Cambridge Centre for Computational Chemical Engineering

University of Cambridge

Department of Chemical Engineering

Preprint ISSN 1473 – 4273

An Efficient Stochastic Algorithm For Simulating Nano-particle Dynamics

Mike Goodson, Markus Kraft[1]

submitted: 13th July 2002

[1] Department of Chemical Engineering
University of Cambridge
Pembroke Street
Cambridge CB2 3RA
UK
E-Mail: markus_kraft@cheng.cam.ac.uk

Preprint No. 3

c4e

Key words and phrases. Stochastic modelling, population balance equation, nano-particle dynamics, coalescence.

Edited by
Cambridge Centre for Computational Chemical Engineering
Department of Chemical Engineering
University of Cambridge
Cambridge CB2 3RA
United Kingdom.

Fax: + 44 (0)1223 334796
E-Mail: c4e@cheng.cam.ac.uk
World Wide Web: http://www.cheng.cam.ac.uk/c4e/

Abstract

Nano-particles produced at high temperatures often undergo rapid coalescence with complex associated rate laws. In this paper we develop and study the numerical properties of a stochastic algorithm for the modelling of nano-particle dynamics in the free molecular regime. Following the work in (A. Eibeck and W. Wagner, SIAM J. Sci Comput. 22(3):802–821,2000), we model the system as a Markov process and introduce a new majorant kernel that enables us to extend the use of fictitious jumps to a wider class of problems. We also include a source term. We study the convergence properties of the algorithm; the systematic error decreases as $\frac{1}{N}$. We then examine the efficiency of the algorithm by comparing it to a direct simulation Monte Carlo (DSMC) algorithm, that described by Gillespie (D.T. Gillespie, J. Atmos. Sci. 32(10):1977-1989, 1975). We also compare the efficiency of our new majorant with the linear majorant used in (A. Eibeck and W. Wagner, Ann. Appl. Prob. 11(4):1137–1165,2001). The improved stochastic algorithm compares very favourably with the DSMC algorithm. The CPU time required for simulation is orders of magnitude lower, and for low particle numbers, the CPU time increases linearly with particle number, rather than as the square of the particle number (as with the DSMC algorithm). Our majorant kernel enables us to simulate solutions to a wider variety of problems than the linear majorant and also gives a significant gain in efficiency. The results of this report promise excellent efficiency of simulation of problems such as soot formation and synthesis of fumed silica, and also for extension to a more general class of problems in which the population balance equation occurs.

Contents

1 Introduction

In this paper we examine the following equation:

$$\frac{\partial}{\partial t}c(t,x) = \frac{1}{2}\sum_{y=1}^{x-1} K(x-y,y)c(t,x-y)c(t,y) - \sum_{y=1}^{\infty} K(x,y)c(t,x)c(t,y) + Ic^{in}(x), \tag{1.1}$$

with initial condition

$$c(0,x) = c_0(x) \geq 0 \tag{1.2}$$

where $K(x,y)$ is given by

$$K(x,y) = \left(\frac{3}{4\pi}\right)^{\frac{2}{3}} \left(\frac{8\pi kT}{\rho}\right)^{\frac{1}{2}} \left(\frac{m_1}{\rho}\right)^{\frac{1}{6}} \left(\frac{1}{x}+\frac{1}{y}\right)^{\frac{1}{2}} \left(x^{\frac{1}{D_F}} + y^{\frac{1}{D_F}}\right)^2 \tag{1.3}$$

and

$$c^{in}(x) = \delta(x-1) \tag{1.4}$$

Equation (1.1) is the discrete formulation of the **Smoluchowski coagulation equation** [18] with the inclusion of a source term, which, along with equations (1.2)–(1.4), is applicable to several areas in the field of nano-particle technology, including soot formation [4] and ceramic formation by flame aerosol technology [5].

The first term on the RHS of equation (1.1) describes the rate of increase of $c(t,x)$, the concentration of particles of size x, due to coagulation of two smaller particles. The second term describes the rate of decrease of $c(t,x)$ due to coagulation of a particle of size x with another particle. $K(x,y)$ is the coagulation rate kernel, which describes the rate of coagulation of particles of size x and y. The third term on the RHS of equation (1.1) is a source term, describing the rate of production of particles of size x, which is linked to gas-phase reaction rates.

The size of a particle, x, is taken to mean the mass (or volume) of a particle divided by the mass (or volume) of the smallest possible particle. Thus, the size, x, can be considered as the number of monomers contained in a particle, and takes integer values.

Equations (1.1)–(1.4) describe the formation and subsequent Brownian coagulation of particles in the free molecular regime. D_F is a fractal dimension typically in the range 1.7 to 2.5 for non-spherical particles. When spherical particles are considered, such as in the case of soot formation, we use $D_F = 3$.

Flame aerosol technology is an established industrial process, but the fundamentals of the process are not yet well understood. It is a difficult process to study due to the fact that chemical reaction and particle growth take place extremely fast in typical process conditions. The characteristic times for reaction and particle growth (under a second) are smaller than the typical residence time in the reactor, and it is this, coupled with high process temperatures, that makes it difficult to collect representative samples for particle characterisation and model development.

Nevertheless, models have been proposed, which make use of the **population balance equation** (pbe). The general form of the population balance equation originates from the statistical mechanical formulations of Hulbert and Katz [10] and is specifically described in [15]. Coagulation of particles was, however described by Smoluchowski [18] before then. The models proposed for SiO_2 formation include complicated coagulation rate terms and therefore cannot be solved except by numerical methods. Previously, restrictions have been introduced into the models, to allow solution by the method of moments. These have included assuming the particle size distribution to be monodisperse [13] or log-normal [19], or being restricted to a particular value of the coagulation rate kernel [3]. It is often assumed that particle growth rate merely depends on coagulation, and that with a homogeneous collision kernel (see section 2.2), self-preserving size distributions are obtained [12]. However, this is not true when there is a significant rate of formation of monomer particles [20]. In this case, bulk properties (number density, mean particle size) can be well predicted, but polydispersity effects are predicted poorly.

Alternatively, sectional techniques have been proposed for batch [9] and continuous [8] systems. These techniques go some way towards modelling particle size distributions, but become prohibitively computationally expensive when extended to multiple dimensions. It is with this in mind that a new solution method is sought for the **population balance equation**.

The **purpose of this paper** is to formulate a solution method for the model describing the formation and coagulation of fumed silica in the process of flame synthesis. The model will take the form of the Smoluchowski coagulation equation and will include a source term, modelling monomer formation. There will be no assumptions regarding the form of the particle size distribution, rather, the initial conditions will be formulated, and the time evolution of the particle size distribution (PSD) will be examined.

To solve Equation (1.1), we introduce a **stochastic particle system**. Stochastic methods have been used before to simulate solutions to the population balance equation. These have taken the form of direct simulation methods [7, 14], constant number simulation [17], Nanbu type simulation [16] and simulation using fictitious jumps [1]. These methods have only rarely included a source term [7]. We formulate the simulation algorithm to model this extra term, and include it in the general solution algorithm. To simulate the coagulation steps, we introduce a new, efficient **majorant kernel** and use **fictitious jumps** following the ideas in [1]. Although a linear majorant kernel has been proposed in [2] for the case of spherical particles, our new majorant kernel is applicable to a wider variety of problems. Combining these techniques leads to a highly efficient simulation procedure, which will then be extendable to other situations. To give an idea of its efficiency, we compare it to direct simulation Monte Carlo (DSMC) methods as outlined by Ramkrishna [14] and proposed as an algorithm by Gillespie [6, 7] among others, and also to the method in [1] using the linear majorant.

This paper is **organised as follows**: In **Section 2** we describe the formulation of the stochastic model. **Section 2.1** introduces the stochastic particle system and the notion of fictitious jumps for simulation of coagulation. **Section 2.2** details where the use of the linear majorant kernel breaks down, and the formulation of a new, efficient majorant kernel. **Section 2.3** discusses an efficient method of generating the required distribution functions. **Section 2.4** describes how the source term is simulated and **Section 2.5** describes in detail the stochastic algorithm used to simulate solutions to Equation (1.1).

In **Section 3** we present the results of numerical studies on our solution algorithm. **Section 3.1** contains sample results in the form of particle size distributions and moments of the PSD. **Section 3.2** studies the convergence properties of the algorithm and in **Section 3.3** the efficiency of the algorithm is examined. This is done by looking at the simulation times in comparison to both the DSMC algorithm, proposed by Gillespie [7] and to simulation using the linear majorant kernel. We also look at the acceptance efficiency of the linear majorant kernel and our new majorant.

Finally, in **Section 4** we present our conclusions.

2 The Stochastic Model

2.1 Markov Process With Fictitious Jumps

To simulate the coagulation process in Equation (1.1), we follow the method introduced in [1], using a **Markov Process** with **fictitious jumps**.

For a detailed description of the method, we refer to section 2 of [1], and give a brief summary of the motivation and method here.

Consider a stochastic particle system

$$x_i(t), \quad i = 1, 2, \ldots, n(t), \qquad t \geq 0, \tag{2.1}$$

where $x_i(t)$ represents the size of particle i at time t, and $n(t)$ is the (time varying) number of particles in the system. Recall that the particle sizes only take integer values. The initial system is chosen so as to approximate the initial condition (1.2).

An approximation to a measure valued version of (1.1) is sought; our measure valued solution takes the form:

$$U^N(t, x) = p(x) = \frac{1}{N} \sum_{i=1}^{n} \delta(x - x_i), \qquad x_i > 0, \quad n, N = 1, 2, \ldots \tag{2.2}$$

Here, $U^N(t, x)$ is a sequence of jump processes, i.e. a series of random variables, whose subsequent state depends only on the current state, not on its history. It is a Markov process. $p(x)$ indicates a general representation of the state of the system at any point.

Using this representation, the concentration, $c(t,x)$ is approximated by:

$$c(t,x) \sim \frac{1}{N}\#\{i \;:\; x_i(t) = x\} \tag{2.3}$$

From this relation, it can be seen that the parameter N, known as the particle number, can be considered to be the equivalent of a normalisation parameter or sample size.

The subsequent coagulation of particles is a Markov Process, in which individual coagulation events are separated by an exponentially distributed waiting time [7]. The waiting time, τ, is generated according to:

$$\mathrm{Prob}\{\tau(p) \geq s\} = \exp(-\rho_K(p) \cdot s), \qquad s \geq 0, \tag{2.4}$$

where $\rho_K(p)$ is the waiting time parameter (for the coagulation process), given by

$$\rho_K(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} K(x_i, x_j). \tag{2.5}$$

For each event, particle indices i and j must be chosen according to the coagulation rate kernel, i.e. according to the probabilities:

$$\frac{K(x_i, x_j)}{\sum_{1 \leq i \neq j \leq n} K(x_i, x_j)}, \tag{2.6}$$

and then the system makes the jump:

$$p(x) \to p(x) - \frac{\delta(x - x_i)}{N} - \frac{\delta(x - x_j)}{N} + \frac{\delta(x - x_i - x_j)}{N}, \tag{2.7}$$

i.e. particles of sizes x_i and x_j are removed and a particle of size $x_i + x_j$ is added.

