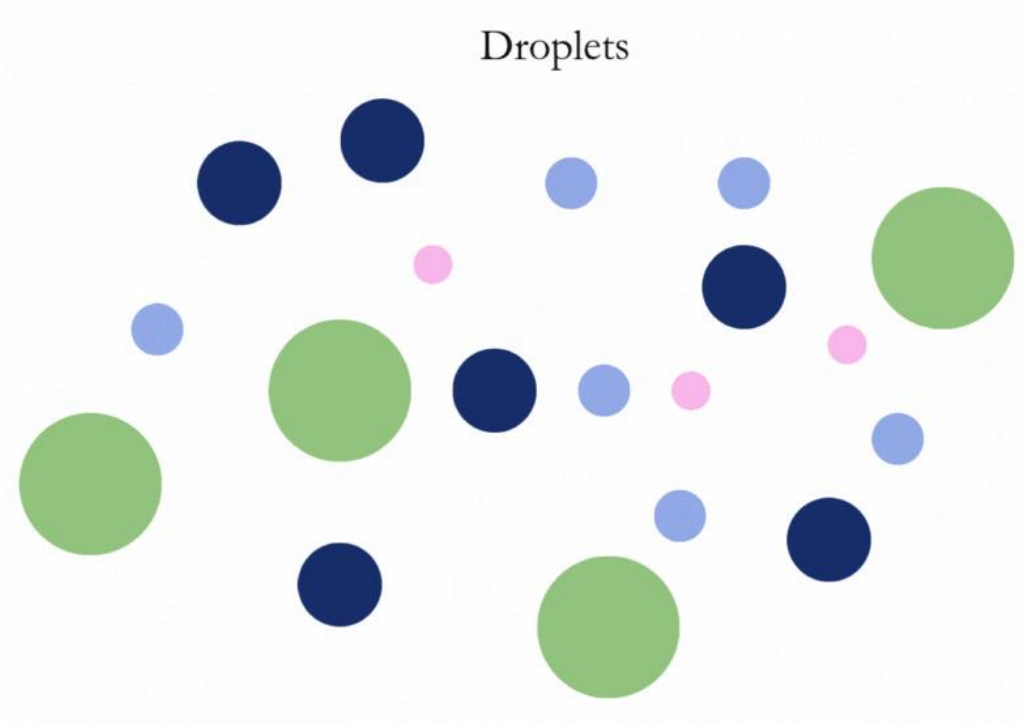
- Clouds
- Aerosol
- Shrimp Populations
- Fuel in pistons

- Bubbles
- Oil Spills
- Planetary Formation from dust

(Smoluchowski, 1916)

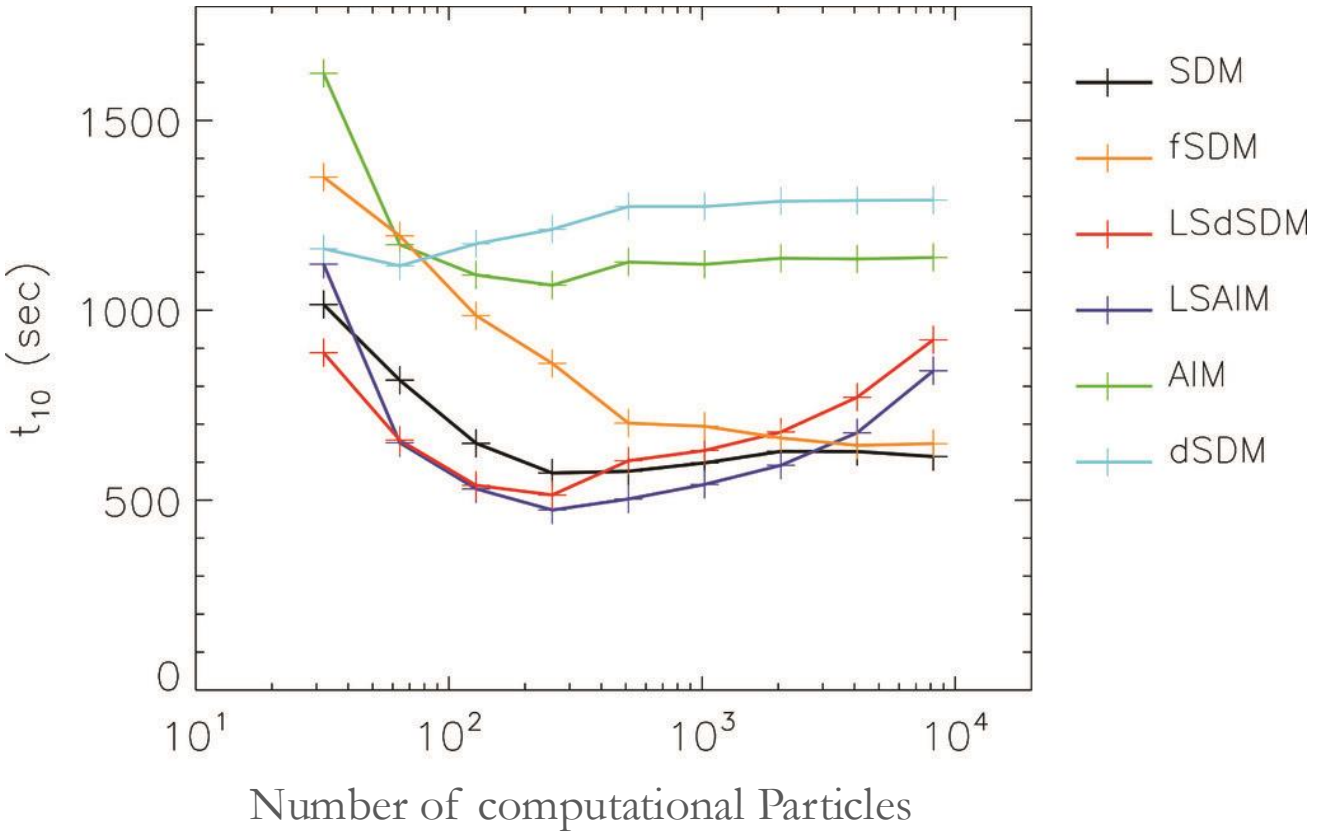
Particle-Based Methods (Superdroplets)

- Lagrangian tracers
- One computational superdroplet represents many computational droplets
- These particles can have attributes such as radius, isotopic composition, chemical composition, hygroscopicity, position
- Retain histories
- Well suited for Monte-Carlo coagulation algorithms



Still huge variability *just* in coagulation algorithms!!

t_{10} = time for 10% of system cloud water mass to reach "raindrop" size (>40um)

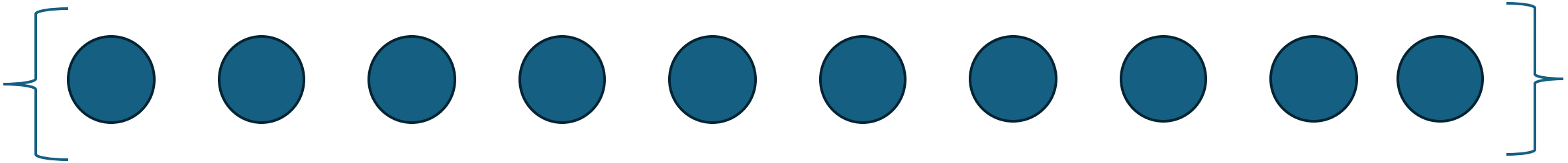


- 3D LES Cumulus Congestus Cloud
- Using piggybacking technique to separate microphysical behavior
- This shows variations on the Super Droplet Method (SDM, Shima et al., 2009) and Average Impact Method (AIM, Riechelmann et al. 2012)
- Everything here is a particle-based representation

- Background on Cloud Processes
- **Superdroplet Method for collision coalescence**
- Defining the Deficit Problem/Adaptive Timestepping
- Adaptivity in a Box model
- Adaptivity in 2d prescribed flow

Super Droplet Method (Shima et al. 2009)

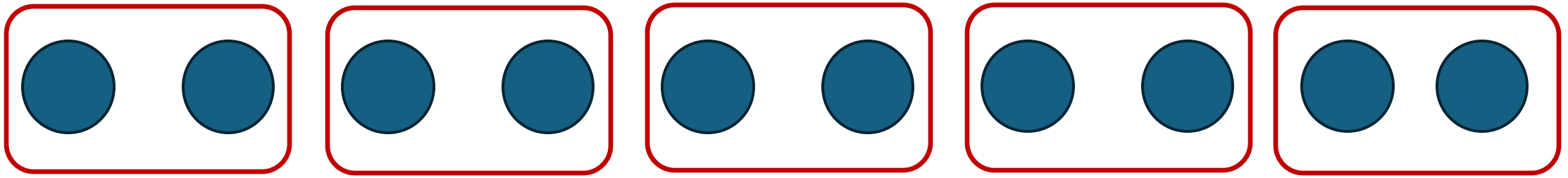
- When initialized efficiently, it is shown to outperform other algorithms in terms of convergence to analytic solutions (Unterstrasser et al, 2017)
- Monte-Carlo algorithm
- Tests pairs of superdroplets for collision events
- SDM Stochasticity resolves lucky rain formation (Morrison et al., 2024)

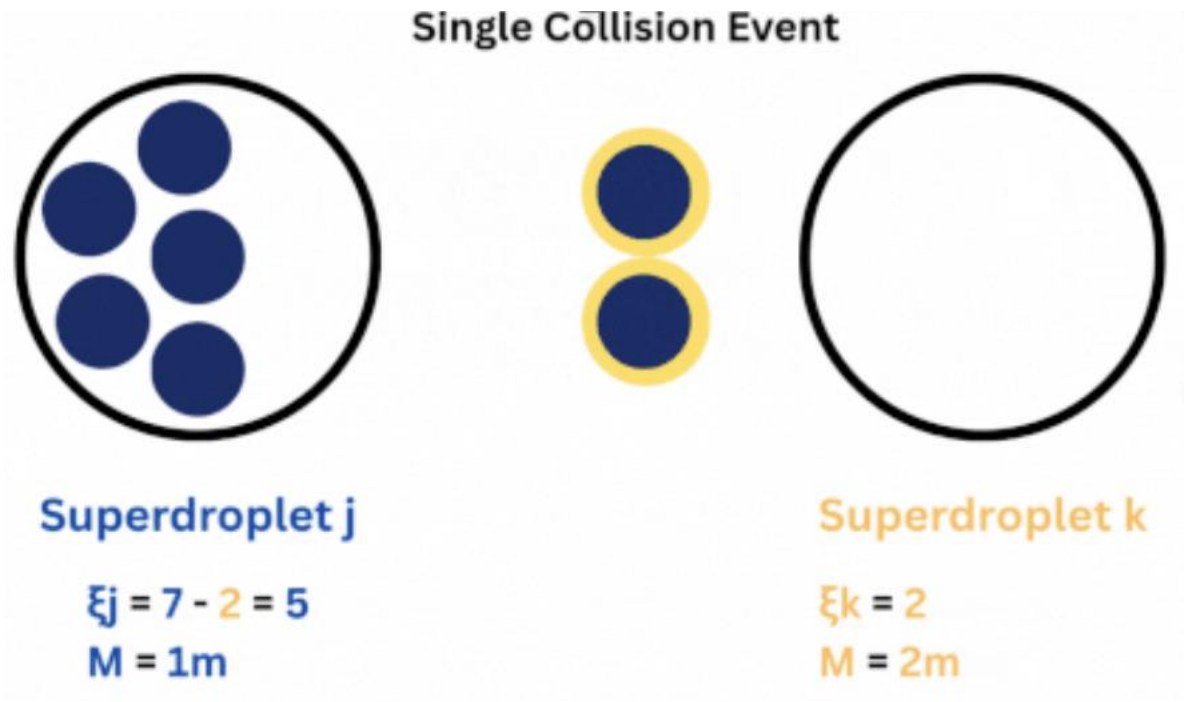


Super Droplet Method (Shima et al. 2009)

