Response to the reviews of "University of Warsaw Lagrangian Cloud Model (UWLCM) 1.0: a modern Large-Eddy Simulation tool for warm cloud modeling with Lagrangian microphysics" by Dziekan et al. (doi:10.5194/gmd-2018-281)

May 13, 2019

We thank the reviewers for the work they have put into improving the manuscript. Before we respond to their comments, we need to point out that an error was found in our implementation of the radiation scheme used in the DYCOMS RF02 simulations. The error resulted in wrong distribution of radiative flux within cloud layer - radiative cooling was decreasing temperature practically only in the uppermost cloudy cell and only the lowermost cloudy cell was being radiatively heated. The error has been fixed and all simulations were repeated. The most profound difference in results is that the LWP has become higher and that there is more surface precipitation in 2D simulations. The large amount of surface precipitation in 2D simulations prompted us to study how precipitation formation in Lagrangian microphysics depends on the time step for coalescence. This sensitivity study is now presented in the section about 2D simulations.

Answer to the Anonymous Referee #1.

Major Comments

Diffusion of superdroplets (p. 10, ll. 5 6; p. 5, l. 14; p. 13, l. 25). The motion of superdroplets in only determined by the resolved-scale LES air motion. Turbulent diffusion, which is considered in the LES implicitly due to numerical diffusion, is not considered for the superdroplets. This underestimates the diffusion of superdroplets and liquid water in all simulations, indicating that the fields of water vapor,

temperature, and liquid water are not in physical agreement. I appreciate that the authors are candid about this issue, but they should address the implications of this discrepancy more clearly. Especially because there are methods available and to consider subgrid-scale motion of Lagrangian particles (e.g., Weil et al. 2004, doi:10.1175/JAS-3302.1), which are already in use in other Lagrangian cloud models (Slch and Krcher 2010, doi: 10.1002/qj.689; Hoffmann et al. 2017, doi: 10.1175/JAS- D-16-0220.1). One example where this neglect probably matters is the number of simulated cloud droplets N C . The authors state that N C is higher in UWLCM compared to other models (p. 13, l. 25). They explain this by the lack of numerical diffusion. This is right. However, the neglected turbulent diffusion of superdroplets also contributes to a higher N C and needs to be mentioned.

The issue of SGS diffusion was also brought up by the Reviewer #2. To resolve it, we have added results of 3D simulations using the Smagorinsky scheme, with and without SGS motion of Lagrangian particles. The section presenting 3D simulations is now focused on comparing these different SGS modeling techniques. After fixing the radiative scheme, ILES gives larger LWP than reference simulations. It is shown that to obtain agreement with the reference simulations it is necessary to use the Smagorinsky scheme and to include the SGS motion of Lagrangian particles.

Comparison of different time sub-stepping schemes. The comparison of the per-particle and per-cell sub-stepping approaches with a simulation without sub-stepping but a commensurately reduced timestep of 0.1 s is not very helpful due to the strong interaction of microphysics and dynamics. This becomes very clear for the three-dimensional simulations, in which the 0.1 s simulation enables a more detailed, and probably more adequate representation of this interaction. As a result, the entrainment rates vary significantly among the different model setups as seen in Fig. 5b, with commensurate effects on the liquid water path (decreases due to stronger entrainment), cloud base height (increases due to stronger entrainment), and indirectly precipitation (increases with liquid water path). To derive useful conclusions, it is necessary to untangle dynamical and microphysical effects. Therefore, I strongly suggest using either a kinematic driver providing each setup the identical dynamical forcing or to use the piggy-backing approach, which is actually part of UWLCM as stated on p. 22, ll. 12 13.

Following the comment, we performed 2D piggybacking simulations in which flow field from a simulation with $\Delta t = 0.1$ s is used to drive two simulations with $\Delta t = 1$ s and different substepping techniques. The conclusion is that, for stratocumulus clouds, the *per-cell* algorithm is better, but the *per-particle* algorithm also works well. All the stratocumulus simulations presented in the revised paper use *per-cell* substepping. We expect *per-cell* substepping to give errors for a fast moving cloud edge. To test this, we also present idealized 1D simulations of a moving cloud edge. There, *per-cell* substepping causes significant errors and *per-particle* algorithm works well. Discussion of differences between results of different substepping algorithms has been moved to the Appendix B. Section of the main text discussing 2D simulations deals now with sensitivity of Lagrangian microphysics to the coalescence time step and to the number of computational particles.

Minor Comments

P. 2, l. 9: Please clarify: Automated tests for what?

Some more information has been added:

” A set of automated tests greatly helps in ensuring the correctness of the model. The automated tests include a 2D moist thermal simulation, a 2D kinematic stratocumulus simulation and a test of different combinations of model options. Moreover, modeling of physical processes, e.g. condensation, advection, coalescence, sedimentation, is tested separately by the libmpdata++ and libcloudph++ test suites. ”

P. 2, l. 16 17: Focusing on precipitation is one aspect. Cloud cover might be an additional and very important second aspect to consider since precipitation might result in the transition from closed to open cells.

The paper introduces a new model, therefore we focus on basic cloud properties and do not study more complex behavior. However, in the discussion of 3D results we now mention that cloud cover is close to 100% in our simulations:

” Also, cloud cover, defined as fraction of columns with LWP $> 20\text{gm}^{-2}$, is close to 100 % in all 3D UWLCM simulations. ”

P. 3, l. 6 8: How does the auxiliary environmental state increase the precision of numerical calculations? Usually, these environmental states are necessary requirements to solve the system of equations. Furthermore, the word precision usually refers to the number of significant digits of the solution. I do not believe that this is meant by authors.

Our notion of the environmental state (also known as the ambient state) is distinct from the reference state. Introduction of environmental states is optional. However, their able choice can facilitate the design of initial or boundary conditions, improve the conditioning of the elliptic boundary value problems, and/or enhance the accuracy of calculations in finite-precision arithmetics (Smolarkiewicz et al., 2014; **?**). We admit that the word "precision" was confusing and we have changed it to "accuracy".

P. 3, Eq. (3): It is explained later, but a brief description of what is might be helpful at this point.

The sentence right after Eq. (3) now states: "where D_t denotes the material derivative: $D_t = \partial_t + \vec{u} \cdot \nabla$ and π is normalized pressure perturbation."

P. 4, Eq. (10): What are r and r_d ?

They are the dry and wet radius, respectively. We believe this should be clear, as it is stated in the first paragraph of the section and in the table in Appendix A.

P. 4, l. 12: What is so special about this definition of the relative humidity (the ratio of actual and saturation water vapor mixing ratio) to cite Lipps and Hamler (1982)? E.g., Clark (1973, doi:10.1175/1520-0469(1973)030¡0857:NMOTDA¿2.0.CO;2) defined the supersaturation (his Eq. (15)) also as the ratio between actual and saturation water vapor mixing ratio.

Small differences in definition of relative humidity can have visible impact on results. RH $= q_v/q_{vs}$ is an approximation of the more correct RH $= e/e_s$. More importantly, it is not obvious for us how to calculate dry air partial pressure p_d in the anelastic approximation. Should it follow from the ideal gas law, like vapor partial pressure does? Or should it be selected so that $e+p_d = p^e$, where e is calculated from the ideal gas law? Lipps and Hemler use the second approach and we also adopt it to be consistent with the Lipps-Hemler approximation, so we explicitly reference their paper.

P. 4, l. 13: Consider replacing 0.622 by the ratio of the specific gas constant of dry air to the specific gas constant of water vapor (i.e., R a /R v).

Done.

P. 4, l. 14: Please comment if D and K include gas kinetic or ventilation effects.

They include both, an appropriate comment has been made in the text.

P. 4, l. 19: For clarity, add real between two and droplets.

Done.

P. 4, ll. 24 25: Superdroplets do not collide. Equation (12) states the probability that one real droplet of superdroplet j (or k) collects any real droplet of superdroplet k (or j).

Incorrectly, we were using the words ”collide” and ”coalesce” to describe coalescence. This has been fixed by changing instances of ”collide” with ”coalesce”. The nomenclature that superdroplets coalesce is used following Shima et al. (2009). How we interpret coalescence of superdroplets is explained in the paragraph directly following eq. (12) (eq. 17 in the revised manuscript). Equation (12) does not state the probability that one real droplet of superdroplet j (or k) collects any real droplet of superdroplet k (or j). Instead, it states the probability that each of ξ_j real droplets of superdroplet j coalesces with a single real droplet of superdroplet k, where j and k labels are chosen so that $\xi_j \leq \xi_k$.

P. 4, l. 27: Starting from (12), there are not necessarily ξ_j pairs of real droplets coalescing. The correct number is min (ξ_j, ξ_k).

As stated on p.4 l. 28, SDs are labeled so that $\xi_j \leq \xi_k$. Then min (ξ_j, ξ_k) = ξ_j. To make this convention more clear, now we introduce it right after eq. (12):

” where SDs are labeled so that $\xi_j \leq \xi_k$ and this convention is assumed throughout the rest of this paragraph. ”

P. 5, l. 14: The sedimentation velocity is explicitly considered in the motion of superdroplets. I believe this counts the (admittedly small) contribution of sedimentation twice since it is already considered in the LES velocity vector u, according to (3).

We believe that the Referee has the large scale subsidence in mind and not sedimentation of droplets. Large scale subsidence is added as an RHS of the prognostic Eulerian variables. Adding it to the RHS of $\vec{u}$ in eq. (3) means that the velocity vector is moved downwards by large scale subsidence, but does not mean that the vertical velocity component includes the large scale subsidence velocity. Therefore this velocity is added to velocities of superdroplets and that way it is included only once as it should be.

P. 5, ll. 23 24: Equation (14) is still Eulerian in the sense that is contains an advection term.

The equation is now written in a form that is usually referred to as Lagrangian: $D_t\psi = R$.

P. 6, ll. 3 5: State clearly that π is the pressure perturbation. Furthermore, I think the introduction [that] it is characteristic for anelastic models that the pressure perturbation does not follow the

ideal gas law causes more confusion than clarification. I would omit it.

The sentence has been changed to:
" Pressure perturbation π is adjusted so that velocity field satisfies eq.(7). "

P. 6, 13 14, Shima et al. (2009) were not the first to advocate the integration of the squared wet radius. See, e.g., J.-P. Chen (1992): Numerical simulations of the redistribution of atmospheric trace chemicals through cloud processes (Doctoral dissertation, Pennsylvania State University), especially his Eq. (3.81).

We added a citation of the PhD thesis of J.-P. Chen.

P. 6, ll. 18 19: In what sense is condensation a fast process here? I think you need to be more specific. Arnason and Brown (1971, doi:10.1175/1520-0469(1971)028¡0072:GOCDBC¿2.0.CO;2) showed that for condensation a timestep corresponding to the phase relaxation timescale is sufficient, i.e., about 1 s or even longer for clean clouds. The requirement for a 0.1 s timestep arises, in my eyes, from the rapid change in droplet radius during growth at small radii. This is a well-known feature of stiff differential equations, as it is the case for the diffusional growth equation for droplets. Furthermore, how do you know that a sub-stepping timestep of 0.1 s is sufficient? In similar simulations of Grabowski et al. (2011, doi:10.1016/j.atmosres.2010.10.020) an initial timestep of 10 -6 s that might increase to 0.1 s is used to integrate the diffusional growth equation (see their Appendix). Of course, they integrated the linear growth equation (dr/dt) and not the quadratic (dr 2 /dt) as done here. But additional stand-alone integrations of superdroplets with different aerosol masses and a prescribed supersaturation using different timestep lengths are necessary to verify if a 0.1 s sub-timestep is actually sufficient.

What we mean by fast process is that it needs to be resolved on shorter time scales than other processes, precisely because small droplet radius changes rapidly by condensation. We state that the 0.1 s time step is sufficient based on tests we did in kinematic stratocumulus setup, in which concentration of cloud droplets converged for 0.1 s. It is possible that in other setups, e.g. with giant aerosols or stronger updrafts, a shorter time step would have to be used. The following text has been added to the manuscript:
" Condensation can rapidly change radii of small droplets. Therefore to correctly model condensation, in particular during the crucial moment of droplet activation, it is necessary to model condensation with a relatively short time step. Tests we performed in a kinematic 2D model of stratocumulus clouds have shown that number of activated droplets converges for condensation time step of around 0.1s. "

Based on our own experience and on personal communications with Shinichiro Shima, the relatively long time step of 0.1 s can be used thanks to the fact that we use the predictor-corrector algorithm described in the paper and that we solve growth equation for r^2 and not for r.

P. 8, l. 8: Consider changing a pair to the same pair for clarity.

Done

P. 8, l. 26: RHS of what?

For clarity, we changed that sentence to:
" In principle, liquid water is resolved by the SDM and could be diagnosed from the super-droplet size spectrum each time it is needed in the buoyancy term in eq. (3) or radiative term in eq. (4) "

P. 8, ll. 30 31: These sentences contradict each other since the UWLCM contains an LCM and an LES. Therefore, specify [all] of the model dependent variables more precisely.

Changed to:
" Eulerian dependent variables of the model are co-located. "

P. 9, ll. 9 10, Fig. 2: Figure 2 confuses me. If only the shaded part is used as a coalescence cell, certain volumes filled with superdroplets are neglected in the collection process. However, I do not believe that this is what the authors are doing. Could it be the case that the lowest line of grid point always equals the first, and that the right-most column of grid point equals the left-most? In other words, how do the authors implement so-called ghost layers of grid points to facilitate a cyclic model domain?

Superdroplets fill only the shaded region, as stated on p.9 l. 2:
" Super-droplets are restricted to the physical space, which is the shaded region in fig. 2. "

The domain is cyclic in horizontal directions, so left-most grid points (i.e. nodes of the primary grid) are equal to the right-most. This is not true for lower-most and upper-most grid points, because domain is rigid in the vertical direction. Ghost layers are implemented in such way that arrays stored in memory are larger than the grid shown in fig.2 and processes exchange values of ghost layers.

P. 9, l. 18 19: Important for the formation of drizzle is the is the microphysical model, and usually not the LES dynamical core.

In our understanding, a LES model of cloud needs to include some microphysical model. Therefore by LES model we mean dynamical core + microphysical model.

P. 10, ll. 25 26: Please comment on these options if they are essential for the conducted simulations. If they are not essential, I would omit this sentence for clarity.

Using other options would affect results, e.g. by giving more numerical diffusion. Therefore we chose to keep this sentence in case in future someone would try to reproduce our results.

P. 10, ll. 30 32: State that turbulence in two dimensions behaves fundamentally different from turbulence in three dimensions.

Added:
" However, it has to be kept in mind that the turbulence behavior in 2D is fundamentally different from 3D. "

P. 10, l. 33 p. 11, l. 1: Small random perturbations are not the reason for the variability, it is a fundamental property of a chaotic system, reacting to small changes in the initial values.

Because of the random perturbation, initial conditions are a little different for each run. Since the system is chaotic, small differences in initial conditions result in large differences after some time. Therefore we think that the statement that small initial perturbations cause variability is correct. If there were no random perturbations of initial conditions and microphysics were deterministic, each run would give the same result even though the system is chaotic, given that numerical calculations are exactly reproducible.

Figs. 3 6: For the final version of this manuscript, please make sure that the location of the figures matches the text.

We are doing our best. Additional formatting will need to be done afterwards, as the manuscripts in GMD are in a double column layout.

P. 12, ll. 1 2: The entrainment is usually not calculated from the increase of the inversion height alone. Commonly, the subsidence velocity at cloud top height is subtracted.

The sentence has been removed from the manuscript. Entrainment rate does take into account subsidence velocity. Definition of entrainment rate is now given in the caption of fig. 3 using symbols defined in Appendix A:

” Time series of the domain averaged liquid water path, entrainment rate (equal to $dz_i/dt + w_{\mathrm{LS}}z_i$), maximum of vertical velocity variance, surface precipitation, concentration of cloud droplets in cloudy cells and cloud base height. ”

P. 12, ll. 23 24: I suggest rewriting this sentence to: [...] where the autoconversion efficiency increases with N SD .

Done.

P. 12, ll. 26 28: Since the characteristics of turbulence in two dimensions are fundamentally different from three dimensions, the better agreement with observations must be seen as purely coincidental.

We agree. Still it is interesting to see.

P. 13, l. 29: Why is the iLES approach responsible for the simulated behavior of the third moment of the vertical velocity?

We suspected that based on Pressel et al. 2017 (doi:10.1002/2016MS000778). New simulations with SGS scheme, added in the revised paper, show that it is true - adding the SGS scheme gives skewness in agreement with reference models.

P. 13, l. 33 p. 15, l. 2: Spurious cloud edge supersaturations are known to result in the artificial activation of cloud droplets at the top of stratocumulus (e.g., Stevens et al. 1996, doi:10.1175/ 1520- 0493(1996)124,1034:TSPOCE.2.0.CO;2). Physical activations are largely impossible there since the top of stratocumulus is not dominated by strong, long-lasting updrafts resulting in physical supersaturations.

The sentence has been removed from the revised version.

P. 17, ll. 1 2: Maybe it is worthwhile to add references to the models DHARMA and RAMS.

Definitely, references have been added.

P. 17, sec. 4.5: How is activation determined? I assume a droplet is considered activated when it exceeds a critical radius. This is a valid assumption if the aerosol is small, and diffusional growth is not kinetically limited. However, for aerosols smaller than 0.1 m, the typical timescale for activation is usually similar or even smaller than the timestep of the applied model, making the treatment of activation in UWLCM, DHARMA, and RAMS practically identical. The

activation timescale becomes only important if the aerosol is large, typically larger than 0.1 m in radius, for which the critical radius exceeds a couple of micrometers. However, once located in a saturated environment, these inactivated particles exhibit behavior very similar to regularly activated droplets once their wet radius exceeds one micrometer, beyond which curvature and solute effects are usually negligible. Accordingly, the reduced susceptibility of aerosol activation on the cloud-base supersaturation maximum might also be just a result of the applied criterion for activation, which is not appropriate for the entire aerosol spectrum.

Droplet is considered activated when it becomes a cloud droplet, i.e. when its radius exceeds 0.5 μm (cloud droplet definition is in the caption of table A1). We also tested the definition assumed by the Referee, i.e. that activation happens when droplet radius exceeds the critical radius. Profiles of N_c are very similar for both definitions.

