https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

# BinMod1D v1.0.10: A Python package for explicitly simulating 1D collisional coalescence/breakup processes with corresponding polarimetric radar signatures

Edwin Lee Dunnavan[1,2]

[1]Cooperative Institute for Severe and High-Impact Weather Research and Observations, Norman, OK, USA
[2]National Severe Storms Laboratory, Norman, OK, USA

**Correspondence:** Edwin Lee Dunnavan (edwin.dunnavan@noaa.gov)

**Abstract.** This paper details a computationally efficient and versatile Python package (BinMod1D v1.0.10) that explicitly evolves spectral bin distributions and corresponding polarimetric radar variables for rain or snow according to atmospheric collisional coalescence and breakup processes. BinMod1D can be executed as a box model, a 1D steady-state model in height, or a full (time and height) 1D column model utilizing multiple particle categories, each of which can have their own densities, aspect ratios, and fall speeds. Forward simulations of polarimetric radar variables are implemented using standard Rayleigh analytic scattering equations. Two-moment (mass and number) or one-moment (mass only) particle interaction calculations follow a source-based approach but with parallelizable just-in-time (JIT) compilation for high performance. BinMod1D box model solutions are validated using analytic solutions of collision-coalescence using a variety of kernels as well as for breakup and the steady-state balance of coalescence with breakup. BinMod1D capabilities are demonstrated through steady-state simulations of rainfall and snow signatures, as well as vertical profiles of diverse meteorological scenarios. Convergence and timing tests are provided for the meteorological scenario of a cloud to rain transition using a realistic collision kernel and fragment distribution. BinMod1D is intended to enable cloud microphysics and weather radar researchers to efficiently simulate vertical profiles of complex weather events. Such a tool can be used to provide reference solutions for training machine learning models and validating various retrieval methodologies.

## 1 Introduction

Reference solutions to complex weather phenomena provide a useful way to develop and evaluate new methods for retrieving microphysical variables from weather radar measurements. Such retrievals can involve simple analytic formulas (e.g., Ryzhkov and Zrnić, 2019) or more involved machine learning models (e.g., Chase et al., 2021) that connect radar products like reflectivity to microphysical parameters such as rain or snow rates. Typically, these retrieval equations are developed through a combination of theory and observations, but often without rigorous consideration of microphysical and scattering uncertainties as well as the precise nature of the simulated particle size distribution (PSD). Instead, parameteric approaches, often by assuming an underlying exponential or gamma distribution, are used to connect observed radar variables to desirable microphysical quantities. As a result, however, these parametric approaches sever the important connections between the quantities that are inferred from radar signatures with their underlying microphysical causes.

Of the many microphysical processes in precipitation physics, collision-coalescence and collisional breakup represent two of the most important yet difficult to explicitly simulate. This difficulty can be seen directly from the general equation for collision-coalesence and collsional breakup in time and in height (ignoring horizontal advection terms), which can be expressed as (cf. Mitchell, 1991)

$$
\begin{aligned}
\frac{\partial n(m,z,t)}{\partial t} + \frac{\partial\left[v_t(m)n(m,z,t)\right]}{dz} = & \frac{1}{2}\int\limits_{x=0}^{m} C(m-x,x)n(m-x,z,t)n(x,z,t)dx - n(m,z,t)\int\limits_{x=0}^{\infty} C(m,x)n(x,z,t)dx + \\
& \frac{1}{2}\int\limits_{x=0}^{\infty}\int\limits_{y=0}^{\infty} B(x,y)P(m|x,y)n(x,z,t)n(y,z,t)dydx - \\
& n(m,z,t)\int\limits_{y=0}^{\infty}\frac{B(m,y)n(y,z,t)}{m+y}dy\int\limits_{x=0}^{m+y} xP(x|m,y)dx,
\end{aligned}
\tag{1}
$$

where $n(m,z,t)$ is the number distribution function for mass $m$, height $z$, and time $t$; $v_t(m)$ is the fall speed for particles of mass $m$; $C(x,y)$ and $B(x,y)$ are the collision-coalescence and collision-breakup kernels, respectively; and $P(m|x,y)$ is the conditional fragment distribution that determines the number of mass $m$ fragments generated from collisions involving particles of mass $x$ and $y$. The left hand side of Equation (1) represents the tendency and sedimentation terms of the evolving PSD (in terms of mass) whereas the first two right hand side terms represent the gain and loss terms for coalescence and the last two terms represent the gain and loss terms for breakup. Equation (1) reduces to the well-known Smoluchowski equation (Smoluchowski, 1917) when the sedimentation and breakup terms are neglected. The Smoluchowski equation is also frequently referred to by many other names such as the coagulation equation, the stochastic collection equation (Tzivion et al., 1987; Feingold et al., 1988; Bott, 2000; Simmel et al., 2002; Prat and Barros, 2007a), the kinetic collection equation (Alfonso et al., 2008) and the scalar transport equation (McLeod, 1964; Drake, 1972; Drake and Wright, 1972). Equation (1) is difficult to solve for several reasons. First, the integro-differential equation is functional in the coalescence gain term through $C(m-x,x)$ and $n(m-x,z,t)$. As such, standard finite difference numerical methods are not sufficient for directly solving Equation (1). Secondly, the integral regions are constrained by the particle masses themselves as seen in Equation (1) with the coalescence gain term and the breakup loss term. This entails that different integration methods are potentially required for different particle $m$ masses. Finally, there are large uncertainties involving the nature and form of the breakup conditional distribution $P(m|x,y)$. By its nature, $P(m|x,y)$ should respect detailed mass conservation in the loss term (i.e., $\int\limits_{x=0}^{m+y} xP(x|m,y) = m+y$), however this condition isn't always exactly met (Feingold et al., 1988). For rain collisional breakup, this has led to several intricate methods that use mixture distributions involving, for example, Gaussian and lognormal distributions to represent the different types of breakup events (i.e., filament, sheet, or disk) as constrained by laboratory measurements (Low and List, 1982a, b; McFarquhar, 2004; Straub et al., 2010).

The complexity of Equation (1) has historically led to roughly three different types of numerical solution methods: *point-based* methods, *target-based spectral moment* methods, and *source-based spectral moment* methods.

*Point-based* approaches such as Berry and Reinhardt (1974) approximate the number distribution function as a set of discrete points and attempt to solve Equation (1) by interpolating the functional gain term. This approach is computationally intensive compared to other methods but is considered to be quite accurate and is often used to provide reference solutions for other methods (e.g., Wang et al., 2007). The gain and loss terms are handled separately which leads to mass not being exactly conserved, although with enough points this lack of conservation can be mitigated somewhat.

*Target-based spectral moment* methods instead evolve bin moments (i.e., number and/or mass) and were originally developed by Bleck (1970) and Enukashvily (1980) and improved upon by Tzivion et al. (1987) and Feingold et al. (1988). In the moment-based approach, Equation (1) is integrated for each $k$ bin with the kernel $x^r dx$ from the left bin edge $x_k$ to the right bin edge $x_{k+1}$. This transforms Equation (1) into

$$
\begin{aligned}
\frac{\partial M_k^r(z,t)}{\partial t} + \frac{\partial S_k^r(z,t)}{dz} = & \frac{1}{2}\int\limits_{m=x_k}^{x_{k+1}} m^r dm \int\limits_{x=0}^{m} C(m-x,x)n(m-x,z,t)n(x,z,t)dx - \\
& \int\limits_{m=x_k}^{x_{k+1}} m^r n(m,z,t)dm \int\limits_{x=0}^{\infty} C(m,x)n(x,z,t)dx + \\
& \frac{1}{2}\int\limits_{m=x_k}^{x_{k+1}} m^r dm \int\limits_{x=0}^{\infty}\int\limits_{y=0}^{\infty} B(x,y)P(m|x,y)n(x,z,t)n(y,z,t)dydx - \\
& \int\limits_{m=x_k}^{x_{k+1}} m^r n(m,z,t)dm \int\limits_{y=0}^{\infty} \frac{B(m,y)n(y,z,t)}{m+y}dy \int\limits_{x=0}^{m+y} xP(x|m,y)dx,
\end{aligned}
\tag{2}
$$

where $M_k^r(z,t) \equiv \int\limits_{m=x_k}^{x_{k+1}} m^r n(m,z,t)dm$ is the $r$th mass moment of bin $k$ and $S_k^r(z,t) \equiv \int_{x_k}^{x_{k+1}} m^r v_t(m)n(m,z,t)dm$ is the bin's $r$th mass moment vertical flux. Often, Equation (2) is used to generate ordinary differential equations for evolving the mass moment ($r=1$) and the number moment ($r=0$). While this integral transform increases the dimensionality by one, the advantage is that differential equations for evolving moments of the number distribution function can be represented, at least partially, in terms of the moments themselves. This can be seen schematically in Figure 1 for collision-coalescence (panel a) and breakup (panel b). The collision-coalescence gain term for bin $k$ can be calculated by splitting the full integral region (red) into quadrilateral regions for all $i$ and $j$ bin interactions. The loss region for the $k$ bin (blue) are simply rectangular integration regions for each $k$ ($j$) bin interaction with the $i$ bin. The gain term is often very lengthy and suffers from the problem of closure, i.e. the subregions are expressed in terms of higher-order moments of the $i$ and $j$ bins. This has led to different closure techniques. For example, Tzivion et al. (1987) uses a nondimensional integral parameter to connect higher order moments to lower order moments whereby the error is small and the maximum possible error can be determined using Schwarz's inequality for a given bin resolution. Because of this error, mass conservation might not be completely satisfied, however the error is apparently minimal. These tricky integral regions are even more complicated when considering collisional

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

breakup as shown in Figure 1b where these quadrilateral gain and loss regions are represented as three dimensional volumes that are constrained and separated by the $x+y=m$ and $x=y+m$ planes. Other methods such as Prat and Barros (2007a) have simplified the lengthy gain and loss integral terms by using the so-called "fixed-pivot" method pioneered by Kovetz and Olund (1969) and later by Kumar and Ramkrishna (1996). In this approach, particles are redistributed by conserving mass and number through a linear interpolation between bin edges and the newly generate particles from the gain terms.

*Source-based spectral moment* methods are perhaps the most conceptually simple to understand and implement. In these approaches such as detailed in Bott (1998, 2000); Simmel et al. (2002); Wang et al. (2007), each bin-pair interaction (i.e., each pair of $i$ and $j$ bin) and integral region are computed *separately* and then redistributed to the target distribution bins according to the gain integral regions for each target bin that map to the bin pair interaction source region. These methods utilize Eulerian geometric mass grids as this ensures that the redistribution will always include a maximum of two bins: a $k$ bin and $k+1$ bin. By considering each source bin-pair interaction, mass conservation for each bin-pair interaction can be ensured unlike the first two methods. Various source-based models treat the $k$ and $k+1$ bin partitioning differently. Bott (1998) uses an advection or "flux" scheme whereby the total mass of the $i$ and $j$ collection event are placed in the $k$ bin and a certain fraction of this total mass is then transferred to the $k+1$ bin by calculating a fractional parameter that is analogous to a weighted Courant number. The Simmel et al. (2002) method is even simpler in that it approximates the collection kernel as a constant such that each source bin-pair interaction provides a fully closed set of ordinary differential equations in terms of the mass and number moments of the interacting distribution bins. A time-dependent linear distribution is then fit between the $k$ and $k+1$ bins in order to partition the total mass and number between each gain bin.

The source-based method of the Bin Integral Method with Gauss Quadrature (BIMGQ) model by Wang et al. (2007) provided a more rigorous and exact way to perform the $k$ and $k+1$ bin partitioning compared to the linear flux method of Bott (1998) and the improved version of Bott (2000) as well as the linear discrete method of Simmel et al. (2002). Wang et al. (2007) found that the source bin-pair integral area regions can be calculated exactly through purely geometric considerations and a bilinear approximation of the collection kernel leads to theoretically polynomial terms for each source bin pair. In this way, the gain terms for the $k$ or $k+1$ bin are expressed in terms of the full *source* region defined by the $i$ and $j$ bin interaction. Therefore, only the $k$ or $k+1$ integration calculation is needed as the other bin's calculation can be determined by subtracting the other bin's contribution from the total rectangular source region. In order to simplify these integrations however, Wang et al. (2007) used a Gaussian quadrature routine which, with enough nodes, would yield the exact polynomial values. Wang et al. (2007) further uses the two predicted moments to diagnose a subgrid linear distribution using the inversion formulas of Enukashvily (1980). Thus, the model evolves both total $k$ bin mass and number whereas Bott (1998) and Bott (2000) evolves the mass distribution function directly by transferring bin-pair interacting masses and then using this mass transfer to fit the new mass distribution function.

While classic Fortran-based collision-coalescence spectral bin solvers exist such as the TAU model (Feingold et al., 1988) and the Bott (1998, 2000) scheme, there has recently been an adoption toward more accessible, modern languages. This transition toward modern accessibility is also reflected in 1D column models and coalescence-based microphysics solvers. For instance, the LCM1D Lagrangian super-droplet model by Brdar and Seifert (2020) provides a 1D column model that

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

simulates cloud microphysics. Similarly, Cloudy.jl from Bieli et al. (2022) provides a Julia-based bulk microphysics scheme for simulating cloud microphysics and coalescence. Python-based packages such as the DustPy package from Stammler and Birnstiel (2022) also simulate coagulation but for different disciplines such as astrophysics instead. For weather radar purposes, it is often not necessary to produce full 3D or 2D simulations when data is often displayed and analyzed in a time/height format such as for quasi-vertical profiles (Ryzhkov et al., 2016), range-defined quasi vertical profiles (Tobin and Kumjian, 2017), columnar vertical profiles (Murphy et al., 2020), Path-columnar vertical profiles (Bukovčić and Krause, 2026). In these cases, it would be convenient for users to specify some initial distribution and then efficiently produce corresponding spectral distributions as well as microphysical and forward simulated radar variables representing mean vertical profiles. Thus, users could use such a simplified model to investigate plausible microphysical explanations for observed vertical radar profiles or for developing retrieval methods from such vertical profiles.

To this end, this paper details a new class-based Python package “BinMod1D” which provides an efficient and versatile spectral bin model along with useful plotting methods. The previous collision-coalescence and breakup methods discussed above often focus on a box model setup and only account for interactions within a single particle class. In contrast, BinMod1D can incorporate as many particle classes as desired as well as custom coalescence and breakup kernels and fragment distributions. Furthermore, the model can be run as a box model, a height-based steady-state model, or a full 1D time/height model. In particular, the steady-state versions of Equations (1) and (2) are functionally distinct from the box model and 1D column model in the sense that the tendency term is omitted. Therefore, the ability of BinMod1D to directly simulate steady-state vertical profiles drastically decreases the computational time compared to running the 1D column model to its steady state as done in several rainshaft studies (e.g., Prat and Barros, 2007b; Prat et al., 2012; Kumjian and Prat, 2014). The BinMod1D model is based upon the source-based approach of Wang et al. (2007) but with several computational optimizations. For example, the model heavily utilizes Numba (Lam et al., 2015) which automatically parallelizes calculations for computationally intensive model runs. The model can also be run using the 2-moment (mass and number) subgrid linear distribution approach of Wang et al. (2007) or as a 1-moment (mass only) subgrid uniform distribution approach. The BinMod1D methodology is described in more detail in Section 2 along with the Python class architecture. Section 3 provides several box model, steady-state, and column model results derived from Jupyter notebook examples available in the BinMod1D v1.0.10 repository (Dunnavan, 2026a). Section 4 performs a computation speed and accuracy test using a realistic meteorological scenario for both two-moment and one-moment calculations. Finally, Section 5 discusses several potential uses of BinMod1D and likely future additions and improvements to the package.