- When initialized efficiently, it is shown to outperform other algorithms in terms of convergence to analytic solutions (Unterstrasser et al, 2017)
- Monte-Carlo algorithm
- Tests pairs of superdroplets for collision events
- SDM Stochasticity resolves lucky rain formation (Morrison et al., 2024)

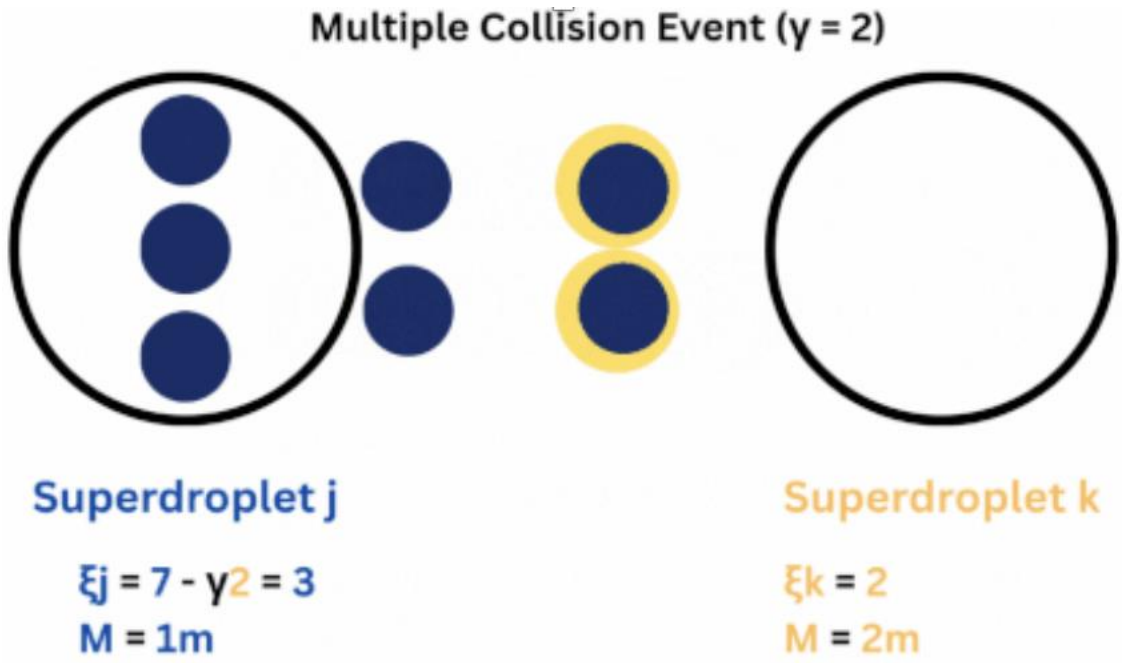
$$p_{pair\,j,k} = \frac{timestep}{collision\ volume} \cdot \xi_j \cdot \frac{total\ pairs}{pairs\ tested} \cdot Kernel(j,k), \qquad choosing\ \xi_j > \xi_k$$





- Choosing j and k so that $\xi_j > \xi_k$
- Superdroplet k collects ξ_k droplet from donator superdroplet j
- Superdroplet k grows in mass, superdroplet j decreases multiplicity

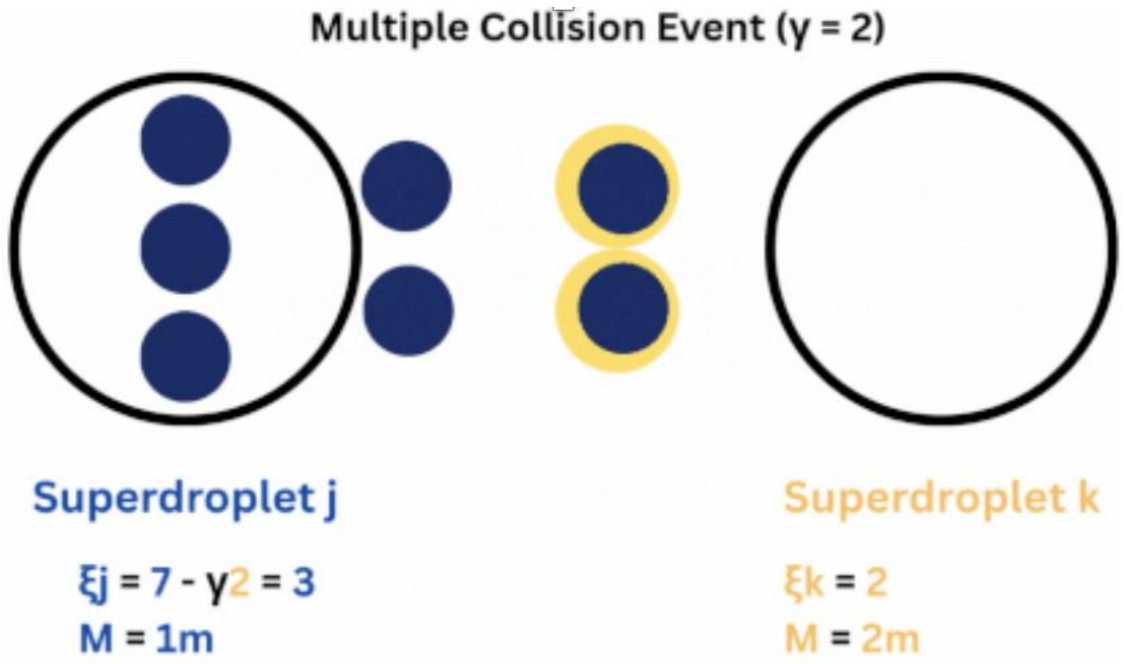
Mass and number of superdroplets are conserved



$\gamma =$ integer number of events predicted by probability and Monte-Carlo success

- Choosing j and k so that $\xi_j > \xi_k$
- Superdroplet k collects ξ_k droplet from donator superdroplet j
- Superdroplet k grows in mass, superdroplet j decreases multiplicity

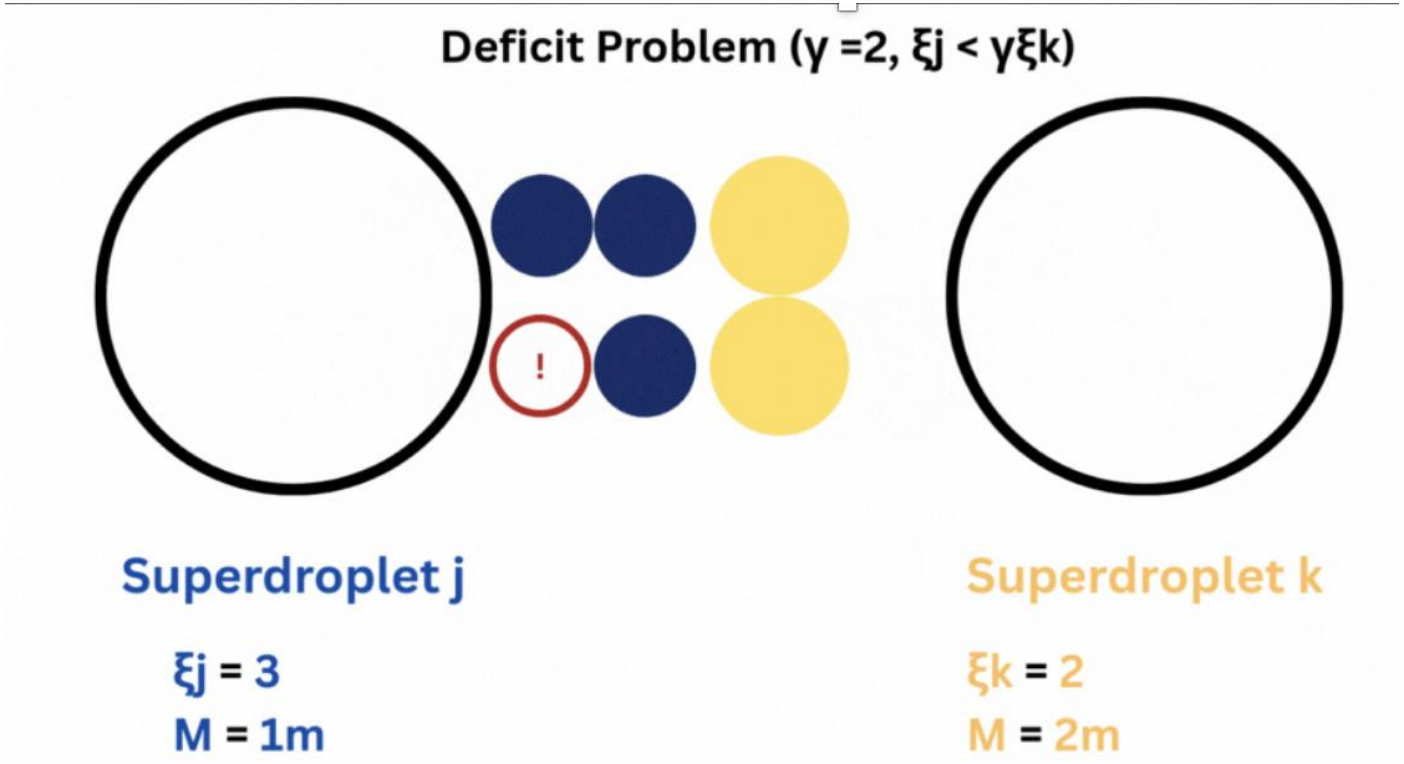
Mass and number of superdroplets are conserved



$\gamma =$ integer number of events predicted by probability and Monte-Carlo success

- Faster than bin methods in high dimensional attribute space
- Scales efficiently with number of attributes and particles
- Embarrassingly Parallel
- No numerical diffusion (retains particle identity)
- Conserves mass and Ns

- Background on Cloud Processes
- Superdroplet Method for collision coalescence
- **Defining the Deficit Problem/Adaptive Timestepping**
- Adaptivity in a Box model
- Adaptivity in 2d prescribed flow

