Numerical Algorithms (2005) **40:** 233–249
DOI 10.1007/s11075-005-4186-3

A deterministic scheme for Smoluchowski's coagulation equation based on binary grid refinement

Jens Struckmeier

Department of Mathematics, University of Hamburg, Bundesstr. 55, D-20146 Hamburg, Germany
E-mail: struckmeier@math.uni-hamburg.de

Received 23 April 2004; accepted 14 February 2005
Communicated by C. Brezinski

We present a deterministic scheme for the discrete Smoluchowski's coagulation equation based on a binary grid refinement. Starting from the binary grid $\Omega_0 = \{1, 2, 4, 8, 16, \ldots\}$, we first introduce an appropriate grid refinement by adding at each level 2^l grid points in every binary subsection of the grid Ω_l. In a next step we derive an approximate equation for the dynamic behavior on each level Ω_l based on a piecewise constant approximation of the right hand side of Smoluchowski's equation. Numerical results show that the computational effort can be drastically decreased compared to the corresponding complete integer grid. When considering unbounded kernels in Smoluchowski's equation we use an adaptive time step method to overcome numerical instabilities which may occur at the tails of the density function.

Keywords: Smoluchowski's coagulation equation, refined binary grids, deterministic time integration, adaptive time step method

1. Introduction

Mathematical models of coalescence, like coagulation, gelation, aggregation, etc., are based on the so-called Smoluchowski's equation introduced by Smoluchowski in [18,19]. In discrete form, the model is defined by the infinite system of ordinary differential equations

$$\frac{\mathrm{d}c_i}{\mathrm{d}t}(t) = \frac{1}{2} \sum_{1 \leqslant j < i} \tilde{k}(i-j, j) c_{i-j}(t) c_j(t) - \sum_{j \geqslant 1} \tilde{k}(i, j) c_i(t) c_j(t), \quad i \in \mathbb{N}, \tag{1.1}$$

where $N_0 c_i(t)$ denotes the number of clusters of mass i at time t, N_0 is the total number of clusters at time $t = 0$ and $\mathbb{N} = \{1, 2, \ldots\}$. The coagulation kernel $\tilde{k}(i, j)$ in (1.1) is assumed to be positive and symmetric, i.e.

$$\tilde{k}(i, j) \geqslant 0, \qquad \tilde{k}(i, j) = \tilde{k}(j, i) \quad \forall i, j \in \mathbb{N}.$$

Introducing a density function by $f_i = ic_i$ equation (1.1) may be rewritten in the form

$$\frac{\mathrm{d}f_i}{\mathrm{d}t}(t) = \sum_{1\leqslant j<i} k(i-j,j)f_{i-j}(t)f_j(t) - \sum_{j\geqslant 1} k(i,j)f_i(t)f_j(t), \quad i = 1,2,\ldots, \tag{1.2}$$

where the modified coagulation kernel reads

$$k(i,j) = \frac{1}{j}\tilde{k}(i,j).$$

Assuming that an interchange of the summation order on the right-hand side of (1.2) is valid, one directly obtains the conservation principle (mass conservation) in the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_{i\geqslant 1} f_i(t) = 0. \tag{1.3}$$

In particular, for numerical algorithms it seems to be more appropriate to use (1.2) instead of (1.1), because numerical errors may be controlled looking at the mass conservation (1.3) for approximate solutions. To our knowledge, Babovsky in [2] was the first who proposed a stochastic scheme based on the form (1.2).

Exact solutions for Smoluchowski's equation are known for particular kernels and initial conditions, see [1]. In the case of mono-disperse initial data, i.e. $f_1 = 1$, $f_j = 0$, $j \geqslant 2$ and the constant kernel $\tilde{k}(i,j) = 1$, one has

$$f_i(t) = \frac{4i}{(t+2)^2}\left(\frac{t}{t+2}\right)^{i-1}. \tag{1.4}$$

For the unbounded kernel $\tilde{k}(i,j) = i + j$ the exact solution – again for mono-disperse data – reads

$$f_i(t) = \frac{i^i}{i!}\left(1-\mathrm{e}^{-t}\right)^{i-1}\mathrm{e}^{-i(1-\mathrm{e}^{-t})-t}. \tag{1.5}$$

These analytic expressions are perfectly suited to validate and compare numerical simulation schemes, either deterministic or stochastic ones and we will validate our numerical approach in section 3 on the basis of the expressions given above.

Another important observation is that if the density functions f_i are non-negative at time $t = 0$, then the same holds for larger times, see [12,14]. This yields a further error indicator for (deterministic) numerical schemes, which should respect this property at the discretized level. Actually this restriction yields the necessaries to introduce an adaptive time step method, see also the discussion on deterministic schemes given below.

It is widely believed that from a computational point of view Smoluchowski's equation is more tractable by stochastic algorithms compared to deterministic ones and several authors proposed algorithms based on the Monte Carlo approach, see (in chronological order) [2,5–8,13,16,20]. The computational effort of deterministic schemes is of the order $\mathrm{O}(N^2)$, where N denotes the total number of grid points, which is used to approximate the infinite system of differential equations as well as the infinite summation on the right-hand side of Smoluchowski's equation. The number N has to be chosen large enough in order to satisfies a discrete mass conservation, although the number of clusters with large mass, i.e. the tail of the density function $\{f_1, f_2, \ldots\}$ is often rather small. This even prevents to use higher order schemes to integrate the system of differential equations in time.

Besides the large computational effort a further problem occurs – at least for unbounded kernels $\tilde{k}(i, j)$ – when one tries to integrate Smoluchowski's equation numerically. When integrating the system (1.2) spurious oscillations appear at the tails of the density function, which can lead to negative values for the density values f_i at large i. These negative values, even arbitrary small, may lead in the sequel to instabilities of the whole system. Hence, one should take care that the positivity of solutions of Smoluchowski's equation is taken over to the discretized system and in our numerical results we force the strict positivity of the numerical approximates using an adaptive time step method, see sections 2 and 3. This further indicates why deterministic schemes for Smoluchowski's equation are not at all straightforward.

In stochastic algorithms one performs a direct simulation of the dynamic behavior of M clusters, each carrying a specific mass. The mass of each cluster changes according to a stochastic algorithm based on a weak form of the right-hand side of Smoluchowski's equation. The computational effort of such schemes is linear in the number of clusters, but the order of convergence is with $\mathrm{O}(1/M^{1/2})$ rather slow. Moreover, stochastic algorithms contain statistical fluctuations such that independent samples have to be performed and the number M has to be chosen large enough, in particular $M \gg N$, in order to guarantee a sufficient accuracy, e.g., at the tail of the density function.