Generation of the joint probability distribution of i and j requires summing over a number of terms of the order of n^2, so a majorant kernel, $\hat{K}(x_i, x_j)$, is introduced, according to:

$$K(x_i, x_j) \leq \hat{K}(x_i, x_j), \qquad x_i, x_j > 0, \tag{2.8}$$

such that the waiting time, τ, is then efficiently generated with the parameter:

$$\hat{\rho}_K(p) = \frac{1}{2N} \sum_{1 \leq i \neq j \leq n} \hat{K}(x_i, x_j), \tag{2.9}$$

and the joint probability distribution:

$$\frac{\hat{K}(x_i, x_j)}{\sum_{1 \leq i \neq j \leq n} \hat{K}(x_i, x_j)} \tag{2.10}$$

enables independent generation of the indices i and j, reducing the computational time to the order of n.

In order that this method still gives convergence to the solution of a measure valued version of (1.1) in the limit $N \to \infty$, fictitious jumps are introduced. These are additional, null events that occur with probability

$$1 - \frac{K(x_i, x_j)}{\hat{K}(x_i, x_j)} \tag{2.11}$$

when the indices i and j have been chosen.

2.2 Majorant Kernel

A linear majorant kernel has been proposed in [2] for the kernel (1.3), in the case $D_F = 3$, namely:

$$\sqrt{\frac{1}{x} + \frac{1}{y}} \left(x^{1/3} + y^{1/3}\right)^2 \leq c(x + y), \qquad c > 0. \tag{2.12}$$

However, this majorant has several drawbacks, especially when we try to apply it to the case of $D_F < 3$, stemming from the fact that it does not have the same degree of homogeneity as the coagulation kernel itself.

Recall that a kernel is homogeneous with degree γ if:

$$K(\lambda x, \lambda y) = \lambda^{\gamma} K(x, y), \tag{2.13}$$

So in this case, the majorant is homogeneous with degree 1, and the kernel (1.3) itself is homogeneous with degree $\frac{2}{D_F} - \frac{1}{2}$.

It can easily be shown that the linear majorant (2.12) is only valid for $D_F \geq 2$ by considering the leading order terms of the kernel and its linear majorant. We have, for any x, y:

$$\sqrt{\frac{1}{x} + \frac{1}{y}} \left(x^{1/D_F} + y^{1/D_F}\right)^2 \geq \sqrt{\frac{1}{y}}\, x^{2/D_F} \tag{2.14}$$

Clearly for $D_F < 2$ this RHS increases faster than linearly with increasing x and there is not a value of c such that the required majorant inequality (2.8) holds for all x. Typical values for D_F for non-spherical particles are between 1.7 and 2.5 [12], so for the general case a new majorant kernel must be found.

Even for the case $D_F \geq 2$ we encounter problems. The value of the constant c depends on the minimum value, x_{min}, that the size can take:

$$c = 2\sqrt{2}\, x_{min}^{\frac{2}{D_F} - \frac{3}{2}}. \tag{2.15}$$

It is then clear that the acceptance efficiency of the majorant:

$$\frac{K(x, y)}{\hat{K}(x, y)} \tag{2.16}$$

is also dependent on the minimum size. For $x = y$ we find:

$$\frac{K(x,x)}{\hat{K}(x,x)} = \left(\frac{x}{x_{min}}\right)^{\frac{2}{D_F}-\frac{3}{2}}, \tag{2.17}$$

and for $y = x_{min}$, we find, in the limit $x \gg y$:

$$\lim_{x\to\infty} \frac{K(x,x_{min})}{\hat{K}(x,x_{min})} = \lim_{x\to\infty} \frac{1}{2\sqrt{2}} \left(\frac{x}{x_{min}}\right)^{\frac{2}{D_F}-1}. \tag{2.18}$$

These both have a lower limit of zero.

In our case, $x_{min} = 1$, so there is a well defined minimum size, which is equal to the size of the particles introduced in the source term in (1.1) and is therefore of the same order of magnitude as most of the particles in the system, at least for low simulation times. However, there are situations in which this is not the case; the ratio $\frac{x}{x_{min}}$ is then correspondingly larger, and much lower acceptance efficiency is encountered. For example, in the study of oxidation kinetics in flames, it may be desirable to seed a flame with particles much larger than the minimum size available.

In cases like this, the minimum particle size has a direct bearing on the efficiency of the majorant kernel, and therefore on the efficiency of simulation. We seek a majorant kernel that is homogeneous with the same degree as the actual coagulation kernel.

In the subsequent analysis, we use the dimensionless form of the coagulation kernel (1.3):

$$K(x,y) = \left(\frac{1}{x}+\frac{1}{y}\right)^{\frac{1}{2}} \left(x^{\frac{1}{D_F}} + y^{\frac{1}{D_F}}\right)^2 \tag{2.19}$$

Note that the function

$$\tilde{c}(t,x) = \alpha_2 c(\alpha_1\alpha_2 t, x), \qquad t \geq 0, \quad x > 0, \qquad \alpha_1, \alpha_2 \geq 0, \tag{2.20}$$

solves Equation (1.1) with the kernel $\tilde{K} = \alpha_1 K$ instead of K and initial condition $\tilde{c}_0 = \alpha_2 c_0$ instead of c_0. The scaling factors

$$\alpha_1 = \left(\frac{3}{4\pi}\right)^{\frac{2}{3}} \left(\frac{8\pi kT}{\rho}\right)^{\frac{1}{2}} \left(\frac{m_1}{\rho}\right)^{\frac{1}{6}} \tag{2.21}$$

and α_2, which will be specific to a given situation, can be calculated after a general simulation takes place, and the dimensionless time and concentration can be scaled by these factors to give a dimensional answer.

A majorant kernel is constructed by using the inequalities:

$$(a+b)^\varepsilon \leq 2^{\varepsilon-1}(a^\varepsilon + b^\varepsilon) \qquad \varepsilon \geq 1, \quad a,b > 0 \tag{2.22}$$

and

$$(a+b)^\varepsilon \leq (a^\varepsilon + b^\varepsilon) \qquad 0 < \varepsilon < 1, \quad a,b > 0. \tag{2.23}$$

For a proof of these, see **Appendix A**.

We now have:

$$\left(\frac{1}{x}+\frac{1}{y}\right)^{\frac{1}{2}}\left(x^{\frac{1}{D_F}}+y^{\frac{1}{D_F}}\right)^2 \leq 2\left(x^{-\frac{1}{2}}+y^{-\frac{1}{2}}\right)\left(x^{\frac{2}{D_F}}+y^{\frac{2}{D_F}}\right) \tag{2.24}$$

In fact, the maximum value of the ratio $\frac{K}{\hat{K}}$ is less than one, so this majorant is not as efficient as it could be. By examining the maximum value of $\frac{K}{\hat{K}}$, we find that it is more useful to use (see **Appendix B** for justification):

$$\left(\frac{1}{x}+\frac{1}{y}\right)^{\frac{1}{2}}\left(x^{\frac{1}{D_F}}+y^{\frac{1}{D_F}}\right)^2 \leq \sqrt{2}\left(x^{-\frac{1}{2}}+y^{-\frac{1}{2}}\right)\left(x^{\frac{2}{D_F}}+y^{\frac{2}{D_F}}\right) \tag{2.25}$$

in the case $1.7 < D_F < 2.5$, and

$$\left(\frac{1}{x}+\frac{1}{y}\right)^{\frac{1}{2}}\left(x^{\frac{1}{3}}+y^{\frac{1}{3}}\right)^2 \leq 1.4178\left(x^{-\frac{1}{2}}+y^{-\frac{1}{2}}\right)\left(x^{\frac{2}{3}}+y^{\frac{2}{3}}\right) \tag{2.26}$$

for $D_F = 3$. Both of these are clearly more efficient majorant kernels.

In this paper we take a value for D_F from the middle of the typical range for non-spherical particles: $D_F = 2.1$.

Now when we consider the acceptance efficiency of the majorant, we find, for $x = y$:

$$\frac{K(x,x)}{\hat{K}(x,x)} = 1 \tag{2.27}$$

And for $y = x_{min}, x \gg y$ we find:

$$\lim_{x\to\infty}\frac{K(x,x_{min})}{\hat{K}(x,x_{min})} = \frac{1}{\sqrt{2}}. \tag{2.28}$$

Because the kernel (1.3) and its majorant (2.25) are homogeneous with the same degree, the dependence of the acceptance efficiency on the minimum size is removed. Also, we find that we can use this new majorant for the case $D_F < 2$.

The majorant kernel can be multiplied out to give

$$\hat{K}(x_i,x_j) = \sqrt{2}\left(x_i^{\frac{2}{D_F}-\frac{1}{2}}+x_j^{\frac{2}{D_F}-\frac{1}{2}}+x_i^{\frac{2}{D_F}}x_j^{-\frac{1}{2}}+x_i^{-\frac{1}{2}}x_j^{\frac{2}{D_F}}\right) \tag{2.29}$$

which can be considered as being composed of

$$\hat{K} = \hat{K}_1+\hat{K}_2+\hat{K}_3+\hat{K}_4. \tag{2.30}$$

The choice of which of these majorant components to use to generate indices i and j is then made probabilistically, as described in section 2.2 of [1].