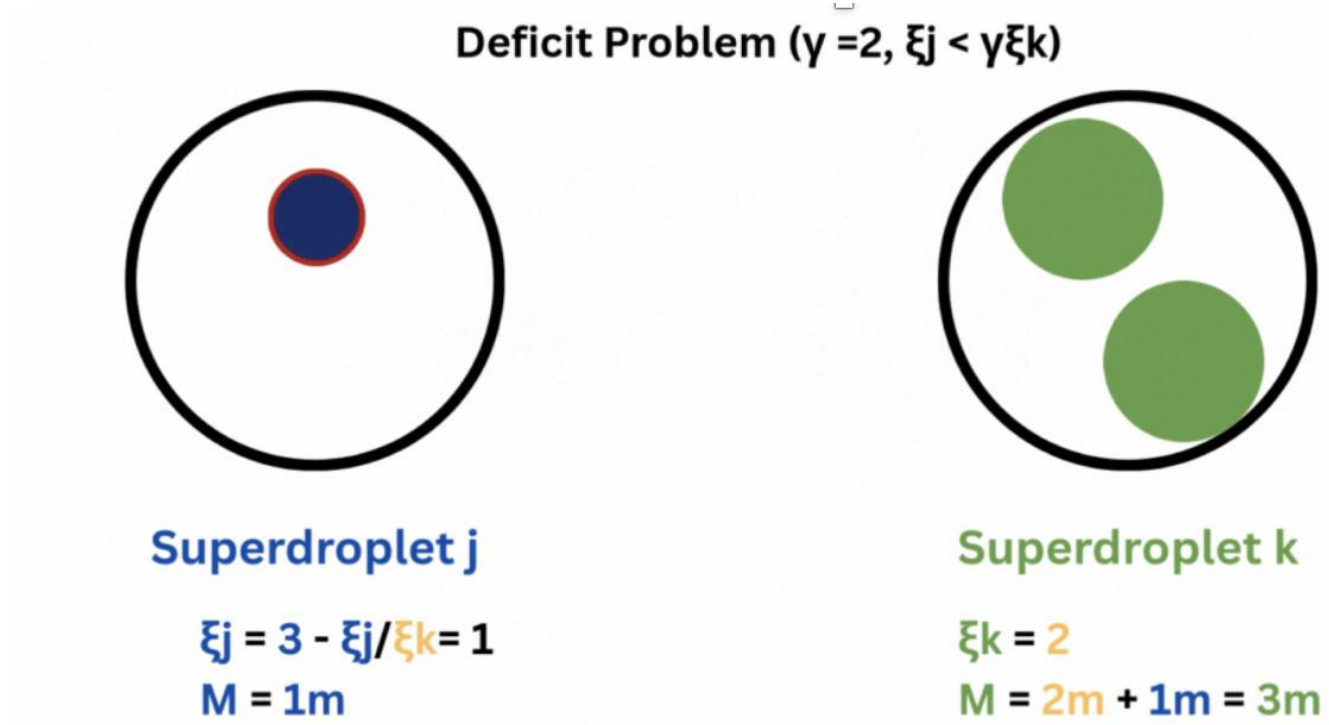


$$\gamma > \frac{\xi_j}{\xi_k} \; !!$$

$$\gamma_{occured} = \left\lfloor \frac{\xi_j}{\xi_k} \right\rfloor$$

$$\gamma_{occured} < \gamma_{predicted}$$

- Shima et al., 2009 limit γ by $\frac{\xi_k}{\xi_j}$
- Pointed out in Dziekan and Pawlowska (2017)
- Quantified in Bartman, P., & Arabas, S. (2023).



$$\gamma > \frac{\xi_j}{\xi_k} \text{ !!}$$

$$\gamma_{occured} = \left\lfloor \frac{\xi_j}{\xi_k} \right\rfloor$$

$$\gamma_{occured} < \gamma_{predicted}$$

- Shima et al., 2009 limit γ by $\frac{\xi_k}{\xi_j}$
- Pointed out in Dziekan and Pawlowska (2017)
- Quantified in Bartman, P., & Arabas, S. (2023).

$$deficit = \left(\gamma_{predicted} - \gamma_{occured}\right) \cdot \xi_k$$

Adaptive Time-Stepping Algorithm

$$deficit = \left(\gamma_{predicted} - \gamma_{occured}\right) \cdot \xi_k \qquad p_{pair\,j,k} = \frac{timestep}{collision\ volume} \cdot \xi_j \cdot \frac{total\ pairs}{pairs\ tested} \cdot Kernel(j,k),$$

We want to avoid $\gamma_{predicted} > \left|\frac{\xi_j}{\xi_k}\right|$:

Find the maximum timestep allowed for the pair:

$$p_{pair\,j,k} = \tilde{p}_{pair}\Delta t$$

$$\Delta t\_max_{pair} = \left|\frac{\xi_j}{\xi_k}\right| / \tilde{p}_{pair}$$

Find the limiting timestep for the cell:

$$\Delta t_{adaptive} = \min(\{\Delta t\_max_{pair} \ldots\}, \Delta t_{parent})$$

Repeat the substep until the model step is complete:

$$\Delta t_{left} = \Delta t_{parent} - \Delta t_{adaptive}$$

Adaptive Time-Stepping Analogies

If an algorithm does not have a way to handle multiple collisions, it must have an analogous method to constrain the probability < 1

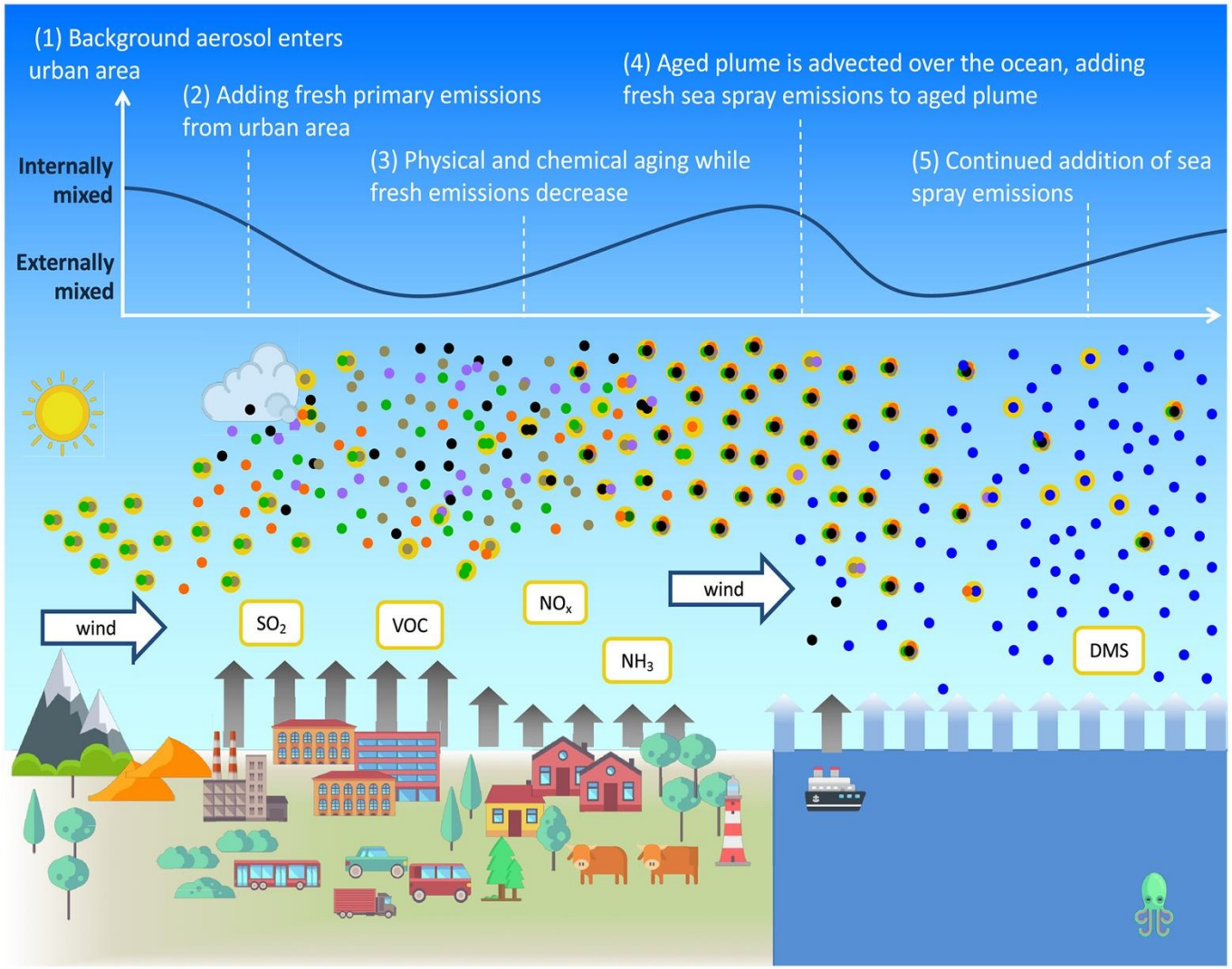
RemappingAlgorithm(RMA), Andrejczuk et al. (2010)

- RMA remaps between Superdroplets and binned mass every coalescence step
- Unterstrasser et al. (2017) proposes an adaptive substep when limited by bin concentration
- Comments on the stiffness of the Hall kernel when there are large droplets

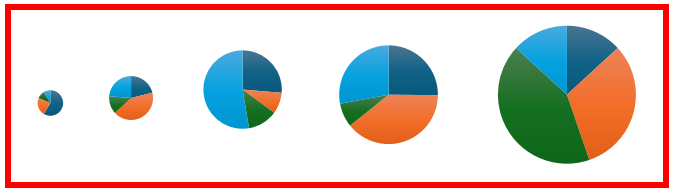
Weighted Flow Algorithm (WFA), Deville et al. (2011)

- used in PartMC (West, Riemer, et al., 2022)
- Particle-based Monte-Carlo algorithm
- Weighting is not a particle attribute
- optimizes the number of pairs tested out of the possible pairs

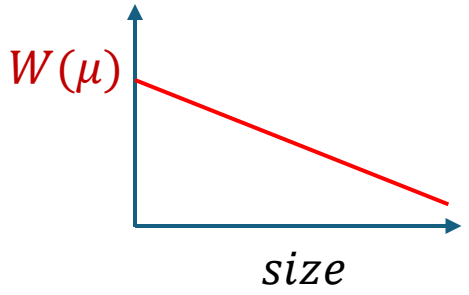
Adaptive Time-Stepping Analogies: WFA



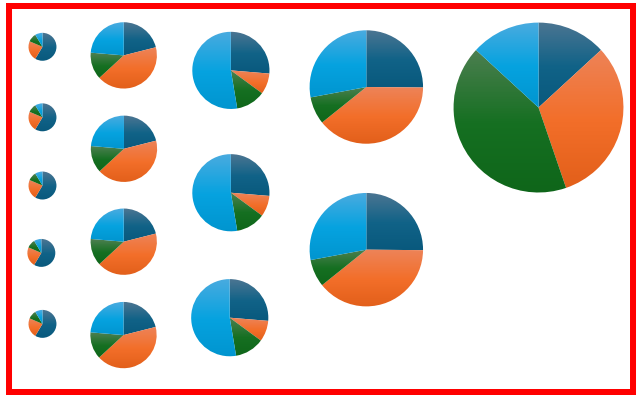
superparticle population:



weighting function:



