A STOCHASTIC APPROACH FOR SIMULATING SPATIALLY INHOMOGENEOUS COAGULATION DYNAMICS IN THE GELATION REGIME

FLAVIUS GUIAS

Abstract. We present a stochastic approach for the simulation of coagulation-diffusion dynamics in the gelation regime. The method couples the mass flow algorithm for coagulation processes with a stochastic variant of the diffusion-velocity method in a discretized framework. The simulation of the stochastic processes occurs according to an optimized implementation of the principle of grouping the possible events. A full simulation of a particle system driven by coagulation-diffusion dynamics is performed with a high degree of accuracy, which allows a qualitative and quantitative analysis of the behaviour of the system. Its performance becomes more evident especially in the gelation regime, where the computations become usually very time-consuming.

Key words. coagulation processes, diffusion-velocity method, mass flow algorithm

AMS subject classifications. 65C05, 65C35, 82C22

1. Introduction. The present paper illustrates the application of several principles and techniques which allow the development of a stochastic numerical method for a large class of partial differential evolution equations involving transport and reaction phenomena. The transport part can correspond to linear or nonlinear diffusion, convection, or fluid dynamics, while by the term “reaction” we understand in principal the so-called population balance dynamics. We consider a spectrum of particles which are characterized by one or more property parameters, e.g. the “size”. Due to the interaction with other particles in the system, this parameter may change in a con-

*Department of Mathematics, Dortmund University, Vogelpothsweg 87, 44221 Dortmund, Germany (flavius.guias@math.uni-dortmund.de).
tinuous way, but also discontinuously, for example by aggregation or breakage. The macroscopic dynamics induced by these reaction properties are in general described by certain integral equations or by infinite-dimensional systems of differential equations (if the property parameters take discrete values). This class of models includes of course the chemical reactions with a finite number of reactants.

The population balance dynamics have applications especially in chemical engineering, see [19], [20], but also in biology, astrophysics, meteorology. For details see [1] and [6]. In [23] is presented an application with relevance in polymer chemistry, where the property parameter is a two-dimensional vector: one component gives the size, while the second one is a shape parameter which measures how close or how far the surface of the polymer is from the spherical shape.

A typical example for population balance dynamics is given by coagulation processes. Particles with size parameters \( x \) and \( y \) can coalesce at rate \( K(x, y) \) in order to form a particle of size \( x + y \). Formally, the total mass in the system is conserved (since particles are neither created, nor destroyed) and by this conservation principle one arrives at the Smoluchowski equations.

If the size parameter is integer-valued, for example in the case of polymerization, where it equals the number of building blocks which form a polymer, these equations take the form

\[
\frac{d}{dt} u^1(t) = -u^1 \sum_{i=1}^{\infty} K(1, i) u^i \\
\frac{d}{dt} u^k(t) = \frac{1}{2} \sum_{i=1}^{k-1} K(i, k-i) u^i u^{k-i} - u^k \sum_{i=1}^{\infty} K(k, i) u^i \quad k = 2, 3, \ldots
\]

where \( u^k \) denotes the concentration of polymers of size \( k \). The most usual initial condition is the monodisperse initial data: \( u^1(0) = 1, u^k(0) = 0, k \geq 2 \).

The total mass \( M(t) = \sum_{k=1}^{\infty} k u^k(t) \) is formally conserved, since particles are
neither created, nor destroyed. However, for large coagulation rates, e.g. $K(i, j) \geq (ij)^q$ with $q > \frac{1}{2}$ (see [16]), one can observe the so-called *gelation phenomenon*, which means the decay of the total mass: $M(t) < M(0)$ for $t > t_{gel}$ (the gelation time). This is caused by the formation in finite time of infinitely large clusters, which are not described by the variables $u^k$.

The above equations can also be extended in order to include breakage (or fragmentation) effects. At the particle level, this means that particles of size $k$ can split into a particle of size $i$ and a particle of size $k - i$ at a rate given by $F(k, i) = F(k, k - i)$. Existence, uniqueness and mass conservation properties for coagulation-fragmentation equations can be found in [3]. Under certain growth conditions of the total fragmentation rate $F_k = \sum_{i=1}^{k-1} F(k, i)$ (which include the uniform boundedness in $k$) and for the coagulation rates, essentially if $K(i, j) \leq C_{ij}$, the system (1.1) is shown to possess at least a solution. If $K(i, j) \leq C(i + j)$ we have uniqueness and mass conservation.

In the case of a continuous size parameter the equations have the integral form

$$\frac{d}{dt} u(t, x) = \frac{1}{2} \int_{0}^{x} K(x - y, y)u(t, x - y)u(t, y)dy - \int_{0}^{\infty} K(x, y)u(t, x)u(t, y)dy$$

(1.2)

Existence, uniqueness and mass conservation properties for such an equation are derived for example in [18]. Uniqueness is shown to hold up to the gelation point. The general problem of uniqueness after the gelation time, for arbitrary gelling kernels, is still open. For further details concerning the properties of coagulation equations and of the corresponding stochastic models we indicate the surveys [1], [22] and the references within.

The numerical methods which are best suited for simulating population balance dynamics are stochastic Monte Carlo methods, like *direct simulation*, see [7], [8], [11],
where the numerical particles model real particles, or the *mass flow algorithm*, see [9],
where the numerical particles stand for the amount of mass concentrated at a given
value in the size space. Both methods are based on simulating all elementary events in
the system, e.g. all aggregation or breakage events, by performing the corresponding
transitions of a Markov jump process. A description of the two methods is given in
the *Appendix*.

The main goal of this paper is to extend the above stochastic algorithms in a
spatially inhomogeneous setting by coupling population balance dynamics with spa-
tial transport. We are able to simulate in an efficient way and with a high degree of
precision the full dynamics of the corresponding particle system, from which we can
derive the values of usual functionals such as moments of different order, concen-
trations of particles of different sizes, mass of the gel, etc. The methods used up to now
were concentrated either on the simulation of certain functionals (like moments) and
exhibited strong fluctuations around the deterministic limiting profiles, see [2].

