Cambridge Centre for Computational Chemical Engineering

University of Cambridge

Department of Chemical Engineering

Preprint ISSN 1473 – 4273

Coupling algorithms for calculating sensitivities of Smoluchowski's Coagulation equation

Peter L. W. Man [1] James R. Norris [2] Ismaël F. Bailleul [2] Markus Kraft [1]

released: 23 April 2009

[1] Department of Chemical Engineering and Biotechnology
University of Cambridge
New Museums Site
Pembroke Street
Cambridge, CB2 3RA
UK
E-mail: mk306@cam.ac.uk

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

Preprint No. 70

c4e

Key words and phrases: modelling, simulation, coupling, sensitivity, Smoluchowski Coagulation Equation

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

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 have 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.

Contents

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. Given a real-valued function f we shall write (f, μ_t) for $\sum_x f(x)\mu_t^\lambda(x)$. Quantities measured by the experimenter (such as moments) are of this form. Smoluchowski's description of the evolution of μ_t^λ is

$$\frac{\mathrm{d}}{\mathrm{d}t}(f, \mu_t^\lambda) = \frac{1}{2} \sum_{x,y \geqslant 1} \big\{ f(x+y) - f(x) - f(y) \big\} \, K_\lambda(x,y) \, \mu_t(x) \, \mu_t(y). \tag{1.1}$$

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 $\big\{f(x+y) - f(x) - f(y)\big\}$ is the change which has occurred in the quantity (f, μ_t) 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.

There is a large amount of literature concerning the solving of the continuous particle sized version of equation eq. (1.1) and its many variations such as particle inception, surface growth, sintering, and fragmentation [3, 4, 6–8, 10, 13, 19, 20]. 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 [2, 16–18]. 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 [16, 18] 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 [17]. 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 [1, 9, 12], 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 and $(f, \mu_t) = \sum_x f(x)\mu_t(x)$ as the integral of f against μ_t. In Marcus-Lushnikov's approach, μ_t is approximated by a random finite measure of the form[1]

$$\mu_t^N = \frac{1}{N} \sum_n \delta_{x_n(t)}$$

whose dynamics are that of a Markov chain. We shall talk of $x_n(t)$ of $\delta_{x_n(t)}$ as a particle of the system at time t. Start from $\mu_0^N = \frac{1}{N}\sum \delta_{x_n}$; 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

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

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

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

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 $(\mu_t^{\lambda+\frac{1}{2}\varepsilon\,;\,N} - \mu_t^{\lambda-\frac{1}{2}\varepsilon\,;\,N})/\varepsilon$. No *a priori* independence or dependence between $\mu_t^{\lambda+\frac{1}{2}\varepsilon\,;\,N}$ and $\mu_t^{\lambda-\frac{1}{2}\varepsilon\,;\,N}$ 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

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

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.

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 $I := \int_0^1\int_0^1 |x - y|\, f(x, y)\, \mathrm{d}x\, \mathrm{d}y$, subject to the condition that the two marginals of the probability $f(x, y)\, \mathrm{d}x\, \mathrm{d}y$ on the x and y axis are uniform[2]. 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 $\mathrm{d}x\, \mathrm{d}y$ is such a coupling, but it does not

[1] δ_x is a Dirac mass at x.

[2] That is $\int_0^1 f(x, y)\mathrm{d}y = 1$ for any x, and $\int_0^1 f(x, y)\, \mathrm{d}x = 1$ for any y.

minimise I. This minimum is attained for the singular probability on the square with support on the diagonal, and uniform on it.

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 $\left(\{\mu_t^{\lambda\,;N}\}_{t\in[0,T_0]}\right)$ with values in the set of finite measures of the form $\frac{1}{N}\sum \delta_{x_n}$; but the basic idea is the same.

Denote by X_t^- and X_t^+ the set of particles of $\mu_t^{\lambda-\frac{1}{2}\varepsilon\,;N}$ and $\mu_t^{\lambda+\frac{1}{2}\varepsilon\,;N}$, respectively. The more particles X_t^- and X_t^+ have in common the closer the particle systems are; the set

$$X_t^{\odot} = X_t^- \cap X_t^+$$

is made up of those particles in common. The first version of our algorithm, called **Single Coupling Algorithm**, tries to keep the number of particles from $X^{\odot}$ as large as possible, imposing that (as much as possible) when two particles, which are present in both X_t^- and X_t^+, are chosen to coagulate in one system, they also coagulate in the other. This helps to keep $X_t^{\odot}$ large. Of course, as the coagulation rates in X^- and X^+ are different 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 $X^{\pm}\backslash X^{\odot}$ as rare as possible. In addition to the above coupling, it considers what happens when a particle of $X^{\odot}$ coagulates with a particle of $X^{\pm}\backslash X^{\odot}$. If, when a particle of $X_t^{\odot} \cap X_t^{\pm}$ coagulates with a particle of $X^{\pm}\backslash X^{\odot}$, the same particle of $X^{\odot}$ can be used in a coagulation event in $X^{\mp}$ with a particle of $X^{\mp}\backslash X^{\odot}$, then, out of the four particles involved in the process we create only two particles (possibly in $X^{\pm}\backslash X^{\odot}$); three particles are created if this coupling cannot hold. Details of the single and double couplings are given below.

Note, however, that one ultimately expects that given enough time, the two systems will behave almost independently, but the hope is that the divergence in their trajectories is slow enough over the time span of interest. The simulation of the sensitivity with the help of two independent Marcus-Lushnikov processes will be referred to as the **Independent Algorithm**; it will be used for comparison with the other algorithms.

Labelling As should be clear from the preceding paragraphs, the couplings require us to keep track of the evolution of X_t^- and X_t^+ simultaneously. We encode all the necessary information using one ensemble of *computational* particles, by taking note what is common between the two ensembles, and what is different. These particles are labelled particles whose labels are defined in Table 1. The set of all the events that can happen to the whole system $\{X^+, X^-\}$ is represented in Table 2.

