Weierstraß-Institut
für Angewandte Analysis und Stochastik
Leibniz-Institut im Forschungsverbund Berlin e. V.

Preprint ISSN 0946 – 8633

Stochastic weighted particle methods for population balance equations

Robert I. A. Patterson[*,1] Markus Kraft[2] , Wolfgang Wagner[1]

submitted: March 16, 2011

[1] Weierstraß-Institut
Mohrenstraße 39
10117 Berlin
Germany
E-Mail: robert.patterson@wias-berlin.de
E-Mail: wolfgang.wagner@wias-berlin.de

[2] Department of Chemical Engineering
and Biotechnology
University of Cambridge
Pembroke Street
Cambridge, CB2 3RA
United Kingdom
E-Mail: mk306@cam.ac.uk

No. 1597
Berlin 2011

2010 *Mathematics Subject Classification.* 60J28, 65C05, 65C35.

Key words and phrases. Monte Carlo, weighted particle, coagulation, surface growth, Smoluchowski, simulation .

*) corresponding author.

Edited by
Weierstraß-Institut für Angewandte Analysis und Stochastik (WIAS)
Leibniz-Institut im Forschungsverbund Berlin e. V.
Mohrenstraße 39
10117 Berlin
Germany

Fax: +49 30 2044975
E-Mail: preprint@wias-berlin.de
World Wide Web: http://www.wias-berlin.de/

Abstract A class of stochastic algorithms for the numerical treatment of population balance equations is introduced. The algorithms are based on systems of weighted particles, in which coagulation events are modelled by a weight transfer that keeps the number of computational particles constant. The weighting mechanisms are designed in such a way that physical processes changing individual particles (such as growth, or other surface reactions) can be conveniently treated by the algorithms. Numerical experiments are performed for complex laminar premixed flame systems. Two members of the class of stochastic weighted particle methods are compared to each other and to a direct simulation algorithm. One weighted algorithm is shown to be consistently better than the other with respect to the statistical noise generated. Finally, run times to achieve fixed error tolerances for a real flame system are measured and the better weighted algorithm is found to be up to three times faster than the direct simulation algorithm.

1. Introduction

In the Direct Simulation Monte Carlo approach to kinetic equations, like the Boltzmann and the Smoluchowski equation, every computational particle represents the same number of physical particles. The size of this number determines the discretisation error—smaller values mean better approximations of the mean field. The direct proportionality between physical particle concentration and computational particle number implies that the direct simulation approach yields relatively little information about the rarest particles. In gas dynamics this means that regions of low density are subject to large amounts of noise. Coagulation problems require large amounts of computational effort to be spent on very common particles in order to obtain useful estimates concerning the concentrations of rarer, physically significant large particles.

To allow more efficient use of computational resources, stochastic weighted particle methods have been used for gas dynamics [24, 31, 26]. An implicit weighting scheme particularly suited to coagulation and fragmentation problems was developed in [7] (see also the review in [28]). In this method the statistical weight of a computational particle was a function of the physical particle it described and so there was no need to explicitly tag computational particles with statistical weights. The particular statistical weight used was the inverse of the physical particle mass, which has the useful consequence that each computational particle represents the same mass of physical particles per unit volume. The method is therefore known as the Mass Flow Algorithm (MFA). A discrete (both in time and size) version of the MFA was proposed in [2]. A more general consideration of statistical weights as functions of the physical particle properties was undertaken in [13]. The algorithms included the special case of the MFA, but the greater generality came at the price of coagulation events that sometimes increased and sometimes decreased the number of computational particles.

In [5], where the physical motivation was atmospheric aerosols, computational particles were assigned weights that were not simply functions of the physical particles they represented. Using operator splitting and deterministic integration, surface process simulation included updating the statistical weight of a computational particle so that it represented the same number of physical particles before and after the surface events. Coupled with an MFA approach to coagulation this meant that the only change in the number of computational particles during simulations was due to processes of particle inception. Explicit statistical weights were used in [10], which follows the lines of [2] and [5] by using a fixed time step to calculate the correct number of random events. Increasingly complex weighting rules for coagulation, as well as splitting coagulation into several different sub-processes, have been proposed in [33].

In [30] the MFA was applied to a coagulation and sintering problem in nanoparticle dynamics. Consideration of processes of engineering interest inevitably led to a desire to simulate systems which exchange mass with their surroundings and thus a MFA cannot be expected to maintain a constant number of computational particles without continual resampling of the distribution. Nevertheless in [17, 18] the MFA was successfully extended to systems with particle inception and where individual particle masses changed by interaction with the environment. In [17], surface growth was simulated as two separate processes. In one, the mass of physical particles referred to by a computational particle is changed and in the other, new computational particles are introduced to the system to account for the increase in mass.

The purpose of this paper is to present a family of stochastic algorithms with explicitly weighted computational particles for physical coagulation and growth problems, principally soot formation in laminar premixed flames [3]. These algorithms offer most of the benefits of the MFA (modelling coagulation with a constant number of computational particles, better resolution of the concentration of large particles) in a framework that also makes the simulation of mass addition and removal from the surface of particles convenient.

In Section 2 the population balance equation that defines the problem to be solved is presented. Some explanation of its physical applications in combustion modelling is given. In Section 3 the class of stochastic weighted particle methods is described in detail. Properties of algorithms are studied, including convergence to the solution of the population balance equation. In Section 4 the numerical properties of two of the weighted algorithms are explored by applying them to realistic problems and comparing them to a direct simulation algorithm. Finally, conclusions and an outlook to future research are given in Section 5.

2. Population Balance Equations

The population balance problems considered in this paper are taken from work on soot formation and growth. Other physical processes can be modelled by equations with the same structure; examples include the formation of inorganic nano-particles [14], proto-planet formation [19] and the development of cloud and rain drops [5].

2.1. Extended Smoluchowski Equation

We consider a population balance equation which is an extended version of the Smoluchowski coagulation equation. The original Smoluchowski equation deals only with the coagulation of pairs of particles at rates specified by a collision kernel K. The extension represents physical processes which exchange mass between individual particles and the environment, for example the gas in which the population is contained, and introduce new particles into the population. The equation, which is for spatially homogeneous systems, is formulated for the concentration c at time t of particles of size $x = 1, 2, \ldots$ as

$$\frac{\mathrm{d}}{\mathrm{d}t} c(t,x) = \mathcal{K}(c)(t,x) + \sum_{l=1}^{L} \mathcal{S}_l(c)(t,x) + I(x)\,. \tag{1}$$

Particles of size x coagulate with ones of size y at rate $K(x,y)$ so that

$$\mathcal{K}(c)(x) = \frac{1}{2} \sum_{y,z \in \mathcal{X}: y+z=x} K(y,z)\, c(t,y)\, c(t,z) - c(t,x) \sum_{y \in \mathcal{X}} K(x,y)\, c(t,y)\,, \tag{2}$$

where $\mathcal{X} = \{1, 2, \ldots\}$. The first term in (2) represents the formation of new larger particles and the second term the disappearance of the smaller particles that are consumed to make the larger ones. Each process of mass exchange between individual particles and the environment is represented by an $\mathcal{S}_l$. Single particle processes can helpfully be thought of as happening on the surface of particles—the interface between particles and their surroundings—without implying any claims about particular physical–chemical mechanisms. Particles of size x interact with their surroundings at rate $\beta_l(x)$ and are transformed into particles of size $g_l(x)$. One then has

$$\mathcal{S}_l(c)(t,x) = \sum_{y \in \mathcal{X}: g_l(y)=x} \beta_l(y)\, c(t,y) - \beta_l(x)\, c(t,x)\,, \tag{3}$$

where, as for coagulation, the equation is written with the formation term preceding the loss term. Finally, particles of size x are incepted at rate $I(x)$.