The organization of the paper is as follows. Section 2 contains the description
of the stochastic model for coagulation-diffusion dynamics, while Section 3 presents
the techniques used for simulating this model. The numerical results are presented in
Section 4.

Our basic approach is to consider a discretization of a spatial domain into cells,
which are considered as spatially homogeneous “reactors”. Inside each cell we simulate
the population balance dynamics with a usual algorithm. In addition to this, we have
to couple these reactions with a mechanism which allows exchange of mass between
the cells. The principle is illustrated in Figure 1.

A model in this sense was considered in [21], where the behaviour of a coagulation-
diffusion model around the gelation point was analyzed for the coagulation kernel
Fig. 1. Spatial discretization of a domain into microscopic homogeneous reactors with exchange of mass

\[ K(i, j) = ij \] and equal diffusion coefficients either for all cluster sizes, or only up to a certain size, beyond which the particles are considered as imobile. In [12] the coupling of directly simulated coagulation-fragmentation dynamics with random walks was analyzed from a theoretical point of view. Conditions were given under which the corresponding deterministic equation has a unique solution and a convergence result of the family of stochastic particle models to this solution was proven. For details see also Section 2. However, this approach turns out to be extremely inefficient from the numerical point of view.

In order to be able to perform numerical simulations for models of the type described above, one needs on the one hand to improve the efficiency of the existing algorithms for population balance dynamics and on the other hand to simulate the diffusive motion of particles by a method which surpasses significantly - concerning speed and accuracy - the performance of the random walk and similar methods for nonlinear diffusion (where the jump probabilities of the particles depend also on the
local density).

The first step is to work directly with approximations of the macroscopic quantities (like densities or mass corresponding to particles of a given size, at a given place) rather than treating each particle individually. We make therefore no distinction between particles located at the same point in the size space or in the physical space. The set of possible events of the Monte Carlo simulation decreases significantly, while the state of the system continues to change according to the same particle dynamics. In this way, the total number of particles does not influence the speed of the algorithm directly, but only through the length of the time steps between two consecutive jumps. These intervals decrease at larger particle numbers, even in the case of a smaller number of possible events. This means that for advancing with the computations over a given time interval we need more iterations, which increases at the same time the accuracy of the simulation.

The second step towards higher speed is to group possible events. The basic idea is to divide the set of possible events into a number of groups, choose a group according to its probability and then select the transition event inside this group. The expected number of operations needed to compute one transition decreases in this way significantly compared to the situation when we try to consider the whole set of events at once. Here, we use the method introduced in [14] and [15] to optimize the underlying grouping principle. The optimization is done by a simple method based on binary partitions of the groups. Usually the events to be computed correspond to a list which is ordered in a natural way (e.g. by size, in the case of coagulating particles). Having a given group, we choose from all possible splittings in two consecutive parts the one which minimizes the number of expected operations needed to compute an event (to be precise: an approximation of it). We call this division a binary par-
tion of the original group. The algorithm which computes the optimized structure starts with a single group which contains all possible events and follows then the next procedure: having the current group structure, do successive binary partitions of the existing groups. If by replacing a group with the two parts the expected number of operations decreases, then we keep this binary partition and update the group structure, otherwise we reject it. We stop if the binary partition steps are rejected for all existing groups. For the present simulation, where the data has a two-dimensional spatial structure, we can perform a similar decomposition into rectangular patches. We take now binary partitions in both directions: horizontally and vertically. The procedure is described in Section 3.

The structure of the events can change very rapidly, as shown by the simulations of coagulation dynamics. Therefore, we cannot perform such a group optimization at every computation step without slowing down the algorithm significantly. If new possible events appear, we add them to one of the existing groups in a natural way (by respecting the existing order), or we may have to delete events which are not possible anymore. The renewal of the group structure takes place only periodically, when the expected number of operations for choosing events from the groups has grown by a certain factor.

In order to simulate particle diffusion, we use the method introduced in [13]. It is based on the idea of associating a velocity vector to the particles, which can be computed formally as $-\nabla u/u$, where $u$ denotes the density of the quantity which diffuses. The exact dynamics is presented in Section 2. Unlike in the random walk method, where the particles can jump in all directions, we simulate here essentially only the flux between neighbouring cells. The jumps of the particles are oriented and, for a spatial discretization step $\varepsilon$, the efficiency of the method increases significantly,
as shown in the following table, which give the CPU times for simulating a 1D-problem with different types of boundary conditions. The computations were performed on a SUN workstation using an UltraSPARC III processor at 900Mhz, with performance equivalent to a mid-range PC.

<table>
<thead>
<tr>
<th>Problem parameters</th>
<th>CPU time of the method</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>boundary conditions</td>
</tr>
<tr>
<td>0.05</td>
<td>0</td>
</tr>
<tr>
<td>0.05</td>
<td>Neumann</td>
</tr>
<tr>
<td>0.02</td>
<td>0</td>
</tr>
<tr>
<td>0.02</td>
<td>Neumann</td>
</tr>
</tbody>
</table>

The jump rates of the random walk method are proportional to the particle density. In the case of 0 boundary conditions the density decays in time, so the algorithm becomes faster as time evolves. In the case of Neumann boundary conditions this is not the case, the density tending to a constant value. This explains the discrepancy in CPU times in the case of the random walk algorithm between the problems with the two types of boundary conditions. Regarding the velocity-based method, the things are quite opposite. The jump rates are proportional to the gradient of the density (in absolute value). This quantity decays in the situation of Neumann boundary conditions, making the algorithm for this problem faster than for 0 boundary conditions. As a conclusion, the new algorithm leads to a huge improvement in efficiency compared to the random walk method, especially in the case of Neumann boundary conditions, which is the case of main interest when we try to simulate coagulation-diffusion dynamics, since we assume that the mass does not leak out of the domain.

