Preprint Cambridge Centre for Computational Chemical Engineering ISSN 1473 – 4273

A New Iterative Scheme For Solving The Discrete Smoluchowski Equation

Alastair J. Smith [1], Clive G. Wells [2], Markus Kraft [1,3]

released: 4 April 2017

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

[2] Hughes Hall
University of Cambridge
Wollaston Road
Cambridge, CB1 2EW
United Kingdom

[3] School of Chemical and Biomedical Engineering
Nanyang Technological University
62 Nanyang Drive
6357459
Singapore

Preprint No. 183

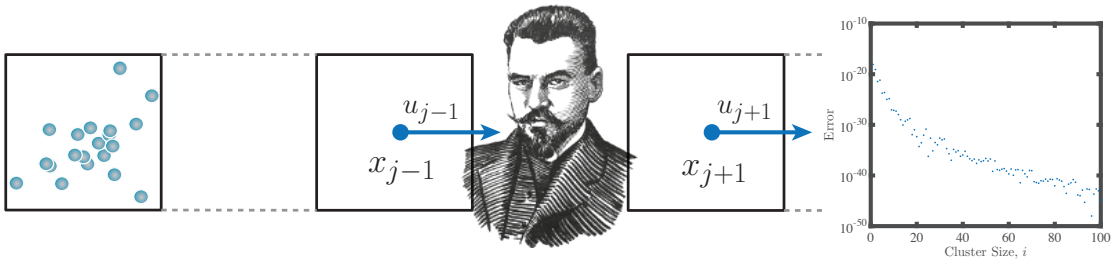
Keywords: Mathematical modelling, Simulation, Population balance, deterministic method, iterative scheme

Edited by
Computational Modelling Group
Department of Chemical Engineering and Biotechnology
University of Cambridge
New Museums Site
Pembroke Street
Cambridge CB2 3RA
United Kingdom

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

Abstract

This paper introduces a new iterative scheme for solving the discrete Smoluchowski equation and explores the numerical convergence properties of the method for a range of kernels admitting analytical solutions, in addition to some more physically realistic kernels typically used in kinetics applications. The solver is extended to spatially dependent problems with non-uniform velocities and its performance investigated in detail.



Highlights

- A new iterative scheme for the discrete Smoluchowski equation is presented.
- The numerical properties of the method are explored for a range of kernels.
- The solver is extended to spatially dependent problems with non-uniform velocities.
- It is suggested how the performance of the method could render it useful in CFD applications to industrial coagulation problems.

Contents

1 Introduction

The modelling of particle size distributions is of central importance in gaining a detailed understanding of a wide range of scientifically significant processes. Accurate models for the particle process rates are key, but in order to capitalise on this accuracy, numerically reliable methods to solve the population balance equations (PBEs) are required. Traditional approaches simplify the equations governing particle growth by excluding terms for particle transport (so called batch reactors), or assume the reactor model represents an axial streamline through the system (plug flow reactor) [10, 30, 31]. However, there are a great many systems where advective or diffusive particle transport is important [25, 32], necessitating the use of efficient population balance solvers which can be coupled to computational fluid dynamics (CFD) codes in order to accurately capture the particle dynamics.

An approach used frequently for the numerical solution to the partial integro-differential PBE is the method of classes (CM) of Valentas and Amundson [38]. The most widely used numerical schemes for CM are those of Hounslow et al. [22] and Kumar and Ramkrishna [26, 27]. Both of these schemes conserve only two moments in the discretised solution, generally particle number and volume (for coalescence/fragmentation problems). For this reason, in many applications this method is considered to be of low order and the accuracy of the solution improves only slowly with increasing number of classes. The use of such a method for detailed modelling requires a very large number of classes and is thus quite computationally intensive, especially if the solution is to be used in a wider computational fluid dynamics (CFD) framework or for a parameter fitting study.

Alternative methods for solving the coupled equations include monodisperse [11, 23] and moment [3, 5, 20, 28] methods. These methods lead to a system of ODEs which are solved within the CFD framework [25]. They are included in many commercial packages, for example STAR-CCM+ [12] and Fluent [7]. However, such approaches do not resolve the full particle size distribution, in many cases rendering them of limited use for detailed particle modelling applications.

The **purpose of this paper** is to develop a method for steady-state problems in a low number of dimensions, which has higher order accuracy, and has improved computational efficiency. The method relies upon the fact that in steady-state, the number density can be factorised, suggesting an iterative scheme for each particle size, the convergence of which can be enhanced by employing a number of numerical acceleration techniques. It is shown how the approach can be extended to one dimensional geometries by linking a sequence of (zero dimensional) cells together. Furthermore, the same methodology can be used to solve transient problems by observing the invariance of the underlying equation under a space-time transformation with a constant unit background velocity.

The structure of the paper is as follows. In §2 we introduce the fundamental PBE studied in this work, followed in §3 by a detailed description of an iterative algorithm to solve this equation and discussion of a number of improvements to enhance its computational efficiency. In §4, the detailed properties of the solver are analysed and its performance studied by comparison with a kernel admitting an analytic solution.

The remainder of this section extends the method to spatially dependent problems with non-uniform flows, and demonstrates how the method can be used to calculate the time evolution of the full particle size distribution. In §5 the solver is applied to a range of physically realistic kernels typically encountered in molecular dynamics applications. The numerical behaviour and performance of the algorithm is studied by comparison with an existing moment method.

2 The Smoluchowski Equation

Consider some domain Ω of volume V. Let $n^{\text{in}}(x)$ denote the number of particles of size $x \in \mathbb{R}^+$ entering Ω every $\alpha > 0$ units of time, and $\beta(x) > 0$ denote their size-dependent residence time. The number density of particles of size x at time $t > 0$, $n(x,t)$, obeys a more general form of the coagulation equation first introduced by Smoluchowski [36] (which we shall hereafter refer to as the *Smoluchowski equation*)

$$\frac{\partial n(x,t)}{\partial t} = \frac{1}{2}\int_0^x K(x-y,y)n(y,t)n(x-y,t)\,\mathrm{d}y + \frac{n^{\text{in}}(x)}{\alpha V} - \int_0^\infty K(x,y)n(y,t)n(x,t)\,\mathrm{d}y - \frac{n(x,t)}{\beta(x)}, \tag{1}$$

where $K : \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ is the symmetric coagulation kernel, and the probability that two particles of sizes x and y coalesce during a small time interval $\mathrm{d}t$ and volume $\mathrm{d}V$ is proportional to $K(x,y)\,\mathrm{d}t/\,\mathrm{d}V$ (note that K is not a pure rate because it has dimensions of volume/time rather than 1/time). In this same time interval, a particle can leave Ω with probability proportional to $\mathrm{d}t/\beta(x)$, or a new particle can enter Ω with probability proportional to $\mathrm{d}t/\alpha$. This equation is a fundamental mean-field model for cluster growth and arises in a diverse range of fields including physical chemistry, astrophysics, meteorology and the dynamics of biological systems. Aldous [6] gives a comprehensive general survey of existing literature in coalescence theory and discusses many of the applications of this equation, in addition to a number of interesting open problems. Pego [34] gives a review of some of the more recent work in the field. Melzak [29] discusses the local existence and uniqueness of solutions in general terms.

According to (1), the particle concentration $n(x,t)$ can increase either by the coagulation of particles of sizes $y < x$ and $x - y$ (first term) or simply by a particle of size x being incepted into Ω (second term), and can decrease by the coagulation of a particle of size x with any other particle of size y (third term) or simply by a particle of size x leaving the system after a size dependent time $\beta(x)$ (last term).

The discrete form of (1) is given by

$$\frac{\mathrm{d}n_i(t)}{\mathrm{d}t} = \frac{n_i^{\text{in}}}{\alpha V} + \frac{1}{2}\sum_{j=1}^{i-1}\beta_{i-j,j}n_{i-j}(t)n_j(t) - \sum_{j=1}^{N}\beta_{i,j}n_i(t)n_j(t) - \frac{n_i(t)}{\beta_i}, \tag{2}$$

where $n_i(t)$ describes the number density of particles of size $i \in \mathbb{N}$ at time t, $\beta_{i,j}$ is the discrete form of the collision kernel, describing the probability of clusters of

size i and j colliding and coagulating and N is the maximum cluster size in the system. The discrete and continuous cases can be analysed in the same framework by constructing the weak formulation of the equation [15, 16].