It should be noted that

$$\hat{K}_1(x_i, x_j) = \sqrt{2}(x_i^{\frac{2}{D_F}-\frac{1}{2}}) \qquad \text{and} \qquad \hat{K}_2(x_i, x_j) = \sqrt{2}(x_j^{\frac{2}{D_F}-\frac{1}{2}})$$

give identical distributions for i and j, except with the indices swapped. The same is true for $\hat{K}_3$ and $\hat{K}_4$. Because the coagulation step of the stochastic algorithm is symmetrical with respect to the indices i and j, the order in which they are generated does not matter. Therefore, for the purposes or generating the particle indices, we can consider just two different functions, $\hat{K}_1$ and $\hat{K}_3$. Thus we calculate a value of $\hat{\rho}(p)$ associated with each kernel component according to:

$$\hat{\rho}_1(p) = \frac{1}{2N} \sum_{1 \le i \ne j \le n} \sqrt{2}(x_i^{\frac{2}{D_F}-\frac{1}{2}}) \quad \text{and} \quad \hat{\rho}_3(p) = \frac{1}{2N} \sum_{1 \le i \ne j \le n} \sqrt{2}(x_i^{\frac{2}{D_F}} x_j^{-\frac{1}{2}}), \tag{2.31}$$

and choose $\hat{K}_1$ (or otherwise $\hat{K}_3$) to generate indices i and j with probability:

$$\frac{\hat{\rho}_1(p)}{\hat{\rho}_1(p) + \hat{\rho}_3(p)} \tag{2.32}$$

Note that we could have simply multiplied out the second bracket of (2.19) rather than using the inequality (2.22) and obtained:

$$\hat{K} = x^{\frac{2}{D_F}-\frac{1}{2}} + y^{\frac{2}{D_F}-\frac{1}{2}} + x^{\frac{2}{D_F}} y^{-\frac{1}{2}} + x^{-\frac{1}{2}} y^{\frac{2}{D_F}} + 2x^{\frac{1}{D_F}-\frac{1}{2}} y^{\frac{1}{D_F}} + 2x^{\frac{1}{D_F}} y^{\frac{1}{D_F}-\frac{1}{2}} \tag{2.33}$$

However, using this kernel would require at each coagulation step the storage and update of two more quantities:

$$\sum_{i=1}^{n} x_i^{\frac{1}{D_F}-\frac{1}{2}} \quad \text{and} \quad \sum_{i=1}^{n} x_i^{\frac{1}{D_F}} \tag{2.34}$$

which would reduce the efficiency of the simulation.

2.3 Organisation Of Particle System

For efficient simulation and generation of size dependent distributions, the stochastic particle sizes $(x_1, x_2 \ldots)$ are stored in a series of arrays (or bins) as described in [1]. The particles are organised into γ groups, i.e. their sizes are denoted by

$$y_{z,k}, \qquad z = 1, \ldots, \gamma, \quad k = 1, \ldots, \alpha_z, \tag{2.35}$$

so that

$$b_{z-1} < y_{z,k} \le b_z, \qquad \forall z = 1, \ldots, \gamma \quad k = 1, \ldots, \alpha_z, \tag{2.36}$$

where

$$0 \equiv b_0 < b_1 < \ldots < b_\gamma \tag{2.37}$$

and

$$x_{max} \le b_\gamma. \tag{2.38}$$

x_{max} is the upper bound for the particle size, which will depend on the initial condition and the inflow rate.

The group size bounds, b_z, are chosen to be related by

$$b_z = \beta^{z-1}, \quad z = 1, 2, \dots, \tag{2.39}$$

where $\beta > 1$.

Then, in order to generate the distribution

$$\frac{x_j^\varepsilon}{\sum_{l=1}^n x_l^\varepsilon}, \qquad j = 1, \dots, n, \tag{2.40}$$

the choice of group can be made by the discrete inversion (linear search) method, while the choice of particle index within the group can be done by acceptance-rejection, with a minimum efficiency of $1/\beta^\varepsilon$

2.4 Source Term

To model the source term in Equation (1.1), we introduce a stochastic particle system for the source particle size distribution, $x_i^{in}, \quad i = 1, 2, \dots, n^{in}$. In this case, the source particle size distribution can be modelled as

$$U_{in}^N(x) = \frac{1}{N} \sum_{i=1}^{n^{in}} \delta(x - x_i^{in}) \tag{2.41}$$

The relationship between the source particle number, n^{in}, and the initial particle number, $n(0)$ is given by

$$\lim_{N\to\infty} \int \phi(x) U_{in}^N(x) dx = \int \phi(x) c^{in}(x) dx \tag{2.42}$$

In the case $\phi(x) = 1$:

$$\int c^{in}(x) dx = \lim_{N\to\infty} \int U_{in}^N(x) dx = \frac{n^{in}}{N}. \tag{2.43}$$

Similarly:

$$\int c^0(x) dx = \lim_{N\to\infty} \int U^N(0, x) dx = \frac{n(0)}{N} \tag{2.44}$$

Thus the relationship between n^{in} and $n(0)$ is given by:

$$\frac{n^{in}}{n(0)} = \frac{\int c^{in}(x) dx}{\int c^0(x) dx} \tag{2.45}$$

and in our case of $c^{in}(x) = \delta_1$ we have $n^{in} = N$.

Stochastic simulation of a source term proceeds similarly to the case of coagulation. Single events are separated by an exponentially distributed waiting time (2.4), with the parameter ρ_{in} given by:

$$\rho_{in}(p) = In^{in}. \tag{2.46}$$

Each event is described by the jump:

$$p(x) \rightarrow p(x) + \frac{\delta(x-1)}{N}, \tag{2.47}$$

i.e. a particle of size 1 is added to the system.

2.5 Algorithm

Extension of stochastic simulation to include more than one process proceeds as follows. For each process (labelled α) occurring, an associated waiting time parameter, $\rho_\alpha(p)$ is derived. Then, the waiting time between events is exponentially distributed (2.4) with parameter $\rho(p) = \sum_\alpha \rho_\alpha(p)$. The event, α occurring after this time is chosen according to the probabilities $\frac{\rho_\alpha(p)}{\rho(p)}$, and the stochastic jump associated with this event is performed.

Combining the simulation methods for the processes of coagulation and particle inception results in the **improved stochastic algorithm** we use to simulate solutions to Equation (1.1).

1. Generate the initial state $U^N(0) = p \in \mathcal{S}^N$
2. Wait an exponentially distributed time step τ with parameter (c.f. (2.4))

$$\begin{aligned} \hat{\rho}(p) &= \hat{\rho}_K(p) + \rho_{in}(p) \\ &= \frac{\sqrt{2}}{N}\left((n-2)\sum_{i=1}^{n} x_i^{\frac{2}{D_F}-\frac{1}{2}} + \sum_{i=1}^{n} x_i^{\frac{2}{D_F}} \sum_{i=1}^{n} x_i^{-\frac{1}{2}}\right) + In^{in} \end{aligned}$$

3. With probability

$$\frac{\rho_{in}(p)}{\hat{\rho}(p)}$$

   go to step 4 Otherwise go to step 5.
4. Perform a source step, i.e.
   (a) Add a cluster of size 1 to the system.
   (b) Go to step 2.
5. Perform a coagulation step, i.e.

- (a) With probability (c.f. (2.31))

  $$\frac{\hat{\rho}_1(p)}{\hat{\rho}_1(p)+\hat{\rho}_3(p)}$$

  go to step 5b. Otherwise go to step 5c.
- (b) Use $\hat{K}_1$ to generate indices i and j, i.e.
  - i. Generate i according to
    - A. Choose the group index, z, according to the probabilities

      $$P_z = \frac{1}{c}\sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{D_F}-\frac{1}{2}}, \qquad z = 1,\ldots,\gamma.$$

      where

      $$c = \sum_{l=1}^{n} x_l^{\frac{2}{D_F}-\frac{1}{2}} = \sum_{z=1}^{\gamma}\sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{D_F}-\frac{1}{2}}$$
    - B. Choose the particle index $k = 1,\ldots,\alpha_z$ uniformly within the group z.
    - C. The particle index is accepted with probability

      $$\frac{y_{z,k}^{\frac{2}{D_F}-\frac{1}{2}}}{b_z^{\frac{2}{D_F}-\frac{1}{2}}}$$

      Otherwise, go to step 5(b)iB.
  - ii. Generate j uniformly on the set $i = 1,\ldots,n$.
  - iii. If $i = j$ return to step 5b. Otherwise go to step 5d
- (c) Use $\hat{K}_3$ to generate indices i and j, i.e.
  - i. Generate i according to
    - A. Choose the group index, z, according to the probabilities

      $$P_z = \frac{1}{c}\sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{D_F}}, \qquad z = 1,\ldots,\gamma.$$

      where

      $$c = \sum_{l=1}^{n} x_l^{\frac{2}{D_F}} = \sum_{z=1}^{\gamma}\sum_{k=1}^{\alpha_z} y_{z,k}^{\frac{2}{D_F}}$$
    - B. Choose the particle index $k = 1,\ldots,\alpha_z$ uniformly within the group z.
    - C. The particle index is accepted with probability

      $$\frac{y_{z,k}^{\frac{2}{D_F}}}{b_z^{\frac{2}{D_F}}}$$

      Otherwise, go to step 5(c)iB.

ii. Generate j according to

A. Choose the group index, z, according to the probabilities

$$P_z = \frac{1}{c} \sum_{k=1}^{\alpha_z} y_{z,k}^{-\frac{1}{2}}, \qquad z = 1, \ldots, \gamma.$$

where

$$c = \sum_{l=1}^{n} x_l^{-\frac{1}{2}} = \sum_{z=1}^{\gamma} \sum_{k=1}^{\alpha_z} y_{z,k}^{-\frac{1}{2}}$$

B. Choose the particle index $k = 1, \ldots, \alpha_z$ uniformly within the group z.

C. The particle index is accepted with probability

$$\frac{y_{z,k}^{-\frac{1}{2}}}{b_{z-1}^{-\frac{1}{2}}}$$

Otherwise, go to step 5(c)iiB.

iii. If $i = j$ return to step 5c. Otherwise go to step 5d

(d) With probability

$$\frac{K(x_i, x_j)}{\hat{K}(x_i, x_j)},$$

perform a coagulation jump, i.e remove the particles x_i and x_j and add a particle of size $x_i + x_j$. Otherwise, the interaction is fictitious, i.e. nothing changes.

(e) Go to step 2

3 Numerical Results

3.1 Sample Results

To study the efficiency of our improved stochastic algorithm we perform repeated simulations with varying values of N and calculate confidence intervals for the results we obtain. In all our simulations we take $I = 0.5$ and $D_F = 2.1$ and we set the initial condition to be

$$c(0, x) = c_0(x) = \begin{cases} 1 & \text{for } x = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Figures 1 and **2** show typical results that can be obtained from the stochastic simulation. **Figure 1** shows a histogram of the particle size distribution, with the lower confidence bound given by the solid line and the upper confidence bound given by the dotted line. The particle sizes are grouped in bins as described in **Section 2.3**

with a value of $\beta = 2$, in order to show the increasing concentration of the larger particles as time increases. It can also be seen that the concentration of smaller particles remains approximately constant as time increases, due to the formation of new monomer particles, as modelled by the source term.

Figure 2 shows the time evolution of three moments of the PSD. As expected, the higher the moment, the more sensitive it is to the larger particle sizes, and therefore the wider the confidence bounds.