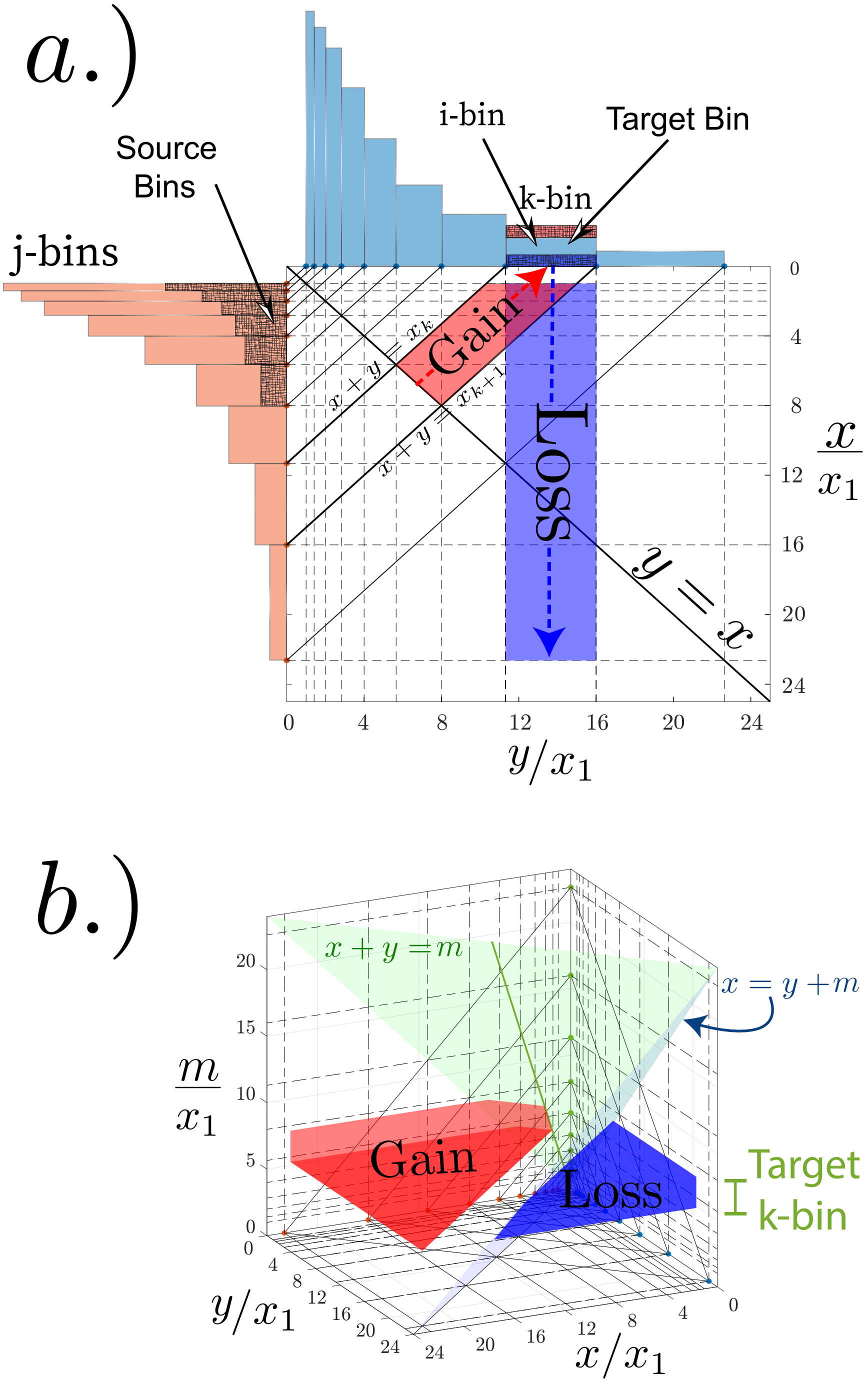

**Figure 1.** The spectral bin "method-of-moments" target-based integration regions from Feingold et al. (1988) for: (a) collision-coalescence and (b) collisional breakup where $s_{\mathrm{bin}} = 2$.

## 2 Methodology

### 2.1 Numerical integration method

For a single distribution with bin $k$, the moment equations (Equation (2)) can be cast in the following discrete form for moment order $r$:

$$\frac{\partial M_k^r(z,t)}{\partial t} + \frac{\partial [S_k^r(z,t)]}{dz} \approx \frac{\partial M_k^r(z,t)}{\partial t} + v_{t,k}\frac{\partial M_k^r(z,t)}{dz} = \sum_{i+j\rightarrow k} \hat{C}^r_{\text{gain},k}(i,j,z,t) + \sum_{\max(i,j)\geq k\rightarrow k} \hat{B}^r_{\text{gain},k}(i,j,z,t) - \sum_{i=k,j|i,j=k} \hat{D}^r_{\text{loss},k}(i,j,z,t) = I_k^r(z,t), \qquad (3)$$

where $\hat{D}^r_{\text{loss},k}(i,j,z,t) \equiv \hat{C}^r_{\text{loss},k}(i,j,z,t) + \hat{B}^r_{\text{loss},k}(i,j,z,t)$ are the combined bin moment losses from the $i-j$ bin-pair interaction[1], and the sedimentation shown as the second term on the left hand side of Equation (3) is approximated using the bin mass midpoint fall speed value in order to achieve moment closure when solving the stochastic collection/breakup equation. The terms $C^r_{\text{gain},k}(i,j,z,t)$ and $B^r_{\text{gain},k}(i,j,z,t)$ are the gain terms for each $k$ bin moment $M_k^r$ due to coalescence and breakup, respectively for $i-j$ bin-pair. Each term on the right hand side represents a separate integral

$$\hat{C}^r_{\text{gain},k}(i,j,z,t) = \int_{\Omega_k} x^r C(x,y|i,j) n_{x|i,j,z,t}(x) n_{y|i,j,z,t}(y) d\Omega_k \qquad (4a)$$

$$\hat{B}^r_{\text{gain},k}(i,j,z,t) = \int_{m=k}^{k+1} m^r \int_{x=x_i}^{x_{i+1}} \int_{y=y_i}^{y_{i+1}} B(x,y|i,j) P(m|i,j) n_{x|i,j,z,t}(x) n_{y|i,j,z,t}(y) dm dx dy \qquad (4b)$$

$$\hat{D}^r_{\text{loss},k}(i,j,z,t) = \int_{x=x_i}^{x_{i+1}} \int_{y=y_i}^{y_{i+1}} x^r \left[C(x,y|i,j) + B(x,y|i,j)\right] n_{x|i,j,z,t}(x) n_{y|i,j,z,t}(y) dx dy \qquad (4c)$$

where

$$C(x,y|i,j) \equiv E_{\text{col}} E_s K(x,y|i,j) \qquad (5a)$$

$$B(x,y|i,j) \equiv E_{\text{col}} (1-E_s) E_b K(x,y|i,j), \qquad (5b)$$

and $K(x,y|i,j)$ is the collection kernel which dictates the physics of the $x$ and $y$ interactions in the $x_i \leq x \leq x_{i+1}$ and $y_i \leq y \leq y_{i+1}$ rectangular space for the $i-j$ bin-pair. $\Omega_k$ represents the region of the $i-j$ bin-pair $(x,y)$ space that leads to a gain in bin $k$.

---

[1] Combining the coalescence and breakup losses like this only works because $E_s$ and $E_b$ are constants. Future work will allow for $E_s$ and $E_b$ to be parameterized separately.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

The awkward summation notation in Equation (3) indicates the conditions under which $i-j$ bin-pair interactions leads to a gain or loss in the $k$ bin. For the first term (coalescence), this means summing all $i-j$ pairs where $x+y=m$ such that $x_k \leq m \leq x_{k+1}$. For the second term (breakup) this means summing all $i-j$ pairs where $k \leq \max(i,j)$ such that the bin-pair interaction leads to a gain in the $k$ bin. For the third term (combined total loss), this means summing all $i-j$ pairs that involve bin $k$. Note that cases of self-collection (i.e., collection of bins from the same distribution) necessarily requires the calculation of only one of these pairs. $I_k^r(z,t)$ therefore represents the full interaction of all $i-j$ bin pairs that lead to moment changes in the $k$ bin. Therefore, BinMod1D solves one of the three following equations depending upon the user's chosen mode

$$\frac{dM_k^r(t)}{dt} = I_k^r(t) \qquad \text{(Box Model)} \tag{6a}$$

$$v_{t,k}\frac{dM_k^r(z)}{dz} = I_k^r(z) \qquad \text{(Steady-state Model)} \tag{6b}$$

$$\frac{\partial M_k^r(z,t)}{\partial t} + v_{t,k}\frac{\partial M_k^r(z,t)}{\partial z} = I_k^r(z,t) \qquad \text{(Full 1D Column Model),} \tag{6c}$$

for either $r=1$ (mass; 1 moment) or $r=1$ and $r=0$ (mass and number; 2 moments). These equations for $M_k^1(z,t) \equiv M_k(z,t)$ (1 moment) or $M_k(z,t)$ and $M_k^0(z,t) \equiv N_{t,k}(z,t)$ (2 moment) are numerically solved using an upwind advection scheme in height and a generalized Runge-Kutta method (with options of 1st to 4th order) in time. Adaptive time steps are also employed to ensure numerical stability where the target timstep is determined from either the `dt` parameter or, for the steady-state mode, the calculated residence time for each bin while solving $I_k^r(z,t)$. Generally these split time steps are not necessary and are not even used. However, these additional steps become necessary in regimes with, for example, large numbers of particle interactions with aggressive breakup.

The main computational load and complexity of the BinMod1D package comes from using the Wang et al. (2007) approach to solve $I_k^r$ for $r=1$ (mass; one-moment prediction) or $r=1$ and $r=2$ (mass and number; two-moment prediction). To simplify the calculations, each $I_k^r$ term is computed *separately* and then recombined for each $k$ bin for each distribution. This is because each $i-j$ interaction will generate gains in multiple, scattered bins. To do this, BinMod1D efficiently utilizes "just-in-time" (JIT) compilation using the Numba Python package to perform these interaction calculations *in parallel for all bin-pair interactions at all heights at once*. This leads to BinMod1D effectively using machine code to loop through each possible bin-pair interaction which, in turn, leads to fast calculations with very low memory usage even when utilizing multiple distributions in the full 1D column model mode. Several core BinMod1D components, including the parallelizable interaction kernel computations, were algorithmically streamlined using Google Gemini Pro as a code-refinement tool to ensure maximum computational speed in standard Python environments.

Figure 2 shows a schematic of example interactions for self-collection ($j-j$) and cross-collection ($i-j$) following the collision-coalescence source-based approach of Wang et al. (2007). For this schematic, it is assumed that a distribution with $i$ bins is interacting with a distribution of $j$ bins to produce particles that will be added to a distribution with $k$ bins. Each

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

time/height step therefore calculates each combination of $i$ and $j$ bin-pair interactions and transfers particles (i.e., total mass or total mass and number) from the $i$ and $j$ bin to the $k$ and $k+1$ bins. Similarly, breakup transfers particles from the $i$ and $j$ bins to the $k = 1,2,\ldots,k$ bins. Here, distribution $k$ and $j$ are assumed identical although the procedure works if $k$ and $j$ are different.

As in Wang et al. (2007) and other models, BinMod1D uses a geometric mass grid, where bin edges are determined by $x_i \equiv x_1 2^{(i-1)/s}$ for $i = 1,2,3,\ldots,N_{\text{bin}}+1$ (equation 9 from Wang et al., 2007), self-collection of a $j$ and $j$ bin results in a gain to the $k = j+s$ bin (green lines in Figure 2a). This can be seen by $x_j + x_j = 2x_j \rightarrow x_{j+s}$ and $x_{j+1} + x_{j+1} = 2x_{j+1} \rightarrow x_{j+s+1}$ mappings (skewed green arrows). The gain integrals for cross collection, in general, will cover two bins: $k$ and $k+1$. An example of cross-collection is shown by the rectangular red/purple region in Figure 2a. Here the $x+y = x_{k+1}$ line which delineates the $k$ and $k+1$ bins cuts directly through the rectangular source region defined by the $i$ and $j$ bin interaction. As a result, the quadrilateral region that is above this line maps to the $k$ (red) bin whereas the quadrilateral region below this line maps to the $k+1$ (purple) bin. As shown by Figure 2b, these quadrilateral regions can be segmented into rectangular and triangular parts, each of which can be integrated separately. As stated in Section 1, only one of these quadrilateral regions need to be calculated as the other one can be determined through continuity with the full rectangular region integration needed for computing the loss term for the $i$ and $j$ bin interaction. While the projected integration region for the mass and number gain integrals cuts through both the $k$ and $k+1$ bins, the values will be added to each respective bin in its entirely. This is shown schematically in Figure 2 as "smearing" the total added amounts from the left bin edge to the right edge (shown here as red and purple colored transparencies). Each integration region type is therefore treated separately and the general solution is to treat each $k$ or $k+1$ bin sub region as either a triangle, rectangle, or a triangle and a rectangle as in Figure 2b. As in Wang et al. (2007), moment closure is ensured by performing a bilinear interpolation of the collision/coalescence kernel in the form: $K(x,y) \equiv a + bx + cy + dxy$, for each source bin-pair interaction. Notice that this form is slightly different than equation 22 in Wang et al. (2007) in order to elucidate the polynomial form of the bilinear collection kernel approximate in the integrand of each integral:

$$K(x,y)n_x(x)n_y(y) \approx K_{00} + K_{10}x + K_{01}y + K_{11}xy + K_{20}x^2 + K_{21}x^2y + K_{02}y^2 + K_{12}xy^2 + K_{22}x^2y^2. \tag{7}$$

As can be seen from Equation (4) and Equation (5), all integrals use the same collection kernel $K(x,y|i,j)$. Therefore, the four types of integrals that need to be solved for each term are

$$\int\limits_{\Omega_R} K(x,y)n_x(x)n_y(y)d\Omega_R \tag{8a}$$

$$\int\limits_{\Omega_R} x\,K(x,y)n_x(x)n_y(y)d\Omega_R \tag{8b}$$

$$\int\limits_{\Omega_R} y\,K(x,y)n_x(x)n_y(y)d\Omega_R \tag{8c}$$

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

$$\int_{\Omega_R} (x+y)\, K(x,y) n_x(x) n_y(y) d\Omega_R, \tag{8d}$$

where $\Omega_R$ is either a rectangular region, a triangular region, or a polygonal region that can be split into a triangular and a rectangular region. Rectangular regions can simply be integrated directly into polynomials of one degree higher than in the integrand. However, the number of polynomials from analytical integration of the triangular regions increases geometrically with higher order integrand polynomials. For these triangular regions, the number of monomials for the mass gain integrals becomes very high. To simplify the calculations, the Wang et al. (2007) quadrature idea is used to calculate these triangular region integrals using the 7-point Dunavant rule (Dunavant, 1985) to determine weights and locations for the quadrature rule. The 7-point rule is used here specifically because this rule produces exact values for triangular region integrations with polynomial integrands up to order 5 which is the maximum possible order for subgrid linear distributions of BinMod1D 2-moment scheme.

The Wang et al. (2007) method described above pertains to coalescence only. However, the same general source-based method can be extended to include collisional breakup as well. This new method for collisional breakup is shown schematically in Figure 3 for an example $i$ and $j$ bin-pair interaction. The BinMod1D model first calculates the total breakup mass (1 moment) or mass and number (2 moments) for the $i-j$ interaction using Equation (4c). Then, the conditional fragment distribution $P(m|x,y)$ is determined by taking the assumed fragment distribution form $P_f(m)$ and scaling this distribution by ensuring mass continuity with the total mass integral from the full rectangular $i-j$ region (i.e., $P_{fm,i-j} = P_{fm}/\Delta M_{b,i-j}$). Then, the total mass (or mass and number) are binned by integrating each $k$ bin from $k=0$ to $k=\max(i,j)$[2]. This is shown schematically as partitioning the full rectangular integration region into $k$ rectangular subregions, each of which represents the integrated amounts added to each $k$ bin for the $j$-$i$ bin interaction. Users can specify various fragment distributions by either using examples from the `habits.py` module or by specifying the fragment dictionaries themselves. These integrations are computed beforehand for a fixed fragment distribution form.