3 The Algorithm

3.1 Essential Idea

If we assume that there is a steady-state solution to (2), then as $t \to \infty$, we have $\mathrm{d}n_i(t)/\mathrm{d}t \to 0$ and we can rearrange (2) in terms of n_i. This defines a map $\mathcal{F} : \mathbb{R}^N \to \mathbb{R}^N$ given by

$$n_i \mapsto \mathcal{F}(n_i) = \frac{\frac{1}{2}\sum_{j=1}^{i-1} \beta_{i-j,j} n_{i-j} n_j + n_i^{\text{in}}/\alpha V}{\sum_{j=1}^{N} \beta_{i,j} n_j + 1/\beta_i}, \tag{3}$$

and a sequence of iterates for each cluster size $i \in \mathbb{N}$, which completely describe the discrete space. We assume monodispersed initial conditions, with all clusters having size 1, so $n_i(0) = \delta_{i1}$. The simulations performed in this paper are all with constant particle residence time $\beta_i = \beta, \forall i \in \mathbb{N}$ and we choose units in which $V = 1$.

This suggests an iterative method, **Algorithm 1**, in which we iterate to convergence (3) for each $i \in \mathbb{N}$. It should be noted that the only changes required to this algorithm when switching from a number density to a mass density representation of particles, is that the factor of $1/2$ disappears from the B_i term and n'_j is replaced by n'_j/j in both the D_i and B_i terms, with the appropriate re-interpretation of inflowing distribution.

Under appropriate conditions on the structure and behaviour of $\beta_{i,j}$, it may be possible to prove rigorously convergence of the method for specific coagulation kernels. However, in this paper, we are more interested in the speed and numerical properties of the scheme for specific classes of kernel.

3.2 Accelerating Convergence

It should be noted that it is straightforward to calculate the Jacobian of (3), which we can use to Taylor expand $\mathcal{F}$ about the point $n_i^{(p)} \in \mathbb{R}^N$

$$\mathcal{F}(n_i^{(p+1)}) \sim \mathcal{F}(n_i^{(p)}) + J\mathcal{F}(n_i^{(p)})(n_i^{(p+1)} - n_i^{(p)}).$$

Solving this for the fixed point suggests the Newton scheme

$$n_i^{(p+1)} = \left[J\mathcal{F}(n_i^{(p)})\right]^{-1} \mathcal{F}(n_i^{(p)}),$$

Algorithm 1: Steady-State Iterative Population Balance Solver

input : Maximum cluster size N
Initial number density n_i
Inflowing number density $n_i^{\text{in}} \longleftarrow \delta_{i1}$
α^{-1} and β^{-1} (the in and outflow rates respectively)
Number of moments $K+1$
Maximum residual tolerance for first $K+1$ moments $r_{\max}$

output : Steady-state number density n_i
First $K+1$ moments m_k

update : $\delta m_{\max} \longleftarrow r_{\max} + 1$
update : $m'_k \longleftarrow 0$
while $\delta m_{max} > r_{max}$ **do**
  for $i \longleftarrow 1$ **to** N **do**
    update : $n'_i \longleftarrow n_i$
    calculate:

$$D_i \longleftarrow \sum_{j=1}^{N} \beta_{i,j} n'_j$$

    calculate:

$$B_i \longleftarrow \frac{1}{2} \sum_{j=1}^{i-1} \beta_{i-j,j} n'_{i-j} n'_j$$

    update :

$$n_i \longleftarrow \frac{n_i^{\text{in}}/\alpha + B_i}{1/\beta + D_i}$$

  for $k \longleftarrow 0$ **to** K **do**
    calculate: $m_k \longleftarrow \sum_{i=1}^{N} i^k n_i$
  update : $\delta m_{\max} \longleftarrow \max_{0 \leqslant k \leqslant K} |m_k - m'_k|$
  update : $m'_k \longleftarrow m_k$

which was found to have strikingly fast convergence, but unfortunately when N is large the calculation of the inverse can be expensive and the Jacobian is often singular without very restrictive conditions on $n_i(0)$.

Therefore, we instead adopt an acceleration strategy known as *Aitken's delta-squared process* [2]. The method is as follows. Given a sequence $\left(n_i^{(p)}\right)_{p\in\mathbb{Z}}$, we associate a new sequence

$$A(n_i^{(p)}) = n_i^{(p)} - \frac{(\Delta n_i^{(p)})^2}{\Delta^2 n_i^{(p)}},$$

where

$$\Delta n_i^{(p)} = n_i^{(p+1)} - n_i^{(p)},$$

and

$$\Delta^2 n_i^{(p)} = n_i^{(p)} - 2n_i^{(p+1)} + n_i^{(p+2)},$$

where $p \in \{0\} \cup \mathbb{N}$. The sequence is well-defined provided $\Delta^2 n_i^{(p)} \neq 0$. Assuming $\Delta^2 n_i^{(p)} = 0$ for only a finite number of indices p, we consider the sequence $A(n_i^{(p)})$ restricted to indices $p > p_0$ with a sufficiently large p_0. We must be careful to stop the calculation when rounding errors become too large in the denominator, i.e., when too many significant digits cancel in the calculation of Δ^2, leading to a loss of precision upon division. We modify the updating of the number density in Algorithm 1 using **Algorithm 2**[1].

Algorithm 2: Aitken's Delta-Squared Process

for $i \longleftarrow 1$ **to** N **do**

calculate:

$$n_i^{(1)} \longleftarrow \mathcal{F} n_i^{(0)}$$

calculate:

$$n_i^{(2)} \longleftarrow \mathcal{F} n_i^{(1)}$$

update :

$$\Delta^2 n_i^{(2)} \longleftarrow n_i^{(2)} - 2n_i^{(1)} + n_i^{(0)}$$

if $\left|\Delta^2 n_i^{(2)}\right| < \epsilon_{machine}$ **then**

Warning: denominator is too small

No solution found!

halt

update :

$$n_i^{(0)} \longleftarrow n_i^{(2)} - \frac{\left(n_i^{(2)} - n_i^{(1)}\right)^2}{\Delta^2 n_i^{(2)}}$$

[1] Whilst this method is most applicable to linearly convergent processes, it nevertheless seems to afford us modest increase in the speed of convergence.

3.3 Implementation

We can further enhance the speed of the algorithm by exploiting the symmetry of the kernel $\beta_{i,j} = \beta_{j,i} \forall i, j \in \mathbb{N}$ and, particularly in those cases where the kernel has a complicated structure, significant computational speed increases can be realised by pre-computing the values of the kernel, a variation of an optimisation technique known as memoization [1]. Calculation of the birth and death terms then requires a table lookup rather than a calculation which is quadratic in the cluster sizes and would have to be performed many times per iteration.

In addition to this, when the number density of clusters of a given size is very small, the contribution to the sum in the birth and death terms will be small, so we can skip over these values when looping over all cluster sizes, in some cases affording a significant increase in computational speed, at the expense of only a moderate loss of precision.

The algorithm was implemented in C++ using all of the enhancements mentioned.

4 Analytical & Numerical Solutions

4.1 Multiplicative Kernel

It is known that at least three particular classes of kernel are analytically soluble: the constant, additive and multiplicative. Calculations were conducted for all of these kernels, and the behaviour of the solutions and the properties of the iterative solver studied in detail. However, of these, we present only results for the multiplicative kernel, not only because the behaviour of the solver for this class is entirely indicative of the others, but also because it has the additional property of being a gelling kernel [6, 14, 39], so offers a slightly more challenging numerical experiment than the others.