2.2. Soot in Laminar Premixed Flames

Laminar premixed flames are an important class of systems for the study of soot kinetics. They allow experimentalists and modellers to avoid complexities such as mixing (except for the diffusion of very light chemical species) and turbulence. In such a system a perfect mixture of carbon based fuel, oxidiser and an inert gas flows into a flame through a device similar to a Bunsen burner. Under suitable conditions a steady flame forms, which is approximately uniform across any horizontal plane, that is, the flame is a 1-dimensional system and all properties can be adequately described as functions only of the height above the burner through which the gas enters the flame.

Simulations of soot formation begin with the calculation of chemical species concentrations in the flame; from a mathematical point of view this means the solution of a set of reaction–diffusion PDEs in one spatial dimension. The solution, which consists of species concentrations on an adapted grid, is stored in a file which can be reused for every simulation of the flame with linear interpolation between the points [3]. The solution is assumed to vary at a rate that is much slower than the rates of the particle processes, the rate of variation can conveniently be estimated from the time spacing of the adapted grid, since this is small in regions where the solution is changing quickly.

The 1-dimensional structure of the laminar gas flow in the flame means that there is a one-to-one transformation between the residence time of a small volume of gas in the flame and the height of that volume above the burner through which it entered. The soot simulation is performed by calculating the time evolution of the soot population in such a closed, homogeneous volume as it moves through the flame along a Lagrangian trajectory. Initial results are therefore expressed as a function of time but are usually transformed back to be functions of height. Within such a volume, as it moves up the flame expanding and contracting, the following processes occur:

1 Soot particles are incepted as spheres containing 32 carbon atoms at a rate I, which depends on the precalculated chemical species concentrations.

2 Each soot particle of type x coagulates with particles of type y at rate $K(x, y)$. The form of K used is the 'transition regime kernel' as described in more detail in [21].

3 Pyrene molecules (16 carbon atoms, hydrogen is ignored) condense on the surface of soot particles of type x at rate $\beta_1(x)$, the new particle type is $g_1(x)$.

4 Acetylene molecules react with the surface of a soot particle of type x at rate $\beta_2(x)$ causing the particle to grow by 2 carbon atoms; the new particle type is $g_2(x)$.

5 Oxygen molecules react with the surface of a soot particle of type x at rate $\beta_3(x)$ causing the particle to lose 2 carbon atoms; the new particle type is $g_3(x)$.

6 OH radicals react with the surface of a soot particle of type x at rate $\beta_4(x)$ causing the particle to lose 1 carbon atom; the new particle type is $g_4(x)$.

For the numerical examples in this work all soot particles are assumed to be spheres with density 1.8 g cm^{-3} and the particle size is measured by mass expressed in units of 1 Carbon atom so that the type space $\mathcal{X} = \{1, 2, \ldots\}$ can be used. More advanced soot models require very complex type spaces [4]. Therefore, the theory of stochastic weighted particle methods will be developed for rather general type spaces.

3. Stochastic Weighted Particle Methods

Stochastic weighted particle methods are based on Markov jump processes that are related to the corresponding population balance equation. In this section a detailed description of these connections will be given.

Here we consider a general type space $\mathcal{X}$, which might be a combination of various finite-dimensional domains (formally, it should be a locally compact separable metric space). This generality allows the theory to be applied to particles composed of several substances, or even having a more complicated internal structure. It is supposed that an operation "+" is defined in $\mathcal{X}$, which describes the coagulation of two particles. We use concentration measures $c(t, dx)$ (instead of densities $c(t, x)$) so that cases, where the type of a particle has both discrete and continuous components, are covered.

We consider the population balance equation in a weak form

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\mathcal{X}}\varphi(x)\,c(t,dx) = \frac{1}{2}\int_{\mathcal{X}}\int_{\mathcal{X}}[\varphi(x+y)-\varphi(x)-\varphi(y)]\,K(x,y)\,c(t,dx)\,c(t,dy) + \sum_{l=1}^{L}\int_{\mathcal{X}}[\varphi(g_l(x))-\varphi(x)]\,\beta_l(x)\,c(t,dx) + \int_{\mathcal{X}}\varphi(x)\,I(dx) \tag{4}$$

where φ belongs to an appropriate class of test functions (e.g., sufficiently smooth, with compact support). Equation (4) reduces to (1) in the discrete case $\mathcal{X}=\{1,2,\ldots\}$.

We consider systems of particles

$$\Big(x_i(t),u_i(t)\Big) \qquad i=1,\ldots,N(t)\,, \qquad t\geq 0\,, \tag{5}$$

where $x_i(t)\in\mathcal{X}$ is the type of the particle and $u_i(t)\in(0,u_{\max}]$ is interpreted as the particle weight. The stochastic particle system (5) approximates the solution of the deterministic equation (4) in the sense

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{N(t)}u_i(t)\,\varphi(x_i(t)) = \int_{\mathcal{X}}\varphi(x)\,c(t,dx)\,, \tag{6}$$

where $\lim_{n\to\infty}$ denotes convergence in probability. The numerical parameter n characterizes the level of discretization. It is related to the number of numerical particles, and to the inverse of the number of physical particles represented by one numerical particle.

In the next subsection the time evolution of the Markov jump process (5) will be specified by defining several jump transformations (instantaneous discontinuous transitions) and the rates (frequencies) at which the various jumps happen. Then the property (6) will be derived from general convergence theory. Finally, algorithmic issues of stochastic weighted particle methods will be addressed.

3.1. Jump Dynamics

Here we define the time evolution of the particle system (5). Given the system is in a state

$$z=\Big((x_1,u_1),\ldots,(x_N,u_N)\Big), \tag{7}$$

it stays there for a random time τ , which has an exponential distribution,

$$\mathbb{P}(\tau\geq s)=\exp(-\lambda(z)\,s)\,, \qquad s\geq 0\,, \tag{8}$$

where

$$\lambda(z)=\lambda_{\mathrm{inc}}(z)+\lambda_{\mathrm{sur}}(z)+\lambda_{\mathrm{coa}}(z)\,. \tag{9}$$

The rate functions λ_{inc} , λ_{sur} , λ_{coa} will be specified below. At time τ , a jump mechanism is chosen according to the probabilities

$$\frac{\lambda_{\mathrm{inc}}(z)}{\lambda(z)}\,, \qquad \frac{\lambda_{\mathrm{sur}}(z)}{\lambda(z)}\,, \qquad \frac{\lambda_{\mathrm{coa}}(z)}{\lambda(z)}\,, \tag{10}$$

and a corresponding jump is performed.

3.1.1. Inception jumps

A particle
$$(x, w(x)) \tag{11}$$
is added to the system. The inception rate is
$$\lambda_{\mathrm{inc}}(z) = n\,\tilde{I}(\mathcal{X})\,. \tag{12}$$
The type x of the particle is generated according to the distribution
$$\frac{1}{\tilde{I}(\mathcal{X})}\,\tilde{I}(dx)\,. \tag{13}$$
Its weight is calculated as a deterministic function w of the type.

