Cambridge Centre for Computational Chemical Engineering

University of Cambridge

Department of Chemical Engineering

Preprint ISSN 1473 – 4273

A New Numerical Approach for the Simulation of the Growth of Inorganic Nanoparticles

Neal Morgan, Clive Wells, Mike Goodson, Markus Kraft[1], and

Wolfgang Wagner[2]

submitted: 24th May 2004

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

[2] Weierstrass Institute
for Applied Analysis
and Stochastics
Mohrenstraße 39
D - 10117 Berlin
Germany
E-Mail: wagner@wias-berlin.de

Preprint No. 22

c4e

Key words and phrases. Particle Size Distribution,Laminar Premixed Flames, Nano-Particles, Stochastic Processes.

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 we derive and test an extended mass-flow type stochastic particle algorithm for simulating the growth of nanoparticles that are formed in flames and reactors. The algorithm is able to simulate coagulation that dominates such systems, along with particle formation and surface growth. We simulate three different configurations for the creation of nanoparticles. The oxidation of SiH_4 to SiO_2 and $Fe(CO)_5$ to Fe_2O_3 in a premixed $H_2/O_2/Ar$ flame were investigated under different initial concentrations of SiH_4 and $Fe(CO)_5$ respectively. In addition, the oxidation of $TiCl_4$ to TiO_2 in a tubular flow reactor was investigated. A simple reaction mechanism for the conversion of $Fe(CO)_5$ to Fe_2O_3 was suggested, based on prior experimental data along with estimated transport properties for the species considered in this system. The simulation results were compared to experimental data available in the literature.

Contents

1 Introduction

In this paper we present a solution method for population balance models which describes the synthesis and the dynamics of inorganic nanoparticles. One important route of their production is the synthesis in flames and plug flow reactors. The models which describe this process can divided into three parts: a particle source linked to the gas-phase rate of production, a surface growth term linked to the gas-phase concentration of a precursor species and the surface area of the particles, and a coagulation term whose rate is determined by a coagulation kernel.

There exist three main ways to simulate such systems: the method of moments [1], the sectional method [2] and stochastic particle methods [3, 4]. The method of moments constructs moment evolution equations for the system, which then progresses through time so giving various moments of the particle system. This method is relatively simple to implement and fast as regards computational time; however it cannot fully recover the particle size distribution (PSD) and introduces a closure problem. For the PSD we require the sectional method. This is best thought of as a finite element method. If we wish to expand the model to more than one variable, for example mass and surface area, then the sectional method takes an exceedingly long time to execute. In one experiment, the simulation took over 100 days to run [5].

Stochastic particle methods have been developed for some years. In 1972, Gillespie first used a stochastic model to simulate cloud droplet growth [6]. More recently, Eibeck and Wagner applied these ideas to coagulation and fragmentation, deriving both the direct simulation algorithm (DSA) and mass-flow algorithm (MFA) with accompanying convergence proofs and introducing fictitious jumps for the reduction of the complexity of the algorithm [7, 8, 9]. These methods have since been applied to chemical engineering by Goodson and Kraft who studied the convergence properties of the algorithm [10] and by Grosschmidt *et al.* [4] who applied the algorithm to the production of silica. A similar extension to the model, subsequently solved by a stochastic MFA has been performed by Debry *et al.* in [11]. However, there are a number of differences between their algorithm and the one used in [4]. In [11] they make use of a deterministic time step equal to the maximum value of a majorized coagulation kernel, but do not discuss the introduction of fictitious jumps. They also use operator splitting methods to simulate processes other than coagulation and make use of a bin method for particle storage and selection. Their algorithm scales with time as $N\sqrt{N}$ rather than linearly with N and demonstrated in [10].

In **this report** we introduce the extension of the MFA [12, 9] for the solution of the Smoluchowski coagulation equation with surface growth and a particle source. The algorithm presented in this paper calculates an exponentially distributed time step based on a majorant kernel and introduces fictitious jumps to compensate for the use of the majorant, uses stochastic jumps for all processes and makes use of a binary tree method for the determination of the particle partners. The algorithm presented in this paper is derived from first principles from the original equation

stated.

The algorithm is then compared to test situations and real physical systems for which experimental measurements have been obtained. The surface growth term is proportional to the surface area of the particles and accounts for the deposition of new mass onto the surface of a particle. We apply this algorithm to the production of titania. In this system, the surface reaction rate and the gas phase oxidation rate are not independent; however the algorithm is also able to simulate systems where this is not the case.

The report is organized as follows. Section 2 briefly describes solving the 1D flame equations and coupling the results with the population balance model. Section 3 describes the particle model and details the three main processes that are simulated: surface growth, particle inception and coagulation. Section 4 derives the stochastic solution method for the population balance model, and presents a description of the mass-flow algorithm for its solution. In Section 5 the performance of the mass-flow algorithm is compared to that of the direct-simulation algorithm. Section 6 introduces the three simulated particle systems and the results of those computations. Finally, Section 7 presents the conclusions of the work.

2 Coupling the Flame Simulations to the Population Balance Model

In order to simulate the 1D inhomogeneous particle system, we first solve the flame equations associated with that system. The flame equations [13] describe the evolution of the various species created and destroyed in the flame and include convection and diffusion along with other mechanisms. We solve the flame equations using the 1D flame code, PREMIX [14]. PREMIX solves the equations using a damped form of Newton's method, thus allowing various properties of the flame to be determined. For the Stochastic simulation, we require three pieces of information from the solution to the flame model: The velocity field, the temperature field and the rate of production of the particulate species along the length of the axis.

It is important to mention at this point that the population balance model is spatially homogeneous in nature. As such, what is simulated is a Lagrangian view of particles in a control volume that is convected along with the velocity field. We assume that no particles leave the control volume up or down stream and that mass can only enter the volume from the gas phase production of the desired species (from the particle source or surface growth).

In order to use the results from the PREMIX code in the stochastic coagulation code, it is necessary to convert the independent variable from a spatial coordinate to a time coordinate. It is straightforward to calculate the velocity, V, of the flow field at any point, n, using the flow rate, Q, and density, ρ:

$$V_n = \frac{Q_n}{\rho_n}. \tag{1}$$

The time, t_n, is then calculated from the average speed between points n and $n-1$ and the distance between these points:

$$t_n = 2\sum_{n=1}^{n} \frac{x_n - x_{n-1}}{V_n + V_{n-1}} \quad , \quad t_0 = 0. \tag{2}$$

This is the same as numerically integrating $1/velocity$ with respect to *distance*.

The Temperaure, T_n, and rate of production of the particle species, $\dot{\omega}_n$, can now be written in terms of time and used for the population balance model.

3 Population Balance Model

The model we construct for the growth of nanoparticles in flames contains three different processes: a particle source, surface growth and coagulation. The governing equation is given below

$$\begin{aligned}\frac{\partial}{\partial t}c(t,x) = \frac{1}{2}\sum_{y=1}^{x-1} K(x-y,y)c(t,x-y)c(t,y) \\ -\sum_{y=1}^{\infty} K(x,y)c(t,x)c(t,y) + I^{\text{inf}}c^{\text{in}}(x) \\ +I^{\text{surf}}\left[a_{x-1}c(t,x-1) - a_x c(t,x)\right],\end{aligned} \tag{3}$$

where $c(t,x)$ is the concentration of particles of size x (where x is a dimensionless volume $= v/v_0$), at time t. We also state the initial condition:

$$c(0,x) = c_0(x) \geqslant 0. \tag{4}$$

The sections which follow describe the terms in equation (3).

3.1 Particle Source

New mass is allowed to enter the system as particles produced from the gas phase. The equation that describes the time evolution of this process is shown below:

$$\frac{\partial}{\partial t}c(t,x) = I^{\text{inf}}c^{\text{in}}(x), \tag{5}$$

where c^{in} describes a source of particles of unit size:

$$c^{\text{in}}(x) = \delta(x-1) \ = \delta_1, \tag{6}$$

and the rate of production, I^{inf} comes directly from the gas phase rate of production of these monomers.

3.2 Surface Growth

Some particle systems, for example $TiCl_4 \rightarrow TiO_2$, allow particles to grow through the deposition of new mass directly onto the surface of an existing oxide particle. In certain circumstances, it is an important operation and thus should be included in a simulation [15]. The size evolution of particles due to surface growth is described by:

$$\frac{\partial}{\partial t}c(t,x) = I^{\text{surf}}\left[a_{x-1}c(t,x-1) - a_x c(t,x)\right], \tag{7}$$

where I^{surf} is the rate of surface deposition of new mass and a_x is the surface area of a particle of size x. Both parameters are described below.

In the case of TiO_2 it is possible to approximate the system into one reaction as opposed to a set of coupled reactions. The reaction of $TiCl_4 + O_2 \rightarrow TiO_2 + 4Cl_2$ has an overall observed rate such that the rate of loss of $TiCl_4$ is:

$$\frac{\mathrm{d}C}{\mathrm{d}t} = -KC,$$

where C is the concentration of $TiCl_4$ remaining in the system and K is the rate constant. The parameter K is made up of two competing mechanisms; one which reacts $TiCl_4$ with O_2 to form monomers of TiO_2, and has an associated rate constant of k_g, and another process which tries to react $TiCl_4$ on the surface of an oxide particle (and hence is termed surface growth), with an associated rate of $k_s A_D$. Here A_D is the Area Density of the system, with units of cm^2/cm^3 (i.e. the total surface area of the particles divided by the total volume of all the particles). The two rates are related through the overall rate constant thus:

$$K = k_g + k_s A_D \tag{8}$$

The total amount of $TiCl_4$ removed from the system remains the same irrespective as to whether surface growth is included or not. The reacted $TiCl_4$ is then split between surface growth and particle inception with the rates being:

$$I_{\text{inf}} = k_g C N_A\,[\#/m^3 s], \quad \text{and} \quad I_{\text{surf}} = k_s C N_A\,[\#/m^2 s] \tag{9}$$

where N_A is Avogadro's constant. Note that k_g is calculated from prior knowledge of K and k_s by rearranging equation (8).