Notation. To be consistent with the above ± notations, we shall write K^- for the kernel $K_{\lambda-\frac{1}{2}\varepsilon}$ and K^+ for the kernel $K_{\lambda+\frac{1}{2}\varepsilon}$.

Table 1: *Particle labels and their meaning*

Label	Meaning
$\oplus_x$	a **single** real particle (of size x) present **only** in X^+.
$\ominus_x$	a **single** real particle (of size x) present **only** in X^-.
$\odot_x$	a **pair of identical** real particles (both of size x), one from X^- and the other from X^+.

Table 2: *Possible events described using the labelling notation*

Type	Event	Explanation

(a)
$1\ (b)$
(c)
	$\odot_x + \odot_y \to \begin{cases} \odot_{x+y} \\ \ominus_x + \ominus_y + \oplus_{x+y} \\ \oplus_x + \oplus_y + \ominus_{x+y} \end{cases}$	if occurs in **both** X^- and X^+
		if occurs **only in** X^+
		if occurs **only in** X^-

2(a)	$\ominus_x + \odot_y \to \ominus_{x+y} + \oplus_y$	This represents a coagulation between a pair of particles, one of which is only present in X^-, so the $\ominus$ particle must increase in size and the $\odot$ particle becomes a $\oplus$ (since this particle is no longer in the X^- system).
2(b)	$\oplus_x + \odot_y \to \oplus_{x+y} + \ominus_y$	Same logic — the reaction can only happen in the X^+ system.
2(c)	$\oplus_x + \odot_y + \ominus_z \to \oplus_{x+y} + \ominus_{y+z}$	See Double coupling algorithm explanation.
3	$\ominus_x + \ominus_y \to \ominus_{x+y}$	Particles present *only* in X^- coagulate.
4	$\oplus_x + \oplus_y \to \oplus_{x+y}$	As in event type 3 except in X^+.

reject	$\oplus_x + \ominus_y$	This coagulation cannot occur since each of the particles cannot see the other.

2.2 Single coupling system

2.2.1 The idea

The coupling procedure is implemented using a **majorant kernel**. This is a symmetric non-negative function $\widehat{K}(\cdot,\cdot)$ which is bigger than K^- and K^+. Run both systems X^- and X^+ at the *same rate*, given by $\widehat{K}$; a coagulation happening at that rate is called *potential*. If a potential coagulation of particles x_i and x_j happens in X^- or X^+, perform it with probability

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

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. 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.

Denote by (x_i, x_j) the pair of $\odot$-particles (possibly) involved in the potential coagulation event. Set

- $K_S^0 := \min\{K^+(x_i, x_j), K^-(x_i, x_j)\}$ — rate at which a coagulation of the type $\odot + \odot \to \odot$ occurs,
- $\Delta_S^+ := \max\{K^+(x_i, x_j) - K^-(x_i, x_j), 0\}$ — rate at which a coagulation of the type $\odot + \odot \to \ominus + \ominus + \oplus$ occurs,
- $\Delta_S^- := \max\{K^-(x_i, x_j) - K^+(x_i, x_j), 0\}$ — rate at which a coagulation of the type $\odot + \odot \to \oplus + \oplus + \ominus$ occurs.

Figure 1 gives a schematic picture of the procedure.

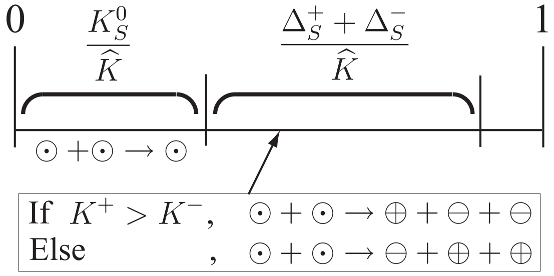

0
$\frac{K_S^0}{\widehat{K}}$
$\frac{\Delta_S^+ + \Delta_S^-}{\widehat{K}}$
1
$\odot + \odot \to \odot$
If $K^+ > K^-$, $\odot + \odot \to \oplus + \ominus + \ominus$
Else , $\odot + \odot \to \ominus + \oplus + \oplus$

Figure 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 whole system for the Single Coupling dynamics; they were named $1, 2a, 2b, 3$, and 4. Each of the following rates $\widehat{\rho}_k$ represents the majorant rate at which a coagulation of type k happens[3].

$$\begin{array}{lll} \widehat{\rho_1} := \frac{1}{N}\sum_{(\odot,\odot)} \widehat{K}(\odot, \odot) & , & \widehat{\rho_{2a}} := \frac{1}{N}\sum_{(\odot,\ominus)} \widehat{K}(\odot, \ominus), \\ \widehat{\rho_{2b}} := \frac{1}{N}\sum_{(\odot,\oplus)} \widehat{K}(\odot, \oplus) & , & \widehat{\rho_3} := \frac{1}{N}\sum_{(\ominus,\ominus)} \widehat{K}(\ominus, \ominus), \\ \widehat{\rho_4} := \frac{1}{N}\sum_{(\oplus,\oplus)} \widehat{K}(\oplus, \oplus). & & \end{array} \tag{2.1}$$

The total majorant rate of coagulation is defined as

$$\widehat{\rho} := \widehat{\rho_1} + \widehat{\rho_{2a}} + \widehat{\rho_{2b}} + \widehat{\rho_3} + \widehat{\rho_4}. \tag{2.2}$$

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

[3] $\sum_{(\odot,\odot)} \widehat{K}(\odot, \odot)$ is for example the sum of the $\widehat{K}(x_i, x_j)$ for all the pairs (x_i, x_j) belonging to $X^{\odot}$. Similar meaning is given to the other $\widehat{\rho}_\alpha$. Recall a $\oplus$-particle belongs to $X^+ \backslash X^{\odot}$, and likewise for $\ominus$.