The performance of Monte Carlo schemes may be improved using the concept of variance reduction, see [10] for an application to Smoluchowski's equation. Moreover it is known, that the use of quasi-random sequences in stochastic algorithms may improve the accuracy as well as the computational effort [15]. Recently, Lecot and Wagner proposed a quasi-Monte Carlo method on the basis of the Faure sequence in base 3 [11]. They proved an error estimate of the order $\mathrm{O}(1/M^{1/3})$, where M denotes the number of simulation particles, which seems to be a lower order of convergence compared to standard Monte-Carlo schemes. Nevertheless, the numerical experiments given in [11] indicate, that the quasi-Monte Carlo approach yields a higher accuracy then standard Monte-Carlo schemes.

In the present work we propose a deterministic scheme which is based on a particular refinement technique starting from the binary grid $\{1, 2, 4, 8, 16, \ldots\}$. At each grid level l a finer grid is defined by adding exactly 2^l grid points in every binary subsection of Ω_l such that the new grid Ω_{l+1} is uniformly in each binary subsection. This technique allows us

- to catch the tails of the distribution function,
- using a total number of grid points on Ω_l, which is much smaller compared to the full integer grid.

Given a certain refined grid Ω_l we derive in a next step a suitable approximation of Smoluchowski's dynamic on the reduced grid Ω_l. The reduced dynamic is obtained from a piecewise constant approximation of the gain term on the right-hand side of (1.2). The approximation of the loss term then follows from a physically motivated detailed-balance relation and yields a dynamic on the refined grids which is invariant at each level except that a modified kernel is introduced at each level. Finally, the resulting system is numerically integrated using standard schemes for systems of ordinary differential equations, like a fourth-order Runge–Kutta method. To ensure non-negativity of the numerical approximation, which is the discrete analogue of the non-negativity of solutions of (1.2), an adaptive time step method is used in the case of the unbounded kernel $\tilde{k}(i, j) = i + j$.

Because the number of grid points on a refined grid Ω_l can be much smaller than the number of points on a complete grid, even the computational effort, which is at least quadratic in the number of unknown, when applying deterministic schemes, is drastically decreased: numerical results (section 3) show that (without any significant loss in the accuracy) the computational effort may be reduced by up to one order of magnitude and even more.

2. A deterministic scheme based on grid refinement

The deterministic scheme presented in the following is based on a successive grid refinement starting from the binary grid Ω_0 given by

$$\Omega_0 = \left\{ j \in \mathbb{N}\colon\ j = 2^k,\ k = 0, 1, \ldots \right\} = \{1, 2, 4, 8, 16, \ldots\} =: \left\{ x_1^0, x_2^0, x_3^0, \ldots \right\}.$$

The idea is to introduce a hierarchy of grids Ω_l, $l \geqslant 1$, which tends for $l \to \infty$ to the integer set $\mathbb{N}$. Then we use at each grid level Ω_l Smoluchowski's equation given in the form (1.2), by introducing an appropriate way how to derive a modified kernel $K_l(i, j)$ out of the given kernel $k(i, j)$.

In section 2.1 we formulate our grid refinement technique and state some basic properties of the grid levels Ω_l, which are useful to implement the grid refinement on a computer. An approximation of Smoluchwoski's dynamic which can be used on an arbitrary grid Ω_l is proposed in section 2.2. The procedure is based on the construction of a modified kernel for the gain term on the right-hand side of (1.2) using a piecewise constant approximation. The approximation of the loss term in (1.2) then directly follows from a physical detailed-balance principle. Section 2.3 deals with an adaptive time step method, which should ensure positivity of the numerical approximation and is necessary at least for unbounded kernels to prevent the formation of instabilities.

2.1. Hierarchic grid refinement

Let us first look at a way how to define a successive refinement of the initial binary grid Ω_0 formulated above: given a grid Ω_l at level l we define the next finer grid Ω_{l+1} by introducing between two binary points 2^k and $2^{k+1} \in \Omega_l$ exactly 2^l new points x_m, such that the grid Ω_{l+1} is uniformly between 2^k and 2^{k+1}. If the grid Ω_l already contains a complete binary subsequence, no new grid points are introduced. Like above, we will denote the grid points of Ω_l by $x_1^l, x_2^l, \dots$.

E.g., the refined grids Ω_l, $l = 1, \dots, 4$, are given by

$$
\begin{aligned}
\Omega_1 &= \{1, 2, 3, 4, 6, 8, 12, 16, 24, 32, 48, 64, \dots\},\\
\Omega_2 &= \{1, \dots, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32, \dots\},\\
\Omega_3 &= \{1, \dots, 8, 9, 10, 11, 12, 13, 14, 15, 16, 18, 20, 22, 24, 26, 28, 30, 32, \dots\},\\
\Omega_4 &= \{1, \dots, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, \dots\}.
\end{aligned}
$$

Let us summarize some basic properties of the grid refinement given above.

Proposition 1.

1. The grid Ω_l contains the first 2^{l+1} integers, i.e.

$$\left\{x_1^l, x_2^l, \dots, x_{2^{l+1}}^l\right\} = \left\{1, 2, \dots, 2^{l+1}\right\}.$$

2. The number of grid points of Ω_l in the interval $[2^n, 2^{n+1} - 1]$, $n = 0, 1, \dots$, is given by the following:

$$\#\left\{x_m^l \in \Omega_l\colon\ 2^n \leqslant x_m^l < 2^{n+1}\right\} = \min\left\{2^n, 2^l\right\}.$$

3. The number of grid points of Ω_l with $x_n^l \leqslant 2^q$, $q \in \mathbb{N}$ is given by the following:

$$\#\left\{x_n^l \in \Omega_l\colon\ x_n^l \leqslant 2^q\right\} = \begin{cases} 2^q, & q \leqslant l,\\ (q - l + 1) \cdot 2^l, & q > l. \end{cases}$$

Proof. Looking at the grid level Ω_l the total number of grid points included in the binary section $\{2^n, \dots, 2^{n+1} - 1\}$ is equal to $2^{\min(n,l)}$: hence the results. □

2.2. *A Smoluchowski-type equation on grid level Ω_l*

Let us first consider the binary grid Ω_0 and look for a way how to formulate an equation on Ω_0, which approximates the dynamics given by the complete Smoluchowski-system. The task is to define an appropriate approximation of the right-hand side of Smoluchowski's equation using the reduced number of grid points on Ω_0 and this is obviously connected to a way how to recover values f_i for $i \in \mathbb{N}$ from discrete values given on the binary grid Ω_0.