In order to perform this algorithm we must calculate the surface area a from the volume using the expression

$$a = 4\pi\left(\frac{3m_0}{4\pi\rho}\right)^{2/D_f} x^{\frac{2}{D_f}} = \sigma x^{\frac{2}{D_f}},$$

where the fractal dimension D_f is defined by

$$\left(\frac{v}{v_0}\right)^{\frac{2}{D_f}} = \left(\frac{a}{a_0}\right). \tag{10}$$

The subscript $(_0)$ denotes the smallest particle size. Typical values for D_f range from 1.7 to 2.5 for non-spherical particles. $D_f = 3$ for spherical particles.

3.3 Coagulation Model

Coagulation of particles is modelled using Smoluchowski's coagulation equation [16] with a kernel suitable for the flames simulated:

$$\frac{\partial}{\partial t}c(t,x) = \frac{1}{2}\sum_{y=1}^{x-1} K(x-y,y)c(t,x-y)c(t,y) - \sum_{y=1}^{\infty} K(x,y)c(t,x)c(t,y). \tag{11}$$

The first term in equation (11) describes the increase in concentration of particles of size x due to the coagulation of two smaller particles whose size sum to x. The second term describes the decrease in the concentration of particles of size x that occurs whenever a particle of size x coagulates with any other particle. These two operators describe the coagulation behaviour of the system.

The rate of coagulation of any two particles is given by:

$$K(x,y) = \underbrace{\left(\frac{3}{4\pi}\right)^{\frac{2}{3}}\left(\frac{8\pi k_b T}{\rho_s}\right)^{\frac{1}{2}}\left(\frac{m_1}{\rho_s}\right)^{\frac{1}{6}}}_{\alpha}\underbrace{\left(\frac{1}{x}+\frac{1}{y}\right)^{\frac{1}{2}}\left(x^{\frac{1}{D_f}}+y^{\frac{1}{D_f}}\right)^{2}}_{\beta}, \tag{12}$$

where α can be thought of as the scaling factor of the kernel and β is the dimensionless kernel. In the scaling factor, k_b is Boltzmann's constant, T is the temperature, ρ_s is the particle density and m_1 is the mass of the smallest particle.

The Kernel used in this investigation (eq. 12) is for coagulation occurring in the free-molecular regime. This is applicable when the mean free path of a particle is considerably greater than its effective diameter.

4 Solving the Population Balance Model

4.1 Deriving the Mass-Flow Weak Form

In this section we introduce the mass-flow weak form of the extended Smoluchowski equation. From this we are able to choose the generators of the stochastic process.

We start with the integral form of equation (3):

$$\frac{\partial}{\partial t}c(t,x) = \frac{1}{2}\int_0^x K(x-y,y)c(t,x-y)c(t,y)\,\mathrm{d}y - \int_0^{\infty} K(x,y)c(t,x)c(t,y)\,\mathrm{d}y + I^{\mathrm{inf}}c^{\mathrm{in}}(x) + I^{\mathrm{surf}}\left[a_{x-1}c(t,x-1) - a_x c(t,x)\right]. \tag{13}$$

The equation is now multiplied by a continuous, compactly-supported test function, $\phi(x)$, and integrated over all x:

$$\begin{aligned}\frac{\partial}{\partial t}\int_0^\infty \phi(x)c(t,x)\,\mathrm{d}x &= \int_0^\infty \phi(x)c_0(x)\,\mathrm{d}x + \int_0^\infty \phi(x)I^{\mathrm{inf}}c^{\mathrm{in}}(x)\,\mathrm{d}x \\ &+ \int_0^\infty \phi(x)I^{\mathrm{surf}}\sigma(x-1)^{2/D_f}c(t,x-1)\,\mathrm{d}x \\ &- \int_0^\infty \phi(x)I^{\mathrm{surf}}\sigma x^{2/D_f}c(t,x)\,\mathrm{d}x \\ &+ \frac{1}{2}\int_0^\infty \phi(x)\int_0^x K(x-y,y)c(t,x-y)c(t,y)\,\mathrm{d}y\,\mathrm{d}x \\ &- \int_0^\infty \phi(x)\int_0^\infty K(x,y)c(t,x)c(t,y)\,\mathrm{d}y\,\mathrm{d}x. \end{aligned} \tag{14}$$

In order to get the equation into a form that allows the generators to be evaluated simply, certain substitutions can be made. The variable x in the first surface growth term is substituted for another variable, $y = x - 1$, yielding,

$$\int_0^\infty \phi(y+1)I^{\mathrm{surf}}\sigma y^{2/D_f}c(t,y)\,\mathrm{d}y - \int_0^\infty \phi(x)I^{\mathrm{surf}}\sigma x^{2/D_f}c(t,x)\,\mathrm{d}x. \tag{15}$$

In consequence the above equation may be written as:

$$\int_0^\infty \left[\phi(x+1)-\phi(x)\right] I^{\mathrm{surf}}\sigma x^{2/D_f}c(t,x)\,\mathrm{d}x. \tag{16}$$

Now, using the identity:

$$\int_0^\infty\int_0^\infty \Psi(x,y)\,\mathrm{d}y\,\mathrm{d}x = \int_0^\infty\int_0^x \Psi(x-y,y)\,\mathrm{d}y\,\mathrm{d}x, \tag{17}$$

we can rewrite the last two terms of equation (14) as

$$\int_0^\infty\int_0^\infty \left[\frac{1}{2}\phi(x+y)-\phi(x)\right] K(x,y)c(t,x)c(t,y)\,\mathrm{d}y\,\mathrm{d}x. \tag{18}$$

Combining equations (16) and (18) with equation (14) and allowing P to be a measure-valued solution of the equation, we obtain

$$\begin{aligned}\frac{\partial}{\partial t}\langle\phi,P(t)\rangle &= \langle\phi,P_0\rangle + \int_{\mathbb{N}} \phi(x)I^{\mathrm{inf}}P^{in}(t,\,\mathrm{d}x) \\ &+ \int_{\mathbb{N}} [\phi(x+1)-\phi(x)]I^{\mathrm{surf}}\sigma x^{2/D_f}P(t,\,\mathrm{d}x) \\ &+ \int_{\mathbb{N}^2} \left[\frac{1}{2}\phi(x+y)-\phi(x)\right] K(x,y)P(t,\,\mathrm{d}x)P(t,\,\mathrm{d}y), \end{aligned} \tag{19}$$

and

$$\langle\phi,P(t)\rangle = \int_{\mathbb{N}} \phi(x)P(t,\,\mathrm{d}x). \tag{20}$$

The total mass of the system is calculated by:

$$\int_{\mathbb{N}} xP(t,\,\mathrm{d}x). \tag{21}$$

We now call the measure,

$$Q(t,\,\mathrm{d}x) = xP(t,\,\mathrm{d}x), \quad t \geqslant 0, \tag{22}$$

the mass density and introduce a new continuous compactly-supported test function, $\psi(x)$, such that $x\psi(x) = \phi(x)$. Thus we can now consider the Mass Flow Equation:

$$\begin{aligned}\frac{\partial}{\partial t}\langle\psi, Q(t)\rangle = \langle\psi, Q_0\rangle &+ \int_{\mathbb{N}} \psi(x) I^{\inf} Q^{in}(t,\,\mathrm{d}x) \\ &+ \int_{\mathbb{N}} \left[(x+1)\psi(x+1) - x\psi(x)\right] I^{\mathrm{surf}} \sigma x^{2/D_f} \frac{Q(t,\,\mathrm{d}x)}{x} \\ &+ \int_{\mathbb{N}^2} \left[\psi(x+y) - \psi(x)\right] \frac{K(x,y)}{y} Q(t,\,\mathrm{d}x) Q(t,\,\mathrm{d}y).\end{aligned} \tag{23}$$

4.2 Deriving the Stochastic Generators

In order to solve equation (23) we introduce a sequence of measure-valued jump processes $Q^{(N)} \in \{\lambda \sum_{i=1}^{N} \delta_{x_i} \mid \lambda \in \mathbb{R},\ x_i \in \mathbb{N}\}$, such that the sequence converges in distribution to Q as $N \to \infty$. We now let

$$Q^{(N)} = \lambda^{(N)} R^{(N)}. \tag{24}$$

Accordingly set

$$R^{(N)} = \sum_{i=1}^{N} \delta x_i \quad \text{and} \quad \lambda^{(N)} = \lambda_0^{(N)} \left(\frac{N}{N-1}\right)^{\alpha}. \tag{25}$$

For notational convenience in what follows we will drop the superscript $^{(N)}$.

The measure R represents a set of stochastic particles and λ an overall scaling factor. The parameter λ_0 is calculated from the initial concentration of particles in the system $c_0(x)$ by

$$\lambda_0 = \frac{c_0(x)}{N}. \tag{26}$$

Let us define the quantity $\Psi(\alpha, R)$ as

$$\Psi(\alpha, R) = \lambda_0 \int_{\mathbb{N}} \left(\frac{N}{N-1}\right)^{\alpha} R(\,\mathrm{d}x)\psi(x) = \int_{\mathbb{N}} Q(\,\mathrm{d}x)\psi(x). \tag{27}$$

In consequence we will generate solutions to

$$\frac{\partial}{\partial t}\Psi(\alpha, R) = G_i^N \Psi(\alpha, R) + G_s^N \Psi(\alpha, R) + G_c^N \Psi(\alpha, R), \tag{28}$$

for all N, where G_i^N, G_s^N and G_c^N are the stochastic generators for particle inception, surface growth and coagulation respectively. Matching terms from equation (28) with terms in equation (23) allows us to write down the forms of the generators.