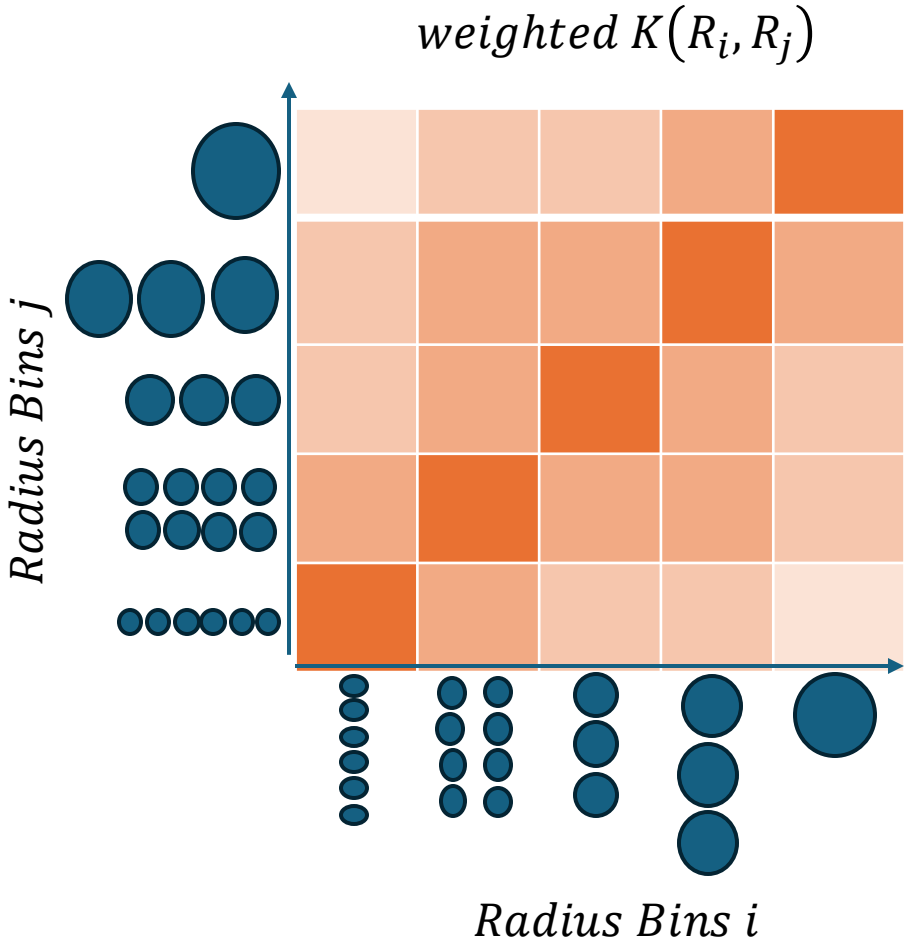
particle population:



Z. D'Aquino, et al.. PyPartMC (2024), N. Riemer, et al (2019), J. H. Curtis et al., (2017)

PyPartMC

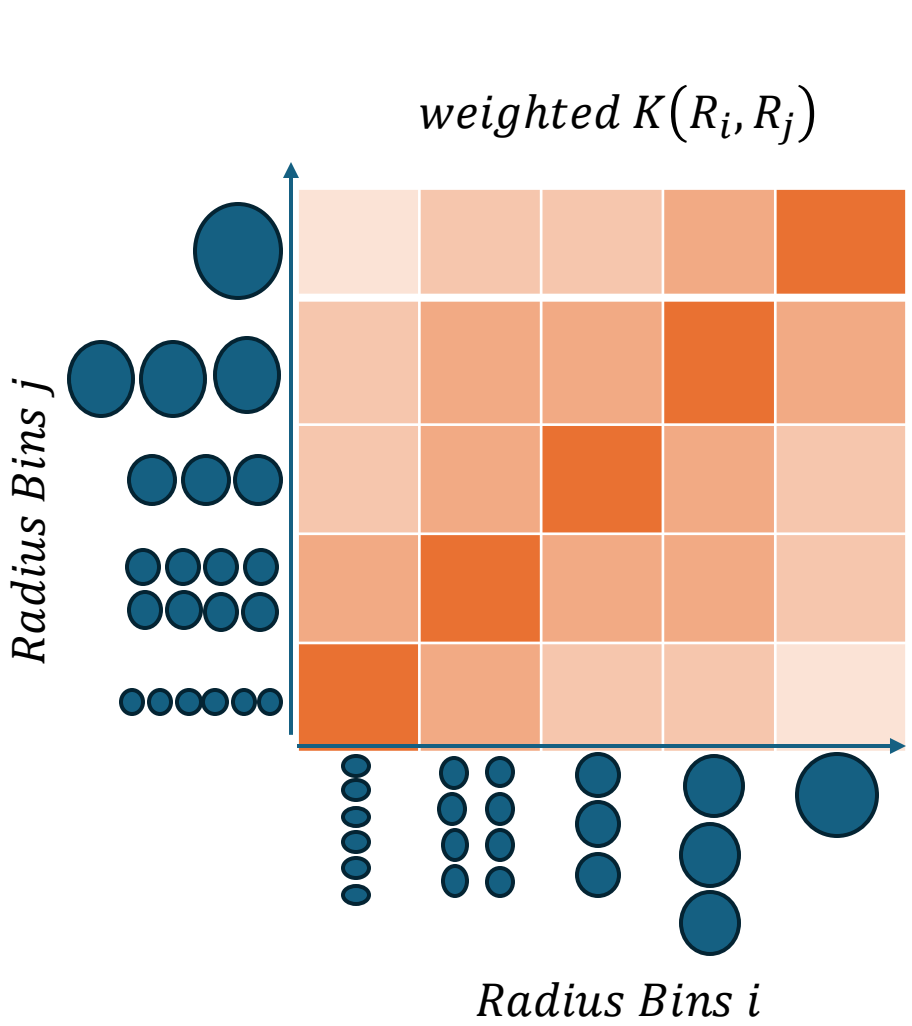
Adaptive Time-Stepping Analogies: WFA

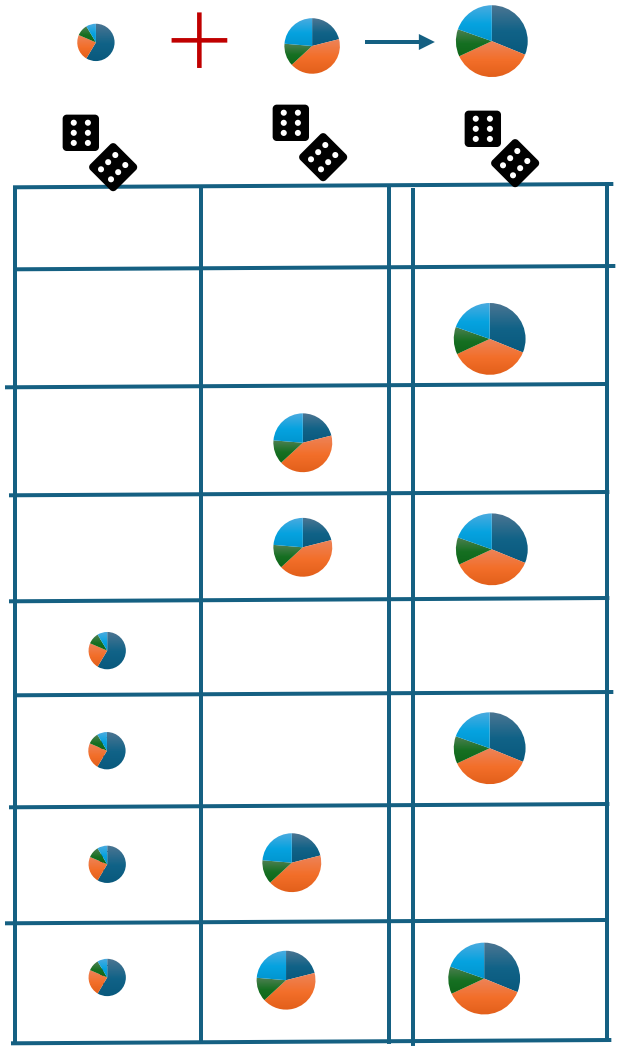


$$p_\alpha \propto \frac{total\ pairs\ in\ bin_{ij}}{pairs\ tested} \cdot weighted\ Kernel(R_i, R_j)$$

- Weighted Kernel of bin pair does not change in time
- Pairs tested is optimized at coarse bins by maximizing p_α to 1

Adaptive Time-Stepping Analogies: WFA

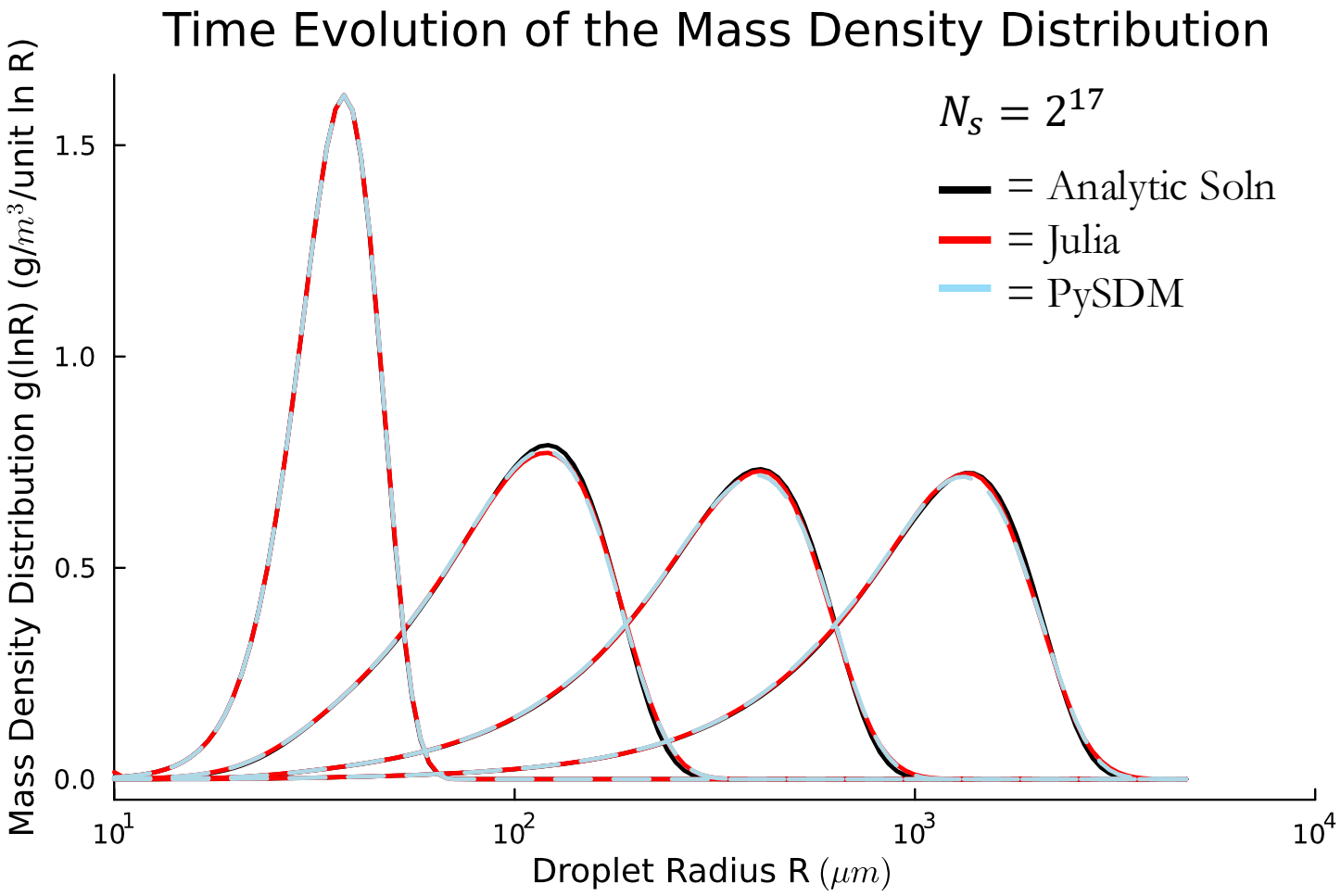




- Weights are helpful for the optimization
- Resulting superparticle might not be the weight of adding two smaller superparticles
- All 3 are probability checked for existence after a collision event based on their weights