---

[2] The formulation of the stochastic breakup equation used in target-based spectral moment approaches such as Feingold et al. (1988) actually considers the fragment distribution to span from $m=0$ to $m=x+y$ as shown in Equations (1) and (2). The physical intuition behind these bounds stems from the logic that a fragment cannot be greater than the combined mass of the two interacting species. This upper bound could be plausible if coalescence and breakup are conceptually and mathematically thought of as mutually independent processes (i.e., coalesced particles can then break up in the same time step). However, because BinMod1D considers coalescence and breakup to be mutually exclusive events within a timestep, the maximum bin is instead taken to be the maximum bin of the $i-j$ pair. The $m=x+y$ assumption is therefore used in the idealistic kernels in order to compare the BinMod1D results with the analytical breakup and steady-state coalescence-breakup solutions of Feingold et al. (1988) in Section 3.1

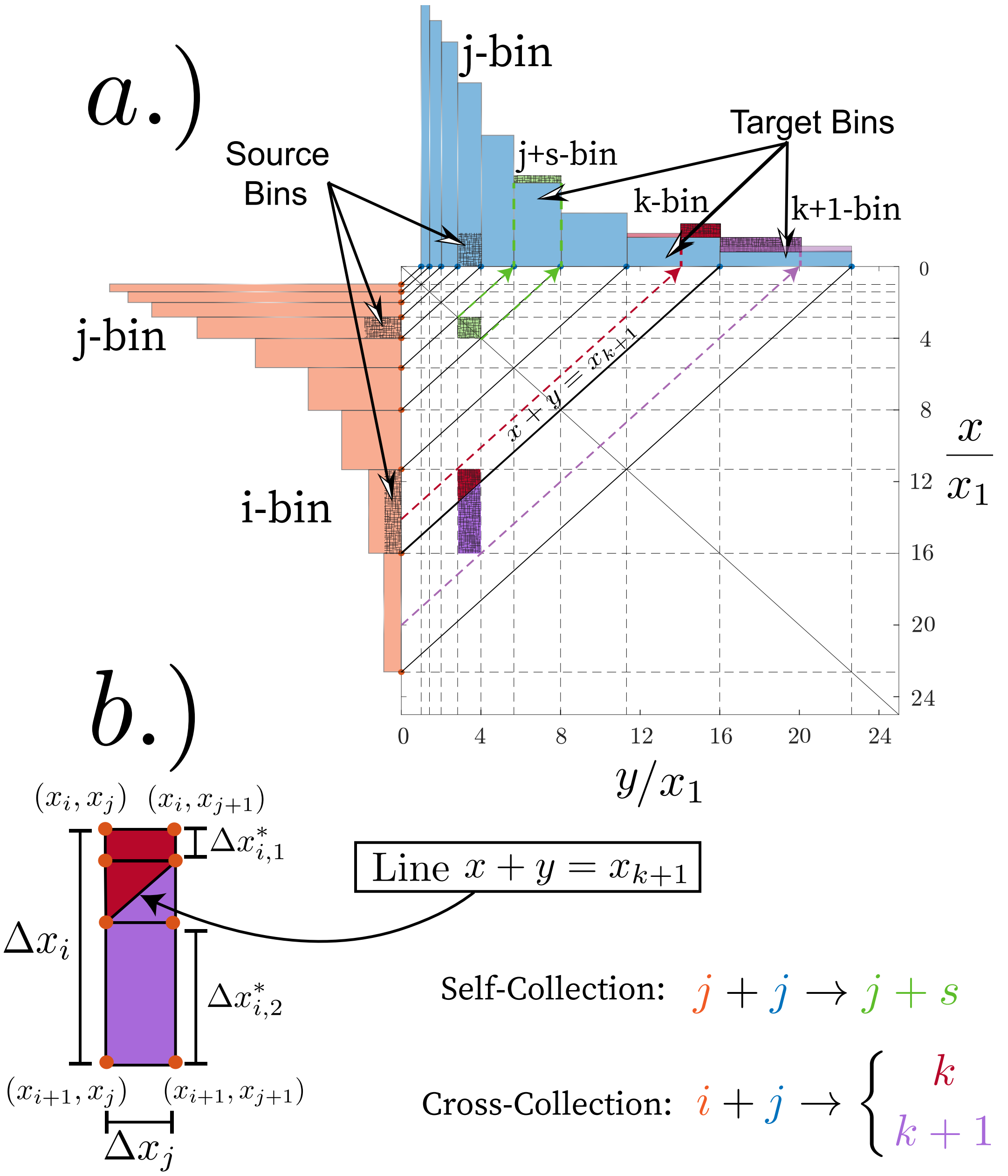

**Figure 2.** The source-based spectral bin model of Wang et al. (2007) (cf. their Figure 2) using $s = 2$ bin resolution. (a) Shows an example of the self-collection square integration region and target mapping (j+j; green), whereas the rectangular region (b) illustrates cross-collection (i+j). The geometry in (b) illustrates how the integration is handled exactly by splitting the integrals into rectangles and triangles that are used to segregate the target amounts in k and k+1 bins (see Figure 4 from Wang et al., 2007).

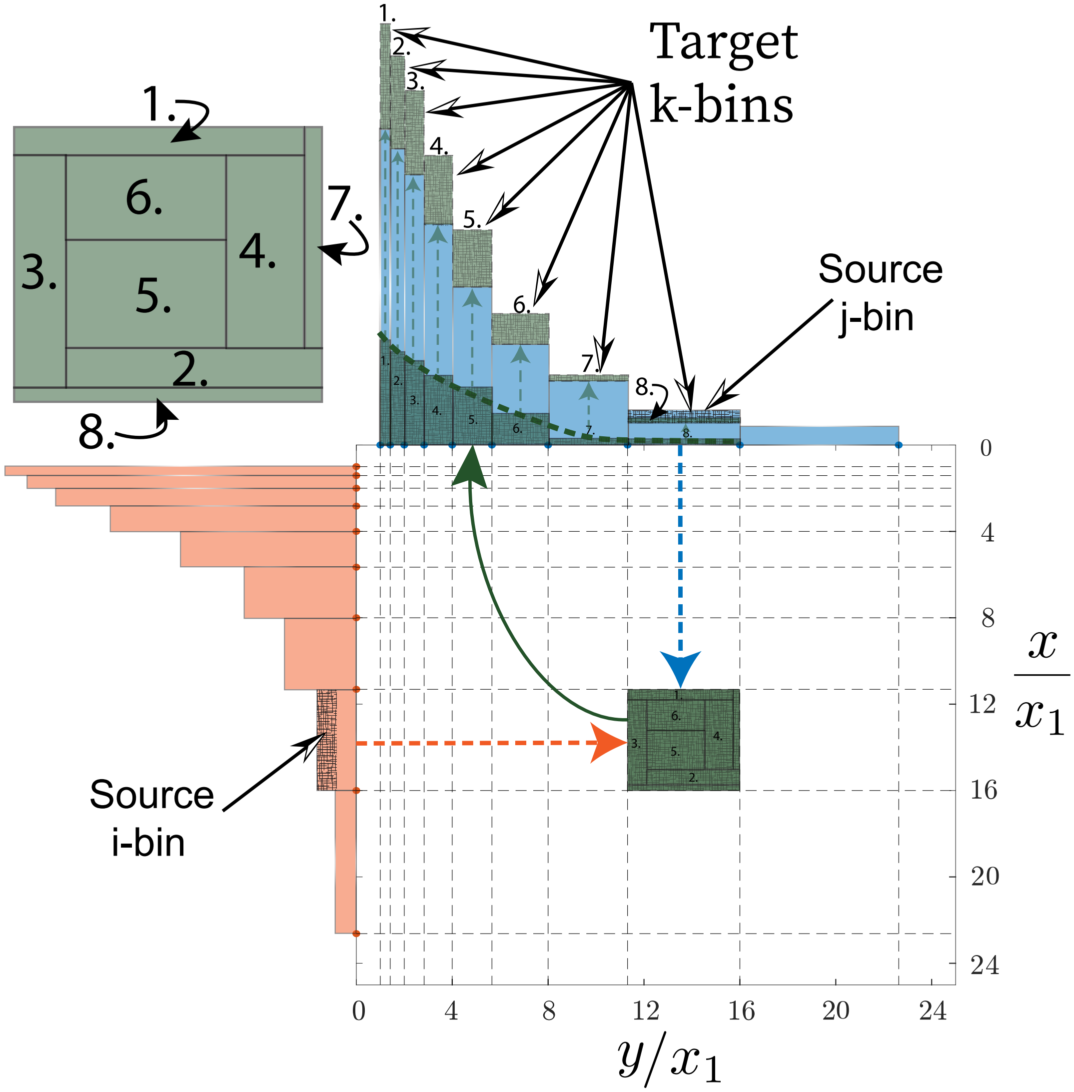

**Figure 3.** The collisional breakup methodology of BinMod1D. Each collision event between $i$ and $j$ bins represents a rectangular integration space from which the total mass is determined. The total mass is binned following the $P(m|x,y)$ conditional distribution (dashed green line) and then rescaled by ensuring mass continuity of the entire rectangular integral region. Schematically, this can be thought of as breaking the source bin-pair interaction integration region into $k$ pieces (see inset) which are then normalized by the total mass, binned, and added to the $k$ distribution (here shown to be identical to the $j$ distribution).

BinMod1D diagnoses each distribution's linear subgrid distribution set of parameters after each time/height step when $N_k$ and/or $M_k$ evolves. This form is given as

$$n(x) = a_k x + c_k, \qquad \text{2-moment} \tag{9a}$$

$$n(x) = c_k \qquad \text{1-moment.} \tag{9b}$$

where the Equation (9a) formulas for $a_k$ and $c_k$ can be derived from equations 15 through 20 in Wang et al. (2007). Each bin therefore has four separate parameters that govern the subgrid linear distribution function: $a_k$, $c_k$, $x_{k,1}$, and $x_{k,2}$. Note that when `moments=1`, a subgrid uniform distribution across each bin is used such that these parameters are diagnosed as: $x_{k,1} = x_k$, $x_{k,2} = x_{k+1}$, $a_k = 0$, and $c_k = 2M_k/(x_{k+1}^2 - x_k^2)$ and $N_k = M_k/\overline{x}_k$, where $\overline{x}_k$ is the bin midpoint mass.

### 2.2 BinMod1D class architecture and workflow

Comprehensive documentation and step-by-step tutorials that cover BinMod1D model initialization, parameter specification, customization, and more are provided in the software documentation (Dunnavan, 2026b). This section provides only a technical overview of the BinMod1D methodology and classes.

Regardless of how the model is run, the general workflow is the same. Users first initialize a `spectral_1d` object with a set of input parameters that govern the initial distribution, habit parameters, processes included, and model mode. Each `spectral_1d` object represents a separate model instance of BinMod1D where the object performs several tasks:

1. It uses the user's prescribed number of distributions and their parameters and initializes an `Interaction()` object which is used to determine the bin pair interactions when `run()` is called. If `moments=1` is specified, then bin-pair interactions for collision-coalescence is precalculated whereas `moments=2` bin interaction calculations are computed during model runtime.

2. It runs the BinMod1D model with the `run()` method which runs the model as a box model, a steady-state model, or a full 1D (time/height) model with either one or two predicted moments for each bin. The run mode selection depends upon the prescribed inputs for parameters $t_{\text{max}}$, $z_{\text{bot}}$ and $z_{\text{top}}$. If $t_{\text{max}} > 0$ and $z_{\text{bot}} = z_{\text{top}}$ then the box model is run, if $t_{\text{max}} = 0$ and $z_{\text{bot}} < z_{\text{top}}$ then the steady-state model is run, and if $t_{\text{max}} > 0$ and $z_{\text{bot}} < z_{\text{top}}$ then the full 1D model is run.

3. It stores each distribution's $M_k$ or ($M_k$ and $N_k$) value in a 4D array where the array is indexed by distribution, height, bins, and output time.

4. It plots the model results using methods that start with `plot_` (e.g., `spectral_1d.plot_dists()`).

5. It writes model output after the `run()` method is called to a netcdf file with the `write_netcdf()` method.

Users can initialize the particle size distribution(s) using one of six different methods using the `spectral_1d()` input `init_method`:

1. `init_method = 'gamma'`: The initial distribution is a gamma distribution with parameters specified by the user as additional arguments in `spectral_1d()`.

2. `init_method = 'analytical'`:

   The initial distribution is defined using a custom Python lambda function.

3. `init_method = 'empirical'`:

   The initial distribution is prescribed using edges and number distribution function value evaluated at the prescribed bin midpoints.

4. `init_method = 'empirical_counts'`:

   As in `init_method = 'empirical'` but where users prescribe bin counts rather than number distribution function values.

5. `init_method = 'direct'`:

   The user directly prescribes the bin masses ($M_{\mathrm{bins}}$) and/or bin numbers ($N_{\mathrm{bins}}$) along with bin edges.

6. `empty`:

   The initial distribution is initialized with no bin mass or number amounts.

When `spectral_1d()` is first initialized, an initial distribution using the `dist()` class is binned according to the user's input parameter specifications. If users use the input `init_method = 'gamma'` (default), then the initial binned distribution is calculated from a gamma distribution of the form

$$n_0(D) = 1000 \frac{N_{t,0}}{\Gamma(\mu_0+1)}(\mu_0+4)^{\mu_0+1}\frac{1}{D_{m,0}}\left(\frac{D}{D_{m,0}}\right)^{\mu_0}\exp\left[-(\mu_0+4)\frac{D}{D_{m,0}}\right] \tag{10}$$

where $n_0(D)$ has units of $1/\mathrm{m}^3\,1/\mathrm{mm}$ with a full initial number concentration $N_{t,0}$ in $L^{-1}$, full mean volume diameter $D_m$ in mm, and shape parameter $\mu$. Note the prescribed $N_{t,0}$ and $D_{m,0}$ values represent values that are valid for the full distribution from $(0,\infty)$ whereas the total binned values for $N_t$ and $D_m$ will depend upon the bin resolution $s_{\mathrm{bin}}$ and the minimum and maximum bin masses (sizes). Users who instead prefer the gamma distribution form

$$n_0(D) = N_0 D^{\mu_0}\exp(-\Lambda_0 D) \tag{11}$$

can diagnose the `spectral_1d()` parameters with the equations

$$\mu_0 = \mu_0 \tag{12a}$$

$$D_{m,0} = \frac{\mu_0+4}{\Lambda_0} \tag{12b}$$

$$N_{t,0} = N_0\frac{\Gamma(\mu_0+1)}{\Lambda^{\mu_0+1}}, \tag{12c}$$

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

where $\Lambda_0$ has units of $\mathrm{mm}^{-1}$ and $N_0$ has units of $\mathrm{m}^{-3}\,\mathrm{mm}^{-(\mu_0+1)}$. The `gam_norm` parameter determines if instead the initial gamma distribution is normalized to have a total mass concentration of unity. Thus, when `gam_norm=True` the form of the initial gamma distribution is instead with respect to mass $m$ and normalized as

$$\tilde{n}_0(m) = \frac{N_{t,0}}{\Gamma(\nu_0)}\frac{1}{\overline{m}}\nu_0^{\nu_0}\left(\frac{m}{\overline{m}}\right)^{\nu_0-1}\exp\left[-\nu_0\frac{m}{\overline{m}}\right], \tag{13}$$

where $\nu_0 = \mu_0+1$. Equation (10) is generally used in practice but Equation (13) is often used when comparing numerical results to known analytical solutions of the stochastic collection equation (see Section 3.1 and the BM1D_analytical_examples.ipynb Jupyter notebook for example). When Equation (13) is used for `gam_norm=True`, users will specify the $N_{t,0}$, $M_{t,0}$, $\mu_0$, and $x_0$ parameters with `dist_var="mass"` (where $\overline{m} \equiv M_{t,0}/N_{t,0}$) whereas when Equation (10) is used for `gam_norm=False` users will specify $N_{t,0}$, $D_{m,0}$, $\mu_0$, $D_1$ parameters along with `dist_var="size"`. Alternatively, users can directly specify $\overline{m}$ with the `mbar0` input parameter. The $D_1$ ($x_0$) parameter determines the minimum left hand bin edge size (mass) used for the first bin when specifying `dist_var="size"` (`dist_var="mass"`).

Users can specify the number and type of distributions used in `specral_1d()` by adding subsequent distributions to the list in the `habit_params` input parameter (e.g., `habit_params=["snow","fragments"]`). The length of `habit_params` determines the number of separate distributions that are solved where the names refer to the particle microphysical parameters in `habits.py` that are used for each distribution. Users need to explicitly set the `cc_dest` and `br_dest` input parameters when more than one distribution is specified. `cc_dest` and `br_dest` determine the distribution destination of coalesced and breakup bin-pair interactions, respectively where integer values are given from `1` to `len(habit_params)`. Collision ($E_{\text{col}}$), coalescence ($E_s$) and breakup ($E_b$) efficiencies are treated as constants that can be passed into `spectral_1d()` (e.g., `spectral_1d(Ecol=1.0,Es=0.5,Eb=0.5)`). Here, the total collision-coalescence efficiency is given by $E_{CC} = E_{\text{col}}E_s$ whereas the total breakup efficiency $E_{\text{BR}}$ is determined by: $E_{\text{BR}} = E_{\text{col}}(1-E_s)E_b$. In this way, $E_b$ represents the fraction of collisions that occur that do not coalesce but do break up. Therefore, $E_b = 1.0$ indicates the impossibility of rebounding whereas $0 < E_b < 1$ indicates a combination of breakup and rebounding.