We define the coagulation generator, G_c^N to be

$$G_c^N\Psi(\alpha,R)=\lambda_0^2\left(\frac{N}{N-1}\right)^{2\alpha}\int_{\mathbb{N}^2}[\psi(x+y)-\psi(x)]\frac{K(x,y)}{y}R(\,\mathrm{d}x)R(\,\mathrm{d}y). \tag{29}$$

In order to cast this into the form of a Markov jump generator, we define

$$J_c(\alpha,R,x,y)=(\alpha,R+\delta_{x+y}-\delta_x), \tag{30}$$

and hence substituting equation (27) into equation (29) gives

$$G_c^N\Psi(\alpha,R)=\lambda_0\left(\frac{N}{N-1}\right)^{\alpha}\int_{\mathbb{N}^2}[\Psi(J_c(\alpha,R,x,y)-\Psi(\alpha,R)]\frac{K(x,y)}{y}R(\,\mathrm{d}x)R(\,\mathrm{d}y). \tag{31}$$

Likewise we define the particle inception generator, G_i^N to be

$$G_i^N\Psi(\alpha,R)=\lambda I^{\mathrm{inf}}\int_{\mathbb{N}}\psi(x)R^{\mathrm{in}}(t,\,\mathrm{d}x)=I^{\mathrm{inf}}\psi(1). \tag{32}$$

Recalling that $c^{\mathrm{in}}=\delta(x-1)$. We define

$$J_i(\alpha,R,y)=(\alpha+1,R+\delta_1-\delta_y), \tag{33}$$

thus giving

$$G_i^N\Psi(\alpha,R)=\frac{I^{\mathrm{inf}}}{\lambda_0}\left(\frac{N-1}{N}\right)^{\alpha+1}\int_{\mathbb{N}}[\Psi(J_i(\alpha,R,y))-\Psi(\alpha,R)]\frac{R(\,\mathrm{d}y)}{N}. \tag{34}$$

Finally we let the surface growth generator, G_s^N, be

$$G_s^N\Psi(\alpha,R)=I^{\mathrm{surf}}\sigma\lambda_0\left(\frac{N}{N-1}\right)^{\alpha}\int_{\mathbb{N}}x^{2/D_f}\left[\left(1+\frac{1}{x}\right)\psi(x+1)-x\psi(x)\right]R(\,\mathrm{d}x), \tag{35}$$

with jumps

$$J_{s_1}(\alpha,R,x)=(\alpha,R+\delta_{x+1}-\delta_x) \tag{36}$$

and

$$J_{s_2}(\alpha,R,x,y)=(\alpha+1,R+\delta_{x+1}-\delta_y). \tag{37}$$

It follows that the generator may be written

$$\begin{aligned}G_s^N\Psi(\alpha,R)&=I^{\mathrm{surf}}\sigma\int_{\mathbb{N}}x^{2/D_f}[\Psi(J_{s_1}(\alpha,R,x))-\Psi(\alpha,R)]R(\,\mathrm{d}x)\\&+\frac{N-1}{N}I^{\mathrm{surf}}\sigma\int_{\mathbb{N}^2}x^{2/D_f-1}[\Psi(J_{s_2}(\alpha,R,x,y))-\Psi(\alpha,R)]R(\,\mathrm{d}x)\frac{R(\,\mathrm{d}y)}{N}.\end{aligned} \tag{38}$$

4.3 An Algorithm for the Simulation of Nanoparticles with Surface Growth and Inflow (MFA)

In this section we describe the stochastic mass-flow algorithm. For a particle system $x_1, ..., x_N$ the generators (29-38) describe how and when these particles interact either with themselves or the surrounding gas phase. The principle works as follows. We wait a exponentially distributed time with parameter equal to the sum of all rates. After this time we let the particles interact. This interaction will be called jump, motivated by the underlying 'jump-process'. The nature of interaction is determined by the probability of each possible physical process, which is a function of the current state of the system. The corresponding event is chosen probabilistically according to the rates of each process.

The surface growth term is implemented as two separate possible jumps, one which changes the overall scaling and one which does not. This is required due to the way that mass flow in implemented. A majorant form of the coagulation kernel, $\hat{K}(x_i, x_j)$, will also be introduced into the algorithm [10]. This is done to change the way the double sum over all particle pairs in the coagulation generator is calculated. Instead of one double sum with complexity $\mathcal{O}(N^2)$ we instead calculate two independent sums and multiply them, thus reducing the complexity down towards $\mathcal{O}(N)$. This introduces the concept of fictitious jumps where time advances but no jump is performed. Fictitious jumps occur with probability

$$1 - \frac{K(x_i, x_j)}{\hat{K}(x_i, x_j)}, \tag{39}$$

and so care must be taken in choosing the majorant such that we reduce the number that occur.

The algorithm is as follows:

1. Generate initial state $(x_1, ..., x_N, \lambda = \lambda_0, \alpha = 0)$ and choose t_{stop}
2. Calculate the total area density, A_D from:

$$A_D = \lambda \sum_{i=1}^{N} \frac{a_i}{x_i} = \lambda\sigma \sum_{i=1}^{N} \frac{x_i^{2/D_f}}{x_i}$$

   (note that we must divide through by an extra x_i to account for the fact that we are in the mass-flow regime)
   and hence calculate k_g from:

$$k_g = K - k_s A_D.$$

3. Wait an exponentially distributed time step τ, with parameter

$$\hat{\rho}(p) = \hat{\rho}_c(p) + \rho_i(p) + \rho_{S_1}(p) + \rho_{S_2}(p)$$

$$\hat{\rho}(p) = \lambda \sum_{i=1}^{N} \sum_{j=1}^{N} \frac{\hat{K}(x_i, x_j)}{x_j} + \frac{I^{\text{inf}}}{\lambda} + I^{\text{surf}} \sum_{i=1}^{N} \sigma x_i^{2/D_f} + I^{\text{surf}} \left(\frac{N-1}{N} \right) \sum_{i=1}^{N} \sigma x_i^{2/D_f - 1}$$

and increase time according to

$$t \mapsto t + \tau$$

if $t \geqslant t_{stop}$ then stop the simulation, else go to 4.

4. With probability

$$\frac{\rho_i(p)}{\hat{\rho}(p)}$$

goto 5, else goto 6.

5. Perform a particle inception step:
   (a) Add a cluster of Size 1 to the system and remove one of size x where x is chosen uniformly from the particle array.
   (b) $\alpha \mapsto \alpha + 1$.
   (c) go to 2.

6. With probability

$$\frac{\rho_{S_1}(p)}{\hat{\rho}(p) - \rho_i(p)}$$

goto 7 else goto 8.

7. Perform a surface growth step (type 1):
   (a) Choose a particle, i, according to the distribution:

$$\frac{x_i^{2/D_f}}{\sum_{k=1}^{N} x_k^{2/D_f}}$$

   (b) Replace particle i and with a particle of size $x_i + 1$.
   (c) Go to 2.

8. With probability

$$\frac{\rho_{S_2}(p)}{\hat{\rho}(p) - \rho_i(p) - \rho_{S_1}(p)}$$

goto 9 else goto 10.

9. Perform a Surface Growth step (type 2):
   (a) Choose a particle, i, according to the distribution:

$$\frac{x_i^{2/D_f - 1}}{\sum_{k=1}^{N} x_k^{2/D_f - 1}}$$

   and a particle j uniformly.

(b) Replace particle j with a particle of size $x_i + 1$.

(c) $\alpha \mapsto \alpha + 1$.

(d) Go to 2.

10. Perform a coagulation step:

(a) Choose particles i and j according to the distribution:

$$\frac{\frac{\hat{K}(x_i,x_j)}{x_j}}{\sum_{i,j} \frac{\hat{K}(x_j,x_j)}{x_j}}, \qquad i \neq j.$$

(b) With probability

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

add a particle of size $(x_i + x_j)$ to the particle array and remove one of size x_i. Otherwise the jump is fictional and the particles have not interacted.

(c) Go to 2.

5 Numerical Investigation of the Mass-Flow Algorithm

5.1 The Test Simulation

A test case was used in order to study the convergence properties of the MFA against the DSA for speed and accuracy. The parameters for the test case were as follows:

Table 1: *Parameters for the test case.*

Name	Parameter	value
Kernel scaling factor	α	1.0
Inflow rate (inception)	I^{inf}	0.5
Inflow rate (surface)	I^{surf}	0.0
Fractal dimension	D_f	2.1
Number of particles	N	2097152
Number of runs	L	1
Initial concentration	$c_o(x)$	1.0
Length of simulation	t_{stop}	5.0
mass of monomer	m_0	1.0
volume of monomer	v_0	1.0

To study the systematic error of the algorithm, we use an approximation parameter $\zeta(t)$ which corresponds to the solution obtained using one run at the largest feasible N.

Typical macroscopic properties such as the moments of a distribution are of the form

$$F(t) = \int_0^\infty \phi(x)c(t,x)\,\mathrm{d}x, \tag{40}$$

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

$$\xi^{(N)}(t) = \frac{1}{N}\sum_{i=1}^{N}\phi(x_i(t)). \tag{41}$$

One can then estimate the expected value of the estimator $\xi^{(N)}$ over a number of L independent runs. The corresponding values of the random variable are denoted by $\xi^{(N,1)}(t), ..., \xi^{(N,L)}(t)$. The empirical mean of these being determined by

$$\eta^{(N,L)}(t) = \frac{1}{L}\sum_{l=1}^{L}\xi^{(N,l)}(t). \tag{42}$$

using these results one is now in the position to write down an estimate to the true absolute error, $e_{\rm abs}^{(N,L)}(t)$ thus:

$$\tilde{e}_{\rm abs}^{(N,L)}(t) = |\eta^{(N,L)}(t) - \zeta(t)|, \tag{43}$$

and an estimate of the percentage relative error:

$$\tilde{e}_{\rm rel}^{(N,L)}(t) = 100 \times \frac{|\eta^{(N,L)}(t) - \zeta(t)|}{\zeta(t)}. \tag{44}$$

From this an estimate of the error over the time interval $[0, T]$ can be calculated,

$$c_{\rm tot} = \frac{1}{M+1} \sum_{i=0}^{M} \tilde{e}^{(N,L)}(t_i) \tag{45}$$

where M is the number of discrete divisions over the time interval. In a similar way, an average for the percentage relative error can be found.