Let us denote by g_n^0, $n = 1, 2, \dots$, the approximation of f_i with $i = 2^{n-1}$ on the binary grid Ω_0 and by $L_{\Omega_0}(i)$ the approximation of f_i, $i \in \mathbb{N}$, using the values g_n^0, $n = 1, 2, \dots$.

Then the gain term of the right-hand side of (1.2) for $i = 2^{n-1}$ may be written as

$$\sum_{j\geqslant 1} k(i, j) f_i f_j \approx g_n^0 \sum_{l\geqslant 1} k(i, l) L_{\Omega_0}(l). \tag{2.1}$$

One should notice that the summation on the right-hand side of (2.1) still runs over the whole set of integers $\mathbb{N}$, i.e. one further needs to simplify the right-hand side in order to reduce it to a sum over the binary grid points $i = 2^{m-1}$. The most simple reconstruction is a piecewise constant recovery, e.g., one defines

$$L_{\Omega_0}(l) = g_m^0, \quad l = 2^{m-1} + 1, 2^m - 1,\ m \geqslant 2,$$

i.e. the values g_m^0 are extended to binary sections to the right. Using the same approximation for the kernel $k(i,l)$, i.e.

$$k(i,l) = k\bigl(i, 2^{m-1}\bigr), \quad l = 2^{m-1} + 1, 2^m - 1,\ m \geqslant 2$$

the right-hand side of (2.1) may be approximated by

$$g_n^0 \sum_{l\geqslant 1} k(i,l) L_{\Omega_0}(l) \approx g_n^0 \sum_{m\geqslant 1} 2^{m-1} k\bigl(2^{n-1}, 2^{m-1}\bigr) g_m^0,$$

where the summation on the right-hand side now runs only over the grid points of Ω_0.

The piecewise constant reconstruction $L_{\Omega_0}(l) = g_{m+1}^0$, $l = 2^{m-1} + 1, 2^m - 1$, $m \geqslant 2$ as well as a similar expression for the approximation of the kernel yields the corresponding approximation

$$g_n^0 \sum_{l\geqslant 1} k(i,l) L_{\Omega_0}(l) \approx g_n^0 \sum_{m\geqslant 1} \max\bigl\{1, 2^{m-2}\bigr\} k\bigl(2^{n-1}, 2^{m-1}\bigr) g_m^0,$$

i.e. the values g_m^0 as well as the kernel are extended to binary sections to the left.

One may even consider a convex combination of both approximations, i.e. we introduce a modified kernel of the form

$$K_0(n,m) = \bigl(\omega_0 \cdot 2^{m-1} + (1-\omega_0) \cdot \max\bigl\{1, 2^{m-2}\bigr\}\bigr) k\bigl(2^{n-1}, 2^{m-1}\bigr) \tag{2.2}$$

with $\omega_0 \in (0,1)$ in order to improve the crude approximation by piecewise constant functions.

To obtain an equivalent approximation for the loss term on the right-hand side of (1.2) we make use of the physically motivated concept of detailed-balance, which relates the gain and loss terms of kinetic models, like in the original Smoluchowski's equation. Hence, given for a binary grid point $i = 2^{n-1}$, $n = 1, 2, \ldots$, the approximation reads

$$\sum_{1\leqslant j<i} k(i-j, j) f_{i-j}(t) f_j(t) \approx \sum_{1\leqslant m<n} K_0(n-m, m) g_{n-m}^0(t) g_m^0(t)$$

with kernel $K_0(n-m, m)$ given by (2.2).

In summary, on the binary grid Ω_0 we consider Smoluchowski's equation given by the infinite system of ordinary differential equations in the form

$$\frac{\mathrm{d}g_i^0}{\mathrm{d}t}(t) = \sum_{1\leqslant j<i} K_0(i-j, j) g_{i-j}^0(t) g_j^0(t) - \sum_{j\geqslant 1} K_0(i,j) g_i^0(t) g_j^0(t) \quad (i \in \mathbb{N}), \tag{2.3}$$

where the modified kernel $K_0(i,j)$ is given by (2.2).

A nice feature of this approach is that the dynamic equation itself remains invariant when going from full integer grid $\mathbb{N}$ down to to the binary grid $\Omega_0 = \{1, 2, 4, 8, \ldots\}$ and only the kernel of Smoluchowski's equation need to be modified. As a direct consequence the ode-system (2.3) satisfies the same conservation principle, namely, mass conservation, as the original model:

Lemma 2. Assume that an interchange of the summation order on the right-hand side of (2.3) is valid, then (2.3) satisfies the conservation principle

$$\frac{\mathrm{d}}{\mathrm{d}t} \sum_{i \geqslant 1} g_i^0(t) = 0.$$

Using the same piecewise constant approximation on the higher grids Ω_l, $l \geqslant 1$, yields at each level the infinite systems of ordinary differential equations

$$\frac{\mathrm{d}g_i^l}{\mathrm{d}t}(t) = \sum_{1 \leqslant j < i} K_l(i-j, j) g_{i-j}^l(t) g_j^l(t) - \sum_{j \geqslant 1} K_l(i, j) g_i^l(t) g_j^l(t) \quad (i = 1, 2, \ldots), \tag{2.4}$$

where the index i at level l stands for the ith grid point x_i^l of Ω_l. The kernel $K_l(n, m)$ now reads $K_l(n, m) = I_l(m) k(x_n^l, x_m^l)$ with

$$I_l(m) = \begin{cases} 1, & x_m^l < 2^{l+1}, \\ 2^{p-l}, & x_m^l > 2^{l+1},\ x_m^l \notin \Omega_0, \\ \omega_l \cdot 2^{p-l} + (1 - \omega_l) \cdot 2^{p-l-1}, & x_m^l \geqslant 2^{l+1},\ x_m^l \in \Omega_0, \end{cases} \tag{2.5}$$

with $p = [\ln x_m / \ln 2]$ and $\omega_l \in (0, 1)$.

Example 3. As an example, let us look at the grid Ω_4 given by

$$\Omega_4 = \{1, \ldots, 32, 34, 36, \ldots, 62, 64, 68, 72, \ldots, 124, 128, \ldots\}.$$

The grid contains the first 32 integers, such that

$$I_4(m) = 1, \quad m \in \{1, 2, \ldots, 31\}.$$

The binary points $x_m^4 = 2^k$ with $k \geqslant 5$ are extended exactly 2^{k-4}-times to the left and 2^{k-5}-times to the right. Using a convex combination this exactly yields the expression

$$I_4(m) = \omega_l \cdot 2^{p-l} + (1 - \omega_l) \cdot 2^{p-l-1}$$

if $x_m^4 \in \Omega_0$. The other points in a binary section are extended by the same amount to the left and the right, namely, 2^{p-l}-times, where $p = [\ln x_m^4 / \ln 2]$. This gives the remaining expression in (2.5).

