

Exploring A Deterministic Model for Stochastic Coalescence

Class project - Numerical ODE's (MATH-269B)

Louise Nuijens

March 20, 2008

Introduction

Warm rain formation in clouds is initiated by collision and coalescence of cloud droplets. Given a distribution of cloud droplets in the cloud updraft, determined by the condensation of water vapor on cloud condensation nuclei (CCN), coalescence processes lead to a shift of the initial distribution to larger drop sizes. This ultimately leads to the formation of (rain)drops that have collected sufficient mass to fall out of a cloud.

This report describes the implementation of a recently proposed deterministic scheme for the stochastic coalescence process that is based on binary grid refinement (Struckmeier, 2005). This approach allows one to specify an increasingly refined grid to model the discrete stochastic coalescence equation (SCE) to achieve higher accuracy. In the next two sections I will shortly describe the stochastic coalescence process and review the numerical approaches taken in past studies to model the coalescence process of cloud droplets, being only one example of a physical process, among many others, that can be described by the SCE. After a verification of the proposed scheme for a simplified problem and an error analysis of solutions for different refined grids, a start is made to use the proposed scheme to study more realistic distributions and kernels (for cloud drops).

1. The stochastic coalescence equation

The collision-coalescence process can be described by two different masses (droplets) x and y moving in a given cloud volume, that may merge into a single drop with mass $x + y$ if they are sufficiently close. Considering a function $K(x, y)dt$, the collection kernel, that describes the probability that two droplets in a homogeneous cloud will occupy the same volume and collide in a given time interval dt , a 'full' stochastic model of the coalescence process would predict the probability of a certain droplet mass distribution at any given time t . The more commonly applied and less complex 'quasi-stochastic' models, that are based on the deterministic Smoluchowski coagulation equations, instead predict the average number of droplets $f(x, t)$ in a cloud at any time t , either in discrete mass space ($x = 1, 2, 3, \dots$) or in continuous mass space ($x, x + dx$):

$$\frac{d}{dt}f(x, t) = \frac{1}{2} \sum_{y=1}^{x-1} k(y, x-y)f(y, t)f(x-y, t) - f(x, t) \sum_{y=1}^{\infty} k(x, y)f(y, t) \quad (1)$$

$$\frac{d}{dt}f(x, t) = \frac{1}{2} \int_0^x k(y, x-y)f(y, t)f(x-y, t)dy - f(x, t) \int_0^{\infty} k(x, y)f(y, t)dy \quad (2)$$

In these equations the first term describes the average rate of production of droplets of mass x , due to coalescence of droplets whose masses sum to x . The second term describes the average rate of depletion of x droplets due to the coalescence with other droplets. Details about the local motion of droplets and their tendency to coalesce upon colliding is assumed to be included in the (stochastic) collection kernel $K(x, y)$, which is assumed to be symmetric ($K(x, y) = K(y, x)$). Though usually referred to as the stochastic coalescence equation (SCE), this equation is truly deterministic (except for the definition of the kernel) and relies on the assumption that the average number of coalescences ($f(x, y)$) is equal to the average number of x droplets times the average number of y droplets, hence that they are statistically uncorrelated. Any statistical fluctuations about the average number $f(x, t)$ are ignored and it is assumed that the 'average' represents the outcome after a sufficient number M of independent samples. Any solution to the SCE will therefore always be just an approximation to the real average droplet number concentration.

2. Numerical approaches

The collection of small cloud droplets by large cloud drops (a process called 'accretion') dominates the droplet growth by the collision-coalescence process (Berry and Reinhardt, 1974). The presence of few large cloud drops in the initial cloud droplet spectrum, which speed up the coalescence process, is therefore thought to explain the rapid onset of warm rain in shallow clouds. This is relevant with respect to modeling coalescence: the resolution, or mass grid spacing, of the cloud droplet spectrum should be fine enough to accurately represent both the growth of smaller droplets as well as the evolution of the large droplets.

To catch the tail of the droplet distribution and to conserve total mass, a large number of grid points N is needed if a deterministic model is used. This can become an expensive procedure as the computational effort for numerically solving the right-hand side of the Smoluchowski equation is on the order of N^2 . Stochastic models on the other hand use (quasi-) Monte Carlo methods by directly simulating the probabilistic behavior of M number of masses. Although less computationally expensive (the CPU time increases at least linearly with M), these methods have a slow convergence rate ($1/\sqrt{M}$) and require a large number of independent samples in order to achieve high accuracy. Very accurate Monte Carlo simulations have been successfully performed by Gillespie (1972, 1975) to model the cloud droplet coalescence process, but such approaches are less suited for application in a very detailed cloud model since only a small cloud portion can be simulated in a reasonably short time period.