The multiplicative kernel takes the form $\beta_{i,j} = Kij$, for some constant $K \in \mathbb{R}^+$ (which we can always take to be 1 by rescaling time). For a coagulation process with both in and outflow, the discrete Smoluchowski equation (2) for the number density n_i takes the form

$$\frac{\mathrm{d}n_i}{\mathrm{d}t} = \frac{n_i^{\mathrm{in}}}{\alpha} + \frac{K}{2}\sum_{j=1}^{i-1}(i-j)jn_{i-j}n_j - Kin_i m_1 - \frac{n_i}{\beta}, \tag{4}$$

where m_1 is the first moment, which we calculate from the general definition of the k^{th} moment

$$m_k(t) = \sum_{i=1}^{\infty} i^k n_i(t). \tag{5}$$

Multiplying (4) by i^k and summing over all $i \in \mathbb{N}$ furnishes us with a differential

equation for the moments

$$\frac{\mathrm{d}m_k}{\mathrm{d}t} = \frac{m_k^{\mathrm{in}}}{\alpha} + \frac{K}{2}\sum_{p=1}^{k-1}\binom{k}{p} m_{p+1}m_{k-p+1} - \frac{m_k}{\beta}. \tag{6}$$

The equation for $k = 1$ decouples from the equations for all other k, hence we first solve (6) for $k = 1$. We find

$$m_1(t) = \gamma + (1-\gamma)e^{-t/\beta},$$

and so $m_1(t) = 1\,\forall t \in \mathbb{R}$ when $\gamma = \alpha/\beta = 1$.

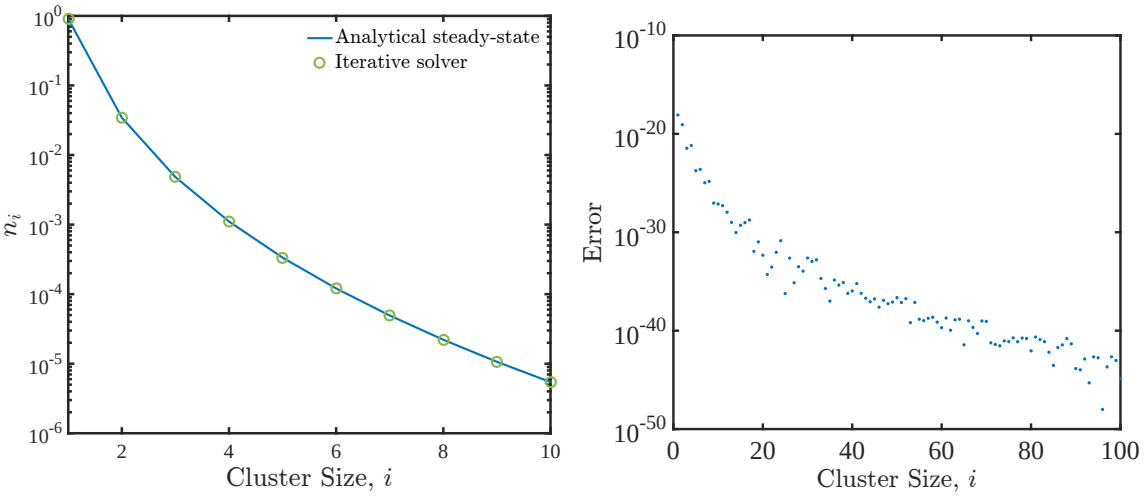
Equipped with knowledge of the first moment, we are able to calculate the steady-state[2] number density

$$\widetilde{n}_i = \frac{\frac{K}{2}\sum_{j=1}^{i-1}(i-j)j\widetilde{n}_{i-j}\widetilde{n}_j}{iK\widetilde{m}_1 + 1/\beta} = \frac{\frac{K}{2}\sum_{j=1}^{i-1}(i-j)j\widetilde{n}_{i-j}\widetilde{n}_j}{iK + 1/\beta},$$

and similarly the zeroth moment. We find

$$m_0(t) = \widetilde{m}_0 + (m_0(0) - \widetilde{m}_0)\, e^{-t/\beta},$$

where the steady-state of the zeroth moment is given by $\widetilde{m}_0 = \gamma - K\beta/2$.



(a) *Steady-state values of number density.* **(b)** *Relative error in number density.*

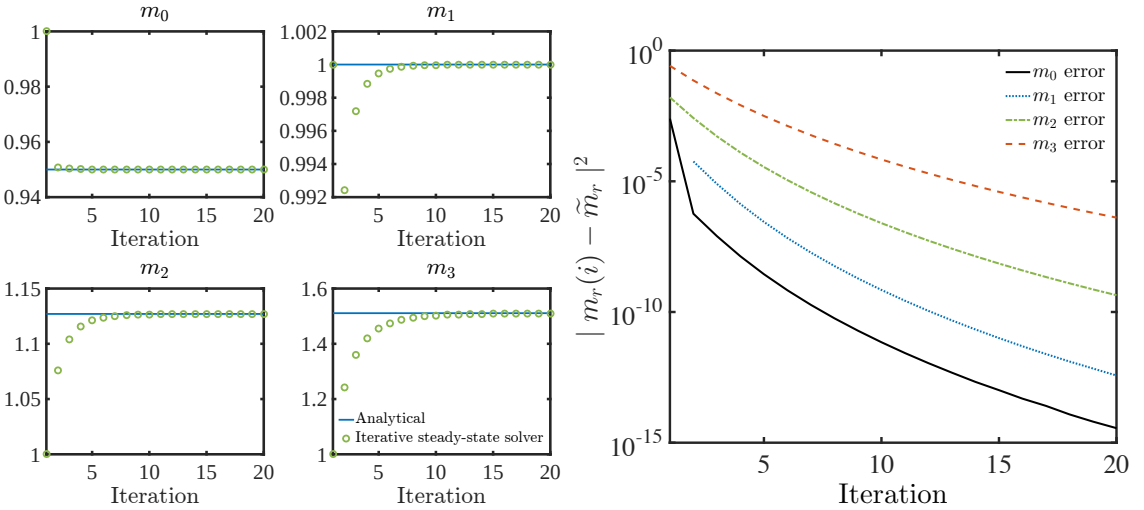
Figure 1: *Comparison between values of number density calculated using iterative solver with analytical solution when* $\beta_{i,j} = ij$ *and* $\alpha = \frac{1}{10} = \beta$.

Numerical simulations with $\alpha = \frac{1}{10} = \beta$ and $\beta_{i,j} = ij$, and with a maximum cluster size of $N = 2^{16}$ were performed. The comparison between the analytical expressions for the number density with the steady-state values (we plot only the number density of clusters up to size 10, because the number density very rapidly diminishes with

[2]We use tildes over variables to denote their steady-state values.

increasing cluster size) obtained by the iterative solver are given in **Figure 1a**. **Figure 1b** shows the relative error in approximating the steady-state analytical solution for clusters up to size 100.

The first four moments of number density are plotted along with the solutions obtained from the iterative solver in **Figure 2a**. **Figure 2b** shows the error in estimating the steady-state moments of number density for each iteration of the solver. We see that 14 iterations are required for the error in the first 4 moments to be less than 10^{-5}, and a further 4 for the error to be less than 10^{-6}.



(a) *Evolution to steady-state of first 4 moments* **(b)** *Error in estimating the steady-state moments of number density*[3].

Figure 2: *Comparison between moments calculated using iterative solver with analytical solution when* $\beta_{ij} = ij$ *and* $\alpha = \frac{1}{10} = \beta$.

4.2 Computational Efficiency

In order to establish the computational efficiency of the algorithm, we defined a maximum residual error over all moments and recorded the CPU times required for the solver to be within this tolerance of the true solutions, in addition to the number of iterations required to achieve this. We used the same parameters and initial conditions as those used in the simulations of the previous section. All calculations where performed on a single core of an Intel® Xeon® X5472 CPU with a clockspeed of 3.00 GHz and 12 Mb of L2 cache.

The results of the simulations are plotted in **Figure 3**. This figure shows, for example, that to achieve a maximum residual of 10^{-8} requires approximately 30 iterations of the solver (green solid curve), which requires around 1 s of CPU time (blue dashed curve).

[3]N.B., the initial condition for the first moment is the steady-state solution, so the error is zero and therefore does not appear on the logarithmic scale.