The finite measure $\tilde{I}$ and the function w provide degrees of freedom. They are assumed to satisfy the relation
$$w(x)\,\tilde{I}(dx) = I(dx)\,. \tag{14}$$

3.1.2. Surface reaction jumps

One particle changes its type according to some surface reaction, without changing its weight, that is
$$(x_i, u_i) \to (g_l(x_i), u_i)\,. \tag{15}$$
The surface reaction rate is
$$\lambda_{\mathrm{sur}}(z) = \sum_{i=1}^{N}\sum_{l=1}^{L}\beta_l(x_i)\,. \tag{16}$$
The particle index i and the reaction index l are chosen according to the probabilities
$$\frac{\beta_l(x_i)}{\lambda_{\mathrm{sur}}(z)}\,. \tag{17}$$

Surface processes have no effect on particle concentration, they just modify the type of a particle. Thus, there are no degrees of freedom due to the introduction of weights.

3.1.3. Coagulation jumps

Two particles are involved. One particle gets the type, which is the result of coagulation, and changes its weight. The other particle does not change, but influences the transformation of the first particle. The jump has the form
$$(x_i, u_i),\quad (x_j, u_j) \quad\to\quad (x_i + x_j, \gamma(x_i, u_i, x_j, u_j)),\quad (x_j, u_j)\,. \tag{18}$$
The coagulation rate is
$$\lambda_{\mathrm{coa}}(z) = \frac{1}{2n}\sum_{i,j=1}^{N} K(x_i, x_j)\,u_j\,. \tag{19}$$

The particle indices i,j are chosen according to the probabilities

$$\frac{K(x_i,x_j)\,u_j}{2n\,\lambda_{\mathrm{coa}}(z)}\,. \tag{20}$$

Consider the following form for the function γ, which defines the weight transformation in (18):

$$\gamma(x,u,y,v) = u\,\alpha(x,u,y,v), \tag{21}$$

where

$$\alpha(x,u,y,v) + \alpha(y,v,x,u) = 1. \tag{22}$$

The idea is that a proportion $\alpha(x,u,y,v)$ of the coagulation events will lead to a particle with weight $\gamma(x,u,y,v)$ and the remaining events will lead to a particle with weight $\gamma(y,v,x,u)$. The extra factor of u is to balance the appearance of u_j in (20) so that there is sufficient symmetry to allow the derivation of a coagulation algorithm that only changes one computational particle.

The positive function α provides a degree of freedom.

3.1.4. Conservation property

In this section a conservation property is derived, which holds in mean for any weight transfer function γ satisfying (21), (22). Let M be a quantity conserved during coagulation, that is

$$M(x+y) = M(x) + M(y)\,. \tag{23}$$

Denote the states of the system (cf. (7)) before and after a coagulation jump by z and $\tilde{z}$, respectively. Then

$$\mathbb{E}\left(\sum_{k=1}^{N} \tilde{u}_k\, M(\tilde{x}_k)\right) = \sum_{k=1}^{N} u_k\, M(x_k)\,, \tag{24}$$

where $\mathbb{E}$ denotes mathematical expectation. Indeed, according to (18), (20), one obtains

$$\mathbb{E}\left(\sum_{k=1}^{N}\left[\tilde{u}_k\, M(\tilde{x}_k) - u_k\, M(x_k)\right]\right) = \sum_{i,j=1}^{N} \frac{K(x_i,x_j)\,u_j}{2n\,\lambda_{\mathrm{coa}}(z)}\left[\gamma(x_i,u_i,x_j,u_j)\,M(x_i+x_j) - u_i\,M(x_i)\right] \tag{25}$$

so that (24) is equivalent to

$$\sum_{i,j=1}^{N} K(x_i,x_j)\,u_j\,\gamma(x_i,u_i,x_j,u_j)\,M(x_i+x_j) = \sum_{i,j=1}^{N} K(x_i,x_j)\,u_j\,u_i\,M(x_i)\,. \tag{26}$$

Using (21), (22) and (23), the left-hand side of (26) is transformed into

$$\sum_{i,j=1}^{N} K(x_i,x_j)\,u_j\,u_i\,\alpha(x_i,u_i,x_j,u_j)\,M(x_i+x_j) = \frac{1}{2}\sum_{i,j=1}^{N} K(x_i,x_j)\,u_j\,u_i\,[M(x_i)+M(x_j)] \quad (27)$$

which equals the right-hand side of (26). Note that K is assumed to be symmetric.

3.1.5. Example coagulation weightings

Here we derive several examples of weight transfer functions γ (cf. (18)) satisfying (21), (22). If γ is assumed to be symmetric, then (21), (22) imply

$$u\,\alpha(x,u,y,v) = v\,[1-\alpha(x,u,y,v)] \quad (28)$$

so that

$$\gamma(x,u,y,v) = \frac{u\,v}{u+v}\,. \quad (29)$$

If α is assumed to be symmetric, then (21), (22) imply

$$\gamma(x,u,y,v) = \frac{1}{2}\,u\,. \quad (30)$$

The choice (30) was used in [10].

Let M be any quantity satisfying (23). Then the coagulation jump is conservative in the sense (cf. (18))

$$\gamma(x,u,y,v)\,M(x+y) + v\,M(y) = u\,M(x) + v\,M(y) \quad (31)$$

if and only if (cf. (21))

$$\gamma(x,u,y,v) = u\,\frac{M(x)}{M(x)+M(y)}\,. \quad (32)$$

A one-dimensional version of the weight transfer function (32) was used in [5].

If the weights are given as a deterministic function κ of the type, then one obtains (cf. (18))

$$\gamma(x,u,y,v) = \kappa(x+y)\,, \qquad u=\kappa(x)\,, \qquad v=\kappa(y)\,. \quad (33)$$

Conditions (21), (22) imply

$$\frac{\kappa(x+y)}{\kappa(x)} + \frac{\kappa(x+y)}{\kappa(y)} = 1 \quad (34)$$

so that

$$\kappa(x) = \frac{1}{M(x)}\,, \quad (35)$$

for some conserved quantity M (cf. (23)). One obtains (cf. (33))

$$\gamma(x,u,y,v) = \frac{1}{M(x)+M(y)} = u\,\frac{M(x)}{M(x)+M(y)} = \frac{\frac{1}{M(x)\,M(y)}}{\frac{1}{M(x)}+\frac{1}{M(y)}} = \frac{u\,v}{u+v} \tag{36}$$

so that the weight transfer functions (29) and (32) are identical in this situation. In the one-dimensional case, they reduce to the standard mass flow model [7].

3.2. Convergence

Consider inception jumps and surface reaction jumps as described in Section 3.1 (cf. (11), (15)). Let coagulation jumps of the form

$$(x_i,u_i),\quad (x_j,u_j) \quad\to\quad (z_1,\gamma_1),\quad (z_2,\gamma_2) \tag{37}$$

happen according to some intensity $\tilde{K}(x_i,u_i,x_j,u_j)$ (cf. (18), (19)), where the functions $z_1,\gamma_1,z_2,\gamma_2$ depend on x_i,u_i,x_j,u_j . Then general convergence results (e.g., Cor. 2.7 in [8]) imply

$$\lim_{n\to\infty}\frac{1}{n}\sum_{i=1}^{N(t)}\psi(x_i(t),u_i(t)) = \int_E \psi(x,u)\,f(t,dx,du)\,, \tag{38}$$

where

$$E = \mathcal{X}\times(0,u_{\max}] \tag{39}$$

is an extended state space and the function f is characterised by the equation

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\int_E \psi\,(x,u)\;f\,(t,dx,du) = &\\
&\int_E\int_E \Big[\psi(z_1,\gamma_1)+\psi(z_2,\gamma_2)-\psi(x,u)-\psi(y,v)\Big]\times\\
&\qquad\qquad \tilde{K}(x,u,y,g)\,f(t,dx,du)\,f(t,dy,dv)\\
&+\sum_{l=1}^{L}\int_E\Big[\psi(g_l(x),u)-\psi(x,u)\Big]\beta_l(x)\,f(t,dx,du)\\
&\qquad\qquad +\int_{\mathcal{X}}\psi(x,w(x))\,\tilde{I}(dx)
\end{aligned} \tag{40}$$

where the functions $z_1,\gamma_1,z_2,\gamma_2$ depend on x,u,y,v . It is assumed that the rate functions are such that existence and uniqueness of a solution hold for equation (40).