In our opinion, UWLCM treats activation of small aerosol (smaller than 0.1 μm) differently than DHARMA or RAMS. Assume that after model timestep supersaturation in a given cell exceeds critical supersaturation. In DHARMA and RAMS that means that some droplets are activated. In UWLCM, condensation is resolved with a timestep of 0.1 s, shorter than timescale of activation of most aerosols. Therefore condensation can decrease supersaturation to values lower than critical supersaturation before any droplets exceed critical radius, hence it is possible that no droplets are activated.

P. 17, ll.. 26 27: Please clarify: The cloud-base supersaturation maximum still causes activation in UWLCM, but it might not have an as immediate effect as in other cloud models because of the (presumably) applied criterion for activation (see last comment).

See answer to the last comment.

P. 19, ll. 10 11: I agree, that the number of superdroplets has no impact on domain-averaged quantities. However, it might be worthwhile to refer to the study of Schwenkel et al. (2018, doi:10.5194/gmd-11-3929-2018) in which small-scale effects of the superdroplets concentration are addressed. Technical Comments

This issue is now addressed in section 4.2:

" For example, larger number of SDs would probably be needed in simulations in which SDs have more attributes, e.g. when modeling aqueous chemistry. Also, we expect that observables other than domain averages, e.g. related to the spatial structure of a cloud, are more sensitive to the number of SDs. Schwenkel et al. (2018) present in more detail how cloud structure depends on the number of SDs. "

P. 5, l. 4: Change format of citation: [...] in Gillespie (1972), [...], not [...] in (Gillespie, 1972), [...]

Done.

P. 10, l. 16: Change format of citation: [...] in Ackerman et al. (2009), [...], not [...] in (Ackerman et al., 2009), [...]
Done.

Answer to the Anonymous Referee #2.

Major Comments

1. As the main components have been described elsewhere, and the coupling of the two parts seems to be rather straightforward, a stronger emphasis could be put on the model verification. You compare it to an ensemble of 11 reference models with quite some spread. But no one knows what the truth is. So far, I am not really sure what conclusions are to be drawn from your comparison and how I should interpret your results? Can you conclude anything e.g. from the fact that your model lies above or below the ensemble mean for some physical quantity? Better describe what you expect from your comparison exercise. Focusing on one specific test case gives only a snapshot of the models overall behavior and it is not clear how robust and general your findings are. It would be interesting to see how your model behaves in another well-chosen test case.

The paper introduces a new model, which, like almost any other model, is based on the research of others. The equations solved, numerical methods, etc. have been used before. However, in order for other researchers to be able to use the model or to make a comparison with it, it is important to present these known components in one place. Besides providing such reference, new methods for coupling of the components are presented in the paper. We do not agree that these methods are straightforward. For example, Shima et al. (2009) uses other methods for spatial coupling and condensation substepping. Also, time stepping algorithm presented in Fig. 1 is not straightforward - time stepping could be done in many different ways.

The Dycoms RF02 setup is devised to reproduce observed clouds. Observations are the truth, albeit there are many difficulties in comparing modeling with observations. Nevertheless, models do reasonably well in reproducing observations. Therefore, if our model was far off from other models, that would indicate that something is wrong in it. Of course, if two models give slightly different results it is impossible to say that one is better than the other. Comparing a model with Lagrangian microphysics with an ensemble of other models is also a novelty - we are not aware of similar studies.

Besides showing that the model gives results in general agreement with other models, 2D and 3D tests in the revised paper have additional purposes. 2D simulations are done to test sensitivity of the microphyscisc scheme - something of interest for other users of Lagrangian microphysics. 3D simulations are done to test sensitivity of the model to the description of SGS turbulence, including a SGS model for motion of Lagrangian computational particles.

2. To be frank, resorting to the iLES approach comes in handy as you dont have to implement a SGS scheme. I could live with it if your model is purely Eulerian. As the Lagrangian model has no implicit numerical diffusivity (neither in spectral nor spatial space) and the iLES approach is not applicable in the microphysics part, SGS random perturbation velocities could be included in the transport equation of the superdroplets in order to mimic subgrid scale motions. However, without a proper SGS scheme that estimates TKE it is not straightforward to prescribe such perturbations. This shortcoming should be clearer mentioned.

We agree that this has been a major drawback of the initial manuscript. Therefore the 3D simulations section now contains a comparison of ILES vs Smagorinsky vs Smagorinsky + SGS perturbation of superdroplets. For details, please see the answer to the Major Comment 1. of Referee #1.

Minor comments

P1. Last row: Isnt libmpdata++ the dynamical core? What does it mean it is built on top of it?

libmpdata++ is designed to be applicable to variety of problems. This means that some aspects, such as numerical integration procedure or details of the SGS scheme, have to be defined in the software that uses libmpdata++. In addition, all forcings are implemented in UWLCM. The sentence has been changed to:

"The dynamical core is implemented using the the libmpdata++ software library"

p.4, l.22: Without defining what a collision between two SDs is, it makes no sense to say the probability needs to be increased. Please rephrase.

We rephrased it from "collision" to "coalescence" of SDs. What a coalescence of SDs is is defined right after eq. (13) (eq. 17 in the revised manuscript) that presents how probability needs to be increased.

p.5, l.14: I do not understand the inclusion of w LS. This would mean that the SDs move relative to the surrounding (Eulerian) air!?

Large-scale subsidence is not included in air velocity, but is implemented as a RHS. SDs are advected with air velocity, i.e. without subsidence. Therefore the subsidence velocity needs to be separately added to the SD velocity.

Sec 3.2.: The implementation of the various condensation algorithms is not clear to me. Given that new and old are known, you do a linear time interpolation between the two values. And the difference between the two approaches is the choice of the grid box from which you pick the values. What I stumble upon is the quantity new . Is it known beforehand? In my understanding, sub-stepping would simply mean that condensation (growth of droplets, depletion of water vapor and latent heat release) is treated with a smaller time step and clearly involves a dynamic update of the variables and q v in each sub- time step.

As stated in section 3.2:
"ψ_{new} [are] values of Eulerian variables before the start of the substepping algorithm in the current time step".

Therefore it is known beforehand. $\psi_{\mathrm{new}} - \psi_{\mathrm{old}}$ is a change of Eulerian variables caused by sources other than condensation, e.g. surface fluxes, radiation, advection, etc. When substepping, we do a linear interpolation of this change and at each substep we add to that changes caused by condenation. An exact mathematical description is given in the Appendix B.

Sec 3.3: I do not fully understand why you solve a prognostic equation for q l in the Eulerian model part. Wouldnt it suffice to diagnose q l from the SDs? I understand that q l is used for the computation of the buoyancy term (Eq. 6). Do you need it elsewhere? Can you estimate the error of using two different definitions of q l ? You write that you want to avoid an additional synchronization? Would this issue still matter in a parallelised implementation?

q_l is needed in buoyancy and radiation terms. Synchronization is needed precisely because our implementation is parallelised - calculations are done at the same time by CPU cores and by GPUs. As stated in Sec. 3.3, it would suffice to diagnose q_l from SDs each time it is needed:
"In principle, liquid water is resolved by the SDM and could be diagnosed from the super-droplet size spectrum each time it is needed in the buoyancy term in eq. (3) or radiative term in eq. (4)."

However, the buoyancy term is integrated with a trapezoidal rule, hence we need to know liquid water at the next time step: $q_l(n+1)$. In principle we could wait for GPUs to finish calculating advection, subsidence and sedimentation of SDs and then diagnose $q_l(n+1)$ from SDs and launch the pressure solver afterwards. However, plenty of computational time can be saved by running advection, subsidence and sedimentation in parallel with the pressure solver. This is achieved by adding the auxiliary q_l field. To clarify our approach, the paragraph now reads:

" Liquid water is resolved by the SDM and q_l could be diagnosed from the super-droplet size spectrum each time it is needed in the buoyancy term in eq. (3) or radiative term in eq. (4). Buoyancy is integrated with a trapezoidal scheme, which requires q_l after advection to be known. In a straightforward implementation, in which q_l is diagnosed from SDs after advection of SDs, pressure solver calculations can only be started after advection of SDs has been calculated. Then, there is little parallelism of calculations on GPUs and CPUs. To achieve more parallelism, we introduce an auxiliary Eulerian field for q_l. Value of q_l is diagnosed from SDs once per timestep, after condensation calculation. Then, q_l advection is done using a first-order accurate upwind scheme. Using the auxilliary q_l field, it is possible to calculate coalescence and motion of SDs simultaneously with calculations of advection of Eulerian fields and of the pressure problem. "

We expect the error associated with this procedure to be low, because q_l is diagnosed from SDs at each time step.

Sec 4.1: Can you comment why you use a split definition (Hall +Davies) for the collision efficiencies?

Hall (1980) does not give collision efficiencies for collisions of droplets that are both smaller than 10 μm, therefore for such collisions we use values from Davies (1972).

Sec 4.2: In particular, the differences between the per-cell and per-particle approach are so small that I am not fully convinced that the one is superior over the other one. It would also help to see the spread of the 10-member ensemble of a specific 2D simulation. Is it really significant that in the one case the N c -profiles slightly decrease with altitude, whereas in the other case they slightly increase? Can you be sure that in other test case, your finding (superiority of the per-particle) would be the same? This is one example why I recommend a second test case.

In the revised manuscript, substepping algorithms are tested using kinematic approach, i.e. both simulations are run with the same flow field. Results of single runs are compared, without averaging over an ensemble. This improved test case has shown that the *per-cell* algorithm works a little better for stratocumulus clouds, contrary to what we initially concluded. We also added a second test case for substepping that represents idealized advection of cloud edge. In that case, the *per-particle* algorithm works much better. These tests are described in Appendix B of the revised manuscript.

Typos, language issues and other formal things:

In general, the usage of articles a and the is not correct on several occasions. Sometimes you miss the article, sometimes it is misplaced. Please try your best, the rest will be handled by Copernicus services.

We are doing our best.

The Exner function pi should be defined close to Eq. 3

π is pressure perturbation, what is now stated in the sentence following Eq. 3. Definition of it remains in the table in the Appendix A.

p.4, l.16: collisionS

Fixed.

There is a difference between which and that: https://www.wisegeek.com/what-is-the- difference-between-that-and-which.htm Accordingly, which in p.4, l.19 and l.24 must be replaced by that. There might be more such mistakes.

Thank you for this language tip. Several more occurrences of "which" have been replaced with "that".

p.4, l. 23: dropletS

Fixed.

p.12, l.10. not sure if VAR is self-explaining?

It is now defined in the caption of Fig. 3.

p.12, l.31: visible IN

Fixed.

p.13, l.11: impact IN 3D simulations than IN 2D simulations

Fixed.

Caption of Fig. 4: Please correct On the vertical axis is height ...

Changed to:
" Vertical axis is altitude normalized by inversion height."

P.17, l.17: concentration

Fixed.

References

Schwenkel, J., Hoffmann, F., and Raasch, S.: Improving collisional growth in Lagrangian cloud models: development and verification of a new splitting algorithm, Geoscientific Model Development, 11, 3929–3944, 2018.

Shima, S.-i., Kusano, K., Kawano, A., Sugiyama, T., and Kawahara, S.: The super-droplet method for the numerical simulation of clouds and precipitation: A particle-based and probabilistic microphysics model coupled with a non-hydrostatic model, Quarterly Journal of the Royal Meteorological Society, 135, 1307–1320, 2009.

Smolarkiewicz, P. K., Khnlein, C., and Wedi, N. P.: A consistent framework for discrete integrations of soundproof and compressible PDEs of atmospheric dynamics, Journal of Computational Physics, 263, 185 – 205, https://doi.org/https://doi.org/10.1016/j.jcp.2014.01.031, URL `http://www.sciencedirect.com/science/article/pii/S0021999114000588`, 2014.

University of Warsaw Lagrangian Cloud Model (UWLCM) 1.0: a modern Large-Eddy Simulation tool for warm cloud modeling with Lagrangian microphysics

Piotr Dziekan, Maciej Waruszewski, and Hanna Pawlowska

Institute of Geophysics, Faculty of Physics, University of Warsaw, Poland

Correspondence: Piotr Dziekan (pdziekan@fuw.edu.pl)

Abstract. A new anelastic large-eddy simulation model with an Eulerian dynamical core and ~~a~~ Lagrangian particle-based microphysics is presented. The dynamical core uses the MPDATA advection scheme and the generalized conjugate residual pressure solver, while the microphysics scheme is based on the Super-Droplet Method. Algorithms for coupling of ~~the~~ Lagrangian microphysics with ~~the~~ Eulerian dynamics are presented, including spatial and temporal ~~discretizations~~ discretisations and a condensation ~~sub-stepping algorithm.~~ substepping algorithm. The model is free of numerical diffusion in the droplet size spectrum. Activation of droplets is modeled explicitly, making the model less sensitive to local supersaturation maxima than models in which activation is ~~parametrised~~parameterized. Simulations of a drizzling marine stratocumulus give results in agreement with other LES models. ~~Relatively~~ It is shown that in the Super-Droplet Method a relatively low number of computational particles is sufficient to obtain ~~the~~ correct averaged properties of a cloud, but condensation and coalescence have to be modeled with a time step of the order of 0.1 s. Simulations with and without explicit subgrid-scale turbulence model are compared. Effects of modeling subgrid-scale motion of super-droplets are investigated. ~~High computational performance is achieved thanks to the use of~~ The model achieves high computational performance by using GPU accelerators.

1 Introduction

In the last decade, Lagrangian particle-based cloud microphysics schemes have been drawing increasing attention. They are similar to ~~the~~ Eulerian bin schemes in that they explicitly model the size spectrum of droplets and explicitly resolve microphysical processes, but have a number of advantages over them (Grabowski et al., 2018b). One of the advantages is that ~~the~~ Lagrangian schemes have no numerical diffusion in the spectrum of droplet sizes. Several Lagrangian schemes for warm cloud microphysics have been developed so far (Andrejczuk et al., 2008; Shima et al., 2009; Riechelmann et al., 2012). Arguably, the most important difference between these schemes is in the way collision-coalescence is modeled. The coalescence algorithm used in the Super-Droplet Method (SDM) of Shima et al. (2009) seems to be the most promising, as it was found to be the most accurate of the coalescence algorithms used in various Lagrangian microphysics schemes (Unterstrasser et al., 2017, where it is called the "all-or-nothing" algorithm). A numerical implementation of the SDM is a major part of the libcloudph++ library (Arabas et al., 2015) developed by the cloud modeling group at the University of Warsaw.

In this paper, we document development of a new Large-Eddy Simulation (LES) model called the University of Warsaw Lagrangian Cloud Model (UWLCM). It is an anelastic model with a finite-difference Eulerian dynamical core and a Lagrangian microphysics scheme. The Lipps-Hemler anelastic approximation (Lipps and Hemler, 1982) is used, which is applicable to a wide range of atmospheric flows (Klein et al., 2010; Smolarkiewicz, 2011). The dynamical core is ~~built on top of~~ implemented using the libmpdata++ software library (Jaruga et al., 2015) also developed by the cloud modeling group at the University of Warsaw. libmpdata++ is a collection of solvers for the generalized transport equation. In libmpdata++, advection is modeled using the multidimensional positive-definite advection transport algorithm (MPDATA) – see Smolarkiewicz (2006) for a recent review. Liquid water is modeled with the ~~Lagrangian~~ SDM implemented in libcloudph++. We do not assume any artificial categorization of liquid water particles. In consequence, all particles, i.e. humidified aerosols, cloud droplets and rain drops, evolve according to the same set of basic equations.

One of the key reasons for developing a new model is to use a modern software development approach. ~~All of the~~ The model code is written in the C++ programming language and makes use of many mature libraries available in that language (e.g. Blitz++, Boost, Thrust). The code is open-source and under a version-control system. A set of automated tests greatly ~~simplifies code development~~helps in ensuring the correctness of the model. The automated tests include a 2D moist thermal simulation, a 2D kinematic stratocumulus simulation and a test of different combinations of model options. Moreover, modeling of physical processes, e.g. condensation, advection, coalescence, sedimentation, is tested separately by the libmpdata++ and libcloudph++ test suites. UWLCM makes efficient use of modern computers ~~,~~ that have both central processing units (CPUs) and graphics processing units (GPUs). The Eulerian computations of the dynamical core are done on CPUs and, simultaneously, ~~the~~ Lagrangian microphysical computations are done on GPUs. However, it is also possible to run the Lagrangian microphysics on CPUs.

Some results obtained using earlier versions of UWLCM ~~are already~~ have already been published. In Grabowski et al. (2018a), UWLCM was used to model a 2-dimensional moist thermal and in Grabowski et al. (2018b), an idealized 3-dimensional cumulus cloud was modeled. Here, we present simulations of a drizzling marine stratocumulus using the DYCOMS-II RF02 setup. UWLCM results are compared with 11 LES models that took part in the Ackerman et al. (2009) intercomparison. Sensitivity of UWLCM to the parameters of the microphysics scheme and to the description of the subgrid-scale (SGS) turbulence is studied. It is of particular interest how much drizzle a LES model with Lagrangian microphysics produces, compared to models with bin or bulk microphysics that took part in the intercomparison. To our knowledge, LES simulations with warm cloud Lagrangian microphysics were used to study drizzling stratocumulus only by Andrejczuk et al. (2008, 2010). This type of models was more often employed to study cumulus clouds (Riechelmann et al., 2012; Naumann and Seifert, 2015; Arabas and Shima, 2013; Hoffmann et al., 2015, 2017).

Section 2 presents the governing equations of the model, section 3 describes the numerical algorithms for solving these equations, the stratocumulus simulation results are discussed in section 4, and section 5 contains a summary and planned developments of the model. A list of symbols used and their definitions are given in appendix A, appendix B ~~defines the two sub-stepping algorithms tested in the paper~~ compares two substepping algorithms for condensation and appendix C contains a brief description of the software implementation of the model.