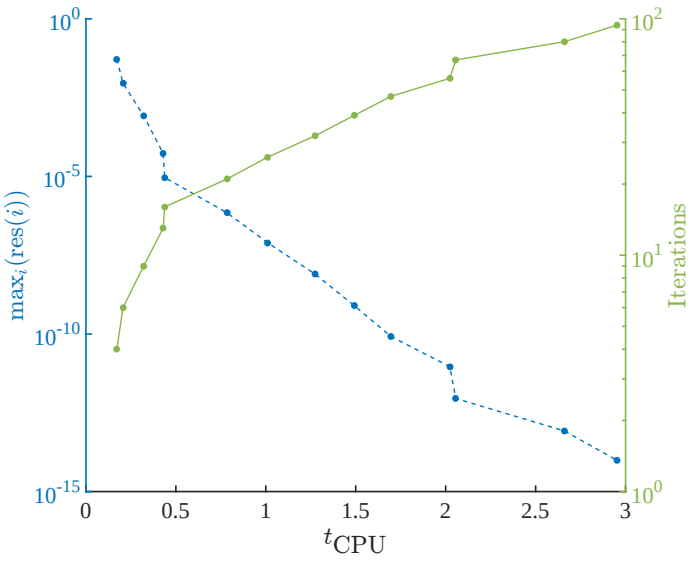


Figure 3: *Computational efficiency of iterative solver. Maximum residual error and number of iterations against CPU time required.*

It should be noted that the steady-state equation system can, in principle, also be solved using a general purpose nonlinear solver, such as Matlab's `fsolve`, and indeed the performance of the iterative solver was found to be similar to this solver when N is small. However, for larger cluster sizes ($N \gg 2^8$), the system becomes intractable with `fsolve`, whereas the iterative approach remains practicable, even for the maximum cluster sizes far exceeding 2^{16}.

Using the refinements of §3.3, the iterative population balance solver is shown to be rather efficient in its domain of application.

4.3 Spatially Inhomogeneous Geometries

In the previous section we did not account for any spatial variation, considering only the evolution to steady-state. This is equivalent to the assumption of spatial homogeneity, i.e., a zero-dimensional system. However, we can consider these zero-dimensional systems to be cells, Ω_j, which we can network together to form a quasi-one dimensional geometry, the inflowing distribution of particles being given by the steady-state distribution of particles in the previous cell. This will afford us full spatial resolution in (at least) one dimension. The construction of this one-dimensional geometry from a string of zero-dimensional cells is illustrated in **Figure 4**.

We have two viable iterative strategies: (i) we can iterate to steady-state in each cell before transporting to the next, the distribution of inflowing particles in the next cell being given by the steady-state distribution in the previous cell; or (ii) we can transport the particles from each cell and iterate until we have global convergence across the entire domain. Both of these strategies were tested, and it was found that strategy (i) (cell-wise convergence), with the obvious cell ordering was more

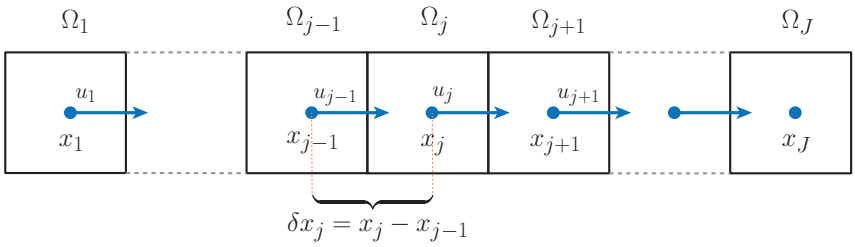


Figure 4: *One-dimensional geometry. The geometry is constructed by linking J cells Ω_j, with the inflow rate of each cell determined by the local velocity in that cell along with its grid spacing. The first cell, Ω_1, gives the boundary condition, and is equivalent to the distribution of inflowing particles, n^{in}, in the zero-dimensional case.*

computationally efficient, at least when the velocity is positive, so it is this approach that we adopt throughout this study. It should, however, be noted that in more complex system with negative or variable velocities in each cell, that strategy (ii) (global convergence), might be required.

Stategy (i) leads to **Algorithm 3**.

Algorithm 3: Quasi-One Dimensional Iterative Population Balance Solver

input : Maximum cluster size N
Initial condition: $n_i(x,0) \forall x$
Boundary condition: $n_i(0,t) = \delta_{i1} \forall t > 0$
Grid $(\Omega_j, x_j, u_j), 1 \leqslant j \leqslant J$,
(cell numbers, centres and velocity of fluid in cell Ω_j)
Number of moments K
Maximum residual tolerance for first K moments $r_{\max}$

output : Spatially resolved steady-state number density $\widetilde{n}_i(x)$
First K spatially resolved, steady-state moments $\widetilde{m}_k(x)$

$j \longleftarrow 1$;
while $j \leqslant J$ **do**
  if $j = 1$ **then**
    $n_i^{\text{in}} \longleftarrow n_i(0,t)$;
  else
    $n_i^{\text{in}} \longleftarrow \widetilde{n}_i(x_{j-1})$ (where $\widetilde{n}_i(x) = \lim_{t\to\infty} n_i(x,t)$);
  $\delta x_j \longleftarrow x_j - x_{j-1}$;
  $\alpha \longleftarrow \frac{\delta x_j}{u_j}$;
  $\beta \longleftarrow \frac{\delta x_j}{u_j}$;
  Call 0D solver (Algorithm 1+2) with inputs $(n_i^{\text{in}}, \alpha, \beta)$;
  $j \longleftarrow j + 1$;

Recall that we have been solving the Smoluchowski equation (2) for the number density $n_i(t)$, of particles of size i at time t. We now consider the number density $n_i(x_j, t)$, of particles of size i at time t *and* position x_j (the centroid of cell Ω_j). (2) now becomes a partial differential equation

$$\frac{\partial n_i(x_j,t)}{\partial t} = \frac{n_i(x_{j-1},t)}{\alpha} + \frac{1}{2}\sum_{\ell=1}^{i-1}\beta_{i-\ell,\ell}n_{i-\ell}n_\ell - \sum_{\ell=1}^{N}\beta_{i,\ell}n_i n_\ell - \frac{n_i(x_j,t)}{\alpha}, \tag{7}$$

where we now have the inflowing distribution given by $n_i(x_{j-1}, t)$, the (steady-state) distribution of particles in cell Ω_{j-1} (the previous spatial discretisation step), and the inflow rate α^{-1} is fixed by β^{-1}, the outflow rate in the previous cell.

Now, given that α^{-1} is a rate, we take

$$\alpha^{-1} = \frac{u}{\delta x},$$

where u is the uniform background velocity in all cells, and $\delta x = x_j - x_{j-1}$ (uniform spacing), we find that

$$\frac{n_i(x_{j-1},t)}{\alpha} - \frac{n_i(x_j,t)}{\alpha} = -u\frac{n_i(x_j,t) - n_i(x_{j-1},t)}{\delta x} \to -u\frac{\partial n_i(x,t)}{\partial x}$$

as $\delta x \to 0$. We thus find that as we pass to the limit in (7), $n_i(x, t)$ satisfies the advection equation

$$\frac{\partial n_i(x,t)}{\partial t} + u\frac{\partial n_i(x,t)}{\partial x} = \frac{1}{2}\sum_{\ell=1}^{i-1}\beta_{i-\ell,j}n_{i-\ell}n_\ell - \sum_{\ell=1}^{N}\beta_{i,\ell}n_i n_\ell. \tag{8}$$

Multiplying by i^k and summing over all i in the usual way, furnishes us with the equation for the moments of number density.

$$\frac{\partial m_k(x,t)}{\partial t} + u\frac{\partial m_k(x,t)}{\partial x} = \sum_{p=1}^{k-1}\binom{k}{p}\sum_{\ell=1}^{N}\sum_{j=1}^{N}\ell^{k-p}j^{p-1}\beta_{\ell,j}n_\ell n_j. \tag{9}$$

This is the general transport equation for the moments of number density for an arbitrary coagulation kernel. It is a one dimensional linear partial differential equation, so, in order to solve it we must specify both initial conditions and boundary conditions: $m_k(x, 0)$ and $m_k(x, t)|_{\partial\Omega}$. Note that, in the 1D case, the boundary condition is equivalent to the inflowing distribution of particles in the 0D case, i.e., $m_k(x, t)|_{\partial\Omega} = m_k^{\text{in}}$.