It has been shown that solutions to the (deterministic) SCE contain enough information to describe the mean behavior of cloud droplets in a finite volume so that the SCE is a justified approximation, though in fact this has only been achieved for the simplest collection kernel, one being constant $K(x, y) = 1$ (Bayewitz et al., 1974). The perhaps most accurate simulation of cloud droplet coalescence has been performed by Berry and Reinhardt (1974), later implemented in the detailed microphysical model of Hall (1980). Berry and Reinhardt (1974) solved the continuous SCE using Lagrange interpolation to

approximate the integrands on an exponentially increasing grid (with a grid spacing of $\ln(r)$, with r being the droplet radius. In more complex multidimensional numerical cloud models, the SCE is not explicitly resolved, but in recent work has provided a theoretical basis for improved double-moment parameterizations of cloud microphysics (Seifert and Beheng, 2001).

Numerical solutions of both the discrete and continuous SCE have been compared by Young (1974) and the latter appeared to differ only slightly from its discrete version. The stability of these schemes is studied by Brown (1975) (using a trapezoidal scheme to solve for the integrant) who found that the ideal time step for a simulation depends largely on the grid selection, as well as the collection kernel and peak of the initial distribution. In specific, the allowed time step is longer when grid points have sufficient spread in the tail of the distribution, when the maximum drop size is smaller and the spectral peak is lower.

3. Description of the binary grid and numerical scheme

The deterministic scheme proposed by Struckmeier (2005) can be used to solve the discrete SCE on a binary grid Ω_0 given by:

$$\begin{aligned}\Omega_0 &= \{x \in N : x = 2^k; k = 0, 1, 2, \dots\} \\ x_i &= \{1, 2, 4, 8, 16, \dots\} = x_1, x_2, x_3, x_4, x_5, \dots, x_N\end{aligned}\tag{3}$$

which is truncated at any value q such that $x_N = 2^q$. This grid can be further refined to any new level l by introducing exactly 2^l new grid points between the existing binary points $2^n, 2^{n+1}$. As an example, consider the refined grid Ω_l with $l = 2$ given by:

$$\begin{aligned}x_i &= \{1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 16, \dots\} \\ &= \{x_1, x_2, x_3, \dots, x_N\}\end{aligned}\tag{4}$$

where the first 2^{l+1} values represent a full integer grid, followed by grid points that are separated more than 1 unit in mass space. The total number of grid points for a grid level l for $x_i^l \leq 2^q$ is given by 2^q if $q \leq l$ or $(q - l + 1)2^l$ if $q > l$. More details on useful characteristics of this grid can be found in Struckmeier (2005).

The discrete SCE can be written as a density function $g(x, t)$:

$$\begin{aligned}\frac{d}{dt} g(x, t) &= \sum_{1 \leq y < x} \tilde{k}(y, x - y)g(y, t)g(x - y, t) - g(x, t) \sum_{1 \leq y} \tilde{k}(x, y)g(y, t) \\ &= Q_1(x, t) - Q_2(x, t)\end{aligned}\tag{5}$$

where

$$\tilde{k}(x, y) = \frac{k(x, y)}{y}\tag{6}$$

This set of ODE's can be numerically solved on the (globally refined) binary grid x_i without changing the actual equation, except for introducing a modified version of the collection kernel $K(x_i, y_j) = I_i \tilde{k}(x_i, y_j)$. I_i gives a certain weight to the kernel depending on the mass at each grid point, whereby I_i equals 1 for the full integer grid up to 2^{l+1} , after which it increases with mass x_i . This piecewise reconstruction is physically based on the properties of different (binary) masses colliding with each other.

For the loss term in the SCE equation, this implies that $g(x, t)$ for masses x in between the binary grid points, for instance $x = 3, 5 - 7$, and so on for Ω_0 , are approximated by $g(x, t)$ at $x = 2, 4$ and the kernels at the grid points for the latter are multiplied by 2 and 4 respectively. For the gain term this implies that any mass, say $x = 4$, is estimated by computing the collision of masses at grid points $x = 1, 2$, where the kernel $k(x = 2, y = 2)$ is increased by a factor 2 allowing for the gain of mass $x = 4$ by collision of mass $x = 3$ with $x = 1$. The modification of the kernel therefore depends also on the level l of grid refinement. This procedure is extended to the entire grid (see Struckmeier (2005)).