2 Governing equations

2.1 Eulerian variables

Eulerian prognostic variables of the model are the potential temperature θ, the water vapor mixing ratio q_v and the air velocity $\boldsymbol{u}$. Equations governing time evolution of these variables are obtained through the Lipps-Hemler approximation, which relies on the assumption that the atmosphere does not depart far from some stationary state, called the *reference* state (Lipps and Hemler, 1982). The reference state is assumed here to be a dry, hydrostatically balanced state with constant stability S^r. S^r is equal to the average stability of the sounding used to initialize ~~the~~ a simulation. Surface density and pressure of the reference state are equal to those of the initial sounding. Vertical profiles of potential temperature and density of dry air in the reference state are (Clark and Farley, 1984):

$$\theta^r(z) = \theta_v^0 \exp(S^r z), \tag{1}$$

$$\rho_d^r(z) = \rho^0 \exp(-S^r z)\left[1 - \frac{g}{c_{pd} S^r \theta_v^0}\left(1 - \exp(-S^r z)\right)\right]^{(c_{pd}/R_d)-1}, \tag{2}$$

where θ_v^0 and ρ^0 are values of the virtual potential temperature and of the air density taken from the initial sounding at the ground level. An auxiliary *environmental* state is introduced to increase ~~precision~~ accuracy of numerical calculations ~~(Smolarkiewicz et al., 2~~ (Smolarkiewicz et al., 2014, 2019). It is a hydrostatically balanced moist state with stationary profiles $\theta^e(z)$, $p^e(z)$, $T^e(z)$, $q_v^e(z)$ and $q_l^e(z)$ calculated from the initial sounding. If the initial sounding is supersaturated, all supersaturation is assumed to be condensed in the environmental state.

The set of anelastic Lipps-Hemler equations (Lipps and Hemler, 1982; Grabowski and Smolarkiewicz, 1996; Clark and Farley, 1984) that govern time evolution of the Eulerian prognostic variables is

$$D_t \boldsymbol{u} = -\nabla \pi + \boldsymbol{k} B + \boldsymbol{F}_u + \boldsymbol{\mathcal{D}}_u, \tag{3}$$

$$D_t \theta = \frac{\theta^e}{T^e}\left(\frac{l_v}{c_{pd}} C\right) + F_\theta + \mathcal{D}_\theta, \tag{4}$$

$$D_t q_v = -C + F_{q_v} + \mathcal{D}_{q_v}, \tag{5}$$

where D_t denotes the material derivative: $D_t = \partial_t + \boldsymbol{u} \cdot \nabla$ and π is the normalized pressure perturbation. Following Grabowski and Smolarkiewicz (1996), ~~the~~ buoyancy is defined as

$$B = g\left[\frac{\theta - \theta^e}{\theta^r} + \epsilon(q_v - q_v^e) - (q_l - q_l^e)\right]. \tag{6}$$

The condensation rate C in eqs. (4) and (5) and the liquid water mixing ratio q_l in eq. (6) come from the Lagrangian microphysics scheme. The terms F_* represent a total forcing due to surface fluxes, radiative heating/cooling, large-scale subsidence and absorbers, while the terms $\mathcal{D}_*$ represent contributions from a SGS turbulence model. The dry-air density is assumed to be equal to the reference state density profile ρ_d^r and, ~~what is characteristic~~ characteristically for the anelastic approximation, the dry-air density at given position does not change with time: $\partial_t \rho_d^r = 0$. By putting $\partial_t \rho_d^r = 0$ into the continuity equation, the following constraint on the velocity field is obtained:

$$\nabla \cdot (\rho_d^r \boldsymbol{u}) = 0, \tag{7}$$

which will be referred to as the *anelastic constraint*. Throughout the model, the pressure is assumed to be equal to the environmental pressure profile $p^e(z)$. The only exception is the pressure gradient term appearing in eq. (3), in which the pressure is adjusted so that $\boldsymbol{u}$ satisfies the anelastic constraint (eq. 7) (Lipps and Hemler, 1982; Grabowski and Smolarkiewicz, 1996).

UWLCM offers two methods for modeling diffusion of Eulerian variables due to the SGS turbulence. The first is an implicit LES (ILES) approach, in which there is no explicit parametrisation of SGS mixing, i.e. $\mathcal{D}_* \equiv 0$. Instead, numerical diffusion of the advection scheme is used to mimic the SGS turbulence (Grinstein et al., 2007). The MPDATA algorithm is argued to be well-suited for ILES simulations (Margolin and Rider, 2002; Margolin et al., 2006). The other method is a Smagorinsky-type model (Smagorinsky, 1963; Lilly, 1962) with the SGS effects parametrised as

$$\boldsymbol{\mathcal{D}}_u = \frac{1}{\rho_d^r} \nabla \cdot (\rho_d^r K_m \boldsymbol{E}), \tag{8}$$

$$\mathcal{D}_\theta = \frac{1}{\rho_d^r} \nabla \cdot (\rho_d^r K_h \nabla \theta), \tag{9}$$

$$\mathcal{D}_{q_v} = \frac{1}{\rho_d^r} \nabla \cdot (\rho_d^r K_q \nabla q_v), \tag{10}$$

where K_m is the eddy viscosity, K_h and K_q are the eddy diffusivities, and $\boldsymbol{E} = \nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T - \frac{2}{3}(\nabla \cdot \boldsymbol{u})\boldsymbol{I}$ is the deformation tensor. The eddy viscosity is given by

$$K_m = \begin{cases} (c_s \lambda)^2 |\boldsymbol{E}| \left(1 - \frac{K_h}{K_m}\mathrm{Ri}\right)^{1/2}, & \text{if } \frac{K_h}{K_m}\mathrm{Ri} < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where c_s is the Smagorinsky constant, λ is the mixing length, and Ri is the Richardson number. The eddy diffusivities are

$$K_h = K_q = K_m/\mathrm{Pr}, \tag{12}$$

where Pr is the Prandtl number. Following Schmidt and Schumann (1989) the mixing length is set to $\lambda = \min(\Delta, c_L z)$. Given highly anisotropic grid cells used in stratocumulus simulations we set $\Delta = \Delta z$, as in the Colorado State University System for Atmospheric Modeling (Khairoutdinov and Randall, 2003). The values of the numerical constants are taken following Schmidt and Schumann (1989) as $c_s = 0.165$, $\mathrm{Pr} = 0.42$, and $c_L = 0.845$.

2.2 Lagrangian particles

Liquid water is modeled with a Lagrangian, particle-based microphysics scheme ~~of~~ from the libcloudph++ library (Arabas et al., 2015). It is an implementation of the Super-Droplet Method (SDM) (Shima et al., 2009). The key idea is to represent all liquid particles using a small number of computational particles, called super-droplets (SDs). Each SD represents a large number of real particles. The number of real particles represented by a given SD is called the multiplicity (also known as the weighting factor), and is denoted by ξ. Other attributes of SDs are the dry radius r_d, the wet radius r, the hygroscopicity parameter κ and the position $\boldsymbol{x}$ in the model domain.

The condensational growth rate of a SD is equal to that of a single real particle. We calculate it using the Maxwell-Mason approximation (see Arabas et al. 2015):

$$r\frac{dr}{dt} = \frac{D'_{\text{eff}}}{\rho_w}\left(\underline{1}q_v - \underline{\frac{a_w\left(r,r_d,\kappa\right)\exp(A/r)}{\phi}}q_{vs}a_w\left(r,r_d,\kappa\right)\exp(A/r)\right), \tag{13}$$

where

$$\frac{1}{D'_{\text{eff}}} = \left(D\rho_d\underline{q_v}\right)^{-1} + K^{-1}\underline{\frac{1}{\phi}}q_{vs}\frac{l_v}{T}\left(\frac{l_v}{R_vT}-1\right)\underline{,} \tag{14}$$

and water activity is calculated using the κ-Köhler parametrisation (Petters and Kreidenweis, 2007):

$$a_w\left(r,r_d,\kappa\right) = \frac{r^3-r_d^3}{r^3-r_d^3\left(1-\kappa\right)}. \tag{15}$$

~~and, following~~Following Lipps and Hemler (1982), the relative humidity is defined as $\phi = q_v/q_{vs}$ ~~, with~~ and the saturation water vapor mixing ratio is calculated using the formula ~~$q_{vs} = 0.622e_s/(p^e-e_s)$~~$q_{vs} = (R_d/R_v)e_s/(p^e-e_s)$. Formulas for ~~the parameters appearing in eqs. (13) and (14)~~ A and l_v can be found in ~~Arabas et al. (2015)~~Arabas and Pawlowska (2011). The vapor and heat diffusion coefficients D and K include gas kinetic and ventilation effects and are evaluated as in ~~Arabas and Pawlowska (2011) (see eqs. 20 and 21 therein)~~Arabas et al. (2015).

Collision-coalescence of droplets is treated as a stochastic process (Gillespie, 1972). ~~Collision~~ Collisions are possible only between droplets that are located within the same spatial cell, called the coalescence cell. It is assumed that coalescence cells are well-mixed, i.e. that droplets are randomly and uniformly distributed within a coalescence cell. Then, the probability that any two real droplets j and k ~~, which~~ that are located in the same coalescence cell ~~, collide~~ coalesce during the time interval Δt_c is given by the equation (Shima et al., 2009)

$$P_{j,k} = K_{j,k}\frac{\Delta t_\text{c}}{\Delta V}, \tag{16}$$

where $K_{j,k}$ is the coalescence kernel for these two droplets and ΔV is the volume of the coalescence cell. ~~Probability of a collision between~~ The probability of coalescence of SDs needs to be increased to account for the fact that each SD represents a large number of real droplets. ~~Probability~~ The probability that any two SDs j and k ~~, which~~ that are in the same coalescence cell ~~, collide~~ coalesce during the time interval Δt_c is related to the probability of ~~collision~~ coalescence of real droplets in the following manner (Shima et al., 2009):

$$P^{\text{SD}}_{j,k} = \underline{\max}\underline{\xi_j,}\xi_k P_{j,k}\underline{.}, \tag{17}$$

where SDs are labeled so that $\xi_j \leq \xi_k$ and this convention is assumed throughout the rest of this paragraph. Coalescence of the two SDs is interpreted as a coalescence of ξ_j pairs of real droplets. Each pair consists of one real droplet represented by the j-th SD and one real droplets represented by the k-th SD~~and SDs are labeled so that $\xi_j \leq \xi_k$~~. The remaining $\xi_k - \xi_j$ real droplets represented by the k-th SD are not affected by the coalescence of these two SDs.

Such treatment of coalescence, sometimes referred to as the *all-or-nothing* algorithm, assures that the number of SDs does not increase due to the collision-coalescence. This algorithm was found to give the best results in a recent comparison of various coalescence algorithms used in Lagrangian schemes for microphysics (Unterstrasser et al., 2017). Dziekan and Pawlowska (2017) showed that the *all-or-nothing* algorithm produces correct realizations of the stochastic coalescence process described in ~~(Gillespie, 1972)~~Gillespie (1972), but only for $\xi = 1$. For $\xi > 1$, an average over realizations of the *all-or-nothing* algorithm is in good agreement with the expected value of the stochastic process, but the variability between realizations is much higher. This is because the number of SDs is much smaller than the number of real droplets. In consequence, the statistical sample for $\xi > 1$ is much smaller than in the more realistic case of $\xi = 1$. The collision-coalescence algorithm is not the only cause of the high variability for $\xi > 1$. ~~The motion~~ Motion of SDs is also expected to give a high variability, because when a SD moves from one spatial cell to another, a large number of real particles is abruptly moved between these cells. It is not certain if the high variability in the SDM associated with collision-coalescence of SDs and motion of SDs has ~~some~~ any impact on averaged properties of a modeled cloud. To determine if it does have an effect, we conduct simulations for various number of SDs (see section 4.2).

Super-droplets are treated as non-inertial particles that always sediment with ~~the~~ their terminal velocity. ~~The~~ There is an option to model diffusion of liquid water due to the SGS turbulence by adding a random velocity component $\boldsymbol{u'}_{\mathrm{SD}}$ that is specific to each SD. Each component of this velocity perturbation evolves according to eq. (10) from Grabowski and Abade (2017). It is important to note that this SGS velocity can only be added when the Smagorinsky scheme is used for Eulerian variables. Altogether, velocity of a SD is equal to ~~$\boldsymbol{u}_{SD} = \boldsymbol{u} + (0,0,w_t) + (0,0,w_{\mathrm{LS}})$~~$\boldsymbol{u}_{\mathrm{SD}} = \boldsymbol{u} + \boldsymbol{u'}_{\mathrm{SD}} + (0,0,w_t) + (0,0,w_{\mathrm{LS}})$. This formula represents the combined effects of advection by air flow, SGS turbulence, sedimentation and large-scale subsidence.

3 Numerical algorithms

3.1 Numerical integration of Eulerian equations

Numerical integration of the governing Eulerian equations is done using the MPDATA algorithm implemented in libmpdata++ (Jaruga et al., 2015). MPDATA is an algorithm for solving the generalized transport equation (Smolarkiewicz, 2006)

$$\partial_t (G\psi) + \nabla \cdot (G\boldsymbol{u}\psi) = GR, \tag{18}$$

where ψ is a scalar field advected by the velocity field $\boldsymbol{u}$, R is the source/sink right-hand side (RHS) and G can represent the fluid density, the Jacobian of coordinate transformation or their product. The equivalent of eq. (18) in ~~the~~ Lagrangian description is:

$$\cancel{\partial}D_t\psi\cancel{+\cdot\nabla\psi} = R. \tag{19}$$

Equation (3) for components of vector $\boldsymbol{u}$ and eqs. (4) and (5) have the same form as eq. (19). Equation (19) introduces notation that is convenient for presenting the numerical integration procedure of UWLCM. All RHS terms, except buoyancy and pressure gradient terms in eq. (3), are integrated with the forward Euler method. These terms are denoted by R_{E}. The buoyancy and pressure gradient terms, denoted by R_{T}, are applied using the trapezoidal rule. The integration algorithm is:

$$\psi^{[n+1]} = ADV\left(\psi^{[n]} + \Delta t R_{\mathrm{E}}^{[n]} + 0.5\Delta t R_{\mathrm{T}}^{[n]}, \boldsymbol{u}^{[n+1/2]}\right) + 0.5\Delta t R_{\mathrm{T}}^{[n+1]}, \tag{20}$$

where $ADV(\psi, \boldsymbol{u})$ is an operator representing MPDATA advection of a scalar field ψ by the velocity field $\boldsymbol{u}$. Superscripts denote the time level. The mid-time-level velocity field $\boldsymbol{u}^{[n+1/2]}$ is obtained by linear extrapolation from $\boldsymbol{u}^{[n-1]}$ and $\boldsymbol{u}^{[n]}$.

~~It is characteristic for anelastic models that the pressure perturbation does not follow the ideal gas law, but~~ Pressure perturbation π is adjusted so that the velocity field satisfies eq. (7). By applying eq. (7) to the equation for $\boldsymbol{u}^{[n+1]}$ ~~discretized~~ discretised in the form of eq. (20), the following elliptic equation for $\pi^{[n+1]}$ is obtained:

$$\nabla\cdot\left[\rho_d^r\left(\hat{\boldsymbol{u}} + 0.5\Delta t\boldsymbol{k}B^{[n+1]} - 0.5\Delta t\nabla\pi^{[n+1]}\right)\right] = 0, \tag{21}$$

where

$$\hat{\boldsymbol{u}} = ADV\left[\boldsymbol{u}^{[n]} + \Delta t\left(\boldsymbol{F}_u^{[n]} + \boldsymbol{\mathcal{D}}_u^{[n]}\right) + 0.5\Delta t\left(-\nabla\pi^{[n]} + \boldsymbol{k}B^{[n]}\right), \boldsymbol{u}^{[n+1/2]}\cancel{,}\right] \tag{22}$$

and the thermodynamic fields required in $B^{[n+1]}$ are already available when the equation has to be solved. The pressure problem stated in eq. (21) is solved with the generalized conjugate residual solver (Smolarkiewicz and Margolin, 2000; Smolarkiewicz and Szmelter, 2011).

3.2 Numerical algorithms for super-droplets

For numerical reasons~~the~~, condensational growth of SDs is solved in terms of the squared wet radius~~, as advocated by Shima et al. (2009)~~ (Chen, 1992; Shima et al., 2009). Integration of eq. (13) is done with a scheme that is implicit with respect to the wet radius and explicit with respect to q_v and θ:

$$r^{2[n+1]} = r^{2[n]} + \Delta t\left.\frac{dr^2}{dt}\right|_{r^{2[n+1]}, q_v^{[n]}, \theta^{[n]}}. \tag{23}$$

Solution for eq. (23) is found with a predictor-corrector procedure. We refer the reader to Arabas et al. (2015) for details of this procedure. Condensation ~~is a fast process and the above procedure converges for time steps~~ can rapidly change radii of small droplets. Therefore to correctly model condensation, in particular during the crucial moment of droplet activation, it is necessary to model condensation with a relatively short time step. Tests performed in a kinematic 2D model of stratocumulus clouds have shown that the number of activated droplets converges for condensation time step of around 0.1s. A typical time

