

COUPLING ALGORITHMS FOR CALCULATING SENSITIVITIES OF SMOLUCHOWSKI'S COAGULATION EQUATION

PETER L. W. MAN^{†‡} AND JAMES R. NORRIS[§] AND ISMAËL F. BAILLEUL[§] AND MARKUS KRAFT[¶]

Abstract. In this paper, two new stochastic algorithms for calculating parametric derivatives of the solution to the Smoluchowski coagulation equation are presented. It is assumed that the coagulation kernel is dependent on these parameters. The new algorithms (called ‘Single’ and ‘Double’) work by coupling two Marcus-Lushnikov processes in such a way as to reduce the difference between their trajectories, thereby significantly reducing the variance of central difference estimators of the parametric derivatives. In the numerical results, the algorithms are shown to have a $O(1/N)$ order of convergence as expected, where N is the initial number of particles. It was also found that the Single and Double algorithms provide much smaller variances. Furthermore, a method for establishing ‘efficiency’ is considered, which takes into account the variances as well as CPU run times, and the ‘Double’ is significantly more ‘efficient’ compared to the ‘Independent’ algorithm in most cases.

Key words. Modelling, simulation, coupling, sensitivity, coagulation, Smoluchowski

AMS subject classifications. 65C05, 65C35, 68U20, 82C22

1. Introduction. The simplest of pure coagulation processes puts into play chemical species characterised by a single scalar quantity, say their mass, with values in a discrete set, say the positive integers. The evolution of the process is modelled by a differential equation which gives the time evolution of the concentration $\mu_t^\lambda(x)$ of particles of mass $x \in \mathbb{N}$. Given a real-valued function f we shall write (f, μ_t^λ) for $\sum_{x \in \mathbb{N}} f(x) \mu_t^\lambda(x)$. Quantities measured by the experimenter (such as moments) are of this form. Smoluchowski’s description of the evolution of μ_t^λ is

$$\frac{d}{dt}(f, \mu_t^\lambda) = \frac{1}{2} \sum_{x, y \geq 1} \{f(x+y) - f(x) - f(y)\} K_\lambda(x, y) \mu_t^\lambda(x) \mu_t^\lambda(y). \quad (1.1)$$

The kernel $K_\lambda(\cdot, \cdot)$ is a symmetric non-negative function which represents the rate at which a pair of particles of masses x and y coagulate to create a particle of mass $x+y$. The term $\{f(x+y) - f(x) - f(y)\}$ is the change which has occurred in the quantity (f, μ_t^λ) as a result of this coagulation. The letter λ in the kernel stands for a d -dimensional parameter. Our aim in this article is to devise a new numerical scheme for investigating how the solution μ_t^λ to Smoluchowski equation depends on λ . We shall concentrate on the case of a one dimensional parameter.

[†]Department of Chemical Engineering and Biotechnology, University of Cambridge, New Museums Site, Pembroke Street, Cambridge, CB2 3RA, UK.

[‡]A preliminary form of this paper appeared in conference proceedings as the following article: P. L. W. Man, M. Kraft and J. R. Norris, Coupling Algorithms for Calculating Sensitivities of Population Balances, Numerical Analysis and Applied Mathematics, AIP Conference Proceedings, 1048 (2008), pp. 927–930.

[§]Department of Pure Mathematics and Mathematical Statistics, Centre for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WB, UK.

[¶]Corresponding author. Department of Chemical Engineering and Biotechnology, University of Cambridge, New Museums Site, Pembroke Street, Cambridge, CB2 3RA, UK (mk306@cam.ac.uk). This research was supported by EPSRC grant EP/E01772X/1.

There is a large amount of literature concerning the solving of the continuous particle sized version of eq. (1.1) and its many variations such as particle inception, surface growth, sintering, and fragmentation [1]-[9]. However, little is devoted to a systematic method of sensitivity analysis other than merely simulating the physical system in question for various parameter values to measure the change in some quantity as a result of the parameter change. In this paper, we wish to conduct sensitivity analysis by explicitly calculating the parametric derivative of equation (1.1).

There are only a few sources which report on this approach [10]-[13]. One method involves a weighted particle method, which assigns to each particle a weight with the interpretation of the number of physical particles it represents. This particular method [10, 12] considers a finite difference approach where both particle systems (with different parameters) are simulated together, the only difference being the particles' weights. A potentially very powerful method based on the Lagrangian formalism is considered in [11]. The idea is to consider an adjoint equation which solves for the parametric derivative directly rather than eq. (1.1). This allows the solving of the derivative for all values of the parameter simultaneously.

The aim of this paper is to present two new stochastic algorithms for the calculation of parametric derivatives of eq. (1.1) with emphasis on variance reduction. These algorithms are based on the simple Marcus-Lushnikov process, but we consider how two such processes with different parameters can be solved simultaneously in order to reduce the estimator variance. These algorithms are presented in section 2. Their mathematical formalisation is detailed in section 2.3.3, in which we describe how this formalism can be used to justify that these algorithms do indeed provide approximations of the sensitivity. The quality of these approximations is investigated in section 3 where numerical results are analysed.

2. Central difference estimation of parametric derivatives. As Marcus-Lushnikov's process is our main ingredient, let us recall first what it is. Dropping the index λ , equation (1.1) makes it clear that μ_t should be seen as a non-negative discrete measure on \mathbb{N} and $(f, \mu_t) = \sum_{x \in \mathbb{N}} f(x) \mu_t(x)$ as the integral of f against μ_t . In Marcus-Lushnikov's approach, μ_t is approximated by a random finite measure of the form¹

$$\mu_t^N = \frac{1}{N} \sum_{i=1}^n \delta_{x_i(t)}$$

whose dynamics are that of a Markov chain with state space

$$Q_N := \left\{ \mu \in \mathcal{M}(\mathbb{N}) \mid \mu = \frac{1}{N} \sum_{i=1}^n \delta_{x_i}, x_i \in \mathbb{N} \forall i, n = 1, 2, \dots \right\} \quad (2.1)$$

where $\mathcal{M}(\mathbb{N})$ is the space of all measures on \mathbb{N} . We shall talk of each $x_i(t)$ or $\delta_{x_i(t)}$ as a particle of the system at time t . Start from $\mu_0^N = \frac{1}{N} \sum \delta_{x_i}$; associate to each pair (x_i, x_j) of distinct particles an exponential random time T_{ij} with parameter $\frac{K(x_i, x_j)}{N}$, independent of the other exponential times, and set

$$T := \min\{T_{ij}; i < j\}.$$

¹ δ_x is a Dirac mass at $x \in \mathbb{N}$.

The process μ_t^N remains constant on the time interval $[0, T)$ and has a jump at time T . If $T = T_{pq}$, set

$$\mu_T^N = \mu_0^N + (\delta_{x_p+x_q} - \delta_{x_p} - \delta_{x_q});$$

this operation amounts to removing the particles x_p and x_q from the system and adding the particle $x_p + x_q$. The dynamics then starts afresh.

We shall write $\mu_t^{\lambda; N}$ for the Marcus-Lushnikov process corresponding to the kernel K_λ . An obvious way of estimating the sensitivity is to approximate it by the (random) ratio $(\mu_t^{\lambda+\frac{1}{2}\epsilon; N} - \mu_t^{\lambda-\frac{1}{2}\epsilon; N})/\epsilon$. No *a priori* independence or dependence between $\mu_t^{\lambda+\frac{1}{2}\epsilon; N}$ and $\mu_t^{\lambda-\frac{1}{2}\epsilon; N}$ is imposed. We are mainly interested in this article in producing a stochastic approximation of the sensitivity with a low variance. We shall thus try to minimise the variances

$$\text{Var}\left(\frac{\mu_t^{\lambda+\frac{1}{2}\epsilon; N}(x) - \mu_t^{\lambda-\frac{1}{2}\epsilon; N}(x)}{\epsilon}\right)$$

as much as we can for all values of x . For that purpose we shall couple the evolution of the two Marcus-Lushnikov processes so as to keep them as close as possible, noting that increasing the covariance between $\mu_t^{\lambda+\frac{1}{2}\epsilon; N}(x)$ and $\mu_t^{\lambda-\frac{1}{2}\epsilon; N}(x)$ decreases the variance. For notational simplicity, we rename $\mu_t^{\lambda\pm\frac{1}{2}\epsilon; N}(x)$ as $\mu_t^{\pm; N}(x)$. It is perfectly possible to describe both trajectories by the \mathbb{R}^2 -valued discrete measure $(\mu_t^{+,N}, \mu_t^{-,N}) \in Q_N^2$, but to encapsulate the following coupling, we consider the following approach.

2.1. Coupling. Here is an example of coupling with framework the unit square of the plane. Denote by $f(x, y)$ any probability density on the square, and consider the problem of minimizing $I := \int_0^1 \int_0^1 |x - y| f(x, y) dx dy$, subject to the condition that the two *marginals* of the probability $f(x, y) dx dy$ on both the x and y axes are uniform². Any measure on the square satisfying this condition is said to realise a *coupling* between the uniform probability on the x -segment $[0, 1]$ and the uniform probability on the y -segment $[0, 1]$ (regardless of the above optimisation problem). The probability $dx dy$ is such a coupling, but it does not minimise I . This minimum is attained for the singular probability on the square with support on the diagonal, and uniform on it, signifying maximum correlation between the x and y axes.

Our framework is more complicated than above as the role of $[0, 1]$ is now played by the set of particle approximations, i.e. trajectories $(\{\mu_t^{\pm; N}\}_{t \in [0, t_{\text{end}}]})$ (for some final time t_{end}) with values in the set Q_N of finite measures of the form $\frac{1}{N} \sum_i \delta_{x_i}$; but the basic idea is the same. The minimisation of I is replaced by the minimisation of difference between trajectories as seen in the following paragraphs.