Finally, users can also programatically save their `spectral_1d` model runs using the `write_netcdf()` method. This method saves the 4D mass and number bins for all distributions, heights, and times as well as model attributes. Users can also load back these model runs by initializing `spectral_1d` with the `load` input parameter.

## 2.3 Radar forward operator

BinMod1D v1.0.10 calculates radar reflectivity ($Z$), differential reflectivity ($Z_{\text{DR}}$), specific differential phase ($K_{\text{dp}}$), and correlation coefficient ($\rho_{\text{hv}}$). Radar calculations for each bin as well as for the entire distribution are calculated when `radar=True` for a `spectral_1d()` object or when the `calc_radar()` method is called. These polarimetric radar calculations follow the Rayleigh formulas from Ryzhkov et al. (2011) for rain and dry snow modeled as homogeneous soft spheroids. Scattering amplitudes are calculated at each bin edge with the formula (cf. equation 22 from Ryzhkov et al., 2011)

$$f_i^{a,b} = \frac{\pi^2 D_i^3}{6\lambda^2}\frac{1}{L_i^{a,b} + \frac{1}{\epsilon_i - 1}}, \tag{14}$$

where $\lambda$ is the radar wavelength (defaulted to S band with $\lambda = 110\,\mathrm{mm}$), $D_i$ is the equivolume diameter at bin edge $i$, $L_i^{a,b}$ is the spheroidal shape factor at bin edge $i$ determined by the aspect ratio at $i$, and $\epsilon_i$ is the dieletric constant for bin edge $i$. Indices $a$ and $b$ refer to the radar scattering along the particle's major ($a$) and minor ($b$) axes, respectively. Angular moments for a two-dimensional axisymetric Gaussian orientation distribution ($A_1$ through $A_7$) are used along with these scattering amplitudes to calculate radar contributions for each bin edge with the formulas

$$g_i^{Z_h} = \frac{4\lambda^4}{\pi^4|K_w|^2}\left\{|f_i^a|^2 - 2A_2\mathrm{Re}\left[f_i^{a*}\left(f_i^a - f_i^b\right)\right] + A_4\left|f_i^a - f_i^b\right|^2\right\} \tag{15a}$$

$$g_i^{Z_v} = \frac{4\lambda^4}{\pi^4|K_w|^2}\left\{|f_i^a|^2 + 2A_1\mathrm{Re}\left[f_i^{a*}\left(f_i^a - f_i^b\right)\right] + A_3\left|f_i^a - f_i^b\right|^2\right\} \tag{15b}$$

$$g_i^{K_{\mathrm{dp}}} = \frac{0.18\lambda}{\pi}A_7\mathrm{Re}\left(f_i^a - f_i^b\right) \tag{15c}$$

$$g_i^{Z_{hhvv}} = \frac{4\lambda^4}{\pi^4|K_w|^2}\left[|f_i^a|^2 + A_5\left|f_i^a - f_i^b\right|^2 - A_1 f_i^{a*}\left(f_i^a - f_i^b\right) - A_2 f_i^a\left(f_i^{a*} - f_i^{b*}\right)\right], \tag{15d}$$

where $*$ denotes a complex conjugate and $|K_w|^2 = 0.93$ is a dielectric constant for liquid water. Here, an assumed, fixed standard deviation of angles ($\sigma$) is used to calculate each angular moment according to the equations in Ryzhkov et al. (2011). Each habit therefore can be assigned a separate $\sigma$ value in its `habit_params` dictionary. The left and right bin edge values of each $g$ parameter are then used to determine a linear approximation from $g_i$ to $g_{i+1}$ in the form: $g_i'(m) = a_g m + c_g$, where

$$a_g^V = \frac{g_{i+1}^V - g_i^V}{\Delta x_i} \tag{16a}$$

$$c_g^V = g_i^V - a_g^V x_i. \tag{16b}$$

The radar variables $V$ for each bin are then analytically integrated from $x_i$ to $x_i$

$$V_i \approx \int_{x_i}^{x_{i+1}} g_i'(m)n_i(m)dm = a_g^V M_{i,1} + c_g^V M_{i,0} = a_g^V M_i + c_g^V N_i, \tag{17}$$

where $M_i$ is the total bin mass and $N_i$ is the total bin number.

## 3 Example Results

Most of the results presented in this section are derived from Jupyter notebook examples: “BM1D_analytical_examples.ipynb”, “BM1D_steady_state_examples.ipynb”, and “BM1D_full_1D_examples.ipynb”, which are provided in the BinMod1D repository (Dunnavan, 2026a). Users can execute these examples locally or within a pre-configured Binder environment which can be accessed through the project's documentation interface in Dunnavan (2026b).

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

### 3.1 Box model examples

Several previous studies (see Tzivion et al., 1987; Bott, 1998; Wang et al., 2007, for example) have leveraged the analytical solutions of the stochastic collection equation provided by Scott (1968) using the Golovin (cf. Golovin, 1963; Safronov, 1963) or sum (i.e., $K(x,y) \propto x+y$), product (i.e., $K(x,y) \propto xy$), and constant (i.e., $K(x,y) =$ const.) kernels to validate their spectral models. Similarly, the BinMod1D model in this section is compared to these analytical solutions for $\mu_0 = 3$ and for the normalized gamma distribution (Equation (13)) where $\overline{m}$ is unity as in Scott (1968).

These comparisons are shown in Figure 4 using a high-resolution mass grid ($s_{\mathrm{bin}} = 32$) in order to highlight the accuracy of the model. As in previous studies, these distributions are plotted with a log scaling[3]. Plotting in this way using the variable $\log(m)$ as the independent variable produces attractive bell-like plots that permit an easier way to compare these numerical solutions to the analytical solutions from Scott (1968). All plots here use the default `moments=2` (i.e., two-moment solution) `spectral_1d()` option. The Golovin solution shows a rapid decrease in number at $t = 20$ min near the initial gamma distribution mode and the emergence of a long tail which shifts the mass distribution mode to 10 times the initial gamma mode. The product solution at $t = 10$ min, on the other hand, maintains the mass distribution mode at the initial $\overline{m}$ value but produces a long tail toward much larger masses[4]. The constant kernel shows a much more gradual transition and shifting of the number and mass distribution modal values. Overall, the BinMod1D solutions are indistinguishable from the analytical solutions.

---

[3] This is the default plotting method for `spectral_1d.plot_dists()` as follows: $n(m)dm = mn(m)d\log(m)$ and $n_M(m)dm \equiv mn(m)dm = m^2 n(m)d\log(m)$.

[4] It is worth pointing out here that the product solution is known to exhibit gellation (see Alfonso et al., 2008, for more information) and, as such, does not actually preserve mass after a certain period or time. Therefore, the shorter time here is used to avoid the point of gellation.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

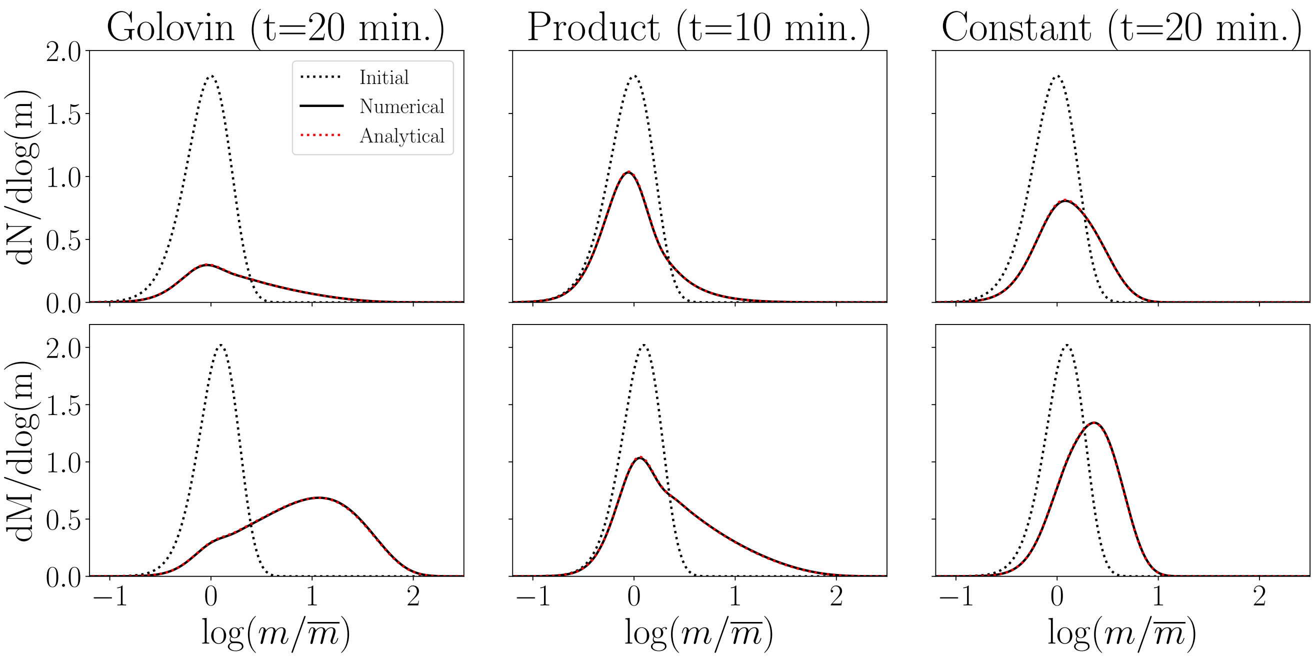

**Figure 4.** BinMod1D Jupyter notebook examples for the number (top row) and mass (bottom row) distribution functions with the Golovin, product, and constant kernel ($s_{\text{bin}} = 32$) with analytical solutions from Scott (1968) for $\mu_0 = 3$ and $\Delta t = 1\,\text{s}$. Here, particle masses and mass distributions are normalized according to Scott (1968) such that the initial total number and total mass are both unity. The `spectral_1d()` option `moments=2` is used for all model runs.

Figure 5 shows examples of collisional breakup only and the steady-state balance of coalescence with breakup along with the analytical solutions from Feingold et al. (1988). In these cases, the mass distributions very closely match the analytical solutions. There are somewhat larger differences in the number distributions. However, it is important to realize that the validity of the Feingold et al. (1988) analytical solutions depend upon several assumptions. In particular, the Feingold solutions do not exactly preserve explicit mass conservation for each breakup event although it does conserve the overall total distribution mass. BinMod1D, by contrast, explicitly calculates each bin-pair interaction such that mass is always conserved for each interaction event *regardless of the fragment distribution*. In this way, BinMod1D represents a much more physically realistic way to explicitly simulate collisional breakup compared to the target-based method-of-moments models such as Feingold et al. (1988).

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

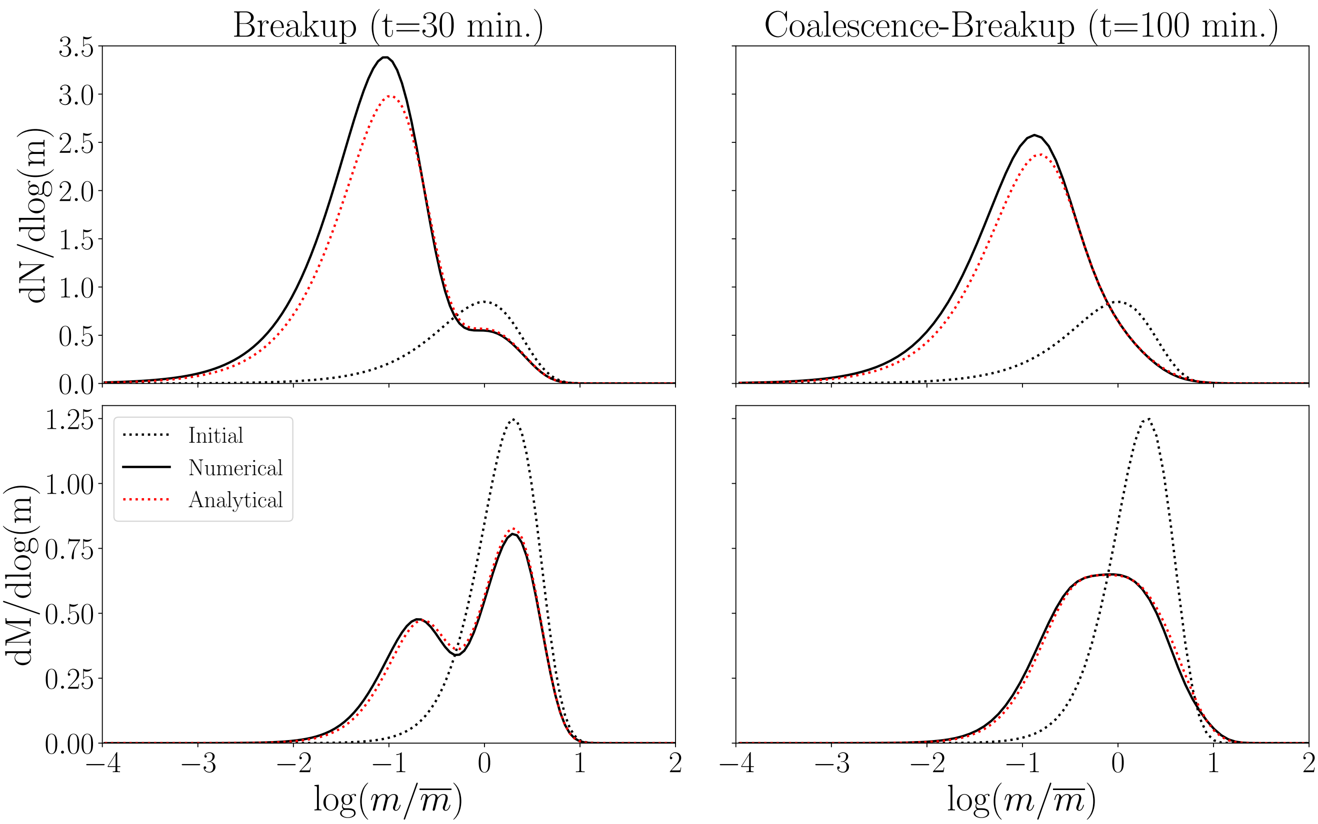

**Figure 5.** As in Figure 4 but for the breakup ($s = 6$, $E_s = 1.0$, and $E_b = 0.1$) and coalescence-breakup ($s = 4$, $E_s = 0.8$, $E_b = 1.0$) constant kernel examples and analytical solutions from Feingold et al. (1988) assuming $\mu_0 = 0$ and $\Lambda_f = 10\,g^{-1}$. The collision-coalescence and collisional breakup analytical solution is a steady-state solution.

### 3.2 Steady-state examples

Figure 6 shows two examples of the steady-state (i.e., height only) BinMod1D simulation for a rain case using the `plot_moments_radar()` and `plot_dists_height()` methods of the `spectral_1d` class. For these simulations, the initial distribution is given by $D_{m,0} = 0.8$ mm, $N_{t,0} = 15\,\mathrm{L}^{-1}$, and $\mu_0 = 0$. A 3-km layer is considered here with `ztop=3000.`, `zbot=0.`, and $\Delta z = 10$ m (`dz=10.`). For collision-coalescence only ($E_s = 0.8$), number concentration and liquid water content (LWC) decrease with decreasing height whereas mean volume diameters increase with decreasing height. Similarly, $Z$, $Z_{\mathrm{DR}}$, and $K_{\mathrm{dp}}$ all increase with decreasing heights as coalescence transfers mass from smaller sizes to larger sizes. This, in turn, increases raindrop sizes and decreases raindrop aspect ratios according to the Brandes et al. (2002) aspect ratio-diameter polynomial relationship. Figure 6 also shows the size distributions at 3.0 km, 1.5 km, and 0.0 km. As the size distributions evolve downward, the initial inverse-exponential distribution transitions to a more gamma-like shape.