We use test functions of the form

$$\psi(x,u) = u\,\varphi(x)\,. \tag{41}$$

Then (38) implies (6), where

$$c(t,dx) = \int_0^{u_{\max}} u\,f(t,dx,du)\,. \tag{42}$$

Consider (cf. (18))

$$z_1(x,u,y,v) = x + y \qquad\qquad \gamma_1(x,u,y,v) = \gamma(x,u,y,v)$$
$$z_2(x,u,y,v) = y \qquad\qquad \gamma_2(x,u,y,v) = v$$

where γ satisfies (21), (22), and (cf. (19))

$$\tilde{K}(x,u,y,v) = K(x,y)\, v\,. \tag{44}$$

Note that

$$\begin{aligned}\Big[\gamma_1\,\varphi(z_1) + \gamma_2\,\varphi(z_2) - u\,\varphi(x) - v\,\varphi(y)\Big]\,\tilde{K}(x,u,y,v)&\\ = \Big[\alpha(x,u,y,v)\,\varphi(x+y) - \varphi(x)\Big]\,K(x,y)\,u\,v&\end{aligned} \tag{45}$$

and (since K is symmetric)

$$\begin{aligned}&\int_E\int_E \Big[\alpha(x,u,y,v)\,\varphi(x+y) - \varphi(x)\Big]\,K(x,y)\,u\,v\,f(t,dx,du)\,f(t,dy,dv)\\ &\quad = \int_E\int_E \Big[\alpha(y,v,x,u)\,\varphi(x+y) - \varphi(y)\Big]\,K(x,y)\,u\,v\,f(t,dx,du)\,f(t,dy,dv)\\ &\quad = \frac{1}{2}\int_E\int_E \Big[\varphi(x+y) - \varphi(x) - \varphi(y)\Big]\,K(x,y)\,u\,v\,f(t,dx,du)\,f(t,dy,dv).\end{aligned} \tag{46}$$

According to (41), (42), (45), (46) and (14), equation (40) takes the form (4).

3.3. *Implementation issues*

The stochastic weighted algorithm is used to generate the time evolution of the particle system (5) using pseudo-random numbers. Property (6) ensures that the solution to equation (4) is approximated by the algorithm, when the numerical parameter n becomes large. Approximation is understood in the probabilistic sense, that is, the result provided by the algorithm and the solution are close to each other with sufficiently large probability. Confidence intervals are constructed using independent trajectories of the particle system.

In this section we briefly discuss issues related to efficiency, that is, how to generate the trajectories of the system (5) efficiently. Various general mathematical and computational techniques are needed to reduce the computational cost. The main thing is to avoid calculating rates for each possible jump individually, but to consider classes of jumps and then to have an efficient way of selecting an individual particle or pair of particles once an overall event class has been chosen.

3.3.1. *Majorant techniques*

An important tool for reducing computational costs are majorant techniques [6, 21]. Provided the coagulation kernel satisfies, e.g.,

$$\tilde{K}(x,y) \le h_1^{(1)}(x)\,h_2^{(1)}(y) + h_1^{(2)}(x)\,h_2^{(2)}(y)\,, \tag{47}$$

the calculation of the majorant coagulation rate can be factorised as

$$\sum_{i,j=1}^{N(t)} \tilde{K}(x_i, x_j) \leq \left(\sum_{i=1}^{N(t)} h_1^{(1)}(x_i)\right)\left(\sum_{j=1}^{N(t)} h_2^{(1)}(x_j)\right) + \left(\sum_{i=1}^{N(t)} h_1^{(2)}(x_i)\right)\left(\sum_{j=1}^{N(t)} h_2^{(2)}(x_j)\right) = \lambda_1^{(1)}\lambda_2^{(1)} + \lambda_1^{(2)}\lambda_2^{(2)}, \quad (48)$$

where

$$\lambda_l^{(k)} := \sum_{i=1}^{N(t)} h_l^{(k)}(x_i), \quad k, l = 1, 2. \quad (49)$$

In this way an upper bound for the coagulation rate can be calculated without considering the $\mathcal{O}\left(N(t)^2\right)$ possible coagulation pairs individually.

With probability

$$\frac{\lambda_1^{(1)}\lambda_2^{(1)}}{\lambda_1^{(1)}\lambda_2^{(1)} + \lambda_1^{(2)}\lambda_2^{(2)}} \quad (50)$$

the index, i and j, of the first and second coagulating partners respectively are chosen according to the repective probabilities

$$\frac{h_1^{(1)}(x_i)}{\lambda_1^{(1)}} \quad \text{and} \quad \frac{h_2^{(1)}(x_j)}{\lambda_2^{(1)}}. \quad (51)$$

Otherwise i and j are chosen according to

$$\frac{h_1^{(2)}(x_i)}{\lambda_1^{(2)}} \quad \text{and} \quad \frac{h_2^{(2)}(x_j)}{\lambda_2^{(2)}} \quad (52)$$

respectively.

This results in coagulation events being generated according to the majorant rate $h_1^{(1)}(x_i)\, h_2^{(1)}(x_j) + h_1^{(2)}(x_i)\, h_2^{(2)}(x_j)$, which is reduced to the correct rate by treating an event as fictitious with probability

$$\frac{\tilde{K}(x_i, x_j)}{h_1^{(1)}(x_i)\, h_2^{(1)}(x_j) + h_1^{(2)}(x_i)\, h_2^{(2)}(x_j)}. \quad (53)$$

In a fictitious event, time is advanced, but the particle population is not altered.

3.3.2. Binary tree data structure

Computational performance then depends heavily on the speed at which the $\lambda_l^{(k)}$ can be calculated and the associated probabiliy distributions sampled. Binary trees provide a way to carry out all these operations in $\mathcal{O}(\log N(t)) \sim \mathcal{O}(\log n)$ per jump and since the total jump rate is $\mathcal{O}(n)$ this gives a computational complexity of $\mathcal{O}(n \log n)$; memory usage is $\mathcal{O}(n)$.

This is achieved by means of stratified caches—binary trees—of partial sums of the $h_l^{(k)}(x_i)$ (one tree for each pair l, k; although for software purposes it may be convenient to use a vector valued tree). The lowest level of a tree is initialised with the pairwise sums

$$h_l^{(k)}(x_{2i}) + h_l^{(k)}(x_{2i+1}) \tag{54}$$

resulting in a list of length $N(t)/2$ (additional 0 elements are inserted as necessary). The next level of the tree is defined in the same way on this new list and the tree is constructed recursively until, after $\lceil \log_2 N(t) \rceil$ levels, a list of length one is produced, which contains the value $\lambda_l^{(k)}$.

Such a structure has the property that each particle contributes to precisely one value on each of the $\lceil \log_2 N(t) \rceil$ levels of the tree Therefore, when one particle changes as a result of a jump event, the tree can be brought back into a correct state in $\mathcal{O}(\lceil \log_2 N(t) \rceil) \sim \mathcal{O}(\log n)$ time.