Denote by X_t^- and X_t^+ the set of particles³ from $\mu_t^{+,N}$ and $\mu_t^{-,N}$, respectively. (The more particles X_t^- and X_t^+ have in common the closer the particle systems are, and thus increasing the correlation; the set

$$X_t^\odot := X_t^- \cap X_t^+ \tag{2.2}$$

²That is the marginal $\int_0^1 f(x, y) dy = 1$ for each $x \in [0, 1]$, and the marginal $\int_0^1 f(x', y) dx' = 1$ for each $y \in [0, 1]$.

³For a measure $\mu = \frac{1}{N} \sum_{i=1}^n \delta_{x_i}$, the set of particles is $\frac{1}{N}(\delta_{x_1}, \dots, \delta_{x_n})$ for $x_i \in \mathbb{N}$ for all i , thus allowing multiple particles with the same size to exist in the set.

is made up of those particles in common⁴, with corresponding measure $\mu_t^{\ominus,N} \in Q_N$ being the sum of those particles in $X_t^{\ominus,N}$. Similarly, we consider

$$X_t^{\oplus} := X_t^+ \setminus X_t^{\ominus} \quad , \quad X_t^{\ominus} := X_t^- \setminus X_t^{\oplus} \quad (2.3)$$

which are the set of those particles present in X_t^+ but *not* in X_t^- (and the other way round respectively)—we wish to minimise the numbers of particles in these sets. We denote $\mu_t^{\oplus,N}, \mu_t^{\ominus,N} \in Q_N$ as the measures corresponding to $X_t^{\oplus}, X_t^{\ominus}$. From this definition, we can recover $\mu_t^{+,N}$ from $\mu_t^{+,N} = \mu_t^{\oplus,N} + \mu_t^{\ominus,N}$ and similarly for $\mu_t^{-,N}$. See that the information held in (X_t^+, X_t^-) is the same as that held in $(X_t^{\oplus}, X_t^{\ominus}, X_t^{\ominus})$ —thus we seek to describe the full stochastic process by the \mathbb{R}^3 -valued measure $(\mu_t^{\oplus,N}, \mu_t^{\ominus,N}, \mu_t^{\ominus,N}) \in Q_N^3$ rather than $(\mu_t^{+,N}, \mu_t^{-,N}) \in Q_N^2$. Note that the coagulations for X_t^{\pm} particles are governed by the rates determined by $\lambda \pm \frac{1}{2}\epsilon$, and that certain coagulations, such as between a particle in X_t^{\oplus} and a particle in X_t^{\ominus} , cannot occur in the X_t^- set because the particle in X_t^{\oplus} does not exist in X_t^- . Furthermore, coagulations between a particle in X_t^{\oplus} and a particle in X_t^{\ominus} cannot occur at all.

The first version of our algorithm, called **Single Coupling Algorithm**, tries to keep the number of particles from X_t^{\ominus} as large as possible, imposing that (as much as possible) when two particles, *both* of which are present in X_t^- and X_t^+ , are chosen to coagulate in one of these systems, they also coagulate in the other. The resulting particle must also be present in both X_t^- and X_t^+ , thus helping to keep X_t^{\ominus} large. Of course, as the coagulation rates in X_t^- and X_t^+ differ, we cannot prevent a coagulation event of the above kind from happening in only one of the systems; we can however minimise the rate at which it happens.

The **Double Coupling Algorithm** is a refinement of the previous one in which we try to make the creation of particles of X_t^{\oplus} and X_t^{\ominus} as rare as possible. In addition to the above coupling, it considers what happens when a particle from X_t^{\ominus} coagulates with a particle from $X_t^{\oplus/\ominus}$ (which can only occur in X_t^{\pm} set as mentioned earlier). In such an event, the same particle from X_t^{\ominus} can be used in a coagulation event with a particle from $X_t^{\oplus/\ominus}$. Out of the three particles from X_t^{\oplus} , X_t^{\ominus} and X_t^{\ominus} , the X_t^{\ominus} particle contributes size to the other two particles, and is itself removed. More details about both couplings are described in detail later.

One ultimately expects that given enough time, the two systems will behave almost independently (i.e. there will be few particles in X_t^{\ominus}), but the hope is that the divergence in their trajectories is slow enough over the time span of interest. The simulation of the sensitivity using two independent Marcus-Lushnikov processes will be referred to as the **Independent Algorithm**; it will be used for comparison with the other algorithms.

Labelling. The usage of the triple measure $(\mu_t^{\oplus,N}, \mu_t^{\ominus,N}, \mu_t^{\ominus,N}) \in Q_N^3$ captures the similarities and differences between the $\mu_t^{+,N}$ and $\mu_t^{-,N}$ trajectories. This is the way we store the particles in a computer—particles present in $X_t^{\oplus/\ominus}$ are attached with \oplus and \ominus labels respectively. The resulting possible Markov steps are given in Table 2.2.

Note that there is a certain degeneracy in the state space Q_N^3 —if there exist particles $\frac{1}{N}\delta_x$ in **both** $\mu_t^{\oplus,N}$ and $\mu_t^{\ominus,N}$ then these two particles can be removed and

⁴The intersection here is used in the multiset sense - if there are s_- particles of size $x \in \mathbb{N}$ in X_t^- and s_+ particles also of size x in X_t^+ , then the intersection contains $\min\{s_-, s_+\}$ particles of size x . Also, X_t^{\oplus} in eq. (2.3) then has $\max\{s_+ - s_-, 0\}$ particles of size x .

TABLE 2.1
Particle labels and their meaning

Label	Meaning
\oplus_x	a real particle (of size x) present <i>only</i> in X_t^+ i.e. present in X_t^\oplus .
\ominus_x	a real particle (of size x) present <i>only</i> in X_t^- i.e. present in X_t^\ominus .
\odot_x	a computational particle (of size x) present in X_t^\odot i.e. a pair of identical real particles, one in X_t^+ and one in X_t^- .

TABLE 2.2

Possible events described using the labelling notation. Events of type 1 represent the Single Coupling (or rejection of), those of type 2 represent the Double Coupling (or rejection of), and those of type 3 without couplings.

Type	Event	Explanation
1 (a) 1 (b) 1 (c)	$\odot_x + \odot_y \rightarrow \begin{cases} \odot_{x+y} \\ \ominus_x + \ominus_y + \oplus_{x+y} \\ \oplus_x + \oplus_y + \ominus_{x+y} \end{cases}$	if occurs in both X_t^- and X_t^+ if occurs only in X_t^+ if occurs only in X_t^-
2(a)	$\oplus_x + \odot_y + \ominus_z \rightarrow \oplus_{x+y} + \ominus_{y+z}$	See Double Coupling algorithm explanation. Only occurs in Double Coupling algorithm.
2(b)	$\oplus_x + \odot_y \rightarrow \oplus_{x+y} + \ominus_y$	This represents a coagulation between a pair of particles from (X_t^\oplus, X_t^\odot) , so the \oplus particle must increase in size and the \odot particle becomes a \ominus (since this particle is no longer in the X_t^+ system).
2(c)	$\ominus_x + \odot_y \rightarrow \ominus_{x+y} + \oplus_y$	Same logic as 2(b)—this reaction can only happen in the X_t^- system.
3(a)	$\oplus_x + \oplus_y \rightarrow \oplus_{x+y}$	Particles present in X_t^\oplus coagulate.
3(b)	$\ominus_x + \ominus_y \rightarrow \ominus_{x+y}$	As in event type 3(a) except for X_t^\ominus .
reject	$\oplus_x + \ominus_y$	This coagulation cannot occur since each of the particles cannot ‘see’ the other.

a single particle $\frac{1}{N}\delta_x$ added to $\mu_t^{\odot, N}$. Note that this **cleanup operation** is not a Markov jump but simply a computational enforcement of the definition of X_t^\odot , and does not affect X_t^\pm at all.

2.2. Single Coupling system. To be consistent with the above \pm notations, we shall write K^- for the kernel $K_{\lambda-\frac{1}{2}\epsilon}$ and K^+ for the kernel $K_{\lambda+\frac{1}{2}\epsilon}$.

2.2.1. The idea. The coupling procedure is implemented using a **majorant kernel**. This is a symmetric non-negative function $\widehat{K}(\cdot, \cdot)$ satisfying $\widehat{K}(\cdot, \cdot) \geq K^\pm(\cdot, \cdot)$. Run both systems X_t^- and X_t^+ at the *same rate*, given by \widehat{K} ; a coagulation happening at that rate is called *potential*. If a potential coagulation between particles of sizes x_i

and x_j *only* happens in X_t^\pm , perform it with the respective probabilities

$$p_\oplus = \frac{K^+(x_i, x_j)}{\widehat{K}(x_i, x_j)} \quad \text{or} \quad p_\ominus = \frac{K^-(x_i, x_j)}{\widehat{K}(x_i, x_j)} \quad , \quad (2.4)$$

otherwise leave the system as it is. This way each system behaves as a Marcus-Lushnikov process with the correct rate.

The coupling itself takes place when the potential coagulation involves a pair of \odot -particles (where the coagulation can potentially occur in both X_t^+ and X_t^- systems). In this case, the *same* uniform random variable on $(0, 1)$ is used to decide whether or not we perform the coagulation event in each system. In other cases the potential coagulation involves only one system. More explicitly, denote by (x_i, x_j) the sizes of the pair of \odot -particles (possibly) involved in the potential coagulation event.