If the above equation is discretized using an explicit first-order Euler scheme and the numerical solution to $g(x_i)$ is written as G_i one obtains:

$$G_i^{l,n+1} = G_i^{l,n} \Delta t + \sum_{1 \leq j < i}^N K(j, i-j) G_j^{l,n} G_{i-j}^{l,n} - G_i^{l,n} \sum_{1 \leq j}^N K(i, j) G_j^{l,n} \quad (7)$$

Alternatively the iterative Runge-Kutta scheme can be used, which evaluates the differential $\frac{dg(x,t)}{dt}$ in four subsequent steps for slightly modified functions $Q_{1,2}(x, t)$ at each time step. The advantage is a higher accuracy (fourth-order) and one can use a longer time step Δt . For any non-monotone scheme however, spurious oscillations can occur, especially at the tail of the solution where values are close to zero. This occurs for the unbounded kernel when applying the Runge Kutta scheme. To conserve the non-negativity of the solution, an adaptive time stepping is used. When the solution G_i is negative at any i , the same computation is repeated but for half the time step, which is then used throughout the remainder of the simulation.

A cubic spline is used to interpolate the values of G_i back to the full integer grid by solving for piecewise third-order polynomials, which as desired has a higher order interpolation (i.e., it uses more grid points) than the piecewise construction of the binary grid from a full integer grid. The schemes are programmed in Matlab and included in the Appendix.

4. Numerical solutions

(a) Verification of the scheme

In their study, Struckmeier (2005) test the numerical solutions of two simple problems for different refined grids. Using initial monodisperse data i.e. $G_1^{0,l} = 1, G_i^{0,l} = 0$ for $i >= 2$ for the constant kernel $\tilde{k}(x_i, y_j) = 1$ and unbounded kernel $\tilde{k}(x_i, y_j) = x_i + y_j$, it is shown that the computational effort is greatly reduced when using a lower grid level l , yet the

absolute errors between the numerical and exact solution are rather small.

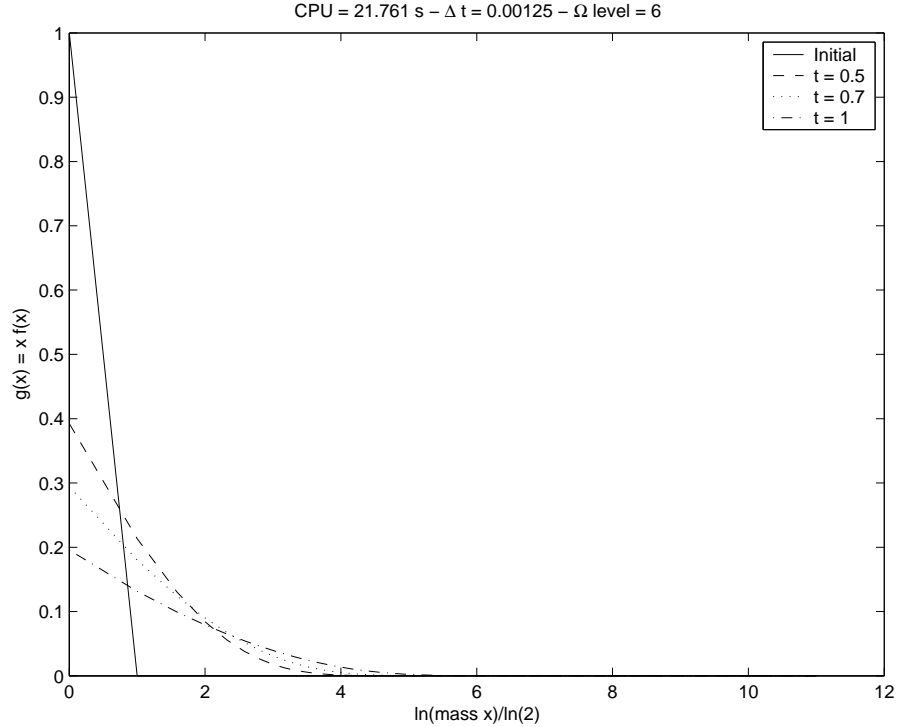


Figure 1: The density function g_x [mass x^{-1}] after 0.5,0.7 and 1 second for initial monodisperse data and unbounded kernel $k(x, y) = x + y$ for the Runge-Kutta time scheme with grid Ω_6 and $\Delta t = 0.01$.

To test the efficiency of my scheme (written in Matlab and included in the Appendix), the solutions for the unbounded kernel and mono-disperse data are plotted in Figure 1 for the Runge Kutta scheme respectively. The x-axis represents the exponent k of the binary grid 2^k , which is truncated at $k=q=11$. The solutions are calculated using the grid Ω_6 , with an initial $\Delta t = 0.01$ and plotted after 0.5, 0.7 and 1 seconds. The time step has to be adjusted three times in order to conserve non-negativity. The figure clearly demonstrates the shift of the initial spectrum to a larger mass, similar to the results in presented in Struckmeier (2005).