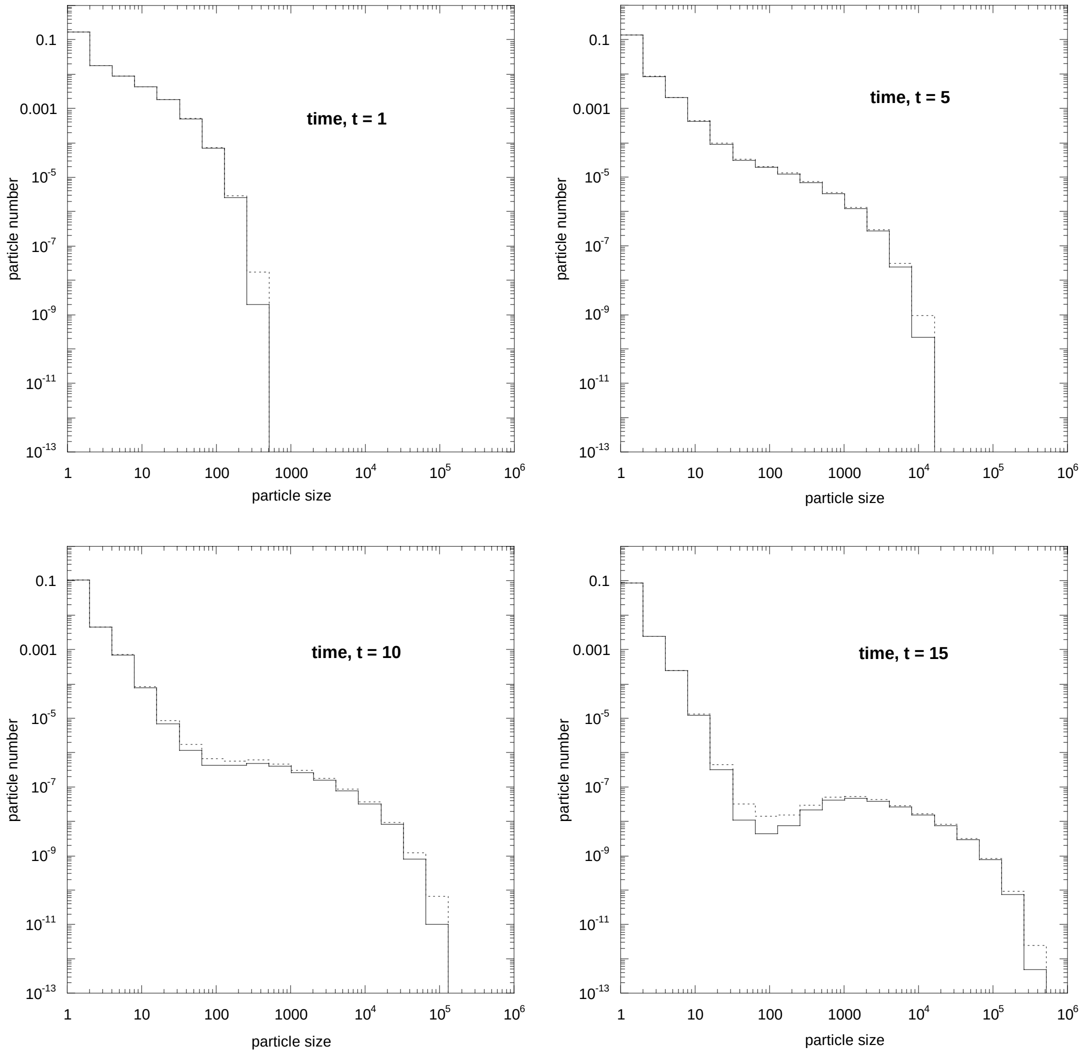

Figure 1: Particle size distributions at various times.

3.2 Convergence

To study the validity of our improved stochastic algorithm, it is useful to consider the convergence properties, i.e. as we increase N, and therefore increase the required CPU times, how quickly does the simulated solution converge to the exact solution?

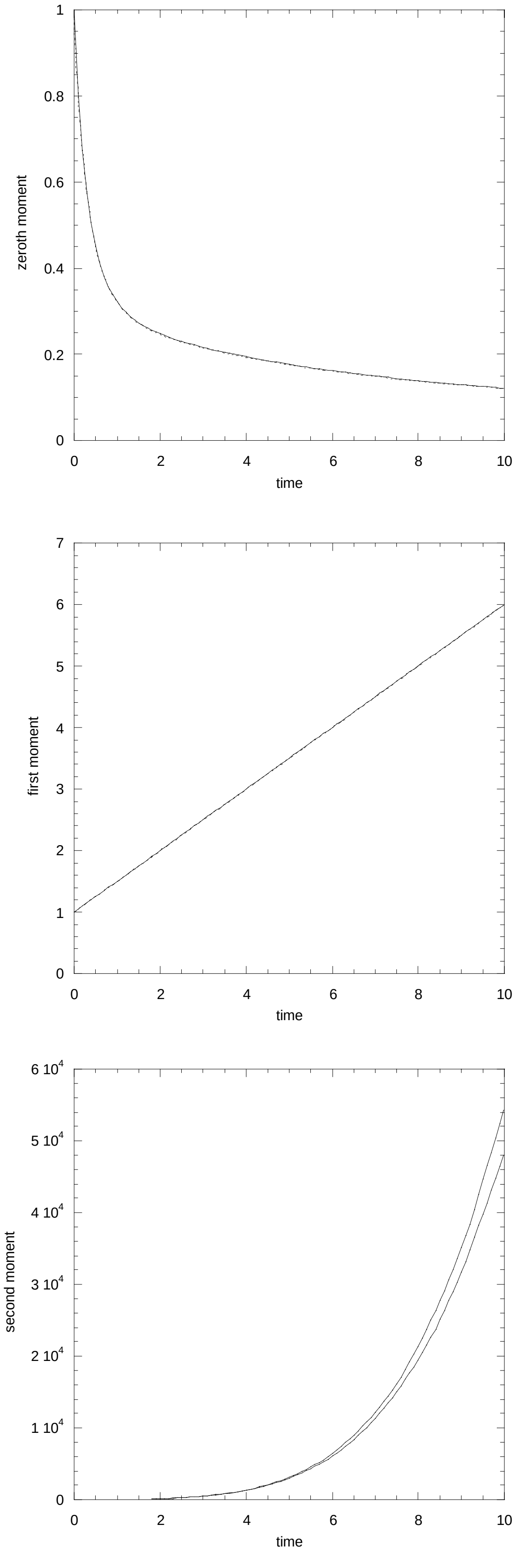

Figure 2: Moments of the particle size distribution.

Following [11], confidence intervals are calculated as follows:

Typical macroscopic properties, such as moments of the particle size distribution, are functionals of the form

$$F(t) = \int_0^\infty \phi(x)c(t,x)dx. \tag{3.2}$$

These functionals are approximated (as $N \to \infty$) by the random variable

$$\xi^{(N)}(t) = \frac{1}{N}\sum_{i=1}^{n(t)} \phi(x_i(t)) \tag{3.3}$$

In order to estimate the expectation and the random fluctuations of the estimator $\xi^{(N)}(t)$, a number L of independent runs are performed. The corresponding values of the random variable are denoted by $\xi^{(N,1)}(t), \ldots, \xi^{(N,L)}(t)$. The **empirical mean value** of $\xi^{(N)}(t)$ is defined as

$$\eta_1^{(N,L)}(t) = \frac{1}{L}\sum_{l=1}^{L} \xi^{(N,l)}(t). \tag{3.4}$$

The variance of $\xi^{(N)}(t)$ satisfies

$$\text{Var } \xi^{(N)}(t) \equiv E\left[\xi^{(N)}(t) - E\xi^{(N)}(t)\right]^2 = E\left[\xi^{(N)}(t)\right]^2 - \left[E\xi^{(N)}(t)\right]^2 \tag{3.5}$$

and is estimated by the **empirical variance** defined as

$$\eta_2^{(N,L)}(t) = \frac{1}{L}\sum_{l=1}^{L} \left[\xi^{(N,l)}(t)\right]^2 - \left[\eta_1^{(N,L)}(t)\right]^2. \tag{3.6}$$

The empirical mean (3.4) is used to approximate the macroscopic quantity (3.2). The error of this approximation is denoted as

$$e^{(N,L)} = |\eta_1^{(N,L)}(t) - F(t)| \tag{3.7}$$

and consists of the following two components. The **systematic error** is the difference between the mathematical expectation of the random variable (3.3) and the exact value of the functional, i.e.

$$e_{sys}^{(N,L)} = E\xi^{(N)}(t) - F(t). \tag{3.8}$$

The **statistical error** is the difference between the empirical mean value and the expected value of the random variable, i.e.

$$e_{stat}^{(N,L)}(t) = \eta_1^{(N,L)}(t) - E\xi^{(N)}(t). \tag{3.9}$$

The random variable

$$\frac{\eta_1^{(N,L)}(t) - E\xi^{(N)}(t)}{\sqrt{\text{Var } \eta_1^{(N,L)}(t)}} \tag{3.10}$$

has asymptotically (for $L \geq 50$) a standard normal distribution, as a consequence of the central limit theorem. Thus

$$\text{Prob}\left\{ \frac{\left|\eta_1^{(N,L)}(t) - E\xi^{(N)}(t)\right|}{\sqrt{\text{Var }\eta_1^{(N,L)}(t)}} \leq a_p \right\} \sim p, \qquad p \in (0,1), \tag{3.11}$$

where the value of a_p is determined from statistical tables.

Note that

$$\text{Var }\eta_1^{(N,L)}(t) = \frac{1}{L}\text{Var }\xi^{(N)}(t) \sim \frac{1}{L}\eta_2^{(N,L)}(t). \tag{3.12}$$

A **confidence interval** can therefore be constructed as

$$I_p = \left[\eta_1^{(N,L)}(t) - a_p\sqrt{\frac{\eta_2^{(N,L)}(t)}{L}}, \eta_1^{(N,L)}(t) + a_p\sqrt{\frac{\eta_2^{(N,L)}(t)}{L}}\right], \tag{3.13}$$

where p is the **confidence level**. This means that

$$\text{Prob}\left\{E\xi^{(N)}(t) \in I_p\right\} = \text{Prob}\left\{ |e_{stat}^{(N,L)}(t)| \leq a_p\sqrt{\frac{\eta_2^{(N,L)}(t)}{L}} \right\} \sim p. \tag{3.14}$$

Thus, the value

$$c_p^{(N,L)}(t) = a_p\sqrt{\frac{\eta_2^{(N,L)}(t)}{L}} \tag{3.15}$$

is a probabilistic upper bound for the statistical error.

In this study, a confidence level of 99.9% or $p = 0.999$ with $a_p = 3.29$ has been used.