Set

(i) $K_S^0(x_i, x_j) := \min\{K^+(x_i, x_j), K^-(x_i, x_j)\}$ — rate at which a coagulation of the type $\odot_{x_i} + \odot_{x_j} \rightarrow \odot_{x_i+x_j}$ occurs,

(ii) $\Delta_S^+(x_i, x_j) := \max\{K^+(x_i, x_j) - K^-(x_i, x_j), 0\}$ — rate at which a coagulation of the type $\odot_{x_i} + \odot_{x_j} \rightarrow \ominus_{x_i} + \ominus_{x_j} + \oplus_{x_i+x_j}$ occurs,

(iii) $\Delta_S^-(x_i, x_j) := \max\{K^-(x_i, x_j) - K^+(x_i, x_j), 0\}$ — rate at which a coagulation of the type $\odot_{x_i} + \odot_{x_j} \rightarrow \oplus_{x_i} + \oplus_{x_j} + \ominus_{x_i+x_j}$ occurs.

Figure 2.1 gives a schematic picture of the procedure.

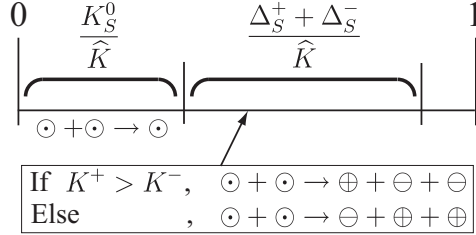


FIG. 2.1. Rate correction for the Single Coupling — generate $\mathbf{U} \sim U(0, 1)$ and perform jump event according to the given probabilities.

2.2.2. The Algorithm. Recall the different types of coagulation that can happen in the Single Coupling algorithm; they were named 1, 2b, 2c, 3a, and 3b. Each of the following rates $\widehat{\rho}_k$ represents the majorant rate at which a *potential* coagulation of type $k \in \{1, 2b, 2c, 3a, 3b\}$ happens⁵.

$$\widehat{\rho}_1 := \frac{1}{N} \sum_{(\odot, \odot)} \widehat{K}(\odot, \odot) \quad , \quad (2.5a)$$

$$\widehat{\rho}_{2b} := \frac{1}{N} \sum_{(\odot, \oplus)} \widehat{K}(\odot, \oplus) \quad , \quad \widehat{\rho}_{2c} := \frac{1}{N} \sum_{(\odot, \ominus)} \widehat{K}(\odot, \ominus) \quad , \quad (2.5b)$$

$$\widehat{\rho}_{3a} := \frac{1}{N} \sum_{(\oplus, \oplus)} \widehat{K}(\oplus, \oplus) \quad , \quad \widehat{\rho}_{3b} := \frac{1}{N} \sum_{(\ominus, \ominus)} \widehat{K}(\ominus, \ominus) \quad . \quad (2.5c)$$

⁵We use an abuse of notation. $\sum_{(\odot, \odot)} \widehat{K}(\odot, \odot)$ is, for example, the sum of the $\widehat{K}(x_i, x_j)$ for all the pairs (x_i, x_j) belonging to X_t^\odot . Similar meaning is given to the other $\widehat{\rho}_k$.

The total majorant rate of potential coagulation is defined as

$$\widehat{\rho} := \widehat{\rho}_1 + \widehat{\rho}_{2b} + \widehat{\rho}_{2c} + \widehat{\rho}_{3a} + \widehat{\rho}_{3b}. \quad (2.6)$$

Adopting this notation, one can read the details in Algorithm 1.

2.2.3. Implementation and Complexity. The main implementation issue deals with how step 3 of Algorithm 1 is performed. We make the assumption that \widehat{K} can be expressed as (for some A):

$$\widehat{K}(x_i, x_j) =: \sum_{\alpha=1}^A f_{\alpha}(x_i)g_{\alpha}(x_j); \quad (2.8)$$

such a form of \widehat{K} enables an easy performance of step 3. As an example of the factorisability condition, the additive kernel $\widehat{K}(x_i, x_j) := \lambda(x_i + x_j)$ can be expressed as $\widehat{K}(x_i, x_j) = \lambda \cdot x_i + x_j \cdot \lambda$, implying that $f_1(x) = \lambda$, $g_1(x) = x$, $f_2(x) = x$ and $g_2(x) = \lambda$. The assumption is not so strict—one need only find a *majorant* kernel with this feature. More details on this majorant kernel factorisation can be found in the articles by Eibeck and Wagner [1, 2] and Kraft and coworkers [3, 8, 14].

To see how step 3 is performed, eq. (2.7) can be written as:

$$\frac{\widehat{K}(x_i, x_j)}{2N\widehat{\rho}} = \frac{\sum_{\alpha} f_{\alpha}(x_i)g_{\alpha}(x_j)}{\sum_{p \neq q} \sum_{\alpha'} f_{\alpha'}(x_p)g_{\alpha'}(x_q)} \quad (2.9a)$$

$$= \sum_{\alpha} \left[\frac{1}{\sum_{p \neq q} \sum_{\alpha'} f_{\alpha'}(x_p)g_{\alpha'}(x_q)} \frac{f_{\alpha}(x_i)g_{\alpha}(x_j)}{1} \right] \quad (2.9b)$$

$$= \sum_{\alpha} \left[\frac{\sum_{p \neq q} f_{\alpha}(x_p)g_{\alpha}(x_q)}{\sum_{p \neq q} \sum_{\alpha'} f_{\alpha'}(x_p)g_{\alpha'}(x_q)} \left(\frac{f_{\alpha}(x_i)}{\sum_p f_{\alpha}(x_p)} \frac{g_{\alpha}(x_j)}{\sum_{q; q \neq p} g_{\alpha}(x_q)} \right) \right] \quad (2.9c)$$

where the last equality follows since $\sum_{i \neq j} f_{\alpha}(x_i)g_{\alpha}(x_j) = \sum_i f_{\alpha}(x_i) \sum_{j; j \neq i} g_{\alpha}(x_j)$. Thus the user must choose α according to the first fraction of eq. (2.9c) whilst the last two fractions are for the generation of the pair of particles. The advantage of these methods is two-fold: first we can store the values $f_{\alpha}(x_i)$ and $g_{\alpha}(x_i)$ in ‘binary’ tree structures which also stores their sums (over i). This allows efficient generation from the distributions

$$\frac{f_{\alpha}(x_i)}{\sum_p f_{\alpha}(x_p)} \quad \text{and} \quad \frac{g_{\alpha}(x_i)}{\sum_q g_{\alpha}(x_q)}$$

respectively. Also, updating the values in this data structure is efficient. If the number of stochastic particles in a binary tree is n , then the complexity of updating and generating particle operations take $O(\log n)$ steps. Furthermore, the generation of a particle pair is simple—the factorisation allows one to generate each particle in the pair separately meaning that generation of the pair of particles is $O(\log n)$ rather than $O(n^2)$.

In Step 14 of Algorithm 1, a search of particles for cleanup is required for each iteration. This can be achieved by maintaining linked lists of information about where particles of certain size and label can be found on the particle ensemble list. In short, the Single Coupling algorithm may be faster than the ‘Independent’ algorithm since we need to simulate for one particle ensemble rather than two. On the other hand,

Algorithm 1: Single Coupling algorithm

For simplicity of exposition, we suppose particles from $X_t^\oplus, X_t^\ominus, X_t^\ominus$ are all stored in one single array, which is indexed by i and j below.

1 Set $t = 0$. Set all N initial particles to have \odot labels.

while $t < t_{end}$ **do**

2 Generate a realisation of the holding time $\Delta t \sim \text{Exp}(\hat{\rho})$ where $\hat{\rho}$ is specified in eq. 2.5 and eq. 2.6, and set $t \leftarrow t + \Delta t$.

The following step simultaneously chooses the process $k \in \{1, 2b, 2c, 3a, 3b\}$ and the particle pair with indices (i, j) which have the correct labels for the process of type k .

3 Generate an unordered pair of particles with indices (i, j) for potential coagulation according to the **index distribution**

$$\frac{\hat{K}(x_i, x_j)}{2N\hat{\rho}} = \frac{\hat{\rho}_k \hat{K}(x_i, x_j)}{\hat{\rho} 2N\hat{\rho}_k}. \quad (2.7)$$

switch the value of k chosen **do**

4 **case** $k = 1$

This case represents the Single Coupling part of the algorithm and the following steps (5-9) exactly identify with Figure 2.1.

5 Generate random variable $U \sim U(0, 1)$.

if $0 < \hat{K}U \leq K_S^0$ **then**

6 perform event type 1a: $\odot_{x_i} + \odot_{x_j} \rightarrow \odot_{x_i+x_j}$.

else if $K_S^0 < \hat{K}U \leq K_S^0 + \Delta_S^+ + \Delta_S^-$ **then**

if $K^+ > K^-$ **then**

7 perform event type 1b: $\odot_{x_i} + \odot_{x_j} \rightarrow \ominus_{x_i} + \ominus_{x_j} + \oplus_{x_i+x_j}$.

else

8 perform event type 1c: $\odot_{x_i} + \odot_{x_j} \rightarrow \oplus_{x_i} + \oplus_{x_j} + \ominus_{x_i+x_j}$.

endif

endif

9 **break**.

For following cases 2b, 2c, 3a and 3b, assume the labels on the particle pair (i, j) match with those as described in steps 10-13, otherwise swap the indices i and j . Let 'w. p.' mean 'with probability'. Recall also the definitions of p_\oplus and p_\ominus from eq. (2.4).

10 **case** $k = 2b$ w. p. p_\oplus , perform $\oplus_{x_i} + \odot_{x_j} \rightarrow \oplus_{x_i+x_j} + \ominus_{x_j}$. **break**.

11 **case** $k = 2c$ w. p. p_\ominus , perform $\ominus_{x_i} + \odot_{x_j} \rightarrow \ominus_{x_i+x_j} + \oplus_{x_j}$. **break**.

12 **case** $k = 3a$ w. p. p_\ominus , perform $\ominus_{x_i} + \oplus_{x_j} \rightarrow \ominus_{x_i+x_j}$. **break**.