The main macroscopic quantity we look at in this paper are the average mass and the first and second moments of the distribution. In the mass-flow formulation, the average mass proportional to the harmonic mean of the sizes $\{x_i\}$

$$m_0 \left(\frac{1}{N} \sum_{i=1}^{N} x_i^{-1} \right)^{-1}, \tag{46}$$

the $n^{\rm th}$ mass moment is calculated by

$$\mathrm{M}n = \frac{c_0 v_0 m_0^n}{N} \left(\frac{N}{N-1} \right)^{\alpha} \sum_{i=1}^{N} x_i^{n-1}. \tag{47}$$

In this investigation, the simulation was run for eight hours at each value of N, allowing as many runs within that time as the simulation could perform. The input parameters were as in table 1 with the exception of N (which varied from 128 to 16384) and L (which varied inversely with N). In this way the product $N \times L$ was kept approximately constant.

5.2 Results of the Test Simulation

The results for the cpu times and errors in the first three moments were as follows:

One can see from table 2 and figure 1 that the CPU time for the mass flow algorithm scales linearly with N. This is in agreement with the improved direct simulation algorithm (DSA) as investigated by Goodson and Kraft in [10] and is a consequence of using a linearly majorant kernel for the coagulation calculations.

The reduction of the total error in average size (figure 2) seems to fall as $1/\sqrt{N}$ after an initial rate proportional to $1/N$. This is in contrast to the convergence

Table 2: *CPU times and errors for algorithm.*

N	t_{sr} [s]	c_{tot} Ave Size	$\%c_{tot}$ Ave Size	$\%c_{tot}$ M2	$\%c_{tot}$ M3
128	0.39425	0.079237	0.96699	0.81405	2.1103
256	0.76793	0.038381	0.48252	0.43410	1.1106
512	1.5545	0.019825	0.23995	0.22174	0.56378
1024	3.1492	0.011917	0.13249	0.12075	0.29748
2048	6.4304	0.0080560	0.081518	0.085047	0.18720
4096	13.188	0.0056460	0.052420	0.068252	0.14136
8192	29.219	0.0047430	0.043318	0.051557	0.10644
16384	61.637	0.0039410	0.033091	0.046613	0.092308

properties reported in [10] which seems to indicate that the algorithm converges as $1/N$ throughout. The discrepancy here would seem to come from the way in which the investigation was carried out. In this paper, the investigation was run for eight hours as this gave statistical errors in the solution of the same order of magnitude of those calculated in [10] (of approximately 0.07). In [10] this was sufficient to ensure that the statistical error (c_{stat}) was considerably less than the systematic error (c_{tot}). Using the mass flow algorithm however, the initial systematic error was two orders of magnitude less than if the DSA had been used. As such the statistical error was larger than the systematic error and thus the order of convergence as defined in [17] and stated in [10] could not be calculated.

Figure 3 shows the relative errors for average size and for the second and third moments of the distribution. As with the total error in average size, the errors in these quantities also decrease as $1/\sqrt{N}$.

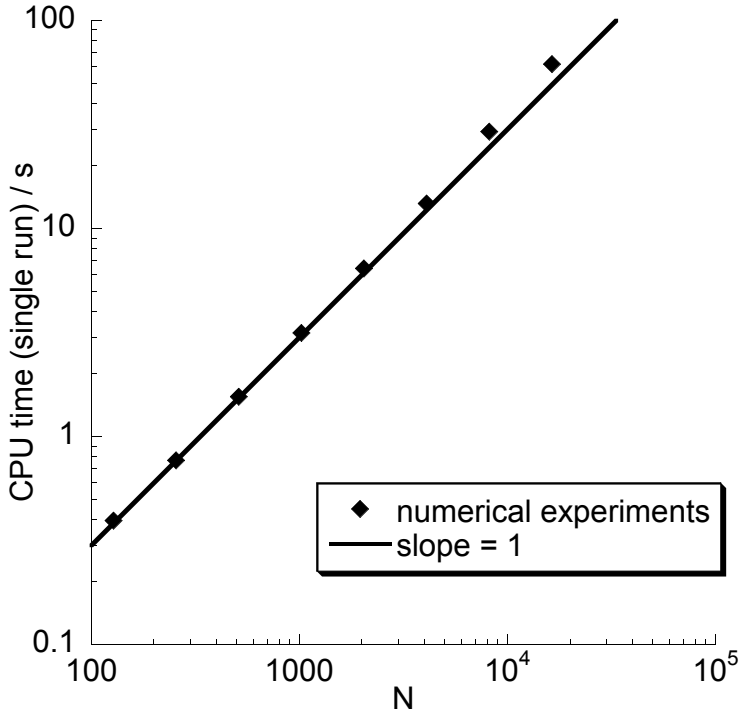


Figure 1: *CPU time for single runs at various values of N.*

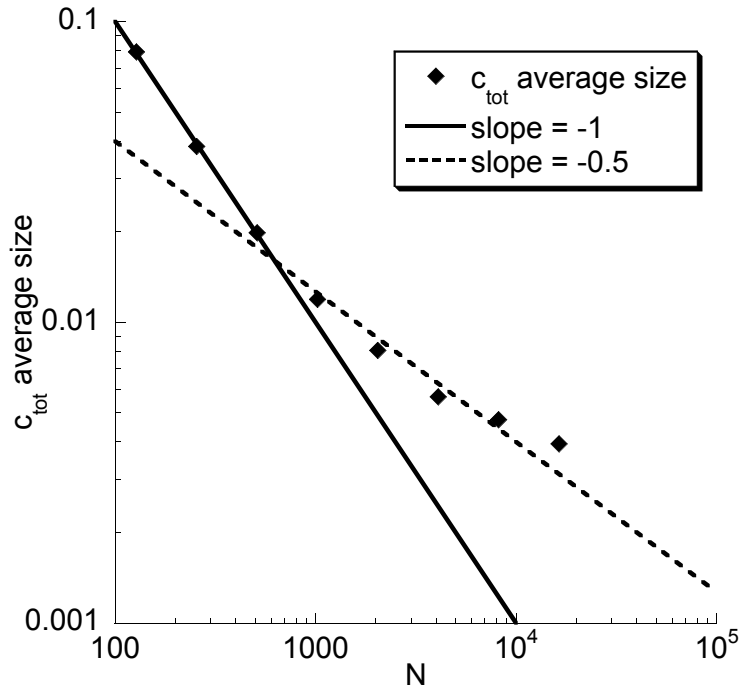


Figure 2: *Total error in average size vs N.*

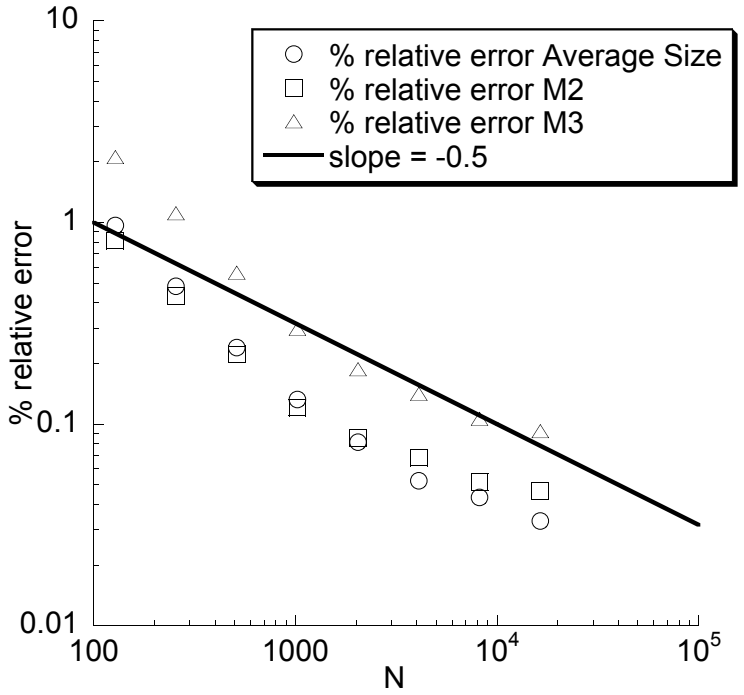


Figure 3: *Percentage relative error in average size, M2 and M3 vs N.*

6 Simulated Systems

Two of the three particle systems are simulated as low pressure $H_2/O_2/Ar$ flames doped with a precursor (SiH_4 and $Fe(CO)_5$ to generate particles of SiO_2 and Fe_2O_3, respectively). The third reaction scheme describes the oxidation of $TiCl_4$ to TiO_2 at atmospheric pressure in a simple plug-flow reactor.

6.1 Low Pressure $H_2/O_2/Ar$ Flame with SiH_4 Precursor

The SiH_4 gas phase kinetics are described by a combined mechanism, which contains 18 $H_2/O_2/Ar$ reactions plus a further 26 reactions to describe the oxidation of SiH_4 to SiO_2[18]. All thermodynamic and transport properties were found in the CHEMKIN [19] and TRANFIT [20] libraries, respectively. The flames in this section were simulated at low pressures (30 mbar) and at a velocity of 1.32 m/s.