Algorithm 1: Single Coupling algorithm

```
1   Set $t = 0$. Set all $N$ initial particles to have $\odot$ label.
    while $t < t_{end}$ do
2       Generate a realisation of the holding time $\Delta t \sim \mathrm{Exp}(\widehat{\rho})$ where $\widehat{\rho}$ is specified
        in eq. 2.1 and eq. 2.2, and set $t \leftarrow t + \Delta t$ .
3       Generate an unordered pair of particles $(i, j)$ according to the index
        distribution
            $$\frac{\widehat{K}(x_i, x_j)}{2N\widehat{\rho}} = \frac{\widehat{\rho}_k}{\widehat{\rho}} \frac{\widehat{K}(x_i, x_j)}{2N\widehat{\rho}_k}. \qquad (2.3)$$
        This chooses the process $k \in \{1, 2a, 2b, 3, 4\}$ and the particle pair that is of
        the correct type for the process in one go.
4       Define the following probabilities:
            $$p_{\oplus} = \frac{K^{+}(x_i, x_j)}{\widehat{K}(x_i, x_j)}, \;\; p_{\ominus} = \frac{K^{-}(x_i, x_j)}{\widehat{K}(x_i, x_j)}. \qquad (2.4)$$
        switch the value of $k$ chosen do
5           case $k = 1$
                This is the Single Coupling part. Do exactly as shown in Figure 1, i. e.
6               Generate random variable $\mathbf{U} \sim U(0, 1)$.
                if $0 < \widehat{K}\boldsymbol{U} \leqslant K_S^0$ then
7                   perform $\odot + \odot \to \odot$.
                else if $K_S^0 < \widehat{K}\boldsymbol{U} \leqslant K_S^0 + \Delta_S^+ + \Delta_S^-$ then
                    if $K^+ > K^-$ then
8                       perform $\odot_{x_i} + \odot_{x_j} \to \ominus_{x_i} + \ominus_{x_j} + \oplus_{x_i + x_j}$.
                    else
9                       perform $\odot_{x_i} + \odot_{x_j} \to \oplus_{x_i} + \oplus_{x_j} + \ominus_{x_i + x_j}$.
                else
10                  Go to Step 16.
            For cases 2a, 2b, 3 and 4, assume the labels on the particles in the
            following description are correct, otherwise swap the indices i and j.
11          case $k = 2a$ w. p. $p_{\ominus}$, perform $\ominus_{x_i} + \odot_{x_j} \to \ominus_{x_i + x_j} + \oplus_{x_j}$ (w. p. means
            'with probability').
12          case $k = 2b$ w. p. $p_{\oplus}$, perform $\oplus_{x_i} + \odot_{x_j} \to \oplus_{x_i + x_j} + \ominus_{x_j}$.
13          case $k = 3$ w. p. $p_{\ominus}$, perform $\ominus_{x_i} + \ominus_{x_j} \to \ominus_{x_i + x_j}$.
14          case $k = 4$ w. p. $p_{\oplus}$, perform $\oplus_{x_i} + \oplus_{x_j} \to \oplus_{x_i + x_j}$.
            otherwise
15              Go to Step 16.
16      For each particle that has just been involved in a coagulation, or newly
        formed, search for a particle of the same size in the other system. If there is
        such a particle, perform a cancellation operation: $\ominus_x + \oplus_x \to \odot_x$.
17      if there is only one particle left in the system then STOP.
```

2.2.3 Implementation and Complexity

The main implementation issue considers the form of the majorant kernel $\widehat{K}$. We assume that $\widehat{K}$ can be expressed as:

$$\widehat{K}(x_i, x_j) =: \sum_\alpha \widehat{K}_\alpha(x_i, x_j) =: \sum_\alpha f_\alpha(x_i) g_\alpha(x_j); \tag{2.5}$$

such a form of $\widehat{K}$ enables an easy computation of the following quantity used in the algorithm.

$$\frac{\widehat{K}(x_i, x_j)}{2N\widehat{\rho}} = \sum_\alpha \frac{\sum_{a \neq b} f_\alpha(x_a) g_\alpha(x_b)}{\sum_{a \neq b} \sum_{\alpha'} f_{\alpha'}(x_a) g_{\alpha'}(x_b)} \frac{f_\alpha(x_i)}{\sum_a f_\alpha(x_a)} \frac{g_\alpha(x_j)}{\sum_{b\,;\,b \neq a} g_\alpha(x_b)} \tag{2.6}$$

Details on the methods used here can be found in the articles of Eibeck and Wagner [6, 7] and Kraft and coworkers [8, 13, 14]. The advantage of these methods is two-fold: first we can store the values $f_p(x_i)$ and $g_p(x_i)$ in 'binary' tree structures which also holds their sums (over i). This allows efficient generation from the distributions

$$\frac{f(x_i)}{\sum_i f(x_i)} \qquad \frac{g(x_i)}{\sum_i g(x_i)}$$

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 generation of a 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 16 of Algorithm 1, a search of particles for cancellation 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, the cancellation 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 $\oplus$ and $\ominus$ particle numbers increase 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 minimize the rate at which particles of type $\ominus$ or $\oplus$ are created; it was described in section 2.1. Figure 2 presents a pictorial illustration of this coupling.

More formally, we:

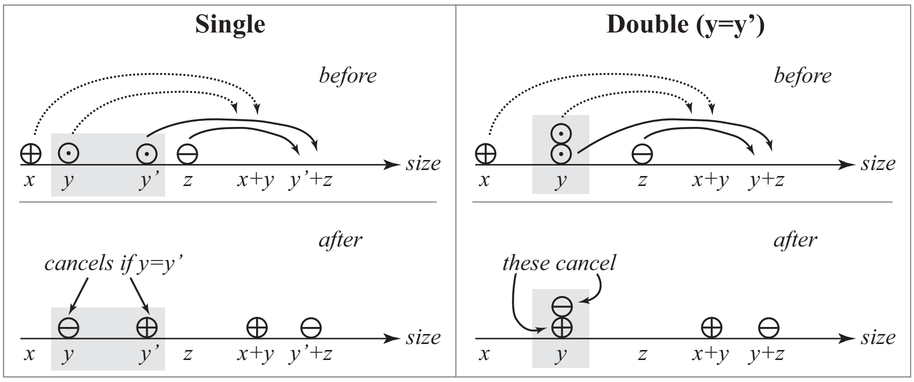


Figure 2: *Pictorial explanation of the Double coupling algorithm.*

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

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

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

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

- rejection steps are performed to correct the rates.