Figure 6 also shows the same simulation but with the incorporation of breakup with $E_b = 0.005$ and an exponential fragment distribution (`frag_dist='exp'`). By default, the mean volume diameter of `frag_dist='exp'` is 0.25 mm. The combined coalescence-breakup simulation overall is quite similar to the coalescence only simulation, however there is a noticable stagnation effect that is most obvious for $D_m$, liquid water content (LWC) and the radar variables. This stagnation effect results from the presence of breakup stunting the coalescence growth effects. The size distribution at 1.5 km is very similar to the coalescence only simulation, especially in the tail portion of the distribution. However, the exponential fragment distribution is clearly visible and becomes more pronounced at $z = 0.0$ km.

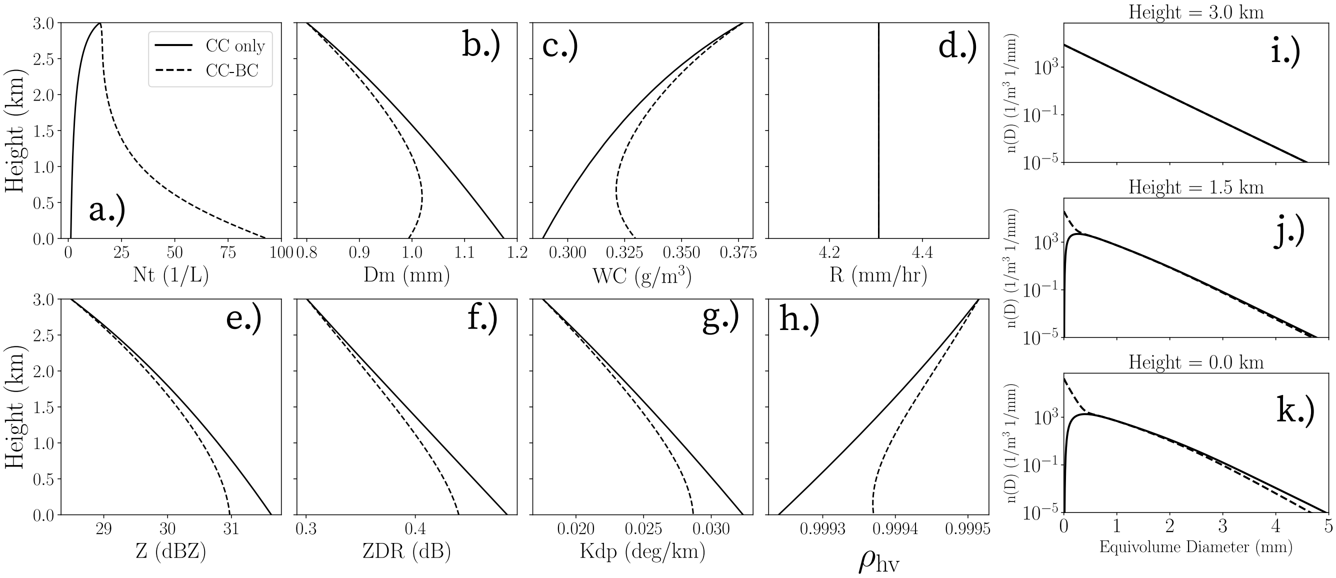

**Figure 6.** Various microphysical and radar variables for the steady-state rain example including: a.) Number concentration, b.) Mean volume diameter, c.) Liquid water content, d.) Rain rate, e.) Radar reflectivity, f.) Differential reflectivity, g.) Specific differential phase, and h.) Correlation coefficient. Solid lines represent the collision-coalescence only simulation (CC only) whereas dashed lines show the combined collision-coalescence/collisional-breakup (CC-BC) simulations. Subpanels i.) through k.) show the number distribution function at 3.0 km, 1.5 km, and 0.0 km, respectively.

Figure 7 shows a steady-state example of snow aggregation and collisional breakup. This scenario can be thought of as the effects of aggregation and breakup below the dendritic growth layer (DGL). Many studies have investigated these polarimetric radar signatures (Kennedy and Rutledge, 2011; Andrić et al., 2013; Schrom et al., 2015; Griffin et al., 2018; Dunnavan et al., 2022) and it is well known that reflectivity typically increases downward below whereas differential reflectivity and specific differential phase typically decreases downward. Here, BinMod1D is run with an initial gamma distribution with $N_{t,0} = 50\,\mathrm{L}^{-1}$, $D_{m,0} = 1.0$ mm, and $\mu_0 = 3$. Furthermore, two distinct category distributions are used: snow and fragments, where each sample distribution has microphysical parameters determined by the `habits.py` module (i.e., `habit_params=['snow','fragments']`). In particular, snow particles have a fixed aspect ratio of 0.6 with an orientation parameter of 0° (horizontal orientation) whereas fragments have aspect ratios of 0.8 and an orientation parameter of 20°. The simulation is also set up so that aggregated particles go to the snow distribution (`cc_dest=1`) whereas the breakup fragments go to the fragment distribution (`br_dest=2`) that is governed by an assumed lognormal distribution with a mode at $D_{\mathrm{mode}} = 0.2$ mm and a median of $D_{\mathrm{med}} = 0.25$ mm (i.e., `frag_dist='LGN'`). As shown in Figure 7, the microphysical variables can be shown independently for each category when running `spectral_1d` with multiple distribution categories. This simulation shows the domination of aggregation from 3.0 km to 2.0 km where reflectivity and $D_m$ rapidly increase downward. From 2.0 km to 1.0 km, breakup overwhelms aggregation which leads to a U-shaped reversal of the height profiles. Finally, 1.0 km to 0.0 km shows a balance of aggregation with collisional breakup such that the total distribution and its bulk variables more-or-less stagnate. It is important to note that this stagnation effect is typically ignored in models as collisional breakup is often not explicitly taken into account.

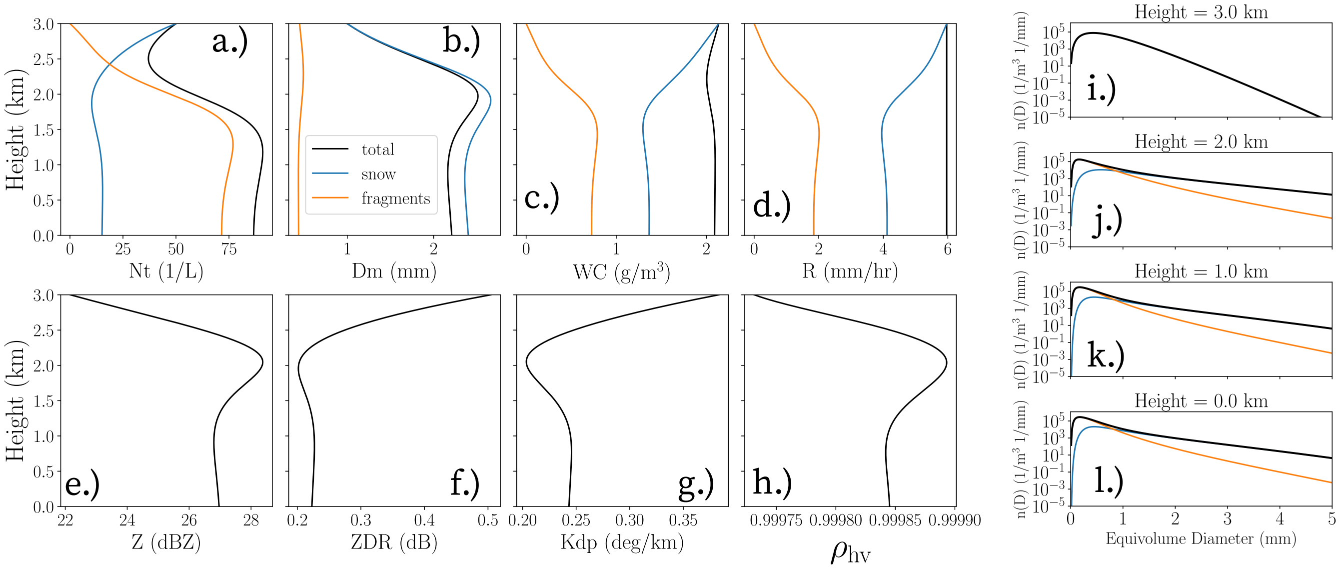

**Figure 7.** As in Figure 6 but for snow aggregation and collisional breakup as described in text. Here, $s_{\mathrm{bin}} = 3$ and $\Delta z = 5$ m.

### 3.3 Full model examples

Figure 8 shows an example of a rain "fallout" when `boundary=None`. In this case, the same initial rain distribution from Section 3.2 is used but with $E_{\mathrm{col}} = 1.0$, $E_s = 0.25$, $s_{\mathrm{bin}} = 2$, $\Delta t = 2$ s, $\Delta z = 20$ m, and 60 bins. With these parameters, this full simulation out to $t_{\mathrm{max}} = 1500$ s takes about 50 seconds on a laptop (see Section 4 for more information regarding the laptop specifications). The initial inverse-exponential distribution is only in the top height grid box initially. As time progresses, the initial distribution both sediments and engages in collision-coalescence. Size sorting of drops in different bins accounts for the gradual spreading of the reflectivity signature as the largest drops land before smaller drops.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

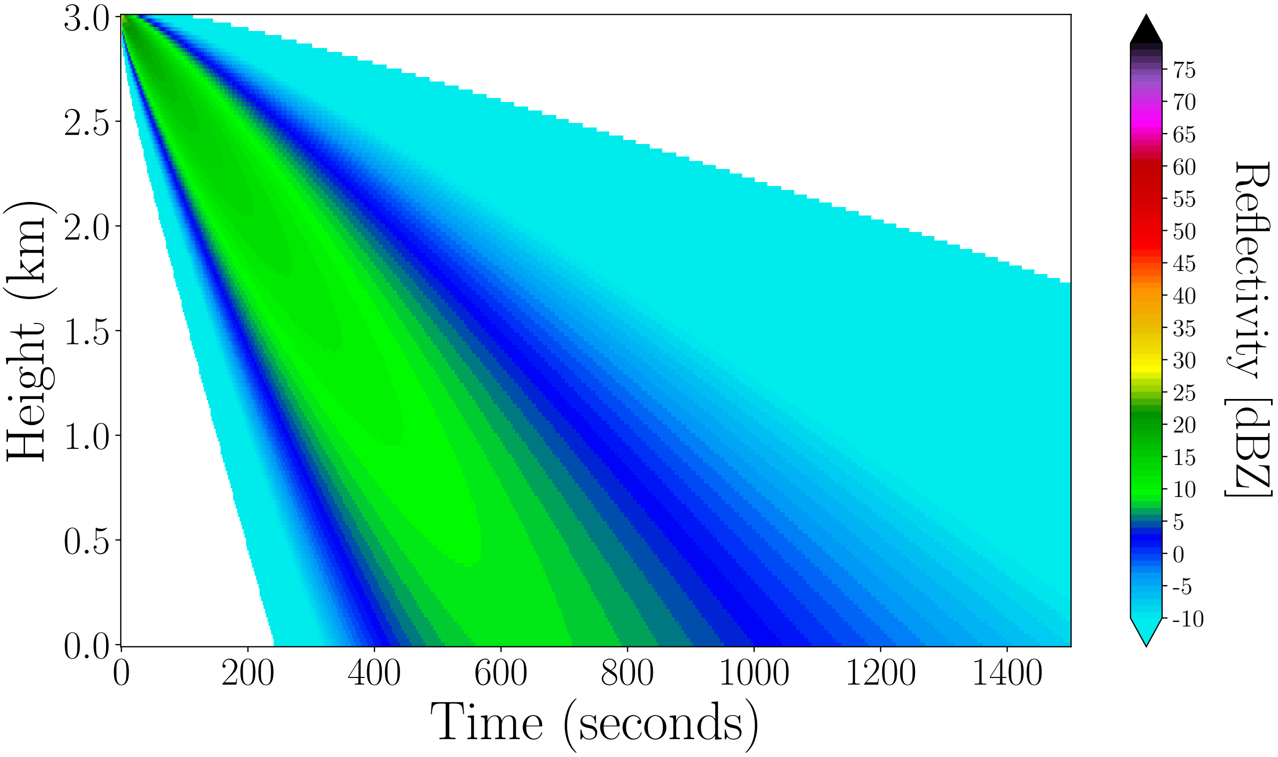

**Figure 8.** A time/height reflectivity profile (for a radar wavelength of 110 mm) for a rain fallout example where `boundary=None`.

Figure 9 shows a full 1D column model simulation for snow with aggregation ($E_s = 0.35$) representing the DGL simulation from Section 3.2 but with slightly larger initial snow particles ($D_{m,0} = 1.25$ mm). Unlike in the rain fallout example, here `boundary='fixed'` which indicates that the top particle size distribution (and therefore the topmost mass and number fluxes) is held constant. Therefore, this simulation will tend toward the steady-state solution as $t \rightarrow \infty$. As in the steady-state simulation, $s_{\text{bin}} = 2$ with 60 total bins. The model is run with only one category out to 5400 seconds with $\Delta t = 2$ s and $\Delta z = 20$ m and an output frequency of 1 (i.e., every $\Delta t$ timestep). The simulation itself takes just less than 5 minutes to run on the laptop used to conduct timing and convergence tests in Section 4. The radar time/height profiles in Figure 9 show the classic aggregation signature of an increasing reflectivity and decreasing $Z_{\text{DR}}$ and $K_{\text{dp}}$ toward ground. The system itself evolves slower than the rain case due to the slower fall speeds as it takes slightly less than an hour for the largest snowflakes to reach the surface. The $Z_{\text{DR}}$ time/height profile also shows the presence of size sorting as the larger, more oblate snowflakes reach the surface before the smaller, slower falling snowflakes. The simulation itself reaches a quasi steady-state after about 4000 seconds.

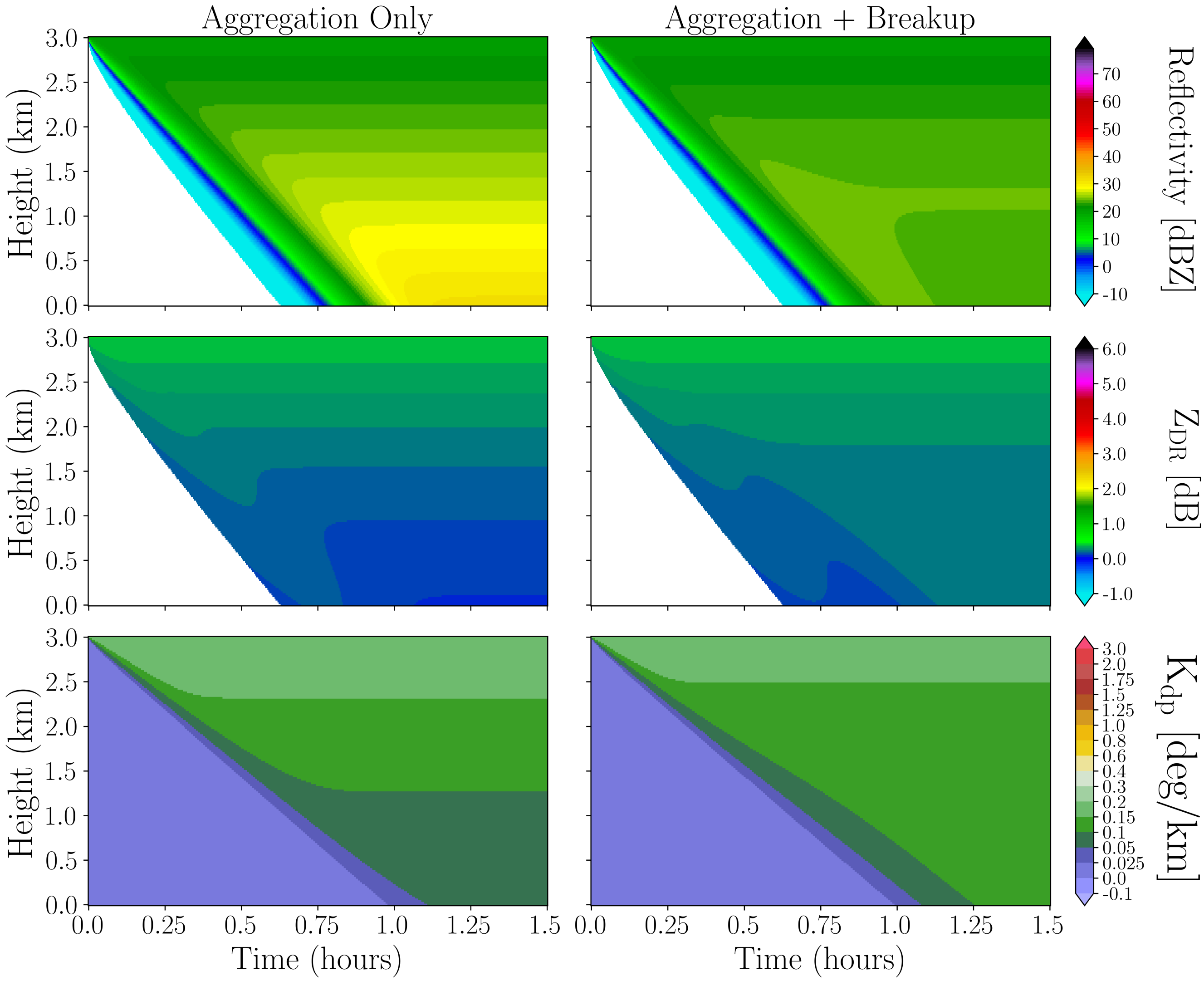

**Figure 9.** A snow aggregation example (left column) for reflectivity (top), differential reflectivity (middle), and specific differential phase (bottom) of BinMod1D where `boundary=fixed`. Right column represents the 2 category aggregation and breakup simulation.