In order to reconstruct the values of the density function for integers $j \notin \Omega_l$, one should use a higher-order approximation than the piecewise constant approximation used in the previous section. A classical approach is to use a cubic spline interpolation based on the grid Ω_l, i.e. on the interval $[1, \infty)$ (or the truncated interval $[1, x^l_{N_l}]$, where $x^l_{N_l}$ denotes the largest grid point) one constructs the piecewise cubic polynomial.

Another possible approach from approximation theory are the so-called linear (or nonlinear) l-point subdivision schemes. In these methods one starts from the given data on the grid Ω_l and successively refines the grid by introducing new data points using a linear combination of the previous data taking from a stencil of length l, see [3]. In [4] this approach was recently extended to ENO- or WENO-reconstruction techniques introduced by Harten et al. in [9] for higher order finite-difference methods applied to hyperbolic conservation laws. We do not go in more detail and leave the question about an appropriate (polynomial) recovery for integer $j \notin \Omega_l$ for further investigations.

2.3. Adaptive time step method

As mentioned in the introduction it is shown in [12,14] that if the density values f_i are non-negative at time $t = 0$, then the same holds for larger times. This is an important feature of the continuous model which should be taken over to the discretized levels, i.e. when numerically integrating Smoluchowski's equation. Hence we require that during the integration all approximate values remain non-negative, if this is satisfied at initial time.

Numerical simulations with a standard fourth-order Runge–Kutta scheme (see 3.2) on a finite grid (either complete or refined) show that for the unbounded kernel $\tilde{k}(i, j) = i + j$ the non-negativity of the numerical approximates is violated at the tails of the density function at a certain simulation time $t = t_*$.

At what time $t = t_*$ these phenomena occur depend on

- the underlying grid (complete or refinement level),
- the truncation to a finite grid,
- as well as on the size of the time step.

In the sequel of the simulation the integration becomes unstable and yields arbitrary large oscillations. These phenomena even occur using a fourth-order TVD-Runge–Kutta scheme as given by Shu and Osher in [17].

To overcome these instabilities we force the non-negativity of the numerical approximates by continuing the time integration with a half step size. This approach yields in each numerical test performed in the sequel non-negative numerical approximations and in all problems it turned out to be sufficient to reduce the step size 2 or 3 times during the complete computation. More details and numerical results on this problem are given in section 3.2.

One should notice that the problem of negative values did not occur when using the constant kernel $\tilde{k}(i, j) = 1$ and this indicates that negative values at the tails occur due to some round-off errors close to zero.

3. Numerical examples

In the following we give some numerical results based on the method proposed in the previous section, namely, the simulation of Smoluchowski's equation formulated on a reduced number of grid points using $\omega_l = 1/2$ for $l = 0, 1, \ldots$:

(a) in section 3.1 we compare the exact solution with numerical approximates obtained from a first order explicit time integration on a finite (truncated) integer grid $\Omega_* = \{1, 2, \ldots, x_{\max}\}$ as well as the grid levels Ω_l constructed by the grid refinement technique given in 2.1 and the approximate dynamics from 2.2. The validity of a truncation to a finite grid is controlled by looking at the discrete mass conservation of the resulting systems. The main emphasis is to show that the method presented in section 2 works quite well and that the computational costs using a restricted number of grid points can be significantly reduced,

(b) in section 3.2 we apply a higher-order time integration scheme, namely a classical fourth-order Runge–Kutta method, using the same grids Ω_* and Ω_l like discussed in 3.1. The time step is enlarged by a factor of 10 compared to the previous simulations and an adaptive time step method, like discussed in 2.4 in this case is necessary to ensure non-negativity of the numerical approximates.

3.1. Explicit first order time integration

In order to integrate Smoluchowski's equation either on $\mathbb{N}$ or the grids Ω_l introduced above, it is necessary to truncate the infinite grids to finite ones. How many grid points are necessary mainly depends on the behavior of the tail of the distribution function $\{f_1, f_2, \ldots\}$. Here, the mass conservation of Smoluchowski's equation written in the form (1.2) may be used to check the validity of the truncation: if the discrete mass conservation is not satisfied satisfactorily accurate, the number of grid points in the truncated grid is obviously too small. On the other hand, if the number of grid points is too large, because the tail starts at sufficiently small cluster masses, the computational effort may be drastically reduced using a smaller number of grid points.

If a finite integer grid $\{1, 2, \ldots, N\}$ is fixed, an explicit first order time integration of (1.2) is given by

$$f_i^{n+1} = f_i^n + \Delta t \cdot \left(\sum_{1 \leqslant j < i} k(i-j, j) f_{i-j}^n f_j^n - \sum_{1 \leqslant j \leqslant N} k(i, j) f_i^n f_j^n \right)$$
$$(i = 1, 2, \ldots, N), \tag{3.1}$$

where Δt is the time step and the f_i^n's, $i = 1, 2, \ldots, N$, denote the numerical approximates at time level $t_n = n \cdot \Delta t$, $n \geqslant 0$.

The corresponding time integration scheme on a finite grid $\Omega_l = \{x_1^l, \ldots, x_{N_l}^l\}$ of level l reads

$$g_i^{l,n+1} = g_i^{l,n} + \Delta t \cdot \left(\sum_{1 \leqslant j < i} K_l(i-j, j) g_{i-j}^{l,n} g_j^{l,n} - \sum_{1 \leqslant j \leqslant N} K_l(i, j) g_i^{l,n} g_j^{l,n} \right) \tag{3.2}$$

for $i = 1, \ldots, N_l$ and kernels K_0 and K_l, $\geqslant 1$, are given by (2.2) and (2.5), respectively.

3.1.1. The constant kernel $\tilde{k}(i, j) = 1$

In the first numerical experiment we use mono-disperse initial data, i.e. $f_1 = 1$, $f_j = 0$, $j \geqslant 2$, together with the constant kernel $\tilde{k}(i, j) = 1$, which yields the exact solution given by (1.4).

We denote by Ω_* the finite integer grid given by

$$\Omega_* = \{1, 2, \ldots, 1024\}$$

and by $\Omega_5, \ldots, \Omega_8$ the finite refined grids of levels $5, \ldots, 8$, respectively, e.g., the grid Ω_6 consists of the 320 points

$$\Omega_6 = \{1, 2, \ldots, 128, 130, \ldots, 254, 256, 260, \ldots, 508, 512, 520, \ldots, 1016, 1024\},$$

i.e. on Ω_6 we use approximately one third of the grid points of Ω_*.