(b) *Error analysis*

The numerical solution of different grids, from Ω_0 up to the full integer grid Ω_{11} , are compared to the exact analytical solution. Figure 2 plots the absolute error defined as $|g(x) - G(x)|$ for the first $2^k, k \leq 8$ binary grid points. Errors decrease substantially when increasing the level l , and hence the solution converges when the grid spacing is refined. The solutions for the levels with $l > 7$ do not show much difference and overlap so that it can be concluded that any grid level $7 \leq l \leq 10$ is an excellent approximation to the solution obtained for the full integer grid (and the exact solution), at least up to

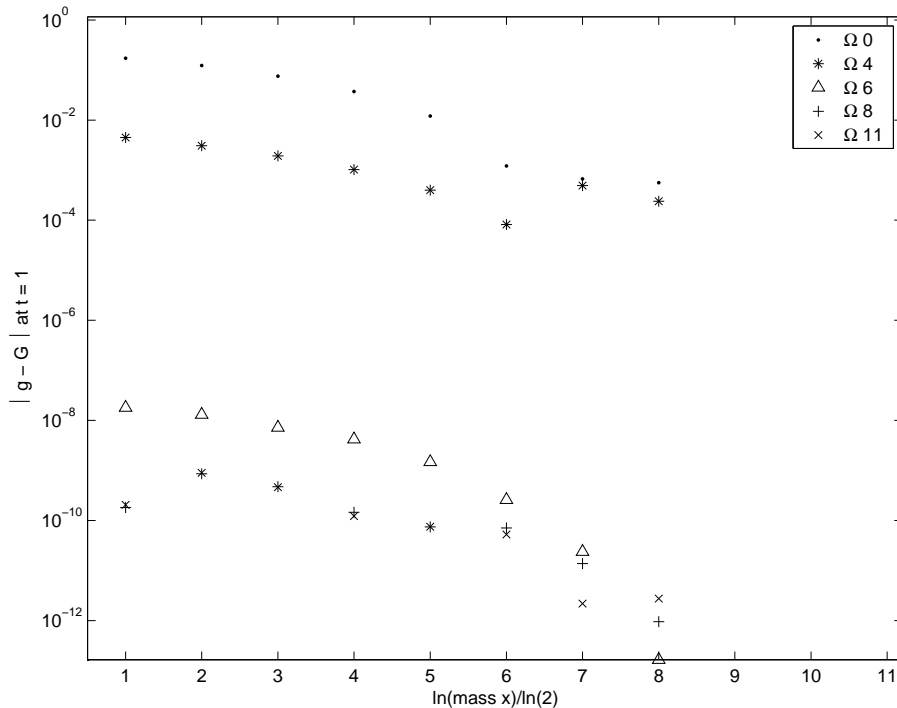


Figure 2: The absolute error defined by $|g(x) - G(x)|$ is plotted for $\Omega_{0,4,6,8,11}$ after time = 1 second for the first 2^k , $k \leq 8$ binary grid points. Further characteristics are as in Figure 1.

$t = 1$ s. Most of the improvement is obtained in the tail of the solution, where the grid continues to be refined with increasing l as opposed to the lower end of the spectrum. The error of gridpoint $x = 2^8$ appears to be even larger for Ω_{11} compared to Ω_8 , which is probably due to the fact that the time step on Ω_8 is reduced to 0.0025 at $t = 0.75$ and 0.76 during the simulation.

Similar conclusions can be drawn from solutions of the explicit Euler time scheme, yet the absolute errors with this scheme can be an $O(10^5)$ larger (if a time step of $\Delta t = 0.001$ is used).

Because none of the grid levels (excluding the full grid) have an equidistant grid spacing Δx , it does not seem straightforward to determine the rate of convergence between grids with different levels of refinement by comparing the L2 maximum error, $e_{\Delta x}^n = (\|g^n - G^n\|_{\Delta x}^2)^{1/2}$, with g^n and G^n as the exact and numerical solution after n time steps. Instead one may compare different grids by defining a relative deviation δ^n :

$$\delta = \frac{\sum_{i \geq 1}^N |G_i - G_{i, \Omega_{q-1}}|}{\sum_{i \geq 1}^N |G_i|} \quad (8)$$

that measures the average relative deviation of G^n from the solution $G_{\Omega_{q-1}}^n$ for the full integer grid given a truncation at 2^q . The summation occurs only over the grid points i of the grid G^n , so that as long as mass conservation ($\frac{d}{dt} \sum_{i \geq 1} G_i = 0$) is not violated, the

denominator equals 1. This methods would also allow for an evaluation of the benefits of a refined grid for more complex kernels for which no analytical solution is available.