To choose a particle according to the weights $h_l^{(k)}(x_i)$ one starts from the single entry top list and descends recursively. At each level one must choose one of the two entries on the next level down that sum to make the currently chosen entry, these are known as the children. The choice between the two children is made at random with probabilities proportional to their values. Since one choice has to be made on each level this also has a cost that is $\mathcal{O}(\log n)$.

A more detailed discussion of binary trees and their use in this context is given in [22, pp46–50]. They are used in exactly the same way to select particles for surface reaction events.

3.3.3. Additional numerical details

For the tests in Sections 4.2 and 4.4 the LPDA [20] was used to accelerate the simulation of the surface processes. In LPDA some or all of the surface reaction processes represented by the operators $\mathcal{S}_l$ are removed from the main simulation, that, is they are deferred. When a particle is selected for a non-deferred event, deferred events are simulated and the updated particle is used to calculate the numerator of (53). Provided the majorants are large enough the eventual solution is unaffected.

In applications the rate functions may depend on external factors like temperature that vary with time. In this paper, time-dependent jump rates $I(t,dx)$, $\beta_l(t,x)$, $K(t,x,y)$ are approximated by piecewise constant functions. More precisely, the length of time $\Delta t_{\max}$ is determined, over which the rates can be approximated as constant. The process is stopped after $\Delta t_{\max}$ and the rates are adjusted.

Note that, in the case of inception jumps, a time-dependent rate I can be handled either by adjusting the measure $\tilde{I}$ or the function w (cf. (14)):

- If the weights w of the incepted particles are kept constant, then the inception rate (12) of numerical particles increases with the physical inception rate I.
- If the numerical inception rate $\tilde{I}$ is kept constant, then the weights of the incepted particles increase with an increasing physical inception rate I.

4. Numerical Tests

In this section two of the stochastic weighted algorithms introduced in Section 3 are tested. They correspond to the weight transfer functions (29) and (30), and are denoted by SWA1 and SWA2 (abbreviated w1 and w2), respectively. For comparison, the direct simulation algorithm (with particle doubling[15, 16]) is used and denoted by DSA.

The numerical examples are taken from the area of soot kinetics in premixed laminar flames described in Section 2.2. The exact kinetic expressions for various rate functions are those used in previous work [21]. A more detailed chemical discussion is found in [1, 29] from where the information was originally taken. The coagulation rate increases with temperature because the particles move faster as the surrounding gas gets hotter. Surface reactions and inception are affected by changes in temperature and reactant concentration. These all change due to the progress of the chemical reactions as one moves through the flame. Time dependence of the rates is handled as described in Section 3.3.3.

No type-dependence of the inception rate is relevant in this work, because the model systems have only one value of type for which this rate is non-zero. Since the initial particle concentration is zero, the inception mechanism (cf. Section 3.1) is the only way to control the computational cost–accuracy trade off. Large values of the numerical parameter n lead to high quality results at high computational cost; lower values of n lead to faster computation of lower quality results (quality refers to both systematic error and statistical noise).

4.1. Initial Validation

Validation of the stochastic weighted algorithms began using problems from [21], for which the entire particle size distribution can be calculated using an ODE solver. Direct solution of the population balance equations is possible because all but the first few thousand size classes can be neglected. Initial tests simulated only inception and coagulation processes, specifically the conditions were

1 physical particle inception rate $I_t = 2.63 \times 10^{13} \times \left(\frac{0.05-t}{0.05}\right)^2$ cm^{-3} s^{-1},
2 constant temperature of 500 K
3 constant pressure of 600 bar.

The particle size distributions showed good agreement between the ODE solver results and the stochastic weighted particle methods.

The statistical noise associated with the methods was also considered. The 95% confidence interval sizes were calculated from a central limit theorem estimate based on 30 realisations of the Markov chain for each simulation method, each realisation used just under 2^{16} computational particles. The results indicated that SWA2 is generally more noisy than SWA1. The data is an initial indication that SWA1 is to be preferred to SWA2 since fewer realisations would be needed to obtain the same size of confidence interval.

Further comparisons to the particle size distribution produced by the ODE solver were performed for a test problem including a surface reaction (pyrene condensation). Good agreement was found between the stochastic weighted particle methods and the deterministic solution of the size distribution for the limited case for which this was possible.

4.2. Accuracy for Full Sooting Flames

Having established that the algorithms and their implementations worked correctly, in limited cases for which the population balance equations could be solved directly, testing moved on to premixed laminar flames. The first flame based test compared the accuracy of the moments of the soot particle size distribution calculated by SWA1 and SWA2 for the flame JW10.68 [12]. The second moment of the mass distribution is shown in figure 1. The 95% confidence intervals for the DSA and SWA1 data are within $\pm 2\%$ of the plotted values and so confidence intervals are only shown for SWA2. The calculations for the weighted algorithms were performed with 30 runs using around 2000 computational particles from the end of the inception peak until the end of each simulation. The large difference in the statistical variability generated by the two weighting methods should be observed. SWA2 leads to a variance for the second moment of the size distribution that is more than 10 times larger than that obtained with SWA1. The situation with the zeroth and first moments is similar.

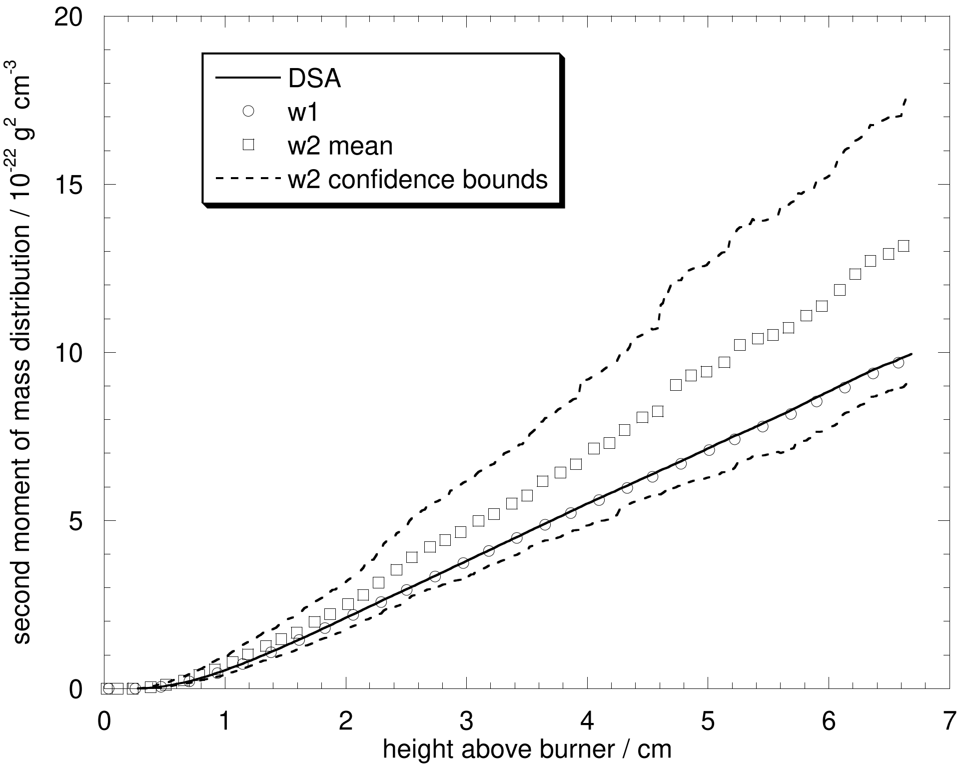

Abbildung 1: Second moment of JW10.68 mass distribution

The real attraction of stochastic particle methods is that they provide an explicit estimate of the particle distribution. As a test case for the distribution the flame JW1.69 [12] was used. This flame is known to have a bimodal particle size distribution [3] and therefore to present an interesting test case for the way in which SWA1 transfers computational effort to larger particle sizes. SWA2 was not used for this comparison, because the results above show that far more realisations would be required than for SWA1 in order to achieve the same precision.