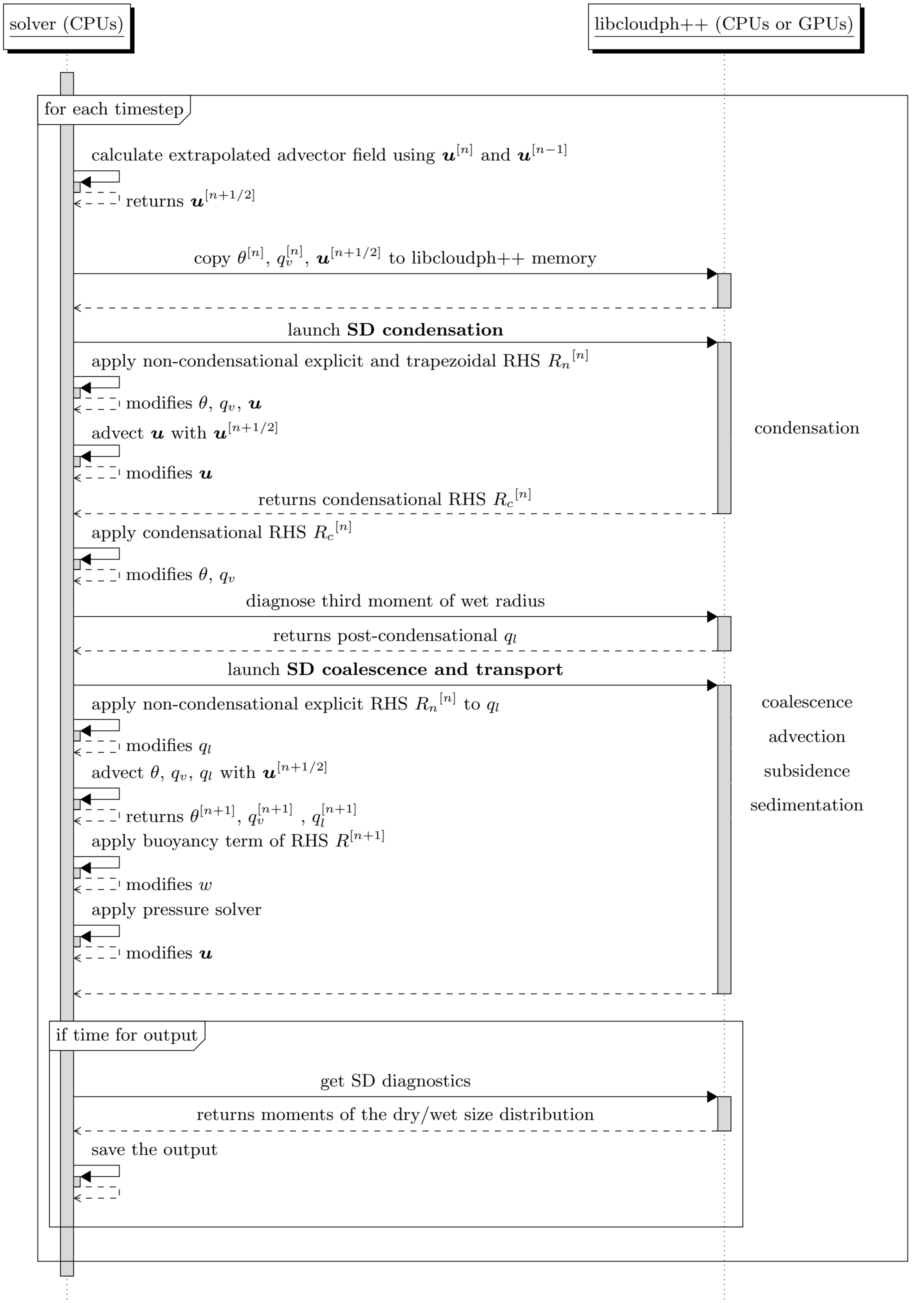
solver (CPUs)
libcloudph++ (CPUs or GPUs)
for each timestep
calculate extrapolated advector field using $\boldsymbol{u}^{[n]}$ and $\boldsymbol{u}^{[n-1]}$
returns $\boldsymbol{u}^{[n+1/2]}$
copy $\theta^{[n]}$, $q_v^{[n]}$, $\boldsymbol{u}^{[n+1/2]}$ to libcloudph++ memory
launch **SD condensation**
apply non-condensational explicit and trapezoidal RHS $R_n{}^{[n]}$
modifies θ, q_v, $\boldsymbol{u}$
advect $\boldsymbol{u}$ with $\boldsymbol{u}^{[n+1/2]}$
modifies $\boldsymbol{u}$
condensation
returns condensational RHS $R_c{}^{[n]}$
apply condensational RHS $R_c{}^{[n]}$
modifies θ, q_v
diagnose third moment of wet radius
returns post-condensational q_l
launch **SD coalescence and transport**
apply non-condensational explicit RHS $R_n{}^{[n]}$ to q_l
modifies q_l
advect θ, q_v, q_l with $\boldsymbol{u}^{[n+1/2]}$
returns $\theta^{[n+1]}$, $q_v^{[n+1]}$, $q_l^{[n+1]}$
coalescence
advection
subsidence
sedimentation
apply buoyancy term of RHS $R^{[n+1]}$
modifies w
apply pressure solver
modifies $\boldsymbol{u}$
if time for output
get SD diagnostics
returns moments of the dry/wet size distribution
save the output

Figure 1. UML sequence diagram showing the order of operations within a single time step. Calls in boldface start microphysical calculations that are done on GPUs simultaneously with solver operations done on CPUs. The RHS is divided into condensational and non-condensational parts, $R = R_n + R_c$.

step Δt of a LES model is around 1s. Therefore it is necessary to do several condensation time steps in a single LES time step, ~~what we call sub-stepping~~a procedure we call substepping. To explain the idea of the ~~sub-stepping~~ substepping algorithm, we introduce the following notation: S_c for the number of ~~sub-steps~~substeps, $\boldsymbol{\psi} = (\theta, q_v)$ for a vector of ~~the~~ Eulerian variables, $\boldsymbol{\psi}_{\text{old}}$ for values of ~~the~~ Eulerian variables after the ~~sub-stepping~~ substepping algorithm finished in the previous time step and $\boldsymbol{\psi}_{\text{new}}$ for values of ~~the~~ Eulerian variables before the start of the ~~sub-stepping~~ substepping algorithm in the current time step. In the first ~~sub-step, the~~ substep, Eulerian variables are set to $\boldsymbol{\psi}_{\text{old}} + \frac{\boldsymbol{\psi}_{\text{new}} - \boldsymbol{\psi}_{\text{old}}}{S_c}$ and then condensation is calculated using the procedure defined in eq. (23). Please note that this condensation procedure changes ~~the~~ Eulerian variables. In each subsequent time step, $\frac{\boldsymbol{\psi}_{\text{new}} - \boldsymbol{\psi}_{\text{old}}}{S_c}$ is added to ~~the~~ Eulerian variables and then the condensation procedure is run again. Two types of the ~~sub-stepping~~ substepping algorithm are considered ~~, which~~ that differ only in the spatial cell from which the value of $\boldsymbol{\psi}_{\text{old}}$ is diagnosed. In the *per-particle* algorithm, $\boldsymbol{\psi}_{\text{old}}$ is diagnosed from the cell in which the given SD was in the *previous* time step. In the *per-cell* algorithm, $\boldsymbol{\psi}_{\text{old}}$ is diagnosed from the cell in which given SD is in the *current* time step. The *per-cell* algorithm is ~~less accurate, but~~ computationally less demanding, because $\boldsymbol{\psi}_{\text{old}}$ is the same for all SDs ~~that are in the same~~ in a given spatial cell. ~~This is not true in~~ In the *per-particle* algorithm ~~, in which~~ $\boldsymbol{\psi}_{\text{old}}$ can be different for different SDs in the same cell, so each SD needs to remember its own value of $\boldsymbol{\psi}_{\text{old}}$. Moreover, in the *per-particle* algorithm, values of pressure and density also need to vary between ~~sub-steps~~substeps. This is not necessary in the *per-cell* algorithm, because pressure and density in a given cell are constant in time. ~~Both sub-stepping algorithms are tested in section 4.2 to determine if~~ A more detailed description of the substepping algorithms and a comparison of results they produce is given in appendix B. The conclusion from that comparison is that the *per-cell* algorithm is ~~sufficient, or is~~ correct for stationary clouds, but gives significant errors if cloud edge moves, while the *per-particle* ~~sub-stepping necessary. A more detailed description of the sub-stepping algorithms is given in appendix B~~algorithm is correct in both cases. All presented results of modeling stratocumulus clouds were obtained using the *per-cell* algorithm.

The stochastic collision-coalescence process described in section 2.2 is modeled with a Monte Carlo algorithm developed by Shima et al. (2009). The key feature of this algorithm is that each SD can collide only with one other SD during a time step. Thanks to that, ~~the~~ computational cost of the algorithm scales linearly, and not quadratically, with the number of SDs. Dziekan and Pawlowska (2017) showed that this ”linear sampling” technique does not affect the mean, nor the standard deviation, of the results. Note that in the coalescence algorithm of Shima et al. (2009), ~~a~~ the same pair of SDs can collide multiple times during one time step. This feature was not implemented in libcloudph++ at the time when the paper Arabas et al. (2015) was published. libcloudph++ has been modified since then and now multiple collisions are allowed. It is possible to run the coalescence algorithm with a shorter time step than the model time step. Then, coalescence is calculated more than once in each model time step, a procedure we call coalescence substepping.

The procedure for initialization of SD sizes is described in detail in Dziekan and Pawlowska (2017), where it is called the "constant SD" initialization. In short, the range of initial values of r_d is divided into N_{SD} bins, which have the same size in $\log(r_d)$. In each bin, a single value or dry radius is randomly selected and assigned to a single SD. Multiplicity of the SD is readily calculated from the initial aerosol size spectrum. Next, wet radius is initialized to be in equilibrium with the initial relative humidity. If the initial relative humidity is higher than 0.95, the wet radii are initialized as if it was equal to 0.95.

This procedure is performed for each spatial cell. This initialization algorithm gives a good representation of the initial size spectrum even for small values of N_{SD}.

Advection of SDs is modeled with a predictor-corrector algorithm described in Grabowski et al. (2018a). Simpler, first-order algorithms for advection were found to cause inhomogeneous spatial distributions of SDs, with less SDs in regions ~~with~~ of high vorticity.

3.3 Order of operations

The sequence of operations done in a single time step is presented on a Unified Modeling Language (UML) sequence diagram in fig. 1. The diagram is a convenient way of showing how coupling between ~~the~~ Eulerian dynamics and ~~the~~ Lagrangian microphysics is done. The diagram also shows operations that are done simultaneously on CPUs and GPUs. Please note how the liquid water mixing ratio q_l is treated. ~~In principle, liquid~~ Liquid water is resolved by the SDM and q_l could be diagnosed from the super-droplet size spectrum each time it is needed in the ~~RHS. This would however require additional synchronization between CPUs and GPUs . To avoid this computational performance problem~~buoyancy term in eq. (3) or radiative term in eq. (4). Buoyancy is integrated with a trapezoidal scheme, which requires q_l after advection to be known. In a straightforward implementation, in which q_l is diagnosed from SDs after advection of SDs, pressure solver calculations can only be started after advection of SDs has been calculated. Then, there is little parallelism of calculations on GPUs and CPUs. To achieve more parallelism, we introduce an auxiliary Eulerian field for q_l. ~~Its value~~ Value of q_l is diagnosed from SDs ~~after condensation and its~~ once per time step, after condensation calculation. Then, q_l advection is done using ~~the~~ a first-order accurate upwind scheme. Using the auxiliary q_l field, it is possible to calculate coalescence and motion of SDs simultaneously with calculations of advection of Eulerian fields and of the pressure problem.

3.4 Spatial ~~discretization~~discretisation

~~All~~ Eulerian dependent variables of the model ~~dependent variables~~ are co-located. Their positions form the nodes of the primary grid. However, the libmpdata++ advection algorithms are formulated using a dual, staggered Arakawa-C grid (Arakawa and Lamb, 1977). The cell centers of the dual grid are the nodes of the primary mesh. Schematic of a 2D computational domain with the Arakawa-C grid is shown in fig. 2. Throughout this paper, by ”grid cells”, ”Eulerian cells” or simply ”cells”, we refer to the cells of the dual grid. To form the Arakawa-C arrangement, components of the vector $\boldsymbol{u}$ are linearly interpolated to the edges of the dual grid (see Jaruga et al. (2015) for details). Super-droplets are restricted to the physical space, which is the shaded region in fig. 2. Coupling of ~~the~~ Eulerian variables with SDs is done using the dual grid. All SDs that are located in the same cell of the dual grid are subjected to the same conditions ~~, which~~ that are equal to the values of scalars residing at the center of the cell. Similarly, condensation of a given SD affects scalars in the center of the dual grid cell, in which this SD is located. To calculate ~~the~~ velocity of air that advects a given SD, velocities, which reside at the edges of the dual grid, are interpolated to the position of the SD. The interpolation is done linearly, separately in each dimension, as advocated by Grabowski et al. (2018a). Spatial ~~discretization~~ discretisation is also necessary in the algorithm for modeling collision-coalescence (cf. section 2.2). We use the dual grid cells also as coalescence cells, with the exception of the cells at the domain edges. There, only the physical (shaded) part of dual grid cells is used as coalescence cells.

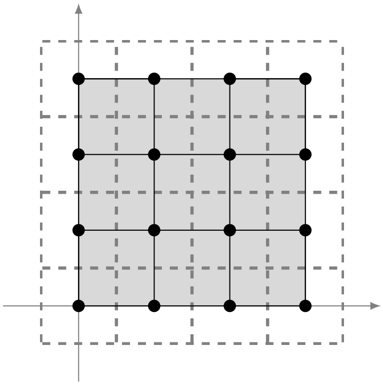

Figure 2. Schematic of a 2D computational domain. Bullets mark the data points for the dependent variable ψ in eq. (18), solid lines depict edges of primary grid and dashed lines mark edges of dual grid. Reproduced from Jaruga et al. (2015).

4 Comparison with other models - marine stratocumulus simulations

UWLCM ~~is validated by running~~ simulations of a marine stratocumulus ~~using a setup~~ cloud are presented in this section. The main goal is to validate UWLCM by comparing it with other LES models. We study sensitivity of the model to the way the SGS turbulence is modeled and to values of the microphysics scheme parameters. Results of this sensitivity study may provide some guidance to other users of Lagrangian microphysical schemes. The simulation setup is based on observations made during the second Dynamics and Chemistry of Marine Stratocumulus (DYCOMS-II) field study (Stevens et al., 2003). The setup, described in detail in Ackerman et al. (2009), is an idealization of conditions observed during the ~~research flight 2~~ second research flight (RF02) of this campaign. Both heavily drizzling open cells and lightly drizzling closed cells were sampled by RF02. The initial thermodynamic conditions are an average from both types of cells and the microphysical conditions are an average over heavily drizzling cells only. Comparison of simulation results from 11 different LES models is presented in Ackerman et al. (2009). There is a large variability in the amount of drizzle ~~in different models. It~~ predicted by different models, which illustrates how difficult it is for LES models to reproduce ~~drizzle~~ precipitation formation. One of the reasons why we chose to test UWLCM using this setup is to test how well ~~the~~ Lagrangian microphysics performs in modeling drizzle. The models that took part in the intercomparison use either bin microphysics (one model with single-moment bin and one model with double-moment bin) or bulk microphysics (2 models with single-moment bulk and 7 models with double-moment bulk). ~~Another difference between UWLCM and models discussed in Ackerman et al. (2009) is that the models from the intercomparison use explicit sub-grid scale (SGS) models, while UWLCM uses the implicit LES (iLES) approach (Grinstein et al., 2007). In the implicit LES approach, it is assumed that the numerical diffusion of the advection scheme is enough to reproduce diffusion due to the SGS mixing. The MPDATA algorithm is argued to be well-suited for iLES~~

~~simulations (Margolin and Rider, 2002; Margolin et al., 2006). It is important to note that there is no subgrid-scale mixing of liquid water in UWLCM, because the liquid water is represented by Lagrangian particles.~~

4.1 Simulation setup

The simulation setup follows Ackerman et al. (2009). The domain size is 6.4km x 6.4km x 1.5km with a regular grid of cells of 50m x 50m x 5m size. Rigid and periodic boundary conditions are used at the vertical and horizontal edges of the domain, respectively. ~~Simulations are run with two values of the time step length, $\Delta t = 1$s and $\Delta t = 0.1$s. In simulations with $\Delta t = 1$s, 10 sub-steps for condensation are done per single time step.~~ The simulations are run for 6 hours. The initial profiles of q_v and θ give high values of supersaturation in the layer ~~,~~ in which a cloud was observed. However, the simulation is initialized without any cloud water, because it is not known analytically what should be the initial wet radius distribution. First part of the simulation, called the *spinup* period, is dedicated to obtaining a stationary distribution of wet radii. During the spinup, the collision-coalescence process is turned off and the supersaturation in the condensational growth equation is limited to 1%. Please note that in ~~(Ackerman et al., 2009)~~ Ackerman et al. (2009) this supersaturation limit is applied only to the activation and not to the condensational growth. This approach can not be used in UWLCM, because in UWLCM activation is not modeled as a separate process. The spinup period is 1 hour long, which was found to be long enough to reach a stationary concentration of cloud droplets ~~, which should be a good proxy for having~~ – indicating a stationary spectrum of the wet radius. Aerosol is assumed to consist of ammonium sulfate with the initial size distribution as defined in Appendix A of Ackerman et al. (2009). Following Petters and Kreidenweis (2007), the hygroscopicity parameter for ammonium sulfate is $\kappa = 0.61$. ~~Surface fluxes are exponentially distributed in each column with a 25 m e-folding height.~~ Collision efficiencies are taken from Hall (1980) for large droplets and from Davis (1972) for small droplets. Coalescence efficiency is set to 1. ~~The terminal velocity is calculated using the~~ Terminal velocities are calculated using a formula from Khvorostyanov and Curry (2002). libmpdata++ allows the user to choose from a number of MPDATA options. In the presented simulations, we use the "infinite-gauge" option *iga* for handling variable-signed fields ~~and~~ combined with the non-oscillatory option *fct*.

4.2 Two-dimensional simulations: sensitivity study of SDM

~~The 2D~~ Two-dimensional simulations are used to investigate ~~differences between the per-celland per-particlesubstepping algorithms (see Sec. B), and to test sensitivity of results to the number of SDs. Simulations are done for $N_{\mathrm{SD}} = 40$ (which is the~~ sensitivity of the model to parameters of the SDM: the coalescence time step length Δt_{coal} and the initial number of SDs ~~per cell also used in 3D simulations) and for $N_{\mathrm{SD}} = 1000$~~ N_{SD}. Results are compared with 3D simulations from the Ackerman et al. (2009) in order to assert if 2D simulations, which are computationally cheap, give reasonable representation of some of the features of 3D simulations. However, it has to be kept in mind that the turbulence behavior in 2D is fundamentally different from 3D. Simulations are run for two model time step lengths, $\Delta t = 0.1$s and $\Delta t = 1$s. No substepping is done for $\Delta t = 0.1$s. Results of this simulations provide a reference for simulations with longer time steps. In simulations with $\Delta t = 1$s, 10 substeps for condensation are done, hence condensation time step is $\Delta t_{\mathrm{cond}} = 0.1$s. Using a longer condensation time step results in activation of too many aerosols (result not shown in the following figures for clarity). Two values of the coalescence time step are tested: $\Delta t_{\text{coal}} = 1$s (no coalescence substepping) and $\Delta t_{\text{coal}} = 0.1$s (10 coalescence substeps) in combination with two values of the initial number of SDs: $N_{\text{SD}} = 40$ (which is used in 3D simulations) and $N_{\text{SD}} = 1000$. Since the goal of 2D simulations is to study the microphysical model, SGS turbulence is modeled using the ILES approach. In 2D, we ~~observed~~ observe significant variability in results of simulation runs done for the same ~~values of N_{SD}, Δt and using the same substepping algorithm~~parameter values. The variability comes from two sources. One is that the initial thermodynamic conditions include a small random perturbation. The other is that initialization of SD radii and collision-coalescence of SDs are modeled with Monte Carlo algorithms. To compensate for this inherent variability, all shown UWLCM results of 2D simulations are averages from ensembles of 10 simulations.