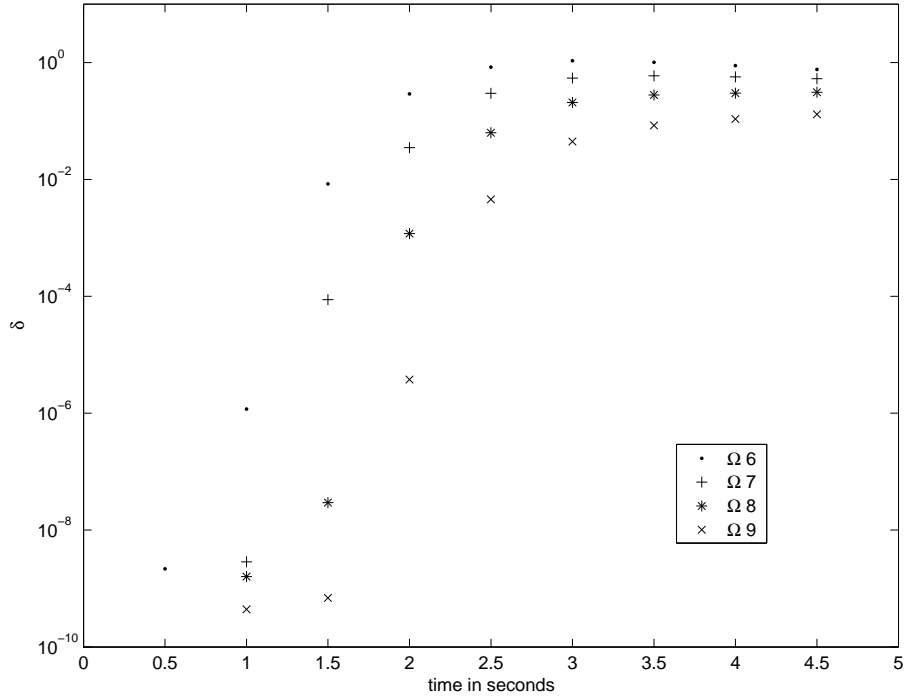


Figure 3: For the same kernel and initial distribution as Figure 1 the average relative deviation δ , defined in Equation 8 is plotted against time in seconds for four refined grid levels, $\Omega_{6,7,8,9}$ compared to Ω_{10} .

In Figure 3 δ is plotted for the four refined grids $\Omega_{6,7,8,9}$ relative to the full grid Ω_{10} . The values are plotted as a function of time, for every 0.5 s. Again the Runge-Kutta scheme has been used with an initial time step of 0.01. The results demonstrate that the relative deviation of the solution of each scheme quickly increases with time. After roughly 2-3 seconds (≈ 300 -1200 time steps depending on the grid level) the benefits of using a more refined grid have drastically decreased. It should be noted here that after a time of 1.5, 1.86, 2.18 and 2.46 seconds respectively mass conservation of each of the four grid levels is violated for the first time as a consequence of the truncation at $x = 2^{11}$. This means that comparison of δ becomes less meaningful with increasing time for this particular problem.

The figure suggests that adaptive grid refinement (i.e., starting with the binary grid and refining it during the computation) for this particular problem, where broadening of the spectrum occurs quickly, may not be that beneficial when time passes. Also, if one would start the computation with a (coarser) binary grid, the time step needs to be reduced several times in order to preserve non-negativity, thereby increasing the computational effort and counteracting its initial benefit.

(c) *A first look at cloud drops*

When studying coalescence of cloud drops, one could define a Gamma distribution as initial data for cloud droplets with radii $< 40 \mu\text{m}$, with the first gridpoint corresponding to a radius of $1 \mu\text{m}$. These are typical values corresponding to observed cloud droplet spectra and used in studies by Berry and Reinhardt (1974); Seifert and Beheng (2001). $40 \mu\text{m}$ is considered as a typical drop radius that separates the cloud drops from the rain drops, where the latter follow a different distribution (exponential) and have a different collection kernel. The collection kernel $k(x, y)$ for cloud drops is often empirically related to a linear collision efficiency $E(x, y)$, the drop radius r of the largest drop and the difference in fall velocity of the drops $\Delta V(r_x, r_y)$ as $K(x, y) = \pi r^2 \Delta V(r_x, r_y) E(x, y)^2$. The piecewise polynomial collection kernel derived by Long (1974) however provides a simple solution that could be implemented in the deterministic scheme: for $r_x \wedge r_y < 40 \mu\text{m}$: $\tilde{k}(x, y) = k_c(x^2 + y^2)$ and for $r_x \vee r_y \geq 40 \mu\text{m}$: $\tilde{k}(x, y) = k_r(x + y)$, where $k_{r,c}$ are constants.