The accuracy of the new method is also much better than the results provided
by simulating particle random walks. The results coincide practically with the results provided by deterministic finite difference schemes. In fact, for fixed \( \varepsilon \) and the number of particles tending to infinity, the method approaches this deterministic scheme -in the same way as the random walks- but with a much better convergence behaviour. Note that here we do not have to concern about the size of the time discretization steps, since by simulating all elementary events we advance always step by step, with infinitesimally small time intervals. The stability of the method is provided automatically by the stability properties of the continuous-time, spatially discrete problem which we approximate. The method can be generalized in a straightforward way to nonlinear diffusion.

The numerical simulations of coagulation-diffusion equations in 2D are realized by applying the principles discussed before. The most important feature of these simulations is that they allow computations beyond the gelation time, i.e. where we have also particles of large size, or “giant clusters”. According to the mass flow algorithm, they are removed from the system and account for the gel phase. The computations in this regime are extremely time consuming, even in the spatially homogeneous setting, and were made possible by the optimization of the stochastic algorithms as described above and by the simulation of diffusion using a new scheme, instead of random walks.

We aim to approximate the solution in \( D = [0, 1] \times [0, 1] \) of

\[
\begin{align*}
\frac{d}{dt} u^1(t, x) &= d_1 \Delta u^1 - u^1 \sum_{i=1}^{\infty} K(1, i) u^i \\
\frac{d}{dt} u^k(t, x) &= d_k \Delta u^k + \frac{1}{2} \sum_{i=1}^{k-1} K(i, k-i) u^i u^{k-i} - u^k \sum_{i=1}^{\infty} K(k, i) u^i \quad k = 2, 3, \ldots
\end{align*}
\]

(1.3)

with diffusion coefficients \( d_k \geq 0, k = 1, 2, \ldots \)
For the properties of these type of equations we refer to [4] and [17], where existence of solutions for certain classes of coagulation coefficients is proved. A stochastic approach for diffusive coagulation equations is considered for example in [5] and [12].

In our numerical simulations we consider monodisperse initial conditions, constant on a rectangular subset of \([0, 1]^2\) and 0 outside this set, scaled such that the total mass of the system (integrated over the domain) equals 1. That is, we take

\[
\begin{align*}
u_1^1(0, x) &= u_0^1 \geq 0 \text{ on } [a, b] \times [c, d] \subset [0, 1]^2, \\
u_1^1(0, x) &= 0 \text{ on } [0, 1]^2 \setminus [a, b] \times [c, d], \\
u_k^1(0, x) &= 0 \text{ for } k \geq 2 \text{ and all } x \in [0, 1]^2
\end{align*}
\]

and we consider Neumann boundary conditions: \(\partial_x u_k^1(t, x) = 0\) for all \(k \in \mathbb{N}\), \(t \geq 0\) and \(x \in \partial D\). We perform the simulations of the complete dynamics of the corresponding stochastic process with diffusion coefficients \(d_k = 1/(2k)\) and coagulation kernel \(K(i, j) = ij\) from which we derive numerical approximations of the values of several functionals, such as:

- the total mass: \(M_1(t, x) = \sum_{k=1}^{\infty} k u_k^1(t, x)\)
- the second moment: \(M_2(t, x) = \sum_{k=1}^{\infty} k^2 u_k^1(t, x)\)
- the mass of monomers: \(v_1^1(t, x) = u_1^1(t, x)\)
- the mass of particles with sizes between \(n_1\) and \(n_2\): \(v_{n_1}^{n_2}(t, x) = \sum_{k=n_1}^{n_2} k u_k^1(t, x)\)
- the mass of the gel \(G(t, x)\).

The illustration of the profiles of these quantities at different time moments allows one to make observations of the qualitative behaviour of the system. The profiles turn out to be smooth and with very small stochastic fluctuations relative to the macroscopic scale. The numerical computations allow therefore also good quantitative estimations of the properties of the system.
The most important feature of the method turns out to be the ability of computing the above functionals well beyond the gelation time, when the coagulation rates become very large, leading thus to very small time steps. For such computations we can consider up to 25000 particles per cell on a 40x40 grid. If we do not have a gelling coagulation kernel, or if we want to stop the simulations before the gelation time, we can use a more refined spatial resolution and a larger particle number in order to increase the precision. Here we may take up to $10^5$ particles per cell on a 100x100 grid.

2. The stochastic model. Consider a domain $\Omega \subset \mathbb{R}^2$ with piecewise smooth boundary and consider an $\varepsilon$-grid, i.e. a division of $\mathbb{R}^2$ into quadratic cells of size $\varepsilon$. Let $\mathcal{G}_\varepsilon$ be the set of cells which are completely included in $\Omega$ and let

$$\Omega_\varepsilon := \bigcup_{G \in \mathcal{G}_\varepsilon} G \subset \Omega$$

be the inner approximation of $\Omega$ by quadratic cells of size $\varepsilon$.

For a cell $G \in \mathcal{G}_\varepsilon$ let $\mathcal{N}(G)$ be the set of neighbouring cells from $\mathcal{G}_\varepsilon$. For a test vector $\phi = (\phi_G)_{G \in \mathcal{G}_\varepsilon}$, the discretized Laplace operator corresponding to Neumann boundary conditions is defined by

$$(\Delta_\varepsilon \phi)_G := \varepsilon^{-2} \sum_{G' \in \mathcal{N}(G)} (\phi_{G'} - \phi_G).$$

If $G$ is not a boundary cell, i.e. if all four neighbours are contained in $\mathcal{G}_\varepsilon$, this formula is the usual finite difference discretization scheme of the Laplacian. It can be checked easily that the following identity holds (partial integration):

$$\langle \Delta_\varepsilon \phi^1, \phi^2 \rangle = \langle \phi^1, \Delta_\varepsilon \phi^2 \rangle = -\frac{1}{2} \sum_{G \in \mathcal{G}} \sum_{G' \in \mathcal{N}(G)} (\phi^1_G - \phi^1_{G'}) (\phi^2_G - \phi^2_{G'})$$

for all $\phi^1, \phi^2 \in H^2$ (the space of vectors indexed by the elements of $\mathcal{G}_\varepsilon$), where $\langle \cdot, \cdot \rangle$ denotes the usual duality pairing which coincides here with the (finite dimensional)
scalar product on $H^\varepsilon$.