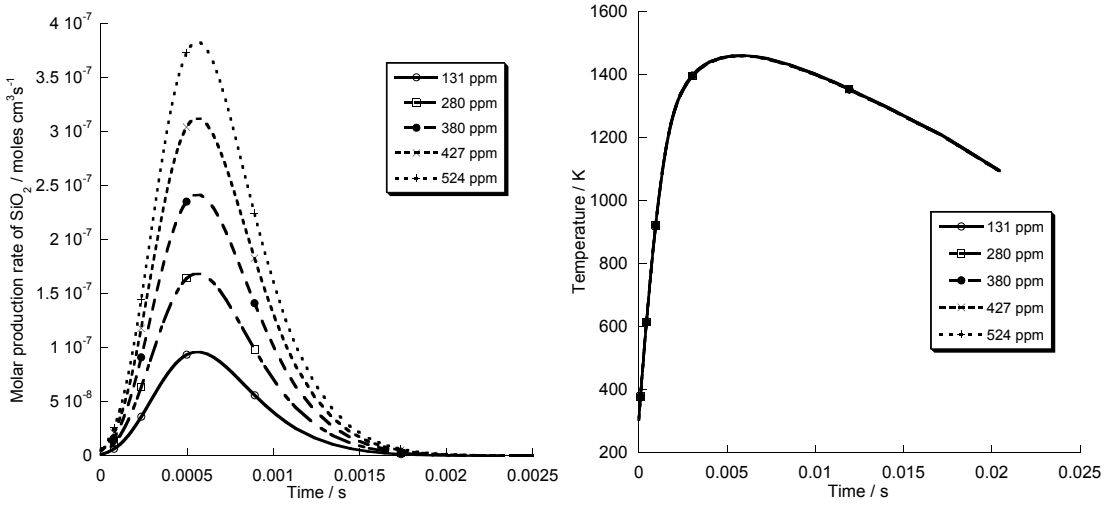
The evolution of SiO_2 particles under varying input concentrations of SiH_4 was investigated. The concentration was varied between 131 and 524 ppm with all other input parameters held constant as shown in table 3. For the coagulation simulation, the particle number N was set to 1024 and the simulation results averaged over 50 runs ($L = 50$). The fractal dimension of the silica particles, D_f, was set to be 2.23.

The CPU time for the 50 runs was 34.18 seconds during which there were on average 35000 jumps performed per run.

Figure 4(a) shows how the particle inception rate, I^{inf} evolves over time at various

Table 3: *Input parameters for* SiH_4 *to* SiO_2 *flame.*

Parameter	Value
initial flame velocity	1.32 m/s
Pressure	30 mbar
$H_2 : O_2$	1.69 mol:mol
$Ar : (H_2 + O_2)$	1.36 mol:mol
SiH_4 concentration	131 - 524 ppm

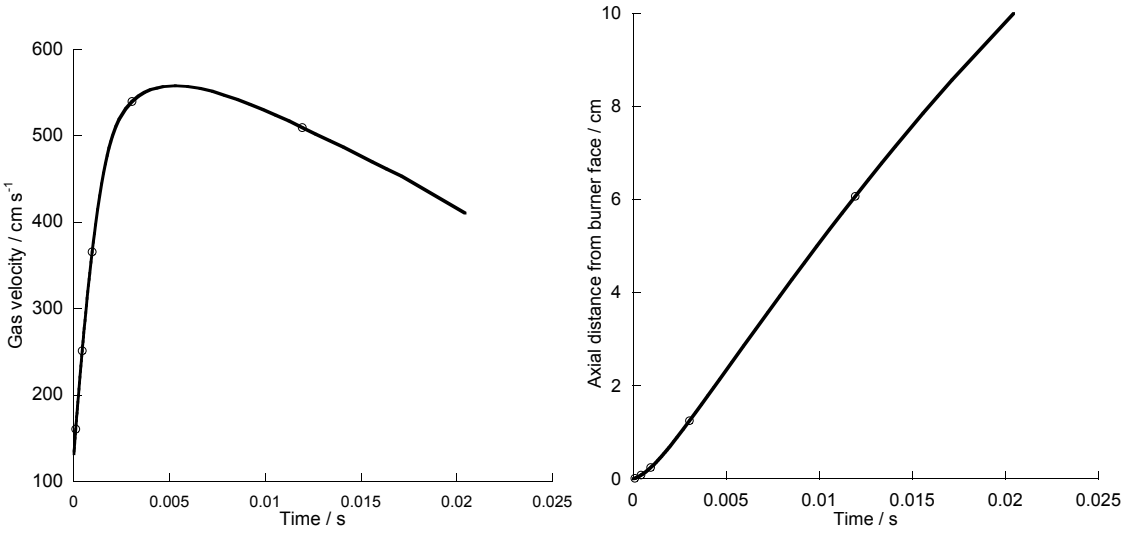


(a) *Production rate vs time* (b) *Temperature vs time*

Figure 4: *Production rates and temperature profiles for laminar flames producing* SiO_2 *with varying initial concentrations of* SiH_4.

initial concentrations of SiH_4. The important features of the plot are that the initial time for particle inception ($t = 0s$) and for the end of particle inception ($t \approx 0.002s$) remains the same regardless of the initial concentration of SiH_4. All that does change is the height of the maxima that occurs at $t = 0.00058s$.

Figure 4(b) shows that the temperature profile of the flame changes negligibly with the change in initial concentration. This indicates that the flow field (figure 5(a)) and hence the coupling between distance and time along the flame (figure 5(b)) will be unchanged also.



(a) *Velocity vs time* (b) *Distance vs time*

Figure 5: *Velocity and distance profiles for laminar flames producing* SiO_2 *with varying initial concentrations of* SiH_4.

Figure 6 shows the result of the coagulation simulation. The simulated results agree excellently with the experimental measurements of Lindackers *et al.* [18], predicting the increase in average particle mass with the initial concentration of SiH_4.

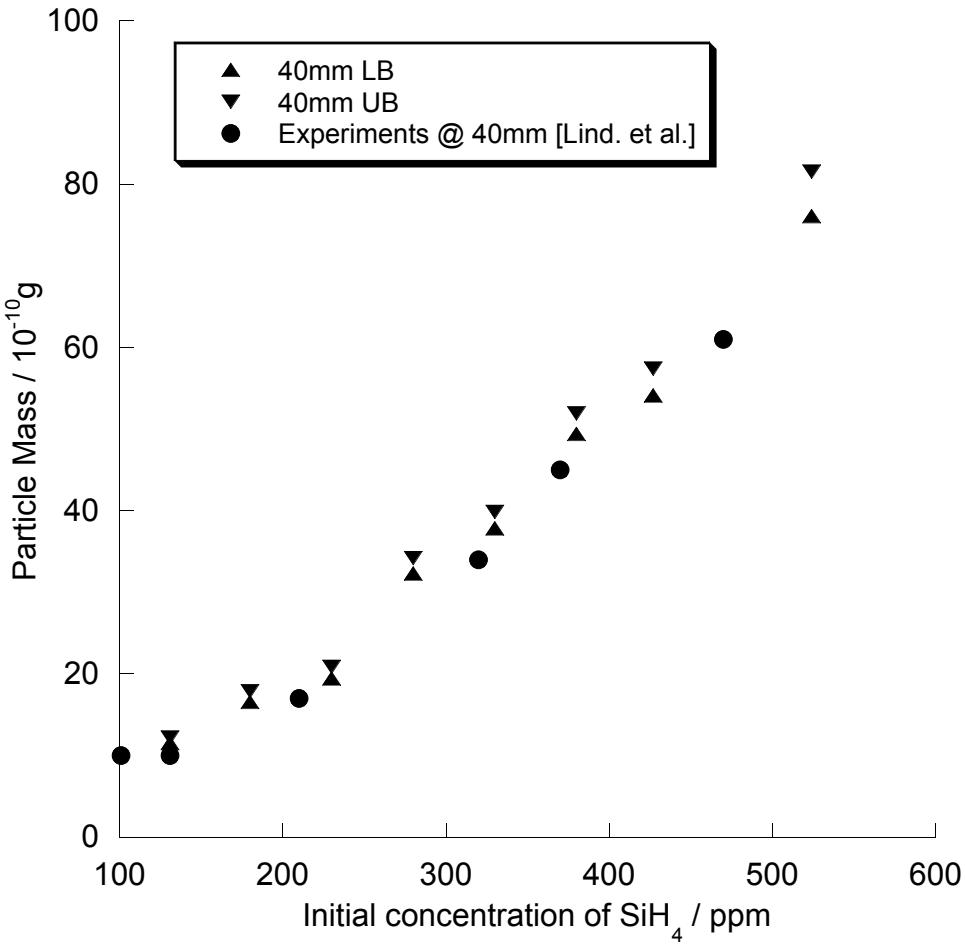


Figure 6: *Effect on average particle mass of initial concentration of* SiH4 *precursor, simulation vs experimental data. The triangles indicate the lower bound (LB) and the upper bound (UB) of the 99.9% confidence interval for the empirical mean of the particle mass.*

6.2 Low pressure $H_2/O_2/Ar$ Flame with $Fe(CO)_5$ Precursor

Previous investigations into simulating the coagulation of Fe_2O_3 particles have assumed instantaneous decomposition of the $Fe(CO)_5$ into Fe_2O_3 [21]. For this paper a simple skeletal mechanism was proposed. The following $Fe(CO)_5$ kinetics were estimated from the experiments of Giesen *et al.*[22] and Woiki *et al.*[23]:

$$Fe(CO)_5 \xrightarrow{k_1} FeCO + 4CO \tag{48}$$

$$FeCO \xrightarrow{k_2} Fe + CO \tag{49}$$

$$4Fe + 3O_2 \xrightarrow{k_3} 2Fe_2O_3 \tag{50}$$

The three rate constants were fit to a modified Arrhenius equation:

$$k_f = AT^{\beta} \exp\left(\frac{-E_A}{RT}\right), \tag{51}$$

with the coefficients for the three chemical equations shown in table 4.