13 **case** $k = 3b$ w. p. p_\oplus , perform $\oplus_{x_i} + \oplus_{x_j} \rightarrow \oplus_{x_i+x_j}$. **break**.

endswitch

14 If a coagulation occurred, for **each** particle that has just been involved in the coagulation, or newly formed, search for a particle of the same size of the 'opposite sign'. If there is such a particle (of size x , say), perform a *cleanup operation*: $\ominus_x + \oplus_x \rightarrow \odot_x$.

15 **if** there is only one particle left in the system **then** STOP.

endwhile

16 STOP.

the cleanup procedure in the Single Coupling requires extra storage of information, and computational time to update this information.

The Single Coupling algorithm is good for the initial prevention of creation of \oplus and \ominus particles, however, as the \ominus particle numbers decrease over time, the Single Coupling should become less effective. This motivates the Double Coupling procedure which is designed to minimise the rate of creation of \oplus and \ominus particles for later times.

2.3. Double Coupling system. The aim of the Double Coupling algorithm is to try to minimise the rate at which particles of type \ominus or \oplus are created; it was briefly described in section 2.1. Figure 2.2 presents a pictorial illustration of this coupling.

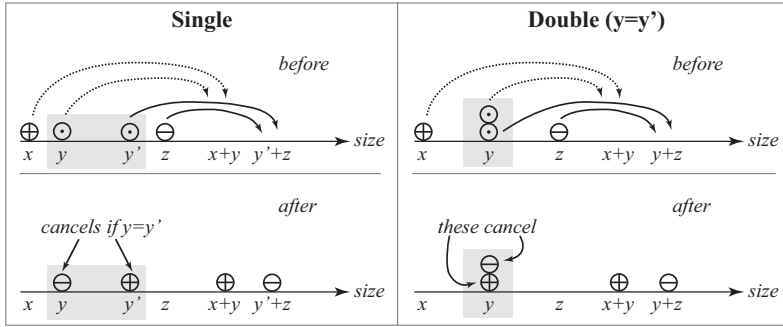


FIG. 2.2. Pictorial explanation of the Double Coupling algorithm.

More formally, we:

(i) choose a \ominus particle as the common particle for the $\ominus + \ominus$ and $\oplus + \ominus$ coagulations. This is done at the maximum potential rate at which the two reactions can occur simultaneously (for a common \ominus particle)

$$\max \left\{ \sum_{\oplus} \hat{K}(\oplus, \ominus), \sum_{\ominus} \hat{K}(\ominus, \ominus) \right\},$$

(ii) choose a \ominus (for the $\ominus + \ominus$ coagulation) and a \oplus particle (for $\oplus + \ominus$) with respective distributions

$$\frac{\hat{K}(\ominus, \ominus)}{\sum_{\ominus} \hat{K}(\ominus, \ominus)} \quad \text{and} \quad \frac{\hat{K}(\oplus, \ominus)}{\sum_{\oplus} \hat{K}(\oplus, \ominus)}$$

(iii) rejection steps are performed to correct the rates according to whether the coagulation happens in each of the X_t^+ and X_t^- systems.

2.3.1. The algorithm. The algorithm is the same as the Single Coupling version, except that we merge processes 2b and 2c into a new process 2 whose majorant rate is $\hat{\rho}_2 := \hat{\rho}_{2b} + \hat{\rho}_{2c}$. In describing the algorithm, and given a particular \ominus particle, we write $\hat{T}(+, \ominus)$ for $\sum_{\oplus} \hat{K}(\oplus, \ominus)$, and $\hat{T}(-, \ominus)$ for $\sum_{\ominus} \hat{K}(\ominus, \ominus)$; the maximum of these two quantities is denoted by $\hat{T}(\vee, \ominus)$. See Algorithm 2 for the description of the Double Coupling algorithm. In the next paragraph we directly check that the algorithm produces coagulations with the correct rates.

Algorithm 2: Double Coupling algorithm - only the part which differs from Algorithm 1.

The Double Coupling algorithm is almost identical to the Single Coupling algorithm, but is modified by replacing cases $k = 2b$ and $k = 2c$ (steps 10 and 11) in Algorithm 1 with a new combined case $k = 2$ containing the following steps (ignore any particle pair already chosen).

- 1 Choose a \odot particle using the distribution

$$\frac{\widehat{T}(+, \odot) + \widehat{T}(-, \odot)}{\sum_{\odot} [\widehat{T}(+, \odot) + \widehat{T}(-, \odot)]} = \frac{\widehat{T}(+, \odot) + \widehat{T}(-, \odot)}{N\widehat{\rho}_2} . \quad (2.10)$$

The first rejection step—the following step is there purely to transform the total potential rate of process 2 into $\sum_{\odot} \widehat{T}(v, \odot)$ from $\widehat{\rho}_2$.

- 2 With probability

$$\frac{\widehat{T}(v, \odot)}{\widehat{T}(+, \odot) + \widehat{T}(-, \odot)}$$

we continue, else reject by going to Step 8.

- 3 Choose a \oplus and a \ominus particle according to the respective distributions

$$\frac{\widehat{K}(\oplus, \odot)}{\widehat{T}(+, \odot)} \quad \text{and} \quad \frac{\widehat{K}(\ominus, \odot)}{\widehat{T}(-, \odot)} . \quad (2.11)$$

We now have generated a triplet of particles $(\oplus_x, \odot_y, \ominus_z)$, say. Define the probabilities of the (\oplus_x, \odot_y) and (\ominus_z, \odot_y) coagulations occurring in X_t^+, X_t^- respectively as:

$$p_{\oplus+\odot} := \frac{\widehat{T}(+, \odot) K^+(\oplus, \odot)}{\widehat{T}(v, \odot) \widehat{K}(\oplus, \odot)} \quad \text{and} \quad p_{\ominus+\odot} := \frac{\widehat{T}(-, \odot) K^-(\ominus, \odot)}{\widehat{T}(v, \odot) \widehat{K}(\ominus, \odot)} . \quad (2.12)$$

The second rejection (steps 4-7)—this occurs in an almost identical fashion to Figure 2.1, just with different rates.

- 4 Generate random variable $\mathbf{U} \sim U(0, 1)$.
if $U < \min\{p_{\oplus+\odot}, p_{\ominus+\odot}\}$ **then**
 5 perform event type 2a: $\oplus_x + \odot_y + \ominus_z \rightarrow \oplus_{x+y} + \ominus_{y+z}$.
else if $\min\{p_{\oplus+\odot}, p_{\ominus+\odot}\} \leq U < \max\{p_{\oplus+\odot}, p_{\ominus+\odot}\}$ **then**
 if $p_{\oplus+\odot} > p_{\ominus+\odot}$ **then**
 6 perform event type 2b: $\oplus_x + \odot_y \rightarrow \oplus_{x+y} + \ominus_y$.
 else
 7 perform event type 2c: $\ominus_z + \odot_y \rightarrow \ominus_{y+z} + \oplus_y$.
 endif
endif
 8 Go to Step 14 of the Single Coupling algorithm (Algorithm 1) .
-

Double Coupling algorithm rates. In the specification of the state space earlier, we recall that $\mu_t^{\oplus,N}, \mu_t^{\ominus,N}, \mu_t^{\ominus,N}$ are the empirical measures for the \ominus, \oplus, \ominus particles at time t respectively. Given a particular \ominus particle of mass y , the total majorant potential rate at which this particle reacts with any \oplus particle is equal to

$$\widehat{T}(+, y) := \sum_{x \geq 1} \widehat{K}(x, y) \mu_t^{\oplus,N}(x). \quad (2.13)$$

Also, the majorant rate at which a coagulation event of the form $\oplus_x + \ominus_y + \ominus_z \rightarrow \oplus_{x+y} + \ominus_{y+z}$ occurs is⁶

$$\begin{aligned} K_D^0(x, y, z) := & \underbrace{\widehat{\rho}_2}_{\text{Total rate of process 2}} \cdot \underbrace{\sum_{y \geq 1} \left[\widehat{T}(+, y) + \widehat{T}(-, y) \right] \mu_t^{\ominus,N}(y)}_{\text{Choose } \ominus \text{ particle}} \cdot \underbrace{\frac{\widehat{T}(\vee, y)}{\widehat{T}(+, y) + \widehat{T}(-, y)}}_{\text{1st Rejection step}} \\ & \cdot \underbrace{\frac{\widehat{K}(x, y)}{\widehat{T}(+, y)} \cdot \frac{\widehat{K}(z, y)}{\widehat{T}(-, y)}}_{\text{Choose a } \oplus_x \text{ Choose a } \ominus_z} \cdot \underbrace{\min \left\{ \frac{K^-(z, y) \widehat{T}(-, y)}{\widehat{K}(z, y) \widehat{T}(\vee, y)}, \frac{K^+(x, y) \widehat{T}(+, y)}{\widehat{K}(x, y) \widehat{T}(\vee, y)} \right\}}_{\text{Probability of rejecting neither coagulation}} \end{aligned}$$

which simplifies to

$$K_D^0(x, y, z) = \min \{ r^-(x, y, z), r^+(x, y, z) \}, \quad (2.14)$$

where

$$r^-(x, y, z) := \frac{\widehat{K}(x, y) K^-(y, z)}{\widehat{T}(+, y)}, \quad r^+(x, y, z) := \frac{\widehat{K}(y, z) K^+(x, y)}{\widehat{T}(-, y)}. \quad (2.15)$$

Similarly (and dropping the (x, y, z) for convenience), the rate at which **only** the X_t^- reaction occurs is

$$\begin{aligned} \Delta_D^- &:= [\max \{ r^-, r^+ \} - \min \{ r^-, r^+ \}] \mathbb{1}_{r^+ < r^-} \\ &= \max \{ r^- - r^+, 0 \}, \end{aligned} \quad (2.16a)$$

and the rate at which only the X_t^+ reaction occurs is

$$\Delta_D^+ := \max \{ r^+ - r^-, 0 \}. \quad (2.16b)$$

As verification for the above rate expressions, we note that the rate at which a pair of particles (\oplus_x, \ominus_y) coagulates (in X_t^+) is

$$\begin{aligned} \sum_{z \geq 1} (K_D^0(x, y, z) + \Delta_D^+(x, y, z)) \mu_t^{\ominus,N}(z) &= \sum_{z \geq 1} r^+(x, y, z) \mu_t^{\ominus,N}(z) \\ &= K^+(x, y) \sum_{z \geq 1} \frac{\widehat{K}(y, z)}{\widehat{T}(-, y)} \mu_t^{\ominus,N}(z) \\ &= K^+(x, y). \end{aligned}$$

A similar computation is made to check that (\ominus_y, \ominus_z) coagulate in X_t^- at rate $K^-(y, z)$.