Let $E = \mathbb{R}_+$ or $E = \mathbb{N}_+$ be the space of particle sizes. The state space of our stochastic process is $\mathcal{D}([0, \infty), \mathcal{M}(E)^{\mathcal{G}_\varepsilon})$ i.e. the space of right continuous functions with left limits which take values in the set of Borel measures on $E$ denoted by $\mathcal{M}(E)$. That is, to every cell $G \in \mathcal{G}_\varepsilon$ and at every time $t \geq 0$ we associate a Borel measure $\mu_t^G$ on $E$ which describes the mass spectrum of the particles in cell $G$.

For a given particle size $x$ let $m_G(x)$ be the mass of particles of size $x$ in cell $G$ and $d(x)$ the diffusivity of particles of size $x$. For two adjacent cells $G, G'$ denote $\xi_{G, G'}^x = \text{sign}(m_G(x) - m_{G'}(x))$. We have thus $\mu_t^G = \sum_x m_G(x)\delta_x$ where the sum runs over all particle sizes $x$ which are present in cell $G$ at time $t$ ($\delta_x$ denotes the Dirac measure concentrated at size $x$).

By the dynamics of the mass flow algorithm (see [9]) and of the diffusion simulation method from [13], the transitions of the stochastic process can be written as follows:

\begin{align*}
\mu \to \mu + h \cdot \xi_{G, G'}^x (\delta_x^G - \delta_x^{G'}) \text{ at rate } h^{-1} \varepsilon^{-2} \frac{d(x)}{2x} |m_G(x) - m_{G'}(x)| & \quad \text{(diffusion)} \quad (2.2) \\
\mu \to \mu + h(-\delta_x^G + \delta_{x+y}^G) \text{ at rate } h^{-1} \frac{K(x, y)}{y} m_G(x)m_G(y) & \quad \text{(coagulation)} \quad (2.3)
\end{align*}

where $x, y$ run over all existent particle sizes, $G$ over all cells in $\mathcal{G}$ and $G'$ over all cells in $\mathcal{N}(G)$, that is, over all neighbouring cells of $G$ which are included in $\Omega_\varepsilon$. By $\delta_x^G$ we denote the element in $\mathcal{M}(E)^{\mathcal{G}_\varepsilon}$ which has the component corresponding to cell $G$ equal to $\delta_x$, all other components being equal to 0. $h$ is a scaling parameter: $h^{-1}$ stands for the average number of (numerical) particles per cell. The presence of the
factor $1/x$ in the transition rates (2.2) is due to the fact that the rate is proportional to densities of particles of size $x$ which is the quotient of mass and size $x$. Having a pair of adjacent cells $G$ and $G'$, the algorithm moves at the given rate an amount of $h$ units of mass of particles of size $x$ from the cell where they have a higher concentration to the cell with a lower concentration.

The mass flow algorithm used for simulating coagulation dynamics considers a passive gel. This means that particles with size exceeding a given (very large) number are removed from the system and do not interact anymore. The change of the system after such a step is obtained by replacing the term $-\delta_x^G + \delta_x^{G'}$ in the coagulation dynamics (2.3) by $-\delta_x^G$.

Taking a test vector $\phi = (\phi_G)_{G \in \mathcal{G}}$ with components $\phi_G \in C_b(E)$ (the space of bounded continuous functions on $E$), by Dynkin’s formula we obtain the following martingale characterization of the dynamics of the process $(\mu_t)_{t \geq 0}$:

$$
\langle \mu_t, \phi \rangle = \langle \mu_0, \phi \rangle + \int_0^t \sum_{\mu \to \mu'} \langle \mu - \mu', \phi \rangle \cdot r_{\mu \to \mu'} ds
$$

$$
= \int_0^t \sum_{G \in \mathcal{G}} \sum_{G' \in N(G)} h \cdot \xi_{G,G'} (\delta_{x'} - \delta_x) \cdot h^{-1} \frac{d(x)}{2x} |m_G(x) - m_{G'}(x)| + \sum_{G \in \mathcal{G}} \int_0^\infty \int_0^\infty [\phi_G(x + y) - \phi_G(x)] K(x, y) \frac{y}{h} d\mu_G^{G'} (x) d\mu_G^{G'} (y) ds + M_\phi(t)
$$

$$
= -\int_0^t \sum_{G \in \mathcal{G}} \sum_{G' \in N(G)} \varepsilon^{-2} \frac{d(x)}{2x} (\phi_G(x) - \phi_{G'}(x))(m_G(x) - m_{G'}(x)) + \sum_{G \in \mathcal{G}} \int_0^\infty \int_0^\infty [\phi_G(x + y) - \phi_G(x)] K(x, y) \frac{y}{h} d\mu_G^{G'} (x) d\mu_G^{G'} (y) ds + M_\phi(t)
$$

where $M_\phi(t)$ is a martingale with respect to the filtration generated by the process. $r_{\mu \to \mu'}$ denotes here the generic transition rate from state $\mu$ to state $\mu'$. By the partial
integration formula (2.1) we obtain the equation:

\[ \langle \mu_t, \phi \rangle = \langle \mu_0, \phi \rangle + \int_0^t \sum_{G \in G} \left[ \int_0^\infty \frac{d(x)}{x} \Delta_\varepsilon \phi_G(x) d\mu_t^G(x) + \int_0^\infty \int_0^\infty [\phi_G(x + y) - \phi_G(x)] K(x, y) \frac{d\mu_t^G(x) d\mu_t^G(y)}{y} ds + M(\phi(t)) \right] ds. \]

If we are in the case \( E = \mathbb{N}_+ \) and consider \( \phi = p^k_G \) i.e. the projection on the cell \( G \) and on the size \( k \) (smoothly cutted off at large values, in order to remain in the range of bounded continuous functions) we arrive at the following dynamics in terms of the particle densities:

\[ u^k_G(t) = u^k_G(0) + \int_0^t \left[ \frac{d(k)}{k} \Delta_\varepsilon u^k_G(s) + \frac{1}{2} \sum_{i+j=k} K(i, j) u^i_G(s) u^j_G(s) \right] ds + M_k(t). \]

Regarding the convergence properties of the stochastic algorithm for spatially inhomogeneous coagulation equations with diffusion we can state basically two things.