In order to describe the statistical error in $[0, T]$ we split this time interval into M equidistant subintervals of length Δt according to the discretisation

$$t_i = i\Delta t, \qquad i = 0, 1, \ldots, M, \tag{3.16}$$

with $t_M = T$ and use the quantity

$$c_{stat} = \max_i \left\{c_p^{(N,L)}(t)\right\} \tag{3.17}$$

as a measure for the statistical error.

To study the systematic error of the solution algorithm we use an approximation $\zeta(t)$ of the corresponding macroscopic quantity $F(t)$ obtained using a single run of the algorithm with as high a value of N as is feasible. Here we use $N = 10^7$. Then the error

$$\tilde{e}^{(N,L)}(t) = |\eta_1^{(N,L)}(t) - \zeta(t)| \tag{3.18}$$

is a good approximation of the true error $e^{(N,L)}(t)$. In order to get an expression for (3.18) on $[0, T]$ we calculate the quantity

$$c_{tot} = \frac{1}{M+1}\sum_{i=0}^{M} \tilde{e}^{(N,L)}(t_i) \tag{3.19}$$

as an estimate for the average error in the time interval $[0, T]$.

The errors c_{tot} and c_{stat} are calculated for the average particle size, i.e. the ratio of the first moment of the PSD to the zeroth. By using this term, we study a property both of the total number of particles, and of the size of the particles in the population. **Table 1** contains the results of our numerical study. In the table, t_{sr} is the CPU time (in seconds) needed for a single run. The simulations were all performed on a 866MHz Pentium PC.

Table 1: Computational study for improved stochastic algorithm ($N \times L = 6.4 \times 10^7$).

N	c_{stat}	c_{tot}	t_{sr}	$t_{sr}/N \times 10^4$
125	0.0773	2.13	0.012	0.94
250	0.0681	1.26	0.024	0.97
500	0.0633	0.697	0.048	0.96
1000	0.0607	0.368	0.097	0.97
2000	0.0601	0.180	0.22	1.1
4000	0.0601	0.0896	0.42	1.0
8000	0.0609	0.0423	0.92	1.2
16000	0.0616	0.0156	2.1	1.3

For a large enough number of runs, so that the systematic error is larger than the statistical error, we can estimate the **order of convergence**. **Figure 3** shows $c_{tot} \pm c_{stat}$ plotted against N. The solid line shows the slope $\frac{1}{N}$.

3.3 Efficiency

To give an idea of the efficiency of the improved stochastic algorithm, we compare it with a standard **direct solution Monte Carlo algorithm**, (**DSMC**); we use the algorithm proposed by Gillespie [7]. We also examine the efficiency of our new majorant kernel by comparison with the linear majorant (2.12). **Table 2** and **Figure 4** show greatly improved efficiency of the new algorithm. It can be seen that for the DSMC algorithm, the single-run CPU time, t_{sr}, increases as N^2. According to (2.5), the average number of time steps in a given time interval will be proportional to N, and due to the method of selecting i and j, the CPU time to perform each coagulation step is proportional to N.

In contrast, we have a number of different cases, depending on the value of N, that our improved stochastic algorithm gives for the relationship between t_{sr} and N. For low N, our algorithm gives t_{sr} proportional to N. The number of time steps in a given time interval is still proportional to N, but the slowest step in the process is the calculation of the confidence intervals as described in **Section 3.2**. This process is not dependent on N, so the overall CPU time for a single run is proportional to

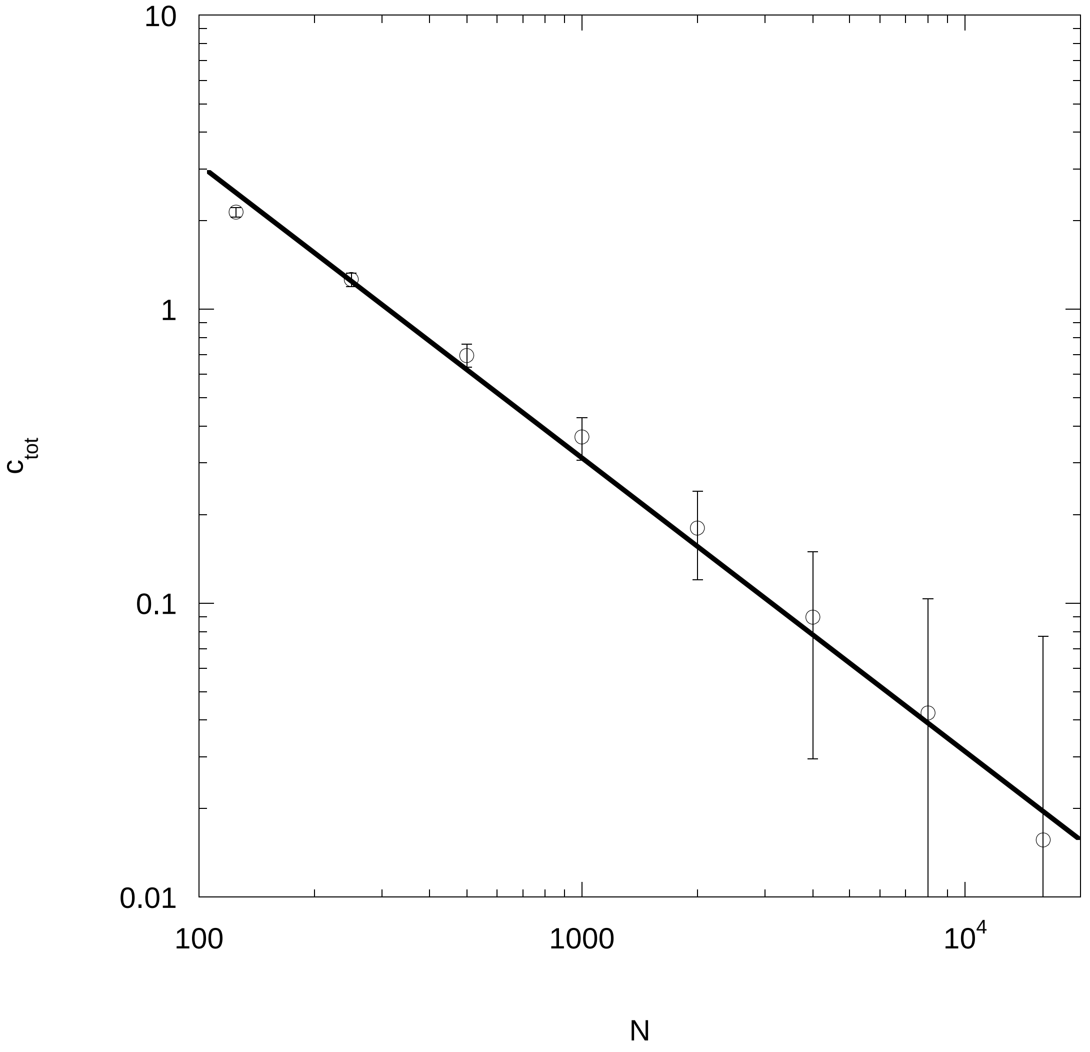

Figure 3: Order of convergence for improved stochastic algorithm. The solid line shows the slope $\frac{1}{N}$

N. It is only as N increases that the rate of increase of t_{sr} goes beyond linear. As N increases, the next step of the algorithm to dominate is the bin selection step (step 5(b)iB, 5(c)iB or 5(c)iiB in the simulation algorithm). As the number of bins to choose from increases as the logarithm of the particle number (cf. (2.38) and (2.39)), the CPU time of each coagulation step will increase as $(\ln N)$. Finally, due to the necessity of reorganisation of the particle size array at each coagulation step, we reach the case where the CPU time for each coagulation step increases as N and therefore t_{sr} is proportional to N^2 (as in the case of the DSMC algorithm). But this does not happen until the particle number is much greater than is necessary for a reasonably accurate simulation.

Table 2: Computational study for DSMC algorithm ($N \times L = 8 \times 10^5$).

N	c_{stat}	c_{tot}	t_{sr}	$t_{sr}/N \times 10^2$
125	0.684	2.16	0.14	0.11
250	0.615	1.27	0.55	0.21
500	0.568	0.704	2.1	0.42
1000	0.525	0.399	8.3	0.83
2000	0.555	0.162	32	1.6
4000	0.533	0.118	130	3.2
8000	0.611	0.0566	497	6.2
16000	0.612	0.0459	1980	12

Figure 5 compares the single run CPU time (for constant N) as a function of t_{sim}, the time variable in the simulation, using our new majorant kernel and the linear majorant. Without the source term, the CPU time for both kernels levels off fairly soon, as by this point there are very few particles left in the system (with coagulation as the only process, the number of particles is strictly decreasing). When the source term is included, the CPU time is significantly increased. When our new majorant is used, CPU time increases linearly with t_{sim}. For the linear majorant, CPU time increases slightly faster than linearly. In both cases, our new majorant gives a reasonable (10 – 50%) cut in CPU time as compared with the linear kernel, but the ratio of CPU times (linear to new) increases with increasing t_{sim}. This can be explained by considering the acceptance efficiency (2.16) of the two kernels. At later simulation times, the particle system contains larger particles. The ratio $\frac{x}{x_{min}}$ increases, and the efficiency of the simulation decreases. **Figure 6** shows the relative number of fictitious jumps for both majorants, with and without a source term.

The new majorant kernel gives an increase in efficiency of simulation that is more significant for longer simulation times. These results, for a constant particle number, N, can be considered to hold for other particle numbers by examining **Figure 7**, which shows CPU time increasingly linearly with particle number. Thus, we can expect a similar gain in efficiency regardless of particle number.