The solution is computed up to time $t = 10$ using 10^4 explicit time steps with step size $\Delta t = 10^{-3}$.

The exact solution at various times on the continuous interval $[0 : 32]$ is shown in figure 1, which shows the decay to the tails of the density function in dependence of the time.

Table 1 gives a comparison between the exact solution and the numerical approximates obtained from the grids $\Omega_*, \Omega_5, \ldots, \Omega_8$. The (absolute) error between the exact solution and the numerical approximates obtained on Ω_* are due to first order time integration by an explicit Euler method. On all grids considered the discrete mass conservation is fulfilled exactly, which indicates that the truncation at $x = 1024$ is reasonable.

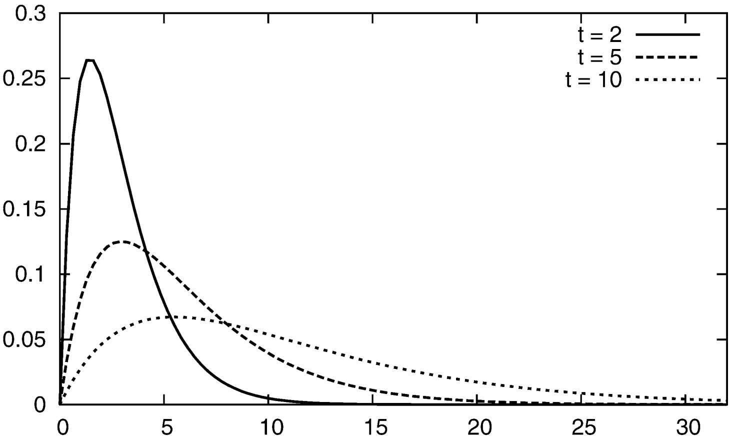

Figure 1. Exact solution at various times for mono-disperse initial data and the constant kernel $\tilde{k}(i, j) = 1$.

Table 1
Error between the exact solution and the numerical approximations obtained from the grids Ω_*, $\Omega_5, \ldots, \Omega_8$ at the binary grid points $2^0, \ldots, 2^8$ at time $t = 10$.

Point	exact	error (Ω_*)	error (Ω_5)	error (Ω_6)	error (Ω_7)	error (Ω_8)
1	0.02778	$1.99 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$	$1.99 \cdot 10^{-5}$
2	0.04630	$1.44 \cdot 10^{-5}$	$1.44 \cdot 10^{-5}$	$1.44 \cdot 10^{-5}$	$1.44 \cdot 10^{-5}$	$1.44 \cdot 10^{-5}$
4	0.06430	$3.17 \cdot 10^{-6}$	$3.23 \cdot 10^{-6}$	$3.17 \cdot 10^{-6}$	$3.17 \cdot 10^{-6}$	$3.17 \cdot 10^{-6}$
8	0.06202	$6.49 \cdot 10^{-6}$	$6.43 \cdot 10^{-6}$	$6.49 \cdot 10^{-6}$	$6.49 \cdot 10^{-6}$	$6.49 \cdot 10^{-6}$
16	0.02885	$2.79 \cdot 10^{-6}$	$2.76 \cdot 10^{-6}$	$2.79 \cdot 10^{-6}$	$2.79 \cdot 10^{-6}$	$2.79 \cdot 10^{-6}$
32	0.00312	$8.70 \cdot 10^{-7}$	$8.74 \cdot 10^{-7}$	$8.70 \cdot 10^{-7}$	$8.70 \cdot 10^{-7}$	$8.70 \cdot 10^{-7}$
64	$1.83 \cdot 10^{-5}$	$2.83 \cdot 10^{-8}$	$2.83 \cdot 10^{-8}$	$2.83 \cdot 10^{-8}$	$2.83 \cdot 10^{-8}$	$2.83 \cdot 10^{-8}$
128	$3.13 \cdot 10^{-10}$	$1.55 \cdot 10^{-12}$	$8.81 \cdot 10^{-8}$	$1.55 \cdot 10^{-12}$	$1.55 \cdot 10^{-12}$	$1.55 \cdot 10^{-12}$
256	$4.58 \cdot 10^{-20}$	$6.11 \cdot 10^{-22}$	$4.27 \cdot 10^{-10}$	$4.43 \cdot 10^{-15}$	$6.11 \cdot 10^{-22}$	$6.11 \cdot 10^{-22}$

Table 2
CPU-time in seconds used for the simulation on the grids Ω_*, $\Omega_5, \ldots, \Omega_8$ and corresponding number of grid points.

	Ω_*	Ω_5	Ω_6	Ω_7	Ω_8
CPU [sec]	79.89	1.52	4.54	13.85	38.34
nr. pts	1024	192	320	512	768

The results show that the numerical values obtained at the grid Ω_6 already coincide with the values of the full integer grid Ω_* for f_i with $i \leqslant 128$. For the numerical approximation of f_i with $i \geqslant 256$ the derivation in the tail becomes obvious on the coarse grid Ω_6, but the exact solution is less than 10^{-20} and therefore negligible. Because the computational effort is of the order $\mathrm{O}(n^2)$, where n denotes the total number of grid points, one expects that the computational costs are drastically decreased when going from Ω_* to Ω_6. This is validated by the results shown in table 2: the computational costs between Ω_* and Ω_6 differ by a factor of more than 17.

Even the results obtained on the coarse grid Ω_5 containing 192 points, which decreases the numerical effort compared to Ω_* by the significant factor of about 50, are nearly as accurate as on the full grid Ω_* up to $i = 128$.

This first example indicates that the (crude) piecewise constant approximation of the right-hand side of Smoluchowski's equation (1.2) is – at least in this case – sufficiently accurate.

3.1.2. The unbounded kernel $\tilde{k}(i, j) = i + j$

In a second example we apply a first order explicit time integration with the unbounded kernel $\tilde{k}(i, j) = i + j$ such that the exact solution with mono-disperse initial data is given by (1.5), see figure 2 for the exact solution at $t = 1, 2$ and 5, respectively.