It appeared tricky to get reasonable numerical solutions using a typical cloud droplet spectrum with the deterministic scheme and adaptive time stepping used in previous sections. After extending the truncation to $x_N = 2^{13}$ to ensure mass conservation, neither the Runge-Kutta nor the Euler forward scheme were able to produce non-negative solutions without decreasing the time step infinitely, for none of the refined (not even the full) grids. Although the initial time steps seem promising (Figure 4), solutions quickly become unstable, and to produce the results in Figure 4 the variance of the initial distribution and initial 'mean' mass needed to be small.

The simulation was also carried out for the unbounded kernel to have a look at to what extent the different initial data itself caused a problem. In this case, although one could use a coarser grid Ω_8 (Figure 5), negative solutions were absent, at least in the first two seconds (≈ 500 time steps), after which mass conservation was increasingly violated.

Conclusions

A recently proposed deterministic scheme for the discrete stochastic coalescence equation based on binary grid refinement has been tested for a simplified case using the Runge-Kutta time integration scheme. Results indicate that absolute errors can be significantly decreased when increasing the level refinement, as compared to the (coarsest) binary grid, but as the full integer grid is approached, the absolute error for different grids is similar, except in the tail of the distribution for large masses. With time, errors increase substantially and the differences between the solutions using different levels of refinement become small. Although initially the coarser grids require less computational time, the more frequently one needs to adjust the time step in order to conserve non-negativity, the less beneficial this grid. For simulations carried out over a longer time (up to a factor 4) than those in Struckmeier (2005), the deviations of the solution of different grid levels relative to the full integer grid become similar. This would imply adaptive grid refinement may not be that useful. Taking a first look at the use of this deterministic scheme for a more complex kernel does not lead to good results i.e., a different approach or adjustment to the scheme is necessary to prevent instability in the numerical solution.

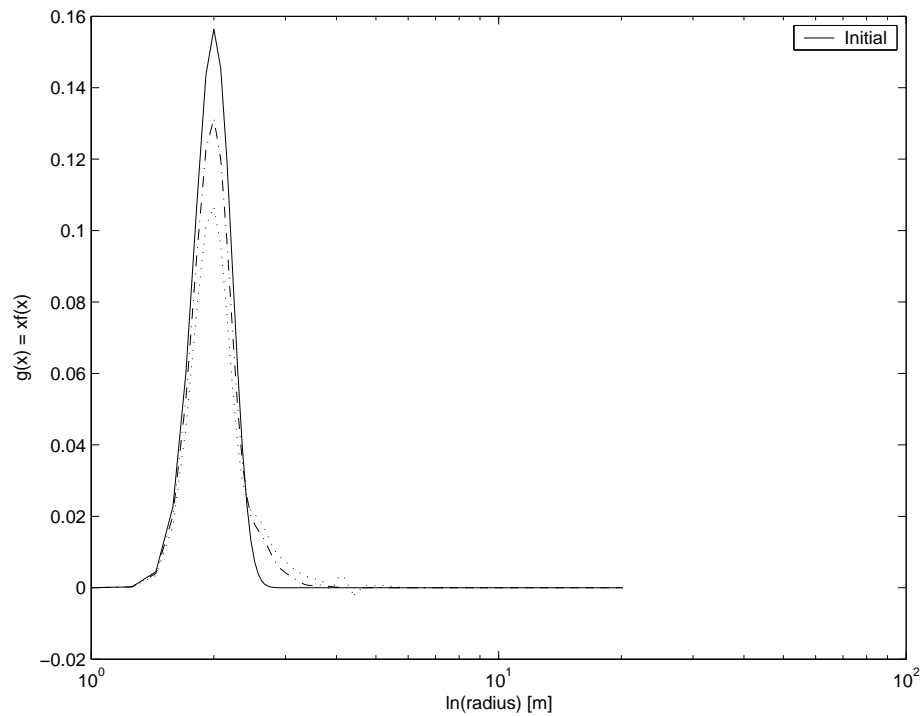


Figure 4: Solutions for an initial Gamma distribution with $\text{var}(x) = 0.1$ and $\langle x \rangle$ corresponding to $r = 2 \mu\text{m}$, after three time steps of $\Delta t = 0.01$ for a full grid truncated at $x_N = 2^{13}$ and the kernel $\tilde{k}(x, y) = (x^2 + y^2)$.