⁶We use the index D for ‘‘Double’’; this distinguishes the quantities to be introduced from the similar ones introduced above for the Single Coupling algorithm.

2.3.2. Implementation and Complexity. Looking at Step 3 of the Single Coupling algorithm where the particle pair is chosen, and simultaneously the process k , we note that the combined process $\hat{\rho}_2$ for the Double Coupling is chosen by choosing either a (\odot, \ominus) or a (\odot, \oplus) particle pair. Either way, a \odot particle is automatically chosen with the correct distribution in equation 2.10 and one of \oplus and \ominus is also automatically chosen with the correct distribution specified in equation 2.11 respectively. This only leaves the remaining particle left to be chosen.

The complexity of this algorithm should be similar to that of the Single Coupling, except that the combined process $\hat{\rho}_2$ requires slightly more work than in the Single Coupling. However, the Double Coupling hopefully reduces the number of \oplus and \ominus and therefore would reduce the total rate of reactions. Consequently, there might be slightly fewer coagulation events in total.

2.3.3. Limit coupled processes. Recall the definitions of the measures $\mu_t^{\oplus, N}$, $\mu_t^{\odot, N}$, $\mu_t^{\ominus, N}$ as given in section 2.1. In the same way as one can prove that the Marcus-Lushnikov process converges to the solution of Smoluchowski equation (when it is unique)⁷, one can prove a similar result for the triple of *stochastic* processes $(\mu_t^{\oplus, N}, \mu_t^{\odot, N}, \mu_t^{\ominus, N})$. The limiting object $(\mu_t^{\oplus}, \mu_t^{\odot}, \mu_t^{\ominus})$ is a *deterministic* non-negative measure-valued path. Given three bounded functions f, g, h , it satisfies the system⁸

$$\begin{aligned} \frac{d}{dt}(f, \mu_t^{\odot}) &= \frac{1}{2} \sum_{x, y \geq 1} [f(x+y) - f(x) - f(y)] K_S^0(x, y) \mu_t^{\odot}(x) \mu_t^{\odot}(y) \\ &\quad - \sum_{x, y \geq 1} f(x) [\Delta_S^+(x, y) + \Delta_S^-(x, y)] \mu_t^{\odot}(x) \mu_t^{\odot}(y) \\ &\quad - \sum_{x, y, z \geq 1} f(y) K_D^0(x, y, z) \mu_t^{\oplus}(x) \mu_t^{\odot}(y) \mu_t^{\odot}(z) \\ &\quad - \sum_{x, y, z \geq 1} f(y) [\Delta_D^+(x, y, z) + \Delta_D^-(x, y, z)] \mu_t^{\oplus}(x) \mu_t^{\odot}(y) \mu_t^{\odot}(z) \end{aligned} \quad (2.17a)$$

and

$$\begin{aligned} \frac{d}{dt}(g, \mu_t^{\oplus}) &= \frac{1}{2} \sum_{x, y \geq 1} g(x+y) \Delta_S^+(x, y) \mu_t^{\odot}(x) \mu_t^{\odot}(y) + \sum_{x, y \geq 1} g(x) \Delta_S^-(x, y) \mu_t^{\odot}(x) \mu_t^{\odot}(y) \\ &\quad + \sum_{x, y, z \geq 1} [g(x+y) - g(x)] (K_D^0 + \Delta_D^+)(x, y, z) \mu_t^{\oplus}(x) \mu_t^{\odot}(y) \mu_t^{\odot}(z) \\ &\quad + \sum_{x, y, z \geq 1} g(y) \Delta_D^-(x, y, z) \mu_t^{\oplus}(x) \mu_t^{\odot}(y) \mu_t^{\odot}(z) \\ &\quad + \frac{1}{2} \sum_{x, y \geq 1} [g(x+y) - g(x) - g(y)] K^+(x, y) \mu_t^{\oplus}(x) \mu_t^{\oplus}(y) \quad . \end{aligned} \quad (2.17b)$$

A similar equation to eq. (2.17b) holds for $\frac{d}{dt}(h, \mu_t^{\ominus})$. The reader will get a clear insight on the reason why these equations appear by seeing the generator of the

⁷See for instance the article [15] of J. Norris or [16] of I. Jeon.

⁸Another slight abuse notation: the definitions (eqs. (2.14) to (2.16)) of rates K_D^0 and Δ_D^{\pm} depend on the definitions of $\hat{T}(\pm, y)$ (eq. (2.13)) which in turn depends on the measure in question. This dependence is implicit in eqs. (2.17) and later (2.18).

discrete measure valued Markov chain $(\mu_t^{\oplus,N}, \mu_t^{\ominus,N}, \mu_t^{\ominus/\oplus,N})$ (this will be shown in the next subsection 2.3.4). Recall that

$$\mu_t^{+,N} = \mu_t^{\ominus,N} + \mu_t^{\oplus,N}, \quad \mu_t^{-,N} = \mu_t^{\ominus,N} + \mu_t^{\ominus/\oplus,N},$$

and so it is easily shown that for any bounded functions f and g

$$\frac{d}{dt}(f, \mu_t^+) = \frac{1}{2} \sum_{x,y \geq 1} [f(x+y) - f(x) - f(y)] K^+(x,y) \mu_t^+(x) \mu_t^+(y)$$

and

$$\frac{d}{dt}(g, \mu_t^-) = \frac{1}{2} \sum_{x,y \geq 1} [g(x+y) - g(x) - g(y)] K^-(x,y) \mu_t^-(x) \mu_t^-(y) \quad .$$

This is in accordance with the fact that $\mu_t^{\ominus,N} + \mu_t^{\oplus/\ominus,N}$ is a Marcus-Lushnikov process with rate $K^{+/-}$, therefore its limit is a solution to Smoluchowski equation with the corresponding rate.

2.3.4. Generator. This section gives a description of the generator of the Markov chain corresponding to the Double Coupling Algorithm. The stochastic jumps are described by the following elementary operations on measures corresponding to the jumps indicated in Figure 2.2. We adopt the notations μ for a generic element of Q_N^3 and x, y, y', z for integer masses.

$$\begin{aligned} J_{1a}^N(\mu, y, y') &= \mu + \frac{1}{N} \begin{pmatrix} 0 & , & \delta_{y+y'} - \delta_y - \delta_{y'} & , & 0 \\ \delta_{y+y'} & , & -\delta_y - \delta_{y'} & , & \delta_y + \delta_{y'} \\ \delta_y + \delta_{y'} & , & -\delta_y - \delta_{y'} & , & \delta_{y+y'} \end{pmatrix} \\ J_{1b}^N(\mu, y, y') &= \mu + \frac{1}{N} \begin{pmatrix} 0 & , & \delta_{y+y'} - \delta_y - \delta_{y'} & , & 0 \\ \delta_{y+y'} & , & -\delta_y - \delta_{y'} & , & \delta_y + \delta_{y'} \\ \delta_y + \delta_{y'} & , & -\delta_y - \delta_{y'} & , & \delta_{y+y'} \end{pmatrix} \\ J_{1c}^N(\mu, y, y') &= \mu + \frac{1}{N} \begin{pmatrix} 0 & , & \delta_{y+y'} - \delta_y - \delta_{y'} & , & 0 \\ \delta_{y+y'} & , & -\delta_y - \delta_{y'} & , & \delta_y + \delta_{y'} \\ \delta_y + \delta_{y'} & , & -\delta_y - \delta_{y'} & , & \delta_{y+y'} \end{pmatrix} \\ J_{2a}^N(\mu, x, y, z) &= \mu + \frac{1}{N} \begin{pmatrix} -\delta_x + \delta_{x+y} & , & -\delta_y & , & -\delta_z + \delta_{y+z} \\ -\delta_x + \delta_{x+y} & , & -\delta_y & , & \delta_y \\ \delta_y & , & -\delta_y & , & -\delta_z + \delta_{y+z} \end{pmatrix} \\ J_{2b}^N(\mu, x, y, z) &= \mu + \frac{1}{N} \begin{pmatrix} -\delta_x + \delta_{x+y} & , & -\delta_y & , & \delta_y \\ -\delta_x + \delta_{x+y} & , & -\delta_y & , & \delta_y \\ \delta_y & , & -\delta_y & , & -\delta_z + \delta_{y+z} \end{pmatrix} \\ J_{2c}^N(\mu, x, y, z) &= \mu + \frac{1}{N} \begin{pmatrix} -\delta_x + \delta_{x+y} & , & -\delta_y & , & -\delta_z + \delta_{y+z} \\ -\delta_x + \delta_{x+y} & , & -\delta_y & , & \delta_y \\ \delta_y & , & -\delta_y & , & -\delta_z + \delta_{y+z} \end{pmatrix} \\ J_{3a}^N(\mu, x, y) &= \mu + \frac{1}{N} \begin{pmatrix} 0 & , & 0 & , & \delta_{x+y} - \delta_x - \delta_y \\ \delta_{x+y} - \delta_x - \delta_y & , & 0 & , & 0 \end{pmatrix} \\ J_{3b}^N(\mu, x, y) &= \mu + \frac{1}{N} \begin{pmatrix} 0 & , & 0 & , & \delta_{x+y} - \delta_x - \delta_y \\ \delta_{x+y} - \delta_x - \delta_y & , & 0 & , & 0 \end{pmatrix} \end{aligned}$$