First, the simulation results correspond to a spatially discretized version of these equations with discretization step \( \varepsilon \). In numerical applications we cannot let \( \varepsilon \) tend to 0, but take it in the range 0.01-0.05. We have a finite number of cells of size \( \varepsilon \) where usual spatially homogeneous coagulation processes take place, while the mass exchange between the cells is realized by discrete diffusion. We can thus adapt the convergence result from [9] which states that if we let the particle number tend to infinity, the sequence of stochastic processes is relatively compact in an appropriate space (measure-valued cadlag functions) and that every weak limit point solves almost surely the coagulation equation. This holds for coagulation kernels which satisfy essentially the condition \( K(x, y) < xy \), stating nothing about the possible uniqueness of the limits. For the precise formulation of the result see [9]. In our situation we will obtain convergence towards solutions of a discretized system of spatially homogeneous
coagulation reactors, which are coupled by discrete diffusion.

Secondly, we analyze the convergence properties for $\varepsilon \to 0$ either of the discrete deterministic system which is obtained by letting first the particle number tending to $\infty$ as above, or directly from the dynamics of the stochastic process. In the latter case we have to correlate the convergence speed to 0 of $\varepsilon$ with the convergence speed of the particle number to $\infty$. Here, we can adapt the result from [12], which states essentially that for bounded coagulation kernels and on time intervals where the deterministic coagulation-diffusion equation has a unique solution (it is given also a proof for the existence of such intervals) we have stochastic convergence of the corresponding $L^2$-norms. In [12] the diffusion was modeled by random walks and the coagulation dynamics by a direct simulation method. The boundedness condition of the coagulation kernels which is required means in fact (from the numerical point of view) that the result applies for general kernels on time intervals on which we do not yet have very large clusters, basically before the gelation time.

To equation 2.4 we can apply directly the results from [12] in order to formulate the following statement concerning the convergence of the stochastic algorithm. Define the piecewise constant functions $u^k_h(t, x)$ by $u^k_h(t, x) = u^k_G(t)$ for $x \in G$. Since we will work in an $L^2$-setting, the values on the cell boundaries have no importance.

**Theorem 2.1.** Assume that $d(\cdot)$ and $K(\cdot, \cdot)$ are uniformly bounded and let $[0, T]$ be a time interval where equation (1.3) has a unique mild (semigroup) solution $(u^k)$. If the number of particles tends to infinity and $\varepsilon \to 0$ such that $h\varepsilon^{-4+\delta} \to 0$ for some (small) parameter $\delta > 0$, and if the initial conditions converge:

$$\sum_{k=1}^{\infty} k^q\|u^k_h(0) - u^k(0)\|_{L^\infty} \to 0, \text{ in probability for some } q > 0,$$

then we have the convergence:

$$\sup_{t \in [0, T]} \sum_{k=1}^{\infty} \|u^k_h(t) - u^k(t)\|_{L^2} \to 0 \text{ in probability.}$$

The proof uses techniques based on the Trotter-Kato theorem on the convergence
of discrete semigroups and Doob-like inequalities for convolution-type integrals involving the martingale terms. The condition $h\varepsilon^{-4+\delta} \to 0$ is a sufficient condition which indicates that the needed particle number per cell may be very large in order to obtain good approximation properties. In our simulations with $\varepsilon = 1/40$ and $h^{-1} = 25000$ we have $h\varepsilon^{-4} = 102.4$, but the results are nevertheless very good, so it is to expect that the condition which couples the convergence of $h$ and $\varepsilon$ can be significantly weakened. In fact, the condition needed in [12] in order to obtain a similar convergence result up to the stopping time $\tau_M = \inf\{t > 0 : \sum_k k^d \|u_k^h(t)\|_{L^\infty} \geq M\} \land T$ for an arbitrarily large constant $M > 0$ is only $h\varepsilon^{-2} \to 0$. The value of this quantity for the parameters used in our simulations is 0.064, so in fact it is a small one. The stronger condition used in the formulation of the theorem was needed in order to show that the stopping time $\tau_M$ coincides with probability tending to 1 with the right end of the interval $[0,T]$. 

3. Description of the algorithm. The algorithm used for simulating the solution of (1.3) combines the mass flow algorithm introduced in [9] implemented with the grouping procedure described in [15], the algorithm for particle diffusion described in [13] and a two-dimensional variant of the grouping procedure which performs an adaptive partition of the spatial domain domain into rectangular patches.

We describe next the basic steps of the algorithm. The cells are grouped into rectangular patches using a procedure which will be described below. In each cell we have possible diffusion or coagulation events (first and second coagulating particle), so we consider in every cell a grouping of the particles adapted cf. [15] to the probability distribution of each of these event classes. We have therefore three different groupings. The next event is then chosen as follows: select a patch according to its probability (that the next event takes place in a cell from the given patch). Select then the cell
inside this patch in a similar fashion. Choose further the next event which occurs in 
this cell as in the spatially homogeneous setting: select first a group of events and 
then the event inside this group, according to its probability.

The idea of the two-dimensional grouping procedure is similar to that in one 
dimension, but here we apply it in two directions: horizontally and vertically. We try 
to obtain a partition of the domain into rectangular patches \( \mathcal{G} = \{G_1, \ldots G_m\} \) by a 
binary partition algorithm which is performed as long as the quantity

\[
M_\mathcal{G} = m + \sum_{i=1}^{m} P_i n_i
\]
decreases. This quantity is an approximation of the expected number of operations 
used by the inverse transform method. \( P_i \) is the probability that the next event 
(coagulation event or diffusion event) will take place in a cell from patch \( G_i \), while 
\( n_i \) is the number of cells in patch \( G_i \). According to the analysis in [15], the inverse 
transform method is more appropriate than the acceptance-rejection method in the 
case when we have large variations in the probabilities of the events and when the 
events with high probabilities are sorted at the beginning of the probability table 
which has to be simulated. The occurrence of events with high probabilities occurs in 
the context of coagulation dynamics especially in the gelation regime.