1 Appendix

```

% SCE.m
%
% Numerical solution to stochastic coalescence equation solved
% on a binary grid following Struckmeier (2005)
%
% Louise Nuijens ~ March 2008

%% Definitions

clear all;
global N K;

cput = cputime;

%method = 'FW' % Explicit first order forward time differencing
method = 'RK' % Fourth-order Runge Kutta scheme

```

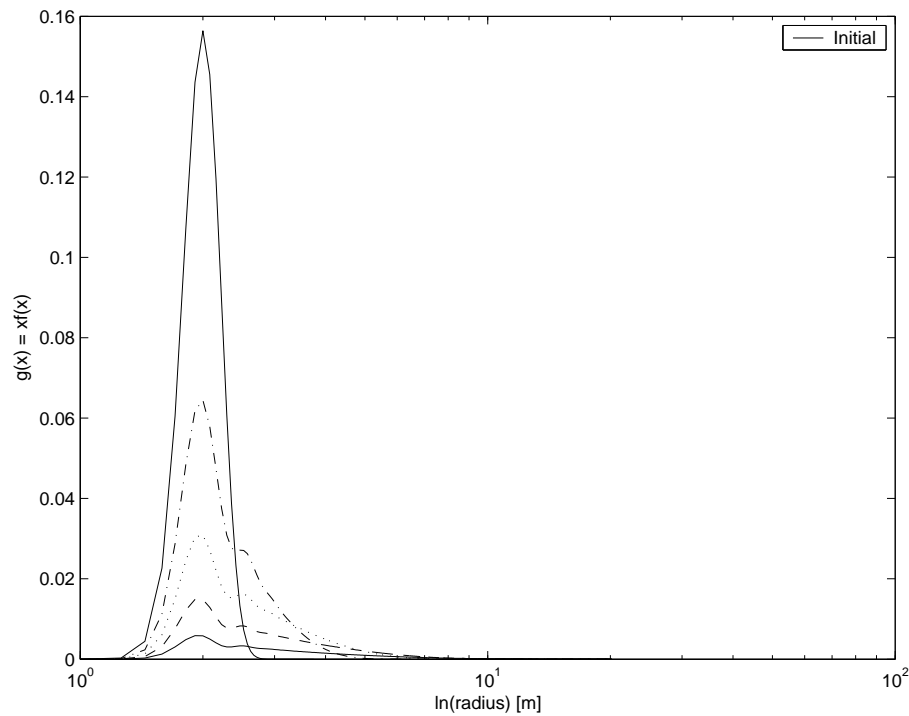


Figure 5: Similar as in Figure 4, yet for a coarser grid Ω_8 and the unbounded kernel $\tilde{k}(x, y) = (x + y)$ after a $t = 0.05, 1, 1.5$ and 2 s.

```

%
% Set time interval (t)
%
deltat = 0.01;      % Time step
tfinal = 1.;       % Final time
tinitial = 0.;

%
% Define grid boundaries (x) and truncation of grid (q)
%
level = 6;         % for a full integer grid, choose level = q
q = 11;
xleft= 1;
xright= 2^(q) ;   % truncate at 2^q

%
% Define total number of spacesteps on refined binary grid
%
if q <= level
    N = 2^q;
else

```

```

        N = (q-level+1)*(2^level);
    end

%
% Construct mass x(m) on binary grid and weighing points I(m) for piecewise
% reconstruction of the collection kernel
%
m = 1;
nsum = 0;
for n=0:(q-1)
    ngrid = min(2^n,2^level); % number of grid points on interval [2^n - 2^n+1]
    nsum = nsum+ngrid;
    dx = (2^n)/ngrid ; % gridspacing on interval [2^n - 2^n+1]
    x(m:nsum) = (2^n)+ (0:ngrid-1)*dx; % x values on interval [2^n - 2^n+1]
    index(n+1) = m;

    if m < 2^(level+1)
        I(m:nsum) = 1.;
    else
        p = log(x(m))/log(2);
        I(m) = 2^(p-level) ; % extend binary sections to the right

        for k=(m+1:nsum)
            I(k) = 2^((log(x(k))/log(2))-level);
        end
    end

    m = nsum+1;
end

x(m) = xright; % assign rightmost boundary point
p = log(x(m))/log(2);
I(m) = 2^(p-level)+2^(p-level-1);
index(m) = N;
clear p m nsum;

%
% Define modified collection kernel K(i,j)
%
K = zeros(N);

for (i = 1:N)
    for (j = 1:N)
        K(i,j) = I(j)*((x(i)+x(j))/x(j)); % Modified kernel K
    end
end
end