- Background on Cloud Processes
- Superdroplet Method for collision coalescence
- Defining the Deficit Problem/Adaptive Timestepping
- **Adaptivity in a Box model**
- Adaptivity in 2d prescribed flow

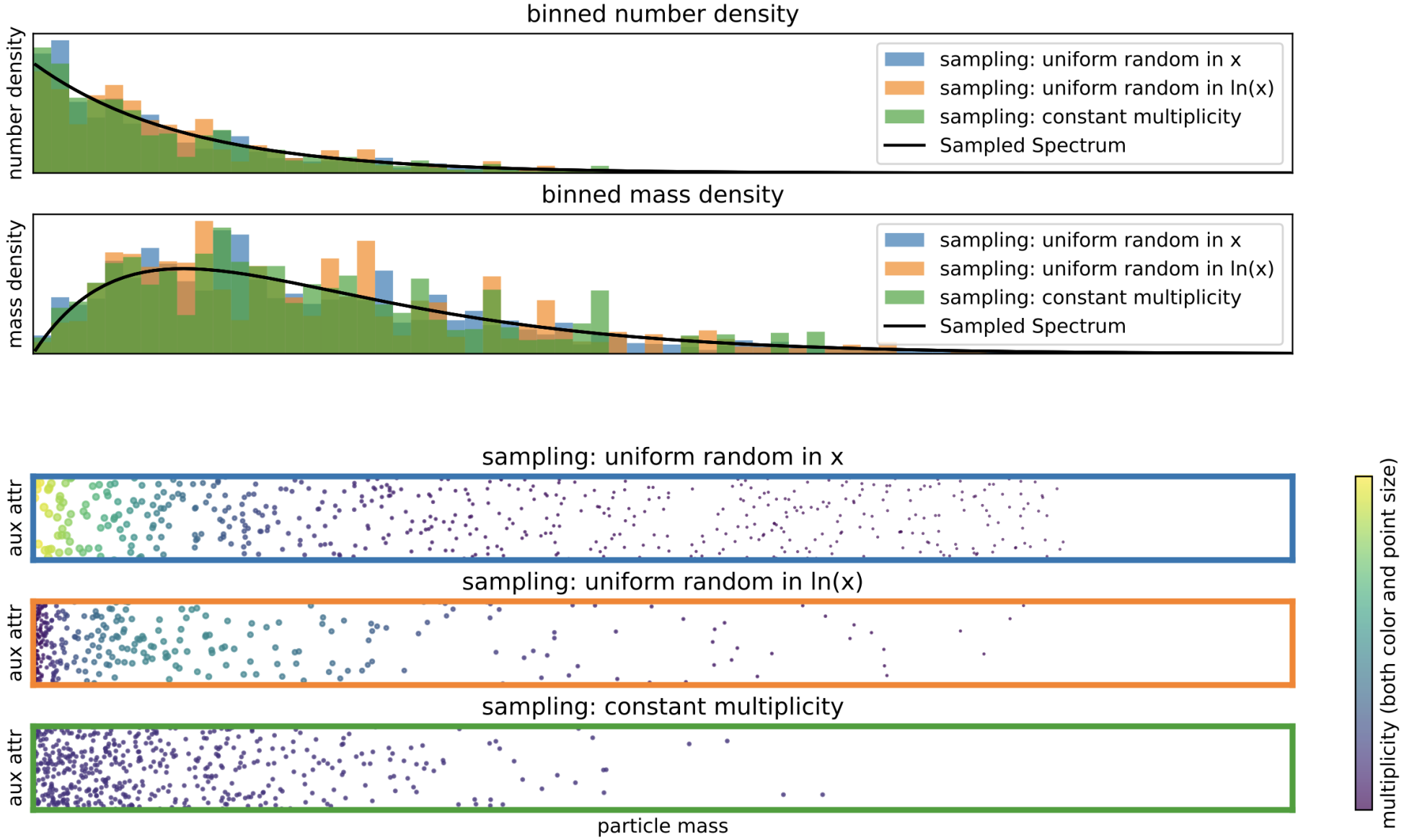
Box Model Test Setup



PySDM

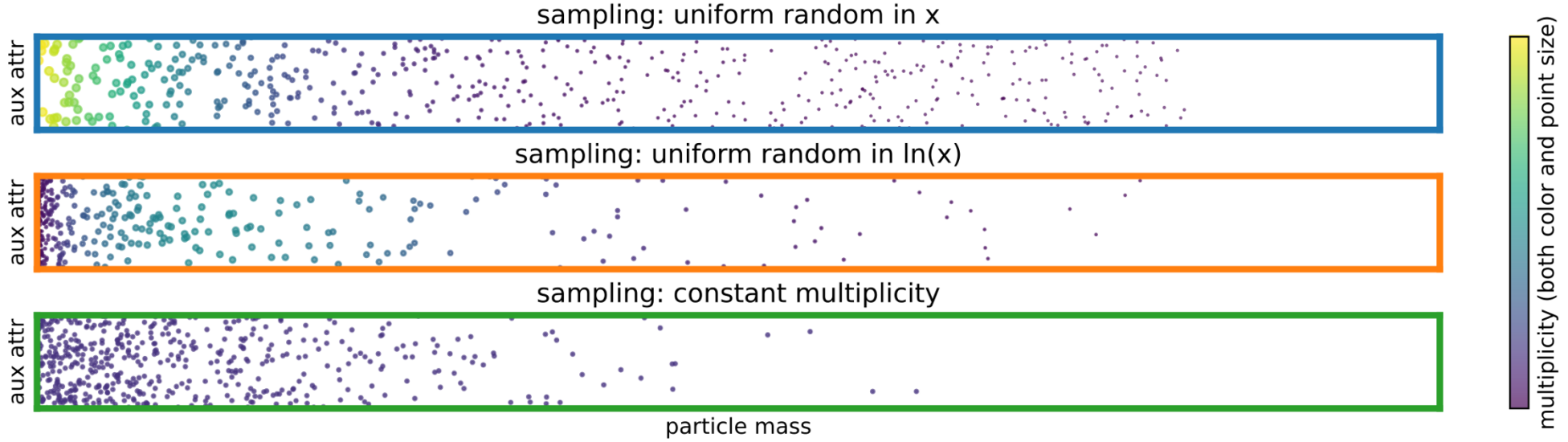
Droplets.jl

Initialization



Initialization

- Optimization and comparison of initialization is explored in Unterstrasser et al. 2017 and 2019, Dziekan and Pawlowska 2017, Matsushima et al. 2023)
- Initializations with dynamic ranges of multiplicities are shown to do best
- Matsushima et al. (2023) proposed a new init method that parameterizes this dynamic spread



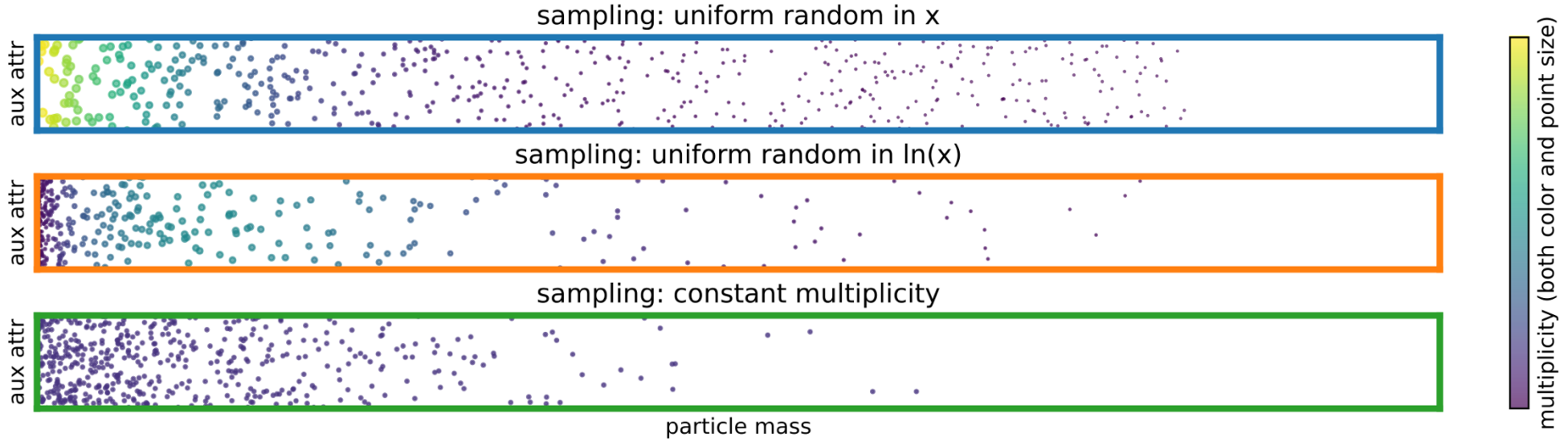
Initialization

$$p_{superdroplet\ pair} = \xi_j \cdot p_{droplet\ pair}, \quad choose\ \xi_j > \xi_k$$

$$droplet\ collisions\ per\ event = \xi_j \cdot \xi_k \cdot p_{droplet\ pair}$$

Dynamic range

It is advantageous to have many superdroplet events of small multiplicities ξ_k



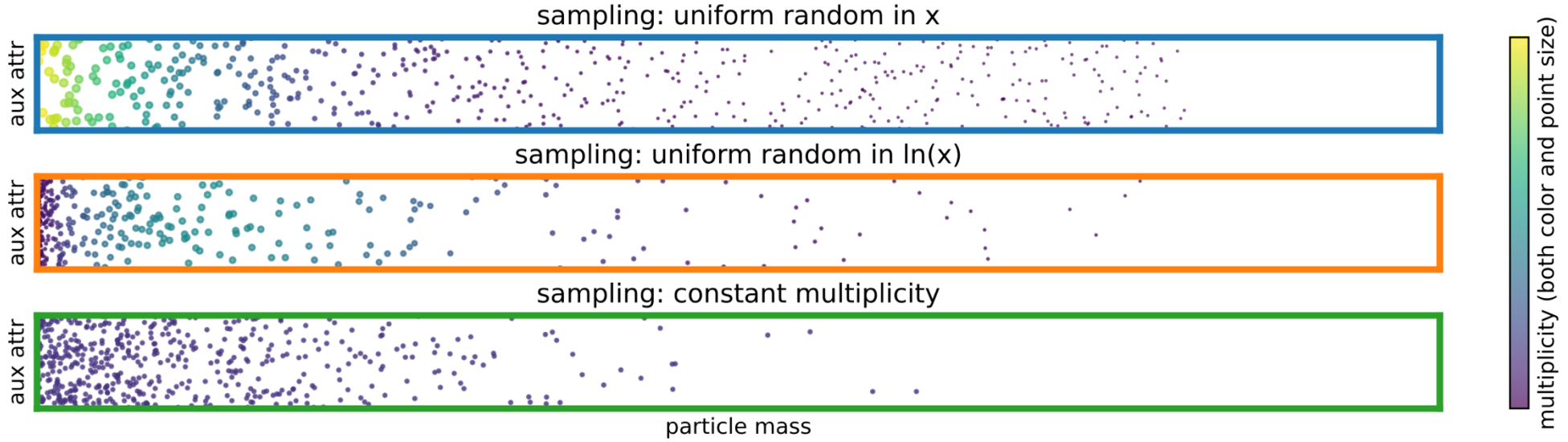
Initialization

$$p_{superdroplet\ pair} = \xi_j \cdot p_{droplet\ pair}, \quad choose\ \xi_j > \xi_k$$