The unbounded kernel is more difficult to simulate because clusters with higher masses are generated more quickly than in the previous example. Hence we even use larger grids for this test problem: we denote by Ω_* the full integer grid consisting of

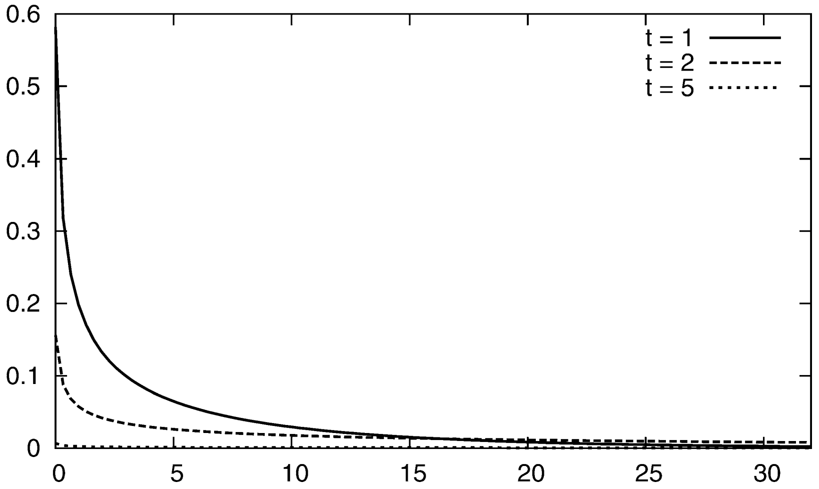

Figure 2. Exact solution at various times for mono-disperse initial data and the constant kernel $\tilde{k}(i, j) = i + j$.

Table 3
Error between the exact solution and the numerical approximations obtained from the grids $\Omega_*, \Omega_6, \dots, \Omega_8$ at the binary grid points $2^0, \dots, 2^9$ at time $t = 1$.

Point	exact	error (Ω_*)	error (Ω_6)	error (Ω_7)	error (Ω_8)
1	0.19551	$3.00 \cdot 10^{-4}$	$3.00 \cdot 10^{-4}$	$3.00 \cdot 10^{-4}$	$3.00 \cdot 10^{-4}$
2	0.13137	$1.17 \cdot 10^{-5}$	$1.18 \cdot 10^{-5}$	$1.17 \cdot 10^{-5}$	$1.17 \cdot 10^{-5}$
4	0.07907	$1.01 \cdot 10^{-4}$	$1.01 \cdot 10^{-4}$	$1.01 \cdot 10^{-4}$	$1.01 \cdot 10^{-4}$
8	0.03929	$7.99 \cdot 10^{-5}$	$7.99 \cdot 10^{-5}$	$7.99 \cdot 10^{-5}$	$7.99 \cdot 10^{-5}$
16	0.01351	$7.64 \cdot 10^{-6}$	$7.64 \cdot 10^{-6}$	$7.64 \cdot 10^{-6}$	$7.64 \cdot 10^{-6}$
32	0.00224	$1.69 \cdot 10^{-5}$	$1.69 \cdot 10^{-5}$	$1.69 \cdot 10^{-5}$	$1.69 \cdot 10^{-5}$
64	$8.68 \cdot 10^{-5}$	$2.92 \cdot 10^{-6}$	$2.92 \cdot 10^{-6}$	$2.92 \cdot 10^{-6}$	$2.92 \cdot 10^{-6}$
128	$1.84 \cdot 10^{-7}$	$1.92 \cdot 10^{-8}$	$1.92 \cdot 10^{-8}$	$1.92 \cdot 10^{-8}$	$1.92 \cdot 10^{-8}$
256	$1.17 \cdot 10^{-10}$	$3.18 \cdot 10^{-13}$	$6.00 \cdot 10^{-10}$	$3.18 \cdot 10^{-13}$	$3.18 \cdot 10^{-13}$
512	$6.64 \cdot 10^{-23}$	$3.86 \cdot 10^{-23}$	$5.67 \cdot 10^{-12}$	$1.03 \cdot 10^{-17}$	$3.86 \cdot 10^{-23}$

2048 points and the corresponding finite refined grids of levels 6, 7 and 8, denoted by Ω_6, Ω_7 and Ω_8, respectively.

The numerical solutions are computed up to the time $t = 1$ again using a time step $\Delta t = 0.001$. Table 3 gives a comparison between the exact solution and the numerical approximates computed on the grids $\Omega_*, \Omega_6, \dots, \Omega_8$, respectively. The corresponding CPU-times in seconds together with the total number of grid points are shown in table 4.

The numerical approximations on the various grids behave qualitatively like in the previous example, although the tail of the cluster distribution is shifted by about one order of magnitude to the right. The deviation between the exact solutions and the results obtained on grid Ω_* are again due to the first order time integration scheme. The

Table 4
CPU-time in seconds used for the simulation on the grids $\Omega_*, \Omega_6, \dots, \Omega_{10}$ and corresponding number of grid points.

	Ω_*	Ω_6	Ω_7	Ω_8	Ω_9	Ω_{10}
CPU [sec]	609.57	4.20	12.91	37.54	135.75	629.09
nr. pts	2048	384	640	1024	1536	2048

numerical approximates obtained on the different grids $\Omega_*, \Omega_6, \dots, \Omega_8$ coincide up to the point $i = 128$. Differences are observed again in the tail of the density function starting at $i = 256$. Concerning the computational effort table 4 shows that the CPU-times used on Ω_* and Ω_6 differ by a factor of about 145, whereas on Ω_8, where the numerical approximates coincide with the one on Ω_* even at the points $i = 256$ and $i = 512$, the computational effort is reduced still by a factor of about 16.

Remark 4. Due to the computational overhead when constructing the grid Ω_{10} the CPU-time on Ω_{10} is about 3% more high compared to Ω_*.

In summary, the results of the numerical simulations using a simple and straightforward explicit first order time integration given above demonstrate that the computational effort may be drastically reduced when using a hierarchic grid refinement and corresponding Smoluchowski-type equation like formulated in section 2.

3.2. *Higher-order time integration schemes*

Given an ordinary differential equation for $u(x)$ in the form

$$\frac{\mathrm{d}u(x)}{\mathrm{d}x} = f(x, u)$$

a standard fourth-order Runge–Kutta method with step size h is given by

$$\begin{aligned}
k_1 &= hf(x_n, u_n),\\
k_2 &= hf\left(x_n + \frac{h}{2}, u_n + \frac{k_1}{2}\right),\\
k_3 &= hf\left(x_n + \frac{h}{2}, u_n + \frac{k_2}{2}\right),\\
k_4 &= hf(x_n + h, u_n + k_3),\\
u_{n+1} &= u_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4).
\end{aligned}$$

3.2.1. *The constant kernel $\tilde{k}(i, j) = 1$*

In the following we apply the fourth-order scheme given above to the numerical integration of Smoluchowski's equation for the test problem of section 3.1.1, i.e. the

Table 5
Error between the exact solution and the numerical approximations obtained from the grids Ω_*, $\Omega_5, \ldots, \Omega_8$ at the binary grid points $2^0, \ldots, 2^8$ at time $t = 10$ using a fourth-order Runge–Kutta scheme.