Figure 10 shows the results of the previous DGL simulation but with the addition of collisional breakup. As in the steady-state simulation, the default lognormal fragment distribution is used. However, here $E_s = 0.6$ and $E_b = 0.05$ and the full simulation took 26 minutes to run. Figure 10 shows that the moments and radar variables for the full 1D column model simulation at 445 $t = 1.5$ hours is nearly identical to the corresponding steady-state simulation. The behavior of the profiles are also similar to Figure 7 but where reflectivity maximizes at 1.0 km rather than 2.0 km. The corresponding time/height profiles are shown alongside the aggregation only results in Figure 9. The simulations are very similar up until 2000 s where the presence of fragmentation begins to influence the radar fields. Although the start of the surface snowfall occurs at roughly the same time, the combined aggregation and breakup simulation produces a lower surface reflectivity than the aggregation only simulation. 450 The presence of fragments also leads to a higher $ZDR$ and $K_{\mathrm{dp}}$ which both stay very constant in the bottom 1.5 km of the model domain.

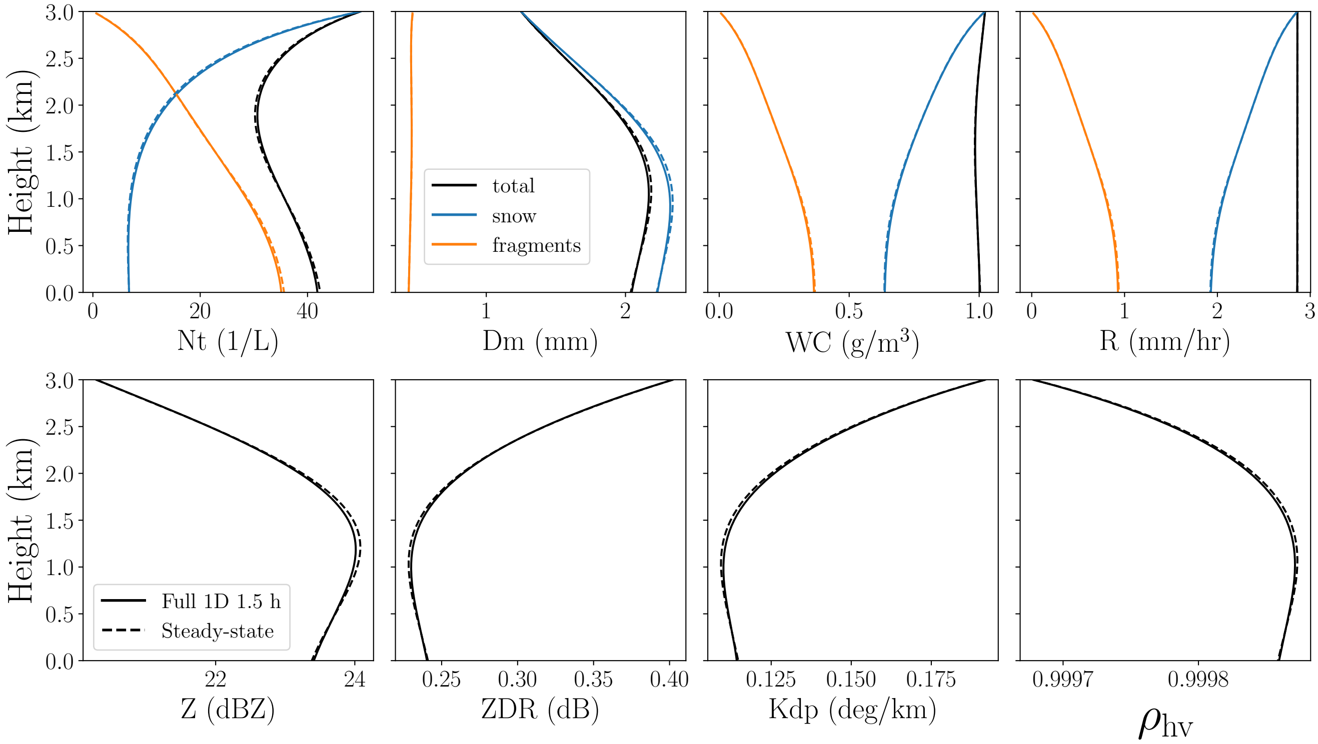

**Figure 10.** Moments and forward simulated radar variable comparison of a full time and height snow aggregation/breakup BinMod1D model run at $t = 1.5$ h compared to the corresponding steady-state BinMod1D model run.

## 4 Accuracy, computational speed, and resolution

Timing tests are performed on a workstation laptop equipped with a 12th generation Intel(R) Core i7-12850HX processor (2.1 GHz and 16 cores) and 64 GB of RAM running in a standard Python environment. These specifications represent standard 455 high-end consumer hardware, thereby demonstrating that the model can be effectively used without the need for specialized high-performance computing (HPC) resources.

These timing and convergence tests use the box model capability of BinMod1D to simulate a transition from continental cloud droplets to moderate rain with $Z \approx 40$ dBZ. This is similar to simulations performed by Simmel et al. (2002) and Wang et al. (2007). However, unlike those simulations, the BinMod1D simulations include both collision-coalescence and collisional 460 breakup. Here, the precipitation type is considered to be rain with one category and the Best (1950) fall speed parameterization is used rather than the default Atlas and Ulbrich (1977) power-law parameterization. The Long kernel (`kernel='Long'`) is used where the collision efficiency[5] is given by (cf. Long, 1974; Simmel et al., 2002)

$$E_{\text{col}} = E_{\text{col}}(x,y)\,[\text{mm}^3/\text{s}] = \begin{cases} \max\left[112.5\,D_y^2\left(1-\frac{0.006}{D_x}\right), 0.001\right] & D_y < 0.1\,\text{mm} \\ 1.0 & D_y \geq 0.1\,\text{mm} \end{cases} \tag{18}$$

and the hydrodynamic kernel is used such that

$$K_{\text{hydro}} = E_{\text{col}} E_s \frac{\pi}{4}\left(D_x + D_y\right)^2 \left|v_{t,x} - v_{t,y}\right|, \tag{19}$$

and the sticking efficiency $E_s$ is set to 0.6. The Straub et al. (2010) fragment distribution parameterization with $Eb = 1.0$ (i.e., no rebounding) is used as well (`frag_dist='Straub'`) which characterizes rain-rain fragments with a lognormal distribution, two Gaussian distributions, and a Dirac delta function for the residual rain drop. In order to prevent the breakup of cloud droplets, two linear ramp functions are used to scale $E_b$ based on both the maximum size of the (i,j) collection pair diameter bin midpoints (1.2 mm $<\max(D_i, D_j) <$ 2.8 mm ) and based on the size ratio $[q \equiv \min(D_i, D_j)/\max(D_i, D_j)]$) between the collection pair diameter bin midpoints ($0.1 < q < 0.25$). Pairs with values less than the first edge have $E_{\text{BR}}$ scaled by 0 (i.e., breakup is not considered) whereas pairs with values greater than the second edge values are scaled by unity (i.e., breakup is only scaled by $E_{\text{BR}}$ as it is by default). In addition to the Long kernel being a more accurate representation of the autoconversion process for cloud droplets, previous studies such as Wang et al. (2007) have also found that more realistic kernels like the Long kernel can have a slower numerical convergence for different $s_{bin}$ values than for less realistic kernels (i.e., Golovin, product, sum, etc.). Furthermore, these tests are, generally speaking, much more numerically demanding than what most users will typically experience. There are a number of reasons for this. First, the initial distribution consists of a very large number concentration ($\mathcal{O}(10^5\,L^{-1})$). This means that the number of binary interactions that occur at the beginning of the simulation is much greater than for simulations involving fully developed rain and snow. This also means that breakup requires adaptive stepping in order to ensure numerical stability.

In each test, a $s_{\text{bin}} = 16$ value with two-moment calculations is used as the reference solution to determine the accuracy of the BinMod1D model runs. The results shown herein are similar to the box model results from Simmel et al. (2002) and Wang et al. (2007) where the initial distribution is a gamma distribution with $\mu = 0$ (i.e., an exponential distribution) and normalized according to mass similar to Scott (1968) but with $\overline{D}_0 = 18.6\,\mu\text{m}$ and $M_{t,0} = 1\,\text{g}\,\text{m}^{-3}$ (parameters consistent with both Simmel et al. (2002) and Wang et al. (2007)). This leads to $N_{t,0} \approx 300,000\,\text{L}^{-1}$. In this sense, these convergence and timing tests are representative of continental cloud systems that transition from autoconversion of cloud droplets to accretion of cloud droplets by rain droplets and the resulting steady-state balance of coalescence and collisional breakup.

Figure 11 shows the timeseries of microphysical and radar variables for each simulation. Each simulation, regardless of resolution or moment prediction, produces the same qualitative behavior in the 3-hour period. Initially, autoconversion of cloud droplets is gradual and the distributions do not change much. For the reference solution $s_{\text{bin}} = 16$ and `moments=2`,

---

[5] The `kernel='Long'` option actually modifies the collision kernel $K(x,y)$ rather than $E_{\text{col}}$ to keep $E_{\text{col}}$ as a constant parameter. In these cases, $E_{\text{col}}$ can be set to unity.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

it is clear that it takes roughly 40 minutes for accretion to finally become dominant. However, the one-moment simulations struggles to capture this delay effect and instead accelerates the accretion process sooner. This is primarily because the collision-coalescence and collisional breakup processes are very sensitive to number concentration as the number of collisions increases rapidly with number concentration. The explicit prediction of number concentration with the two-moment simulations offsets this rapid acceleration and produces the correct transition point even with a somewhat coarse $s_{\mathrm{bin}} = 2$ value. The $s_{\mathrm{bin}} = 1$ with `moments=2` simulation actually over-delays the autoconversion-accretion process slightly. As shown in Table 1, the root-mean-square (RMSE) errors for BinMod1D one-moment number concentrations with $s_{\mathrm{bin}} = 16$ (1282.16 $\mathrm{L}^{-1}$) is actually more than twice as large as the BinMod1D two-moment number concentrations with $s_{\mathrm{bin}} = 2$. Despite this, the absolute errors of number concentration at 3 hours is overall quite low for all simulations where the maximum error is only 0.46 $\mathrm{L}^{-1}$ for $s_{\mathrm{bin}} = 1$ and `moments=1` for a reference value of 3.17 $\mathrm{L}^{-1}$ from the $s_{\mathrm{bin}} = 16$ and `moments=2` simulation. This behavior is consistent with the other microphysical variables as well. Even the coarse bin resolution $s_{\mathrm{bin}} = 1$ two-moment model run leads to only a slight overestimation of $N_t$, $D_m$, and $R$ by 0.22 $\mathrm{L}^{-1}$, 0.088 mm and 0.81 mm $\mathrm{h}^{-1}$, respectively. Similarly, $Z$, $Z_{\mathrm{DR}}$, and $K_{\mathrm{dp}}$ are within 1.17 dBZ, 0.19 dB, and 0.038 ° $\mathrm{km}^{-1}$ of the $s_{\mathrm{bin}} = 16$ reference solution, respectively.

The radar variables, particular the polarimetric ones, show more sensitivity. The diffusive nature of the one-moment solutions exacerbates the accelerated autoconversion-accretion transition shown in the reflectivity timeseries due to the $D^6$ weighting. As a result, the $s_{\mathrm{bin}} = 1$ with `moments=1` simulation almost immediately increases rapidly with reflectivity and differential reflectivity whereas specific differential phase has a slightly longer apparent transition period. Most notably the $s_{\mathrm{bin}} = 1$ with `moments=1` simulation dramatically overestimates the $Z_{\mathrm{DR}}$ by 3.36 dB and underestimates $\rho_{\mathrm{hv}}$ by 0.0732. These exaggerated values are due to the extreme diffusion at the large end of the particle mass distribution (Figure 12) which shows a near exponential behavior with size rather than the super-exponential behavior from collisional breakup that the more accurate, higher resolution simulations show. Increasing $s_{\mathrm{bin}}$ from 1 to 2 noticeably improves this diffusive behavior much closer to the reference solutions. Overall, even $s_{\mathrm{bin}} = 2$ with `moments=2` faithfully captures the correct timeseries of each variable and the correct bimodal steady-state number and mass distribution functions at 3 hours. Run times range from 13 seconds for the coarsest run to 5380 seconds for the reference simulation. However, even the second highest resolution run ($s_{\mathrm{bin}} = 8$ for `moments=2`) only takes slightly longer than 10 minutes to perform on a laptop computer. As stated before, these simulations are typically much more demanding than what most users will experience due to the large initial number concentration and the adaptive stepping needed to accurately simulate breakup for these large number of interactions.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

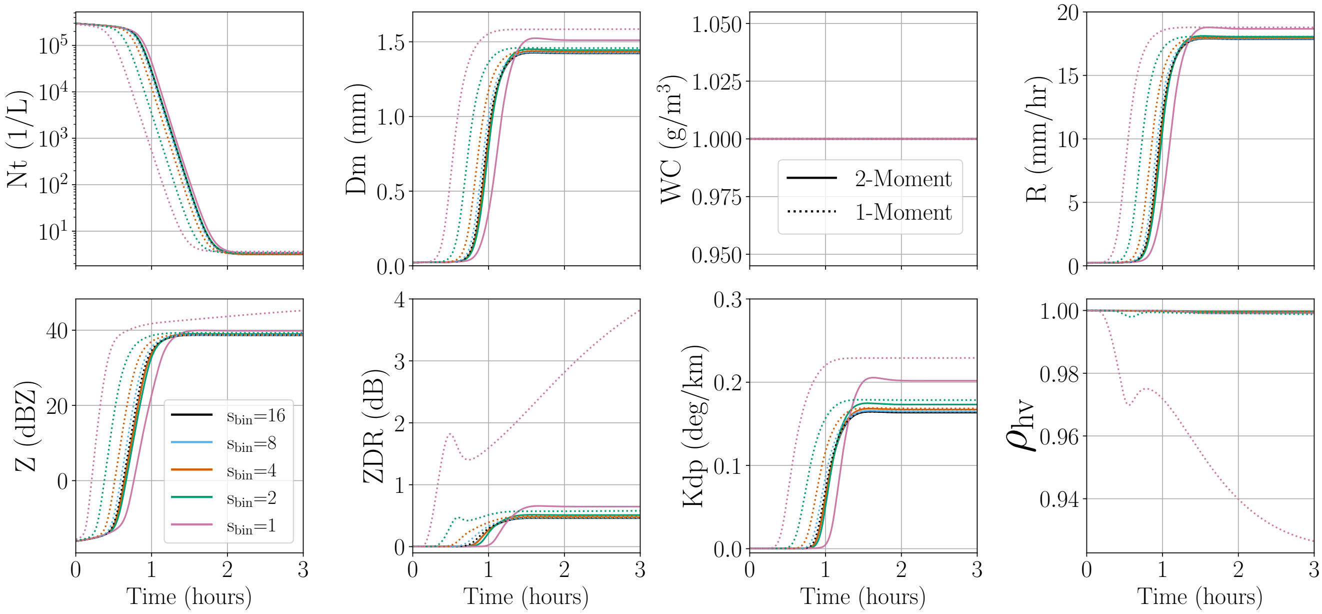

**Figure 11.** The timeseries evolution of number concentration, mean volume diameter, liquid water content, rain rates as well as radar reflectivity, differential reflectivity, specific differential phase, and correlation coefficient for cloud-rain convergence and testing simulations. Here, the radar wavelength is 110 mm which is consistent with S band radars. Each simulation uses $\Delta t = 1$ s and an output frequency of 10 seconds.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