Time series of selected ~~domain-averaged~~ domain averaged variables are shown in fig. 3. The ~~3D simulations from Ackerman et al. (2009) are referred to as reference simulations~~only significant relationship between model parameters and results is that the amount of surface precipitation is ca. two times higher for $\Delta t_{\text{coal}} = 1$ s than for $\Delta t_{\text{coal}} = 0.1$ s. Stronger precipitation induces differences in LWP that, up to the onset of precipitation, is the same for all parameter combinations. The amount of ~~liquid water~~ surface precipitation does not depend on ~~Δt nor on N_{SD}. It slowly decreases with time, but is within the range of results of the reference simulations. The entrainment rate, which is calculated as the rate of increase of the inversion height, also agrees with the reference simulations , except for differences during the spinup period. However, there is a noticeable~~ the model time step nor on the number of SDs. Interestingly, 2D simulations show an abrupt increase in the entrainment rate ~~during the 3h - 5h period of the simulations with $\Delta t = 1$s. This increase coincides with a more pronounced increase~~ and in the maximum of variance of ~~vertical velocity, which up to this point is in agreement with the reference~~ w around 3h of the simulations. This ~~in turn~~ increase is preceded by ~~a period of increased surface precipitation from 2h to 4h of the simulation. That would suggest that the increase in the maximum of variance of w is caused by rain evaporation. This hypothesis is backed by the fact that simulations with most rain (for $N_{\text{SD}} = 1000$) have the highest maximum of variance of w. However, for $\Delta t = 0.1$s, a sudden~~ the moment when first precipitation reaches the surface. This suggests that the increase in the maximum of variance of w ~~is also seen, albeit at a later period starting at 5 h, but is not preceded by an increase in the surface precipitation. It is probable that for $\Delta t = 0.1$s rain evaporationalso drives the increase of VAR(w), but all rain drops evaporate before reaching the ground. Outside of the periods of increased precipitation that were discussed above, there is almost no surface precipitation in UWLCM simulations. There is a large spread in the amount of surface precipitation in the reference simulations, with some models producing as little rain as the 2D UWLCM~~and in the entrainment rate is caused by rain evaporation. The need for the spinup period for microphysics is best seen on the N_c time series. Initially, due to the large initial supersaturation, cloud droplets form on all aerosol particles. Afterwards, N_c quickly decreases and after 1h ~~arrives at~~ reaches the value of ca. 60 cm^{-3}~~. Then, the number of cloud droplets remains constant, aside of the periods with increased variance of w, when the activation rate increases. Concentration of cloud droplets after the spinup is~~ , in agreement with the 3D reference simulations. ~~It is worth to note that eight of the reference models do not predict the value of cloud droplet concentration, but use a fixed value $N_c = 55$. The cloud base height in 2D simulations remains constant until the third hour of simulation. Then it starts to increase and catches up with the reference simulations.~~

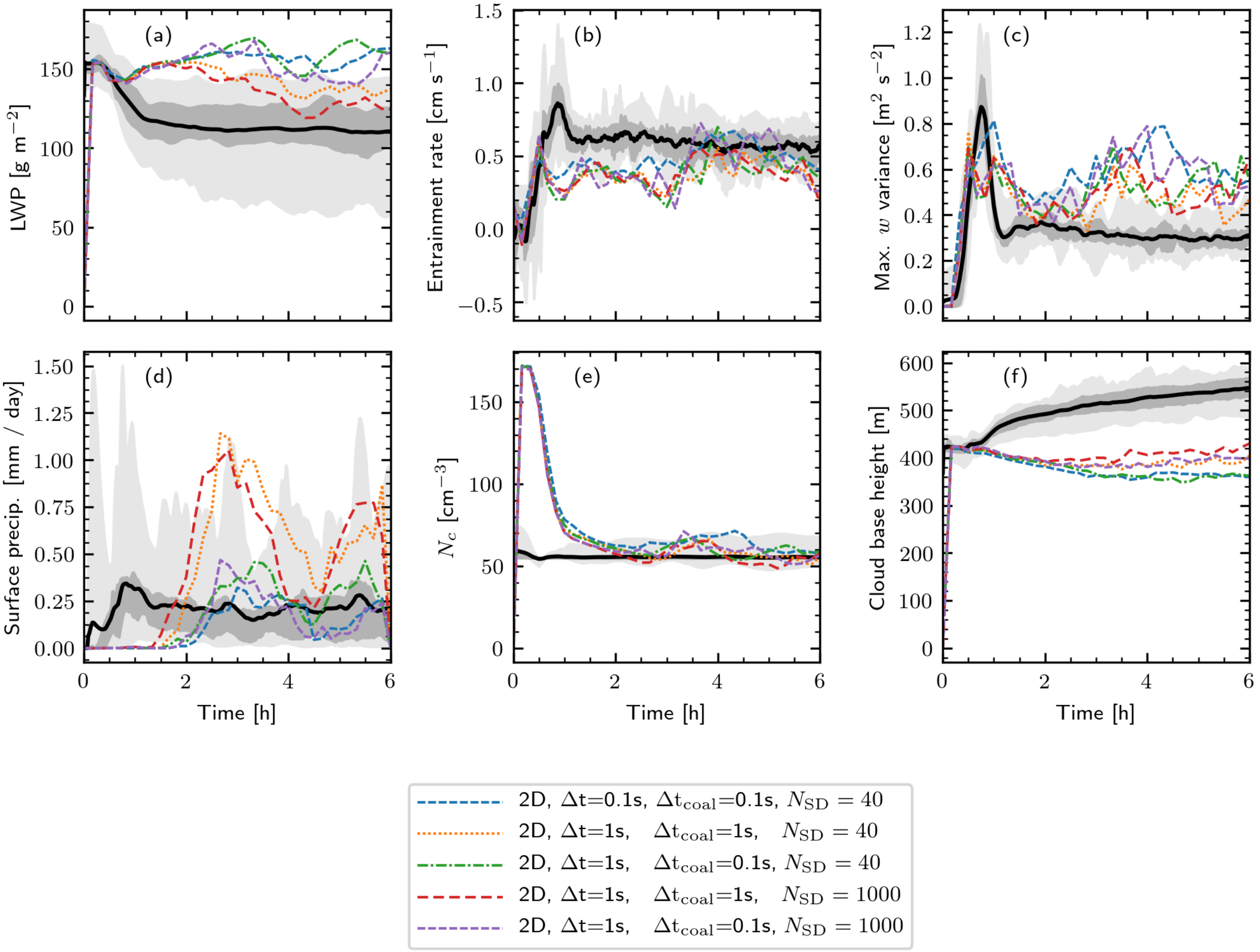

Figure 3. 2D UWLCM results. Time series of ~~a~~ the domain averaged liquid water path, entrainment rate (equal to $dz_i/dt + w_{\mathrm{LS}}z_i$), ~~maximum of~~ vertical velocity variance VAR(w) maximum, surface precipitation, concentration of cloud droplets in cloudy cells and cloud base height. UWLCM simulations were done for different ~~values of time step length, number~~ model and ~~type of sub-steps,~~ coalescence time steps and different initial ~~number~~ numbers of SDs per cell. Each colored line represents an average from 10 UWLCM simulations of a given type. Results of 3D simulations from an ensemble of 11 models are shown for reference (Ackerman et al., 2009). Mean, middle two quartiles, and range of that reference ensemble are plotted with the black solid line, the dark shaded region and the light shaded region, respectively.

Vertical profiles from the 2D simulations are shown in fig. 4. ~~Vertical distributions of liquid water, total water and liquid-water potential temperature do not depend on the time step length nor on the number of SDs, and are in agreement with reference simulations~~As already observed in time series plots, precipitation flux strongly depends on Δt_{coal}. Precipitation flux ~~for $\Delta t = 1$s is similar to the smallest values of the reference results and is slightly higher~~ profile reveals that precipitation flux also weakly depends on N_{SD}; it is slightly lower for $N_{\mathrm{SD}} = 1000$ than for $N_{\mathrm{SD}} = 40$. A similar observation was made in Dziekan and Pawlowska (2017), where the ~~mean autoconversion time was found~~ autoconversion efficiency was shown to increase with ~~decreasing~~ N_{SD}. The most striking differences between the 2D UWLCM and 3D reference simulations are seen on the profiles of moments of the vertical velocity distribution. This is associated with the decreased dimensionality of our simulations. Interestingly, profiles of VAR(w) and of the third moment of w are in better agreement with observations (see fig. 3 in Ackerman et al. (2009)) in the 2D UWLCM than in the 3D reference simulations. ~~It is worth to note that the profile of VAR(w) is not constant during the 2h - 6h period over which results in fig. 4 are averaged. Instead, it goes through significant changes with the maximum of VAR(w) varying between 0.4 and 1.2 (see fig. 3) .~~

~~Differences between the *per-particle* and *per-cell* sub-stepping algorithms are visible on the profiles of N_c. Firstly, fewer cloud droplets are produced in the *per-particle* than in the *per-cell* case. Secondly, N_c within the cloud layer decreases with height in the *per-particle* case, but increases with height in the *per-cell* case. In simulations without sub-stepping and with $\Delta t = 0.1$s, a decrease in N_c with height is seen, which is in agreement with the *per-particle* sub-stepping. However,the *per-cell* sub-stepping gives values of N_c closer to the value obtained for $\Delta t = 0.1$s, but we consider it a coincidence, since supersaturation is higher in the $\Delta t = 0.1$s simulations.~~

~~Judging from results of 2D simulations ,~~ A conclusion for SDM modeling is that coalescence needs to be resolved with a time step of the ~~*per-particle* substepping is necessary to properly represent condensational growth~~ order of 0.1 s, although more rigorous convergence tests should be carried out in the future. This conclusion is surprising, because coalescence tests of SDM in box models give correct results for time steps larger than 1s (result not shown) and Shima et al. (2009) estimated that the coalescence algorithm should work well for Δt_{coal} of the order of 1 s. Moreover, one might expect that using large Δt_{coal} should give too little precipitation, as large Δt_{coal} can make the mean number of collisions to be lower than expected. This is because SDM handles large Δt_{coal} by allowing multiple collisions between SDs and sometimes, when one of the SDs has low multiplicity, not all of these multiple collisions can be realized. However, we see that surface precipitation increases with Δt_{coal}. A possible explanation is that for large Δt_{coal} some of the SDs become extremely lucky and grow much faster than expected due to multiple collisions. Then, even if the mean number of collisions is lower than it should be, some SDs become very large and cause the observed high surface precipitation. The second conclusion for SDM modeling that can be drawn from the sensitivity test is that N_{SD} of the order of 40 is sufficient to obtain correct domain averaged results. Certainly, this does not mean that this relatively low number of SDs is sufficient in all cases. For example, larger number of SDs would probably be needed in simulations in which SDs have more attributes, e.g. when modeling aqueous chemistry. Also, we expect that observables other than domain averages, e.g. related to the spatial structure of a cloud, are more sensitive to the number of SDs. ~~Regarding~~ Schwenkel et al. (2018) present in more detail how cloud structure depends on the number of SDs. In general, 2D ~~simulations showed that using relatively few SDs ($N_{\mathrm{SD}} = 40$) does not affect the results, aside froma small decrease in the~~

~~amount of precipitation~~UWLCM results do not deviate very much from the 3D simulations from Ackerman et al. (2009). The biggest difference is in the cloud base height – cloud layer is significantly deeper in 2D. This shows that cheap 2D simulations can be used to coarsely study microphysical effects in stratocumulus clouds.

4.3 Three-dimensional simulations: model validation and SGS effects

~~The~~ Based on conclusions of the 2D sensitivity test, 3D simulations ~~were~~ are done for $N_{\mathrm{SD}} = 40$~~and for two values of time step length, $\Delta t = 0.1$s without substepping and $\Delta t = 1$swith 10 substeps using the *per-particle* algorithm~~, $\Delta t = 1$ s, $\Delta t_{\mathrm{cond}} = 0.1$ s and $\Delta t_{\mathrm{coal}} = 0.1$ s. Three different models of SGS turbulence are tested: implicit LES, the Smagorinsky scheme and the Smagorinsky scheme with turbulent SGS motion of SDs. Contrary to the 2D simulations, the 3D simulations show very little variability between realizations~~thanks to a~~ , thanks to the larger simulation domain. Therefore averaging over an ensemble of simulations, which was necessary in the 2D case, is not needed here and results shown come from ~~a single simulation run~~single simulation runs. Time series of the results are shown in fig. 5. ~~It is seen that the time step length has greater impact on 3D simulations than on 2D simulations. Nevertheless, results for both values of Δt are within the range of~~ The biggest difference between different descriptions of the SGS turbulence is in the liquid water content. In ILES, in which there is no diffusion of liquid water, LWP increases over time and is much higher than in the reference simulations~~, with the exception of surface precipitation. Liquid-water path slowly decreases with time and is ca. 30% higher for $\Delta t = 1$s than for $\Delta t = 0.1$s. The entrainment rate is close to the highest values of the reference results, with slightly higher values for $\Delta t = 0.1$s than for $\Delta t = 1$s, which may be the cause why LWP is smaller in the former case. The maximum of VAR(w) is the same for both runs and is in very good~~ . Using the Smagorinsky scheme alone, i.e. without diffusion of liquid water, gives less liquid water than ILES. This indicates that diffusion of Eulerian variables in simulations with the Smagorinsky scheme is higher than in ILES. Still, LWP in that case is close the maximum from the reference models. Using the Smagorinsky scheme with turbulent motion of SDs, which models SGS diffusion of liquid water, further decreases LWP and results in better agreement with the ~~mean from the reference simulations. Like in 2D simulations, there is very little surface precipitation in the 3D runs. Concentration of cloud drops after the spinup period is in~~ reference models. Using the Smagorinsky scheme gives the best agreement with the reference ~~simulations, although simulations with $\Delta t = 0.1$s give a little higher values of N_c than simulations with $\Delta t = 1$s. Contrary to the 2D results,~~ models also in other variables: entrainment rate, maximum of VAR(w) and cloud base height~~is in agreement with the reference simulations.~~ . Diffusion of liquid water has visible impact on LWP and therefore needs to be included in Lagrangian microphysics models. Unfortunately, this can not be done in ILES with SDM, because in that case a measure of the SGS energy dissipation is not readily available. Aside from decreasing LWP, SGS diffusion of liquid water is seen to decrease N_c, giving better agreement with the reference models right after the spinup period. Afterwards, SGS diffusion of liquid water causes N_c to slowly decrease with time. A possible explanation is that in regions with little vertical motion, cloud droplets that diffuse out of supersaturated cells will evaporate, but aerosol particles that diffuse into supersaturated cells will not necessarily be activated, because condensational growth of larger cloud droplets already present in this region may consume all available supersaturation. Surface precipitation is very low in all 3D UWLCM simulations. 2D UWLCM simulations give larger surface precipitation than 3D UWLCM, which is attributed to a deeper cloud layer in 2D. There is a

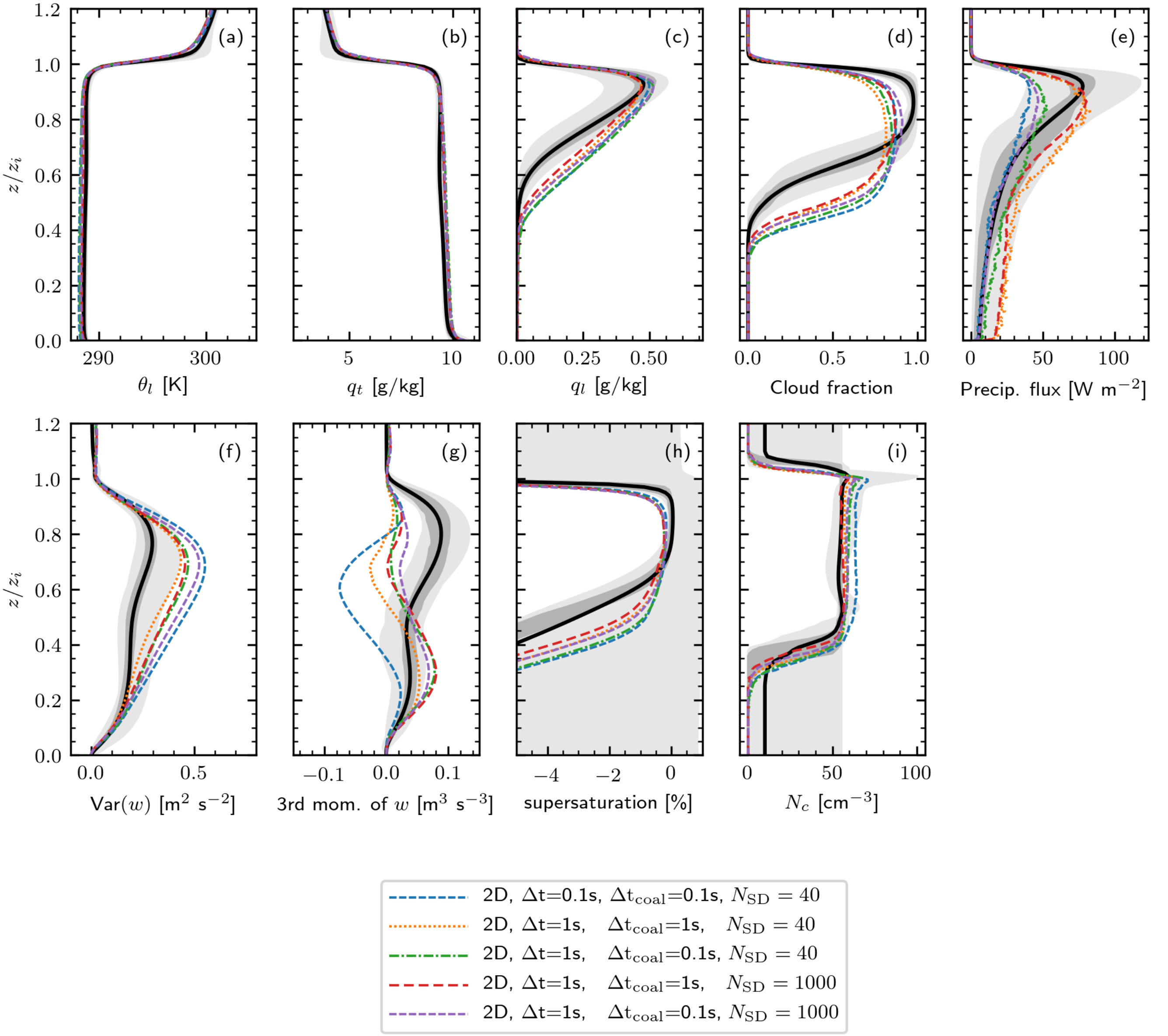

Figure 4. 2D UWLCM results. As in fig. 3, but showing horizontally-averaged profiles of liquid water potential temperature (defined in Ackerman et al. (2009)), total water mixing ratio, liquid water mixing ratio, cloud fraction (defined in appendix A), precipitation flux (defined in appendix A), variance of vertical velocity, third moment of vertical velocity, supersaturation and concentration of droplets in cloudy cells~~,~~ ~~which is shown on the two panels with different scales of the horizontal axis~~. ~~On the vertical~~ Vertical axis is ~~height~~ altitude normalized by ~~the~~ inversion height. The profiles are averaged over the 2h to 6h period.