The order of convergence for the DSMC algorithm is compared with $\frac{1}{N}$ in **Figure**

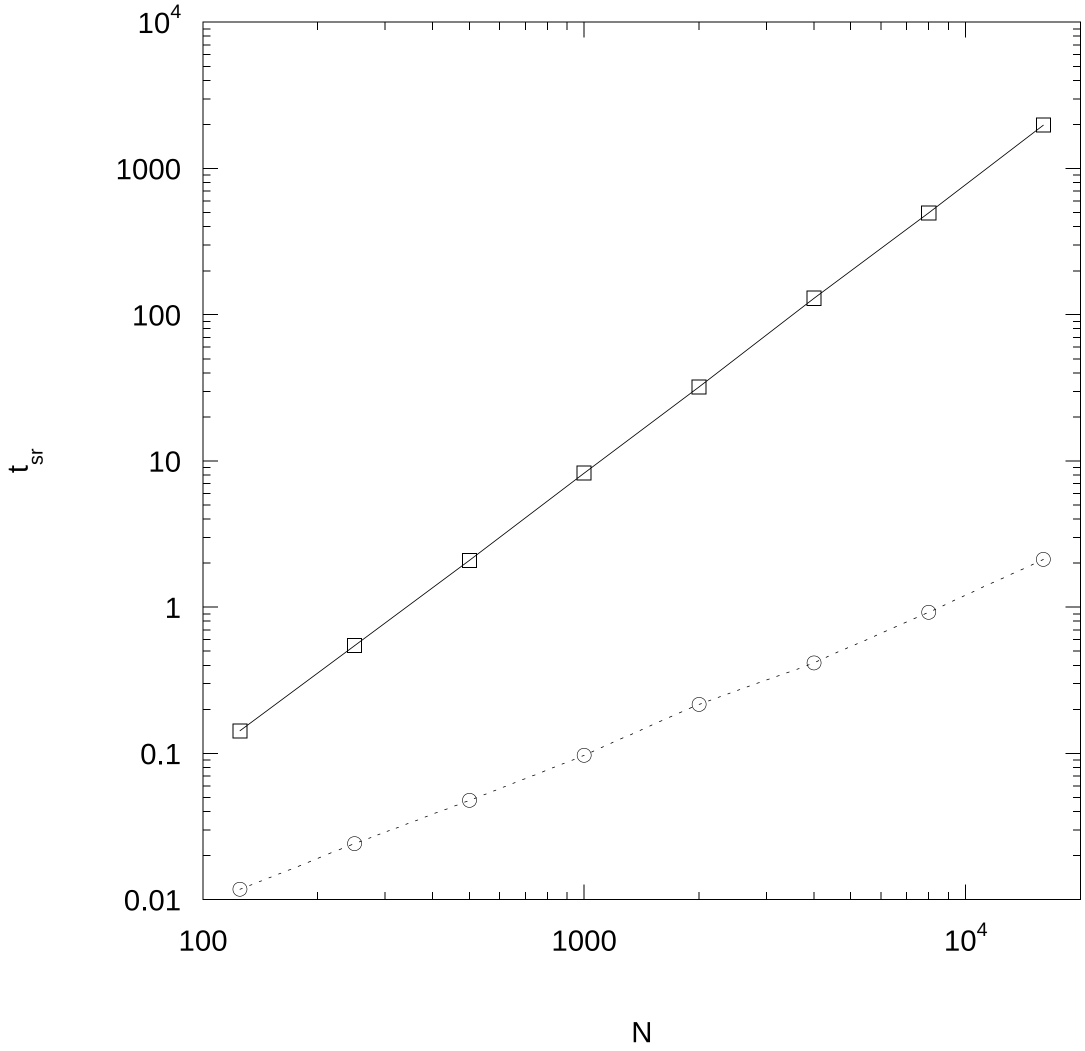

Figure 4: CPU time for a single run for the DSMC algorithm (solid) and our improved algorithm (dotted).

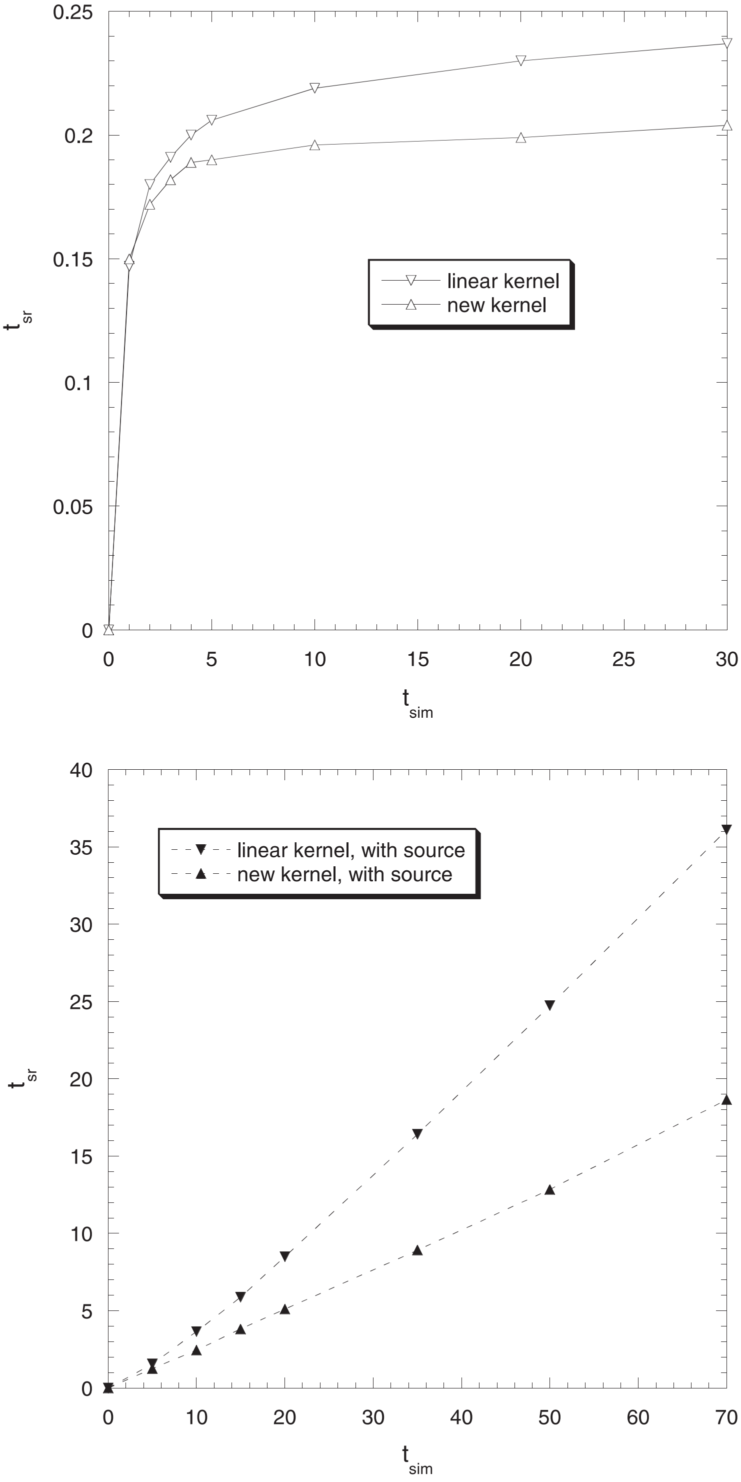

Figure 5: CPU time for a single run for the linear majorant and for our new majorant. t_{sim} is the simulation time.

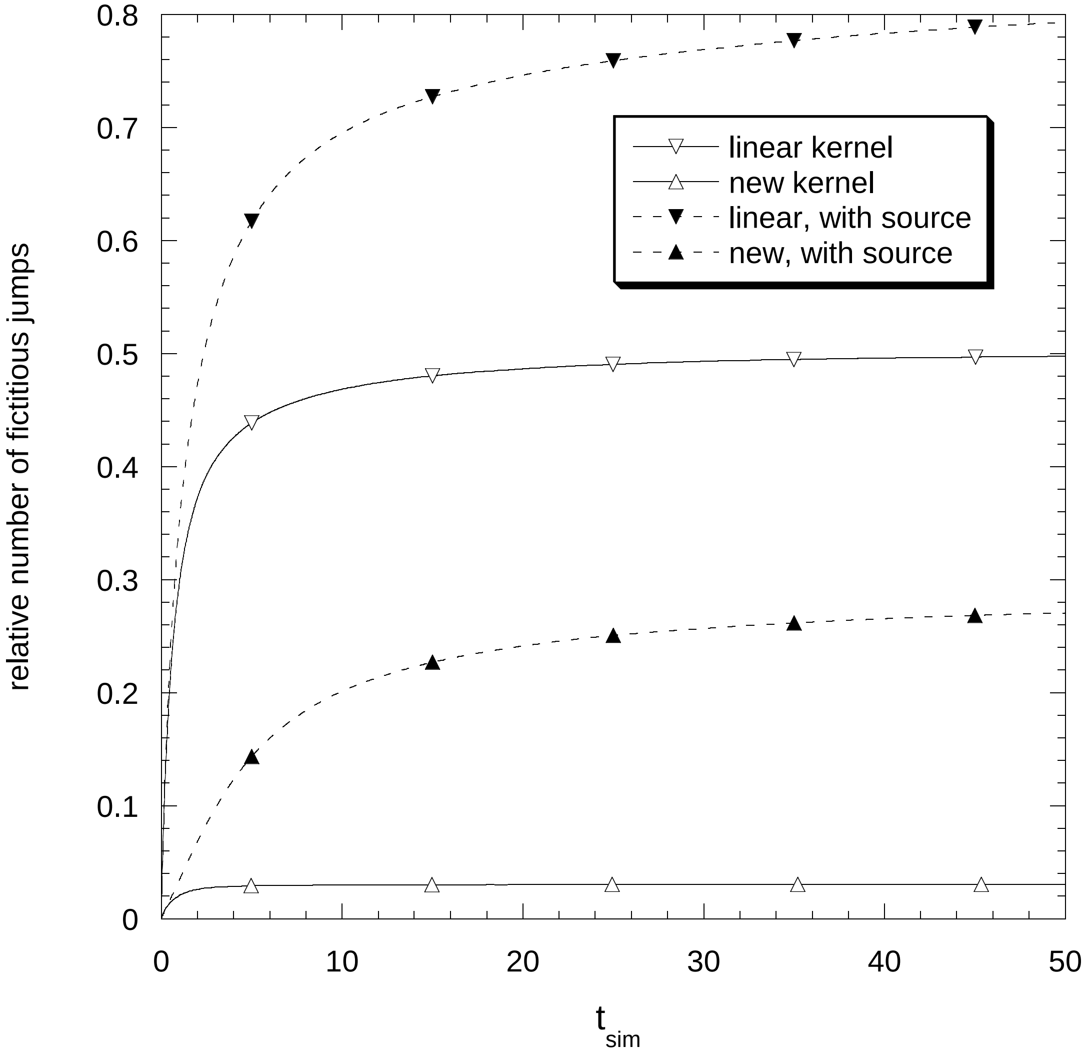

Figure 6: Comparison of the efficiency of the two majorants.