2.3.1 The algorithm

The algorithm is the same as the Single Coupling version, except that we merge processes 2a and 2b into a new process 2 whose majorant rate is $\widehat{\rho}_2 := \widehat{\rho}_{2a} + \widehat{\rho}_{2b}$. In describing the algorithm, and given a particular $\odot$ particle, we write $\widehat{T}(+,\odot)$ for $\sum_{\oplus}\widehat{K}(\oplus,\odot)$, and $\widehat{T}(-,\odot)$ for $\sum_{\ominus}\widehat{K}(\ominus,\odot)$; the maximum of these two quantities is denoted by $\widehat{T}(\vee,\odot)$. 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.

Double Coupling algorithm rates Denote by $\mu_t^{\odot,N} = \frac{1}{N}\sum_x \delta_x$ the empirical measure of the distribution of the $\odot$ particles at time t: an x appears in the sum only if there is $\odot$ particle of mass x at time t. Give similar definitions for $\mu_t^{\ominus,N}$ and $\mu_t^{\oplus,N}$. Given a $\odot$ particle of mass y, the total majorant rate at which this particle reacts with a $\oplus$ particle is equal to

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

Also, the majorant rate at which a coagulation event of the form $\oplus_x + \odot_y + \ominus_z \to \oplus_{x+y} + \ominus_{y+z}$ occurs is[4]

$$K_D^0(x,y,z) :=$$

$$\underbrace{\widehat{\rho}_2}_{\substack{\text{Total rate}\\\text{of process 2}}} \cdot \underbrace{\frac{\widehat{T}(+,y)+\widehat{T}(-,y)}{\sum_{y\geqslant 1}\left[\widehat{T}(+,y)+\widehat{T}(-,y)\right]\mu_t^{\odot,N}(y)}}_{\text{Choose }\odot\text{ particle}} \cdot \underbrace{\frac{\widehat{T}(\vee,y)}{\widehat{T}(+,y)+\widehat{T}(-,y)}}_{\text{1st Rejection step}} \cdot$$

$$\underbrace{\frac{\widehat{K}(x,y)}{\widehat{T}(+,y)}}_{\text{Choose a }\oplus_x} \cdot \underbrace{\frac{\widehat{K}(z,y)}{\widehat{T}(-,y)}}_{\text{Choose a }\ominus_z} \cdot \underbrace{\min\left\{\frac{K^-(z,y)}{\widehat{K}(z,y)}\frac{\widehat{T}(-,y)}{\widehat{T}(\vee,y)}, \frac{K^+(x,y)}{\widehat{K}(x,y)}\frac{\widehat{T}(+,y)}{\widehat{T}(\vee,y)}\right\}}_{\text{Probability of rejecting neither coagulation}}$$

which simplifies to

$$\min\left\{r^-(x,y,z), r^+(x,y,z)\right\},$$

where

$$r^-(x,y,z) := \frac{\widehat{K}(x,y)K^-(y,z)}{\widehat{T}(+,y)} \quad , \quad r^+(x,y,z) := \frac{\widehat{K}(y,z)K^+(x,y)}{\widehat{T}(-,y)} \tag{2.7}$$

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

$$\begin{aligned}\Delta_D^- &:= \left[\max\left\{r^-, r^+\right\} - \min\left\{r^-, r^+\right\}\right]\mathbb{1}_{r^+<r^-} \\ &= \max\{r^- - r^+, 0\},\end{aligned} \tag{2.8a}$$

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

$$\Delta_D^+ := \max\{r^+ - r^-, 0\}. \tag{2.8b}$$

As verification, we note that the rate at which a pair of particles $(\oplus_x, \odot_y)$ coagulates (in X^+) is

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

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

[4] We use the index D for "Double"; this distinguishes the quantities to be introduced from the similar ones introduced above for the Single coupling algorithm.

Algorithm 2: Double Coupling algorithm

```
   **The Double Coupling algorithm is the same as the Single Coupling
   algorithm, but modified by removing cases $k = 2a$ and $k = 2b$ and
   replacing 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

   $$\frac{\widehat{T}(+,\odot)+\widehat{T}(-,\odot)}{\sum_{\odot}[\widehat{T}(+,\odot)+\widehat{T}(-,\odot)]} = \frac{\widehat{T}(+,\odot)+\widehat{T}(-,\odot)}{\widehat{\rho}_2}. \qquad (2.9)$$

2  *Perform first rejection step*: with probability

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

   we continue, else reject by going to Step 10. This step is purely to
   transform the rate of process 2 into $\sum_{\odot}\widehat{T}(\vee,\odot)$ from $\widehat{\rho}_2$.
3  Choose a $\oplus$ and a $\ominus$ particle according to the respective distributions

   $$\frac{\widehat{K}(\oplus,\odot)}{\widehat{T}(+,\odot)} \quad \text{and} \quad \frac{\widehat{K}(\ominus,\odot)}{\widehat{T}(-,\odot)}. \qquad (2.10)$$

   We now have generated a triplet of particles $(\oplus,\odot,\ominus)$.
4  *Now consider the second rejection step* — define the respective
   probabilities of the $\ominus+\odot$ and $\oplus+\odot$ reactions occurring as:

   $$p_{\oplus+\odot} := \frac{\widehat{T}(+,\odot)}{\widehat{T}(\vee,\odot)}\frac{K^{+}(\oplus,\odot)}{\widehat{K}(\oplus,\odot)} \text{ and } p_{\ominus+\odot} := \frac{\widehat{T}(-,\odot)}{\widehat{T}(\vee,\odot)}\frac{K^{-}(\ominus,\odot)}{\widehat{K}(\ominus,\odot)}. \qquad (2.11)$$

   *In an almost identical fashion to Figure 1, do the following*:
5  Generate random variable **U** $\sim U(0,1)$.
   **if** $\boldsymbol{U} < \min\{p_{\oplus+\odot}, p_{\ominus+\odot}\}$ **then**
6      perform $\oplus+\odot+\ominus \rightarrow \oplus+\ominus$.
   **else if** $\min\{p_{\oplus+\odot}, p_{\ominus+\odot}\} \leqslant \boldsymbol{U} < \max\{p_{\oplus+\odot}, p_{\ominus+\odot}\}$ **then**
       **if** $p_{\oplus+\odot} > p_{\ominus+\odot}$ **then**
7          perform $\odot+\oplus \rightarrow \oplus+\ominus$.
       **else**
8          perform $\odot+\ominus \rightarrow \ominus+\oplus$.
   **else**
9      Go to Step 10.
10 Go to Step 16 of the Single Coupling algorithm (Algorithm 1) .
```