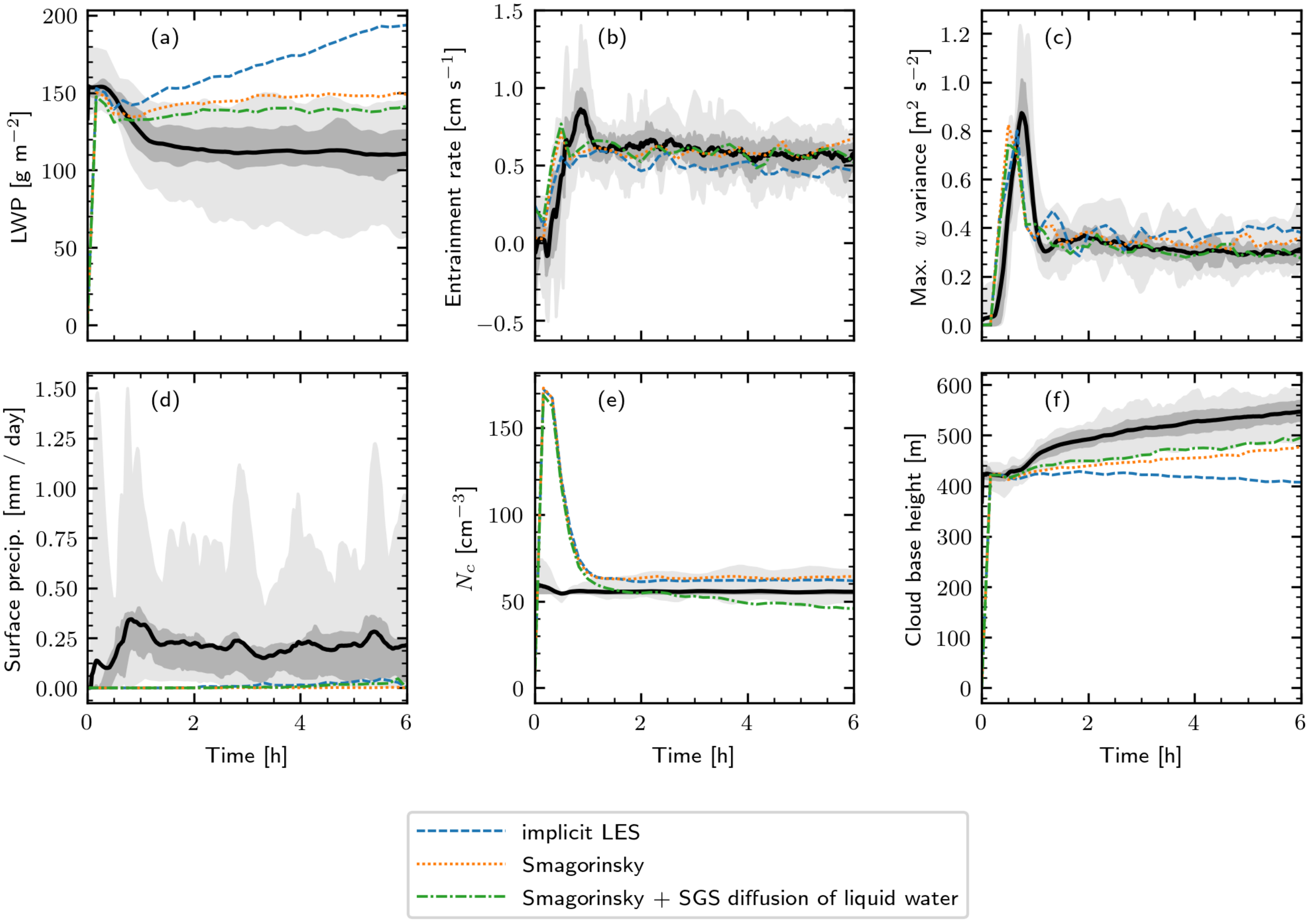

Figure 5. As in fig. 3, but for 3D UWLCM simulations. No averaging over ensembles is done, i.e. each line comes from a single UWLCM run.

very large spread in surface precipitation in the reference models, with some of them producing as little as 3D UWLCM. The subject of discrepancy in surface precipitation is discussed in more detail in Sec. 4.4.

~~Profiles obtained from 3D simulations are presented in fig. 6. Regardless of the time step length, liquid-water potential temperature, total water mixing ratio and liquid-water mixing ratio are in agreement with reference simulations. Cloud fraction, which is the fraction of cells with $N_c > 20$ cm^{-3}, is slightly lower than in reference simulations, especially for $\Delta t = 0.1$s. Judging from the profile of N_c, average value of N_c in UWLCM is slightly higher than in the reference simulations. Therefore spatial variability in N_c has to be higher in UWLCM than in the reference models. The reason for this may be twofold. Firstly, there is no numerical diffusion of N_c in UWLCM, thanks to the use of Lagrangian microphysics. Secondly, the SDM used in UWLCM produces too high spatial variability, because it represents liquid water using a relatively small number of computational droplets . The precipitation flux is lower than in reference simulations. Variance of w does not depend on~~

~~Δt and is in agreement with the reference simulations . The absolute value of the third moment of w is close to zero, in contrast to the reference results. The difference is potentially caused by the iLES approach used in UWLCM~~Profiles from 3D simulations are presented in fig. 6. Using the Smagorinsky scheme with SGS diffusion of liquid water gives the best agreement with the reference models, with the exception cloud fraction that is smaller than reference. ~~Arguably, the third moment of w within cloud layer is in better agreement with observations than the reference simulations (observational results are plotted in Ackerman et al. (2009)). In the sub-cloud layer the contrary is true: the reference simulations are in better agreement with observations. The supersaturation profile for $\Delta t = 1$s is in agreement with referenceresults. For $\Delta t = 0.1$s supersaturation is a little lower than for $\Delta t = 1$s, which is the opposite of what was observed in 2D simulations. The~~ However, cloud fraction profile strongly depends on definition of cloudy cells. Following Ackerman et al. (2009), we define cloudy cells as those with concentration of cloud droplets ~~is slightly higher than the mean~~ greater than 20 cm^{-3}. On the other hand, most of the reference ~~results. As in the 2D case, N_c is a little higher for the shorter time step. Droplet activation rate near the cloud top is higher for $\Delta t = 0.1$s than for $\Delta t = 1s$, possibly due to a better resolved maxima of supersaturation at the cloud top or a better resolved time of residence of aerosols near the cloud top. Finally, it is seen that droplet activation near the cloud base happens at higher altitudes than in most of the reference simulations.~~ models use parametrised microphysics. Therefore these models define cloudy cells as saturated cells. Using this definition, all UWLCM runs give maximum cloud fraction of ca. 95 %, in agreement with the reference models. Also, cloud cover, defined as fraction of columns with LWP > 20 g m^{-2}, is close to 100 % in all 3D UWLCM simulations. Choice of the SGS diffusion model also affects structure of the velocity field. Increasing SGS diffusion strength decreases variance of w and increases skewness of w, which shifts from negative for ILES to positive for the Smagorinsky scheme with diffusion of liquid water.

4.4 Precipitation results

The purpose of this section is to study ~~the~~ discrepancy between the amount of surface precipitation observed during the DYCOMS-II campaign (from ca. $0.25\ \mathrm{mm/day}$ to ca. $0.45\ \mathrm{mm/day}$, Ackerman et al. (2009)) and modeled by 3D UWLCM (almost none). ~~The precipitation~~ Precipitation flux in UWLCM is ca. two times lower than the average of reference simulations (cf. fig. 6). ~~We suspect that the reason is that the number of cloud droplets is a little higher in UWLCM. To check if this is the case, a simulation with a decreased concentration of aerosols is presented. We also~~ To better understand this issue we make a comparison with the only models with bin microphysics that took part in the reference intercomparison: Distributed Hydrodynamic Aerosol and Radiative Modeling Application (DHARMA, Stevens et al. (2002)) and Regional Atmospheric Modeling System (RAMS, RAMS Technical Description). DHARMA uses single-moment bin microphysics ~~, while~~ and RAMS uses double-moment bin microphysics. We compare our results only with these two models, because, contrary to the bulk schemes, bin schemes explicitly resolve ~~the size spectrum~~ size spectrum of droplets and do not rely on parametrisations of the collision-coalescence process, i.e. they are at a similar level of precision as the SDM. Bin microphysics are troubled by artificial broadening of the size spectrum of droplets due to numerical diffusion associated with advection in the physical space (Morrison et al., 2018). Such artificial broadening increases the rate of collision-coalescence, hence models with bin microphysics might

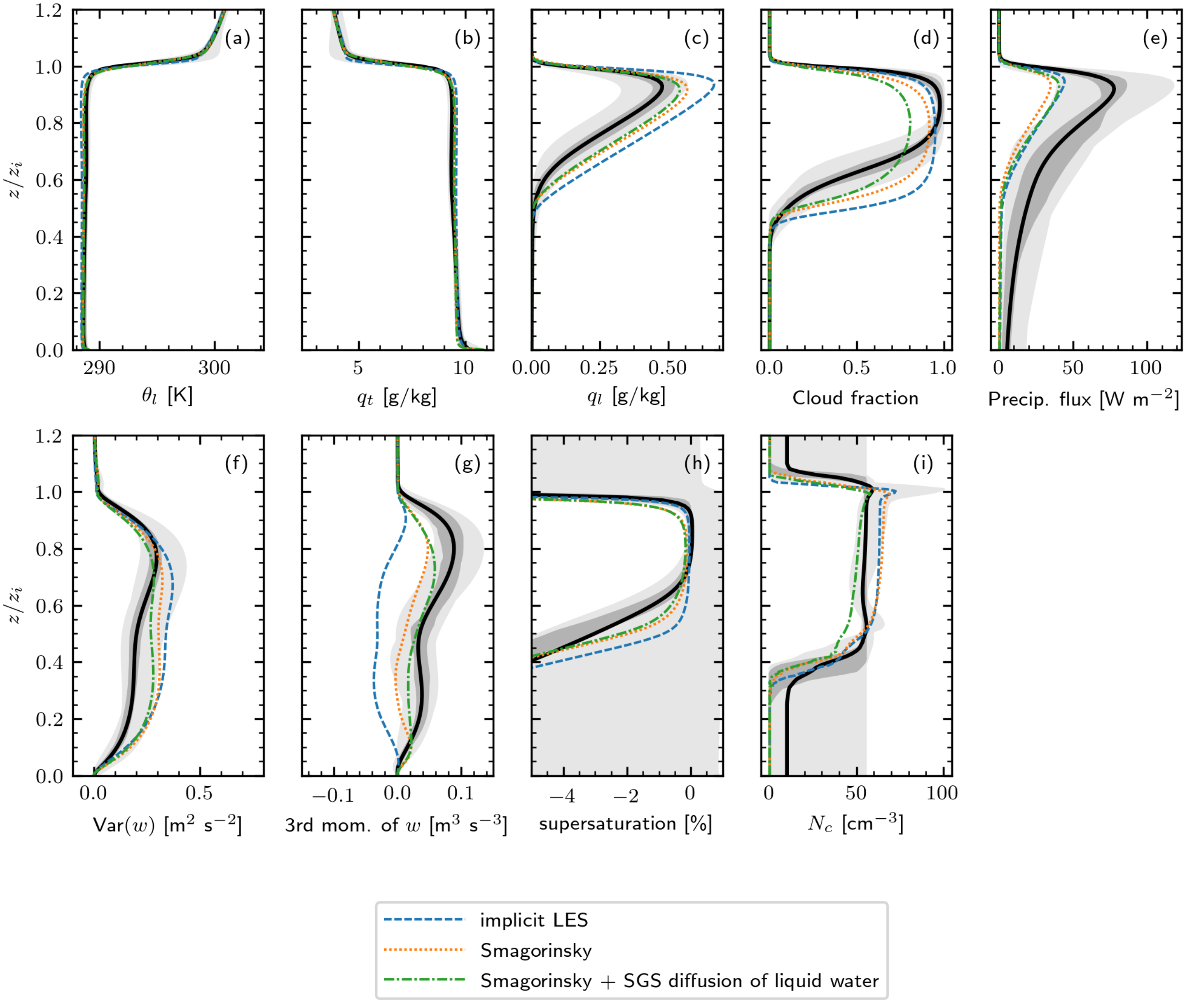

Figure 6. As in fig. 4, but for 3D UWLCM simulations. No averaging over ensembles is done, i.e. each line comes from a single UWLCM run.

produce too much ~~drizzle~~precipitation. Lagrangian, particle-based schemes such as SDM have no numerical diffusion in the size spectrum.

Time series and profiles showing the amount of liquid water, surface precipitation and concentration of cloud droplets from UWLCM, DHARMA and RAMS are plotted in fig. 7. ~~In addition to the UWLCM run already shown in figs. 5 and 6, which we will refer to as the default UWLCM , we present a UWLCM simulation in which the initial concentration of aerosols in the larger mode is 45 (default value is 65). Surface precipitationin the RAMS model is close to zero~~Precipitation flux and surface precipitation are similar in UWLCM and RAMS. Both models produce almost no surface precipitation, aside from a short period at the start of the RAMS simulation, when the simulation has not yet reached a stationary state. ~~This result is in agreement with the default UWLCM simulation.~~ The DHARMA model stands out in that the amount of surface precipitation it produces is higher, in agreement with observations. ~~It might seem that the reason why it produces more precipitation is that the concentration of cloud droplets is lower than in the default UWLCMor RAMS. However, the UWLCM simulation with lower aerosol concentration produces even lower concetration of cloud droplets than DHARMA, but does not give as much surface precipitation . The total amount of liquid water is in good agreement between DHARMA and UWLCM. Together, this suggests that the large amount of surface precipitation in DHARMA may be caused by the~~ Cloud depth, LWP and N_c are similar in DHARMA and UWLCM. So why does DHARMA give a much higher precipitation flux? One possible explanation is that it is a result of artificial broadening of size spectra caused by numerical diffusion. On the other hand, why does UWLCM give less precipitation than observed? Possibly precipitation is affected by some physical processes that are not modeled by UWLCM, e.g. SGS turbulence affecting condensation and coalescence of droplets, lucky droplets effect or giant CCN.

4.5 ~~Explicit modeling of activation~~

In ~~UWLCM,~~ bin microphysics of RAMS and DHARMA, water droplets are artificially divided into haze particles and cloud droplets, and droplet activation is modeled ~~explicitly, in contrast to the models with bin microphysics, which parametrise this process. Difference between these two approaches is~~ as an instantaneous process (Stevens et al., 1996; Ackerman et al., 1995) . Therefore even a short-lived maximum of supersaturation results in activation of new droplets. In contrast, in Lagrangian microphysics of UWLCM all water droplets grow according to the same equations, which include curvature and solute terms, and droplet activation is not modeled as a separate process. Thanks to that activation is not instantaneous, but happens over some time. Differences between treatment of activation in bin and Lagrangian microphysics are apparent in the profile of N_c in fig. 7. DHARMA and RAMS ~~models~~ predict local maxima of N_c near the cloud base, where supersaturation is ~~the heighest. This is because parametrisations of activation assume that it is an instantaneous process, therefore even a short-lived maximum of supersaturation results in activation of new droplets~~highest. In UWLCM, thanks to the explicit treatment of activation, the time scale of activation is resolved and the local maximum of supersaturation near cloud base does not cause activation of new droplets. Therefore N_c in UWLCM monotonously increases near the cloud base. ~~The activation height is in good agreement between UWLCM and RAMS. In DHARMA activation takes place at a much lower altitude.~~

4.5 ~~Aerosol processing~~

~~The SDM models spectrum not only of the wet radius, but also of the dry radius. This makes it ideal for studying processing of aerosols in clouds. One example is an effect observed in the time series of N_c in fig. 7. The UWLCM simulation with decreased initial aerosol concentration predicts a slow decrease of N_c with time. We interpret it as a manifestation of depletion of large aerosols, which are washed out with surface precipitation. Then, less large aerosols are present, hence less cloud droplets are formed. In the default UWLCM simulation, in which there is no surface precipitation, N_c remains constant in time.~~

5 Summary

~~We have presented~~

We presented University of Warsaw Lagrangian Cloud Model (UWLCM), a new large-eddy simulations model with ~~a~~ Lagrangian particle-based cloud microphysics. The model ~~was build~~ is built by combining two open-source libraries, one for handling ~~the~~ Eulerian dynamics and the other implementing the Lagrangian microphysics scheme. Methods for coupling ~~the~~ Lagrangian microphysics with ~~the~~ Eulerian dynamics were presented, including spatial ~~discretizations, a sub-stepping algorithm~~ discretisation, substepping algorithms and an algorithm for simultaneous computations of ~~the~~ Eulerian and Lagrangian components. Simulations of a marine stratocumulus ~~have shown~~ show that the model gives results in agreement with reference results from 11 other LES models~~. The model was found to work well with a 1s time step~~, which proves the capability of Lagrangian microphysics to model realistic clouds. Two-dimensional and three-dimensional simulations of the stratocumulus ~~setup were~~ have been performed. The two-dimensional simulations with UWLCM ~~were shown to~~ give reasonable results regarding microphysical phenomena at a fraction of the computational cost of the three-dimensional simulations~~. Number of computational particles used in~~ , and were used to study sensitivity of the Lagrangian microphysics ~~was not found to have impact on~~ scheme. It was found that the condensational and collisional growth of droplets has to be modeled with a 0.1 s time step and that number of computational particles does not affect domain averages, apart from ~~a small increase in the amount of surface precipitation~~small changes in precipitation flux. Different approaches to modeling SGS turbulence were compared in three-dimensional simulations. The best agreement with other models is obtained by using the Smagorinsky scheme and an algorithm for SGS turbulent motion of computational particles. The implicit LES approach is troubled by the lack of diffusion of liquid water represented by Lagrangian computational particles. Surface precipitation modeled by UWLCM is lower than ~~in most models with bulk microphysics. Compared to models with bin microphysics, surface precipitation in UWLCMis in agreement with the RAMS model, but smaller than in the DHARMA model.~~ observed. This suggests that some physical phenomena not modeled by UWLCM, such as SGS turbulence affecting condensation and coalescence of droplets or giant CCN, are important for precipitation formation. In UWLCM, all particles, including humidified aerosols, evolve according to the same set of equations. Therefore it is not necessary to include droplet activation as an additional process. Advantages of such approach are most apparent near the cloud base, where bin schemes ~~discussed~~ produce local maxima of cloud droplet concentration, while in UWLCM cloud droplet concentration increases monotonously. ~~A discrepancy in the third moment of the vertical velocity was found between UWLCM and the reference models. Probably, this discrepancy is due to the implicit LES approach~~