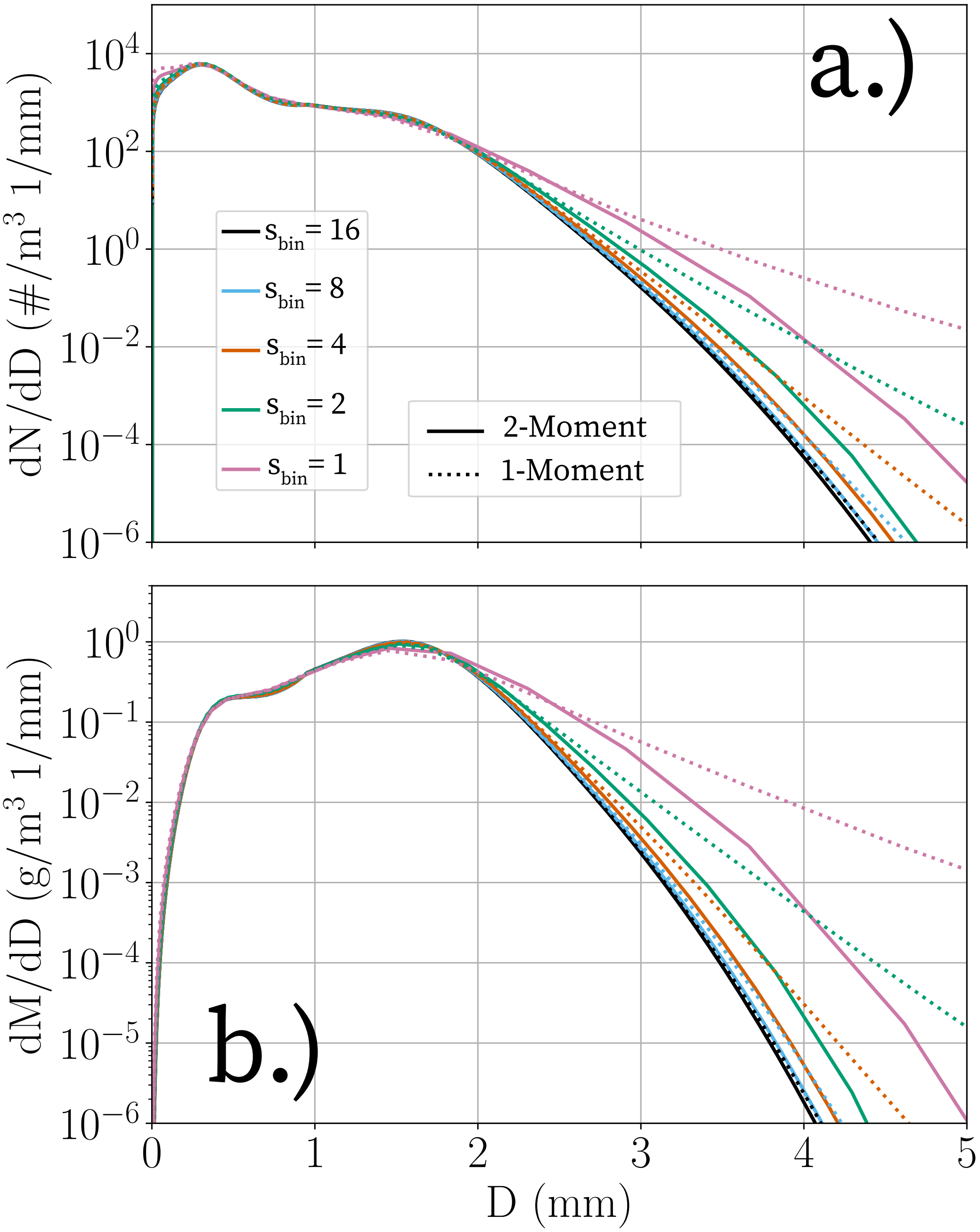

**Figure 12.** Number (a) and mass (b) distribution functions from the box model simulations at 3 hours for different $s_{\mathrm{bin}}$ parameters and for 2-moment simulations (solid lines) and 1-moment simulations (dotted lines).

**Table 1.** Timing and convergence tests of BinMod1D. For each variable, the top row represents RMSE and the bottom row represents the absolute error at 3 hours. Numbers in parentheses indicate 1-moment runs; those without represent 2-moment runs.

| Variable | Metric | $s_{\mathrm{bin}} = 16$ | $s_{\mathrm{bin}} = 8$ | $s_{\mathrm{bin}} = 4$ | $s_{\mathrm{bin}} = 2$ | $s_{\mathrm{bin}} = 1$ |
|---|---|---|---|---|---|---|
| $N_t$ (L$^{-1}$) | RMSE | — (1282.16) | 188.00 (4953.89) | 611.70 (15679.88) | 532.46 (38109.81) | 7711.81 (67253.75) |
| | Abs. Err | — (0.0036) | 0.0055 (0.018) | 0.030 (0.068) | 0.15 (0.25) | 0.22 (0.46) |
| $D_m$ (mm) | RMSE | — (0.0171) | 0.0026 (0.0497) | 0.0079 (0.1280) | 0.0234 (0.2710) | 0.1520 (0.4830) |
| | Abs. Err | — (0.00021) | 0.0032 (0.0039) | 0.0098 (0.013) | 0.022 (0.035) | 0.088 (0.16) |
| $R$ (mm h$^{-1}$) | RMSE | — (0.210) | 0.025 (0.613) | 0.074 (1.591) | 0.241 (3.332) | 1.785 (5.547) |
| | Abs. Err | — (-0.00075) | 0.028 (0.026) | 0.088 (0.081) | 0.19 (0.19) | 0.81 (0.93) |
| $Z$ (dBZ) | RMSE | — (0.775) | 0.038 (2.143) | 0.102 (5.291) | 0.954 (10.675) | 5.158 (18.079) |
| | Abs. Err | — (0.0059) | 0.039 (0.059) | 0.13 (0.20) | 0.33 (0.63) | 1.17 (6.57) |
| $Z_{\mathrm{DR}}$ (dB) | RMSE | — (0.0134) | 0.0035 (0.0337) | 0.0127 (0.0797) | 0.0414 (0.1950) | 0.1510 (2.0970) |
| | Abs. Err | — (0.0011) | 0.0046 (0.0085) | 0.016 (0.030) | 0.050 (0.12) | 0.19 (3.36) |
| $K_{\mathrm{dp}}$ (°/km) | RMSE | — (0.0025) | 0.0008 (0.0071) | 0.0028 (0.0179) | 0.0079 (0.0388) | 0.0338 (0.0902) |
| | Abs. Err | — (0.00010) | 0.0011 (0.0014) | 0.0036 (0.0049) | 0.0097 (0.015) | 0.038 (0.066) |
| $\rho_{\mathrm{hv}}$ ($\times 10^{-4}$) | RMSE | — (0.114) | 0.081 (0.346) | 0.290 (1.140) | 0.945 (7.240) | 3.850 (488.0) |
| | Abs. Err | — (-0.042) | -0.11 (-0.25) | -0.38 (-0.96) | -1.23 (-8.78) | -5.27 (-732.40) |
| **Run Time** | | 5380.3 (1219.8) | 705.1 (217.4) | 154.3 (78.5) | 54.9 (22.0) | 31.5 (13.0) |

## 5 Conclusions

BinMod1D is a simple, economic spectral bin microphysics model that should be accessible for those familiar with the Python programming language. This article provides only a small set of examples for using the model. However, due to its flexibility, BinMod1D can be customized for a variety of different purposes. For example, users could use BinMod1D to perform in-situ-type bin model simulations similar to Field et al. (2006) where actual in-situ-derived particle size distributions are used to initialize the bin model and investigate the theoretical effects of aggregation (and/or breakup) compared to measured distributions taken from such Lagrangian aircraft spirals. Similarly, users could also develop theoretical particle size distributions valid for laboratory studies such as done in Connolly et al. (2012).

Figure 13 presents another possible application: retrieval methodology validation. In this example, BinMod1D is run in the steady-state mode to produce synthetic snow size distributions in height. Here, $s_{\mathrm{bin}} = 6$ with 180 bins in order to produce a high resolution set of PSDs. This simulation uses an initial exponential distribution with parameters $N_{t,0} = 50$ L$^{-1}$ and $D_{m,0} = 1.25$ mm which produces an initial reflectivity of about 20 dBZ. $E_s = 0.35$ is used to simulate the explicit steady-state PSD evolution in height due only to aggregation. This increases the reflectivity to approximately 30 dBZ at the surface. The $Z/K_{\mathrm{dp}}$ Bukovčić et al. (2020) (B20) and the $Z/Z_{\mathrm{DR}}/K_{\mathrm{DP}}$ Ryzhkov and Zrnić (2019) (RZ19) power-law retrievals are then used to retrieve $N_t$, $D_m$, IWC, and $R$ *solely from the BinMod1D polarimetric radar height signatures* (see Table 1 from Dunnavan et al., 2022, for these retrieval equations). In this way, these comparisons shows the limitations of these retrieval methods due to using a fixed distribution shape. The BinMod1D simulation and Ryzhkov and Zrnić (2019) retrievals assumes the same snow parameters as used in Bukovčić et al. (2020) to derive their retrievals where the snow aspect ratios are $0.65$, $\sigma = 0°$, the density-size power-law relation is $\rho(D) = 0.178D^{-1}$, and the fall speed-size relation is $v_t(D) = 0.81D^{0.15}$. Both sets of retrievals assume $\mu = 0$ for all heights. Initially, the retrieved size distribution parameters (Figure 13a,b) are within 5 $\mathrm{L}^{-1}$ and $0.05$ mm of the initial BinMod1D exponential distribution $N_t$ and $D_m$, respectively. The initial differences between the BinMod1D microphysical variables are due to the capping of the density at $0.92$ g $\mathrm{cm}^{-3}$ at small sizes which the analytical retrieval formulas do not consider (for the retrievals to be simple power-law formulas, $\rho(D \to 0) = \infty$). Both B20 and RZ19 demonstrate the same general behavior of $N_t(z)$ and $D_m(z)$ from the explicit BinMod1D steady-state simulation. However, both retrievals underestimate the increase in $D_m$ due to aggregation toward ground by nearly $1.0$ mm. Figure 13 panels i.) through k.) shows that this is due to the retrievals tendency to approximate the quasi-exponential behavior in the millimeter size range but underestimate the size distribution of large aggregates present in the centimeter range. These sparse but large aggregates that generate in the BinMod1D simulation overall heavily weights the $D_m$ calculation. Both B20 and RZ19 overestimate the snow rates particularly at the surface where both retrievals overestimate the size distribution in the mid-millimeter range (Figure 13k). Figure 13c show that B20 retrieves a near constant IWC profile whereas RZ19 retrieves an IWC profiles that increases toward ground unlike the slightly decreasing IWC profile from BinMod1D. Overall, while the retrievals exhibit the correct overall behavior, these retrievals are still fundamentally limited by their inability to account for a changing distribution shape due to aggregation. This is one example where the reference solution provided by BinMod1D could be used to improve the retrieval equations of B20, RZ19, and others. Similarly, BinMod1D reference solutions can also be used to train retrieval-based artificial intelligence models.

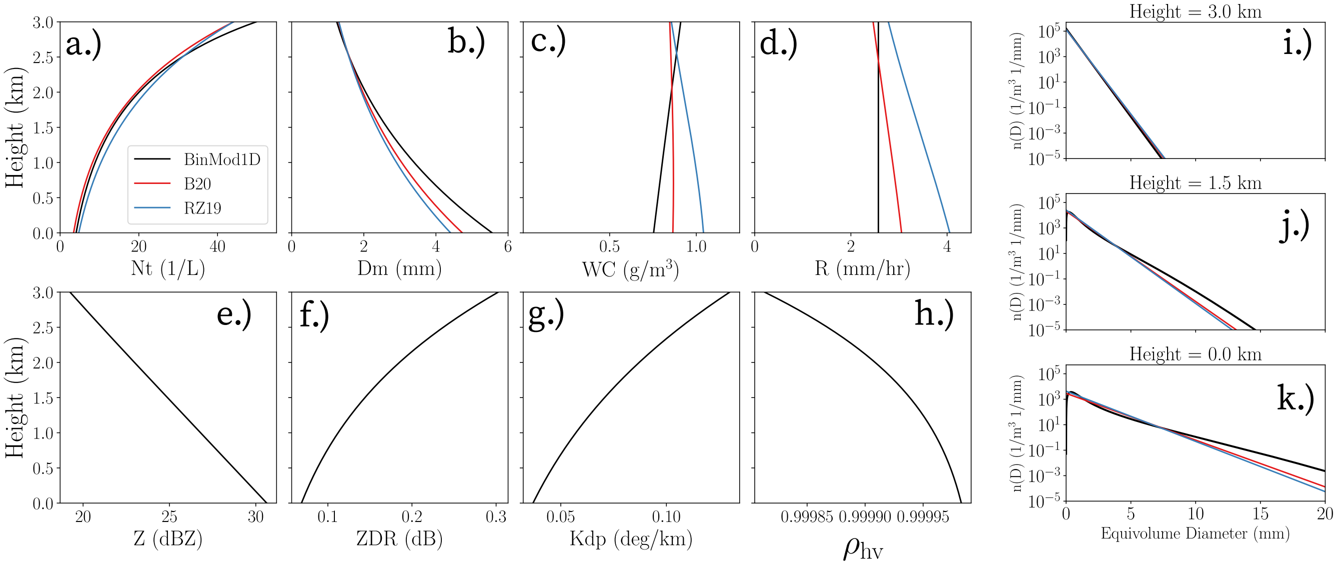

**Figure 13.** BinMod1D (black lines) steady-state results for a heavy snow case. Panels a.) through d.) and i.) through k.) also show corresponding retrievals using the Bukovčić et al. (2020) (B20; red lines) and Ryzhkov and Zrnić (2019) (RZ19; blue lines) power law relations.

There are a number of planned future improvements to the BinMod1D package including but not limited to

1. T-matrix radar forward simulation calculations.
2. Additional microphysical processes such as riming, melting, vapor deposition, condensation, evaporation, sublimation, and refreezing.
3. Incorporating non-constant coalescence and breakup efficiency parameterizations.
4. Incorporation of mixed-phase species (i.e., melting snow).
5. Incorporation of additional detailed collision-coalescence parameterizations including those of Phillips et al. (2015, 2017) for snow aggregation and collisional breakup.
6. The ability to specify a user-prescribed time-dependent top boundary condition for the `boundary` input parameter or an initial set of distributions in height.
7. Variable order Weighted Essentially Non-Oscillatory (WENO) advection calculations.
8. Ability to run the model using GPUs via the CuPy package (Okuta et al., 2017).

*Code availability.* The current version of BinMod1D is available at the project website (https://github.com/NOAA-National-Severe-Storms-Laboratory/BinMod1D) and documented at https://binmod1d.readthedocs.io/ which is archived on

Zenodo at https://doi.org/10.5281/zenodo.19890222 (Dunnavan, 2026b). The exact version of the model (v1.0.10) used to produce Figures 4 through 13 presented in this paper is archived on Zenodo at https://doi.org/10.5281/zenodo.19889377 (Dunnavan, 2026a). Readers can reproduce these figures by running the `reproduce_paper.py` Python script that is available in the scripts directory. The BinMod1D package can be installed via PIP (`pip install binmod1d`), and a Conda Forge distribution is currently under review. For the most up-to-date installation instructions, please visit https://binmod1d.readthedocs.io/en/latest/installation.html. BinMod1D is distributed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0) license, supplemented by specific redistribution and attribution conditions required by the Board of Regents of the University of Oklahoma.