By a horizontal binary partition of a rectangular patch \( G \) we mean a division 
into a left rectangle \( G_1 \) and a right rectangle \( G_2 \), such that \( M_\mathcal{G} \) is minimal among all 
such divisions. The vertical binary partition is defined analogously.

The partition algorithm can be described now as follows:

- Suppose we have a group structure \( \mathcal{G} = \{G_1, \ldots G_m\} \). For \( i = 1 \) to \( m \) do the 
  following steps:
    - perform a horizontal binary partition of the group \( G_i \).
    - if the replacing of \( G_i \) by the two new groups leads to a smaller value of
$M_G$, then keep this binary partition, otherwise reject it.

– if the binary partition of $G_i$ was accepted, update the value of $M_G$.

• perform the same steps as above in the vertical direction

• If all binary partition steps (horizontal and vertical) were rejected, then

STOP, otherwise perform another cycle of similar operations with the new group structure.

The numerical experience shows that in this two-dimensional case it is sufficient to recompute the patch structure at larger time intervals, which belong already to a macroscopic time scale. Therefore, we can afford to sort the patches into the array where their coordinates are stored in the decreasing order of their probabilities. This reduces the expected number of operations needed for choosing the patch where the next event will take place. At the same time, it contrasts to the grouping principle at coagulation dynamics, where due to the fast changes in the particle structure we need to recompute the group structure quite often, keeping always track of the natural ordering corresponding to the particle sizes.

Figures 2-4 illustrate the application of this adaptive partition principle. Each pair of pictures is a grayscale representation of the cell probabilities (left) and patch probabilities (right). Bright values indicate higher probabilities, dark values lower probabilities.

Figure 2 presents a situation where the coagulation events are dominating due to the higher mass and the larger cluster sizes concentrated in the middle of the domain. In Figure 3 we have the opposite situation: we are in the early stages of the simulation, when the mass localized in the middle is still almost constant due to the structure of the initial data and is surrounded by a diffusion front. The dynamics are dominated by the motion of particles and not by coagulation. Finally, in Figure
4 we present a snapshot of the probabilities which correspond to a problem where the initial concentration of monomers was taken uniformly equal to 1 over the whole domain. Due to coagulation of particles we have nevertheless small variations of the densities, which lead to a slight diffusion phenomenon. This yields the noisy and irregular pattern of the cell probabilities.

We can remark that in all three situations the patches are well adapted to the structure of the cell probabilities, being refined in the places where the cell probabilities are higher (as in the first two examples) and delivering an almost uniform pattern in the last example, where the cell probabilities are irregularly but quite uniformly distributed across the domain.
We point out that the shape of the patches has no importance at all, what counts is only the product between their area and their probabilities. For this reason, the small patches which appear have higher probabilities, while the cells with lower probabilities are grouped together in larger patches. Elongated and thin shapes of some of the patches (for example in Figure 2) do not create trouble in our numerical algorithms, as they would do for example in finite element methods.

The use of an optimized structure of the patches leads to an efficiency gain of
about 10% compared to the use of a regular structure of the patches, as illustrated in Figure 5. The initial data considered here had a symmetrical structure, so it is to expect that the advantage of the adaptive patch pattern should be even more pregnant in general situations.

We note that in the early stages of the simulation the CPU time shows a relatively steep increase due to discontinuity of the initial data and to the high gradients at early time moments. For a short time interval we have in fact a free boundary problem, a moving diffusion front. The diffusion rates in this region are very high as illustrated also in Figure 3. But, after the mass profile reaches a certain stabilization, the gradients become smaller, while the coagulation rates are still not very large. Thus, the computations in this regime are thus much faster. If the initial data would not have such large gradients due to the discontinuity, this would be the normal speed of the computations. When approaching the gelation point, we can remark a steep increase of the CPU time, much stronger than in the early stages. In this setting it takes about 1 hour to compute the dynamics up to the gelation time, while the computations beyond this point (on a similar time scale of the problem) need several days.

The data presented above are based again on simulations performed on a SUN workstation with an UltraSPARC III processor at 900Mhz.

4. Numerical results. We will present next simulation results performed on a machine with an AMD Opteron 250 processor at 2393 Mhz, which is over two times faster than the system used for the previous test cases. This allows for the use of a more refined spatial resolution, as well as a larger number of numerical particles. For the simulations presented here, we considered a 40x40 grid and a number of 25000 particles per cell.
4.1. Time evolution of the total mass. Figure 6 shows the time evolution of the total mass before the gelation time. The profiles are smooth (with stochastic fluctuations of a very low magnitude relative to the considered macroscopic scale) and have the typical shape imposed by the Neumann boundary conditions.

![Fig. 6. Time evolution of the total mass](image)

4.2. Time evolution of the second moment. Figure 7 shows the blow up of the second moment which starts in the region with more total mass, i.e. in the middle of the domain. The value of the second moment increases sharply as we approach the (numerical) gelation time, defined as the time of the first appearance of the gel phase.

4.3. Time evolution of the mass concentrated at different sizes. Figure 8 shows two snapshots of $v_{50}^{100}$, i.e. of the mass of the particles with sizes in the interval [50, 100]. In the continuous deterministic equations these quantities are positive at
all positive times, even if the initial condition is monodisperse, but the numerical algorithm can capture only the values which are in the range of its resolution, namely the mass of one numerical particle. Otherwise, the values of such functionals are computed as 0, that is, we don’t have (numerical) particles in this range.

Fig. 7. Time evolution of the second moment

Fig. 8. Time evolution of the mass $m_{[50, 100]} = v_{50}^{100}$
Figure 9 shows the time evolution of the mass of monomers. We can notice the change of the profile of the surface from concave to convex passing through an intermediate state, corresponding to the profile at $t = 0.25$. We can remark also the various symmetric patterns which appear in this context. This behaviour can be explained by the interaction of coagulation and diffusion dynamics. In the early stages of the time evolution diffusion is dominating and the mass (in particular the monomers) spreads out across the domain. Nevertheless, the total mass maintains a peak in the center of the domain, as can be seen in Figure 6. Later on, the coagulation processes are accelerated in the region which concentrates a larger amount of mass, with the consequence that the monomers are consumed more rapidly than elsewhere by coagulating with larger particles. Since diffusion means transport from higher concentrations to lower concentrations, when the profile of the mass of the monomers
becomes convex, the monomers tend to move into the center of the domain, with larger total mass, but smaller mass of monomers.