Densities were calculated using the statistical computation package R [23] by performing Gaussian blurring of the observations with a bandwidth of 0.0245. The densities presented here were calculated in logarithmic size space, that is, they are (estimates of)

$$\frac{\mathrm{d}}{\mathrm{dlog}_{10} x} N\left(\log_{10} x\right) \tag{55}$$

where $N\left(\log_{10} x\right)$ is the number of particles per cubic centimetre comprising no more than x carbon atoms. Data calculated from 50 repetitions of SWA1 with just under 2^{13} computational particles are

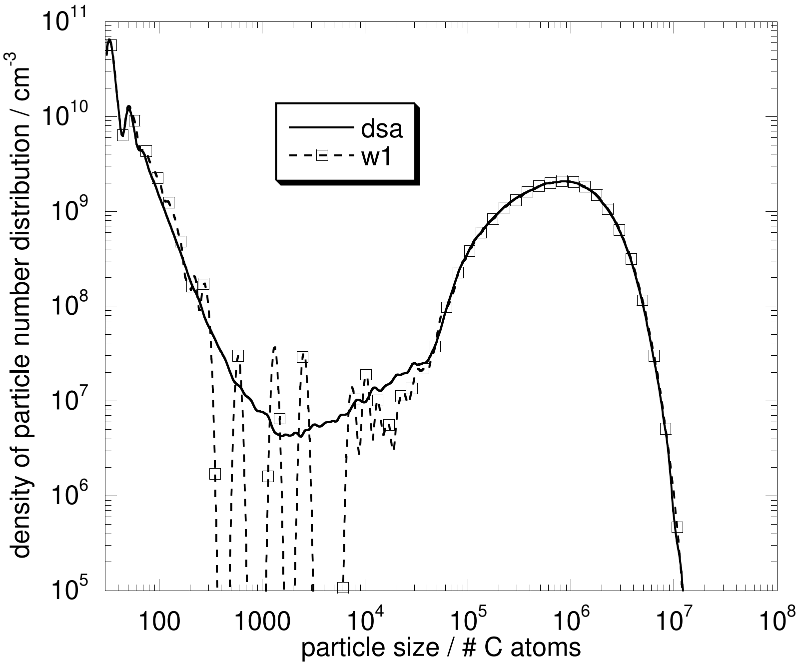

Abbildung 2: Physical particle size distribution for JW1.69

compared to data from high precision DSA calculations which used 30 repetitions with between 2^{15} and 2^{16} computational particles. The results in figure 2 show a very high degree of agreement between the two algorithms. These particular data apply to the top of the flame—about 4.2 cm above the burner.

In figure 2 large oscillations in the density generated from the SWA1 data can be seen for particle sizes between 300 and 10,000 carbon atoms. These oscillations are a symptom of the way the weighted algorithm transfers computational resolution to large particle sizes (cf. comments related to figure 3). The weighted method clearly would be rather unsuitable for this flame if the number of particles containing 300–10,000 carbon atoms was the main quantity of interest. In such a case DSA should be used. However, as will be seen for other measures of solution accuracy, the weighted method can offer as good or better performance than the un-weighted alternative.

It is also interesting to look at the distribution of computational particles on the size spectrum since the number of particles is what controls the precision of the calculation. In figure 3 the normalised densities of the computational particle distribution for the calculations used for figure 2 are plotted. The normalisation ensures that the area under both the SWA1 and the DSA curve is 1 (when integrated against $\mathrm{d}\,(\log_{10} x)$) and so there are no effects due to the different numbers of computational particles used with the two algorithms. Figure 3 gives a very clear view of the way in which SWA1 and DSA

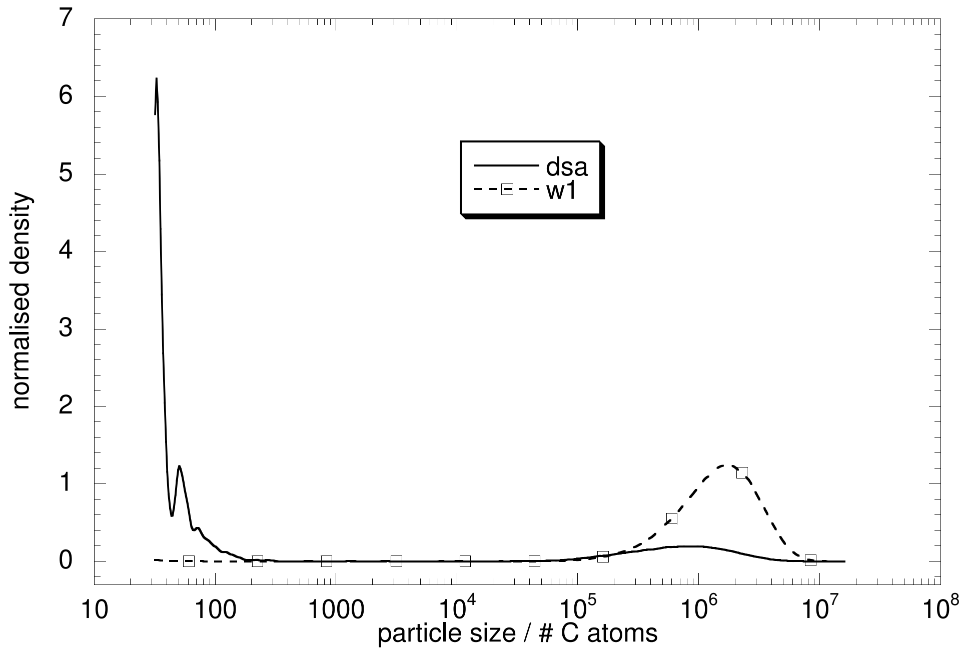

Abbildung 3: Relative computational particle distribution for JW1.69

concentrate computational effort on different parts of the size range. This shows that the choice of algorithm will depend, to some extent, on the problem that is being solved. Problems that mainly concern the largest particles are likely to be addressed best using a weighted algorithm, problems concerning the smallest particles should be tackled with a DSA. The remainder of this section attempts to investigate this choice in a quantitative way.

4.3. Coagulation Performance

Simulations of the flame HWA3 [32] were performed using DSA and SWA1. The first set of tests reported ignored the acetylene, OH and O_2 surface reactions since simulation of these processes takes a significant amount of time and is the same whether or not weighted particles are used. The results obtained in this way give no information about the soot produced by the flame but provide a comparison that should focus a little more on the properties of the weighted algorithms for coagulation. Simulation size is described by the maximum number of computational particles in the simulation.