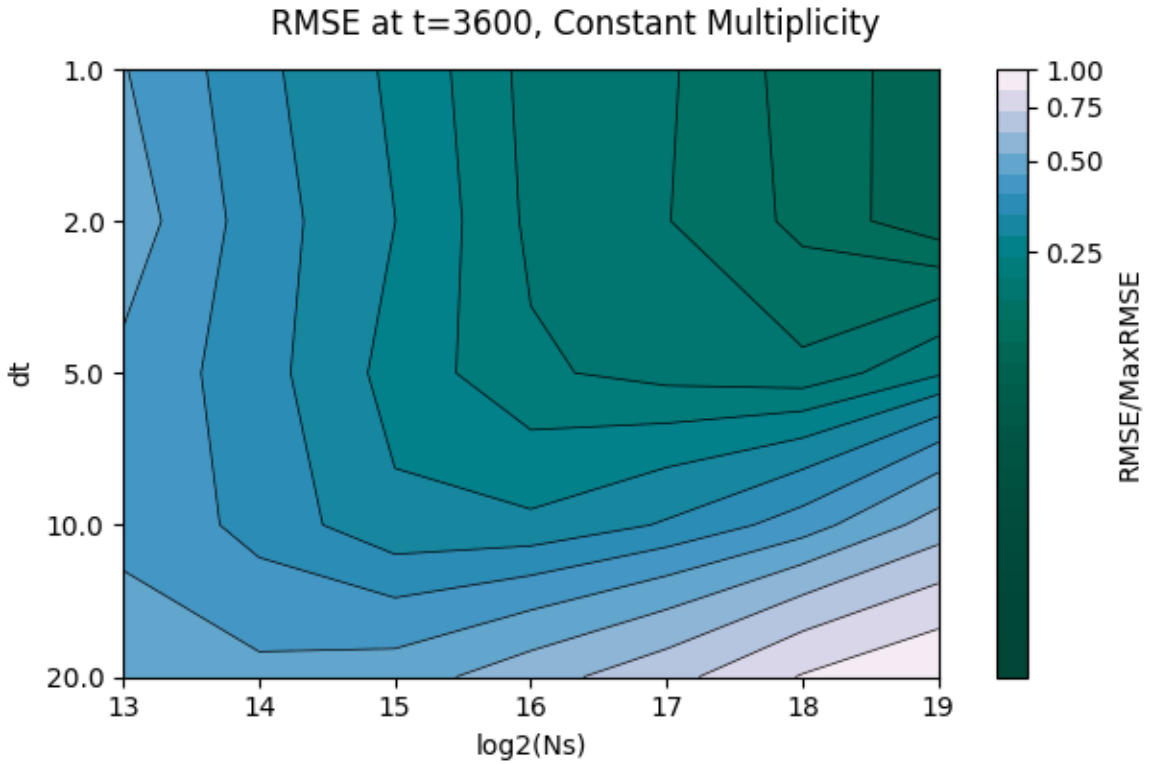
$$droplet\ collisions\ per\ event = \xi_j \cdot \xi_k \cdot p_{droplet\ pair}$$

Dynamic range

Lucky for deficit, dynamic range also maximizes $\overline{\xi_j/\xi_k}$

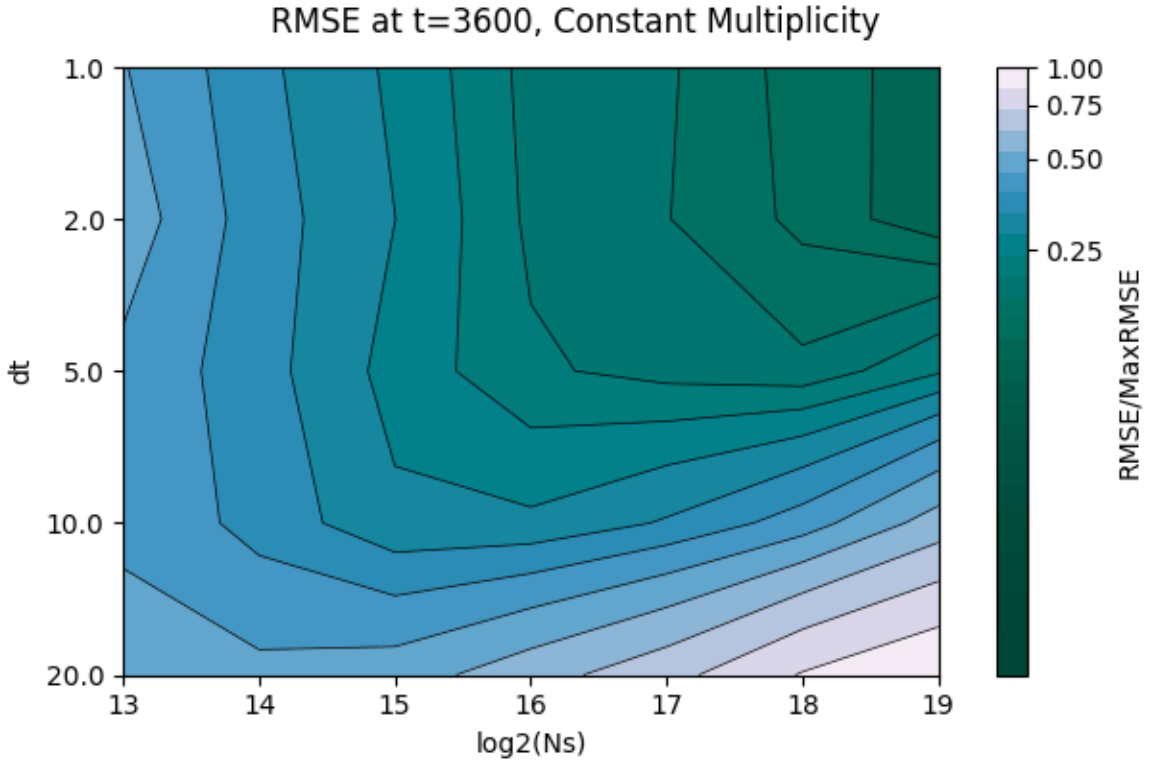


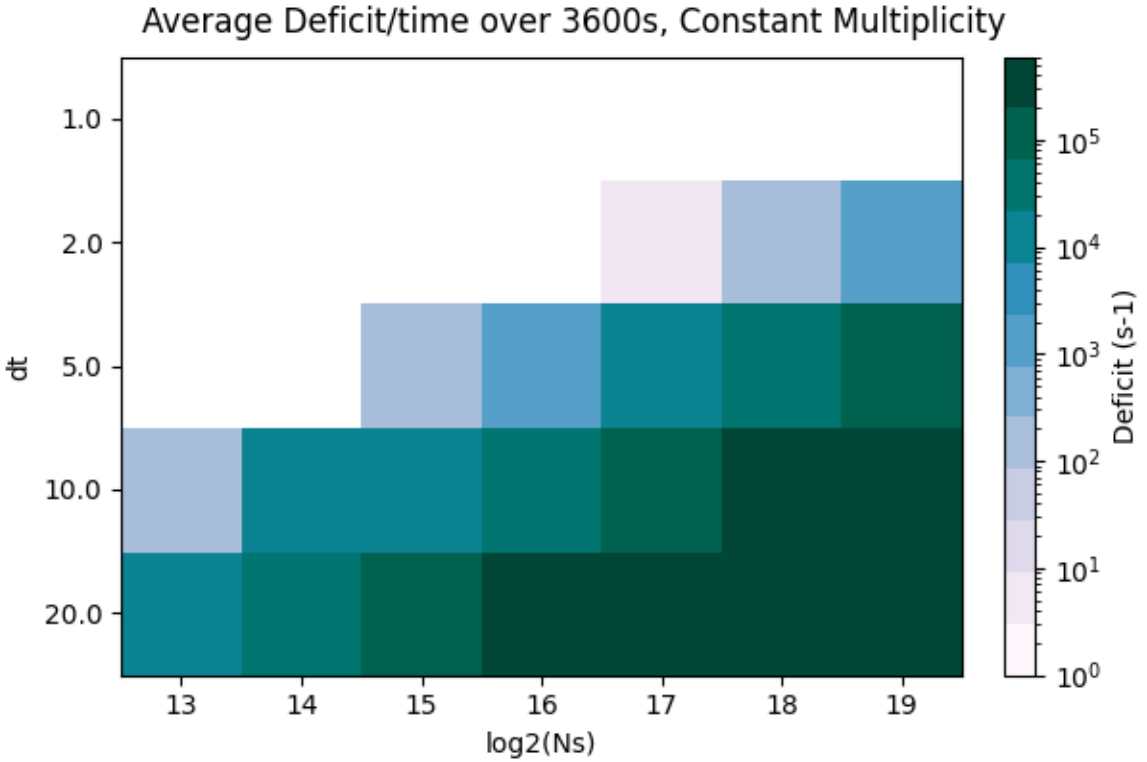
SDM Adaptivity in a Box Model



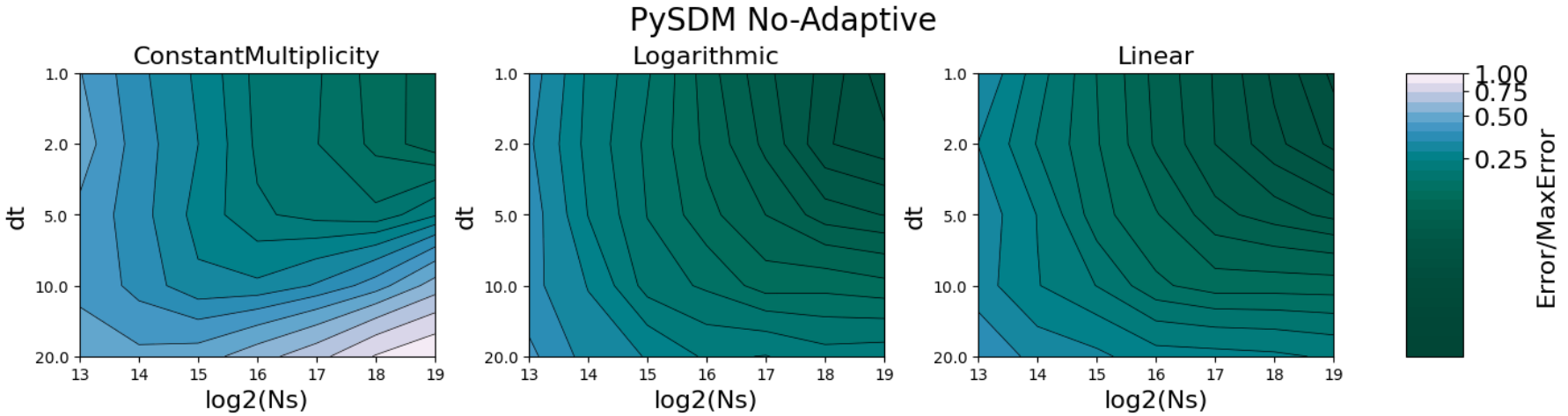
- 15 runs
- Increasing error with more superdroplets??