The answer to the question which of the two competing processes prevails, depends of course on the problem parameters. For the values of coagulation and diffusion coefficients mentioned at the beginning of the chapter, we observed a sort of equilibrium of these two phenomena. The shape of the profile from Figure 9 at $t = 0.30$ stays almost identical, having only a slight pulsatory behaviour, and moves down uniformly in time from the value of $0.7$ to the value of about $0.525$ at the gelation time and to lower values afterwards. This can be best observed on a movie or on a tight sequence of successive snapshots.

4.4. Computations beyond the gelation time. The mass flow algorithm used here to simulate the coagulation dynamics computed the first appearance of the gel phase at $t_{gel} = 0.575$. Figure 10 illustrates clearly how the gelation starts in the region with larger total mass. Here we have also an interaction of the diffusive and coagulation dynamics. The particles on the edge of the “volcano crater” at $t = 0.60$ tend to move inside the “hole” in the total mass profile produced by the appearance of the gel at the same time when gelation starts in this edge region. This interaction leads to the irregular patterns observed at later times in the region where gelation takes place with an area which increases with the passage of time.

We also observe that between the region of the mass profile where no gelation takes place, delimited by a sharp edge, and the middle region with irregular oscillations we have a sudden decay of the mass profile, a “ditch”, which corresponds in some sense to the snapshot at $t = 0.60$ (first picture). Figure 11 represents a zoom into the gelling region. In this circular layer, which belongs to the gelling region, the gelation is maximal and the total mass is locally on a lower level that in its neighbourhood,
Figure 10 shows the values of the second moment in the gelling regime. In the region where we have the gel phase the second moment strongly oscillates between very large values. The particles which are removed from the system by surpassing the gel threshold lead to a decrease of the second moment, which then increases again by the appearance of other large clusters. In the limit, the second moment in this region inside as well as outside.
would be infinite. Again, we observe the existence of a “gelation front”, described by the circular region characterized by large values of the second moments. This region corresponds to the edge in the profile of the total mass from Figures 10 or 11 where gelation is about to start.

Figure 13 shows two snapshots of the mass of the gel. Due to the symmetric structure of the mass profile, the gel appears first in the middle of the domain and then spreads out radially, increasing in the same time in mass.
4.5. Linear versus nonlinear diffusion. The method presented in [13] for simulating linear diffusion can be adapted without problems in order to model nonlinear diffusive dynamics. The jump rates are modified by taking additionally the product with the particle density (the mass divided by the corresponding size). The
limit equations will have then the terms $d_k \Delta u^k(t, x)$ replaced by $d_k \text{div}(u^k \nabla u^k)$. In the early stages the dynamics are quite different, due to the infinite propagation speed of linear diffusion versus the finite propagation speed of nonlinear diffusion. This is illustrated by the sequences of pictures presented in Figures 14 in the linear case and in 15 in the nonlinear case. In the former case we observe a smooth diffusion front, due to the fact that the mass spreads out instantaneously across the whole domain,
while in the latter one can follow the motion of a sharply distinguished free boundary until the moment when the mass becomes supported in the whole domain.

At later stages, as illustrated in the case of nonlinear diffusion by the snapshot taken at $t = 0.20$, the total mass profile approaches an equilibrium state similar to that in the situation of linear diffusion. The behaviour in the gelation regime is qualitatively the same in both situations.

5. Conclusions. In this paper we make use of several principles which allow an efficient stochastic numerical simulation of various nonlinear dynamics. They are illustrated by applications to coagulation-diffusion equations, but can be extended in order to be able to treat a wider class of problems. Typical applications are transport problems (including linear and nonlinear diffusion or fluid dynamics) which are coupled with population-balance dynamics. By this we mean a population which is indexed by a parameter, for example the “size”, which may change discontinuously due to interactions with other individuals (collision, breakage, etc).

The basic idea of our method is to simulate the microscopic dynamics of the problem by Markov jump processes, based on a system with a large number of numerical particles. Every elementary event is simulated individually and the time progression is realized with small time steps.

We make use of an idea which improves the implementation of Monte Carlo particle methods. In order to minimize the computational effort for computing the next event, we divide the possible events in a natural way into groups. The algorithm first chooses one of the groups according to its probability and subsequently the next event among the elements of this group. This method is already known in the literature, but without further thoughts about the structure of the groups, which are usually chosen by a given scale which characterizes the set of events. In the ideal situation
the group structure should be chosen adaptively, such that the expected number of operations for computing the next event is minimal. However, this problem is too complex and the structure of the possible events may change very rapidly (as in the case of coagulation), so we cannot optimize the structure after every computational step. Instead, we use instead a simple and fast method which leads to significantly better results than using unoptimized partitions. We start with a single group which contains all possible events and then perform successive binary partitions of the existing groups. They are accepted if the expected value of the total number of operations decreases, being rejected if we do not obtain such an improvement. We stop if we are not able to obtain any further improvement.

This principle can be applied also for higher dimensional data structures, if we have a spatial domain which is divided into discrete cells. In order to select the cell where the next event (reaction event or motion of a particle) will take place, we perform a similar adaptive partition of the set of cells into patches or blocks, by taking binary partitions as above in all spatial directions. We select first the group according to its probability and then the corresponding cell inside this group. The population spectrum inside this cell is structured again in such groups as mentioned above, so, after having chosen the cell, we select the next event according to this group structure and to the dynamics of the process.

The diffusion is simulated by means of velocity-driven trajectories of the particles. The motion is always oriented according to the velocity vector which is attached to every cell of the spatial discretization. Its value is determined by the local structure of the particle system. This method turns out to be overwhelmingly faster and preciser than using independent random walks.