We shall now investigate the performance of this 1D iterative solver for the multiplicative kernel, $\beta_{i,j} = Kij$. We again choose mono-dispersed boundary conditions (i.e., delta distributed) $n_i(0, t) = \delta_{ia}$, so the boundary conditions for the moments are $m_k(0, t) = a^k, \forall t$. We assume mono-dispersed initial conditions, $m_k(x, 0) = 1, \forall x$.

In this case, (9) reduces to

$$\frac{\partial m_k(x,t)}{\partial t} + u\frac{\partial m_k(x,t)}{\partial x} = \frac{K}{2}\sum_{p=1}^{k-1}\binom{k}{p}m_{p+1}m_{k-p+1}. \tag{10}$$

In steady-state, the time derivative drops out, so this reduces to a simple set of ODEs for the moments, which are solved trivially:

$$m_0(x) = 1 - \frac{Ka^2}{2u}x,\ m_1(x) = a,\ m_2(x) = \frac{a^2}{1 - Ka^2x/u},\ m_3(x) = \frac{a^3}{(1 - Ka^2x/u)^3}.$$

Notice that we have a potential problem in that the higher moments blow up at a finite position, when $x = u/(Ka^2)$. Therefore, we will find the solution breaks down if we attempt to simulate the coagulation process in a geometry of length exceeding this. In such cases the iterative solver will be unable to resolve the solution near the discontinuity. In particular, we found that m_0 loses accuracy from this point onwards, and the solver in some sense attempts to smooth over the discontinuity in the higher order moments. This is because the multiplicative kernel is gelling [6].

In general, care is therefore needed to ensure that the particular choice of constants does not lead to singular solutions. The multiplicative coagulation process was sim-

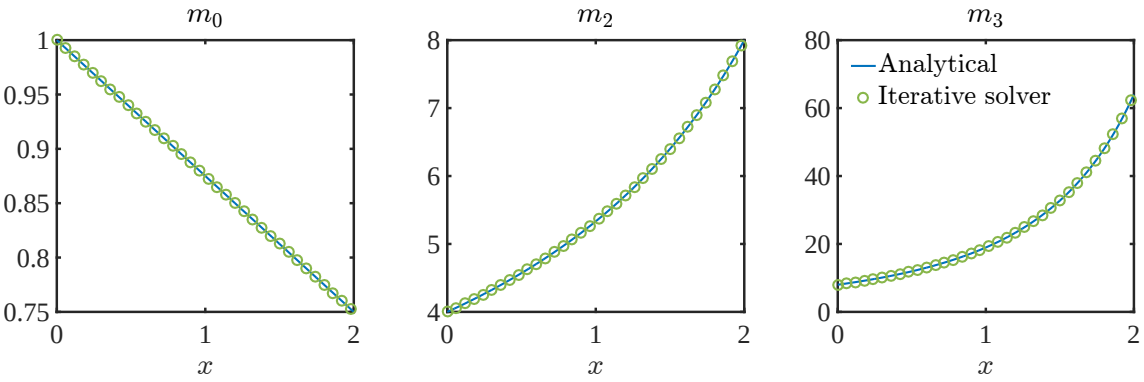


Figure 5: *Comparison between distribution of steady-state values for moments of number density for iterative solver with analytical solution in a 1D geometry of length 2 (discretised into 1000 cells) with $\beta_{i,j} = Kij$ in a uniform background velocity field of $u = 1$ with $K = 1/16$ and $a = 2$.*

ulated in a geometry of length 2 units (discretised into 1000 cells), with coagulation constant $K = 1/16$, with mono-dispersed inflowing particles of size $a = 2$, and with a maximum cluster size of $N = 2^{16}$. This gives rise to the steady-state distribution of moments shown in **Figure 5**.

4.4 Non-Uniform Velocity Fields

The method can also be used to solve the coagulation equation describing particles flowing in a non-uniform velocity field, for example, particles entrained in a fluid, by reading in a grid containing the velocities in each cell, along with the centroid to centroid cell spacing (not necessarily uniform). Consider the simple case of fluid flowing with a constant acceleration, g, with the particle coagulation process described by the multiplicative kernel, $\beta_{i,j} = Kij$. The velocity at a point x will then

be given by $u(x) = gx + u_0$, and the steady-state transport equation for the zeroth moment becomes

$$(gx + u_0)\frac{\mathrm{d}m_0}{\mathrm{d}x} = -\frac{K}{2}m_1^2.$$

This has solution

$$m_0(x) = 1 + \frac{Ka^2}{2g}\log\left(\frac{1}{1 + gx/u_0}\right),$$

since, as usual, we have $m_1(x) = m_1(0) = a, \forall x$. Similarly, it can be shown that the second and third moments are given by

$$m_2(x) = \frac{a^2}{2m_0(x) - 1}, \quad \text{and} \quad m_3(x) = \left(\frac{a}{2m_0(x) - 1}\right)^3.$$

These solutions are compared with those from the iterative solver for a geometry of length 2 units in **Figure 6**

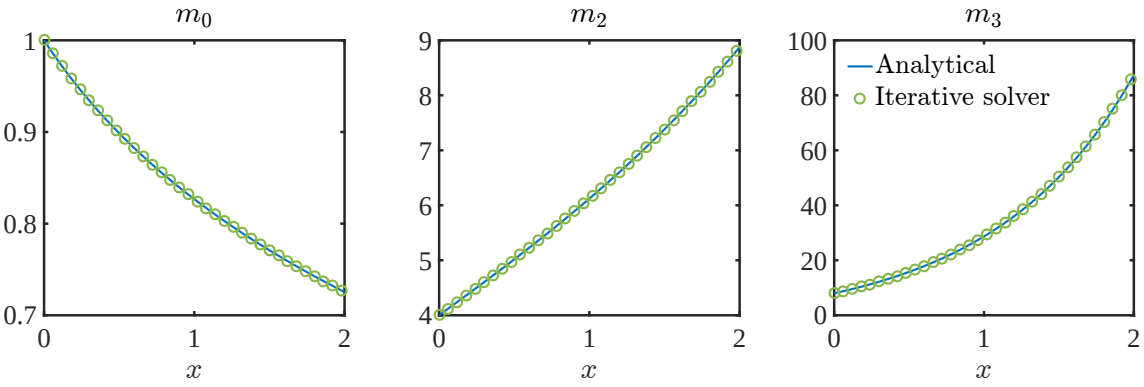


Figure 6: *Comparison between distribution of steady-state values for moments of number density for iterative solver with analytical solution in a 1D geometry of length 2 units (discritisied into 1000 cells) with $\beta_{i,j} = Kij$ in a background velocity field of $u = \frac{1}{2}(x + 1)$ with $K = 1/16$ and $a = 2$.*

4.5 Time-Dependent Case

Notice that if we solve (9) in a uniform background velocity of $u = 1$, then under the interchange of space and time coordinates, the equation describes moment transport in a time-dependent zero-dimensional system

$$\frac{\partial m_k(x,t)}{\partial t} = \sum_{p=1}^{k-1}\binom{k}{p}\sum_{\ell=1}^{N}\sum_{j=1}^{N}\ell^{k-p}j^{p-1}\beta_{\ell,j}\mu_\ell\mu_j,$$

so we may use our quasi-one dimensional steady-state solver to resolve the time evolution in a zero-dimensional system, no longer being confined to steady-state. Essentially, we are considering our grid to discretise physical time, and we reach a

steady-state in "pseudo time" at each physical time step. This idea could be very useful in coupling to CFD, for example, it could be used to solve particle dynamics in a complex geometries by post-processing streamlines calculated in CFD [5, 9]. The particle size distributions in a full three-dimensional geometry could then by reconstructed using statistical techniques, for example, kernel density estimation. This method will be employed in the next section.

5 Application to Kinetics: Physically Realistic Kernels

In this section we shall study the convergence behaviour of the iterative solver for a number of more physically realistic kernels, typically arising in kinetics applications. We will compare the solutions with those calculated using a well known quadrature moment method [18, 28], combined with an interpolative closure treatment of the fractional order moments [3–5, 19, 20].