Table 4: *Rate constants for the proposed* $Fe(CO)_5$ *to* Fe_2O_3 *mechanism.*

	A[cm,mol,s]	β[ ]	E_A[cal/mol]
k_1	4.00×10^9	0.0	17290.0
k_2	7.00×10^9	0.0	17290.0
k_3^*	7.00×10^{12}	0.0	0.0

Note that k_3 has been modified to k_3^* to allow for the reaction rate to be first order with respect to iron concentration and zeroth order with respect to oxygen concentration. The third reaction (eq. 50) is a much simplified form of the mechanism to account for the oxidation of Fe to Fe_2O_3. Its kinetics have been estimated in order to make that particular reaction faster than the other two reactions. We justify making this assumption as the paper by Janzen and Roth [21] assumes instant formation of Fe_2O_3 from $Fe(CO)_5$ at the start of the flame, whereas the paper by Giesen *et al.*[22] gives a finite rate for the decomposition of $Fe(CO)_5$. Since there is no further information about the rates, we chose to allow the oxidation part of the mechanism to not be the rate limiting step, hence its more rapid rate.

The transport properties of the Fe compounds were also estimated. This was done by comparing the molecules' size and shape to other species within the TRANFIT database and extrapolating/estimating values as required. The resulting TRANFIT constants are contained in table 5.

Table 5: *Estimated transport properties for* Fe *compounds.*

Species	C1	C2	C3	C4	C5	C6
Fe	0	2999.0	3.76	0.00	0.00	0.00
$Fe(CO)_5$	2	400.0	5.9	0.00	0.00	1.00
Fe(CO)	1	400.0	3.9	0.00	0.00	1.00
Fe_2O_3	2	400.0	4.5	0.00	0.00	1.00

The various constants are as follows:
C1 = Shape constant where 0 = single atom, 1 = Linear molecule, 2 = Non-linear molecule
C2. The Lennard-Jones potential well depth ϵ/k_B [K]
C3. The Lennard-Jones collision diameter σ [Å]
C4. The dipole moment μ in Debye $[10^{-18} cm^{3/2} erg^{1/2}]$
C5. The polarizability α [Å^3]
C6. The rotational relaxation collision number Z_{rot} at $298k$

The equations that use these constants can be found in the TRANFIT manual [20].

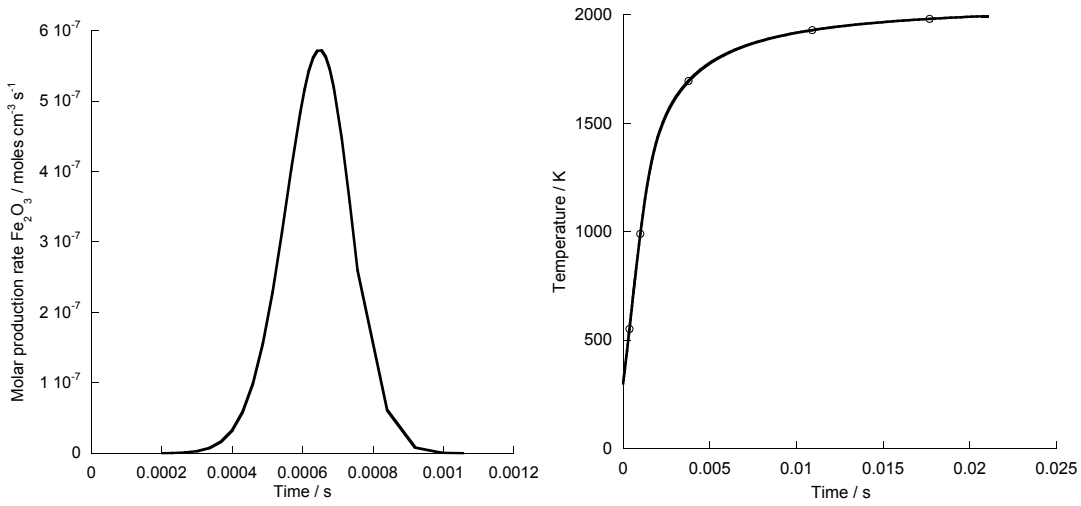
The evolution of Fe_2O_3 particles was investigated for a flame with initial conditions as shown in table 6. The coagulation simulation parameters were set to $N = 1024$, $L = 50$ and $D_f = 3.0$. The simulation took 35.28 seconds for the 50 runs.

Table 6: *Input parameters for* $Fe(CO)_5$ *to* Fe_2O_3 *flame.*

Parameter	Value
initial flame velocity	1.32 m/s
Pressure	30 mbar
$H_2 : O_2$	1.00 mol:mol
$Ar : (H_2 + O_2)$	1.04 mol:mol
$Fe(CO)_5$ concentration	524 ppm

Figures 7(a) and 7(b) show how the particle inception rate and temperature vary with time for this particular flame. Note that in [21] the total amount of Fe_2O_3 that enters the system (the integral of figure 7(a) over time) is already in the system at time $t = 0$.

Figure 8 shows the result of the coagulation simulation. The evolution of the particle diameter over time is in good agreement with the experimental data of Janzen *et al.* [21]. This result was obtained using a fractal dimension of 3.0, which implies that the particles sinter rapidly to spherical particles.



(a) *Production rate vs time* (b) *Temperature vs time*

Figure 7: *Production rates and temperature profiles for a laminar flame producing* Fe_2O_3.

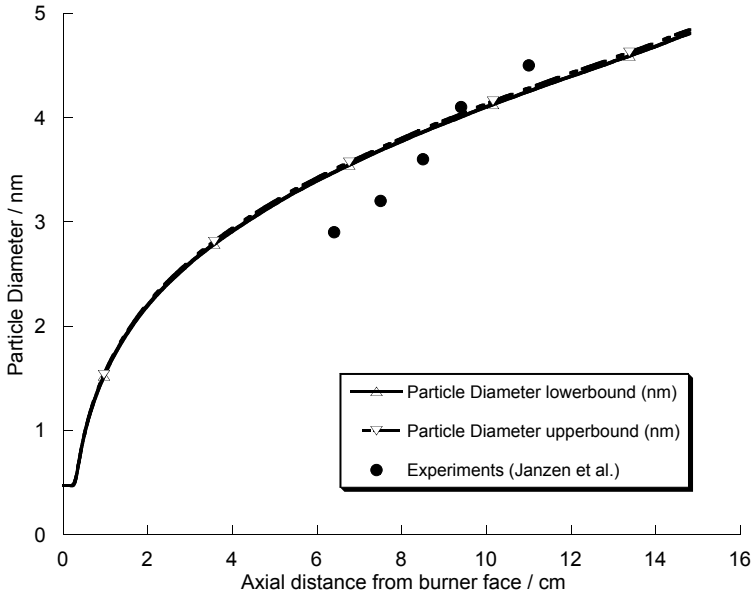


Figure 8: *Comparison of simulated particle size evolution vs experimental data. The lower bound and upper bound lines represent the limit of a 99.9% confidence interval of the empirical mean of the particle diameter.*

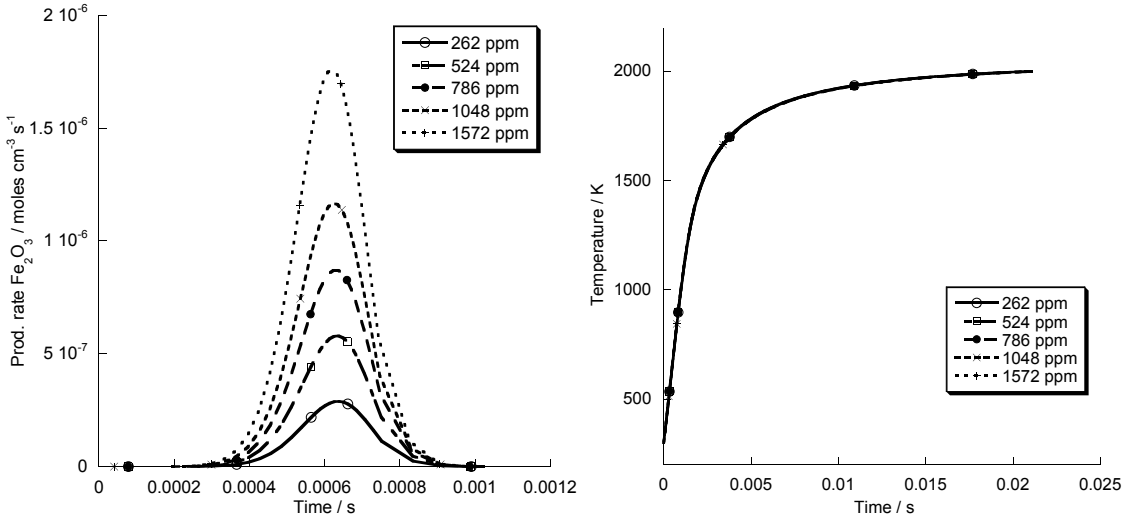
The next set of simulations involved changing the initial concentration of $Fe(CO)_5$ between 262 and 1572 ppm whilst keeping all other variables constant. The parameters of the flame were as in table 7. The simulation parameters for the coagulation were again $N = 1024$, $L = 50$ and $D_f = 3.0$. The simulations took between 30 and 50 seconds to complete.

Table 7: *Input parameters for* $Fe(CO)_5$ *to* Fe_2O_3 *flames with varying initial concentrations of* $Fe(CO)_5$.