These principles are applied here to two-dimensional coagulation-diffusion equa-
tions, where they give good results. In addition, the flexibility of the method allows applications to a various range of problems which couple population balance dynamics with spatial transport of all sorts. Such problems arise in biology, chemistry, astrophysics, meteorology, etc. We may simulate without problems the same type of dynamics in random media, where the transport and reaction parameters may vary randomly across the spatial locations. The simulation of the motion of free boundaries can also be done, as shown in the examples of nonlinear diffusive dynamics. Our method can be used even for the simulation of fluid dynamics driven by the Euler or Navier-Stokes equations, after some adaptations. This is one of the important aspects which have to be followed in the future.

A possibility to further decrease the runtime of the method can be its parallelization. This can be done in a very natural way. In order to obtain good and accurate results, one has either to choose a larger number of particles, or perform several simulations with less particles and average over the results. From the point of view of efficiency the two approaches behave similar. But with the parallelization, which means simply the simultaneous run of independent simulations with a given number of particles and a subsequent averaging over the results, we may either improve the computation time (by using less particles in every simulation) or the accuracy, by taking averages over several simulations based on the given number of particles.

Appendix A. Stochastic algorithms for coagulation dynamics. The mass flow algorithm presented in [9] is based on another formulation of the equation system (1.1). Let \( m_k(t) = k u^k(t) \) denote the total mass of \( k \)-mers. The idea is to work with these quantities, rather than with concentrations. Consider as test function an element \( \phi \in \ell^2 \) with a finite number of nonzero components. Multiplying the \( k \)-th equation with \( k \phi^k \) and summing up all equations, after some manipulations one obtains cf. [9]
the following weak formulation:

\begin{equation}
\frac{d}{dt} \langle m, \phi \rangle = \sum_{k=1}^{\infty} \sum_{i=1}^{\infty} \phi(k+i) - \phi(k) \frac{K(k,i)}{i} m_k m_i \tag{A.1}
\end{equation}

where \( \langle m, \phi \rangle = \sum_k m_k \phi^k \) denotes the \( \ell^2 \) scalar product. The above formulation suggests a flow of mass from size \( k \) to size \( k + i \) (take for example \( \phi(j) = \delta_{kj} \) and then \( \phi(j) = \delta_{k+i,j}, \ j = 1, 2, \ldots \)). Note that there is no upper bound for this flow.

For gelling kernels this means that mass can be transported to infinity in finite time. From a numerical point of view this phenomenon has to be avoided. One considers thus a very large upper bound \( B \) and all mass which is transported beyond this value is eliminated from the system, accounting for the mass of the gel phase.

In the continuous version the solutions are measure-valued and satisfy for appropriate test functions \( \phi \) the weak formulation

\begin{equation}
\frac{d}{dt} (\mu_t, \phi) = \int_0^{\infty} \int_0^{\infty} [\phi(x+y) - \phi(x)] \frac{K(x,y)}{y} d\mu_t(x) d\mu_t(y). \tag{A.2}
\end{equation}

In the case of discrete measures \( \mu = \sum_{i=1}^{n} m_i \delta_{x_i} \) and a multiplicative form of the coagulation kernel \( K(x,y) = r(x)r(y) \), the weak formulation version reads:

\begin{equation}
\frac{d}{dt} (\mu_t, \phi) = \sum_{k=1}^{\infty} \sum_{i=1}^{\infty} [\phi(x_k + x_i) - \phi(x_k)] \cdot r(x_k) m_k \cdot \frac{r(x_i)}{x_i} m_i. \tag{A.3}
\end{equation}

The \textit{mass flow algorithm} (MFA) can be described in the general setting as follows. Consider \( N \) particles, each of mass \( 1/N \), distributed in the size space. Let \( m_i \) be the mass concentrated at the size \( x_i \) and let \( K(x,y) \leq r(x)r(y) \). Equation (A.3) suggests an iteration cycle consisting in the following steps.

1. Wait an exponentially distributed time step with parameter

\[ N \sum_k r(x_k) m_k \sum_i \frac{r(x_i)}{x_i} m_i. \]
2. Choose the first coagulating particle of size $x_k$ according to the distribution

$$
\frac{r(x_k)m_k}{\sum_j r(x_j)m_j}
$$

3. Choose independently the second coagulating particle of size $x_i$ according to the distribution

$$
\frac{r(x_i)x_i}{\sum_j r(x_j)x_jm_j}
$$

4. With probability

$$
\frac{K(x_k,x_i)}{r(x_k)r(x_i)}
$$

remove a particle from location $x_k$. If $x_k + x_i < B$ then add a particle at location $x_k + x_i$, otherwise increase the mass of the gel by $1/N$.

The direct simulation algorithm (DSA) considers particles which have attached a size $x_j$. These consist of aggregations of a total number $N$ of monomers. The algorithm is similar with that presented above, with the difference that the coagulating particles behave now symmetrically. One has to compute their sizes $x_k$ and $x_i$ by repeating two times step 2, while in the distributions which appear, the masses $m_j$ are replaced by the corresponding concentrations $u_j$ of $x_j$-mers. Note that in this situation one has no clear definition of the gel phase. Often it is considered as the biggest particle, if its size is of magnitude $O(N)$. The maximal possible size is therefore equal to $N$, while in the MFA the constant $B$ which is the threshold size to the gel phase is usually taken much larger, for example $100 \cdot N$.

In [9] it is shown that the MFA converges to the solutions of (1.2), even after the gelation time. The gel computed by the MFA corresponds thus to value given
by the Smoluchowski equations. It is a passive gel, since the gel phase does not interact anymore with the existing particles. The process computed by the DSA converges before the gelation time to the same limit, but after the gelation time it may converge to a different limit than the system (1.2), as shown in [10] for a large class of coagulation kernels. In this situation we have an active gel, since no particles are removed from the system and the large particles continue to interact.

REFERENCES


[23] C. G. WELLS, M. KRAFT: Direct simulation and mass flow stochastic algorithms to solve a sintering-coagulation equation, Monte Carlo Meth.Appl., 11(2) (178-197), 2005