```

```

clear I;
%
% Define initial distribution (monodisperse)
%
g0(1) = 1;
g0(2:N) = 0.;

%
% Define initial spectrum on binary (refined) grid
%
gold = g0;
clear g0,x;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Computing mass density G(i) using finite differencing on
% the binary grid
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

nplot = 1;
t = tinitial;
cnt = 1;
while(t <= (tfinal-deltat))

    Q = compute(K,gold,N);

    if strcmp(method,'FW')
        gnew = gold + Q*deltat;
        clear Q;
        gold = gnew;
    else
        gnew = rungekutta(deltat,gold,Q,K,N);
    %
        % Apply adaptive time stepping to ensure non-negativity of solution
        %
        while any(gnew<0)
            deltat = deltat/2.
            gnew = rungekutta(deltat,gold,Q,K,N);
        end

        gold = gnew;
    end

    t = t+deltat;

end

```

```

    cnt = cnt+1;
    clear gold K x N;
deltat = 0.01;
    %
% Calculate exact solution
%
f = zeros(1,N);

for i=1:N

    % kernel k(i,j) = (i+j)/j
    f(i) = (x(i)^(x(i))/factorial(x(i)))*((1-exp(-1.*t))
    ^x(i)-1))*exp(-1.*x(i)*(1-exp(-1.*t))-t);

    % constant kernel k(i,j) = 1./j
    %f(i) = ((4*x(i))/(tfinal+2)^2)*(tfinal/(tfinal+2))^x(i)-1) ;
end

%
% Calculate absolute error and on binary grid points
%
    for n = 1:q
%error(1,n) = abs(f(index(n))-gnew(index(n)));
    end

    clear f gnew index;
cput2(1) = cputime - cput;

end

%
% Interpolate distribution on binary grid to full integer grid
%
%xx = xleft:xright;
%gfull = spline(x,gnew,xx);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Compute the time derivative dG/dt ~ Stochastic Coalescence Equation
% for a given distribution gold, a defined kernel k, and N grid points
%
function [Q] = compute(K,gold,N);

for (i=1:N)
    Q1 = 0.;
    Q2 = 0.;

```

```

        for (j=1:i-1)
            Q1 = Q1 + (K(i-j,j)*gold(i-j)*gold(j));
        end
        for (j=1:N)
            Q2 = Q2 + (K(i,j)*gold(i)*gold(j));
        end
        Q(i) = Q1 - Q2;

end
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
% Apply Runge Kutta time integration
%
function [gnew] = rungekutta(deltat,gold,Q,K,N);

Q = deltat*Q;
deltat2 = deltat/2.;
g1 = gold+Q/2.;
Q_2 = deltat*compute(K,g1,N);
clear g1;

g2 = gold+Q_2/2.;
Q_3 = deltat*compute(K,g2,N);
clear g2;

g3 = gold+Q_3;
Q_4 = deltat*compute(K,g3,N);

clear g3;
gnew = gold+((Q+(2.*Q_2)+(2.*Q_3)+Q_4)/6.);

clear g4 Q Q_2 Q_3 Q_4;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Bibliography

- Bayewitz, M., J. Yerushalmi, S. Katz and R. Shinnar, 1974: The extent of correlations in a stochastic coalescence process. *J.Atmos.Sci.*, **31**.
- Berry, E. and R. Reinhardt, 1974: An analysis of cloud drop growth by collection: Part i. Double distributions. *J.Atmos.Sci.*, **31**.
- Brown, P. J., 1975: Stability analysis of a finite-difference approximation to the stochastic coalescence equation. *J.Atmos.Sci.*, **32**.
- Gillespie, D., 1972: The stochastic coalescence model for cloud droplet growth. *J.Atmos.Sci.*, **29**.
- Gillespie, D., 1975: An exact method for numerically simulating the stochastic coalescence process in a cloud. *J.Atmos.Sci.*, **32**.
- Hall, W., 1980: A detailed microphysical model within a two-dimensional dynamic framework: model description and preliminary results. *J.Atmos.Sci.*, **37**.
- Long, A., 1974: Solutions to the droplet collection equation for polynomial kernels. *J.Atmos.Sci.*, **31**.
- Seifert, A. and K. Beheng, 2001: A double-moment parameterization for simulating autoconversion, accretion and selfcollection. *Atmospheric Research*, **59**.
- Struckmeier, J., 2005: A deterministic scheme for Smoluchowski's coagulation equation based on binary grid refinement. *Numerical algorithms*, **40**.
- Young, K., 1974: The evolution of drop spectra due to condensation, coalescence and breakup. *J.Atmos.Sci.*, **32**.