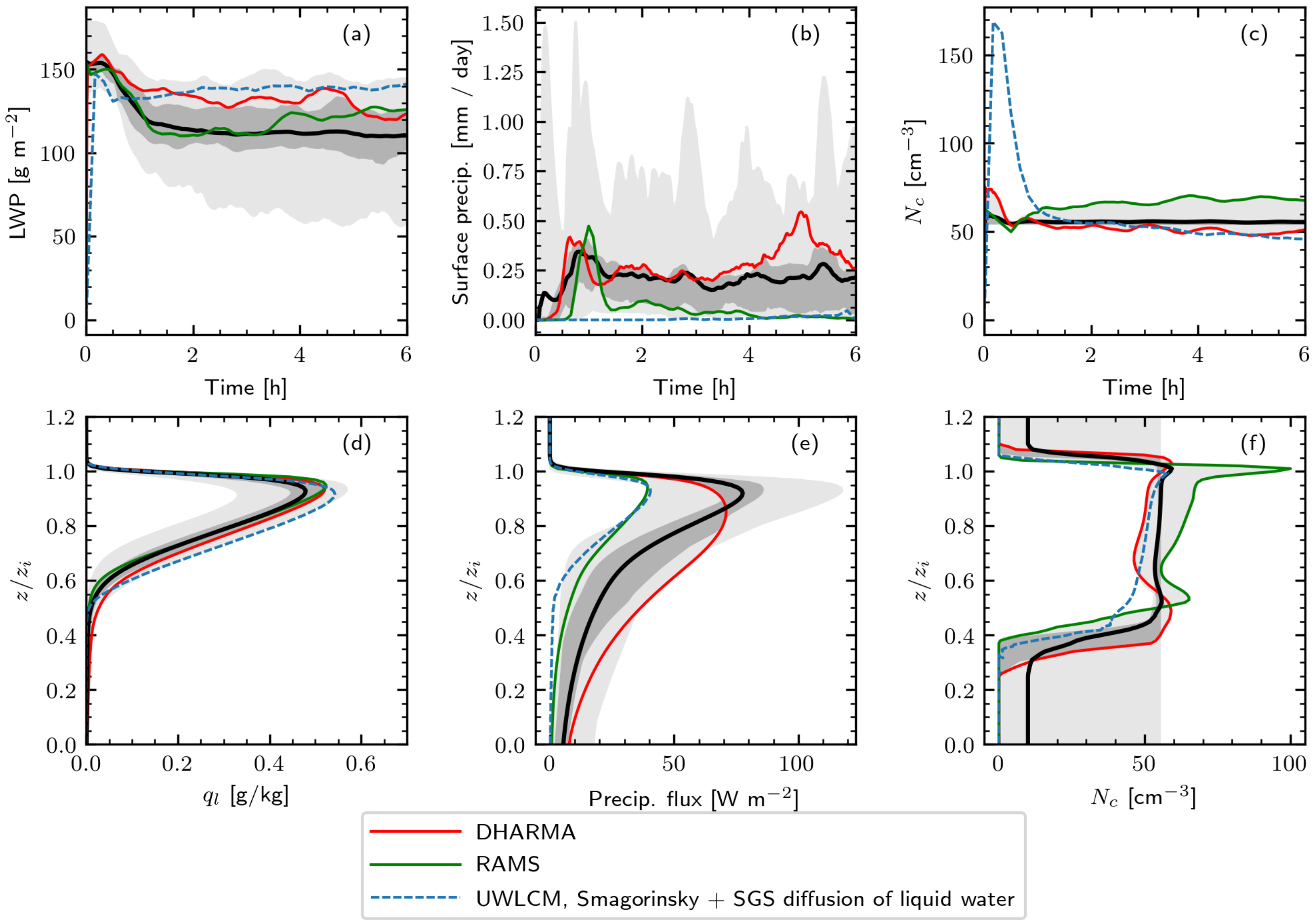

Figure 7. Selected time series of domain averages (upper row) and vertical profiles (lower row) from the 3D UWLCM and the two models with bin microphysics that took part in the Ackerman et al. (2009) intercomparison: DHARMA and RAMS. ~~UWLCM results come from a 3D simulations with $\Delta t = 1$s and 10 *per-particle* sub-steps. The ”lower concentration” run of UWLCM uses a decreased concentration of aerosols in the larger mode, 45 (default value is 65).~~ Out of the two DHARMA runs done for the intercomparison, the DHARMA_BO run is shown, because it uses coalescence efficiency closer to unity, ~~which~~ that is the value used in UWLCM and RAMS. Profiles are averaged and scaled as in fig. 4. Reference results from all models discussed in Ackerman et al. (2009) are depicted as in fig. 4.

~~used in UWLCM, while all the reference models use explicit SGS schemes. This hypothesis will be tested once explicit SGS schemes are added to UWLCM. Implementation of the Smagorinsky scheme is currently under development. Other extensions of UWLCM that we are working on include a distributed-memory implementation, addition of a single-moment and double-moment microphysics schemes and addition of a SGS scheme for condensation (Grabowski and Abade, 2017).~~

6 Code availability

UWLCM, libmpdata++ and libcloudph++ source codes are available at https://github.com/igfuw. In the study, the following code versions were used: UWLCM v1.0 (https://doi.org/10.5281/zenodo.2791156), libmpdata++ v1.2.0 (https://doi.org/10.5281/zenodo.2787740) and libcloudph++ v2.1.0 (https://doi.org/10.5281/zenodo.2790277).

7 Data availability

For simulation results, please contact P. Dziekan.

Appendix A: List of symbols

Table A1. List of symbols. As in Ackerman et al. (2009), cloudy cells are those with concentration of cloud droplets greater than 20 cm^{-3}. Cloud droplets are liquid particles with radius in the range 0.5 µm $< r <$ 25 µm. Cloud fraction is the ratio of cloudy cells to the total number of cells. Precipitation flux in a cell is calculated as $\left(\sum \xi \frac{4}{3}\pi r^3 w_t\right)\rho_w l_v / V$, where V is volume of the grid cell and the sum is done over all SDs in the cell.

Symbol	SI unit	Description
$\theta = T(p_{1000}/p)^{\frac{R_d}{c_{pd}}}$	[K]	potential temperature
$p_{1000} = 10^5$	[Pa]	reference pressure
~~θ_v~~ $\theta_v, \theta_l = (p_{1000}/p)^{\frac{R_d}{c_{pd}}}(T - l_{v0}\frac{q_l}{c_{pd}})$	[K]	virtual~~potential temperature $\theta_l = (p_{1000}/p)^{\frac{R}{c_p}}$~~
R_d, R_v	[J K^{-1} kg^{-1}]	gas constants for dry air/water vapor
$c_\mathrm{pd} = 1005$	[J K^{-1} kg^{-1}]	specific heat at const. pressure for dry air
$l_v(T)$	[J kg^{-1}]	latent heat of evaporation (cf. Arabas et al. (20
$l_{v0} = 2.5 \times 10^6$	[J kg^{-1}]	latent heat of evaporation at the triple point
$q_v = m_v/m_d$, q_{vs}	[kg kg^{-1}]	water vapor mixing ratio/saturation vapor mix
q_l, $q_t = q_v + q_l$	[kg kg^{-1}]	liquid-water/total water mixing ratio
m_v, m_d	[kg]	mass of water vapor / dry air
$\boldsymbol{x} = (x, y, z)$	[m]	Cartesian coordinates
$\boldsymbol{u} = (u, v, w)$	[m s^{-1}]	velocity field in Cartesian coordinates
$\pi = (p - p^e)/\rho_d^r$	[m^{2}s^{-2}]	normalized pressure perturbation
$\boldsymbol{k}$	[1]	vertical unit vector
B	[m s^{-2}]	buoyancy
F_X	[(unit of X) s^{-1}]	forcing of X (surface fluxes, radiation, absorb
X^e, X^r	[(unit of X)]	environmental/reference profile of X
~~$l_v(T)$ J latent heat of evaporation (cf. Arabas et al. (2015))~~ E_c, E_e, $C = E_c - E_e$	[s^{-1}]	condensation/evaporation rate and their balanc
g	[m s^{-2}]	magnitude of Earth's gravitational acceleratio
$\epsilon = R_v/R_d - 1$	[1]	
ρ, ρ_d	[kg m^{-3}]	density of air/dry air
$S = \mathrm{d}_z\theta_v/\theta_v$	[m^{-1}]	non-dimensional stability of the atmosphere
~~$\boldsymbol{x} = (x, y, z)$mCartesian coordinates~~ r, r_d	[m]	wet/dry radius of a SD
κ	[1]	hygroscopicity parameter of a SD
ξ	[1]	multiplicity of a SD
e_s	[Pa]	saturation partial pressure of vapor
N_c	[m^{-3}]	concentration of cloud droplets in cloudy grid
N_SD	[1]	initial number of SDs per grid cell
Δt	[s]	time step length of the dynamical core
z_i	[m]	mean height of the $q_t = 8$ g kg^{-1} isosurface
~~w_T~~w_t, w_LS	[m s^{-1}]	terminal velocity of a SD~~w_LSm s^{-1}~~/large-scal
ρ_w	[kg m^{-3}]	density of water
$\mathcal{D}_\mathrm{X}$	[(unit of X) s^{-1}]	SGS model forcing of X
K_m, K_h, K_q	[m^2 s^{-1}]	eddy viscosity/eddy diffusivities
$\boldsymbol{E}$	[s^{-1}]	deformation tensor
c_s	[1]	Smagorinsky constant
λ	[m]	mixing length
Δ	[m]	cell length scale
c_L	[1]	parameter characterizing mixing length growt
Pr, Ri	[1]	Prandtl/Richardson number

Appendix B: Condensation ~~sub-stepping~~ substepping algorithm

Consider condensation of SDs within cell i at time step n. Vector of thermodynamic conditions in that cell at the moment right before condensation is calculated is denoted by $\boldsymbol{\psi}_i^{[n]} = \left(\theta^{[n]}, q_v^{[n]}\right)_i$. Number of time steps is denoted by S_c and ~~sub-steps~~ substeps are indexed by ν, starting at $\nu = 1$. Super-droplets within cell i are numbered by μ. Vector of thermodynamic conditions that a given SD experiences at ~~sub-step~~ substep ν is denoted by $\breve{\boldsymbol{\psi}}_\mu^{[\nu]}$. Using this notation, the ~~sub-stepping~~ substepping algorithm is

$$\breve{\boldsymbol{\psi}}_\mu^{[\nu+1/2]} = \breve{\boldsymbol{\psi}}_\mu^{[\nu]} + \frac{\boldsymbol{\psi}_i^{[n]} - \breve{\boldsymbol{\psi}}_\mu^{[\nu=1]}}{S_c}, \tag{B1}$$

$$r_\mu^{2[\nu+1]} = r_\mu^{2[\nu]} + \frac{\Delta t}{S_c} \left.\frac{dr^2}{dt}\right|_{r_\mu^{2[\nu+1]}, \breve{\boldsymbol{\psi}}_\mu^{[\nu+1/2]}}, \tag{B2}$$

$$\breve{\boldsymbol{\psi}}_\mu^{[\nu+1]} = \breve{\boldsymbol{\psi}}_\mu^{[\nu+1/2]} + \boldsymbol{A}\frac{4}{3}\frac{\pi \rho_w V}{\rho_d^r} \sum_{\mu=1}^{\mu=N_i^{[n]}} \xi_\mu \left[\left(r_\mu^{2[\nu+1]}\right)^{3/2} - \left(r_\mu^{2[\nu]}\right)^{3/2} \right], \tag{B3}$$

where r_μ^2 is the square of the wet radius of the μ-th SD, $N_i^{[n]}$ is the number of SDs in cell i at time step n and $\boldsymbol{A} = \left(\theta^e l_v / \left(c_{pd} T^e\right), -1\right)$. Sum in eq. (B3) is done over all SDs in cell i at time step n. For details of the predictor-corrector algorithm for calculation of the change of radius in eq. (B2), see Eqs. (17)-(19) in Arabas et al. (2015). After the last ~~sub-step~~substep, the value of $\breve{\boldsymbol{\psi}}_\mu^{[\nu=S_c]}$ is the same for all SDs in the cell and the condensational RHS returned from the condensation algorithm is

$$R_c^{[n]} = \frac{\breve{\boldsymbol{\psi}}_{\mu=1}^{[\nu=S_c]} - \boldsymbol{\psi}_i^{[n]}}{\Delta t}. \tag{B4}$$

The initial value $\breve{\boldsymbol{\psi}}_\mu^{[\nu=1]}$ is equal to the thermodynamic conditions after condensation finished in the previous time step. Two ways of defining $\breve{\boldsymbol{\psi}}_\mu^{[\nu=1]}$ are considered ~~, which~~ that differ in the spatial cell from which this initial condition is diagnosed:

$$\breve{\boldsymbol{\psi}}_\mu^{[\nu=1]} = \left(\boldsymbol{\psi}^{[n-1]} + R_c{}^{[n-1]}\right)_{i(n-1)}, \tag{B5}$$

referred to as *per-particle* ~~sub-stepping~~substepping, and

$$\breve{\boldsymbol{\psi}}_\mu^{[\nu=1]} = \left(\boldsymbol{\psi}^{[n-1]} + R_c{}^{[n-1]}\right)_{i(n)}, \tag{B6}$$

~~what~~ a procedure we call *per-cell* ~~sub-stepping~~substepping. The notation $i(n)$ stands for the index of the cell in which the μ-th SD was at time step n. The *per-cell* ~~sub-stepping~~ substepping is less accurate, but requires less computational time and uses less memory. The reason is that in the *per-cell* method, all SDs in a given cell have the same values of $\breve{\boldsymbol{\psi}}_\mu^{[\nu]}$. Moreover, values of pressure and density do not need to be ~~sub-stepped~~ substepped in the *per-cell* method, since they are constant in time in each cell.

We expect the difference between the two substepping algorithms to be particularly large near a moving cloud edge. We test this hypothesis by simulating cloud edge advection in an idealized one-dimensional setup. Consider two spatial cells, one at saturation ($q_l = 0.029$g/kg, $N_c \approx 52$cm^{-3}) and the other subsaturated (relative humidity of 94%). The boundaries are periodic and the only processes are diffusional growth and advection. A single time step ($\Delta t = 2$s) is performed, in which Eulerian fields and SDs are advected with the Courant number equal to 1. The expected result is that the two cells exchange their contents without any condensation/evaporation taking place. The simulations are conducted for different substepping algorithms and different numbers of substeps. Deviations from the expected result are presented in table B1.

Table B1. Errors caused by substepping in the one-dimensional simulation of cloud edge advection. The error is defined as $\epsilon = \left(q_l^{\mathrm{sim}} - q_l^{\mathrm{exp}}\right)/q_l^{\mathrm{exp}}$, where q_l is the liquid water mixing ratio diagnosed at the end of the simulation from the cell to which cloud droplets were advected, i.e. the cell that initially was subsaturated. The superscripts "sim" and "exp" denote the simulated and expected values, respectively.

substepping algorithm	number of substeps	ϵ [%]
per-cell	1	0
	2	-26
	5	-40
	10	-44
per-particle	1	0
	2	0
	5	0
	10	0

The *per-cell* algorithm causes artificial evaporation of droplets and this error increases with the number of substeps, while the *per-particle* algorithm gives correct results independently of the number of substeps. Therefore we conclude that in simulations in which there is a lot of cloud edge advection it is necessary to use the *per-particle* algorithm.

However, in simulations with little cloud edge advection, the per-cell algorithm might be sufficient. We check this by using different substepping algorithms in a two-dimensional marine startocumulus simulation (the DYCOMS RF02 case, cf. section 4). To limit differences in dynamics between runs, we use the piggybacking approach (Grabowski, 2014): velocity fields from a dynamical "driver" simulation with $\Delta t = 0.1$s are used in two other "piggybacking" simulations with $\Delta t = 1$ s. No substepping is done in the driver simulation and the piggybacking simulations have 10 substeps for condensation, one with the *per-cell* and the other with the *per-particle* algorithm. Timestep length in the "driver" run is shorter than in the piggybacking runs, because we want to properly model condensation in the "driver" simulation without substepping, in order to obtain reference results. Obviously, due to the difference in time step length, velocity field used in "piggybacking" simulations is not exactly the same as in the "driver" simulations. However, averaged vertical profiles of moments of the vertical velocity are similar in the "driver" and "piggybacker", so we expect concentration of cloud droplets in the "piggybacker" to be similar to that of the "driver". We are interested in cloud droplet concentration, because droplet activation is the process that requires short time steps for condensation. To limit variability between runs caused by the Monte Carlo scheme used to initialize SD radii, a large number of SDs is used in these simulations ($N_{SD} = 1000$). Vertical profiles of concentration of cloud droplets from the ”driver” and ”piggybacker” runs are shown in fig. B1. Interestingly, the *per-cell* algorithm is in better agreement with the ”driver” simulation than the *per-particle* algorithm. The latter slightly underestimates concentration of cloud droplets, by ca. 5%. Nevertheless, we conclude that both substepping algorithms work well in simulations in which cloud edge does not move significantly.