Point	exact	error (Ω_*)	error (Ω_5)	error (Ω_6)	error (Ω_7)	error (Ω_8)
1	0.02778	$8.03 \cdot 10^{-13}$	$2.30 \cdot 10^{-8}$	$6.86 \cdot 10^{-13}$	$8.03 \cdot 10^{-13}$	$8.03 \cdot 10^{-13}$
2	0.04630	$1.81 \cdot 10^{-12}$	$3.89 \cdot 10^{-8}$	$2.00 \cdot 10^{-12}$	$1.81 \cdot 10^{-12}$	$1.81 \cdot 10^{-12}$
4	0.06430	$3.58 \cdot 10^{-15}$	$5.55 \cdot 10^{-8}$	$2.79 \cdot 10^{-13}$	$3.58 \cdot 10^{-15}$	$3.58 \cdot 10^{-15}$
8	0.06202	$2.20 \cdot 10^{-15}$	$5.65 \cdot 10^{-8}$	$2.75 \cdot 10^{-13}$	$2.20 \cdot 10^{-15}$	$2.20 \cdot 10^{-15}$
16	0.02885	$1.01 \cdot 10^{-15}$	$2.90 \cdot 10^{-8}$	$1.35 \cdot 10^{-13}$	$1.01 \cdot 10^{-15}$	$1.01 \cdot 10^{-15}$
32	0.00312	$3.56 \cdot 10^{-17}$	$3.73 \cdot 10^{-9}$	$1.60 \cdot 10^{-14}$	$3.56 \cdot 10^{-17}$	$3.56 \cdot 10^{-17}$
64	$1.83 \cdot 10^{-5}$	$1.50 \cdot 10^{-18}$	$2.87 \cdot 10^{-11}$	$1.34 \cdot 10^{-16}$	$1.50 \cdot 10^{-18}$	$1.50 \cdot 10^{-18}$
128	$3.13 \cdot 10^{-10}$	$1.28 \cdot 10^{-20}$	$8.84 \cdot 10^{-8}$	$1.53 \cdot 10^{-20}$	$1.28 \cdot 10^{-20}$	$1.28 \cdot 10^{-20}$
256	$4.58 \cdot 10^{-20}$	$5.47 \cdot 10^{-29}$	$4.29 \cdot 10^{-10}$	$4.47 \cdot 10^{-15}$	$5.47 \cdot 10^{-29}$	$5.47 \cdot 10^{-29}$

Table 6
CPU-time in seconds used for the simulation on the grids Ω_*, $\Omega_5, \ldots, \Omega_8$ using a fourth-order Runge–Kutta scheme and corresponding number of grid points.

	Ω_*	Ω_5	Ω_6	Ω_7	Ω_8
CPU [sec]	32.63	0.62	1.85	5.55	15.43
nr. pts	1024	192	320	512	768

constant kernel $\tilde{k}(i, j) = 1$, where we enlarge – due to the higher accuracy – the time step to $\Delta t = 0.01$.

The results of the simulations are given in tables 5 and 6. Although the time step is enlarged by a factor of 10, the error drops down significantly compared to the first order scheme, where the gain obtained on the coarse grid Ω_5 is less compared with the other ones.

Even the computational effort drops down by a factor of 2.5 on all grids, the ratio of the CPU-times between the full integer grid Ω_* and the refined grids $\Omega_5, \ldots, \Omega_8$ remains the same like in section 3.1.1. One should notice that using a fourth-order Runge–Kutta scheme one needs to evaluate the right-hand side of the system four times in each time step.

3.2.2. *The unbounded kernel* $\tilde{k}(i, j) = i + j$

In the final test problem we apply the fourth-order Runge–Kutta scheme to the unbounded kernel of section 3.1.2 and the numerical results at time $t = 1$ are given in the tables 7 and 8.

On the refined grids Ω_6, Ω_7 and Ω_8 it appears that the strict positivity of the numerical approximates starting with a time step $\Delta t = 0.01$ is violated after a certain simulation time $t < 1$. Hence we apply the heuristic algorithm of section 2.4. In par-

Table 7
Error between the exact solution and the numerical approximations obtained from the grids $\Omega_*, \Omega_6, \dots, \Omega_8$ at the binary grid points $2^0, \dots, 2^9$ at time $t = 1$.

Point	exact	error (Ω_*)	error (Ω_6)	error (Ω_7)	error (Ω_8)
1	0.19551	$2.04 \cdot 10^{-10}$	$1.15 \cdot 10^{-8}$	$1.53 \cdot 10^{-10}$	$1.81 \cdot 10^{-10}$
2	0.13137	$8.87 \cdot 10^{-10}$	$8.65 \cdot 10^{-9}$	$8.04 \cdot 10^{-10}$	$8.55 \cdot 10^{-10}$
4	0.07907	$4.54 \cdot 10^{-10}$	$4.46 \cdot 10^{-9}$	$4.77 \cdot 10^{-10}$	$4.74 \cdot 10^{-10}$
8	0.03929	$1.16 \cdot 10^{-10}$	$2.78 \cdot 10^{-9}$	$1.95 \cdot 10^{-10}$	$1.46 \cdot 10^{-10}$
16	0.01351	$7.54 \cdot 10^{-11}$	$9.45 \cdot 10^{-10}$	$3.17 \cdot 10^{-11}$	$7.44 \cdot 10^{-11}$
32	0.00224	$4.99 \cdot 10^{-11}$	$1.55 \cdot 10^{-10}$	$5.74 \cdot 10^{-11}$	$7.08 \cdot 10^{-11}$
64	$8.68 \cdot 10^{-5}$	$2.03 \cdot 10^{-13}$	$1.85 \cdot 10^{-11}$	$1.35 \cdot 10^{-11}$	$1.37 \cdot 10^{-11}$
128	$1.84 \cdot 10^{-7}$	$3.12 \cdot 10^{-12}$	$1.50 \cdot 10^{-13}$	$3.01 \cdot 10^{-13}$	$9.55 \cdot 10^{-13}$
256	$1.17 \cdot 10^{-10}$	$7.27 \cdot 10^{-15}$	$7.40 \cdot 10^{-10}$	$8.26 \cdot 10^{-18}$	$5.35 \cdot 10^{-17}$
512	$6.64 \cdot 10^{-23}$	$9.12 \cdot 10^{-25}$	$7.90 \cdot 10^{-12}$	$1.89 \cdot 10^{-17}$	$1.92 \cdot 10^{-26}$

Table 8
CPU-time in seconds used for the simulation on the grids Ω_*, $\Omega_6, \dots, \Omega_8$ and corresponding number of grid points.