The introduction of the following notation clarifies the description of the generator of the Markov chain corresponding to the double coupling algorithm. For any $\gamma \in Q_N$, define the rescaled counting measure $\tilde{\gamma} \in Q_N$ on ordered pairs of masses of distinct particles as

$$\tilde{\gamma}(A \times A') := \gamma(A)\gamma(A') - \frac{1}{N}\gamma(A \cap A') \quad , \quad A, A' \subset \mathbb{N}.$$

Set also for any $\phi = (\phi_1, \phi_2, \phi_3)$ and $\mu = (\mu_1, \mu_2, \mu_3) \in Q_N^3$

$$\langle \phi, \mu \rangle := (\langle \phi_1, \mu_1 \rangle, \langle \phi_2, \mu_2 \rangle, \langle \phi_3, \mu_3 \rangle).$$

For any measure $\mu := (\mu^{\oplus}, \mu^{\ominus}, \mu^{\oplus}) \in Q_N^3$, with the corresponding *sets* of particles being $(X^{\oplus}, X^{\ominus}, X^{\ominus})$, we have

$$\langle (\phi_1, \phi_2, \phi_3), \mathcal{G}_t^N(\mu) \rangle := \left(\langle \phi_1, \mathcal{G}_t^{N,\oplus}(\mu) \rangle, \langle \phi_2, \mathcal{G}_t^{N,\ominus}(\mu) \rangle, \langle \phi_3, \mathcal{G}_t^{N,\ominus}(\mu) \rangle \right)$$

where

$$\begin{aligned}
& \left\langle \phi_1, \mathcal{G}_t^{N,\oplus}(\mu) \right\rangle \\
&= \frac{1}{2} \sum_{y,y' \geq 1} \phi_1(y+y') \Delta_S^+(y,y) \tilde{\mu}^\ominus(y,y') + \sum_{y,y' \geq 1} \phi_1(y) \Delta_S^-(y,y') \tilde{\mu}^\ominus(y,y') \\
&+ \sum_{x,y,z \geq 1} [\phi_1(x+y) - \phi_1(x)] (K_D^0 + \Delta_D^+)(x,y,z) \mu^\oplus(x) \mu^\ominus(y) \mu^\ominus(z) \\
&+ \sum_{x,y,z \geq 1} \phi_1(y) \Delta_D^-(x,y,z) \mu^\oplus(x) \mu^\ominus(y) \mu^\ominus(z) \\
&+ \frac{1}{2} \sum_{x,x' \geq 1} [\phi_1(x+x') - \phi_1(x) - \phi_1(x')] K^+(x,x') \tilde{\mu}^\oplus(x,x') \quad (2.18a)
\end{aligned}$$

and

$$\begin{aligned}
& \left\langle \phi_2, \mathcal{G}_t^{N,\ominus}(\mu) \right\rangle \\
&= \frac{1}{2} \sum_{y,y' \geq 1} [\phi_2(y+y') - \phi_2(y) - \phi_2(y')] K_S^0(y,y') \tilde{\mu}^\ominus(y,y') \\
&- \sum_{y,y' \geq 1} \phi_2(y) [\Delta_S^+(y,y') + \Delta_S^-(y,y')] \tilde{\mu}^\ominus(y,y') \\
&- \sum_{x,y,z \geq 1} \phi_2(y) K_D^0(x,y,z) \mu^\oplus(x) \mu^\ominus(y) \mu^\ominus(z) \\
&- \sum_{x,y,z \geq 1} \phi_2(y) [\Delta_D^+(x,y,z) + \Delta_D^-(x,y,z)] \mu^\oplus(x) \mu^\ominus(y) \mu^\ominus(z) \quad , \quad (2.18b)
\end{aligned}$$

with $\left\langle \phi_3, \mathcal{G}_t^{N,\ominus}(\mu) \right\rangle$ being defined analogously to eq. (2.18a).

3. Numerical Results. The results presented consider two kernels: the additive kernel $K(x,y) = \lambda(x+y)$ and a kernel that is used in modelling soot formation in a free molecular regime (thus we shall call it the ‘Soot Kernel’)⁹

$$K(x,y) = \left(\frac{1}{x} + \frac{1}{y} \right)^{\frac{1}{2}} \left(x^{\frac{1}{\lambda}} + y^{\frac{1}{\lambda}} \right)^2.$$

The reference value of λ for the additive kernel will be 1 and for the soot kernel 2.1. We shall always take as initial condition for the Marcus-Lushnikov process N particles with mass equal to 1. Throughout this section we shall denote by N the initial number of particles in each system (which is the same), by λ the above reference value of the parameter, whose perturbation will be denoted by ϵ (i. e. the X^\pm systems are governed by the parameter values $\lambda \pm \frac{1}{2}\epsilon$), and by L the number of simulations with the same initial conditions. The remaining notation is given below.

- (i) t = time of evolution of the particle system
- (ii) t_{run} = time taken to run the algorithms (CPU time).

⁹This kernel is studied extensively in [3] and used in [17, 18, 19].

(iii) The estimate of $\frac{\partial}{\partial \lambda}(f, \mu_t^\lambda)$ given by the l^{th} simulation is denoted by $F_l^{(\lambda)}$ ⁽¹⁰⁾, where f is a suitable test function.

(iv) The estimate of $\frac{\partial}{\partial \lambda}(f, \mu_t^\lambda)$ given by L simulations is denoted by $\bar{F}^{(\lambda)}$; it is equal to $\frac{1}{L} \sum_{l=1}^L F_l^{(\lambda)}$.

3.1. Some initial plots. Figure 3.1 shows what the derivative of the parametric solution of $\mu_t^\lambda(x)$ looks like for the Soot kernel for two different evolution times t . Figure 3.2 shows similar quantities, but for the Additive kernel; it is in good agreement with the analytic solution given in [2].

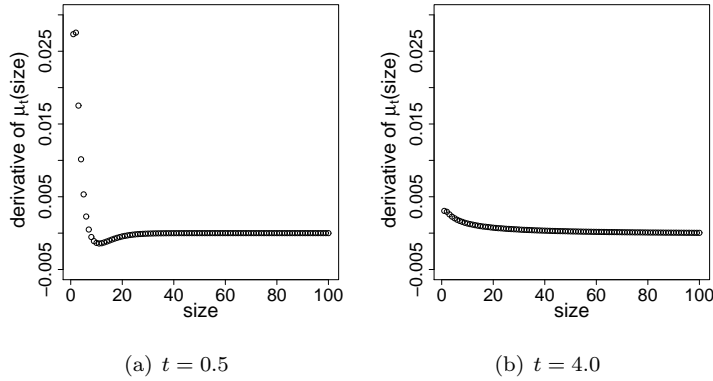


FIG. 3.1. Derivative $\frac{\partial}{\partial \lambda} \mu_t^\lambda(x)$ versus particle size x , for Soot kernel using the Double Coupling algorithm, $\lambda = 2.1$, $\epsilon = 0.03$, $N = 10^7$, $L = 300$. Confidence intervals have been omitted since they are visually negligible.

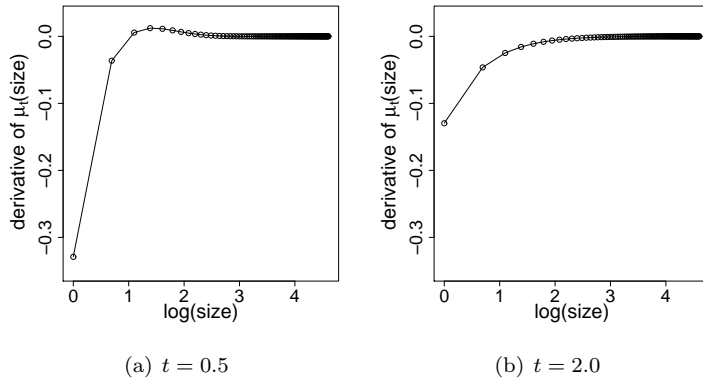


FIG. 3.2. Derivative $\frac{\partial}{\partial \lambda} \mu_t^\lambda(x)$ versus particle size x , for additive kernel using the Double Coupling algorithm, $\lambda = 1$, $\epsilon = 0.06$, $N = 10^6$, $L = 300$. The line joins the points of the derivative of the analytic solution of eq. (1.1) with monodisperse initial conditions, as given in [2]. Confidence intervals have been omitted since they are visually negligible.

3.2. Convergence study. There are two sources of systematic error in using the central difference estimator — one due to using a non-zero value of ϵ and the

¹⁰Note that taking $f(y) = \mathbf{1}_{y=x}$ gives $\frac{\partial}{\partial \lambda} \mu_t^\lambda(x)$.

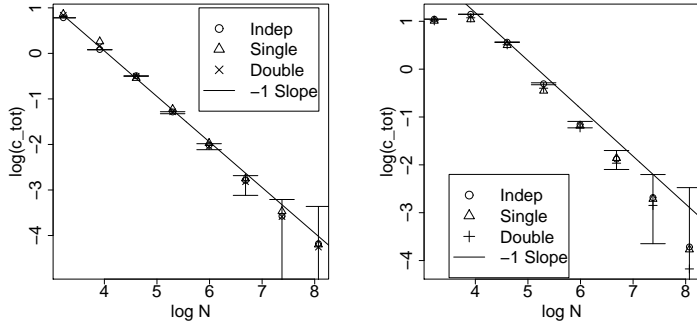
(a) Additive kernel, $\lambda = 1$, $\epsilon = 0.06$ (b) Soot kernel, $\lambda = 2.1$, $\epsilon = 0.03$

FIG. 3.3. $\log c_{tot}$ versus $\log N$, $N = 25 \times 2^i$ for $i = 0, \dots, 7$, $N \times L = 10^8$ and $0 \leq t \leq 3$. Confidence intervals given are for the Independent case only.

other due to assuming a finite particle system. It is the latter we investigate — here we estimate the order of convergence of the systematic error as N varies. The value of ϵ will be fixed here.