*Author contributions.* ELD conceived of the study, created the BinMod1D Python package, developed the Jupyter notebooks, wrote the manuscript and produced the figures in the paper.

*Competing interests.* The author declares no competing interests in this work.

*Acknowledgements.* The author would like to thank Ted Mansell (NSSL) and Jacob Carlin (NSSL) for their suggestions that improved this paper. Funding for this work was provided by the NOAA/Office of Oceanic and Atmospheric Research under NOAA–University of Oklahoma Cooperative Agreement NA21OAR4320204, U.S. Department of Commerce. The author acknowledges the use of Google Gemini Pro for assistance with optimization and refinement of the manuscript's software architecture.

https://doi.org/10.5194/egusphere-2026-2004
Preprint. Discussion started: 4 May 2026

## References

Alfonso, L., Raga, G. B., and Baumgardner, D.: The validity of the kinetic collection equation revisited, Atmos. Chem. Phys., 8, 969–982, https://doi.org/10.5194/acp-8-969-2008, 2008.

Andrić, J., Kumjian, M. R., Zrnić, D. S., Straka, J. M., and Melnikov, V. M.: Polarimetric signatures above the melting layer in winter storms: An observational and modeling study, J. Appl. Meteor. Climatol., 52, 682–700, https://doi.org/10.1175/JAMC-D-12-028.1, 2013.

Atlas, D. and Ulbrich, C. W.: Path- and Area-integrated rainfall measurement by microwave attenuation in the 1-3 cm band, J. Appl. Meteor. Climatol., 16, 1322–1331, https://doi.org/10.1175/1520-0450(1977)016<1322:PAAIRM>2.0.CO;2, 1977.

Berry, E. X. and Reinhardt, R. L.: An analysis of cloud drop growth by coalescence: Part I. Double distributions , J. Atmos. Sci., 31, 1814–1824, https://doi.org/1520-0469(1974)031<1814:AAOCDG>2.0.CO;2, 1974.

Best, A. C.: Empirical formulae for the terminal velocity of water drops falling through the atmosphere, Quart. J. Roy. Meteor. Soc., 76, 302–311, https://doi.org/10.1002/qj.49707632905, 1950.

Bieli, M., Dunbar, O. R. A., de Jong, E. K., Jaruga, A., Schneider, T., and Bischoff, T.: An Efficient Bayesian Approach to Learning Droplet Collision Kernels: Proof of Concept Using “Cloudy”, a New $n$-Moment Bulk Microphysics Scheme, Journal of Advances in Modeling Earth Systems, 14, e2022MS002 994, https://doi.org/10.1029/2022MS002994, 2022.

Bleck, R.: A fast, approximative method for integrating the stochastic coalescence equation, J. Geophys. Res., 75, 5165–5171, https://doi.org/10.1029/JC075i027p05165, 1970.

Bott, A.: A flux method for the numerical solution of the stochastic collection equation, J. Atmos. Sci., 55, 2284–2293, https://doi.org/10.1175/1520-0469(1998)055<2284:AFMFTN>2.0.CO;2, 1998.

Bott, A.: A flux method for the numerical solution of the stochastic collection equation: Extension to two-dimensional particle distributions, J. Atmos. Sci., 57, 284–294, https://doi.org/10.1175/1520-0469(2000)057<0284:AFMFTN>2.0.CO;2, 2000.

Brandes, E. A., Zhang, G., and Vivekanandan, J.: Experiments in Rainfall Estimation with a Polarimetric Radar in a Subtropical Environment, J. Appl. Meteor., 41, 674–685, https://doi.org/10.1175/1520-0450(2002)041<0674:EIREWA>2.0.CO;2, 2002.

Brdar, S. and Seifert, A.: LCM1D: a 1D vertical column model for simulating cloud microphysics with a Lagrangian cloud model, Geoscientific Model Development, 13, 5119–5144, https://doi.org/10.5194/gmd-13-5119-2020, 2020.

Bukovčić, P. and Krause, J.: Path-CVP (pCVP) – polarimetric radar data snapshot along the predefined path based on Columnar Vertical Profiles, Atmos. Meas. Tech., 19, 775–791, https://doi.org/10.5194/amt-19-775-2026, 2026.

Bukovčić, P., Ryzhkov, A. V., and Zrnić, D.: Polarimetric relations for snow estimation – radar verification, J. Appl. Meteor. Climatol., 59, 1–52, https://doi.org/10.1175/JAMC-D-19-0140.1, 2020.

Chase, R. J., Nesbitt, S. W., and McFarquhar, G. M.: A dual-frequency radar retreival of two parameters of the snowfall particle size distribution using a neural network, J. Appl. Meteor. Climatol., 60, 341–359, https://doi.org/10.1175/JAMC-D-20-0177.1, 2021.

Connolly, P. J., Emersic, C., and Field, P. R.: A laboratory investigation into the aggregation efficiency of small ice crystals, Atmos. Chem. Phys., 12, 2055–2076, https://doi.org/10.5194/acp-12-2055-2012, 2012, 2012.

Drake, R. L.: The scalar transport equation of coalesence theory: Moments and kernels, J. Atmos. Sci., 29, 537–547, https://doi.org/10.1175/1520-0469(1972)029<0537:TSTEOC>2.0.CO;2, 1972.

Drake, R. L. and Wright, T. J.: The scalar transport equation of coalesence theory: New families of exact solutions, J. Atmos. Sci., 29, 548–556, https://doi.org/10.1175/1520-0469(1972)029<0548:TSTEOC>2.0.CO;2, 1972.

Dunavant, D. A.: High degree efficient symmetrical Gaussian quadrature rules for the triangle, International Journal for Methods in Engineering, 21, 1129–1148, https://doi.org/10.1002/nme.1620210612, 1985.

Dunnavan, E. L.: BinMod1D v1.0.10: A Python package for explicitly simulating 1D collisional coalescence/breakup processes with corresponding polarimetric radar signatures, https://doi.org/10.5281/zenodo.19889377, version 1.0.10, https://doi.org/10.5281/zenodo.19889377, 2026a.

Dunnavan, E. L.: BinMod1D (v1.0.10) Documentation, https://doi.org/10.5281/zenodo.19890222, 2026b.

Dunnavan, E. L., Carlin, J. T., Hu, J., Bukovčić, P., Ryzhkov, A. V., McFarquhar, G. M., Finlon, J. A., Matrosov, S. Y., and Delene, D. J.: Radar retrieval evaluation and investigation of dendritic growth layer polarimetric signatures in a winter storm, J. Appl. Meteor. Climatol., 61, 1685–1711, https://doi.org/10.1175/JAMC-D-21-0220.1, 2022.

Enukashvily, I. M.: A numerical method for integrating the kinetic equation of coalescence and breakup of cloud droplets, J. Atmos. Sci., 37, 2521–2534, https://doi.org/10.1175/1520-0469(1980)037<2521:ANMFIT>2.0.CO;2, 1980.

Feingold, G., Tzivion, S., and Levin, Z.: Evolution of raindrop spectra. Part I: Solution to the Stochastic Collection/Breakup equation using the method of moments, J. Atmos. Sci., 45, 3387–3399, https://doi.org/10.1175/1520-0469(1988)045<3387:EORSPI>2.0.CO;2, 1988.

Field, P. R., Heymsfield, A. J., and Bansemer, A.: A test of ice self-collection kernels using aircraft data, J. Atmos. Sci., 63, 651–666, https://doi.org/10.1175/JAS3653.1, 2006.

Golovin, A. M.: The solution of the coagulation equation for cloud droplets in a rising air current, Bull. Acad. Sci. SSSR Geophys. Ser. (English Transl.), pp. 482–487, http://mi.mathnet.ru/dan27630, 1963.

Griffin, E. M., Schuur, T. J., and Ryzhkov, A. V.: A polarimetric analysis of ice microphysical processes in snow, using quasi-vertical profiles, J. Appl. Meteor. Climatol., 57, 31–50, https://doi.org/10.1175/JAMC-D-17-0033.1, 2018.

Kennedy, P. C. and Rutledge, S. A.: S-band dual-polarization radar observations of winter storms, J. Appl. Meteor. Climatol., 50, 844–858, https://doi.org/10.1175/2010JAMC2558.1, 2011.

Kovetz, A. and Olund, B.: The effect of coalescence and condensation on rain formation in a cloud of finite vertical extent, J. Atmos. Sci., 26, 1060–1065, https://doi.org/10.1175/1520-0469(1969)026<1060:TEOCAC>2.0.CO;2, 1969.

Kumar, S. and Ramkrishna, D.: On the solution of population balance equations by discretization – I. A fixed pivot technique, Chem. Eng. Sci., 51, 1311–1332, https://doi.org/10.1016/0009-2509(96)88489-2, 1996.

Kumjian, M. R. and Prat, O. P.: The impact of raindrop collisional processes on the polarimetric radar variables, J. Atmos. Sci., 71, 3052–3067, https://doi.org/10.1175/JAS-D-13-0357.1, 2014.

Lam, S. K., Pitrou, A., and Seibert, S.: Numba: A LLVM-based Python JIT compiler, in: Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC, LLVM '15, pp. 1–6, Association for Computing Machinery, New York, NY, USA, https://doi.org/10.1145/2833157.2833162, 2015.

Long, A. B.: Solutions to the droplet collection equation for polynomial kernels, J. Atmos. Sci., 31, 1040–1051, https://doi.org/10.1175/1520-0469(1974)031<1040:STTDCE>2.0.CO;2, 1974.

Low, T. B. and List, R.: Collision, coalescence, and breakup of raindrops. Part I: Experimentally established coalescence efficiencies and fragment size distributions in breakup, J. Atmos. Sci., 39, 1591–1606, https://doi.org/10.1175/1520-0469(1982)039<1591:CCABOR>2.0.CO;2, 1982a.

Low, T. B. and List, R.: Collision, coalescence, and breakup of raindrops. Part II: Parameterization of fragment size distributions, J. Atmos. Sci., 39, 1607–1618, https://doi.org/10.1175/1520-0469(1982)039<1607:CCABOR>2.0.CO;2, 1982b.

McFarquhar, G. M.: A new representation of collision-induced breakup of raindrops and its implications for the shapes of raindrop size distributions, J. Atmos. Sci., 61, 777–794, https://doi.org/10.1175/1520-0469(2004)061<0777:ANROCB>2.0.CO;2, 2004.

McLeod, J. B.: On the Scalar Transport Equation, Proceedings of the London Mathematical Society, 14, 445–458, https://doi.org/10.1112/plms/s3-14.3.445, 1964.

Mitchell, D. L.: Evolution of snow-size spectra in cyclonic storms. Part II: Deviations from exponential form, J. Atmos. Sci., 48, 1885–1899, https://doi.org/10.1175/1520-0469(1991)048<1885:EOSSSI>2.0.CO;2, 1991.

Murphy, A. M., Ryzhkov, A., and Zhang, P.: Columnar vertical profiles (CVP) methodology for validating polarimetric retrievals in ice using in situ aircraft measurements, J. Atmos. Oceanic Technol., 37, 1623–1642, https://doi.org/10.1175/JTECH-D-20-0011.1, 2020.

Okuta, R., Unno, Y., Nishino, D., Hido, S., and Loomis, C.: CuPy: A NumPy-Compatible Library for NVIDIA GPU Calculations, in: Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS), http://learningsys.org/nips17/assets/papers/paper_16.pdf, 2017.

Phillips, V. T. J., Formenton, M., Bansemer, A., Kudzotsa, I., and Lienert, B.: A parameterization of sticking efficiency for collisions of snow and graupel with ice crystals, J. Atmos. Sci., 72, 4885–4902, https://doi.org/10.1175/JAS-D-14-0096.1, 2015.

Phillips, V. T. J., Yano, J.-I., and Khain, A.: Ice multiplication by breakup in ice-ice collisions. Part I: Theoretical formulation, J. Atmos. Sci., 74, 1705–1719, https://doi.org/10.1175/JAS-D-16-0224.1, 2017.

Prat, O. P. and Barros, A. P.: A robust numerical solution of the stochastic collection–breakup equation for warm rain, J. Atmos. Sci., 46, 1480–1497, https://doi.org/10.1175/JAM2544.1, 2007a.

Prat, O. P. and Barros, A. P.: Exploring the use of a column model for the characterization of microphysical processes in warm rain: Results from a homogeneous rainshaft model, Adv. Geosci., 10, 145–152, https://doi.org/10.5194/adgeo-10-145-2007, 2007b.

Prat, O. P., Barros, A. P., and Testik, F. Y.: On the influence of raindrop collision outcomes on equlibrium drop size distributions, J. Atmos. Sci., 69, 1534–1546, https://doi.org/10.1175/JAS-D-11-0192.1, 2012.

Ryzhkov, A., Pinsky, M., Pokrovsky, A., and Khain, A.: Polarimetric radar observation operator for a cloud model with spectral microphysics, J. Appl. Meteor. Climatol., 50, 873–894, https://doi.org/10.1175/2010JAMC2363.1, 2011.

Ryzhkov, A., Zhang, P., Reeves, H., Kumjian, M., Tschallener, T., Trömel, S., and Simmer, C.: Quasi-Vertical Profiles – A new way to look at polarimetric radar data, J. Atmos. Oceanic Technol., 33, 551–562, https://doi.org/10.1175/JTECH-D-15-0020.1, 2016.

Ryzhkov, A. V. and Zrnić, D. S.: Radar polarimetry for weather observations, Springer Atmospheric Sciences, 2019.

Safronov, V. S.: A particular case of the solution of the coagulation equation, Soviet Physics – Doklady, 7, 967–969, https://mathnet.ru/dan27172, translated from Dokl. Akad. Nauk SSSR, 147(1), 64–67, 1962, 1963.

Schrom, R. S., Kumjian, M. R., and Lu, Y.: Polarimetric radar signatures of dendritic growth zones within Colorado winter storms, J. Appl. Meteor. Climatol., 54, 2365–2388, https://doi.org/10.1175/JAMC-D-15-0004.1, 2015.

Scott, W. T.: Analytical studies of cloud droplet coalesence I, J. Atmos. Sci., 25, 54–65, https://doi.org/10.1175/1520-0469(1968)025<0054:ASOCDC>2.0.CO;2, 1968.

Simmel, M., Trautmann, T., and Tetzlaff, G.: Numerical solution of the stochastic collection equation – comparison of the Linear Discrete Method with other methods, Atmos. Res., 61, 135–148, https://doi.org/10.1016/S0169-8095(01)00131-4, 2002.

Smoluchowski, M.: Versuch einer mathematischen Theorie der Koagulations-kinetik kolloider Lösungen, Zeitschrift für Physikalische Chemie, 92, 129–168, https://doi.org/10.1515/zpch-1918-9209, 1917.

Stammler, S. M. and Birnstiel, T.: DustPy: A Python Package for Dust Evolution in Protoplanetary Disks, The Astrophysical Journal, 935, 35, https://doi.org/10.3847/1538-4357/ac7d58, 2022.

Straub, W., Beheng, K. D., Seifert, A., Schlottke, J., and Weigand, B.: Numerical investigation of collision-induced breakup of raindrops. Part II: Parameterizations of coalescence efficiencies and fragment size distributions, J. Atmos. Sci., 46, 576–588, https://doi.org/10.1175/2009JAS3175.1, 2010.

Tobin, D. M. and Kumjian, M. R.: Polarimetric radar and surface-based precipitation-type observations of ice pellet to freezing rain transitions, Wea. Forecasting, 32, 2065—-2082, https://doi.org/10.1175/WAF-D-17-0054.1, 2017.

Tzivion, S., Feingold, G., and Levin, Z.: An efficient numerical solution to the Stochastic Collection Equation, J. Atmos. Sci., 44, 3139–3149, https://doi.org/10.1175/1520-0469(1987)044<3139:AENSTT>2.0.CO;2, 1987.

Wang, L.-P., Xue, Y., and Grabowski, W. W.: A bin integral method for solving the kinetic collection equation, Journal of Computational Physics, 226, 59–88, https://doi.org/10.1016/j.jcp.2007.03.029, 2007.