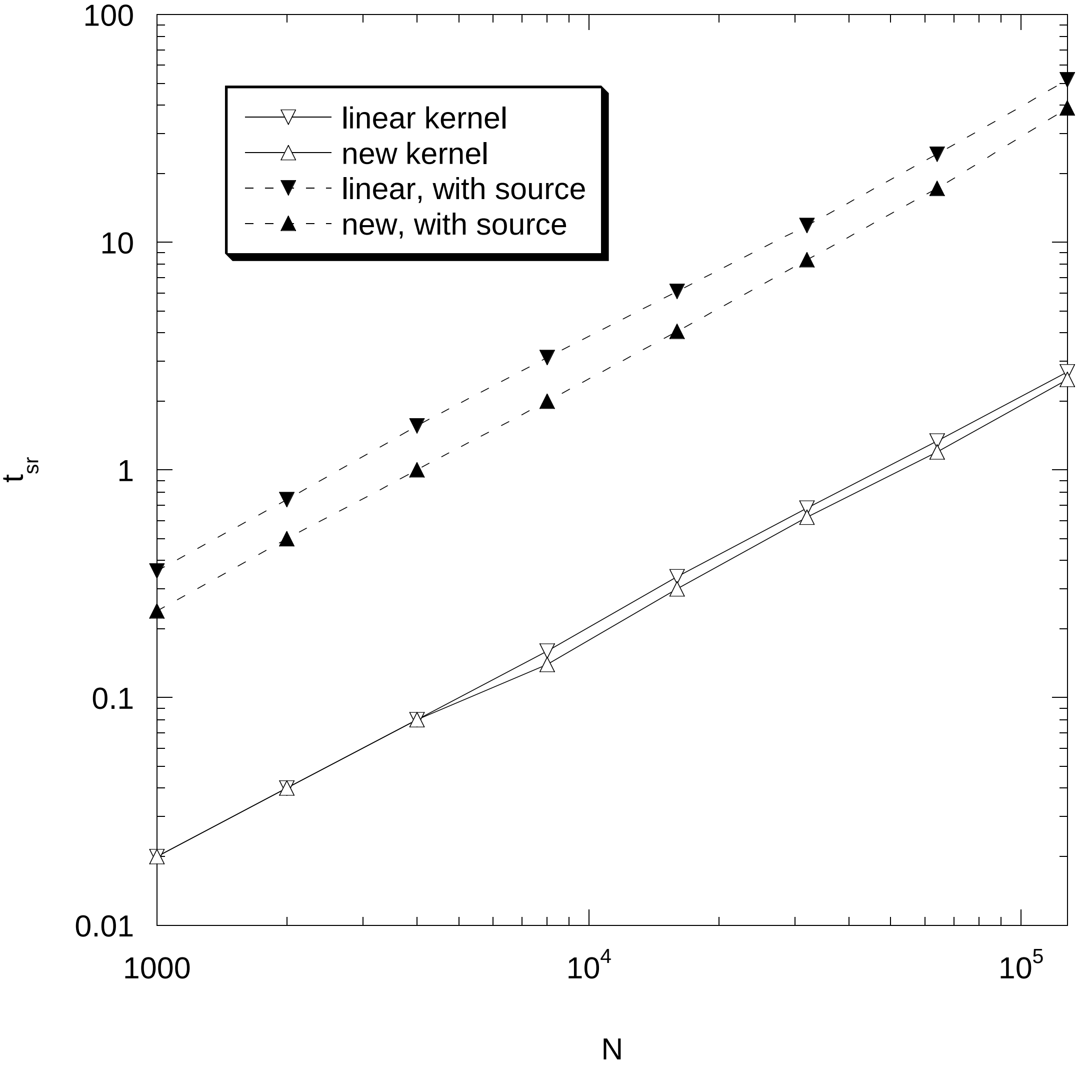

Figure 7: CPU time for a single run for the linear majorant and our new majorant. N is the particle number.

8. Due to the poor efficiency of this method it is not feasible to perform sufficient runs such that the systematic error is significantly larger than the statistical error. However, the data do seem to fit the $\frac{1}{N}$ line plotted on the figure. This is to be expected, as the convergence of $U^N(t,x)$ does not depend on any majorant that is used (or not used).

4 Conclusions

We have developed and studied an improved stochastic simulation algorithm to model the nano-particle dynamics applicable to, for example, the production of silica by flame aerosol synthesis (cf. Equations (1.1)–(1.4)). The coagulation was simulated according to [1], with the introduction of a new majorant kernel. The new majorant kernel allows us to study a wider range of problems than the linear majorant, previously suggested in [2]. A source term was included in the population balance equation and was also simulated.

The results can easily be presented in the form of particle size distributions (PSDs) or any function of the PSDs, such as moments.

The convergence properties of the algorithm were studied, to give an indication of the decrease of the systematic error when the particle number, N, is increased. For a very large number of runs, the statistical error is smaller than the systematic error, and the systematic error decreases as $\frac{1}{N}$. It is likely however, that for normal simulations, the number of runs will be sufficiently small that the systematic error is smaller than the statistical error and hence the exact solution will lie within the calculated confidence bounds.

The simulation algorithm was compared to a standard direct simulation Monte Carlo (DSMC) algorithm (i.e. one not using fictitious jumps) to determine the improvement in efficiency. We also compared the efficiency of our new majorant kernel with that of the linear kernel.

By comparing it to the DSMC algorithm proposed by Gillespie [7] we see that we have reduced CPU times by orders of magnitude, and made it possible to achieve accuracy that the inefficiency of the DSMC algorithm prohibited. We also achieve good gains in efficiency (up to 50% cut in CPU time) by using our new majorant kernel in place of the linear majorant, while also being able to study a wider range of problems. This is very promising for future work, both in the study of fumed silica and in other fields, such as soot formation, where the population balance equation is used.

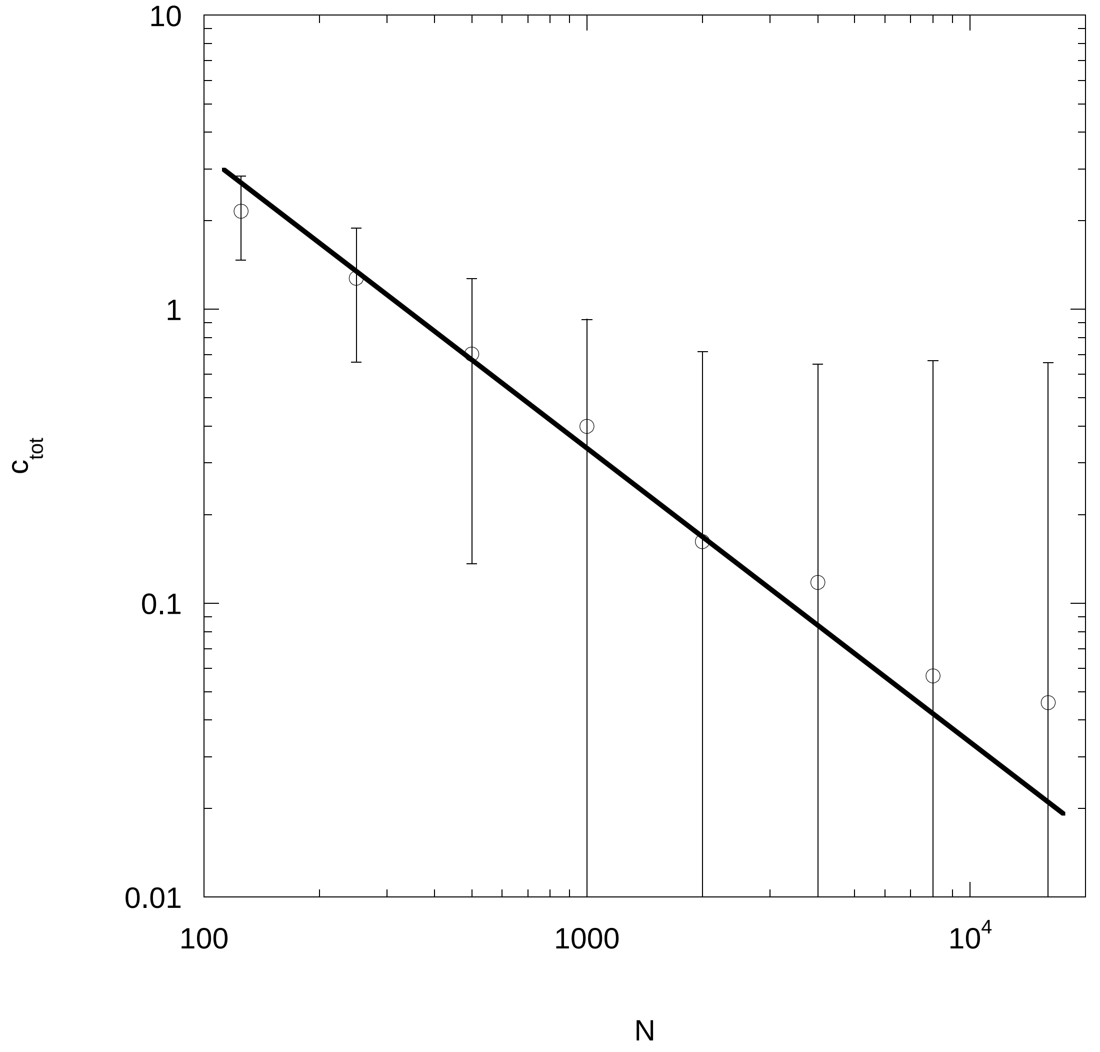

Figure 8: Order of convergence for the DSMC algorithm. The solid line shows the slope $\frac{1}{N}$

A Justification Of Choice Of Majorant Kernel

To prove that:

$$(a+b)^n \leq 2^{n-1}\left(a^n+b^n\right) \qquad n>1, \quad a,b>0 \tag{A.1}$$

Consider the function:

$$\begin{aligned} f(a,b) &= 2^{n-1}\left(a^n+b^n\right)-(a+b)^n \\ &= a^n\left[2^{n-1}\left(1+\left(\frac{b}{a}\right)^n\right)-\left(1+\frac{b}{a}\right)^n\right] \\ &= a^n\left[2^{n-1}\left(1+r^n\right)-(1+r)^n\right] \end{aligned} \tag{A.2}$$

Where $r=\frac{b}{a}$.

We require:

$$f(r)=2^{n-1}\left(1+r^n\right)-(1+r)^n \geq 0 \qquad \forall r>0. \tag{A.3}$$

Now,

$$\frac{df}{dr} = n(2r)^{n-1}-n(1+r)^{n-1}=0 \qquad \text{at } r=1. \tag{A.4}$$

$$\begin{aligned} \frac{d^2f}{dr^2} &= 2n(n-1)(2r)^{n-2}-n(n-1)(1+r)^{n-2} \\ &= n(n-1)2^{n-2}>0 \qquad \text{at } r=1. \end{aligned} \tag{A.5}$$

Giving a global minimum at $r=1$, or $a=b$.

At $r=1$, the minimum value of f is:

$$f(1)=\left(2^{n-1}\times 2\right)-2^n=0 \tag{A.6}$$

Now, to prove that:

$$(a+b)^n \leq \left(a^n+b^n\right) \qquad 0<n<1, \quad a,b>0 \tag{A.7}$$

We require:

$$(1+r)^n \leq 1+r^n \tag{A.8}$$

Where $r=\frac{b}{a}$ as before.