Table 1 summarises performance on the simplified flame; run times were measured on the same desktop PC which has a 2 GHz Athlon XP CPU (2400+). The memory requirement of the simulations are low, only a few MB are required, even for the largest simulations. In table 1 the standard deviation

Tabelle 1: Variability of algorithms for reduced HWA3

tree size	method	time per run / s	population std. dev. /%				
			m0	m1	m2	m3	5–6$\times 10^3$
2^{12}	SWA1	0.6	9.7	6.4	15.0	25.4	11.1
2^{14}	SWA1	2.5	4.4	3.6	7.7	12.0	6.5
2^{16}	SWA1	10.4	3.0	1.8	3.9	6.0	2.9
2^{12}	DSA	0.5	5.4	4.2	14.6	34.9	24.4
2^{14}	DSA	1.9	2.7	2.3	7.6	19.5	11.7
2^{16}	DSA	7.6	1.5	1.1	4.0	9.6	7.4

of the population of samples for certain functionals of the solution are given for 1.34 cm above the burner, which is approximately the end of the flame. The functionals used are the zeroth, first, second, third moments of the mass distribution (denoted m0, m1, m2, m3 respectively) and the number of particles containing 5000–6000 carbon atoms.

From table 1 one sees that, for a given tree size, DSA simulations take around 75% of the computational time. For the zeroth moment DSA yields an estimate that is only half as variable as the SWA1 approach. However, the advantage of DSA drops as one moves to higher moments and by the third moment DSA is significantly inferior to SWA1. The value of the higher moments is primarily determined by the larger particles in the distribution and it is not surprising the SWA1 offers an advantage in this case since it increases the computational resolution for this part of the distribution. The variance in the estimates of the number of particles containing 5000–6000 carbon atoms is an even more extreme example of the way in which SWA1 resolves the larger sizes better (at the expense of the smaller ones) compared with DSA.

For all the functionals both algorithms appeared to have converged in mean to the true value for the largest tree sizes reported in table 1. This was verified by performing larger simulations on other hardware, which were not timed, and so are not reported in detail. These results suggest that, for some functionals, which heavily emphasize the distribution of larger particles SWA1 offers a faster way of getting good estimates than DSA.

4.4. Performance on Full Sooting Flames

Further tests were carried out, with the same flame, HWA3 [32], but including all reactions on the surfaces of soot particles. A couple of results for SWA2 are included for interest but fit the pattern discussed above and will not receive any further attention. Stochastic simulation of this flame is of considerable interest because measured particle size distributions have been published in [32]. Computational times are not comparable to those from table 1 because different hardware was used. The simulations with the full flame model take much longer because of the high rates of surface reactions and so Opteron 252 processors running at 2.6 GHz in 64 bit mode were used for the computations. The results are summarised in table 2.

Tabelle 2: Variability of algorithms for physical system

tree size	method	time per run / s	population std. dev. /%				
			m0	m1	m2	m3	9–10$\times 10^5$
2^{12}	SWA1	4.4	9.6	1.9	3.8	6.2	18.9
2^{14}	SWA1	18.0	5.2	1.1	2.3	3.6	9.8
2^{16}	SWA1	73.7	2.4	0.6	1.3	2.1	4.9
2^{18}	SWA1	274	1.1	0.2	0.4	0.7	2.1
2^{14}	SWA2	18.9	5.3	3.3	2.3	7.5	36.1
2^{16}	SWA2	75.1	2.2	1.5	3.4	5.7	16.8
2^{12}	DSA	3.0	4.6	1.1	2.9	7.1	39.1
2^{14}	DSA	12.2	3.0	0.5	1.4	3.3	19.6
2^{16}	DSA	49.5	1.3	0.3	0.6	1.8	9.3
2^{18}	DSA	213	0.7	0.1	0.4	1.0	5.3

In common with the results for the simplified problem DSA is seen to be around one third faster than the SWA1 approach for a given tree size. All the sets of simulations produced reasonably accurate estimates of the quantities considered in table 2: For the moments, the mean from 80 repetitions with each tree size was within 1% of the values from extremely high precision calculations. For the number of particles containing 9–10$\times 10^5$ C atoms, the difference between the mean and the high precision solution just reached 4% in some cases, which is not statistically significant. As in the simplified case DSA generates less statistically noisy estimates for the first few moments of the size distribution but SWA1 becomes more attractive for functionals that place a greater stress on the largest sizes of particles. However, for the reduced case, SWA1 was significantly less noisy for the third moment (m3) than DSA for a given tree size, but for the full flame the crossover is only just beginning at m3. It can be seen that for the number of particles in the size range 9–10$\times 10^5$ the SWA1 algorithm produces an estimate with roughly the same variance as the DSA with 4 times the number of particles. Examination of the 'time per run' column of table 2 shows that SWA1 can therefore provide an estimate, of any given precision, of the number of particles in the size range 9–10$\times 10^5$ in roughly one third of the time of DSA. This column also illustrates the marginally superlinear, theoretically $\mathcal{O}\left(N \log N\right)$ scaling of the computational times in the number of computational particles, with the exception of the anomalous value for SWA1 with tree size 2^{18}.

5. Conclusions and Outlook

A class of stochastic algorithms for the numerical treatment of population balance equations has been introduced. The algorithms are based on systems of weighted particles, in which coagulation events are modelled by a weight transfer that keeps the number of computational particles constant. The weighting mechanisms are designed in such a way that physical processes changing individual particles (such as growth, or other surface reactions) can be conveniently treated by the algorithms.

Numerical experiments have been performed for complex laminar premixed flame systems. Two members of the class of stochastic weighted particle methods were compared to each other and to a direct simulation algorithm. One weighted algorithm was shown to be consistently better than the other with respect to the statistical noise generated. Finally, run times to achieve fixed error tolerances for a real flame system have been measured and the better weighted algorithm has been found to be up to three times faster than the direct simulation algorithm.

Properties shared by all members of the new class

All members of the new class of stochastic weighted particle methods keep the number of computational particles constant when simulating coagulation events. This is one of the distinctive features compared to direct simulation methods, which imitate the physical process of coagulation forming a new particle by joining together two existing particles, thus reducing the particle number concentration. Because the number of computational particles does not change, there is better control of the accuracy, which strongly depends on this number. In addition, a constant number of computational particles is rather convenient from a computational point of view (storage of the vector of all particles).

All members of the class are consistent in the sense that they converge (in probability) to the solution of the corresponding population balance equation. They are also conservative “on average”. The class contains several particular cases previously studied in the literature, thus unifying different approaches.

Optimisation within the new algorithm class

Beside the advantage in controlling the number of computational particles, there is another promising feature of the new class of stochastic weighted particle methods. Some members of the class provide a better resolution (improved accuracy and reduced variance) of higher moments and tails of the size distribution. Due to the weight transfer mechanisms, the algorithms effectively redistribute computational particles over the size space, placing more computational particles at larger particle sizes and fewer at smaller sizes as compared with the constant weighting direct simulation method. Consequently the high end of the particle size distribution is calculated more accurately at the expense of the low end.

The class of weighted algorithms is rather large and the problem of optimal choice within this class was not addressed in this work. The particular algorithm “SWA1”, that was used in the numerical tests, does not seem to have appeared in the literature before. This algorithm worked quite well, though it conserves mass during coagulation jumps only “on average” and not “pathwise”. This seems to be mainly due to the fact that inception and surface reactions occur at high rates compared to coagulation. Inception and surface reactions do not conserve mass. When coagulation is the dominating mechanism, the “pathwise” mass conserving choice of the weight transfer function should be preferable. The “optimal choice” of an algorithm from the new class may also depend on the specific application. It would be interesting to find out if other elements of the class work well in specific applications, e.g., when extremely high tails of the size distribution are of importance.