The form of the kernel is dictated by the physics of the interactions of pairs of particles. The chief drivers of particle transport are Brownian motion, turbulence and gravitational settling. Different kernel types can be classified for different pressure regimes on the basis of the *Knudsen number*, $\mathrm{Kn} = 2\lambda/d$, where λ is the mean free path of the fluid and d is a characteristic particle diameter. For $\mathrm{Kn} \leqslant 0.1$, the particles are said to be in the *continuum regime*. For $0.1 < \mathrm{Kn} \leqslant 1$, they are in the *slip-flow regime*, and for $\mathrm{Kn} > 10$, they are in the *free-molecular regime*. The remaining case $1 < \mathrm{Kn} \leqslant 10$ is the *transition regime*.

5.1 Continuum Kernel

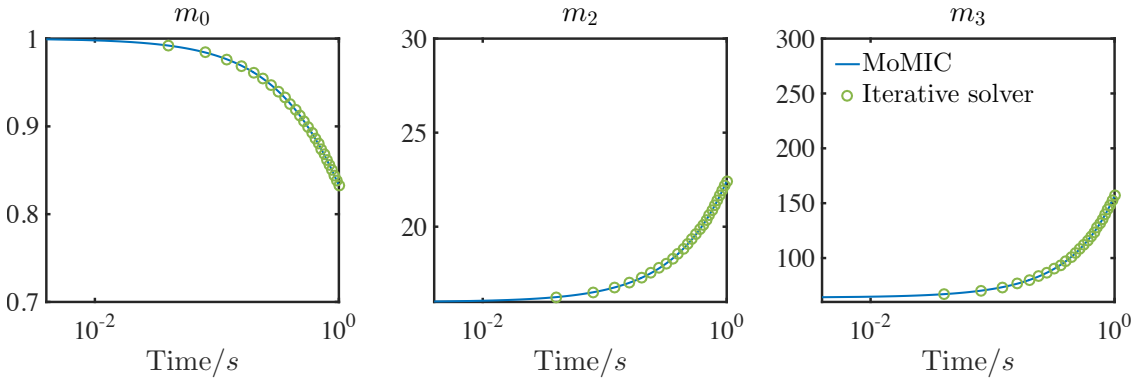
The continuum kernel becomes applicable when the particle size is large relative to the mean free path of the fluid molecules and hence the particles act as a continuum, particle transport being dictated by diffusion processes. The particles, typically smaller than $1\,\mu$m, will collide as a result of Brownian motion.

The form of this kernel can be established by solving the 1D diffusion equation in spherical coordinates and applying the Stokes-Einstein relation for the diffusion coefficient [17, 37], which is valid when the particle diameters are much larger than the mean free path. This gives the *continuum regime kernel*:

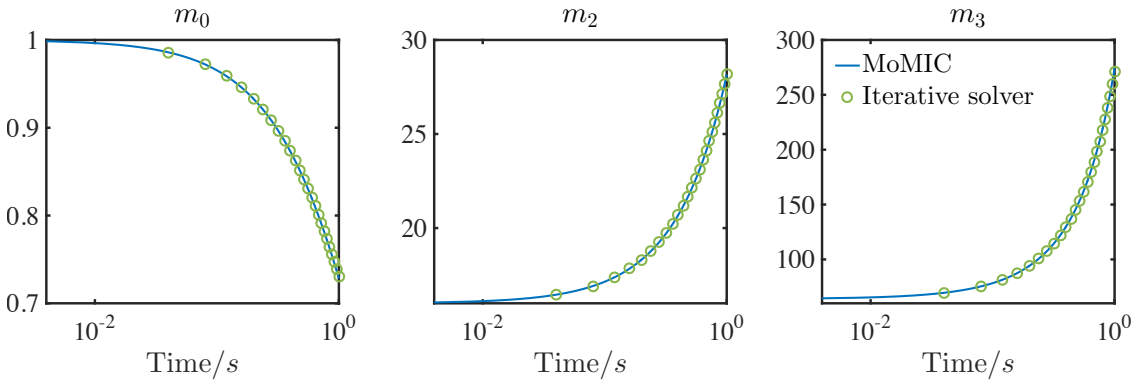
$$\beta^{\mathrm{c}}(v_i, v_j) = K_0 \left(\frac{1}{v_i^{1/3}} + \frac{1}{v_j^{1/3}} \right) \left(v_i^{1/3} + v_j^{1/3} \right), \tag{11}$$

where $K_0 = 2k_B T/3\mu$, with k_B the Boltzmann constant, T the temperature and μ the absolute viscosity of the fluid. It is assumed that the diffusion coefficients do not change as the particles approach each other.

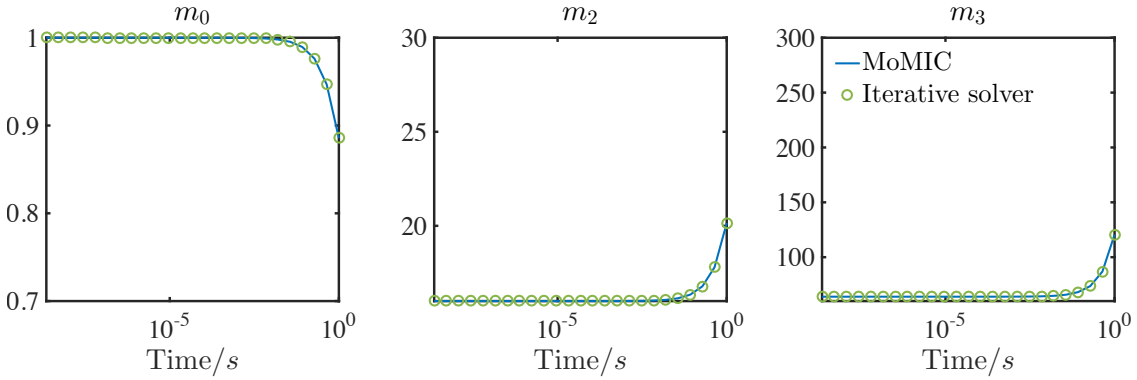
The iterative solver was used to simulate the dynamics of particles undergoing coagulation in this regime in a geometry of length 1 unit (1000 cells), using the technique



(a) *Continuum kernel:* $\beta_{i,j} = K_0 \left(i^{\frac{1}{3}} + j^{\frac{1}{3}}\right)\left(i^{-\frac{1}{3}} + j^{-\frac{1}{3}}\right)$.



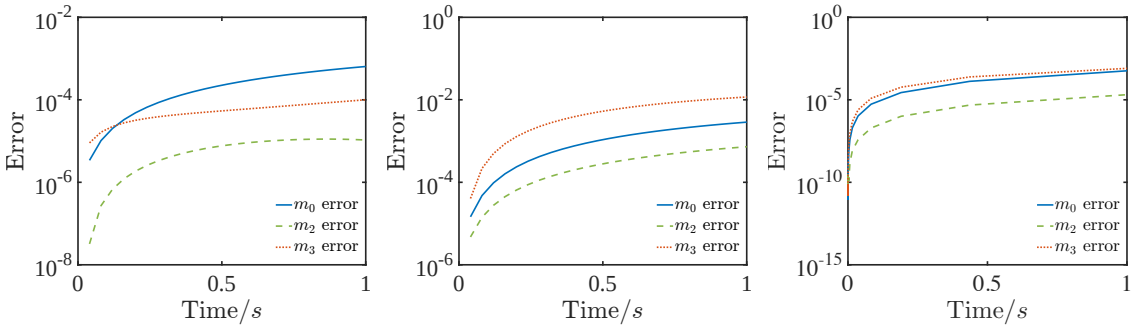
(b) *Freemolecular kernel:* $\beta_{i,j} = K_f \sqrt{i^{-1} + j^{-1}} \left(i^{\frac{1}{3}} + j^{\frac{1}{3}}\right)^2$.