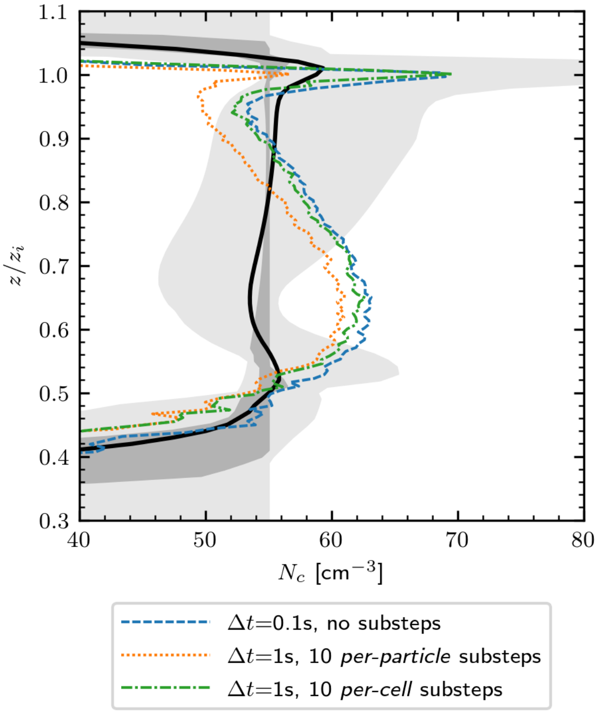

Figure B1. Vertical profiles of concentration of cloud droplets from the 2D piggybacking simulations with different substepping algorithms, averaged over the 2h-4h period. The black line and shaded regions show results from Ackerman et al. (2009) (cf. fig. 3).

Appendix C: Software implementation

UWLCM is ~~written~~ coded in the C++ language. It relies heavily on two C++ libraries developed by the cloud modeling group at the University of Warsaw: libmpdata++ (Jaruga et al., 2015) for the Eulerian component and libcloudph++ (Arabas et al., 2015) for the Lagrangian component of the model. Structure of the libmpdata++ and libcloudph++ codes and how they are used in UWLCM is schematically depicted in Fig. C1.

libmpdata++ is a set of solvers for the generalized transport equations that use the MPDATA advection scheme. The solvers are organized in a hierarchy, ordered from solvers for simple flows to solvers for more complex flows. Each more complex solver inherits from the simpler solver in the hierarchy. Such design simplifies code development, maintenance and reusability.

Apart from the hierarchy of solvers, libmpdata++ contains three other independent modules: boundary conditions, concurrency handlers and output handlers, all of which are used in UWLCM.

libcloudph++ is an implementation of three microphysical models: SDM, a single-moment bulk model and a double-moment bulk model. The SDM is implemented using the Thrust library and the CUDA programming language. Thanks to that, the SDM can be run on multi-threaded CPUs as well as on multiple GPUs.

UWLCM code is built on top of the libmpdata++ solvers. Separate parts of the UWLCM code handle different types of simulations. The ”piggybacking” code makes it possible to run kinematic simulations, i.e. simulations with a prescribed velocity field. The ”2D/3D” part of the code handles the dimensionality of the problem. The ”forcings” code specifies external forcings, so it is the part of the code that depends on simulation setup. The ”microphysics” module is responsible for handling the choice of microphysics (only Lagrangian microphysics is available in the current UWLCM release). Thanks to such code structure, different types of simulations, e.g. 2D and 3D simulations, different simulation setups or kinematic simulations, are using mostly the same source code. The highest performance is achieved when UWLCM is run on a system with GPUs. In that case, the Eulerian component is calculated on CPUs and the Lagrangian component on GPUs. Large part of these computations is done simultaneously (cf. fig. 1). The UWLCM code is open-source, under a version-control system and available from a git repository. Model output is done in the HDF5 format, ready for plotting in *Paraview*. UWLCM also includes simple software for plotting time series and vertical profiles. A number of test programs was developed for UWLCM, libcloudph++ and libmpdata++. UWLCM currently can run in parallel only on shared-memory systems. An implementation for distributed-memory systems is currently under development. UWLCM code is inspired by the *icicle* kinematic model developed by S. Arabas and A. Jaruga (https://github.com/igfuw/libcloudphxx/models/kinematic_2D).

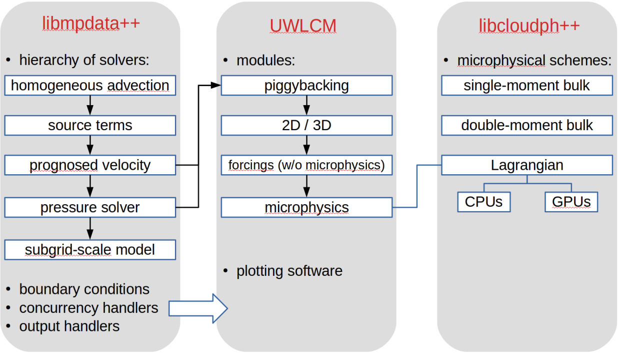

Figure C1. Schematic depiction of the structure of the code of UWLCM, libmpdata++ and libcloudph++. Black arrows denote inheritance between classes.

Author contributions. PD developed the model code with contributions from MW. PD performed the simulations. PD and HP prepared the manuscript with contributions from MW.

Acknowledgements. We thank Wojciech Grabowski for discussions about the model formulation and about the manuscript. We are grateful to the AGH Cyfronet computation center for providing computing power. This research was supported by the Polish National Science Center grant 2016/23/B/ST10/00690.

References

Ackerman, A. S., Hobbs, P. V., and Toon, O. B.: A model for particle microphysics, turbulent mixing, and radiative transfer in the stratocumulus-topped marine boundary layer and comparisons with measurements, Journal of the atmospheric sciences, 52, 1204–1236, 1995.

Ackerman, A. S., vanZanten, M. C., Stevens, B., Savic-Jovcic, V., Bretherton, C. S., Chlond, A., Golaz, J.-C., Jiang, H., Khairoutdinov, M., Krueger, S. K., Lewellen, D. C., Lock, A., Moeng, C.-H., Nakamura, K., Petters, M. D., Snider, J. R., Weinbrecht, S., and Zulauf, M.: Large-Eddy Simulations of a Drizzling, Stratocumulus-Topped Marine Boundary Layer, Monthly Weather Review, 137, 1083–1110, https://doi.org/10.1175/2008MWR2582.1, https://doi.org/10.1175/2008MWR2582.1, 2009.

Andrejczuk, M., Reisner, J., Henson, B., Dubey, M., and Jeffery, C.: The potential impacts of pollution on a nondrizzling stratus deck: Does aerosol number matter more than type?, Journal of Geophysical Research: Atmospheres, 113, 2008.

Andrejczuk, M., Grabowski, W., Reisner, J., and Gadian, A.: Cloud-aerosol interactions for boundary layer stratocumulus in the Lagrangian Cloud Model, Journal of Geophysical Research: Atmospheres, 115, 2010.

Arabas, S. and Pawlowska, H.: Adaptive method of lines for multi-component aerosol condensational growth and CCN activation, Geoscientific Model Development, 4, 15–31, https://doi.org/10.5194/gmd-4-15-2011, https://www.geosci-model-dev.net/4/15/2011/, 2011.

Arabas, S. and Shima, S.-i.: Large-eddy simulations of trade wind cumuli using particle-based microphysics with Monte Carlo coalescence, Journal of the Atmospheric Sciences, 70, 2768–2777, 2013.

Arabas, S., Jaruga, A., Pawlowska, H., and Grabowski, W.: libcloudph++ 1.0: a single-moment bulk, double-moment bulk, and particle-based warm-rain microphysics library in C++, Geoscientific Model Development, 8, 1677–1707, 2015.

Arakawa, A. and Lamb, V. R.: Computational design of the basic dynamical processes of the UCLA general circulation model, General circulation models of the atmosphere, 17, 173–265, 1977.

Chen, J.-P.: Numerical simulations of the redistribution of atmospheric trace chemicals through cloud processes, Ph.D. thesis, Pennsylvania State University, 1992.

Clark, T. L. and Farley, R. D.: Severe Downslope Windstorm Calculations in Two and Three Spatial Dimensions Using Anelastic Interactive Grid Nesting: A Possible Mechanism for Gustiness, Journal of the Atmospheric Sciences, 41, 329–350, https://doi.org/10.1175/1520-0469(1984)041<0329:SDWCIT>2.0.CO;2, htps://doi.org/10.1175/1520-0469(1984)041<0329:SDWCIT>2.0.CO;2, 1984.

Davis, M. H.: Collisions of small cloud droplets: Gas kinetic effects, Journal of the Atmospheric Sciences, 29, 911–915, 1972.

Dziekan, P. and Pawlowska, H.: Stochastic coalescence in Lagrangian cloud microphysics, Atmospheric Chemistry and Physics, 17, 13 509–13 520, https://doi.org/10.5194/acp-17-13509-2017, https://www.atmos-chem-phys.net/17/13509/2017/, 2017.

Gillespie, D. T.: The stochastic coalescence model for cloud droplet growth, Journal of the Atmospheric Sciences, 29, 1496–1510, 1972.

Grabowski, W. W.: Extracting Microphysical Impacts in Large-Eddy Simulations of Shallow Convection, Journal of the Atmospheric Sciences, 71, 4493–4499, https://doi.org/10.1175/JAS-D-14-0231.1, https://doi.org/10.1175/JAS-D-14-0231.1, 2014.

Grabowski, W. W. and Abade, G. C.: Broadening of Cloud Droplet Spectra through Eddy Hopping: Turbulent Adiabatic Parcel Simulations, Journal of the Atmospheric Sciences, 74, 1485–1493, https://doi.org/10.1175/JAS-D-17-0043.1, 2017.

Grabowski, W. W. and Smolarkiewicz, P. K.: Two-Time-Level Semi-Lagrangian Modeling of Precipitating Clouds, Monthly Weather Review, 124, 487–497, https://doi.org/10.1175/1520-0493(1996)124<0487:TTLSLM>2.0.CO;2, https://doi.org/10.1175/1520-0493(1996)124<0487:TTLSLM>2.0.CO;2, 1996.

Grabowski, W. W., Dziekan, P., and Pawlowska, H.: Lagrangian condensation microphysics with Twomey CCN activation, Geoscientific Model Development, 11, 103–120, https://doi.org/10.5194/gmd-11-103-2018, https://www.geosci-model-dev.net/11/103/2018/, 2018a.

Grabowski, W. W., Morrison, H., Shima, S.-i., Abade, G., Pawlowska, H., and Dziekan, P.: Modeling of cloud microphysics: Can we do better?, Bulletin of the American Meteorological Society, https://doi.org/https://doi.org/10.1175/BAMS-D-18-0005.1, 2018b.

Grinstein, F. F., Margolin, L. G., and Rider, W. J.: Implicit large eddy simulation: computing turbulent fluid dynamics, Cambridge university press, 2007.

Hall, W. D.: A detailed microphysical model within a two-dimensional dynamic framework: Model description and preliminary results, Journal of the Atmospheric Sciences, 37, 2486–2507, 1980.

Hoffmann, F., Raasch, S., and Noh, Y.: Entrainment of aerosols and their activation in a shallow cumulus cloud studied with a coupled LCM–LES approach, Atmospheric Research, 156, 43 – 57, https://doi.org/https://doi.org/10.1016/j.atmosres.2014.12.008, http://www.sciencedirect.com/science/article/pii/S016980951400444X, 2015.

Hoffmann, F., Noh, Y., and Raasch, S.: The route to raindrop formation in a shallow cumulus cloud simulated by a Lagrangian cloud model, Journal of the Atmospheric Sciences, 74, 2125–2142, 2017.

Jaruga, A., Arabas, S., Jarecka, D., Pawlowska, H., Smolarkiewicz, P. K., and Waruszewski, M.: libmpdata++ 1.0: a library of parallel MPDATA solvers for systems of generalised transport equations, Geoscientific Model Development, 8, 1005–1032, https://doi.org/10.5194/gmd-8-1005-2015, https://www.geosci-model-dev.net/8/1005/2015/, 2015.

Khairoutdinov, M. F. and Randall, D. A.: Cloud resolving modeling of the ARM summer 1997 IOP: Model formulation, results, uncertainties, and sensitivities, Journal of the Atmospheric Sciences, 60, 607–625, 2003.

Khvorostyanov, V. I. and Curry, J. A.: Terminal Velocities of Droplets and Crystals: Power Laws with Continuous Parameters over the Size Spectrum, Journal of the Atmospheric Sciences, 59, 1872–1884, https://doi.org/10.1175/1520-0469(2002)059<1872:TVODAC>2.0.CO;2, https://doi.org/10.1175/1520-0469(2002)059<1872:TVODAC>2.0.CO;2, 2002.

Klein, R., Achatz, U., Bresch, D., Knio, O. M., and Smolarkiewicz, P. K.: Regime of validity of soundproof atmospheric flow models, Journal of the Atmospheric Sciences, 67, 3226–3237, 2010.

Lilly, D. K.: On the numerical simulation of buoyant convection, Tellus, 14, 148–172, 1962.

Lipps, F. B. and Hemler, R. S.: A Scale Analysis of Deep Moist Convection and Some Related Numerical Calculations, Journal of the Atmospheric Sciences, 39, 2192–2210, https://doi.org/10.1175/1520-0469(1982)039<2192:ASAODM>2.0.CO;2, https://doi.org/10.1175/1520-0469(1982)039<2192:ASAODM>2.0.CO;2, 1982.

Margolin, L., Smolarkiewicz, P., and Wyszogradzki, A.: Dissipation in implicit turbulence models: A computational study, Journal of applied mechanics, 73, 469–473, 2006.

Margolin, L. G. and Rider, W. J.: A rationale for implicit turbulence modelling, International Journal for Numerical Methods in Fluids, 39, 821–841, 2002.

Morrison, H., Witte, M., Bryan, G. H., Harrington, J. Y., and Lebo, Z. J.: Broadening of modeled cloud droplet spectra using bin microphysics in an Eulerian spatial domain, Journal of the Atmospheric Sciences, https://doi.org/10.1175/JAS-D-18-0055.1, https://doi.org/10.1175/JAS-D-18-0055.1, 2018.

Naumann, A. K. and Seifert, A.: A Lagrangian drop model to study warm rain microphysical processes in shallow cumulus, Journal of Advances in Modeling Earth Systems, 7, 1136–1154, 2015.

Petters, M. D. and Kreidenweis, S. M.: A single parameter representation of hygroscopic growth and cloud condensation nucleus activity, Atmospheric Chemistry and Physics, 7, 1961–1971, https://doi.org/10.5194/acp-7-1961-2007, https://www.atmos-chem-phys.net/7/1961/2007/, 2007.

RAMS Technical Description: The Regional AtmosphericModeling System, Technical Description, http://www.atmet.com/html/docs/rams/rams_techman.pdf.

Riechelmann, T., Noh, Y., and Raasch, S.: A new method for large-eddy simulations of clouds with Lagrangian droplets including the effects of turbulent collision, New Journal of Physics, 14, 065 008, 2012.

Schmidt, H. and Schumann, U.: Coherent structure of the convective boundary layer derived from large-eddy simulations, Journal of Fluid Mechanics, 200, 511–562, 1989.

Schwenkel, J., Hoffmann, F., and Raasch, S.: Improving collisional growth in Lagrangian cloud models: development and verification of a new splitting algorithm, Geoscientific Model Development, 11, 3929–3944, 2018.

Shima, S.-i., Kusano, K., Kawano, A., Sugiyama, T., and Kawahara, S.: The super-droplet method for the numerical simulation of clouds and precipitation: A particle-based and probabilistic microphysics model coupled with a non-hydrostatic model, Quarterly Journal of the Royal Meteorological Society, 135, 1307–1320, 2009.

Smagorinsky, J.: General circulation experiments with the primitive equations: I. The basic experiment, Monthly weather review, 91, 99–164, 1963.

Smolarkiewicz, P. and Margolin, L.: Variational methods for elliptic problems in fluid models, in: Proc. ECMWF Workshop on Developments in numerical methods for very high resolution global models, pp. 137–159, 2000.

Smolarkiewicz, P. K.: Multidimensional positive definite advection transport algorithm: an overview, International Journal for Numerical Methods in Fluids, 50, 1123–1144, https://doi.org/10.1002/fld.1071, https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1071, 2006.

Smolarkiewicz, P. K.: Modeling atmospheric circulations with soundproof equations, in: Proc. of the ECMWF Workshop on Nonhydrostatic Modelling, 8-10 November, 2010, Reading, UK, pp. 1–15, 2011.

Smolarkiewicz, P. K. and Szmelter, J.: A nonhydrostatic unstructured-mesh soundproof model for simulation of internal gravity waves, Acta Geophysica, 59, 1109, 2011.

Smolarkiewicz, P. K., Kühnlein, C., and Wedi, N. P.: A consistent framework for discrete integrations of soundproof and compressible PDEs of atmospheric dynamics, Journal of Computational Physics, 263, 185 – 205, https://doi.org/https://doi.org/10.1016/j.jcp.2014.01.031, http://www.sciencedirect.com/science/article/pii/S0021999114000588, 2014.

Smolarkiewicz, P. K., Kühnlein, C., and Wedi, N. P.: Semi-implicit integrations of perturbation equations for all-scale atmospheric dynamics, Journal of Computational Physics, 376, 145–159, 2019.

Stevens, B., Feingold, G., Cotton, W. R., and Walko, R. L.: Elements of the microphysical structure of numerically simulated nonprecipitating stratocumulus, Journal of the atmospheric sciences, 53, 980–1006, 1996.

Stevens, B., Lenschow, D. H., Vali, G., Gerber, H., Bandy, A., Blomquist, B., Brenguier, J. L., Bretherton, C. S., Burnet, F., Campos, T., Chai, S., Faloona, I., Friesen, D., Haimov, S., Laursen, K., Lilly, D. K., Loehrer, S. M., Malinowski, S. P., Morley, B., Petters, M. D., Rogers, D. C., Russell, L., Savic-Jovcic, V., Snider, J. R., Straub, D., Szumowski, M. J., Takagi, H., Thornton, D. C., Tschudi, M., Twohy, C., Wetzel, M., and van Zanten, M. C.: Dynamics and Chemistry of Marine Stratocumulus—DYCOMS-II, Bulletin of the American Meteorological Society, 84, 579–594, https://doi.org/10.1175/BAMS-84-5-579, https://doi.org/10.1175/BAMS-84-5-579, 2003.

Stevens, D. E., Ackerman, A. S., and Bretherton, C. S.: Effects of domain size and numerical resolution on the simulation of shallow cumulus convection, Journal of the atmospheric sciences, 59, 3285–3301, 2002.

Unterstrasser, S., Hoffmann, F., and Lerch, M.: Collection/aggregation algorithms in Lagrangian cloud microphysical models: rigorous evaluation in box model simulations, Geoscientific Model Development, 10, 1521, 2017.