	Ω_*	Ω_6	Ω_7	Ω_8
CPU [sec]	232.78	9.50	24.80	34.87
nr. pts	2048	384	640	1024

ticular, on the coarse grid Ω_6 the time step need to be decreased by the factor 2 at times $t = 0.38$ (two times) and $t = 0.39$, on Ω_7 at times $t = 0.54$, $t = 0.55$ and $t = 0.585$ and on Ω_8 at times $t = 0.75$ and $t = 0.76$. For the remaining simulation the time steps stay constant at $\Delta t = 0.00125$ on Ω_6, Ω_7 and $\Delta t = 0.0025$ on the fine grid Ω_8. On the full grid Ω_* the same occurs but at a later time $t > 1$, such that the adaptive time step method is not applied during the simulation.

Let us first discuss the results on the full grid Ω_*: like in the previous case the errors drop down significantly by several orders of magnitude compared to the first order time integration. Because the time step is enlarged by a factor of 10, besides the higher accuracy, even the computational effort is reduced by a factor of 2.5, like in the previous section.

The effect of the adaptive time step method can be observed in the results of tables 7 and 8 for the reduced grids Ω_6, Ω_7 and Ω_8. First of all, one should notice that the CPU-times on Ω_6 and Ω_7 increase by a factor of 2 compared to the first order scheme. As a consequence the gain compared to the full grid simulation is reduced by a factor of 5, but still remains at least one order of magnitude, namely, about 25 and 10 on Ω_6 and Ω_7, respectively. On the fine grid Ω_8 the CPU-time remains nearly constant, such that the gain compared to Ω_* is reduced by a factor of only 2.5 and the CPU-time is still reduced by a factor of 7.

4. Conclusion

In the present paper we proposed a deterministic scheme for Smoluchowski's equation in discrete form based on reduced grids obtained from a binary grid refinement technique. The differential equations on the reduced grid exactly coincide with the one of the full integer grid except that the kernel of Smoluchowski's equation is modified using a piecewise-constant approximation.

The numerical results given in the paper indicate that the computational effort may be drastically decreased without a significant loss in the accuracy of the numerical approximates, where the gain turns out to be more significant using a bounded kernel. The numerical instabilities, which may appear at the tails of the density function due to round-off errors, are compensated using a heuristic adaptive time step control.

The results even indicate that it is worthwhile to implement an adaptive grid refinement technique starting from the coarse complete binary grid consisting only of the binary points $i = 2^j$, $j = 1, \dots, m$ and extend the binary grid to finer grids during the computation. Such an adaptive grid refinement technique is currently under investigation [21]. A further important task will be to compare the computational effort of the deterministic scheme given here with stochastic methods proposed in the literature.

Acknowledgement

The author would like to thank Nicoletta Ianiro and the people from the Dipartimento di Metodi e Modelli Matematici per le Scienze Applicate, Università La Sapienza, Rome, Italy, for their kind hospitality, where parts of the research were done.

References

[1] D.J. Aldous, Deterministic and stochastic models for coalescence (aggregation and coagulation): A review of the mean-field theory for probabilists, Bernoulli 5 (1999) 3–48.

[2] H. Babovsky, On a Monte-Carlo scheme for Smoluchowski's coagulation equation, Monte Carlo Methods Appl. 5 (1999) 1–18.

[3] A.S. Cavaretta, W. Dahmen and C.A. Michelli, Stationary subdivision, Mem. Amer. Math. Soc. 93 (1991).

[4] A. Cohen, N. Dyn and B. Matei, Quasilinear subdivision schemes with applications to ENO interpolation, Appl. Comput. Harmon. Anal. 15 (2003) 89–116.

[5] Y.R. Domilovskiy, A.A. Lushnikov and V.N. Piskurov, Monte Carlo simulation of coagulation processes, Izv. Atmos. Ocean. Phys. 15 (1979) 129–134.

[6] A. Eibeck and W. Wagner, An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena, SIAM J. Sci. Comput. 22 (2000) 802–821.

[7] A.L. Garcia, C. Van den Broeck, M. Aertsens and R. Serneels, A Monte Carlo simulation of coagulation, Phys. A Math. Gen. 143 (1987) 535–546.

[8] D.N. Gillespie, An exact method for numerically simulating the stochastic coalescence process in a cloud, J. Atmospheric Sci. 32 (1975) 1977–1989.

[9] A. Harten, S. Osher, B. Engquist and S.R. Chakravarthy, Some results on uniformly high-order accurate essentially nonoscillatory schemes, Appl. Numer. Math. 2 (1986) 347–377.

[10] A. Kolodko and K. Sabelfeld, Stochastic particle methods for Smoluchowski coagulation equation: Variance reduction and error estimations, Preprint No. 842, Weierstraß-Institut für Angewandte Analysis und Stochastik (2003).

[11] C. Lecot and W. Wagner, A quasi-Monte Carlo scheme for Smoluchowski's coagulation equation, Math. Comp. 73 (2004) 1953–1966.

[12] F. Leyvraz and H.R. Tschudi, Singularities in the kinetics of coagulation processes, J. Phys. A Math. Gen. 14 (1981) 3389–3405.

[13] K. Liffman, A direct simulation Monte Carlo method for cluster coagulation, J. Comput. Phys. 100 (1992) 116–127.

[14] J.B. McLeod, On an infinite set of nonlinear differential equations I,II, Quart. J. Math. Oxf. II Ser. 13 (1962) 119–128, 193–205.

[15] H. Niederreiter, ed., *Monte Carlo and Quasi-Monte Carlo Methods 2002* (Springer, Berlin, 2004).

[16] K.K. Sabelfeld, S.V. Rogasinsky, A.A. Kolodko and A.I. Levykin, Stochastic algorithms for solving Smoluchowsky coagulation equation and application to aerosol growth simulation, Monte Carlo Methods Appl. 2 (1996) 41–87.

[17] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (1988) 439–471.

[18] M. v. Smoluchowski, Versuch einer mathematischen Theorie des Koagulationskinetik kolloider Lösungen, Z. Phys. Chem. 92 (1916) 129–168.

[19] M. v. Smoluchowski, Drei Vorträge über Diffusion, Brownsche Molekularbewegung und Koagulation von Kolloidteilchen, Phys. Z. 17 (1916) 557–571, 585–599.

[20] J.L. Spouge, Monte Carlo results for random coagulation, J. Colloid Interface Sci. 107 (1985) 38–43.

[21] J. Struckmeier, Adaptive grid refinement for Smoluchowski's coagulation equation, in preparation.