We define the **systematic error due to N** as the difference between the expected central difference (for finite particle number N) and the analytic central difference.

$$e_{\text{sys}}(N; \lambda, \epsilon, t) = \mathbb{E}\bar{F}^{(\lambda, \epsilon)}(N; t) - f^{(\lambda, \epsilon)}(t) \quad (3.1)$$

However, as we nearly always do not know the analytic central difference $f^{(\lambda, \epsilon)}$, we estimate it using

$$\bar{F}^{(\lambda, \epsilon)}(N_{\text{large}}; t) \quad (3.2)$$

for very large N_{large} . Also, $\mathbb{E}\bar{F}^{(\lambda, \epsilon)}(N; t)$ is estimated by $\bar{F}^{(\lambda, \epsilon)}(N; t)$. Now we set test function $f(y) = \mathbb{1}_{\{y=i\}}$ again, to ensure that $\bar{F}^{(\lambda, \epsilon)}(N; t)$ is an estimate of the number density for particle size i . Therefore we rename $\bar{F}^{(\lambda, \epsilon)}(N; t)$ as $\bar{F}^{(\lambda, \epsilon, i)}(N; t)$ for number density estimate at particle size $i \in \mathbb{N}$; we adopt analogous notation $f^{(\lambda, \epsilon, i)}$ and $e_{\text{sys}}(N; \lambda, \epsilon, t, i)$ for this particular choice of test function. Our metric for considering convergence in N is simply the absolute estimated systematic error, summed over chosen evolution times $(t_k)_{k=1}^T$ ¹¹ and summed over particle sizes i :

$$c_{\text{tot}} = \sum_{k=1}^T \sum_{i \in \mathbb{N}} |e_{\text{sys}}(N; \lambda, \epsilon, t_k, i)| \quad (3.3)$$

Figure 3.3 shows what we expect — the $c_{\text{tot}} \sim \frac{1}{N}$. We obtain similar plots for different values of ϵ .

3.3. Statistical error. This quantity is defined as

$$e_{\text{stat}}(N; t, \lambda, \epsilon) = \bar{F}^{(\lambda, \epsilon)}(N; t) - \mathbb{E}\bar{F}^{(\lambda, \epsilon)}(N; t). \quad (3.4)$$

¹¹We take the t_k to be $(0.5, 1.0, \dots, 7.0)$

This is a signed measure which associates to each $i \geq 1$ the number $e_{\text{stat}}(N; t, \lambda, \epsilon, i)$. We consider in this paragraph how it behaves according to the different algorithms and kernels. The variance of each estimator F_i can be estimated for each $i \geq 1$ by

$$v_F^{(i)} := \frac{1}{L-1} \sum_{l=1}^L (F_l(i) - \bar{F}(i))^2. \quad (3.5)$$

This implies that the asymptotic $100(1 - \alpha)\%$ confidence interval for $\mathbb{E}\bar{F}^{(\lambda, \epsilon)}(N; t; i)$ is:

$$\left[\bar{F}^{(\lambda, \epsilon)}(N; t; i) - z_{\alpha/2} \sqrt{\frac{v_F^{(i)}}{L}}, \bar{F}^{(\lambda, \epsilon)}(N; t; i) + z_{\alpha/2} \sqrt{\frac{v_F^{(i)}}{L}} \right] \quad (3.6)$$

where $z_{\alpha/2}$ is the upper $\alpha/2$ point of the standard normal distribution. Hence,

$$\mathbb{P}\left(|e_{\text{stat}}(N; t, \lambda, \epsilon)| \leq z_{\alpha/2} \sqrt{\frac{v_F^{(i)}}{L}}\right) \approx 1 - \alpha. \quad (3.7)$$

For this paper, we set $\alpha = 0.05$ i.e. we consider 0.95% confidence intervals. Also, consider the sum $e_{\text{totalstat}}$ of the single-simulation variances over the particle sizes:

$$e_{\text{totalstat}} := \sum_{i \geq 1} v_F^{(i)} \quad (3.8)$$

We wish to see how this quantity behaves with N . Figure 3.4 demonstrates that for all three algorithms, the total variance $e_{\text{totalstat}}$ behaves as $\frac{1}{N}$ since the slopes of the fitted lines are approximately -1 . More importantly, the intercept for the Double algorithm is lower than the Single case, and much lower than that of the Independent case indicating that $e_{\text{totalstat}}$ is much smaller for the Double and Single cases than for the Independent— $e_{\text{totalstat}}$ for the Independent case at one point is approximately 20 times larger than that for the Double case at $t = 1.0$. One interesting observation is that the difference in the intercepts (of the fitted lines) between the Double and Indep decreases, showing that the benefits of smaller statistical error in the Double case become less pronounced as t increases. The same happens for the Single algorithm, but at a faster rate, showing that the X_t^- and X_t^+ systems diverge from each other, but faster for the Single algorithm than for the Double algorithm.

3.4. Efficiency. In this subsection, we will discuss which algorithm is ‘best’. A quantification of this quality needs to be defined which will take into account the accuracy and the run time of each algorithm. One such measure can be described as the run time needed to achieve a certain fixed statistical error (assuming N and ϵ are fixed). Before defining it, we shall

- (i) Let $t_{\text{run}}(t)$ be the CPU time actually taken to perform L_{run} simulations.
- (ii) Set $e_{\text{fixed}} = \frac{e_{\text{totalstat}}}{L}$; this “total standard error” will be artificially **fixed**.
- (iii) Let $L_{\text{est}}(t)$ be the estimated number of simulations required to acquire the fixed $e_{\text{fixed}}(t)$ value and $t_{\text{est}}(t)$ be the CPU time required to perform $L_{\text{est}}(t)$ runs.

The condition $e_{\text{fixed}}(t) = \sum_{i \geq 1} \frac{v_F^{(i)}}{L_{\text{est}}(t)}$ implies that $L_{\text{est}}(t) = \frac{\sum_{i \geq 1} v_F^{(i)}}{e_{\text{fixed}}}$; we then have that $t_{\text{est}}(t) = \frac{t_{\text{run}}(t)}{L_{\text{run}}(t)} L_{\text{est}}(t)$. We shall use the following quantity to compare different

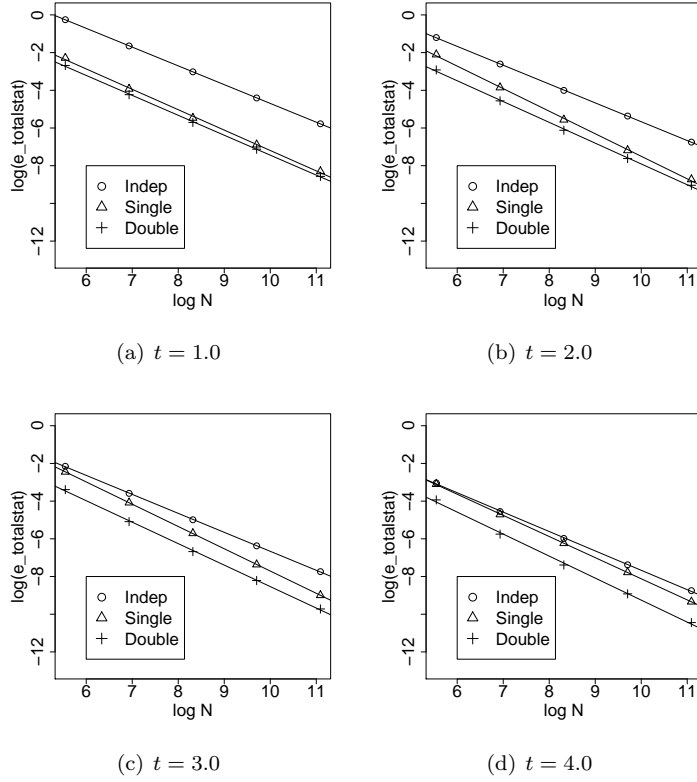


FIG. 3.4. $\log e_{\text{totalstat}}$ versus $\log N$ over different values of t for all the algorithms for the Additive kernel, where $N = 2^{10}, 2^{12}, 2^{14}, 2^{16}, 2^{18}, N \times L = 2^{26}$ and $\epsilon = 0.06$.

algorithms.

$$\text{Inefficiency}^{\text{algorithm}} := \frac{t_{\text{est}}^{\text{algorithm}}(t)}{t_{\text{est}}^{\text{Double}}(t)}; \quad (3.9)$$

we call it the **Inefficiency with respect to the Double Coupling algorithm**. Hence, if an algorithm has an Inefficiency of more than unity, the algorithm does not perform as well as the Double Coupling algorithm. Figure 3.5 plots these inefficiencies. One can see that the Independent algorithm has large inefficiencies for small t —this is due to the vastly smaller statistical errors of the Double and Single Coupling algorithms, as well as all three central difference algorithms taking comparable times to run. In fact, the Double and Single algorithms are generally quicker for smaller ϵ since the Independent algorithm requires two simulations to generate a derivative estimate, as well as the fact that the Single and Double algorithms have fewer \ominus and \oplus particles to deal with. The inefficiencies of the Single algorithm lie between 1.0 and 2.0 (note that Figure 3.5 uses log scales) indicating that the Double algorithm has a significant improvement over the Single in terms of accuracy. Also, the inefficiencies decrease with ϵ since larger ϵ implies that \oplus and \ominus particles are created, thus meaning that both coupling algorithms have larger CPU run times (the run times increase almost linearly with ϵ for the Single and Double, whereas they are almost constant with respect to ϵ for the Independent algorithm). Also, the ratio of variances for