2.3.2 Implementation and Complexity

Looking at Step 3 of the Single coupling algorithm where the particle pair, and simultaneously the process k, we note that the combined process $\widehat{\rho_2}$ 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 2.9 and one of $\oplus$ and $\ominus$ is also automatically chosen with the correct distribution specified in equation 2.10 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 $\widehat{\rho_2}$ 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 we write $\mu_t^{\odot,N} = \frac{1}{N}\sum_x \delta_x$ for the empirical measure of the distribution of the $\odot$ particles at time t: an x appears in the sum only if there is $\odot$ particle of mass x at time t. The empirical measures $\mu_t^{\ominus,N}$ and $\mu_t^{\oplus,N}$ are defined in similar terms. In the same way as one can prove that the Marcus-Lushnikov process converges to the solution of Smoluchowski equation (when it is unique)[5], one can prove a similar result for the triple of *stochastic* processes $\left(\mu_\cdot^{\odot,N}, \mu_\cdot^{\ominus,N}, \mu_\cdot^{\oplus,N}\right)$. The limiting object $\left(\mu_\cdot^{\odot}, \mu_\cdot^{\ominus}, \mu_\cdot^{\oplus}\right)$ is a *deterministic* non-negative measure-valued path. Given three bounded functions f, g, h, it satisfies the system

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}(f, \mu_t^{\odot}) = {} & \frac{1}{2}\sum_{x,y\geqslant 1} \left[f(x+y) - f(x) - f(y)\right] K_S^0(x,y)\, \mu_t^{\odot}(x)\, \mu_t^{\odot}(y) \\ & - \sum_{x,y\geqslant 1} f(x)\, \left[\Delta_S^+(x,y) + \Delta_S^-(x,y)\right]\, \mu_t^{\odot}(x)\, \mu_t^{\odot}(y) \\ & - \sum_{x,y,z\geqslant 1} f(y)\, K_D^0(x,y,z)\, \mu_t^{\oplus}(x)\, \mu_t^{\odot}(y)\, \mu_t^{\ominus}(z) \\ & - \sum_{x,y,z\geqslant 1} f(y)\, \left[\Delta_D^+(x,y,z) + \Delta_D^-(x,y,z)\right]\, \mu_t^{\oplus}(x)\, \mu_t^{\odot}(y)\, \mu_t^{\ominus}(z)\end{aligned}$$

[5] See for instance the article [12] of J. Norris or [9] of I. Jeon.

and

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\big(g,\mu_t^{\oplus}\big) = {} & \frac{1}{2}\sum_{x,y\geqslant 1} g(x+y)\,\Delta_S^{+}(x,y)\,\mu_t^{\odot}(x)\,\mu_t^{\odot}(y) + \sum_{x,y\geqslant 1} g(x)\,\Delta_S^{-}(x,y)\mu_t^{\odot}(x)\mu_t^{\odot}(y) \\
& + \sum_{x,y,z\geqslant 1}[g(x+y)-g(x)]\,\big(K_D^0+\Delta_D^+\big)(x,y,z)\,\mu_t^{\oplus}(x)\,\mu_t^{\odot}(y)\,\mu_t^{\ominus}(z) \\
& + \sum_{x,y,z\geqslant 1} g(y)\,\Delta_D^{-}(x,y,z)\,\mu_t^{\oplus}(x)\,\mu_t^{\odot}(y)\,\mu_t^{\ominus}(z) \\
& + \frac{1}{2}\sum_{x,y\geqslant 1}[g(x+y)-g(x)-g(y)]\;K^+(x,y)\,\mu_t^{\oplus}(x)\mu_t^{\oplus}(y)
\end{aligned}
$$

A similar equation holds for $\frac{\mathrm{d}}{\mathrm{d}t}\big(h,\mu_t^{\ominus}\big)$. The reader will get a clear insight on the reason why these equations appear writing down the generator of the discrete-measure-valued Markov chain $\big(\mu^{\oplus,N},\mu^{\odot,N},\mu^{\ominus,N}\big)$. Note that if we set

$$
\mu_t^+ := \mu_t^{\odot} + \mu_t^{\oplus}, \qquad \mu_t^- := \mu_t^{\odot} + \mu_t^{\ominus},
$$

then we have for any bounded functions f and g

$$
\frac{\mathrm{d}}{\mathrm{d}t}(f,\mu_t^+) = \frac{1}{2}\sum_{x,y\geqslant 1}[f(x+y)-f(x)-f(y)]\;K^+(x,y)\,\mu_t^+(x)\,\mu_t^+(y)
$$

and

$$
\frac{\mathrm{d}}{\mathrm{d}t}(g,\mu_t^-) = \frac{1}{2}\sum_{x,y\geqslant 1}[g(x+y)-g(x)-g(y)]\;K^-(x,y)\,\mu_t^-(x)\,\mu_t^-(y)
$$

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

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')[6]