SDM Adaptivity in a Box Model

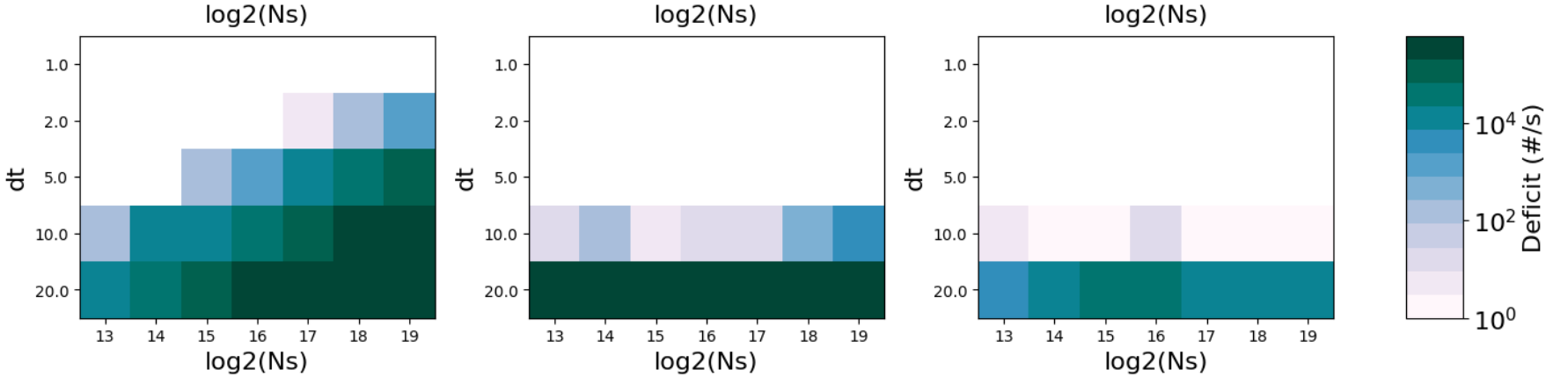




(RMSE t =3600s)
Fraction of largest error

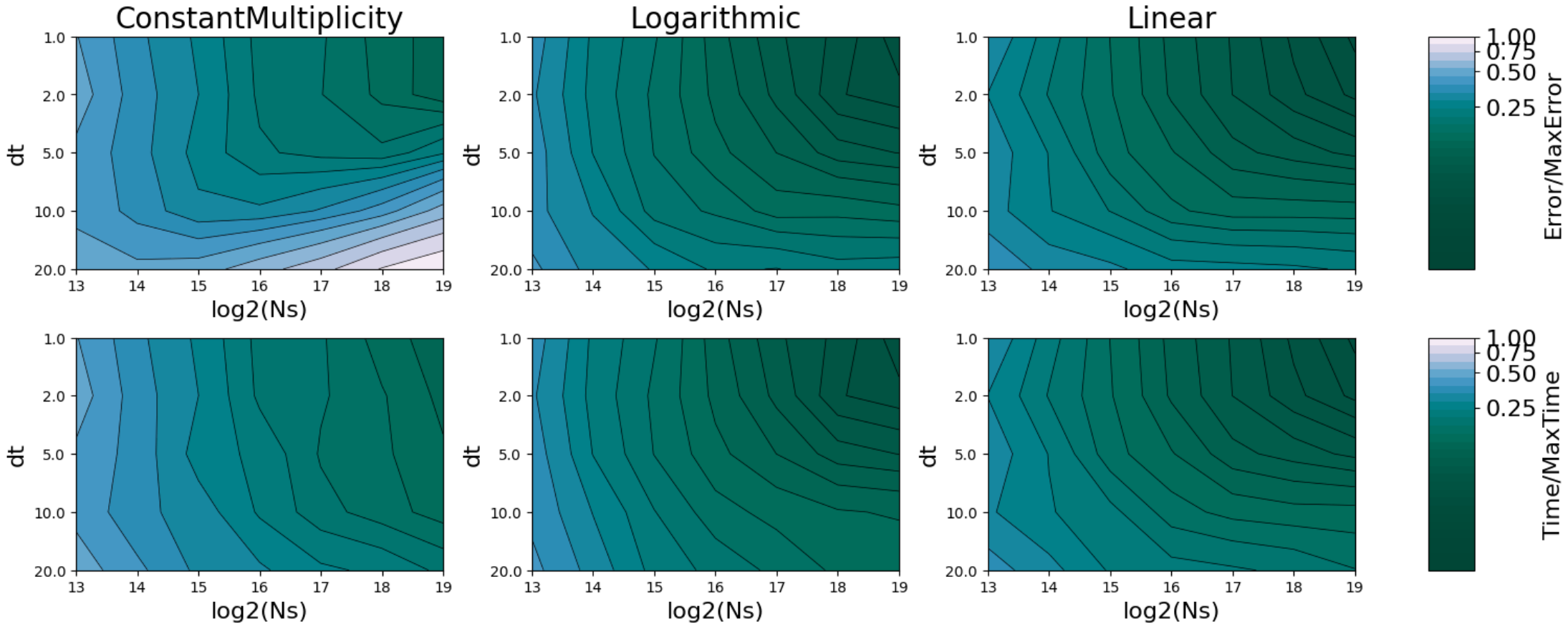


Deficit (#/s)



No Adaptivity

Adaptivity



- Background on Cloud Processes
- Superdroplet Method for collision coalescence
- Defining the Deficit Problem/Adaptive Timestepping
- Adaptivity in a Box model
- **Adaptivity in 2d prescribed flow**

2D prescribed velocity field

- Kessler 1969 Kinematic framework
- How does it behave with flow, grid exchange
- Unterstrasser et al. (2019) show less SD needed for 1d convergence
- 35 runs
- The results here only show varying time step
- 1.5x1.5 km domain, 50x50 grid, 40SD/cell
- Initialized by Uniform Log(dry radius)
- Geometric Kernel