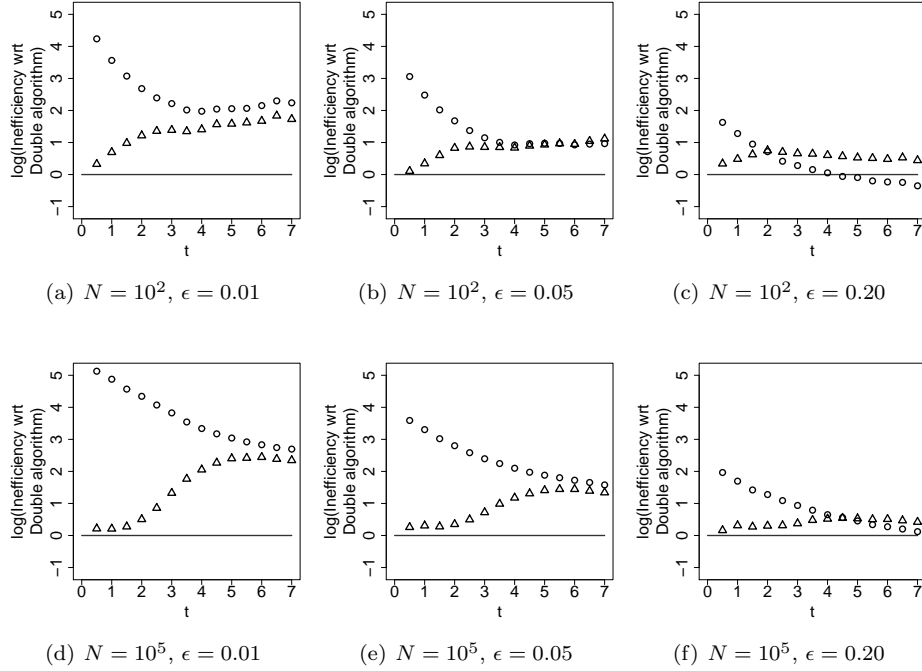


FIG. 3.5. Additive kernel — Inefficiency relative to Double for Indep and Single algorithms, as a function of t for different values of ϵ and N . The Independent algorithm is represented by circles, Single by triangles and the Double threshold by the horizontal line.

the Independent and Single algorithms relative to the Double algorithm decreases for larger ϵ , due to the increasing similarity between the Independent algorithm and the coupling algorithms, thus decreasing the inefficiencies. Note however that the case $\epsilon = 0.2$ is unlikely to be computationally useful as it amounts to a 20% change in the parameter, which enters multiplicatively into the kernel.

Furthermore, one notices that the larger value of $N = 10^5$ results in larger inefficiencies for the Independent and Single algorithms for the Additive kernel. This appears to be because as N increases, the capacity for cancellations is larger, implying that the Double is more accurate (and faster) than one expects. Also, as t increases, we find that the number of particles for all algorithms decreases dramatically, and so there appears to be little difference in accuracy between all three algorithms for larger t . This is to be expected since μ_t^+ and μ_t^- will become increasing dissimilar, thus indicating smaller covariances, and so the variances for the Double and Single Coupling algorithms will be similar to those for the Independent algorithm, this loss of efficiency of the coupling algorithms is ultimately unavoidable in this class of algorithms, and one can only hope to minimise this decrease.

It is important to realise that this analysis does not take into account the systematic error due to ϵ or N since the Inefficiency metric only uses estimated variances. A related problem with the analysis is that the number of particles for $t \in [3.5, 7.0]$ becomes quite small¹², and therefore the systematic errors and estimated variances are not very reliable.

¹²about $O(1) - O(10)$ if $N = 10^2$ but $O(100) - O(1000)$ for $N = 10^6$.

4. Conclusions. In this paper, two new stochastic algorithms were described which solve for parametric derivatives of the solution to the discrete Smoluchowski's coagulation equation. These algorithms consider two Marcus-Lushnikov processes which are coupled together in order to reduce the difference in their trajectories. The hope was that this would significantly reduce the variance of the central difference estimators of the parametric derivatives. In the numerical results section, we first validated the fact that the order of convergence for these algorithms is indeed $O(1/N)$. Furthermore, it was shown from the statistical error plots that the accuracy is order of magnitudes better than that of the worst case (the Independent algorithm), at least for larger N and smaller ϵ . Subsequently, we considered a method of comparing the algorithms which considers both the variances of the derivative estimators as well as the CPU run times. It was shown that the Double algorithm is mostly more 'efficient' than Single over variations in ϵ and t , whilst being significantly more 'efficient' than the Independent algorithm for small t , large N and small ϵ , though some of this advantage is lost for larger t and ϵ .

It is worth noting that these approaches to parametric derivative estimation suffer from the fact that the parameter must be scalar—if a vector parameter is to be considered, then each component must be calculated separately. Furthermore, the derivatives can only be computed at fixed values of the parameter. We leave these challenges for later work.

REFERENCES

- [1] A. EIBECK AND W. WAGNER, *Stochastic Particle Approximations for Smoluchowski's Coagulation Equation*, Ann. Appl. Probab., 11 (2001), pp. 1137–1165.
- [2] A. EIBECK AND W. WAGNER, *An Efficient Stochastic Algorithm for Studying Coagulation Dynamics and Gelation Phenomena*, SIAM J. Sci. Comput., 22 (2000), pp. 802–821.
- [3] M. GOODSON AND M. KRAFT, *An Efficient Stochastic Algorithm for Simulating Nano-particle Dynamics*, J. Comput. Phys., 183 (2002), pp. 210–232.
- [4] A. KOLODKO AND K. SABELFELD, *Stochastic Particle Methods for Smoluchowski Coagulation Equation: Variance Reduction and Error Estimations*, Monte Carlo Methods Appl., 9 (2003), pp. 315–339.
- [5] H. BRIESEN, *Hierarchical characterization of aggregates for Monte Carlo simulations*, AIChE J., 52 (2006), pp. 2436–2446.
- [6] H. ZHAO, A. MAISELS, T. MATSOUKAS AND C. ZHENG, *Analysis of four Monte Carlo methods for the solution of population balances in dispersed systems*, Powder Technol., 173 (2007), pp. 38–50.
- [7] A. BRAUMANN, M. J. GOODSON, M. KRAFT AND P. R. MORT, *Modelling and validation of granulation with heterogeneous binder dispersion and chemical reaction*, Chem. Eng. Sci., 62 (2007), pp. 4717–4728.
- [8] R. I. A. PATTERSON AND M. KRAFT, *Models for the aggregate structure of soot particles*, Combust. Flame, 151 (2007), pp. 160–172.
- [9] H. B. ZHAO AND C. G. ZHENG, *A new event-driven constant-volume method for solution of the time evolution of particle size distribution*, J. Comput. Phys., 228 (2009), pp. 1412–1428.
- [10] A. VIKHANSKY AND M. KRAFT, *A Monte Carlo methods for identification and sensitivity analysis of coagulation processes*, J. Comput. Phys., 200 (2004), pp. 50–59.
- [11] A. VIKHANSKY AND M. KRAFT, *Two methods of sensitivity analysis of coagulation processes in population balances by a Monte Carlo method*, Chem. Eng. Sci., 61 (2006), pp. 4966–4972.
- [12] A. VIKHANSKY, M. KRAFT, M. SIMON, S. SCHMIDT AND H. J. BART, *Droplets population balance in a rotating disc contactor: an inverse problem approach*, AIChE J., 52 (2006), pp. 1441–1450.
- [13] S. ASMUSSEN AND P. W. GLYNN, *Stochastic Simulation: Algorithms and Analysis*, Springer, 2007, pp. 224–230.
- [14] R. I. A. PATTERSON, J. SINGH, M. BALTHASAR, M. KRAFT AND J. R. NORRIS, *The Linear Process Deferment Algorithm: A new technique for solving population balance equations*, SIAM J. Sci. Comput., 28 (2006), pp. 303–320.

- [15] J. R. NORRIS, *Smoluchowski's Coagulation Equation: Uniqueness, Nonuniqueness and a Hydrodynamic Limit for the Stochastic Coalescent*, Ann. Appl. Probab., 9 (1999), pp. 78–109.
- [16] I. JEON, *Existence of gelling solutions for coagulation-fragmentation equations*, Comm. Math. Phys., 194 (1998), pp. 541–567.
- [17] R. I. A. PATTERSON, J. SINGH, M. BALTHASAR, M. KRAFT AND W. WAGNER, *Extending stochastic soot simulation to higher pressures*, Combust. Flame, 145 (2006), pp. 638–642.
- [18] N. MORGAN, M. KRAFT, M. BALTHASAR, D. WONG, M. FRENKLACH AND P. MITCHELL, *Numerical simulations of soot aggregation in premixed laminar flames*, Proc. Combust. Inst., 31 (2007), pp. 693–700.
- [19] M. CELNIK, R. I. A. PATTERSON, M. KRAFT AND W. WAGNER, *Coupling a stochastic soot population balance to gas-phase chemistry using operator splitting*, Combust. Flame, 148 (2007), pp. 158–176.
- [20] D. J. ALDOUS, *Deterministic and Stochastic Models for Coalescence (Aggregation and Coagulation) : a Review of the Mean-Field Theory for Probabilists*, Bernoulli, 5 (1999), pp. 3–48.
- [21] P. L. W. MAN, M. KRAFT AND J. R. NORRIS, *Coupling Algorithms for Calculating Sensitivities of Population Balances*, Numerical Analysis and Applied Mathematics, AIP Conf. Proc., 1048 (2008), pp. 927–930.