High dimensions

In the original "mass flow algorithm", the weight of a computational particle was implicitly determined as the inverse of its mass. The motivation for this work was to remove this restriction, while keeping the other specific features, in order to extend the applicability of the algorithm to physical processes that change individual particles (such as growth, or other surface reactions). The resulting class of algorithms is applicable to type spaces of arbitrary dimensions. This is of particular importance, since stochastic particle methods are used for two main reasons:

- Computational complexity for this type of algorithm grows no faster than linearly with the dimension of the type space;
- As has be shown in Section 3.3.2 and Table 2, complexity of $\mathcal{O}\left(N \log N\right)$ in the number of computational particles is practical.
- Processes within individual particles can be simulated, not just overall average effects. The internal structure of particles is important in many applications.

The new class of algorithms considerably extends the choice of stochastic particle methods. Potential applications with a high-dimensional type space are, for example, recent sophisticated soot models [4].

Spatially inhomogeneous systems

The extension of the stochastic weighted particle methods to spatially inhomogeneous systems is straightforward. Particle populations are simulated on a grid of spatial cells. For problems in rarefied gas dynamics, the ability to explicitly control the weighting has been shown to be important to capture effects in regions with low particle densities [11], when a small proportion of the population has very important effects [25]. Stochastic weighted particle methods would therefore seem attractive for simulations of spatially resolved coagulating systems [9]. Explicit weights offer options for the implementation of particle transport between cells by accounting automatically for differences in the statistical weight assigned to computational particles in different cells and facilitating conservative resampling of particle populations [27]. It is also possible to exploit weighting to adjust computational resolution independently of the main kinetic simulation process by resampling the computational particle population.

Acknowledgements

RP would like to acknowledge support from the EPSRC under EP-C547241-1 and from the Trinity College, Cambridge. The underlying computational work was carried out at the Department of Chemical Engineering in the University of Cambridge.

Literatur

[1] J Appel, H Bockhorn, and M Frenklach. Kinetic modeling of soot formation with detailed chemistry and physics: Laminar premixed flames of C_2 hydrocarbons. *Combust. Flame*, 121:122–136, 2000.

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

[3] M Balthasar and M Kraft. A stochastic approach to solve the particle size distribution function of soot particles in laminar premixed flames. *Combust. Flame*, 133:289–298, 2003.

[4] Matthew S. Celnik, Markus Sander, Abhijeet Raj, Richard H. West, and Markus Kraft. Modelling soot formation in a premixed flame using an aromatic-site soot model and an improved oxidation rate. *Proc. Combus. Inst*, 32(1):639 – 646, 2009.

[5] E Debry, B Sportisse, and B Jourdain. A stochastic approach for the numerical simulation of the general dynamics equation for aerosols. *J. Comput. Phys.*, 184:649–669, 2003.

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

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

[8] A. Eibeck and W. Wagner. Stochastic interacting particle systems and nonlinear kinetic equations. *Ann. Appl. Probab.*, 13(3):845–889, 2003.

[9] Flavius Guiaş. A stochastic numerical method for diffusion equations and applications to spatially inhomogeneous coagulation processes. In Harald Niederreiter and Denis Talay, editors, *Monte Carlo and Quasi-Monte Carlo Methods 2004*, pages 147–161. Springer, 2006.

[10] Zhao Haibo, Zheng Chuguang, and Xu Minghou. Multi-Monte Carlo approach for general dynamic equation considering simultaneous particle coagulation and breakage. *Powder Tech.*, 154:164–178, 2005.

[11] Keith C Kannenberg and Iain D Boyd. Strategies for efficient particle resolution in the direct simulation Monte Carlo method. *J. Comput. Phys.*, 157:727–745, 2000.

[12] A Kazakov, H Wang, and M Frenklach. Detailed modeling of soot formation in laminar premixed ethylene flames at a pressure of 10 bar. *Combust. Flame*, 100:111–120, 1995.

[13] A Kolodko and K Sabelfeld. Stochastic particle methods for Smoluchowski coagulation equation: variance reduction and error estimations. *MCMA*, 9(4):315–339, 2003.

[14] Markus Kraft. Modelling of particulate processes. *KONA, Powder and Particle*, 23:18–35, 2005.

[15] Kurt Liffman. A direct simulation Monte-Carlo method for cluster coagulation. *J. Comput. Phys.*, 100(1):116–127, 1992.

[16] A Maisels, F E Kruis, and H Fissan. Direct simulation Monte Carlo for simultaneous nucleation, coagulation and surface growth in dispersed systems. *Chem. Eng. Sci*, 59:2231–2239, 2004.

[17] Neal Morgan, Clive Wells, Mike Goodson, Markus Kraft, and Wolfgang Wagner. A new numerical approach for the simulation of the growth of inorganic nanoparticles. *J. Comput. Phys.*, 211(2):638–658, 2006.

[18] Neal Morgan, Clive Wells, Markus Kraft, and Wolfgang Wagner. Modelling nanoparticle dynamics: coagulation, sintering, particle inception and surface growth. *Combust. Theor. Model.*, 9(3):449–461, 2005.

[19] C W Ormel and M Spaans. Monte Carlo simulation of particle interactions at high dynamic range: Advancing beyond the googol. *Astrophys. J.*, 684:1291–1309, 2008.

[20] 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(1):303–320, 2006.

[21] Robert Patterson, Jasdeep Singh, Michael Balthasar, Markus Kraft, and Wolfgang Wagner. Extending stochastic soot simulation to higher pressures. *Combust. Flame*, 145(3):638–642, 2006.

[22] Robert I A Patterson. *Numerical Modelling of Soot Formation*. PhD thesis, University of Cambridge, 2007. an electronic copy may be found at http://como.cheng.cam.ac.uk/index.php?Page=People&Section=riap2.

[23] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2005. ISBN 3-900051-07-0.

[24] S Rjasanow and W Wagner. A stochastic weighted particle method for the Boltzmann equation. *J. Comput. Phys.*, 124:243–253, 1996.

[25] S Rjasanow and W Wagner. Simulation of rare events by the stochastic weighted particle method for the Boltzmann equation. *Math. and Comput. Modelling*, 33:907–926, 2001.

[26] S. Rjasanow and W. Wagner. *Stochastic Numerics for the Boltzmann Equation*. Springer, Berlin, 2005.

[27] Alexander Vikhansky and Markus Kraft. Conservative method for the reduction of the number of particles in the Monte Carlo simulation method for kinetic equations. *J. Comput. Phys.*, 203:371–378, 2005.

[28] W. Wagner. Stochastic, analytic and numerical aspects of coagulation processes. *Math. Comput. Simulation*, 62(3-6):265–275, 2003.

[29] H Wang and M Frenklach. A detailed kinetic modeling study of aromatics formation in laminar premixed acetylene and ethylene flames. *Combust. Flame*, 110:173–221, 1997.

[30] Clive G Wells and Markus Kraft. Direct simulation and mass flow stochastic algorithms to solve a sintering-coagulation equation. *MCMA*, 11:175–197, 2005.

[31] J S Wu, Hsiao J S, Y Y Lian, and K C Tseng. Assessment of conservative weighting scheme in simulating chemical vapour deposition with trace species. *Int. J. Numer. Meth. Fluids*, 43:93–114, 2003.

[32] Bin Zhao, Zhiwei Yang, Zhigang Li, Murray V Johnston, and Hai Wang. Particle size distribution function of incipient soot in laminar premixed ethylene flames: effect of flame temperature. *Proc. Combust. Inst.*, 30:1441–1448, 2005.

[33] Haibo Zhao, F Einar Kruis, and Chuguang Zheng. A differentially weighted Monte Carlo method for two-component coagulation. *J. Comput. Phys.*, 229:6931–6945, 2010.