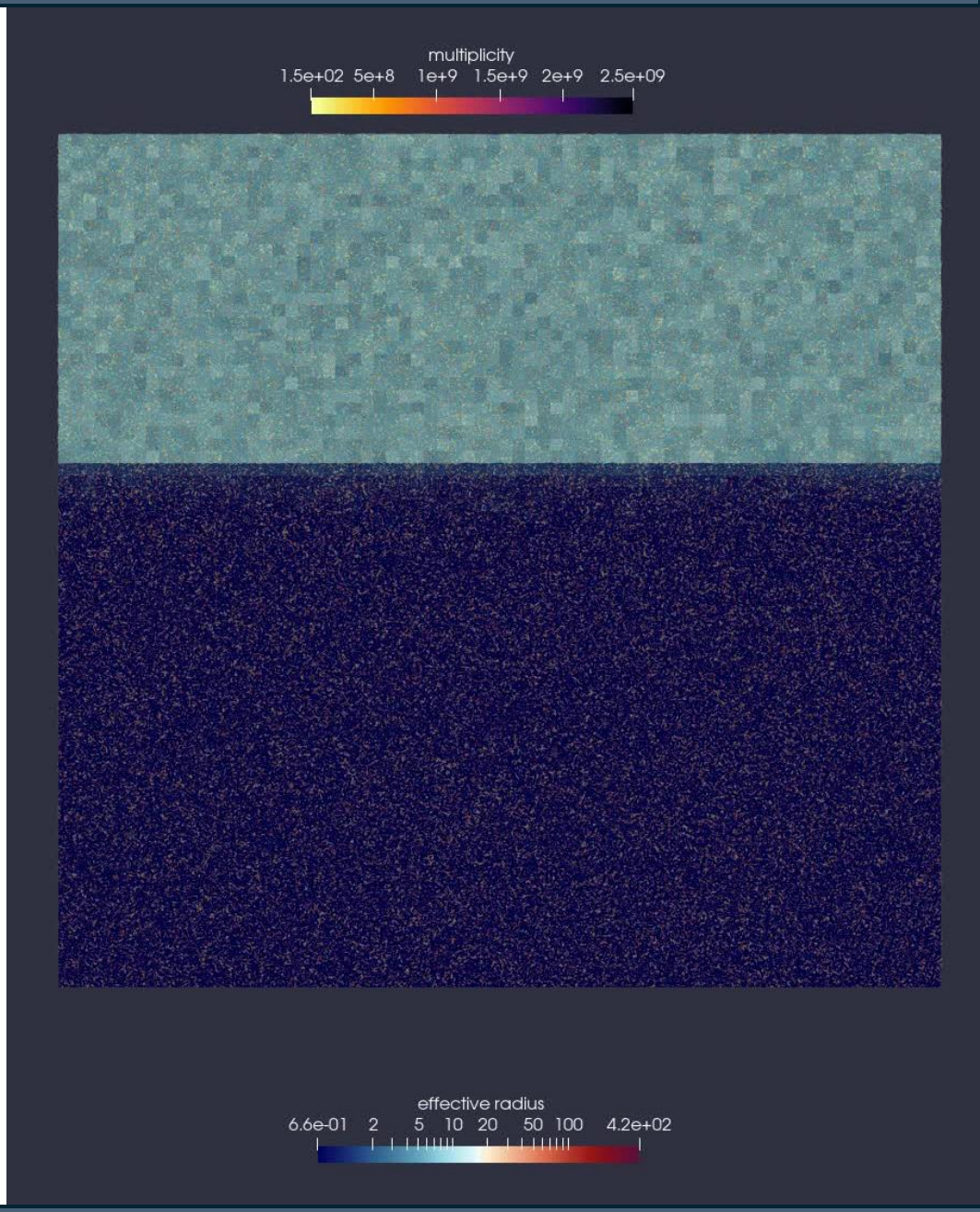
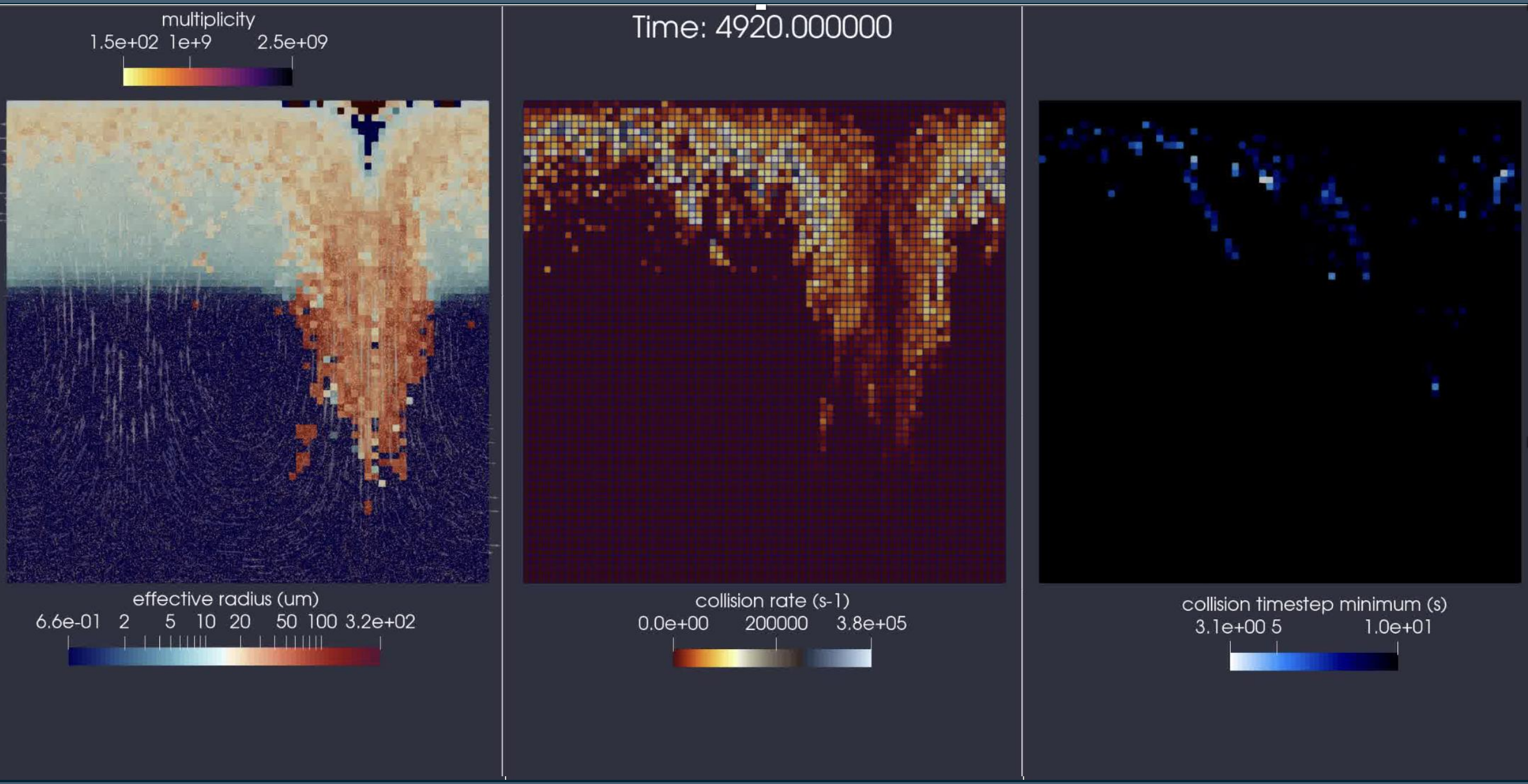
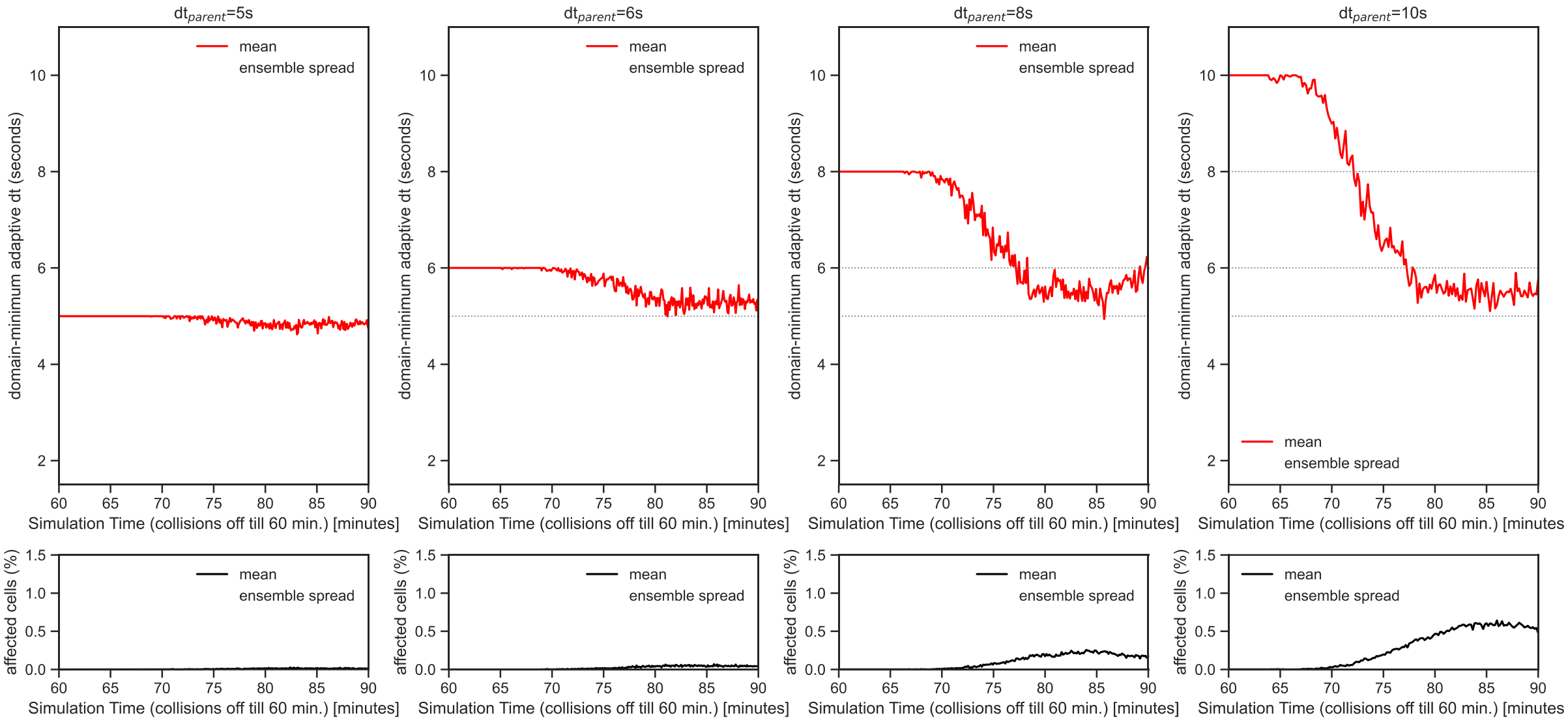
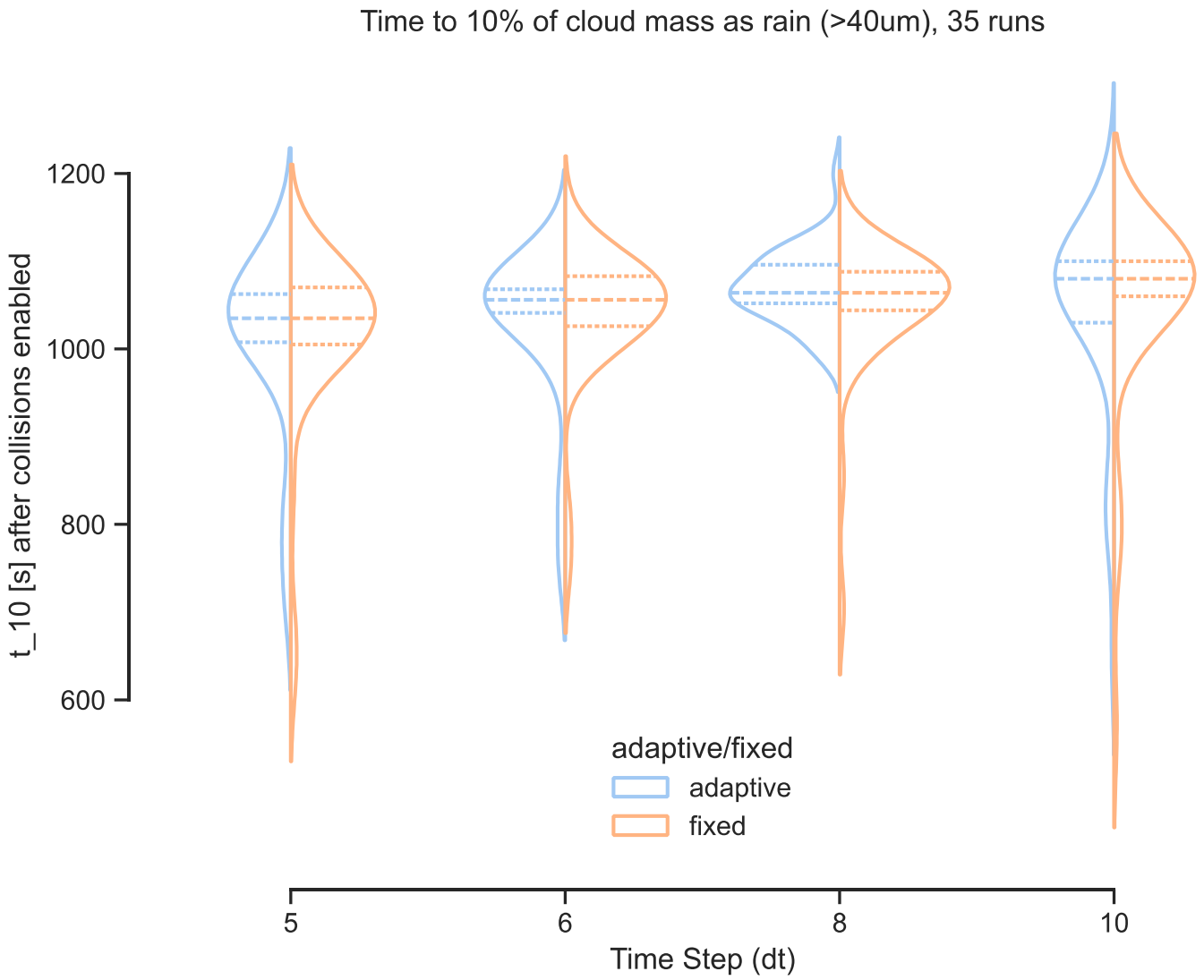


Fig. help thanks to Ola Strząbała


multiplicity
1.5e+02 1e+9 2.5e+09
Time: 4920.000000
effective radius (um)
6.6e-01 2 5 10 20 50 100 3.2e+02
collision rate (s-1)
0.0e+00 200000 3.8e+05
collision timestep minimum (s)
3.1e+00 5 1.0e+01

35 runs, adaptivity on (zero deficit), grid=50x50, init n_{sd}≈40/cell





- Given these conditions, we do not see increased convergence of bulk properties with adaptive time stepping

Despite Low Effect on the Convergence shown here, we *still recommend implementation*

$$p_\alpha \propto \frac{\Delta t}{\Delta V} \cdot \xi_J \cdot \frac{(N_{SD}-1)N_{SD}}{N_{SD}} \cdot K(R_i, R_j), \qquad \overline{\xi_J} \propto f(init) \cdot \left(\frac{n_{cell\ droplets} \cdot \Delta V}{N_{SD}}\right)$$

Despite Low Effect on the Convergence shown here, we *still recommend implementation*

$$p_\alpha \propto \frac{\Delta t}{\Delta V} \cdot \xi_J \cdot \frac{(N_{SD}-1)N_{SD}}{N_{SD}} \cdot K(R_i, R_j), \qquad \overline{\xi_J} \propto f(init) \cdot \left(\frac{n_{cell\ droplets} \cdot \Delta V}{N_{SD}}\right)$$

$$\overline{p_\alpha} \propto \frac{\Delta t}{\Delta V} \cdot f(init) \cdot \frac{n_{cell\ droplets} \cdot \Delta V}{N_{SD}} \cdot \frac{(N_{SD}-1)N_{SD}}{N_{SD}} \cdot K(R_i, R_j)$$

$$\overline{p_\alpha} \propto \Delta t \cdot f(init) \cdot n_{cell} \cdot K(R_i, R_j)$$

Despite Low Effect on the Convergence shown here, we *still recommend implementation*

- The setups used might not indicate any urgent need to reduce the deficit, but embracing the adaptivity makes the scheme more robust and less sensitive to arbitrary user settings
- Easily quantified nonphysical error that only lowers the growth
- Only affecting a small fraction of the grid cells (<1%) – not a reason to lower timestep
- This is not a domain wide setting: extra computational effort is focused where needed
- Increased superdroplet resolution does not reduce deficit

In summary…

- Sensitivity to deficit is much smaller than convergence on init methods/#sd
- Dynamic multiplicity range helps convergence *and* deficit while increasing Nsd does not
- PySDM and Droplets.jl are open source
- Ware, Bartman et al. (in prep.)