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

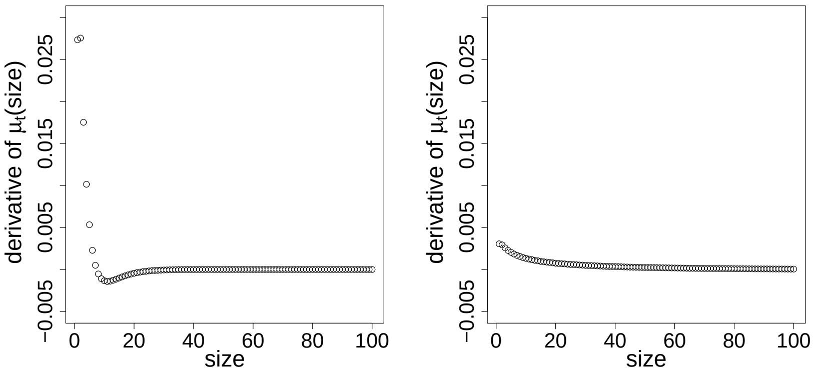
The reference value of λ for the additive kernel will be 1 and for the soot kernel 2.1. We shall always take as initial condition for the Marcus-Lushnikov process N particles with mass equal to 1. Throughout this section we shall denote by N the initial number

[6] This kernel is studied extensively in [8] and used in [5, 11, 15].

of particles in each system (which is the same), by λ the above reference value of the parameter, whose perturbation will be denoted by δ (i. e. the $X^{\pm}$ systems are governed by the parameter values $\lambda \pm \frac{1}{2}\delta$), and by L the number of simulations with the same initial conditions. The remaining notation is given below.

- t = time of evolution of the particle system
- t_{run} = time taken to run the algorithms (CPU time).
- The estimate of $\frac{\partial}{\partial\lambda}(f, \mu_t^{(\lambda)})$ given by the l^{th} simulation is denoted by $F_l^{(\lambda)}$([7]), where f is a suitable test function.
- The estimate of $\frac{\partial}{\partial\lambda}(f, \mu_t^{(\lambda)})$ given by L simulations is denoted by $\overline{F}^{(\lambda)}$; it is equal to $\frac{1}{L}\sum_{l=1}^{L} F_l^{(\lambda)}$.

3.1 Some initial plots



(a) $t = 0.5$

(b) $t = 4.0$

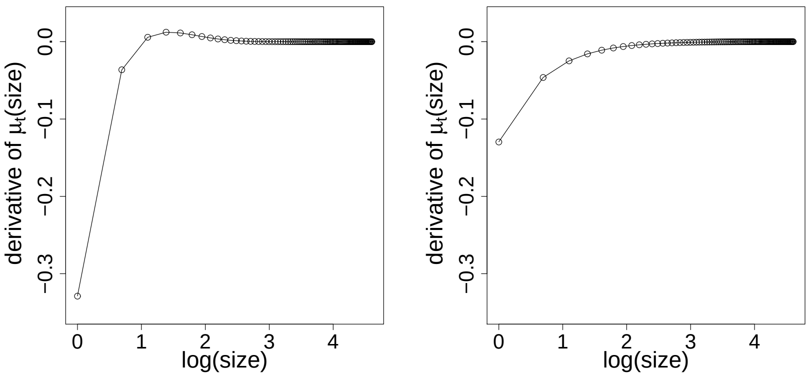
Figure 3: *Derivative $\frac{\partial}{\partial\lambda}\mu_t^{(\lambda)}(x)$ versus particle size x, for Soot kernel using the Double coupling algorithm, $\lambda = 2.1$, $\delta = 0.03$, $N = 10^7$, $L = 300$. Confidence intervals have been omitted since they are visually negligible.*

Figure 3 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 4 shows similar quantities, but for the Additive kernel; it is in good agreement with the analytic solution given in [6].

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 δ and the other due to assuming a finite particle system. It

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



(a) $t = 0.5$

(b) $t = 2.0$

Figure 4: *Derivative $\frac{\partial}{\partial\lambda}\mu_t^{(\lambda)}(x)$ versus particle size x, for additive kernel using the Double coupling algorithm, $\lambda = 1$, $\delta = 0.06$, $N = 10^6$, $L = 300$. The broken line joins the points of the analytic solution. Confidence intervals have been omitted since they are visually negligible.*

is the latter we investigate — here we estimate the order of convergence of the systematic error as N varies. The value of δ 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.

$$e_{\text{sys}}(N; \lambda, \delta, t) = \mathbb{E}\overline{F}^{(\lambda,\delta)}(N; t) - f^{(\lambda,\delta)}(t) \tag{3.1}$$

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

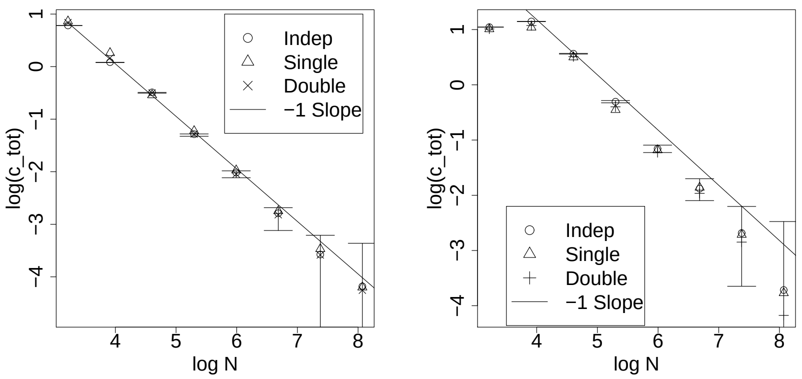
$$\overline{F}^{(\lambda,\delta)}(N_{\text{large}}; t) \tag{3.2}$$

for very large N_{large}. Also, $\mathbb{E}\overline{F}^{(\lambda,\delta)}(N; t)$ is estimated by $\overline{F}^{(\lambda,\delta)}(N; t)$. Now we set test function $f(y) = \mathbb{1}_{\{y=i\}}$ again, to ensure that $\overline{F}^{(\lambda,\delta)}(N; t)$ is an estimate of the number density for particle size i. Therefore we rename $\overline{F}^{(\lambda,\delta)}(N; t)$ as $\overline{F}^{(\lambda,\delta,i)}(N; t)$ for number density estimate at particle size $i \in \mathbb{N}$; we adopt analogous notation $f^{(\lambda,\delta,i)}$ and $e_{\text{sys}}(N; \lambda, \delta, t, i)$ 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 $(t_k)_{k=1}^{T}$[8] and summed over particle sizes i:

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

Figure 5 shows what we expect — the $c_{\text{tot}} \sim \frac{1}{N}$. We obtain similar plots for different values of δ.