We have, for $0<n<1$ and $0<r<1$:

$$(1+r)^n \leq 1+r \tag{A.9}$$

$$1+r^n \geq 1+r \tag{A.10}$$

Therefore:

$$(1+r)^n \leq 1+r^n \tag{A.11}$$

As required.

In the case $r > 1$ we can rearrange:

$$(1+r)^n \leq 1 + r^n \tag{A.12}$$

To give

$$\left(1+\frac{1}{r}\right)^n \leq 1 + \left(\frac{1}{r}\right)^n \tag{A.13}$$

And the subsequent argument still holds.

In the case of $n = 1$, we have (trivially)

$$(a+b)^n = 2^{n-1}(a^n + b^n) = a^n + b^n = a + b \tag{A.14}$$

So no majorant kernel is required.

B Justification Of Majorant Kernel Multiplication Factor

The first majorant kernel considered in **Section 2.2** is

$$\left(\frac{1}{x_i}+\frac{1}{x_j}\right)^{\frac{1}{2}}\left(x_i^{\frac{1}{D_F}}+x_j^{\frac{1}{D_F}}\right)^2 \leq 2\left(x_i^{-\frac{1}{2}}+x_j^{-\frac{1}{2}}\right)\left(x_i^{\frac{2}{D_F}}+x_j^{\frac{2}{D_F}}\right) \tag{B.1}$$

For an efficient majorant kernel we expect:

$$\max_{i,j}\frac{K(x_i,x_j)}{\hat{K}(x_i,x_j)} = \max_{i,j}\frac{\left(\frac{1}{x_i}+\frac{1}{x_j}\right)^{\frac{1}{2}}\left(x_i^{\frac{1}{D_F}}+x_j^{\frac{1}{D_F}}\right)^2}{2\left(x_i^{\frac{2}{D_F}-\frac{1}{2}}+x_j^{\frac{2}{D_F}-\frac{1}{2}}+x_i^{\frac{2}{D_F}}x_j^{-\frac{1}{2}}+x_i^{-\frac{1}{2}}x_j^{\frac{2}{D_F}}\right)} = 1 \tag{B.2}$$

In fact, we can rewrite the above expression as:

$$\max_{i,j}\frac{K(x_i,x_j)}{\hat{K}(x_i,x_j)} = \max_r f(r) = \max_r\frac{\left(1+\frac{1}{r}\right)^{\frac{1}{2}}\left(1+r^{\frac{1}{D_F}}\right)^2}{2\left(1+r^{\frac{2}{D_F}-\frac{1}{2}}+r^{-\frac{1}{2}}+r^{\frac{2}{D_F}}\right)}, \quad 0<r<1. \tag{B.3}$$

We wish to find the maximum value of this function, and if it is less than 1, alter the multiplication factor to achieve maximum efficiency.

Splitting up $f(r)$ into two functions gives

$$f(r) = f_1(r)\times f_2(r) = \frac{(1+\frac{1}{r})^{\frac{1}{2}}}{1+r^{-\frac{1}{2}}}\times\frac{(1+r^{\frac{1}{D_F}})^2}{2(1+r^{\frac{2}{D_F}})} \tag{B.4}$$

These two functions are monotonically varying functions between 0 and 1, with $f_1(r)$ decreasing from 1 to $\frac{1}{\sqrt{2}}$ and $f_2(r)$ increasing from $\frac{1}{2}$ to 1.

Due to the symmetric nature of $f(r)$, there must be a turning point at $r = 1$, where $f(r) = \frac{1}{\sqrt{2}}$, thus, depending on the value of D_F we have two cases of interest:

- $f(r)$ is monotonically increasing as r varies between 0 and 1, so the maximum occurs at $r = 1$. This is characterised by the second derivative of $f(r)$ at $r = 1$ being negative.
- $f(r)$ contains a local maximum on the range $0 < r < 1$ which may be greater than $f(1)$. This is characterised by $f(1)$ being a local minimum and the second derivative at this point being positive.

So, by differentiating $f(r)$ twice with respect to r and setting $r = 1$, we obtain a function in D_F that can be solved numerically to distinguish between the two cases above. The crossover occurs at $D_F = 2.82843$, with the second derivative being negative for $D_F < 2.82843$ and positive for $D_F > 2.82843$. For non-spherical particles, typical values for D_F are between 1.7 and 2.5 [12], so the first case applies and

$$\max_r f(r) = \frac{1}{\sqrt{2}} \tag{B.5}$$

So to make the majorant kernel more efficient, it can be multiplied by $\frac{1}{\sqrt{2}}$, to give the multiplication factor as $\sqrt{2}$ instead of 2.

If spherical particles are considered, $D_F = 3$ and the second case applies. Differentiating $f(r)$ and setting it to zero to find the maximum value gives $\max_r f(r) \approx 0.708895$, so the factor of two becomes a factor of $2 \times 0.708895 = 1.41778$.

C Gillespie Algorithm

When using the stochastic solution algorithm as proposed by Gillespie [7], there is no majorant kernel, and therefore the calculation of the time increment requires a double sum of the coagulation kernel over i and j (cf. (2.5)). The most efficient way of updating this double sum after each time step is to store for each value of i the function:

$$C_i = \sum_{j=i+1}^{n} K(x_i, x_j) \qquad i = 1, \ldots, n-1 \tag{C.1}$$

These functions can be updated at each time step, in conjunction with a summation over i to give the function

$$C_0 = \sum_{i=1}^{n-1} C_i = \sum_{i=1}^{n-1} \sum_{j=1+1}^{n} K(x_i, x_j) \tag{C.2}$$

which is used in the calculation of the time increment τ. Thus, each time step requires a number of calculations that is of the order of n. (The notation C_i and C_0 follows the notation used in [7].)

The algorithm as applied to the problem of silica formation and coagulation is given as follows.

1. Generate the initial state $U^N(0) = p \in \mathcal{S}^N, \quad U^N_{in}(x) = p^{in} \in \mathcal{S}^N$.
2. Wait an exponentially distributed time step τ with parameter (c.f. (2.5),(C.1))

   $$\rho(p) = \rho_K(p) + \rho_{in}(p) = \frac{1}{N}\sum_{i=1}^{n-1} C_i + In^{in}$$

3. With probability

   $$\frac{\rho_{in}(p)}{\rho(p)}$$

   go to step 4. Otherwise go to step 5.
4. Perform a source step, i.e.
   (a) Add a cluster of size 1 to the system.
   (b) Update the stored values of C_i.
   (c) Go to step 2.
5. Perform a coagulation step, i.e.
   (a) Uniformly generate index i on the domain $1, 2, \dots, n$ and index j on the domain $1, \dots, i-1, i+1, \dots, n$.
   (b) With probability

   $$\frac{K(x_i, x_j)}{\max\limits_{i,j} K(x_i, x_j)}$$

   accept the indices i and j, otherwise return to step 5.
   (c) Remove the clusters x_i and x_j and add a cluster of size $x_i + x_j$.
   (d) Update the stored values of C_i.
   (e) Go to step 2.

References

[1] A. Eibeck and W. Wagner. An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena. *SIAM J. Sci. Comput.*, 22(3):802–821, 2000.

[2] A. Eibeck and W. Wagner. Stochastic particle approximations for smoluchowski's coagulation equation. *Ann. Appl. Probab.*, 11(4):1137–1165, 2001.

[3] M. Frenklach. Dynamics of discrete distribution for Smoluchowski coagulation model. *Journal of Colloid and Interface Science*, 108(1):237–242, 1985.

[4] M. Frenklach and H. Wang. Detailed mechanism and modeling of soot particle formation. In H. Bockhorn, editor, *Soot Formation in Combustion - Mechanisms and Models*, pages 165–192. Springer Verlag, 1994.

[5] S.K. Friedlander. *Smoke, Dust and Haze.* Wiley, New York, 1977.

[6] D. T. Gillespie. The stochastic coalescence model for cloud droplet growth. *J. Atmospheric Sci.*, 29:1496–1510, 1972.

[7] D. T. Gillespie. An exact method for numerically simulating the stochastic coalescence process in a cloud. *Journal of Atmospheric Sciences*, 32(10):1977–1989, 1975.

[8] M. J. Hounslow. A discretised population balance for continuous systems at steady state. *AIChE J.*, 36(1):106–116, 1990.

[9] M. J. Hounslow, R. L. Ryall, and V. R. Marshall. A discretized population balance for nucleation, growth and aggregation. *AIChE J.*, 34(11):1821–1832, 1988.

[10] H. M. Hulbert and S. Katz. Some problems in particle technology. A statistical mechanical formulation. *Chemical Engineering Science*, 19:555–574, 1964.

[11] M. Kraft and W. Wagner. Numerical study of a stochastic particle method for homogeneous gas phase reactions. Technical Report 570, Weierstrass Institute for Applied Analysis and Stochastics, 2000.

[12] T. Matsoukas and S. K. Friedlander. Dynamics of aerosol agglomerates formation. *J. Colloid Sci.*, 146(2):495–506, 1991.

[13] S. Panda and S. E. Pratsinis. Modeling the synthesis of aluminum particles by evaporation-condensation in an aerosol flow reactor. *Nanostructured Materials*, 5(7-8):755–767, 1995.

[14] D. Ramkrishna. *Population Balances. Theory and Applications to Particulate Systems in Engineering.* Academic Press, San Diego, 2000.

[15] A. D. Randolph and M. A. Larson. *Theory of Particulate Processes.* Academic Press, London, 1988.

[16] K. K. Sabelfeld, S. V. Rogansinsky, A. A. Kolodko, and A. I. Levykin. Stochastic algorithms for solving Smoluchovsky coagulation equation and applications to aerosol growth simulation. *Monte Carlo Methods and Applications*, 2(1):41–87, 1996.

[17] M. Smith and T. Matsoukas. Constant-number Monte Carlo simulation of population balances. *Chemical Engineering Science*, 53(9):1777–1786, 1998.

[18] M. von Smoluchowski. Drei Vorträge über Diffusion, Brownsche Molekularbewegung und Koagulation von Kolloidteilchen. *Phys. Z.*, 17:557–571, 585–599, 1916.

[19] K. T. Whitby. Determination of aerosol growth-rates in the atmosphere using lumped mode aerosol dynamics. *Journal of Aerosol Science*, 12(3):173–178, 1981.

[20] M. R. Zachariah and H. G. Semerjian. Simulation of ceramic particle formation: Comparison with *in-situ* measurements. *AIChE J*, 35(12):2003–2012, 1989.