(c) *Transition kernel: (14).*

Figure 7: *Comparison between moments of number density for iterative solver with moment method solution for* $0 \leqslant t \leqslant 1\,\mathrm{s}$ *(discretised into 1000 cells).*

of §4.5. The properties of a test fluid were chosen in units such that the coagulation constant was given by $K_0 = 1$ and the inflowing particles were mono-dispersed with a size of $a = 4$ units. The evolution of the (nontrivial) moments are compared with the moment method solutions in **Figure 7a**. **Figures 8a & 9a** show the relative errors in calculating these moments and the residuals at each time step respectively.



(a) *Continuum kernel.* **(b)** *Freemolecular kernel.* **(c)** *Transition kernel.*

Figure 8: *Relative error in calculating moments of number density using the iterative solver.*

5.2 Slip Flow Kernel

The diffusion model can be extended up to $\mathrm{Kn} = 1$ by modifying Stokes' law using the Cunningham correction factor [13], $C_i = 1 + 1.257\,\mathrm{Kn}_i$, where the particle dependent Knudsen number is given by $\mathrm{Kn}_i = 2\lambda/d_i$, with d_i the diameter of a particle containing i atoms. For spherical particles this gives [24]

$$\beta^{\mathrm{sf}}(v_i, v_j) = \beta^{\mathrm{c}}(v_i, v_j) + K_0 K_0' \left(v_i^{-\frac{2}{3}} + v_j^{-\frac{2}{3}}\right)\left(v_i^{\frac{1}{3}} + v_j^{\frac{1}{3}}\right), \tag{12}$$

where $K_0' = 2.514\lambda\,(6\rho_s/\pi m_1)^{1/3}$, m_1 is the mass of the smallest particle and ρ_s is the particle density.

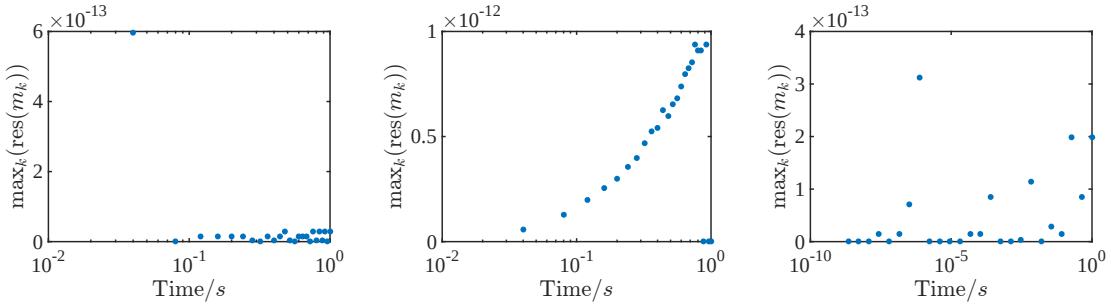
We do not simulate coagulation in the slip-flow regime in this work, but instead use it to construct the transition kernel in §5.4.

5.3 Free-Molecular Kernel

For particles much smaller than the mean free path, the collision frequency is obtained from an expression derived in the kinetic theory of gases for collisions among molecules which behave like rigid elastic spheres. It can be shown that

$$\beta^{\mathrm{fm}}(v_i, v_j) = K_f \left(v_i^{-1} + v_j^{-1}\right)^{\frac{1}{2}} \left(v_i^{1/3} + v_j^{1/3}\right)^2, \tag{13}$$

where $K_f = \varepsilon_{ij}\,(3m_1/4\pi\rho_s)^{\frac{1}{6}}\,(6k_BT/\rho_s)^{\frac{1}{2}}$, where ε_{ij} is a size-dependent coagulation enhancement factor due to attractive or repulsive inter-particle forces, which for



(a) *Continuum kernel.* **(b)** *Free-molecular kernel.* **(c)** *Transition kernel.*

Figure 9: *Cell-wise residuals in moments of number density.*

uncharged particles, following Balthasar [8], we set to 2.2. This is the *free-molecular regime kernel*.

Once again, the properties of a test fluid were chosen in units such that the coagulation constant was given by $K_f = 1$ and the inflowing particles were mono-dispersed with a size of $a = 4$ units. The calculation for the evolution of particles using the iterative solver in the same geometry as the previous section are compared with the moment method solutions in **Figure 7b**. The relative errors in the moments are shown in **Figure 8b**. The errors show a similar pattern to the continuum case, though they are a little larger and the error in the third moment remains larger than the other two. The residuals are shown in **Figure 9b**. Notice how the residuals increase slightly with increasing time. This is because the time resolution comes from solving the 1D case, and so the variance in predicting the inflowing distribution slightly increases as we progress along the geometry.

5.4 Transition Kernel

The final kernel we will consider deals with the regime where $1 < \mathrm{Kn} \leqslant 10$. Fuchs [21] proposed a general interpolation formula for β for the transition from (13) to (11) and from this Pratsinis [35] developed an approximate transition kernel based on a harmonic mean, which is valid across a wide range of Knudsen numbers. Kazakov and Frenklach [24], Patterson et al. [33] increased the efficiency of the rate calculation by taking half of the harmonic mean of the slip flow and free-molecular kernels (which is distinct from the harmonic mean kernel). This kernel is given by

$$\beta^{\mathrm{tr}}(v_i, v_j) = \beta^{\mathrm{sf}}(v_i, v_j)\left(1 + \frac{\beta^{\mathrm{sf}}(v_i, v_j)}{\beta^{\mathrm{fm}}(v_i, v_j)}\right)^{-1}. \tag{14}$$

The results of the numerical simulations with this kernel are presented in **Figure 7c**. The corresponding relative error and residual plots are given in **Figures 8c & 9c**

6 Conclusions

This paper introduced a new iterative algorithm for solving population balance equations and studied its mathematical and numerical properties under a range of conditions. The algorithm relies on the fact that in steady-state, the number density can be factorised from the population balance equation, enabling an iterative map to be defined for each cluster size.

The solver was found to perform very well even when considering the complex collision kernels typically encountered in physical systems. In particular, the method was shown to offer accurate resolution of the moments of the number density with minimal computational effort, in contrast to many existing class based solution schemes for population balance equations.

The algorithm was extended to one dimensional geometries with non-uniform flow fields, and found to compare favourably with a conventional moment method. Using a space-time correspondence under a constant background velocity, it was shown that the method could be extended from a steady-state solver to a full transient solver.

The computational efficiency of the method may render it suitable for coupling to CFD, and as such this method for solving population balance equations may play a key role in detailed particle modelling applications. For example, it could be used to calculate particle properties along streamlines in a complex geometry (e.g., nanoparticle formation in an industrial reactor). This work has contributed to the understanding of the numerical aspects of such an approach.

References

[1] U. A. Acar, G. E. Blelloch, and R. Harper. Selective memoization. *SIGPLAN Not.*, 38(1):14–25, 01 2003. ISSN 0362-1340. doi:10.1145/640128.604133.

[2] A. Aitken. On Bernoulli's numerical solution of algebraic equations. *Proceedings of the Royal Society of Edinburgh*, 46:289–305, 1926.

[3] J. Akroyd, A. Smith, L. R. McGlashan, and M. Kraft. Numerical investigation of DQMoM-IEM as a turbulent reaction closure. *Chemical Engineering Science*, 65(6):1915–1924, 2010. doi:10.1016/j.ces.2009.11.010.

[4] J. Akroyd, A. Smith, L. R. McGlashan, and M. Kraft. Comparison of the stochastic fields method and DQMoM-IEM as turbulent reaction closures. *Chemical Engineering Science*, 65(20):5429–5441, 2010. doi:10.1016/j.ces.2010.06.039.

[5] J. Akroyd, A. J. Smith, R. Shirley, L. R. McGlashan, and M. Kraft. A coupled CFD-population balance approach for nanoparticle synthesis in turbulent reacting flows. *Chemical Engineering Science*, 66(17):3792–3805, 2011. ISSN 0009-2509. doi:10.1016/j.ces.2011.05.006.

[6] D. J. Aldous. Deterministic and stochastic models for coalescence (aggregation and coagulation): a review of the mean-field theory for probabilists. *Bernoulli*, 5(1):3–48, 02 1999. doi:10.2307/3318611.