[8] We take the t_k to be $(0.5, 1.0, \ldots, 7.0)$



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

Figure 5: $\log c_{tot}$ *versus* $\log N$, $N = 25 \times 2^i$ *for* $i = 0, \ldots, 7$, $N \times L = 10^8$ *and* $0 \leqslant t \leqslant 3$. *Confidence intervals given are for the Independent case only.*

3.3 Statistical error

This quantity is defined as

$$e_{\text{stat}}(N; t, \lambda, \delta) = \overline{F}^{(\lambda,\delta)}(N; t) - \mathbb{E}\overline{F}^{(\lambda,\delta)}(N; t). \tag{3.4}$$

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

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

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

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

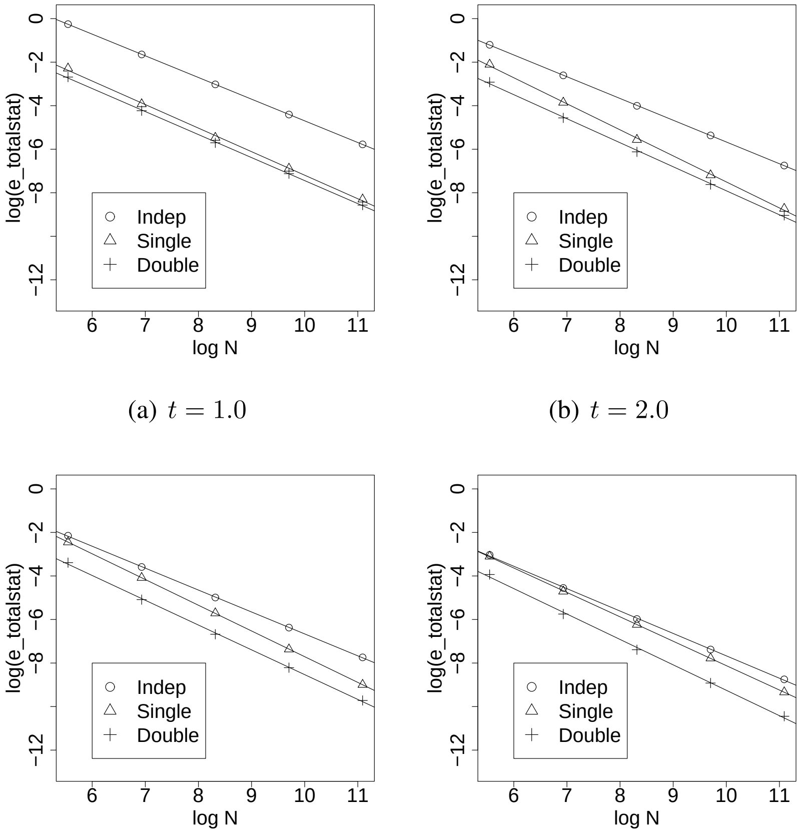
where $z_{\alpha/2}$ is the upper $\alpha/2$ point of the standard normal distribution. Hence,

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

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:

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

We wish to see how this quantity behaves with N. Figure 6 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^- and X^+ systems diverge from each other, but faster for the Single algorithm than for the Double algorithm.



(a) $t = 1.0$

(b) $t = 2.0$

(c) $t = 3.0$

(d) $t = 4.0$

Figure 6: $\log e_{\textit{totalstat}}$ *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* $\delta = 0.06$.

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 δ are fixed). Before defining it, we shall

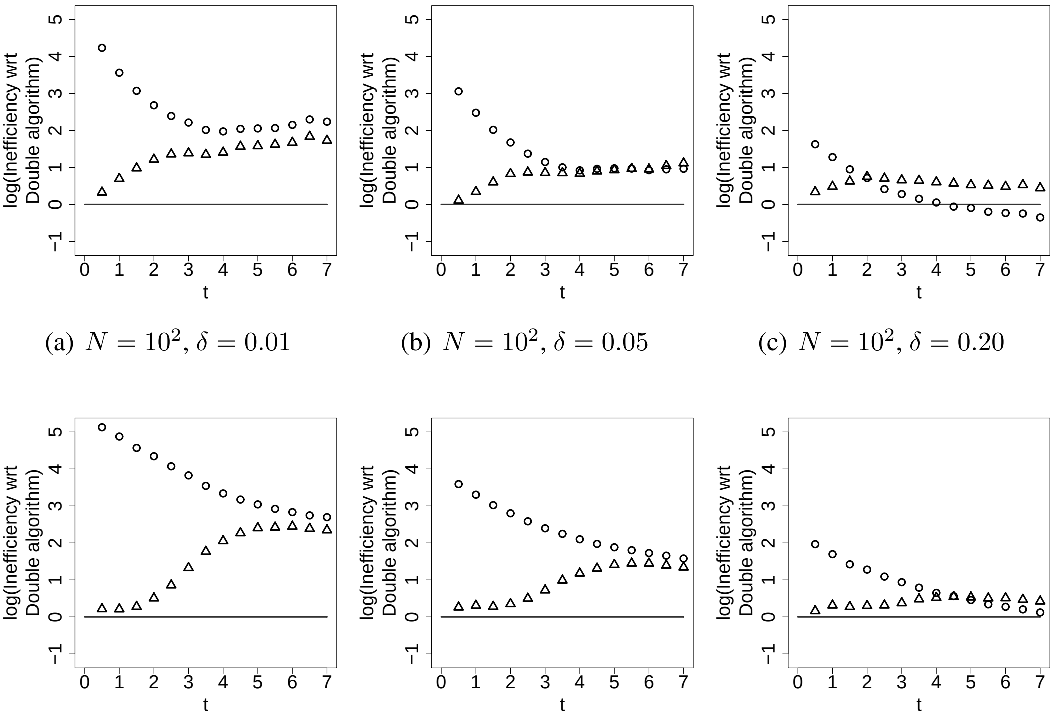
- Set $e_{\text{tot}} = \frac{e_{\text{totalstat}}}{L}$; this "total standard error" will be artificially **fixed**.
- Let $L_{\text{mod}}(t)$ be the estimated number of simulations required to acquire the fixed $e_{\text{tot}}(t)$ value and $t_{\text{mod}}(t)$ be the CPU time required to perform $L_{\text{mod}}(t)$ runs.