Parameter	Value
initial flame velocity	1.32 m/s
Pressure	30 mbar
$H_2 : O_2$	1.00 mol:mol
$Ar : (H_2 + O_2)$	1.04 mol:mol
$Fe(CO)_5$ concentration	262 - 1572 ppm

In figures 9(a) and 9(b) we see how the particle inception rate and temperature evolve over time. Note that the behaviour shown for the production rate curves is similar to the silica flames simulated under varying inlet concentration. However the inception period is more rapid in this case.

Figure 10 shows the results of the coagulation simulation. The illustrations and measurements show different trends, which seems odd especially when considering the excellent agreement that was obtained for the similar silica system. It is of note that the simulations carried out in [21] also over predict the particle diameters to a similar order of magnitude. This discrepancy is put down to a lack of kinetic data being available for the oxidation of $FeCO_5$ to Fe_2O_3 along with subtleties that may lie in the coagulation. The flatness of the experimental results would indicate that there is only a slight increase in the amount of Fe_2O_3 entering the system. As such there may be some concentration-dependant limiting factor in the oxidation rate.



(a) Production Rate vs ppm

(b) Temperature vs ppm

Figure 9: *Production rates and Temperature profiles for Laminar flames producing* Fe_2O_3 *with varying initial concentrations of* $Fe(CO)_5$.

6.3 $TiCl_4 \rightarrow TiO_2$ Plug-Flow Reactor System

The final system simulated was the oxidation of $TiCl_4$ to form titania in a plug-flow reactor. Because this is a simple system, information about the kinetics of the reactions, the initial concentrations of reactants and the temperature profile of the reactor are sufficient for a simulation.

The oxidation of $TiCl_4$ to form TiO_2 is modelled as a single reaction:

$$TiCl_4 + O_2 \xrightarrow{K} TiO_2 + 2Cl_2, \tag{52}$$

where K is the sum of the gas-phase and surface-growth rates as explained in Section 3.2. The kinetic data for this reaction is taken from the paper by Spicer *et al.*[15] whilst the thermodynamic data is taken from the NIST website [24]. No transport data are required.

The system was first simulated at 1400 K, with initial concentrations of $TiCl_4$ of 10^{-4} and 10^{-6} mol/m^3. At these conditions and with 4096 stochastic particles, the simulations took little over 28 seconds per run.

Figure 11 shows how the first moment of the particle size distribution (total mass) increases over time. The rate of inception of new mass (the slope of the curve) is the same for both systems as is the point in time at which inception ceases. Figure 12 shows how the average particle mass increases over time. The extra mass speeds up the coagulation process and allows a rapid increase in average particle mass at around $t = 0.1$ s. This effect is due to the large particles in the system coagulating very rapidly with the newly incepted particles in accordance with the coagulation

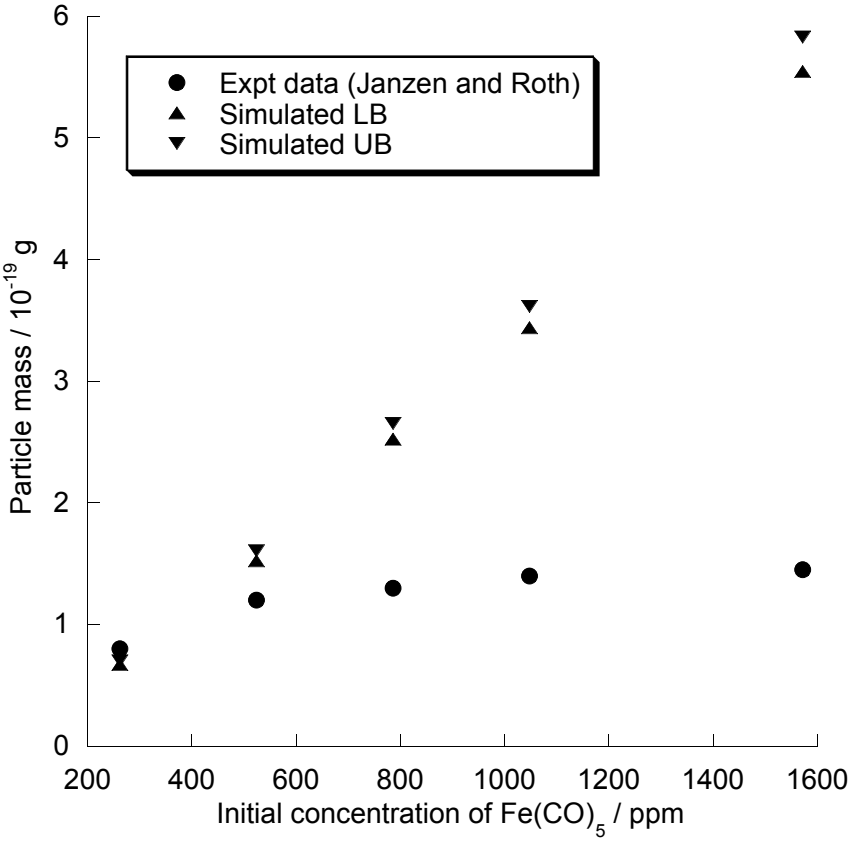


Figure 10: *Effect on average particle mass of initial concentration of* $\mathrm{Fe(CO)_5}$ *precursor, simulation vs experimental data. The triangles mark the upper bound (UB) and lower bound (LB) of a 99.9% confidence interval for the empirical mean of the particle mass.*

kernel. Once inception has ceased, the rate of the increase in particle mass becomes independent of the initial concentration of $TiCl_4$.

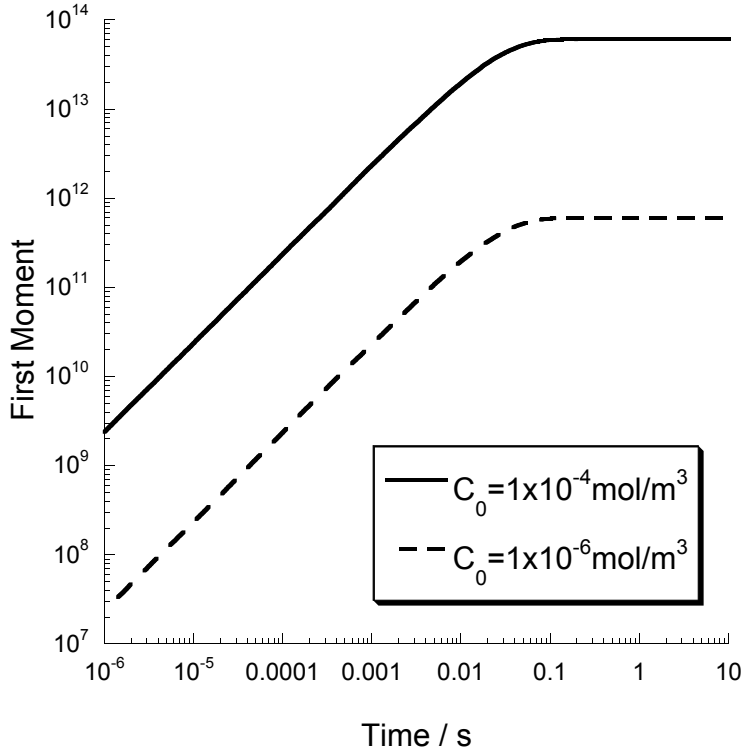


Figure 11: *M1 vs Time for* TiO_2 *system*

The titania system was next simulated at a much higher initial concentration of $TiCl_4$ so as to make surface growth more noticeable. The temperature was again 1400 K but the initial concentration of $TiCl_4$ was now 1.0 mol/m^3. Under these conditions and simulating 4096 stochastic particles, the simulations took 59.54 seconds per run when simulating surface growth and 52.60 seconds per run when not.

Figure 13 shows various properties of the $TiCl_4$ system, simulated both with and without surface growth. In figure 13(a) we see that the average mass of the particles is larger when surface growth is included in the simulation compared to when we simulate particle inception and coagulation only. Figure 13(b) shows that the same amount of mass is entering the system irrespective of whether there is surface growth or not. This shows that the addition of mass directly to a particle is more important than the increase in the concentration of particles for coagulation. Finally figure 13(c) shows the PSD of the systems at time $t = 0.001$ s. We see that when surface growth is included in the simulation, a small additional peak is formed at particle mass = $8m_0$. This corresponds to small particles that normally would not form in large numbers through coagulation being formed via surface growth.