[7] Ansys Fluent. User's guide version 12.0. Software documentation, 2009.

[8] M. Balthasar. *Detailed Soot Modelling in Laminar and Turbulent Reacting Flows*. PhD thesis, Lund Institute of Technology, 2000.

[9] M. Balthasar and M. Frenklach. Detailed kinetic modeling of soot aggregate formation in laminar premixed flames. *Combustion and Flame*, 140(1–2):130–145, 2005. doi:10.1016/j.combustflame.2004.11.004.

[10] M. Balthasar and M. Kraft. A stochastic approach to calculate the particle size distribution function of soot particles in laminar premixed flames. *Combustion and Flame*, 133(3):289–298, 2003. doi:10.1016/S0010-2180(03)00003-8.

[11] V. S. Buddhiraju and V. Runkana. Simulation of nanoparticle synthesis in an aerosol flame reactor using a coupled flame dynamics?monodisperse population balance model. *Journal of Aerosol Science*, 43(1):1–13, 2012. ISSN 0021-8502. doi:10.1016/j.jaerosci.2011.08.007.

[12] CD-adapco. STAR-CCM+ v7.02 Help, 2012.

[13] E. Cunningham. On the velocity of steady fall of spherical particles through fluid medium. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 83(563):357–365, 1910. ISSN 0950-1207. doi:10.1098/rspa.1910.0024.

[14] A. Eibeck and W. Wagner. An efficient stochastic algorithm for studying coagulation dynamics and gelation phenomena. *SIAM Journal of Scientific Computing*, 22(3):802–821, 2000. doi:10.1137/S1064827599353488.

[15] A. Eibeck and W. Wagner. Approximative solution of the coagulation fragmentation equation by stochastic particle systems. *Stochastic Analysis and Applications*, 18(6):921–948, 2000. doi:10.1080/07362990008809704.

[16] A. Eibeck and W. Wagner. Stochastic particle approximations for Smoluchoski's coagualtion equation. *Annals of Applied Probability*, 11(4):1137–1165, 2001. doi:10.1214/aoap/1015345398.

[17] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322(8):549–560, 1905. ISSN 1521-3889. doi:10.1002/andp.19053220806. (On the movement of small particles suspended in a stationary liquid demanded by the molecular-kinetic theory of heat).

[18] R. O. Fox. *Computational Models for Turbulent Reacting Flows*. Cambridge University Press, Cambridge, 2003.

[19] M. Frenklach. Method of moments with interpolative closure. *Chemical Engineering Science*, 57(12):2229–2239, 2002. doi:10.1016/S0009-2509(02)00113-6.

[20] M. Frenklach and S. J. Harris. Aerosol dynamics modeling using the method of moments. *Journal of Colloid and Interface Science*, 118(1):252–261, 1987. doi:10.1016/0021-9797(87)90454-1.

[21] N. A. Fuchs. The mechanics of aerosols. *Science*, 146(3647):1033–1034, 1964. ISSN 0036-8075. doi:10.1126/science.146.3647.1033-b. Translated from the Russian edition (Moscow) by R. E. Daisley and Marina Fuchs. C. N. Davies, Ed. Pergamon, London; Macmillan, New York, 1964.

[22] M. J. Hounslow, R. L. Ryall, and V. R. Marshall. A discretized population balance for nucleation, growth, and aggregation. *AIChE Journal*, 34(11):1821–1832, 1988. ISSN 1547-5905. doi:10.1002/aic.690341108.

[23] T. Johannessen, S. E. Pratsinis, and H. Livbjerg. Computational fluid-particle dynamics for the flame synthesis of alumina particles. *Chemical Engineering Science*, 55(1):177–191, 2000. ISSN 0009-2509. doi:10.1016/S0009-2509(99)00183-9.

[24] A. Kazakov and M. Frenklach. Dynamic modeling of soot particle coagulation and aggregation: Implementation with the method of moments and application to high-pressure laminar premixed flames. *Combustion and Flame*, 114(3–4):484–501, 1998. doi:10.1016/S0010-2180(97)00322-2.

[25] F. E. Kruis, J. Wei, T. van der Zwaag, and S. Haep. Computational fluid dynamics based stochastic aerosol modeling: Combination of a cell-based weighted random walk method and a constant-number monte-carlo method for aerosol

dynamics. *Chemical Engineering Science*, 70:109–120, 2012. ISSN 0009-2509. doi:10.1016/j.ces.2011.10.040. 4th International Conference on Population Balance Modeling.

[26] S. Kumar and D. Ramkrishna. On the solution of population balance equations by discretization—I. A fixed point pivot technique. *Chemical Engineering Science*, 51(8):1311–1332, 1996. doi:10.1016/0009-2509(96)88489-2.

[27] S. Kumar and D. Ramkrishna. On the solution of population balance equations by discretization—II. A moving pivot technique. *Chemical Engineering Science*, 51(8):1333–1342, 1996. doi:10.1016/0009-2509(95)00355-X.

[28] D. L. Marchisio and R. O. Fox. Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science*, 36(1): 43–73, 2005. doi:10.1016/j.jaerosci.2004.07.009.

[29] Z. A. Melzak. A scalar transport equation. *Trans. Amer. Math. Soc.*, 85: 547–560, 1957. ISSN 0002-9947. doi:10.1090/S0002-9947-1957-0087880-6.

[30] W. J. Menz, S. Shekar, G. P. Brownbridge, S. Mosbach, R. Körmer, W. Peukert, and M. Kraft. Synthesis of silicon nanoparticles with a narrow size distribution: A theoretical study. *Journal of Aerosol Science*, 44:46–61, 2012. ISSN 0021-8502. doi:10.1016/j.jaerosci.2011.10.005.

[31] N. M. Morgan, C. G. Wells, M. J. Goodson, M. Kraft, and W. Wagner. A new numerical approach for the simulation of the growth of inorganic nanoparticles. *Journal of Computational Physics*, 211(2):638–658, 2006. ISSN 0021-9991. doi:10.1016/j.jcp.2005.04.027.

[32] R. I. A. Patterson and W. Wagner. A stochastic weighted particle method for coagulation–advection problems. *SIAM Journal on Scientific Computing*, 34 (3):B290–B311, 2012. doi:10.1137/110843319.

[33] R. I. A. Patterson, J. Singh, M. Balthasar, M. Kraft, and W. Wagner. Extending stochastic soot simulation to higher pressures. *Combustion and Flame*, 145(3): 638–642, 2006. doi:10.1016/j.combustflame.2006.02.005.

[34] R. L. Pego. *Lectures on Dynamics in Models of Coarsening and Coagulation*, chapter 1, pages 1–61. Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore: Volume 9, 2007. doi:10.1142/9789812770226_0001.

[35] S. E. Pratsinis. Simultaneous nucleation, condensation, and coagulation in aerosol reactors. *Journal of Colloid and Interface Science*, 124(2):416–427, 1988. doi:10.1016/0021-9797(88)90180-4.

[36] M. Smoluchowski. Drei Vorträge Über Diffusion, Brownsche Molekularbewegung und Koagulation von Kolloidteilchen. *Zeitschrift für Physik*, 17:557–585, 1916. ISSN 0045-7825. doi:10.1016/0045-7825(74)90029-2. (Translation: Three presentations on diffusion, Brownian motion, and coagulation of colloidal particles).

[37] M. Smoluchowski. Versuch einer mathematischen Theorie der Koagulationskinetik kolloidaler Lösungen. *Zeitschrift für Physikalische Chemie*, 92(2):129–168, 1917. (Translation: Attempt at a mathematical theory of the kinetics of coagulation for colloidal solutions).

[38] K. J. Valentas and N. R. Amundson. Breakage and coalescence in dispersed phase systems. *Industrial & Engineering Chemistry Fundamentals*, 5(4):533–542, 1966. doi:10.1021/i160020a018.

[39] P. G. J. van Dongen and M. H. Ernst. On the occurrence of a gelation transition in smoluchowski's coagulation equation. *Journal of Statistical Physics*, 44(5): 785–792, 1986. ISSN 1572-9613. doi:10.1007/BF01011907.