The condition $e_{\text{tot}}(t) = \sum_{i \geqslant 1} \frac{v_F^{(i)}}{L_{\text{mod}}(t)}$ implies that $L_{\text{mod}}(t) = \frac{\sum_{i \geqslant 1} v_F^{(i)}}{e_{\text{tot}}}$; we thus have $t_{\text{mod}}(t) = \frac{L_{\text{mod}}(t)}{L}$. We shall use the following quantity to compare different algorithms.

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

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 7 plots these inefficiencies. One can see that the Indep 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 δ 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 7 uses log scales) indicating that the Double algorithm has a significant improvement over the Single in terms of accuracy. Also, the inefficiencies decrease with δ. Furthermore, one notices that as the larger value of $N = 10^5$ results in a larger inefficiency for the Independent and Single algorithms for the Additive kernel, but that this increase is smaller for larger δ, and that the inefficiency of the Single coupling algorithm appears to not be so dependent on N for the Soot 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 you would expect. More importantly, as δ increases, the inefficiencies dramatically decrease in general (to the point that the inefficiencies go below the threshold value) as a result of significantly larger run times of the Single and Double for large δ (the run times increase linearly with δ for the Single and Double, whereas they are almost constant with respect to δ for the Independent algorithm). Note however that the case $\delta = 0.2$ is unlikely to be computationally useful as it amounts to a 20% change in the parameter, which enters multiplicatively into the kernel. 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.

It is important to realise that this analysis does not take into account the systematic error due to δ or N since the Inefficiency metric only uses estimated variances. A related



(a) $N = 10^2$, $\delta = 0.01$

(b) $N = 10^2$, $\delta = 0.05$

(c) $N = 10^2$, $\delta = 0.20$

(d) $N = 10^5$, $\delta = 0.01$

(e) $N = 10^5$, $\delta = 0.05$

(f) $N = 10^5$, $\delta = 0.20$

Figure 7: *Additive kernel — Inefficiency relative to Double for Indep and Single algorithms, as a function of t for different values of δ and N. The Independent algorithm is represented by circles, Single by triangles and the Double threshold by the horizontal line.*

problem with the analysis is that the number of particles for $t \in [3.5, 7.0]$ becomes quite small, and therefore the systematic errors and estimated variances are not very reliable.

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 δ. 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 δ and t, whilst being significantly more ‘efficient’ than the Independent algorithm for small t, large N and small δ, though some of this advantage is lost for larger t and δ.

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] D. J. Aldous. Deterministic and Stochastic Models for Coalescence (Aggregation and Coagulation) : a Review of the Mean-Field Theory for Probabilists. *Bernoulli*, 5, 1999.

[2] S. Asmussen and P. W. Glynn. *Stochastic Simulation: Algorithms and Analysis*. Springer, 2007.

[3] 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.*, 62:4717–4728, 2007.

[4] H. Briesen. Hierarchical characterization of aggregates for Monte Carlo simulations. *AIChE J.*, 52:2436–2446, 2006.

[5] 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*, 148:158–176, 2007.

[6] A. Eibeck and W. Wagner. An Efficient Stochastic Algorithm for Studying Coagulation Dynamics and Gelation Phenomena. *SIAM J. Sci. Comput.*, 22:802–821, 2000.

[7] A. Eibeck and W. Wagner. Stochastic Particle Approximations for Smoluchoski's Coagulation Equation. *Ann. Appl. Probab.*, 11:1137–1165, 2001.

[8] M. Goodson and M. Kraft. An Efficient Stochastic Algorithm for Simulating Nanoparticle Dynamics. *J. Comput. Phys.*, 183:210–232, 2002.

[9] I. Jeon. Existence of gelling solutions for coagulation-fragmentation equations. *Comm. Math. Phys.*, 194:541–567, 1998.

[10] A. Kolodko and K. Sabelfeld. Stochastic Particle Methods for Smoluchowski Coagulation Equation: Variance Reduction and Error Estimations. *Monte Carlo Methods Appl.*, 9:315–339, 2003.

[11] 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.*, 31:693–700, 2007.

[12] J. R. Norris. Smoluchowski's Coagulation Equation: Uniqueness, Nonuniqueness and a Hydrodynamic Limit for the Stochastic Coalescent. *Ann. Appl. Probab.*, 9: 78–109, 1999.

[13] R. I. A. Patterson and M. Kraft. Models for the aggregate structure of soot particles. *Combust. Flame*, 151:160–172, 2007.

[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.

[15] R. I. A. Patterson, J. Singh, M. Balthasar, M. Kraft, and W. Wagner. Extending stochastic soot simulation to higher pressures. *Combust. Flame*, 145, 2006.

[16] A. Vikhansky and M. Kraft. A Monte Carlo methods for identification and sensitivity analysis of coagulation processes. *J. Comput. Phys.*, 200(1):50–59, 2004.

[17] 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: 4966–4972, 2006.

[18] 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: 1441–1450, 2006.

[19] 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.*, 173:38–50, 2007.

[20] 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: 1412–1428, 2009.