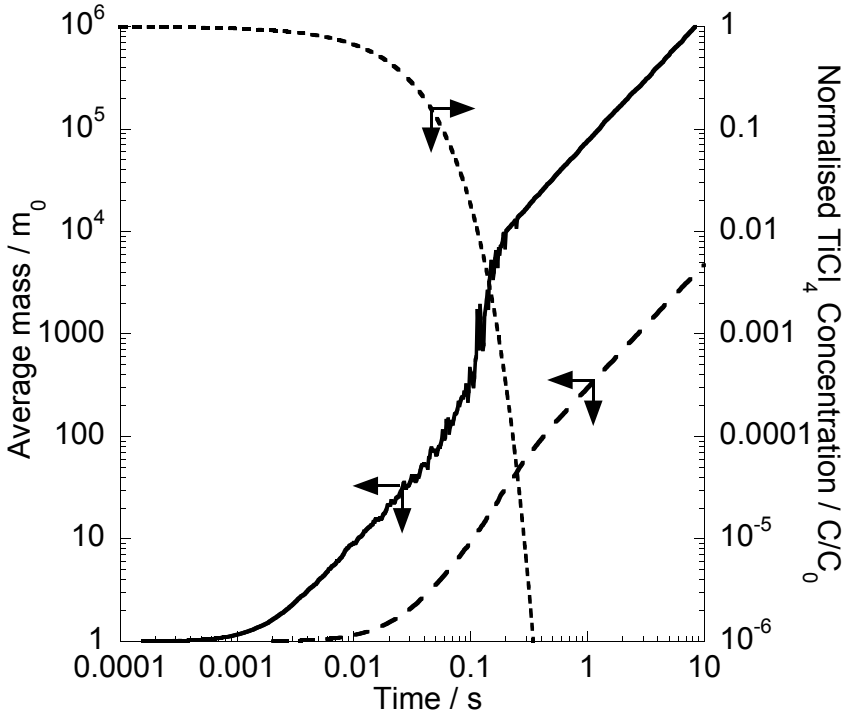


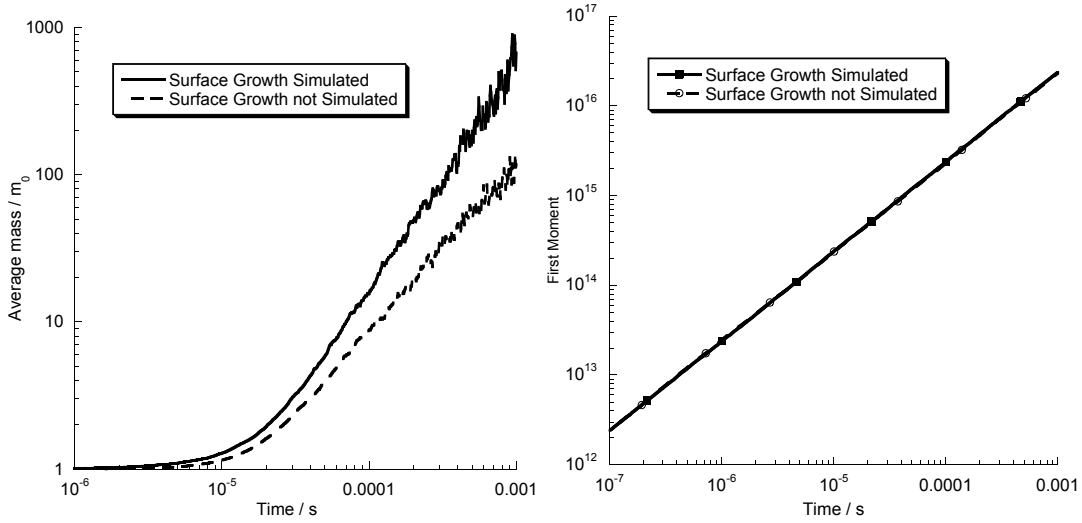
Figure 12: *Average mass of* TiO_2 *vs Time. The solid line represents an initial concentration of* $TiCl_4$ *of* $10^{-4}\,\mathrm{mol/m^3}$ *whilst the dashed line represents an initial concentration of* $10^{-6}\,\mathrm{mol/m^3}$.

7 Conclusions

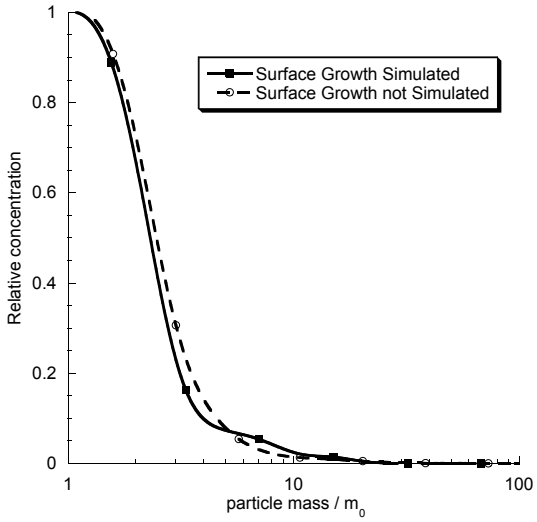
In this paper we have shown how a model for nanoparticle growth may be transformed into a weak form and hence how the generators of the stochastic process are deduced. This led us to derive a new mass-flow algorithm, which is able to simulate the coagulation, inception and surface-growth processes.

The algorithm was used to simulate a test situation for which various convergence properties could be calculated. The algorithm was found to scale linearly with the normalization parameter, N, as regards CPU time whilst the total errors in average size and second and third moments were found to decrease as $1/\sqrt{N}$. For small N, when the systematic errors dominate the solution, the error was found to decrease linearly in N. The total error in the solution for these simulations was initially two orders of magnitude less than when compared to the similar direct-simulation algorithm.

Three particle systems were simulated using the mass-flow algorithm. The silica system was simulated for increasing initial concentrations of the precursor SiH_4, with the results calculated being in excellent agreement with the experimental data obtained from the literature. Iron oxide was also simulated under varying initial concentrations of the precursor. At low concentrations, the results were in very good agreement with the experimental data obtained from the literature. At higher



(a) *Average Mass vs Time* (b) *First moment vs Time*



(c) *PSD of* TiO_2 *size at* t = 0.001s

Figure 13: *Various properties of* TiO_2 *system with and without surface growth simulated.*

concentrations however, the difference between the model and the experiments differed greatly. This has been attributed to a lack of kinetic data for the oxidation of $FeCO_5$. Finally the titania system was simulated at high and low initial concentrations of $TiCl_4$, both with and without surface growth. It was found that surface growth is most important for high initial concentrations of $TiCl_4$. When simulated with surface growth, the average mass of the system was increased and the PSD showed a secondary peak for small particles.

Acknowledgements

The authors would like to thank the EPSRC (grant number GR/R85662/01) for the financial support of Neal Morgan under the title 'Mathematical and Numerical Analysis of Coagulation-Diffusion Processes in Chemical Engineering' and the Oppenheimer Fund for the support of Clive Wells.

References

[1] M. Frenklash and S. Harris. Aerosol dynamics modelling using the method of moments. *Journal of colloid and interface science*, 118(1):252–262, 1986.

[2] S.E. Pratsinis, H. Bai, P. Biswas, M. Frenklach, and S.V.R. Mastrangelo. Kinetics of $TiCl_4$ oxidation. *Journal of the American Ceramics Society*, 73:2158–, 1990.

[3] P. Tandon and Daniel E. Rosner. Monte Carlo simulation of particle aggregation and simultaneous restructuring. *J. Colloid Interface Sci.*, 213:273, 1999.

[4] D. Grosschmidt, H. Bockhorn, M. Goodson, and M. Kraft. Two approaches to the simulation of silica particle systems. *Proc. Combust. Inst.*, 29:1039–1046, 2002.

[5] H. Mühlenweg, A. Gutsch, A. Schild, and S.E. Pratsinis. Process simulation of gas-to-particle-synthesis vis population balances: Investigation of three models. *Chemical Engineering Science*, 57:2305–2322, 2002.

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

[7] A. Eibeck and W. Wagner. Approximate solution of the coagulation-fragmentation equation by stochastic particle systems. *Stochastic Analysis and Application*, 18:921–948, 2000.

[8] A. Eibeck and W. Wagner. An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena. *SIAM Journal of Scientific Computating*, 22:802–821, 2000.

[9] A. Eibeck and W. Wagner. Stochastic particle approximations for Smoluchowski's coagulation equation. *Annals of Applied Probability*, 11:1137–1165, 2001.

[10] M. Goodson and M. Kraft. An efficient algorithm for simulating nano-particle dynamics. *Journal of Computational Physics*, 183:210–232, 2002.

[11] E. Debry, B. Sportisse, and B. Jourdain. A stochastic approach for the numerical sinulation of the general dynamics equation for aerosols. *Journal of Computational Physics*, 184:649–669, 2003.

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

[13] N Peters and J Warnatz, editors. *Numerical methods in laminar flame propagation: a GAMM-Workshop*. Braunschweig: Vieweg, 1982.

[14] J. Kee, K. Grcar, M. D. Smooke, and J. A. Miller. PREMIX: A FORTRAN program for modelling steady laminar one-dimensional premixed flames. Technical report, SANDIA National Laboratories, 1985.

[15] P.T. Spicer, O. Chaoul, S. Tsantilis, and S.E. Pratsinis. Titania formation by $TiCl_4$ gas phase oxidation, surface growth and coagulation. *Journal of Aerosol Science*, 33:17–34, 2002.

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

[17] M. Kraft and W. Wagner. Numerical study of a stochastic particle method for homogeneous gas phase reactions. *Computers & Mathematics with Applications*, 45:329–349, 2003.

[18] D. Lindackers, M. G. D. Strecker, P. Roth, C. Janzen, and S.E. Pratsinis. Formation and growth of SiO_2 particles in low pressure $H_2/O_2/Ar$ flames doped with SiH_4. *Combustion Science and technology*, 123:287–315, 1997.

[19] R.J. Kee, F.M. Rupley, E. Meeks, and J.A. Miller. CHEMKIN-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics. Technical report.

[20] R.J. Kee, G. Dixon-Lewis, J. Warnatz, M.E. Coltrin, and J.A. Miller. A FORTRAN computer code package for the evaluation of gas-phase multicomponent transport properites. Technical report, SANDIA National Laboratories, 1992.

[21] C. Janzen and P. Roth. Formation and characteristics of Fe_2O_3 nano-particles in doped low pressure $H_2/O_2/Ar$ flames. *Combustion and Flame*, 125:1150–1161, 2001.

[22] A. Giesen, J. Herzler, and P. Roth. Kinetics of Fe-atom condensation based on Fe-condensation measurement. *Journal of Physical Chemistry A*, 107(26).

[23] D. Woiki, A. Giesen, and P. Roth. A shock tube study on the theraml decomposition of $Fe(CO)_5$. Technical report, Institut fur Verbrennung und Gasdynamik, Gerhard-Mercator-Universitat Duisburg, 2002. (Pre-Print - to be submitted).

[24] NIST website. *http://webbook.nist.gov/chemistry*, 2003.